# zenon driver manual

## IEC870_10332

v.7.20

# Contents

# 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. IEC870_10332

Driver for the IEC 60870-5-103 protocol (serial or TCP/IP).

Communication between the control system and the PLC is based on IEC 60870-5-103. On protocol level, the control system (''the control system"- primary station according to IEC 60870-5-2) acts as the Master and the PLC ("the protection equipment") as the Slave.

The driver sends the ASDUs (Application Service Data Units) in a so-called control direction, the PLC in monitor direction.

In the serial case, zenon works as the master in unbalanced communication mode. The communication channel can be shared between a 60870 Master and several 60870 Slaves. For TCP/IP, the control system acts as the Master on protocol level and as a Client on TCP level.

The communication is spontaneous, Additionally, the driver cyclically sends out General Interrogations according to the **GenInt** setting in the driver configuration.

# 3. IEC870_10332 - Data sheet

| **General:** | |
|---|---|
| Driver file name | IEC870_10332.exe |
| Driver name | IEC60870 103 |
| PLC types | IEC60870-5-103 compatible controls |
| PLC manufacturer | IEC; |

| **Driver supports:** | |
|---|---|
| Protocol | IEC 60870-5-103; |
| Addressing: Address-based | x |
| Addressing: Name-based | - |
| Spontaneous communication | x |
| Polling communication | - |
| Online browsing | - |
| Offline browsing | - |
| Real-time capable | - |
| Blockwrite | - |
| Modem capable | - |
| Serial logging | x |
| RDA numerical | - |
| RDA String | - |

| Requirements: | |
|---|---|
| Hardware PC | Standard Networkcard; serial port RS232 or RS 485. |
| Software PC | - |
| Hardware PLC | - |
| Software PLC | - |
| Requires v-dll | x |

| Platforms: | |
|---|---|
| Operating systems | Windows 7, 8, 8.1 Server 2008R2, Server 2012, Server 2012R2; |
| CE platforms | -; |

# 4. Driver history

| Date | Driver version | Change |
|---|---|---|
| 07.07.08 | 400 | Created driver documentation |
| 11/14/2008 | 1200 | - Driver object type MONITORING TRANSIENT<br>- INVALID bit is available<br>- Measurands as REAL<br>- Time stamp taken over from external device<br>- Entries to LOG<br>- Subdirectories (per Net address) for the disturbance data files |
| 3/23/2009 | 1400 | - Driver object type CONNECTION STATE<br>- Driver object type TIME SYNC<br>- PN Bit for commands is available<br>- SB Bit is available (for variables that have not answered in GI)<br>- the Cause of Transmission (COT) is available |

#### 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 form 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.

# 6. Configuration

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

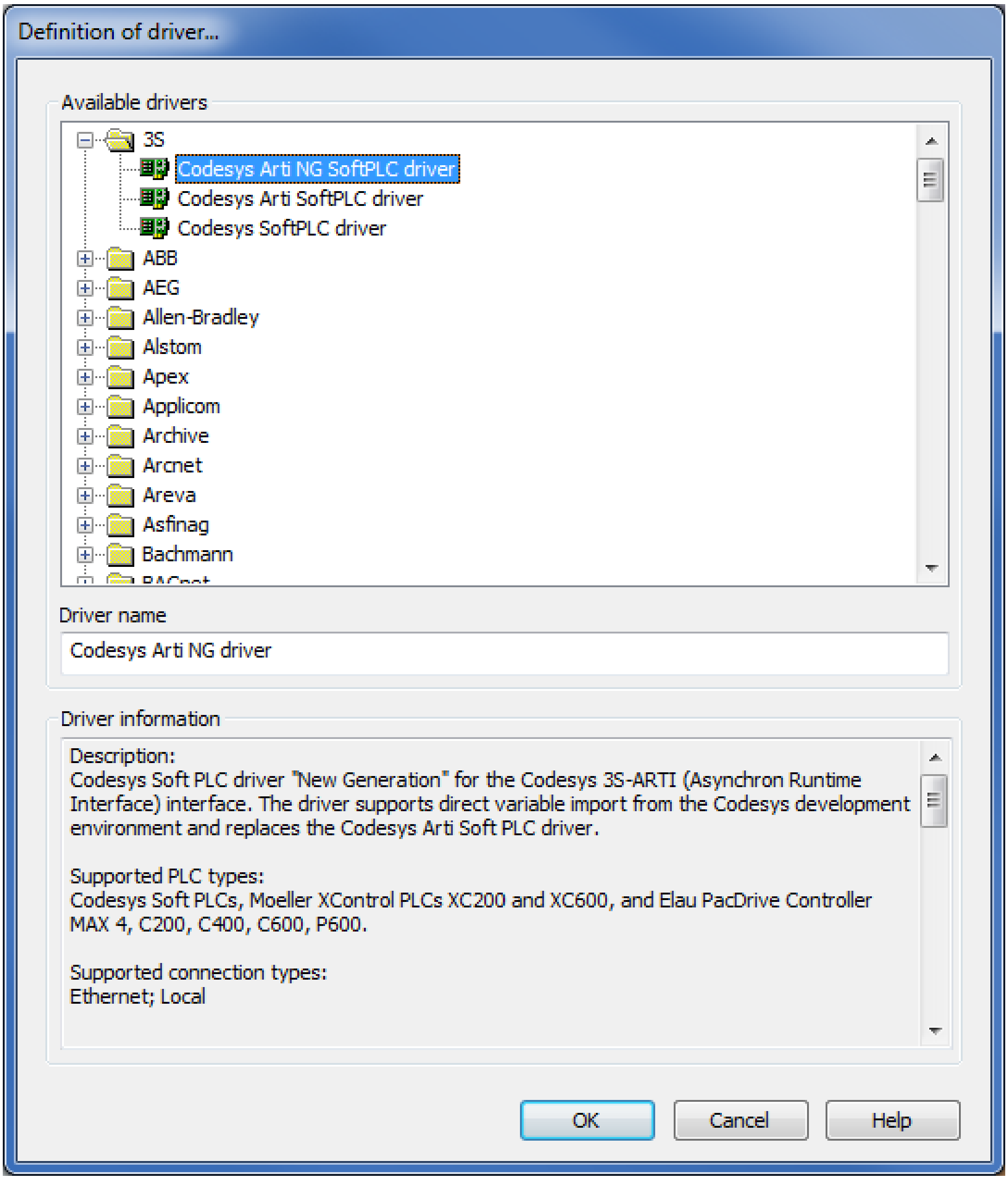
> **Information**
>
> *Find out more about further settings for zenon variables in the chapter Variables (main.chm::/15247.htm) of the online manual.*

## 6.1 Creating a driver

In order to create a new driver:

1. Right-click on `Driver` in the Project Manage and select `Driver new` in the context menu.

2. In the following dialog the control system offers a list of all available drivers.



Driver name

Codesys Arti NG driver

Driver information

Description:
Codesys Soft PLC driver "New Generation" for the Codesys 3S-ARTI (Asynchron Runtime Interface) interface. The driver supports direct variable import from the Codesys development environment and replaces the Codesys Arti Soft PLC driver.

