



► PVIS OPC Server UA

PILZ
THE SPIRIT OF SAFETY

System Description-1003917-EN-08

- User software



This document is the original document.

Where unavoidable, for reasons of readability, the masculine form has been selected when formulating this document. We do assure you that all persons are regarded without discrimination and on an equal basis.

All rights to this documentation are reserved by Pilz GmbH & Co. KG. Copies may be made for the user's 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.



SD means Secure Digital

1	Introduction	5
1.1	Validity of the documentation	5
1.2	Retaining the documentation	5
1.3	Definition of symbols	6
1.4	Third-party manufacturer licence information	7
2	Diagnostics and visualisation with PVIS	8
2.1	Aim of diagnostics and visualisation	8
2.2	Overview of the procedure	9
2.3	Supported hardware and software	11
3	PVIS diagnostic configuration	12
3.1	Introduction	12
3.2	Event messages and remedy	14
3.2.1	Priorities	16
3.2.2	Scopes	17
3.2.3	Responsibilities	18
3.2.4	Event list	19
3.3	Device's diagnostic configuration	20
3.4	Diagnostic configuration for safety devices	21
3.4.1	Copies	21
3.5	Diagnostic configuration of operands	22
3.5.1	Bit diagnostic type	22
4	PVIS OPC Server UA	23
4.1	Function of the PVIS OPC Server UA	23
4.2	OPC data type assignment	25
4.3	Installing the PVIS OPC Server UA	28
4.3.1	System requirements	28
4.3.2	Installation files	28
4.3.3	Installation on a PC (manual copying)	29
4.3.4	Installation on a PMI v8 or PC with PASvisu	31
4.4	Information security certificates	32
4.4.1	Directory structure of the certificate store	33
4.4.2	Procedure for authentication	34
4.4.2.1	Authentication in the case of local installation	34
4.4.2.2	Authentication in the case of remote installation	34
4.4.2.3	Authentication in the case of OPC UA Clients of a third-party manufacturer	38
4.4.3	Generating and using custom certificates	39
4.4.4	Invalidating a certificate	41
4.4.5	Verifying a certificate	41
4.5	Event view	42
4.6	Commissioning the PVIS OPC Server UA	42
4.7	Transfer PVIS OPC project manually	43
4.8	Language management	43
5	Namespace	44
5.1	Basics	44

5.1.1	Definition of symbols	45
5.2	Structure of the namespace	46
5.2.1	OPCProjectName.....	46
5.2.2	DeviceProjectName	47
5.2.3	DeviceProjectName/SummaryInformation	49
5.2.4	DeviceProjectName/OverviewInformation	53
5.2.4.1	CentralDevices.....	54
5.2.4.2	OverviewInformation	55
5.2.4.3	SafetyDevices	55
5.2.4.4	Operands	57
5.2.5	DeviceProjectName/DetailInformation	58
5.2.5.1	Event.....	59
5.2.5.2	Actions	61
5.2.5.3	PNOZmulti/PNOZmulti Mini/PNOZmulti 2: SystemInfo.....	63
5.2.5.4	PMC: SystemInfo	66
5.2.6	DeviceProjectName/Generic.....	69
5.2.6.1	PNOZmulti/PNOZmulti Mini/PNOZmulti 2: Generic	70
5.2.7	DeviceProjectName/EventLog	79
6	Examples of application	82
6.1	Diagnostics for a PNOZmulti	82
6.1.1	Create diagnostic configuration.....	82
6.1.2	Download the diagnostic configuration to the PNOZmulti.....	83

1 Introduction

This operating manual describes the PVIS expanded diagnostics. PVIS expanded diagnostics is a concept for visualising diagnostics information for PVIS-compatible controllers from Pilz.

The concept comprises

- ▶ PVIS-compatible controllers from Pilz
 - Configurable small controllers PNOZmulti
 - Motion control systems
- ▶ PNOZmulti Configurator

Software from Pilz, which is used to create a PNOZmulti project containing the diagnostic configurations and the OPC data of the PNOZmulti.

This operating manual does not cover operation of the PNOZmulti Configurator. Please refer to the online help for the PNOZmulti Configurator.
- ▶ PVIS OPC Server UA

Software from Pilz that

 - serves as a diagnostic device,
 - is installed on a PMI from Pilz or a PC,
 - imports the diagnostic data of the associated controllers
 - makes the imported diagnostic data available in the namespace for OPC UA Clients
- ▶ PASvisu Builder

Pilz software used to create projects for visualisation.

This operating manual does not cover operation of the PASvisu Builder. Please refer to the online help for the PASvisu Builder.

1.1 Validity of the documentation

This operating manual is valid for:

- ▶ PVIS OPC Server UA from Version 2.7.0

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).

1.2 Retaining the documentation

This documentation is intended for instruction and should be retained for future reference.

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.



CAUTION!

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.



NOTICE

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.

1.4 Third-party manufacturer licence information

This product includes Open Source software with various licenses.

Further information is available in the document "Third-party manufacturer license information PVIS OPC Server UA" (document number 1003919) at www.pilz.com.

The relevant CDDL/CPL/EPL/MPL licensed source codes can be requested via open-source@pilz.de.

Your request should include the following: (a) the firmware/software name, (b) the firmware/software version, (c) your name, (d) your company name (if applicable), (e) your reply address and (f) your E-mail address (if possible).

Pilz can charge a fee for the data medium and for sending. The download from the Internet is free of charge.

Portions copyright © by OPC Foundation, Inc. and licensed under the Reciprocal Community License (RCL).

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (<http://www.openssl.org/>). This product includes software written by Tim Hudson (tjh@cryptsoft.com).

Due to the use of commercial and Open Source components, using the device is subject to the following special restrictions:

The user is prohibited, without prior written permission from Pilz, to use the device in aircraft, in applications for planning, construction, production or delivery of aircraft and parts thereof, in nuclear power plant applications, for navigation, for the production of arms, in medical applications which may lead to personal injury, in applications that may cause water pollution, or in plants that are subject to a licensing or notification requirement.

2 **Diagnostics and visualisation with PVIS**

2.1 **Aim of diagnostics and visualisation**

PVIS is a concept for expanded diagnostics. What does PVIS expanded diagnostics intend to achieve?

- ▶ PVIS minimises machine downtimes by rectifying errors quickly and efficiently.

How is this achieved?

- ▶ Detailed plain text messages are available for each event.
- ▶ Location information as plain text or as an Equip ID (equipment identifier) shows where a problem lies.
- ▶ Priorities highlight the events which require an immediate reaction.
- ▶ Responsibilities tell you who can rectify the problem.
- ▶ The remedy provides a step-by-step solution.
- ▶ Only current messages are displayed.
- ▶ PVIS distinguishes between a first fault and subsequent faults. Only first faults are displayed.
- ▶ Users can modify and adapt default messages.
- ▶ The OPC Server "PVIS OPC Server UA" from Pilz provides a standardised interface to visualisation systems.

2.2 Overview of the procedure

Diagnostic configurations can be created for all PVIS-compatible controllers. This is achieved using the respective system software for the controller, e.g. with the PNOZmulti Configurator.

The diagnostic configuration contains event messages, which can be displayed when there are faults in or on the controller, when the operating state of the controller changes or when safety devices have a defined state, for example.

The OPC Server "PVIS OPC Server UA" from Pilz is used to display event messages on a visualisation system. The OPC Server is installed on a PC or PMI. The PNOZmulti Configurator from Pilz is used to create an OPC project. The download to the OPC Server is achieved via the visualisation system PASvisu from Pilz, or by manual copying.

The OPC Server establishes a connection to the controllers, imports the data and makes it available within the namespace. Event messages can be retrieved in the namespace, as well as status information and process data from the controllers (see figure: Diagnostics with the OPC Server "PVIS OPC Server UA from Pilz").

The visualisation system PASvisu from Pilz can be used to retrieve a controller's event messages from the OPC Server and display them in a visualisation software package.



NOTICE

The PVIS OPC Server UA can only be used with OPC UA Clients (e.g. PASvisu Client from the visualisation system PASvisu from Pilz). OPC Clients (classic) cannot access the PVIS OPC Server UA.

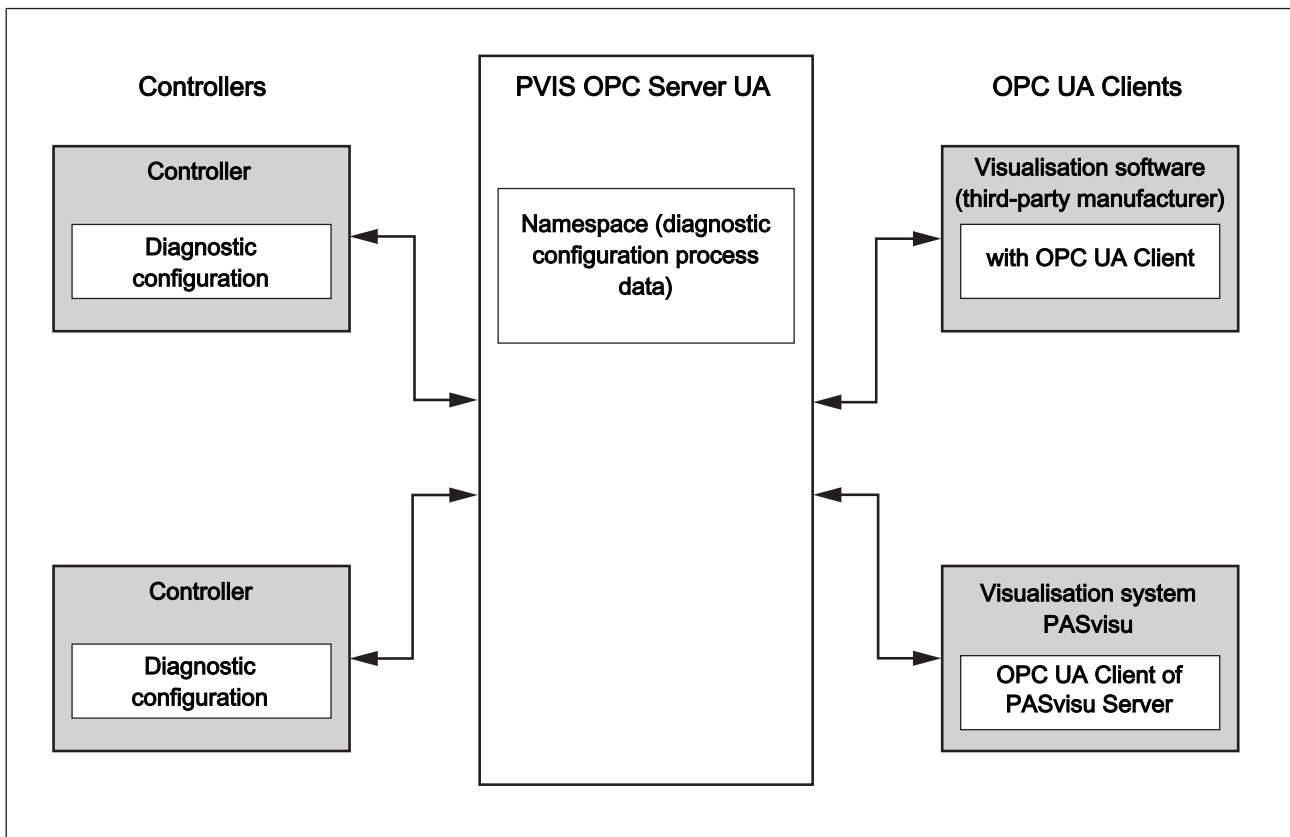


Fig.: Diagnostics with the OPC Server "PVIS OPC Server UA" from Pilz

2.3 Supported hardware and software

The functions of the PVIS expanded diagnostics are supported by the controllers, system software and display or visualisation devices as follows:

Diagnostics on the configurable small controllers PNOZmulti

- ▶ Base units from the following versions:
 - Configurable safe small controllers PNOZmulti Classic (previously PNOZmulti)
 - PNOZ m0p from Version 2.0
 - PNOZ m1p from Version 5.0
 - PNOZ m2p from Version 2.0
 - Configurable safe compact controllers PNOZmulti Mini
 - PNOZ mm0.1p from Version 1.0
 - PNOZ mm0.2p from Version 1.0
 - Configurable safe small controllers PNOZmulti 2
 - PNOZ m B0 from Version 1.0
 - PNOZ m B1 from Firmware Version 1.1.1
- ▶ PNOZmulti Configurator from Version 5.0.0
- ▶ PVIS OPC Server UA from Version 2.0.0
- ▶ PASvisu Server from Version 1.3.0
- ▶ PASvisu Builder from Version 1.3.0

3 PVIS diagnostic configuration

3.1 Introduction

The diagnostic configuration contains all the information required for expanded diagnostics. The diagnostic configuration enables event messages to be displayed in the case of:

- ▶ Faults in or on the controller
- ▶ Changes in the operating state of the controller
- ▶ Defined status conditions of safety devices
 - PNOZmulti: Elements
- ▶ PNOZmulti: Defined states of diagnostic points (inputs, outputs and connection points)

The controller's event messages can also be supplemented through additional information, which is helpful during diagnostics.

With expanded diagnostics, a diagnostic device (e.g. PVIS OPC Server UA) is connected to the controller. If an event occurs in or on the controller, an event telegram is sent to the diagnostic device. The event telegram is evaluated within the diagnostic device, i.e. a decision is made as to whether the corresponding event message should be displayed and entered in the event list. The event message contains a description of the event. A remedy can be displayed for each event message. The remedy describes how to react to the event. The remedy contains a maximum of 8 steps, the so-called "Actions".

The diagnostic configuration consists of three independent parts:

- ▶ Device's diagnostic configuration
 - Contains the event messages which are output if a fault occurs in or on the controller or if there is a change in the operating state of the controller
- ▶ Diagnostic configuration for safety devices
 - Contains the event messages that are triggered when safety devices have a defined state
- ▶ Diagnostic configuration of operands
 - PNOZmulti: Contains the event messages that are output when inputs, outputs and connection points have a defined state

The diagnostic configuration is project-related, i.e. a separate diagnostic configuration is created for each PNOZmulti project.

The diagnostic configuration is downloaded to the controller and to the diagnostic device. The whole diagnostic configuration is not downloaded, just the data that the controller and diagnostic device require from the diagnostic configuration. The device's diagnostic configuration is not downloaded to the controller at all, because it is part of the controller's operating system.

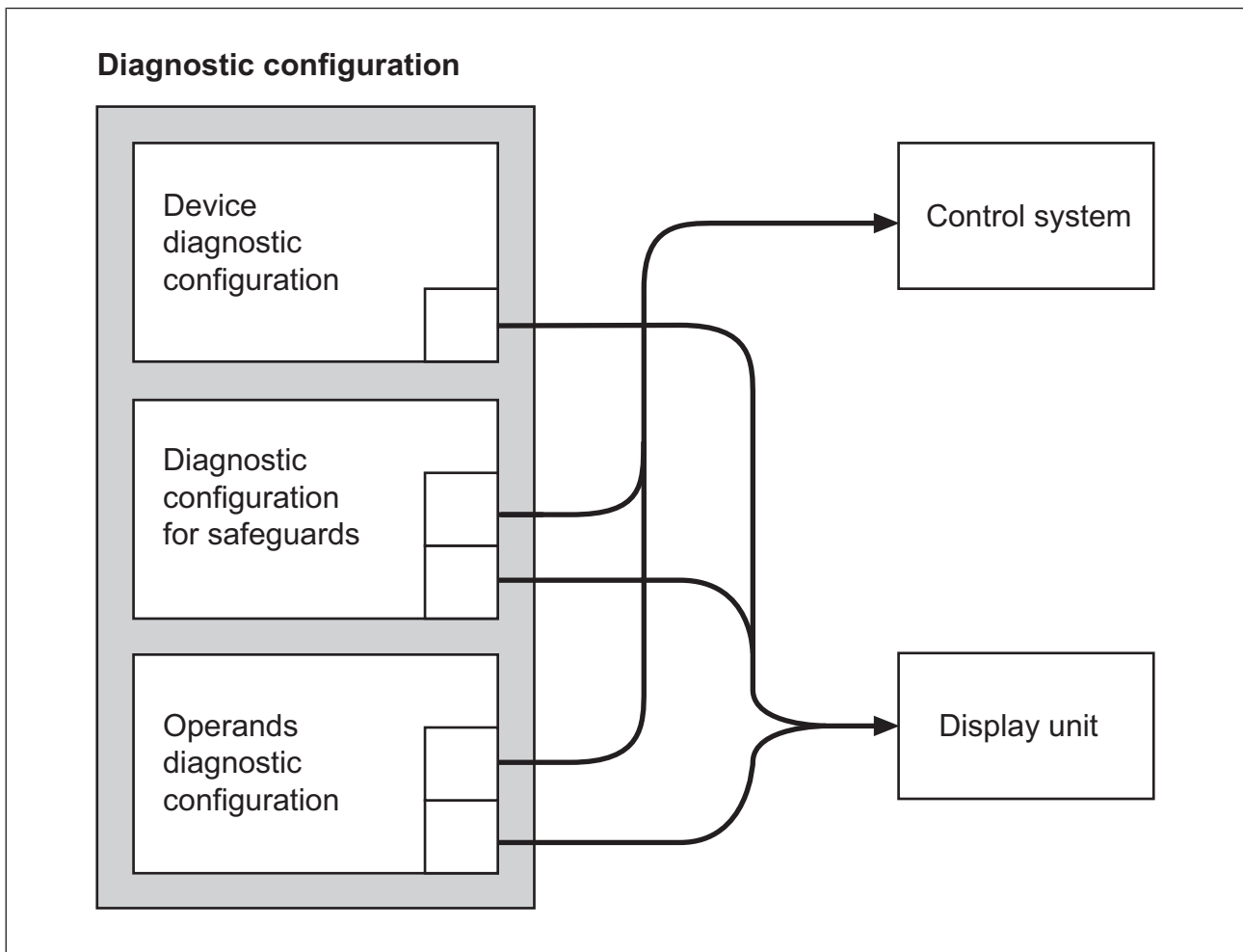


Fig.: Download diagnostic configuration

3.2 Event messages and remedy

If an event occurs in or on the diagnosed controller, the corresponding event message is displayed. The event message contains a description of the event. A remedy can be displayed for each event message. The remedy describes how to react to the event. The remedy contains a maximum of 8 steps, the so-called "Actions".

Event message components

- ▶ **Priority**
Importance of the event message
- ▶ **Scope**
Approximate classification of the event message
- ▶ **Description of event**
Description of what happened
- ▶ **Location information**
Information on where the event occurred.
The location information may consist of the following details:

- **Device**
Name of the device/module, details of the faulty object, type of safety device (e.g. safety gate) or information on the operand (e.g. reset input)
- **Equip-ID**
Equipment identifier of the device/module or symbolic name of the safety device or operand
- **Address**
More precise information on where the fault occurred on a device, e.g. information on the terminals
- **Location description**
Text description of the location, e.g. specification of the control cabinet in which the device is located.

Action components

- ▶ **Scope**
Approximate classification of the action
- ▶ **Responsibility**
Information on who should undertake the action.
- ▶ **Description of action**
Description of what has to be done.
- ▶ **Location information**
Information on where the action is to be undertaken.
The location information may consist of the following details:

- **Device**
Name of the device/module, information on a program section, type of safety device (e.g. safety gate) or information on an operand (e.g. reset input)
- **Equip-ID**
Equipment identifier of the device/module or symbolic name of the safety device or operand

- Address
More precise information on where the action is required on a device, e.g. information on the terminals
- Location description
Text description of the location, e.g. specification of the control cabinet in which the device is located.

3.2.1 Priorities

Each event message is assigned a priority, which describes the importance of an event message. 15 priorities can be defined. Of these, priority 3 “Error”, priority 8 “Warning” and priority 13 “Status information” are pre-defined by Pilz.

You can change the name of the pre-defined priorities but you cannot delete them.

Priority 1 is the highest priority and priority 15 the lowest.

The 15 priorities are divided into three priority classes:

► **Error**

This priority class contains event messages to which the user must react immediately.

These event messages provide information on the status of a technical device on which functions have failed or are compromised.

The "Error" priority class includes the pre-defined priority, "Error".

► **Warning**

This priority class contains event messages which demand the user's attention. These event messages warn the user about a potential or imminent critical machine status.

The "Warning" priority class includes the pre-defined priority, "Warning".

► **Status information**

This priority class contains event messages which describe the current status of the process and/or machine. These event messages are provided for information; the production cycle is neither impeded nor compromised.

The "Status information" priority class includes the pre-defined priority, "Status information".

