

SIEMENS

WinCC

Communication Manual

Manual 1

This manual is part of the documentation package
with the order number:

6AV6392-1CA05-0AB0

C79000-G8276-C155-01

Release: September 1999

WinCC, SIMATIC, SINEC, STEP are trademarks of Siemens.

The other names used in this manual may be trademarks; their owners' rights may be violated if they are used by third parties for their own purposes.

(The transmission and reproduction of this document, and utilization and disclosure of its contents are not permitted unless expressly authorized.
Offenders will be liable for damages. All rights, including rights created by patent grant or registration of a utility model or design, are reserved.)

(We have checked the contents of this manual for agreement with the hardware and software described. Since deviations cannot be precluded entirely, we cannot guarantee full agreement. However, the data in this manual are reviewed regularly and any necessary corrections included in subsequent editions. Suggestions for improvements are welcomed.)

© Siemens AG 1994 - 1999 All rights reserved

Technical data subject to change

C79000-G8276-C155-01
Printed in the Federal Republic of Germany

Siemens Aktiengesellschaft

Table of contents

1	Communication Manual	1-1
2	General Information Chapter	2-1
2.1	Basics of Communication.....	2-1
2.1.1	Basic Communication Terms	2-1
2.1.2	Network Topology	2-3
2.1.3	Classification of Networks	2-6
2.1.4	Access Methods	2-7
2.1.5	ISO-OSI Reference Model	2-8
2.1.6	Connection of Bus Systems	2-10
2.2	Communication Networks	2-11
2.2.1	Industrial Communication Overview.....	2-11
2.2.2	Industrial Communication Subnets.....	2-13
2.2.3	Industrial Communication via MPI.....	2-14
2.2.4	Industrial Communication via PROFIBUS.....	2-15
2.2.4.1	The PROFIBUS Access Method	2-16
2.2.4.2	PROFIBUS Protocol Architecture.....	2-17
2.2.4.3	Transfer Media	2-17
2.2.5	Industrial Communication via Ethernet	2-18
2.2.5.1	Industrial Ethernet Protocol Profiles.....	2-19
2.2.5.2	Transport Protocols	2-19
2.2.5.3	Transfer Media	2-20
2.2.6	OPC Interface Standard	2-20
2.3	Communication Configuration	2-21
2.3.1	WinCC Process Communication	2-21
2.3.2	WinCC Communication Configuration	2-23
3	Selection Guide.....	3-1
3.1	Project Analysis.....	3-1
3.1.1	Specifications	3-2
3.1.2	Configuration Notes.....	3-7
3.1.2.1	Data Update Cycle	3-7
3.1.2.2	Data Update Type	3-8
3.1.2.3	Data Organization	3-9
3.2	Performance Data	3-11
3.2.1	Comparison of Communication Systems	3-11
3.2.2	Serial Communication	3-14
3.2.3	Communication via MPI	3-15

3.2.3.1	Communication Processors	3-15
3.2.3.2	SIMATIC S7 PROTOCOL SUITE.....	3-15
3.2.4	Communication via PROFIBUS	3-16
3.2.4.1	Communication Processors	3-16
3.2.4.2	Communication Drivers	3-17
3.2.4.3	PROFIBUS DP	3-19
3.2.4.4	PROFIBUS FMS.....	3-20
3.2.4.5	SIMATIC S7 PROTOCOL SUITE.....	3-21
3.2.4.6	SIMATIC S5 PROFIBUS FDL	3-22
3.2.5	Communication via Industrial Ethernet.....	3-23
3.2.5.1	Communication Processors	3-23
3.2.5.2	Communication Drivers	3-24
3.2.5.3	Communication with the SIMATIC S5.....	3-25
3.2.5.4	Communication with the SIMATIC S7	3-27
4	Diagnosis of the Communication Connection	4-1
4.1	Error Detection	4-1
4.1.1	WinCC Explorer.....	4-2
4.1.2	Channel Diagnosis	4-3
4.1.3	Status Monitoring using the Dynamic Dialog.....	4-5
4.1.4	Status Monitoring using Global Actions.....	4-7
4.2	Troubleshooting Guideline	4-8
4.2.1	Is the Communication Partner operational?	4-8
4.2.2	Is the Network Connection operational?	4-10
4.2.3	Is the Communication Module in the Computer operational?	4-12
4.2.4	Is the Communication Configuration correct?	4-17
4.3	Further Diagnosis	4-24
4.3.1	Creating Connections via API Functions.....	4-25

Preface

Purpose of this Manual

This manual details the planning, installation and startup of a communication connection from WinCC to a PLC. It is organized into the following sections:

- A general chapter about the topic communication
- A chapter assisting in selecting the most suitable communication solution
- A chapter about error diagnostics

This manual is available in printed form as well as an electronic online document. The table of contents or the index will quickly point you to the information required. The online document also provides an expanded search function.

Additional Support

For technical questions, please contact your Siemens representative at your local Siemens branch.

In addition, you can contact our Hotline at the following number:
+49 (911) 895-7000 (Fax -7001)

Information about SIMATIC Products

Constantly updated information about SIMATIC products can be found in the CA01 catalog. This catalog can be accessed at the following Internet address:
<http://www.ad.siemens.de/ca01online/>

In addition, the SIEMENS Customer Support provides you with current information and downloads. A collection of frequently asked questions is listed at the following Internet address:
http://www.ad.siemens.de/support/html_00/index.shtml

1 Communication Manual

Introduction

The Communication Manual is part of the WinCC documentation. It details the planning, installation and startup of a communication connection from WinCC to a PLC.

The objective of the communication manual is to support the user beginning with the selection of the optimum communication, its configuration and installation, up to the startup. Additionally, the communication manual points the way to a quick solution in case of a problem.

Contents and Structure

The communication manual is organized into four main chapters:

- **General Information Chapter**The first chapter provides the reader with a general overview on the subject of communication, with an emphasis on the communication to WinCC. The necessary technical background knowledge, components used and general configuration steps are detailed.
- **Selection Guide**The second chapter guides the reader through the process of selecting the optimum communication type for his or her application. The general factors that must be considered in making a decision are explained.
- **Diagnosis of the Communication Connection**The third chapter contains instructions for locating and correcting errors, should there be problems with the startup of the communication connection.
- **Sample Projects**The fourth chapter contains a selection of sample projects with very detailed descriptions pertaining to different communication options. Each sample project details the steps from the installation of the hardware up to the successful startup of the connection.

Conventions

The communication manual uses the following conventions:

Convention	Description
	Denotes an operation using the left mouse button.
	Denotes an operation using the right mouse button.
	Denotes an operation using a double-click of the left mouse button.
<i>Italic</i>	Denotes terms of the WinCC environment and terms referring to the elements of the program's interface.
<i>Italic, Green</i>	Denotes an operating sequence or entry to be followed by the user (color visible only in the online document).
Blue	Cross references (links) are in blue (color visible only in the online document).

Finding Information

In the printed version of the communication manual, information can be found in the following ways:

- The table of contents lists information organized by topic.
- The index lists information organized by key word.

In the online document, information can be found in the following ways:

- The Contents tab lists information organized by topic.
- The Index tab lists information organized by key word.
- The Find tab allows you to search for words in the entire document.

The sample projects described in this manual can directly be copied from the online document to your hard disk drive

2 General Information Chapter

This chapter provides you with a general overview on the subject of communication, with an emphasis on the communication to WinCC. The necessary technical background knowledge, components used and general configuration steps are detailed.

This chapter is organized into the following sections:

- Basics of Communication
- Communication Networks
- Communication Configuration

2.1 Basics of Communication

This section of the manual provides you with basic information on the subject of communication. The information presented is universal, i.e. it is not only valid for the communication with WinCC.

This section contains information about the following topics:

- Basic Communication Terms
- Network Topology
- Classification of Networks
- Access Methods
- ISO-OSI Reference Model
- Connection of Bus Systems

2.1.1 Basic Communication Terms

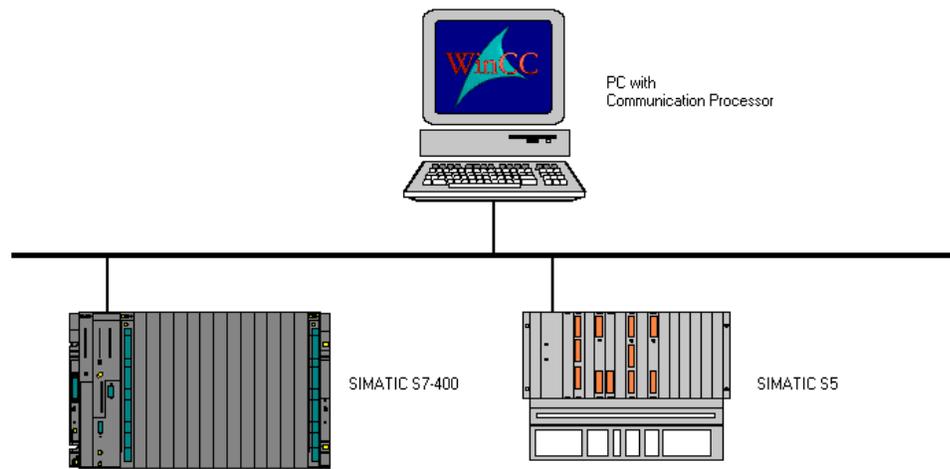
This section explains the basic terms pertaining to the subject of communication. It mainly deals with the areas important for the information exchange between a PLC and WinCC. The following explains the most important communication terms, which are used for the information exchange between PLCs and WinCC:

Communication

Communication is the term used to describe the transfer of data between two communication partners.

The data transferred can serve different purposes. In the case of the communication between a PLC and WinCC, the data can be used to:

- Control the communication partner
- Display the states of the communication partner
- Report unforeseen states in the communication partner
- Archive



Communication Partner

Communication partners are modules capable of communicating with each other, i.e. they can exchange data with each other. They can be central processor boards and communication processors in PLCs or communication processors in PCs.

Station

A station is a device that can be connected to one or multiple subnets as a unit. This can be a PLC or PC.

Subnet(work)

A subnet is the term used to describe the unit consisting of all physical components necessary to build a data link and the associated data exchange method.

Network

A network is a unit consisting of one or multiple subnets connected to each other, which can be identical or different. It includes all stations that can communicate with each other.

Connection

A connection is a configured logical assignment of two communication partners for the execution of a defined communication service.

Each connection has two end points, which contain necessary information for addressing the communication partner and additional properties for establishing the connection.

Communication Functions

Communication functions are functions provided by a software interface, which are used by the communication services.

Communication functions can transfer data between communication partners having different performance specifications. They can control communication partners or retrieve their current operating state.

Communication Service

A communication service describes communication functions with defined performance characteristics, e.g. pertaining to the exchange of data, controlling or monitoring devices.

Software Interface

Software interfaces provide communication services in end systems. A software interface does not necessarily provide all communication functions of a communication service.

A communication service in the respective end system (PLC, PC) can be made available using different software interfaces.

Protocol

A protocol is an exact declaration by bit between the communication partners in order to perform a specified communication service.

The protocol defines the content-related structure of the data traffic on the physical line. Among other things, it defines the operating mode, procedure for establishing a connection, data protection and data transfer rate.

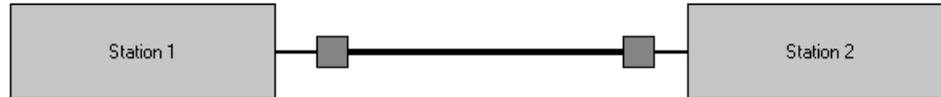
2.1.2 Network Topology

This section describes the different structures within a subnet.

If multiple, independent automation components are to exchange data with each other, they must be physically connected to each other. This physical connection can be structured differently. Network topology is the basic geometric arrangement of this structure. The individual communication partners form the nodes of this structure.

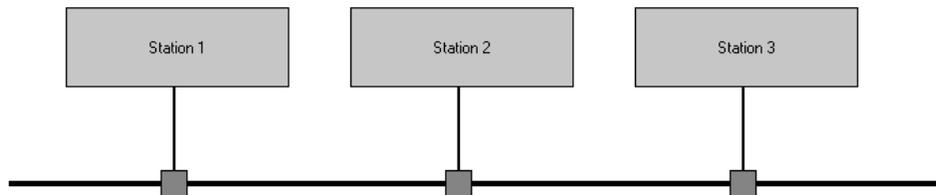
Point to Point

This most simple structure is the result of a network that only consists of two communication partners. This arrangement is referred to as a Point to Point connection.



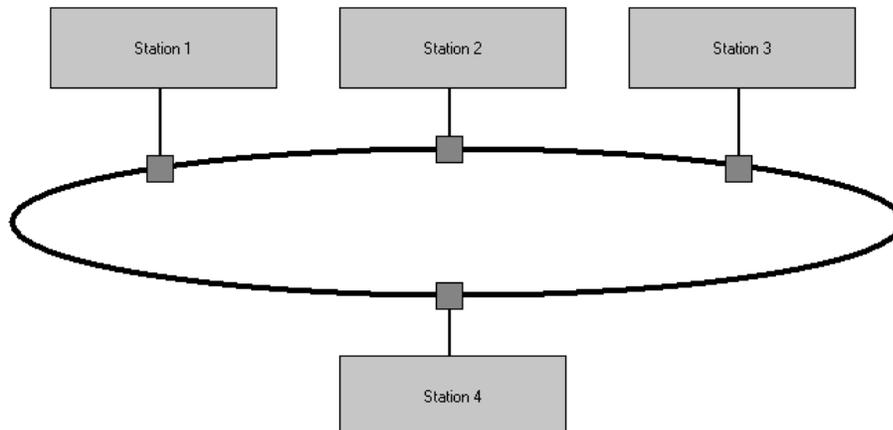
Line

A network arrangement in line structure follows a main line, the so-called bus. All communication partners are connected to the bus via feeder lines. Multiple communication partners can not talk simultaneously. Only one communication partner can send at any one time. This necessitates rules that are called bus access methods. These rules are also required for the following structures. The failure of a communication partner has little or no effect on the entire network.



Ring

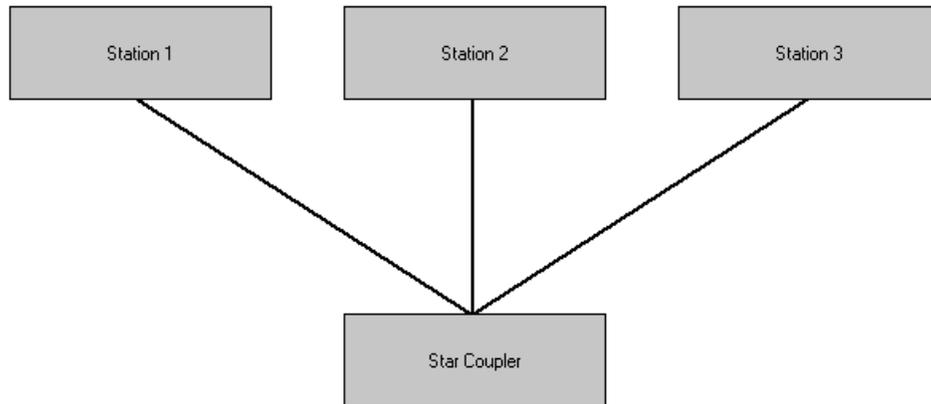
In this structure, the communication partners are connected with each other in such a way that they form a ring. A ring can be composed of sequentially arranged point to point connections. In a network structured like this, each node can act as a repeater. This allows greater distances to be bridged. The failure of a communication partner, however, causes greater problems in the ring structure than in the line structure.



Star

In a star structure, all communication partners are connected to a central star coupler. This star coupler controls the entire communication.

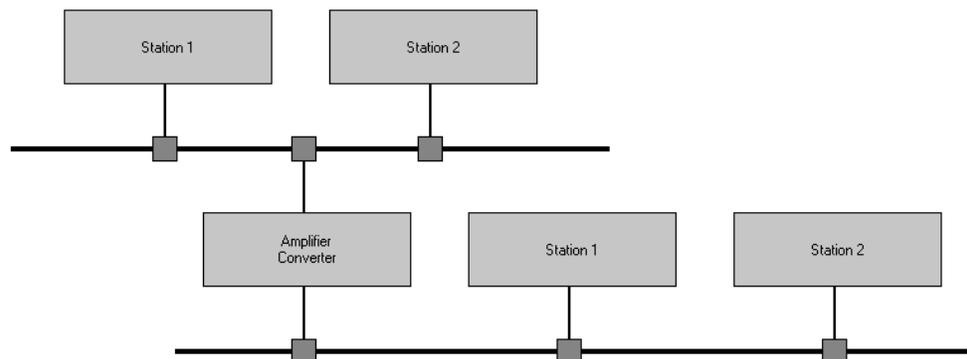
Generally, the failure of the star coupler brings down the entire network. The failure of a communication partner has little or no effect on the entire network.



Tree

The tree structure can be viewed as linked line structures. The line structures can have different dimensions and be of different types.

The elements connecting the individual lines are of special importance. If the lines to be connected are of the same type, these elements can be pure repeaters. If the lines to be connected are of different types, converters are necessary.



2.1.3 Classification of Networks

This section explains the different network classes as pertaining to their geographic size.

- With regard to geographic size, networks can be classified into three classes. These are:
- LAN (Local Area Network) : Size < 5 km
- MAN (Metropolitan Area Network) : Size < 25 km
- WAN (Wide Area Network) : Size > 25 km

An exact classification into one of these classes is sometimes not possible due to uncertain limits.

Topology

Because the distances to be bridged differ, information is available with regard to the topologies used.

The topology of a WAN is dictated by its geographic requirements. In most cases, due to economic reasons, irregularly meshed tree structure networks are the result. The topology of a LAN is more clearly structured, since overall functionality is of greater importance than line economies. Typical topologies for a LAN are line, ring and star.

Transfer Medium

The selection of the physical transfer medium depends on the desired network size, interference immunity and transfer rate. Sorted by increasing complexity and performance capability, particularly the following transfer media are found:

- Two-Wire Non-Twisted, Unshielded
- Two-Wire Twisted, Unshielded
- Two-Wire Twisted, Shielded
- Coaxial Cable
- Fiber Optic Cable

2.1.4 Access Methods

This section explains the mechanisms used to regulate how the individual communication partners access the bus.

Only one message can be sent to the bus at a time. This requires the regulation of when a communication partner can send. In this case, the number of listening recipients is of no importance.

The rules determining when a communication partner can send messages are called access methods.

Master Slave

In the master slave method, the master directs the entire bus traffic. The master sends data to the connected slaves and also prompts them to send data.

In most cases, a direct communication among the slaves is not provided. This method distinguishes itself for being a simple and therefore efficient bus controller.

The master slave method is also used for field busses. An example for this is the PROFIBUS-DP.

Token Passing

In the token passing method, a token travels through the network signaling the right to transmit. The token is a fixed bit pattern.

The owner of the token has the right to transmit. However, he can not hold the token for longer than the previously agreed upon time.

The bus access of the masters in the PROFIBUS network is regulated by the token passing method.

CSMA/CD

In the CSMA/CD method (Carrier Sense Multiple Access with Collision Detection), each communication partner can send at any time. However, no other communication partners can send at the same time.

Conflict arise, if two communication partners start sending at the same time due to signal delays. In this case, both will detect a collision and stop sending. After a certain time, they will attempt to send again.

Industrial Ethernet uses the CSMA/CD method.

2.1.5 ISO-OSI Reference Model

If the data exchange between two devices is carried out on a shared bus system, it is necessary to define the transmission system and access method. For this reason, the International Standardization Organization (ISO) has defined a 7 layer model. This section deals with this model.

General Information

Layers 1, 2 and 4 are required for a satisfactory and secure communication.

- Layer 1 defines the physical conditions, e.g. power and voltage levels.
- Layer 2 defines the access mechanism and the addressing of the communication stations.
- Layer 4 (transport layer) ensures the data security and consistency. In addition to controlling the transport, the transport layer is also responsible for data flow control, blocking and acknowledgment tasks.

ISO-OSI Reference Model

The layers defined in the ISO-OSI reference model control the behavior of the communication partner. The layers are arranged one on top of the other, with layer 7 being the top layer. Only layers that are alike communicate with each other.

How the individual layers are implemented is not defined by the reference model. The implementation depends on the actual case.

The individual layers are defined as follows:

Layer	Name	Description
7	Application Layer	Makes application-specific communication services available.
6	Presentation Layer	Converts the data from the standard format of the communication system to the device-specific format.
5	Session Layer	Is responsible for establishing, terminating and monitoring the communication connection.
4	Transport Layer	Is responsible for controlling the transport.
3	Network Layer	Is responsible for guiding the data from one address to another.
2	Data Link Layer	Is primarily responsible for detecting and correcting errors. Defines the bus access method.
1	Physical Layer	Defines the physical properties of the data transfer.

Physical Layer

This layer provides for a transparent transmission of the bits via the physical medium. In here, the electrical and mechanical properties as well as the transfer types are defined.

Data Link Layer

This layer ensures the transfer of bit strings between two systems. This includes the detection and correction of errors or the forwarding of transmission errors as well as the flow control. In local networks, the connection layer also grants exclusive access to the transfer medium.

- The connection layer is divided into two sub-layers (these sub-layers are also called layer 2a and layer 2b):ž XE "Data Link Layer" ž XE "Data Link Layer"
- Medium Access Control (MAC)
- Logic Link Control (LLC)

Network Layer

This layer guides the data between two end systems. End systems are the sender and recipient of a message, which might be routed through multiple transit systems. The network layer selects the route.

Transport Layer

The transport layer ensures that a reliable start to end connection is available. The services provided include establishing the transport, transferring the data and terminating the connection. Generally, the user of the service can specify a certain level of quality. Quality parameters include the transfer rate or the undetected error rate.

Session Layer

The primary task of the session layer is to synchronize communication relationships. The session layer services can break down a long transmission into smaller transmissions, i.e. set synchronization points. In case the transmission is terminated before completion, the entire transmission process must not be repeated again (the transmission will resume from a specified synchronization point).

Presentation Layer

Generally, different systems initially talk different languages at the onset of a data exchange. The presentation layer translates the different languages of the communication stations into a uniform language with an abstract syntax.

Application Layer

The application layer includes the application-specific services of the different communication applications. Since a multitude of applications exist, it is especially difficult to come up with uniform standards.

2.1.6 Connection of Bus Systems

To ensure a continuous flow of information between two different subnets, special connection elements are required. The following section briefly introduces the different types of connection elements.

General Information

Depending on the complexity of the connection or the differences in the subnets to be connected, distinctions can be made between repeaters, bridges, routers or gateways for the network connection.

Based on their tasks, these connection elements can be illustrated using the ISO reference model.

Repeater

The repeater copies the information received from the line to the other side and amplifies it. A repeater is transparent to all layers of the communicating stations, i.e. the physical layer of both networks must be identical.

Repeaters are not only used to connect similar subnets, but also to extend an existing subnet (e.g. bus system).

