Software License Agreement

Before installing this software product, please read the following conditions carefully. Purchase or (at the latest) installation of the software product constitutes agreement with the conditions stated below. Should you not be in agreement with these conditions, return the software and the manual within 14 days from the date shown on the invoice. The purchase price can only be reimbursed within this time period.

I. Preamble

The subject of the agreement is the computer program contained on the data medium (subsequently called the "SOFTWARE PRODUCT"). The LICENSER points out that today's technology does not permit creation of computer software which will function correctly with all applications and in all combinations. For this reason, the subject of the agreement shall only be a software program which functions correctly as stated in the program description and user's manual.

The SOFTWARE PRODUCT and all its contents such as screens, graphics, text and sample applications is and shall remain the property of the LICENSER and is protected by copyright law.

The LICENSER is willing to give the LICENSEE the rights of use defined in this agreement in return for an appropriate license fee.

This license agreement between LICENSEE and LICENSER shall go into effect when the SOFTWARE PRODUCT is received by the LICENSEE.

II. Terms

SOFTWARE PRODUCT	**OPC server** TCPIPHI	Order No: 100-3100-01
	OPC server H1 (ISO)	Order No: 100-3110-01
	OPC server TCPIP for INAT and Siemens CPs	Order No: 100-3120-01
	OPC server TCPIP for INAT CPs	Order No: 100-3150-01
	OPC server MPI (without MPI adapter)	Order No: 100-3300-01
	OPC server MPI (with MPI adapter)	Order No: 100-3310-01
	DDE server TCPIPHI	Order No: 100-3050-01
	DDE server H1 (ISO)	Order No: 100-3060-01
	DDE server TCPIP for INAT and Siemens CPs	Order No: 100-3070-01
	DDE server TCPIP for INAT CPs	Order No: 100-3080-01
	DDE server MPI (without MPI adapter)	Order No: 100-3350-01
	DDE server MPI (with MPI adapter)	Order No: 100-3360-01
	PC-H1	Order No: 100-4500-01
	NetSpector	Order No: 100-2260-01
	IP driver for MS DOS to S5	Order No: 100-5000-01
	IP driver for MS DOS to S7	Order No: 100-5100-01
	Net-Plugin for the Soft PLC under DOS	Order No: 100-5100-01

The modules can be purchased individually or in different combinations. The license agreement applies equally to each individual version. For this reason, the term "SOFTWARE PRODUCT" will subsequently be used to include all versions.

LICENCER	INAT GmbH Ostendstraße 50A D-90482 Nürnberg
LICENSEE	Natural or legal person who purchases the license for the use of the SOFTWARE PRODUCT
THIRD PARTIES	Other natural or legal persons

III. Scope

1. The SOFTWARE PRODUCT is licensed exclusively for use by the LICENSEE. The LICENSEE shall be permitted to resell the SOFTWARE PRODUCT to THIRD PARTIES provided the THIRD PARTY agrees to this software license agreement. In this case, the software license agreement shall no longer pertain to the LICENSEE. Rights of use of the SOFTWARE PRODUCT shall be canceled for the LICENSEE since the rights of use have been transferred to a THIRD PARTY who is now the LICENSEE himself/herself.

2. The SOFTWARE PRODUCT may only be used on one single computer. The LICENSEE shall be obligated to purchase a separate license for each computer on which the SOFTWARE PRODUCT is being used.
Purchase of multiple and network licenses shall be negotiated separately between LICENSEE and LICENSER.

3. Duplication of the SOFTWARE PRODUCT and all its documentation is prohibited except for the creation of an archive copy for exclusive use by the LICENSEE.
If the original data medium becomes defective while in possession of the LICENSEE, it can be exchanged for a replacement data medium from the LICENSER. The costs of the exchange shall be carried by the LICENSEE.

4. The SOFTWARE PRODUCT is delivered on a data medium as a demo version with limited function scope.
The demonstration version may not be used for commercial purposes. After remittance of the license fee for the particular version desired, the LICENSEE shall receive a code number from the LICENSER to release the applicable SOFTWARE PRODUCT.

5. When the LICENSEE is provided with a new program version because he/she purchased an update or for any other reason, the rights of use to earlier program versions shall become invalid after installation of the new program version (i.e., only the latest version may be used).

6. The LICENSEE and THIRD PARTIES may not modify or reverse-engineer the SOFTWARE PRODUCT (i.e., no decompiling or deassembly).

The LICENSEE shall be liable for all damages caused by violation of these conditions.

IV. Limited Warranty

The data mediums on which the SOFTWARE PRODUCT was delivered are free of material and manufacturing faults under normal use during the first 6 months after the delivery date. In the judgment of the LICENSER, the data mediums are free of viruses. However, if hidden viruses have crept in, the LICENSER shall assume no liability for any subsequent damages.

1.The above limited warranty shall not include data mediums which were damaged by chance or misuse or unauthorized manipulations (persons other than employees of the LICENSER).

2. During the stated warranty period, the LICENSER shall be obligated to replace a faulty data medium if this medium is sent to the LICENSER together with a copy of the invoice. Other claims (in particular, for damage compensation) shall be excluded subject to paragraph 4 and Roman numeral V.

3. Other warranty obligations shall be excluded. In particular, no warranty shall be granted for program content, its freedom from errors or its suitability for certain purposes. Responsibility for software and hardware selection, installation, use and expected results as well as data security and data backup with backup copies shall rest exclusively with the LICENSEE.

4. Promised features shall be excluded from the above warranty restrictions. To take effect, these features must be stated in writing and signed by an authorized representative of the LICENSER. The LICENSER shall only be liable for damage compensation stated under Roman numeral V.

V. LIMITED LIABILITY

1. The LICENSER shall assume no liability for damages unless the damage was caused by intent or gross negligence on the part of the LICENSER.

2. Liability due to any features promised by the LICENSER shall not be affected by this. Liability for subsequent damages (due to deficiencies) not related to promises by the LICENSER shall be excluded.

3. No liability shall be assumed for indirect damages (lost profit in particular).

4. Any damage claims shall be restricted to the amount of damage whose possible occurrence the LICENSER reasonably had to reckon with at the time the agreement was concluded, considering the circumstances known at that time. In all cases, liability shall be restricted to twice the sales price paid (license fee) regardless of whether claims of contractual rights, damage claims or other liability claims are involved.

VI. Validity and Termination

1. The license issued to the LICENSEE shall remain valid until terminated by the LICENSEE or the LICENSER.

2. The LICENSEE can terminate the license at any time by returning the SOFTWARE PRODUCT (including the archive copy and all related documentation) to the LICENSER. If the return is not a result of a warranty claim, no license fee shall be reimbursed.

3. The LICENSER shall be permitted to withdraw the license from the LICENSEE without prior notice when this license agreement is violated by the LICENSEE. In such cases, the LICENSER shall have the right to enforce all rights granted by law, with regard to criminal prosecution and damage compensation.

4. The LICENSEE terminates the license by resale to a THIRD PARTY in accordance with III, paragraph 1.

VII. Applicable Law

1. With regard to applicable law, this license shall be subject to the laws of the Federal Republic of Germany and exclusively subject to the court decisions of the German courts.

2. For full merchants, the place of jurisdiction for all legal disagreements resulting from the contractual relationship as well as its creation and effectiveness shall be Nuremberg. However, the LICENSER shall be authorized to bring charges against the LICENSEE at the place where the LICENSEE's business is located.

3. Place of fulfillment for all obligations resulting from the contractual agreement shall be Nuremberg.

License

- Compare the entries under **Product** and **Order No.** with your ordering data!

- Enter your company name and the user name in order to get a unique license.

- Use the Button **->Clipboard.** The date for the licensing request are now available in the clipboard for pasting it into your favorite Email program:

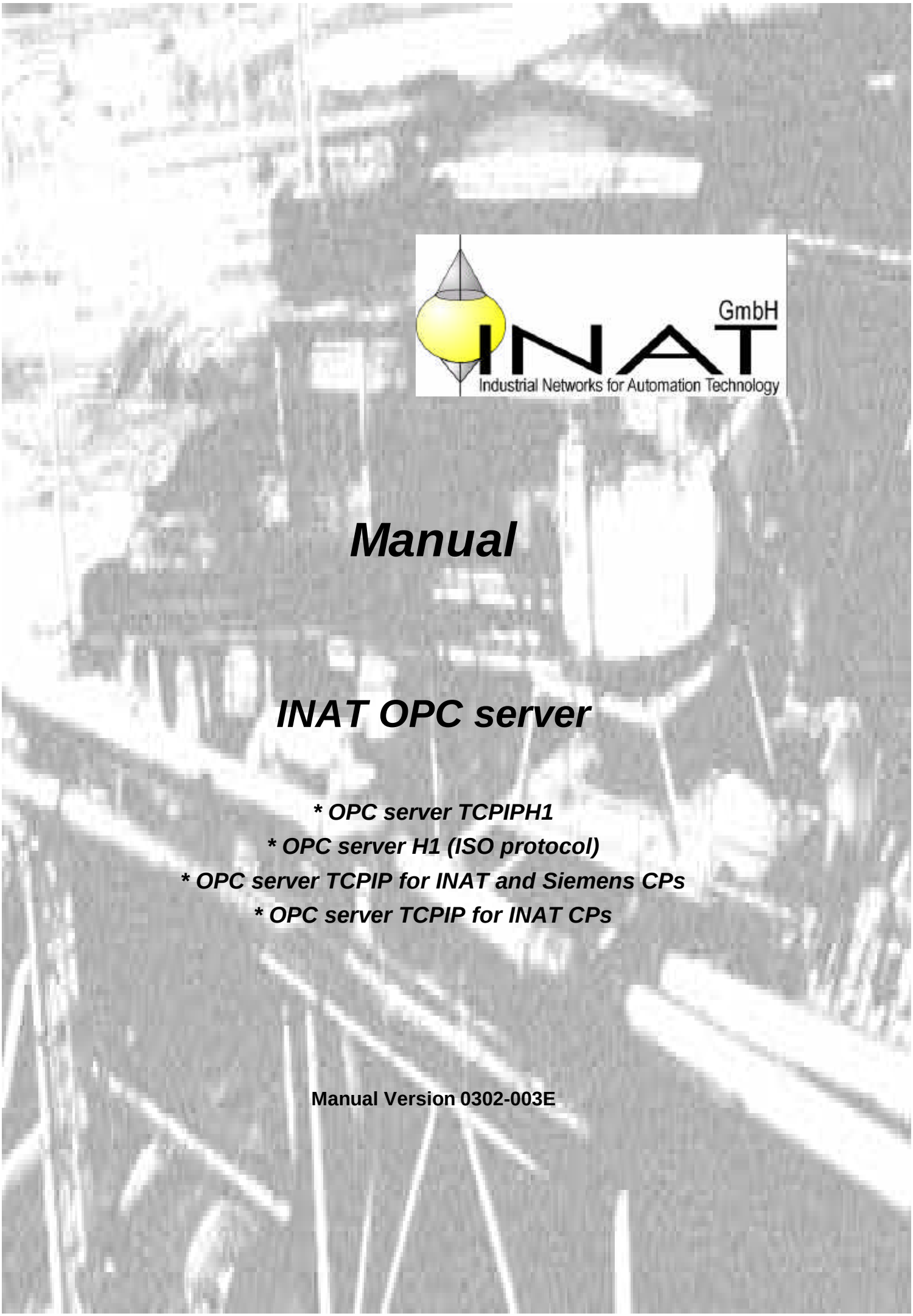
  OPC-Server TCPIPH1
  100-3100-01
  INAT
  Dipl. Ing. Thomas Muster
  5IMMQ0H0CH2FQQH
  NB The Request Code of your system is unique.

- Send the **License Request Code**
  - by Fax to +49 911 / 544 27-27 or
  - by Email to info@inat.de

- Due to the contract with INAT you will get an unlimited **License Confirm Code** or a limited License Confirm Code for test purposes.

- Under License the following messages are possible:

License status	ok
License status	The checksum is wrong: Check the entered codes
License status	no valid license:If the the Test runtime is over (by default 2h) the server ends with the message "The server ends, because there exists no valid license"



Manual

INAT OPC server

**** OPC server TCPIPH1***
**** OPC server H1 (ISO protocol)***
**** OPC server TCPIP for INAT and Siemens CPs***
**** OPC server TCPIP for INAT CPs***

Manual Version 0302-003E

The contents of this manual and the related software are the property of Werner Mehrbrodt, INAT GmbH. This information is subject to the conditions of a general or special licence agreement (one-time licence) and may only be used or copied in accordance with the specifications of this agreement.

The information in these documents is not binding. Changes to the contents can occur at any time without prior notification. Subject to change due to technological advances.

Industrial Networks for Automation Technology
Ostendstraße 50A
D-90482 Nürnberg
Phone: 49 911 / 5 44 27-0
Fax: +49 911 / 5 44 27-27
BBS: +49 911 / 5 44 27-28
Email: info@inat.de
Internet www.inat.de

This manual describes the installation and use of the following four INAT OPC servers:

- ***OPC server TCPIPH1***
- ***OPC server H1 (Iso protocol)***
- ***OPC server TCPIP for INAT and Siemens CPs***
- ***OPC server TCPIP for INAT CPs***

Numerous illustrations taken from the running programs are used to explain the individual steps involved in using the INAT OPC servers.

Recent issues version 2.03.01
Automatical copying of the H1 driver

Recent issues version 2.03.00
Fetch on Event
Handling S5 and S7 strings
Suffix RI
Suffix RU

Recent issues version 2.01.01
Language switch
Access Path Syntax
Suffix T
Suffix R

Recent issues version 1.23.19
(Alias Browsing)
Access Path Configuration
Date
Date_Time
Show Server Icon in the Taskbar
Licence
Indication mode

Recent issues version 1.23.17
New special items for monitoring and controlling communication with a PLC

Recent issues version 1.23.16
S7 syntax
Read optimization
Quality Bit

Contents

1 Communication Server for the S5/S7

1.1 INAT OPC server

When demands on communication between process and host levels are high, there is a growing need to use Process Visualization systems in both the SINEC® H1 and TCP/IP networks to communicate with S5 and S7 PLCs.

PV =
P*rocess* ***V****isualization*

The INAT OPC servers are Windows applications which function as a OLE server and permit other Windows programs to exchange data with the S5 and S7 family of Siemens PLCs. The several OPC servers support different protocols. The OPC server TCPIPH1 supports all four protocols standard TCP/IP, SINEC® H1 (ISO), PLC Header and RFC1006. This special TCP/IP protocol RFC1006 is supported by the OPC servers TCPIP for INAT (and Siemens) CPs, too. An overview about the supported protocols can be seen in passage „Protocols“ on page 20.

OLE =
O*bject* ***L****inking* ***E****mbedding*

Process visualization programs such as the familiar WinCC and FIX provide the OPC interfaces for data communication with the programmable controllers at the process level.

Using the INAT OPC server, you can link your process visualization to either a TCP/IP or SINEC® H1 network to communicate with S5 and S7 PLCs. Thanks to optimized data transmission between OPC server and PLC, you can take full advantage of the performance potential of your process visualization software.

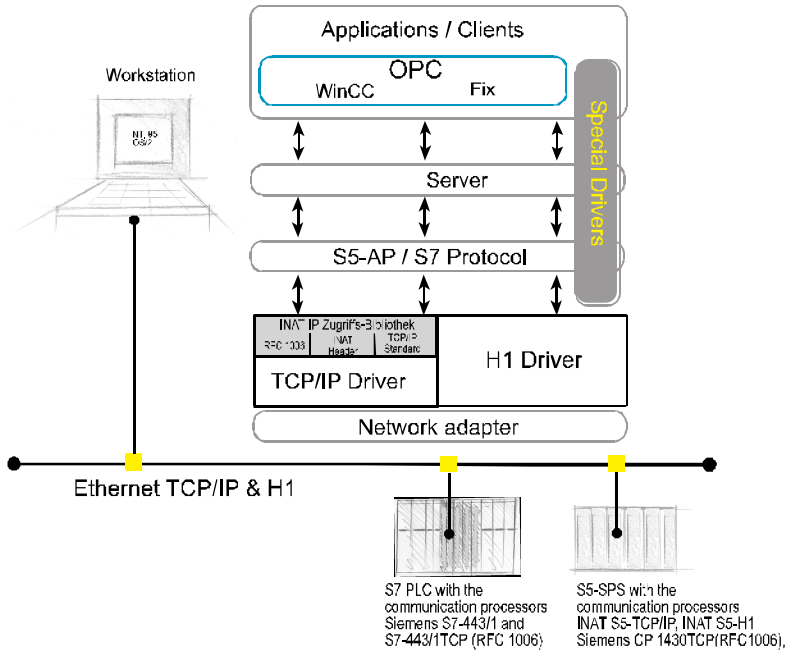


FIGURE 1.
Overview of the OPC servers

1.1.1 Supported Process Visualizations

The INAT OPC servers support the following process visualizations

DIADEM®	Procon WIN®
Factory Link™	RAMSES®
FIX™	RSView™
GENESIS32™	VISIWIN®
Graph PIC®	WinCC®
InTouch™	WIZCON™
InVISU®	ZENON®

1.2 Network Protocols

Since the INAT OPC servers work with either TCP/IP or H1, communication between different programmable controllers and computers is possible. The job of the transmission protocols is to provide uniform communication between different systems. Secure connections between different systems can be established with both LAN and the public networks.

A few examples of the tasks that can be handled with TCP/IP and H1 are listed below.

- ***Central control and monitoring of manufacturing plants***
- ***Transmission of machine states or management information***
- ***Creation of production statistics***
- ***Transmission of large amounts of data***
- ***Remote diagnosis, remote programming and remote parameterization of the programmable controllers and communication processors***

1.3 TCP/IP

These transmission protocols have become today's industrial standard for networking systems with LAN.

TCP /IP =
T*ransmission* ***C****ontrol* ***P****rotocol*
I*nternet* ***P****rotocol*

TCP/IP protocols first became familiar in the UNIX world and in Ethernet LANs. Today they are widely used in all other networks and are available for all modern operating systems.

1.3.1 IP

A TCP/IP network is dependent on clear addressing of individual devices, subnetworks and so on. Since the length of the addresses is always identical (i.e., 32 bits), there must be different classes of addresses for different distributions of the number of computers and networks.

Not only this address can be used for the destination computer but also a previously assigned, common everyday name. The Name Server Protocol handles the conversion of this name to the actual destination address. This requires that the address of a DNS is specified. The DNS converts all symbolic TCP/IP addresses into numbers.

DNS =
D*omain* ***N****ame* ***S****erver*

IP permits data packets (i.e., datagrams) to be transmitted from the sender via several networks to the receiver. Since there is no acknowledgment mechanism, IP guarantees neither datagram sequence nor secure delivery to the receiver. This is handled by the TCP layer. However, since IP has fragmentation capability, large datagrams whose maximum permissible length is shorter can also be transmitted over networks.

The Internet module packs TCP segments into Internet datagrams and sends them to the destination Internet modules or gateways. In a local network, the datagram is packed into the protocols of the local network. At the gateway, the packet is adapted to the new network structure and routed to its destination or to the next gateway. If necessary, the gateway can divide the datagrams into smaller fragments.

The destination Internet module removes the datagram envelope of the segment and transfers it to the destination TCP.

The following figure shows a diagram of this architecture.

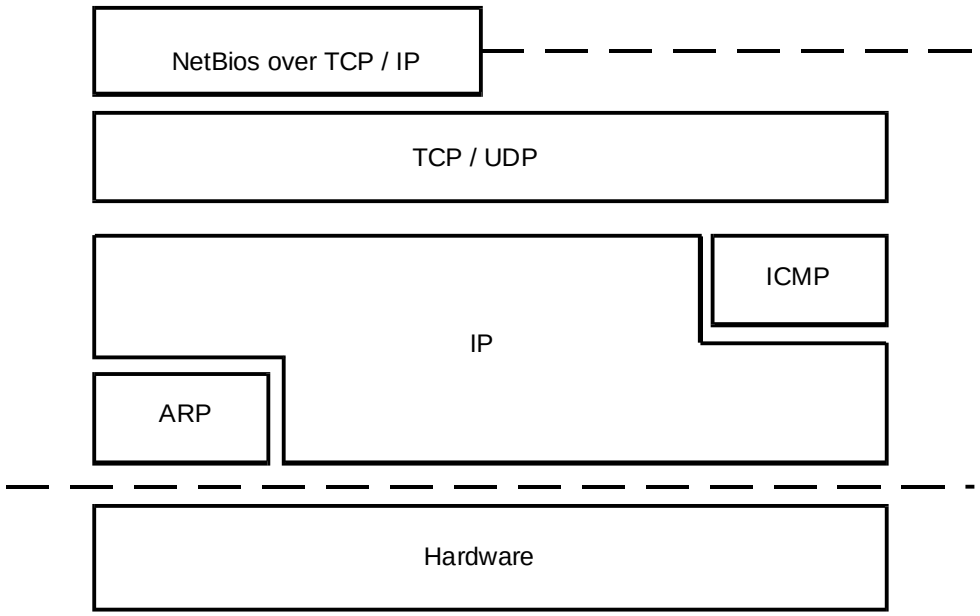


FIGURE 2.
TCP/IP

1.3.2 TCP

TCP is a connection-oriented, end-to-end protocol. It is approximately comparable to layer 4 of the ISO-OSI-7 layer model.

Layer 7	ALI/API,
Layer 6	Presentation
Layer 5	Session
Layer 4	Transport
Layer 3	Network
Layer 2	Data Link
Layer 1	Physical Link

TABLE 1.
Layer 4 of the ISO/ OSI-layer model

TCP is an interprocess communication protocol between computers which are connected to packet-transmitting networks. It can be used with both LAN and WAN.

Like the OSI protocol, TCP is not dependent on the network type. It requires a simple, independent datagram service (e.g., usually IP) but can also be run with an Internet protocol with similar performance features.

As with most other protocols, TCP is executed in three phases.

1. Connection establishment

2. Connection

3. End of connection

Transmission path

TCP divides the data into segments and calls IP to transfer the segments to the destination TCP. The destination TCP stores the data in the buffer of the destination process and informs the called process.

Transmission security

TCP has security mechanisms for data which arrive from the Internet communications system (e.g., fragmented, lost, duplicated and so on). Each segment receives a sequence number. This number must be confirmed by the partner TCP. The segment is transmitted again if confirmation does not arrive within the specified time. The receiver uses the sequence number to put the packets in the correct order and, if necessary, remove duplicates.

1.4 SINEC H1

Like TCP/IP, SINEC®H1 permits the establishment of a large decentralized automation system at the top level of communications technology. SINEC®H1 can be used to implement extensive networking of the individual areas within a company. The PG 685, PG 730, PG 750 and PG 770 programmers can be connected to SINEC®H1.

1.5 Setup of an Ethernet Network

An Ethernet network connects various network stations so that they can communicate with each other. Network stations can be personal computers, industrial computers, SIMATIC S5s, and so on. Separated by a minimum distance, the network stations are connected separately with the network cable. Together, the network stations and the network cable form a total segment.

Although the total segment is restricted in its length and the number of stations which it can handle, a network can be expanded by connecting two or more segments with a repeater (i.e., bus amplifier).

Repeaters are used to expand the physical topology of the bus system. A repeater is transparent over the entire network and combines two segments into a single logical channel.

The job of the repeater is to regenerate and amplify the signals in both directions. It must also be able to recognize, process and forward segment-overlapping collisions. The repeater cannot be addressed by its own network address since the connected stations are not aware of its existence.

The total network cable is the sum of all segment cables.

Ethernet consists of individual segments. One segment has up to 100 stations (e.g., programmable controllers and computers) and a maximum length of 500 m. Segments can be connected together with repeaters. Not more than 2 repeaters may be located between two stations. The repeater can also be used to implement so-called remote repeater configurations. This requires two repeaters and two additional Ethernet bus couplers. The bus line consists of a thick Ethernet cable (i.e., yellow cable) or thin Ethernet cable (i.e., cheapernet).

An interface multiplier can also be connected to a bus coupler of the triaxial cable. Up to 8 stations can be connected to this interface multiplier.

1.6 Method of Function

As prescribed by the standardized Ethernet system, Ethernet works according to the principle of random bus access (i.e., each station accesses the bus when necessary). Bus access is coordinated by the CSMA/CD procedure. Each station continuously monitors the bus line and receives the sending data addressed to it. A station does not start to send data unless the line is free. If two stations start to send data at the same time, they recognize this, stop the transmission and start again after a random amount of time. Other stations may access the bus during this time.

CSMA/CD=
C*arrier* ***S****ense* ***M****ultiple* ***A****ccess*
*/****C****ollision* ***D****etection*

Bus couplers can be used to connect the PG 685, PG 730, PG 750 and PG 770 programmers directly to Ethernet with SINEC®H1. This permits all programmable controllers connected to the bus to be programmed remotely. The 923C coordinator or the programmer multiplexer can be used to program several modules via one connection.

Services in the controller

Both the user services of level 4/4a (i.e., handling blocks for the S5 or libraries for PC modules) and of level 7 are available.

2 Features of the INAT OPC servers

2.1 General

You use the INAT OPC server to define communication channels between the process visualization (WinCC, FIX) and a S5 or S7 PLC. You then select the desired protocol TCP/IP or SINEC® H1 and enter your connection parameters. All settings can be modified e.g., the protocol of an already parameterized connection can be changed.

2.2 What is OLE?

OLE=
O*bject* ***L****inking* ***E****mbedding*

COM=
C*omponent* ***O****bject* ***M****odel*

DCOM=
D*istributed* ***COM***

OPC=
O*LE for* ***P****rocess* ***C****ontrol*

The OLE interface is based on the Client/Server technology, which provides data of an application for linking and presenting in another application. The server creates and formates the data, while the client links and presents the data.

The OLE requests are defined by the Component Object Model COM (in this case COM ist not the serial interface of an PC). If several stations are requested this is called Distributed COM model (DCOM).

Introduction of OPC as an all-purpose standard interface promises simple data communication at the actuator/sensor level up to the process-level and will put an end to today´s driver chaos. Since the arrival of Windows 95, PC users have enjoyed the convenience of OLE. Excel is started automatically when a linked object is double clicked (e.g., an Excel tabel in a Word document). In terms of automation technology, this means that the all-purpose OPC interface provides a uniform link to visualization. Actually the OLE standard interface provides more comfort and more versatile purposes than DDE. The use of OPC will drastically reduce the number of interfaces in a plant. This is important in view of the enormous costs for adapting and maintaining different interfaces.

There are many Windows applications that provide the OLE inteface.

- ***Process visualization***
- ***Data bases***
- ***Programs for measured value acquisition***
- ***Statistik programs***
- ***Development tools (e.g. Visual Basic)***
- ***Presentation software***

2.3 Features of the INAT OPC servers in Overview

- Universal for all OPC clients, especially visualization programs (WinCC, FIX and so on)
- Can be executed under Windows 95/98/2000/NT/ME
- Reduction of OPC communication via new-old comparison of the data
- Reduction of the PLC telegrams to be written to the PLC by a group function
- Fast and precise localization of errors with OPC server log and the status display in the main window of the OPC server
- Simple parameterization of the connections
- Clearly organized list of the parameterized connections
- Simultaneous support of several clients

At the moment the OPC servers support the following data types:

TABLE 2.
Supported data types

Tag Name	compatible up to version 1.22	also unsigned types	only standard-types (default)
DB5D1.0	VT_BOOL	VT_BOOL	VT_BOOL
DB5DB1	VT_I4	VT_UI1	VT_UI1
DB5DB1KF	VT_I4	VT_BSTR	VT_BSTR
DB5DL1	VT_I4	VT_UI1	VT_UI1
DB5DR1	VT_I4	VT_UI1	VT_UI1
DB5DW1	VT_I4	VT_UI2	VT_I4
DB5DW1KF	VT_I4	VT_I2	VT_I2
DB5DW1KT	VT_BSTR	VT_BSTR	VT_BSTR
DB5DW1KC	VT_I4	VT_UI2	VT_I4
DB5DW1BCD	VT_I4	VT_I2	VT_I2
DB5DD1	VT_I4	VT_UI4	VT_R8
DB5DD1KF	VT_I4	VT_I4	VT_I4
DB5DD1KG	VT_R4	VT_R4	VT_R4
DB5DD1IEEE	VT_R4	VT_R4	VT_R4
DB5S1.10	VT_BSTR	VT_BSTR	VT_BSTR
DB5DW1KF	VT_I4	VT_I2	VT_I2
DB5DW2KF	VT_I4	VT_I2	VT_I2
DB5DW0.10	----------	VT_ARRAY	VT_ARRAY

Compatible... up to version 1.22:

Select this mode if you want to use older applications with the recent version of the OPC server. This mode doesn´t support arrays.

Also unsigned types

Select this mode if you are working with the visualization WinCC.

Only standard types (default):

This mode has to be set for most of the available OPC clients and therefore is default setting. The server returns data types, which corresponds to the specified item.

As from version 1.22 all data types are displayed by the logger fully decoded.

