

# zenon driver manual Beckh\_32

v.7.00



© 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.	Welco	Welcome to COPA-DATA help5			
2.	Beckł	eckh_325			
3.	BECK	H_32 - D	ata sheet	5	
4.	Drive	r history	/	7	
5.	Requ	irement	S	7	
	5.1	PC		8	
	5.2	PLC		8	
6.	Confi	guration	1	8	
	6.1	Creating	g a driver	8	
	6.2	Settings	in the driver dialog	10	
		6.2.1	General	10	
		6.2.2	Driver dialog configuration file	13	
		6.2.3	Driver dialog Beckhoff settings	14	
		6.2.4	Driver dialog Browse	16	
7.	Creat	ing varia	ables	18	
	7.1	Creating	yariables in the Editor	18	
	7.2	Address	ing	21	
	7.3	Driver o	bjects and datatypes	22	
		7.3.1	Driver objects	22	
		7.3.2	Mapping of the data types	23	
	7.4	Creating	y variables by importing	24	
		7.4.1	XML import of variables from another zenon project	24	
		7.4.2	DBF Import/Export	25	
	7.5	Driver v	ariables	31	
8.	Drive	r-specifi	c functions	36	
9.	Drive	r comma	ands	37	



10. Error	analysis	. 39
10.1	Analysis tool	39
10.2	Check list	40



# 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. Beckh\_32

# 3. BECKH\_32 - Data sheet



General:	
Driver file name	BECKH_32.exe
Driver description	Beckhoff TwinCat ADS
PLC types	TwinCat SoftPLC
PLC manufacturer	Beckhoff;

Driver supports:	
Protocol	ADS;
Addressing: address based	x
Addressing: name based	x
Spontaneous communication	-
Polling communication	x
Online browsing	x
Offline browsing	x
Real-time capable	-
Blockwrite	-
Modem capable	-
Serial logging	-
RDA numerical	x
RDA String	-



Prerequisites:	
Hardware PC	-
Software PC	The software PLC TwinCat 2.6 or higher has to be installed on the same computer.
Hardware PLC	-
Software PLC	-
Requires v-dll	-

Platforms:	
Operating systems	Windows XP, Vista, 7, Server 2003, Server 2008/R2;
CE platforms	-;

# 4. Driver history

Date	Driver version	Change
07.07.08	1500	Created driver documentation

# 5. Requirements

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



### 5.1 PC

TwinCAT

Copy the driver file Beckh\_32.EXE to the current zenon directory (unless it is already there).

### 5.2 PLC

# 6. Configuration

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

# 💡 Info

*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.



	lesys Arti NG SoftSPS Treiber	-
	esys Arti NG Sorce's Treber lesys Arti SoftSPS Treiber	E
	lesys SoftSPS Treiber	
	599 50 (SF5 Heldel	
AEG		
🗄 📄 Allen Bra	adlev	
Alstom		
🗄 🧰 Apex		
🗄 🦲 Applicon	n	
🗄 📄 Archive		
🔄 🧰 Arcnet		
🗈 🚞 Areva		
🗄 📄 Asfinag		
🖲 📄 Bachma	กก	-
n Ca Pacest		
[reiberbezeichnu	ng	
Codesys Arti NG	Treiber	
Treiberinformatio	nen	
Beschreibung:		
	25 Treiber "Neue Generation" für die Codesys 35-ARTI (Asynchron Run	
	ttstelle. Der Treiber unterstützt die direkte Variablenübernahme aus ( lungsumgebung und ersetzt den Codesys Arti Soft SPS Treiber.	Jer =
CODESYS ENWICH	angsangebang and ersetzt den codesys Ard Sort SPS freiber.	
	5 Typen geeignet:	
	PSen, Moeller XControl SPSen XC200 und XC600, sowie Elau PacDrive AX 4, C200, C400, C600, P600.	
scederungen ma	n 1, C200, C100, C000, P000.	
Eelaanda Kanak	ingsarten werden unterstützt:	
Ethernet; Local		

► 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.



