

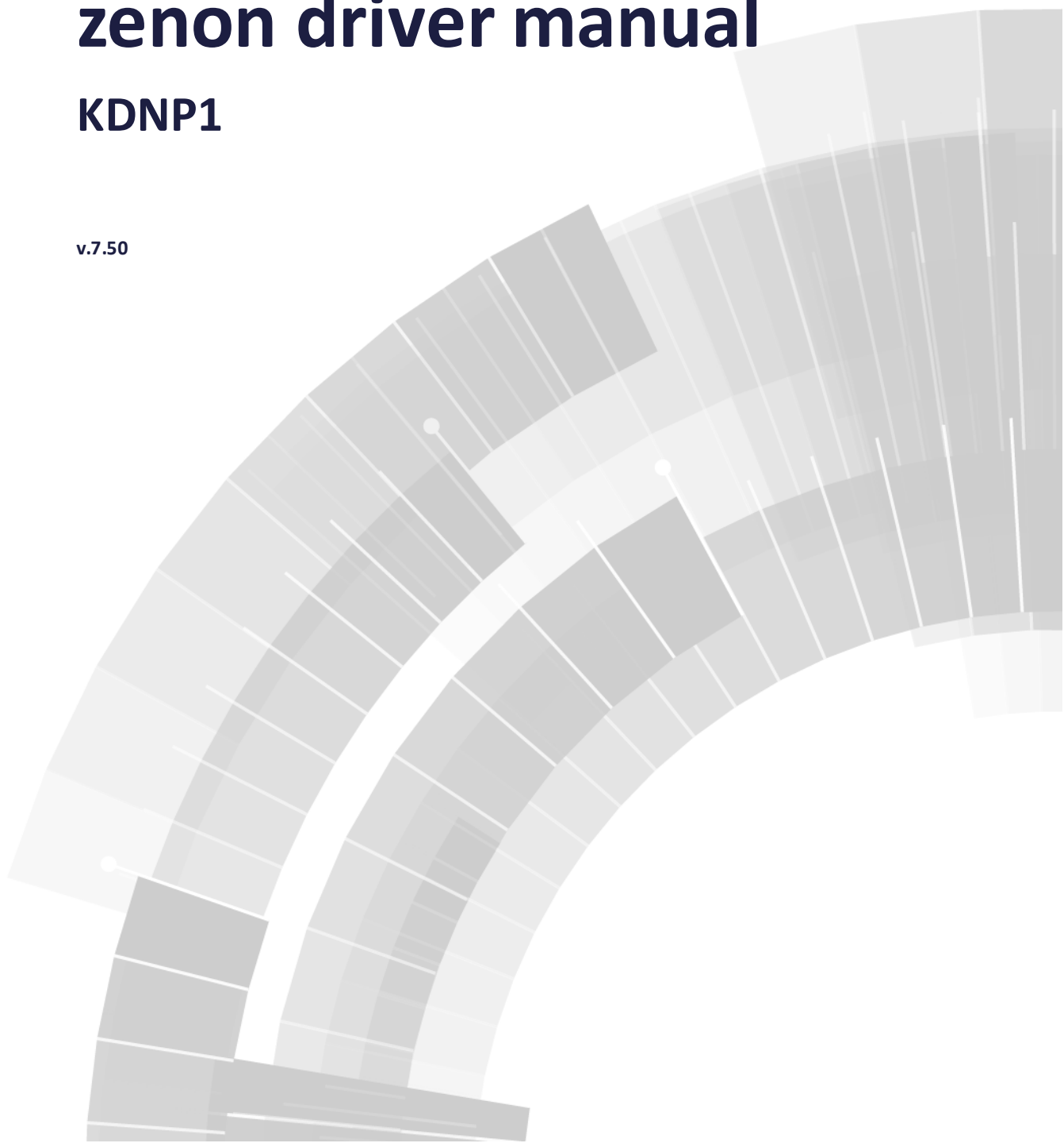


COPADATA
do it your way

zenon driver manual

KDNP1

v.7.50





© 2016 Ing. Punzenberger COPA-DATA GmbH

All rights reserved.

Distribution and/or reproduction of this document or parts thereof in any form are permitted solely with the written permission of the company COPA-DATA. Technical data is only used for product description and are not guaranteed qualities in the legal sense. Subject to change, technical or otherwise.

Contents

1. Welcome to COPA-DATA help	5
2. KDNP1.....	5
3. Driver history	7
4. Requirements.....	7
4.1 PC	7
5. Configuration.....	7
5.1 Creating a driver.....	8
5.2 Settings in the driver dialog.....	9
5.2.1 General	10
5.2.2 Com	13
5.2.3 Options.....	15
5.2.4 Connections	18
6. Creating variables.....	30
6.1 Creating variables in the Editor	30
6.2 Addressing.....	33
6.3 Driver objects and datatypes	34
6.3.1 Driver objects	34
6.3.2 Mapping of the data types	35
6.4 Creating variables by importing.....	40
6.4.1 XML import	40
6.4.2 DBF Import/Export	41
6.5 Driver variables.....	46
7. Driver-specific functions	53
8. Driver commands	55
9. Error analysis.....	57
9.1 Analysis tool	57
9.2 Check list.....	58

1. Welcome to COPA-DATA help

GENERAL HELP

If you cannot find any information you require in this help chapter or can think of anything that you would like added, please send an email to documentation@copadata.com (<mailto:documentation@copadata.com>).

PROJECT SUPPORT

You can receive support for any real project you may have from our Support Team, who you can contact via email at support@copadata.com (<mailto:support@copadata.com>).

LICENSES AND MODULES

If you find that you need other modules or licenses, our staff will be happy to help you. Email sales@copadata.com (<mailto:sales@copadata.com>).

2. KDNP1

The **KDNP1** driver is for communication with Kepco's **KDNP1** protocol, which is based on the DNP3 standard. In doing so, the driver acts as a Master. At protocol level, serial communication and IP communication to several Outstations are supported via TCP.

- ▶ The **KDNP1** driver polls for objects. This polling is cyclical. In addition, this polling can be activated or deactivated with a driver command.
- ▶ In addition, the driver supports communication with Confirmed Data link layer. To do this, the **Data link confirm** property in the **Outstation** driver dialog can be activated. Timeout and the number of attempts can be configured for this property.

DEFINITION OF TERMS

In order for you to understand this document better, please find the definition of important terms in the following list.

Term	Definition
Master	Controlling station. A Master sends read queries and control queries to the Outstation .
Objekt Group	Data type or data type in the Outstation with a defined functionality. Examples of assignments: <ul style="list-style-type: none"> ▶ Object Group 30 is a statistical analog entry. ▶ A value change for an analog input is assigned to Object Group 32. ▶ Object Group 40 is for reading an analog output. ▶ Object Group 41 is for writing to an analog output.
Outstation	SPS or RTU in KDNP1 . An Outstation is a slave at protocol level and sends data to the Master on request.
Point	Equivalent of a variable in the Outstation. A Point is addressed with a Point Number (Offset) per Objekt Group , whereby the following is applicable: Point 12 for Object Group 30 and Point 12 for Object Group 32 both have the same analog input as a basis, but Point 12 for Object Group 1 is a completely-independent binary input. The general term for a value, time or status change is DNP Object .
Variation	Format in which the Outstation saves a static value or a value change in Class 0 or Class 1, 2 or 3. This can be configured in the Outstation for each Objekt Group or for each Point . The Variation defines (e.g.) <ul style="list-style-type: none"> ▶ a static value (Class 0) or a value change, if this is an integer or a floating point. ▶ a time stamp is saved or not, or whether Object Flags are saved or not.

COMMUNICATION

The communication is mainly polling. In doing so, the driver carries out a corresponding Class Poll. Different polling intervals can be defined for binary inputs, binary outputs and analog inputs.

The driver processes the received values and sends them to the Runtime.

3. Driver history

Date	Build number	Change
15.08.15	21913	Created driver documentation

4. Requirements

This chapter contains information on the requirements that are necessary for use of this driver.

4.1 PC

HARDWARE

- ▶ Serial interface
- ▶ Ethernet TCP

CE

The **KDNP1** driver is not available for CE.

5. Configuration

In this chapter you will learn how to use the driver in a project and which settings you can change.



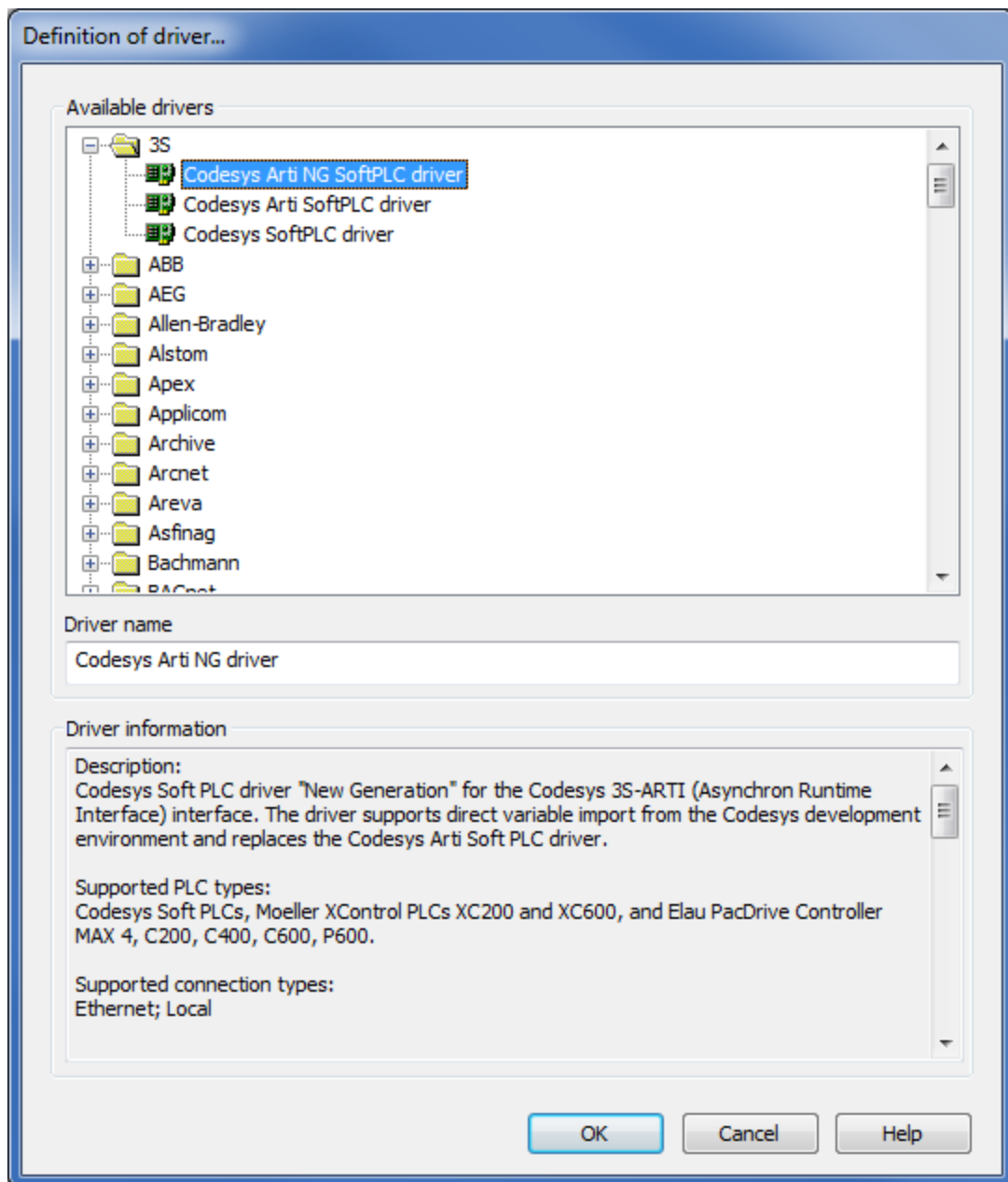
Information

Find out more about further settings for zenon variables in the chapter Variables (main.chm::/15247.htm) of the online manual.

