

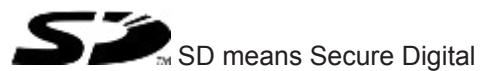
Automation system PSS 4000

pilz

This document is the original document.

All rights to this documentation are reserved by Pilz GmbH & Co. KG. Copies may be made for internal purposes. Suggestions and comments for improving this documentation will be gratefully received.

Pilz®, PIT®, PMI®, PNOZ®, Primo®, PSEN®, PSS®, PVIS®, SafetyBUS p®, SafetyEYE®, SafetyNET p®, the spirit of safety® are registered and protected trademarks of Pilz GmbH & Co. KG in some countries.



Section 1	Introduction	9
	1.1 Validity of the documentation	9
	1.2 Overview of documentation	10
	1.3 Definition of symbols	11
Section 2	Automation system PSS 4000	12
	2.1 Introduction to PSS 4000	12
	2.1.1 General	12
	2.1.2 SafetyNET p	13
	2.1.3 Devices in the automation system PSS 4000	14
	2.1.4 PAS4000	22
	2.1.5 Diagnostics	23
	2.1.6 OPC Server	24
	2.2 Modularity with PSS 4000	25
	2.2.1 Centralised view of a networked system	25
	2.2.2 Scalability of the devices	26
	2.2.3 Modularisation within programming	27
Section 3	Safety	28
	3.1 General guidelines	28
	3.2 Intended use	29
	3.2.1 Safety Integrity Level (SIL) and Performance Level (PL)	30
	3.2.2 General safety assessments	31
	3.3 Use of qualified personnel	32
	3.4 Failsafe inputs/outputs	33
Section 4	PSSu system basics	34
	4.1 Structure of a PSSu system	35
	4.2 Module layout	38
	4.3 Module descriptions	40
	4.4 Module bus (MBUS)	42
	4.5 Connections on the base modules	44
	4.5.1 Connection levels on input/output modules	45
	4.5.2 Colour marking on the connection levels	46
	4.6 Connections on the compact modules	47
	4.7 Maximum system expansion of a PSSu system	48
	4.7.1 Number of modules and number of inputs/outputs on a PSSu system	49
	4.7.2 Maximum current load	50
	4.8 Slots of a PSSu system	52
	4.8.1 Set/actual hardware registry on a PSSu system	54
	4.9 Supply voltage and insulation	57
	4.10 Supply groups	60
	4.11 Status LEDs on head modules	62
	4.12 I/O data	69
	4.12.1 I/O data of PSSu modules on the module bus	70
	4.12.2 I/O data with external communication	72

Section 5	Ethernet	73
	5.1 Auto IP mechanism	73
	5.2 Ethernet interface	74
	5.2.1 RJ45 interface	75
	5.2.2 RJ45 interface assignment	76
	5.2.3 Connector and connection cable for RJ45	77
	5.2.4 Status LEDs on the RJ45 interface	79
Section 6	SafetyNET p	80
	6.1 Overview	80
	6.1.1 SafetyNET p in ISO/OSI's 7-layer reference model	81
	6.1.2 RTFN (Real Time Frame Network)	82
	6.1.3 RTFL (Real Time Frame Line)	83
	6.1.4 Transfer of safety-related and non-safety-related data	84
	6.2 Topology on SafetyNET p	85
	6.2.1 Linear topology	85
	6.2.2 Star topology	86
	6.2.3 Dynamic structures	87
	6.3 Selecting SafetyNET p components	88
	6.3.1 Selecting components for RTFL	88
	6.3.2 Selecting components for RTFN	88
	6.4 SafetyNET p interface	89
	6.5 Process data exchange via SafetyNET p RTFN	90
	6.5.1 Cyclical processing of connections	92
	6.5.2 Data transfer limits	93
	6.6 Network without Routers	97
	6.7 Network with Routers	98
	6.7.1 System variable PILZ_SNP_MULTICAST_TTL	100
	6.8 Prevent data exchange with projects with identical project name	101
Section 7	External communication	102
	7.1 Communication modes for external communication	102
	7.2 Modbus/TCP	103
	7.2.1 "IP connections" system section	104
	7.2.2 Modbus/TCP connections	105
	7.2.3 Modbus/TCP Server connection (server connections)	106
	7.2.4 Function Codes (Client connections)	107
	7.2.5 Relationship between function codes and Modbus/TCP data areas	108
	7.2.6 Data transfer limits	109
	7.2.7 Data exchange via Modbus/TCP	111
	7.2.8 Modbus/TCP I/O data	114
	7.2.9 Address display of Modbus/TCP addresses	119
	7.2.10 Data consistency on Modbus/TCP	120
	7.2.11 Special features when using the data areas 0x and 4x	121
	7.3 Raw UDP	126
	7.3.1 "IP connections" system section	127
	7.3.2 Raw UDP connections	128

7.3.3	Data exchange via Raw UDP	129
7.3.4	Data transfer limits	130
7.3.5	Raw UDP I/O data	131
7.3.6	Address format of Raw UDP addresses	134
7.3.7	Data consistency on Raw UDP	135
7.4	PROFIBUS-DP slave	136
7.4.1	"PROFIBUS-DP Slave" system section	137
7.4.2	Data exchange via PROFIBUS-DP	138
7.4.3	Data transfer limits	144
7.4.4	I/O data of PROFIBUS-DP Slave	145
7.4.5	Address display of PROFIBUS-DP addresses	148
7.4.6	PROFIBUS diagnostics	150
7.4.7	Data consistency on PROFIBUS-DP	155
7.5	PROFIBUS-DP Slave with PROFI-safe	156
7.5.1	"PROFIBUS-DP Slave with PROFI-safe" system section	157
7.5.2	PROFIBUS-DP in the "PROFIBUS-DP Slave with PROFI-safe" system section	158
7.5.3	PROFI-safe in the "PROFIBUS-DP Slave with PROFI-safe" system section	159
7.5.4	GSD file for virtual PROFIBUS modules	165
7.5.5	Data transfer limits	166
7.5.6	PROFI-safe I/O data	167
7.5.7	Address format of PROFI-safe addresses	170
7.5.8	Deactivation and reactivation of an F-module	172
7.5.9	F-parameters	173
7.5.10	PROFI-safe diagnostics	174
7.5.11	Data consistency on PROFI-safe	175
7.6	EtherNet/IP Adapter	176
7.6.1	"EtherNet/IP Adapter" system section	178
7.6.2	EtherNet/IP Adapter connections	179
7.6.3	Data exchange via EtherNet/IP	180
7.6.4	Data transfer limits	183
7.6.5	EtherNet/IP Adapter I/O data	184
7.6.6	Address format of EtherNet/IP addresses	187
7.6.7	EtherNet/IP diagnostics	189
7.6.8	Data consistency on EtherNet/IP	191
7.6.9	Supported EtherNet/IP objects and Common Services	192
Section 8	Project	199
8.1	Project structure	199
8.2	Validity of process data	202
8.2.1	Substitute values	205
8.2.2	Evaluation of the valid bits in the user program	206
8.3	Task basics	207
8.3.1	Task properties	208
8.3.2	Task execution	209
8.3.3	Default configuration of tasks	212
8.4	Data exchange between FS resource and ST resource	213

8.5	Data consistency	214
8.5.1	Consistency ranges	215
8.5.2	Influence of scheduling on data consistency	218
8.6	Safety status of blocks	219
8.7	Device naming via PAS4000	221
8.7.1	Device identification data	222
8.7.2	Naming data	223
8.7.3	Device naming for PSS 4000-exclusive devices: PSSu systems	225
8.7.4	Device naming for PSS 4000-operable devices from Pilz	230
8.7.5	Device naming for PSS 4000-operable devices from a third-party manufacturer	233
8.8	Device naming when exchanging devices of PSSu systems	237
8.9	Project download	238
8.10	Sealed FS blocks	243
Section 9	Mode of operation	244
9.1	Communication between PAS4000 and PSS 4000 devices	244
9.1.1	Establish a point-to-point connection	245
9.1.2	Establish a remote connection	246
9.2	Operating states and change in operating status of a PSSu system	247
9.2.1	Flow chart	248
9.2.2	Operating states on the PSSu system	249
9.2.3	Changes in the operating status of a PSSu system	260
9.3	Operating states and change in operating status of the FS resource	261
9.3.1	Flow chart	262
9.3.2	Operating states of the FS resource	263
9.3.3	Change in operating status of the FS resource	270
9.4	Operating states and change in operating status of the ST resource	273
9.4.1	Flow chart	274
9.4.2	Operating states of the ST resource	275
9.4.3	Change in operating status of the ST resource	281
9.5	Operating states and change in operating status of the FS module bus	284
9.5.1	Flow chart	285
9.5.2	Operating states of the FS module bus	286
9.5.3	Change in operating status of the FS module bus	295
9.6	Operating states and change in operating status of the ST module bus	297
9.6.1	Flow chart	298
9.6.2	Operating states of the ST module bus	299
9.6.3	Change in operating status of the ST module bus	308
9.7	Operating states and change in operating status of the FS SafetyNET p RTFN	310
9.7.1	Flow chart	311
9.7.2	Operating states of FS SafetyNET p RTFN	312
9.7.3	Change in operating status of FS SafetyNET p RTFN	317
9.8	Operating states and change in operating status of the ST SafetyNET p RTFN	319
9.8.1	Flow chart	320

9.8.2	Operating states of ST SafetyNET p RTFN	321
9.8.3	Change in operating status of ST SafetyNET p RTFN	325
9.9	Operating states and change in operating status with external communication	327
9.9.1	Flow chart	328
9.9.2	Operating states with external communication	329
9.9.3	Operating status change with external communication	335
9.10	Reset, restart, start and stop	337
9.10.1	Overview	338
9.10.2	Cold start and Reboot command	342
9.10.3	Reset commands	343
9.10.4	Hot start commands	345
9.10.5	Restart commands	346
9.10.6	Stop commands	348
9.11	Functions of the reset button	350
9.11.1	Perform a warm reset for a PSSu system	351
9.11.2	Deliberate operator action for an original reset of the PSSu system	353
9.11.3	Deliberate operator action to transfer the naming data and/or device project	356
9.12	Original reset	361
9.12.1	Effects of an original reset	362
9.12.2	Perform original reset	366
9.13	Enable principle	367
9.14	Reaction times	369
9.14.1	Times that influence reaction times	370
9.14.2	Processing rules during data processing	373
9.14.3	Formation of data paths	374
9.14.4	Maximum reaction times for data subpaths	377
9.15	System time synchronisation	383
9.15.1	System time synchronisation with SNTP	384
Section 10	Commissioning	385
10.1	General requirements	385
10.2	Initial commissioning	386
10.3	Recommissioning	389
Section 11	Maintenance, repair, decommissioning	392
11.1	Maintenance and repair	392
11.2	Decommissioning	393
Section 12	OPC Server	394
12.1	Overview	394
12.1.1	Design options	395
12.1.2	OPC properties	397
12.2	System requirements	398
12.2.1	PC with Microsoft Windows operating system	398
12.2.2	Device with Microsoft Windows Embedded Standard 7	399

12.2.3	Operator terminals PMI 4 from Pilz	401
12.2.4	Operator terminals PMI 5 from Pilz	402
12.3	Licensing	403
12.4	Installation	404
12.4.1	Install PSS 4000 firmware on a PC	405
12.4.2	Install PSS 4000 firmware on a PMI operator terminal	407
12.5	Communication and security settings for PCs	410
12.5.1	DCOM configuration	411
12.5.2	Firewall configuration	444
12.6	Commissioning of the OPC Server	445
12.7	Data type assignment	446
12.8	Namespace	447
12.8.1	General project data	449
12.8.2	Device-based project data	456
Section 13	Diagnostic basics	470
13.1	Diagnostic list	471
13.1.1	Diagnostic messages	472
13.2	Diagnostic log	485
13.2.1	Log entries	486
13.3	Diagnostic server	488
Section 14	Appendix	490
14.1	Factory default settings for port numbers for various services	490
14.2	Factory default settings for the Ethernet parameters	491
Section 15	Glossary	492

1 Introduction

1.1 Validity of the documentation

This documentation is valid for the automation system PSS 4000. It is valid until new documentation is published.

The current documentation is available in the download area of the Pilz homepage (www.pilz.de -> Downloads).

This document often uses the term "PSS 4000" as an abbreviation of "Automation system PSS 4000".

1.2 Overview of documentation

1 Introduction to the Documentation

The introduction is designed to familiarise you with the contents, structure and specific order of this system description.

2 Automation System PSS 4000

This chapter contains an introduction to the automation system PSS 4000.

3 Safety

This chapter must be read as it contains important information on safety regulations.

4 PSSu System Basics

This chapter contains essential information on the design and power supply of a PSSu system, including the system limits and factory-set defaults.

5 Ethernet Interface

This chapter contains essential information on the Ethernet interface on PSS 4000 devices.

6 SafetyNET p

This chapter contains an introduction to SafetyNET p, usable components and network subscribers.

7 External Communication

This chapter contains information on data exchange via Ethernet-based communication modes such as Modbus/TCP and on data exchange via fieldbus systems such as PROFIBUS-DP Slave.

8 Project

This chapter contains essential information on the programming model, device naming, project download and access protection for blocks.

9 Mode of Operation

This chapter explains the operating states of a PSSu system, options for user intervention, special procedures and modes of operation.

10 Commissioning

This chapter describes the requirements during commissioning.

11 Maintenance and Repair, Decommissioning

This chapter describes the requirements during maintenance, repair and decommissioning.

12 OPC Server

This chapter describes the Pilz PSS 4000 OPC Server.

13 Diagnostics

This chapter contains an overview and introduction to diagnostics on the PSS 4000.

14 Appendix

The appendix documents various factory-set defaults.

15 Glossary

1.3 Definition of symbols

Information that is particularly important is identified as follows:



DANGER!

This warning must be heeded! It warns of a hazardous situation that poses an immediate threat of serious injury and death and indicates preventive measures that can be taken.



WARNING!

This warning must be heeded! It warns of a hazardous situation that could lead to serious injury and death and indicates preventive measures that can be taken.



ATTENTION!

This refers to a hazard that can lead to a less serious or minor injury plus material damage, and also provides information on preventive measures that can be taken.



CAUTION!

This describes a situation in which the product or devices could be damaged and also provides information on preventive measures that can be taken. It also highlights areas within the text that are of particular importance.



Information

This gives advice on applications and provides information on special features.



In Preparation

In development

This function or description is not yet available.
It will not concern safety-related functions or descriptions.

2 Automation system PSS 4000

2.1 Introduction to PSS 4000

2.1.1 General

PSS 4000 is an automation system that can be used to tackle the widest range of automation functions. These include requirements from the fields of control technology (PLC), motion control and visualisation.

PSS 4000 can be used at every stage of automation, ranging from a small, decentralised automation terminal to a complex network at control level.

The automation system PSS 4000 handles non-safety-related functions (ST functions) as well as safety-related functions (FS functions) for control, motion control and visualisation tasks.

2.1.2 SafetyNET p

SafetyNET p is used for networked automation functions. SafetyNET p is an Ethernet-based multi-master bus system suitable for industrial use.

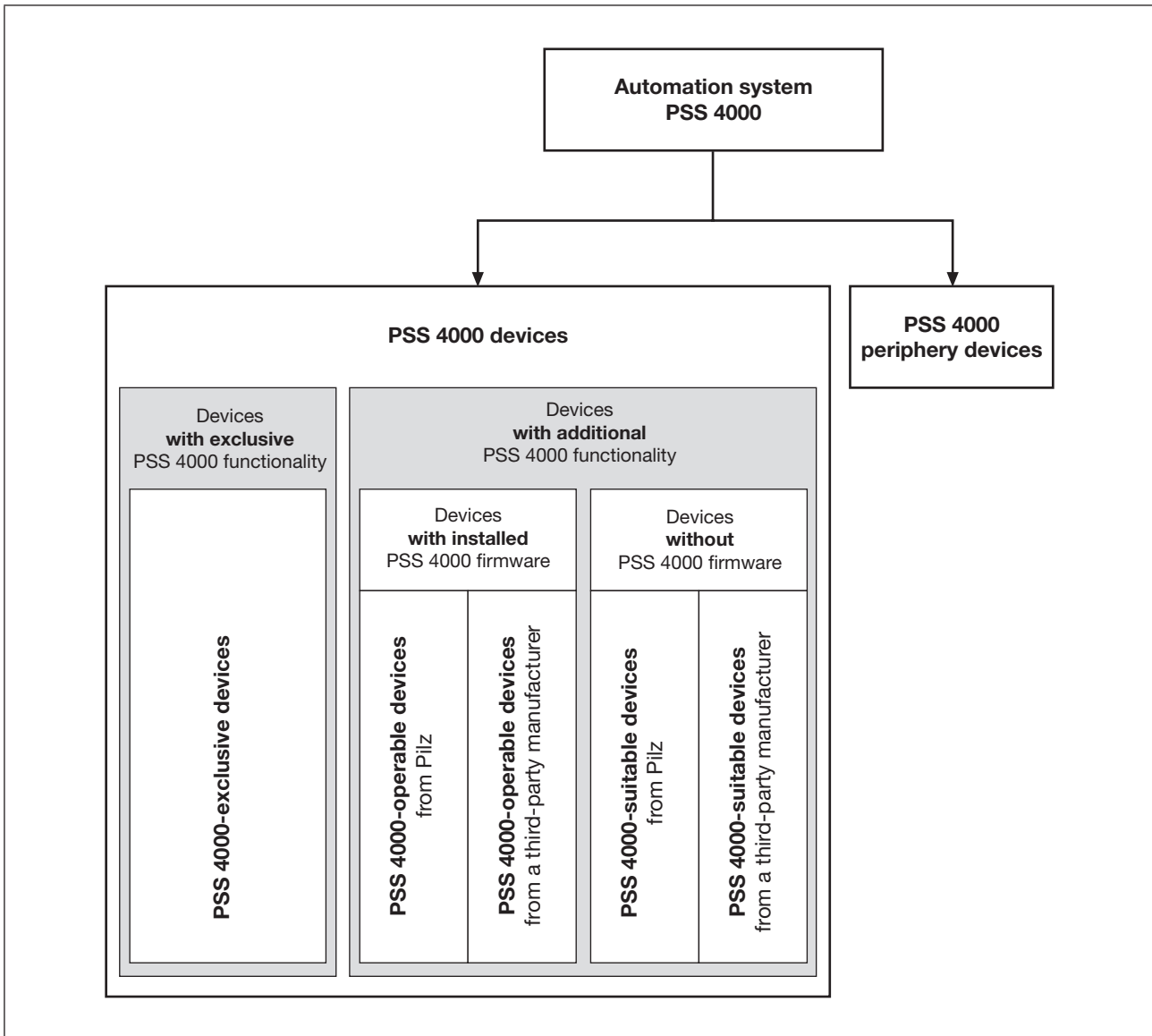
The SafetyNET p protocol contains a safe data channel for safety-related data. This channel is certified for data transfer in accordance with SIL 3 of IEC 61508. On SafetyNET p, safety and non-safety-related data is transferred via different data channels, but using the same connection cable.

In terms of the topology, SafetyNET p supports various familiar Ethernet architectures. So modular system architectures and individual subnetworks can be implemented. If necessary, this method can be used to divide automation functions into decentralised and largely independent function units. The main benefits of this type of architecture lie in the reusability, modification and availability of function units.

2.1.3 Devices in the automation system PSS 4000

2.1.3.1 System overview

Devices in the automation system PSS 4000 are divided into **PSS 4000 devices** and **PSS 4000 periphery devices**. The diagram below provides an overview of how the devices are divided and subdivided:



Devices of the automation system PSS 4000

PSS 4000 devices

PSS 4000 devices are devices on which the PSS 4000 firmware is installed or on which the PSS 4000 firmware can be installed. As such, these devices have the basic properties of the automation system PSS 4000. The most important basic properties include:

- ▶ Ability to find the devices via PAS4000 using "Scan PSS 4000 Network"
- ▶ Incorporation into the PSS 4000 diagnostic system

- ▶ Respective configuration obtained via the PSS 4000 project download
- ▶ Respective device project obtained via the PSS 4000 project download

A further subdivision of the PSS 4000 devices is as follows:

- ▶ Devices which exclusively have PSS 4000 functionality (e.g. PSSu system head modules) are called **PSS 4000-exclusive devices**.
- ▶ Devices which have PSS 4000 functionality as an additional feature are subdivided as follows:
 - **PSS 4000-operable devices**

These are devices on which the PSS 4000 firmware is installed. They may be Pilz devices or devices from a third party manufacturer. If they are Pilz devices, they will be called "PSS 4000-operable devices from Pilz", otherwise they will be called "PSS 4000-operable devices from a third-party manufacturer".
 - **PSS 4000-suitable devices**

These are devices on which the PSS 4000 firmware is not installed, (e.g. PMI operator terminals, PC), but on which the PSS 4000 firmware can be installed. They may be Pilz devices or devices from a third party manufacturer. If they are Pilz devices, they will be called "PSS 4000-suitable devices from Pilz", otherwise they will be called "PSS 4000-suitable devices from a third-party manufacturer".

If the PSS 4000 firmware is installed on a PSS 4000-suitable device, that device then becomes a PSS 4000-operable device.

PSS 4000 periphery devices

PSS 4000 periphery devices are devices that can be used with the automation system PSS 4000. They are subdivided as follows:

- ▶ Devices on which the PSS 4000 firmware cannot be installed (e.g. sensors, devices for external communication such as PROFIBUS-DP Master)
- ▶ Devices on which the PSS 4000 firmware can be installed, but is not installed (e.g. PSS 4000-suitable devices such as PMI operator terminals with exclusive PSS 4000 Diag Control)

2.1.3.2 System sections

A system section is a defined unit of functionalities, which provides utilities on a device, which are visible to the user. To ensure that a system section can run on a PSS 4000-exclusive device or PSS 4000-operable device, the corresponding hardware requirements must be met.

The system sections that are available on a PSS 4000-exclusive device or PSS 4000-operable device depend on the device. Details of the PSS 4000-exclusive devices and PSS 4000-operable devices from Pilz are available in the operating manual for the relevant device.

Overview of available system sections

- ▶ Resources
 - FS resource system section
Execution unit for safety-related control programs
 - ST resource system section
Execution unit for non-safety-related control programs
- ▶ Module bus
 - FS module bus system section
Unit for data exchange between FS electronic modules/FS compact modules and other system sections on the same device
 - ST module bus system section
Unit for data exchange between ST electronic modules/ST compact modules and other system sections on the same device
- ▶ SafetyNET p RTFN
 - FS SafetyNET p RTFN system section
Unit for transporting FS-I/O data via SafetyNET p RTFN
 - ST SafetyNET p RTFN system section
Unit for transporting ST-I/O-data via SafetyNET p RTFN
- ▶ External communication
 - IP connections system section
Unit for transporting external I/O data via IP connections (e.g. Modbus/TCP, Raw UDP)
 - PROFIBUS-DP Slave system section
Unit for transporting external I/O data as PROFIBUS-DP Slave
 - PROFIBUS-DP Slave with PROFIsafe system section
Unit for transporting non-safety-related external I/O data as PROFIBUS-DP Slave and safety-related external I/O data as PROFIsafe F-device
 - EtherNet/IP Adapter system section
Unit for transporting non-safety-related external I/O data as EtherNet/IP Adapter

- ▶ Server
 - Diagnostic Server system section
Unit for providing diagnostic data
 - OPC Server system section
Unit with standardised software interface for providing visualisation data

Overview of the system sections of PSS 4000-exclusive/operable devices

Product	FS resource	ST resource	FS module bus	ST module bus	FS SafetyNET p RTFN	ST SafetyNET p RTFN	IP connections	PROFIBUS-DP Slave	PROFIBUS-DP Slave with PROFIsafe	EtherNet/IP Adapter	Diagnostic Server	OPC Server
PSSu H PLC1 FS DP SN SD	◆	◆	◆	◆	◆	◆	◆	◆	---	---	---	---
PSSu H PLC1 FS SN SD	◆	◆	◆	◆	◆	◆	◆	---	---	◆	---	---
PSSu H m F DP SN SD	◆	---	◆	◆	◆	◆	◆	◆	---	---	---	---
PSSu H m F DPsafe SN SD	◆	---	◆	◆	◆	◆	◆	---	◆	---	---	---
PSSu H m F DP ETH SD	◆	---	◆	◆	---	---	◆	◆	---	---	---	---
PSSu H FS SN SD	---	---	◆	◆	◆	◆	---	---	---	---	---	---
PSS 4000 firmware for PMI 4 series	---	---	---	---	---	---	---	---	---	---	◆	◆
PSS 4000 firmware for PMI 5 series	---	---	---	---	---	---	---	---	---	---	◆	◆
PSS 4000 firmware for PC	---	---	---	---	---	---	---	---	---	---	◆	◆

2.1.3.3 PSS 4000-exclusive devices: PSSu systems

PSSu systems are PSS 4000-exclusive devices. The system sections that are available on a PSSu system depend on the device (see System sections, [Page 16](#)). Details can also be found in the operating manual of the relevant device.

A PSSu system is assembled from a variety of modules, to suit the functions required. The modular structure makes it easier to react to modifications and adjustments. With a PSSu system, not only is it possible to adapt the scope retrospectively (e.g. extend the number of I/Os, extend/modify the I/O technology, such as using relay outputs, adding I/O functions such as analogue evaluation, counter functions, temperature evaluation), it's also possible to convert a PSSu system with a lower performance rating (e.g. decentralised system PS-Suniversal I/O) into a PSSu system with a higher performance rating (e.g. control system PSSuniversal PLC).

A PSSu system is configured in the PSSu Module Editor of the PAS4000.

2.1.3.4 PSS 4000-operable devices from Pilz: PMI

Various Pilz PMI operator terminals can be made into "PSS 4000-operable devices from Pilz" by installing the PSS 4000 firmware (see System requirements, [Page 398](#)). When the PSS 4000 firmware is installed, the PMI obtains various system sections, as e.g the OPC Server and Diagnostic Server (see Overview of the system sections of PSS 4000-exclusive/operable devices, [Page 17](#)).

2.1.3.5 **PSS 4000-operable devices from a third-party manufacturer: PC**

A PC with the corresponding system requirements can be made into a "PSS 4000-operable device from a third-party manufacturer" by installing the PSS 4000 firmware (see System requirements, [Page 398](#)). In this case the PC will also be known as a PSS 4000-PC. When the PSS 4000 firmware is installed, the PC obtains e.g. the OPC Server system section and the Diagnostic Server system section (see System sections available, [Page 17](#)).

2.1.3.6 PSS 4000-suitable devices

A PSS 4000-suitable device must meet certain system requirements in order for the PSS 4000 firmware to be installed. These system requirements are defined through the PSS 4000 firmware that is to be installed on the PSS 4000-suitable device.

Example for PSS 4000 OPC Server

- ▶ The PSS 4000 OPC Server can be installed on a PC with various Microsoft Windows operating systems (see System requirements, [Page 398](#)).
- ▶ The PSS 4000 OPC Server can be installed on various generations of Pilz PMI operator terminals (see System requirements, [Page 398](#)).

2.1.4 PAS4000

Design, configuration, programming, commissioning and diagnostics take place centrally via PAS4000 (Pilz Automation Suite 4000).

PAS4000 is a collection of various tools which can be used to create and edit a project, e.g.:

- ▶ PSSu Module Editor

The required PSSu systems are assembled and configured in the PSSu Module Editor.

- ▶ I/O Mapping Editor

The connection between the user program and hardware is established in the I/O Mapping Editor.

- ▶ IL Editor

In the IL Editor, the user program is programmed in the instruction list (IL) programming language.

2.1.5 Diagnostics

Effective diagnostics are required in order to keep machine downtimes to a minimum. Diagnostics must provide all the information necessary to rectify faults quickly, prevent faults and analyse machine conditions.

The PSS 4000 distinguishes between system diagnostics and process diagnostics.

- ▶ System diagnostics

Diagnostics on hardware and firmware, including faults in the hardware configuration or user program.

The system diagnostics are provided by Pilz in full. Only a few adjustments can be made by the user, e.g. entering additional information for location information.

- ▶ Process diagnostics

Diagnostics on procedures and conditions. The process diagnostics are created by the user within the user program. Pilz component blocks are supplied with process diagnostics, which users can adapt to suit their own requirements.

A diagnostic server is required if you wish to display a project's diagnostic data (see Diagnostic Server, [Page 488](#)). A project may only contain one Diagnostic Server.

If the project contains an OPC Server (PSS 4000 OPC Server), the PSS 4000 device on which the OPC Server is installed will provide the Diagnostic Server. If there is no OPC Server, PAS4000 will start up its own Diagnostic Server if the diagnostic list or diagnostic log is to be displayed in PAS4000.

2.1.6 OPC Server

OPC is a standardised software interface, through which PC-based software from a range of manufacturers can exchange data. The OPC Foundation is responsible for maintaining and propagating the standard.

On networked plants, an OPC Server will typically collect diagnostic data, process data etc. and display it in the defined namespace. OPC Clients access and process the data provided by the OPC Server for process visualisation purposes and/or to record operating data.

In the PSS 4000 automation system a PC or a PMI operator terminal obtains the OPC Server system section as part of the installation of PSS 4000 firmware. By installing the PSS 4000 Firmware a PC becomes a "PSS 4000-operable device from a third-party manufacturer" and a PMI operator terminal PMI becomes a "PSS 4000-operable device from Pilz" (see System Overview, [Page 14](#)).

The following boundary conditions apply to the OPC Server:

- ▶ The OPC Server supports various OPC properties for the access of OPC Clients (see OPC properties of the OPC Server, [Page 397](#)).
- ▶ The OPC Server can be installed on a PC or Pilz PMI operator terminal (see System requirements, [Page 398](#)).
- ▶ An OPC Server can collect the data of exactly one project that has been created with PAS4000 and provide it for the OPC Clients.
- ▶ Each project created with PAS4000 may contain precisely one OPC Server.
- ▶ It is possible to access the OPC Server with more than one Client.
- ▶ All project-based settings for the OPC Server are made in PAS4000 and are transferred to the OPC Server as an OPC device project when the project is downloaded.

2.2 Modularity with PSS 4000

2.2.1 Centralised view of a networked system

PSS 4000 combines the benefits of a centralised control system at the highest automation level with the benefits of small, decentralised control units.

- ▶ Benefits at the highest automation level:
 - Software tool PAS4000, which can be used from a central location
 - Centralised configuration, even of decentralised PSSu systems
 - User program that has been created centrally can be distributed to decentralised PSSu systems
 - Rapid overview of the entire plant during troubleshooting, commissioning/re-commissioning and modification
- ▶ Benefits at the lowest automation level:
 - Decentralised function units can be commissioned independently
 - Increased availability (no “Single Point of Failure”)

2.2.2 Scalability of the devices

The term “scalability” is used for a device when it has a certain degree of adaptability to changing requirements.

A distinction can be made between “horizontal” and “vertical” scalability:

▶ Horizontal scaling

Functions are distributed to other or additional devices. On PSS 4000 for example, horizontal scaling can be used to:

- Distribute the control power to various PSSu systems on a networked plant, to increase the processing power, for example
- Divide the user program into independent subfunctions in order to distribute it to the project devices
- Divide the project into reusable parts

▶ Vertical scaling

A device or parts of a device are replaced by a more powerful alternative. On PSS 4000 for example, vertical scaling can be used to:

- Adapt the performance class of a PSSu system retrospectively by using a head module of a different performance class
- Expand and adapt a PSSu system retrospectively using I/O modules

2.2.3 Modularisation within programming

On PSS 4000, control functions can be distributed across an entire plant. As a result, the plant architecture can be modularised in line with the task assignment. This modularisation applies not only to the modular structure of the PSSu systems and network but also to the user program. With component-oriented programming, the plant can be completely divided into functionalities. The aim is to achieve the highest possible level of standardisation and reusability while maintaining maximum flexibility and adaptability.

The following can be achieved with a modularised user program on PSS 4000 for example:

- ▶ Modifications will only impact locally
- ▶ Control functions can be standardised

A user program can be adopted for all functions with an identical structure; and once a modification has been made, it can be transferred to all the relevant PSSu systems.

- ▶ Availability is increased, as error reactions are restricted locally

3 Safety

3.1 General guidelines

Please refer to the safety guidelines in the operating manual for the products you are using, and in the “Safety Manual PSS 4000”. The safety manual also includes check lists designed to help you with the safety-related planning, construction and operation of a plant.

You should also refer to the information provided

- ▶ In the PSSuniversal Installation Manual
- ▶ In PAS4000’s online help.

The standards current at the time of going to print shall apply.

3.2 Intended use

PSSu systems of all performance classes are designed for use in an industrial environment, as follows:

- ▶ Safety-related applications via the FS resource of a control system PSSu PLC/PSSu multi, including connection to SafetyNET p for decentralised tasks.
- ▶ Non-safety-related applications via the ST resource of a control system PSSu PLC, including connection to SafetyNET p and/or other standard bus systems for decentralised tasks.

PSSu systems of all performance classes are primarily suitable for use in machine safety circuits in which a safe condition is brought about by the removal of power.

Examples:

- ▶ Presses
- ▶ Transfer lines
- ▶ Tank storage facilities
- ▶ E-STOP functions
- ▶ Burner control systems
- ▶ Cable cars/traction operations
- ▶ Stage technology

The following is deemed improper use in particular:

- ▶ Any component, technical or electrical modification to a product
- ▶ Use of a product outside the areas described in the product documentation
- ▶ Any use that is not in accordance with the documented technical details.

Foreseeable misuse and abuse

- ▶ Without additional measures, the automation system PSS 4000 is not suitable for use in areas with increased environmental requirements (e.g. potentially explosive areas).
- ▶ The automation system PSS 4000 is not suitable for plants in which the removal of power does not lead to a safe condition.
- ▶ On passenger transportation systems, appropriate evacuation measures must be made or self-contained rescue equipment installed, should the plant come to a standstill due to a fault.
- ▶ Where locked zones have an electrical release, emergency release devices should be provided.
- ▶ Safety devices may not be overridden until appropriate measures of equal value are put in place (e.g. muting function). Measures of equal value may only be used if the preceding safety assessment permits it.

Examples:

- The safety gates may be open in setup mode provided there is a muting function, which is triggered by the operating mode selector switch.
- A light curtain can be interrupted by transported materials provided there is a muting function with the corresponding components (e.g. muting sensors including control and muting monitoring of the muting sensors via a muting block in the user program, for example).

3.2.1 Safety Integrity Level (SIL) and Performance Level (PL)

PSSu systems are safety components in accordance with Annex V of the Machinery Directive 2006/42/EC. Depending on the application area and its respective regulations, PSSu systems of all performance classes can be used up to SIL 3 of EN 62061 and up to PL e (Cat. 4) of EN ISO 13849-1. The relevant technical details for a risk assessment can be found in the respective operating manual.

SafetyNET p is used for networked automation functions. This safe communication was developed in accordance with relevant standards, such as EN IEC 61508. Independent certification bodies such as BG have confirmed the security of the concept and the suitability of SafetyNET p in safety-related applications up to PL e (Cat. 4) of EN ISO 13849-1 or SIL 3 of EN/IEC 62061. The relevant technical details for the modules, as required for the risk assessment, can be found in the operating manual for the respective module.

3.2.2 General safety assessments

Before using a PSSu system it is necessary to perform a safety assessment in accordance with the Machinery Directive. A PSSu system as an individual component is a safety-related system for the purposes of EN/IEC 61508. It guarantees functional safety against hardware and firmware errors, for example. However, it does not guarantee the safety of the overall process and design, nor of the project.

The user is responsible for the safety of the project. Pay special attention when programming and observe local standards and regulations.

A faulty user program can jeopardize the safety of the entire process!

Define the safety requirements for the entire plant, for all phases of the service life and the entire safety lifecycle, and also define how these will be implemented from a technical and organisational standpoint.

3.3 Use of qualified personnel

The products may only be assembled, installed, programmed, commissioned, operated, maintained and decommissioned by competent persons.

A competent person is someone who, because of their training, experience and current professional activity, has the specialist knowledge required to test, assess and operate the work equipment, devices, systems, plant and machinery in accordance with the general standards and guidelines for safety technology.

It is the company's responsibility only to employ personnel who:

- ▶ Are familiar with the basic regulations concerning health and safety / accident prevention
- ▶ Have read and understood the safety guidelines given in this description
- ▶ Have a good knowledge of the generic and specialist standards applicable to the specific application.

3.4 Failsafe inputs/outputs

In FS applications, it is essential that short circuits and open circuits are unable to cause a hazardous condition within a plant/machine. The way in which this is done will depend on the degree of hazard from the plant/machine, the switching frequency of the sensors and the level of safety of the sensors and actuators. These points must be assessed in conjunction with the certification body responsible for safety (e.g. BG, TÜV).

Various tests may be configured on FS input/output modules. These tests can be used to detect many short circuits and open circuits. Examples of such tests are:

- ▶ Off tests for outputs that are switched on
- ▶ On tests for outputs that are switched off
- ▶ Tests for shorts between the outputs
- ▶ Test pulses for detecting shorts between contacts on inputs

Wiring guidelines and connection examples are available in the operating manuals or data sheets for the input/output modules.

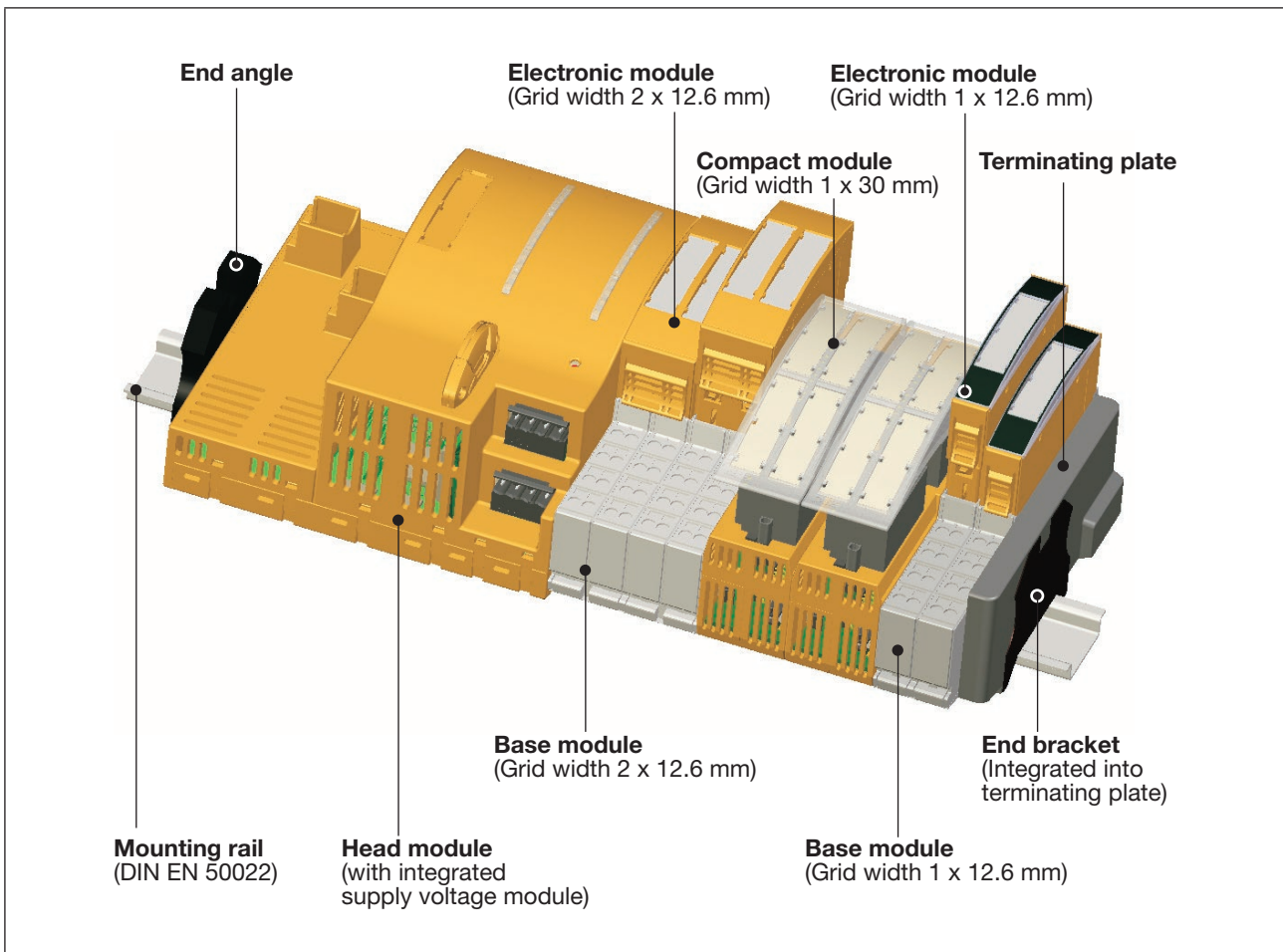
4 PSSu system basics

PSSu systems are PSS 4000-exclusive devices. The system sections that are available on a PSSu system depend on the device (see System sections, [Page 16](#)). Details can also be found in the operating manual of the relevant device.

4.1 Structure of a PSSu system

A PSSu system consists of:

- ▶ A head module with integrated supply voltage
- ▶ Input/output modules for standard and failsafe applications
- ▶ Supply voltage modules (only if necessary)
- ▶ End bracket to secure the system at the start
- ▶ Terminating plate with integrated bus terminating resistors and end bracket to secure the system at the end



Head module

The head module co-ordinates a PSSu system's entire data traffic and determines its performance class. There are three different performance classes:

- ▶ Decentralised system PSSu I/O
 - PSSu system without control functionality
 - Consists of head module, electronic modules and/or compact modules
 - I/Os are controlled via SafetyNET p by means of a control system PSSu PLC/PSSu multi

- ▶ Control system PSSu multi
 - PSSu system with the functionality of a compact safety control system
 - Consists of head module, electronic modules and/or compact modules
 - FS resource available
 - One task only can be configured
 - Programmed via the PAS 4000 Multi Editor
- ▶ Control system PSSu PLC
 - PSSu system to control automation plants
 - Consists of head module, electronic modules and/or compact modules; however, the electronic modules/compact modules are not needed if the PSSu system is used exclusively to control decentralised I/Os with SafetyNET p
 - Control of I/Os from decentralised PSSu systems of all performance classes when networking with SafetyNET p
 - ST resource and FS resource available
 - Max. of 9 tasks can be configured per resource
 - Programmed via the PAS4000 Multi Editor or programmed in PAS4000 in accordance with IEC 61131

Input/output modules

Input/output modules are available in different designs:

- ▶ Electronic module and base module
- ▶ Compact module

The input/output modules are available for a wide range of input/output functions.

Supply voltage modules

The head module provides the supply voltage for the remaining modules. Additional power supplies may also be required (e.g. potential isolation, voltage refresh).

Supply voltage modules consist of an electronic module and base module.

Designs

Electronic modules and base modules:

The electronic modules determine the function of the supply voltage or input/output module.

- ▶ Electronic modules
 - Are plugged into the base modules
 - Communicate with the head module via the module bus

The base modules are the carrier units for the electronic modules.

- ▶ Base modules
 - Are used to connect the field wiring
 - Are available with the following connection types:
Cage clamp terminals or screw terminals

Compact modules:

The compact modules combine the function unit (inputs and/or outputs) and connection levels in one housing. Wiring is via multi-pin connectors with spring-loaded terminals, which are plugged into the connector strips on the module.

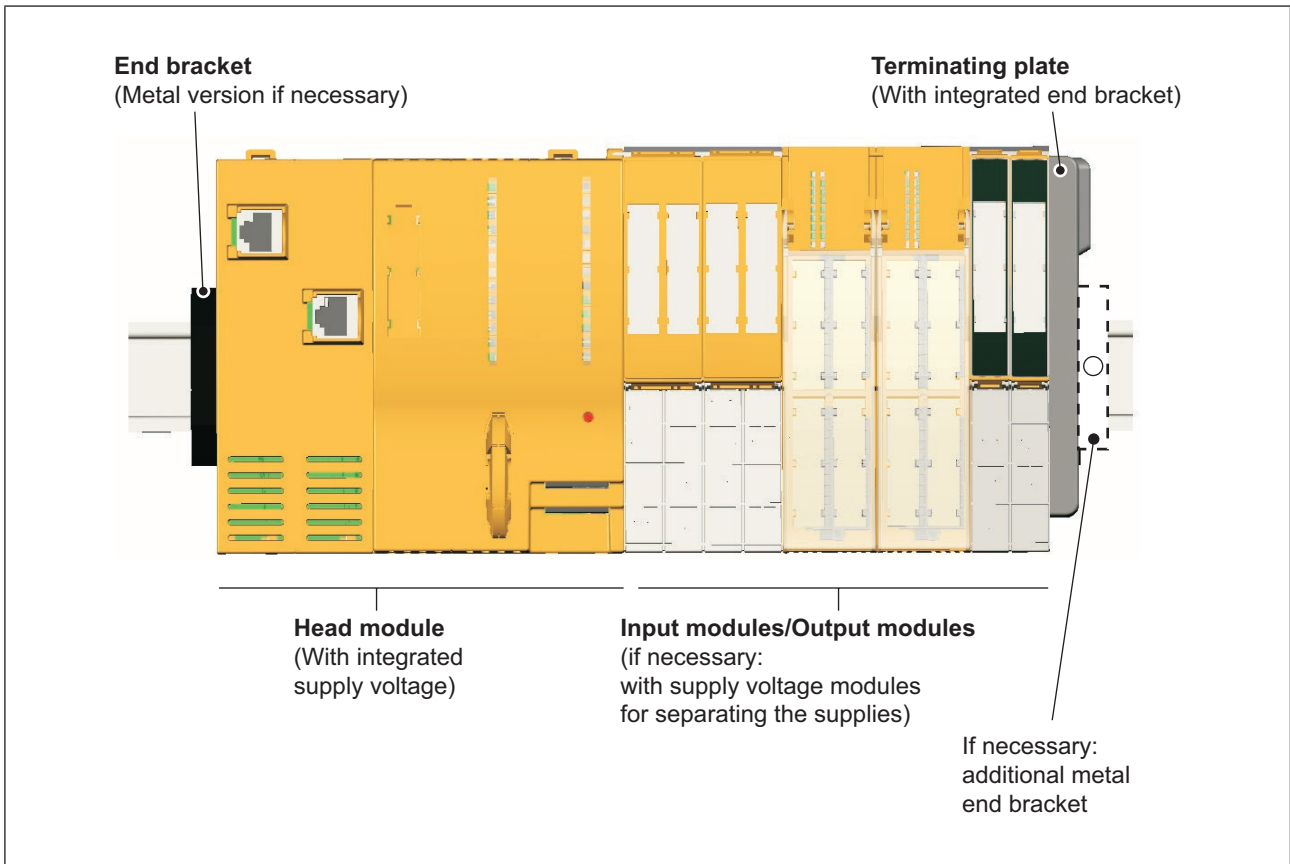
Compact modules

- ▶ Do not need base modules
- ▶ Are used to connect the field wiring
- ▶ Communicate with the head module via the module bus

4.2 Module layout

- ▶ The first module on a PSSu system is always a head module. The supply voltage is integrated within the head module:
 - Module supply (to supply the head module and input/output modules)
 - Periphery supply (to supply the sensors and actuators)
 - The supply voltage is already integrated on the InduraNET p head module. There is no need for a supply voltage module.
- ▶ Input/output modules for FS and ST applications may follow on to the right:
 - The input/output modules may be installed in any order.
 - Input/output modules for FS applications and ST applications may be combined as required.
 - If modules of the same function are combined into groups it makes things clearer and simplifies the wiring.
 - Base modules with screw terminals cannot be mixed in one PSSu system with base modules with cage clamp terminals.
 - A base module with screw terminals may not be installed to the left of a compact module.
 - The maximum number of input/output modules is determined by the system limits.
 - Input/output modules for FS applications may not be used with head modules that are only designed for ST applications.
- ▶ Additional supply voltage modules may be required in order to refresh the module supply or periphery supply.
- ▶ Supply groups require an additional voltage supply module at the start of each group. The modules belonging to the supply group will then follow to the right.
- ▶ The final element in a PSSu system is always a terminating plate containing the terminating resistors for the module bus.
- ▶ The system is attached to the mounting rail using fastening elements at the start and end of the system. The following end brackets are available, depending on vibration and shock stress:
 - Plastic version (standard)
 - Metal version (where there is increased stress)

Layout of the fastening elements



4.3 Module descriptions

The module descriptions provide information on their function. The descriptions are made up of multi-stage combinations of letters and numbers. All module names begin with **PSSu**. This is followed by:

- ▶ **H** for head modules
e. g.: PSSu **H** PLC1 FS DP SN SD
- ▶ **E** for electronic modules, then:
 - **S** for standard module,
e.g.: PSSu **E S** 4DI
 - **F** for failsafe module,
e. g.: PSSu **E F** 4DI
- ▶ **K** for compact modules, then:
 - **S** for standard module,
e.g.: PSSu **K S** 16DI
 - **F** for failsafe module,
e. g.: PSSu **K F** 16DI
- ▶ **B** for base modules,
e.g.: PSSu **BP** 1/8S

Base modules	Basic function	Size	Connection type	Additional function
PSSu B...	PSSu BP... – Base module for periphery (input/output)	PSSu BP-C 1/8... – 1 x grid width, 8 connections	PSSu BP-C 1/8C – Cage clamp terminals	PSSu BP 1/8S- J – Integrated cold junction compensation
	PSSu BP-C... – Base module for periphery with C-rail	PSSu BP-C 1/12... – 1 x grid width, 12 connections	PSSu BP-C 1/8S – Screw terminals	
	PSSu BS... – Base module for power supply	PSSu BP-C 2/16... – 2 x grid width, 16 connections		
	PSSu BS-R... – Base module for refreshing the power supply	PSSu BP-C 2/8... – 2 x grid width, 8 connections		

Link modules have individual letter codes, depending on their application, e. g.:

- ▶ PSSu **WB** S IDN, PSSu **WR** S IDN (for wireless communication with InduraNET p)
- ▶ PSSu **XB** F-T, PSSu **XR** F-T (for cable-based subdivision)

Accessories for the PSSu system are identified by **PSSu A...** (A for "Accessory").

PSSu modules are available as different product types:

- ▶ Base type
The basic versions of PSSu modules are called base type modules (e.g. PSSu E F DI OZ 2).

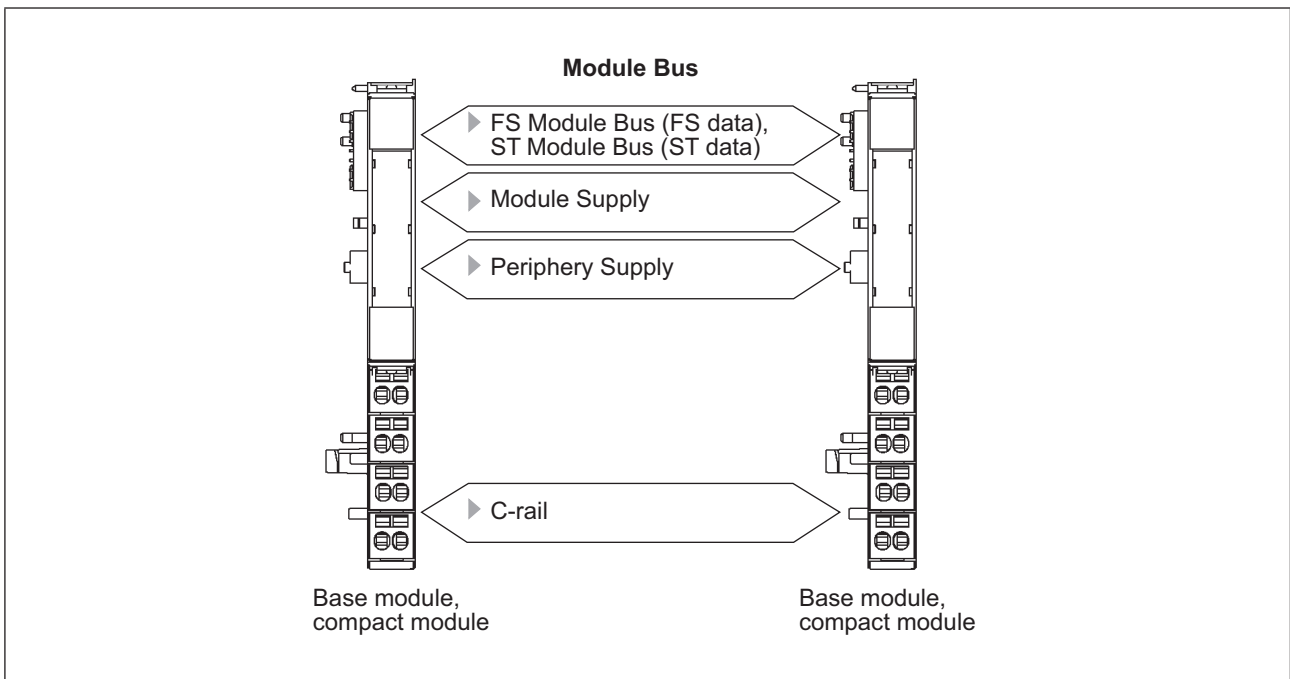
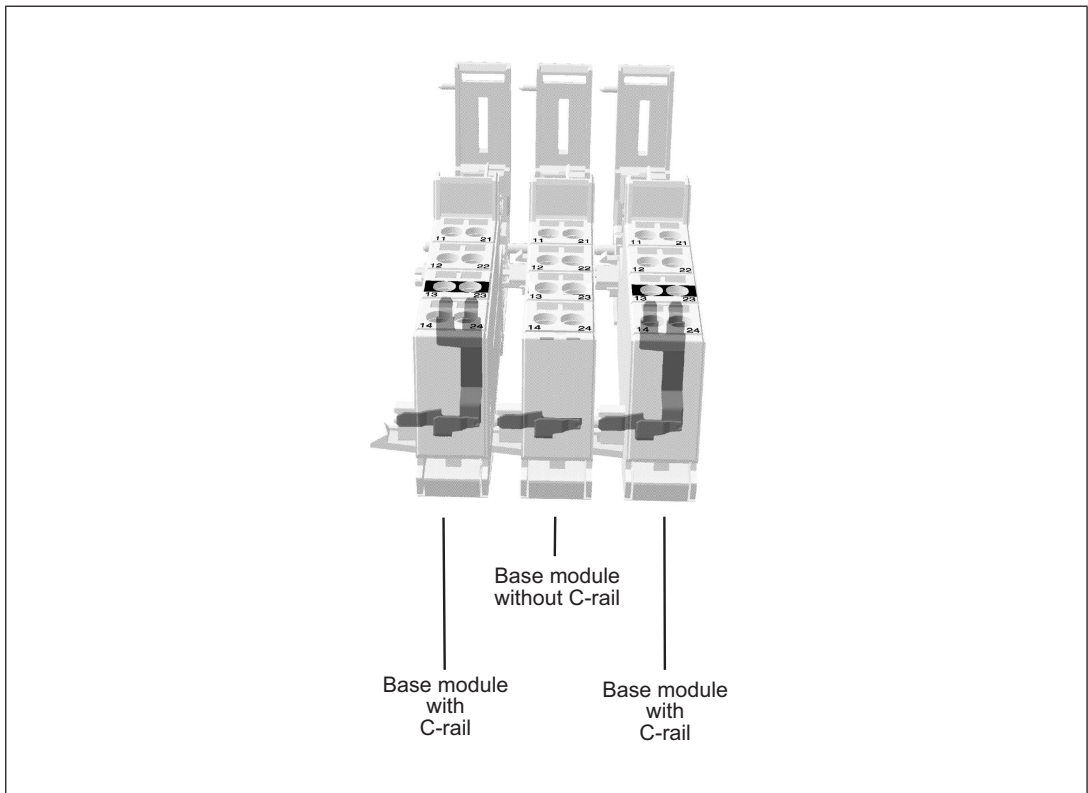
- ▶ **T-type**
PSSu modules that are suitable for use where there are increased environmental requirements on temperature and humidity are called T-type modules. Their functionality is no different from that of the base type modules. They are identified by a "**-T**" at the end of the product name, e.g.: PSSu E F DI OZ 2-**T**.
- ▶ **R-type**
PSSu modules that are specifically designed for railway applications and have railway approval are called R-type modules. They are identified by an "**-R**" in the product name (e.g. PSSu E F DI OZ 2-**R**).

4.4 Module bus (MBUS)

The module bus supplies the modules of a PSSu system with voltage and transfers data between the head module and the electronic/compact modules. The module bus is formed by arranging the base modules/compact modules together and connecting them via a mechanical latch.

The module bus is structured as follows:

- ▶ **Data bus**
two separate bus systems for failsafe data (FS module bus) and standard data (ST module bus).
- ▶ **Supply voltage for:**
 - **Module Supply**
The module supply is the internal supply voltage for the head module and electronic modules.
 - **Periphery Supply**
The periphery supply supplies 24 VDC to the sensors and actuators on digital input/outputs on electronic modules. On compact modules, the supply for the inputs is generated from the periphery supply; the supply for the outputs must be fed to the module terminals. The connection on the module bus can be disconnected in order to form supply groups.
- ▶ **C-rail ("cross connection")**
The C-rail is an additional, free power rail. Access to the C-rail is available via base modules that contain the letter "**-C**" in their description (e.g. PSSu BP-**C** 1/8 S, PSSu BP-**C** 1/8 C).
The C-rail supply must belong to the same circuit as the other base module connections. Connecting PE to the C-rail provides a simple form of shield. On the compact modules, the C-rail is not looped through and is not available at the terminals.



4.5 Connections on the base modules

The connections on the base modules are divided into connection levels and connection columns. The connections have a two-digit number.

- ▶ The first digit denotes a base module's connection column (e.g.: connection **23** is in the second column).
- ▶ The second digit denotes a base module's connection level (e.g.: connection **23** is in the third level).

The function of the base module connections depends on the electronic module.

4.5.1 Connection levels on input/output modules

On input/output modules, the connection levels are typically arranged as follows:

- ▶ Connection level 1 and 4
Input and output connections
- ▶ Connection levels 2 and 3
Connections for the common supplies (periphery supply, analogue inputs/outputs or screening)

Many input/output modules may have two more connection levels:

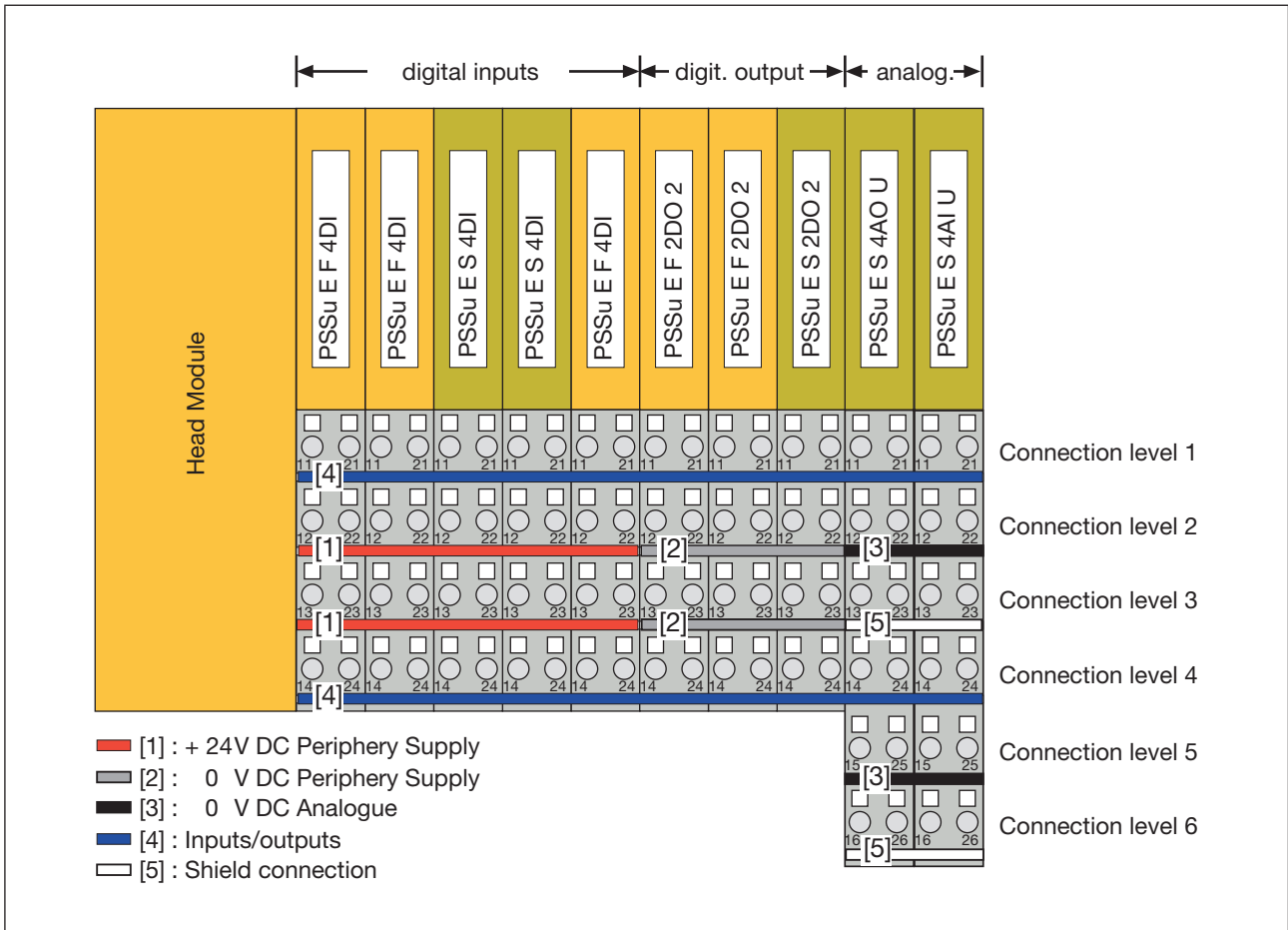
- ▶ Connection level 5 and 6
Connections for the common supplies (periphery supply, analogue inputs/outputs or screening)

Input/output modules can be installed in any order. However, for the connection diagram to be consistent it makes sense to arrange input/output modules of the same type into groups and not have them interrupted by compact modules (see example).

4.5.2 Colour marking on the connection levels

The colour marking on the connection levels is a wiring aid. Various colour markers are available for labelling. The colour markers can be used to label different supplies. This way the connection levels remain clearly visible, even after wiring.

Example:



4.6 Connections on the compact modules

The connections on the compact modules are divided into terminal rows. Single or three-row connectors with spring-loaded terminals can be plugged into the device's single-row 10-pin connector strip.

The function of the connections depends on the compact module.

▶ Terminal row 1

Connection of the inputs and outputs, periphery supply (24 V and 0 V).

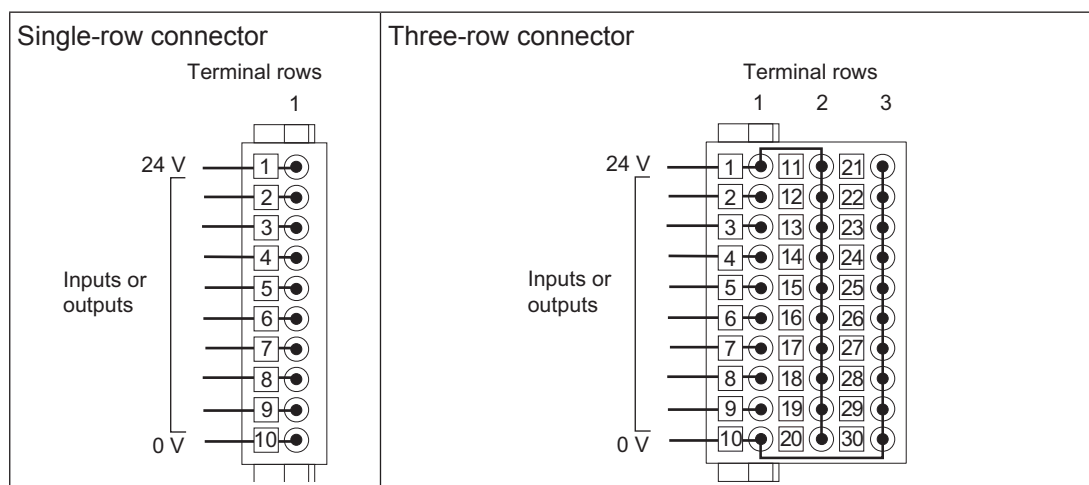
▶ Terminal row 2

All connections are linked; bridged to the 24 V periphery supply connection from the first connection level

▶ Terminal row 3

All connections are linked; bridged to the 0 V periphery supply connection from the first connection level

Input/output modules can be installed in any order. However, for the connection diagram to be consistent it makes sense to arrange input/output modules of the same type into groups.



4.7 Maximum system expansion of a PSSu system

The PSSu Module Editor on PAS4000 provides support in assembling a PSSu system. The tool checks the system limits as the data is entered.

The expansion of a PSSu system is limited by the following values

- ▶ Number of modules
- ▶ Number of FS modules
- ▶ Number of ST modules
- ▶ Number of failsafe inputs/outputs
- ▶ Number of standard inputs/outputs
- ▶ Current load of module bus

The system limits are listed and explained below.

4.7.1 Number of modules and number of inputs/outputs on a PSSu system

The following system limits apply:

Module type	Max. number per PSSu system
- - -	Total of 64 modules
ST modules	64 modules
FS modules	64 modules
PSSu E F DIOZ 2	16 modules
PSSu E F BSW	24 modules
ST bit modules	256 ST input bits
	240 ST output bits
FS bit modules	256 FS input bits
	256 FS output bits
ST byte modules	16 modules
FS byte modules	16 modules
PSSu E S RS232	6 modules
PSSu E S RS485	6 modules

A PSSu system may have a max. 64 modules. When counting up the modules, please note the following peculiarities:

- ▶ The head module and passive junction modules (e. g. PSSu E PD, PSSu E PD1) are not counted.
- ▶ The following modules in a PSSu system are included in the count:
 - Supply voltage modules
(identifier: PSSu E F PS...)
 - Standard modules
(Identifier: PSSu E S..., PSSu K S...)
 - Failsafe modules
(identifier: PSSu E F..., PSSu K F...)

4.7.2 Maximum current load

4.7.2.1 Maximum current load capacity of the supply voltage modules

System supply	Max. current load
Module supply when supplied via - PSSu E F PS(-T)	1.5 A
Module supply when supplied via - PSSu E F PS1(-T)	2 A
Periphery supply	10 A
C-rail	10 A

▶ **Module supply**

The module supply's current load is the total current consumption resulting from the electronic modules and compact modules. If the total current consumption is higher than the supply module's current load capacity, the module supply must be refreshed with an additional supply module (see modules' technical details). The system software takes this into account.

▶ **Periphery supply**

Maximum current load: 10 A

The current load of the periphery supply is the sum of the current consumption of the sensors and actuators supplied via the input/output modules. If the current load is higher, the periphery supply must be refreshed with an additional supply module to prevent overload. Please refer to the derating diagram.

▶ **C-rail**

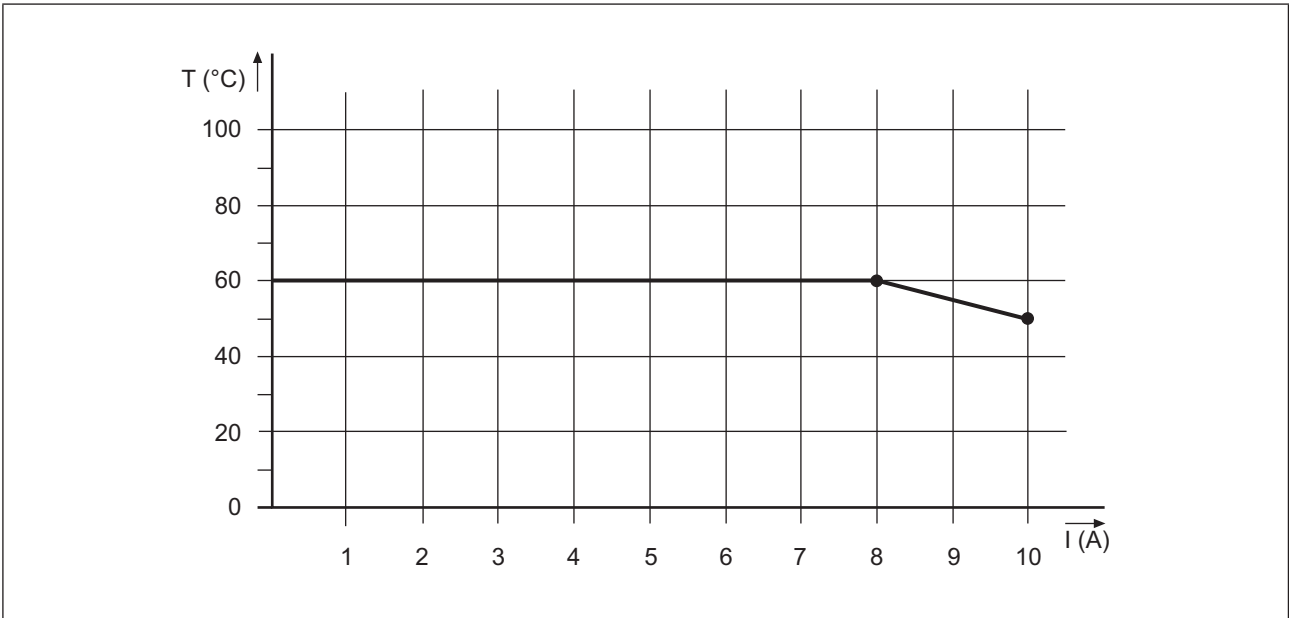
Maximum current load: 10 A

If the current load is higher, the C-rail must use a different supply to prevent overload. Please refer to the derating diagram.

4.7.2.2 Maximum current load capacity based on temperature

The maximum current load capacity of the module bus with regard to the periphery supply and C-rail can only be achieved if the max. permitted ambient temperature is adhered to.

Derating diagram for periphery supply and C-rail: Temperature T dependent on load current I



Coated version: Derating diagram for periphery supply and C-rail: Temperature T dependent on load current I



4.8 Slots of a PSSu system

The slot identifies the location of an electronic module, compact module or supply voltage module on a PSSu system.

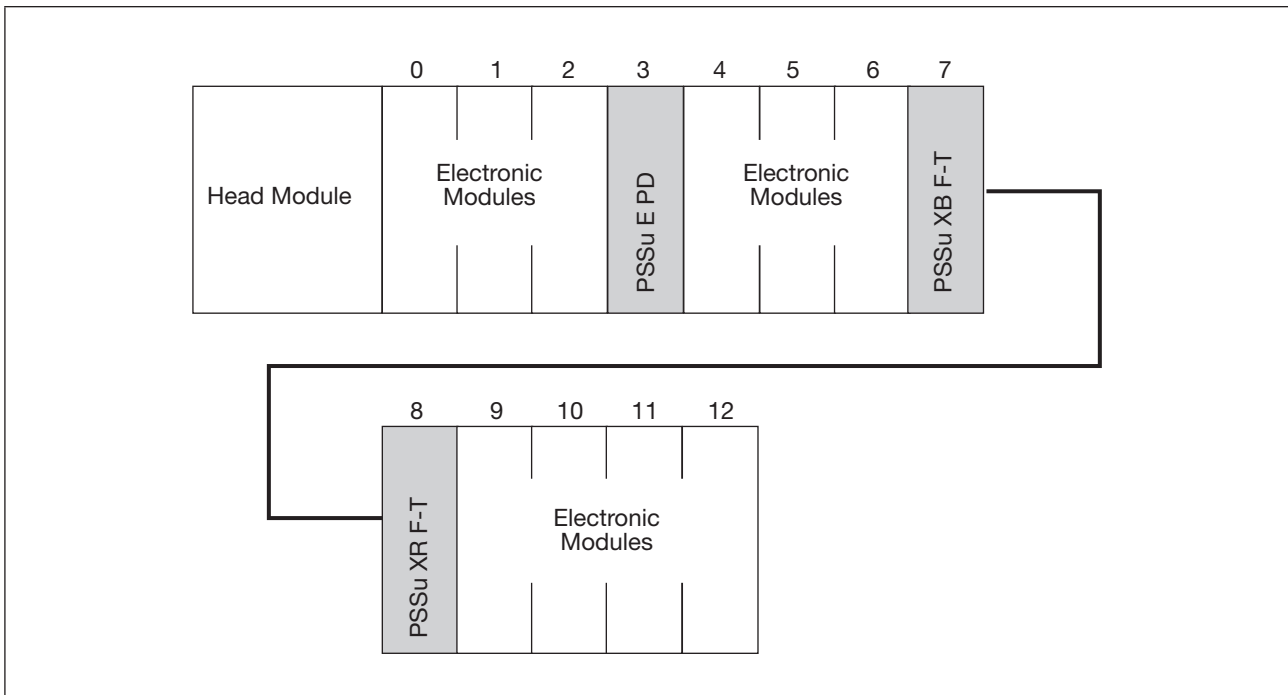
There are different ways of looking at the structure of a PSSu system. A distinction is made between the physical structure and the logical structure. As a result, there are physical slots and logical slots.

Physical slots

The following applies for physical slot numbers on slots:

- ▶ The counting sequence starts with the 1st module after the head module.
- ▶ The counting sequence starts with slot number 0.
- ▶ For slot numbering the modules are counted from left to right in ascending order.

Example

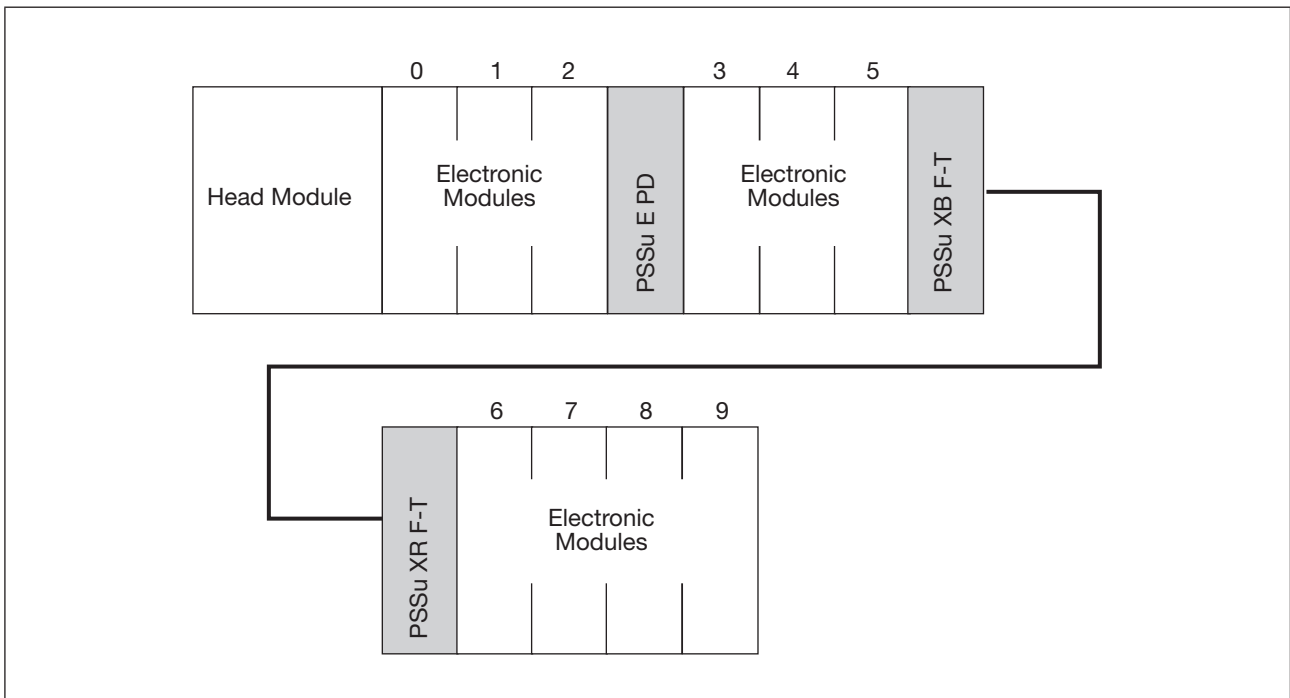


Physical slot numbers (example)

Logical slots

The same rules apply for logical slot numbers as physical slot numbers, but link modules (e.g. PSSu XB F-T) and voltage distribution modules (e.g. PSSu E PD) are not counted.

Example



Logical slot numbers (example)



Information

If the diagnostics provide information about a slot, that information refers to the logical slot number.

4.8.1 Set/actual hardware registry on a PSSu system

The device project in a PSSu system contains details of the sequence configured for the modules within their module group.

PSSu modules are divided into the following module groups:

- ▶ FS bit modules (e.g. PSSu E F 4DO 0.5)
- ▶ ST bit modules (e.g. PSSu E S 2DO 2)
- ▶ FS byte modules (e.g. PSSu E F AI U)
- ▶ ST byte modules (e.g. PSSu E S 2AI se)

If the actual hardware registry on the local PSSu system does not match the set hardware registry as regards the module sequence configured for the module group in PAS4000, a configuration error will occur.



Information

Electronic modules are supplied with a two-part coding element. When an electronic module is plugged into a base module for the first time, one part of the coding element remains on the electronic module, while its counterpart is fixed on to the base module. This is how the base module is coded. The coding element is designed to help prevent plugging errors (e.g. when exchanging an electronic module). An electronic module can only be inserted on a base module with the same mechanical coding. The coding is identified by a letter and a colour. Codings with the same letters are mechanically identical, but their colour may vary.



CAUTION!

Please note:

When modules are exchanged, an ST electronic module must not be inserted on a base module that is coded with a coding element for an FS electronic module. You can recognise the coding elements for FS modules by the fact that they are yellow. If coding elements have been removed from a base module, you will need to recommission the system in accordance with the commissioning/recommissioning specifications when electronic modules are exchanged, see PSS 4000 Safety Manual.

Example 1: Variations in the set/actual hardware registry without configuration error



The set hardware registry differs from the actual hardware registry in terms of the slot, but the sequence within the module groups matches. This variation in the set/actual hardware registry **will not lead a configuration error**.



CAUTION!

Please note that in this case, diagnostic messages will contain incorrect information about the location.

Example 2: Variations in the set/actual hardware registry with configuration error



The set hardware registry does not match the actual hardware registry either in terms of the slot or the sequence within the module groups. This variation in the set/actual hardware registry will lead to a **configuration error**.

4.9 Supply voltage and insulation

All voltages are supplied via the head module. If more voltages are needed (voltage refresh, potential isolation etc.), these will be supplied via additional supply voltage modules. The voltages are distributed to the modules via the module bus.

▶ Module supply

The module supply is the internal supply voltage for the head module, the compact modules and the electronic modules.

▶ Periphery supply

The periphery supply supplies 24 VDC to the sensors and actuators on digital input/outputs on electronic modules. On compact modules, the supply for the inputs is generated from the periphery supply; the supply for the outputs must be fed to the module terminals.

The connection on the module bus can be disconnected in order to form supply groups. On digital failsafe modules, the periphery supply is used to provide test pulses. Failsafe input/output modules will not start until the periphery supply is connected correctly. On analogue input/output modules, the periphery supply is galvanically isolated from the inputs/outputs.

▶ C-rail

The C-rail is an additional, free power rail. Access to the C-rail is available via base modules that contain the letter “-C” in their description (e.g. PSSu BP-C 1/8 S, PSSu BP-C 1/8 C).

The C-rail supply must belong to the same circuit as the other base module connections. Connecting PE to the C-rail provides a simple form of shield. On the compact modules, the C-rail is not looped through and is not available at the terminals. Permitted infeed at the C-rail:

- PE
- 0 V
- Screen
- - 30 VDC ... + 30 VDC
- - 48 VAC ... + 48 VAC

▶ Inputs/outputs on the analogue modules and counter modules

The inputs and outputs on the analogue modules and counter modules are galvanically isolated from the periphery supply. On the base modules, the supply for the outputs is generated from the periphery supply.

▶ Earth (Ground)

The module supply and periphery supply have separate connections for the respective earth potential (Ground).

▶ Functional earth

A contact spring on the base modules and compact modules connects the modules' functional earth to the mounting rail.

The supply voltages for module supply and periphery supply must be extra low voltages with safe electrical separation (PELV or SELV) in accordance with VDE 0100, Part 410. Further requirements of the power supplies can be found in the data sheets for the supply voltage modules or head modules.

When the voltages are fed separately using two power supplies, the module supply and periphery supply are galvanically isolated. If galvanic isolation is not required, a common power supply may be used for the periphery supply and module supply.

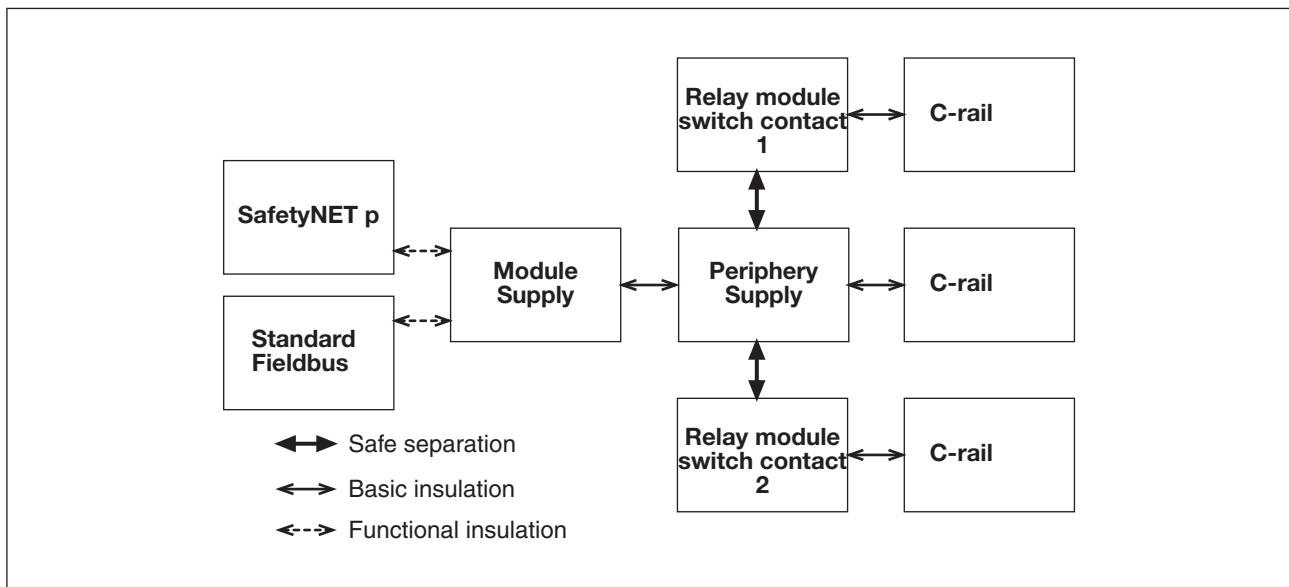
Separate power supplies for module supply and periphery supply increase availability: In the case of a short circuit in the periphery supply, the module supply is maintained and the head module can continue to exchange data with fieldbus systems, e.g. for diagnostic purposes.



Information

Please refer to the PSSuniversal Installation Manual.

Insulation of supplies in a PSSu system:



WARNING!

Risk of electrocution!

When voltage is applied, contact with live components could result in serious or even fatal injury from an electric shock.

If voltages higher than 50 VAC or 120 VDC are connected to a PSSu system, e.g. to switch contacts on relay modules, please note the following:

- Specific accident prevention regulations apply.
- The C-rail of the supply group should only be connected to the protective earth (PE).

**Information**

FS output circuits have been designed to guarantee maximum safety. Extensive tests are carried out internally for this purpose. If the periphery supply should dip during a test function, the relevant output will be deemed to be defective.

An error reaction is triggered: all safety-related system sections on the PSSu system will switch to a STOP condition and all FS outputs on the PSSu system will switch to a safe condition.

In the event of repeated voltage dips, power supplies with a buffered supply voltage should be used for greater availability.

4.10 Supply groups

When used with the relevant base module, any electronic supply voltage module is suitable to separate supplies and form supply groups.

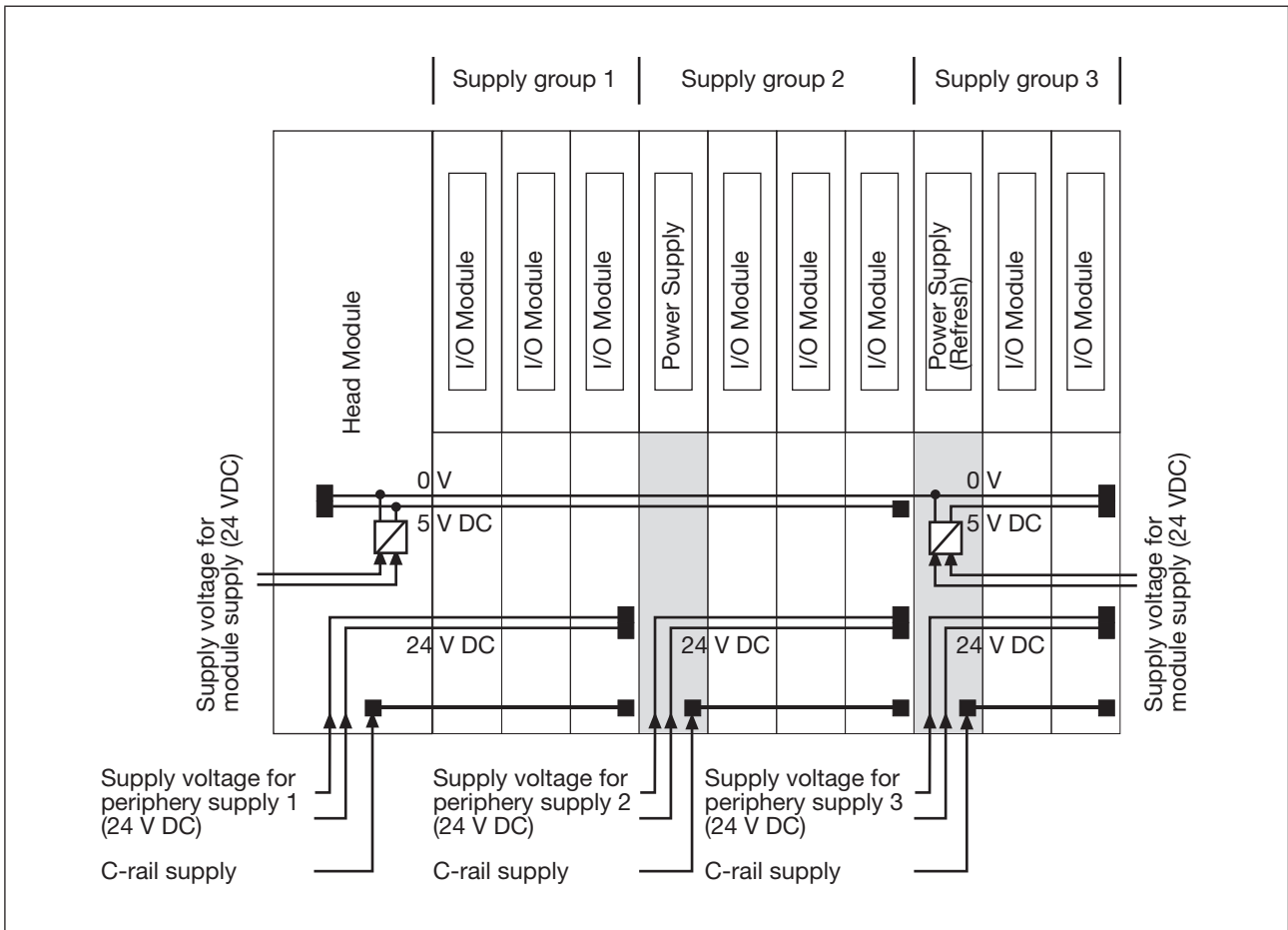
To separate supplies, the supply (periphery supply and C-rail supply) to the preceding (left-hand) modules is disconnected and a new supply provided for subsequent (right-hand) modules.

With a new supply group it is also possible to refresh the module supply by providing a new supply voltage. The 0 V supply on the module supply is never interrupted, but is always looped through across the whole module bus.

You should always refer to the description of the relevant electronic supply voltage module.

The following modules are suitable for separating supplies, for example:

Electronic module	Base module	Module supply	Periphery supply and C-rail
PSSu E F PS	PSSu BS-R 1/8 S PSSu BS-R 1/8 C	Refreshing (new supply voltage is provided, 0 V supply is not interrupted).	Supply of the left-hand modules is interrupted and a new supply is provided for the right-hand modules.
PSSu E F PS1	PSSu BS-R 2/8 S PSSu BS-R 2/8 C		
PSSu E F PS-P	PSSu BS 1/8 S PSSu BS 1/8 C	No refreshing (5 V is not separated).	
PSSu E F BSW	PSSu BS 2/8 S PSSu BS 1/8 C		



The supply voltage for the first supply group can also be integrated within the head module.

4.11 Status LEDs on head modules

The head modules contain status LEDs, which provide information on the status of various system sections. However, not every head module provides all of the status LEDs described below. Consequently, please refer to the operating manual for the relevant head module.



Information

The diagnostic table and diagnostic log are available for diagnostic messages and remedies.

Legend:

	LED on
	LED flashes
	LED off







MBUS

The "MBUS" LED indicates the status of the FS and ST module bus.

Colour	State	Meaning
---		No modules present
Green		FS and ST module bus are operating error-free
Red		Operating status "Safe condition of all FS outputs on the PSSu system" or Unable to locate at least one module (e.g. a module has been removed during operation, actual/registered hardware does not match)
		Operating status "FS module bus in a STOP condition with error: Major FS error"






SD CARD

The "SD CARD" LED shows the status of the removable data medium.

Colour	State	Meaning
---		Supply voltage for module supply is missing
Red		SD card is missing or SD card not recognised or SD card defective
Green		Naming data and device project on the PSSu system and SD card match
		Product type on the SD card does not match the head module or No device project on the SD card
Green-red		Naming data and device project on the PSSu system and SD card do not match
Orange		Device identification activated by user






FS RUN

The "FS RUN" LED shows the status of the FS resource.

Colour	State	Meaning
---		FS resource has not been started or is in a STOP condition
Green		Operating state "FS resource in RUN condition without error": The FS resource tasks are running without fault. The project is licensed.
		Operating state: "FS resource in RUN condition with error": - Task in TERMINATED state or - Task in STOP state At least one FS resource task is not running. The project is licensed.
Orange		Operating state "FS resource in RUN condition without error": The FS resource tasks are running without fault. The project is unlicensed.
		Operating state: "FS resource in RUN condition with error": - Task in TERMINATED state or - Task in STOP state At least one FS resource task is not running. The project is unlicensed.








ST RUN

The "ST RUN" LED indicates the status of the ST resource.

Colour	State	Meaning
- - -		ST resource has not been started or is in a STOP condition
Green		Operating state "ST resource in RUN condition without error": The ST resource tasks are running without fault. The project is licensed.
		Operating state: "ST resource in RUN condition with error": - Task in TERMINATED state or - Task in STOP state At least one ST resource task is not running. The project is licensed.
Orange		Operating state "ST resource in RUN condition without error": The ST resource tasks are running without fault. The project is unlicensed.
		Operating state: "ST resource in RUN condition with error": - Task in TERMINATED state or - Task in STOP state At least one ST resource task is not running. The project is unlicensed.



DIAG

The "DIAG" LED indicates whether there is an error on a system section of the PSSu system. Precise evaluation can occur via the diagnostic table.

Colour	State	Meaning
---		No system section is started, module supply is missing.
Green		No message of "Error" or "Warning" severity is present for the device.
		Device diagnostic list and device diagnostic log are being prepared
Red		A message of "Error" severity is present for at least one system section (see diagnostic table).
		A major FS error is present for at least one FS system section (see diagnostic list).
Orange		A message of at least "Warning" severity is present for the PSSu system (see diagnostic list).
Red - green		Start of "deliberate operator action" (function of reset button)



FS FORCE

The "FS FORCE" LED shows the status of forcing on the FS resource.

Colour	State	Meaning
---		Forcing on the FS resource is inactive
yellow		Forcing on the FS resource is active






ST FORCE

The "ST FORCE" LED indicates the status of forcing on the ST resource.

Colour	State	Meaning
---		Forcing on the ST resource is inactive
yellow		Forcing on the ST resource is active






FS Snp

The "FS Snp" LED indicates the status of the safety-related system section FS-SafetyNET p RTFN.

Colour	Status	Key
- - -		System section FS SafetyNET p RTFN has not been started
Green		Operating status "FS SafetyNET p RTFN in RUN condition without error"
		Operating status "FS SafetyNET p RTFN in RUN condition with minor error"
Red		Operating status "FS SafetyNET p RTFN in STOP condition with error: Major FS error"
		Operating status "FS SafetyNET p RTFN in STOP condition with error: Major FS+ST error"


ST Snp

The "ST Snp" LED indicates the status of the non-safety-related system section ST-SafetyNET p RTFN.


Colour	Status	Key
- - -		System section ST SafetyNET p RTFN has not been started
Green		Operating status "ST SafetyNET p RTFN in RUN condition without error"
		Operating status "ST SafetyNET p RTFN in RUN condition with minor error"
Red		Operating status "ST SafetyNET p RTFN in STOP condition with error: Major FS+ST error"
		

5V, 24V

The "5 V" LED shows the status of the module supply.

Colour	Status	Meaning
---	●	No supply voltage for module supply or supply voltage is faulty
Green		Module supply is available on the module bus


The "24 V" LED shows the status of the periphery supply.

Colour	Status	Meaning
---	●	No supply voltage for periphery supply or supply voltage is faulty
Green		Periphery supply is available on the module bus

X3: LNK, X3: TRF, X4: LNK, X3: TRF


A PSSu system can have either one Ethernet interface (X3) or two Ethernet interfaces (X3 and X4) (see Ethernet interface, [Page 74](#)). An Ethernet interface is assigned two status LEDs on the head module as display elements. The status LEDs indicate various connection and communication states.

X3: LNK, X4: LNK

Colour	State	Meaning
---	●	No network connection
Green		Network connection is error-free

The designation "LNK" stands for "LINK".

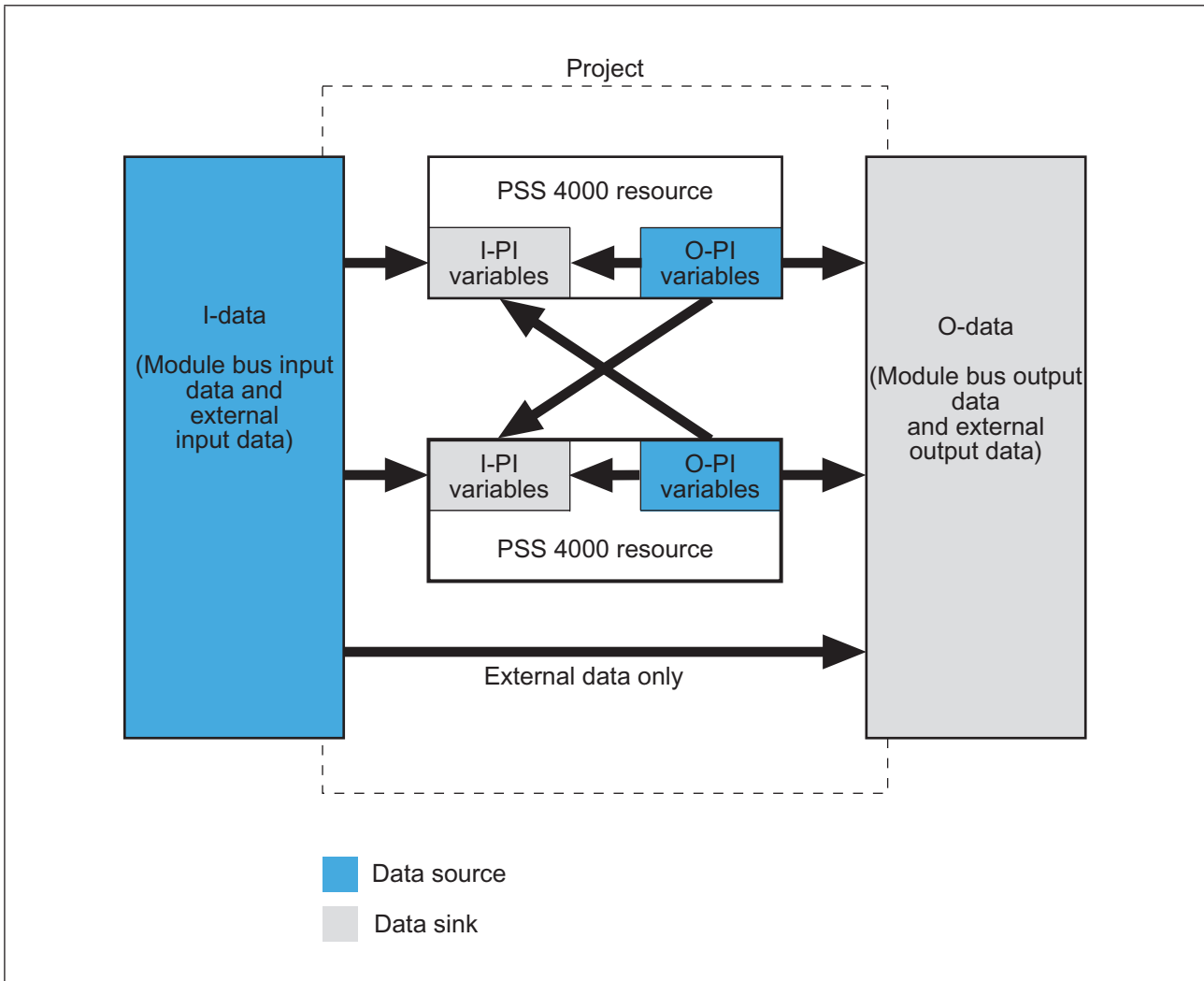
X3: TRF, X4: TRF

Colour	State	Meaning
---	●	No data traffic
Yellow		Data traffic is error-free

The designation "TRF" stands for "TRAFFIC".

4.12 I/O data

I/O data



I-data is:

- ▶ Input data that comes from the module bus, i.e. from the modules on the PSSu systems
- ▶ Input data that come in from externally, e.g. via Modbus/TCP, via fieldbuses; it does not include data of the system-internal communication via SafetyNET p

O-data is:

- ▶ Output data that goes to the module bus, i.e. that are transferred to the modules on the PSSu systems.
- ▶ Output data that go out externally, e.g. via Modbus/TCP, via fieldbuses; it does not include data of the system-internal communication via SafetyNET p

4.12.1 I/O data of PSSu modules on the module bus

Each I/O module reads in or outputs data.

A module's I/O data is organised into I/O data types, which combine one or more I/O data elements within a structure.

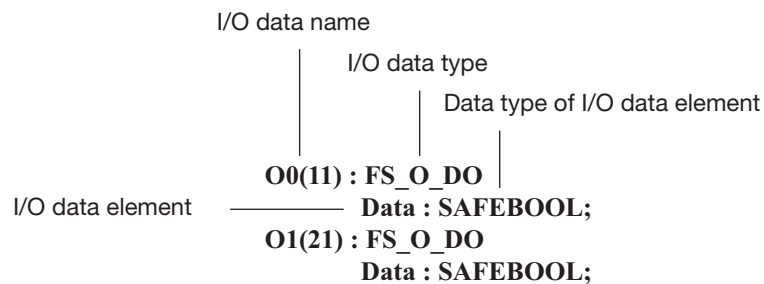
A module's I/O data and I/O data types are stated in the module's operating manual.

On hardware inputs and outputs, the I/O data name is composed of the input/output designation and the terminal number. With I/O data names from other I/O data, a descriptive name is used.

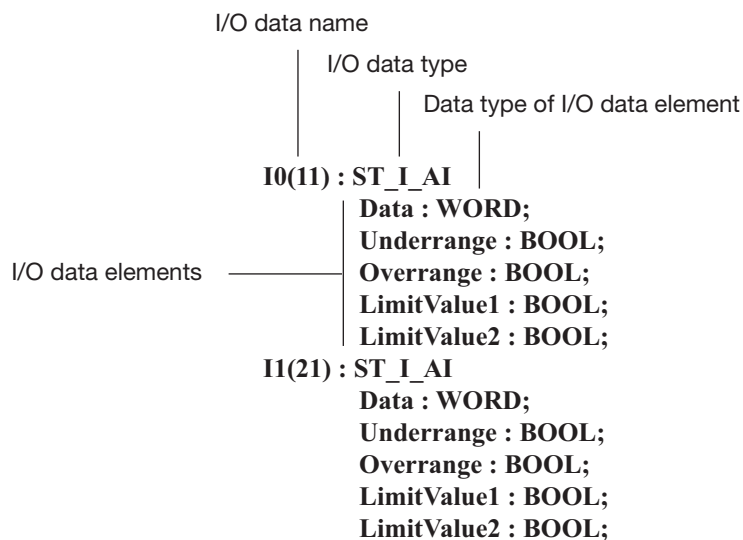
The name of the I/O data type consists of the safety state of the PSSu module (FS or ST) and the data flow direction (I- or O-data). It's important to note that FS-I/O data types can also contain ST data. ST-I/O data types only contain ST data.

The I/O data of modules in the user program is accessed via PI variables.

Example for the I/O data of the module PSSu E F 2DO



Example for the I/O data of the module PSSu E S AI U



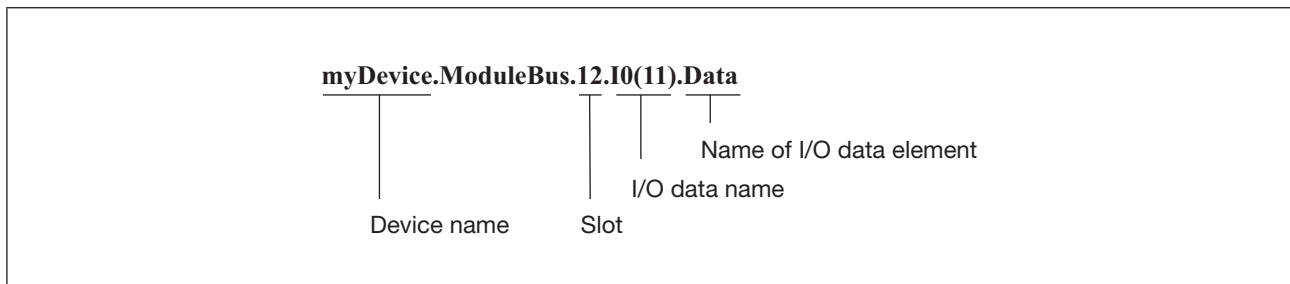
4.12.1.1 Address display

The I/O data addresses are displayed at several points, e.g. in PAS4000 and in the diagnostics. The addresses are structured as follows:

<Device Name>.ModuleBus.<Slot>.<I/O Data Name>.<Name of I/O Data Element>

Example

The I/O data element "Data" from input I0 of module PSSu E S AI U on slot 12 of the PSSu system with the device name "myDevice" will have the following address:



4.12.2 I/O data with external communication

External communication refers to data exchange in which output data (external O-data) leaves the automation system PSS 4000 and input data (external I-data) enters the system. This can be performed using various communication modes (e.g. via Modbus/TCP, UDP, fieldbuses).

Depending on the communication mode a description of the I/O data and corresponding address display can be found under “Ethernet-based communication”.

5 Ethernet

5.1 Auto IP mechanism

The Auto-IP mechanism is used to assign a free IP address automatically and without manual configuration to a device in an Ethernet network. For this purpose, the IANA provides the address range 169.254.1.0/16.

PSS 4000-exclusive devices

The Auto-IP mechanism is activated for PSS 4000-exclusive devices in delivery status or after an original reset. Activation is performed when [Booting system sections](#). During booting, the Auto-IP mechanism assigns these devices a free IP address from the range 169.254.1.1 ... 169.254.1.254.

If there is an individual PSS 4000-exclusive device with activated Auto-IP mechanism in the network, the device receives the default IP address 169.254.1.1, if this IP address is still free. When the default IP address is no longer free or when several devices with activated Auto-IP mechanism are in the network simultaneously, a device receives a free IP address from the range 169.254.1.1 ... 169.254.1.254.

The Auto-IP mechanism is deactivated during the device naming procedure (see Device naming procedure for PSS 4000-exclusive devices, [Page 225](#)).

PSS 4000-operable devices

PSS 4000-operable devices from Pilz and PSS operable devices from third party manufacturers select a free IP address from the range 169.254.1.0/16.

5.2 Ethernet interface

PSS 4000 devices have an Ethernet interface for connecting to the Ethernet. The specification of the Ethernet interface (e. g. internal switch with 2 free switch ports), the connection technology (e. g. RJ45) and the naming of the Ethernet interface (e. g. ETHERNET, SafetyNET p) is dependent on the device. Please refer to the information stated in the operating manual of the device you are using.