Priority class	Priority	Name of the priority
Error	1	
	2	
	3	Error
	4	
	5	
Warning	6	
	7	
	8	Warning
	9	
	10	
Status information	11	
	12	
	13	Status information
	14	
	15	

3.2.2 Scopes

Each event message and each action can be assigned to a scope. The scope roughly describes the kind of event/action.

15 scopes can be defined. Pilz has predefined three scopes, but these can be changed.

Pre-defined scopes:

- ▶ Plant/device

Affects the mechanics, electrics, electronics or software.

Examples: Defective device, periphery error (e.g. wiring, actuator/sensor or supply voltage is faulty), project configuration error (e.g. PNOZmulti configuration or user program is faulty), bus error (e.g. error in standard fieldbus)

- ▶ Process

Affects sequence or operation.

Examples: Sequence error (cylinder fails to reach an end position), safety device was triggered (e.g. E-STOP operated)

- ▶ Communication

Data transfer between the controller and diagnostic device is disrupted

- ▶ ---

No details of the scope are possible or necessary.

3.2.3 Responsibilities

An area of responsibility can be assigned to each action. The responsibility stands for the person or group who is supposed to perform an action.

Some responsibilities are pre-defined by Pilz and a maximum of 15 responsibilities can be defined by the user. The user can change the name of the pre-defined responsibilities, but cannot delete them.

Pre-defined responsibilities:

- ▶ **Safety device**
This action involves the release, actuation or operation of a safety device.
Responsible person: plant operator for example
- ▶ **Periphery**
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: e.g. maintenance engineer
- ▶ **Device malfunction**
This action refers directly to an error within the device, e.g. check device, swap device.
Responsible person: e.g. maintenance engineer
- ▶ **Project configuration**
This action can only be performed using the programming device or with the PNOZmulti Configurator (PNOZmulti), or requires some programming knowledge, e.g. check or adapt configuration, check or adapt programming.
Responsible person: maintenance engineer, programmer, project engineer
- ▶ **Bus**
This action can be performed where there is technical understanding of the fieldbus, e.g. check bus load, check cable lengths.
Responsible person: maintenance engineer, bus specialist
- ▶ **Sequence**
The process sequence was wrong and must be corrected.
Responsible person: operator, maintenance engineer, programmer, project engineer, for example
- ▶ **---**
No details of responsibility are possible or necessary.

3.2.4 Event list

The event list is shown on the diagnostic device. For each event, a check is made to examine whether the event message must be displayed and entered in the event list, depending on its priority. For example, if it relates to an event that has occurred as a consequence of a previous event, the event telegram will be ignored.

If there are several event messages with equal priority, the time stamp will decide the order in which the event messages are entered in the event list. The most recent event message will be rated higher.

A check is also made to assess whether the user requires further information about an event.

The event messages are also deleted from the event list as soon as the cause of the error has been rectified.

The event list is substantially different from the PNOZmulti error stack. In the error stack, all events that occur are simply entered in turn.

3.3 Device's diagnostic configuration

The device's diagnostic configuration contains the event messages which are output if an error occurs in or on the diagnosed control system or if there is a change in the operating status of the diagnosed control system.

The device's diagnostic configuration is supplied by Pilz in full and cannot be modified by the user.

3.4 Diagnostic configuration for safety devices

The diagnostic configuration for safety devices contains the event messages that are triggered when safety devices have a defined status.

Safety devices are:

- ▶ PNOZmulti:
 - Elements
 - Input devices that are monitored by function elements

Pilz has a pre-defined "Safety Device" diagnostic type for each standard safety device. The diagnostic type contains an event message, including remedy, for every potential safety device status, i.e. for each event. Users cannot modify the Pilz pre-defined diagnostic types for safety devices. However, copies may be created for you to adapt individually.

3.4.1 Copies

Copies of a "Safety device" diagnostic type enable the event message and remedies to be adapted, should a function element be used for an alternative purpose, e.g.:

- ▶ PNOZmulti: The function element "E-STOP" is not to be used to monitor an E-STOP pushbutton but to monitor an end switch.

A copy is valid for the same function element as the original.

Copies may be made of each Pilz pre-defined "Safety device" diagnostic type. More copies can also be created from copies. A copy is given a default name, which may be changed.

Events may not be deleted from the copy, but they may be edited. The priority and the event's scope and description may be changed.

The remedy can be changed as required: existing actions may be edited, actions may be deleted and new actions may be added.

3.5 Diagnostic configuration of operands

Operand's diagnostic configuration

- ▶ PNOZmulti: Contains the event messages that are output when inputs, outputs and connection points have a defined status.

3.5.1 Bit diagnostic type

Bit operands are:

- ▶ PNOZmulti
 - Inputs and outputs
 - Inputs and outputs of elements in the PNOZmulti Configurator

The "Bit" diagnostic type can be configured for bit operands. A new "Bit" diagnostic type is given a default name, which you can change.

The "Bit" diagnostic type contains a single event message, including remedies. The event message is displayed during diagnostics when the operand assumes the state "1". On a PNOZmulti it is also possible to evaluate by a rising or falling edge.

4 PVIS OPC Server UA

4.1 Function of the PVIS OPC Server UA

OPC stands for Openness, Productivity, Collaboration and UA for Unified Architecture. OPC UA complements the OPC industry standard now frequently referred to as Classic OPC with substantive properties such as platform independence, scalability, high availability, internet capability and others.

OPC UA provides a standardised software interface which enables applications from a wide variety of manufacturers to exchange data. The OPC Foundation is responsible for maintaining and propagating the standard.

The PVIS OPC Server UA is software from Pilz that serves as diagnostic software and provides process data. It runs on a PMI or PC. The PVIS OPC Server UA reads the data from the connected controllers and displays it in the namespace. The OPC UA Client retrieves it from there.

All the settings for the PVIS OPC Server UA are made in the PNOZmulti Configurator and are transferred to the PVIS OPC Server UA as an OPC project via PASvisu. The OPC project can also be transferred to the PVIS OPC Server UA manually (see [Transfer PVIS OPC project manually](#)  43).

With the PVIS OPC Server UA it is possible to download diagnostic data, the process data display and additional information to any visualisation software or to other applications.

The key features of the PVIS OPC Server UA are:

- ▶ TCP support
- ▶ Address of the PVIS OPC Server UA in the network:
opc.tcp://<Server>:<Port>
 - <Server>: End point URL of the PVIS OPC Server UA
 - <Port>: Port number of the PVIS OPC Server UA (default setting: 4852)
- ▶ Support for OPC UA services (part 4):
 - **Discovery**
(Discovery Server functions)
 - **SecureChannel**
Supported service sets:
 - OpenSecureChannel
 - CloseSecureChannel
 - **Session**
Supported service sets:
 - CreateSession
 - ActivateSession
 - CloseSession
 - Cancel
 - **View**
Supported service sets:

- Browse
- BrowseNext
- TranslateBrowsePathsToNodeIds

– **Attribute**

Supported service sets:

- Read
- Write

– **MonitoredItems/Subscriptions**

Supported service sets for MonitoredItems:

- CreateMonitoredItems
- ModifyMonitoredItems
- SetMonitoringMode
- SetTriggering
- DeleteMonitoredItems

Supported service sets for Subscription:

- CreateSubscription
- ModifySubscription
- SetPublishingMode
- Publish
- Republish
- TransferSubscription
- DeleteSubscription

- ▶ Data Access (DA) support
- ▶ Support for generic access to process data in the namespace with pre-defined syntax

4.2 OPC data type assignment

Various data types are defined through the OPC UA standard. The table below contains the data types supported via the PVIS OPC Server UA and their assignment to the data types of PNOZmulti, PNOZmulti Mini, PNOZmulti 2 and the diagnostics in the namespace:

OPC UA data type	Meaning		
Boolean	Bit value/Boolean Data width: 1 bit Value range: TRUE/FALSE		
	Assignment:		
	PNOZmulti, PNOZmulti Mini PNOZmulti 2	▶ Input bit, output bit ▶ Virtual I/Os	
	Diagnostics	Boolean (see namespace: "SummaryInformation" and "OverviewInformation")	
Byte	Positive integer value (Integer) Data width: 8 Bit (1 Byte) Value range: 0 ... 255		
	Assignment		
	PNOZmulti, PNOZmulti Mini	- - -	
	PNOZmulti 2	PNOZmulti mB0	▶ Namespace: .../Integrity/ DiagnosticConfigPNOZmulti: – Status ▶ Namespace: .../Generic/ .../Service-Advanced-data: – LED-IO – LED status
		PNOZmulti mB1	▶ Namespace: .../Integrity/ DiagnosticConfigPNOZmulti: – Status ▶ Namespace: .../Generic/ .../Service-Advanced-data: – LED-IO – LED status ▶ Namespace: .../Generic/ ConnectedDevices/ PITreader: – Status – Permission

OPC UA data type	Meaning		
Double	Floating point number with double precision (in accordance with IEEE) Data width: 64 Bit (8 Bytes) Value range:		
	Assignment:		
	PNOZmulti, PNOZmulti Mini PNOZmulti 2	- - -	
Float	Floating point number with single precision (as per IEEE) Data width: 32 Bit (4 Bytes)		
	Assignment:		
	PNOZmulti, PNOZmulti Mini	- - -	
	PNOZmulti 2	PNOZmulti mB0	Input value analogue module
		PNOZmulti mB1	Input value analogue module
Int16	Integer value (Integer) Data width: 16 Bit (1 word) Value range: -32768 ... 32767		
	Assignment:		
	PNOZmulti, PNOZmulti Mini	Input word and output word for analogue modules	
	PNOZmulti 2	- - -	
Int32	Integer value (Integer) Data width: 32 Bit (2 words) Value range: – 2147483648 ... 2147483647		
	Assignment:		
	PNOZmulti, PNOZmulti Mini PNOZmulti 2	- - -	
Int64	Integer value (Integer) Data width: 64 Bit (4 words) Value range: – 9223372036854775808 ... 9223372036854775807		
	Assignment:		
	PNOZmulti, PNOZmulti Mini PNOZmulti 2	- - -	
LocalizedText	Text that can have a local identifier as an option For coding see "String"		
	Assignment: Character string for diagnostics (see namespace: "DetailInformation" and "SummaryInformation")		

OPC UA data type	Meaning		
SByte	Integer value (Integer) Data width: 8 Bit (1 Byte) Value range: -128 ... 127		
	Assignment:		
	PNOZmulti, PNOZmulti Mini PNOZmulti 2	- - -	
String	Character string in Unicode format Coding: 2 bytes (octets) per character		
	Assignment: Character string for diagnostics (see namespace: "SystemInfo")		
UInt16	Positive integer value (Integer) Data width: 16 Bit (1 word) Value range: 0 ... 65535		
	Assignment:		
	PNOZmulti, PNOZmulti Mini PNOZmulti 2	- - -	
	Diagnostics	See namespace: "OverviewInformation" -> All "Status" nodes	
UInt32	Positive integer value (Integer) Data width: 32 Bit (2 words) Value range: 0 ... 4294967295		
	Assignment:		
	PNOZmulti, PNOZmulti Mini	- - -	
	PNOZmulti 2	PNOZmulti mB0	► Input value: Speed, position
		PNOZmulti mB1	► Input value: Speed, position ► PITreader: KeyOrderNo, KeySerialNo
UInt64	Positive integer value (Integer) Data width: 64 Bit (4 words) Value range: 0 ... 18446744073709551615		
	Assignment:		
	PNOZmulti, PNOZmulti Mini	- - -	
	PNOZmulti 2	PNOZmulti mB0	- - -
		PNOZmulti mB1	► PITreader: SecurityID

4.3 Installing the PVIS OPC Server UA

4.3.1 System requirements

The PVIS OPC Server UA can be installed on the following diagnostic devices:

- ▶ PC with Microsoft Windows 10 operating system (64 Bit)
- ▶ PC with Microsoft Windows 11 operating system (64 Bit)
- ▶ Operator terminals PMI v8 from Pilz with the Microsoft Windows 10 IoT operating system (64 Bit)

4.3.2 Installation files

Installation file for PC

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 installation file is available in compressed form (Zip file). The installation is available once it is unzipped.

- ▶ Zip file

PVIS_OPC_Server_UA-PC_Installer_<Version>_<Document-ID>.zip
(e.g. PVIS_OPC_Server_UA-PC_Installer_2_0_0_3000931B00.zip)

- ▶ Installation directory

As the Zip file is unzipped a directory is created into which the installation file is copied:

PVIS_OPC_Server_UA-PC_Installer_<Version>
(e.g. PVIS_OPC_Server_UA-PC_Installer_2_0_0)

- ▶ Installation file for installing the PVIS OPC Server UA on a PC

PVIS_PC_OPC_ServerUA.exe

Installation file for PMI v8

For a PMI v8 the installation is performed via the PMI v8 Assistant. The PMI Assistant and the installation file are part of a PMI v8.

Further information can be found in the PMI v8 operating manual.

4.3.3 Installation on a PC (manual copying)

If the PVIS OPC Server UA is being operated on a PC, the following structure will result, for example:

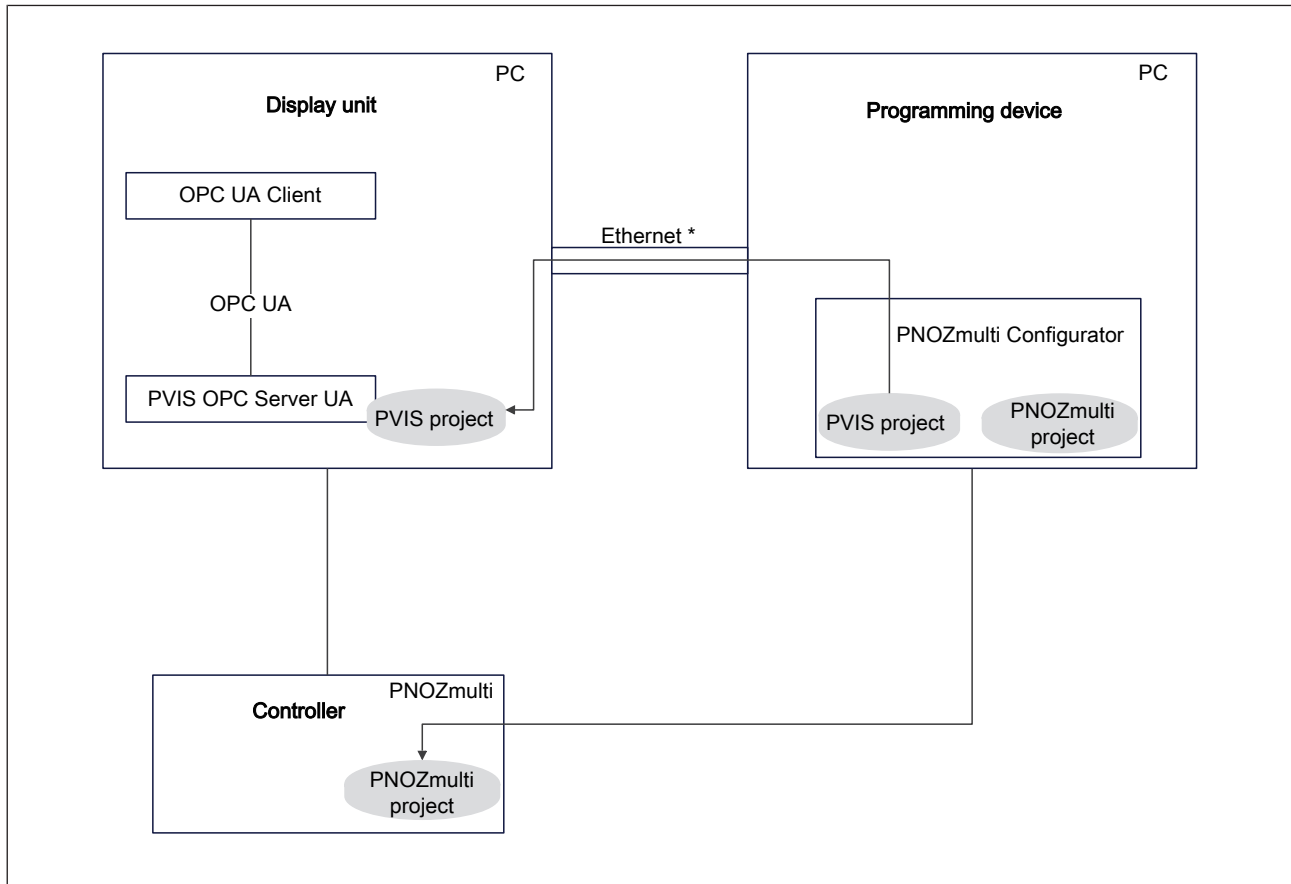


Fig.: Structure of a diagnostic environment with a PVIS OPC Server UA (example)

* The Ethernet connection can be replaced by manual copying (e.g. via a USB stick).

In the example in the diagram, the PVIS OPC Server UA and PNOZmulti Configurator are installed on different PCs. However, it is also possible to install the OPC Server and PNOZmulti Configurator on the same PC.

The OPC Server allows diagnostics on multiple PVIS-compatible controllers from Pilz. The controllers can be connected to the OPC Server either via a network (Ethernet interface) or via the serial interface (COM).

Display units connected via Ethernet can use visualisation software (e.g. visualisation system PASvisu) to display the diagnostic data from the namespace of the OPC Server.

Install the following programs:

- ▶ PVIS OPC Server UA
- ▶ PNOZmulti Configurator

Install the OPC Server on the PC that serves as the diagnostic device. The diagnostic device may also be used simultaneously as a display unit.

Procedure for first-time installation

1. Start the PC on which the PVIS OPC Server UA is to be installed.
2. If you haven't already done so:
Copy the installation file (Zip file) on to the PC and unzip it (see [Installation files](#)  28]).
3. Double-click on the file "PVIS_PC_OPC_ServerUA.exe".
4. Follow the on-screen instructions to complete the installation.
The following tasks are executed on the PC:
 - The software is registered in the registry file
 - PVISOPCServerUA-Starter.exe is entered in the PC's Autostart menu.
 - PVISOPCServerUA-Starter.exe must be executed in order to start the PVIS OPC Server UA.
5. This completes the installation.

Updating the PVIS OPC Server UA or reinstalling on the same PC

1. Start the installer by double-clicking on the file "PVIS_PC_OPC_ServerUA.exe".
2. Select the required option.
3. Follow the on-screen instructions to complete the installation.

4.3.4 Installation on a PMI v8 or PC with PASvisu

If the PVIS OPC Server UA is being operated on a PMI, the following structure will result, for example:

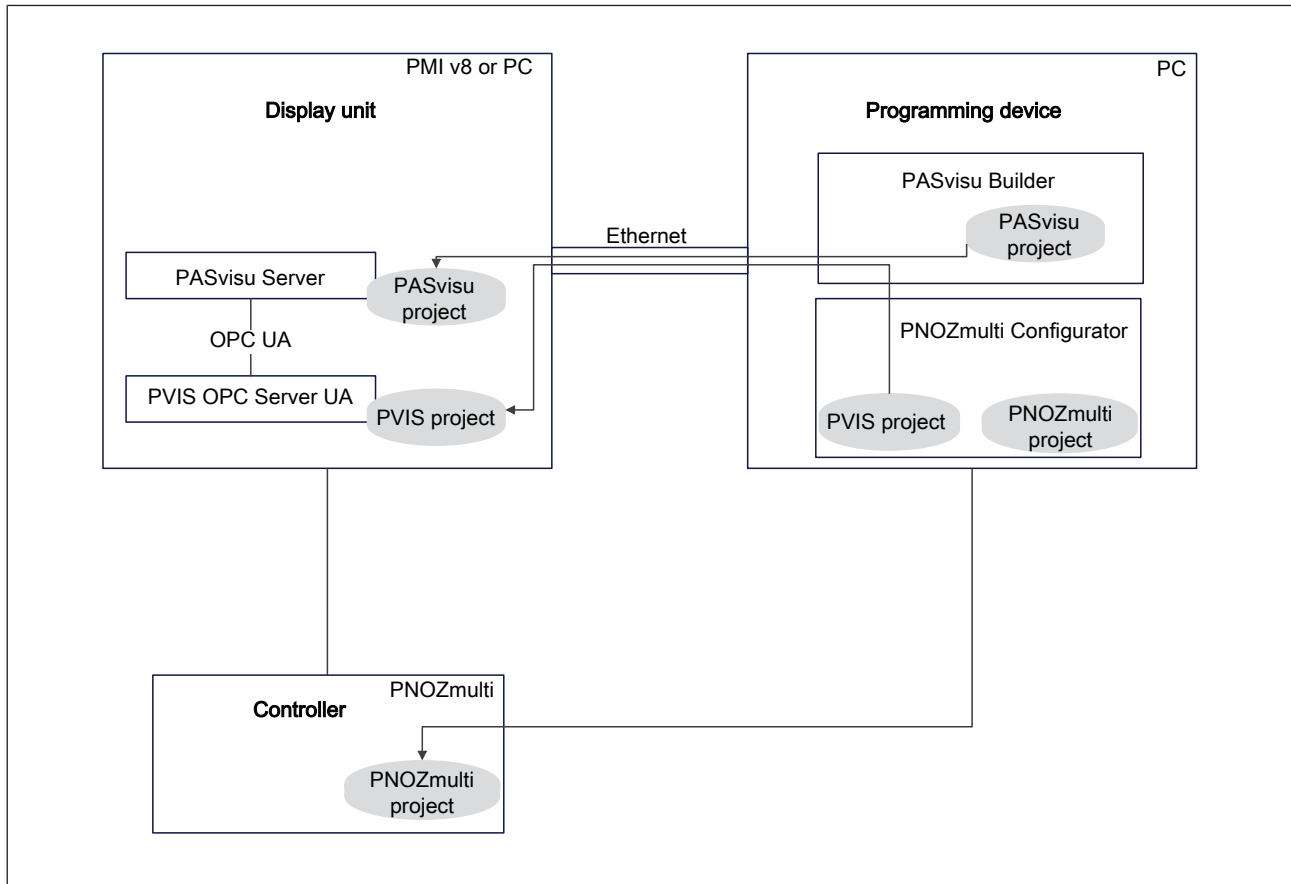


Fig.: Structure of a diagnostic environment with a PVIS OPC Server UA (example)

The PVIS OPC Server UA is installed on the PMI that serves as the diagnostic device. The diagnostic device may also be used simultaneously as a display unit. The PASvisu Builder and PNOZmulti Configurator are always installed on a PC. The PVIS OPC Server UA and PASvisu Builder must be connected via Ethernet.