Connection with

OPC server / CPs	TCPIPH1	H1	TCPIP for INAT and Siemens CPs	TCPIP for INAT CPs
INAT S5-TCP/IP	✓	✓	✓	✓
INAT S7-TCP/IP	✓	✓	✓	✓
INAT echolink	✓	✓	✓	✓
INAT S5-H1	✓	✓		
Siemens CP 1430	✓	✓		
Siemens CP 1430-TCP	✓		✓	
Siemens CP 143	✓	✓		
Siemens CP 343-1*	✓	✓	✓	
Siemens CP 443-1*	✓	✓	✓	

*depending on firmware release

Protocols

OPC server / CPs	TCPIPH1	H1	TCPIP for INAT and Siemens CPs	TCPIP for INAT CPs
Standard TCP/IP	✓		✓	✓
PLC Header	✓		✓	✓
RFC1006	✓		✓	
H1 (ISO)	✓	✓		
Modbus	✓		✓	✓

3 Installation and Program Start

3.1 Hardware Installation

3.1.1 PC-TCP/IP Interface

A standard Ethernet card is sufficient for TCP/IP. For information on installation and configuration conventions, see the user's guide of your network card.

3.1.2 PC-H1 Interface

You can implement the H1 interface of your PC with the INAT PC-H1 communications package. For detailed information on installation and any configuration conventions, see the manual of the INAT PC-H1.

3.1.3 S5 Interface TCP/IP & H1

You can implement the TCP/IP or SINEC® H1 interface for the SIMATIC® S5 with communications processors INAT S5-TCP/IP or S5-H1. For information on installation and configuration, see the applicable manual.

3.1.4 S7 Interface

You can implement the TCP/IP or SINEC® H1 interface for the S7 PLC with communications processors Siemens S7-443/1 and S7-443/1TCP. For information on installation and configuration, see the applicable manual.

3.2 Software Installation

1. Insert the CDROM OPC serve.
2. Start the Setup.exe program. You will be guided through the installation.
3. If you use a OPC server that supports the H1 protocol, the H1 driver will automatically be copied to the hard disc during the installation. c:\INAT\TCPIPH1 is suggested as directory
   e.g. c:\INAT\TCPIPH1\PCH1

NOTE!

The directories which you specify will be set up automatically if they do not already exist.

4. After installation, the window „network“ in the system control opens. Here you have to add a new protocol. Confirm the button „disc“ and enter the directory where the h1 driver has been saved. After the restart of your PC the server can be licensed.

NOTE!

Before licensing your OPC server the H1 driver has to be installed and the PC has to be restarted

3.3 Program Start

1. Start your PC and the operating system.
2. Click the icon of the OPC server. The program is started, and the start dialog window appears (the window differs from server to server).

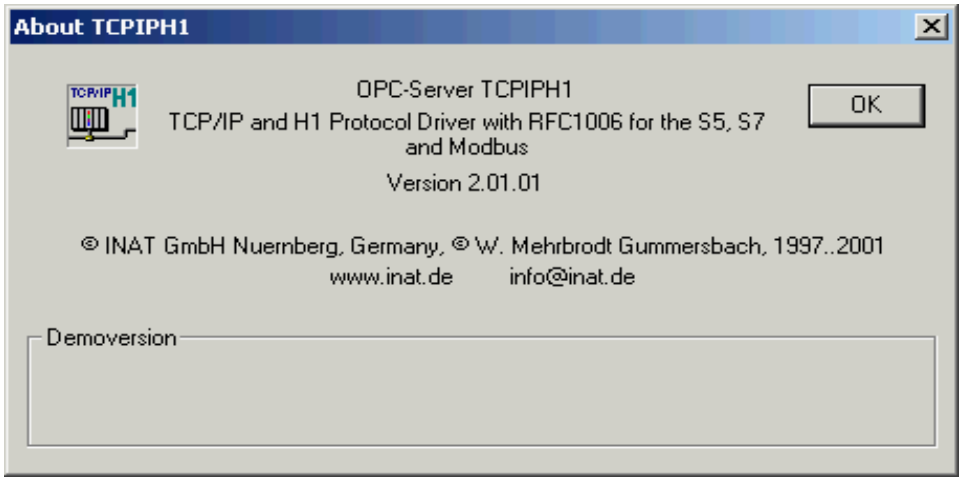


FIGURE 3.
Start dialog window of the OPC server

3. If you press the F1 key while this window is being displayed, the licensing conditions of the OPC server appear.
4. If you click OK, the main dialog window appears. The menu items and input options of this window will be discussed in the next chapter. After approximately 4 seconds, the start dialog window is automatically replaced by the main dialog window.

4 Dialog window and Functions of the INAT OPC servers

4.1 General

4.1.1 Starting the Program

The conventional method of starting the program is described in chapter „Program Start“ on page 24. The INAT OPC servers also recognize the command line parameter **/D:path**. If this parameter is specified, the OPC servers read the parameters from the TCPIPH1.NET file to the directory path specified under path. However, if the parameter is not specified, the path from which the TCPIPH1.NET file is read is indicated during the configuration dialog for OPC server settings.

4.1.2 Using Online Help

No matter where you are in the program, you can request help with the <F1> key. You will be given help appropriate to the context for the particular dialog box or menu item. In addition, you will find help buttons at many locations within the program which will provide you with further information on that particular topic by pressing the left mouse button.

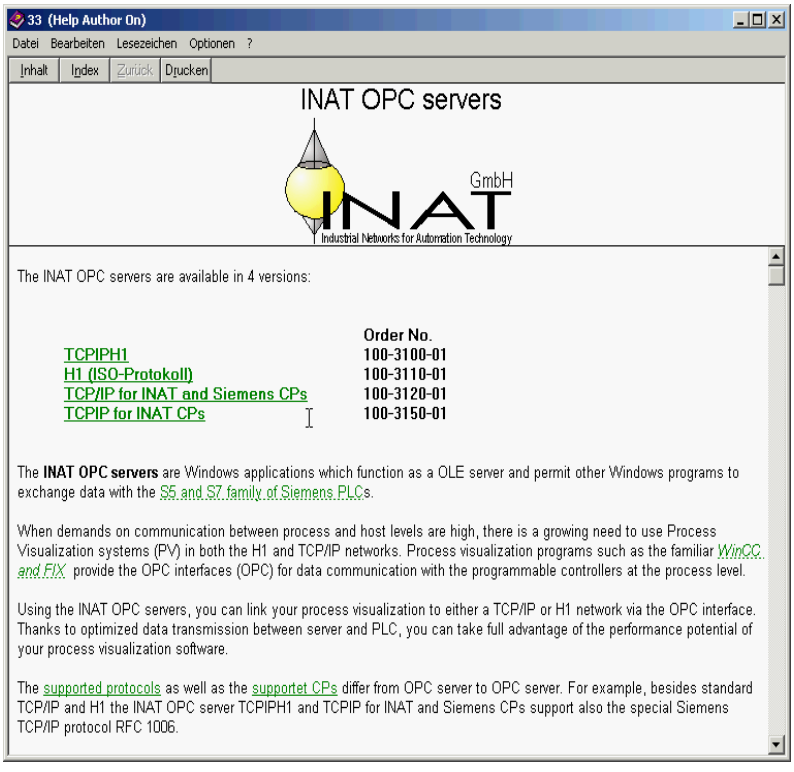


FIGURE 4.
Help dialog window

NOTE!

Maximize the window if you want to view all entries in the list.

If you would like to learn how to use the help functions, press the <F1> key while the help window is open.

Portions of the help text are highlighted. These highlighted portions contain cross references to related topics above or below this level. One of these cross references can be selected with the <Tab> key. Press the Enter key to branch to the cross-referenced help topics. If you are using a mouse, you can accomplish the same thing by double-clicking the cross reference with the left mouse button.

The following buttons are used for the help dialog window (see next page).

. .

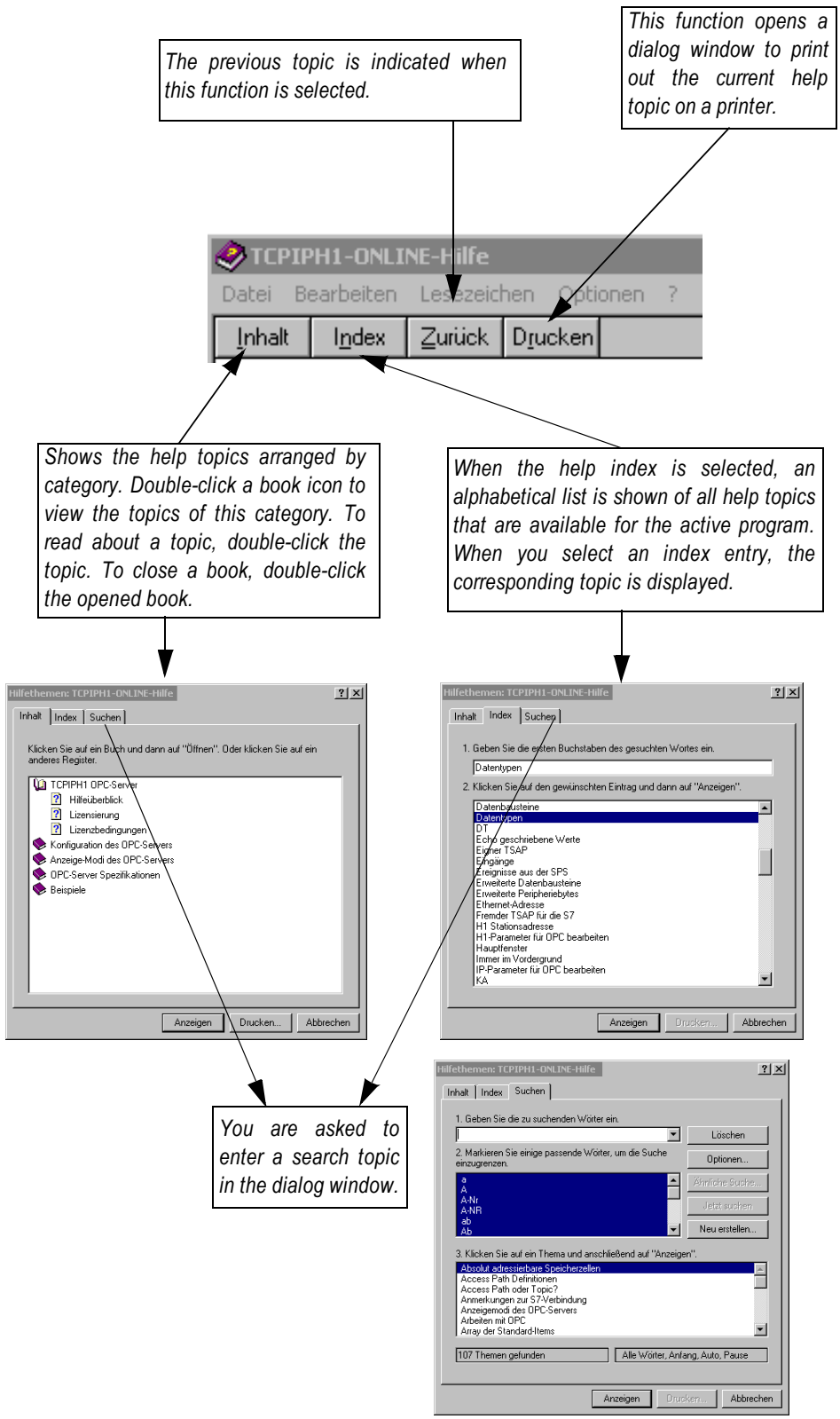


FIGURE 5.
Using the help dialog window

NOTE!

Maximize the window if you want to view all entries in the list.

4.2 Icons in the main window

Symbol	Meaning	Symbol	Meaning
	***Acces Path Definitions*** *Here you can define and delete the connections. You also have the possibility to edit the connection parameters for H1 and TCP/IP connections.*		***Show Logger*** *Here you get listed the whole data traffic of a running OPC communication.*
	***Server Settings*** *This dialog box is used to change the timer interval protocol, change the standard directory of the configuration file and set the operating mode to "unlimited repetition of write-access attempts".*		***Show Symbols*** *In the symbol view you can see the symbolic item names, you have entered in the text editor, according to the Access Path.*
	***Server Optimization*** *Here you may regulate the optimization of write accesses in three grades.*		***Edit Symbols*** *With this function, the server will be prepared for the alias browsing with OPC clients. The standard windows editor is opened.*
	***Server Options*** *Here you select the string format*		***Always on top*** *If this view is activated, the window of the OPC server will be shown on the top of the Windows surface.*
	***H1 station address*** *In this dialogue you are able to display the status of the network adapter and to change the current runtime address if neccessary.*		***Help Contents*** *Here you reach the online help.*
	***Logger Settings*** *Each event can presented with "Date" and "Time".* *With „ Cursor at the end“ the logger shows always the last message at the end of the list.*		***License*** *Here you can license the available software.*
	***Logger Clear*** *The previous entries of the logger will be deleted here.*		***About TcpIpH1*** *Here you get information about the version number and the copyright of the available software.*
	***Refresh*** *The status of the active access paths and the activated read areas can be updated here.*		***Exit*** *Here you can exit the OPC server.*
	***Show Status*** *The status of the active access paths and the activated read areas are indicated here.*		

4.3 Functions of the main window

The main window of the INAT OPC servers show you the system project programmed in the process visualization software. The main window of the OPC servers is shown below.

During visualization, the status of the active access paths together with the activated read areas are indicated in the main window. These values are not cyclically updated automatically. You can obtain an update by clicking the menu item „Refresh“ or by double-clicking the window itself.

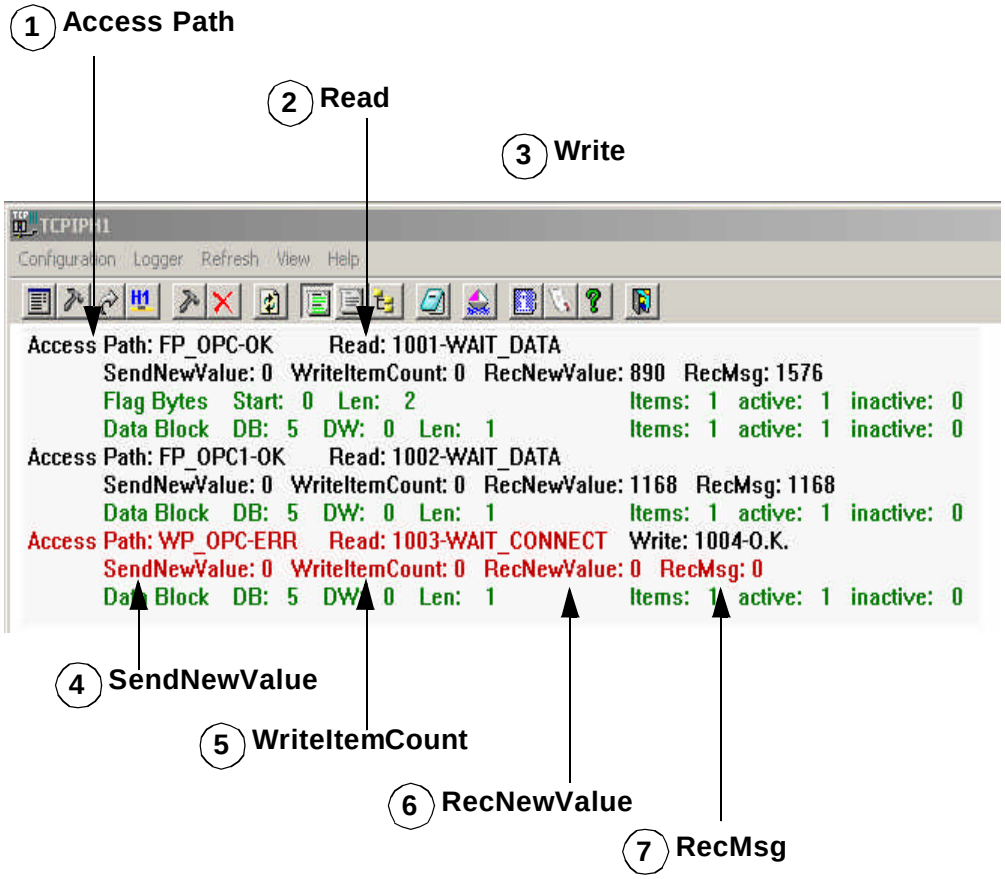
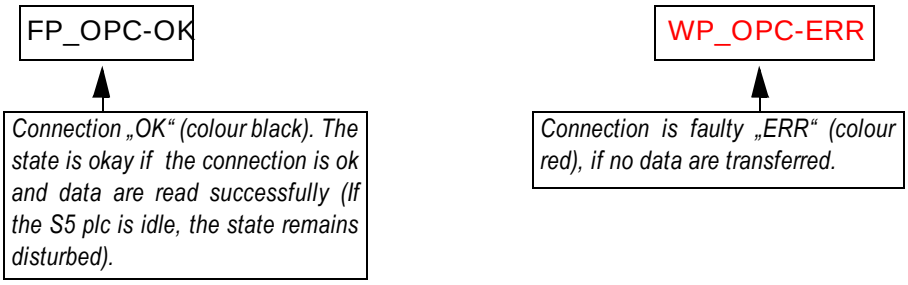


FIGURE 6.
The main window of the OPC server

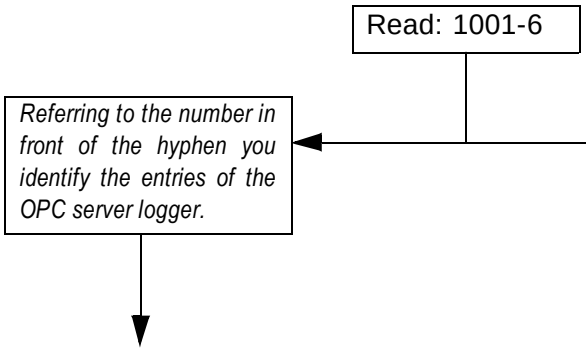
(1) State of Access Path FP_OPC-OK

Access path (here FP_OPC, FP_OPC1, WP_OPC): Name of the access path, which was specified as the name of the connection during the parameterization.



(2) Read (FETCH)

Run time number of the fetch connection:



An overview of all possible error codes is given in chapter „Description of the special items“ on page 75 under WriteStatusMsg / connection errors

Number in front of the hyphen	Type of connection
0	connection is not used
1-999	H1 connections
> 1000	TCP/IP connections

Number after the hyphen	Message
0	no error
3	WAIT_CONNECT
6/7	WAIT_DATA/ WAIT_SEND

(3) Write (WRITE)

Run time number of the write connection

Referring to the number in front of the hyphen you identify the entries of the OPC server logger.	*The number after the hyphen shows the connection state.*

Number in front of the hyphen	Type of connection
0	Connection is not used
1-999	H1 connections
> 1000	TCP/IP connections

Number after the hyphen	Message
0	no error
3	WAIT_CONNECT
6/7	WAIT_DATA/ WAIT_SEND

(4) SendNewValue (logger entry: Show Send Values)

SendNewValue: 0

Number of items, which are written to the OPC server up to now. If the writing speed of the visualization is very fast, the OPC server optimizes and is writing only the last written value to the PLC. Due to this procedure a difference between “new value” and the special item “WriteItemCount” can occur.

(5) WriteItemCount (logger entry: Show Send)

WriteItemCount: 0

Number of values, which were written from the server to the PLC, up to now.

(6) RecNewValue (logger entry: Show Rec Values)

RecNewValue: 890

Number of items which were written from the server to the vizualiztation, up to now. Since the data are transferred from the server to the visualization only in case of a value change, this value is an important clue for the OPC traffic.

(7) RecMsg (logger entry: Show Receive)

RecMsg: 1576

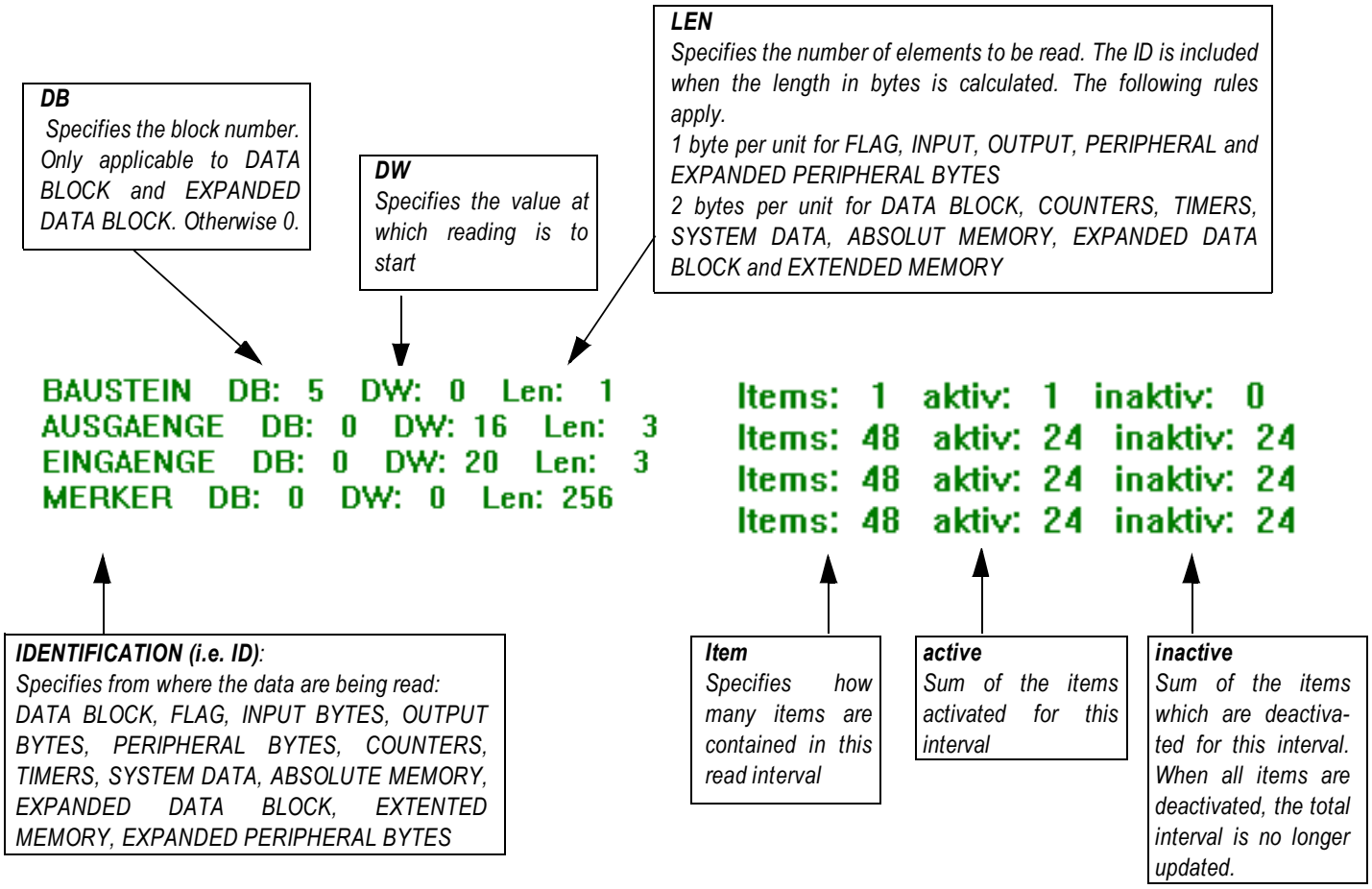
Number of frames, which the OPC server has read from the PLC, up to now.

The info line of the topics provides a list of data read from the PLC or written to the PLC.

Text colour

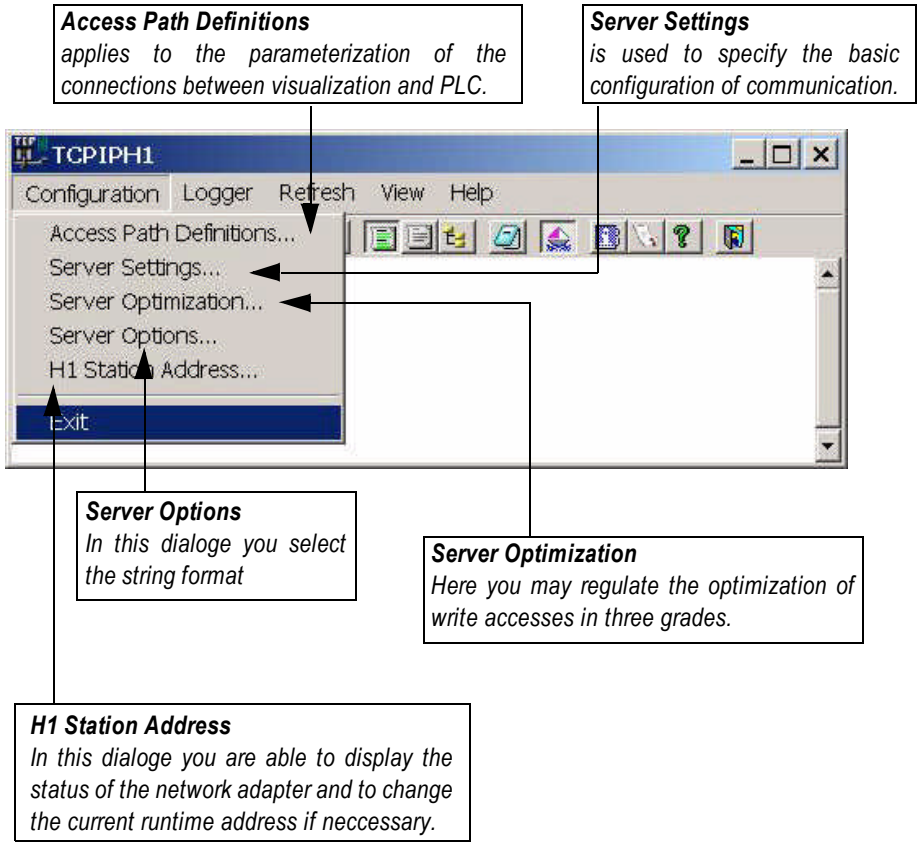
Colour	Meaning
Green	the interval is being processed
Gray	no data are being requested
Red	the data block is not available or too short

The values have the following meaning (see also chapter „Description of the special items“ on page 75):



The „Configuration" menu offers a choice of the items „Access Path Definitions", „Server Settings", „Server Optimization" and „H1 Station Address". You also have the possibility to exit the OPC server.

FIGURE 7.
Configuration menu



4.4 Access Path Definitions

A list of all parameterized connections is indicated. The names of these connections correspond to those of „OPC Access Path Definitions“ in process visualization.

