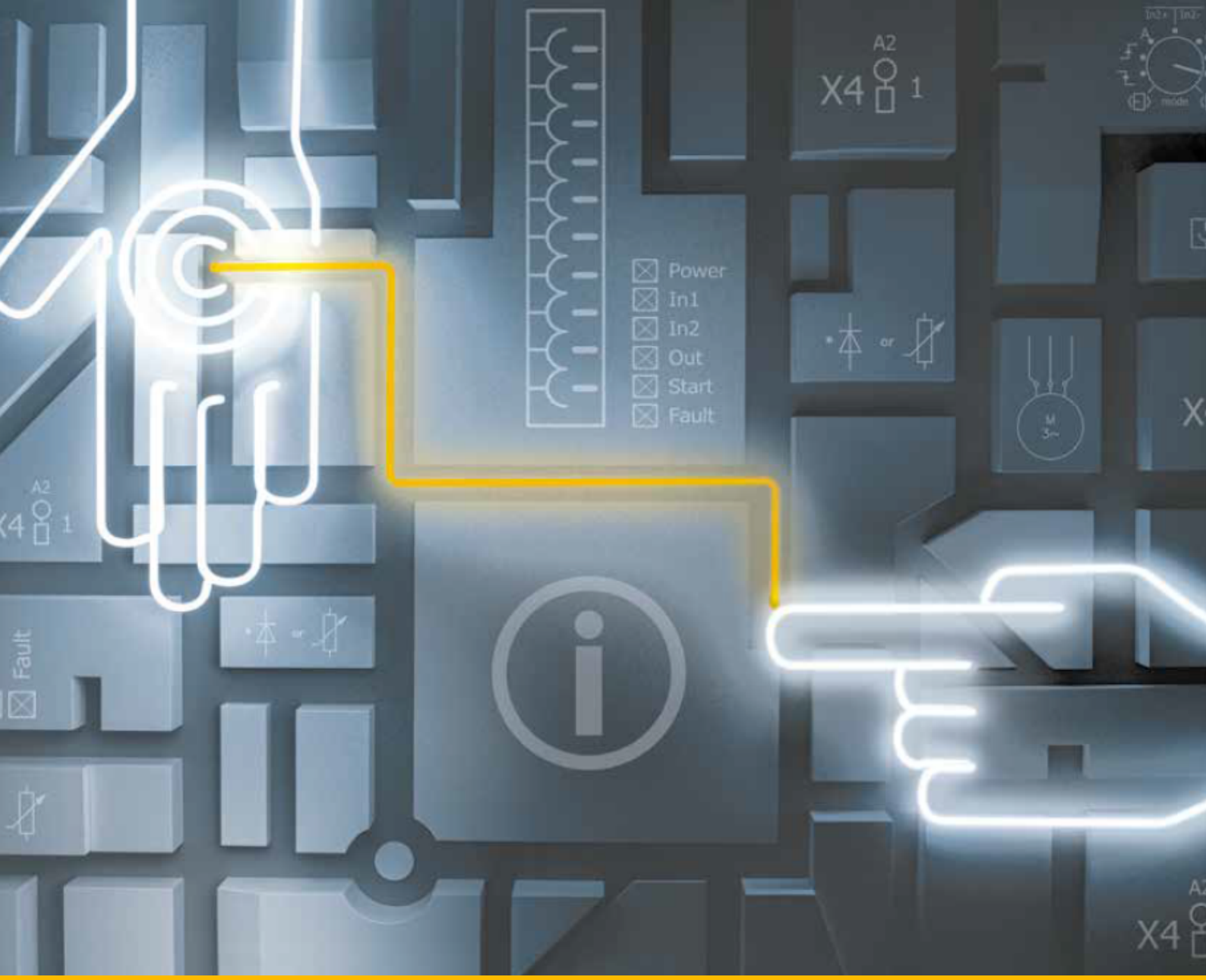

PNOZmulti OPC Server

PILZ
THE SPIRIT OF SAFETY

Operating Manual-1007149-EN-01
- 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.

1 Introduction

1.1 Validity of documentation

This documentation is valid for the product PNOZmulti OPC Server. It is valid until new documentation is published.

1.2 Using 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.

2 Intended use

The product PNOZmulti OPC Server is a standardised software interface in accordance with the OPC UA standard.

The product PNOZmulti OPC Server is intended to make process and diagnostic data from a configurable safe small controller PNOZmulti 2 from Pilz available to an OPC UA Client. The OPC UA Client may be the OPC UA Client of the visualisation system PASvisu from Pilz, for example, or an OPC UA-enabled Client from a third-party manufacturer.

The product PNOZmulti OPC Server is intended for use with the following components:

- Configurable safe small controllers PNOZmulti 2 (2nd generation) with Ethernet interface:
  - PNOZ m B0
  - PNOZ m B0.1
  - PNOZ m B1
  - PNOZ m B1 Burner
- PNOZmulti Configurator from Version 11.4.0

  Software tool used to create a PNOZmulti project
  A PNOZmulti project contains the diagnostic configurations and the OPC data from a PNOZmulti 2 device.

  This operating manual does not cover operation of the PNOZmulti Configurator. For this please refer to the online help for the PNOZmulti Configurator.
- Visualisation software
  - PASvisu from Version 1.15.0

    Software tool from Pilz, used to visualise process and diagnostic data

    This operating manual does not cover the operation of the PASvisu (PASvisu Builder, PASvisu Server, PASvisu Client). For this please refer to the online help for PASvisu.
  - OPC UA-enabled visualisation software from a third-party manufacturer
- Operator and display devices for visualisation (OPC UA Clients)

  Suitable visualisation software must be available on the devices.

  Potential operator and display devices:
  - PMI v70<*x*>e from Pilz (e.g. PMI v704e, PMI v707e)

    The visualisation software PASvisu (PASvisu Server and Client) from Pilz is pre-installed on these operator terminals.
  - PMI v8 from Pilz (e.g. PMI v807, PMI v812, PMI v815)

    The visualisation software PASvisu (PASvisu Server and Client) from Pilz is pre-installed on these operator terminals.
  - Operator and display devices from third-party manufacturers

    You can either use the operator and display device from a third-party manufacturer with the visualisation software PASvisu from Pilz or an operator and display device from a third-party manufacturer with an OPC UA-enabled visualisation software from a third-party manufacturer.

When using operator and display devices from a third-party manufacturer in conjunction with the visualisation software PASvisu from Pilz, a 64 Bit version of the Microsoft Windows operating system must be installed on the device. Further information can be found in the PASvisu Readme.

2.1 Third-party manufacturer licence information

The product contains open source software, whose terms of use could further limit the product's application area. It is essential that you observe the third-party manufacturer licence information.

Further information is available in the documents "Third-party manufacturer licence information PNOZmulti OPC Server (Linux)" (document number 1007038) and "Third-party manufacturer licence information PNOZmulti OPC Server (Windows)" (document number 1007139) at www.pilz.com.

3 Security

The term "security" refers to information security in plants, systems, machines and networks. This includes protection from external attacks (remote access) as well as protection from attacks from insiders. Measures are taken that protect the integrity and confidentiality of sensitive data and guarantee plant/machine availability. So all measures that concern the integrity, availability, confidentiality and authenticity of data fall under the term "security" (see also the IEC 62443 series of standards).

Planning and implementation

Securing plants, systems, machines and networks against cyberthreats requires a holistic industrial security concept that is state of the art and is continuously developed further.
All plants, systems, machinery and networks that require protecting must be assessed individually. The assessment must take into account organisational as well as technical measures. If necessary, seek advice from Pilz Customer Support.
Planning and implementation should be carried out in accordance with a defined security process. For example, your security process should take the following into account:

- Note the regulatory specifications and requirements from the locally applicable safety and application standards with regard to protection from manipulation.
- Perform a risk analysis (e.g. in accordance with IEC 62443-3-2) and then plan the resulting security measures carefully.
- Where possible, counter the security risks using a multi-layered concept (e.g. defence-in-depth concept).
- Name the necessary security measures and document them.
- Check the security measures after implementation.
- Carry out a new risk analysis after every change and adapt the security measures if necessary.
- It is advisable to review the security measures regularly, for example annually.

Defence-in-depth concept

In accordance with IEC 62443-4-1, the defence-in-depth concept is an approach in which multiple layers of independent security methods are used.

The defence-in-depth concept is based on the following rationale:

- Every protective measure can be overcome, and probably will be overcome.
- A multi-layer structure (e.g. onion model) implements several concerted and coordinated measures.
- Due to independent defence mechanisms in each layer, successful exploitation of a weakness in one layer is prevented, or made more difficult, by a measure in another layer.
- Generally with the defence-in-depth concept, an attack is made more difficult because the attacker has to circumvent different, independent security measures, one after the other.

The standard series IEC 62443 distinguishes between the following three areas of responsibility:

- Manufacturer
- System integrator

- Operator

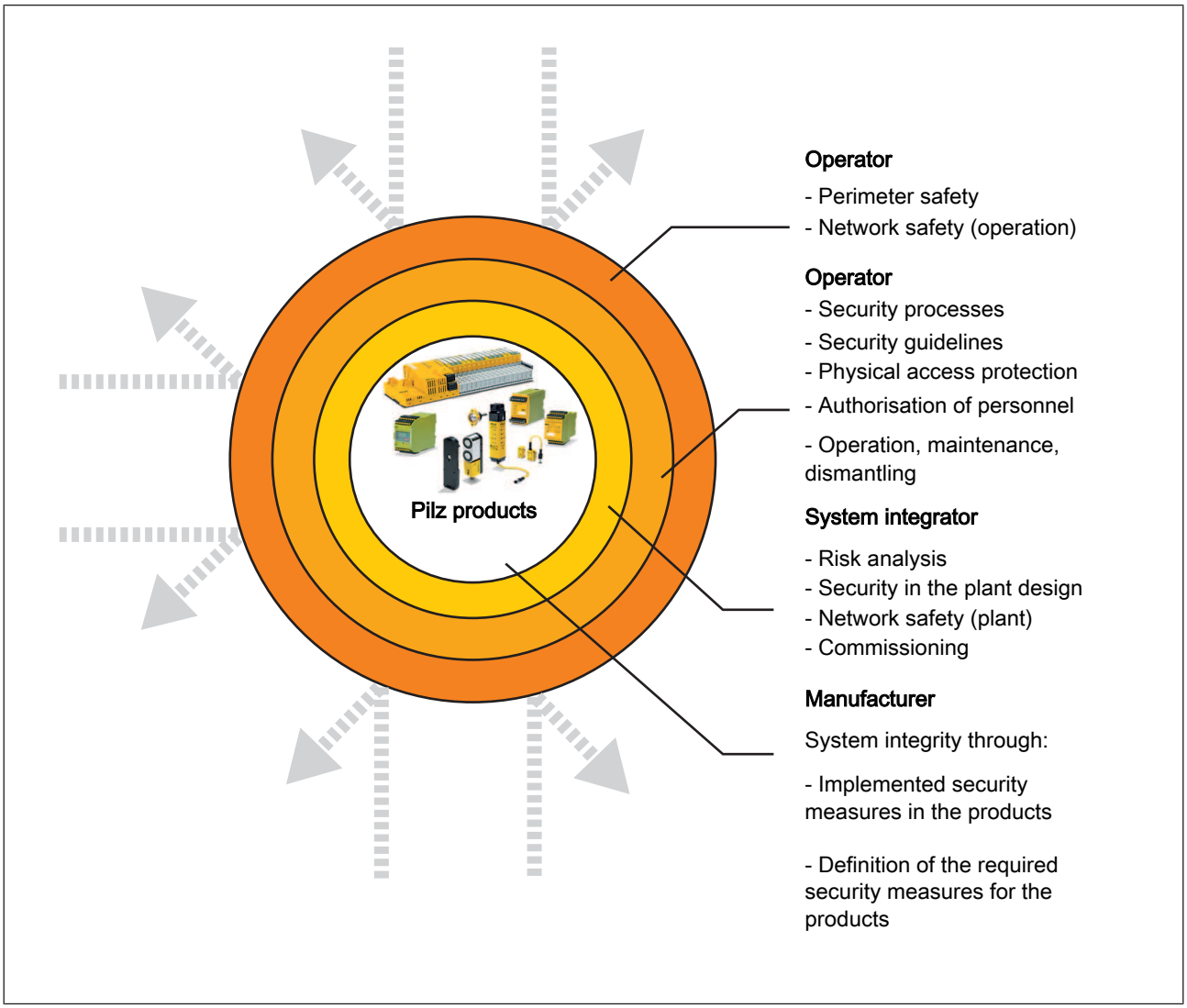

Fig.: Roles and responsibilities in the defence-in-depth onion model

Security incident management

Should the product have a security vulnerability, this will be published in a security advisory. You'll find the security advisories under www.pilz.com/security. You can also report any security problems there.

Implemented and required security measures

The section below describes the implemented and required security measures. To protect the product PNOZmulti OPC Server and any projects created using the OPC UA Server from cyber threats, it is essential that you use the implemented security measures and implement the required security measures.

3.1 Implemented security measures

- The OPC Server supports the following security profiles:
  - TransportSecurity - TLS 1.2 with PFS (Part 7 6.6.160)

  - SecurityPolicy [B] - Basic256Sha256 (Part 7 6.6.165)
- The OPC Server uses RSA certificates with a key length of 2048 bit and SHA2 (256 bit).
- An OPC Client can only establish a connection to the OPC Server if there is mutual trust; i.e. the exchanged certificates must be trustworthy.
- The OPC Server only accepts connections to OPC Clients that can prove their identity with a Client certificate registered on the product PNOZmulti OPC Server.
- The OPC Server only accepts encrypted data transfer in "Sign" or "Sign&Encrypt" mode.
- The OPC Server validates the validity period (time, date) of a Client certificate. If the validation fails, the connection is not established.
- The OPC Server validates the Application URI of a Client certificate. If the validation fails, the connection is not established.
- The OPC Server validates the SubjectAlternativeName of its own Server certificate. If the validation fails, the OPC Server is closed.

3.2 Required security measures

- Treat the Client and Server certificates and in particular the corresponding private keys with the same care as a password.
- Protect the Client and Server certificates and in particular the corresponding private keys from unauthorised access.
- Before you add the Server certificate to the list of trusted certificates on the OPC Client, check the Server certificate fingerprint. Only add the Server certificate to the list of trusted certificates if the Server certificate fingerprint on the OPC Client is identical to the Server certificate fingerprint.
- Before you add the Client certificate to the list of trusted certificates on the OPC Server, check the Client certificate fingerprint. Only add the Client certificate to the list of trusted certificates if the Client certificate fingerprint on the OPC Server is identical to the Client certificate fingerprint.

4 Function description

4.1 Functions of the PNOZmulti OPC Server

The product PNOZmulti OPC Server is an OPC UA Server in accordance with the OPC UA standard.

Operating principle

In normal operation, the OPC Server establishes a connection to the connected PNOZmulti 2 devices, reads the diagnostic data and makes it available in the namespace of the corresponding instance of the OPC Server. There, the diagnostic data can be retrieved and displayed by an OPC UA-enabled Client. For example, the diagnostic data includes event messages, status information and process data from the connected PNOZmulti 2 devices.

PNOZmulti projects with diagnostic configuration

Using the PNOZmulti Configurator, a PNOZmulti project must be created for each PNOZmulti 2 device that is connected to the OPC Server. The PNOZmulti project contains the diagnostic configuration. The diagnostic configuration must be downloaded to the OPC Server. There are various options, depending on the application:

- Download diagnostic configuration when using the visualisation software PASvisu from Pilz
  - If the PASvisu Server and the product PNOZmulti OPC Server are configured on the same device and started under the same user, the diagnostic configuration can be downloaded to the OPC Server via the project download of the PASvisu Builder.
  - If the PASvisu Server and the product PNOZmulti OPC Server are configured on different devices, the diagnostic configuration must be downloaded to the product PNOZmulti OPC Server manually.
- Using the visualisation software from a third-party manufacturer

  When using the visualisation software from a third-party manufacturer, the diagnostic configuration must be downloaded to the product PNOZmulti OPC Server manually.

OPC Client

The OPC Client retrieves the data (e.g. event messages, status information, process data) belonging to the connected PNOZmulti 2 devices from the OPC Server and displays it in its visualisation environment. The OPC Client may be the OPC Client of the visualisation system PASvisu from Pilz, for example, or the visualisation software from a third-party manufacturer.

Installing the product PNOZmulti OPC Server

The OPC Server is pre-installed on the relevant operator terminal from Pilz (e.g. PMI v7, PMI v8). The OPC Server can also be installed on devices with the Windows operating system.

Further information is available under System requirements [📖 28].

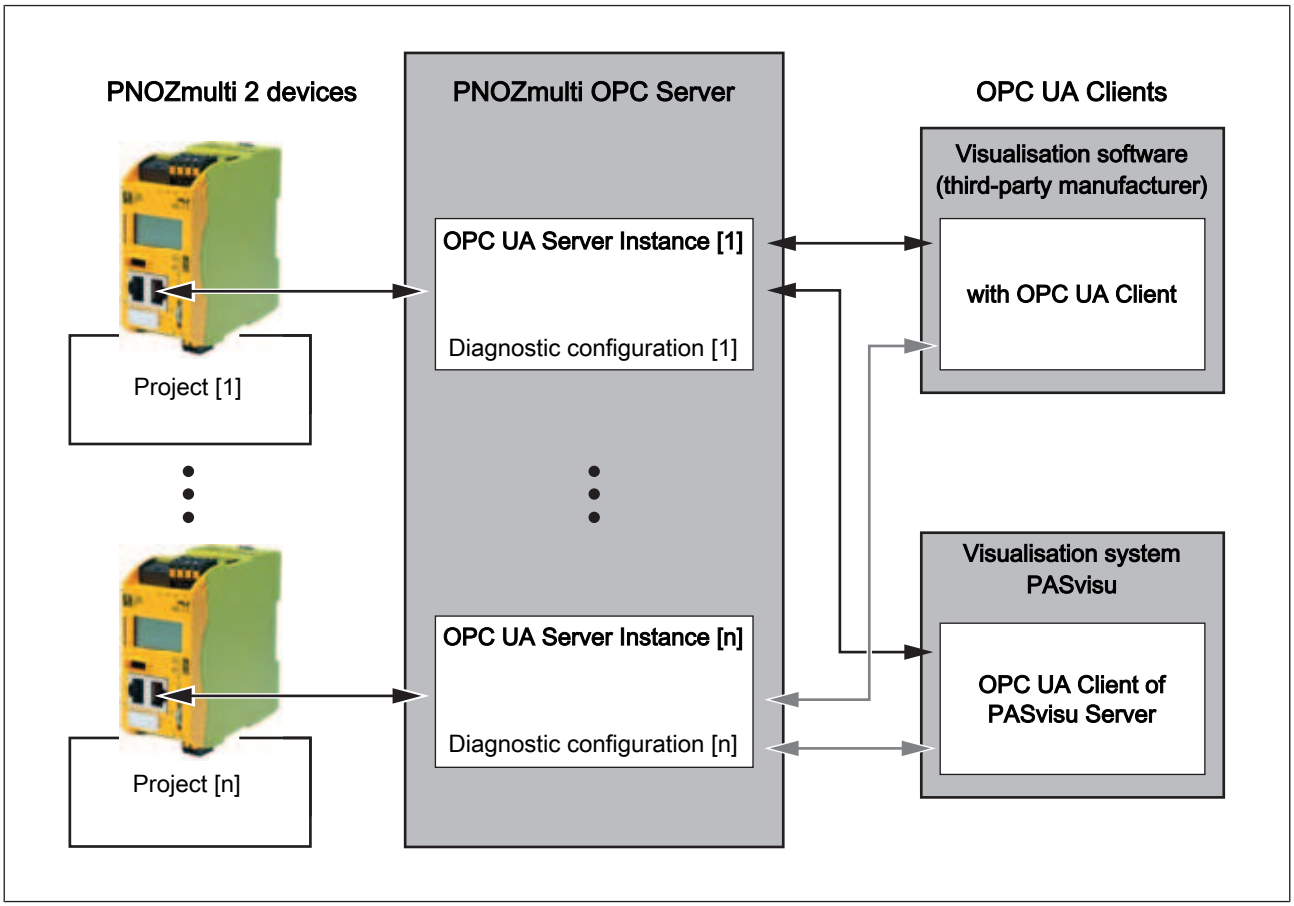

Fig.: Diagnostics with the PNOZmulti OPC Server from Pilz (principle)

4.2 OPC UA properties

The product PNOZmulti OPC Server supports the following OPC UA properties for access from OPC UA Clients:

- Supports the UA binary protocol via TCP
- Supports Data Access (DA)
- Supports Historical Data Access (HDA)
- Endpoint URI / endpoint URL of the OPC Server:
  opc.tcp://*<Hostname>*:*<Port>*

<Hostname>	IP address or DNS name of the device on which the OPC Server is installed. Default setting: *<DNS name>* Note: The DNS name is the device name under which the device can be identified in a network (on Windows operating systems, see ***Control Panel -> System***).	
<Port>	Port number of the OPC Server Default setting: 4862	(see [1])

  Example: opc.tcp://myPC-Name:4862

  [1]

  If there are multiple PNOZmulti projects, an OPC UA Server instance is created for each project; i.e. a separate OPC endpoint is provided. The port number of the endpoint URL is assigned automatically by the OPC Server. As a result, each PNOZmulti project has its own endpoint URL. Further information is available under Assigning port numbers [📖 17].
- Supported OPC Event and HDA Services

  in accordance with the HDA specification:

  – Part 11: Historical Access Release 1.04

  – Part 4: Services Release 1.05.00

  in accordance with BaseEvent

  – Part 5: Information Model Release 1.05.02
- Supported OPC UA services and service sets

  Only the services and service sets documented here are supported.