Bridge

A bridge is used to connect subnets that work with the same protocols on the data link layer (Logical Link Control LLC). The transfer media and the bus access methods (Medium Access Control MAC) of the connecting subnets can be different.

The bridge is primarily used to connect local networks of different topologies. Bridges are also used if special applications require the connection of certain structures to subnets.

Router

The router connects ISO networks that differ in layers 1 and 2.

The router also determines the best communication path for a message through an existing network (routing). Criteria for determining the best path can be path length or smallest transfer delay. In order to perform its tasks, the router changes the target and source addresses of the arriving data packages in the network layer before forwarding them. Since routers have to perform considerably more complex tasks than bridges, their processing rate is slower.

Gateway

A gateway connects networks with differing architectures, i.e. any two subnets can be connected. Referring to the ISO reference model, the task of the gateway is to translate the communication protocols of all layers. A gateway also makes the connection of an ISO network to a non-ISO network possible.

Network connections via gateways generally require considerably greater effort and offer slower speeds.

2.2 Communication Networks

This section of the manual contains information about the different industrial communication options. It explains the different communication types, their properties and components.

This section provides information about the following topics:

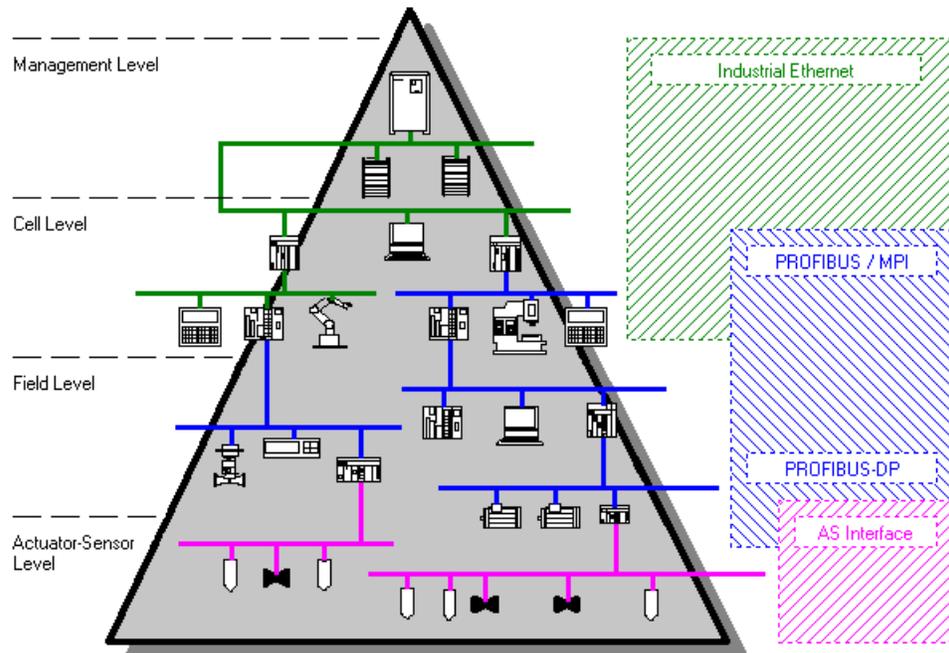
- Industrial Communication Overview
- Industrial Communication Subnets
- Industrial Communication via MPI
- Industrial Communication via PROFIBUS
- Industrial Communication via Ethernet
- OPC Interface Standard

2.2.1 Industrial Communication Overview

This section provides you with an industrial communication overview and classifies the different communication types according to their industrial environment.

General Information

Depending on the requirements, different communication networks are available for the industrial communication. The following graphic shows the individual automation levels and their suitable communication networks.



Management Level

At the management level, tasks effecting the entire operation are processed. This includes the archiving, processing, evaluating and reporting of process values and messages.

The operating data can also be collected and processed from multiple sites - from the management level, other sites can also be accessed. The number of stations in such a network can exceed 1000.

Ethernet is the dominant network type at the management level. To bridge greater distances, the TCP/IP protocol is used in most cases.

Cell Level

At the cell level, the automation tasks are processed. At this level, PLCs, operating and monitoring devices and PCs are connected with each other.

Depending on the performance demanded, the dominant network types are the Industrial Ethernet and PROFIBUS.

Field Level

The field level is the connecting link between the PLCs and the plant. The devices used at the field level supply the process values, messages, etc. and also forward commands to the plant.

In most cases, the amounts of data transferred at the field level are small.

PROFIBUS is the dominant network type at the field level. To communicate with the field devices, the DP protocol is frequently used.

Actuator-Sensor Level

At the actuator-sensor level, a master communicates with the actuators and sensors connected to its subnet. A characteristic of this level is the transfer of extremely small data amounts, but with a quick response time.

2.2.2 Industrial Communication Subnets

This section introduces you to the different subnets used in the industrial communication. Only those subnets relevant for a communication with WinCC will be described. An explanation of the AS-i (Actuator-Sensor Interface) will therefore be omitted.

General Information

Depending on the requirements, different subnets are available for the industrial communication. The following lists the different subnets by increasing order of performance:

- MPI
- PROFIBUS
- Industrial Ethernet

MPI

MPI (Multi Point Interface) is suitable for small size networks at the field and cell level. It can only be used in conjunction with the SIMATIC S7. The MPI subnet uses the MPI interface of the PLCs central processor card for the communication. This interface has been designed as a programming interface and quickly reaches its performance limits with increasing communication demands.

A PC can access an MPI subnet via an installed MPI card. Communication processors providing access to the PROFIBUS can also be used.

PROFIBUS

The PROFIBUS (Process Field Bus) is a subnet designed for the cell and field level. It is an open, manufacturer-independent communication system.

PROFIBUS is used to transfer small to medium sized data amounts between few communication partners.

With the DP (Decentralized Periphery) protocol, PROFIBUS is used for the communication to intelligent field devices. This type of communication is characterized by quick, cyclic data transfers.

Industrial Ethernet

The Industrial Ethernet is a subnet suitable for the management and cell level. It is used to transfer large amounts of data over long distances between a large number of stations.

The Industrial Ethernet is the most powerful subnet available for the industrial communication. It can be configured and expanded easily without great effort.

2.2.3 Industrial Communication via MPI

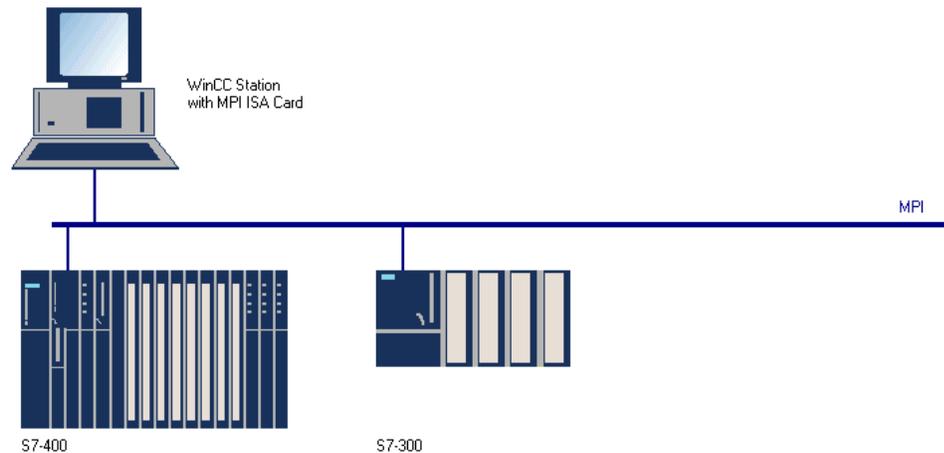
This section introduces you to the MPI subnet. It describes its properties, applications and components used.

General Information

The MPI subnet can be used at the cell and field level. The communication partners to be linked must be members of the SIMATIC S7 family.

The MPI subnet can economically link a small number of stations. However, the lower performance of an MPI solution must be acceptable. The maximum number of stations is limited to 32.

MPI uses the available MPI interface of the SIMATIC S7 family PLCs for the communication. This interface has been designed as a programming interface. The following graphic shows an example of an MPI network. The bus access of the individual communication stations is realized via the programming interface of the corresponding central processor card.



Access Method

MPI uses the token passing access method. The right to access the bus is forwarded from station to station. This access right is called a token. If a station receives the token, it has the right to send messages. If the station has no messages to send, it forwards the token directly to next station in the logical ring. Otherwise the token will be forwarded after a specified hold time.

Transfer Media

The same transfer method used for a PROFIBUS network can be used for an MPI network. It can be designed as an optical or electrical network. The transfer rate generally used is 187.5 kBit/s. However, the latest versions of the S7-400 can achieve transfer rates of up to 12 MBit/s.

2.2.4 Industrial Communication via PROFIBUS

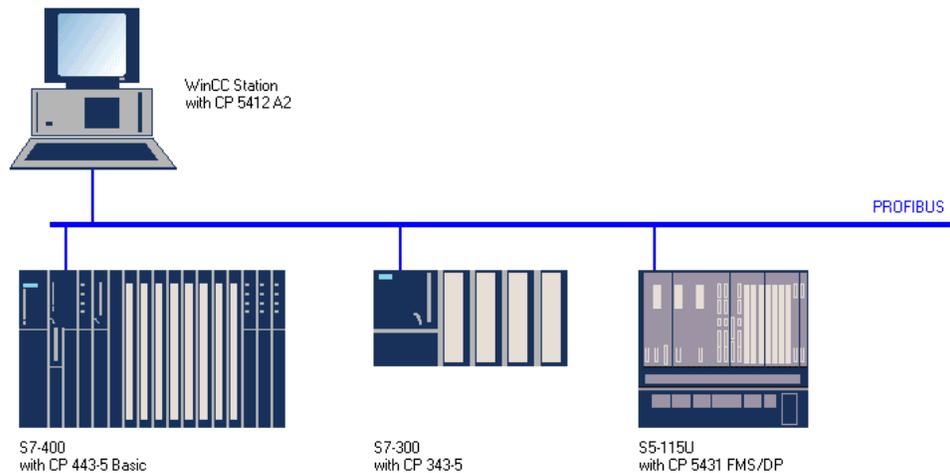
This section introduces you to the PROFIBUS subnet. It describes its properties, applications and components used.

General Information

The PROFIBUS is a subnet designed for the cell and field level with only a limited number of stations. The maximum number of stations is 127.

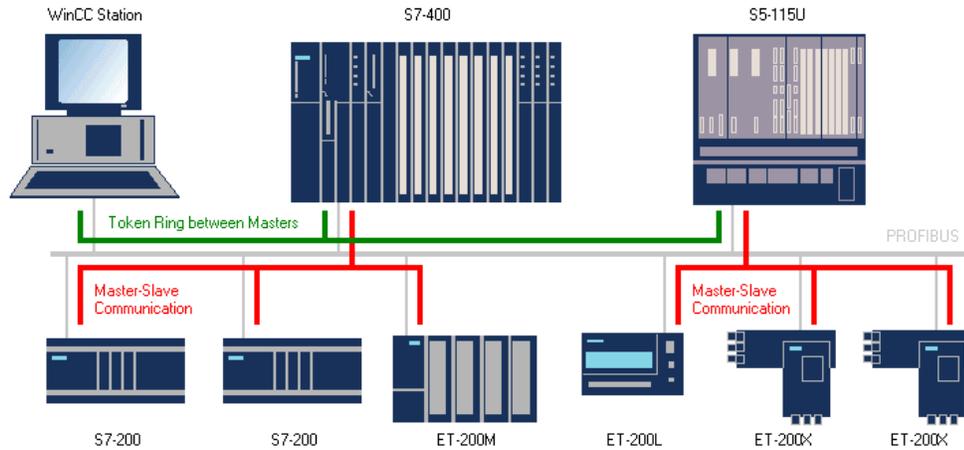
PROFIBUS is an open, manufacturer-independent communication system. It is based on the European Standard EN 50170, Volume 2, PROFIBUS. By meeting these requirements, PROFIBUS guarantees openness for the connection of third-party components conforming to this standard.

The following graphic shows a PROFIBUS network. It illustrates the components used by the individual communication partners to realize their bus access. The openness of the PROFIBUS concept of course permits the connection of third-party devices to the communication network.



2.2.4.1 The PROFIBUS Access Method

A PROFIBUS network distinguishes between active and passive network stations. Active stations use the token passing access method, whereas passive stations use the master slave access method. Because of this, the PROFIBUS access method is also called Token Passing with subordinated Master Slave.



All active stations, following a set sequence, form a logical ring. Each active station knows the other active stations and their sequence on the PROFIBUS. The sequence is independent of the physical station arrangement on the bus.

The bus access right is passed from active station to active station. This access right is called a token. If a station receives the token, it has the right to send messages. If the station has no messages to send, it forwards the token directly to next station in the logical ring. Otherwise the token will be forwarded after a specified hold time.

If an active station possessing subordinated passive stations receives the token, it will query its passive stations and send data to them. Passive stations themselves never receive the token.

2.2.4.2 PROFIBUS Protocol Architecture

- PROFIBUS protocols optimized for different applications are available. The following lists the three available protocol versions:
- PROFIBUS-FMS (Fieldbus Message Specification) is suitable for the communication of PLCs in smaller cell networks and for the communication to field devices with FMS interfaces. The powerful FMS services offer a broad application scope and great flexibility for the management of extensive communication tasks.
- PROFIBUS-DP (Decentralized Periphery) is suitable for the connection of decentralized peripheral devices, such as the ET 200, with very fast response times.
- PROFIBUS-PA (Process Automation) has been designed specifically for process engineering and is a communication-compatible expansion of PROFIBUS-DP. It allows the connection of field devices in areas potentially threatened by explosions.

All protocols use the same transfer technology and uniform bus access protocol - they can therefore all be operated on the same line.

- In addition to the above protocols, the following communication options are supported:
- Via FDL services (SEND/RECEIVE), the communication to any communication partner supporting FDL (Field Data Link) can be quickly and easily implemented.
- S7 functions allow the implementation of an optimized communication within the SIMATIC S7 family.

2.2.4.3 Transfer Media

A PROFIBUS network can be designed as an optical or electrical network. Mixed structures, consisting of optical and electrical PROFIBUS networks, can also be implemented.

Electrical Network

An electrical PROFIBUS network uses a shielded, twisted, two-wire cable as its transfer medium. The RS 485 interface operates with voltage differences. It is therefore less susceptible to interference than a voltage or current interface.

The various PROFIBUS stations are connected to the bus via a bus terminal or bus connector plug. A maximum of 32 stations can be connected to each segment. The individual segments are connected to each other via repeaters. The transfer rate can be set in steps from 9.6 kBit/s to 12 MBit/s. The maximum segment length is determined by the transfer rate.

The following table lists transfer rates and their respective maximum distances. The distances listed are either with or without the use of repeaters.

Transfer Rate	Size without Repeaters	Size with Repeaters
9.6 - 93.75 kBit/s	1000 m	10 km
187.5 kBit/s	800 m	8 km
500 kBit/s	400 m	4 km
1.5 MBit/s	200 m	2 km
3 - 12 MBit/s	100 m	1 km

Optical Networks

An optical PROFIBUS network uses fiber optic cables as its transfer medium. Optical networks are not susceptible to electromagnetic interference, suitable for long distances and use either plastic or glass fiber optic cables. The transfer rate can be set in steps from 9.6 kBit/s to 12 MBit/s. The maximum segment length is independent of the transfer rate, with the exception of redundant optical rings.

Two connection methods are available to build an optical PROFIBUS network:

- Optical Link Modules (OLM) using plastic or glass fiber optic cables. OLM allows the configuration of an optical network in a line, ring or star structure. End devices are connected directly to the OLM. Optical rings can use single fiber cables (cost-optimized) or dual fiber cables (increased network availability).
- Optical Link Plugs (OLP) allows passive bus stations to be connected to an optical single fiber cable ring. The OLP is inserted directly into the PROFIBUS interface of the bus station.

In an optical PROFIBUS network, the maximum distance for all transfer rates is greater than 100 km.

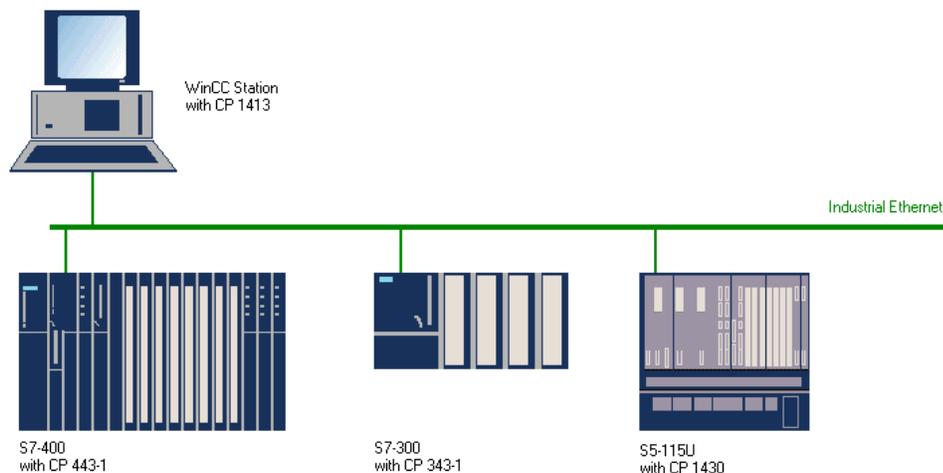
2.2.5 Industrial Communication via Ethernet

This section introduces you to the Industrial Ethernet subnet. It describes its properties, applications and components used.

General Information

The Industrial Ethernet is the most powerful subnet in the industrial environment. It is suitable for the management level as well as for the cell level. Industrial Ethernet makes the transfer of large data amounts over great distances between a large number of stations possible.

Industrial Ethernet is an open communication network following the IEEE 802.3 standard. It has been specifically designed to economically solve demanding communication tasks in the industrial environment. The major advantages of this subnet are its speed, simple expandability and openness as well as its high availability and global distribution. The configuration of an Industrial Ethernet subnet requires only small efforts.



Access Method

The Industrial Ethernet uses the CSMA/CD (Carrier Sense Multiple Access/Collision Detection) access method. Before sending a message, each communication station must check if the bus line is clear. If the bus line is clear, the station can send the message right away.

If two communication stations start sending messages at the same time, a collision will occur. This collision is detected by both stations and they will stop sending. After a randomly selected waiting period, they will attempt to send again.

2.2.5.1 Industrial Ethernet Protocol Profiles

- Communication via the Industrial Ethernet can be performed using one of the following protocol profiles:
 - MAP (Manufacturing Automation Protocol) uses the MMS services of the user interface.
 - The TF protocol contains the open SINEC AP automation protocol that has proven itself in many applications. Based on it, technological functions are available.
 - SEND/RECEIVE provides functions which allow a quick implementation of the communication between S5, S7 and PC.
 - S7 functions provide optimized communication within the SIMATIC S7 family.
- Communication profiles can be changed without changes to the application programs.

2.2.5.2 Transport Protocols

- For the communication via the Industrial Ethernet, the following transport protocols are available:
- The ISO transport provides services for the transfer of data using the connections. Active data can be blocked into multiple data messages.
- The ISO-on-TCP transport corresponds to the TCP/IP standard with the RFC 1006 expansion. This expansion is required, since TCP/IP uses the data stream communication which does not block data.
- UDP can only perform unsecured data transfers.

2.2.5.3 Transfer Media

An Industrial Ethernet network can be designed as an optical or electrical network. Mixed structures, consisting of optical and electrical networks, can also be implemented. This allows you to take advantage of the configuration options of both network types.

Electrical Network

An electrical Industrial Ethernet network can be designed using two different cabling types: The triaxial cable (AUI) or the Industrial Twisted Pair cable (ITP). To connect communication modules possessing only AUI interfaces to an ITP network, a twisted pair transceiver (TPTR) must be used.

Optical Network

An optical network can be designed as a line, ring or star structure. As the transfer medium, glass fiber optic cables are used exclusively.

2.2.6 OPC Interface Standard

This section introduces you to the new interface standard OPC and describes the applications of this concept.

General Information

OPC (OLE for Process Control) is a new communication standard for components in the automation field. This concept integrates office applications, HMI systems such as WinCC, PLCs and field devices.

OPC is defined as an open interface standard by the OPC Foundation, whose members include more than 120 companies from the automation sector. The current OPC specifications can be viewed via the Internet. The site also contains information about the individual OPC Foundation members and their product spectrum.

The Internet address of the OPC Foundation is: <http://www.opcfoundation.org>

Communication Concept

Minimum components of an OPC configuration always include an OPC server and an OPC client. The OPC server is an application that makes data available to an OPC client, who retrieves this data and further processes it.

WinCC and OPC

WinCC can communicate as an OPC client with any PLC, for which a corresponding OPC server is available. Among others, several SIMATIC NET OPC servers are available. WinCC also has an OPC server. This allows data exchanges with other applications possessing an OPC client interface (which also includes WinCC).

2.3 Communication Configuration

This section of the manual contains information about the configuration of a communication connection in a WinCC project. It will illustrate the general concept of the WinCC process communication, its configuration and diagnosis.

The following topics will be discussed:

- WinCC Process Communication
- WinCC Communication Configuration

2.3.1 WinCC Process Communication

Data Manager

The WinCC Data Manager manages the database. This data manager is not visible to the user. The data manager works with the data generated by the WinCC project and stored in the project database. It manages the WinCC tags during runtime. All applications of WinCC must request data from the data manager in the form of WinCC tags. These applications include Graphics Runtime, Alarm Logging Runtime and Tag Logging Runtime.