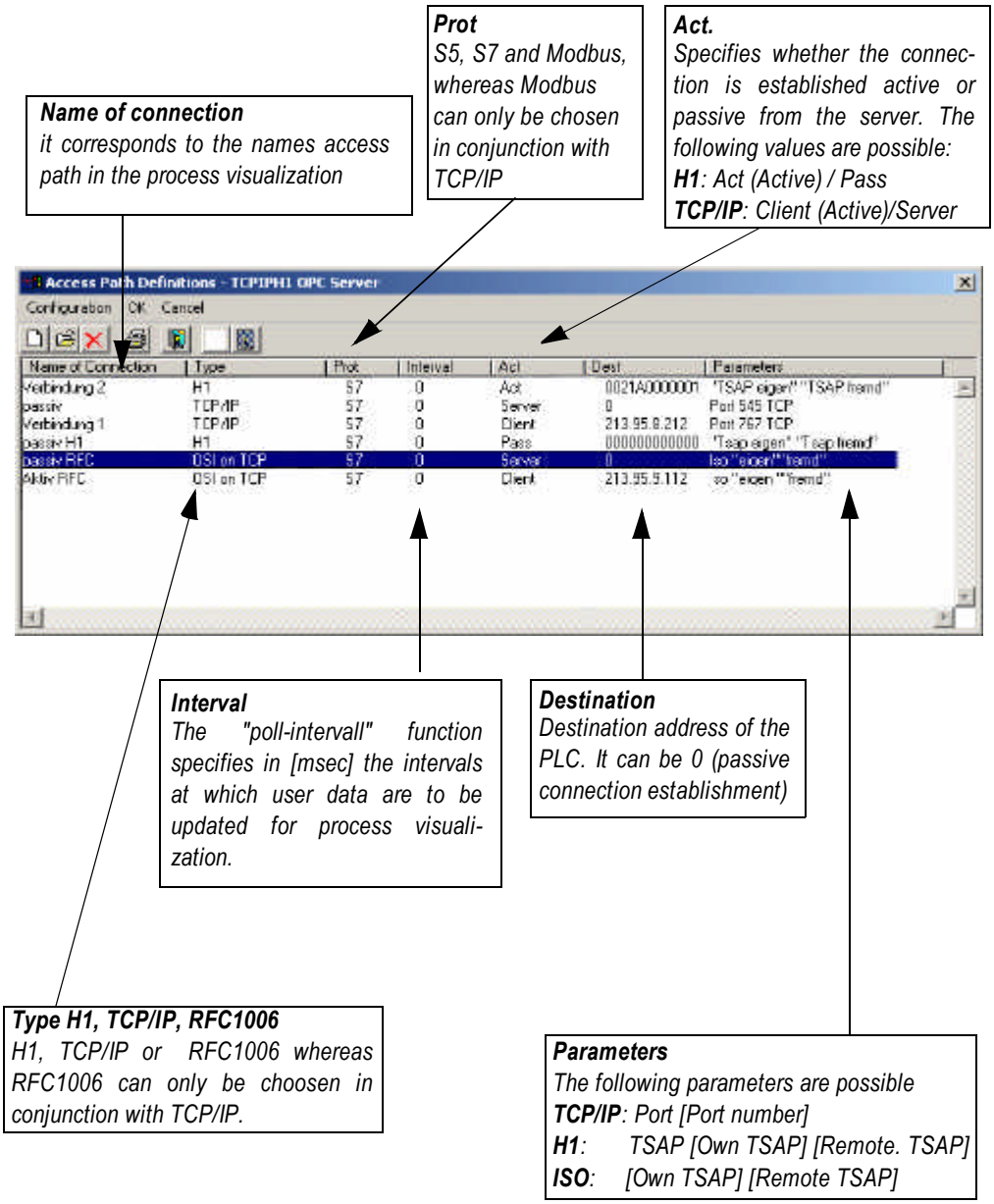
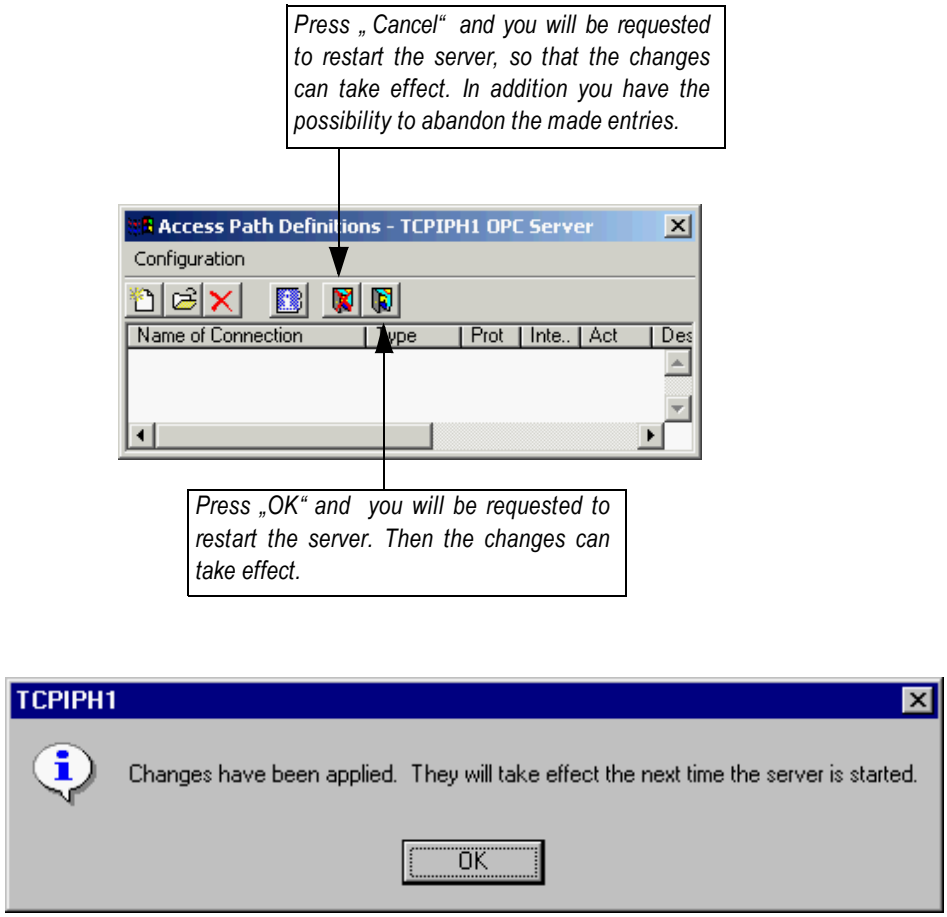


FIGURE 8.
Dialog box for parameterization of the server

FIGURE 9.
Dialoge box to execute the changes



4.4.1 New connection

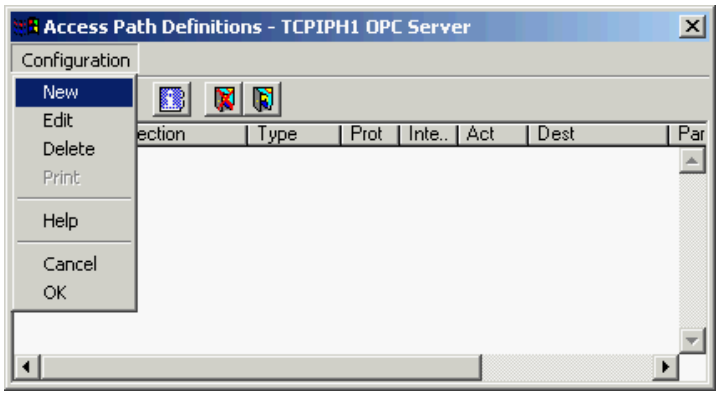
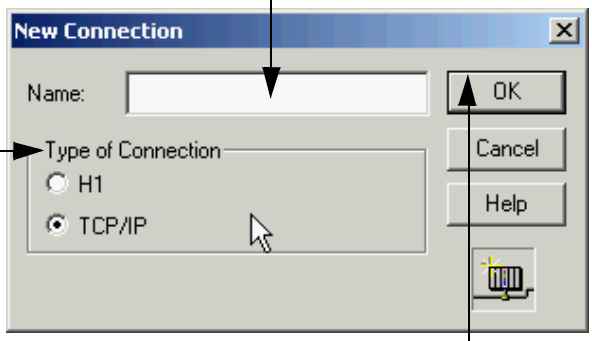


FIGURE 10.
Dialog box for creating a new connection

The following entries are available:

***Name**:*
All parameter data are assigned to the connections. Each connection is identified by a connection name. The connection name may contain all characters including blanks. The name may have a length of up to 27 characters.

Type of connection
H1:
A connection can be handled with the SINEC®H1 telegram or with the TCP/IP telegram. Connections between controllers should be handled by H1.
TCP/IP:
Connections are primarily used for the link to host computers (usually UNIX systems).



Click the OK button after making your entries. Depending on the selection under "Type of connection," either H1 or TCP/IP parameterization is called.

NOTE

OPC server TCPIPH1	Choice between H1 and TCP/IP (as shown above)
OPC server H1	No choice, only H1
OPC server for INAT and Siemens CPs	No choice, only TCP/IP
OPC server for INAT CPs	No choice, only TCP/IP

NOTE

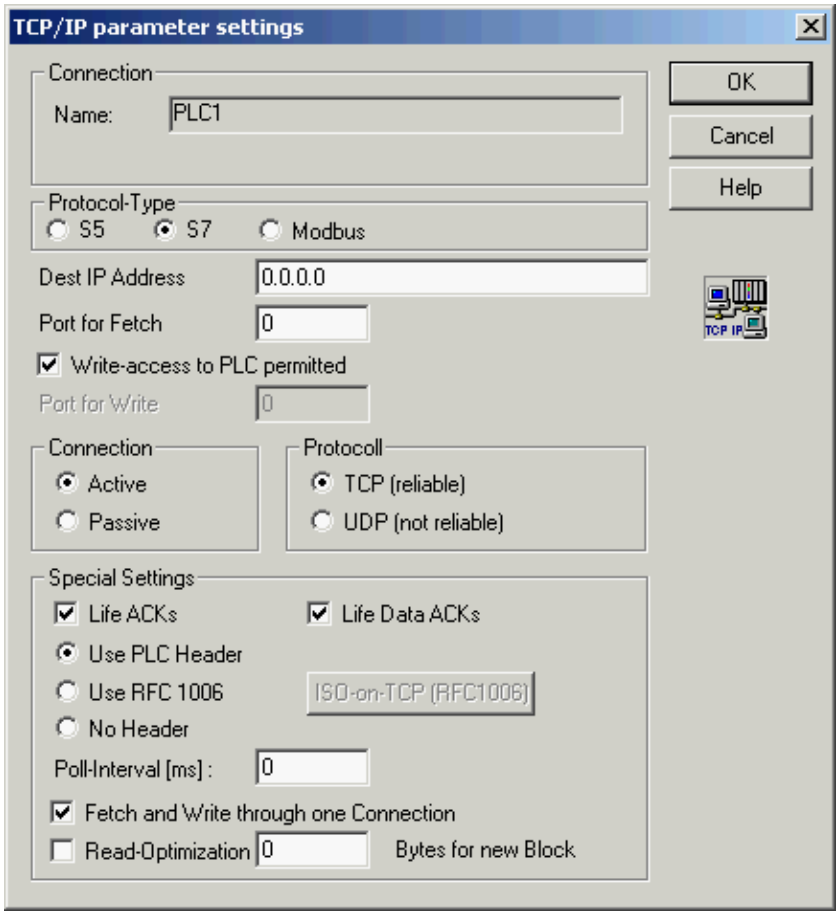
Users of the OPC server H1

The following chapter doesn´t pertain OPC server H1. You can go to chapter „H1 Parameterization“ on page 46

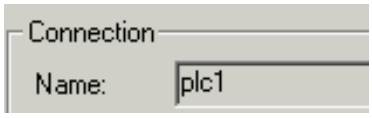
4.4.2 TCP/IP Parameterization

The PLC is linked to the control computer with Ethernet and the TCP/IP protocol. The TCP/IP protocol communicates at the system level. Each computer is addressed with the destination address together with the port address. The following entries are available:

FIGURE 11.
Input screen for editing TCP/IP parameters

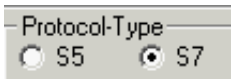


Connection Name



Under Connection/Name enter the name of the connection. The name of connection corresponds to the Access Path.

Protocol Type



If the button „S7“ is set, you can communicate with S7 PLCs. If the button „S5“ is set, you communicate via S5.

Dest IP Address

Each computer is addressed by a unique destination address. "Dest IP address or name of the destination station" is a 32-bit number specified in four groups of numbers. Each number may be from 0 to 255. The groups are separated by a dot. A symbolic name can also be entered in this field if the computer is configured for DNS and a DNS server is available in the network to convert this symbolic name to an IP address.

Example for a dest IP address

IP address 193.0.9.4

If you want to use regular names for the TCP/IP addresses instead of numbers, you must specify the address of a Domain Name Server. This server converts all symbolic TCP/IP addresses to IP addresses. The address of your DNS server can be obtained from your SysOp or from your provider (DNS, see also Chapter). After you have specified a DNS server and this server is ready for operation, the name station1.inat.de will be indicated instead of "193.0.9.4."

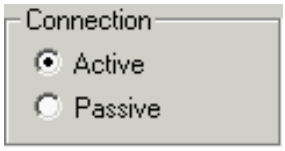
Port for Fetch

Port for fetch is the port number for the read connection. The port number is the channel number via which a connection is identified. It can be a number between 1 and 65535. It corresponds to TSAP for H1.

"Write-access to PLC permitted" (Port for Write)

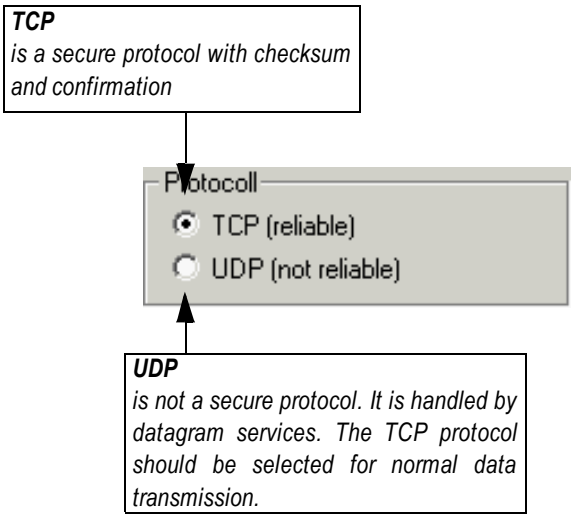
This item specifies whether a write access to the PLC is to be established. Write-accesses can be disabled in general here. The port number specifies the channel number via which a write-access is established. Port numbers for fetch-access and write-access may never be identical. Attention! Pleae use “port 102”, if you communicate via RFC 1006.

Connection Active Passive

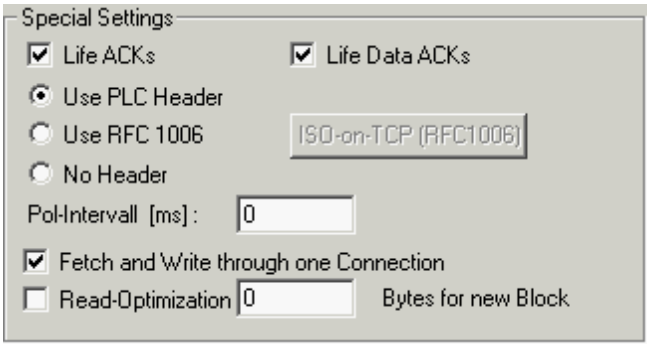


specifies whether the own station actively initiates connection establishment or passively waits for the connection to be established by the other station. The same value may not be entered for both sides of the connection. The PC is usually parameterized active. This is also the default value.

Protocol TCP/UDP



Special Settings



Life Acks / Life Data Acks

This item activates the connection monitoring function which is restricted in many socket libraries. These frames are used to transfer the connection status to process visualization.

If you are using the OPC server in a LAN with networked controllers, we recommend activating both functions. If the connection is being handled by a WAN, we recommend deactivating these functions to save costs.

The default setting of the life acks and data life acks is ”activated”.

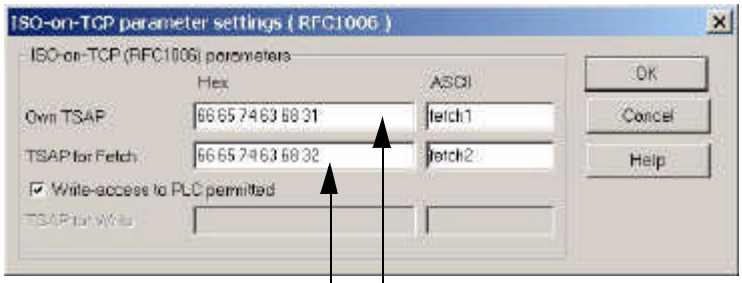
Use PLC Header

If you activate this button for the TCP/IP communication you are provided with the optimized data transmission to the S5-TCP/IP. Apart from a frame counter the PLC Header contains also length information and the possibility so send acknowledges (see also chapter „The PLC Header“ on page 133).

Not relevant for OPC server TCPIP for INAT CPs

Use RFC1006

With the activation of RFC1006 the H1 frames, which have to be transmitted, are implemented in a TCP/IP frame. With that sepcial TCP/IP channel you are able to communicate with the Siemens CPs of the S5 and S7 series.

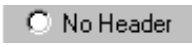


For the ISO parameters "Own TSAP", "TSAP for FETCH" and "TSAP for WRITE" particular rules are valid:
The TSAP entries should have at least the length of two characters and at most the length of eight characters.
More information about TSAPs you will get in the chapter „H1 Parameterization" on page 46" H1 parameterization".

NOTE!

Note Pleae use "port 102", if you communicate via RFC 1006

No Header

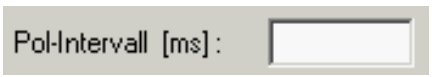


The header at the beginning of the frame can be disabled. If this is done, the application program on both sides is responsible for monitoring. Keep the following points in mind.

- ***Particularly in the case of the Send Direct and Receive Direct jobs, certain time limits until frame acceptance may not be exceeded. If these time limits were violated, the internal buffers would be overloaded (e.g., with inquiries), and synchronization of request and response would no longer be possible.***
- ***The receiving side must ensure that the frames are read from the receiving buffer before the partner station sends the next frame.***
- ***A certain mechanism for blocked data transmission must be adhered to so that the end of the user data can be detected.***

Creation of connection monitoring in the application program is indispensable.

Pol Interval



The "Pol-Interval" function specifies in msec the intervals at which user data are to be updated for process visualization. To keep network load as low as possible, the intervals should be based on the importance of the particular user data (i.e., as short as necessary but as long as possible). If the specified read interval is shorter than the actual transmission, the data are read as fast as possible.

Fetch and Write through one connection

This function is activated automatically. The S7 CPs provide only a limited number of connections for communictaion. The setting „Fetch and Write through one connection“ saves resources.

We recommend to use this function at OPC communication via echolink, too.

Read Optimization

With activation of the radio button Read optimization, the OPC-Server combines data bytes to blocks of a size, which can determine under Bytes for a new block. This procedure is very helpful if large gaps exist in the datablock structure of your plc.

Example:

If you enter the value 100 Byte, all data ranges of 100 bytes and bigger are ignored by the server, if they contain no data. In this way you are able to adjust the read procedure to the data structure of your PLC.

Recommended Values

Minimum of 10 Bytes

For S5 PLCs **512 Bytes**

For S7-300 PLCs **212 Bytes**

For S7-400 PLCs **400 Bytes**

Syntax of IP addresses

A TCP/IP address is a 32-bit number which is specified with four groups of numbers. Each number can be from 0 to 255. The gruops are seperated by a dot.
***Example**: 192.0.9.4*

Some IP addresses are reserved and should not be used:

0.0.0.0
255.255.255.255
127.x.x.x (x = any number)

If the values outside your subnet mask are set to „1“ , as for example 255.255.255.255.0, then the value x.x.x.255 should not be used!

If xou want to specify the TCP/IP addresses as regular names and not numbers, you must specify the address of at least one Domain Name Server (DNS). This server converts all symbolic TCP/IP addresses to the numerical TCP/IP addresses. This address is used to establish the connection.
The address of your DNS server can be obtained from your SysOp or your provider.
After you have specified a DNS server and it is ready for operation, www.inat.de will be indicated instead of 193.0.9.4 for example.
Names are easier to use than numbers since people remember names better than numbers. Names are always used in Internet.

-

NOTE

Users of the OPC servers TCPIP for INAT (and Siemens) CPS
The following chapter doesn´t pertain your OPC server.

4.4.3 H1 Parameterization

Ethernet H1 connections have many parameters which are governed by several rules.

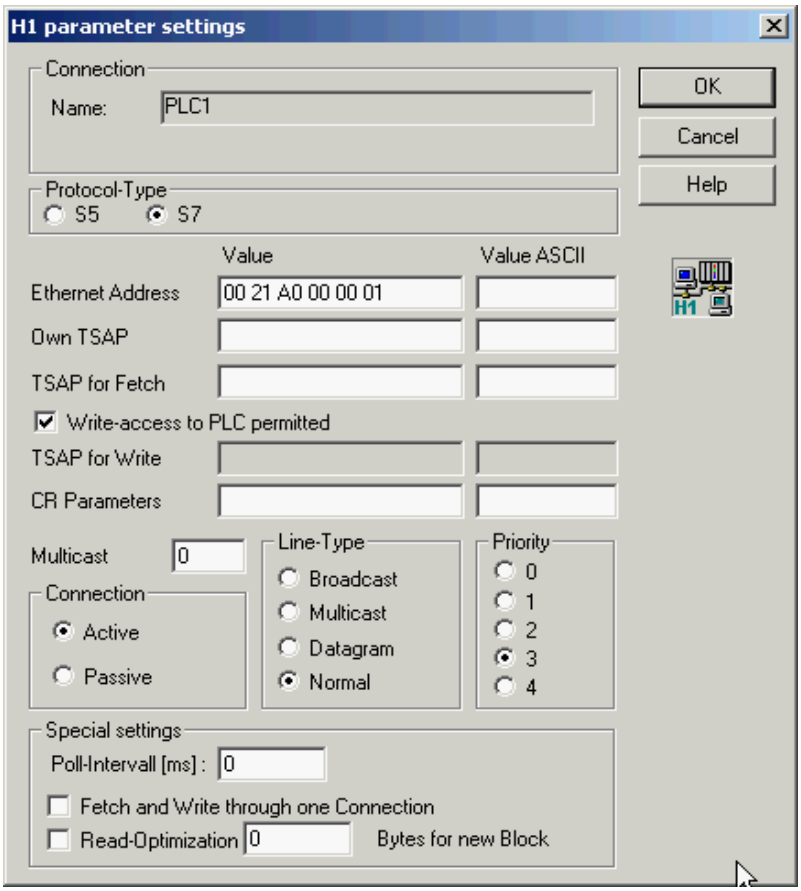
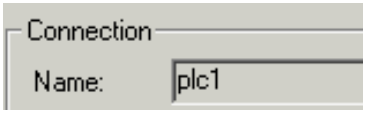


FIGURE 12.
Input screen for parameterization of an H1 connection

NOTE!

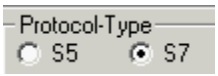
In systems, these Ethernet parameters are usually specified centrally by the network manager. Experimenting with your own Ethernet parameters may affect the stability of the entire network.

Connection Name



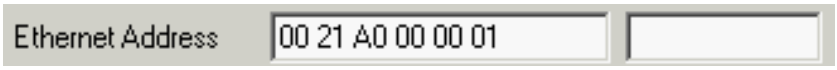
Here you enter the name of the connection. The connection name corresponds to the names Access Path. When H1 is used, connections cannot be assigned a symbolic name.

Protocol



If "S7" is set you can communicate with PLCs of the series 700. If "S5" is set you can communicate with PLCs of the series 500

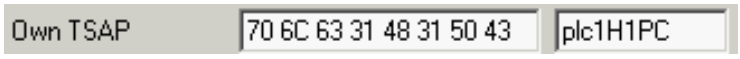
Ethernet Adress



Ethernet addresses are binary. All byte states from 0 hex to FF hex may occur. The other station (e.g. a PLC) is determined with the address. Such ethernet addresses are always 6 characters in length. The first three characters specify the manufacturer of the destination system. These bytes are assigned centrally by the IEEE committee. If the manufacturer´s code ist not specified, make sure that the first byte is an even value (i.e., ist must be divisible by two without remainder). The last three bytes can be specified as desired. Two or more stations with the same ethernet address are not permitted in one network.

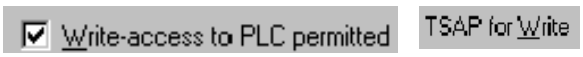
***IEEE**:*
Treats the data stored on the PLC as 4 Byte Real and converts it to OPC real format.

Own TSAP, TSAP for Fetch



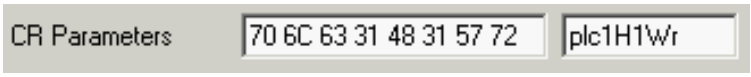
The Own TSAP specifies the connection address in the own system via which data are to be exchanged. „Remote TSAP“ specifies the connection address of the other system. To be able to establish the connection, the own TSAP must correspond to the external TSAP of the other system. This requirement is easy to meet since „Own TSAP“ and „TSAP for Fetch“ may be the same but „TSAP for Fetch“ and „TSAP for Write“ may not. Although the length of such a TSAP is often 8 bytes, it can have a length from 1 to 16 bytes. SINEC systems use the value 20 for the first 3 bytes.

Write-access to PLC permitted / TSAP for Write



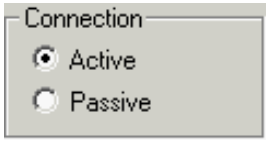
Specifies whether a write connection to the PLC is to be established. Write Accesses can be disabled in general here via this access path. The TSAP for Write (i.e., Transport Service Access Point) specifies the connection address for write jobs.The length of a TSAP is often 8 bytes, but it can also vary from 1 to 16 bytes. SINEC systems use the value 20 for the first 3 bytes.

CR Params



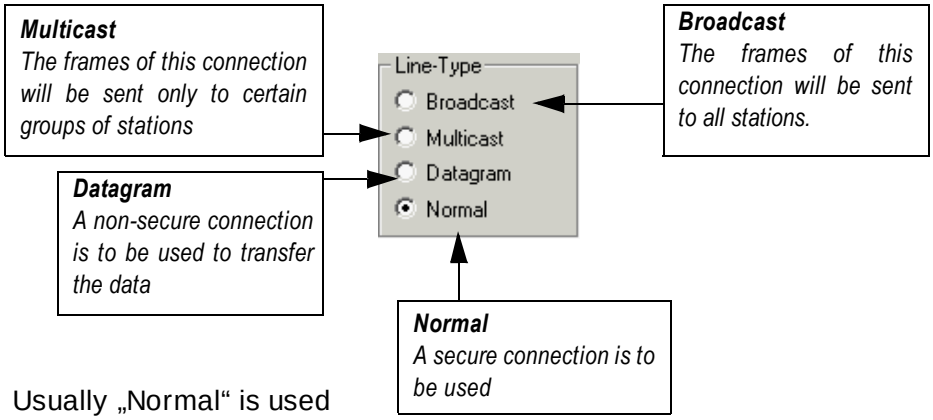
specify the operating mode of the destination system and are not specifed anywhere. Since some H1 interface converters use this to specify the parameters of the second (i.e., serial) interface, you will have to check the user´s guide of the destination system to determine whether entries must be made here, and, if so, which entries.

Connection active / passive



specifies whether the own station actively initiates connection establishment or passively waits for the connection to be established by the other station. The same value may not be entered for both sides of the connection. The PC is usually parameterized active. This is also the default value.

Line Type

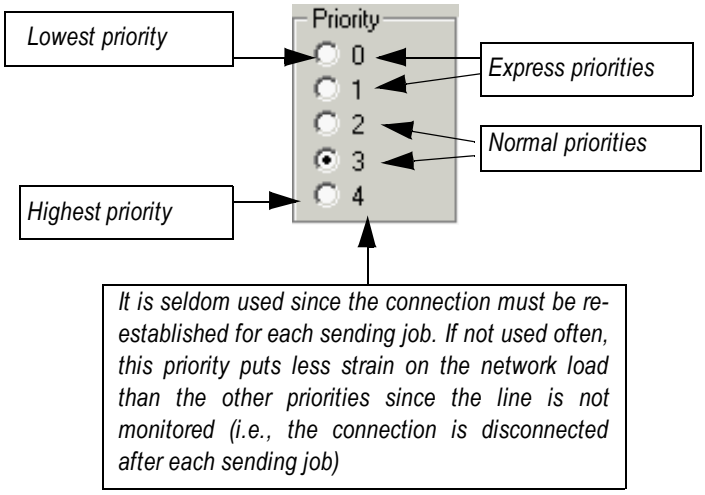


Usually „Normal“ is used

Multicast

specifies that the device to be parameterized belongs to a certain group of stations which are to be reached when a multicast connection is used. The multicast circle window only applies when multicast has been selected as the type of line.

Priority



NOTE!

Remember that the express priorities do not increase the transmission speed any more than the normal priorities. For some controllers, the data are transferred to the RAM with an interrupt when priority 0 is used. This may increase the speed of the total data transmission. When priorities 0 and 1 are used, the data length may not exceed 16 bytes. Priority 3 is usually used her.

Pol Intervall

The read interval function specifies in ms the interval at which user data are to be updated for process visualization. To keep the network load as low as possible, the intervals should be based on the importance of the particular user data (i.e., as short as necessary but as long as possible.

4.4.4 Remarks on the S7 connection

- The recent versions of the OPC servers enables also the communication with the S7 PLCs S7-300 and S7-400.
- If you want to establish a connection to one of these S7 modules, please choose the button „S7 connection“ as described in the chapters „TCP/IP Parameterization“ on page 38 and „H1 Parameterization“ on page 46.
- Furthermore the OPC server supports the simultaneous access to a S5 and a S7 PLC.

4.4.4.1 S7 and the Standard Items

- Internally the S7 is working with bytes; for that reason the bit range of the DBs reaches from 0 to 7.
- With S7 formats DL, DR, DW and DD can also be read. But if „1“ is written to DW1, DD1 contains a 65536!!
- There is a new item name „DBxDBy“ data block x data byte y.
- The suffix IEEE represents the S7 real numbers as for example DB5DD24IEEE.

4.4.4.2 Writing not allowed

As for the S5 it is also possible for the S7 to read by one connection and write by another. This way writing can be suppressed here from a central point.

4.4.4.3 TSAP settings for the S7

TSAP are necessary for the RFC1006 and H1 communication. If that kind of communication shall be performed, take the following remarks into account:

„Own TSAP“ specifies the connection address in the own system via which data are to be exchanged. „Remote TSAP“ specifies the connection address of the other system. To be able to establish the connection, the own TSAP must correspond to the external TSAP of the other system. This requirement is easy to meet since „Own TSAP“ and „TSAP for Fetch“ may be the same.

Due to the presetted addresses in the S7-300 and S7-400 the remote TSAPs „TSAP for Fetch“ and „TSAP for Write“ are the same, too.

The presetted TSAP is specified by the device type, the rack no. and the slot no. (see also Remote TSAP). The length of the TSAP is 2 bytes.

Own TSAP (Server)

It is specified by two groups of byte. Each group contains two hex characters. The both groups are seperated by the space character.

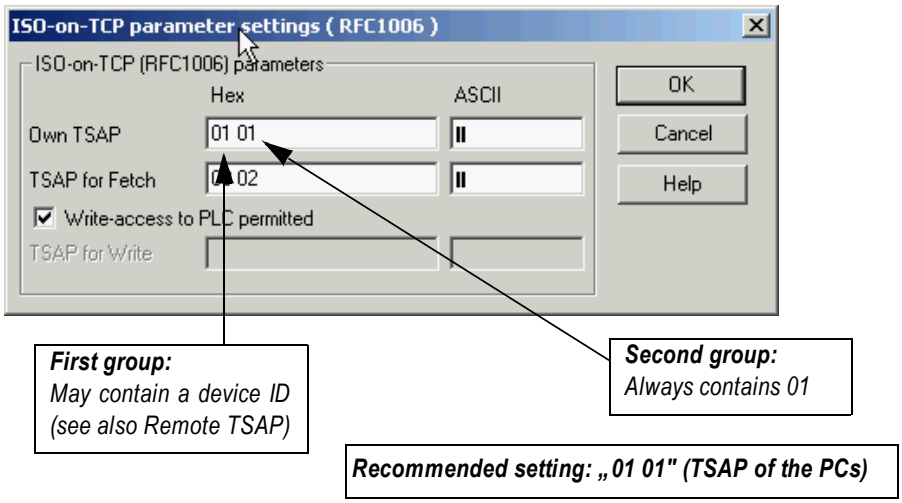
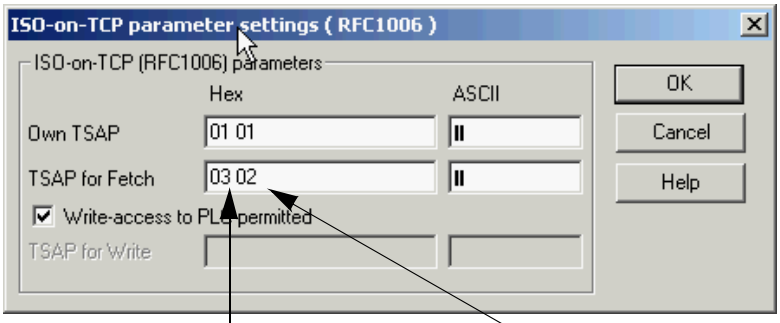


FIGURE 13.
HEX window of the TSAP dialogu „Own TSAP“

Remote TSAP („TSAP for Fetch" and „TSAP for Write")

The TSAP for the remote station is specified by two groups of byte. Each group contains two hex characters. The both groups are seperated by the space character.