Service set	Supported services	Meaning
Secure Channel	▸ OpenSecureChannel ▸ CloseSecureChannel	Connection management between OPC UA Clients and OPC UA Servers
Session	▸ CreateSession ▸ ActivateSession ▸ CloseSession ▸ Cancel	Connection management between OPC UA Clients and OPC UA Servers
View	▸ Browse ▸ BrowseNext ▸ TranslateBrowsePathsToNodeIds	Retrieval of information in the address space

Service set	Supported services	Meaning
Attributes	▸ Read ▸ HistoryRead for EventTypes ▸ Write	Read and write data and metadata
MonitoredItem	▸ CreateMonitoredItems ▸ ModifyMonitoredItems ▸ SetMonitoringMode ▸ SetTriggering ▸ DeleteMonitoredItems	See [2]
Subscription	▸ CreateSubscription ▸ ModifySubscription ▸ SetPublishingMode ▸ Publish ▸ Republish ▸ TransferSubscription ▸ DeleteSubscription	See [2]
Discovery	▸ FindServers ▸ GetEndpoints	Local Discovery Server ME functions (mDNS)

[2] Service Sets **"Subscription"** and **"MonitoredItems"**

An OPC Client can subscribe to data provided by the OPC Server. The purpose of a subscription is to group data using monitored items. The OPC Client can use MonitoredItems to be notified when values are changed.

Please note the limit values for the service sets "Subscription" and "MonitoredItem" (see Limits of the PNOZmulti OPC Server [📖 16]).

4.3 Limits of the PNOZmulti OPC Server

The documented limit values for the PNOZmulti OPC Server UA are fixed and cannot be configured.

Name		Limit value of the PNOZmulti OPC Server
Service set "Subscription" (see also OPC UA properties [📖 14])		
	PublishingInterval	min. 200 ms
		max. 15000 ms
Service set "MonitoredItem" (see also OPC UA properties [📖 14])		
	SamplingInterval	min. 100 ms
		max. 15000 ms

4.4 Assigning port numbers

If there are multiple PNOZmulti projects, an OPC UA Server instance is created for each project; i.e. a separate OPC endpoint is provided. The port number of the endpoint URL is assigned automatically by the OPC Server. As a result, each PNOZmulti project has its own endpoint URL.

The OPC Server assigns a port number to each endpoint URL by incrementing the port numbers by 1, starting from the configured port number. The port number range starts with the default port number 4862.

The port numbers are assigned depending on the alphanumeric sorting of the project names in the "PVISOPCData" project directory of the PNOZmulti OPC Server.

Example:

C:\Users\<*user*>\AppData\Local\Pilz\PVISOPCData**01Project_opc**\DeviceSettings.xml (Port: 4862)

C:\Users\<*user*>\AppData\Local\Pilz\PVISOPCData**02Project_opc**\DeviceSettings.xml (Port: 4863)

C:\Users\<*user*>\AppData\Local\Pilz\PVISOPCData**03Project_opc**\DeviceSettings.xml (Port: 4864)

NOTICE

Access problems when port numbers are assigned more than once

If port numbers are assigned more than once, by other applications or services for example, it may not be possible to access the OPC Server. In this case, error messages are not displayed.
It is the responsibility of the user (project engineer, system integrator for example) to assign the port numbers for their software applications so that none are used more than once.

Ensure that the port numbers configured for the OPC Server are only used by the OPC UA Server instances of the OPC Server.

4.5 Diagnostic configuration

A PNOZmulti 2 device has a PNOZmulti project (device project). The PNOZmulti project is created using the PNOZmulti Configurator.

The diagnostic configuration is also stored in a PNOZmulti project. The OPC Server can be configured with the help of this diagnostic configuration.

The diagnostic configuration also contains all the information required for symbolic access to process and diagnostic data. Equipment identifiers can be assigned for a PNOZmulti 2 device in the PNOZmulti Configurator for this purpose. The equipment identifiers for a PNOZmulti 2 device are displayed in the symbolic namespace. In the symbolic namespace, data on a PNOZmulti 2 device can be accessed via the equipment identifier. This type of access is called symbolic access.

4.6 Basics of diagnostics

In order for the process and diagnostic data of a PNOZmulti 2 device to be mapped in the corresponding instance on the PNOZmulti OPC Server, a project with a diagnostic configuration must be created for each PNOZmulti 2 device using the PNOZmulti Configurator; i.e., a separate diagnostic configuration is created for each PNOZmulti project.

The diagnostic configuration consists of three independent parts:

- Device's diagnostic configuration

  Includes the event messages that are triggered when there are errors in or on the PNOZmulti 2 or if there are changes in the operating state of the PNOZmulti 2
- Diagnostic configuration for safety devices

  Includes the event messages that are triggered when safety devices have a defined state
- Operand's diagnostic configuration - "Bit" diagnostic type

  Includes the event messages that are triggered when inputs, outputs, elements and connection points have a defined state

Once created, the diagnostic configuration must be downloaded to the OPC Server.

Note: The device's diagnostic configuration is not downloaded to the PNOZmulti 2 device because it is part of the operating system of a PNOZmulti 2 device. The device's diagnostic configuration simply needs to be activated on the PNOZmulti 2 device using the PNOZmulti Configurator.

Further information is available under Create and download diagnostic configuration [📖 36].

4.6.1 Event messages and remedy

If an event occurs in or on the diagnosed PNOZmulti 2 device, 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.

4.6.1.1 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 is the lowest priority.

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 users about potential or imminent critical machine states.
  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 (Name of priority)	Pilz priority	OPC UA priority	Note
Error	0	1000.00	Maximum severity
	1	1000.00	Error
	2	950.00	
	3	900.00	
	4	850.00	
	5	800.00	
Warning	6	600.00	Warning
	7	550.00	
	8	500.00	
	9	450.00	
	10	400.00	
Status information	11	200.00	Status information
	12	150.00	
	13	100.00	
	14	50.00	
	15	1.00	No alarm active

4.6.1.2 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.

4.6.1.3 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 (project engineer, system integrator for example). 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: maintenance engineer, for example
- 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, system integrator for example
- 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, system integrator for example
- ---
  No details of responsibility are possible or necessary.

4.6.1.4 Event list

When using the visualisation software PASvisu from Pilz, the event list can either be displayed via the diagnostic widget or can be evaluated directly via the namespace of the OPC Server. For visualisation software from third-party manufacturers, only direct evaluation via the namespace of the OPC Server is available.

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, then the event 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. A current 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 error stack on the PNOZmulti 2. The event list only lists device-relevant messages chronologically. It is not possible to diagnose the user program here, for example.

4.6.2 Device’s diagnostic configuration

The device's diagnostic configuration includes the event messages that are triggered when there are errors in or on the diagnosed PNOZmulti 2 or if there are changes in the operating state of the diagnosed PNOZmulti 2

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

4.6.3 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:

- Elements of a PNOZmulti 2
- Input devices that are monitored by function elements of a PNOZmulti 2

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.

4.6.3.1 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.:

The "E-STOP" function element of a PNOZmulti 2 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.

4.6.4 Diagnostic configuration of operands

The operand's diagnostic configuration of a PNOZmulti 2 contains user-defined event messages that can be used in the user program, and event messages that are triggered when inputs, outputs and connection points have a defined state.

4.6.4.1 Bit diagnostic type

Bit operands of a PNOZmulti 2 are:

- 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 "1" state. On a PNOZmulti 2 it is also possible to evaluate by a rising or falling edge.

5 Install, uninstall and update the PNOZmulti OPC Server

The product PNOZmulti OPC Server can be installed on a device with the Windows operating system (e.g. PC) (see System requirements [📖 28]).

The OPC Server can be installed on the device either as a process or as a service (see Install PNOZmulti OPC Server as process [📖 29] and Install PNOZmulti OPC Server as a service [📖 31]). It is not possible to install it as a process and as a service at the same time. To switch between a process installation and a service installation, the existing installation must first be closed and uninstalled (see Start and stop PNOZmulti OPC Server [📖 41] and Uninstall PNOZmulti OPC Server as a process [📖 33] or Uninstall PNOZmulti OPC Server as a service [📖 33]). The installation can then be carried out again.

An executable installation file is available for the OPC Server. With the help of the installation file, the OPC Server can be installed either as a process or as a service, or an existing installation can be updated (see Installation file [📖 28]).

A self-signed certificate with a public key and associated private key is created for the OPC server when it is started for the first time (see Information security certificates [📖 42]).

5.1 System requirements

The product PNOZmulti OPC Server can be operated on the following diagnostic devices:

- PC or operator and display device with Microsoft Windows 10 operating system (64 Bit)
- PC or operator and display device with Microsoft Windows 11 operating system (64 Bit)
- Operator terminals PMI v8 and operator terminals PMI v7 from Pilz

  Note: The OPC Server is pre-installed on the Pilz operator terminals However, updates can be installed.

5.2 Installation file

Installation file used to install the product PNOZmulti OPC Server:

- PNOZmultiOPCServer_<*Version number*>_Installer.exe

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

NOTICE

Potential security vulnerability from downloading files from the Internet

Downloading files from the Internet is a potential security risk.

Ensure that the downloaded installation file is identical to the installation file provided by Pilz in the download area of the Pilz homepage. To do this, compare the SHA256 hash value of the file you have downloaded with the SHA256 hash value published by Pilz. The SHA256 hash value is also available in the download area of the Pilz homepage.

When carrying out your security risk analysis, consider that downloading from the Internet can represent a security vulnerability for cyber threats.

5.3 Installation

5.3.1 Install PNOZmulti OPC Server as process

Install the product PNOZmulti OPC Server on the device that is to be used as the diagnostic device. The diagnostic device may also be used simultaneously as a display unit. The OPC Server can be installed either manually or silently.

Prerequisites

- The PVIS OPC Server UA is not installed on the device.
  If the PVIS OPC Server UA is installed, it will need to be uninstalled first.
- No version of the product PNOZmulti OPC Server is installed on the device.
- A newer version of the product PNOZmulti OPC Server is not installed on the device.
- The current installation file has been downloaded from the Pilz homepage and is available on the device.
- The user (e.g. project engineer, system integrator) has administrator rights.

Manual install

Procedure

1. If the OPC Server is currently running as a service:
   ⇨ Stop the OPC Server and uninstall it.
2. Double-click on the installation file "PNOZmultiOPCServer_<*Version number*>_Installer.exe".
3. Select the option ***Install as process*** (default setting).
4. Define start behaviour

   When installing the OPC Server as a process, the startup behaviour must be defined by the user (e.g. project engineer, system integrator). The user can insert a link to the OPC Server in the Autostart menu for this purpose.

Silent install

Procedure

- Start the install by calling up the installation file in the Windows command line, the Windows search field or in a batch file.

  Example:

  PNOZmultiOPCServer_<*Version number*>_Installer.exe /Silent

Storage location

After installation, the configuration data and certificate store (PKI directory) of the OPC Server will be located in the local AppData folder ***PNOZmultiOPCServer***:

C:\Users\<*user*>\AppData\Local\Pilz\PNOZmultiOPCServer\

Entries in the Windows Start menu

The installation creates the folder ***PNOZmultiOPCServer*** in the Start menu, with the following entries:

- ***PNOZmulti OPC Server*** <*Version number*>

With a link to the executable file PNOZmultiOPCServer.exe.

Example: PNOZmulti OPC Server 1.0.0

- ***PNOZmulti OPC Server Info***

  With a link to the information file of the PNOZmulti OPC Server and its current configuration.

- ***PNOZmulti OPC Server PKI Configuration***

  With a link to the certificate store (PKI directory).

- ***PNOZmulti OPC Server Projects Configuration***

  With a link to the directory containing the configuration data (project directory PVISOPCData).

Note:

The file PNOZmultiOPCServerInfo.txt and the two folders for the PKI directory and the project directory are not created until the PNOZmulti OPC Server is started for the first time.

5.3.2 Install PNOZmulti OPC Server as a service

Install the product PNOZmulti OPC Server on the device that is to be used as the diagnostic device. The OPC Server is entered as a local service (LocalService) in the operating system under the name ***PNOZmultiOPCServer***. The OPC Server can only be operated as a local service (LocalService).

The OPC Server is installed so that it starts automatically each time the operating system is restarted. If the OPC Server should not be started during restart, the startup behaviour must be changed in the service settings dialogue box. This setting can only be made with administrator rights.

Entries in the registration database:

- Service name PNOZmultiOPCServer
- Service description Pilz PNOZmulti OPC Server

  Note: The service description is valid for all languages.
- Display name Pilz PNOZmulti OPC Server

Prerequisites

- The PVIS OPC Server UA is not installed on the device.
  If the PVIS OPC Server UA is installed, it will need to be uninstalled first.
- No version of the product PNOZmulti OPC Server is installed on the device.
- A newer version of the product PNOZmulti OPC Server is not installed on the device.
- The current installation file has been downloaded from the Pilz homepage and is available on the device.
- The user (e.g. project engineer, system integrator) has administrator rights.

Procedure

1. If the OPC Server is currently running as a process:
   ⇨ Stop the OPC Server and uninstall it.
2. Double-click on the installation file ***PNOZmultiOPCServer_<Version number>_Installer.exe***
3. Select the option ***Install as service***.

   The OPC Server is registered as a service.
4. Start the OPC Server either via the service settings dialogue box (Windows) or restart the device.

Storage location

After installation, the configuration data and certificate store (PKI directory) of the OPC Server will be located in the local Windows AppData folder for services.

C:\Windows\ServiceProfiles\LocalService\AppData\Local\Pilz\PNOZmultiOPCServer.

Entries in the Windows Start menu

The installation creates the folder ***PNOZmultiOPCServer*** in the Start menu, with the following entries:

- ***PNOZmulti OPC Server*** <*Version number*>

  With a link to the executable file PNOZmultiOPCServer.exe.

Example: PNOZmulti OPC Server 1.0.0

- ***PNOZmulti OPC Server Info***

  With a link to the information file of the PNOZmulti OPC Server and its current configuration.

- ***PNOZmulti OPC Server PKI Configuration***

  With a link to the certificate store (PKI directory).

- ***PNOZmulti OPC Server Projects Configuration***

  With a link to the directory containing the configuration data (project directory PVISOPCData).

Note:

The file PNOZmultiOPCServerInfo.txt and the two folders for the PKI directory and the project directory are not created until the PNOZmulti OPC Server is started for the first time.

5.3.3 Installation on a PMI v8

The product PNOZmulti OPC Server and the 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.

5.4 Uninstall

5.4.1 Uninstall PNOZmulti OPC Server as a process

If multiple users with an account are logged in to the device, then any user who is logged in can perform the uninstall.

Prerequisites

- The user has administrator rights.

Procedure

1. Right-click on the PNOZmulti OPC Server in the Windows Start menu
   ⇨ Select ***Uninstall***.

After the uninstall, all the configuration data and certificates are still available in the corresponding directories (see Target directory for configuration data when installed as a service [📖 34] and Target directory for certificates when installed as a service [📖 35]).

We recommend that you delete the configuration data and certificates manually.

5.4.2 Uninstall PNOZmulti OPC Server as a service

Any user with administrator rights can uninstall the OPC Server.

Prerequisites

- The user (e.g. project engineer, system integrator) has administrator rights.

Procedure

1. Right-click on the PNOZmulti OPC Server in the Windows Start menu
   ⇨ Select ***Uninstall***.

After the uninstall, all the configuration data and certificates are still available in the corresponding directories (see Target directory for configuration data when installed as a service [📖 34] and Target directory for certificates when installed as a service [📖 35]).

We recommend that you delete the configuration data and certificates manually.

5.5 Target directories

Depending on the installation type (process or service), the configuration data is stored in the corresponding local AppData folder during installation. The certificates can be found in the relevant subdirectory.

5.5.1 Target directory for the OPC UA Server when installed as a process

After installation, the OPC Server is located in the local AppData folder ***PNOZmultiOPC-Server***.

Target directory

C:\Users\<*user*>\AppData\Local\Pilz\PNOZmultiOPCServer\

For example, the OPC Server can be started via the Windows Start menu in Windows Explorer:

- In the Windows Start menu, select ***PNOZmulti OPC Server*** <*Version number*>.

Note:

To ensure that the OPC Server is available after a computer starts up, the user can insert a link to the OPC Server in the Autostart menu.

5.5.2 Target directory for configuration data when installed as a process

If the OPC Server is installed as a process, the configuration data can be found in the local AppData folder.

Target directory

C:\Users\<*user*>\AppData\Local\Pilz\PVISOPCData\

For example, the directory can be opened via the Windows Start menu in Windows Explorer:

- In the Windows Start menu, select ***Pilz -> PNOZmulti OPC Server Projects Configuration***.

INFORMATION

If multiple users with an account are logged in to the device with Windows operating system, then each user that is logged in has their own local AppData folder on the device; i.e., after installation, a user who is logged in only has access to the local AppData folder.

Each user who starts the OPC Server has their own AppData folder and therefore their own configuration data folder.

Further information is available under Download diagnostic configuration manually [📖 37].

5.5.3 Target directory for configuration data when installed as a service

If the OPC Server is installed as a service, the configuration data can be found in the local AppData folder of Windows for Services.

Target directory

C:\Windows\ServiceProfiles\LocalService\AppData\Local\Pilz\PVISOPCData\

INFORMATION

Only a user with administrator rights can make changes to the configuration of the OPC Server.

Further information is available under Download diagnostic configuration manually [📖 37].

5.5.4 Target directory for certificates when installed as a process

After the OPC Server has been installed as a process and restarted, a self-signed certificate is generated and is stored in the target directory for the configuration data in the subfolder \pki. Further information is available under Directory structure of the certificate store [📖 43].

The certificate is only generated if a certificate has not yet been stored for the executing user.

Target directory

C:\Users\<*user*>\AppData\Local\Pilz\PNOZmultiOPCServer\pki\

For example, the PKI directory can be opened via the Windows Start menu in Windows Explorer:

- In the Windows Start menu, select ***Pilz -> PNOZmulti OPC Server PKI Configuration***.

5.5.5 Target directory for certificates when installed as a service

After the OPC Server has been installed as a service and restarted, a self-signed certificate is generated and is stored in the target directory for the configuration data in the subfolder \pki. Further information is available under Directory structure of the certificate store [📖 43].

The certificate is only generated if a certificate has not yet been stored for the executing user.

Target directory

C:\Windows\ServiceProfiles\LocalService\AppData\Local\Pilz\PNOZmultiOPCServer\pki\

For example, the PKI directory can be opened via the Windows Start menu in Windows Explorer:

- In the Windows Start menu, select ***Pilz -> PNOZmulti OPC Server PKI Configuration***.

6 Create and download diagnostic configuration

6.1 Create PNOZmulti project with diagnostic configuration

A PNOZmulti project with diagnostic configuration must be created in the PNOZmulti Configurator for each PNOZmulti 2 device that is connected to the product PNOZmulti OPC Server. Once created, the PNOZmulti project must be downloaded to the PNOZmulti 2 device and to the OPC Server.

Procedure

1. Create PNOZmulti project
   ⇨ In the PNOZmulti Configurator, create a PNOZmulti project for each PNOZmulti 2 device and then save the PNOZmulti projects.
   In each PNOZmulti project, the device must be configured to support PVIS diagnostics.
   Further information is available in the online help for the PNOZmulti Configurator.
2. Download PNOZmulti project
   ⇨ Download the PNOZmulti projects to the PNOZmulti 2 devices.
   Information about the procedure is available in the online help for the PNOZmulti Configurator.
3. Link diagnostic configuration

   The diagnostic configuration must be linked before it can be downloaded to the OPC Server. The files required for diagnostics are created during the link process.

   ⇨ In the PNOZmulti Configurator, select ***Diagnostics -> Link diagnostic configuration***.

   The required files are stored in the same directory as the PNOZmulti project.
4. Download diagnostic configuration
   ⇨ Download the linked diagnostic configurations for the PNOZmulti projects to the OPC Server.
   Information about the procedure is available under Download diagnostic configuration manually [📖 37].

6.2 Download diagnostic configuration manually

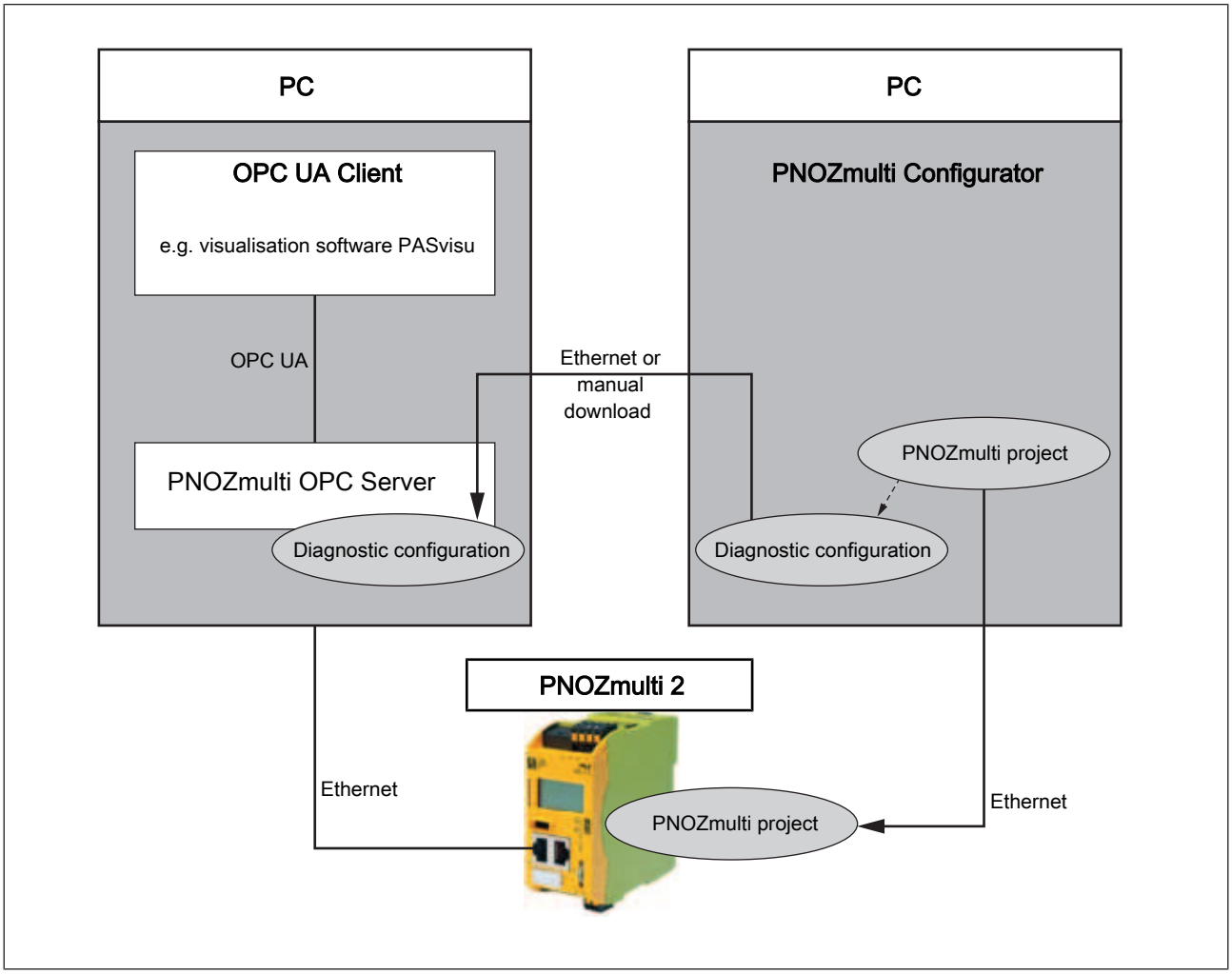

Fig.: Structure of a diagnostic environment with a PNOZmulti OPC Server (example)

In the example structure, the product PNOZmulti OPC Server and the 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 is always installed on the device that is used as the diagnostic device. With the help of visualisation software (e.g. visualisation software PASvisu from Pilz), the diagnostic device can also be used as a display unit. The display unit operates as an OPC UA Client. The diagnostic data from the namespace of the OPC Server is displayed on the OPC UA Client.