CAUTION!

Only use Ethernet subscribers (e.g. switches, hubs) that support the auto-sensing/autonegotiation function. Otherwise a duplex conflict (duplex mismatch) may arise, which can cause considerable delays or sporadic data transmission errors.

In the PSS 4000 automation system Ethernet interfaces are used as follows:

- ▶ Connection for data exchange via SafetyNET p (see SafetyNET p, [Page 80](#))
- ▶ Connection for data exchange via external communication with IP connections as Modbus/TCP, Raw UDP etc. (see External Communication, [Page 102](#))
- ▶ Connection of PAS4000 PC (programming interface)
- ▶ Connection of the OPC Server (see OPC Server, [Page 394](#))

5.2.1 RJ45 interface

PSS 4000 devices can have an Ethernet interface with RJ45 connection technology (RJ45 interface). The number of RJ45 interfaces depends on the device. Details of the PSS 4000-exclusive devices and PSS 4000-operable devices from Pilz are available in the operating manual for the relevant unit. PSS 4000-exclusive devices with two RJ45 interfaces have an internal switch with two free switch ports. As a result, these devices do not need to use any additional switches or hubs.

The RJ45 interfaces on PSS 4000-exclusive devices or PSS 4000-operable devices from Pilz have an autosensing and autocrossing function.

The autosensing function automatically detects whether data transfer is occurring at 10 MBit/s or 100 MBit/s. The autocrossing function means there is no need to distinguish between patch cable (uncrossed data line connection) and crossover cable (crossover data line connection) on the connection cable. The correct data line connection is automatically established internally. Patch cable can therefore be used as the connection cable for end devices as well as cascading.

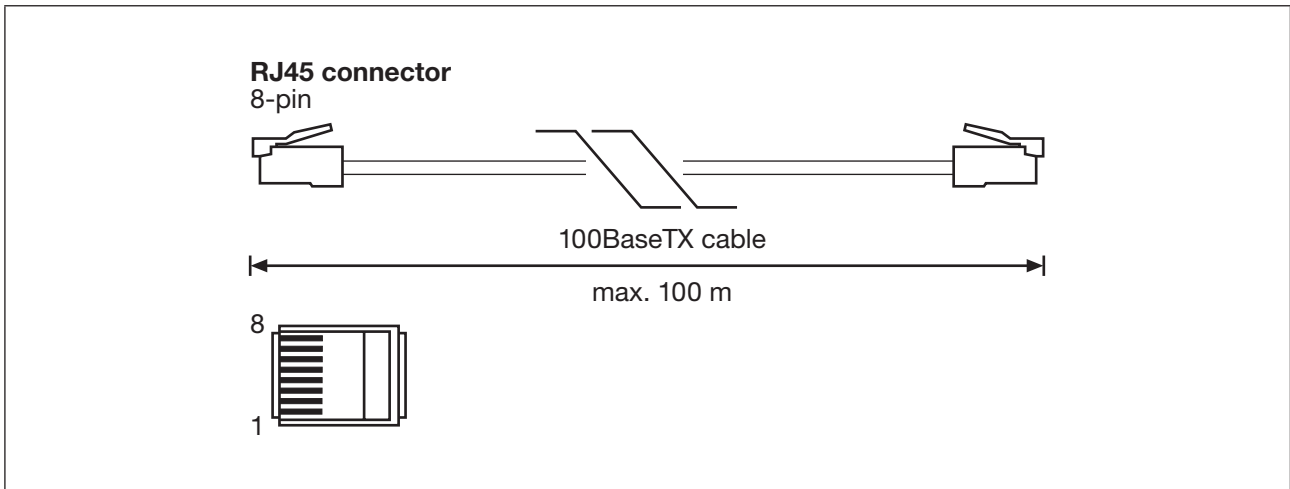
5.2.2 RJ45 interface assignment

PSS 4000-exclusive devices and PSS 4000-operable devices from Pilz with RJ45 interface have the following interface assignment:

RJ45 socket (8-pin)	PIN	Standard	Crossover
	1	TD+ (Transmit+)	RD+ (Receive+)
	2	TD- (Transmit-)	RD- (Receive-)
	3	RD+ (Receive+)	TD+ (Transmit+)
	4	n.c.	n.c.
	5	n.c.	n.c.
	6	RD- (Receive-)	TD- (Transmit-)
	7	n.c.	n.c.
	8	n.c.	n.c.
n.c. = not connected			

Socket for RJ45 connector (plan view)

5.2.3 Connector and connection cable for RJ45



Assignment of the RJ45 connector (viewed from below)

Please note the following minimum requirements of the connector and cable:

100BaseT cable

Bit rate	100 MBit/s
Max. segment length	100 m
Medium	STP, S/STP [1] (shielded/double-shielded twisted pair cable) 2 pairs Category 5 [2]
Connection	Shielded 8-pin RJ45 connector
Topology	Point-to-point/hub (hub, switch)

[1]

For Ethernet applications in an industrial environment, Pilz recommends double-shielded twisted pair cable (S/STP).

[2]

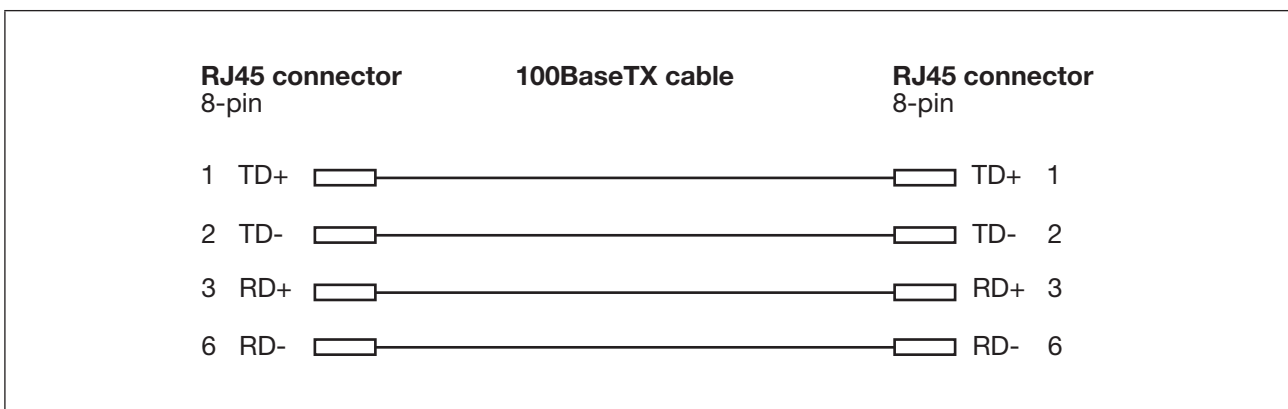
The classification of TP cables is as follows:

Category 1	No performance criteria
Category 2	Up to 1 MHz
Category 3	Up to 16 MHz (10BaseT)
Category 4	Up to 20 MHz (Token-Ring, 10BaseT, 100BaseT4)
Category 5	Up to 100 MHz (10Base-T, 100BaseT4, 100BaseTX)
Category 5e	Up to 100 MHz (1000Base-T)
Category 6	Up to 250 MHz

Category 6a	up to 500 MHz (10GBase-T)
Category 7	Up to 600 MHz
Category 7a	Up to 1000 MHz

TP cables for Ethernet applications generally have an impedance level of 100 Ohm.

The connection can be made via a 100BaseTX cable with uncrossed data line connection (patch cable).



100BaseTX cable (standard)



Information

If you assemble the connection cable yourself we recommended that you use connector and cable made by the same manufacturer. This is because considerable variances may arise regarding connector dimensions and cable cross sections, due to the permitted tolerances.

5.2.4 Status LEDs on the RJ45 interface

An RJ45 version of the Ethernet interface generally has status LEDs to display various connection and communication states. Details can be found in the operating manual of the relevant device.


6 SafetyNET p

6.1 Overview

SafetyNET p is an Ethernet-based multi-master bus system suitable for industrial use. It is used as standard on PSS 4000 as a fieldbus system, to resolve decentralised tasks.

The SafetyNET p protocol contains a safe data channel, which is certified for data transfer in accordance with SIL 3 of IEC 61508. On SafetyNET p, both safety-related and non-safety-related data is transferred via the same bus cable.

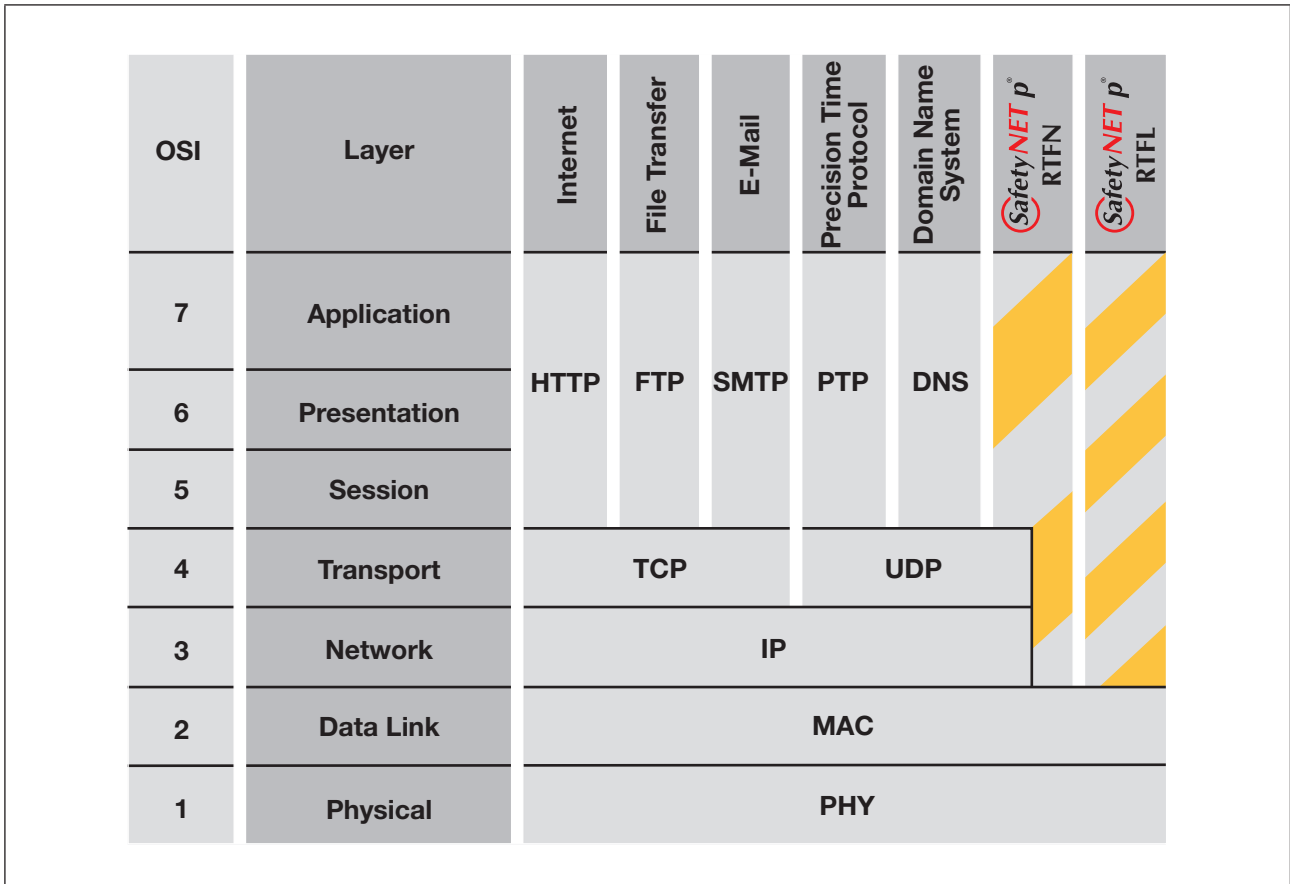
Two versions of SafetyNET p are available:

- ▶ RTFN (Real Time Frame Network)
- ▶  RTFL (Real Time Frame Line)

The application layer on SafetyNET p (Layer 7 in ISO/OSI's 7-layer model) is based on CANopen. SafetyNET p combines the CAN Bus protocol with a standardised industrial application layer. This includes standardisation of the communication as well as the technical and functional performance data, which enables distributed automation units to be networked. Defined application objects are used in device profiles.

6.1.1 SafetyNET p in ISO/OSI's 7-layer reference model

The ISO/OSI (Open System Interconnection) reference model is a standardised format for describing all data communication functions for all data processing communication systems. Functionalities are divided to form a layer model, with 7 different functionalities. Field-bus communication is described by three layers (Layers 1, 2 and 7).



SafetyNET p in ISO/OSI's 7-layer reference model

6.1.2 RTFN (Real Time Frame Network)

The RTFN transport layer can be used at process control and cell level, where the demand is for standard Ethernet protocols and the requirements for real-time are lower. RTFN is primarily designed to network the RTFL real-time cells and connect standard Ethernet subscribers, such as visualisation devices or service PCs for example.

RTFN is fully compatible with the standard Ethernet familiar from office automation, as it is based on the same transport mechanisms. As RTFN is designed to be mixed with other utilities in an Ethernet network, with no specific restrictions, the RTFN subscribers communicate with other devices in the network (laptops, printers etc.) on an equal basis.



Information

If non-RTFN-enabled devices place a high load on the network, availability of the SafetyNET p RTFN can be adversely affected. Where the availability requirements are high we recommend that you do not run any other communication in parallel with SafetyNET p RTFN, or that you use a switch with a management function.

6.1.3 RTFL (Real Time Frame Line)



If higher speeds are required, the RTFL transport layer can be used. The RTFL transport layer enables scan times to be optimised and can therefore be used in extremely time-critical applications, such as drive controllers.

6.1.4 Transfer of safety-related and non-safety-related data

SafetyNET p can be used to transfer safety-related data through an integrated safe communication layer. The security mechanisms are designed to meet up to SIL3 of IEC 61508. The safety-related data is sent encapsulated within SafetyNET p telegrams. As a result, all other network components such as switches or cable can be designed as standard Ethernet components. They do not affect the transfer of safety-related data.

Non-safety-related SafetyNET p subscribers such as PCs or standard control systems, for example, do not influence safety-related communication either. As a result it is possible to mix the operation of safety and non-safety-related subscribers within a network.

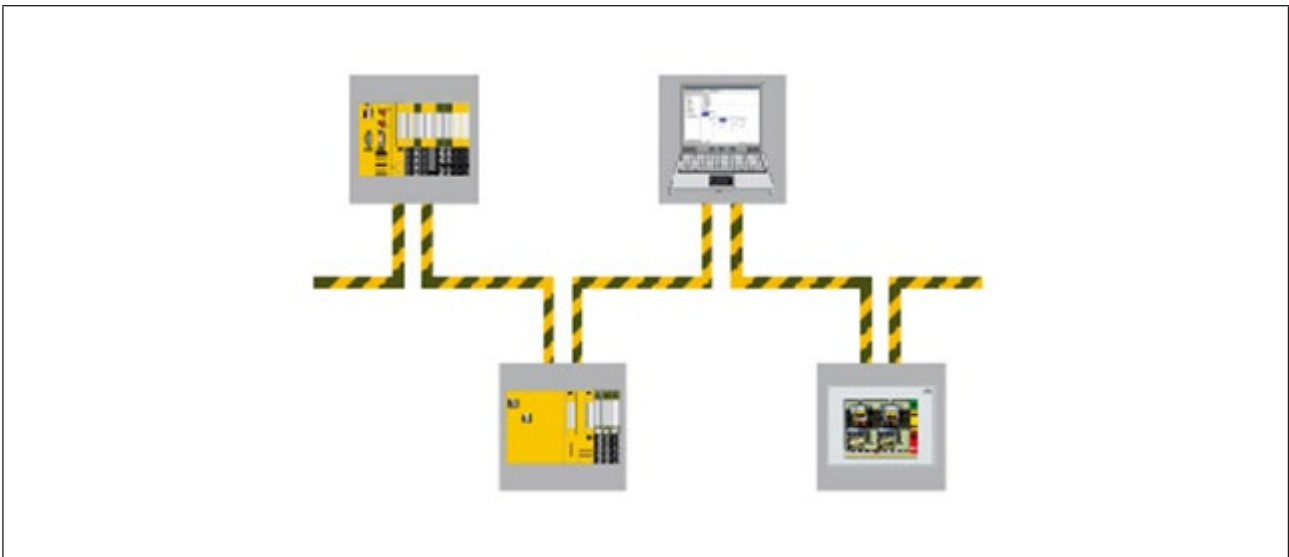
6.2 Topology on SafetyNET p

As SafetyNET p is an Ethernet-based bus system, all valid structures familiar to Ethernet, as line topology or star topology, are possible. You can also implement dynamic structures.

6.2.1 Linear topology

A linear topology is created by connecting several bus subscribers in a line, without branches. This topology is the familiar fieldbus structure. A linear topology is preferable on plants that extend over a wide area (e.g. conveyor systems, interconnecting manufacturing cells).

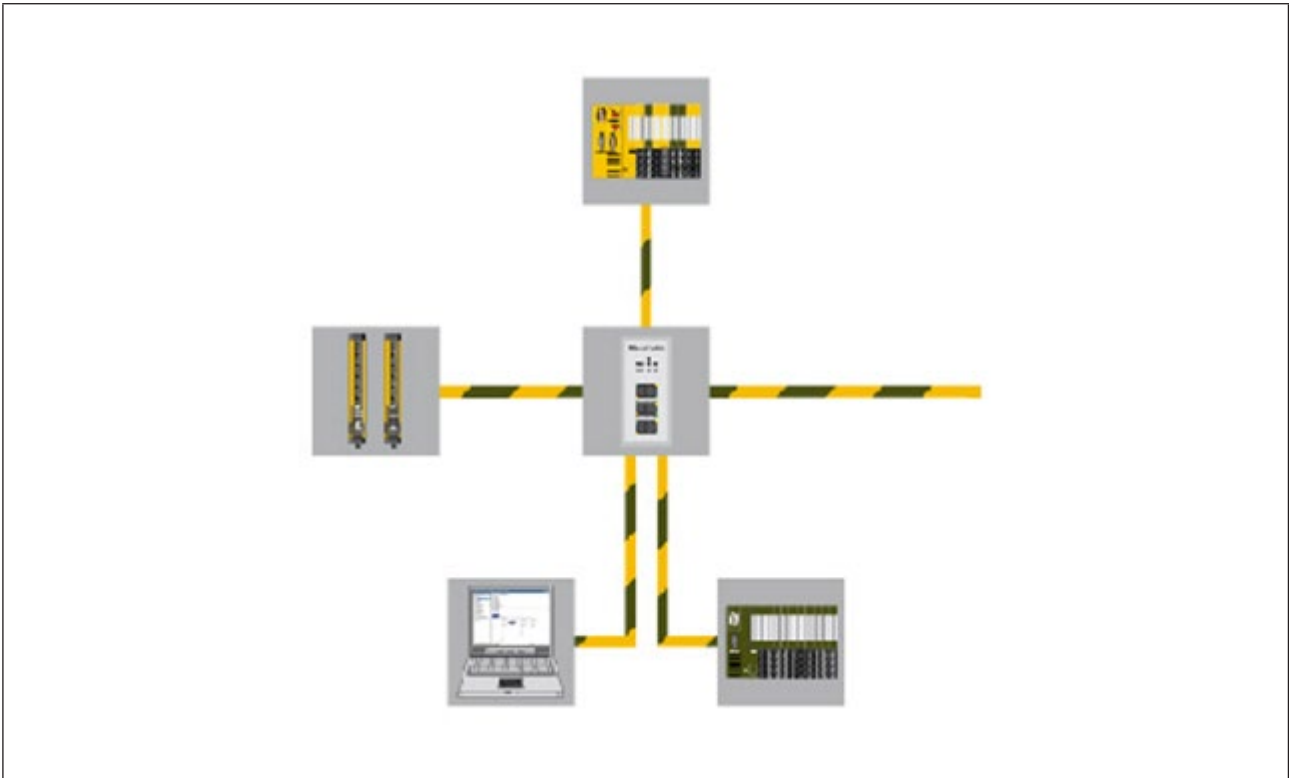
SafetyNET p subscribers often have two Ethernet interfaces (autosensing switch with two free ports) to enable a linear topology. That way there is no need for any additional network components, such as switches.



Linear topology (principle)

6.2.2 Star topology

A star topology is formed when several subscribers are connected at a central point. A switch is generally used as the central signal distributor. The communication partners communicate via the switch in point-to-point connections. A star topology is preferable in areas with a high density of devices over a small physical area (e.g. small manufacturing cells, standalone production machines).



Star topology (principle)

6.2.3 **Dynamic structures**

SafetyNET p supports the familiar Ethernet capacity to dynamically hot swap subscribers. This is particularly useful for service, maintenance and diagnostics.

6.3 Selecting SafetyNET p components

SafetyNET p supports the familiar, robust fieldbus connection technology with established plug-in connectors, as well as economical IP20 cabling in the form of RJ45 connectors in the control cabinet.

Conventional, standard Ethernet infrastructures can be used if the performance is satisfactory. This includes connectors, cables, routers, switches, gateways and communication channels.

6.3.1 Selecting components for RTFL



For RTFL generally those components are mandatory that have a conformity test in accordance with Safety Network International e.V. and that correspond to the respective SafetyNET p specification in its current form. This is the case for components from the Pilz range, for example.

6.3.2 Selecting components for RTFN

For RTFN generally those components must be used that have a conformity test in accordance with Safety Network International e.V. and that correspond to the respective SafetyNET p specification in its current form. This is the case for components from the Pilz range, for example.

In principle it is possible to use non-certified components, provided they are designated for use in an industrial environment and the requirements of Safety Network International e.V. have been met.



Information

If you assemble the connection cable yourself we recommended that you use connector and cable made by the same manufacturer. This is because considerable variances may arise regarding connector dimensions and cable cross sections, due to the permitted tolerances.

6.4 SafetyNET p interface

The physical connection for data exchange via SafetyNET p is provided via an Ethernet interface (see Ethernet interface, [Page 74](#)).



CAUTION!

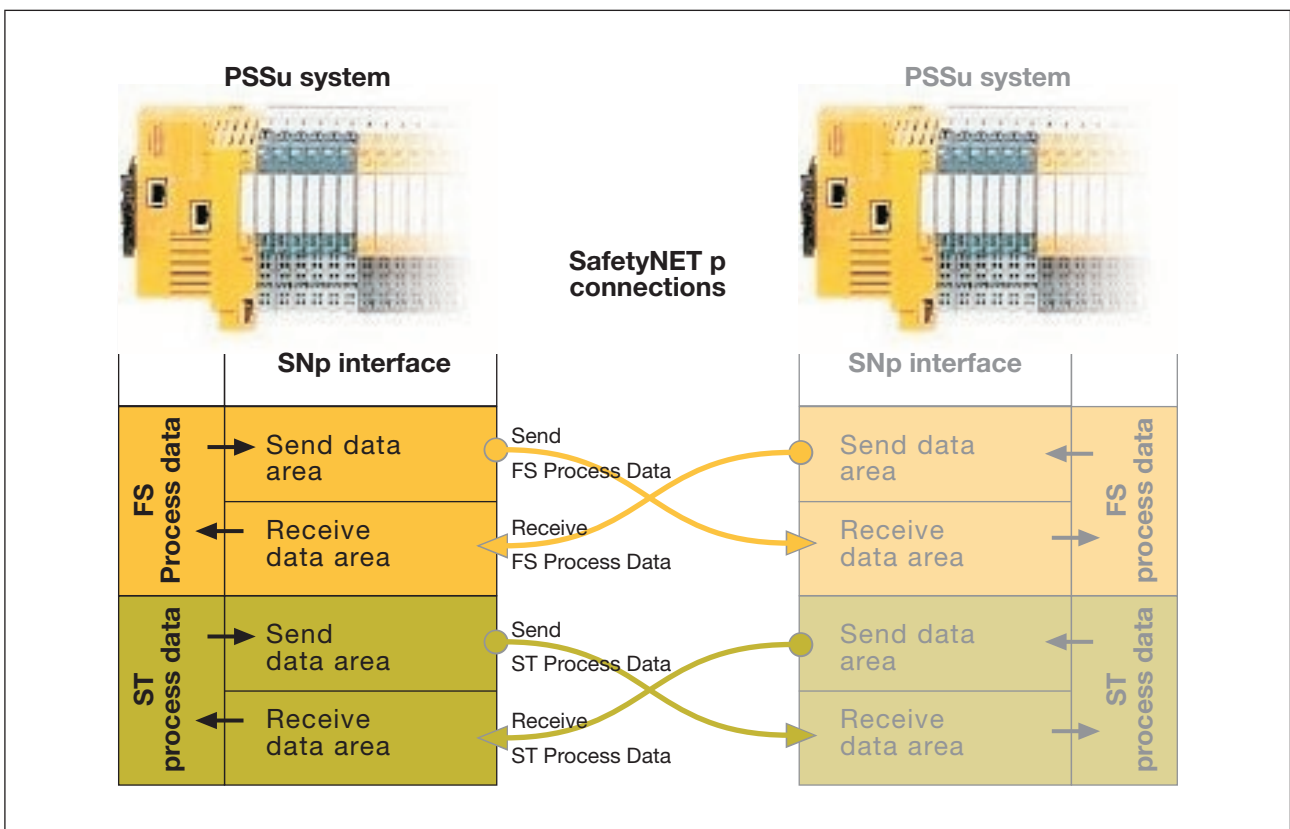
Only use SafetyNET p subscribers (e.g. switches, hubs) that support the autosen-
sensing/autonegotiation function, otherwise a duplex conflict (duplex mis-
match) may arise, which can cause considerable delays or sporadic data
transmission errors.

6.5 Process data exchange via SafetyNET p RTFN

For process data exchange via SafetyNET p RTFN, PSS 4000 devices must have the system sections "FS-SafetyNET p RTFN" and/or "ST-SafetyNET p RTFN".

Process data exchange on SafetyNET p RTFN occurs via so-called connections. The FS/ST SafetyNET p RTFN system section of a PSS 4000 device sends process data from its send data area via a transmit connection (Tx connection) and receives process data in its receive data area.

Safety-related process data (FS process data) is sent via secure Tx connections (FS-Tx connections) on SafetyNET p's secure data channel. Non-safety-related process data (ST process data) is sent via non-secure Tx connections (ST-Tx connections) on SafetyNET p's non-secure data channel.



Process data exchange principle (e.g. PSSu systems)

Each Tx-connection consists of a transmit part and a receive part. The transmit part corresponds to an area in the send data area (Tx data area) of the transmitting PSS 4000 device. The receive part corresponds to an area in the receive data area (Rx data area) of the receiving PSS 4000 device.



Connection principle (e.g. PSSu systems)

6.5.1 Cyclical processing of connections

The FS/ST SafetyNET p RTFN system section on the transmitting PSS 4000 device starts a Tx connection cyclically with the cycle time $t_{\text{SNp RTFN}}$ and sends the process data via SafetyNET p (see Reaction times, [Page 369](#)).

In the FS/ST SafetyNET p RTFN system section of the receiving PSS 4000 device, event-driven tests check whether new process data has arrived via the Rx connection that belongs to the Tx connection.

After the TX connection is started, the process data must be available in the corresponding receive data area within the transmission delay time $t_{\text{DelaySNp RTFN}}$ (see Reaction times, [Page 369](#)).

6.5.2 Data transfer limits

The data transfer limits are determined by the following data:

▶ **Maximum number of ST connections per ST SafetyNET p RTFN system section**

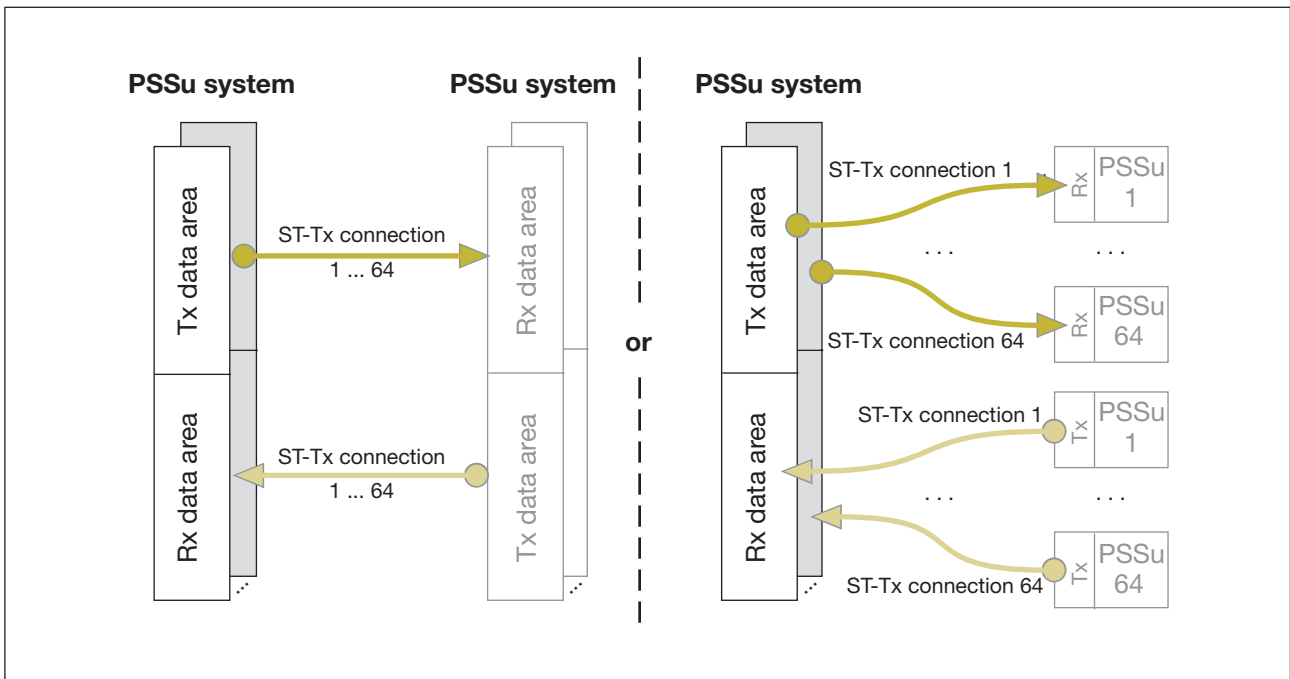
The maximum number of ST-Tx connections is limited by the number of ST send data areas that an ST SafetyNET p RTFN system section can manage. The number of ST send data areas always corresponds to the number of ST receive data areas.

The number of send and receive data areas that an ST SafetyNET p RTFN system section can manage depends on the PSS 4000 device that is used (e.g. head module for control systems PSSu PLC, control systems PSSu multi, decentralised system PSSu I/O). Details can be found in the operating manual of the relevant device.

The ST-Tx connections can either be established to a single PSS 4000 device or to the SafetyNET p RTFN system sections of various PSS 4000 devices.

Example: Control system PSSu PLC:

The ST SafetyNET p RTFN system section of a control system PSSu PLC can manage the send data areas for a max. 64 ST-Tx connections; consequently it can also manage the receive data areas for a max. 64 ST-Tx connections.



Max. number of ST-Tx connections on control systems PSSu PLC

▶ **Maximum number of FS connections per FS SafetyNET p RTFN system section**

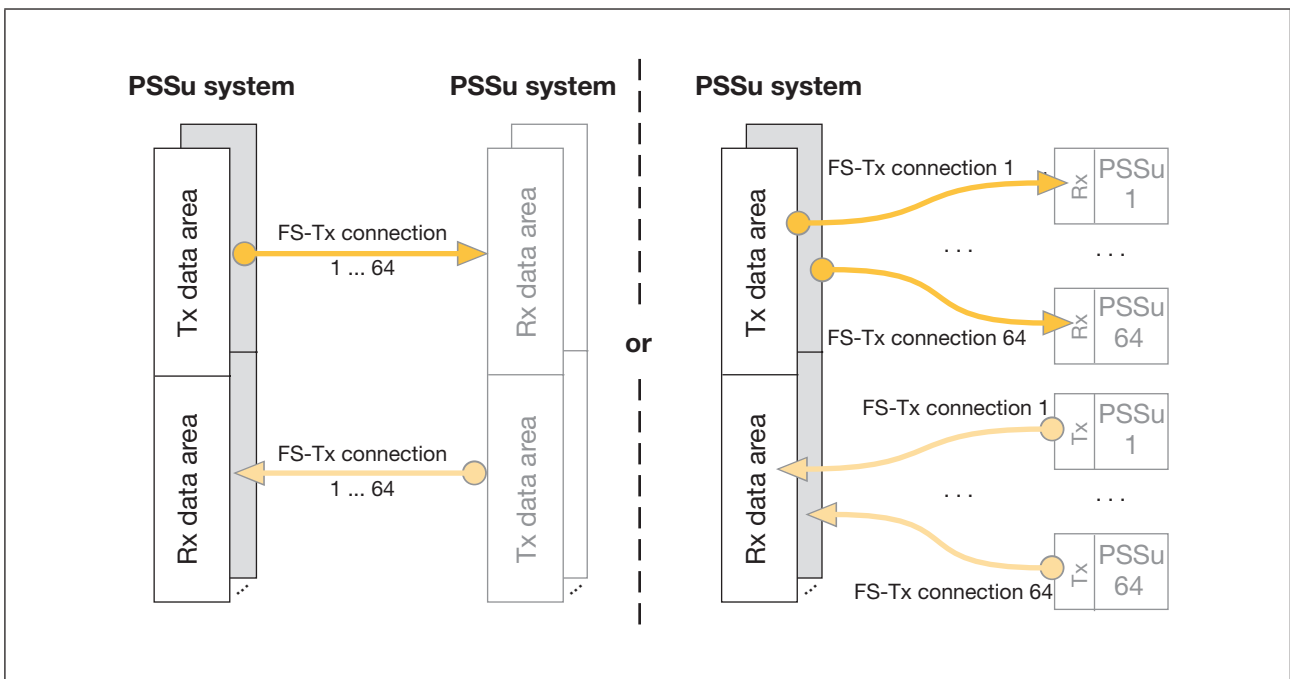
The maximum number of FS-Tx connections is limited by the number of FS send data areas that an FS SafetyNET p RTFN system section can manage. The number of FS send data areas always corresponds to the number of FS receive data areas.

The number of send and receive data areas that an FS SafetyNET p RTFN system section can manage depends on the PSS 4000 device that is used (e.g. head module for control systems PSSu PLC, control systems PSSu multi, decentralised system PSSu I/O). Details can be found in the operating manual of the relevant device.

The FS-Tx connections can either be established to a single PSS 4000 device or to the FS SafetyNET p RTFN system sections of various PSS 4000 devices.

Example: Control system PSSu PLC:

The FS SafetyNET p RTFN system section of a control system PSSu PLC can manage the send data areas for a max. 64 FS-Tx connections; consequently it can also manage the receive data areas for a max. 64 FS-Tx connections.



Max. number of FS-Tx connections on control systems PSSu PLC

▶ **Max. number of variables per PSS 4000 device**

The number of variables that can be managed depends on which PSS 4000 device is used (e.g. head module for control systems PSSu PLC, control systems PSSu multi or decentralised system PSSu I/O). Details can be found in the operating manual of the relevant PSS 4000 device.

Examples:

- **Control system PSSu PLC:**
 - 5000 variables with elementary ST data types
 - 4000 variables with elementary FS data types
- **Control system PSSu multi:**
 - 2500 variables with elementary ST data types
 - 2000 variables with elementary FS data types

▶ **Max. number of bytes per connection**

The number of bytes per connection depends on the PSS 4000 device you are using.

- ST process data sends from the Tx data area:
246 Bytes per ST-Tx connection
- ST process data receives in the Rx data area:
246 Bytes per ST-Tx connection
- FS process data sends from the Tx data area:
115 Bytes per FS-Tx connection
- FS process data receives in the Rx data area:
115 Bytes per FS-Tx connection

▶ **Max. number of variables per connection**

The number of variables that can be transmitted in a connection depends on their bit requirement. A variable's bit requirement depends on its data type and any possible expansion (valid bit).

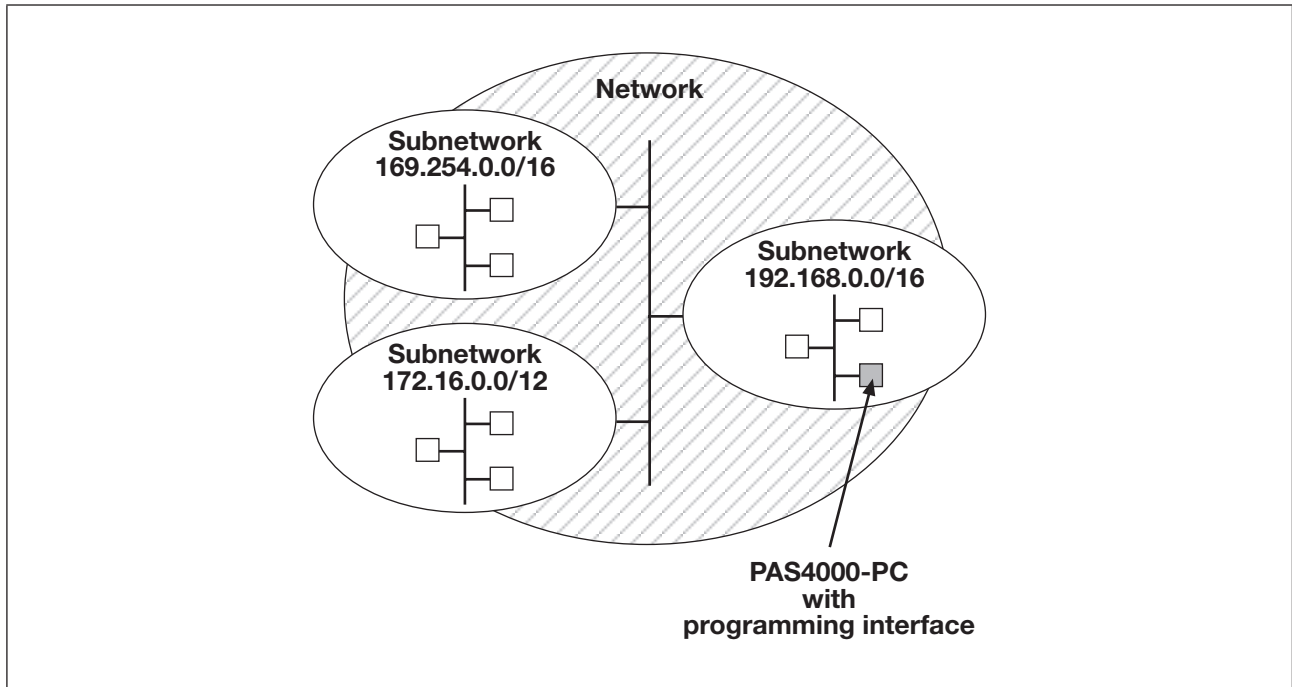
If the variables to be transmitted exceed the max. number of bytes that can be transmitted with a connection, the variables are automatically split over additional connections. Please note that this may affect data consistency (see Data consistency, [Page 214](#)).

- **Variables with elementary data type**
The bit requirement depends on the variables' data type (e.g. BOOL/SAFEBOOL: 1 Bit, WORD/SAFEWORD: 16 Bits).
- **Variables with directly derived data type**
The bit requirement corresponds to the bit requirement of the elementary data type from which the variable was derived.
- **Variables of ARRAY data type**
The bit requirement corresponds to the sum of bits needed for each element. For example, the bit requirement of a one-dimensional array can be calculated by multiplying an element's bit requirement by the number of elements.
In terms of data consistency, the system ensures that the elements of an array are not split over different connections.

- **Variables of STRUCT data type**
The bit requirement corresponds to the sum of bits needed for each element.
In terms of data consistency, the system ensures that the elements of a structure are not split over several connections.
- **Variables with VALID extension**
The variables' bit requirement is increased by 1 Bit. The valid bit is transmitted once only. This also applies for a variable of an I/O data type.

6.6 Network without Routers

There are a number of rules to be observed when designing a SafetyNET p network for a project. If you are not using Routers, the following applies:



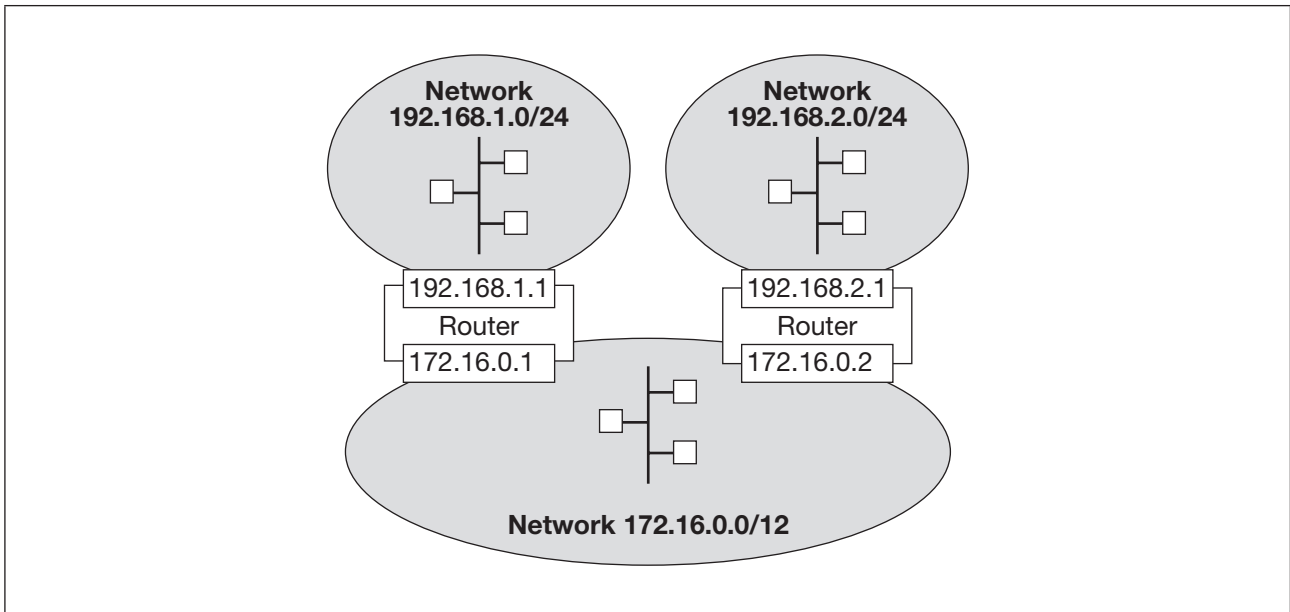
Example for an Ethernet network without Router

- ▶ Scan
 - In PAS4000 there are various options for finding SafetyNET p subscribers (e.g. scan network, scan project).
 - The "Scan network" process finds all the PSS 4000-exclusive/operable devices in a network connected to the same network adapter as the programming interface
 - The "Scan project" process finds all the PSS 4000-exclusive/operable devices that have been configured within the project and are located in a network connected to the same network adapter as the programming interface
- ▶ Device naming via PAS4000
 - Device naming can be performed on devices found during the scanning process.
- ▶ Process data exchange
 - To enable process data exchange between the PSS 4000 devices within a project, the devices must be in the same subnetwork. Switches and hubs can be interconnected.
- ▶ Other online actions
 - Other online actions include all online actions with the exception of scanning and device naming, e.g.:
 - Project download
 - Diagnostics (diagnostic list, diagnostic log)
 - Debugging (show, control, force variables; dynamic program display, etc.)

To enable these online actions to be performed, the programming interface on the PAS4000-PC must be in the same subnetwork as the project's PSS 4000 devices. Switches and hubs can be interconnected.

6.7 Network with Routers

On Ethernet, a Router connects two different Ethernet networks. As an Ethernet network is identified by the Net-ID (= part of the IP address), the Net-ID can be used to decide whether the subscriber is in the same network or whether data packets must be transferred to a Router to be forwarded.



Example for connecting Ethernet networks using Routers

If Routers are used in a SafetyNET p network, please note the following:

- ▶ **Requirements of Routers**
Only Routers that support a maximum transmission unit (MTU) of min. 296 Bytes can be used in a SafetyNET p network.
- ▶ **Scan**
In PAS4000 there are various options for finding SafetyNET p subscribers (e.g. scan network, scan project).
To prevent the system searching for all potential PSS 4000 devices worldwide, the scan function on PAS4000 works on the basis of an IP broadcast. The default setting for "Time To Live" (TTL = 1) prevents scanning across Routers with PAS4000 (see System variable PILZ_SNP_MULTICAST_TTL, [Page 100](#)).
- ▶ **Device naming via PAS4000**
If networks are connected via Routers, the default setting for "Time To Live" (TTL = 1) must have device naming performed directly from the respective network and not performed centrally (see System variable PILZ_SNP_MULTICAST_TTL, [Page 100](#)).
- ▶ **Process data exchange**
Provided a Router is configured correctly (e.g. Routing table), process data exchange will also work with PSS 4000-exclusive/operable devices which, from the perspective of the PAS4000-PC, are in a network behind a Router.

▶ Other online actions

Provided a Router is configured correctly (e.g. Routing table), the other online actions will also work for PSS 4000-exclusive/operable devices which, from the perspective of the PAS4000-PC, are in a network behind a Router. Other online actions include all on-line actions with the exception of scanning and device naming, e.g.:

- Project download
- Diagnostics (diagnostic list, diagnostic log)
- Debugging (show, control, force variables; dynamic program display, etc.)

6.7.1 System variable PILZ_SNP_MULTICAST_TTL

Time To Live (TTL) is a property of the Internet Protocol (IP). TTL can be used to restrict the routing of IP packets to other networks.

Each router that passes an IP packet on the way from the sender to the transmitter to the address reduces the TTL value of the IP packet by 1. An IP packet whose value falls to 0 is not transmitted by a router and it is discarded.

Scanning and device naming are performed depending on the settings of the relevant TTL. The TTL value for these two actions can be configured on the PAS4000 PV via the system variable PILZ_SNP_MULTICAST_TTL.

Beside the correct configuration of a router, the setting for PILZ_SNP_MULTICAST_TTL is the key factor for reaching the PSS 4000 devices that are behind a router, as seen from the PAS4000 PC, when scanning and naming devices.



CAUTION!

Please note that if `PILZ_SNP_MULTICAST_TTL > 1` it is possible to intervene across all routers in other networks. This may jeopardise network safety.

With the configuration of the system variable `PILZ_SNP_MULTICAST_TTL`:

- ▶ Default TTL when system variable `PILZ_SNP_MULTICAST_TTL` is not available: 1
To prevent the system searching for all PSS 4000 devices worldwide, the default setting prevents scanning across Routers and device naming across Routers.
It is not necessary to create and to configure the system variable `PILZ_SNP_MULTICAST_TTL` when PAS4000 is to use the default setting. PAS4000 automatically uses the default setting when the system variable is not found at startup.
- ▶ System variables are configured in the system environment with the Windows operating systems (e. g. Windows XP, Windows Vista, Windows 7)
Example for Windows XP:
Start -> Control Panel -> System -> Advanced tab -> Environment Variables button
- ▶ valid configuration for `PILZ_SNP_MULTICAST_TTL`: integer value in the range of 1 ... 64
 - When the configuration is invalid (e. g. letters, value is not an integer) the default setting is used for `PILZ_SNP_MULTICAST_TTL`.
 - If a value > 64 is configured, the value 64 is used for `PILZ_SNP_MULTICAST_TTL`.
- ▶ The creation/change of the system variables `PILZ_SNP_MULTICAST_TTL` will not become effective until the PAS4000 PC is restarted.



CAUTION!

If the OPC Server is installed on the same PC as PAS4000, the OPC server sets the system variable `PILZ_SNP_MULTICAST_TTL` to 64 during installation.

6.8 Prevent data exchange with projects with identical project name

Data exchange between projects is not permitted and it is prevented by PSS 4000 for FS data. However, it can happen that projects in which FS data are processed have identical project names. If such projects are operated in the same SafetyNET p network, PSS 4000 cannot prevent cross-connections in the data exchange without additional measures.

Operate projects with identical project name and processing of FS data in closed SafetyNET p networks. A SafetyNET p network will be deemed as "closed" when it ensures that no SafetyNET p data traffic takes place beyond network limits.



Information

If you can **not** ensure that the projects are in closed SafetyNET p networks, the projects may not have identical project names. Rename the projects.

7 External communication

7.1 Communication modes for external communication

External communication refers to data exchange in which output data (external O-data) leaves the automation system PSS 4000 and input data (external I-data) enters the system. This is achieved with the help of various communication modes. The communication mode supported by a PSS 4000 device depends on the device. Details can be found in the operating manual of the relevant PSS 4000 device.

Communication modes for external communication are divided into "Fieldbuses" and "IP connections".

Fieldbuses

The term "Fieldbuses" embraces industrial fieldbuses with a standardised protocol (e.g. PROFIBUS-DP). For external communication via a fieldbus, a PSS 4000 device must have the corresponding system section (e.g. PROFIBUS-DP Slave system section). The physical connection is established via defined interfaces, depending on the fieldbus.

IP connections

The term "IP connections" embraces all protocols and communication links that are addressed via IP addresses (Modbus/TCP, Raw UDP). For this type of external communication a PSS 4000 device must have the "IP connections" system section. The physical connection is established via an Ethernet interface.

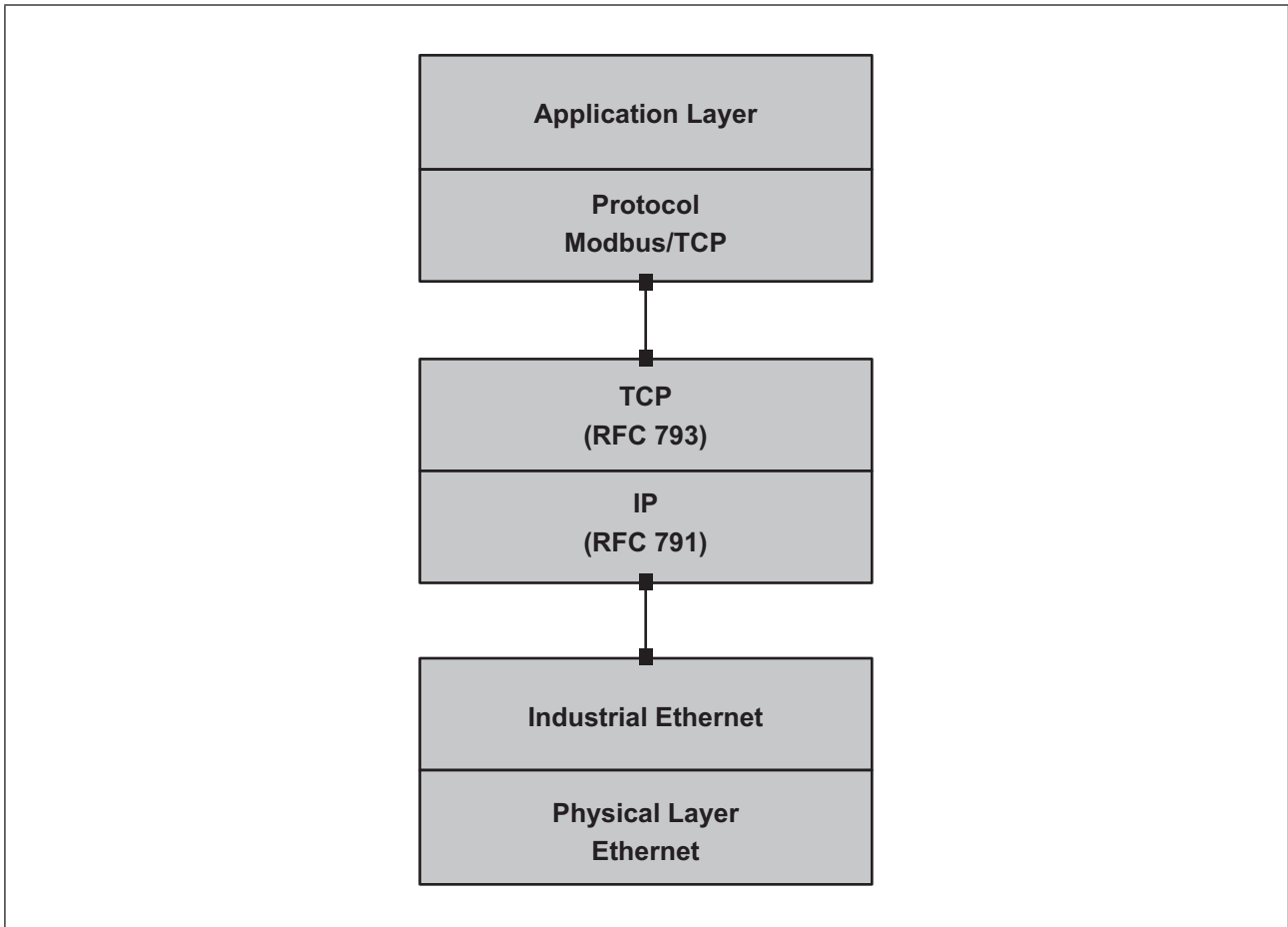


CAUTION!

The following applies for the **max. number of IP connections**:
The "IP connections" system section on a PSS 4000 device can manage a maximum of 32 IP connections in total. This upper limit also applies if you are operating the "IP connections" system section of a PSS 4000 device with a variety of protocols or communication links.
SafetyNET p connections are **not** managed by the "IP connections" system section and so do not affect this upper limit.

7.2 Modbus/TCP

Modbus is an open fieldbus standard published by the User Group MODBUS-IDA (see www.Modbus-IDA.org).



Communication of a PSS 4000 device via Modbus/TCP

Modbus/TCP is a protocol based on Industrial Ethernet (TCP/IP over Ethernet). It is one of the protocols with Client/Server communication. Data is transferred via a request/response mechanism using function codes (FC).

Modbus/TCP is connection-oriented, i.e. before usable data is transferred via Modbus/TCP, a connection must first be established between two Modbus/TCP interfaces. The initiator of the connection is called the "Client". The communications partner with which the Client establishes the connection is called the "Server". Whether a connection on a device assumes the role of Client or Server is defined when a connection is configured. As a result, the Server/Client role only applies for that specific connection.

Data exchange via Modbus/TCP can be used on the automation system PSS 4000 for non-safety-related applications.

7.2.1 "IP connections" system section

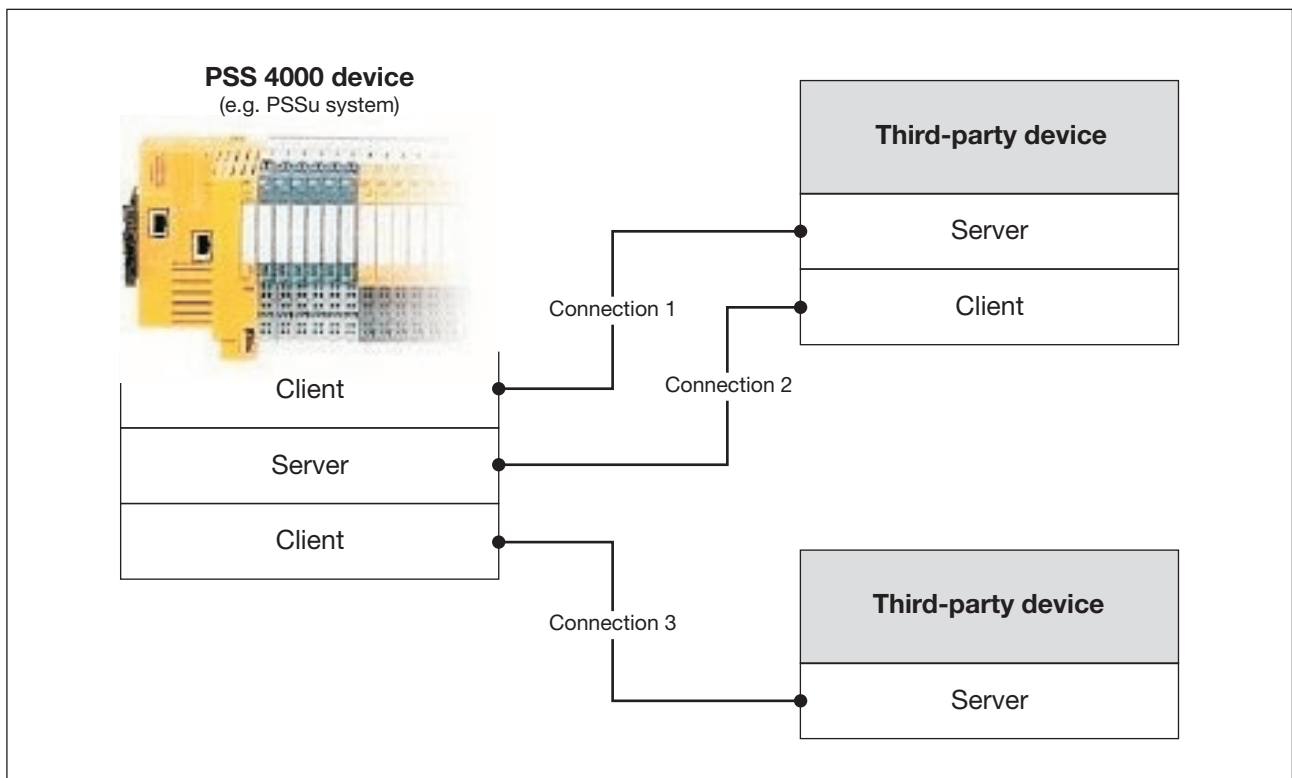
In the automation system PSS 4000, external communication via Modbus/TCP is possible with all PSS 4000 devices that have the "IP connections" system section. In this case, the "IP connections" system section of a PSS 4000 device performs all the tasks required for communication. For example, these include:

- ▶ Management of the Modbus/TCP connections(see Modbus/TCP connections, [Page 105](#))
- ▶ Management of the Modbus/TCP data areas(see Modbus/TCP data areas, [Page 106](#))
- ▶ Support for function codes (see Function codes, [Page 107](#))
- ▶ Data exchange (see Data exchange via Modbus/TCP, [Page 111](#))

7.2.2 Modbus/TCP connections

The "IP connections" system section of a PSS 4000 device manages the Modbus/TCP connections that are configured for the device. The connection is configured in the PAS4000's IP Connections Editor.

Several Modbus/TCP connections can be configured for a PSS 4000 device (see Data transfer limits, [Page 109](#)). When a connection is configured you have the option to configure the connection for the role of Client or Server. If a Client role is configured for a connection, this is called a "Client connection" from the perspective of the PSS 4000 device, otherwise it is called a "Server connection" from the perspective of the PSS 4000 device.



Role assignment on Modbus/TCP connections (example)

Port numbers must be defined for data exchange via a Modbus/TCP connection. Server connections on a PSS 4000-exclusive device use Port 502 as a default setting. However, a different port number may be configured.



Information

Known applications (e.g. Telnet) are assigned fixed port numbers, to which each application can refer during data transfer. The range 0 ... 1023 contains reserved port numbers. Under no circumstances should these be used for your own applications. The complete list of reserved port numbers and their utilities are listed in RFC 1700.

Send and receive data is transferred cyclically on PSS 4000-exclusive devices. With cyclical transfer, a permanent connection is maintained between the connection Client and Server.

7.2.3 Modbus/TCP Server connection (server connections)

The "IP connections" system section of a PSS 4000 device supports the following Modbus/TCP data areas for connections with a Server role:

Data area	Modbus syntax	Example
Coils (Bit) 0x00000 ... 0x65535 [read/write]	0x[xxxxx]	0x05888
Discrete Inputs (Bit) 1x00000 ... 1x65535 [read only]	1x[xxxxx]	1x05888
Input Register (Word/16 Bits) 3x00000 ... 3x65535 [read only]	3x[xxxxx]	3x05888
Holding Register (Word/16 Bits) 4x00000 ... 4x65535 [read/write]	4x[xxxxx]	4x05888



Information

The "IP connections" system section of a PSS 4000 device starts the addressing for Modbus/TCP data areas at "0". On devices from other manufacturers, addressing may start at "1".

Please refer to the operating manual provided by the relevant manufacturer.

7.2.4 Function Codes (Client connections)

The "IP connections" system section of a PSS 4000 device supports the following function codes (FC) for connections with a Client role:

Function code	Function	
FC 01	Read Coils	The connection Client reads bit data from the connection Server, data length \geq 1 Bit, e.g. digital inputs (receive data from 0x)
FC 02	Read Discrete Input	The connection Client reads bit data from the connection Server, data length \geq 1 Bit, e.g. digital inputs (receive data from 1x)
FC 03	Read Holding Register	The connection Client reads word data from the connection Server, data length \geq 1 Word, e.g. analogue inputs, counter inputs (receive data from 4x)
FC 04	Read Input Register	The connection Client reads word data from the connection Server, data length \geq 1 Word, e.g. analogue inputs, counter inputs (receive data from 3x)
FC 05	Write Single Coil	The connection Client writes to one bit datum in the connection Server, data length = 1 Bit, e.g. digital input (send data to 0x)
FC 06	Write Single Register	The connection Client writes to one word datum in the connection Server, data length = 1 Word, e.g. analogue output, counter output (send data to 4x)
FC 15	Write Multiple Coils	The connection Client writes to multiple bit data in the connection Server, data length > 1 Bit, e.g. digital outputs (send data to 0x)
FC 16	Write Multiple Registers	The connection Client writes to multiple word data in the connection Server, data length > 1 Word, e.g. analogue inputs, counter inputs (send data to 4x)
FC 23	Read/Write Multiple Registers	The connection Client reads and writes multiple word data within a telegram (receive data from 4x and send data to 4x)



Information

A function code must be configured for each Client connection. This can be done in the PAS4000's IP Connections Editor.

7.2.5 Relationship between function codes and Modbus/TCP data areas

For Client connections, the following relationship exists between function codes and Modbus/TCP data areas:

Data area (Remote Address)	Send (Client)		Receive (Client)
0x Coils (Bit)	Data length	Function code	Function code
	= 1Bit	FC = 05 (Write Single Coil)	FC = 01 (Read Coils)
	> 1 Bit	FC = 15 (Write Multiple Coils)	
1x Discrete Inputs (Bit)	---		FC = 02 (Read Discrete In-puts)
3x Input Register (Wort)	---		FC = 04 (Read Input Register)
4x Holding Register (Word)	Data length	Function code	FC = 03 (Read Holding Regis- ters)
	= 1 Word	FC = 06 (Write Single Regis- ter)	
	> 1 Word	FC = 16 (Write Multiple Regis- ters)	
	If there is an entry for both send and receive data, function code FC = 23 is used (Read/Write Multiple Registers).		

7.2.6 Data transfer limits

Max. number of Modbus/TCP connections

The max. number of Modbus/TCP connections (Client connections and/or Server connections) depends on the device. Details can be found in the operating manual of the relevant PSS 4000 device.

Please also note the general upper limit of 32 IP connections for one PSS 4000 device.



CAUTION!

The following applies for the **max. number of IP connections**:

The "IP connections" system section on a PSS 4000 device can manage a maximum of 32 IP connections in total. This upper limit also applies if you are operating the "IP connections" system section of a PSS 4000 device with a variety of protocols or communication links.

SafetyNET p connections are **not** managed by the "IP connections" system section and so do not affect this upper limit.

Limits on Client connections

On Client connections, the maximum data length per telegram is restricted. The following table contains information on the maximum data length per telegram that is supported:

Data transfer		Data length per telegram
Read data (Bit)	FC 01 (Read Coils)	1 ... max. 2000
	FC 02 (Read Discrete Inputs)	
Read data (Bit)	FC 05 (Write Single Coil)	1 Bit
	FC 15 (Write Multiple Coils)	1 ... max. 1968
Read data (Word)	FC 03 (Read Holding Registers)	1 ... max. 125
	FC 04 (Read Input Register)	
Write data (Word)	FC 06 (Write Single Register)	1 Word
	FC 16 (Write Multiple Registers)	1 ... max. 123 Words
Read and write data (Word)	FC23 (Read/Write Multiple Registers)	Read 1 ... max. 125 Words Write 1 ... max. 121 Words

FC 03, FC 04 and FC 16 support data transfer with multiple telegrams. If the max. supported data length per telegram is exceeded when transferring data with FC 03, FC 04 or FC 16, the data will be spread over multiple telegrams. The following table contains details of the data length when transferring data with multiple telegrams:

Data transfer with multiple telegrams		Data length
Read data (Word)	FC 03 (Read Holding Registers)	From 126 ... max. 2000 Words
	FC 04 (Read Input Register)	
Write data (Word)	FC 16 (Write Multiple Registers)	From 124 ... max. 2000 Words

When a Client connection is configured in PAS4000 it is possible to influence the transfer of multiple telegrams. If multiple telegrams are to be transferred, you can configure whether it is necessary to wait for a response to one (sub)telegram before sending the next (sub)telegram or whether to send all (sub)telegrams one after the other, without waiting for a response. The responses to the (sub)telegrams will only be collected once the transfer is complete, which optimises data transfer.



Information

When devices from a third-party manufacturer are used as communication partners, limits regarding data lengths may vary. Please refer to the information stated in the operating manual of the relevant unit.

7.2.7 Data exchange via Modbus/TCP

It is always the connection Client that initiates data transfer. For a Client connection, the direction of the data flow (send/receive) and data area (Coils, Discrete Inputs, Input Registers, Holding Register) is established by defining the function code when the connection is configured. The data length is also defined when the connection is configured.

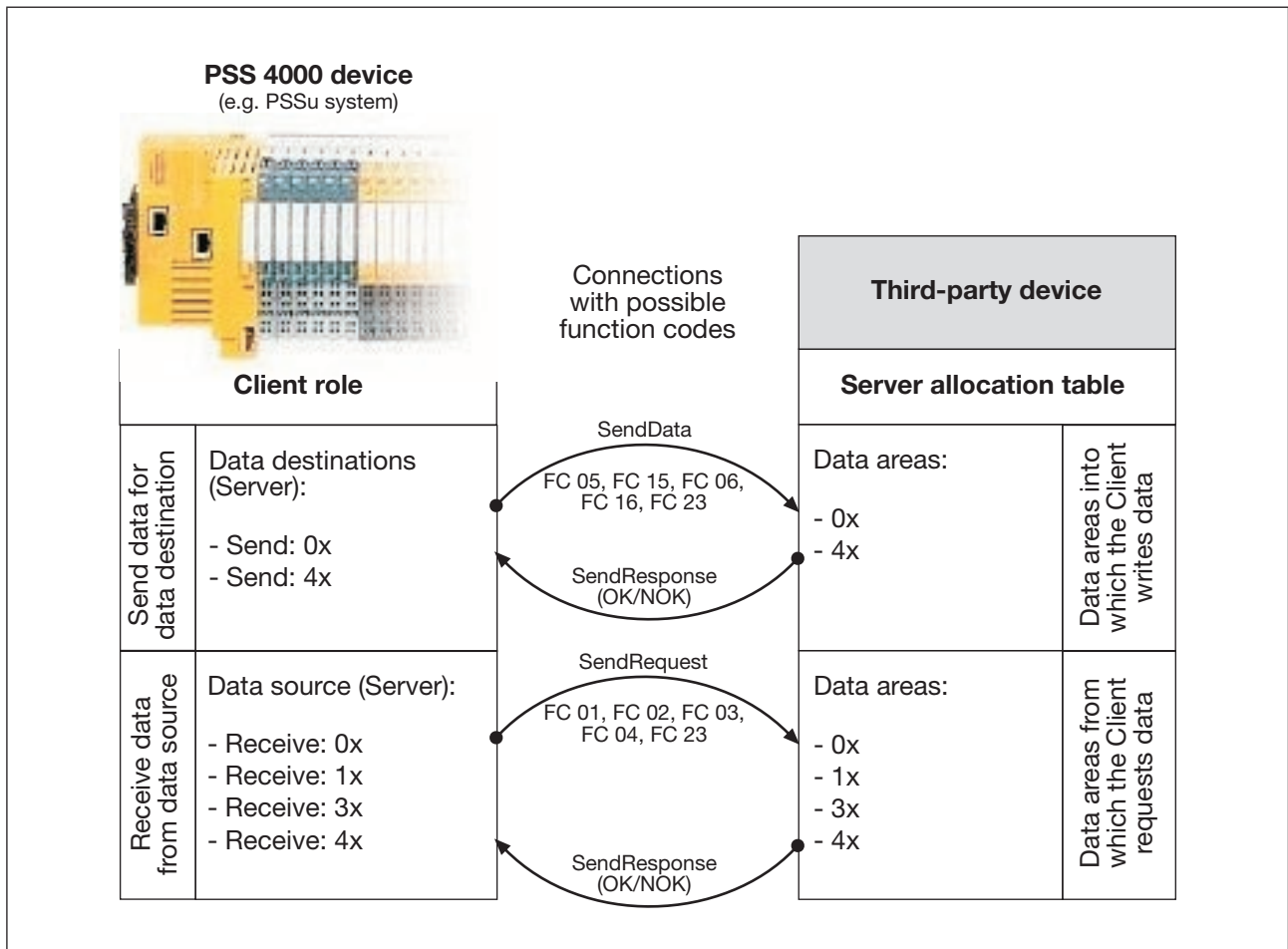
7.2.7.1 Send data and receive data

When sending, the connection Client sends data to a data area belonging to the connection Server. This is also known as "Writing data". For this to work, the target data destination must be configured on the device with the role of Client.

When receiving, the connection Client requests and receives data located in a data area belonging to the connection Server. This is also known as "Reading data". For this to work, the data source must be configured on the device with the role of Client.

From the perspective of the automation system PSS 4000, receive data is called external I-data, and send data is called external O-data (see Validity of process data, [Page 202](#)).

Please ensure that the relevant data area (data destination, data source) is available on the Server (see Server allocation table, [Page 112](#)).



PSS 4000 device with "IP connections" system section and Client role for connections (principle)

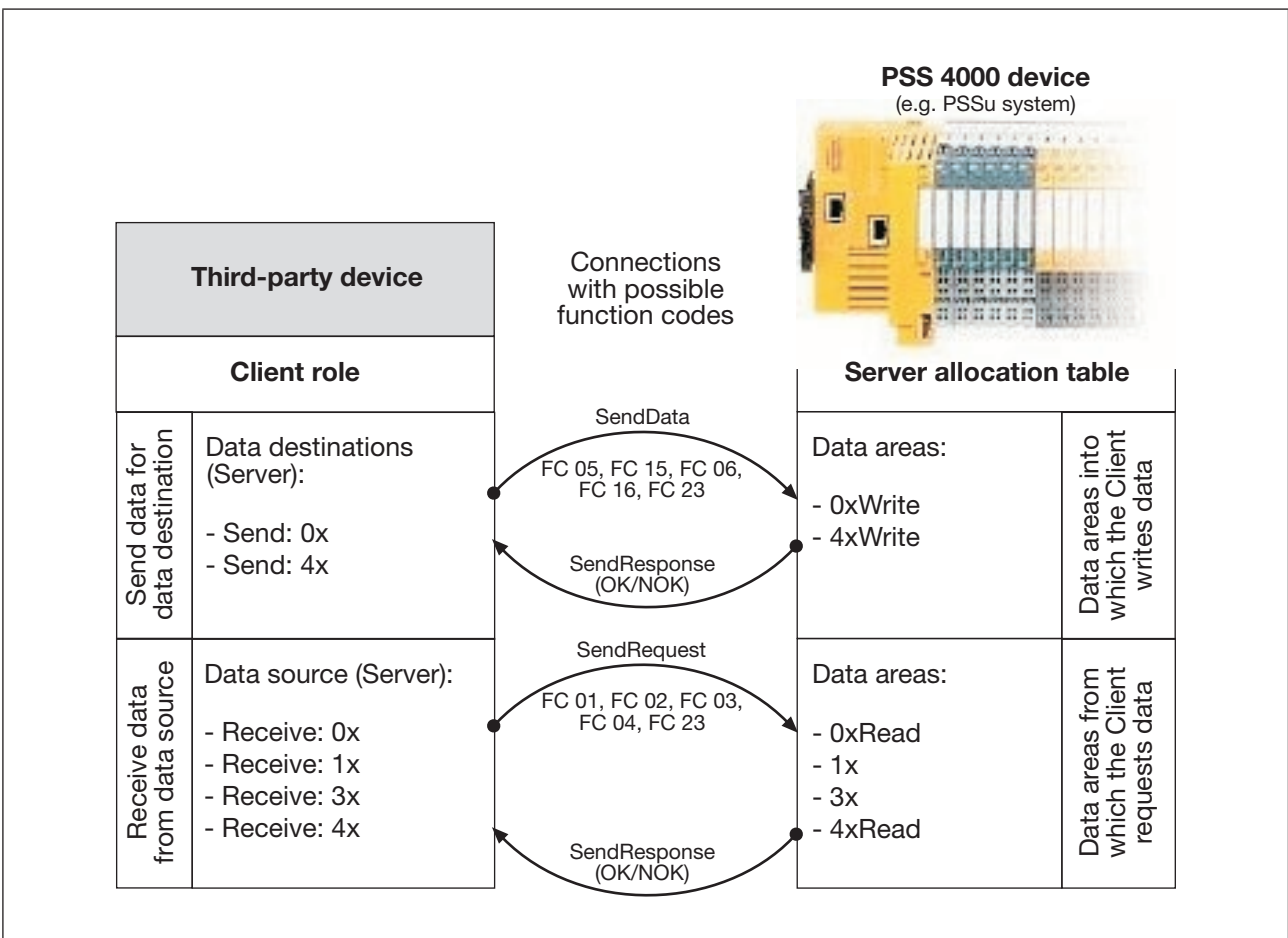
7.2.7.2 Server allocation table

To ensure that the connection Client can send data to the connection Server and receive data requested from the connection Server, the relevant data areas must be configured on the Server.

With Server connections this is achieved by configuring a Server allocation table when the connection is configured.

The following applies:

- ▶ The connection Client writes data to the Server allocation table's write data area (0xWrite, 4xWrite).
- ▶ The connection Client requests data from the Server allocation table's read data area (0xRead, 1x, 3x, 4xRead).



PSS 4000 device with "IP connections" system section and Server role for connections (principle)

Limits when configuring a server allocation table in PAS4000

Each data area can have a start address in the range 0 ... (65535 – data length) and a maximum data length of 8000:

Data area	Start address	Max. data length
0xRead	0 ... (65535 – data length)	8000
0xWrite		
1x		
3x		
4xRead		
4xWrite		

Example structure of a server allocation table in PAS4000:

Data area	Start address	Data length
0xRead	0	4
0xWrite	100	4
1x	0	0
3x	0	0
4xRead	0	2
4xWrite	100	2

7.2.8 Modbus/TCP I/O data

The "IP connections" system section enables a PSS 4000 device to exchange I/O data with third-party devices via Modbus/TCP. For this to work it is necessary for the I/O data in PAS4000's I/O Mapping Editor to be mapped to the corresponding Modbus/TCP data areas.

I/O mapping of Modbus/TCP data areas

I/O mapping is used to define which data source supplies data to which data sink.

Possible I/O mapping for Modbus/TCP data areas:

		Modbus/TCP			
		Client role		Server role (Server allocation table)	
		Send data areas (0x, 4x)	Receive data areas (0x, 1x, 3x, 4x)	Read data areas (0x, 1x, 3x, 4x)	Write data areas (0x, 4x)
Module bus	I-datum (e.g. HW input)	◆	---	◆	---
	O-datum (e.g. HW output)	---	◆	---	◆
PI variables	I-PI variable	---	◆	---	◆
	O-PI variable	◆	---	◆	---

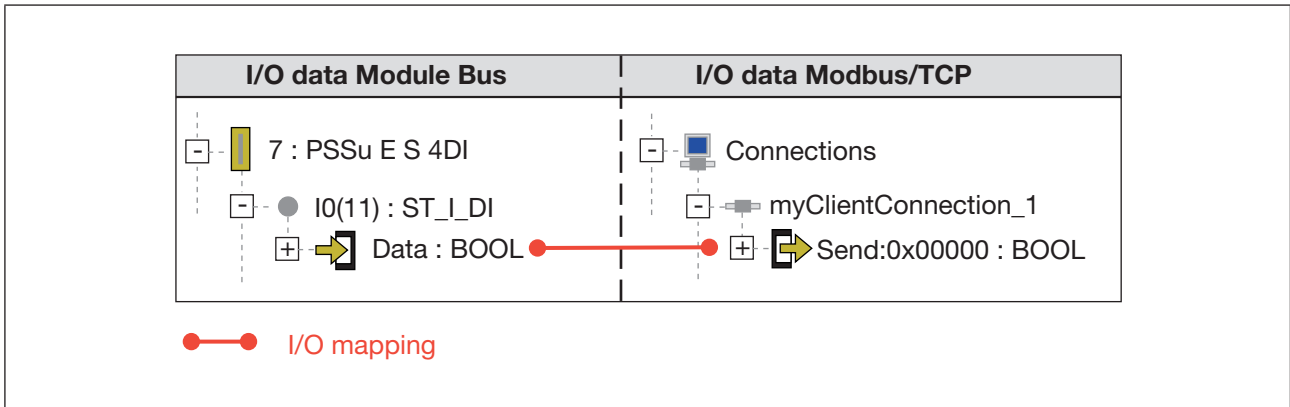
◆ I/O mapping possible

--- I/O mapping impossible

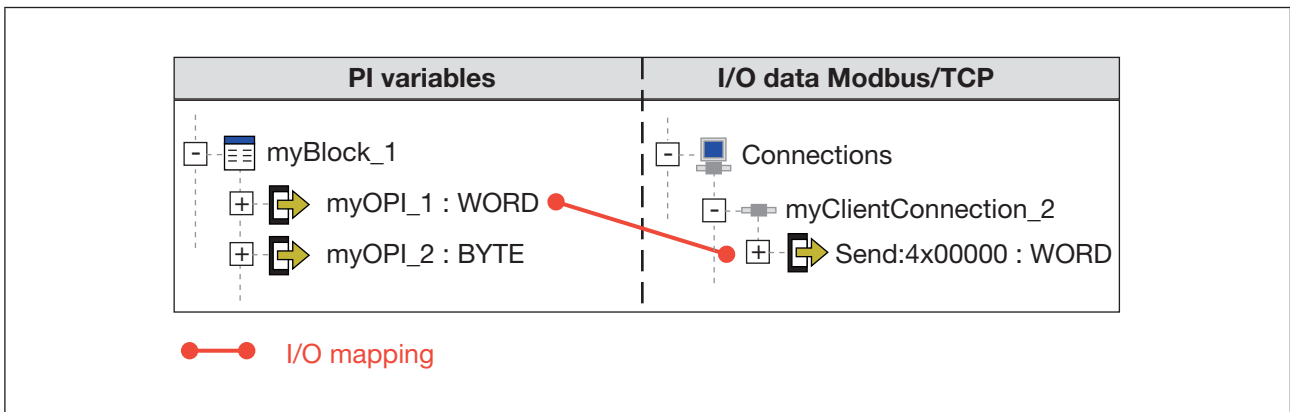
Examples:

I/O mapping for Client roles

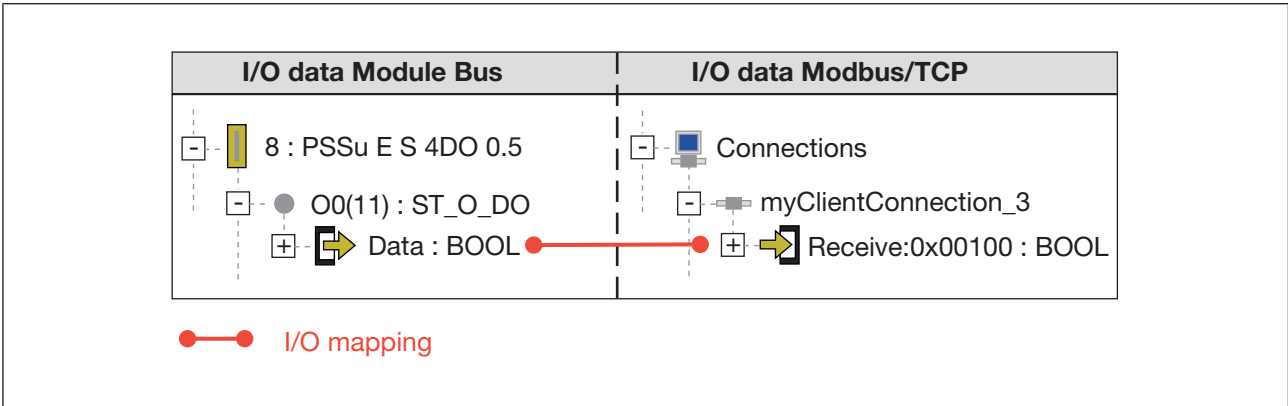
- ▶ The input signal at a hardware input is to be sent to a device from another manufacturer via a Modbus/TCP connection.
 - Direction of data flow: Hardware input -> Modbus/TCP
 - Used function code: FC05/FC15 (send)
 - Possible I/O mapping (example):



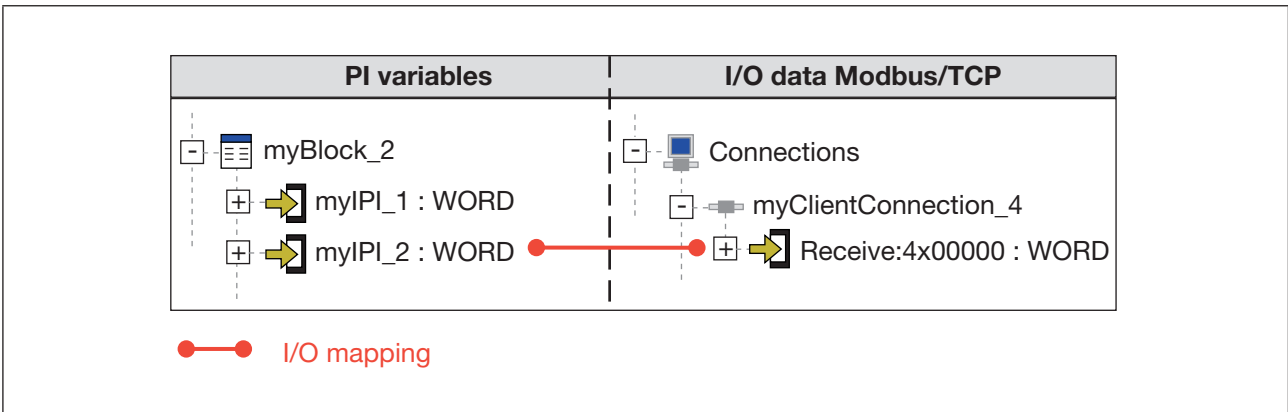
- ▶ The content of an O-PI variable from the user program is to be sent to a device from another manufacturer via a Modbus/TCP connection.
 - Direction of data flow: O-PI variable -> Modbus/TCP
 - Used function code: FC06/FC16/FC23 (send)
 - Possible I/O mapping (example):



- ▶ The hardware output on a PSSu system is to be controlled by a device from another manufacturer via a Modbus/TCP connection.
 - Direction of data flow: Modbus/TCP -> Hardware output
 - Used function code: FC01 (receive)
 - Possible I/O mapping (example):

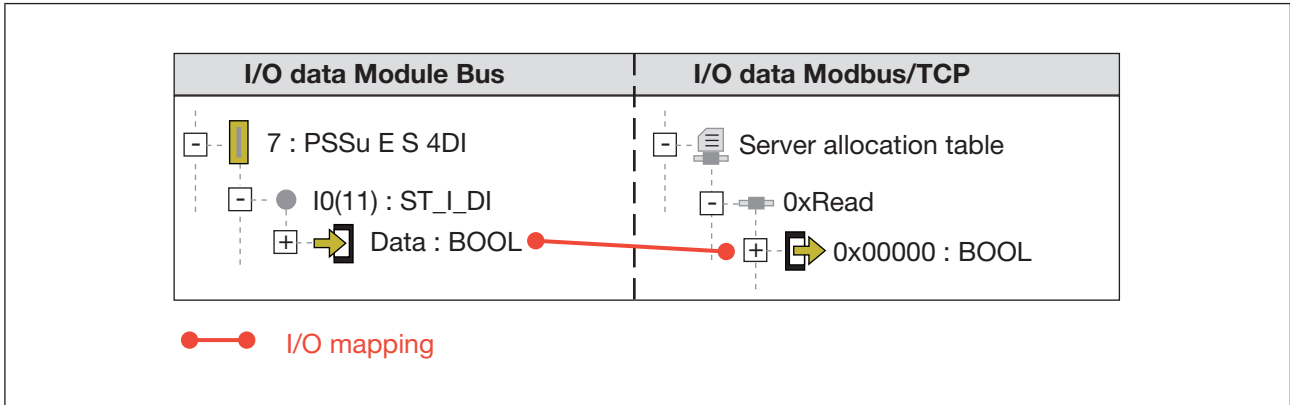


- ▶ A piece of data is to be requested by a device from another manufacturer via a Modbus/TCP connection. Once the data has been received it is to be made available for further processing within the user program.
 - Direction of data flow: Modbus/TCP -> I-PI variable
 - Used function code: FC03/FC23 (receive)
 - Possible I/O mapping (example):

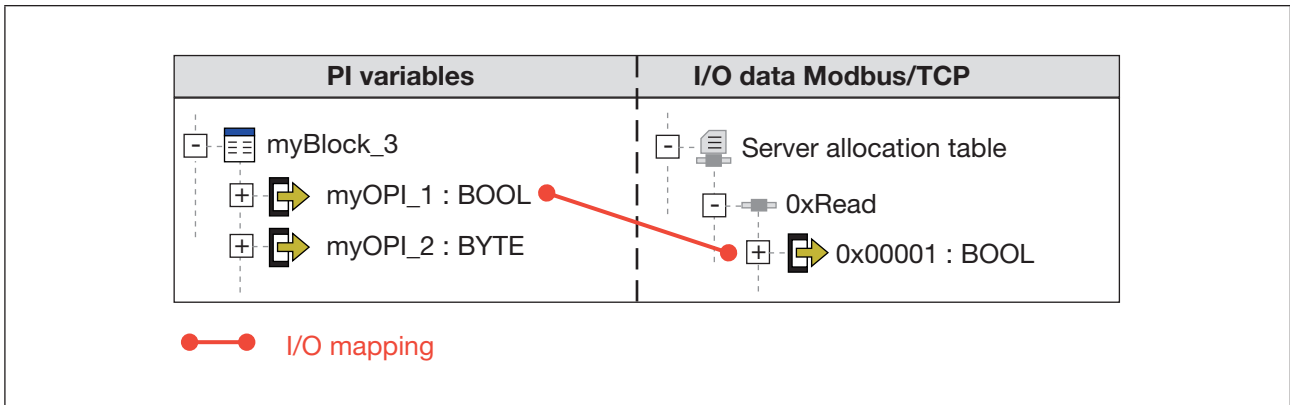


I/O mapping for Server roles (Server allocation table)

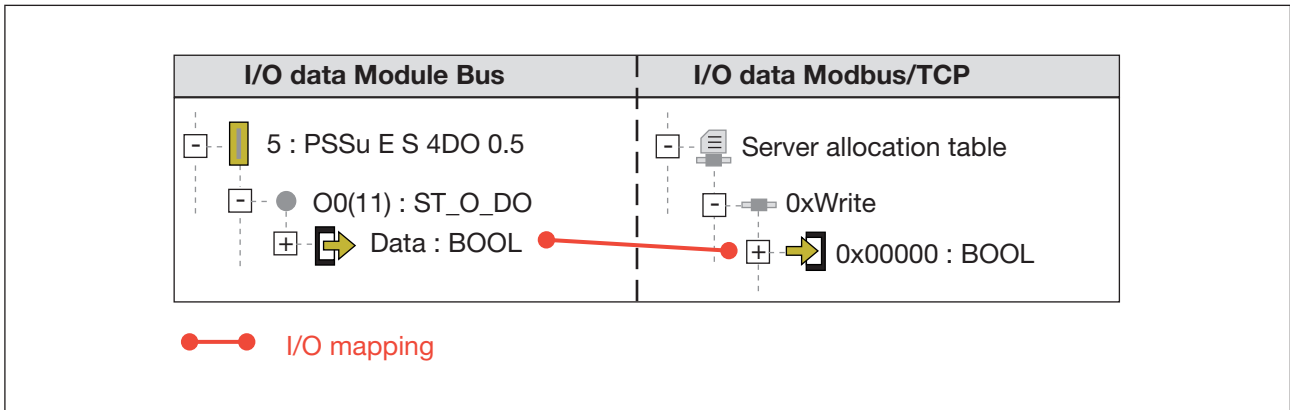
- ▶ The signal from a hardware input is made available in the Server allocation table's read data area. The piece of data can be read by a device from another manufacturer via a Modbus/TCP connection.
 - Direction of data flow: Hardware input -> Modbus/TCP
 - Possible I/O mapping:



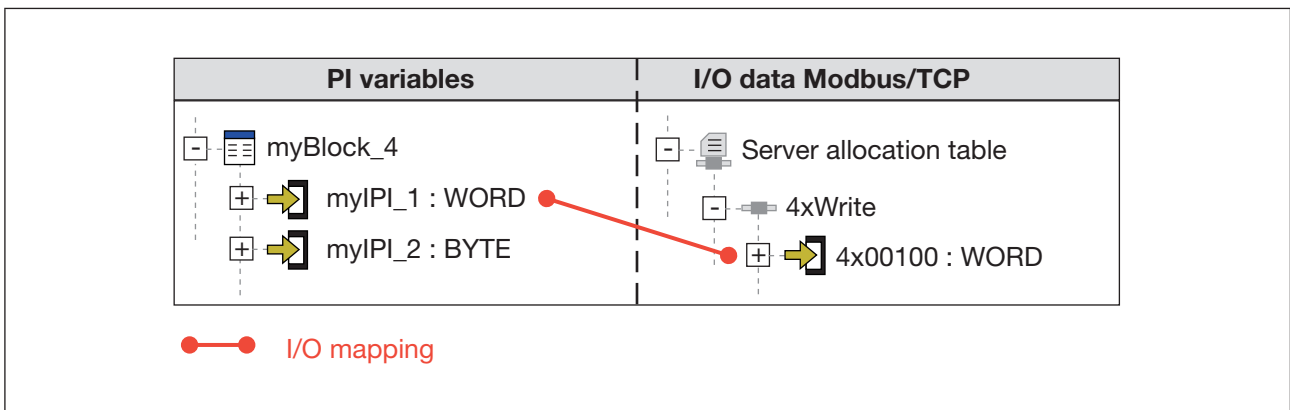
- ▶ The content of an O-PI variable is made available in the Server allocation table's read data area. The piece of data can be read by a device from another manufacturer via a Modbus/TCP connection.
 - Direction of data flow: O-PI variable -> Modbus/TCP
 - Possible I/O mapping:



- ▶ The hardware output on a PSSu system is to be controlled by a device from another manufacturer via a Modbus/TCP connection. The device from another manufacturer writes to the Server allocation table's write data area.
 - Direction of data flow: Modbus/TCP -> Hardware output
 - Possible I/O mapping:



- ▶ A piece of data is to be written to the Server allocation table's write data area by a device from another manufacturer via a Modbus/TCP connection. The data is to be processed further within the user program.
 - Direction of data flow: Modbus/TCP -> I-PI variable
 - Possible I/O mapping:



7.2.9 Address display of Modbus/TCP addresses

The addresses for the I/O data of a PSS 4000 device are displayed at several points, e.g. in PAS4000 and in the diagnostics.

7.2.9.1 Address format of send data

The addresses of Modbus/TCP send data (Client role) are structured as follows:

<Device name>.Modbus.Client.<Connection name>.Send.<Start address>

Examples:

myDevice_1.Modbus.Client.myConnection1.Send.0x00000

myDevice_1.Modbus.Client.myConnection2.Send.4x00005

7.2.9.2 Address format of receive data

The addresses of Modbus/TCP receive data (Client role) are structured as follows:

<Device name>.Modbus.Client.<Connection name>.Receive.<Start address>

Examples:

myDevice_1.Modbus.Client.myConnection3.Receive.0x00000

myDevice_1.Modbus.Client.myConnection4.Receive.1x00010

myDevice_1.Modbus.Client.myConnection5.Receive.3x00000

myDevice_1.Modbus.Client.myConnection6.Receive.4x00005

7.2.9.3 Address display in the Server allocation table

The addresses of the Modbus/TCP Server allocation table (Server role) are structured as follows:

<Device name>.Modbus.Server.<Selectable data area in the Server allocation table>.<Start address>

Examples:

myDevice_2.Modbus.Server.0xRead.0x00000

myDevice_2.Modbus.Server.0xWrite.0x00100

myDevice_2.Modbus.Server.1x.1x00010

myDevice_2.Modbus.Server.3x.3x00000

myDevice_2.Modbus.Server.4xRead.4x00005

myDevice_2.Modbus.Server.4xWrite.4x00005

7.2.10 Data consistency on Modbus/TCP

The data consistency of data sent (Client role) or provided (Server role) by third party devices is the responsibility of those devices. From the PSS 4000's perspective, no statement can be made regarding the data consistency of those devices.

Data consistency on receive data from a Client connection

If a PSS 4000-exclusive/operable device is the connection Client, from the device's perspective, the receive data from this connection (FC 01, FC 02, FC 03, FC 04, FC 23) is external I-data.

The external I-data from a Client connection is treated as consistent data in PSS 4000, (see Data consistency, [Page 214](#)) provided it is transferred in a single telegram (see Data transfer limits (Client connections), [Page 109](#)).

The external I-data from a Client connection forms a consistency range of the consistency range type "ST-I data of an external communication connection". The exact description is consistency range "ST-I data of a Modbus/TCP Server connection" (see Consistency ranges, [Page 215](#)).

Data consistency on receive data from a Server connection

If a PSS 4000-exclusive/operable device is the connection Server, from the device's perspective, the receive data from this connection (data areas 0xWrite or 4xWrite) is external I-data.

The external I-data from a Server connection is treated as consistent data in PSS 4000, (see Data consistency, [Page 214](#)) provided it is transferred in a single telegram (see Data transfer limits (Client connections), [Page 109](#)).

The external I-data from a Server connection forms a consistency range of the consistency range type "ST-I data of an external communication connection". The exact description is consistency range "ST-I data of a Modbus/TCP Server connection" (see Consistency ranges, [Page 215](#)).

Data consistency on a Client connection's send data

If a PSS 4000-exclusive/operable device is the connection Client, from the device's perspective, the send data from this connection is external O-data.

The external O-data of a Client connection (FC 05, FC 15, FC 06, FC 16, FC 23) is consistent if it comes from the same consistency range on a data source (see Data consistency, [Page 214](#)).

Data consistency on a Server connection's send data

If a PSS 4000-exclusive/operable device is the connection Server, from the device's perspective, the send data (data areas 0xRead, 1x, 3x, 4xRead) is external O-data. The external O-data of a Server connection is consistent if it comes from the same consistency range on a data source (see Data consistency, [Page 214](#)).

7.2.11 Special features when using the data areas 0x and 4x

On some devices (e.g. operator terminals such as Pilz PMI), Client connections are used to write data to the data areas (0x, 4x) of the Server allocation table and then automatically request the data back immediately, to show on the display for example.

If the connection's Server is a PSS 4000-exclusive/operable device, this "data mirroring" will only work if additional measures are taken for the device: There are minor differences in handling for IEC 61131 programming and Multi programming.

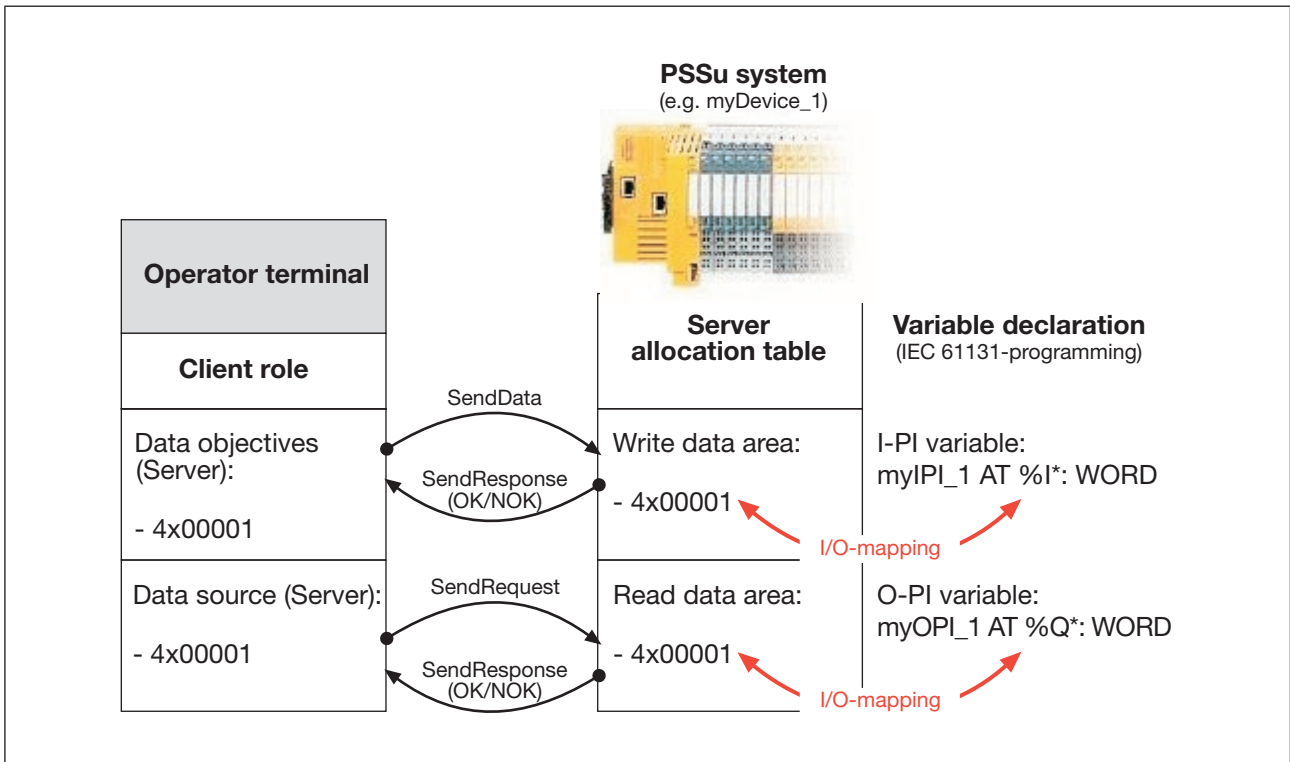
Procedure for IEC 61131 programming

- ▶ Create a block
e.g. program-type POU
- ▶ Declare the required I-PI variables in the block
Via the I-PI variables, I/O-mapping is used to access the write data areas (0xWrite, 4xWrite) of the Server allocation table.
e.g. myIPI_1 AT %I* : WORD
- ▶ Declare the corresponding O-PI variables in the block
The O-PI variables are needed in order to restore the I-PI variables in the block. This is a condition for "data mirroring" the I-data to the Read data areas (0xRead, 4xRead) of the Server allocation table.
e.g. myOPI_1 AT %Q* : WORD
- ▶ "Mirror data" by restoring the I-PI variables to the O-PI variables in the block

Example (programming in IL):

```
RESTORE:           LD           myIPI_1
                  ST           myOPI_1
```

- ▶ I/O-map the I-PI variables to the write data areas (0xWrite, 4xWrite) of the Server allocation table (I/O Mapping Editor of PAS4000)
- ▶ I/O-map the O-PI variables to the read data areas (0xRead, 4xRead) of the Server allocation table (I/O Mapping Editor of PAS4000)
- ▶ Assign the block to a resource from a PSS 4000-exclusive/operable device
e.g. ST resource from myDevice_1

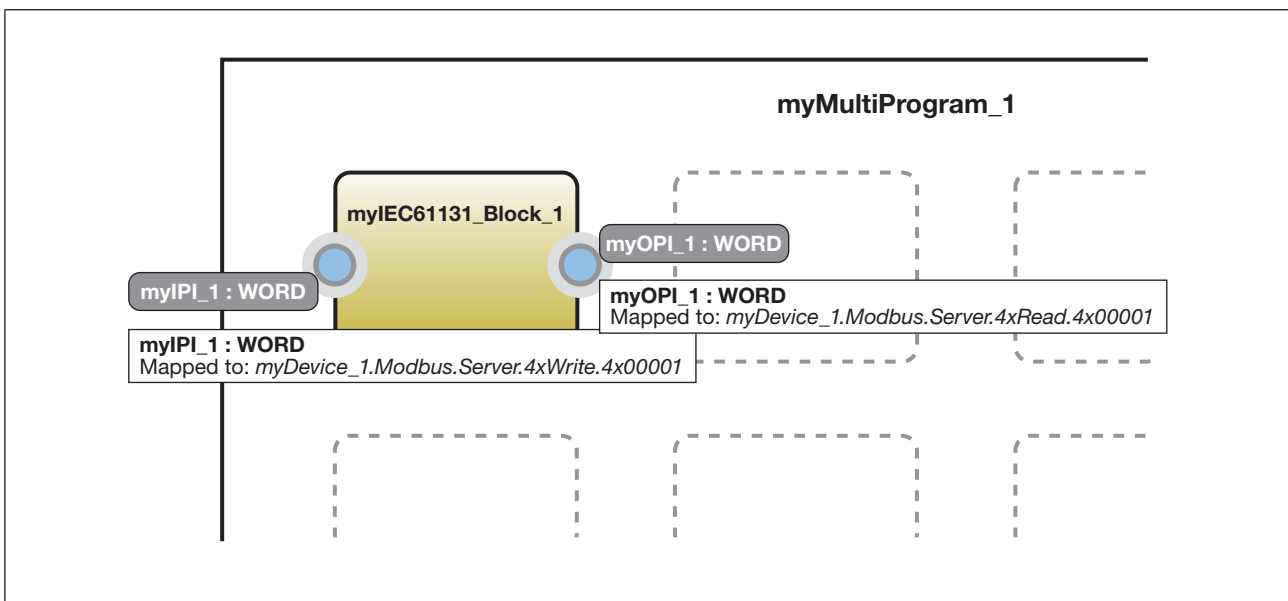


Data mirroring in IEC 61131 programming (example)

Procedure for Multi programming

Option 1:

- ▶ Create an IEC 61131 basic block in the Multi program
- ▶ Declare the required PI variables in the IEC 61131 basic block and program "data mirroring"
This procedure is identical to the procedure for IEC-61131 programming (see above).
- ▶ I/O-map the I-PI variables to the write data areas (0xWrite, 4xWrite) of the Server allocation table
- ▶ I/O-map the O-PI variables to the read data areas (0xRead, 4xRead) of the Server allocation table
- ▶ Assign the block to a resource from a PSS 4000-exclusive/operable device
e.g. ST resource from myDevice_1

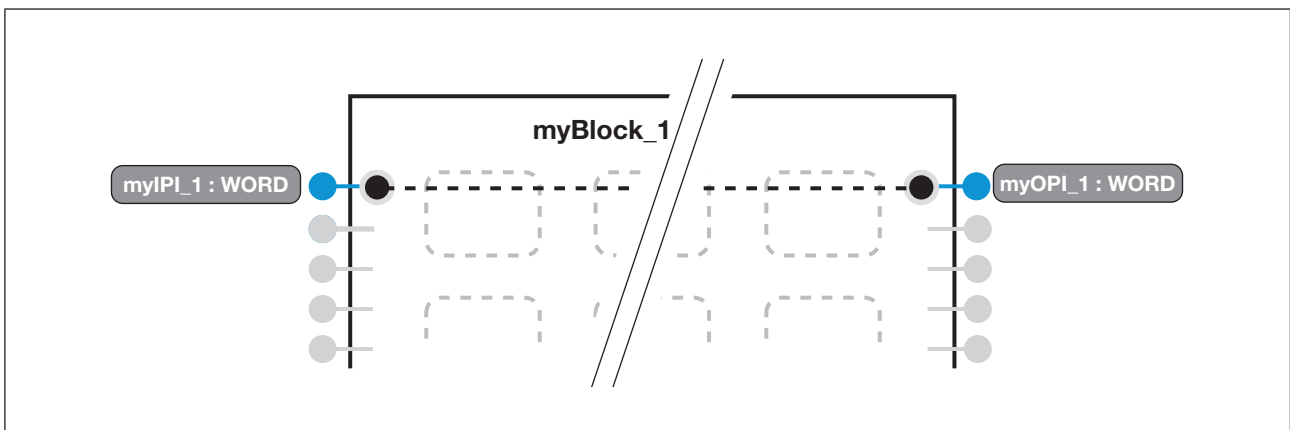


I/O-mapping in Multi programming – IEC 61131 basic block (example)

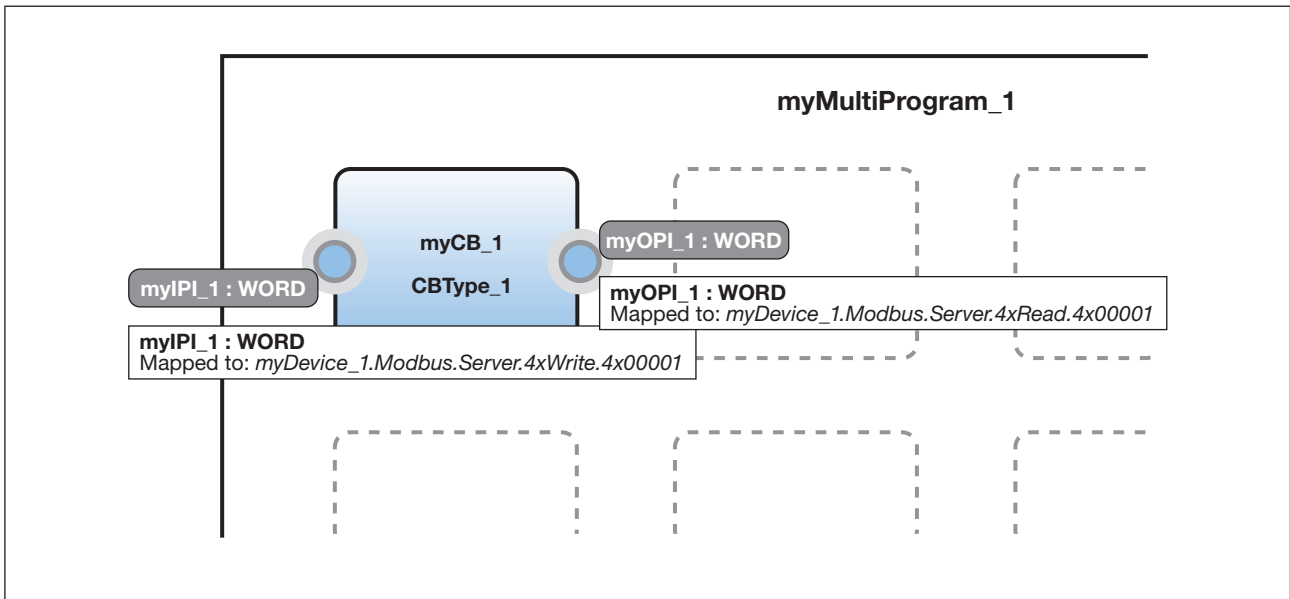
Option 2:

- ▶ Create a component block (CB) in the Multi program
- ▶ Configure the block interface to
 - Configure the required I-PI points, which represent the I-PI variables
Via the I-PI points, I/O-mapping is used to access the write data areas (0xWrite, 4xWrite) of the Server allocation table.
e.g. myIPI_1: WORD
 - Configure the corresponding O-PI points, which represent the O-PI variables
The O-PI points are needed in order to connect the I-PI points to the O-PI points within the block. This is a condition for "data mirroring" the I-data to the read data areas (0xRead, 4xRead) of the Server allocation table.
e.g. myOPI_1: WORD
- ▶ "Mirror data" by connecting the I-PI points to the block's O-PI points

Example:



- ▶ Configure I/O mapping
 - I/O-map the I-PI points to the write data areas (0xWrite, 4xWrite) of the Server allocation table
 - I/O-map the O-PI points to the read data areas (0xRead, 4xRead) of the Server allocation table



I/O-mapping in Multi programming – Component block (example)

7.3 Raw UDP

The User Datagram Protocol (UDP) is implemented in accordance with RFC768. "Raw UDP" stands for data exchange via UDP, without an upper layer protocol.

