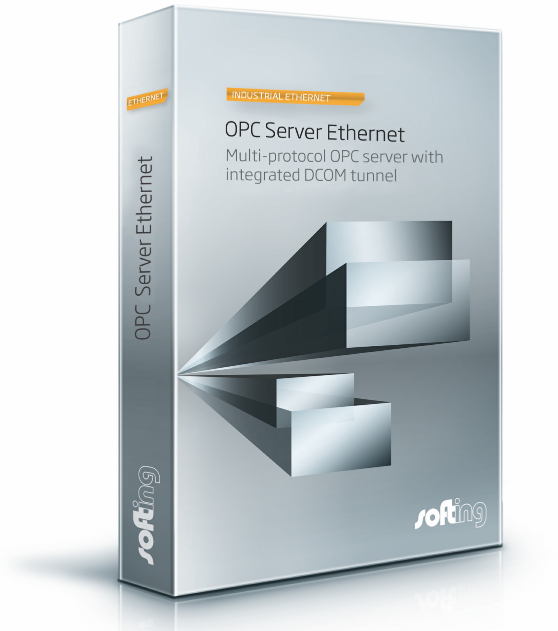
MANUAL

OPC-Server Ethernet

Multi-Protocol OPC Server Ethernet with
integrated DCOM Tunnel (OPCpipe)



Version: E-032012-01

Disclaimer of liability

We have checked the contents of the document for conformity with the hardware and software described. Nevertheless, we are unable to preclude the possibility of deviations so that we are unable to assume warranty for full compliance. The information given in the publication is,however, reviewed regularly.
Necessary amendments are incorporated in the following editions.
We would be pleased to receive any improvement proposals which you may have.

This document may not be passed on nor duplicated, nor may its contents be used or disclosed unless expressly permitted. Violations of this clause will necessarily lead to compensation in damages.

All rights reserved, in particular rights of granting of patents or registration ofutility-model patents.

Softing Industrial Networks GmbH

Ostendstr. 50 A

D-90482 Nürnberg

Tel: + 49 (0)911 544 27-0

Fax: + 49 (0)911 544 27-27

Internet: www.softing-in.de

Email: info-in@softing.com

Important notes

ead the manual before the start. For damages due to improper connection, implementation or operation Softing refuses any liability according to our existing guarantee obligations.

The recent version of this manual is available in the Download Area of the Softing at: www.softing-in.de.

Table of Contents

Chapter 1

Help-Overview

1 Help-Overview

INAT OPC Server Ethernet

INAT OPC Server - General

- Overview
- System Prerequisites
- License Requirements
- Server Revision

The Basics

- DCOM
- OPC Basics
- PLC Connections
  - S7 - General Information
  - S5 - General Information
  - Modbus - General Information
  - CLX - General Information
  - PLC-5 / SLC - General Information
  - netLINK - General Information
  - MELSEC-Q - General Information
  - Send/Receive - General Information
  - OPCpipe - General Information

Operation and configuration

- How To Proceed
- Licensing
- Tool Bar
- Screens
- Connection
- Configuration
- Dialog Screens
- Menu
- Troubleshooting

Accessories

- NetCon OPC
- Selecting the Server Type
- Server Tray
- Logger
- OPC Client

Item-Syntax

- S7 Item Syntax
- S5 Item Syntax
- Modbus Item Syntax
- PLC-5 / SLC Item Syntax
- MELSEC-Q Item Syntax
- Send/Receive Item Syntax
- OPCpipe Item Syntax
- Arrays
- Suffixe
- Data Types
- Special Items

Chapter 2

INAT OPC Server – General

2 INAT OPC Server – General

This chapter describes the following subjects:

2.1 Overview

INAT OPC Server Ethernet

The INAT OPC server Ethernet makes it possible to exchange data between the field devices (PLCs) of a wide variety of manufacturers and OPC clients via Ethernet TCP/IP, RFC1006 and ISO (H1).

Separation of OPC Core and NetCon OPC User Interface (GUI)

The core and the user interface were separated from each other in the INAT OPC Server 4.05. The Core contains the OPC server functionality and is started automatically as a service when the system starts.

NetCon OPC

The interface for configuration and diagnostics is handled by the NetCon OPC program. The GUI connects to the server via TCP/IP (port 982) which also enables remote access to other INAT OPC 4.05 servers in the network. See also NetCon OPC.

Server Tray

Setup contains a server tray for local operation of NetCon OPC and INAT OPC Server 4.05. The server tray shows the operating status of the INAT OPC server, starts or concludes the server service and starts the interface for local parameterization, diagnosis and logger.
The server service and the server tray communicate with the NetCon OPC via TCP/IP (Port 982). A maximum of four stations can be connected via one INAT OPC server via TCP/IP (Port 982). One station can be both the NetCon OPC and for terminal servers the server tray. The maximum number of stations must be handled particularly carefully since one server tray is started in Autostart for every new session. When the maximum number of stations is reached, there is no way left to configure the access of server with the NetCon OPC interface. See also Server Tray.

Access protection

As soon as several stations access the server, the write access rights are regulated by the master-slave principle. To prevent unauthorized persons from changing the configuration, it is possible to create a password. All stations have read-access rights. See also NetCon OPC.

Logger

A separate logger service was implemented for the INAT OPC Server 4.05. This service can be observed via the NetCon OPC and configured. This also makes it possible to log in the file. See also Logger.

2.2 System Prerequisites

The following minimal system prerequisites apply to the operation of the INAT OPC server:

Processor:	Intel Pentium II 200 MHz processor
Working storage:	Minimum: 64 MB RAM
Hard drive capacity:	25 MB hard drive memory
Operating system:	Windows XP Windows 2003 Server 32 or 64 Bit Windows 2008 Server 32 or 64 Bit Windows 2008 R2 Server 32 or 64 Bit Windows 7 32 or 64 Bit

2.3 License Requirements

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

SOFTWARE PRODUCT

Order no.: 100-3100-01	OPC Server TCPIPH1 for communication via Ethernet TCP/IP, RFC1006 and ISO (H1)
Order no.: 100-3120-01	OPC Server TCPIP-S for communication via TCP/IP and RFC1006
Order no.: 100-3140-01	OPC Server Modbus for communication via Ethernet Modbus TCP
Order no.: 100-3150-01	OPC Server TCPIP-I for communication via Ethernet TCP/IP
Order no.: 100-3160-01	OPC Server TCPIP-M for communication via Mitsubishi Melsec-Q-TCP/IP controllers via Ethernet TCP/IP
Order no.: 100-3300-01	OPC Server MPI/PPI for communication with Siemens S7-controllers via MPI/PPI
Order no.: 100-1700-01	OPCpipeClient
Order no.: 100-3050-01	DDE Server TCPIPH1 for data exchange via Ethernet TCP/IP, RFC1006 and ISO (H1)
Order no.: 100-3070-01	DDE Server TCPIP-S for data exchange via Ethernet TCP/IP and RFC1006
Order no.: 100-3350-01	DDE Server MPI/PPI for data exchange with Siemens S7 controllers via MPI/PPI
Order no.: 100-2265-03	NetSpector; protocol analysis software for analyzing Ethernet networks

The modules can be purchased individually or in different combinations. The licensing contract applies the same to each individual version. This is why the term SOFTWARE PRODUCT is used for all versions in this manual.

LICENSER
Softing Industrial Networks GmbH
Ostendstraße 50A
D-90482 Nürnberg

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

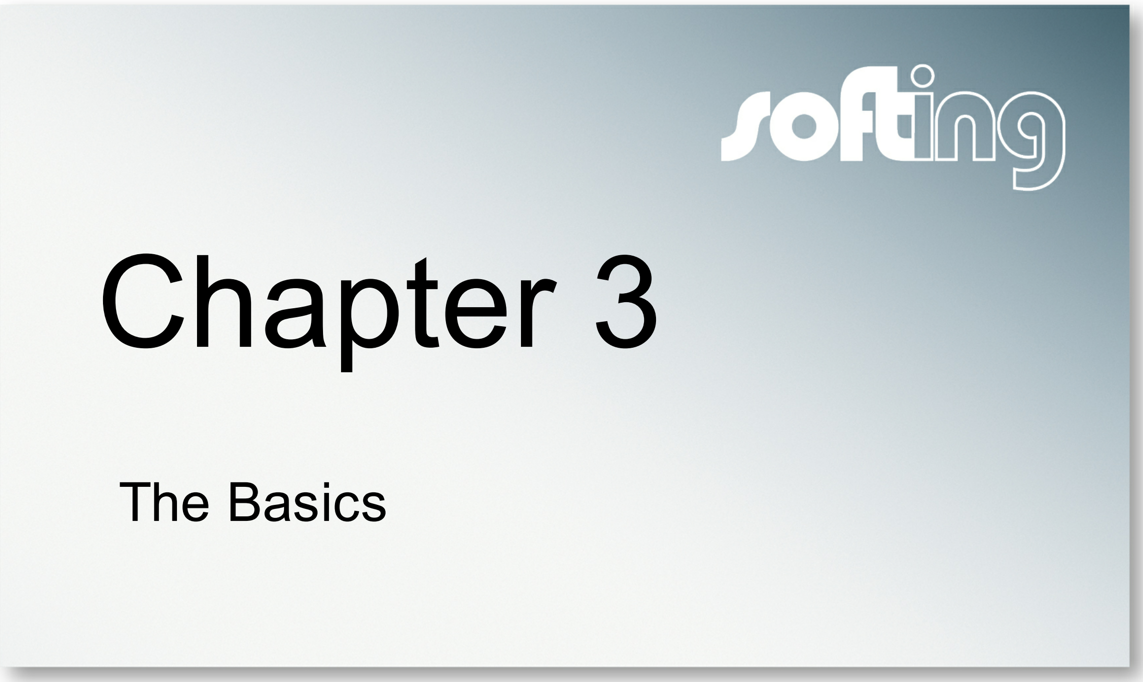
1. The license given to the LICENSEE is valid until it is terminated by the licensee or the LICENSER.
2. The LICENSEE can terminate the license at all times by returning the SOFTWARE PRODUCT including the archivation copy and all related documentation to the LICENSER.
   If the return is not due to a 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.

2.4 Server Revision

The revision of the server is located in the installation directory of the server in either the file **revision_opc_d.htm** or **revision_opc_e.htm**.



Chapter 3

The Basics

3 The Basics

This chapter describes the following subjects:

- DCOM
- OPC Basics
- PLC Connections

3.1 DCOM

This chapter describes the following subjects:

- What is DCOM?
- DCOM configuration
- DCOM Configuration: Server Computer
- DCOM Configuration: Client Computer

3.1.1 What Is DCOM?

DCOM (Distributed Component Object Model) is an object-oriented RPC system which is based on the DCE standard. It was defined by Mircosoft so that COM technology could communicate via a network. DCOM (Distributed Component Object Model) is an expansion of COM. While COM makes the communication of objects on a computer possible, objects which are located on different computers can also use DCOM to communicate with each other. DCOM enables COM calls between distribution computers within one network.

In addition to COM technology, OPC also uses DCOM to achieve network capability. This means that not only are the services of the OPC server which are located on the local PC available to an OPC client but also the services of all OPC servers in the network.

DCOM is always used then when client and server are located on different computers (i.e., remote servers).

Several settings in DCOM configuration are necessary so that client and server can communicate with each other via DCOM.

When more security is needed on a PC, the DCOM settings must also be changed when OPC server and OPC client are running on one PC.

NOTE:
One way to get around DCOM and the time-consuming configuration is to use the OPCpipe.

3.1.2 DCOM Configuration – General

DCOM has been available on all Windows platforms since Windows NT.

The program **dcomcnfg.exe** is available to make changes to the DCOM configuration.

The DCOM configuration is started via one of the following:

- Calling the menu **Settings** - **DCOM settings** of NetCon OPC
- By entering "**DCOMCNFG**" in the command line of the prompt
- Via **System control** - **Administration** – **Component Services.**

The start rights should be set correctly on the OPC server so that the OPC server can be started automatically by the OPC client.

NOTE:
The registered user must have administrator rights before he/she can change the DCOM configuration.

NOTE:
Use of the OPCpipe is one way to get around the DCOM and thus the time-consuming configuration.

OPC server and OPC client on the same computer

If you want to use the OPC server as a service, it is mandatory that the **interactive user** be entered as standard in the access rights of the global COM security settings.

OPC server and OPC client on different computers

The DCOM configuration must be adapted on both the server computer and the client computer.

3.1.3 DCOM Configuration: Server Computer

The server computer is the computer on which the **OPC server** is running.

NOTE:
The registered user must have administrator rights before he/she can change the DCOM configuration.

Both "Properties of INAT OPC server" and "Workplace Settings" must be adapted on the server side.

3.1.3.1 Workplace Settings

The workplace settings are made by browsing in **Console master** via **Component services** > **Computer** > **Workplace**. One click with the right mouse button on **Workplace** opens the menu and permits Properties to be processed.

3.1.3.1.1 General

No changes are necessary for the **General** settings. The default setting can be used.

3.1.3.1.2 Options

No changes are necessary for the **Options** settings. The default setting can be used.

3.1.3.1.3 Standard Properties

Standard authentication level: **Connect**

Standard identification change level: **Identify**

3.1.3.1.4 Standard Protocols

Connection-oriented TCP/IP should be listed first for the **Standard protocols**.
This setting is used to handle the DCOM connections via TCP/IP.

3.1.3.1.5 MSDTC

No changes are necessary for the "**MSDTC**" settings. The default setting can be used.

3.1.3.1.6 Standard Security

Access rights:
The users who are allowed to access the application can be specified for applications which do not provide access rights settings themselves.

Start rights:
The users who are allowed to start the application can be specified for applications which do not provide start rights settings themselves.

Limits
The limits can be set for applications which change the rights.

3.1.3.2 OPC Server Settings

The OPC server settings are made by browsing in **Console master** via > **Component services** > **Computer** > **Workplace** > **DCOM configuration** > **INAT TcpIpH1 OPC Server**.
A click with the right mouse button on **INAT TcpIpH1 OPC Server** opens the menu and permits **Properties** to be accessed.

3.1.3.2.1 General

No changes are necessary for the **General settings**. The default setting **Authentication level = Standard** can be used.

3.1.3.2.2 Execution Location

No changes are necessary for the **Execution Location** settings. The default setting can be used.

3.1.3.2.3 Security

The security settings for the start, access and configuration of the OPC server are set here.
The appropriate rights are specified with the "**Edit**" button.

START RIGHTS

These rights are changed as shown below:

- Pressing the "**Edit**" button.
- Clicking the "**Add"** button to add group or user names.
- Click the "**Path"** button.
- Start rights should be assigned for:
  - Administrators
  - ANONYMOUS
  - SERVICE
  - INTERACTIVE
  - NETWORK
  - SYSTEM
  - The user who starts the client

- Highlight the desired names and press the "**OK**" button.
- Then select each one separately in the list and click "**Permit"** for all possibilities.
- Then click the "**OK**" button.

ACCESS RIGHTS

The access rights are changed the same way the start rights are changed.
Access rights should be provided for:

- Administrators
- ANONYMOUS
- SERVICE
- INTERACTIVE
- NETWORK
- SYSTEM
- The user who starts the client

- Highlight the desired names and press the "**OK**" button.
- Then select each one separately in the list and click "**Permit"** for all possibilities.
- Then click the "**OK**" button.

CONFIGURATION RIGHTS

Proceed as described for **ACCESS RIGHTS.**

These settings will work with most OPC clients. If access by another user is also desired, this user can still be added.
If access is to be further limited, the applicable entry can be removed.

NOTE:
Some changes do not take effect until after a new start of the PC although no special message is output.

Alternate action: The standards can be adjusted and **Standard** can be selected at the top.

3.1.3.2.4 End points

No changes are required in the "**End points"** settings. The default setting **"Standard system protocols"** can be used. This means that the protocols are used which are set under
Standard protocols.

3.1.3.2.5 Identity

No changes are required in the **"Identity"** settings. The default setting "**Interactive user**" should be used.

Interactive user:
The interactive user is the user who is logged in at the moment. This is the only way that the OPC server can be made available to additional clients.

User who starts the application:
This setting should not be used. The security settings of the client application will be used instead.

This user:
The server will use the security settings of the specified person. Every time the server is started, the security settings of the specified person are used.
If this is the case, start rights must only be given to this person on the client side.
The user “Interactive User” is set as the default during installation.

System account (only for service):
This setting can be selected if the server is being run as a service.

NOTE:
All settings except "Interactive user" and "System account" start the OPC server in an own process area. An additional server then starts the OPC server again.
Then the clients do not use the same information.

3.1.4 DCOM Configuration: Client Computer

The client computer is the computer on which the **OPC client** is running.

> **NOTE:**
> **The registered user must have administrator rights before he/she can change the DCOM configuration.**

The work place settings are made by browsing in the **console master** and going to > **Component services** > **Computer** > **Work place**. Click once with the right-hand mouse button on **Work place** to open the menu so that you can process the **Properties**.

3.1.4.1 Options

No changes are required in the "**Options"** settings. The default setting can be used.

3.1.4.2 Standard Characteristics

"**Activate DCOM on this computer**" should be activated. The standard authentication level and the standard identity level should be set as shown below:

Client computer in one work group:
Authentification level: None
Identity change level: Anonymous

Client computer in one domain:
Authentification level: Connect
Identity change level: Identification

Client computer in a mixed configuration (Client computer in a work group, Server computer in a domain):
Authentication level: None
Identity change level: Anonymous

3.1.4.3 Standard Protocols

The **"Connection-oriented TCP/IP"** should be preferred of all the standard protocols. When this setting is selected, DCOM connections are handled via TCP/IP.

3.1.4.4 MSDTC

No changes are required with the "**MSDTC**" settings. The default setting can be used.

3.1.4.5 Standard Security

The "**Start and activation rights**" and "**Access rights**" can be changed here for all COM objects. The settings must be changed in such a way that the client application can be started and accessed on the OPC server.

Start and access rights for:
- Administrators
- Interactive users
- System
- Network
- or the **Group** or the **User** under which the OPC client is running

The **Configuration rights** do not need to be changed on the client computer.

3.2 OPC Basics

This chapter describes the following subjects:

- What Is OPC?
- OPC DA Specification

3.2.1 What Is OPC?

OPC (OLE for Process Control) is an open standard for the uniform access of different Windows-based software applications (HMI, table calculation, archivation software, and so on) on the various data sources at the automation level (PLC, scales, scanner, etc.). Before the OPC was introduced, one driver had to be provided for every single software application. This meant that controller X required one driver for the connection to software application A, one driver for the connection to software application B, one driver for the connection to application C, and so on. The data had to be provided individually to each of the data sources (e.g., once for the inquiry of application A, once for the inquiry of application B, once for the inquiry of application C, and so on). And naturally the same thing applied to controller Y. One driver per application had to be provided here too. The number of drivers was further increased by different operating systems on the application side and different communication protocols and bus systems on the hardware side.

A uniform solution was needed for the access of the software applications to the various data sources (i.e., a manufacturer-independent software interface). OPC is a manufacturer-independent interface standard for automation technology which simplifies data communication significantly. Data can be transferred with OPC to any data source to any application.