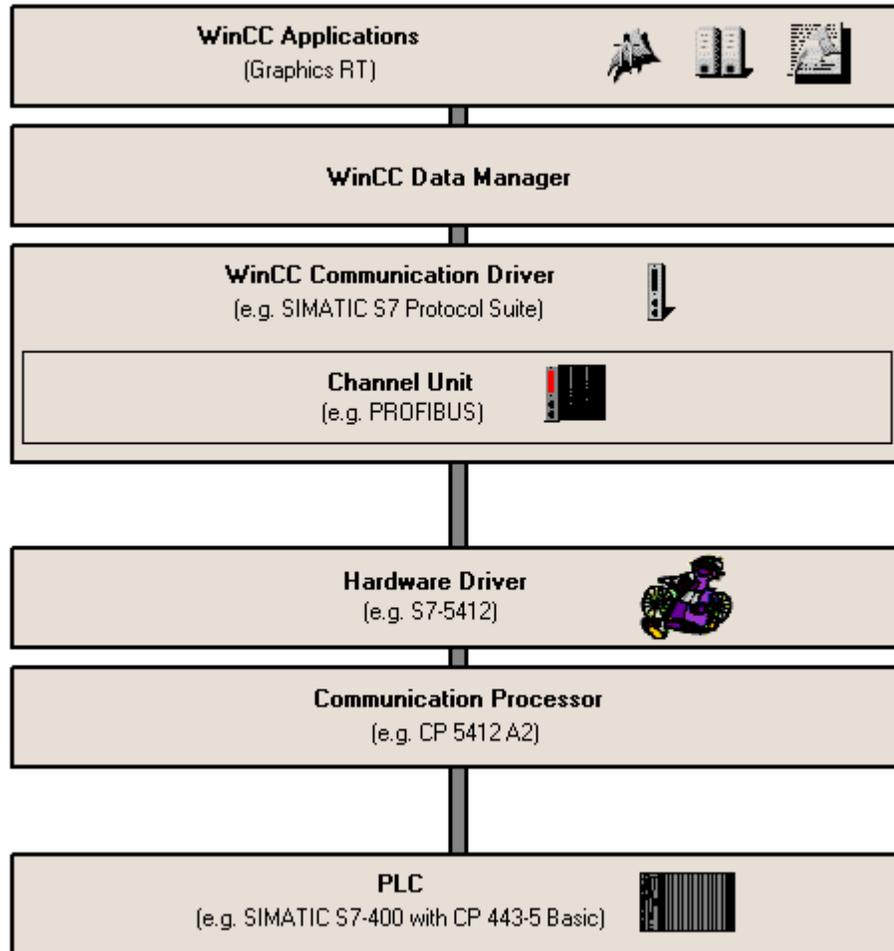
Communication Drivers

In order for WinCC to communicate with different types of PLCs, communication drivers are utilized. The WinCC communication drivers connect the data manager to the PLC. A communication driver consists of a C++ DLL, which communicates with the data manager's interface - the so-called Channel API. The communication driver supplies the WinCC tags with process values.

Communication Structure

The WinCC data manager manages the WinCC tags in runtime. The various WinCC applications request tag values from the data manager.

The data manager's task is to fetch the requested tag values from the process. It accomplishes this via the communication driver integrated into the WinCC project. The communication driver forms the interface between WinCC and the process using its channel units. In most cases, the hardware-based connection to the process is implemented using a communication processor. The WinCC communication driver uses the communication processor to send request messages to the PLC. The communication processor then sends the requested process values in corresponding answer messages back to WinCC.



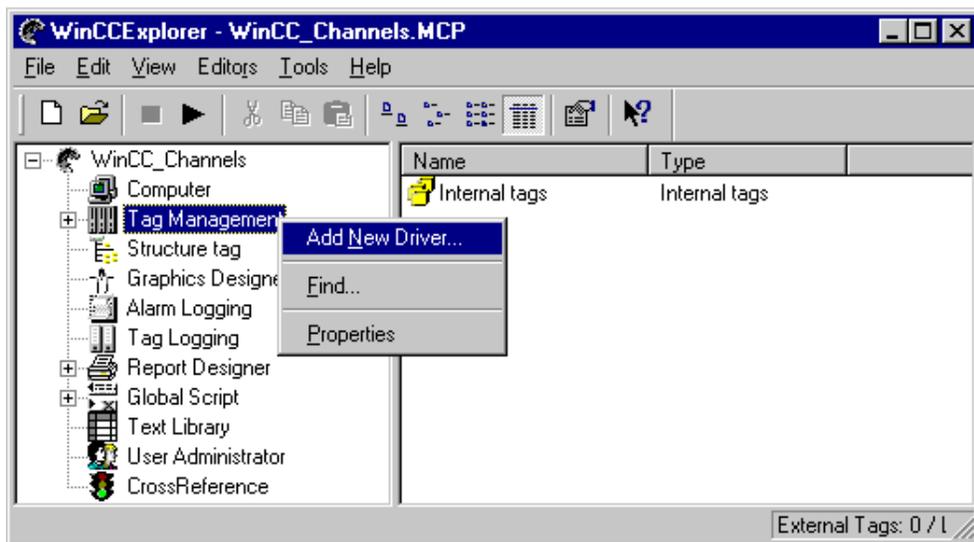
2.3.2 WinCC Communication Configuration

This section describes the configuration steps necessary in WinCC to establish a communication connection to a PLC.

Communication Drivers

The communication in WinCC is performed using various communication drivers. A number of communication drivers are available for the connection of different PLCs on different bus systems.

A communication driver is added to a WinCC project in the *WinCC Explorer*. In there, the communication driver is added to *Tag Management*. Generally, this is performed via a R on the *Tag Management* entry and the selection of *Add New Driver* from the pop-up menu. This action will display all communication drivers installed on the computer in a dialog box. Each communication driver can only be added once to a WinCC project.

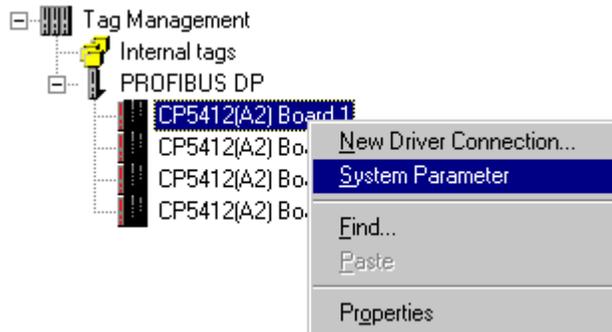


Communication drivers are files with the extension *.chn*. The communication drivers installed on the computer are located in the *Bin* subfolder of the WinCC installation folder. After the communication driver has been added to the WinCC project, it will be listed in the *WinCC Explorer*, as a sub-entry next to *Internal Tags*, under *Tag Management*.

Channel Unit

The communication driver entry in Tag Management contains at least one sub-entry. This is the communication driver's so-called channel unit. Each channel unit forms the interface to exactly one subordinated hardware driver and thus to the PC's communication module. The communication module addressed by the channel unit must be defined.

This communication module is assigned in the System Parameters dialog box. This dialog box is opened by  on the corresponding channel unit entry and selecting System Parameters from the pop-up menu



This dialog box's appearance depends on the selected communication driver. Generally, the module to be used by the channel unit is specified in here. However, it might also be necessary to specify additional communication parameters.

Connection

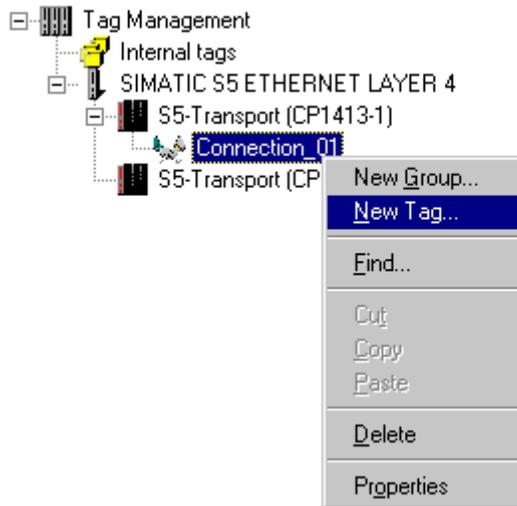
For a channel unit to read and write process values of a PLC, a connection must be set up to this PLC. A new connection is set up by  on the corresponding channel unit entry and selecting New Driver Connection from the pop-up menu.

The connection parameters to be set depend on the selected communication driver. The connection must always be assigned a name unique to the project. The additional parameters generally specify the communication partner to be reached.

WinCC Tags

To gain access to certain data in a PLC, WinCC tags must be configured. These are also referred to as external tags as opposed to internal tags, which do not possess process driver connections.

WinCC tags must be created for each configured connection. To create a new WinCC tag,  on the corresponding connection entry and select *New Tag* from the pop-up menu.



The Tag Properties dialog box will be opened, in which different properties of the tag are defined.

The tag must be assigned a name unique to the WinCC project.

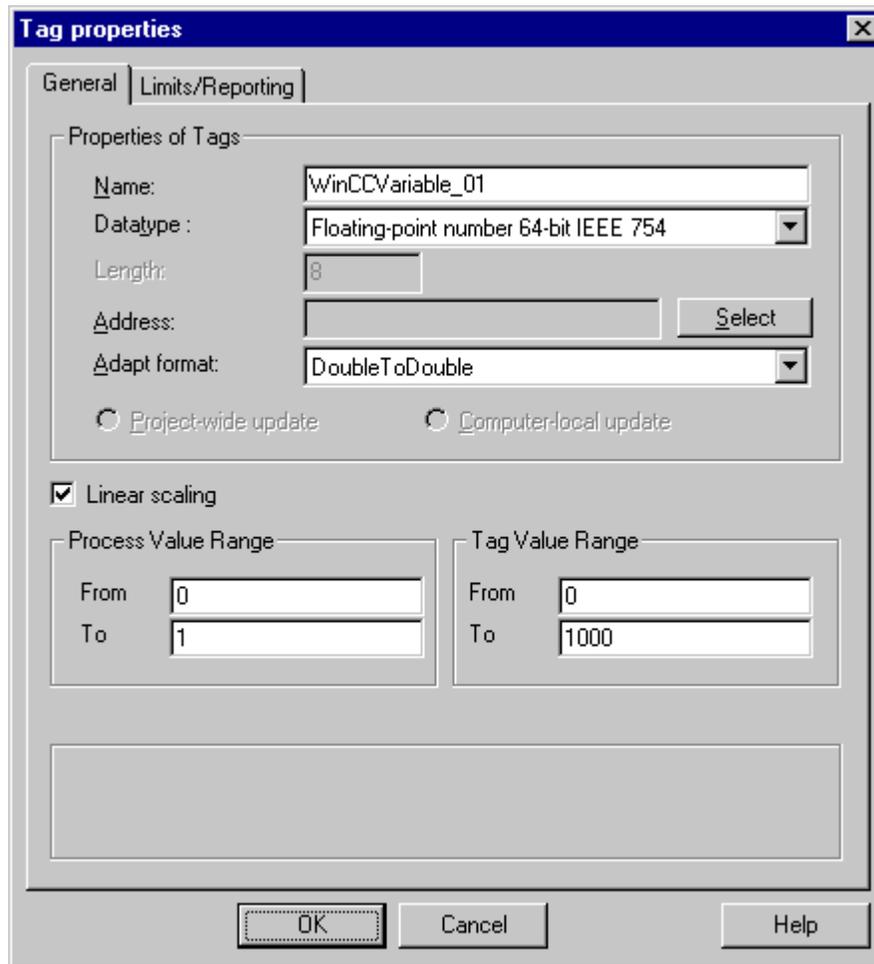
- Additionally, the data type of the tag must be specified. WinCC supports the following external tag data types:
 - Binary Tag
 - Unsigned 8-Bit Value
 - Signed 8-Bit Value
 - Unsigned 16-Bit Value
 - Signed 16-Bit Value
 - Unsigned 32-Bit Value
 - Signed 32-Bit Value
 - Floating-Point Number 32-Bit IEEE 754
 - Floating-Point Number 64-Bit IEEE 754
 - Text Tag 8-Bit Character Set
 - Text Tag 16-Bit Character Set
 - Raw Data Type

For numerical data types, with the exception of the *Binary Tag* data type, there is the option to perform a *Format Adaptation*. This means that the WinCC tag can refer to a data area in the PLC, which does not correspond to the data type of the WinCC tag.

By default, the Adapt Format option is not active. The WinCC tag is then assigned a data area in the PLC, which corresponds to the data type of the WinCC tag.

Also, for numerical data types, with the exception of the Binary Tag data type, a *Linear Scaling* can be performed. This means that the value range of a process value will be mapped linearly to a defined value range of a WinCC tag.

For example, the process can request an entered set-point value in the unit mbar, whereas the value is entered in WinCC by the user in the unit mbar. The easiest solution for such a problem is the application of a liner format adaptation, as illustrated in the following graphic.



Tags of the data type *Text Tag 8-Bit Character Set* and *Text Tag 16-Bit Character Set* require a length specification. This length is specified in characters. This means that a tag of the *Text Tag 16-Bit Character Set* data type, which is to receive 10 characters, must be configured with a length of 10.

WinCC tags must be allocated a data area in the communication partner. The tags must be addressed in a certain way in the communication partner. The addressing type depends on the type of communication partner. The dialog box for specifying a tag address can be opened via the Select button.

3 Selection Guide

This chapter helps you in selecting the right communication solution for your application. It outlines a general procedure for selecting the most suitable communication solution for a project based on the existing conditions and requirements.

The factors influencing the decision for a communication solution are complex. The stage (time) at which the decision is made, is of major importance. The best time to make a decision is at the planning stage of a plant. Should the plant already exist, the room available for making a decision is more restricted because of the existing conditions. Based on the experiences in a number of projects, the following can be said: The budget for a communication solution should always include reserves of approximately 20 percent. This comparably small additional cost can generally be recuperated in savings realized in the later configuration and expansion stages of the communication solution.

The main factors that enter into the decision making process for a certain communication solution include the amount of data to be transferred, the number of connected stations, the network size and its expandability.

This chapter is organized into the following sections:

- **Project Analysis** Exact specification of the communication demands made in your project.
- **Performance Data** Comparison of the performance of the different communication solutions and how they meet the required demands.

3.1 Project Analysis

This section provides you with a guideline, which helps you specify the project requirements made to the communication system.

To select the correct communication solution, it is important to know what requirements need to be met. This means that all necessary data has to be collected and analyzed. All following information cover the subject of determining project requirements made to the communication.

An important base for defining your own requirements is the Specifications framework. In addition, the specifications of already existing hardware and software must be considered. This chapter is organized into the following sections:

- **Specifications** Determine the amounts of data to be processed and the origin of the data.
- **Configuration Notes** Reduce the communication load and the increase the performance through supporting configurations.

3.1.1 Specifications

The specifications framework helps determine the amount of data that has to be processed by the communication system. This includes determining the amount of data required by the WinCC project. Also, it must be determined from which locations the WinCC project is collecting this data.

To assess the amount of data required by the WinCC project, simply add up of the amounts of data required by the individual applications per time unit. In detail, this means the determination of the requirements made to the communication system by:

- Graphics Runtime
- Alarm Logging Runtime
- Tag Logging Runtime
- Global Script Runtime (also of C actions in individual WinCC screens)
- Customer-specific applications

Requirements of Graphics Runtime

Graphics Runtime only requests the data from the data manager needed to update the values of the currently displayed WinCC screen. This means that the requirements made to the communication system can vary from screen to screen.

The screen with the highest communication volume should enter into your decision for a Graphics Runtime communication system.

Requirements of Alarm Logging Runtime

Alarm Logging Runtime request all tags to be monitored, in a cycle defined by the system, from the data manager. This can be event tags or tags for monitoring limit values.

If Alarm Logging has not been configured yet, the data for determining the number of messages can be deduced from measurement point and I/O lists.

Requirements of Tag Logging Runtime

Tag Logging Runtime requests all tags to be archived (using the specified update cycle) from the data manager. The display of these archived tag values - trend or table - has no effect on the requirements made to the communication system.

If Tag Logging has not been configured yet, the data for determining the number of tags (whose values are going to be archived) can be deduced from the customer requirements for archiving and reporting processes.

Requirements of Global Script Runtime

The requirements made by Global Script Runtime to the communication system depend on the type of tag request in the WinCC scripts and their execution cycle.

Additionally, consider that there are WinCC scripts executed globally and WinCC scripts executed locally (in various WinCC screens). The determination of the requirements made by Global Script Runtime to the communication system is in most cases only a rough estimate of the amount of data to be processed in the worst case.

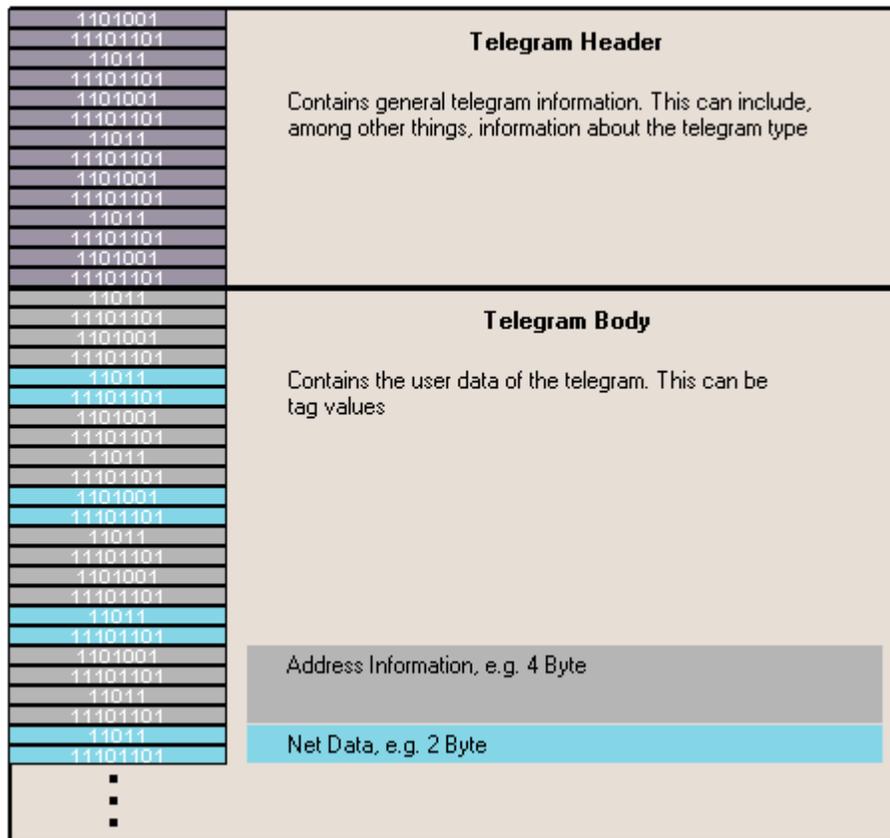
Communication Messages

- The communication between the individual communication partners is carried out by sending messages. This includes the communication between
- PLCs and HMI stations
- PLCs themselves
- PLCs and their peripheral devices

Instead of the term message, the term PDU (Protocol Data Unit) is also often used.

For example, a WinCC station requests data from a PLC by sending a request message. The PLC will then send the requested data to the WinCC station in a return message.

The following graphic shows the structure of a message:



Net Space Requirement of the WinCC Tags

In order to determine the data volume required by WinCC, it is necessary to have information regarding the space requirement of the individual WinCC tags. The space requirement varies greatly depending on the data type. The following table lists the space requirements of the various data types applicable to WinCC tags:

Pos.	Tag Type	Space Required
1	Binary Tag	1 Byte
2	Unsigned 8-Bit Value	1 Byte
3	Signed 8-Bit Value	1 Byte
4	Unsigned 16-Bit Value	2 Bytes
5	Signed 16-Bit Value	2 Bytes
6	Unsigned 32-Bit Value	4 Bytes
7	Signed 32-Bit Value	4 Bytes
8	Floating-Point Number 32-Bit IEEE 754	4 Bytes
9	Floating-Point Number 64-Bit IEEE 754	8 Bytes
10	Text Tag 8-Bit Character Set	1 Byte per Character
11	Text Tag 16-Bit Character Set	2 Bytes per Character
12	Raw Data Type	Set Length

Note that certain data types permit the configuration of a format adaptation. In this case, the net space requirement in the communication system corresponds to the space occupied by the WinCC tag's adapted format, not its original format.

Example of a format adaptation: A WinCC tag of the Unsigned 32-Bit Value data type is only mapped using 16 Bits in the memory area of the PLC. This is carried out using the DwordToUnsignedWord adapt format. The space requirement of the WinCC tag in the communication message will no longer correspond to the value of 4 Bytes listed in the table above, but to only 2 Bytes.

Formatanpassung :

Gross Space Requirement of the WinCC Tags

To transfer a tag in a message from one communication partner to another, not only the net data is relevant. In addition, address information for the assignment of the individual tags in the communication partner is required.

For example, this additional information requires 4 Bytes per tag for the SIMATIC S7 communication. This can significantly increase the space requirements for the individual tags. Among other things, the space requirement of a tag with a net space requirement of 1 Byte increases by a factor of five because of the additionally transferred information.

These numbers are specific to the communication with the SIMATIC S7, however, similar numbers can be expected using different communication systems.

Update Cycle

An update cycle must be specified for all WinCC tags, whether they are requested by Graphics Runtime, Alarm Logging Runtime or any other application. This update cycle greatly influences the requirements made by the WinCC project to the communication system. The update cycles should therefore be selected very carefully.

In connection with WinCC, the update cycles listed in the following table are generally used. Additionally, a maximum of 5 user-defined cycle times can be set. To determine the data traffic caused by WinCC, a table using the template below could be devised. In this table, the amounts of data required by the various applications are entered in Bytes.

Update Cycle	Graphics RT	Alarm Lg. RT	Tag Lg. RT	Global Sc. RT
Upon Change				
250 ms				
500 ms				
1 s				
2 s				
5 s				
10 s				
1 min				
5 min				
10 min				
1 h				
Used-Defined Cycle 1				
Used-Defined Cycle 2				
Used-Defined Cycle 3				
Used-Defined Cycle 4				
Used-Defined Cycle 5				

For the communication system, the update cycle Upon Change means that the tag is requested in a cycle of 250 ms.

Determining the Aggregate Data Volume

The amounts of data determined for the individual applications must now be added together. This results in the total communication load caused by WinCC.

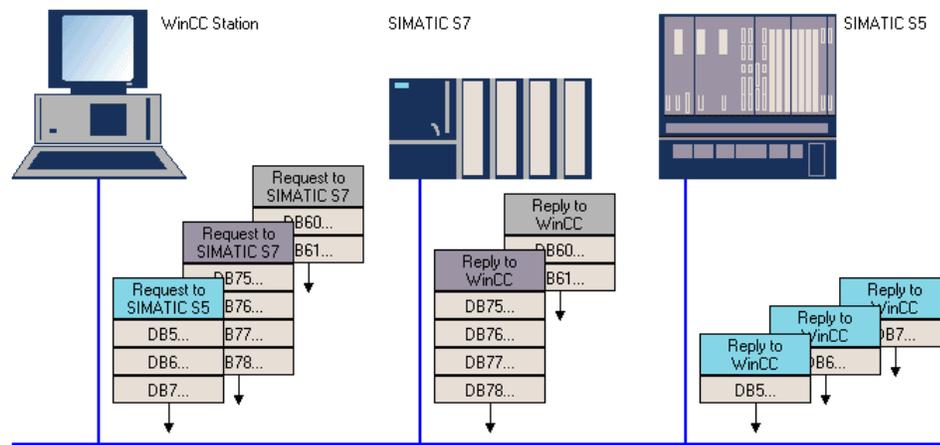
Note: The method used in this section to determine the requirements made by WinCC to the communication system are not exact to the Byte. Rather, it is an estimation of the data throughput that the communication system will encounter during plant operation.

Determining the Number of Messages

After the data volume to be processed per application and the time unit has been determined, an approximation of the number of messages needed could be made. This approximation also requires the maximum message length to be known, which is specific to a communication solution. The maximum message length varies greatly depending on the communication network and communication modules used.