The OPC Server can be used to diagnose multiple PNOZmulti 2 devices from Pilz. The PNOZmulti 2 devices are connected to the OPC Server via a network (Ethernet interface) for this purpose.

When saving a PNOZmulti project or device project with the PNOZmulti Configurator, a "PVISOPCData" directory is created on the device on which the PNOZmulti Configurator is installed. This directory contains another directory with the name of the PNOZmulti project and the abbreviation "_opc".

Example:

…\Documents\PNOZmulti\<*Project name*>\PVISOPCData\<*Project name*>*_opc*

Procedure

1. Use one of the following options to download the diagnostic configuration to the OPC Server:

   Option 1:

   Use this option primarily if you are using visualisation software from a third-party manufacturer.

   ⇨ Copy the directory <*Project name*>*_opc* into the configuration directory of the OPC Server.

   Example when installing the OPC Server as a process:
   C:\Users\<*user*>\AppData\Local\Pilz\PVISOPCData\<*Project name*>_opc\

   Example when installing the OPC Server as a service:
   C:\Windows\ServiceProfiles\LocalService\AppData\Local\Pilz\PVISOPCData\<Project name>_opc\.

   Option 2:

   Use this option primarily if you are using visualisation software from a third-party manufacturer.

   ⇨ In the Windows Start menu, select ***Pilz -> PNOZmulti OPC Server Projects Configuration***, click on the link that opens the relevant configuration directory and copy the contents of the directory PVISOPCData (<*Project name*>_opc) into the configuration directory.

2. Restart the OPC Server to apply the new configuration.

6.3 Download diagnostic configuration with the PASvisu Builder

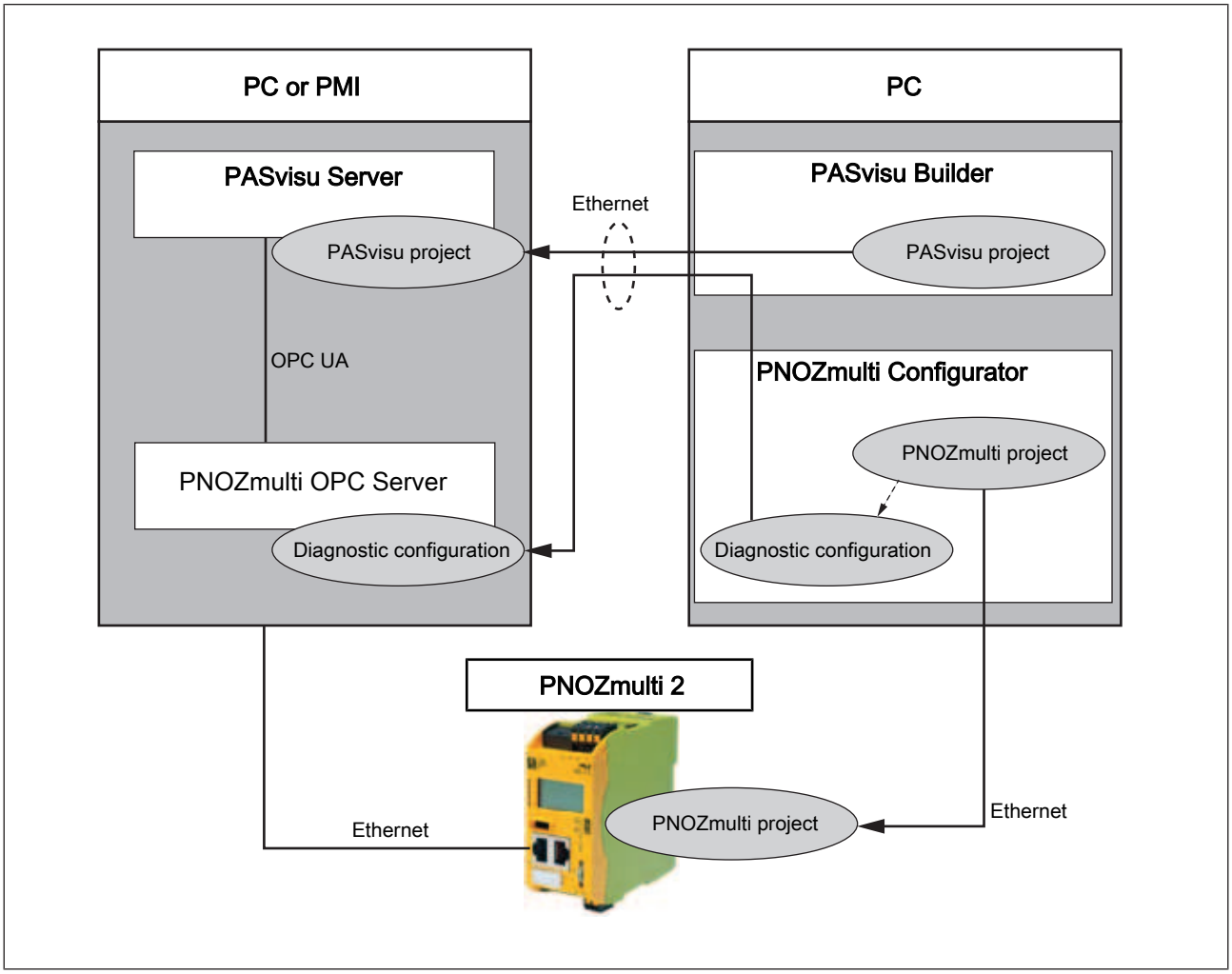

Fig.: Structure of a diagnostic environment with a PNOZmulti OPC Server (example)

In the example structure, the product PNOZmulti OPC Server and the PNOZmulti Configurator are installed on different PCs. However, it is also possible to install the OPC Server and PNOZmulti Configurator on the same PC. When using a PMI operator terminal from Pilz, the diagnostic environment illustrated is the preferred setup.

The OPC Server is always installed on the device (PMI or PC) that is used as the diagnostic device. With the help of the visualisation software PASvisu from Pilz, the diagnostic device can also be used as a display unit. The display unit operates as an OPC UA Client. The diagnostic data from the namespace of the OPC Server is displayed on the OPC UA Client.

The PASvisu Builder and PNOZmulti Configurator are always installed on a PC.
The OPC Server and PASvisu Builder must be connected via Ethernet.

The OPC Server can be used to diagnose multiple PNOZmulti 2 devices from Pilz. The PNOZmulti 2 devices are connected to the OPC Server via a network (Ethernet interface) for this purpose.

When PASvisu is used as an OPC UA Client, it is necessary to create a PASvisu project in the PASvisu Builder. In the PASvisu project, the visualisation must be programmed depending on the application, which among other things requires you to specify a data source.

The desired PNOZmulti project should be stated as the data source. The data source contains all the necessary information, including the diagnostic configuration.

Prerequisites

- The PASvisu Server is running on the same device and under the same user as the OPC Server.
- A PNOZmulti project has been created (PNOZmulti Configurator).
- The diagnostic configuration has been linked (PNOZmulti Configurator).
- A PASvisu project has been created (PASvisu Builder).

  Further information is available in the online help for the PASvisu Builder.

Procedure

- In PASvisu, download the PASvisu project.

  The diagnostic configuration is downloaded to the OPC Server automatically. Further information is available in the online help for PASvisu.

  After the download, both PASvisu and the OPC Server will restart.

7 Start and stop PNOZmulti OPC Server

The PASvisu Builder stops the OPC Server before downloading the PASvisu project and then starts it again afterwards.

When a PMI operator terminal from Pilz is used, the OPC Server is started automatically when the device is started.

Further possible procedures are described below.

Start/stop OPC Server as a process

- Start process
  - Start the process via the Windows Start menu.

    Note: The process is not started automatically.
- Start process automatically when using a device from a third-party manufacturer
  - Create a corresponding entry in the Autostart in the Windows operating system.
- Stop/end process
  - End the process in the Windows Task Manager

Start, temporarily disable or stop OPC Server as a service

- Start service

  Provided the service is not disabled, it can be started in several ways:
  - Restart the device.
    The OPC Server starts automatically.

  or
  - Start the service as an administrator via the Windows services dialogue box, by starting the service via the Windows command line ***services.msc*** for example.
- Temporarily disable service
  - Disable the service as an administrator via the Windows services dialogue box, by setting the start type to "Manual".
- Stop/end service
  - Stop the service as an administrator via the Windows services dialogue box, by setting the service status to "End".

  or
  - Stop the service as an administrator via the Windows services dialogue box, by stopping the service via the Windows command line ***services.msc*** for example.

INFORMATION

If there is no diagnostic configuration available on the OPC Server, the OPC Server will start but is immediately shut down again.

Make sure that there is a diagnostic configuration available on the OPC Server. Further information is available under Download diagnostic configuration manually [📖 37].

8 Information security certificates

With the product PNOZmulti OPC Server, information security is established with the aid of certificates compliant with the standard X.509. The certificates are used to safeguard communication between an OPC Client and the OPC Server; i.e. with a certificate, data traffic from a unidirectional communication connection is encrypted and communication between one subscriber and another is authenticated.

A separate certificate is required for each subscriber in the Client-Server communication. An OPC Client cannot communicate with the OPC Server until the OPC Server has a valid certificate from the OPC Client and the OPC Client has a valid certificate from the OPC Server.

8.1 Directory structure of the certificate store

If the OPC Server does not have its own certificate, a self-signed certificate with a public key and associated private key is created when it is started for the first time. This is the case following installation, for example.

The certificate, the public key and the non-public (private) key are stored in a certificate store on the installation device. During installation, a memory area for certificates is reserved on the installation unit, i.e. a folder with a predefined directory structure is created.

The location of the certificate store depends on whether the OPC Server is installed as a process or service (see Target directory for certificates when installed as a process [📖 35] and Target directory for certificates when installed as a service [📖 35]). The directory structure is always the same:

<basepath>\pki

\trusted	Location for trusted Client certificates These may be trusted certificates that have been copied over from the rejected folder, for example.
\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. File name: cert_Pilz_PNOZmulti_OPC_Server_UA.der
\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. File name: private_key_Pilz_PNOZmulti_OPC_Server_UA.pem
\rejected	Location for Client certificates rejected by the OPC Server CN, DC and fingerprint are used as the file name for rejected certificates. File name: CN@DC_*[fingerprint]*.der Example: UaExpert@DE_C3048L7D_P[b6ded749db8eeabe7e97ae-a7a4e9e8632035e317].der Note: The certificate's CN and DC must be checked for invalid characters before the file is created. Invalid characters must be replaced by an underscore ("_"). Valid characters: [A … Z], [a … z] and [0 … 9].

8.2 Security policies

Various security policies can be activated for the product PNOZmulti OPC Server. If several security policies are activated, all activated security policies can be used for communication with the OPC Clients.

Security policy (endpoints)	Availability for connections
None - None exclusively for Discovery (OpenSession)	---
Basic256-Sha256 Sign	♦
Basic256-Sha256 SignAndEncrypt	♦
Aes256-Sha256-RsaPss Sign	♦
Aes256-Sha256-RsaPss SignAndEncrypt	♦

Note:

The OPC Server accepts sha256WithRSAEncryption (key length = 2048 Bit).

8.3 Procedure for certificate authentication

To ensure that Client-Server communication can take place, the communication partners must exchange their certificates. To exchange the certificates, the certificate files must be stored in the relevant folder of the OPC Client and the product PNOZmulti OPC Server (see Directory structure of the certificate store [📖 43]).

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

- Using the PASvisu Client as OPC UA Client
  - OPC Server and visualisation system PASvisu are on the same device (see Authentication in the case of local installation [📖 45])
  - OPC Server and visualisation system PASvisu are installed on different devices (see Authentication in the case of remote installation [📖 45])
- Using the OPC UA Client from a third party manufacturer (see Authentication for OPC Clients from a third-party manufacturer [📖 47])

8.3.1 Authentication in the case of local installation

If the OPC Client of the visualisation system PASvisu from Pilz is used as OPC Client and the product PNOZmulti OPC Server and PASvisu Client are installed on the same device and under the same user, then the certificates are exchanged automatically during installation:

- The certificate of the OPC Server (cert_Pilz_PNOZmulti_OPC_Server_UA.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 OPC Server

The user does not need to do anything else.

8.3.2 Authentication in the case of remote installation

If the visualisation system PASvisu and the product PNOZmulti OPC Server are installed on different devices, then the users must exchange certificates the first time the systems are started after installation.

Procedure

1. Start the OPC Server and the visualisation system PASvisu.

   The PASvisu Server's OPC Client attempts to establish a connection to the OPC Server. As no valid certificate from the OPC Server is available to the OPC Client after installation, the OPC Client reacts as follows:

   – The OPC Client does not accept the Server certificate.

   The Server certificate is stored in the PASvisu certificate store.
   Directory: <…>\pki\rejected

   The name of the certificate file is composed of the Common Name, Domain Component and fingerprint.
   Structure of the file names: *<CN>@<DC>_[Fingerprint]*.der

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

Example:
PNOZmultiOPCServer@<*Hostname*>_[0ff981ea8ba58ad04bad16f89a3c-c9c0c60e7fac].der
- CN: PNOZmultiOPCServer
- DC: <*Hostname*>
- Fingerprint: 0ff981ea8ba58ad04bad16f89a3cc9c0c60e7fac

– Connection to the OPC Server is not established.

An error message appears.

2. Check the trustworthiness of the Server certificate (see Verify Client certificate [📖 49]).

The Server certificate is trustworthy:

⇨ Move the certificate in the PASvisu certificate store from the rejected directory to the trusted directory.

The OPC Server is now authenticated on the PASvisu Server's OPC Client.

There are doubts about the trustworthiness of the Server certificate:

⇨ Check the security technology of the IT infrastructure or contact your system administrator.

3. Start PASvisu, to establish a connection between OPC Server and OPC Client, and so that the OPC Server receives a valid certificate from the OPC Client.

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

– The OPC Server does not accept the Client certificate

– The Client certificate is not stored in the OPC Server certificate store.
Directory <…>\pki\rejected

The name of the certificate file is composed of the Common Name, Domain Component and fingerprint.
Structure of the file names: <*CN*>@<*DC*>*_[Fingerprint]*.der
- CN: Common Name
- DC: Domain Component
- Fingerprint: 40-digit HEX value

Example:

Pilz_PASvisu_OPC_UA_self_signed_@192_168_1_110_[e0e759ab82345ee7207f1fda4ced-b0153e0d7692].der
- 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 are replaced with an underscore.

– Connection to the OPC Server is not established.

An error message appears.

4. Check the trustworthiness of the Client certificate (see Verify Client certificate [📖 49]).

   The Client certificate is trustworthy:

   ⇨ Move the certificate in the OPC Server certificate store from the rejected directory to the trusted directory.

   The OPC Client is now authenticated on the OPC Server.

   There are doubts about the trustworthiness of the Server certificate:

   ⇨ Check the security technology of the IT infrastructure or contact your system administrator.

5. Restart the OPC Server after the certificate exchange.
6. Following mutual authentication, the PASvisu Server's OPC Client automatically establishes the connection to the OPC Server.

NOTICE

During authentication, a certificate (Client certificate/Server certificate) is only saved in the directory "<…>\pki\rejected" 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"

8.3.3 Authentication for OPC Clients from a third-party manufacturer

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

Procedure

- OPC Client
  Follow the authentication instructions in the third-party manufacturer's operating manual.
  Note:
  Some OPC Clients need information about the directory in which the certificate of the OPC Server is stored. After installation, the certificate of the OPC Server from Pilz is located in the directory <…>\pki\own (see Directory structure of the certificate store [📖 43]).
- Product PNOZmulti OPC Server
  In principle, the procedure for the OPC Server is the same as the procedure when the PASvisu Client is used as OPC Client (see Authentication in the case of remote installation [📖 45]).

8.4 Generate and incorporate your own Server certificates

It is possible to use a self-generated certificate instead of the certificate that is created when the OPC Server is installed. 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.

Requirements of Server certificates

A self-generated certificate must have the following parameters and parameter values

▸ Basic conditions

Parameter	Value	
Encryption algorithm	2048 Bit, SHA256	
CA	CA:FALSE	
Period of validity	"notAfter" attribute	99991231235959Z
	"notBefore" attribute	19700101000000Z
Organisation name	Customer of Pilz GmbH & Co. KG	

If the parameter ***NID_key_usage*** is used, the value ***cRLSign*** must not be set.

▸ Parameters for devices with Windows operating system

Parameter		Value
CN		PNOZmultiOPCServer@*<HOSTNAME>*
SubjectAlternativeName		
	Application URI (URL)	urn:*<HOSTNAME>*:Pilz:PNOZmultiOPCServer
	DNS name (dNSName)	*<HOSTNAME>*
	Fully qualified domain name (dNSName)	*<HOSTNAME>.<DOMAINNAME>* Example: *<PC name in the network>.<Name of the network>*.de
	IP address (Only with static IP address)	*<IP address>* Example: IP address=192.168.0.1

NOTICE

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

- Specification of the OPC Foundation (OPC Unified Architecture; Part 2: Security Model)
- 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 [📖 43]).

8.5 Renew Server certificate

If you want to renew the generated OPC Server certificate, follow the instructions below:

1. Delete the "own" and "private" directories.
2. Restart the OPC Server.

   A new self-signed certificate with a public key and associated private key is created.

8.6 Verify Client certificate

Before a certificate is copied from the directory <…>\pki\rejected to the directory <…>\pki\trusted during authentication, the user should verify that the certificate is trustworthy.

- 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 OPC Server. It is sufficient just to compare the fingerprints.
- Product PNOZmulti OPC Server
  The content of the certificate in the directory <…>\pki\rejected of the OPC Server 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.

8.7 Invalidate Client certificate

To invalidate a Client certificate, follow the instructions below:

1. Delete the certificate from the directory <…>\pki\trusted (see Directory structure of the certificate store [📖 43]).
2. Restart the OPC Server.

8.8 Delete Client certificates

If you want to delete all trustworthy Client certificates, follow the instructions below:

- Delete the "trusted" directory.

9 OPC data type assignment

Various data types are defined through the OPC UA standard. The table below contains the data types supported by the product PNOZmulti OPC Server and their assignment to the PNOZmulti 2 data types and the diagnostics in the namespace:

OPC UA data type	PNOZmulti 2 data types	
Boolean	Boolean/Bit value Data width: 1 Bit Value range: TRUE/FALSE	
	Assignment:	
	PNOZmulti 2	▸ Input bit, output bit ▸ Virtual I/Os
	Diagnostics	Boolean (see namespace: "SummaryInformation")
Byte	Positive integer value (Integer) Data width: 8 Bit (1 Byte) Value range: 0 ... 255	
	Assignment	
	PNOZ m B0	▸ Namespace: .../DetailInformation/SystemInfo/Integrity/DiagnosticConfigPNOZmulti: – Status ▸ Namespace: .../Generic/Service-Advanced-data: – LED-IO – LED status
	PNOZ m B1	▸ Namespace: .../DetailInformation/SystemInfo/Integrity/DiagnosticConfigPNOZmulti: – Status ▸ Namespace: .../Generic/Service-Advanced-data/: – LED-IO – LED status ▸ Namespace: .../Generic/ConnectedDevices/PITreader/: – Permission – Status
Double	*<not supported>*	

OPC UA data type	PNOZmulti 2 data types	
Float	Floating point number with single precision (as per IEEE) Data width: 32 Bit (4 Bytes) Note: If a value of the Float data type is not initialised or not accessible, then the QNAN data type is used.	
	Assignment:	
	PNOZmulti 2	Input value analogue module
Int16	Integer value (Integer) Data width: 16 Bit (1 Word) Value range: -32768 … 32767	
	Assignment:	
	PNOZmulti 2	Input value, output value analogue module
Int32	*<not supported>*	
Int64	*<not supported>*	
LocalizedText	Text that can have a local identifier as an option For coding see "String"	
	Assignment: String for diagnostics (see namespace: .../DetailInformation/SystemInfo/)	
SByte	*<not supported>*	
String	String in Unicode format Coding: 2 Bytes (Octets) per character	
	Assignment: String for diagnostics (see namespace: .../DetailInformation/SystemInfo/)	
UInt16	Positive integer value (Integer) Data width: 16 Bit (1 Word) Value range: 0 ... 65535	
	Assignment:	
	Diagnostics	See namespace: "OverviewInformation" -> All "Status" nodes

OPC UA data type	PNOZmulti 2 data types	
UInt32	Positive integer value (Integer) Data width: 32 Bit (2 Words) Value range: 0 ... 4294967295	
	Assignment:	
	PNOZ m B0	▸ Input value: Ivalue-Speed, Ivalue-Position See namespace: .../Generic/ Service-Advanced-data/
	PNOZ m B1	▸ Input value: Ivalue-Speed, Ivalue-Position See namespace: .../Generic/ Service-Advanced-data/ ▸ PITreader: KeyOrderNo, KeySerialNo See namespace: .../Generic/ ConnectedDevices/…
UInt64	Positive integer value (Integer) Data width: 64 Bit (4 Words) Value range: 0 ... 18446744073709551615	
	Assignment:	
	PNOZ m B0	- - -
	PNOZ m B1	▸ PITreader: SecurityID See namespace: .../Generic/ Service-Advanced-data/ KeyInPocketLists/… *<List1..4>.<Line1..20>*.SecurityId

10 Namespace

With OPC UA, every information unit in the namespace is called a node. The address space is formed from the sum of all the information units / nodes. The OPC specification requires that each node can be identified uniquely in the address space via an identifier (Node ID). The nodes are linked to each other via references.