The controllers can be connected to the OPC Server either via a network (Ethernet interface) or via the serial interface (COM).

Display units connected via Ethernet can use visualisation software (e.g. visualisation system PASvisu) to display the diagnostic data from the namespace of the OPC Server.

Install the following programs:

- ▶ PNOZmulti Configurator (programming device)
- ▶ PASvisu Builder (programming device)
- ▶ PASvisu Server (PC as display unit)
- ▶ PVIS OPC Server UA (PC as display unit)

Procedure for a PMI v8

The PVIS OPC Server UA and PASvisu Server are installed on a PMI v8 with the aid of the "PMI v8 Assistant". The PMI v8 Assistant and all the necessary installation files are part of a PMI v8. Further information can be found in the PMI v8 operating manual.

4.4 Information security certificates

In networked automation technology systems, increasing demands are being placed on information security. Information security encompasses the authenticity, confidentiality, integrity and availability of data as well as protection of access to data.

In the PVIS OPC Server UA from Pilz, information security is established with the aid of certificates compliant with the X.509 standard. The certificates are used to safeguard communication between an OPC UA Client and the OPC UA Server, i.e. a certificate encrypts the data traffic of a unidirectional communication connection and authenticates one communication participant to the other.

A separate certificate is required for each subscriber in client-server communication. A OPC UA Client cannot communicate with the PVIS OPC Server UA until the PVIS OPC Server UA has a valid certificate of the OPC UA Client and the OPC UA Client has a valid certificate of the PVIS OPC Server UA.

4.4.1 Directory structure of the certificate store

A self-signed certificate with a public key and associated private key is created when installing the PVIS OPC Server UA. The certificate, the public key and the non-public (private) key are stored in a certificate store on the installation unit. During installation a memory area for certificates is reserved on the installation unit, i.e. a folder with a pre-defined directory structure is created. The location of the certificate store depends on the device on which the PVIS OPC Server UA is installed. However, the directory structure is always the same:

<basepath>\pki

\own	Location for the public part of the self-signed certificate The public part of the self-signed certificate is always stored in this folder during installation.
\private	Location for the non-public (private) part of the self-signed certificate The non-public part of the self-signed certificate is always stored in this folder during installation.
\rejected	Location for unknown or rejected certificates
\trusted	Location for trusted certificates

Location when installed on a PC or a PMI v8

The certificate store is created in the public area of the PC or PMI v8:

► PVIS OPC Server UA

C:\ProgramData\Pilz\PVISOPCServerUA\pki

4.4.2 Procedure for authentication

To ensure that Client-Server communication can take place, the communication partners must exchange their certificates. For the certificates to be exchanged, the certificate files must be stored in the relevant folder of the certificate store of an OPC UA Client and PVIS OPC Server UA (see [Directory structure of the certificate store](#)  33).

There are different procedures for mutual authentication of the employed PVIS OPC Server UA and OPC UA Client. The procedures can be differentiated according to the following criteria:

- ▶ Using the PASvisu Client as OPC UA Client
 - PVIS OPC Server UA and the visualisation system PASvisu are located on the same device
(see [Authentication in the case of local installation](#)  34)
 - PVIS OPC Server UA and the visualisation system PASvisu are installed on different devices
(see [Authentication in the case of remote installation](#)  34)
- ▶ Using the OPC UA Client of a third-party manufacturer
(see [Authentication in the case of OPC UA Clients of a third-party manufacturer](#)  38)

4.4.2.1 Authentication in the case of local installation

If the Client of the visualisation system PASvisu is used as OPC UA Client and the PVIS OPC Server UA and PASvisu Client are installed on the same device, the certificates are exchanged automatically during installation:

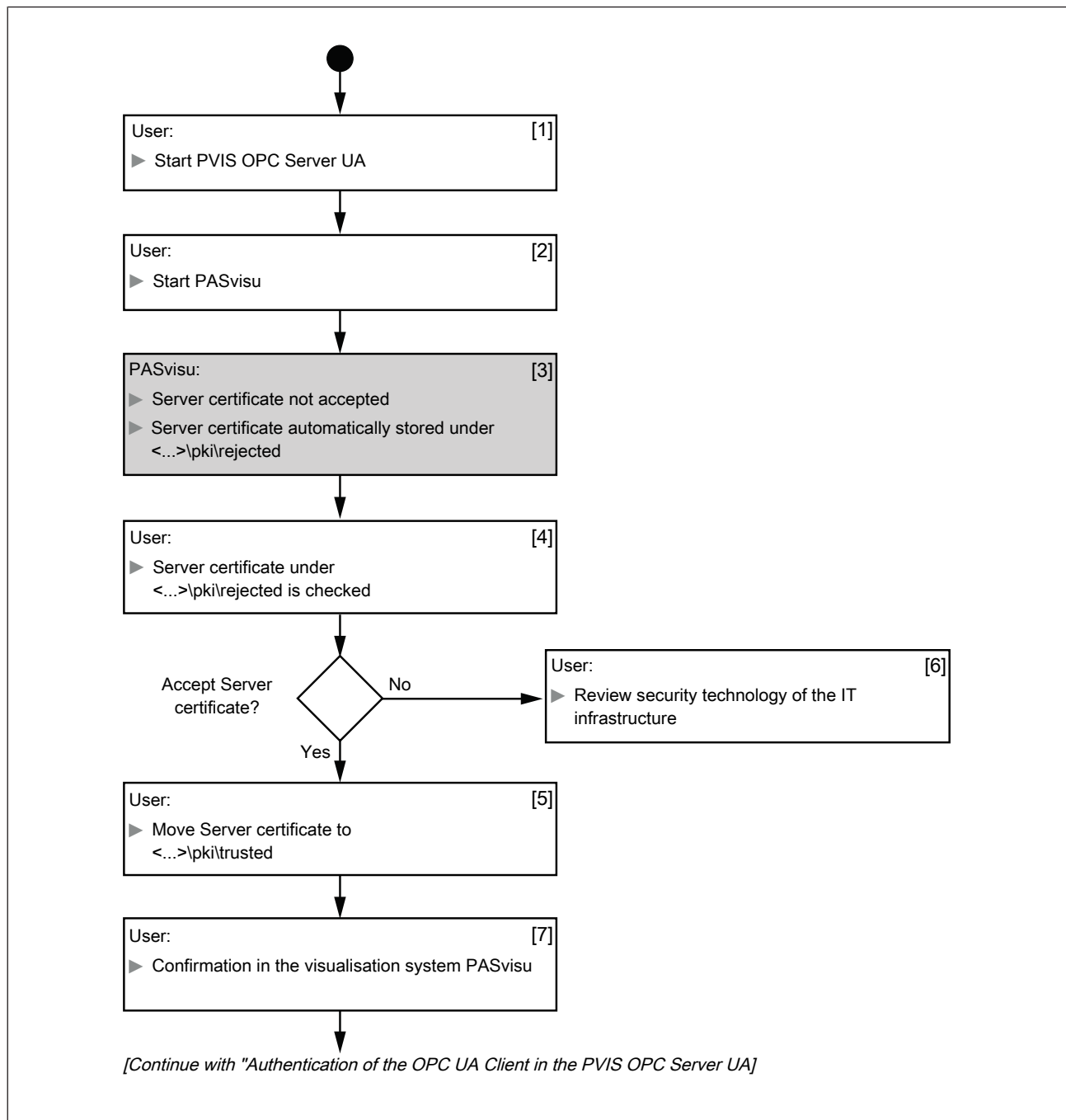
- ▶ The certificate of the PVIS OPC Server UA (cert_Pilz_PVIS_OPC_Server_UA_v2.der) is copied to the "trusted" folder of the PASvisu.
- ▶ The certificate of the PASvisu Client (cert_Pilz_PASvisu_Client_UA.der) is copied to the "trusted" folder of the PVIS OPC Server UA

The user does not need to do anything else.

4.4.2.2 Authentication in the case of remote installation

If the visualisation system PASvisu and the PVIS OPC Server are installed on different devices, then the certificates must be exchanged by the user.

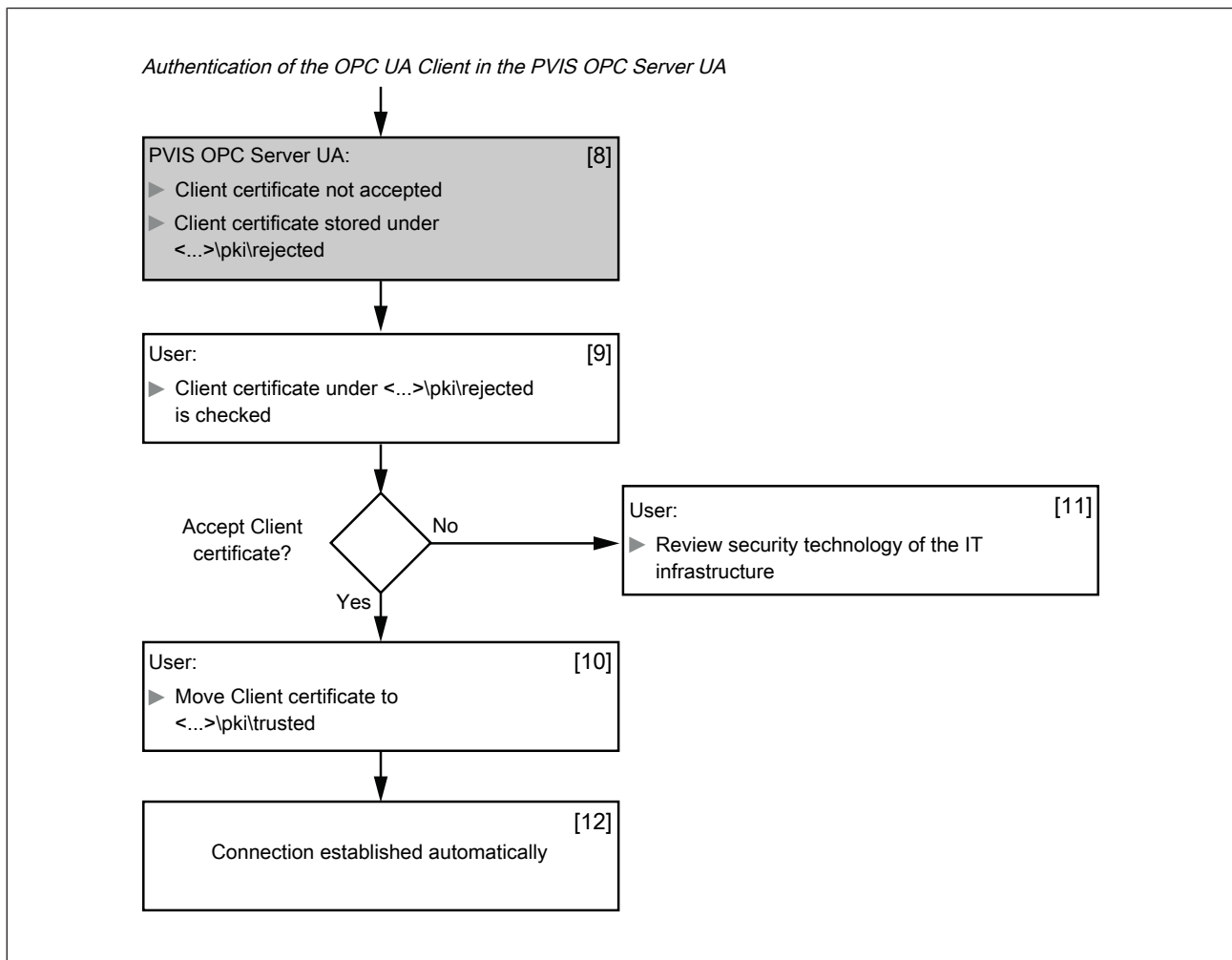
Procedure



Legend

- [1] First time the PVIS OPC Server UA is started after being installed by the user
- [2] First time the visualisation system PASvisu is started after being installed by the user.

- [3] The PASvisu Server's OPC UA Client attempts to establish a connection to the PVIS OPC Server UA. As no valid certificate from the PASvisu OPC Server UA is available to the OPC UA Client after installation, the OPC UA Client reacts as follows:
- ▶ The PASvisu Server's OPC UA Client does not accept the Server certificate
 - ▶ Server certificate is saved in the certificate store of the visualisation system PASvisu in the directory "<...>\pki\rejected".
The name of the certificate file is composed of the common name, domain component and fingerprint.
Structure of the file name: <CN>@<DC>_[Fingerprint].der
 - CN: Common Name
 - DC: Domain Component
 - Fingerprint: 40-digit HEX value
 Example:
 Pilz_PVIS_OPC_Server_UA_self_signed_@192_168_1_101_[0ff981ea8ba58ad04bad16f89a3cc9c0c60e7fac]
 - CN: Pilz PVIS OPC Server UA(self signed)
 - DC: 192.168.1.101
 - Fingerprint: 0ff981ea8ba58ad04bad16f89a3cc9c0c60e7fac
 Note:
 Any characters in the CN and DC that are not in the range A ... Z, a ... z and 0 ... 9 are replaced with an underscore.
 - ▶ Connection to the PVIS OPC Server UA is not established
An error message is displayed.
- [4] The user must check whether the Server certificate is trustworthy (see [Verifying a certificate](#)  41))
- [5] If the user considers the Server certificate to be trustworthy, he must move it from the "rejected" directory to the "trusted" directory in the certificate store of the visualisation system PASvisu.
The PVIS OPC Server UA is now authenticated on the PASvisu Server's OPC UA Client.
- [6] If there are doubts about the trustworthiness of the Server certificate, the security technology of the IT infrastructure should be reviewed.
- [7] For the PVIS OPC Server UA to receive a valid certificate from the PASvisu Server's OPC UA Client, a connection between the two must be established first. To do this, the visualisation system PASvisu must be started.



Legend

[8] As no valid certificate from the PASvisu Server's OPC UA Client is available to the PVIS OPC Server UA after installation, the PVIS OPC Server UA reacts as follows:

- ▶ The PVIS OPC Server UA does not accept the Client certificate
- ▶ The Client certificate is stored in the "<...>\pki\rejected" directory of the certificate store of the PVIS OPC Server UA.

The name of the certificate file is composed of the common name, the domain component and the fingerprint.

Structure of the file name: <CN>@<DC>_[Fingerprint].der

- CN: Common Name
- DC: Domain Component
- Fingerprint: 40-digit HEX value

Example:

Pilz_PAS-visu OPC UA_self_signed_@192_168_1_110_[e0e759ab82345ee7207f1f-da4cedb0153e0d7692]

- CN: Pilz PASvisu OPC UA (self signed)
- DC: 192.168.1.110
- Fingerprint: e0e759ab82345ee7207f1fda4cedb0153e0d7692

Note:

Any characters in the CN and DC that are not in the range A ... Z, a ... z and 0 ... 9 will be replaced with an underscore

- Connection to the PASvisu Server's OPC UA Client is not established

An error message appears.

- [9] The user must check whether the Client certificate is trustworthy (see [Verifying a certificate](#) [41]).
- [10] If the user considers the Client certificate to be trustworthy, he must move it from the "rejected" directory to the "trusted" directory in the certificate store of the visualisation system PASvisu.
The PASvisu Server's OPC UA Client is now authenticated on the PVIS OPC Server UA.
- [11] If there are doubts about the trustworthiness of the Client certificate, the security technology of the IT infrastructure should be reviewed.
- [12] Following mutual authentication, the PASvisu Server's OPC UA Client automatically establishes the connection to the PVIS OPC Server UA.



NOTICE

During authentication, a certificate (Client or Server certificate) is only saved in the directory "<...>\pki\rejected" directory under the following conditions:

- There is no certificate with the same fingerprint in the directory "<...>\pki\rejected"
- There is no valid certificate with the same fingerprint in the directory "<...>\pki\trusted"

4.4.2.3

Authentication in the case of OPC UA Clients of a third-party manufacturer

If the OPC UA Client of a third-party manufacturer is used with the PVIS OPC Server UA from Pilz, the certificates must be exchanged by the user. For the OPC UA Client, follow the instructions in the third-party manufacturer's operating manual.

Procedure

- OPC UA Client

Follow the authentication instructions in the third-party manufacturer's operating manual.

Note:

Some OPC UA Clients need information about the directory in which the OPC Server's certificate is stored. After installation, the certificate of the PVIS OPC Server UA from Pilz is located in the directory <...>\pki\own (see [Directory structure of the certificate store](#) [33]).

- PVIS OPC Server UA

The procedure for the PVIS OPC Server UA from Pilz is in principle the same as the procedure when the PASvisu Client is used as the OPC UA Client (see [Authentication in the case of remote installation](#) [34]).

4.4.3 Generating and using custom certificates

Instead of the certificate that is created when the PVIS OPC Server UA and/or the visualisation system PASvisu is installed, it is possible to generate a custom certificate to be used for safeguarding communication. This requires a suitable tool from a third-party manufacturer. The tool must be used to generate a certificate that has a public part and a non-public (private) part.



INFORMATION

Only use custom certificates if you have experience in generating certificates and in using self-generated certificates.

Requirements of certificates

The certificates should contain the following options:

- ▶ CA: FALSE
- ▶ If the parameter NID_key_usage is used, the value cRLSign must not be set.
- ▶ Encryption algorithm: SecurityPolicy [B] – Basic256Sha256

The PVIS OPC Server UA and the visualisation system PASvisu use RSA certificates with a key length of 2048 Bit and SHA2 (256 Bit).



INFORMATION

It is essential that you note the following documentation when generating and using custom certificates:

- OPC UA specification of the OPC Foundation
- BSI specification
- Documentation of the relevant OPC Client
- Documentation of the tool used to create the certificate

Incorporate certificates

For a certificate to be effective, the public part and the private part of the certificate must have the correct file name and the parts must be located in the correct directory of the certificate store (see [Directory structure of the certificate store](#) [📖 33]).

- ▶ File name of the public part:
 - PVIS OPC Server UA: "cert_Pilz_PVIS_OPC_Server_UA_v2.der"
 - PASvisu: "cert_Pilz_PASvisu_UA.der"
- ▶ File name of the private part:
 - PVIS OPC Server UA: "private_key_Pilz_PVIS_OPC_Server_UA_v2.pem"
 - PASvisu: "private_Pilz_PASvisu_OPC_UA.pem"

Location for the public and private part of a certificate

Location when installed on a PC or PMI v8:

- ▶ <%ALLUSERSPROFILE%>\Pilz\PVISOPCServerUA\pki
 - \own\cert_Pilz_PVIS_OPC_Server_UA_v2.der
 - \private\private_key_Pilz_PVIS_OPC_Server_UA_v2.pem
- ▶ <%LOCALAPPDATA%>\Pilz\PASvisu\pki
 - \own\cert_Pilz_PASvisu_UA.der
 - \private\private_Pilz_PASvisu_OPC_UA.pem

4.4.4 Invalidating a certificate

A certificate is invalidated if the user removes the certificate from the directory <...>\pki\trusted (see [Directory structure of the certificate store](#) [ 33]).

