Konfiguration Plc Engine

Telefon: 0911/98037354, Internet: http://tanindustrie.de,
Email: info@tanindustrie.de

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.*

Konfiguration Plc Engine

Created: Februar 2016 in Nuremberg, Germany

Tani GmbH
Freiligrathstrasse 12
D-90482 Nuremberg
Internet: http://tanindustrie.de
Email: info@tanindustrie.de
Tel.: +49 911 980 373 54
HRB: Amtsgericht Nürnberg 29562
USt-Id: 214/140/619.

Table of contents

1 Help overview

2 Tani OPC Server and PlcEngine - General

This chapter provides general information about the Tani configuration:

2.1 Übersicht

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 ISO (H1).

Tani PLC Engine
The Tani PLC Engine 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 on other stations also.

Services
The runtime of OPC server and PLC Engine 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 Lizenzbedingungen

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 Nuremberg
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-Client

Im Lieferumfang des OPC-Servers sind OPC Test-Clients enthalten. Einer dient dem OPC UA Test, der Andere ist für Microsoft Classic OPC DCOM basierend. Diese OPC-Client werden während der Installation des OPC-Servers mit installiert. Beide Client ist sehr einfach zu bedienen und ermöglicht ein Testen des OPC-Servers.

Verfügbare OPC-Server anzeigen

Der OPC-Client ermittelt die OPC-Server, die auf dem PC installiert sind und bietet sie an im Menü über Server > Connect. Der Client durchsucht die Registry des lokalen Computers und generiert eine Liste der verfügbaren OPC-Server.

Server und Client verbinden

Wird der Ethernet OPC-Server „Tani OPC Server“ ausgewählt und mit OK bestätigt, so wird dieser automatisch gestartet. Kommt eine Fehlermeldung, kann es folgende Ursachen haben:
die OPC Core Components sind nicht richtig installiert ==> bitte neu installieren
bei den DCOM-Einstellungen sind die Startberechtigungen nicht zugeteilt, bitte den Zugriff erlauben.

Gruppen hinzufügen

Um eine neue OPC-Gruppe anzulegen, wird im Menü Group > Add Group gewählt. Der Gruppenname ist beliebig, die Update Rate (msec) legt fest, mit welcher maximalen Datenrate die einzelnen Datenpunkte neue Werte bekommen.

Items hinzufügen

Um Items zu der Gruppe hinzuzufügen, wird die Gruppe markiert und im Menü über Item > Add Items angewählt. Für jedes Item muss eine Item ID und der Access Path angegeben werden. Zusätzlich kann der angeforderte Datentyp und der Aktiv-Status ausgewählt werden.

Der Access Path muss exakt dem Namen entsprechen, der beim Anlegen des Access Path in PLC Engine verwendet wurde. Bei einigen OPC-Clients ist kein eigenes Dialogfenster für den Access Path vorhanden. Bei diesen Clients wird der Access Path und das Item durch einen Punkt getrennt.
<Access Path Name>.Item-ID
z.B.
PLC1.DB5.W0

Im Fenster links unten, werden alle symbolischen Namen angezeigt, die anstelle der Item-IDs verwende werden können. Diese stellt der OPC-Server über das Browsing Interface zur Verfügung.
siehe auch
Symbole bearbeiten
S7 Symbolimport

Wenn das Item angegeben ist, kann mit Validate geprüft werden, ob der Server das Item akzeptiert.

Mit OK wird das Item beim Server angemeldet. Im Hauptfenster wird die Variable angezeigt mit Wert Quality und Updates.

3 Basics

This chapter covers the following topics:

- NetConConfiguration
- PLC Connections
- Logger
- ARP
- H1 Driver

3.1 Tani Konfiguration

Die grafische Oberfläche (GUI) zum Konfigurieren und Diagnostizieren der Engine Geräte. Die GUI verbindet sich mit dem Gerät über TCP/IP (Port 2468).

Programmstart

Konfiguration lässt sich über folgenden Weg starten:

- Über Start - (Alle)Programme - Plc Engine - Tani Konfiguration
  Das Programm startet mit dem Startfenster und erlaubt eine Konfiguration über TCP/IP oder eine Konfiguration über H1 (wenn der H1-Protokolltreiber installiert ist). Über diesen Weg wird dann das Stationsfenster Teilnehmer auswählen geöffnet und die Verbindung kann zu dem gewünschten Gerät aufgebaut werden. Die Engine-Geräte können auch offline konfiguriert werden und später in das Gerät geladen werden. Üblicherweise wird jedoch online konfiguriert.