5.1 Creating a driver

In order to create a new driver:

1. Right-click on **Driver** in the Project Manager and select **Driver new** in the context menu.
2. In the following dialog the control system offers a list of all available drivers.



3. Select the desired driver and give it a name:

- The driver name has to be unique, i.e. if one and the same driver is to be used several times in one project, a new name has to be given each time.
 - The driver name is part of the file name. Therefore it may only contain characters which are supported by the operating system. Invalid characters are replaced by an underscore (_).
 - **Attention:** This name cannot be changed later on.
4. Confirm the dialog with **OK**. In the following dialog the single configurations of the drivers are defined.

Only the respective required drivers need to be loaded for a project. Later loading of an additional driver is possible without problems.



Information

For new projects and for existing projects which are converted to version 6.21 or higher, the following drivers are created automatically:

- ▶ Internal
- ▶ MathDr32
- ▶ SysDrv.

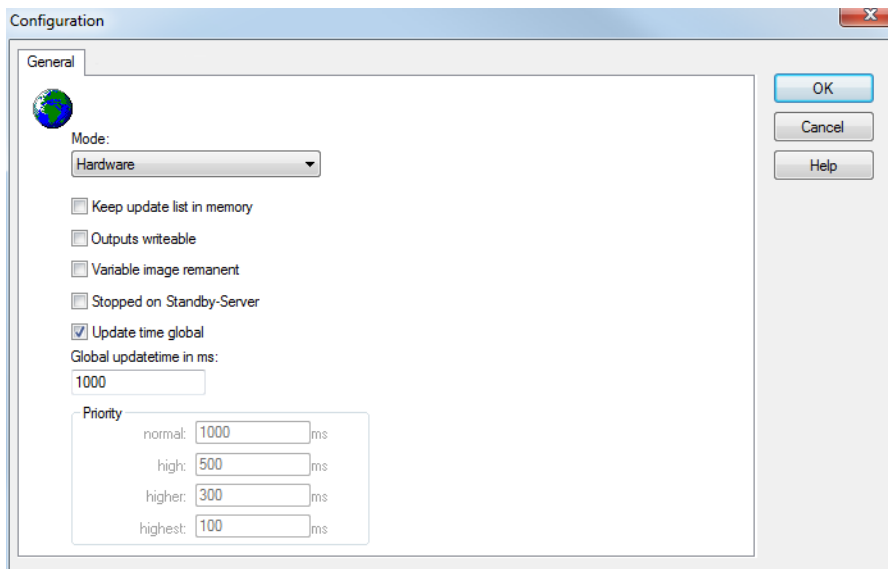
▶

5.2 Settings in the driver dialog

You can change the following settings of the driver:

5.2.1 General

The configuration dialog is opened when a driver is created. In order to be able to open the dialog later for editing, double click on the driver in the list or click on the **Configuration** property.



Parameters	Description
Mode	<p>Allows to switch between hardware mode and simulation mode</p> <ul style="list-style-type: none"> ▶ Hardware: <p>A connection to the control is established.</p> ▶ Simulation static <p>No communication between to the control is established, the values are simulated by the driver. In this modus the values remain constant or the variables keep the values which were set by zenon Logic. Each variable has its own memory area. E.g. two variables of the type marker with offset 79 can have different values in the Runtime and do not influence each other. Exception: The simulator driver.</p> ▶ Simulation - counting <p>No communication between to the control is established, the values are simulated by the driver. In this modus the driver increments the values within a value range automatically.</p> ▶ Simulation - programmed <p>N communication is established to the PLC. The values are calculated by a freely programmable simulation project. The simulation project is created with the help of the zenon Logic Workbench and runs in a zenon Logic Runtime which is integrated in the driver. For details see chapter Driver simulation (main.chm::/25206.htm).</p>
Keep update list in the memory	<p>Variables which were requested once are still requested from the control even if they are currently not needed. This has the advantage that e.g. multiple screen switches after the screen was opened for the first time are executed faster because the variables need not be requested again. The disadvantage is a higher load for the communication to the control.</p>
Output can be written	<p>Active: Outputs can be written.</p> <p>Inactive: Writing of outputs is prevented.</p> <p>Note: Not available for every driver.</p>
Variable image remanent	<p>This option saves and restores the current value, time stamp and the states of a data point.</p> <p>Fundamental requirement: The variable must have a valid value and time stamp.</p>

	<p>The variable image is saved in mode hardware if:</p> <ul style="list-style-type: none"> ▶ one of the states S_MERKER_1(0) up to S_MERKER8(7), REVISION(9), AUS(20) or ERSATZWERT(27) is active <p>The variable image is always saved if:</p> <ul style="list-style-type: none"> ▶ the variable is of the object type Driver variable ▶ the driver runs in simulation mode. (not programmed simulation) <p>The following states are not restored at the start of the Runtime:</p> <ul style="list-style-type: none"> ▶ SELECT(8) ▶ WR-ACK(40) ▶ WR-SUC(41) <p>The mode Simulation - programmed at the driver start is not a criterion in order to restore the remanent variable image.</p>
Stop on Standby Server	<p>Setting for redundancy at drivers which allow only on communication connection. For this the driver is stopped at the Standby Server and only started at the upgrade.</p> <p>Attention: If this option is active, the gapless archiving is no longer guaranteed.</p> <p>Active: Sets the driver at the not-process-leading Server automatically in a stop-like state. In contrast to stopping via driver command, the variable does not receive status switched off (statusverarbeitung.chm::/24150.htm) but an empty value. This prevents that at the upgrade to the Server irrelevant values are created in the AML, CEL and Historian.</p> <p>Note: Not available if the CE terminal serves as a data server. You can find further information in the zenon Operator manual in the CE terminal as a data server chapter.</p>
Global Update time	<p>Active: The set Global update time in ms is used for all variables in the project. The priority set at the variables is not used.</p> <p>Inactive: The set priorities are used for the individual variables.</p>
Priority	<p>The polling times for the individual priority classes are set here. All variables with the according priority are polled in the set time.</p> <p>The allocation to the variables takes place separately in the settings of the variable properties.</p> <p>The communication of the individual variables are graduated in respect of importance or necessary topicality using the priorities.</p>

	<p>Thus the communication load is distributed better.</p> <p>Attention: Priority classes are not supported by each driver For example, drivers that communicate spontaneously do not support it.</p>
--	--

CLOSE DIALOG

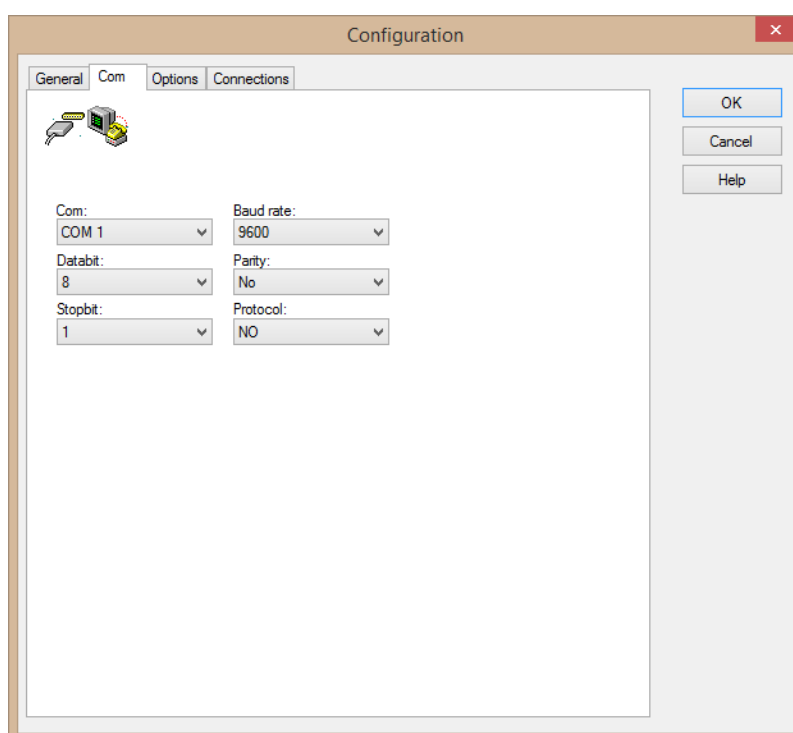
Parameters	Description
OK	Applies all changes in all tabs and closes the dialog.
Cancel	Discards all changes in all tabs and closes the dialog.
Help	Opens online help.

UPDATE TIME FOR CYCLICAL DRIVERS

The following applies for cyclical drivers:

For **Set value**, **Advising** of variables and **Requests**, a read cycle is immediately triggered for all drivers - regardless of the set update time. This ensures that the value is immediately available for visualization after writing. Update times can therefore be shorter than pre-set for cyclical drivers.

5.2.2 Com




Parameters	Description
Com	Selection Com port. Select from drop-down list: COM 1 to COM 64 Default: 1
Baud rate	Selection baud rate. Amend to PLC. Select from drop-down list: 110 to 256000 Default: 9600
Data bit	Number of data bits. Amend to PLC. Select from drop-down list: 5, 6, 7, 8 Default: 8
Stop bit	Selection stop bit. Amend to PLC. Select from drop-down list: <ul style="list-style-type: none"> ▶ 1 ▶ 1.5 ▶ 2 Default: 1
Parity	Selection of the parity. Amend to PLC. Selection from drop-down list: <ul style="list-style-type: none"> ▶ No ▶ Odd ▶ Even Default: No
Protocol	Selection protocol. Amend to PLC. Selection from drop-down list: <ul style="list-style-type: none"> ▶ No ▶ Xon/Xoff ▶ Rts/CTS ▶ DTR/DSR Default: No

CLOSE DIALOG**Parameters****OK****Cancel****Description**

Applies all changes in all tabs and closes the dialog.