The OPC interface uses OLE mechanisms. In turn, the basis for OLE is COM.

COM (Component Object Model)
- Client / server architecture
- COM defines a generally valid way to access software services.
- COM objects are implemented within the server. The COM objects offer services via methods which are combined in interfaces.
- Clients are applications which use the services of the object by calling a method of an interface.

Manufacturers of the devices which supply process data provide an OPC server interface with the module. The OPC server is responsible for the link to the data source. Manufacturers of applications which record, visualize, etc. data provide an OPC client interface with the software application. The client can generate and utilize objects on the server via the OPC server interface.

3.2.2 OPC DA Specification

The OPC Foundation has defined a series of specifications, whereby each specification is provided for a certain area of use. The OPC DA specification is the first specification of the OPC Foundation and is provided for process data communication. It defines an interface between client programs and server programs for the data exchange. An OPC server allows one (or more) OPC client(s) to access process data via this interface.

Class model

The OPC specification for data access defines three hierarchical classes in the class model: OPCServer, OPCGroup and OPCItem.

The client program uses COM calls of the operating system to generate an object of the OPCServer class. The other objects are generated by OPC methods.

Classes: OPCServer

An object of this class represents a manufacturer-specific OPC server. OPC servers implement the link to the hardware and are provided with the process data (CPs, measuring devices, etc.) by the manufacturers of the modules. The manufacturer gives the OPC server an unambiguous name so that it can be identified. This name is known as the ProgID in accordance with the COM standard. This program identifier contains a read-accessible string which describes the components. All types of the INAT OPC server Ethernet have the following Prog.ID:

- **INAT TCPIPH1 OPC Server**

The OPC server class contains methods which can be used by the client to administer objects of the OPCGroup class.

Class: OPCGroup

An object of this class structures the variables used by the OPC server. The OPCGroup administers the individual process variables called the OPC items. With the aid of the OPCGroup, the client is able to create useful units of variables.

Class: OPCItem

An object of this class represents the actual process variable. The item in the address area of the OPC server is identified by its Item-ID (item syntax). An item is the connection to the data source and NOT the data source itself. Since the item ID is specified by the manufacturer of the server, it must be unambiguous within the name area of the server. Each item has the following properties:

Value	The last acquired value of the variable.
Quality	Quality of the value
Time stamp	Time at which the current value was determined for the first time.

Types of data access

Data communication between the DA server and the DA client can be synchronous or asynchronous. The variables can also be observed. In addition, the values can be read out of cache or directly out of the device.

Device	The hardware itself
Cache	Intermediate storage for all variables. An internal server image of the process data.

Read synchronously
During synchronous communication, the client calls the SyncRead method and waits for the return value. A new job cannot be started until this happens. The client is blocked as long as no return value has been received from the server.

Write synchronously
The client sends a write job to the server. One or more items of a group are written with the SyncWrite method. The write operation does not return a result message unless the write operation was not successful. To check whether the write operation was successful, the value must be read back immediately after the write access.

> NOTE:
> **The synchronous access should be used in cases where an interruption of the user program plays a subordinate role.**

Read and write asynchronously
With asynchronous communication, the program that sent the asynchronous function call to access process data receives a response message immediately as to whether the job was successfully given to the OPC server. The program then continues its work. The client receives the return value from the server as soon as this is finished. The server calls the AsyncReadComplete / AsyncWriteComplete function of the OPC client.

> NOTE:
> **Asynchronous accessing is helpful when large amounts of data are to be read and the user program must react during processing.**

Monitoring variables
During the monitoring of variables, all active OPC items in all active OPC groups are monitored. If a value or the quality of a variable changes, the OPC client is informed of this. The client is not strained during monitoring. The update rate must be set so that the client is not overloaded by change messages when process variables are changing quickly.

> NOTE:
> **Monitoring variables is helpful if the user program has to be informed at all times as to the current value and the status of the data.**

3.3 PLC Connections

This chapter describes the following subjects:

- S7- General Information
- S5 - General Information
- Modbus - General Information
- CLX - General Information
- PLC-5 / SLC - General Information
- netLINK - General Information
- MELSEC-Q - General Information
- Send/Receive General Information
- OPCpipe - General Information

3.3.1 S7 - General Information

The S7 protocol is used for communication with S7 controllers and for communication with CPs which can handle the S7 protocol.
The S7 protocol can be used in connection with both the TCP/IP and the H1.

Read (fetch active) and write (write active)

- With communication with S7 controllers, both read and write accesses can always be performed via an access path.
- The S7 protocol is usually parameterized with RFC-1006.
- In addition to the address of the S7 controller (IP address or MAC address), a “TSAP for read and write accesses” is required.
- Data can be read and written from/to an S7 controller (layer 7 communication) using:
  - Non-parameterizable access paths
  - Parameterizable access paths

Non-parameterizable access paths
An access path is only set up on the OPC side – NEVER on the controller side.
Standard access paths are required that are handled via standard TSAPs or standard ports.
There are only a limited number of standard access paths. If this is not enough, “parameterizable access paths” must be used.
Standard access paths with Siemens CPs are handled by RFC1006 or H1 (see Standard-TSAPs and TSAP with S7 access paths).

Parameterizable access paths
The access path must be parameterized on both sides of the communication (a Fetch/Write active access path on the OPC server AND a Fetch/Write passive access path for the S7-CP). If communication uses echolink, parameterizable access paths are required.

NOTE:
Both the S7 Item Syntax and the S5 Item Syntax can be used for communication with S7 controllers.

3.3.2 S5 - General Information

The S5 protocol is used for communication with S5 controllers (S5-AP header) and for communication with CPs which can handle the S5 protocol such as echolink and echocollect.
The S5 protocol can be used both in connection with TCP/IP and with H1.

Read (fetch active):_

- A read connection is set up on the OPC server so that data can be actively read from an S5 controller.
- A "port for read accesses" or a "TSAP for read accesses" is required in addition to the address of the S5 controller (IP address or MAC address)
- The read connection must be parameterized on both sides of the communication (fetch active connection on the OPC server, fetch passive connection on the S5).

Write (write active):

- If data is also to be written to the S5 controller, a write connection is set up in addition to the read connection.
- The write connection path needs a "port for write accesses" or a "TSAP for write accesses."
- The OPC server offers a way to enable both jobs (read and write) within ONE OPC connection. Both "Port für Lesen" (port for read access) / "TSAP für Lesen" (TSAP for read access) and "Port für Schreiben" (port for write access / "TSAP für Schreiben" (TSAP for write access) are defined within this OPC access path.
- The write connection must be parameterized on both sides of the communication (write active connection on the OPC server, write/receive passive connection on the S5).
- The more recent S5-TCP/IP cards - echolink echocollect - support read and write accesses via a connection.

NOTE:
Both the S5 Item Syntax and the S7 Item Syntax can be used for communication with S5 controllers.

3.3.3 Modbus - General Information

The Modbus on TCP protocol is used for the communication with the controllers of Wago, Groupe Schneider, Beckhoff and all CPs which can handle the Modbus-on-TCP protocol (Modicon, Ethernet terminals from Phoenix, Wago, Beckhoff and many more).
The Modbus protocol is only possible with TCP/IP.

Read (fetch active) and (Write (write active):

- With communication with Modbus controllers, you can always read and write via a connection.
- When standard port 502 is used, the connection must only be parameterized on the OPC server side. No settings are required on the controller side.

Modbus Symbol Import
A Modbus Symbol Import for the variables that were programmed in CoDeSys and to which absolute addresses are assigned is offered for the PLC program of Wago controllers.
See Modbus Symbolimport

NOTE:
A special Modbus Item-Syntax is used for communication via Modbus on TCP.

3.3.4 CLX - General Information

The CLX protocol is used for communication with ControlLogiX / Compact Logix controllers from Rockwell Automation.
The CLX protocol can only be used in connection with TCP/IP.

Read (fetch active) and (Write (write active):

- Both read and write accesses via a connection are always possible for communication with CLX controllers.
- If the standard port 44818 is used, the connection must only be parameterized on the OPC server side. No further settings are necessary on the controller side.

NOTE:
On the OPC client, all the item IDs are only addressed via the symbolic names from the CLX.

Alias Browsing:
When the CLX connection is set up, the symbol information is read out from the PLC and stored intermediately in the configuration directory.
If the PLC program is changed by the programmer, the symbol information must be read in again. The server recognizes a program change when the program starts. If the program changes while the server is in operation, you will have to update the symbol information by hand. This is done in the dialog AB ControlLogiX Symbols.

3.3.5 PLC-5/SLC - General Information

The protocol is used for the communication both for PLC-5 and SLC controllers of Rockwell Automation Allen-Bradley.
The PLC-5/SLC protocol can only be used in connection with TCP/IP.

Read (fetch active) and write (write active):

- An access path can always be read and written during communication with SLC and PLC-5 controllers.
- If standard port 2222 is used, the connection must only be parameterized on the OPC server side. No further settings are necessary on the controller side.

NOTE:
A special PLC-5 / SLC Item Syntax is used for the PLC-5 and SLC communication.

3.3.6 netLINK - General Information

netLINK is an application protocol which you can use to actively read and actively write data to and from S7 controllers via a netLINK adapter.

The netLINK protocol can only be used in connection with TCP/IP.

Read (fetch active) and write (write active):

- Read and write accesses via a connection are always possible for communication via netLINK.
- Port number 1099 should not be changed.

NOTE:
The S7 Item Syntax or the S5 Item Syntax are used for communication via netLINK.

NOTE:
More recent netLINKs use the S7 protocol and don’t need the netLINK connection. See S7 - General Information.

3.3.7 MELSEC-Q - General Information

The MELSEC-Q protocol is used for communication with Mitsubishi controllers of the MELSEC Q series.
The MELSEC-Q protocol is only possible in connection with TCP/IP.

Read (fetch active) and (Write (write active):

- Read and write accesses via a connection are always possible during communication with MELSEC-Q controllers.
- The connection must be parameterized on both sides of the communication (one connection for the OPC server and one connection for the MELSEC-Q controller).
- One “Port für Lesen und Schreiben” (port for read and write accesses) is required on the OPC side in addition to the address of the MELSEC-Q controller (IP address).
- Port 8192 is used as the standard. The port must be identical on both sides of the connection.
- If the MELSEC-Q controller does not have parameterizable connection with fixed ports (“standard connections”), the connection must only be parameterized on the OPC server side. No further settings are necessary on the controller side. Remember to use the same ports on the OPC server that are used on the MELSEC-Q controller.

NOTE:
Addresses are often entered in HEX for Mitsubishi PLCs (port numbers must then be converted into decimal format for the OPC side).

NOTE:
A special Melsec-Q Item-Syntax is used for Mitsubishi communication.

3.3.8 Send/Receive – General Information

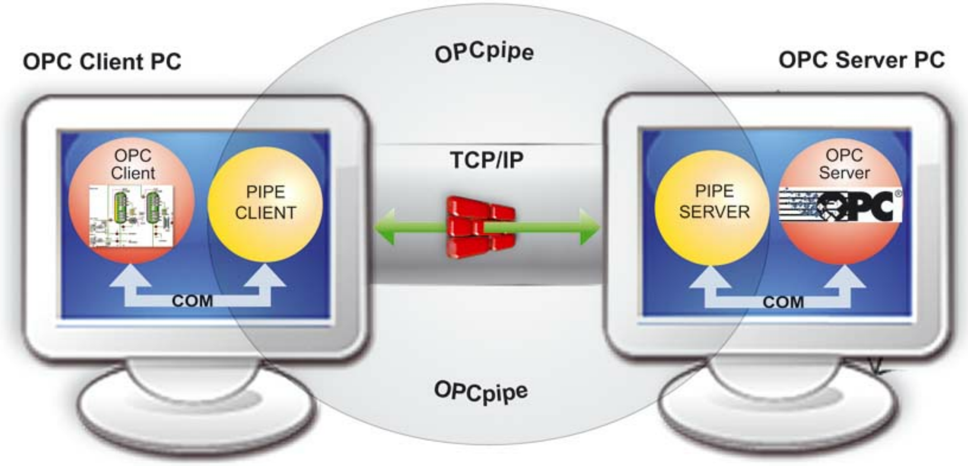
Data transmission is done without application headers (i.e., raw data are transmitted) for send/receive communication.
The communication partner can be any station which supports the “send/receive direct” interface.
The send/receive protocol is possible with both TCP/IP and H1.

Receive data and send data

- **"Events from PLC permitted"** must be activated before data can be received.
- **"Write to PLC permitted"** must be activated before data can be sent.
- Sending and receiving are always possible via a connection.
- In addition to the address of the S/R controller (IP address or MAC address), a "**Port for events and write accesses**" or a "**TSAP for events and write accesses**" is required.

NOTE:
A special Send/Receive Item Syntax is used for send/receive communication.

3.3.9 OPCpipe – General Information



OPCpipe is a tunnel protocol during which the data are “tunneled” (i.e., exchanged) for OPC communication between client and server.
The OPCpipe consists of two parts.

OPCpipe Server: Runs on the same computer as the OPC server.
OPCpipe Client: Runs on the same computer as the OPC client.

The OPCpipe client receives the inquiry of the OPC client and converts OPC communication into “normal” TCP/IP communication. Tunneled like this, the data are transmitted via a standard TCP/IP connection via the network to the target computer. Upon arrival, the OPCpipe server receives the data, “decrypts” them again into OPC communication and passes the data on the OPC server. The server executes the inquiry and then sends the data back to the OPC client. The tunnel mechanism is identical in both directions.

Setting up an OPCpipe connection

The OPCpipe connection must be configured on both the server side (computer on which the INAT OPC server is running) and on the client side (computer on which the OPC client is running).
This means that the OPC server with the OPCpipe function must be installed on both sides of the communication.
Usually the OPCpipe client initiates the active establishment of the connection.

Firewall

To ensure that the OPCpipe server will accept the connection, it should be ensured that the ports configured in the firewall are activated. The default port is the number 4444.

NOTE:
A special OPCpipe Item Syntax is used for OPCpipe communication.

Chapter 4

Operation and configuration

4 Operation and configuration

This chapter describes the following subjects:

- How To Proceed
- Licensing
- Tool Bar
- Screens
- Connection (Access Path)
- Configuration
- Dialog Screens
- Menu
- Troubleshooting

4.1 How To Proceed

Parameterization of the OPC server is very simple and takes only a few steps. The following steps must be performed for OPC communication.

OPC server configuration

A connection is set up with NetCon OPC on the OPC server for each piece of hardware from which data are to be read or to which data are to be written. The connection defines the access path to the hardware (i.e., the channel between the two communication partners). All parameters that are required for the communication between OPC server and the hardware are defined here. The protocols that support the communication partners (i.e., S7 protocol and RFC1006) are selected here. A certain piece of hardware is specified via the target addresses (e.g., IP address and port numbers, or MAC address and TSAP). You can also select whether this connection is always to be write accessed or not.
Set up new access path

Connecting the OPC client and the OPC server

Before OPC server and OPC client can exchange data with each other, an access path must be established between the two. The OPC client that uses the “Connect” function to connect itself with the server always takes the initiative. The access rights are set via the DCOM settings. See also DCOM settings.

Configuring the OPC client

On the OPC client, the data points are either selected via tag browsing or entered via a name. The data points address the hardware via the Item syntax. Addressing can be either absolute or symbolic. Access path, name and item name are specified for addressing. The access path name corresponds to the connection name (see OPC server configuration).
With absolute addressing, the protocol-specific item syntax must be adhered to.
See Item-Syntax.
With icon addressing, the icon file on the OPC server is used.
The icons can either be entered manually (see Process server icons), or imported via icons.
See S7 icon import, Modbus icon import und AB ControlLogiX icons.
The icon item names that are defined are then indicated automatically (prerequisite: OPC client supports Alias Browsing).

OPC test client

The OPC Client that was installed together with the OPC server can be used for initial testing. Since this client is very easy to handle, you will be able to read out first data points immediately.
See OPC Client.

Diagnostics

The Online diagnosis is available for the diagnosis of the connections (access paths).

Logger

An external Logger is available to record the events which occur on the OPC server.

Server tray

A Server Tray is available to start and stop the OPC server and to start the NetCon OPC interface.

See also Overview and Accessories.

4.2 Licensing

A license is required to ensure continuous operation of the OPC server.
Without a valid license, the server will run 72 hours and then stop running.
There are two types of licenses.

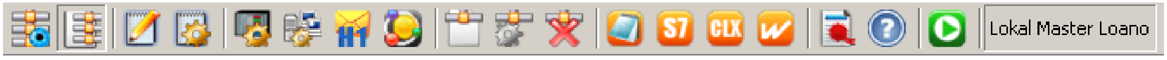
- Via a hardware dongle. The order number on the dongle and the type listed under **Help - Versions - Server Type** sollteshould agree, whereby the last digit of the order number of software and dongle is different. The function can be checked in the License overview dialog screen.
- Via a software activation code. The server indicates a request code which is sent to Softing Industrial Networks GmbH. We then supply an activation code which releases the server. The procedure is described in the chapters License overview and License entry.

The server tray monitors the server function cyclically. The license status is indicated in the icon of the server tray. The chapter entitled Server tray contains additional information.

Please adhere to the License conditions.

4.3 Tool Bar

The tool bar is located under Menu. It gives you quick access to frequently used functions.



A tool tip indicates the function when you move the mouse to an icon and stop there.

4.4 Screens

The program essentially consists of the following main screens:

- Online diagnosis
- Access path list
- Logger
- Configuration

Additional functions that are indicated in the dialogs are described in the chapters entitled Configuration and Dialogs.

4.4.1 Online Diagnosis

Online diagnosis of the server runtime is presented in the main screen. This screen shows the registered access paths (connections) with the corresponding diagnostic information. A colored identifier tells you immediately whether “everything’s just fine” or whether malfunctions are lurking. The main screen can be switched between the “Access Path Liste” (Access path list) and “Online Diagnose” (Online diagnosis).

The individual lines have the following meaning.

LINE 1

Access Path
Indicates the name and status of the connection. A colored envelop on red indicates an error.

Name	Name of the connection (Access Path)
Read	Status of the read access path.
Write	If write accesses are allowed via a connection, the status of the write connection is indicated. If read and write access via a connection is enabled, the status is the same as the read connection.
Event	Indicates the status of the event connection. This is possible for S5 and send/receive connections.

Colored envelope	
Green	The connection to the communication partner was correctly established and is okay. The connection is shown in black. Data have been received successfully.
Red	The connection to the communication partner is not okay. Either the connection has been configured incorrectly or the connection path is disconnected. The device may be off or the connection line is disconnected.

LINE 2

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

LINE 3

The individual data areas which the server requested from the PLC are listed starting with the third and subsequent lines. An appropriately colored envelop indicates errors.

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

Colored envelope	
Green	The area is registered (switched active) and data are being read.
Red	Reading this area causes an error. The error text is indicated.
Gray	All registered items for the area are switched inactive by the OPC client. The area is not read from the PLC.

