Configuration Plc Engine

Tani GmbH

Networks in industry

.

The name Tani stands for communication in industrial production.
The focus is on communication systems:

- *OPC Server for widespread PLCs*
- *Equipment and software for connecting PLCs, SCADA systems and databases.*
- *Fieldbus diagnostic systems.*

Configuration Plc Engine

Created: Oktober 2025 in Nuremberg, Germany

Tani GmbH
Freiligrathstrasse 12
D-90482 Nuremberg
Internet: https://tanindustrie.de
Email: info@tanindustrie.de
Tel.: +49 911 980 373 54
HRB: Amtsgericht Nürnberg 29562
USt-Id: DE 289 906 852

Table of contents

1 Help overview

Configuration for OPC Server or PLC Engine Collect

2 OPC Server or PLC Engine Collect - General

This chapter provides general information about the Tani configuration:

- Overview
- License Conditions
- OPC DA Client
- Tani NetTest

2.1 Overview

Tani OPC Server

The Tani OPC Server allowing data exchange between field devices (PLC) from various manufacturers and OPC clients via Ethernet TCP/IP, RFC1006 and OSI (H1).

PLC Engine Collect

The PLC Engine Collect processes data from controllers, devices, databases independently from OPC. The core are Logic tables.

Tani Configuration

The front end Configuration allows the configuration and diagnostics if all Tani products. The GUI connects with the product over TCP/IP (Port 2468). This allows the configuration to running products OPC Server or PLC Engine Collect on other stations also.

Services

The Runtime of OPC Server or PLC Engine Collect considers from multiple services. Tani Configserver Service, Tani Logger Service and Tani OPC Server. They will be started during the computer boot. Internally the Tani Configserver service starts the dependent services.

Access

On standard all users will have all rights.
If users are configured the access will be possible for defined users only. User can have passwords. Each user can have more or less rights. This will prevent unauthorized access.

Tani Logger

The Tani logger records runtime events. The user defines the amount of logging. Logging into files is supported.

2.2 License terms

Please read the following conditions carefully before you install the software product. You must declare that you accept the following conditions either when you buy the software product or, at the latest, when it is installed. If you do not accept these conditions, send the software and the manual back to us within 14 days starting with the date of the invoice. The purchase price can only be reimbursed within the specified time frame.

I. Preamble

The object of the contract is the computer program (called “SOFTWARE PRODUCT” hereafter) stored on the data medium. The LICENSER states that the state of technology makes it impossible to make computer software so that it will work correctly in all applications and combinations. Thus the object of the contract is only a piece of software which works correctly in the sense of the program description and user’s guide. The SOFTWARE PRODUCT including all contents (e.g., figures, graphics, text and

sample applications) is and will remain the property of the LICENSER and is thus protected by copyright.
The LICENSER is willing to give user's rights as defined in this contract in return for an appropriate licensing fee.
This license contract is concluded between LICENSEE and the LICENSER by the acceptance of the SOFTWARE PRODUCT by the LICENSEE.

II. Terms

LICENSER
Tani GmbH, Freiligrathstraße 12, D-90482 Nürnberg, HRB: Amtsgericht Nürnberg 29562 USt-Id: DE289906852

LICENSEE
Natural or legal persons who purchase this license for the purpose of using the SOFTWARE PRODUCT.

THIRD PARTIES
Other natural or legal persons.

III. Scope

1. The SOFTWARE PRODUCT is exclusively licensed for use by the LICENSEE. The LICENSEE may sell the SOFTWARE PRODUCT to THIRD PARTIES under the condition that the THIRD PARTY agrees to this software licensing contract. In this case, the software license contract is invalid for the LICENSEE and the utilization rights to the SOFTWARE PRODUCT are cancelled for him since these rights have been passed on to a THIRD PARTY who has now become the LICENSEE him/herself.
2. The SOFTWARE PRODUCT may only be used on a single computer. The LICENSEE is obligated to purchase a separate license for every computer on which the SOFTWARE PRODUCT is used. A different agreement between LICENSEE and LICENSER applies to the purchase of multiple and network licenses.
3. The duplication of the SOFTWARE PRODUCT and all related documentations is forbidden except for the creation of an archive copy for exclusive use by the LICENSEE.
If the LICENSEE's original data medium becomes defective, it can be exchanged at the LICENSER for a replacement data medium. The cost of the exchange will be carried by the LICENSEE.
4. The SOFTWARE PRODUCT is delivered on data medium as a demo version with restricted function scope.
The demo version may not be used for commercial purposes. After payment of the licensing fee for the particular desired version, the LICENSEE will receive a code number from the LICENSER for the release of the corresponding SOFTWARE PRODUCT.
5. If a new program version is given to the LICENSEE due to the purchase of an update or for any other reason, the user's rights for the earlier program versions become void. This means that you may only work with the latest version.
6. The LICENSEE and THIRD PARTIES are forbidden to modify the SOFTWARE PRODUCT or reverse engineer it (i.e., de-compile or disassemble it).
The LICENSEE is liable for all damages which occur because of the violation of these conditions.

IV. Limited Warranty

The data carriers on which the SOFTWARE PRODUCT was delivered are free from material and manufacturer's flaws during normal use during a period of time of 6 months starting on the delivery date. The data carriers are free of viruses according to the LICENSER. If, however hidden viruses have snuck in, the LICENSER will not be liable for any subsequent damages which may occur.
1. The preceding limited warranty does not cover data carriers which have been damaged by chance or misuse or manipulation of unauthorized parties (people other than employees of the LICENSER).
2. During the stated guarantee period, the LICENSER is obligated to replace a defective data carrier if this was sent to the LICENSER with a copy of the invoice. Other claims, in particular damage claims,

will not be honored subject to para. 4 and Roman numeral V.
3. No further guarantee claims will be allowed. In particular, no guarantee claims will be accepted for program content and its freedom from errors or suitability for certain purposes. The responsibility for software and hardware selection, for installation, use, expected results and data protection and data backup with backup copies is the exclusive business of the LICENSEE.
4. Exceptions to the preceding warranty restrictions are promised characteristics. These promises must be in writing to be valid and accompanied by the signature of the authorized representative of the LICENSER. The LICENSER is only liable for damage claims in accordance with Roman numeral V.

V. RESTRICTION OF LIABILITY

1. The LICENSER is not liable for damages unless the damage is caused by intent or gross negligence of the LICENSER.
2. Liability due to any characteristics which may have been promised by the LICENSER is not affected by this. Liability for subsequent damages due to deficiencies which were not included in the promises of the LICENSER are excluded.
3. No liability is assumed for calculable damages, in particular lost profit.
4. Any damage claims are limited to the amount of the damage whose possible occurrence the LICENSER logically had to expect under the circumstances known at that time when the contract was concluded. In any case, liability is limited to the amount of twice the purchase price that was paid (license fee), regardless of whether claims pertaining to contract law, damage claims or other liability claims are concerned.
VI. Validity and Conclusion
1. The license given to the LICENSEE is valid until it is terminated by the licensee or the LICENSER.
2. The LICENSEE can terminate the license at all times by returning the SOFTWARE PRODUCT including the archivation copy and all related documentation to the LICENSER.
If the return is not due to a warranty claim, the license fee will not be reimbursed.
3. No liability is assumed for calculable damages, in particular lost profit.
4. The LICENSEE terminates the license by resale to a THIRD PARTY in accordance with III, para. 1.

VII. Applicable Laws

1. Regarding the laws which may apply, this license is subject to the laws of the Federal Republic of Germany and the exclusive adjudication of the German courts of law.
2. Court of jurisdiction for all legal disputes resulting from the contractual relationship and its creation and effectiveness for general merchants is Nuremberg. However, the LICENSER has the right to take the LICENSEE to court at his location.
3. Place of execution for all obligations from this contractual relationship is Nuremberg.

2.3 OPC DA Client

Two OPC test clients are shipped witt the OPC Server or PLC Engine Collect product family. One is for OPC-UA and OpcPipe tests, the other allows Windows OPC Classic tests.

Show available OPC-Server

The OPC Classic test client looks for all locally registered servers. Select one from the list.

Connect server and client

After the selection of „Tani OPC Server“ press OK. If an error message occurs:
- Check for the installation of the OPC core components. Mostly other installed software did remove some of the OPC core components.
- Check the DCOM-Settings. More information to DCOM comes from Microsoft.

Add Groups

As minimum one group must exist. From the menu select Group > Add Group. Just pressing OK is possible.

Add Items

Over the menu select Item > Add Items. Best choose the items with the small + shown in the tree.

If no symbols are be browsable (in Siemens S7, import them before) a S7 connection is defined enter in the Access Path field: "s7.mw0"

Handle Symbols
S7 Symbol Import

If the item is specified, Validate can be used to check whether the server accepts the item.

With OK the item is registered with the server. The variable is displayed in the main window with the Value Quality and Updates.

2.4 Tani NetTest

During the installation the tool Tani NetTest is installed.
It allows testing of OPC UA and OpcPipe connections.
It is very helpful for initial tests and functional tests.

3 Basics

This chapter covers the following topics:

- What is OPC?
- Tani Configuration
- Logger
- ARP
- H1 Driver

3.1 What is OPC?

OPC is the interoperability standard for the secure and reliable exchange of data in the industrial automation space and in other industries. It is platform independent and ensures the seamless flow of information among devices from multiple vendors. The OPC Foundation is responsible for the development and maintenance of this standard.

The OPC standard is a series of specifications developed by industry vendors, end-users and software developers. These specifications define the interface between Clients and Servers, as well as Servers and Servers, including access to real-time data, monitoring of alarms and events, access to historical data and other applications.

When the standard was first released in 1996, its purpose was to abstract PLC specific protocols (such as Modbus, Profibus, etc.) into a standardized interface allowing HMI/SCADA systems to interface with a “middle-man” who would convert generic-OPC read/write requests into device-specific requests and vice-versa. As a result, an entire cottage industry of products emerged allowing end-users to implement systems using best-of-breed products all seamlessly interacting via OPC.

Initially, the OPC standard was restricted to the Windows operating system. As such, the acronym OPC was borne from OLE (object linking and embedding) for Process Control. These specifications, which are now known as OPC Classic, have enjoyed widespread adoption across multiple industries, including manufacturing, building automation, oil and gas, renewable energy and utilities, among others.

With the introduction of service-oriented architectures in manufacturing systems came new challenges in security and data modeling. The OPC Foundation developed the OPC UA specifications to address these needs and at the same time provided a feature-rich technology open-platform architecture that was future-proof, scalable and extensible.

Today the acronym OPC stands for Open Platform Communications.

These are just some of the reasons why so many members and other technology organizations (collaborations) are turning to OPC UA for their interoperability platform.

Classic

The OPC Classic specifications are based on Microsoft Windows technology using the COM/DCOM (Distributed Component Object Model) for the exchange of data between software components. The specifications provide separate definitions for accessing process data, alarms and historical data.

OPC Data Access (OPC DA)
The OPC DA specification defines the exchange of data including values, time and quality information.

OPC Alarms & Events (OPC AE)

The OPC A&E specification defines the exchange of alarm and event type message information, as well as variable states and state management.

OPC Historical Data Access (OPC HDA)
The OPC HDA specification defines query methods and analytics that may be applied to historical, time-stamped data.

The OPC Classic specifications have served the OPC community well. However, as technology evolved, so did the need for OPC specifications.

In 2008, the OPC Foundation released OPC Unified Architecture (OPC UA), a platform independent service-oriented architecture that integrates all the functionality of the existing OPC Classic specifications. Several factors influenced the decision to create OPC UA:

Microsoft has de-emphasized COM (Component Object Model) and DCOM (Distributed COM) in favor of cross-platform SOA (Service-Oriented Architecture).
OPC vendors want a single set of services to expose the OPC data models, such as Data Access, Alarms & Events, Historical Data Access, etc.
To stay competitive, OPC vendors need to implement OPC on non-Microsoft systems, including embedded devices.
Other collaborating organizations need a reliable, efficient way to transport high-level structured data.
Users require the ability to access OPC servers through firewalls in a secure manner.
To learn more about the OPC UA specifications see this page.

Unified Architecture

The OPC Unified Architecture (UA), released in 2008, is a platform independent service-oriented architecture that integrates all the functionality of the individual OPC Classic specifications into one extensible framework.

This multi-layered approach accomplishes the original design specification goals of:

- Functional equivalence: all COM OPC Classic specifications are mapped to UA
- Platform independence: from an embedded micro-controller to cloud-based infrastructure
- Secure: encryption, authentication, and auditing
- Extensible: ability to add new features without affecting existing applications
- Comprehensive information modeling: for defining complex information

Functional Equivalence
Building on the success of OPC Classic, OPC UA was designed to enhance and surpass the capabilities of the OPC Classic specifications. OPC UA is functionally equivalent to OPC Classic, yet capable of much more:

- Discovery: find the availability of OPC Servers on local PCs and/or networks
- Address space: all data is represented hierarchically (e.g. files and folders) allowing for simple and complex structures to be discovered and utilized by OPC Clients
- On-demand: read and write data/information based on access-permissions
- Subscriptions: monitor data/information and report-by-exception when values change based on a client’s criteria
- Events: notify important information based on client’s criteria
- Methods: clients can execute programs, etc. based on methods defined on the server
- Integration between OPC UA products and OPC Classic products is easily accomplished with COM/ Proxy wrappers that are available in the download section.

Platform Independence

Given the wide array of available hardware platforms and operating systems, platform independence is essential. OPC UA functions on any of the following and more:

- Hardware platforms: traditional PC hardware, cloud-based servers, PLCs, micro-controllers (ARM etc.)
- Operating Systems: Microsoft Windows, Apple OSX, Android, or any distribution of Linux, etc.

OPC UA provides the necessary infrastructure for interoperability across the enterprise, from machine-to-machine, machine-to-enterprise and everything in-between.

Security
One of the most important considerations in choosing a technology is security. OPC UA is firewall-friendly while addressing security concerns by providing a suite of controls:

- Transport: numerous protocols are defined providing options such as the ultra-fast OPC-binary transport or the more universally compatible JSON over Websockets, for example
- Session Encryption: messages are transmitted securely at various encryption levels
- Message Signing: with message signing the recipient can verify the origin and integrity of received messages
- Sequenced Packets: exposure to message replay attacks is eliminated with sequencing
- Authentication: each UA client and server is identified through X509 certificates providing control over which applications and systems are permitted to connect with each other
- User Control: applications can require users to authenticate (login credentials, certificate, web token etc.) and can further restrict and enhance their capabilities with access rights and address-space “views”
- Auditing: activities by user and/or system are logged providing an access audit trail

Extensible
The multi-layered architecture of OPC UA provides a “future proof” framework. Innovative technologies and methodologies such as new transport protocols, security algorithms, encoding standards, or application-services can be incorporated into OPC UA while maintaining backwards compatibility for existing products. UA products built today will work with the products of tomorrow.

Information Modeling and Access
The OPC UA information modeling framework turns data into information. With complete object-oriented capabilities, even the most complex multi-level structures can be modeled and extended.

This framework is THE fundamental element of OPC Unified Architecture. It defines the rules and base building blocks necessary to expose an information model with OPC UA. While OPC UA already defines several core models that can be applied in many industries, other organizations build their models upon them, exposing their more specific information with OPC UA.

OPC UA also defines the necessary access mechanisms to information models.

- Look-up mechanism (browsing) to locate instances and their semantic
- Read and write operations for current data and historical data
- Method execution
- Notification for data and events

For Client-Server communication the full range of information model access is available via services and in doing so follows the design paradigm of service-oriented architecture (SOA), with which a service provider receives requests, processes them and sends the results back with the response.

Publish-Subscribe
Publish-Subscribe (PubSub), provides an alternative mechanism for data and event notification. While in Client-Server communication each notification is for a single client with guaranteed delivery, PubSub has been optimized for many-to-many configurations.

With PubSub, OPC UA applications do not directly exchange requests and responses. Instead, Publishers send messages to a Message Oriented Middleware, without knowledge of what, if any, Subscribers there may be. Similarly, Subscribers express interest in specific types of data, and process messages that contain this data, without a need to know where it originated from.

3.2 Tani Configuration

The graphical interface will be used for the configuration of the products OPC Server or PLC Engine Collect. The program uses the network port 2468 for accessing the products.

Program Start

Start the program over the operation system menu.
For the configuration of products not running on the actual system please use the station selection. It normally connects over TCP/IP. If the OSI-H1 driver is installed on the PC the configuration can use this also.

User

Each device allows the configuration of multiple users. Each user can have a password and different functionality .
If a user is configured the configuration always will request for it.
As minimum one user need to have administrator rights.

General usage and configuration

In part Usage and Configuration all of the program will be described.

Logger

Most devices are supporting a Diagnostics Logger. The logger content is shown here.

3.3 Logger

The Logger feature allows you to log the device's communication over the configured connection. You can choose to keep the log data only temporarily in the RAM or save it to a log file. The Logger records the selected data traffic of a communication going over a particular connection. Such a log can be very useful, especially for troubleshooting. Each event is recorded with a timestamp, the name of the application and the actual message.

The Logger window displays the entries that have been recorded.
To set the parts of the communication to be logged, open the Logger Configuration dialog box.
To specify whether to create the log in the RAM only or to also save it to a file, click Logger Settings in the Logger dialog box.

3.4 ARP

The **Address Resolution Protocol (ARP)** is a network protocol which allows mapping network addresses to hardware addresses.

To map an IP address to a MAC address, an ARP request is used. After encapsulating this request in a frame, it is sent out as a broadcast. All stations in the network receive and evaluate this request. They compare the IP address submitted in the ARP request with their own IP address. The station whose the IP address matches the one in the request replies by sending an ARP response.
The response is sent specifically to the station which had sent out the request, it contains the requested MAC address.

The MAC address is added to an ARP cache along with the associated IP address. The ARP cache is a temporary memory which is part of the respective station's RAM. Each time another IP address needs to be resolved to a MAC address, the station first checks whether it can do so using the ARP cache and thus without sending a new ARP request. If this is not possible, it sends an ARP request. Thus, ARP resolution is advantageous as it saves time and reduces the network load.

If an entry in the ARP cache is not accessed for a certain period of time, it will be removed. This behavior ensures that changes in the network can be detected and reflected by sending new ARP requests.

See also: ARP Cache Stati

3.5 OSI/H1 driver

To configure the devices via H1, you need the H1 driver (Tani OSI-H1 protocol). This driver is installed with the software if selected and is available for the configuration.
A detailed description is included in the Tani H1 Driver documentation.

3.6 Special Settings

There are some settings for very specific things. This changes can be done by editing ConfigServer.Settings. The file lies in the same directory where the ConfigServer exebutable is.

Lines starting with a hash symbol are comments.

Remove the comment on the parameter you try changing. Edit the specific value to your needs.
All changes take effect after restarting the ConfigServer exebutable.

Hint: Wrong settings or conflicts in the settings will result in malfunction.

```
[GeneralSettings]
# This section MUST be last for Linux
# VendorName="me"
# VendorNameLong="I am the vendor"
# ProductName="MyProduct"
# ProductNameLong="My Product"
# ProductVersion="1.0"
# CertificatePath="/path/to/the/certificates"
# SymbolPath="/path/to/the/symbols"
# IniFile="/path/to/the/inifile/MyInifile.ini"
# UserDbPath="/path/to/the/userdb"
# VarRunPath is the path where the sockets reside. If not set it is "/var/run" for Linux and "\\.\pipe\Wmk" on Windows
VarRunPath=/var/run/Tani
# NetworkNamespaceConfig - if it is empty the default or possibly only namespace is used
# NetworkNamespaceConfig="MyNetworkNamespace"
# The param tsap must be eight letters long. The default is "WMKKTani"
# ParamTsap="ParaTSAP"
# ParamPort is the port number the connection between the configuration clients and the config server is using. Default is 2468
# ParamPort="2468"
```

SetStationAllowed enables the network adapter parameter settings and the station name. On standard this is disabled.
SetStationAllowed="yes"
PrereleaseVersion can be "0" or "false" or "no", or "1" or "true" or "yes". It can be any text also, this defines a prerelease version also
PrereleaseVersion="MyPrerelease - not for resale"
ParamUseSSL can be "never", "default" or "always"
ParamUseSSL="default"
ParamUseZlib can be "never", "default" or "always"
ParamUseZlib="default"

4 Connections

This chapter covers the following topics:

4.1 S7 General

The S7 protocol is used to communicate with S7 PLCs and with CPs capable of using the S7 protocol. The S7 protocol can be used in conjunction with TCP/IP, H1, or MPI.

Read (Fetch active) and Write (Write active)

- When communicating with S7 PLCs, it is generally possible to both read and write data over a single connection.
- Usually, the parameterization of the S7 protocol is based on RFC 1006.
- Besides the S7 PLC address (IP address or MAC address), a read and write TSAP is required. When using MPI, enter the MPI address of the CPU.
- To read and write data from and to an S7 PLC (layer 7 communication), the following can be used:
  - non-parameterizable connections
  - parameterizable connections

Non-parameterizable connections

Only on the Engine side will a connection be set up and NOT on the PLC side.
Standard connections are required, which are handled via standard TSAPs or standard ports.
There are only a limited number of standard connections available. If these are not sufficient, "parameterizable connections" must be used.
Standard connections to Siemens CPs are handled via RFC1006 or H1 (see TSAP, TSAP for S7 Connection and Routing TSAPs.

Parameterizable connections

The connection must be configured on both sides of the communication (a fetch / write active connection in the Engine AND a fetch / write-passive connection in the S7-CP).
If the communication is via the Engine, parameterizable connections are required.

Note:
For communication with S7 controllers, both the S7 Item Syntax and the S5 Item Syntax can be used.

See also S7 Symbol Import

4.2 S5 General

The S5 protocol is used to communicate with S5 PLCs (S5 AP headers) and with CPs capable of using the S5 protocol.
The S5 protocol can be used in conjunction with TCP/IP, H1 or for serial communication with AS511.

Read (Fetch active):

- To read data actively from an S5 PLC, you need to establish a read connection in the device.
- Besides the S5 PLC address (IP address or MAC address), a read port or a read TSAP must be specified.
- The read connection needs to be parameterized on both sides of the communication (Fetch active connection in the OPC Server, Fetch passive connection in the S5).

Write (Write active):

- If you want to write data to the S5 PLC as well, a write connection needs to be established besides the read connection. When using S5 via TCP/IP or via AS511, a single connection may be used for this purpose.
- For the write connection, a write port or a write TSAP must be specified.
- The write connection needs to be parameterized on both sides of the communication (Write active connection in the device, Write / Receive passive connection in the S5).
- Newer S5 TCP/IP cards and devices support reading and writing over a single connection.

Read the hint for bit writing.

4.3 Modbus General

The Modbus over TCP protocol is used for the communication with PLCs from Wago, Groupe Schneider, Beckhoff and all CPs capable of using the Modbus over TCP protocol (Modicon, Ethernet terminals from Phoenix, Wago, Beckhoff and many more).
The Modbus protocol only works in conjunction with TCP/IP.

Read (Fetch active) and Write (Write active):

- When communicating with Modbus PLCs, it is generally possible to both read and write data over a single connection.
- If default port 502 is used, you only need to parameterize the connection on the device side. On the PLC side, no other settings are required.

NOTE:
For the communication via Modbus over TCP, a specific Modbus Item Syntax is used.

See also Modbus Symbol Import.

4.4 ControlLogix/CompactLogix General

The Rockwell protocol is used for the communication with ControlLogiX / Compact Logix, Micro series, SLC and PLC5 PLCs from Rockwell Automation.
The Rockwell protocol only works in conjunction with TCP/IP.

Read (Fetch active) and Write (Write active):

- When communicating with Rockwell PLCs, it is generally possible to both read and write data over a single connection.
- If default port 44818 is used, you only need to parametrize the connection on the device side. On the PLC side, no other settings are required.

NOTE:
To address item IDs, you need to specify their symbolic names in the Rockwell.

Alias browsing:

Once the Rockwell connection has been established, symbol information will be read from the PLC and displayed for selection in the corresponding dialog boxes.

See also Item Syntax for symbolic PLCs.

4.5 PLC-5 / SLC General

This protocol is used for the communication with PLC-5 and SLC PLCs from Rockwell Automation Allen-Bradley.
The PLC-5 / SLC protocol only works in conjunction with TCP/IP.

Read (Fetch active) and Write (Write active):

- When communicating with SLC or PLC-5 PLCs, it is generally possible to both read and write over a single connection.
- If default port 2222 is used, you only need to parameterize the connection on the device side. On the PLC side, no other settings are required.

NOTE:
For PLC-5 and SLC communication, a specific PLC-5 / SLC Item Syntax is used.

4.6 Mitsubishi General

The MELSEC-Q protocol is used for the communication with Mitsubishi PLCs from the MELSEC-Q, QL and FX5 series.
The MELSEC-Q protocol only works in conjunction with TCP/IP.

When communicating with MELSEC-Q PLCs, it is generally possible to both read and write data over a single connection.
The connection needs to be configured on both sides of the communication (one connection in the device and one connection in the MELSEC-Q PLC).
By default, port 8192 is used. The same port number must be used on both sides of the connection.
If the MELSEC-Q PLC uses non-parameterizable connections with fixed ports (“default connections”), you only need to parameterize the connection on the device side. On the PLC side, no other settings are required. Please make sure that the device uses the same ports as the MELSEC-Q PLC.

NOTE:
In Mitsubishi PLCs, addresses are often given in HEX notation (in this case, you need to convert the port numbers to decimal notation for the device side).

NOTE:
For communication with Mitsubishi PLCs, a specific Melsec-Q Item Syntax is used.

The apply as well Melsec-Q Protocol settings.

4.7 KNX General

KNX is a building protocol. Mostly it is used for light settings.

4.8 BACnet General

BACnet is often used in building automation.

See also BACnet Protocol

4.9 Send/Receive General

In a Send / Receive communication, the data is transmitted without application headers, i.e. as raw data. Any station that supports the Send / Receive direct interface can be used as the communication partner. The Send / Receive protocol works both with TCP/IP and H1.

Receiving and sending data

- It is generally possible to both send and receive over a single connection.
- Besides the address of the Send / Receive PLC (IP address or MAC address), you need to specify a port or TSAP.
- Using the echoaktiv function, it is possible to perform active jobs via AS511 over the otherwise passive programming port. For further Information, see the echoaktiv section.

NOTE:
For Send / Receive communication, a specific Send/Receive Item Syntax is used.

4.10 Database General

A database connection needs additional to the IP parameters more information. These details are different from one database to the other.

Mostly all databases need a user and password for accessing it. All other parameters are different between the databases. Some databases will need special settings in the database itselve like detail settings or licenses.

MySql
The MySql database knows schema. A standard schema can be defined in the database. If doing so the schema field can be left empty.
More details are encryption, packaging and more. Detailed information to these parameters are described in the original database manuals.

Microsoft SQL
The MS-SQL database uses instances. So an instance must be given always. If the required parameters can not checked online please ask your database system operator for the details. Do not forget entering the port number which is different between instances.
The Sybase database uses the same protocol as Ms-SQL. Normally all will work, but it is more secure checking "Enable Sybase compatibility" if Sybase is used.
A hint: Browsing works in Microsoft databases only if the user has administrator rightss.

There are two methods to connect to a server:

1. IP-Address or Domain Name an Port
2. Microsoft Domain and Share name. Enter this as "domain\share".

Postgres SQL
A user, password and database need to be configured. Postgres uses the database as a security feature, so it can not be browsed.

ODBC
Under Windows please configure the ODBC data source as system DSN. PLC Engine Collect works as a service, so user DSN can not be accessed..
Configure the ODBC driver completely..

Browsing
Use the browse functionality if ever possible. This will need online access to the database. In the database core the users will need the rights for browsing.

In case of questions ask your database administrator.

Most databases are giving detailed error information. They will be written into the diagnostics logger.

