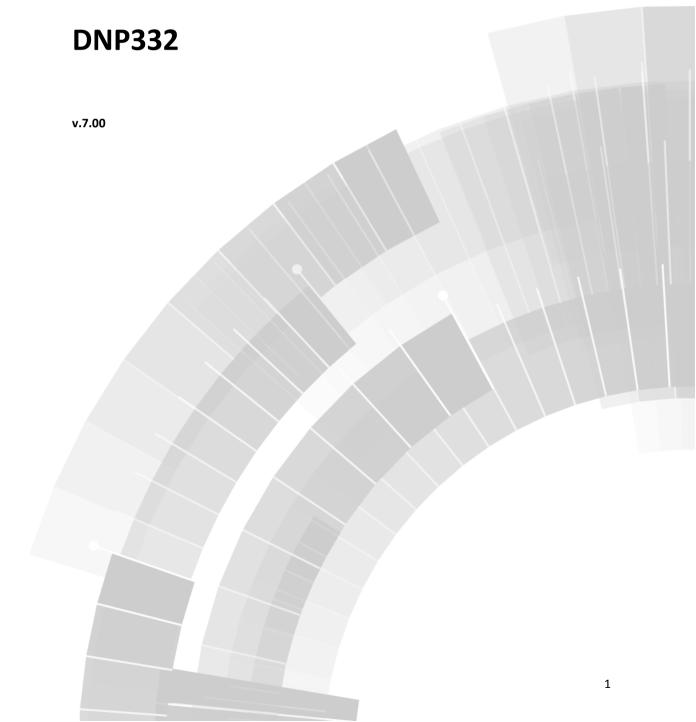


zenon driver manual





© 2012 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. The technical data contained herein has been provided solely for informational purposes and is not legally binding. Subject to change, technical or otherwise.



Contents

| 1. | DNP332 | | | |
|-----|---------------------|----------|--|------------|
| 2. | | | | 5 |
| 3. | DNP332 - Data sheet | | | |
| 4. | Drive | r histor | γ | |
| 5. | Requ | iremen | ts | |
| | 5.1 | PC | | |
| 6. | Confi | guratio | on | |
| | 6.1 | Creatin | ng a driver | |
| | 6.2 | Setting | s in the driver dialog | 10 |
| | | 6.2.1 | General | 11 |
| | | 6.2.2 | Com | 14 |
| | | 6.2.3 | Connections | 16 |
| 7. | Creat | ing vari | iables | 19 |
| | 7.1 | Creatin | ng variables in the Editor | 20 |
| | 7.2 Addressing | | | |
| | 7.3 | Driver o | objects and datatypes | 23 |
| | | 7.3.1 | Driver objects | 23 |
| | | 7.3.2 | Mapping of the data types | 25 |
| | 7.4 | Creatin | ng variables by importing | 26 |
| | | 7.4.1 | XML import of variables from another zenon project | 26 |
| | | 7.4.2 | DBF Import/Export | 26 |
| | 7.5 | Driver \ | variables | 33 |
| 8. | Drive | r-specif | fic functions | 38 |
| 9. | . Driver commands4 | | | 4 1 |
| 10. | Devic | e Profil | le | 42 |
| 11 | Frror | analysi | is. | 43 |



| 11.1 | Analysis tool | . 43 |
|------|---------------|------|
| 11.2 | Check list | . 44 |



1. Welcome to COPA-DATA help

GENERAL HELP

If you miss any information in this help chapter or have any suggestions for additions, please feel free to contact us via e-mail: documentation@copadata.com (mailto:documentation@copadata.com).

PROJECT SUPPORT

If you have concrete questions relating to your project, please feel free to contact the support team via e-mail: support@copadata.com (mailto:support@copadata.com)

LICENSES AND MODULES

If you realize that you need additional licenses or modules, please feel free to contact the sales team via e-mail: sales@copadata.com (mailto:sales@copadata.com)

2. DNP332

TESTED WITH THE FOLLOWING HARDWARE AND SOFTWARE

zenon ProcessGateway with AccessDNP.DLL



TESTING ENVIRONMENT

Tested with simulation driver variables, which serve as DNP3 slave via the zenon Process Gateway. zenon accesses these variables via the driver.