Kennwort

Das Verändern von Parametern und Einstellungen kann durch Setzen eines Kennwortes geschützt werden. Das geschieht im Menü Station - Stationskennwort. Ist ein Kennwort gesetzt, so wird der Anwender beim Speichern oder Verändern von Parametern nach dem Kennwort gefragt. Lesezugriff besteht auch ohne Kennwort Eingabe. Ein Beobachten und Diagnostizieren ist auch ohne einloggen möglich.
Hat man das Kennwort vergessen, so hilft nur ein Übertragen der Parameter von der externer SD Karte. Auf der SD Karte muss sich im Root-Verzeichnis die Datei \netfile.net befinden. Entweder eine mit bekanntem Kennwort, oder eine Leere Datei. Dann die Karte in das Gerät stecken und die Spannungsversorgung unterbrechen und das Gerät dann wieder einschalten. Beim Einschalten überschreiben die Parameter von der Karte die Parameter im Gerät. Einen anderen Weg gibt es nicht.

Benutzer

Ein jedes Gerät erlaubt es, dass mehrere Konfiguration zugreifen. Um ein gegenseitiges Beeinflussen zu unterbinden ist kann der Zugriff über ein Benutzen und Kennworte Verfahren geregelt werden.
Es können mehrere Benutzer angelegt werden.
Ist mindestens ein Benutzer angelegt so ist zum Zugriff immer ein Benutzername notwendig.
Die Benutzer können unterschiedliche Rechte haben. Mindestens ein Nutzer muss die vollen Rechte haben.

Bedienung und Konfiguration

Im Abschnitt Bedienung und Konfiguration wird die grafische Oberfläche der Konfiguration detailliert beschrieben.

Logger

In Konfiguration kann der Logger auf dem Gerät konfiguriert werden. Der Logger Inhalt wird im Fenster Logger angezeigt.

3.2 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 Settings dialog box.
To specify whether to create the log in the RAM only or to also save it to a file, click Settings in the Logger dialog box.

3.3 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

3.4 OSI/H1 driver

To configure the devices via H1, you need the H1 driver (Tani H1 ISO 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.

4 Connections

Das Kapitel beschreibt folgende Themen:

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:

The connection should be parameterized on both sides of the communication (one Fetch/Write active connection on the device side AND one Fetch/Write passive connection in the S7 CP)

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.

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.

4.4 ControlLogix/CompactLogix General

The CLX protocol is used for the communication with ControlLogiX / Compact Logix PLCs from Rockwell Automation.
The CLX protocol only works in conjunction with TCP/IP.

Read (Fetch active) and Write (Write active):
- When communicating with CLX 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 parameterize 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 CLX.
Alias browsing:
Once the CLX connection has been established, symbol information will be read from the PLC and displayed for selection in the corresponding dialog boxes.

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 MELSEC-Q General

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

Read (Fetch active) and Write (Write active):
- 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 parameterized 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.

4.7 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.

4.8 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 on the 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.

OPC-UA

5 General usage

This chapter covers the following topics:

- How to Proceed
- Basic Configuration
- Licensing

5.1 Vorgehensweise

The devices are very easy to parameterize in just a few steps. To set up communication, proceed as follows:

Basic Configuration

To be able to communicate with the 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 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.

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 Grundkonfiguration

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 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 Lizenzierung

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.

5.4 S7 Symbolimport

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.

5.5 Online-Diagnose

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.

5.6 Symbole bearbeiten

Der Dialog erlaubt ein Bearbeiten der Symbole.

Symbol Name
Der Name wird im Browsing angeboten. Über diesen wird der Datenpunkt angesprochen.

Itemsyntax
in dem Feld wird die korrekte Item Syntax für die Adresse eingegeben.
Siehe Item Syntax

Symbolkommentar
Der Symbolkommentar ist optional. Er wird im Browsing als Eigenschaft angezeigt.

5.7 User Settings

In dem Dialog können Benutzer mit verschiedenen Rechen angelegt werden.

Ist kein Benutzer angelegt so wird nicht nach einem Benutzer und Kennwort gefragt, alle Benutzer haben Vollzugriff.

Werden Benutzer angelegt so muss mindestens ein Benutzer Vollzugriff haben, die Konfiguration prüft das.

Diese Rechte Optionen gibt es
- Systemzugriff
- FTP Server Zugriff
- Webserver Zugriff
- SSH Zugriff

Jeder Zugriff kann Schreiben freischalten.

Sind Benutzer angelegt so ist ohne Benutzen/Kennwort kein Zugriff möglich.