Cache
If data are written only into the database the data can use a file cache if the database connection is broken.
The data are stored into the given file.
Often files lying on flash devices. Such flash memory can not be written very often. So PLC Engine Collect has an additional memory cache before it writes the data into the file. The memory cache has 1MB in size. If the memory runs over before the time limit is reached the data will be written into the real file earlier.
If the database connection will work again the file data will be sent to the database. If all data are send and verified by the database the file will be deleted.

See also: Enable file caching in Database request

4.11 OpcPipe General

OpcPipe is a tunnel protocol that tunnels the data exchanged in an OPC communication between a client and a server.
OpcPipe comprises two parts:

OpcPipe server
Runs on the same computer as the OPC Server or PLC Engine Collect device.

OpcPipe client
Runs on the same computer as the OPC client.

The OpcPipe client receives the request from the OPC client and converts the OPC communication to a “normal” TCP/IP communication. The data can thus be transmitted in a tunnel to the destination computer or device via the network by using a standard TCP/IP connection. There, the OpcPipe server receives the data, “decrypts” it by converting it back to OPC communication and forwards the data to the OPC server or the protocol logic of the device. The server or device performs the requested action and sends the data back to the OPC client. The tunnel mechanism works the same way in both directions.

Establishing an OpcPipe connection

The OpcPipe connection must be configured both on the server side (computer or device) and the client side (computer where the OPC client is running).

For this purpose, the OPC server with OpcPipe functionality must be installed on both sides of the communication. Usually, the OpcPipe client initiates the active connection. The default port number is 4444.

4.12 OPC-UA General

OPC-UA is an up to data protocol for data exchange in plants. It handles a lot of functions, and it supports encryption.
In Industrie 4.0 it will be used frequently.

OPC-UA Server
Offers OPC-UA functionality to others

OPC-UA Client
Uses OPC-UA from others.

OPC-UA Connections

OPC-UA uses TCP/IP as base. The connection must be configured on both sides. The default port is **4840**.
If multiple OPC-UA servers are running on one machine different ports need to be used.
OPC-UA has some extensions doe the connection establishing. The details of it will be find in the manual of the server.
The Tani OPC-UA server does not require any specific connection parameters.

OPC UA can be operated securely. In this case, certificates are used. More information see OPC UA Certificates.

OPC-UA allows file access. This need to be enabled in the server.

4.13 OPC-DA General

An OPC-DA Client connection communicates with an OPC Server and is the active part.
Values can be read and written, and the data structure can be determined by browsing.

It is recommended to connect to an OPC-DA Server on the same computer.
A tunnel connection is more stable via OPC-UA or OpcPipe, easier to configure and has security mechanisms.
In addition, the DCOM settings are complicated and different between the different operating systems.

4.14 MQTT General

MQTT is a simple and easy protocol known from the IoT (Internet of Things) world.

The active part ist named "Client", the passive part "Broker".

A broker can support many clients. Each client needs a unique station name.
On standard the clients are writing data into the broker. The broker dispatches the data to other clients if they will need the data. So one station will dispatch the data to multiple partners.

Security can be used. If used an user name and password is required.

The quality of service allows stable communication in unstable networks also. Messages can be sent without acknowledges (QOS0), or an acknowledge ist used (QOS1). QOS2 uses a four way handshake. This is more stable but can be slowly in unstable networks.

The two different protocol versions are only different in the allowed station name length.

MQTT does not define data types. It can handle any data.
For special situations as offer MQTT data in OPC some optional settings exist:
- any data (default)
- all data are texts
- handle the Item Syntax neutral.

4.15 IEC60870-5-104 General

This protocol sometimes will be called IEC 104.
It is a telecontrol protocol.

4.16 Email General

It is possible to send an email with a Logic Table.
Therefor an email connection is needed.

See: Email Settings

4.17 Redundancy

Redundancy connections to controllers will prevent data loss in case of connection interrupting.

A controller needs to have two network adapters. The computer need to have two network adapters, too.

For redundancy connections two or three sets of connection parameters will be defined.

At runtime all connections will be established. One will be master, this will handle the data.
The second connection will be the reserve. Its status will be checked only.
There are some options for the redundancy:
- Check the connection for running
- Check the controller run status
- Check a watchdog element for periodically changes

In case of interrupting the master all elements will be switched to the slave.

This will work transparently to the OPC application. The items don't deliver errors.

Over OPC the status of both connections can be checked. Details here.

Technical Details

There are two modes of operation
- Dynamic master
  One connection becomes the master. If it breaks down another connection becomes the master
- The first connection is master always. If it breaks down a slave will handle the data. If the master connection is available again it becomes master again.

Switch connections

For testing purposes the active connection can be switched by the OPC client.

- Dynamic master
  The connection switched to becomes the master
- Static master
  It will be switched to the slave. After one minute it will be switched back to the master if it will work.

Hints:

- Redundancy connections can not be simulated.
- Redundancy connections do not become suspended it no item uses them.
- Each single connection of the redundancies will use one connection from the licensed pool.
- The controllers used in a redundancy connection needs to be from the same family.

See also Redundancy Settings

4.18 OPC UA event priorities

Events haben oft eine Priorität. Im OPC UA Jargon wird das "Severity" genannt.

Die Regeln:
0 ist die niedrigste Priorität, 1000 die Höchste.

Bekannte Zwischenstufen sind:
Klein (1 bis 333)
Mittel (334 bis 666)
Hoch (667 bis 1000)

Viele Systeme kennen 15 Prioritäten. Dann gilt:

Hoch	15	1000
	14	955
	13	910
	12	865
	11	820
	10	775
	9	730
	8	685
Mittel	7	650
	6	575
	5	500
	4	425
	3	350
	2	300
Niedrig	1	150
	0	1

5 General usage

This chapter covers the following topics:

- How to Proceed
- Basic Configuration
- Licensing
- Online Diagnostics
- S7 Symbol import
- Modbus Symbol Import
- Edit Symbols
- User Settings
- OPC UA Certificates

5.1 How to Proceed

The OPC Server or PLC Engine Collect are very easy to configure in just a few steps. To set up communication, proceed as follows:

Basic Configuration

To be able to communicate with the Engine device via TCP/IP, the device needs an IP address. How to assign the address is described in the Basic Configuration section.

Configuring the connection

For every hardware from which you want to read data or to which you want to write data, you create a connection to the device by using the Tani configuration software. This connection defines the access path to the hardware, i.e. the channel between the two communication partners. All the parameters needed for the communication between the device and the hardware are defined in the respective dialog boxes. The protocols that are supported by the communication partner (e.g. S7 protocol and RFC 1006) are provided for selection. To access specific hardware, you specify the destination addresses (e.g. IP address and port numbers, or MAC address and TSAP). You can also choose whether or not writes are basically allowed over this connection.
See the New Connection section.

Connect OPC client to OPC server

Before the OPC server and OPC client can exchange data with each other, a connection must be established between the two. The initiative always starts from the OPC client, which uses the "Connect" function to connect to the server. The access rights are set via the DCOM settings.
The Prog-ID, or name of the OPC server is called:

Tani OPC Server

Configure OPC client

In the OPC client, the data points are either selected via tag browsing or entered via a name. The data points address the hardware via the item syntax. The addressing can be absolute or symbolic. For addressing, Access Path Name and Item Name are specified. The Access Path Name corresponds to the connection name (see OPC Server Configuration). For absolute addressing, the protocol-specific item syntax must be adhered to.
See Item Syntax.

Symbolic addressing

For symbolic addressing, the symbol file is used in the OPC Server.

The symbols can either be entered manually (see Symbol List) or imported via Symbol Import.
See also S7 Symbol Import and Modbus Symbol Import.
The defined symbolic item names are then displayed automatically (prerequisite: OPC client supports alias browsing).

OPC Test Client

For initial tests, the OPC client installed together with the OPC Server can be used. This client is very easy to use, so that first data points can be read from the hardware immediately.
See OPC client

Diagnostics

To diagnose the connections, you can use the Diagnostics window, which is described in the Online Diagnostics section.

Logger

The Logger is provided for recording error events that have occurred in the device.

5.2 Basic Configuration

During the basic configuration, the device is provided with all the parameters needed for communication. The device is registered on the network.

- Start the configuration configuration software.
- Select the type of configuration (usually IP).

NOTE:
The station you are configuring via H1 or IP has to be in the same network as the configuration computer. Stations connected behind a router cannot be discovered.

Configuration via H1 is only available if the Tani OSI-H1 Driver is installed.

Configuration via TCP/IP or H1

- The Select a Station in Your Network dialog box appears. The new device is displayed together with its MAC address.
- From the MAC address, you can determine the station to be configured. Double-click the station or select it and then click OK.
- The Station Parameters dialog box appears. Edit the settings and confirm your entries with OK.
- The configuration is complete.

NOTE:
If you want to use both network connections provided by the device, both configurations should be configured.

5.3 Licensing

The devices are delivered with the functionality that was specified in the order. If, during commissioning or later when adding extensions, you find that you need to expand the functionality, you can do this by activating a license.
In the License Overview dialog box, you can see which functionalities are licensed and which not. The configuration displays a License Request Code that you can send to Tani GmbH. You will then receive

an activation code that enables the desired functionality. How this is done is described in the License Overview and Licenses sections.

Please note the license conditions.

5.3.1 License Overview

Displays a list of the available modules with order number and license status. A function can be enabled by entering an activation code. This allows adding functionality to the device without having to return it to the manufacturer.
Show License
To edit an entry, double-click it or use the button. The Licenses dialog box appears.
To edit licenses, master access is required.
Refresh List
This button reads the license status from the device and refreshes the displayed status.

A hint:
A connected dongle is recognized by the software only if it is connected during the software start. Under Windows the software runs as a service, under Linux as a daemon. So the dongle need to be connected during power on.

The list of available modules with order number and license status is shown. A function can be unlocked via an activation code. Thus, the device can be extended by functions without being sent to the manufacturer.

Request license details
The license selected in the left list can be viewed and edited.
Opens the dialog License activation

Enable freeware license
Enables switching to the OPC Server Tunnel, which does not require a license.

Enter license codes
An entry can be edited by double-clicking or with the button. The dialog License opens.
Master access is required to process licenses.

Switch demo license
It is switched to the license selected in the left list. All functions that would be possible with this license are then available. The others are not anymore. So the plant can be put into operation, and if everything goes well, this license can be bought.

Refresh list
The button reads the license status again from the device. This allows the status to be displayed again.

List of dongles
The Plugged dongles dialog shows all Wibu dongles plugged in and recognized by the device, which provide licenses for OPC Server or PLC Engine Collect licenses.

A note:
A plugged dongle is only used by the running software if it is present at the start of the software. The software runs as a service (Windows) or daemon (Linux). So the dongle at power on must already be plugged in.

5.3.2 License activation

The dialog is used to request a software activation.
The list shows all licenses that are possible for the previously selected product.

To request a license activation, please fill in the fields
Company name
Customer Name
Authorization code, the code that Tani GmbH has communicated to you on the invoice / delivery note, so that the licensing is clear.

Request Trial License
If a temporary license is required to test the server extensively, please select this option. There will be no authorization code required.

Generate activation request
The button opens the License dialog. The list is displayed. It is now possible to send this information to Tani GmbH, via email directly to info@tanindustrie.de or via the clipboard.
Tani GmbH then sends the activation code, which can then be entered in the License window.

5.3.3 License

The dialog allows the following:

1. Sending the request code directly by email directly to info@tanindustrie.de or via the clipboard.
Please click the button "**-> Clipboard**" or "**-> Email**" or "**-> Online**".
e.g .:

```
-----------------------------------------
Customer: Muster GmbH
User: Dipl.-Inform. Uwe Mustermann
-----------------------------------------
Product: OPC Server Standard Siemens
Order No: 101-0011-01
Authorization: 123456
Request: AIYUZ5I0YY7MFKK1XQLR5SISG5A
Confirm:
=========================================
```

2. Entering the activation code that has been sent by Tani GmbH. Please press the button "Save" if the content of the window contains the activation code, for example:

```
-----------------------------------------
Customer: Muster GmbH
User: Dipl.-Inform. Uwe Mustermann
-----------------------------------------
Product: OPC Server Standard Siemens
Order No: 101-0011-01
Authorization: 123456
Request: AIYUZ5I0YY7MFKK1XQLR5SISG5A
Confirm: XK1XQLR5SISG5IYUZ5I0YY7MFKX
=========================================
```

NOTE:
The request code is unique for the device!
The device accepts only valid activation codes. Invalid can not be saved.

To use the online license activation an account at Tani is needed. With this account you can activate your product on the Tani website also. This may be used if the machine running the product does not have access to the internet.

The license information is stored in the file **LicenseKey.txt**.

Under Windows the file resides in **c:\Users\Public\Documents\Tani**.
Under Linux this is **/etc/Tani**.

5.3.4 Plugged dongles

The list is containing all dongle from company Wibu.

The list actualizes automatically.

5.3.5 License Login

Please log in with the user name and password that you also use when logging in on the website https://tanindustrie.de.

If you do not have an account yet, please send an email to: info@tanindustrie.de

Please have the invoice ready where the "Software Authorization Code" is printed.

If the order number specified there matches the previously selected one, the OPC Server or PLC Engine Collect is successfully licensed.

5.4 Online Diagnostics

The online diagnostics for the device are displayed in the main window. It shows the configured connections and the associated diagnostic information. The overview enables you to quickly check whether everything is OK or whether problems have occurred.
You can toggle the main window between the Connections list and the online diagnostics.

The columns have the following meanings:

Type	Type of the connection
Name of the Connection	Name assigned to the connection
Typ	Connection type The status of the read connection is displayed.
Dr In	Disconnect counter
Dr Out	Disconnect counter
Send In	Frame counter
Rec In	Frame counter
Send Out	Frame counter
Rec Out	Frame counter

If a connection is disabled, it is grayed out and a number of parameters from the Connections list are displayed.

The contents are cyclically refreshed.

5.5 S7 Symbol Import

The symbol import is needed for the S7 200 / 300 / 400 / 1200 only.
The S7 1500 uses the symbols directly. No symbol import is necessary.
Projects from Step7 and from TIA Portal can be imported.
The import of S7-1500 is not necessary because the symbols are present in the PLC and are read out online.

In the dialog box, you can import a symbol from an existing PLC program.

S7 Project Filename
This field shows the name of the S7 project in which the individual PLC programs are stored. Use the >> button on the right to open a file selection box and browse the directory structure of the hard drive or the network.

Internal S7 Project Name
Displays the internal S7 project name.

PLC Program
The list box displays all available PLC programs of the S7 project selected above. When you choose a PLC program, the tree view on the right displays the programmed data structure.

PLC Blocks
The tree view shows the items used in the selected PLC program. By double-clicking a symbol, you can apply the entry to the previously opened dialog box.

Data blocks with symbolic names
If the check box is selected, the DBs are imported with the symbolic name, if not, with the corresponding DB number.

Selected Import
After all selection is finished, the button starts the symbol import.

Delete symbols before import
All existing symbols are deleted before the import is started. If the option is not selected, the existing symbols are smart merged with the imported ones.

5.6 Modbus Symbol Import

It is possible to import the symbols from a Wago Codesys project via the exported symbol file *.sym.
This file is created with the PLC programming software.

Select that *.sym file first.
Afterward select the symbols to import.
<Cntrl>-A selects all visible entries.

5.7 Symbol List

The server offers the possibility to use a symbolic name for a specific direct address in the PLC.
The **Symbollist** dialog shows the existing symbols and allows you to create, edit, and delete individual symbols, as well as the symbol import.

Left window
In the tree view can be navigated through the entire symbol file. The connections are displayed. Any hierarchies can be created.
A context menu (right mouse button) allows you to insert, delete, edit.

Right window
The symbols of the selected entry of the tree view are displayed.
A double click in a list entry will release it for editing.
A context menu (right mouse button) allows you to insert, delete, edit, as well as the symbol import.

Import into selected topic
Opens the S7 Symbol Import or Modbus Symbol Import dialog and allows you to transfer the symbols from an S7 project or Wago Modbus Export File that has been created with the programming software.

New symbol
Opens the Add Symbol dialog and allows you to add a symbol.

Edit symbol
Opens the Edit Symbols dialog and allows you to edit the selected entry.

Delete symbol
Deletes the selected entry from the list, without query.

Note:
Editing the symbols for Rockwell controls is NOT possible !!!

Global symbols
It is possible to create symbols that are present in all connections of the same type.

5.8 User Settings

Users with different access rights can be defined.

If no user is defined all users will have unrestricted access. No question for user or password occurs.

If users are defined one user must have full access. The configuration is checking this.

The following access rights are possible:
- System access
- FTP server access
- Web server access
- SSH access

Each access can allow writing separately.

If users are defined unauthorized access is not possible. Each user must log on.

If the password is lost there is only one accessing the system again: Do directly access to the file system of the server with login into the operating system. Then remove the file /etc/Tani/userdb.sqldat (Linux) or $WINDIR\Public\Documents\Tani\userdb.sqldat (Windows). Restart the server services. Now no users are defined and without login all will have full access again.

5.9 OPC UA Certificates

OPC UA can be used in a secure and encrypted mode. Mostly this needs certificates.

A certificate is a technical passport. Each station has a certificate.
An OPC-UA secure connection needs the both certificates of the partners.

The following certificates exists:
- Own certificate. It is used on network access from other stations in the network.
- Own client certificate. It is used in active client connections to other stations.
- Trusted certificates. This is a list of certificates which will be trusted.
- Non trusted certificates. The list contains valid certificates which are not trusted.
- Revoked certificates The list contains certificates which are revoked by the issuer, but they are valid for checks.

Each certificate has an issuer, a time period in which it is valid, and more information as details it may be used for, type of signing and more.

If you try trusting a certificate please call the issuer and check weather you can trust him. This decision can be tricky if the issuer is not well known by you. Possibly you can check the issuers entourage, ask if other people trust him.

Move all certificates you trust into the list of trusted certificates.

If you notice that a certificate is revoked - this can be by a phone call or an email - move the certificate to the list of revoked certificates.

On standard only certificates from the "Trusted certificates" will be trusted. For startup a plant the setting can changed to "Accept non trusted certificates" in the general system settings.

The management of certificates needs accuracy. Look for descriptions how certificates are managed securely, this is no simple job. In big environments it may be easier to create a chain of trusted certificates, ask your IT department for this. There are several service providers on the market doing this for you. In this case you need trust the service partner.

6 Menu

The section describes some menu items.
Some called menu items do not allow online help.

File

Choose Station
Connects to the OPC Server or PLC Engine Collect on a remote computer via TCP / IP port 2468. Select a station

Choose Station over H1
Connects to the OPC Server or PLC Engine Collect on a remote machine via OSI H1. Select a station

Local Connection
Connects to the OPC Server or PLC Engine Collect on the computer where the configuration is running.

Load Configuration File in Device
Allows to load a saved configuration into the currently selected Runtime OPC Server or PLC Engine Collectt. The parameters contained in it are overwritten.

Store Device Configuration in File
Allows to save the parameters of the Runtime OPC Server or PLC Engine Collect currently selected online in a file. This is a complete backup of the runtime system.

Download user file from remote device
Allows a file transfer from the online selected Runtime OPC Server or PLC Engine Collect. This is usually the case with remotely connected Runtime OPC Server or PLC Engine Collect where no file transfer is possible otherwise.

Upload user file to remote device
Allows a file transfer to the online selected Runtime OPC Server or PLC Engine Collect. This is usually the case with remotely connected Runtime OPC Server or PLC Engine Collect where no file transfer is possible otherwise.

Close the window
Ends the configuration
...

Station

Certificate Settings
The dialog allows you to manage the certificates. OPC UA Certificates

General System Settings
The dialog allows the setting of general valid settings of the Runtime OPC Server or PLC Engine Collect. General System Settings

OPC UA System Settings
The dialog allows the setting of the OPC-UA behavior, as well as the setting of the Discovery service. OPC UA System Settings

Settings

Configuration Settings
Allows the settings of the configuration, not the Runtime. Configuration Client Settings

H1 System Values

Handles the behavior of the H1 Protocol. H1 System Settings

TCP/IP System Values
Handles the behavior of the TCP/IP Protocol. TCP/IP System Settings

Remove Complete Configuration
Allows a reset and wipe of all user settings.

Import configuration from INAT and Softing OPC or DDE Server
Allows upgrading from the INAT and Softing OPC or DDE server to the OPC Server or PLC Engine Collect.
The configuration files and symbols are transferred to the new format and loaded into the Runtime.
Import configuration from INAT and Softing OPC or DDE Server

Help

Versions
Shows the versions of the online selected RuntimeOPC Server or PLC Engine Collect.
Not to be confused with the version of the just started configuration. Versions

Load path
Shows the full path to the loaded modules. Helps with troubleshooting. LibraryPaths

Device Licenses
Allows viewing, switching and entering licenses. License Overview

About
Shows the versions of the configuration.
Not to be confused with the version of the online selected Runtime OPC Server or PLC Engine Collect.
About

7 Windows

The configuration offers following windows:

- Tani Configuration
- Main window
- Connection List
- OpcDiagnostics
- Logic tables
- Edit Symbols
- Status variables
- Logger

Further functions are described in the following sections:

- Dialogs
- Connections
- Logic tables
- Menu

7.1 Tani Configuration

The Tani configuration is required to configure the devices. The software is used to make basic settings, create and diagnose connections and display data recorded by the Logger feature. To configure configuration for a device, use one of the following methods:

Configuration via TCP/IP

A socket connection to the device is established via TCP/IP. Any entries you make are transmitted directly to the device and stored there. This method allows you to configure the device from anywhere within the TCP/IP network.

Configuration via H1

A configuration via H1 is only possible if the H1 protocol driver is installed. An H1 connection is established to the device you want to configure. Any entries you make are transmitted directly to the station and stored there. This method allows you to configure the station from anywhere within the H1 network.

Offline configuration

You can create a configuration file offline on the PC and transfer it to the device. This is not the common way of configuration. It is recommended to configure the device online until everything works properly. Then, transfer the parameters from the device to the PC and save them there.

Language selection (buttons)

Use the buttons to select the desired user interface language. After a restart, configuration will display all texts in the selected language. Please note that this does not toggle the texts displayed in the Logger.

7.1.1 Station Select

This dialog allows the selection of the station to handle

- available over the network. See Select remote station
- Handle the local pc running software.

7.1.2 Select a station

When you perform a **configuration via TCP/IP** or a **configuration via H1** the **Select a Station in Your Network** window opens.

All stations with an arrow (->) at the beginning of the row are currently online in the network. They are detected automatically by configuration. The software displays all devices and PCs with servers.

NOTE:
Stations connected behind a router cannot be discovered automatically. For these, you need to define a direct connection using the New button.

Selecting the desired station
When you double-click the desired station (or click the OK button), configuration will connect to the corresponding station. The connection attempt has a timeout of about 3 seconds.
Successful connection: the Connections main window opens.
Unsuccessful connection: If there is no reply from the station within 3 seconds, the connection attempt aborts.

New Station
If the station is not listed in the automatic online display (parameterization via IP) because it is located outside your network, you can click this button to configure a direct connection to this station in the Edit Access to a Station in Your List.

Edit
Using this button, you can change the parameters for the station you created using the Edit Access to a Station in Your List dialog box.

Delete Station
Using this button, you can delete the parameters for the station. A prompt asks you to confirm that you really want to delete the parameters. **Deletions cannot be undone**.

Rescan stations [Network Adapter]
Starts the station scan and the automatic discovery of online stations and servers. This scan will only find stations that are accessible in the local network.

Settings
The dialog Configuration Adapter Settings allows you to setup the network card and the timing.

Language
The dialog allows you to toggle the space of the configuration.

7.1.2.1 Configuration Adapter Settings

The dialog is used to specify which adapter the configuration is to perform the automatic station search.

Timeout for connections
With the connection timeout, the default timeout can be changed for this connection during the station search of 3s. This is e.g. For connections on the Internet, in which the life telegrams have been deactivated. The reaction time in the case of connection disturbances is thus increased (for example, in cable breakage).

Use Timeout
This button should only be activated if the default connection timeout is too short or a very slow transfer

connection is present. In normal circumstances, however, it is switched off.

Show all addresses of a station
In the Station Search window, all the station addresses are displayed, even if the configuration for this address can not be established.

Search over all adapters
A station search is performed over all installed network adapters

Active adapter
The selected adapter is displayed

Available network adapters and addresses
The list shows all available adapters. The desired one is selected here.

7.1.3 Add a station

If a connection to the desired station was not possible and the Station Not Found dialog box appears, you can configure a direct connection to this station there. The entry will then be added to the list of available stations and displayed in the Select a Station in Your Network dialog box. After the station scan has been performed, an arrow symbol (->) preceding a station indicates that this station is available online.

Station Name
The name to be displayed in the list.

H1
Establishes a connection via the H1 protocol. Specify the MAC address and the TSAP.

TCP/IP
Establishes a connection via the TCP/IP protocol. Specify the IP address and the port number.

Extended
Extends the dialog box so that you can enter additional parameters.

H1 Settings

MAC address
The MAC address of the station to be configured and diagnosed.

Use Default TSAP / Special TSAP
Here, you can indicate the TSAP for H1 or RFC 1006 connections.

TCP/IP Settings

IP Address
Here, you can indicate the IP address or the DNS name of the station to be configured and diagnosed. 127.0.0.1 addresses the local device.

Use Default Port / Special Port
The OPC Server or PLC Engine Collect uses port 2468.
The connection is encrypted with TLS.

Standard
Reduces the dialog box showing only a few parameters.

Optional Product Select
These entries will be ignored.

7.1.4 Station not found

The dialog box shows the message **A timeout occurred**.
If a station cannot be found, this can be due to the following reasons:

OPC Server or PLC Engine Collect
- The server service has stopped.

Configuration via TCP/IP or Configuration via H1
- The selected station is switched off.
- A network cable is not properly connected or faulty.
- A switch, hub, router or gateway is switched off.
- The station is connected behind a router and therefore not accessible.
- On the operating PC, the protocol to be used for connecting to the station (H1 or TCP/IP) has not been installed or configured properly.
- The H1 or TCP/IP protocol on the remote network station has not been installed or configured properly.
- A station (e.g. a reference router) has disabled the forwarding of the frames.
- The firewall has not been configured properly.

NOTE:
Stations connected behind a router cannot be discovered automatically. For these, you need to define a direct connection using the New button (see Edit Access to a Station in Your List).

Serial connection
- The connection cable is not plugged in.
- The pins of the connection cable are not assigned correctly. The connection to the station only uses three wires: Send data, receive data, signal ground.
- The interface adapter installed in your PC is faulty.
- The interface of your PC is used by another task.

7.2 Main window

The main window consists of two parts. The left panel shows all online stations. The right panel shows the parameterized connections of the station selected in the left panel, i.e. the Connections list.
When you select a station in the left panel, the right panel shows the associated connections.

Double-clicking a connection opens the dialog box where you can edit the connection.

Right-clicking opens a context menu that provides more functions.

The menu bar provides all functions required to parameterize the devices.

Frequently used functions are available on a toolbar where they can be accessed by a single click.

Press the **ESC** key to exit the window.

7.3 Connection List

The Connections list of the device is displayed in the main window. It shows an overview of the configured connections including their parameters. The connections are displayed in different colors to indicate their current status: active (**black**) or inactive (**gray**).
To edit a connection, double-click it. To do so, you need master access.
You can toggle the main window between the Connections list and Online Diagnostics by clicking the ***Diagnosis - Show All Connections*** menu item or the corresponding icon from the toolbar.

See also: Online Diagnostics

Clicking the table header sorts the table by this column in ascending order. Another click on the table header changes the sort order to descending.