Discards all changes in all tabs and closes the dialog.

Parameters	Description
Help	Opens online help.

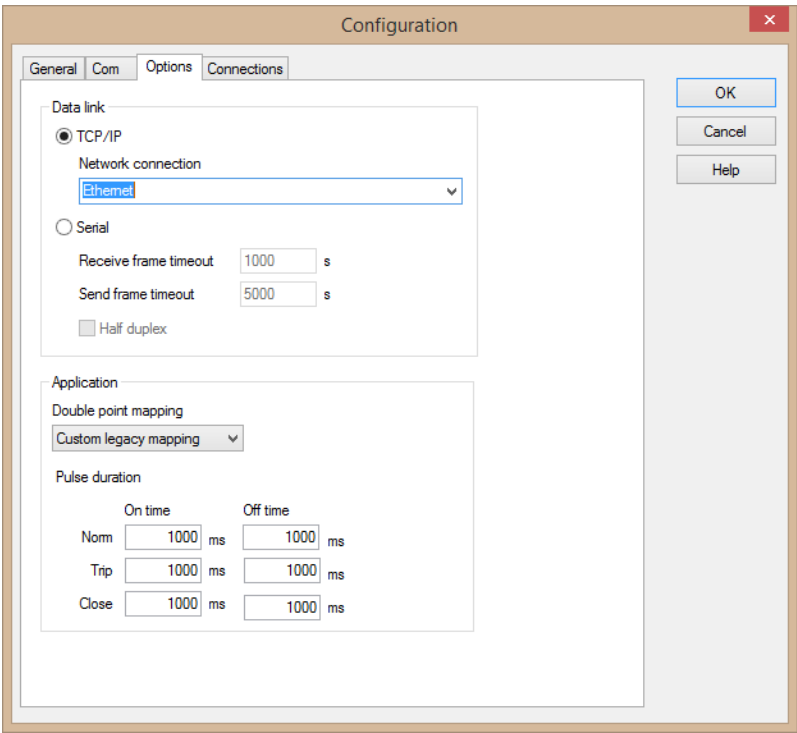
 **Info**

The exact settings depend on the used PLCs. Take the valid values from the manual of your PLC.

5.2.3 Options

You configure the communication parameters in this tab.

Note: This dialog is only available in English.



DATA LINK

Parameters	Description
TCP/IP	<ul style="list-style-type: none"> ▶ Active: Communication is via TCP/IP.
Network connection	<p>Selection of the network connection that is used for communication. Entry in the field or selection from a drop-down list from:</p> <ul style="list-style-type: none"> ▶ Name or ▶ Network connection or ▶ IP address. <p>The names of the network connections available on the local computer are shown in the drop-down list.</p> <p>For remote Runtime computers, enter the name of the network card or the IP address of the remote computer.</p>
Serial	<p>Active: Communication is via the serial interface.</p> <p>Default: Inactive</p>
Receive frame timeout [ms]	<p>Timeout in milliseconds for the receipt of a complete telegram. The starttime is the first character of the telegram.</p> <p>Default: 1000</p> <p>Note: Only active if serial has been activated.</p>
Send frame timeout [ms]	<p>Timeout in milliseconds for the sending of a telegram. If the serial output buffer is filled for longer than this time period, this is considered a transmission error</p> <p>Default: 5000</p> <p>Note: Only active if serial has been activated.</p>
Half duplex	<ul style="list-style-type: none"> ▶ Active: A telegram is sent again if a response has been received for the telegram that was last sent on the interface. <p>Default: Inactive</p> <p>Note: Only active if serial has been activated.</p>

APPLICATION

Parameters	Description
Double Point Mapping	<p>Selection of Double Point Mapping to an integer value. Select from drop-down list:</p> <ul style="list-style-type: none"> ▶ SCADA default mapping

	<ul style="list-style-type: none"> ▶ DNP3 binary mapping ▶ Custom legacy mapping <p>Default: SCADA default mapping</p>
--	--

PULSE DURATION

Setting of the pulse duration for **Norm**, **Trip** and **Close** for each connection. **Norm**, **Trip** and **Close** define which relay is switched

Parameters	Description
Norm	<p>Corresponds to NUL:</p> <p>Pulse duration norm:</p> <ul style="list-style-type: none"> ▶ On time: Period of time in milliseconds in which the Output is ON Default: 1000 ▶ Off time: Period of time in milliseconds in which the output is OFF Default: 1000
Trip	<p>Pulse duration trip:</p> <ul style="list-style-type: none"> ▶ On time: Period of time in milliseconds in which the output is ON Default: 1000 ▶ Off time: Period of time in milliseconds in which the output is OFF Default: 1000
Close	<p>Pulse duration close:</p> <ul style="list-style-type: none"> ▶ On time: Period of time in milliseconds in which the output is ON Default: 1000 ▶ Off time: Period of time in milliseconds in which the output is OFF Default: 1000

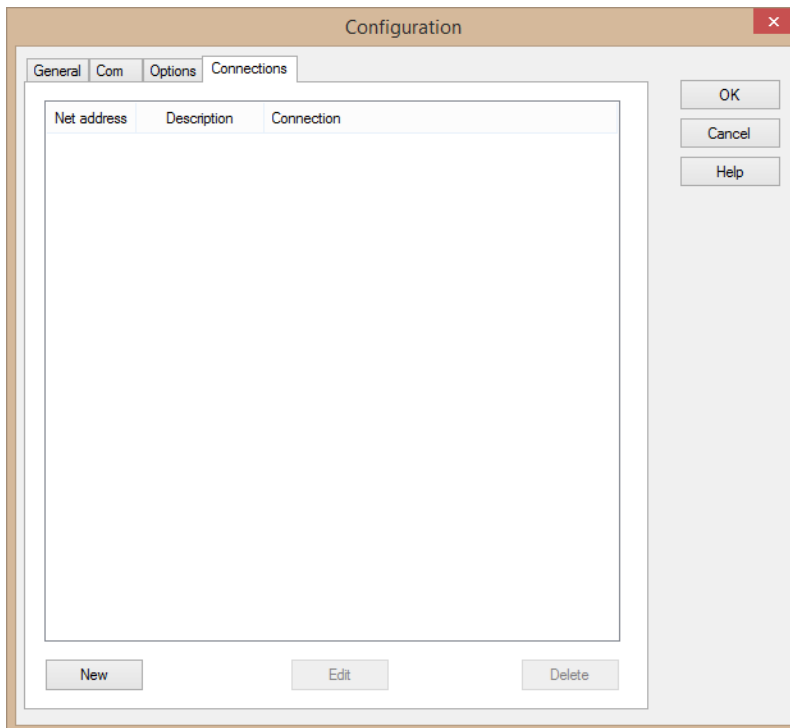
CLOSE DIALOG

Parameters	Description
OK	Applies all changes in all tabs and closes the dialog.
Cancel	Discards all changes in all tabs and closes the dialog.
Help	Opens online help.

5.2.4 Connections

You configure the connection in this tab.

Note: This dialog is only available in English.



Parameters	Description
List of connections	<p>Displays the configured connections.:</p> <ul style="list-style-type: none"> ▶ Net address Network address of the outstation. This address must be unique and can only be issued once. ▶ Description Description of the outstation. Corresponds to the Friendly name. ▶ Connection Connection port. <ul style="list-style-type: none"> • With a TCP connection TCP/IP address, TCP port • With a serial connection: Corresponds to the Com setting configured in the Com tab.
New	Opens the dialog for creating a new connection (on page 20).
Edit	Opens dialog for editing the selected connection.
Delete	Deletes the selected connection.

CLOSE DIALOG

Parameters	Description
OK	Applies all changes in all tabs and closes the dialog.
Cancel	Discards all changes in all tabs and closes the dialog.
Help	Opens online help.

CREATE NEW CONNECTION

1. Click on the **New** button.
The dialog for the configuration of the outstation (on page 20) is opened.
2. Enter the connection details.
3. Click on **OK**.
The dialog is closed. The connection that has just been created is listed in the connections list.

EDIT CONNECTION

1. Select the connection in the connection list.
2. Click on the **Edit** button.
The dialog for the configuration of an outstation (on page 20) is opened.

3. Change the connection parameters.
4. Click on **OK**.

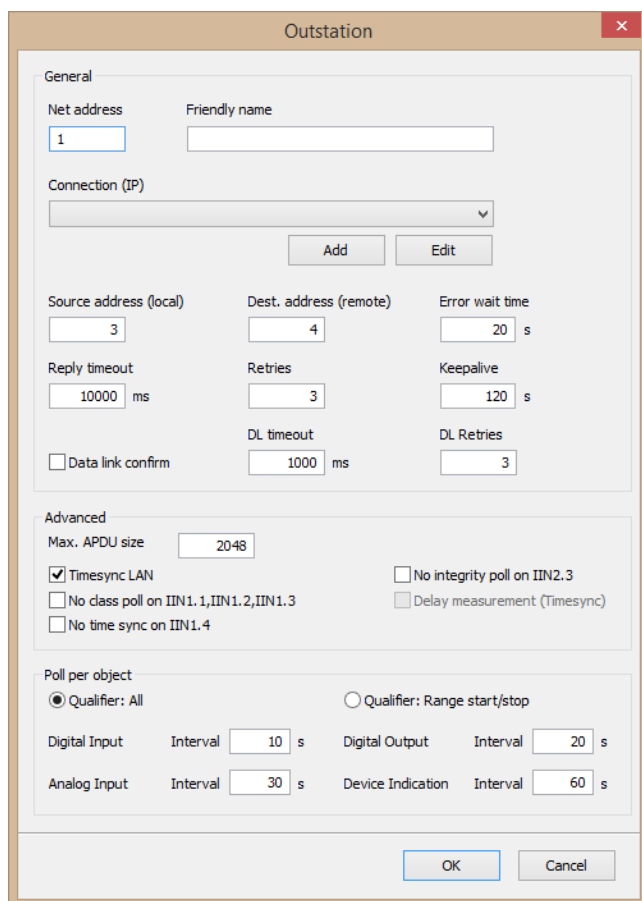
DELETE CONNECTION

1. Select the connection in the connection list.
2. Click on the **Delete** button.
3. The connection will be removed from the list

Outstation

You configure connections in this dialog. The dialog is opened if you click, in the **Connections** tab, on the **New** or **Edit** button.

Note: This dialog is only available in English.