4.4.2 Access Path List

The access path list of the server runtime appears on the main screen. It shows an overview of the configured access paths (i.e., connections) with the corresponding parameters. A colored identifier quickly shows you whether an access path is active (**black**), inactive (gray) or on simulation (**red**).
A double click allows you to process an access path. This requires master access.
The main screen can be switched between the Access Path List and Online Diagnosis.

A click on the header line of the table causes the table to be sorted by this column in ascending order.
One more click on the header line and the table is sorted in descending order.

The individual columns have the following meanings.

Column	Description
Type	The transport protocol that is used is indicated. **- TCP/IP** **- H1** **- Iso TCP (RFC1006)** If the connection is switched inactive, **(off)** appears behind the text and the text is shown in gray. If the connection is switched as simulation, **(sim)** appears after the text and the text is shown in red.
Access Path Name	The connection name corresponds to the “OPC Access Path” on the OPC client.
Job	The application protocol that is being used is indicated. **- S7** **- S5** **- Modbus** **- CLX** **- NetLink** **- MELSEC-Q** **- PLC-5** **- Send / Receive** **- OPCpipe Client** **- OPCpipe Server**
Network protocol	The network protocol that is being used and the type of connection setup from the server is indicated. **TCP/IP: Client** [active] / **Server** [passive] ' **H1: Akt** [active] / **Pass** [passive]
Target address	The address of the target hardware is indicated **(IP address, IP name** or **Ethernet address**) which can also be 0 with a passive connection establishment.
Parameters	Several of the parameters of the connection are specified here: - **Port number** of the read access connection - Transport protocol **TCP** or **UDP** - Own **TSAP**, someone else’s TSAP for H1 connections
Parameter details	Additional parameters of the connection are specified here: - Whether write accessing is permitted - **Write** - Value of the **Read interval** (poling interval) in ms - Optimized protocol - **HPP** - **OPCpipe** access: Read and write: **RW**; only read: **RO**

4.4.3 Logger

This screen offers a tool bar at the top that provides the following functions.

Set merker
Writes a flag text with the current time stamp in the list.
"-----------------"

Clear
Deletes the entries in working storage. If write to file is selected, the contents are written to the file beforehand.

Settings
Opens the Logger storage settings

AutoScroll
Updates the contents of the screen cyclically and always automatically indicates the last line with the latest entry.

Suspend
Logging is stopped with this button. This allows the user to diagnose the current content without entries which are located way back being overwritten.

NOTE:
No further entries are logged.

Close
Closes the logger dialog. This does not affect the logging of the events. Logging continues even when the screen is closed.

The screen also permits the selection of entries with the mouse or with the space bar and copying the selected entries to intermediate storage.
Execution of the function requires master access.
A **context menu** containing the following functions can then be opened with the right mouse button:

Copy
The selected entries are copied to intermediate storage where they can then be inserted into a program.

Clear Logger
Deletes the entries from working storage. When Write to file is selected, the contents are written to a file beforehand.

Clear Selection
All selected entries are deselected.

Which parts of communication are to be logged is set in Logger settings.
Logger Logger storage settings is used to set whether the information is to be saved in working storage or in a file.
For further information, go to the Logger service.

4.4.4 Configuration

The Configuration screen shows if NetCon OPC is started via **Start** - (All) **Programs** - **INAT** - **NetCon OPC** - **NetCon OPC**. If NetCon OPC is started via the server tray, the Online-Diagnosis is indicated directly and not this screen.

NetCon OPC is needed for the configuration of the INAT OPC server. It is used to make basic settings on the server, set up and diagnose connections and indicate the contents of the logger. Connection of NetCon OPC to ServerRuntime can be done in one of the following ways.

Configuration via TCP/IP
A socket connection to the server is established via TCP/IP. This can be a connection to the server that is running on the same PC or a server that is connected via the Ethernet network. The entires are transmitted directly to the server and stored there. This method permits the server to be configured from any point in the TCP/IP network.

Configuration via H1
The installation of the H1 protocol driver is required for configuration via H1. An H1 connection is established to the server that is to be configured. This can be a connection to a server on which another PC is running and is connected via the Ethernet network. A connection to a server which is running on the same PC is not possible. The entries are transmitted directly to the stations and stored there. This method allows the stations to be configured from any point in the H1 network.

Configuration offline
Offline configuration is not provided for the server.

Language selection (buttons with flags)
Buttons with flags can be used to switch to another language. After a NetCon OPC new start, all texts can then be indicated in the desired language. The texts that the logger indicates are not changed by this.

4.5 Connection (Access Path)

This chapter describes the dialogs that are used for handling the connections (access paths) of the server.
Additional dialog screens are described in the chapters Dialogs and Configuration.

The following dialogs are described here.

4.5.1 New Connection (Access Path)

OPC server – new connection
At least one connection on the OPC server must be set up for every piece of hardware which is to be accessed (i.e., with which the server is to communicate). The connection defines the access path to the hardware (i.e., the channel between the two communication stations – the access path). The number of TCP/IP connections is NOT restricted. The number of H1 connections is restricted to 64 although this can be increased to 200 with the registry entry "200H1Connections. reg."
Go to the menu **Connection** - **New Access Path** to set up a new connection. This dialog offers the following possible entries.

Connection Name
The name of the connection can be selected as desired except that only letters and numbers should be used. Special characters and blanks are not permitted because they may cause the connection to function incorrectly or to not be able to be addressed via OPC. The connection name which is used here will also be specified later when the access path is specified for the OPC client.

Connection type
This is where the type of connection is selected, whether a new **PLC** connection or a **OPCpipe** connection is to be set up.
The type **OPCpipe Server** or **OPCpipe Client** is also selected under the OPCpipe type. After confirmation with OK, the Network protocol dialog screen opens.

4.5.2 Network Protocol

Transport protocol

This is where the transport protocol is selected that is to handle communication on the Ethernet network.

- **TCP/IP** is included in the operating system

or

- **H1** which reguires an H1 protocol driver which has to be installed for this purpose. If the driver is not installed, H1 will not be available.
- TCP/IP should be selected for RFC1006 (ISO on TCP) communication.

Application protocol

The application protocol is selected here.

Application protocol	PLC
Send/Receive	Communication to other PLCs or devices which cannot handle any of the protocols listed below. User data is transmitted without protocol information.
S5 AP	Communication to Siemens S5 CPs or INAT echolink, echocollect or echochange.
S7	S7 protocol to Siemens S7-400, S7-300 or S7-200 automation systems, INAT echolink and echocollect
Modbus TCP	Communication with PLCs or devices which can handle the Modbus TCP protocol such as Wago, Beckhoff, Group Schneider, Modicon, and so on.
Netlink	IBHLink, NetLink
CLX - Ethernet/IP	Allen Bradley Control LogiX, Compact LogiX
Slc/Plc5	Allen Bradley PLC-5 und SLC
Melsec-Q	Mitsubishi Melsec-Q

OK

Confirmation with OK opens the screen OPC server connection.

4.5.3 OPC Server Connection

General

Enable cyclic read access and read interval
Cyclic read access is activated as the standard. The server reads the values of the variables from the hardware via a fetch active connection at a certain time interval. It can be specified in Leseintervall [in ms] (read interval) the time intervals at which the server is to send a read request to the PLC. The setting "0" means that it should be read as soon as possible. If "Zyklisches Lesen einschalten" (Enable cyclic read) is selected, read access is performed once when the client requests and no longer cyclically. However, this is only necessary in special cases and requires a precise knowledge of the communication.

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

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

Use Read (Events)
Read stands for receiving for Send/Receive connections.
Events from the PLC are only possible with S5 connections.
See also Events from the PLC.

PLC protocol settings
The button opens a further dialog screen in which specific parameters can be set for the following protocols:
Modbus protocol settings
CLX protocol settings
Melsec-Q protocol settings

TCP/IP communication

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

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

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

RFC1006
When RFC1006 is activated, H1 frames are packed into a TCP/IP frame for transportation. The "ISO on TCP (RFC1006)" button is clicked for the TSAP settings.

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

> **NOTE:**
>
> **The TSAPs must coincide crosswise before the connection can be established!**
>
> **The local TSAP of the one system must correspond to the remote TSAP of the other system.**

For further information on the TSAPs
TSAPs for S7 connections

Routing TSAPs
Routing is the capability of the OPC server to also address controllers even if they are outside the subnet boundaries. This requires special routing TSAPs.
For further information on routing TSAPs

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

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

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

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

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

Type active (client) / passive (server)
Here you specify whether the own station actively initiates the connection setup or passively expects the connection setup by the other station. You may not enter the same value on both sides of the connection. Usually, the PC with OPC server is parameterized as active. This is also used here as the default value.

TCP/UDP protocol
The TCP is a secure protocol with checksum and confirmation. UDP is not secure and is handled via datagram services. The TCP protocol should be selected for normal data transmission.

Osi/H1 communication

Destionation station: Ethernet address, card (adapter)
The Ethernet address of the target station is entered here. The target address consists of 6 bytes. In addition, the number of the Ethernet card is specified which is connected to the ISO (H1) network.

Local TSAP, remote TSAP
TSAPs (Transport Service Access Points) are addresses which are used within the transport layer to address applications. Both the local TSAP and the TSAP of the communication partner are specified to establish the connection. The following must be kept in mind.

NOTE:

The TSAPs must correspond crosswise before a connection can be set up!

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

For further information on the TSAPs
TSAPs for S7 connections

Connection setup type active/passive
This is where you specify whether the own station actively initiates the connection setup or passively expects the connection to be set up by the other station. The same value may not be entered for both sides. Usually the PC is parameterized as active. This is also the default value here.

Priority
The line priority can be from 0 (highest priority) to 4 (lowest priority). 0 and 1 are the so-called express priorities while 2 and 3 are the normal priorities. “Prio 4” is only seldom used because the connection is set up again for every send job. On the other hand, when seldom used, it requires less of the network than the other priorities because the line is not monitored (it is set up again after every send job). Remember though that, with the express priorities, the transmission is not faster than with the normal priorities. With some controllers, the data are transferred via interrupt to working storage even when priority 0 is used. That can result in faster total data transmission. With priority 0 and 1, the data cannot be longer than 16 bytes. Usually “Prio 3” is entered here.

Protocol
Specifies whether the frames of this connection are sent to everyone (broadcast), whether a certain group of stations is to be reached (multi-cast), whether a secure connection will be used (normal) or whether the data are to be transmitted unsecured (datagram). “Normal” is usually set here.

Expanded settings

Read optimization
Reading from the controller can be optimized here. If several variables are requested which are located in the same operand area (e.g., in a data block or in the flag area), the complete area is read from the first variable to the last variable without read optimization. Gaps between the individual block areas are also read no matter how large these gaps are. The maximum size of the gaps (in bytes) between the requested individual items within a data block without a new block being created is entered. If the gap equals the specified value, a new block is created.

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

Access over OPCpipe allowed
With this option, the access of an OPCpipe server permits connection to the connection configured here. This makes it possible to access a remote PC via this connection to the PLC. OPCpipe connections can be used to circumvent the time-consuming DCOM configuration and to avoid enabling the NetBIOS ports in the firewall. Only one, freely selectable port in the firewall needs to be enabled which is then specified for the OPCpipe connection.

Allow Write over OPCpipe
The write access is turned on or off with this option.

Simulate connection and data
If a real PLC is not available for test purposes, the option can be enabled to simulate a PLC. To the OPC client it looks as if a device were connected and the connection was working. Write accesses are also possible on the registered data points.

4.5.3.1 IP Address

The basics

In order to establish communication between two technical devices, each of the devices must be able to send data to the other device. To ensure that these data arrive at the right place on the other station, this station must be named (i.e., addressed) unambiguously. In IP networks this is done with an IP address.
An OPC server can address a PLC directly via its IP address (e.g., 192.168.1.20). A PLC can also be addressed via its name. This means that the name of the PLC must be stored with the applicable IP address and the domain server must be entered in the TCP/IP of the server. The server asks for a domain name (e.g., “PLC1”, the IP address for a name server) and then addresses the PLC directly via its IP address “192.168.1.20.”

IP address

Every IP data package begins with an information area for transportation through the IP layer – the IP header. This header also contains two fields in which the IP addresses of both the sender and the receiver are entered before the data package is sent. Relaying occurs in layer 3 of the OSI model – the relaying layer.

Setup

IPV4
The IPv4 addresses predominantly used since the introduction of version 4 of the Internet protocol consist of 32 bits (i.e., 4 octets – bytes). This means that 232 (i.e., 4,294,967,296) addresses can be represented. In dotted decimal notation, the 4 octets are written in decimal presentation as four whole numbers separated from each other by dots from 0 to 255.

Example: 192.168.1.20.

IPV6 – new version with larger address area
Due to the quickly increasing need for IP addresses, the day is coming when the usable address area of IPv4 will be exhausted.
The IANA address pool was used up on February 3, 2011. IPv6 was primarily developed for that reason. It uses 128 bits to store addresses which means that $2^{128} = 256^{16}$
(= 340.282.366.920.938.463.463.374.607.431.768.211.456 ˜ $3,4 \cdot 10^{38}$) addresses can be represented. This number is sufficient to provide at least 665.570.793.348.866.944
(= $6,65 \cdot 10^{17}$) IP addresses for every square millimeter of the earth’s surface.

Since the decimal representation ddd.ddd.ddd.ddd.ddd.ddd.ddd.ddd.ddd.ddd.ddd.ddd.ddd.ddd.ddd.ddd would be confusing and difficult to handle, we usually represent IPv6 addresses as hexadecimal numbers. To further simplify this representation, two octets each of the address are combined and presented in groups separated by colons. XXXX:XXXX:XXXX:XXXX: XXXX:XXXX:XXXX:XXXX.

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

To further shorten the presentation, zeros can be left off at the beginning of a block.

A sequence of blocks which only consist of zeros is replaced by :: but only once per IPV6 address.

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

When IPV6 is used, usually names are specified instead of the IP addresses.

4.5.3.2 Port

Port numbers are addresses which are used within the transport layer to address applications. Port numbers are required for TCP/UPD connections.
The port number is a 16-bit number from 0 to 65535.
Certain applications use port numbers which are permanently assigned to them by the IANA and are general knowledge. They are usually between 0 and 1023 and are known as “well known” ports. The registered ports are located between ports 1024 and 49151. If needed, makers of applications can have ports registered for their own protocols.
The remaining ports (49152 to 65535) are the so-called “Dynamic” and/or Private
Ports. The use of these ports is variable since they are not registered and therefore do not belong to any application.

4.5.3.3 TSAP

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

ISO (H1) uses so-called TSAPs in the transport layer to address the applications. These connection points are required for both ISO (H1) connections and RFC1006 connections.

Parameter for ISO (H1) Connections		**Parameter for RFC1006 Connections**	
Station A	***Station B***	***Station A***	***Station B***
Local TSAP A Remote TSAP B MAC address A	Local TSAP B Remote TSAP A MAC address B	Local TSAP A Remote TSAP B IP address A	Local TSAP B Remote TSAP A IP address B

Before a connection can be established, the TSAPs must correspond crosswise (i.e., the “local TSAP” of communication partner A must correspond to the “remote TSAP” of communication partner B). In contrast, the “remote TSAP” of station A must correspond to the “local TSAP” of station B.

Local TSAP (station A) = remote TSAP (station B)
Remote TSAP (station A) = local TSAP (station B)

This requirement is easy to adhere to if you select identical values for “local TSAP” and “remote TSAP.” Local TSAP and remote TSAP may be identical.
If several connections are set up between 2 stations, the TSAPs of the individual connections must be different.
The combination of TSAP + MAC address (or IP address) must be unambiguous so that a connection can be precisely specified.

Rules for entering the TSAPs for S7 connections

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

Rules for the entry of the TSAPs for S5 connections

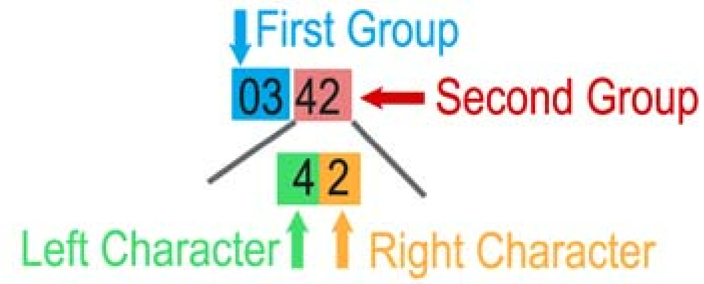
- TSAPs have a minimum length of 2 bytes and a maximum length of 8 bytes for TCP/IP and 16 bytes for H1.
- TSAPs can be entered either as hex or as ASCII characters.
- The TSAPs for S5 connections have no special meaning. They are selected at random.

Routing TSAPS

- Routing TSAPs are used for S7 connections to address controllers that are networked via MPI. See Routing TSAPs.

4.5.3.4 TSAP for S7 connections

So-called standard TSAPs are used for connections that cannot be parameterized (i.e., standard connections). The following rules apply to these.



First group
Contains device identifiers for those that are provided in S7 resources.
01: PG or PC
02: OS (operator or monitoring device)
03: Other (e.g., OPC server, Simatic S7 PLC)

Second group
Contains the addresses of these components
Left character (bit 7...4):
Rack number multiplied by 2
Right character (bit 3...0):
Slot (< 16) of the CPU. Slot 2 is always used for the S7-300.

The standard TSAPs **MUST** be used on the controller side (remote TSAP of the OPC server station).
Although the own TSAP (local TSAP of the OPC server station) can be selected as desired, its should have the same format. We recommend using TSAP 01 01 as the own TSAP.

Examples:
01 01 PC running on the server; direct addressing
03 43 OPC communication with the S7 CPU on rack 2, slot 3

03 02 OPC communication with the S7 CPU on rack 0, slot 2
03 2E OPC communication with the S7-CPU on rack 1, slot 14

NOTE:
The binary standard TSAPs are entered in the HEX window (left).

4.5.3.5 Routing TSAPs

Routing makes it possible for the OPC server controller to communicate over subnet boundaries. OPC communication is possible with all stations which can be reached over network crossings. It does not matter how many network crossings there are or how many networks are located between the Ethernet network (in which the OPC server station is located) and the target network. A connection only has to be parameterized on the OPC side for the fetch/write communication. It is not necessary to set up a connection on the S7. Establishment of the connection path requires that information on the target network and the target station be known. The following settings must be made for routing.

S7 routing – general information

The first device between network and MPI is echolink.
When an echolink is used to provide the connection between Industrial Ethernet and the subnetwork, this box should be activated. In this case, information on the routing PLC is necessary.

Routing PLC

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

Destination PLC

Settings for the target PLC are made here (i.e., the controller from which the values were read or to which the values are to be written).

S7 subnet ID
Each subnet on the S7 receives a subnet ID which is automatically generated by Step 7. The subnet ID of the target network must be specified here. For example, the S7 subnet ID can be determined as shown below.

- Open the "NetPro" screen in the Step 7 program.
- With the right mouse button click the network and the object properties.

MPI/PROFIBUS Address
If this is an MPI network, the MPI address of the PLC is entered here. If this is a PROFIBUS network, the PROFIBUS address is entered here.

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

Service
The "3" service is provided for OPC connections.

COM Port
The COM port of the echolink device via which echolink is connected with the MPI interface of the S7 PLC is selected here.