3. DNP332 - Data sheet

| General: | |
|--------------------|--------------------------|
| Driver file name | DNP332.exe |
| Driver description | DNP3 driver |
| PLC types | All DNP3 compatible PLCs |
| PLC manufacturer | DNP3; GE Harris; |

| Driver supports: | |
|---------------------------|------------------------|
| Protocol | DNP3; GE Harris(DNP3); |
| Addressing: address based | х |
| Addressing: name based | - |
| Spontaneous communication | - |
| Polling communication | x |
| Online browsing | - |
| Offline browsing | - |
| Real-time capable | х |



| Blockwrite | - |
|----------------|---|
| Modem capable | - |
| Serial logging | x |
| RDA numerical | - |
| RDA String | - |

| Prerequisites: | |
|----------------|--|
| Hardware PC | Serial interface; Standard LAN Adapter |
| Software PC | - |
| Hardware PLC | - |
| Software PLC | - |
| Requires v-dll | - |

| Platforms: | |
|-------------------|--|
| Operating systems | Windows CE 5.0, CE 6.0; Windows XP, Vista, 7, Server 2003, Server 2008/R2; |
| CE platforms | x86; ARM; Pocket-PC; |



4. Driver history

| Date | Driver version | Change |
|------------|----------------|--|
| 18.06.2008 | 1500 | SBO timeout can be set in the driver configuration |
| 07.07.2008 | 1600 | Created driver documentation |

5. Requirements

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

5.1 PC

HARDWARE

Serial interface, Ethernet TCP/IP.

SOFTWARE

Unless it is already there, copy the driver DNP332.exe to the zenon program directory.

CE

Copy the driver DNP332.dll to the zenon CE program directory.

6. Configuration

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





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:

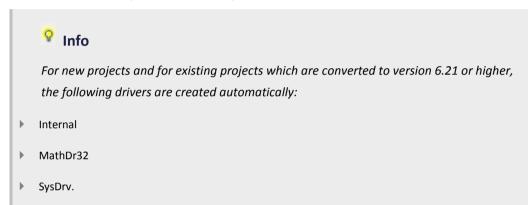
- ▶ Right-click on Driver in the Project Manage and select Driver new in the context menu.
- ▶ In the following dialog the control system offers a list of all available drivers.



- 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, every time 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.
- ► Confirm the dialog with ox. 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.



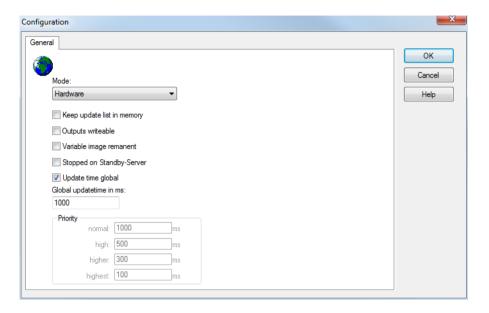
6.2 Settings in the driver dialog

You can change the following settings of the driver:

10



6.2.1 General





| Parameters | Description |
|--------------------------------|---|
| Mode | Allows to switch between hardware mode and simulation mode Hardware: A connection to the control is established. Simulation static 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. Simulation - counting |
| | 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. Simulation - programmed 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 |
| Keep update list in the memory | simulation (main.chm::/25206.htm). 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. |
| Output can be written | Active: Outputs can be written. Inactive: Writing of outputs is prevented. 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. |
|-------------------------------|---|
| | Fundamental requirement: The variable must have a valid value and time stamp. |
| | The variable image is saved in mode hardware if: |
| | one of the states S_MERKER_1(0) up to S_MERKER8(7), REVISION(9), AUS(20) or ERSATZWERT(27) is active |
| | The variable image is always saved if: |
| | ▶ the variable is of the object type Driver variable |
| | the driver runs in simulation mode. (not programmed simulation) |
| | The following states are not restored at the start of the Runtime: |
| | ▶ SELECT(8) |
| | ▶ WR-ACK(40) |
| | ▶ WR-SUC(41) |
| | The mode Simulation - programmed at the driver start is not a criterion in order to restore the remanent variable image. |
| Stop at the 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. |
| | Attention: If this option is active, the gapless archiving is no longer guaranteed. |
| | 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. |
| 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. Inactive: The set priorities are used for the individual variables. |
| Priority | Here you set the polling times for the individual priorities. All variables with the according priority are polled in the set time. The allocation is taken |
| | |