Ist das Kennwort verlorengegangen so ist der Zugriff nur so möglich: Zugriff zum Dateisystem erlangen mit Login am Betriebssystem. Dann die Datei /etc/Tani/userdb.sqldat (Linux) oder $WINDIR\Public \Documents\Tani\userdb.sqldat (Windows) löschen. Danach den Tani Konfigurations Serverdienst neu starten.

6 Windows

Das Programm besteht im Wesentlichen aus den folgenden Hauptfenstern:

- Tani Konfiguration
- Hauptfenster der Konfiguration
- Verbindungsliste
- Logiktabellen
- Online-Diagnose
- Logger

Weiter Funktionen werden in Dialogen angezeigt, die in dem Kapitel Verbindung beschrieben sind.

6.1 Tani configuration

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.

6.1.1 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
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
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**.

Scan for stations
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.

Execute station scan on start
Automatically starts the station scan when you open the dialog box. Since this may take a long time if many stations are online, you can disable this function.

Use Timeout
It is recommended to enable this option only if the default connection timeout is too short or the transmission is very slow. By default, this function is disabled.

Timeout
You can use the connection timeout option to change the default timeout (3 seconds) specifically for this connection. This is useful e.g. for Internet connections for which heartbeat monitoring (sending of life time acks) has been disabled. This also increases the response time in case of connection problems (such as a broken cable).

6.1.2 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 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.

6.1.3 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
· 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.

6.2 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.

6.3 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**

6.4 Logic tables

Logic tables are containing functionality which is independent from OPC and other accesses. They are using variables and calculation for processing results.

Logic tables are used for calculations, for collecting data, processing data, for accessing databases, for sending e-mails and more.

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
- Triggers: Time, Value changes, Bits and Reset
- Calculations, Conversions and Rounding
- Structures collect the data und disassemble them
- Constants in Logic tables
- Subroutines and their Parameters
- Decisions und Data gates
- Comments
- Connections of the logic elements can be done with the mouse or touch screen.
- Online Diagnostics

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

6.5 Symbole bearbeiten

Der Dialog erlaubt ein Bearbeiten der Symbole.

Symbol Name
Der Name wird im Browsing angeboten. Über diesen wird der Datenpunkt angesprochen.

Itemsyntax
in dem Feld wird die korrekte Item Syntax für die Adresse eingegeben.
Siehe Item Syntax

Symbolkommentar
Der Symbolkommentar ist optional. Er wird im Browsing als Eigenschaft angezeigt.

6.6 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 zu zero.
Arrays are supported up to 100 elements each. If they are bigger they can not be written.
For writing uf arrays separate the elements by a **space**.

Attention: Writing affects the plant !

6.7 Logger

A toolbar in the top part of the window provides the following functions:
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.
Close
Closes the Logger dialog box. This does not affect the logging process. Logging continues even if the window is closed.

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.

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

6.7.1 Logger Einstellungen

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 Logic tables

Logic tables are containing functionality which is independent from OPC and other accesses. They are using variables and calculation for processing results.

Logic tables are used for calculations, for collecting data, processing data, for accessing databases, for sending e-mails and more.

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
- Triggers: Time, Value changes, Bits and Reset
- Calculations, Conversions and Rounding
- Structures collect the data und disassemble them
- Constants in Logic tables
- Subroutines and their Parameters
- Decisions und Data gates
- Comments
- Connections of the logic elements can be done with the mouse or touch screen.
- Online Diagnostics

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

7.1 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.

7.2 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.

7.3 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.

7.4 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

7.5 Constants

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

All supported data types can be used.

7.6 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.

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

7.7 Struktures

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.

7.8 Struktur into elements

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

7.9 Comments

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

7.10 Roundings

Rounding simplify the reading of reading of "bandy" 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.

7.11 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.

7.12 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.

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.

7.13 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.

7.14 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.

7.15 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.

7.16 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.

7.17 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.

7.18 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.

7.19 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.

7.20 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.

7.21 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**
- **Handle data with databases**
- **Subroutine : Limit test**

Hint:
The product OPC server only supports the "Data Optimizing".

7.21.1 OPC optimizings

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.

7.21.2 PLC data copy

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.

7.21.3 PLC data into database

Data will be read from controllers and written into a database. The opposite direction can be used also.

7.21.4 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.

7.22 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 zu zero.
Arrays are supported up to 100 elements each. If they are bigger they can not be written.
For writing uf arrays separate the elements by a **space**.

Attention: Writing affects the plant !

8 Connection

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.