The individual columns have the following meanings:

Column	Description
Type	Displays the transport protocol used: **- TCP/IP** **- H1** **- Iso TCP (RFC 1006)** **- Collect Type** If the connection is inactive, its name is followed by **(off)** and grayed out.
Connection Name	The name of the connection.
Job	Shows the application protocol used.
NetProt	Shows the network protocol used and whether the connection has been established actively or passively: **TCP/IP: Client** [active] / **Server** [passive] **H1: Act** [active] / **Pass** [passive]
Dest. Address	Shows the address of the destination hardware (**IP address, IP name** or **Ethernet address**) which can be 0 if the connection has been established passively.
Parameters	This column displays some connection parameters: - **Port number** - Transport protocol **TCP** or **UDP** - Local **TSAP**, Dest TSAP for RFC 1006 and H1 connections
Parameter Details	Shows additional connection parameters and details, including: - **OpcPipe** access: Read and write: **RW**, read only: **RO**

7.4 Opc Diagnostics

The **OPC diagnostics** shows the topics and their items.

The showed contends is depending on the elements and the optimizer settings.
The items are showed from the sight of the plc connection. If multiple items are combined to a longer area the start and length of the area is showed. Single items will be showed with its item name.

Each element area shows how many items it contains.

On top some global statistics are shown.
Clear them over the general system settings.

Hint: If the controller connection update rate is much faster as the OPC group update rate most of the time no items are active. The list of registered items will show no items in this time.

The **connection overview** shows for each connection the status and some statistics as frame counters and number of disconnects.

7.5 Logic tables

Logic tables are reading input data, processes them and are writing the results into output data.

Logic tables are used for calculations, for collecting data, processing data, for accessing databases, for sending e-mails and more. They are using variables and calculations for processing the results.
Logic tables are working independently from OPC and other accesses. They can use this services for getting data or offering the results.

For creation of new logic table one of the wizards should be used. This saves time. Later the wizard generated logic table can be changed to normal hand made logic table for changing ore details of it.
Use the Wizards if ever possible.

This chapter describes:

- Variables for input and output values
- Trigger : Time, Value changes, Events, UA internal variables, raw data, Bits and Reset
- Database: requests, handling the data
- Calculations, Conversions and Rounding
- Structures collect the data und disassemble them
- Constants in Logic tables
- Subroutines and their Parameters
- Calling external programs
- Remote Procedure Calls, call results.
- Event send, event priorities
- Decisions und Data gates
- Comments, Debug prints
- Connections of the logic elements can be done with the mouse or touch screen.
- Online Diagnostics
- Examples for the logic tables
- Send Email

Hint:
The logic tables will exist in the pure OPC Server for data optimizations only.

Handling
The logic tables will be created and edited graphically.
Elements can be placed everywhere.

The logic elements can be moved. Select them with mouse or touch, move them with mouse button down or the finger down. The new position will be marked. Release the mouse or tahe your finger lets it save the new position. Connections will be dragged automatically. If an element will be dropped over

another the existing element will be moved away.

Delete elements with dragging them over the trashcan. They can be marked and deleted over the context menu or the menu line, too. If an element is marked it can be deleted with the "del" key.
Elements can be arranged new with "Beautify Display". Additionally a predefined option "horizontal" and "vertical" exist. Optimizing will be recommended after heavily editing.
The layout will be saved. If it is opened later it will be shown equal.

A hint: Moving elements does not affect the logic.

Create or edit the elements as following

- With the menu
- With the context menu
- With double click on an element
- With click on the "edit" symbol
- With drag and drop

An output can drive any number of inputs. An input can have one connection only. Lot of logic elements can have multiple inputs.

A logic table can contain multiple paths. For secure handling that all paths are working correctly they need to be combined logically on end. This can be done with connecting all to the final trigger reset element.

Logic tables can be diagnosed online. All states and variable values are showed.
A hint: In subroutines the online diagnostics does not work.

7.6 Symbol List

The server offers the possibility to use a symbolic name for a specific direct address in the PLC.
The **Symbollist** dialog shows the existing symbols and allows you to create, edit, and delete individual symbols, as well as the symbol import.

Left window

In the tree view can be navigated through the entire symbol file. The connections are displayed. Any hierarchies can be created.
A context menu (right mouse button) allows you to insert, delete, edit.

Right window

The symbols of the selected entry of the tree view are displayed.
A double click in a list entry will release it for editing.
A context menu (right mouse button) allows you to insert, delete, edit, as well as the symbol import.

Import into selected topic

Opens the S7 Symbol Import or Modbus Symbol Import dialog and allows you to transfer the symbols from an S7 project or Wago Modbus Export File that has been created with the programming software.

New symbol

Opens the Add Symbol dialog and allows you to add a symbol.

Edit symbol

Opens the Edit Symbols dialog and allows you to edit the selected entry.

Delete symbol
Deletes the selected entry from the list, without query.

Note:
Editing the symbols for Rockwell controls is NOT possible !!!

Global symbols
It is possible to create symbols that are present in all connections of the same type.

7.6.1 Edit Symbol

Symbol Name
The symbol name will be offered for browsing.

Itemsyntax
This field allows using of symbols from the other station. For special purposes the Tani internal item syntax can be used directly.
See Item Syntax

Symbol comment
The comment is optional. Normally the comments are imported directly from the controller or device. It is optional. Browsing shows the comments. It is available in Item Properties, also.

7.7 Status variables

The status variable lists are for monitoring content from controllers or devices.

Each variable list will be used over its name.

If write is allowed data can be written into the controllers also.

In case of writing into arrays all variables should be given. Not given values are written to zero.
Arrays are supported up to 100 elements each. If they are bigger they can not be written.
For writing of arrays separate the elements by a **space**.

Attention: Writing affects the plant !

Press the "Save" button to save the current list of variables.

With the PLC Engine Collect it is possible to store multiple lists of "Status Variables".

7.8 Logger

A toolbar in the top part of the window provides the following functions:

Close
Closes the Logger dialog box. This does not affect the logging process. Logging continues even if the window is closed.

Set Marker

Adds a marker text including the current timestamp to the list.
"----------"

Clear
Deletes the entries stored in the RAM. If you have selected to write the log data to a file, the RAM content will be saved to the file before deletion.

Settings
Opens the following dialog box: Logger Settings

Auto Scroll
Cyclically refreshes the window contents and automatically scrolls to the last line containing the most recent entry.

Suspend
Clicking this button suspends logging. This allows the user to diagnose the current content without overwriting older entries.

NOTE:
No more entries will be recorded.

Master / Slave
Master / Slave shows the current mode of configuration parameterization. In slave mode, it is not possible to make any settings in the Logger. The mode is selected automatically with the first configuration instance always being the master; any other instances will run in Slave mode.

As a general rule, it is not recommended to access an device by more than one configuration instance.

Operation
In this window, you can also select entries with the mouse or by pressing the space bar, and copy them to the clipboard. To access this function, master access is required.
Right-clicking then opens a **context menu** which provides the following functions:

Copy
The selected entries are copied to the clipboard and can be pasted into other programs.

Clear Logger
Deletes the entries stored in the RAM. If you have selected to write the log data to a file, the RAM content will be saved to the file before deletion.

Clear Selection
All selected entries are deselected.

Refresh
It is possible to refresh the display, using the key <F5> or by Menu File - Refresh

To specify whether to create the log in the RAM only or to also save it to a file, click the Logger Settings button in the Logger window.
For more information on the logging function in the device, see the Logger section.

7.8.1 Logger Settings

In the Logger Settings dialog box, you can choose to keep the log data only temporarily in the RAM or save it to a log file. For this, you can specify the logging intervals as well as the directory and the file to

which the log data will be saved.

Cache Size
Specify how much RAM you want to allocate to data logging. The cache acts as a ring buffer. If you increase the cache size, more entries will be buffered in the RAM.

Use File Logging
Select this check box if you also want to save the data to a file.

File Settings

Maximum Disk Space
Specifies how much disk space may be occupied by log data. When the value you set here is reached, the old files are deleted. This setting serves to prevent the log files from taking up all the hard disk space.

NOTE:
When the selected setting is reached, the Logger will automatically delete files from the directory without a confirmation prompt.

Directory
Specifies the directory in which you want to save the log files.

File Prefix
The file name is made up of the prefix followed by a sequential number and the date and time.

Save Every ...
Specifies the interval at which the file will be saved if the cache in the RAM has not yet completed one loop. When the ring buffer is full and the write interval time has not been reached, the ring buffer loops around more quickly and the old data in the ring buffer is being overwritten.

NOTE:
When you save to a flash drive, the number of writes is limited. The fewer writes, the longer the lifetime of the flash drive will be.

New File Every ...
Specifies when a new file will be started. This facilitates the evaluation.

File Name Example
The display field shows an example of a file. Here you can see where the log files are located and what the file names look like.

7.8.2 Logger Configuration

Switching on the corresponding option should cause the corresponding logger entries to be logged. These are available in the logger window.

Errors

Errors	
Errors	Any kind of errors should be recorded in the logger. Exceptions are PLC errors.
PLC element error	If a range is not present in the PLC or a range that is requested is too small, a corresponding entry is generated in the logger. This also applies to all other errors generated by the PLC.

SQL error	Any kind of errors with SQL connections should be recorded in the logger.
SQL error log file	The file name for the SQL error logging

Status

The reparameterization of the connections and the logon for reparameterization are recorded in the logger.

Status	
Connection configuration	If parameter connections are changed and saved, an entry is logged.
Connection status change	If the status of connections changes, an entry is logged.

PLC data

The parameters of the corresponding PLC requests from the server to the PLC are recorded in the logger.

PLC data	
Send	Content of the telegrams in the write direction. The area being written is logged.
Send data	The data content of the telegrams in the write direction is logged.
Receive	Content of the telegrams in the reading direction. The area being read is logged.
Receive data	The data content of the telegrams in the reading direction is logged.

SQL data

The queries and data of the Engine target connection are recorded in the logger. This can be an SQL connection, but also a different type. The direction can be selected.

SQL data	
Send	Content of the telegrams in the write direction. The area being written is logged.
Send data	The data content of the telegrams in the write direction is logged.
Receive	Content of the telegrams in the reading direction. The area being read is logged.
Receive data	The data content of the telegrams in the reading direction is logged.

Specials

Specials	
Access path create	generates an entry when a connection is established or disconnected.
Item create	generates an entry when a item is created or deleted by a client.
Item activate	generates an entry when a previously created is activated, or inactivated by a client.
Sequence chain step change	generates an entry when a sequence chain goes into an other step.
Logic table output changing	generates an entry when an output element of a logic table changes its value
Log events	generates some more verbose entries.

8 Dialogs

This chapter covers the following topics:

Further functions are described in the following sections:

8.1 Server Diagnostics

The window shows a detailed diagnosis of the logged data points.
It can be arranged and searched according to different aspects:

- Request order
  - From topic
  - From client

- Item display
  - Flat
  - Hierarchic

8.2 General System Settings

The dialog allows you to set general valid settings.

Reset Diagnostics

The diagnostic counters are then set to 0 again in order to be easier to observe.

Reload configuration

An updated symbol information can also be updated. It is necessary when the configuration or symbol files are edited with an text editor.

OPC DA server (only for Windows)

OPC DA server functions

Here the OPC DA functionality can be switched off and on. If switched off, the calls from the OPC client

will hang or get errors because OPC DA is switched off.

OPC DA Topics are case sensitive.
Some OPC clients do not support this, so this can be disabled.
Note: In some languages, case-sensitive rules are not well defined.

System Topic

OPC System Topic
The predefined topic "System" exists in the list of the configured topics. If the topic is not needed, this can be disabled.
Attention: When switched off, no connection monitoring and statistics are possible via OPC.

Update system elements every second
This function is rarely called. However, if necessary, this can be updated every second.

Browse Arrays

All array elements are offered individually when browsing. For large and many fields this can be very slow.
If this is disabled, the array is offered as a whole when browsing.
However, the item syntax rules can still be used to address each item individually.

Symbol Import Settings

Add Element names in import
The datablocks, inputs, outputs and flags are grouped together, or grouping is omitted.

Import S7 instance data blocks
Normally, the instance DBs are not needed in OPC. So turn it on if needed.

OPC UA Settings

Accept untrusted certificates
For machine setup, it is helpful to accept untrusted certificates, which should then be switched off for normal operation.

8.3 Configuration Client Settings

The following question dialogs can be turned off or on:

Question on program end
Normally, a question dialog appears when the configuration of a station is connected and terminated

Question on delete
If a single connection, or symbol, or list ist deleted, this dialog appears, or not

Question on delete all
If all connections, or all symbol, or all lists are deleted, this dialog appears, or not
..

Browse fetches names only
On big tag configurations, browsing could consume a lot of time. To reduce it, enable this option. Browse tree displays only names, no data type, no access rights.

8.4 Services

The dialog shows the state of the services required for the Engine function.
If the appropriate rights are sufficient, the services can be stopped and started.

Note:
Stopping services can cause unpredictable effects.

8.5 Reboot

With the ***Reboot*** menu item you can reboot the device via software (warm start). When doing so, you may be prompted for a reboot password.

The reboot password is determined in the following way:

- Open the ***Help*** menu and select ***Versions***.
- In the Versions dialog box, the value **Version Param Server** is used as the password in the following format: xyyzzz.
- Example: 2.06, Build 16 ==> the password is 206016

During rebooting, all connections are closed and then reestablished.
If users are defined You will need the system management permission.

8.6 Configuration Langage Setting

Here, you can select the language in which the configuration is displayed.
If something does not appear in the selected language, a restart of the configuration is required, or not all text entries are translated.

8.7 Configuration Adapter Settings

The dialog is used to specify which adapter the configuration is to perform the automatic station search.

Timeout for connections

With the connection timeout, the default timeout can be changed for this connection during the station search of 3s. This is e.g. For connections on the Internet, in which the life telegrams have been deactivated. The reaction time in the case of connection disturbances is thus increased (for example, in cable breakage).

Use Timeout

This button should only be activated if the default connection timeout is too short or a very slow transfer connection is present. In normal circumstances, however, it is switched off.

Show all addresses of a station

In the Station Search window, all the station addresses are displayed, even if the configuration for this address can not be established.

Search over all adapters

A station search is performed over all installed network adapters

Active adapter
The selected adapter is displayed

Available network adapters and addresses
The list shows all available adapters. The desired one is selected here.

8.8 Memory Variable List

Memory Variables are holding data that is located in the OPC Server or PLC Engine Collect.
Memory variables can be read and written very quickly.
The amount of memory for the memory data is limited by the used hardware only. Most systems allow lot of megabytes of data or more.

Hint:
Memory data are set to zero during power off.
Some devices are supporting NV RAM. Ask your system manufacturer if this is available.

8.9 Structure List

Structures are fixed lists of variables. A structure can contain other structures.
To access structures via OPC UA or OpcPipe it is necessary to have this structures imported from the controllers.
Following operations are possible:

- Edit
- Import
- Export
- Reimport
- Delete

Since UPC UA transfers all structures before the browsing information, keep only the needed structures in this dialog.

For big plants with many controllers structures can be imported globally. This only works if the same structure is in all controllers exactly the same. There is no online check for this - this would generate too much load on the controllers.

8.10 Item Redirects

The OPC Server or PLC Engine Collect knows the Optimization for long fields - often error texts.
This can also be used without interfering with the SCADA system. To achieve this, logged elements must be assigned to the optimization.

The Optimize long data fields wizard automatically inserts the elements when the checkbox is set to "Name assignment required "".

The name redirection list contains the list of
SCADA element name -> optimization element name

8.11 OPC UA System Settings

The dialog allows the setting of the OPC-UA behavior, as well as the setting of the UA Discovery service.

Accept untrusted certificates
Use this for the plant startup. If this is not set you need copy and register all certificates before OPC UA will accept the connections.

Use external discovery server
Allows to enable or disable the the use of the UA Discovery service.

Use multicast server discovery
Allows to send multicast requests.

Client lifeack send rate
Keep alive time in miliseconds.

Discovery Server
Entry filed for the url of the discovery server, simple or complex, and the Message Security Mode.

Login
The Login information for secure usage.

Edit expert settings
Allows to edit the parameters of the Security Policy.

Security Policy
Encryption

8.12 Remove all Configuration Parameters

All settings can be deleted and will be irretrievably gone if you did not backup before.
It is possible to select which settings should be deleted.

8.13 Import configuration from INAT and Softing OPC or DDE Server

The dialog allows you to accept the connection parameters and symbols of INAT and Softing OPC or DDE servers.
All connections and symbols are transferred and loaded into the currently connected Runtime.

Only import symbols
If the connection already exists and only the symbols should be imported without overwriting the connection parameters, please select this setting.

9 Logic tables

Logic tables are reading input data, processes them and are writing the results into output data.

Logic tables are used for calculations, for collecting data, processing data, for accessing databases, for sending e-mails and more. They are using variables and calculations for processing the results.
Logic tables are working independently from OPC and other accesses. They can use this services for getting data or offering the results.

For creation of new logic table one of the wizards should be used. This saves time. Later the wizard generated logic table can be changed to normal hand made logic table for changing ore details of it.
Use the Wizards if ever possible.

This chapter describes:

- Variables for input and output values
- Trigger : Time, Value changes, Events, UA internal variables, raw data, Bits and Reset
- Database: requests, handling the data
- Calculations, Conversions and Rounding
- Structures collect the data und disassemble them
- Constants in Logic tables
- Subroutines and their Parameters
- Calling external programs
- Remote Procedure Calls, call results.
- Event send, event priorities
- Decisions und Data gates
- Comments, Debug prints
- Connections of the logic elements can be done with the mouse or touch screen.
- Online Diagnostics
- Examples for the logic tables
- Send Email

Hint:
The logic tables will exist in the pure OPC Server for data optimizations only.

Handling
The logic tables will be created and edited graphically.
Elements can be placed everywhere.

The logic elements can be moved. Select them with mouse or touch, move them with mouse button down or the finger down. The new position will be marked. Release the mouse or tahe your finger lets it save the new position. Connections will be dragged automatically. If an element will be dropped over another the existing element will be moved away.

Delete elements with dragging them over the trashcan. They can be marked and deleted over the context menu or the menu line, too. If an element is marked it can be deleted with the "del" key.
Elements can be arranged new with "Beautify Display". Additionally a predefined option "horizontal" and "vertical" exist. Optimizing will be recommended after heavily editing.
The layout will be saved. If it is opened later it will be shown equal.

A hint: Moving elements does not affect the logic.

Create or edit the elements as following
- With the menu
- With the context menu
- With double click on an element
- With click on the "edit" symbol
- With drag and drop

An output can drive any number of inputs. An input can have one connection only. Lot of logic elements can have multiple inputs.

A logic table can contain multiple paths. For secure handling that all paths are working correctly they need to be combined logically on end. This can be done with connecting all to the final trigger reset element.

Logic tables can be diagnosed online. All states and variable values are showed.
A hint: In subroutines the online diagnostics does not work.

9.1 Wizards

The wizards are creating or edit of predefined logic functions.

The function is depending from the selected wizard.

This wizards are existing:
- **Data optimizing for OPC**
- **Data copy in multiple variants**
- **Write data into a database**
- **Read data from a database**
- **Update data in a database**
- **Delete data from a database**
- **Raw data text, CSV, binary**
- **Subroutine : Limit test**

The wizards are for an easily start. But they also are generating the hull for a logic table.
The wizards naturally do not offer all the details which can be done by a manual work.

You can edit wizard generated logic tables by hand. Please switch on the detail view in wizard tables.
This lies in the View menu "Show details of wizard tables".
On the first attempt editing the logic table a warning occurs, after manual editing the wizard view becomes impossible for this logic table. Click on "Yes".

Hint:
The product OPC Server only supports the "Data Optimizing". PLC Engine Collect supports all.

9.1.1 Optimize Long Data

The frames from the controller will be optimized. So the controller network load becomes less - the data are read not so often.
The index element is polled. If it changes the data field will be read, too. Additional a item redirect is created. So the OPC item redirects to the optimized item.

Especially with error texts from the controller, getting the data from the controller is often slow. The

reason for this is that the long error text is cyclically read from the controller to see if it has changed. Error texts usually have an additional error code in the control. It is sufficient to read this error code cyclically and only after a change the complete error text.

The function "Optimize long data" cyclically reads the index. If it changes, the data part is read.
This also works without intervention in the existing OPC client variables. Internal memory variables are used. To do this, variable names are assigned to the results of the optimization.

9.1.2 PLC data copy 1

Controller data will be read from the source and written to the destination if they are changing.
Source and destination can be in different controllers. The data will be converted if necessary.

No change in any of the plc programs need to be made.

9.1.3 PLC data copy 2

This function is the INAT Echochange functionality.
It will be used frequently for RAW data.

Data incoming on any side will be sent to the other side. If protocol conversions are necessary they will be done.

One use case is converting from OSI-H1 to TCP/IP.
RFC1006 can be translated to normal port connections.

9.1.4 Raw binary data

Raw data sometimes are sent by plc programs in binary form.
The raw data wizard binary uses such data.

9.1.5 PLC data into database

Data will be read from controllers and written into a database.

In SQL databases this uses the INSERT statement.

Ideal the database is available and will be browsed directly. If doing so the Add button will be grayed. If this is not possible the column names and their data types need to be entered by hand.
During saving the configuration database column elements which are not configured to a process variable will be ignored. Exceptions are the elements which do not allow null values.

9.1.6 Plc and opc data into a csv file

Data will be read from industrial controllers or over OPC. The data will be written into a CSV file.

If the CSV file does not exist the headline will be written. The column names are taken from the variable names.

The new line will be added to the CSV file if the trigger becomes true.
For later more easily CSV handling a timestamp should be written in the file also.

9.1.7 Database data into PLC

Data will be read from a database.

With an index an entry is read from the database.
The data will be stored in the PLC or other variables.

In SQL databases this uses the SELECT statement.

Ideal the database is available and will be browsed directly. If doing so the Add button will be grayed. If this is not possible the column names and their data types need to be entered by hand.

9.1.8 Raw text data

Barcode scanner or wages often offering text based data.
The raw data wizard text uses such data.

9.1.9 Raw CSV data

CSV stands vor "Colon Separated Data". Spreadsheets are using this.
In Europe the colon often is the semicolon (;). The comma is used for the separation of numbers in pre and after colon parts. In the US or England the dot (.) is used here, so the name CSV is correctly.

The raw data wizard CSV uses such data.

9.1.10 Limit tests

The wizard "limit test" creates a subroutine which checks a value against an upper and lower level. The value will be inside this limits.
If the input value lies outside the limit the specified output will be set.

9.2 Table name

Each logic table has a unique name.

A logic table can be part of a s sequence chain. It can be a subroutine also. It can be part of a group. All this will be defined in the table header.

9.3 Database request

Databases are working passive. They need to be requested.
Requests are

- SELECT — Reads data from the database. This brings the data.
- INSERT — Writes data into the database.
- UPDATE — Actualizes existing data.
- DELETE — Deletes a data set.
- CALL/EXEC Procedure — Calls a subroutine without parameters

- CALL/SELECT Function Calls a subroutine with parameters
- User defined database request

Input data are connected directly to the database element.
The results a database request brings (mostly SELECT or CALL) will be used with the database results element.
The result data are valid on "Ready" and "Ok" and "Result Count not zero".
More complex data handling are easier to handle if they are realized with sequence chains. The result data are corresponding the database request for the same database connection and the same request id.

Parameters can be used in the SQL statement with a leading ":". If the SQL statement requires a ":" it must be doubled.

User defined database requests require detailed knowledge for SQL and the specific database used. Even if the request does not use input data one input element need to be defined, it is used for the request.

Detailed information about the status of the database request will be returned in the output named "Error Codes". This field contains three variables:
Byte 0: Server error. The database server is reachable and it works well, but the request can not be handled properly. The database server brings an error text in this case. The text will be shown in the logic table online diagnostics. It will be shown also in the diagnostics logger it this is not switched off.
Byte 1: Connection error. The database server is unreachable.
Byte 2: Cache error. The cache will be used only if this is configured in the database connection. The cache can return an error if the path or file is unreachable, or of the file is read only. It can come also if the medium runs out of disk space.

Hint for ODBC with Oracle:
Please enter the username in capital letters for the schema name. Oracle does not support the schema browsing.

Enable file caching
If the connection to the database is interrupted, the calls are cached in a file.
The following conditions must be met for this.
- The buffering must be activated in the connection.
- The buffering must be activated for each database command block.

This function can only be activated if the respective command should NOT return any data. This is always the case with INSERT, UPDATE, DELETE, never with SELECT. With CALL and with user-defined commands only if no results are expected ("Maximum expected results" = 0).
If this has been fulfilled, then stored procedure calls are cached also.

The background to this regulation is that the response data can no longer be assigned when requests are caches. Subsequent caching of commands that provide answers is therefore not intended.

9.4 Database results

The data of a database request will be used here.

To insert a database result, proceed as follows:
- Complete the SELECT command with the desired parameters

- Then right mouse click on the SELECT command - at the bottom of the menu - copy result block
- go to an empty field and paste

This inserts the corresponding database event.
Now it can be wired.

9.5 Data gate

The data gate allows conditional handling of data. If the triggering input is "true" it switches the data to the output. A data gate can be used for optimizing in data comparisons.

9.6 Handle structure

A structure is a fixed layout for multiple variables.

Create structure: A structure will be filled from its variable parts.
Not connected values will be zero or the empty string.
The output is the complete structure.

Disassemble structure: A structure will be disassembled to its parts.

You can use the clipboard for structure handling: In the input or out element handling a structure use "copy structure block". The next clipboard insert will insert the block using the correct structure.

9.7 Calculations

Calculations are handling two or more variables or constants.

The following calculations will exist:

Arithmetic as addition, subtraction, multiplication, division.

Logic as and, or, xor.

Comparisons as "equal" "not equal" "bigger" "less" and more.

Floating value test

Rounding as "to next" "against zero" or "away from zero". The number of decimal signs can be given.

Other as
- Modulo The rest of a division
- Negation The sign is changing
- Absolute The sign is removed

Calculations can have errors So a division with zero is impossible. An addition of two numbers can exceed the value limits. In all of these conditions the **OK Bit** is not set.
The result in calculation errors

- Division by 0 -> the maximum possible value
- Overflow -> The rest which does not fit the result

Elements which can have multiple inputs can be expanded with the "add" element. Connections can be dropped on the "add" element also creating more inputs.

9.8 Constants

Constants will have fixed data values for numbers, texts and arrays.

All supported data types can be used.

9.9 Variables

Variables are data coming from the process or are handled internally. They can contain input or output data. They can be used for calculations also.

Variables can accesses with static or dynamic configurated connections. The connection name and the variable description can be configurated in the variable itself or they are used as parameters.

Local variables lie in the "memory" topic.
Local variables will loose their value during power off.

9.10 Structures

Structures are collections of data.

The structure element allows it modifying parts of it. It can be used as a field also.

A structure can contain other structures.
A structure can not use itself internally. This is true in recursive using, too.

9.11 Structur into elements

The structure separate allows disassembling of a structure to its parts.
All structure elements are added as outputs.

9.12 Comments

The comment element is for commenting only. It does not consume computation time.

9.13 Debug prints

The Debug Print logic element prints test output into the system diagnostics logger.

In the diagnostics logger the checkmark for Logic Table Output Changing must be set for this element becoming working.

9.14 Roundings

Rounding simplify the reading of "crooked" numbers. For float values the xx can be rounded. For integers, only signs before the colon can be rounded.

Rule in rounding of a number:
- 0 -- 5 -> 0
- 6 -- 9 -> 1

So:

- 3,141 rounded to two digit will be 3,14
- 2,718 rounded to two digit will be 2,72

The number of digits behind the colon will be given. Negative values are affecting digits before the colon. All digits behind the colon becomes to zero.

Rounding is possible to next, up, down, zero or not zero.