10.1 Basics

A namespace is defined via its URI (Unique Resource Identifier). The namespace URI describes a node set resource. Each namespace has a namespace index. The allocation of the namespace indices to the namespace URIs is stored in a table on the Server.

INFORMATION

We do not recommend that you access a namespace of the product PNOZmulti OPC Server via its namespace index because the namespace index is assigned temporarily by the OPC Server. The index may change after the OPC Server is restarted, if additional namespaces are added, for example.

We recommend that you access a namespace via the namespace table of the OPC Server.

Namespace table of the OPC Server:

Namespace URI	Description
http://opcfoundation.org/UA/	Basic specifications defined by the OPC Foundation, such as basic data types for example. ▸ The OPC Server supports a reduced namespace for a minimal OPC UA Server (NS0). ▸ The OPC Server supports a reduced namespace for BaseEvent communication in the OPC UA Server (NS0).
urn:<*DNS name*>:Pilz: PNOZmultiOPCServer	URI for unique identification of the Server application (NS1)
urn:Pilz:PNOZmultiOPCServer: PNOZmultiNamespace	URI for the PNOZmulti namespace
urn:Pilz:PNOZmultiOPCServer: PNOZmultiNamespace:Symbolic	URI for the symbolic namespace

The application name of the OPC Server is ***Pilz PNOZmulti OPC Server***.

The namespaces supported by the OPC Server are language-independent; i.e. the default language setting is English.

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 2 device. 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. An element can be accessed via the hierarchical node name (path specification). 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 and its elements are accessed is stated at the start of each area.

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.

  Example: "1 .. 2" means that up to two nodes are available.
- "/" is used as the separator when stating the path.

10.2 Structure of the namespace

10.2.1 DeviceProjectName

The PNOZmulti OPC Server supports node identifiers (NodeId) specified by Pilz for project data.

Node set namespace URI

urn:Pilz:PNOZmultiOPCServer:PNOZmultiNamespace

Structure and content

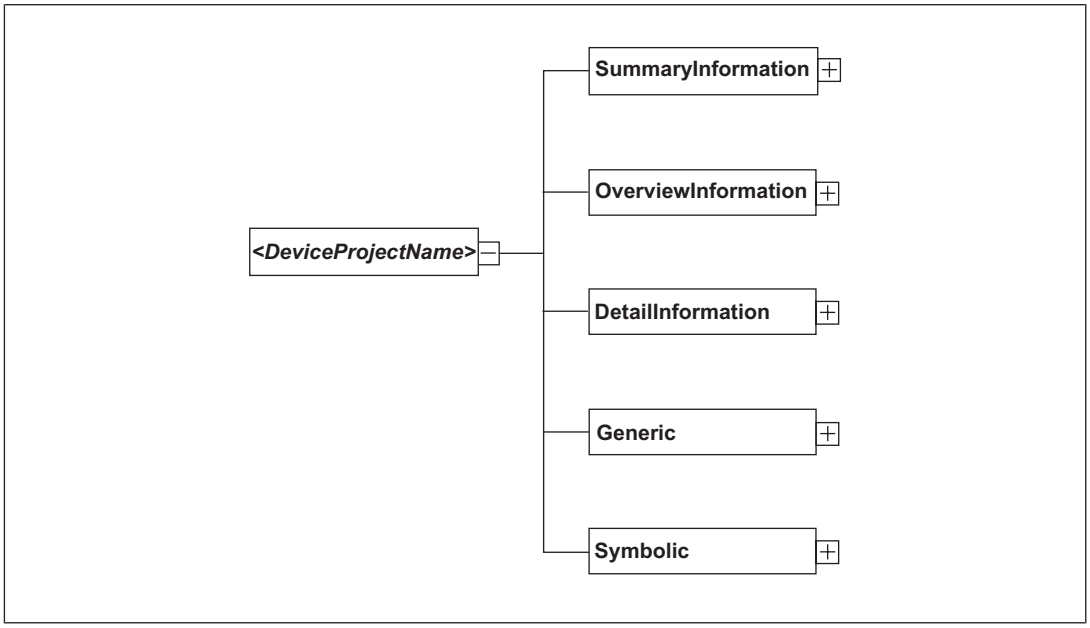

Fig.: *<DeviceProjectName>/*

Node name		Meaning
<DeviceProjectName>		Name of the device project e.g. MyDeviceProject_1
Path: *<DeviceProjectName>/*		
	SummaryInformation	Area containing summary information about entries in the event list (see DeviceProjectName/SumaryInformation [📖 58])
	Path: *<DeviceProjectName>*/SummaryInformation/	
	OverviewInformation	Area containing status information about the event list for modules from the configurable systems PNOZmulti 2 (see DeviceProjectName/OverviewInformation [📖 62])
	Path: *<DeviceProjectName>*/OverviewInformation/	

Node name		Meaning
	DetailInformation	Area containing detailed information about entries in the event list (see DeviceProjectName/DetailInformation [📖 68])
	Path: <*DeviceProjectName*>/DetailInformation/	
	Generic	Area containing device-specific process data The process data can be accessed via the absolute addresses (see DeviceProjectName/Generic [📖 77])
	Path: <*DeviceProjectName*>/Generic	
	Symbolic	Area for access to process and diagnostic data via equipment identifiers (Equip-ID) (see DeviceProjectName/Symbolic [📖 87])
	Path: <*DeviceProjectName*>/Symbolic	

10.2.2 DeviceProjectName/SumaryInformation

The PNOZmulti OPC Server supports node identifiers (NodeId) specified by Pilz for project data.

Node set namespace URI

urn:Pilz:PNOZmultiOPCServer:PNOZmultiNamespace

Structure and content

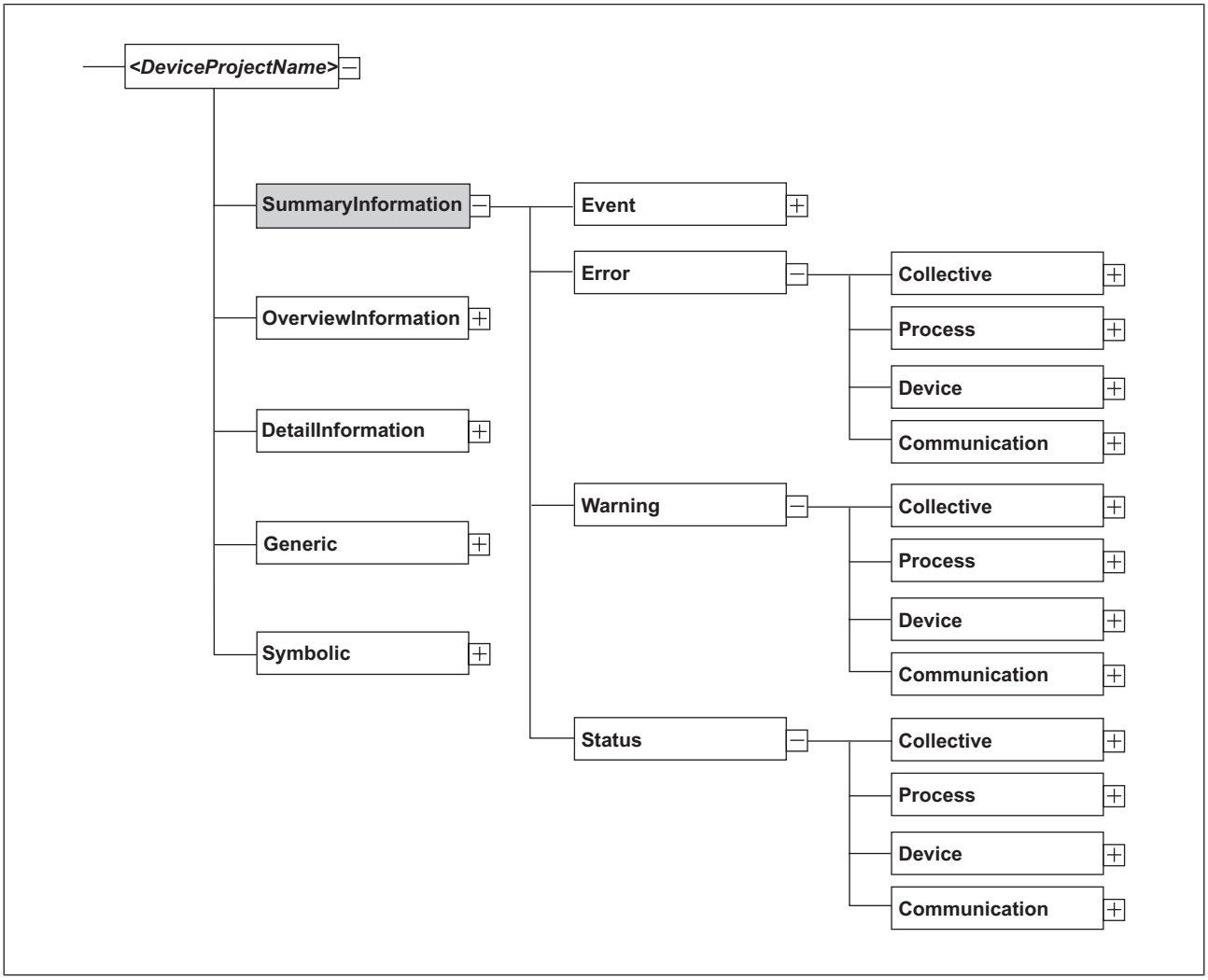

Fig.: Area <*DeviceProjectName*>/SummaryInformation/

Node name	Meaning
SummaryInformation	Area containing summary information about entries in the event list
	A maximum of the first 100 entries from the event list are evaluated.
Path: <*DeviceProjectName*>/SummaryInformation/	

Node name	Meaning	
Event	Event occurred/not occurred Data type: Boolean	
	Display	**Meaning**
	FALSE	No event occurred
	TRUE	Event occurred
Path: <*DeviceProjectName*>/SummaryInformation/Event		
Error	Area containing information on event messages of priority class "Error"	
Path: <*DeviceProjectName*>/SummaryInformation/Error/		
Collective	Information on event messages of priority class "Error" Data type: Boolean	
	Display	**Meaning**
	FALSE	No error present
	TRUE	Error present
Path: <*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
Path: <*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
Path: <*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
Path: <*DeviceProjectName*>/SummaryInformation/Error/Communication		

Node name	Meaning	
Warning	Area containing information on event messages of priority class "Warning"	
Path: <*DeviceProjectName*>/SummaryInformation/Warning/		
Collective	Event message of the priority class "Warning" present/not present Data type: Boolean	
	Display	**Meaning**
	FALSE	No warning present
	TRUE	Warning present
Path: <*DeviceProjectName*>/SummaryInformation/Warning/Collective		
Process	Information on event messages of priority class "Warning" that belong to the "Process" area Data type: Boolean	
	Display	**Meaning**
	FALSE	No warning present
	TRUE	Warning present
Path: <*DeviceProjectName*>/SummaryInformation/Warning/Process		
Device	Information on event messages of priority class "Warning" that belong to the "Plant/device" area Data type: Boolean	
	Display	**Meaning**
	FALSE	No warning present
	TRUE	Warning present
Path: <*DeviceProjectName*>/SummaryInformation/Warning/Device		
Communication	Information on event messages of priority class "Warning" that belong to the "Communication" area Data type: Boolean	
	Display	**Meaning**
	FALSE	No warning present
	TRUE	Warning present
Path: <*DeviceProjectName*>/SummaryInformation/Warning/Communication		
Status	Area containing information on event messages of priority class "Status information"	
Path: <*DeviceProjectName*>/SummaryInformation/Status/		

<table>
<tr><th>Node name</th><th colspan="2">Meaning</th></tr>
<tr><td rowspan="5">Collective</td><td colspan="2">Event message of the priority class "Status information" present/ not present
Data type: Boolean</td></tr>
<tr><td>Display</td><td>Meaning</td></tr>
<tr><td>FALSE</td><td>No status information available</td></tr>
<tr><td>TRUE</td><td>Status information available</td></tr>
<tr><td colspan="3">Path: <DeviceProjectName>/SummaryInformation/Status/Collective</td></tr>
<tr><td rowspan="4">Process</td><td colspan="2">Information on event messages of priority class "Status information" that belong to the "Process" area
Data type: Boolean</td></tr>
<tr><td>Display</td><td>Meaning</td></tr>
<tr><td>FALSE</td><td>No status information available</td></tr>
<tr><td>TRUE</td><td>Status information available</td></tr>
<tr><td colspan="3">Path: <DeviceProjectName>/SummaryInformation/Status/Process</td></tr>
<tr><td rowspan="4">Device</td><td colspan="2">Information on event messages of priority class "Status information" that belong to the "Plant/device" area
Data type: Boolean</td></tr>
<tr><td>Display</td><td>Meaning</td></tr>
<tr><td>FALSE</td><td>No status information available</td></tr>
<tr><td>TRUE</td><td>Status information available</td></tr>
<tr><td colspan="3">Path: <DeviceProjectName>/SummaryInformation/Status/Device</td></tr>
<tr><td rowspan="4">Communication</td><td colspan="2">Information on event messages of priority class "Status information" that belong to the "Communication" area
Data type: Boolean</td></tr>
<tr><td>Display</td><td>Meaning</td></tr>
<tr><td>FALSE</td><td>No status information available</td></tr>
<tr><td>TRUE</td><td>Status information available</td></tr>
<tr><td colspan="3">Path: <DeviceProjectName>/SummaryInformation/Status/Communication</td></tr>
</table>

10.2.3 DeviceProjectName/OverviewInformation

The PNOZmulti OPC Server supports node identifiers (NodeId) specified by Pilz for project data.

Node set namespace URI

urn:Pilz:PNOZmultiOPCServer:PNOZmultiNamespace

Structure and content

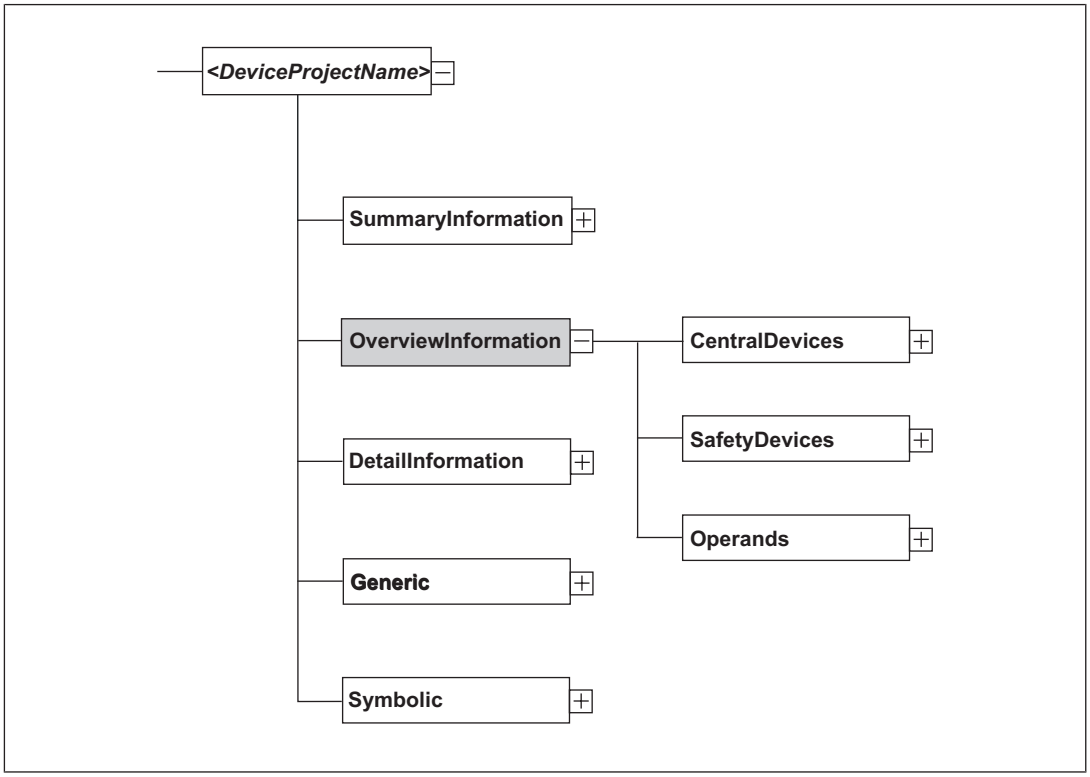

Fig.: Area *<DeviceProjectName>*/OverviewInformation/

Node name		Meaning
OverviewInformation		Area containing status information about modules of PNOZmulti 2 devices A maximum of the first 100 entries from the event list are evaluated.
Path: *<DeviceProjectName>*/OverviewInformation/		
	CentralDevices	Area containing status information about modules (see CentralDevices [📖 64])
	Path: *<DeviceProjectName>*/OverviewInformation/CentralDevices/	
	SafetyDevices	Area containing status information about safety devices (see SafetyDevices [📖 65])
	Path: *<DeviceProjectName>*/OverviewInformation/SafetyDevices/	

Node name		Meaning
	Operands	Area containing information about the operands for which operand diagnostics are configured (see Operands [📖 67])
	Path: <*DeviceProjectName*>/OverviewInformation/Operands/	

10.2.3.1 CentralDevices

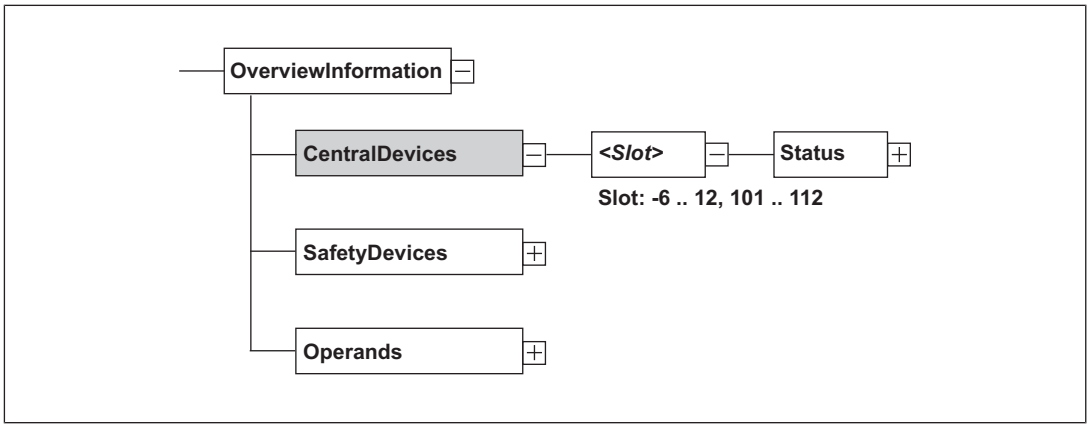

Fig.: Area *<DeviceProjectName>*/OverviewInformation/CentralDevices/

Node name	Meaning	
CentralDevices	Area containing status information about modules	
Path: *<DeviceProjectName>*/OverviewInformation/CentralDevices/		
<Slot>	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: *<DeviceProjectName>*/OverviewInformation/CentralDevices/*<Slot>*/		
Status	Event occurred/not occurred The event is displayed in Bit 00. Data type: UInt16	
	Display	**Meaning**
	FALSE	No event occurred
	TRUE	Event occurred
Path: *<DeviceProjectName>*/OverviewInformation/CentralDevices/*<Slot>*/Status		