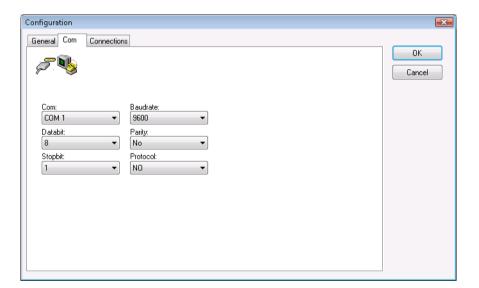
place for each variable separately in the settings of the variable properties. The communication of the individual variables are graduated in respect of importance or necessary topicality using the priorities. Thus the communication load is distributed better.

UPDATE TIME FOR CYCLICAL DRIVER

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





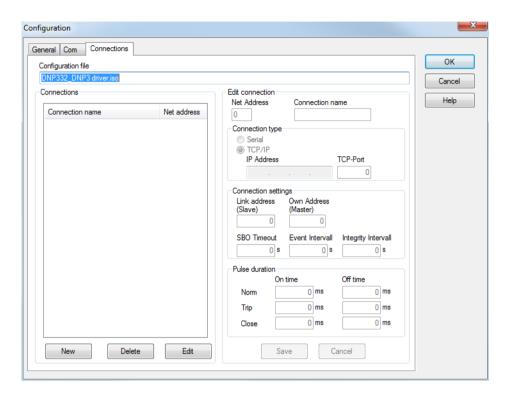
| Parameters | Description |
|------------|---------------------------------------|
| Com | Selection Com port. |
| | Default: 1 |
| Baud rate | Selection baud rate. Adapting to PLC. |
| | Default: 9600 |
| Data bits | Number of data bits. Adapting to PLC. |
| | Default: 8 |
| Stop bit | Selection stop bit. Adapting to PLC. |
| | Default: 1 |
| Parity | Selection parity. Adapting to PLC. |
| | Default: No |
| Protocol | Selection protocol. Adapting to PLC. |
| | Default: No |

Info

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



6.2.3 Connections





| Parameters | Description |
|----------------------|---|
| Configuration file | Name of the file, in which the connection information is stored. For your information only. Cannot be changed. |
| Connections | Connection settings |
| List of connections | Displays the configured connections. |
| New | Creates new connection |
| Delete | Deletes selected connection. |
| Edit | Edits selected connection. |
| Edit connection | Properties for configuration of selected/new connection. |
| Net address | Net address of the connection. |
| | Value between 0 and 255. |
| Connection name | Name of connection. Freely selectable. |
| Serial | Active: serial connection |
| TCP/IP | Active: Connection via TCP/IP |
| IP address | IP address of PLC. |
| TCP-port | Port that is used for communication. |
| | Default: 20000 |
| Connection settings | |
| Link address (slave) | Link address of the PLC (of the DNP3 slave). |
| Own address (master) | Own address (of the DNP3 master driver itself). |
| SBO time-out | Time in seconds which is waited for an answer from the PLC after the Select. If the time expires, the Select is considered as denied. |
| | Default: 30 |
| Event interval | Prompting the driver for new events in seconds. |
| | Default: 1 |
| Integrity interval | Integrity poll of the driver in seconds. |
| | Default: 3600 |
| | I . |



| Pulse duration | Defines pulse duration for Norm, Trip and Close for each connection. Norm, Trip and Close define which relays are switched. |
|----------------|---|
| Norm | Equals NUL. Pulse duration norm: On time: Period of time in milliseconds in which the output is ON Default: 1000 Off time Period of time in milliseconds in which the output is OFF Default: 1000 |
| Trip | Pulse duration trip: On time: Period of time in milliseconds in which the output is ON Default: 1000 Off time Period of time in milliseconds in which the output is OFF Default: 1000 |
| Close | Pulse duration close: On time: Period of time in milliseconds in which the output is ON Default: 1000 Off time Period of time in milliseconds in which the output is OFF Default: 1000 |
| Save | Saves parameters for connection and deactivates editing mode. |
| Reject | Cancels changes and deactivates editing mode without saving. |
| OK | Accept changes in the dialog and close dialog. Only available if no connection is in the "edit" state. |
| Cancel | Discards changes and closes dialog. |
| Help | Opens online help. |





Maximum number of connections: 256 (0-255).