Outstation

General

Net address: 1 Friendly name:

Connection (IP):

Source address (local): 3 Dest. address (remote): 4 Error wait time: 20 s

Reply timeout: 10000 ms Retries: 3 Keepalive: 120 s

☐ Data link confirm DL timeout: 1000 ms DL Retries: 3

Advanced

Max. APDU size: 2048

☒ Timesync LAN ☐ No integrity poll on IIN2.3

☐ No class poll on IIN1.1, IIN1.2, IIN1.3 ☐ Delay measurement (Timesync)

☐ No time sync on IIN1.4

Poll per object

☒ Qualifier: All ☐ Qualifier: Range start/stop

Digital Input Interval: 10 s Digital Output Interval: 20 s

Analog Input Interval: 30 s Device Indication Interval: 60 s

GENERAL

General settings.

Parameters	Description
Net address	<p>Net address of the connection.</p> <p>Value between 0 and 255.</p> <p>Corresponds to the Net address property for the variable.</p> <p>The network address must be unique.</p> <p>Default: Last network address issued + 1</p>
Friendly name	Name of connection. Freely selectable.
Connection (IP)	<p>Selection of the TCP channel via which communication with this Outstation takes place.</p> <p>With virtual Outstations, it is possible that several stations share one channel.</p> <p>Select from drop-down list. Administration of the connection with the Add and Edit buttons.</p> <p>(Not available for serial communication.)</p>
Add	Adds a new TCP link configuration (on page 26). The TCP links dialog is opened.
Edit	Opens the dialog to edit (on page 28) all existing TCP links.
Source address (local)	<p>The driver's own address (KDNP1 Master).</p> <p>Default: 3</p>
Dest. address (remote)	<p>Link address of the PLC (KDNP1 Slave).</p> <p>Default: 4</p>
Error wait time	<p>Waiting time in the event of a communication error. After an error, the given time is waited before another attempt at establishing a connection is made.</p> <p>Default: 20 s</p>
Reply timeout [ms]	<p>Time in milliseconds that is waited for a response from the controller. If this time has expired, a repetition is sent.</p> <p>If a fragment of a multi-part response is received. The interval starts over with each fragment received.</p> <p>Default: 10000 ms</p>
Retries	<p>Number of communication retries after timeouts.</p> <p>Default: 3</p>
Keepalive	<p>If data is received or sent beyond the given time, a LINK_STATUS_REQUEST telegram is sent.</p> <p>Default: 120 s</p>

Data link confirm	<p>If the Data link confirm property is activated, Confirmed data link layer telegrams are used in communication.</p> <p>This check is configured in the DL timeout and DL Retries properties.</p> <p>Default: <i>Inactive</i></p> <p>Note: This property should only be activated if Confirmed data link layer is also activated on the outstation.</p>
DL timeout	<p>Time in milliseconds that is waited for a data link response from the controller . If this time has expired, a repetition is sent.</p> <p>In contrast to the Reply timeout [ms] that is applicable for an application layer telegram, the DL timeout is applicable for a data link layer telegram.</p> <p>An Application Layer telegram can consist of several Data link layer telegrams.</p> <p>Default: 1000 ms</p> <p>Maximum time: 99999 ms</p> <p>Only active if Data link confirm is active.</p>
DL Retries	<p>Maximum number of repeats for a connection check after a connection error on the Data link layer.</p> <p>Default: 3</p> <p>Only active if Data link confirm is active.</p>

ADVANCED

Parameters	Description
Max. APDU size	<p>Maximum size of a KDNF Application Layer Fragment.</p> <ul style="list-style-type: none"> ▶ Minimum: 249 ▶ Maximum: 65535 <p>Default: 2048</p>
Timesync LAN	<ul style="list-style-type: none"> ▶ Active: Time synchronization via the LAN. <p>Can only be configured for a TCP connection.</p> <p>Default: <i>Activated</i></p> <p>If serial connection type (on page 15) is selected, the property in the driver dialog is automatically activated and grayed out.</p> <p>Note: Use variation 3 of the time object. This is not accepted by</p>

	all stations.
No class poll on IIN1.1, IIN1.2, IIN1.3	<p>Active: Driver ignores Internal Indication Flags IIN1.1, IIN1.2 and IIN1.3</p> <p>Compatibility setting: Ensure that polling for events is carried out regularly in order to avoid possible buffer overflows in the Outstation.</p>
No time sync on IIN1.4	<p>Active: The driver ignores the internal indication flag IIN1.4.</p> <p>Compatibility setting: Time synchronization required.</p>
No integrity poll on IIN2.3	<p>Active: Driver ignores the Internal Indication Flag IIN2.3 (Buffer Overflow). Compatibility setting.</p>
Delay Measurement (Timesync)	<p>If active, the driver carries out a Delay Measurement when time synchronizing.</p> <p>Active if:</p> <ul style="list-style-type: none"> ▶ Connection type Serial ▶ IP connection type (TCP/IP) and Timesync LAN not active. <p>Note: With a serial connection, Timesync_LAN is active by default and grayed out, Delay Measurement (Timesync) can however be selected.</p>

POLL PER OBJECT

In this area, you configure the query parameters to the outstation.

The following can be configured:

- ▶ Type of polling (query)
- ▶ Polled objects
- ▶ Time interval of the polling

Parameters	Description
Qualifier: All	<p>If this property is active, all available data points for the respective object type of the outstation are read.</p> <p>The interval time of the query can be configured for the respective property.</p> <p>Default: <code>activated</code></p>
Qualifier: Range start/stop	<p>If this property is active, only variables in the project can be polled for object types.</p> <p>This means: The driver determines the smallest and largest offset and uses this for the object poll.</p> <p>Default: <code>Not activated</code></p>
Binary Input	<p>Polling interval for Binary Input object-type variables.</p> <p>Default: <code>10 s</code></p> <p>Note: Value 0 deactivates the polling for this object type.</p>
Binary Output	<p>Polling interval for Binary Output object-type variables.</p> <p>Default: <code>20 s</code></p> <p>Note: Value 0 deactivates the polling for this object type.</p>
Analog Input	<p>Polling interval for Analog Input object-type variables.</p> <p>Default: <code>30 s</code></p> <p>Note: Value 0 deactivates the polling for this object type.</p>
Device Indication	<p>Polling interval for the Device Indication object type. This object type provides information on the status of the modules in the outstation:</p> <ul style="list-style-type: none"> ▶ 0 = ok ▶ 1 = fault <p>Default: <code>60 s</code></p> <p>Note: Value 0 deactivates the polling for this object type.</p>

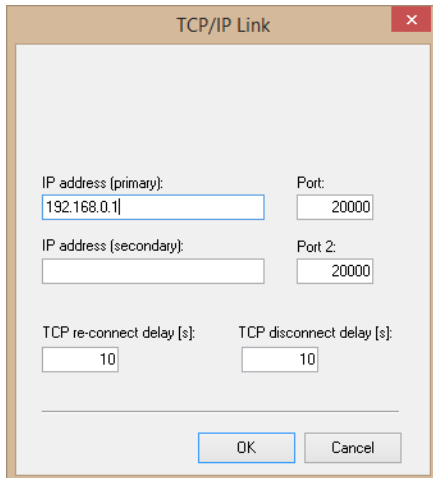
CLOSE DIALOG

Parameters	Description
OK	Applies settings and closes the dialog.
Cancel	Discards all changes and closes the dialog.

TCP/IP link configuration

Configuration of TCP links.

Note: This dialog is only available in English.



The screenshot shows a dialog box titled "TCP/IP Link" with a standard Windows-style title bar (minimize, maximize, close buttons). The dialog contains several input fields and two buttons at the bottom. The fields are arranged in a grid-like fashion. The "IP address (primary)" field contains the text "192.168.0.1". The "Port" field contains "20000". The "IP address (secondary)" field is empty. The "Port 2:" field contains "20000". The "TCP re-connect delay [s]:" field contains "10". The "TCP disconnect delay [s]:" field contains "10". At the bottom, there are two buttons: "OK" and "Cancel".

Field Label	Value
IP address (primary):	192.168.0.1
Port:	20000
IP address (secondary):	
Port 2:	20000
TCP re-connect delay [s]:	10
TCP disconnect delay [s]:	10

Buttons: OK, Cancel

Parameters	Description
IP address (primary)	Entry of the primary IP address.
Port	Primary TCP port of the PLC. Default: 20000
IP address (secondary)	Backup address for redundant connection to the PLC.
Port 2	Port of the backup address. Default: 20000
TCP re-connect delay [s]	Minimum time period in seconds that must be between two TCP connection attempts. Default: 10 s
TCP disconnect delay [s]	If no data is sent or received over the given time period, the TCP connection is disconnected automatically. Default: 10 s Note: In this case, the Polling intervals must be configured with a correspondingly low time period. If the time period set here = 0 the TCP connection is not separated.

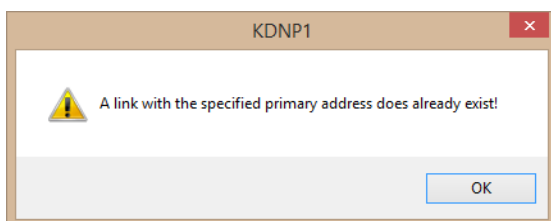
NAVIGATION

OK	Applies settings and closes the dialog.
Cancel	Discards all changes and closes the dialog.

ERROR DIALOG - ADDRESS ALREADY EXISTS

Only one outstation with a port number can be configured per IP address. If two outstations on the same IP address with different port numbers need to be addressed, please use one more driver.

If an IP address conflict occurs during the configuration of an outstation, this is visualized with a warning dialog.



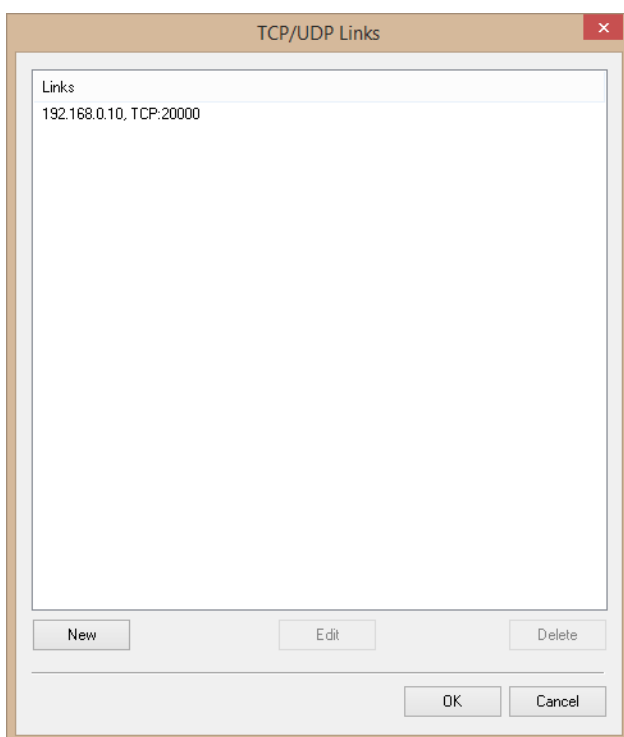
Click on the **Browse...** button to return to the configuration dialog.