- New connection
- Connection settings
- Edit a connection
- Copy a connection
- Delete a connection
- Change vonnection status

8. .1 Network protocol

Name der Verbindung
Hier kann der Verbindungsname nochmals geändert werden. Es sind alle Zeichen erlaubt, außer eckige Klammern.

Expertenmodus
Ist der Expertenmodus ausgeschaltet, so werden die möglichen Einstellungen, die nicht üblich sind ausgeblendet. Sollte nur dann eingeschaltet werden, wenn spezielle Einstellungen gemacht werden sollen. Ist üblicherweise aus.

Transportprotokoll
Hier wird das Transportprotokoll ausgewählt, über das im Ethernet-Netzwerk kommuniziert werden soll.

Transportprotokoll	
TCP/IP	das Transmission Control Protocol/Internet Protocol ist routing fähig und wird heutzutage zur Kommunikation zu SPSen und Feldgeräten verwendet. Für RFC 1006 (ISO on TCP) Kommunikation sollte TCP/IP gewählt werden.
H1	ISO-H1-Protokoll zum Verbinden an Siemens S5 oder S7 Steuerungen, die kein TCP/IP Protokoll unterstützen.

Applikationsprotokoll
Hier wird das Applikationsprotokoll ausgewählt.

Applikationsprotokoll	SPS
Senden/Empfangen	Kommunikation zu anderen SPSen oder Geräten, die keines der unten aufgeführten Protokolle beherrscht. Es werden reine Nutzdaten (Rohdaten) übertragen, ohne Protokoll-Information.
S5 AP	Kommunikation zu Siemens S5-CPs oder Engine-Geräten.
S7	S7-Protokoll zu Siemens S7-400, S7-300 oder S7-200 Steuerungen oder Engine-Geräten

Modbus TCP	Kommunikation zu SPSen oder Geräten, die das Modbus TCP Protokoll beherrschen, wie Wago, Beckhoff, Group Schneider, Modicon ...
ControlLogix/ CompactLogix - Ethernet/ IP ControlLogiX	Kommunikation zu Allen Bradley ControlLogiX, CompactLogiX, SoftLogiX
Slc / PLC-5	Kommunikation zu Allen Bradley PLC-5 und SLC
Melsec-Q	Kommunikation zu Mitsubishi Melsec-Q
SNMP	SNMP Client Verbindung zu Geräten, die SNMP nach MIB II unterstützen
DLMS	Kommunikation zu einem intelligenten Stromzähler über das DLMS Protokoll.

OK
Ein Bestätigen mit OK übernimmt die Auswahl.

8. .2 Network PLC protocol

Name of the Connection
Here, you can edit the connection name, if required. All characters except brackets may be used.
Plc Protocol
Here, you can select the PLC application protocol.

Plc Protocol	PLC
Send/Receive	Used for the communication with other PLCs or devices that do not support any of the protocols listed below. Only payload data (raw data) will be transmitted, but no protocol information.
S5 AP	Communication with Siemens S5 CPs or devices.
S7	S7 protocol used to connect to Siemens S7-400, S7-300 or S7-200 PLCs, or devices
Modbus TCP	Communication with PLCs or devices supporting the Modbus TCP protocol, such as Wago, Beckhoff, Groupe Schneider, Modicon ...
CLX - Ethernet/IP	Communication with Allen-Bradley ControlLogiX, CompactLogiX, SoftLogiX
Rockwell Slc / Plc5	Communication with Allen-Bradley PLC-5 and SLC
Melsec-Q	Communication with Mitsubishi Melsec-Q
SNMP	SNMP client connection to devices supporting SNMP using MIB II

OK
Click OK to apply your selection.

8. .3 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

8. .3.1 Spezielle TCP/IP Einstellungen

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.

8. .3.2 IP-Adresse

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.

8. .3.3 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.

8. .3.4 TSAP

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

On the transport layer, ISO (H1) uses so-called TSAPs to address applications. These connection endpoints are required both for ISO (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 Connections.

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.

8. .3.5 TSAP bei S7 Connection

For non-parameterizable 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

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

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).

8. .3.6 SPS 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.

8. .4 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 ISO (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.

8. .5 S7 protocol settings

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

Very seldom - in old controllers so or older compatibles - the S7-200 protocol need to be set by hand.

For the item access please use the S7 item syntax.

8.2 Connection edit

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.

Simulation
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	Network PLC Protocol
Network parameters	TCP/IP Connect Parameters or H1 Connect Parameters or Mail Connect Parameters
OPCpipe parameters	OPCpipe Parameters
OPC-UA settings	OPCUA settings
Protocol parameters	Opens the dialog box for protocol-specific parameters.
Save and Online Test	The data you have entered is saved in the device and then tested. For this purpose, the device will log on to the server.