Supported PLC types:
Codesys Soft PLCs, Moeller XControl PLCs XC200 and XC600, and Elau PacDrive Controller MAX 4, C200, C400, C600, P600.

Supported connection types:
Ethernet; Local

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:*

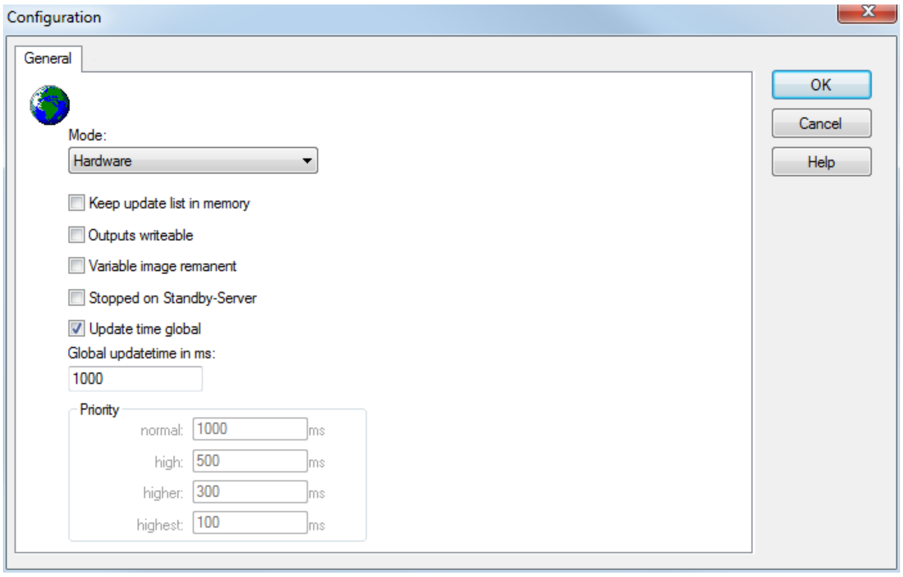
- Internal
- MathDr32
- SysDrv.

## 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.



Configuration

General

Mode: Hardware

□ Keep update list in memory

□ Outputs writeable

□ Variable image remanent

□ Stopped on Standby-Server

☑ Update time global

Global updatetime in ms: 1000

Priority

normal: 1000 ms

high: 500 ms

higher: 300 ms

highest: 100 ms

| Parameters | Description |
|---|---|
| **Mode** | Allows to switch between hardware mode and simulation mode<br>- Hardware:<br>A connection to the control is established.<br>- Simulation static<br>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.<br>- Simulation - counting<br>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.<br>- Simulation - programmed<br>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). |
| **Keep update list in the memory** | Variables which were requested once are still requested from the control even if they are currently not needed.<br>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. |
| **Output can be written** | `Active`: Outputs can be written.<br>`Inactive`: Writing of outputs is prevented.<br>**Note**: Not available for every driver. |
| **Variable image remanent** | This option saves and restores the current value, time stamp and the states of a data point.<br>Fundamental requirement: The variable must have a valid value and time stamp. |

| | |
|---|---|
| | The variable image is saved in mode hardware if:<br>- one of the states S_MERKER_1(0) up to S_MERKER8(7), REVISION(9), AUS(20) or ERSATZWERT(27) is active<br>The variable image is always saved if:<br>- the variable is of the object type **Driver variable**<br>- the driver runs in simulation mode. (not programmed simulation)<br>The following states are not restored at the start of the Runtime:<br>- SELECT(8)<br>- WR-ACK(40)<br>- WR-SUC(41)<br>The mode **`Simulation - programmed`** at the driver start is not a criterion in order to restore the remanent variable image. |
| **Stop on Standby Server** | 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.<br>**Attention:** If this option is active, the gapless archiving is no longer guaranteed.<br>`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.<br>**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. |
| **Global Update time** | `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.<br>`Inactive`: The set priorities are used for the individual variables. |
| **Priority** | The polling times for the individual priority classes are set here. All variables with the according priority are polled in the set time.<br>The allocation to the variables takes place separately in the settings of the variable properties.<br>The communication of the individual variables are graduated in respect of importance or necessary topicality using the priorities. |

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

### 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. |

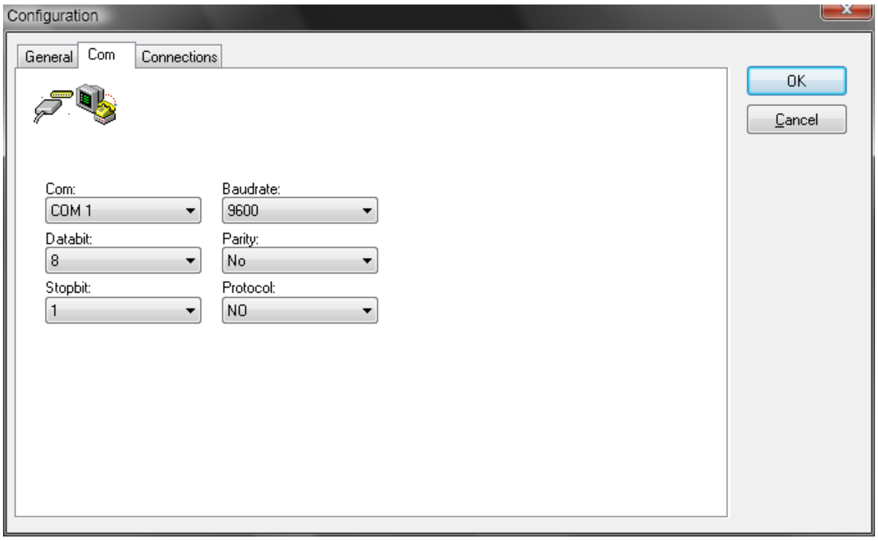
### UPDATE TIME FOR CYCLICAL DRIVERS

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.