Note: This dialog is only available in English.

TCP/UDP Links

Display and administration of all TCP links.

Note: This dialog is only available in English.



Parameters	Description
Links	Displays the configured TCP channels. The IP address and the TCP port is displayed: [IP address], TCP:[port number]
New	Opens the dialog to create a new (on page 26) TCP channel.
Edit	Opens the dialog to edit (on page 26) the selected TCP channel.
Delete	Deletes selected TCP link. The TCP link is deleted without requesting confirmation.

NAVIGATION

Parameters	Description
OK	Applies settings and closes the dialog.
Cancel	Discards all changes and closes the dialog.

CREATE A NEW TCP CHANNEL

1. Click on the **New** button.
The TCP/IP link configuration (on page 26) dialog is opened.
2. Enter the connection details.
3. Click on the **OK** button.
The dialog is closed. The configured TCP channel is displayed in the **Links** list.

EDIT A TCP CHANNEL

1. Select the desired Link in the list.
2. Click on the **Edit** button.
The TCP/IP link configuration (on page 26) dialog is opened.
3. Change the link parameter.
4. Close it with **OK**.
The dialog is closed. The configured TCP channel is displayed in the **Links** list.

DELETE A TCP CHANNEL

1. Select the desired Link in the list.
2. Click on the **Delete** button.

3. The connection will be deleted from the list without requesting confirmation.

6. Creating variables

This is how you can create variables in the zenon Editor:

6.1 Creating variables in the Editor

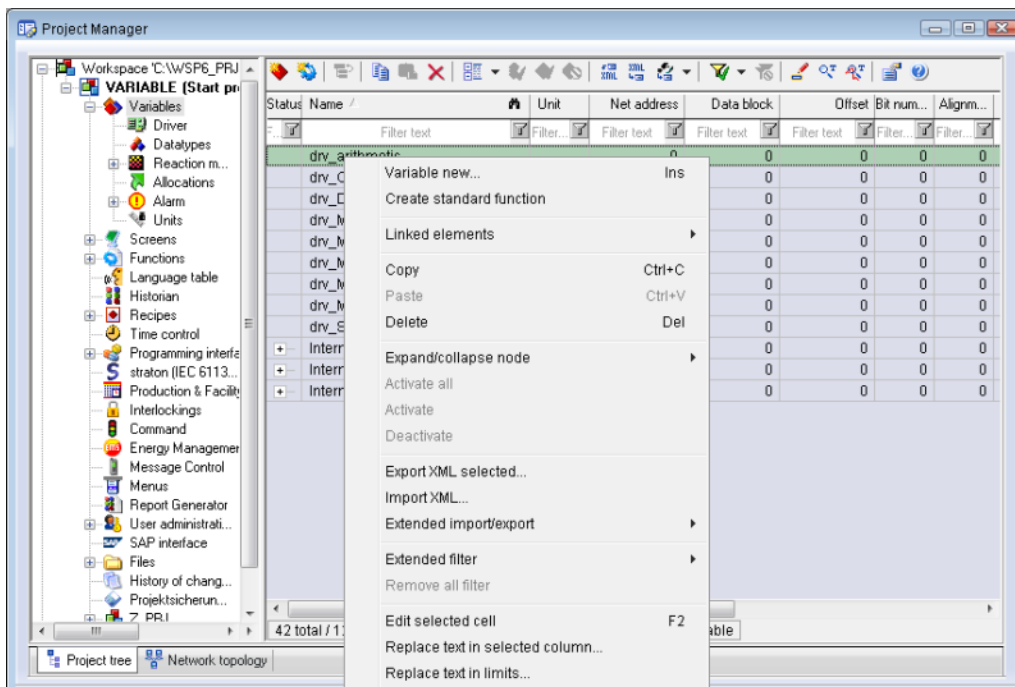
Variables can be created:

- ▶ as simple variables
- ▶ in arrays (main.chm::/15262.htm)
- ▶ as structure variables (main.chm::/15278.htm)

VARIABLE DIALOG

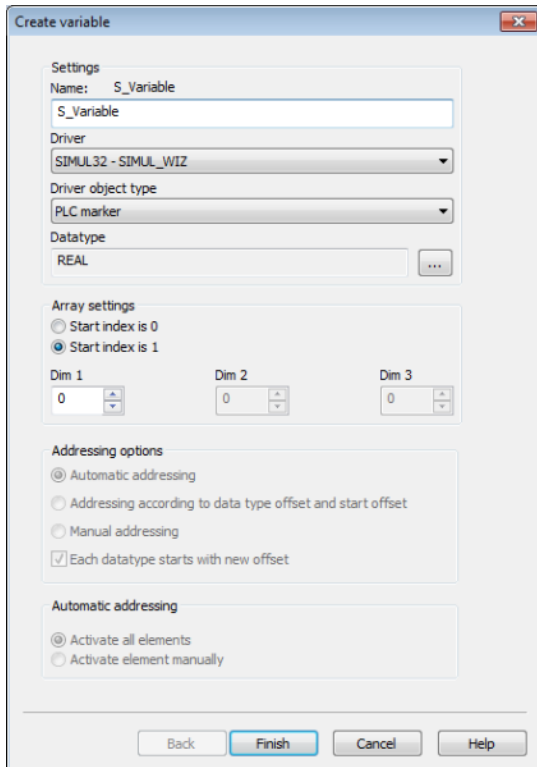
To create a new variable, regardless of which type:

1. Select the **New variable** command in the **Variables** node in the context menu



2. The dialog for configuring variables is opened

3. configure the variable
4. The settings that are possible depends on the type of variables



The screenshot shows the 'Create variable' dialog box with the following settings:

- Settings**
 - Name: S_Variable
 - Driver: SIMUL32 - SIMUL_WIZ
 - Driver object type: PLC marker
 - Datatype: REAL
- Array settings**
 - ☐ Start index is 0
 - ☒ Start index is 1
 - Dim 1: 0
 - Dim 2: 0
 - Dim 3: 0
- Addressing options**
 - ☒ Automatic addressing
 - ☐ Addressing according to data type offset and start offset
 - ☐ Manual addressing
 - ☒ Each datatype starts with new offset
- Automatic addressing**
 - ☒ Activate all elements
 - ☐ Activate element manually

Buttons at the bottom: Back, Finish, Cancel, Help.

Property	Description
Name	Distinct name of the variable. If a variable with the same name already exists in the project, no additional variable can be created with this name. Maximum length: 128 character Attention: The characters # and @ are not permitted in variable names. If non-permitted characters are used, creation of variables cannot be completed and the Finish button remains inactive. Note: For some drivers, the addressing is possible over the property Symbolic address , as well.
Drivers	Select the desired driver from the drop-down list. Note: If no driver has been opened in the project, the driver for internal variables (Intern.exe (Main.chm::/Intern.chm::/Intern.htm)) is automatically loaded.
Driver object type (cti.chm::/28685.htm)	Select the appropriate driver object type from the drop-down list.
Data type	Select the desired data type. Click on the ... button to open the selection dialog.
Array settings	Expanded settings for array variables. You can find details in the Arrays chapter.
Addressing options	Expanded settings for arrays and structure variables. You can find details in the respective section.
Automatic element activation	Expanded settings for arrays and structure variables. You can find details in the respective section.

SYMBOLIC ADDRESS

The **Symbolic address** property can be used for addressing as an alternative to the **Name** or **Identification** of the variables. Selection is made in the driver dialog; configuration is carried out in the variable property. When importing variables of supported drivers, the property is entered automatically.

Maximum length: 1024 characters.

INHERITANCE FROM DATA TYPE

Measuring range, **Signal range** and **Set value** are always:

- ▶ derived from the datatype
- ▶ Automatically adapted if the data type is changed

Note for **signal range**: If a change is made to a data type that does not support the set **signal range**, the **signal range** is amended automatically. For example, for a change from **INT** to **SINT**, the **signal range** is changed to 127. The amendment is also carried out if the **signal range** was not inherited from the data type. In this case, the **measuring range** must be adapted manually.

6.2 Addressing

The offset determines the KDNP Point Nummer. In combination with the driver object type, the KDNP object group is determined.

Group/Property	Description
General	General variable properties.
Name	Freely definable name. Attention: For every zenon project the name must be unambiguous.
Identification	Freely-assignable identification, e.g. for descriptions and comments.
Addressing	Properties of the variable addressing.
Net address	Bus address or Net address of the variable. This address refers to the station address in the connection configuration of the driver. This defines the PLC, on which the variable resides. Attention: The DNP address of the Outstation is configured separately during the connection.
Data block	not used for this driver
Offset	For variables of all driver object types: KDNP Point in the Outstation. To see always in combination with the driver object type.
Alignment	not used for this driver
Bit number	not used for this driver
String length	Only available for String variables: Maximum number of characters that the variable can take.
Driver connection	Driver-specific properties.
Driver Object Type	Depending on the employed driver, an object type is selected during the creation of the variable; the type can be changed here later.
Data Type	Data type of the variable. Is selected during the creation of the variable; the type can be changed here later. Attention: If you change the data type later, all other properties of the variable must be checked and adjusted, if necessary.

6.3 Driver objects and datatypes

Driver objects are areas available in the PLC, such as markers, data blocks etc. Here you can find out which driver objects are provided by the driver and which IEC data types can be assigned to the respective driver objects.

6.3.1 Driver objects

The following object types are available in this driver:

Driver object type	Channel type	Read	Write	Supported data types	Comment
Analog input	68	X	--	UDINT, DINT, REAL, LREAL, UINT, INT	
Analog output	69	X	X	UDINT, DINT, REAL, LREAL, UINT, INT	
Binary Input	64	X	--	BOOL	
Binary Output	65	X	X	BOOL, USINT	
Command	11	X	X	STRING	
Device indication	79	X	--	USINT	
Driver variable	35	X	X	BOOL, SINT, USINT, INT, UINT, DINT, UDINT, REAL, STRING	<p>Variables for the static analysis of the communication; is transferred between driver and Runtime (not to the PLC).</p> <p>Note: The addressing and the behavior is the same for most zenon drivers.</p> <p>Find out more in the chapter about the Driver variables (on page 46)</p>

6.3.2 Mapping of the data types

DRIVER OBJECT TYPES AND SUPPORTED IEC DATA TYPES FOR VARIABLES IN ZENON

Driver object types	Channel type	Supported data types (DataType)	Read	Write	Comment
Binary Input Static object group 1 Event object group 2	64	BOOL	X	--	
Binary Output Static object group 10 Event object Group 11 CROB object group 12	65	BOOL	--	X	Value 1: LATCH_ON Value 0: LATCH_OFF
Binary Output Static object group 10 Event object group 11 CROB object group 12	65	USINT	--	X	Value 1: PULSE_ON Value 2: PULSE_OFF Value 3: LATCH_ON Value 4: LATCH_OFF Value 65: CLOSE Value 129: TRIP Value 1 for PULSE_ON, LATCH_ON and CLOSE, value 0 for PULSE_OFF, LATCH_OFF and TRIP
Analog Input Static object group 30 Event object group 32	68	INT, DINT, REAL, LREAL	X	--	
Analog Output Static object group 40 Event object group 42 Analog Output Block object	69	INT, DINT, REAL, LREAL	--	X	written value is mirrored as a response after successful writing

group 41					
Command	11	STRING	X	X	Initiates an explicit command For example, explicit reading, ...
Device indication	79	USINT	x	--	Provides information on the status of the modules in the outstation: ► 0 = ok ► 1 = fault

BINARY OUTPUTS

The writing of **Binary Outputs** is always carried out by means of a **CROB** (group 12), with a choice of **Direct Operate**, **Select Before Operate**, or **Direct Operate No Ack**. In doing so, the setting for the variable-specific property **Command Mode** (accessible via XML export/import and VBA **COMMAND_MODE**) is used. Direct writing to group 10 is not supported by the driver.