UDP is used for communication between various communication partners. The communication service on a communication partner is identified via a port number. UDP uses the Internet Protocol (IP) for addressing, i.e. a UDP datagram is encapsulated in an IP frame and is sent to the communication partner as a data packet using the IP address.

UDP is a "non-connection-oriented" or "connectionless" protocol. This means:

- ▶ A connection to the communication partner is not established before starting transmission
- ▶ There is no guarantee that
 - Sent data packets arrive
 - Sent data packets arrive only once
 - Data packets arrive in the same order in which they were sent

As a result, UDP can only be used if the application is insensitive to lost and unsorted datagrams or if the application itself provides appropriate control and correction mechanisms on the communication partners.

Although UDP is a connectionless protocol, for simplicity's sake, the description below refers to the communication relationship between communication partners as a "connection".

Data exchange via Raw UDP can be used on the automation system PSS 4000 for non-safety-related applications.

7.3.1 "IP connections" system section

In the automation system PSS 4000, external communication via Raw UDP is possible with all PSS 4000 devices that have the "IP connections" system section. In this case, the "IP connections" system section of a PSS 4000 device performs all the tasks required for communication. These include:

- ▶ Managing the Raw UDP connections (see Raw UDP connections, [Page 128](#))
- ▶ Data exchange (see Data exchange via Raw UDP, [Page 129](#))

7.3.2 Raw UDP connections

The "IP connections" system section of a PSS 4000 device manages the Raw UDP connections that are configured for the device. The connection is configured in the PAS4000's IP Connections Editor.

The following Raw UDP connections can be configured for a PSS 4000 device:

▶ **Send connections (role = transmitter)**

- Unicast (1:1 connection)
- Broadcast (1:"all" connection)
Loopback is configurable
- Multicast (1:n connection) in accordance with RFC 1112, IGMVP V1

▶ **Receive connections (role = receiver)**

A FIFO memory can be enabled for receive data

- Unicast (1:1 connection)
- Broadcast (1:"all" connection)
- Multicast (1:n connection) in accordance with RFC 1112, IGMVP V1

▶ **Send and receive connections (role = transmitter/receiver)**

A FIFO memory can be enabled for receive data

- Unicast (1:1 connection)
- Broadcast (1:"all" connection)
Loopback when sending is configurable
- Multicast (1:n connection) in accordance with RFC 1112, IGMVP V1

7.3.3 Data exchange via Raw UDP

Raw UDP is not connection-oriented. Consequently Raw UDP does not have any mechanisms for establishing and ending a connection.

With connections that are configured in PAS4000 as send connections (role = transmitter), the PSS 4000 device sends data to one or more communication partners with a receiver role for this connection.

With connections that are configured in PAS4000 as receive connections (role = receiver), the connection waits for the communication partner with a transmitter role for this connection to send its data.

Connections that are configured in PAS4000 as send and receive connections (role = transmitter/receiver) are used to both send and receive data. A communication partner that opens communication, i.e. sends the first telegram via this connection, must be configured for send and receive connections. This communication partner is called the initiator. The communication partner without initiator functionality waits to receive the first telegram via this connection.

From the perspective of the automation system PSS 4000, receive data is called external I-data, and send data is called external O-data (see Validity of process data, [Page 202](#)).

A send and receive connection with initiator functionality is configured in PAS4000 according to the following rule of thumb:

Role	Network settings	
Transmitter/receiver	Remote port number	Does not equal 0
	Remote IP address	Does not equal 0.0.0.0



Information

The initiator of a send and receive connection must meet the condition for both the remote port number and the remote IP address.

If just one of the conditions is **not** met, i.e. the remote port number is 0 and/or the remote IP address is 0.0.0.0, the send and receive connection will not have initiator functionality.

7.3.4 Data transfer limits

When transferring data via Raw UDP, the following limits apply for a PSS 4000 device:

- ▶ The max. number of Raw UDP connections depends on the device. Details can be found in the operating manual of the relevant PSS 4000 device.
Please also note the general upper limit of 32 IP connections for one PSS 4000 device.



CAUTION!

The following applies for the **max. number of IP connections**:

The "IP connections" system section on a PSS 4000 device can manage a maximum of 32 IP connections in total. This upper limit also applies if you are operating the "IP connections" system section of a PSS 4000 device with a variety of protocols or communication links.

SafetyNET p connections are **not** managed by the "IP connections" system section and so do not affect this upper limit.

- ▶ A data length of 1 ... max. 1024 Bytes (external I-data) can be configured for a receive connection.
- ▶ A data length of 1 ... max. 1024 Bytes (external O-data) can be configured for a send connection.
- ▶ A data length of 1 ... max. 1024 Bytes for each transmission direction can be configured for a send and receive connection.
- ▶ The maximum configurable data length is supported for each telegram; i.e. data from send connections will not be spread over multiple telegrams.
- ▶ The number of data bytes transmitted in a telegram must exactly equal the number that was configured for the connection:
 - If a telegram received by a PSS 4000 device contains more data bytes than were configured for the connection, the surplus data bytes will be rejected. A diagnostic message will not be displayed in the diagnostic list, as this is the defined behaviour for the PSS 4000.
 - If a telegram received by a PSS 4000 device contains fewer data bytes than were configured for the connection, a diagnostic message will be displayed in the diagnostic list.
- ▶ A FIFO memory can be enabled for receive data. In this case, all incoming data packets are collected in the FIFO memory and are processed in the order in which they arrived. This procedure prevents the situation in which an "old" data packet is rejected when a data packet arrives, even if it has not yet been processed.
The size of the FIFO memory depends on the device. Details can be found in the operating manual of the relevant PSS 4000 device. If a memory overflow is imminent, the TCP protocol ensures that the communication partner sends less data, or no data at all. This behaviour is maintained until sufficient FIFO memory is once again available. In the event of an imminent or actual memory overflow, a diagnostic message will be entered in the diagnostic list.

7.3.5 Raw UDP I/O data

The "IP connections" system section enables a PSS 4000 device to exchange I/O data with third-party devices via Raw UDP. For this to work it is necessary for the I/O data in PAS4000's I/O Mapping Editor to be mapped to the corresponding Raw UDP data areas.

I/O mapping of Raw UDP data areas

I/O mapping is used to define which data source supplies data to which data sink.

Possible I/O mapping for Raw UDP data areas:

		Raw UDP	
		Receive data (= external I-data)	Send data (= external O-data)
Module bus	I-datum (e.g. HW input)	---	◆
	O-datum (e.g. HW output)	◆	---
PI variables	I-PI variable	◆	---
	O-PI variable	---	◆

◆ I/O mapping possible

--- I/O mapping impossible

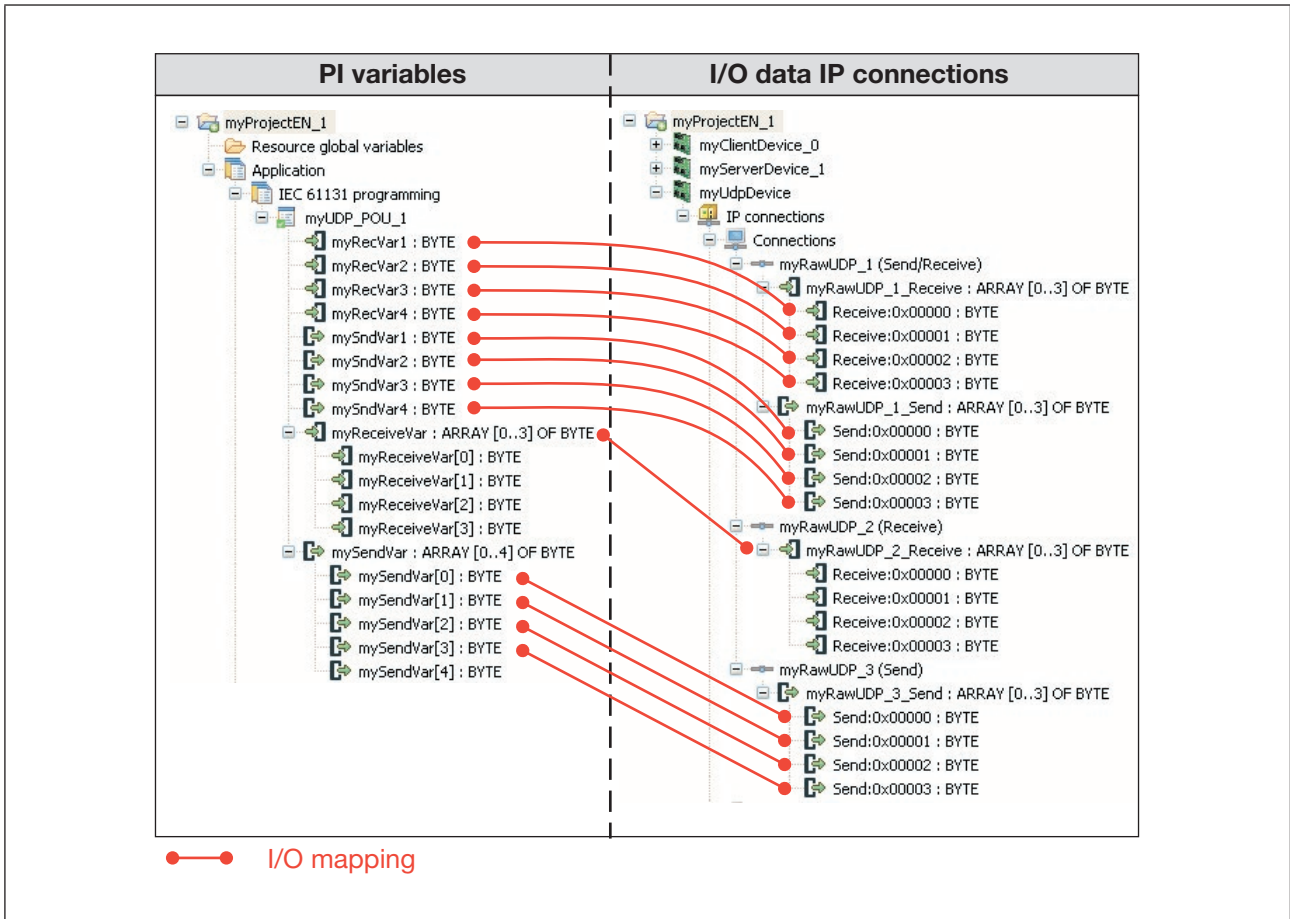
Examples:

▶ **I/O mapping of I/O data**

The receive data of the electronic module PSSu K S RS232 is to be transmitted to the communication partner and data received by the communication partner is to be output on the PSSu K S RS232.

The send and receive connection "myRawUDP_1" is configured for this purpose:

Possible I/O mapping (example):

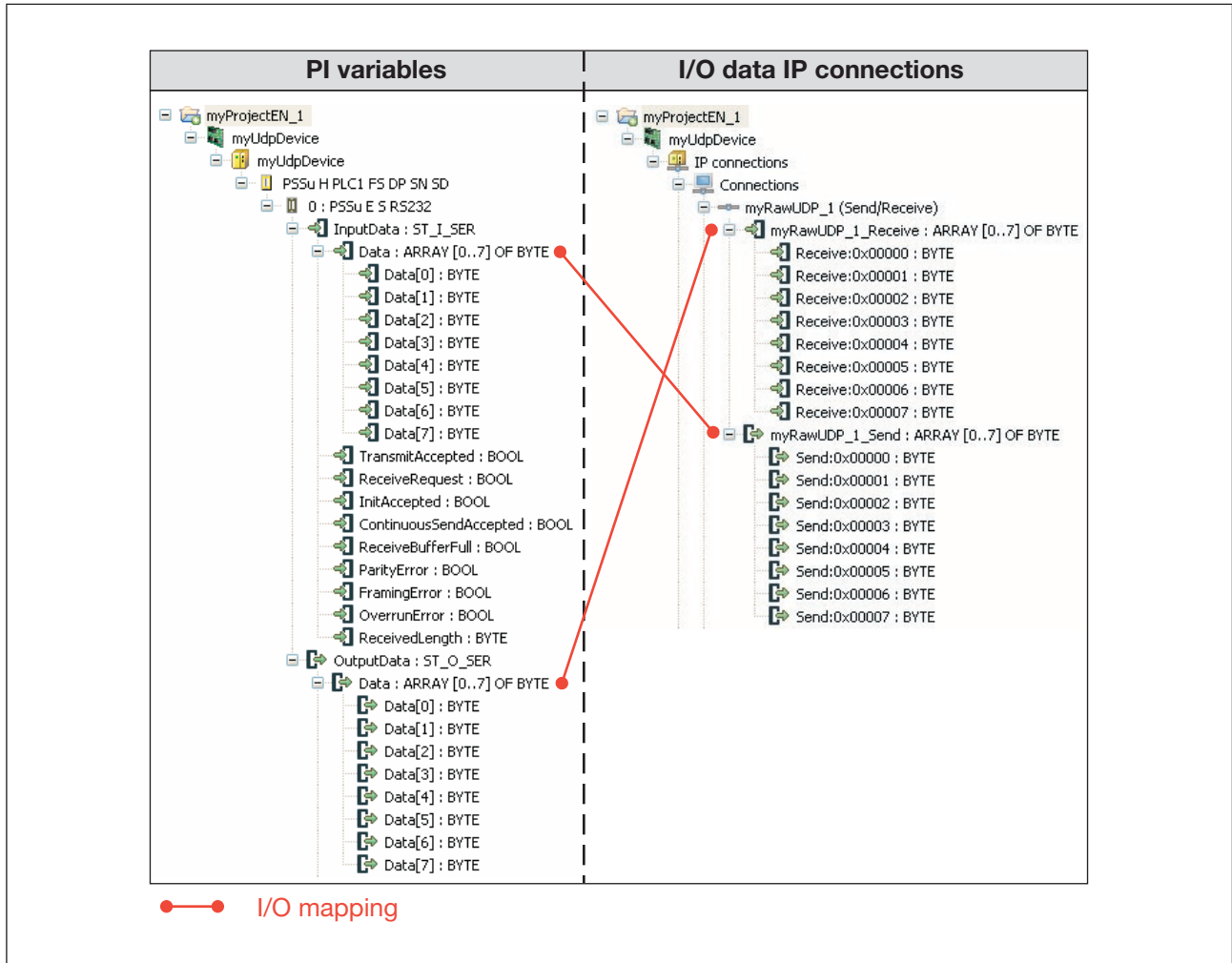


▶ **I/O mapping of PI variables**

The contents of a block's O-PI variables are to be transmitted to the communication partner and I-PI variables are to be assigned to the receive data. 3 connections are configured for this purpose:

- myRawUDP_1 (send and receive connection)
- myRawUDP_2 (send connection)
- myRawUDP_3 (receive connection)

Possible I/O mapping (example):



7.3.6 Address format of Raw UDP addresses

The addresses for the I/O data of a PSS 4000 device are displayed at several points, e.g. in PAS4000 and in the diagnostics.

7.3.6.1 Address format of send data

The addresses of Raw UDP send data are structured as follows:

<Device name>.UDP.<Connection name>.Send.Byte[<Element number>]

Example:

myDevice_1.UDP.myTxConnection_1.Send.Byte[0]

...

myDevice_1.UDP.myTxConnection_1.Send.Byte[1023]

7.3.6.2 Address format of receive data

The addresses of Raw UDP receive data are structured as follows:

<Device name>.UDP.<Connection name>.Receive.Byte.[<Element number>]

Example:

myDevice_1.UDP.myRxConnection_1.Receive.Byte[0]

...

myDevice_1.UDP.myRxConnection_1.Receive.Byte[1023]

7.3.7 Data consistency on Raw UDP

Where receive data is sent by third-party devices, these devices are responsible for the data consistency of this data. From the PSS 4000's perspective, no statement can be made regarding the data consistency of those devices.

Data consistency on receive data of a Raw UDP connection

If a PSS 4000-exclusive/operable device is the recipient of data from a Raw UDP connection, from the perspective of the PSS 4000 device the receive data is external I-data. External I-data from a Raw UDP connection is treated as consistent data in PSS 4000 (see Data consistency, [Page 214](#)).

The external I-data from a Raw UDP connection forms a consistency range of the consistency range type "ST-I data of an external communication connection". The exact description is consistency range "ST-I data of a Raw UDP connection" (see Consistency ranges, [Page 215](#)).

Data consistency on send data of a Raw UDP connection

If a PSS 4000-exclusive/operable device in a Raw UDP connection is the transmitter of data, from the perspective of the PSS 4000 device the send data is external O-data. The external O-data of a Raw UDP connection is consistent if it comes from the same consistency range on a data source (see Data consistency, [Page 214](#)).

7.4 PROFIBUS-DP slave

PROFIBUS is an open fieldbus standard whose communication is defined in the international standards IEC 61158 and IEC 61784. Further provisions have been defined in specifications published by the PROFIBUS User Group. These specifications are available from PROFIBUS International (see www.profibus.com).

For non-safety related applications, a PROFIBUS-DP enabled PSS 4000 device can be incorporated into PROFIBUS-DP as a Slave.

A PROFIBUS-DP enabled PSS 4000 device can be used with a Class 1 DP Master that supports the communication protocol PROFIBUS-DP with performance level DP-V0, DP-V1 or DP-V2.

As a PROFIBUS-DP Slave, a PROFIBUS-DP enabled PSS 4000 device has the following features:

- ▶ Uses the communication protocol PROFIBUS-DP with performance level DP-V0
- ▶ Devices addressed via DIP switches (e.g. on the head module of a PSSu system) or via the PROFIBUS-DP Master when using device address 126
- ▶ Selectable transmission rates (9.6 kBits/s ... 12 MBits/s)
- ▶ Supports the mode for automatic detection of the transmission rate (auto baud mode)
- ▶ Supports sync mode
- ▶ Supports freeze mode
- ▶ Supports the PROFIBUS-DP diagnostic telegram

7.4.1 "PROFIBUS-DP Slave" system section

In the automation system PSS 4000, external communication via PROFIBUS-DP is possible with all PSS 4000 devices that have the "PROFIBUS-DP Slave" system section. In this case, the "PROFIBUS-DP Slave" system section performs all the tasks required for communication. For example these include managing the send and receive data areas and data exchange (see Data exchange via PROFIBUS-DP, [Page 138](#)).

The connection is configured in the PAS4000's PROFIBUS-DP Slave Editor.

7.4.2 Data exchange via PROFIBUS-DP

PSS 4000 devices with the "PROFIBUS-DP Slave" system section operate as PROFIBUS-DP Slaves when exchanging data via PROFIBUS-DP. It is always the PROFIBUS-DP Master that initiates data transfer. From the perspective of the automation system PSS 4000, the PROFIBUS-DP Master is outside the system.

7.4.2.1 Send data and receive data

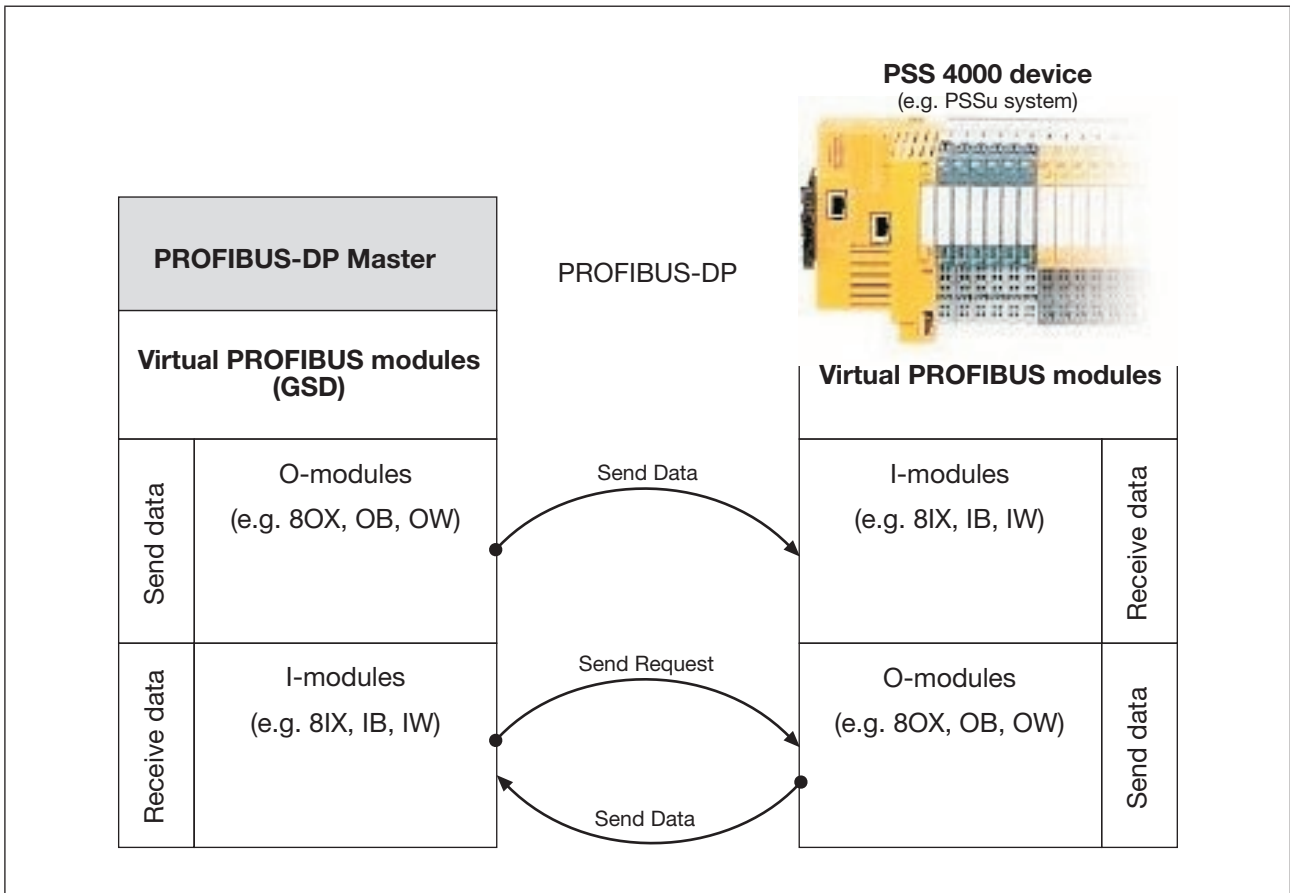
The PROFIBUS-DP Master sends data from its send data area to the receive data area of the "PROFIBUS-DP Slave" system section of a PSS 4000 device and requests data from the "PROFIBUS-DP Slave" system section of a PSS 4000 device. The "PROFIBUS-DP Slave" system section of a PSS 4000 device sends the requested data from its send data area to the receive data area of the PROFIBUS-DP Master. In the automation system PSS 4000, the data areas are called "virtual PROFIBUS modules". Virtual PROFIBUS modules are available with a variety of data types.

For data exchange, corresponding virtual PROFIBUS modules must be configured for both the PROFIBUS-DP Master and "PROFIBUS-DP Slave" system section of a PSS 4000 device. The virtual PROFIBUS modules for send data are called O-modules. They have an "O" for "Output" in the module name (e.g. 8OX, OB, OW). The virtual PROFIBUS modules for receive data are called I-modules. They have an "I" for "Input" in the module name (e.g. 8IX, IB, IW).

The PROFIBUS DP Master obtains the required virtual PROFIBUS modules from the GSD file. This is supplied with PAS4000 and is also available in the download area of the Pilz homepage (www.pilz.de -> Downloads). "GSD" can be entered as the search term, for example. The software download is only available for registered users. Unregistered users can register free of charge.

The virtual PROFIBUS modules required for the "PROFIBUS-DP Slave" system section of a PSS 4000 device are configured during the PROFIBUS-DP Slave configuration in PAS4000. After configuration the virtual PROFIBUS modules are located on virtual slots (virtual slots 0 ... 79).

From the perspective of the automation system PSS 4000, the receive data is called external I-data, and the send data is called external O-data (see Validity of process data, [Page 202](#)).



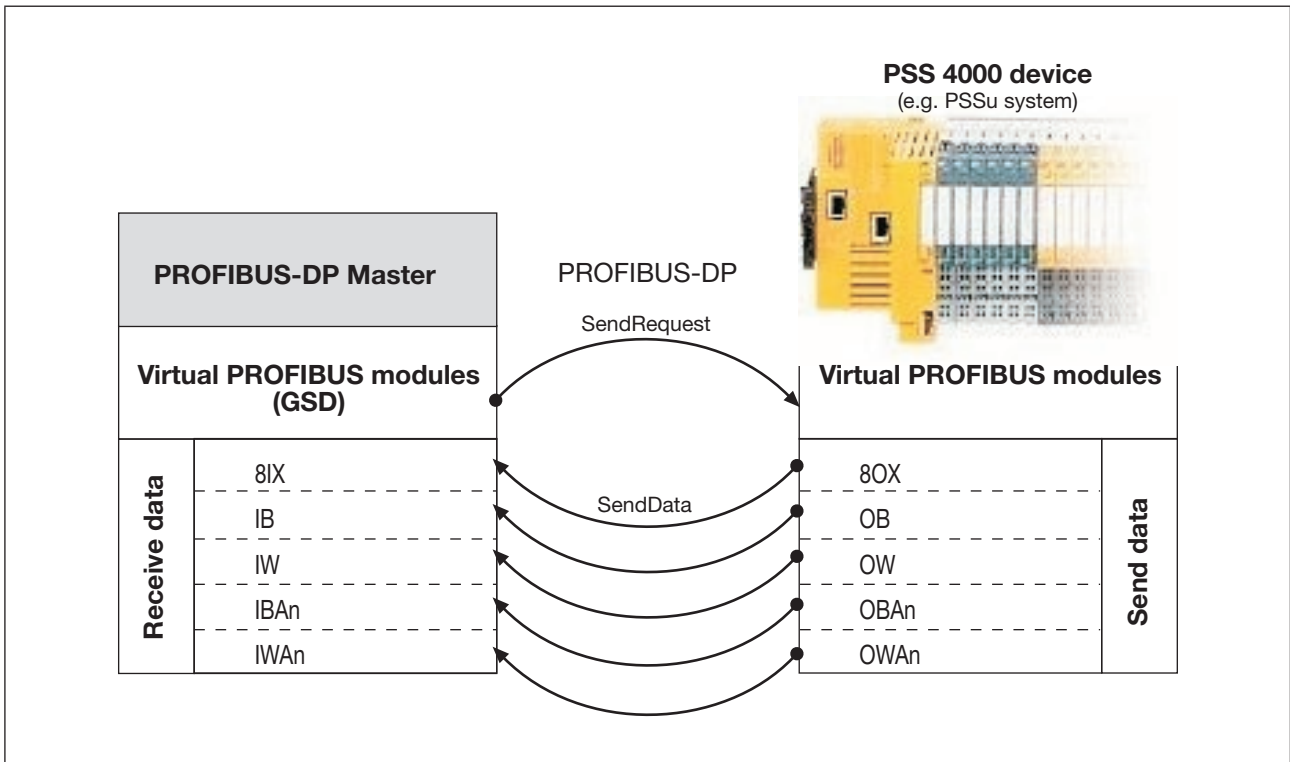
Data exchange via virtual PROFIBUS modules (principle)

7.4.2.2 Virtual PROFIBUS modules for send data

The following virtual PROFIBUS modules (O-modules) can be configured in PAS4000 for the send data of a "PROFIBUS-DP Slave" system section:

Virtual PROFIBUS module	Data type	Application
8OX	ARRAY [0..7] OF BOOL	Virtual PROFIBUS bit module To send an array with 8 elements of data type BOOL
OB	BYTE	Virtual PROFIBUS byte module To send data of data type BYTE
OW	WORD	Virtual PROFIBUS byte module To send data of data type WORD
OD	DWORD	Virtual PROFIBUS byte module To send data of data type DWORD
OBAn	ARRAY [0..(n-1)] OF BYTE	Virtual PROFIBUS byte module To send an array with n elements of data type BYTE n = 2, 4, 8, 16, 32 or 64
OWAn	ARRAY [0..(n-1)] OF WORD	Virtual PROFIBUS byte module To send an array with n elements of data type WORD n = 2, 4, 8, 16, 32 or 64

During data exchange, the PROFIBUS-DP Master requests data from O-modules on the automation system PSS 4000 and receives this data via its I-modules. Consequently, the O-modules on the automation system PSS 4000 must have corresponding I-modules in the PROFIBUS-DP Master. Also, the data type of an I-module in the PROFIBUS-DP Master must correspond to the data type of an O-module in the automation system PSS 4000.



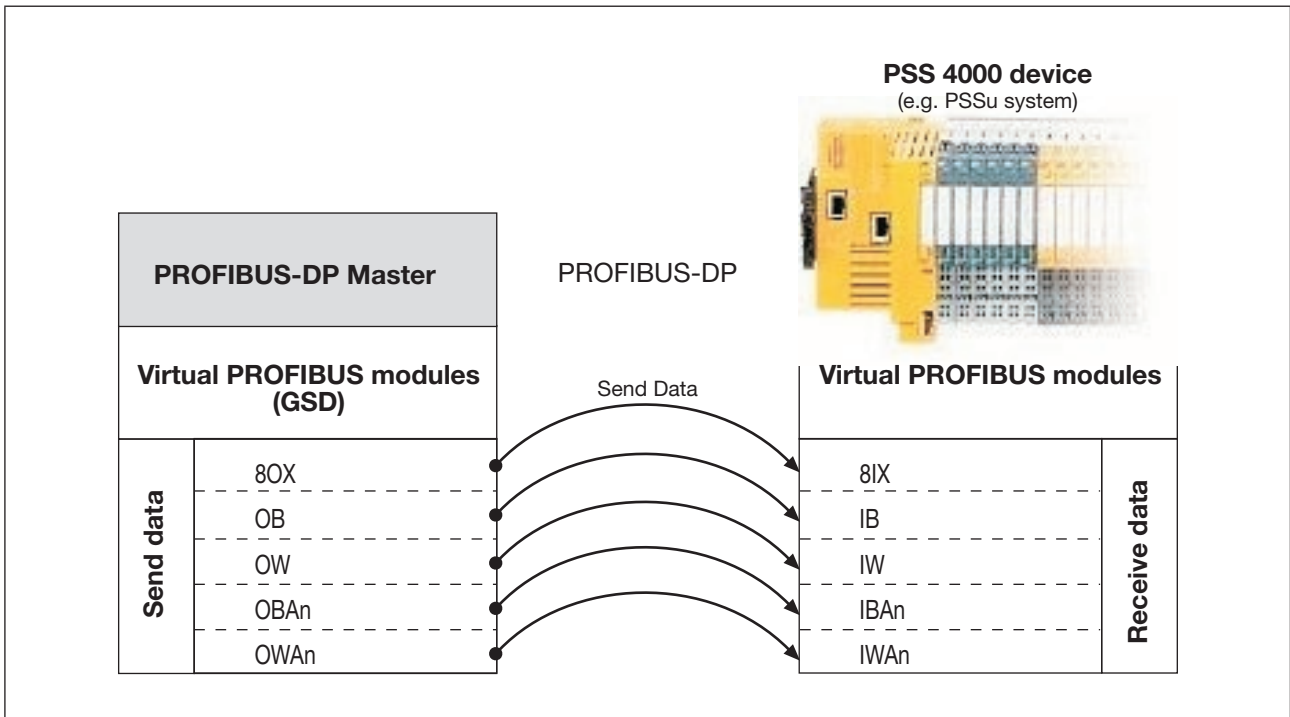
Virtual PROFIBUS modules and their corresponding data types (send data of a PSS 4000 device)

7.4.2.3 Virtual PROFIBUS modules for receive data

The following virtual PROFIBUS modules (I-modules) can be configured in PAS4000 for the receive data of a "PROFIBUS-DP Slave" system section:

Virtual PROFIBUS module	Data type	Application
8IX	ARRAY [0..7] OF BOOL	Virtual PROFIBUS bit module To receive an array with 8 elements of data type BOOL
IB	BYTE	Virtual PROFIBUS byte module To receive data of data type BYTE
IW	WORD	Virtual PROFIBUS byte module To receive data of data type WORD
ID	DWORD	Virtual PROFIBUS byte module To receive data of data type DWORD
IBAn	ARRAY [0..(n-1)] OF BYTE	Virtual PROFIBUS byte module To receive an array with n elements of data type BYTE n = 2, 4, 8, 16, 32 or 64
IWAn	ARRAY [0..(n-1)] OF WORD	Virtual PROFIBUS byte module To receive an array with n elements of data type WORD n = 2, 4, 8, 16, 32 or 64

During data exchange, the PROFIBUS-DP Master transmits its send data to the I-modules in the automation system PSS 4000 via O-modules. Consequently, the O-modules on the PROFIBUS-DP Master must have corresponding I-modules in the automation system PSS 4000. Also, the data type of an I-module must correspond to the data type of the O-module in the PROFIBUS-DP Master.



Virtual PROFIBUS modules and their corresponding data types (receive data of a PSS 4000 device)

7.4.3 Data transfer limits

The following limits apply for data transfer via virtual PROFIBUS modules:

- ▶ A maximum of 80 virtual PROFIBUS modules with virtual slots 0 ... 79 can be configured for the "PROFIBUS-DP Slave" system section of a PSS 4000 device.
- ▶ A maximum of 244 Bytes of receive data (= external I-data) can be configured for the "PROFIBUS-DP Slave" system section of a PSS 4000 device.
- ▶ A maximum of 244 Bytes of send data (= external O-data) can be configured for the "PROFIBUS-DP Slave" system section of a PSS 4000 device.

7.4.4 I/O data of PROFIBUS-DP Slave

The "PROFIBUS-DP Slave" system section enables a PSS 4000 device to exchange I/O data with the PROFIBUS-DP Master via PROFIBUS-DP. For this to work it is necessary for the I/O data in PAS4000's I/O Mapping Editor to be mapped to the corresponding virtual PROFIBUS modules.

I/O-mapping of virtual PROFIBUS modules

I/O mapping is used to define which data source supplies data to which data sink.

Possible I/O-mapping for virtual PROFIBUS modules:

		Virtual PROFIBUS modules	
		Receive data (= external I-data of I modules as e.g B. 8IX, IB, IW, ID, IBAn, IWAn)	Send data (= external O-data of O-modules as e.g. 8OX, OB, OW, OD, OBAn, OWAn)
Module bus	I-datum (e.g. HW input)	---	◆
	O-datum (e.g. HW output)	◆	---
PI variables	I-PI variable	◆	---
	O-PI variable	---	◆

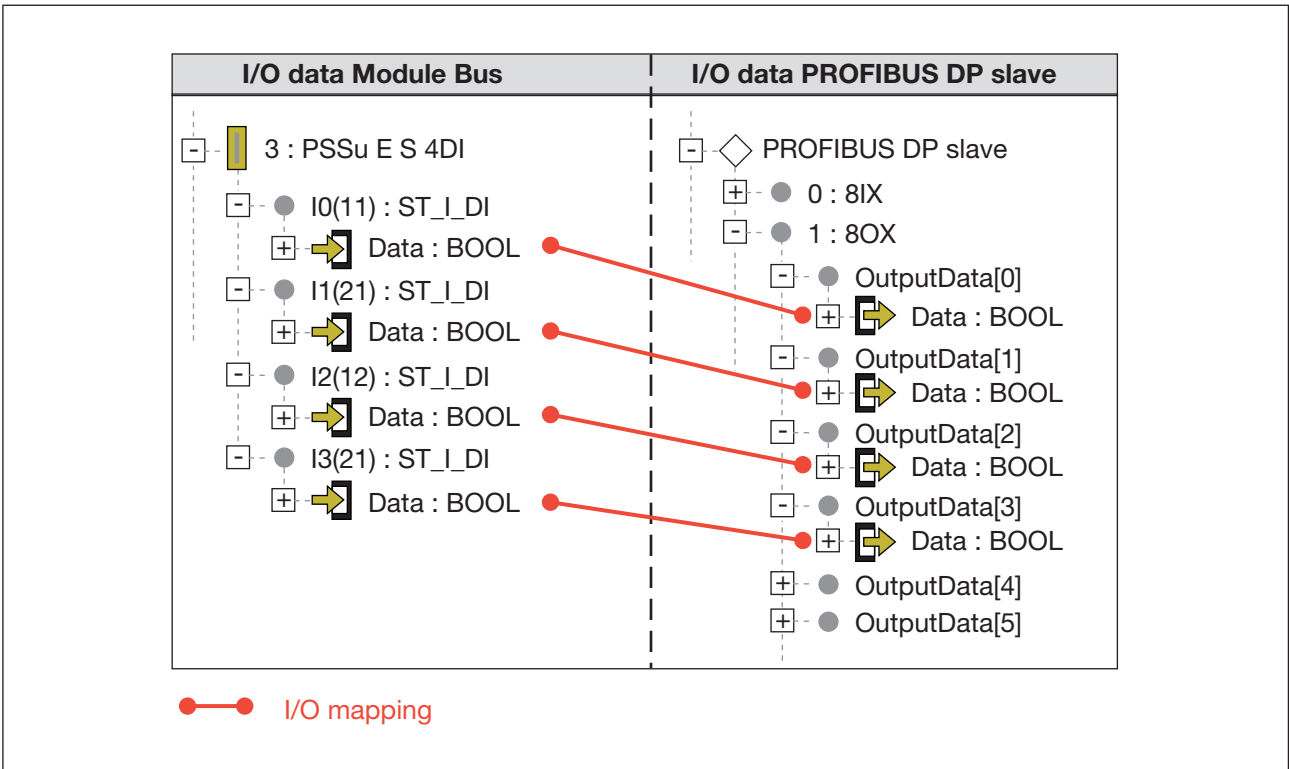
◆ I/O mapping possible

--- I/O mapping impossible

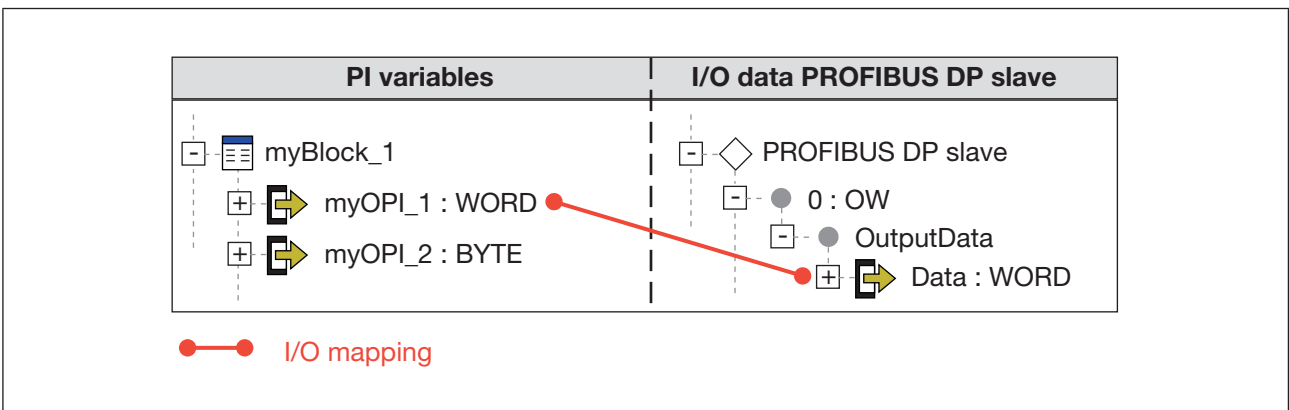
Examples:

I/O-mapping for virtual PROFIBUS modules with send data

- ▶ The input signals of the hardware inputs of a PSSu E S 4DI are to be transferred to the PROFIBUS DP master.
 - Direction of data flow: Hardware input-> PROFIBUS DP
 - Virtual PROFIBUS module used: 8OX
 - Possible I/O mapping (example):

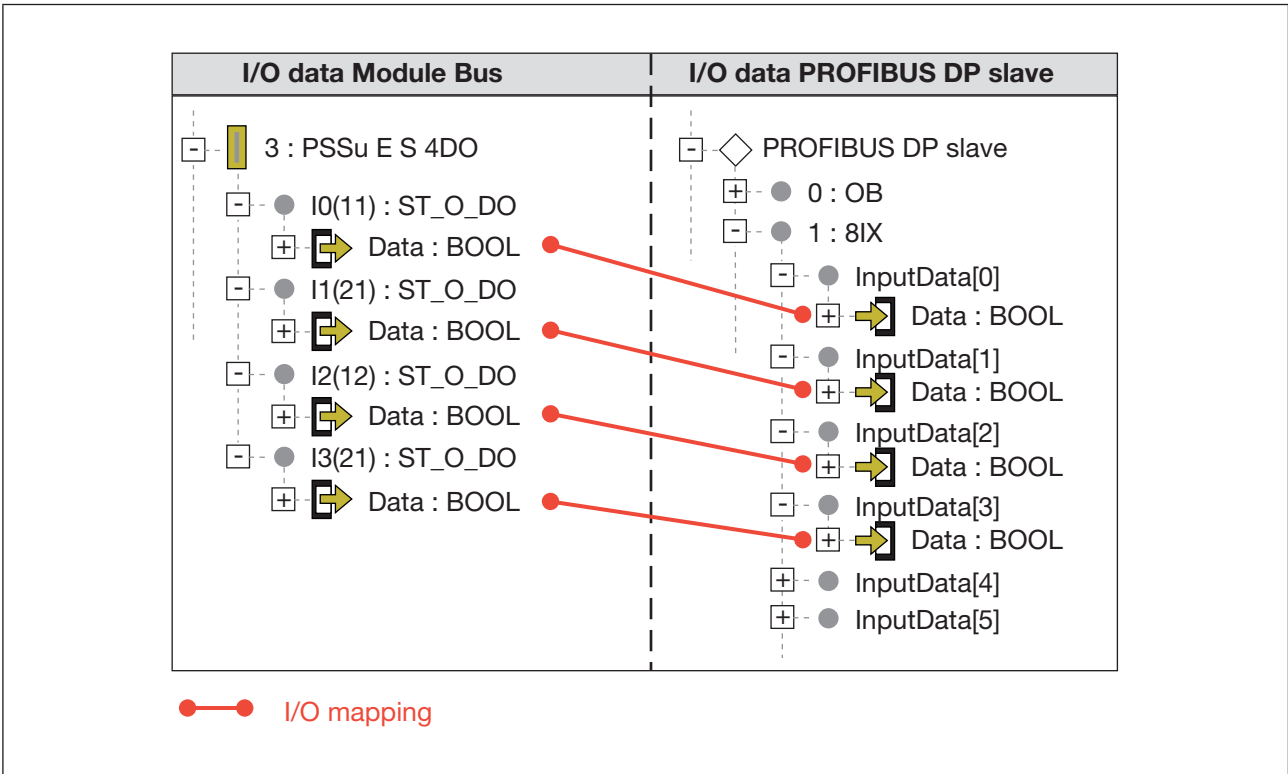


- ▶ The content of an O-PI variable of a block is to be transferred to the PROFIBUS DP master.
 - Direction of data flow: O-PI variable -> PROFIBUS DP
 - Virtual PROFIBUS module used: OW
 - Possible I/O mapping (example):

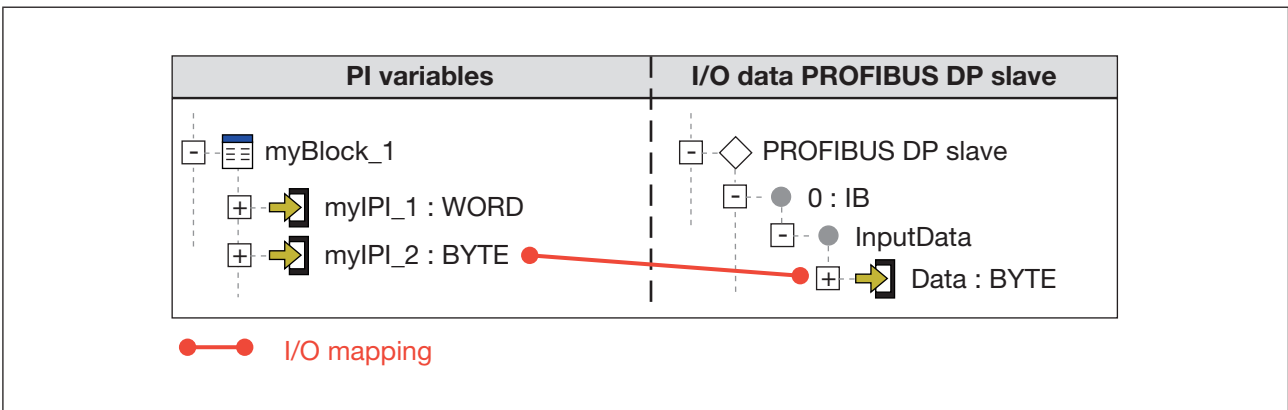


I/O-mapping for virtual PROFIBUS modules with receive data

- ▶ PROFIBUS-DP master is to control the hardware outputs of a PSSu E S 4DO 0.5.
 - Direction of data flow: PROFIBUS-DP -> Hardware outputs
 - Virtual PROFIBUS module used: 8IX
 - Possible I/O mapping (example):



- ▶ A date is to be transferred from the PROFIBUS-DP master to the PSSu system. Once the data has been received it is to be made available for further processing within a block.
 - Direction of data flow: PROFIBUS-DP -> I-PI variable
 - Virtual PROFIBUS module used: IB
 - Possible I/O mapping (example):



7.4.5 Address display of PROFIBUS-DP addresses

The addresses for the I/O data of a PSS 4000 device are displayed at several points, e.g. in PAS4000 and in the diagnostics.

7.4.5.1 Address format of send data

The addresses of PROFIBUS-DP send data (O-data of virtual PROFIBUS modules) are structured as follows:

<Device name>.ProfibusDP.<Virtual slot>.OutputData[<Element number>]

Examples:

- ▶ Address format for an 8OX:
myDevice_1.ProfibusDP.2.OutputData[0]
myDevice_1.ProfibusDP.2.OutputData[1]
myDevice_1.ProfibusDP.2.OutputData[2]
myDevice_1.ProfibusDP.2.OutputData[3]
myDevice_1.ProfibusDP.2.OutputData[4]
myDevice_1.ProfibusDP.2.OutputData[5]
myDevice_1.ProfibusDP.2.OutputData[6]
myDevice_1.ProfibusDP.2.OutputData[7]
- ▶ Address format for an OB:
myDevice_1.ProfibusDP.3.OutputData
- ▶ Address format for an OW:
myDevice_1.ProfibusDP.4.OutputData
- ▶ Address format for an OBA2:
myDevice_1.ProfibusDP.5.OutputData[0]
myDevice_1.ProfibusDP.5.OutputData[1]
- ▶ Address format for an OWA4:
myDevice_1.ProfibusDP.6.OutputData[0]
myDevice_1.ProfibusDP.6.OutputData[1]
myDevice_1.ProfibusDP.6.OutputData[2]
myDevice_1.ProfibusDP.6.OutputData[3]

7.4.5.2 Address format of receive data

The addresses of PROFIBUS-DP receive data (I-data of virtual PROFIBUS modules) are structured as follows:

<Device name>.ProfibusDP.<Virtual slot>.InputData[<Element number>]

Examples:

- ▶ Address format for an 8IX:
myDevice_1.ProfibusDP.7.InputData[0]
myDevice_1.ProfibusDP.7.InputData[1]
myDevice_1.ProfibusDP.7.InputData[2]
myDevice_1.ProfibusDP.7.InputData[3]
myDevice_1.ProfibusDP.7.InputData[4]
myDevice_1.ProfibusDP.7.InputData[5]
myDevice_1.ProfibusDP.7.InputData[6]
myDevice_1.ProfibusDP.7.InputData[7]
- ▶ Address format for an IB:
myDevice_1.ProfibusDP.8.InputData
- ▶ Address format for an IW:
myDevice_1.ProfibusDP.9.InputData
- ▶ Address format for an IBA4:
myDevice_1.ProfibusDP.10.InputData[0]
myDevice_1.ProfibusDP.10.InputData[1]
myDevice_1.ProfibusDP.10.InputData[2]
myDevice_1.ProfibusDP.10.InputData[3]
- ▶ Address format for an IWA2:
myDevice_1.ProfibusDP.11.InputData[0]
myDevice_1.ProfibusDP.11.InputData[1]

7.4.6 PROFIBUS diagnostics

PROFIBUS-DP diagnostics can be performed via the PSS 4000 system diagnostics (see Diagnostic basics, [Page 470](#)).

However, the standard defines proprietary diagnostic options for PROFIBUS-DP. The current diagnostic information can be polled by a PROFIBUS-DP Master on the PROFIBUS-DP Slave via a diagnostic telegram.

The standard divides the diagnostic information in the diagnostic telegram into the following areas:

- ▶ [System-internal diagnostics](#)
- ▶ [External diagnostics](#)
 - Device-based diagnostics
 - ID-based diagnostics
 - Channel-based diagnostics

The system-internal diagnostic data is located in the first 6 Bytes of the diagnostic telegram and its meaning is firmly specified in the standard. These obligatory 6 Bytes may be followed by one block or several blocks containing external diagnostic data.

On PSS 4000 devices with a "PROFIBUS-DP Slave" system section, a PROFIBUS DP diagnostic telegram consists of a min. 6 Bytes of diagnostic data (obligatory diagnostic data) and a max. of 9 Bytes of diagnostic data (6 Bytes of obligatory diagnostic data + 3 Bytes of external diagnostic data).

7.4.6.1 System-internal diagnostics

Structure and content of the system-internal diagnostics data (Bytes 1 ...6):

Byte No.	Bit No.	Meaning
1	0	Station_Non_Existent
	1	Station_Not_Ready
	2	Cfg_Fault
	3	Ext_Diag 0: no external diagnostics data present 1: external diagnostics data present
	4	Not_Supported
	5	Invalid_Slave_Response
	6	Prm_Fault
	7	Master_lock
2	0	Prm_req
	1	Stat_Diag
	2	Always "1"
	3	WD_on
	4	Freeze_Mode
	5	Sync_Mode
	6	Reserved
	7	Deactivated
3	6 ... 0	Reserved
	7	Ext_Diag_Overflow
4	7 ... 0	Address of the PROFIBUS-DP master
5	7 ... 0	Ident number of the PROFIBUS-DP slave: High-Byte PSSu systems: 0C
6	7 ... 0	Ident number of PROFIBUS-DP slave: Low-Byte PSSu systems: B0



Information

The exact meaning of the Bytes 1 ... 6 can be found in the IEC 61158-6-3 (5.3.1) and IEC 61158-6-5 (6.2.3.3.1).

7.4.6.2 External diagnostics data

In accordance with the standard, a diagnostic block starts with a Header Byte. This Header Byte determines the type and length of the diagnostic block. The standard defines three formats:

- ▶ Device-based diagnostics
- ▶ ID-based diagnostics
- ▶ Channel-based diagnostics

PSS 4000 supports device-based diagnostics. 3 Bytes including the Header Byte are required for device-based diagnostic data on a PSS 4000 device with "PROFIBUS-DP Slave" system section.

Structure and content of device-based diagnostic data of PSS 4000 devices with "PROFIBUS-DP Slave" system section

Byte No.	Bit No.	Value	Meaning
n	0 ... 7	03h	Header Byte
			Bits 7 and 6: PSS 4000 device with "PROFIBUS-DP Slave" system section: Code for device-based diagnostics
			Bits 5 ... 0: Number of bytes in the device-based diagnostics (including Header Byte) PSS 4000 device with "PROFIBUS-DP Slave" system section: 3 Bytes

Byte No.	Bit No.	Value	Meaning
n+1	0	0/1	Error message for the PSS 4000 device: 0: No message 1: A message of "Error" severity is present for at least one system section of the PSS 4000 device or A major FS error is present for at least one FS system section of the PSS 4000 device (on a PSSu system, this corresponds to the status of the "DIAG" LED on the head module: "lights up red" or "flashes red")
	1	0/1	Error in the FS/ST module bus system sections of the PSS 4000 device: The following applies for a PSSu system: 0: No message 1: Operating status "Safe condition of all FS outputs on the PSSu system" or At least one module cannot be accessed (e.g. a module has been removed during operation, set/actual hardware registry does not match) or Operating status "FS module bus in a STOP condition with error: Major FS error" (on a PSSu system, this corresponds to the status of the "MBUS" LED on the head module: "lights up red" or "flashes red")
	2	0/1	Error in the "FS-SafetyNET p RTFN" system section of the PSS 4000 device: 0: No message 1: Operating status "FS SafetyNET p RTFN in STOP condition with error: Major FS error" or Operating status "FS SafetyNET p RTFN in STOP condition with error: Major FS+ST error" (on PSSu systems, this corresponds to the status of the "FS SNp" LED on the head module: "lights up red" or "flashes red")
	3	0/1	Error in the "FS-SafetyNET p RTFN" system section of the PSS 4000 device: 0: No message 1: Operating status "ST SafetyNETp RTFN in STOP condition with error": Major FS+ST error" (on a PSSu system, this corresponds to the status of the "ST SNp" LED on the head module: "lights up red" or "flashes red")
	4	0	Reserved
	5	0	Reserved
	6	0	Reserved
	7	0	Reserved

Byte No.	Bit No.	Value	Meaning
n + 2	0	0/1	Warning message for the PSS 4000 device 0: No message 1: A message of "Warning" severity is present for the PSS 4000 device (on PSSu systems, this corresponds to the status of the "DIAG" LED on the head module: lights up "orange")
	1	0/1	Forcing on the FS resource 0: No message 1: Forcing on the FS resource is active (on PSSu systems, corresponds to the status of the "FS FORCE" LED on the head module: "lights up yellow")
	2	0/1	Forcing on the ST resource 0: No message 1: Forcing on the ST resource is active (on PSSu systems, corresponds to the status of the "ST FORCE" LED on the head module: "lights up yellow")
	3 ... 7	0	Reserved

7.4.7 Data consistency on PROFIBUS-DP

The data consistency of the data exchanged between the PROFIBUS-DP Master and PSS 4000 depends on the configuration of the PROFIBUS-DP Master.

PROFIBUS-DP supports three types of data consistency:

- ▶ Byte consistency
Bytes are transmitted as whole units.
- ▶ Word consistency
The two bytes in a word are always transmitted together.
- ▶ Buffer consistency
The whole data buffer is transmitted as one unit without a break.
From the perspective of the PSS 4000, the data buffer corresponds to the respective virtual PROFIBUS module for send data or receive data.

Data consistency of the receive data

From the perspective of the PSS 4000, the receive data of a virtual PROFIBUS module is external I-data.

As soon as external I-data is available in a virtual PROFIBUS module, it is treated as consistent data (see Data consistency, [Page 214](#)). The external I-data from a PROFIBUS module forms a consistency range of the consistency range type "ST-I data of an external communication connection". The exact description is consistency range "ST-I data of a PROFIBUS-DP connection" (see Consistency ranges, [Page 215](#)).

Data consistency of the send data

From the perspective of the PSS 4000, the send data of a virtual PROFIBUS module is external O-data.

The external O-data of a virtual PROFIBUS module is consistent if it comes from the same consistency range on a data source.

PSS 4000 always treats the external O data of a virtual PROFIBUS module as one unit. It is the task of the PROFIBUS-DP Master to guarantee the necessary data consistency.

7.5 PROFIBUS-DP Slave with PROFIsafe

PROFIsafe functionality can be added to the non-safety-related fieldbus system PROFIBUS-DP for safety-related communication. In this case, the fieldbus system is superimposed with the PROFIsafe protocol in accordance with the black channel principle. Communication via PROFIsafe is defined in the international standard IEC 61784-3-3.

▶ PROFIBUS-DP

A PSS 4000 device with the "PROFIBUS-DP Slave with PROFIsafe" system section can be incorporated into PROFIBUS-DP as a Slave for **non-safety-related** applications.

▶ PROFIsafe

A PSS 4000 device with the "PROFIBUS-DP Slave with PROFIsafe" system section can be incorporated into PROFIsafe as an F-device (F-slave) for **safety-related** applications.

A PSS 4000 device that operates as a PROFIBUS-DP Slave and/or F-device can be used with a Class 1 DP Master that supports the communication protocol PROFIBUS-DP with performance level DP-V0, DP-V1 or DP-V2.

Supported features for PROFIBUS-DP

- ▶ Uses the communication protocol PROFIBUS-DP with performance level DP-V0 for data exchange via PROFIBUS-DP
- ▶ Uses the communication protocol PROFIBUS-DP with performance level DP-V1 for data exchange via PROFIsafe
- ▶ Devices addressed via DIP switches (e.g. on the head module of a PSSu system) or via the PROFIBUS-DP Master when device address 126 is used on the DIP switch
- ▶ Selectable transmission rates (9.6 kBits/s ... 12 MBits/s)
- ▶ Supports the mode for automatic detection of the transmission rate (auto baud mode)
- ▶ Supports sync mode
- ▶ Supports freeze mode
- ▶ Supports the PROFIBUS-DP diagnostic telegram
- ▶ Data exchange via pre-defined virtual PROFIBUS modules (I-modules, O-modules)

Supported features for PROFIsafe

- ▶ Uses the communication protocol PROFIBUS-DP with performance level DP-V1 for data exchange via PROFIsafe
- ▶ PROFIsafe address is assigned via DIP switch (e.g. on the head module of a PSSu systems) or via PAS4000 when PROFIsafe address 0 is used on the DIP switch
- ▶ F-parameters are displayed and edited via the Master Tool (e.g. Siemens-Software HW Config)
- ▶ Data exchange via pre-defined virtual F-modules

7.5.1 "PROFIBUS-DP Slave with PROFI-safe" system section

In the automation system PSS 4000, external communication via PROFIBUS-DP and/or PROFI-safe is possible with all PSS 4000 devices that have the "PROFIBUS-DP Slave with PROFI-safe" system section. In this case, the "PROFIBUS-DP Slave with PROFI-safe" system section performs all the tasks required for communication. For example, these include:

- ▶ Management of the send and receive data areas
- ▶ Data exchange via PROFIBUS-DP
- ▶ Data exchange via PROFI-safe

7.5.2 PROFIBUS-DP in the "PROFIBUS-DP Slave with PROFIsafe" system section

PSS 4000 devices with the "PROFIBUS-DP Slave with PROFIsafe" system section send **non-safety-related data** via PROFIBUS-DP. When exchanging data via PROFIBUS-DP they operate as PROFIBUS-DP Slaves. It is always the PROFIBUS-DP Master that initiates data transfer. From the perspective of the automation system PSS 4000, the PROFIBUS-DP Master is outside the system; i.e. the receive data is called external I-data and the send data is called external O-data (see Validity of process data, [Page 202](#)).

Send data and receive data

The PROFIBUS-DP Master sends data from its send data area to the receive data area of the "PROFIBUS-DP Slave with PROFIsafe" system section of a PSS 4000 device and requests data from the "PROFIBUS-DP Slave with PROFIsafe" system section of a PSS 4000 device.

The "PROFIBUS-DP Slave with PROFIsafe" system section of a PSS 4000 device sends the requested data from its send data area to the receive data area of the PROFIBUS-DP Master.

In the automation system PSS 4000, these data areas are called "virtual PROFIBUS modules". The virtual PROFIBUS modules for receive data are called I-modules. They have an "I" for "Input" in the module name (e.g. 8IX, IB, IW). The virtual PROFIBUS modules for send data are called O-modules. They have an "O" for "Output" in the module name (e.g. 8OX, OB, OW).

Virtual PROFIBUS modules must be configured for both the PROFIBUS-DP Master and the "PROFIBUS-DP Slave with PROFIsafe" system section of a PSS 4000 device. The PROFIBUS-DP Master contains all the necessary information on the potential virtual modules from the GSD file (see GSD file for virtual PROFIBUS modules, [Page 165](#)).

The virtual PROFIBUS modules required for the "PROFIBUS-DP Slave with PROFIsafe" system section of a PSS 4000 device are configured during the hardware configuration in PAS4000's PROFIBUS-DP Slave with PROFIsafe Editor. After configuration the virtual modules are located on virtual slots.

The following functionalities and procedures are described for the "PROFIBUS-DP Slave" system section, but also apply to the "PROFIBUS-DP-Slave with PROFIsafe" system section:

- ▶ Virtual PROFIBUS modules for receive data (I-modules)
(see Virtual PROFIBUS modules for receive data, [Page 142](#))
- ▶ Virtual PROFIBUS modules for send data (O-modules)
(see Virtual PROFIBUS modules for send data, [Page 140](#))
- ▶ PROFIBUS-DP Slave I/O data
(see PROFIBUS-DP Slave I/O data, [Page 145](#))
- ▶ Address format of PROFIBUS-DP addresses
Address format of PROFIBUS-DP addresses
- ▶ PROFIBUS diagnostics
(see PROFIBUS diagnostics, [Page 150](#))
- ▶ Data consistency on PROFIBUS-DP
(see Data consistency on PROFIBUS-DP, [Page 155](#))

7.5.3 PROFIsafe in the "PROFIBUS-DP Slave with PROFIsafe" system section

PSS 4000 devices with the "PROFIBUS-DP Slave with PROFIsafe" system section send and receive **safety-related data** via PROFIsafe. When exchanging data via PROFIsafe they operate as F-devices. It is always the F-host (F-master) that initiates data transfer. From the perspective of the automation system PSS 4000, the F-host is outside the system; i.e. the receive data is called external I-data and the send data is called external O-data (see Validity of process data, [Page 202](#)).

Send data and receive data

The F-host sends data from its send data area to the receive data area of the "PROFIBUS-DP Slave with PROFIsafe" system section of a PSS 4000 device and requests data from the "PROFIBUS-DP Slave with PROFIsafe" system section of a PSS 4000 device. The "PROFIBUS-DP Slave with PROFIsafe" system section of a PSS 4000 device sends the requested data from its send data area to the receive data area of the F-host.

In the automation system PSS 4000, these data areas are called "virtual PROFIBUS modules". Various pre-defined virtual F-modules are available for data exchange via PROFIsafe. The F-modules consist of virtual PROFIsafe submodules for receive and send data. An F-module has an overall data length of 24 Bytes; 8 of these Bytes are assigned with virtual PROFIsafe submodules for receive data and 8 Bytes are assigned with virtual PROFIsafe submodules for send data.

The virtual PROFIsafe submodules for receive data are called I-submodules. They have an "I" for "Input" in the module name (e.g. 8I_SAFEBOOL, I_SAFEBYTE, I_SAFEWORD, I_SAFEDWORD, I_SAFEINT). The virtual PROFIsafe submodules for send data are called O-submodules. They have an "O" for "Output" in the module name (e.g. 8O_SAFEBOOL, O_SAFEBYTE, O_SAFEWORD, O_SAFEDWORD, O_SAFEINT).

Virtual PROFIBUS modules must be configured for both the F-host and the "PROFIBUS-DP Slave with PROFIsafe" system section of a PSS 4000 device. The F-host obtains the required virtual PROFIBUS modules from the GSD file (see GSD file for virtual PROFIBUS modules, [Page 165](#)).

The F-modules required for the "PROFIBUS-DP Slave with PROFIsafe" system section of a PSS 4000 device are configured during the hardware configuration in PAS4000's PROFIBUS-DP Slave with PROFIsafe Editor. After configuration, an F-module is located on a virtual slot with subslots for the virtual PROFIsafe submodules.

7.5.3.1 Pre-defined F-modules

Pre-defined F-modules are available for a PSS 4000 device in PAS4000. These F-modules are pre-assigned with virtual PROFIsafe submodules (I/O submodules).

The F-module of a PSS 4000 device must have a corresponding F-module in the PROFIsafe F-host. The assignment of the F-modules is as follows:

F_32IX_32IX_32OX_32OX

The F-module F_32IX_32IX_32OX_32OX of a PSS 4000 device must have a corresponding F_32OX_32OX_32IX_32IX in the F-host.

F-host		PSS 4000 device		
F_32OX_32OX_32IX_32IX		F_32IX_32IX_32OX_32OX (default assignment)		
		Virtual subslot	I/O submodules	
Output	Unsigned32	0	8I_SAFEBOOL	Input
		1	8I_SAFEBOOL	
		2	8I_SAFEBOOL	
		3	8I_SAFEBOOL	
	Unsigned32	4	8I_SAFEBOOL	
		5	8I_SAFEBOOL	
		6	8I_SAFEBOOL	
		7	8I_SAFEBOOL	
Input	Unsigned32	8	8O_SAFEBOOL	Output
		9	8O_SAFEBOOL	
		10	8O_SAFEBOOL	
		11	8O_SAFEBOOL	
	Unsigned32	12	8O_SAFEBOOL	
		13	8O_SAFEBOOL	
		14	8O_SAFEBOOL	
		15	8O_SAFEBOOL	

F_32IX_IWA2_32OX_OWA2

The F-module F_32IX_IWA2_32OX_OWA2 of a PSS 4000 device must have a corresponding F_32OX_OWA2_32IX_IWA2 in the F-host.

F-host		PSS 4000 device		
F_32OX_OWA2_32IX_IWA2		F_32IX_IWA2_32OX_OWA2 (default assignment)		
		Virtual subslot	I/O submodules	
Output	Unsigned32	0	8I_SAFEBOOL	Input
		1	8I_SAFEBOOL	
		2	8I_SAFEBOOL	
		3	8I_SAFEBOOL	
	Integer16	4	I_SAFEINT	
Integer16	5	I_SAFEINT		
Input	Unsigned32	6	8O_SAFEBOOL	Output
		7	8O_SAFEBOOL	
		8	8O_SAFEBOOL	
		9	8O_SAFEBOOL	
	Integer16	10	O_SAFEINT	
Integer16	11	O_SAFEINT		

F_IWA4_OWA4

The F-module F_IWA4_OWA4 of a PSS 4000 device must have a corresponding F_OWA4_IWA4 in the F-host.

F-host		PSS 4000 device		
F_OWA4_IWA4		F_IWA4_OWA4		
		Virtual subslot	I/O submodules	
Output	Integer16	0	I_SAFEINT	Input
	Integer16	1	I_SAFEINT	
	Integer16	2	I_SAFEINT	
	Integer16	3	I_SAFEINT	
Input	Integer16	4	O_SAFEINT	Output
	Integer16	5	O_SAFEINT	
	Integer16	6	O_SAFEINT	
	Integer16	7	O_SAFEINT	

7.5.3.2 Virtual PROFIsafe submodules for send data

If an F-module contains O-submodules of SAFEBOOL data type, these can be assigned with O-submodules of another data type in PAS4000. This applies to the two F-modules F_32IX_32IX_32OX_32OX and F_32IX_IWA2_32OX_OWA2, for example. The following O-submodules are available for the configuration:

O-submodule	Data type	Usage
8O_SAFEBOOL	ARRAY [0..7] OF SAFE- BOOL	Virtual bit module Sends an array with 8 elements of SAFEBOOL data type
O_SAFEBYTE	SAFEBYTE	Virtual byte module Sends data of SAFEBYTE data type
O_SAFEWORD	SAFEWORD	Virtual WORD module Sends data of SAFEWORD data type
O_SAFEDWORD	SAFEDWORD	Virtual DWORD module Sends data of SAFEDWORD data type
O_SAFEINT	SAFEINT	Virtual INT module Sends data of SAFEINT data type

7.5.3.3 Virtual PROFIsafe submodules for receive data

If an F-module contains I-submodules of a SAFEBOOL data type, these can be assigned with I-submodules of another data type in PAS4000. This applies to the two F-modules F_32IX_32IX_32OX_32OX and F_32IX_IWA2_32OX_OWA2, for example. The following I-submodules are available for the configuration:

I-submodule	Data type	Usage
8I_SAFEBOOL	ARRAY [0..7] OF SAFE- BOOL	Virtual bit module Receives an array with 8 elements of BOOL data type
I_SAFEBYTE	SAFEBYTE	Virtual byte module Receives data of SAFEBYTE data type
I_SAFEWORD	SAFEWORD	Virtual WORD module Receives data of SAFEWORD data type
I_SAFEDWORD	SAFEDWORD	Virtual DWORD module Receives data of SAFEDWORD data type
I_SAFEINT	SAFEINT	Virtual INT module Receives data of SAFEINT data type

7.5.4 GSD file for virtual PROFIBUS modules

All virtual PROFIBUS modules available for the "PROFIBUS-DP-Slave with PROFIsafe" system section are contained in the corresponding GSD file.



Information

Only use the GSD file that belongs to the "PROFIBUS-DP-Slave with PROFIsafe" system section. It contains both the I-modules and O-modules for the PROFIBUS-DP Master, as well as the F-modules for the PROFIsafe F-host.

The GSD file is supplied with PAS4000 and is also available in the download area of the Pilz homepage (www.pilz.de -> Downloads). "GSD" can be entered as the search term, for example. The software download is only available for registered users. Unregistered users can register free of charge.

7.5.5 Data transfer limits

On a PSS 4000 device that has the "PROFIBUS-DP Slave with PROFIsafe" system section, data transfer is via virtual PROFIBUS modules (I-modules, O-modules, F-modules). The following limits apply for the "PROFIBUS-DP Slave with PROFIsafe" system section of a device:

- ▶ A maximum of 80 virtual PROFIBUS modules can be configured with the virtual slots 0 ... 79
- ▶ A maximum of one F-module can be configured
Please note: An F-module reduces the total number of virtual PROFIBUS modules available not by 1 but by 5.
- ▶ An F-module needs a total of 24 Bytes, of which
 - Exactly 8 Bytes are reserved for I-submodules (virtual PROFIsafe submodules for receive data)
 - Exactly 8 Bytes are reserved for O-submodules (virtual PROFIsafe submodules for send data)
- ▶ An F-module has a max. of 16 subslots for virtual PROFIsafe submodules
- ▶ A max. of 244 Bytes of receive data (= external I-data) can be configured
- ▶ A max. of 244 Bytes of send data (= external O-data) can be configured

7.5.6 PROFIsafe I/O data

The "PROFIBUS-DP Slave with PROFIsafe" system section enables a PSS 4000 device to exchange safety-related I/O data with the F-host via PROFIsafe. For this to work it is necessary for the I/O data in PAS4000's I/O Mapping Editor to be mapped to the virtual PROFIsafe submodules of an F-module.

I/O mapping on F-modules

I/O mapping is used to define which data source supplies data to which data sink.

Possible I/O-mapping for F-modules:

		F-modules	
		Receive data (= external I-data of virtual PROFIsafe submodules such as 8I_SAFEBOOL, I_SAFEBYTE, I_SAFEWORD, I_SAFEDWORD, I_SAFEINT for example)	Send data (= external O-data of virtual PROFIsafe submodules such as 8O_SAFEBOOL, O_SAFEBYTE, O_SAFEWORD, O_SAFEDWORD, O_SAFEINT for example)
Module bus	I-datum (e.g. HW input)	---	◆
	O-datum (e.g. HW output)	◆	---
PI variables	I-PI variable	◆	---
	O-PI variable	---	◆

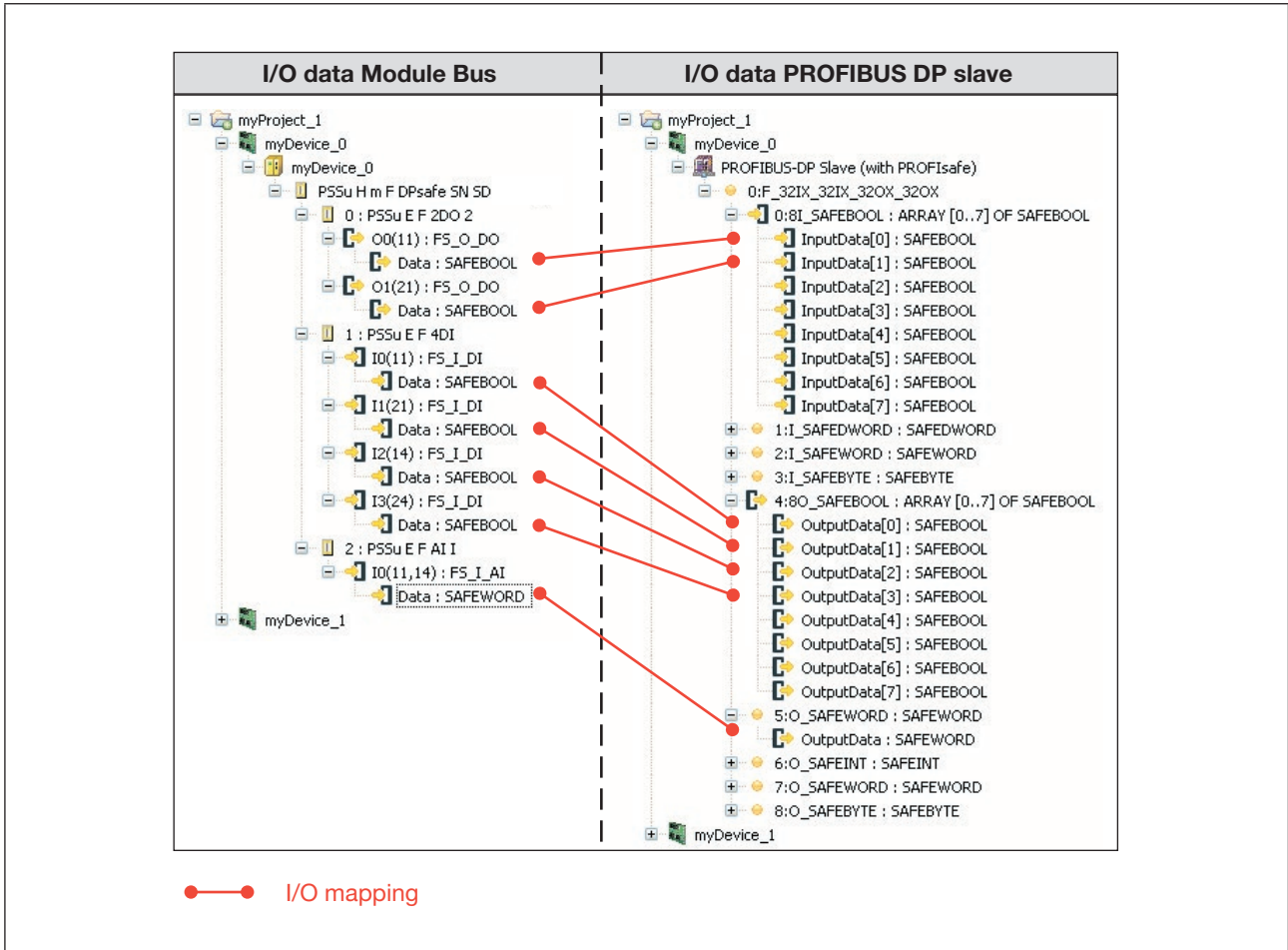
◆ I/O mapping possible

--- I/O mapping impossible

Examples:

▶ **I/O mapping of I/O data**

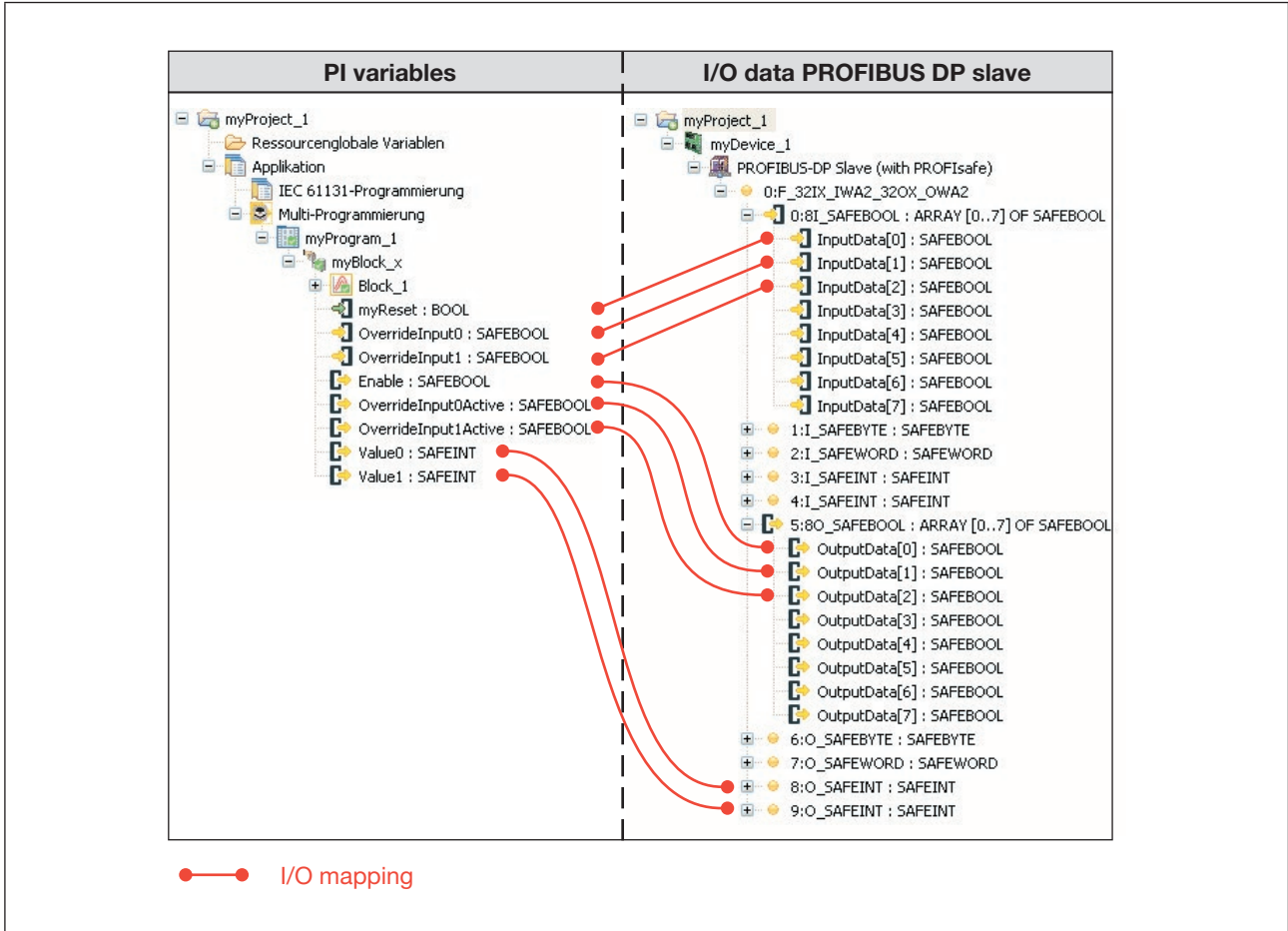
The F-host is to activate a PSSu E F 2DO 2 and the I-data of a PSSu E F 4DI and a PSSu E F AI I is to be transmitted to the F-host. The F-module F_32IX_32IX_32OX_32OX is configured with the necessary virtual I/O submodules for this purpose. Possible I/O-mapping (example):



▶ **I/O mapping of PI variables**

The contents of a block's O-PI variables are to be transmitted to the F-host; receive data arriving from the F-host is to be assigned to I-PI variables. The F-module F_32IX_IWA2_32OX_OWA2 is used for this purpose.

Possible I/O mapping (example):



7.5.7 Address format of PROFIsafe addresses

The addresses for the I/O data of a PSS 4000 device are displayed at several points, e.g. in PAS4000 and in the diagnostics.

7.5.7.1 Address format of send data

The addresses of PROFIsafe send data (O-data of virtual PROFIsafe submodules) are structured as follows:

<Device name>.ProfibusDP.<Virtual slot>.<Virtual subslot>.OutputData[<Element number>]

Examples:

- ▶ **Address format for an O_SAFEBOOL:**
 - myDevice_1.ProfibusDP.1.4.OutputData[0]
 - myDevice_1.ProfibusDP.1.4.OutputData[1]
 - myDevice_1.ProfibusDP.1.4.OutputData[2]
 - myDevice_1.ProfibusDP.1.4.OutputData[3]
 - myDevice_1.ProfibusDP.1.4.OutputData[4]
 - myDevice_1.ProfibusDP.1.4.OutputData[5]
 - myDevice_1.ProfibusDP.1.4.OutputData[6]
 - myDevice_1.ProfibusDP.1.4.OutputData[7]
- ▶ **Address format for an O_SAFEBYTE:**
 - myDevice_1.ProfibusDP.1.5.OutputData
- ▶ **Address format for an O_SAFEWORD:**
 - myDevice_1.ProfibusDP.1.6.OutputData
- ▶ **Address format for an O_SAFEDWORD:**
 - myDevice_1.ProfibusDP.4.3.OutputData
- ▶ **Address format for an O_SAFEINT:**
 - myDevice_1.ProfibusDP.4.4.OutputData

7.5.7.2 Address format of receive data

The addresses of PROFIsafe receive data (I-data of virtual PROFIsafe submodules) are structured as follows:

<Device name>.ProfibusDP.<Virtual slot>.<Virtual subslot>.InputData[<Element number>]

Examples:

▶ **Address format for an 8I_SAFEBOOL:**

myDevice_1.ProfibusDP.1.0.InputData[0]
myDevice_1.ProfibusDP.1.0.InputData[1]
myDevice_1.ProfibusDP.1.0.InputData[2]
myDevice_1.ProfibusDP.1.0.InputData[3]
myDevice_1.ProfibusDP.1.0.InputData[4]
myDevice_1.ProfibusDP.1.0.InputData[5]
myDevice_1.ProfibusDP.1.0.InputData[6]
myDevice_1.ProfibusDP.1.0.InputData[7]

▶ **Address format for an I_SAFEBYTE:**

myDevice_1.ProfibusDP.1.1.InputData

▶ **Address format for an I_SAFEWORD:**

myDevice_1.ProfibusDP.1.2.InputData

▶ **Address format for an I_SAFEWORD:**

myDevice_1.ProfibusDP.4.0.InputData

▶ **Address format for an I_SAFEINT:**

myDevice_1.ProfibusDP.4.1.InputData

7.5.8 Deactivation and reactivation of an F-module

Send data that is transmitted to an F-module can be brought to a safe condition selectively via the F-host. This is called deactivation/reactivation of an F-module.

- ▶ Deactivation of an F-module
 - The VALID bit becomes FALSE (see Validity of process data, [Page 202](#)).
 - The I-submodules of an F-module are set to their safe values (= zero).
- ▶ Reactivation of an F-module
 - The VALID bit becomes TRUE, provided there is no PROFIsafe module error.
 - The I-submodules of an F-module contain their process value.

7.5.9 F-parameters

F-parameters are available to adapt the PROFIsafe protocol to the respective F-device. The F-parameters are part of the GSD file.

During the network configuration, the Master Tool (e.g. Siemens Software HW Config) can be used to display and configure the F-parameters and to download them to the F-device. The F-host uses the F-parameters to check the connection with an F-device. For various F-parameters, indication will be provided if the content of the F-parameter on the F-host does not match that of the F-device (see PROFIsafe diagnostics, [Page 174](#)).

As an F-Device, a PSS 4000 device supports the following F-parameters:

F-parameters	Default configuration	Configuration in the Master Tool	Meaning
F_Check_SeqNr	1	Not configurable	The sequence number is included in the consistency check (CRC calculation).
F_Check_iPar	No Check	Not configurable	No iParameters
F_SIL	SIL3	Not configurable	SIL of F-device
F_CRC_Length	3 Octets	Not configurable	Length of the CRC2 key
F_Block_ID	No F_iPar_CRC	Not configurable	Type identification for parameters
F_Par_Version	V2 Mode	Not configurable	PROFIsafe version implemented within F-device
F_Source_Add	1	Value range: 1 ... 65534	F-address of F-host (automatically assigned by the Master Tool)
F_Dest_Add	1	Value range: 1 ... 65534	F-address of F-device
F_WD_Time	200 ms (because default t_{ExtCo} of PROFIBUS-DP Slave with PROFIsafe = 100 ms)	Value range: F_WD_Time_min ... 65535	Monitoring time in F-device; A valid, current safety telegram must arrive from the F-host within the monitoring time F_WD_Time. If the monitoring time is maintained, safe process data will be delivered to the other system sections via PROFIsafe. If the monitoring time is exceeded, data transfer via PROFIsafe is stopped and the substitute values are used for the affected process data with Valid Bits = FALSE. F_WD_Time must correspond to min. 2 x the cycle time (t_{ExtCo}) of PROFIBUS-DP Slave with PROFIsafe. F_WD_Time_min depends on the device.

7.5.10 PROFIsafe diagnostics

PROFIsafe diagnostics can be performed via the PSS 4000 system diagnostics (see Diagnostic basics, [Page 470](#)).

F-devices/F-modules can register diagnostic information via the diagnostic mechanisms defined by PROFIBUS-DP. The diagnostic information is downloaded in the form of special codes. The codes and their meaning are permanently specified through the standard.

F-modules of a PSS 4000 device support the following diagnostic information:

Hex	Number	Diagnostic information
0x0040	64	Mismatch of safety destination address (F_Dest_Add) The value entered in the F-host for the parameter "F_Dest_Add" does not match the local PROFIsafe address.
0x0041	65	Safety destination address not valid (F_Dest_Add) The value entered in the F-host for the parameter "F_Dest_Add" is invalid. Valid entry in the F-host: 1 ... 65534
0x0042	66	Safety source address not valid (F_Source_Add) The value entered in the F-host for the parameter "F_Source_Add" is invalid. Valid entry in the F-host: 1 ... 65534
0x0043	67	Safety watchdog time value is 0 ms (F_WD_Time) The value entered in the F-host for the parameter "F_WD_Time" (0 ms) is invalid. Valid entry in the F-host [ms]: 1 ... 65534
0x0044	68	Parameter "F_SIL" exceeds SIL from specific device application The value entered in the F-host for the parameter "F_SIL" is greater than the max. SIL on PSS 4000 (SIL 3). Valid entry in the F-host (max.): "SIL3"
0x0045	69	Parameter "F_CRC_Length" does not match the generated values The value entered in the F-host for the parameter "F_CRC_Length" is not supported. Valid entry in the F-host: "3 octet CRC"
0x0046	70	Version of F-Parameter set incorrect The value entered in the F-host for the parameter "F_Parameter_Version" is not supported. Valid entry in the F-host: "V2 Mode"
0x0047	71	CRC1-Fault The "CRC1" check sums in the F-host and virtual F-module do not match.

7.5.11 Data consistency on PROFIsafe

The F-host is responsible for the data consistency of the receive data sent by the F-host. From the PSS 4000's perspective, no statement can be made regarding the data consistency of the F-host.

Data consistency of the receive data

From the perspective of the PSS 4000, the receive data of an F-module is external I-data. As soon as external I-data is available in an F-module, it is treated as consistent data (see Data consistency, [Page 214](#)). The external I-data of a virtual F-module forms a consistency range of the consistency range type "FS-I data of an external communication connection". The exact description is consistency range "FS-I data of a virtual PROFIsafe F-module" (see Consistency ranges, [Page 215](#)).

Data consistency of the send data

From the perspective of the PSS 4000, the send data of an F-module is external O-data. The external O-data of an F-module is consistent if it comes from the same consistency range on a data source (see Data consistency, [Page 214](#)). PSS 4000 always treats the external O-data of an F-module as one unit. It is the task of the F-host to guarantee the necessary data consistency.

7.6 EtherNet/IP Adapter

EtherNet/IP (Ethernet Industrial Protocol, abbreviation EIP) is an Ethernet-based fieldbus. With EtherNet/IP an open industrial standard has been defined, which expands the classic Ethernet with an industrial protocol (Common Industrial Protocol, abbreviation CIP), as an application layer for automation applications.

The open industrial standard EtherNet/IP was drawn up jointly by ControlNet International (CI) and the Open DeviceNet Vendor Association (ODVA), with help from the Industrial Ethernet Association (IEA). EtherNet/IP is standardised in the international IEC 61158 series of standards.

Common Industrial Protocol (CIP)

Implicit and explicit messages are used to transfer CIP message data in EtherNet/IP networks.

Implicit messages are typically smaller data packets containing I/O data for time-critical data transfers. Implicit messages are transferred via UDP.

Non-time-critical messages are transferred via explicit messages. Explicit messages contain configuration and information data, for example. They use the TCP/IP communication mechanism.

CIP is object-oriented. ODVA uses device profiles to define which objects and attributes must be supported by devices of a certain device class. Optional and manufacturer-defined objects and attributes are also possible. CIP uses an electronic data sheet, (abbreviation EDS) as a common basis for configuration.

Further information on the Common Industrial Protocol (CIP) or on EtherNet/IP is available from the ODVA (www.odva.org).

EtherNet/IP-enabled PSS 4000 devices from Pilz

For **non**-safety-related applications, an EtherNet/IP-enabled PSS 4000 device from Pilz can be incorporated as an adapter in EtherNet/IP.

An EtherNet/IP-enabled PSS 4000 device from Pilz supports the following as an EtherNet/IP Adapter:

- ▶ Communications Adapter device profile, device type 12_D in accordance with the "Common Industrial Protocol Specification", Volume 1 of the ODVA. Development and conformance test have been carried out in accordance with Edition 3.12, April 2012.
- ▶ Cyclical data exchange with an EtherNet/IP scanner
 - I/O Server functionality for I/O data from implicit messages
 - The I/O data is transferred via UDP and uses Port 2222.
- ▶ Acyclical data exchange as "Explicit Message Server"
 - Explicit Message Server functionality for explicit data
 - Support for acyclical data exchange with the following objects: Identity Object, TCP/IP Object and Link Object
- ▶ Identity Object
- ▶ TCP/IP Object
- ▶ Link Object

- ▶ Assembly Object with
 - An Output Assembly Object instance
 - An Input Assembly Object instance

7.6.1 "EtherNet/IP Adapter" system section

In the automation system PSS 4000, external communication via EtherNet/IP is possible with all PSS 4000 devices that have the "EtherNet/IP Adapter" system section. In this case, the EtherNet/IP Adapter system section of a PSS 4000 device performs all the tasks required for communication. These include:

- ▶ Administration of the supported objects (see Supported EtherNet/IP objects and Common Services, [Page 192](#))
- ▶ Data exchange via EtherNet/IP (see Data exchange via EtherNet/IP, [Page 180](#))

7.6.2 EtherNet/IP Adapter connections

Precisely one EtherNet/IP Adapter connection is currently supported per PSS 4000 device (Adapter). With this EtherNet/IP Adapter connection, data exchange can occur via precisely one Output Assembly Object instance and precisely one Input Assembly Object instance (see Assembly Object, [Page 195](#)).

7.6.3 Data exchange via EtherNet/IP

PSS 4000 devices with EtherNet/IP Adapter system section support data exchange as EtherNet/IP Adapters. It is always the EtherNet/IP scanner that initiates data transfer. From the perspective of the automation system PSS 4000, the EtherNet/IP scanner is outside the system.

7.6.3.1 Send data and receive data

With EtherNet/IP, the actual usable data (process data) is exchanged via the Assembly Object. A PSS 4000 device with EtherNet/IP Adapter system section supports the Assembly Object with one Output Assembly Object instance and one Input Assembly Object instance respectively. On the PSS 4000, the EtherNet/IP terms "Input" and "Output" are defined from the scanner's perspective. The advantage of this definition is that the views are identical in both the scanner tool and in PAS4000; i.e. the Output Assembly Object instance in the scanner tool is also the Output Assembly Object instance in PAS4000 and the Input Assembly Object instance in the scanner tool is also the Input Assembly Object instance in PAS4000.

To illustrate this point, on PSS 4000 the Output Assembly Object instance is called the "Scanner Output Assembly Object instance" and the Input Assembly Object instance is called the "Scanner Input Assembly Object instance".

Scanner Output Assembly Object instance

The Scanner Output Assembly Object instance contains the data that the scanner produces and places on the network; i.e. it contains the data received by a PSS 4000 device (Adapter). The data length of the data transferred using the Scanner Output Assembly Object instance must be configured equally in both the scanner and the PSS 4000 device. For the PSS 4000 device, this configuration is made in the EtherNet/IP Adapter Editor of PAS4000. As the Scanner Output Assembly Object instance contains receive data (= external I-data) from the perspective of the PSS 4000 device, virtual I-modules must be configured in the EtherNet/IP Adapter Editor. Virtual I-modules have an "I" for "Input" in the module name (e.g. 16IX, IB, IW, ID).

A PSS 4000 device expects the scanner to always send the run/idle header.

Scanner Input Assembly Object instance

The Scanner Input Assembly Object instance contains the data that the scanner consumes from the network; i.e. it contains the data sent by a PSS 4000 device (Adapter). The data length of the data transferred using the Scanner Input Assembly Object instance must be configured equally in both the scanner and the PSS 4000 device. For the PSS 4000 device, this configuration is made in the EtherNet/IP Adapter Editor of PAS4000.

As the Scanner Input Assembly Object instance contains send data (= external O-data) from the perspective of the PSS 4000 device, virtual O-modules must be configured in the EtherNet/IP Adapter Editor. Virtual O-modules have an "O" for "Output" in the module name (e.g. 16OX, OB, OW, OD).

A status word for sending PSS 4000 status information to the scanner can be configured as an option.

7.6.3.2 Virtual modules for send data

From the perspective of the PSS 4000 device (Adapter), the Scanner Input Assembly Object instance contains send data (= external O-data). The following virtual modules can be configured in PAS4000 for the send data of a PSS 4000 device:

Virtual module	Data type	Usage
16OX	ARRAY [0..15] OF BOOL	Virtual EtherNet/IP bit module To send an array with 16 elements of data type BOOL
OBA _n	ARRAY [0...(n-1)] OF BYTE	Virtual EtherNet/IP byte module To send an array with n elements of data type BYTE n = 2, 4, 8, 16, 32 or 64
OW	WORD	Virtual EtherNet/IP byte module To send data of data type WORD
OWA _n	ARRAY [0...(n-1)] OF WORD	Virtual EtherNet/IP byte module To send an array with n elements of data type WORD n = 2, 4, 8, 16, 32 or 64
OD	DWORD	Virtual EtherNet/IP byte module To send data of data type DWORD

The data length of the data transferred using the Scanner Input Assembly Object instance must be configured equally in both the scanner and the PSS 4000 device.

7.6.3.3 Virtual modules for receive data

From the perspective of the PSS 4000 device (Adapter), the Scanner Output Assembly Object instance contains receive data (= external I-data). The following virtual modules can be configured in PAS4000 for the receive data of a PSS 4000 device:

Virtual module	Data type	Usage
16IX	ARRAY [0..15] OF BOOL	Virtual EtherNet/IP bit module To receive an array with 16 elements of data type BOOL
IBAn	ARRAY [0...(n-1)] OF BYTE	Virtual EtherNet/IP byte module To receive an array with n elements of data type BYTE n = 2, 4, 8, 16, 32 or 64
IW	WORD	Virtual EtherNet/IP byte module To receive data of data type WORD
IWAn	ARRAY [0...(n-1)] OF WORD	Virtual EtherNet/IP byte module To receive an array with n elements of data type WORD n = 2, 4, 8, 16, 32 or 64
ID	DWORD	Virtual EtherNet/IP byte module To receive data of data type DWORD

The data length of the data transferred using the Scanner Output Assembly Object instance must be configured equally in both the scanner and the PSS 4000 device.

7.6.4 Data transfer limits

When transferring data via EtherNet/IP, the following limits apply for a PSS 4000 device (Adapter):

- ▶ A maximum of one connection (EtherNet/IP Adapter connection) to the scanner can be configured per PSS 4000 device.
- ▶ A PSS 4000 device supports a maximum of one Scanner Output Assembly Object instance and one Scanner Input Assembly Object instance.
- ▶ Virtual I-modules (external I-data) with a total data length of 0 ... 504 Bytes can be configured for a Scanner Output Assembly Object instance.
The total data length of a Scanner Output Assembly Object instance is a maximum of 510 Bytes (504 Bytes for virtual I-modules, 4 Bytes for the run/idle header and 2 Bytes for the Sequence Count).
- ▶ Virtual O-modules (external O-data) with a total data length of 0 ... 508 Bytes can be configured for a Scanner Input Assembly Object instance.
The total data length of a Scanner Input Assembly Object instance is a maximum of 510 Bytes (508 Bytes for virtual O-modules, 2 Bytes for the Sequence Count).

7.6.5 EtherNet/IP Adapter I/O data

The "EtherNet/IP Adapter" system section enables a PSS 4000 device to exchange I/O data with the EtherNet/IP scanner via EtherNet/IP. For this to work it is necessary for the I/O data in PAS4000's I/O Mapping Editor to be mapped to the corresponding virtual modules.

I/O-mapping of virtual modules

I/O mapping is used to define which data source supplies data to which data sink.

Possible I/O-mapping for virtual modules:

		Virtual modules	
		Data of the Scanner Output Assembly Object instance (= external I-data of I-modules such as 16IX, IBAn, IWAn, ID)	Data of the Scanner Input Assembly Object instance (= external O-data of O-modules such as 16OX, OBAn, OWAn, OD)
Module bus	I-datum (e.g. HW input)	---	◆
	O-datum (e.g. HW output)	◆	---
PI variables	I-PI variable	◆	---
	O-PI variable	---	◆

◆ I/O mapping possible

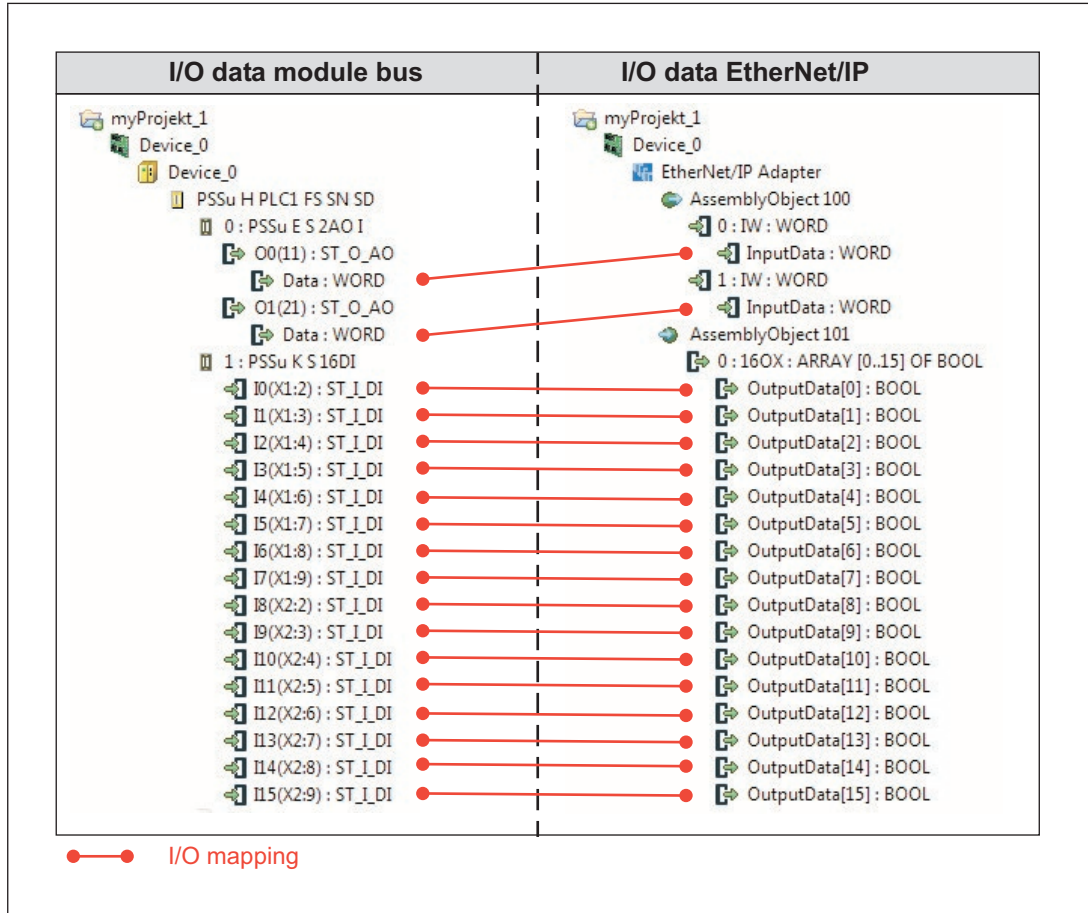
--- I/O mapping impossible

Examples

▶ **I/O mapping of I/O data**

The input signals from hardware inputs on a PSSu K S 16DI are to be transmitted to the scanner and the scanner is to control the hardware outputs on the PSSu E S 2 AO I.