4.4.5 Verifying a certificate

Before a certificate is copied from the directory <...>\pki\rejected to the directory <...>\pki\trusted during authentication, the user should verify the trustworthiness of the certificate (see [Authentication in the case of remote installation](#) [ 34]).

- ▶ Visualisation system PASvisu

The content of the certificate in the directory <...>\pki\rejected of the visualisation system PASvisu should be compared with the content of the certificate in the directory <...>\pki\own of the PVIS OPC Server UA. It is sufficient just to compare the fingerprints.

- ▶ PVIS OPC Server UA

The content of the certificate in the directory <...>\pki\rejected of the PVIS OPC Server UA should be compared with the content of the certificate in the directory <...>\pki\own of the visualisation system PASvisu. It is sufficient just to compare the fingerprints.

Procedure with a PC or PMI v8

- ▶ Double-click on a certificate to view its content.

The contents of the two certificates to be compared must be identical.

4.5 Event view

The OPC Server "PVIS OPC Server UA" stores all the event messages in the event log. The event log is a ring memory, in other words, the most recent event messages are stored. The number of events in the ring memory is configurable.

The OPC Server should be synchronised with all the connected control systems to ensure that the events are stored in the event view with the correct time stamp.

4.6 Commissioning the PVIS OPC Server UA

You can use the PASvisu Builder to transfer the diagnostic configuration of a PNOZmulti project to the PVIS OPC Server UA.

Prerequisites

- ▶ The PASvisu Builder is installed.
- ▶ The PNOZmulti project with diagnostic configuration is available. The PASvisu Builder can access the PNOZmulti project as data source.
- ▶ The PVIS OPC Server UA is installed.
If the PVIS OPC Server UA is run on a PMI, there must be an Ethernet connection between the PC with the PASvisu Builder and the PMI.
- ▶ The PNOZmulti is connected to the PC or PMI that is running the PVIS OPC ServerUA.
This is described in the operating manuals for the PMI and PNOZmulti.

Starting the PVIS OPC Server UA on a PC

The PVIS OPC Server UA is started when the PC is booted up (Autostart -> PVISOPC-ServerUA-Starter.exe). It is also possible to start the OPC Server using the taskbar of the PC:

- ▶ Right-click on the icon  in the taskbar and select "Restart".
The "Pilz PVIS OPC Server UA – Starter" starts the PVIS OPC Server UA.

Starting the PVIS OPC Server UA on a PMI

The PVIS OPC Server UA is started when the PMI is booted up.



INFORMATION

Note:

- The PVIS OPC Server UA loads the OPC project on each boot process or restart.
- The PVIS OPC Server UA only starts if an OPC project has been transferred or an OPC project is present.

4.7 Transfer PVIS OPC project manually

A PVIS OPC project can be transferred to the PVIS OPC Server UA either via the PASvisu Builder or manually.

To transfer manually, proceed as follows:

- ▶ Save the project in the PNOZmulti Configurator. A directory is created, "PVISOPCData", which contains another directory with the name of the PNOZmulti project and the abbreviation "_opc" (... \Documents\PNOZmulti\<Project name>\PVISOPCData\<Project name>_opc). This contains the OPC Server UA configuration.
- ▶ Copy this directory into the configuration memory of the PVIS OPC Server UA. The configuration memory is stored in the directory "PVISOPCData" (C:\Program-Data\Pilz\PVISOPCData).
- ▶ Restart the PVIS OPC Server UA to apply the new configuration. Either by restarting the device or via **Restart** in the tray icon of the PVIS OPC Server UA.

4.8 Language management

In the PNOZmulti Configurator, the user can define the languages that are to be downloaded to the OPC Server "PVIS OPC Server UA". These languages must be available in each device project in the OPC project. The user defines the languages for the device projects when they create the diagnostic configuration in the system software's diagnostic configurator (PNOZmulti Configurator). A device project's diagnostic configuration is made up of one part that is pre-defined by Pilz and one part that is created by the user.

The user defines the language required for both parts:

- ▶ Pilz diagnostic language/alternative language
Language for the device's diagnostic configuration and the Pilz pre-defined diagnostic types for safety devices
- ▶ Project language
This is the language in which the user creates the texts for the diagnostic configuration, e.g. device names, equipment identifiers and user's diagnostic types

The user also defines the default language. The default language is used as follows: If a visualisation system or another OPC UA Client requests event messages in a language that's unavailable, the OPC Server will display the event messages in the default language.

Example: Certain languages have been selected for an OPC project. The OPC UA Client calls up information from the OPC Server in one of these languages. If this language is not available within a device project, the OPC UA Client will receive the information in the default language.

5 Namespace

5.1 Basics

The namespace is a memory area in which the OPC Server "PVIS OPC Server UA" stores all the diagnostic data and process data from the connected controllers. For the most part the namespace can be viewed using the browser on an OPC UA Client.

Users themselves must program access to the process data.

Within the namespace there is read/write access to the variables of the virtual inputs on a PNOZmulti. Access to all other variables is exclusively read-only.

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 (= elements). The elements are found exclusively on the lowest level. The full name that uniquely identifies an element in the namespace is called the fully qualified name (FQN). The FQN consists of the hierarchical node name (stated path) and the name of the element. 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.

5.1.1 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. <OPCProjectName>, for example, becomes MyProject.
- ▶ Small numbers below a box signify the possible number of nodes.
 - "1 .. 2" means that up to two nodes are available.
 - The infinity symbol (∞) means that the number of nodes is open, in other words, you can have as many nodes as you like. The actual number of nodes depends on the configuration of the OPC project.
- ▶ Indication above a box means that these nodes and everything below them are only valid for this product:
 - PNOZmulti: configurable safe small controllers PNOZmulti Classic
 - PNOZmulti Mini: configurable safe compact controllers PNOZmulti Mini
 - PNOZmulti 2: configurable safe small controllers PNOZmulti 2

Note: If required, a distinction is made between PNOZmulti m B0 and PNOZmulti m B1.

 - PMC: Motion control systems
- ▶ "/" is used as the separator when stating the path or indicating the FQN.

5.2 Structure of the namespace

5.2.1 OPCProjectName

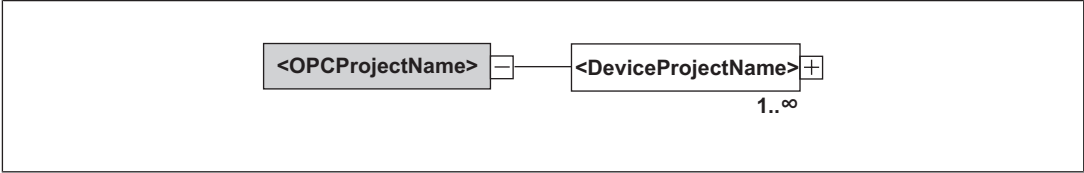


Fig.: Area <OPCProjectName>.

Node name	Meaning
<OPCProjectName>	Name of the OPC project Default name: "DeviceProjects"

5.2.2 DeviceProjectName

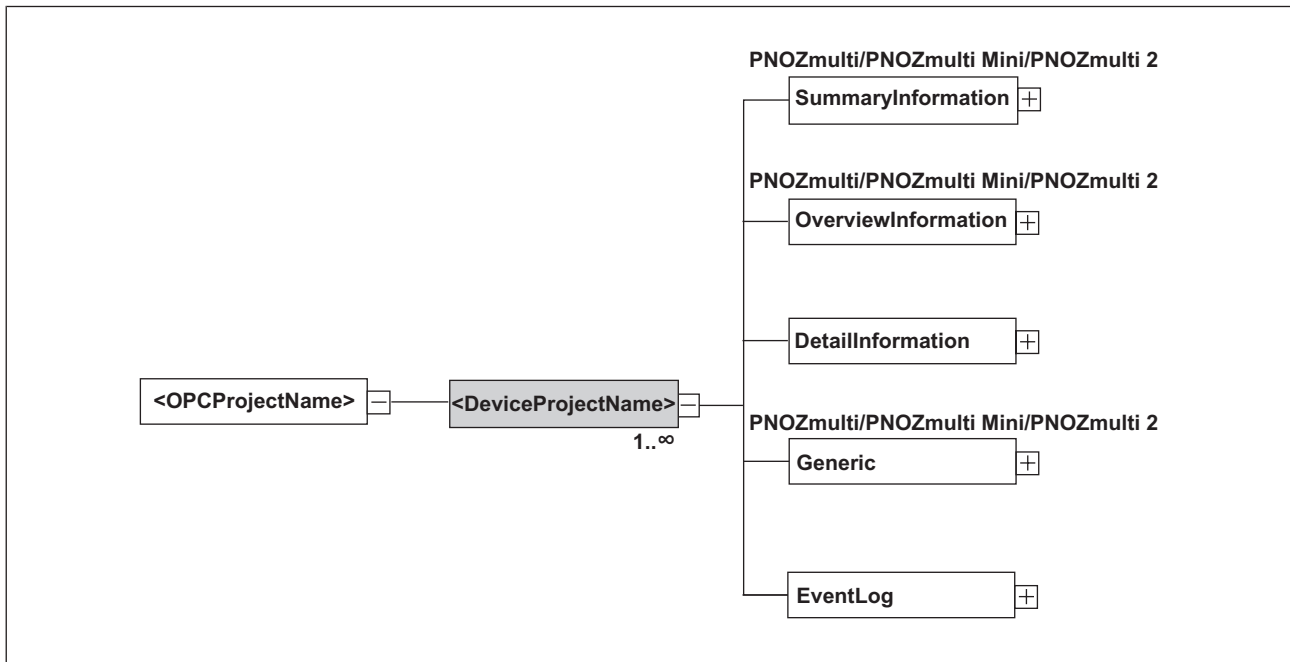


Fig.: <OPCProjectName>/<DeviceProjectName>/

Node name	Meaning
<DeviceProjectName>	Name of the device project e.g.: MyDeviceProject
Path: <OPCProjectName>/<DeviceProjectName>/	
SummaryInformation	Area containing summary information about entries in the event list (see DeviceProjectName/SummaryInformation [49])
Path: <OPCProjectName>/<DeviceProjectName>/SummaryInformation/	
OverviewInformation	Area containing status information about the event list for modules from the configurable systems PNOZmulti, PNOZmulti Mini and PNOZmulti 2 (see DeviceProjectName/OverviewInformation [53])
Path: <OPCProjectName>/<DeviceProjectName>/OverviewInformation/	
DetailInformation	Area containing detailed information about entries in the event list (see DeviceProjectName/DetailInformation [58])
Path: <OPCProjectName>/<DeviceProjectName>/DetailInformation/	
Generic	Area containing device-specific process data The process data can be accessed via the absolute addresses (see DeviceProjectName/Generic [69])
Path: <OPCProjectName>/<DeviceProjectName>/Generic	

Node name		Meaning
	EventLog	Area containing the entries of the event log (see DeviceProjectName/EventLog  79)
	Path: <OPCProjectName>/<DeviceProjectName>/EventLog/	

5.2.3 DeviceProjectName/SummaryInformation

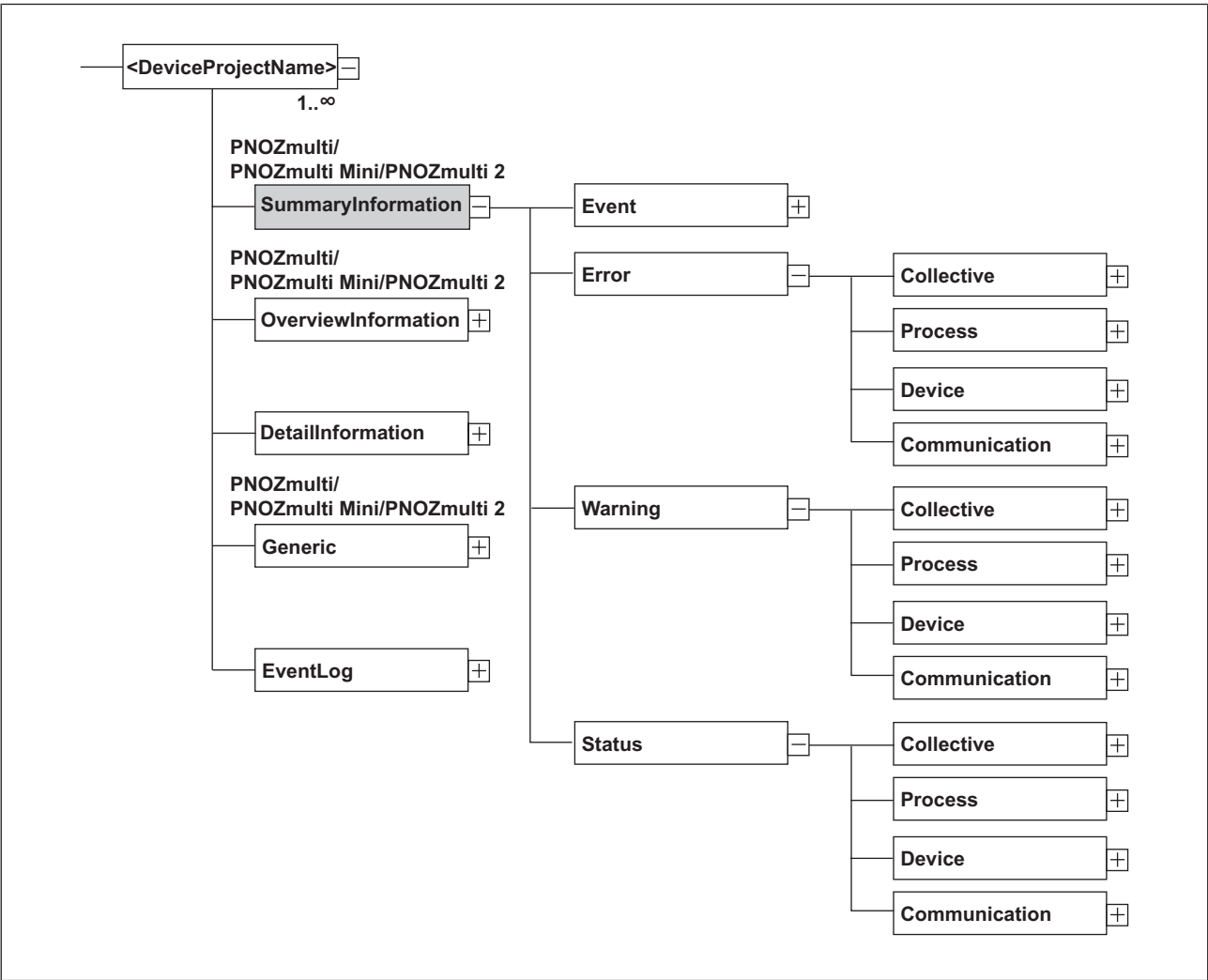


Fig.: Area <OPCProjectName>/<DeviceProjectName>/SummaryInformation/

Node name		Meaning	
SummaryInformation		Area containing summary information on entries in the event list	
		A maximum of the first 100 entries from the event list are evaluated.	
Path: <OPCProjectName>/<DeviceProjectName>/SummaryInformation/			
Event		Event occurred/not occurred	
		Data type: Boolean	
		Display	Meaning
		FALSE	No event occurred
		TRUE	Event occurred
FQN: <OPCProjectName>/<DeviceProjectName>/SummaryInformation/Event			

Node name	Meaning	
Error	Area containing information on event messages of priority class "Error"	
Path: <OPCProjectName>/<DeviceProjectName>/SummaryInformation/Error/		
Collective	Information on event messages of priority class "Error"	
	Data type: Boolean	
	Display	Meaning
	FALSE	No error present
	TRUE	Error present
FQN: <OPCProjectName>/<DeviceProjectName>/SummaryInformation/Error/Collective		
Process	Information on event messages of priority class "Error" that belong to the "Process" area	
	Data type: Boolean	
	Display	Meaning
	FALSE	No error present
	TRUE	Error present
FQN: <OPCProjectName>/<DeviceProjectName>/SummaryInformation/Error/Process		
Device	Information on event messages of priority class "Error" that belong to the "Plant/device" area	
	Data type: Boolean	
	Display	Meaning
	FALSE	No error present
	TRUE	Error present
FQN: <OPCProjectName>/<DeviceProjectName>/SummaryInformation/Error/Device		
Communication	Information on event messages of priority class "Error" that belong to the "Communication" area	
	Data type: Boolean	
	Display	Meaning
	FALSE	No error present
	TRUE	Error present
FQN: <OPCProjectName>/<DeviceProjectName>/SummaryInformation/Error/Communication		
Warning	Area containing information on event messages of priority class "Warning"	
Path: <OPCProjectName>/<DeviceProjectName>/SummaryInformation/Warning/		

Node name	Meaning
Collective	Event message of the priority class "Warning" present/not present Data type: Boolean
	Display
	FALSE
	TRUE
FQN: <OPCProjectName>/<DeviceProjectName>/SummaryInformation/Warning/Collective	
Process	Information on event messages of priority class "Warning" that belong to the "Process" area Data type: Boolean
	Display
	FALSE
	TRUE
FQN: <OPCProjectName>/<DeviceProjectName>/SummaryInformation/Warning/Process	
Device	Information on event messages of priority class "Warning" that belong to the "Plant/device" area Data type: Boolean
	Display
	FALSE
	TRUE
FQN: <OPCProjectName>/<DeviceProjectName>/SummaryInformation/Warning/Device	
Communication	Information on event messages of priority class "Warning" that belong to the "Communication" area Data type: Boolean
	Display
	FALSE
	TRUE
FQN: <OPCProjectName>/<DeviceProjectName>/SummaryInformation/Warning/Communication	
Status	Area containing information on event messages of priority class "Status information"
Path: <OPCProjectName>/<DeviceProjectName>/SummaryInformation/Status/	

Node name	Meaning
Collective	Event message of the priority class "Status information" present/not present Data type: Boolean
	Display
	FALSE
	TRUE
FQN: <OPCProjectName>/<DeviceProjectName>/SummaryInformation/Status/Collective	
Process	Information on event messages of priority class "Status information" that belong to the "Process" area Data type: Boolean
	Display
	FALSE
	TRUE
FQN: <OPCProjectName>/<DeviceProjectName>/SummaryInformation/Status/Process	
Device	Information on event messages of priority class "Status information" that belong to the "Plant/device" area Data type: Boolean
	Display
	FALSE
	TRUE
FQN: <OPCProjectName>/<DeviceProjectName>/SummaryInformation/Status/Device	
Communication	Information on event messages of priority class "Status information" that belong to the "Communication" area Data type: Boolean
	Display
	FALSE
	TRUE
FQN: <OPCProjectName>/<DeviceProjectName>/SummaryInformation/Status/Communication	