FIGURE 14.
HEX window of the TSAP dialogu „Remote TSAP"



First group: *contains device IDs, for which ressources are provided in the S7. Possible IDs are:*
01 = PG
02 = OS (Control and monitor device)
03 = Other (i.e SIMATIC)

Second group: *contains the addresses of these components. For example:*
Bit (7..5) = Rack no.
Bit (4..0) = slot of the CPU01 = PG

Example:

03 02	*OPC communication to the S7-300/400, Rack 0, CPU slot 2*
03 43	*OPC communication to the S7-300/400, Rack 2, CPU slot 3*

How to calculate the rack/slot no. (<16)

Left character = Rack no. multipled by 2

Right character = slot

Special case:
The second group contains the value „00" if the device is addresses directly

NOTE!

Enter the TSAP values of the example into the HEX window of the TSAP dialogue

4.4.4.4 Parameterization of the S7-CP

For communication via ISO layer 4 (transport layer), e.g. with the OPC server, no parameter assignments for the S7 communication processor is necessary.

Parameter assignment is also unnecessary for layer 2 communication (e.g. PC networks such as NetWare, LAN-Manager with NetBEUI, or TCP/IP). If you want to communicate only via ISO layer 2 or 4, you only have to enter the TCP/IP or H1 destination address. For that case the connections are predefined.

4.5 Configuration / Server Settings

This dialogu box is used to change the timer interval protocol, change the standard directory of the configuration file and set the operating mode to „unlimited repetition of write-access attempts.“

Protocol Timer Tick:
Enter in milliseconds the frequency at which the server is to check data from the process. This should be performed two to four times as often as the data update from the system. The OPC server suggests a standard time interval of 50 milliseconds. We recommend using an interval from1 to 10 milliseconds to start with.

Kompatibly:
select this mode if you want to use older applications with the recent version of the OPC server. Note: This mode doesn´t support arrays.

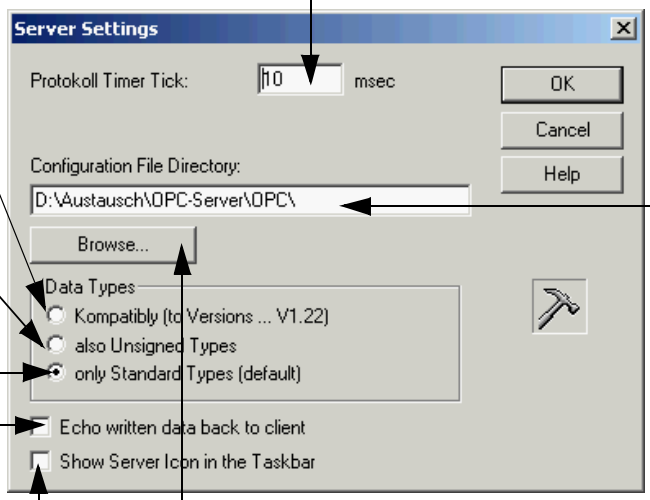


FIGURE 15.
Server settings

also unsigned types:
Select this mode if you are working with the visualization WinCC.

only Standard Types (Default):
this mode has to be set for most of the available OPC clients and therefore is default setting.

Browse
*Press this button if you want to search the directory, which contains the *.net configuration file. For the file name TCPIPH1.NET is presetted*

Configuration file directory:
To create a new configuration file, enter here the complete path of the directory where the file shall be stored. This new path is automatically stored in the WIN.INI file. The server will load this configuration file with the next startup. Make sure that this directory exists or create it. Only the path can be changed. The file name TCPIPH1.NET is presetted by the server.
Note! *There is no limit to the number of configuration files which you can create as long as the files are stored in different directories. Store the configuration files in your application directory.*

Echo written data back to client:
a constant value in the plc is overwritten by a client, it depends on the client, which of the values is displayed in the server.
Clients WinCC and WIZCON: mode has to be set!
Client iFIX: mode shall not be set!

If this icon is set, the server icon is displayed besides the system icons in the task bar.

4.5.1 Data types

Only Standard Types

According to the listed modes below, different data types are available for the OPC communication:

Tag Name	compatible up to version 1.22	also unsigned types	only standard-types (default)
DB5D1.0	VT_BOOL	VT_BOOL	VT_BOOL
DB5DB1	VT_I4	VT_UI1	VT_UI1
DB5DB1KF	VT_I4	VT_BSTR	VT_BSTR
DB5DL1	VT_I4	VT_UI1	VT_UI1
DB5DR1	VT_I4	VT_UI1	VT_UI1
DB5DW1	VT_I4	VT_UI2	VT_I4
DB5DW1KF	VT_I4	VT_I2	VT_I2
DB5DW1KT	VT_BSTR	VT_BSTR	VT_BSTR
DB5DW1KC	VT_I4	VT_UI2	VT_I4
DB5DW1BCD	VT_I4	VT_I2	VT_I2
DB5DD1	VT_I4	VT_UI4	VT_R8
DB5DD1KF	VT_I4	VT_I4	VT_I4
DB5DD1KG	VT_R4	VT_R4	VT_R4
DB5DD1IEEE	VT_R4	VT_R4	VT_R4
DB5S1.10	VT_BSTR	VT_BSTR	VT_BSTR
DB5DW1KF	VT_I4	VT_I2	VT_I2
DB5DW2KF	VT_I4	VT_I2	VT_I2
DB5DW0.10	----------	VT_ARRAY	VT_ARRAY

Compatible

select this mode if you want to use older applications with the recent version of the OPC server. This mode doesn´t support arrays.

Also unsigned types

select this mode if you are working with the visualization WinCC.

Only standard types (default)

this mode has to be set for most of the available OPC clients and therefore is default setting. The server returns data types, which corresponds to the specified item. A data word with the range 0-65535 cannot be stored in a signed 16 Bit variable. Hence choose the suffix "KF" with the range -32768 to 32767 (e.g. DB5DW0KF). If you choose the item without suffix, the corresponding data type is a signed 32 Bit value (LONG or VT_I4).

Storing several configuration files

There is no limit to the number of configuration files which you can create as long as the files are stored in different directories. You can access several configuration files by using the qualifier /d. This setting can be overwritten by specifying the parameter **/d: path**.

For example, if you want to start the OPC server with a configuration file from another directory (C:\data\project1), enter the following command line:

TCPIPH1.EXE/d:C:\data\project1

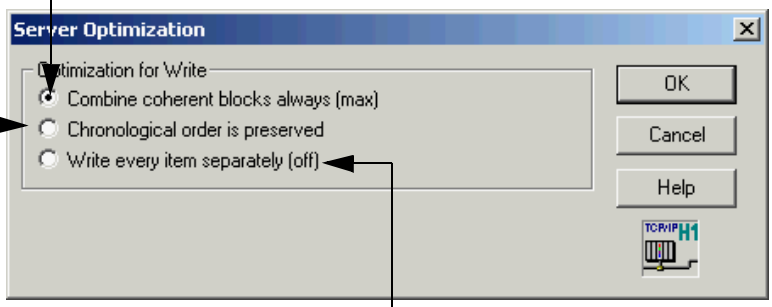
4.6 Configuration / Server Optimization

The OPC server is perfectly tuned for the communication with PLCs; two examples:

- ***If in a short time interval the same value is written three times consecutively, the server transmitts the value to the PLC only once.***
- ***If values have to be written which are changing very often within a short time interval, the server transmitts always the last value of that row to the plc.***

Due to the different cycle times of OPC clients on Win NT, 9X and 2000 computers and PLCs this data handling is very useful. With respect to that data handling you may regulate the optimization of write accesses in three grades (see next page).

1.Combine coherent blocks always (max)
This was the optimization mode in all earlier server versions. This maximum form of optimization is very useful and is set active on default because:
a) less PLC frames are transmitted, and thererfore
b) the network traffic is reduced!
If a counter increments very fast, only the last value (or the recent value) is written to the PLC



2.Chronological order is preserved
The items are queued in chronological order and are written to the PLC after the read access to the corresponding data block was successful. Coherent blocks are combined but overlapping items are written separately. Single bits of a word are also written separately. That performance provides a secure forwarding of data to the PLC even if a bit is resetted and deleted in very short time intervals. The correct data flow is provided.

3.Write every item separately (off)
In this case the optimization is off. The data handling corresponds with the process under 2) „Chronological order is preserved“

NOTE!

For 2) „ Chronological order..." and 3) „Write every item..." many and very quick write accesses may slow down the read accesses.

4.7 Configuration Server Options (version 2.03.00)

There´s a difference between S5 and S7 strings:

S5-String: no length indication

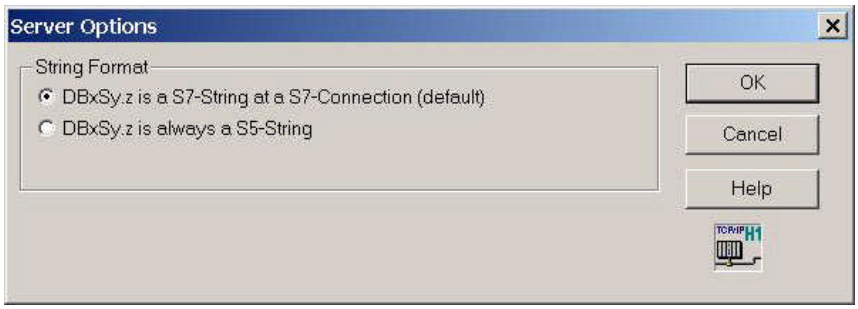
S7-String: in the first byte maximum length, in the second byte real length

The following notations are possible:

S7 syntax: db9.STRING0.10

S5 syntax: db9s0.10

S7 String: db9g0.10 (always forces a S7 string)



4.7.1 Examples S5/S7 syntax

	S7 Syntax	S5 Syntax	S7 String
BOOL	db9.X15.0	db9d15.0	
BYTE	db9.B14	db9dB14	
BYTE(CHAR)	db9.CHAR6 *	db9db6KF	
WORD	db9.INT4	(vorzeichenbehaf-tet) db9dw4KF	
WORD	db9.W12	db9dw12	
DWORD	db9.D8	db9dd8	
DWORD	db9.DINT8	(vorzeichenbehaf-tet) db9dd8KF	
REAL	db9.REAL0	db9dd0ieee	
FLOATING POINT		db9db60kg	
ARRAY	db9.B16.10 'Array of Byte **	db9db16.0	
STRING	db9.STRING0.10 ****	db9s0.10	db9g0.10 ****
TIME	db9.D0t	db6dd0t	
DATE	db9.W214d	db9dw214d	
DATE AND TIME	db9.W216dt	db9dw216dt	
FLAGS***	M10.0 BOOL MB10 BYTE MW10 WORD MD10 DWORD	M10.0 BOOL MB10 BYTE MW10 WORD MD10 DWORD	

* presentation ASCII/Hex depending on client

** arrays can be shown with any data format, except the data type BOOL, TIME, DATE and DATE AND TIME.

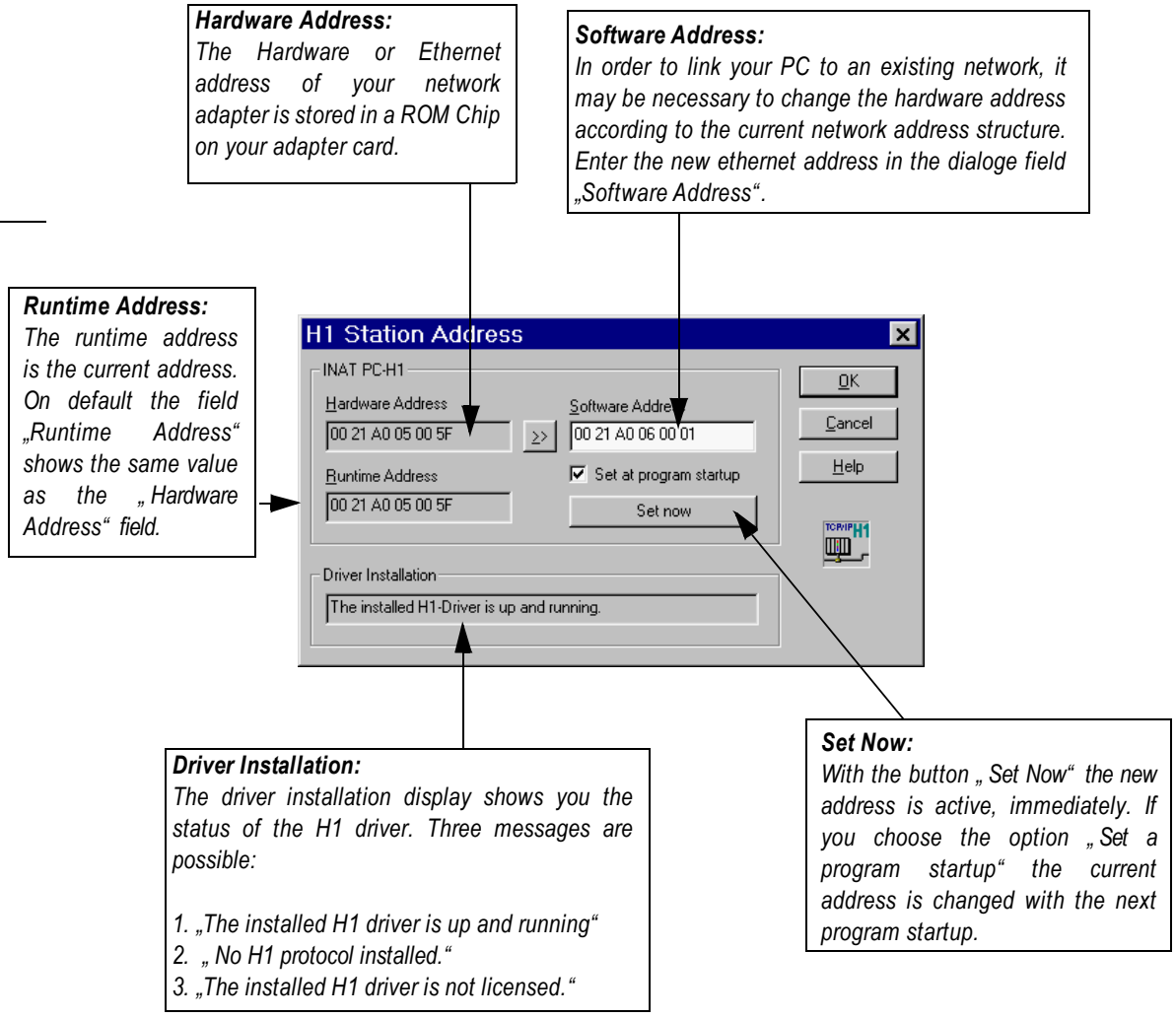
*** S-Flags can not be indicated, because they can only be addresses by the CPU.

**** Presentation depends on the choosen STRING FORMAT (only with S7 connection!!!). Available since version 2.01.06 under the point "Server Options"). The syntax db9g0.10 forces a S7 string at every adjustment

4.8 Configuration / H1 station address

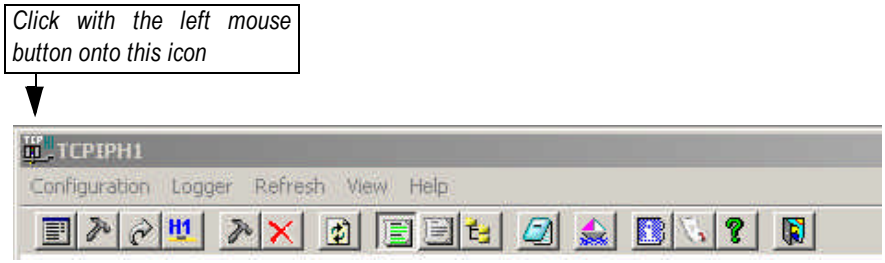
In this dialoge you are able to display the status of the network adapter and to change the current runtime address if necessary.

FIGURE 16.
H1 station address settings



4.9 The emergency-exit of the OPC server

Click with the left mouse button onto the ICON of the OPC server header.



Besides the standard functions as „Minimize“ or „Close“ you will receive some helpful tools for the OPC server communication.

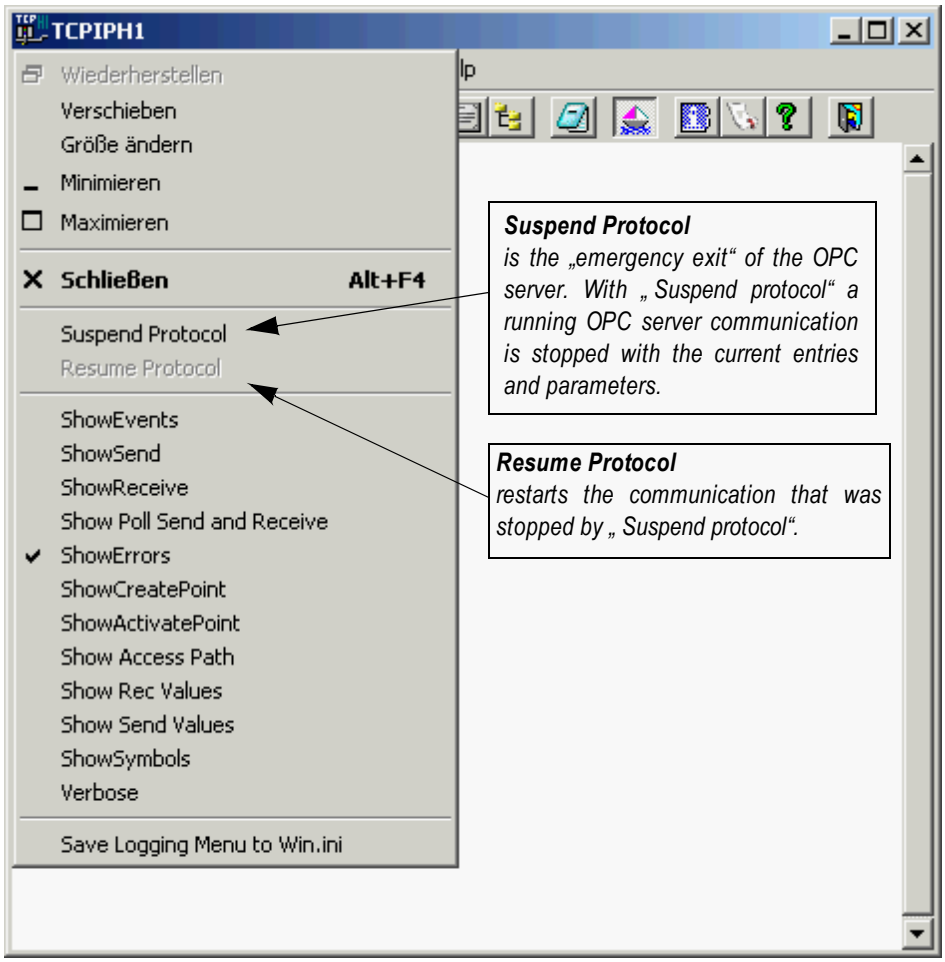
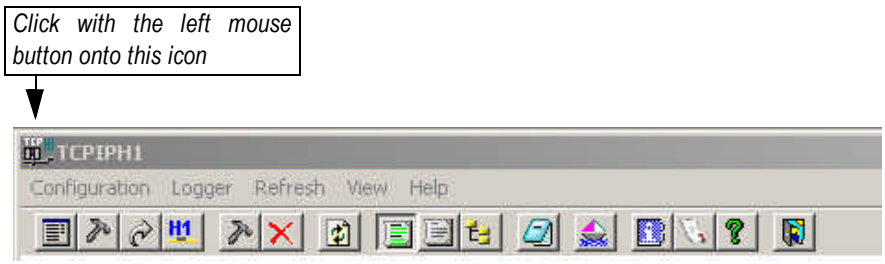


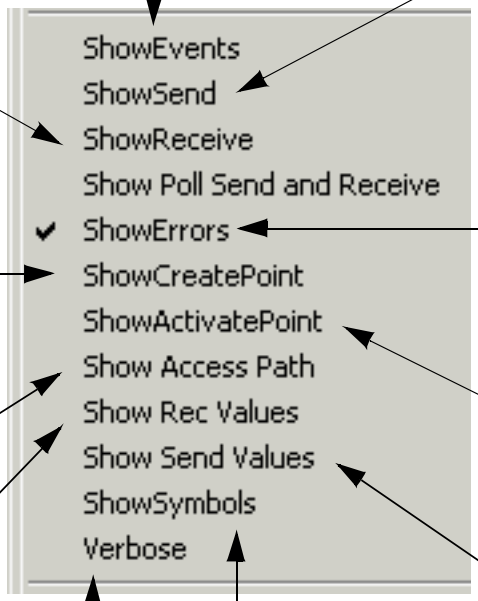
FIGURE 17.
Stop and Restart the server communication

4.10 The Logger of the OPC server

With the logger function of the server the entire data traffic of the current OPC communication can be monitored. Click with the left mouse button onto the ICON of the OPC server header.



Besides the standard functions as „Minimize“ or „Close“ you will receive some helpful tools for the OPc server communication (see next page).



ShowEvents: *activates the entries of a) changes in the topics state and b) other events*

ShowReceive: *Number of frames, which the OPC server has read from the plc, up to now.*

ShowSend: *Number of values, which are written from the server to the plc, up to now.*

ShowCreatePoint: *shows the sum of items, which were created for this connection / topic in the visualization and were registered in the server.*

ShowErrors: *Corresponds to the connection error for read and write in the main ogger entries, respectively. from the server to the visualization, up to now.*

ShowAccessPath: *shows the connections, which are triggered by the visualization.*

ShowActivatePoint: *is the sum of items, which are active for this topic at the moment.*

ShowRecValues*: RecNewValue in the main window: Number of items which were written from the server to the visualization, up to now.*

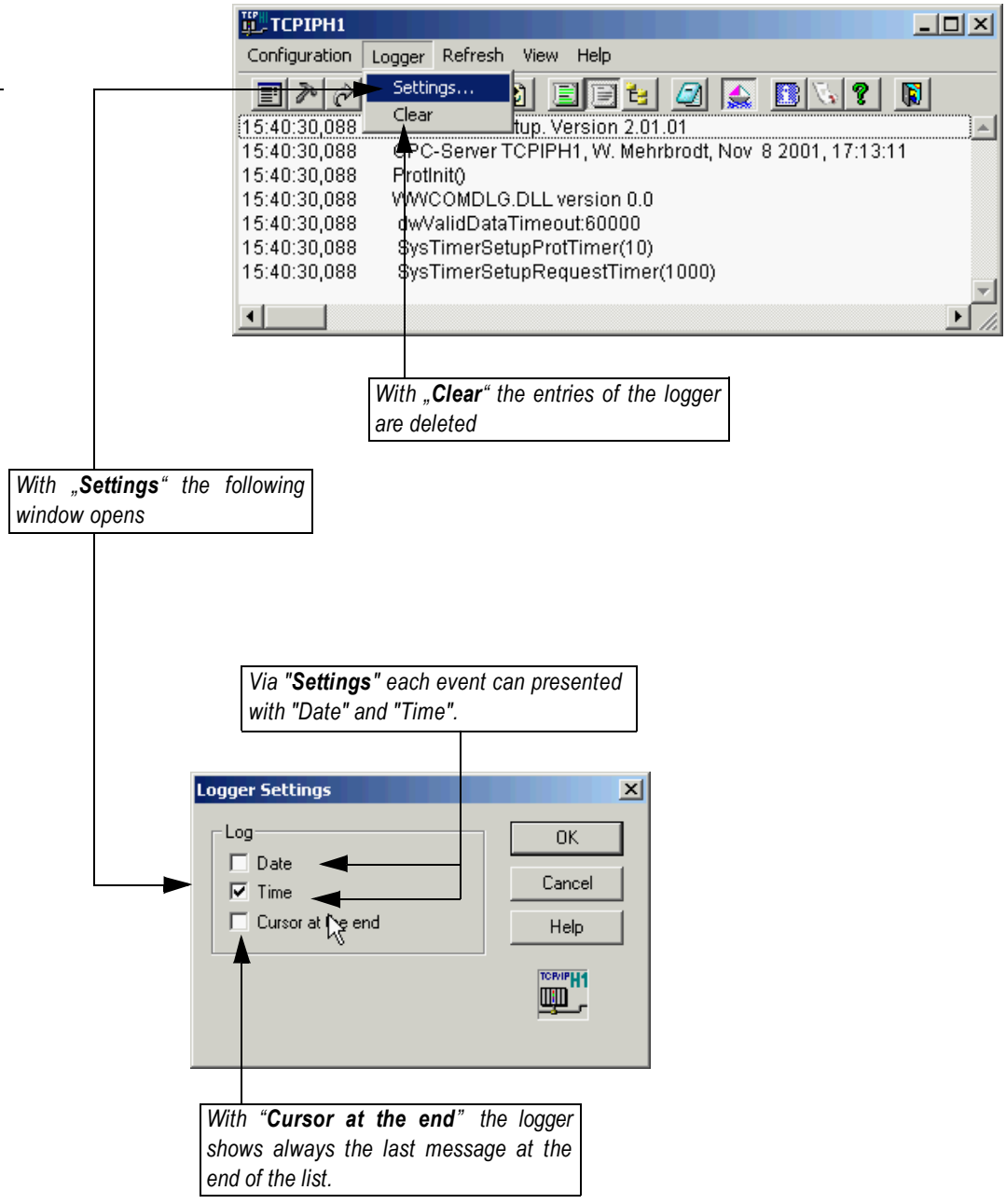
ShowSendValues: *Number of items, wihich are written to the server up to now.s all logger entries, respectively. from the server to the visualization, up to now.*

Verbose: *activates or deactivates all logger entries, respectively. from the server to the visualization, up to now.*

Show Symbols: *In the Show Symbols window the symbolic item names are shown, which you have entered in the symbols editor!*

4.10.1 The logger window of the OPC server

FIGURE 18.
Logger window of the OPC server



4.11 The Win.ini entries

The logger entries may also be stored in the file Win.ini under [TCPIPH1] . Beside the logger entries you will find the following entries: [TCPIPH1]

TABLE 3.
Entries in the Win.ini

WinIconic = 0	Standard windows function
WinFullScreen = 0	Standard windows function
WinTop = 0	Standard windows function
WinLeft = 0	Standard windows function
WinWidth = 65	Standard windows function
WinHeight = 200	Standard windows function
ShowEvents = 1	1 Logger entry active / 0 inactive
ShowSend = 1	1 Logger entry active / 0 inactive
ShowReceive = 1	1 Logger entry active / 0 inactive
ShowErrors = 0	1 Logger entry active / 0 inactive
Verbose = 0	1 Logger entry active / 0 inactive
DebugMenu = 1	1 Logger entry active / 0 inactive the Logging-, Suspend and Win.ini entries in the rollup menue of the main window
ShowCreatePoint = 1	1 Logger entry active / 0 inactive
ShowActivatePoint = 1	1 Logger entry active / 0 inactive
ShowAccessPath = 1	1 Logger entry active / 0 inactive
ShowRecValues = 1	1 Logger entry active / 0 inactive
ShowSendValues = 1	1 Logger entry active / 0 inactive
WriteRetryCount = 2	Number of write attempts, if plc is not ready
WriteCountBeforeNext-Read = 10	Number of write commands before the server performes the next read. This function avoids the blocking of an specific topic by permament write accesses.
AppTimeout = 3000	If the PLC doesn´t answer, the given time [msec] in „App-timeout“ is awaited three times before a connection is shut down and established again.
AppTimeout = 0	This function is inactive.