NOTE:
Some CPs have their own MPI address. This MPI address is automatically determined and assigned by the CPU. If such a CP is being used, you must be absolutely sure that

the MPI address of the CP is used and NOT the MPI address of the CPU.

Determining the MPI address of the routing PLC (Back plane MPI CP port or MPI address of the CPU)
The MPI address of the routing PLC is determined in “NetPro” by highlighting the station and then selecting "Object properties" with the right mouse button.
If a CP-MPI address appears here in addition to a CPU-MPI address, the MPI address of the CP should be used.

Another way is to highlight the CP in the hardware configuration, and then select "Objekteigenschaften" (Object properties) with the right mouse button. The MPI address to be used is then indicated under "Back plane connection".

Generate TSAPs
After all settings have been made, press the “TSAPs erstellen” (Create TSAPs) button. The TSAPs are then created and automatically entered in the two dialog screens.

4.5.3.6 Events from the PLC

In addition to cyclic read (fetch active) and write (write active) accesses, the server also supports events from the PLC. A channel is reserved which is used to deliver the spontaneous data from the PLC. The event channel is its own connection. Accordingly, a separate “port for events” (for TCP/IP connections) or a separate (remote) “TSAP for events” (for H1 and/or RFC1006 connections) is defined.

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

Events from the PLC are possible with the following connections.

- S5
- Send / Receive

Active / passive connection establishment for events
The type of connection establishment (active or passive) can be set for the “events from the PLC” mode can be set independently from the connection establishment for read and write connections.

In the example below, the connection establishment for read and write accesses is active while the connection establishment for “events” is passive

Reason:
Some CPs can only be parameterized for send jobs actively (required for „events from the PLC).

4.5.3.7 Modbus Protocol Settings

This dialog lets you change special parameters of a connection to Modbus controllers.

	Description	Default
Slave address	The slave address that is set is transmitted in the Modbus frame.	1
Start address 0	The start address is transmitted as specified in the frame.	Yes
Start address 1	Some Modbus devices expect a start address in the frame which is 1 less than the entered address. The first address is 1.	No

Use Byte Swab	Switch bytes within a word	Off
Use Word Swab	Reverse words within a double word (32 bits)	Off
Use Write Single Coil (05)	The Modbus opcode Write Single Coil 05 is used for write accesses instead of Write Multiple Coils 15.	Off
Use Write Single Register (06)	The Modbus opcode Write Single Register 06 is used for write accesses instead of Write Multiple Register 16.	Off
User Item Unit ID	The slave address is specified via the item syntax (e.g., Id2.R2). For more information, go to Modbus Item Syntax	1
IO Address is Octal	The start address of certain devices is octal. This means that the numbers 8 and 9 are invalid characters. This applies to the inputs and outputs. In addition, J is specified for inputs and P is specified for outputs in the item syntax.	Off

4.5.3.8 CLX Protocol Settings

This dialog screen lets you change special parameters of a connection to Allen Bradley ControlLogiX controllers.

Connection name
The name of the connection is indicated here.

CPU slot
The slot of the ControlLogiX CPU is specified here.

Browse filter masks
Some PLC programs contain a great many variables. These are created by PLC program code generators and are used internally. Not all tags available on the PLC are used by SCADA systems. Usually only a few of these variables are involved.
The start of the OPC server and the access of OPC clients to indicate the browsing list can take a very long time if there are many variables. This is particularly true when the Ethernet module of the PLC is not integrated in the CPU.

Setting the browser filter:
The CLX browser filters are entered in the input fields **Inclusive browser mask** and **Exclusive browser mask**. A combination is possible. The inclusive screen is evaluated first followed by the exclusive.

Inclusive browser mask:
Only the specified values are shown and available for OPC browsing.

Exclusive browser mask:
The specified variables are not shown.

If a mask is blank, it will be ignored.

A mask is a text which consists of one or more entries.

An entry without a dot "." refers to the variables in "controller tags."
Example: Variable1.

An asterisk "*" can be used at the end of the entry as the placeholder for any character string.
Example: Var*
This determines all variables which begin with the text "Var." These would also be: Variable1, variable2, VAR3, variance.

If "program tags" are to be addressed, an entry contains a dot ("."). The name of the subprogram is specified in front of the dot. The asterisk "*" can be used as a placeholder at the end. The variable

name is specified after the dot. The asterisk "*" can also be used behind that.
Example: Program:MainProgram.Var*
Example: Program:A*.*

A program name without a dot is ignored.

Individual entries are separated by a semicolon (";").

Upper case and lower case letters are ignored.

Blanks are not allowed.

Variable names may not contain a dot ("."), a semicolon (";") or an asterisk ("*"). If such characters are included anyway, a search cannot be made for them.

The maximum number of characters for a filter is 300 characters.

A more detailed example:
Var*;Prog*.subvar*
All variables in the "Controller Tags" list are determined that begin with the text "Var". In addition, all variables in all subprograms that begin with the program name "Prog" are searched for variables which begin with "subvar."

Variables which are filtered out during browsing cannot be addressed via OPC later on. The filter must be adjusted for that.

4.5.3.9 Melsec-Q Protocol Settings

The dialog permits you to change special parameters of a connection to Mitsubishi Melsec-Q controllers.

These parameters must be set when Mitsubishi Net is set up and the hardware supports it.

	Description	**Hex**	**Decimal**
Destination Module	Network address	0x3FF	1023
Monitor Timer	Time monitoring during which the answer should have been received.	0x100	256
Network number	Network number in the hardware configuration of the programming software.	0	0
PLC Number	An additional intermediate PLC number.	0xFF	255
PC Number	Is only applicable when rights management is set up. Access of the PC can be permitted or prohibited with this.	0xFF	255
Destination Station	The module address of an additional module in a PLC.	0	0

The effects of these values depend on the hardware that is being used.

4.5.4 Edit Connection

Name of the Connection
The connection name is indicated and can be changed here.

Connection Active
Indicates whether the connection is activated. If not, items cannot be registered and connections to the PLC cannot be established. This makes it possible to briefly shut down a connection without deleting it and having to enter it again later.

Simulation
Indicates whether the connection is marked as simulation. This function can be used to simulate connections although no communication with the PLC occurs. Items can be registered and write accessed. The server administers the data internally. The window frames light up red in this mode. A simulation connection appears in **“red”** in the screen Access Path List.

Button
Additional parameters can be processed depending on the type of connection.

- Server Settings opens the dialog screen OPC Server Connection
- Network Parameters opens the dialog screen TCP/IP Connect Parameters
- OPCpipe settings opens the dialog screen OPCpipe Settings

The number of available options varies depending on the type of connection.

4.5.4.1 Server settings

The button opens the dialog screen OPC Server Connection.

4.5.4.2 Network TCP/IP parameters

This dialog screen makes it possible to change the TCP/IP parameters.

Connection Name
The connection name can be selected as desired. In the “Verbindungsname” (Connection name) field all characters are allowed except the dot and the square parentheses. The connection name that is used here MUST also be used later when the Access Path is entered on the OPC client.

Destination IP-Address
The IP address of the target station (IP address of the PLC) or the DNS name of the target station is entered here. If the computer was configured for DNS and a DNS server is available in the network, the symbolic name of the target station can also be entered.
IP Address explains how an IP address is set up.

Destination Port
Port numbers are addresses which are used within the transport layer to address applications. TCP connections need port numbers. The port is a parallel to the TSAPs for RFC1006 and H1 connections. The port number is a 16-bit number from 1 to 65535. See the following for more information on setting up the connection

NOTE:

The port must be identical on both sides of the connection otherwise the connection cannot be established.

For further information on ports

Connection establishment (Type):
It is specified here whether the own station is to actively initiate the connection establishment or whether it passively expects the connection to be established by the target station. The same value

may not be entered on both sides of the connection.

client (Active)	Connection establishment is actively initiated.
Server (Passive)	Wait until the target station initiates the establishment of the connection.

Life Data Acks
When "Daten-Lebenstelegramme" (Data life frames) is enabled, user data frames with no data content (only the header) are transferred to keep a connection alive which is not cyclically used. Both communication partners must support this function. With OPCpipe communication, we recommend enabling this option.

4.5.4.3 OPCpipe Parameters

Connection Name
Each connection must have an unambiguous name so that it can be quickly identified later in the connection overview

OPCpipe recognizes several **Security settings**.

Password
The code word is used for access security while the connection is being established. It is transmitted over the network encrypted.

Connection Active
An access path can be switched inactive if it is not supposed to communicate with the PLC. This makes it possible to deactivate the connection temporarily without having to delete it and re-enter it again later.

Write allowed
If this is not set, the connection can only be read accessed.

Memory Limit
An OPCpipe frame can have up to 4 gigabytes. Such large frames are usually not needed. But since OPCpipe must provide memory space for such frames, this memory can be limited to prevent a slow down. The memory limit should be set so that the maximum frame length of an single frame will not be cut off by this setting.
If the runtime of the memory limit is exceeded, the data are thrown out and the communication partner is informed.

Use Send Acknowledge
OPCpipe recognizes the acknowledge mode with which the data that were read are confirmed with an acknowledgement. The data that were sent are also acknowledged even if this option is not enabled.
This mode slows down communication and should only be used in cases where it must be ensured that all received data have actually arrived.
This mode can only be used when the target station also supports acknowledgement mode.

4.5.5 Copy connection

A connection can be copied. All parameters (with the exception of the connection's name) of the highlighted connection are copied. The new connection name should be adjusted before the connection is saved. After it has been saved, the connection is indicated in the Access Path List.

4.5.6 Delete connection

The highlighted connection is deleted. This procedure cannot be reversed!
As an alternative to deletion, a connection can also be switched inactive (Switch on/off).
You can also select whether the symbols for this connection are to be retained or also deleted.

4.5.7 Switching a Connection Inactive

Below are the following ways that a connection can be switched inactive.

- Menu **Connection > Switch on/off**
- • Highlight the connection and select **Switch on/off** with the right mouse button.

The text **"off"** then appears after the type in the connection overview Access Path List.

4.6 Configuration

This chapter describes the dialog screens that are used to configure the server.
Additional dialog screens are described in the chapter entitled Dialog screens and Connection (Access Path).

The following dialog screens are described here.

- Server System Settings
- Logger Settings
- Logger Memory Settings
- Station Password
- H1 MAC Address
- Server Symbole Bearbeiten
- S7 Symbolimport
- AB ControlLogiX Symbole
- Modbus Symbolimport
- Verzeichnis für Konfigurationsdatei

4.6.1 Server System Settings

These dialog screens allow you to change the server settings. Some take effect immediately while others require a new start of the server.

Protocol Timer Tick

At the time intervals specified here the CPU releases computing time for the OPC server. The OPC server then checks to determine whether data need to be processed. The default setting can usually be retained.
If the utilization load is very high, this can be reduced by assigning a greater value in the timer protocol. This reduction is NOT possible if you are using the high performance protocol (in this mode, the server also takes CPU computing time between the timer ticks). So, if you are using the high performance protocol, you should use the default setting.

Configuration File Directory

The directory for the Configuration files is entered in this field or selected.
If the configuration software accesses the remote server and not the local one, this path cannot be selected.

Data Types

The representation of the two unsigned Data Types VT_UI2 and VT_UI4 can be set on the OPC interface.

Optimizations for Write

Write optimization takes effect differently depending whether the high performance protocol is enabled or not for the applicable access path.

Write Optimization	HPP = No	HPP = Yes
Related blocks always placed together (max.)	With multiple write accesses to an item, only the last value should be written.	No effect
Time sequence is retained.	Possible	If HPP=Yes, the time sequence should always be adhered to for all set write optimizations.
Prioritized write accesses, time sequence is retained.	Possible	Should not have any effect

Combine Coherent Blocks always (max)
This maximum form of optimization causes fewer PLC frames to be transmitted which reduces the load.
When a counter counts up very fast, the respective last value (or the current value at the moment) is written to the PLC.

Chronological Order is Preserved
The items that are written are arranged chronologically one behind the other and, after successfully read accessing of the applicable DBs, are written to the PLC.
Related blocks are placed together. Overlapping items are written separately. Individual bits of a word are also written individually. This ensures that a bit that can also be set and deleted very quickly and can also be forwarded to the PLC. This ensures the signal flow.

Write every Item separately
Every item is transmitted individually. Nothing is optimized. The procedure is the same as with "**Chronological Order is Preserved**".

Write with Priority, Chronological Order is Preserved

Write accesses are handled with a high priority.
A read access that is currently running is finished first, however. Afterwards, the write access is then initiated immediately. If several items of a block are to be write-accessed, these are written one after the other in the order that they were received.
The block is then read back so that the user receives an immediate response message.
If items of another block are still to be written, the write procedure is initiated without delay and continued as described above.
Typing via the keyboard or mouse is possible. This write optimization may somewhat delay the entire write procedure since the area that was written must also be read out again.

Write with Immediate Positive Confirm (default: No)
Waiting for the acknowledgement of a write procedure on the PLC may block the client which can be a nuisance. In this case, we recommend "Schreiben mit sofortiger positiver Quittung" (Write with immediate positive acknowledgement). When this optimization is selected, the user receives an immediate positive acknowledgement and not the actual result of the write access. The option can be selected to speed up synchronous write procedures on the PLC since you don’t have to wait for the acknowledgement. This setting is intended for clients that cannot select asynchronous write accesses.

Text format

S5 and S7 strings are organized differently. S5 strings do not have length information while the first two bytes of an S7 string contain this information.

S7 String:	**S5 String:**
Contains the maximum length in the 1st byte and the actual length in the 2nd byte.	Does not contain length information

Since the item syntax for S5 and S7 strings is identical (DBxSy.z), you set here whether the string format is to be interpreted as an S7 string or an S5 string.
It is standard to interpret the string as an S5 string with an S5 access path and an S7 string with an S7 access path. This setting can be changed to interpret every string as an S5 string.
The following notations are possible.
S7 syntax: db9.STRING0.10 or db9.s0.10
S5 syntax: db9s0.10
S7 string: db9g0.10 (always forces an S7 string)

Timeouts

Dialog Fields	
AppTimeout	Application timeout (AppTimeout) that begins running from when a read or write inquiry is sent to the PLC up to the time a response is received. The set time is multiplied by the factor (AppCount).
Factor (AppCount)	The set time of the logical timeout is multiplied by the factor (AppCount). The factor is 3 and cannot be changed. With Modbus connections with the user device number, the factor is always 2.
Conn Timeout	The timeout for the establishment of a connection to the PLC expires after the OPC item receives the BAD NOT CONNECTED quality. If the connection is established within the set time, the OPC client receives the GOOD quality for the registered item. The set time is multiplied by the factor (ConnCount). The procedure is switched off with the value 0. A value 1000 is recommended here.
Factor (ConnCount)	The set time of the connection time limit is multiplied by the factor (ConnCount) and this results in the total time.
Auto Logoff Timeout	Specifies the timeout value that expires when a user has established a connection to the server runtime with the NetCon OPC and does not actively use it. This permits another user to

	establish a NetCon OPC access path after the set time and to assume the master role and thus change the settings. The online diagnosis and the logger screen do not affect this timeout. A value=0 switches off the Auto Logoff and the master role continues to exist until the master selects **Log out station** or ends the NetCon OPC.

Specific settings

Echo Written Data back to Client

When a value written to the PLC from the outside is overwritten by the PLC or immediately reset again, it is up to the client as to which value appears in the visualization.

Echo *No* Written Data back to Client	Echo Written Data back to Client
If the button is not activated, the value that was read last is indicated (default). The button should not be set with the iFIX client.	If the button is activated, the value that was written last is indicated. The button should be set with the clients WinCC and WIZCON.

Use tag browsing

When an OPC server has a great number of data points in the icon table, these are transmitted to the OPC client on request. This requires working storage space and computing time. When the commissioning of OPC communication is concluded and this functionality is no longer needed, it can be switched off. This reduces the time that the server requires to boot after a new start.

Don't provide data outside the Poll Intervall

If you adapt the Group Update Rate com OPC Client to the set read interval for the Access Path Definition of a connection, this option can be enabled so that the server does not supply any data changes which arrive in less time than the set read interval. This option is usually off.

Wait for browse ready with item create

After program start, the tags of a CLX connection are compared with the online PLC. Under some circumstances, this can take quite a while. If data points are registered for such a connection, this may cause the server to output an error message saying that the item is not available. This option can be enabled to keep this from happening. The registration of an item blocks the OPC client until CLX browsing has been concluded.

4.6.2 Logger Settings

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

Errors

Dialog Fields	
Errors	Any type of error should be indicated on the logger. Exception: PLC errors.
PLC errors	If an area does not exist on the PLC, or an area is requested that is too small, a corresponding entry is indicated on the logger. This also applies to all other errors which are generated by the PLC.

Status

Reparameterization of the Access Path and the logging in for reparameterization are indicated on the logger.

PLC inquiries

The parameters of the applicable PLC inquiries from the server to the PLC are indicated on the logger.

Dialog Fields	
Send	Write to the PLC
Receive	Read from the PLC
Event	Event connection for the S5 protocol and read for send/receive connection.

OPC data

The data between OPC client and server are indicated on the logger. The direction of the data can be selected.

Dialog Fields	
Sent data	Values written to the PLC
Received data	Values read from the PLC
Event data	Values for event connection of the S5 protocol and values for read accessing the send/receive connection

Special information

Special events can be enabled here for indication on the logger.

Dialog Fields	
Symbols	The list of symbols is indicated on the logger when the symbols are read in.
Access Path	When an access path to the PLC is set up or disconnected, the logger indicates the parameters.
Create Items	When data points (items, tags) are registered or de-registered with the OPC server by the OPC client, the logger indicates this.
Activate items	When registered data points (items, tags) are switched active or inactive on the OPC server by the OPC client, the logger indicates this.
Statistics	Additional entries are indicated on the logger which are not of great interest during normal operation of the server. It should be enabled for trouble-shooting.

4.6.3 Logger Memory Settings

In the logger memory dialog, it is specified whether the log entries should only be stored in working storage or also in a file. The time of the storage, the directory and the name of the file in which the logs are saved is set.

Cache size
Specifies how much working storage is to be used for logging data. This buffer functions as a ring buffer. When expanded, this means that more entries can be stored intermediately in working storage.

Use file logging
When this option is selected, the information is written to a file.

File settings

Max disk space
Specifies the size of the hard drive after which the old files should be deleted. This is supposed to prevent the hard drive from overflowing.

> NOTE:
> **When the set size is reached, the logger deletes files from the directory without asking. If files from this directory are still needed, they should be archived early enough beforehand.**

Directory
Specifies the directory in which the log entries are saved.

File prefix
The file name is created from a consecutive number and the date plus time. The prefix is added in front.

Save every ...
Specifies the time at which the file is to be saved if the buffer has not run in a circle in working storage. If the buffer is running faster in a circle, information is saved more often.

> NOTE:
> **The number of write procedures is limited when saving to FLASH drives. To increase the lifespan of FLASH drives, write accesses should not be used often.**

New file every ...
Specifies when a new file is to be started. This simplifies the evaluation.

File Name Example
The output field shows an example of a file. This specifies where the logging files are located and how the file names look.