10.2.3.2 SafetyDevices

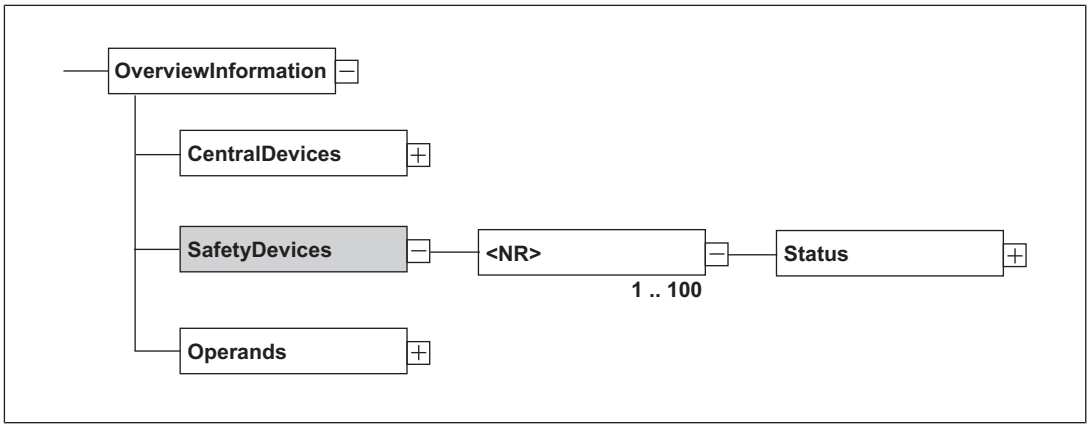

Fig.: Area *<DeviceProjectName>*/OverviewInformation/SafetyDevices/

Node name	Meaning		
SafetyDevices	Area containing status information about safety devices		
Path: *<DeviceProjectName>*/OverviewInformation/SafetyDevices/			
<NR>	Element ID The node name can be "1" … "100".		
Path: *<DeviceProjectName>*/OverviewInformation/SafetyDevices/*<NR>*/			
Status	Status of the safety device Data type: UInt16		
	Bit 00: Safety device active/inactive		
		Display	**Meaning**
		FALSE	Safety device active
		TRUE	Safety device inactive
	Bit 01: Safety device was/was not 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		
		Display	**Meaning**
		FALSE	Safety device not ready for reset
		TRUE	Safety device ready for reset
	Bit 03: Function test		
		Display	**Meaning**
		FALSE	Function test not required
		TRUE	Function test required

<table>
<tr><th>Node name</th><th colspan="3">Meaning</th></tr>
<tr><td rowspan="4"></td><td colspan="3">Bit 04: Application error</td></tr>
<tr><td></td><td>Display</td><td>Meaning</td></tr>
<tr><td></td><td>FALSE</td><td>Error not present</td></tr>
<tr><td></td><td>TRUE</td><td>Error present</td></tr>
<tr><td rowspan="4"></td><td colspan="3">Bit 05: Plausibility error</td></tr>
<tr><td></td><td>Display</td><td>Meaning</td></tr>
<tr><td></td><td>FALSE</td><td>Error not present</td></tr>
<tr><td></td><td>TRUE</td><td>Error present</td></tr>
<tr><td></td><td colspan="3">Bit 06 … Bit 14: Reserved</td></tr>
<tr><td rowspan="4"></td><td colspan="3">Bit 15: Device error</td></tr>
<tr><td></td><td>Display</td><td>Meaning</td></tr>
<tr><td></td><td>FALSE</td><td>Error not present</td></tr>
<tr><td></td><td>TRUE</td><td>Error present</td></tr>
<tr><td colspan="4">Path: <DeviceProjectName>/OverviewInformation/SafetyDevices/<NR>/Status</td></tr>
</table>

10.2.3.3 Operands

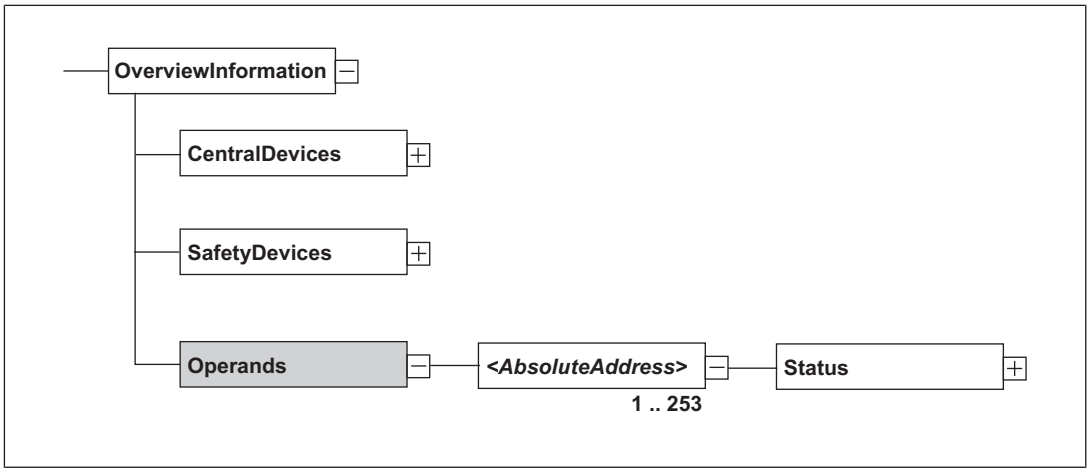

Fig.: Area *<DeviceProjectName>*/OverviewInformation/Operands/

Node name	Meaning	
Operands	Area containing information about the operands for which operand diagnostics are configured	
Path: *<DeviceProjectName>*/OverviewInformation/Operands/		
<AbsoluteAddress>	Operand ID The node name can be "1" … "253".	
Path: *<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
Path: *<DeviceProjectName>*/OverviewInformation/Operands/*<AbsoluteAddress>*/Status		

10.2.4 DeviceProjectName/DetailInformation

The PNOZmulti OPC Server supports node identifiers (NodeId) specified by Pilz for project data.

Node set namespace URI

urn:Pilz:PNOZmultiOPCServer:PNOZmultiNamespace

Structure and content

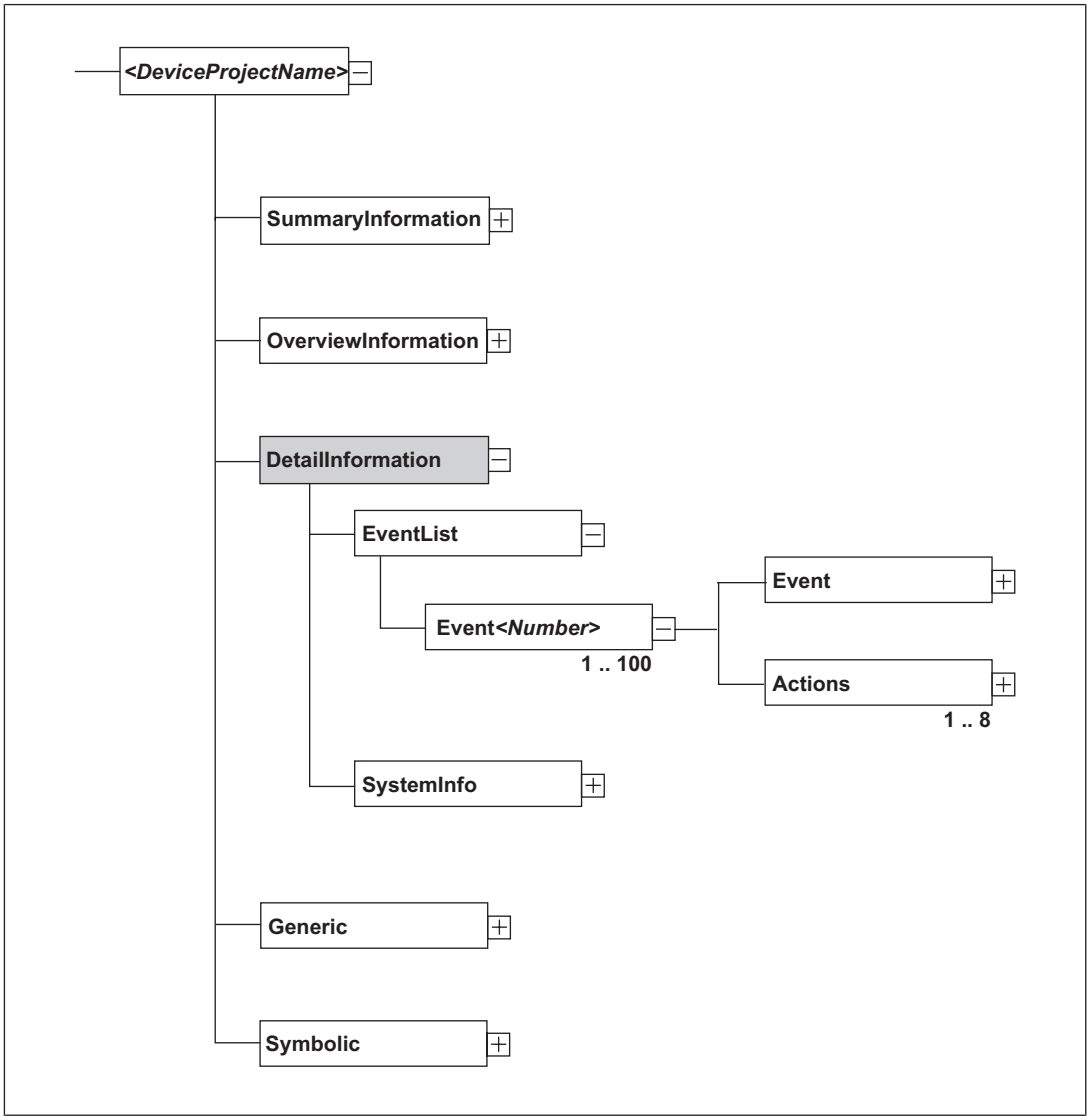

Fig.: Area <*DeviceProjectName*>/DetailInformation/

Node name	Meaning
DetailInformation	Area containing detailed information about entries in the event list
Path: <*DeviceProjectName*>/DetailInformation/	

Node name	Meaning
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: <*DeviceProjectName*>/DetailInformation/EventList/	
Event<*Number*>	Number of the event (node name, e.g. "Event 8")
Path: <*DeviceProjectName*>/DetailInformation/EventList/Event<*Number*>/	
Event	Details of the event see Event [📖 70]
Path: <DeviceProjectName>/DetailInformation/EventList/Event<*Number*>/Event/	
Actions	Details of the actions of the event see Actions [📖 72]
Path: <DeviceProjectName>/DetailInformation/EventList/Event<*Number*>/Actions/	
SystemInfo	Area containing system information about the diagnostic configuration and PNOZmulti project see SystemInfo [📖 74]
Path: <*DeviceProjectName*>/DetailInformation/SystemInfo/	

10.2.4.1 Event

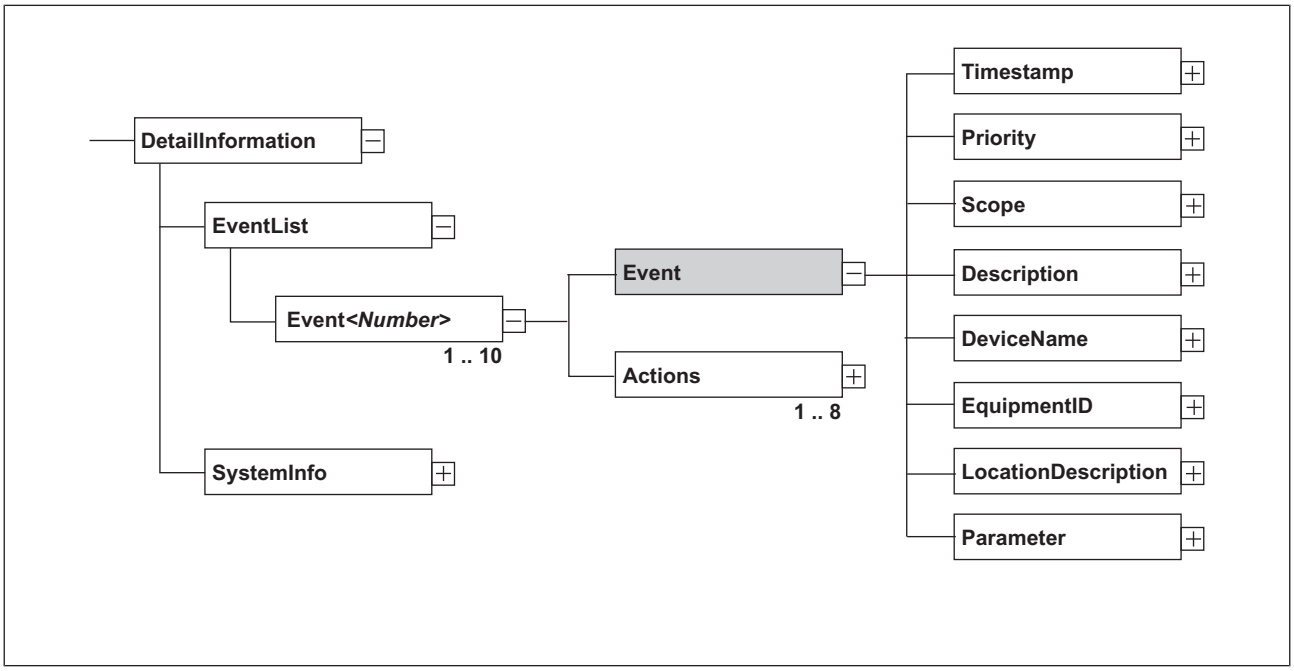

Fig.: Area <*DeviceProjectName*>/DetailInformation/EventList/Event<*Number*>/Event/

Node name		Meaning
Event		Details of the event
Path: <*DeviceProjectName*>/DetailInformation/EventList/Event<*Number*>/Event/		
	Timestamp	Time stamp Data type: String
	Path: <*DeviceProjectName*/DetailInformation/EventList/Event<*Number*>/Event/Timestamp	
	Priority	Priority Data type: String
	Path: <*DeviceProjectName*>/DetailInformation/EventList/Event<*Number*>/Event/Priority	
	Scope	Scope Data type: String
	Path: <*DeviceProjectName*>/DetailInformation/EventList/Event<*Number*>/Event/Scope	
	Description	Description Data type: String
	Path: <*DeviceProjectName*>/DetailInformation/EventList/Event<*Number*>/Event/Description	

Node name		Meaning
	DeviceName	Device name Data type: String
	Path: <*DeviceProjectName*>/DetailInformation/EventList/Event<*Number*>/Event/DeviceName	
	EquipmentID	Equipment identifier (Equip-ID) Data type: String
	Path: <*DeviceProjectName*>/DetailInformation/EventList/Event<*Number*>/Event/EquipmentID	
	LocationDescription	Location description Data type: String
	Path: <*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
	Path: <*DeviceProjectName*>/DetailInformation/EventList/Event<*Number*>/Event/Parameter	

10.2.4.2 Actions

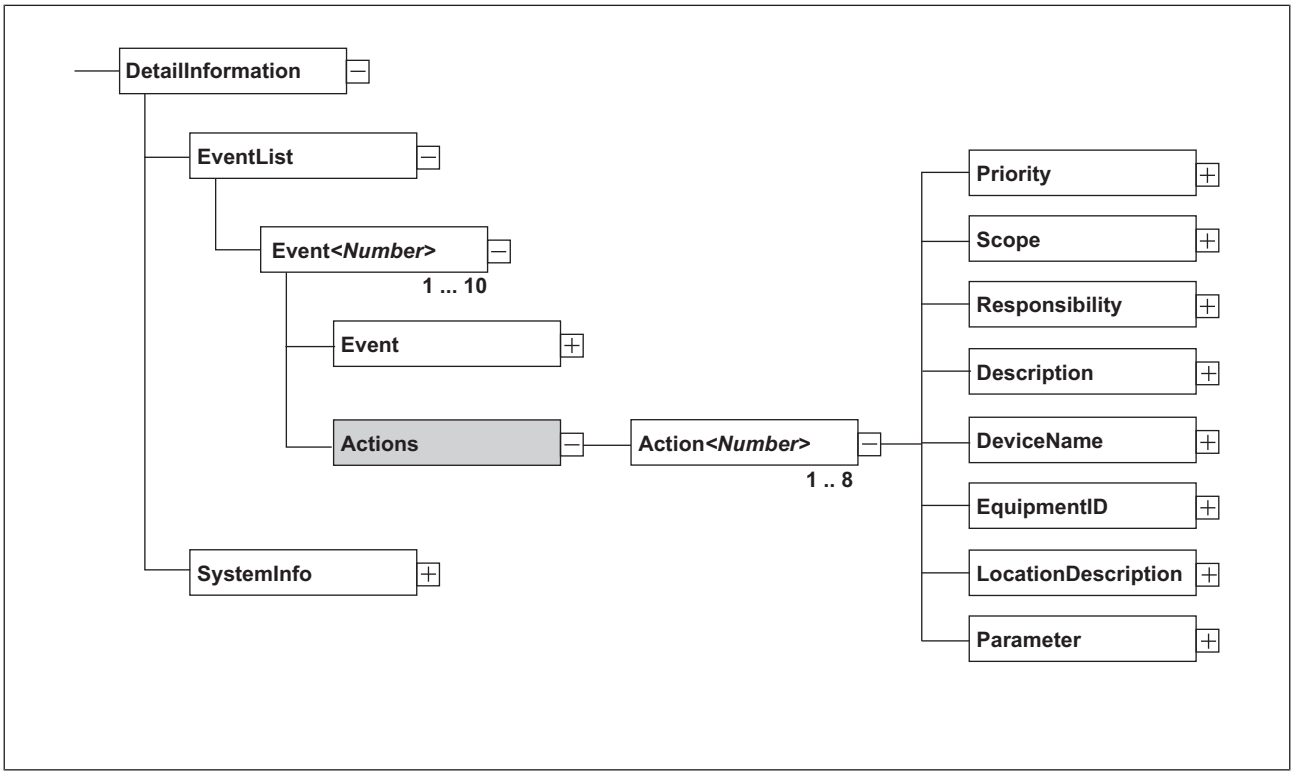

Fig.: Area <*DeviceProjectName*>/DetailInformation/EventList/Event<*Number*>/Actions/

Node name		Meaning
Actions		Details of the actions of an event
Path: <*DeviceProjectName*>/DetailInformation/EventList/Event<*Number*>/Actions/		
Action<*Number*>		Number of the action (Node name e.g. "Action5")
Path: <*DeviceProjectName*>/DetailInformation/EventList/Event<*Number*>/Actions/Action<*Number*>/		
	Priority	Priority Data type: String
	Path: <*DeviceProjectName*>/DetailInformation/EventList/Event<*Number*>/Actions/Action<*Number*>/Priority	
	Scope	Scope Data type: String
	Path: <*DeviceProjectName*>/DetailInformation/EventList/Event<*Number*>/Actions/Action<*Number*>/Scope	
	Responsibility	Responsibility Data type: String
	Path: <*DeviceProjectName*>/DetailInformation/EventList/Event<*Number*>/Actions/Action<*Number*>/Responsibility	

Node name		Meaning
	Description	Description Data type: String
	Path: <*DeviceProjectName*>/DetailInformation/EventList/Event<*Number*>/Actions/Action<*Number*>/Description	
	DeviceName	Device name Data type: String
	Path: <*DeviceProjectName*>/DetailInformation/EventList/Event<*Number*>/Actions/Action<*Number*>/DeviceName	
	EquipmentID	Equipment identifier (Equip-ID) Data type: String
	Path: <*DeviceProjectName*>/DetailInformation/EventList/Event<*Number*>/Actions/Action<*Number*>/EquipmentID	
	LocationDescription	Location description Data type: String
	Path: <*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
	Path: <*DeviceProjectName*>/DetailInformation/EventList/Event<*Number*>/Actions/Action<*Number*>/Parameter	

10.2.4.3 SystemInfo

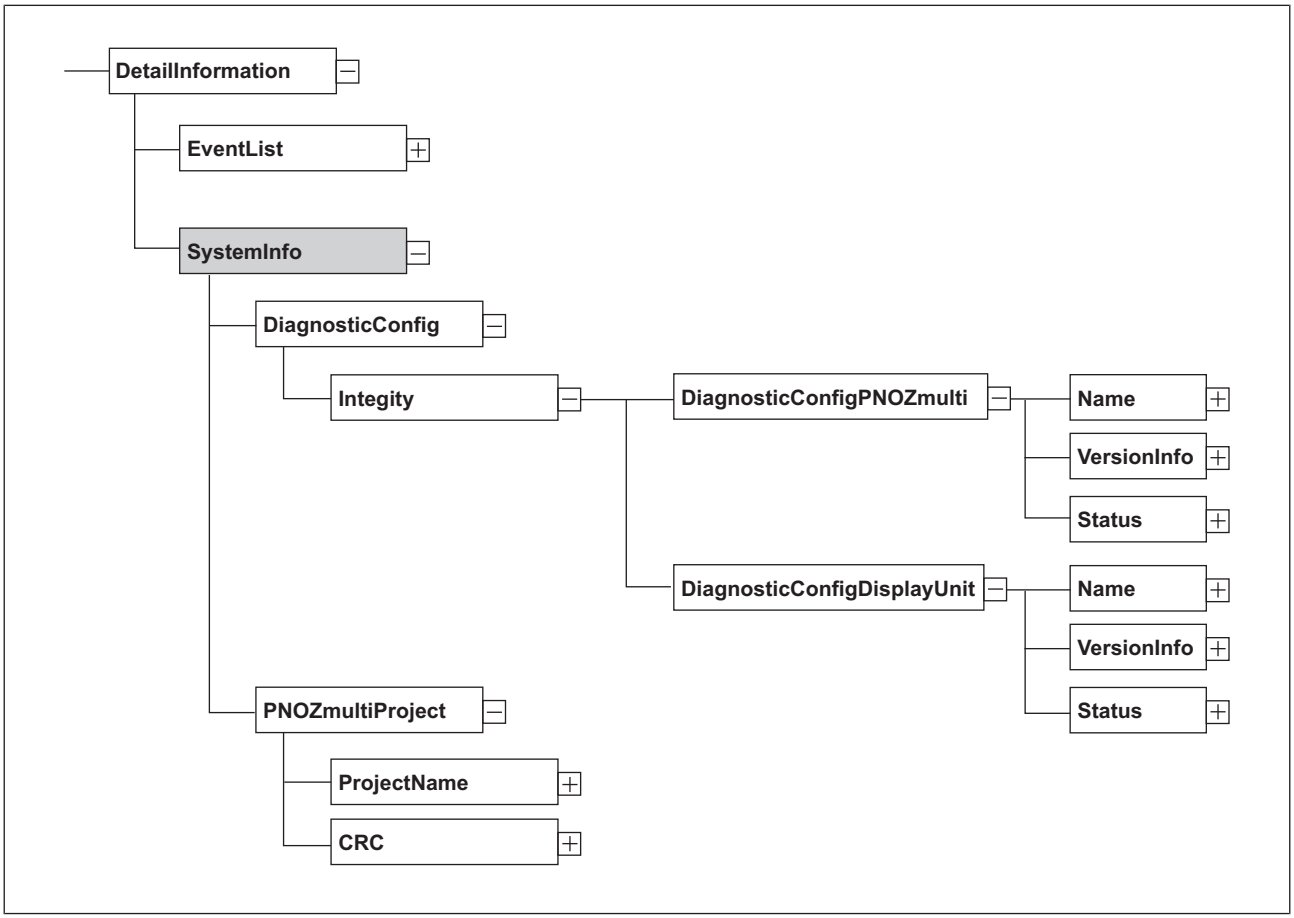

Fig.: Area <*DeviceProjectName*>/DetailInformation/SystemInfo/

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

Area containing information on the diagnostic configuration

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

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

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.
	Path: <*DeviceProjectName*>/DetailInformation/SystemInfo/DiagnosticConfig/Integrity/DiagnosticConfigDisplayUnit/Status		

Area containing information on the PNOZmulti program

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

10.2.5 DeviceProjectName/Generic

The OPC Server supports node identifiers (NodeId) specified by Pilz for generic device data. It is only possible to access the generic data that is also used in the PNOZmulti project.