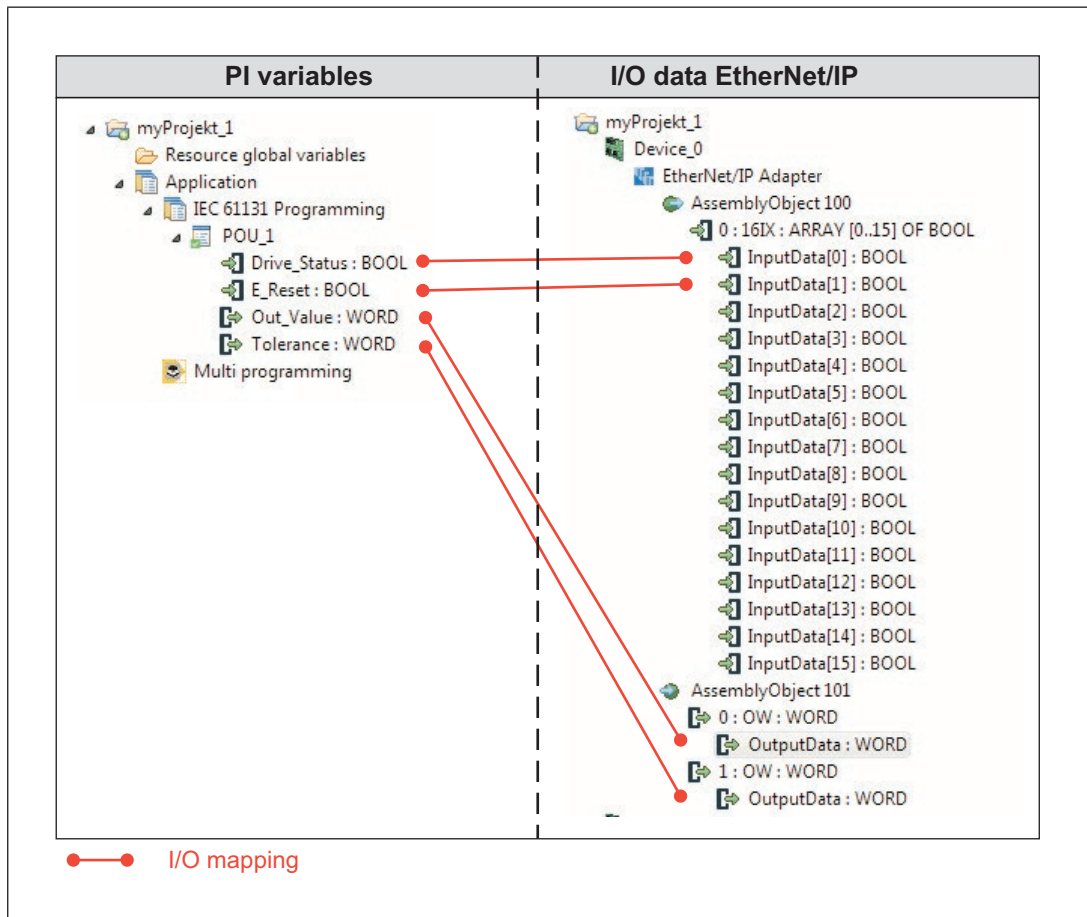
Possible I/O mapping (example):



▶ **I/O mapping of PI variables**

The contents of a block's O-PI variables are to be transmitted to the scanner and I-PI variables are to be assigned to the receive data.

Possible I/O-mapping (example):



7.6.6 Address format of EtherNet/IP addresses

The addresses for the I/O data of a PSS 4000 device are displayed at several points, e.g. in PAS4000 and in the diagnostics.

7.6.6.1 Address format of send data

The addresses of EtherNet/IP send data (O-data of virtual modules) are structured as follows:

<Device Name>.EtherNetIP.AObj<Instance Number>.<Virtual Slot>.OutputData[<Element Number>]

Examples:

- ▶ Address format for a 16OX:

```
myDevice_1.EtherNetIP.AObj101.VirtualModule2.OutputData[0]
myDevice_1.EtherNetIP.AObj101.VirtualModule2.OutputData[1]
myDevice_1.EtherNetIP.AObj101.VirtualModule2.OutputData[2]
myDevice_1.EtherNetIP.AObj101.VirtualModule2.OutputData[3]
myDevice_1.EtherNetIP.AObj101.VirtualModule2.OutputData[4]
myDevice_1.EtherNetIP.AObj101.VirtualModule2.OutputData[5]
myDevice_1.EtherNetIP.AObj101.VirtualModule2.OutputData[6]
myDevice_1.EtherNetIP.AObj101.VirtualModule2.OutputData[7]
myDevice_1.EtherNetIP.AObj101.VirtualModule2.OutputData[8]
myDevice_1.EtherNetIP.AObj101.VirtualModule2.OutputData[9]
myDevice_1.EtherNetIP.AObj101.VirtualModule2.OutputData[10]
myDevice_1.EtherNetIP.AObj101.VirtualModule2.OutputData[11]
myDevice_1.EtherNetIP.AObj101.VirtualModule2.OutputData[12]
myDevice_1.EtherNetIP.AObj101.VirtualModule2.OutputData[13]
myDevice_1.EtherNetIP.AObj101.VirtualModule2.OutputData[14]
myDevice_1.EtherNetIP.AObj101.VirtualModule2.OutputData[15]
```

- ▶ Address format for an OBA2:

```
myDevice_1.EtherNetIP.AObj101.VirtualModule4.OutputData[0]
myDevice_1.EtherNetIP.AObj101.VirtualModule4.OutputData[1]
```

- ▶ Address format for an OW:

```
myDevice_1.EtherNetIP.AObj101.VirtualModule5.OutputData
```

- ▶ Address format for an OWA4:

```
myDevice_1.EtherNetIP.AObj101.VirtualModule6.OutputData[0]
myDevice_1.EtherNetIP.AObj101.VirtualModule6.OutputData[1]
myDevice_1.EtherNetIP.AObj101.VirtualModule6.OutputData[2]
myDevice_1.EtherNetIP.AObj101.VirtualModule6.OutputData[3]
```

- ▶ Address format for an OD:

```
myDevice_1.EtherNetIP.AObj101.VirtualModule7.OutputData
```

7.6.6.2 Address format of receive data

The addresses of EtherNet/IP receive data (I-data of virtual modules) are structured as follows:

<Device Name>.EtherNetIP.AObj<Instance Number>.<Virtual Slot>.InputData[<Element Number>]

Examples:

- ▶ Address format for a 16IX:

```
myDevice_1.EtherNetIP.AObj100.VirtualModule8.InputData[0]
myDevice_1.EtherNetIP.AObj100.VirtualModule8.InputData[1]
myDevice_1.EtherNetIP.AObj100.VirtualModule8.InputData[2]
myDevice_1.EtherNetIP.AObj100.VirtualModule8.InputData[3]
myDevice_1.EtherNetIP.AObj100.VirtualModule8.InputData[4]
myDevice_1.EtherNetIP.AObj100.VirtualModule8.InputData[5]
myDevice_1.EtherNetIP.AObj100.VirtualModule8.InputData[6]
myDevice_1.EtherNetIP.AObj100.VirtualModule8.InputData[7]
myDevice_1.EtherNetIP.AObj100.VirtualModule8.InputData[8]
myDevice_1.EtherNetIP.AObj100.VirtualModule8.InputData[9]
myDevice_1.EtherNetIP.AObj100.VirtualModule8.InputData[10]
myDevice_1.EtherNetIP.AObj100.VirtualModule8.InputData[11]
myDevice_1.EtherNetIP.AObj100.VirtualModule8.InputData[12]
myDevice_1.EtherNetIP.AObj100.VirtualModule8.InputData[13]
myDevice_1.EtherNetIP.AObj100.VirtualModule8.InputData[14]
myDevice_1.EtherNetIP.AObj100.VirtualModule8.InputData[15]
```

- ▶ Address format for an IBA2

```
myDevice_1.EtherNetIP.AObj100.VirtualModule9.InputData[0]
myDevice_1.EtherNetIP.AObj100.VirtualModule9.InputData[1]
```

- ▶ Address format for an IW:

```
myDevice_1.EtherNetIP.AObj100.VirtualModule10.InputData
```

- ▶ Address format for an IWA4:

```
myDevice_1.EtherNetIP.AObj100.VirtualModule11.InputData[0]
myDevice_1.EtherNetIP.AObj100.VirtualModule11.InputData[1]
myDevice_1.EtherNetIP.AObj100.VirtualModule11.InputData[2]
myDevice_1.EtherNetIP.AObj100.VirtualModule11.InputData[3]
```

- ▶ Address format for an ID:

```
myDevice_1.EtherNetIP.AObj100.VirtualModule12.InputData
```

7.6.7 EtherNet/IP diagnostics

As an option, the transmission of a status word can be configured in PAS4000 for the Scanner Input Assembly Object instance. The status word contains device-based status information and is transmitted to the scanner for diagnostic purposes.

Structure and content of the status word

Octet 1 contains error messages from the PSS 4000 device:

Bit	Value	Meaning
0	0/1	Error message for the PSS 4000 device
		0 No message
		1 <ul style="list-style-type: none"> ▶ A message of "Error" severity is present for at least one system section of the PSS 4000 device or ▶ A major FS error is present for at least one FS system section of the PSS 4000 device On PSSu systems, this corresponds to the status of the "DIAG" LED on the head module: Status: "lights up red" or "flashes red".
1	0/1	Error in the FS/ST module bus system sections of the PSS 4000 device
		0 No message
		1 <ul style="list-style-type: none"> ▶ Operating state: "Safe condition of all FS outputs on the PSSu system" or ▶ At least one module cannot be accessed (e.g. a module has been removed during operation, actual/registered hardware does not match) or ▶ Operating state: "FS module bus in a STOP condition with error: Major FS error" On PSSu systems, this corresponds to the status of the "MBUS" LED on the head module: Status: "lights up red" or "flashes red".
2	0/1	Error in the "FS-SafetyNET p RTFN" system section of the PSS 4000 device
		0 No message
		1 <ul style="list-style-type: none"> ▶ Operating state: "FS SafetyNET p RTFN in STOP condition with error: Major FS error" or ▶ Operating state: "FS SafetyNET p RTFN in STOP condition with error: Major FS+ST error" On PSSu systems, this corresponds to the status of the "FS Snp" LED on the head module: Status: "lights up red" or "flashes red".

Bit	Value	Meaning
3	0/1	Error in the "FS-SafetyNET p RTFN" system section of the PSS 4000 device
		0 No message
		1 Operating state: "ST SafetyNETp RTFN in STOP condition with error": Major FS+ST error On PSSu systems, this corresponds to the status of the "ST SNp" LED on the head module: Status: "lights up red" or "flashes red".
4	0	Reserved
5	0	Reserved
6	0	Reserved
7	0	Reserved

Octet 2 contains additional status information from the PSS 4000 device:

Bit	Value	Meaning
0	0/1	Warning message for the PSS 4000 device
		0 No message
		1 A message of "Warning" severity is present for the PSS 4000 device. On PSSu systems, this corresponds to the status of the "DIAG" LED on the head module: Status: "lights up orange".
1	0/1	Forcing on the FS resource
		0 No message
		1 Forcing on the FS resource is active On PSSu systems, this corresponds to the status of the "FS FORCE" LED on the head module: Status: "lights up yellow".
2	0/1	Forcing on the ST resource
		0 No message
		1 Forcing on the ST resource is active On PSSu systems, this corresponds to the status of the "ST FORCE" LED on the head module: Status: "lights up yellow".
4	0	Reserved
5	0	Reserved
6	0	Reserved
7	0	Reserved

7.6.8 Data consistency on EtherNet/IP

The scanner is responsible for the data consistency of the receive data (Scanner Output instance of the Assembly Object), which is sent by the scanner via an EtherNet/IP connection. From the PSS 4000's perspective, no statement can be made regarding the data consistency of the scanner.

Data consistency on receive data via the Scanner Output Assembly Object instance

If a PSS 4000-exclusive/operable device is the recipient of data from an EtherNet/IP connection, from the perspective of the device the receive data is external I-data.

External I-data from an EtherNet/IP connection is treated as consistent data in PSS 4000. The external I-data from an EtherNet/IP connection forms a consistency range of the consistency range type "ST-I data of an external communication connection". The exact description is consistency range "ST-I data of an EtherNet/IP Assembly Object instance".

Data consistency on send data via the Scanner Input Assembly Object instance

If a PSS 4000-exclusive/operable device in an EtherNet/IP connection is the transmitter of data, from the perspective of the device the send data is external O-data.

The external O-data of an EtherNet/IP connection is consistent if it comes from the same consistency range on a data source.

7.6.9 Supported EtherNet/IP objects and Common Services

For data exchange via EtherNet/IP, an EtherNet/IP subscriber is composed of a collection of CIP objects. CIP objects are divided into classes, instances and attributes.

PSS 4000 devices with EtherNet/IP Adapter system section support the following object classes as Adapters.

Class ID	Name of Object Class
0x01	Identity Object
0x02	Message Router Object
0x04	Assembly Object
0x06	Connection Manager Object
0xF5	TCP/IP Interface Object
0xF6	(Network Specific) Link Object

Appropriate diagnostic tools (e.g. tools from Rockwell) can be used to display the content of attributes. Various [Common Services](#) are supported.

7.6.9.1 Identity Object (Class ID 0x01)

The Identity Object is available on each PSS 4000 device with EtherNet/IP Adapter system section. It contains the device identification and other general information about the device.

Instance attributes of the Identity Object instance

Attribute ID	Name	Value	Description
1	Vendor ID	181 (0x00B5)	Manufacturer identification from Pilz
2	Device Type	12 (0x000C)	General product type description (12 = Communications Adapter)
3	Product Code	Default: 1 (0x0001)	Product's EtherNet/IP product code (e.g. 2070 for PSSu H PLC1 FS SN SD)
4	Revision	- - -	Version of the product's PSS 4000 firmware <Major-No>. <Minor-No> (e.g. 1.8) The 3rd digit of the version number is not displayed.
	Major Revision	<Major-No>	
	Minor Revision	<Minor-No>	

Attribute ID	Name	Value		Description
		Bit	Content	
5	Status			Current product status
		0	0	Unused
		1	0	Reserved
		2	1	EtherNet/IP has been configured
		3	0	Reserved
		4 to 7	0	Unused
			1	
			2	At least one EtherNet/IP connection is faulty
			3	No EtherNet/IP connection has been established
			4	Unused
			5	A major FS error is present for at least one FS system section (see diagnostic list) or A message of "Error" severity is present for at least one system section (see diagnostic list)
			6	At least one EtherNet/IP connection is in a RUN condition
			7	At least one EtherNet/IP connection has been established; all EtherNet/IP connections are idle
			8, 9	Reserved
			10 to 15	Unused
		8	0	No message of "Warning" severity is present for the device.
			1	A message of "Warning" severity is present for the device (see diagnostic list)
9	0	Unused		
10	0	There is no major FS error present on any FS system section and there is no message of "Error" severity present on any system section.		
	1	A major FS error is present for at least one FS system section (see diagnostic list) or A message of "Error" severity is present for at least one system section (see diagnostic list)		
11	0	Unused		
12 to 15	0			
6	Serial Number	<Serial No.>		Product serial number (e.g. serial number of head module on PSSu systems)
7	Product Name	Default: PSS 4000		Product's EtherNet/IP product name (e.g. PSSu H PLC1 FS SN SD)

7.6.9.2 Assembly Object (Class ID 0x04)

Data exchange between the scanner and adapter occurs via the Scanner Output Assembly Object instance and the Scanner Input Assembly Object instance of the Assembly Object.

Attributes of the Scanner Output Assembly Object instance

Attribute ID	Name	Value	Description
1	Attributes are not supported		
2			
3	Data	<Data>	The attribute contains the data that the scanner produces and places on the network in accordance with the configuration. From the perspective of the PSS 4000 device, this is the external I-data.
4	Size	<Data Length>	Data length in accordance with the configuration

Attributes of the Scanner Input Assembly Object instance

Attribute ID	Name	Value	Description
1	Attributes are not supported		
2			
3	Data	<Data>	The attribute contains the data that the scanner consumes from the network in accordance with the configuration. From the perspective of the PSS 4000 device, this is the external O-data.
4	Size	<Data Length>	Data length in accordance with the configuration

7.6.9.3 TCP/IP Interface Object (Class ID 0xF5)

The TCP/IP Interface Object can be used to scan a device's TCP/IP properties (IP address, subnet mask, Gateway address etc.). On a PSS 4000 device, all TCP/IP properties contained in the naming data (see Naming data, [Page 223](#)) are supported. So the TCP/IP Interface Object of a PSS 4000 device with EtherNet/IP Adapter system section obtains this configuration via the naming data during device naming.

Attributes of the TCP/IP Interface Object

Attribute ID	Name	Value	Description
1	State	0	No data exchange configured via EtherNet/IP
		1	Data exchange configured via EtherNet/IP
2	Configuration capability	32 (0x00000020)	Configuration of the TCP/IP properties via device naming
3	Configuration control	0	Configuration of the TCP/IP properties via device naming is static
4	Physical Link Object	<Path to Physical Link Object>	The device has exactly one instance of the Physical Link Object; the attribute contains the path to this instance of the Physical Link Object
		0	The device has more than one instance of the Physical Link Object
5	Interface configuration	<IP address, subnet mask, Gateway address etc.>	TCP/IP properties in accordance with the naming data The attribute is "read only"; i.e. the Set-Service is not supported.
6	Host name	<Device name>	Device name from naming data The attribute is "read only"; i.e. the Set-Service is not supported.
		0	No host is configured and no host can be configured
7	Attribute is not supported		
8	TTL Value	1	Time To Live (TTL) for EtherNet/IP packets via IP-Multicast
9	Mcast Config	- - -	Configuration of the IP-Multicast addresses that are used for EtherNet/IP Multicast packets; Overall data length of attribute: 8 Bytes
	Alloc Control (1 Byte)	0	The IP-Multicast addresses are assigned via the specific default allocation algorithm.
		1	The IP-Multicast addresses are assigned using "Num Mcast" and "Mcast Start Addr".
		2	Reserved
	Reserved (1 Byte)		
	Num Mcast (2 Bytes)	<Device-dependent>	Number of assigned IP-Multicast addresses; Start address see "Mcast Start Addr"
	Mcast Start Addr (4 Bytes)	<Device-dependent>	Start address for the number of IP-Multicast addresses defined under "Num Mcast"

7.6.9.4 Link Object (Class ID 0xF6)

The Link Object contains information about the status of the physical interface assigned to it.

Attribute of Link Object

Attribute ID	Name	Value	Description	
1	Interface Speed	0	Poor connection or no connection	
		<Transmission rate>	Current transmission rate (10, 100, 1000, 10000 MBit/s)	
2	Interface Status Flags	Bit	Content	
		0	0	No Ethernet connection available
			1	Ethernet connection available
		1	0	Half duplex mode
			1	Full duplex mode
		2 to 4	0	Performing autonegotiation
			1	Autonegotiation failed
			2	Autonegotiation failed but transmission rate was detected
			3	Autonegotiation was successful
		5	0	Not supported
			1	Hardware error detected
		6	0	No hardware error found
1	Hardware error detected			
3	Physical Address	<MAC Address>	Device's MAC address	
4, 5, 6	Attributes are not supported			
7	Interface Type	0	Unknown	
		1	Internal (not supported)	
		2	Twisted Pair	
		3	Fibre-optic cable	
8	Interface State	0	Unknown	
		1	Interface is activated and ready to send/receive data	
		2	Deactivated	
		3	Test mode	
9	Attribute is not supported			
10	Interface Label	<Interface name>	Text that names the interface (e.g. "Port X3")	

7.6.9.5 Common Services

Common Services are supported for the class and instance attributes as follows:

▶ **Get Services**

The contents of attributes with the Get-Services **Get_Attribute_All** (Service Code 0x01) and **Get_Attribute_Single** (Service Code 0x0E) can be polled for diagnostic purposes. Tools from third-party manufacturers (e.g. tools from Rockwell) support these Services. If non-supported attributes are accessed, the error code for "Attribute not supported" (0x14) is issued.

▶ **Set Services**

Set Services are supported for the following attributes:

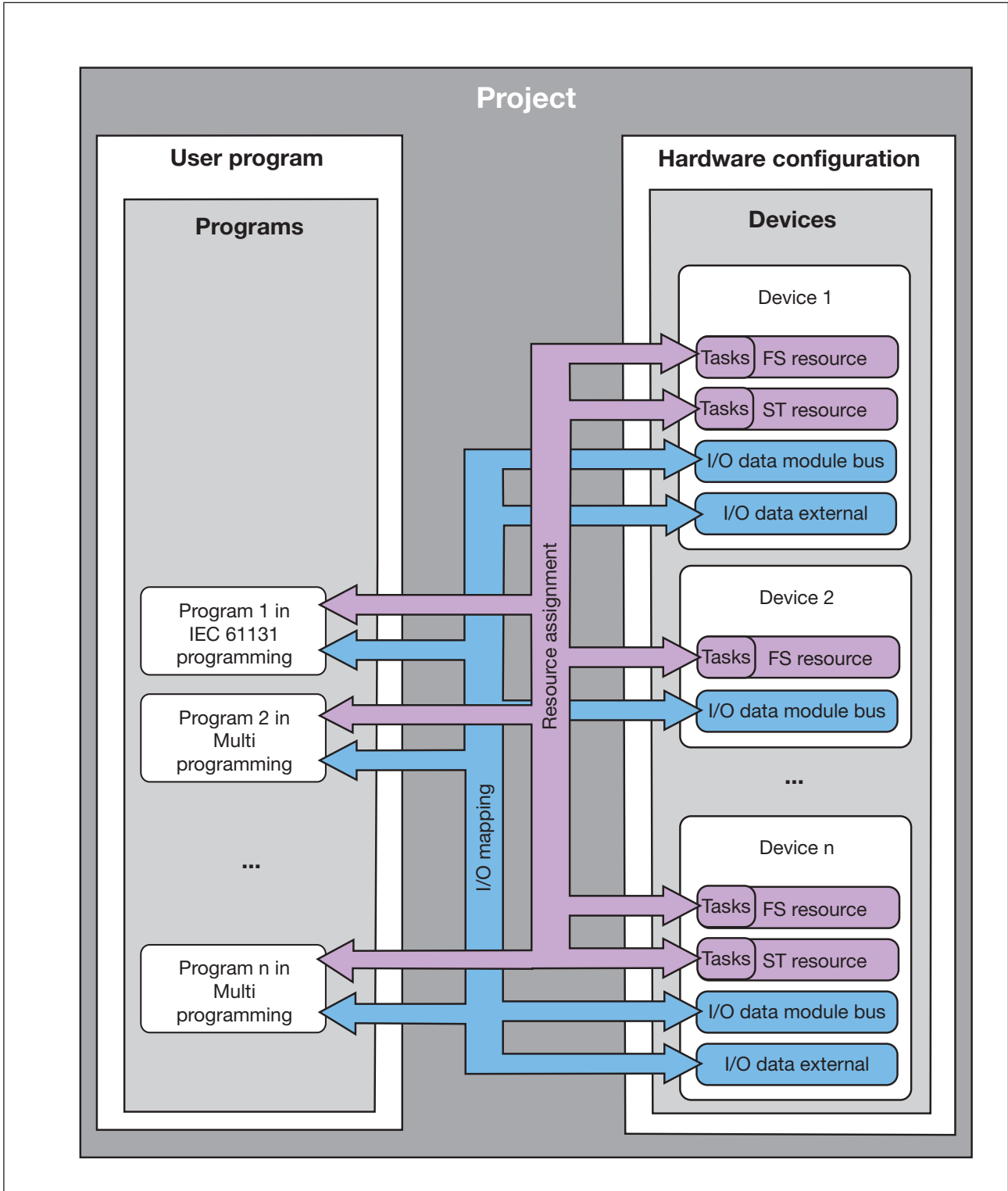
- Scanner Output Assembly Object instance - Attribute 3
Set Services are available for Attribute 3, provided it is not participating in a cyclical connection.
- Connection Manager Object – Attributes 1 ... 8

▶ **Reset Service**

If the Reset Service is called by the EtherNet/IP Scanner using Service Code 0x00 or 0x01, all the EtherNet/IP Adapter connections on the relevant PSS 4000 device will be reset. An entry is added to the diagnostic log and diagnostic list.

8 Project

8.1 Project structure



Project structure

To operate a plant with the automation system PSS 4000, a project must be created in PAS4000.

A project consists of the hardware configuration and the user program.

The hardware configuration defines the devices that will be used in the project and the settings with which the devices will be operated. The configuration is specified for each PSSu system, for example. The module bus I/O data, which is processed or provided by the PSSu system, is derived from the specification of the I/O modules. External communication can also be configured for PSS 4000 devices with external communication (e.g. PROFIBUS-DP-Slave). This produces the external I/O data, which is processed or provided by the device.

The user program consists of one or more programs. Programs for control systems PSSu PLC can be created in IEC 61131 programming and Multi programming. Programs for PSSu control systems may only be created in Multi programming.

▶ IEC 61131 Programming

When programming in accordance with IEC 61131, three types of program organisation units (POUs) are available for structuring the user program:

- Programs (PRG)
- Function blocks (FB)
- Functions (FUN)

The program forms the higher structural level. Functions and function blocks may be called up within a program.

Function blocks and functions undertake specific individual tasks within the user program.

The POUs can be formulated in the IL (Instruction List) or STL (Structured Text) programming language as required.

▶ Multi programming

Multi programming is performed according to the component model in the graphics Multi Editor. Three types of blocks are available for structuring the user program.

- Multi program

A Multi program is used exclusively to structure the user program. Multi blocks can be called up in a Multi program. Multi blocks are component blocks and basic blocks.

- Component block (CB)

Component blocks are used to combine Multi blocks. Several separate sub-solutions for automation functions can be combined in this way. Component blocks can be used to reproduce plant and machine structures.

Component blocks are self-contained units, which can be easily reused.

- Basic block (BB)

Basic blocks are used to implement any complex automation solutions.



Information

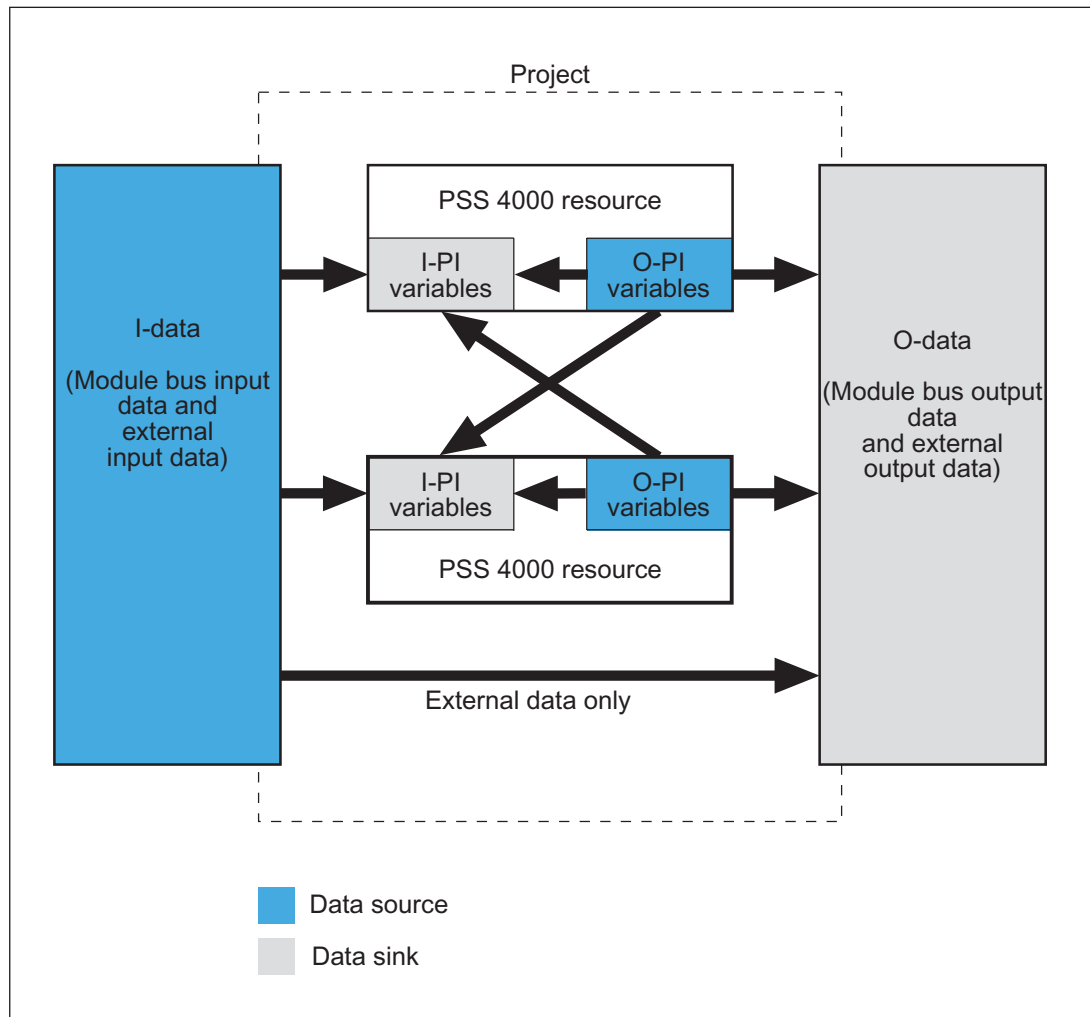
The term "block" is used as a generic term for Multi program, component block, basic block and program organisation unit of the type program, function block and function.

The connection between the blocks and the I/O hardware data is made via "PI variables". PI-variables can also be used for data exchange between blocks.

If no resource global variables are used, the user program can be created independently of the used hardware. Information on which section of the user program is to be executed on which resource under which task does not need to be defined in the resource assignment until programming is finished. The tasks determine the run-time properties of sections of the user program, i.e. the priority with which they are executed.

Once programming is finished, the connection between the PI variables and the automation system I/O data must be established. This is done in the I/O mapping.

8.2 Validity of process data



Data flow within the project

Process data includes the I/O data and PI variables. Inside a project, process data is exchanged between data sources and data sinks.

Data sources are:

- ▶ Module bus I-data
Input data that comes from the FS/ST module bus system section, e.g. from modules on the PSSu systems
- ▶ External I-data
Input data that comes from the system sections for external communication, e.g. from the PROFIBUS-DP Slave.
- ▶ O-PI variables
O-PI variables from PSS 4000 resources, e.g. from the FS resource or ST resource system sections; the resources may be on the same device or on different devices

Data sinks are:

- ▶ Module bus O-data
Output data that is written to the FS/ST module bus system section, i.e. is downloaded to the modules on the PSSu systems

- ▶ External O-data
Output data that goes out externally via system sections for external communication, e.g. via PROFIBUS-DP Slave.
- ▶ I-PI variables
I-PI variables from PSS 4000 resources, e.g. from the FS resource or ST resource system sections

The data flow is shown in the diagram using arrows. I/O mapping defines which data source supplies data to which data sink.

During the data flow, either the data source or the communication channel between the data source and data sink may malfunction, so that the data sink fails to receive a valid process value from the data source. To enable the data sink to establish whether the process value it has received is valid, each process value in the data sink has what's termed a "valid bit".

The valid bit is FALSE if the process datum in the data source is marked invalid or if it's impossible to access the process datum in the data source.

If the valid bit is FALSE, the process value is invalid and may not be used. If this is the case, the data sink has a substitute value which is used instead of the process value.

Situations which set the valid bit to FALSE:

- ▶ On module bus I-data
 - Error on an input (e.g. test pulse error)
 - Error on an I/O module (e.g. module overheating)
 - Error on the module bus
 - Module bus starting
- ▶ On external I-data
 - Communication is stopped
 - Communication timeout
- ▶ On O-PI variables
 - Task is in a STOP condition
 - Resource is in a STOP condition

Please note:

- ▶ Start-up behaviour

On start-up, situations may occur in which the user program in a device's resource is already being run, although data sources (e.g. the local module bus) have not yet supplied any valid data. In this case, the user program will operate with the substitute values (see Substitute values, [Page 205](#)). As soon as the system section with the data source is up and running, the current data will be processed rather than the substitute values. The valid bits can be evaluated to ensure that this signal change does not provoke any unwanted reaction (see Evaluation of the valid bits in the user program, [Page 206](#)).

A valid bit should be evaluated as a random sample from each system section from which data is processed in the user program (ST resource, FS resource, ST module bus, FS module bus, ST SafetyNET p RTFN, FS SafetyNET p RTFN, system section for external communication, e.g. PROFIBUS-DP Slave). For example, if an I-PI variable is mapped to an I-datum on the local FS module bus and the valid bit of that I-PI varia-

ble is TRUE, it's possible to conclude that the local FS module bus is running. At most some individual FS-I modules may be in a fault condition. If an I-PI variable is assigned to an I-datum on the ST module bus on another device and the valid bit of that I-PI variable is TRUE, it's possible to conclude that both the ST module bus on the other device and communication via ST-SafetyNET p RTFN are running. At most some individual ST-I modules on the other device may be in a fault condition.

▶ Online action: "Forcing"

When forcing an I-PI variable for which a valid bit is declared, the valid bit is automatically forced with the value TRUE.

Exceptions:

- The user changes the force value for the valid bit to FALSE.
- The user deactivates forcing for the valid bit.
- The I-PI variable belongs to a write and read-protected block or sealed block. In this case, it will not be possible to force the valid bit.

8.2.1 Substitute values

The substitute values for process data are fixed.

Substitute values can be:

- ▶ Zero (= safe condition)
- ▶ Last valid value

Process datum	Substitute value
I-PI variable on an FS resource	Zero
Output of an FS bit module	Zero
O-datum of an FS byte module	Zero
I-PI variable on an ST resource	Zero
Output from an ST bit module	Zero
O-datum from an ST byte module	Zero
ST-O datum PROFIBUS-DP Slave	Zero
ST-O datum Modbus/TCP Server connection	Last valid value
ST-O datum Modbus/TCP Client connection	Zero
ST-O datum Raw UDP	Zero
FS-O datum PROFIBUS-DP Slave with PROFIsafe	Zero

8.2.2 Evaluation of the valid bits in the user program

If a data source fails to supply a valid process datum, the valid bit in the data sink is set to FALSE and the data sink is assigned the substitute value. This process can be monitored by evaluating the valid bit.

The valid bit from an I-PI variable can be evaluated in the user program. It's not yet possible to monitor the valid bits from other data sinks.

If the I-PI variable is a multiple-element variable (e.g. a structure of the I/O data type ST_I_AI), the valid bit for the overall structure is identical to the valid bits of the structure's individual elements.

Evaluation of the valid bit in Multi programming

The PSS 4000 system block "VALID" is used to evaluate the valid bit.

Evaluation of the valid bit in IEC 61131 programming

The VALID extension must be declared for the I-PI variable in order to evaluate the valid bit.

8.3 Task basics

Tasks determine the run-time properties of basic blocks (Multi programming) and of program-type program organisation units (IEC 61131 programming), i.e. the conditions under which they are executed, how often and with what priority.

One or more basic blocks and/or program-type POUs are assigned to a task.

A resource may contain a maximum of 9 tasks. The number of tasks depends on the device. Details can be found in the device's operating manual.

The system organises task execution. The tasks' properties influence when the tasks are executed and in which order.



Information

For many projects it's not necessary to configure task properties. The default configuration is sufficient (see Default configuration of tasks, [Page 212](#)). So you don't necessarily have to deal with the issue of "Tasks".

8.3.1 Task properties

Task properties are defined in the task configuration:

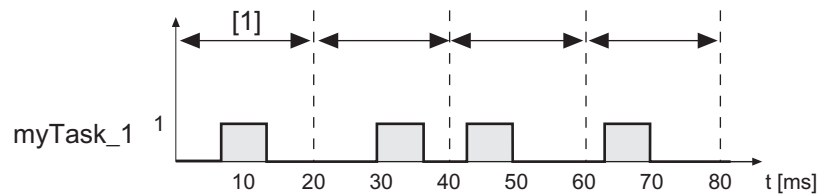
▶ Type

The task type "periodic task" is currently available.

The periodic task is executed once only within a definable task cycle time. The intervals between executions may vary.

If the execution is not completed within the task cycle time, the operating status "Resource in RUN condition with task fault" will result.

Example: Possible execution of periodic task "myTask_1" with a task cycle time of 20 ms



[1] Task cycle time

▶ Priority

The priority can be defined for each task. The priority determines the importance of a task in comparison with other tasks on a resource. As such it determines the order in which tasks on a resource are executed. The priorities are: high, medium and low. The number of times a priority may be assigned to a resource is specified. For example, a resource with 9 tasks may have 3 tasks with high priority, 3 with medium and 3 with low.

If a lot of tasks are programmed with high priority, it can slow down the execution of certain online functions in PAS4000 (e.g. variable watch, dynamic program display) as well as access from the PSS 4000 OPC Server to data on the devices.

▶ Task cycle time (only on periodic tasks)

The task cycle time defines the period within which the task must be executed.

The task cycle time also serves as a subpriority. Where tasks have the same priority, the task with the shortest task cycle time is executed first.

▶ Exclusivity

Exclusivity determines whether one or more basic blocks/program-type POU's can be assigned to a task.

8.3.2 Task execution

The PSS 4000 is a multi-tasking system with pre-emptive scheduling. Up to 9 tasks may be available on one resource. The number of tasks depends on the device. Details can be found in the device's operating manual.

The system organises task execution. The tasks' properties influence when the tasks are executed and in which order. The execution of lower priority tasks is interrupted if higher priority tasks are pending execution.

The following rules apply when executing tasks:

- ▶ Execution controlled per device

Each device within the project independently controls task execution. A device can have a maximum of one FS resource and one ST resource. Each resource may contain a maximum of 9 tasks. The number of tasks depends on the device. Details can be found in the device's operating manual.

The number of times a priority may be assigned to a resource is specified. For example, a resource with 9 tasks may have 3 tasks with high priority, 3 with medium and 3 with low.

- ▶ Order by priority

Tasks are executed in accordance with their priority. High priority tasks are executed first, followed by medium and then low priority tasks. Where tasks have equal priority, the task cycle time decides. The task with the shortest task cycle time is executed first. If there are tasks with equal priority and task cycle time, the order is random.

- ▶ Interaction between FS and ST resource

If a device has an FS and an ST resource, both are executed by the same CPU. The following order applies:

1. FS tasks, high priority
2. ST tasks, high priority
3. FS tasks, medium priority
4. ST tasks, medium priority
5. FS tasks, low priority
6. ST tasks, low priority

Execution of the FS task with high priority always has precedence.

- ▶ Periodic check for pending tasks

With each system clock pulse the system checks whether there are any tasks pending execution and redefines the order of execution. If a task is being executed, it will be interrupted if a higher priority task is pending.

- ▶ Execution order

If several programs have to be run within a task, the programs are executed in alphabetical order, based on the program names.

Multi programming: The execution order for the basic blocks within a Multi program is defined by PAS4000.

- ▶ Reading and outputting the process image and resource global variables

When task execution begins, a copy is created of the section of the process image (I-PI and O-PI) needed by the blocks (basic blocks/program-type POU) in the task. The necessary resource global variables are also copied.

When the task is executed, only the values from the copies are read and written.

Not until the task is complete, in other words, when all the task's blocks have been executed, is the data from the copies transferred to the resource global variables and the process image of outputs (O-PI).

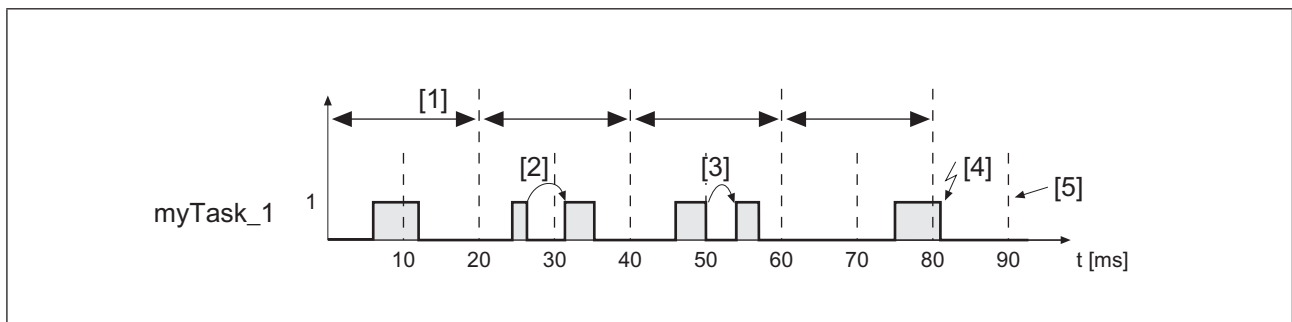
By using this procedure, data cannot be modified externally while the task is being executed. Even if a task is interrupted during execution, it will subsequently continue working with the data in its copies.

► Task interruption

A higher priority task is not the only event that can interrupt a task. A task will also be interrupted, for example, if a block is waiting for a system function to be executed. As a result, when a block is executed, it is not certain whether a block has been executed previously in a higher priority task and whether its current data is available in the PI. If one task is to process data from another task, the user must ensure that the necessary data from the other task has already been executed, by evaluating a counter for example.

Example 1

Execution of periodic task "myTask_1", with a task cycle time of 20 ms and a system clock pulse of 10 ms



[1] Task cycle time

[2], [3] Interruption

[4] Error

[5] System clock pulse

With each system clock pulse the system checks whether there are any tasks pending execution. The task "myTask_1" is available for execution every 20 ms. However, it is not always executed immediately because higher priority tasks are executed first. The task may also be interrupted because the program is waiting for a system function to be executed for example [2], or because a higher priority task is present on the system clock pulse [3]. If the task execution is not completed within the task cycle time, an error is triggered [4].

Example 2

A medium priority task is being executed. A high priority task is present on the system clock pulse. The first task is interrupted and the high priority task is executed. Execution of the first task is then resumed.

Example 3

A high priority task with a task cycle time of 20 ms is being executed. A high priority task with a task cycle time of 10 ms is present on the system clock pulse. The first task is interrupted and the task with the shorter task cycle time is executed.

Example 4

A medium priority ST task is being executed. A medium priority FS task is present on the system clock pulse. The ST task is interrupted and the FS task is executed first.

8.3.3 Default configuration of tasks

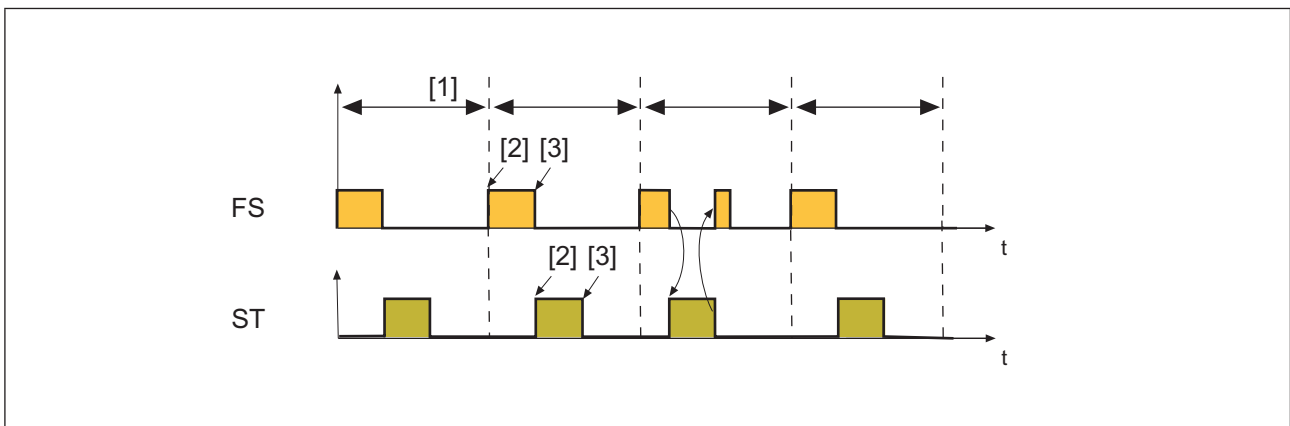
For many projects it's not necessary to configure task properties. The default configuration is sufficient.

The default configuration is designed so that the FS task is executed first on a device, followed by the ST task. The circumstance may arise where an FS block, for example, is waiting for a system function to be executed. In this case, execution of the FS task is interrupted and the ST task is executed.

If several programs have to be run within a task, the programs are executed in alphabetical order, based on the program names.

Multi programming: The execution order for the basic blocks within a Multi program is defined by PAS4000.

The FS task and ST task are executed once within the task cycle time of 100 ms.



[1] Task cycle time

[2] Resource global variables and process image of inputs (I-PI) and outputs (O-PI) are read in

[3] Output of resource global variables and process image of outputs (O-PI)

8.4 Data exchange between FS resource and ST resource

All the PI variables declared in the user program must be mapped to an I/O datum or another PI variable in the I/O Mapping Editor. It is also possible to map I-data directly to O-data.

The overview shows which I/O mappings are permitted.

Module bus data is data from the PSSu modules (e.g. hardware inputs/hardware outputs). External data is data from external communication (e.g. Modbus/TCP, fieldbuses); this does not include data transmitted via SafetyNET p.

I/O mapping		To (data sink)					
		Module bus ST-O datum	Module bus FS-O datum	ST-I-PI variable	FS-I-PI variable	External ST-O datum	External FS-O datum
From (data source)	Module bus ST-I datum	---	---	◆	---	◆	---
	Module bus FS-I datum	---	---	+	◆	+	◆
	ST-O-PI variable	◆	+	◆	---	◆	---
	FS-O-PI variable	+	◆	+	◆	+	◆
	External ST-I datum	◆	+	◆	---	◆	---
	External FS-I datum	+	◆	+	◆	+	◆

Legend:

- Not permitted (prevented by PAS4000)
 - ◆ Permitted
 - +
- Permitted, but the FS datum may only be used for **non-safety-related** tasks.

Please note the following when I/O mapping:

▶ **I/O mappings between FS and ST**

Even if ST output data (e.g. ST hardware outputs, ST-O data for external communication) is controlled via the FS resource, this output data may only be used for **non-safety-related** tasks. PAS4000 will warn you of such I/O mappings. This will include, for example, the I/O mapping of:

- Module bus FS-I datum to external ST-O datum
- FS-O-PI variable to module bus ST-O datum

With this procedure, appropriate measures must be put in place to verify safety, e.g. by means of

- A separate risk analysis and
- Appropriate commissioning tests, feasibility tests

The user is responsible for the correct implementation.

▶ **I/O mapping between PI variable and PI variable in the same task**

After writing the O-PI variables, the value will not be available to read in to the mapped I-PI variable until the start of the next task cycle.

8.5 Data consistency

With PSS 4000, data consistency is generally understood as keeping together data that is recorded at a certain time and is considered current and related during a defined period.

Data consistency of a data source

In terms of data consistency, data sources (see Validity of process data, [Page 202](#)) are divided into "consistency ranges". A data source's consistency range contains an amount of consistent data. This means that the process data on a data source is consistent if it comes from the same consistency range on the data source.

Data consistency on system-internal data transfer via SafetyNET p

When process data in a consistency range is transmitted system-internally (e. g. PSSu system <-> SafetyNET p <-> PSSu system), it is consistent provided it is transmitted using a single SafetyNET p connection.

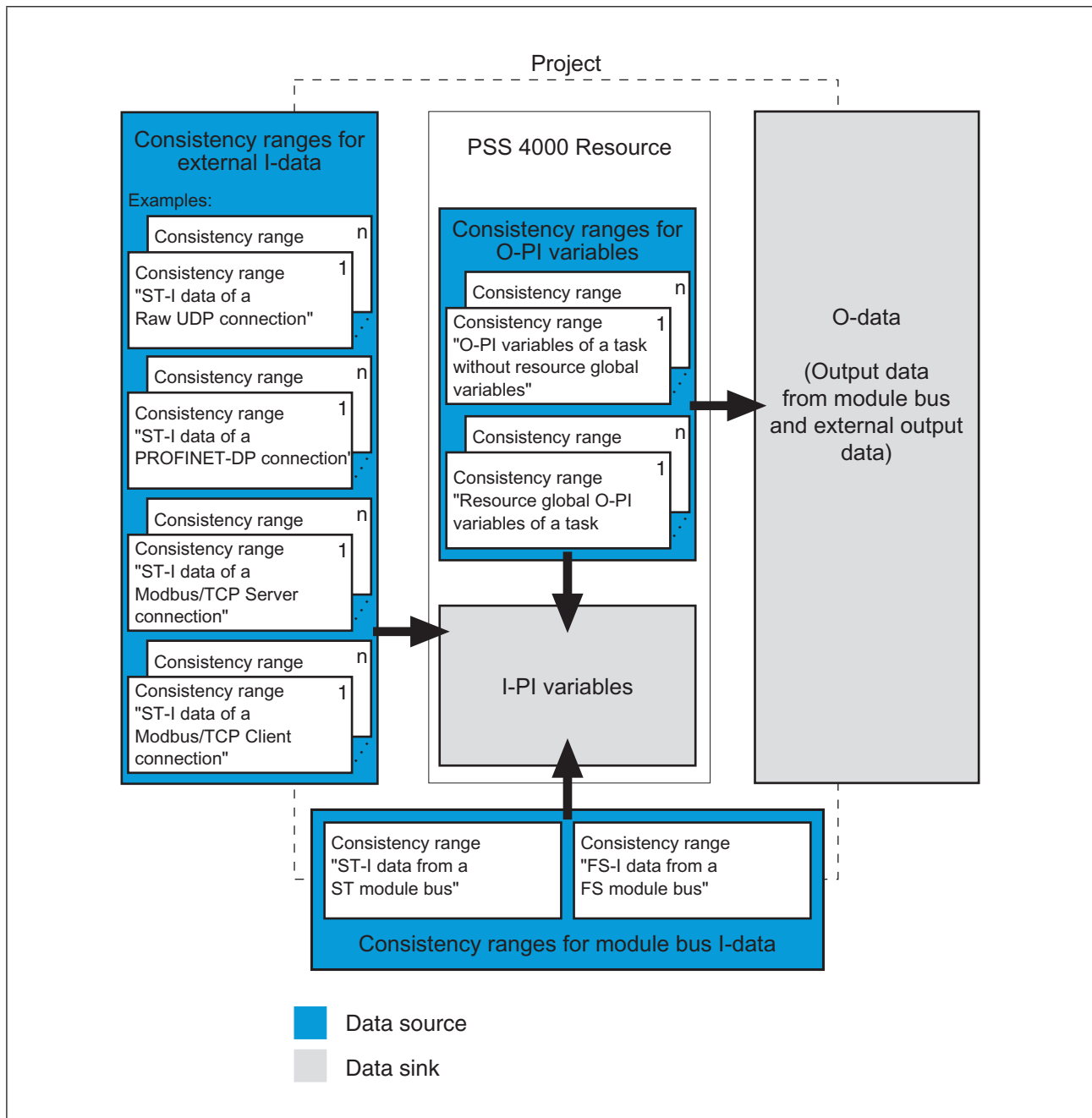
If the documented [SafetyNET p limits](#) are exceeded during system-internal data transfer, the data from the consistency range is automatically split over several connections. In this case, data consistency is no longer guaranteed.

During the build process, PAS4000 issues a warning if the process data from a consistency range is split over several connections.

Data consistency of a data sink

A data sink's process data (see Validity of process data, [Page 202](#)) is consistent if it comes from the same consistency range on a data source. No statement can be made regarding the relationship between a data sink's process data when it comes from various consistency ranges.

8.5.1 Consistency ranges



Overview of a project's consistency ranges

A project may have the following consistency ranges:

- ▶ **Module bus I-data**
 - Consistency range "FS-I data of an FS module bus"
 - Consistency range "ST-I data of an ST module bus"
- ▶ **External I-data**
 - Consistency range "ST-I data of an external communication connection"
 - Consistency range "FS-I data of an external communication connection"

No statement can be made regarding the data consistency of data that is transmitted by external devices to a PSS 4000 device with corresponding system section via a communication connection. It depends on the communication mode (e.g. Modbus/TCP, PROFIBUS-DP, Raw UDP) and the options available with the communication mode. As soon as the data is available as external I-data in PSS 4000, it is treated as consistent data.

There are consistency ranges for the external I-data of any communication mode or connection type. Each consistency range for external I-data behaves identically in terms of the data consistency of its data. In most cases, therefore, only the general consistency range for external I-data is stated (consistency range "ST-I data of an external communication connection" or consistency range "FS-I data of an external communication connection"). If necessary, however, the consistency range for each communication mode or each connection type can be stated separately.

Examples for the consistency range "ST-I data of an external communication connection":

- Consistency range "ST-I data of a Modbus/TCP Client connection"
- Consistency range "ST-I data of a Modbus/TCP Server connection"
- Consistency range "ST-I data of a PROFIBUS-DP connection"
- Consistency range "ST-I data of a Raw UDP connection"

Examples for the consistency range "FS-I data of an external communication connection"

- Consistency range "FS-I data of a virtual PROFIsafe F-module"

▶ **O-PI variables**

- Consistency range "O-PI variables of a task without resource global variables"

Such a consistency range includes all O-PI variables of blocks that are executed in the same task.

For this type of consistency range, no statement can be made regarding the relationship to resource global O-PI variables that are also executed in this task.

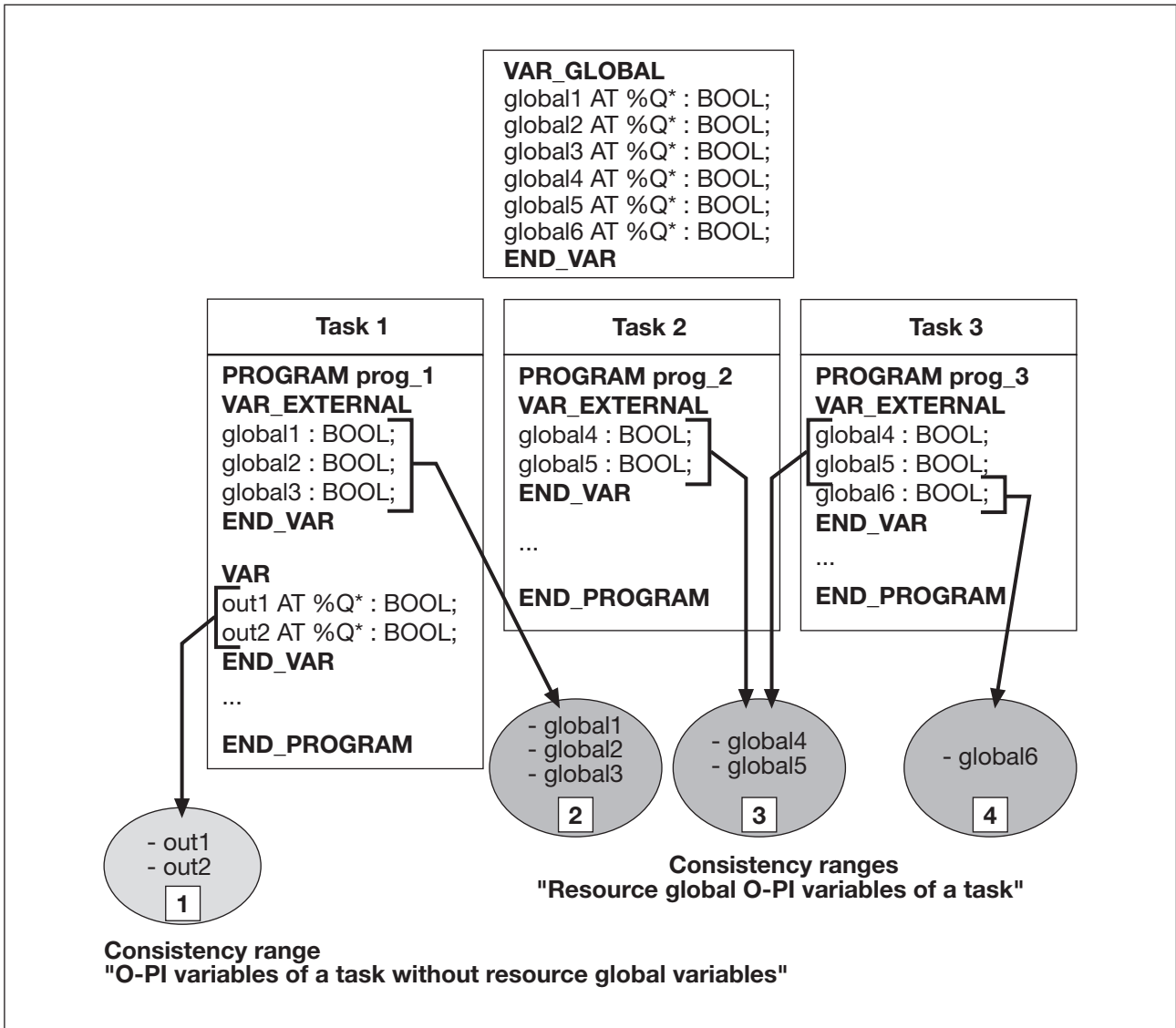
- Consistency range "Resource global O-PI variables of a task"

No statement can be made regarding the relationship of resource global variables to a consistency range "O-PI variables of a task without resource global variables". This is also the case if they are executed in the same task (see "Task 1" in the example below).

The following applies for data consistency of resource global O-PI variables:

- Resource global O-PI variables that are accessed exclusively from blocks that are executed in the same task are consistent and form a consistency range (see "Task 1" and consistency ranges [1] and [2] in the example below).
- The intersection of resource global O-PI variables that are accessed from blocks that are executed in different tasks is consistent and forms a consistency range (see "Task 2", "Task 3" and consistency range [3] in the example below).
- No statement can be made regarding the data consistency of variables in a consistency range which are mapped to an I-PI variable as well as a resource global I-PI variable and which are accessed from blocks that are executed in the same task.

Example:



Example for consistency ranges of O-PI variables

8.5.2 Influence of scheduling on data consistency

Scheduling regulates the order in which data in a project is processed and also ensures optimum data flow, taking into account the various cycle times, execution times and processing times (see Reaction times, [Page 370](#)).

In PAS4000, if configurable cycle times (e.g. cycle time with external communication) are calculated automatically by the system, the system will determine the optimum conditions and reaction times for the project. Data consistency in the consistency ranges is guaranteed.

If configurable cycle times are configured by the user, the system will determine the optimum reaction times for the configured conditions. These may deteriorate in comparison with the reaction times that are determined automatically. Data consistency in the consistency ranges is guaranteed.

8.6 Safety status of blocks

In IEC 61131 programming it is possible to program function block-type and function-type POU, which have a defined safety state. In Multi programming, blocks cannot be programmed with a safety state. However, in Multi programming the POU, programmed in IEC 61131 are used as basic blocks (IEC 61131 basic blocks). These basic blocks have a safety state.

Non-safety-related tasks may be programmed in ST blocks. ST blocks may be executed on either an ST or an FS resource. ST blocks are shown in green in PAS4000.

Safety-related tasks must be programmed in FS blocks. FS blocks may only be executed on an FS resource. FS blocks are shown in yellow in PAS4000.

In IEC 61131 programming, a POU becomes an FS-POU as soon as an FS data type variable is declared or an FS-POU is called/instantiated.

If it is possible to access non-safety-related data (ST data) within an FS block, the block becomes an "FS block with mixed input interface". They are given a special marking in PAS4000:

- ▶ Multi programming: half green and half yellow
- ▶ IEC 61131 programming: half white and half yellow

This special marking is intended to identify a block as potentially hazardous.

In IEC 61131 programming, an FS-POU will become an FS-POU with mixed input interface in the following circumstances:

- ▶ Declaration of an I-PI variable that is **not** an FS data type
- ▶ Declaration of a variable of type VAR_INPUT, VAR_IN_OUT or VAR_EXTERNAL, that is **not** an FS data type
- ▶ Call/instantiation of a POU in which an I-PI variable or a VAR_EXTERNAL type variable is declared, which is **not** an FS data type

When using FS blocks with mixed input interface, please note:

- ▶ Data on the ST resource is **not** secure. It may only be an **additional** criterion for a safety-related action (e.g. start/stop) and must **not** prevent safety functions. For example, if for whatever reason a stop signal has not been activated in the ST resource, or a start signal is continuously activated, this must not be able to compromise the safety of the machine or process.
- ▶ Appropriate procedures must be put in place to verify safety, e.g. by means of
 - A separate risk analysis and
 - Appropriate commissioning tests, feasibility tests
- ▶ We recommend that you test an FS-POU with mixed interface fully in the first instance, if necessary have it tested and certified by a notified body (e.g. BG, TÜV) and then use PAS4000 to seal it.
- ▶ The user is responsible for the correct implementation.

**CAUTION!**

When sealed, FS-POUs with mixed input interface lose their label as potentially hazardous (half white and half yellow) and are shown in yellow. It is the responsibility of the manufacturer or notified body to ensure that access to non-safety-related data (ST data) within the block is not safety-critical.

8.7 Device naming via PAS4000

Device naming is performed by the user in PAS4000. The data required to address a device uniquely within the project is defined for each device in the project. This data is called [naming data](#).

To enable the naming data to be transmitted to the devices within the project during device naming, they must be identifiable on the network. This is achieved using fixed data stored in each device as a factory setting. This data is called [device identification data](#).

During device naming, a distinction is made between the following device types:

- ▶ [PSS 4000-exclusive devices \(e.g. PSSu systems\)](#)
These support all device identification data and all naming data.
- ▶ [PSS 4000-operable devices from Pilz \(e.g. operator terminal PMI\)](#)
These support all device identification data and all naming data.
- ▶ [PSS 4000-operable devices from a third-party manufacturer \(e.g. PC\)](#)
These support part of the device identification data and naming data.



Information

If networks are connected via Routers, device naming can only be performed centrally if the system variable `PILZ_SNP_MULTICAST_TTL` is configured accordingly (see Network with Routers, [Page 98](#)).

8.7.1 Device identification data

PSS 4000-exclusive devices (e.g. PSSu systems)

Device identification data	Key	Example
Product type	Kind of device (e.g. head module of a PSSu system)	PSSu H PLC1 FS SN SD
Order number	6-digit number by which the device can be ordered at Pilz (e.g. head module of a PSSu system)	312070
Serial number	6-digit number by which the device can be identified at Pilz (e.g. head module of a PSSu system)	100076

PSS 4000-operable devices from Pilz (e.g. operator terminals PMI)

Device identification data	Meaning	Example
Product type	Kind of device	PMI o438
Order number	6-digit number by which the device can be ordered at Pilz	264260
Serial number	6-digit number by which the device can be identified at Pilz	100057

PSS 4000-operable devices from a third-party manufacturer (e.g. PC)

Device identification data	Meaning
Product type	Kind of device: PSS 4000-PC
Order number	"0" is used as the order number
Serial number	When the PSS 4000 firmware is first started, a random 6-digit number is generated, which is then used internally as a serial number for various identification processes

8.7.2 Naming data

PSS 4000-exclusive devices (e.g. PSSu systems)

Naming data	Default setting	Meaning
Device name	Combination of a prefix (e.g. PSS), 6-digit order number and 6-digit serial number Format: <Prefix>_<Order number>_<Serial number> Example: PSS_312070_100076	<ul style="list-style-type: none"> ▶ Uniquely identifies a device within the project ▶ Used to identify the device on the network when the project is downloaded ▶ Forms part of an address (e.g. I/O mapping)
IP address	IP address in accordance with the Ethernet convention (see Factory default settings for the Ethernet parameters, Page 491)	Used in conjunction with the serial number to identify the device on the network during device naming
Subnet mask	Settings in accordance with the Ethernet convention (see Factory default settings for the Ethernet parameters, Page 491)	In accordance with the Ethernet convention
Gateway address	Settings in accordance with the Ethernet convention	<ul style="list-style-type: none"> ▶ In accordance with the Ethernet convention ▶ Available as an option
Address of preferred DNS Server	(see Factory default settings for the Ethernet parameters, Page 491)	
Address of alternative DNS Server		

PSS 4000-operable devices from Pilz (e.g. operator terminals PMI)

Naming data	Default setting	Meaning
Device name	Combination of a prefix (e.g. PMI) and at least the device generation Examples: PMI4visu, PMI4open	<ul style="list-style-type: none"> ▶ Uniquely identifies a device within the project ▶ Used to identify the device on the network when the project is downloaded ▶ If the user has assigned a device name at operating system level, this device name will be displayed (e.g. myPMI_1)
IP address	IP address in accordance with the Ethernet convention (see Auto-IP mechanism, Page 73)	Used in conjunction with the serial number to identify the device on the network during device naming
Subnet mask	Settings in accordance with the Ethernet convention	In accordance with the Ethernet convention

Naming data	Default setting	Meaning
Gateway address	Settings in accordance with the Ethernet convention	<ul style="list-style-type: none"> ▶ In accordance with the Ethernet convention ▶ Available as an option
Address of preferred DNS Server		
Address of alternative DNS Server		

PSS 4000-operable devices from a third-party manufacturer (e.g. PC)

Naming data	Meaning, content and default setting
Device name	<ul style="list-style-type: none"> ▶ Uniquely identifies a device within the project ▶ Used to identify the device on the network when the project is downloaded <p>On a PC, the default device name corresponds to the computer name defined by the user when setting up the Windows operating system.</p>
IP address	<p>When the device is named, used in conjunction with the generated serial number to identify the device on the network</p> <p>IP address in accordance with the Ethernet convention</p>
Subnet mask	Subnet mask in accordance with the Ethernet convention
Gateway address	Available as an option, depending on the device
Address of preferred DNS Server	Available as an option, depending on the device
Address of alternative DNS Server	

8.7.3 Device naming for PSS 4000-exclusive devices: PSSu systems

When device naming for PSS 4000-exclusive devices (e.g. PSSu systems), a distinction can be made between the following cases:

▶ **PSS 4000-exclusive device in its delivery condition or after an original reset**

The factory-set defaults are stored on the device that is in its delivery condition or has undergone an original reset (see Naming data, [Page 223](#)). The Auto-IP mechanism is activated (see Auto-IP mechanism, [Page 73](#)). No device project is present.

▶ **Changing the device name**

The device name of a PSS 4000-exclusive device which has already been through the naming process can be changed retrospectively. In this case, user-specific naming data is already present on the device. A device project may be present. However, it will be deleted during device naming.

Whether or not the Ethernet parameters are also changed is irrelevant for the device naming procedure.

▶ **Changing the Ethernet parameters**

The Ethernet parameters of a PSS 4000-exclusive device which has already been through the naming process can be changed retrospectively. In this case, user-specific naming data is already present on the device. A device project may be present. This will be retained during device naming.

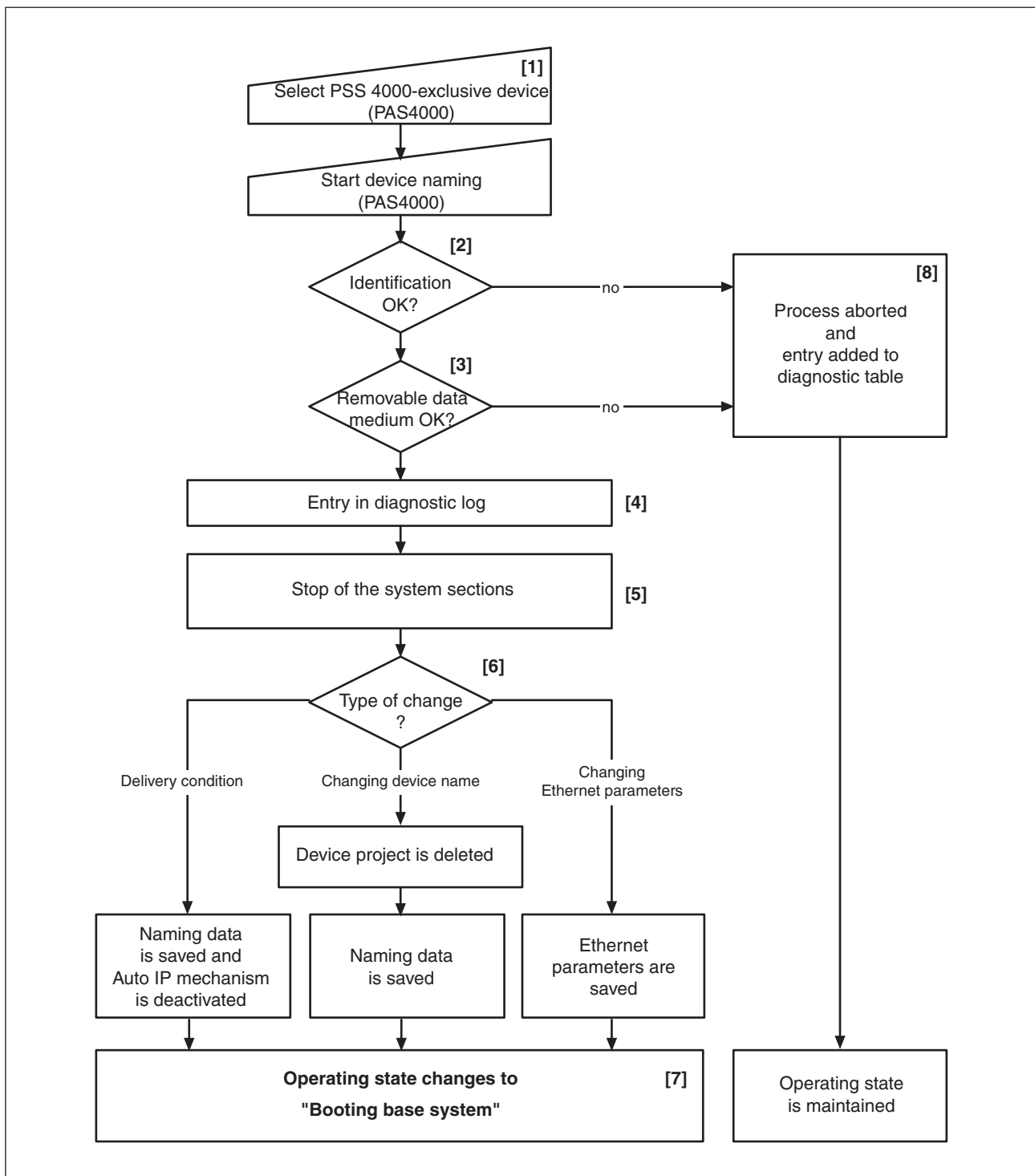
Whether just one Ethernet parameter is changed or several is irrelevant for the device naming procedure. It is usually only the IP address that is changed. However, if the device name is also changed, the procedure will be as described under "Changing the device name".



Information

The description below assumes that all preparatory measures have already been carried out. These include:

- A removable data medium is connected to the PSS 4000-exclusive device.
- The relevant naming data has been edited in PAS4000.



Flowchart for device naming on PSS 4000-exclusive devices

Device naming procedure on a PSS 4000-exclusive device

The following description refers to the flowchart, Fig. "Flowchart for device naming on PSS 4000-exclusive devices".

[1]

In PAS4000, the PSS 4000-exclusive device on which device naming is to be performed is selected. A test can be carried out with PAS4000 to identify the selected device in the field.

The test is called "Device Identification". During device identification, the LED on the removable data medium will be activated on the selected device. The status LED (e.g. "SD CARD" LED on a PSSu system) flashes orange.

[2]

Once device naming has started, the device checks that the local serial number and IP address match the serial number and IP address stated as the destination for transferring naming data in PAS4000.

[3]

The device checks whether the removable data medium is accessible, i.e. that the removable data medium is inserted, is not write protected and is not defective.

[4]

An entry in the diagnostic log shows that device naming has been carried out.

[5]

The device's system sections are stopped. On a PSSu system this corresponds to the operating status: "Device in STOP condition without error".

[6]

The rest of the procedure depends on which naming data has been changed:

▶ **PSS 4000-exclusive devices in their delivery condition or after an original reset**

The factory-set default naming data in the device memory and on the removable data medium are replaced by the configured naming data. The Auto-IP mechanism is deactivated.

▶ **Changing the device name**

The existing naming data is deleted from the device memory and removable data medium. If the removable data medium already contains a device project, the device project will be deleted from the removable data medium. On devices with an FS resource, the check sum of the FS device project and the project name will also be deleted from the device memory. The new naming data is then stored on the removable data medium.

▶ **Changing the Ethernet parameters**

The existing naming data and any existing device project are retained. The IP address and all other Ethernet parameters are deleted from the device memory and removable data medium. The new IP address and all other Ethernet parameters are then stored on the removable data medium.

[7]

The PSS 4000-exclusive device automatically switches to the [operating status "Booting base system"](#). After this boot process, the device switches to the operating status "Booting system sections".

The rest of the procedure depends on which naming data has been changed:

▶ **PSS 4000-exclusive devices in their delivery condition and Changing the device name**

During the operating status "Booting system sections", one of the effects is that the new naming data is loaded from the removable data medium to the device memory. The device starts up without a device project. The status LED for the removable data medium (e.g. "SD CARD" LED on a PSSu system) flashes green.

For the device to obtain a device project, a project will need to be downloaded (see Project download, [Page 238](#)).

▶ **Changing the Ethernet parameters**

During the operating status "Booting system sections", one of the effects is that the new Ethernet parameters are loaded from the removable data medium to the device memory. As soon as a device project is available, the device is started up with the device project.

If other devices in the network exchange data with the PSS 4000-exclusive device, a new project will need to be downloaded once the IP address and/or other Ethernet parameters have been changed. If not, the other devices will no longer be able to access the PSS 4000-exclusive device.

[8]

If any of the checks is negative, the process is aborted. An entry will be added to the diagnostic list and the operating status of the PSS 4000-exclusive device remains unchanged.

8.7.3.1 Error behaviour during device naming

If any error should occur during device naming, the following error behaviour applies generally:

- ▶ Device naming process is aborted
- ▶ All system sections of the PSS 4000-exclusive device switch to a safe condition
- ▶ All FS outputs on the PSS 4000-exclusive device switch to a safe condition.
- ▶ An entry is added to the diagnostic list.

A cold start/reboot command must be executed for the PSS 4000-exclusive device. In this case, the PSS 4000-exclusive device will start up with the "old" naming data and, if available, the "old" device project.

8.7.4 Device naming for PSS 4000-operable devices from Pilz

When device naming for PSS 4000-operable devices from Pilz (e.g. operator terminals PMI), a distinction can be made between the following cases:

- ▶ **PSS 4000-operable device from Pilz during initial device naming**

After the PSS 4000 firmware has been installed, the factory set defaults are stored (see Naming data, [Page 223](#)). No device project is present.

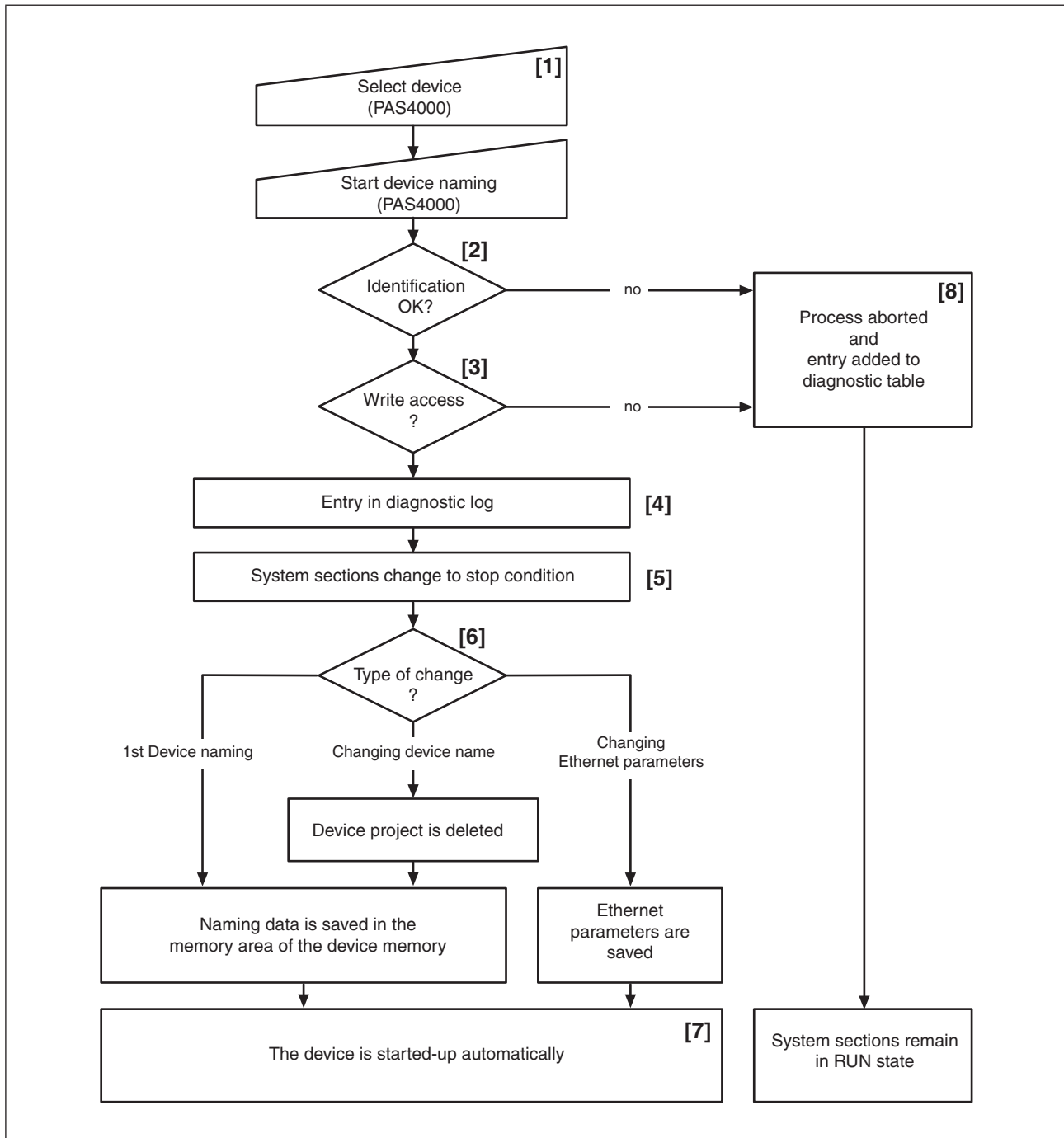
- ▶ **Changing the device name**

The device name of a PSS 4000-operable device from Pilz which has already been through the device naming process can be changed retrospectively. In this case, user-specific naming data is already present on the device. A device project may be present. However, it will be deleted during device naming. Whether or not the Ethernet parameters are also changed is irrelevant for the device naming procedure.

- ▶ **Changing the Ethernet parameters**

The Ethernet parameters of a PSS 4000-operable device from Pilz which has already been through the naming process can be changed retrospectively. In this case, user-specific naming data is already present on the device. A device project may be present. This will be retained during device naming.

Whether just one Ethernet parameter is changed or several is irrelevant for the device naming procedure. It is usually only the IP address that is changed. However, if the device name is also changed, the procedure will be as described under "Changing the device name".



Flowchart for device naming on PSS 4000-operable devices from Pilz

Device naming procedure on PSS 4000-operable devices from Pilz

The following description refers to the flowchart, Fig. "Flowchart for device naming on PSS 4000-operable devices from Pilz".

[1]

In PAS4000, the PSS 4000-operable device from Pilz on which device naming is to be performed is selected. A test can be carried out with PAS4000 to identify the selected device. The test is called "Device Identification". During device identification, a pop-up window with

the following content is opened on the selected device:

PSS 4000 Device name: <Device name>.

The pop up window does not disappear until device identification is explicitly ended.

[2]

Once device naming has started, the device checks that the local serial number and IP address match the serial number and IP address stated as the destination for transferring naming data in PAS4000.

[3]

A test is carried out to check whether "write" permission is available for the relevant memory area in the device's security settings.

[4]

An entry in the diagnostic log shows that device naming has been carried out.

[5]

The device stops the system sections of the PSS 4000 firmware (e.g. OPC Server, Diagnostic Server).

[6]

The rest of the procedure depends on which naming data has been changed:

▶ **PSS 4000-operable device from Pilz during initial device naming**

The factory-set naming data is deleted from the relevant memory area of the device. The new naming data is then stored in the memory area. A project must then be downloaded, so that the device obtains a device project.

▶ **Changing the device name**

The existing naming data and any existing device project are deleted from the relevant memory area of the device. The new naming data is then stored in the memory area. A project must then be downloaded, so that the device obtains a device project.

▶ **Changing the Ethernet parameters**

The existing naming data and any existing device project are retained. The IP address and all other Ethernet parameters are deleted from the relevant memory area of the device. The new IP address and all other Ethernet parameters are then stored in the memory area.

[7]

The device is automatically restarted so that the naming data can be adopted.

[8]

If any of the checks is negative, the process is aborted. An entry will be added to the diagnostic list and the system sections of PSS 4000 (e.g. OPC Server, Diagnostic Server) remain in RUN condition.



CAUTION!

If the IP address of a PSS 4000-operable device is changed, other devices that do not belong to the PSS 4000 project will no longer be able to access the PSS 4000-operable device via any existing Ethernet communication relationships. The IP address for these Ethernet communication relationships must be adapted in the relevant devices.

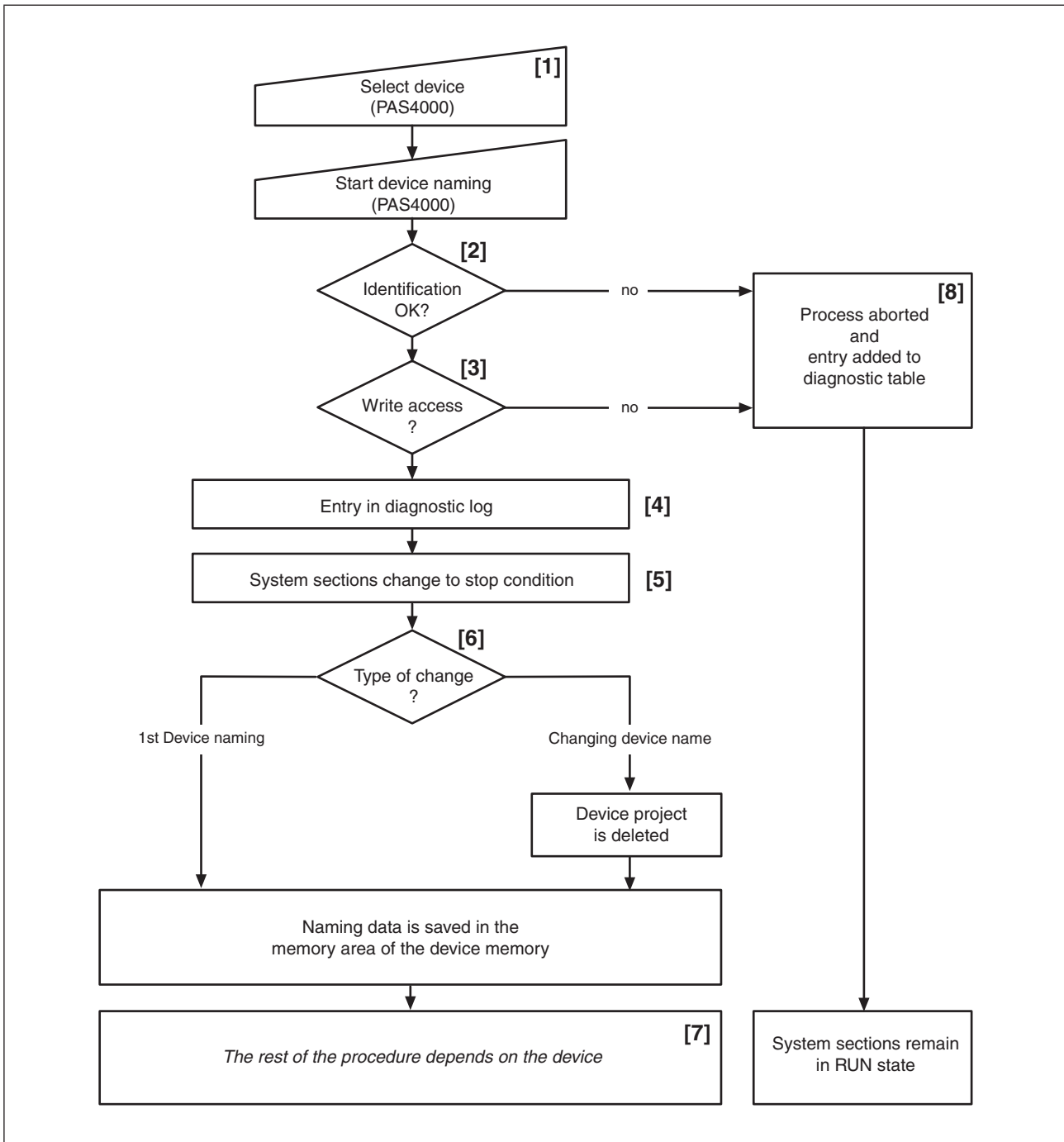
8.7.5 Device naming for PSS 4000-operable devices from a third-party manufacturer

When device naming for PSS 4000-operable devices from a third-party manufacturer, please note the following:

- ▶ **Device name**
A device name edited during device naming in PAS4000 will appear only in PAS4000, in the diagnostic list and in the diagnostic log. A device name assigned at operating system level (computer name), will not be overwritten during device naming.
- ▶ **IP address**
The IP address at the device's operating system level cannot be changed during device naming. This must be done at operating system level, for PCs with Microsoft Windows operating system for example.

When device naming for PSS 4000-operable devices from a third-party manufacturer, a distinction can be made between the following cases:

- ▶ **PSS 4000-operable device from a third-party manufacturer during initial device naming**
After the PSS 4000 firmware has been installed, the factory set naming data is stored (see Naming data, [Page 223](#)). No device project is present.
- ▶ **Changing the device name**
When the device name is changed retrospectively, user-specific naming data will already be present on the PSS 4000-operable device. A device project may be present. However, it will be deleted during device naming.



Flowchart for device naming on PSS 4000-operable devices from a third-party manufacturer

Device naming procedure on PSS 4000-operable devices from a third-party manufacturer

The following description refers to the flowchart, Fig. "Flowchart for device naming on PSS 4000-operable devices from a third-party manufacturer".

[1]

In PAS4000, the PSS 4000-operable device from a third-party manufacturer on which device naming is to be performed is selected. A test can be carried out with PAS4000 to identify the selected device. The test is called "Device Identification". During device identifica-

tion, a pop-up window with the following content is opened on the selected device:

PSS 4000 Device name: <Device name>.

The pop up window does not disappear until device identification is explicitly ended.

[2]

Once device naming has started, the device checks that the local serial number and IP address match the serial number and IP address stated as the destination for transferring naming data in PAS4000.

[3]

A test is carried out to check whether "write" permission is available for the relevant memory area in the device's security settings.

[4]

An entry in the diagnostic log shows that device naming has been carried out.

[5]

The device stops the system sections of the PSS 4000 firmware (e.g. OPC Server, Diagnostic Server).

[6]

The rest of the procedure depends on which naming data has been changed:

▶ **PSS 4000-operable device from a third-party manufacturer during initial device naming**

The factory-set naming data is deleted from the relevant memory area of the device.

The new naming data is then stored in the memory area. A project must then be downloaded, so that the device obtains a device project.

▶ **Changing the device name**

The existing naming data and any existing device project are deleted from the relevant memory area of the device. The new naming data is then stored in the memory area. A project must then be downloaded, so that the device obtains a device project.

[7]

The rest of the procedure depends on the device. If the device is a PC, the PSS 4000 firmware on the PC must be manually restarted after device naming so that the naming data can be adopted.

[8]

If any of the checks is negative, the process is aborted. An entry will be added to the diagnostic list and the system sections of PSS 4000 firmware (e.g. OPC Server, Diagnostic Server) remain in RUN condition.

**CAUTION!**

Please note the following special features for the IP address of PSS 4000-operable devices from a third-party manufacturer:

- The IP address must be configured during the hardware configuration in PAS4000, to enable the other devices in the PSS 4000 project to communicate with this device.
- The IP addresses in the hardware configuration, device naming and at operating system level must match. During device naming with PSS 4000-operable devices from a third-party manufacturer, the IP address at operating system level is **not** automatically adapted to the setting for device naming.
- The IP address at operating system level must be adapted using the setting options provided by the operating system (e.g. Microsoft Windows XP: Start -> Control Panel -> Network Connections).
- If the IP address of a PSS 4000-operable device is changed, other devices that do not belong to the PSS 4000 project will no longer be able to access the PSS 4000-operable device via any existing Ethernet communication relationships. The IP address for these Ethernet communication relationships must be adapted in the relevant devices.

8.8 Device naming when exchanging devices of PSSu systems

When a PSSu system is exchanged it is not strictly necessary to undergo a new device naming procedure via PAS4000. The device naming data and device project on the existing PSSu system can be transferred to the exchange device by re-using the existing removable data medium containing the naming data and device project for that device.



Information

If the PSSu system is used for safety-related applications, when exchanging devices it is absolutely essential that you comply with the information regarding changes as stated in the "PSS 4000 Safety Manual".

Prerequisites

The exchange PSSu system must still be in its delivery condition or the PSSu system must undergo an original reset prior to the exchange.

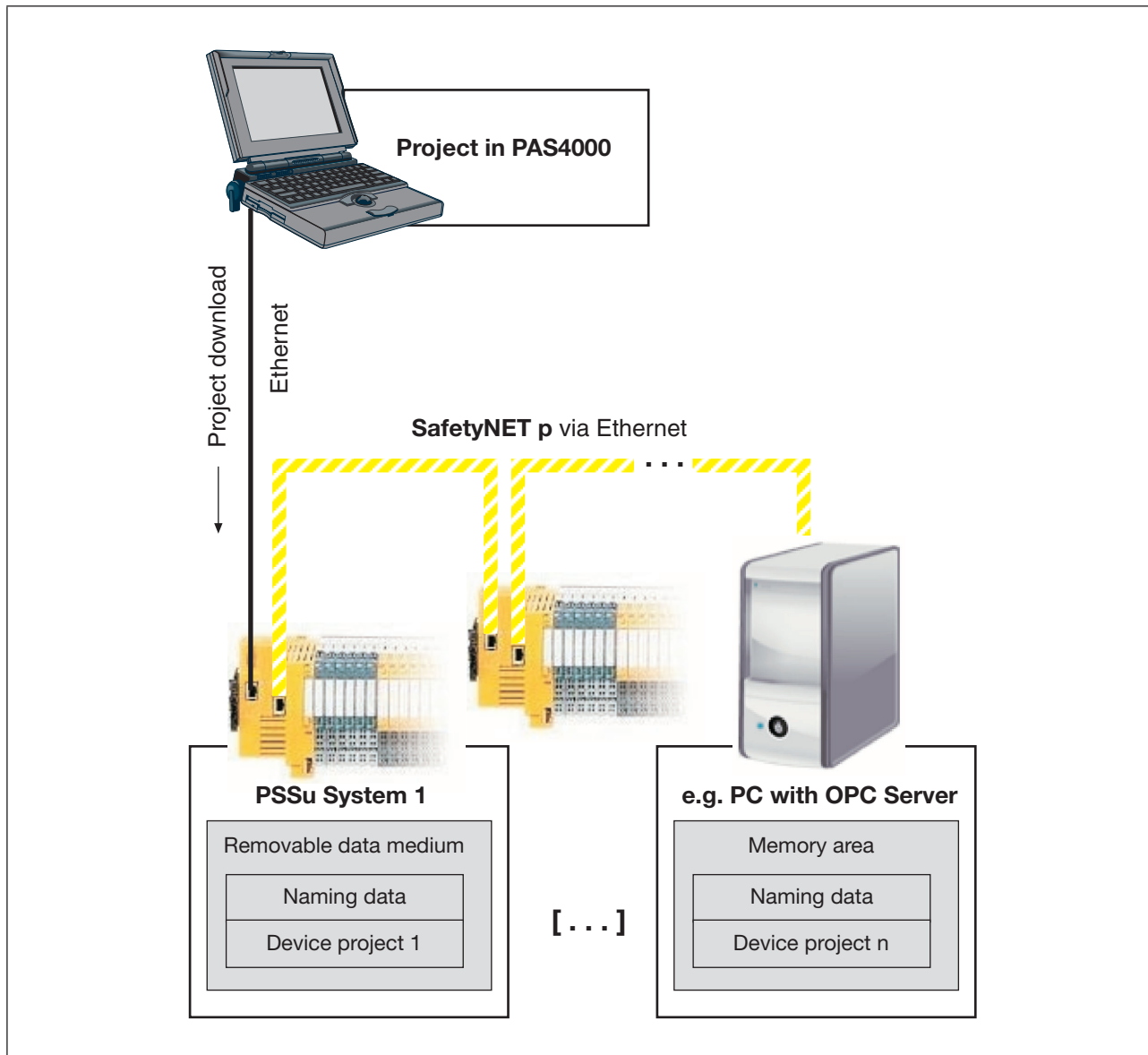
Procedure

1. Switch the existing PSSu system to a STOP condition and remove the supply voltage connectors.
2. Uninstall the existing PSSu system.
3. Remove the data medium from the head module of the existing PSSu system and plug it into the exchange device.
4. Install the exchange device.
5. Plug the supply voltage connectors into the exchange unit.
6. Follow the procedure below to transfer the device naming data and device project:
 - Press the reset button on the exchange device's head module and keep it held down for longer than 5 s
 - Perform the deliberate operator action for device naming:
Release the reset button - press the reset button - release the reset button
The deliberate operator action must be performed within 10 s, otherwise the transfer is aborted.

Once the naming data and device project have been transferred, the exchange PSSu system is restarted; the behaviour is the same as for a cold start/reboot command.

8.9 Project download

During the project download the project is divided into device projects and downloaded to the PSS 4000-exclusive devices (e.g. PSSu systems) and, where available, to the project's PSS 4000-operable devices.



Principle of project download (example)

Prerequisites

- ▶ There must be a physical connection (Ethernet cable) between the PAS4000-PC and the network containing the project's PSS 4000 devices. Establish connection between PAS4000-PC and devices.
- ▶ The project must be activated in PAS4000.
- ▶ Device naming must have been performed. Perform device naming in PAS4000.
- ▶ The project must have been built. Build project.
- ▶ On the project's PSS 4000 devices with a removable data medium, the removable data medium must be inserted.

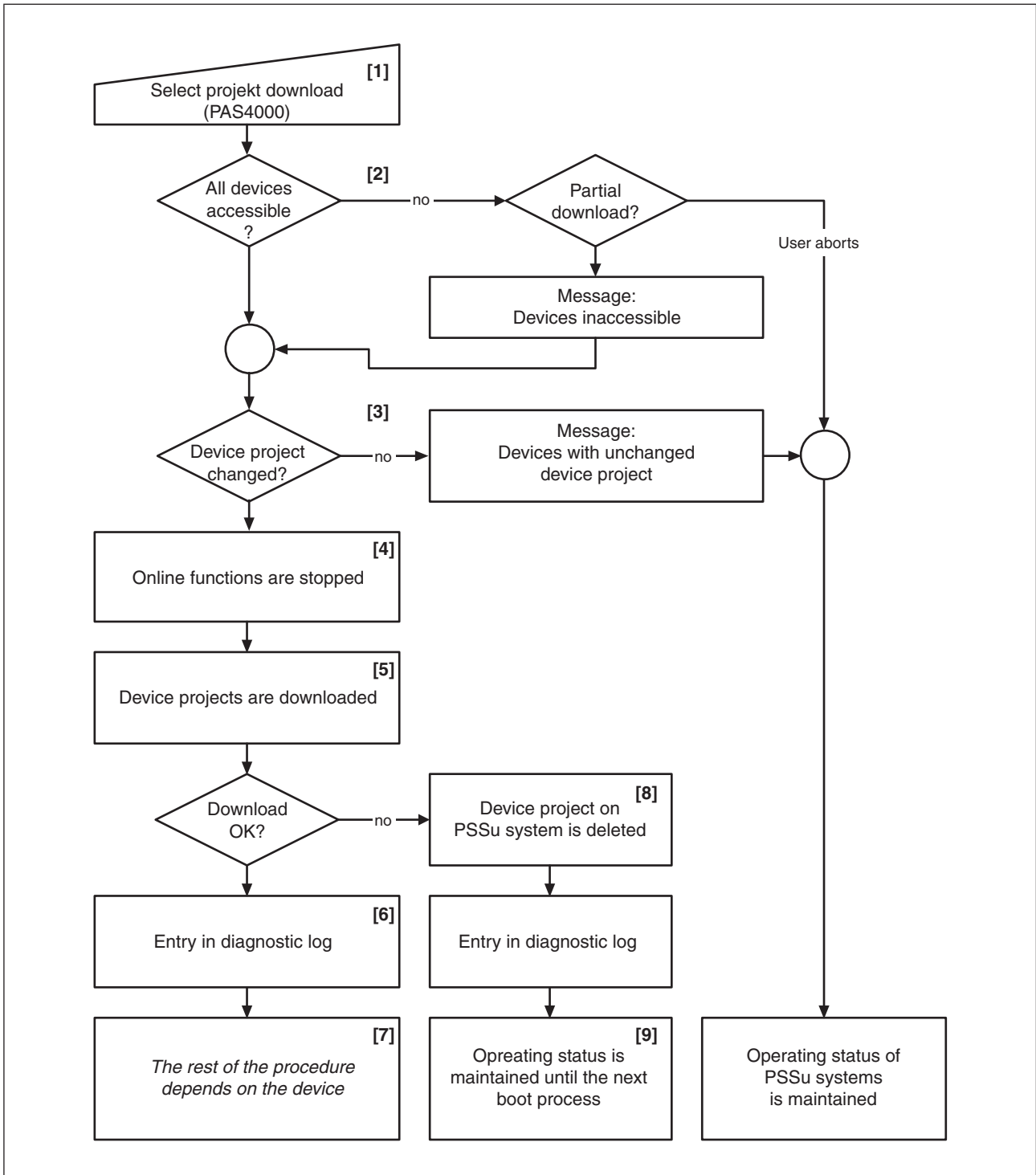
- ▶ There must be write access to the relevant memory areas.
For example:
 - Write protection on the removable data media must have been removed
 - "Write" permission must be set in the security settings for the relevant memory area of a PC.

**CAUTION!**

Please note that the download for a PSS 4000 device will not be performed if an internal error is detected in any of its system sections. An internal error leads to a corresponding operating state on the affected system section(s):

- Operating state: Major FS error
Affected system sections: FS resource, FS module bus, FS SafetyNET p RTFN and safe communication channel of a system section for external communication (e.g. PROFIsafe from the PROFIBUS-DP Slave with PROFIsafe system section):
- Operating state with communication via SafetyNET p RTFN: Major FS+ST error
Affected system sections: FS SafetyNET p RTFN and ST SafetyNET p RTFN:
- Operating state with external communication: Major FS+ST error
(This operating state can only occur if the system section has a safety-related communication channel; e.g. PROFIsafe from the PROFIBUS-DP Slave with PROFIsafe system section)
Affected system section: System section for external communication (e.g. PROFIBUS-DP Slave with PROFIsafe).

Procedure for project download



[1]
"Project download" is selected in PAS4000.

[2]
PAS4000 checks that all the project's devices are accessible (e.g. devices are switched on, network connection cables to the devices are plugged in). If some of the project's devices are inaccessible, the following options are available:

- ▶ User can abort the project download
The current operating status of all devices is maintained.
- ▶ Project download is continued
All accessible devices are selected for the project download. The devices excluded from the download will be displayed in PAS4000.

[3]

PAS4000 checks each device to see if there is any difference between the existing device project and the new device project.

If the device project is unchanged, the device will be excluded from the project download. Devices on which the device project is unchanged are displayed in PAS4000. The current operating state of these devices is maintained.

[4]

The online functions (e.g. force variables, watch variables) are stopped. The device remains in its current operating state.

[5]

The device projects are downloaded for the relevant devices.

- ▶ PSS 4000-exclusive devices (e.g. PSSu systems)
If the removable data medium on a PSS 4000-exclusive device already contains a device project, the device project will be deleted from the removable data medium and the check sum of the FS device project and project name will be deleted from the device memory. The new device project will then be stored on the removable data medium.
- ▶ PSS 4000-operable devices (e.g. operator terminal PMI, PC)
If available, on a PSS 4000-operable device the device project is deleted from the relevant memory area first and then the new device project is stored in the memory area.

[6]

Once a project has been downloaded successfully, an entry is made in the diagnostic log.

[7]

Once a device project has been successfully downloaded, the rest of the procedure depends on the device:

- ▶ PSS 4000-exclusive devices (e.g. PSSu systems)
The device is restarted automatically.

Impact of start-up:

- The impact of start-up is that of a warm reset.
- PSS 4000-exclusive devices that have been stopped manually prior to the project download via one of the stop commands will not be started automatically. These devices will need to be restarted manually.
- If the user program from the old device project contained non-volatile variables, the device will start up with the following non-volatile variable values:

	Variables with the attribute RETAIN PERSISTENT
The unchanged variable is available in the user program	Saved values
Variable in the user program is new	Set to initial values
Variable is no longer available in the user program	Delete from the variable memory

Note

When the project is downloaded to the devices, the content of a variable with the attribute RETAIN PERSISTENT is retained if

- the resource assignment of the block in which the variable is used,
- the instance path of the variable and
- its data type

are unchanged compared with the previous download.

- ▶ PSS 4000-operable devices from Pilz (e.g. operator terminals PMI)
The device is restarted automatically.
- ▶ PSS 4000-operable devices from a third-party manufacturer (e.g. PC)
The PSS 4000 firmware on the device is restarted automatically.

[8]

Various errors may arise as the device project is downloaded to a PSS 4000 device.

Examples:

- ▶ User aborts
- ▶ Timeout
- ▶ Error occurs while saving the device project

Impact

On PSS 4000-exclusive devices, the device project on the PSS 4000 device is deleted from the removable data medium and the project name and check sum of the FS device project are deleted from the device memory. On PSS 4000-operable devices, the device project is deleted from the relevant memory area. An entry is added to the diagnostic log.

Remedy

Download project again

[9]

The device's operating status is maintained until the next boot process. The next time the system is booted (e.g. after a cold start/reboot command), the device will start up without a device project.

8.10 Sealed FS blocks

Sealed blocks are a programming aid. Sealed blocks cannot be modified and are safe from manipulation.

Sealed FS blocks process safety-related tasks that are required frequently (e.g. E-STOP).

They can be created by the user or are available from Pilz.

The use of certified, sealed FS blocks can speed up the certification process on projects, as the user program they contain will not require re-testing, even when used on machinery that is subject to mandatory inspection. There are

- ▶ Sealed user blocks

Additional information may be stored for sealed user blocks. This includes, for example:

- Name of the person who tested the sealed block
- Company name
- Name of the notified body
- Certification number

- ▶ Sealed Pilz blocks

Pilz provides various blocks which have been tested and certified by a notified body (e.g. BG, TÜV).

It is preferable to use blocks that have been certified by a notified body. You should check that the assumed operating conditions for these blocks match the conditions required by the user program.

9 Mode of operation

9.1 Communication between PAS4000 and PSS 4000 devices

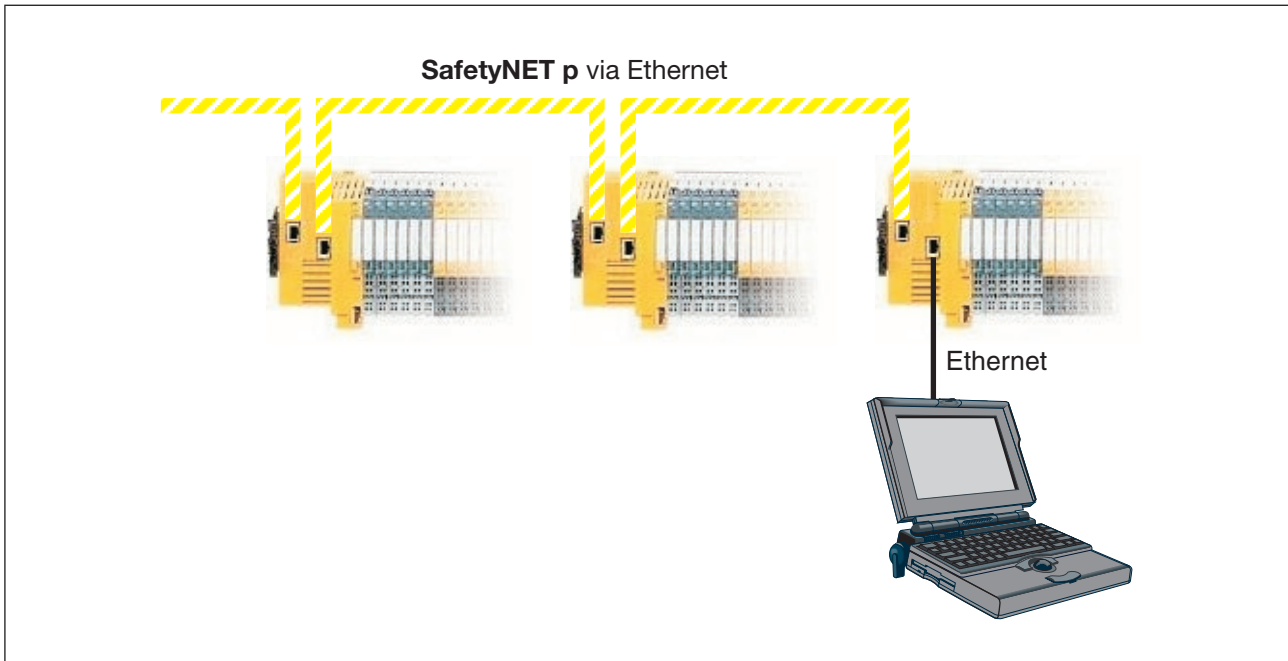
In order to perform online actions, the computer on which PAS4000 is installed (PAS4000-PC) must be connected to the network containing the project's PSS 4000 devices (PSS 4000-exclusive devices such as PSSu systems and/or PSS 4000-operable devices).