4.6.4 Station Password

The dialog screen for the creation or deletion of a code word is opened. This dialog can be used to specify whether the user can change the server parameters after the program starts or whether the user must identify him/herself with the code word first.

4.6.5 H1 MAC Address

The dialog screen shows the status of the H1 protocol driver. It allows you to set the MAC address of the network card.

The dialog screen is not offered unless the H1 protocol drive is installed.

The status of the network card can be determined in this dialog field and, if necessary, the current runtime address can be changed.

H1 license

This field indicates whether the installation of the H1 driver was successful.

Set MAC Address possible
The MAC address of the network card to which the H1 protocol driver is linked cannot be overwritten during runtime (default setting). This setting can be changed with the registry entry “H1ChangeAddress. reg.”
Select: **“H1ChangeAddress.reg”** via **Start** > **Programs** > **INAT** > **PC-H1** > **H1 Registry Scripts**.
The confirmation prompt should be confirmed with “Ja” (Yes). The change is confirmed with a message. It is now possible to change the MAC address of the network card during runtime. It should be considered that although the operating system can be changed with the setting under which the MAC address can be changed, this may slow down the operating system.

NOTE:
The PC must be started again after the registry entry is changed!

Set MAC Address not possible
The registry entry "H1ChangeAddress.reg" can be cancelled again with this registry entry “H1DoNotChangeAddress.reg."
To do this, select "**H1DoNotChangeAddress.reg**" via **Start** > **Programs** > **INAT** > **PC-H1** > **H1 Registry Scripts**. The confirmation query should be answered with “Ja” (Yes).
The change is confirmed with a message. It is now no longer possible to change the MAC address of the network card during runtime.

NOTE:
The PC must be started again after the registry entry is changed!

MAC Address

The current MAC address of runtime is indicated here. To link the computer to an existing network, it may be necessary to adjust the MAC address on the software side to the address structure used by the network. The desired Ethernet address is entered in the MAC address dialog field and changed via the button “<- Stellen” (set).
The MAC address of the network card to which the H1 protocol driver is linked cannot be overwritten during runtime (default setting). This setting can be changed with the registry entry **"H1ChangeAddress.reg"**.

ROM Address

The hardware or Ethernet address of the network card is stored on a ROM chip on the network card of the PC. It is read out and indicated at this location.

<- Set

The address that was entered in the MAC address field becomes active immediately with the "**<- Set**" button.

Set Addresses at server start

By activating the option "Set Addresses at server start" the address is changed the next time the program starts.

NOTE:
A running connection is terminated when a new address is set.

NOTE:
The Ethernet address may only be assigned once in the network.

4.6.6 Server Symbol Edit

The server offers the possibility of using a symbolic name for a certain direct address on the PLC. These symbolic names are stored in the icon file in the configuration directory with the name TcpIpH1.txt. This file can also be processed with any text editor.

The dialog **Server Symbol Edit** offers a convenient way to process the icons.

New access path

Existing access path names of the icon file are listed in the drop-down box. One of these can be selected or a new name can be entered.

Edit topic symbol

This field contains the symbolic names followed by an equals sign ("=") and the PLC address.
Example for the S7:

```
Motor1.Auto_Ein=db5.X0.0
Motor1.Drehzahl=db5.W2
```

Search / Replace

The lower area permits the search for/replacement of:
- Individual character strings or whole words
- Cursor backwards or forwards
- Differentiate or not between upper and lower case letters
- All of them or single ones

Save
Saves the icon information that was changed.

Delete
Delete all icons of the access path selected above.

End
Leaves the dialog and requests a new start so that the changes can take effect.

NOTE:
Symbols CANNOT be processed for CLX controllers!!!

The icon editor saves the icons in Unix file format. There is only a carriage return at the end of a line and not a line feed. Some text editors cannot handle this and indicate all the lines in a single line. This can be corrected with some text editors and converted into a PC form (e.g., Notepad++).

NOTE:
An icon import is available for S7 and Modbus controllers (Wago). This saves your having to type in the information several times. For more information go to S7 Symbol Import and Modbus Symbol Import.

4.6.7 S7 Symbol Import

The S7 icon import dialog screen can be used to import icons from a Step7 program to the icon table of the server. During OPC client browsing the icon names are indicated the way they are also used in Step7. The following steps are required.

- In the menu Symbols - S7 Symbol Import
- First, select the Step7 project with the file ending s7p.
- Then select the PLC program in the Step7 project.
- The list of programmed DBs is indicated. Or the icon table can be displayed. To do this, select the **Show All DBs** option or **Show Symbol Table**.
- Then select the Access Path from the list of available ones or enter one in the input field New access path.
- You can now select which symbol you want to import:

• Symbol Table
All symbolic names which are stored in the icon table of the S7 program are imported. These include the symbols for inputs, outputs and symbols in the flag area.
• DBs
The symbolic variable names of the data blocks of the S7 program are imported. In addition, only those blocks can be selected from the PLC blocks list that are also supposed to be visible on the server.
• Resolve arrays in addition
Every single element of an array is given its own icon name. **NOTE:** **This may greatly enlarge the icon file of the OPC server and this takes more time during program start.** It is usually switched off.
• Add comments
When the project contains commentaries for individual icons, these commentaries are also visible in the icon table although they cannot be registered as items. This is usually switched off. **Example** Motor1.EIN=M1.1 (Motor1.ON=M1.1) Motor1.EIN-comment (Motor1.ON-comment)=(=Motor1 is switched on with this flag). **NOTE:** **The commentaries should not be registered since this would cause an error message.**

• PLC Blocks

> Here is a selection of individual data blocks which can be selected to be imported. With large PLC projects, we recommend only importing those DBs that are really required on the server. The button **Import selected** uses this selection.

- The import is started with the button **Import all** or **Import selected**.

The Step7 icons are imported to the icon file of the OPC server ("TcpIpH1.txt") and are now available with the browsing functions of your OPC client.

4.6.8 AB ControlLogiX Symbol Import

As soon as the connection between the OPC server and the CLX has been made, the icon information can be read in from the CLX and is now available in the icon display of the server. These icons are then used on the OPC client as Item IDs.

Since importation of icons can be time-consuming, the server saves the icon information in the file TCPIPH1.CLX. When the program starts, the server compares already existing icon information with the icon information from the PLC. If the root icons of the global tags (controller tags) and the root icons of the program tags have not changed, a new icon import does NOT take place.

If the PLC program in the controller was changed while tags are registered, the server finds out about this, closes the connection to the PLC, reports BAD QUALITY for the items and enters the message in the logger:"The PLC program [.] has changed. We strongly recommend starting the server again!". The symbols are imported again during the new start.

NOTE:
There are also types of PLCs for which the server does not automatically recognize a change. In such cases, you must take action manually. Select an entry from the list of existing Access Paths and then press the button Refresh Symbols.

4.6.9 Modbus Symbol Import

The icons from the CoDeSys programming software can be imported to the server for **Wago** PLCs.
To do this, proceed as shown below.

The variable must be assigned an address before the variables can be addressed via Modbus. The dialog screen **"Variable declaration"** contains a field called **"Address"**. The variables can also be entered directly as shown below. The result looks like this:

```
(*Modbus Samples*)
    My_BOOL_MX0_0      AT %MX0.0 : BOOL;
    My_WORD            AT %MW1   : WORD;
    My_DateAndTime     AT %MW2   : DATE_AND_TIME;
    My_DWORD           AT %MD4   : UDINT;
    My_Real            AT %MD6   : REAL;
    My_Output_Q0_0     AT %QX0.0 : BOOL;
    My_Intput_I0_0     AT %IX0.0 : BOOL;
```

The following settings must be made in CoDeSys before the symbol file can be created.

- "Create symbol entries" should be enabled in the menu "Project" - "Options" in the catagory "Symbol configuration".
- "Output variables of the object" should be enabled in the dialog screen "Set

object attributes" which can be accessed via the button "Configure symbols".

After the project is compiled, a file with the project name and the ending “.SYM” is created. The server needs to import the icons.

The dialog screen is called in the menu "Symbols" - "Modbus Symbol Import".

- The *.SYM file is selected first.
- The access path is then selected to which the icons are to be imported. The name should be located in the "New access path" field. If the connection has not yet been set up, the new name can be specified here.
- The icons are imported when the "Import" button is pressed.

The server must be started again before the icons become available in Browsing.

A text can be added at the beginning or end of all icon names with the "Prefix" and "Postfix" fields. Under normal circumstances, these fields remain empty.

4.6.10 Directory for Configuration File

All established connections are stored automatically in a configuration file. The standard path for the configuration file is under Windows XP "**c:\Dokumente und Einstellungen\All Users\Dokumente\INAT**." And for later editions of Windows "**c: \Users\Public\Documents\INAT**" and the file name **TcpIpH1.NET**. The path is automatically stored in the **OpcServer.ini** file which is located in the above stated path so that the configuration file is loaded the next time the server starts and the connections contained therein are available. The icons are also stored in the same directory in the file **TcpIpH1.txt**. The intermediately stored icons for the CLX connections are stored in the file **TcpIpH1.clx**.

NOTE:
The file name of the configuration file CANNOT be changed. It must have the name Tcpiph1.net.

Changing the directory of the configuration file

The path for the configuration file can be entered in the dialog screen OPC Server System Settings in the input field **Path for the configuration file**, or via the >> button to search through and select. To accept the change, the server must be started again.

Saving several configuration files

There is no limit to the number of configuration files that can be created except that the files must be located in different directories.

Starting the server with another configuration file

This is not possible unless the server is registered as an application!
If the server is to be started with a different configuration file than the one saved last, the following command line should be entered in the request for input.
TCPIPH1.EXE /d:path
Remember:

- A blank between.EXE and /d.
- The path is the complete, asbolute or relative path of the configuration file.
- If it contains blanks, the path should be entered between quotation marks.

Example:
TCPIPH1.EXE /d:C:\Data\Project1
if the configuration file is located under C:\Data\Project1 or is to be set up here anew.

The applicable path (e.g., C:\Data\Projekt1) is then located as the directory for the configuration file in the menu **Settings** > **Server System Settings** in the dialog screen OPC Server System Settings.

4.7 Dialog Screens

This chapter describes the dialog screens which were not described in the chapters Configuration or Connection (Access Path).
The following dialog screens are described.

- Network Settings (Station Parameters)
- Versions
- License Overview
- License entry
- About
- Entering the Password
- Selecting the Station
- New Station
- Station not found

4.7.1 Network Settings (Station Parameters)

The dialog screen shows the information on the own station.
The values cannot be changed. The dialog screen shows the information that has been determined.

Station Name
Computer name of the PC

MAC (Ethernet) Address
The MAC address of the network card. Each station in the Ethernet network has an unambiguous Ethernet address.

TCP/IP Station Parameter:

Use DHCP
It cannot be precisely determined whether the automatic configuration is switched on via a DHCP.

IP Address
This determines the address of the station when used with the TCP/IP protocol. See IP Addresses.

Subnet mask
This determines which station inquiries are allowed to reach the active network. Addresses whose masked portion differs are sent to the router. If no router is parameterized, the subnet mask is meaningless.

Domain Name
The domain name is used for symbolic inquiries when no domain is explicitly specified there. That is usually the case with names which do not contain a dot.

Maximum three Domain Server Addresses:

DNS Addresses
These addresses determine the servers which resolve the symbolic Internet names into IP number addresses in their network

Router Addresses

Routers are used when you also have to communicate with stations outside your own network.

4.7.2 Versions

The dialog screen shows the versions of the **Server runtime** and all its individual components.
The contents can be copied to intermediate storage by pressing the **"Clipboard button"** and inserted again for Email or text processing.
These versions are always required if you have questions on the product or need support.
NetCon OPC versions in the dialog screen About are also helpful.

4.7.3 License Overview

The list of available licenses with order numbers and status are shown.

Show License
An entry can be processed with a double click or by pressing the button (Please select the desired License previously). The dialog screen License entry.
Master access rights are required before you can process licenses.

4.7.4 License entry

The dialog screen License Overview for the license of the server is opened with the aid of the menu **Help** - **Device Licenses**.

Via activation code:

Adhere to the following if you need to obtain a product license.

- The information under product and order number should agree with the desired ordering data. If not, change the program server type under **Start** - **Programs** - **INAT** - **OPC Server** - **Tools** - **Server type**. After the server type has been changed, a new start must be performed by the server.
- Enter the company name and the person to contact so that the license is unambiguous.
- Enter the activation code which you will find on the delivery slip.
- Then press the **-> Clipboard** button. The data required for requesting the license are located in intermediate storage ready to be inserted in an Email or a text processing program. An example is shown below.

```
-------------------------------------------
Product       : OPC-Server TCPIPH1
Order No      : 100-3100-01
Customer      : Softing Industrial Networks GmbH
User          : Dipl. Inform. Thomas Muster
Authorization : 123456789012345
Request       : 2E2380H0CH8SFMH
Confirm       :
===========================================
```

NOTE:
The request code must be unambiguous for the system (the PC)!

- Send this data via fax to 0911/54427-27 or Email to info-in@softing.com.
- Softing Industrial Networks GmbH will send you a license activation code.
- This activation code must then be entered in the Lizenz Freischaltcode (License activation code) field.

- The **License status** field indicates whether the entry is correct and the function has been activated.

NOTE:
When used with RAID systems or Cluster computers, licensing is only possible via dongle!

4.7.5 About

This dialog screen shows the copyright, the versions of software and the addresses to contact if you have questions, ideas or need support.
These versions are required if you have questions or need product support. The versions of the Server-Runtime are also required. They are shown in the Versions dialog screen.

Licensing conditions

4.7.6 Entering the Password

This dialog screen allows you to set or delete the code word. The screen is also called up during registration.
A code word provides protection against someone changing the configuration.

The dialog screen is called for this in the menu **Station** - **Station Password**. The password is entered and confirmed a second time. This means that the Netcon OPC is logged in and the configuration can be changed.

The menu **Station** - **Station Logoff** disables the right to perform write accesses. This means that only those dialog screens can be opened that do not require write accessing. Online diagnostics and logger are still possible, however.

The status field next to the toolbar under the menu shows whether the parameters can be changed.

4.7.7 Selecting the Stations

When the NetCon OPC is started via Start -> Programs) ->INAT -> OPC Server -> NetCon OPC, the stations are selected which are to be communicated with. When the NetCon OPC is started via the Server Tray, this dialog capability is not available.

The station screen **Select station** is opened during the **Configuration via TCP/IP** or **Configuration via H1**.

All stations with an arrow (->) at the beginning of the line are stations which are currently available online on the network.
They are automatically recognized by the NetCon OPC. All PCs with servers are shown.

NOTE:
Stations behind routers are not automatically recognized. A direct connection must be established to them via the " New" button.

Select station
A connection to the appropriate station is established by double clicking on the desired station (or confirming with the “OK” button). This connection establishment is time-monitored at approx. 3 sec. Connection is made: The main screen Access Path List is opened.
Connection is not made: If no reply is received within 3 sec, connection establishment is terminated.

New
If the station was not found in the automatic online indication (parameterization via IP) because it is located outside its own network, a direct connection to the station can be configured with this button. This is done in the New Station dialog screen.

Edit
The parameters of the station which was created with the dialog New Station can be changed here.

Delete
The parameters of the station can be deleted here. An "are you sure?" question appears beforehand. **Remember that deletions cannot be reversed**.

Scan for stations
Starts the station search and the automatic recognition of the stations and server online. Remember that only stations in the local network can be reached.

Execute station scan on start
Starts the search for stations automatically when this dialog screen is opened. Since this can take a long time if there are many stations in the network, this function can be switched off.

Use Timeout
This button should be switched on when the connection timeout is used. Normally this button is off, however.

Timeout
The connection timeout can be used to change the default timeout for a station search of 3 sec specifically for this connection. For example, this is useful for connection on the Internet for which the life frames have been deactivated. This increases the reaction time for connection interruptions (e.g., for cable breaks).

4.7.8 New Station

If a connection could not be established with the desired station and the dialog screen Station not found appears, a direct connection can be configured to this station. The entry is then added to the list of available stations and the entry appears in the dialog screen Select station. If the station is available online, this is indicated with an arrow (**->**) preceding the station when the station search was performed.

Name
The name that is to be indicated in the list.

H1
A connection is to be established via the H1 protocol. MAC address and TSAP are specified.

TCP/IP
A connection is to be established via the TCP/IP protocol. IP address and port number are indicated.

Extended
Expands the dialog for entry of additional parameters.

H1 Settings

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

Default TSAP / Special TSAP
The TSAP for H1 or RFC1006 connections.

TCP/IP Settings

IP Address
The IP address or DNS name of the station which is to be configured and diagnosed.
127.0.0.1 addresses the local device.

Default Port / Special Port
The OPC server uses port **982**.
The INAT devices use **997**, newer devices also use **982**.
If configuration and diagnosis is to take place via RFC1006, the TSAP must be entered.

Standard
Reduces the dialog and only shows just a few parameters.

Optional Product Select

These entries are ignored.

4.7.9 Station not found

The dialog screen shows "**A timeout occured**".
A station may not be able to be found for the following reasons.

OPC-Server

- The server service is concluded

Configuration via TCP/IP or Configuration via H1

- The selected station is off.
- A network cable is not plugged in or is defective.
- A switch, HUB, router or gateway is not on.
- The station cannot be reached because it is located behind a router.
- The protocol which is to be used for the connection with the station (H1 or TCP/IP) is not correctly installed or configured on the operator computer.
- H1 or TCP/IP is not correctly installed or configured for the other network station.
- A station (e.g., a reference router) has blocked further transmission of the frames
- The firewall is not configured correctly.

NOTE:
Stations behind routers are not automatically recognized. A direct connection must be established to them via the “Neu” (New) button (see New Station)

4.8 Menu

This menu offers the following menu items.

- File
- Connection
- Diagnostics
- Station
- Settings
- Symbols
- Help

4.8.1 File

This menu item offers the following points.

Refresh

The connection to the OPC server runtime is established again. If a code word is set, it will need to be entered again before you can begin processing.

Restart Server

The server runtime is started again after a security check.
After a new server start, the data points (items) must be registered again on the OPC client.
This function requires master access.

Close Window

Concludes the configuration of the server.

4.8.2 Connection

This menu item has the following points.

Show Access Path List

The main screen is switched from Online Diagnosis to Access Path List. The list with the configured connections is indicated and can be processed.

Switch on/off

An access path can be switched inactive if it is not supposed to communicate with the PLC. This means that it can be temporarily turned off without having to delete and reenter.
See Switch connection inactive.
This function requires master access.

New Access Path

A new access path can be entered and then the dialog screen New Connection (Access Path) opens. First, the name and type of connection are selected.
Then the necessary parameters are entered.
The connection parameters take immediate effect for most connections. However, a server new start may be necessary for some of them. An appropriate dialog screen points this out.
This function requires master access.

Edit Access Path

The parameters of the selected access path can be processed. The dialog screen Edit Connection (Edit Access Path) opens.
Master access is required for this function.

Copy Access Path

The access path can be copied. The dialog screen Copy Connection (Copy Access Path) opens.
When saving, remember that the name must be adjusted accordingly.
Master access is required for this function.