However, the number of messages for different communication solutions can be determined. Even if these numbers are most likely not exact, they still help in making a decision.

Several factors must be considered in determining the approximate number of messages per time unit. One of these factors is the number of communication partners, from which data is requested. Also, the type and manner in which a communication partner answers requests must be considered. For example, the SIMATIC S5 uses one message for each data block, whereas the SIMATIC S7 is capable of grouping multiple data blocks into one message.



Limitations on the Number of Messages

In most cases, a PLC is connected to the communication system via a communication processor. Such a communication processor can only process a certain number of messages per time unit. This value generally lies between 15 and 20 message per second.

A defining property of a communication system is its transfer rate. This value allows the approximation of the number of messages of a certain length, which can be processed per time unit. The higher the transfer rate, the higher the maximum number of possible communication messages per time unit.

Considering additional Communication Stations

In addition to the requirements made by WinCC to the communication system, other factors impact the decision for a certain communication solution. This section explores these factors in greater detail.

The following points must be considered in making a decision for a certain communication solution:

- The communication between the individual PLCs.
- The communication between the PLCs and connected field devices.
- The communication of other WinCC stations.
- The communication of other connected stations (operator panels, tele-service stations and such).

Stated more simply, the number of additional stations participating in the communication must be considered.

3.1.2 Configuration Notes

The type of configuration has a major impact on the requirements made by WinCC to the communication system. The user can - by following some basic guidelines - configure a good performing and easily expandable communication system.

This section explains how some options can minimize the required load on the communication system already in the planning stage. The user should be aware what effects different configurations have on the communication system.

The advantages of a configuration that has been coordinated with the communication system include greater performance of the end product and greater flexibility with later expansions. This will result in greater customer satisfaction and reduced costs (for the current project and also for later expansions).

3.1.2.1 Data Update Cycle

Well selected update cycles in a configuration have a major impact on the performance of a communication system.

When determining the update cycles, always consider the system as a whole: From a technical point of view, determine what kind of values are being processed and how often new values should reasonably be requested from the PLC.

For example, the following could be considered in making a selection for an update cycle:

- Is it a major disadvantage for a plant operator, if he is notified about a 1° Celsius change in temperature 10 seconds after the actual event occurs instead of after 250 milliseconds?
- Is it necessary to archive to temperature progress of a tank measuring 5000 liters in 500 millisecond cycles?

By selecting update cycles matched to the process, the quantity of data can be significantly reduced in most cases.

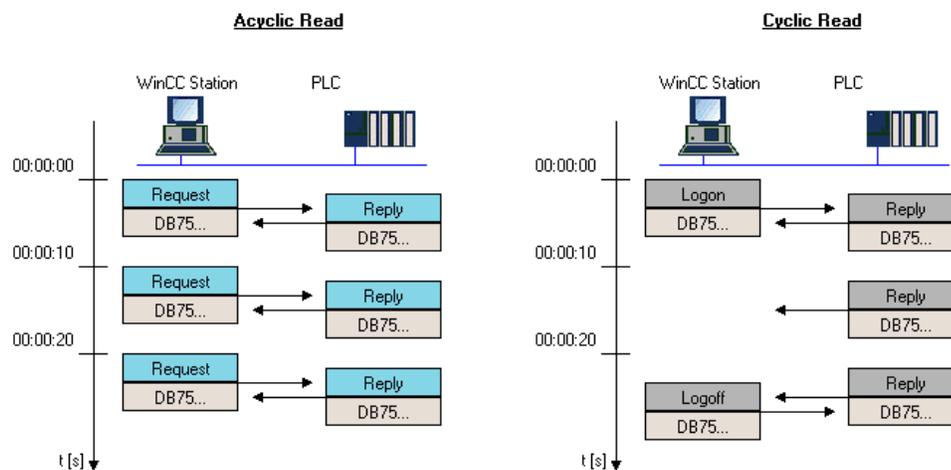
Request messages group tags that have the same update cycle, i.e. if a lot of different update cycles are used, the total number of messages increases, which negatively impacts the performance of the communication system.

3.1.2.2 Data Update Type

The WinCC data manager's task is to supply the various applications of the WinCC project with process data. For this purpose, the data manager must update its database at the requested cycles. How the data manager updates the database influences the load on the communication system.

Active WinCC Station

There are several possibilities of how the required data can be updated. If the WinCC station is appearing as an active partner, the update can be performed by so-called acyclic or cyclic read services.

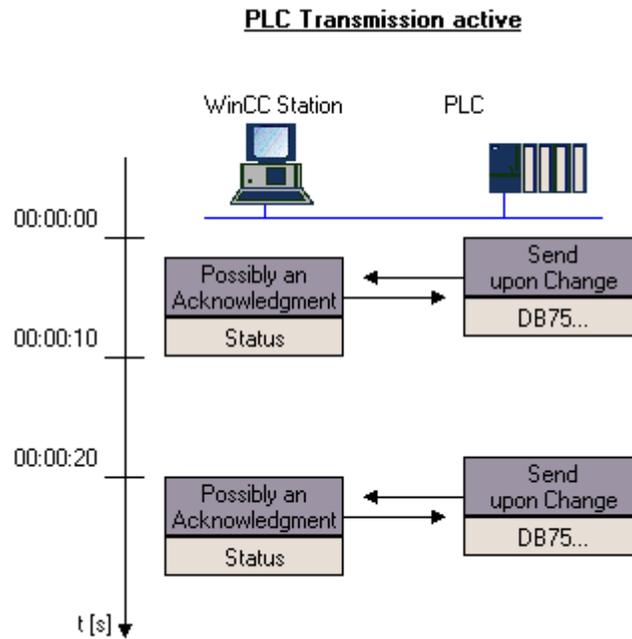


The acyclic read service requires two communication messages for each update. The WinCC station sends a request to the PLC, which then processes it using a reply message (or multiple reply messages).

If the cyclic read services is used, the WinCC station registers a read request at the PLC, which then processes it during the corresponding cycle. If the data is no longer needed or its content changes, the WinCC station will cancel the corresponding request.

Active PLC

In this data update variation, the PLC is actively sending data to WinCC, if it recognizes a change of data. This reduces the data traffic to a minimum, however, the configuration of the PLC becomes more complex.



Configuration Guideline

In general, a meaningful combination of both data update types described is the most cost-effective configuration approach.

3.1.2.3 Data Organization

The organization of the data present in the PLC can have a significant influence on the communication load. This depends greatly on the type of PLC used.

Grouping Data Areas

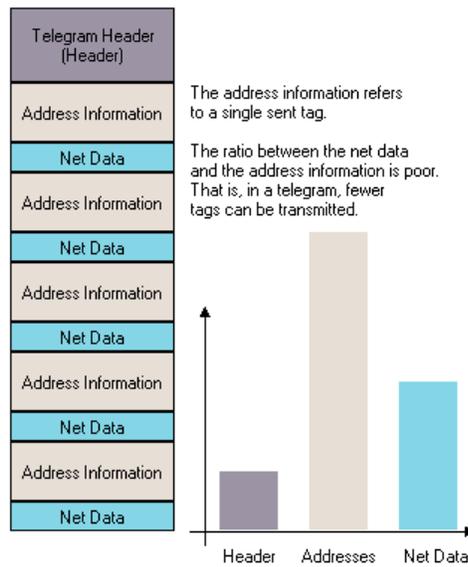
SIMATIC S5 systems group the requested data into datablocks. The larger the degree of scattering of the required data in the PLC, the larger the number of required messages. It is recommended to have the communication-relevant data contained in 3 to 5 datablocks in the PLC. If a scattering of data cannot be avoided, it can still be advantageous to locate the scattered data in the same data area. However, weigh the advantages attained for the communication system with the disadvantages created for the PLC.

Optimizing Messages

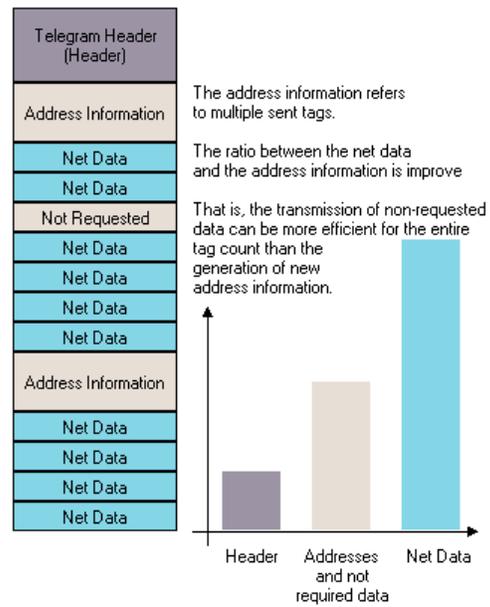
SIMATIC S7 systems are capable of packaging even scattered data into one communication message. However, grouping communication-relevant data into a few datablocks can still provide advantages.

The PLC is capable of performing an optimization of the message structure. This results in the transfer of more requested user data via one message. In general, for each requested tag, its address information must be transferred in addition to its process value (net data). If tags are located in neighboring or adjacent data areas, the required address information can be reduced.

Telegram without Optimization



Telegram with Optimization



3.2 Performance Data

This section contains detailed information about the performance of the various communication systems and their strengths and weaknesses.

The first part compares the differing communication systems. This is followed by detailed performance data about the individual communication systems and information about the communication options WinCC has for using this system.

This allows the user to select the best communication solution for the requirements determined in the previous chapter. The decision for a communication solution involves the selection of an available communication system and the hardware to be used.

3.2.1 Comparison of Communication Systems

To make a decision for a certain communication system, you need to know the performance capabilities of the available communication options. The following makes a comparison of the individual communication systems.

The criteria used to measure the performance of the individual communication systems are, among others:

- the transfer rate
- the number of stations
- the message length
- the network dimension
- the possible communication partners
- the costs

Communication Data

The following table provides an overview of the performance features of the individual communication systems.

	Serial	MPI	PROFIBUS	Ind. Ethernet
Application Area	Field Level	Field Level, Cell Level	Field Level, Cell Level	Cell Level, Management Level
Transfer Rate	9.6 kBit/s to 256 kBit/s	187.5 kBit/s to 12 MBit/s	9.6 kBit/s to 12 MBit/s	10 MBit/s (100 MBit/s)
Typical Number of Stations	2	2-10	5-20	5-100
Max. Number of Stations	2 (PtP)	32	127	1024
Typical Message Length	60 Byte	60 Byte	120 Byte	240 Byte
Max. Message Length	128 Byte	240 Byte	240 Byte	512 Byte to 4096 Byte
Network Dimension	50m	50m to 100m	10km to 90km	1km to global

WinCC Communication Data

The maximum number of stations listed in the previous table refers to the communication system in general. The actual number of PLCs with which WinCC can communicate does not only depend on the communication system itself, but also on the communication driver and communication cards used, the type of PLC and such.

The following table lists the maximum number of communication stations for some sample communication models. Detailed information about this topic can be found in the following descriptions about the respective WinCC communication drivers.

	Communication Driver	PLC	Number
Serial	S5 SERIAL 3964R	SIMATIC S5	2
Serial	S5 P.P. AS511	SIMATIC S5	4
MPI	S7 MPI	SIMATIC S7	29
PROFIBUS	S5 PROFIBUS FDL	SIMATIC S5	24
PROFIBUS	PROFIBUS FMS	SIMATIC S5, SIMATIC S7	32
PROFIBUS	S7 PROFIBUS	SIMATIC S7-400	32
PROFIBUS	S7 PROFIBUS	SIMATIC S7-300	118
PROFIBUS	PROFIBUS DP	PROFIBUS DP Slaves	126
Ind. Ethernet	S5 Ethernet TF	SIMATIC S5	30
Ind. Ethernet	S5 Ethernet Layer 4	SIMATIC S5	60
Ind. Ethernet	S7 Industrial Ethernet	SIMATIC S7	60

Comments:

- The correct name of the entries in the communication driver column starting with S5 includes the prefix SIMATIC.
- The entries in the communication driver column starting with S7 correspond to the various channel units of the SIMATIC S7 PROTOCOL SUITE.

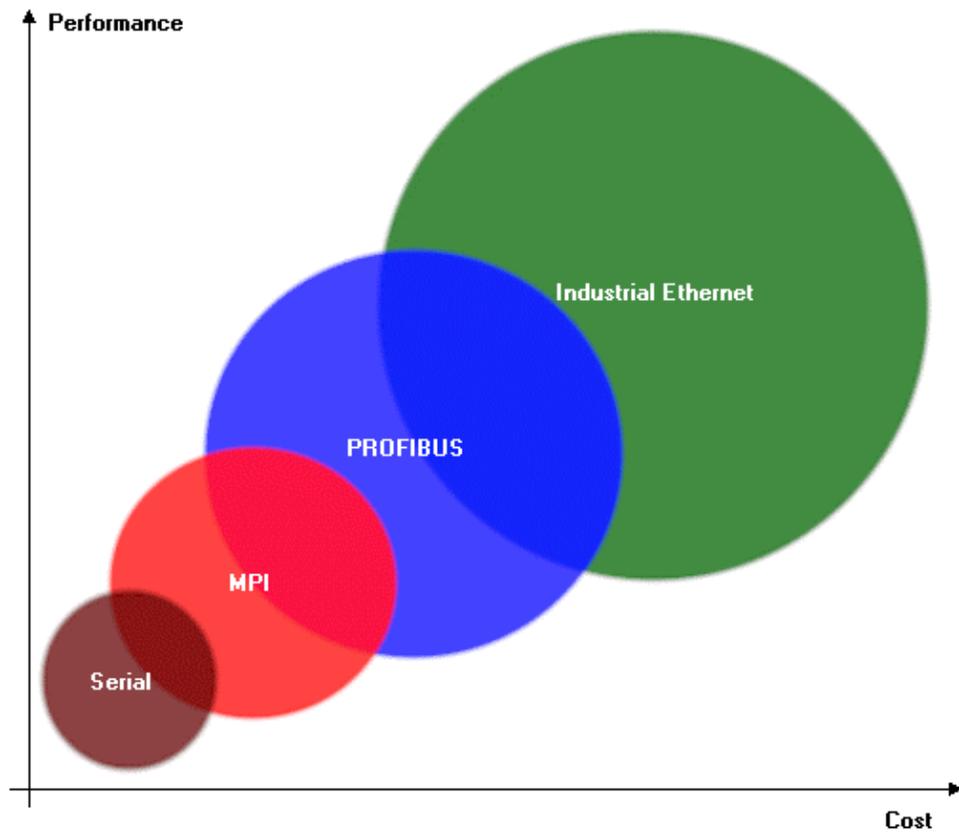
Communication Partner

The following table shows which PLCs can be addressed by WinCC using what communication system.

PLC	Serial	MPI	PROFIBUS	Ind. Ethernet
S5-90U, S5-95U, S5-100U	✓		✓	
S5-115U, S5-135U, S5-155U	✓		✓	✓
S7-200			✓	
S7-300		✓	✓	✓
S7-400		✓	✓	✓

Cost Factor

The greater the demands on a communication system, the higher the cost of implementation. The following diagram illustrates the relationship of a communication system to its expected implementation cost.



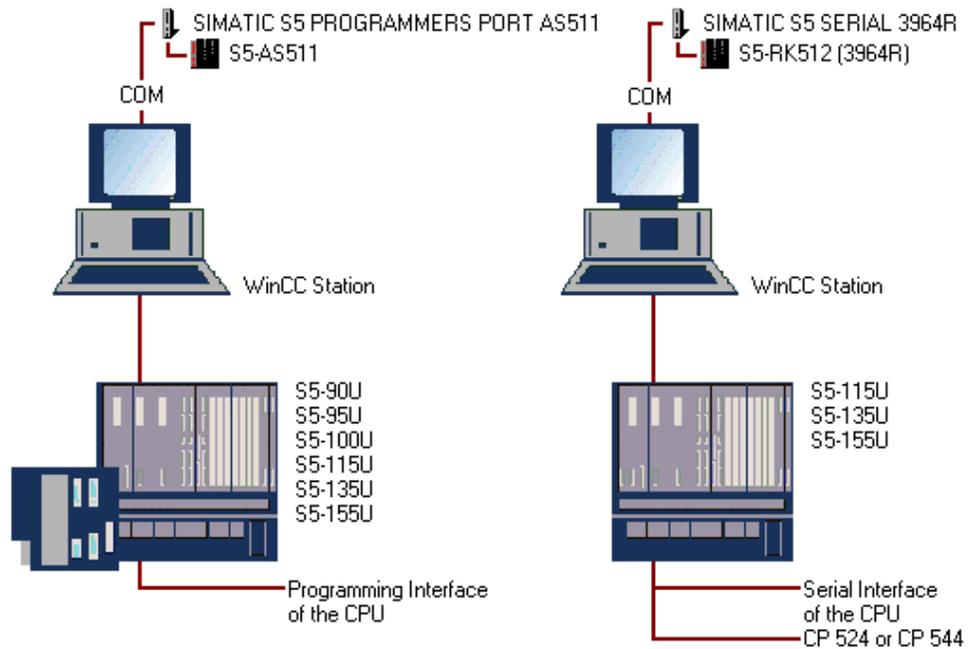
3.2.2 Serial Communication

The most cost-effective option to communicate from WinCC to a PLC is the formation of a serial communication connection. The COM interface of a WinCC station is used for the communication module.

Communication Partner

- Two WinCC communication drivers are available for establishing a serial communication connection to the SIMATIC S5 product line.
- SIMATIC S5 PROGRAMMERS PORT AS511 Communication via the programming interface of the respective CPU.
- SIMATIC S5 SERIAL 3964R Communication via the serial interfaces

The following graphic illustrates the PLCs that can be addressed with each communication driver.



Communication Data

Each COM interface of the WinCC station can address a maximum of one PLC. The total number of addressable PLCs can be looked up in the following table. The enclosed values represent the recommended maximum number.

Communication Driver	PLC
SIMATIC S5 PROGRAMMERS PORT AS511	4 (2)
SIMATIC S5 SERIAL 3964R	2

3.2.3 Communication via MPI

3.2.3.1 Communication Processors

To implement a communication connection to an MPI network, the WinCC station must be equipped with a suitable communication processor.

The same communication processors can be used that are used for the connection to the PROFIBUS network. Additional information can be found in the Communication Processors section of the PROFIBUS description.

The following table contains the communication processors available for the connection of a WinCC station to the MPI network and the driver software to be used.

Communication Processor	Configuration/Type	Driver Software
CP 5412 A2	ISA Card/Hardnet	PB S7-5412
CP 5613	PCI Card/Hardnet	PB S7-5613
CP 5511	PCMCIA Card/Softnet	S7-DOS
CP 5611	PCI Card/Softnet	S7-DOS

Comment:

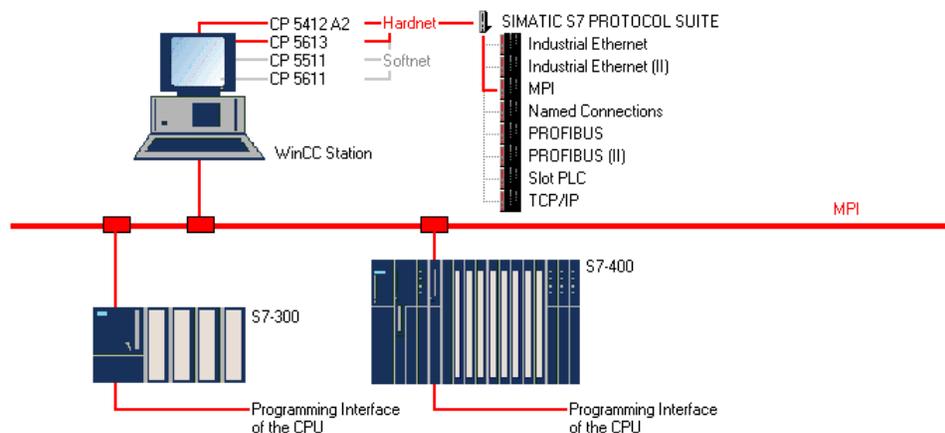
- The S7-DOS software is included in the delivery scope of WinCC.

3.2.3.2 SIMATIC S7 PROTOCOL SUITE

The SIMATIC S7 PROTOCOL SUITE communication driver provides the communication to SIMATIC S7-300 and S7-400 PLCs using various channel units. Among others, an MPI channel unit for the communication via MPI is available.

Communication Partner

The communication driver SIMATIC S7 PROTOCOL SUITE enables the communication with SIMATIC S7-300 and S7-400 PLCs. The following graphic illustrates the possible communication partners of the MPI channel unit.



Communication Data

The MPI channel unit supports the communication via Hardnet and Softnet modules. Per PC, only one module for the MPI communication can be used.

The following table lists the number of PLCs that can be addressed by each communication processor used in the WinCC station. The enclosed values represent the recommended maximum number.

Communication Processor	Communication Connections
Hardnet Module	29 (6)
Softnet Module	29 (4)

3.2.4 Communication via PROFIBUS

3.2.4.1 Communication Processors

To implement a communication connection to a PROFIBUS network, the WinCC station must be equipped with a suitable communication processor. In addition, suitable driver software must be installed for the desired communication protocol.

Two types of communication processors are available for WinCC. These are communication processors for the so-called Hardnet and Softnet. The main difference between these two is that Hardnet modules have their own microprocessors to reduce the loads on the computer's CPU and Softnet modules do not. XE "PROFIBUS"

Hardnet

- The entire protocol software runs on the module.
- The simultaneous operation of two protocols is possible (multi-protocol operation).
- Module is more powerful than a Softnet module.

Softnet

- The entire protocol software runs on the computer's CPU.
- Only one protocol can be operated at a time (single-protocol operation).
- Module costs less than a Hardnet module.

The following table lists the available communication processors for the system connection to a WinCC station.

Communication Processor	Configuration	Type
CP 5412 A2	ISA Card	Hardnet
CP 5613	PCI Card	Hardnet
CP 5511	PCMCIA Card	Softnet
CP 5611	PCI Card	Softnet

3.2.4.2 Communication Drivers

In WinCC, several communication drivers for the communication via PROFIBUS are available.

Communication Protocols

The communication drivers available for the PROFIBUS realize the communication via certain communication protocols.

The following table shows the assignment of the communication protocols to the communication drivers.

Communication Driver	Protocol
SIMATIC S7 PROTOCOL SUITE (PROFIBUS)	S7-Functions
SIMATIC S5 PMC PROFIBUS	S5-PMC
SIMATIC S5 PROFIBUS FDL	FDL
PROFIBUS FMS	FMS
PROFIBUS DP	DP

Communication Connections

The following table lists the number of communication connections that can be established by a certain communication driver. These values always relate to a channel unit, i.e. to a communication processor used in a WinCC station.

Communication Driver	Connections per Channel Unit	Channel Units
S7 PROFIBUS	59	2
S5 PMC PROFIBUS	24	1
S5 PROFIBUS FDL	24	1
PROFIBUS FMS	32	1
PROFIBUS DP	62	4

Comments:

- The number of available channel units listed in the table equals the number of communication processors that can be operated by the communication driver in the WinCC station.
- The maximum number of stations in a PROFIBUS network is 127 (with repeaters).