After successful writing, the variable receives the value 0 for **Binary Outputs** with **PULSE_OFF**, **LATCH_OFF** and **TRIP**, and value 1 for **PULSE_ON**, **LATCH_ON** and **CLOSE**. The value is also updated if a **Binary Output Status** (group 10) or **Binary Output Status Event** (group 11) object is received.

Command processing with **Binary Output** variables:

- Only **dual command** or **switching commands** can be used.
- The property **Select Before Operate** must be deactivated for the variable; the KDNP1 driver uses **Auto-SBO** if necessary.

Use of the **Qualifier of Command** is recommended for the Command Processing in order to determine the type:

Type	QoC
PULSE OFF/PULSE ON	1
LATCH OFF/LATCH ON	0
TRIP/CLOSE	2

The **Qualifier of Command** option can be used with both **Binary Output** variables of the **BOOL** data type as well as the **USINT** data type.

ANALOG OUTPUTS

The writing of Analog Outputs is always with a group 41, with the choice of `Direct Operate`, `Select Before Operate` or `Direct Operate No Ack`. In doing so, the setting for the variable-specific property **Command Mode** (accessible via XML export/import and VBA `COMMAND_MODE`) is used.

After successful writing, the variable first receives the value written in Runtime. The value is also updated if an `Analog Output Status` (group 40) or `Analog Output Status` (group 42) object is received.

Use the `write set value` action text for Analog Outputs. According to the **Command Mode** option for the variable, for Analog Outputs either a `direct operate` or a `select before operate` is carried out. The **Select Before Operate** option must be deactivated for the variable. The **Qualifier of Command** option is not available for the `Write set value` action and has no influence with Analog Outputs.

For two-stage command processing, the command is only sent to the outstation at the second stage of the command, including **Select** with `Auto-SBO`.

EXPLICIT COMMAND

`Command` driver object type variables are used to instigate a certain action once. The action is initiated by the writing of a string value to the command variable. The result of the action is assigned to the variables.

- ▶ **Syntax:**
A command consists of the command name and parameters. The command name and the individual parameters are separated by spaces.
Example: `READ 1 1` Read all binary inputs for network address 1 (Object Group 1, Variation 0 (default variation))

Attention: It is possible, with the compilation of parameters for an explicit command, that there are combinations that do not correspond to a valid command in the meaning of the IEEE 1815 standard. Commands are sent to the outstation without further checking and can, under certain circumstances, trigger incorrect actions here.

EXPLICIT COMMAND

`Command` driver object type variables are used to instigate a certain action once. The action is initiated by the writing of a string value to the command variable. The result of the action is assigned to the variables.

- ▶ **Syntax:** A command consists of the command name and parameters. The command name and the individual parameters are separated by spaces.
Example: `READ_VAR 1 0 254` Read all device attributes for network address 1 (Object Group 0, Variation 254)

Example: READ 1 1 Read all binary inputs for network address 1 (Object Group 1, Variation 0 (default variation))

Attention: It is possible, with the compilation of parameters for an explicit command, that there are combinations that do not correspond to a valid command in the meaning of the IEEE 1815 standard. Commands are sent to the outstation without further checking and can, under certain circumstances, trigger incorrect actions here.

EXPLICIT READ, VARIATION 0

Command: READ

Read all points of a group: (Qualifier Code 06 - you can use this number of parameters for static Object Groups and Event Object Groups)

- ▶ Parameter 1: **Net address**
- ▶ Parameter 2: Group

Read n-points of a group: (Qualifier Code 07, 08, 09 - you can use this number of parameters for the Event Object Groups. The use of a number greater than 65535 (Qualifier Code 09) is not recommended))

- ▶ Parameter 1: **Net address**
- ▶ Parameter 2: Group
- ▶ Parameter 3: Number of the points to be read

Read certain points of a group: (Qualifier Code 00, 01 - you can use this number of parameters for static Object Groups, for example: Group 1 - Binary Input).

- ▶ Parameter 1: **Net address**
- ▶ Parameter 2: Group
- ▶ Parameter 3: Start index
- ▶ Parameter 4: Stop index

EXPLICIT READ

Command: READ_VAR

Read all points of a group: (Qualifier Code 06 - you can use this number of parameters for static Object Groups and Event Object Groups)

- ▶ Parameter 1: **Net address**
- ▶ Parameter 2: Group
- ▶ Parameter 3: Variation

Read n-points of a group: (Qualifier Code 07, 08, 09 - you can use this number of parameters for Event Object Groups. The use of a number greater than 65535 (Qualifier Code 09) is not recommended))

- ▶ Parameter 1: **Net address**
- ▶ Parameter 2: Group
- ▶ Parameter 3: Variation
- ▶ Parameter 4: Number of the points to be read

Read certain points of a group: (Qualifier Code 00, 01 - you can use this number of parameters for static Object Groups, for example: Group 1 - Binary Input).

- ▶ Parameter 1: **Net address**
- ▶ Parameter 2: Group
- ▶ Parameter 3: Variation
- ▶ Parameter 4: Start index
- ▶ Parameter 5: Stop index

ACTIVATE/DEACTIVATE OBJECT POLL

Command: CYCLIC_POLL

- ▶ Parameter 1: **Net address**
The command is rejected if an invalid network address is entered.
- ▶ Parameter 2: Polling:
 - 0 - deactivated
 - 1 - activated

COMMAND STATUS CODES

Status code	Description
16	Syntax error.
17	Unknown command.
18	Net address Not existing.
3	Command has been sent.
4	Command completed successfully.
5	Command could not be processed.
6	Command received a negative response.
7	Error when executing the command (Timeout.
8	Execution of the command was interrupted.

6.4 Creating variables by importing

Variables can also be imported by importing them. The XML and DBF import is available for every driver.



Information

You can find details on the import and export of variables in the Import-Export (main.chm::/13028.htm) manual in the Variables (main.chm::/13045.htm) section.

6.4.1 XML import

For the import/export of variables the following is true:

- ▶ The import/export must not be started from the global project.
- ▶ The start takes place via:
 - Context menu of variables or data typ in the project tree
 - or context menu of a variable or a data type
 - or symbol in the symbol bar variables



Attention

When importing/overwriting an existing data type, all variables based on the existing data type are changed.

Example:

There is a data type XYZ derived from the type `INT` with variables based on this data type. The XML file to be imported also contains a data type with the name XYZ but derived from type `STRING`. If this data type is imported, the existing data type is overwritten and the type of all variables based on it is adjusted. I.e. the variables are now no longer `INT` variables, but `STRING` variables.

6.4.2 DBF Import/Export

Data can be exported to and imported from dBase.



Information

Import and Export via CSV or dBase supported; no driver specific variable settings, such as formulas. Use export/import via XML for this.

IMPORT DBF FILE

To start the import:

1. right-click on the variable list
2. in the drop-down list of **Extended export/import...** select the **Import dBase** command
3. follow the import assistant

The format of the file is described in the chapter File structure.



Information

Note:

- ▶ Driver object type and data type must be amended to the target driver in the DBF file in order for variables to be imported.
- ▶ dBase does not support structures or arrays (complex variables) at import.

EXPORT DBF FILE

To start the export:

1. right-click on the variable list
2. in the drop-down list of **Extended export/import...** select the **Export dBase...** command
3. follow the export assistant



Attention

DBF files:

- ▶ must correspond to the 8.3 DOS format for filenames (8 alphanumeric characters for name, 3 character suffix, no spaces)
- ▶ must not have dots (.) in the path name.
e.g. the path C:\users\John.Smith\test.dbf is invalid.
Valid: C:\users\JohnSmith\test.dbf
- ▶ must be stored close to the root directory in order to fulfill the limit for file name length including path: maximum 255 characters

The format of the file is described in the chapter File structure.



Information

dBase does not support structures or arrays (complex variables) at export.

File structure of the dBase export file

The dBaseIV file must have the following structure and contents for variable import and export:



Attention

dBase does not support structures or arrays (complex variables) at export.

DBF files must:

- ▶ conform with their name to the 8.3 DOS format (8 alphanumeric characters for name, 3 characters for extension, no space)
- ▶ Be stored close to the root directory (Root)

STRUCTURE

Identification	Type	Field size	Comment
KANALNAME	Char	128	Variable name. The length can be limited using the MAX_LAENGE entry in project.ini .
KANAL_R	C	128	The original name of a variable that is to be replaced by the new name entered under "VARIABLENNAME" (field/column must be entered manually). The length can be limited using the MAX_LAENGE entry in project.ini .
KANAL_D	Log	1	The variable is deleted with the 1 entry (field/column has to be created by hand).
TAGNR	C	128	Identification. The length can be limited using the MAX_LAENGE entry in project.ini .
ENHEIT	C	11	Technical unit
DATENART	C	3	Data type (e.g. bit, byte, word, ...) corresponds to the data type.
KANALTYP	C	3	Memory area in the PLC (e.g. marker area, data area, ...) corresponds to the driver object type.
HWKANAL	Num	3	Bus address
BAUSTEIN	N	3	Datablock address (only for variables from the data area of the PLC)
ADRESSE	N	5	Offset
BITADR	N	2	For bit variables: bit address For byte variables: 0=lower, 8=higher byte For string variables: Length of string (max. 63 characters)
ARRAYSIZE	N	16	Number of variables in the array for index variables ATTENTION: Only the first variable is fully available. All others are only available for VBA or the Recipegroup Manager