Delete Access Path

The access path can be deleted with the dialog screen Delete Connection (Delete Access Path) after a security check. You can also decide whether the related icons are to be deleted too. Master access is required for this function.

4.8.3 Diagnostics

The menu item offers the following points.

Show Diagnostics

The main screen is switched from the Access Path List to the Online Diagnosis. The list with the running access paths (connections) is indicated.
Colored identification makes it easy to see whether “everything is okay” or whether problems have occurred.
Statistics concerning sent and received frames and data points (tags, items) are indicated for every access path.
The requested data areas of the PLCs that reported the number and have active data points are also indicated.

Logger

The Logger screen is opened. The logged messages are indicated there.
This screen can be opened parallel to the rest of operator control of the program. When the logger screen is open, you can continue configuring and other operations.

Logger Settings

The dialog screen Logger Settings is opened. There you can select which events are to be recorded on the logger.
The file and memory settings of the logger are available under Settings in the logger screen.

4.8.4 Station

The menu item has the following sub items.

Station Login

The dialog screen Entering the Password for entering the code word is opened. Here you can log in and obtain write access rights for changing the server parameters.

Station Logoff

This is where you can log out and cancel access rights for changing the server parameters. Another station can then log in and change the server parameters.

Station Password

The dialog screen Entering the Password for creating or deleting a code word is opened. Here you can specify whether the user can change the server parameters after program start or whether he must register with the code word first.

H1 Address

The dialog screen H1 MAC Addresses for indicating and changing the H1 MAC address is opened. There the MAC address can be indicated, changed and highlighted for setting during the server start. Master access is required for executing this function with write access rights. The menu item is then only offered when the H1 protocol driver is installed.

4.8.5 Settings

This menu item has the following sub items.

Server System Settings

The dialog screen Server System Settings is opened and enables you to change the server settings.

DCOM Settings

A management console of the operating system is started for the administration of the component services.
The administration program for component services can be used to configure and manage the COM components and COM+ applications. If you don't have access rights, the DCOMCNFG.exe program can also be started directly from the workplace interface.

4.8.6 Symbols

This menu item has the following sub items.

Symbol Edit

The dialog screen Server Symbol Edit is opened. The symbol editor can be used to assign icon names directly to PLC addresses.
These then also appear in the browsing interface for the OPC client.
Master access is required for editing the icons.

S7 Symbol Import

The dialog screen S7 Symbol Import is opened. This screen can be used to import the variables of a STEP7 PLC program to the server and assign an access path. These are then available to the OPC client on the browsing interface.
Master access is required for the execution of this function.

CLX Symbol Import

The dialog screen AB ControlLogiX Symbol Import is opened. This screen can be used to delete the intermediately stored information of the variables of a PLC program and reload.
This must be done when the PLC program has been changed.
Master access is required for the execution of this function.

Modbus Symbol Import

The dialog screen Modbus Symbol Import is opened. This screen can be used to import the variables of a CoDeSys PLC program for a Wago PLC to the server and assign an access path. These are then available to the OPC client on the browsing interface.
Master access is required for the execution of this function.

4.8.7 Help

This menu item has the following sub items.

Help

Starts the Help Overview of the online help.

Versions

Opens the Versions dialog screen and shows the versions of the server runtime.

Device Licenses

Opens the License Overview dialog screen and lets you monitor and process the server licenses.

About

Opens the About dialog screen and indicates the manufacturer and the versions of the program.

4.9 Troubleshooting

Problem: Access of the OPC server to the S7 controller doesn’t work.

Solution: First, ping the S7 controller. If you don’t receive an answer, there is a network problem. Or you have used the wrong IP address or a router is located in the network and the gateway address is not set at all or is set incorrectly. If pinging is possible, check the following Access Path Einstellungen (access path settings):

1. SPS-Header (PLC header) must be deactivated.
2. RFC1006 should always be activated. Addressing is done with TSAPs and not via ports.
3. The easiest way to access the S7 is to use standard TSAPs.

NOTE:
Be sure that the standard TSAPs are entered in the HEX window!

Problem: After 72 hours, the OPC server doesn’t supply any more new values for the items.

Solution: The server is still not licensed and the demo time has expired. Start the server again and all the functions of the server will be available to you again for 72 hours. License the server with the hardware key (dongle) to eliminate this time limitation.

Problem: The demo time of 72 hours is not reached.

Solution: The system time of the PC was probably changed. They can be caused for the following reasons.

- Some programs change the system time because they are not compatible with the operating system (e.g., STEP5 V6.x under Win2000 – this combination has not been released).
- Some programs which are actually supposed to synchronize the system time on the network, change the system time for a minimal amount of time. The user usually doesn’t notice this.

Check the logger for how long the server has actually run.

Problem: The OPC server “freezes up” and can no longer be operated.

Solution: Probably very many or all logger functions are activated on the OPC server logger. With some PCs, this can cause the server to freeze up. The logger functions should then be switched off with the Logger Settings dialog screen (default is if only "error" and "PLC error" are activated and the server starts running again).

Problem: On the NetCon OPC, the user is requested to enter a code word although none is set.

Solution:

- The NetCon OPC was probably started several times. Only a NetCon OPC can be the master. All the others are slaves and only have write access rights after the code word is entered.
- A value other than 0 has been entered for the Auto Logoff Timeout in the Server System Settings dialog screen. After this set time, the NetCon OPC loses its master role if it is not actively operated

Chapter 5

Accessories

5 Accessories

This chapter describes the following subjects.

- NetCon OPC
- Selecting the Server Type
- Server Tray
- Logger
- OPC Client

5.1 NetCon OPC

The **NetCon OPC** program is the graphic interface (GUI) for configuring and diagnosing (starting with the version 4.05 OPC server). GUI connects to the OPC server core via TCP/IP (Port 982). This allows remote access to OPC servers which are running on another PC and can be accessed on the network.

Program start

NetCon OPC can be started in different ways. It provides different functionalities depending on how it is started.

- Via the **Server Tray**
  Usually the program is started via the Server Tray, menu item **Server Status - NetCon OPC**.
  The local OPC server is configured with this.
  The program starts with the Online-Diagnosis.
- Via **Start - Programs - INAT - OPC Server - NetCon OPC**
  If an OPC server is to be configured on another PC and diagnosed, this is the method to use. The program starts with the start screen and permits configuration via TCP/IP or a configuration via H1 (if the H1 protocol driver is installed). This method can then be used to open the station screen Selecting the Station and the connection can be established to the desired server. If you select this method, the program is started with the command line parameter **-s** which means that only the server in the network is offered for selection.
- In Explorer, double-click ParamqInt.exe.
  Starting the program with this method causes all OPC servers and INAT devices (echolink, echocollect and echochange) to be indicated in the station screen Selection the Station. This is not the usual method.

Password

Changing server parameters and settings can be protected by setting a code word. This is done in the menu **Station – Station Password**. If a password is set, this can be registered via the menu **Station - Station Login** and deregistertered via **Station – Station Logoff**. Monitoring and diagnosing are also possible without logging in.
When the server is installed again, it should not have a code word (i.e., write accesses are also permitted from Remote).
If you forget the code word, the only thing to do is to locally access TCPIPH1.Net with a text editor and delete the code word in section [Own address] of the entry [station password =...] and then perform a new start of the server. If the server refuses to perform a new start due to the missing access rights, the PC must perform a new start.

Master-Slave

The OPC server allows accessing by several NetCon OPCs. To prevent them from influencing each other, access is handled by a master-slave procedure.
The registration procedure is performed as shown below.

- If no NetCon OPC is registered yet, the first one registers. If a code word is set, then as slave or otherwise master.

- All others register as slaves.
- If a code word is set, the code word is queried during the first write command.
- If the code word is entered correctly, it is no longer queried after that.
- If this is a slave, it can register itself as master via the menu **Station** - **Station Login**. It can do this then when the master has de-registered itself in the meantime.
- remote NetCon OPC cannot take the master role away from anyone.
- A local NetCon OPC can withdraw the master role from a remote master and become master itself. The remote master then becomes the slave. However, it does not notice this until the first write access when it has to enter a code word.
- When a connection is briefly interrupted, the code word does not have to be entered again.
- When the server performs a new start, the code word must be entered again.
- The Auto Logoff function de-registers the master automatically after 100 s (1.5 min.) if the connection is interrupted. After 1000 s (16.6 min.) when the user is logged onto the NetCon OPC and does nothing more than status or logger. If it does do something else, the 1000 s starts again. The value Auto Logoff Timeout can be changed in the dialog screen Server System Settings Auto Logoff can be disabled with 0.
- A master can give up its master role via the menu **Station** - **Station Logoff**.
- A slave can try to gain write access rights with the login even when no code word is set.

Operator Control and Configuration

The graphic interface of the NetCon OPC is described in detail in the section Operation and configuration.

Logger

The exteral Logger can be configured on the NetCon OPC. The contents of the logger are shown in the Logger dialog screen.

5.2 Selecting the Server Type

The term **INAT OPC-Server Ethernet** includes a total of seven types of servers which permit data exchange between different automation devices and OPC clients via Ethernet. Depending on the type of server being used, certain transport and application protocols are available. The TCPIPH1 OPC server is the "all-in-one" OPC server with a maximum protocol scope. During installation, the user is asked to select a certain server type. This can still be changed later on. However, the desired server type should be selected before a license code is requested. With the OPC server with TCP/IP, the **OPCpipe protocol** is supported which allows OPC server/OPC client communication between 2 PCs without permitting the DCOM access.

The following types of OPCs are available.

OPC Server Type	Transport Protocols	Application Protocols
OPC server TCPIPH1 (Order no: 100-3100-01)	• TCP/IP • RFC1006 • SPS-Header • ISO (H1)	• S7 protocol • S5-AP • CLX • Modbus on TCP • AB PLC-5 • Melsec-Q • Send / Receive • NetLink • OPCpipe client • OPCpipe server
OPC server H1 (Order no: 100-3110-01)	• ISO (H1)	• S7 protocol • S5-AP • Send / Receive
OPC server TCPIP-S (Order no: 100-3120-01)	• TCP/IP • RFC1006 • PLC header	• S7 protocol • S5-AP • CLX • Modbus on TCP • AB PLC-5 • Melsec-Q • Send / Receive • NetLink • OPCpipe client • OPCpipe server
OPC server TCPIP-I (Order no: 100-3150-01)	• TCP/IP • PLC header	• S7 protocol • S5-AP • Modbus on TCP • Send / Receive • NetLink • OPCpipe client • OPCpipe server
OPC server TCPIP-M (Order no: 100-3160-01)	• TCP/IP • PLC header	• Melsec-Q • Send / Receive • OPCpipe client • OPCpipe server
OPC server Modbus (Order no: 100-3140-01)	• TCP/IP • PLC header	• Modbus on TCP • Send / Receive • NetLink • OPCpipe client • OPCpipe server
OPCpipe Client (Order no: 100-1700-01)	• TCP/IP	• OPCpipe client • OPCpipe server

5.3 Server Tray

The server tray is installed during the server installation. In the lower, right-hand corner in the “Info area” of the work interface a mini application is started which appears as an icon. The icon has 3 different colors and signals the operating status of the OPC server.

Symbol	Meaning
▶	The OPC server has been started, is functioning and has been licensed correctly. The server tray has a connection to the OPC server.
🟥	The OPC server has been stopped or the server tray does not have a connection to the OPC server.
🟡	The OPC server has been started and is functioning but there is no valid license. After expiration of the test period of 72 hours, it stops running. The server tray has a connection to the OPC server.

The "Server Tray" mini application has a menu via which the following functions can be executed.

Menu Item	Action
Start INAT OPC Server	Start INAT OPC server service
Stop INAT OPC Server	Stop INAT OPC server service. If the OPC server is registered as an application, it cannot be stopped via the server tray.
Server Status - NetCon OPC	Starts the user interface NetCon OPC for the configuration and diagnosis of the INAT OPC server and connects to the OPC server that is running locally on the PC.
Exit INAT OPC Server Tray	Conclude Server Tray. The operating status of the OPC server is not affected by this.

The Server Tray can be started via **Start** - **Programs** - **INAT** - **OPC Server** - **Server Tray** after it has been concluded.

The Server Tray uses a socket connection via the TCP port 982. If a firewall is being used on the PC, this should be allowed.

5.4 Logger

The logger enables the listing of communication of the OPC server via the configured access path. You can specify whether the loggings are only stored in working storage or also in a file. The selected data exchange during running communication via an access path is recorded. This recording can be particularly helpful during trouble-shooting. Events are supplied with a time stamp, the name of the application and the message.
The logger includes the system service **INAT WmkLogger Service** which is automatically started when the system starts. If the service is not started, the logger cannot be indicated and nothing is logged.

In NetCon OPC, the Logger dialog screen indicates the logged entries.
The dialog screen Logger Settings is used to set which parts of the communication are to be logged.
The dialog screen Logger Memory Settings is used to set whether working storage or a file is to be used for storage.

5.5 OPC Client

Im Lieferumfang des OPC-Servers ist ein OPC Test-Client enthalten. Dieser OPC-Client wird während der Installation des OPC-Servers mit installiert. Der Client ist sehr einfach zu bedienen und ermöglicht ein Testen des OPC-Servers.

Indicate available OPC servers

The OPC client determines the OPC servers which are installed on the PC and offers them in the menu via "**Server"** > "**Connect...**". The client searches through the registry of the local computer and generates a list of available OPC servers.

Connect server and client

If the Ethernet OPC server "**INAT TcpIpH1 OPC Server**" is selected and confirmed with OK, this will automatically be started. If an error message appears, this may be because of the following reasons.
The **OPC Core Components** are not installed correctly ==> please reinstall.
The **Start rights** are not assigned in the DCOM settings. Please permit the access. For further information, go to DCOM Configuration – General

Add groups

To set up a new OPC group, the selection is made in the menu **Group** > **Add Group**. Any group name can be selected. The update rate (msec) specifies at what maximum data rate the individual data points will receive new values.

Add Item

To add items to the group, the group is highlighted and then selected with the menu **Item** > **Add Item**. Each item must have an item ID and access path. In addition, the required data type and active status can be selected.
The access path must correspond exactly with the name which was used when the access path was set up in NetCon OPC. An own dialog screen for the access path is not available for some OPC clients. With these clients the access path for communication with the server can be specified as shown below.

[Access Path Name]Item-ID or **Access Path Name.Item-ID**
Example:
[PLC1]DB5.W0
PLC1.DB5.W0

In the dialog screen to the bottom left all icon names are indicated which can be used instead of the item IDs.
The OPC server provides these via the browsing interface.

See also:

Server Symbol Edit
S7 Symbol Import
Modbus Symbol Import
AB ControlLogiX Symbol Import

When the item is specified, Validate can be used to check whether the server has accepted the item.

The item is registered with the server with OK. The variable is indicated in the main dialog screen with the value Quality and Updates.

Chapter 6

Item Syntax

6 Item Syntax

This chapter describes the item syntax which can be used to set up items (tags) for the corresponding connection (access path).

- S7 Item Syntax
- S5 Item Syntax
- Modbus Item-Syntax
- PLC-5 / SLC Item Syntax
- MELSEC-Q Item Syntax
- Send/Receive Item Syntax
- OPCpipe Item Syntax

The following chapters apply in general.

- Arrays
- Suffixe
- Data Types
- Special Items

6.1 S7 Item Syntax

The S7 item syntax is set up as shown below:

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

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

[Group.]<Area><Data type><Start address><.Bit no.>[.Array size][Suffix]

Legend: <> Mandatory [ ] optional

<Area>

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

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

<Data type>

Type	Syntax	DB / DI	F (M)	I/O (E/A)	PE /PA	C/T	With Array	Sig. Suffixes	OPC Data type
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.	BA BCD RI RU	VT_BOOL | VT_ARRAY VT_I2 VT_R4 VT_R4
WORD VT_UI2 VT_I4	**W** **Word**	DB5.W3 DB5.Word3	MW4 MWord4 FW4	EW4 EWord4 IW4	PEW4 PEWord4	C5 Z5 T5	DB5.W3.2 DB5.	BA BCD D	VT_BOOL | VT_ARR

			FWord4	IWord4 AW5 AWord5 OW5 OWord5 QWord5	PIW4 PIWord4 PAWord5 PAWord5 POW5 POWord5		Word3.2 MW4.2 EWord4.2 PAWord5.3 C5.3 Z5.10 T5.2 etc.	DT DU KF KT RI RU S5T SWAB TR	AY VT_I4* / VT_UI2** VT_BSTR VT_BSTR VT_BSTR VT_I2 VT_BSTR VT_R4 VT_R4 VT_BSTR VT_I4* / VT_UI2** VT_R4
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.		VT_BOOL | VT_ARRAY VT_I4 VT_BSTR VT_BSTR VT_BSTR VT_R4 VT_R4 VT_BSTR VT_I4 VT_R4
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 VT_BSTR	**S** **** **String**	DB5.S1.2 DB5.String1.2	MS2.3 MString2.3	-------	-------	-------	-------	KA	VT_BSTR
S7-STRING VT_BSTR	**G** ****	DB5.G2.2	MG2.3	-------	-------	-------	-------	KA	VT_BSTR

DOUBLE WORD VT_R8 VT_UI4	**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.		VT_BOOL | VT_ARRAY VT_I4 VT_I4 VT_R4 VT_R4 VT_R4 VT_I4 VT_R4 VT_BSTR VT_BSTR
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.	BA BCD KG RI RU SWABT TOD	VT_BOOL | VT_ARRAY VT_I4 VT_R4 VT_R4 VT_R4 VT_I4 VT_BSTR VT_BSTR

* If only standard types are selected under data types (server settings).
** If also data types without signs (server settings) are selected.
*** Caution! Danger of mixup 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 = 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, go to here.

[Suffix]

A value can be represented in another format with the aid of a suffix. For more information on suffixes, go to here.

6.2 S5 Item Syntax

The S5 syntax has the following structure.

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

The bit number must be specified if the data type is BOOL.

[Group.]<Area><Data type><Start address><.Bit no.>[.Array size][Suffix]

Legend: <> mandatory [ ] optional

[Group.]

Operand areas can be combined into groups. If there are gaps in the I/O (e.g., O10.1 and O10.3), these can be skipped by creating different groups. The gaps are then not requested. The group name is always set in front of the item name and separated by a period (Groupname.Itemname). The group name consists of “G” and the number.
Example:
G1.O10.1
G2.O10.3
The gap is skipped. The data points are accessed separately and not combined.