Further information is available in PAS4000's online help.

9.1.1 Establish a point-to-point connection

With a point-to-point connection, PAS4000 can access an individual PSS 4000-exclusive/operable device as well as all PSS 4000-exclusive/operable devices on the network.

Principle



Principle of a point-to-point connection with PAS4000

Please refer to the information on networks without Routers (see [Network without Routers, Page 97](#))

9.1.2 Establish a remote connection

With a remote connection, PAS4000 can access an individual PSS 4000-exclusive/operable device as well as all PSS 4000-exclusive/operable devices on the network.

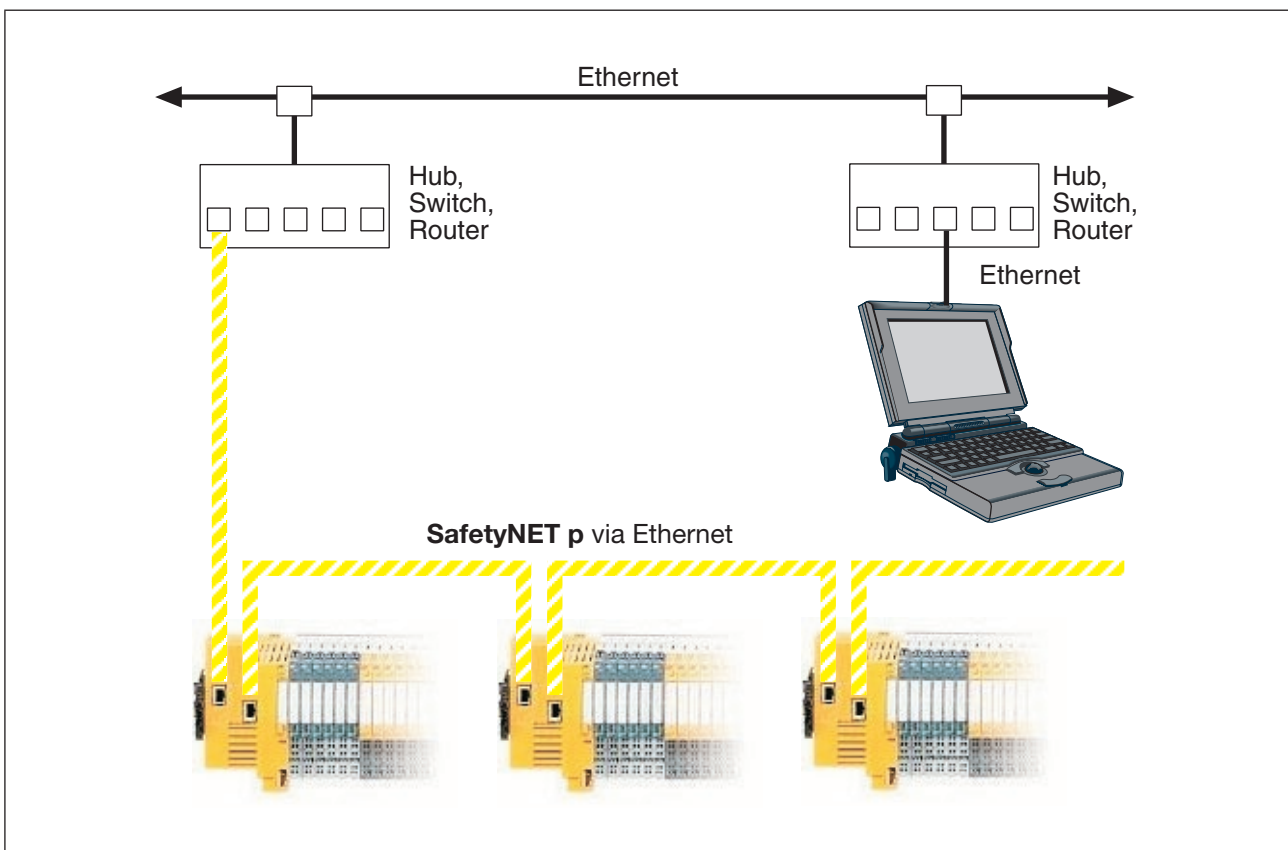


CAUTION!

Data security risk

Please note the special features regarding data security when communicating via remote connections (e.g. for diagnostics) (see "PSS 4000 Safety Manual").

Principle



Principle of a remote connection with PAS4000

Please note the restrictions when using Routers Network with Routers.

9.2 Operating states and change in operating status of a PSSu system

There are some operating states and changes in operating status that affect all system sections on a PSSu system and others whose impact is restricted to one plant section. A PSSu system may contain all or just a subset of the following system sections:

- ▶ Resources
 - FS resource system section
 - ST resource system section
- ▶ Module bus
 - FS module bus system section
 - ST module bus system section
- ▶ SafetyNET p RTFN
 - FS SafetyNET p RTFN system section
 - ST SafetyNET p RTFN system section
- ▶ External communication
 - IP connections system section (Modbus/TCP, Raw UDP)
 - PROFIBUS-DP Slave system section

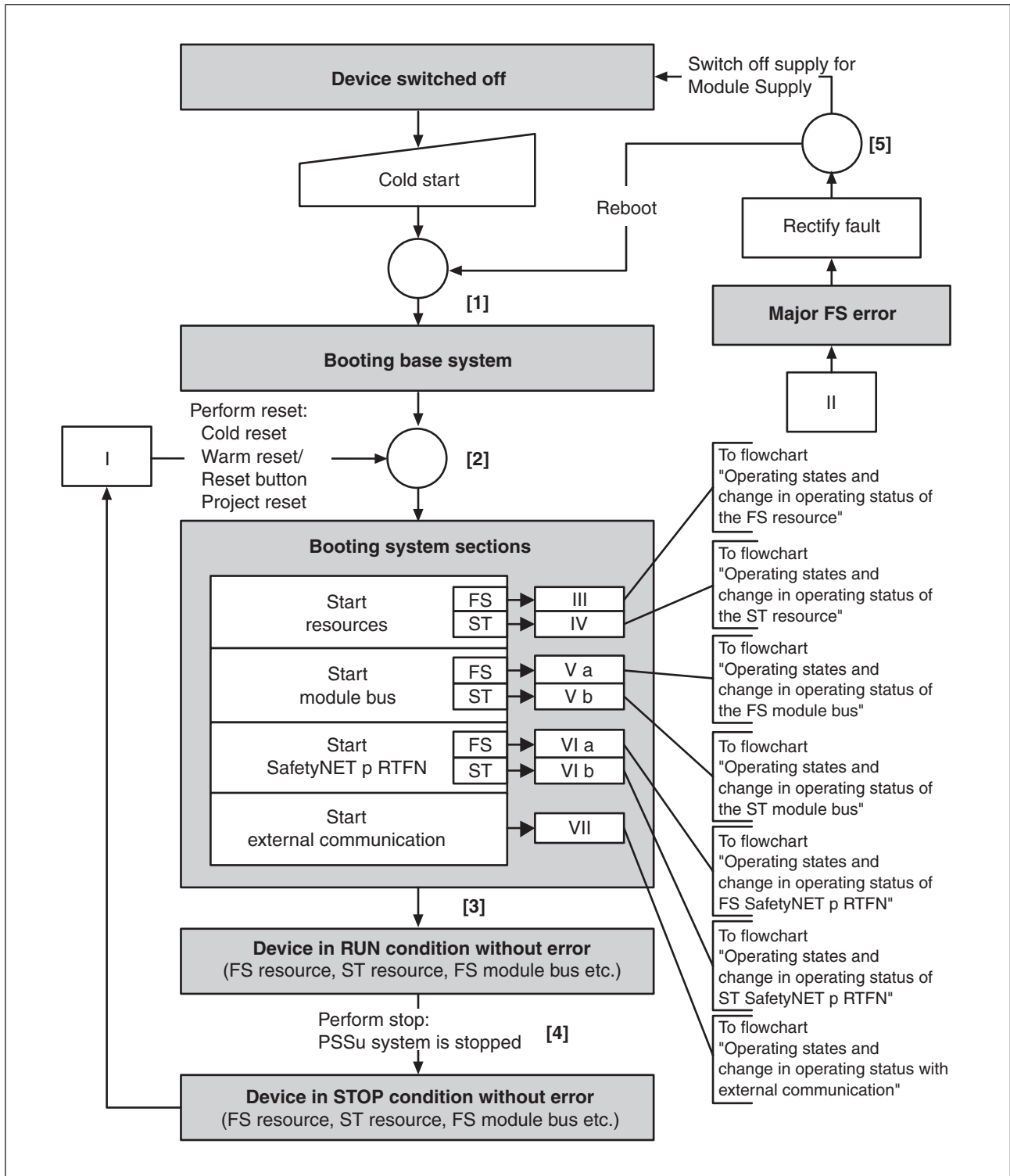
The description below deals exclusively with the operating states that affect all of the system sections on a PSSu system.

The diagram "Operating states and change in operating status of a PSSu system" (see Flowchart, [Page 248](#)) provides an overview of the operating states (grey):

- ▶ The figures in square parentheses indicate the various changes in operating status.
- ▶ The sections with Roman numerals I and II are connection points, which are referenced from other diagrams.
- ▶ The sections with Roman numerals III, IV, V a/b, VI a/b and VII refer to diagrams showing operating states and changes in the operating status of system sections.

The following description assumes that the PSSu system contains a valid device project.

9.2.1 Flow chart



Operating states and change in operating status of a PSSu system

9.2.2 Operating states on the PSSu system

A PSSu system may assume the following operating states (see Flowchart, [Page 248](#)):

- ▶ Operating status "Device switched off"
- ▶ Operating status "Booting base system"
- ▶ Operating status "Booting system sections"
- ▶ Operating status: "Device in RUN condition without error"
- ▶ Operating status: "Device in STOP condition without error"
- ▶ Operating status "Major FS error"



Information

The description of each operating status includes a list of the options for triggering a change in operating state. These options are described separately (see Reset, restart, start and stop options, [Page 337](#)).

9.2.2.1 Device switched off

"Device switched off" is the PSSu system in a power-free state. Switching on the supply voltage for the module supply (cold start) switches the PSSu system to the operating status "Booting base system".

9.2.2.2 Operating status "Booting base system"

Various start-up tests are performed (e.g. memory test). If there are any faults during the boot process, the PSSu system will not start. The "DIAG" LED on the PSSu system's head module flashes red.

Remedy

- ▶ Check the technical details (e.g. supply voltage for module supply, ambient temperature)
- ▶ It may be necessary to swap the device

9.2.2.3 Operating status "Booting system sections"

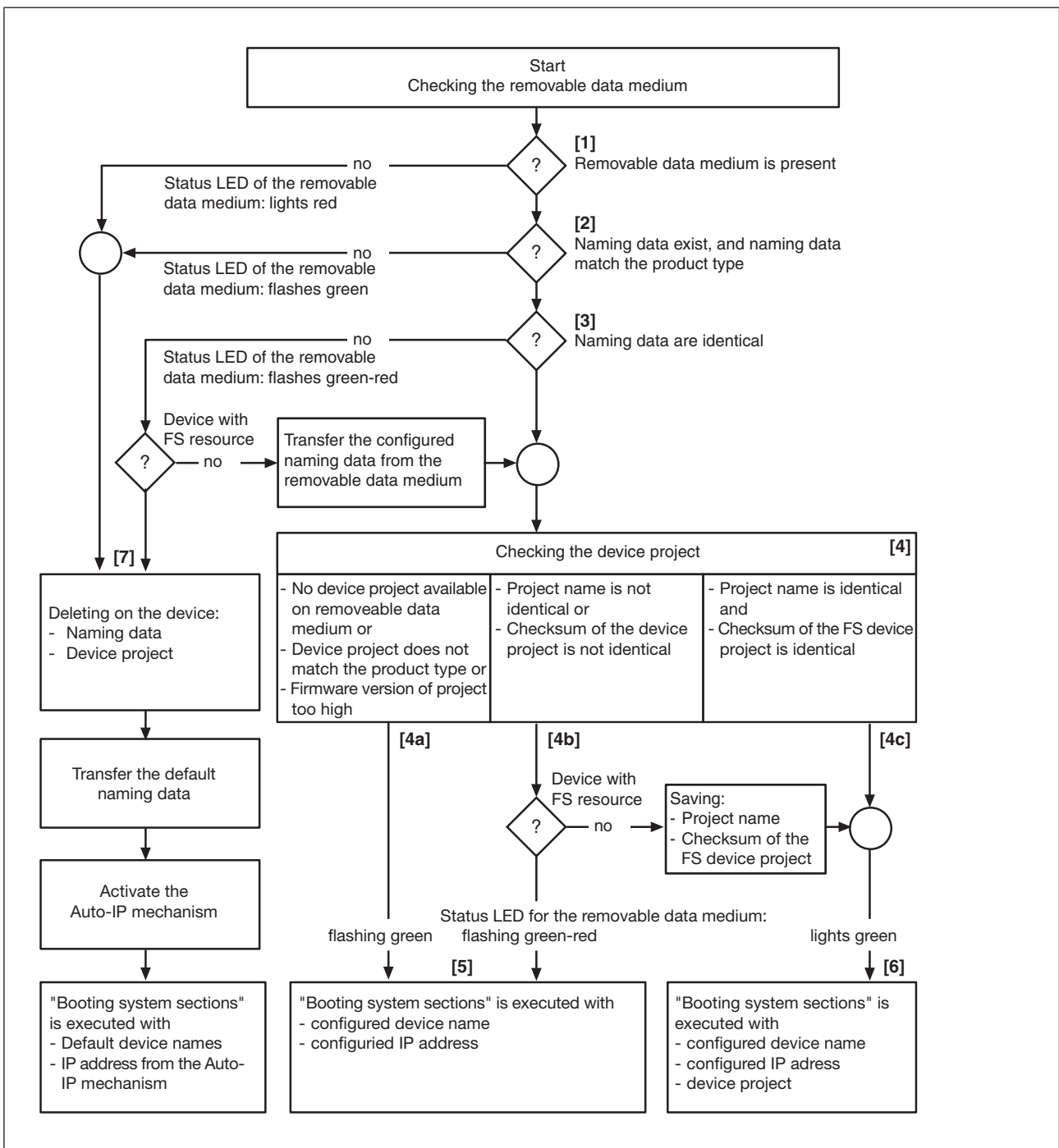
"Booting system sections" is either the phase

- ▶ After the base system is booted, until all system sections of a PSSu system are in a RUN condition or
- ▶ After a reset command is triggered (cold reset, reset button/warm reset), until all system sections of the PSSu system are restored to a RUN condition or
- ▶ After the base system is booted or a reset command is triggered, until a system section is in an error condition

These error conditions are described in the respective system section.

Effects

Before the system sections are booted, various self tests and initialisations are performed. This may take some time. One of the self tests deals is concerned with checking the removable data medium and its data.



Flowchart for checking the removable data medium

[1]

The first test checks whether the device's removable data medium is accessible, i.e. it checks whether the removable data medium is

- ▶ plugged in
- ▶ intact
- ▶ not empty

If the removable data medium is not accessible, the device will start without a device project and with the factory-set default naming data (see [7]).

[2]

A test determines whether the removable data medium contains naming data (see Naming data, [Page 223](#)) and whether the naming data on the removable data medium matches the device's product type.

If the removable data medium contains no naming data or the naming data does not match the device's product type, the device will start without a device project and with the factory-set default naming data (see [7]).

[3]

A test determines whether the naming data on the device matches that of the removable data medium.

- ▶ Identical naming data:
The device project is tested as the next step (see [4]).
- ▶ Non-identical naming data on a device **without** FS resource (e.g. device belonging to the "decentralised system PSSu I/O"):
The naming data on the device is deleted and the device adopts the naming data on the removable data medium. The device project is tested as the next step (see [4]).
- ▶ Non-identical naming data on a device **with** FS resource (e.g. control system PSSu PLC, control system PSSu multi):
The device will start without a device project and with the factory-set default naming data (see [7]).

[4]

Various tests are carried out for the device project. The rest of the procedure depends on the result of the test:

- ▶ Procedure follows [4a] if at least one of the following conditions is met:
 - There is no device project on the removable data medium;
This may occur during device naming for example.
 - The device project on the removable data medium does not match the device's product type;
This may occur if the removable data medium has been exchanged, for example.
 - The device requires a higher firmware version for the device project on the removable data medium

In all these cases, the device will start with the naming data configured by the user (device name, IP address), but without a device project (see [5]). For the device to obtain a device project, a project will either need to be downloaded (see Project download, [Page 238](#)) or the "Deliberate operator action" will need to be performed (see Deliberate operator action to transfer the naming data and/or device project, [Page 356](#)).

- ▶ Procedure follows [4b] if at least one of the following conditions is met:
 - The project name on the device is not identical to the project name on the removable data medium;
This may occur if the removable data medium has been exchanged, for example.

- On a device with FS resource, the check sum of the FS device project on the device is not identical to the check sum of the FS device project on the removable data medium;

This may occur if the removable data medium has been exchanged, for example.

A device **with** FS resource will start with the naming data configured by the user (device name, IP address), but without a device project (see [5]). For the device to obtain a device project, a project will either need to be downloaded (see Project download, [Page 238](#)) or the "Deliberate operator action" will need to be performed (see Deliberate operator action to transfer the naming data and/or device project, [Page 356](#)).

A device **without** FS resource will delete the project name and the check sum of the FS device project from its device memory and adopt the project name and check sum of the FS device project on the removable data medium. The device will start with the naming data configured by the user (device name, IP address) and with the device project from the removable data medium (see [6]).

- ▶ Procedure follows [4c] if one of the following conditions is met:
The project name and check sum of the FS device project are identical on the removable data medium and device.
This procedure corresponds to the device in a "sound condition", i.e. the device meets all the requirements necessary for the operation to run smoothly. For example, these requirements include:
 - Removable data medium is accessible
 - Device naming and project download have been carried out successfully
 - Removable data medium has not been exchanged improperly

The device will start with the naming data configured by the user (device name, IP address) and with the device project from the removable data medium (see [6]).

[7]

The procedure follows [7] in the following cases:

- ▶ A removable data medium is not plugged into the device or it is empty or defective.
- ▶ The removable data medium contains no naming data or the naming data on the removable data medium does not match the device's product type. This may occur, for example, if the removable data medium that is used is unsuitable or is from a device with a different product type.
- ▶ On a device with FS resource, the naming data on the removable data medium is not identical to that of the device. This may occur, for example, if the naming data has been changed during a new device naming process.

The following is performed when the procedure follows [7]:

- ▶ The naming data on the device is deleted.
- ▶ The project name and check sum of the FS device project on the device are deleted.
- ▶ The device activates the factory-set default naming data.
- ▶ The device activates its Auto-IP mechanism.
- ▶ The operating status "Booting system sections" is performed. The device uses
 - the factory-set default device name
 - the IP address that was assigned via the Auto-IP mechanism

Behaviour during "Booting system sections":

- ▶ If no faults are detected during the boot process, all system sections will be started. The starts are executed in parallel (see Sections III ... VI in the flowchart, [Page 248](#)).
- ▶ If a fault is detected during the boot process, the affected system section will switch to one of the error conditions. All unaffected system sections continue with the boot process and switch to a RUN condition (see Sections III ... VII in the flowchart, [Page 248](#)).
- ▶ The boot process has been completed when
 - All system sections have successfully loaded their configuration ("DIAG" LED switches from "Flashing green" to "Lights green")
 - A system section registers a fault ("DIAG" LED switches from "Flashing green" to "Lights orange", "Lights red" or "Flashes red"), then switches to an error condition.
In these cases, unaffected system sections have still loaded their configuration successfully.

▶ Status LEDs

The status LEDs on the head module provide information on the status of the system sections.

9.2.2.4 Operating status "Device in RUN condition without error"

This operating status is characterised by the fact that all system sections are in a RUN condition and are operating error-free (see Sections III ... VII in the flow chart, [Page 248](#)).

▶ Status LEDs

The status LEDs on the head module provide information on the status of the system sections.

9.2.2.5 Operating status "Device in STOP condition without error"

This operating status is achieved by triggering an explicit stop via PAS4000 (PSSu system is stopped).

Effects

- ▶ The system sections finish what they are currently processing. Any further processing is stopped. All system sections are in a STOP condition. The behaviour during the stop state is described separately (see Stop commands – Behaviour during the STOP condition, [Page 348](#)).

- ▶ Status LEDs

The status LEDs on the head module provide information on the status of the system sections.

Remedy

The following commands are available for starting the PSSu system:

- ▶ Warm reset (PAS4000) or reset button (PSSu system)
- ▶ Cold reset (PAS4000)
- ▶ Start project (PAS4000)

9.2.2.6 Operating status "Major FS error"

If a fault results in the operating status: "Major FS error", the affected sections of the PSSu systems are ended as defined. A defined status is achieved.

Possible causes

- ▶ Feasibility error
 - e.g. irregularities between the microprocessors (different calculation results)
- ▶ Memory error

Effects

- ▶ In the event of a major error in a safety-related system section, all other safety-related system sections are stopped and/or switched to a safe condition:
 - The FS resource is in a STOP condition and the FS resource tasks are no longer executed.
 - All hardware FS outputs are switched to a safe condition.
 - FS process data communication via SafetyNET p is stopped.
- ▶ For process data whose values are determined via the FS resource and which is accessed via other resources, the substitute values are used with Valid Bits = FALSE.
- ▶ Please refer to PAS4000's online help for details of how to handle variables (volatile variables, non-volatile variables) whose values are determined via the FS resource
- ▶ Hardware ST outputs are unaffected
- ▶ Status LEDs on the head module
 - The "FS RUN" LED is off, the "FS SNp" LED lights up red and the "MBUS" LED flashes red.

Remedy

1. Diagnostics
 - e.g. use PAS4000 to read the project diagnostic list and/or the diagnostic log
2. Rectify errors
3. Execute a cold start/reboot command

9.2.3 Changes in the operating status of a PSSu system

This description of the change in operating status refers to the diagram "Operating states and change in operating status of a PSSu system" (see Flowchart, [Page 248](#))

[1] Change in operating status after a cold start/reboot command

When a PSSu system's module supply is switched on (cold start) or following a reboot command (PAS4000), the PSSu system initially switches to the status "Booting base system", performs various start-up tests and then switches to the status "Booting system sections".

Fundamentally, a change in operating status after "booting the base system" can be enforced via a cold start/reboot command when the PSSu system is in any status (see Section II in the flowchart, [Page 248](#)). For reasons of availability, we recommend that you only enforce a change in operating status when it is really necessary. For example, after rectifying a fault that has led to a major error.

[2] Change in operating status after a reset

Fundamentally, a reset (cold reset, reset button/warm reset or project reset) can be triggered when the PSSu system is in any status (see Section I in the flowchart, [Page 248](#)). The effect (e.g. how variables are handled) depends on which reset is used. After a reset, the PSSu system switches to the status "Booting system sections".

For reasons of availability, we recommend that you only trigger a reset when it is really necessary.

The effects of the reset are described separately (see Reset commands, [Page 343](#)).

[3] Switching from "Booting system sections" to "Device in RUN condition without error"

This change of operating status takes place as soon as all the system sections on the PSSu system are in a RUN condition and are operating faultlessly.

Effects

- ▶ The millisecond timer is started.
- ▶ The content of non-volatile and volatile variables is documented in the section entitled "Reset, restart, start and stop options (see Effects on local and resource-global variables, [Page 339](#)).
- ▶ All values in the PII/PIO are set to ZERO.
It's possible that the module bus may have already been started by the FS resource and/or ST resource. In this case, the PII may contain current input values.
- ▶ An entry is added to the diagnostic list.

[4] Switching from "Device in RUN condition without error" to "Device in STOP condition without error"

This change of operating status must be triggered explicitly via PAS4000 (e.g. stopping the PSSu system).

The effects of a stop are described separately (see Stop commands, [Page 348](#)).

[5] Switching from "Major FS error" to "Booting system sections"

Once the fault has been rectified, the PSSu system can only be started via a cold start/reboot command. As a result of the cold start/reboot command the PSSu system switches to "Booting base system", performs the start-up tests and then runs through the complete boot process for the status "Booting system sections".

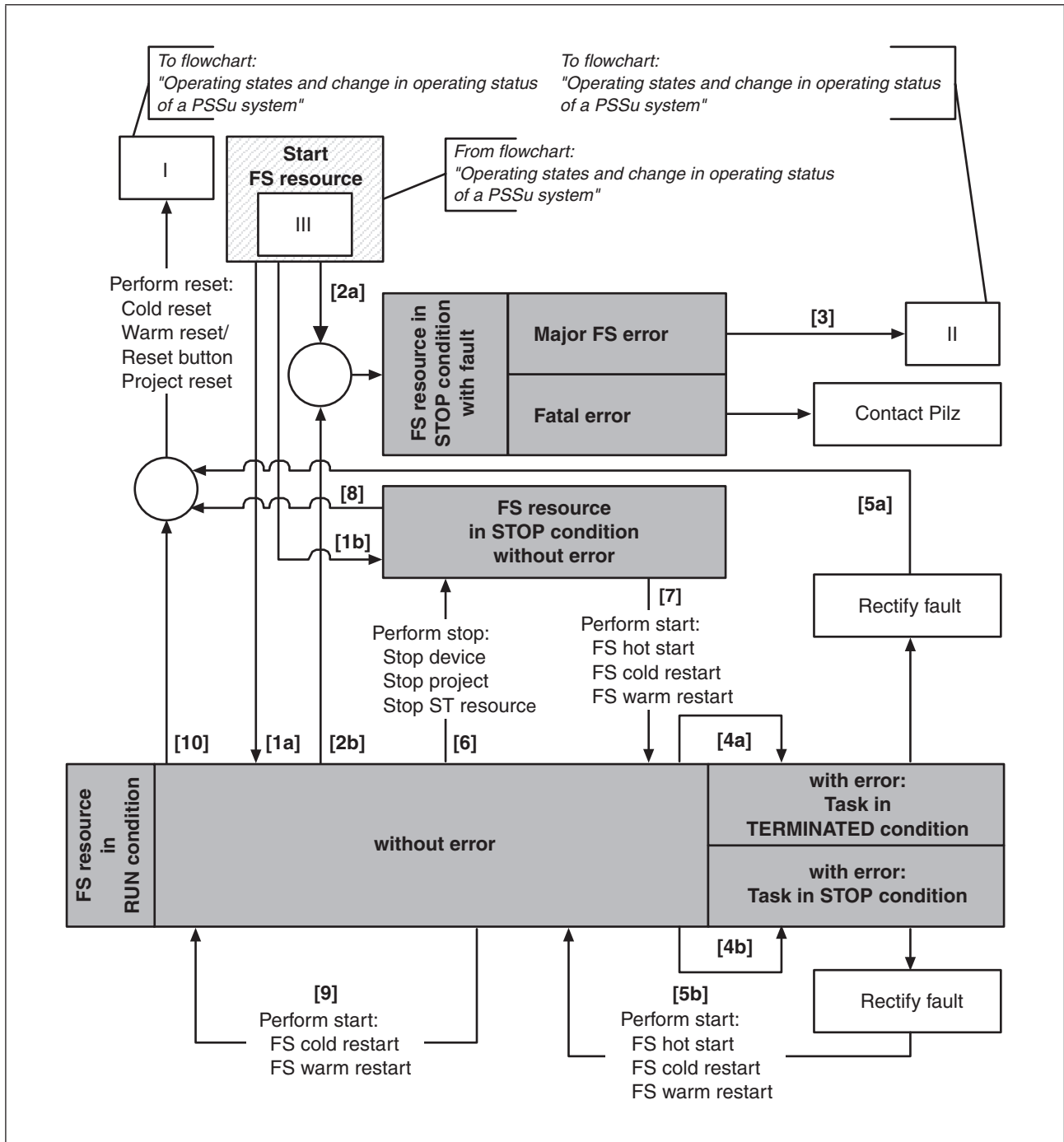
9.3 Operating states and change in operating status of the FS resource

The description below deals exclusively with the operating states of the FS resource, referring to the operating states of the PSSu system where necessary.

The diagram "Operating states and change in operating status of an FS resource" (see Flowchart, [Page 262](#)) provides an overview of the operating states (grey):

- ▶ The figures in square parentheses indicate the various changes in operating status.
- ▶ The sections marked with the Roman numerals I and II refer to the diagram "Operating states and change in operating status of a PSSu system" (see Flowchart, [Page 262](#)), as the effects impact the whole PSSu system and not just the FS resource.
- ▶ The section marked with the Roman numeral III refers to the diagram "Operating states and change in operating status of a PSSu system", (see Flowchart, [Page 248](#)) as the FS resource is started when booting the system sections of a PSSu system.

9.3.1 Flow chart



Operating states and change in operating status of the FS resource

9.3.2 Operating states of the FS resource

The description of the various operating states assumes that the PSSu system contains a valid device project.

The FS resource may assume the following operating states (see Flow chart, [Page 262](#)):

- ▶ Operating status conditions "FS resource in RUN condition":
 - FS resource in RUN condition without error
 - FS resource in RUN condition with error: Task in TERMINATED condition
 - FS resource in RUN condition with error: Task in STOP condition
- ▶ Operating status "FS resource in STOP condition without error"
- ▶ Operating status conditions "FS resource in STOP condition with error"
 - Fatal error
 - Major FS error



Information

The descriptions of various operating states include a list of the options for triggering a change in operating status. These options are described separately (see Reset, restart, start and stop options, [Page 337](#)).

9.3.2.1 FS resource in RUN condition without error

This operating status is achieved through one of the following options:

- ▶ After a successful boot process, which was triggered by
 - Cold starting the PSSu system or a reboot command for the PSSu system via PAS4000
 - Pressing the reset button on the PSSu system
 - Triggering a reset via PAS4000
(cold reset, warm reset, project reset)
- ▶ After a start or restart via PAS4000
(FS cold restart, FS warm restart, FS hot start)

Effects

- ▶ In the operating status "FS resource in RUN condition without error", the FS resource tasks are run faultlessly.
- ▶ Status LEDs on the head module
The "FS RUN" LED lights up green.

9.3.2.2 FS resource in STOP condition without error

This operating status is achieved by triggering a stop via PAS4000 (PSSu system is stopped, FS resource is stopped, project is stopped).

Effects

- ▶ The behaviour of the hardware inputs, hardware outputs, process data and system sections while a resource is in a STOP condition is described separately (see Stop commands – Behaviour during the STOP condition, [Page 348](#)).
- ▶ The effects on local and resource-global variables are described separately (see Effects on local and resource-global variables, [Page 339](#)).
- ▶ Status LEDs on the head module
The FS-RUN LED will be off.

Remedy

- ▶ FS resource is (re-)started:
 - FS hot start (PAS4000)
 - FS cold restart (PAS4000)
 - FS warm restart (PAS4000)
- ▶ PSSu system is reset:
 - Warm reset (PAS4000) or reset button (PSSu system)
 - Cold reset (PAS4000)
- ▶ Project is reset
 - Warm reset of project (PAS4000)
 - Cold reset of project (PAS4000)

9.3.2.3 FS resource in RUN condition with error: Task in TERMINATED condition

If an error results in this operating status, the affected task is ended as defined. A defined state is achieved.

Possible causes

(see also IEC 61131-3, Table E1)

- ▶ Execution time error

For example, timeout error because the execution time has been exceeded

Effects

- ▶ The affected FS resource task is no longer executed, all other FS resource tasks continue to be executed.
- ▶ The FS resource continues to be in a RUN condition
- ▶ The following applies for all O-PI variables that are declared as local variables or are referenced as VAR_EXTERNAL in the POU's of the task that has been stopped: Substitute values are used in the I/O-mapped data sinks (e.g. I-PI variables, hardware outputs) and Valid Bits = FALSE.
- ▶ The following applies for all O-PI variables that are created in Multi blocks as O-PI points or are referenced as VAR_EXTERNAL: Substitute values are used in the I/O-mapped data sinks (e.g. I-PI variables, hardware outputs) and Valid Bits = FALSE.
- ▶ The following applies for resource global variables (not O-PI variables) that are referenced via "VAR_EXTERNAL" in the blocks of the task that has been ended: They retain the last valid, stored values and are not set to their initial values.
- ▶ If resource global variables are referenced via "VAR_EXTERNAL" in the blocks of a task that has not been ended, the value there can be overwritten. As a result of this overwrite, resource global O-PI variables will again be marked as valid.
- ▶ Please refer to PAS4000's online help for details of how to handle variables (volatile variables, non-volatile variables)
- ▶ Status LEDs on the head module
The "FS RUN" LED flashes green

Remedy

1. Diagnostics
e.g. use PAS4000 to read the diagnostic list and/or the diagnostic log
2. Rectify errors
3. Perform reset:
Use a warm reset or cold reset

9.3.2.4 FS resource in RUN condition with error: Task in STOP condition

If an error results in this operating status, the affected task is ended as defined. A defined state is achieved.

Possible causes

(see also IEC 61131-3, Table E1)

- ▶ Runtime error

For example, a variable value exceeds the defined range, the numeric result of a function exceeds the value range for a data type, division by 0

Effects

- ▶ The affected FS resource task is no longer executed, all other FS resource tasks continue to be executed.
- ▶ The FS resource continues to be in a RUN condition
- ▶ The following applies for all O-PI variables that are declared as local variables or are referenced as VAR_EXTERNAL in the POU's of the task that has been stopped: Substitute values are used in the I/O-mapped data sinks (e.g. I-PI variables, hardware outputs) and Valid Bits = FALSE.
- ▶ The following applies for all O-PI variables that are created in Multi blocks as O-PI points or are referenced as VAR_EXTERNAL: Substitute values are used in the I/O-mapped data sinks (e.g. I-PI variables, hardware outputs) and Valid Bits = FALSE.
- ▶ The following applies for resource global variables (not O-PI variables) that are referenced via "VAR_EXTERNAL" in the blocks of the task that has been stopped: They retain the last valid, stored values and are not set to their initial values.
- ▶ If resource global variables are referenced via "VAR_EXTERNAL" in the blocks of a task that has not been stopped, the value there can be overwritten. As a result of this overwrite, resource global O-PI variables will again be marked as valid.
- ▶ Please refer to PAS4000's online help for details of how to handle variables (volatile variables, non-volatile variables)
- ▶ Status LEDs on the head module
The "FS RUN" LED flashes green

Remedy

1. Diagnostics
e.g. use PAS4000 to read the diagnostic list and/or the diagnostic log
2. Rectify errors
3. Perform a reset or (re)start:
Use an FS hot start, FS warm restart or FS cold restart
One of the reset commands can also be used. A reset command impacts the whole PSSu system and not just the FS resource.

9.3.2.5 Major FS error

If a fault results in this operating status, the affected sections of the FS resource are ended as defined. A defined state is achieved.

The effects of a major FS error impact all safety-related sections of a PSSu system (see Operating state "Major FS error of a PSSu system, [Page 259](#)).

9.3.2.6 Fatal error

If a fault results in the operating status "Fatal error", as far as possible an attempt will be made to exit all the processes running on the PSSu system in an orderly manner. Safety-related system sections are stopped and/or switched to a safe condition.

Possible causes

- ▶ Initialisation error
- ▶ Internal firmware error
(e.g. memory error which, although it was detected, has led to a major error in the PSSu system's firmware, as a result of which the firmware can no longer be executed correctly.)
- ▶ Hardware defect

Effects

- ▶ FS hardware outputs are switched to a safe condition.
- ▶ The content of variables and the status of ST hardware inputs/ST hardware outputs is random.
- ▶ Status LEDs on the head module
The status of the LEDs "FS RUN", "ST RUN", "FS SNp", "ST SNp" and "DIAG" is random.

Remedy

It is not possible for the user to rectify the error. If the PSSu system is in this operating state, follow the instructions below:

1. Note the conditions under which the error occurred.
2. Execute a cold start/reboot command in order to read/print the diagnostic list and/or diagnostic log using PAS4000.
However, this will only be possible if the fatal error was not caused by a fault in the communication area.
3. Contact Pilz

9.3.3 Change in operating status of the FS resource

This description of the change in operating status assumes that the PSSu system contains a valid device project. The figures refer to the change in operating status in the diagram "Operating states and change in operating status of the FS resource" (see Flowchart, [Page 262](#)).



Information

The description of each change of operating status includes a list of the options for triggering the change in operating status. These options are described separately (see Reset, restart, start and stop options, [Page 337](#)).

[1a] Switching from "Booting system sections" to "FS resource in RUN condition without error"

"Booting system sections" is one of the PSSu system's operating states (see Flowchart, [Page 248](#)). If no fault is detected in a safety-related system section as the system sections are booted, the FS resource is started, switching to the status "FS resource in RUN condition without error".

[1b] Switching from "Booting system sections" to "FS resource in STOP condition without error"

"Booting system sections" is one of the PSSu system's operating states (see Flowchart, [Page 248](#)). Under various circumstances the FS resource may not achieve the operating status "FS resource in RUN condition without error", it changes instead to "FS resource in STOP condition without error":

- ▶ A manually executed STOP command (see [6]) causes the FS resource to be in the operating condition "FS resource in STOP condition without error". Instead of one of the start commands (see [7]) one of the reset commands is executed (see [8]). However, in this case, the FS resource cannot achieve a RUN condition via any of the reset commands; instead it switches back to "FS resource in STOP condition without error".
- ▶ The FS resource is in the operating status "FS resource in RUN condition with error": Task in a STOP condition" After the fault has been rectified, one of the reset commands is executed instead of one of the start commands (see [5b]). However, in this case, the FS resource cannot achieve a RUN condition via any of the reset commands; instead it switches to "FS resource in STOP condition without error".

[2a] Switching from "Booting system sections" to an error condition

If a fault is detected in a safety-related system section as the system sections are booted (see Flowchart, [Page 248](#)), the FS resource switches to either of the operating states "Major FS error" or "Fatal error", depending on the fault.

[2b] Switching from "FS resource in RUN condition without error" to an error condition

If a fault is detected in a safety-related system section while the FS resource is in a RUN condition, the FS resource switches to either of the operating states "Major FS error" or "Fatal error", depending on the fault.

[3] Switching from "Major FS error" to "Bootling system sections"

Once the error has been rectified, the FS resource can only be started via a cold start/reboot command. As a result of the cold start/reboot command the PSSu system switches to "Bootling base system", performs the start-up tests and then runs through the complete boot process for the status "Bootling system sections" (see Flowchart, [Page 248](#)).

[4a] Switching from "FS resource in RUN condition without error" to "FS resource in RUN condition with error: Task in TERMINATED condition"

If an error is detected in an FS resource task, e.g. an execution time error is found, the task is aborted and is no longer executed. All other FS resource tasks continue running.

[4b] Switching from "FS resource in RUN condition without error" to "FS resource in RUN condition with error: Task in STOP condition"

If an error is detected in an FS resource task, e.g. a runtime error is found, the task is stopped and is no longer executed. All other FS resource tasks continue running.

[5a] Switching from "FS resource in RUN condition with error: Task in TERMINATED condition" after "FS resource in RUN condition without error"

Once the fault has been rectified, the FS resource can be restarted via one of the reset commands:

- ▶ Warm reset (PAS4000) or reset button (PSSu system)
- ▶ Cold reset (PAS4000)
- ▶ Warm reset of project (PAS4000)
- ▶ Cold reset of project (PAS4000)

[5b] Switching from "FS resource in RUN condition with error: Task in STOP condition" after "FS resource in RUN condition without error"

Once the fault has been rectified, the affected task can be restarted via one of the (re)start commands:

- ▶ FS hot start
- ▶ FS warm restart
- ▶ FS warm start
- ▶ One of the reset commands can also be used. A reset command impacts the whole PSSu system and not just the FS resource.

[6] Switching from "FS resource in RUN condition without error" to "FS resource in STOP condition with error"

If the FS resource is in a RUN condition without error, the user can explicitly cause the FS resource to switch to a STOP condition. The STOP condition will be maintained until the user explicitly restarts the FS resource.

It may be useful to trigger this change in operating status when commissioning plant sections, for example, or when making adjustments.

The STOP status of the FS resource can be triggered by one of the following options:

- ▶ Stopping the project (PAS4000)
- ▶ Stopping the PSSu system (PAS4000)
- ▶ Stopping the FS resource (PAS4000)

[7] Switching from "FS resource in STOP condition without error" to "FS resource in RUN condition with error"

This change in operating status can be brought about using one of the following commands:

- ▶ FS hot start (PAS4000)
- ▶ FS cold restart (PAS4000)
- ▶ FS warm restart (PAS4000)

[8] Reset in status "FS resource in STOP condition without error"

If one of the reset commands is executed in this operating status (see [8]), as the system sections are booted the FS resource will switch back to "FS resource in STOP condition without error" (see [1b]). A change in operating status to "FS resource in RUN condition without error" cannot be achieved using any of the reset commands.

[9] Restart in status "FS resource in RUN condition without error"

If it is necessary to set variables to defined values, during commissioning for example, one of the restart commands can be used (see Impact on local variables and resource global variables, [Page 339](#)).

This change in operating status can be triggered using one of the following commands:

- ▶ FS warm restart (PAS4000)
- ▶ FS cold restart (PAS4000)

[10] Switching from "FS resource in RUN condition without error" to "Booting system sections"

This change in operating status affects all the system sections of the PSSu system (see Flowchart, [Page 248](#)).

This change in operating status can be triggered using one of the following commands:

- ▶ Warm reset (PAS4000) or reset button (PSSu system)
- ▶ Cold reset (PAS4000)
- ▶ Warm reset of project (PAS4000)
- ▶ Cold reset of project (PAS4000)

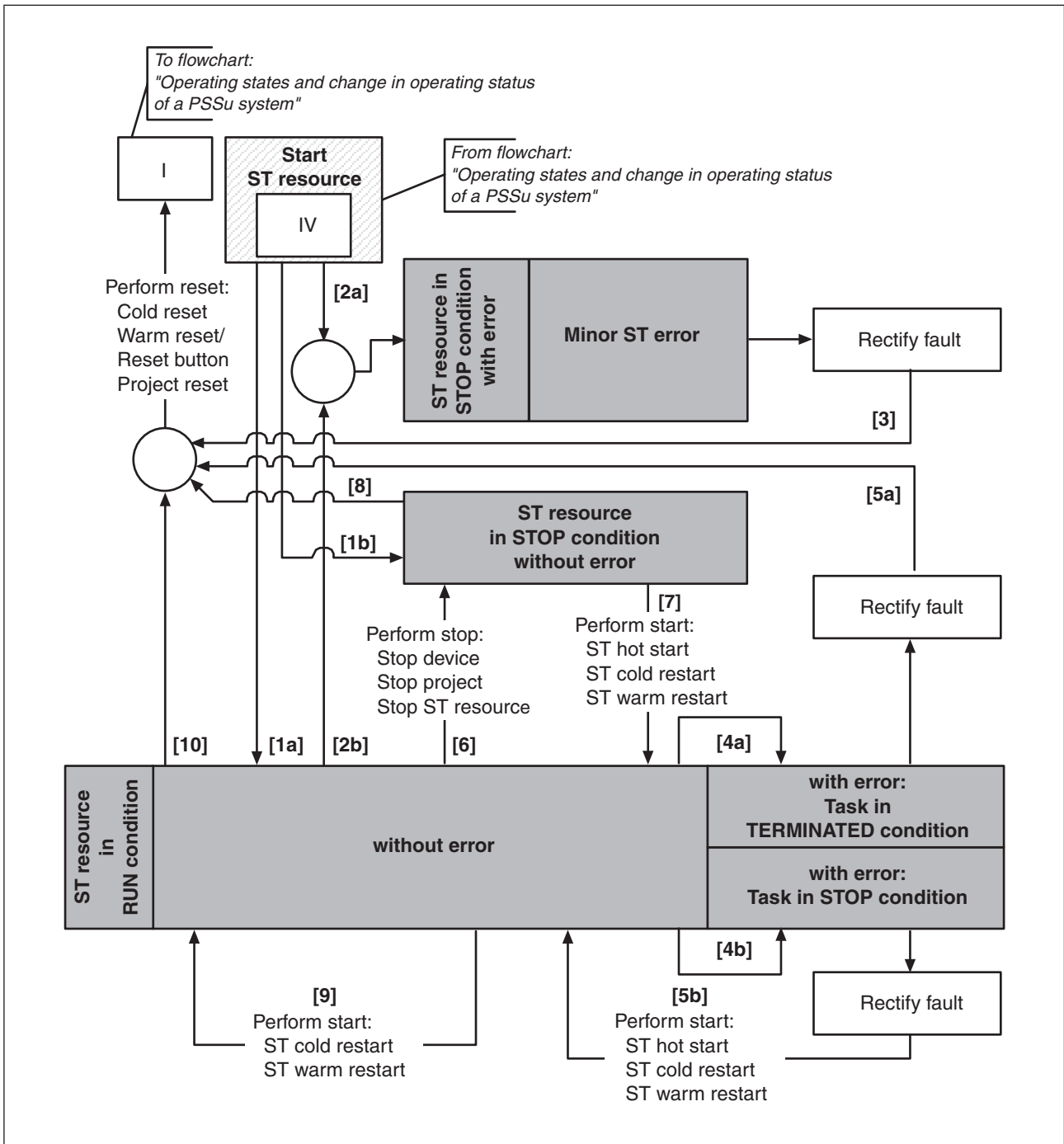
9.4 Operating states and change in operating status of the ST resource

The description below deals exclusively with the operating states of the ST resource, referring to the operating states of the PSSu system where necessary.

The diagram "Operating states and change in operating status of an ST resource" (see Flowchart, [Page 274](#)) provides an overview of the operating states (grey):

- ▶ The figures in square parentheses indicate the various changes in operating status.
- ▶ The section marked with the Roman numeral I refers to the diagram "Operating states and change in operating status of a PSSu system" (see Flowchart, [Page 248](#)), as the effects impact the whole PSSu system and not just the ST resource.
- ▶ The section marked with the Roman numeral IV refers to the diagram "Operating states and change in operating status of a PSSu system", (see Flowchart, [Page 248](#)) as the ST resource is started when booting the system sections of a PSSu system.

9.4.1 Flow chart



Operating states and change in operating status of the ST resource

9.4.2 Operating states of the ST resource

The description of the various operating states assumes that the PSSu system contains a valid device project.

The ST resource may assume the following operating states (see Flow chart, [Page 274](#)):

- ▶ Operating statuses "ST resource in RUN condition":
 - ST resource in RUN condition without error
 - ST resource in RUN condition with error: Task in TERMINATED condition
 - ST resource in RUN condition with error: Task in STOP condition
- ▶ Operating status: "ST resource in STOP condition without error"
- ▶ Operating status: "ST resource in STOP condition with error":
 - Minor ST error



Information

The description of each operating status includes a list of the options for triggering a change in operating status. These options are described separately (see Reset, restart, start and stop options, [Page 337](#)).

9.4.2.1 ST resource in RUN condition without error

This operating status is achieved through one of the following options:

- ▶ After a successful boot process, which was triggered by
 - Cold starting the PSSu system or a reboot command for the PSSu system via PAS4000
 - Pressing the reset button on the PSSu system
 - Triggering a reset via PAS4000
(cold reset, warm reset, project reset)
- ▶ After a start or restart via PAS4000
(ST cold restart, ST warm restart, ST hot start)

Effects

- ▶ In the operating status "ST resource in RUN condition without error", the ST resource tasks are run faultlessly.
- ▶ Status LEDs on the head module
The "ST RUN" LED lights up green.

9.4.2.2 ST resource in STOP condition without error

This operating status is achieved by triggering a stop via PAS4000 (PSSu system is stopped, ST resource is stopped, project is stopped).

Effects

- ▶ The behaviour of the hardware inputs, hardware outputs, process data and system sections while a resource is in a STOP condition is described separately (see Stop commands – Behaviour during the STOP condition, [Page 348](#)).
- ▶ The effects on local and resource-global variables are described separately (see Effects on local and resource-global variables, [Page 339](#)).
- ▶ Status LEDs on the head module
The ST-RUN LED will be off.

Remedy

- ▶ ST resource is (re-)started:
 - ST hot start (PAS4000)
 - ST cold restart (PAS4000)
 - ST warm restart (PAS4000)
- ▶ PSSu system is reset:
 - Warm reset (PAS4000) or reset button (PSSu system)
 - Cold reset (PAS4000)
- ▶ Project is reset
 - Warm reset of project (PAS4000)
 - Cold reset of project (PAS4000)

9.4.2.3 ST resource in RUN condition with error: Task in TERMINATED condition

If a fault results in this operating status, the affected task is ended as defined. A defined state is achieved.

Possible causes

(see also IEC 61131-3, Table E1)

- ▶ Execution time error
Task was not completed within the task cycle time

Effects

- ▶ The affected ST resource task is no longer executed, all other ST resource tasks continue to be executed.
- ▶ The ST resource continues to be in a RUN condition.
- ▶ The following applies for all O-PI variables that are declared as local variables or are referenced as VAR_EXTERNAL in the POU's of the task that has been stopped: Substitute values are used in the I/O-mapped data sinks (e.g. I-PI variables, hardware outputs) and Valid Bits = FALSE.
- ▶ The following applies for all O-PI variables that are created in Multi blocks as O-PI points or are referenced as VAR_EXTERNAL: Substitute values are used in the I/O-mapped data sinks (e.g. I-PI variables, hardware outputs) and Valid Bits = FALSE.
- ▶ The following applies for resource global variables (not O-PI variables) that are referenced via VAR_EXTERNAL in the blocks of the task that has been stopped: They retain the last valid, stored values and are not set to their initial values.
- ▶ If resource global variables are referenced via VAR_EXTERNAL in the POU's of a task that has not been stopped, the value there can be overwritten. As a result of this overwrite, resource global O-PI variables will again be marked as valid.
- ▶ Please refer to PAS4000's online help for details of how to handle variables (volatile variables, non-volatile variables)
- ▶ Status LEDs on the head module
The "ST RUN" LED flashes green

Remedy

1. Diagnostics
e.g. use PAS4000 to read the diagnostic list and/or the diagnostic log
2. Rectify errors
3. Perform reset:
Use a warm reset or cold reset

9.4.2.4 ST resource in RUN condition with error: Task in STOP condition

If a fault results in this operating status, the affected task is ended as defined. A defined state is achieved.

Possible causes

(see also IEC 61131-3, Table E1)

▶ Runtime error

For example, a variable value exceeds the defined range, the numeric result of a function exceeds the value range for a data type, division by 0

Effects

- ▶ The affected ST resource task is no longer executed, all other ST resource tasks continue to be executed.
- ▶ The ST resource continues to be in a RUN condition.
- ▶ The following applies for all O-PI variables that are declared as local variables or are referenced as VAR_EXTERNAL in the POU's of the task that has been stopped: Substitute values are used in the I/O-mapped data sinks (e.g. I-PI variables, hardware outputs) and Valid Bits = FALSE.
- ▶ The following applies for all O-PI variables that are created in Multi blocks as O-PI points or are referenced as VAR_EXTERNAL: Substitute values are used in the I/O-mapped data sinks (e.g. I-PI variables, hardware outputs) and Valid Bits = FALSE.
- ▶ The following applies for resource global variables (not O-PI variables) that are referenced via VAR_EXTERNAL in the blocks of the task that has been stopped: They retain the last valid, stored values and are not set to their initial values.
- ▶ If resource global variables are referenced via VAR_EXTERNAL in the POU's of a task that has not been stopped, the value there can be overwritten. As a result of this overwrite, resource global O-PI variables will again be marked as valid.
- ▶ Please refer to PAS4000's online help for details of how to handle variables (volatile variables, non-volatile variables)
- ▶ Status LEDs on the head module
The "ST RUN" LED flashes green

Remedy

1. Diagnostics
e.g. use PAS4000 to read the diagnostic list and/or the diagnostic log
2. Rectify errors
3. Perform a reset or (re)start
Use an ST hot start, ST warm restart or ST cold restart
One of the reset commands can also be used. A reset command impacts the whole PSSu system and not just the ST resource.

9.4.2.5 Minor ST error

If a fault results in this operating status, the affected sections of the ST resource are ended as defined. A defined state is achieved.

Possible causes

- ▶ Error in the memory of the non-volatile variables

Effects

- ▶ The affected ST resource task is no longer executed, all other ST resource tasks are stopped.
- ▶ The ST resource is in a STOP condition.
- ▶ For process data whose values are determined via this task, the substitute values are used and with PI variables also the corresponding Valid Bit = FALSE.
- ▶ Please refer to PAS4000's online help for details of how to handle variables (volatile variables, non-volatile variables)
- ▶ Status LEDs on the head module
The "ST RUN" LED will be off.

Remedy

1. Diagnostics
e.g. use PAS4000 to read the diagnostic list and/or the diagnostic log
2. Rectify errors
3. Perform reset

9.4.3 Change in operating status of the ST resource

This description of the change in operating status assumes that the PSSu system contains a valid device project. The figures refer to the change in operating status in the diagram "Operating states and change in operating status of the ST resource" (see Flowchart, [Page 274](#)).



Information

The description of each change of operating status includes a list of the options for triggering a change in operating status. These options are described separately (see Reset, restart, start and stop options, [Page 337](#)).

[1a] Switching from "Booting system sections" to "ST resource in RUN condition without error"

Booting system sections is one of the PSSu system's operating states (see Flowchart, [Page 248](#)). If no fault is detected in a safety-related system section as the system sections are booted, the ST resource is started, switching to the status "ST resource in RUN condition without error".

[1b] Switching from "Booting system sections" to "ST resource in STOP condition without error"

Booting system sections is one of the PSSu system's operating states (see Flowchart, [Page 248](#)). Under various circumstances, as the system sections are booted the ST resource may not achieve the operating status "ST resource in RUN condition without error", but switches instead to "ST resource in STOP condition without error":

- ▶ A manually executed STOP command (see [6]) causes the ST resource to be in the operating status "ST resource in STOP condition without error". Instead of one of the start commands (see [7]), one of the reset commands is executed (see [8]). However, in this case, the ST resource cannot achieve a RUN condition via any of the reset commands; instead it switches back to "ST resource in STOP condition without error".
- ▶ The ST resource is in the operating status "ST resource in RUN condition with error": Task in a STOP condition" After the fault has been rectified, one of the reset commands is executed instead of one of the start commands (see [5b]). However, in this case, the ST resource cannot achieve a RUN condition via any of the reset commands; instead it switches to "ST resource in STOP condition without error".

[2a] Switching from "Booting system sections" to error condition "Minor ST error"

If a fault is detected as the system sections are booted (see Flowchart, [Page 248](#)), the ST resource switches to the operating status "Minor ST error".

[2b] Switching from "ST resource in RUN condition without error" to error condition "Minor ST error"

If a fault is detected while the ST resource is in a RUN condition, the ST resource switches to the operating status "Minor ST error".

[3] Switching from "Minor ST error" to "Booting system sections"

Once the fault has been rectified, the ST resource can be restarted via one of the reset commands, provided an FS error is not present at the same time:

- ▶ Warm reset (PAS4000) or reset button (PSSu system)
- ▶ Cold reset (PAS4000)

- ▶ Warm reset of project (PAS4000)
- ▶ Cold reset of project (PAS4000)

[4a] Switching from "ST resource in RUN condition without error" to "ST resource in RUN condition with error": Task im TERMINATED-Zustand"

Wird in einem Task der ST-Ressource z. B. ein Ausführungszeitfehler festgestellt, wird der Task beendet und nicht mehr ausgeführt. Alle anderen Tasks der ST-Ressource werden weiter ausgeführt.

[4b] Switching from "ST resource in RUN condition without error" to "ST resource in RUN condition with error": Task im STOP-Zustand"

Wird in einem Task der ST-Ressource z. B. ein Laufzeitfehler festgestellt, wird der Task gestoppt und nicht mehr ausgeführt. Alle anderen Tasks der ST-Ressource werden weiter ausgeführt.

[5a] Switching from "ST resource in RUN condition with error: Task in TERMINATED condition" after "ST resource in RUN condition without error"

Once the fault has been rectified, the ST resource can be restarted via one of the reset commands:

- ▶ Warm reset (PAS4000) or reset button (PSSu system)
- ▶ Cold reset (PAS4000)
- ▶ Warm reset of project (PAS4000)
- ▶ Cold reset of project (PAS4000)

[5b] Switching from "ST resource in RUN condition with error: Task in STOP condition" to "ST resource in RUN condition without error"

Once the fault has been rectified, the affected task can be restarted via one of the (re)start commands:

- ▶ ST hot start
- ▶ ST warm restart
- ▶ ST warm start

One of the reset commands can also be used. A reset command impacts the whole PSSu system and not just the ST resource.

[6] Switching from "ST resource in RUN condition without error" to "ST resource in STOP condition without error"

If the ST resource is in a RUN condition without error, the user can explicitly cause the ST resource to switch to a STOP condition. The STOP condition will be maintained until the user explicitly restarts the ST resource.

It may be useful to trigger this change in operating status when commissioning plant sections, for example, or when making adjustments.

The STOP status of the ST resource can be triggered by one of the following commands:

- ▶ Stopping the project (PAS4000)
- ▶ Stopping the PSSu system (PAS4000)
- ▶ Stopping the ST resource (PAS4000)

[7] Switching from "ST resource in STOP condition without error" to "ST resource in RUN condition without error"

This change in operating status can be brought about using one of the following commands:

- ▶ ST hot start (PAS4000)
- ▶ ST cold restart (PAS4000)
- ▶ ST warm restart (PAS4000)

[8] Reset in status "ST resource in STOP condition without error"

If one of the reset commands is executed in this operating status (see [8]), as the system sections are booted the ST resource will switch back to "ST resource in STOP condition without error" (see [1b]). A change in operating status to "ST resource in RUN condition without error" cannot be achieved using any of the reset commands.

[9] Restart in status "ST resource in RUN condition without error"

If it is necessary to set variables to defined values, during commissioning for example, one of the restart commands can be used (see Impact on local variables and resource global variables, [Page 339](#)).

This change in operating status can be triggered using one of the following commands:

- ▶ ST warm restart (PAS4000)
- ▶ ST cold restart (PAS4000)

[10] Switching from "ST resource in RUN condition without error" to "Booting system sections"

This change in operating status affects all the system sections of the PSSu system (see Flowchart, [Page 248](#)).

This change in operating status can be triggered using one of the following commands:

- ▶ Warm reset (PAS4000) or reset button (PSSu system)
- ▶ Cold reset (PAS4000)
- ▶ Warm reset of project (PAS4000)
- ▶ Cold reset of project (PAS4000)

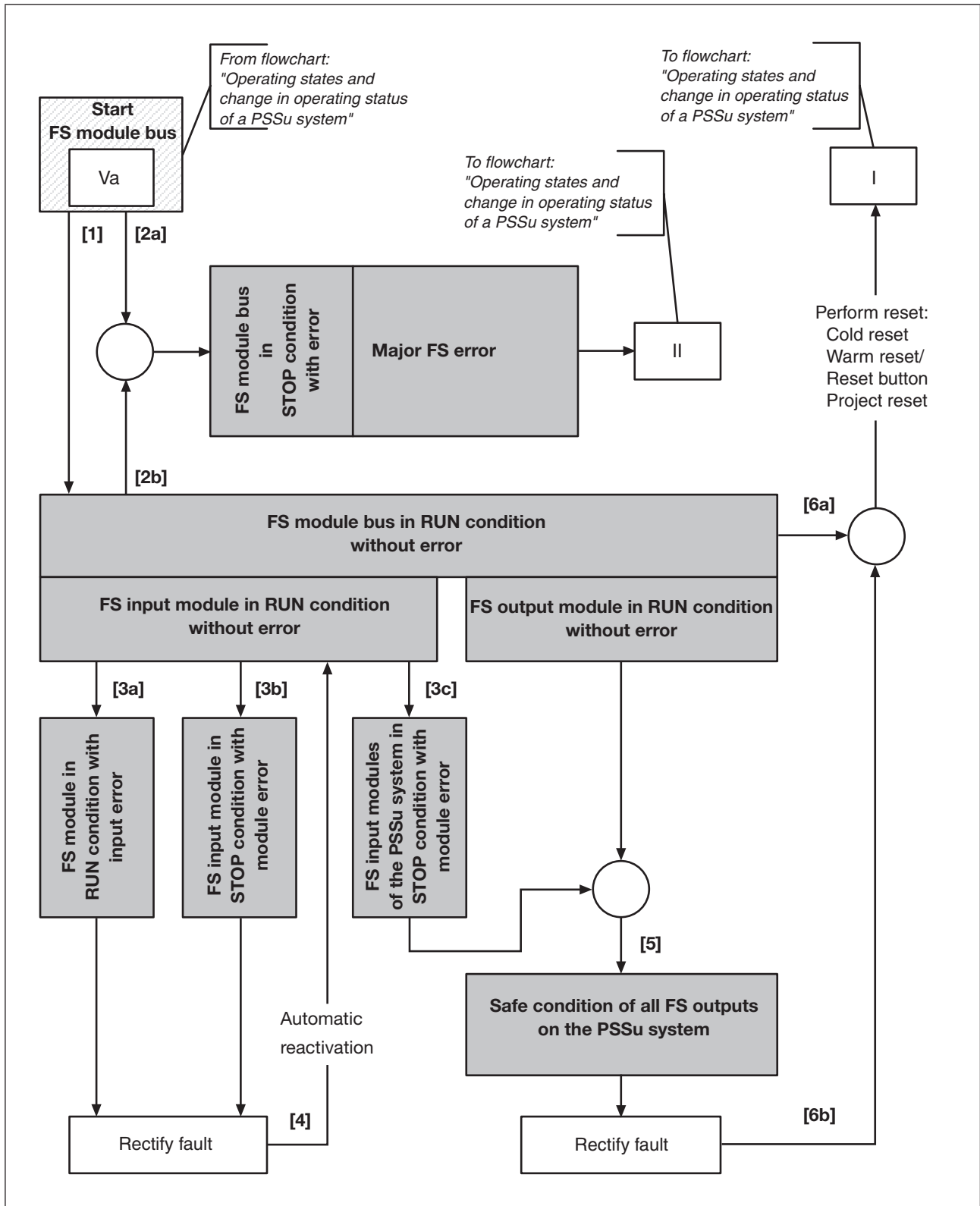
9.5 Operating states and change in operating status of the FS module bus

The description below deals exclusively with the operating states of the FS module bus, referring to the operating states of the PSSu system where necessary.

The diagram "Operating states and change in operating status of an FS module bus" (see Flowchart, [Page 285](#)) provides an overview of the operating states (grey):

- ▶ The figures in square parentheses indicate the various changes in operating status.
- ▶ The sections marked with the Roman numerals I and II refer to the diagram "Operating states and change in operating status of a PSSu system" (see Flowchart, [Page 248](#)), as the effects impact the whole PSSu system and not just the FS module bus.
- ▶ The section marked with the Roman numeral Va refers to the diagram "Operating states and change in operating status of a PSSu system" (see Flowchart, [Page 248](#)), as the FS module bus is started when booting the system sections of a PSSu system.

9.5.1 Flow chart



Operating states and change in operating status of the FS module bus

9.5.2 Operating states of the FS module bus

The description of the various operating states assumes that the PSSu system contains a valid device project.

The FS module bus may assume the following operating states (see Flow chart, [Page 285](#)):

- ▶ Operating states "FS module bus in RUN condition without error" with
 - "FS input module in RUN condition without error"
 - "FS input module in RUN condition with input error"
 - "FS input module in STOP condition with module error"
 - "FS input module of the PSSu system in STOP condition with module error"
 - "FS output module in RUN condition without error"
 - "Safe condition of all FS outputs on the PSSu system"
- ▶ Operating status "FS module bus in a STOP condition with error":
 - "Major FS error"



Information

The descriptions of various operating states include a list of the options for triggering a change in operating status. These options are described separately (see Reset, restart, start and stop options, [Page 337](#)).

9.5.2.1 FS module bus in RUN condition without error

This operating status is achieved through one of the following options:

- ▶ After a successful boot process, which was triggered by
 - Cold starting the PSSu system or a reboot command for the PSSu system via PAS4000
 - Pressing the reset button on the PSSu system
 - Triggering a reset via PAS4000
(cold reset, warm reset, project reset)

Effects

- ▶ The FS module bus is in RUN condition without error.
- ▶ The FS modules are operating faultlessly, i.e.
 - the input signals/input values present at the inputs are recorded correctly and transmitted faultlessly via the FS module bus
 - the outputs are activated correctly.
- ▶ Status LEDs on the head module

The "MBUS" LED lights up green, provided the ST module bus is also operating faultlessly.

9.5.2.2 FS input module in RUN condition without error

The operating status is achieved as soon as the module is ready for operation, irrespective of the module bus status. The FS module bus is where the FS input module provides the input data for data transport to the head module.

9.5.2.3 FS module in RUN condition with input error

An FS input error only influences the affected FS hardware input. The FS module bus remains in the operating state "FS module bus in RUN condition without error".

Possible causes

- ▶ Test pulse error
- ▶ Input error

Effects

- ▶ The substitute value is used for the input signal/input value, with Valid Bit = FALSE.
- ▶ The FS input module remains in RUN condition and the unaffected FS hardware inputs continue to be processed.

Remedy

1. Diagnostics
e.g. use PAS4000 to read the diagnostic list and/or the diagnostic log
2. Rectify errors
3. Once the error has been rectified, the FS hardware input is reactivated automatically.

9.5.2.4 FS input module in STOP condition with module error

A module fault only impacts the affected FS input module. The FS module bus remains in the operating state "FS module bus in RUN condition without error".

Possible causes

- ▶ Temperature error
- ▶ Voltage error (module supply)
- ▶ A module was disconnected during operation
- ▶ A defective module was detected during operation

Effects

- ▶ The substitute values are used for all the module's input signals/input values, with Valid Bits = FALSE.
- ▶ The unaffected FS input modules on the PSSu system continue to be processed.

Remedy

1. Diagnostics
e.g. use PAS4000 to read the diagnostic list and/or the diagnostic log
2. Rectify errors
3. Once the error has been rectified, the FS input module is reactivated automatically.

9.5.2.5 FS input modules of the PSSu system in STOP condition with module error

The FS input error affects all FS input modules and FS outputs of the PSSu system. The FS module bus remains in the operating state "FS module bus in RUN condition without error".

Possible causes

- ▶ The PSSu system detects an FS input error where the error location cannot be uniquely identified (e. g. PSSu E F AI U).

Effects

- ▶ The substitute values are used for all the PSSu system's input signals/input values, with Valid Bits = FALSE.
- ▶ All FS hardware outputs on the PSSu system switch to a safe condition.
- ▶ The substitute values are used for the module's FS outputs, with Valid Bits = FALSE.
- ▶ Status LEDs on the head module
 - The "MBUS" LED lights up red.

Remedy

1. Diagnostics
 - e.g. use PAS4000 to read the diagnostic list and/or the diagnostic log
2. Rectify errors
3. Perform reset
 - (warm reset/reset button, cold reset, project reset)

9.5.2.6 FS output module in RUN condition without error

The operating status is achieved as soon as the module is ready for operation, irrespective of the module bus status. The FS output module processes the output data provided by the FS module bus.

9.5.2.7 Safe condition of all FS outputs on the PSSu system

If an error results in this operating state, the effects of the error will be limited to the FS hardware outputs of the PSSu system on which it was detected. The FS module bus remains in the operating state "FS module bus in RUN condition without error".

Possible causes

- ▶ Output error
e.g. shorts across contacts, short circuit, error in the output test
- ▶ Temperature error
- ▶ Voltage error (module supply)
- ▶ A module was disconnected during operation
- ▶ A defective module was detected during operation
- ▶ Operating state "FS input module in STOP condition with module error"

Effects

- ▶ All FS hardware outputs on the PSSu system switch to a safe condition.
- ▶ The substitute values are used for the FS outputs of the modules.
- ▶ Status LEDs on the head module
The "MBUS" LED lights up red.

Remedy

1. Diagnostics
e.g. use PAS4000 to read the diagnostic list and/or the diagnostic log
2. Rectify errors
3. Perform reset
(warm reset/reset button, cold reset, project reset)

9.5.2.8 Major FS error

If a fault results in this operating status, the affected sections of the FS module bus are ended as defined. A defined state is achieved.

The effects of a major FS error impact all safety-related sections of a PSSu system (see Operating state "Major FS error of a PSSu system, [Page 259](#)).

9.5.3 Change in operating status of the FS module bus

This description of the change in operating status assumes that the PSSu system contains a valid device project. The figures refer to the change in operating status in the diagram "Operating states and change in operating status of the FS module bus" (see Flowchart, [Page 285](#)).



Information

The description of each change of operating status includes a list of the options for triggering a change in operating status. These options are described separately (see Reset, restart, start and stop options, [Page 337](#)).

[1] Switching from "Bootling system sections" to "FS module bus in RUN condition without error"

"Bootling system sections" is one of the PSSu system's operating states (see Flowchart, [Page 248](#)). If no fault is detected in a safety-related system section as the system sections are booted, the FS module bus is started, switching to the status "FS module bus in RUN condition without error".

[2] Switching from "Bootling system sections" to "Major FS error"

If a fault is detected in a safety-related system section as the system sections are booted, the FS module bus switches to the operating status "Major FS error".

The status "Bootling system sections" and the status "Major FS error" (see Flowchart, [Page 248](#)) are both operating states that impact the PSSu system.

[2b] Switching from "FS module bus in RUN condition without error" to "Major FS error"

If a fault is detected in a safety-related system section while the FS module bus is in a RUN condition, the FS module bus switches to the operating status "Major FS error".

The operating status "Major FS error" (see Flowchart, [Page 248](#)) is an operating status that impacts the PSSu system.

[3a] Switching from "FS input module in RUN condition without error" to "FS module in RUN condition with input error"

If an input error is detected on an FS hardware input, the FS input module switches to the operating status "FS module in RUN condition with input error".

[3b] Switching from "FS input module in RUN condition without error" to "FS input module in STOP condition with module error"

If a fault is detected on an FS input module, the relevant FS input module on the PSSu system switches to the operating status "FS input module in STOP condition with module error".

[3c] Switching from "FS input module in RUN condition without error" to "FS input modules on the PSSu system in STOP condition with module error"

If an FS input error is detected and the PSSu system cannot assign it uniquely to one of its FS hardware inputs, all the FS input modules on the PSSu system switch to the operating status "FS input module in STOP condition with module error" and all the FS outputs on the PSSu system switch to the operating status "Safe condition of all FS outputs on the PSSu system" (see [5]).

[4] Switching from one of the operating states "FS input module in RUN condition with input error" or**"FS input module in STOP condition with module error" to "FS input module in RUN condition without error"**

If the relevant fault (input error, module error) is rectified, the affected section on the FS input module is reactivated automatically and the module switches to the operating status "FS input module in RUN condition without error".

[5] Switching from "FS output module in RUN condition" to "Safe condition of all FS outputs on the PSSu system"

If a fault is detected on an FS output module or FS output, or if the FS input modules on the PSSu system switch to the operating status "FS input modules on the PSSu system in STOP condition with module error", all FS outputs on the PSSu system will switch to a safe condition.

[6a] Switching from "FS module bus in RUN condition without error" to "Boot- ing system sections"

This change in operating status affects all the system sections of the PSSu system (see Flowchart, [Page 248](#)).

This change in operating status can be triggered using one of the following commands:

- ▶ Warm reset (PAS4000) or reset button (PSSu system)
- ▶ Cold reset (PAS4000)
- ▶ Warm reset of project (PAS4000)
- ▶ Cold reset of project (PAS4000)

[6b] Switching from "Safe condition of all FS outputs on the PSSu system" to "Boot- ing system sections"

This change in operating status affects all the system sections of the PSSu system (see Flowchart, [Page 248](#)).

Once the fault has been rectified, the change in operating status can be triggered using one of the following commands:

- ▶ Warm reset (PAS4000) or reset button (PSSu system)
- ▶ Cold reset (PAS4000)
- ▶ Warm reset of project (PAS4000)
- ▶ Cold reset of project (PAS4000)

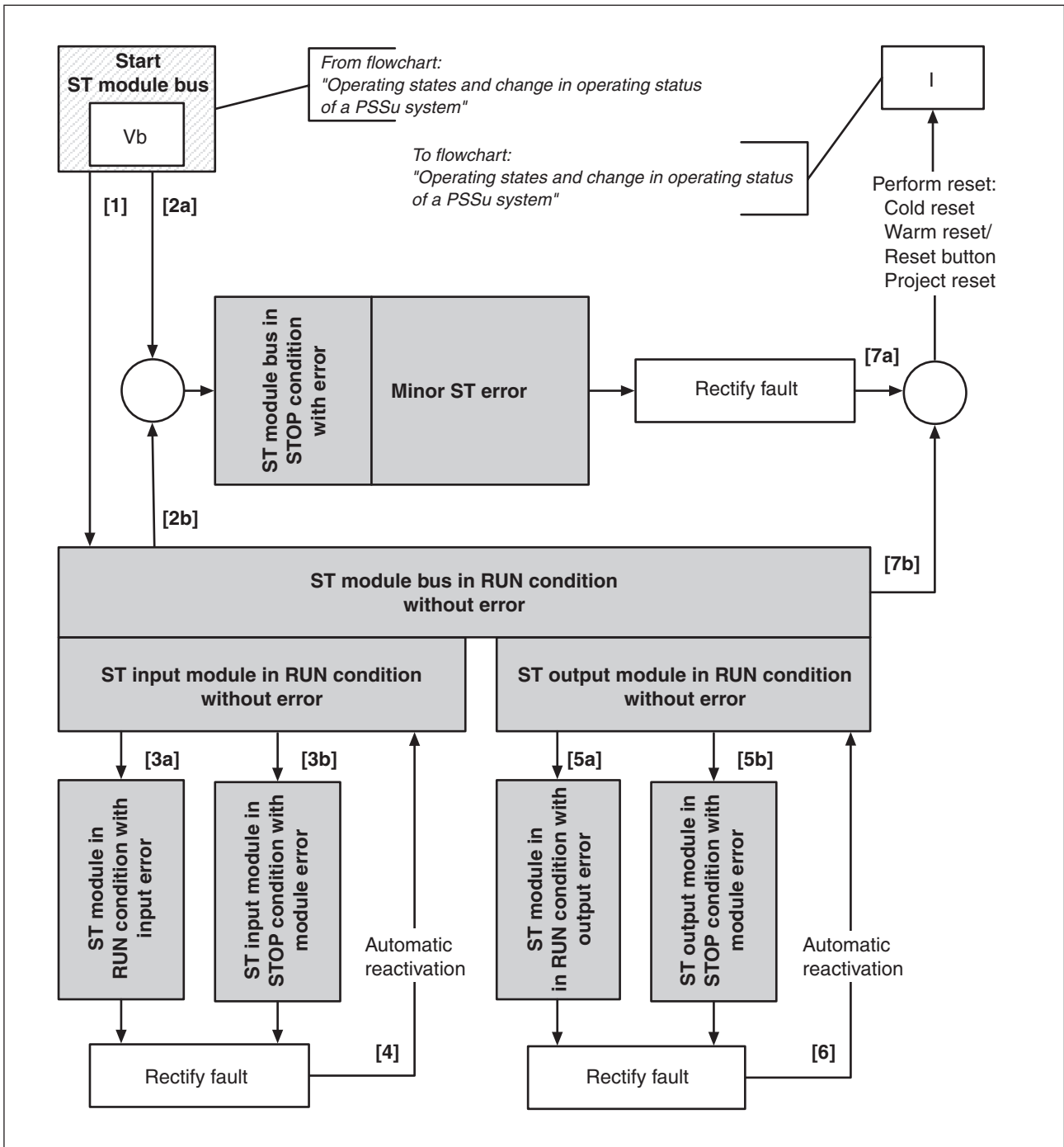
9.6 Operating states and change in operating status of the ST module bus

The description below deals exclusively with the operating states of the ST module bus, referring to the operating states of the PSSu system where necessary.

The diagram "Operating states and change in operating status of an ST module bus" (see Flowchart, [Page 298](#)) provides an overview of the operating states (grey):

- ▶ The figures in square parentheses indicate the various changes in operating status.
- ▶ The sections marked with the Roman numerals I and II refer to the diagram "Operating states and change in operating status of a PSSu system" (see Flowchart, [Page 248](#)), as the effects impact the whole PSSu system and not just the ST module bus.
- ▶ The section marked with the Roman numeral Vb refers to the diagram "Operating states and change in operating status of a PSSu system" (see Flowchart, [Page 248](#)), as the ST module bus is started when booting the system sections of a PSSu system.

9.6.1 Flow chart



Operating states and change in operating status of the ST module bus

9.6.2 Operating states of the ST module bus

The description of the various operating states assumes that the PSSu system contains a valid device project.

The ST module bus may assume the following operating states (see Flow chart, [Page 298](#)):

- ▶ Operating states "ST module bus in RUN condition without error" with
 - "ST input module in RUN condition without error"
 - "ST module in RUN condition with input error"
 - "ST input module in STOP condition with module error"
 - "ST output module in RUN condition without error"
 - "ST module in RUN condition with output error"
 - "ST output module in STOP condition with module error"
- ▶ Operating status "ST module bus in STOP condition with error":
 - "Minor ST error"



Information

The descriptions of various operating states include a list of the options for triggering a change in operating status. These options are described separately (see Reset, restart, start and stop options, [Page 337](#)).

9.6.2.1 ST module bus in RUN condition without error

This operating status is achieved through one of the following options:

- ▶ After a successful boot process, which was triggered by
 - Cold starting the PSSu system or a reboot command for the PSSu system via PAS4000
 - Pressing the reset button on the PSSu system
 - Triggering a reset via PAS4000
(cold reset, warm reset, project reset)

Effects

- ▶ The ST module bus is in RUN condition without error.
- ▶ The ST modules are operating faultlessly, i.e.
 - the input signals/input values present at the inputs are recorded correctly and transmitted faultlessly via the ST module bus
 - the outputs are activated correctly.
- ▶ Status LEDs on the head module

The "MBUS" LED lights up green, provided the FS module bus is also operating faultlessly.

9.6.2.2 ST input module in RUN condition without error

The operating status is achieved as soon as the module is ready for operation, irrespective of the module bus status. The ST module bus is where the ST input module provides the input data for data transport to the head module.

9.6.2.3 ST module in RUN condition with input error

An input error only influences the affected ST hardware input. The ST module bus remains in the operating status "ST module bus in RUN condition without error".

Possible causes

- ▶ Input error

Effects

- ▶ The substitute value is used for the input signal/input value, with Valid Bit = FALSE.
- ▶ The ST input module remains in RUN condition and the unaffected ST hardware inputs continue to be processed.

Remedy

1. Diagnostics
e.g. use PAS4000 to read the diagnostic list and/or the diagnostic log
2. Rectify errors
3. Once the fault has been rectified, the ST hardware input is reactivated automatically.

9.6.2.4 ST input module in STOP condition with module error

A module fault only impacts the affected ST input module. The ST module bus remains in the operating status "ST module bus in RUN condition without error".

Possible causes

- ▶ Temperature error
- ▶ Voltage error (module supply)
- ▶ A module was disconnected during operation
- ▶ A defective module was detected during operation

Effects

- ▶ The substitute values are used for all the module's input signals/input values, with Valid Bits = FALSE.
- ▶ The unaffected ST input modules on the PSSu system continue to be processed.

Remedy

1. Diagnostics
e.g. use PAS4000 to read the diagnostic list and/or the diagnostic log
2. Rectify errors
3. Once the error has been rectified, the ST input module is reactivated automatically.

9.6.2.5 ST output module in RUN condition without error

This operating status is achieved as soon as the module is ready for operation, irrespective of the module bus status. The ST output module processes the output data provided by the ST module bus.

9.6.2.6 ST module in RUN condition with output error

An ST output error only influences the affected ST hardware output. The ST module bus remains in the operating status "ST module bus in RUN condition without error".

Possible causes

- ▶ Output error
(e.g. overload, short circuit)

Effects

- ▶ The substitute value is used for the output signal/output value, with Valid Bit = FALSE.
- ▶ The ST output module remains in RUN condition and the unaffected ST hardware outputs continue to be processed.

Remedy

1. Diagnostics
e.g. use PAS4000 to read the diagnostic list and/or the diagnostic log
2. Rectify errors
3. Once the fault has been rectified, the ST hardware output is reactivated automatically.

9.6.2.7 ST output module in STOP condition with module error

A module fault in an ST output module only impacts the affected ST output module. The ST module bus remains in the operating status "ST module bus in RUN condition without error".

Possible causes

- ▶ Temperature error
- ▶ Voltage error (module supply)
- ▶ A module was disconnected during operation
- ▶ A defective module was detected during operation

Effects

- ▶ The substitute values are used for all the module's output signals/output values, with Valid Bits = FALSE.
- ▶ The unaffected ST output modules on the PSSu system continue to be processed.
- ▶ Status LEDs on the head module
The "MBUS" LED lights up red.

Remedy

1. Diagnostics
e.g. use PAS4000 to read the diagnostic list and/or the diagnostic log
2. Rectify errors
3. Once the error has been rectified, the ST output module is reactivated automatically.

9.6.2.8 Minor ST error

If a fault results in this operating status, the affected sections of the ST module bus are ended as defined. A defined state is achieved.

Possible causes

- ▶ Configuration error
(e.g. differences between the set/actual hardware registry)
- ▶ Defective ST module detected when starting the module bus

Effects

- ▶ If the fault is detected when starting the ST module bus, the ST outputs are not activated and the ST inputs are not read.
- ▶ If other PSSu systems access the ST I/O data on this PSSu system, the substitute values are used for all signals/values, with Valid Bits = FALSE.

Remedy

1. Diagnostics
e.g. use PAS4000 to read the diagnostic list and/or the diagnostic log
2. Rectify errors
3. Once the fault has been rectified, the PSSu system must be restarted via one of the re-set commands.

9.6.3 Change in operating status of the ST module bus

This description of the change in operating status assumes that the PSSu system contains a valid device project. The figures refer to the change in operating status in the diagram "Operating states and change in operating status of the ST module bus" (see Diagram, [Page 298](#)).



Information

The description of each change of operating status includes a list of the options for triggering a change in operating status. These options are described separately (see Reset, restart, start and stop options, [Page 337](#)).

[1] Switching from "Booting system sections" to "ST module bus in RUN condition without error"

"Booting system sections" is one of the PSSu system's operating states (see Diagram, [Page 248](#)). If no fault is detected as the system sections are booted, the ST module bus is started, switching to the status "ST module bus in RUN condition without error".

[2a] Switching from "Booting system sections" to "Minor ST error"

If a fault is detected as the system sections are booted, the ST module bus switches to the operating status "Minor ST error".

The operating status "Minor ST error" is an operating status that only impacts the ST module bus of a PSSu system.

[2b] Switching from "ST module bus in RUN condition without error" to "Minor ST error"

If a fault is detected while the ST module bus is in a RUN condition, the ST module bus switches to the operating status "Minor ST error".

The operating status "Minor ST error" is an operating status that only impacts the ST module bus of a PSSu system.

[3a] Switching from "ST input module in RUN condition without error" to "ST input module in RUN condition with input error"

If an input error is detected on an ST hardware input, the ST input module switches to the operating status "ST module in RUN condition with input error".

[3b] Switching from "ST input module in RUN condition without error" to "ST input module in STOP condition with module error"

If a module error is detected on an ST input module, the ST input module switches to the operating status "ST input module in STOP condition with module error".

[4] Switching from one of the operating states "ST input module in RUN condition with input error" or

"ST input module in STOP condition with module error" to "ST input module in RUN condition without error"

If the relevant fault (input error, module error) is rectified, the affected section on the ST input module is reactivated automatically and the module switches to the operating status "ST input module in RUN condition without error".

[5a] Switching from "ST output module in RUN condition without error" to "ST output module in RUN condition with output error"

If an output error is detected on an ST hardware output, the ST output module switches to the operating status "ST module in RUN condition with output error".

[5b] Switching from "ST output module in RUN condition without error" to "ST output module in STOP condition with module error"

If a module error is detected on an ST output module, the ST output module switches to the operating status "ST output module in STOP condition with module error".

[6] Switching from one of the operating states "ST output module in RUN condition with output error" or "ST output module in STOP condition with module error" to "ST output module in RUN condition without error"

If the relevant fault (output error, module error) is rectified, the affected section on the ST output module is reactivated automatically and the module switches to the operating status "ST output module in RUN condition without error".

[7a] Switching from "Minor ST error" to "Bootling system sections"

This change in operating status affects all the system sections of the PSSu system (see Fig. 8-3 in the section entitled "Operating states and change in operating status of a PSSu system"). Once the fault has been rectified, the ST module bus can be restarted via one of the reset commands:

- ▶ Warm reset (PAS4000) or reset button (PSSu system)
- ▶ Cold reset (PAS4000)
- ▶ Warm reset of project (PAS4000)
- ▶ Cold reset of project (PAS4000)

[7b] Switching from "ST module bus in RUN condition without error" to "Bootling system sections"

This change in operating status affects all the system sections of the PSSu system (see Diagram, [Page 248](#)).

This change in operating status can be triggered using one of the following commands:

- ▶ Warm reset (PAS4000) or reset button (PSSu system)
- ▶ Cold reset (PAS4000)
- ▶ Warm reset of project (PAS4000)
- ▶ Cold reset of project (PAS4000)

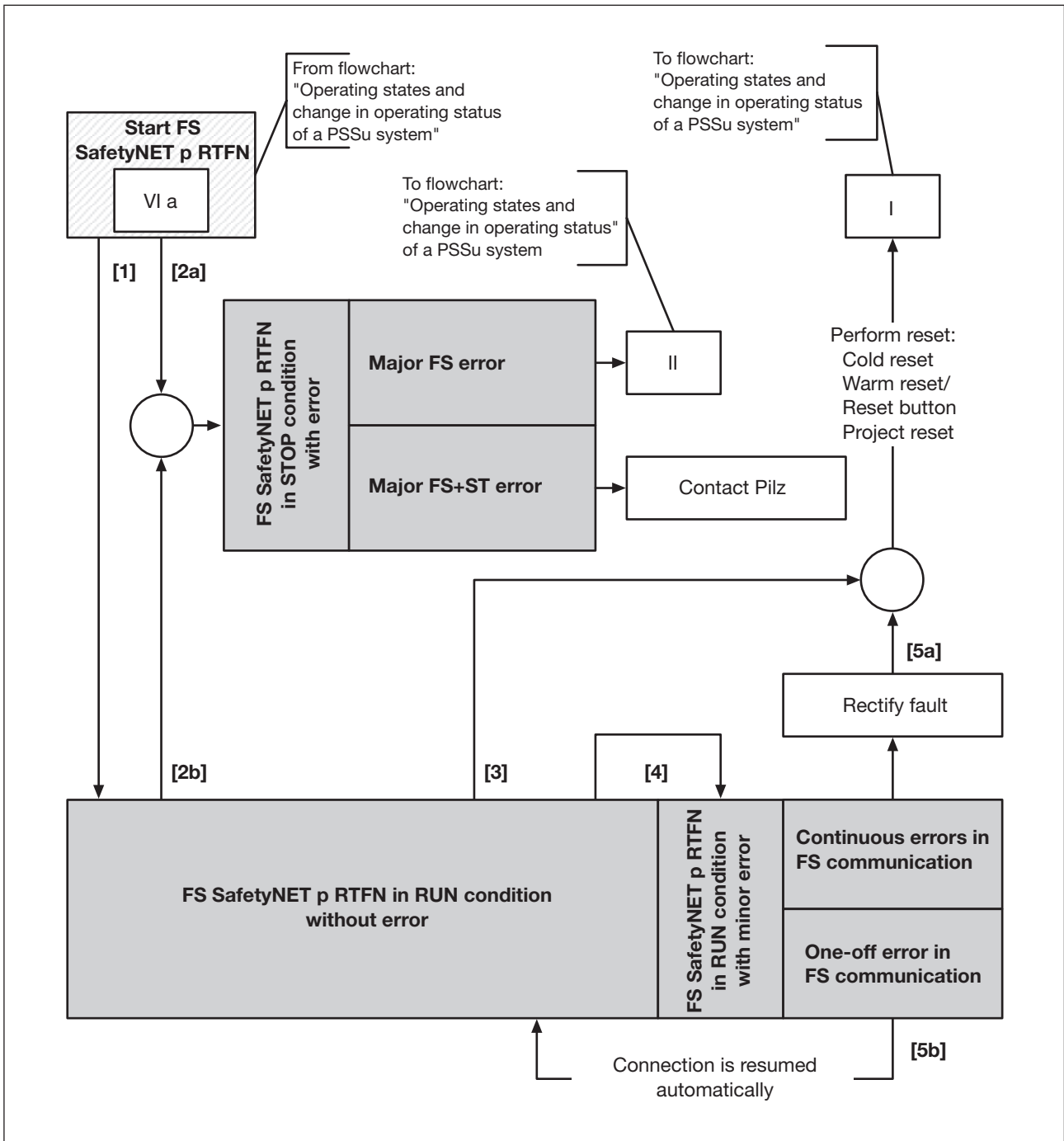
9.7 Operating states and change in operating status of the FS SafetyNET p RTFN

The description below deals exclusively with the operating states of the system section FS SafetyNET p RTFN, referring to the operating states of the PSSu system where necessary.

The diagram "Operating states and change in operating status of FS SafetyNET p RTFN" (see Flowchart, [Page 311](#)) provides an overview of the operating states (grey):

- ▶ The figures in square parentheses indicate the various changes in operating status.
- ▶ The sections marked with the Roman numerals I and II refer to the diagram "Operating states and change in operating status of a PSSu system" (see Flowchart, [Page 248](#)), as the effects impact the whole PSSu system and not just the system section FS SafetyNET p RTFN.
- ▶ The section marked with the Roman numeral VIa refers to the diagram "Operating states and change in operating status of a PSSu system" (see Flowchart, [Page 248](#)), as the system section FS SafetyNET p RTFN is started when booting the system sections of a PSSu system.

9.7.1 Flow chart



Operating states and change in operating status of the FS SafetyNET p RTFN

9.7.2 Operating states of FS SafetyNET p RTFN

The description of the various operating states assumes that the PSSu system contains a valid device project.

The system section FS SafetyNET p RTFN may assume the following operating states (see Flowchart, [Page 311](#)):

- ▶ Operating status "FS SafetyNET p RTFN in RUN condition without error"
- ▶ Operating status "FS SafetyNET p RTFN in STOP condition with error":
 - "Major FS error"
 - "Major FS+ST error"
- ▶ Operating status "FS SafetyNET p RTFN in RUN condition with minor error"
 - "Continuous errors in FS communication"
 - "One-off error in FS communication"



Information

The description of each operating status includes a list of the options for triggering a change in operating status. These options are described separately (see Reset, restart, start and stop options, [Page 337](#)).

9.7.2.1 FS SafetyNET p RTFN in RUN condition without error

This operating status is achieved through one of the following options:

- ▶ After a successful boot process, which was triggered by
 - Cold starting the PSSu system or a reboot command for the PSSu system via PAS4000
 - Pressing the reset button on the PSSu system
 - Triggering a reset via PAS4000
(cold reset, warm reset, project reset)

Effects

- ▶ In the operating status "FS SafetyNET p RTFN in RUN condition without error", pending FS process data can be transmitted faultlessly via SafetyNET p.
- ▶ Status LEDs on the head module
The "FS SNp" LED lights up green.

9.7.2.2 Major FS error

In the event of a fault that leads to this operating status, the affected parts of the system section FS SafetyNET p RTFN are ended as defined. A defined status is achieved. The effects of a major FS fault impact all safety-related sections of a PSSu system. (see Operating state "Major FS error of a PSSu system, [Page 259](#)).

9.7.2.3 Major FS+ST error

In the event of a fault that leads to the operating status "Major FS+ST error", the affected sections of FS SafetyNET p RTFN are ended as defined. A defined status is achieved.

Possible causes

- ▶ Internal fault

Effects

- ▶ The system section FS SafetyNET p RTFN and the system section ST SafetyNET p RTFN both switch to a STOP condition. It will no longer be possible to exchange process data with the PSSu system
- ▶ The substitute values are used for the affected process data, with Valid Bits = FALSE.
- ▶ Status LEDs on the head module
The "FS SNp" and "ST SNp" LEDs flash red.

Remedy

Contact Pilz

9.7.2.4 FS SafetyNET p RTFN in RUN condition with minor error

If a fault results in a minor error in FS communication, the affected section is stopped or ended as defined. A defined status is achieved.

Possible causes

- ▶ Continuous errors in FS communication
e.g. a PSSu system cannot be reached temporarily because it has been removed from SafetyNET p; the bus connector on a PSSu system has been unplugged
- ▶ One-off error in FS communication
e.g. transient, one-off error on the bus cable

Effects

- ▶ FS process data is no longer exchanged with the affected PSSu system; FS process data continues to be exchanged with all other PSSu systems.
- ▶ The system section FS SafetyNET p RTFN continues to be in a RUN condition
- ▶ The substitute values are used for the affected process data, with Valid Bits = FALSE.
- ▶ Status LEDs on the head module
"FS SNp" LED flashes green.

Remedy

1. Diagnostics
e.g. use PAS4000 to read the diagnostic list and/or the diagnostic log
2. Rectify errors
3. Reset or resume the connection automatically
 - Continuous errors in FS communication:
Perform reset (warm reset/reset button, cold reset, project reset)
 - One-off error in FS communication:
The connection is resumed automatically. Once the connection is restored, the process values are output at the outputs.

9.7.3 Change in operating status of FS SafetyNET p RTFN

This description of the change in operating status assumes that the PSSu system contains a valid device project. The figures refer to the change in operating status in the diagram "Operating states and change in operating status of FS SafetyNET p RTFN" (see Flowchart, [Page 311](#)).



Information

The description of each change of operating status includes a list of the options for triggering a change in operating status. These options are described separately (see Reset, restart, start and stop options, [Page 337](#)).

[1] Switching from "Booting system sections" to "FS SafetyNET p RTFN in RUN condition without error"

"Booting system sections" is one of the PSSu system's operating states (see Flowchart, [Page 248](#)). If no fault is detected in a safety-related system section as the system sections are booted, the system section FS SafetyNET p RTFN is started, switching to the status "FS SafetyNET p RTFN in RUN condition without error".

[2a] Switching from "Booting system sections" to an error condition

If a fault is detected in a safety-related system section as the system sections are booted (see Flowchart, [Page 248](#)), FS SafetyNET p RTFN switches to a STOP condition, achieving the operating status "Major FS error" or "Major FS+ST error", depending on the fault.

[2b] Switching from "FS SafetyNET p RTFN in RUN condition without error" to an error condition

If a fault is detected in a safety-related system section while FS SafetyNET p RTFN is in a RUN condition, FS SafetyNET p RTFN switches to a STOP condition, achieving the operating status "Major FS error" or "Major FS+ST error", depending on the fault.

[3] Switching from "FS SafetyNET p RTFN in RUN condition without error" to "Booting system sections"

This change in operating status affects all the system sections of the PSSu system (see Flowchart, [Page 248](#)).

This change in operating status can be triggered using one of the following commands:

- ▶ Warm reset (PAS4000) or reset button (PSSu system)
- ▶ Cold reset (PAS4000)
- ▶ Warm reset of project (PAS4000)
- ▶ Cold reset of project (PAS4000)

[4] Switching from "FS SafetyNET p RTFN in RUN condition without error" to "FS SafetyNET p RTFN in RUN condition with minor error"

The change in operating status has resulted due to an FS communication error. Faults that result in this error are classified as follows:

- ▶ Continuous error in FS communication
- ▶ One-off error in FS communication.

[5a] Switching from "FS SafetyNET p RTFN in RUN condition with error" to "FS SafetyNET p RTFN in RUN condition without error"

The operating status "FS SafetyNET p RTFN in RUN condition with error" has resulted due to a continuous error in FS communication. Once the fault has been rectified, the system section can be restarted via one of the reset commands:

- ▶ Warm reset (PAS4000) or reset button (PSSu system)
- ▶ Cold reset (PAS4000)
- ▶ Warm reset of project (PAS4000)
- ▶ Cold reset of project (PAS4000)

[5b] Switching from "FS SafetyNET p RTFN in RUN condition with error" to "FS SafetyNET p RTFN in RUN condition without error"

The operating status "FS SafetyNET p RTFN in RUN condition with error" has resulted due to a transient, one-off error in FS communication. The connection will be resumed automatically.

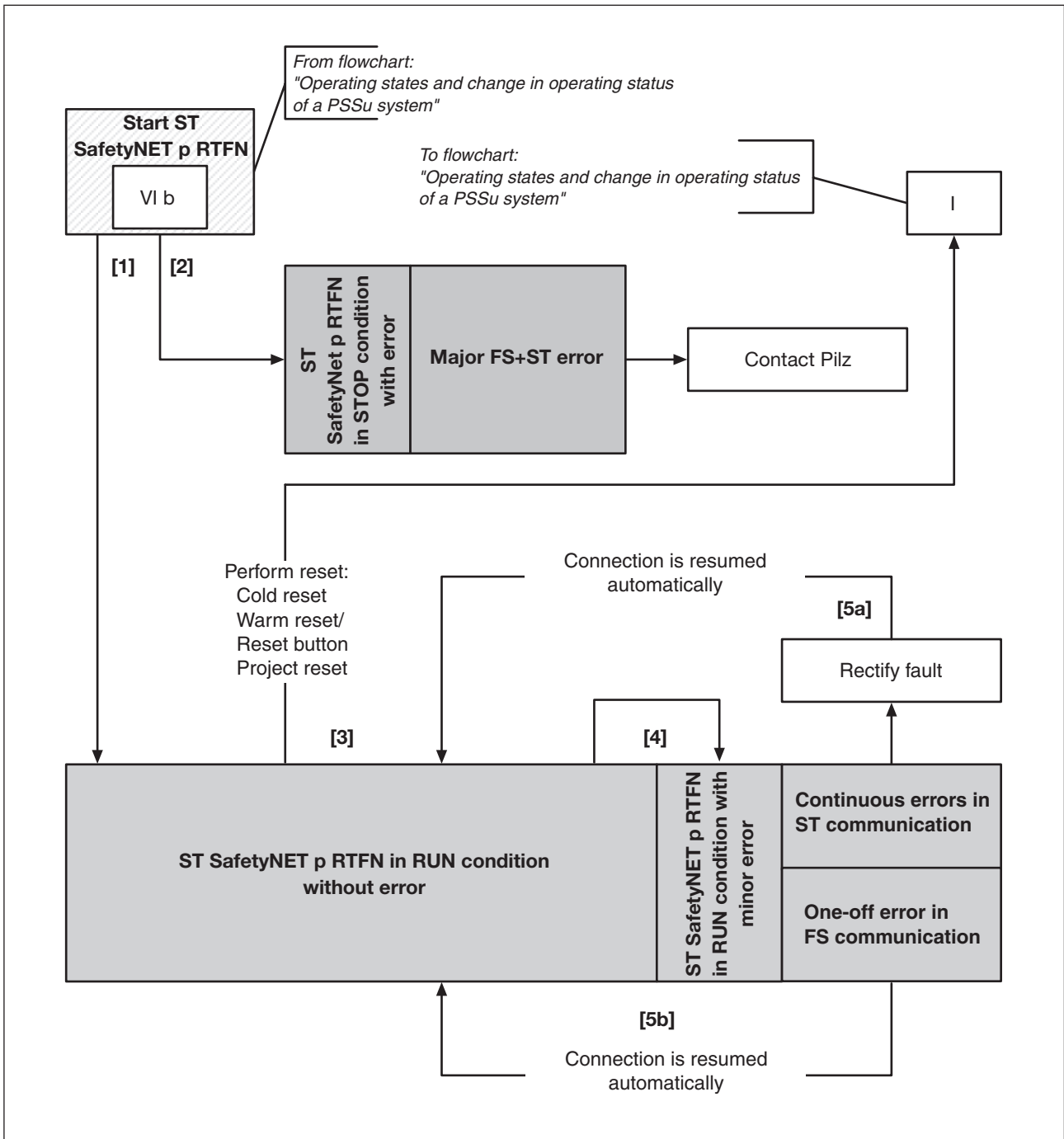
9.8 Operating states and change in operating status of the ST SafetyNET p RTFN

The description below deals exclusively with the operating states of the system section ST SafetyNET p RTFN, referring to the operating states of the PSSu system where necessary.

The diagram "Operating states and change in operating status of ST SafetyNET p RTFN" (see Flowchart, [Page 320](#)) provides an overview of the operating states (grey):

- ▶ The figures in square parentheses indicate the various changes in operating status.
- ▶ The sections marked with the Roman numerals I and II refer to the diagram "Operating states and change in operating status of a PSSu system" (see Flowchart, [Page 248](#)), as the effects impact the whole PSSu system and not just the system section ST SafetyNET p RTFN.
- ▶ The section marked with the Roman numeral VIb refers to the diagram "Operating states and change in operating status of a PSSu system" (see Flowchart, [Page 248](#)), as the system section ST SafetyNET p RTFN is started when booting the system sections of a PSSu system.

9.8.1 Flow chart



Operating states and change in operating status of the ST SafetyNET p RTFN

9.8.2 Operating states of ST SafetyNET p RTFN

The description of the various operating states assumes that the PSSu system contains a valid device project.

The system section ST SafetyNET p RTFN may assume the following operating states (see Flowchart, [Page 320](#)):

- ▶ Operating status "ST SafetyNET p RTFN in RUN condition without error"
- ▶ Operating status "ST SafetyNET p RTFN in STOP condition with error":
 - "Major FS+ST error"
- ▶ Operating status "ST SafetyNET p RTFN in RUN condition with minor error":
 - "Continuous errors in ST communication"
 - "One-off error in ST communication"



Information

The description of each operating status includes a list of the options for triggering a change in operating status. These options are described separately (see Reset, restart, start and stop options, [Page 337](#)).

9.8.2.1 ST SafetyNET p RTFN in RUN condition without error

This operating status is achieved through one of the following options:

- ▶ After a successful boot process, which was triggered by
 - Cold starting the PSSu system or a reboot command for the PSSu system via PAS4000
 - Pressing the reset button on the PSSu system
 - Triggering a reset via PAS4000 (cold reset, warm reset, project reset)

Effects

- ▶ In the operating status "ST SafetyNET p RTFN in RUN condition without error", pending ST process data can be transmitted faultlessly via SafetyNET p.
- ▶ Status LEDs on the head module
The "ST SNp" LED lights up green.

9.8.2.2 Major FS+ST error

In the event of a fault that leads to the operating status "Major FS+ST error", the affected sections of FS SafetyNET p RTFN are ended as defined. A defined status is achieved.

Possible causes

- ▶ Internal fault

Effects

- ▶ The system section FS SafetyNET p RTFN and the system section ST SafetyNET p RTFN both switch to a STOP condition. It will no longer be possible to exchange process data with the PSSu system
- ▶ The substitute values are used for the affected process data, with Valid Bits = FALSE.
- ▶ Status LEDs on the head module
The "FS SNp" and "ST SNp" LEDs flash red.

Remedy

Contact Pilz

9.8.2.3 ST SafetyNET p RTFN in RUN condition with minor error

In the event of a fault that leads to a minor error in ST communication, the affected section is stopped or ended as defined. A defined status is achieved.

Possible causes

- ▶ Continuous errors in ST communication
e.g. a PSSu system cannot be reached temporarily because it has been removed from SafetyNET p; the bus connector on a PSSu system has been unplugged
- ▶ One-off error in ST communication
e.g. transient, one-off error on the bus cable

Effects

- ▶ ST process data is no longer exchanged with the affected PSSu system; ST process data continues to be exchanged with all other PSSu systems.
- ▶ The system section ST SafetyNET p RTFN continues to be in a RUN condition
- ▶ The substitute values are used for the affected process data, with Valid Bits = FALSE.
- ▶ Status LEDs on the head module
The "FS Snp" LED flashes green.

Remedy

1. Diagnostics
e.g. use PAS4000 to read the diagnostic list and/or the diagnostic log
2. Rectify fault
3. Resume the connection
 - Continuous errors in ST communication:
The connection is resumed automatically.
 - One-off error in ST communication:
The connection is resumed automatically.

9.8.3 Change in operating status of ST SafetyNET p RTFN

This description of the change in operating status assumes that the PSSu system contains a valid device project. The figures refer to the change in operating status in the diagram "Operating states and change in operating status of ST SafetyNET p RTFN" (see Flowchart, [Page 320](#)).



Information

The description of each change of operating status includes a list of the options for triggering a change in operating status. These options are described separately (see Reset, restart, start and stop options, [Page 337](#)).

[1] Switching from "Booting system sections" to "ST SafetyNET p RTFN in RUN condition without error"

"Booting system sections" is one of the PSSu system's operating states (see Flowchart, [Page 248](#)). If no fault is detected as the system sections are booted, the system section ST SafetyNET p RTFN is started, switching to the status "ST SafetyNET p RTFN in RUN condition without error".