Node set namespace URI

urn:Pilz:PNOZmultiOPCServer:PNOZmultiNamespace

Structure and content

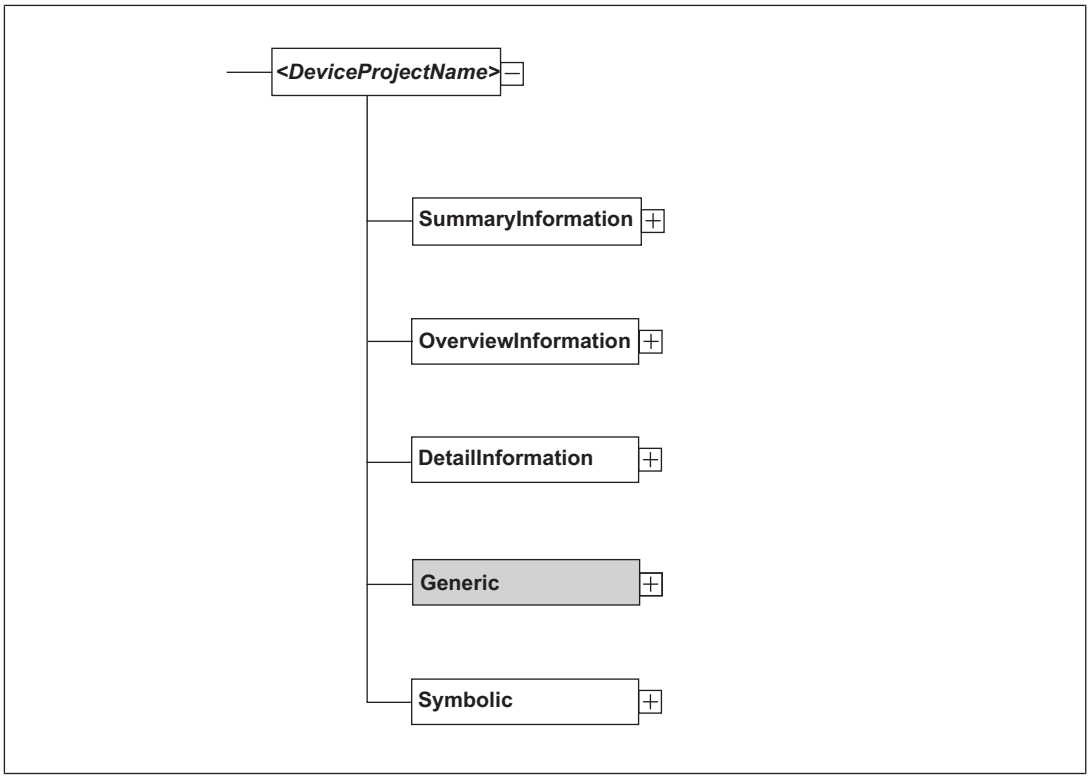

Fig.: Area <*DeviceProjectName*>/Generic/

Node name	Meaning
Generic	Area containing device-specific process data
	Access to the process data via absolute addresses see (Generic [📖 78])
Path: <*DeviceProjectName*>/Generic/	

10.2.5.1 Generic

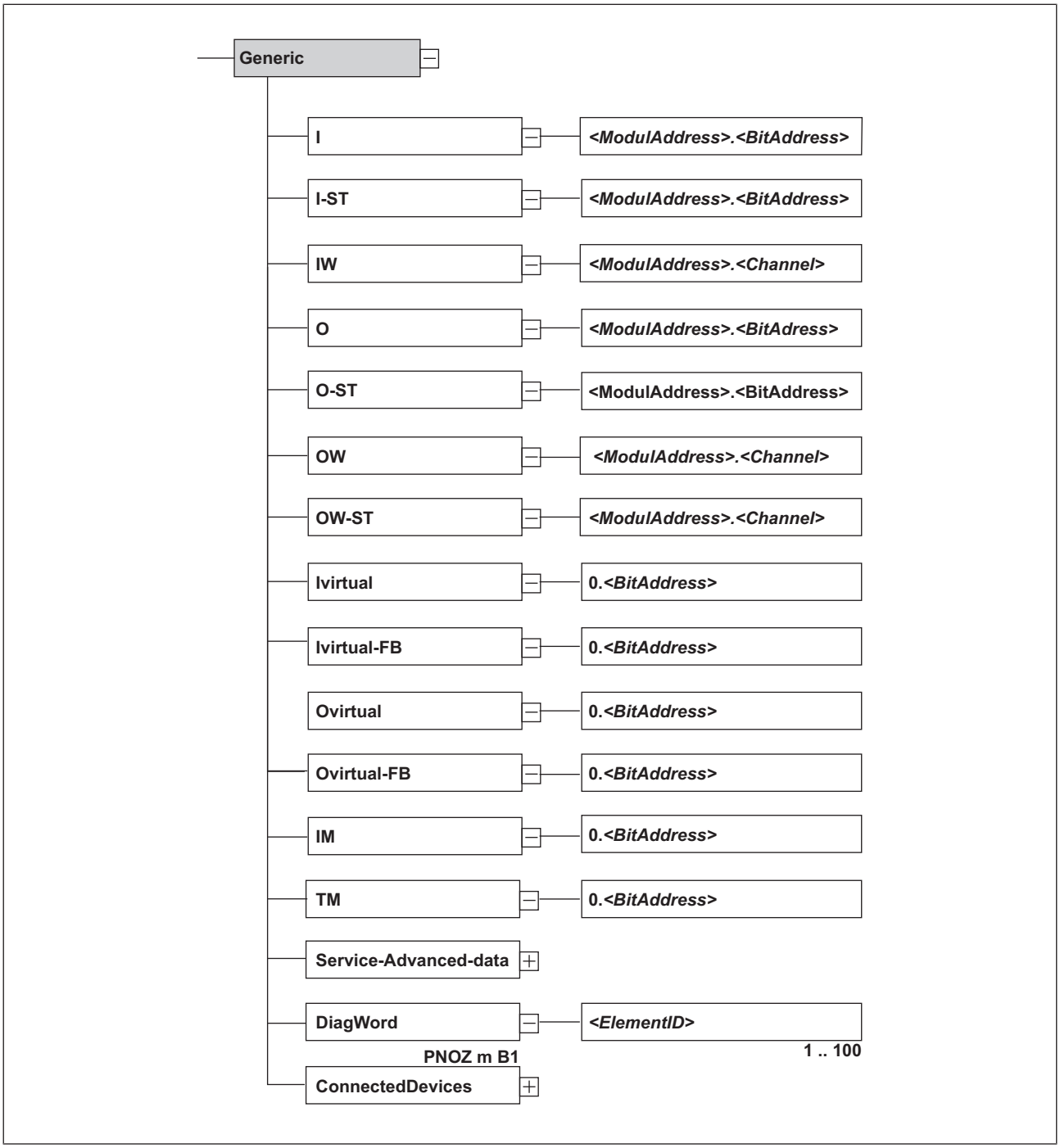

Fig.: Area *<DeviceProjectName>/Generic/*

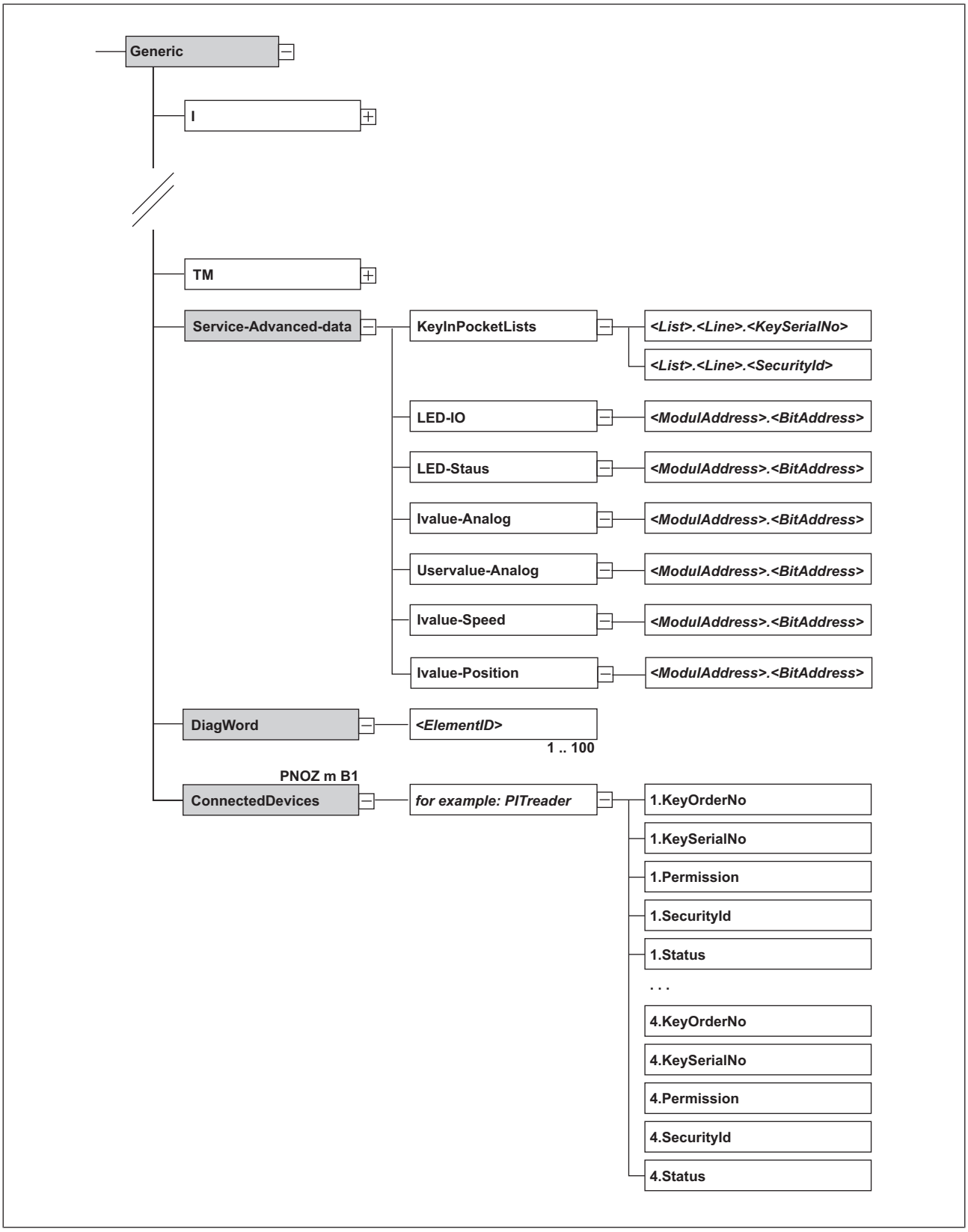

Fig.: Area *<DeviceProjectName>/Generic/Service-Advanced-data, DiagWord* and *ConnectedDevices*

Node name		Meaning
I		Input bits
Path: <*DeviceProjectName*>/Generic/I/		
	<ModuleAddress>.<BitAddress>	▸ PNOZmulti 2 addresses – 0.*<0 .. 19>* e.g.: "I/0.0" – *<-6 .. -2>.<0 .. 63>* e.g.: "I/-5.59" – -1.*<0 .. 127>* e.g.: "I/-1.105" – *<1 .. 12>.<0 .. 31>* e.g.: "I/7.15"
		Data type: Boolean
	Path: <*DeviceProjectName*>/Generic/I/<*ModuleAddress*>.<*BitAddress*>	
I-ST		ST input bits
Path: <*OPCProjectName*>/<*DeviceProjectName*>/Generic/I-ST/		
	<ModuleAddress>.<BitAddress>	PNOZmulti 2 addresses: *<101 .. 112>.<0 .. 31>* e.g. "I-ST/110.15"
		Data type: Boolean
	Path: <*DeviceProjectName*>/Generic/I-ST/<*ModuleAddress*>.<*BitAddress*>	
IW		Input words
Path: <*OPCProjectName*>/<*DeviceProjectName*>/Generic/IW/		
	<ModuleAddress>.<Channel>	▸ PNOZmulti 2 addresses: – *<-6 .. -1>.<0 .. 1>* e.g. "IW/-1.0" – 0.*<0 .. 1>* e.g. "IW/0.0"
		Data type: Int16
	Path: <*DeviceProjectName*>/Generic/IW/<*ModuleAddress*>.<*Channel*>	
O		Output bits
Path: <*OPCProjectName*>/<*DeviceProjectName*>/Generic/O/		

Node name	Meaning
<ModuleAddress>.<BitAddress>	▸ PNOZmulti 2 addresses – 0.*<0 .. 5>* e.g.: "O/0.0" – *<-6 .. -1>.<0 .. 63>* *e.g.: O/-5.15* – *<1 .. 12>.<0 .. 31>* e.g.: "O/7.7"
	Data type: Boolean
Path: *<DeviceProjectName>/Generic/O/<ModuleAddress>.<BitAddress>*	
O-ST	ST output bits
Path: *<OPCProjectName>*/*<DeviceProjectName>*/Generic/O-ST/	
<ModuleAddress>.<BitAddress>	PNOZmulti 2 addresses: *<101 .. 112>.<0 .. 31>* e.g. "O-ST/101.7"
	Data type: Boolean
Path: *<DeviceProjectName>/Generic/O-ST/<ModuleAddress>.<BitAddress>*	
OW	Output words
Path: *<OPCProjectName>*/*<DeviceProjectName>*/Generic/OW/	
<ModuleAddress>.<Channel>	PNOZmulti 2 addresses: *<-6 .. -1>.<0 .. 1>* e.g. "OW/-3.0"
	Data type: Int16
Path: *<DeviceProjectName>/Generic/OW/<ModuleAddress>.<Channel>*	
Isec	Input bit of a safe Ethernet connection
Path: *<DeviceProjectName>*/Generic/Isec/	
0.<BitAddress>	PNOZmulti 2 addresses: 0.*<0 .. 47>* e.g.: "Isec/0.40"
	Data type: Boolean
Path: *<DeviceProjectName>/Generic/Isec/0.<BitAddress>*	
Osec	Output bit of a safe Ethernet connection
Path: *<DeviceProjectName>*/Generic/Osec/	
0.<BitAddress>	PNOZmulti 2 addresses: 0.*<0 .. 47>* e.g. "Osec/0.7"
	Data type: Boolean
Path: *<DeviceProjectName>/Generic/Osec/0.<BitAddress>*	
Ivirtual	Input bits

Node name	Meaning
Path: <*DeviceProjectName*>/Generic/Ivirtual/	
0.<BitAddress>	PNOZmulti 2 addresses: 0.<*0 .. 127*> e.g. "Ivirtual/0.125"
	Note: ▸ PNOZ m B0: Either Ivirtual/Ovirtual or Ivirtual-FB/Ovirtual-FB may be displayed. ▸ PNOZ m B1: Ivirtual/Ovirtual and Ivirtual-FB/Ovirtual-FB can be displayed simultaneously.
	Data type: Boolean
Path: <*DeviceProjectName*>/Generic/Ivirtual/0.<*BitAddress*>	
Ivirtual-FB	Input bits of a fieldbus
Path: <*OPCProjectName*>/<*DeviceProjectName*>/Generic/Ivirtual-FB/	
0.<BitAddress>	PNOZmulti 2 addresses: 0.<*0 .. 127*> e.g. "Ivirtual-FB/0.63"
	Note: ▸ PNOZ m B0: Either Ivirtual/Ovirtual or Ivirtual-FB/Ovirtual-FB may be displayed. ▸ PNOZ m B1: Ivirtual/Ovirtual and Ivirtual-FB/Ovirtual-FB can be displayed simultaneously.
	Data type: Boolean
Path: <*DeviceProjectName*>/Generic/Ivirtual-FB/0.<*BitAddress*>	
Ovirtual	Output bits
Path: <*OPCProjectName*>/<*DeviceProjectName*>/Generic/Ovirtual/	

Node name		Meaning
	0.<BitAddress>	PNOZmulti 2 addresses: 0.*<0 .. 127>* e.g. "Ovirtual/0.127"
		Note: ▸ PNOZ m B0: Either Ivirtual/Ovirtual or Ivirtual-FB/Ovirtual-FB may be displayed. ▸ PNOZ m B1: Ivirtual/Ovirtual and Ivirtual-FB/Ovirtual-FB can be displayed simultaneously.
		Data type: Boolean
	Path: *<DeviceProjectName>/Generic/Ovirtual/0.<BitAddress>*	
Ovirtual-FB		Output bits on a fieldbus
Path: *<DeviceProjectName>*/Generic/Ovirtual-FB/		
	0.<BitAddress>	PNOZmulti 2 addresses: 0.*<0 .. 127>* e.g. "Ovirtual-FB/0.27"
		Note: ▸ PNOZ m B0: Either Ivirtual/Ovirtual or Ivirtual-FB/Ovirtual-FB may be displayed. ▸ PNOZ m B1: Ivirtual/Ovirtual and Ivirtual-FB/Ovirtual-FB can be displayed simultaneously.
		Data type: Boolean
	Path: *<DeviceProjectName>/Generic/Ovirtual-FB/0.<BitAddress>*	
IM		Configurable input/output bits
Path: *<DeviceProjectName>*/Generic/IM/		
	0.<BitAddress>	▸ PNOZmulti 2 addresses: – 0.*<0 .. 3>* e.g. "IM/0.2" – 0.*<16 .. 19>* e.g. "IM/0.16"
		Data type: Boolean
	Path: *<DeviceProjectName>*/Generic/IM/*0.<BitAddress>*	
TM		Configurable test pulse/auxiliary output bits
Path: *<DeviceProjectName>*/Generic/TM/		

Node name	Meaning
0.<BitAddress>	PNOZmulti 2 addresses: 0.*<20 .. 23>* e.g. "TM/0.20"
	Data type: Boolean
Path: *<DeviceProjectName>*/Generic/TM/*0.<BitAddress>*	
Service-Advanced-data	Areas with additional diagnostic data
Path: *<DeviceProjectName>/Generic/Service-Advanced-data/*	
KeyInPocketLists	
Path: *<DeviceProjectName>/Generic/Service-Advanced-data/KeyInPocketLists/*	
<List>.<Line>.<KeySerialNo>	Addresses: <1 .. 4>.<01 .. 20> e.g. "1.05.KeySerialNo"
	Data type: UInt32
Path: *<DeviceProjectName>/Generic/Service-Advanced-data/KeyInPocketLists/<List>.<Line>.<KeySerialNo>*	
<List>.<Line>.<SecurityId>	Addresses: <1 .. 4>.<01 .. 20> e.g. "1.05.SecurityId"
	Data type: UInt64
Path: *<DeviceProjectName>/Generic/Service-Advanced-data/KeyInPocketLists/<List>.<Line>.<SecurityID>*	
LED-IO	Status of a module's I/O LEDs
Path: *<DeviceProjectName>/Generic/Service-Advanced-data/LED-IO/*	
<ModuleAddress>.<BitAddress>	Addresses: *<0 .. 12>.<0 .. 31>* e.g. "LED-IO/7.15"
Data type: UInt8	
Path:*<OPCProjectName>/<DeviceProjectName>/Generic/Service-Advanced-data/LED-IO/<ModuleAddress>.<BitAddress>*	
LED status	Status of a module's status LEDs
Path: *<DeviceProjectName>/Generic/Service-Advanced-data/LED-Status*	
<ModuleAddress>.<BitAddress>	Addresses: *<-6 .. 12>.<0 .. 7>* e.g. "LED-Status/7.7"
Data type: UInt8	
Path: *<DeviceProjectName>/Generic/Service-Advanced-data/LED-Status/<ModuleAddress>.<BitAddress>*	

Node name		Meaning
Ivalue-Analog		Numeric input value of an analogue module
Path: *<DeviceProjectName>/Generic/Service-Advanced-data/Ivalue-Analog/*		
	<ModuleAddress>.<BitAddress>	Addresses: *<1 .. 12>.<0 .. 3>* e.g. "Ivalue-Analog/7.0"
	Data type: Float	
	Path: *<DeviceProjectName>/Generic/Service-Advanced-data/Ivalue-Analog/<ModuleAddress>.<BitAddress>*	
Uservalue-Analog		Scaled input value of an analogue module
Path: *<DeviceProjectName>/Generic/Service-Advanced-data/Uservalue-Analog/*		
	<ModuleAddress>.<BitAddress>	Addresses: *<1 .. 12>.<1 .. 6>* e.g. "Uservalue-Analog/7.6"
	Data type: Float	
	Path: *<DeviceProjectName>/Generic/Service-Advanced-data/Uservalue-Analog/<ModuleAddress>.<BitAddress>*	
Ivalue-Speed		Input value (speed) of a motion monitoring module
	<ModuleAddress>.<BitAddress>	Addresses: *<1 .. 12>.<1, 2>* e.g. "Ivalue-Speed/7.1"
	Data type: UInt32	
	Path: *<DeviceProjectName>/Generic/Service-Advanced-data/Ivalue-Speed/<ModuleAddress>.<BitAddress>*	
Ivalue-Position		Input value (position) of a motion monitoring module
	<ModuleAddress>.<BitAddress>	Addresses: *<1 .. 12>.<1, 2>* e.g. "Ivalue-Position/7.2"
	Data type: UInt32	
	Path: *<DeviceProjectName>/Generic/Service-Advanced-data/Ivalue-Position/<ModuleAddress>.<BitAddress>*	
DiagWord		Area to display diagnostic data (e.g. operating status, error messages)
Path: *<DeviceProjectName>/Generic/DiagWord/*		
<ElementID>		Element ID of a diagnostic word
		Data type: UInt16
Path: *<DeviceProjectName>/Generic/DiagWord/<Element-ID>*		
ConnectedDevices		Area for the diagnostic data of devices that are connected to a PNOZ m B1. Note: The sequence of the variables depends on the options for displaying the employed OPC Client.
Path: *<DeviceProjectName>/Generic/ConnectedDevices/*		

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

NOTICE

Note:

– 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).