9.15 Number conversions

Numbers will be converted to the destination format.
If the conversion will result in a changed value the **OK Bit** is not set. Example is a division with zero. Even if the OK Bit is set a try is done for a result. The value will be so near the expected result as possible. For integers the result overflows and the rest of the operation will be returned.

Examples:
Division by zero will result 65535 in 16 bit.
200 plus 56 for an unsigned 8 bit returns zero.

For time values there exist some predefined conversions. If none of these will fit a format string need to be entered.
The conversions will exist in text output only.
Rules for the format text:
The % sign is the format specifier. If a % should be in the final text it must be given twice.
Letters which can follow behind the format specifier:
- **d** day
- **H** hour from 0 to 23
- **I** hour from 0 to 12
- **N** nanoseconds. Additionally a length can be given (%3N -> milliseconds)
- **p** or **P** brings the AM or PM
- **s** UNIX time stamp (seconds since 1.1.1970)
- **S** seconds

- **y** year in two letters
- **Y** year in four letters
- **%** the percent sign

9.16 Connections between elements

Connections are connecting the elements of a logic table. Often outputs will be connected to inputs.

Not connected inputs will be zero. Trigger inputs are true.

The wires will be created with the mouse: Much elements have small rectangles on its sides. Examples are "Data" and "OK Bit". With the mouse such an element can be selected (left mouse button down). Drag the mouse with the pressed left button A line from the starting point to the mouse is shown. If the mouse is oven a possible connection destination point it will be marked. Leaving the mouse button does the final connection.
An output can have any number of inputs connected. An input can have only one connection. Some logic elements allowing add more inputs.

A logic table can have multiple inputs and outputs.

If a connection will generate an error it is displayed in the error color. The logic table will not work. Connections with warnings are shown in the warning color. The logic table will work, but they will not work as expected in all conditions. Changing signed to unsigned and back will generate a warning.

9.17 Connection of an element

Lot of logic elements have input and output connectors. During the creation of the element the minimal necessary connectors are created. More input connectors can be added. Example is the "add" element. It adds all input values.

9.18 Subroutines

Subroutines are logic which can be used multiple times.

A subroutine can have inputs and outputs.

Each logic table can be a subroutine. This is defined in the table header .

Beware that subroutines can overwrite any variable in memory or the connected devices.

With the Wizards skeletons of often used logic tables can be created.

9.19 Parameters of subroutines

A subroutine can have multiple inputs and outputs.

Open inputs will become zero, in case of strings they are containing the empty string.

9.20 External program call

This logic element allows calling external programs

Please notice:
- The program call runs on the system level. The called Programm need supporting this
- Parameters can be given. The rules of the parameter usage is handled by the called program. Multiple parameters are realized with a string array.
- The return value of the program is returned on the output parameter Exitcode
- The call works synchronously

Mostly programs using a graphical frontend can not be called.

Some ways exist for returning complex data from the called program:
- Files.The called program writes data into a file. After the program end the file is read in the logic table.The logic tables can not access the whole file system for security reasons. So a file "MyFile.txt" as parameter for a logic table will be c:\Users\Public\Documents\Tani\Userfiles\MyFile.txt for a Windows program called in the logic table. In Linux the same file will be /etc/Tani/Userfiles/MyFile.txt.
- Network connections. Mostly this are send/receive connections in PLC Engine Collect. The called program writes the return data into the connection. The logic table reads the data from the connection.

OPC UA. The called program needs to be an OPC UA client. It writes the return data into memory variables of PLC Engine Collect.
In all of this cases it is recommended to give the parameters for the returned data system as a parameter to the called program.

9.21 Remote Procedure Call (rpc)

A remote procedure call is a special logic table with the special rpc bit set.
The call will be triggered from a remote communication partner.

Rpc can contain many parameters. They always hace a trigger condition which becomes true if the partner calls it.

A rpc has a result.The result can offer one or multiple parameters. The trigger always exists.

A rpc call should not run for a long time. If the result will need time a second rpc for polling the result is needed. Often this call is the result of an event which triggers this.

In OPC UA the rpc declarations are coming from XML files. This files mostly come from companion specifications as Euromap, TMC and others.

9.22 Remote Results

E remote procedure call always has a result. The minimum is a trigger only.

More information are under Remote Procedure Calls.

9.23 Event Send

This element sends an OPC UA event.
The node which the event is connected will be seen as the event source in the event receiver.

9.24 Read array elements

Especially for raw devices as barcode scanners or wages parts of a fixed array can be extracted.

Example:
The frame contains multiple numbers
"123141256"
The first number has three letters, the next two letters, the last has four letters.
With reading from zero with the length of three the first number will be extracted.

The opposite function is Write array elements.

9.25 Write array elements

Especially for raw devices as barcode scanners or scales parts of a fixed array can be created.

If no constant values for start or length is given the values from the parameters are used.

The opposite function is Read array elements.

9.26 Search in arrays

In the given field will be searched the desired content.

Often this is used for raw data e.g. from a scale. Then the system searches for a given start value like a semicolon.

9.27 Write into text fields

The text will be written to the desired space.

If no fixed start or length is configured the values from the parameters are used.

9.28 Search text

Searches text in the data.

9.29 Trigger new file

A directory will be checked for a new file. If this becomes true a trigger signal is returned and the name of the new file.

Hint:
New files must not be created faster than this element is called.

9.30 Check file for existence

A file will be checked if it will exist.
General file hints here.

There are two ways configure the file name
- Fixed configured in the element. Path and file will be selected.
- Variable. The file name is given as a parameter to the element.

9.31 Trigger file changing

A file will be checked for changes in its content.
General file hints here.

There are two ways configure the file name
- Fixed configured in the element. Path and file will be selected.
- Variable. The file name is given as a parameter to the element.

On software start normally all files are assumed as not changed. If this is not the case please set the check mark "Trigger event if file exists on startup".

9.32 Delete file

Deletes content of a file or the whole file.
General file hints here.

If the first line will be deleted after usage and another application or logic table writes at the end of the file a cache is realized. The database cache is using this. It writes to the file if the database is not available, and it reads back and deletes if it comes again.

There are two ways configure the file name
- Fixed configured in the element. Path and file will be selected.
- Variable. The file name is given as a parameter to the element.

9.33 Read file

Reads the first or last line from the file, or the complete file will be read.
General file hints here.

There are two ways configure the file name
- Fixed configured in the element. Path and file will be selected.
- Variable. The file name is given as a parameter to the element.

Hint:
There will be read data up to 64K Bytes.

9.34 Write file

Write the complete file or appends to the file.
General file hints here.

There are two ways configure the file name
- Fixed configured in the element. Path and file will be selected.
- Variable. The file name is given as a parameter to the element.

9.35 Switch RAW data read on/off

In raw connections the data may be come faster than it can be handled. For this the receiving can be switched of and on again.
Data during the off state are stored internally. They will be given after the receive is switched on again.

The switch on will declare the last received data as handled. The next receive will return the next data.

Incoming data during the off phase will be stored internally. They will be returned in the next receive call after the receiving is switched on again.

Often this will be used in sequence chains.

Example:
A scanner returns data.
Additionally something will be fetched from a controller.
After this the data will be written into a database.

All this can run for a longer time as the next data will come from the scanner.
So no data will be lost if they come in a faster rate the database can handle them.

9.36 Test connection status

Returns the status on a connection. If it is connected correctly the OK bit is set.

Not all connections can be checked successfully.

UDP or datagram connections always will return Ok.

Connections to a Ethernet serial converter will return Ok if the Ethernet connection is established. It does not return the status of the serial line.

9.37 Trigger

Triggers are used in the logic tables.
These must not be mixed with normal signals.

A trigger signal is "now". This is neither 1 nor 0. Rather, it can be described with an extremely quick change from 0 to 1 and back again.
If something like that is required, then only a logic table helps that sets a bool constant to 1 with a trigger and as quickly as possible to 0 again. That works with constant bool 1, then a trigger high value. It strikes immediately. Then use the trigger result to control a boolean constant with 0. You can then use the 1 or the 0 value. However, this is not the normal way and does not always lead to the goal.

Triggers are not a number and never will be. In older versions of PLC Engine Collect there was a bug that a trigger signal could be used as a bool and therefore did not work or only worked by chance. Because bool is a state that lasts. A trigger, ie "now" is an arbitrarily short signal.
All trigger inputs that are connected to a trigger output save this "now". All other inputs of course also save the pending value, but the processing is different. A trigger as a bool (bug) randomly returns sometimes 1 and sometimes 0. The chance depends solely on the signal runtime. This is the classic mistake, where you spend hours looking for the cause.

See also:
Time trigger
Value change trigger
Bit Trigger
Trigger reset

9.38 Time trigger

The time trigger fires after the configured time.
The trigger will be cleared again with the Reset Trigger element.

The resolution is a millisecond.

Notice: The timer starts again if the reset is called. The runtime of the table must be added..
Times under 10ms are not guaranteed. If only few logic tables are defined and they do not use complex calculations the speed can reach up to 2ms.

9.39 Value change trigger

The value change trigger signals a changing value.
For floating point values the value should be rounded before the comparision preventing noise von analog

values not signaling the trigger.

Options are
- Value change
- Rising edge
- Falling edge

The trigger will be cleared again with the Reset Trigger element.

9.40 Bit Trigger

Bit trigger sets the result if the input value is true.

Options are
- Trigger on high level
- Trigger on low level
-

The trigger will be cleared again with the Reset Trigger element.

9.41 UA intern variables trigger

In the usage of OPC UA internal variables sometimes a trigger is needed if someone writes the same value multiple. The UA variables trigger is for that.

9.42 Event Trigger

Events will occur on special conditions of the connected machine.
Mostly this will be used in conjunction with OPC UA.

In the OPC server one or multiple event conditions will be programmed. If the programmed condition becomes true the event is sent.

The client connects to this event.

The parameters of the event will be taken from the XML files which define the event.

Events are working hierarchical. A browsed location in the tree will deliver all events from that node and other sub nodes.
All events will be given if the global message event is connected.

9.43 Trigger reset

The trigger must be reset again for switch it active. This is be done with the trigger reset. The element is created automatically during the creation of a trigger element.

The trigger reset can have multiple input elements. The inputs are working AND combined. So the trigger will be reset it all inputs are TRUE.

9.44 If then else

Queries are switching the input to the output with a selector.

For "true" at the selector the "true" data will be switched to the output. Otherwize the "false" data will be switched.

9.45 Switch Sequence Chain

A sequence chain is a procedure of multiple logic tables. At any time on step will be active only.
All steps of a sequence chain need to be in the same group.
The step number will be defined in the logic table header. Additional a step need to be defined as a member of a sequence chain.
Elements in the same group which are not part of the sequence chain are handled as normal logic tables.

The switching to another step sets the active sequence chain member to the given step. A step number can set directly, or it can be taken from a variable.
In case of switching to invalid step numbers - the destination step does not exist - the actual step number will be preserved. The OK bit will not be set.

9.46 CSV Edit

CSV Format
This creates a CSV block with the separator from the input strings.

CSV Split
This separates a part of a CSV line into the variable, it returns the rest text for further handling

Both functions need the separator letter and the quotation sign.
Quotations are needed if the separator letter is used inside variables.

9.47 Status variables

The status variable lists are for monitoring content from controllers or devices.

Each variable list will be used over its name.

If write is allowed data can be written into the controllers also.

In case of writing into arrays all variables should be given. Not given values are written to zero.
Arrays are supported up to 100 elements each. If they are bigger they can not be written.
For writing of arrays separate the elements by a **space**.

Attention: Writing affects the plant !

Press the "Save" button to save the current list of variables.

With the PLC Engine Collect it is possible to store multiple lists of "Status Variables".

9.48 RemoteFileDialog

If files will be handled in lines the line end is the letter 10 (0ah \n).

A file will be selected as in other programs also.

Notice that the file access rights may be limited for network files.
Windows
PLC Engine Collect works as service. A service does not have a login for network access on standard. Please configure the system user network access rights with the tools of the operating system. Ask your system operator.
Linux
In network mounts on standard the "World" user is used.

9.49 Send Email

An email will be send over the selected connection.
One or multiple receivers, the subject and the mail content need to be given.

All the paths can be defined fixed in the dialog. They can be used as parameters also. Mostly the mail content will come generated from real plant data.

To configure the email connection see: Email Settings

9.50 Examples for the Logic Tables

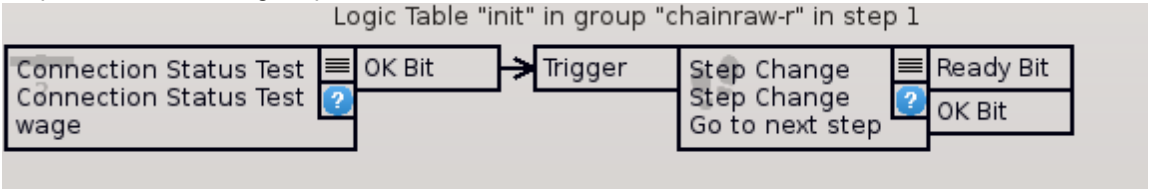
Example 1: Send a request, read the answer, handle the data.
Example Logic Table: RAW Text Receive

9.50.1 Example Logic Table: RAW Text Receive

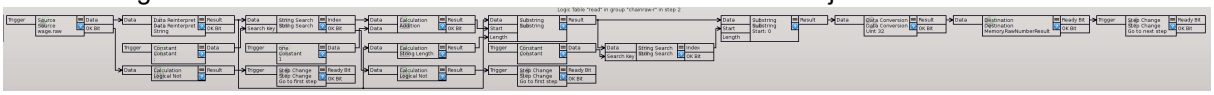
Raw handling: Receive text data, search separator and get number behind.

A sequence chain will bring the easiest result.

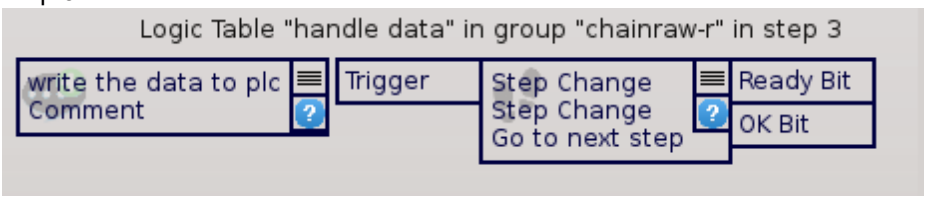
Step 1 is the initializing step. It checks the connection.



Step 2 reads the data. A separator is searched. The text behind the ":" will be converted into a number until a blank. Technical the text on letter behind the index will be converted into the number.
For decoding a line of a .csv file the wizard "Raw csv decode" does this job.



Step 3 handles the data.



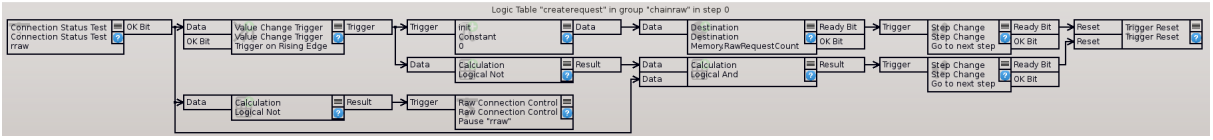
Step 4 switches back to step 1.

9.50.2 Example Logic Table: RAW Send and Receive

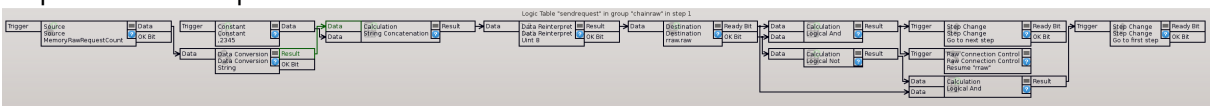
Raw handling: Send a request, read the answer, handle the result

A sequence chain will bring the easiest result.

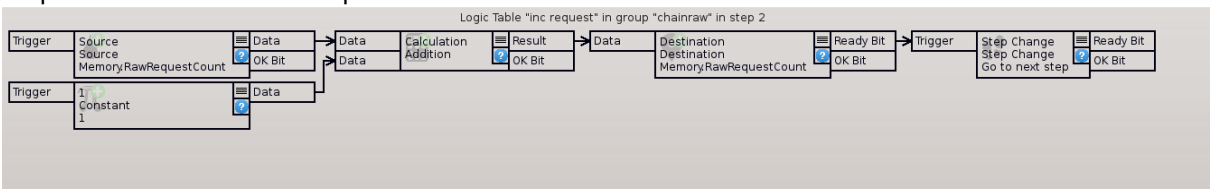
Step 1 is the initializing step. It checks the connection. The ramp from "not established" to "established" will be used for initialize functions. It switches the receiving on.



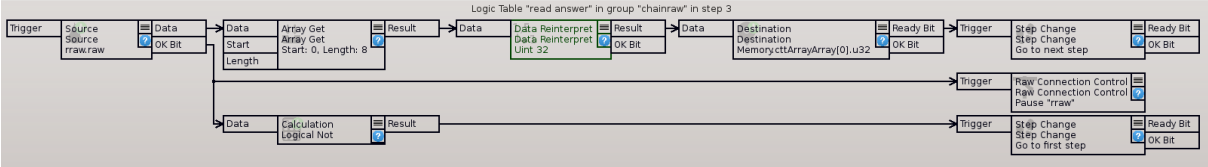
Step 2 sends the request.



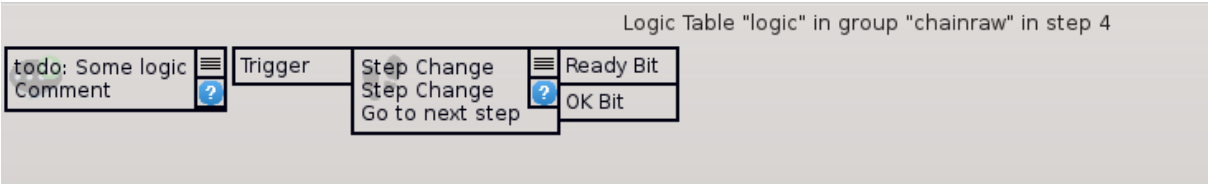
Step 3 calculates the next request.



Step 4 reads the answer. It it was received the receiving is suspended.



Step 5 handles the received data.



Step 6 switches back to step 1.

10 Connections

A connection defined the details for a network connection. Each connection has an unique name In OPC it will be used as topic - the first part of any item. Please use letters only for the connection name Your OPC client will support.

More dialogs are handled under Windows.

10.1 New Connection

For each hardware that is to be accessed (ie with which to communicate), at least one connection must be created in the device. The connection defines the access path to the hardware, e.g. the channel between the two communication subscribers.
The dialog offers the following input possibilities:

Name of the connection

The connection name is freely selectable. Only letters and numbers should be used. The connection name used here can be selected in the further dialogs.

Connection type

The type of connection is selected here. The selection is multi-level. If an image or a text button is selected, a wider selection opens under it. If something is selected there, another selection can open under it.

Active data request	With this type of connection, active requests are made to the other communication subscribers, which then respond with the corresponding data, so-called client connections.
PLC access	Connection to the different PLC types and raw data. These types of connections allow for the use of redundancy.
Raw data connection	A connection to a device that transmits a stream of user data. Often scales, barcode readers or similar. Pure user data (raw data) is transmitted without protocol information.
OpcPipe Client	A connection to a device or PC via the OpcPipe protocol, which communicates via a simple TCP / IP port connection.
OPC UA Client	A connection to a device or PC with an OPC UA server. A current protocol, standardized and specified by the OPC Foundation
MQTT Client	Connection to an MQTT broker (or MQTT server) used in IoT (Internet of Things).

Data Server	This type of connection provides variables with values that can be fetched via active connections, so-called server connections.

OPC UA Server	OPC UA connection that can read data from all connections where OPC access is allowed.
OpcPipe Server	OpcPipe connection that can read data from all connections where OPC access is allowed.
MQTT Broker (Server)	MQTT connection that provides variables that are created and written with an MQTT client.

Database Access	This type of connection connects to known databases. The PLC Engine Collect is used to read or write the records from the database via tables.
MySql	Connection to a MySql database.
MsSql	Connection to a MsSql database.
Postgre SQL	Connection to a Postgre SQL database.
ODBC	Connection to a database not listed above, e.g. Oracle, IBM, or any else. The device must then have the appropriate ODBC driver installed and set up.

Enable Redundancy

defines this connection for using network redundancy.
Available are **Static Master** or **Dynamic Master**.
The static master tries using the first connection it ever possible.
The dynamic master remains in the new connection after failure of the previous connection.
Redundancy can be double or triple.

See also Redundancy.

10.1.1 PLC protocol

Name of the connection

All connections are defined over their names. Most letters are allowed but [ ] and the dot. The names will be used as part of the item name in OPC access. Possibly some OPC clients do not accept all names. Details will show the manual of the client and the user operating system running it.

Network Transport protocol

The transport protocol is selected here, via which Ethernet network the communication is working.

Transport protocol	
TCP/IP	This is mostly used. The Transmission Control Protocol / Internet Protocol is routable and is now used for communication with PLCs and field devices. TCP/IP should be selected for RFC 1006 (OSI on TCP) communication.
OSI-H1	OSI OSI-H1-Protocol for connections to Siemens S5 oder S7. Older types may require this protocol.
Serial	On some devices, mostly Linux, the Serial communication is available.

PLC controller protocol

The PLC application protocol is selected here, which is supported by the PLC.

PLC protocol	PLC type

Siemens S7	S7 protocol to Siemens S7-400, S7-300 or S7-200 controllers, or to S7-1200, S7-1500.
Siemens S5	Communication to Siemens S5 CPs or INAT CPs
Modbus TCP	Communication to PLCs or devices that use the Modbus TCP protocol, such as Wago, Beckhoff, Group Schneider, Modicon ...
Rockwell Logix Family, Slc, PLC5	Communication to Allen Bradley ControlLogiX, CompactLogiX, SoftLogiX, Rockwell over Ethernet / IP or to Slc or PLC-5. General Electric PAC systems are using this protocol also.
BACnet	Communication to devices with BACnet protocol
Mitsubishi Melsec-Q, FX5	Communication to Mitsubishi Melsec-Q, QL and FX5

The application protocol of the connection can not be changed later. The connection should be deleted and newly created.

10.2 Connection settings

A connection defined the details for a network connection. Each connection has an unique name. In OPC it will be used as topic - the first part of any item. Please use letters only for the connection name Your OPC client will support.

The following chapters are described here:

- Network parameter TCP/IP
  - Special TCP/IP Settings
  - IP-Address
  - Port
  - TSAP
  - TSAP for S7 Connection
  - Routing TSAPs
  - PLC Header
  - Network stations available
- Network Parameter OSI-H1
- Serial Parameters
- S7 protocol settings
- S5 protocol settings
- Modbus protocol settings
- Mitsubishi protocol settings
- Rockwell protocol settings
- BACnet Protocol
- Raw protocol settings
- Database General
- OPC UA Server Settings
- OPC DA Client settings
- OpcPipe Parameters
- Optimizer settings
- Redundancy Settings

10.2.1 Network parameter TCP/IP

In this dialog box, you can edit the TCP/IP parameters.

Name of the Connection

You can choose any connection name you like. All characters except brackets may be used.

Destination IP Address (or name)

Here, you can specify the IP address of the destination station (IP address of the PLC) or the DNS name of the destination station. If the device was configured for DNS and a DNS server is available in the network, you can also enter the symbolic name of the destination station.
For information on the structure of IP addresses, please refer to the IP Address section.

Port

Port numbers are addresses that are used on the transport layer in order to address applications. Port numbers are required for TCP connections. Ports are similar to the TSAPs used for RFC 1006 and H1 connections. Each port number is a 16-bit number in the range from 1 to 65535. To establish a connection, please note the following:

NOTE:
You can only establish a connection if the port number is identical on both sides.

More detailed information on ports

Type:

Here, you can specify whether your own station will actively initiate the connection attempt or will wait passively for the destination station to establish the connection. Please make sure that different values are selected on both sides of the connection.

Client (Active)	The station will actively initiate the connection attempt.
Server (Passive)	The station will wait for the destination station to establish the connection.

Protocol

The TCP and UDP protocols are available for selection. TCP is a secured protocol.

PLC Header
Enable this option for the communication with devices.
For details on the structure of the PLC header, refer to the PLC Header section.

Life Data Acks
If you enable this option, payload frames without content (only headers) are transmitted to keep a connection alive that is not used cyclically (heartbeat monitoring). Both communication partners must support this feature. For the OpcPipe communication, it is recommended to enable this option.

RFC1006
If you enable the RFC1006 option, H1 frames will be “wrapped” in a TCP/IP frame for transport. To configure the TSAPs, click the **RFC1006 TSAPs** button.
Own TSAP, Dest TSAP
TSAPs (Transport Service Access Points) are addresses that are used on the transport layer in order to address applications. TSAPs are required for RFC 1006 connections. To connect, enter the local TSAP into the Own TSAP field and the TSAP of the communication partner into the Dest TSAP field. In this

context, please note the following:

NOTE:
In order to be able to establish the connection, the value in the Own TSAP field of one system must match the Dest TSAP value in the other system, i.e. they must match crosswise.

More detailed information on TSAPs
TSAPs for S7 Connections

10.2.1.1 Spcial TCP/IP Settings

Use PLC Header
The PLC header can be used for communication with other devices where the PLC header can also be enabled.
See also PLC Header.

Life Data Acks
If you enable this option, payload frames without content are transmitted to keep a connection alive that is not used cyclically (heartbeat monitoring). Both communication partners must support this feature. For the communication with S7 PLCs, we recommend to disable this option.

Life Acks as TCP/IP Standard
By enabling this option, you can activate connection monitoring, a function that is poorly supported in many socket libraries. Life acks are frames that pass the connection status. It is recommended to enable this setting (default setting). If a WAN connection is used, you might want to disable this function for cost reasons.

Ignore TCP End Check
This setting only makes sense for Receive Direct connections.
Option enabled:
Reading in a TCP/IP frame disables its end tag. Thus, the reception buffer passes exactly the amount of data the PLC user program had requested. When data is read cyclically, this may cause problems if different data lengths were configured for the “Send Direct” and “Receive Direct” (user program) jobs.
Option disabled:
(default setting, recommended)
Excess data bytes will be ignored. This setting is only relevant to the currently edited connection (Receive Direct connection).

Life Acks with Previous Received Data
Same function as for “Life Acks as TCP/IP Standard”
If you enable this option, the data byte received last will be sent back, but with a wrong sequence number so that the other communication partner replies by sending a life ack with the correct sequence number.

Send an Ack Immediately after Received Data
It is recommended to enable this parameter if data is transmitted cyclically in a LAN. In most cases, the TCP/IP protocol collects multiple data blocks before sending an acknowledgment. If you enable this setting, the acknowledgment (ACK) is sent immediately. This increases the data rate when data blocks are sent cyclically in short intervals.

Send a Life Data Ack after the last Frame in a Sequence
This option is only available if PLC Header or RFC 1006 is used for communication.
Many socket implementations (socket libraries) do not send the IP acknowledgment frame. If you enable this function, a life data ack will be returned as an acknowledgment.

Do not Wait for Send Acknowledge
This option is only available if PLC Header or RFC 1006 is used for communication.
Option enabled:
The sending station initiates frames without waiting for the frame acknowledgment (ACK) relating to the previous frame.
Option disabled:
(default setting)
The sending station waits for the frame acknowledgment (ACK) relating to the previous frame before sending a new frame.

End Connection with FIN Instead of RST
Option enabled:
As a reply to the end connection frame (FIN), a corresponding end connection frame is sent.
Option disabled:
When a station receives an end connection frame, it resets the connection.

Use the same Port Number for both Ports
This setting sets both the source and the destination ports to the port selected in the parameterization. This disables the automatic setting of the source port to a value > 1024 in some devices (see also TCP/IP System Settings). This setting is required for the communication with CPs that do not correctly handle frame traffic via UDP.