[2] Switching from "Booting system sections" to "Major FS+ST error"

If a fault is detected as the system sections are booted (see Flowchart, [Page 248](#)), the system section ST SafetyNET p RTFN switches to a STOP condition, achieving the operating status "Major FS+ST error".

[3] Switching from "ST SafetyNET p RTFN in RUN condition without error" to "Booting system sections"

This change in operating status affects all the system sections of the PSSu system (see Flowchart, [Page 248](#)).

This change in operating status can be triggered using one of the following commands:

- ▶ Warm reset (PAS4000) or reset button (PSSu system)
- ▶ Cold reset (PAS4000)
- ▶ Warm reset of project (PAS4000)
- ▶ Cold reset of project (PAS4000)

[4] Switching from "ST SafetyNET p RTFN in RUN condition without error" to "ST SafetyNET p RTFN in RUN condition with minor error"

The change in operating status has resulted due to an ST communication error. Faults that result in this error are classified as follows:

- ▶ Continuous error in ST communication
- ▶ One-off error in ST communication.

[5a] Switching from "ST SafetyNET p RTFN in RUN condition with error" to "ST SafetyNET p RTFN in RUN condition without error"

The operating status "ST SafetyNET p RTFN in RUN condition with error" has resulted due to a continuous error in ST communication. Once the fault has been rectified, the connection is resumed automatically.

[5b] Switching from "ST SafetyNET p RTFN in RUN condition with error" to "ST SafetyNET p RTFN in RUN condition without error"

The operating status "ST SafetyNET p RTFN in RUN condition with error" has resulted due to a transient one-off error in ST communication. The connection will be resumed automatically.

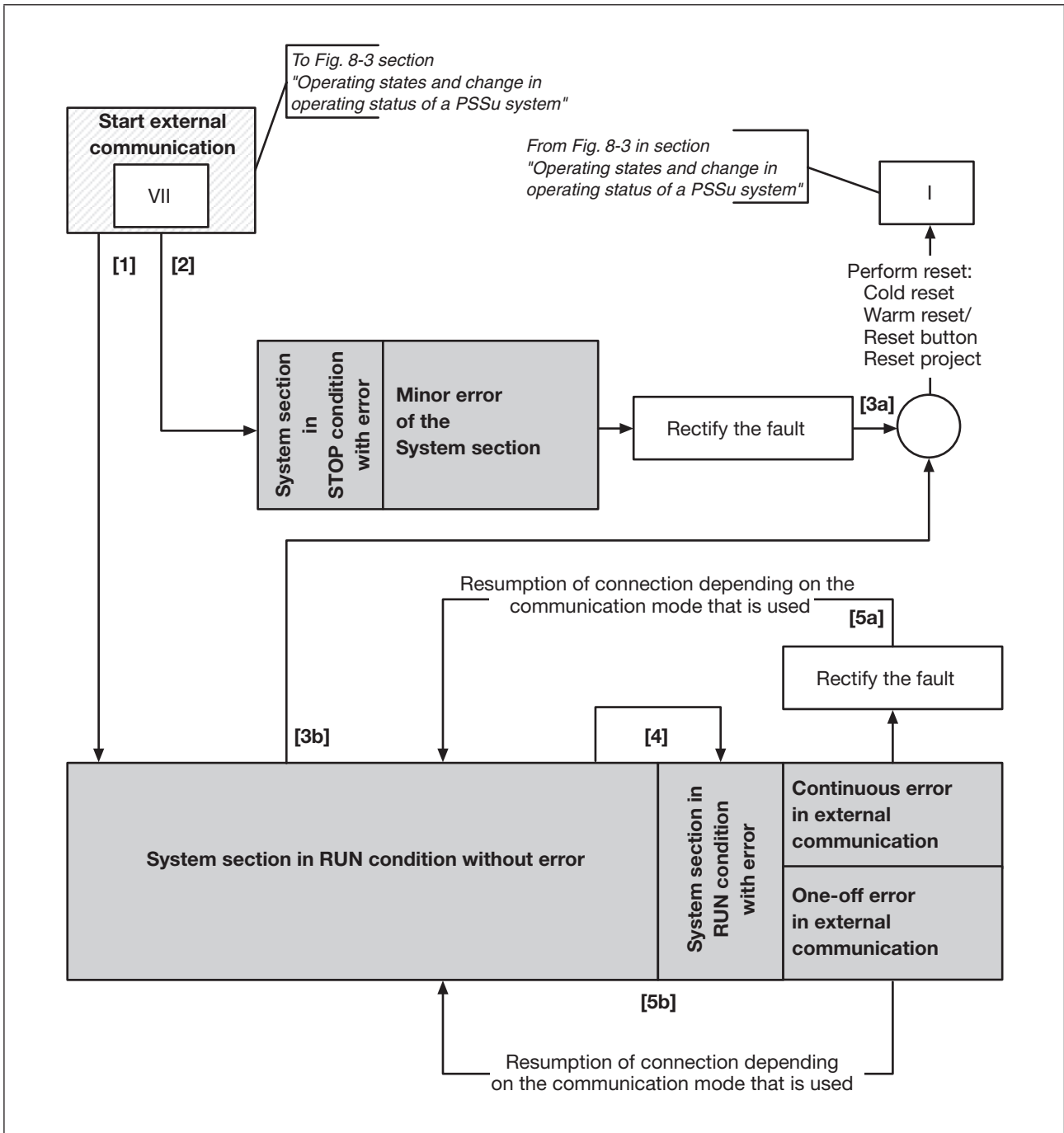
9.9 Operating states and change in operating status with external communication

The description below deals exclusively with the operating states with external communication (see Communication modes for external communication, [Page 102](#)). The operating states are identical for the respective system section (e.g. system section "IP connections", system section "PROFIBUS-DP Slave"). Where necessary, reference is made to the operating states of the PSSu system.

The diagram "Operating states and change in operating status with external communication" (see Flowchart, [Page 328](#)) provides an overview of the operating states (grey):

- ▶ The figures in square parentheses indicate the various changes in operating status.
- ▶ The section marked with the Roman numeral I refers to the diagram "Operating states and change in operating status of a PSSu system" (see Flowchart, [Page 248](#)), as the effects impact the whole PSSu system and not just the system section for the respective external communication.
- ▶ The section marked with the Roman numeral VII refers to the diagram "Operating states and change in operating status of a PSSu system" (see Flowchart, [Page 248](#)), as the system section for the respective external communication is started when booting the system sections of a PSSu system.

9.9.1 Flow chart



Operating states and change in operating status with external communication

9.9.2 Operating states with external communication

The description of the various operating states assumes that the PSSu system contains a valid device project.

The system section for the respective external communication (e.g. Modbus/TCP, Raw UDP, PROFIBUS-DP Slave) may assume the following operating states (see Flowchart, [Page 328](#)):

- ▶ Operating status "System section in RUN condition without error"
- ▶ Operating status "System section in STOP condition with error"
 - "Minor error on the system section for external communication"
- ▶ Operating status "System section in RUN condition with error"
 - "Continuous errors in external communication"
 - "One-off error in external communication"
- ▶ Major FS error
(only with communication via a safe communication medium such as PROFI-safe!)
- ▶ Major FS+ST error
(only with communication via a safe communication medium such as PROFI-safe!)



Information

- The descriptions of various operating states include a list of the options for triggering a change in operating status. These options are described separately (see Reset, restart, start and stop options, [Page 337](#)).
- In some communication modes various operating states can be displayed via status LEDs of the head module. The presence of status LEDs and their meaning depends on the communication mode used. Examples:
Modbus/TCP and Raw UDP are protocols that are based on Industrial Ethernet. They have no status LEDs to display operating states.
On PROFIBUS-DP Slave the two status LEDs "SF" and "BF" display various states.
On PROFI-safe the "SO" status LED displays various states.

9.9.2.1 System section in RUN condition without error

This operating status is achieved through one of the following options:

- ▶ After a successful boot process, which was triggered by
 - Cold starting the PSSu system or a reboot command for the PSSu system in PAS4000
 - Pressing the reset button on the PSSu system
 - Triggering a reset via PAS4000

(cold reset, warm reset, project reset)

Effects

- ▶ When the operating status of the respective external communication (e.g. Modbus/TCP, Raw UDP, PROFIBUS-DP Slave, PROFI-safe) is "System section in RUN condition without error", pending process data is transmitted faultlessly using the respective communication mode.
- ▶ Status LEDs on the head module
Depends on the communication mode that is used

9.9.2.2 Minor error on the system section for external communication

In the event of a fault that leads to this operating status, the affected parts of the system section are ended as defined. A defined status is achieved.

Possible causes

- ▶ Configuration error

Effects

- ▶ System section for the respective external communication switches to a STOP condition. With the PSSu system It is not possible to exchange process data via the corresponding communication mode.
- ▶ The substitute values are used for the affected process data, with Valid Bits = FALSE.
- ▶ Status LEDs on the head module
Depends on the communication mode that is used

Remedy

1. Diagnostics
e.g. use PAS4000 to read the diagnostic list and/or the diagnostic log
2. Rectify the fault
e. g. synchronize the configuration of the communication partners
3. Perform reset

9.9.2.3 System section in RUN condition with error

If a fault results in an error in external communication, the affected section is stopped or ended. A defined state is achieved.

Possible causes

- ▶ Continuous errors in external communication
e. g. the communication partner cannot be reached because it has been removed; the bus connector on a PSSu system has been unplugged
- ▶ One-off error in ST communication
e.g. transient, one-off error on the bus cable, loss of telegram

Effects

- ▶ There is not longer process data exchange with the communication partner.
- ▶ The interface continues to be in a RUN condition.
- ▶ The substitute values are used for the affected process data, with Valid Bits = FALSE.
- ▶ Status LEDs on the head module
depends on the communication mode that is used

Remedy

1. Diagnostics
e.g. use PAS4000 to read the diagnostic list and/or the diagnostic log
2. Rectify errors
3. Resume the connection
The options for resuming the connection depend on the communication mode that is used and in various cases also on the configuration options of the communication partners.
If this is supported by the communication mode, the interface of the PSSu system automatically resumes the connection. Otherwise, the interface must be restarted via one of the reset commands.
 - Warm reset (PAS4000) or reset button (PSSu system)
 - Cold reset (PAS4000)
 - Warm reset of project (PAS4000)
 - Cold reset of project (PAS4000).



Information

Please note:

- Under some circumstances it is necessary to restart the communication partner
- The fault may have been caused by the communication partner. In this case the fault with the communication partner must be rectified first, before the interface of the PSSu system can restore the connection.

9.9.2.4 Major FS+ST error

This operating status can only occur if the system section has a safety-related communication channel (e.g. PROFIsafe from the PROFIBUS-DP Slave with PROFIsafe system section). In the event of a fault that leads to this operating status, the system section (e.g. PROFIBUS-DP Slave with PROFIsafe) is ended as defined.

Possible causes

- ▶ Internal fault

Effects

- ▶ The system section (e.g. PROFIBUS-DP Slave with PROFIsafe) switches to a STOP condition.
- ▶ The substitute values are used for the affected process data, with Valid Bits = FALSE.

Remedy

Contact Pilz

9.9.2.5 Major FS error

This operating status can only occur if the system section has a safety-related communication channel (e.g. PROFIsafe from the PROFIBUS-DP Slave with PROFIsafe system section). In the event of a fault that leads to this operating status, communication via the system section's safety-related communication channel is ended as defined (e.g. PROFIsafe from the section system PROFIBUS-DP Slave with PROFIsafe). A defined state is achieved.

The effects of a major FS error impact all safety-related sections of a PSSu system (see Operating state "Major FS error of a PSSu system", [Page 268](#)).

9.9.3 Operating status change with external communication

This description of the change in operating status assumes that the PSSu system contains a valid device project. The figures refer to the change in operating status in the diagram "Operating states and change in operating status with external communication" (see Flowchart, [Page 328](#)).



Information

The description of each change of operating status includes a list of the options for triggering a change in operating status. These options are described separately (see Reset, restart, start and stop options, [Page 337](#)).

[1] Switching from "Booting system sections" to "System section in RUN condition without error"

"Booting system sections" is one of the PSSu system's operating states (see Flowchart, [Page 248](#)). If no fault is detected as the system sections are booted, the system section for the respective external communication is started, switching to the status "System section in RUN condition without error".

[2] Switching from "Booting system sections" to "Minor error on the system section for external communication"

If a fault is detected as the system sections are booted (see Flowchart, [Page 248](#)), the system section switches to a STOP condition, achieving the operating status "Minor error on the system section for external communication".

[3a] Switching from "Minor error on the system section for external communication" to "Booting system sections"

Once the fault has been rectified, the system section for the external communication can be restarted via one of the reset commands. This change in operating status affects all the system sections of the PSSu system (see Flowchart, [Page 248](#)).

- ▶ Warm reset (PAS4000) or reset button (PSSu system)
- ▶ Cold reset (PAS4000)
- ▶ Warm reset of project (PAS4000)
- ▶ Cold reset of project (PAS4000)

[3b] Switching from "System section in RUN condition without error" to "Booting system sections"

This change in operating status affects all the system sections of the PSSu system (see Flowchart, [Page 248](#)).

This change in operating status can be triggered using one of the following commands:

- ▶ Warm reset (PAS4000) or reset button (PSSu system)
- ▶ Cold reset (PAS4000)
- ▶ Warm reset of project (PAS4000)
- ▶ Cold reset of project (PAS4000)

[4] Switching from "System section in RUN condition without error" to "System section in RUN condition with error"

The change in operating status has resulted due to a communication error. Faults that result in this error are classified as follows:

- ▶ Continuous error in external communication
- ▶ One-off error in external communication.

[5a] Switching from "System section in RUN condition with error" to "System section in RUN condition without error"

The operating status "System section in RUN condition with error" has resulted due to a continuous error in external communication. Once the fault has been rectified, the options for resuming the connection depend on the communication mode that is used and in various cases also on the configuration options of the communication partners.

If the actual cause of the error does not lie with the communication partner, the system section for external communication can be restarted with one of the following commands:

- ▶ Warm reset (PAS4000) or reset button (PSSu system)
- ▶ Cold reset (PAS4000)
- ▶ Warm reset of project (PAS4000)
- ▶ Cold reset of project (PAS4000)

[5b] Switching from "System section in RUN condition with error" to "System section in RUN condition without error"

The operating status "System section in RUN condition with error" has resulted due to a transient one-off error in external communication. The options for resuming the connection depend on the communication mode that is used and in various cases also on the configuration options of the communication partners.

If the actual cause of the error does not lie with the communication partner, the system section for external communication can be restarted with one of the following commands:

- ▶ Warm reset (PAS4000) or reset button (PSSu system)
- ▶ Cold reset (PAS4000)
- ▶ Warm reset of project (PAS4000)
- ▶ Cold reset of project (PAS4000)

9.10 Reset, restart, start and stop

PSS 4000 operates in accordance with the principle of "de-escalation". The aim of the principle of de-escalation is to limit the effects of intervening in the control process (e.g. when a fault occurs) to the affected areas if possible. All unaffected areas should continue operating without disruption if possible, and without repercussions on the safety function.

Various options are available to stop and start specific plant/machine sections, during commissioning for example. The implications of intervening in the control process depend on the specific command that is used.

9.10.1 Overview

9.10.1.1 Reset, restart, start and stop options

The following table provides an overview of the reset, restart, start and stop options and the corresponding commands in PAS4000:

Options	PAS4000
Cold start a PSSu system (affects all system sections on a PSSu system)	- - -
Reboot a project (affects all system sections on all the project's PSSu systems)	Reboot
Reboot a PSSu system (affects all system sections on a PSSu system)	Reboot
Reset a project (affects all system sections on all the project's PSSu systems)	Cold reset
	Warm reset
Reset a PSSu system (affects all system sections on a PSSu system)	Cold reset
	Warm reset
Reset via the reset button of a PSSu system	- - -
Restart a project (affects all resources on all PSSu systems)	Cold restart
	Warm restart
Restart a resource (affects a resource on a PSSu system)	FS cold restart
	ST cold restart
	FS warm restart
	ST warm restart
Start tasks on a resource (affects one or more tasks on a resource)	FS hot start
	ST hot start
Stop a project (affects all resources on all the project's PSSu systems)	Stop
Stop a PSSu system (affects all resources on a PSSu system)	Stop
Stop a resource (affects a resource on a PSSu system)	FS stop
	ST stop

9.10.1.2 Impact on local and resource global variables

The following table provides an overview of the impact of a reset, restart, start or stop on local variables and resource global variables (not O-PI variables).

After one of the stated commands, the affected sections (e.g. tasks on a resource, system sections) are started up with the following variable values:

Action	Variable with attribute RETAIN PERSISTENT	Volatile variable
Cold start Reboot	Saved value	Variable is set to initial value
Warm restart Warm reset		
Cold restart Cold reset	Variable is set to initial value	
Hot start	Saved value	
Stop	Saved value is retained	

Notes:

- ▶ The content of the variable with the attribute RETAIN PERSISTENT is saved after a task is executed and is retained even in the event of a power failure or restart/reboot.
- ▶ The saved content of the variable with the attribute RETAIN PERSISTENT may come from the penultimate task execution before the power failure.
If several variables are saved, some may come from the penultimate task execution and others from the last execution.

9.10.1.3 Impact on active "variable forcing"

The following table provides an overview of the impact of a reset, restart, start or stop on active "variable forcing":

Action	Impact on "variable forcing"
Cold start Reboot	"Variable forcing" is stopped, "variable forcing" will then need to be started manually
Warm restart Warm reset	
Cold restart Cold reset	
Hot start	"Variable forcing" is stopped, "variable forcing" will then be started automatically through the system
Stop	"Variable forcing" is stopped, the way in which "variable forcing" is restarted depends on the subsequent (re-) start action/reset action: <ul style="list-style-type: none"> - Warm restart/warm reset - Cold restart/cold reset - Hot start

9.10.1.4 Impact on "variable controlling"

"Variable controlling" is executed once only. A reset, restart, start or stop command only has an impact if "variable controlling" has not yet been run.

The following table provides an overview of the impact of a reset, restart, start or stop on "variable controlling":

Action	Impact on "variable controlling"
Cold start Reboot	"Variable controlling" actions that have not yet been run are deleted
Warm restart Warm reset	
Cold restart Cold reset	
Hot start	"Variable controlling" actions that have not yet been run are run after a hot start
Stop	"Variable controlling" is not run, whether the "variable controlling" action is run depends on the subsequent (re-) start action/reset action: <ul style="list-style-type: none"> - Warm restart/warm reset - Cold restart/cold reset - Hot start

9.10.2 Cold start and Reboot command

The effect of a cold start (supply voltage for the module supply is switched off and on) and a reboot command is device-based, i.e. it affects all the system sections on a PSSu system (FS resource, ST resource etc.).

Procedure

Performing a cold start/reboot command initiates the following procedure:

1. Processing is aborted
2. PSSu system switches to "Booting base system" operating status.
3. "Booting system sections" operating status is achieved and all system sections are started
4. Operational entry in the diagnostic log
5. Impact after the cold start/reboot command

The impact on local variables and resource global variables is described separately (see Impact on local and resource global variables, [Page 339](#)). Impact on "variable forcing" is also described separately (see Impact on active "variable forcing", [Page 340](#)).

Behaviour during a cold start/reboot command

The table below describes the behaviour of the process data and system sections:

Process data, system sections		Behaviour	
Process data	I-data of the module bus with I/O mapping [1]	Bit modules	Inputs = current physical input signals
		Byte modules	Inputs = current physical input values
	O-data of module bus	Bit modules	Outputs = ZERO
		Byte modules	Outputs = switch-on values (see module's operating manual)
O-PI variables with I/O mapping [1]		O-PI variables are marked as invalid	
System sections	FS resource	See "Procedure"	
	ST resource		
	FS module bus		
	ST module bus		
	FS SafetyNET p RTFN		
	ST SafetyNET p RTFN		
	External communication e.g. "IP connections" system section, "PROFIBUS-DP Slave" system section		

[1]

Process data that is accessed by other PSSu systems is set to substitute values in the other PSSu systems; Valid Bits = FALSE (see Validity of process data, [Page 202](#)).

9.10.3 Reset commands

The effect of a reset (cold reset, warm reset) is device-based, i.e. it affects all system sections of a PSSu system (FS resource, ST resource etc.). The main difference between a cold reset and a warm reset is the way in which the non-volatile variables are handled.

A cold reset can be used when a defined output status is to be brought about on a PSSu system in respect of the variables.

A warm reset can be used when the desired behaviour on a PSSu system is the same as that for a cold start/reboot command in respect of the non-volatile variables.

The following resets are available:

- ▶ Cold resets
 - Cold reset of all a project's PSSu systems
 - Cold reset of one PSSu system
- ▶ Warm resets:
 - Warm reset of all a project's PSSu systems
 - Warm reset of one PSSu system (PAS4000 or reset button)

Procedure

Performing a reset initiates the following procedure:

1. All system sections are stopped
 - Resetting a project:

The tasks on all resources and all other system sections on all the project's PSSu systems are run through to the end; any further execution is then stopped.
 - Resetting a PSSu system:

The tasks on all resources and all other system sections of the PSSu system are run through to the end; any further execution is then stopped.
2. Booting system sections
 - Resetting a project:

All the project's PSSu systems switch to "Booting system sections" operating status, including starting all system sections and all tasks on all resources that were in a RUN condition or had a task error prior to the reset.
 - Resetting a PSSu system:

Selected PSSu system switches to "Booting system sections" operating status, including starting all system sections and all tasks on all resources that were in a RUN condition or had a task error prior to the reset.
3. Diagnostic log
 - Resetting a project:

Operational entry is made in the diagnostic log for each of the project's PSSu systems
 - Resetting a PSSu system:

Operational entry is made in the diagnostic log for the selected PSSu system

4. Diagnostic list

- Resetting a project:
Diagnostic list is deleted
- Resetting a PSSu system:
Messages about the selected PSSu system are deleted from the diagnostic list

5. Impact after a reset

The impact on local variables and resource global variables is described separately (see Impact on local and resource global variables, [Page 339](#)). Impact on "variable forcing" is also described separately (see Impact on active "variable forcing", [Page 340](#)).

Behaviour during a reset

The table below describes the behaviour of the process data and system sections during a reset:

Process data, system sections		Behaviour	
Process data	I-data of the module bus with I/O mapping [1]	Bit modules	Inputs = current physical input signals
		Byte modules	Inputs = current physical input values
	O-data of module bus	Bit modules	Outputs = ZERO
		Byte modules	Outputs = switch-on values (see module's operating manual)
	O-PI variables with I/O mapping [1]	O-PI variables are marked as invalid	
System sections	FS resource	See section entitled "Procedure"	
	ST resource		
	FS module bus		
	ST module bus		
	FS SafetyNET p RTFN		
	ST SafetyNET p RTFN		
	External communication e.g. "IP connections" system section, "PROFIBUS-DP Slave" system section		

[1]

Process data that is accessed by other PSSu systems is set to substitute values in the other PSSu systems; Valid Bits = FALSE (see Validity of process data, [Page 202](#)).

9.10.4 Hot start commands

A hot start (FS hot start, ST hot start) impacts on the following system sections of a PSSu system:

- ▶ FS hot start: affected tasks belonging to the FS resource
- ▶ ST hot start: affected tasks belonging to the ST resource

With a hot start, the process is continued exactly at the point at which it was previously stopped (runtime error or stop command). A hot start has no effect on the diagnostic list, i.e. the entries are retained.

A hot start is suitable for commissioning, for example. One of the stop commands (see Stop commands, [Page 348](#)) can be used to halt a targeted subprocess (e.g. for adjustment, step-by-step commissioning), which is then continued with a hot start.

Hot start in operating status "Resource in RUN condition with task error"

Procedure for an FS/ST hot start while the FS/ST resource is in a "FS/ST resource is in a RUN condition with a task error: Task in a STOP condition"

Prerequisite: The fault has been rectified

1. Operational entry is made in the diagnostic log
2. Affected tasks on the FS/ST resource are started

Only those resource tasks that were no longer run due to a fault are started. A hot start does not affect any tasks on the FS/ST resource that have run faultlessly.

3. Impact after a hot start

The impact on local variables and resource global variables is described separately (see Impact on local and resource global variables, [Page 339](#)). Impact on "variable forcing" is also described separately (see Impact on active "variable forcing", [Page 340](#)).

Hot start after a stop command

Procedure for an FS/ST hot start after a stop command:

Prerequisite: No task errors are present.

1. Operational entry is made in the diagnostic log
2. Affected tasks on the FS/ST resource are started

All tasks on the selected resource are started. The user program is continued at the point at which it was previously stopped via the stop command.

3. Impact after a hot start

The impact on local variables and resource global variables is described separately (see Impact on local and resource global variables, [Page 339](#)). Impact on "variable forcing" is also described separately (see Impact on active "variable forcing", [Page 340](#)).

9.10.5 Restart commands

The effect of a restart (cold restart, warm restart) is resource-based, i.e. it affects the selected resource on a PSSu system (ST/FS resource). The main difference between a cold restart and a warm restart is the way in which it handles the non-volatile variables used to start a resource after a restart.

A cold restart can be used when a defined output status is to be brought about for a resource (FS resource/ST resource) in respect of the variables.

A warm restart can be used when the desired behaviour for a resource (FS resource/ST resource) is the same as that for a cold start/reboot command in respect of the non-volatile variables.

The following restarts are available:

- ▶ Cold restart for a project
- ▶ Warm restart for a project
- ▶ FS cold restart for the FS resource on a PSSu system
- ▶ ST cold restart for the ST resource on a PSSu system
- ▶ FS warm restart for the FS resource on a PSSu system
- ▶ ST warm restart for the ST resource on a PSSu system

As a restart is always preceded by an automatic stop (fault on the ST/FS resource) or a manually executed stop (stop command), the impact on the affected process data and system sections is determined by the resource's STOP condition. While a resource is in a STOP condition, the process data and system sections always behave in the same way, so it does not matter how the STOP condition was triggered (see Stop commands – Behaviour during the STOP condition, [Page 348](#)).

Restart in the operating status "Resource in RUN condition without error"

Procedure for a restart while the selected resource is in a RUN condition without error:

1. Resources are stopped
 - Restarting the resources on a project's PSSu systems:
On all the project's PSSu systems, the resources (FS and ST resources) are run through to the end; any further execution is then stopped.
 - Restarting the resource on a PSSu system:
The selected resource (FS/ST resource) on the PSSu system is run through to the end; any further execution is then stopped.
2. Diagnostic log
 - Restarting a project:
Operational entry is made in the diagnostic log for each of the project's resources
 - Restarting a resource on a PSSu system:
Operational entry is made in the diagnostic log for the selected resource
3. Diagnostic list
 - Restarting a project:
Messages for all the resources are deleted from the diagnostic list.
 - Restarting a resource on a PSSu system:

Messages for the selected resource are deleted from the diagnostic list

4. Resources are started

- Cold/warm restart for a project:

The resources (FS and ST resources) are started on all the project's PSSu systems.

- Cold/warm restart for a resource on a PSSu system:

The selected resource (FS/ST resource) on the PSSu system is started.

5. Impact after a cold/warm restart

The impact on local variables and resource global variables is described separately (see Impact on local and resource global variables, [Page 339](#)). Impact on "variable forcing" is also described separately (see Impact on active "variable forcing", [Page 340](#)).

Restart after a stop command/error

Procedure for restarting after a stop command/error:

Prerequisite in the event of an error: The fault has been rectified

1. Diagnostic log

- Restarting a project:

Operational entry is made in the diagnostic log for each of the project's resources

- Restarting a resource on a PSSu system:

Operational entry is made in the diagnostic log for the selected resource

2. Diagnostic list

- Restarting a project:

Messages for all the resources are deleted from the diagnostic list

- Restarting a resource on a PSSu system:

Messages for the selected resource are deleted from the diagnostic list

3. Resources are started

- Restarting a project:

The resources (FS and ST resources) are started on all the project's PSSu systems.

- Cold/warm restart of a PSSu system:

The selected resource (FS/ST resource) is started.

4. Impact after a restart

The impact on local variables and resource global variables is described separately (see Impact on local and resource global variables, [Page 339](#)). Impact on "variable forcing" is also described separately (see Impact on active "variable forcing", [Page 340](#)).

9.10.6 Stop commands

The following stop commands are available:

▶ Stopping a resource

The command is resource-based, i.e. it affects the selected resource on a PSSu system:

- Stop FS resource: Stops the FS resource on a PSSu system
- Stop ST resource: Stops the ST resource on a PSSu system

▶ Stopping a PSSu system

The command is device-based, i.e. it affects all resources (FS resource, ST resource) on the selected PSSu system.

▶ Stopping a project

The command is device-based, i.e. it affects all resources (FS resource, ST resource) on all a project's PSSu systems.

Procedure for stopping a resource

Stopping the selected resource (FS resource/ST resource) on a PSSu system initiates the following procedure:

1. All tasks on the selected resource are stopped

The impact on local and resource global variables processed via the selected resource is described separately (see Impact on local and resource global variables, [Page 339](#)). Impact on "variable forcing" is also described separately (see Impact on active "variable forcing", [Page 340](#)).

2. Operational entry is made in the diagnostic log for the selected resource
3. Message is entered in the diagnostic list

Procedure for stopping a PSSu system

Stopping a PSSu system initiates the following procedure:

1. All resources on the selected PSSu system are stopped

The impact on local and resource global variables processed via the selected resources is described separately (see Impact on local and resource global variables, [Page 339](#)). Impact on "variable forcing" is also described separately (see Impact on active "variable forcing", [Page 340](#)).

2. Operational entries are made in the diagnostic log for the resources on the PSSu system
3. Messages are entered in the diagnostic list

Procedure for stopping a project

Stopping a project initiates the following procedure:

1. All resources on all the project's PSSu systems are stopped

The impact on local and resource global variables processed via the selected resources is described separately (see Impact on local and resource global variables, [Page 339](#)). Impact on "variable forcing" is also described separately (see Impact on active "variable forcing", [Page 340](#)).

2. Operational entries are made in the diagnostic log for all of the project's resources
3. Messages are entered in the diagnostic list

Behaviour during the STOP condition

The behaviour of the process data and system sections while a resource is in a STOP condition is always the same, whether the STOP condition was triggered by a fault or a manual stop command. The table below describes the behaviour of the process data and system sections while a resource is in a STOP condition:

Process data, system sections		Behaviour	
Process data	I-data of the module bus with I/O mapping	Bit modules	Inputs = current physical input signals
	(for "Stopping a PSSu system" see [1])	Byte modules	Inputs = current physical input values
	O-data of the module bus with I/O mapping	Bit modules	Outputs = ZERO
	(for "Stopping a resource" see [2])	Byte modules	Outputs = switch-on values (see module's operating manual)
	O-PI variables of the stopped resource or resources	O-PI variables are marked as invalid (see [1] and [2])	
System sections	FS resource	See sections - Procedure for stopping a resource - Procedure for stopping a PSSu system - Procedure for stopping a project	
	ST resource		
	FS module bus		
	ST module bus		
	FS SafetyNET p RTFN		
	ST SafetyNET p RTFN		
	External communication e.g. "IP connections" system section, "PROFIBUS-DP Slave" system section		

[1]

Process data that is accessed by other PSSu systems is set to substitute values in the other PSSu systems; Valid Bits = FALSE (see Validity of process data, [Page 202](#)).

[2]

Only the hardware outputs that are assigned to the O-PI variables on the stopped resource behave as described in the above table. Hardware outputs that are assigned to the O-PI variables on other (running) resources continue to be written by these variables.

9.11 Functions of the reset button

The availability of a reset button depends on the device. Details can be found in the operating manual of the relevant device. On modular PSSu systems, the reset button is on the head module, for example,

The reset button has various functions:

- ▶ Perform a warm reset for a PSSu system
- ▶ Deliberate operator action
 - To perform an original reset for the PSSu system
 - To transfer the naming data and/or device project from the removable data medium

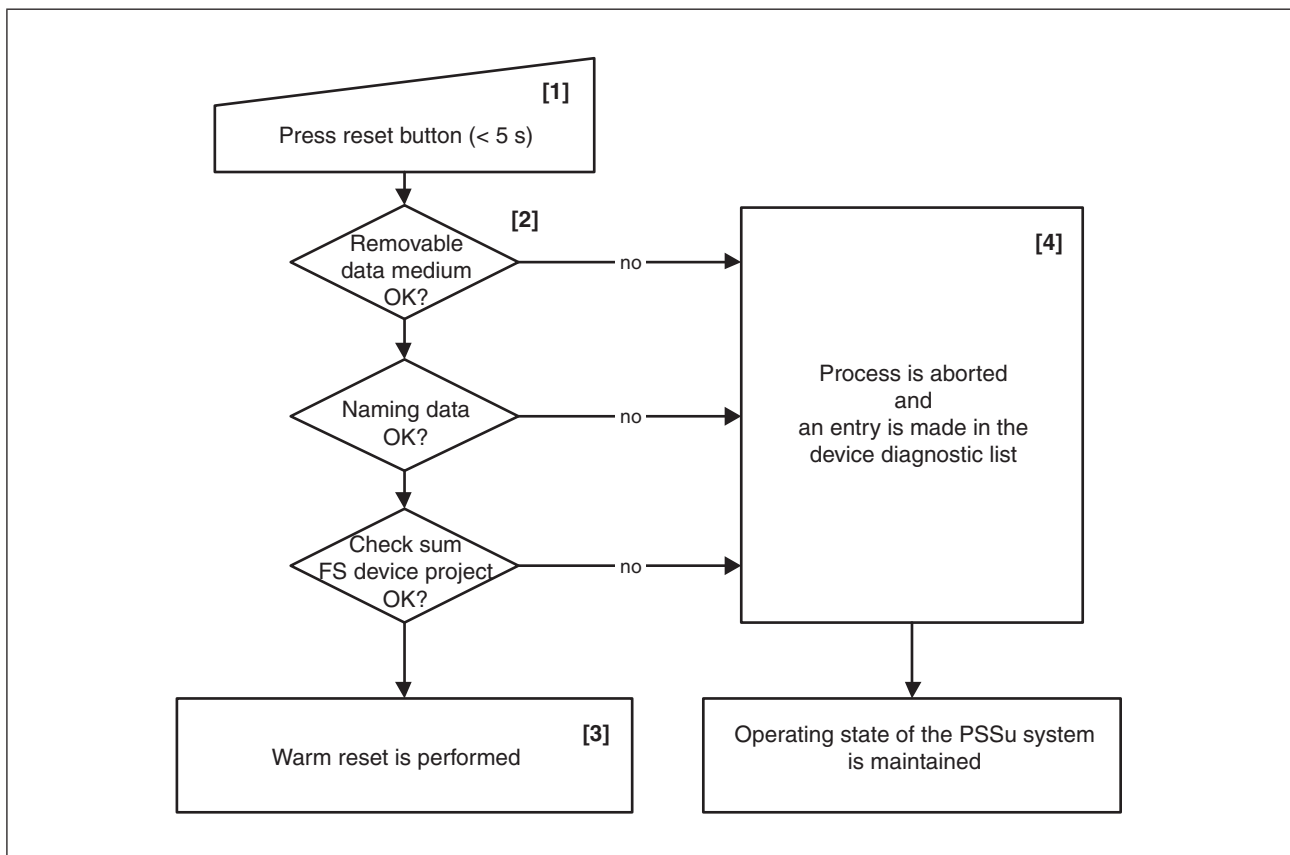
9.11.1 Perform a warm reset for a PSSu system

On a PSSu system, a warm reset triggers a switch to the operating status "Booting system sections".

The general effects of a warm reset are described separately (see Reset commands, [Page 343](#)).

This section describes how to perform a warm reset via the reset button on the head module of a PSSu system. The tool's online help describes how to perform a warm reset in PAS4000.

Procedure for a warm reset via the reset button



[1]

A reset via the reset button is a warm reset. For a warm reset, the reset button on the head module of the PSSu system must be pressed for max. 5 s.

[2]

Various checks are made before performing the warm reset. While these checks are carried out, the PSSu system remains unchanged in its current operating status. The following checks are performed:

- ▶ Is the removable data medium accessible, i.e.
 - The removable data medium is present
 - The removable data medium is intact
- ▶ Does the naming data on the removable data medium match that of the PSSu system

- ▶ Is the check sum of the FS device project on the removable data medium identical to that of the PSSu system.
Further information on check sums is available in PAS4000's online help under "Security mechanisms".

[3]

If all the checks are positive, the warm reset is performed.

[4]

If any of the checks is negative, the process is aborted. The PSSu system remains unchanged in its current operating status.

9.11.2 Deliberate operator action for an original reset of the PSSu system

An original reset can be performed using the reset button on the head module of a PSSu system.

Prerequisites

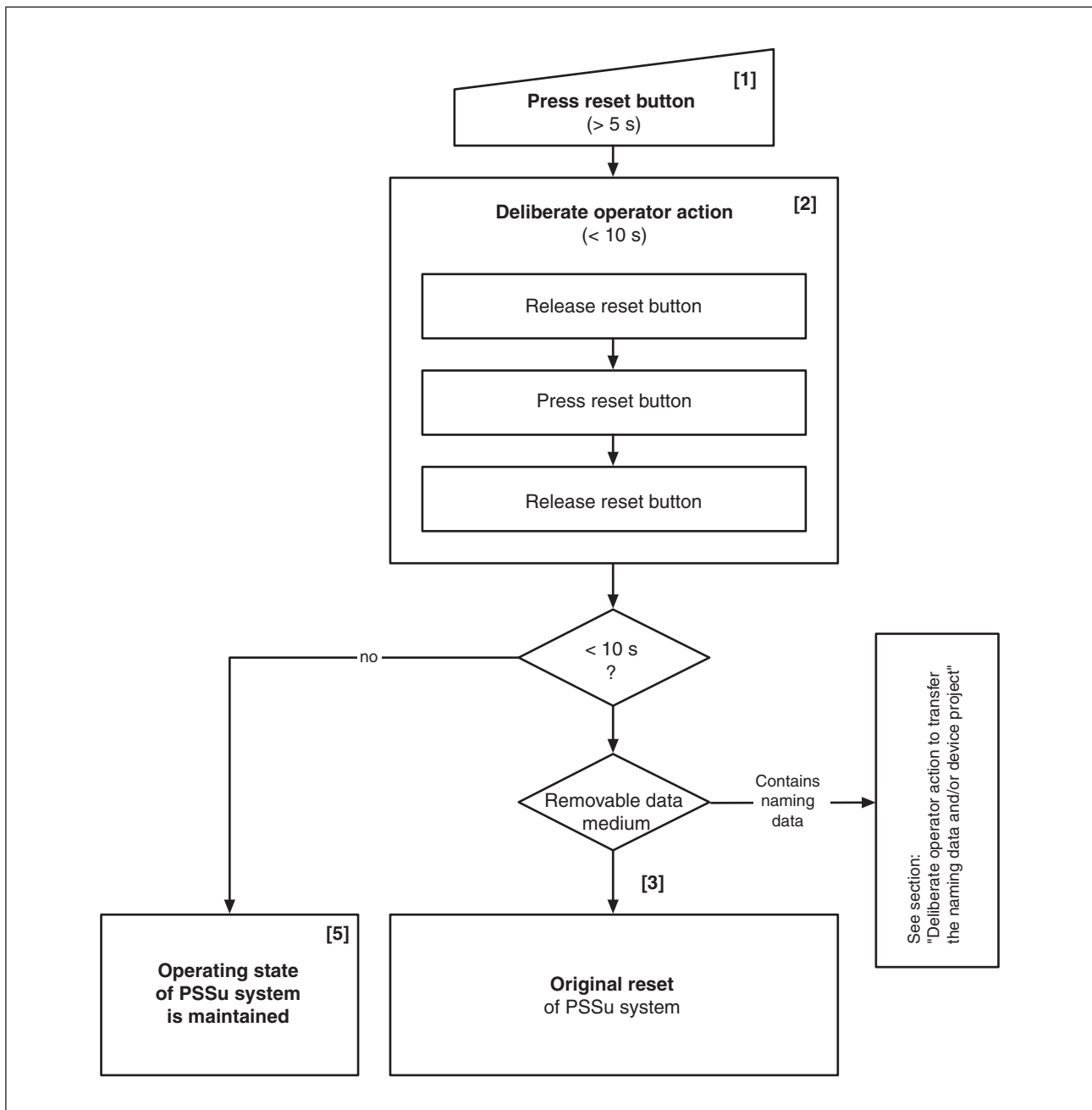
The removable data medium on the head module of the PSSu system must be either disconnected or empty.



Information

Please note that a defective removable data medium will also trigger an original reset.

Procedure for an original reset



[1]
The reset button on the head module of the PSSu system is pressed for longer than 5 s. The PSSu system remains unchanged in its current operating status.

[2]
Deliberate operator action within 10 s:
Release reset button – Press reset button – Release reset button
The start of deliberate operator action is displayed through the DIAG LED (flashing red-green).

[3]

The removable data medium is checked. If an empty removable data medium is connected, or no data medium is connected, the original reset is performed (see Original reset, [Page 361](#)).

Please note:

If the removable data medium contains naming data, an original reset will not be performed (see “Deliberate operator action to transfer the naming data and/or device project”, [Page 356](#)).

[4]

If deliberate operator action is not performed within 10 s, the PSSu system will remain unchanged in its current operating status.

9.11.3 Deliberate operator action to transfer the naming data and/or device project

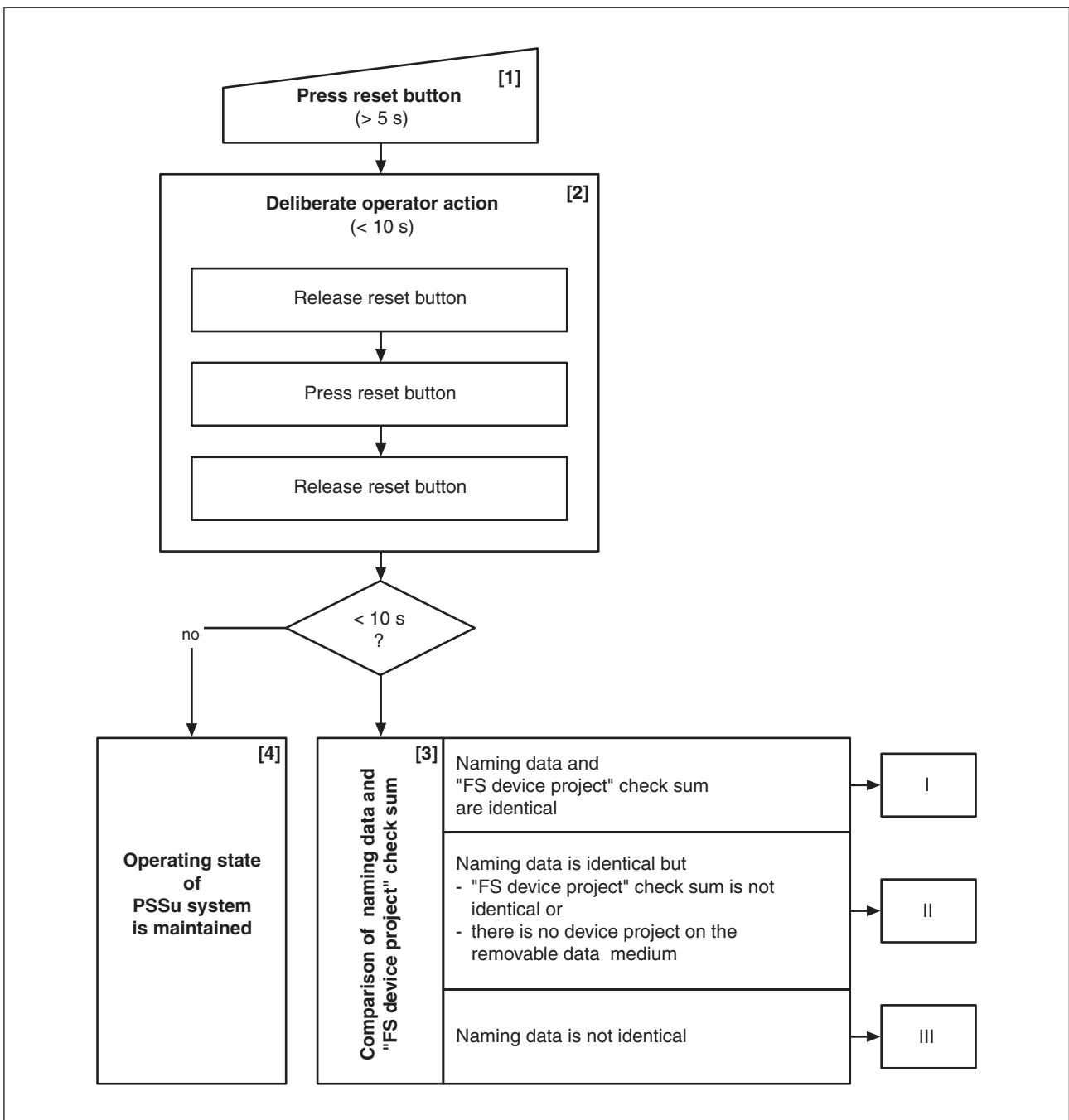
The naming data and/or device project can be transferred from the removable data medium to the PSSu system using the reset button. In many cases, a warm reset will be sufficient to enable data to be transferred using the reset button (see Perform a warm reset for a PSSu system, [Page 351](#)). In the following cases, deliberate operator action is absolutely essential for transferring data:

- ▶ The device has an FS resource.
- ▶ The device is **not** a new device and **not** a device for which an original reset has been performed.

Prerequisites

- ▶ A removable data medium must be connected on the PSSu system.
- ▶ The removable data medium must be neither empty nor defective.
- ▶ The device project on the removable data medium must match the product type.

Procedure for transferring the naming data and/or the device project



[1]
 The reset button on the head module of the PSSu system is pressed for longer than 5 s. The PSSu system remains unchanged in its current operating state.

[2]
 Perform deliberate operation action within 10 s:
 Release reset button – Press reset button – Release reset button
 The start of deliberate operator action is displayed through the DIAG LED (flashing red-green).

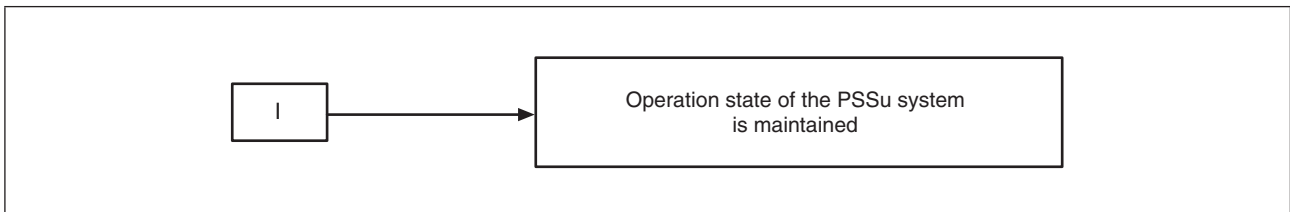
[3]

When the naming data and the check sum of the FS device project on the removable data medium are compared with the corresponding data on the PSSu system, a distinction is made between various cases. The sections marked with the Roman numerals I, II and III refer to the corresponding diagrams and description.

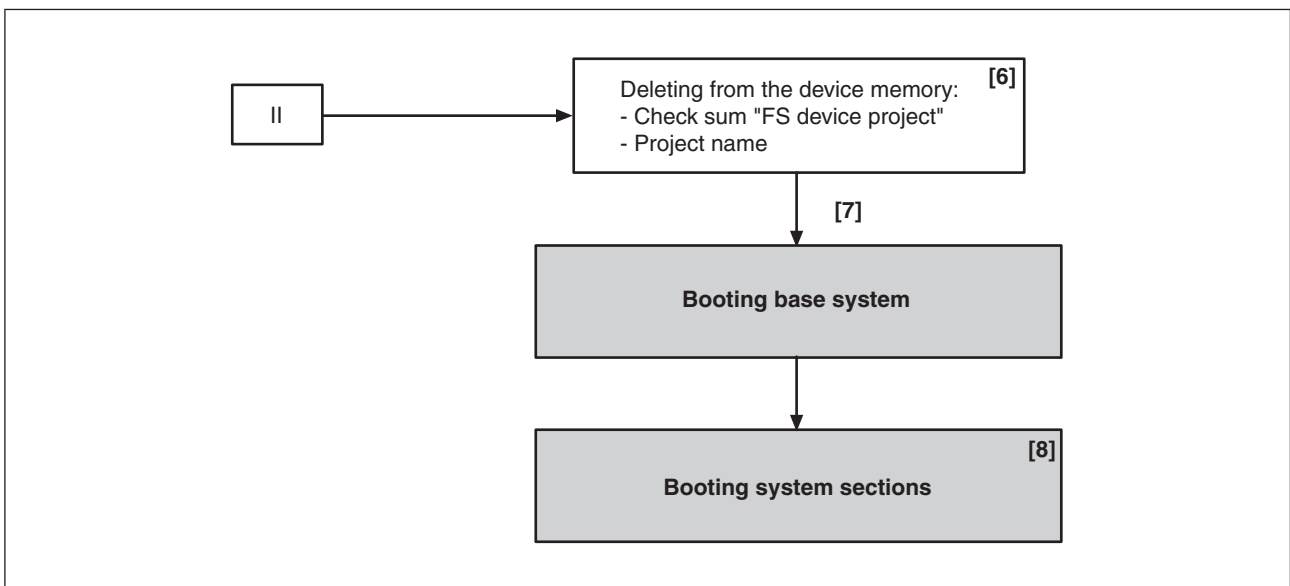
[4]

If deliberate operator action is not performed within 10 s, the PSSu system will remain unchanged in its current operating state.

Procedures after the comparison



The naming data and check sum of the FS device project are identical. The PSSu system remains unchanged in its current operating status.



[6]

If the naming data is identical, but the check sum of the FS device project is not identical, or there is no device project on the removable data medium, the check sum of the FS device project and the project name will be deleted from the device memory of the PSSu system.

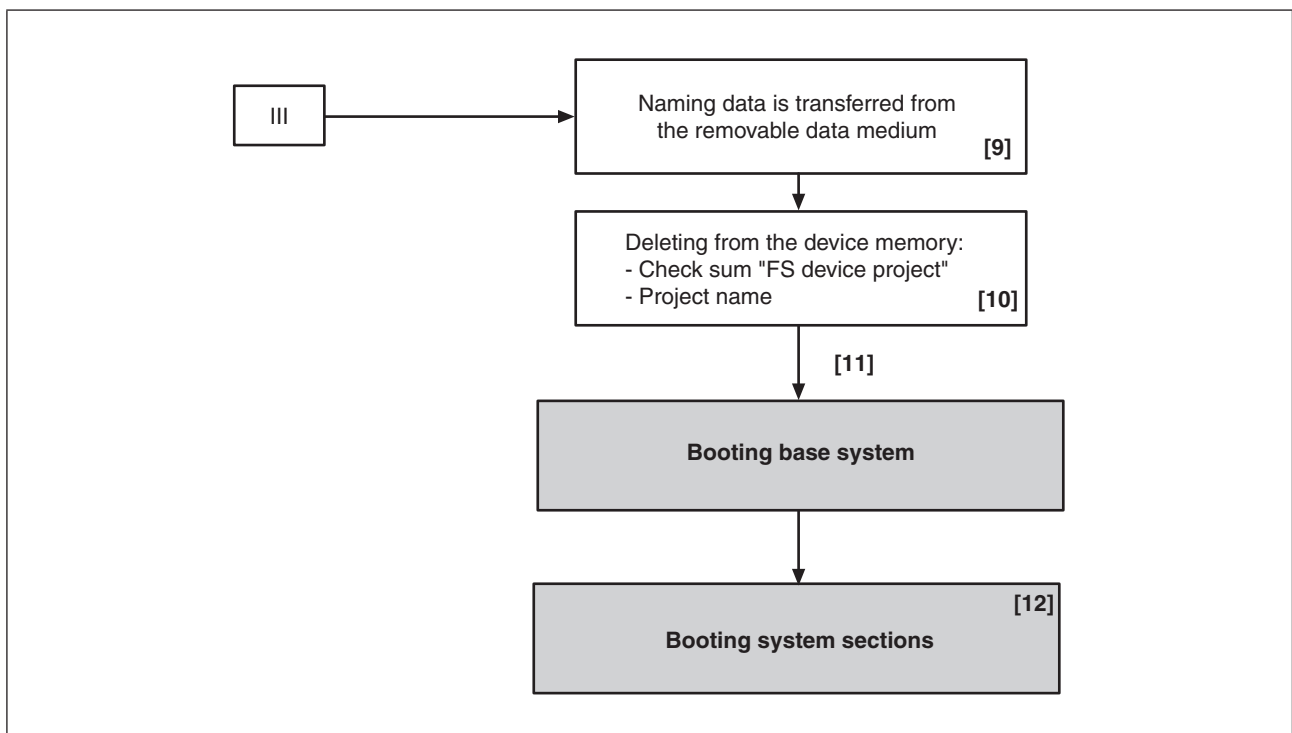
[7]

The PSSu system is restarted; the behaviour is the same as for a cold start/reboot command (see Operating states and change in operating status of a PSSu system, [Page 248](#)).

[8]

If the removable data medium contains a device project that matches the product type, the device project will be transferred from the removable data medium in the operating status "Booting system sections".

If the removable data medium does not contain a device project, or contains a device project which does not match the product type, the PSSu system will start without a device project. In this case, for the PSSu system to obtain a device project, a project will either need to be downloaded or deliberate operator action will need to be carried out again, this time with a removable data medium containing a device project that matches the product type.



[9]

If the naming data is not identical, the naming data from the removable data medium will be transferred to the PSSu system.

This procedure also applies if the PSSu system is in its delivery condition or if an original reset has been performed (see Device naming when exchanging devices, [Page 237](#))

[10]

The check sum of the FS device project and the project name are deleted from the device memory of the PSSu system.

[11]

The PSSu system is restarted; the behaviour is the same as for a cold start/reboot command (see Operating states and change in operating status of a PSSu system, [Page 248](#)).

[12]

If the removable data medium contains a device project that matches the product type, the device project will be transferred from the removable data medium in the operating status

"Booting system sections". If the user program from the old device project contained non-volatile variables, the PSSu system will start up with the following non-volatile variable values:

	Variables with the attribute RETAIN PERSISTENT
The unchanged variable is available in the user program	Saved values
Variable in the user program is new	Set to initial values
Variable is no longer available in the user program	Delete from the variable memory

If the removable data medium does not contain a device project, or contains a device project which does not match the product type, the PSSu system will start its system sections without a device project. For the PSSu system to obtain a device project, a project will either need to be downloaded or deliberate operator action will need to be carried out again, this time with a removable data medium containing a corresponding device project.

9.12 Original reset

An original reset is always sensible or indeed necessary when a PSS 4000-exclusive device or PSS 4000-operable device is removed from an existing plant/machine, possibly to re-use it in another application.

If a PSS 4000-exclusive device/operable device is re-used in a project belonging to the automation system PSS 4000 following an original reset, a new device naming procedure will need to be carried out.

9.12.1 Effects of an original reset

An original reset has the following general effects on a PSS 4000-exclusive device or a PSS 4000-operable device:

- ▶ All the project-related data and working directories stored on the device will be deleted. These include:
 - The project name
 - With a device for safety-related applications: The check sum of the FS device project
 - With a device with non-volatile memory: Non-volatile variables
- ▶ The device name is reset to the default device name (see Naming data, [Page 223](#)).
- ▶ In devices with removable data medium, all the project-related data and working directories stored on the removable data medium are deleted. These include:
 - The device project (includes the configuration of all related system sections)
 - With a device for safety-related applications: The check sum of the FS device project
- ▶ The messages on the device diagnostic list changes to "inactive" state. The status change is not logged in the device diagnostic log.
- ▶ The device diagnostic log is retained.
- ▶ The current version of the PSS 4000 firmware remains on the device.
- ▶ After an original reset the PSS 4000 firmware restarts on the device. The effect is as in a reboot command.

In addition to these general effects there also device-specific effects for

- ▶ PSS 4000-exclusive devices
- ▶ PSS 4000-operable devices from Pilz
- ▶ PSS 4000-operable devices from a third-party manufacturer:

9.12.1.1 Effects on a PSS 4000-exclusive device

An original reset has the following device-specific effects on a PSS 4000-exclusive device:

- ▶ All outputs on the PSSu system are set to ZERO.
- ▶ Process data that is accessed by other devices is set to its substitute values in the other devices; Valid Bits = FALSE (see Validity of process data, [Page 202](#)).
- ▶ The device resets the IP address to the default setting and activates its Auto-IP mechanism (see Auto-IP mechanism, [Page 73](#)).
- ▶ Device with the system section PROFIBUS-DP Slave:
If the DIP switch on the device is set to 126_D, the device's station address is reset to 126_D.

9.12.1.2 Effects on a PSS 4000-operable device from Pilz

The original reset has the following device-specific effects on a PSS 4000-operable device from Pilz:

- ▶ The network settings (IP address etc.) of the device remains unchanged, i.e. the original reset is without influence on the network settings of the device (see Naming data, [Page 223](#)).

9.12.1.3 Effects on a PSS 4000-operable device from a third-party manufacturer

The original reset has the following device-specific effects on a PSS 4000-operable device from a third-party manufacturer:

- ▶ An original reset only has an effect on the PSS 4000 firmware of a device.
- ▶ The network settings (IP address etc.) of the device remains unchanged, i.e. the original reset is without influence on the network settings of the device (see Naming data, [Page 223](#)).

9.12.2 Perform original reset

Depending on the device, the original reset can be performed as follows:

- ▶ PSS 4000-exclusive devices
An original reset is performed either in PAS4000 or by using the reset button on the PSS 4000-exclusive device.
- ▶ PSS 4000-operable devices from Pilz
An original reset is performed in PAS4000.
- ▶ PSS 4000-operable devices from a third-party manufacturer
An original reset is performed in PAS4000.

9.13 Enable principle

With the enable principle, an ST data source can control an FS hardware output, provided there is an enable from an FS data source. In the event of an error, the FS data source must ensure that the FS hardware output controlled by the ST data source can no longer be set.

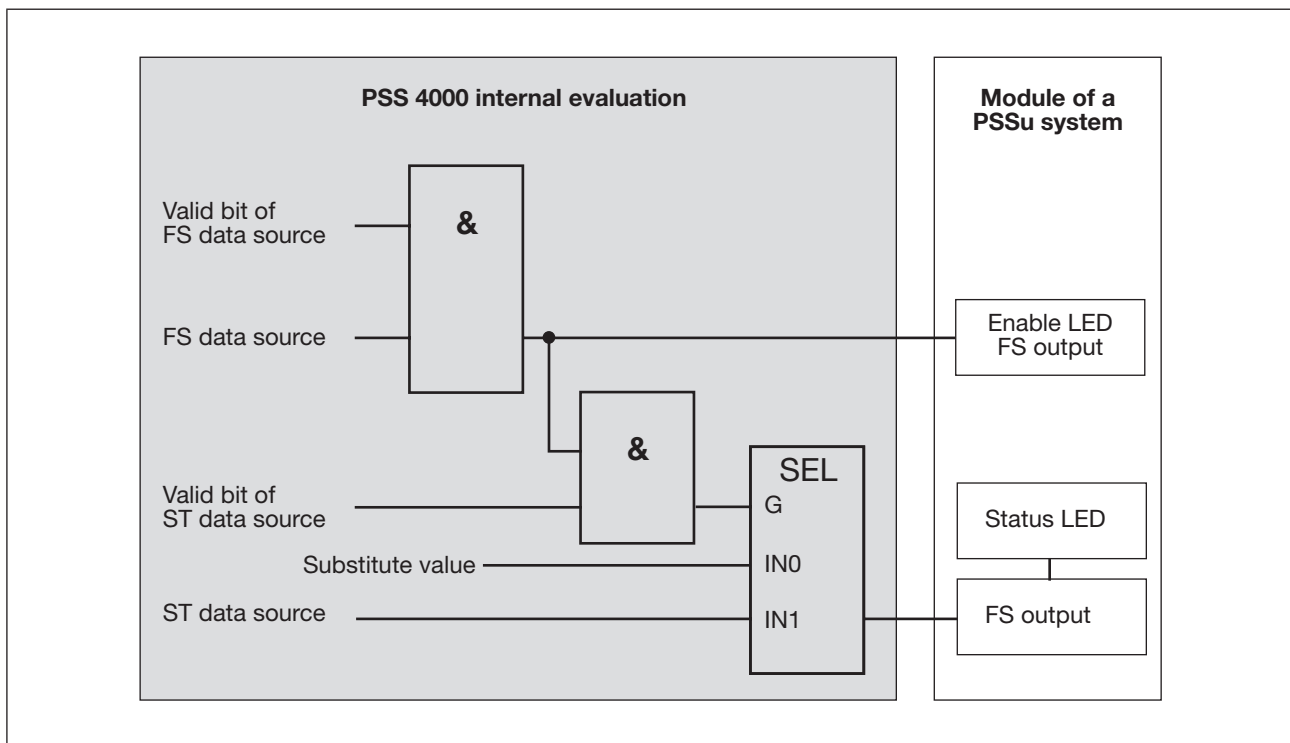
- ▶ Valid ST data sources are:
 - ST-O-PI variables
 - ST-I data from external communication (e.g. Modbus/TCP, fieldbuses)
- ▶ Valid FS data sources are:
 - FS-O-PI variables
 - FS-I data from external communication (e.g. FS bus system of another manufacturer)

Principle

In I/O mapping in PAS4000, the user maps an ST data source and an FS data source to the same FS data sink (e.g. FS hardware output). PSS 4000 performs the evaluation internally, based on this I/O mapping. This means that the evaluation is not performed in a PSSu system's resource, so no programming is required in the user program. The ST data source, the valid bit from the ST data source, the FS data source and the valid bit from the FS data source are logically linked and evaluated during the internal evaluation. The result of the logic operation controls the FS hardware output.

The following rules apply:

- ▶ Provided the FS data source and the valid bit from the FS data source are "TRUE", the FS hardware output can be switched on and off with the ST data source. The enable LED on the FS hardware output will light.
- ▶ As soon as the FS data source and/or the valid bit from the FS data source is "FALSE", the substitute value is used for the FS hardware output. With a digital FS hardware output this means that it is shut down safely. The enable LED on the FS hardware output will not be lit. Control via the ST data source is ignored.
- ▶ As soon as the valid bit from the ST data source is "FALSE", the substitute value is used for the FS hardware output. With a digital FS hardware output this means that it is shut down safely. The enable LED on the FS hardware output will light.



Evaluation (principle)

Please note:

ST data that is processed on the ST resource is **not** safe. It may only be an **additional** criterion for a safety-related action (e.g. start/stop) and must **not** prevent safety functions. For example, if for whatever reason a stop signal has not been activated in the ST resource, or a start signal is continuously activated, this must not be able to compromise the safety of the machine or process.

9.14 Reaction times

The following description of the reaction times refers exclusively to the inputs and outputs on PSS 4000-exclusive devices. If the inputs and outputs are distributed over various PSS 4000-exclusive devices, the processing time of the SafetyNET p interfaces and the SafetyNET p transmission time must also be taken into account.

If necessary, the relevant external communication relationships (e.g. communication via Modbus/TCP, UDP, PROFIBUS-DP), third party periphery devices, sensors/actuators and third party control systems must also be taken into account when determining the overall reaction times. These influences are not described here.



Information

The following description of reaction times is valid for PAS4000 from Version 1.2.0

9.14.1 Times that influence reaction times

The reaction times are determined by the following times:

▶ **Cycle time of a periodic task (t_{Task})**

The task cycle time t_{Task} is defined by the user during the task configuration in PAS4000. It defines the time within which a task must be executed (see Task basics, [Page 207](#)).

Value range:

ST tasks: 2 ms ... 2 000 000 ms

FS tasks: 6 ms ... 10 000 ms

▶ **Task execution time (t_{ProcTask})**

The execution time t_{ProcTask} is the length of time from the start of the task cycle to the end of the task execution, i.e. the point at which the process image of outputs and the resource global variables are available (see Formation of data paths – Data paths with tasks, [Page 374](#)).

▶ **Module bus cycle time (t_{MBUS})**

There is one cycle time each for the FS module bus and for the ST module bus of a PSS 4000-exclusive device. They are calculated and displayed by PAS4000. The cycle time t_{MBUS} is the time needed to transport data between the head module and module.

Value range:

ST module bus: 2 ms ... 45 ms

FS module bus: 6 ms ... 30 ms

Please note:

- The cycle time of the respective module bus always corresponds to that of the task with the shortest cycle time, provided the task cycle time is less than/equal to the maximum possible cycle time for the corresponding module bus:

$$t_{\text{MBUS}} = t_{\text{Task min}}$$

- If only one task is executed on a PSS 4000-exclusive device, the module bus cycle time will correspond to the task's cycle time, provided the task cycle time is less than/equal to the maximum possible cycle time of the corresponding module bus:

$$t_{\text{MBUS}} = t_{\text{Task}}$$

▶ **Processing time for a module's input data (t_{ProcIM})**

The processing time t_{ProcIM} : see module's technical details

The processing time t_{ProcIM} is the time a module requires internally to process the I-data. The processing time takes into account the input filter time and various influences such as internal run times, temperature drifts, spread of components, etc. It varies depending on the module.

▶ **Processing time for a module's output data (t_{ProcOM})**

Processing time t_{ProcOM} : see module's technical details

The processing time t_{ProcOM} is the time a module requires internally to process the O-data. The processing time takes into account various influences such as internal run times, temperature drifts, spread of components, etc. It varies depending on the module.

▶ **Basic cycle time of SafetyNET p RTFN ($t_{\text{BasicSNp RTFN}}$)**

The basic cycle time $t_{\text{BasicSNp RTFN}}$ describes the time it takes to run tests within a project as part of each cycle, to check whether data is available to send via SafetyNET p RTFN.

The base cycle time is calculated and displayed by PAS4000.

▶ **Cycle time of SafetyNET p RTFN ($t_{\text{SNp RTFN}}$)**

The cycle time refers to a transmit connection (Tx connection) on a PSS 4000 device. In other words, each TX connection on a PSS 4000 device has its own cycle time $t_{\text{SNp RTFN}}$. The cycle time $t_{\text{SNp RTFN}}$ describes the time it takes to start a Tx connection as part of each cycle, in order to send data. The cycle times of the Tx connections are calculated and displayed by PAS4000.

▶ **Transmission delay time of SafetyNET p RTFN ($t_{\text{DelaySNp RTFN}}$)**

The transmission delay time $t_{\text{DelaySNp RTFN}}$ describes the time that may elapse within a project from the point at which data transmission starts via a transmit connection (Tx connection) on a PSS 4000 device A until the point at which the data becomes available for further processing in a PSS 4000 device B. The maximum time for the transmission delay time is $t_{\text{DelaySNp RTFN}} = 2 \times t_{\text{BasicSNp RTFN}}$.

▶ **Cycle time with external communication (t_{ExtCo})**

External communication refers to data exchange in which external O-data leaves the automation system PSS 4000 and external I-data enters the system (e.g. via Modbus/TCP, UDP, fieldbuses). Depending on the communication mode that is used, external O-data is sent from the O-data area of a PSS 4000-exclusive device. Depending on the communication mode that is used, external I-data can be found in the I-data area of a PSS 4000-exclusive device and is read from there.

In this description, the cycle times for send and receive are designated with t_{ExtCo} . Their meaning depends on the communication mode that is used:

– **Modbus/TCP Client connection:**

The cycle time t_{ExtCo} describes the time used by a connection's client to cyclically start the data transmission.

The cycle time can be set separately in PAS4000 for each connection Client on a PSS 4000-exclusive device; in other words, each connection Client on a PSS 4000-exclusive device has its own cycle time t_{ExtCo} . It can either be calculated automatically or it can be configured by the user (configuration in PAS4000).

– **Modbus/TCP Server connection:**

The cycle time t_{ExtCo} describes the time it takes to read data from a Server allocation table of a PSS 4000-exclusive device and for the data requested from a Server allocation table to be transferred to the connection Client, as part of each cycle.

The cycle time can be set separately in PAS4000 for each server allocation table on a PSS 4000-exclusive device; in other words, each server allocation table on a PSS 4000-exclusive device has its own cycle time t_{ExtCo} . It can either be calculated automatically or it can be configured by the user (configuration in PAS4000).

– **Raw UDP communication "Receive data" and "Send data":**

The cycle time t_{ExtCo} describes the time it takes to read in external I-data from a UDP connection and start transmitting data (external O-data) via a UDP connection as part of each cycle.

The cycle time can be set separately in PAS4000 for each UDP connection on a PSS 4000-exclusive device; in other words, each UDP connection has its own cycle time t_{ExtCo} . It can either be calculated automatically or it can be configured by the user (configuration in PAS4000).

– **PROFIBUS-DP Slave "Receive data" and "Send data":**

The cycle time t_{ExtCo} describes the time it takes to read in external I-data from the PROFIBUS-DP Master and start transmitting the requested data (external O-data) as part of each cycle.

The cycle time can be defined in PAS4000 for each "PROFIBUS-DP Slave" system section, i.e. each "PROFIBUS-DP Slave" system section has its own cycle time t_{ExtCo} for data transfer with PROFIBUS-DP. It can either be calculated automatically or it can be configured by the user (configuration in PAS4000).

– **PROFIBUS-DP Slave with PROFIsafe "Receive data" and "Send data":**

The cycle time t_{ExtCo} describes the time it takes to read in external I-data from the PROFIBUS-DP Master/F-host and start transmitting the requested data (external O-data) as part of each cycle.

The cycle time can be defined in PAS4000 for each "PROFIBUS-DP Slave with PROFIsafe" system section, i.e. each "PROFIBUS-DP Slave with PROFIsafe" system section has its own cycle time t_{ExtCo} for data transfer. The cycle time t_{ExtCo} applies to data transfer via both PROFIBUS-DP and PROFIsafe. It can either be calculated automatically or it can be configured by the user (configuration in PAS4000).

Please note:

If PROFIsafe communication is used, only a limited value range will be available for the cycle time t_{ExtCo} . In this case, this limited value range will also apply to PROFIBUS-DP Slave communication. The cycle time t_{ExtCo} depends on the device.

– **EtherNet/IP Adapter "Receive data" and "Send data":**

The cycle time (t_{ExtCo}) describes the time it takes to read in external I-data from the Scanner and start transmitting external O-data sent to the Scanner, as part of each cycle.

The cycle time can be determined in PAS4000 for the "EtherNet/IP Adapter" system section. It can either be calculated automatically or it can be configured by the user (configuration in PAS4000).

9.14.2 Processing rules during data processing

The following applies for data processing on a PSSu system:

▶ **Processing times of modules**

The processing times of modules (t_{ProcOM} and t_{ProcIM}) are constant. They are not influenced by the current task cycle.

▶ **Module bus cycle times**

The ST module bus and the FS module bus each have their individual cycle times. The module bus cycle time t_{MBUS} always corresponds to that of the task with the shortest cycle time, provided that the task cycle time is less than/equal the maximum possible cycle time of the corresponding module bus:

$$t_{\text{MBUS}} = t_{\text{Task min}}$$

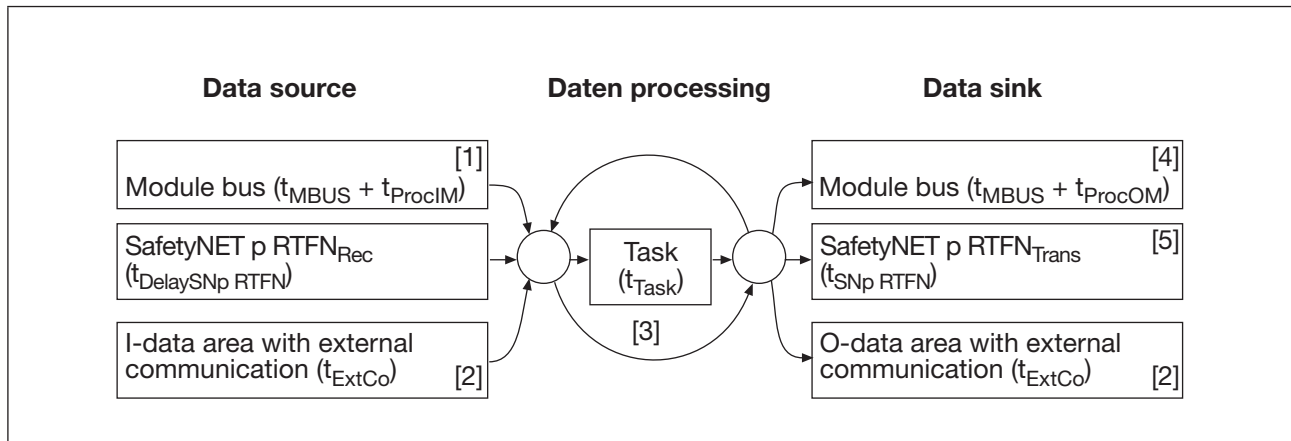
▶ **Cycle times of tasks**

The cycle times of tasks (t_{Task}) are constant. They are not influenced by the current task cycle.

9.14.3 Formation of data paths

Data subpaths are available to enable you to track a single I/O datum over a complete data path. A data path, therefore, is made up of the relevant data subpaths.

The reaction time of a data path is calculated by adding the reaction times of the relevant data subpaths. A data path always starts with the I-datum in a data source and ends with the O-datum in a data sink. A data source/data sink is always located in a system section (see System sections, [Page 16](#)).



Data paths on a PSSu system

[1]

If the data source is an I-datum from a PSSu module, the processing time for the I-datum on the module bus (t_{ProcIM}) needs to be considered in addition to the module bus cycle time (t_{MBUS}).

[2]

With external communication, the I-data from the I-data area is also referred to as "external I-data" and the O-data from the O-data area as "external O-data".

[3]

The following data paths are impossible:

- ▶ Data source: Module bus (I-datum) -> Data sink: Module bus (O-datum)
- ▶ Data source: SafetyNET p RTFN_{Rec} -> Data sink: SafetyNET p RTFN_{Trans}

[4]

If the data sink is an O-datum on a PSSu module, the processing time for the O-datum on the module (t_{ProcOM}) needs to be considered in addition to the module bus cycle time (t_{MBUS}).

[5]

If there are several Snp connections between two devices, the data that is transmitted is the data in the connection or in one of the connections that has the same cycle time as the data source of the data on this device.

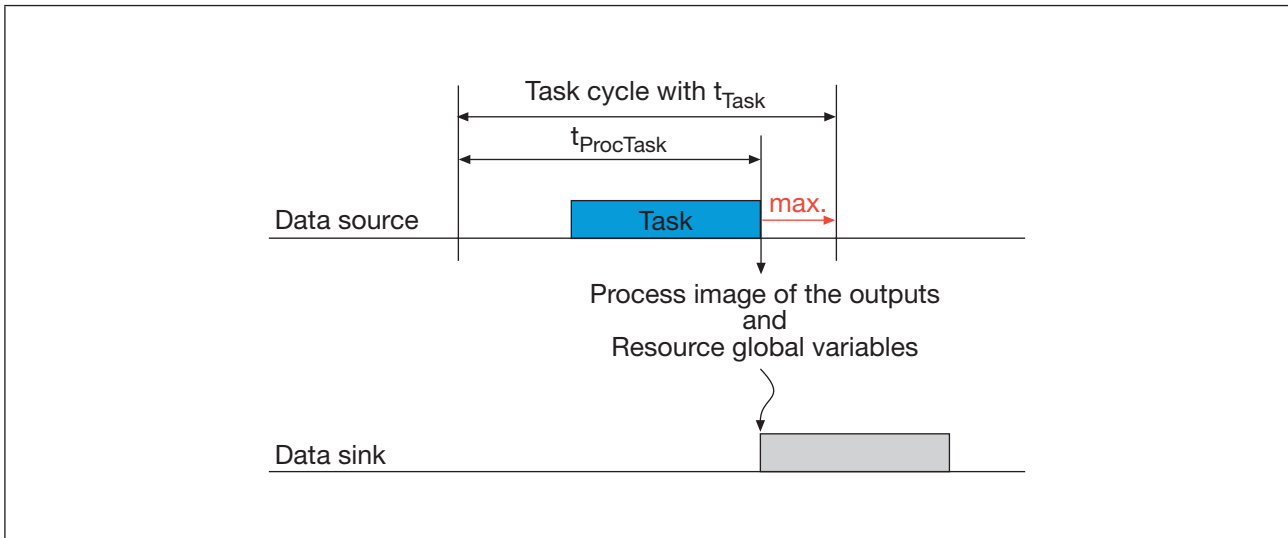
Data paths with tasks

For data paths with tasks, please note:

▶ Task as data source

The reaction time for a datum starts at the beginning of the task cycle and ends after the execution time t_{ProcTask} with the process image of outputs and the resource global variables available in the data sink for further processing.

When the processing time t_{ProcTask} almost corresponds to the task cycle time t_{Task} , that is the maximum reaction time.

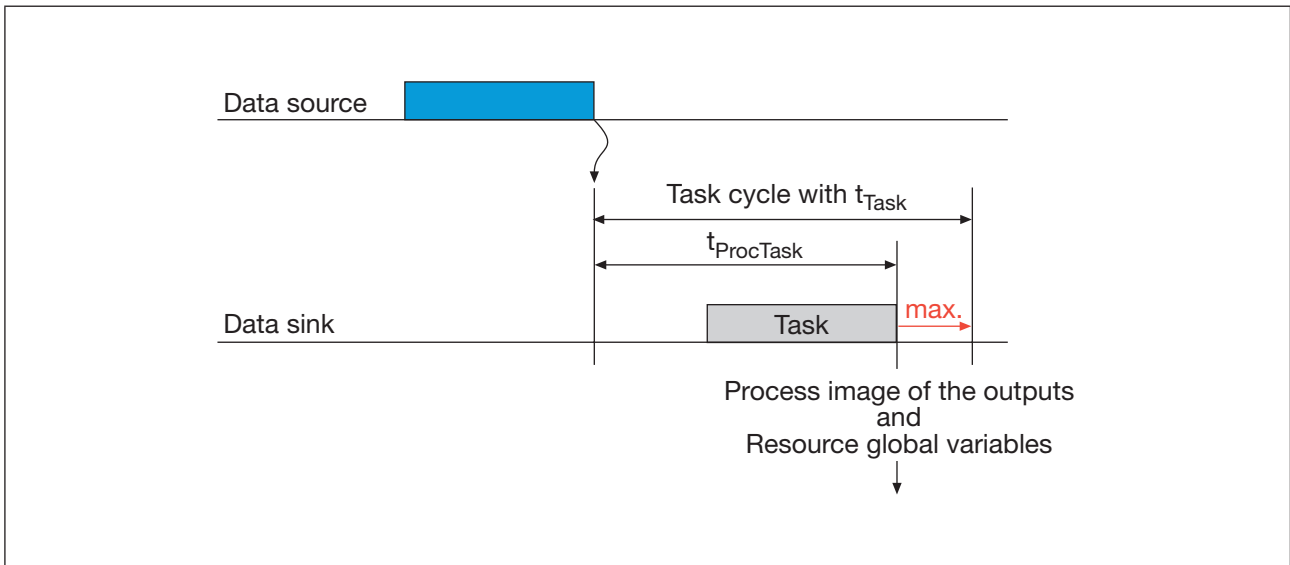


Reaction time with a task's POUs as data source

▶ Task as data sink

The reaction time for a datum starts at the beginning of the task cycle and ends after the execution time t_{ProcTask} with the process image of outputs and the resource global variables available for further processing.

When the processing time t_{ProcTask} almost corresponds to the task cycle time t_{Task} , that is the maximum reaction time.



Reaction time with a task as data sink

9.14.4 Maximum reaction times for data subpaths

The following description contains the formulas for calculating the maximum reaction time for all permitted data subpaths.

9.14.4.1 Maximum reaction time for data subpaths with module bus as data source

▶ Module bus -> Task

PSS 4000-Firmware from version 1.2.0 to 1.3.x:

$$t_{\text{MBUS_Task_max}} = t_{\text{ProclM}} + 2t_{\text{Task}} + t_{\text{MBUS}}$$

PSS 4000 firmware from version 1.4.0:

$$t_{\text{MBUS_Task_max}} = t_{\text{ProclM}} + 2t_{\text{Task}}$$

$t_{\text{MBUS_Task_max}}$:	Max. reaction time for this data subpath
.	
t_{ProclM} :	Processing time for a module's I-data
.	
t_{MBUS} :	PSSu system's module bus cycle time
.	
t_{Task} :	Task cycle time of the task on the PSSu system
.	

▶ Module bus -> SafetyNET p RTFN_{Trans}

$$t_{\text{MBUS_SNp_max}} = t_{\text{ProclM}} + 2t_{\text{SNp RTFN}}$$

$t_{\text{MBUS_SNp_max}}$:	Max. reaction time for this data subpath
.	
t_{ProclM} :	Processing time for a module's I-data
.	
t_{MBUS} :	PSSu system's module bus cycle time
.	
$t_{\text{SNp RTFN}}$:	Cycle time for the cyclical start of the SNp Tx connection (send data)
.	

▶ Module bus -> O-data area with external communication

$$t_{\text{MBUS_ExtCo_max}} = t_{\text{ProclM}} + 2t_{\text{ExtCo}}$$

$t_{\text{MBUS_ExtCo_max}}$:	Max. reaction time for this data subpath
.	
t_{ProclM} :	Processing time for a module's I-data
.	
t_{MBUS} :	PSSu system's module bus cycle time
.	
t_{ExtCo} :	Cycle time for the cyclical start of data transfer from the O-data area with external communication
.	

9.14.4.2 Data subpaths with SafetyNET p RTFN as data source

▶ SafetyNET p RTFN_{Rec} -> Task

$$t_{\text{SNp_Task_max}} = t_{\text{DelaySNp RTFN}} + 2t_{\text{Task}}$$

$t_{\text{SNp_Task_max}}$:	Max. reaction time for this data subpath
$t_{\text{DelaySNp RTFN}}$:	Transmission delay time of SafetyNET p RTFN in a project
t_{Task} :	Task cycle time of the task on the PSSu system

▶ SafetyNET p RTFN_{Rec} -> Module bus

PSS 4000-Firmware from version 1.2.0 to 1.3.x:

$$t_{\text{SNp_MBUS_max}} = t_{\text{DelaySNp RTFN}} + 3t_{\text{MBUS}} + t_{\text{ProcOM}}$$

PSS 4000 firmware from version 1.4.0:

$$t_{\text{SNp_MBUS_max}} = t_{\text{DelaySNp RTFN}} + 2t_{\text{MBUS}} + t_{\text{ProcOM}}$$

$t_{\text{SNp_MBUS_max}}$:	Max. reaction time for this data subpath
$t_{\text{DelaySNp RTFN}}$:	Transmission delay time of SafetyNET p RTFN in a project
t_{MBUS} :	PSSu system's module bus cycle time
t_{ProcOM} :	Processing time for O-data on a module with outputs

▶ SafetyNET p RTFN_{Rec} -> O-data area with external communication

$$t_{\text{SNp_ExtCo_max}} = t_{\text{Delay RTFN}} + 2t_{\text{ExtCo}}$$

$t_{\text{SNp_ExtCo_max}}$:	Max. reaction time for this data subpath
$t_{\text{DelaySNp RTFN}}$:	Transmission delay time of SafetyNET p RTFN in a project
t_{ExtCo} :	Cycle time for the cyclical start of data transfer from the O-data area with external communication

9.14.4.3 Data subpaths with external I-data as data source

▶ I-data area with external communication -> Module bus

$$t_{\text{ExtCo_MBUS_max}} = t_{\text{ExtCo}} + \min(t_{\text{ExtCo}}, t_{\text{MBUS}}) + t_{\text{MBUS}} + t_{\text{ProcOM}}$$

$t_{\text{ExtCo_MBUS_max}}$	Max. reaction time for this data subpath
.	.
t_{ExtCo}	Cycle time for cyclical reading of the external I-data from the I-data area with external communication
.	.
t_{MBUS}	PSSu system's module bus cycle time
.	.
t_{ProcOM}	Processing time for O-data on a module
.	.

▶ I-data area with external communication -> Task

$$t_{\text{ExtCo_Task_max}} = 2t_{\text{ExtCo}} + 2t_{\text{Task}}$$

$t_{\text{ExtCo_Task_max}}$	Max. reaction time for this data subpath
.	.
t_{ExtCo}	Cycle time for cyclical reading of the external I-data from the I-data area
.	.
t_{Task}	Task cycle time of the task on the PSSu system
.	.

▶ I-data area with external communication -> SafetyNET p RTFN_{Trans}

$$t_{\text{ExtCo_SNp_max}} = 2t_{\text{ExtCo}} + 2t_{\text{SNp RTFN}}$$

$t_{\text{ExtCo_SNp_max}}$	Max. reaction time for this data subpath
.	.
t_{ExtCo}	Cycle time for cyclical reading of the external I-data from the I-data area with external communication
.	.
$t_{\text{SNp RTFN}}$	Cycle time for the cyclical start of the SNp Tx connection (send data)
.	.

▶ I-data area with external communication -> O-data area with external communication

$$t_{\text{ExtCo1_ExtCo2_max}} = 2t_{\text{ExtCo1}} + 2t_{\text{ExtCo2}}$$

$t_{\text{ExtCo1_ExtCo2_max}}$	Max. reaction time for this data subpath
.	.
t_{ExtCo1}	Cycle time for cyclical reading of the external I-data from the I-data area (external communication with communication mode 1)
.	.
t_{ExtCo2}	Cycle time for the cyclical start of data transfer from the O-data area (external communication with communication mode 2)
.	.

9.14.4.4 Data subpaths with a task as data source

▶ Task -> Module bus

PSS 4000-Firmware from version 1.2.0 to 1.3.x:

$$t_{\text{Task_MBUS_max}} = 2t_{\text{MBUS}} + t_{\text{ProcOM}}$$

PSS 4000 firmware from version 1.4.0:

$$t_{\text{Task_MBUS_max}} = t_{\text{MBUS}} + t_{\text{ProcOM}}$$

$t_{\text{Task_MBUS_max}}$: Max. reaction time for this data subpath

.

t_{MBUS} : PSSu system's module bus cycle time

.

t_{ProcOM} : Processing time for O-data on a module

.

▶ Task -> SafetyNET p RTFN_{Trans}

$$t_{\text{Task_SNp_max}} = 2t_{\text{SNp}}$$

$t_{\text{Task_SNp_max}}$: Max. reaction time for this data subpath

.

$t_{\text{SNp RTFN}}$: Cycle time for the cyclical start of the SNp Tx connection (send data)

.

▶ Task1 -> Task2

$$t_{\text{Task1_Task2_max}} = 2t_{\text{Task2}}$$

$t_{\text{Task1_Task2_max}}$: Max. reaction time for this data subpath

.

t_{Task2} : Task cycle time of task2

.

▶ Task -> O-data area with external communication

$$t_{\text{Task_ExtCo_max}} = 2t_{\text{ExtCo}}$$

$t_{\text{Task_ExtCo_max}}$: Max. reaction time for this data subpath

.

t_{ExtCo} : Cycle time for the cyclical start of data transfer from the O-data area with external communication

.

9.14.4.5 Example for calculating a data path on one PSSu system

The example was created for a device that has a PSS 4000 firmware from version 1.4.0. For the data path in this example, the following applies:

- ▶ I-datum (e.g. HW input of a module) and O-datum (e.g. HW output of a module) are on the same PSSu system.
- ▶ The data path is composed as follows:
Module bus (I-datum) -> Task -> Module bus (O-datum)
- ▶ To calculate the maximum reaction time, the data path is divided into the following data subpaths:

- **Data subpath 1: Module bus (I-datum) -> Task**

Applicable for the maximum reaction time of data subpath 1:

$$t_{\text{MBUS_Task_max}} = t_{\text{ProciM}} + 2t_{\text{Task}}$$

- **Data subpath 2: Task -> Module bus (O-datum)**

The following applies for the maximum reaction time of data subpath 2:

$$t_{\text{Task_MBUS_max}} = t_{\text{MBUS}} + t_{\text{ProcOM}}$$

The following applies for the data path's maximum reaction time:

$$t_{\text{React_max}} = t_{\text{ProciM}} + 2t_{\text{Task}} + t_{\text{MBUS}} + t_{\text{ProcOM}}$$

9.14.4.6 Example for calculating a data path via three PSSu systems

The example was created for devices with a PSS 4000 firmware from Version 1.4.0. For the data path in this example, the following applies:

- ▶ The I-datum (e.g. HW input) is on PSSu system A. The datum is transferred to PSSu system B via SafetyNET p RTFN.
- ▶ On PSSu system B, the datum is processed in a task and is then transferred to PSSu system C via SafetyNET p RTFN.
- ▶ On PSSu system C, the datum is transferred to a module with outputs.
- ▶ The data path is composed as follows:

Module bus (I-datum) to PSSu system A -> SafetyNET p RTFN -> Task on PSSu system B -> SafetyNET p RTFN -> Module bus (O-datum) on PSSu system C

- ▶ To calculate the maximum reaction time, the data path is divided into the following data subpaths:

- **Data subpath 1: Module bus_(A) (I-datum) -> SafetyNET p RTFN_{Trans(A)}**

The following applies for the maximum reaction time of data subpath 1:

$$t_{\text{MBUS_SNp_max(A)}} = t_{\text{ProclM}} + 2t_{\text{SNp RTFN(A)}}$$

- **Data subpath 2: SafetyNET p RTFN_{Rec(B)} -> Task_(B)**

The following applies for the maximum reaction time of data subpath 2:

$$t_{\text{SNp_Task_max(B)}} = t_{\text{DelaySNp RTFN}} + 2t_{\text{Task(B)}}$$

- **Data subpath 3: Task_(B) -> SafetyNET p RTFN_{Trans(B)}**

The following applies for the maximum reaction time of data subpath 3:

$$t_{\text{Task_SNp_max(B)}} = 2t_{\text{SNp RTFN(B)}}$$

- **Data subpath 4: SafetyNET p RTFN_{Rec(C)} -> Module bus_(C) (O-datum)**

The following applies for the maximum reaction time of data subpath 4:

$$t_{\text{SNp_MBUS_max(C)}} = t_{\text{DelaySNp RTFN}} + 2t_{\text{MBUS(C)}} + t_{\text{ProcOM}}$$

The following applies for the data path's maximum reaction time:

$t_{\text{React_max}}$.	$= t_{\text{ProclM}} + 2t_{\text{SNp RTFN(A)}} + t_{\text{DelaySNp RTFN}} + 2t_{\text{Task(B)}} + 2t_{\text{SNp RTFN(B)}} + t_{\text{DelaySNp RTFN}} + 2t_{\text{MBUS(C)}} + t_{\text{ProcOM}}$.
------------------------------	---

9.15 System time synchronisation

As devices in industrial automation systems are often involved in complex networks, each device is required to have the same system time. This is all the more important when precise data evaluation and processing of time-critical data (e.g. process data, reaction times for safety-related tasks, time stamp for logs) must not be delayed by the network load, the size of the data packets or mechanisms within the communication protocol.

In Ethernet-based industrial automation systems this requirement can be met using a system clock that provides the date and time. At least one real-time clock is used, depending on the applied technology, and this provides the reference time within the automation system (Server). All other network subscribers (Clients) synchronise their device time based on this reference time. In this way, each network subscriber has a standardised system time which is subject only to very minor deviations.

9.15.1 System time synchronisation with SNTP

The Simple Network Time Protocol (SNTP) is a synchronisation protocol used to communicate and synchronise time in Ethernet-based networks. SNTP operates in accordance with the Client-Server principle. It uses UDP as the communication protocol. The (time) Server uses the port with port number 123 for this service by default. On PSS 4000-exclusive devices the port is permanently set and cannot be modified by the user. The (time) Server is called an SNTP Server in PAS4000.

In the automation system PSS 4000, a project can be operated with or without an SNTP Server.

If an SNTP Server is configured for a project in PAS4000, the Simple Network Time Protocol (SNTP) is used to ensure the same time base for diagnostic messages and log entries for all devices within a project. This guarantees the correct chronological sequence of diagnostic messages coming from a variety of devices with their own real-time clock. STNP is also used to synchronise the time automatically after exchanging devices. PAS4000 only needs to be used to set the time on the new device if the exchanged unit is to be used as an SNTP Server.

The automation system PSS 4000 supports SNTP Unicast as a communication mechanism. This means that there can only be one SNTP Server in a project. All other network subscribers automatically become SNTP Clients.

With the Unicast communication mechanism, an SNTP Client sends a time synchronisation request to the SNTP Server and receives time information from the SNTP Server.

On PSS 4000, time synchronisation between an SNTP Client and the configured SNTP occurs with a cycle time of ≤ 60 s. The user cannot configure the cycle time.

Any device within the project that has a real-time clock can be configured as an SNTP Server for the project. However, PAS4000 also supports the use of external devices. These may be special switches designed for this purpose, for example.

If **no** SNTP Server is configured for a project in PAS4000, then each device with a real-time clock operates with its own time base. As a result, there is no standard time base for diagnostic messages and log entries within the project. The correct chronological sequence of diagnostic messages cannot be guaranteed.

10 Commissioning

10.1 General requirements

Secure the site in accordance with the regulations (barrier, warning signs etc.). The system may only be commissioned/recommissioned by a competent person.

- ▶ Please refer to the information and specifications stated in the relevant documentation (e.g. PSS 4000 Safety Manual, PSSuniversal Installation Manual, operating manuals for the modules you are using).
- ▶ During commissioning/recommissioning, make sure that no personal injury and/or material damage can occur, even if the plant/machine moves unintentionally.
(e.g. physical barrier for the danger zone or for the whole plant/machine if necessary).

10.2 Initial commissioning

The following description assumes that

- ▶ The following documentation has been considered when deciding on the installation site, mechanical structure and electrical connection
 - PSSuniversal Installation Manual
 - "SafetyNET p" chapter in this System Manual (if necessary)
 - Operating manuals for the modules you are using
 - Installation guidelines for the standard fieldbus you are using (if necessary)
 - Relevant documents from the risk analysis (e.g. assembly instructions, wiring diagram)
- ▶ All the required PSSu systems are fully assembled, wired and connected
- ▶ A removable data medium is plugged into all PSSu systems
- ▶ The project in PAS4000 is complete
(see PAS4000's online help)
- ▶ The project has been downloaded
(see PAS4000's online help)
- ▶ Any advice provided during the project download has been heeded and any errors rectified

(e.g. all the project's PSSu systems are accessible with PAS4000, set hardware registry matches the actual hardware registry)
- ▶ Description is available of all the activities required for commissioning (e.g. check lists and test specifications from the risk analysis)
- ▶ Supply voltages to the plant/machinery are **not** switched on at this point

Procedure

1. Check installation and wiring

Check the assembly and wiring based on the specific assembly instructions and wiring diagram for the plant/machinery.

Please refer to the relevant documents from the risk analysis (see PSS 4000 Safety Manual).

2. Switch on supply voltages

Switch on all the PSSu systems' supply voltages together.

It may take a little time (max. 10 s for a PSSu system) for all the PSSu systems to be ready for operation. The LEDs on the head module will show you when a PSSu system is ready for operation.

3. Use commissioning aids

PAS4000 provides some important aids to help with any problems during commissioning:

- Watch variables
 - Watches the current state of variables dynamically
- Control variables

The value of variables is set once to a specified value. This can be changed afterwards by the user program.

Make sure you heed the information and requirements stipulated in the PSS 4000 Safety Manual, even when controlling non-safety-related variables!

- Force variables

The value of variables is fixed and cannot be changed by the user program. If "Force variables" is stopped, the variable will retain its force value until it is overwritten with a new value.

Make sure you heed the information and requirements stipulated in the PSS 4000 Safety Manual, even when forcing non-safety-related variables!

- Diagnostic log

Log that shows when diagnostic messages arrived, when they were cleared; It also contains entries regarding operation and debugging.

- Diagnostic list

List of current diagnostic messages (project diagnostic list, device diagnostic list)

- Project comparison

Option in PAS4000 to find out whether the original project in PAS4000 is identical to the project on the machine

The online projekt is compared with the offline project. Differences in the check sum are displayed.

4. Test safety functions and fault detection facilities

If the project contains safety functions and fault detection facilities, use the test specification to test all the plant-specific safety functions and fault detection facilities.

It is **essential** that all the specified safety functions and fault detection facilities are fully tested.

It is **essential** that you heed the information and requirements stipulated in the PSS 4000 Safety Manual.

5. Test FS blocks with mixed input interface

If FS blocks with mixed input interface have been used, test these using the defined test procedures (e.g. feasibility tests).

It is **essential** that you heed the information and requirements stipulated in the PSS 4000 Safety Manual.

6. Optimise task cycle time

During commissioning you can optimise the task cycle time.

7. Create backup copy

Create a backup copy of the project, declare the project as the original project and save it in accordance with the project backup plan.

8. Documentation

- Document and log all actions taken during commissioning.

For all safety-related actions it is **essential** that you heed the information and requirements stipulated in the PSS 4000 Safety Manual.

- Document the check sum of the original project

9. **Normal operation**

The project is running normally once all PSSu systems are switched on and no faults are present.

10.3 Recommissioning

Recommissioning is always necessary when a change is made to the plant/machine or process. In all cases, the following changes must result in recommissioning:

- ▶ A change to a safety requirement
- ▶ A change due to the occurrence of a systematic error
- ▶ A change due to new operation or production requirements
- ▶ A change to the plant/machine by:
 - Adding or removing components (e.g. PSSu system, a PSSu system module)
 - Exchanging components (e.g. PSSu system, a PSSu system module, a PSSu system removable data medium)
- ▶ A change to the process cycle

The following description assumes that

- ▶ A risk analysis has been performed prior to any change
- ▶ The following documentation has been considered when adding, removing or exchanging components
 - PSSuniversal Installation Manual
 - "SafetyNET p" chapter in this System Manual (if necessary)
 - Operating manuals for the modules you are using
 - Installation guidelines for the standard fieldbus you are using (if necessary)
 - Relevant documents from the risk analysis for the changes
- ▶ All modified components are fully assembled/plugged in and wired
- ▶ A removable data medium is plugged into all PSSu systems
- ▶ The changes to the project in PAS4000 are complete
(see PAS4000's online help)
- ▶ The project has been downloaded
(see PAS4000's online help)
- ▶ Any advice provided during the project download has been heeded and any errors rectified
(e.g. all the project's PSSu systems are accessible with PAS4000, set hardware registry matches the actual hardware registry)
- ▶ Description is available of all the activities required for recommissioning (e.g. check lists and test specifications from the risk analysis)
- ▶ Supply voltages to the plant/machinery are **not** switched on at this point

Procedure

1. Check installation and wiring

Check the assembly and wiring of the modified components based on the specific assembly instructions and wiring diagram for the plant/machinery.

Please refer to the relevant documents from the risk analysis for the changes (see PSS 4000 Safety Manual).

2. Switch on supply voltages

Switch on all the PSSu systems' supply voltages together.

It may take a little time (max. 10 s for a PSSu system) for all the PSSu systems to be ready for operation. The LEDs on the head module will show you when a PSSu system is ready for operation.

3. Use commissioning aids

PAS4000 provides some important aids to help with any problems during recommissioning:

- Watch variables

Watches the current state of variables dynamically

- Control variables

The value of variables is set once to a specified value. This can be changed afterwards by the user program.

Make sure you heed the information and requirements stipulated in the PSS 4000 Safety Manual, even when controlling non-safety-related variables!

- Force variables

The value of variables is fixed and cannot be changed by the user program. If "Force variables" is stopped, the variable will retain its force value until it is overwritten with a new value.

Make sure you heed the information and requirements stipulated in the PSS 4000 Safety Manual, even when forcing non-safety-related variables!

- Diagnostic log

Log that shows when diagnostic messages arrived, when they were cleared; It also contains entries regarding operation and debugging.

- Diagnostic list

List of current diagnostic messages (project diagnostic list, device diagnostic list)

- Project comparison

The FS project contains a check sum, on which the user can base the comparison

- Comparison of POU's

Each POU contains a check sum on which the user can base the comparison

4. Test safety functions and fault detection facilities

Following a change, if the safety analysis has shown that safety functions need to be validated and tested, the change itself must be tested, as must the course of the entire process.

Please note:

Based on the check sum for the FS project, a project comparison can be used to determine whether the FS project has changed and therefore whether safety functions need to be validated and tested. However, the project comparison may only be used as an additional aid. Under no circumstances can it replace a prior safety analysis for the changes.

It must be ensured that only the desired changes are made to the project. It is **essential** that you heed the information and requirements stipulated in the PSS 4000 Safety Manual.

5. Test FS blocks with mixed input interface

If FS blocks with mixed input interface are affected by the changes, test these using the defined test procedures (e.g. feasibility tests).

It is **essential** that you heed the information and requirements stipulated in the PSS 4000 Safety Manual.

6. Optimise task cycle time

During recommissioning you can optimise the task cycle time.

7. Create backup copy

Create a backup copy of the amended project, declare the amended project as the new original project and save it in accordance with the project backup plan.

8. Documentation

- Document all changes
- Document and log all actions taken during recommissioning.

For all safety-related actions it is **essential** that you heed the information and requirements stipulated in the PSS 4000 Safety Manual.

- Document the check sum of the new original project

9. Normal operation

The project is running normally once all PSSu systems are switched on and no faults are present.

11 Maintenance, repair, decommissioning

11.1 Maintenance and repair

Before starting maintenance or repair work, please note the following:

- ▶ Maintenance and repair work may only be carried out by competent persons.
- ▶ Make sure that power to the plant or machine is removed in accordance with the regulations.
- ▶ The plant or machine should be safeguarded against inadvertent reconnection.
- ▶ If you exchange components (PSSu system, a PSSu system module, a PSSu system removable data medium):
 - Heed the information and requirements stipulated in the PSS 4000 Safety Manual
 - Switch the PSSu system to a STOP condition before exchanging components
 - Note that if other PSSu systems access the process data on the PSSu system, substitute values will be used in the other PSSu systems!
 - When exchanging the removable data medium on a PSSu system, be aware that a new device naming procedure may need to be carried out (see Device naming when exchanging devices, [Page 237](#))
 - Heed the information on recommissioning (see Recommissioning, [Page 389](#))

Maintenance and repair of PSSu systems

No maintenance work is necessary on a PSSu system. Please send any faulty PSSu systems and/or modules back to Pilz.



Information

When exchanging modules and PSSu systems, please comply with the information regarding assembly/removal as stipulated in the PSSuniversal Installation Manual.

11.2 Decommissioning

Please comply with the mission time t_M in the safety-related characteristic data of FS modules.

Please comply with the information regarding assembly/removal as stipulated in the PS-Suniversal Installation Manual and in the operating manuals for the relevant modules.

When decommissioning, please comply with local regulations regarding the disposal of electronic devices (e.g. Electrical and Electronic Equipment Act).

12 OPC Server

12.1 Overview

OPC (Openness, Productivity and Collaboration) is a standardised software interface, through which PC-based software from a range of manufacturers can exchange data. The OPC Foundation is responsible for maintaining and propagating the standard. Around one hundred companies are currently members, including Pilz.

The current OPC standard is specified as platform-dependent for the Windows operating systems (e.g. Windows 2000, Windows XP, Windows Vista). It is based on Windows technologies such as OLE (Object Linking and Embedding), COM (Component Object Model) and DCOM (Distributed COM).

On networked plants, an OPC Server will typically collect diagnostic data, process data etc. and display it in the defined namespace. An OPC Client is typically used for process visualisation and/or to record operating data. It accesses and then processes the data provided by the OPC Server. In order to process the data the OPC Client must contain the relevant OPC-enabled software (e.g. visualisation software with ActiveX Control, process visualisation system PMI-PRO from Pilz).