## 6.2.2 Com

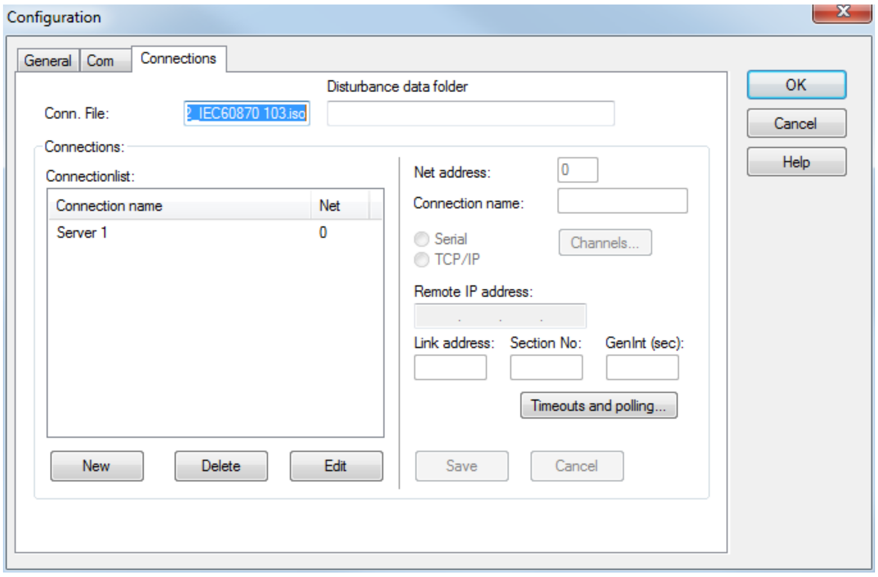
In the serial case, select the appropriate communication settings.



Com: COM 1
Baudrate: 9600
Databit: 8
Parity: No
Stopbit: 1
Protocol: NO

| Parameters | Description |
|---|---|
| **Com** | COM port 1-16 |
| **Baud rate** | Adjust after setting on PLC |
| **Data bits** | 8 |
| **Stop bit** | 1 |
| **Parity** | Adjust after setting on PLC |
| **Protocol** | None |

## 6.2.3 Connections



Conn. File: ?_IEC60870 103.iso

Net address: 0

Connection name:

| Connection name | Net |
|---|---|
| Server 1 | 0 |

| Parameters | Description |
|---|---|
| **Connection file** | Enter the name of the configuration file here. This file is required for the definition of the driver connection. |
| **Error data folder** | Specify the folder on the Runtime computer on which you want to store files with transferred error data. |
| **Connections** | Area for configuration; as many desired configurations for PLCs on different hardware addresses. |
| **Connection list** | Displays the connection names with the corresponding Net addresses. Select the connection name to display its connection parameters. |
| New | Create new connection. Enter connection parameters and close with **Save**. |
| Clear | Delete existing connection. |
| Edit | Edit existing connection. Edit connection parameters and close with **Save**. |
| Net address | Complies with the net address in the variable definition. Can be any number between 0-255. |
| Connection name | Freely definable name of the connection, e.g. the label of the PLC |
| Serial | Active: A serial connection is made. |
| TCP/IP | Active: Connection is made via TCP/IP. |
| Channels | Click on the button to open the dialog for configuring channels (on page 17) for the Comtrade format. |
| Remote IP address | IP-Address of PLC |
| Link address | Link address of the PLC (of the IEC60870-103 slave).<br>Every device is identified by the Net address on the Runtime side and by the Link address on the IEC60870-103 protocol side.<br>**Note:** The **link address** 255 is used as a broadcast channel.<br>If the **link address** of a connection is set to 255, the **channels** and the **section** can no longer be edited. |
| Section Nr. | Section of the PLC (the IEC60870-103 Slave).<br>An IEC60870-103 device can contain multiple sectors. |
| GenInt (sec) | The General Interrogation Interval in seconds.<br>The driver sends out GIs cyclically, according to this setting.<br>It is recommended to configure a time of more than 15 minutes (900s). |
| Time-out and polling... | Clicking on the button opens the dialog to configure the time-out settings (on |

| | |
|---|---|
| | page 18).<br>The dialog is opened for editing in **New** or **Edit** mode, otherwise it is only displayed. |
| Save | Save connection settings. |
| Reject | Leave configuration without saving. |

**Note:** Driver messages can be read with the Diagnosis Viewer (main.chm::/12464.htm).

### CREATE NEW CONNECTION

1. click on the button **New**
2. Enter the connection details.
3. Click on **Save**

### EDIT CONNECTION

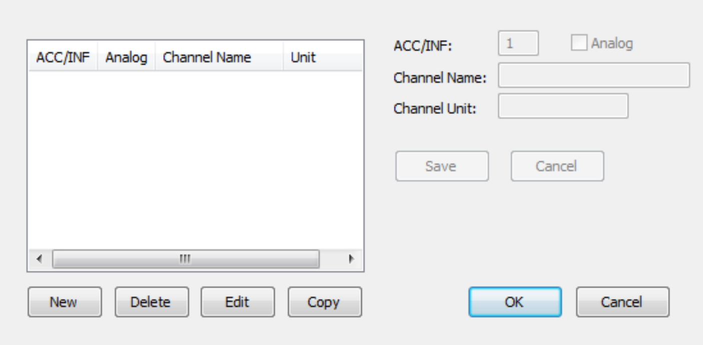
1. select the connection in the connection list
2. click on the button **Edit**
3. change the connection parameters
4. finish with **Save**

### DELETE CONNECTION

1. select the connection in the connection list
2. click on the button **Delete**
3. the connection will be removed from the list

### Configuring the channels

The IEC870_10332 driver writes Disturbance Data as Comtrade files in accordance with the standard of 1999. In order for zenon to process this information, additional information on the Disturbance channels is necessary. You can define the channels in the following dialog.



| Parameters | Description |
|---|---|
| **List field (left -hand side)** | List of all already defined channels. |
| **New** | Activates the configuration for a new channel. |
| **Delete** | Deletes selected channel from the list. |
| **Edit** | Opens the configuration for the selected channel. |
| **Copy** | Copies entries of another hardware address. |
| **Configuration (right-hand side)** | |
| **ACC/INF** | Enter the channel number for identifying the associated Comtrade channel.<br>ACC: analog<br>INF: digital |
| **Analog** | Activate this check box to specify the channel as analog. Otherwise, it will be digital. |
| **Channel Name** | Enter the name of the channel here.<br>The name must not contain a comma. |
| **Unit** | Only available if property **Analog** is active. Enter a unit for the channel here. |
| **Save** | Saves the settings for the channel. |
| **Cancel** | Discards unsaved changes and cancels processing of the channel. |

After finishing, click the button OK to close the dialog and confirm the modifications you made. Click on Cancel to close the dialog and discard all changes you made.

### DATA STORAGE

- The driver saves the disturbance data in a **`comtrade .cfg`** file and a **`comtrade .dat`** file.
- The name of the channel for analog and digital channels is appointed to **`ch_id`**. The unit for analog channels is appointed to **`uu`**.
- The channel information are stored for each hardware address in the driver connection file (xxx matches the respective hardware address):

  [FETCH_HWxxx]

  CH_COUNT = (Number of channels)
- For each channel there is a section (ccc is the consecutive number of the channels 0-(CH_COUNT-1)):

  [HWxxxCHANNELccc]

  CH_ANALOG = (1 analog, 0 digital)

  CH_NO = (ACC/information number)

  CH_NAME = (Name of the channel)

  CH_UNIT = (Unit of the channel)
- If the driver receives a channel that wa not defined in the list, the fields in **`Comtrade.cfg`** remain empty.