10.2.6 DeviceProjectName/Symbolic

In the symbolic namespace, data on a PNOZmulti 2 device can be accessed via the equipment identifier. This type of access is called symbolic access.

INFORMATION

Requirements of equipment identifiers for symbolic access

When the OPC Server is used, equipment identifiers in a PNOZmulti project must be unique, i.e. an equipment identifier must not be assigned twice.

Node set namespace URI

urn:Pilz:PNOZmultiOPCServer:PNOZmultiNamespace:Symbolic

Structure and content

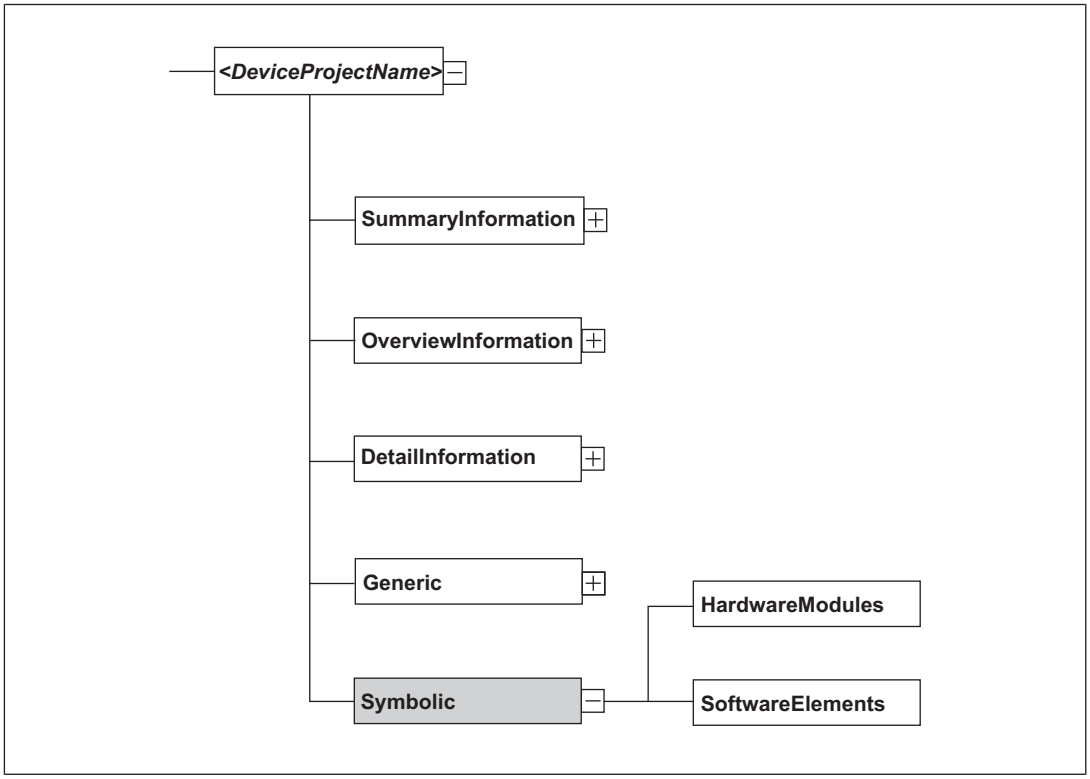

Fig.: Area *<DeviceProjectName>*/Symbolic

Node name	Meaning
Symbolic	Area for access to process and diagnostic data via equipment identifiers (Equip-ID)
	Symbolic access
Path: *<DeviceProjectName>*/Symbolic/	

Node name		Meaning
	HardwareModules	Symbolic access to the modules that were configured in the PNOZmulti Configurator during the hardware configuration of a PNOZmulti project see HardwareModules [📖 89]
	Path: <*DeviceProjectName*>/Symbolic/HardwareModules	
	SoftwareElements	Symbolic access to the elements that were configured in the PNOZmulti Configurator of a PNOZmulti project see SoftwareElements [📖 91]
	Path: <*DeviceProjectName*>/Symbolic/SoftwareElements	

10.2.6.1 HardwareModules

Equipment identifiers for hardware modules can be assigned for a PNOZmulti 2 device in the PNOZmulti Configurator.

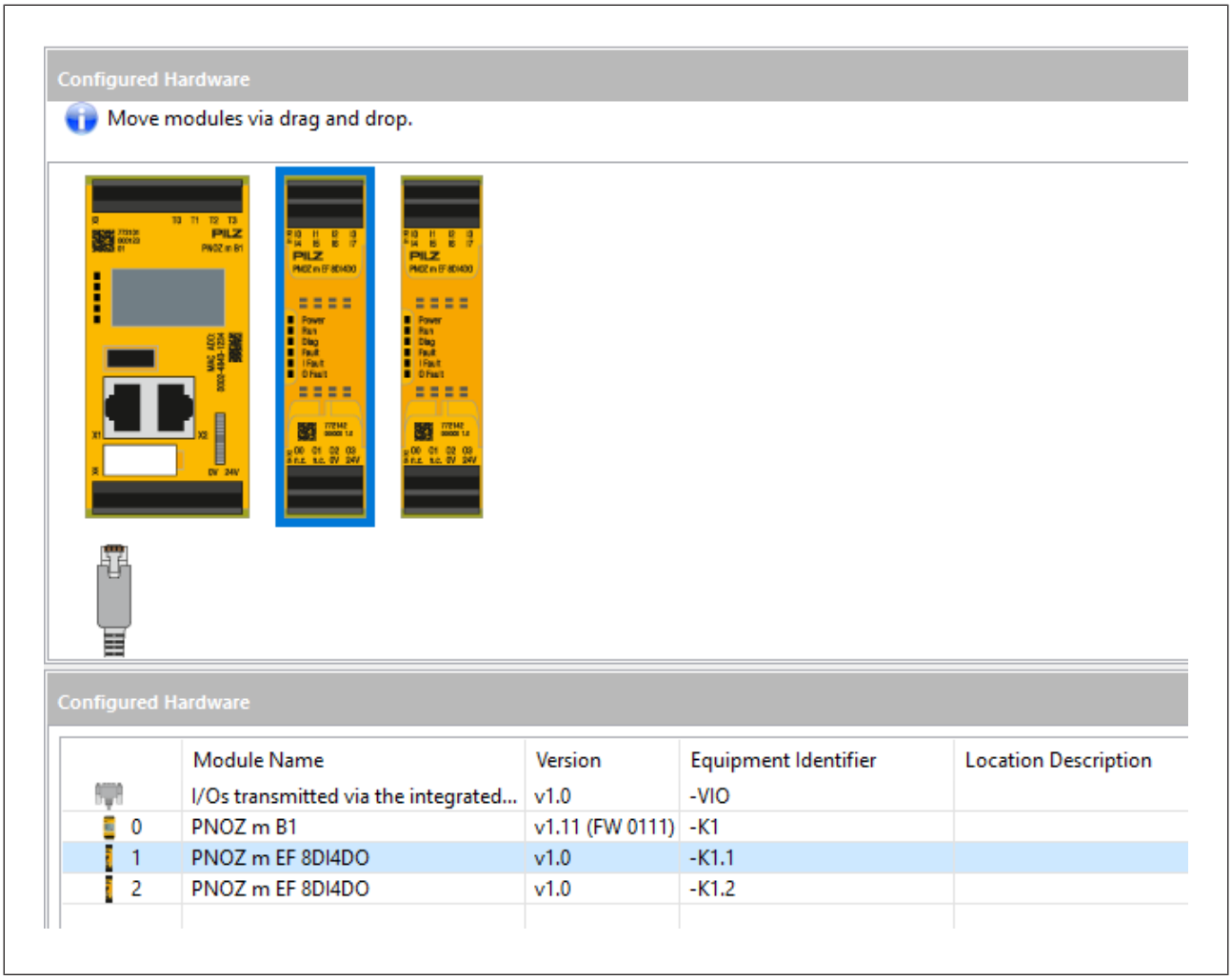

Fig.: Equipment identifiers for a module (PNOZmulti Configurator)

The equipment identifiers for a hardware module are displayed in the symbolic namespace of the OPC UA Server. The process data of the modules configured for a PNOZmulti project can be accessed in the symbolic namespace using the equipment identifiers.

In the ***UAExpert*** test client from Unified Automation, symbolic access is illustrated as follows, for example:

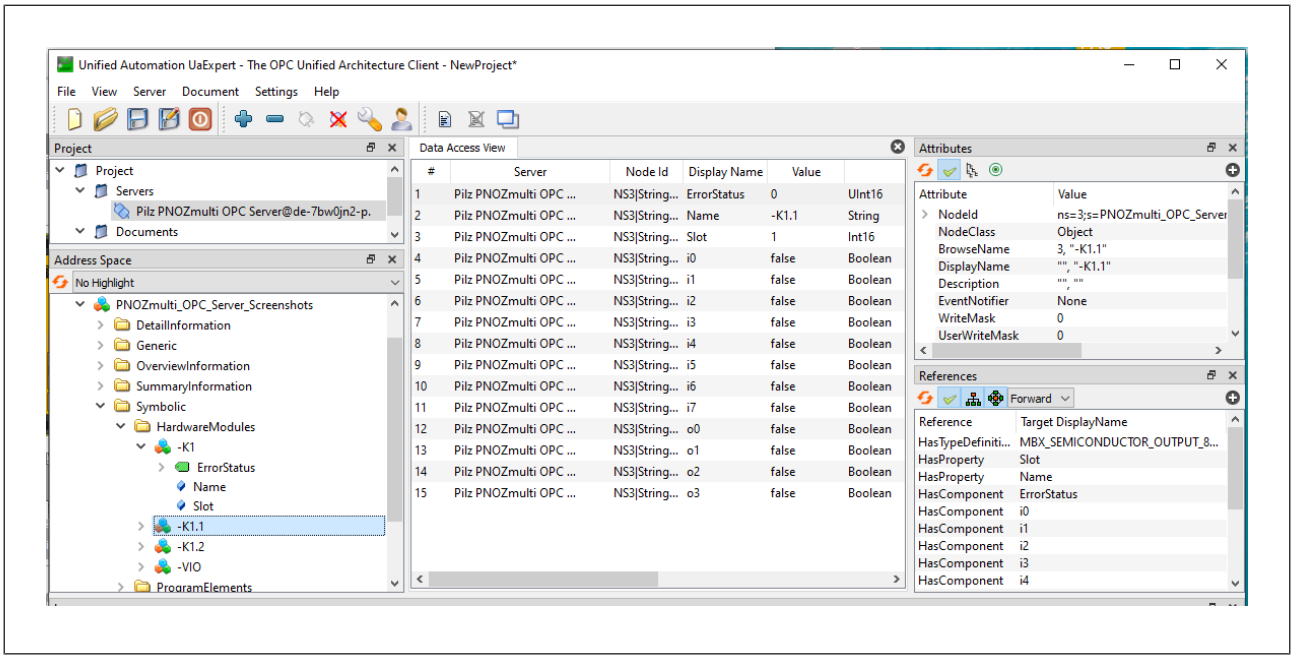

Fig.: Symbolic access to process data via a module's equipment identifiers (***UAExpert*** as example)

The ***UAExpert*** test client from Unified Automation is available under https://www.unified-automation.com/products/development-tools/uaexpert.html.

10.2.6.2 SoftwareElements

Equipment identifiers for software elements can be assigned for a PNOZmulti 2 device in the PNOZmulti Configurator.

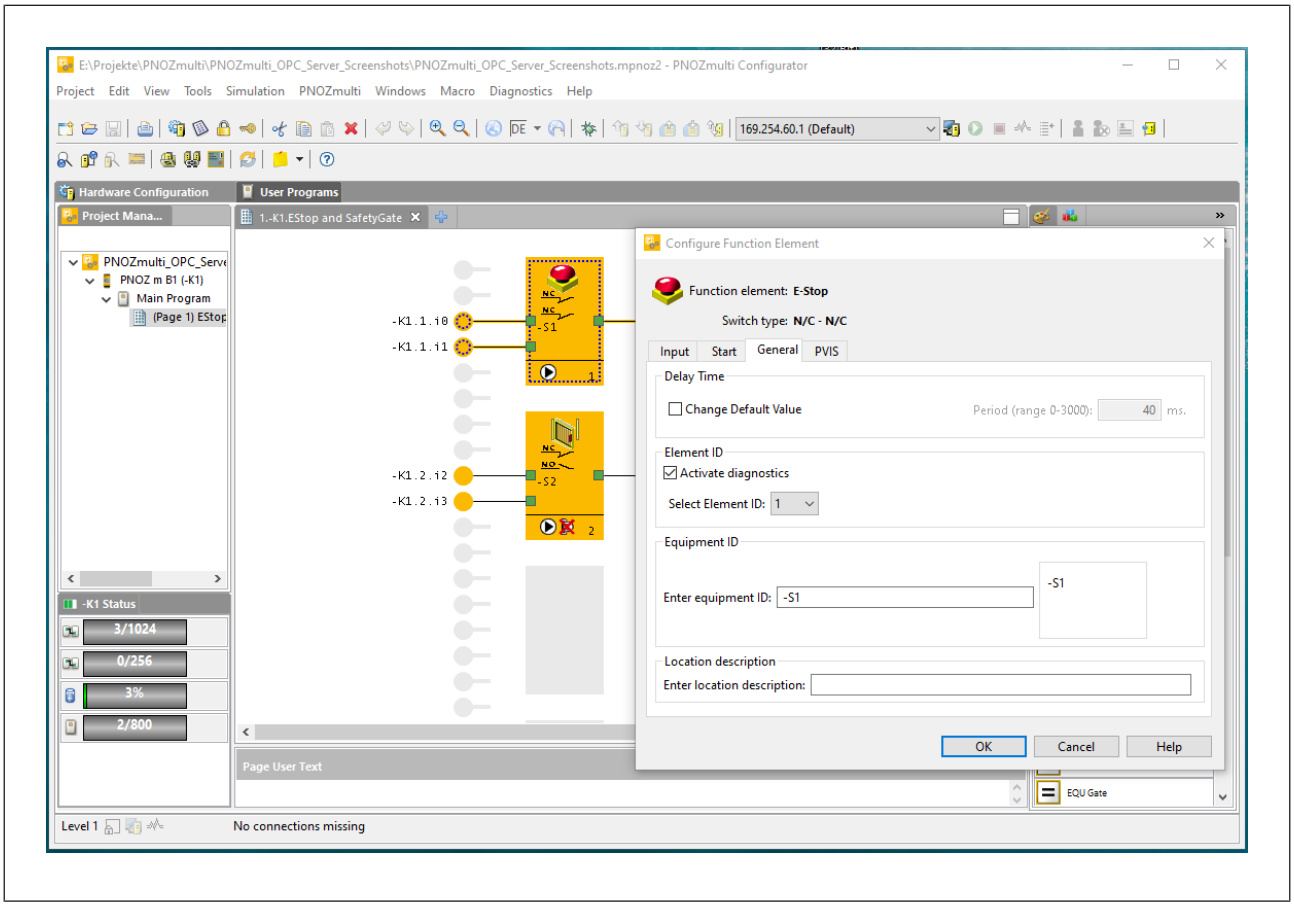

Fig.: Equipment identifiers for a software element (PNOZmulti Configurator)

The equipment identifiers for a PNOZmulti 2 device are displayed in the symbolic namespace of the OPC UA Server. The process data of the software elements in a PNOZmulti project can be accessed in the symbolic namespace using the equipment identifiers.

In the ***UAExpert*** test client from Unified Automation, symbolic access is illustrated as follows, for example:

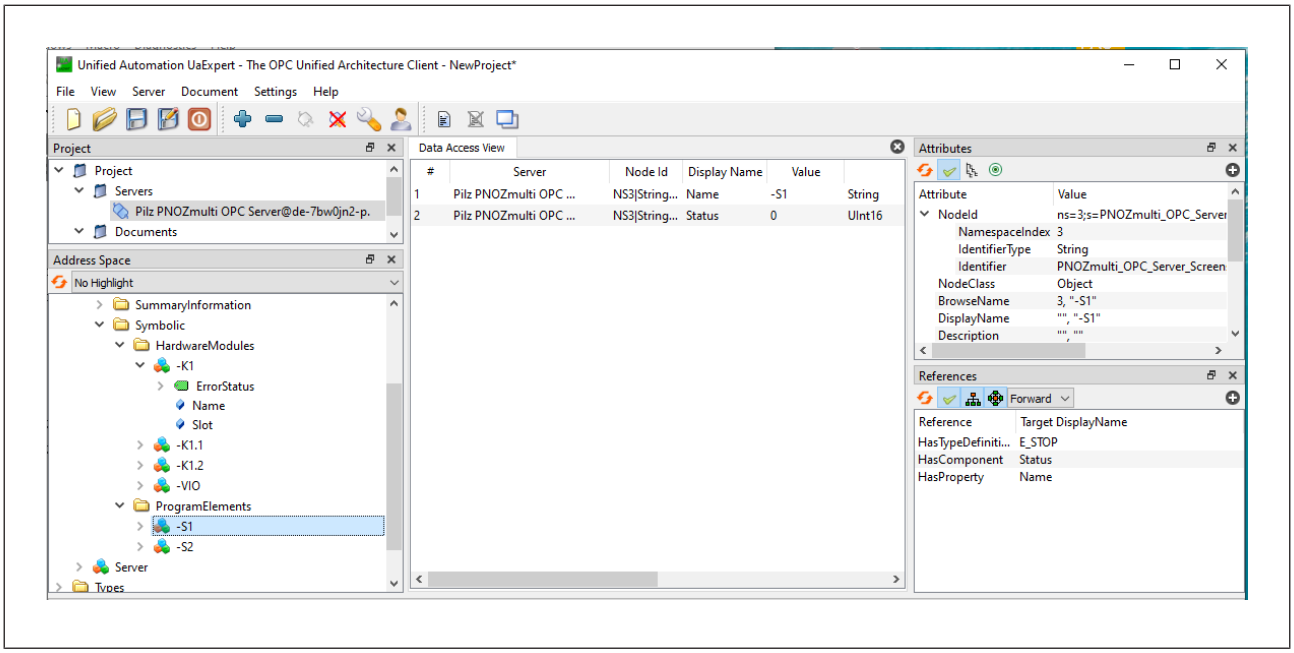

Fig.: Symbolic access to process data via a software element's equipment identifiers (***UAExpert*** as example)

The ***UAExpert*** test client from Unified Automation is available under https://www.unified-automation.com/products/development-tools/uaexpert.html.

11 Event log

OPC UA Events and HDA (Historical Data Access) Services are used for access to entries in the event log (EventLog).

Access to the entries in the event list (Events) and event history is explained and illustrated below using the ***UAExpert*** test client from Unified Automation (see https://www.unified-automation.com/products/development-tools/uaexpert.html).

11.1 Activate event view

The event list (Events) and event history can be displayed in the ***UAExpert***. To do this, the event view must be displayed after ***UAExpert*** is started.

Procedure

- In the menu bar, select ***Document*** [1] ***-> Add*** and, in the ***Add Document*** window under ***Document Type***, select ***Event View***.

  The event view [2] is displayed.

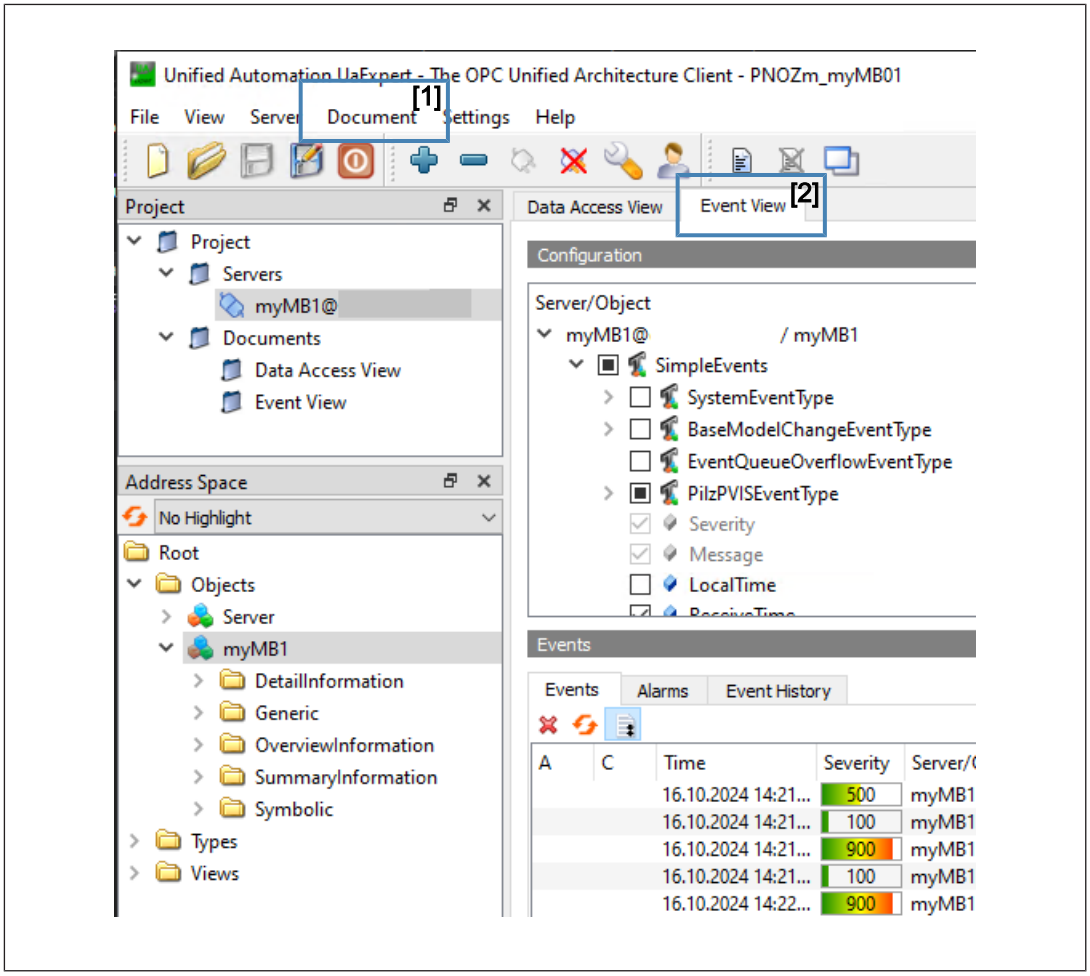

Fig.: Display event view in ***UAExpert***

11.2 Subscribe Client to Server's event source

To access diagnostic data, the Client must be subscribed to the Server's event source.