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

7. Creating variables

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



7.1 Creating variables in the Editor

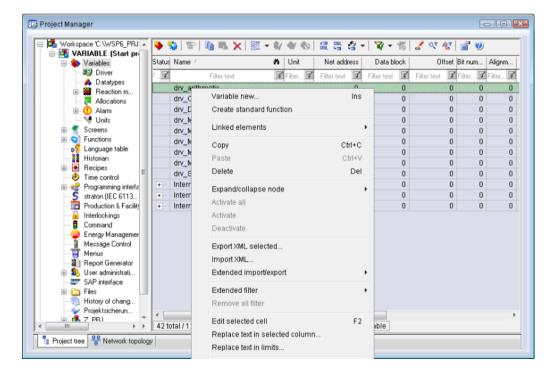
Variables can be created:

- 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



| 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. Attention: The # character is not permitted in variable names. If non-permitted characters are used, creation of variables cannot be completed and the Finish button remains inactive. |
| Drivers | Select the desired driver from the drop-down list. Note: If no driver has been opened in the project, the driver for internal variables (Intern.exe (Main.chm::/Intern.chm::/Intern.htm)) is automatically loaded. |
| Driver object type (cti.chm::/28685.h tm) | 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

VARIABLE ADDRESSING

The variables are allocated to the memory area of the PLC with the help of property Offset.

WINDOWS CE

This driver is available for Windows CE (x86 platforms).



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 | Comment |
|---------------------------|--------------|-----------------|---|--|
| Analog inputs | 68 | R | REAL | |
| Analog Output Statuses | 69 | R/W | UDINT, DINT, REAL, UINT, INT | |
| Binary inputs | 64 | R | BOOL | |
| Binary Inputs Double | 71 | R | USINT | |
| Binary Output Statuses | 65 | R/W | BOOL, USINT | |
| Frozen Counters | 67 | R/W | BOOL, UDINT, DINT, USINT, UINT, INT, SINT | |
| Running Counters | 66 | R/W | BOOL, UDINT, DINT, USINT, UINT, INT, SINT | |
| String Data | 70 | R/W | STRING | |
| Driver variable | 35 | R/W | BOOL, SINT, USINT, INT, UINT, DINT, UDINT, REAL, STRING | Variables for the statistical analysis of communication. Find out more in the chapter about the Driver variables (on page 33) |

COMMANDS

You can use a command input to first send a Select and then an Operate to data points. The SBO timeout can be configured in the driver configuration. A counter starts running if a Select is sent. After the configured SBO timeout has passed, a COT_actterm with a set PN bit will be sent back. If the Select action is cancelled, a COT_deact will be sent back to notify you of this. A COT_deactcon will be sent once the SBO timeout has passed. If an Operate is sent within the timeout, the procedure will stay as usual.



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.

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

| Driver object types | Channel type | Supported datatypes | Read | Write | Comment |
|------------------------|--------------|---------------------|------|-------|--|
| | | (DataType) | | | |
| Binary Input | 64 | BOOL | Υ | N | |
| Binary Output | 65 | BOOL | N | Y | Value 1: LATCH_ON Value 0: LATCH_OFF 1 if writing was successful, 0 if an error occurred |
| Binary Output | 65 | USINT | N | Y | Value 1: PULSE_ON Value 2: PULSE_OFF Value 3: LATCH_ON Value 4: LATCH_OFF 1 for successful writing, 0 for error |
| Running Counter | 66 | BOOL,SINT,INT,DI | Y | N | |
| Frozen Counter | 67 | BOOL,SINT,INT,DI | Y | Y | Writing freezes the counter |
| Analog Input | 68 | REAL | Υ | N | |
| Analog Output | 69 | REAL | N | Y | 1 for successful writing, 0 for error |
| String Data | 70 | STRING | Υ | Υ | |

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.

7.4.1 XML import of variables from another zenon project

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.



IMPORT DBF FILE

To start the import:

- 1. right-click on the variable list
- 2. in the drop-down menu 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.



Info

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 menu 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.



Info

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) when exporting.

