



zenon
by COPA-DATA

zenon driver manual

IEC62056

v.8.20



© 2020 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 properties in the legal sense. Subject to change, technical or otherwise.

Contents

1	Welcome to COPA-DATA help	4
2	IEC62056.....	4
3	IEC62056 - data sheet	5
4	Driver history.....	6
5	Requirements	7
5.1	PC.....	7
5.2	PLC.....	7
6	Configuration	8
6.1	Creating a driver.....	9
6.2	Settings in the driver dialog	12
6.2.1	General	13
6.2.2	Com	17
6.2.3	Settings	19
6.3	Communication	25
7	Creating variables	25
7.1	Creating variables in the Editor	25
7.2	Addressing.....	29
7.3	Driver objects and datatypes	30
7.3.1	Driver objects.....	30
7.3.2	Mapping of the data types.....	31
8	Driver command function.....	32
9	Error analysis	37
9.1	Analysis tool.....	37
9.2	Driver monitoring	38
9.3	Check list.....	39

1 Welcome to COPA-DATA help

ZENON VIDEO TUTORIALS

You can find practical examples for project configuration with zenon in our YouTube channel (https://www.copadata.com/tutorial_menu). The tutorials are grouped according to topics and give an initial insight into working with different zenon modules. All tutorials are available in English.

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.

PROJECT SUPPORT

You can receive support for any real project you may have from our customer service team, which you can contact via email at 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.

2 IEC62056

The driver implements the protocol C of the IEC 62056-21 specification, but only for reading.

Only values with address information will be represented.

3 IEC62056 - data sheet

General:	
Driver file name	IEC62056.exe
Driver name	IEC-62056-21 driver
PLC types	IEC 62056-21 or IEC 61107-compatible energy meter
PLC manufacturer	Siemens; IEC

Driver supports:	
Protocol	IEC 62056-21; IEC 61107
Addressing: Address-based	Name based
Addressing: Name-based	--
Spontaneous communication	--
Polling communication	X
Online browsing	--
Offline browsing	--
Real-time capable	--
Blockwrite	--
Modem capable	--
RDA numerical	--
RDA String	--
Hysteresis	--
extended API	--
Supports status bit WR-SUC	--
alternative IP address	--

Requirements:	
Hardware PC	Standard network card or RS232 interface
Software PC	--
Hardware PLC	--
Software PLC	--
Requires v-dll	X

Platforms:	
Operating systems	Windows 10; Windows 7; Windows 8; Windows 8.1; Windows Server 2008 R2; Windows Server 2012; Windows Server 2012 R2; Windows Server 2016

4 Driver history

Date	Driver version	Change
8/20/2008	200	Created driver documentation
12/15/2016	33765	Timeout and repetition for devices can be configured.

DRIVER VERSIONING

The versioning of the drivers was changed with zenon 7.10. There is a cross-version build number as of this version. This is the number in the 4th position of the file version,

For example: **7.10.0.4228** means: The driver is for version **7.10** service pack **0**, and has the build number **4228**.

Expansions or error rectifications will be incorporated into a build in the future and are then available from the next consecutive build number.

Example

A driver extension was implemented in build **4228**. The driver that you are using is build number **8322**. Because the build number of your driver is higher than the build number of the extension, the extension is included. The version number of the driver (the first three digits of the file version) do not have any significance in relation to this. The drivers are version-agnostic

5 Requirements

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

5.1 PC

- ▶ For communication via TCP/IP:
A standard network card in the PC is required for communication.
- ▶ For serial communication:
A serial interface at the PC is required. Alternatively, you can also use a COM port server or a USB/serial converter.

5.2 PLC

The counter can be used with:

- ▶ Optical interface:
Tested with RS232/TVS9 **Landis & Gyr FDC 1.3** converter.
- ▶ Serial interface module:
Tested with **MOXA UPort 1150** converter of RS232 (2 wires) to USB and a **Köhler AEL.TF.09** energy counter.
- ▶ Ethernet communication:
Tested with the **CU-E20 Ethernet** module by **Landis & Gyr**.

SETTINGS FOR SERIAL CONNECTION FOR MOXA UPORT 1150 AND KÖHLER AEL.TF.09

Settings for the COM connection in the zenon driver:

- ▶ **Com:** COM 2
- ▶ **Data bit:** 7
- ▶ **Stop bit:** 1
- ▶ **Baud rate:** 300
- ▶ **Parity:** Even
- ▶ **Protocol:** RTS/CTS

Settings in the zenon driver:

- ▶ **Delay before baud-rate switch:** 200 ms

Settings in the Moxa UPort 1150 converter:

- ▶ **Port:** 1
- ▶ **COM No.:** COM2
- ▶ **Fast Flush:** Disable
- ▶ **Interface:** RS-485 2W