Big Endian Format in the PLC Header
In the PLC header, the sequence number will be transmitted in the MOTOROLA format (big-endian). Usually (flag not set) the INTEL format (little-endian) is used to transmit the sequence number in the PLC header.

Change Connection Timeout
Using this setting, you can specifically change the timeout setting for connection monitoring that will end a connection when parameterizing a station remotely. This makes sense in cases where you disabled the transmission of life data acks for an Internet connection, for example. Please note that increasing this value will also increase the response time in case of connection problems (such as a broken cable).

Change Connection Memory
This setting allows you to change the memory size reserved for a connection. The minimum memory size for a connection is 1460 bytes (maximum number of payload for Ethernet connections). For a broadcast Receive connection (UDP), this setting might not be sufficient. If the partner station sends data faster than the PLC can accept it, the data will be buffered in this memory. The UDP data will only be discarded if this memory overflows.

10.2.1.2 IP-Address

Basics

To establish the communication between two technical devices, each device must be capable of sending data to the other device. This data can only be received by the intended remote station if it has been addressed properly. In IP networks, this is ensured by specifying an IP address.
An OPC Server is able to address a PLC directly by its IP address. Example: 192.168.1.20
It is also possible to address a PLC by its name. For this purpose, specify the name of the PLC and the corresponding IP address and enter the domain server in the TCP/IP settings of the server. For a domain name (e.g. “PLC1”), the server queries the name server to get the IP address and then addresses the PLC directly using its IP address (192.168.1.20).

IP Address

Each IP data packet starts with an IP header. This is an information section used for transmission on the IP layer. This header also includes two fields where the IP addresses of both the sender and the receiver are entered before transmitting the data packet. Routing is done on layer 3 of the OSI model, the network layer.

IPV4
The IPv4 addresses that have been used predominantly since the introduction of Internet Protocol Version 4 consist of 32 bits, i.e. 4 octets (bytes). This means that it is possible to represent 232 = 4,294,967,296 addresses. In dotted decimal notation, the four octets are written as four integers, from 0 to 255, in decimal format, separated by dots.

Example: 192.168.1.20.

Actually all IPV4 addresses are occupied. New systems will need an IPV& address.

IPV6 – new version with a bigger address space
Due to the rapidly increasing IP address demand, it was foreseeable that the usable address space provided by IPv4 would soon be exhausted. The IANA address pool was depleted on February 03, 2011. This was the main reason for developing the IPv6 protocol. IPv6 uses 128 bits to store addresses. This 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 each square millimeter of the surface of the earth with at least 665,570,793,348,866,944 (= 6.65 · 1017) IP addresses.

Since a decimal representation with
ddd.ddd.ddd.ddd.ddd.ddd.ddd.ddd.ddd.ddd.ddd.ddd.ddd.ddd.ddd.ddd would neither be very transparent nor handy, IPv6 addresses are usually represented in hexadecimal format. To further simply this representation, every two address octets are grouped and separated by colons.
XXXX:XXXX:XXXX:XXXX:XXXX:XXXX:XXXX:XXXX

Example: 2001:0db8:85a3:0000:0000:8a2e:0370:7344

To further shorten the string, leading zeros in a block can be omitted.
A sequence of blocks consisting only of zeros can be replaced with :: but only once in each IPV6 address.

Example: 2001:db8:85a3::8a2e:370:7344

For IPV6, usually names are specified instead of the IP addresses.

Domain Name

An IP station can be accessed over a domain name. Ask your system administrator. Possibly he has done this.

10.2.1.3 Port

Port numbers are addresses that are used on the transport layer in order to address applications. Port numbers are required for TCP / UDP connections.
The port number is a 16 bit number from 0 to 65535.
Certain applications use fixed port numbers that have been assigned by the IANA and that are generally known. Usually, they are between 0 and 1023. They are referred to as well-known ports. The registered ports are between port 1024 and port 49151. Application providers may register ports for their proprietary protocols.
The remaining ports (49152 through 65535) are referred to as dynamic and/or *private ports.* They can be

used variably because they are not registered and do not belong to any application.

10.2.1.4 TSAP

TSAP = **T**ransport **S**ervice **A**ccess **P**oint

On the transport layer, OSI (H1) uses so-called TSAPs to address applications. These connection endpoints are required both for OSI (H1) and RFC 1006 connections.

Parameters for ISO (H1) connections		**Parameters for RFC 1006 connections**	
Station A	***Station B***	***Station A***	***Station B***
Own TSAP A Dest TSAP B MAC Address A	Own TSAP B Dest TSAP A MAC Address B	Own TSAP A Dest TSAP B IP Address A	Own TSAP B Dest TSAP A IP Address B

In order to establish the connection, the value in the Own TSAP field of communication partner A must match the value in the Dest TSAP field of communication partner B, i.e. they must match crosswise. Accordingly, the Dest TSAP field of station A must match the Own TSAP field of station B.

Own TSAP (station A) = Dest TSAP (station B)
Dest TSAP (station A) = Own TSAP (station B)

This condition can easily be met if you set the same value for the Own TSAP and Dest TSAP fields. These values may be identical.
If multiple connections are to be established between 2 stations, the TSAPs for the individual connections be must different.
To exactly specify a connection, the combination of TSAP + MAC address (or IP address) must be unique.

Rules for entering TSAPs for S7 connections

- TSAPs have a minimum length of 2 bytes and a maximum length of 8 bytes
- TSAPs can be entered as hex or ASCII characters.
- The TSAPs for S7 connections have a special meaning. See TSAP for S7 Connection and Routing TSAPs

Rules for entering TSAPs for S5 connections

- TSAPs have a minimum length of 2 bytes and a maximum length of 8 bytes for TCP/IP, and 16 bytes for H1.
- TSAPs can be entered as hex or ASCII characters.
- The TSAPs for S5 connections do not have a special meaning. They are selected arbitrarily.

10.2.1.5 TSAP for S7 Connection

The S7 has the family types
S7 200 / 300 / 400 / 1200
and the
S7-1500
If the "optimized" data blocks need to be used set the check mark on the "1500" in the protocol settings.

S7 200 / 300 / 400 / 1200

For non-configurable connections (= default connections), so-called standard TSAPs are used. For these, the following rules apply:

First group
contains device IDs for which resources are provided in the S7:
01: PG or PC
02: OS (operating or monitoring device)
03: Others, such as OPC server, Simatic S7 PLC...

Second group
contains the addresses of these components
Left character (bits 7....4):
Rack number multiplied by 2
Right character (bits 3...0):
CPU slot (< 16). S7-300 always uses slot 2
The standard TSAPs **MUST** be used on the PLC side (Dest TSAP field of the device).
The local TSAP of the device (Own TSAP field) may be selected freely, but should have the same format.
We recommend to use 01 01 in the Own TSAP field.

Examples:
03 02 Communication with the S7 CPU in rack 0, slot 2
03 43 Communication with the S7 CPU in rack 2, slot 3
03 2E Communication with the S7 CPU in rack 1, slot 14

S7-1200
The S7-1200 is usually addressed with the TSAP 02 01 (binary).

S7-300
The S7-300 is usually addressed with the TSAP 03 02 (binary).

S7 1500
The S7 1500 does not use the service and rack addressing.
The own TSAP should be binary 06 00, the destination TSAP is ASCII "SIMATIC-ROOT-ES"

Note:
To enter the binary standard TSAPs, use the HEX field (left field).

Note:
The S7 300 or 400 protocol can be used for the S7 1500 too. But then the optimized data blocks can not be accessed.

see also: Routing TSAPs

10.2.1.6 Routing TSAPs

S7 Routing General

Routing is the ability of the OPC Server to address PLC controllers across subnet boundaries as well. OPC communication is possible with all stations that can be reached via gateways. It does not matter how many gateways it has and how many are between the Ethernet network (where the OPC Server station is located) and the destination network. For the FETCH / WRITE communication, a connection must only be configured on the OPC side. In S7 it is not necessary to create a connection. For the connection path, information of the destination network and the destination station must be known.

However, the TSAPs would have to be assembled manually. The following settings are necessary for the routing:

Own TSAP:
01 00 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 01 00

Dest TSAP:
01 06 01 02 [aa bb] 00 00 [cc dd] [ee] 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ff] [gg]
[aa bb] first part of the subnet ID from Step7
[cc dd] second part of subnet ID from Step7
[ee] MPI Profibus address
[ff] Service
[gg] Slot of the CPU

Destination PLC results in the Dest TSAP

Here, settings are made for the destination PLC, the controller from which values are to be read out or written to the values.

S7 subnet ID
Each subnet in the S7 receives a subnet ID, which is automatically generated by Step 7. The subnet ID of the destination network must be specified here. The S7 subnet ID can be e.g. be determined as follows:

- in the program Step 7 open the window "NetPro".
- Click with the right mouse button on the network and select object properties.

MPI / Profibus address
If it is an MPI network, the MPI address of the PLC is entered here. If it is a Profibus network, the Profibus address is entered.

Service
The service "3" is intended for OPC connections

Slot of the CPU
The slot number of the CPU is entered here.

Special case INAT echolink

Own TSAP:
Is the first device between network and MPI an echolink, then
FF FF FF FF FF FF 56 43 4F 4D 00 02 04 [aa] 00 00 00 00 00 00 00 00 00 00 [bb] 00
[aa] 0x80 plus backplane bus MPI CP Port, or MPI address of the CPU
[bb] = Com Connection of the echo

The first device between network and MPI is an echolink
If an echolink is in use that establishes the connection between Industrial Ethernet and the subnet, information about the routing PLC is required.

Routing PLC
The routing PLC is the S7 controller, which is connected to the echolink via MPI and represents the first transition from Industrial Ethernet to MPI. The MPI address of the routing module is entered here (MPI address of the CPU or CP).

COM port
Here, the COM port of the echolink device is selected, via which echolink is connected to the MPI interface of the S7 PLC.

Note:
Some CPs have their own MPI address. This MPI address is automatically determined and assigned by the CPU. If such a CP is in use, it is essential to ensure that the MPI address of the CP and NOT

the MPI address of the CPU is used.

Determine the MPI address of the routing PLC
The MPI address of the routing PLC is determined in "NetPro" by marking the station and selecting "Object Properties" with the right mouse button. If a CP-MPI address is displayed next to a CPU MPI address, the MPI address of the CPs should be used.

Alternatively, the CP can also be marked in the hardware configuration and the "Object properties" selected using the right mouse button. The MPI address to be used is then displayed under "Backbone Connection".

Destination TSAP
The Dest TSAP is the same as described above.

10.2.1.7 PLC Header

The data-stream oriented TCP/IP protocol is capable of grouping multiple short data units into a larger unit. This increases the data throughput of the network. However, it requires a header for the protocol above TCP. This corresponds to the procedure common in other protocols (FTP, HTTP).
For this purpose, the 8-byte PLC header was developed.

Only use the PLC header if the communication partner supports it. Otherwise, the connection will fail!

PLC Header Format

Byte no.	Meaning
Byte 0	0x4d ´M´
Byte 1	0x4b ´K´
Byte 2***	Number of payload bytes following the header (LSB*).
Byte 3***	Number of payload bytes following the header (MSB**).
Byte 4***	Bit 0 = 1, if other frames follow
Byte 5***	0
Byte 6***	SeqNo. LSB*
Byte 7***	SeqNo. MSB**
Datalen in bytes	Payload

*LSB: Least (Lower) significant byte
**MSB: Most significant byte
*** Bytes 2 / 3, bytes 4 / 5 and bytes 6 / 7 together form the “short” data value. They are represented in the INTEL format

Acknowledges
If DataLen equals 0, the frame does not contain payload, but a life data ack. Data acknowledgments allow connection monitoring, a feature that TCP/IP, as a wide area protocol, does not include inherently. The default times for connection monitoring are the same as for the H1 protocol. This makes the S5 TCP/IP system H1-compatible from the PLC or PC perspective.

Sequence Numbers
Bytes 6 and 7 contain a sequence number that is 0 when establishing the connection and will be

incremented by 1 each time payload is sent. This frame counter additionally secures the data transmission. If live data acks are sent, the sequence numbers are not incremented and Datalen is 0.

Fetch and Write Connections
For Fetch and Write jobs, the first 16 data bytes at the start of a job correspond to the SINEC AP header. The SINEC AP header is also used for H1 communication.

Sending / Receiving Data
When sending data over the S5 TCP/IP, a frame can include a maximum payload of 512 bytes. This maximum value is preset by the default tile block size. Received data packets can contain up to 1460 bytes. These limits are monitored automatically by the TCP/IP protocol so that no monitoring is required on the user side.

Transmission without Frame Header
The header at the beginning of the frame may be disabled. In this case, the application programs on both sides are responsible for monitoring. In this context, please note the following:
1. In particular with Send Direct and Receive Direct jobs, certain time limits until frame receipt must not be exceeded. If these time limit were ignored, the internal buffers would be full, e.g. due to requests. Thus, it would be impossible to synchronize request and response.
2. A certain blocked data transmission mechanism must be used so that it is possible to recognize the end of payload.
3. On the recipient side, you need to make sure that the frames from the reception buffer have been read before the partner station sends the next frame.
It is mandatory to set up connection monitoring in the application program.

10.2.1.8 Network stations available

All available stations in the network are displayed. An automatic station search is initiated.

10.2.2 Network Parameter OSI/H1

In this dialog box, you can edit the OSI-H1 parameters.

Name of the Connection
You can choose any connection name you like. All characters except brackets may be used.

Adapter
Here, you can specify the name of the Ethernet adapter used to establish the H1 connection.

Destination MAC
Each Ethernet station is addressed using a unique MAC address (also referred to as Ethernet address or hardware address). Enter the Ethernet address of the destination station here. The destination address consists of 6 bytes (to be entered in hexadecimal notation).

NOTE:
In a single network, each station must have a unique Ethernet address.

Own TSAP, Dest TSAP
TSAPs (Transport Service Access Points) are addresses that are used on the transport layer in order to address applications. To connect, enter the local TSAP into the Own TSAP field and the TSAP of the communication partner into the Dest TSAP field. In this context, please note the following:

NOTE:
In order to be able to establish the connection, the value in the Own TSAP field of one system must

match the Dest TSAP value in the other system, i.e. they must match crosswise.

More detailed information on TSAPs
TSAPs for S7 Connections

CR Parameters
With OSI (H1) connections, a transport connection is established. For this purpose, an active transport instance sends a CR TPDU (Connection Request Transport Protocol Data Unit) signaling that it wants to connect. This CR TPDU is used to send some parameters to the partner, such as the desired TPDU size, the TPDU format, and others. Since there is no standard CR parameter definition, please refer to the operating instructions of the destination system to find out which parameters you need to specify here, if any. If no information is available, do not enter any CR parameters.

Type (active / passive)
Here, you can specify whether your own station will actively initiate the connection attempt or wait passively for the partner station to establish the connection. Please make sure that different values are selected on both sides of the connection. Usually, the PC will be parameterized as active. This is the default.

Protocol
Using this option, you can specify whether frames on this connection will go to all devices (Broadcast), whether a certain group of stations should be addressed (Multicast), whether a secured connection will be used (Normal) or whether the data will be transmitted via an unsecured connection (Datagram). Usually, you select “Normal” here.

Priority
The line priority can range from 0 (highest priority) to 4 (lowest priority). 0 and 1 are so-called express priorities, 2 and 3 are normal priorities. Priority 4 is only used on rare occasions because it causes the connection to be reestablished for each send transaction. On the other hand, if it is only used intermittently, it puts less load on the network than the other priorities because the line will not be monitored (the connection is closed after each send transaction). Please note that when using express priorities, the transmission will not be faster than with normal priorities. For some PLCs, however, the data will be transferred to the RAM using an interrupt if you select priority 0. This may result in a faster overall data transmission. For priorities 0 and 1, the maximum data length is 16 bytes. Usually, Prio 3 is used here.

10.2.3 Serial Parameters

Name of the Connection
You can choose any connection name you like. All characters except brackets may be used.

Connection Active
Shows whether the connection is active or not. If not, you cannot register items and there will be no connection to the PLC. This feature allows you to disable a connection temporarily without deleting it so that you can re-enable it later without having to enter all parameters again.

Write allowed
For some connections, you can disable the Write function.

Line
Shows the selected serial interface. It is possible to choose the line with the drop box.

Baud Rate

The baud rate (also referred to as modulation rate) indicates the number of state changes in the transmitted signal per second. You can set baud rates from 75 to 115200. The same baud rate must be used for the sending and the receiving stations.

Data Bits

Indicate the number of bits per character here: 5, 6, 7 or 8

Parity

The same parity must be used for the sending and the receiving stations.

Even

For even parity, the number of 1s (including the parity bit) is even.

Odd

For odd parity, the parity bit is set in a way to obtain an odd number of 1s (including the parity bit).

None

Select this option if you do not want to use the parity check.

Mark

Select this option if you want the parity bit to be permanently 1, i.e. if your device expects a 1 parity bit.

Space

Select this option if you want the parity bit to be permanently 0, i.e. if your device expects a 0 parity bit.

Stop Bits

Duration of the stop bit transmission, relative to the transmission time of an information bit. For asynchronous, serial lines, the transmission of a character is terminated by sending a stop bit. Specify here whether 1 or 2 bits will be used for termination.

Handshake Protocol

Here, you can select the protocol mode to be used. The devices use it to communicate whether they are ready to send / receive data. There are two types of handshake: software handshake and hardware handshake.
Hardware handshake means that synchronization is done via electrical wires. V.24 typically uses either RTS / CTS or DTR / DSR for signaling.
Software handshake means that synchronization is done by transmitting control characters. XON / XOFF are the most common software handshake characters.

RTS / CTS

RTS= Request To Send, signal of the V.24 interface
CTS= Clear To Send, signal of the V.24 interface

DTR / DSR

DTR= Data Terminal Ready (to send), signal of the V.24 interface
DSR= Data Set Ready (for operation), signal of the V.24 interface

XON / XOFF

The communication partners add ASCII characters to the communication data stream. These control characters in the data stream indicate whether the data flow is possible or not:
XON: data can be received
XOFF: data cannot be received

RS485

Select RS485 if the serial device is connected to the echo device via an RS485 cable.

DTR RTS = 1

Select DTR RTS = 1 if DTR or RTS default to 1

None

Select None if you do not want to use any of the protocols listed above for the serial line.

Timeout

This option specifies the time after which the device will no longer wait for a response and assume the partner is not present.

10.2.4 S7 protocol settings

S7 CPU
In Siemens S7 controllers and compatible the connection parameter details will be defined automatically.

If you have not set up the control online, the control type is not recognized.

If you are using a controller of the S7-1500 family, please select "s7 1500".
The own TSAP is hex: 06 00
The remote TSAP is hex: 53 49 4D 41 54 49 43 2D 52 4F 4F 54 2D 45 53
As text: SIMATIC-ROOT-ES

In rare cases, with very old controllers and compatibles, the S7-200 protocol must be selected manually.

For the item access of S7-200, 300 and 400 please use the S7 item syntax.

Show symbol comments
To reduce communication via OPC, symbol comments can be disabled.

Use array start index
Some fields in the PLC do not start with 0, as is common in C programming and OPC UA. In this case, the corresponding offset of the PLC program is used with this setting.
For compatibility with earlier versions, this option is disabled, but should be enabled.

Allow Item Syntax
With S7-1500 and S7-1200, item syntax is possible for non-optimized blocks, even though the modern and secure TIA protocol is used.
The item syntax would then be, for example:
PLC1,ItemSyntax.DB1.DBW0
where PLC1 is the connection name, "ItemSyntax" is a fixed component, and the familiar item syntax of the S7-300/400 can be used after it. All separated by a period.
If the fixed component "ItemSyntax" is causing problems, it can be redirected and eliminated via a virtual redirect connection.
If the controller is configured as a S7 1500 the variable access will work with symbol names directly. They can be found over online browsing. An item syntax does not exist in this case.

Use folders in symbol Exploring
Normally the elements are grouped in folders for the inputs, outputs, flags and the data blocks. If you do not like that uncheck the "Use folders in symbol browse".

Show block numbers in comment
The data blocks have numbers. This is true for the optimized data blocks also. Normally the data block numbers are not needed. If you need them please check "Show block numbers". Then the comments of the data blocks begin with the data block number.

Hint:
In the S7 1500 data requested over networks are not synchronized with the PLC cycle. More information in the Siemens PLC manuals.

10.2.5 S5 protocol settings

Allow Bit writing
The Siemens S5 does not support bit write over network. So for writing a bit the element needs to be read from the controller. Then the bit will be changed. On end the element will be written back into the controller.
In the PLC program modifies a bit in the same time this information may be lost.

So the factory setting is off for this.

You can turn this on at your own risk.

Best is designing the plant so that no element is written by the controller and over network.

Use ANSI charset as default
In the past there was the ANSI charset used inside the strings of the plcs. Nowadays UTF8 is used. to be compatible with older PLCs please chech zhis option.

10.2.6 Modbus protocol settings

In this dialog box, you can edit the specific parameters for the connection to Modbus PLCs.

	Description	Default
Name of the Connection	This field displays the connection name which can be changed here.	
Slave address	The slave address entered here will be transmitted in the Modbus frame.	9
Start Address 0	The starting address in the frame will be transmitted as specified.	Yes
Start Address 1	Some Modbus devices expect a starting address in the frame that is 1 less than the one you entered. The first address is 1.	No
Use Byte Swap	Swaps the bytes within a word	Off
Use Word Swap	Swaps the words within a double word (32 bits)	Off
Output Write	For writes, the Modbus opcode “Write Single Coil 05” is used or “Write Multiple Coils 15” or both	both
Register Write	For writes, the Modbus opcode “Write Single Register 06” is used or “Write Multiple Registers 16” or both	both
Register Bitwrite	Nearly all Modbus devices are supporting the register bitwrite using opcode 22. For devices without this an insecure version can be used: Read a register, modify a bit and write the register back.	Opcode 22

The slave address will be indicated in the item syntax.
Example: Id2.R2
For more details, see Modbus Item Syntax

10.2.7 Mitsubishi Protocol settings

In this dialog box, you can edit the specific parameters for a connection to Mitsubishi Melsec-Q QL FX5 PLCs.

You need to set these parameters if a Mitsubishi Net has been set up and the hardware supports it.

Routing	Description	Hex	Decimal
Network Number	Network ID in the hardware configuration of the programming software	0	0
PC Number	Only relevant if a permission management system has been set up, allowing or denying PC access.	0xFF	255
Destination Module IO Number	Network address	0x3FF	1023
Monitor Timer	Monitors the time within which the response should be received.	0x100	256
Destination Module Station Number	Module address of another module in a PLC.	0	0

Monitoring Timer
0 for infinite, or a timeout in units of 250 ms.

The effect of these values depends on the hardware used.

10.2.8 Rockwell protcol settings

The Allen-Bradley Rockwell controls of the Control LogiX family, Compact LogiX or Soft LogiX and require the CPU number as a parameter. This is determined automatically if the IP address of the PLC is entered correctly and the PLC is switched on and connected.

For a few special cases the symbols are not sufficient, see special item syntax in Item Syntax for symbolic PLCs.

Routing over underlying controllers is fully supported.

10.2.9 Rockwell Routing

Rockwell offers multiple internal networks. Because of the long time Rockwell sells its devices these are used for accessing the PLC.
If a device need to be accessed over underlying networks a access path is needed. Rockwell offers RsWho as a graphical online configuration.

Tani uses its own online path search. This solves the fact that no version of RsWho can access all devices.

A access path always starts with a PLC over Ethernet.
Underlying networks can be any, ofter DH+ is used in the plants.
A path can have multiple entries. Each selection is stored in a path section entry.

DH+ networks are browsed automatically. In IP networks the ip addresses need to be entered by hand.

10.2.10 BACnet Protocol

BACnet is a protocol suite for the building automation.

The connection is using IP and UDP.

During the startup of the connection the device identifier is requested online using an UDP broadcast. If the network security settings are not allowing this the device id can be entered by hand.

Remote networks can be simply accessed over a BBMD device. BBMD stands for **B**acnet **B**roadcast **M**ulti **D**evice.
BBMD mostly will be used during the connection configuration. It handles the broadcast for finding the BACnet devices in the remote network and returns the connection parameters accessing them.

10.2.11 Raw protocol settings

The dialog allows the control and interpretation of the receive and send data.

Telegram Boundary

Fixed length
A constant length of the receive data is expected entered in the **Constant Length** field.

8, 16, 32.64 Bit length
The length is in the telegram and is, according to the setting, an n-bit number, it starts from the **Length Offset**, and the data starts from the **Header Length** byte offset

Transport Layer Boundary
If a transport protocol is used that includes a length, such as RFC1006, IsoOnTcp, H1, the length of the telegram is taken into account.

Use start sign

The characters before the start sign are ignored. The data begins behind this character.

10.2.12 OPC UA Server settings

An OPC UA server connection can be used simultaneously by as many partners as specified under **Number of instances**.

Additional connection parameters

Some OPC-UA servers need additional connection parameters. If not leave the field empty. The documentation of the device provides the required information.

Security and Login

OPC-UA allows lot of options for security. All are basing on Certificates. Additional a password may be needed.

Memory limit

Each OPC frame can have a size of up to 4 gigabytes. Usually, frames of that size are not required. Since OPC must provide the memory for such frames, you can restrict this value in order to prevent reduced performance. This value should be selected in a way that it will not cause a frame of maximum length to be truncated.
If at runtime, this limit is exceeded, the data will be discarded and the communication partner will receive a corresponding message.

Other settings

Allow writ with compatible data types
Using OPC UA, writing is only possible with the correct data type. In real plants this behavior is not ideal. This option allows the server to convert the data type to fit .

Return IP address instead of the station name
When connecting, the name of the target device is queried. However, if the name resolution is not configured correctly, a connection to this name will fail. So it is possible to return the IP address instead of the name.

Allow access to user files via OPC UA
OPC-UA allows file functions. If you want to use them, turn this option on.

10.2.13 OPC DA Client settings

Connection Active
Shows whether the connection is active or not. If not, you cannot register items and there will be no connection to the PLC. This feature allows you to disable a connection temporarily without deleting it so that you can re-enable it later without having to enter all parameters again.

Write allowed
You can disable the Write function.

Host name
If a computer name is given the OPC server will be used from that machine. For local OPC servers leave the field empty, or use the reserved name "localhost".

Server name
The server name can be explored - press the ">>" button for the dialog Explore OPC DA Servers. But the OPC server name can be entered by hand.

Group update rate
The group rate defines the minimum time the OPC server will return changed values.

Element separator
Most servers use a dot for separator. Few servers differ.
If you can browse and then a data point can not be added, it could be due to the separator.
In the logger window, a corresponding message is then displayed which separator is probably the right one.
If the field is empty, the point is used as separator.

Event settings
- Automatic Refresh: Default - The Refresh is triggered in the new connection establish and when adding/ changing an event instance. The UA server is always answered from the cache.
- Forwarded Refresh: If a refresh is triggered via OPC UA server and none has run in the last 30 minutes, then the Refresh is also triggered via Classic AE Client, otherwise not. The UA server gets the answer from the cache.
- No Refresh - switched off, refresh never comes. The UA server always gets the answer from the cache.

10.2.13.1 Explore OPC DA Servers

The list shows all installed OPC DA servers.
If a computer name is given the list shows the OPC servers on that machine, otherwise the local

available OPC servers are shown.

Choose the desired OPC server.

10.2.14 OpcPipe settings

Name of the Connection
Each connection must have a unique name that allows you to quickly identify it later in the Connections list.

For OpcPipe, some **security settings** can be specified.

Connection Active
You can disable a connection if it is not intended for communication with a PLC. Thus, it is possible to disable it temporarily without having to delete and redefine it.
Number of instances

Write allowed
If this option is disabled, this connection is read-only.