LES_SCHR	L	1	Write-Read-Authorization 0: Not allowed to set value. 1: Allowed to set value.
MIT_ZEIT	L	1	time stamp in zenon (only if supported by the driver)
OBJEKT	N	2	Driver-specific ID number of the primitive object comprises TREIBER-OBJEKTYP and DATENTYP
SIGMIN	Float	16	Non-linearized signal - minimum (signal resolution)
SIGMAX	F	16	Non-linearized signal - maximum (signal resolution)
ANZMIN	F	16	Technical value - minimum (measuring range)
ANZMAX	F	16	Technical value - maximum (measuring range)
ANZKOMMA	N	1	Number of decimal places for the display of the values (measuring range)
UPDATERATE	F	19	Update rate for mathematics variables (in sec, one decimal possible) not used for all other variables
MENTIEFE	N	7	Only for compatibility reasons
HDRATE	F	19	HD update rate for historical values (in sec, one decimal possible)
HDTIEFE	N	7	HD entry depth for historical values (number)
NACHSORT	L	1	HD data as postsorted values
DRRATE	F	19	Updating to the output (for zenon DDE server, in [s], one decimal possible)
HYST_PLUS	F	16	Positive hysteresis, from measuring range
HYST_MINUS	F	16	Negative hysteresis, from measuring range
PRIOR	N	16	Priority of the variable
REAMATRIZE	C	32	Allocated reaction matrix
ERSATZWERT	F	16	Substitute value, from measuring range
SOLLMIN	F	16	Minimum for set value actions, from measuring range
SOLLMAX	F	16	Maximum for set value actions, from measuring range
VOMSTANDBY	L	1	Get value from standby server; the value of the variable is not requested from the server but from the Standby Server in redundant networks
RESOURCE	C	128	Resources label. Free string for export and display in lists. The length can be limited using the MAX_LAENGE entry in project.ini .
ADJWVBA	L	1	Non-linear value adaption: 0: Non-linear value adaption is used 1: Non-linear value adaption is not used

ADJZENON	C	128	Linked VBA macro for reading the variable value for non-linear value adjustment.
ADJWVBA	C	128	ed VBA macro for writing the variable value for non-linear value adjustment.
ZWREMA	N	16	Linked counter REMA.
MAXGRAD	N	16	Gradient overflow for counter REMA.



Attention

When importing, the driver object type and data type must be amended to the target driver in the DBF file in order for variables to be imported.

LIMIT VALUE DEFINITION

Limit definition for limit values 1 to 4, or status 1 to 4:

Identification	Type	Field size	Comment
AKTIV1	L	1	Limit value active (per limit value available)
GRENZWERT1	F	20	technical value or ID number of a linked variable for a dynamic limit value (see VARIABLEx) (if VARIABLEx is 1 and here it is -1, the existing variable linkage is not overwritten)
SCHWWERT1	F	16	Threshold value for limit value
HYSTERESE1	F	14	Is not used
BLINKEN1	L	1	Set blink attribute
BTB1	L	1	Logging in CEL
ALARM1	L	1	Alarm
DRUCKEN1	L	1	Printer output (for CEL or Alarm)
QUITTIER1	L	1	Must be acknowledged
LOESCHE1	L	1	Must be deleted
VARIABLE1	L	1	Dyn. limit value linking the limit is defined by an absolute value (see field GRENZWERTx).
FUNC1	L	1	Functions linking
ASK_FUNC1	L	1	Execution via Alarm Message List
FUNC_NR1	N	10	ID number of the linked function (if "-1" is entered here, the existing function is not overwritten during import)
A_GRUPPE1	N	10	Alarm/Event Group
A_KLASSE1	N	10	Alarm/Event Class
MIN_MAX1	C	3	Minimum, Maximum
FARBE1	N	10	Color as Windows coding
GRENZTXT1	C	66	Limit value text
A_DELAY1	N	10	Time delay
INVISIBLE1	L	1	Invisible

Expressions in the column "Comment" refer to the expressions used in the dialog boxes for the definition of variables. For more information, see chapter Variable definition.

6.5 Driver variables

The driver kit implements a number of driver variables. These are divided into:

- ▶ Information
- ▶ Configuration
- ▶ Statistics and
- ▶ Error message

The definitions of the variables implemented in the driver kit are available in the import file **drvvar.dbf** (on the installation medium in the \Predefined\Variables folder) and can be imported from there.

Note: Variable names must be unique in zenon. If driver variables are to be imported from **drvvar.dbf** again, the variables that were imported beforehand must be renamed.



Information

Not every driver supports all driver variants.

For example:

- ▶ Variables for modem information are only supported by modem-compatible drivers
- ▶ Driver variables for the polling cycle only for pure polling drivers
- ▶ Connection-related information such as ErrorMSG only for drivers that only edit one connection at a time

INFORMATION

Name from import	Type	Offset	Description
MainVersion	UINT	0	Main version number of the driver.
SubVersion	UINT	1	Sub version number of the driver.
BuildVersion	UINT	29	Build version number of the driver.
RTMajor	UINT	49	zenon main version number
RTMinor	UINT	50	zenon sub version number
RTSp	UINT	51	zenon Service Pack number
RTBuild	UINT	52	zenon build number
LineStateIdle	BOOL	24.0	TRUE, if the modem connection is idle
LineStateOffering	BOOL	24.1	TRUE, if a call is received
LineStateAccepted	BOOL	24.2	The call is accepted
LineStateDialtone	BOOL	24.3	Dialtone recognized
LineStateDialing	BOOL	24.4	Dialing active
LineStateRingBack	BOOL	24.5	While establishing the connection
LineStateBusy	BOOL	24.6	Target station is busy

LineStateSpecialInfo	BOOL	24.7	Special status information received
LineStateConnected	BOOL	24.8	Connection established
LineStateProceeding	BOOL	24.9	Dialing completed
LineStateOnHold	BOOL	24.10	Connection in hold
LineStateConferenced	BOOL	24.11	Connection in conference mode.
LineStateOnHoldPendConf	BOOL	24.12	Connection in hold for conference
LineStateOnHoldPendTransfer	BOOL	24.13	Connection in hold for transfer
LineStateDisconnected	BOOL	24.14	Connection terminated.
LineStateUnknow	BOOL	24.15	Connection status unknown
ModemStatus	UDINT	24	Current modem status
TreiberStop	BOOL	28	Driver stopped For <code>driver stop</code> , the variable has the value <code>TRUE</code> and an OFF bit. After the driver has started, the variable has the value <code>FALSE</code> and no OFF bit.
SimulRTState	UDINT	60	Informs the status of Runtime for driver simulation.

CONFIGURATION

Name from import	Type	Offset	Description
ReconnectInRead	BOOL	27	If <code>TRUE</code> , the modem is automatically reconnected for reading
ApplyCom	BOOL	36	Apply changes in the settings of the serial interface. Writing to this variable immediately results in the method <code>SrvDrvVarApplyCom</code> being called (which currently has no further function).
ApplyModem	BOOL	37	Apply changes in the settings of the modem. Writing this variable immediately calls the method <code>SrvDrvVarApplyModem</code> . This closes the current connection and opens a new one according to the settings PhoneNumberSet and ModemHwAdrSet .

PhoneNumberSet	STRING	38	Telephone number, that should be used
ModemHwAdrSet	DINT	39	Hardware address for the telephone number
GlobalUpdate	UDINT	3	Update time in milliseconds (ms).
BGlobalUpdaten	BOOL	4	TRUE, if update time is global
TreiberSimul	BOOL	5	TRUE, if driver in sin simulation mode
TreiberProzab	BOOL	6	TRUE, if the variables update list should be kept in the memory
ModemActive	BOOL	7	TRUE, if the modem is active for the driver
Device	STRING	8	Name of the serial interface or name of the modem
ComPort	UINT	9	Number of the serial interface.
Baudrate	UDINT	10	Baud rate of the serial interface.
Parity	SINT	11	Parity of the serial interface
ByteSize	USINT	14	Number of bits per character of the serial interface Value = 0 if the driver cannot establish any serial connection.
StopBit	USINT	13	Number of stop bits of the serial interface.
Autoconnect	BOOL	16	TRUE, if the modem connection should be established automatically for reading/writing
PhoneNumber	STRING	17	Current telephone number
ModemHwAdr	DINT	21	Hardware address of current telephone number
RxIdleTime	UINT	18	Modem is disconnected, if no data transfer occurs for this time in seconds (s)

WriteTimeout	UDINT	19	Maximum write duration for a modem connection in milliseconds (ms).
RingCountSet	UDINT	20	Number of ringing tones before a call is accepted
ReCallIdleTime	UINT	53	Waiting time between calls in seconds (s).
ConnectTimeout	UINT	54	Time in seconds (s) to establish a connection.

STATISTICS

Name from import	Type	Offset	Description
MaxWriteTime	UDINT	31	The longest time in milliseconds (ms) that is required for writing.
MinWriteTime	UDINT	32	The shortest time in milliseconds (ms) that is required for writing.
MaxBlkReadTime	UDINT	40	Longest time in milliseconds (ms) that is required to read a data block.
MinBlkReadTime	UDINT	41	Shortest time in milliseconds (ms) that is required to read a data block.
WriteErrorCount	UDINT	33	Number of writing errors
ReadSucceedCount	UDINT	35	Number of successful reading attempts

MaxCycleTime	UDINT	22	Longest time in milliseconds (ms) required to read all requested data.
MinCycleTime	UDINT	23	Shortest time in milliseconds (ms) required to read all requested data.
WriteCount	UDINT	26	Number of writing attempts
ReadErrorCount	UDINT	34	Number of reading errors
MaxUpdateTimeNormal	UDINT	56	Time since the last update of the priority group Normal in milliseconds (ms).
MaxUpdateTimeHigher	UDINT	57	Time since the last update of the priority group Higher in milliseconds (ms).
MaxUpdateTimeHigh	UDINT	58	Time since the last update of the priority group High in milliseconds (ms).
MaxUpdateTimeHighest	UDINT	59	Time since the last update of the priority group Highest in milliseconds (ms).
PokeFinish	BOOL	55	Goes to 1 for a query, if all current pokes were executed

ERROR MESSAGE

Name from import	Type	Offset	Description
ErrorTimeDW	UDINT	2	Time (in seconds since 1.1.1970), when the last error occurred.
ErrorTimeS	STRING	2	Time (in seconds since 1.1.1970), when the last error occurred.
RdErrPrimObj	UDINT	42	Number of the PrimObject, when the last reading error occurred.
RdErrStationsName	STRING	43	Name of the station, when the last reading error occurred.
RdErrBlockCount	UINT	44	Number of blocks to read when the last reading error occurred.