6 Configuration

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

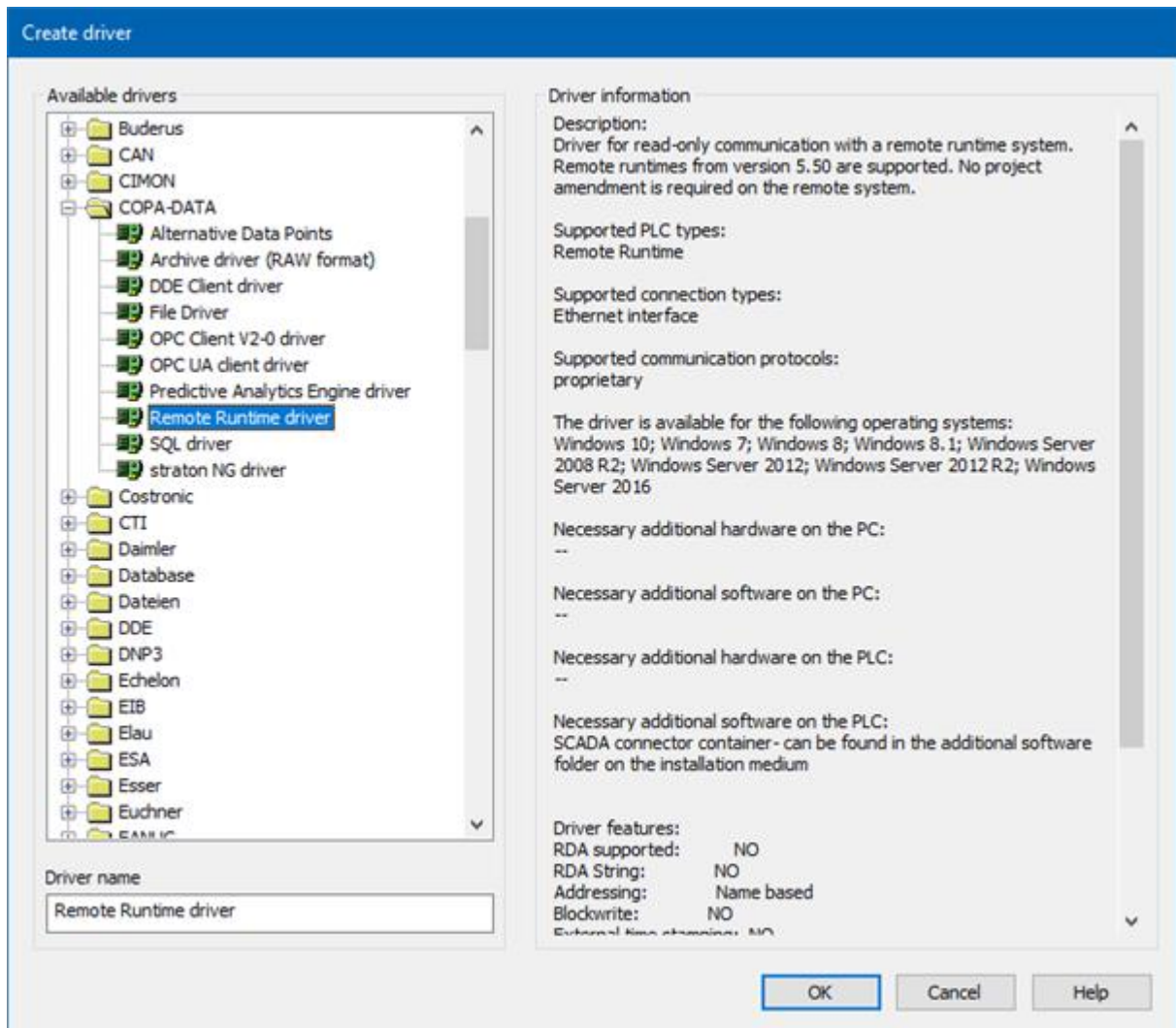


Information

Find out more about further settings for zenon variables in the chapter Variables of the online manual.

6.1 Creating a driver

In the **Create driver** dialog, you create a list of the new drivers that you want to create.



Parameter	Description
Available drivers	<p>List of all available drivers.</p> <p>The display is in a tree structure: [+] expands the folder structure and shows the drivers contained therein. [-] reduces the folder structure</p> <p>Default: <i>No selection</i></p>
Driver name	<p>Unique Identification of the driver.</p> <p>Default: <i>empty</i></p> <p>The input field is pre-filled with the pre-defined</p>

Parameter	Description
	Identification after selecting a driver from the list of available drivers.
Driver information	Further information on the selected driver. Default: <i>empty</i> The information on the selected driver is shown in this area after selecting a driver.

CLOSE DIALOG

Option	Description
OK	Accepts all settings and opens the driver configuration dialog of the selected driver.
Cancel	Discards all changes and closes the dialog.
Help	Opens online help.



Information

The content of this dialog is saved in the file called Treiber_[Language].xml. You can find this file in the following folder:

C:\ProgramData\COPA-DATA\zenon[version number].

CREATE NEW DRIVER

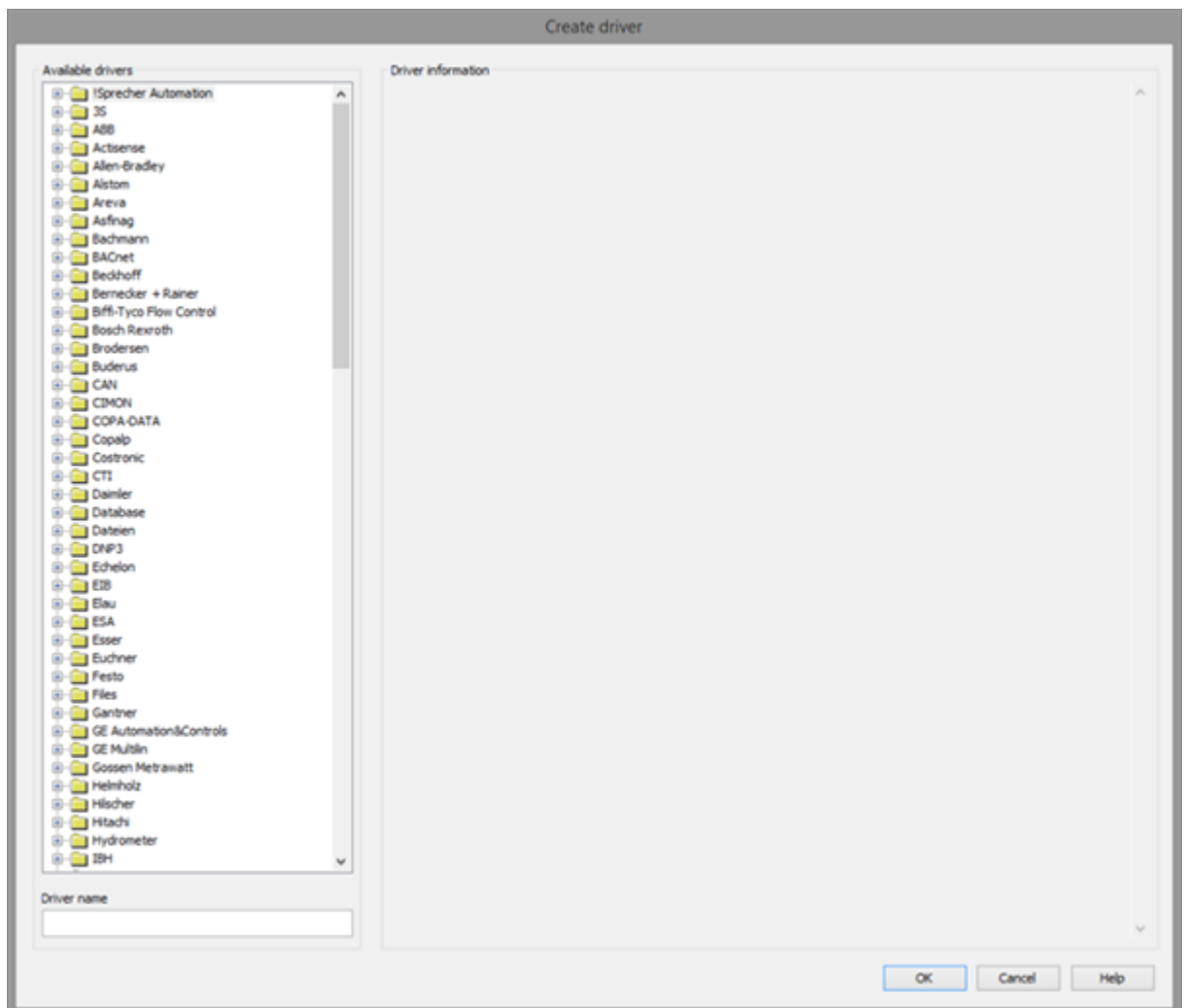
In order to create a new driver:

1. Right-click on **Driver** in the Project Manager and select **New driver** in the context menu.

Optional: Select the **New driver** button from the toolbar of the detail view of the **Variables**. The Create driver dialog is opened.

The **Create simple data type** dialog is opened.