PLCs

In general, PLCs can be connected to a PROFIBUS network in two ways. Either integrated interfaces on the central module or special communication modules can be used.

The following table provides an overview of the communication options offered by the various WinCC communication drivers for the PROFIBUS network. For a better understanding of the table, note the legend at the end.

System	Module	PB DP	PB FMS	S5 FDL	S7 PB
S5-90U, S5-95U, S5-100U	CPU 95U	✓		✓	
	CP 541	✓			
S5-115U, S5-135U, S5-155U	CP 5431		✓	✓	
	IM 308-C	✓			
S7-200	CPU 215	✓			
	CPU 242-8	✓			
S7-300	CPU 315-2 DP	✓			✓
	CP 342-5	✓			✓
	CP 343-5		✓		✓
S7-400	CPU 41x-2 DP				✓
	CP 443-5 Ext.				✓
	CP 443-5 Basic		✓		✓
	IM 467				✓
DP Slaves	e.g. ET 200	✓			

Legend:

- **PB DP:** PROFIBUS DP
- **PB FMS:** PROFIBUS FMS
- **S5 FDL:** SIMATIC S5 PROFIBUS FDL
Comment: SIMATIC S5 PMC PROFIBUS only via CP 5431
- **S7 PB:** SIMATIC S7 PROTOCOL SUITE (PROFIBUS Channel Unit)

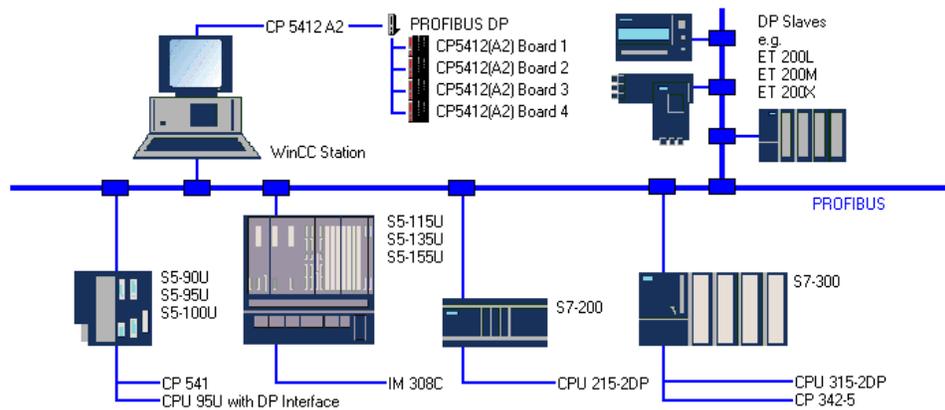
3.2.4.3 PROFIBUS DP

Via the communication driver PROFIBUS DP, a WinCC station can communicate with all PLCs and field devices that can be operated as DP slaves.

The application of the communication driver PROFIBUS DP in WinCC makes sense, if there is communication to a large number of subordinated devices with only a small data volume. A very fast tag update can be achieved despite scattered data. The communication runs via the cyclic data exchange of PROFIBUS DP, where the WinCC Station acts as the DP Master.

Communication Partner

Via the communication driver PROFIBUS DP, communication can be established to all PLCs and field devices that can be operated as DP slaves. The following graphic illustrates the possible communication partners.



Communication Data

The communication driver PROFIBUS DP exclusively supports the communication via the communication processor CP 5412 A2. In a WinCC station, up to four of these modules can be used. However, the system resources available to the PC acting as the WinCC station might result in limitations.

Each CP 5412 A2 communication processor can communicate with up to 62 DP slaves. This number only applies if a repeater are used. Otherwise, the general limitation of 32 connected stations to a PROFIBUS network without a repeater applies.

Required Software

The following table lists the software components that are required for the implementation of the communication connection of a WinCC station via PROFIBUS DP.

Name	Description
WinCC	Communication Driver PROFIBUS DP
SIMATIC NET	Driver Software PB DP-5412 Configuration Software COM PROFIBUS

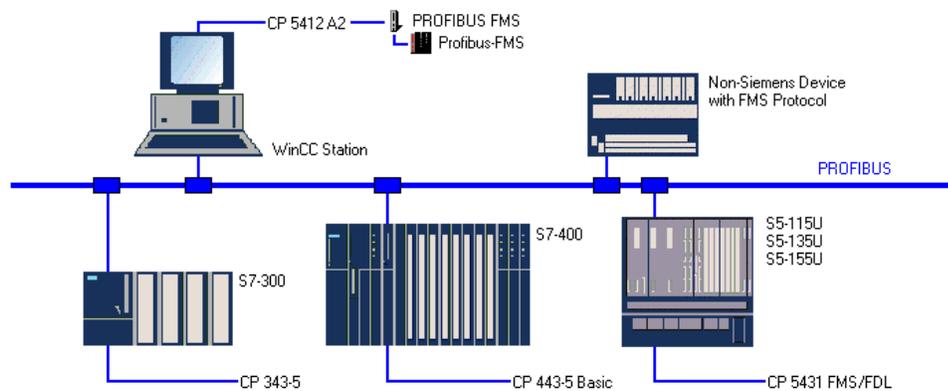
3.2.4.4 PROFIBUS FMS

Via the communication driver PROFIBUS FMS, a WinCC station can communicate with PLCs that support the FMS protocol.

The communication driver PROFIBUS FMS can be used to communicate with devices from different manufacturers. This type of communication can manage a high quantity of data.

Communication Partner

Via the communication driver PROFIBUS FMS, WinCC can communicate with all PLCs supporting the FMS protocol. The following graphic illustrates the possible communication partners.



Communication Data

The communication driver PROFIBUS FMS exclusively supports the communication via the communication processor CP 5412 A2. Per computer, only one module can be used for the FMS communication.

In theory, it is possible to communicate with up to 32 FMS devices. However, due to performance reasons you should not communicate with more than 12 FMS devices.

Communication Software

The following table lists the software components that are required for the implementation of the communication connection of a WinCC station via PROFIBUS FMS.

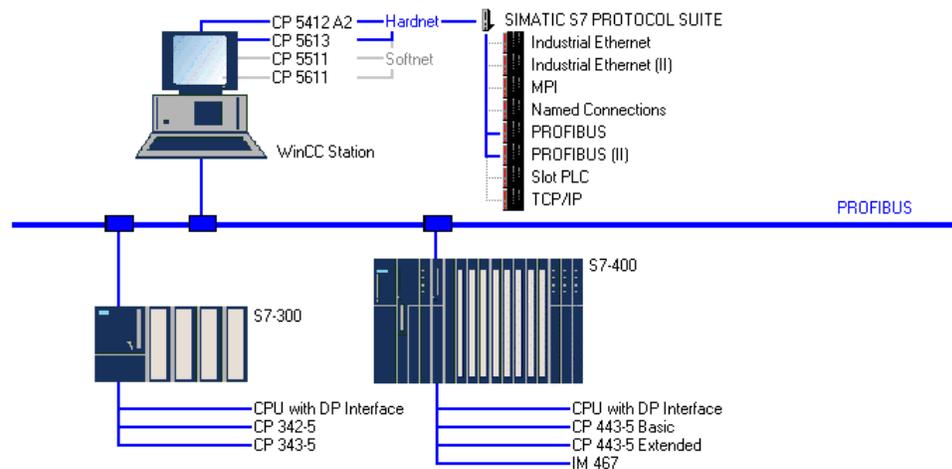
Name	Description
WinCC	Communication Driver PROFIBUS FMS
SIMATIC NET	Driver Software PB FMS-5412 Configuration Software COM PROFIBUS

3.2.4.5 SIMATIC S7 PROTOCOL SUITE

The SIMATIC S7 PROTOCOL SUITE communication driver provides the communication to SIMATIC S7-300 and S7-400 PLCs using various channel units. Among others, two PROFIBUS channel units are available for the communication via the PROFIBUS.

Communication Partner

The communication driver SIMATIC S7 PROTOCOL SUITE enables the communication with SIMATIC S7-300 and S7-400 PLCs. The following graphic illustrates the possible communication partners of the PROFIBUS channel units.



Communication Data

The PROFIBUS channel units support the communication via Hardnet and Softnet modules. A WinCC station can use up to two of these modules. The following table lists the driver software required for the various communication processors.

Communication Processor	Driver Software
CP 5412 A2	PB S7-5412
CP 5613	PB S7-5613
CP 5511	S7-DOS (included in the WinCC delivery scope)
CP 5611	S7-DOS (included in the WinCC delivery scope)

The following table lists the number of PLCs that can be addressed by each communication processor used in the WinCC station. The values depend on the PLC type. The enclosed values represent the recommended maximum number.

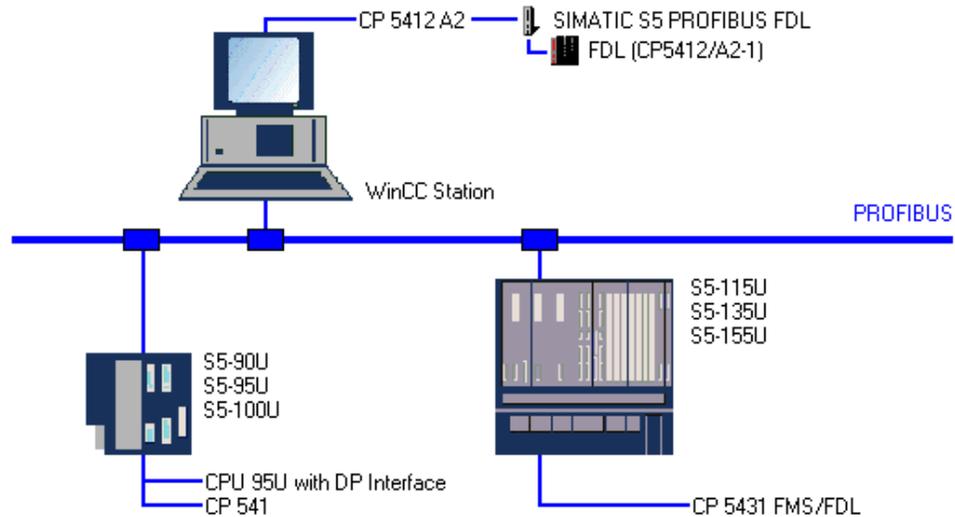
Communication Processor	SIMATIC S7-300	SIMATIC S7-400
Hardnet Module	59 (12)	16 (12)
Softnet Module	59 (6)	16 (6)

3.2.4.6 SIMATIC S5 PROFIBUS FDL

The communication driver SIMATIC S5 PROFIBUS FDL supports the communication to the SIMATIC S5 PLCs via the SEND/RECEIVE interface (FDL).

Communication Partner

The following graphic illustrates the possible communication partners.



Communication Data

The communication driver SIMATIC S5 PROFIBUS FDL exclusively supports the communication via the communication processor CP 5412 A2. Per computer, only one module can be used for the FDL communication.

In theory, it is possible to communicate with up to 24 SIMATIC S5 stations. However, due to performance reasons you should not communicate with more than 12 SIMATIC S5 stations.

Communication Software

The following table lists the software components that are required for the implementation of the communication connection of a WinCC station to SIMATIC S5 via FDL.

Name	Description
WinCC	Communication Driver SIMATIC S5 PROFIBUS FDL
SIMATIC NET	Driver Software PB DP-5412, PB S7-5412 or PB FMS-5412
STEP5	Standard Communication Blocks

3.2.5 Communication via Industrial Ethernet

3.2.5.1 Communication Processors

To implement a communication connection to the Industrial Ethernet, the WinCC station must be equipped with a suitable communication processor. In addition, suitable driver software and a communication protocol must be installed for the desired communication. Two types of communication processors are available for WinCC. These are communication processors for the so-called Hardnet and Softnet. The main difference between these two is that Hardnet modules have their own microprocessors to reduce the loads on the computer's CPU and Softnet modules do not.

Hardnet

- The entire protocol software runs on the module.
- The simultaneous operation of two protocols is possible (multi-protocol operation).
- Module is more powerful than a Softnet module.

Softnet

- The entire protocol software runs on the computer's CPU.
- Only one protocol can be operated at a time (single-protocol operation).
- Module costs less than a Hardnet module.

The following table lists the available communication processors for the system connection to a WinCC station.

Communication Processor	Configuration	Type
CP 1413	ISA Card	Hardnet
CP 1613	PCI Card	Hardnet
CP 1411	ISA Card	Softnet
CP 1511	PCMCIA Card	Softnet

3.2.5.2 Communication Drivers

In WinCC, several communication drivers for the communication via Industrial Ethernet are available.

Communication Protocols

The communication drivers available for Industrial Ethernet implement the communication to a certain PLC via a certain communication protocol.

The following table displays the assignment of communication drivers and their transport/communication protocols.

Communication Driver	Transport/Communication
SIMATIC S5 ETHERNET LAYER 4	ISO with SEND/RECEIVE
SIMATIC S5 PMC ETHERNET	ISO with SEND/RECEIVE
SIMATIC S5 ETHERNET TF	ISO with TF
SIMATIC S7 PROTOCOL SUITE (Industrial Ethernet)	ISO with S7-Functions
SIMATIC S7 PROTOCOL SUITE (TCP/IP)	ISO-on-TCP with S7-Functions

PLCs

The connection of a PLC to the Industrial Ethernet is realized via special communication modules.

The following table gives an overview of the communication options offered by the various WinCC communication drivers for the Industrial Ethernet. For a better understanding of the table, note the legend at the end.

System	Module	S5 SR	S5 TF	S7 ISO	S7 TCP
S5-115U, S5-135U, S5-155U	CP 1430	✓	✓		
S7-300	CP 343-1			✓	
	CP 343-1 TCP				✓
S7-400	CP 443-1			✓	
	CP 443-1 TCP				✓
	CP 443-1 IT				✓

Legend:

- **S5 SR:** SIMATIC S5 ETHERNET LAYER 4/SIMATIC S5 PMC ETHERNET
- **S5 TF:** SIMATIC S5 ETHERNET TF
- **S7 ISO:** SIMATIC S7 PROTOCOL SUITE (Industrial Ethernet Channel Unit)
- **S7 TCP:** SIMATIC S7 PROTOCOL SUITE (TCP/IP Channel Unit)

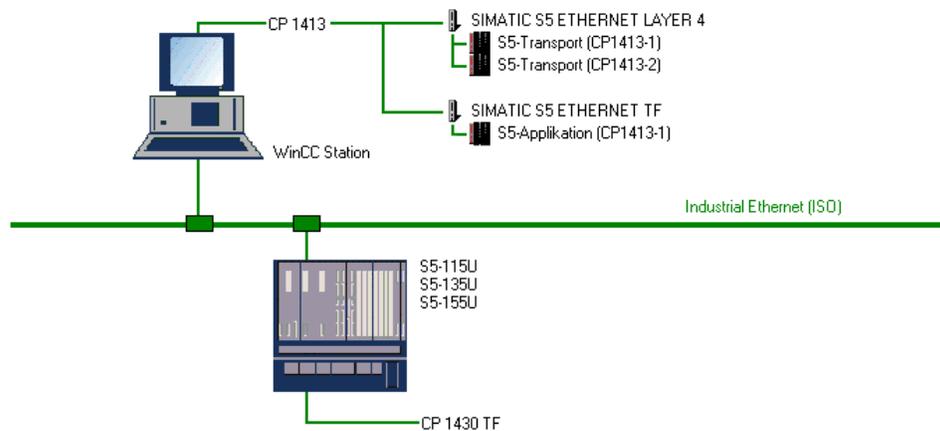
3.2.5.3 Communication with the SIMATIC S5

- For the communication to the SIMATIC S5 115U, S5 135U and S5 155U PLCs via the Industrial Ethernet, the following communication drivers are available.
- SIMATIC S5 Ethernet Layer 4 Communication via the SEND/RECEIVE and the WRITE/FETCH interface.
- SIMATIC S5 PMC Ethernet Communication via the SEND/RECEIVE interface with PMC handling blocks.
- SIMATIC S5 Ethernet TFC Communication via TF (Technological Functions)

For performance reasons, the communication driver SIMATIC S5 Ethernet Layer 4 should be used preferably over the communication driver SIMATIC S5 Ethernet TF.

Communication Partner

The various communication drivers enable the communication to the SIMATIC S5 115U, S5 135U and S5 155U PLCs. They must be equipped with the communication processor CP 1430 TF.



SIMATIC S5 ETHERNET LAYER 4

The communication driver SIMATIC S5 ETHERNET LAYER 4 enables the communication via the SEND/RECEIVE and the WRITE/FETCH interface. The WRITE/FETCH interface permits the active sending of data from the PLC.

The communication driver SIMATIC S5 ETHERNET LAYER 4 exclusively supports the communication via the communication processor CP 1413. Per computer, up to two modules can be used for the communication.

It is possible to communicate with 30 SIMATIC S5 stations per installed communication processor.

The following table lists the software components that are required for the implementation of the communication connection of a WinCC station to SIMATIC S5 via the SEND/RECEIVE interface.

Name	Description
WinCC	Communication Driver SIMATIC S5 ETHERNET LAYER 4
SIMATIC NET	Driver Software IE TF-1413

SIMATIC S5 ETHERNET TF

The communication driver SIMATIC S5 ETHERNET TF enables the communication via the TF interface (Technological Functions).

The communication driver SIMATIC S5 ETHERNET TF exclusively supports the communication via the communication processor CP 1413. Per computer, only one module can be used for the TF communication, which allows you to communicate with up to 30 SIMATIC S5 stations.

The following table lists the software components that are required for the implementation of the communication connection of a WinCC station to SIMATIC S5 via the TF interface.

Name	Description
WinCC	Communication Driver SIMATIC S5 ETHERNET TF
SIMATIC NET	Driver Software IE TF-1413 Configuration Software COML TF

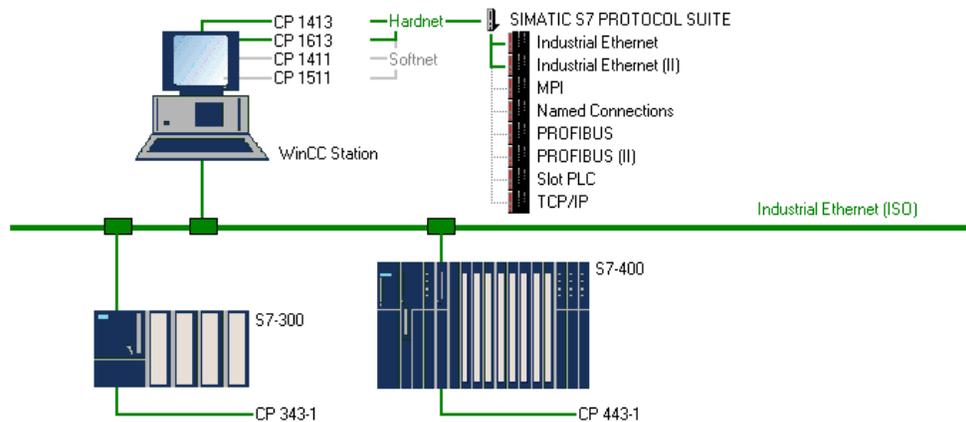
3.2.5.4 Communication with the SIMATIC S7

- The communication to the SIMATIC S7 is realized via the communication driver SIMATIC S7 PROTOCOL SUITE. It uses various channel units to provide the communication to SIMATIC S7-300 and S7-400 PLCs.
- ISO Transport Protocol Two Industrial Ethernet channel units are available for the communication via the ISO transport protocol
- ISO-on-TCP Transport Protocol The channel unit TCP/IP is available for the communication via the ISO-on-TCP transport protocol.

It is recommended to use the ISO transport protocol for networks with small dimensions due to its higher performance. If the communication stretches across more expansive networks connected by routers, the ISO-on-TCP transport protocol should be used.

Communication Partner

The communication driver SIMATIC S7 PROTOCOL SUITE enables the communication to the SIMATIC S7-300 and S7-400 PLCs. They must be equipped with communication processors supporting either the ISO or ISO-on-TCP transport protocol. The following graphic shows the possible communication partners.



Communication Data

The Industrial Ethernet and TCP/IP channel units support the communication via Hardnet and Softnet modules. The following table lists the driver software required for the various communication processors.

Communication Processor	Driver Software
CP 1413	IE S7-1413
CP 1613	IE S7-1613
CP 1411	IE SOFTNET-S7 BASIC
CP 1511	IE SOFTNET-S7 BASIC

The following table lists the number of PLCs that can be addressed by each communication processor installed in the WinCC station.

Module	Connections per Module
Hardnet Module	30
Softnet Module	8

With its two Industrial Ethernet channel units using the ISO transport protocol, the communication driver SIMATIC S7 PROTOCOL SUITE supports the communication with up to two modules. With its TCP/IP channel unit using the ISO-on-TCP transport protocol, it also supports the communication with one module.

4 Diagnosis of the Communication Connection

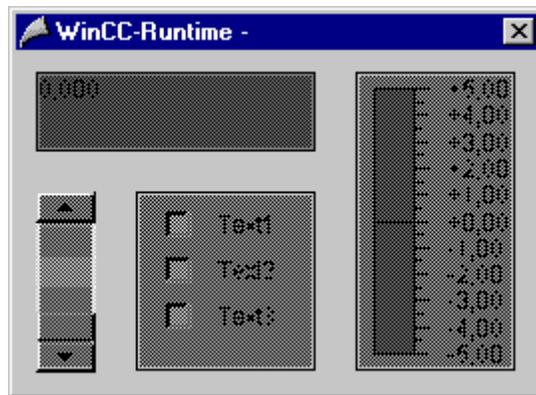
The following section is a guideline for troubleshooting and describes procedures for correcting errors.

If no communication can be established between a WinCC station and a PLC after configuring the connection, the biggest obstacle lies in finding the cause of error. The configuration of the connection between computer systems and PLCs can become a very complex task. Errors can creep in at various locations, preventing the connection between communication partners from being established successfully. In this context, the easiest errors are often the hardest ones to find. This section attempts to cover as many problem points as possible.

The correction of the found error is then comparably easy.

4.1 Error Detection

An error or the failure of establishing a communication connection is generally first recognized in runtime. If a WinCC picture is switched into runtime and a connection error is present, this fact will be displayed optically. Objects that have been made dynamic with WinCC tags and are not supplied with current process values will be displayed grayed out. Among others, these objects can be I/O fields, sliders, check-boxes or bar displays.