RdErrHwAdresse	DINT	45	Hardware address when the last reading error occurred.
RdErrDatablockNo	UDINT	46	Block number when the last reading error occurred.
RdErrMarkerNo	UDINT	47	Marker number when the last reading error occurred.
RdErrSize	UDINT	48	Block size when the last reading error occurred.
DrvError	USINT	25	Error message as number
DrvErrorMsg	STRING	30	Error message as text
ErrorFile	STRING	15	Name of error log file

7. Driver-specific functions

The driver supports the following functions:

Function	Description
Blockwrite	Not supported.
Redundancy	If the Outstation supports several Masters.
RDA	Not supported.
Real time stamping	If the Variation is configured accordingly in the Outstation.
Browse	Not supported.
	Polling for objects, in configurable cycles or controlled manually.
Spontaneous	Not supported.
Number of PLCs	One driver can connect to any number of Outstations. For serial communications, several Outstations can be configured for a serial interface (bus). Mixed operation serial/TCP is also possible with a driver.

ERROR FILE

The driver supports central logging on the diagnosis server (on page 57).

DRIVER VARIABLES

Communication monitoring is possible using the **INVALID** status bit.

KDNP1 OBJECT FLAGS MAPPING

The DNP object flags **ONLINE** and **COMM_LOST** are evaluated by the KDNP1 driver. With **ONLINE** = **false** or **COMM_LOST** = **true**, the **INVALID** bit for the variable is set in Runtime.

Note that the variation configured in the outstation determines whether objects are sent with or without flags for an object group.

In addition, the following flags/IIN bits listed are transferred to the variable status bits:

KDNP1 Flag / IIN	Status bit
Restart	NT Bit
Remote forced	SB Bit
Local forced	SB Bit
Rollover	OV Bit
IIN.1.4Need time	Time-inval Bit

TIME SYNCHRONIZATION

The driver supports time synchronization of Outstations with the time of the Master station. If the Outstation reports a need for time synchronization by means of an **Internal Indication** flag 1.4 an, the driver sends the current system time in accordance with the options for UTC/local time and the LAN time synchronization.

Ensure that the Outstation gets the time from a different source in this case (such as a GPS receiver) and as a result may possibly not send any requirement for time synchronization to the Master. In this case, ensure that the Runtime computer with the KDNP1 Master station is synchronized with the same time source.

HYSTERESIS

The driver supports hysteresis for spontaneous values (unsolicited responses). Hysteresis is not taken into account by:

- ▶ Values that are received as a response to a read request for an event class
- ▶ Variables that are explicitly read as classless variables

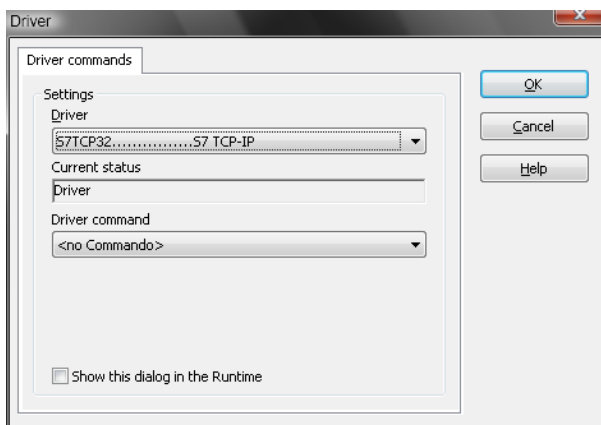
If an identical value, however with a more recent time stamp is received, then this value is sent by the driver as a new value in Runtime.

8. Driver commands

This chapter describes standard functions that are valid for most zenon drivers. Not all functions described here are available for every driver. For example, a driver that does not, according to the data sheet, support a modem connection also does not have any modem functions.

Driver commands are used to influence drivers using zenon; start and stop for example. The engineering is implemented with the help of function **Driver commands**. To do this:

- ▶ create a new function
- ▶ select Variables -> Driver commands
- ▶ The dialog for configuration is opened



Parameter	Description
Drivers	Drop-down list with all drivers which are loaded in the project.
Current status	Fixed entry which has no function in the current version.
Driver command	Drop-down list for the selection of the command.
▶ Start driver (online mode)	Driver is reinitialized and started.
▶ Stop driver (offline mode)	Driver is stopped. No new data is accepted. Note: If the driver is in offline mode, all variables that were created for this driver receive the status <code>switched off</code> (OFF; Bit 20).
▶ Driver in simulation mode	Driver is set into simulation mode. The values of all variables of the driver are simulated by the driver. No values from the connected hardware (e.g. PLC, bus system, ...) are displayed.
▶ Driver in hardware mode	Driver is set into hardware mode. For the variables of the driver the values from the connected hardware (e.g. PLC, bus system, ...) are displayed.
▶ Driver-specific command	Enter driver-specific commands. Opens input field in order to enter a command.
▶ Driver - activate set setpoint value	Write set value to a driver is allowed.
▶ Driver - deactivate set setpoint value	Write set value to a driver is prohibited.
▶ Establish connecton with modem	Establish connection (for modem drivers) Opens the input fields for the hardware address and for the telephone number.
▶ Disconnect from modem	Terminate connection (for modem drivers)
Show this dialog in the Runtime	The dialog is shown in Runtime so that changes can be made.

DRIVER COMMANDS IN THE NETWORK

If the computer, on which the **driver command** function is executed, is part of the zenon network, additional actions are carried out. A special network command is sent from the computer to the project server, which then executes the desired action on its driver. In addition, the Server sends the same driver command to the project standby. The standby also carries out the action on its driver.

This makes sure that Server and Standby are synchronized. This only works if the Server and the Standby both have a working and independent connection to the hardware.

9. Error analysis

Should there be communication problems, this chapter will assist you in finding out the error.

9.1 Analysis tool

All zenon modules such as Editor, Runtime, drivers, etc. write messages to a joint log file. To display them correctly and clearly, use the Diagnosis Viewer (main.chm::/12464.htm) program that was also installed with zenon. You can find it under Start/All programs/zenon/Tools 7.50 -> Diagviewer.

zenon driver log all errors in the LOG files. The default folder for the LOG files is subfolder **LOG** in directory `ProgramData`, example:

`%ProgramData%\COPA-DATA\LOG`. LOG files are text files with a special structure.

Attention: With the default settings, a driver only logs error information. With the **Diagnosis Viewer** you can enhance the diagnosis level for most of the drivers to "Debug" and "Deep Debug". With this the driver also logs all other important tasks and events.

In the Diagnosis Viewer you can also:

- ▶ Follow newly-created entries in real time
- ▶ customize the logging settings
- ▶ change the folder in which the LOG files are saved

Note:

1. The Diagnosis Viewer displays all entries in UTC (coordinated world time) and not in local time.
2. The Diagnosis Viewer does not display all columns of a LOG file per default. To display more columns activate property **Add all columns with entry** in the context menu of the column header.
3. If you only use **Error-Logging**, the problem description is in the column **Error text**. For other diagnosis level the description is in the column **General text**.
4. For communication problems many drivers also log error numbers which the PLC assigns to them. They are displayed in **Error text** or **Error code** or **Driver error parameter (1 and 2)**. Hints on the meaning of error codes can be found in the driver documentation and the protocol/PLC description.
5. At the end of your test set back the diagnosis level from **Debug** or **Deep Debug**. At **Debug** and **Deep Debug** there are a great deal of data for logging which are saved to the hard drive and which can influence your system performance. They are still logged even after you close the **Diagnosis Viewer**.



Attention

In Windows CE errors are not logged per default due to performance reasons.

You can find further information on the Diagnosis Viewer in the Diagnose Viewer (main.chm::/12464.htm) manual.

9.2 Check list

Questions and hints for fault isolation:

GENERAL TROUBLESHOOTING

- ▶ Is the PLC connected to the power supply?
- ▶ Analysis with the **Diagnosis Viewer** (on page 57):
-> Which messages are displayed?
- ▶ Are the participants available in the **TCP/IP** network?
- ▶ Can the PLC be reached via the `Ping` command?
Ping: Open command line -> ping <IP address> (e.g. ping 192.168.0.100) -> press Enter.
Do you receive an answer with a time or a time-out?
- ▶ Can the PLC be reached at the respective port via `TELNET`?
Telnet: Command line Enter open, telnet <IP address port number> Input (e. g. telnet 192,168,0,100 20000) -> press enter.
If the monitor turns black and the cursor blinks, a connection could be established.
- ▶ Analysis by using a network monitoring program (Sniffer, e.g. Wireshark, Microsoft Network Monitor / Microsoft Message Analyzer)
- ▶ Are you using the correct cable which is recommended by the manufacturer for the connection between the PLC and the PC?
- ▶ Did you select the right COM port?
- ▶ Do the communication parameters match (Baud rate, parity, start/stop bits,...)?
- ▶ Is the COM port blocked by another application?
- ▶ Did you configure the Net address in the address properties of the variable correctly?
 - Does the addressing match with the configuration in the driver dialog?
 - Does the net address match the address of the target station?
- ▶ Did you use the right object type for the variable?

Example: Driver variables are purely statistics variables. They do not communicate with the PLC. (See chapter Driver variable (on page 46).)

- ▶ Does the offset addressing of the variable match the one in the PLC?

SOME VARIABLES REPORT INVALID.

- ▶ INVALID bits always refer to a net address.
- ▶ At least one variable of the net address is faulty.

VALUES ARE NOT DISPLAYED, NUMERIC VALUES REMAIN EMPTY

Under circumstances, no answer can be received from the PLC for a reading-request.

Driver is not working. Check the:

- ▶ Installation of zenon
- ▶ the driver installation
- ▶ The installation of all components
-> Pay attention to error messages during the start of the Runtime.

VARIABLES ARE DISPLAYED WITH A BLUE DOT

The communication in the network is faulty:

- ▶ With a network project:
Is the network project also running on the server?
- ▶ With a stand-alone project or a network project which is also running on the server:
Deactivate the property Only read from Standby Server in node Driver connection/Addressing.

VALUES ARE DISPLAYED INCORRECTLY

Check the information for the calculation in node Value calculation of the variable properties.

Check the configuration of the Outstation, if the desired variation is sent (p. e.: **Float** with decimal places).

VALUES ARE NOT DISPLAYED PROMPTLY

Check in the Outstation, if events are generated for the selected values. Also check in which class they are generated. Check the polling of this event class in the driver configuration.

Check the configuration of **unsolicited responses** in the outstation.

THE TIME STAMP OF THE VARIABLE IS INCORRECT.

Check the configuration of the Outstation, to see whether the KDNP1 objects are actually sent with a time stamp. (corresponding variation selected with time stamp.)

Check to see if the Outstation uses local time or UTC and set the option in the driver configuration accordingly,

Check the time of the local computer and the time of the Outstation, including settings for the time zone.

DRIVER FAILS OCCASIONALLY

Analysis with the **Diagnosis Viewer** (on page 57):

-> Which messages are displayed?

Check the time-out of the response time in the driver configuration, especially with outstations with many data points and a slow connection.

With `Secure Authentication`, check whether the **pre-shared Update Key** in the driver and in the Outstation is identical.