DBF files must:

- ▶ correspond to the 8.3 DOS format for filenames (8 alphanumeric characters for name, 3 character suffix, no spaces)
- ▶ Be stored close to the root directory (Root)

DESIGN

| Description | Туре | Field size | Comment |
|-------------|------|------------|---|
| KANALNAME | Char | 128 | Variable name. |
| | | | The length can be limited using the MAX_LAENGE entry in project.ini . |
| KANAL_R | С | 128 | The original name of a variable that is to be replaced by the new name entered under "KANALNAME" (field/column must be entered manually). |
| | | | The length can be limited using the MAX_LAENGE entry in project.ini . |
| KANAL_D | Log | 1 | The variable is deleted with the $1\ \rm entry$ (field/column has to be created by hand). |
| TAGNR | С | 128 | Identification. |
| | | | The length can be limited using the MAX_LAENGE entry in project.ini . |
| EINHEIT | С | 11 | Technical unit |
| DATENART | С | 3 | Data type (e.g. bit, byte, word,) corresponds to the data type. |
| KANALTYP | С | 3 | Memory area in the PLC (e.g. marker area, data area,) corresponds to the driver object type. |
| HWKANAL | Num | 3 | Bus address |
| BAUSTEIN | N | 3 | Datablock address (only for variables from the data area of the PLC) |
| ADRESSE | N | 5 | Offset |



| | T | | | |
|------------|----------|----|--|--|
| BITADR | N | 2 | For bit variables: bit address | |
| | | | For byte variables: 0=lower, 8=higher byte For string variables: Length of string (max. 63 characters) | |
| | | | FOI String variables. Length of string (max. 65 characters) | |
| ARRAYSIZE | N | 16 | Number of variables in the array for index variables | |
| | | | ATTENTION: Only the first variable is fully available. All others | |
| | | | are only available for VBA or the Recipe Group Manager | |
| LES SCHR | R | 1 | Write-Read-Authorization | |
| _ | | | 0: Not allowed to set value. | |
| | | | 1: Allowed to set value. | |
| | _ | | | |
| MIT_ZEIT | R | 1 | time stamp in zenon (only if supported by the driver) | |
| OBJEKT | N | 2 | Driver-specific ID number of the primitive object | |
| | | | comprises KANALTYP and DATENART | |
| CICMIN | Floot | 16 | Non-linearized signal minimum (signal resolution) | |
| 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 | |
| OFDAILKAIL | F | 19 | possible) | |
| | | | not used for all other variables | |
| | | | 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 | С | 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 | С | 128 | Resource label. Free string for export and display in lists. The length can be limited using the MAX_LAENGE entry in project.ini. |
| ADJWVBA | R | 1 | Non-linear value adaption: 0: Non-linear value adaption is used 1: non linear value adaption is not used |
| ADJZENON | С | 128 | Linked VBA macro for reading the variable value for non-linear value adjustment. |
| ADJWVBA | С | 128 | Linked 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 ${\it driver in the DBF file in order for variables to be imported.}$

LIMIT DEFINITION

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



| Description | Туре | 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) (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 | Hysteresis in % |
| 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 the limit is defined by an absolute value (see field GRENZWERTx). |
| FUNC1 | R | 1 | Function linking |
| ASK_FUNC1 | R | 1 | With interrogation before execution |
| FUNC_NR1 | N | 10 | ID number of the linked function (if "-1" is entered here, the existing function is not overwritten during import) |
| A_GRUPPE1 | N | 10 | Alarm/event group |
| A_KLASSE1 | N | 10 | Alarm/event class |
| MIN_MAX1 | С | 3 | Minimum, Maximum |
| FARBE1 | N | 10 | Color as Windows coding |
| GRENZTXT1 | С | 66 | Limit value 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** messages

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.

Hint: 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.



Info

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 | Туре | 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 stopped |
| LineStateUnknow | BOOL | 24.15 | Connection status unknown |
| ModemStatus | UDINT | 24 | Current modem status |
| TreiberStop | BOOL | 28 | Driver stopped |



| | | | 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 | Туре | 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 | SINT | 14 | Number of bits per character of the serial interface |
| | | | Value = 0 if the driver cannot establish any serial connection. |
| StopBit | SINT | 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 | UDINT | 54 | Time in seconds (s) to establish a connection. |



STATISTICS

| Name from import | Туре | 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 нідь 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 MESSAGES

| Name from import | Туре | 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 | UDINT | 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 | SINT | 25 | Error message as number |
| DrvErrorMsg | STRING | 30 | Error message as text |
| ErrorFile | STRING | 15 | Name of error log file |

8. Driver-specific functions

This driver supports the following functions:

COMMANDS WITH SELECT BEFORE OPERATE (SBO)

With the help of command input, you can first send a Select and then an Operate to variables.