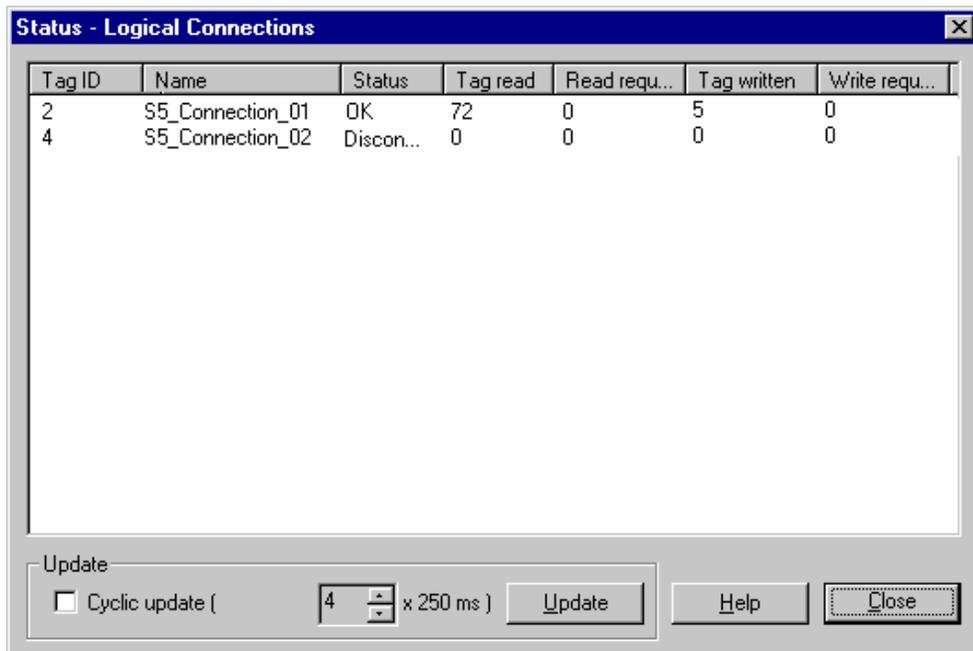
If not all WinCC tags of a connection show an error, this suggests an error source limited to individual WinCC tags. In this case, check the tag addressing, notation and application in the Graphics Designer.

If all WinCC tags of a connection show an error, this suggests an error source affecting the entire connection. WinCC has various information sources that can somewhat limit the possible error sources. However, do not expect a pinpoint location of the error.

4.1.1 WinCC Explorer

The WinCC Explorer includes a separate dialog box for determining the current status of configured connections. This dialog box can only be accessed if the WinCC project is in runtime. If the WinCC project is not in runtime, there are no established connections to the communication partners and therefore their status cannot be monitored.

The dialog box for monitoring the current connection status can be accessed from the WinCC Explorer via the *Tools* → *Status of Driver Connections* menu. The status of all configured connections will be displayed. By selecting the appropriate check-box, a cyclic update in a definable update cycle can be activated.



In the WinCC Explorer, the connection status can also be displayed by pointing the mouse pointer on the connection entry in the right window. The connection status will then be displayed as a Tooltip.

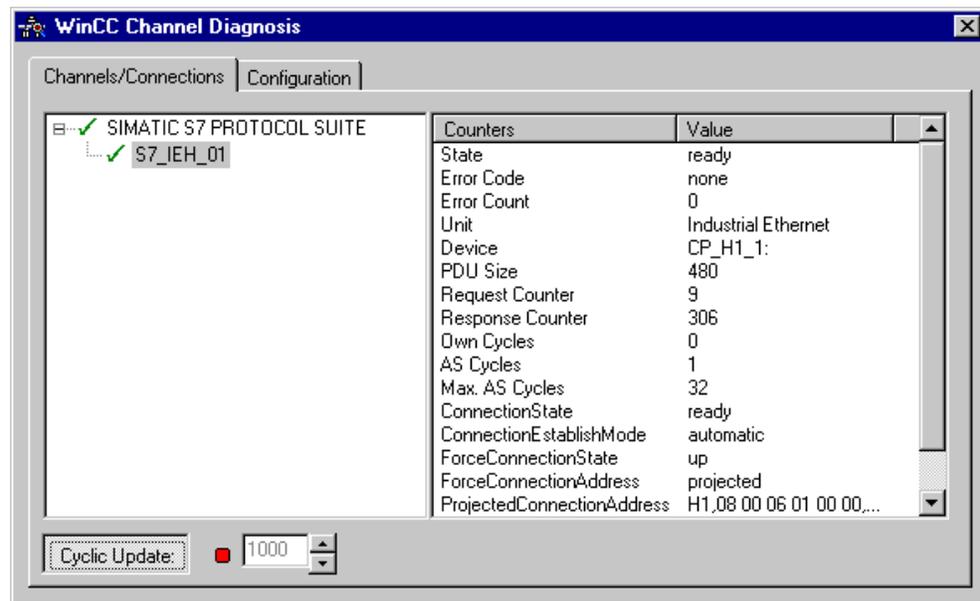
Name	Parameters	Last Change
S5_Connection_01	080006010000,...	28.04.99 13:46:02
S5_Connection_02	080006010002,...	29.04.99 09:26:02

Tooltip: Status: OK

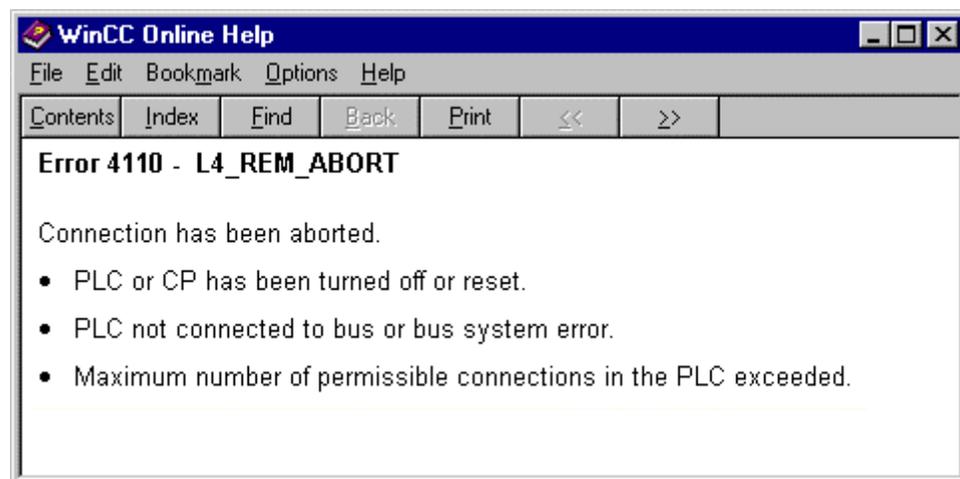
4.1.2 Channel Diagnosis

To diagnose the communication connections of a WinCC project, the Channel Diagnosis program is available. It can be started via Start → Simatic → WinCC → Channel Diagnosis. This program is only available in English.

If the WinCC project is in runtime, the *Channels/Connections* tab will display all configured connections. In addition, information pertaining to the current communication status of each connection will be displayed. The type and quantity of information displayed varies among the different WinCC communication drivers. By default, the displayed information is updated every second.



If there are connection errors, hexadecimal *Error Codes* will be displayed. These *Error Codes* aid in localizing error sources more precisely. To do so, you will need an explanation of the error code. By clicking on the entry of the *Error Code* and *Help*, the explanation of the corresponding *Error Code* will be retrieved from the WinCC Online Help.

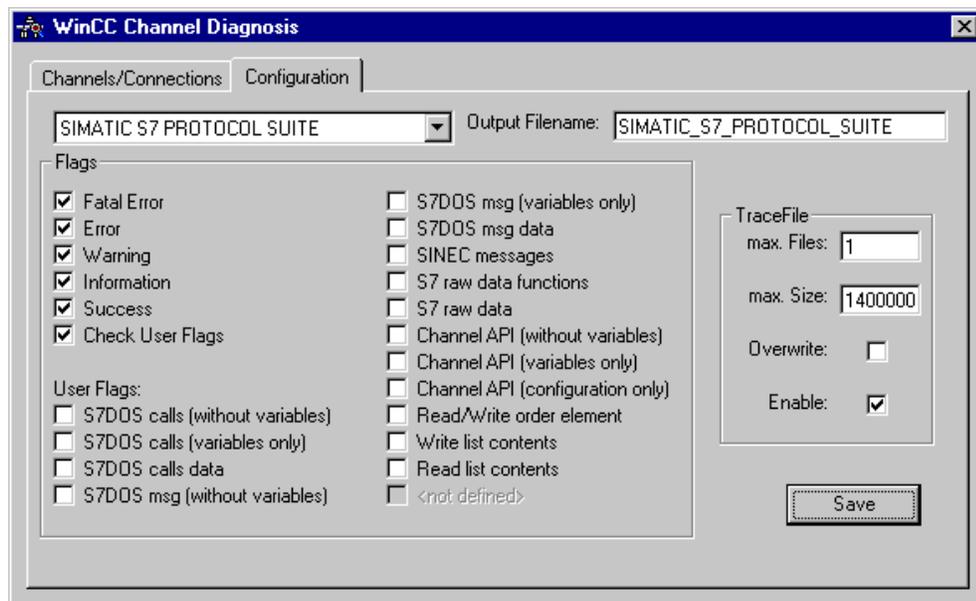


The Configuration tab gives you the option to configure a trace output to a text file. The desired trace depth can be set via the check-boxes in the Flags field.

In the *Output File Name* field, specify the name of the trace file. By default, the trace file is placed in the *Diagnosis* subfolder of the *WinCC* installation folder (*C:\Siemens\WinCC\Diagnosis*). The trace file will receive the extension *.TRC* and can be opened with any text editor.

In the *Trace File* field, additional trace file settings can be made. The *Enable* check-box activates the trace output.

The settings made must be saved by clicking on the *Save* button. A warning message will be displayed, informing you that the output of trace information will negatively impact the performance of the communication connection. You should therefore ensure that the trace output is deactivated while plant operations are running. The trace output should only be used during commissioning or for error detection. Before the trace output can occur, runtime must be exited and then started again.



The trace output can only be activated/deactivated if the project is in runtime.

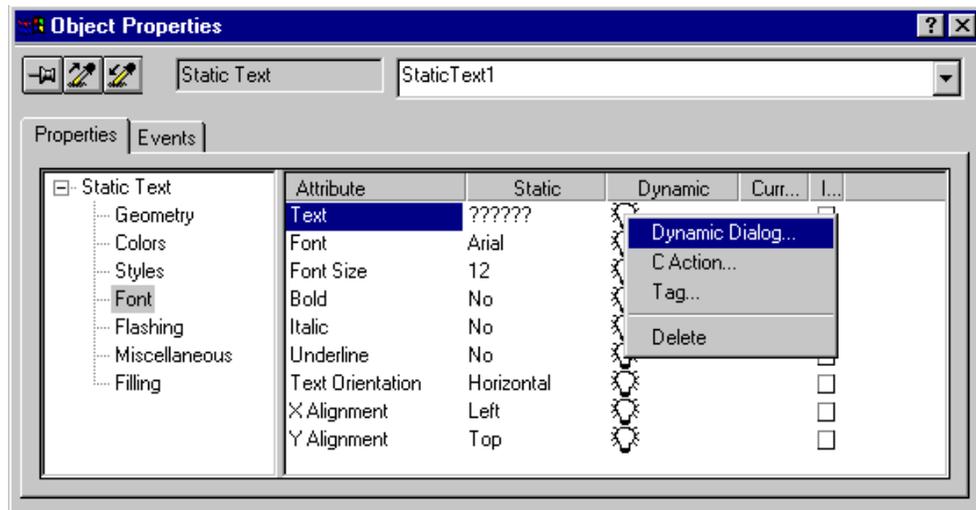
To deactivate the trace output, deselect the *Enable* check-box, store this setting by clicking on the *Save* button and exit runtime.

4.1.3 Status Monitoring using the Dynamic Dialog

The status of the individual WinCC tags can be monitored in WinCC runtime. Via the status of such a monitored WinCC tag, ruling out an addressing error, the status of its connection can be inferred. The setup of the status monitoring requires some configuration effort.

The configuration of the monitoring is performed in the *Graphics Designer*. In there, the status monitoring can be configured at any property of an object. For the monitoring, the *Text* property of a *Static Text* is a good choice. At this property, a *Dynamic Dialog* must be configured.

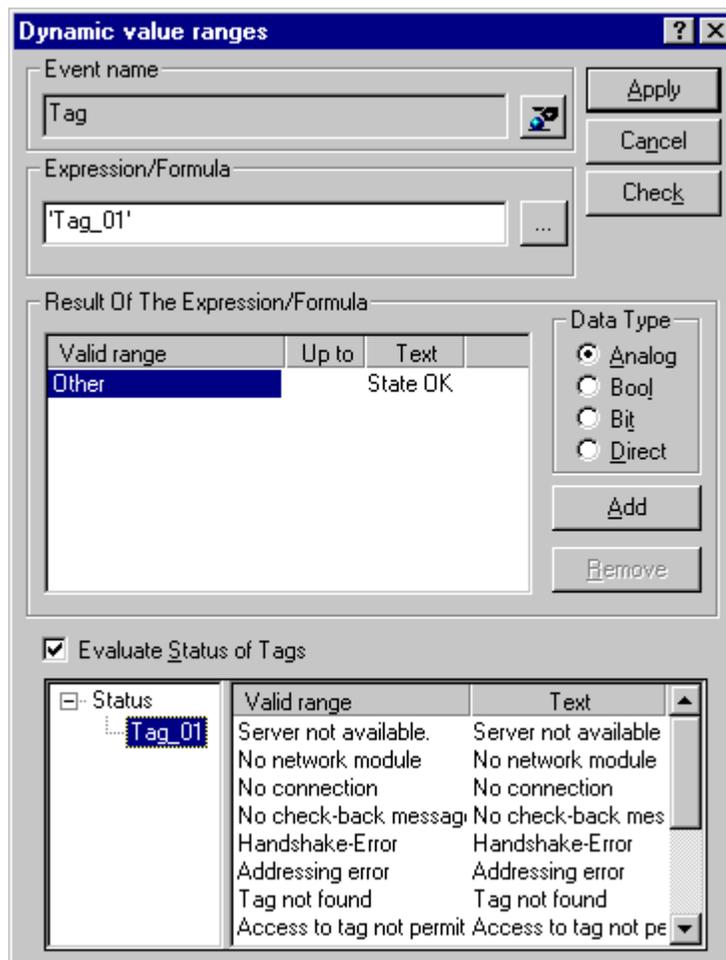
A dynamic dialog is created in the *Properties* tab of the object's properties dialog box. At the desired property,  on the *Dynamic* column and select *Dynamic Dialog* from the pop-up menu. This will open the *Dynamic Value Ranges* dialog box.



In the *Dynamic Value Ranges* dialog box, follow these steps:

- In the *Expression/Formula* field, specify the tag to be monitored.
- In the *Result of the Expression/Formula* field, in the line *Other*, enter the text *Status OK*.
- Activate the *Evaluate the Status of the Tag* check-box.
- In the following field, enter a corresponding status text for each existing valid range.
- Exit the dialog box by clicking on the *Apply* button.

If a different property is used for the status monitoring, corresponding property values must be specified instead of the texts (e.g. colors and such).



If the project is switched to runtime, the just configured object will display one of the entered status texts according to the current status of the tag. If there are no errors, the text *Status OK* will be displayed.

4.1.4 Status Monitoring using Global Actions

Another option to configure the status monitoring is to use the internal functions `GetTagState` and `GetTagStateWait`. Compared to the `GetTag` and `GetTagWait` functions, they not only return the value of a tag, but also its status. This status value can be evaluated and trigger various events.

For monitoring a tag, and with it its connection, the creation of a *Global Action* in the *Global Script* editor is suggested. This action is triggered by the monitored tag, for example every 2 seconds.

The *Global Action* determines the status value of the monitored tag via the *GetTagState* function corresponding to the tag type (each tag type has its own function). The status value 0 indicates an error-free communication connection, which can be evaluated in any form.

```
#include "apdefap.h"

int gscAction( void )
{
    DWORD dwState = 0;
    GetTagSWordState( "Variable_01", &dwState);
    if ( dwState == 0 )
    {
        //Connection OK
        SetTagBit( "BINi_E_CONNECTION", FALSE);
    }
    else
    {
        //Connection Error
        SetTagBit( "BINi_E_CONNECTION", TRUE);
    }
    return 0;
}
```

In the *Global Action* displayed above a WinCC tag of the *Signed 16-Bit Value* type is monitored. To determine the status of the tag, the type-specific *GetTagSWordState* function is used. As the first parameter of this function specify the name of the WinCC tag to be monitored, as the second parameter specify where the determined value is to be written. In the above example, the tag status is further evaluated by the internal tag *BINi_E_CONNECTION* especially created for this purpose. The value of this tag is set to *TRUE* in case of an error. The error can then be displayed in various ways. Among other things, this internal tag can be used by *Alarm Logging* to trigger an alarm or to display a message dialog box.

4.2 Troubleshooting Guideline

In this section, the communication connection is analyzed step-by-step to determine the responsible error cause of a communication error.

4.2.1 Is the Communication Partner operational?

A basic requirement for successfully establishing a connection is the readiness of the PLC. This mainly means that the PLC must be supplied with power and be in the on status. Additionally, all modules must run without errors.

If it is not possible to successfully activate all modules, the checks described in this section can be performed. In general, a successful activation of a module can be assumed, if no error LEDs light up.

Has all necessary Data been loaded?

The configuration of a PLC is primarily done via programming software on a programming device or computer. The data necessary (data blocks, program blocks, etc.) for the operation of a PLC must be loaded into its CPU. Check if all necessary data is present in the CPU.

If communication processors are used, it is also necessary to load their data. Check if all necessary data is present in the communication processor as well.

Are the individual Modules operational?

Determine the current module status of the individual components. A simple aid are the LEDs mounted on the modules. Generally, if an attempt to establish a communication connection is made, no error LEDs should light up on the PLC. The illumination of certain error LEDs (e.g. display of an empty buffer battery) may be uncritical for establishing a functional communication connection, however it is recommended to operate the PLC in a completely error-free state.

With SIMATIC S7 PLCs, the determination of the module status can be performed comfortably via the STEP7 software. To do this, click on the module entry to be checked and then select Target System > Module Status from the pop-up menu to open the Module Status dialog box. In the Status field of the General tab, the current module status and any existing errors are displayed. The Diagnosis Buffer tab contains more detailed information about existing errors and how to correct them.

Is the Hardware Configuration correct?

For SIMATIC S7 PLCs it is necessary to perform an exact configuration of the hardware. This is made possible by the HW-Config program of the STEP7 software. A hardware catalog for the exact definition of the components used is available. Ensure that the hardware configuration specified in the project corresponds to the actual configuration of the PLC. Individual hardware components are identified by their order number in the hardware catalog. These order numbers are also printed on the hardware components. Check these order numbers for agreement.

(0) UR2						
Slot	Module	Order Number	MPI Address	I Address	Q Address	Comment
1	PS 407 10A	6ES7 407-0KA00-0AA0				
3	CPU 416-1	6ES7 416-1XJ00-0AB0	2			
4	CP 443-5 Ext	6GK7 443-5DX00-0XE0		16380		
5						
6						
7						
8						
9						

In addition, load the hardware configuration into the PLC. Check also if all components of the hardware configuration have been loaded into the PLC.

Are the Communication Processors being synchronized?

With SIMATIC S5 PLCs it is necessary to synchronize any existing communication processors during startup of the CPU. This is performed by the SYNCHRON handling block in the startup blocks of the CPU.

As parameters of the *SYNCHRON* handling block, the interface number of the communication processor *SSNR*, the desired block size for the data transfer *BLGR* and the parameterization error byte *PAFE* are transferred.

Below, the call of the *SYNCHRON* handling block in a startup block of a *SIMATIC S5-115U* is displayed. In this PLC, it is the *FB 249*.

```

OB 21
NETZWERK 1 von 1
:SPA FB 249
NAME :SYNCHRON
SSNR : KY 000,000
BLGR : KY 000,000
PAFE : MB 110
:BE

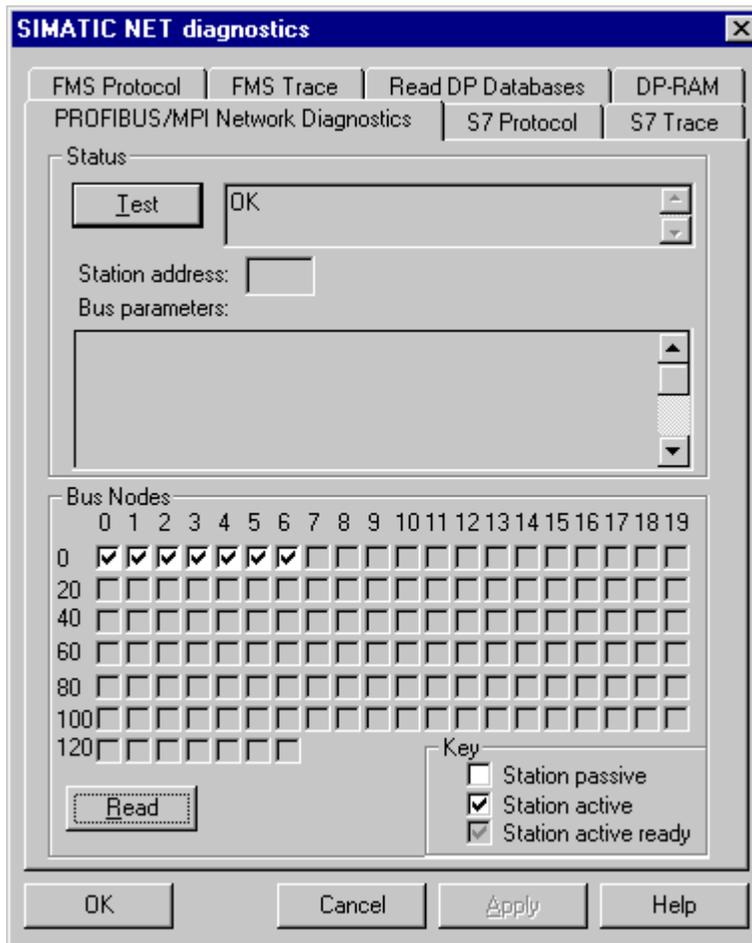
```

4.2.2 Is the Network Connection operational?

Without an operational network connection, a successful communication between a WinCC station and a PLC cannot be established. In this context, the most basic check involves the confirmation that the network connection to each communication partner is working properly (has the network cable been hooked up?).

Checking the Network Connection

There are several options to check a network connection. This means a check to determine if a communication partner can be reached at all via the network connection. For the check of a PROFIBUS or MPI network, the diagnosis of the communication processors used on these networks provides an excellent method. This is done via the program Setting the PG/PC Interface. Select the entry of the corresponding communication processor and click on the Diagnostics button. In the following SIMATIC NET Diagnostics dialog box, all stations accessible via the network connection can be determined by clicking on the Read button.



Are the correct Network Components being used?

To establish a network connection, the specially provided network components should be used. Otherwise there might be problems with establishing a connection, sporadic connection errors or loss of performance.