4.12 Indication modes of the OPC server

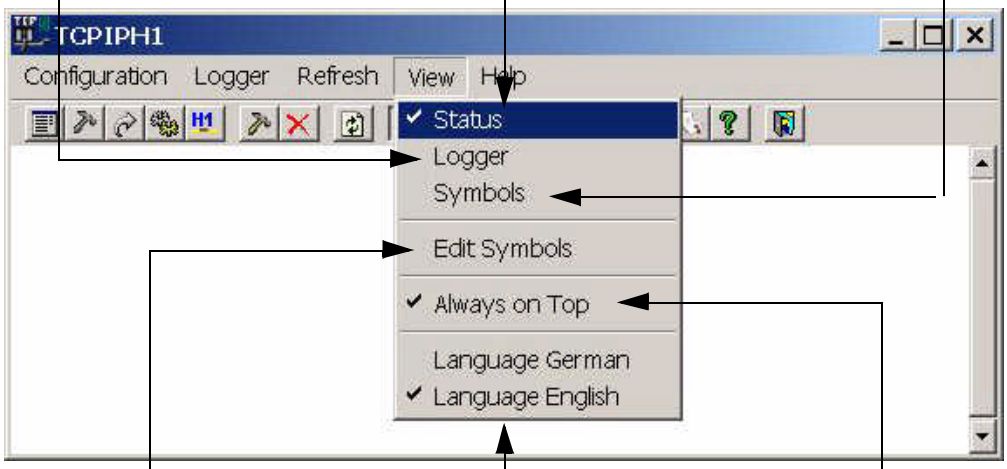
Since version 1.23.19 the OPC server supports the following indication modes:

- ***Status***
- ***Logger***
- ***Symbols***
- ***Edit Symbols***
- ***Always on Top***
- ***Show Server Icon in the taskbar (see„ Configuration / Server Settings“ on page 54)***

Logger
With the logger function of the server the entire data traffic of the current OPC communication can be monitored.

Status
The main window of the OPC servers show you the system project programmed in the process visualization software. During visualization, the status of the active access paths together with the activated read areas are indicated in the main window. These values are not cyclically updated automatically. You can obtain an update by clicking the menu item "Refresh" or by double-clicking the window itself.

Symbols
In the Show Symbols window the symbolic item names are shown, which you have entered in the symbols editor!



Edit symbols
*With the choice **View / Edit symbols** the server is prepared for the Alias Browsing with OPC clients! The standard windows editor is opened! (see chapter „ Edit Symbols“ on page 67).*

Always on top
If this display is activated, the OPC server will be shown on the top level of your windows surface

Language German/English
Here you can switch from the germnan to the english version of the software (and vice versa)

4.12.1 Edit Symbols

Under View / Edit Symbols you prepare the server for alias browsing with OPC clients! The standard windows editor is opened.

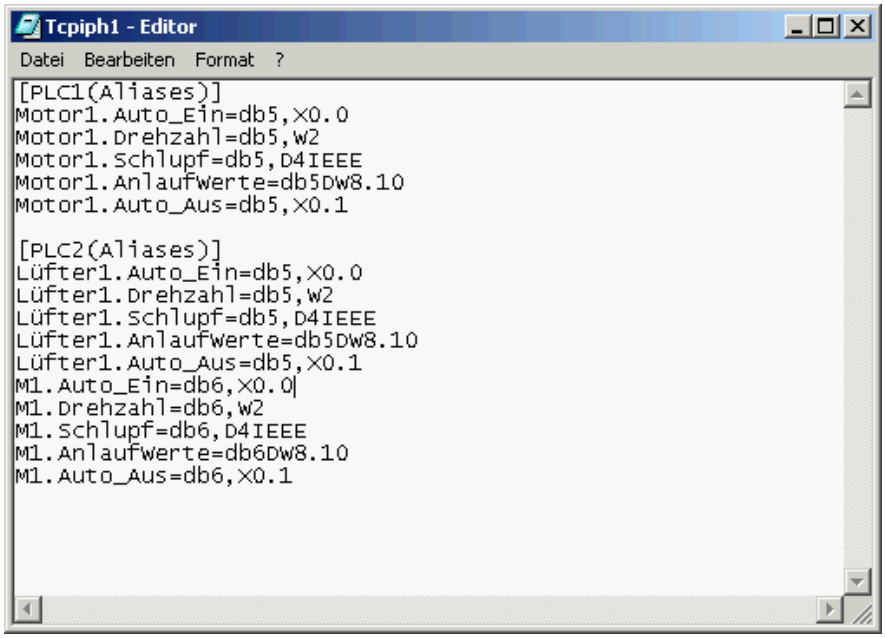


FIGURE 19. ***Standard Windows Editor***

First state the [Access Path (Aliases)]. Then assign every item in the PLC the desired Alias Name.

Example of a syntax

[PLC1(Aliases)]

Motor1.Auto_On=db5,X0.0

Motor1.rpml=db5,W2

Motor1.backlash=db5,D4IEEE

Motor1.initialvalue=db5DW8.10

Motor1.Auto_Off=db5,X0.1

[PLC2(Aliases)]

Fan1.Auto_On=db5,X0.0

Fan1.rpm=db5,W2

Fan1.backlash=db5,D4IEEE

Fan1.initialvalue=db5DW8.10

Fan1.Auto_Off=db5,X0.1

NOTE!

Alias Browsing is not supported by all OPC clients!

The entered Alias names are now available in the symbol indication of the **OPC servers** as well as in the browser of your OPC clients!

In the Show Symbols window the symbolic item names are shown, which you have entered in the symbols editor!!

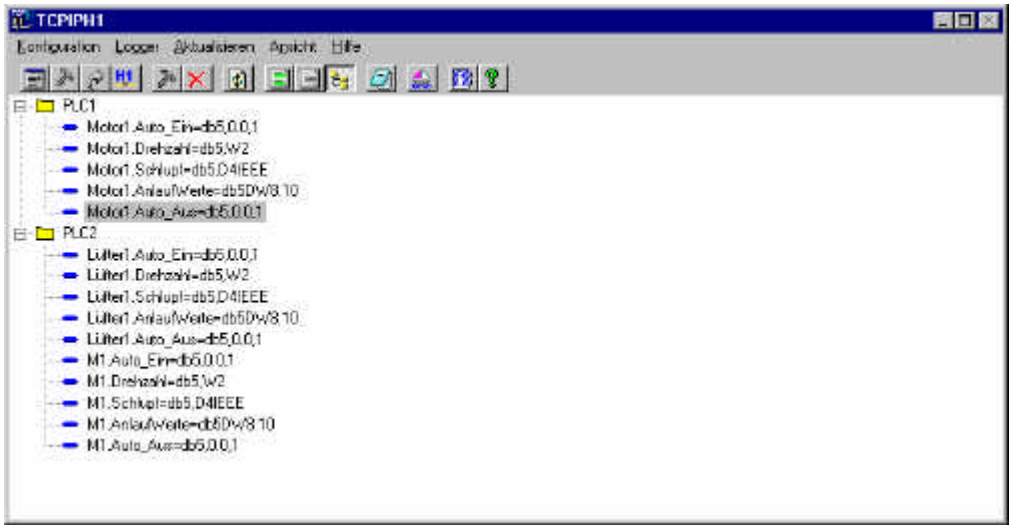
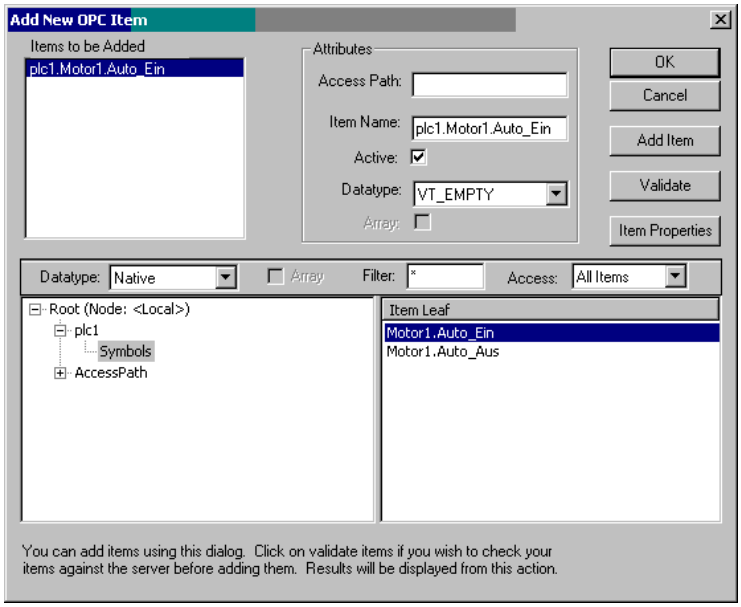


FIGURE 20.
Symbol view of the OPC server

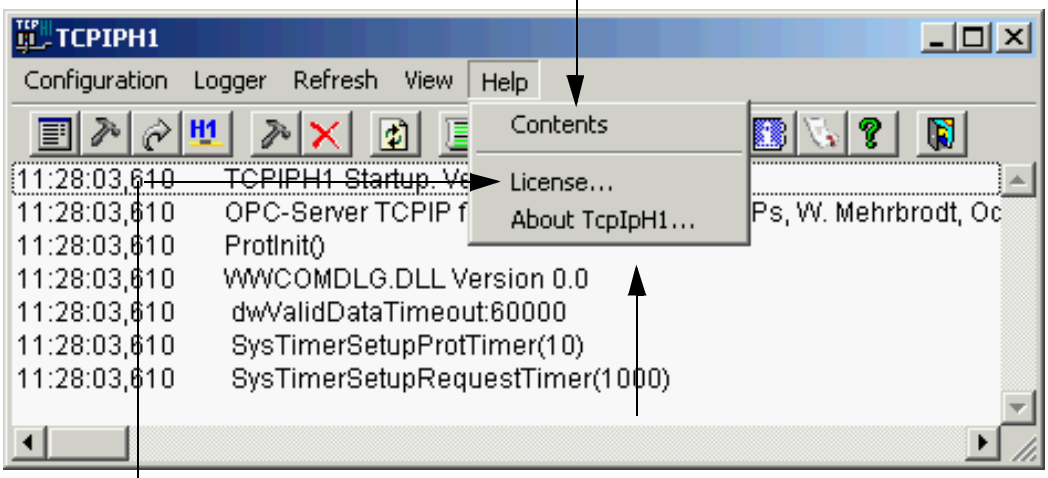


The symbolic names, edited in the server, can now be activated in the Item mask of the client in a very simple way. In the example above, the test client – included in the range of delivery – was used. This activation is possible with all other OPC clients, that support alias browsing.

4.13 Help

Under Help you have the following possibilities:

Contents
If you press „ Contents“ you reach the Content of the online help (see chapter „ Using Online Help“ on page 25)



License
Here you can license the software on hand (see (see chapter „ License“ on page 71)

About TcpIpH1
Here you receive information about the version number and the copyright of the software on hand.

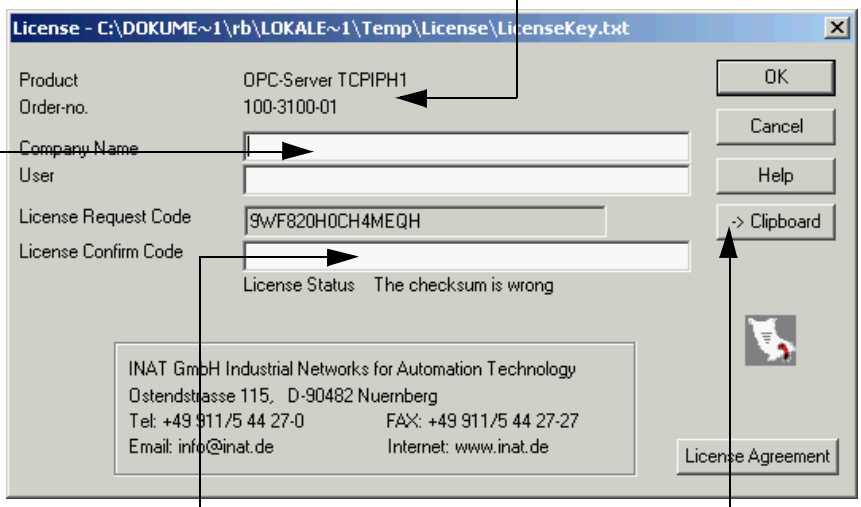
4.13.1 License

1

*Compare the entries under **Product** and **Order No.** with your ordering data!*

2

Enter your company name and the user name inorder to get a unique license.



Under License the following messages are possible:

License status	*ok*
License status	*The checksum is wrong: Check the entered codes*
License status	*no valid license: If the the Test runtime is over (by default 2h) the Server ends with the message "The server ends, because there exists no valid license"*

3

*Use the Button **->Clipboard.** The date for the licensing request are now available in the clipboard for pasting it into your favorite Email program:*
OPC-Server TCPIPH1
100-3100-01
INAT
Dipl. Ing. Thomas Muster
5IMMQ0H0CH2FQQH

4

*Send the **License Request Code***
- by Fax to +49 911 / 544 27-27 or
- by Email to info@inat.de.
The Request Code of your system is unique.
*Due to the contract with INAT you will get an unlimited **License Confirm Code** or a limited License Confirm Code for test purposes.*

NOTE!!

Before licensing the OPC server the H1 driver has to be installed

5 Communication with the OPC server

OLE is a complete communication protocol designed by Microsoft. It permits other Windows applications to exchange data and information with each other. OPC establishes a client-server relationship between two programs being executed simultaneously. The server application makes the data available and accepts requests from other applications which are interested in these data. The applications which make requests are called clients. Some applications (e.g., InTouch and Excel) can be clients and servers at the same time.

To receive data from another application, the client program opens a channel to the server application. Three names are specified.

The **application** name of the server (i.e., name of the *.exe file). If an appropriate OPC server exists, the client receives a positive response.

The **Access Path Name** is an abstract name which is set up when the OPC server is parameterized. Access Path combines the parameters that are to be used for communication (e.g., address of the PLC). The „Access Path“ is equivalent to the OPC standard

The **Item Name**: After communication has been established, the OPC server is informed of the data items required by the client. If the OPC server has fetched new data from the PLC, it sends a message containing a handle to all clients which are interested in these particular data items. The handle permits the client to read the data from memory. Data are transferred from the client to the OPC server in the same way.

Data transmission speed is further increased by combining various data into blocks. The generation of these blocks significantly reduces resource requirements. The two primary points apply.

- The interface between OPC server and client is always the same.
- The client only receives a message when the OPC server has received a new value from the PLC which differs from the previous value.

NOTE!

Der Begriff „Access Path" entspricht dem OPC-Standard. In vielen Clients wird jedoch noch der aus dem DDE-Format bekannte Begriff „Topic" verwendet

5.1 Name Convention „Application Name“

The visualization systems WinCC, Fix and others use OPC to communicate with I/O device drivers and other OPC application programs. The INAT OPC servers provide a communications channel for the SIMATIC via TCP/IP or SINEC H1. The application name for this application is TCPIPH1.

5.2 Name Convention „Access Path“

An example using the Windows application Excel is the application name "Excel." The topic name is the name of a certain work sheet which contains the data. The item name is a certain location on the work sheet. With the InTouch Window Maker, the topics are entered under "*Special-OPC Access Names ...- Add (or modify) - OPC Topic Name*." With the INAT OPC servers, the topic name is entered under "C*onfiguration - Topic definitions ...- New*." Data communication between the INAT OPC server and Excel is described in the appendix.

5.3 Name Convention „Item Name” (Point)

With InTouch, the application name is "view," and the topic name is "tag name" when read or write-accessing based on a tag name in InTouch. The item name is a certain tag name in the InTouch DataDictionary. When a client application has established a connection to another OPC program, the server application always informs it when the contents of a certain item have changed. This connection remains active until the client or the server program concludes the connection or data communication. This is a very efficient way of exchanging data since, once the connection has been established, communication only takes place when the values of certain data change.

The INAT OPC servers use an item name convention based on two-character data-type designators used in programming of Siemens PLCs. Using this supplement, the INAT OPC servers accept standard designators in German and English. The areas which are specified later, depend on the controller used.

5.3.1 Description of the special items

The servers offer additional items for each topic to acquire data about different aspects of the communication and to control the communication properties.

TABLE 4.
Data Types

Data Types	OPC
POLLINTERVAL Item	VT_14
MAXINTERVAL Item	VT_14
STORESETTING Item	VT_BOOL
ITEMCOUNT Item	VT_14
ERRORCOUNT Item	VT_14
WRITECOMPLETE Item	VT_14
Status	VT_BOOL
WriteItemCount	VT_14
WriteStatusMsg	VT_BSTR
ServerCycle	VT_14

POLLINTERVAL Item

The POLLINTERVAL item is used to access the currently set poll interval. It is the same value displayed in the Topic Definition / Access Path Definition window of the server. It indicates the current requested poll interval in milliseconds. A client can write new values to this item. In any case the value can always be read through OPC or SuiteLink. The range of valid values differs with respect to the topic settings.

NOTE!

By writing a negative value to the POLLINTERVAL item, a client can conveniently stop all update activities on the corresponding topic without having to deactivate the items.item count equals error count).

Value	Meaning
0 or greater	new poll interval
negative value 0	polling is disabled. The old positive value is still displayed.

By writing to the POLLINTERVAL item, a client can conveniently stop all update activities on the corresponding topic without having to deactivate the items.

Tag Type:	INTEGER (signed long) / VT_I4	
ACCESS:	READ, WRITE	
RANGE:	0.....2147483647	(valid values)
	-1....-2147483648	topic inactive, no items are updated

MAXINTERVAL Item

You use the MAXINTERVAL Item to access to the measured maximum update interval [in msec] of all items of the connection. This value references to the last total poll cyclus. The item is „read only".

NOTE!

Use the POLLINTERVAL and MAXINTERVAL items to conveniently tune the performance of the communication.

Tag Type:	INTEGER / VT_I4	
ACCESS:	READ	
RANGE:	0.....2147483647	

STORESETTINGS Item

The STORESETTINGS item is used to make the temporary poll interval changes to POLLINTERVAL via OPC or SuiteLink to the hard disk. If the client writes a value of 1 to the STORESETTINGS item, the current set update interval is written to the server's configuration file. If the update interval has been changed via OPC or SuiteLink and STORESETTINGS is not written to 1, the server will use the original update interval for that topic the next time it is started. Reading the item will always return 0. It does not matter, wether the client writes a 1 or a 0.

NOTE!

Use this item to conveniently store your optimized configuration via OPC or SuiteLink.

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

ITEMCOUNT Item

The ITEMCOUNT item is used to access the number of active items in the corresponding topic. It is the same value displayed in the main window of the server the entry "Items". This item is read only.

NOTE!

Use the ITEMCOUNT item to monitor the number of currently active items.

Tag Type:	INTEGER / VT_14	
ACCESS:	READ	
RANGE:	0.....2147483647	

ERRORCOUNT Item

The ERRORCOUNT item is used to access the number of active items with errors in the corresponding topic. If the communication status of a topic is bad, all items have errors (item count equals error count). This item is read only.

Tag Type:	INTEGER / VT_14	
ACCESS:	READ	
RANGE:	0.....2147483647	

NOTE!

Use the ERRORCOUNT item to check for badly configured items or invalid item values. This is the case if the topic status is 1 or is permanently changing from 1 to 0 and ERRORCOUNT is not zero.

WRITECOMPLETE Item

The WRITECOMPLETE item is used to access the state of pending write activities on the corresponding topic. If the topic is opened the value of the WRITECOMPLETE item is initially 1 indicating all write activities are complete - no writes are pending. If values are written into any items of the topic the value of the WRITECOMPLETE item changes to 0 indicating write activity is currently in progress. If the server has completed all write activities the value of the WRITECOMPLETE item changes to 1 if all writes were successful or to -1 if at least one write has failed. If the value of the WRITECOMPLETE item is not zero the client can write 1 or -1 to it (write a 1 to clear errors or a -1 to test a client reaction on write errors).

Tag Type:	INTEGER / VT_14	
ACCESS:	READ/WRITE	
RANGE:	-1,0,1	

Status

- The Status item is derived from the connection status of the fetch and write connection.
- When one of the connections has malfunctioned, the status indicates a malfunction. When no write connection is parameterized, only the status of the read connection is indicated.

WriteItemCount

- The item shows the number of items written.
- The number is incremented when the secure data transmission to the PLC has been concluded successfully.

- When an error occurs, the WriteStatusMsg item receives an error code. If an item is written several times in succession before the server is able to write the value, the last value will be written and not all intermediate values will be buffered. If this happens, the number is 1.
- The item can be set to 0 before writing and scanned after writing.

Write Status Msg

- The item shows the first error when write operation to the PLC fails.
- If this item does not contain a string or the first character is a blank, an error code is assigned. When an error occurs, the error code is retained until it is deleted by process visualization or the operator.
- The error code consists of 6 numbers separated by a semicolon (;) without blanks. Its meaning is shown below:

Identifier; DB; DW; Len; S5 error; connection error

Identifier

TABLE 5. Identifier

1	DATA BLOCK
2	FLAG
3	INPUT
4	OUTPUT
5	PERIPHERAL
6	COUNTER
7	TIMER
8	SYSTEM DATA
9	ABSOLUTE MEMORY
10	EXPANDED DATA BLOCKS
16	EXTMEM
17	EXT_PERIPHERAL

DB

Specifies the block number for BLOCK and EXP_BLOCK. Otherwise 0 (i.e., can be disregarded).

DW

Specifies the start value at which the interval beginsrwise 0 (i.e., can be disregarded).

Len

Specifies the number of elements. Calculation of the actual length in bytes uses the identifier and the following rules:
1 byte per unit for
FLAG, INPUT, OUTPUT, I/O and EXT_PERIPHERAL
2 bytes per unit for
DATA BLOCK, COUNTER, TIMER, SYSTEM_DATA, ABSOLUTE MEMORY, EXPANDED DATA BLOCKS and EXTMEM

S5 error

Indicates the error supplied by the handling block from the PLC.

Is only valid when the connection error is 0.

For a description of the error codes, see the programming manual of the controller unit for i

TABLE 6. S5 error

0	no error
1	Wrong Q/ZTYP in the handling blockfrom the PLC.
2	Area does not exist in the PLC (i.e., DB does not exist)
3	Area in the PLC is too small.
4	QVZ error occurred on the PLC
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 communication partner
10	Connection error (connection failed or could not be established)
11	Message error (Error in the firmware)
12	Trigger error (e.g. RECEIVE to SEND)
13	Abort after RESET
14	Job with READ/WRITE (PLC unable to Trigger)
15	Job does not exist
16	System error

Connection error

TABLE 7.
Connection error

0	no 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	UEBERLAST
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 FETCH/WRITE)

Server Cycle

This Item shows the number of a complete read/write cycles of the servers for a corresponding connection.

Tag Type: INTEGER / VT_I4

ACCESS: READ/WRITE

RANGE: -2147483648.to.2147483647

5.3.2 Description of Standard Items

The following help topics describe the "name convention" items for the SIMATIC S5 and S7. The areas specified in the applicable tables may vary depending on the controller used.

The following topics are described:

These items are supported by the S5 and the S7

- *Data Blocks*
- *Flag Bytes*
- *Input Bytes*
- *Output Bytes*
- *Counters*
- *Conversion suffices*
- *Arrays of the standard items*

These items are supported by the S7 only:

- *DATE*
- *DATE_TIME*
- *Peripheral Input Bytes*
- *Peripheral Output Bytes*

These items are supported by the S5 only

- *Timer*
- *System Dta Area*
- *Peripheral Bytes*
- *Absolute Memory*
- *Expanded Data Blocks*
- *Expanded Peripheral Bytes*

5.3.2.1 Data Blocks

Format	Item S5	Item S7*	Suffix	OPC Tag Typ	Range
Bit	DBxDy.z	DBx,Xy.z		BOOLEAN	0 or 1
String	DBxSy.v	DBx,STRINGy.v		STRING	0 to 9, A to F
			KA**	STRING	
Left Byte (S5)	DBxDLy	--		BYTE	0 to 255
		--	KF	CHAR	-128 to 127***
Right Byte (S5)	DBxDRy	--		BYTE	0 to 255
		--	KF	CHAR	-128 to 127***
Byte (S7)	DBxDBy	DBx,By		BYTE	0 to 255
Word	DBxDWn	DBx,Wn		LONG	0 to 65535
			KF	SHORT	-32768 to 32767
			BCD	SHORT	000 to 9999
			KC	LONG	0 to 65535
			KT	STRING	000.0 to 999.3
Double Word	DBxDDm	DBx,Dm		VT_R8	0 to 4294967295
			BCD	LONG	0 to 9999999
only S5			KG****	REAL	****
only S7			IEEE	REAL	4294967295
					4294967295

TABLE 8.
Data Blocks

Read and write-accesses to data blocks are permitted. The server processes write-accesses to the data blocks (x=1 to 65535, y = 0 to 65535, n = 0 to 65535, m = 0 to 65534, z = 0 to 15 (for the S7 z = 0 to 7), v = 2 to 128 with v = string length). The string requires an even number of characters for example 2, 4, 100, 128 etc. The server processes also one write-access (POKE) to one data block.

* The server supports the comma as well as the point expression!

** KA converts STRING to HEX. This is neccessary because termination characters of PC (0) and PLC (.) are different.Without the suffix KA all characters after a 0 character wouldn´t be defined in the PC.

*** If a character cannot be displayed it is substituted by the character #.

**** The KG range of the PLC is bigger than the range of the PC!"

5.3.2.2 Flag Bytes

Format	Item S5/S7	Suffix	OPC Tag Typ	Range
Bit	Fx.y		BOOLEAN	0 or 1
	Mx.y		STRING	0 to 9, A to F
		KA**	STRING	
Byte	FBx		BYTE	0 to 255
		KF	CHAR	-128 to 127***
	MBx		BYTE	0 to 255
		KF	CHAR	-128 to 127***
Word	FWn		LONG	0 to 65535
	MWn		LONG	0 to 65535
		KF	SHORT	-32768 to 32767
		BCD	SHORT	000 to 999
		KC	LONG	0 to 65535
		KT	STRING	000.0 to 999.3
Double Word	FDm		VT_R8	0 to 4294967295
	MDm		VT_R8	0 to 4294967295
		BCD	LONG	0 to 9999999
only S5		KG****	REAL	****
only S7		IEEE	REAL	4294967295
				4294967295

TABLE 9.
Flag Bytes

Flags can be read and written (x= 0 - 65,535, y= 0 - 7, n=0 - 65,534, m=0 - 65,532).
Example for **Arrays of Flag Bytes S5-Syntax**: FBx.z
Example for **Arrays of Flag Bytes S7-Syntax***: FBx,z
z = number of elements of the array. Arrays are not defined for strings and bits.

* The server supports the comma as well as the point expression!s is neccessary because termination characters of PC (0) and PLC (.) are different.Without the suffix KA all characters after a 0 character wouldn´t be defined in the PC.

*** If a character cannot be displayed it is substituted by the character #.

**** The KG range of the PLC is bigger than the range of the PC!"Arial

5.3.2.3 Input Bytes

Format	Item S5/S7*	Suffix	OPC Tag Typ	Range
Bit	Ix.y		BOOLEAN	0 or 1
	Ex.y		STRING	0 to 9, A to F
		KA**	STRING	
Byte	IBx		BYTE	0 to 255
		KF	CHAR	-128 to 127***
	EBx		BYTE	0 to 255
		KF	CHAR	-128 to 127***
Word	IWn		LONG	0 to 65535
	EWn		LONG	0 to 65535
		KF	SHORT	-32768 to 32767
		BCD	SHORT	000 to 999
		KC	LONG	0 to 65535
		KT	STRING	000.0 to 999.3
Double Word	IDm		VT_R8	0 to 4294967295
	EDm		VT_R8	0 to 4294967295
		BCD	LONG	0 to 9999999
only S5		KG****	REAL	****
only S7		IEEE	REAL	4294967295
				4294967295

TABLE 10.
Input Bytes

Inputs can only be read (x= 0 - 65,535, y= 0 - 7, n=0 - 65,534, m=0 - 65,532). The server ignores write-accesses to inputs.
Example for **Arrays of Input Bytes S5-Syntax**: EBx.z
Example for **Arrays of Input Bytes S7-Syntax***: EBx,z
z = number of elements of the array. Arrays are not defined for strings and bits.

* The server supports the comma as well as the point expression!

** KA converts STRING to HEX. This is neccessary because termination characters of PC (0) and PLC (.) are different.Without the suffix KA all characters after a 0 character wouldn´t be defined in the PC.

*** If a character cannot be displayed it is substituted by the character #.

5.3.2.4 Output Bytes

Format	Item S5/S7	Suffix	OPC Tag Typ	Range
Bit	Qx.y		BOOLEAN	0 or 1
	Ax.y		STRING	0 to 9, A to F
		KA**	STRING	
Byte	QBx		BYTE	0 to 255
		KF	CHAR	-128 to 127***
	ABx		BYTE	0 to 255
		KF	CHAR	-128 to 127***
Word	QWn		LONG	0 to 65535
	AWn		LONG	0 to 65535
		KF	SHORT	-32768 to 32767
		BCD	SHORT	000 to 999
		KC	LONG	0 to 65535
		KT	STRING	000.0 to 999.3
Double Word	QDm		VT_R8	0 to 4294967295
	ADm		VT_R8	0 to 4294967295
		BCD	LONG	0 to 9999999
only S5		KG	REAL	****
only S7		IEEE	REAL	4294967295
				4294967295

TABLE 11.
Output Bytes

Outputs can only be written (x= 0 - 65,535, y= 0 - 7, n=0 - 65,534, m=0 - 65,532). The server ignores read-accesses to outputs..

Example for **Arrays of Output Bytes S5-Syntax**: ABx.z
Example for **Arrays of Output Bytes S7-Syntax***: ABx,z
z = number of elements of the array. Arrays are not defined for strings and bits.

* The server supports the comma as well as the point expression!

** KA converts STRING to HEX. This is neccessary because termination characters of PC (0) and PLC (.) are different.Without the suffix KA all characters after a 0 character wouldn´t be defined in the PC.