5.2.4 DeviceProjectName/OverviewInformation

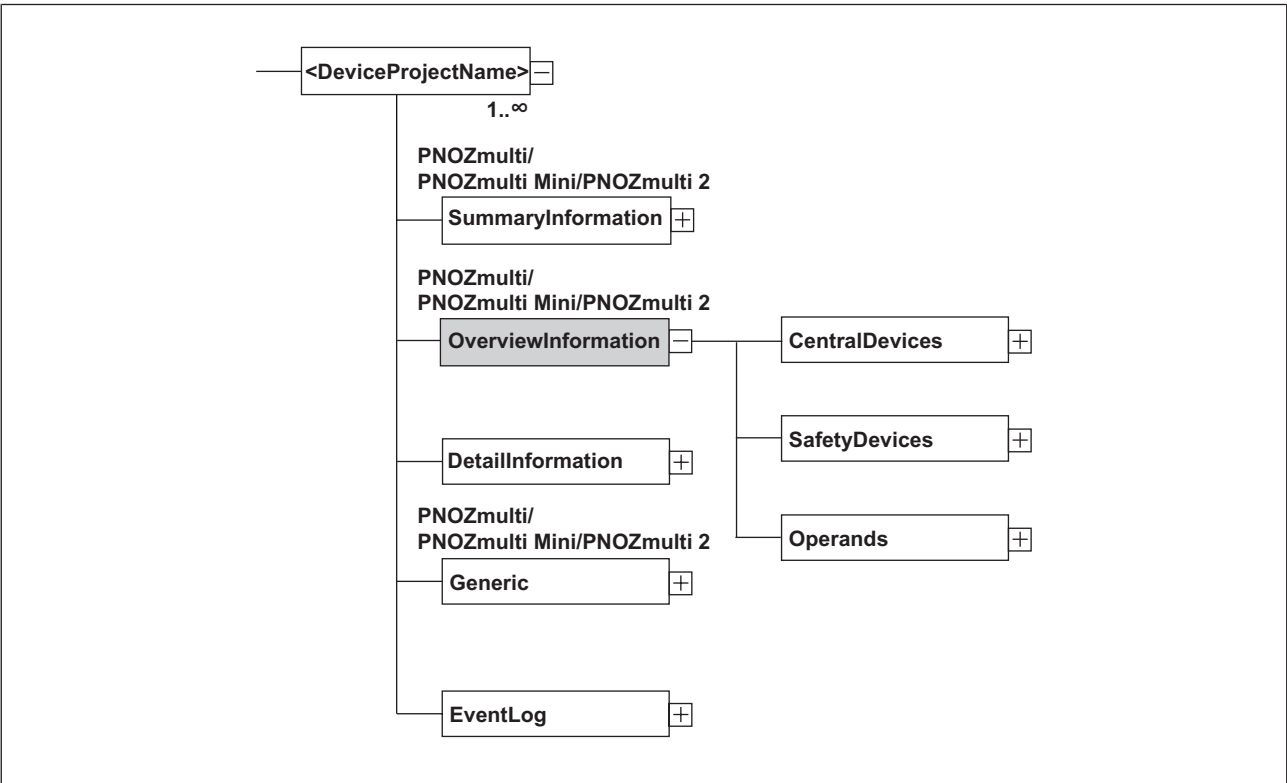


Fig.: Area <OPCProjectName>/<DeviceProjectName>/OverviewInformation/

Node name		Meaning	
OverviewInformation		Area containing status information from the event list A maximum of the first 100 entries from the event list are evaluated.	
		PNOZmulti, PNOZmulti Mini, PNOZmulti 2	Area containing status information about modules from the configurable systems PNOZmulti, PNOZmulti Mini und PNOZmulti 2
Path: <OPCProjectName>/<DeviceProjectName>/OverviewInformation/			
	CentralDevices	PNOZmulti, PNOZmulti Mini PNOZmulti 2	Area containing status information about modules
		(see CentralDevices  54])	
	Path: <OPCProjectName>/<DeviceProjectName>/OverviewInformation/ CentralDevices/		
	SafetyDevices	Area containing status information about safety devices (see SafetyDevices  55])	
		Path: <OPCProjectName>/<DeviceProjectName>/OverviewInformation/Safety- Devices/	

Node name	Meaning
Operands	Area containing information about the operands for which operand diagnostics are configured (see Operands [57])
Path: <OPCProjectName>/<DeviceProjectName>/OverviewInformation/Operands/	

5.2.4.1 CentralDevices

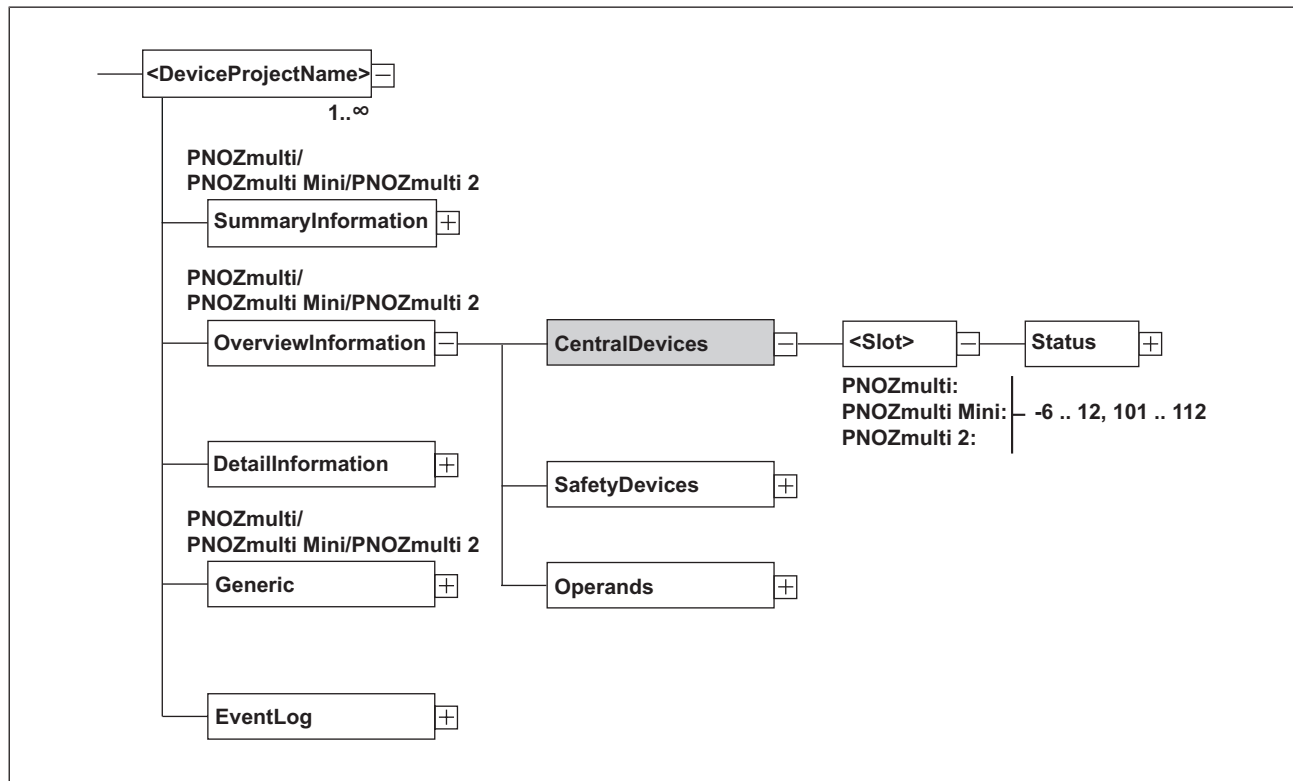


Fig.: Area <OPCProjectName>/<DeviceProjectName>/OverviewInformation/CentralDevices/

Node name	Meaning	
CentralDevices	PNOZmulti, PNOZmulti Mini PNOZmulti 2	Area containing status information about modules
Path: <OPCProjectName>/<DeviceProjectName>/OverviewInformation/CentralDevices/		
<Slot>	PNOZmulti, PNOZmulti Mini PNOZmulti 2	Number of the expansion device ▶ Expansion devices on the left are identified by "-<Number>" (e.g. "-4") ▶ Expansion devices on the right are identified by "+<Number>" (e.g. "+2")
Path: <OPCProjectName>/<DeviceProjectName>/OverviewInformation/CentralDevices/<Slot>/		

Node name	Meaning	
Status	Event occurred/not occurred	
	The event is displayed in Bit 00.	
	Data type: UInt16	
	Display	Meaning
	FALSE	No event occurred
	TRUE	Event occurred
FQN: <OPCProjectName>/<DeviceProjectName>/OverviewInformation/ CentralDevices/<Slot>/Status		

5.2.4.2 OverviewInformation

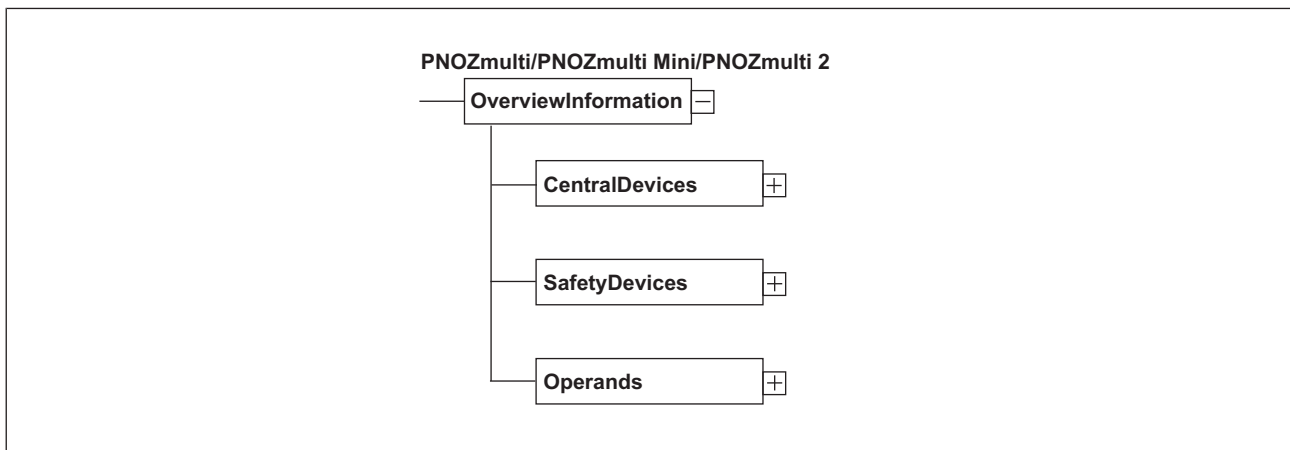


Fig.: Area <OPCProjectName>/<DeviceProjectName>/OverviewInformation

5.2.4.3 SafetyDevices

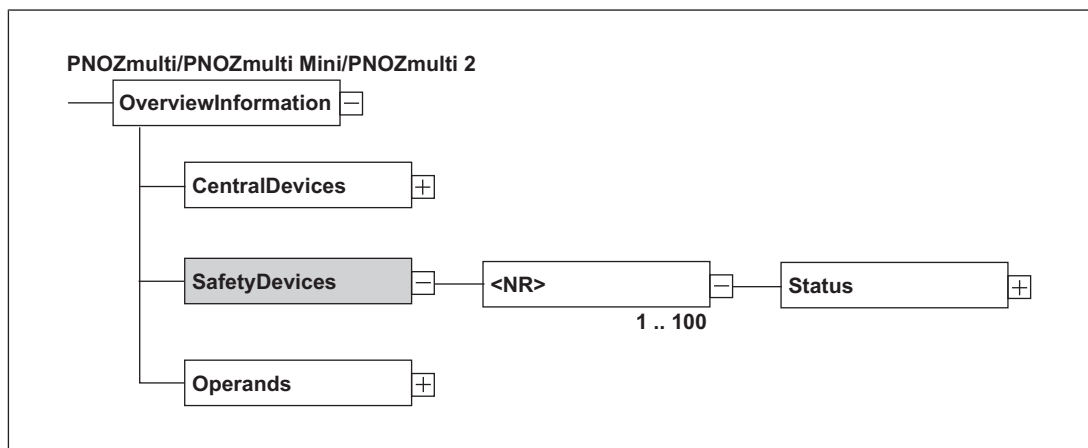


Fig.: Area <OPCProjectName>/<DeviceProjectName>/OverviewInformation/SafetyDevices/

Node name	Meaning							
SafetyDevices	Area containing status information about safety devices							
Path: <OPCProjectName>/<DeviceProjectName>/OverviewInformation/SafetyDevices/								
<NR>	PNOZmulti, PNOZmulti Mini PNOZmulti 2	Element ID The node name can be "1" ... "100".						
Path: <OPCProjectName>/<DeviceProjectName>/OverviewInformation/ SafetyDevices/<NR>/								
Status	Status of the safety device Data type: UInt16							
	Bit 00: Safety device active/inactive							
		<table><tr><th>Display</th><th>Meaning</th></tr><tr><td>FALSE</td><td>Safety device active</td></tr><tr><td>TRUE</td><td>Safety device inactive</td></tr></table>	Display	Meaning	FALSE	Safety device active	TRUE	Safety device inactive
	Display	Meaning						
	FALSE	Safety device active						
TRUE	Safety device inactive							
	Bit 01: Safety device was/was not triggered							
		<table><tr><th>Display</th><th>Meaning</th></tr><tr><td>FALSE</td><td>Safety device was not triggered</td></tr><tr><td>TRUE</td><td>Safety device was triggered</td></tr></table>	Display	Meaning	FALSE	Safety device was not triggered	TRUE	Safety device was triggered
	Display	Meaning						
	FALSE	Safety device was not triggered						
TRUE	Safety device was triggered							
	Bit 02: Safety device ready for reset/not ready for reset							
		<table><tr><th>Display</th><th>Meaning</th></tr><tr><td>FALSE</td><td>Safety device not ready for reset</td></tr><tr><td>TRUE</td><td>Safety device ready for reset</td></tr></table>	Display	Meaning	FALSE	Safety device not ready for reset	TRUE	Safety device ready for reset
	Display	Meaning						
	FALSE	Safety device not ready for reset						
TRUE	Safety device ready for reset							
	Bit 03: Function test							
		<table><tr><th>Display</th><th>Meaning</th></tr><tr><td>FALSE</td><td>Function test not required</td></tr><tr><td>TRUE</td><td>Function test required</td></tr></table>	Display	Meaning	FALSE	Function test not required	TRUE	Function test required
	Display	Meaning						
	FALSE	Function test not required						
TRUE	Function test required							
	Bit 04: Application error							
		<table><tr><th>Display</th><th>Meaning</th></tr><tr><td>FALSE</td><td>Error not present</td></tr><tr><td>TRUE</td><td>Error present</td></tr></table>	Display	Meaning	FALSE	Error not present	TRUE	Error present
	Display	Meaning						
	FALSE	Error not present						
TRUE	Error present							
	Bit 05: Plausibility error							
		<table><tr><th>Display</th><th>Meaning</th></tr><tr><td>FALSE</td><td>Error not present</td></tr><tr><td>TRUE</td><td>Error present</td></tr></table>	Display	Meaning	FALSE	Error not present	TRUE	Error present
	Display	Meaning						
	FALSE	Error not present						
TRUE	Error present							
Bit 06 ... Bit 14: Reserved								
	Bit 15: Device error							
		<table><tr><th>Display</th><th>Meaning</th></tr><tr><td>FALSE</td><td>Error not present</td></tr><tr><td>TRUE</td><td>Error present</td></tr></table>	Display	Meaning	FALSE	Error not present	TRUE	Error present
	Display	Meaning						
	FALSE	Error not present						
TRUE	Error present							

Node name	Meaning
FQN: <OPCProjectName>/<DeviceProjectName>/OverviewInformation/ SafetyDevices/<NR>/Status	

5.2.4.4 Operands

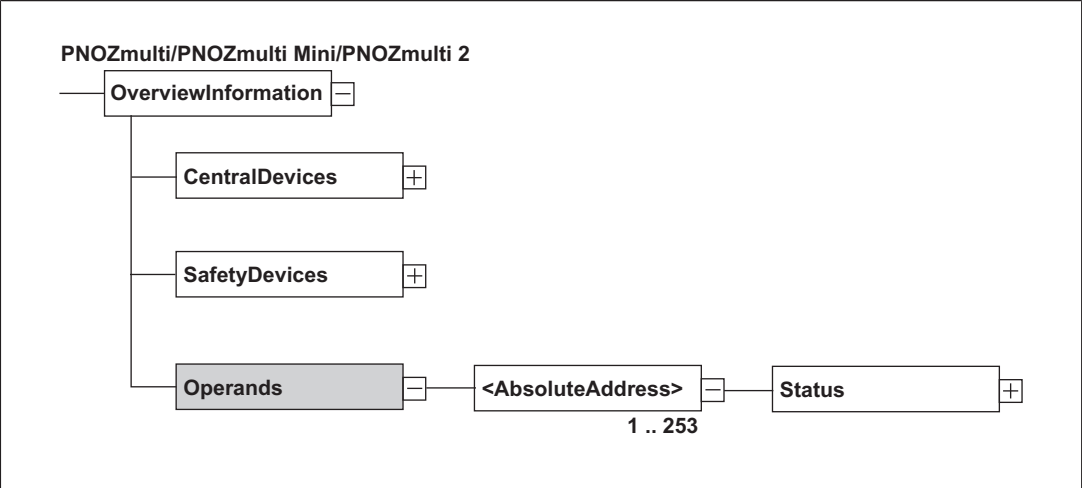


Fig.: Area <OPCProjectName>/<DeviceProjectName>/OverviewInformation/Operands/

Node name	Meaning	
Operands	Area containing information about the operands for which oper- and diagnostics are configured	
Path: <OPCProjectName>/<DeviceProjectName>/OverviewInformation/Operands/		
<AbsoluteAddress>	PNOZmulti, PNOZmulti Mini PNOZmulti 2	Operand ID The node name can be "1" ... "253".
Path: <OPCProjectName>/<DeviceProjectName>/OverviewInformation/ Operands/<AbsoluteAddress>/		
Status	Event for the operand occurred/did not occur (event list)	
	The event is displayed in Bit 00.	
	Data type: UInt16	
	Display	Meaning
	FALSE	No event occurred
	TRUE	Event occurred
FQN: <OPCProjectName>/<DeviceProjectName>/OverviewInformation/ Operands/<AbsoluteAddress>/Status		

5.2.5 DeviceProjectName/DetailInformation

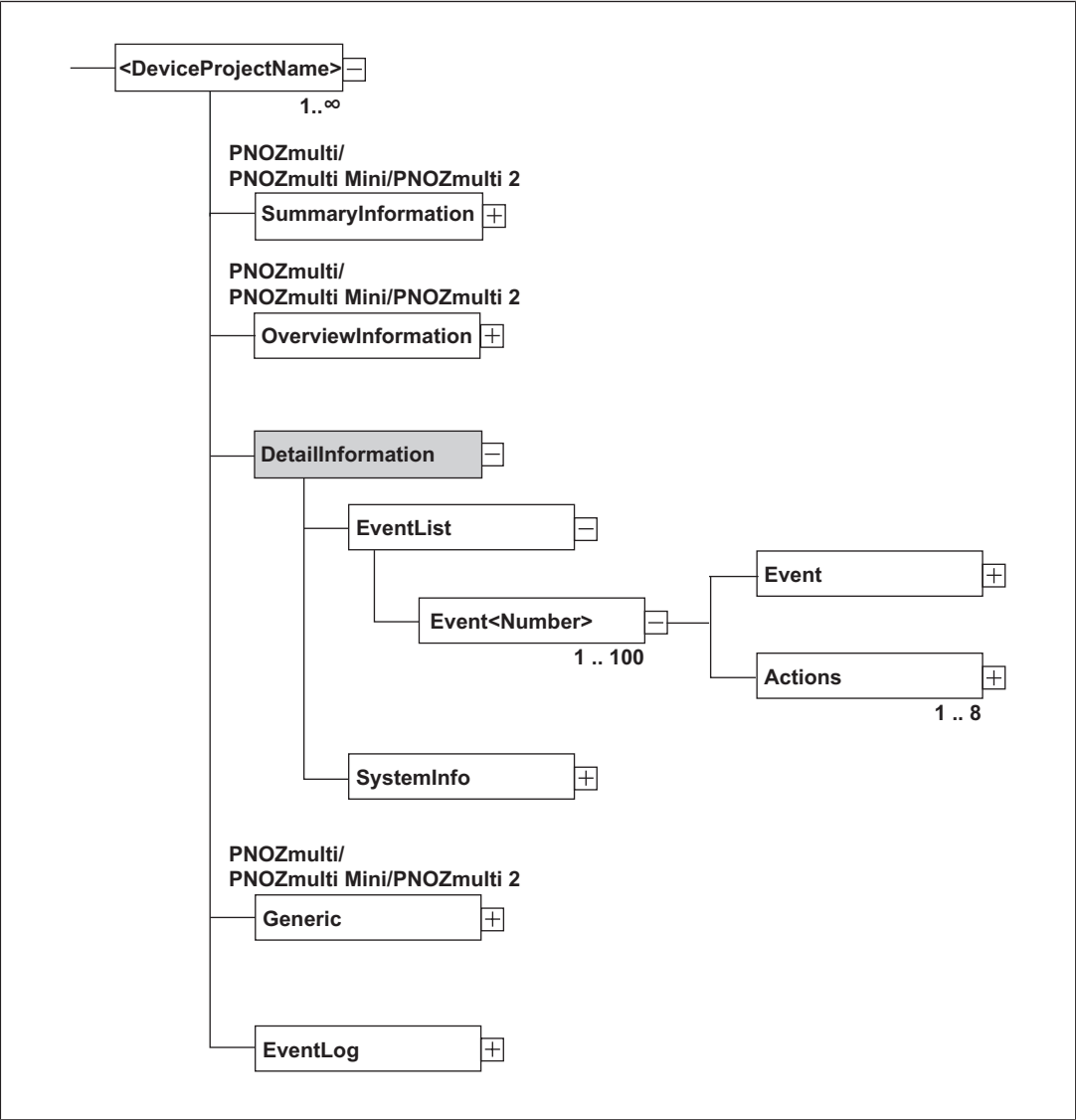


Fig.: Area <OPCProjectName>/<DeviceProjectName>/DetailInformation/

Node name	Meaning
DetailInformation	Area containing detailed information about entries in the event list
Path: <OPCProjectName>/<DeviceProjectName>/DetailInformation/	
EventList	Area containing detailed information about the first 10 events in the event list (default configuration). A value of 1 ... 100 can be configured in the PNOZmulti Configurator for the number of events.
Path: <OPCProjectName>/<DeviceProjectName>/DetailInformation/EventList/	
Event<Number>	Number of the event (node name, e.g. "Event 8")