Number of Instances
Allows a certain number of simultaneous connections of an OpcPipe Server connection.

Password
When you establish a connection, the password is used for access control. It is encrypted during transmission over the network.

Memory limit
Each OpcPipe frame can have a size of up to 4 gigabytes. Usually, frames of that size are not required. Since OpcPipe must provide the memory for such frames, you can restrict this value in order to prevent reduced performance. This value should be selected in a way that it will not cause a frame of maximum length to be truncated.
If at runtime, this limit is exceeded, the data will be discarded and the communication partner will receive a corresponding message.

Optional Parameters

Send timeout
If the acknowledgment mode is used, this is the time within which the acknowledgment must arrive; otherwise an error is generated.

Use Send Acknowledge
In OpcPipe, you can use the acknowledgment mode that sends an acknowledgment to the sending station when data has been received.
This mode slows down communication and should only be used in cases where you need to make sure that all data has been received.
The mode should only be used if the destination station supports the acknowledgment mode as well, notifying the partner that a secured transmission has taken place.

10.2.15 Email Client Settings

Name of the Connection

Each connection must have a unique name so that it can be quickly identified later in the connection overview.

Connection Active
You can disable a connection if it is not intended for communication with a PLC. Thus, it is possible to disable it temporarily without having to delete and redefine it.

Server Address
The name of the server for outgoing messages or the IP address is entered here. If the device has been configured for DNS and a DNS server is available in the network, the symbolic name of the destination station can also be entered.
For information on the structure of IP addresses, please refer to the IP Address section.

Destination Port
Port numbers are addresses used within the transport layer to address applications. Port numbers are required for TCP connections. The port is a parallel to the TSAPs for RFC1006 and H1 connections. The port number is a 16-bit number in the range of 1 to 65535. To establish the connection, the following must be noted:
More detailed information on ports

>> Search
The usual port numbers are searched. This only works if the corresponding server is reachable.
The settings for encryption and authentication are adjusted automatically.

Encryption
The email server specifies which encryption should be selected. It is no longer common to connect without encryption.

Authentication
Here is indicated, how is the login procedure is performed.

User
User name to logon to the server. In many cases this is also the email address.

Password
The appropriate password

Sender Address
The email address used to send the email.

EHLO
Extended HELO (EHLO) is an Extended Simple Mail Transfer Protocol (ESMTP) command sent by an email server to identify itself when connecting to another email server to start the process of sending an email. It is followed with the sending email server's domain name. The EHLO command tells the receiving server it supports extensions compatible with ESMTP.

Disable server certificate verification
With this setting the certificate check can be switched off.

10.2.16 Optimizer settings

The communication to the controls can be optimized in many different details.

The **Poll Interval** determines the rate at which the data from the controllers are requested. The default

value is once per second, 1000ms.
Frequent inquiries increase the response time in SCADA systems, but this can also put a strain on the controls.

The **Application timeout** detects when a controller stops responding e.g. Because it has been turned off.

The **Reconnect timeout** determines when the connection will be reestablished after connection errors.

The **Maximum read gap** determines how large the gap (in bytes) between the individually requested items within a data block may be, without forming a new block.
This increases the throughput, but can also be slow with slow controllers - especially with serial-connected systems.
This value is not used when accessing peripherals such as inputs and outputs.
If the value is 1 or zero, all non-contiguous data is requested individually.
With a connection to an S7-1500 or S7-1200 with optimized access, the value 0 deactivates the optimization, which means that reading can be slowed down accordingly.

Optimize reading input and output elements: some controllers don't allow to read multiple inputs or outputs in a single request. In this case uncheck that check box.

Sync read handling

For synchronous reading, this function can be set to normal cyclic reading. Synchronous reading can also read existing data from the buffer. Normally, synchronous read jobs are processed with high priority and separately.
These settings are used with some OPC clients, which read all data synchronously and often run too slowly.

Write handling

When writing to the controller

- The order in which orders are received via OPC. This allows full control, but with large amounts of data this is slow.
- The orders are summarized if there are no gaps. Fast, but when e.g. A particular element must not be written last

Optionally, the written data can be immediately used as read data. If this is not selected, the read data is not updated until the next reading from the PLC.
Writing actualizes read data immediately: the written value is sent to the client before it has been written to the PLC.
Writing forces read update: if an item is written, the client will definitely receive a value change or update the next time it is read, even if it has not changed.

Special settings

Single request in controller calls: very old controllers don't allow to request more data areas in one ethernet frame. Enable it only when the controller response is an error when this option is disabled.

10.2.17 Redundancy Settings

In the dialog, the behavior of the redundancy connection is set.

Number of connections

Indicates how many connections belong to this redundant connection.

Dynamic Master
You can choose Static Master or Dynamic Master.
The **Static Master** always tries to use the first connection. If this fails, it will switch to another connection. Once the first, so the master connection is available again is switched back to this.
The **Dynamic Master** retains the new connection as long as it is working.
Redundancy can be used with two or three connection channels.

Test method

Test connection
It is looked at the connection status, as soon as the is no longer OK, it is switched.

Use PLC status
For some PLCs, the status of the CPU can be determined, e.g. PLC in RUN.
If this option is selected, the status of the CPU is an indicator for switching.

Watchdog test
To make sure that the PLC is running and operating normally, a variable can be incremented in the PLC cycle.
This uses the redundancy switch as an indication whether everything is still OK.
If the **watchdog source** does not change anymore, the system switches over.

Method of writing

Master
It is written over the currently active connection

All (one good)
It is written over all links. If at least one OK returns, the writing has been successful.

All (all good)
It is written over all links. All must return OK for the letter to succeed.

See also Redundancy

10.3 Edit Connection

Name of the Connection
This field displays the connection name which can be changed here.

Connection Active
Shows whether the connection is active or not. If not, you cannot register items and there will be no connection to the PLC. This feature allows you to disable a connection temporarily without deleting it so that you can re-enable it later without having to enter all parameters again.

Write allowed
For some connections, you can disable the Write function.

OPC Access allowed
Using this option, you can allow an OPC server connection to access the connection configured here. Thus, it is possible to access the PLC from a remote PC via this connection.

OPC Write allowed
Using this option, you can enable or disable Write access via OPC.

Simulate connection
Here, you can specify the poll rate the device will use for reading data from the PLC.

Buttons
Depending on the connection type, you can edit further parameters.

Button	Dialog box that opens
Network protocol	It is not possible to change the Plc protocol
Network parameters	TCP/IP Connect Parameters or H1 Connect Parameters
OpcPipe parameters	OpcPipe Parameters
OPC-UA settings	OPCUA settings
OPC DA settings	OPC-DA settings
BACnet settings	BACnet settings
Performance settings	Performance settings
Protocol parameters	Opens the dialog box for protocol-specific parameters.

The number of available options depends on the type of connection.

10.4 Copy Connection

Copying a connection copies all details but the connection name. Change the new name to the final need. The new connection will be shown in the connection list.

10.5 Delete Connection

The highlighted connection will be deleted. This process cannot be undone!
As an alternative to deleting, you can disable a connection using the Switch on/off option.

You can set the security queries in dialog Configuration Client Settings

10.6 Connection switch on/off

To disable a connection, do one of the following:

- Menu: Connection - Switch on/off
- Highlight the connection, right-click and select Switch on/off
- Highlight the connection, press <Ctrl>A

In the Connections list, the word (**off**) is displayed after the type.

You can disable a connection without losing the connection parameters. Later, you can re-enable the connection using the same parameters as before. By default, connections are enabled.

11 Item Syntax

This chapter describes the Item syntax for the corresponding connection (Access Path) that can be used to create new items (tags) .

General information

- Browsing Tree
- System Tree
- Redundancy

Item Syntax for different PLCs

- S7 Item Syntax
- S5 Item Syntax
- Modbus Item-Syntax
- PLC-5 / SLC Item Syntax
- MELSEC-Q Item Syntax
- IEC 60870-5 Item Syntax
- KNX Item Syntax
- Send/Receive Item Syntax
- Configuration Item Syntax

Information for PLCs with online symbols

- S7-1200/S7-1500, Rockwell CompactLogix/ControlLogix, BACnet

The following chapters are generally:

- Bit mask
- Arrays
- Suffixes

11.1 S7 Item Syntax

The S7 item syntax is set up as shown below:

<Area><Data type><Start address>[.Array size][Suffix]

If the data type is BOOL, the bit number is required:

<Area><Data type><Start address><.Bit number>[.Array size]

If the data type is STRING, the string length is required:

<Area><Data type><Start address><.String length>[.Array size][Suffix]

Legend: <> mandatory [ ] optional

<Area>

	Syntax	Orientation[1]	Access Rights	Notes
Data block	**DBx.** **V synonym for DB1**	BYTE	Read / write	With data blocks, the specification of a block number x is required (x = 1 to 65535). A dot must

Instance block	**DIx.**	BYTE	Read / write	appear after the block number.
Flag	**M or F**	BYTE	Read / write	
Timer	**T**	WORD	Read	directly followed by the timer number. The data type is REAL, with suffix S5T STRING
Counter	**Z or C**	WORD	Read / write	directly followed by the counter number.
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	M	I/O	PI /PO	C/T	with Array	useful Suffixes
BIT VT_BOOL	**X**	DB5.X4.3	MX1.3	---	---	---	---	---
BIT VT_BOOL		DB5.4.3	M1.3	E4.3 I4.3 A4.3 O4.3	PE4.5 PI4.5 PA1.3 PO1.3	---	---	---
BYTE VT_UI1	**B** **Byte**	DB5.B2 DB5.Byte2	MB4 MByte4 FB4 FByte4	EB4 EByte4 IB4 IByte4 AB5 AByte5 OB5 OByte5	PEB4 PEByte4 PIB4 PIByte4 PAB5 PAByte5 POB5 POByte5	---	DB5.B2.4 DB5.Byte2.4 MB4.3 MByte4.3 FB4.4 FByte4.4 POB5.3 etc.	KF BCD
WORD VT_UI2 VT_I4*	**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.	KF BCD KT S5T TR D SWAP
INT VT_I2	**I** **Int**	DB5.I3 DB5.Int3	MI4 MInt4 FI4 FInt4	EI4 EInt4 II4 IInt4 AI5 AInt5	PEI4 PEInt4 PII4 PIInt4 PAI5 PAInt5	---	DB5.I3.2 DB5.Int3.2 MI4.4 FInt4.3 AInt5.3 OI5.2	SWAP

				OI5 OInt5	POI5 POInt5		OInt5.5 POInt5.4 etc.	
REAL VT_R4	**R** **Real**	DB5.R2 DB5.REAL2	MR4 MREAL4	ER4 EREAL4 IR4 IREAL4 AR4 AREAL4 OR4 OREAL4	PER4 PEREAL4 PIR4 PIREAL4 PAR4 PAREAL4 POR4 POREAL4	----	DB5.R2.2 DB5.REAL2.4 MR4.5 ER4.4 AREAL4.5 PER4.2 PAR4.7 POR4.3 etc.	KG SWAP
STRING VT_BSTR	**S** **String**	DB5.S1.80 DB5.String1.80	MS2.80 MString2.80	---	---	---	DB5.S1.80.5 DB5.String1.80.5 MS2.80.3 MString2.80.3	KA[1]
S7-STRING VT_BSTR	**G** **SS**	DB5.G2.80 DB5.SS2.80	MG2.80 MSS2.80	---	---	---	DB5.G2.80.5 DB5.SS2.80.5 MG2.80.5 MSS2.80.5	---
S5-STRING VT_BSTR	**SF**	DB5.SF2.80	FSF2.80	---	---	---	DB5.SF2.80.5 MSF2.80.5	---
DOUBLE WORD VT_UI4 VT_R8*	**DW** **Dword**	DB5.DW3 DB5.Dword3	MDW4 MDWord4 FDW4 FDWord4	EDW4 EDWord4 IDW4 IDWord4 ADW5 ADWord5 ODW5 ODWord5 QDW5	PEDW4 PEDWord4 PIDW4 PIDWord4 PADW5 PADWord5 PODW5 PODWord5	---	DB5.DW3 MDW4.2 FDW4.2 EDW4.4 ADWord5.8 PEDW4.8 PADW5.4 PODW5.4 etc.	KF BCD KG T TOD SWAP
DOUBLE INT VT_I4	**D** **DI** **DInt**	DB5.D3 [2] DB5.DI3 DB5.DInt3	MDI4 MDInt4 FDI4 FDInt4	EDI4 EDInt4 IDI4 IDInt4 ADI5 ADInt5 ODI5 ODInt5 QDI5	PEDI4 PEDInt4 PIDI4 PIDInt4 PADI5 PADInt5 PODI5 PODInt5	---	DB5.D3.2 DB5.DI3.2 DB5.DInt3.4 MDI4.5 EDI4.5 ADI5.2 PEDI4.5 PADInt5.2 etc.	BCD T TOD SWAP
QUAD	**QW**	DB5.QW3	MQW4	EQW4	PEQW4	---	DB5.QW3	KF

WORD VT_UI8	**Qword**	DB5.Qword3	MQWord4 FQW4 FQWord4	EQWord4 IQW4 IQWord4 AQW5 AQWord5 OQW5 OQWord5 QQW5	PEQWord4 PIQW4 PIQWord4 PAQW5 PAQWord5 POQW5 POQWord5		MQW4.2 FQW4.2 EQW4.4 AQWord5.8 PEQW4.8 PAQW5.4 POQW5.4 etc.	BCD SWAP
QUAD INT VT_I8	**Q** **QI** **QInt**	DB5.Q3 DB5.QI3 DB5.QInt3	MQI4 MQInt4 FQI4 FQInt4	EQI4 EQInt4 IQI4 IQInt4 AQI5 AQInt5 OQI5 OQInt5 QQI5	PEQI4 PEQInt4 PIQI4 PIQInt4 PAQI5 PAQInt5 POQI5 POQInt5	---	DB5.Q3.2 DB5.QI3.2 DB5.QInt3.4 MQI4.5 EQI4.5 AQI5.2 PEQI4.5 PAQInt5.2 etc.	SWAP
DOUBLE VT_R8	**QR** **QReal**	DB5.QR2 DB5.QREAL2	MQR4 MQREAL4	EQR4 EQREAL4 IQR4 IQREAL4 AQR4 AQREAL4 OQR4 OQREAL4	PEQR4 PEQREAL4 PIQR4 PIQREAL4 PAQR4 PAQREAL4 POQR4 POQREAL4	----	DB5.QR2.2 DB5.QREAL2.4 MQR4.5 EQR4.4 AQREAL4.5 PEQR4.2 PAQR4.7 POQR4.3 etc.	SWAP
Date and Time VT_DATE	**DT**	DB5.DT3	MDT4	EDT4 IDT4 ADT5 ODT5 QDI5	PEDT4 PIDT4 PADT5 PODT5	---	DB5.DT3.2 MDT4.2 EDT4.2 PEDT4.2 etc.	ISO

[1] KA Suffix is not possible with string arrays

[2] Caution! Danger of mix up with S5 syntax for bit (DB5D1.1)

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 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.
For more information on arrays, see also Arrays.

[Suffix]

A value can be represented in another format with the aid of a suffix.
For more information on suffixes, see also Suffixes.

11.2 S5 Item Syntax

The S5 item syntax is set up as shown below:

<Area><Data type><Start address>[.Array size][Suffix]

If the data type is BOOL, the bit number is required:

<Area><Data type><Start address><.Bitnr.>[.Array size]

If the data type is STRING, the string length is required:

<Area><Data type><Start address><.String length>[.Array size][Suffix]

Legend: <> mandatory [ ] optional

<Area>

	Syntax	Orientation[1]	Access rights	Notes
Data block	**DBx**	WORD	Read / write	With blocks, the specification of a block number x is required (x = 1 to 65535). After the block number no dot is required.
Extended data block	**DXx**	WORD	Read / write	
Flag	**M oder F**	BYTE	Read / write	
Timer	**T**	WORD	Read	
Counter	**Z oder C**	WORD	Read	
Input	**E oder I**	BYTE	Read	
Output	**A oder O oder Q**	BYTE	Read / write	
Periphery	**P**	BYTE	Read / write	
Extended Periphery	**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 extended Data blocks

Type VT_Type	**Syntax**	Sample	with Array	useful Suffixes
BIT VT_BOOL	**D**	DB5D4.12	---	---
BYTE VT_UI1	**DB**	DB5DB3	DB5DB3.5	KF, BCD
LEFT BYTE VT_UI1	**DL**	DB5DL4	DB5DL4.2	---
RIGHT BYTE VT_UI1	**DR**	DB5DR2	DB5DR2.5	---
WORD VT_UI2	**DW**	DB5DW4	DB5DW4.5	KF, BCD, KT, S5T, TR, SWAP
DOUBLE WORD VT_UI4	**DD**	DB5DD3	DB5DD3.2	KF, BCD, KG, T, TOD, SWAP
QUAD WORD VT_UI8	**DQ**	DB5DQ3	DB5DQ3.2	KF, BCD, SWAP
STRING VT_BSTR	**S**	DB5S2.3	DB5S2.80.3	KA [1]
S7-STRING VT_BSTR	**SS**	DB5SS2.3	DB5SS2.80.3	---
S5-STRING VT_BSTR	**SF**	DB5SF2.3	DB5SF2.80.3	---

<Data type> for all other areaes

	Syntax	M or F	I/O	P / OB	C / T / RS / AS	with Array	useful Suffixes
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 QB5.5 PB4.2 OB4.3	KF BCD
WORD VT_UI2	**W**	MW4 FW4	EW4 IW4 AW5 QW5	PW2 OW2	C5 T5 RS4 BS4 AS5	MW4.2 FW4.4 EW4.5 IW4.2 AW5.5 QW5.2	KF BCD KT S5T TR SWAP

						PW2.10 OW2.3 C5.5 T5.3 RS4.2 BS4.7 AS5.2	
DOUBLE WORD VT_UI4	**D**	MD4 FD4	ED4 ID4 AD5 QD5	PD5 OD5	---	MD4.4 FD4.2 ED4.6 ID4.2 AD5.6 QD5.2 PD5.4 OD5.9	KF BCD KG T TOD SWAP
QUAD WORD VT_UI8	**Q**	MQ4 FQ4	EQ4 IQ4 AQ5 QQ5	PQ5 OQ5	---	MQ4.4 FQ4.2 EQ4.6 IQ4.2 AQ5.6 QQ5.2 PQ5.4 OQ5.9	KF BCD SWAP
STRING VT_BSTR	**S**	MS4.80 FS4.80	---	---	---	MSS4.80.3 FSS4.80.3	KA [1]
S7-STRING VT_BSTR	**SS**	MSS4.80 FSS4.80	---	---	---	MSS4.80.3 FSS4.80.3	---
S5-STRING VT_BSTR	**SF**	MSF4.80 FSF4.80	---	---	---	MSF4.80.3 FSF4.80.3	---

[1] KA Suffix is not possible with string arrays

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 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: FX12.1: Bit 1 of flag byte 12 – a flag bit is addressed here.
Note:
Writing to bits of a S5 PLC is not permitted. Please read the byte/word, set the bit and write the byte/word back.

[.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: DB10DW2.5.
For more information on arrays, see also Arrays.

[Suffix]

A value can be represented in another format with the aid of a suffix.
For more information on suffixes, see also Suffixes.

11.3 Modbus Item Syntax

The Modbus item syntax is set up as shown below:

<Area><Data type><Start address>[.Array size][Suffix]

If the data type is BOOL, the bit number is required:

<Area><Data type><Start address><.Bit number>[.Array size]

The following item syntax allows to address a different UnitID as set up at the connection parameters:

[UnitID.]<Area><Data type><Start address><.Bit number>[.Array size][Suffix]

Legend: <> mandatory [ ] optional

[UnitID.]

The UnitId item syntax is specified for an item with the literals "ID" followed by a number and a dot.
The range is 0 - 255.
If the UnitID is not present in the Itemsyntax, the parameterized UnitID is transferred to the PLC.
Samples:
Id1.40001
Id2.R2
Id3.S5.30

<Area>

	Syntax	Syntax with 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** **IR**	3xxxxx	WORD	Read

Register (Holding Register)	**R** **HR**	4xxxxx	WORD	Read / write
Discrete Inputs Oktal [2]	**J**	-	BIT	Read
Discrete Outputs Oktal [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 entries.

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", discrete outputs by letter "O" or number 0, Input registers by "IR" or "3", Registers by "R" or "4".

<Data type>

Data type VT-Typ	**Syntax**	Syntax R	Syntax I / O	Syntax ER	with Array	useful Suffixes
BIT VT_BOOL	**X** [1]	RX5.2 [1] HRX5.2 [1] 4X5.2 [1]	E5 I5 DE5 DI5 100005 A5 O5 Q5 DA5 DO5 DQ5 000005	ERX5.2 [1] 3X5.2 [1]	I1.10 O2.5	
LEFT CHAR RIGHT CHAR VT_I1	**LC** **RC**	RLC5 RRC5	--- ---	ERLC5 ERRC5	RLC5.2 RRC5.2	
LEFT BYTE RIGHT BYTE VT_UI1	**LB** **RB**	RLB5 RRB5	--- ---	ERLB5 ERRB5	RLB5.2 RRB5.2	
INT VT_I2	**ohne** **I**	R5 HR5 45 RI5 HRI5 4I5	---	ER5 IR5 35 ERI5 IRI5 3I5	R5.2 HR5.2 45.2 ERI5.2 IRI5.2 3I5.2	SWAP [2]
WORD VT_UI2	**W**	RW5 HRW5 4W5	----	ERW5 IRW5 3W5	RW5.2 HRW5.2 4W5.2 ERW5.2 IRW5.2 3W5.2	SWAP [2]
DOUBLE INT VT_I4	**D**	RD5 HRD5 4D5	---	ERD5	RD5.2 HRD5.2 4D5.2	TOD SWAP [2]

	DI	RDI5 HRDI5 4DI5				
DOUBLE WORD VT_UI4	**DW**	RDW5 HRDW5 4DW5	---	ERDW5 IRDW5 3DW5	RDW5.2 HRDW5.2 4DW5.2 ERDW5.2 IRDW5.2 3DW5.2	TOD SWAP [2]
QUAD INT VT_I8	**Q** **QI**	RQ5 HRQ5 4Q5 RQI5 HRQI5 4QI5	---	ERQ5 IRQ5 3Q5	ERQ5.2 IRQ5.2 3Q5.2	SWAP [2]
QUAD WORD VT_UI8	**QW**	RQW5 HRQW5 4QW5	---	ERQW5 IRQW5 3QW5	RQW5.2 HRQW5.2 4QW5.2 ERQW5.2 IRQW5.2 3QW5.2	SWAP [2]
REAL VT_R4	**R**	RR5	----	ERR5	RR5.2 ERR5.2	SWAP [2]
DOUBLE VT_R8	**QR**	RQR5	---	ERQR5	RQR5.2 ERQR5.2	SWAP [2]
STRING VT_BSTR [2]	**S**	RS5.80	----	ERS5.80	RS5.80.3	SWAP [2] KA [3]

[1] Diskrete Inputs and Outputs are one single bit in the PLC. When reading from Register or Input Registers, the whole register is read and the bit is extracted.

Note:
Writing of individual bits in the register and input register area is possible if the writing of bits in the word is activated in the configuration of the connection.
The entire register is then read, the bit is set or deleted and the register is written back again.

[2] The suffix SWAP is possible for a string to give the bytes the correct order. For the other data types all bytes are changed in the order.

[3] KA Suffix is not possible with string arrays

<Start address>

The start address specifies the address starting at which read or write accesses begins.
Example:
ER5 -> Input Register 5
O12 -> Output 12

<.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: HRD5.3
For more information on arrays, see also Arrays.

[Suffix]

A value can be represented in another format with the aid of a suffix.
For more information on suffixes, see also Suffixes.

11.4 PLC-5 / SLC Item Syntax

The item syntax for PLC-5 and SLC is set up as shown below:

<Area>[File number]<Start address>[Array size][Tani-Suffix]

<Area>[File number]<Start address>[Array size][RSLinX-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	Read	1	octal
Integer	**N**	WORD	Read / write	7	decimal
Binary	**B**	WORD	Read / write	3	decimal
Float	**F**	DOUBLE WORD	Read / write	8	decimal
String	**ST**	SLC-String	Read / write	9	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	9	decimal
Long Integer	**L**	DOUBLE WORD	Read / write	9	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 number]

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>.

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	Description
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	Bit 15 in word 5 of binary file 12
F8:0	Float 0
N23:4	word 4 of integer file 23
N23:4/2	Bit 2 in word 4 of integer file 23 = Bit 66 in integer file 23
N23/66	Bit 66 in integer file 23

[Array size]

Arrays are created to combine several units of one data type in a field together. Arrays are only possible for word areas and float areas. The array size is initiated with a #.
Examples: N23:4#10

[Tani-Suffix]

A value can be represented in another format with the aid of a suffix. If no suffix is specified, the formats from the column Orientation apply.
For more information on suffixes, see also Suffixes.

[RsLinx-Suffix]

The alternative syntax allows you to use one or more suffixes. Every suffix is initiated. The order is arbitrary.

NOTE:
The combination of RSLINX-Suffixes and Tani-Suffixes is not possible.

Suffix	Description	File Types
,N	Values are represented with sign. (Default is: no sign)	I, O, N, B, S, A
,M	Values are represented in Motorola-Format. (Default is: Intel-Format)	I, O, N, B, S, A, F
,L<xxx>	The len specifier for RSLinX Format. xxx is the count of array length	all
,SC<x	Values are interpreted as a zero-	N, A

xx>	terminated string. The maximum length is XXX.	

11.5 MELSEC-Q Item Syntax

Two syntax versions are available for setting up items.

1. Simple version:

<Area><Start address>[.Array size][Suffix]

2. Expanded version:

<Area><.Type><Start address>[.Array size][Suffix]

Legend: <> mandatory [ ] optional

REMEMBER :

- With the expanded version, a period is required between the <area> and the <type>. If the period 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 arrays in bit areas are not possible.
- Bit arrays in WORD areas with HEX representation, are not possible.

<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
Retentive Timer Contact	**STS**	BIT	decimal
Retentive Timer Coil	**STC**	BIT	decimal
Retentive Timer Current Value	**STN**	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	HEX

[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>

Typ VT_Typ	**Syntax**	Simple Syntax	ex:Bit area	Ex: Word area	with Array in Bit area	with Array in Word area	useful Suffixes
BIT VT_BOOL	**X**	DY1	-----	D.X1.2	-----	-----	-----
BIT [1] VT_BOOL	**-----**	DY1	-----	D.1.2	-----	-----	-----
BYTE VT_UI2	**B** **BYTE**	-----	Y.B10 Y.BYTE10	R.B1 R.BYTE1	Y.B10.5 Y.BYTE10.5	R.B1.3 R.BYTE1.3	-----
INT VT_I2	**I** **INT**	Y.I10 Y.INT10	Y.I10 Y.INT10	R.I2 R.INT2	Y.I10.3 Y.INT10.3	R.I2.3 R.INT2.3	-----
WORD VT_UI2	**W** **WORD**	R20	Y.W10 Y.WORD10	R.W2 R.WORD2	Y.W10.3 Y.WORD10.3	R.W2.3 R.WORD2.3	-----
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	-----
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	-----
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	-----	-----	-----

[1] 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
For more information on arrays, see also Arrays.

[Suffix]
Suffixes can be used to represent a value in another format.
Example: D20.300KF
For more information on suffixes, see also Suffixes.

11.6 KNX Item Syntax

The KNX item syntax is set up as shown below:

<Address>[$<Data type>]

Legend: <...> mandatory [<...>] optional

<Address>
KNX group address to access.
Possible syntax:
· <main>/<middle>/<subgroup> - 3-level address, main = 0 .. 31, middle = 0 .. 7, subgroup = 0 .. 255
· <main>/<subgroup> - 2-level address, main = 0 .. 31, subgroup = 0 .. 2047
· <group> - 1-level address, group = 0 .. 65535
Example: 1/0/1

<Data type>
The data type to use. If not given, the data is handled as byte array. Exactly one of the numeric code, encoding description or the name shall be used.