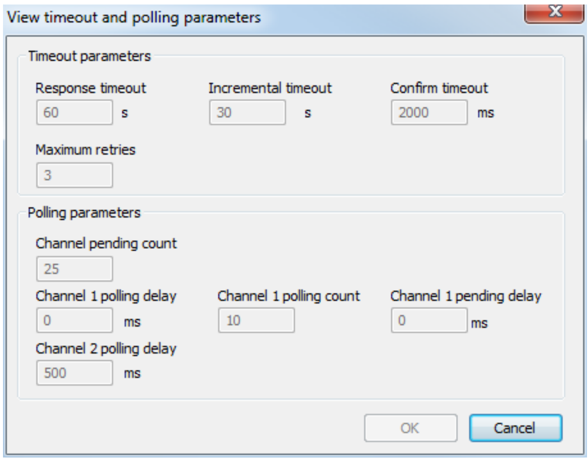
## Time-out and polling

The time-out settings are defined in this dialog. Note that some time-outs are configured in seconds and others in milliseconds.

## Attention

***Response timeout** can be changed individually at any time.*

*All other settings for time-out and polling should be left at the given default values. Changes to these settings are only to be made by expert users and can lead to communication being configured incorrectly and thus unwanted behavior.*



View timeout and polling parameters

Timeout parameters

Response timeout: 60 s

Incremental timeout: 30 s

Confirm timeout: 2000 ms

Maximum retries: 3

Polling parameters

Channel pending count: 25

Channel 1 polling delay: 0 ms

Channel 1 polling count: 10

Channel 1 pending delay: 0 ms

Channel 2 polling delay: 500 ms

| Parameters | Description |
|---|---|
| Timeout parameters | |
| **Response timeout** | General response timeout in seconds.<br>Default: 60 s |
| **Incremental timeout** | Maximum time in seconds that a device waits for a response from another device in the event of a command.<br>Default: 30 s |
| **Confirm timeout** | Maximum time spent waiting for a confirmation.<br>Default: 2000 ms |
| **Maximum retries** | Maximum number of connection attempts:<br>Default: 3 |
| Polling parameters | Polling settings |
| **Channel pending count** | Maximum number of consecutive queries to a device. Has no effect if only one channel was configured.<br>Default: 25 |
| **Channel 1 polling delay** | Polling delay for channel 1 in milliseconds.<br>Default: 0 ms |
| **Channel 1 polling count** | Number of polling attempts for channel 1.<br>Default: 10 |
| **Channel 1 pending delay** | Delay for channel 1 in milliseconds.<br>Default: 0 ms |
| **Channel 2 polling delay** | Polling delay for channel 2 in milliseconds.<br>Default: 500 ms |
| OK | Applies changes and closes dialog. |
| Cancel | Discards changes and closes dialog. |

# 7. Creating variables

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

## 7.1 Creating variables in the Editor

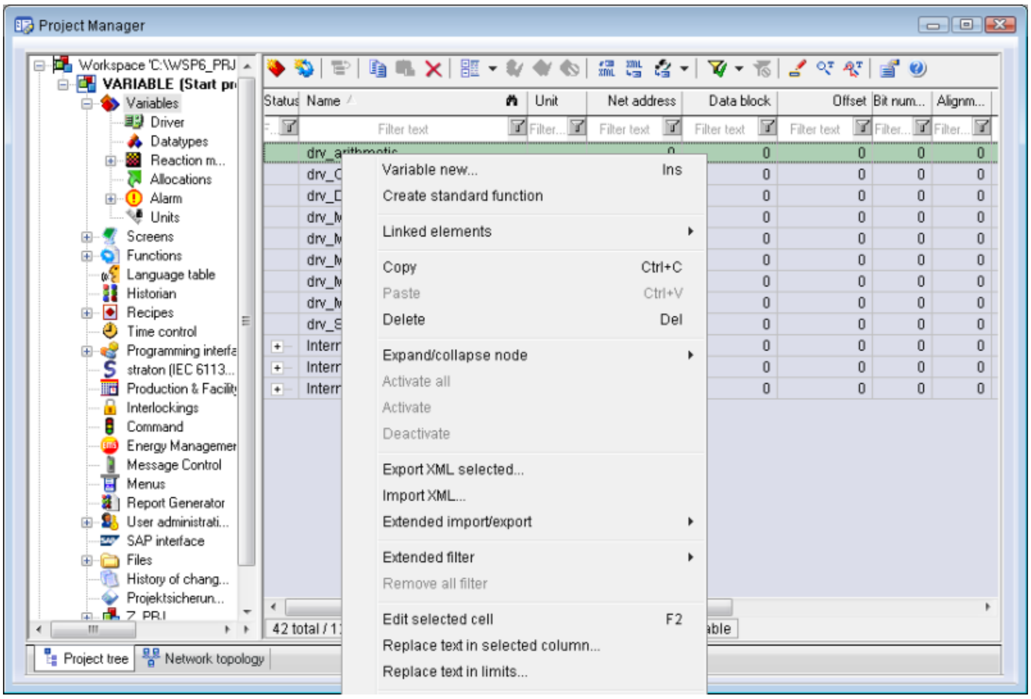
Variables can be created:

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

### VARIABLE DIALOG

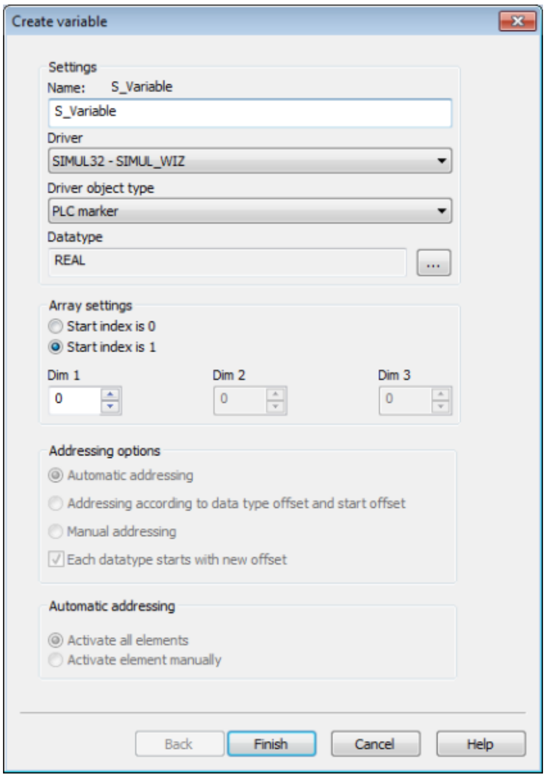
To create a new variable, regardless of which type:

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



Create variable

Settings

Name: S_Variable

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

| 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.<br>Maximum length: 128 Zeichen<br>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.<br>Note: For some drivers, the addressing is possible over the property **Symbolic address**, as well. |
| **Driver** | Select the desired driver from the drop-down list.<br>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. |

### 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

The driver's communication is address-based.