Node name		Meaning
Path: <OPCProjectName>/<DeviceProjectName>/DetailInformation/EventList/Event<Number>/		
	Event	Details of the event see Event  59]
	Path: <OPCProjectName>/<DeviceProjectName>/DetailInformation/EventList/Event<Number>/Event/	
	Actions	Details of the actions of the event see Actions  61]
	Path: <OPCProjectName>/<DeviceProjectName>/DetailInformation/EventList/Event<Number>/Actions/	
SystemInfo	PNOZmulti, PNOZmulti Mini, PNOZmulti 2	Area containing system information about the diagnostic configuration and PNOZmulti project see PNOZmulti/PNOZmulti Mini/PNOZmulti 2: SystemInfo  63]
	PMC	Area containing system information about the diagnostic configuration and the PMC project see PMC: SystemInfo  66]
Path: <OPCProjectName>/<DeviceProjectName>/DetailInformation/SystemInfo/		

5.2.5.1 Event

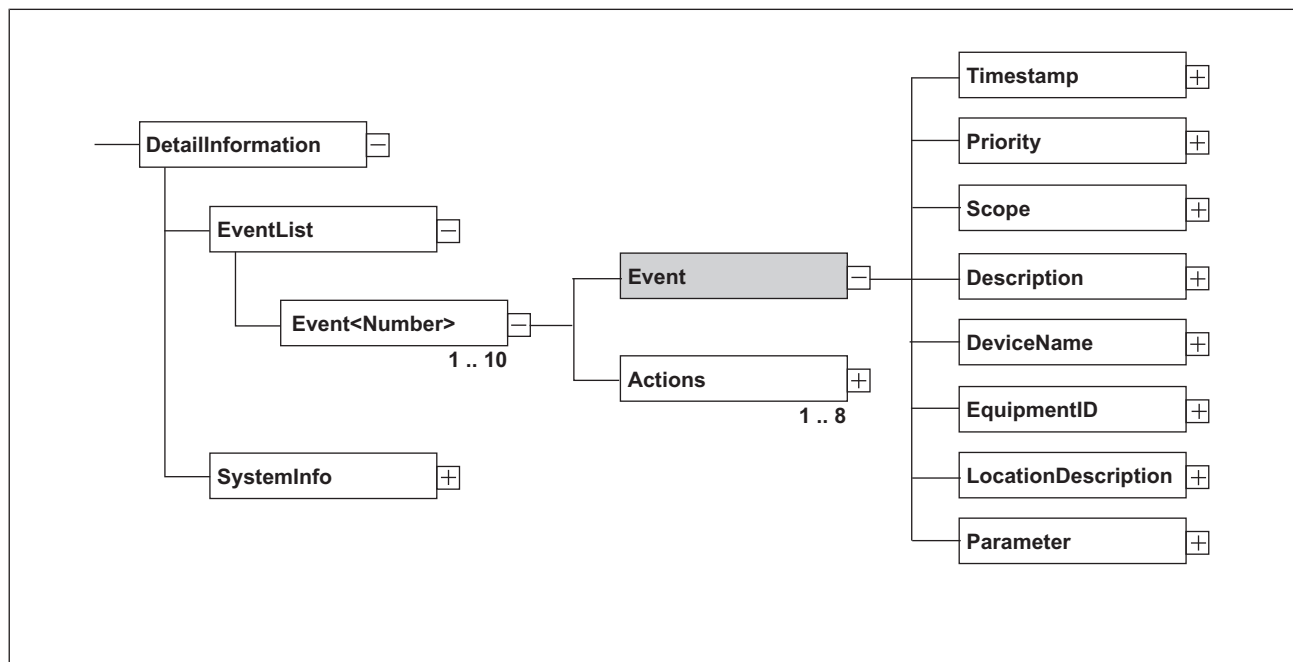


Fig.: Area <OPCProjectName>/<DeviceProjectName>/DetailInformation/EventList/Event<Number>/Event/

Node name		Meaning
Event		Details of the event
Path: <OPCProjectName>/<DeviceProjectName>/DetailInformation/EventList/Event<Number>/Event/		
	Timestamp	Time stamp Data type: String
	FQN: <OPCProjectName>/<DeviceProjectName>/DetailInformation/EventList/Event<Number>/Event/Timestamp	
	Priority	Priority Data type: String
	FQN: <OPCProjectName>/<DeviceProjectName>/DetailInformation/EventList/Event<Number>/Event/Priority	
	Scope	Scope Data type: String
	FQN: <OPCProjectName>/<DeviceProjectName>/DetailInformation/EventList/Event<Number>/Event/Scope	
	Description	Description Data type: String
	FQN: <OPCProjectName>/<DeviceProjectName>/DetailInformation/EventList/Event<Number>/Event/Description	
	DeviceName	Device name Data type: String
	FQN: <OPCProjectName>/<DeviceProjectName>/DetailInformation/EventList/Event<Number>/Event/DeviceName	
	EquipmentID	Equipment identifier (Equip-ID) Data type: String
	FQN: <OPCProjectName>/<DeviceProjectName>/DetailInformation/EventList/Event<Number>/Event/EquipmentID	
	LocationDescription	Location description Data type: String
	FQN: <OPCProjectName>/<DeviceProjectName>/DetailInformation/EventList/Event<Number>/Event/LocationDescription	
	Parameter	More precise information on where an event occurred, e.g. information on the input terminal ("E3.0") Data type: String
	FQN: <OPCProjectName>/<DeviceProjectName>/DetailInformation/EventList/Event<Number>/Event/Parameter	

5.2.5.2 Actions

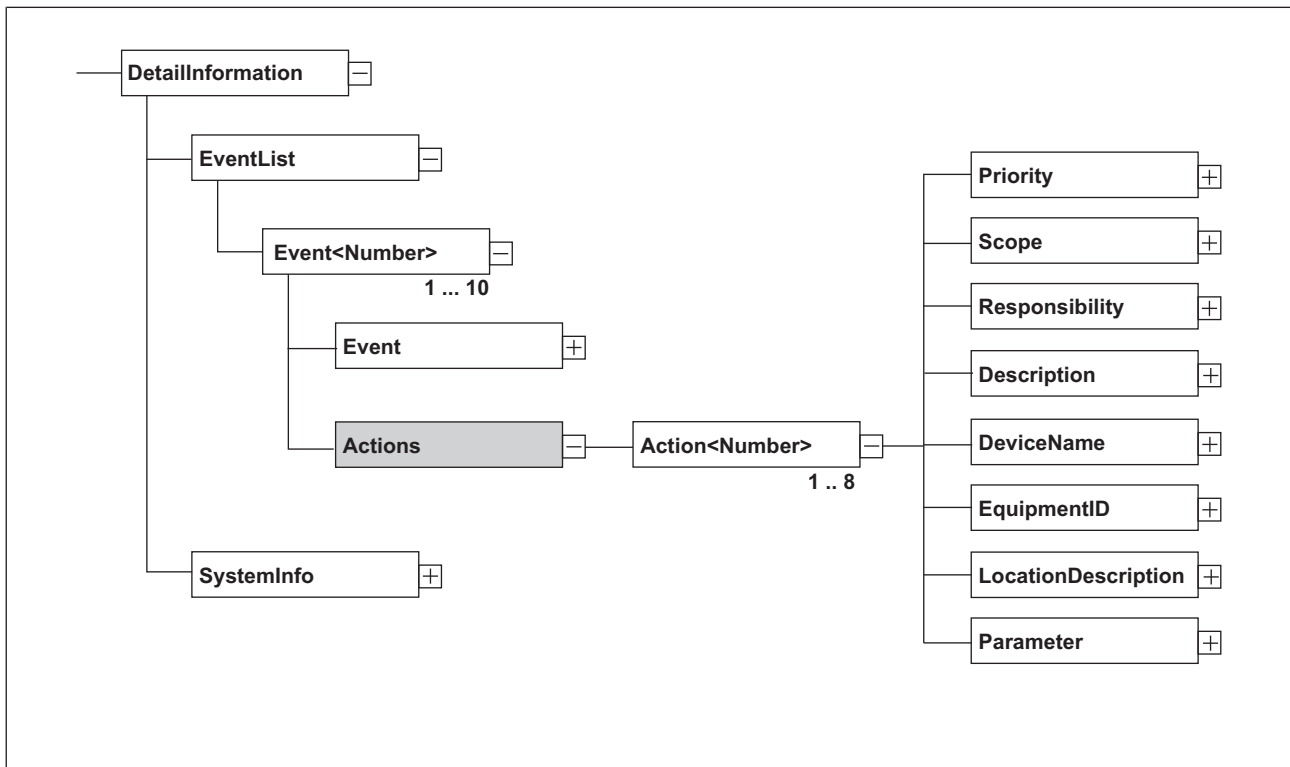


Fig.: Area <OPCProjectName>/<DeviceProjectName>/DetailInformation/EventList/Event<Number>/Actions/

Node name	Meaning
Actions	Details of the actions of an event
Path: <OPCProjectName>/<DeviceProjectName>/DetailInformation/EventList/Event<Number>/Actions/	
Action<Number>	Number of the action (Node name e.g. "Action 5")
Path: <OPCProjectName>/<DeviceProjectName>/DetailInformation/EventList/Event<Number>/Actions/Action<Number>/	
Priority	Priority Data type: String
FQN: <OPCProjectName>/<DeviceProjectName>/DetailInformation/EventList/Event<Number>/Actions/Action<Number>/Priority	
Scope	Scope Data type: String
FQN: <OPCProjectName>/<DeviceProjectName>/DetailInformation/EventList/Event<Number>/Actions/Action<Number>/Scope	
Responsibility	Responsibility Data type: String
FQN: <OPCProjectName>/<DeviceProjectName>/DetailInformation/EventList/Event<Number>/Actions/Action<Number>/Responsibility	

Node name		Meaning
	Description	Description Data type: String
	FQN: <OPCProjectName>/<DeviceProjectName>/DetailInformation/EventList/Event<Number>/Actions/Action<Number>/Description	
	DeviceName	Device name Data type: String
	FQN: <OPCProjectName>/<DeviceProjectName>/DetailInformation/EventList/Event<Number>/Actions/Action<Number>/DeviceName	
	EquipmentID	Equipment identifier (Equip-ID) Data type: String
	FQN: <OPCProjectName>/<DeviceProjectName>/DetailInformation/EventList/Event<Number>/Actions/Action<Number>/EquipmentID	
	LocationDescription	Location description Data type: String
	FQN: <OPCProjectName>/<DeviceProjectName>/DetailInformation/EventList/Event<Number>/Actions/Action<Number>/LocationDescription	
	Parameter	More precise information on where an event occurred, e.g. information on the input terminal ("E3.0") Data type: String
	FQN: <OPCProjectName>/<DeviceProjectName>/DetailInformation/EventList/Event<Number>/Actions/Action<Number>/Parameter	

5.2.5.3 PNOZmulti/PNOZmulti Mini/PNOZmulti 2: SystemInfo

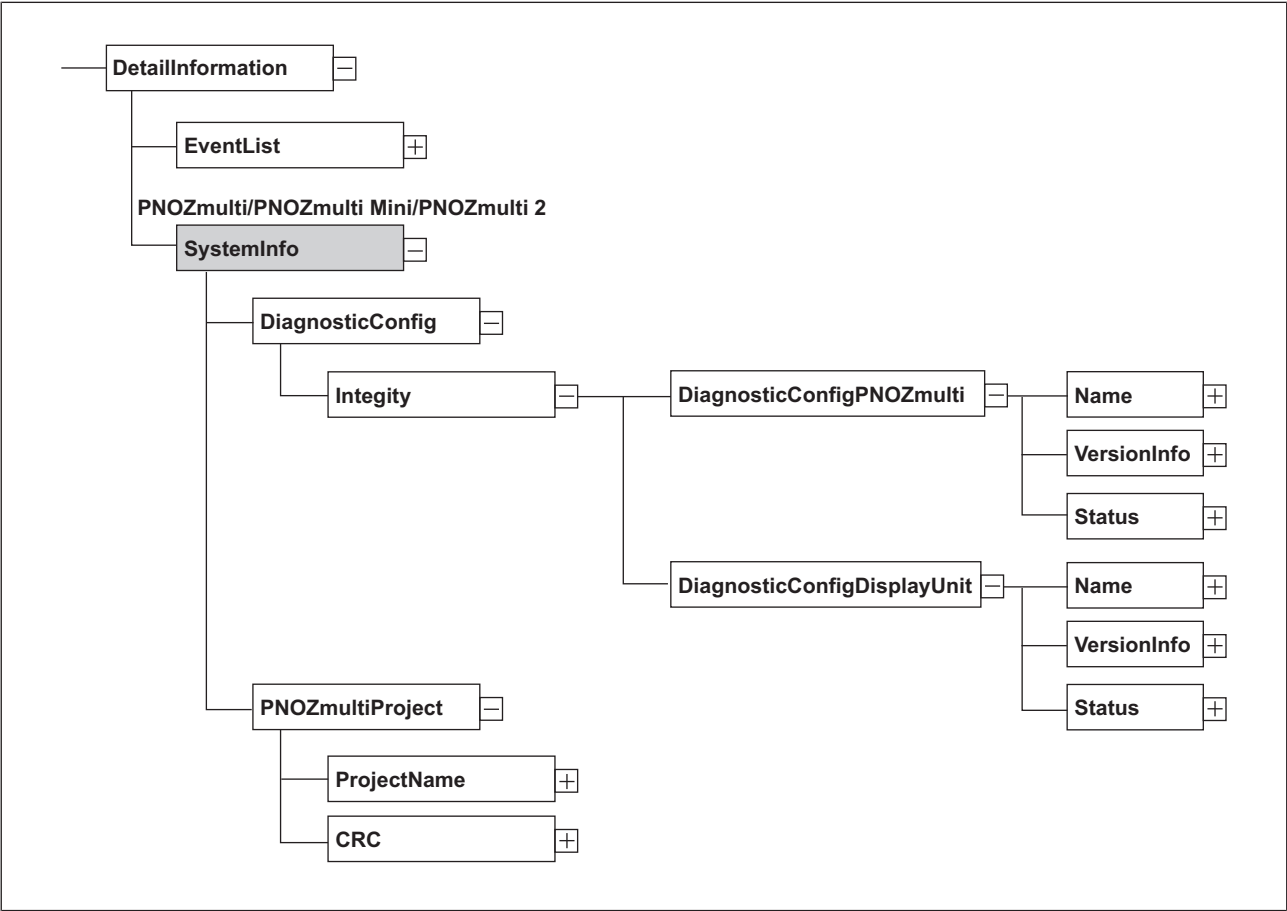


Fig.: Area <OPCProjectName>/<DeviceProjectName>/DetailInformation/SystemInfo/

Node name	Meaning
SystemInfo	Area containing version information on the PNOZmulti project and diagnostic configuration
Path: <OPCProjectName>/<DeviceProjectName>/DetailInformation/SystemInfo/	

Area containing information on the diagnostic configuration

Node name	Meaning
DiagnosticConfig	Area containing information on the diagnostic configuration
Path:<OPCProjectName>/<DeviceProjectName>/DetailInformation/SystemInfo/DiagnosticConfig/	
Integrity	Area containing all PNOZmulti program sections
Path: <OPCProjectName>/<DeviceProjectName>/DetailInformation/SystemInfo/DiagnosticConfig/Integrity/	
DiagnosticConfigPNOZmulti	Area containing all the diagnostic configuration data belonging to the PNOZmulti

Node name		Meaning	
Path: <OPCProjectName>/<DeviceProjectName>/DetailInformation/SystemInfo/DiagnosticConfig/Integrity/DiagnosticConfigPNOZmulti/			
	Name	Name of the project to which the program belongs Data type: String	
	FQN: <OPCProjectName>/<DeviceProjectName>/DetailInformation/SystemInfo/DiagnosticConfig/Integrity/DiagnosticConfigPNOZmulti/Name		
	VersionInfo	Time stamp on the download Data type: String	
	FQN: <OPCProjectName>/<DeviceProjectName>/DetailInformation/SystemInfo/DiagnosticConfig/Integrity/DiagnosticConfigPNOZmulti/VersionInfo		
	Status	Integrity information on the diagnostic configuration Data type: Byte	
		Display	Meaning
		0	The diagnostic configuration is in order.
		1	The diagnostic information belongs to another project.
		2	There is no information on the current diagnostic configuration.
		4	The diagnostic information is obsolete.
	FQN: <OPCProjectName>/<DeviceProjectName>/DetailInformation/SystemInfo/DiagnosticConfig/Integrity/DiagnosticConfigPNOZmulti/Status		
DiagnosticConfigDisplayUnit Area containing all the diagnostic configuration data belonging to the visualisation device			
Path: <OPCProjectName>/<DeviceProjectName>/DetailInformation/SystemInfo/DiagnosticConfig/Integrity/DiagnosticConfigDisplayUnit			
	Name	Name of the project to which the program belongs Data type: String	
	FQN: <OPCProjectName>/<DeviceProjectName>/DetailInformation/SystemInfo/DiagnosticConfig/Integrity/DiagnosticConfigDisplayUnit/Name		
	VersionInfo	Time stamp on the download Data type: String	
	FQN: <OPCProjectName>/<DeviceProjectName>/DetailInformation/SystemInfo/DiagnosticConfig/Integrity/DiagnosticConfigDisplayUnit/VersionInfo		

Node name		Meaning	
	Status	Integrity information on the diagnostic configuration Data type: String	
		Display	Meaning
		0	The diagnostic configuration is in order.
		1	The diagnostic information belongs to another project.
		2	There is no information on the current diagnostic configuration.
		4	The diagnostic information is obsolete.
	FQN:<OPCProjectName>/<DeviceProjectName>/DetailInformation/SystemInfo/DiagnosticConfig/Integrity/DiagnosticConfigDisplayUnit/Status		

Area containing information on the PNOZmulti program

Node name		Meaning
PNOZmultiProject		Information on the PNOZmulti program
Path: <OPCProjectName>/<DeviceProjectName>/DetailInformation/SystemInfo/PNOZmultiProject/		
	ProjectName	Name of the project to which the PNOZmulti program belongs Data type: String
	FQN: <OPCProjectName>/<DeviceProjectName>/DetailInformation/SystemInfo/PNOZmultiProject/Name	
	CRC	Check sum of the PNOZmulti program Data type: String
	FQN: <OPCProjectName>/<DeviceProjectName>/DetailInformation/SystemInfo/PNOZmultiProject/CRC	