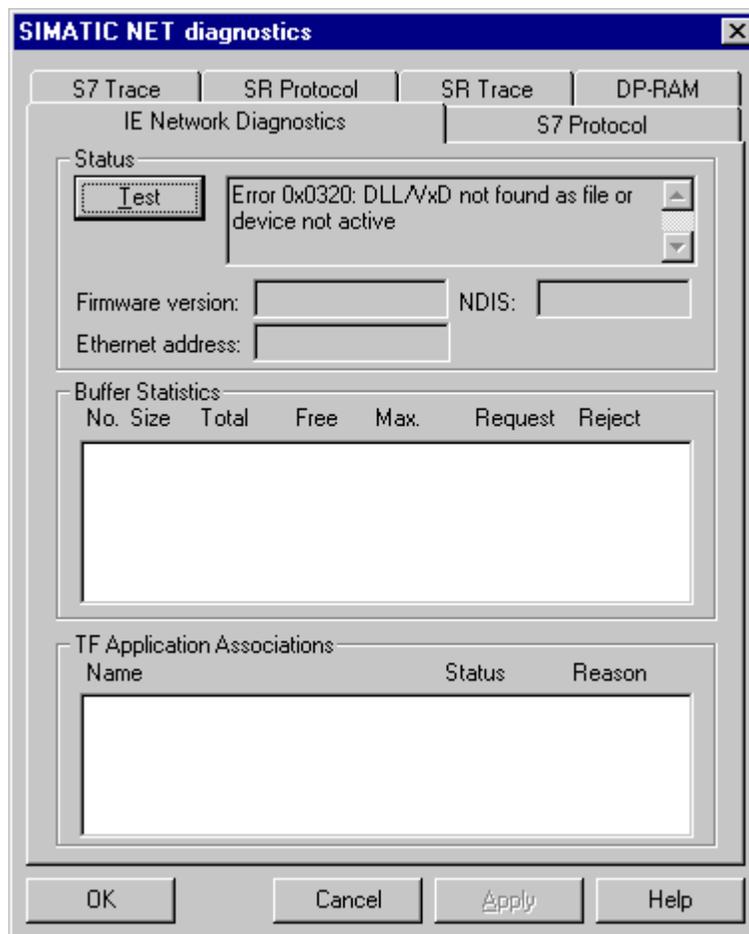
4.2.3 Is the Communication Module in the Computer operational?

To establish a communication connection from a WinCC station to a PLC, an appropriate communication module must be installed on the computer. In most cases, this module is a specifically designed communication processor. In some applications, the use of a regular network card or the computer's COM interface can be sufficient. The following explanations mainly deal with the application of communication processors.

Is the Communication Processor operational?

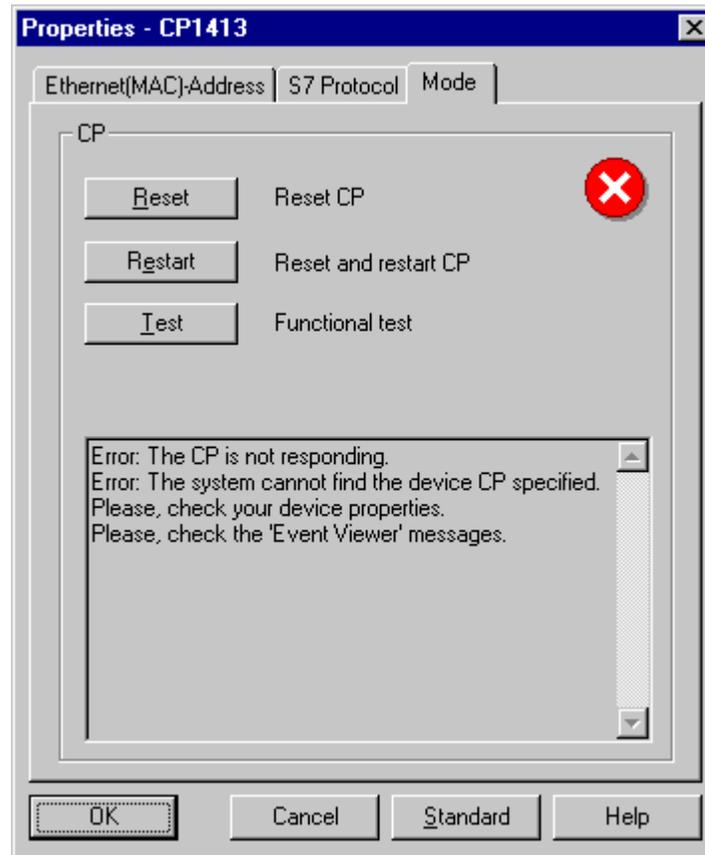
After completing the installation of a communication processor with the *Setting the PG/PC Interface* program, check if the installation has been successful. Generally, after exiting the *Setting the PG/PC Interface* program, the user is immediately notified if the installation has been successful or not.

Various communication processors provide additional diagnostic capabilities that can be accessed from the *Setting the PG/PC Interface* program via the *Diagnostics* button. Clicking on this button will display the *Simatic NET Diagnostics* dialog box. The diagnosis can be started by clicking on the *Test* button in the tab displayed by default. The result of the diagnosis will be displayed after that.



A second option to test a communication processor can be found in its Properties dialog box. This dialog box is accessed by clicking on the *Properties* button in the *Setting the PG/PC Interface* program.

From the *Operating Status* tab, the communication processor can be controlled. It can be reset, restarted or a test be performed. The results of the actions performed can be viewed in the lower output field.



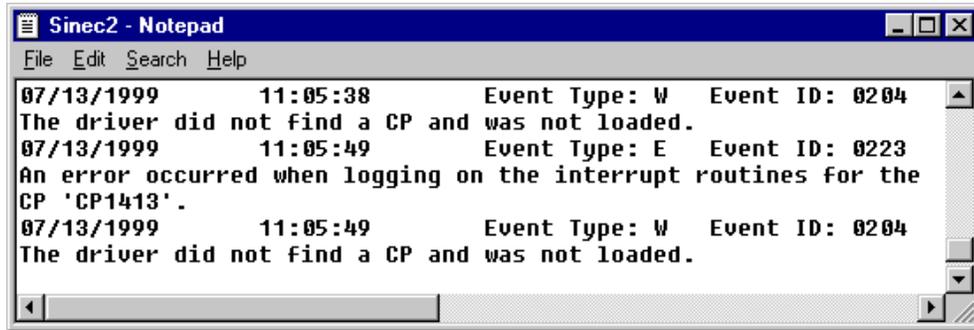
Initially, you can try to restart the module. If the restart fails, you will at least receive some information about the cause of error, which will be displayed in the lower output field and error message boxes.

Note:

While using the *Setting the PG/PC Interface* program, ensure that already set assignments between access points and the interfaces are not changed unintentionally. This mistake can be made easily due to the structure of the program. To avoid them, the program will point out changed assignments while exiting.

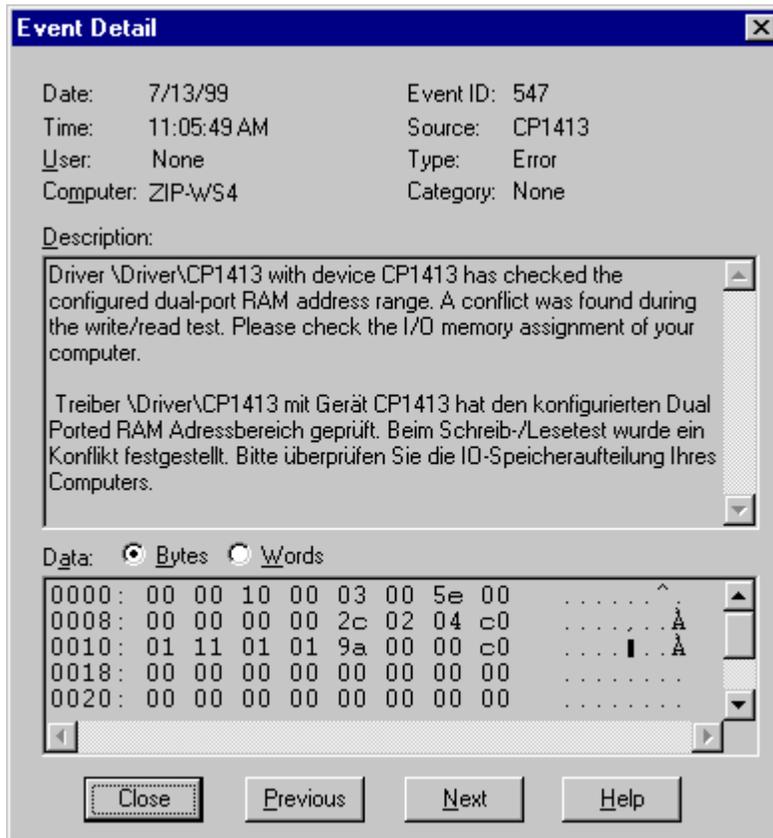
Determining the Error Source

A source of information for determining the error source is the Sinec2.log file. This file is located in the system folder of Windows NT. In the Windows NT Explorer, this file can be found in the C:\\WinNT\\System32 path and opened from there.



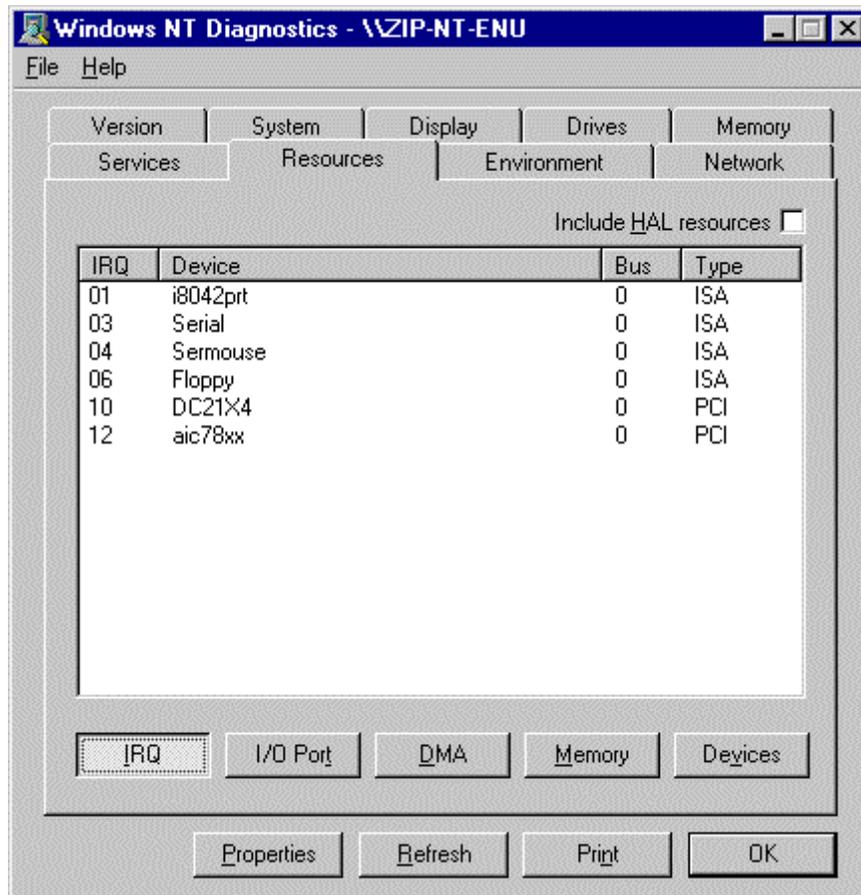
In addition, information can be gathered from the Event Viewer program. This program is started via Start → Programs → Administrative Tools (Common) → Event Viewer.

The *Event Viewer* program lists all events reported by the system.  On a certain event from the list opens the *Event Detail* dialog box, which contains more detailed information about the selected event. Via the *Previous* and *Next* buttons you can jump to adjacent events.



Are the assigned Resources still free?

If system resources have been assigned to the communication processor during installation, check if these resources have not already been taken by other devices. You can use the Windows NT Diagnostics program to check this. This program is started via Start → Programs → Administrative Tools (Common) → Windows NT Diagnostics. In the *Resources* tab of the *Windows NT Diagnostics* program, the *Resources* occupied by the various system components are listed.



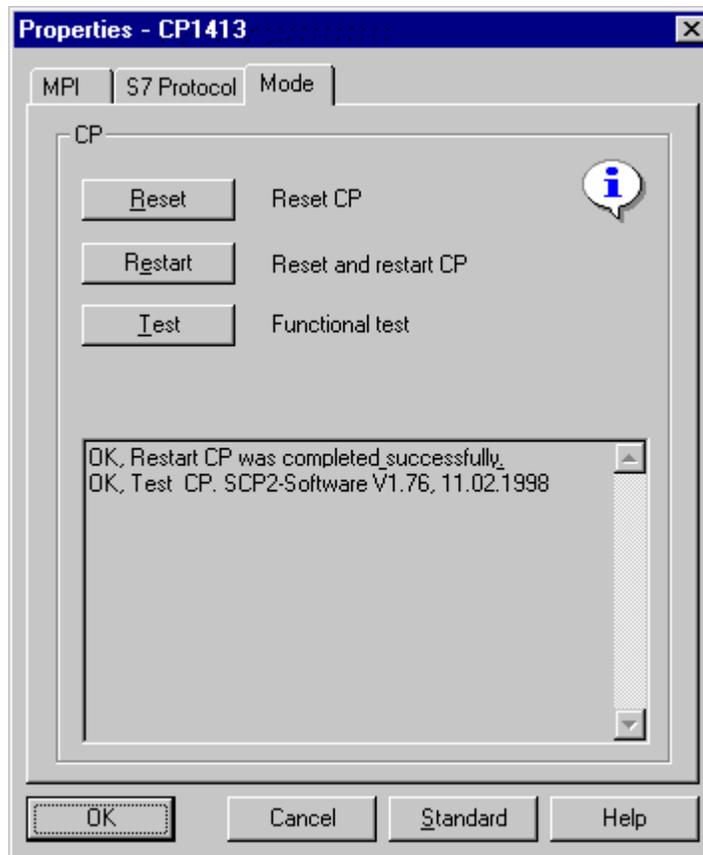
Compare the already occupied system resources to the ones set for the communication processor. The resources set for the communication processor during installation can be determined with the *Setting the PG/PC Interface* program as follows:

- From the *Setting the PG/PC Interface* program, open the *Install/Remove Modules* dialog box by clicking on the *Install* button.
- In the *Install/Remove Modules* dialog box, select the desired module entry from the *Installed* list.
- Click on the *Resources* button to display the resources set for the selected communication processor.

Compare these values to the already occupied resources.

If the resources set are already occupied, the settings for the communication processor must be changed to values that are displayed as not occupied by the *Windows NT Diagnostics* program.

Now, a restart of the communication processor with its new values must be performed. This is done from the Properties dialog box of the corresponding communication processor, which is accessed via the Properties button of the Setting the PG/PC Interface program. In the *Operating Status* tab, click on the Restart button to restart the communication processor. If the restart of the communication processor with its new settings has been successful or not can be gathered from the lower output field.



It is possible that a communication processor might not start even with checked settings. In such a case, experimenting with different settings can lead to success.

4.2.4 Is the Communication Configuration correct?

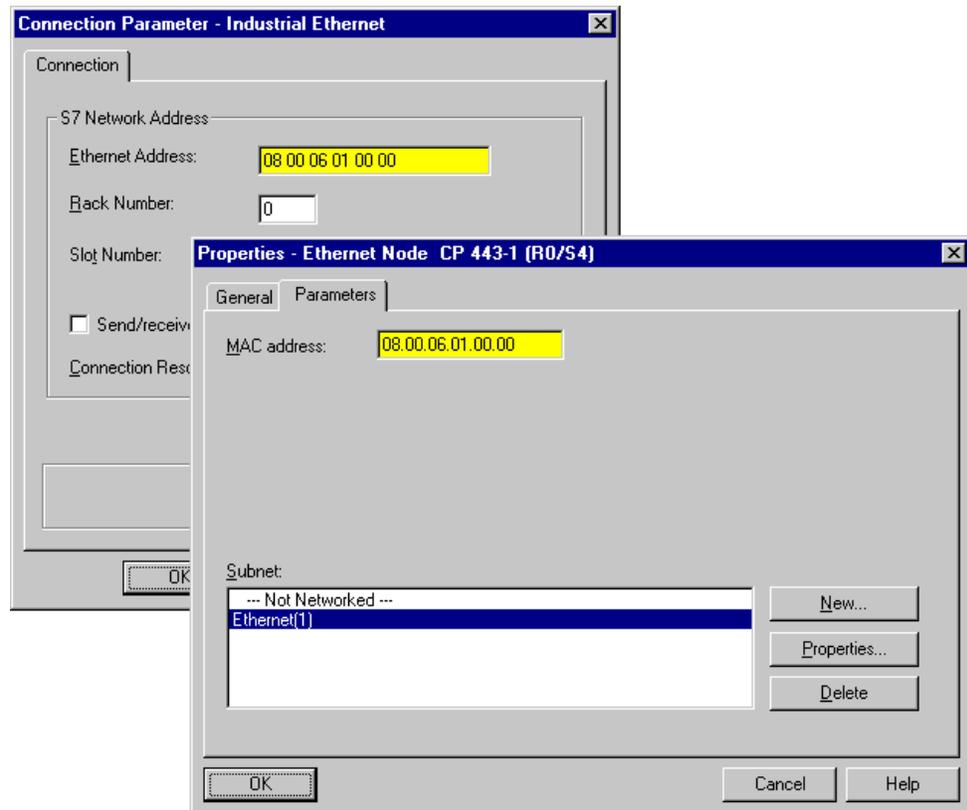
To establish a communication connection between a WinCC station and a PLC, certain information about the components involved (WinCC project, communication processors, PLC programs, etc.) must be known for the configuration of each component. This includes information about station addresses, tag names, etc. If one component does not have the correct information, establishing a connection might fail.

In this section you will check, if each communication partner has the correct information to successfully establish a connection.

Are the Station Addresses set correctly?

A majority of the communication drivers available in WinCC require the specification of the communication partner's station address for the configuration of a connection. Check if the correct station address - configured for the communication partner to be accessed - has been specified.

As an example, we will use connections that can be created via the communication driver *SIMATIC S7 Protocol Suite*. The following graphic illustrates the settings for the channel unit *Industrial Ethernet*. The address set in the *Connection Properties* dialog box must agree with the address configured for the communication partner.

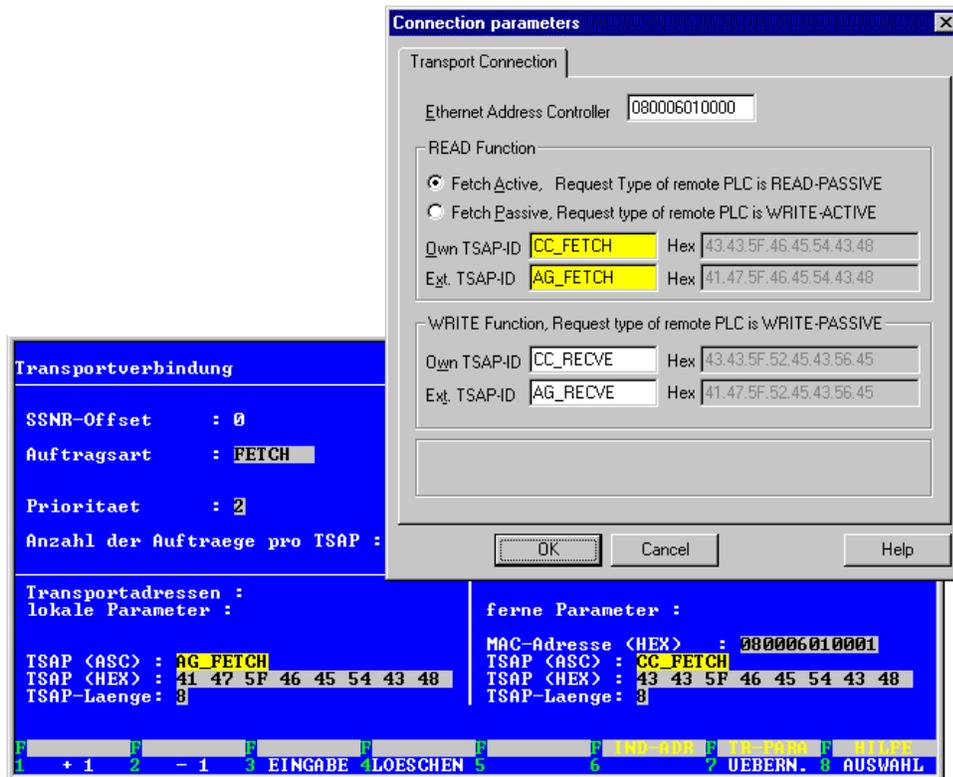


Are the TSAP Values set correctly?

Some communication drivers available in WinCC require the specification of TSAP values for the configuration. These are so-called Service Access Points, which represent the end points of various connections that are used for the execution of certain services. In this context, local and remote parameters are often mentioned. The local parameter concerns the value of the station currently being configured. The remote parameter concerns the value of the station to be accessed by the station currently being configured.

Check if the values entered here are the ones that have been configured for the communication partner to be accessed. Ensure that the local and remote parameters have not been reversed.

As an example, we will use connections that can be created via the communication driver *SIMATIC S5 Ethernet Layer 4*. The TSAP values set in the *Connection Properties* dialog box must agree with the values set for the communication partner. In this example, a comparison of the TSAP values in the hexadecimal format can also be useful.



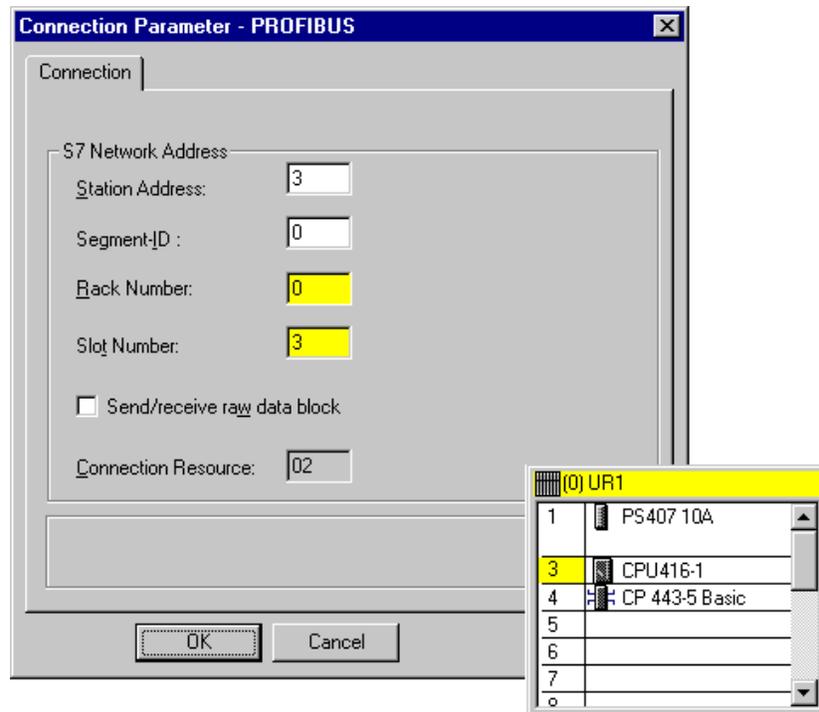
Are the correct Rack and Slot Numbers being used?