2. The dialog offers a list of all available drivers.

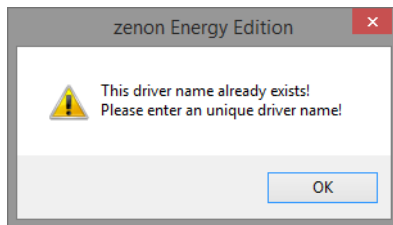


3. Select the desired driver and name it in the **Driver name** input field.
This input field corresponds to the **Identification** property. The name of the selected driver is automatically inserted into this input field by default.
The following is applicable for the **Driver name**:
 - ▶ The **Driver name** must be unique.
If a driver is used more than once in a project, a new name has to be given each time.
This is evaluated by clicking on the **OK** button. If the driver is already present in the project, this is shown with a warning dialog.
 - ▶ 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 by clicking on the **OK** button.
The configuration dialog for the selected driver is opened.

Note: The language of driver names cannot be switched. They are always shown in the language in which they have been created, regardless of the language of the Editor. This also applies to driver object types.

DRIVER NAME DIALOG ALREADY EXISTS

If there is already a driver in the project, this is shown in a dialog. The warning dialog is closed by clicking on the **OK** button. The driver can be named correctly.



ZENON PROJECT

The following drivers are created automatically for newly-created projects:

- ▶ **Intern**
- ▶ **MathDr32**
- ▶ **SysDrv**

Information

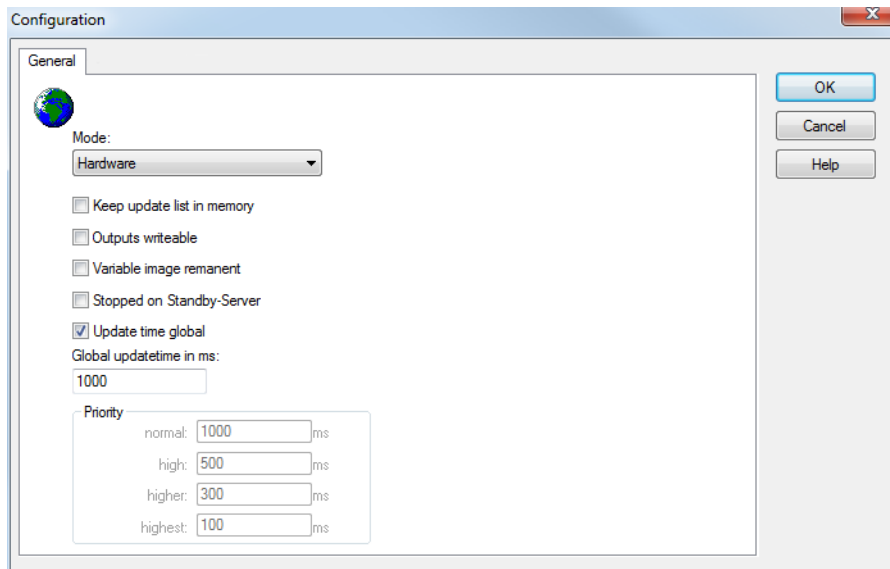
Only the required drivers need to be present in a zenon project. Drivers can be added at a later time if required.

6.2 Settings in the driver dialog

You can change the following settings of the driver:

6.2.1 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.



Option	Description
Mode	<p>Allows to switch between hardware mode and simulation mode</p> <ul style="list-style-type: none"> ▶ <i>Hardware:</i> A connection to the control is established. ▶ <i>Simulation - static:</i> 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. ▶ <i>Simulation - counting:</i> 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. ▶ <i>Simulation - programmed:</i> No communication is established to the PLC. The

Option	Description
	<p>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.</p> <p>For details see chapter Driver simulation.</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	<ul style="list-style-type: none"> ▶ <i>Active:</i> Outputs can be written. ▶ <i>Inactive:</i> Writing of outputs is prevented. <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 hardware mode if one of these statuses is active:</p> <ul style="list-style-type: none"> ▶ User status <i>M1 (0) to M8 (7)</i> ▶ <i>REVISION(9)</i> ▶ <i>AUS(20)</i> ▶ <i>ERSATZWERT(27)</i> <p>The variable image is always saved if:</p> <ul style="list-style-type: none"> ▶ the variable is of the Communication details object type ▶ the driver runs in simulation mode. (not programmed simulation) <p>The following states are not restored at the start of the Runtime:</p>

Option	Description
	<ul style="list-style-type: none"> ▶ <i>SELECT(8)</i> ▶ <i>WR-ACK(40)</i> ▶ <i>WR-SUC(41)</i> <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 one 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> <ul style="list-style-type: none"> ▶ <i>Active:</i> 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 but an empty value. This prevents that at the upgrade to the Server irrelevant values are created in the AML, CEL and Historian. <p>Default: <i>inactive</i></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>Setting for the global update times in milliseconds:</p> <ul style="list-style-type: none"> ▶ <i>Active:</i> The set Global update time is used for all variables in the project. The priority set at the variables is not used. ▶ <i>Inactive:</i> The set priorities are used for the individual variables. <p>Exceptions: Spontaneous drivers ignore this option. They generally use the shortest possible update time. For details, see the Spontaneous driver update time section.</p>

Option	Description
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 variables are allocated separately in the settings of the variable properties.</p> <p>The communication of the individual variables can be graded according to importance or required topicality using the priority classes. Thus the communication load is distributed better.</p> <p>Attention: Priority classes are not supported by each driver, e.g. spontaneously communicating zenon drivers.</p>

CLOSE DIALOG

Option	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.

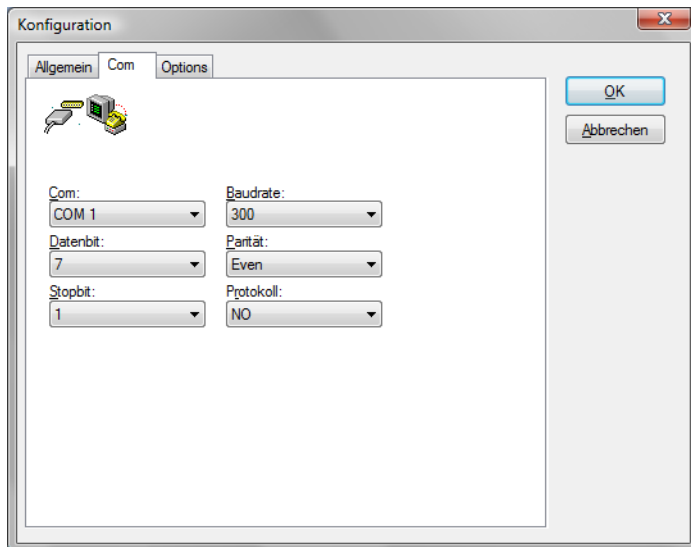
UPDATE TIME FOR SPONTANEOUS DRIVERS

With spontaneous drivers, for **Set value, advising** of variables and **Requests**, a read cycle is triggered immediately - regardless of the set update time. This ensures that the value is immediately available for visualization after writing. The update time is generally 100 ms.