*** If a character cannot be displayed it is substituted by the character #.

**** The KG range of the PLC is bigger than the range of the PC!

5.3.2.5 Counters

TABLE 12.
Counters

Format	Item S5/S7	Suffix	OPC Tag Typ	Range
Word	Cx		Integer	0 to 999
	Zx		Integer	0 to 999

Counters can only be read (x=0 - 65,535). The server ignores write-accesses to counters.

Example for **Arrays of Counters S5-Syntax**: Cx.z
Example for **Arrays of Counters S7-Syntax***: Cx,z
z = number of elements of the array. Arrays are not defined for strings and bits.

* The server supports the comma as well as the point expression! PC (0) and PLC (.) are different.Without the suffix KA all characters after a 0 character wouldn´t be defined in the PC.

*** If a character cannot be displayed it is substituted by the character #.

**** The KG range of the PLC is bigger than the range of the PC!"Arial

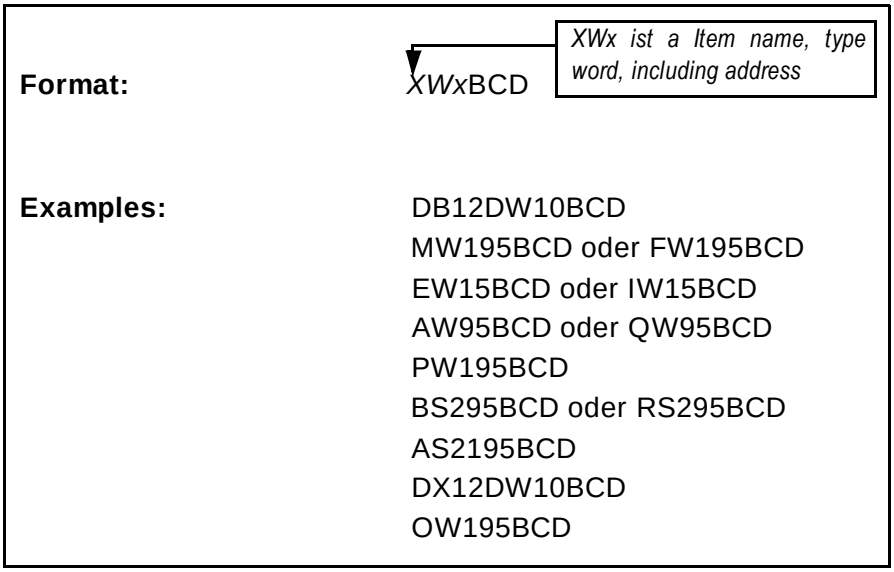
5.3.2.6 Conversion Suffices

The OPC Server processes "bit" items in OPC discrete format, and "byte," "word" and "double word" items in unsigned OPC integer format. (Exception: "Timer" items are in OPC real format). You can provide certain items with a suffix which calls special conversion routines for converting internal PLC formats to OPC formats. For the valid input ranges, see the Siemens manual of the PLC. For further informations click on one of the following suffices:

BCD	**KG**
D	**KT**
DT	**RI**
KA	**RU**
KC	**T**
KF	**TR**

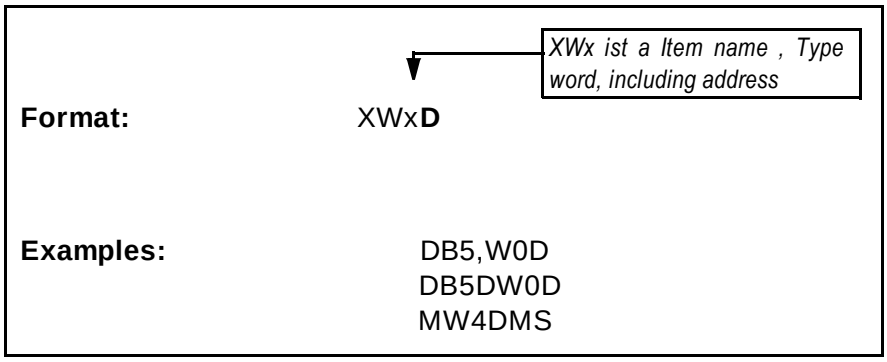
BCD

Treats the data stored on the PLC as unsigned 2 byte (4 decimal digit) BCD, and converts it into OPC integer format. For example, would return 0x1234 on the PLC to the OPC client as 1234 (decimal). .



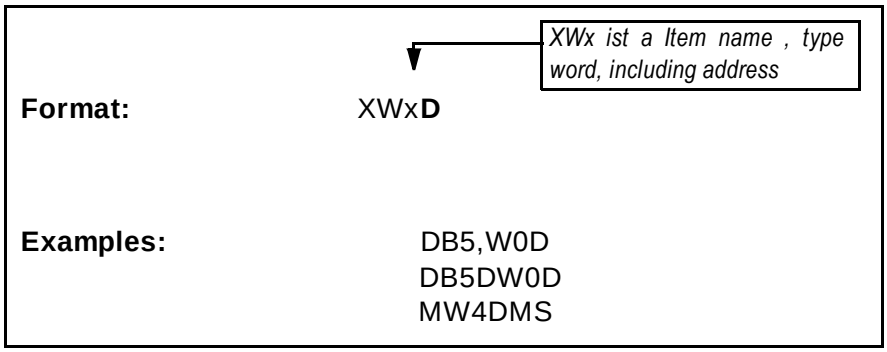
D

Treats the data stored on the PLC as hex and converts it to OPC integer format.format. For example 90-1-1 on the PLC would return to the OPC client as 1990-01-01 (text string). This conversion is used for a word-type item stored in a data block.



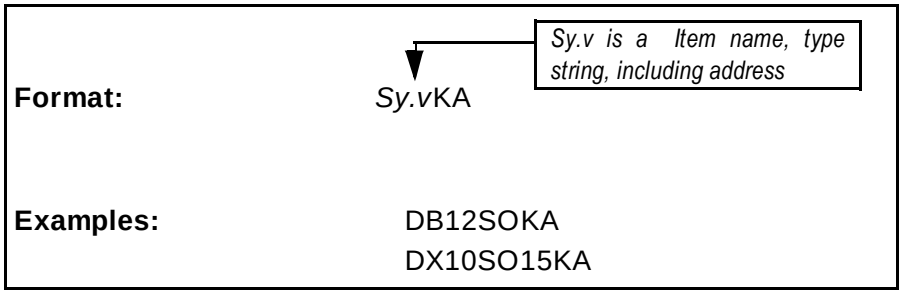
DT

Treats the data stored on the PLC as hex and converts it to OPC integer format.format. For example, would return 90-1-1 on the PLC to the OPC client as 1990-01-01-00:00:00.000 (text string). This conversion is used for a word-type item stored in a data block.



KA

Treats the data stored on the PLC as hex and converts it to OPC integer format.format. For example, would return 0x809B on the PLC to the OPC client as -32613 (decimal).. Do not confuse the counter constants with the counter item for which such a conversion would not be permitted. Counter items are not counter constants. This conversion is used for a word-type item stored in a data block.



KC

Treats the data stored on the PLC as a 2 byte (3 decimal digits) counter, and converts it to OPC integer format. For example, would return 0x0123 on the PLC to the client as 123 (decimal). Do not confuse the counter constants with the counter item for which such a conversion would not be permitted. Counter items are not counter constants. This conversion is used for a word-type item stored in a data block.ored in a data blockual of the PLC.

Format: *XWx*KC — *XWx ist a Item name, type word, including address*

Examples:
DB12DW10KC
BS295KC oder RS295KC
AS2195KC
DX12DW10KC

KF

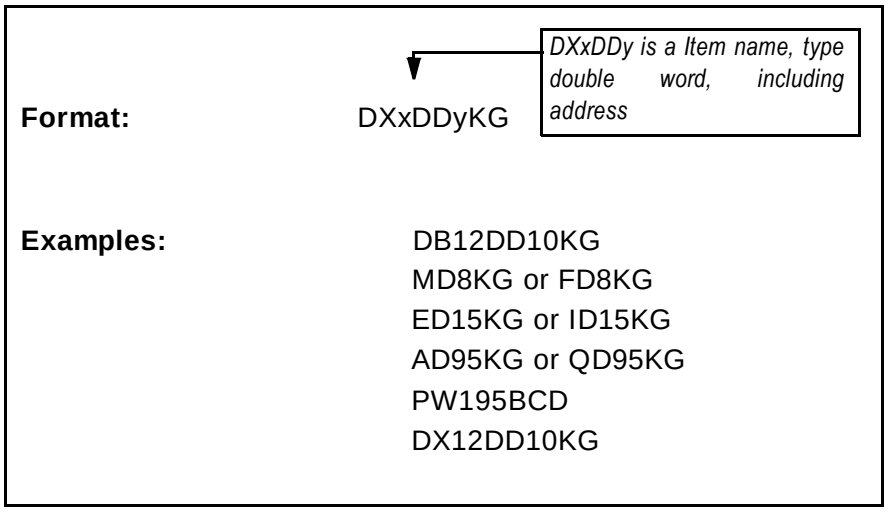
Treats the data stored on the PLC as a signed 2 byte integer, and converts it into OPC integer format. For example, would return 0x809B on the PLC to the OPC client as -32613 (decimal).

Format: *Syv*KA — *Syv is a Item name, type word, including address*

Examples:
DB12DW10KF
MW195KF or FW195KF
EW15KF or IW15KF
AW95KF or QW95KF
PW195KF
BS295KF or RS295KF
AS2195KF
DX12DW10KF
OW195KF

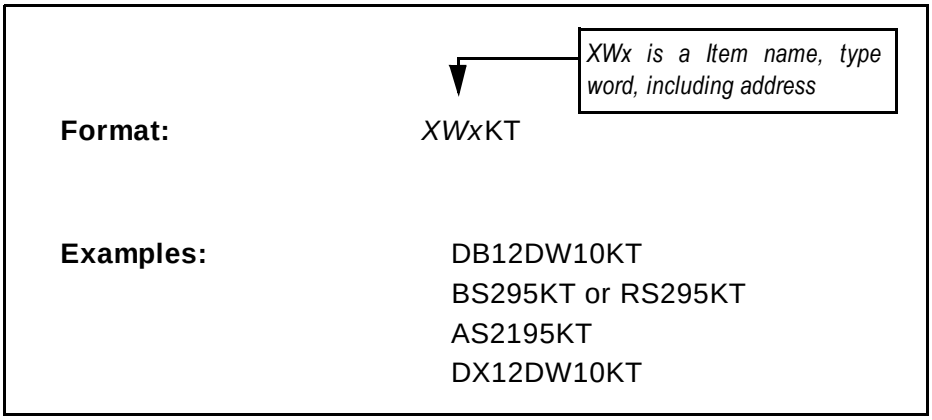
KG

Treats the data stored on the PLC as 4 byte floating point, and converts it to OPC real format



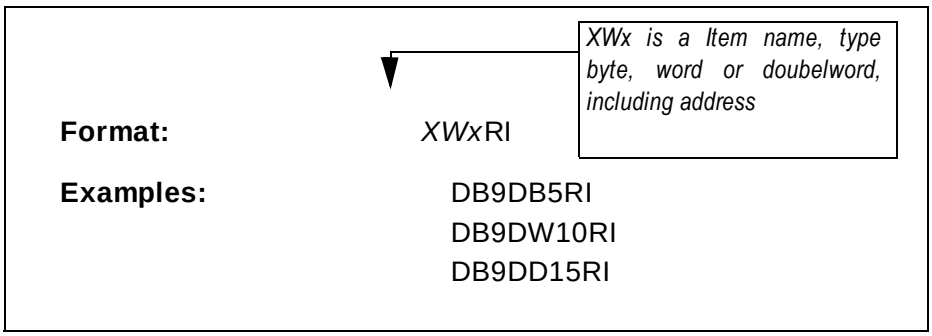
KT

Treats the data stored on the PLC as a 2 byte TIMER constant (3 BCD digits plus area digit), and converts it to OPC message format. For example, would return 0x2456 on the PLC to the OPC client as "456.2" (text string). Do not confuse the timer constants with the timer item for which such a conversion would not be permitted. Timer items are not timer constants. This conversion is used for a word-type item stored in a data block



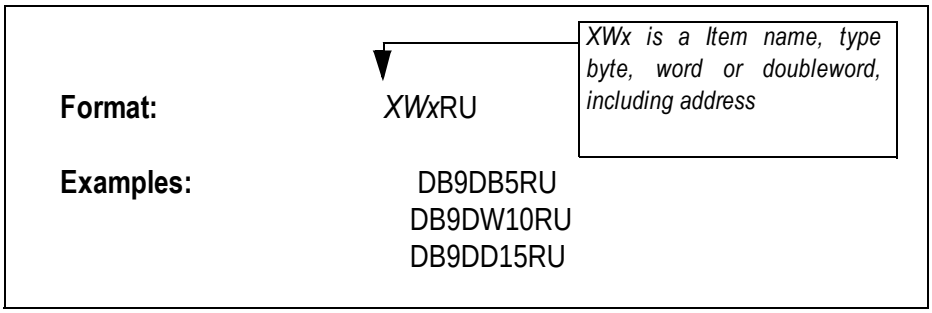
RI

Treats the data stored on the PLC as signed real format and converts it to the OPC message integer format. For example, would return 10.7 on the PLC to the OPC client as rounded 11 (integer). This conversion is used for a byte-type item, word-type item or double-word-type item stored in a data block..



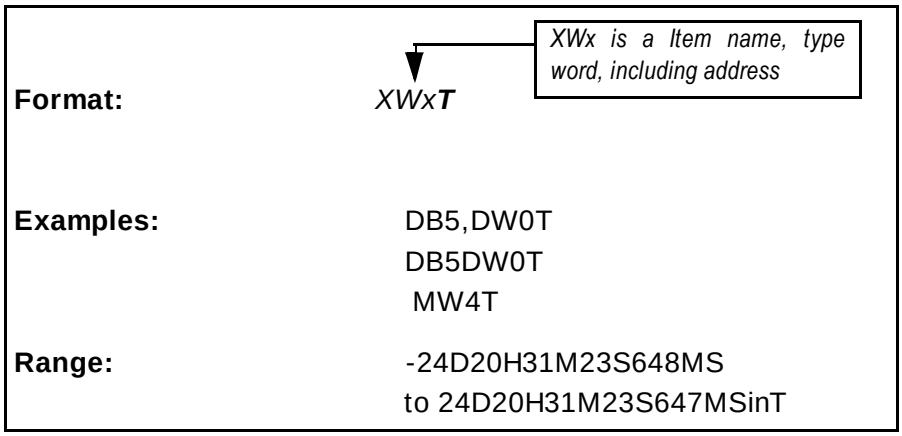
RU

Treats the data stored on the PLC as unsigned real format and converts it to the OPC message integer format. For example, would return 10.7 on the PLC to the OPC client as rounded 11 (integer). This conversion is used for a byte-type item, word-type item or double-word-type item stored in a data block.



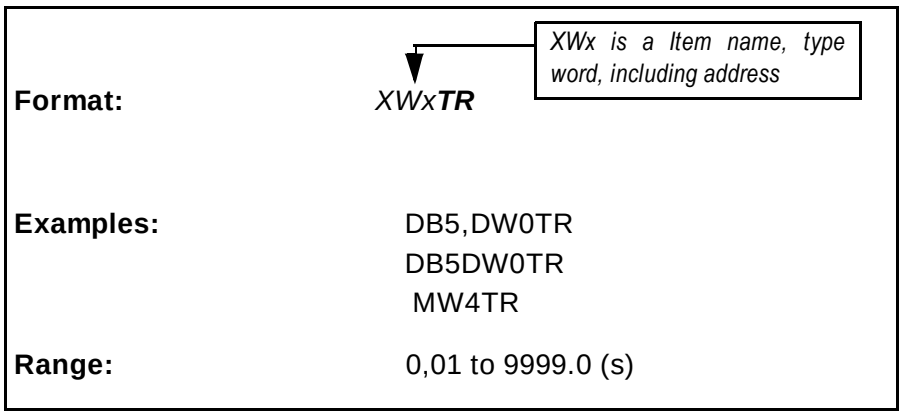
T

Treats the data stored on the PLC as hex and converts it to OPC String format.

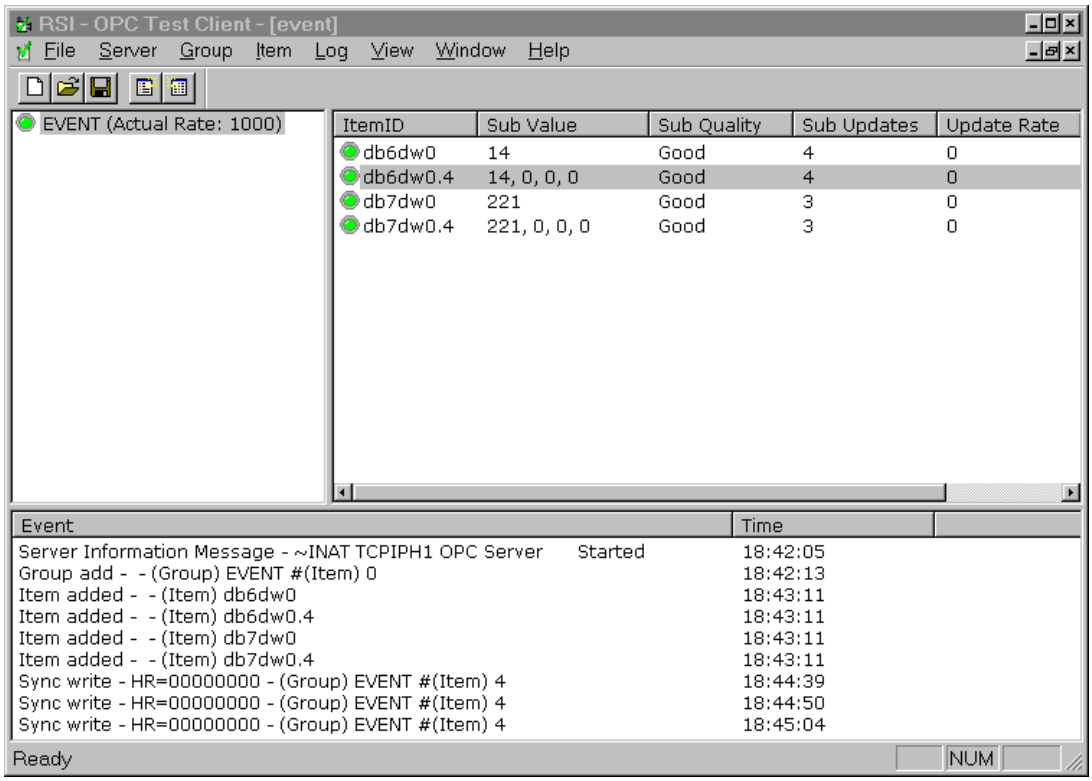


TR

Treats the data stored on the PLC as hex and converts it to OPC REAL format. This conversion is used for a word-type item stored in a data block.



5.3.2.7 Quality Bit



Quality Bit

The Quality Bit indicates the status of the OPC communication in the OPC client.

Quality Indication	***Example***
Not_Connected	*Bus error (Wait_Connect)*
Device_Failure	*e.g. Application_Timeout*
Sensor_Failure	*e.g. data block too short or not available*
Configuration_Error	*Error in the server configuration; e.g. "echo written data" is active, Write to the PLC is initiated but"Write Access to the PLC is not permitted".*

5.3.2.8 Arrays of the standard items

The arrays of the standard items are depicted below. Arrays are valuable also for the other items input bytes, output bytes, counters. Arrays are not supported for the formates *Bit* and *String*.

TABLE 13.
Arrays of the standard items

Format	Item S5	Item S7*	Suffix	Tag Typ	Range
Left Byte	DBxDy.z	--		Integer Array	1 to 255
			KF	Integer Array	-128 to 127
Right Byte	DBxDRy.z	--		Integer Array	1 to 255
			KF	Integer Array	-128 to 127
Byte (S7)	DBxDBy.z	DBx,By.z		Integer Array	0 to 255
Word	DBxDWn.z	DBx,Wn.z		Integer Array	0 to 65535
			KF	Integer Array	-32768 to 32767
			BCD	Integer Array	-999 to 999
			KC	Integer Array	0 to 999
Double Word	DBx / DDm.z			Integer Array	0 to 4294967294
			KF	Integer Array	-21474833648 to 2147483647
			BCD	Integer Array	-9999999 to 9999999
(S7)			KG		Floating point
			IEEE	Real Array	
			KF	Integer Array	

Read and write-accesses to arrays are permitted. (x=1 to 65535, y = 0 to 65535, n = 0 to 65535, m = 0 to 65534, z = number of elements of the array). Arrays are not defined for strings and bits. The server processes also one write-access (POKE) to one data block. For the mode compatible up to version 1.22 arrays are not supported.

5.3.2.9 DATE

Format	Item S5	Item S7*	Suffix	OPC Tag Typ	Range
Word	DBxDWn	DBx,Wn		LONG	1 to 255
			D	STRING	-128 to 127

DATE can be read wnd written (x= 0 - 65,535, n=0 - 65,534).

* The server supports the comma as well as the point expression

DATE format permissive examniation:

- year may be enterd with 2 or 4 digits
  - ***90 - 99 ==> 1990 - 1999***
  - ***00 - 89 ==> 2000 - 2089***
  - ***größer als 2168 ==> 2168***
- Month and day may entered with 1 or 2 digits
- If month is
  - ***< 1 ==> 1***
  - ***> 12 ==> 12***
- If day is
  - ***< 1 ==> 1***
  - ***> than days of the month ==> the value is corrected to the corresponding days of the month***
- Except numbers for hyphen all signs are alowed
- Number of hyphens is arbitrary

5.3.2.10 DATE_TIME

Format	Item S5	Item S7*	Suffix	OPC Tag Typ	Area
Word	DBxDWn	DBx,Wn		LONG	0 to 65535
			DT	STRING	1990-01-01-00:00:00.000 to 2098-12-31-24:59:59:99
					1990-01-01-00:00:00 to 2098-12-31-24:59:59

DATE_TIME can be read and written (x= 0 - 65,535, n=0 - 65,534).

* The server supports the comma as well as the point expression

DATE_TIME accurate examination:

- Hyphen:
  - ***Year-Month-Day-Hour:Min:Sec.ms***
  - ***Year-Month-Day-Hour:Min:Sec other entries are not allowed***
- ms inclusive dot may be obmitted
  - ***Year 4 digits, 1990 - 2089***
  - ***Month 2 digits, 01 - 12***
  - ***Day 2 digits, 01 - 31***
  - ***Hour 2 digits, 00 - 23***
  - ***Minute 2 digits, 00 - 59***
  - ***Second 2 digits, 00 - 59***
  - ***ms 3 digits, 000 - 999***

Minor input errors are not allowed, e.g. 30.02 or 31.04

STEP7 program example DATE_TIME

```
DATA_BLOCK DB 5
VERSION: 0.1


STRUCT
DatumUhrAktuell : DATE_AND_TIME ;
x : WORD ;
DatumUhrVisu : DATE_AND_TIME ;
x1 : WORD ;
DatumUhrZuletzt : DATE_AND_TIME ;
x2 : WORD ;
END_STRUCT ;
BEGIN
DatumUhrAktuell := DT#90-1-1-0:0:0.000;
x := W#16#0;
DatumUhrVisu := DT#90-1-1-0:0:0.000;
x1 := W#16#0;
DatumUhrZuletzt := DT#90-1-1-0:0:0.000;
x2 := W#16#0;
END_DATA_BLOCK
```

FUNCTION FC 3 : VOID

```
TITLE =
VERSION : 0.1

VAR_TEMP
DatumUhrzeit_0 : DATE_AND_TIME ;
DatumUhrzeit_1 : DATE_AND_TIME ;
END_VAR
BEGIN
NETWORK
TITLE =


// Read SystemClock to DB5,B0-7
CALL SFC    1 (
RET_VAL              := MW   200,
CDT                  := #DatumUhrzeit_0);

L     P##DatumUhrzeit_0;
LAR1  ;
```

```
L     LD [AR1,P#0.0]; // Year, month, day
T     DB5.DBD    0;
L     LD [AR1,P#4.0]; // hour, minute, second, millisecond, day of the week
T     DB5.DBD    4;
```

// Write Visu Time from DB5,B10-17 to the System clock
// In DB5,B20-27 the last value is hold.
// If a new value is written from the visu system to the PLC, due to the
// new /old comparison the "SET_CLK" for setting the clock time is called

```
    L     DB5.DBD   10;
    L     DB5.DBD   20;
    <>D   ;
    SPB   now;
    L     DB5.DBD   14;
    L     DB5.DBD   24;
    <>D   ;
    SPB   now;
    BEA   ; // do nothing, exit
now:  L     DB5.DBD   10;
    T     DB5.DBD   20;
    L     DB5.DBD   14;
    T     DB5.DBD   24;

    L     P##DatumUhrzeit_1;
    LAR1  ;

    L     DB5.DBD   10;
    T     LD [AR1,P#0.0]; // year, month, day
    L     DB5.DBD   14;
    T     LD [AR1,P#4.0]; // hour, minute, second, millisecond

    CALL SFC    0 (
       PDT                := #DatumUhrzeit_1,
       RET_VAL              := MW   202);

END_FUNCTION
```

5.3.2.11 Peripheral Input Bytes

Format	Item S5/S7	Suffix	OPC Tag Typ	Range
Byte	PEBx		BYTE	0 to 255
		KF	CHAR	-128 to 127***
Word	PEWn		LONG	0 to 65535
		KF	SHORT	-32768 to 32767
		BCD	SHORT	000 to 9999
		KC	LONG	0 to 65535
		KT	STRING	000.0 to 999.3

TABLE 14.
Peripheral Input Bytes

Peripheral (I/O) bytes can only be read (x= 0 - 65,535, n=0 - 65,534). The server ignores write-accesses to peripheral bytes.

Example for **Arrays of Peripheral Input Bytes S7-Syntax***: PEBx,z

z = number of elements of the array. Arrays are not defined for strings and bits.

* The server supports the comma as well as the point expression!

5.3.2.12 Peripheral Output Bytes

Format	Item S5/S7	Suffix	OPC Tag Typ	Range
Byte	PABx		BYTE	0 to 255
		KF	CHAR	-128 to 127***
Word	PAWn		LONG	0 to 65535
		KF	SHORT	-32768 to 32767
		BCD	SHORT	000 to 9999
		KC	LONG	0 to 65535
		KT	STRING	000.0 to 999.3

TABLE 15.
Peripheral Output Bytes

Peripheral (I/O) bytes can only be read (x= 0 - 65,535, n=0 - 65,534). The server ignores write-accesses to peripheral bytes.

Example for **Arrays of Peripheral Output Bytes S5-Syntax**: PABx.z
Example for **Arrays of Peripheral Output Bytes S7-Syntax***: PABx,z
z = number of elements of the array. Arrays are not defined for strings and bits.

* The server supports the comma as well as the point expression!

5.3.2.13 Timer (only S5)

Format	Item S5 / S7	Suffix	Tag Typ	Range
Word	Tx		Real	0.00 to 999.3

TABLE 16. Timer

Timers can only be read (x=0 to 65,535).The server ignores write-accesses to timers. .

Example for **Arrays of Timers S5-Syntax**: Tx.z
Example for **Arrays of Timers S7-Syntax***: Tx,z
z = number of elements of the array. Arrays are not defined for strings and bits.

* The server supports the comma as well as the point expression! PC (0) and PLC (.) are different.Without the suffix KA all characters after a 0 character wouldn´t be defined in the PC.
*** If a character cannot be displayed it is substituted by the character #.
**** The KG range of the PLC is bigger than the range of the PC!"

5.3.2.14 System Data Area

Format	Item/Punkt	Suffix	Tag Typ	Range
Word	RSx		Integer	0 to 65535
	BSx		Integer	0 to 65535
		KF	Integer	-32768 to 32767
		BCD	Integer	-999 to 999
		KC	Integer	0 to 999
		KT	Message	0.0 to 999.3

TABLE 17. System Data Area

Words in the system data area can only be read (x= 0 - 65,535). The server ignores write-accesses words in the system data area.

5.3.2.15 Peripheral Bytes (only S5)

Format	Item S5 / S7	Suffix	OPC Tag Typ	range
Byte	PBx		BYTE	0 to 255
		KF	CHAR	-128 to 127***
Word	PWn		LONG	0 to 65535
		KF	SHORT	-32768 to 32767
		BCD	SHORT	000 to 999
		KC	LONG	0 to 65535
		KT	STRING	000.0 to 999.3

TABLE 18. Peripheral Bytes

Peripheral (I/O) bytes can only be read (x= 0 - 65,535, n=0 - 65,534). The server ignores write-accesses to peripheral bytes.

Example for **Arrays of Peripheral Input Bytes S5-Syntax**: PEBx,z
z = number of elements of the array. Arrays are not defined for strings and bits.

5.3.2.16 Absolute Memory

Format	Item/Punkt	Suffix	OPC Tag Typ	Range
Word	ASx		Integer	0 to 32767

TABLE 19. Absolute Memory

Words in absolute memory can only be read (x=0 to 65,535). The server ignores write-accesses to words in absolute memory.