The number of available options depends on the type of connection.

8.3 Connection copy

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.

8.4 Connection delete

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.

8.5 Connection switch inactive

To disable a connection, do one of the following:

- Menu: ***Connection > Switch on/off***
- Highlight the connection, **right-click** and select ***Switch on/off***

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.

9 Item Syntax

This chapter describes the Tani Item syntax for the corresponding connection (Access Path) that can be used to create new items (tags) .

General information for the address space

- S7 Item Syntax
- S5 Item Syntax
- Modbus Item-Syntax
- PLC-5 / SLC Item Syntax
- ControlLogix/CompactLogix Item Syntax
- MELSEC-Q Item Syntax
- Send/Receive Item Syntax
- System topics

The following chapters are generally:

- Arrays
- Suffixes

9.1 S7 Item Syntax

The S7 item syntax is set up as shown below:

<Area><Data type><Start address>[.Array size]

If the data type is BOOL, the bit number is required:

<Area><Data type><Start address><.Bit number>[.Array size]

Legend: <> mandatory [ ] optional

<Area>

	Syntax	Orientation[1]	Access Rights	Notes
Data block	**DBx.** **V synonym for DB1**	BYTE	Read / write	With blocks, the specification of a block number x is required (x = 1 to 65535). A period or a comma must appear after the block number.
Instance block	**DIx.**	BYTE	Read / write	
Flag	**M or F**	BYTE	Read / write	
Timer	**T**	WORD	Read	
Counter	**Z or C**	WORD	Read / write	
Input	**E or I**	BYTE	Read	
Output	**A or O or Q**	BYTE	Read / write	
I/O Input	**PE or PI**	BYTE	Read	
I/O Output	**PA or PO or PQ**	BYTE	Read / write	

[1]BYTE-oriented means that a byte is addressed for each physical address.
WORD-oriented means that a word (16 bits) is addressed for each physical address.

<Data type>

Type	**Syntax**	DB / DI	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.	-----
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.	-----
INT VT_I2	**I** **Int**	DB5.I3 DB5.Int3	MI4 MInt4 FI4 FInt4	EI4 EInt4 II4 IInt4 AI5 AInt5 OI5 OInt5	PEI4 PEInt4 PII4 PIInt4 PAI5 PAInt5 POI5 POInt5	-----	DB5.I3.2 DB5.Int3.2 MI4.4 FInt4.3 AInt5.3 OI5.2 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.	-----
STRING	**S ******	DB5.S1.2	MS2.3	-----	-----	-----	-----	-----

VT_BSTR	**String** **SF**	DB5.String 1.2	MString2.3					
S7-STRING VT_BSTR	**G **** **SS**	DB5.G2.2	MG2.3	-----	-----	-----	-----	-----
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.	-----
DOUBLE INT VT_I4	**D** **DI** **DInt**	DB5.D3*** 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.	-----
QUAD WORD VT_UI8	**QW** **Qword**	DB5.QW3 DB5.Qword3	MQW4 MQWord4 FQW4 FQWord4	EQW4 EQWord4 IQW4 IQWord4 AQW5 AQWord5 OQW5 OQWord5 QQW5	PEQW4 PEQWord4 PIQW4 PIQWord4 PAQW5 PAQWord5 POQW5 POQWord5	-----	DB5.QW3 MQW4.2 FQW4.2 EQW4.4 AQWord5.8 PEQW4.8 PAQW5.4 POQW5.4 etc.	-----
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.	-----
DOUBLE VT_R8	**QR** **QReal**	DB5.QR2 DB5.QREAL2	MQR4 MQREAL4	EQR4 EQREAL4 IQR4 IQREAL4 AQR4	PEQR4 PEQREAL4 PIQR4 PIQREAL4	------	DB5.QR2.2 DB5.QREAL2.4 MQR4.5	-----

				AQREAL4 OQR4 OQREAL4	PAQR4 PAQREAL4 POQR4 POQREAL4		EQR4.4 AQREAL4.5 PEQR4.2 PAQR4.7 POQR4.3 etc.	
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

*** Caution! Danger of mix up with S5 syntax for bit (DB5D1.1)
**** With SF you can force an S5 string. With SS you can force an S7.

Notes on counters and timers
Counters and timers are always addressed by words. For this reason, the specification of a data type is not required!
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.