As the control (Remsdaq) does not support Cancel or does not have a Select time-out, you must set this at the driver. Therefore the smo time-out can be configured in the driver configuration (on page 16).

The driver behaves as follows:

- ▶ If a Select is written, it is acknowledged and a timer starts to run.
- ► After the set SBO Timeouts has passed, a COT_actterm with a set N_CONF (PN-Bit) will be sent back.
- ▶ If the Select action is cancelled, a COT_deact will be sent back to notify you of this. After the smo_time-out has passed, a COT_deactcon will be sent.
- ▶ If an Operate is sent within the timeout, the procedure will stay as usual.

ERROR FILE

The driver supports central logging in the Log Server.

EXTENDED ERROR FILE

The driver does not support extended logging.

SERIAL LOGGING

To activate this option, you must add section [RS232LOG] to the project.ini:

[RS232LOG]

LOGCOMx=y



| Parameters | Description | |
|------------|-----------------------------------|--|
| У | Value 0 or 1: | |
| | 0 = turns log off | |
| | 1 = turns log on | |
| х | Number of the selected interface. | |

EXAMPLE:

[RS232LOG]

LOGCOM1=1

Activates logging for COM1.



We recommend activating logging only for a short time, when problems occur. Active logging needs considerable computer performance. Additionally the log file needs a lot of memory within short time.

| Parameters | Description | |
|--------------------|--|--|
| Blockwrite | Not supported. | |
| Redundancy | Yes | |
| RDA | Not supported. | |
| Real time stamping | Yes | |
| Browsing | Not supported. | |
| Polling | The driver requests the variables cyclically every 60 minutes. | |
| Spontaneous | Yes, only value changes are transferred. | |
| | Exception: Cyclic total update every 60 minutes. | |
| Number of PLCs | One driver can connect to any number of PLCs. | |

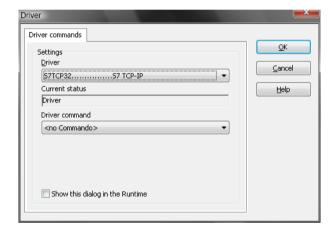


9. Driver commands

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 <code>Driver</code> commands. To do this:

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



| 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. | |
| <pre>Stop driver (offline mode)</pre> | 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 switched off (OFF; Bit 20). | |
| ▶ Driver in simulation | Driver is set into simulation mode. The values of all variables of the driver are simulated by the | |



| | mode | driver. No values from the connected hardware (e.g. PLC, bus system,) are displayed. |
|---|-----------------------------------|--|
| • | Driver in hardware mode | Driver is set into hardware mode. For the variables of the driver the values from the connected hardware (e.g. PLC, bus system,) are displayed. |
| • | Driver-specific command | Enter driver-specific commands. Opens input field in order to enter a command. |
| • | 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) |
| | ow this dialog in the ntime | 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. Device Profile

The device profile for the driver is installed by the setup. You can find the file $\mathtt{DNP332.xml}$ in the following folder:



- 32 bit systems: %ProgramFiles%\COPA-DATA\zenon [Version]\Templates\Dnp3\Driver\
- 64 bit systems: %ProgramFiles(x86)%\COPA-DATA\zenon
 [Version]\Templates\Dnp3\Driver\

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 7.00 -> Diagviewer*.

zenon driver log all errors in the log files. The default folder for the log files is subfolder <u>log</u> in directory ProgramData, example: C:\ProgramData\zenon \zenon700\LOG for zenon version 7.00 SPO. Log files are text files with a special structure.

Attention: With the default settings, a driver only logs error information. With the <code>piagnosis</code> 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

Hints:

- 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**

- Is the configuration of the PLC and the connection correct?
- Is the IP address of the PLC configured correctly (with connection via TCP/IP)?
- Did you analyze the error file (up to zenon version 6.01) or the central log file (version zenon6.20 and higher)?