5.2.5.4 **PMC: SystemInfo**

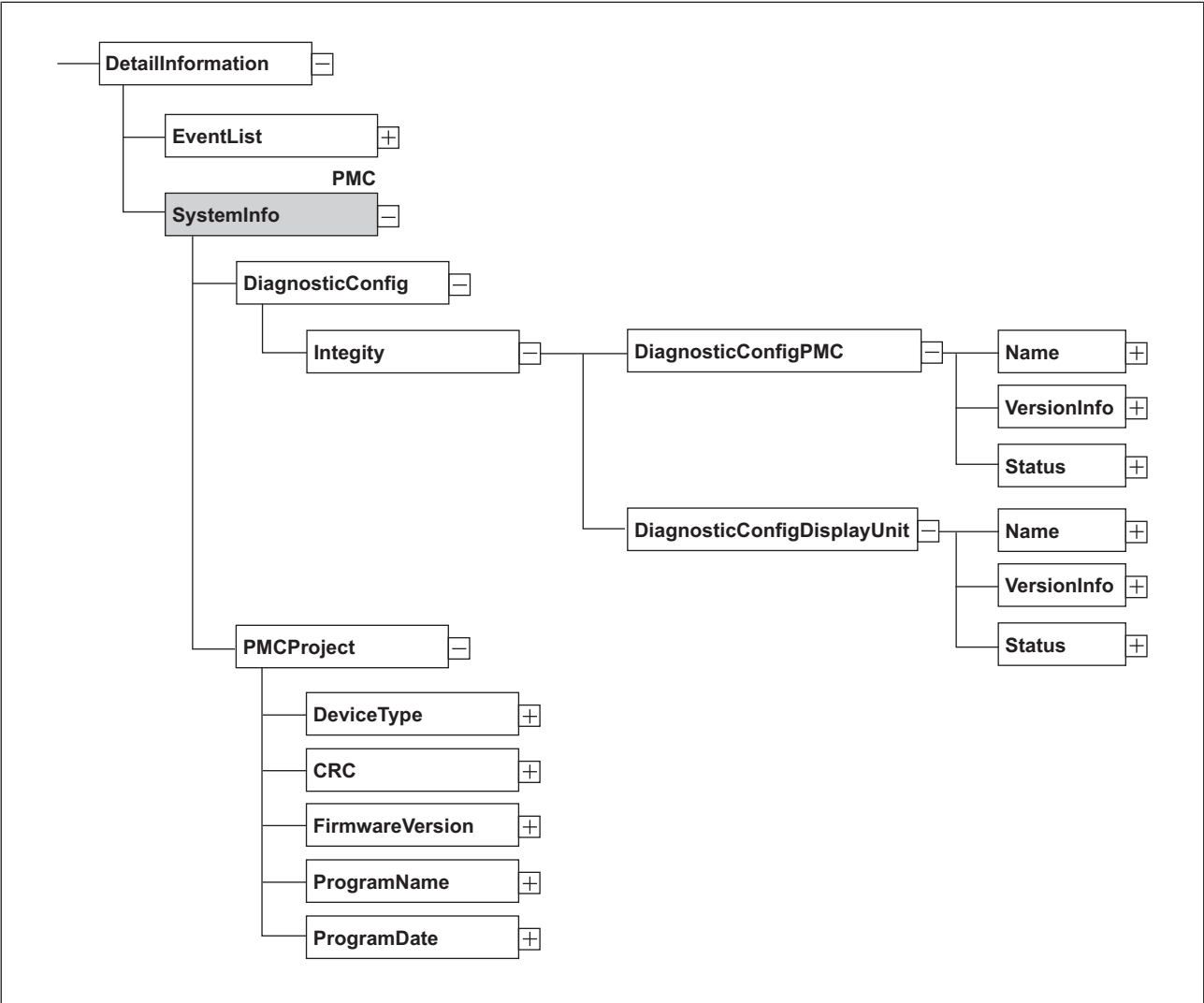


Fig.: Area <OPCProjectName>/<DeviceProjectName>/DetailInformation/SystemInfo/

Node name	Meaning
SystemInfo	Area containing version information on the PMC project and diagnostic configuration
Path: <OPCProjectName>/<DeviceProjectName>/DetailInformation/SystemInfo/	

Area containing information on the diagnostic configuration

Node name	Meaning
DiagnosticConfig	Area containing information on the diagnostic configuration
Path: <OPCProjectName>/<DeviceProjectName>/DetailInformation/SystemInfo/DiagnosticConfig/	
Integrity	Area containing all PMC program sections

Node name		Meaning	
Path: <OPCProjectName>/<DeviceProjectName>/DetailInformation/SystemInfo/DiagnosticConfig/Integrity/			
DiagnosticConfigPMC		Area containing all the diagnostic configuration data belonging to the PMC project	
Path:<OPCProjectName>/<DeviceProjectName>/DetailInformation/SystemInfo/DiagnosticConfig/Integrity/DiagnosticConfigPMC/			
	Name	Name of the project to which the program belongs Data type: String	
	FQN:<OPCProjectName>/<DeviceProjectName>/DetailInformation/SystemInfo/DiagnosticConfig/Integrity/DiagnosticConfigPMC/Name		
	VersionInfo	Time stamp on the download Data type: String	
	FQN:<OPCProjectName>/<DeviceProjectName>/DetailInformation/SystemInfo/DiagnosticConfig/Integrity/DiagnosticConfigPMC/VersionInfo		
	Status	Integrity information on the diagnostic configuration Data type: String	
		Display	Meaning
		0	The diagnostic configuration is in order.
		1	The diagnostic information belongs to another project.
		2	There is no information on the current diagnostic configuration.
	FQN:<OPCProjectName>/<DeviceProjectName>/DetailInformation/SystemInfo/DiagnosticConfig/Integrity/DiagnosticConfigPMC/Status		
DiagnosticConfigDisplayUnit		Area containing all the diagnostic configuration data belonging to the visualisation device	
Path:<OPCProjectName>/<DeviceProjectName>/DetailInformation/SystemInfo/DiagnosticConfig/Integrity/DiagnosticConfigDisplayUnit/			
	Name	Name of the project to which the program belongs Data type: String	
	FQN:<OPCProjectName>/<DeviceProjectName>/DetailInformation/SystemInfo/DiagnosticConfig/Integrity/DiagnosticConfigDisplayUnit/Name		
	VersionInfo	Time stamp on the download Data type: String	
	FQN:<OPCProjectName>/<DeviceProjectName>/DetailInformation/SystemInfo/DiagnosticConfig/Integrity/DiagnosticConfigDisplayUnit/VersionInfo		

Node name		Meaning	
	Status	Integrity information on the diagnostic configuration Data type: String	
		Display	Meaning
		0	The diagnostic configuration is in order.
		1	The diagnostic information belongs to another project.
		2	There is no information on the current diagnostic configuration.
FQN: <OPCProjectName>/<DeviceProjectName>/DetailInformation/SystemInfo/DiagnosticConfig/Integrity/DiagnosticConfigDisplayUnit/Status			

Area containing information on the PMC project

Node name		Meaning
PMCProject		Information on the PMC project
Path: <OPCProjectName>/<DeviceProjectName>/DetailInformation/SystemInfo/PMCProject/		
	DeviceType	Name of device Data type: String
	FQN: <OPCProjectName>/<DeviceProjectName>/DetailInformation/SystemInfo/PMCProject/DeviceType	
	CRC	Check sum of the program Data type: String
	FQN: <OPCProjectName>/<DeviceProjectName>/DetailInformation/SystemInfo/PMCProject/CRC	
	FirmwareVersion	Firmware version of the device Data type: String
	FQN: <OPCProjectName>/<DeviceProjectName>/DetailInformation/SystemInfo/PMCProject/FirmwareVersion	
	ProgramName	Name of program Data type: String
	FQN: <OPCProjectName>/<DeviceProjectName>/DetailInformation/SystemInfo/PMCProject/ProgramName	
	ProgramDate	Date of program Data type: String
	FQN: <OPCProjectName>/<DeviceProjectName>/DetailInformation/SystemInfo/PMCProject/ProgramDate	

5.2.6 DeviceProjectName/Generic

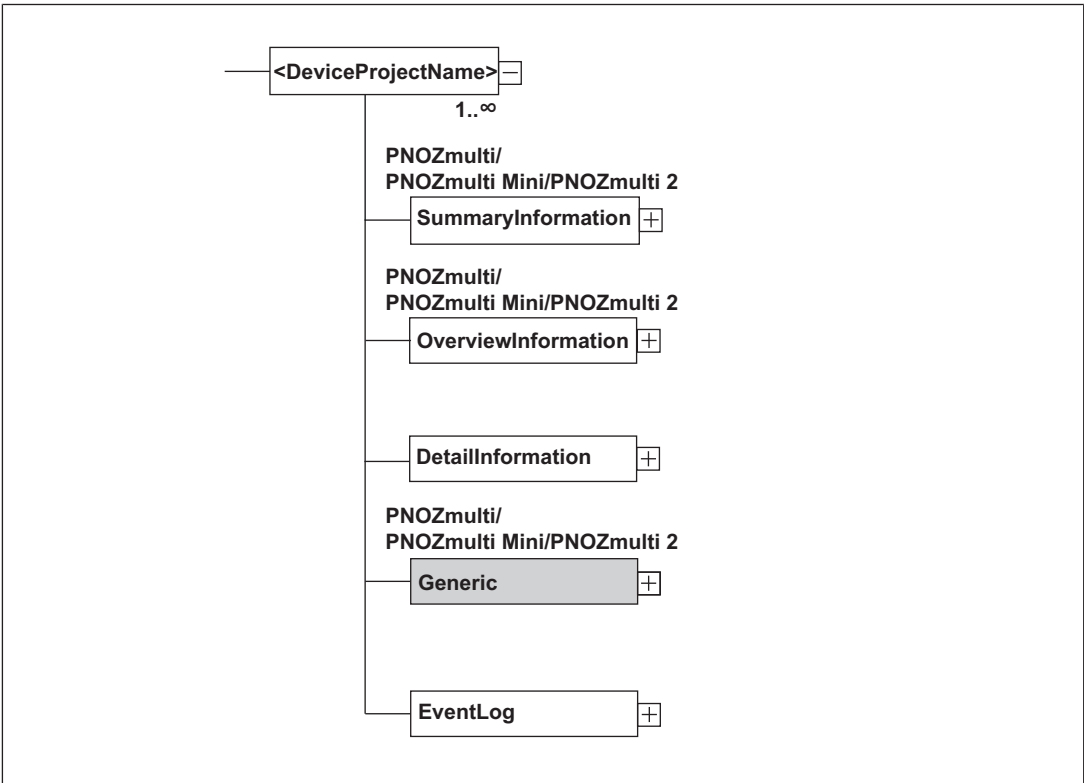


Fig.: Area `<OPCProjectName>/<DeviceProjectName>/Generic/`

Node name	Meaning	
Generic	Area containing device-specific process data	
	PNOZmulti, PNOZmulti Mini, PNOZmulti 2	The process data can be accessed via the FQN see (PNOZmulti/PNOZmulti Mini/PNOZmulti 2: Generic)
Path: <code><OPCProjectName>/<DeviceProjectName>/Generic/</code>		



INFORMATION

From "Generic", it is impossible to navigate the namespace using a browser of an OPC UA Client. Navigation is only possible by entering a path.

5.2.6.1

PNOZmulti/PNOZmulti Mini/PNOZmulti 2: Generic

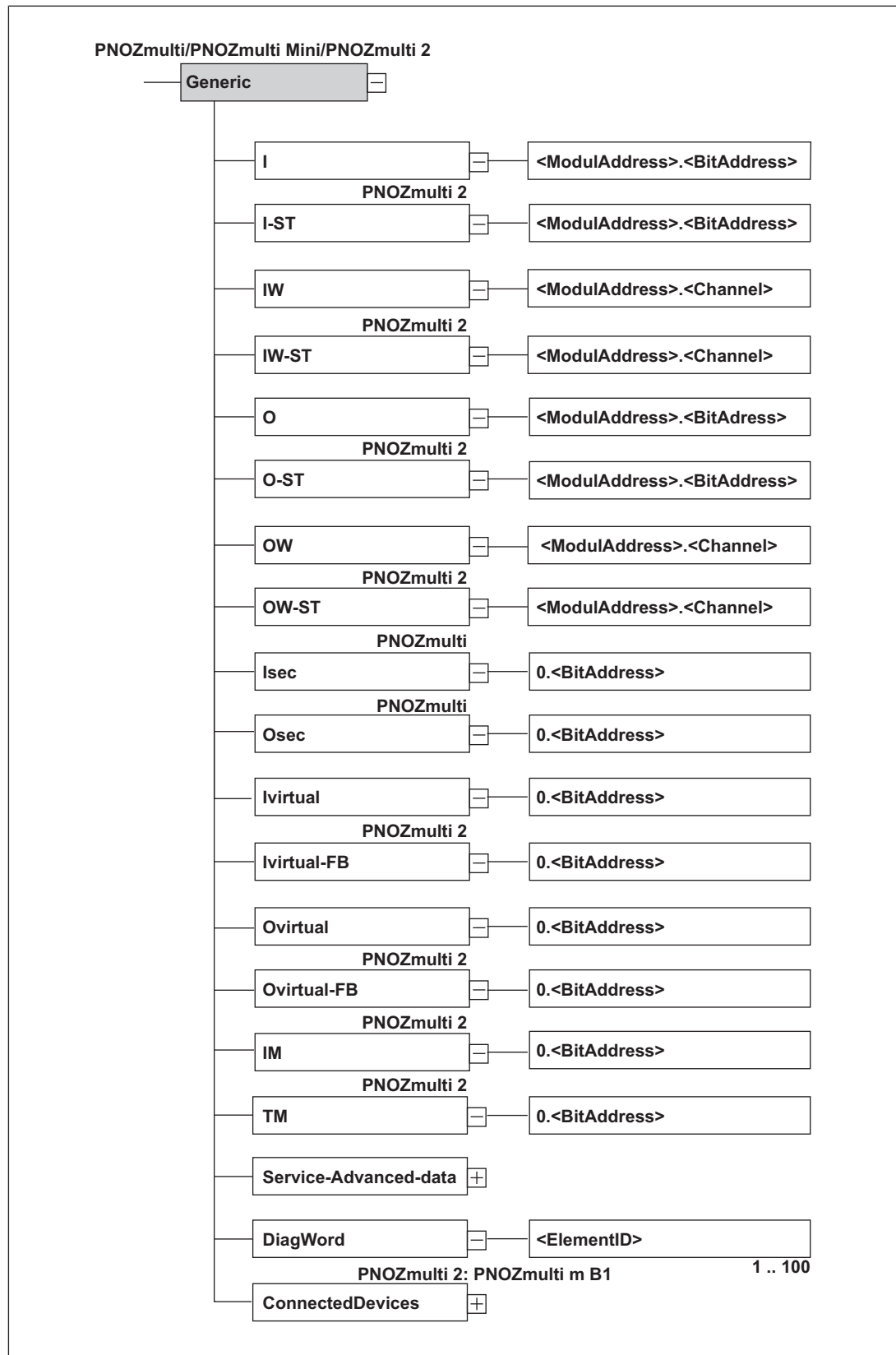


Fig.: Area <OPCProjectName>/<DeviceProjectName>/Generic/

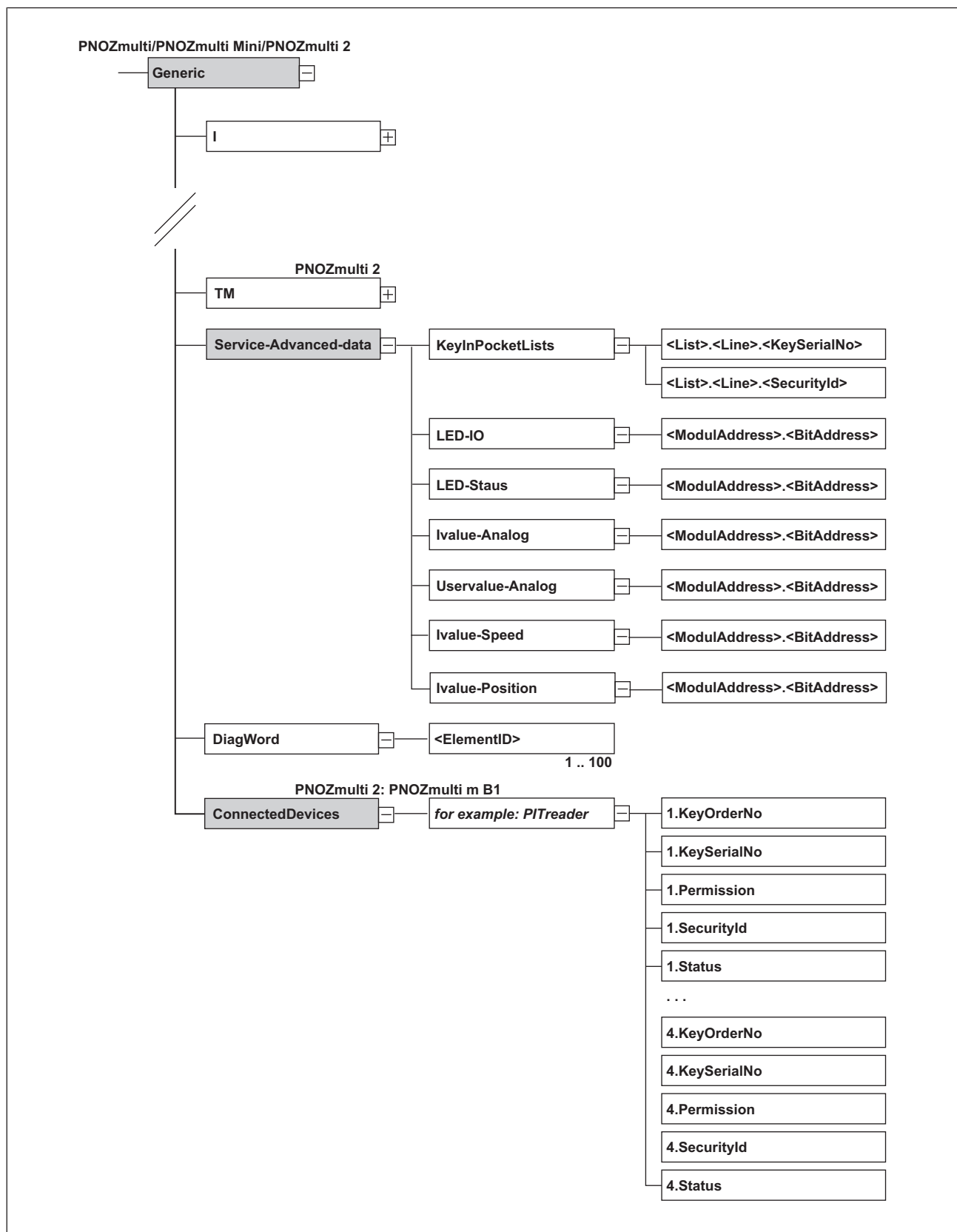


Fig.: Area <OPCProjectName>/<DeviceProjectName>/Generic/Service-Advanced-data and ConnectedDevices

Node name		Meaning
I		Input bits
Path: <OPCProjectName>/<DeviceProjectName>/Generic/I/		
	<ModuleAddress>.<BitAddress>	► PNOZmulti, PNOZmulti Mini and PNOZmulti 2 addresses – 0.<0 .. 19> e.g.: "I/0.0" ► Addresses for PNOZmulti and PNOZmulti Mini only: – <-6 .. -1>.<0 .. 31> e.g.: "I/-6.15" – <1 .. 8>.<0 .. 15> e.g.: "I/5.10" ► Addresses PNOZmulti 2 only: – <-6 .. -1>.<0 .. 127> – <1 .. 12>.<0 .. 31> e.g.: "I/7.15"
		Data type: Boolean
FQN: <OPCProjectName>/<DeviceProjectName>/Generic/I/<ModuleAddress>.<BitAddress>		
I-ST		ST input bits
Path: <OPCProjectName>/<DeviceProjectName>/Generic/I-ST/		
	<ModuleAddress>.<BitAddress>	Addresses PNOZmulti 2 only: <101 .. 112>.<0 .. 31> e.g. "I-ST/110.15"
		Data type: Boolean
FQN: <OPCProjectName>/<DeviceProjectName>/Generic/I-ST/<ModuleAddress>.<BitAddress>		
IW		Input words
Path: <OPCProjectName>/<DeviceProjectName>/Generic/IW/		
	<ModuleAddress>.<Channel>	► PNOZmulti, PNOZmulti Mini and PNOZmulti 2 addresses: – <-6 .. -1>.<0 .. 1> e.g. "IW/-1.0" – 0.<0 .. 1> e.g. "IW/0.0"
		Data type: I16
FQN: <OPCProjectName>/<DeviceProjectName>/Generic/IW/<ModuleAddress>.<Channel>		
O		Output bits
Path: <OPCProjectName>/<DeviceProjectName>/Generic/O/		

Node name		Meaning
<ModuleAddress>.<BitAddress>		► PNOZmulti, PNOZmulti Mini and PNOZmulti 2 addresses – <-6 .. -1>.<0 .. 31> e.g.: O/-5.15 – 0.<0 .. 5> e.g.: "O/0.0" ► Addresses for PNOZmulti and PNOZmulti Mini only: – <1 .. 8>.<0 .. 15> e.g.: "O/5.10" ► Addresses PNOZmulti 2 only: – <1 .. 12>.<0 .. 31> e.g.: "O/7.7"
		Data type: Boolean
		FQN: <OPCProjectName>/<DeviceProjectName>/Generic/O/<ModuleAddress>.<BitAddress>
O-ST		ST output bits
		Path: <OPCProjectName>/<DeviceProjectName>/Generic/O-ST/
<ModuleAddress>.<BitAddress>		Addresses PNOZmulti 2 only: <101 .. 112>.<0 .. 31> e.g. "O-ST/101.7"
		Data type: Boolean
		FQN: <OPCProjectName>/<DeviceProjectName>/Generic/O-ST/<ModuleAddress>.<BitAddress>
OW		Output words
		Path: <OPCProjectName>/<DeviceProjectName>/Generic/OW/
<ModuleAddress>.<Channel>		PNOZmulti, PNOZmulti Mini and PNOZmulti 2 addresses: <-6 .. -1>.<0 .. 1> e.g. "OW/-3.0"
		Data type: I16
		FQN: <OPCProjectName>/<DeviceProjectName>/Generic/OW/<ModuleAddress>.<Channel>
Isec		Input bit of a safe Ethernet connection
		Path: <OPCProjectName>/<DeviceProjectName>/Generic/Isec/
0.<BitAddress>		Addresses PNOZmulti only: 0.<0 .. 47> e.g.: "Isec/0.40"
		Data type: Boolean
		FQN: <OPCProjectName>/<DeviceProjectName>/Generic/Isec/0.<BitAddress>
Osec		Output bit of a safe Ethernet connection
		Path: <OPCProjectName>/<DeviceProjectName>/Generic/Osec/