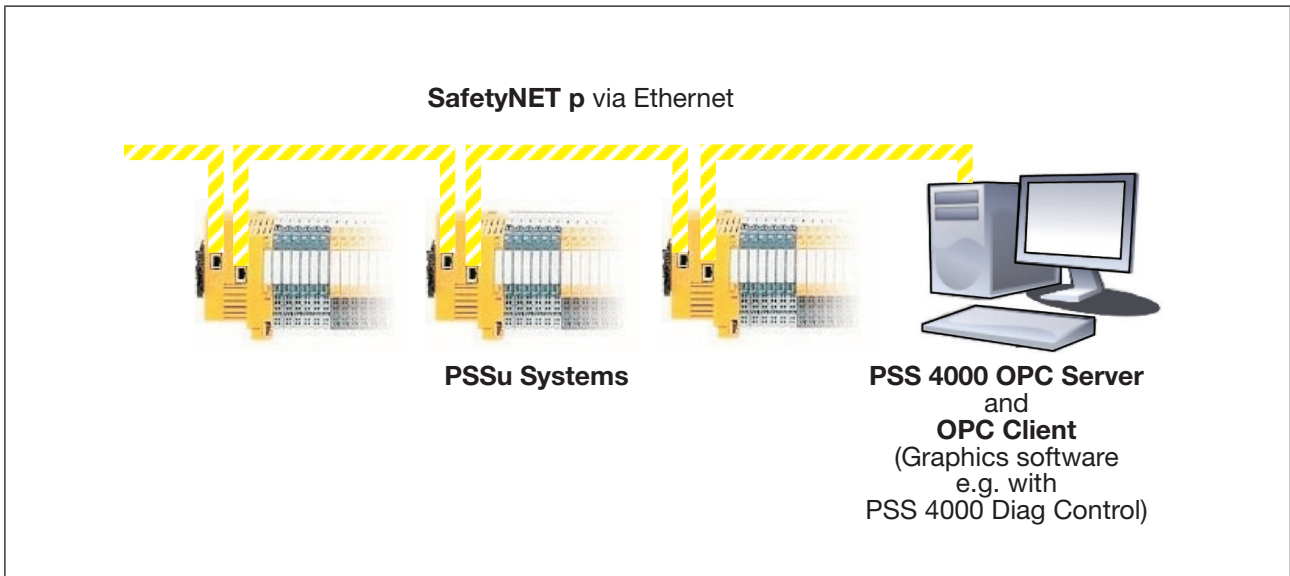
In the PSS 4000 automation system a PC or a PMI operator terminal obtains the OPC Server system section as part of the installation of PSS 4000 firmware. By installing the PSS 4000 Firmware a PC becomes a "PSS 4000-operable device from a third-party manufacturer" and a PMI operator terminal PMI becomes a "PSS 4000-operable device from Pilz" (see System Overview, [Page 14](#)).

The following boundary conditions apply to the OPC Server:

- ▶ The OPC Server supports various OPC properties for the access of OPC Clients (see OPC properties, [Page 397](#)).
- ▶ The OPC Server can be installed on a PC or Pilz PMI operator terminal (see System requirements, [Page 398](#)).
- ▶ An OPC Server can collect the data of exactly one project that has been created with PAS4000 and provide it for the OPC Clients.
- ▶ Each project created with PAS4000 may contain precisely one OPC Server.
- ▶ It is possible to access the OPC Server with more than one Client.
- ▶ All project-based settings for the OPC Server are made in PAS4000 and are transferred to the OPC Server as an OPC device project when the project is downloaded.

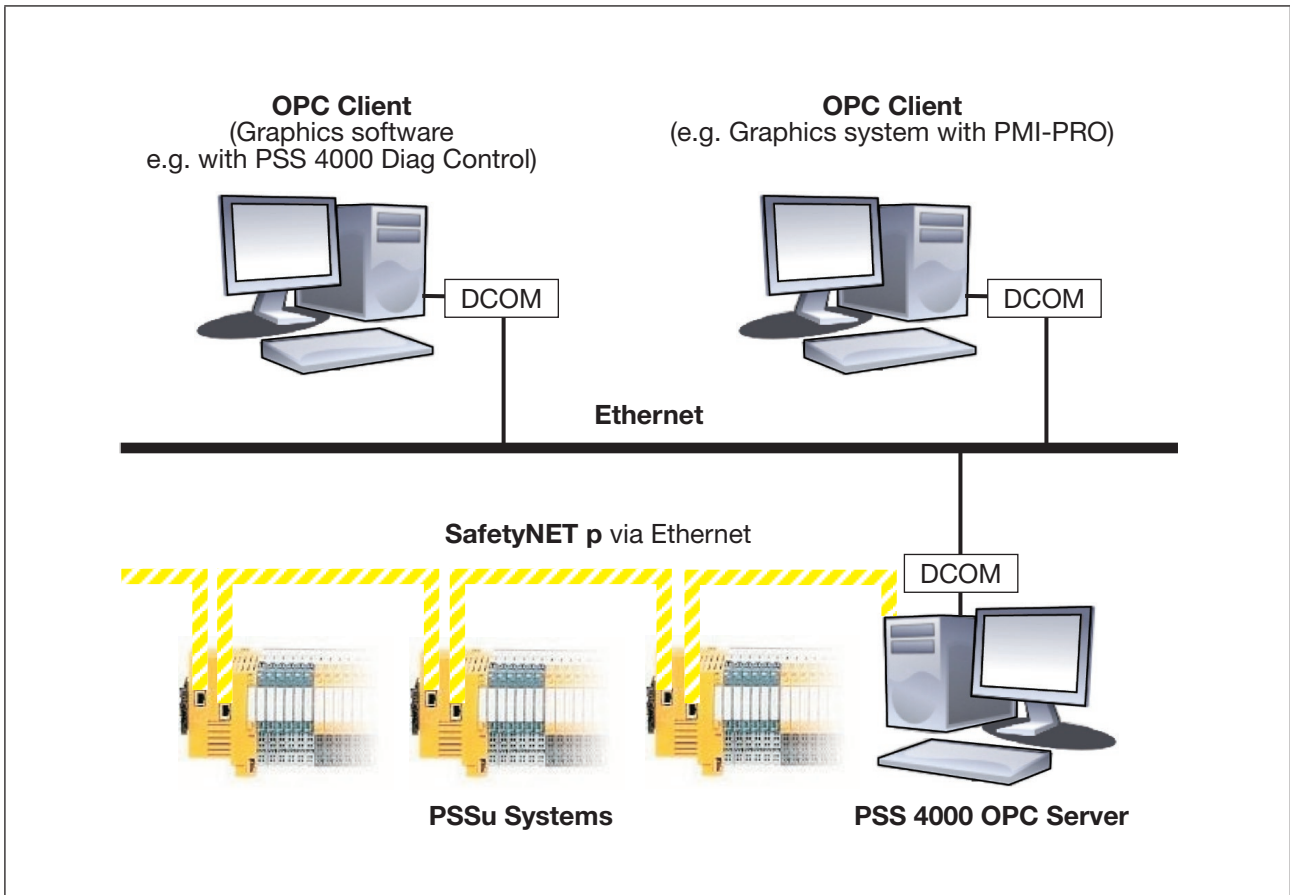
12.1.1 Design options

In a minimum configuration, the OPC Server will be on the same PC as the OPC Client with the graphics software.



Process visualisation with a minimum OPC configuration (principle)

Other types of combinations are also conceivable, such as:



Networking OPC Server and OPC Clients via Ethernet (principle)

12.1.2 OPC properties

The PSS 4000 OPC Server supports the following OPC properties for access from OPC Clients:

- ▶ OPC specification with
 - Data Access 1.0.a
 - Data Access 2.05
 - Data Access 3.0
- ▶ OPC interfaces with
 - All "Mandatory Interfaces" in accordance with the data access specification
 - Symbolic access to process data in the namespace
- ▶ OPC data types
 - VT_I1, VT_I2, VT_I4, VT_I8
 - VT_UI1, VT_UI2, VT_UI4, VT_UI8
 - VT_R4, VT_R8
 - VT_BOOL
 - VT_BSTR
 - VT_DATE

12.2 System requirements

Various system requirements must be met so that the PSS 4000 firmware can be installed on a device.

12.2.1 PC with Microsoft Windows operating system

The PSS 4000 firmware can be installed on a PC with one of the following operating systems:

- ▶ Microsoft Windows XP with ".NET Framework" from version 2.0
- ▶ Microsoft Windows 7 with ".NET Framework" from version 2.0 in one of the versions
 - Home
 - Premium
 - Professional
 - Ultimate
 - Enterprise

12.2.2 Device with Microsoft Windows Embedded Standard 7

The PSS 4000 firmware can be installed on a system (e. g. PC) with the operating system **Microsoft Windows Embedded Standard 7** . ".NET Framework Version 2.0" is required for installation. The following Windows Embedded Standard 7 OS components must be available on the system:

Application and Services Development	
.Net Framework	NET Framework 2.0 Client Profile
Application Support	COM OLE Application Support
	COM+ Application Support
	Win32 Application Runtimes and Libraries
Internet Services	
Browser	Internet Explorer 8.0
	Internet Explorer 8 Browser
	Internet Explorer 8 Engine and Display
	Internet Explorer 8 Foundation
Core OS	
Devices and Printers	Device Framework
	Driver Frameworks
System Services	Core File System
	Remote Client
	Simplified Message Block (SMB)
	Remote Procedure Call
	Windows Installer
File Systems and Data Store	
Data Access and Storage	Windows Data Access Components
Diagnostics	
Diagnostics	Common Diagnostic Tools
Fonts	
Fonts	TrueType Fonts
	Western Fonts
Graphics and Multimedia Technologies	
Graphics and Multimedia	Graphics Platform
Device Management	
Management	Microsoft Management Console (MMC)
	System Management Administrative Tools
	System Management Utilities
	Windows Management Instrumentation (WMI)

Communication Services and Networking	
Networking	Extensible Authentication Protocol (EAP)
	Internet Authentication Service
	Networking Base
	Networking Foundation
	IRDA and UNIMODEM
	Domain Services
	Network and Sharing Center
	Peer Networking
	Remote Access Service (RAS)
	Telephony API Client
	Windows Firewall
Security	
Security	Windows Security Center
	Security Base
Shell and User Interface	
User Interface	Embedded Core Help Content
	Help and Support Engine
	Application User Experience
	Shell Controls and UI Support Functions
	Shell Foundation
	Windows Explorer Shell
	System Control Panel
	Search Indexing

12.2.3 Operator terminals PMI 4 from Pilz

The PSS 4000 firmware can be installed on a PMI operator terminal that meets the following system requirements:

- ▶ Kernel from Kernel version 1.7
- ▶ min. 30 MB free Flash memory.
- ▶ Products:
 - PMlvisu with "4" as the first digit in the device name
(e. g. PMI v416)
 - PMlopen with "4" as the first digit in the device name
(e. g. PMI o438)

PMlvisu already contains the graphics software PMI-PRO, so no other visualisation software is required.

12.2.4 Operator terminals PMI 5 from Pilz

The PSS 4000 firmware can be installed on a PMI operator terminal that meets the following system requirements:

- ▶ min. 30 MB free Flash memory.
- ▶ Products:
PMIvisu with "5" or higher as the first digit in the device name (e. g. PMI 526)

12.3 Licensing

Licensing of the OPC Server takes place when a project is licensed in PAS4000. Further information on licensing is available in PAS4000's online help.

12.4 Installation

The current installation files are available in the download area of the Pilz homepage (www.pilz.de -> Downloads). "Software" can be entered as the search term, for example.

The software download is only available for registered users. Unregistered users can register free of charge. A valid E-mail address is required for registration.

Several installation kits are available for the PSS 4000 firmware, depending on the application area:

- ▶ The PSS 4000 firmware is installed on a **PC**:
Required installation kit:
PSS_4000_PC_Installer_<version number>.zip
(e. g. PSS_4000_PC_Installer_1.1.0.zip)
- ▶ The PSS 4000 firmware is installed on an operator terminal **PMI 4**:
Required installation kit:
PSS_4000_PMI_Installer_<version number>.zip
(e. g. PSS_4000_PMI_Installer_1.1.0.zip)
The installation files are in the subdirectory "PMI_4" after unzipping.
- ▶ The PSS 4000 firmware is installed on an operator terminal **PMI 5**:
Required installation kit:
PSS_4000_PMI_Installer_<version number>.zip
(e. g. PSS_4000_PMI_Installer_1.1.0.zip)
The installation files are in the subdirectory "PMI_5" after unzipping.

12.4.1 Install PSS 4000 firmware on a PC

The PSS 4000 firmware is installed on the PC that serves as the diagnostic unit. The diagnostic unit may also be used simultaneously as a display unit. When the PSS 4000 firmware is installed, the OPC Core Components are also installed automatically. The OPC Core Components consist of various auxiliary programs and software components for OPC Servers. The PSS 4000 Firmware Controller is also installed.



Information

- ▶ If the OPC Server is installed on the same PC as PAS4000, the PAS4000 firmware version must be identical to the version of the PSS 4000 firmware.
- ▶ If the OPC Server is installed on the same PC as PAS4000, we recommend that you install PAS4000 first, followed by the PSS 4000 firmware.
- ▶ If the OPC Server is installed on the same PC as PAS4000, the PSS 4000 firmware sets the system variable `PILZ_SNP_MULTICAST_TTL` to 64 during installation.
- ▶ Before an OPC Client can access the OPC Server, the Server must first be started. There are various procedures for doing this:
 - Automatic start via the Autostart menu
When installing the PSS 4000 firmware, the default setting in the installation window is retained. This means that the PSS 4000 Firmware Controller is entered in the Autostart menu of the Windows operating system. Each time the PC is booted, the OPC Server is started automatically.
 - Manual start
During installation, the option responsible for the entry in the Autostart menu is deselected in the installation window. This means that, after installation and each time the PC is booted, the OPC Server must be started manually via the PSS 4000 Firmware Controller.

12.4.1.1 First-time installation of the PSS 4000 firmware

With a first-time installation, proceed as follows:

1. Start the PC on which the PSS 4000 firmware is to be installed.
2. If you haven't already done so:

Create a directory on the PC, copy the installation kit (ZIP file) into the directory and unzip it. The following file will be available:

- PSS_4000_PC_Installer_<version number>.exe
(e. g. PSS_4000_PC_Installer_1.1.0.exe)

3. Double-click the EXE file.

Follow the on-screen instructions to complete the installation.

The following tasks are executed on the PC:

- OPC Server Software, PSS 4000 Firmware Controller and OPC Core Components are installed
- Software is registered in the registry
- Icon is added to the task bar that displays the state for the PSS 4000 Firmware Controller;

(a change in state (start/stop) can be selected by right-clicking on the icon.)

This completes the installation.



Information

When registering (registry of the PC) the installation program configures, among others, the DCOM setting for the OPC server. When the installation is a first-time installation of the OPC server on the PC, the DCOM settings for the "anonymous remote access" is only effective after the PC is restarted.

12.4.2 Install PSS 4000 firmware on a PMI operator terminal

The PSS 4000 firmware is installed on an operator terminal PMI. In this case the PMI is used as a diagnostic and display device.

Installation on a PMI 4 operator terminal

The installation of the PSS 4000 firmware will cause:

- ▶ Installation of the OPC Server in the directory `\NAND\PSS4000`
- ▶ Installation of the visualisation software PSS 4000 Diag Control in the directory `\NAND\PSS4000_Diag_Control`
- ▶ Installation of core components in the directory `\NAND\OPCCore`
- ▶ Configuration of the DCOM settings to enable OPC Clients to access the OPC Server
- ▶ Adaptation of the batch file `PMIStart.cmd`

Existing entries in any existing batch file are maintained. The entries made during the installation of the batch file will allow the following software to be started during the start of a PMI:

- **PMIopen:**
 - PSS 4000 Firmware Controller
 - OPCEnum
 - PSS 4000 Diag Control
 - PSS4000_Diag.html
- **PMIvisu:**
 - PSS 4000 Firmware Controller
 - OPCEnum
- ▶ Deactivation of the PMI internal SNTP server
- ▶ Generation of the log file `PSS4000_Installer.log`
The file includes information on the installation status (e. g. message on successful installation, error messages).
- ▶ Impact after a first-time installation:
If the first-time installation of the PSS 4000 firmware has been successful, the point for "FirmwareUpdate" is set to the default value "2009-01-01-00:00:00.000" (see namespace: `pss4000.net.<device name>.Device.Product.FirmwareUpdate`).
- ▶ Impact after a firmware update:
 - If a device naming procedure has been performed before the firmware update, the naming data from this device naming procedure are retained on the removable data medium.
 - If a project download has been performed before the firmware update, the device project is retained on the removable data medium.
- ▶ After installation the PMI 4 operator terminal will be restarted.

Installation on a PMI 5 operator terminal

The installation of the PSS 4000 firmware will cause:

- ▶ Installation of the PSS 4000 OPC Server in the directory `\FLASH\PSS4000`

- ▶ Installation of the visualisation software PSS 4000 Diag Control in the directory `\FLASH\PSS4000_Diag_Control`
- ▶ Configuration of the DCOM settings to enable OPC Clients to access the PSS 4000 OPC Server
- ▶ Adaptation of the batch file `PMIStart.cmd`
Existing entries in any existing batch file are maintained. The entries made during the installation of the batch file will allow the following software to be started during the start of a PMI:
 - OPCEnum
 - PSS 4000 Firmware
 - PSS 4000 Diag Control
(only PMI 509 and PMI 515)
- ▶ Deactivation of the PMI internal SNTP server
- ▶ Generation of the log file `PSS4000_Installer.log` in the directory `\FLASH`
The file includes information on the installation status (e. g. message on successful installation, error messages).
- ▶ Impact after a first-time installation:
If the first-time installation of the PSS 4000 firmware has been successful, the point for "FirmwareUpdate" is set to the default value "2009-01-01-00:00:00.000" (see namespace: `pss4000.net.<device name>.Device.Product.FirmwareUpdate`).
- ▶ Impact after a firmware update:
 - If a device naming procedure has been performed before the firmware update, the naming data from this device naming procedure are retained on the removable data medium.
 - If a project download has been performed before the firmware update, the device project is retained on the removable data medium.
- ▶ After installation the PMI 5 operator terminal will be restarted.

12.4.2.1 First-time installation of the PSS 4000 firmware

With a first-time installation, proceed as follows:

1. Start the PMI on which the PSS 4000 firmware is to be installed.
We recommend to operate the PMI via keyboard during the installation.
Exit all programs with a PMI that has already been started.
2. If you haven't already done so:
Copy the installation set (ZIP file) into a directory on a PC and unpack it.
In the "PMI_4" subdirectory you receive the following installation files for operator terminals PMI 4:
 - PSS_4000_PMI_4_Installer_<version number>.exe
 - PSS_4000_PMI_4_Data_<version number>.cabIn the "PMI_5" subdirectory you receive the following installation files for operator terminals PMI 5:
 - PSS_4000_PMI_5_Installer_<version number>.exe
 - PSS_4000_PMI_5_Data_<version number>.cab
3. Transfer installation files to the PMI
We recommend that you copy the installation files from PC to an USB stick and then to connect the USB stick to the PMI.
4. Start the Explorer on the PMI (*Hard Disk*) and switch to the USB stick.
5. Start installation
Please note: Start the installation by double-clicking the EXE file!
6. Run installation
Follow the on-screen instructions to complete the installation. You cannot change the installation directory `\\WAND\PSS4000`.
7. If the installation has been successful, the PMI starts again.
If errors occur during installation, please note the information in the log file `\\WAND\PS4000\PSS4000_Installer.log`.

This completes the installation.



Information

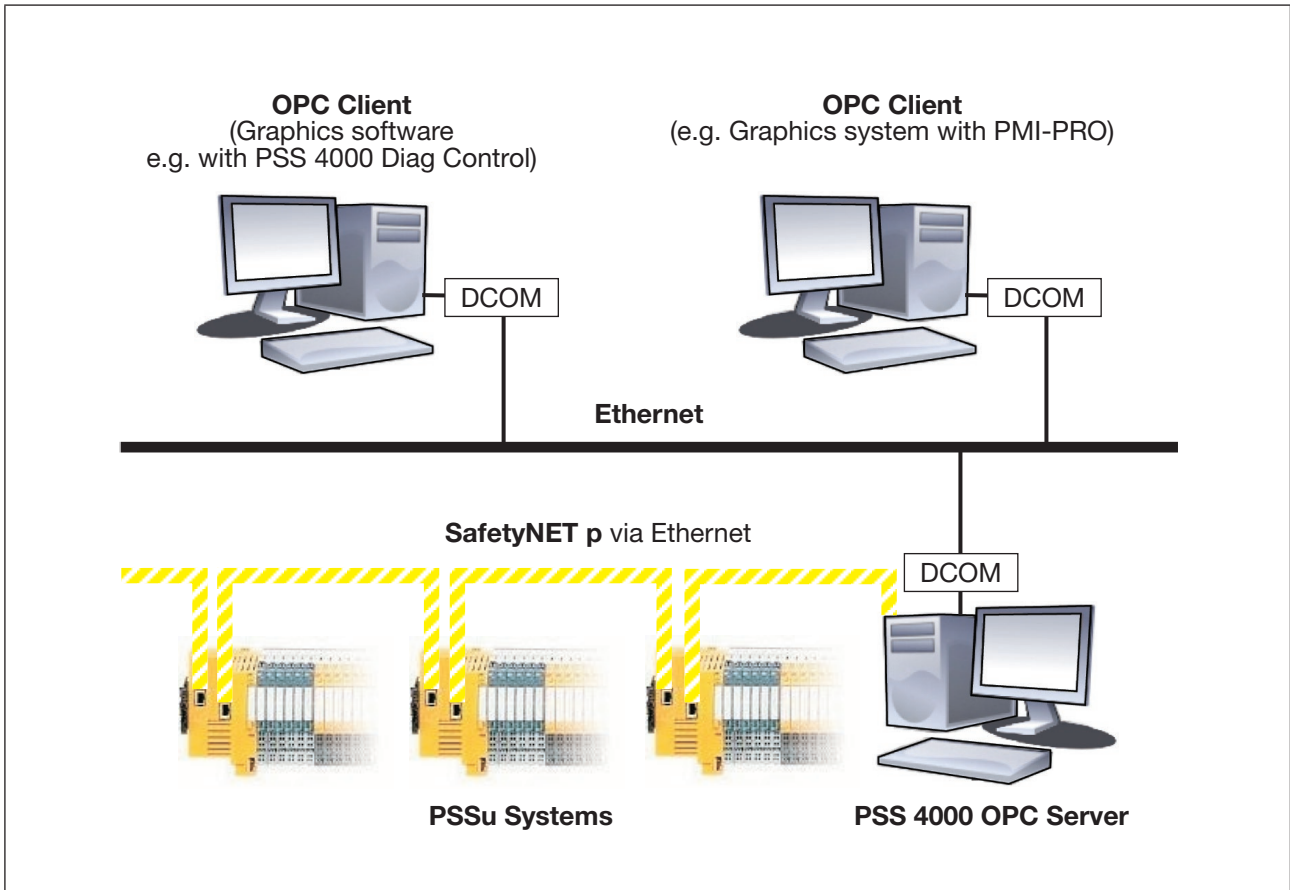
When installing, the installation program configures the DCOM setting for the OPC Server in the registry file. This configuration downgrades the safety settings.

12.5 Communication and security settings for PCs

The settings described below are only required if:

- ▶ The OPC Server is being operated on a PC and
- ▶ OPC Clients that are being operated on PCs access the OPC Server via Ethernet.

Example



OPC Clients accessing the PSS 4000 OPC Server via Ethernet

To enable OPC Clients to access the OPC Server via Ethernet, various settings must be made in the DCOM configuration for all PCs involved in data exchange. These settings are documented below.

DCOM only permits authenticated access between PCs, so we recommend that the PCs (OPC Server, OPC Clients) are registered in the same domain.

The settings in the DCOM configuration grant access rights to the OPC Server for other network users. An incorrect configuration can compromise the safety and stability of the operating system. A network administrator can optimise these configurations to protect the OPC Server from unauthorised access.

12.5.1 DCOM configuration

The description below applies to the following operating systems

- ▶ Microsoft Windows XP with at least SP2
- ▶ Microsoft Windows Embedded Standard 7
- ▶ Microsoft Windows 7

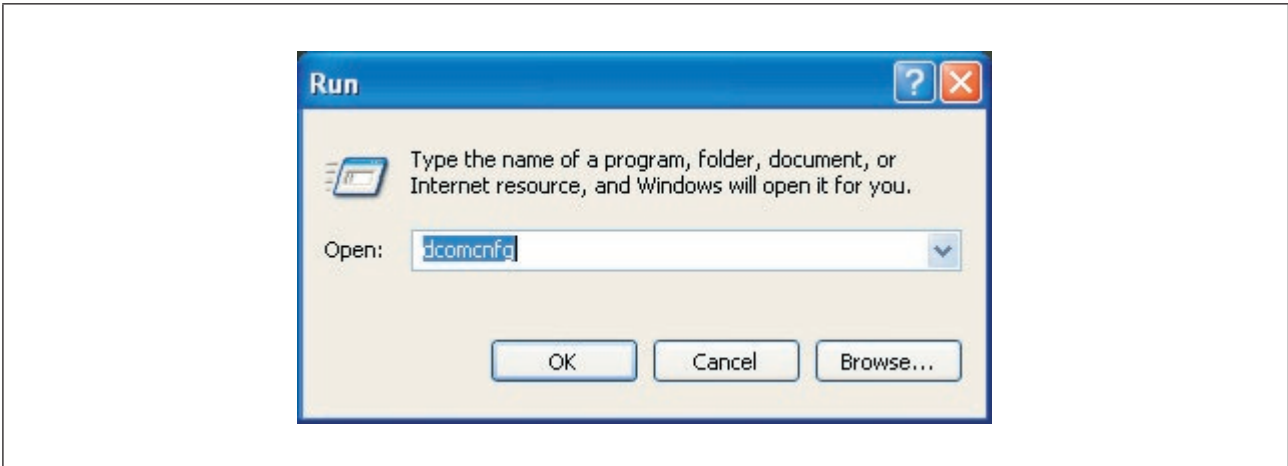
The required settings and configurations are described predominantly as part of the dialogue boxes of Microsoft Windows XP SP2. The dialogue boxes shown may vary depending on the Windows version and service pack. Further guidelines are available under Windows Help.

12.5.1.1 Start DCOM configuration

Start DCOM configuration under Microsoft Windows XP

- ▶ Select **Run...** in the Windows Start menu

The **Run** window opens.

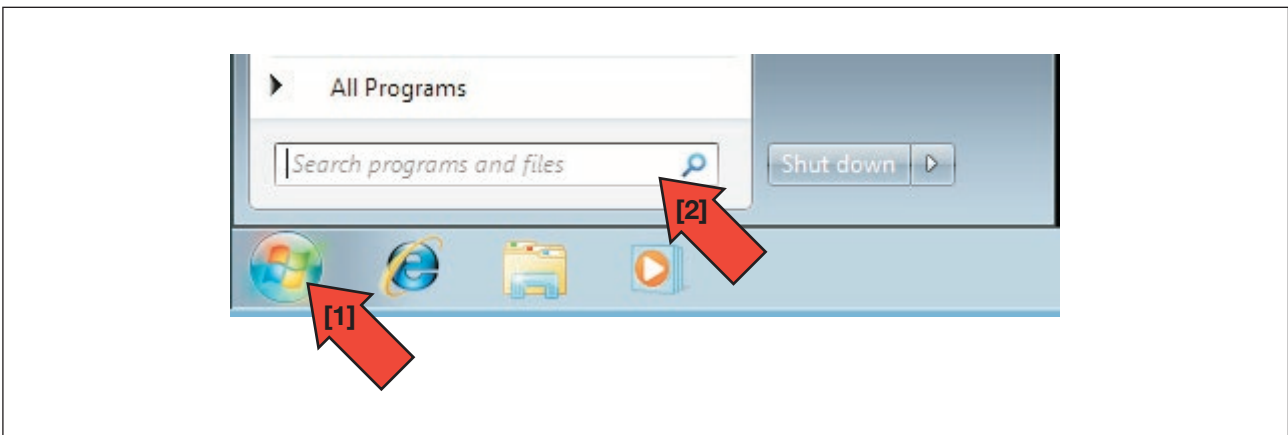


"Run" window

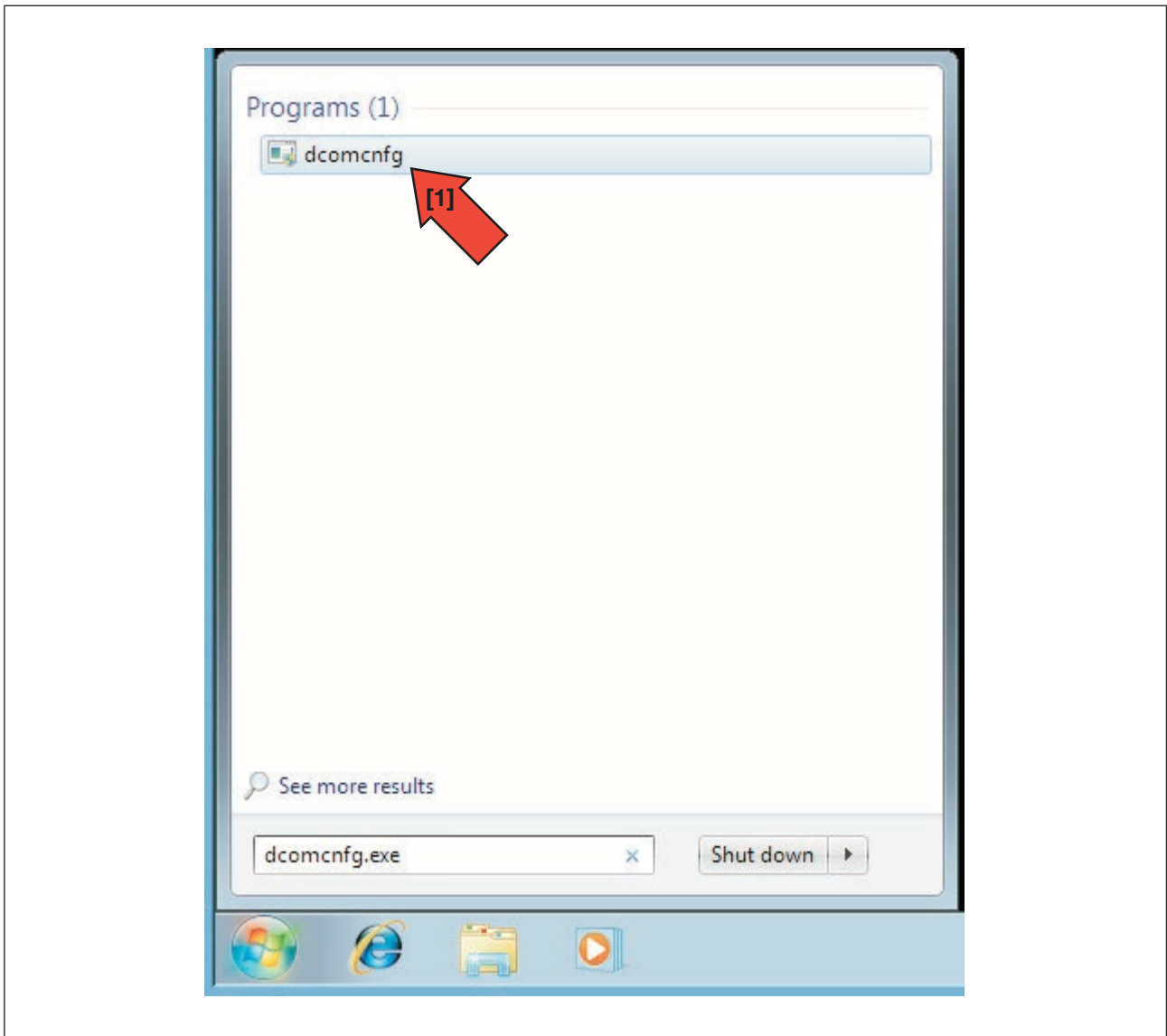
- ▶ Enter "dcomcnfg" and click **OK**.
The **Component Services** window opens.

Start DCOM configuration under Microsoft Windows 7

- ▶ Click the Windows Start symbol [1] and enter "dcomcnfg.exe" as a search criterion [2].
A window is opened, in which the program "dcomcnfg" is displayed as a search result.

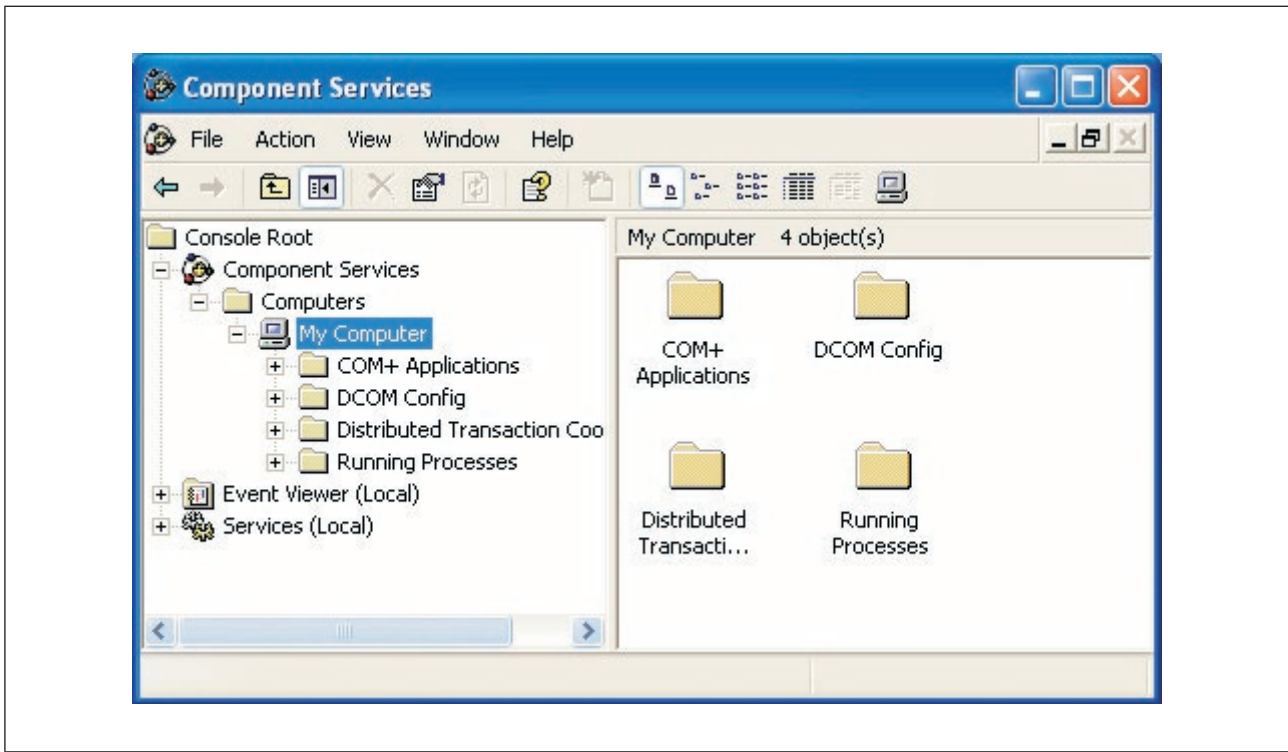


- ▶ Select the program "dcomcnfg" [1] and confirm your selection.
The **Component Services** window opens.



Component services

- ▶ Open the directory in the **Component Services** window (window's left-hand column)
Console Root / Component Services / Computers / My Computer.



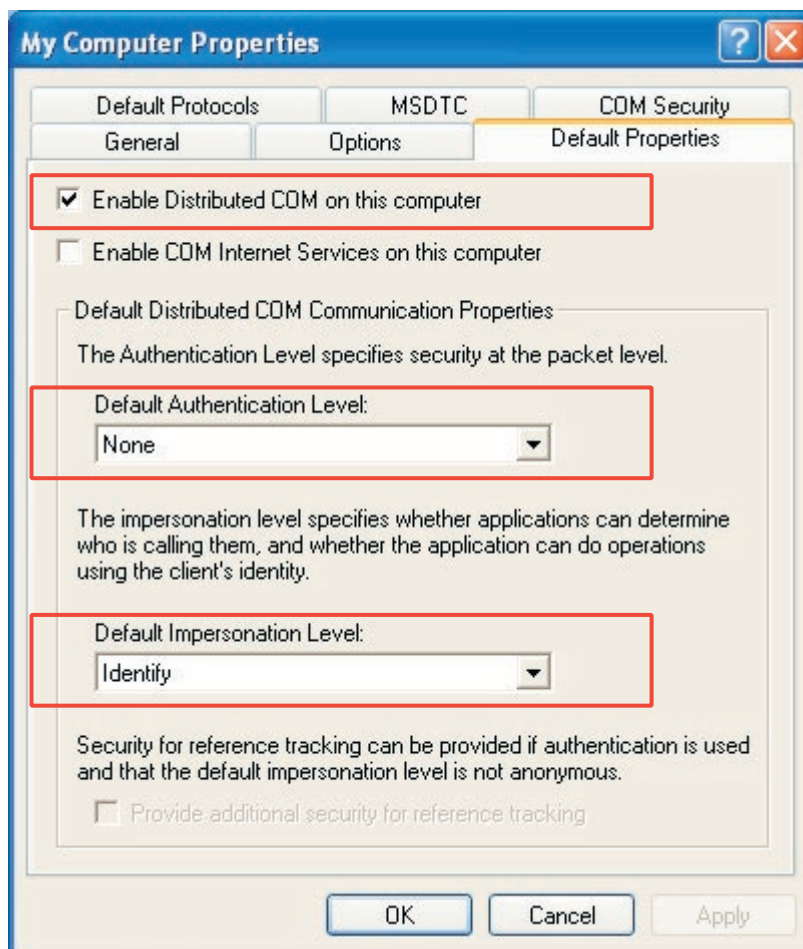
"Component Services" window

- ▶ For further details of the configuration, please refer to the sections:
 - DCOM configuration for the OPC Server and OPC Clients
 - DCOM configuration for the OPC Server
 - OPCEnum configuration for the OPC Server

12.5.1.2 DCOM configuration for the OPC Server and OPC Clients

The following communication and security settings must be made on all PCs (OPC Server, OPC Clients) involved in data exchange. These settings are called the "general DCOM configuration" and specify the access that other PCs may have to a PC.

- ▶ Open the **Component Services** window.
A description can be found under "Start DCOM configuration".
- ▶ Right-click on **My Computer** and select **Properties**.
The **My Computer Properties** window opens.
- ▶ Select the **Default Properties** tab.
 - Select **Enable Distributed COM on this Computer**.
 - Select **None** under **Default Authentication Level**.
 - Select **Accept Identity** or **Identify** under **Default Impersonation Level**.



"Default Properties" tab

- ▶ Select the **Default protocols**.tab.

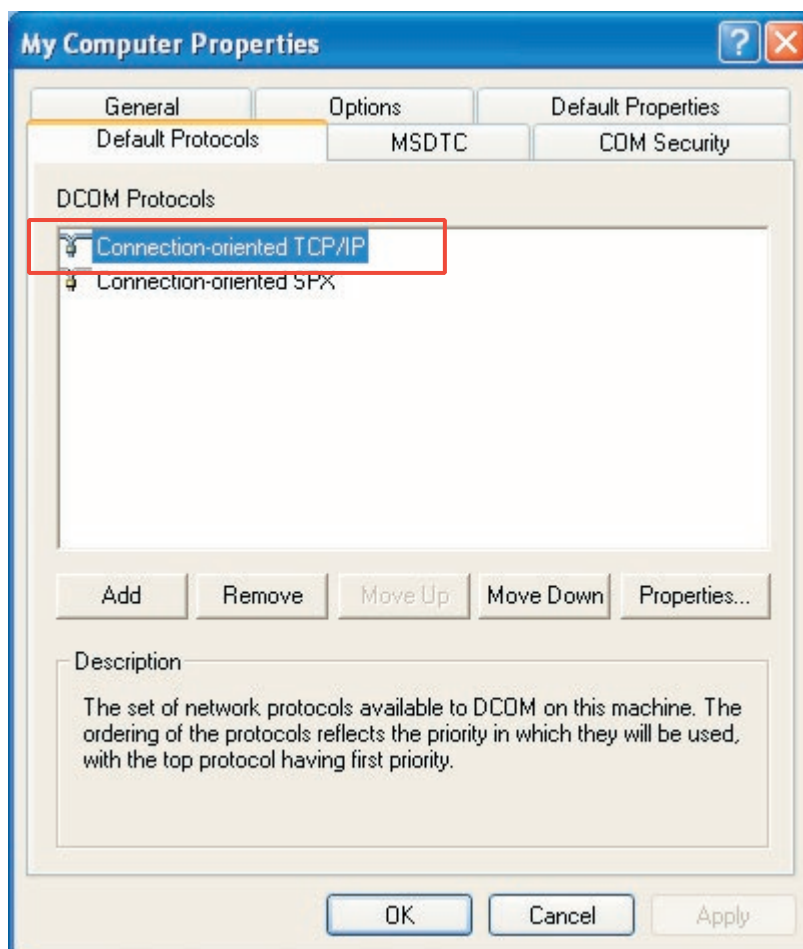
If the connection protocol **Connection-oriented TCP/IP** does not appear in the list, you can add it as follows:

- Click the **Add** button.

The **Select DCOM protocol** dialogue box opens.

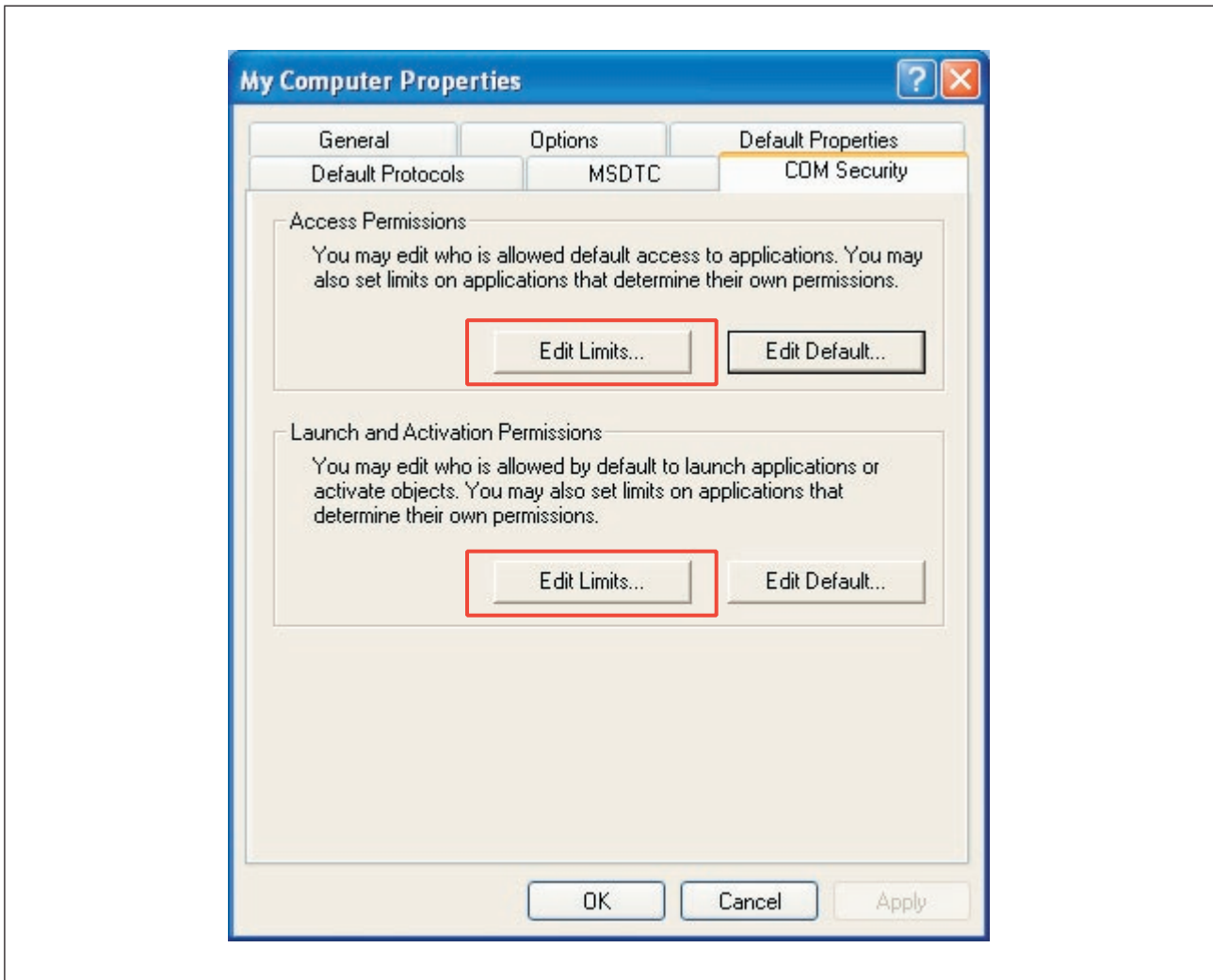
- Under **Protocol Sequence**, select the **Connection-oriented TCP/IP** protocol and click **OK**.

The **Default Protocols** tab is reactivated.



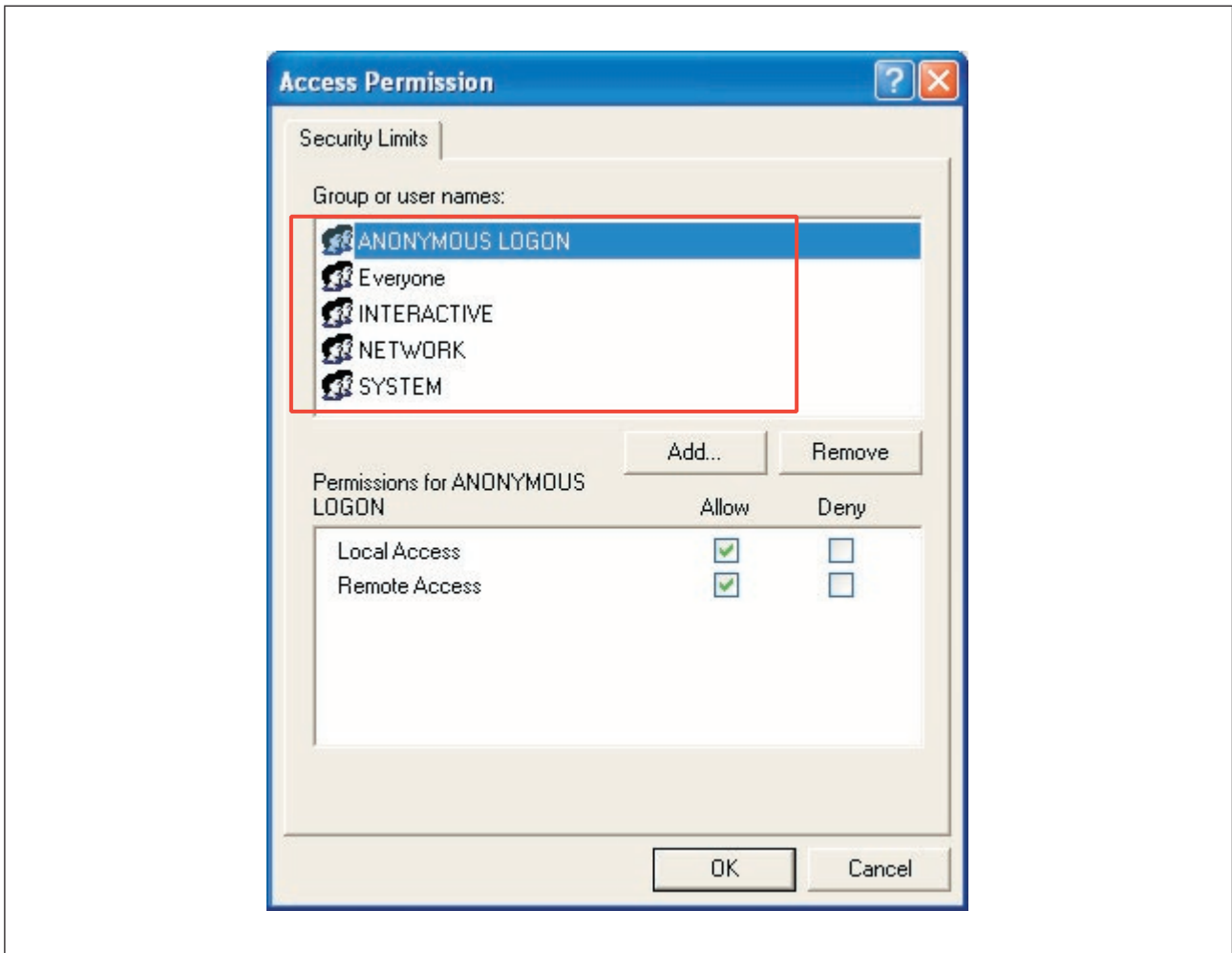
"Default Protocols" tab

- ▶ Highlight the **Connection-oriented TCP/IP** protocol and move it to the top of the list. This will speed up the connection setup.
- ▶ Remove any unneeded protocol entries, as these will cause delays when establishing the connection.
- ▶ Select the **COM Security** tab.



"COM Security" tab

- ▶ Under **Access Permissions**, click on **Edit Limits**
The **Access Permission** window opens.
- ▶ Check whether all the following group or user names are available:
 - ANONYMOUS LOGON
 - INTERACTIVE
 - Everyone
 - NETWORK
 - SYSTEM



"Access Permission" window: Group or user names (COM Security)

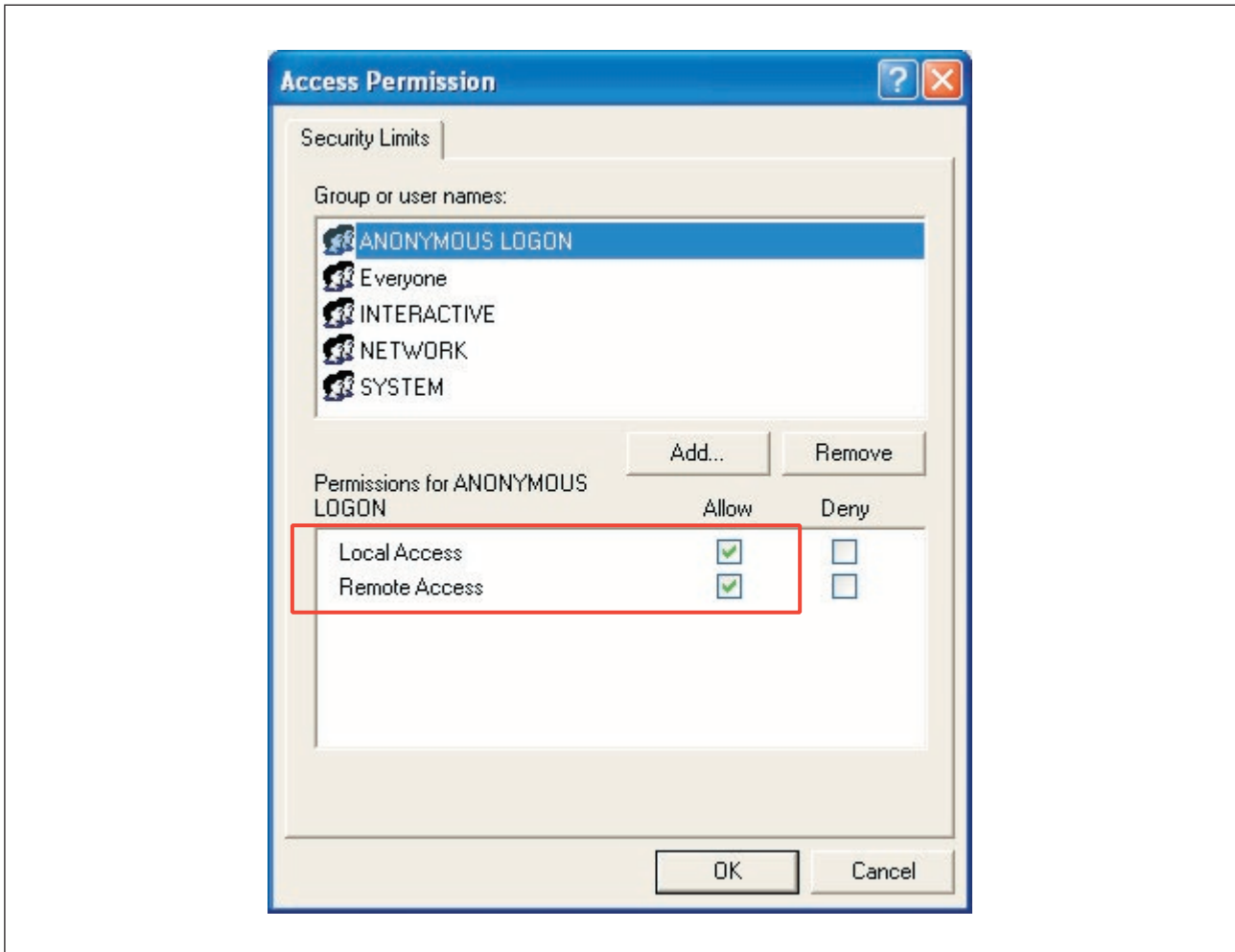
- ▶ If one of the group or user names is not available, add it (**Add... - Advanced – Search Now**).

Please note the additional information for the user NETWORK (see Special feature of the user "NETWORK", [Page 443](#)).

Then confirm with **OK**.

The **Access Permission** window is reactivated.

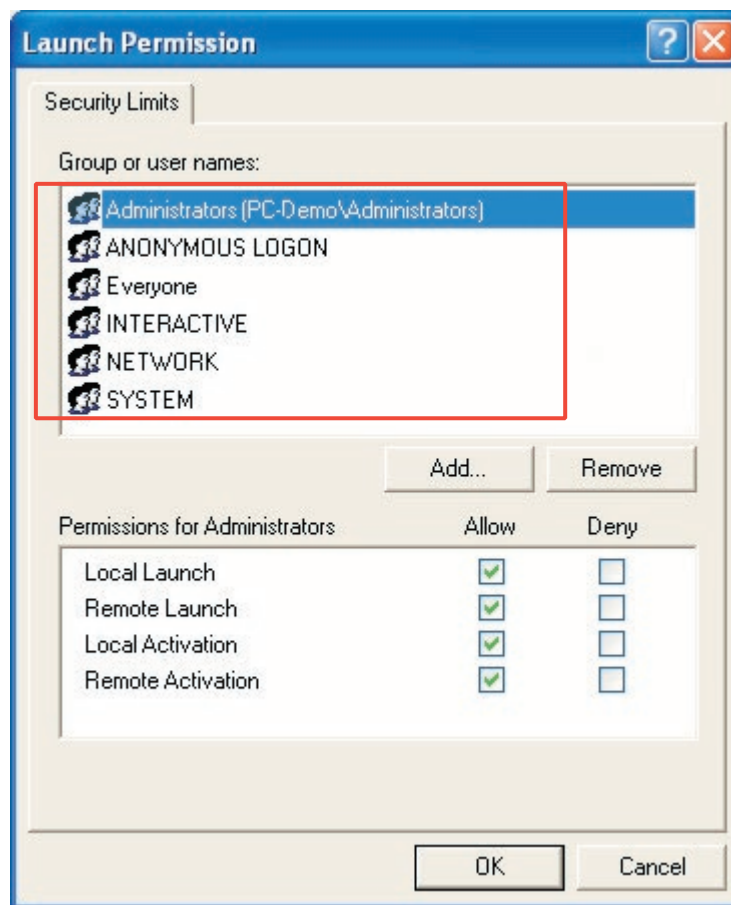
- ▶ Select each user individually and tick **Allow** for the permissions.
 - Local access
 - Remote Access



"Access Permission" window: Permissions (COM Security)

- ▶ Click **OK**.
The **COM Security** tab is reactivated.
- ▶ Under **Launch and Activation Permissions**, click on **Edit Limits**.
The **Launch Permission** window opens.

- ▶ Check whether all the following group or user names are available:
 - Administrators (...)
 - ANONYMOUS LOGON
 - INTERACTIVE
 - Everyone
 - NETWORK
 - SYSTEM



"Launch Permission" window: Group or user names (COM Security)

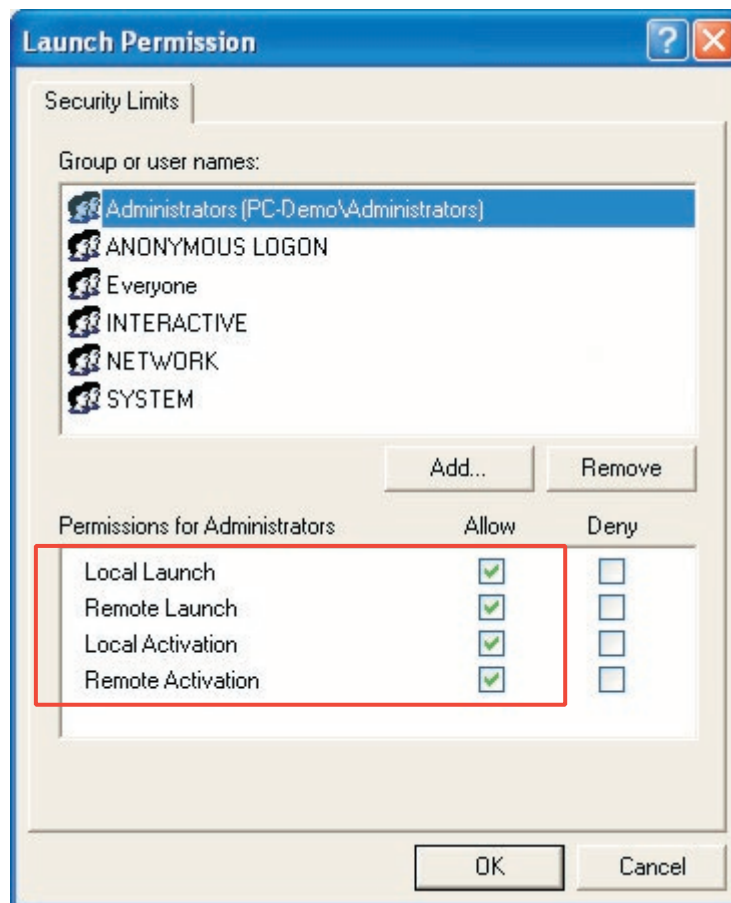
- ▶ If the "Administrators" user is not available, add it. To be able to add the "Administrators" user, the search path has to be adapted (see Special feature of the user "Administrators", [Page 442](#)).
- ▶ If one of the other group or user names is not available, add it also (**Add... - Advanced - Search Now**)

Please note the additional information for the user NETWORK (see Special feature of the user "NETWORK", [Page 443](#)).

Then confirm with **OK**.

The **Launch Permission** window is reactivated.

- ▶ Select each user individually and tick **Allow** for the permissions.
 - Local launch
 - Remote launch
 - Local activation
 - Remote activation



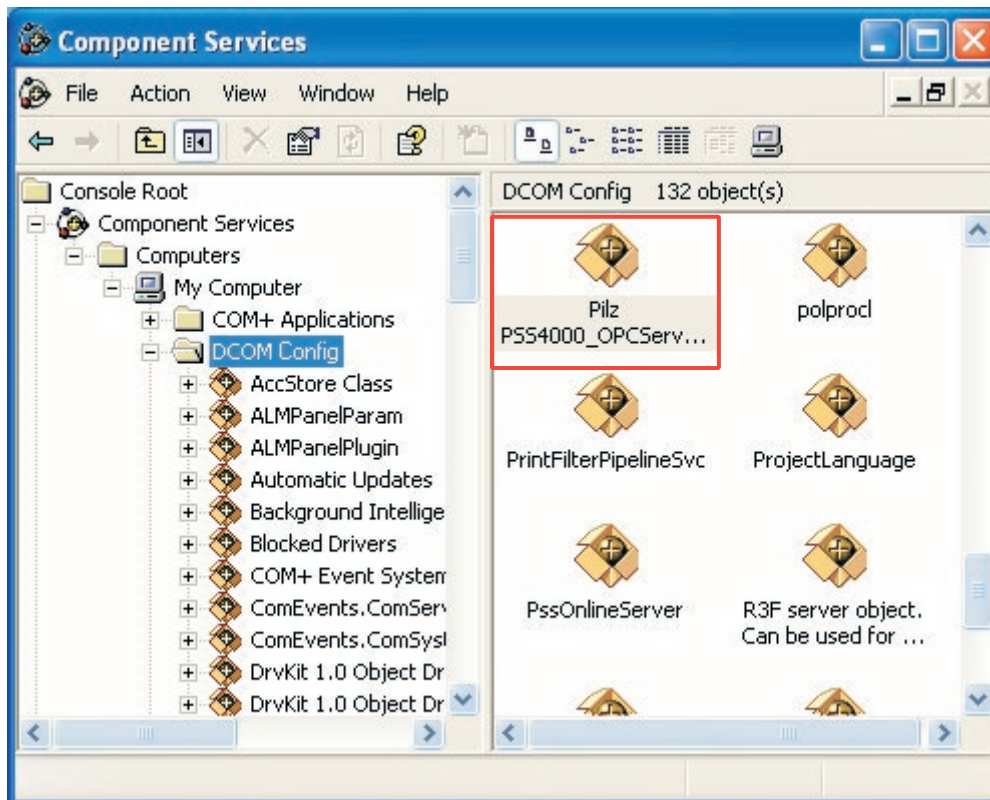
"Launch Permission" window: Permissions (COM Security)

- ▶ Click **OK**.
The **COM Security** tab is reactivated.
- ▶ Click **OK** to save all the settings and close the **My Computer Properties** window.
This completes the general DCOM configuration for this workstation.
- ▶ Close the **Component Services** window.
The settings will take effect the next time the PC is restarted.

12.5.1.3 DCOM configuration for the OPC Server

The following settings are made automatically when the PSS 4000 OPC Server is installed. Check these settings in the event of an error and modify the stated settings if necessary. These settings are only relevant for the PC containing the OPC Server.

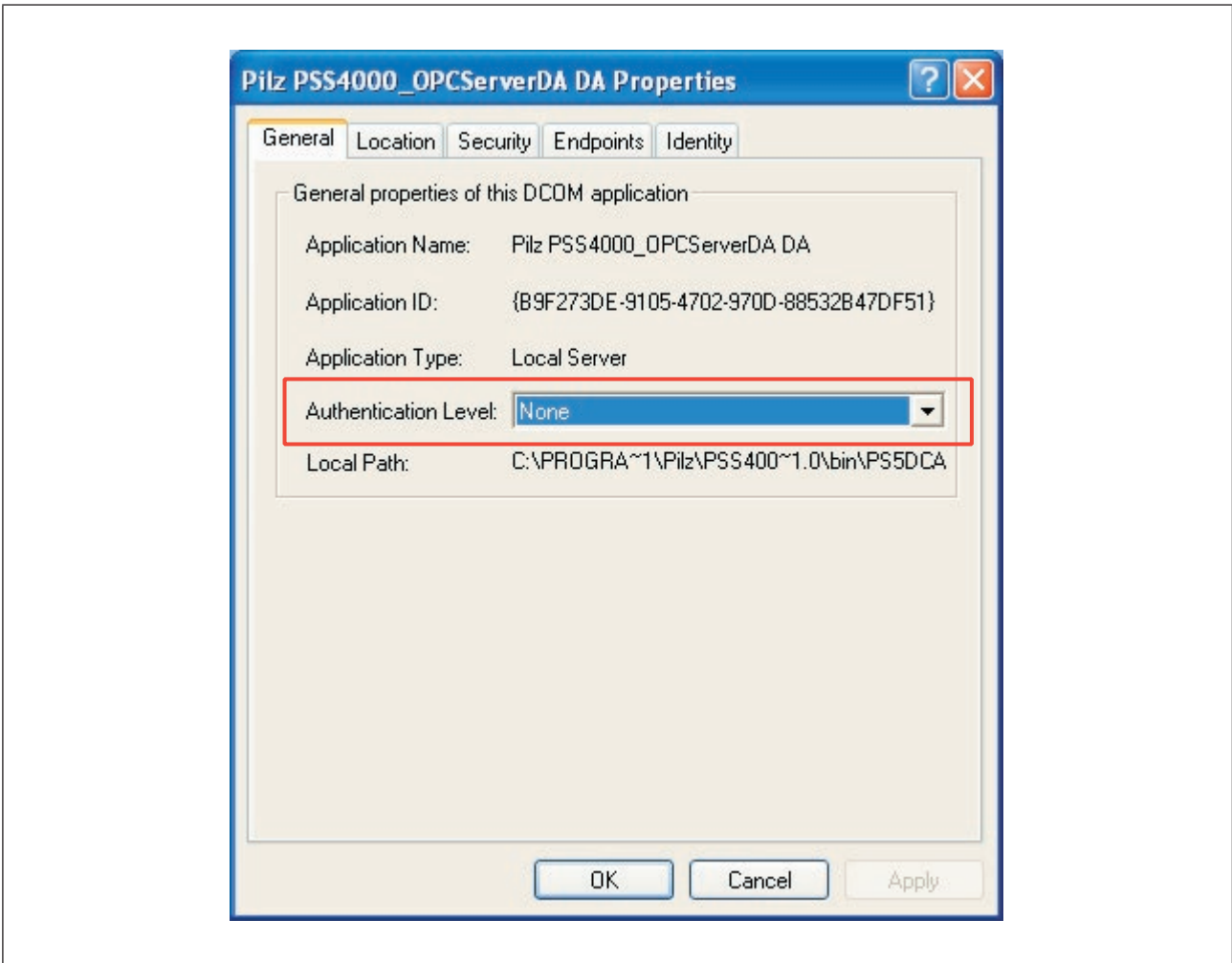
- ▶ Open the **Component Services** window.
A description can be found under "Start DCOM configuration".
- ▶ Open the **DCOM Configuration** directory in the **Component Services** window.



"Component Services" window (DCOM configuration on the OPC Server)

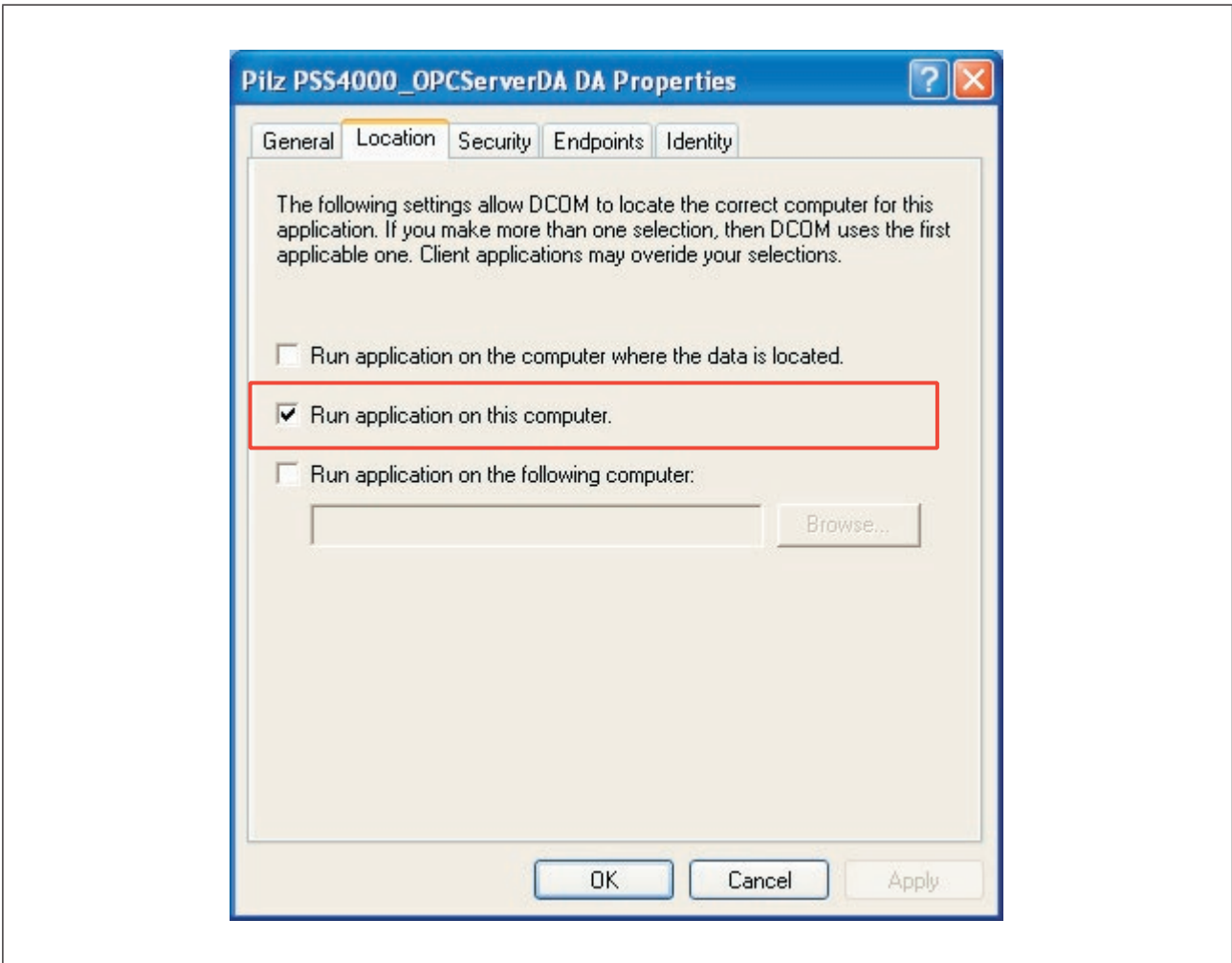
- ▶ Right-click on the application **Pilz PSS4000_OPCTServerDA.DA** and select **Properties**.
The **Pilz PSS4000_OPCTServerDA.DA Properties** window opens.

- ▶ Select the **General** tab.
Select **None** under **Authentication Level**.



"General" tab (DCOM configuration on the OPC Server)

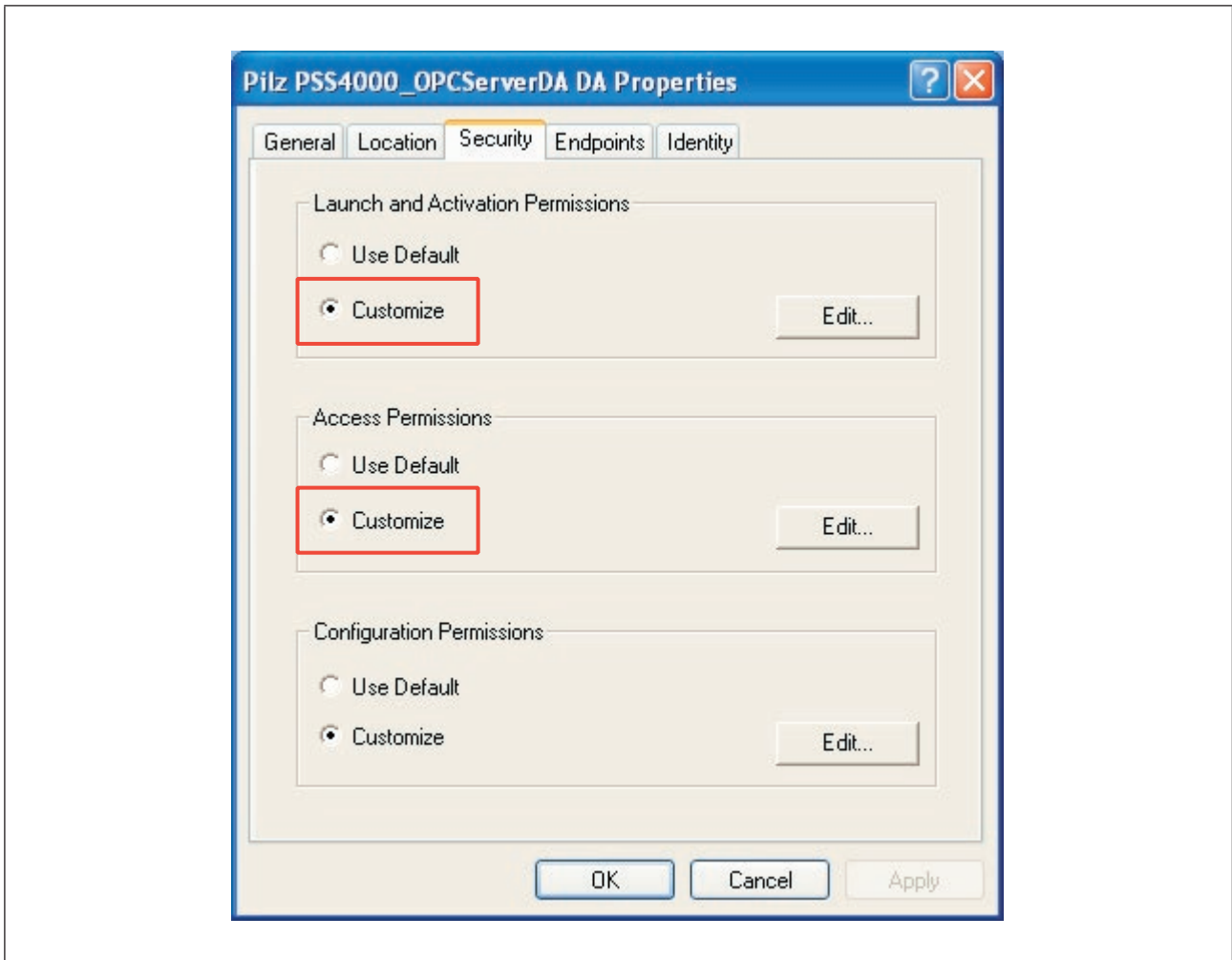
- ▶ Select the **Location** tab
- Enable the option **Run application on this Computer.**



"Location" tab (DCOM configuration on the OPC Server)

- ▶ Select the **Security** tab.

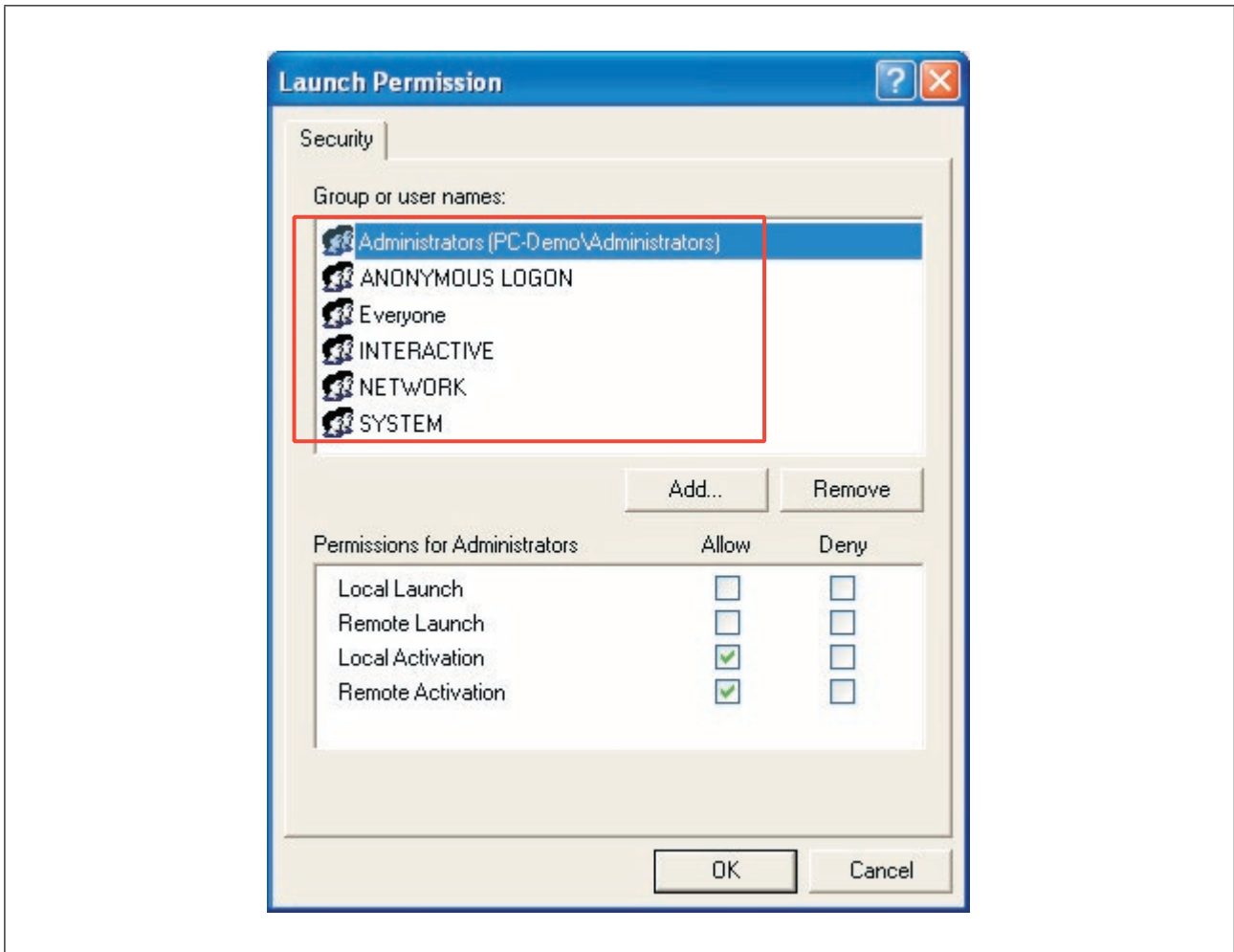
Select **Customize** for **Launch and Activation Permissions** and **Access Permissions** and then click **Apply**.



"Safety" tab (DCOM configuration on the OPC Server)

- ▶ Click on **Edit** in the **Launch and Activation Permissions** field.
The **Launch Permission** window is opened.

- ▶ Check whether all the following group or user names are available:
 - Administrators (...)
 - ANONYMOUS LOGON
 - INTERACTIVE
 - Everyone
 - NETWORK
 - SYSTEM



"Launch Permission" window: Group or user names (COM Security on the OPC Server)

- ▶ If the "Administrators" user is not available, add it. To be able to add the "Administrators" user, the search path has to be adapted (see Special feature of the user "Administrators", [Page 442](#)).
- ▶ If one of the other group or user names is not available, add it also (**Add... - Advanced - Search Now**).

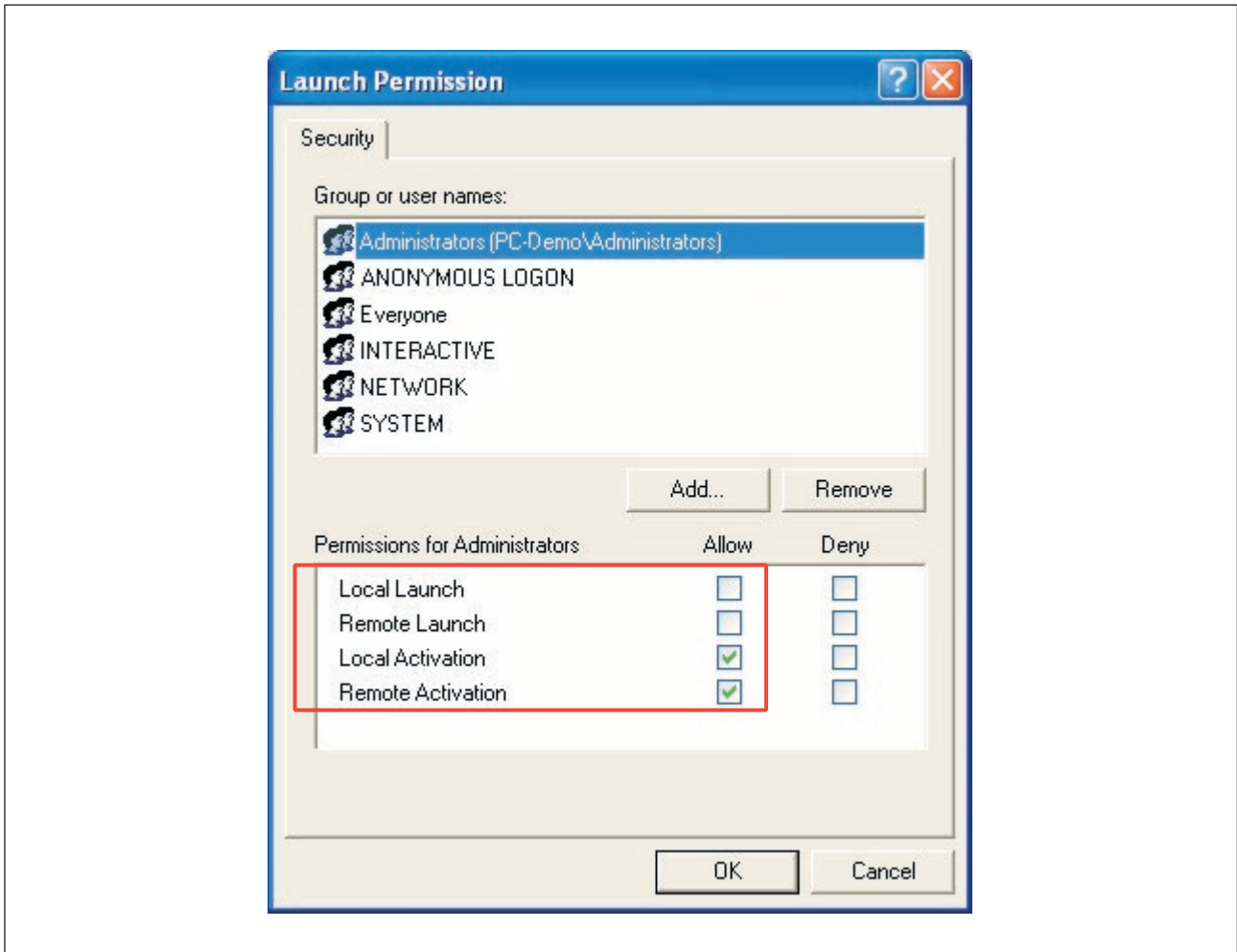
Please note the additional information for the user NETWORK (see Special feature of the user "NETWORK", [Page 443](#)).

Then confirm with **OK**.

The **Launch Permission** window is reactivated.

- ▶ Select each user individually and tick **Allow** for the permissions.
 - Local activation
 - Remote activation

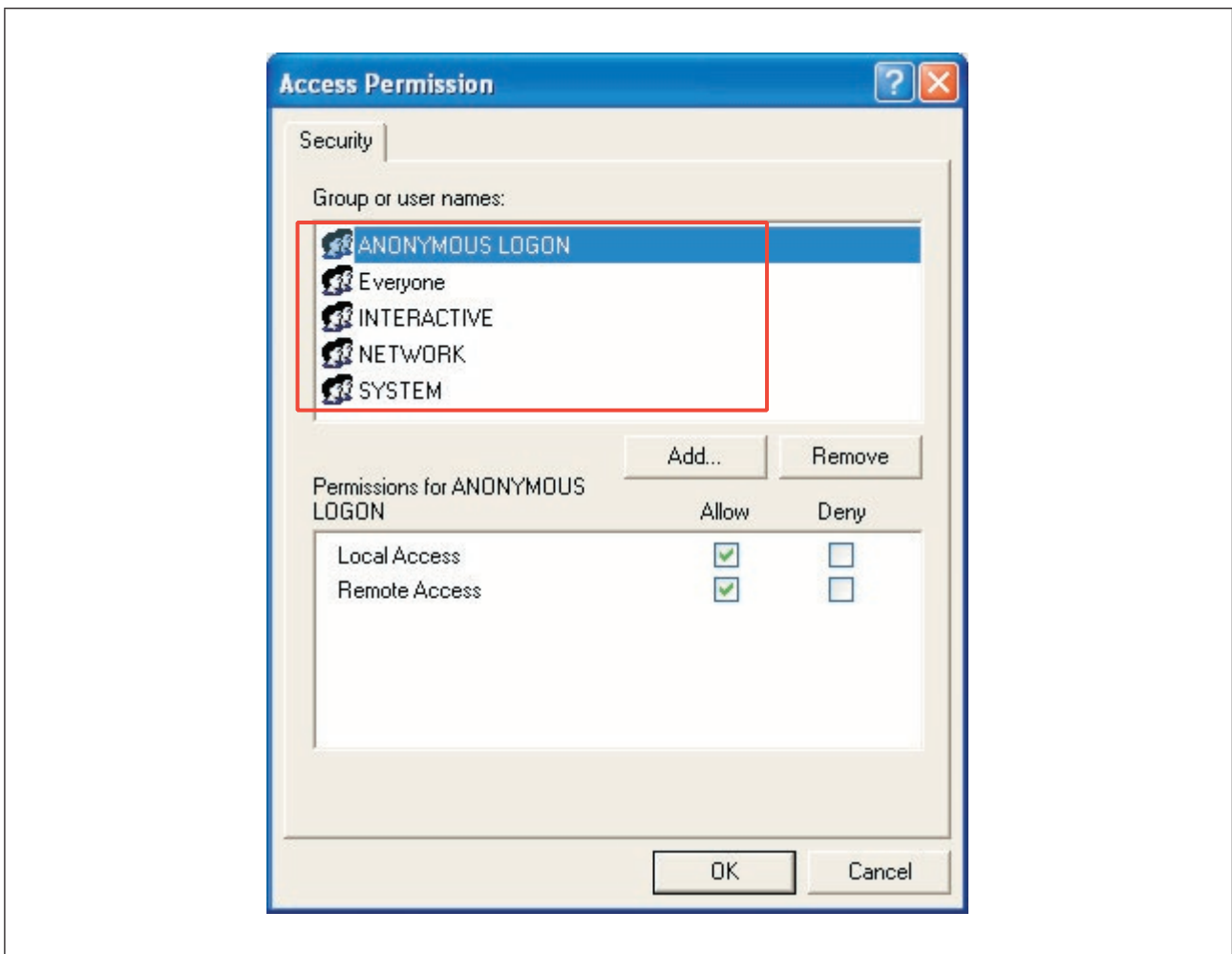
Please note that the PSS 4000 OPC server must **not** be started by clients! This is why the launch permission must not be allowed for **Local Launch** and **Remote Launch**.



"Launch Permission" window: Permissions (COM Security on the OPC Server)

- ▶ Click **OK**.
The **Security** tab is reactivated.
- ▶ Click on **Edit** in the **Access Permissions** field.

- ▶ Check whether all the following group or user names are available:
 - ANONYMOUS LOGON
 - INTERACTIVE
 - Everyone
 - NETWORK
 - SYSTEM



"Access Permission" window: Group or user names (COM Security on the OPC Server)

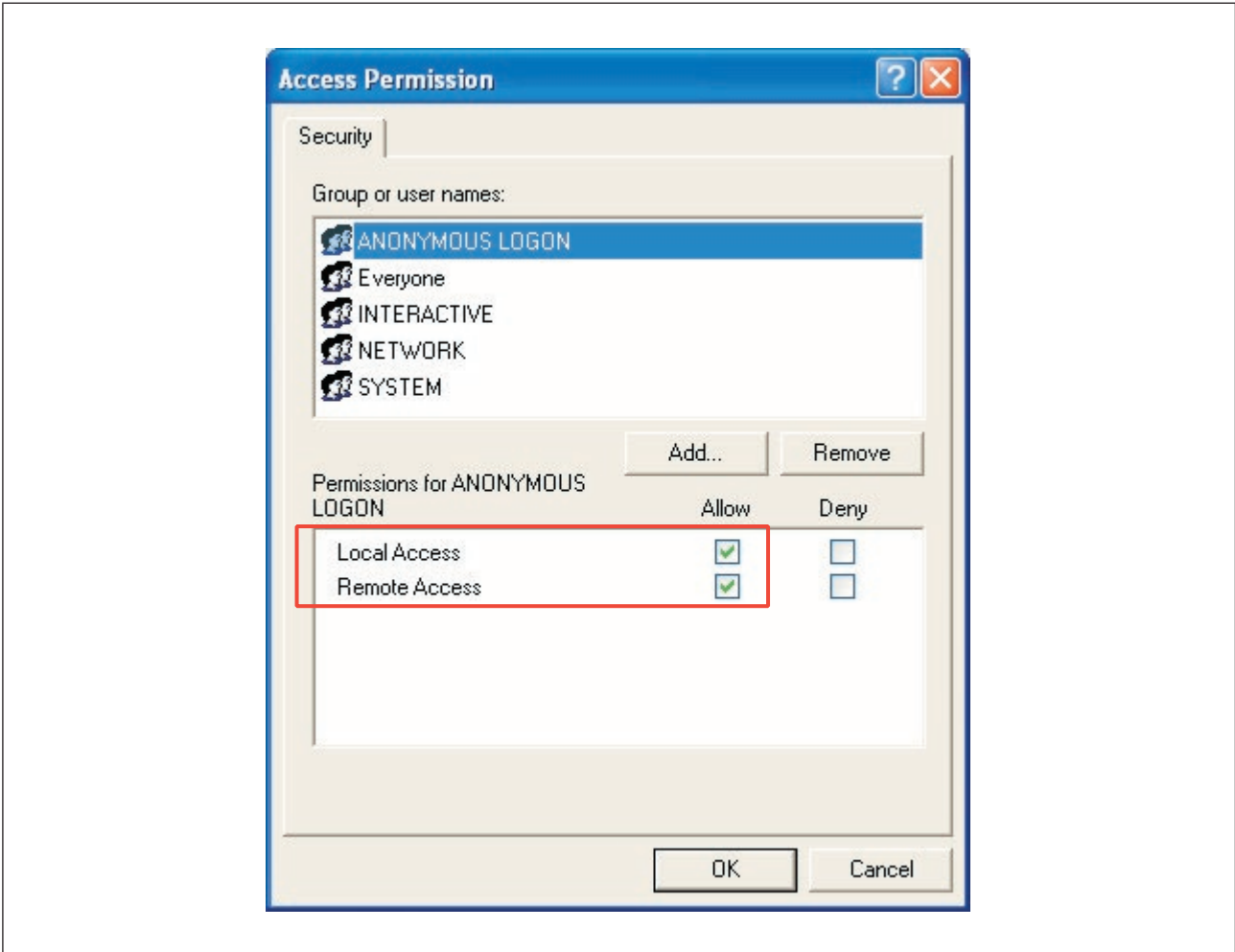
- ▶ If one of the group or user names is not available, add it (**Add... - Advanced – Search Now**).

Please note the additional information for the user NETWORK (see Special feature of the user "NETWORK", [Page 443](#)).

Then confirm with **OK**.

The **Access Permission** window is reactivated.

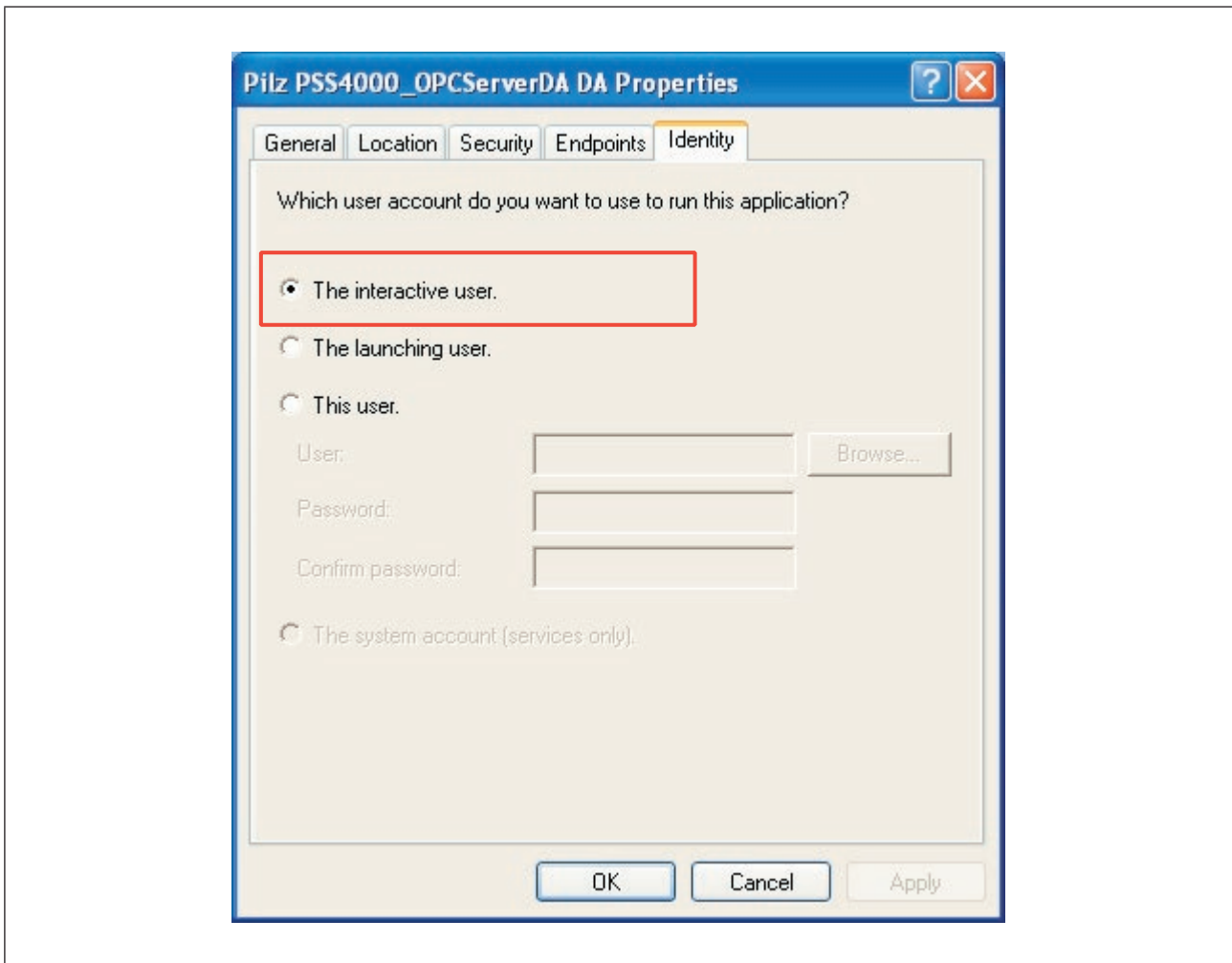
- ▶ Select each user individually and activate the option **Allow** for the permissions.
 - Local Access
 - Remote Access



"Access Permission" window: Permissions (COM Security on the OPC Server)

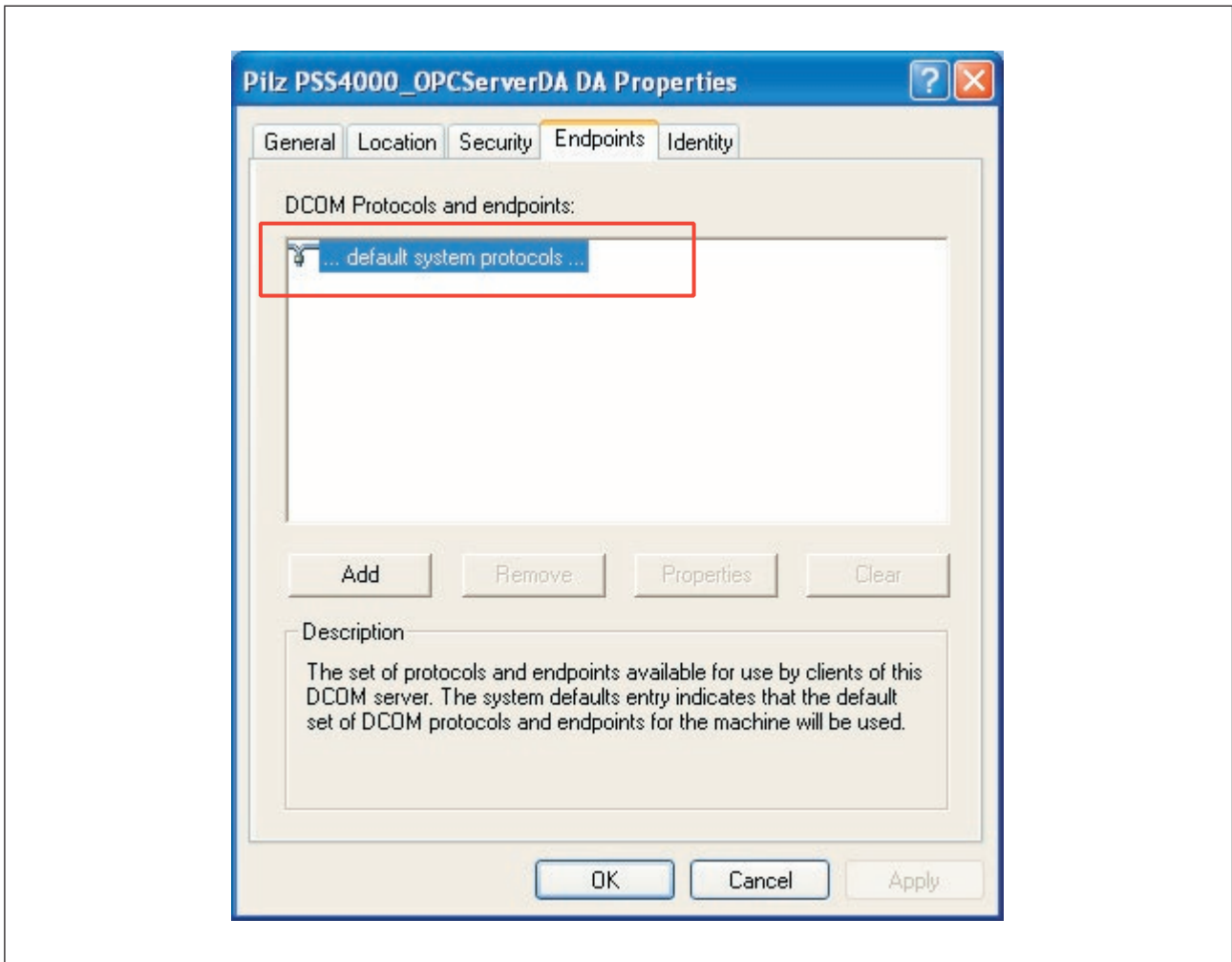
- ▶ Click **OK**.
The **Security** tab is reactivated.

- ▶ Select the **Identity** tab.
Select the option **Interactive User**.



"Identity" window (COM Security on the OPC Server)

- ▶ Select the **Endpoints** tab.
If **Default system protocols** is not available in the list, add the connection protocol as follows:
 - Click on **Add** and select the **Connection-oriented TCP/IP** protocol under **Protocol Sequence**.
 - Click **OK**.
The **Endpoints** tab is reactivated.



"Endpoints" tab (connection protocol on the OPC Server)

- ▶ Click **OK** to save all the settings and close the *Pilz PSS4000 OPCServerDA.DA Properties* window.

This completes the DCOM configuration for the OPC Server.

12.5.1.4 OPCEnum configuration for the OPC Server

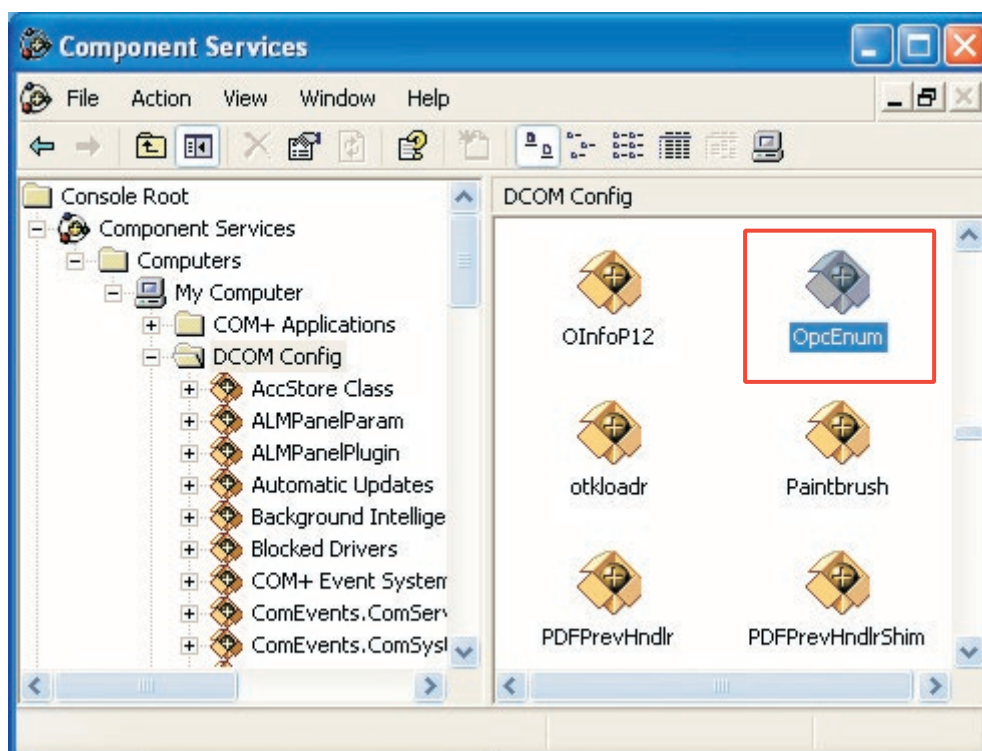
The following settings are only relevant for the PC containing the OPC Server.

- ▶ Open the **Component Services** window.

A description can be found under "Start DCOM configuration".

- ▶ Open the **DCOM Configuration** directory in the **Component Services** window.

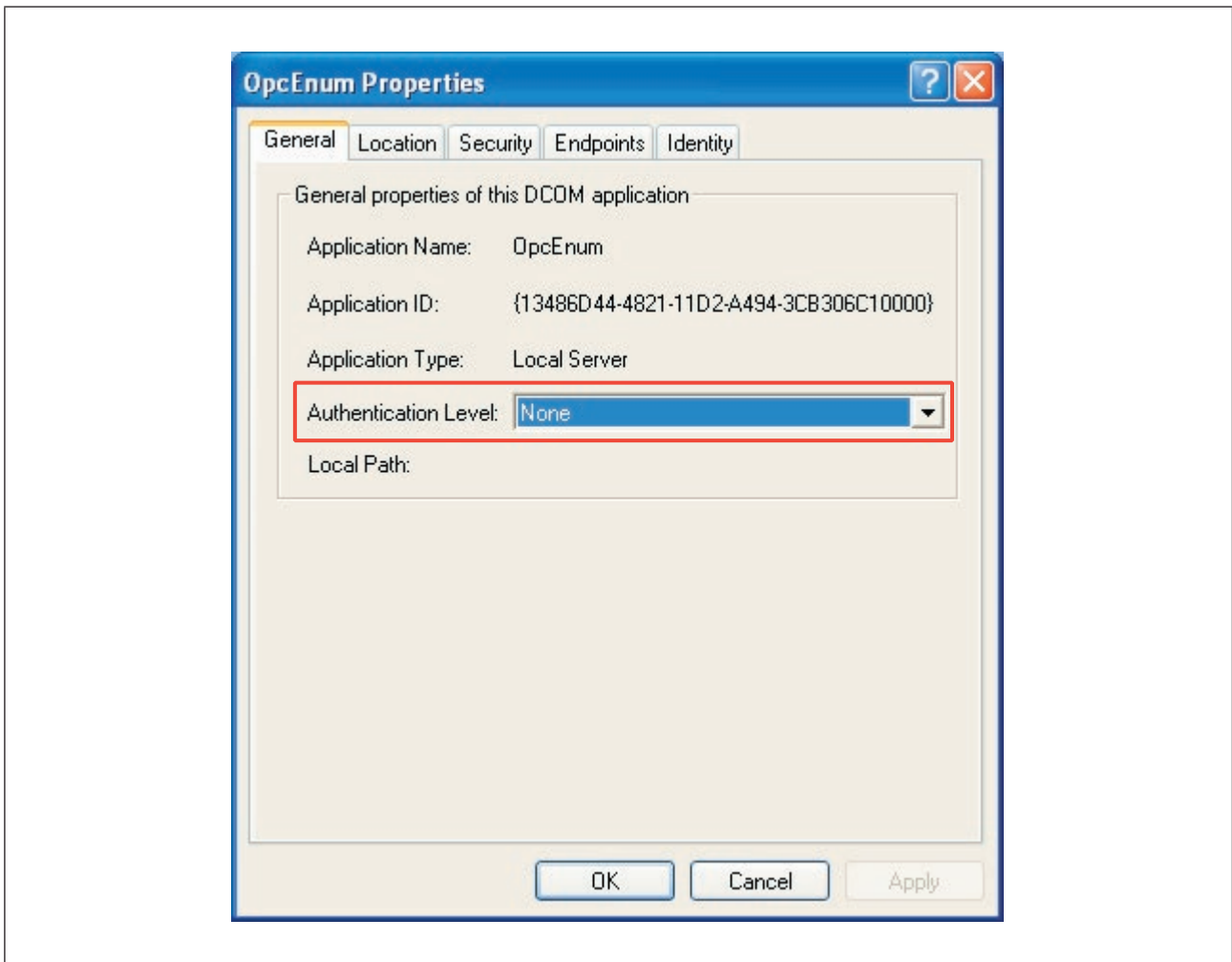
You may receive several DCOM configuration warnings, but you can ignore all of them for this configuration. Each time click **No**.