Spontaneous drivers are **ArchDrv**, **BiffiDCM**, **BrTcp32**, **DNP3**, **Esser32**, **FipDrv32**, **FpcDrv32**, **IEC850**, **IEC870**, **IEC870_103**, **Otis**, **RTK9000**, **S7DCOS**, **SAIA_Slave**, **STRATON32** and **Trend32**.

6.2.2 Com

In this dialog, you configure the connection parameters for the serial communication to the PLC.



⚠ Attention

Note the interface parameters defined in accordance with the IEC62056 standard:

- ▶ **Baudrate:**
300
- ▶ **Datenbit:**
7
- ▶ **Stopbit:**
1
- ▶ **Parität:**
Even
- ▶ **Protokoll:**
NO

These standard-compliant parameters are not the driver's default parameters!

COM (DEFAULT SETTING)

Parameter	Description
Com	Selection of the serial interface (com port) on the computer. Select from drop-down list: <i>COM 1 to COM 256</i>

Parameter	Description
	Default: 1
Baud rate	<p>Selection baud rate. Amend to controller. Select from drop-down list:</p> <p>Default: 9600</p> <p>Input range: 110 to 256000</p>
Data bit	<p>Number of data bits (data word length in bits) for communication to the PLC. The data bit rate must be amended to the controller. Select from drop-down list: 5, 6, 7, 8</p> <p>Default: 8</p>
Stop bit	<p>Selection of the stop bit for communication to the PLC. The stop bit must be amended to the controller. Select from drop-down list:</p> <ul style="list-style-type: none"> ▶ 1 ▶ 1.5 ▶ 2 <p>Default: 1</p>
Parity	<p>Selection of the parity for the communication to the PLC. The parity must be amended to the controller. Amend to controller. Select from drop-down list:</p> <ul style="list-style-type: none"> ▶ No ▶ Odd ▶ Even <p>Default: No</p>
Protocol	<p>Selection of the protocol for communication to the PLC. The protocol must be amended to the controller. Select from drop-down list:</p> <ul style="list-style-type: none"> ▶ No ▶ Xon/Xoff

Parameter	Description
	<ul style="list-style-type: none"> ▶ <i>Rts/CTS</i> ▶ <i>DTR/DSR</i> <p>Default: <i>No</i></p>

CLOSE DIALOG

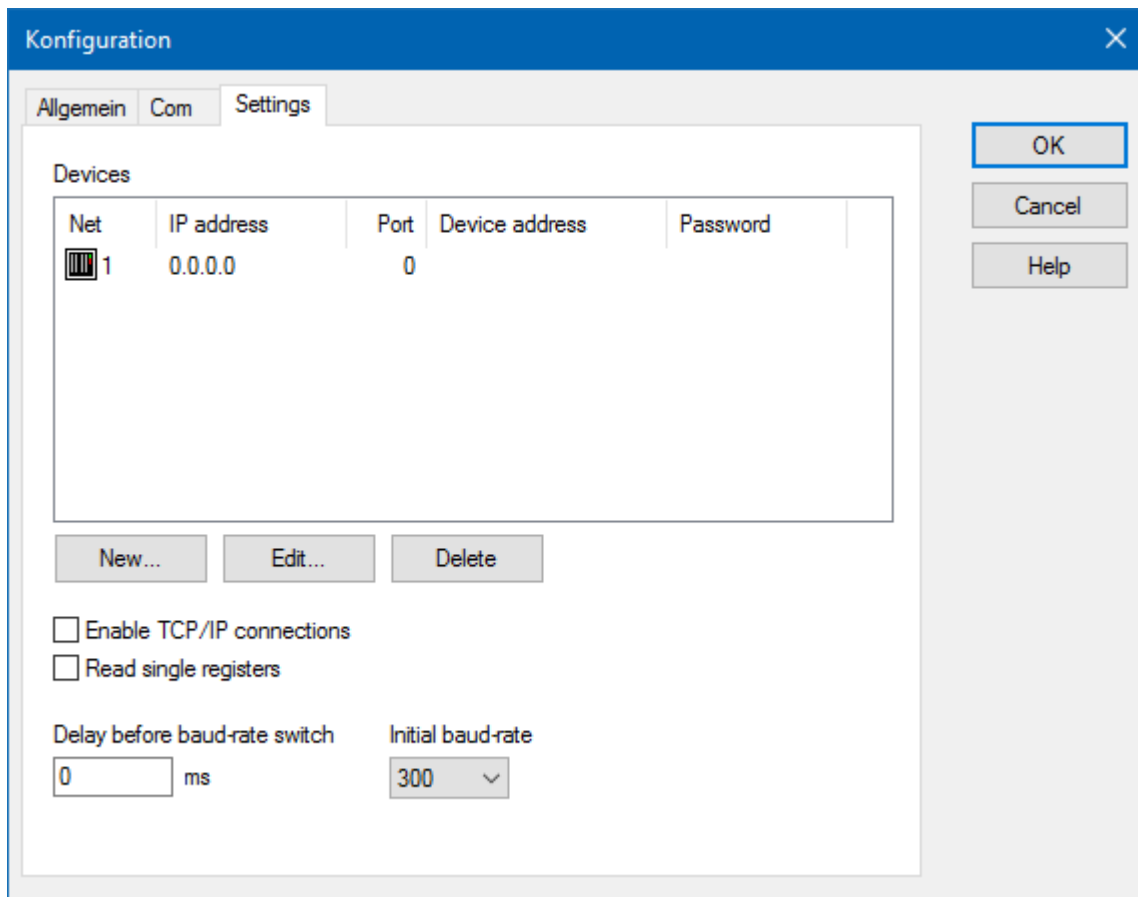
Option	Description
OK	Applies all changes in all tabs and closes the dialog.
Cancel	Discards all changes in all tabs and closes the dialog.
Help	Opens online help.

6.2.3 Settings

The driver support both serial communication and communication via TCP/IP. Only one of the two communication types can be used per device.

Note: This dialog is only available in English.

The buttons are displayed in the system language of the computer.



DEVICES

List of configured stations.

The column width can be set by clicking on the column divider with the mouse button held down.

BUTTONS - MANAGE STATIONS

Buttons for the processing, new creation or deletion of settings for the connections to stations.

Parameter	Description
New...	Opens the New Station (on page 22) dialog to add and configure a station.
Edit...	Opens the New Station (on page 22) for changes to the configuration of the selected station.
Delete	Deletes the selected station. Note: Deletion is only carried out after confirmation in a confirmation

Parameter	Description
	dialog.