<Area>

	Syntax	Orientation[1]	Access Rights	Notes
Data block	DBx.	WORD	Read / write	With blocks, a block number x is required (x = 1 to 65535).
Expanded block	DXx.	WORD	Read / write	
Flag	M or F	BYTE	Read / write	
Timer	T	WORD	Read	
Counter	Z or C	WORD	Read	
Input	E or I	BYTE	Read	
Output	A or O or Q	BYTE	Read / write	
I/O	P	BYTE	Read / write	
Expanded I/O	OB	BYTE	Read / write	
System area		WORD	Read	
Absolute memory cells	AS	WORD	Read	

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

<Data type> for data blocks and expanded data blocks

	Syntax	Example	with Array	Sig. Suffix	OPC Data Type
BIT **VT_BOOL**	D	DB5D4.12	--------	--------	--------
BYTE **VT_UI1**	DB	DB5DB3	DB5DB3.5	BA	VT_BOOL | VT_ARRAY
LEFT BYTE **VT_UI1**	DL	DB5DL4	DB5DL4.2	BA	VT_BOOL | VT_ARRAY
RIGHT BYTE **VT_UI1**	DR	DB5DR2	DB5DR2.5	BA	VT_BOOL | VT_ARRAY
WORD	DW	DB5DW4	DB5DW4.5	BA	VT_BOOL |

	Syntax	Example	with Array	Sig. Suffix	OPC Data Type
VT_UI2 **VT_I4(*)**				BCD D DT DU KF KT RI RU S5T SWAB TR	VT_ARRAY VT_I4* / VT_UI2** VT_BSTR VT_BSTR VT_BSTR VT_I2 VT_BSTR VT_R4 VT_R4 VT_BSTR VT_I4(*) / VT_UI2** VT_R4
DOUBLE WORD **VT_R4(*)** **VT_UI4****	DD	DB5DD3	DB5DD3.2	BA BCD IEEE KF KG RI RU SWAB T TOD	VT_BOOL | VT_ARRAY VT_I4 VT_R4 VT_I4 VT_R4 VT_R4 VT_R4 VT_I4 VT_BSTR VT_BSTR
S7-STRING **VT_BSTR**	S	DB5S2.3	-------	-------	-------

* If only standard types are selected under data types (server settings)

** If also data types w ithout signs are selected (server settings

<Data type> for all other areas

	Syntax	M (F)	I/O	P / OB	C / T / RS / AS	with Array	Sign. Suffix	OPC Data Type
BIT **VT_BOOL**		M4.3 F4.3	E4.3 I4.3 A4.3 Q4.3	-------	-------	-------	-------	-------
BYTE **VT_UI1**	B	MB4 FB4	EB4 IB4 AB5 QB5	PB4 OB4	-------	MB4.3 FB4.4 EB4.4 IB4.5 AB5.2 QB5.5 PB4.2 OB4.3	-------	-------
WORD **VT_UI2** **VT_I4**	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	BA BCD D DT DU KF KT RI RU	VT_BOOL | VT_ARRAY VT_I4* / VT_UI2** VT_BSTR VT_BSTR

	Syntax	M (F)	I/O	P / OB	C / T / RS / AS	with Array	Sign. Suffix	OPC Data Type
						T5.3 RS4.2 BS4.7 AS5.2	S5T SWAB TR	VT_BSTR VT_I2 VT_BSTR VT_R4 VT_R4 VT_BSTR VT_I4* / VT_UI2** VT_R4
DOUBLE WORD **VT_R4(*)** **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	BA BCD IEEE KF KG RI RU SWAB T TOD	VT_BOOL | VT_ARRAY VT_I4 VT_R4 VT_I4 VT_R4 VT_R4 VT_R4 VT_I4 VT_BSTR VT_BSTR
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 is directly connected to the “T” or “C/Z” area. Timers can only be read! Counters can be read and written. Timer values are indicated in seconds (e.g., T = T = 0.7 => T = 0.7 s = 700 ms).
Counters are represented with decimals (0 to 999).

<Start address>

The start address specifies the addresses starting at which can be read or written.
Example: DB5.DW6: Double word 6 of the data block 5 is the start address.
Example: MB17: Flag byte 17 is the start address.
If the start address is a certain bit, the bit number must also be specified.

<.Bit number>

The bit number must always be specified when the data type is BOOL.
Example: I4.3: bit 3 of input byte 4 – an input bit is addressed here.
Example: MX12.1: Bit 1 of flag byte 12 – a flag bit is addressed here.

[.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.DW3.4
For more information on arrays, go to here.

[Suffix]

A value can be represented in another format with the aid of a suffix. The S5 syntax without suffix a data type without a sign. A data type with a sign requires a suffix.
Example: DB10DW3.4BCD.
For more information on suffixes, go to here.

6.3 Modbus Item Syntax

The Modbus Item syntax is set up as shown below.

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

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

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

If the connection "UnitID individual with item syntax" is selected, the following applies:

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

Legend: <> Mandatory [ ] Optional

[Group.]

Operand areas can be combined into groups.
The area: **0 - 65535**. If "**UnitID individual with item syntax**" is selected, the area is reduced to **0 - 255**.
If there are gaps in the I/O, these can be skipped by creating different groups. The gap is then not requested.
The group name is placed in front of the item name and separated by a dot
Gruppenname.Itemname (Groupname.Itemname).
The group name is made up of G and a number.
Example:
G1.40001
G2.R2
G3.S20.30

[UnitID.]

The UnitId is specified for this item with the item syntax.
For this the Unitname is placed set in front of the item name and separated by a dot (Unitidname. Itemname).
The Unitidname is made up of the Id and the number.
The area: **0 - 255**
When a group is specified, the area of the group numbers is reduced to 0 - 255.
If the UnitID is missing in the Itemsyntax, the parameterized UnitID is transferred to the PLC.
Example:
Id1.40001
Id2.R2
Id3.S20.30
Id1.G2.R3 UnitId + Group

<Area>

	Syntax Abbrev.	Abbrev. Number	Orientation*	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 Octal **	**J**	-	BIT	Read
Discrete Outputs Octal **	**P**	-	BIT	Read / write

* BIT-oriented means that one bit is addressed for each physical address.
WORD-oriented means that one w ord (16 bits) is addressed for each physical address.
** Entry of the start address is octal and the numbers 8 and 9 are invalid characters. Internally, the address is handled decimally and must be considered for logger and status.

Discrete inputs and outputs occupy 1 bit on the PLC. With read and write accesses they are processed as 8-bit values. This means that before the write access, 8 bits are read. The affected bit is exchanged and then 8 bits are written.

NOTE:
If another bit of this byte has changed in the meantime, it is overwritten with the old value!

Areas can either be addressed via the above stated alphabetic sequence or via a number. This means that a discrete input can be addressed by the abbreviation "E" the same as with the number "1."

<Data type>

	Syntax	**Syntax R**	**Syntax E / A**	**Syntax ER**	**with Array**	**Sign. Suffix**	**OPC Data Type**
BIT **VT_BOOL**	**X**	RX5.2 HRX5.2 4X5.2	E255 I255 DE255 DI255 125543	----	----	----	----
WORD **VT_I2**	**None**	R50 HR50 400050	----	ER120 IR120 312034	R50.2 HR50.2 400050.2 ER120.2 IR120.2 312034.2	BCD D WDT* KT RI RU S5T SWAB BA	VT_I2 VT_BSTR VT_BSTR VT_BSTR VT_R4 VT_R4 VT_BSTR VT_I2 VT_BOOL| VT_ARRAY
DOUBLE WORD **VT_I4**	**D**	RD50 HRD50 4D50	----	ERD120	RD50.3 HRD50.3 4D50.3	BCD SWAB KG IEEE TR RI RU T WDT*** BA	VT_I4 VT_I4 VT_R4 VT_R4 VT_R4 VT_R4 VT_R4 VT_BSTR VT_BSTR VT_BOOL|

							VT_ARRAY
REAL **VT_R4**	**R**	RR5	----	ERR5	RR5.3 ERR5.2	----	----
STRING **VT_BSTR**	**S**	RS5.4	----	ERS5.4	----	KA SWAB	VT_BSTR VT_BSTR

* When only standard types are selected for Data types
**When "also non-signed" is selected for Data types
*** With the suffix WDT (Wago Date and Time) 4 registers (8 bytes) are read.

<Start address>

The start address specifies the address starting at which read or write accesses begin.
Example: ER120: input register 120
If the start address is a certain bit, the bit number must also be specified.

<.Bit number>

The bit number must always be specified when the data type is BOOL!
Example: HRX5.2: Bit 2 of holding register 5

[.Array size]

Arrays are created to combine several units of one data type together in one field.
Example: HRD50.3
More information on arrays

[Suffix]

A suffix is used to represent a value in another format.
Example: HRD50.3WDT (representation in Wago Date and Time format)
More information on suffixes

6.4 PLC-5 / SLC Item Syntax

The item syntax for PLC-5 and SLC is shown below.

<Area>[File number]<Start address>[.Array size][Suffix]

Legend: <> Mandatory [ ] Optional

<Area>

File Type	Syntax	Orientation*	Access Rights	Default File number	Address
Output	**O**	BIT	Read / Write	0	octal
Input	**I**	BIT	Lesen	1	octal
Integer	**N**	WORD	Read / Write	7	decimal
Binary	**B**	WORD	Read / Write	3	decimal
Float	**F**	DOUBLE WORD	Read / Write	8	decimal
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

* BIT-oriented means that one bit is addressed for each physical address.
WORD-oriented means that one w ord (16 bits) is addressed for each physical address.
DOUBLE WORD-oriented means that one double w ord (32 bits) is addressed for each physical address.

[File numer]

Specification of the file number is optional. If it is not specified, the default file number is used. See column marked **Default File Number** in the table.

<Start address>

The start address specifies the address starting at which read or write accesses begin. The start address can consist of 2 pieces of information. Word (floating) number and when a single bit is accessed, then the bit number. The word number can be omitted with a bit. The word number or the floating number is introduced by a colon (:). The bit number is then introduced with a slash (/) <:word> or <:float> or </bit>. A format can be entered behind the colon. If a format is not entered, the data format in the **Orientation** column is used.

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

The address is octal for some areas. It is decimal for others. See column labeled **Address**.

Either the word number, the float number or the bit number is specified as the start address.
<:word> or <:float> or </bit>
The following syntax is used to address a certain bit within a word.
<:word/bit>

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

Syntax	Explanation
O:0	Word 0 in output file 0
O:0/12	Bit 10 (12 octal = 10 decimal) in output file 0
O/12	Bit 10 (12 octal = 10 decimal) in output file 0
I:37	Word 31 (37 octal = 31 decimal) in input file 1
I4:37/2	Bit 2 in word 31 ((37 octal = 31 decimal) in input file 4
I:1/0	Bit 0 in word 1 in input file 1
B3/26	Bit 26 in binary file 3
B12:5.15	Array of 15 words in binary file 12, starting at word 5
B12:5/15	Bit 15 in word 5 of binary file 12
F8:0	Float 0
F9:10.16	Array of 16 double words of float file 9, starting at double word 10
N23:4	Word 4 of integer file 23
N23:4.10	Array of 10 words in integer file 23, starting at word 4
N23:4/2	Bit 2 of word 4 in integer file 23 = bit 66 in integer file 23
N23/66	Bit 66 in integer file 23

[.Array size]

Arrays are created to combine everal units of one data type in a field together. Arrays are only possible for word areas and float areas.
Examples: N23:4.10

[Suffix]

A suffix can be used to represent a value in another format. If no suffix is specified, the formats from the **Orientation** column are used.
for more information on suffixes

6.5 MELSEC-Q Item Syntax

Two syntax versions are available for setting up items.

1. Simple version:

[Group.]<Area><Start address>[.Array size][Suffix]

2. Expanded version:

[Group.]<Area><.Type><Start address>[.Array size][Suffix]

Legend: <> Mandatory [ ] Optional

REMEMBER :

- With the expanded version, a period or a comma is required between the <area> and the <type>. If the period or comma is omitted, the syntax uses the simple version. The <type> then corresponds to the default type (for bit area BIT, for word area WORD – see table <Bereich> (Area)).
- If the representation of the start address is HEX, all numbers for this area are also HEX.
- The HEX / decimal numbers can be changed with the following prefixes.
  Conversion of HEX -> DEC: Input of 0d (number zero + the letter d) before the decimal address
  Conversion of DEC -> HEX Input of 0x (number 0 + the letter x) before the hexadecimal address
- When words, double words or strings are registered in a bit area, the start address is a bit address and only possible on word boundaries (dec. 0/16/32... or hex: 0/10/20...) (e.g., Y.D10.3).
- Bit masking is possible. The bit mask is specified before the suffix (see bit masking).

[Group.]

Areas can be separated into groups of related addresses.
If the I/O has gaps, these can be left out by creating groups for read and write accesses. The addresses before the gap are put in the one behind and the ones behind the gap in the other one. The gap is not requested.

The letter G and the group number are set in front of the item names and separated by a dot (GNr. Itemname).
Example:
G1.Y10
G2.Y30
Areas in different groups are queried individually and are not combined.

<Area>

	Syntax	Orientation*	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

* BIT-oriented means that one bit is addressed for each physical address. WORD-oriented means that one w ord (16 bits) is addressed for each physical address.

<.Type> <,Type>

	Syntax	**Simple Syntax**	**Ex.: Bit Area**	**Ex.: Word Area**	**with Array Bit Area**	**with Array Word Area**	**Sign. Suffixes**	**OPC Data Type**
BIT VT_BOOL	**X**	DY1	-----	D.X1.2	-----	-----	-----	-----
BIT** 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. I 2 R. INT 2	Y.I10.3 Y. INT10.3	R.I2.3 R.INT2.3	BCD D WDT* KT RI RU S5T SWAB BA	VT_I2 VT_BSTR VT_BSTR VT_BSTR VT_R4 VT_R4 VT_BSTR VT_I2 VT_BOOL|VT_ARRAY
WORD VT_UI2	**W WORD**	R20	Y.W10 Y. WORD10	R.W2 R. WORD2	Y.W10.3 Y. WORD10.3	R.W2.3 R. WORD2.3	BCD D WDT* KT	VT_I2 VT_BSTR VT_BSTR VT_BSTR

							RI RU S5T SWAB BA	VT_R4 VT_R4 VT_BSTR VT_I2 VT_BOOL|VT_ARRAY
DOUBLE WORD **VT_UI4**	**D** **DW** **DWORD**	-----	Y.D10 Y.DWORD10	R.D2 R.DWORD2	Y.D10.3 Y.DWORD10.3	R.D2.3 R.DWORD2.3	BCD SWAB KG IEEE TR RI RU T WDT*** BA	VT_I4 VT_I4 VT_R4 VT_R4 VT_R4 VT_R4 VT_R4 VT_BSTR VT_BSTR VT_BOOL|VT_ARRAY
DOUBLE INT **VT_I4**	**DI** **DINT**	-----	Y.DI10 Y.DINT10	R.DI2 R.DINT2	Y.DI10.3 Y.DINT10.3	R.DI2.3 R.DINT2.3	BCD SWAB KG IEEE TR RI RU T WDT*** BA	VT_I4 VT_I4 VT_R4 VT_R4 VT_R4 VT_R4 VT_R4 VT_BSTR VT_BSTR VT_BOOL|VT_ARRAY
REAL **VT_R4**	**R** **REAL**	-----	Y.R10 Y.REAL10	R.R2 R.REAL2	Y.R10.3 Y.REAL10.3	R.R2.3 R.REAL2.3	-----	-----
STRING **VT_BSTR**	**S** **STRING**	-----	Y.S10.20 Y.STRING10.20	R.S2 R.STRING2.20	-----	-----	KA SWAB	VT_BSTR VT_BSTR

* When only standard types are selected
** When "also unsigned" types are selected
*** For suffix WDT (Wago Date and Time) four registers (8 bytes) are read.
**** 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:
X10.5
D20.300
More information on arrays

[Suffix]

Suffixes can be used to represent a value in another format.
Example: D20.300KF
More information on suffixes

6.6 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>

	Syntax	Syntax Send	Syntax Receive	with Array	Sign. Suffixes	OPC Data Type
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	BA BCD RI RU	VT_BOOL| VT_ARRAY VT_I2 VT_R4 VT_R4
CHAR VT_BSTR	**C CHAR**	SC4 SCHAR4	RC5 RCHAR5	SC4.2 SCHAR4.2 RC5.4 RCHAR5.4	DT BA	VT_BSTR VT_BOOL| VT_ARRAY
WORD VT_I4 (VT_UI2*)	**W WORD**	SW10 SWORD10	RW10 RWORD10	SW10.2 SWORD10.2 RW10.4 RWORD10.4	BA BCD D KF KT RI RU S5T SWAB TR	VT_BOOL| VT_ARRAY VT_I2 VT_BSTR VT_I2 VT_BSTR VT_R4 VT_R4 VT_BSTR VT_I4 (VT_UI2*) VT_R4
INT VT_I2	**I INT**	SI4 SINT4	RI6 RINT6	SI4.2 SINT4.2 RI6.10 RINT6.10	BA BCD KT RI RU S5T SWAB TR	VT_BOOL| VT_ARRAY VT_I4 VT_BSTR VT_R4 VT_R4 VT_BSTR VT_I4 VT_R4
DOUBLE	**D**	SD6	RD6	SD6.2	BA	VT_BOOL|

WORD **VT_R8** **(VT_UI4*)**	**DW** **DWORD**	SDW6 SDWORD6	RDW6 RDWORD6	SDW6.2 SDWORD6.2 RD6.5 RDW6.5 RDWORD6.5	BCD KF KG RI RU SWAB T	VT_ARRAY VT_I4 VT_I4 VT_R4 VT_R4 VT_R4 VT_R8 (VT_UI4*) VT_BSTR
DOUBLE INT **VT_I4**	**DI** **DINT**	SDI6 SDINT6	RDI2 RDINT2	SDI6.2 SDINT6.2 RDI2.5 RDINT2.5	BA BCD KG RI RU SWAB T	VT_BOOL| VT_ARRAY VT_I4 VT_R4 VT_R4 VT_R4 VT_I4 VT_BSTR
REAL **VT_R4**	**R** **REAL**	SR2 SREAL2	RR2 RREAL2	SR2.2 SREAL2.2 RR2.5 RREAL2.5	-----	------
STRING **VT_BSTR**	**S** **STRING**	SS5.2 SSTRING5.2	RS5.2 RSTRING5.2	-----	KA DT	VT_BSTR VT_BSTR
S7-STRING **VT_BSTR**	**G**	SG5.2	RG5.2	-----	KA DT	VT_BSTR VT_BSTR

* If "also no sign” is selected under data types Server System Settings

<Start address>

The start address specifies the 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
More information on arrays

[Suffix]

A suffix can be used to represent a value in another format.
More information on suffixes

6.7 OPCpipe Item Syntax

To address an item via OPCpipe, the normal addressing of the item is expanded at the beginning with the OPCpipe client access path, followed by a question mark.
The OPCpipe item syntax is set up as shown below.

```
<OPCpipe client access path>?<target access path>.<Item>
```

Legend: <> Mandatory [ ] Optional

<OPCpipe client access path>

Specifies the access path on the OPCpipe client via which the corresponding OPCpipe server is addressed.

<Target access path>

Specifies the access path that is configured on the OPCpipe server as the connection to the PLC. When this access path is used, the option **Permit access via OPCpipe** must be enabled with the parameters OPC-Server Connection.

<Item>

Specifies the actual item name or the appropriate icon name.

The following should be adhered to when items are registered via OPCpipe.

- When an item is registered via OPCpipe, the desired data type must be specified. This can be omitted when one of the following is true:
    - The item in browsing is known by the OPCpipe client.
    - The access path of the OPCpipe client starts with certain letters which implicitly address the type of access path and by which the data type can be determined via the item syntax.