"Component Services" window (DCOM configuration for OPCEnum on the OPC Server)

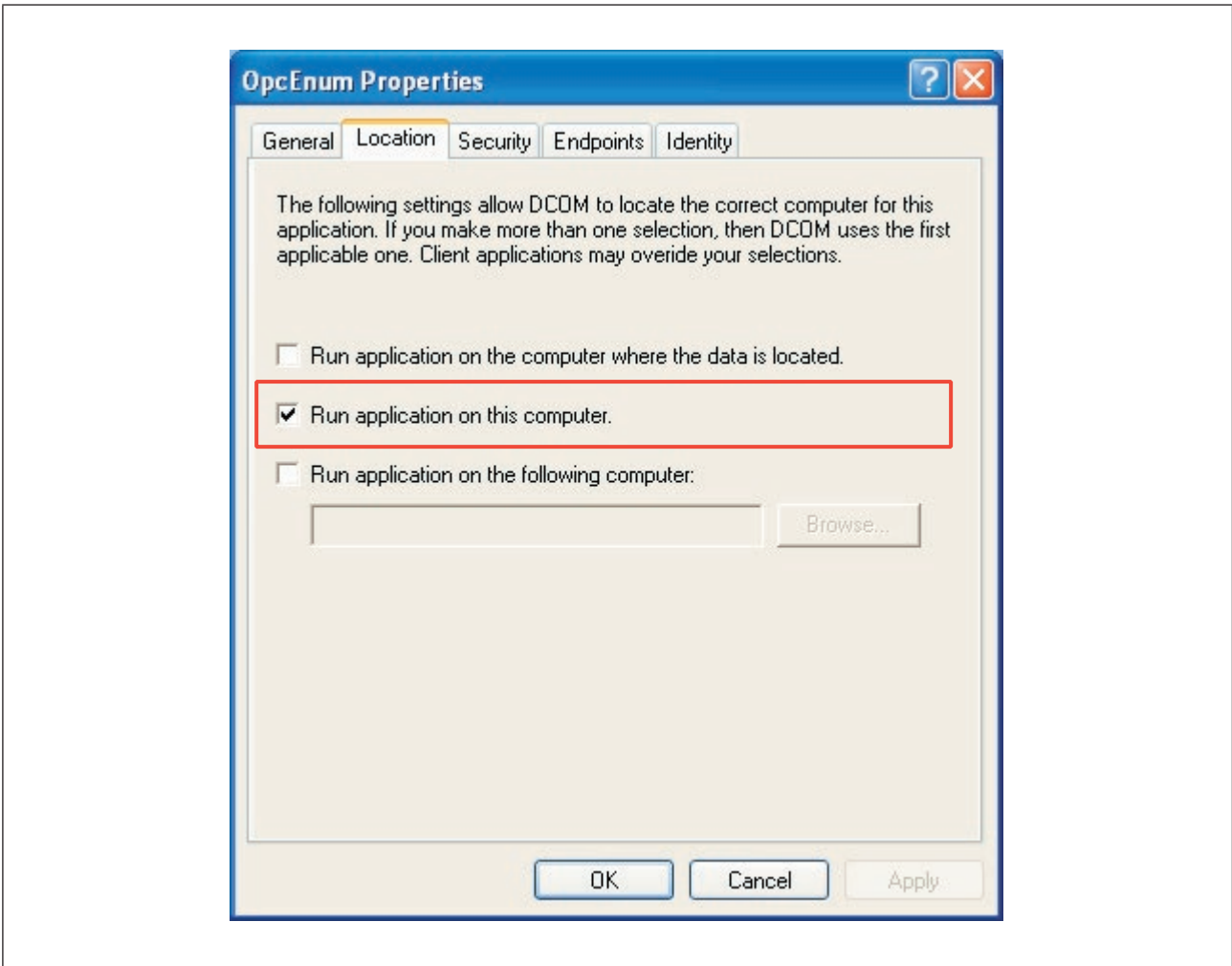
- ▶ Right-click on the **OpcEnum** application and select **Properties**.
The **OpcEnum Properties** window opens.

- ▶ Select the **General** tab.
Select **None** under **Authentication Level**.



"General" tab (DCOM configuration for OPCenum on the OPC Server)

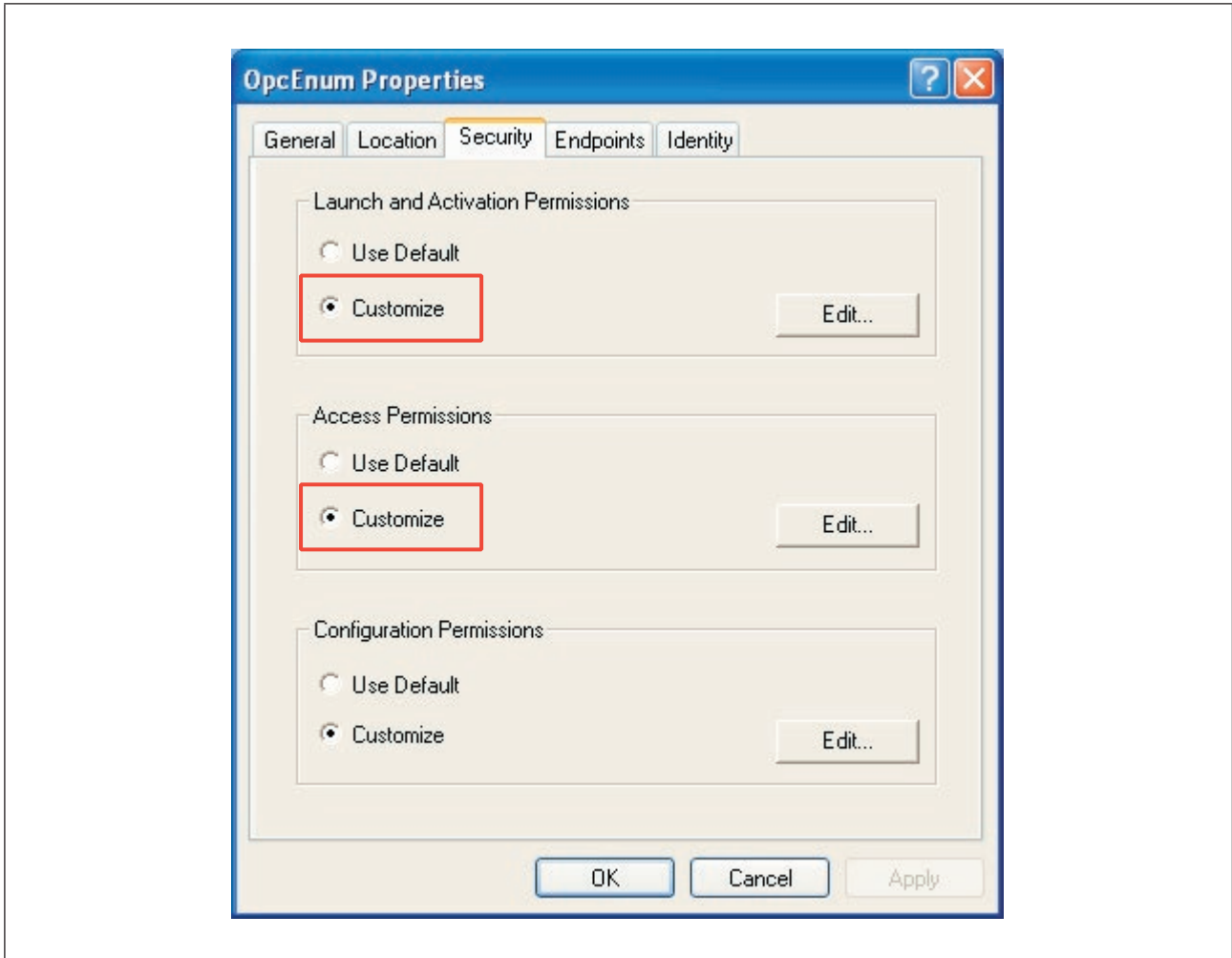
- ▶ Select the **Location** tab
 Enable the option **Run application on this Computer.**



"Location" tab (DCOM configuration for OPCEnum on the OPC Server)

- ▶ Select the **Security** tab.

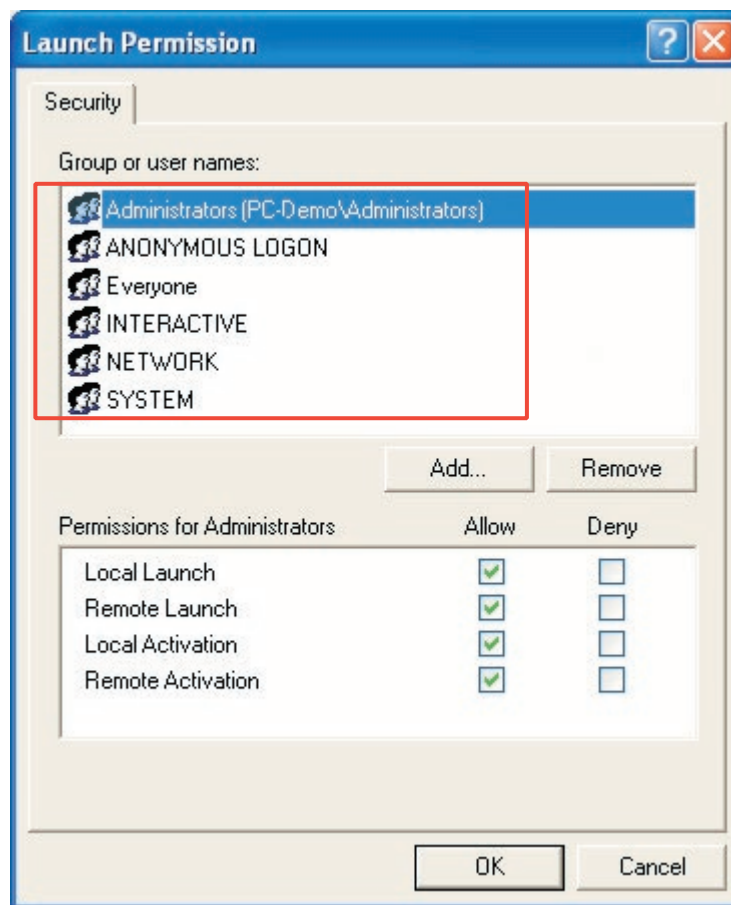
Select **Customize** for **Launch and Activation Permissions** and **Access Permissions** and then click **Apply**.



"Security" tab (DCOM configuration for OPCEnum on the OPC Server)

- ▶ Click on **Edit** in the **Launch and Activation Permissions** field.
The **Launch Permission** window is opened.

- ▶ Check whether all the following group or user names are available:
 - Administrators (...)
 - ANONYMOUS LOGON
 - INTERACTIVE
 - Everyone
 - NETWORK
 - SYSTEM



"Launch Permission" window: Group or user names (COM Security on the OPCEnum)

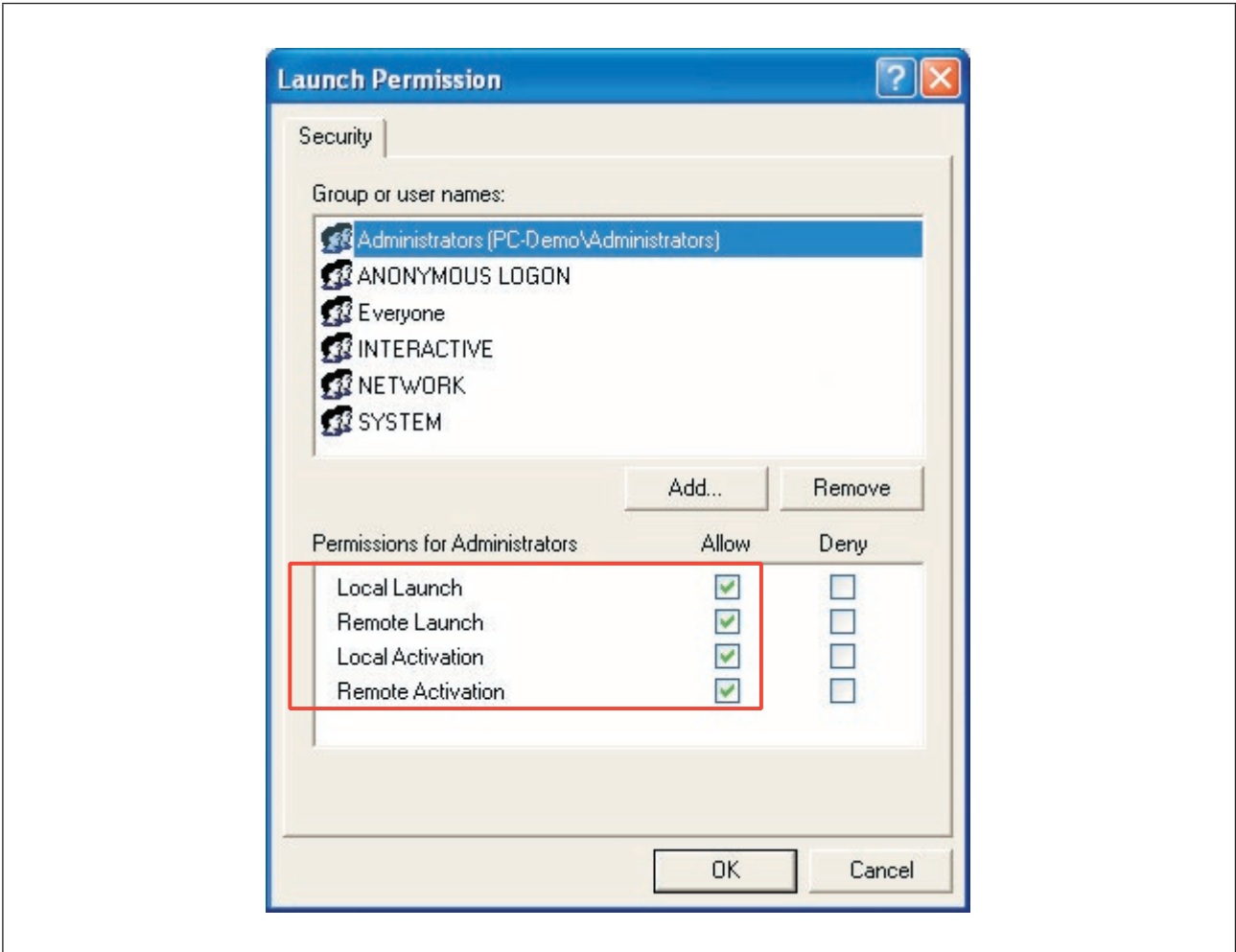
- ▶ If the "Administrators" user is not available, add it. To be able to add the "Administrators" user, the search path has to be adapted (see Special feature of the user "Administrators", [Page 442](#)).
- ▶ If one of the other group or user names is not available, add it also (**Add... - Advanced - Search Now**).

Please note the additional information for the user NETWORK (see Special feature of the user "NETWORK", [Page 443](#)).

Then confirm with **OK**.

The **Launch Permission** window is reactivated.

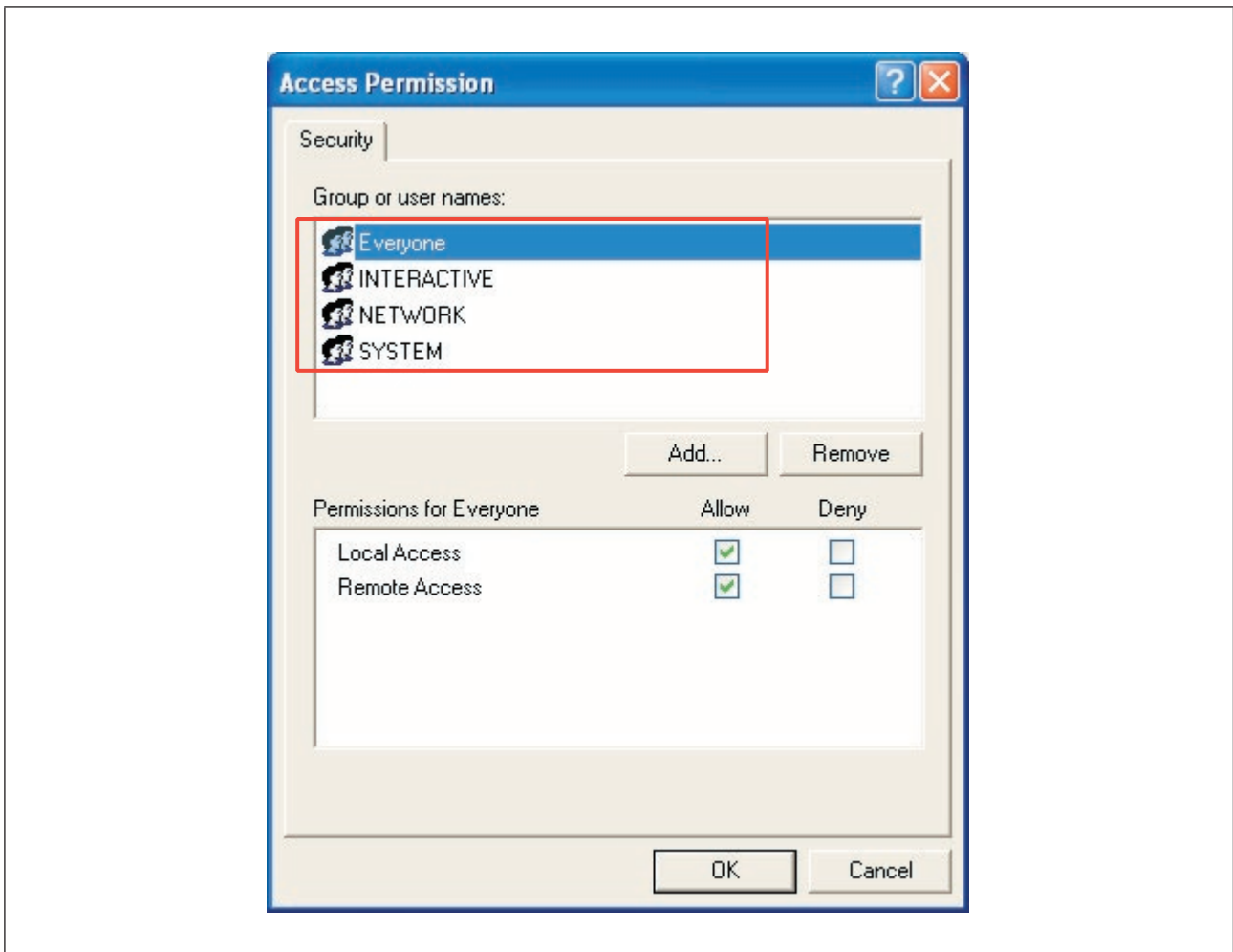
- ▶ Select each user individually and tick **Allow** for the permissions.
 - Local Launch
 - Remote Launch
 - Local Activation
 - Remote Activation



"Launch Permission" window: Permissions (COM Security for OPCEnum)

- ▶ Click **OK**.
The **Security** tab is reactivated.

- ▶ Click on **Edit** in the **Access Permissions** field.
- ▶ Check whether all the following group or user names are available:
 - INTERACTIVE
 - Everyone
 - NETWORK
 - SYSTEM



"Access Permission" window: Group or user names (COM Security on the OPCEnum)

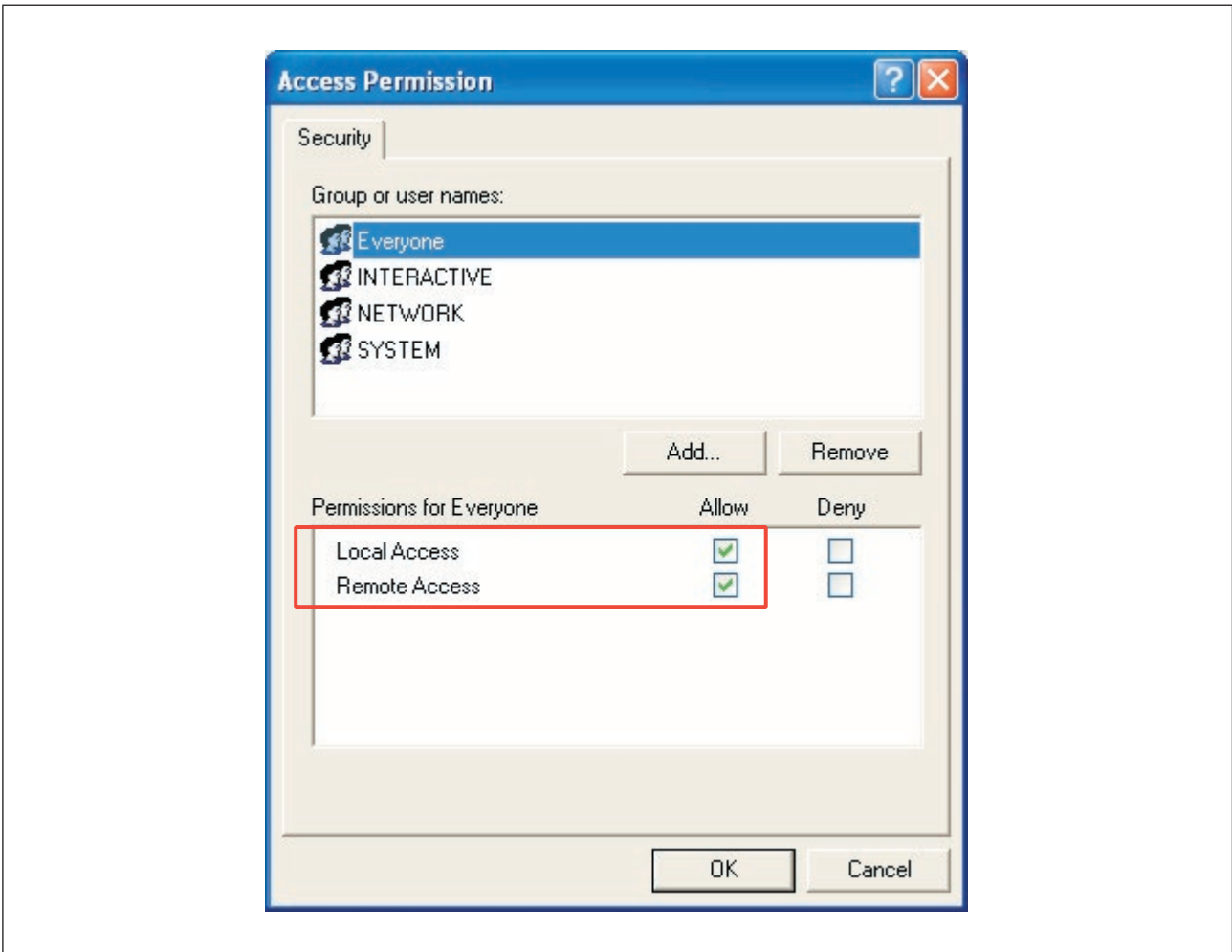
- ▶ If one of the group or user names is not available, add it (**Add... - Advanced – Search Now**).

Please note the additional information for the user NETWORK (see Special feature of the user "NETWORK", [Page 443](#)).

Then confirm with **OK**.

The **Access Permission** window is reactivated.

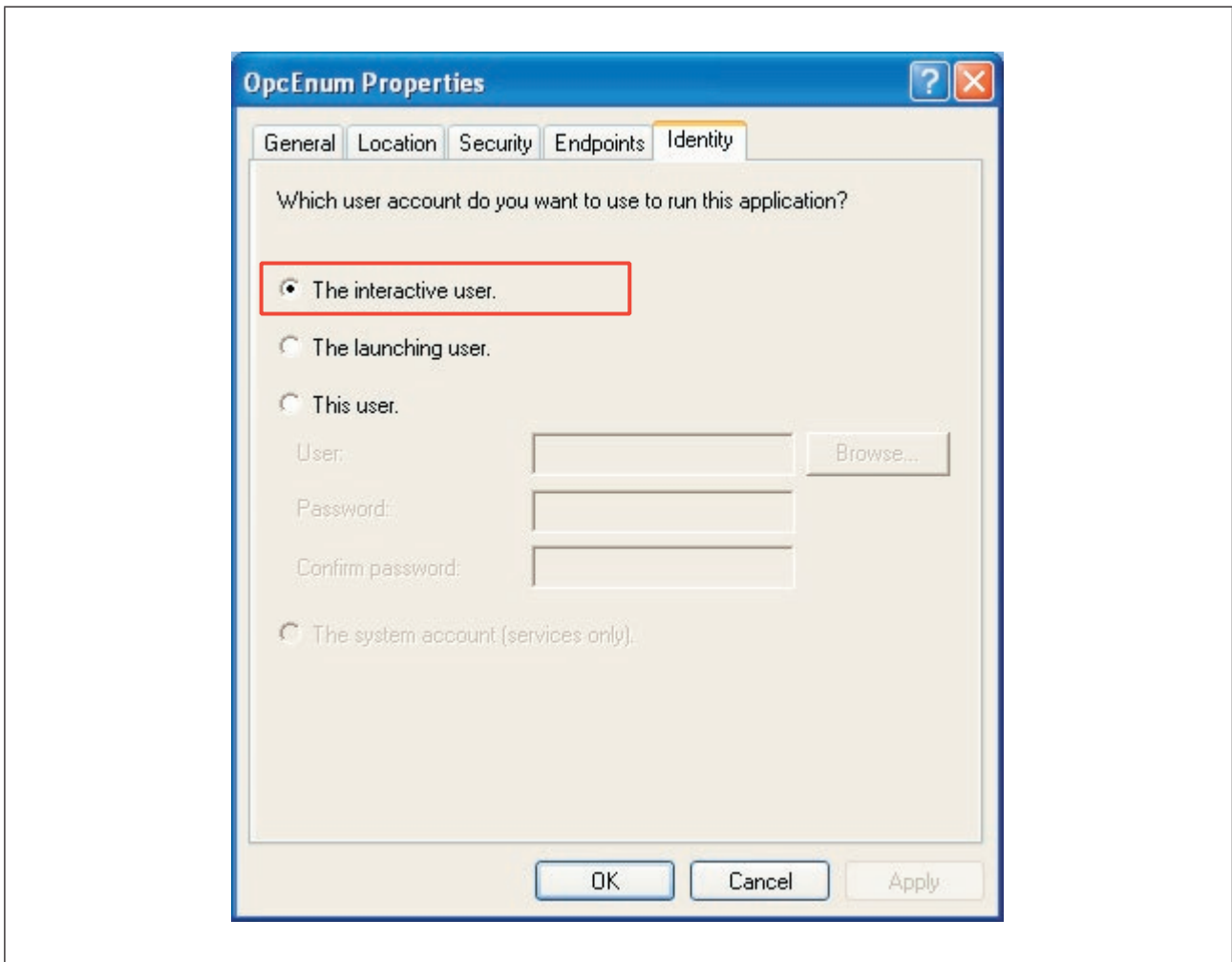
- ▶ Select each user individually and activate the option **Allow** for the permissions.
 - Local Access
 - Remote Access



"Access Permission" window: Permissions (COM Security for OPCEnum)

- ▶ Click **OK**.
The **Security** tab is reactivated.

- ▶ Select the **Identity** tab.
Select the option **Interactive User**.

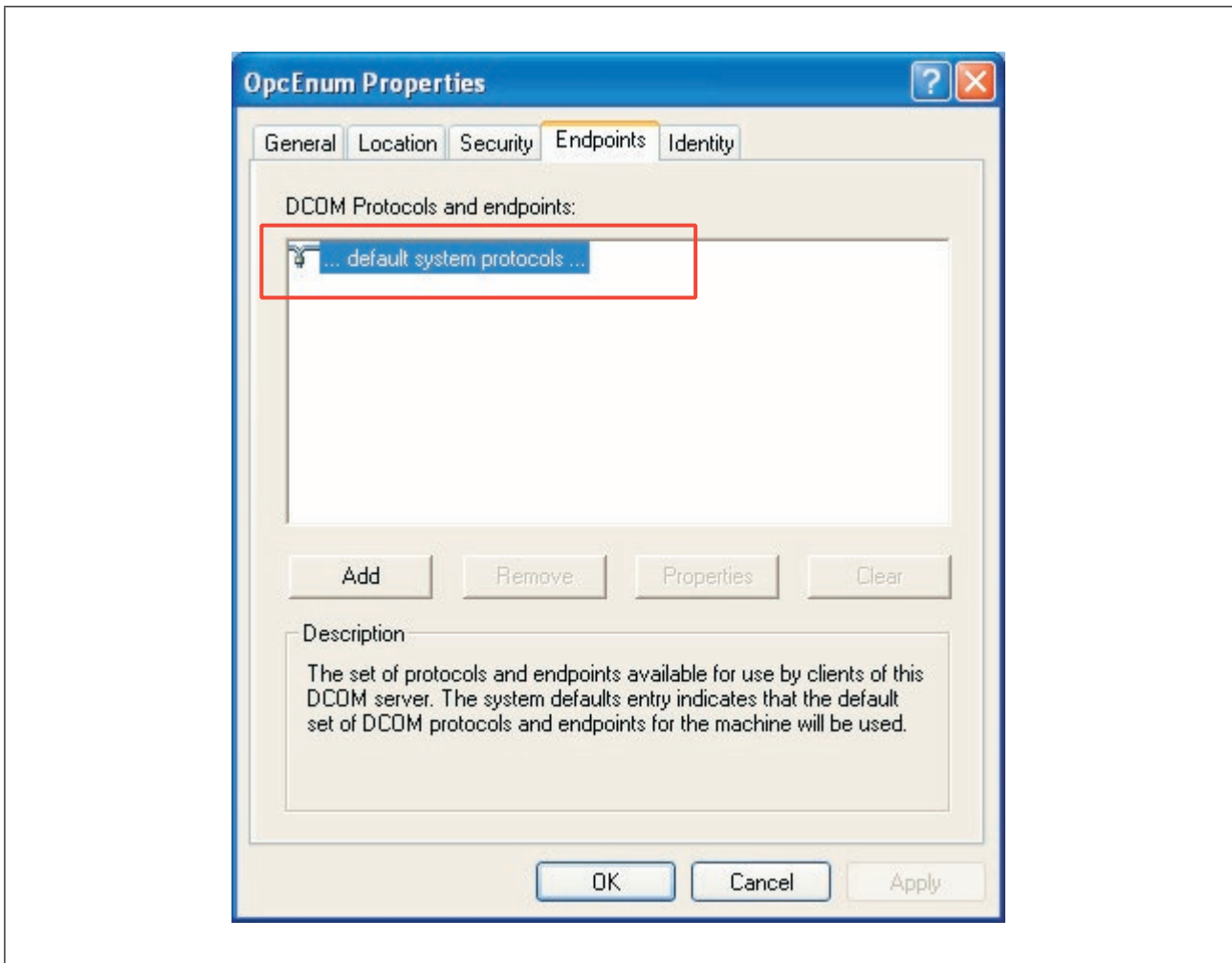


"Identity" tab (COM Security for OPCEnum)

- ▶ Select the **Endpoints** tab.

If **Default system protocols** is not available in the list, add the connection protocol as follows:

- Click on **Add** and select the **Connection-oriented TCP/IP** protocol under **Protocol Sequence**.
- Click **OK**.
The **Endpoints** tab is reactivated.



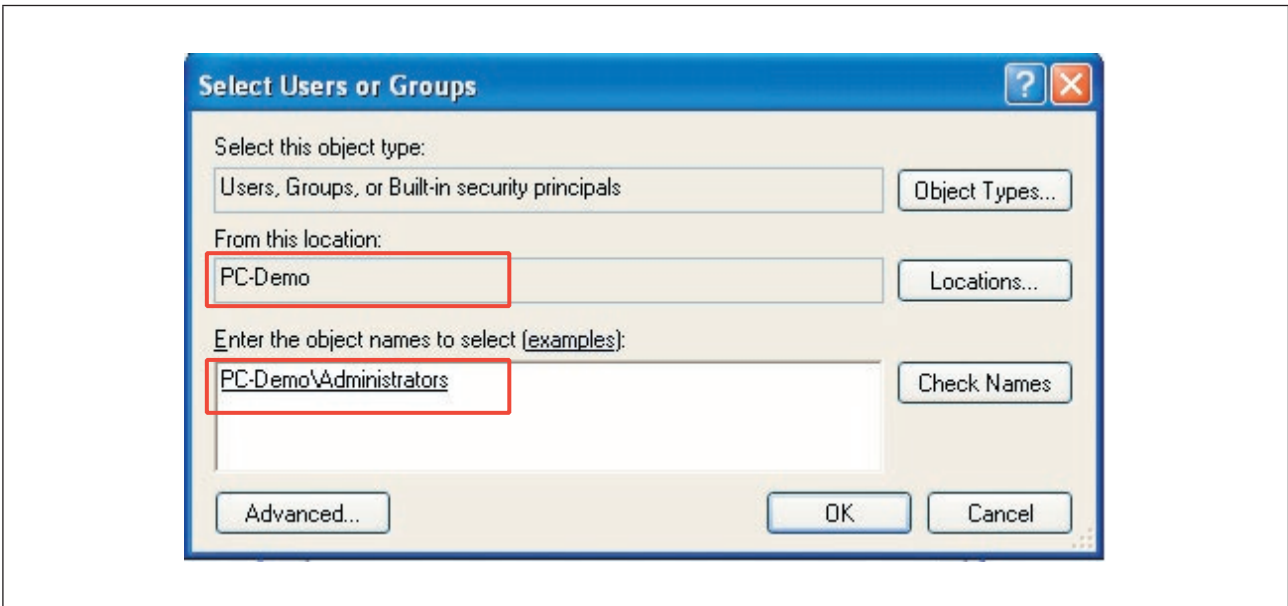
"Endpoints" tab (connection protocols for OPCEnum)

- ▶ Click **OK** to save all the settings and close the **OpcEnum Properties** window.
This completes the OPCEnum configuration for the OPC Server.
- ▶ Close the **Component Services** window.
The settings will take effect the next time the PC is restarted.

12.5.1.5 Special feature of the user "Administrators"

To be able to add the "Administrators" user to the group or user names of a permission, the search path must first be adapted:

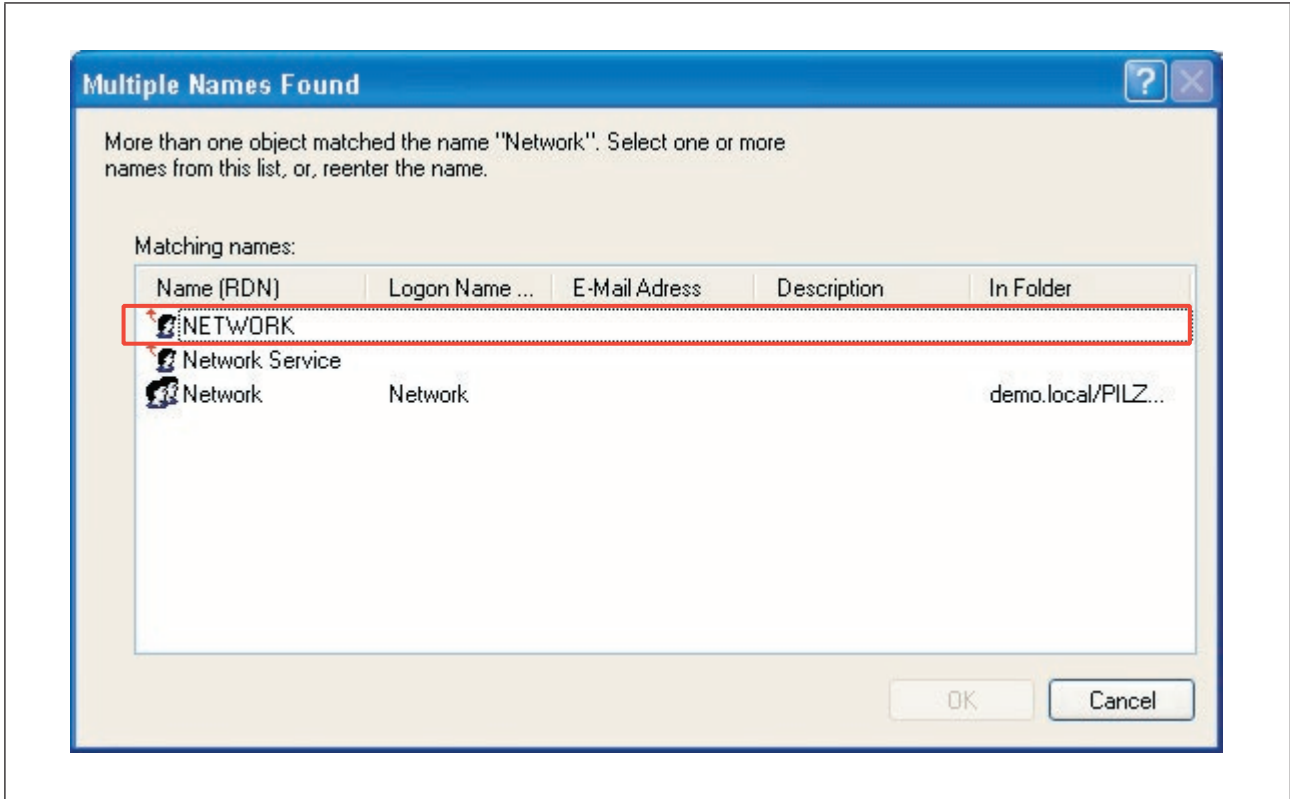
- ▶ In the relevant launch permission window click **Add...** and then **Paths...**
- ▶ Select the PC where you perform the DCOM configuration (in the example this is "PC-Demo") as a (search) path and confirm with **OK**.
- ▶ Enter "Administrators" as an object name and click **OK**.



"Select Users or Groups" window

12.5.1.6 Special feature of the user "NETWORK"

If you want to add the NETWORK user in one of the permission windows (e. g. Launch Permission window, Access Permission window), you may receive several users for selection when entering the user name "Network". In this case, select the "Network" user.



Select NETWORK user

12.5.2 Firewall configuration

A firewall or any other security mechanisms on the PC and network may prevent the OPC Server from communicating correctly with the connected devices. If this is the case it will be necessary to adjust the security settings and firewall settings. Current documentation on the required security settings is available on the website for the OPC Foundation, under "Downloads" and "White Papers": www.opcfoundation.org

Firewalls allow access to explicitly registered programs. The following programs must have access:

- ▶ All OPC Clients
- ▶ Microsoft Management Console
- ▶ PSS4000_OPCTServerDA.exe
- ▶ OPCenum.exe

A port must also be enabled for data exchange:

- ▶ Name: DCOM
- ▶ Port number 135
- ▶ Log: TCP

These enables grant access rights to the PC for other users on the network or from the Internet. A network administrator can optimise these Firewall settings to protect the server from unauthorised access.

12.6 Commissioning of the OPC Server

To commission the OPC Server, proceed as follows:

1. Configure the OPC Server in PAS4000 (see Online help of PAS4000).
2. Perform the project download in PAS4000.

As the project is downloaded, the device project for the OPC Server is created and is stored on the OPC Server. The device project for the OPC Server contains the OPC configuration, which consists of the configuration data and the complete namespace for all the devices configured on the project.

Once the project has been downloaded, the OPC Server makes the complete namespace for all devices configured on the project available to the OPC Clients.

12.7 Data type assignment

Various data types are assigned through the OPC standard. The table below contains the data types supported via the OPC Server and their assignment to the data types available in the user program on the PSS 4000:

OPC	User program (PSS 4000)
VT_I1	SINT
VT_I2	INT
VT_I4	DINT
VT_I8	LINT
VT_UI1	USINT
	BYTE
VT_UI2	UINT
	WORD
VT_UI4	UDINT
	DWORD
VT_UI8	ULINT
	LWORD
VT_R4	REAL
VT_R8	LREAL
VT_BOOL	BOOL
VT_BSTR	STRING
VT_DATE	TIME
	DATE
	TIME_OF_DAY
	DATE_AND_TIME

12.8 Namespace

The namespace is a memory area in which the PSS 4000 OPC Server stores all the general and device-based project data from a PAS4000 project. For the most part the namespace can be viewed using the browser on an OPC Client.

The OPC Server always contains the namespace that was transferred with the device project the last time a project was downloaded.

In the namespace, an OPC Client has the following access permissions with regard to variables in the user program, irrespective of whether these were created within blocks or in the resource global variables:

- ▶ Project's ST resources
 - Read access to the PI variables and local variables
 - Read/write access to local variables
- ▶ Project's FS resources
 - Read access to the PI variables and local variables

In the following section the namespace is represented in a tree structure. The tree branches into various levels, with corresponding sublevels. The branch points are called nodes and are labelled. The nodes are used for data structuring. Data is found exclusively on the lowest level. The display format may vary, depending on the browser on the OPC Client. However, the structure remains the same.

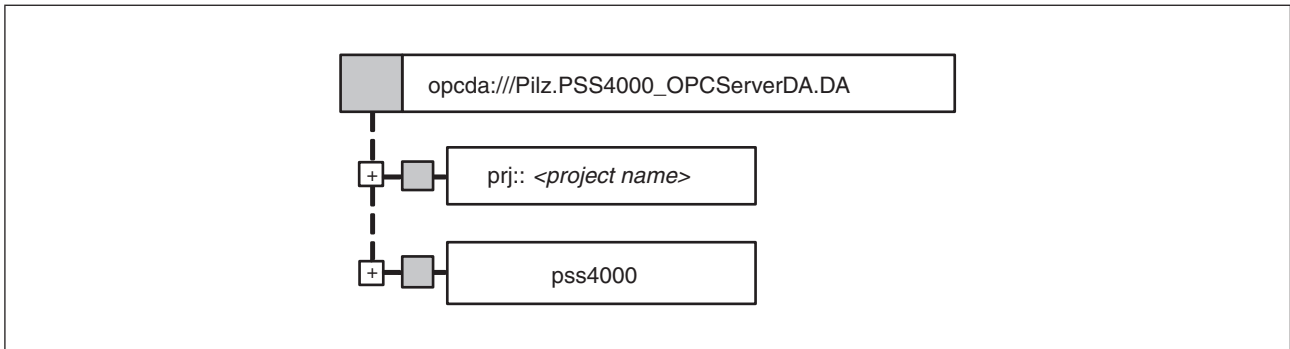
Due to the complexity, the section below can only describe the individual areas consecutively. To ensure that you still retain an overview, the complete path through which the area's node is accessed is stated at the start of each area. As when stating the path, a point is used as the separator between the nodes (e.g. pss4000.net.myDevice1.Device.State).

Definition of symbols

The following symbols are used in the diagrams and tables:

- ▶ Names in angle brackets are markers and are replaced by the respective name. <device> becomes myDevice1, for example.

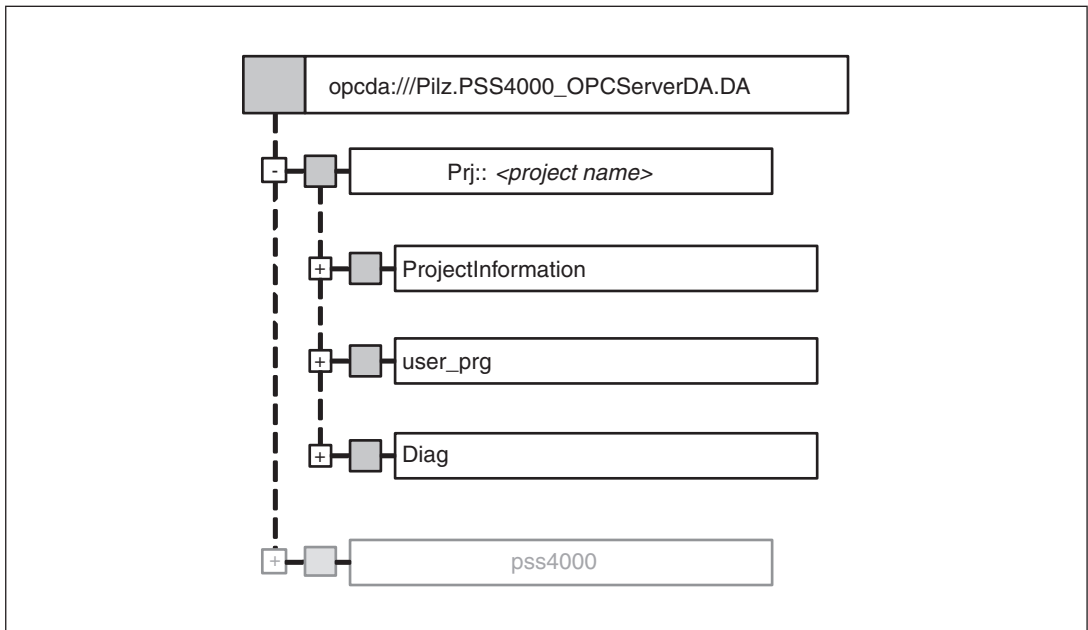
Namespace structuring (main area)



Node names in the main area

Node name	Meaning
<i>prj::<project name></i>	Area for the general project data containing the project name that was configured in PAS4000. e.g.: "myProject"
pss4000	Area for the device-based project data

12.8.1 General project data Area structuring



Structuring of "General project data"

Node name (path statement)	Meaning	
prj::<project name>	ProjectInformation	Area containing project-based information
	(prj::<project name>.ProjectInformation.)	
	user_prg	Area containing information about the program (e.g. variables of POUs, variables of component blocks, resource global variables)
	(prj::<project name>.user_prg.)	
	Diag	Area containing diagnostic list messages and diagnostic log entries
	(prj::<project name>.Diag.)	

12.8.1.1 Projectinformation

Path statement: prj::<project name>.ProjectInformation.

Node number	Node name	Meaning
1	ChecksumFSProject	Check sum "FS-Project" Data type: VT_UI4
	(prj::<project name>.ProjectInformation.ChecksumFSProject)	



Information

The check sum "FS-Project" is displayed as a hexadecimal value in PAS4000 and the PSS 4000 Diag Control. To ensure that the check sum can be displayed correctly on a 3rd party visualisation system without using PSS 4000 Diag Control, please select: "Hexadecimal display" for the VT_UI4 value.

12.8.1.2 user_prg

Path statement: prj::<project name>.user_prg.


Node number	Node name (path statement)	Meaning	
1	var_global	Area for the resource global variables of the project The area includes all resource global variables created by the user.	
		Access rights to resource global variables in the OPC server:	
		FS-PI variables	Read
		FS variables of type VAR	
		ST-PI variables	
		ST variables of type VAR	Read/write
(prj::<project name>.user_prog.var_global.)			
1.1	<var global name>	Name of a resource global variable	
	prj::<project name>.user_prog.var_global.<var global name>		
2	app	Instance path of the variables of the project (see online help of PAS4000) The area includes the instance path of the project with the PI variables and variables of type VAR created by the user. Please note: The variables of type VAR_IN_OUT, VAR_EXTERNAL and VAR_TEMP are not displayed in the OPC server.	
		Access rights to the variables in the OPC server:	
		FS-PI variables	Read
		FS variables of type VAR	
		ST-PI variables	
		ST variables of type VAR	Read/write
Individual element variable: prj::<project name>.user_prog.app.<Instance path of variable> .<Name of variable>			
Multi element variable: prj::<project name>.user_prog.app.<Instance path of variable>.<Name of variable>.<Index of the array element/name of the structure element>			

12.8.1.3 **Diag**

Path statement: prj::<project name>.Diag.

Diagnostic list:

Node number	Node name (path statement)	Meaning
1	List	Area containing diagnostic list messages
	prj::<project name>.Diag.List.)	
1.1	Number	Number of messages for a severity level
	(prj::<project name>.Diag.List.Number.)	
1.1.1	Error	Number of messages of "Error" severity Data type: VT_UI4
1.1.2	Warning	Number of messages of "Warning" severity Data type: VT_UI4
1.1.3	Information	Number of messages of "Status information" severity Data type: VT_UI4
1.2	Line	Lines in the diagnostic list
	(prj::<project name>.Diag.List.Line.)	
1.2.1	<LineNumber>	Information on line number Value range: 1 ... 100
	(prj::<project name>.Diag.List.Line.<line number>.)	
1.2.1.1	PriorityNo	Number of priority Data type: VT_UI1
1.2.1.2	Priority	Name of priority Data type: VT_BSTR
1.2.1.3	Description	Message text Data type: VT_BSTR
1.2.1.4	ScopeNo	Number of scope Data type: VT_UI1
1.2.1.5	Scope	Name of scope Data type: VT_BSTR
1.2.1.6	Timestamp	Time stamp Data type: VT_DATE
1.2.1.7	Did	Diagnostic identifier Data type: VT_BSTR
1.2.1.8	StateNo	Status number of diagnostic message
		Status number Meaning
	1	Message is active
	Data type: VT_UI1	

Node number	Node name (path statement)	Meaning
1.2.1.9	 State	Status of diagnostic message: Data type: VT_BSTR
1.2.1.10	ParameterList	Parameter list Data type: VT_BSTR
<hr/>		
1.2.1.11	Location (prj::<project name>.Diag.List.Line.<line number>.Location.)	Location information
1.2.1.11.1	Address	Address Data type: VT_BSTR
1.2.1.11.2	EquipmentId	Equip-ID Data type: VT_BSTR
1.2.1.11.3	Device	Device Data type: VT_BSTR
1.2.1.11.4	Description	Location description Data type: VT_BSTR
<hr/>		
1.2.1.12	Action (prj::<project name>.Diag.List.Line.<line number>.Action.)	Remedial actions
1.2.1.12.1	<action number> (prj::<project name>.Diag.List.Line.<line number>.Action.<action number>.)	Number of remedy Value range: 1 ... 8
1.2.1.12.1.1	Description	Action text Data type: VT_BSTR
1.2.1.12.1.2	ScopeNo	Number of scope Data type: VT_UI1
1.2.1.12.1.3	Scope	Name of scope Data type: VT_BSTR
1.2.1.12.1.4	ResponsibilityNo	Number of responsibility Data type: VT_UI1
1.2.1.12.1.5	Responsibility	Name of responsibility Data type: VT_BSTR
<hr/>		
1.2.1.12.1.6	Location (prj::<project name>.Diag.List.Line.<line number>.Action.<action number>.Location.)	Location information
1.2.1.12.1.6.1	Address	Address Data type: VT_BSTR
1.2.1.12.1.6.2	EquipmentId	Equip-ID Data type: VT_BSTR
1.2.1.12.1.6.3	Device	Device Data type: VT_BSTR

Node number	Node name (path statement)	Meaning
1.2.1.12.1.6.4	Description	Location description Data type: VT_BSTR

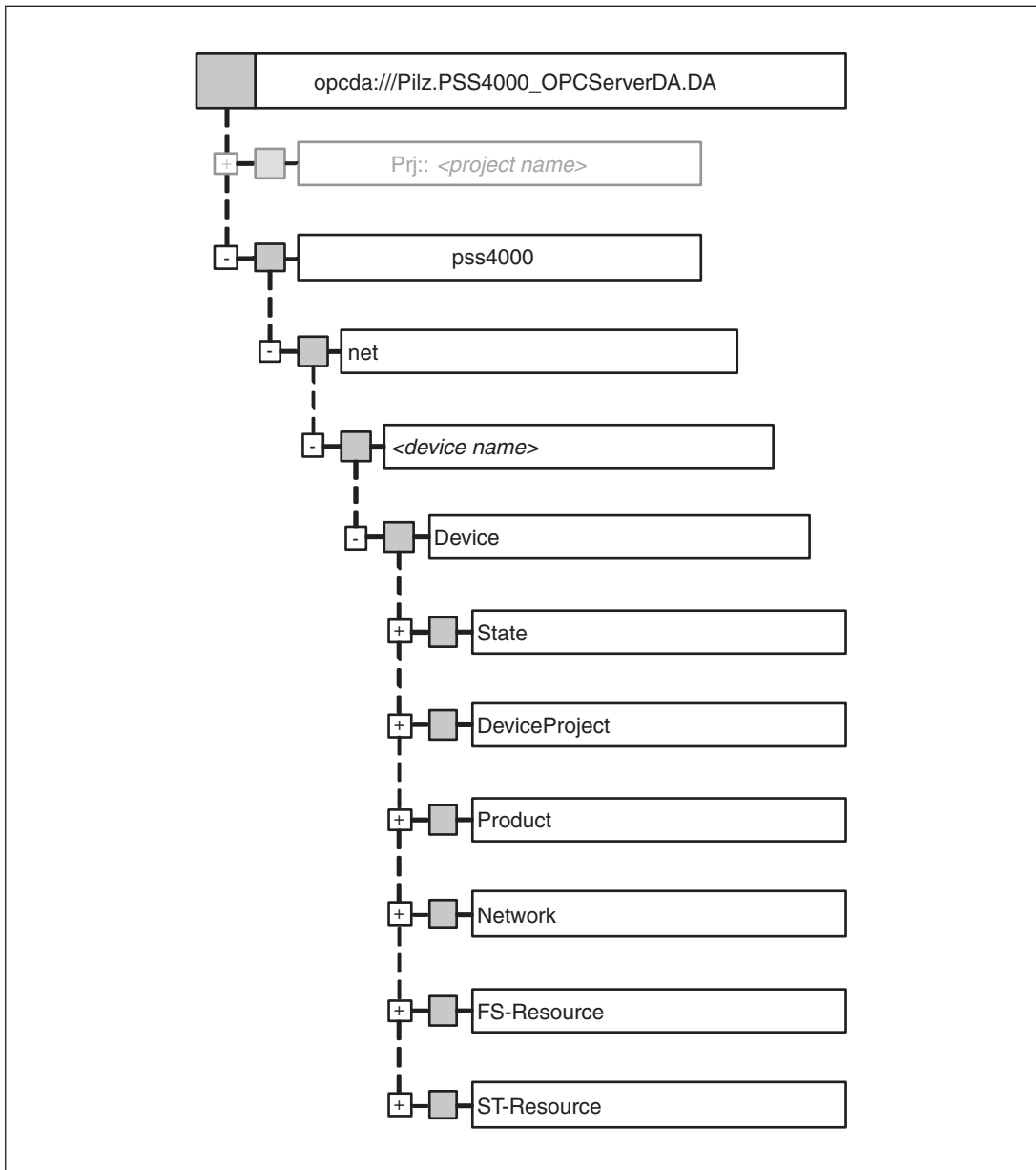
Diagnostic log:

Node number	Node name (path statement)	Meaning	
2	Log	Area containing diagnostic log entries (prj:: <i><project name></i> .Diag.Log.)	
2.1	Number	Number of log entries Data type: VT_UI4	
2.2	Line	Lines in the diagnostic log (prj:: <i><project name></i> .Diag.Log.Line.)	
2.2.1	<line number>	Information on line number Value range 1 ... "Number" (see 2.1; max. 4096) (prj:: <i><project name></i> .Diag.Log.Line.< <i>line number</i> >.)	
2.2.1.1	PriorityNo	Number of priority Data type: VT_UI1	
2.2.1.2	Priority	Name of priority Data type: VT_BSTR	
2.2.1.3	Description	Message text Data type: VT_BSTR	
2.2.1.4	ScopeNo	Number of scope Data type: VT_UI1	
2.2.1.5	Scope	Name of scope Data type: VT_BSTR	
2.2.1.6	Timestamp	Time stamp Data type: VT_DATE	
2.2.1.7	Did	Diagnostic identifier Data type: VT_BSTR	
2.2.1.8	StateChangeNo	Status number for the status change	
		Status number Meaning	
		0	Message cleared
		1	Message arrived
		101	Log entry implemented (for operational entries, self test entries and debugging entries)
		Data type: VT_UI1	

Node number	Node name (path statement)	Meaning	
2.2.1.9	StateChange	Status change	
		Display	Meaning
		cleared	Message cleared
		arrived	Message arrived
		logged	Log entry implemented (for operational entries, self test entries and debugging entries)
		Data type: VT_BSTR	
2.2.1.10	ParameterList	Parameter list Data type: VT_BSTR	
2.2.1.11	Location	Location information	
	(prj::<project name>.Diag.Log.Line.<line number>.Location.)		
2.2.1.11.1	Address	Address Data type: VT_BSTR	
2.2.1.11.2	EquipmentId	Equip-ID Data type: VT_BSTR	
2.2.1.11.3	Device	Device Data type: VT_BSTR	
2.2.1.11.4	Description	Location description Data type: VT_BSTR	

12.8.2 Device-based project data

Area structuring



Structuring of "Device-based project data"

Node name		Meaning
pss4000.	net.	Network area
	<device name>.	Area of device containing the device name configured during device naming (PSS 4000-excluding/functional device)
	Device.	Area containing device-based information
	State	Area containing information about the device's diagnostic state (pss4000.net.<device name>.Device.State.)
	DeviceProject	Area containing information about the device project (pss4000.net.<device name>.Device.DeviceProject.)
	Product	Area containing product information (pss4000.net.<device name>.Device.Product.)
	Network	Area containing the device's network parameters (pss4000.net.<device name>.Device.Network.)
	FS resource	Area containing information about tasks of the FS resource (pss4000.net.<device name>.Device.FS-Resource.)
	ST resource	Area containing information about tasks of the ST resource (pss4000.net.<device name>.Device.ST-Resource.)

12.8.2.1 State

Path statement: pss4000.net.<device name>.Device.State.

Node number	Node name	Meaning		
1	Diag	State of the system sections on a PSS 4000 device		
		Status	Meaning	
		0	Device diagnostic list and device diagnostic log are being prepared	
		1	Start of "deliberate operator action" (function of reset button)	
		2	A major FS error is present for at least one FS system section (see diagnostic list).	
		3	A message of "Error" severity is present for at least one system section (see diagnostic table).	
		4	A message of "Warning" severity is present for the PSSu device (see diagnostic table).	
		5	No message of "Error" or "Warning" severity is present for the device.	
Data type: VT_UI1				

Node number	Node name	Meaning		
2	STRun	State of ST resource		
		Status	Meaning	
		0	ST resource has not been started	
		1	Operating state "ST resource in RUN condition with error": - Task in TERMINATED condition or - Task in STOP condition At least one ST resource task is not running. The project is licensed.	
		2	Operating state "ST resource in RUN condition without error": The ST resource tasks are running without fault. The project is licensed.	
		3	Operating state "ST resource in RUN condition with error": - Task in TERMINATED condition or - Task in STOP condition At least one ST resource task is not running. The project is unlicensed.	
		4	Operating state "ST resource in RUN condition without error": The tasks of the ST resource are running without fault. The project is unlicensed.	
Data type: VT_UI1				

Node number	Node name	Meaning	
3	FSRun	State of FS resource	
		Status	Meaning
		0	FS resource has not been started
		1	Operating state "FS resource in RUN condition with error": - Task in TERMINATED condition or - Task in STOP condition At least one FS resource task is not running. The project is licensed.
		2	Operating state "FS resource in RUN condition without error": The tasks of the FS resource are running without fault. The project is licensed.
		3	Operating state "FS resource in RUN condition with error": - Task in TERMINATED condition or - Task in STOP condition At least one FS resource task is not running. The project is unlicensed.
		4	Operating state "FS resource in RUN condition without error": The tasks of the FS resource are running without fault. The project is unlicensed.
		Data type: VT_UI1	
4	STForce	State of forcing on the ST resource	
		Status	Meaning
		0	Forcing on the ST resource is inactive
		1	Forcing on the ST resource is active
5	FSForce	State of forcing on the FS resource	
		Status	Meaning
		0	Forcing on the FS resource is inactive
		1	Forcing on the FS resource is active

Node number	Node name	Meaning
6	STSafetyNETp	State of the system section ST SafetyNET p RTFN
		Status Meaning
		0 Operating state "ST SafetyNET p RTFN in STOP condition with error: Major FS+ST error"
		1 ST SafetyNET p RTFN has not been started
		2 Operating state "ST SafetyNET p RTFN in RUN condition with minor error"
		3 Operating state "ST SafetyNET p RTFN in RUN condition without error"
		4 Operating state "ST SafetyNET p RTFN in STOP condition with error: Major FS+ST error"
		Data type: VT_UI1
7	FSSafetyNETp	State of the system section FS-SafetyNET p RTFN
		Status Meaning
		0 Operating state "FS SafetyNET p RTFN in STOP condition with error: Major FS error"
		1 FS SafetyNET p RTFN has not been started
		2 Operating state "FS SafetyNET p RTFN in RUN condition with minor error"
		3 Operating state "FS SafetyNET p RTFN in RUN condition without error"
		4 Operating state "FS SafetyNET p RTFN in STOP condition with error: Major FS+ST error"
		Data type: VT_UI1

Node number	Node name	Meaning
8	Module bus	State of the system sections FS and ST module bus
		Status Meaning
		0 Operating state "Safe condition of all FS outputs on the device" or at least one module cannot be accessed (e.g. a module has been removed during operation, set/actual hardware registry does not match)
		1 No modules present
		2 FS and ST module bus are operating without fault
		3 Operating state "FS module bus in a STOP condition with error: Major FS error"
		Data type: VT_UI1
9	Media	State of data medium
		For PSS 4000 excluding/functional devices whose naming data are saved on a removable data medium:
		Status Meaning
		0 Device identification activated by user
		1 Removable data medium is missing or removable data medium not detected or removable data medium is defective
		2 Product type on the removable data medium does not match the PSS 4000-excluding/functional device (on PSSu systems: head module's product type) or no device project available on the removable data medium
		3 Naming data and device project on the PSS 4000-excluding/functional device and the removable data medium do not match
4 Naming data and device project on the PSS 4000-excluding/functional device and the removable data medium match		
Data type: VT_UI1		

Node number	Node name	Meaning										
10	SFProfibus	"SF" status LED (PROFIBUS-DP) The evaluation can only be performed in combination with the LEDs of the "BF" status LED (see operating manual for the device/head module).										
		<table border="1"> <thead> <tr> <th>Status</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>LED "lights red"</td> </tr> <tr> <td>1</td> <td>LED "off"</td> </tr> </tbody> </table>	Status	Meaning	0	LED "lights red"	1	LED "off"				
		Status	Meaning									
		0	LED "lights red"									
		1	LED "off"									
Data type: VT_UI1												
11	BFProfibus	"BF" status LED (PROFIBUS-DP) The evaluation can only be performed in combination with the LEDs of the "SF" status LED (see operating manual for the device/head module).										
		<table border="1"> <thead> <tr> <th>Status</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>LED "lights red"</td> </tr> <tr> <td>1</td> <td>LED "flashes red"</td> </tr> <tr> <td>2</td> <td>LED "lights green"</td> </tr> <tr> <td>3</td> <td>LED "off"</td> </tr> </tbody> </table>	Status	Meaning	0	LED "lights red"	1	LED "flashes red"	2	LED "lights green"	3	LED "off"
		Status	Meaning									
		0	LED "lights red"									
		1	LED "flashes red"									
2	LED "lights green"											
3	LED "off"											
Data type: VT_UI1												
12	SOPROFIsafe	"SO" status LED (PROFIsafe) Status of PROFIsafe in accordance with PROFIsafe specification										
		<table border="1"> <thead> <tr> <th>Status</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>LED "off" No F-module is configured or An F-module is configured but it is neither in "Operator Acknowledge Requested" nor "Safe Operation" state</td> </tr> <tr> <td>1</td> <td>LED "flashes green" "Operator Acknowledgement" is requested by at least one F-module (OA_C requested)</td> </tr> <tr> <td>3</td> <td>LED "lights green" PROFIsafe communication is ok</td> </tr> </tbody> </table>	Status	Meaning	0	LED "off" No F-module is configured or An F-module is configured but it is neither in "Operator Acknowledge Requested" nor "Safe Operation" state	1	LED "flashes green" "Operator Acknowledgement" is requested by at least one F-module (OA_C requested)	3	LED "lights green" PROFIsafe communication is ok		
		Status	Meaning									
		0	LED "off" No F-module is configured or An F-module is configured but it is neither in "Operator Acknowledge Requested" nor "Safe Operation" state									
		1	LED "flashes green" "Operator Acknowledgement" is requested by at least one F-module (OA_C requested)									
3	LED "lights green" PROFIsafe communication is ok											
Data type: VT_UI1												

Node number	Node name	Meaning										
13	MSEtherNetIP	Status LED "MS" (EtherNet/IP) Module status in accordance with EtherNet/IP specification										
		<table border="1"> <thead> <tr> <th>Status</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>LED "lights red" A major FS error is present for at least one FS system section.</td> </tr> <tr> <td>1</td> <td>LED "flashes red" At least one message of "Warning" severity is present for the device, no message of "Error" severity is present and no invalid data is being downloaded to the scanner.</td> </tr> <tr> <td>2</td> <td>LED "lights green" No message of "Error" or "Warning" severity is present for the device.</td> </tr> <tr> <td>3</td> <td>LED "off" No supply voltage or device inactive or device not configured</td> </tr> </tbody> </table>	Status	Meaning	0	LED "lights red" A major FS error is present for at least one FS system section.	1	LED "flashes red" At least one message of "Warning" severity is present for the device, no message of "Error" severity is present and no invalid data is being downloaded to the scanner.	2	LED "lights green" No message of "Error" or "Warning" severity is present for the device.	3	LED "off" No supply voltage or device inactive or device not configured
		Status	Meaning									
		0	LED "lights red" A major FS error is present for at least one FS system section.									
		1	LED "flashes red" At least one message of "Warning" severity is present for the device, no message of "Error" severity is present and no invalid data is being downloaded to the scanner.									
		2	LED "lights green" No message of "Error" or "Warning" severity is present for the device.									
3	LED "off" No supply voltage or device inactive or device not configured											
Data type: VT_UI1												
14	NSEtherNetIP	Status LED "MS" (EtherNet/IP) Network status in accordance with EtherNet/IP specification										
		<table border="1"> <thead> <tr> <th>Status</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>LED "flashes red" Connection in Timeout</td> </tr> <tr> <td>1</td> <td>LED "flashes green" No network connection</td> </tr> <tr> <td>2</td> <td>LED "lights green" Network connection is available and EtherNet/IP communication is ok</td> </tr> <tr> <td>3</td> <td>LED "off" No data traffic or EtherNet/IP not configured or no IP address configured</td> </tr> </tbody> </table>	Status	Meaning	0	LED "flashes red" Connection in Timeout	1	LED "flashes green" No network connection	2	LED "lights green" Network connection is available and EtherNet/IP communication is ok	3	LED "off" No data traffic or EtherNet/IP not configured or no IP address configured
		Status	Meaning									
		0	LED "flashes red" Connection in Timeout									
		1	LED "flashes green" No network connection									
		2	LED "lights green" Network connection is available and EtherNet/IP communication is ok									
3	LED "off" No data traffic or EtherNet/IP not configured or no IP address configured											
Data type: VT_UI1												

12.8.2.2 DeviceProject

Path statement: pss4000.net.<device name>.Device.DeviceProject

Node number	Node name	Meaning
1	Name	Name of PAS4000 project to which the device <device name> belongs Data type: VT_BSTR
2	ChecksumFS	Check sum for the FS device project for the device <device name> Data type: VT_UI4
3	BuildTime	Time at which the PAS4000 project was built Data type: VT_DATE
4	Version	Version of the PAS4000 project to which the device <device name> belongs Data type: VT_BSTR
5	LicenceState	Status information on the licensing state of the project
		Status Meaning
		0 unlicensed or no project active
		1 licensed
		Data type: VT_UI1

12.8.2.3 Product

Path statement: pss4000.net.<device name>.Device.Product.

Node number	Node name	Meaning
1	SerialNo	Serial number of the device <device name>, PSSu systems: Serial number of the head module Data type: VT_BSTR
2	OrderNo	Order number of the device <device name>, PSSu systems: Order number of the head module Data type: VT_BSTR
3	ProductType	Product type of the device <device name>, PSSu systems: Name of the head module Data type: VT_BSTR
4	FirmwareVersion	Firmware version of the device <device name>, PSSu systems: Firmware version of the head module Data type: VT_BSTR
5	FirmwareRevision	Revision number of the firmware for the device <device name>, PSSu systems: Revision number of the firmware for the head module Data type: VT_BSTR
6	FirmwareUpdate	Time at which the firmware for the device <device name> was last updated, PSSu systems: Time at which the firmware for the head module was last updated Data type: VT_DATE
7	HardwareVersion	Hardware version of the device <device name>, PSSu systems: Hardware version of the head module Data type: VT_BSTR

12.8.2.4 Network

Path statement: pss4000.net.<device name>.Device.Network.

Node number	Node name	Meaning
1	IPAddressV4	IP address of the device <device name> Data type: VT_BSTR
2	SubnetMaskV4	Subnet mask Data type: VT_BSTR
3	GatewayV4	IP address of the Standard Gateway Data type: VT_BSTR
4	DHCP	DHCP setting for the device <device name>
		Display Meaning
		FALSE DHCP not activated
		TRUE DHCP activated
Data type: VT_BOOL		
5	DNS1V4	Address of preferred DNS Server Data type: VT_BSTR
6	DNS2V4	Address of alternative DNS Server Data type: VT_BSTR
7	MACAddress	MAC address of the device <device name> Data type: VT_BSTR
8	SNTP	SNTP setting for the device <device name>
		Display Meaning
		FALSE Device is not configured as SNTP server of the project
		TRUE Device works in the project as SNTP Server
Data type: VT_BOOL		
9	SNTPV4	IP address of the SNTP Server from which the device obtains its time Data type: VT_BSTR
10	RealTimeClock	Current time on the device Data type: VT_DATE

12.8.2.5 FS resource

Path statement: pss4000.net.<device name>.Device.FS-Resource.

Node number	Node name (path statement)	Meaning
1	<task name>	Name of task in the FS resource of the device <device name>
	pss4000.net.<device name>.Device.FS-Resource.<task name>.)	
1.1	State	Status of task
		Status Meaning
		1 Task in STOP condition
		2 Task in RUN condition
		3 Task in TERMINATED condition
Data type: VT_UI4		
1.2	ProcessingTime	Task execution time $t_{ProcTask}$ The execution time is only available for periodic tasks (see Reaction times, Page 370)
	(pss4000.net.<device name>. Device.FS-Resource.<task name>.ProcessingTime.)	
1.2.1	Minimum	Minimum execution time The displayed time is the shortest execution time that has been detected since the start of the task. Data type: VT_UI4
1.2.2	Maximum	Maximum execution time The displayed time is the longest execution time that has been detected since the start of the task. Data type: VT_UI4
1.2.3	Current	Current task execution time Data type: VT_UI4

12.8.2.6 ST resource

Path statement: pss4000.net.<device name>.Device.ST-Resource.

Node number	Node name (path statement)	Meaning
1	<task name>	Name of task in the device's ST resource <device name>
	pss4000.net.<device name>.Device.ST-Resource.<task name>.)	
1.1	State	Status of task
		Status Meaning
		1 Task in STOP condition
		2 Task in RUN condition
		3 Task in TERMINATED condition
Data type: VT_UI4		
1.2	ProcessingTime	Task execution time $t_{ProcTask}$ The execution time is only available for periodic tasks (see Reaction times, Page 370)
	(pss4000.net.<device name>. Device.ST-Resource.<task name>.ProcessingTime.)	
1.2.1	Minimum	Minimum execution time The displayed time is the shortest execution time that has been detected since the start of the task. Data type: VT_UI4
1.2.2	Maximum	Maximum execution time The displayed time is the longest execution time that has been detected since the start of the task. Data type: VT_UI4
1.2.3	Current	Current task execution time Data type: VT_UI4

13 Diagnostic basics

Effective diagnostics are required in order to keep machine downtimes to a minimum. Diagnostics must provide all the information necessary to rectify faults quickly, prevent faults and analyse machine conditions.

The PSS 4000 distinguishes between system diagnostics and process diagnostics.

- ▶ System diagnostics

Diagnostics on hardware and firmware, including errors in the hardware configuration or user program.

The system diagnostics are provided by Pilz in full. Only a few adjustments can be made by the user, e.g. entering additional information for location information.

- ▶ Process diagnostics

Diagnostics on procedures and conditions. The process diagnostics are created by the user within the user program. Pilz blocks are supplied with process diagnostics, which users can adapt to suit their own requirements.

During operation, system messages and process messages are entered in a [diagnostic list](#). Each device has its own device diagnostic list. The diagnostic server combines the device diagnostic lists of all the devices in the project to form the project diagnostic list.

The diagnostic list only contains current messages. Once a fault has been rectified, the message is deleted from the diagnostic list.

The order in which the diagnostic messages appear in the diagnostic list is determined by the priority of the diagnostic message. That makes it easy to see which message is the most important.

In system diagnostics, a distinction is made between the first fault and subsequent faults. Only the first fault is reported. This avoids unnecessary messages in the diagnostic list.

As well as describing the event that has occurred and indicating its location, diagnostic messages also contain suggested remedies. These tell you what to do, which area is affected and who is responsible for rectifying the problem.

The [diagnostic log](#) records when a diagnostic message arrived and when it was cleared. The diagnostic log also contains information on any changes in the operating condition, e.g. if a device has been restarted, as well as information for Pilz technical support.

Each device has its own device diagnostic log. The diagnostic server combines the device diagnostic logs of all the devices in the project to form the project diagnostic log.

The [diagnostic server](#) forwards the project diagnostic list and log to the diagnostic clients (e.g. PSS 4000 OPC Server). The diagnostic clients then provide the information to the displays (e.g. PSS 4000 Diag Control).

The project diagnostic list and project diagnostic log are also displayed in PAS4000. The diagnostic list and diagnostic log in PAS4000 may vary slightly from the diagnostic list and diagnostic log on a display (e.g. PSS 4000 Diag Control), because they also contain messages/entries for online actions, for example.

The device diagnostic lists and logs are not displayed.

13.1 Diagnostic list

During operation, [diagnostic messages](#) are entered in diagnostic lists.

Diagnostic lists only contain current messages, i.e. diagnostic messages with the state "active" (see State of diagnostic messages, [Page 474](#)).

A new diagnostic message is entered in the diagnostic list if the state of the diagnostic message switches to "active". The diagnostic message is deleted once its state switches to "inactive".

Each device has its own device diagnostic list. The device diagnostic lists from all the devices in a project are combined within a project diagnostic list.

- ▶ Device diagnostic list
 - Generated by the device's firmware
 - Contains a device's diagnostic messages (raw data only, no texts)
 - Memory for at least 128 diagnostic messages
 - If the maximum number of diagnostic messages is reached, the diagnostic message with the lowest priority is deleted
 - Deleted in the event of an original reset
 - Cannot be displayed; used as the basis for the project diagnostic list
- ▶ Project diagnostic list
 - Generated by the [Diagnostic Server](#)
 - Contains plain text diagnostic messages for all the devices in a project
 - Memory for at least 256 diagnostic messages
 - If the maximum number of diagnostic messages is reached, the diagnostic message with the lowest priority is deleted
 - Always up-to-date as it is continuously synchronised with the device diagnostic lists
 - Should the diagnostic server lose the connection to a device, all the device's diagnostic messages are deleted from the project diagnostic list and a corresponding diagnostic message is entered in the project diagnostic list
 - The diagnostic list in PAS4000 may vary slightly from the diagnostic list on a display (e.g. PSS 4000 Diag Control), because it also contains messages for online actions, for example.

13.1.1 Diagnostic messages

The following diagnostic messages are available:

- ▶ System messages

Message that is issued when an event occurs in system diagnostics.

- ▶ Process messages

Message that is issued when an event occurs in process diagnostics, i.e. the set condition is met.

During operation, diagnostic messages are entered in a diagnostic list.

A diagnostic message contains a description of the event. Each diagnostic message includes a remedy. The remedy describes how to react to the event. The remedy contains a maximum of 8 steps, the so-called "Actions".

Components of a diagnostic message:

- ▶ Time stamp

Time at which the diagnostic message was activated

- ▶ Diagnostic identifier

In system diagnostics, the diagnostic identifier is used for unique identification of a diagnostic message. In process diagnostics, the diagnostic identifier is used for unique identification of the diagnostic item responsible for the diagnostic message.

(see Diagnostic identifier, [Page 483](#))

- ▶ Severity and priority

Importance of the diagnostic message

(see Severity and priorities, [Page 475](#))

- ▶ Scope

Approximate description of the kind of event

(see Scopes, [Page 477](#))

- ▶ Description (message text)

Description of what has happened

- ▶ Location information

Information about the location at which the event occurred (see Location information, [Page 478](#)).

The diagnostic message's location information contains the following details:

- Device
- Equip-ID
- Address
- Location description

- ▶ Parameter list

Contains all the information required by technical support to evaluate a diagnostic message. This is the diagnostic identifier and all parameters.

- ▶ State

Indicates the state of the diagnostic message: "active" or "inactive" (see State of diagnostic messages, [Page 474](#))

Action components

- ▶ Scope

Approximate description of the kind of action
(see Scopes, [Page 477](#))

- ▶ Responsibility

Information about who should undertake the action
(see Responsibilities, [Page 476](#))

- ▶ Description (action text)

Description of what has to be done

- ▶ Location information

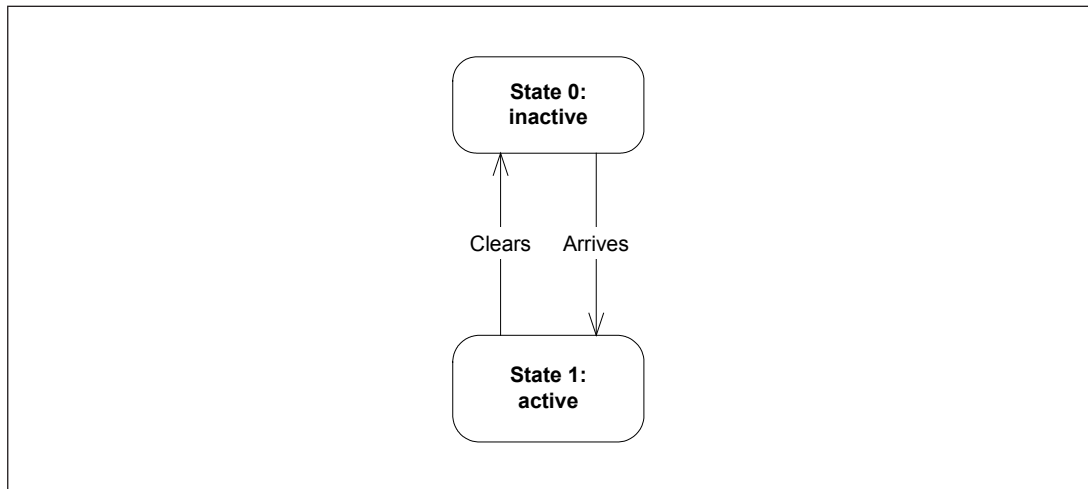
Information about the location at which the action is to be taken (see Location information, [Page 478](#)).

The location information for action components contains the following details:

- Device
- Equip-ID
- Address
- Location description

13.1.1.1 State of diagnostic messages

The state of a diagnostic message depends on the event that triggered it.



State of diagnostic messages

▶ System diagnostics

If an event is present (e.g. short circuit at an input), the diagnostic message becomes "active" (change of state: "arrives"). If the fault has been rectified, the diagnostic message becomes "inactive" (status change: "clears").

▶ Process diagnostics

If the programmed condition is met, the diagnostic message becomes "active" (change of state: "arrives"). If the condition is no longer met, the diagnostic message becomes "inactive" (change of state: "clears").

The diagnostic list only contains diagnostic messages with the state 1.




Each time the state of a diagnostic message changes, a log entry is triggered in the diagnostic log.

Changes in state are symbolised by the following character strings:

- ▶ <+> for "arrived"
- ▶ <-> for "cleared"

13.1.1.2 Severity and priority

Diagnostic messages are differentiated according to their importance. They are divided into three levels of severity:

- 
Error
 Diagnostic message to which the user must react immediately. This diagnostic message provides information on system conditions in which one or more functions have failed or are adversely affected.
- 
Warning
 Diagnostic message which demands the user's attention. This diagnostic message warns the user about potential or imminent critical system conditions.
- 
Status information
 Diagnostic message which describes the current condition of the process and/or system.. This diagnostic message is provided for information; operation is neither impeded nor adversely affected.

Priorities allow a more precise classification. Each level of severity has 5 priorities. Priority 1 is the highest priority and priority 15 the lowest.

Severity	Priority	Name of priority
Error	1	
	2	
	3	Error
	4	
	5	
Warning	6	
	7	
	8	Warning
	9	
	10	
Status information	11	
	12	
	13	Status information
	14	
	15	

The user can assign names to these priorities. The names for priority 3 "Error", priority 8 "Warning" and priority 13 "Status information" are pre-defined by Pilz because they are used in system diagnostics and in Pilz-generated process diagnostics.

These pre-defined names can be modified by the user but cannot be deleted.

13.1.1.3 Responsibilities

The responsibility stands for the person or group who is supposed to perform the remedial action.

Some responsibilities are pre-defined by Pilz and a maximum of 15 responsibilities can be defined by the user. The pre-defined responsibilities are used in system diagnostics and in Pilz-generated process diagnostics. The user can change the name of the pre-defined responsibilities, but cannot delete them.

Pre-defined responsibilities:

▶ Operator

This action involves the release, actuation or operation of a safety device.

Responsible person: Plant operator, for example

▶ Electrical maintenance engineer

This action refers to the mechanics/electrics/electronics outside the Pilz devices, e.g. rectify short circuit, check supply voltage, check switch contacts.

Responsible person: Maintenance engineer, for example

▶ Device exchange

The device is defective and must be exchanged.

Responsible person: Maintenance engineer, for example

▶ Project engineer

This action can only be performed with PAS4000 or requires some programming knowledge, e.g. check or adapt hardware configuration, check or adapt programming.

Responsible person: Maintenance engineer, programmer, project engineer, for example

▶ Communication

This action can be performed by users with a technical understanding of SafetyNET p/IP connections/fieldbuses, who can check the bus load or check cable lengths, for example.

Responsible person: Maintenance engineer, bus specialist, for example

▶ ---

No details of responsibility are possible or necessary.

13.1.1.4 Scopes

The scope roughly describes the kind of event/action. The scope is stated for diagnostic messages and remedial actions.

Some scopes are pre-defined by Pilz and a maximum of 15 scopes can be defined by the user. The pre-defined scopes are used in system diagnostics and in Pilz-generated process diagnostics. The user can change the name of the pre-defined scopes, but cannot delete them.

Pre-defined scopes:

- ▶ Hardware

Affects the mechanics, electrics, electronics or operating system.

Examples: Defective device, periphery error (e.g. wiring, actuator/sensor or supply voltage is faulty), bus error (e.g. error in SafetyNET p or Ethernet)

- ▶ Project configuration

Affects the user program and hardware configuration.

Examples: Division by zero in the user program

- ▶ Operation

Affects sequence or operation.

Examples: Sequence error (cylinder fails to reach an end position), safety device was triggered (e.g. E-STOP operated)

- ▶ Plant

Affects the hardware or project configuration.

- ▶ ---

No details of the scope are possible or necessary.

13.1.1.5 Location information

Location information is available for diagnostic messages and for the individual remedial actions. The diagnostic message contains details of the location in which the event occurred. The information for the action includes details of where the action is to be taken.

The location information contains the following details:

- ▶ Device
Relevant element
- ▶ Equip-ID
The equipment identifier is used to identify a piece of hardware in a project and should be unique.
- ▶ Address
The address is generated automatically.
- ▶ Location description
Additional information about a location, e.g. information about the control cabinet in which a device is located.

The information content depends on the location. The following locations are possible:

- ▶ Module/device
- ▶ Terminal
- ▶ Sensor/actuator
- ▶ Block instance
- ▶ Position in the user program
- ▶ Other locations

The individual locations and location information details are described below.



Information

If "<???" is displayed in the diagnostic list or diagnostic log instead of location information, either the current project has not been downloaded to all devices or not all the I/O mappings have been performed (see Diagnostic Server, [Page 488](#)).

Module/device

Only used in system diagnostic messages and log entries.

- ▶ Device
Text that the user has entered in the Device Editor (e.g. PSSu Module Editor) as "Device" location information for the module/device; the default setting is the product type of the module/device (e.g. "PSSu E F 4DI").
In rare cases the situation can arise on PSSu systems in which it is impossible to determine which module triggered a diagnostic message. In this case, the text "anonymous module" is displayed as the device.
- ▶ Equip-ID
Equip-ID of the module/device (e.g. "-4A1"), entered by the user in the Device Editor (e.g. PSSu Module Editor)

- ▶ Address
Automatically generated address of the module/device
- ▶ Location description
Additional information on the module/device, entered by the user in the Device Editor (e.g. PSSu Module Editor)

Terminal

In the system diagnostics or process diagnostics, if the underlying PI variable is mapped to an I/O datum from the module bus:

- ▶ Device
"Device" location information for the module/device on which the terminal is located (see "Device" under "Module/device")
- ▶ Equip-ID
Equip-ID of the module/device on which the terminal is located, including details of the terminal (e.g. "-4A1:12", whereby "12" refers to the terminal)
- ▶ Address
Automatically generated terminal address
- ▶ Location description
Additional information on the module/device, entered by the user in the Device Editor (e.g. PSSu Module Editor); no location description can be entered for the terminal itself

In the process diagnostics, if the underlying PI variable is mapped to an external I/O datum:

- ▶ Device
"Device" location information for the module/device containing the external I/O datum (see "Device" under "Module/device")
- ▶ Equip-ID
Equip-ID of the module/device containing the external I/O datum (see "Equip-ID" under "Module/device")
- ▶ Address
Automatically generated address of the external I/O datum
- ▶ Location description
Location description of the module/device containing the external I/O datum (see "Location description" under "Module/device")

In the process diagnostics, if the underlying PI variable is mapped to another PI variable:

- ▶ Device
"Device" location information for the module/device on which the resource of the mapped PI variable is located (see "Device" under "Module/device")
- ▶ Equip-ID
Equip-ID of module/device on which the resource of the mapped PI variable is located (see "Equip-ID" under "Module/device")
- ▶ Address
Automatically generated address of the mapped PI variable

- ▶ Location description

Location description of the module/device on which the resource of the mapped PI variable is located (see "Location description" under "Module/device")

Sensor/actuator

In the system diagnostics or process diagnostics, if the underlying PI variable is mapped to an I/O datum from the module bus:

- ▶ Device

Text that the user has entered in the Device Editor (e.g. PSSu Module Editor) as "Device" location information for the sensor/actuator

- ▶ Equip-ID

Equip-ID of the sensor/actuator, entered by the user in the Device Editor (e.g. PSSu Module Editor)

- ▶ Address

Automatically generated address of the terminal to which the sensor/actuator is connected

- ▶ Location description

Additional information on the sensor/actuator, entered by the user in the Device Editor (e.g. PSSu Module Editor)

In the process diagnostics, if the underlying PI variable is mapped to an external I/O datum:

- ▶ Device

"Device" location information for the module/device containing the external I/O datum (see "Device" under "Module/device")

- ▶ Equip-ID

Equip-ID of the module/device containing the external I/O datum (see "Equip-ID" under "Module/device")

- ▶ Address

Automatically generated address of the external I/O datum

- ▶ Location description

Location description of the module/device containing the external I/O datum (see "Location description" under "Module/device")

In the process diagnostics, if the underlying PI variable is mapped to another PI variable:

- ▶ Device

"Device" location information for the module/device on which the resource of the mapped PI variable is located (see "Device" under "Module/device")

- ▶ Equip-ID

Equip-ID of the module/device on which the resource of the mapped PI variable is located (see "Equip-ID" under "Module/device")

- ▶ Address

Automatically generated address of the mapped PI variable

- ▶ Location description

Location description of the module/device on which the resource of the mapped PI variable is located (see "Location description" under "Module/device")

Block instance

Only used in process diagnostic messages and log entries.

- ▶ Device
Name of the type of Multi block/POU
- ▶ Equip-ID
Unused
- ▶ Address
Instance path of the Multi block/POU
- ▶ Location description
Unused

Position in the user program

Only used in system diagnostic messages and log entries.

- ▶ Device
Name of the type of Multi block/POU
- ▶ Equip-ID
Unused
- ▶ Address
Instance path of the Multi block/POU, with additional positional data
Positional data on the Multi block: X and Y co-ordinate in the Multi Editor; X is the column number (counting from left to right, from 1 to 30) and Y is the line number (counting from top to bottom, from 1 to 30)
Positional data with IL-POU and STL-POU: Line number
If a Multi block/POU is sealed, the instance path will end with details of the last unsealed Multi block/POU, before the affected Multi block/POU. The position names the point at which the sealed Multi block/POU is called.
- ▶ Location description
Unused

Other locations

If the location information listed above isn't suitable, the location information can be freely defined.

- ▶ Device
Process diagnostics: Any; defined in the type or instance of the diagnostic item
System diagnostics: Names the location (e.g. "Hardware configuration" or "User program")
- ▶ Equip-ID
Process diagnostics: Any; defined in the type or instance of the diagnostic item
System diagnostics: Unused
- ▶ Address
Unused

- ▶ Location description

Process diagnostics: Any; defined in the type or instance of the diagnostic item

System diagnostics: Unused

13.1.1.6 Diagnostic identifier

In system diagnostics, the diagnostic identifier is used for unique identification of a diagnostic message. In process diagnostics, the diagnostic identifier is used for unique identification of the diagnostic item responsible for the diagnostic message. In process diagnostics, the uniqueness is limited to the project.

The diagnostic identifier consists of three parts:

<Kind> - *<Class>* - *<Number>*

▶ Kind

A letter indicates the kind of diagnostics

S: System diagnostics

P: Process diagnostics, generated by Pilz

C: Process diagnostics, generated by the user

▶ Class

A decimal figure of a maximum of 4 digits indicates the class. The meaning of class depends on the kind of diagnostics.

– System diagnostics

The class defines the kind of error, e.g. device error, user program

Classes in system diagnostics:

100 ... 110 Device error

120 Error caused by a device project

121 Error caused by the naming data

122 Error caused by a boot command

123 Error caused by firmware update

200 ... 220 Error in the hardware registry/hardware configuration

300 Error in the FS or ST resource system section

303 Licensing is missing

310 Error in the VISU resource system section

400 Periphery error

410 Faulty supply voltage

600 ... 602 Error in the FS or ST SafetyNET p RTFN system section

700 ... 710 Error in the FS or ST module bus system section

801 Internal error in PAS4000

908 Ethernet error

910, 911 Error in the IP connections system section

912, 913 Error in the PROFIBUS-DP Slave system section

914, 915 Error in the EtherNet/IP system section

916, 917 Error in the PROFIsafe system section

9000 Operational entries in the diagnostic log

9001 Self test entries in the diagnostic log

9900 Debugging entries in the diagnostic log

- Process diagnostics, generated by Pilz

The class indicates the Pilz block to which the diagnostic message belongs. The class refers to the type of block, not to an instance.

- Process diagnostics, generated by the user

The user can decide what the class means. We recommend that the class is used along similar lines to that of the Pilz process diagnostics.

- ▶ Number

Unique number of a diagnostic message/diagnostic item within a class. The number is a 4-digit decimal figure.

13.2 Diagnostic log

The diagnostic log records when a diagnostic message arrived or when it was cleared. The diagnostic log also contains information on any changes in the operating state, e.g. if a device has been restarted, as well as information for Pilz technical support.

Each device has its own device diagnostic log. The [log entries](#) for all the devices in a project are combined within the project diagnostic log.

- ▶ Device diagnostic log
 - Generated by the device's firmware
 - Contains a device's log entries (raw data only, no texts)
 - Memory for at least 256 log entries; one memory area is reserved for system, process and self test entries and the other memory area for operational entries (see Log entries, [Page 486](#))
 - If the maximum number of log entries has been reached, the oldest log entry is deleted; this is the case respectively within the memory area for system, process and self test entries and the memory area for operational entries
 - Cannot be displayed; used as the basis for the project diagnostic log
- ▶ Project diagnostic log
 - Generated by the [Diagnostic Server](#)
 - Contains plain text log entries for all the devices in a project
 - Memory for at least 4096 log entries; one memory area is reserved for system, process and self test entries and the other memory area for operational entries (see (see Log entries, [Page 486](#)))
 - If the maximum number of log entries has been reached, the oldest log entry is deleted; this is the case respectively within the memory area for system, process and self test entries and the memory area for operational entries (This is why a project diagnostic log may contain very old operational entries while at the same time containing new system, process and self test entries or vice versa.)
 - Always up-to-date as it is continuously synchronised with the device diagnostic logs
 - Should the Diagnostic Server lose the connection to a device, all of the device's previous log entries are retained in the project diagnostic log. There will also be a log entry showing the loss of the connection.
 - The diagnostic log in PAS4000 may vary slightly from the diagnostic log on a display (e.g. PSS 4000 Diag Control), because it also contains entries for online actions, for example.

13.2.1 Log entries

The following log entries are available:

- ▶ System entries
Information about system messages changing state, i.e. when a system message arrived or when it was cleared. System entries are part of the system diagnostics.
- ▶ Process entries
Information about process messages changing state, i.e. when a process message arrived or when it was cleared. Process entries are part of the process diagnostics.
- ▶ Operational entries
Information about changes in the operating condition (e.g. when a device was restarted) or any other changes relevant to the operation (e.g. executing a project download). Operational entries are part of the system diagnostics.
- ▶ Self test entries
Information about a fault occurring during the self test in the "Booting base system" operating status, which has meant that the device could not be started.
- ▶ Debugging entries
Information for Pilz technical support. Debugging entries are part of the system diagnostics and are only visible when enabled by technical support.

During operation, log entries are entered in the diagnostic log.

With system entries and process entries, the log entry matches the diagnostic message, except that some information is left out, such as the remedy for example. The time stamp means something different and the change in state is displayed rather than the state.

Components of a log entry (operational entries and debugging entries are not included in the list):

- ▶ Time stamp
Time at which the state of the diagnostic message changed (see State of diagnostic messages, [Page 474](#))
- ▶ Change of state
Symbol for change of state (see State of diagnostic messages, [Page 474](#))
<+> for "arrived"
<-> for "cleared"
or
<#> for operational entries, self test entries and debugging entries
- ▶ Diagnostic identifier
Diagnostic identifier for the diagnostic message (see Diagnostic identifier, [Page 483](#))
- ▶ Severity and priority
Importance of the diagnostic message(see Severity and priority, [Page 475](#))
- ▶ Scope
Approximate description of the kind of event(see Scopes, [Page 477](#))
- ▶ Description
Description of what has happened

▶ Location information

Information about the location at which the event occurred (see Location information, [Page 478](#)).

The diagnostic log's location information contains the following details:

- Device
- Equip-ID
- Address
- Location description

13.3 Diagnostic server

A Diagnostic Server is required if you wish to display a project's diagnostic data.

The Diagnostic Server forwards the project diagnostic list and log to the diagnostic clients (e.g. PSS 4000 OPC Server). The diagnostic clients then provide the information to the displays (e.g. PSS 4000 Diag Control).

The Diagnostic Server gathers the raw data in the device diagnostic lists and device diagnostic logs of every device in the project and uses it to generate the project diagnostic list and project diagnostic log in plain text. In other words, the Diagnostic Server takes the references to message texts and location information in the device diagnostic lists and device diagnostic logs and replaces them with legible text.

The Diagnostic Server always needs the current project data: user program, hardware configuration, resource assignment and I/O mapping.

If the current project data is unavailable to the Diagnostic Server, references to message texts and location information cannot be replaced correctly. In this case they will be replaced with the wrong message texts/location information or by "<???" if no text is available. This can also happen if the Diagnostic Server has the current project data, but a device within the project is still operating with an older version of the project.

▶ Show on display

If the project diagnostic list and log are to be shown on a display (e.g. PSS 4000 Diag Control), a device within the project must provide the Diagnostic Server system section. If the project contains an OPC Server (PSS 4000 OPC Server), the PSS 4000 device on which the OPC Server is installed will provide the Diagnostic Server. A project may only contain one Diagnostic Server.

The texts displayed for system diagnostics and process diagnostics are those which were available when the project was built in PAS4000 and which were transferred when the project was downloaded.

▶ Display in PAS4000

PAS4000 has its own Diagnostic Server.

The texts displayed for system diagnostics are the diagnostic texts supplied with PAS4000. The texts displayed for process diagnostics are the diagnostic texts which were available when the project was built.



Information

The diagnostic list and diagnostic log are displayed correctly if the current project has been downloaded to the Diagnostic Server and to all the devices in the project.

If the computer on the OPC Server is used as the diagnostic server, you will also need to download the current project to the computer on the OPC Server (this will be done automatically if the OPC Server is connected to SafetyNET p during download).

We also recommend that the version of the PSS 4000 firmware on the device containing the Diagnostic Server should be equal to or higher than the version in the project.

If the texts are to be displayed in PAS4000, you will need to activate the current project in PAS4000. Always use the latest version of PAS4000.

After an extended period of operation, log entries which are no longer available on the project devices may still be stored in the project diagnostic log on the Diagnostic Server. These old log entries will be deleted when the device with the Diagnostic Server has a warm reset or when the PAS4000 is cold started.

14 Appendix

14.1 Factory default settings for port numbers for various services

Port numbers have standard default settings for various services.

Service		Default setting	Meaning
SafetyNET p		40000	SafetyNET p protocol
DNS		53	Domain Name Service Protocol
OPC via DCOM	Connection Setup (Server)	135	Distributed Component Object Model Protocol
	Communication (Server)	Dynamic	
DHCP		67, 68	Dynamic Host Configuration Protocol
Modbus/TCP		502	Modbus over TCP
SNTP		123	Simple Network Time Protocol

14.2 Factory default settings for the Ethernet parameters

The Auto IP mechanism is activated on PSS 4000-exclusive devices in delivery status or after an original reset. The Ethernet parameters are set as follows:

Parameter	Default setting		Meaning
IP address	In the network the Auto IP mechanism is active for exactly one device.	169.254.1.1	If in the network the Auto IP mechanism is active for exactly one device, the Auto IP mechanism assigns the stated IP address for the device if it is still free.
	In the network the Auto IP mechanism is active for several devices.	169.254.1.1 to 169.254.1.254	If in the network the Auto IP mechanism of several devices is active, the Auto IP mechanism assigns a free IP address from the stated address range.
Subnet mask	255.255.0.0		The Auto IP mechanism assigns the stated Subnet mask.
Address of Standard Gateway	Deactivated (0.0.0.0)		Configurable as an option when SafetyNET p is divided into subnetworks through Gateways and, from the communication partner's perspective, the product is behind a Gateway.
Address of preferred DNS Server	Deactivated (0.0.0.0)		Configurable as an option when the Domain Name Service is used.
Address of alternative DNS Server			

Auto IP mechanism

Technology for configuration-free and/or automatic assignment of a free IP address from the address range 169.254.0.0/16

Autocrossing

Automatic crossing of the send and receive lines on twisted pair interfaces; devices that support this function can be connected via a 1:1 wired cable (patch cable) instead of a crossed cable (crossover cable).

Autosensing

Automatic bit rate detection (10 MBit/s or 100 MBit/s)

Basic diagnostic item (BDI)

Diagnostics: Part of the user program in which a process state that is to be monitored for diagnostic purposes is defined. The basic diagnostic item also contains information and texts for one or more diagnostic messages and diagnostic log entries.

BDI

Abbreviation for Basic Diagnostic Item

Bit module

Electronic/compact module whose data requires one bit each in the process image. Example: Digital input/output modules

Block

Umbrella term for Multi blocks and POU's

Build

The code for the individual devices (device projects) is built from the project.

Byte module

Electronic module whose data requires different data widths in the process image (bit, byte and/or word). Example: Counter modules

Certification

Procedure by which a product's compliance with the applicable regulations and standards is tested by a notified body; proven compliance is confirmed through the issue of a certificate.

Certification body

Notified body in accordance with Article 14 of the Directive 2006/42/EC; otherwise: Accredited institution, often a government body, which examines and confirms a product's compliance with applicable regulations and standards.

Check sum

Sum, which is formed via an object, e.g. a project or program organisation unit (POU). This can be used to detect whether the object has been modified.

Consistency range

Amount of consistent data of a data source (e.g. consistency range "FS-I data from a FS module bus")

Control variables

The value of variables is set once to a specified value. It can then be changed by the user program.

Crossover cable

Cross wired cables for wiring Ethernet components and end devices such as PLC or network components (hubs, switches)

Data consistency

Keeping together data that is recorded at a certain time and is considered current and related during a defined period.

Debugging

Troubleshooting and fault recovery during programming or commissioning.

Device Editor

Software tool used to define the settings for a device and call up information from the device

Device project

Part of the project that is downloaded to a device.

Diagnostic item

Diagnostics: Generic term for basic diagnostic item and summary diagnostic item

Diagnostic list

Diagnostics: List of a project's current diagnostic messages

Diagnostic log

Diagnostics: Log that shows when a project's diagnostic messages arrived and when they were cleared. It also contains operation and debugging entries.

Download

Transfer of data to a device, e.g. from a PC to a device

Enable principle

Configuration that enables the ST section to have write access to the FS section; FS outputs can be switched via the ST section. Comment: PSS 4000: By using the enable principle, FS data sinks (e.g. FS hardware outputs) can be switched via ST data sources (e.g. ST hardware inputs).

Ethernet infrastructure

All the hardware (e.g. cables, plug-in connections, adapters, switches, routers, hubs) and software (e.g. network services and their configuration) required for data transfer in an Ethernet network.

Failsafe (FS)

Safety-related, failsafe; a failsafe system switches to a safe condition in the event of an error.

FB

Abbreviation for Function Block

Firmware

Software created by the manufacturer to run on a device and perform device-specific functions.

Firmware update

Updating the device firmware

Force variables

The value of variables is permanently assigned and cannot be changed by the user program.

FS

Abbreviation for FailSafe

FS resource

Execution unit for safety-related control code

Function block (FB)

IEC 61131-3: Programmable controller programming language element consisting of: 1) the definition of a data structure partitioned into input, output, and internal variables; and 2) a set of operations to be performed upon the elements of the data structure when an instance of the function block type is invoked.

FW

Abbreviation for FirmWare

Hardware input

Physical device input

Hardware output

Physical device output

Hub

Hardware of the Ethernet infrastructure, for star-shape connection of several Ethernet subscribers, data packets received at one Port, will be transmitted to all other Ports.

HW

Abbreviation for HardWare

I/O mapping

Mapping of a PI variable to I/O data from the module bus, I/O data from the external communication or another PI variable

IL

Abbreviation for Instruction List

Instruction list (IL)

Text-based, machine-oriented programming language; all operations are shown in text format

I-PI variable

Variable in the process image, which can be mapped to I data from the module bus, I data from external communication or an O-PI variable.

O-PI variable

Variable in the process image, which can be mapped to O data from the module bus, O data from external communication or an I-PI variable.

Patch cable

1:1-wired cable with a max. length of 5 m; it is primarily used to connect Ethernet components within an enclosure (control cabinet etc.).

Person, competent

A person who, through training, experience and current professional activity, has acquired the necessary knowledge and authorisation to test, assess and handle devices, systems, plant and machinery in accordance with the general standards and guidelines for safety technology.

PI variable

Variable in the process image which can be mapped to I/O data from the module bus, I/O data from the external communication or another PI variable.

POU

Abbreviation for Program Organisation Unit

PRG

Abbreviation for PRoGram

Process value

Actual value of process data; in contrast to the substitute value

Program (PRG)

IEC 61131-1: "Logical assembly of all the programming language elements and constructs necessary for the intended signal processing required for the control of a machine or process by a PLC-system."

Program organisation unit (POU)

In accordance with IEC 61131-3: Function, function block or program

Project

A project contains all the data needed to control and monitor a system (plant, machine).
Project = User program + hardware configuration

Quality of process data

Information on whether process data relates to the "process value" or a "substitute" value.

Real-time system

System in which the validity of a calculated result depends not only on logical accuracy but also on the time at which the result is provided. A response that is too early or too late is an incorrect response.

Resource

Unit that executes code, e.g. FS resource, ST resource, VISU resource, VISU resource

Resource assignment

Assignment of program type program organisation units or basic blocks to the resources

Router

Hardware of the Ethernet infrastructure, for connecting different networks, based on the Net-ID (= section of the IP address) the router decides which data packets are to be transmitted to a different network.

Safe condition

Condition in which power is removed from the safety circuits.

Sealed

Property of project elements (e.g. program organisation units, components). Project element is sealed, certified if necessary, and may not be modified and displayed.

ST

Abbreviation for STandard

ST resource

Execution unit for non-safety-related control code

Standard (ST)

Not safety-related, in contrast to failsafe

Standard bus system

Non-safety-related bus system, e.g. DeviceNet or PROFIBUS, in contrast to SafetyBUS p or SafetyNET p from Pilz (see also "Fieldbus")

Substitute value

If the process data cannot be determined due to an error, it is replaced by substitute values.

SW

Abbreviation for SoftWare

Switch

Hardware of the Ethernet infrastructure, for star-shape connection of several Ethernet subscribers, data packets received at one Port, will be analysed and only transmitted to the Port where the subscriber is registered. Broadcast messages are forwarded.

Task

Execution control element providing for periodic or triggered execution of a group of associated IEC 61131 programs and/or basic blocks (see IEC 61131).

Test pulse

The test pulse is wired to an input in order to detect wiring errors in that input, e.g. open circuit, short circuit to 24 V DC or 0 V or shorts between two inputs.

Timeliness

Occurrence of an event within a defined time window; events that occur earlier or later are invalid.

Variable list

List containing the variables for which control, force or watch variables is to be executed.

VISU resource

Execution unit for visualisation



Pilz GmbH & Co. KG
Felix-Wankel-Straße 2
73760 Ostfildern, Germany
Telephone: +49 711 3409-0
Telefax: +49 711 3409-133
E-Mail: pilz.gmbh@pilz.de
Internet: www.pilz.com



...
In many countries we are represented by our subsidiaries and sales partners.

Please refer to our homepage for further details or contact our headquarters.

► Technical support

+49 711 3409-444
support@pilz.com

pilz