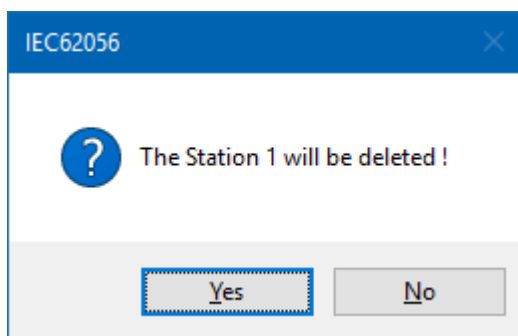
COMMUNICATION DETAILS

Parameter	Description
Enable TCP/IP Connections	<p>Checkbox to activate TCP/IP communication.</p> <ul style="list-style-type: none"> ▶ <i>activated:</i> The driver communicates via TCP/IP. ▶ <i>Not activated:</i> The driver communicates with the PLC by means of a serial interface. <p>Note: As a result, not all parameters in the configuration dialog of a station are active.</p> <p>Default: <i>inactive</i></p>
Read single registers	<p>Checkbox to activate the reading of individual registers.</p> <ul style="list-style-type: none"> ▶ <i>Not activated:</i> Driver reads the complete Table 1 from the counter. ▶ <i>activated:</i> Driver switches to <i>programming mode</i> and reads each variable individually. The corresponding IEC 62056 address of the variable must be configured beforehand. <p>Default: <i>inactive</i></p> <p>Note: Not every counter supports the use of single registers. You can find information on this in the documentation of the Oracle client used.</p>
Delay before baud-rate switch (ms)	<p>Delay before switching the configured baud rate.</p> <p>Default: <i>0 ms</i></p> <p>Tested values:</p> <ul style="list-style-type: none"> ▶ EMH counter with Variomod XC converter: <i>0 ms</i> ▶ Köhler AEL.TF.09 with MOXA UPort 1150 converter: <i>200 ms</i>
Initial baud-rate	<p>Initial Baud rate that is used when a connection is initialized.</p> <p>Default: <i>300 ms</i></p>

CLOSE DIALOG

Option	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.

CONFIRMATION DIALOG ON DELETION

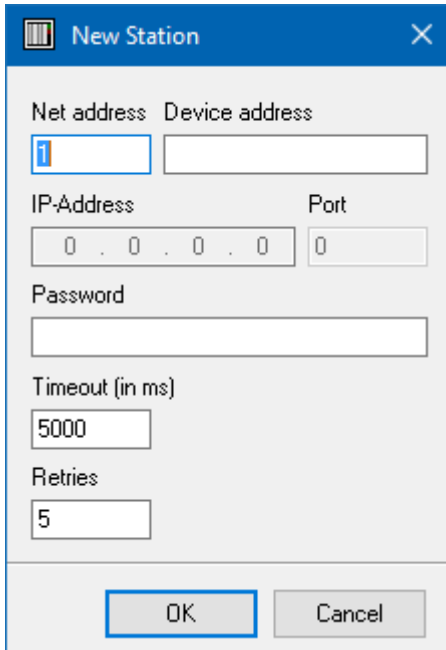


Parameter	Description
Yes	Confirms the deletion of the selected station. The station is deleted and the confirmation dialog is closed after clicking on Yes .
No	Cancels deletion and closes the confirmation dialog.

6.2.3.1 New Station

Configuration dialog of a station.

Note: This dialog is only available in English.



The 'New Station' dialog box contains the following fields and controls:

- Net address:** A text box containing the value '1'.
- Device address:** An empty text box.
- IP-Address:** A field with five segments, each containing a '0'.
- Port:** A text box containing the value '0'.
- Password:** An empty text box.
- Timeout (in ms):** A text box containing the value '5000'.
- Retries:** A text box containing the value '5'.
- Buttons:** 'OK' and 'Cancel' buttons at the bottom.

Parameter	Description
Net address	<p>You can choose any number for allocating variables to the station in the driver. The number is unique for each driver. If the address has already been configured, this is shown with a warning dialog.</p> <p>Note: When configuring a variable, this address is to be used in the Net address.</p> <p>Range: 0-255</p> <p>Default: 1</p> <p>The net address is not automatically counted further when a new station is created.</p>
Device address	<p>Address of the station</p> <p>Default: <i>empty</i></p> <p>Note: not active if Enable TCP/IP connections is activated.</p>
IP address	<p>IP address of the station.</p> <p>Default: 0.0.0.0</p> <p>Note: Incorrect entries are automatically corrected to the value 255.</p> <p>Note: not active for TCP/IP connections (activated option: Enable TCP/IP connections).</p>
IP port	IP port that is used for communication to the station.

Parameter	Description
	Default: 0 Note: not active for TCP/IP connections (activated option: Enable TCP/IP connections).
Password	Input field for optional password for login to the PLC. The password is only required if the PLC needs one for login. Default: <i>empty</i>
Timeout	Waiting time in milliseconds before a communication error occurs. Default: 5000
Retries	Request retries in the event of a communication problem. Default: 5 Note: in the event of an error (communication timeout and repetitions fail) the driver gets the status "invalid". No further data is received from the driver.

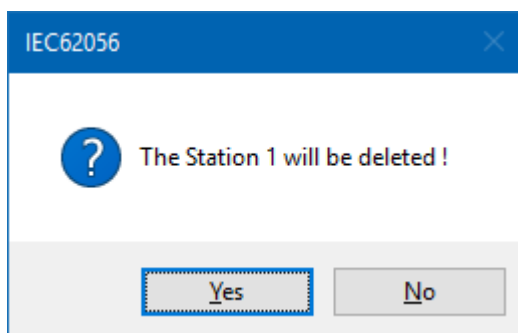
CLOSE DIALOG

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

WARNING DIALOG: NET ADDRESS ALREADY EXISTS

Note: This dialog is only available in English.

The buttons are displayed in the system language of the computer.



Parameter	Description
Yes	The current configuration overwrites the existing configuration for the

Parameter	Description
	station.
No	Discards all changes and closes the dialog.

6.3 Communication

The following is applicable for communication to the PLC:

- ▶ The **IEC62056 driver** communicates in *Mode C "programming mode"*. It can therefore address all data points in the counter of the station. Any access password that may be necessary for this must be configured in the driver configuration for each counter.
- ▶ The variable name can be configured via a driver specific property. If this property is empty, the variable identification is used as the variable name.
- ▶ The time stamp can also be rounded up or down to the value of a start or end of an adjustable interval using two driver-specific properties.
variable property group **Addressing**:
 - ▶ **IEC62056 Time Adjustment**
 - ▶ **IEC62056 Time Interval**
- ▶ To set the time to C003 by means of R2 command, the **Clock Sync (BOOL)** driver object type is used.
 - ▶ Writing to this variable sends the command to the counter with the current time stamp.
 - ▶ Variables in this area are *write-only*; they do not contain an actual value.
 - ▶ This net address is used to reference the counter or router to which the command is sent. (some routers offer this C003 area globally; the time is then forwarded to all counters connected to the router --> See router manual)

7 Creating variables

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

7.1 Creating variables in the Editor

Variables can be created:

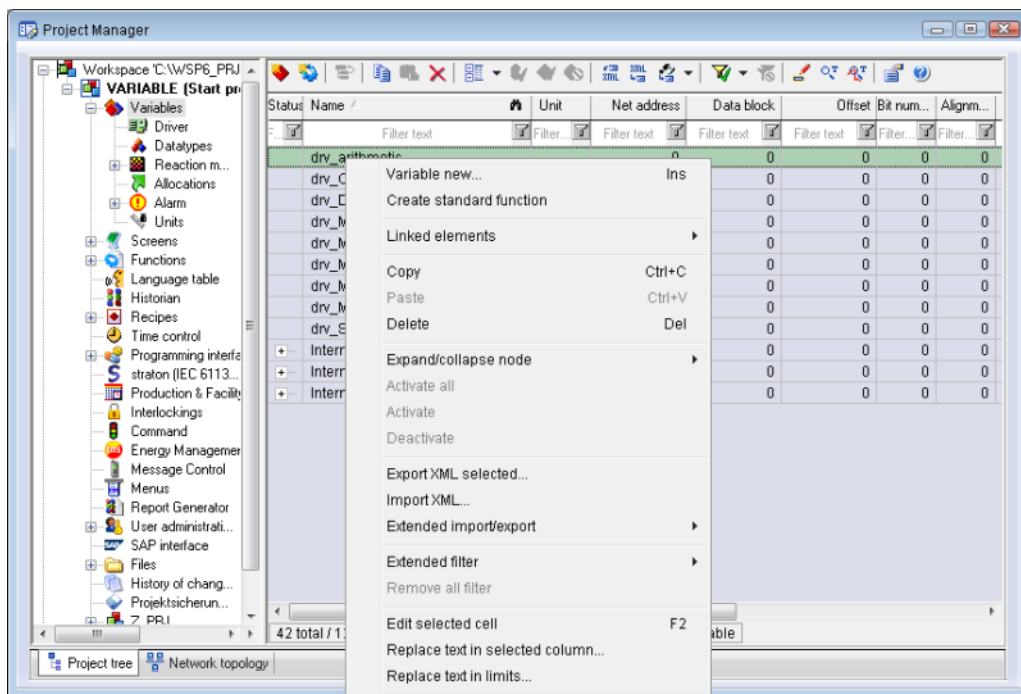
- ▶ as simple variables

- ▶ in arrays
- ▶ as structure variables

VARIABLE DIALOG

To create a new variable, regardless of which type:

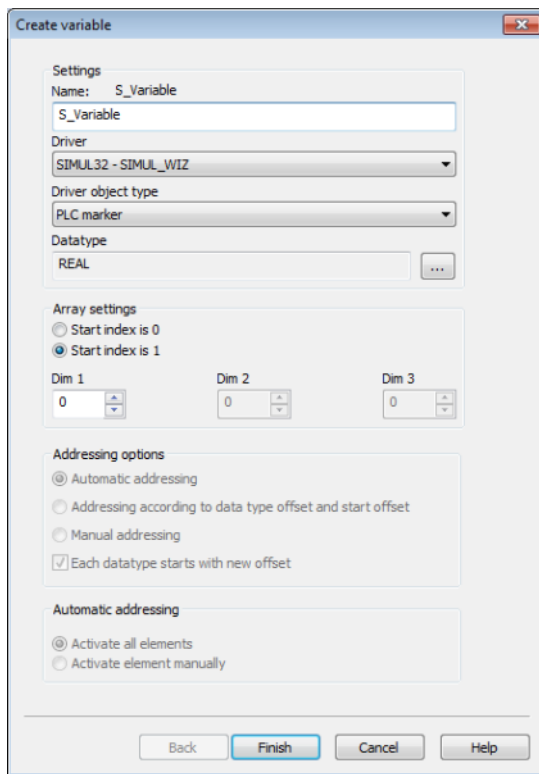
1. Select the **New variable** command in the **Variables** node in the context menu



The dialog for configuring variables is opened

2. Configure the variable
3. The settings that are possible depend on the type of variables

CREATE VARIABLE DIALOG



Property	Description
Name	<p>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.</p> <p>Maximum length: 128 characters</p> <p>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.</p> <p>Note: Some drivers also allow addressing using the Symbolic address property.</p>
Driver	<p>Select the desired driver from the drop-down list.</p> <p>Note: If no driver has been opened in the project, the driver for internal variables (Intern.exe) is automatically loaded.</p>
Driver Object Type	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

Property	Description
	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.

The following drivers support the **Symbolic address**:

- ▶ 3S_V3
- ▶ AzureDrv
- ▶ BACnetNG
- ▶ IEC850
- ▶ KabaDPSTServer
- ▶ OPCUA32
- ▶ Phoenix32
- ▶ POZYTON
- ▶ RemoteRT
- ▶ S7TIA
- ▶ SEL
- ▶ SnmpNg32
- ▶ PA_Drv
- ▶ EUROMAP63

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.

7.2 Addressing

Property	Description
Name	Freely definable name. Attention: For every zenon project the name must be unambiguous.
Identification	Is used for the addressing if the IEC62056 Address option is empty. Example: 1.2.0
Net address	Network address of variable. This address refers to the bus address in the connection configuration of the driver. This defines the station, on which the variable resides.
Data block	not used for this driver
Offset	not used for this driver
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 Object Type	Object type of the variables. Depending on the driver used, is selected when the variable is created and can be changed here.
Driver connection/Data Type	Data type of the variable. Is selected during the creation of the variable; the type can be changed here. Attention: If you change the data type later, all other properties of the variable must be checked and adjusted, if necessary.
Symbolic address	Character sequence for symbolic address of the variables.
IEC62056 Address	Entry of the OBIS address. *? format is supported for historical values. Input historical values: If the address is entered with a subsequent *? , instead of the ? , the

Property	Description
	historical value in Table 1 is assigned. Example: Entering 1.8.0*? leads to the output 1.8.0*47 . This type of addressing is now permitted when reading Table 1 .

7.3 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.

7.3.1 Driver objects

The following object types are available in this driver:

Driver Object Type	Channel type	Read	Write	Supported data types	Description
Input	10	X	--	UDINT, DINT, LREAL, STRING, DATE_AND_TIME	
Clock Sync	9	--	X	BOOL	The net address references the counter/router to which the time is sent
<i>Communication details</i>	35	X	X	BOOL, SINT, USINT, INT, UINT, DINT, UDINT, REAL, STRING	Variables for the static analysis of the communication; Values are transferred between driver and Runtime (not to the PLC). Note: The addressing and the behavior is the same for most zenon drivers. You can find detailed information on this in the

Driver Object Type	Channel type	Read	Write	Supported data types	Description
					Communication details (Driver variables) chapter.

Key:

X: supported

--: not supported

CHANNEL TYPE

The term **Kanaltyp** is the internal numerical name of the driver object type. It is also used for the extended DBF import/export of the variables.

"**Kanaltyp**" is used for advanced CSV import/export of variables in the "**HWObjectType**" column.

7.3.2 Mapping of the data types

All variables in zenon are derived from IEC data types. The following table compares the IEC datatypes with the datatypes of the PLC.

PLC	zenon	Data type
-	BOOL	8
-	USINT	9
-	SINT	10
-	UINT	2
-	INT	1
UDINT	UDINT	4
DINT	DINT	3
-	ULINT	27
-	LINT	26
-	REAL	5

PLC	zenon	Data type
LREAL	LREAL	6
STRING	STRING	12
-	WSTRING	21
-	DATE	18
-	TIME	17
DATE_AND_TIME	DATE_AND_TIME	20
-	TOD (Time of Day)	19

DATA TYPE

The term **data type** is the internal numerical identification of the data type. It is also used for the extended DBF import/export of the variables.