The variables are allocated via:

the Net address - entered in Driver configuration (on page 14) for connection,

the Function Type and the Information Number - corresponds to FUN and INF from the standard

and for Measurands also to the index - corresponds to the sequence of MEA in ASDU <3> or <9>.

The variable of the time synchronization command (driver object type TIME SYNC) has the Function Type 0 and the Information Number 0; the same applies to the variable with the connection state (driver object type CONNECTION STATE).

**Example**

Measurand P of Measurands II (ASDU <9> in monitor direction)

Data type: REAL

Driver object type: MONITORING

Function Type: 128

Information Number: 148

Index: 6

## 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 TYPES AND SUPPORTED IEC DATA TYPES FOR PROCESS VARIABLES IN ZENON

| Driver object type | Channel type | Read / Write | Supported data types | Comment |
|---|---|---|---|---|
| COMMAND | 8 | W | BOOL, INT, UINT, SINT, USINT | For 'General command' - ASDU<20> in control direction<br>In the standard corresponds to DCO (Double command); recommendation: SINT, USINT<br>If sending fails, the variable will get a PN bit. |
| TIME SYNC | 67 | W | BOOL | For time synchronization command - ASDU<6 > in control direction<br>A set value 1 sent to this variable (Function Type 0 and Information Number 0) will be sent to the PLC as a time synchronization.<br>If sending fails, the variable will get a PN bit. |
| MONITORING | 9 | R | BOOL, REAL, INT, UINT, SINT, USINT | In the standard corresponds to DCO (Double-point information); recommendation: SINT, USINT<br>or MVAL (F13); recommendation: REAL<br>or SCL (R32.23); recommendation: REAL |
| MONITORING<br>TRANSIENT | 65 | R* | BOOL, INT, UINT, SINT, USINT | For indicators not contained in GI.<br>In the standard corresponds to DCO (Double-point information); recommendation: SINT, USINT |

| | | | | |
|---|---|---|---|---|
| | | | | *) The received value is immediately set to 0 by the driver. |
| CONNECTION STATE | 66 | R | BOOL | the connection state (Function Type 0 and Information Number 0)<br>If the value of this variable is 1, this means that a connection is active (at least on the data link level). |
| DISTURBANCE DATA | 64 | R / W | STRING | To be able to fetch Disturbance Data files, you have to configure two variables with the same Function Type.<br>The variable with Information Number 0 is the command and status variable, Information Number 1 is the directory variable.<br>The index of these variables is ignored. |
| Driver variable | 35 | R / W | BOOL, SINT, USINT, INT, UINT, DINT, UDINT, REAL, STRING | Variables for the statistical analysis of communication.<br>Find out more in the chapter about the Driver variables (on page 34) |
| Generic Data | 69 | R / W | STRING | |
| Generic Data Trigger | 68 | R / W | INT, UINT | |

| | |
|---|---|
| Channel type | The "Kanaltyp" property is the internal numerical name of the driver object type. It is also used for the extended DBF import/export of the variables. |

Generic Data Triggered and Generic Data

For each generic data variable, there must be a corresponding generic data trigger variable.

The addresses of the variables are mapped as follows:

- **Function Type**: `kod` (normally 1)
- **Information Number**: `0` is not used
- **Index**: `gin`

The value of the trigger variable that is written decides the action to be carried out:

- `244`: Get Generic Data – the variable with `gin/kod` is requested
- `245`: Generic Data GI – a general request for generic data is sent
- `248`: Write Generic Data – the variable with `gin/kod` is written
- `250`: Write and Execute – the variable with `gin/kod` is written and an **`execute`** is triggered

The data of the generic data variables is output as a hex dump in a string.

## 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.

| Control | zenon | Data type |
|---|---|---|
| BOOL | BOOL | 8 |
| USINT | USINT | 9 |
| SINT | SINT | 10 |
| UINT | UINT | 2 |
| INT | INT | 1 |
| - | UDINT | 4 |
| - | DINT | 3 |
| - | ULINT | 27 |
| - | LINT | 26 |
| REAL | REAL | 5 |
| - | LREAL | 6 |
| STRING | STRING | 12 |
| - | WSTRING | 21 |
| - | DATE | 18 |
| - | TIME | 17 |
| - | DATE_AND_TIME | 20 |
| - | TOD (Time of Day) | 19 |

**Data type:** The property **Data type** is the internal numerical name of the data type. It is also used for the extended DBF import/export of the variables.

## 7.4 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.*

### 7.4.1 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.*

### 7.4.2 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 there 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

| Description | Type | Field size | Comment |
|---|---|---|---|
| KANALNAME | Char | 128 | Variable name.<br>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).<br>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.<br>The length can be limited using the MAX_LAENGE entry in **project.ini**. |
| EINHEIT | 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) |
| ADDRESS | N | 5 | Offset |
| BITADR | N | 2 | For bit variables: bit address<br>For byte variables: 0=lower, 8=higher byte<br>For string variables: Length of string (max. 63 characters) |
| ARRAYSIZE | N | 16 | Number of variables in the array for index variables<br>ATTENTION: Only the first variable is fully available. All others are only available for VBA or the Recipe Group Manager |
| LES_SCHR | R | 1 | Write-Read-Authorization |

| | | | |
|---|---|---|---|
| | | | 0: Not allowed to set value.<br>1: Allowed to set value. |
| MIT_ZEIT | R | 1 | time stamp in zenon zenon (only if supported by the driver) |
| OBJEKT | N | 2 | Driver-specific ID number of the primitive object<br>comprises TREIBER-OBJEKTTYP 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)<br>not used for all other variables |
| MEMTIEFE | 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 | R | 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 | R | 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.<br>Free string for export and display in lists.<br>The length can be limited using the MAX_LAENGE entry in **project.ini**. |
| ADJWVBA | R | 1 | Non-linear value adaption:<br>0: Non-linear value adaption is used<br>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 DEFINITION

Limit definition for limit values `1` to `4`, and status `1` bis `4`:

| Description | Type | Field size | Comment |
|---|---|---|---|
| AKTIV1 | R | 1 | Limit value active (per limit value available) |
| GRENZWERT1 | F | 20 | technical value or ID number of a linked variable for a dynamic limit (see VARIABLEx)<br>(if VARIABLEx is 1 and here it is −1, the existing variable linkage is not overwritten) |
| SCHWWERT1 | F | 16 | Threshold value for limit |
| HYSTERESE1 | F | 14 | Is not used |
| BLINKEN1 | R | 1 | Set blink attribute |
| BTB1 | R | 1 | Logging in CEL |
| ALARM1 | R | 1 | Alarm |
| DRUCKEN1 | R | 1 | Printer output (for CEL or Alarm) |
| QUITTIER1 | R | 1 | Must be acknowledged |
| LOESCHE1 | R | 1 | Must be deleted |
| VARIABLE1 | R | 1 | Dyn. limit value linking<br>the limit is defined by an absolute value (see field GRENZWERTx). |
| FUNC1 | R | 1 | Functions linking |
| ASK_FUNC1 | R | 1 | Execution via Alarm Message List |
| FUNC_NR1 | N | 10 | ID number of the linked function<br>(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 text |
| A_DELAY1 | N | 10 | Time delay |
| INVISIBLE1 | R | 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.**