9.2 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][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). No period or a comma after the block number.
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	---
LEFT BYTE VT_UI1	**DL**	DB5DL4	DB5DL4.2	---
RIGHT BYTE VT_UI1	**DR**	DB5DR2	DB5DR2.5	---
WORD VT_UI2	**DW**	DB5DW4	DB5DW4.5	---
DOUBLE WORD VT_UI4	**DD**	DB5DD3	DB5DD3.2	---
QUAD WORD VT_UI8	**DQ**	DB5DQ3	DB5DQ3.2	---
S7-STRING VT_BSTR	**S**	DB5S2.3	---	---

<Data type> for all other areaes

	Syntax	M or F	I/O	P / OB	C / T / RS /	with Array	useful

					AS		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	---
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 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	---
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	---
STRING VT_BSTR	**S**	MS4.5 FS4.5	---	---	---	---	---

Notes on counters and timers
Counters and timers are always addressed by words. For this reason, the specification of a data type is not required!
The start address 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.

9.3 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][Suffix]

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.”

<Data type>

Data type VT-Typ	**Syntax**	Syntax R	Syntax I / O	Syntax ER	with Array
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]	----
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
WORD VT_I2	**W**	RW5 HRW5 4W5	----	ERW5 IRW5 3W5	RW5.2 HRW5.2 4W5.2 ERW5.2 IRW5.2 3W5.2
DOUBLE INT VT_I4	**D** **DI**	RD5 HRD5 4D5 RDI5 HRDI5 4DI5	---	ERD5	RD5.2 HRD5.2 4D5.2
DOUBLE WORD VT_I4	**D**	RDW5 HRDW5 4DW5	---	ERDW5 IRDW5 3DW5	RDW5.2 HRDW5.2 4DW5.2 ERDW5.2 IRDW5.2 3DW5.2
QUAD INT VT_I8	**Q** **QI**	RQ5 HRQ5 4Q5 RQI5 HRQI5 4QI5	---	ERQ5 IRQ5 3Q5	ERQ5.2 IRQ5.2 3Q5.2
QUAD WORD VT_UI8	**QW**	RQW5 HRQW5 4QW5	---	ERQW5 IRQW5 3QW5	RQW5.2 HRQW5.2 4QW5.2 ERQW5.2 IRQW5.2 3QW5.2
REAL VT_R4	**R**	RR5	----	ERR5	RR5.2 ERR5.2
DOUBLE VT_R8	**QR**	RQR5	---	ERQR5	RQR5.2 ERQR5.2
STRING VT_BSTR	**S**	RS5.40	----	ERS5.40	----

[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 to single bit in the area of Register and Input Register is not permitted.
Please read the whole register, set the bit and write the register back.

<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.

9.4 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][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	3	decimal
SFC-Status	**SC**	WORD	Read / write	3	decimal
Status	**S**	WORD	Read / write	2	decimal
Timer	**T**	WORD	Read / write	4	decimal
Counter	**C**	WORD	Read / write	5	decimal
Control	**R**	WORD	Read / write	6	decimal
ASCII	**A**	WORD	Read / write	3	decimal

[1] BIT-oriented means that one bit is addressed for each physical address.
WORD-oriented means that one word (16 bits) is addressed for each physical address.
DOUBLE WORD-oriented means that one double word (32 bits) is addressed for each physical address.

[File 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>. A format can be entered behind the colon. If a format is not entered, the data format in the **Orientation** column is used.

Format	**Syntax**	Item Format
String	**S**	STRING
High Byte	**H**	WORD
Low Byte	**L**	WORD
Byte	**B**	WORD

The address is octal for some areas. It is decimal for others. See column labeled Address.
Either the word number, the float number or the bit number is specified as the start address.
<:word> or <:float> or </bit>
The following syntax is used to address a certain bit within a word.
<:word/bit>

NOTE:
When bits are write-accessed, the whole word is written!

Syntax	Erklärung
O:0	Word 0 in Output file 0
O:0/12	Bit 10 (12 octal = 10 decimal) in output file 0
O/12	Bit 10 (12 octal = 10 decimal) in output file 0
I:37	Word 31 (37 octal = 31 decimal) in input file 1
I4:37/2	Bit 2 in Word 31 ((37 octal = 31 decimal) in input file 4
I:1/0	Bit 0 in Word 1 in input file 1
B3/26	Bit 26 in binary file 3
B12:5.15	Array of 15 words in binary file 12, starting at word 5
B12:5/15	Bit 15 in word 5 of binary file 12
F8:0	Float 0
F9:10.16	Array of 16 double words of float file 9, starting at double word 10
N23:4	word 4 of integer file 23
N23:4.10	Array of 10 words in integer file 23, starting at word 4
N23:4/2	Bit 2 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.
Examples: N23:4.10

[Suffix]

A value can be represented in another format with the aid of a suffix.
For more information on suffixes, see also Suffixes.