8 Driver command function

The zenon **Driver commands** function is to influence drivers using zenon. You can do the following with a driver command:

- ▶ Start
- ▶ Stop
- ▶ Shift a certain driver mode
- ▶ Instigate certain actions

Note: 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.

Attention

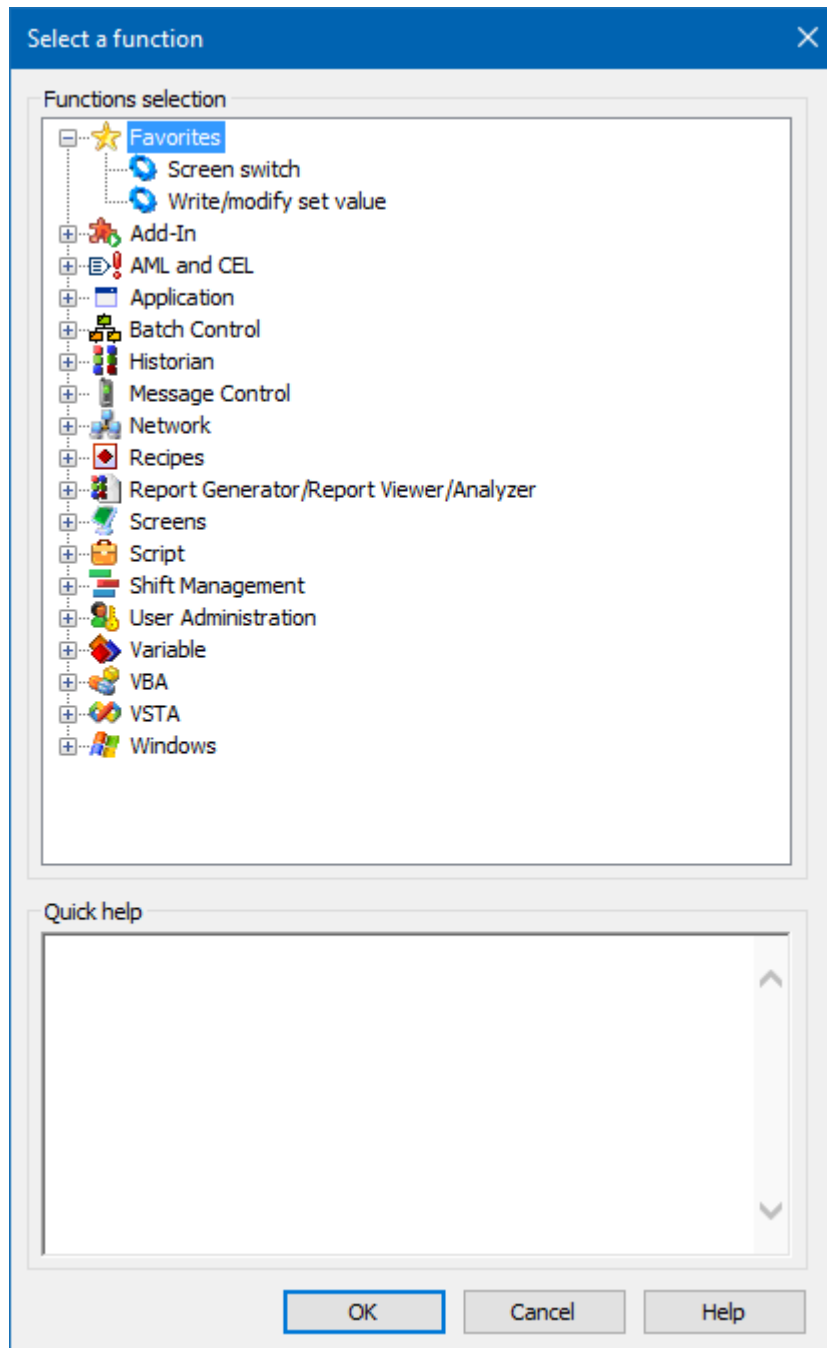
The zenon **Driver commands** function is not identical to driver commands that can be executed in the Runtime with Energy drivers!

CONFIGURATION OF THE FUNCTION

Configuration is carried out using the **Driver commands** function.
To configure the function:

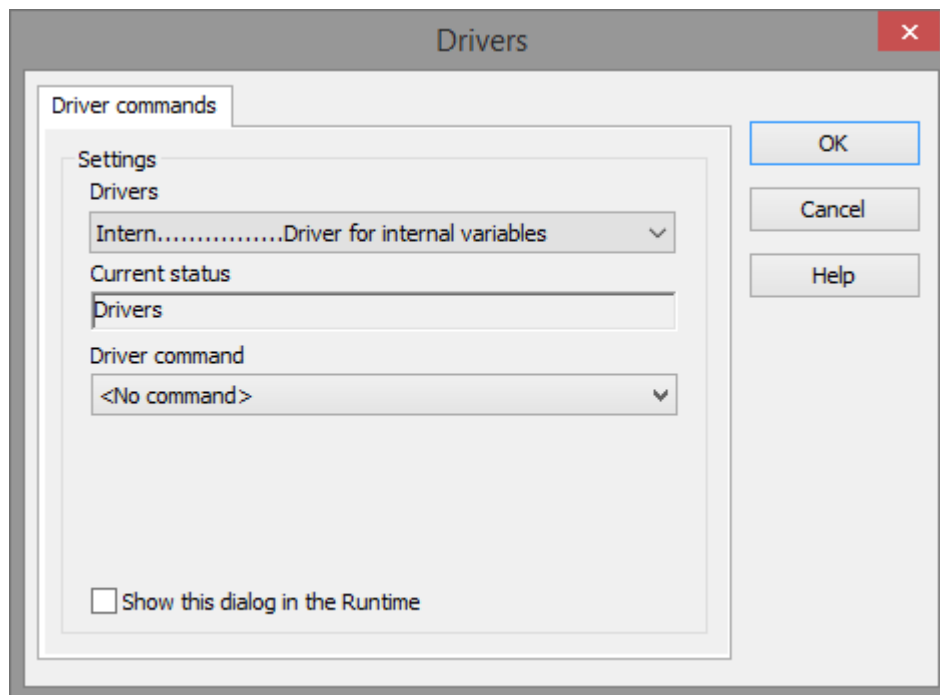
1. Create a new function in the zenon Editor.