## 7.5 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 defined in the driver kit are available in the import file **`drvvar.dbf`** (on the CD in the directory: `CD_Drive:/Predefined/Variables`) 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 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<br>For `driver stop`, the variable has the value `TRUE` and an **OFF** bit. After the driver has started, the variable has the value `FALSE` 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 TRUE, 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 SrvDrvVarApplyCom 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 SrvDrvVarApplyModem. 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. |
| Baud rate | 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<br>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 |

# 8. Driver-specific functions

The driver supports the following functions:

### CONNECTION STATUS

The current connection status can be requested via a BOOL variable of the driver object type CONNECTION STATE. The Function Type and Information Number of this variable must be 0. If the value of this variable is 1, this means that a connection is active (at least on the data link level).

### GENERAL INTERROGATION (GENERAL INTERROGATION - GI)

The driver sends GI (ASDU <7> in control direction) cyclically, depending on the connection setting **GenInt** in the Driver configuration (on page 14). A GI will also be performed immediately after a connection is reestablished (if it was interrupted).

The driver also accepts received values of variables that are marked as not existent in the GI in the standard, in case the PLC sends such values.

A general query can also be sent as a broadcast. To do this, the link address (on page 14) `255` is used as a broadcast channel. For this, the following applies:

- General interrogation on a connection is only sent if the interval for this is not `0` (**GenInt** (on page 14) setting). Exception: Initial GI; this is always sent.

- General interrogation on a connection with link address 255 is sent as broadcast GI, if the interval is not 0 (**GenInt** (on page 14) setting).

## TIME SYNCHRONIZATION

The driver sends time synchronization command (ASDU <6> in control direction) automatically upon every driver start, i.e. upon starting the zenon Runtime and after stopping/starting the driver (with zenon functions - Driver commands: stop driver and start driver).

A clock synchronization can be re-executed at a later time. In order to do so, a variable of the driver object type TIME SYNC must be written to. The Function Type and Information Number of this variable must be 0. If the variable gets the value 1, the current time of the PC will be sent to the PLC. If the time synchronization could not be executed correctly, this variable will get the PN Bit.

Time synchronization can also be sent as a broadcast. To do this, the link address (on page 14) 255 is used as a broadcast channel. For this, the following applies:

- A TIME SYNC variable must be created for time synchronization.
- The variable must have the same network address as that used for the broadcast link address 255.
- If this variable is set, a broadcast time synchronization is sent.

## COMMANDS (GENERAL COMMAND)

To send General command to the PLC (ASDU <20> in control direction), you have to configure a variable of the driver object type COMMAND (on page 23) with the corresponding Function Type and Information Number. A set value sent to this variable (DCO: 1-off, 2-on) will be sent to the PLC as a General Command. If a command could not be sent successfully, the PN Bit of the variable will be set to 1.

## CAUSE OF TRANSMISSION

The COT in the status of a variable is available and can be evaluated via a Rema, for example.

The response variable gets the COT from the PLC, and the driver forwards it to the control system. In the standard, COT 1 means 'spontaneous', COT 9 means 'general interrogation'.

The COT of the command will be set for the control system by the driver itself. In the direction of the PLC, the COT will be sent in accordance with standard 60870-103, but for the control system, the driver uses the COT in accordance with the standard 61870-101/104 (e.g. COT value 7 COT_actcon), in order to ensure the compatibility with the command input. This way, the set value input with watchdog timer via cause of transmission is supported.

### INVALID STATUS

The driver will set the Invalid Bit after the timeout has passed, if there is an error in the communication (TCP/IP) with the driver.

INVALID will also be set for Measurands values in whose structure (MEA) the Quality descriptor has the value ER (MVAL invalid) activated.

### OV BIT STATUS

OV Bit will also be set for Measurands values, in whose structure (MEA) the Quality descriptor has the value OV (Overflow) activated.

### VARIABLES THAT ARE NOT CONTAINED IN GI

Some PLCs not for all indicators (data points) send a reset of the value (from ON to OFF). Even if not all indicators are in GI by default (e.g. status indicator 'LED reset').

For such variables, we recommend the driver object type MONITORING TRANSIENT (on page 23). The received value is immediately set to 0 by the driver and can be evaluated by creating an alarm or function for limit violation for limit 1 (value OFF counter) and/or for limit 2 (value OFF and ON counter).

If a variable of the driver object type MONITORING does not answer to a general interrogation (GI) (e.g. Time-trigged measurand - ASDU<4>), it will get the value 0 and the driver will also set the SB Bit (Substituted).

### VARIABLES TIMESTAMP

Monitoring variables:

- Time-tagged message (ASDU<1>)
- Time-tagged message with relative time (ASDU<2>)
- Time-tagged measurands with relative time (ASDU<4>)

come from IEC 60870-5-103 PLCs signed with CP32Time2a time stamp. The driver uses this time for the time stamp of the variable and sets the T_EXTERN (Real-time external). All modules now use this time stamp, e.g. Alarm Message List, Historian ...

The counters RET and FAN from ASDUs <2> u.<4> will not be considered by the driver.

### DISTURBANCE DATA REQUEST

The Disturbance Data files are stored in the specified directory (in the driver configuration) in subfolders according to their Net address (i.e. for Net address 1, there will be a subdirectory '1', in which the files will be stored).

To fetch Disturbance Data files for a Function Type or to view the available data, you have to configure two String variables of the driver object type DISTURBANCE DATA (on page 23) and the according Function Type.
The variable with Information Number 0 is the command and status variable, the one with Information Number 1 is the directory variable. The index of these variables is ignored.

If you set the value of the variable for Information Number 0 to 'DIR', the directory information will be updated and shown in the variable with Information Number 1. After executing the operation, the variable with Information Number 0 will get the value '`DIR OK`' or '`DIR ERROR`'.

By sending '`GET [FaultNo]` to the variable with Information Number 0, the corresponding Disturbance Data file is retrieved and stored in the configured directory as Comtrade files called '[FktType]-[FaultNo].cfg' and '[FktType]-[FaultNo].dat' (for example: '128-3.cfg, 128-3.dat'). After executing the operation, the variable with Information Number 0 will get the value '`GET OK`' or '`GET ERROR`'.