Numeric	Encoding	Name	Datatype
1.001	B1	DPT_Switch	Bool
1.002	B1	DPT_Bool	Bool
1.003	B1	DPT_Enable	Bool
1.004	B1	DPT_Ramp	Bool
1.005	B1	DPT_Alarm	Bool
1.006	B1	DPT_BinaryValue	Bool
1.007	B1	DPT_Step	Bool
1.008	B1	DPT_UpDown	Bool
1.009	B1	DPT_OpenClose	Bool
1.010	B1	DPT_Start	Bool
1.011	B1	DPT_State	Bool
1.012	B1	DPT_Invert	Bool
1.013	B1	DPT_DimSendStyle	Bool
1.014	B1	DPT_InputSource	Bool
1.015	B1	DPT_Reset	Bool
1.016	B1	DPT_Ack	Bool
1.017	B1	DPT_Trigger	Bool
1.018	B1	DPT_Occupancy	Bool
1.019	B1	DPT_Window_Door	Bool
1.021	B1	DPT_LogicalFunction	Bool
1.022	B1	DPT_Scene_AB	Bool

1.023	B1	DPT_ShutterBlinds_Mode	Bool
1.100	B1	DPT_Heat/Cool	Bool
2.001	B2	DPT_Switch_Control	Bool[2]
2.002	B2	DPT_Bool_Control	Bool[2]
2.003	B2	DPT_Enable_Control	Bool[2]
2.004	B2	DPT_Ramp_Control	Bool[2]
2.005	B2	DPT_Alarm_Control	Bool[2]
2.006	B2	DPT_BinaryValue_Control	Bool[2]
2.007	B2	DPT_Step_Control	Bool[2]
2.008	B2	DPT_Direction1_Control	Bool[2]
2.009	B2	DPT_Direction2_Control	Bool[2]
2.010	B2	DPT_Start_Control	Bool[2]
2.011	B2	DPT_State_Control	Bool[2]
2.012	B2	DPT_Invert_Control	Bool[2]
3.007	B4	DPT_Control_Dimming	Bool[4]
3.008	B4	DPT_Control_Blinds	Bool[4]
4.001	A8	DPT_Char_ASCII	Uint 8
4.002	A8	DPT_Char_8859_1	Uint 8
5.001	U8	DPT_Scaling	Uint 8
5.003	U8	DPT_Angle	Uint 8
5.004	U8	DPT_Percent_U8	Uint 8
5.005	U8	DPT_DecimalFactor	Uint 8
5.006	U8	DPT_Tariff	Uint 8
5.010	U8	DPT_Value_1_Ucount	Uint 8
6.001	V8	DPT_Percent_V8	Int 8
6.010	V8	DPT_Value_1_Count	Int 8
6.020	B5N3	DPT_Status_Mode3	Structure
7.001	U16	DPT_Value_2_Ucount	Uint 16
7.002	U16	DPT_TimerPeriodMsec	Uint 16
7.003	U16	DPT_TimerPeriod10MSec	Uint 16
7.004	U16	DPT_TimerPeriod100MSec	Uint 16
7.005	U16	DPT_TimerPeriodSec	Uint 16
7.006	U16	DPT_TimerPeriodMin	Uint 16
7.007	U16	DPT_TimerPeriodHrs	Uint 16
7.010	U16	DPT_PropDataType	Uint 16
7.011	U16	DPT_Length_mm	Uint 16
7.012	U16	DPT_UElCurrentmA	Uint 16
7.013	U16	DPT_Brightness	Uint 16
8.001	V16	DPT_Value_2_Count	Int 16
8.002	V16	DPT_DeltaTimeMsec	Int 16
8.003	V16	DPT_DeltaTime10MSec	Int 16
8.004	V16	DPT_DeltaTime100MSen	Int 16
8.001	V16	DPT_DeltaTimeSec	Int 16

8.006	V16	DPT_DeltaTimeMin	Int 16
8.007	V16	DPT_DeltaTimeHrs	Int 16
8.010	V16	DPT_Percent_V16	Int 16
8.011	V16	DPT_Rotation_Angle	Int 16
9.001	F16	DPT_Value_Temp	Float
9.002	F16	DPT_Value_Tempd	Float
9.003	F16	DPT_Value_Tempa	Float
9.004	F16	DPT_Value_Lux	Float
9.005	F16	DPT_Value_Wsp	Float
9.006	F16	DPT_Value_Pres	Float
9.007	F16	DPT_Value_Hunidity	Float
9.008	F16	DPT_Value_AirQuality	Float
9.010	F16	DPT_Value_Time1	Float
9.011	F16	DPT_Value_Time2	Float
9.020	F16	DPT_Value_Volt	Float
9.021	F16	DPT_Value_Curr	Float
9.022	F16	DPT_PowerDensity	Float
9.023	F16	DPT_KelvinPerPercent	Float
9.024	F16	DPT_Power	Float
9.025	F16	DPT_Value_Volume_Flow	Float
9.026	F16	DPT_Rain_Amount	Float
9.027	F16	DPT_Value_Temp_F	Float
9.028	F16	DPT_Value_Wsp_kmh	Float
10.001	N3U5r2U6r2U6	DPT_TimeOfDay	Structure
11.001	r3U5r4U4r1U7	DPT_Date	Structure
12.001	U32	DPT_Value_4_Ucount	Uint 32
13.001	V32	DPT_Value_4_Count	Int 32
13.002	V32	DPT_FlowRate_m3/h	Int 32
13.010	V32	DPT_ActiveEnergy	Int 32
13.011	V32	DPT_ApparantEnergy	Int 32
13.012	V32	DPT_ReactiveEnergy	Int 32
13.013	V32	DPT_ActiveEnergy_kWh	Int 32
13.014	V32	DPT_ApparentEnergy_kVAh	Int 32
13.015	V32	DPT_ReactiveEnergy_kVARh	Int 32
13.100	V32	DPT_LongDeltaTimeSec	Int 32
14.000	F32	DPT_Value_Acceleration	Float
14.001	F32	DPT_Value_Acceleration_Angular	Float
14.002	F32	DPT_Value_Activation_Energy	Float
14.003	F32	DPT_Value_Activity	Float
14.004	F32	DPT_Value_Mol	Float
14.005	F32	DPT_Value_Amplitude	Float
14.006	F32	DPT_Value_AngleRad	Float

14.007	F32	DPT_Value_AngleDeg	Float
14.008	F32	DPT_Value_AngularMomentum	Float
14.009	F32	DPT_Value_Angular_Velocity	Float
14.010	F32	DPT_Value_Area	Float
14.011	F32	DPT_Value_Capacitance	Float
14.012	F32	DPT_Value_Charge_DensitySurface	Float
14.013	F32	DPT_Value_Charge_DensityVolume	Float
14.014	F32	DPT_Value_Compressibility	Float
14.015	F32	DPT_Value_Conductance	Float
14.016	F32	DPT_Value_Electrocal_Conductivity	Float
14.017	F32	DPT_Value_Density	Float
14.018	F32	DPT_Value_Electric_Charge	Float
14.019	F32	DPT_Value_Electric_Current	Float
14.020	F32	DPT_Value_Electric_CurrentDensity	Float
14.021	F32	DPT_Value_Electric_DipoleMoment	Float
14.022	F32	DPT_Value_Electric_Displacement	Float
14.023	F32	DPT_Value_Electric_FieldStrength	Float
14.024	F32	DPT_Value_Electric_Flux	Float
14.025	F32	DPT_Value_Electric_FluxDensity	Float
14.026	F32	DPT_Value_Electric_Polatization	Float
14.027	F32	DPT_Value_Electric_Potential	Float
14.028	F32	DPT_Value_Electric_PotentialDifference	Float
14.029	F32	DPT_Value_ElectromagneticMoment	Float
14.030	F32	DPT_Value_Electromotive_Force	Float
14.031	F32	DPT_Value_Energy	Float
14.032	F32	DPT_Value_Force	Float
14.033	F32	DPT_Value_Frequency	Float
14.034	F32	DPT_Value_AngularFrequency	Float
14.035	F32	DPT_Value_Heat_Capacity	Float

14.036	F32	DPT_Value_Heat_FlowRate	Float
14.037	F32	DPT_Value_Heat_Quantity	Float
14.038	F32	DPT_Value_Impedance	Float
14.039	F32	DPT_Value_Length	Float
14.040	F32	DPT_Value_Light_Quanti ty	Float
14.041	F32	DPT_Value_Luminance	Float
14.042	F32	DPT_Value_Luminous_Flux	Float
14.043	F32	DPT_Value_Luminous_Intensity	Float
14.044	F32	DPT_Value_Magnetic_FieldStrength	Float
14.045	F32	DPT_Value_Magnetic_Flux	Float
14.046	F32	DPT_Value_Magnetic_FluxDensity	Float
14.047	F32	DPT_Value_Magnetic_Moment	Float
14.048	F32	DPT_Value_Magnetic_Polarization	Float
14.049	F32	DPT_Value_Manetization	Float
14.050	F32	DPT_Value_MagnetomotiveForce	Float
14.051	F32	DPT_Value_Mass	Float
14.052	F32	DPT_Value_MassFlux	Float
14.053	F32	DPT_Value_Momentum	Float
14.054	F32	DPT_Value_Phase_AngleRad	Float
14.055	F32	DPT_Value_Phase_AngleDeg	Float
14.056	F32	DPT_Value_Power	Float
14.057	F32	DPT_Value_Power_Factor	Float
14.058	F32	DPT_Value_Pressure	Float
14.059	F32	DPT_Value_Reactance	Float
14.060	F32	DPT_Value_Resistance	Float
14.061	F32	DPT_Value_Resistivity	Float
14.062	F32	DPT_Value_SelfInductance	Float
14.063	F32	DPT_Value_SolidAngle	Float
14.064	F32	DPT_Value_Sound_Intensity	Float
14.065	F32	DPT_Value_Speed	Float
14.066	F32	DPT_Value_Stress	Float
14.067	F32	DPT_Value_Surface_Tension	Float

14.068	F32	DPT_Value_Common_Temperature	Float
14.069	F32	DPT_Value_Absolute_Temperature	Float
14.070	F32	DPT_Value_Temperature Difference	Float
14.071	F32	DPT_Value_Thermal_Capacity	Float
14.072	F32	DPT_Value_Thermal_Conductivity	Float
14.073	F32	DPT_Value_ThermoelectricPower	Float
14.074	F32	DPT_Value_Time	Float
14.075	F32	DPT_Value_Torque	Float
14.076	F32	DPT_Value_Volume	Float
14.077	F32	DPT_Value_Volume_Flux	Float
14.078	F32	DPT_Value_Weight	Float
14.079	F32	DPT_Value_Work	Float
15.000	U4U4U4U4U4U4B4N4	DPT_Access_Data	Structure
16.000	A112	DPT_String_ASCII	String
16.001	A112	DPT_String_8859_1	String
17.001	r2U6	DPT_SceneNumber	Structure
18.001	B1r1U6	DPT_SceneControl	Structure
19.001	U8r4U4r3U5U3U5r2U6r2U6B16	DPT_DateTime	Structure
20.001	N8	DPT_SCLOMode	Uint 8
20.002	N8	DPT_BuildingMode	Uint 8
20.003	N8	DPT_OccMode	Uint 8
20.004	N8	DPT_Priority	Uint 8
20.005	N8	DPT_LightApplicationMode	Uint 8
20.006	N8	DPT_ApplicationArea	Uint 8
20.007	N8	DPT_AlarmClassType	Uint 8
20.008	N8	DPT_PSUMode	Uint 8
20.011	N8	DPT_ErrorClass_System	Uint 8
20.012	N8	DPT_ErrorClass_HVAC	Uint 8
20.013	N8	DPT_Time_Delay	Uint 8
20.014	N8	DPT_Beaufort_Wind_Force_Scale	Uint 8
20.017	N8	DPT_SensorSelect	Uint 8
20.020	N8	DPT_ActuatorConnectType	Uint 8
20.100	N8	DPT_FuelType	Uint 8
20.101	N8	DPT_BurnerType	Uint 8
20.102	N8	DPT_HVACMode	Uint 8
20.103	N8	DPT_DHWMode	Uint 8
20.104	N8	DPT_LoadPriority	Uint 8
20.105	N8	DPT_HVACContrMode	Uint 8

20.106	N8	DPT_HVACEmergMode	Uint 8
20.107	N8	DPT_ChangeoverMode	Uint 8
20.108	N8	DPT_ValveMode	Uint 8
20.109	N8	DPT_DamperMode	Uint 8
20.110	N8	DPT_HeaderMode	Uint 8
20.111	N8	DPT_FanMode	Uint 8
20.112	N8	DPT_MasterSlaveMode	Uint 8
20.113	N8	DPT_StatusRoomSetp	Uint 8
20.120	N8	DPT_ADAType	Uint 8
20.121	N8	DPT_BackupMode	Uint 8
20.122	N8	DPT_StartSynchronization	Uint 8
20.600	N8	DPT_Behaviour_Lock_Unlock	Uint 8
20.601	N8	DPT_Behaviour_Bus_Power_Up_Down	Uint 8
20.602	N8	DPT_DALI_Fade_Time	Uint 8
20.603	N8	DPT_BlinkingMode	Uint 8
20.604	N8	DPT_LightControlMode	Uint 8
20.605	N8	DPT_SwitchPBModel	Uint 8
20.606	N8	DPT_PBAction	Uint 8
20.607	N8	DPT_DimmPBModel	Uint 8
20.608	N8	DPT_SwitchOnMode	Uint 8
20.609	N8	DPT_LoadTypeSet	Uint 8
20.610	N8	DPT_LoadTypeDetected	Uint 8
20.801	N8	DPT_SABExcept-Behaviour	Uint 8
20.802	N8	DPT_SABBehaviour_Lock_Unlock	Uint 8
20.803	N8	DPT_SSSBMode	Uint 8
20.804	N8	DPT_BlindsControlMode	Uint 8
20.1000	N8	DPT_CommMode	Uint 8
20.1001	N8	DPT_AddInfoTypes	Uint 8
20.1002	N8	DPT_RF_ModeSelect	Uint 8
20.1003	N8	DPT_RF_FilterSelect	Uint 8
21.001	B8	DPT_StatusGen	Bool[8]
21.002	B8	DPT_Device_Control	Bool[8]
21.100	B8	DPT_ForceSign	Bool[8]
21.101	B8	DPT_ForceSignCool	Bool[8]
21.102	B8	DPT_StatusRHC	Bool[8]
21.103	B8	DPT_StatusSDHWC	Bool[8]
21.104	B8	DPT_FullTypeSet	Bool[8]
21.105	B8	DPT_StatusRCC	Bool[8]
21.106	B8	DPT_StatusAHU	Bool[8]
21.601	B8	DPT_LightActuatorErrorInfo	Bool[8]
21.1000	B8	DPT_RF_ModeInfo	Bool[8]
21.1001	B8	DPT_RF_FilterInfo	Bool[8]

21.1010	B8	DPT_Channel_Activation_16	Bool[8]
22.100	B16	DPT_StatusDHWC	Bool[16]
22.101	B16	DPT_StatusRHCC	Bool[16]
22.1000	B16	DPT_Media	Bool[16]
22.1010	B16	DPT_Media	Bool[16]
23.001	N2	DPT_OnOff_Action	Bool[2]
23.002	N2	DPT_Alarm_Reaction	Bool[2]
23.003	N2	DPT_UpDown_Action	Bool[2]
23.102	N2	DPT_HVAC_PB_Action	Bool[2]
24.001	A[n]	DPT_VarString_8859_1	String
27.001	B32	DPT_CombinedInfoOnOff	Bool[32]
28.001	A[n]$UTF8	DPT_UTF-8	String
29.010	V64	DPT_ActiveEnergy_V64	Int 64
29.011	V64	DPT_ApparentEnergy_V64	Int 64
29.012	V64	DPT_ReactiveEnergy_V64	Int 64
--	U64	--	Uint 64
30.1010	B24	DPT_Channel_Activation_24	Bool[24]
31.101	N3	DPT_PB_Action_HVAC_Extended	Bool[3]
231.001	A8A8A8A8	DPT_Locale_ASCII	String
234.001	A8A8	DPT_LanguageCodeAlpha2_ASCII	String
234.002	A8A8	DPT_RegionCodeAlpha2_ASCII	String

Examples:
1/0/1$DPT_Switch - Monitor the light switch group 1/0/1, or switch all lights belonging to that group
1/0/1$B1 - the same, using the encoding syntax
1/0/1$1.001 - the same, using the numeric syntax

11.7 IEC60870-5-104 Item Syntax

The IEC60870-5-104 item syntax is set up as shown below:

[<CA>.]<Data type>[.<IOA>][.<Option>][.<more Options ...>]

Legend: <...> mandatory [<...>] optional

<CA>

Specifies the CA (Common Address) to use. Required if the connection is set to "Allow all CAs". Not used if a CA is specified in the connection settings.

<IOA>

Specifies the IOA (Information Object Address) to use. Required for ordinary Read and Write operations. Not used for special commands.

<Data type> (read = monitor direction)

	Syntax[1]	Type[2]	Access Rights	Options[3]

Single Point Information	**M_SP** **M_SP_NA_1** **M_SP_TA_1** **M_SP_TB_1**	Bool	Read	Value/SPI - 0 OFF, 1 ON BL - 0 Not Blocked, 1 Blocked SB - 0 Not Substituted, 1 Substituted NT - 0 Topical, 1 Not Topical IV - 0 Valid, 1 Invalid Timestamp - transmitted timestamp, ununsed for M_SP_NA_1
Double Point Information	**M_DP** **M_DP_NA_1** **M_DP_TA_1** **M_DP_TB_1**	Bool[2]	Read	Value/DPI - 00 intermediate state, 01 OFF, 10 ON, 11 invalid state BL - 0 Not Blocked, 1 Blocked SB - 0 Not Substituted, 1 Substituted NT - 0 Topical, 1 Not Topical IV - 0 Valid, 1 Invalid Timestamp - transmitted timestamp, ununsed for M_DP_NA_1
Step Position Information	**M_ST** **M_ST_NA_1** **M_ST_TA_1** **M_ST_TB_1**	Int 8	Read	Value - Position Value T - 0 Not Topical, 1 Topical OV - 0 No Overflow, 1 Overflow BL - 0 Not Blocked, 1 Blocked SB - 0 Not Substituted, 1 Substituted NT - 0 Topical, 1 Not Topical IV - 0 Valid, 1 Invalid Timestamp - transmitted timestamp, ununsed for M_ST_NA_1
Binary State Information	**M_BO** **M_BO_NA_1** **M_BO_TA_1** **M_BO_TB_1**	Bool[32]	Read	Value/BSI - Binary State OV - 0 No Overflow, 1 Overflow BL - 0 Not Blocked, 1 Blocked SB - 0 Not Substituted, 1 Substituted NT - 0 Topical, 1 Not Topical IV - 0 Valid, 1 Invalid Timestamp - transmitted timestamp, ununsed for

				M_BO_NA_1
Measured Value, Normalized	**M_ME_NV** **M_ME_ND_1** **M_ME_NA_1** **M_ME_TA_1** **M_ME_TD_1**	Float	Read	Value/NVA - Normalized Value OV - 0 No Overflow, 1 Overflow, ununsed for M_ME_ND_1 BL - 0 Not Blocked, 1 Blocked, ununsed for M_ME_ND_1 SB - 0 Not Substituted, 1 Substituted, ununsed for M_ME_ND_1 NT - 0 Topical, 1 Not Topical, ununsed for M_ME_ND_1 IV - 0 Valid, 1 Invalid, ununsed for M_ME_ND_1 Timestamp - transmitted timestamp, ununsed for M_ME_ND_1, M_ME_NA_1
Measured Value, Scaled	**M_ME_SV** **M_ME_NB_1** **M_ME_TB_1** **M_ME_TE_1**	Int 16	Read	Value/SVA - Scaled Value OV - 0 No Overflow, 1 Overflow BL - 0 Not Blocked, 1 Blocked SB - 0 Not Substituted, 1 Substituted NT - 0 Topical, 1 Not Topical IV - 0 Valid, 1 Invalid Timestamp - transmitted timestamp, ununsed for M_ME_NB_1
Measured Value, Short Float	**M_ME_FV** **M_ME_NC_1** **M_ME_TC_1** **M_ME_TF_1**	Float	Read	Value/FVA - Floating-point Value OV - 0 No Overflow, 1 Overflow BL - 0 Not Blocked, 1 Blocked SB - 0 Not Substituted, 1 Substituted NT - 0 Topical, 1 Not Topical IV - 0 Valid, 1 Invalid Timestamp - transmitted timestamp, ununsed for M_ME_NC_1
Binary Counter	**M_IT** **M_IT_NA_1**	Int 32	Read	Value/BCR - Binary Counter Reading

	M_IT_TA_1 **M_IT_TB_1**			SQ - Sequence Counter CY - 0 No Carry, 1 Carry SA - 0 Not Adjusted, 1 Counter Adjusted IV - 0 Valid, 1 Invalid Timestamp - transmitted timestamp, ununsed for M_IT_NA_1
Single Event of Protection Equipment	**M_EP_EV** **M_EP_TA_1** **M_EP_TD_1**	Bool[2]	Read	Value/ES - Event State EI - 0 Elapsed Time Valid, 1 Elapsed Time Invalid BL - 0 Not Blocked, 1 Blocked SB - 0 Not Substituted, 1 Substituted NT - 0 Topical, 1 Not Topical IV - 0 Valid, 1 Invalid ElapsedTime - Elapsed Time Timestamp - transmitted timestamp
Start Events of Protection Equipment	**M_EP_SE** **M_EP_TB_1** **M_EP_TE_1**	Bool	Read	Value/GS - General Start of Operation SL1 - Start of Operation Phase L1 SL2 - Start of Operation Phase L2 SL3 - Start of Operation Phase L3 SIE - Start of Operation Earth Current SRD - Start of Operation in Reverse Direction EI - 0 Elapsed Time Valid, 1 Elapsed Time Invalid BL - 0 Not Blocked, 1 Blocked SB - 0 Not Substituted, 1 Substituted NT - 0 Topical, 1 Not Topical IV - 0 Valid, 1 Invalid RelayDurationTime - Relay Duration Time Timestamp - transmitted timestamp
Output Circuit Information of Protection	**M_EP_OC** **M_EP_TC_1** **M_EP_TF_1**	Bool	Read	Value/GC - General Command to Output Circuit

Equipment				CL1 - Command to Output Circuit Phase L1 CL2 - Command to Output Circuit Phase L2 CL3 - Command to Output Circuit Phase L3 EI - 0 Elapsed Time Valid, 1 Elapsed Time Invalid BL - 0 Not Blocked, 1 Blocked SB - 0 Not Substituted, 1 Substituted NT - 0 Topical, 1 Not Topical IV - 0 Valid, 1 Invalid RelayOperatingTime - Relay Operating Time Timestamp - transmitted timestamp
Status and Change Detection	**M_PS** **M_PS_NA_1**	Bool[16]	Read	Value/State - Current State CD - Change Detection OV - 0 No Overflow, 1 Overflow BL - 0 Not Blocked, 1 Blocked SB - 0 Not Substituted, 1 Substituted NT - 0 Topical, 1 Not Topical IV - 0 Valid, 1 Invalid

[1]First line captures all opcodes for the type, other lines capture only the specific opcode.
[2]The data type of the Value element. Most types support additional information which may have different types.
[3]For read, only one option may be given. If missing, all data is returned as a structure.

<Data type> (write = command direction, simple values)

	Syntax[1]	Type[2]	Access Rights	Options[3]
Single Point Command	**C_SC** **C_SC_NA_1** **C_SC_TA_1**	Bool	Write	write mode (required if any options are given): D/Direct - one-step write SE/SelectExecute - two-step write Timestamp mode (only if not specified via type, default NTS): TS/Timestamp - send timestamp (use C_xx_Tx_1 opcode) NTS/NoTimestamp - don't send timestamp

				(use C_xx_Nx_1 opcode) Write Qualifier (default 0 - unspecified): SP/ShortPulse - send the Short Pulse qualifier LP/LongPulse - send the Long Pulse qualifier P/Persistent - send the Persistent qualifier QU0 .. QU31 - specify numerically
Double Point Command	**C_DC** **C_DC_NA_1** **C_DC_TA_1**	Bool[2]	Write	write mode (required if any options are given): D/Direct - one-step write SE/SelectExecute - two-step write Timestamp mode (only if not specified via type, default NTS): TS/Timestamp - send timestamp (use C_xx_Tx_1 opcode) NTS/NoTimestamp - don't send timestamp (use C_xx_Nx_1 opcode) Write Qualifier (default 0 - unspecified): SP/ShortPulse - send the Short Pulse qualifier LP/LongPulse - send the Long Pulse qualifier P/Persistent - send the Persistent qualifier QU0 .. QU31 - specify numerically
Regulating Command	**C_RC** **C_RC_NA_1** **C_RC_TA_1**	Bool[2]	Write	write mode (required if any options are given): D/Direct - one-step write SE/SelectExecute - two-step write Timestamp mode (only if not specified via type, default NTS): TS/Timestamp - send timestamp (use C_xx_Tx_1 opcode) NTS/NoTimestamp - don't send timestamp (use C_xx_Nx_1 opcode) Write Qualifier (default 0 - unspecified): SP/ShortPulse - send the Short Pulse qualifier

				LP/LongPulse - send the Long Pulse qualifier P/Persistent - send the Persistent qualifier QU0 .. QU31 - specify numerically
Binary State Command	**C_BO** **C_BO_NA_1** **C_BO_TA_1**	Bool[32]	Write	write mode (required if any options are given): D/Direct - one-step write Timestamp mode (only if not specified via type, default NTS): TS/Timestamp - send timestamp (use C_xx_Tx_1 opcode) NTS/NoTimestamp - don't send timestamp (use C_xx_Nx_1 opcode)
Set-Point Command, Normalized Value	**C_SE_NV** **C_SE_NA_1** **C_SE_TA_1**	Float	Write	write mode (required if any options are given): D/Direct - one-step write SE/SelectExecute - two-step write Timestamp mode (only if not specified via type, default NTS): TS/Timestamp - send timestamp (use C_xx_Tx_1 opcode) NTS/NoTimestamp - don't send timestamp (use C_xx_Nx_1 opcode) Write Qualifier (default 0 - unspecified): QL0 .. QL127 - specify numerically
Set-Point Command, Scaled Value	**C_SE_SV** **C_SE_NB_1** **C_SE_TB_1**	Int 16	Write	write mode (required if any options are given): D/Direct - one-step write SE/SelectExecute - two-step write Timestamp mode (only if not specified via type, default NTS): TS/Timestamp - send timestamp (use C_xx_Tx_1 opcode) NTS/NoTimestamp - don't send timestamp (use C_xx_Nx_1 opcode) Write Qualifier (default 0 - unspecified):

				QL0 .. QL127 - specify numerically
Set-Point Command, Floating-Point Value	**C_SE_FV C_SE_NC_1 C_SE_TC_1**	Float	Write	write mode (required if any options are given): D/Direct - one-step write SE/SelectExecute - two-step write Timestamp mode (only if not specified via type, default NTS): TS/Timestamp - send timestamp (use C_xx_Tx_1 opcode) NTS/NoTimestamp - don't send timestamp (use C_xx_Nx_1 opcode) Write Qualifier (default 0 - unspecified): QL0 .. QL127 - specify numerically

[1]First line selects the opcode based on the presence/absence of the Timestamp option, other lines use the specific opcode.
[2]The data type to write.
[3]For write, multiple options may be given, depending on the opcode, minimum is D or SE. If none are present, write a structure that gives the options.