Prerequisite

- The event view has been activated (see Activate event view [94]).

Procedure

- In the project view, select the required OPC UA Server [1]
- Using drag&drop, drag the required project [2] from the namespace to ***Event View -> Configuration*** [3].

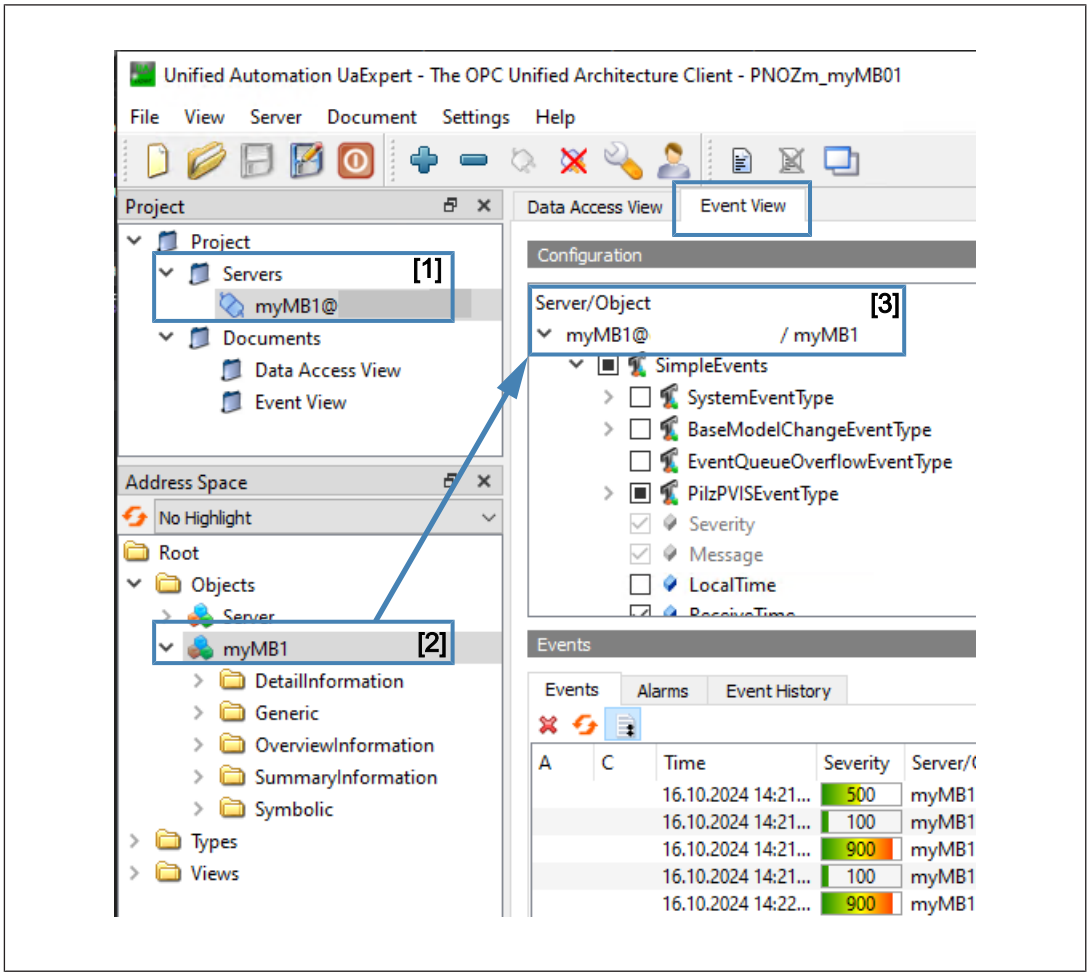

Fig.: Subscribe Client to Server's event source

11.3 Access currently occurring entries

For all events that result in an entry in the event list, an OPC UA event of the type "PilzPVISEventType" is immediately sent from the OPC Server to all OPC Clients that have subscribed to the event source.

The OPC Client can evaluate the diagnostic information.

Subscription to the OPC Server's event source is via the project name in the namespace:

- NodeID:

  urn:Pilz:PNOZmultiOPCServer:PNOZmultiNamespace; s=<*Project name*>.

Prerequisites

- The event view has been activated (see Activate event view [94]).
- The Client has been subscribed to an event source (see Subscribe Client to Server's event source [95]).

Procedure

- To display the entries select ***Event View -> Events -> Events*** [1].

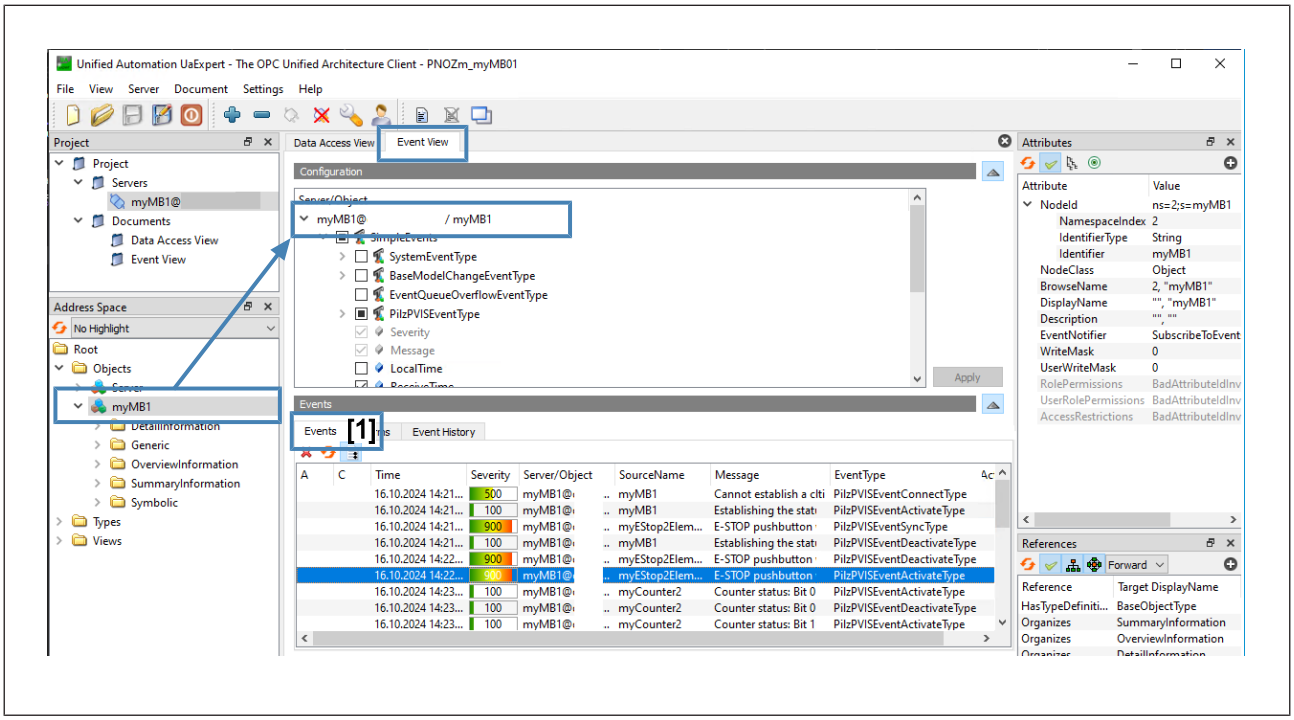

Fig.: Subscription to the OPC Server's event source

11.4 Access an entry in the event list:

The OPC UA EventType "PilzPVISEventType" is defined for access to an entry in the event list. The EventType is derived from a simple OPC UA BaseEvent. The PilzPVISEventType contains all OPC UA-specific and PVIS diagnostic-specific information.

Prerequisites

- The event view has been activated (see Activate event view [94]).
- The Client has been subscribed to an event source (see Subscribe Client to Server's event source [95]).

Procedure

- Activation of EventType

  Switch to ***Event View -> Configuration ->*** *<Project name>* and activate ***PilzPVISEvent-Type*** [1].

- Display area with OPC UA-specific information

  Switch to ***Event View -> Events -> Events*** [2] to display OPC UA-specific information [3].

- Display detailed information about an individual entry

  Select an item of OPC UA-specific information [3] and switch to ***Event View ->Details*** [4] to display the corresponding PVIS diagnostic-specific information.

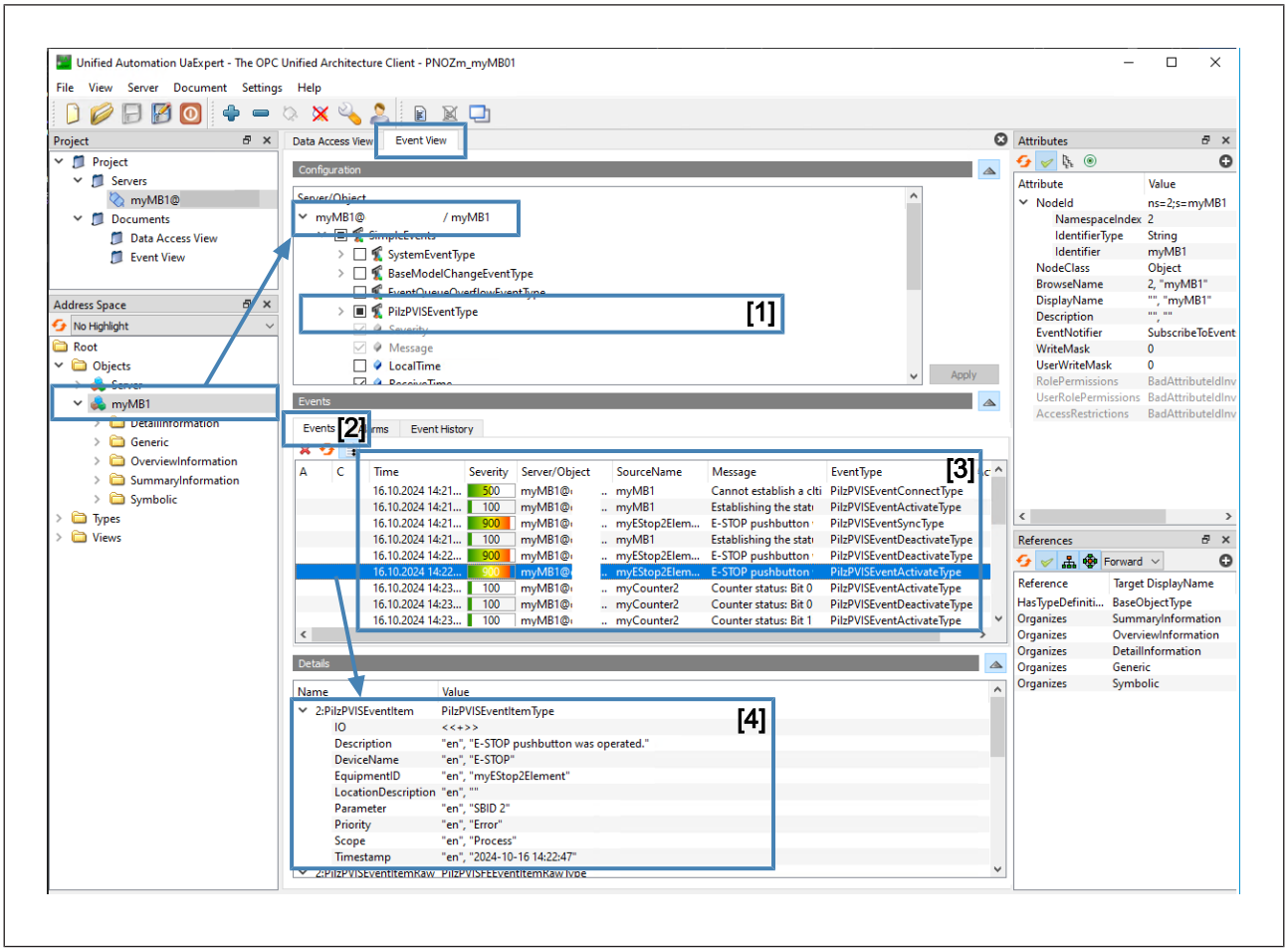

Fig.: View of the event list with specific information

Detailed information about an individual entry [4]

PilzPVISEventItem	Meaning	
IO	<<+>>	New event message that was reported by the connected device
	<<->>	Event message that was reported as cleared by the connected device
	<+>	Event message that already existed on the connected device and has been reported by a new connection (synchronisation)
	<->	Event message that was reported by a connected device but can no longer be monitored as the connection has been lost
	+	From this event onwards, the connection to the connected device has been established; i.e. diagnostics are active.
	-	From this event onwards, the connection to the connected device has been lost; i.e. diagnostics are inactive.
Description	Description e.g.: E-STOP pushbutton was operated	
DeviceName	Device name e.g.: E-STOP	
EquipmentID	Equipment identifier (Equip-ID) e.g.: myEStop2Element	
LocationDescription	Location description e.g.: E-STOP Plant 1	
Parameter	More precise information on where an event occurred e.g.: SBID2	
Priority	Name of priority e.g.: Error	
Scope	Scope e.g.: Area	
Timestamp	Time stamp e.g.: 2024-10-16 14:22:47	

11.5 Access stored entries

Events that have generated an entry in the persistent event list can be read from the OPC Server using the OPC UA HistoryRead service. The persistent event list is a chronological list of OPC UA events of the OPC UA EventType "PilzPVISEventType".

All OPC UA Clients that have subscribed to the OPC Server's event source can request and evaluate these persistent OPC UA events.

Subscription to the OPC Server's event source is via the project name in the namespace:

- NodeID:

  urn:Pilz:PNOZmultiOPCServer:PNOZmultiNamespace; s=<*Project name*>

Prerequisites

- The event view has been activated (see Activate event view [📖 94]).
- The Client has been subscribed to an event source (see Subscribe Client to Server's event source [📖 95]).

Procedure

- Activation of EventType

  Switch to ***Event View -> Configuration ->*** <*Project name*> [1] to display the EventType.

- Start enquiry of persistent OPC UA events
  - Switch to ***Event View -> Events -> Event History*** [2].
  - Select the time range via ***Start Time*** and ***End Time*** [3].
  - Start the enquiry via the refresh button [4].

  The list of all OPC UA-specific and PVIS diagnostic-specific information is shown in the relevant fields.

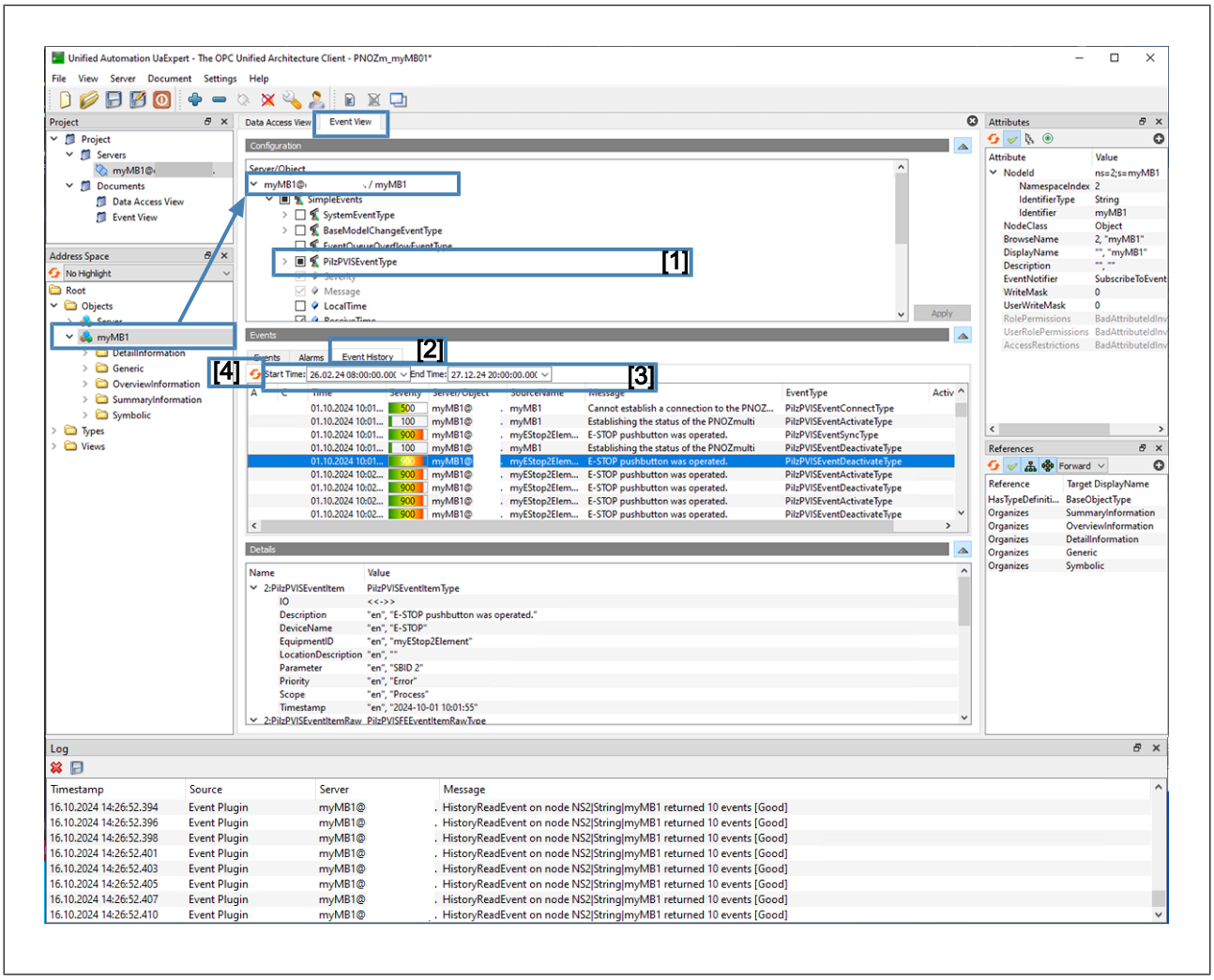

Fig.: View of stored entries

12 Language management

In the PNOZmulti Configurator, the user (e.g. project engineer, system integrator) defines the languages that are to be downloaded to the PNOZmulti OPC Server. These languages must be available in each device project. Each device project can support different languages. The user defines the languages for the device projects when they create the diagnostic configuration in the 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 purpose of the default language is as follows: if an OPC Client (e.g. visualisation software) requests event messages in a language that is unavailable, the OPC Server provides the event messages in the default language.

Example: Certain languages have been selected for an PNOZmulti project. The OPC UA Client calls up information from the OPC UA 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.

13 Network data

Protocol	Direction [*]	Transport protocol	Port no.	Can be deactivated	Time-critical	Description
OPC UA DA	In/out	TCP (IPv4)	1 … 65535 Default: 4862	Yes Default: active	No	**Affected devices:** ▸ PMI/PC **Implemented security measures:** ▸ Mutual authentication of PNOZmulti OPC Server and the employed OPC UA Client via certificates ▸ Certificate in accordance with standard X.509 with signature and encryption method
Local Discovery Server ME (mDNS)	In/out	UDP (IPv4)	5353	No	No	**Affected devices:** ▸ PMI/PC
PNOZmulti PG Interface	In/out	TCP (IPv4)	9000	Yes Default: active	No	**Affected devices:** ▸ PMI/PC ▸ PNOZmulti (programming interface) **Application:** ▸ Download diagnostic data
Syslog	In/out	UDP (IPv4)	514	No	No	**Affected devices:** ▸ PMI/PC **Application:** ▸ Send/receive all Syslog events via the Syslog Port through the PNOZmulti OPC Server

[*]

in: The communication partner starts communication with the product.

out: The product starts communication with the communication partner.

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
Scandinavia
+45 74436332
Spain
+34 938497433
Switzerland
+41 62 88979-32
The Netherlands
+31 347 320477
Türkiye
+90 216 5775552
United Kingdom
+44 1536 462203

You can reach our international hotline on:
+49 711 3409-222
support@pilz.com

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.

BLUEcoMPETENCE Alliance Member
Partner of the Engineering Industry Sustainability Initiative

German Standards
BRAND OF THE CENTURY 2022

Quality Management ISO 9001:2015 REG. NO. 2462
Environmental Management ISO 14001 REG. NO. 2462
Energy Management ISO 50001 REG. NO. 2462

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

1007149-EN-01, 2025-04 Printed in Germany