Abbrev.	Item Syntax
S7	S7 Item Syntax
S5	S5 Item Syntax
MOD	Modbus Item-Syntax
SLC PLC5	PLC-5 / SLC Item Syntax
MELS	MELSEC-Q Item Syntax
SR	Send/Receive Item Syntax

- The following data types are permitted for registering items via OPCpipe.

Simple Data Types	Array Data Types
VT_BOOL	VT_ARRAY | VT_BOOL
VT_I1	VT_ARRAY | VT_I1
VT_I2	VT_ARRAY | VT_I2
VT_I4	VT_ARRAY | VT_I4
VT_UI1	VT_ARRAY | VT_UI1
VT_UI2	VT_ARRAY | VT_UI2
VT_UI4	VT_ARRAY | VT_UI4
VT_R4	VT_ARRAY | VT_R4
VT_BSTR	

6.8 Arrays

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

Arrays are NOT possible for:
BOOL
DATE AND TIME
DATE
TIME
TIME OF DAY
STRINGS (is already an array of characters)
as well as for all suffixes which are string or bool to the client:
BA (already an array of bits)
KA
S5T
KT
D
T
DT
DU
TOD

6.9 Suffixe

Suffixes can be used to represent a value in another format.

Suffixes	**Syntax**	**Can Be Used for**	**Area**	**Data type**	**Variant Data Type**	**Comments**
BitArray	**BA**	Byte, Word, Int, DWord, DInt	No. of bits: Size in bytes times 8 Size in words times16 Size in Int times 16 Size in DWord times 32 Size in DInt times 32	BOOLEAN BOOLEAN BOOLEAN BOOLEAN BOOLEAN	VT_BOOL | ARRAY VT_BOOL | ARRAY VT_BOOL | ARRAY VT_BOOL | ARRAY VT_BOOL | ARRAY	With the BA suffix, the data sotrage saved on the PLC are shown as an array of bits.
BCD	**BCD**	Byte, Word, Int, DWord, DInt	Byte: 0 to 99 Word: 0 to 99 Int: 0 to 999 DWord: 0 to 9999999 DInt: 0 to 9999999	SHORT SHORT	VT_I2 VT_I2 VT_I4 VT_I4 VT_I4	With the BCD suffix, the data saved on the PLC are shown as non-signed, binary-coded values. For example, the decimal value "65535" is shown as "9999".
Date	**D**	Word	1990-01-01 to 2168-12-31	STRING	VT_BSTR	The suffix D is used to show the data saved on the PLC as data type DATE. The DATE data type occupies one word. The content corresponds to the number of days since 01.01.1990. The representation contains the year, the day and the month, separated by a hyphen.September 1, 2006 is shown as 2006-01-09. The value range is from 0 (0 days since 01.01.1990: 1990-01-01) to 65378 (65378 days since 01.01.1990: 2168-12-31). Rules for use of suffix D: • The years 1990 up to and including 2089 can be specified with 2 or 4 positions.The years starting with 2090 must be entered with 4 positions. 90 to 99 ==> 1990 to 1999 00 to 89 ==> 2000 to 2089 • The months and days can be specified with either 1 or 2 positions. • Anything but numbers can be used as a separator (e.g., 89/09/17). Any number of separators can be used.

Date and Time	**DT**	Word, Int	1990-1-1-00:00:00.000 to 2098-12-31-24:59:59.999***	STRING	VT_BSTR	The DT suffix is used to show the data saved on the PLC as combined data type DATE_AND_TIME. The data type DATE_AND_TIME has 8 bytes (64 bits). The year, the month, the day, the hour, the minutes, the seconds and the milliseconds are included. Remember the separators (hyphen, colon and period)!
Date and Time (SQL)	**DU**	Word, Int	1990-01-01 00:00:00.000 to 2098-12-31 24:59:59.999***	STRING	VT_BSTR	The DU suffix is used to show the data saved on the PLC as combined data type DATE_AND_TIME in SQL-compatible format. There is no hyphen in front of the time.
ASCII to Hex	**KA**	String, S7-String	HEX: 0 to 9, A to F	STRING	VT_BSTR	The KA suffix is used to show the data saved on the PLC in hex characters.
Signed	**KF**	Byte (LByte, RByte, S5), Word (S5), DWord (S5)	Byte: -128 to 127 Word: -32768 to 32767 DWord: -2147483648 bis 2147483647	CHAR SHORT LONG	VT_I2 VT_I2 VT_I4	The KF suffix is used to show the data saved on the PLC as a signed, fixed-point number.
S5-KG	**KG**	DWord, DInt	0,1469368E-38 to 0,1701412E39 Caution: The KG area in the PLC is larger than on the PC!	REAL	VT_R4	The KG suffix is used to handle the data saved on the PLC as a 4-byte, floating point number.
S5-KT-Format	**KT**	Word, Int	000.0 to 999.3	STRING	VT_BSTR	The KT suffix is used to show the data saved on the PLC as a 2-byte time constant. The time basis is supplied in addition to the time value. The value range is from 000.0 to 999.3.
Signed to Real	**RI**	Byte, Word, Int, DWord, DInt	Converts item into a real value and rounds off the value during write-access.	REAL	VT_R4	The RI suffix is used to show the data saved on the PLC as a signed, REAL data type. When writing to the PLC, the value is rounded off. Example: db9dw0RI – the value 10.7 is rounded off to 11.
Unsigned to Real	**RU**	Byte, Word, Int, DWord, DInt	Converts an item into a real value and rounds off the value during write-access	REAL	VT_R4	The RU suffix is used to show the data saved on the PLC as a non-signed, REAL data type. The value is rounded when written to the PLC.
S5 Time	**S5T**	Word, Int	0ms to 2h46m30	STRING	VT_BSTR	The S5T suffix is used to show the data saved on the PLC as the Simatic time S5TIME. The data type S5TIME occupies a

						16-bit word and is the product of time value and time frame. The time duration is specified in hours, minutes, seconds and milliseconds. The internal representation is a BCD number. The value range is from 0 ms to 2h46m30s. The smallest value is 10ms.
Swab Bytes	**SWAB**	Word, Int, DWord, DInt	Switches high byte and low byte		VT_I2, VT_I4*/UI2**, VT_I4, VT_I4, VT_I4´	The SWAB suffix is used to switch high byte and low byte of the date stored on the PLC.
Time	**T**	DWord	-24D_20H_31M_23S_648MS to 24D_20H_31M_23S_647MS**	STRING	VT_BSTR	The T suffix is used to show the data saved on the PLC as the data type TIME. The data type TIME occupies one double word (32 bits). The representation contains the information for days (d), hours (h), minutes (m), seconds (s) and milliseconds (ms). Milliseconds can be omitted. The value range goes from -2147483648 (-24d20h31m23s648ms) to 2147483647 (24d20h31m23s647ms).
Time of Day	**TOD**	DWord, DInt	0:0:0.0 to 23:59:59.999		VT_BSTR	The TOD suffix is used to show the data saved on the PLC as data type TIME_OF_DAY. The data type TIME_OF_DAY occupies one double word (32 bits). The representation contains the information for hours:minutes:seconds and . milliseconds. Milliseconds can be omitted. The value range goes from 0:0:0.0 to 23:59:59.999.
TimeReal	**TR**	Word, Int	0.01 to 9990.0	REAL	VT_R4	The TR suffix is used to show the data saved on the PLC as the data type TIME REAL. The value range goes from 0.01 to 9990.0 (s).

* If "only standard types" are selected under data types (server settings)
** If "also data types without signs" (server settings) are selected
*** ms can be omitted.

6.10 Data Types

The representation of both unsigned data types VT_UI2 and VT_UI4 on the OPC interface can be set.

While data types such as BYTE, WORD or S5TIME are used by the PLC user programs, OPC uses so-called variant types. The data types from the PLC user program are mapped accordingly on variance types. Each OPC item then has a canonical (i.e., its own) data type. The server provides the items with a default data format. For example, the OPC item DB5W3 (S7 syntax) is mapped to the variant type VT_I4. See the respective table for item syntax for which variant type the PLC data types are mapped.
When OPC items are set up, the OPC client can assign each item with a desired data type. When the data type EMPTY is specified, the default data format is used. When another format is selected, the server tries to convert the default format into the format desired by the client.

Only standard types

This mode converts unsigned data types into the next greater signed data types. It should only be used when the OPC client does not support the unsigned data types VT_UI2 and VT_UI4.
This mode is the default setting and can be used for most of the OPC clients. It is used when the client does not support the unsigned data types VT_UI2 and VT_I4.

Also unsigned types

This mode is the default setting. It is used when the OPC client supports the unsigned data types VT_UI2 and VT_UI4. For which data types are affected, see the respective table for the item syntax.

6.11 Special Items

The server offers special items which can be used as normal items and can provide information on connection parameters and communications states. The following special items are available.

Data Type / Item	OPC Typ
Status	VT_BOOL
WriteItemCount	VT_I4
WriteStatusMsg	VT_BOOL
ServerCycle	VT_I4
PollInterval	VT_I4
MaxInterval	VT_I4
StoreSettings	VT_BOOL
WriteComplete	VT_I4
ItemCount	VT_I4
ErrorCount	VT_I4
IpAddress	VT_BSTR
EthernetAddress	VT_BSTR
OwnTsap	VT_BSTR
TsapForRead	VT_BSTR
TsapForWrite	VT_BSTR
PortForRead	VT_I4
PortForWrite	VT_I4
EventCount	VT_I4
DbOffset	VT_I4
DwOffset	VT_I4
PlcStatus	VT_I4

Status

The Status special Item is generated from the connection status of the fetch and write connection. If one of the connections is not functioning, the status goes to “malfunction". If no write connection is parameterized, only the status of the read connection is indicated.

WriteItemCount

The WriteItemCount special item shows the number of written items.
The number is incremented when the secure data transmission to the PLC has been successfully concluded.
If an error occurs, the WriteStatusMsg is given an error code. When one item is written several times in succession before the server can “write down” the value, then the last value is written and not all intermediate values are buffered. The number in this case would be 1. The item can be set to 0 before the write access and then be queried after the write access.

WriteStatusMsg

The item shows the first error while writing items to the PLC.
If this item does not have a string or the first character is a blank, an error code is written. The string is then deleted by the operator (or visualization). The error code consists of 6 numbers, separated by a semicolon (“;”) without blanks:

Code word; DB; DW; Len; S5error; Connection error

Code	

1	BLOCK
2	FLAG
3	INPUT
4	OUTPUT
5	I/O
6	COUNTER
7	TIMER
8	SYSTEM DATA
9	ABSOLUTE
10	EXPANDED BLOCK
16	EXTMEM
17	EXT PERIPHERY
DB	
Gives the block number for BAUSTEIN and ERW_BAUSTEIN. Otherwise 0 without meaning.	
DW	
Gives the start value at which the interval begins.	
Len	
Gives the number of elements. To calculate the actual length in bytes, the code is also considered and calculated in acc. with the following rules. 1 Byte per unit for FLAG, INPUT, OUTPUT, I/O FLAG and EXT_IO 2 Byte per unit for BLOCK, COUNTER, TIMER, SYSTEM DATA, ABSOLUTE, EXT_BLOCK and EXT. MEMORY	
PLC error	
Shows the error that the handling block supplies from the PLC. Is only valid when the connection error is 0. For the description of the error codes, see the programming manual of the controller.	
0	No error
1	Wrong Q/ZTYP on handling block
2	Area doesn't exist on the automation device (AG)
3	Area on the AG (automation device) too small
4	QVZ error occurred on the AG (automation device)
5	Error in the indication word (ANZW)
6	No valid ORG format
7	No free data buffer
8	No free transport connections
9	Error of the communications partner
10	Connection error (connection broke down or setup was not possible)
11	Message error (error in the firmware)
12	Triggering error (e.g., RECEIVE to SEND)
13	Termination after RESET

14	Job with READ/WRITE (no triggering of the AG possible)
15	Job not present
16	System error
Connection error	
1	BAD_CR_PARAMS
2	NO_SLOT
3	WAIT_CONNECT
4	NOT_IMPLEMENTED
5	BAD_LINE
6	WAIT_DATA
7	WAIT_SEND
8	INTERNAL_ERROR
9	NO_REQUEST
10	NO_DRIVER
11	OVERLOAD
12	BLOCKED_DATA
13	NO_ADAPTER
14	ALREADY_RUNNING
15	NOT_SUPPORTED
16	TRY_AGAIN
17	NO_MEMORY
18	BAD_SIGNATURE
19	DATA_ERR (data error for FETCH / WRITE)

ServerCycle

The special item ServerCycle shows the number of complete passes of the server for a connection.

Tag Type:	**INTEGER / VT_I4**
ACCESS:	READ, WRITE
RANGE:	-2147483648 to 2147483647

Pole interval

The special item PollInterval is used to access the current poll interval. You will also find this value in the Access Path Definition dialog screen of the OPC server. The current poll interval is indicated in milliseconds. A client can overwrite this item with new values. The entered values have the following meaning. The value 0 or greater indicates a new poll interval. A negative value indicates that polling was disabled. In this case, the last entered positive value remains on the display. If the client write accesses this item, all items of this connection immediately receive an update.

<table>
<tr><th>Tag Type:</th><th>INTEGER (signed long) / VT_I4</th></tr>
<tr><td>ACCESS:</td><td>READ, WRITE</td></tr>
<tr><td>RANGE:</td><td><table><tr><td>0 to 2147483647</td><td>Valid values</td></tr><tr><td>-1 to -2147483648</td><td>Access path inactive,
No update of the items</td></tr></table></td></tr>
</table>

NOTE:
If you write a negative value to the POLLINTERVAL item, the client can stop all update procedures of the corresponding connection (access path) without deactivating the items.

NOTE:
Use the special item Pollinterval to improve the performance of your communication.

MaxInterval

The special item MaxInterval is used to access the measured Maximum Update Interval (in milliseconds) of all items of a corresponding connection. This value refers to the last complete polling cycle.

Tag Type:	**INTEGER / VT_I4**
ACCESS:	READ
RANGE:	0 to 2147483647

StoreSettings

The special item StoreSettings is used to save temporary changes of the PollInterval items via OPC to the hard drive. If the client writes a value of 1 to the StoreSettings item, the current PollInterval is written to the configuration file of the server. If the Update Interval is changed by the OPC and StoreSettings was not written with 1, the server uses the original Update Interval for the next start of the connection. A 0 is always returned when the Item is read although it makes no difference whether the client writes 0 or 1.

Tag Type:	**DISCRETE / VT_BOOL**
ACCESS:	READ, WRITE
RANGE:	0 or 1

WriteComplete

The special WriteComplete item is used to access the state of active write procedures of the current connection. If the Access Path is opened, the value of WriteComplete items is 1 (i.e., all write accesses have been concluded and no write procedures are still pending). If values of the applicable connection are overwritten, the value of WriteComplete items changes to 0 (i.e., no write access is in progress at the moment). When the server has concluded all the write accesses, the value of the WriteComplete items changes to 1 if all write accesses were successful or to -1 if at least one write access was not successful. If the WriteComplete item is not 0, the client can write it with 1 or -1 (1 to delete all errors or -1 to test the client reaction to write errors).

Tag Type:	**INTEGER / VT_I4**
ACCESS:	READ, WRITE
RANGE:	-1 or 0 or 1

ItemCount

The special ItemCount item is used to access the number of active items of the corresponding connection. This value is also indicated in the status window under the heading “Items.”

Tag Type:	**INTEGER / VT_I4**
ACCESS:	READ
RANGE:	0 to 2147483647

ErrorCount

The special ErrorCount item is used to access the number of all active, erroneous items of the applicable Access Path. If the connection is in a “bad” state, all items have errors (i.e., the ItemCount item corresponds to the ErrorCount item).

Tag Type:	**INTEGER / VT_I4**
ACCESS:	READ

RANGE:	0 to 2147483647

NOTE:
The special ErrorCount item is used to determine incorrectly configured or invalid items. Such an error has occurred when the Access Path status is 1 or jumps continuously from 1 to 0 and ErrorCount does not equal 0.

IpAddress

The special item IpAddress is used to access the target IP address of the applicable Access Path.

Tag Type:	**VT_BSTR**
ACCESS:	READ / WRITE
RANGE:	

EthernetAddress

The special EthernetAddress item is used to access the target Ethernet address (only for H1 connections) of the applicable Access Path.

Tag Type:	**VT_BSTR**
ACCESS:	READ / WRITE
RANGE:	6 bytes in Hex, (e.g., 0021A0060012 or 00 1 A0 06 00 12)

OwnTsap

The special OwnTsap item is used to access the local TSAP (only for H1 and RFC1006 connections) for the applicable Access Path.

Tag Type:	**VT_BSTR**
ACCESS:	READ / WRITE
RANGE:	Max. of 8 of bytes in hex (e.g., 01 01 or 01 02 03 04 05 06 07 08)

TsapForRead

The special TsapForRead item is used to access the remote TSAP for read accesses (only for H1 or RFC1006 connections) of the applicable Access Path.

Tag Type:	**VT_BSTR**
ACCESS:	READ / WRITE
RANGE:	Max. of 8 bytes in hex, (e.g., 03 02 or 01 02 03 04 05 06 07 08)

TsapForWrite

The special TsapForWrite item is used to access the remote TSAP for write accessing (only for H1 and RFC1006 connections) of the applicable Access Path.

Tag Type:	**VT_BSTR**
ACCESS:	READ / WRITE
RANGE:	Max. of 8 bytes in hex, (e.g., 01 01 or 01 02 03 04 05 06 07 08

PortForRead

The special PortForRead item is used to access the port for read accessing the applicable Access Path.

Tag Type:	**VT_I4**
ACCESS:	READ / WRITE
RANGE:	1 to 65535

PortForWrite

The special PortForWrite item is used to access the port for write accessing the applicable Access Path.

Tag Type:	**VT_I4**
ACCESS:	READ / WRITE
RANGE:	1 to 65535

EventCount

Tag Type:	**VT_I4**
ACCESS:	READ / WRITE
RANGE:	-2147483648 to 2147483647

DbOffset

The value of DbOffset is added to all DB numbers of this Access Path. This is used to be able to address several machines via one connection.

Tag Type:	**VT_I4**
ACCESS:	READ / WRITE
RANGE:	-65535 to 65535

DwOffset

The value of DbOffset is added to all start addresses of this Access Path. This is used to be able to switch several devices in one DB.

Tag Type:	**VT_I4**
ACCESS:	READ / WRITE
RANGE:	-65535 to 65535

PlcStatus

The special Item PicStatus is used to indicate the status of an S7 CPU. With S7 connections and HPP switched on, it can be registered.

The item is read every 5 seconds from the PLC. If something is written to this item, it is read immediately. If another time base is to be selected, the value "PlcStatusInterval=10000" can be changed in OpcServer.ini. Here the value 10000 means every 10 seconds.

Tag Type:	**VT_I4**
ACCESS:	READ / WRITE
Meaning:	0: Doesn’t exist or cannot be read 8: RUN All other values: STOP

Index