9.5 ControlLogix/CompactLogix Item Syntax

The item syntax for ControlLogix and CompactLogix is set up as shown below:

The server reads the tags online from the PLC on demand. It is recommended to use the Tag Browsing. **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]

NOTE:
String arrays are not supported, String arrays are resolved instead.

9.6 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 or a comma is required between the <area> and the <type>. If the period or comma is omitted, the syntax uses the simple version. The <type> then corresponds to the default type (for bit area BIT, for word area WORD – see table <Bereich> (Area)).
- If the representation of the start address is HEX, all numbers for this area are also HEX.
- The HEX / decimal numbers can be changed with the following prefixes.
  Conversion of HEX -> DEC: Input of 0d (number zero + the letter d) before the decimal address
  Conversion of DEC -> HEX Input of 0x (number 0 + the letter x) before the hexadecimal address
- When words, double words or strings are registered in a bit area, the start address is a bit address and only possible on word boundaries (dec. 0/16/32... or hex: 0/10/20...) (e.g., Y.D10.3).
- Bit 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	**SS**	BIT	decimal
Retentive Timer Coil	**SC**	BIT	decimal
Retentive Timer Current Value	**SN**	WORD	decimal
Counter Contact	**CS**	BIT	decimal
Counter Coil	**CC**	BIT	decimal
Counter Current Value	**CN**	WORD	decimal
Special Link Relay	**SB**	BIT	HEX
Special Link Register	**SW**	WORD	HEX
Step Relay	**S**	BIT	decimal
Direct Input	**DX**	BIT	HEX
Direct Output	**DY**	BIT	HEX
Index Register	**Z**	WORD	decimal
File Register (Normal Access by block Switching)	**R**	WORD	decimal
File Register (Serial No. Access)	**ZR**	WORD	decimal

[1] BIT-oriented means that one bit is addressed for each physical address. WORD-oriented means that one word (16 bits) is addressed for each physical address.

<.Type> <,Type>

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	**DI**	-----	Y.DI10	R.DI2	Y.DI10.3	R.DI2.3	-----

INT VT_I4	**DINT**		Y.DINT10	R.DINT2	Y.DINT10.3	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.

9.7 Send/Receive Item Syntax

The Send/Receive Item syntax is set up as follows.

<Job><Data type><Start address>[.Array size][Suffix]

If the data type is BOOL, this requires specification of the bit number.

<Job><Data type><Start address><.Bit number>[.Array size][Suffix]

Legend: <> mandatory [ ] optional

<Job>

	Syntax
SEND	**S**
RECEIVE	**R**

<Data type>

Type VT_Type	Syntax	Syntax Send	Syntax Receive	with Array	useful Suffixes
BIT VT_BOOL	**X**	SX1.2	RX1.2	---	---
BIT VT_BOOL	**---**	S1.2	R1.2	---	---
BYTE VT_UI1	**B** **BYTE**	SB4 SBYTE4	RB5 RBYTE5	SB4.4 SBYTE4.5 RB5.10 RBYTE5.3	---
CHAR	**C**	SC4	RC5	SC4.2	---

VT_BSTR	**CHAR**	SCHAR4	RCHAR5	SCHAR4.2 RC5.4 RCHAR5.4	
WORD VT_I4 (VT_UI2*)	**W** **WORD**	SW10 SWORD10	RW10 RWORD10	SW10.2 SWORD10.2 RW10.4 RWORD10.4	---
INT VT_I2	**I** **INT**	SI4 SINT4	RI6 RINT6	SI4.2 SINT4.2 RI6.10 RINT6.10	---
DOUBLE WORD VT_R8 (VT_UI4*)	**D** **DW** **DWORD**	SD6 SDW6 SDWORD6	RD6 RDW6 RDWORD6	SD6.2 SDW6.2 SDWORD6.2 RD6.5 RDW6.5 RDWORD6.5	---
DOUBLE INT VT_I4	**DI** **DINT**	SDI6 SDINT6	RDI2 RDINT2	SDI6.2 SDINT6.2 RDI2.5 RDINT2.5	---
QUAD WORD VT_UI8	**QW** **QWord**	SQW SQWord	RQW RQWord	SQW6.2 RQW6.2	---
QUAD INT VT_I8	**Q** **QI** **QInt**	SQ SQI SQInt	RQ RQI RQInt	SQI6.2 RQI6.2	---
REAL VT_R4	**R** **REAL**	SR2 SREAL2	RR2 RREAL2	SR2.2 SREAL2.2 RR2.5 RREAL2.5	---
STRING VT_BSTR	**S** **STRING**	SS5.2 SSTRING5.2	RS5.2 RSTRING5.2	---	---
S7-STRING VT_BSTR	**G**	SG5.2	RG5.2	---	---