The dialog for selecting a function is opened



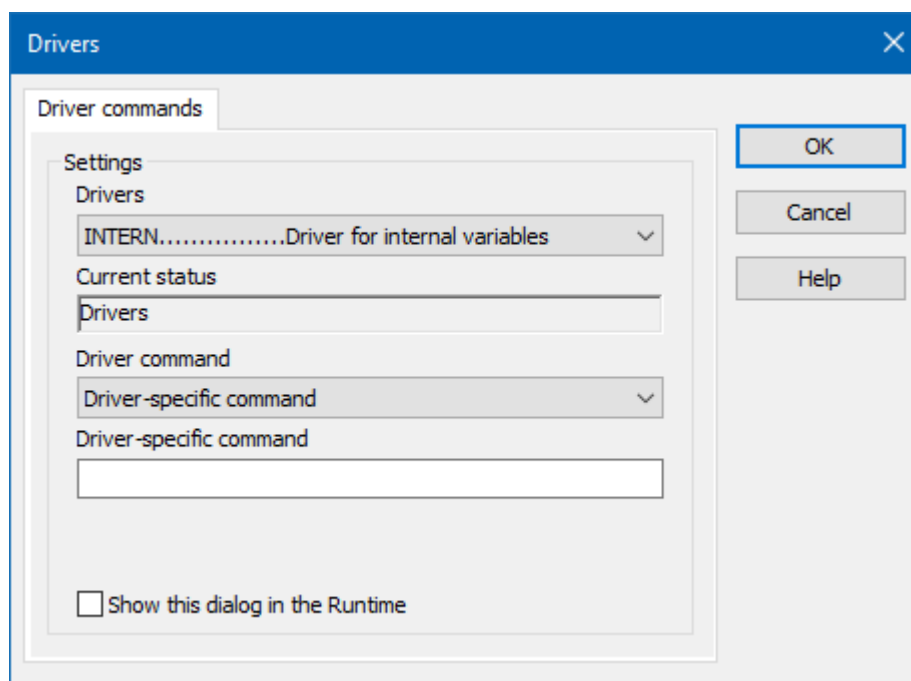
2. Navigate to the node **Variable**.
3. Select the **Driver commands** entry.

The dialog for configuration is opened



4. Select the desired driver and the required command.
5. Close the dialog by clicking on **OK** and ensure that the function is executed in the Runtime. Heed the notices in the **Driver command function in the network** section.

DRIVER COMMAND DIALOG



Option	Description
Driver	Selection of the driver from the drop-down list. It contains all drivers loaded in the project.
Current condition	Fixed entry that is set by the system. no function in the current version.
Driver command	no function in the current version. For details on the configurable driver commands, see the available driver commands section.
Driver-specific command	Entry of a command specific to the selected driver. Note: Only available if, for the driver command option, the <i>driver-specific command</i> has been selected.
Show this dialog in the Runtime	Configuration of whether the configuration can be changed in the Runtime: <ul style="list-style-type: none"> ▶ <i>Active</i>: This dialog is opened in the Runtime before executing the function. The configuration can thus still be changed in the Runtime before execution. ▶ <i>Inactive</i>: The Editor configuration is applied in the Runtime when executing the function. Default: <i>inactive</i>

CLOSE DIALOG

Options	Description
OK	Applies settings and closes the dialog.
Cancel	Discards all changes and closes the dialog.
Help	Opens online help.

AVAILABLE DRIVER COMMANDS

These driver commands are available - depending on the selected driver:

Driver command	Description
<i>No command</i>	No command is sent. A command that already exists can thus be removed from a configured function.

Driver command	Description
<i>Start driver (online mode)</i>	Driver is reinitialized and started. Note: If the driver has already been started, it must be stopped. Only then can the driver be re-initialized and started.
<i>Stop driver (offline mode)</i>	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 <i>switched off</i> (OFF; Bit 20).
<i>Driver in simulation mode</i>	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.
<i>Driver in hardware mode</i>	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.
<i>Driver-specific command</i>	Entry of a driver-specific command. Opens input field in order to enter a command.
<i>Driver - activate set setpoint value</i>	Write set value to a driver is possible.
<i>Driver - deactivate set setpoint value</i>	Write set value to a driver is prohibited.
<i>Establish connection with modem</i>	Establish connection (for modem drivers) Opens the input fields for the hardware address and for the telephone number.
<i>Disconnect from modem</i>	Terminate connection (for modem drivers)
<i>Driver in counting simulation mode</i>	Driver is set into counting simulation mode. All values are initialized with 0 and incremented in the set update time by 1 each time up to the maximum value and then start at 0 again.
<i>Driver in static simulation mode</i>	No communication to the controller is established. All values are initialized with 0.
<i>Driver in programmed simulation mode</i>	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 the zenon Logic Runtime.

DRIVER COMMAND FUNCTION IN THE NETWORK

If the computer on which the **Driver commands** function is executed is part of the zenon network, further actions are also carried out:

- ▶ A special network command is sent from the computer to the project server. It 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 program that was also installed with zenon. You can find it under **Start/All programs/zenon/Tools 8.20 -> Diagviewer**.

zenon driver log all errors in the LOG files. LOG files are text files with a special structure. The default folder for the LOG files is subfolder **LOG** in the folder **ProgramData**. For example:

%ProgramData%\COPA-DATA\LOG.

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 manual.

9.2 Driver monitoring

Runtime monitors the availability of the driver by means of a watchdog. If a driver is no longer available, the *INVALID* status bit is also set for all checked-in variables.

Possible causes for a triggering of the watchdog:

- ▶ The driver process is no longer running.
Check whether the driver EXE file is still running in the Task Manager.
- ▶ Operating system is busy with processes that have a higher priority.
Check the configuration of your system to see whether there is sufficient memory and CPU power. In this case, the driver only resets the *INVALID* status bit if there is a value change on the connected party. Static values retain the *INVALID* status bit until the next time the Runtime or the driver is started.

CONFIGURATION OF WATCHDOG

For the monitoring of communication in the Runtime, the connection to the driver is checked in a fixed, prescribed time period of 60 seconds. This process is repeated several times. If, within 5 attempts (= within 5 minutes), no valid connection to the driver is detected, the *INVALID* bit is set for the checked-in (*advised*) variables. In addition, the *INVALID* bit is also set when new variables are advised. The *INVALID* bit will no longer be reset.

Corresponding LOG entries are created for this.

LOG ENTRY

An error message is logged in the LOG when the watchdog is triggered:

Parameter	Description
<i>Communication with driver:<drvExe>/<drvDesc>(id:<drvId>) timed out. No communication for <time> ms.</i>	No communication with driver within the given time. <ul style="list-style-type: none"> ▶ <time>: Time (in milliseconds) ▶ <drvDesc>: Driver name ▶ <drvExe>: Driver EXE name ▶ <drvId>: Driver ID in the zenon project
<i>Communication with %s timed out. Invalid-Bit will be set.</i>	Communication to the %s driver could not be established after 5 attempts within 60 seconds. The <i>INVALID</i> bit is set for the variable.
<i>Communication with %s timed out. Timeout happened %d times</i>	Communication to the %s driver could not be established after %d times within 60 seconds.

9.3 Check list

Is the station connected to the power supply
Are the participants available in the TCP/IP network
Can the station be reached via the PING command
Can the station be reached via TELNET?
Are the station and the PLC connected with the right cable
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 correctly, both in the driver dialog and in the address properties of the variable
Did you use the right object type for the variable
Does the offset addressing of the variable match the one in the station?

Is the station connected to the power supply

Use the DiagViewer for further analysis -> Which messages does it show