5.3.2.17 Expanded Data Blocks

Format	Item/Punkt	Suffix	OPC Tag Typ	Range
Bit	DXx / Dy.z		Discrete	0 or 1
String	DXx / Sy.v	KA	Message	2 to 128
Left Byte	DXx / DLy		Integer	1 to 255
		KF	Integer	-128 to 127
Right Byte	DXx / DRy		Integer	1 to 255
		KF	Integer	-128 to 127
Word	DXx / DWn		Integer	0 to 65535
		KF	Integer	-32768 to 32767
		BCD	Integer	-999 to 999
		KC	Integer	0 to 999
		KT	Message	0.0 to 999.3
Double Word	DXx / DDm		Integer	-2147483648 to 2147483647
		BCD	Integer	-9999999 to 9999999
		KG	Real	floating point
(S7)		IEEE	Real	Floating point (IEEE format)

TABLE 20. Expanded Data Blocks

Read and write-accesses to expanded data blocks are permitted (x=1 - 65,535, y= 0 - 65,535, n=0 - 65,535, m=0 - 65,534, z=0 - 15, v= 2 - 128 v=Length of string). The string should consist of an even number of characters.fs20 : PEBx,z
z = number of elements of the array. Arrays are not defined for strings and bits.: PABx,z

5.3.2.18 Expanded Peripheral Bytes

TABLE 21. Expanded Peripheral Bytes

Format	Item S5 / S7	Suffix	OPC Tag Typ	Range
Byte	OBx		Integer	0 to 255
		KF	Integer	-128 to 127
Word	OWn		Integer	0 to 65535
		KF	Integer	-32768 to 32767
		BCD	Integer	-999 to 999
		KC	Integer	0 to 999
		KT	Message	0.0 to 999.3

Expanded peripheral bytes can only be read (x= 0 - 65,535, n=0 - 65,534). The server ignores write-accesses to expanded peripheral bytes.=0 - 65,535, m=0 - 65,534, z=0 - 15, v= 2 - 128 v=Length of string). The string should consist of an even number of characters.fs20 : PEBx,z

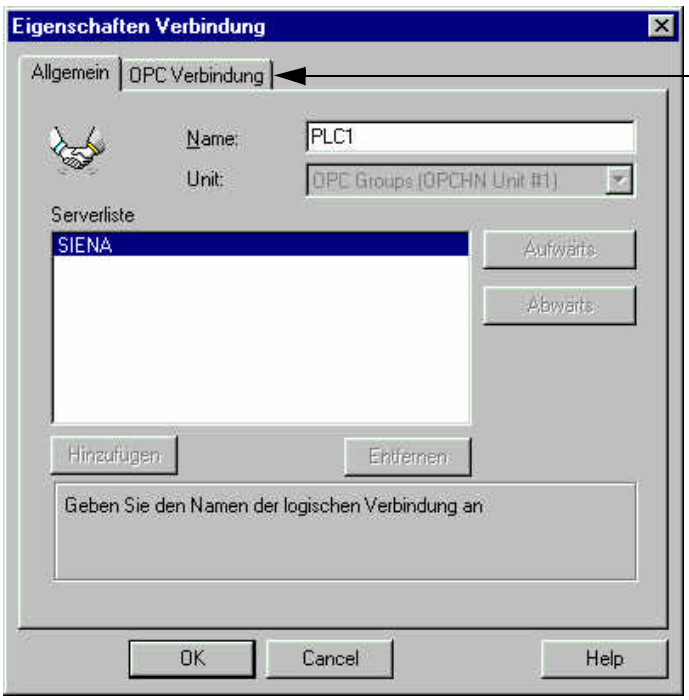
z = number of elements of the array. Arrays are not defined for strings and bits.

5.4 OPC communication in the visualization software

5.4.1 OPC communication with WinCC

The communication with the OPC server is handled in the visualization software WinCC as follows.

- ***Please select the OPC driver „OPC.chn“ from the menu „add variables / driver“.***
- ***Create a new connection and give it a name. This name corresponds to the topic or connection, which you have created in the OPC server.***



The application, in this case OPC server, you activate in the same dialoge box in the menu „OPC Group Settings“

FIGURE 21.
Create the Access Path Name in WinCC

NOTE!

Pay attention to the exact spelling:
TCPIPH1 OPC server

.

FIGURE 22.
Create the Application Name in WINCC

Check with button „Test Server“ if the OPC server is ready to start.

You´ll get the following message

FIGURE 23.
OPC server is „OK“

If not, please check if the OPC server is already started. If the OPC server runs on another PC in the network pay attention you have entered the correct path.

In the following dialogues you enter the „TAG properties“ (ITEMs). Each tag is linked to the corresponding connection (In this case „PLC1“).

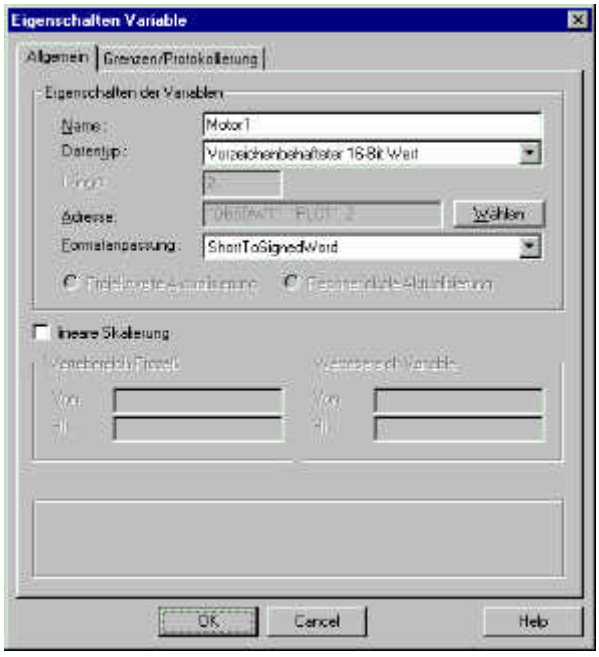


FIGURE 24.
Tag properties (ITEM)

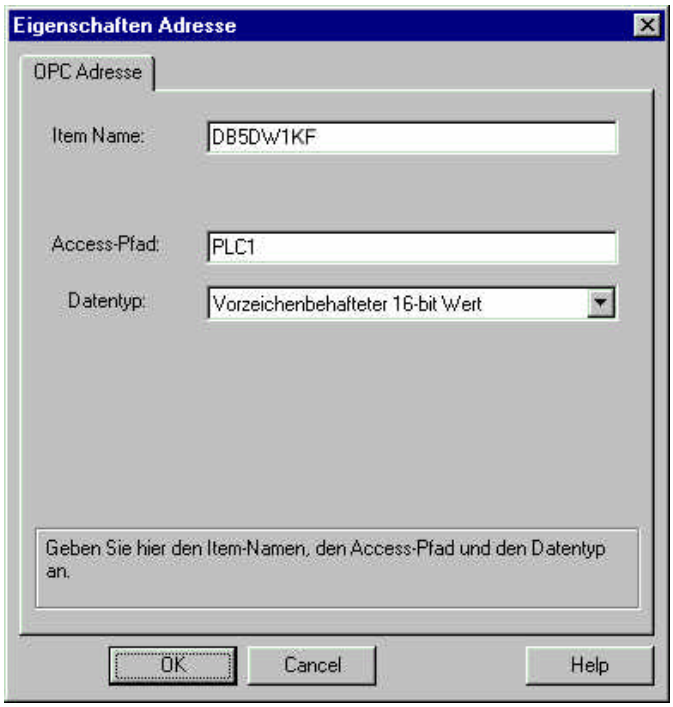


FIGURE 25.
Properties of the tag address

With DB5DW1KF you express, that you read data from the data block DB5 via OPC server with the corresponding connection PLC1. The data word DW1 with format KF is interpreted as 16 bit value (signed). The OPC server supports the following data types at the moment:

Tag Name	Type compatible	also unsigned types	standard types
DB5D1.0	VT_BOOL	VT_BOOL	VT_BOOL
DB5DB1	VT_I4	VT_UI1	VT_UI1
DB5DB1KF	VT_I4	VT_BSTR	VT_BSTR
DB5DL1	VT_I4	VT_UI1	VT_UI1
DB5DR1	VT_I4	VT_UI1	VT_UI1
DB5DW1	VT_I4	VT_UI2	VT_I4
DB5DW1KF	VT_I4	VT_I2	VT_I2
DB5DW1KT	VT_BSTR	VT_BSTR	VT_BSTR
DB5DW1KC	VT_I4	VT_UI2	VT_I4
DB5DW1BCD	VT_I4	VT_I2	VT_I2
DB5DD1	VT_I4	VT_UI4	VT_R8
DB5DD1KF	VT_I4	VT_I4	VT_I4
DB5DD1KG	VT_R4	VT_R4	VT_R4
DB5DD1IEEE	VT_R4	VT_R4	VT_R4
DB5S1.10	VT_BSTR	VT_BSTR	VT_BSTR
DB5DW1KF	VT_I4	VT_I2	VT_I2
DB5DW2KF	VT_I4	VT_I2	VT_I2
DB5DW0.10	----------	VT_ARRAY	VT_ARRAY

Please create the corresponding animation for this tag in the graphics designer of WinCC. Select „New animation“ from the object palette.

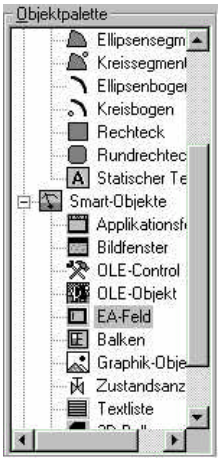


FIGURE 26.
Creating a new animation for an I/O field

The I/O field is configured in the following dialog box. Use the same tag name as you entered in the „Tag properties“.

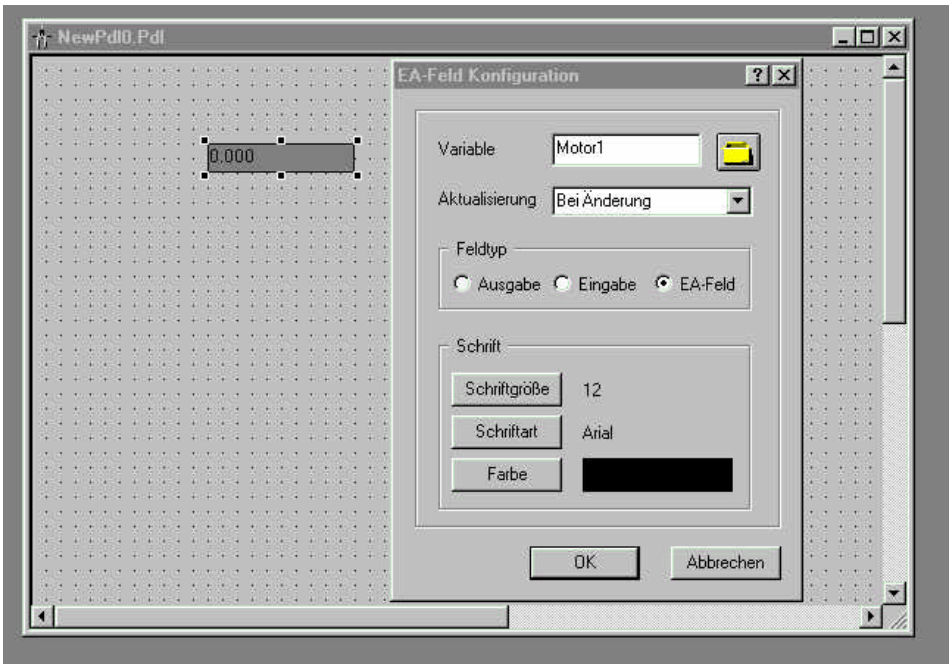


FIGURE 27.
I/O field configuration

With a click on the right mouse button you can select the characteristics of the I/O field. Important: at this point you define the output format.

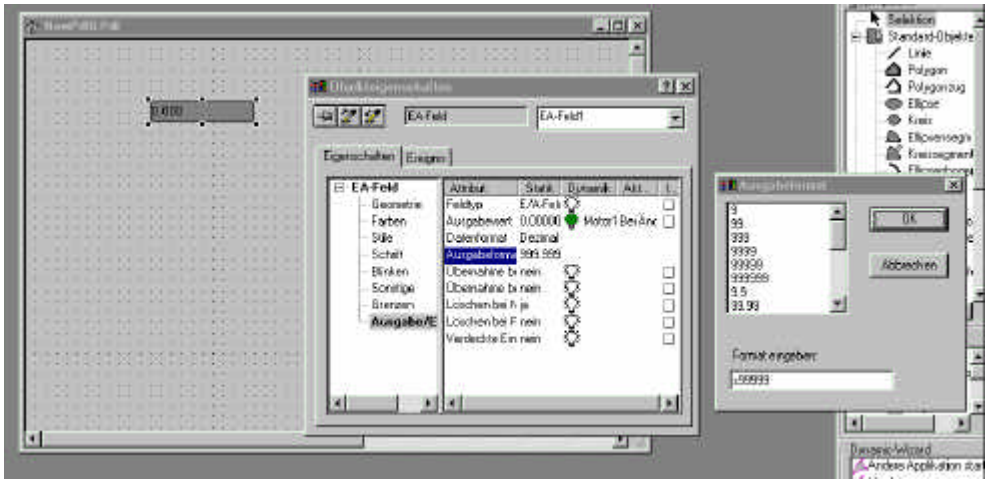


FIGURE 28.
Entering the output format

Once the format is entered start the OPC communication in the main window. Now you will see the data of the PLC that change cyclically. See chapter „Introductory Example“ on page 115 for establishing such an OPC communication on the PLC side.

5.4.2 OPC communication with FIX

As for WinCC you have to activate the OPC driver of the visualization software FIX, too.

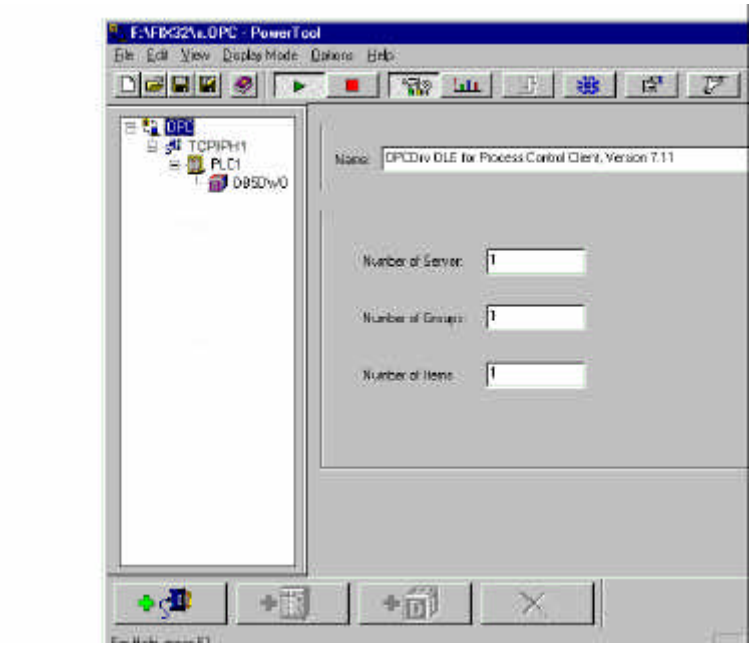


FIGURE 29.
Initializing the OPC server communication

Please enter „INAT“ and „INAT TCPIPH1 OPC Server“ for the applications name.

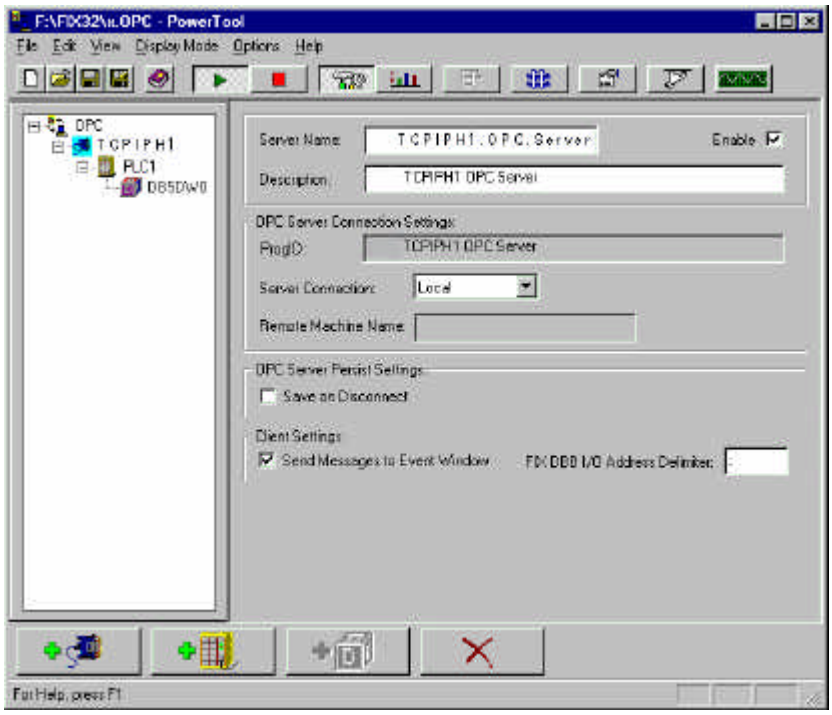
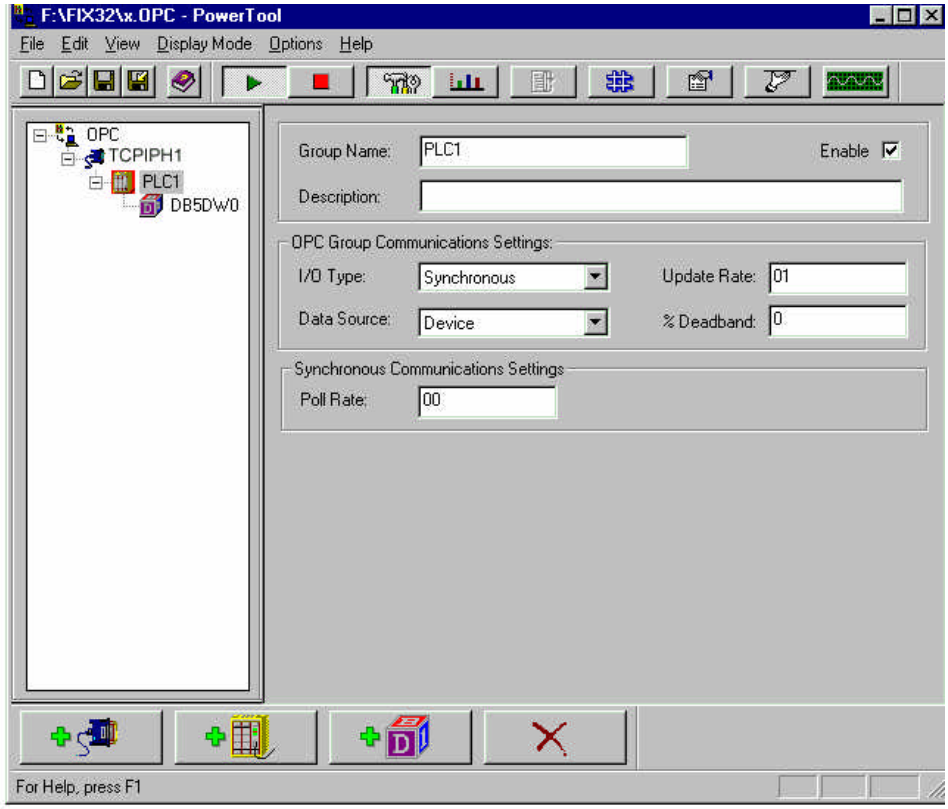


FIGURE 30.
Entering the application name „INAT TCPIPH1 OPC-Server“

In the next window you have to enter the acces path (connection) name. Please enter „PLC1“.

FIGURE 31.
Entering the access path name



At least the Item name has to be entered. Each item is linked to the corresponding connection (in this case „PLC1“).

Mind for the data type entry that the OPC server supports the following data types:

Tag Name	Type compatible	also unsigned types	standard txpes
DB5D1.0	VT_BOOL	VT_BOOL	VT_BOOL
DB5DB1	VT_I4	VT_UI1	VT_UI1
DB5DB1KF	VT_I4	VT_BSTR	VT_BSTR
DB5DL1	VT_I4	VT_UI1	VT_UI1
DB5DR1	VT_I4	VT_UI1	VT_UI1
DB5DW1	VT_I4	VT_UI2	VT_I4
DB5DW1KF	VT_I4	VT_I2	VT_I2
DB5DW1KT	VT_BSTR	VT_BSTR	VT_BSTR
DB5DW1KC	VT_I4	VT_UI2	VT_I4
DB5DW1BCD	VT_I4	VT_I2	VT_I2
DB5DD1	VT_I4	VT_UI4	VT_R8
DB5DD1KF	VT_I4	VT_I4	VT_I4
DB5DD1KG	VT_R4	VT_R4	VT_R4
DB5DD1IEEE	VT_R4	VT_R4	VT_R4
DB5S1.10	VT_BSTR	VT_BSTR	VT_BSTR
DB5DW1KF	VT_I4	VT_I2	VT_I2
DB5DW2KF	VT_I4	VT_I2	VT_I2
DB5DW0.10	----------	VT_ARRAY	VT_ARRAY

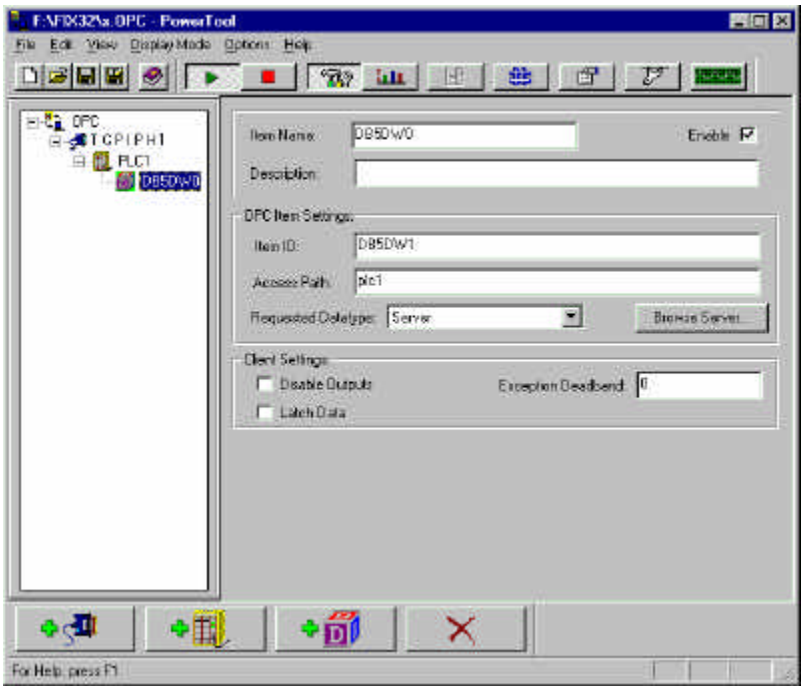


FIGURE 32.
Entering the item name

In the status display you are able to monitor the current state of the defined item

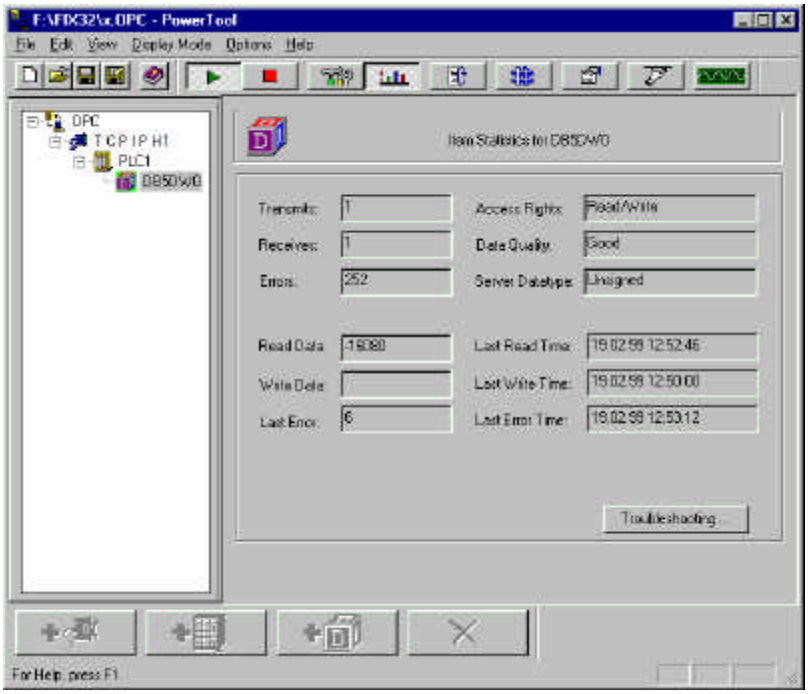


FIGURE 32.
State of the item

5.4.3 Monitoring the Access Path and the Item in the OPC server

If the OPC communication is configured corectly in your process visualization and in the INAT OPC server, the following entries should be displayed in the main window of the OPC Server.

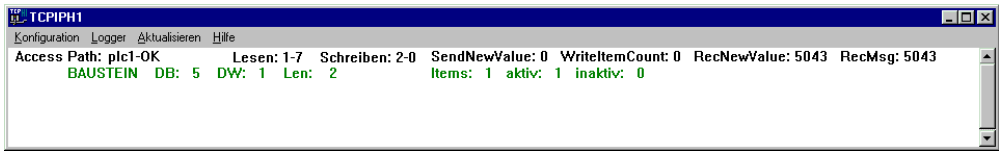


FIGURE 33.
Display of the access path plc1 and the ITEM DB5 DW1 in OPC server

6 Introductory Example

BEACHTEN SIE

Users of the OPC server H1

The following chapter doens´t pertain your OPC server. You can go directly to chapter „Request for Comments“ on page 129.

6.1 Overview and Prerequisites

This chapter introduces you to the communication capabilities of the OPC servers with special emphasis on using the parameterization software of the OPC servers. The purpose of this chapter is to establish a small communication system which reads data from the PLC.

CP handling blocks are standard function blocks of SIMATIC controllers. They permit utilization of the communication functions via programs in the programmable logic controllers (i.e., PLCs)

A knowledge of CP handling blocks is required.

The following examples require the technical components listed below.

- ***PC with Windows NT, InTouch, network card***
- ***Programmable controller***
- ***1 S5-TCP/IP communications processor***
- ***Transmission path consisting of***
  - ***1 bus cable with 2 coaxial T-pieces***
  - ***2 terminators***
  - ***1 serial transmission cable for programming***
- ***Software packages***
  - ***InTouch***
  - ***INAT OPC server***
  - ***INAT parameterization software***
  - ***STEP 5 or S5 for Windows programming package for PLC programming***
  - ***Standard handling blocks for the appropriate PLCs if nor already present***

This section explains communication between PLC and S5-TCP/IP with the standard handling blocks, based on the task stated in section „Task Definition“ on page 116. Section „Communication S5 CPU / INAT S5-TCP/IP“ on page 117 contains the appropriate standard handling blocks for the S5-115U programmable controller. Communication between PLC and process vosualization also requires the parameterization of the OPC server and the S5-TCP/IP communications processor. This is described in section „Parameterization for the Introductory Example“ on page 121.

6.2 Task Definition

The introductory example for the application of the S5-TCP/IP H1 is based on a communication task. The following figure shows the required plabnt configuration with the components listed in section „Overview and Prerequisites“ on page 115.

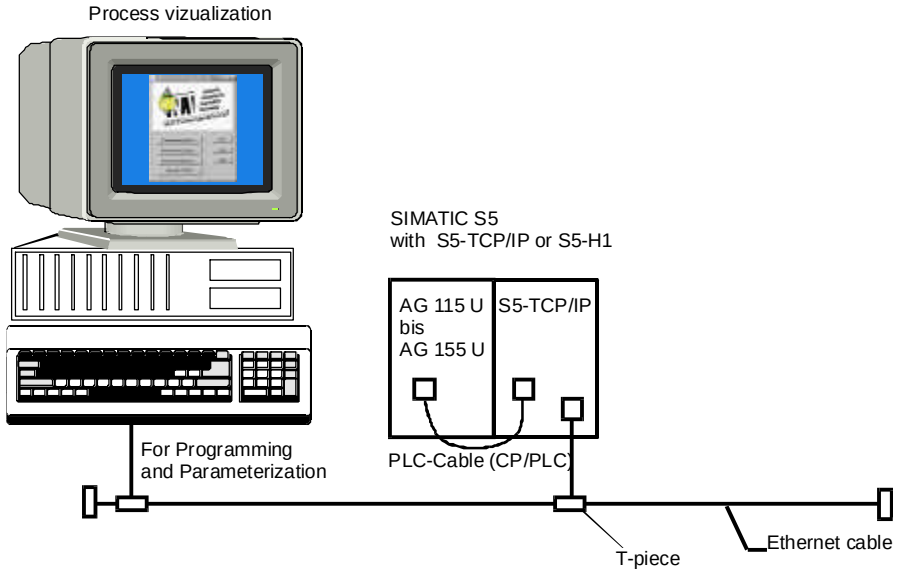


FIGURE 34.
Plant configuration for the introductory example

The task

Process visualization is to use the TCP/IP network to read data from the SIMATIC S5.

This gives you a rough idea of the task and the required presettings.
Section „Communication S5 CPU / INAT S5-TCP/IP“ on page 117 contains additional details on parameterizing the handling blocks. Section„Introductory Example“ on page 115 provides a detailed description of the parameterization of the S5-TCP/IP.