<Start address>

The start address specifies the byte address starting at which sending and receiving begins.
Example: SWORD10: Word 10 is the start address.
If the start address is a certain bit, specification of the bit number is also required.

<.Bit number>

When the data type is BOOL, the bit number must be specified!
Example: SX1.2: Bit 2 of byte 1 is the start address.

[.Array size]

Arrays are created to combine several units of one data type into a field.
Example: RWORD10.4
For more information on arrays, see also Arrays.

[Suffix]

A suffix can be used to represent a value in another format.
For more information on suffixes, see also Suffixes.

9.8 System Tree

The **System tree** contains the s**pecial items** the server supports fixed. They allow accessing internal variables and stati.

The system tree **"System"** lies in
- **OPC-DA and OpcPipe**: direcly 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.

The system topic can be switched of with the configuration software under "general system settings" "Use system topic".

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"

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 datea as the list op topics, version information
Memory	Temporary variables and structures which relays on the system memory.
<**user defined**>	User defined topics created with the configuration software or with the browser or over writing to **System**.**AddTopic**

System

Topic name	Description
AddTopic	Adds connections to controllers. With writing of a string in the defined syntax connecions will be added. The syntax is described in AddTopic.
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.
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.

Values for System.Topics.<topicName>

Topic name	Description

Status	Brings the status of the connection. 0-> ok 3 -> no connection 5 -> access denied 29 -> Suspended. No OPC client or logic table used one item.
Status<n>	On redundancy connections this is containing the status of the lower connection. On double redundancy there exist **Status1** and **Status2**.
ReadCount	Number of handled syncronous 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. It shows which connection actually is the master connection. The first connection has the number one. Writing this item switches the connection, the written number becomes the master connection. Hint: Writing an invalid value does nothing.

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 wheather 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

Topics can be created over this topic.

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**.

Topic name	Description
Name	<Topic Name>. The name of the created connection. Please note the rules for topic names which will existing from various OPC clients.
PlcType	Valid are: **Value** / **Description** s7 — Siemens S7-200, 300, 400, 1200 1500 and campatible as

		Speed7. Can be used for accessing the controller over MPI adapter from Hilscher, Process Informatik, IBH Softec, Softing, Helmholz.
	s5	Siemens S5 with network CP All racks are supported 135, 155, 188. All CPU types.
	compactlogics	Rockwell Control Logix and Compact Logix.
	slc	Rockwell SLC family.
	modbus	Modbus TCP and compatible systems as Wago, Beckhoff, Modicon, Omron and more..
	melsecQ	Mitsubishi Melsec family.
	raw	Raw data.

NetworkProtocol	Valid are: **Value** / **Alternative** / **Description** **ip** / **tcp** / TCP/IP TCP over IpV4 (192.168.1.1) or IpV6 <235b:34aa::0001:0030) or domain names **udp** / / TCP/IP UDP over IpV4 (192.168.1.1) or IpV6 <235b:34aa::0001:0030) or domain namens **h1** / **osi** / OSI/H1 over MAC Addresses (080006010001) or (08:00:06:01:00:01)
NetworkAddress	IpV4 (192.168.1.1) or IpV6 (235b:34aa::0001:0030) or domain names (tanindustrie.de)
Port	1 .. 65534. This is needed for IP connections only.
ConnectType	Valid are: • Server • Client
OwnTSAP	in hexadezimal. This is valid on OSI/H1 or Port 102 (RFC1006) connections only. (0101)
DestinationTSAP	in hexadezimal. This is valid on OSI/H1 or Port 102 (RFC1006) connections only. (0302)
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 functiones.
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.

9.9 Arrays

The word array means a series of equal elements (field, row, data area). An array combines several units of one data tpye in a field. To create an array, the length of the array is added to the standard syntax, separated by a period.

Arrays are impossible in:
• STRINGS (is already an array of characters) • ISO

9.10 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.	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

			999*			day, the hour, the minutes, the seconds and the milliseconds are included. Remember to use the correct separators (hyphen, colon and period)!

* ms can be omitted.