If a connection is configured via the communication driver SIMATIC S7 Protocol Suite, the exact position of the CPU must be specified except if the channel unit Named Connections is used. In other words, the rack number and slot number of the CPU to be accessed must be specified.

Ensure that the values of the communication processor - which are used to establish the network connection - are not entered by mistake in the PLC.

Check if the correct rack and slot numbers are used in WinCC and STEP7, and that they agree with the actual values.

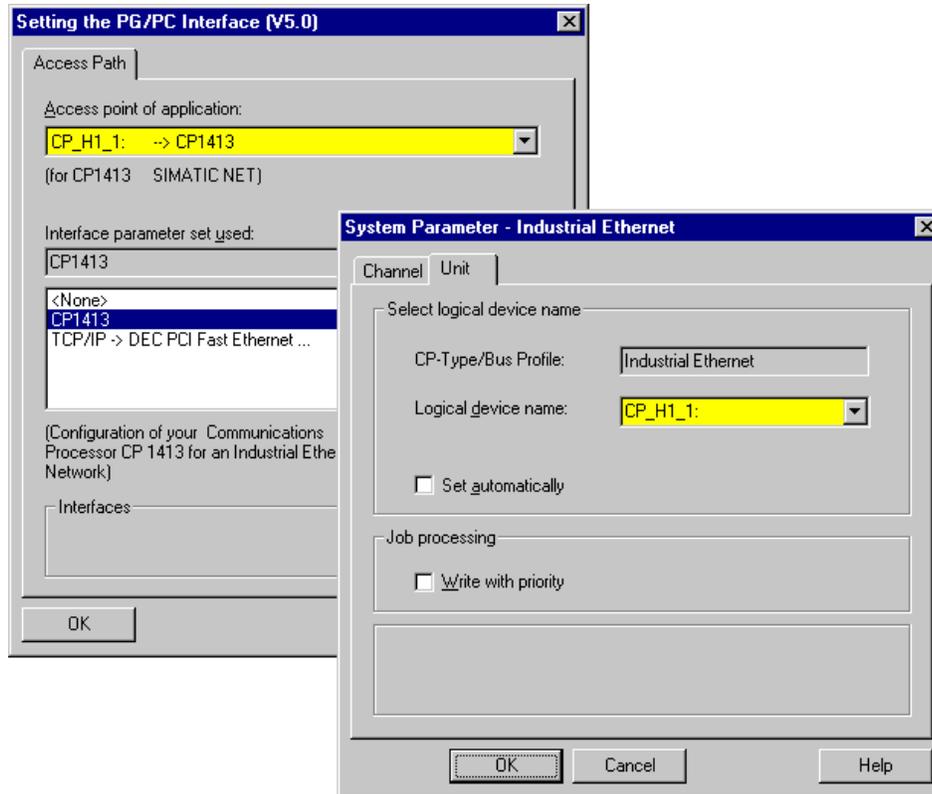
The following graphic shows the settings for the channel unit *PROFIBUS*. The values set in the *Connection Properties* dialog box must agree with the values configured in STEP7.



Has the correct Access Point been set?

In most cases, the definition of the module to be used for the communication by a certain channel unit of a communication driver is implemented by setting an access point. Check if the access point set here has been correctly assigned to the desired module.

As an example, we will use connections that can be created via the communication driver *SIMATIC S7 Protocol Suite*. The following graphic illustrates the settings for the channel unit *Industrial Ethernet*. The desired communication module must be assigned to the access point set in the *System Parameters* dialog box.

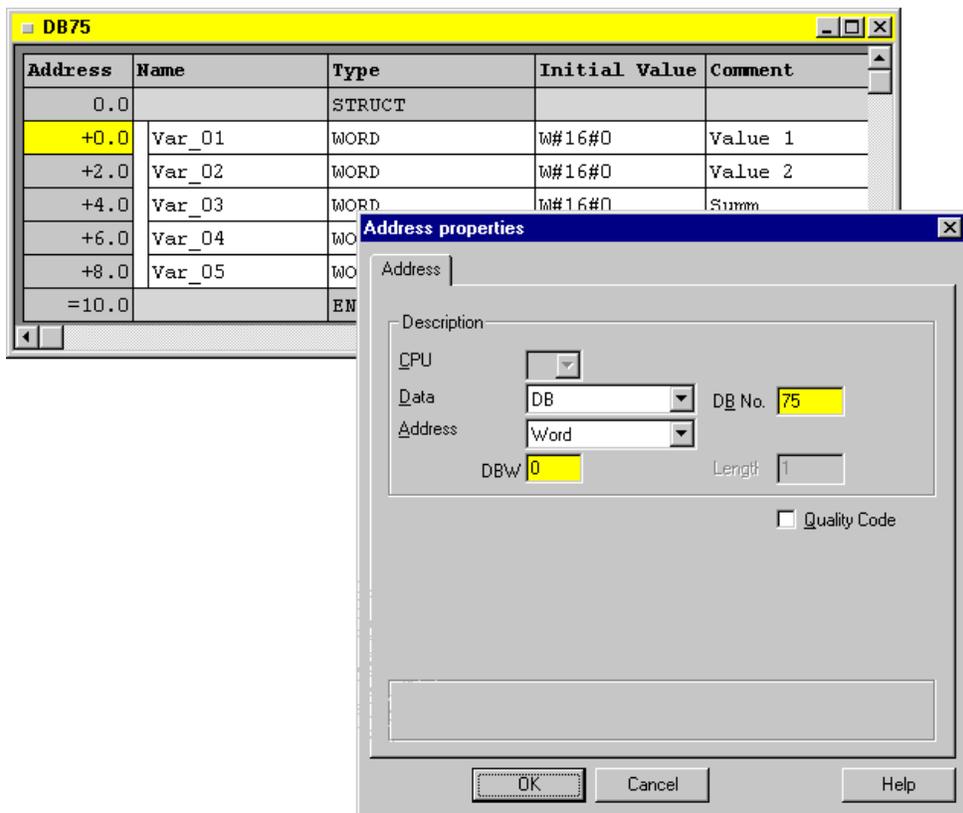


In some cases, you have the option of letting WinCC set the access point automatically. In this case, check if WinCC has selected the correct module.

Is the Addressing of the Tag correct?

If a connection cannot be established to a certain group of WinCC tags created for a connection, the most likely cause is an addressing error of the concerned WinCC tags. The communication partner must be able to map the address configured for a WinCC tag. Check if the addressing has been performed correctly and that the specified addresses exist in the communication partner.

As an example, we will use connections that can be created via the communication driver *SIMATIC S7 Protocol Suite*. The address set in the *Address Properties* dialog box must be available in the SIMATIC S7.

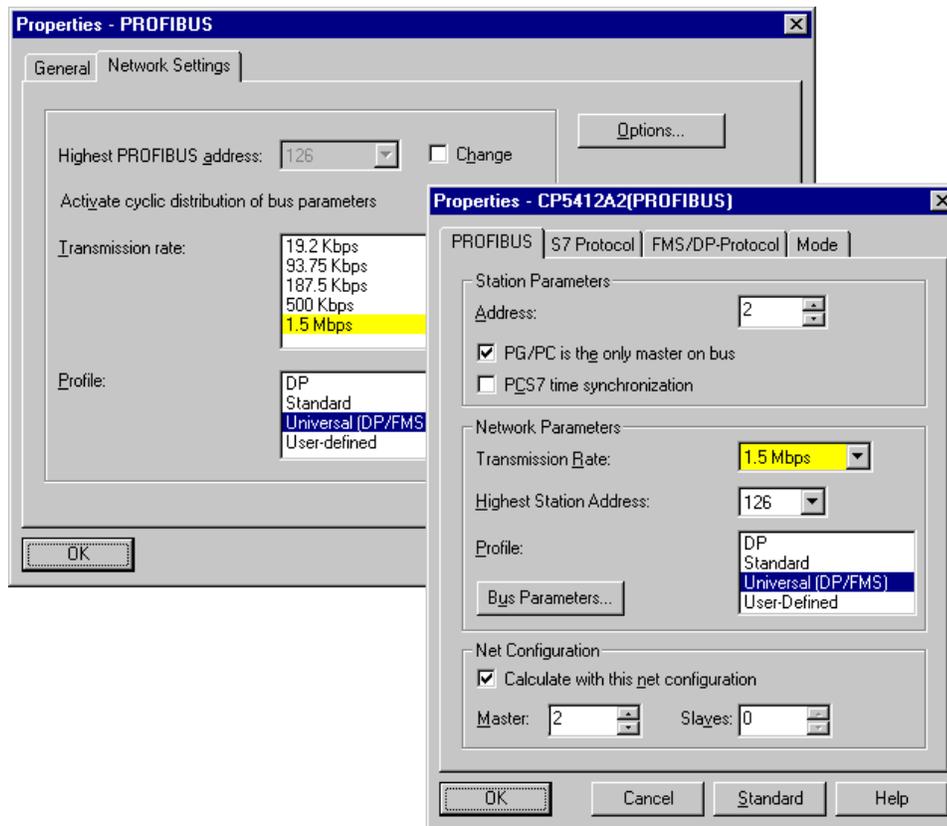


Another often overlooked error source can be attributed to the configuration of an object in the Graphics Designer that has been made dynamic via a WinCC tag. If the name of this WinCC tag has been entered using the keyboard, typing errors can slip in that can lead you to the erroneous conclusion that a connection error occurred. To avoid such errors, use the tag selection dialog box.

Are the Network Settings correct?

To establish a working communication connection, some parameters concerning the data transfer must be specified. The type and number of parameters to be set depend on the type of communication network. These parameters must be reported to the various communication partners. Ensure that all communication partners receive identical parameters.

As an example, we will use connections that can be created via the *PROFIBUS* communication driver. Among other things, the transfer rates set for the individual stations must agree. The following graphic shows the settings at a SIMATIC S7 and WinCC station.



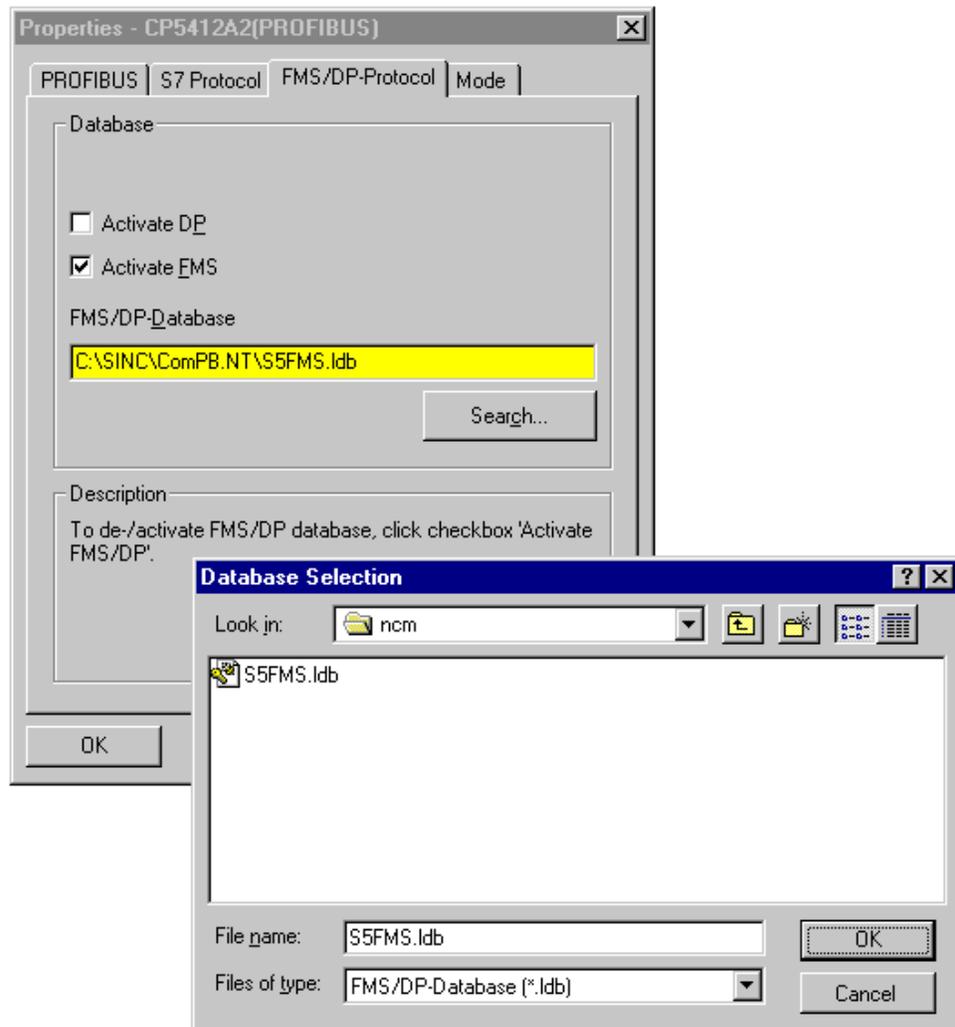
If database files are created for the communication, the network settings in them must also be checked for correctness. As examples for the application of database files, we will use communication connections that can be created via the PROFIBUS protocols DP and FMS.

Is the correct Database File being used?

If a database file is created to realize a connection, it must be accessible by the communication module. The path to this file must be specified. Check if the correct database file has been set and its path and name is correct.

As an example, we will use connections that can be created via the communication driver PROFIBUS FMS. In this case, the database file must be assigned to the communication module used.

Also make sure that the database file is actually being used. For this, restart the communication module after assigning the database file. A restart can be performed from the module's Operating Status tab in its Properties dialog box.



4.3 Further Diagnosis

Additional support for localizing and correcting errors is available from the WinCC Hotline, which can be contacted by phone at +49 (911) 895-7000 (Fax -7001). To receive the best support possible, have the following information ready before calling:

- Information about the operating system used: VersionService Pack
- Information about the software used: WinCC VersionOption PackagesDriver Software
- Information about the hardware used:Communication Processor of the PCBus ComponentsPLC Components
- Trace files created with the Channel Diagnosis program

4.3.1 Creating Connections via API Functions

This section introduces the concept of creating a connection to a PLC using an API function. To establish a connection via API functions, the communication driver used must first be installed in the WinCC Explorer.

The sample illustrates a connection to the SIMATIC S7 via the PROFIBUS channel unit. This new connection contains the connection parameters shown below:

Connection Parameter - PROFIBUS

Connection

S7 Network Address

Station Address: 1

Segment-ID : 2

Rack Number: 3

Slot Number: 4

Send/receive raw data block

Connection Resource: 02

OK Cancel Help

Note:

The C-Action used to create the connection is configured at the Mouse Action event of a button. To execute the C-Action, the WinCC picture containing the configured button must be switched into runtime. The action can be tested in any WinCC project. This procedure is only used to demonstrate the application of the corresponding API functions. There are no actual applications, where the creation of a connection from runtime would become necessary.

C-Action for Creating a Connection

```
#include "apdefap.h"
void OnClick(char* lpszPictureName, char* lpszObjectName, char* lpszPropert
{
    char szProjectFile[_MAX_PATH+1];
    MCP_NEWCONNECTION_DATA mcpConnection;
    CMN_ERROR Error;

    if(!DMGetRuntimeProject(szProjectFile,_MAX_PATH+1,&Error))
    {
        printf("%s\r\n",Error.szErrorText);
    }

    mcpConnection.dwFlags = MCP_NCON_FLAG_CREATE;
    strcpy(mcpConnection.szProjectFile,szProjectFile);
    strcpy(mcpConnection.szUnitName,"PROFIBUS");
    strcpy(mcpConnection.szConnection,"GAPICConnection");
    strcpy(mcpConnection.szCommon,"");
    strcpy(mcpConnection.szSpecific,"L2,1 2,,3,4,02");

    if(!GAPICreateNewConnection(&mcpConnection,&Error))
    {
        printf("%s\r\n",Error.szErrorText);
    }
}
```

- In the first section, the project name is determined. It can be determined while runtime is active via the DMGetRuntimeProject() function.
- In the next section, the desired values are written to the mcpConnection tag of the MCP_NEWCONNECTION_DATA structure type. Pay special attention to the syntax of the value to be entered in the szSpecific structure element. This structure element defines the connection parameters of the new connection to be created.
- In the last section, the new connection is created via the GAPICreateNewConnection() function.

Created Connection

The following illustration depicts the connection created by the C-Action described above in the WinCC Explorer.

Name	Parameters	Last Change
 GAPICConnection	L2,1 2,,3,4,02	03.09.99 17:57:15

Index

A

Access Method 2-7, 2-19
 Industrial Ethernet 2-19
 MPI 2-14
 Access Point 4-20
 Actuator-Sensor Level 2-13
 Adapt Format 3-4
 Addressing 3-1, 4-21
 Alarm Logging Runtime 3-2
 Application Layer 2-10
 AUI 2-20

B

Bridge 2-10
 Bus 2-4
 Bus Parameters 4-22

C

Cell Level 2-12
 Channel Diagnosis 4-3
 Channel Unit 2-24
 COM Interface 3-14
 COM PROFIBUS 3-19, 3-20
 Communication 2-2, 3-11
 Communication Data 3-11
 Communication Drivers 2-21, 2-23
 Communication Functions 2-3
 Communication Partner 2-2
 Communication Service 2-3
 Session Layer 2-9
 WinCC Communication 2-22
 Communication Drivers 3-24
 Industrial Ethernet 3-24
 PROFIBUS 3-17
 Serial Communication 3-14
 Communication Partner 3-12, 3-24
 Industrial Ethernet 3-24
 MPI 3-15
 PROFIBUS 3-18
 PROFIBUS DP 3-19
 PROFIBUS FDL 3-22
 PROFIBUS FMS 3-20
 Serial Communication 3-14
 Communication Processor
 Diagnosis 4-12
 Industrial Ethernet 3-23

MPI 3-15
 PROFIBUS 3-16
 Connection 2-3
 Costs 3-13
 CSMA/CD 2-7
 Cycle 3-5, 3-7

D

Data Manager 2-21
 Data Organization 3-9
 Data Type 2-25, 3-4
 Space Requirement 3-4
 Database File 4-23
 Diagnosis
 Communication Processor 4-12
 PROFIBUS 4-10
 DP 2-17, 3-19
 Dynamic Dialog Box 4-5

E

Event Display 4-14

F

FDL 2-17, 3-22
 Field Level 2-12
 FMS 2-17, 3-20

G

Gateway 2-10
 Global Script Runtime 3-2
 Graphics Runtime 3-2

H

Handling Block
 SYNCHRON 4-9
 Hardnet 3-23
 Industrial Ethernet 3-23
 PROFIBUS 3-16
 Hardware Catalog 4-9
 Hotline 4-24

I

- Industrial Ethernet 2-13, 2-18, 2-20, 3-23
 - Access Methods 2-19
 - Communication Partner 3-24
 - Electrical Network 2-20
 - Protocol Profiles 2-19
 - Standardization 2-18
 - Transfer Media 2-20
 - Transport Protocols 2-19
- ISO Transport 2-19
- ISO-on-TCP Transport 2-19
- ISO-OSI Reference Model 2-8
- ITP 2-20

L

- LAN 2-6
- Line Structure 2-4
- Load 4-9
- Log File 4-14

M

- MAC 2-9
- MAN 2-6
- Management Level 2-12
- MAP 2-19
- Master Slave 2-7
- Message 3-3, 3-10
 - Number of Messages 3-6
 - Optimization 3-10
- Monitoring of State 4-7
- MPI 2-13, 2-14, 3-15
 - Access Methods 2-14
 - Number of Stations 2-14
 - Transfer Medium 2-15

N

- Network 2-3
- Network Layer 2-9

O

- OLM 2-18
- OLP 2-18
- OPC 2-20

P

- PA 2-17
- PDU 3-3
- Physical Layer 2-9
- Point to Point 2-4
- Presentation Layer 2-9
- PROFIBUS 2-13, 2-15, 2-17
 - Communication Partner 3-18
 - Electrical Network 2-17
 - Number of Stations 2-15
 - Optical Network 2-18
 - PROFIBUS DP 2-17, 3-19
 - PROFIBUS FDL 2-17
 - PROFIBUS FMS 2-17, 3-20
 - PROFIBUS PA 2-17
 - Protocol Architecture 2-17
 - S7 Functions 2-17
 - Standard 2-15
 - Transfer Media 2-17
- Protocol 2-3, 2-19
 - Industrial Ethernet 2-19
 - PROFIBUS 2-17

R

- Rack Number 4-19
- Read Acyclically 3-8
- Read Cyclically 3-8
- Repeater 2-10, 2-17
- Resources 4-15
- Restart 4-16
- Ring Structure 2-4
- Router 2-10

S

- S7 Functions 2-19
 - Industrial Ethernet 2-19
 - PROFIBUS 2-17
- SEND/RECEIVE 2-19
 - Industrial Ethernet 2-19, 3-25
 - PROFIBUS 2-17
- Serial Communication 3-14
- Session Layer 2-9
- SIMATIC S5 3-26
 - ETHERNET LAYER 4 3-26
 - ETHERNET TF 3-26
 - PMC Ethernet 3-25
 - PROFIBUS FDL 3-22
 - Synchronization 4-9
- SIMATIC S7 4-9
 - Diagnosis 4-9

- Industrial Ethernet 3-27
- MPI 3-15
- PROFIBUS 3-21
- TCP/IP 3-27
- Slot Number 4-19
- Softnet 3-23
 - Industrial Ethernet 3-23
 - PROFIBUS 3-16
- Software Interface 2-3
- Star Structure 2-5
- Station 2-2
- Station Address 4-17
- Subnet 2-2
- Synchronization 4-9

T

- Tag 2-25, 3-1
 - Addressing 3-1
 - Data Type 2-25
 - Space Requirement 3-4
- Tag Logging Runtime 3-2
- TF 2-19
- Token 2-16

- Token Passing 2-7
- Topology 2-4, 2-6
- Trace 4-4
- Transfer Medium 2-6, 2-20
 - Industrial Ethernet 2-20
 - MPI 2-15
 - PROFIBUS 2-17
- Transport Layer 2-9
- Tree Structure 2-5
- TSAP 4-18

U

- UDP 2-19
- Update Cycle 3-5, 3-7
- Upon Change 3-5

W

- WAN 2-6
- Windows NT Diagnostics 4-15