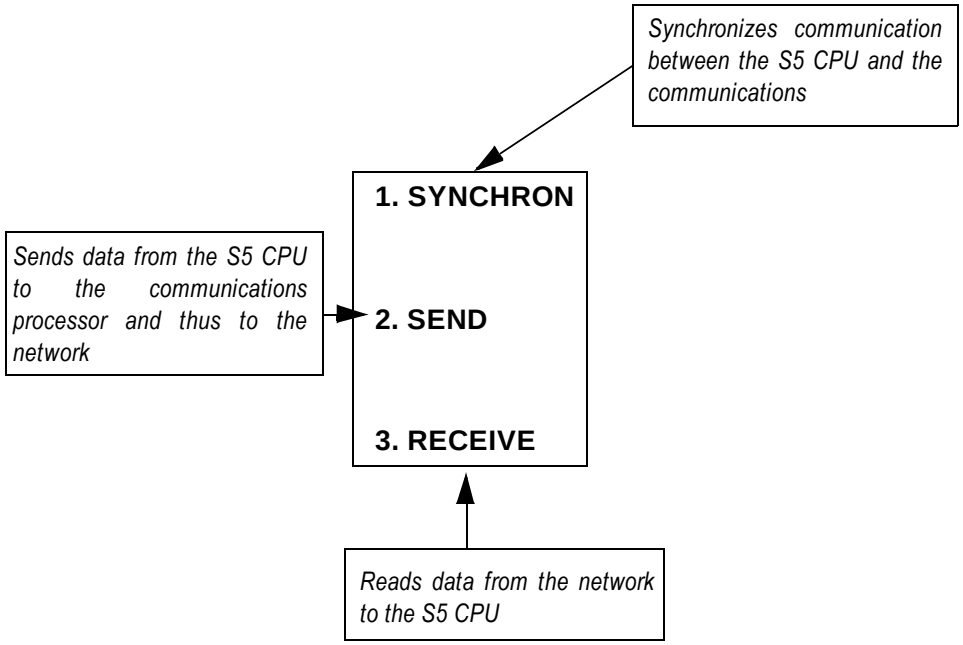
NOTE!

If not already done, set the IP address of your PC under NT. In the "Start/Settings/Control Panel/Networks" menu, enter the values for the IP address and the subnetwork mask in the "Protocols" window.
If you will be executing this example with new S5-TCP/IP modules, the TCP/IP addresses have not yet been set. For how to perform the basic parameterization of a TCP/IP station, see the S5-TCP/IP manual.

6.3 Communication S5 CPU / INAT S5-TCP/IP

Communication of the CPU with the communications processor is handled by a memory which can be accessed by both systems (i.e., a dual port RAM). This permits high-speed data communication between both modules. On the software side, communication is handled by the standard handling blocks.

There are three types of jobs:



NOTE!

The numbers of the blocks vary from controller to controller. All parameters are also not exactly the same

The following example describes the standard handling blocks for the S5-115U CPU.

SYNCHRON should be called for a cold start or a warm start

OB 21 and OB 22	
;	
SPA FB 249	Block numbers vary from PLC to PLC
NAME: SYNCHRON	
SSNR KY 0,0;	The interface number (page frame basis) of the communications processor
BLGR: KY 0,6	;Size of the page frame: 512 Bytes (the maximum)
PAFE MB 255	;Parameterization error indication

The S5 CPUs call block Ob1 to jump to the cyclic portion. This is the best place for the SEND and RECEIVE calls.

OB 1		
O	M 0.0 ;Set RLO	
ON	M 0.0	
SPA	FB 244	Block numbers vary from PLC to PLC
NAME: SEND		
SSNR: KY 0,0	;The interface number (page frame basis) of the communications processor	
A-NR: KY 0,0	;Job number 0, (SEND ALL)	
ANZW: MW 10	;Indication word for error monitoring. Is never addressed for ALL jobs.	
QTYP: KC NN	;Source type NN is not used	
DBNR: KY 0,0	;Block number. Not used with ALL calls	
QANF: KF 0	;Start block. Not used with ALL calls	
QLAE: KF 0	;Block length. Not used with ALL calls	
PAFE: MB 255	;Parameterization error indicattion	
:		
:		
O	M 0.0 ;Set RLO	
ON	M 0.0	
SPA	FB 245	Block numbers vary from PLC to PLC
NAME: RECEIVE		
SSNR: KY 0,0	;The interface number (page frame basis) of the communications processor	
A-NR KY 0,0	;Job number 0, (RECEIVE ALL)	
ANZW:MW 10	;Indication word for error monitoring. Is never addresses for ALL jobs.	
ZTYP KC NN	;Source type NN is not used	
DBNR: KY 0,0	;Block number. Not used with ALL calls	
ZANF: KY 0	;Start block. Not used with ALL calls	

```
OB 1
ZLAE: KF 0          ;Block length. Not used with
                     ALL calls
PAFE MB 255         ;Parameterization error indica-
                     tion
```

If the communications processor is correctly initialized and parameterized, the example controls commmunication from PC to PLC.

For configuration of the communications processor, see applicable manual.

The OPC server also permits data to be written from the S5 to the PC (FB 244 SEND). This function requires a program on the S5 CPU based on the following example:

```
  O     M 0.0  ;Set RLO
  ON    M 0.0
  SPA   FB 244          Block numbers vary from PLC to PLC
NAME: SEND
SSNR: KY 0,0            ;The interface number (page frame basis)
                         communications processor
A-NR:  KY 0,10          ;Job number. 10 in our example
ANZW:MW 10              ;Indication word for error monitoring. Is never
                         addressed for ALL jobs.
QTYP:KC DB              ;Destination type
DBNR:KY 0,11            ;Block number 11
QANF:KF 0               ;Start block. 0 in our example
QLAE:KF 10              ;Length. 10 in our example
PAFE:MB 255             ;Parameterization error indication
```

This is also possible with indirect parameterization starting with the block.

```
  O      M 0.0   ;Set RLO
  ON     M 0.0
  SPA    FB 244                Block numbers vary from PLC to PLC
NAME: SEND
SSNR: KY 255,0                 ;The interface number (page frame basis) communications processor
A-NR:  KY 0,10                 ;Job number. 10 in our example
ANZW:MW 10                     ;Indication word for error monitoring. Is never addressed for ALL jobs.
QTYP:KC RW                     ;Destination type. For indirect parameterization here.
DBNR:KY 0,11                   ;Block number 11
QANF:KF 0                      ;Start block. 0 in our example
QLAE:KF 0                      ;Length. 0 in our example
PAFE:MB 255                    ;Parameterization error indication
```

The following coding must be located in data block 11 starting at word 0:

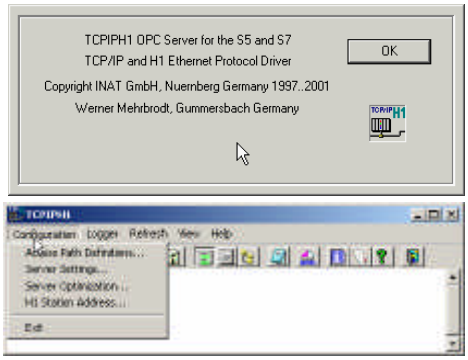
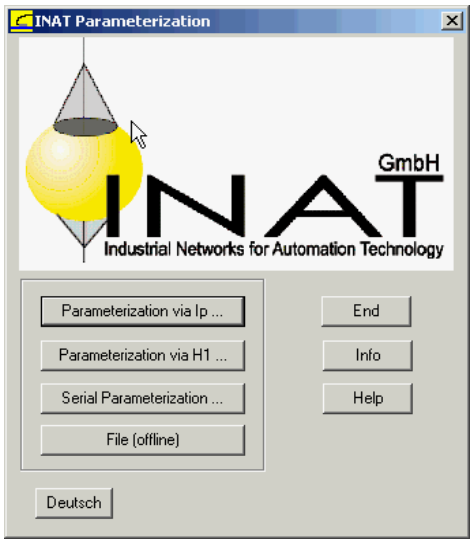
```
DW 0=KC 'DB'                   ;Block type of the active CPU
DW 1=KY 0,10                   ;Block number in the active CPU
DW 2=KF 10                     ;Start block in the active CPU
DW 3=KF 1                      ;Length in the active CPU
DW 4=KC 'DB'                   ;Block type of the destination CPU
DW 5=KY 0,10                   ;Block number of the destination CPU
DW 6=KF 10                     ;Start block of the destination CPU
DW 7=KF 1                      ;Length of the destination CPU
```

The OPC server takes the information for the destination from the communication header. This header contains the destination parameters from the PLC program. To prevent errors, we recommend keeping the values for source and destination parameters the same.

6.4 Parameterization for the Introductory Example

The same procedure is used to parameterize the PC and the PLC. Only the parameters to be set are different. Parameterization of both stations will now be described. Each station must be parameterized separately.

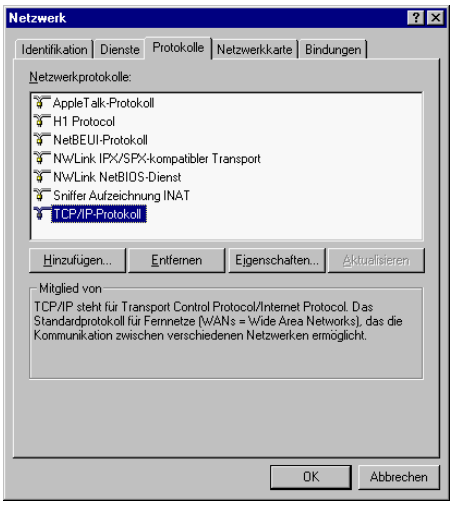
First, establish your network. To parameterize the stations, perform the following steps for each station. Remember: The entries for the OPC server are shown in the left column, while the entries for the parameterization program of the S5-TCP/IP are shown in the right column.

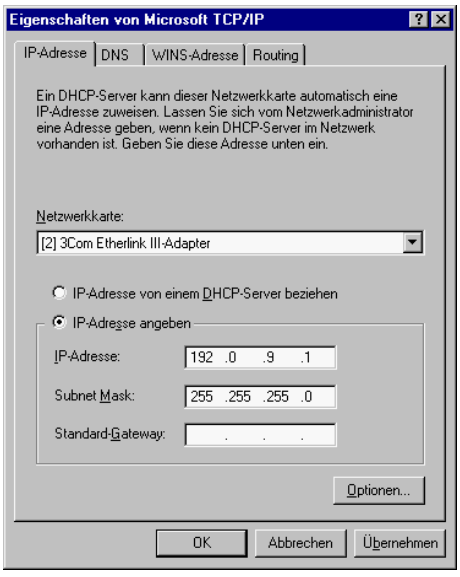
OPC server	S5 TCP/IP Parameterization
If you have not already done so, install the OPC server on your PC now. See chapter „Installation and Program Start“ on page 23 for information on installation and starting the program. Start the OPC server. In the menu „Configuration“ select menu item „Access Path Definitions“ 	If you have not already done so, install the parameterization software on your PC now. See the S5-TCP/IP manual for information on installation and starting the program.  Click „Parameterization via IP“ with the mouse. A dialog window appears with a selection of available stations. H1 stations are also shown. If only one PLC station exists in the network, only this one station is shown.

OPC server

PARAMETERIZING THE STATION

If you have not already done so, set the IP address of your PC under NT. In the menu „Start/Settings/Control Panel/Network“, enter the following values in the „Protocols/Properties“ window for the IP address and the subnet word mask. Then start the OPC server again.

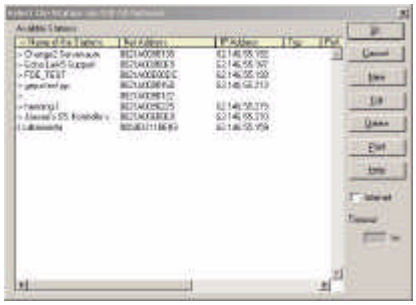




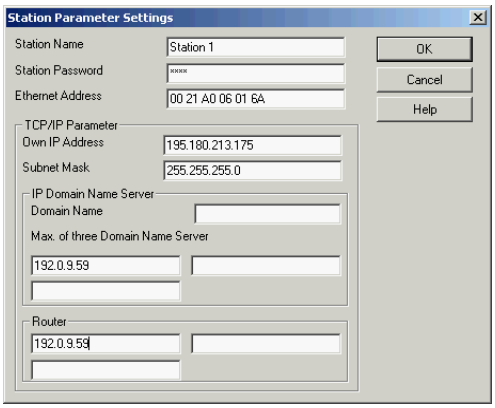
S5 TCP/IP Parameterization

PARAMETERIZING THE STATION

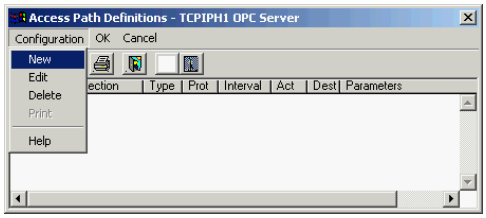
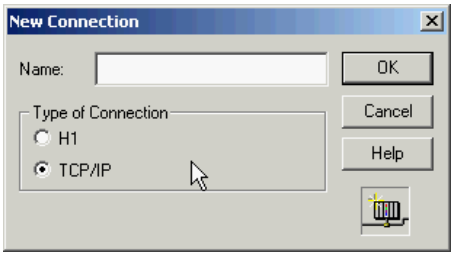
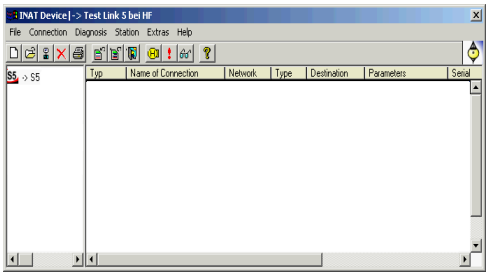
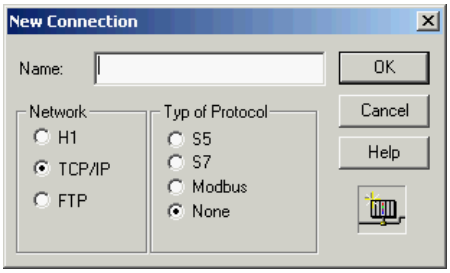
If you will be executing this example with new S5-TCP/IP modules, the station name and the TCP/IP addresses have not yet been set. Only the Ethernet address is assigned in the „Select the station in the network“ window.

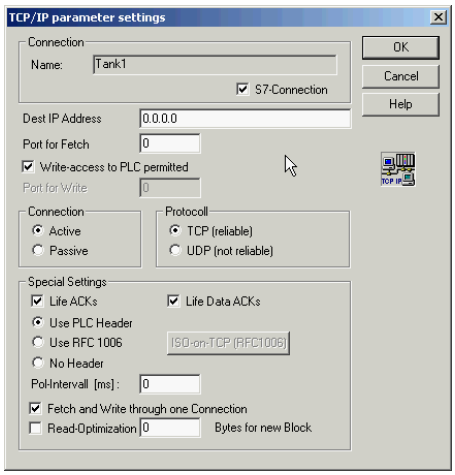
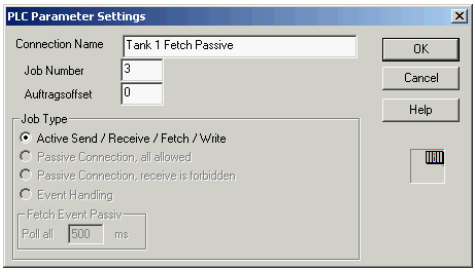


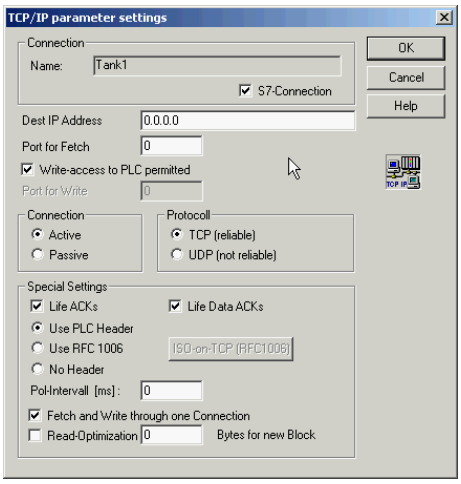
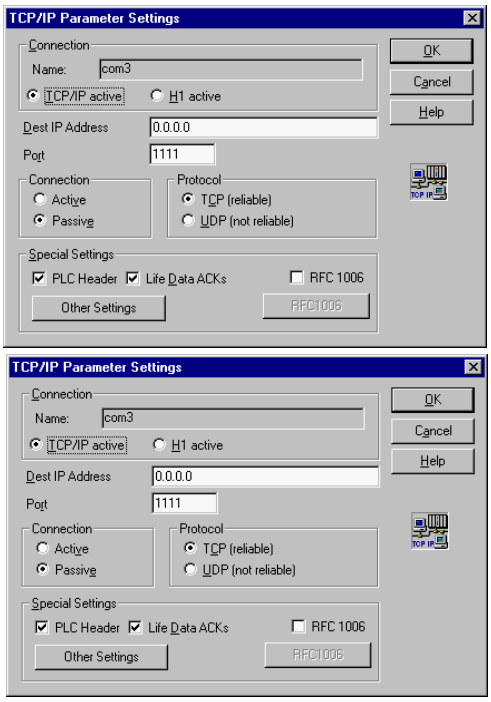
Click the „new“ button. The „New Connection to a station“ window appears. Enter the Name and the IP address and click okay. The window „Station parameter settings“ appears in which the following parameters are entered:

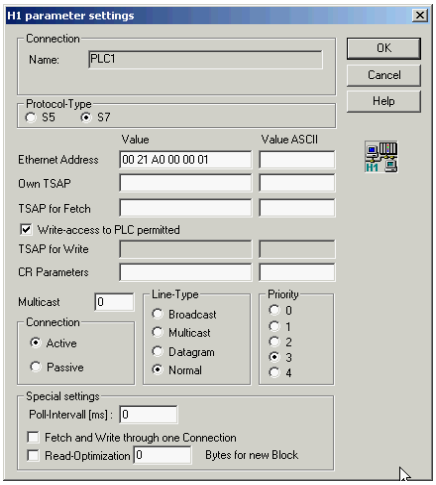
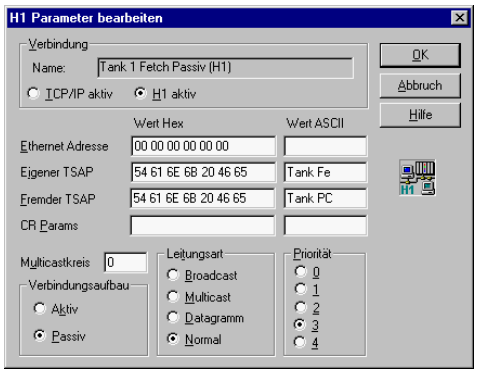
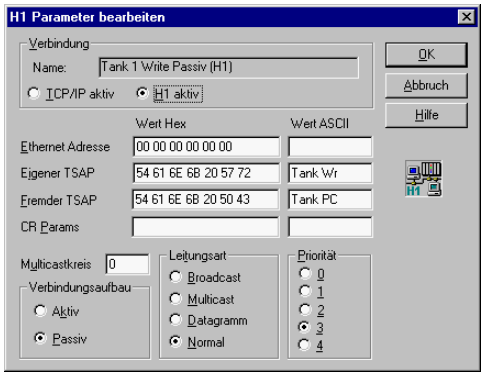


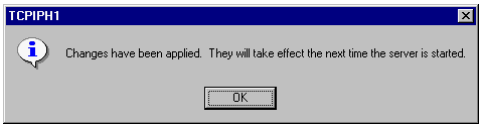
Online stations are identified by an arrow in front.

OPC server	S5 TCP/IP Parameterization
PARAMETERIZE NEW CONNECTION After selection of the „Access Path Definitions“ the following window appears:  At this time, no connections have been parameterized. Click „Configuration/New“ to set up a new connection.  Enter the name (e.g. Tank 1) for communication with the controller and select type TCP/IP. When InTouch is used, the connection name corresponds to the OPC access path.	**PARAMETERIZE NEW CONNECTION** Click the OK button in the „Select a Station“ window to access the input mask S5-TCP/IP Connection list“  At this time, no connections have been parameterized. Click Connection/New to set up a new connection  Enter the name (e.g. Tank 1 Fetch Passive) for the response to the visualization PC, and select connection type TCP/IP.

OPC server	S5 TCP/IP Parameterization
Click OK for editing the TCP/IP parameters.  A seperate dialog window for the S5 parameters does not exist, since the „Job number“ and „job offset“ parameters do not exist and the „job type“ is specified.	Click OK to open the „S5 Parameterization“ window. The S5 parameters control the connection between the S5-TCP/IP module and the S5. In addition to „Job Type“, „Job number“ and „Job offset“ are specified in this window. See the S5-TCP/IP manual for a detailed description of editing the S5 parameters.  Repeat the last two steps with the entries New Connection: plc1 Write Passive, Type of job: Write Passive Job offset: 0 Job number: 2

OPC server	S5 TCP/IP Parameterization
Editing TCP/IP patameters	**Editing TCP/IP patameters**
	
	On the PLC side, seperate connections are parameterized for the connections to the OPC server „Fetch“ from the PLC (plc1 Fetch Passive) and „Write to the PLC (plc1 Write Passive)“.

OPC server	S5 TCP/IP Parameterization
H1 Parameter Settings If you want to use H1 for the connections, activate the H1 protocol in the window „New connection“. The „H1 Parameter Settings“ window appears. Enter the following values for the station „plc1“.  	**H1 Parameter Settings** If you want to use H1 for the connections, activate the H1 protocol in the window „New Connection“. Specify the S5 parameters as shown above Type of job: Fetch Passive/Write Passive Job number: 1/2 Job offset: 0/0 The H1 Parameter Settings window appears. Enter the following value for the respective station.   On the PLC side, seperate connections are parameterized for the connections to the OPc server „Fetch from the PLC (plc1 Fetch Passive)“ and „Write to the PLC (plc1 Write Passive)“.

OPC server	S5 TCP/IP Parameterization
Connection List Click the „Close“ button. The dialog windows for editing the H1 or TCP/ IP parameters are closed. Since the connections have been parametze-rized, they appear in the connection list of the parameterization window.	**Connection list** Click the „Close“ button. The dialog windows for editing the H1 or TCP/ IP parameters are closed. Since the connections have been parametze-rized, they appear in the connection list of the parameterization window.
Close the dialog window. The newly set up and parameterized connec-tions do not take effect until a new start is performed for the OPC ser-ver.  Exit the OPC server and start it again.	Exit the S5-TCP/IP parameteriza-tion or monitor the newly parame-te-rized connections via the menu item „Online Indications“

The connections between the VISU PC and the PLC have now been parame-te-rized.

7 Request for Comments

7.1 Literature

[RFC768] Postel, J., "User Datagram Protocol," STD 6, RFC 768, USC/Information Sciences Institute, August 1980

[RFC791] Postel, J., "Internet Protocol - DARPA Internet Program Protocol Specification," STD 5, RFC 791, DARPA, September 1981

[RFC792] Postel, J., "Internet Control Message Protocol - DARPA Internet Program Protocol Specification," STD 5, RFC 792, USC/Information Sciences Institute, September 1981

[RFC793] Postel, J., "Transmission Control Protocol - DARPA Internet Program Protocol Specification," STD 7, RFC 793, USC/Information Sciences Institute, September 1981

7.2 Protocols of the Network Layer

RFC 791 : **IP** (Internet Protocol)

The Internet protocol provides a way of sending datagrams from source to destination regardless of whether these devices are located in the same network or in different networks. In addition to this IP addressing, the protocol handles fragmentation (if requested) of data packets transferred by the transport instance. IP is not a secure service.

RFC 792 : **ICMP** (Internet Control Message Protocol)

ICMP is an Internet control protocol which informs the network of unexpected events detected by the routers. Each ICMP message type is enclosed in an IP packet.

RFC 826 : **ARP** (Address Resolution Protocol)

Since the hardware of the security layer (i.e., Ethernet card) does not understand 32-bit IP addresses, ARP locates the corresponding 48-bit Ethernet addresses by sending a broadcast packet to the Ethernet searching for the owner of a certain IP address. Every device in the network receives the packet and checks its IP address. The desired host then reports to the requesting host via its Ethernet address so that the transport layer can establish a connection.

RFC 903 : **RARP** (Reverse Address Resolution Protocol)

RARP permits a work station which has just booted to send out its Ethernet address and to request its own IP address. The RARP server "sees" this request, searches its configuration files for the Ethernet address, and sends back the appropriate IP address.

A server is required since routers do not forward broadcast messages.

RFC 951 : **BOOTP** (Bootstrap Protocol)

BOOTP is a protocol which helps diskless computers to boot. It uses UDP messages which are distributed by routers.

7.3 Protocols of the Transport Layer

RFC 793 : **TCP** (Transmission Control Protocol)

TCP is a secure, connection-oriented protocol which sends an error-free byte stream from one device to another via Internet. It splits up the incoming byte stream into single messages and forwards these to the network layer. At their destination, the single messages received by the TCP process are recombined into an output stream. TCP also handles flow monitoring (i.e., to prevent slow receivers from being overwhelmed with messages from high-speed senders). Connection establishment is based on the three-way handshake principle. Communication between senders and receivers is handled by TSAPs, often referred to as *sockets* in TCP/IP jargon.

RFC 1122 : Error corrections for TCP in accordance with RFC 793

RFC 1323 : Expansions for TCP in accordance with RFC 1122

RFC 768 : **UDP** (User Datagram Protocol)

UDP can be used by applications to transfer enclosed raw IP datagrams without having to establish a connection. This protocol is primarily used for one-time requests and applications in client/server environments in which speed is more important than precision (e.g., transmitting voice or video).

7.4 Protocols of the Processing Layer

RFC 1034

RFC 1035: **DNS** (Domain Name Service)

The DNS protocol was developed for conversion of ASCII-character-set host names and E-mail addresses into binary IP network addresses.
For example, the E-mail address "mary@eagle.cs.uni.edu" can be identified with the IP address "192.31.65.5." The DNS protocol uses a distributed data base system based on a hierarchical convention of names.

RFC 1441

bis 1452: **SNMP** (Simple Network Management Protocol)

SNMP is a systematic method of monitoring and administering a computer network.

8 Appendix

8.1 The PLC Header

The data stream-oriented TCP/IP protocol can combine several short data units into longer units. This feature increases data throughput on the network. As with other protocols (i.e., FTP and HTTP), this requires a data header ahead of the TCP in the protocol. The bytes of an 8-byte header contain the following information.

TABLE 22.
Format of the PLC header

No of the Byte	Meaning
Byte 0	0 x 4d ´M‘
Byte 1	0 x 4d ´K´
Byte 2[3]	Datalen LSB[1] Length of the data in the packet after
Byte 3[3]	Datalen MSB[2] the header
Byte 4[3]	Bit 0 = 1, if further frames follow
Byte 5[3]	0
Byte 6[3]	SeqNo. LSB[1]
Byte 7[3]	SeqNo. MSB[2]
Datalen Bytes	User data

[1]LSB: Least (Lower) Significant Byte
[2]MSB: Most Significant Byte
[3] Byte 2 / 3 Byte 4 / 5 und Byte 6 / 7 have the data value „short“. They are represented in Intel format

Acknowledges

If DataLen is 0, a life data acknowledge (Life Data Ack) is involved and not user data. Data acknowledges permit a form of connection monitoring which the TCP/IP actually does not provide for as remote-communication frame.

Sequence numbers

Byte 6 and byte 7 represent a sequence number which has the value 0 when the connection is established and which is incremented by one each time user data are sent. This frame counter is used as an additional safety mechanism for the data transmission. When life data acks are sent, the sequence number is not incremented and DataLen is 0.

Fetch and write connections

For fetch and write jobs, the first 16 data bytes correspond to the SINEC AP header when a job is started. The SINEC AP header is also used for communication via H1.

Sending/receiving data

When data are sent a maximum of 512 bytes of user data are sent in one frame. This maximum value is determined by the size of the page frame block which is set. When data are received, one data packet can contain up to 1460 bytes. Since these limits are automatically monitored by the TCP/IP protocol, no further monitoring is required on the user side.

Transmission without frame header

The header at the beginning of the frame can be disabled. If this is done, the application program on both sides is responsible for monitoring. Keep the following points in mind.

- Particularly in the case of the *Send Direct* and *Receive Direct* jobs, certain time limits until frame acceptance may not be exceeded. If these time limits were violated, the internal buffers would be overloaded (e.g., with inquiries), and synchronization of request and response would no longer be possible.
- A certain mechanism for blocked data transmission must be adhered to so that the end of the user data can be detected.
- The receiving side must ensure that the frames are read from the receiving buffer before the partner station sends the next frame.
- Creation of connection monitoring in the application program is indispensable.

8.2 Designation of the standard handling blocks

The designation of the communication-related standard handling blocks differs from programmable controller to rpogrammable controller. The following tabel shows the assignments.

TABLE 23.
Handling blocks

Handling block	PLC 115U/H	PLC 135U/AG 155U/H	PLC 135U/AG 155U/H
	CPU: 941, 942, 942R, 943, 944, 94	CPU: 922, 928, 948, 948 (R)	CPU: 946/47, 946/47R
SEND	FB 244	FB 120	FB 120
RECEIVE	FB 245	FB 121	FB 121
FETCH-n	FB 246	FB 122	FB 122
CONTROL	FB 247	FB 123	FB 123
REST-n	FB 248	FB 124	FB 124
SYNCHRON	FB 249	FB 125	FB 125
SEND-ALL	FB 244, A-Nr. = 0	FB 126	FB 126
RECEIVE-ALL	FB 245, A-Nr. = 0	FB 127	FB 127

INDEX

Figure Caption