### SPECIFIC LOG ENTRIES

- '`Callback Status: Variable: , Status: I870CHNL_RESP_STATUS_TIMEOUT`'

The driver could not establish a connection with the PLC, the connection was interrupted or the PLC did not response within time.

- '`LOG:ConvertTime: Invalid TimeStamp`'

The timestamp received from the PLC does not match the CP32Time2a format.

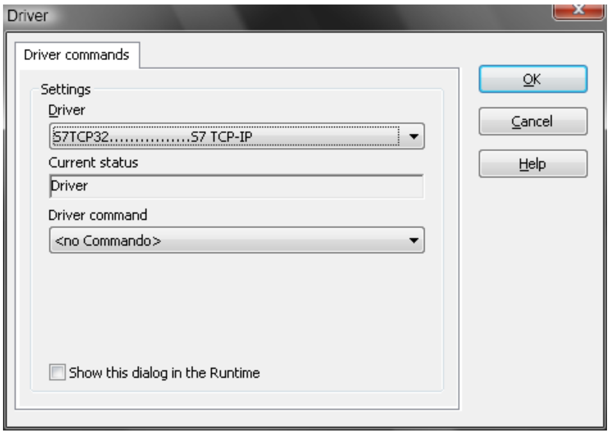
# 9. Driver commands

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



Driver

Driver commands

Settings

Driver: S7TCP32...............S7 TCP-IP

Current status: Driver

Driver command: <no Commando>

□ Show this dialog in the Runtime

| Parameters | Description |
|---|---|
| **Drivers** | Drop-down list with all drivers which are loaded in the project. |
| **Current state** | Fixed entry which has no function in the current version. |
| Driver commands | 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.<br>**Note:** If the driver is in offline mode, all variables that were created for this driver receive the status switched off (OFF; Bit 20). |
| ▸ Driver in simulation mode | Driver is set into simulation mode.<br>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.<br>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. |
| ▸ Activate driver write set value | Write set value to a driver is allowed. |
| ▸ Deactivate driver write set value | Write set value to a driver is prohibited. |
| ▸ Establish connection 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.

# 10. Interoperability List

This companion standard presents sets of parameters and alternatives from which subsets must be selected to implement particular telecontrol systems. Certain parameter values, such as the choice of 'structured' or 'unstructured' fields of the information object address of ASDUs represent mutually exclusive alternatives. This means that only one value of the defined parameters is admitted per system. Other parameters, such as the listed set of different process information in command and in monitor direction allow the specification of the complete set or subsets, as appropriate for given applications. This clause summarizes the parameters of the previous clauses to facilitate a suitable selection for a specific application. If a system is composed of equipment stemming from different manufacturers, it is necessary that all partners agree on the selected parameters.

The interoperability list is defined as in IEC 60870-5-101 and extended with parameters used in this standard. The text descriptions of parameters which are not applicable to this companion standard are strike-through (corresponding check box is marked black).

NOTE In addition, the full specification of a system may require individual selection of certain parameters for certain parts of the system, such as the individual selection of scaling factors for individually addressable measured values.

The selected parameters should be marked in the white boxes as follows:

[   ] Function or ASDU is not used
[ X ] Function or ASDU is used as standardized (default)
[ R ] Function or ASDU is used in reverse mode
[ B ] Function or ASDU is used in standard and reverse mode

The possible selection (blank, X , R , or B ) is specified for each specific clause or parameter.

A black check box indicates that the option cannot be selected in this companion standard.

## 1. PHYSICAL LAYER

### ELECTRICAL INTERFACE

[ X ] EIA RS-485

[ X ] Number of loads 32

### OPTICAL INTERFACE

[ X ] Glass Fiber

[ X ] Plastic Fiber

[ X ] F-SMA type connector

[ X ] BFOC/2,5 type connector

#### TRANSMISSION SPEED

[ X ] 9600 bit/s

[ X ] 12900 bit/s

## 2. LINK LAYER

There are no choices for the Link Layer.

## 3. APPLICATION LAYER

#### TRANSMISSION MODE FOR APPLICATION DATA

Mode 1

#### COMMON ADDRESS OF ASDU

[ X ] One COMMON ADDRESS of ASDU (identical to station address)

[ ] More than one COMMON ADDRESS of ASDU

#### SELECTION OF STANDARD INFORMATION NUMBERS IN MONITOR DIRECTION

| System functions in monitor direction | | |
|---|---|---|
| | **Inf** | **Semantics** |
| [ X ] | <0> | End of general interrogation |
| [ X ] | <0> | Time synchronization |
| [ X ] | <2> | Reset FCB |
| [ X ] | <3> | Reset CU |
| [ X ] | <4> | Start/restart |
| [ X ] | <5> | Power on |

| Status indications in monitor direction | | |
|---|---|---|
| | **Inf** | **Semantics** |
| [ X ] | <16> | Auto-recloser active |
| [ X ] | <17> | Teleprotection active |
| [ X ] | <18> | Protection active |
| [ X ] | <19> | LED reset |
| [ X ] | <20> | Monitor direction blocked |
| [ X ] | <21> | Test mode |
| [ X ] | <22> | Local parameter setting |
| [ X ] | <23> | Characteristic 1 |
| [ X ] | <24> | Characteristic 2 |
| [ X ] | <25> | Characteristic 3 |
| [ X ] | <26> | Characteristic 4 |
| [ X ] | <27> | Auxiliary input 1 |
| [ X ] | <28> | Auxiliary input 2 |
| [ X ] | <29> | Auxiliary input 3 |
| [ X ] | <30> | Auxiliary input 4 |

| Supervision indications in monitor direction | | |
|---|---|---|
| | **Inf** | **Semantics** |
| [ X ] | <32> | Measurand supervision I |
| [ X ] | <33> | Measurand supervision V |
| [ X ] | <35> | Phase sequence supervision |
| [ X ] | <36> | Trip circuit supervision |
| [ X ] | <37> | I>> back-up operation |
| [ X ] | <38> | VT fuse failure |
| [ X ] | <39> | Teleprotection disturbed |
| [ X ] | <46> | Group warning |
| [ X ] | <47> | Group alarm |

| Earth fault indications in monitor direction | | |
|---|---|---|
| | **Inf** | **Semantics** |
| [ X ] | <48> | Earth fault L $_1$ |
| [ X ] | <49> | Earth fault L $_2$ |
| [ X ] | <50> | Earth fault L $_3$ |
| [ X ] | <51> | Earth fault forward, i.e. line |
| [ X ] | <52> | Earth fault reverse, i.e. busbar |