Node name		Meaning
	0.<BitAddress>	Addresses PNOZmulti only: 0.<0 .. 47> e.g. "Osec/0.7"
		Data type: Boolean
	FQN: <OPCProjectName>/<DeviceProjectName>/Generic/Osec/0.<BitAddress>	
lvirtual		Input bits
Path: <OPCProjectName>/<DeviceProjectName>/Generic/lvirtual/		
	0.<BitAddress>	PNOZmulti, PNOZmulti Mini and PNOZmulti 2 addresses: 0.<0 .. 127> e.g. "lvirtual/0.125"
		Please note for PNOZmulti 2: ▶ PNOZmulti m B0: Either lvirtual/Ovirtual or lvirtual-FB/Ovirtual-FB may be displayed. ▶ PNOZmulti m B1: lvirtual/Ovirtual and lvirtual-FB/Ovirtual-FB can be displayed simultaneously.
	Data type: Boolean	
FQN: <OPCProjectName>/<DeviceProjectName>/Generic/lvirtual/0.<BitAddress>		
lvirtual-FB		Input bits of a fieldbus
Path: <OPCProjectName>/<DeviceProjectName>/Generic/lvirtual-FB/		
	0.<BitAddress>	Addresses PNOZmulti 2 only: 0.<0 .. 127> e.g. "lvirtual-FB/0.63"
		Please note for PNOZmulti 2: ▶ PNOZmulti m B0: Either lvirtual/Ovirtual or lvirtual-FB/Ovirtual-FB may be displayed. ▶ PNOZmulti m B1: lvirtual/Ovirtual and lvirtual-FB/Ovirtual-FB can be displayed simultaneously.
	Data type: Boolean	
FQN: <OPCProjectName>/<DeviceProjectName>/Generic/lvirtual-FB/0.<BitAddress>		
Ovirtual		Output bits
Path: <OPCProjectName>/<DeviceProjectName>/Generic/Ovirtual/		

Node name		Meaning
	0.<BitAddress>	PNOZmulti, PNOZmulti Mini and PNOZmulti 2 addresses: 0.<0 .. 127> e.g. "Ovirtual/0.127"
		Please note for PNOZmulti 2: ▶ PNOZmulti m B0: Either lvirtual/Ovirtual or lvirtual-FB/Ovirtual-FB may be displayed. ▶ PNOZmulti m B1: lvirtual/Ovirtual and lvirtual-FB/Ovirtual-FB can be displayed simultaneously.
		Data type: Boolean
	FQN: <OPCProjectName>/<DeviceProjectName>/Generic/Ovirtual/0.<BitAddress>	
Ovirtual-FB		Output bits on a fieldbus
Path: <OPCProjectName>/<DeviceProjectName>/Generic/Ovirtual-FB/		
	0.<BitAddress>	Addresses PNOZmulti 2 only: 0.<0 .. 127> e.g. "Ovirtual-FB/0.27"
		Please note for PNOZmulti 2: ▶ PNOZmulti m B0: Either lvirtual/Ovirtual or lvirtual-FB/Ovirtual-FB may be displayed. ▶ PNOZmulti m B1: lvirtual/Ovirtual and lvirtual-FB/Ovirtual-FB can be displayed simultaneously.
		Data type: Boolean
	FQN: <OPCProjectName>/<DeviceProjectName>/Generic/Ovirtual-FB/0.<BitAddress>	
IM		Configurable input/output bits
Path: <OPCProjectName>/<DeviceProjectName>/Generic/IM/		
	0.<BitAddress>	▶ Addresses PNOZmulti 2 only: – 0.<0 .. 3> e.g. "IM/0.2" – 0.<16 .. 19> e.g. "IM/0.16"
		Data type: Boolean
	FQN:<OPCProjectName>/<DeviceProjectName>/Generic/IM/0.<BitAddress>	
TM		Configurable test pulse/auxiliary output bits
Path: <OPCProjectName>/<DeviceProjectName>/Generic/TM/		

Node name		Meaning
	0.<BitAddress>	Addresses PNOZmulti 2 only: 0.<20 .. 23> e.g. "TM/0.20"
		Data type: Boolean
	FQN: <OPCProjectName>/<DeviceProjectName>/Generic/TM/0.<BitAddress>	
Service-Advanced-data		Areas with additional diagnostic data
Path: <OPCProjectName>/<DeviceProjectName>/Generic/Service-Advanced-data/		
KeyInPocketLists		
Path: <OPCProjectName>/<DeviceProjectName>/Generic/Service-Advanced-data/KeyInPocketLists/		
	<List>.<Line>.<KeySerialNo>	Addresses: <1 .. 4>.<01 .. 20> e.g. "1.05.KeySerialNo"
		Data type: U32
	FQN:<OPCProjectName>/<DeviceProjectName>/Generic/Service-Advanced-data/KeyInPocketLists/ <List>.<Line>.<KeySerialNo>	
	<List>.<Line>.<SecurityId>	Addresses: <1 .. 4>.<01 .. 20> e.g. "1.05.SecurityId"
		Data type: U64
	FQN:<OPCProjectName>/<DeviceProjectName>/Generic/Service-Advanced-data/KeyInPocketLists/ <List>.<Line>.<SecurityID>	
LED-IO		Status of a module's I/O LEDs
Path: <OPCProjectName>/<DeviceProjectName>/Generic/Service-Advanced-data/LED-IO/		
	<ModuleAddress>.<BitAddress>	Addresses:
	Data type: U8	<0 .. 12>.<0 .. 31> e.g. "LED-IO/7.15"
	FQN:<OPCProjectName>/<DeviceProjectName>/Generic/Service-Advanced-data/LED-IO/ <ModuleAddress>.<BitAddress>	
LED status		Status of a module's status LEDs
Path: <OPCProjectName>/<DeviceProjectName>/Generic/Service-Advanced-data/LED-Status		
	<ModuleAddress>.<BitAddress>	Addresses:
	Data type: U8	<-6 .. 12>.<0 .. 7> e.g. "LED-Status/7.7"
	FQN:<OPCProjectName>/<DeviceProjectName>/Generic/Service-Advanced-data/LED-Status/ <ModuleAddress>.<BitAddress>	

Node name		Meaning
Ivalue-Analog	Numeric input value of an analogue module	
	Path: <OPCProjectName>/<DeviceProjectName>/Generic/Service-Advanced-data/Ivalue-Analog/	
	<ModuleAddress>.<BitAddress>	Addresses:
	Data type: Float	<1 .. 12>.<0 .. 3> e.g. "Ivalue-Analog/7.0"
FQN:<OPCProjectName>/<DeviceProjectName>/Generic/Service-Advanced-data/Ivalue-Analog/<ModuleAddress>.<BitAddress>		
Uvalue-Analog	Scaled input value of an analogue module	
	Path: <OPCProjectName>/<DeviceProjectName>/Generic/Service-Advanced-data/Uvalue-Analog/	
	<ModuleAddress>.<BitAddress>	Addresses:
	Data type: Float	<1 .. 12>.<1 .. 6> e.g. "Uvalue-Analog/7.6"
FQN:<OPCProjectName>/<DeviceProjectName>/Generic/Service-Advanced-data/Uvalue-Analog/<ModuleAddress>.<BitAddress>		
Ivalue-Speed	Input value (speed) of a motion monitoring module	
	<ModuleAddress>.<BitAddress>	Addresses:
	Data type: U32	<1 .. 12>.<1, 2> e.g. "Ivalue-Speed/7.1"
	FQN:<OPCProjectName>/<DeviceProjectName>/Generic/Service-Advanced-data/Ivalue-Speed/<ModuleAddress>.<BitAddress>	
Ivalue-Position	Input value (position) of a motion monitoring module	
	<ModuleAddress>.<BitAddress>	Addresses:
	Data type: U32	<1 .. 12>.<1, 2> e.g. "Ivalue-Position/7.2"
	FQN:<OPCProjectName>/<DeviceProjectName>/Generic/Service-Advanced-data/Ivalue-Position/<ModuleAddress>.<BitAddress>	
DiagWord		Area to display diagnostic data (e.g. operating status, error messages)
Path: <OPCProjectName>/<DeviceProjectName>/Generic/DiagWord/		
	<ElementID>	Element ID of a diagnostic word
		Data type: U16
	FQN: <OPCProjectName>/<DeviceProjectName>/Generic/DiagWord/<Element-ID>	
ConnectedDevices		Area for the diagnostic data of devices that are connected to a PNOZmulti m B1 (PNOZmulti 2). Note: The sequence of the variables depends on the options for displaying of the used PPC Client.
Path: <OPCProjectName>/<DeviceProjectName>/Generic/ConnectedDevices/		

Node name		Meaning
PITreader		Area for PITreader diagnostic data Diagnostic data of up to four PITreaders can be displayed.
Path: <OPCProjectName>/<DeviceProjectName>/Generic/ConnectedDevices/PITreader/		
	1.KeyOrderNo	Order no. of the transponder key of the 1st PITreader
	1.KeySerialNo	Serial no. of the transponder key of the 1st PITreader (see operating manual for the PITreader)
	1.Permission	Permission of the first PITreader
	1.SecurityId	Security ID of the first PITreader
	1.Status	Status of the first PITreader
	...	...
	4.KeyOrderNo	Order no. of the transponder key of the 4th PITreader
	4.KeySerialNo	Serial no. of the transponder key of the 4th PITreader (see operating manual for the PITreader)
	4.Permission	Permission of the fourth PITreader
	4.SecurityId	Security ID of the fourth PITreader
	4.Status	Status of the fourth PITreader
Path: <OPCProjectName>/<DeviceProjectName>/Generic/ConnectedDevices/PITreader/<Diagnostic datum such as 1.KeyOrderNo for example>		

**NOTICE**

For PNOZmulti/PNOZmulti Mini/PNOZmulti 2 and PDP67 devices, please note:

- Base units PNOZ m0p/m1p/m2p:
In the namespace, the area for the I/O data is limited to Bit 00 ... 15.
- Base units PNOZ m B0 and PNOZ m B1:
In the namespace, the entire range from Bit 00 ... 31 is available in the area for the I/O data.
- Left-hand expansion devices are identified by "-<No. of the expansion device>" (e.g. ModuleAddress-6.BitAddress15) and right-hand expansion devices with "+<No. of the expansion device>" (e.g. I/ModuleAddress7.BitAddress15).
- PDP67 devices:
The inputs and outputs are numbered consecutively (00 ... 31)
Example:
1st device: 00 ... 07
2nd device: 08 ... 15
3rd device: 16 ... 23
4th device: 25 ... 31

5.2.7 DeviceProjectName/EventLog

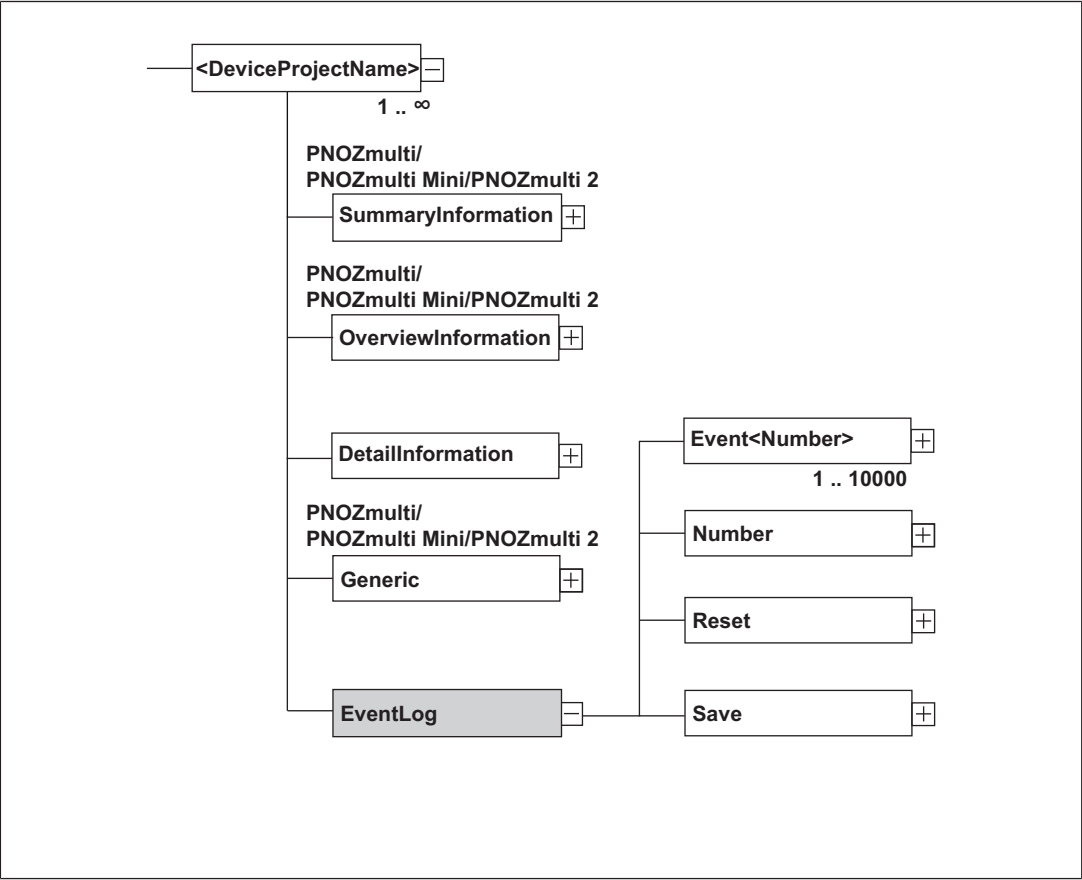


Fig.: Area <OPCProjectName>/<DeviceProjectName>/EventLog/

Node name	Meaning
EventLog	Area containing the entries of the event log The maximum number of events can be configured.
Path: <OPCProjectName>/<DeviceProjectName>/EventLog/	

Node name	Meaning		
Event <Number>	Access to an individual event in the event log e.g.: "Event 499"		
	Note: The entry for an event consists of a character string. The individual pieces of information in the character string are separated by <tab>.		
	Data type: String		
	Content of the character string:		
	IO	<<+>>	New event message
		<<->>	Event message cleared
	Timestamp	Time stamp	
	Priority	Priority	
	PriorityText	Description of priority	
	Scope	Scope	
	Description	Description	
	DeviceName	Device name	
	EquipmentID	Equipment identifier (Equip-ID)	
	EquipmentDescription	Location description	
	Parameter	More precise information on where an event occurred	
	Example: <<+>><tab>2015-03-20 14:45<tab>12<tab>Warning<tab>Scope1<tab>Event 2 occurred<tab>Device23<tab>Equip-ID7<tab>Location<tab>Terminal E3.0		
FQN: <OPCProjectName>/<DeviceProjectName>/EventLog/Event<Number>			
Number	Number of events in the event log Data type: UInt16		
FQN: <OPCProjectName>/<DeviceProjectName>/EventLog/Number			
Reset	Reset the event log The bit is set by the OPC Client and sent to the OPC Server. Data type: UInt16		
	0	Do not reset event log	
	1	Reset event log	
FQN: <OPCProjectName>/<DeviceProjectName>/EventLog/Reset			

Node name	Meaning
Save	Save the event log in an export file: ▶ The file name for the export file is set by the OPC Client and downloaded to the PVIS OPC Server. Please note: If an export file with the same name already exists, it will be overwritten. ▶ The file name can be entered with an absolute or relative path to the storage location. ▶ If no path is downloaded for the export file <i><DeviceProjectName></i> will be used as the default file name and the export file will be stored in the OPC Server's root directory "PVIS OPC Server UA". With a default installation this is: c:\Users\All Users\Pilz\PVISOPCDataUA\<DeviceProjectName>.log ▶ Data type: STRING
FQN: <OPCProjectName>/<DeviceProjectName>/EventLog/Save	

**INFORMATION**

From "EventLog", it is impossible to navigate the namespace using a browser. Navigation is only possible by entering a path.
Example of an FQN: "Namespace1/Project1/EventLog/Event 499"

6 Examples of application

6.1 Diagnostics for a PNOZmulti

6.1.1 Create diagnostic configuration

The diagnostic configuration is project-related, i.e. you will need to create a separate diagnostic configuration for each PNOZmulti project. This can be done in the diagnostic configurator in the PNOZmulti Configurator.

Steps:

- ▶ Select diagnostic language
- ▶ Define priorities
You will only need to perform this step if you wish to change the priorities pre-defined by Pilz or if you wish to define your own priorities.
- ▶ Define scopes
You will only need to perform this step if you wish to change the scopes pre-defined by Pilz or if you wish to define your own scopes.
- ▶ Define responsibilities
You will only need to perform this step if you wish to change the responsibilities pre-defined by Pilz or if you wish to define your own responsibilities.
- ▶ Activate expanded diagnostics
You need to activate the PVIS expanded diagnostics first if you wish to use the PVIS expanded diagnostics or if you want to create your own diagnostic types during the configuration of elements. The diagnostic types can be activated independently.
- ▶ Configure diagnostic types
You will only need to perform this step if you are creating an operand's diagnostic configuration or if you wish to create your own "Safety Device" diagnostic types. There are two different ways to configure the diagnostic types. Both methods are equally valid. The diagnostic types "Safety Device" and "Bit" can be modified or created in the diagnostic configurator or during the element configuration. A separate element is available for the "Bit" diagnostic type: the "diagnostic point". The diagnostic type must be assigned to the element to ensure that PVIS expanded diagnostics are performed for an element used in the configuration.
- ▶ Define device name and Equip-ID
You will only need to perform this step if your own device names are to be displayed during device diagnostics rather than the device names pre-defined by Pilz, or if the devices' Equip-IDs are also to be displayed.

The diagnostic configuration must then be downloaded to the PNOZmulti and to the display unit.

6.1.2 Download the diagnostic configuration to the PNOZmulti

Once the diagnostic configuration has been created it will need to be downloaded to the PNOZmulti and diagnostic unit. The diagnostic configuration is downloaded to the PNOZmulti with the project.

The diagnostic configuration is transferred to the diagnostic device either via the PASvisu Builder or by copying it manually. Please refer to the operating manual of your display unit for information on which method you need to use.

Once the diagnostic configuration has been transferred to the PNOZmulti, diagnostics are run automatically on the PNOZmulti.

► Support

Technical support is available from Pilz round the clock.

Americas

Brazil

+55 11 97569-2804

Canada

+1 888 315 7459

Mexico

+52 55 5572 1300

USA (toll-free)

+1 877-PILZUSA (745-9872)

Asia

China

+86 400-088-3566

Japan

+81 45 471-2281

South Korea

+82 31 778 3300

Australia and Oceania

Australia

+61 3 95600621

New Zealand

+64 9 6345350

Europe

Austria

+43 1 7986263-444

Belgium, Luxembourg

+32 9 3217570

France

+33 3 88104003

Germany

+49 711 3409-444

Ireland

+353 21 4804983

Italy, Malta

+39 0362 1826711

Pilz develops environmentally-friendly products using ecological materials and energy-saving technologies. Offices and production facilities are ecologically designed, environmentally-aware and energy-saving. So Pilz offers sustainability, plus the security of using energy-efficient products and environmentally-friendly solutions.



We are represented internationally. Please refer to our homepage www.pilz.com for further details or contact our headquarters.

Headquarters: Pilz GmbH & Co. KG, Felix-Wankel-Straße 2, 73760 Ostfildern, Germany
Telephone: +49 711 3409-0, E-Mail: info@pilz.com, Internet: www.pilz.com

PILZ
THE SPIRIT OF SAFETY

CECE[®], CHRE[®], CMSE[®], INDUSTRIAL PI[®], Leansafe[®], Myzel[®], PAS4000[®], PAScal[®], PAScontig[®], Pilz[®], PIT[®], PMCPrimo[®], PMCProtego[®], PMCiendo[®], PMD[®], PMT[®], 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. We would point out that product features may vary from the details stated in this document, depending on the status at the time of publication and the scope of the equipment. We accept no responsibility for the validity, accuracy and entirety of the text and graphics presented in this information. Please contact our Technical Support if you have any questions.