<Data type> (command direction, special commands)

	Syntax[1]	Type[2]	Access Rights	Options[3]
Interrogation Command	**C_IC C_IC_NA_1**	Bool	Write	interrogation mode: G/Global - global interrogation INRO1 .. INRO16 - group interrogation QOI0 .. QOI255 - specify numerically
Counter Interrogation Command	**C_CI C_CI_NA_1**	Bool	Write	interrogation mode: G/Global - global interrogation REQCO1 .. REQCO4 - group interrogation RQT0 .. RQT63 - specify numerically counter freeze mode (default 0 - read only): F/Freeze - freeze counters R/Reset - reset counters FR/FreezeAndReset - freeze and reset counters
Clock Synchronication Command	**C_CS C_CS_NA_1**	Bool	Write	

Test Command	**C_TS** **C_TS_NA_1** **C_TS_TA_1**	Bool	Write	write mode (required if any options are given): D/Direct - one-step write Timestamp mode (only if not specified via type, default NTS): TS/Timestamp - send timestamp (use C_xx_Tx_1 opcode) NTS/NoTimestamp - don't send timestamp (use C_xx_Nx_1 opcode)

[1]First line selects the opcode based on the presence/absence of the Timestamp option, other lines use the specific opcode.
[2]The data type to write. For commands, any written value will execute the command.
[3]For write, multiple options may be given, depending on the opcode, minimum is D or SE. If none are present, write a structure that gives the options.
[4]For C_CS, write a Bool value to Item "C_CS".

Examples

M_SP.1000.SPI - Monitor the Single-Point value at IOA 1000; returns a Bool value. Monitors opcodes M_SP_NA_1, M_SP_TA_1, M_SP_TB_1
M_SP_NA_1.1000.SPI - Monitor the Single-Point value at IOA 1000; returns a Bool value. Monitors opcode M_SP_NA_1 only
M_SP.1000.IV - Monitor the Invalid bit of Single-Point value 1000
M_SP.1000.Timestamp - Monitor the Timestamp of Single-Point value 1000
M_SP.1000 - Monitor the Single-Point value 1000; returns a structure SIQ_TS that contains the value, timestamp and all quality bits
C_SC.1000.D - Write a Bool value to execute a Single Point Command to IOA 1000 (uses C_SC_NA_1)
C_SC.1000.D.TS - Write a Bool value to execute a Single Point Command to IOA 1000, include a timestamp (uses C_SC_TA_1)
C_SC.1000.SE - Write a Bool value to execute a Single Point Command to IOA 1000 (uses C_SC_NA_1 and the Select-and-Execute semantic)
C_SC.1000.D.SP - Write a Bool value to execute a Single Point Command to IOA 1000 (uses C_SC_NA_1 and the Short-Pulse qualifier)
C_SC.1000 - Write a structure SCO to execute a Single Point Command to IOA 1000. Value, opcode and write options are specified by the structure contents.
C_IC.G - Write a Bool value to execute a Global Interrogation Command

11.8 Send/Receive Item Syntax

Send/Receive frames will be handled as byte arrays.
The name is raw.
The frames will be sent and received as one piece. In connections without a protocol like PLC Header, RFC1006 or OSI/H1 it can not be guaranteed that the frames will not be concentrated or splitted in pieces.
Ideally RAW data will be handled by PLC Engine Collect. It contains lot of RAW data access functions as search, change and more.

11.9 Browsing Tree

The address space can be browsed online.

The address space on OPC Classic and OPCPipe starts in the root.
Example: <**PLC**>.**flag0**. Or "**System**.**Topics**.<**PLC**>.**Status**

On OPC UA the root starts with "**Objects**". Under this topic there are multiple elements depending on the OPC configuration. The most important element is the element "**Topics**". All elements described on this manual are under this item.
Example: **Objects**.**Topics**.<**PLC**>.**flag0**. Or "**Objects**.**Topics**.**System**.**Topics**.<**PLC**>.**Status**

If browsing of items is not used registering an item needs a prefix. Which prefix your OPC UA server requires can be found in the documentation of your OPC UA server. No standard for a prefix exist, some servers are using names, others UUID or numbers, but it can be a binary field also. There exist no limitation in length. It is not guaranteed that the prefix will be the same after reconnect to the UA server.

The OPC UA server needs if browsing will not be used "**#ns=1;s=**" in front of an item. Leave the "**Objects.Topics**" away.

Example the OPC UA server: Read the flag word zero from a Siemens S7 300, the OPC connection is named "OPC", the controller connection is named "s7": "**OPC.#ns=1;s=s7.mw0"**

11.10 System Tree

The **System tree** contains the s**pecial items** the server supports. They allow accessing internal variables, their status and a lot more.
The system tree can be added or disabled using the configuration with a separate **System** connection.

The system tree **"System"** lies in
- **OPC-DA and OpcPipe**: directly in the root.
- **OPC-UA** under **"Objects.Topic.System"**

With writing to the special item **"AddTopic"** it is possible adding more connections to PLC without using the configuration software. Additionally, the server can be configured via a "Config" connection.

Examples:
- SPS connection status: If a connection named "s7" is configured the connection status lies under **System**.**Topics**.s7.**Status (**over OPC UA **server.System**.**Topics**.s7.**Status)**
  The PLC operation mode lies under
  System.**Topics**.s7.**PlcDetails**.**PlcMode**
- Add a new connection with writing to **System**.**AddTopic**.
  "s7.tcp://192.168.2.200:102?name=\"my name \",typ=client,ownTSAP=0101,destinationTSAP=0302,slave=9,start1=1,WriteAllowed=1"

Elements in the root
- OPCPipe and OPC DA the start floor lies in the root floor.
- Under OPC-UA the start floor lies under "**Objects.Topics**"

Topic name	Description
System	System data as the list of topics, version information
Memory	Temporary variables and structures which relays on the system memory.
Config	Server Internal variables with which you can configure the OPC server or the PlcEngine. The individual variables have comments that describe the functionality. They are not described in detail in the help.
OPC UA System Items	connection for OPC UA special functions
<**user defined**>	User defined topics created with the configuration software or with the browser or over writing to **System**.**AddTopic**

System

Topic name	Description
Topics	The predefined item "**System**.**Topics**" is containing the list of configured connections to controllers. The elements allows accessing information about stati of the OPC server itself and the connections to the controllers. Additional details from the controllers can be read as the name of the controller, its operation mode and more.
AddTopic	Adds connections to controllers. With writing of a string in the defined syntax connections will be added. The syntax is described in AddTopic.
Licenses	information about the current licenses
Versions	Versions of the software and its parts from the OPC server or the PlcEngine.
Internal	Please do not use this. It is made for special applications.
Platform	allows tags for different Platform things, eg: inputs and outputs on Phytec Regor
Status	Extended Server status
CurrentTime **...** and other tags	the current time and time information is available for further processing.

Values for System.Topics.<topicName>

Topic name	Description
Status	Brings the status of the connection. 0-> ok 3 -> no connection 5 -> access denied 6 -> wait for data 7 -> wait for send ready 23 -> The communication partner denies the port. No program did open it 26 -> ARP can not resolve the address. Mostly the partner is switched off or disconnected 29 -> Suspended. No OPC client or logic table used one item. 30 -> Osi reset by peer. Mostly a TSAP is wrong 31 -> The port is already in use, mostly by another program 32 -> The domain name can not be resolved. The domain server is unreachable, or the name was not found. 33 -> The network is unreachable. Mostly the computers network interfaces are

	configured badly.
ReadCount	Number of handled synchronous read jobs. This variable can be written.
WriteCount	Number of handled write jobs. This variable can be written.
ActiveCount	Number of active items.
ServerCycle	Number of cycles with changed PLC data. This variable can be written.
Redundancy	This will exist in connections configured for redundancy only. Details here.
AddVariable	This element only exists in the Memory topic. Adds a variable in the memory variables. This element is write only. The syntax is <variable name>=<variable type><[><array size><]><;comment> Supported variable types are: u8, u16, u32, u64. i8, i16, i32, i64, f32, f64, bit, string. Example: MyVariable=u16[10];Array of UINT 16 with 10 elements If a variable name contains dots all named before the last dot are nodes. Hint: The symbols created with **AddVariable** will be deleted if the OPC or PLC Engine Collect software is stopped.
DeleteVariable	Deletes a variable which was created with **AddVariable**. The name of the variable need to be given. This element is write only. Each variable and node need to be deleted separately. Hint: If a variable will be deleted which was not created with **AddVariable** the access will be denied.
AddSymbol	The element only exists in controller topics which does not support symbols in the controller itself. Mostly this are Siemens S7 300/400 controllers, Modbus devices or Mitsubishi controllers. Adds a symbol. The item syntax of the desired controller type of the topic will be used. SymbolName=<ItemSyntax><;comment> Example for a S7 300: MyFlag=mw0;Flag word zero The symbol name can contain dots. So multiple floors can be created. Hint: The symbols created with **AddSymbol** will be deleted if the OPC or PLC Engine Collect software is stopped.
DeleteSymbol	Deletes a with **AddSymbol** created symbol.
AddStructure	Not available today
DeleteStructure	Not available today
DeleteTopic	If a connection has been added dynamically, this item is visible. If this is written to, this dynamically created connection is deleted.

Values for System.Topics.<topicName>.PlcDetail

If the information can not be readied from the controller the quality becomes bad (sensor failed).

Topic name	Description
PlcMode	0 -> PLC in Stop, 1 -> PLC in RUN/STOP, 2 -> PLC in RUN.
KeySwitch	0 -> Key switch in position Stop, 1 -> Run/Prog, 2 -> Run, 3 -> Mres
PlcName	The name given to the controller. Not all controllers does contain a name.
PlcType	The PLC type the manufacturer did give to it. Not all controllers does contain this.
OrderNumber	The order number of the controller. Not all controllers does contain this.

Force	Information if forces will exist in the controllers program code. Some p,c deliver the number of forces here.
Battery	Status of the battery. 0 -> Ok, 1 -> empty, 2 -> no battery, 3 -> not supported

Not all controller types and controller firmware versions are supporting all items under **PlcDetails**. So the **OrderNumber** will be available in Siemens S7 only. The details can be found in the controllers system manuals.

Values for System.AddTopic

Warning: System.AddTopic is deprecated. Pleas use the configuration topic which has much more functionality.

Topics can be created over this item.

Important: With **AddTopic** created connections are existing non permanent. On power off they will be lost. They must be created again after power on.

A topic created with **System**.**Topics**.<created topic> can be deleted with **System**.**Topics**.<created topic>**DeleteTopic**.

<table>
<tr><th>Topic name</th><th>Description</th></tr>
<tr><td>Name</td><td><Topic Name>. The name of the created connection. Please note the rules for topic names which will existing from various OPC clients.</td></tr>
<tr><td>PlcType</td><td>Valid are:
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>s7</td><td>Siemens S7-200, 300, 400 and compatible as Speed7. Can be used for accessing the controller over MPI adapter from Hilscher, Process Informatik, IBH Softec, Softing, Helmholz.</td></tr>
<tr><td>tia</td><td>Siemens S7 1200 and 1500. This will support the optimized data blocks.</td></tr>
<tr><td>s5</td><td>Siemens S5 with network CP All racks are supported 135, 155, 188. All CPU types.</td></tr>
<tr><td>compactlogics</td><td>Rockwell Control Logix and Compact Logix.</td></tr>
<tr><td>slc</td><td>Rockwell SLC family.</td></tr>
<tr><td>modbus</td><td>Modbus TCP and compatible systems as Wago, Beckhoff, Modicon, Omron and more..</td></tr>
<tr><td>melsecQ</td><td>Mitsubishi Melsec family.</td></tr>
<tr><td>raw</td><td>Raw data.</td></tr>
</table></td></tr>
<tr><td>NetworkProtocol</td><td>Valid are:
<table>
<tr><th>Value</th><th>Alternative</th><th>Description</th></tr>
<tr><td>ip</td><td>tcp</td><td>TCP/IP TCP over IpV4 (192.168.1.1) or IpV6 <235b:34aa::0001:0030) or domain names</td></tr>
<tr><td>udp</td><td></td><td>TCP/IP UDP over IpV4 (192.168.1.1) or IpV6 <235b:34aa::0001:0030) or domain names</td></tr>
<tr><td>h1</td><td>osi</td><td>OSI/H1 over MAC Addresses (080006010001) or (08:00:06:01:00:01)</td></tr>
</table></td></tr>
<tr><td>NetworkAddress</td><td>IpV4 (192.168.1.1) or
IpV6 (235b:34aa::0001:0030) or
domain names (s7.mydomain.com)</td></tr>
</table>

Port	1 .. 65534. This is needed for IP connections only.
ConnectType	Valid are: • Server • Client
OwnTSAP	in hexadecimal. This is valid on OSI/H1 or Port 102 (RFC1006) connections only. (0101)
DestinationTSAP	in hexadecimal. This is valid on OSI/H1 or Port 102 (RFC1006) connections only. (0302)
Slave	In Modbus connections this is the slave or node address. In Rockwell connections this is the CPU number. All other connections do not use this value.
Start1	I Modbus connection this defines the assumed first register. False means the first register is zero otherwise the register count starts with one.
RockwellRouting	This is for reaching specific parts of the controller or access controllers in underlaying connections. This element requires text. Three options exists: • No routing (empty path). This is for reaching CPU Damit with logical integrated network modules in the PLC. Example are Micro800 or L16. • Simple routing. This is for reaching a specific CPU in multi CPU controllers.. **CPU(0)** will access CPU 0. In controllers with logical separate network adapters this is required. The L35E-models will need this. • Extended routing. This is for accessing controllers over underlaying netwoks as DH+ **Port(1;2)** goes over port 1 to the destination 2. **PortEx(4;192.168.2.212)** accesses controllers in an underlaying ip network. In this example the network adapter has the port number 4, the next controller is reached over 192.168.2.212. **DHPlus(1;0;0;15)** reaches a controller over a DHplus on channel A(1), Source Link 0 the plc destination Link 0 Node 15. An example for a longer routing path:**Port(1;0)-Port(2;1)-PortEx(4;192.168.2.212)** The port and slot numbers are taken from the graphical configuration or RsWho. Connection routing paths can be very long if this runs over multiple underlaying busses. Some controllers as the Micro800 models can not have a routing path if they need to be accessed. In case of route over a Micro800 model to other controllers the path is required.
RockwellCharset	This can be: **UTF8** (default) or **ANSI**.
RockwellHeader	Allowed is **ENCAP** (default) or **CSP**.
RockwellProtocol	Allowed is **CIP** (default) or **PCCC**.
RockwellShowHidden	Possible are **None** (default), **Hidden**, **Underscore**, **All**.
Create	Writing a value not zero creates the connection. This variable is write only.
CreateStatus	0 -> Topic is created. This variable is read only. 1 -> Invalid connection parameters. Mostly this will happen on too long TSAP values. 5 -> Invalid adapter. The given adapter is not available. Mostly on variable

	adapters (USB to Ethernet) which was disconnected. 17 -> No free memory available. 101 -> not supported. This will happen on plc protocols which are not available. 1319 -> not supported. Mostly this happens on unsupported network protocols as IpV6 which is not configured. 1324 -> License limit. The maximum of connections are created already. 1325 -> not licensed.
AddTopic	can be used instead of the variables **Name**, **PlcType**, **NetworkProtocol**, **NetworkAddress**, **Port**, **ConnectType**, **OwnTSAP**, **DestinationTSAP** for creation of connections. This variable is write only. The syntax on an example of an s7 connection with the name "s7" "s7.tcp://192.168.2.200:102?name=\"my name \",typ=client,ownTSAP=0101,destinationTSAP=0302" The element **Create** is not used in this case.

Values for System.Versions

A version number of zero says that the component is not loaded.

Topic name	Description
PlcEngine	Version number of the core of the PlcEngine or OPC server.
Wmk	Version number of the Wmk library. This library contains general functions.
IpLib	Version number of the TCPIP library. This library contains all IP socket functions.
H1	Version number of the Osi/H1 driver.
OpcDA	Version number of the OPC DA library. This library contains Classic OPC and is available for Windows only.
OpcUA	Version number of the OPC UA library. This library contains OPC UA.
OpcPipe	Version number of the OpcPipe library. This library contains the item management.
PlcLib	Version number of the PLC protocol library. This library contains all PLC protocols and its management.
ItemSyntax	Version number of the item syntax library. The library converts symbols in elements.
ConfigSubs	Version number of the configuration general functions. The library handles the configuration of the core.
..	Depending of the software or the release more software component version numbers are shown.

Values for System.Platform

Platform offers variables which are different from platform to platform. Mostly this are I/O variables.

Variables for the Regor devices from the manufacturer Phytec:

Topicname	Beschreibung
Digital1Dir	Sets the function of pin 1. "true" defines it as output, "false" as input.
Digital1In	If pin 1 is an input "true" signals a closed connection between pin 1 and ground.
Digital1Out	If pin 1 is an output "true" signals a closed switch between pin 1 and ground.
	Digital2Dir, Digital2In, Digital2Out are for pin 2. Same to pin 3 and 4.

ErrorLED	"true" signals the LED on. Writing "false" switches it off, "true" on again.
RunStopLED	"true" signals the LED on. Writing "false" switches it off, "true" on again.

Values for System.Licenses

The licenses and their details are here.

Topic name	Description
BaseLicense	Order number of the product. The comment of the item contains the product name.
BaseLicenseExpires	End date for time limited and test licenses in the format yyyy-mm-dd hh:mm for example: 2017-12-31 23:59
BaseLicenseState	Bit coded value: Bit 0: A permanent license is present and it is OK. Bit 1: The dongle is recognized and valid. Bit 2: The dongle was recognized in the past, but now it can not be found. Bit 3: Licensing via software confirm code. Bit 4: A software confirm code is detected and OK. Bit 5: A time limited confirm code has been detected. The expiration time is shown in item BaseLicenseExpires. Bit 6: A time limited confirm code has been detected and expired. Bit 7: The hardware (network card) to which the license was attached is no longer present. Bit 8: No license is recognized, it is a test version. The expiration time is shown in item BaseLicenseExpires. Bit 9: The test phase has expired, a restart is required. Bit 10: Free license product is set, no licensing required
AddonLicenseX	Order number of an addon. X is a consecutive number and starts with 1. If multiple addons exist more AddonLicenseX elements will exist.The comment of the item contains the addon product name.

Values for System.Status

Provides data points for extended status of the entire server.

Item name	Description
StructChangeCounter **EnumChangeCounter**	These items are incremented when the internal structure or enum lists are modified. This is not connection-specific, but a single counter for the entire server. If automatic import is switched off, the counters increment when a structure/enum is imported manually. If automatic import is switched on, the counters increment when a new structure/enum is read from the PLC.

11.11 Configuration Tree

With the Configuration Topic all configurations can be done es the graphical configuration does. The topic can be created with the graphical configuration. If the software is installed on a new system the installer asks for the creation of this topic.

All details for the configuration is available in the Connection subtree.
It is separated to

Base	Name of the connection, rights, time details
OPC	All for OPC server and clients
Transport	IP Addresses, TSAPs, OSI
Operations	Handling for read and write connections, assistances

und viele Weitere die sich selbst erklären

The configuration topic knows some dynamic handling.

Example 1: Read the details of a connection.
Write the connection name to Configuration.Connection.Base.ConnectionName and the group name in Configuration.Connection.Base.ConnectionGroup.
After this write true to Configuration.Connection.Operations.Read.The job starts the content of the variable Configuration.Connection.Base.Status will change to 6 (running). If the variable changes zu zero the list of connections is read. In Configuration.Connection.Base.ListResults the results will be delivered. In Configuration.Connection.Base.ListResultMaxLen the number of connections will return. In case of an error the variable Configuration.Connection.Base.ListStatus is containing the error code, Configuration.Connection.Base.ListErrorText is containing an error text for this.
If this is handled over OPC UA so Configuration.Connection.Base.ListResult is an array with the desired length.

Example 2: Read the list of network adapters.
Write the variable Configuration.Operations.Adapter.Start to true. The variable Configuration.Operations.Assistants.Adapter.Status will return 6 (running). If the job finishes without an error the variable will become zero. In Configuration.Operations.Assistants.Adapter.Result the recognized network adapters of the station will return..In case of an error a human readable text will be in Configuration.Operations.Assistants.Adapter.ErrorText, in Configuration.Operations.Assistants.Adapter.Status the corresponding error code is delivered.

Example 3: Read the list of available Rockwell PLC.
Please search the network adapters as in example 2. Copy the needed adapter to Configuration.Operations.Assistants.RockwellHost.Configuration.Adapter. Write true to Configuration.Operations.Assistants.RockwellHost.Start. The job will start, The variable Configuration.Operations.Assistants.RockwellHostStatus will change to 6 (running). If the job will finish (in Configuration.Operations.Assistants.RockwellHost.Status zero comes) the found Rockwell controllers will be in Configuration.Operations.Assistants.RockwellHost.Result die Liste der gefundenen Steuerungen. Searching PLC will need time, some PLC will respond very late. So the variable Configuration.Operations.Assistants.RockwellHost.Start need set to true multiple times.
If this is handled over OPC UA so Configuration.Operations.Assistants.RockwellHost.Result is an array with the desired length.

Please import the structures from the Configuration topic if your OPC client can handle structures.

11.12 Redundancy

Connections with redundancy only provide these values.

Element name	Description
CurrentMaster	Reading brings the current master. 1 is the first connection. Writing switches to the given connection. Hint:

	Writing an invalid value does nothing. In **dynamic master** this switches to the given connection. If the new connection will work stable it will remain. Otherwise an other connection is choose. In **static master** it will be switched back after one minute if the master works stable. If the slave connection breaks it will be switched to the master immediately it it will work stable. Otherwise an outer connection is choose.
Connection<n>	This variable will exist in the number of redundancy connections. . Each Connection<n> Variable will contain **Status** **ServerCycle** **ReadCount** **WriteCount**. These variables are working in the same manner of in connections without redundancy. Details here.

11.13 Bit mask

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 or
- db5.w0#2,8 or
- db5.w0#2#8

Examples

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

11.14 Arrays

The word array means a series of equal elements (field, row, data area). An array combines several units of one data type in a field. To create an array, the length of the array is added to the standard syntax, separated by a period.

Arrays are impossible in:

- with all suffixes that are string to the client:
  - ISO

11.15 Suffixes

Using a suffix, a value can be displayed in a different format.

Suffixes	**Syntax**	Used for	Area	Data type	Variant Data Type	Comment
Date and Time in String Format	**ISO**	DT	1990-1-1-00:00:00.000 bis 2098-12-31-24:59:59.999*	STRING	VT_BSTR	The suffix DT is used to show the data saved in the PLC as a combined data type DATE_AND_TIME and is transfered as a string. The data type DATE_AND_TIME has 8 bytes (64 bits) in the PLC. The year, the month, the day, the hour, the minutes, the seconds and the milliseconds are included. Remember to use the correct separators (hyphen, colon and period)!
BCD	**BCD**	Byte Word DWord QWord	Byte: 0 to 99 Word: 0 to 999 DWord: 0 to 9999999 QWord: 0 to 9999999999999999	VT_I1 VT_I2 VT_I4 VT_I8	VT_I1 VT_I2 VT_I4 VT_I8	With the BCD suffix, the data stored in the PLC is represented as an unsigned, binary-encoded value. For example, the decimal value “65535” will be represented as “9999”.
ASCII to Hex	**KA**	String	HEX: 0 to 9, A to F	STRING	VT_BSTR	With the KA suffix, the data stored in the PLC is represented as HEX characters.
Signed	**KF**	Byte Word DWord QWord	Byte: -128 to 127 Word: -32768 to 32767 DWord: -2147483648 to 2147483647 QWord −9.223.372.036.854.775.808 to 9.223.372.036.854.775.807	CHAR SHORT LONG LLONG	VT_I1 VT_I2 VT_I4 VT_I8	With the KF suffix, the data stored in the PLC is represented as a signed fixed point number.
S5-KG	**KG**	DWord	0.1469368E-38 to 0.1701412E39 Attention: The KG area in the PLC is larger	REAL	VT_R4	With the KG suffix, the data stored in the PLC is treated as a 4-byte floating point number.

			than in the PC!			
S5-KT-Format	**KT**	Word	000.0 to 999.3	STRING	VT_BSTR	With the KT suffix, the data stored in the PLC is represented as a 2-byte time constant. The time base is included in addition to the time value. The value range is from 000.0 to 999.3
S5 Time	**S5T**	Word	0ms to 2h46m30	STRING	VT_BSTR	With the S5T suffix, the data stored in the PLC is represented as S5TIME (SIMATIC time). The S5TIME data type occupies one 16-bit word and is the product of the time value and the time interval (time base). The time duration is given in hours, minutes, seconds and milliseconds. The BCD number format is used for internal representation. The value range is from 0 ms to 2h46m30s. The smallest value is 10ms.
Date	**D**	Word	1990-01-01 bis 2168-12-31	STRING	VT_BSTR	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. • Anything but numbers can be used as a separator (e.g., 89/09/17). Any number of separators can be used.
Time	**T**	DWord	-24d20h31m23s648ms to 24d20h31m23s647ms	STRING	VT_BSTR	With the T suffix, the data stored in the PLC is represented as the TIME data type. The TIME data type occupies one double word (32 bits). The representation contains the days (d), hours (h), minutes (m), seconds (s) and milliseconds (ms). Milliseconds can be omitted. The value range is from -2147483648 (-24d20h31m23s648ms) to 2147483647 (24d20h31m23s647ms)

Time of Day	**TOD**	DWord, DInt	0:0:0.0 to 23:59:59.999	STRING	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.
TimeReal	**TR**	Word	0.01 to 9990.0	REAL	VT_R4	With the TR suffix, the data stored in the PLC is represented as the TIME REAL data type. The value range is from 0.01 to 9990.0 (s)
Swap bytes	**SWAP**	String Word, Int DWord, DInt QWord, QInt Real, Double	Exchange high byte and low byte	original data type	original data type	High byte and low byte of the tags stored in the PLC are exchanged. Important for strings in Modbus PLCs.
Comment	**_comment**	all items	gives the comment of the item	STRING	VT_BSTR	Only for OPC DA returns the comment of the item. It is write protected.

* ms can be omitted.

11.16 Item Syntax neutral

Modern PLCs don't need an item syntax. All symbols are available on the controller. They will be read online. This applies to all Rockwell ControlLogic and CompactLogix PLCs, all Siemens S7 1500 and 1200 models, and BACnet devices.
Some subsystems as MQTT do not know variables with defined data types. For these the neutral item syntax is offered also.
For MQTT the usage of this item syntax need to be configured in the connection parameters.

A tag is fully defined with the tag name, the data type is defined by the PLC. Normally, nothing needs to be added.

However, there are some exceptions:

- Addressing a single element of an array:
  the zero-based index is specified in square brackets after the item name.

<Tagname>[<index>]

If you need offline symbols, you can add them. This may be required:
- to have symbols without a controller online,
- to rename symbols
- for simulation connections.

The "item syntax" is the real symbol name in the controller.

Because no data type is available without online access to the PLC an optional name extension exists for specifying the type.
The rule is:

<Symbolname>{{<Datatype>,<Endian>}}

or for arrays:

<Symbolname>{{<Datatype>,<Arraylength>,<Endian>}}

The following data types are allowed (the spaces are important):
Bool
Int 8
Int 16
Int 32
Int 64
Uint 8
Uint 16
Uint 32
Uint 64
Float
Double
Timestamp
String

These is possible for endian:
LittleEndian, some times known as INTEL format
BigEndian, known as Motorola format

If non endian is given the format of the locally working cpu is used.

The symbol name (without the {{ }} name extension) must be present in the controller (except for simulation connections).