| **Fault indications in monitor direction** | | |
|---|---|---|
| | **Inf** | **Semantics** |
| [ X ] | <64> | Start /pick-up $L_1$ |
| [ X ] | <65> | Start /pick-up $L_2$ |
| [ X ] | <66> | Start /pick-up $L_3$ |
| [ X ] | <67> | Start /pick-up N |
| [ X ] | <68> | General trip |
| [ X ] | <69> | Trip $L_1$ |
| [ X ] | <70> | Trip $L_2$ |
| [ X ] | <71> | Trip $L_3$ |
| [ X ] | <72> | Trip I>> (back-up operation) |
| [ X ] | <73> | Fault location X in ohms |
| [ X ] | <74> | Fault forward/line |
| [ X ] | <75> | Fault reverse/busbar |
| [ X ] | <76> | Teleprotection signal transmitted |
| [ X ] | <77> | Teleprotection signal received |
| [ X ] | <78> | Zone 1 |
| [ X ] | <79> | Zone 2 |
| [ X ] | <80> | Zone 3 |
| [ X ] | <81> | Zone 4 |
| [ X ] | <82> | Zone 5 |
| [ X ] | <83> | Zone 6 |
| [ X ] | <84> | General start/pick-up |
| [ X ] | <85> | Breaker failure |
| [ X ] | <86> | Trip measuring system $L_1$ |
| [ X ] | <87> | Trip measuring system $L_2$ |
| [ X ] | <88> | Trip measuring system $L_3$ |
| [ X ] | <89> | Trip measuring system E |
| [ X ] | <90> | Trip I> |
| [ X ] | <91> | Trip I>> |
| [ X ] | <92> | Trip IN> |
| [ X ] | <93> | Trip IN>> |

| Auto-reclosure indications in monitor direction | | |
|---|---|---|
| | **Inf** | **Semantics** |
| [ X ] | <128> | CB 'on' by AR |
| [ X ] | <129> | CB 'on' by long-time AR |
| [ X ] | <130> | AR blocked |

| Measurands in monitor direction | | |
|---|---|---|
| | **Inf** | **Semantics** |
| [ X ] | <144> | Measurand I |
| [ X ] | <145> | Measurands I, V |
| [ X ] | <146> | Measurands I, V, P, Q |
| [ X ] | <147> | Measurands $I_N$, $V_{EN}$ |
| [ X ] | <148> | Measurands $I_{L1,2,3}$, $V_{L1,2,3}$, P, Q, f |

| Generic functions in monitor direction | | |
|---|---|---|
| | Inf | Semantics |
| [ ] | <240> | Read headings of all defined groups |
| [ ] | <241> | Read values or attributes of all entries of one group |
| [ ] | <243> | Read directory of a single entry |
| [ ] | <244> | Read value or attribute of a single entry |
| [ ] | <245> | End of general interrogation of generic data |
| [ ] | <249> | Write entry with confirmation |
| [ ] | <250> | Write entry with execution |
| [ ] | <251> | Write entry aborted |

### SELECTION OF STANDARD INFORMATION NUMBERS IN CONTROL DIRECTION

| System functions in control direction | | |
|---|---|---|
| | Inf | Semantics |
| [ X ] | <0> | Initiation of general interrogation |
| [ X ] | <0> | Time synchronization |

| General commands in control direction | | |
|---|---|---|
| | Inf | Semantics |
| [ X ] | <16> | Auto-recloser on/off |
| [ X ] | <17> | Teleprotection on/off |
| [ X ] | <18> | Protection on/off |
| [ X ] | <19> | LED reset |
| [ X ] | <23> | Activate characteristic 1 |
| [ X ] | <24> | Activate characteristic 2 |
| [ X ] | <25> | Activate characteristic 3 |
| [ X ] | <26> | Activate characteristic 4 |

| Generic functions in control direction | | |
|---|---|---|
| | **Inf** | **Semantics** |
| [ ] | <240> | Read headings of all defined groups |
| [ ] | <241> | Read values or attributes of all entries of one group |
| [ ] | <243> | Read directory of a single entry |
| [ ] | <244> | Read value or attribute of a single entry |
| [ ] | <245> | General interrogation of generic data |
| [ ] | <248> | Write entry |
| [ ] | <249> | Write entry with confirmation |
| [ ] | <250> | Write entry with execution |
| [ ] | <251> | Write entry abort |

### BASIC APPLICATION FUNCTIONS

| [ X ] | Test mode |
|---|---|
| [ X ] | Blocking of monitor direction |
| [ X ] | Disturbance data |
| [ ] | Generic services |
| [ ] | Private data |

### MISCELLANEOUS

Measurands are transmitted with ASDU 3 as well as with ASDU 9. As defined in 7.2.6.8, the maximum MVAL can either

be 1,2 or 2,4 times the rated value. No different rating shall be used in ASDU 3 and ASDU 9, i.e., for each measurand

there is only one choice.

| Measurand | Max. MVAL | = rated value times |
|---|---|---|
| Current $L_1$ | 1.2 | or 2.4 |
| Current $L_2$ | [ X ] | [ X ] |
| Current $L_3$ | [ X ] | [ X ] |
| Voltage $L_{1\text{-}E}$ | [ X ] | [ X ] |
| Voltage $L_{2\text{-}E}$ | [ X ] | [ X ] |
| Voltage $L_{3\text{-}E}$ | [ X ] | [ X ] |
| Active power P | [ X ] | [ X ] |
| Reactive power Q | [ X ] | [ X ] |
| Frequency f | [ X ] | [ X ] |
| Voltage $L_1$ - $L_2$ | [ X ] | [ X ] |

# 11. Error analysis

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

## 11.1 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 7.20 -> Diagviewer.*

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

`C:\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 currently created entries live
- customize the logging settings
- change the folder in which the log files are saved

**Note:**

1. In Windows CE even errors are not logged per default due to performance reasons.
2. The Diagnosis Viewer displays all entries in UTC (coordinated world time) and not in local time.
3. 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.
4. If you only use `Error logging`, the problem description is in column `Error text`. For other diagnosis level the description is in column `General text`.
5. For communication problems many drivers also log error numbers which the PLC assigns to them. They are displayed in `Error text` and/or `Error code` and/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.
6. 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`.

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

## 11.2 Check list

- TCP/IP:
  • Is the device you are trying to communicate with connected to the power supply?
  • Is the PC and the device connected to the network?
  • Is the TCP/IP protocol installed?
  • Is the IP address in use by another application?
- Serial:
  • Is the COM port in use by another application or are the settings incorrect?
  • Is the device (PLC) you are trying to communicate with connected to the power supply?
  • Is the cable between PLC and PC or IPC connected correctly?
- General:
  • Have you analyzed the Diagnose Viewer entries (which errors have occured)?
  • Are the Driver settings (on page 14) (connection settings) correct, e.g. link address and section no.?
  • Is the Variable addressing (on page 23) correct, e.g. does the Net address match the one in the driver configuration; do the indicators/data points with the entered FUN (Function Type) and INF (Information Number) exist in the PLC?
  • Did you choose a time interval for the General Interrogation (GI) that is long enough (at least 15 minutes recommended)?