# 💡 Info

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

Configuration		×
General		
		ок
Mode:		Cancel
Hardware	•	Help
Keep update list	in memory	
Outputs writeable	•	
Variable image re	emanent	
Stopped on Stan	dby-Server	
Update time glob	al	
Global updatetime in	ms:	
1000		
Priority		
normal:	1000 ms	
high:	500 ms	
higher:	300 ms	
highest:	100 ms	



Parameters	Description
Mode	<ul> <li>Allows to switch between hardware mode and simulation mode</li> <li>Hardware: <ul> <li>A connection to the control is established.</li> </ul> </li> <li>Simulation static</li> <li>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.</li> <li>Simulation - counting</li> <li>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.</li> <li>Simulation - programmed</li> <li>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</li> </ul>
	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. 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:
	<ul> <li>one of the states S_MERKER_1(0) up to S_MERKER8(7), REVISION(9), AUS(20) or ERSATZWERT(27) is active</li> </ul>
	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.
	<b>Attention:</b> 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 Driver dialog configuration file

Here the name and the path of the configuration file (\*.bec) must be entered. This file is needed for the driver communication. (It will be created automatically if it does not exist yet)

If you use several drivers in the project, you must specify a separate configuration file for every driver. There will be no check whether there is already a default.bec file from another driver.

The configuration file contains additional information about the variables that cannot be saved in zenon. Via the net address and offset of the zenon variables, the driver looks up the additional information in the configuration file, like the variable name, AMS-NETID, port number etc. As the variables are accessed in the file via the offset and net address of zenon, you can rename the variables in zenon. You can use a text editor to adjust the AMS-NETID of all variables with Search & Replace, for example.

You should not delete any entries in this file, as this may lead to offset movements when browsing & adding. Therefore, it may happen that zenon requests variable A with offset 3, but in the configuration file, variable B is listed at position 3.



Configuration
General
Configuration file
Beckhoff-Settings
Browse
OK
Config file:
Deckhtc32\_BeckhTc32.dat

Newly added variables receive a new offset number in the configuration file.

# 6.2.3 Driver dialog Beckhoff settings



Parameters	Description
Timeout	Here the timeout in milliseconds must be entered. If there is an error during the communication with the TwinCAT PLC (e.g. the PLC does not respond), the communication will be interrupted after that time and the driver status will be set to invalid.
Selected runtime system	Here the currently selected runtime system is displayed. The selected runtime system is needed for the next dialog "Browse". From this runtime system, the variables are read that can then be transferred to zenon (see 3.4 "Browse").
Runtime systems	<ul> <li>This list displays all defined runtime systems (port and NetID). With a doubleclick on a defined Run-Time System it is set as active Run-Time System and entered in the Box "Selected Run-Time System". Activating is also possible with the button "Apply" (after having selected the desired Run-Time System).</li> <li>A new Run-Time system can be added with pressing the button "New" and entering the necessary data in the following dialog.</li> </ul>

A defined RTS can be edited with selecting the desired RTS and then pressing the button "Edit"; then edit the data in the following dialog.

For a BC9000, the standard port is 800, not 801!

#### **INI FILE**

Here the path and the name of the INI files can be entered, where the defined Run-Time Systems are saved. (It will be created automatically if it does not exist yet)

# DIRECT ACCESS VIA THE VARIABLE NAME (ONLY ON LOCAL COMPUTER): FROM VERSION 5.21

If you activate this option, the driver will access the variable via the variable name. This can only be done with variables that are defined on the same computer as the TwinCAT PLC.

Nevertheless, the variables must be taken over in the menu 'Browse', because the allocations are defined there.



eneral Configuration file Beckhoff-Settings   Browse     Imeout: 60000     ms   Selected Runtime system:    Port: 80000     Runtime system::     Port   New   Edit   Delete   INI-File:   beckh_32_BECKH_32.txt     Image: Configuration file     Beckhoff-Settings     OK     Image: Configuration file     Browse     Image: Configuration file     Image: Configuration file     Beckhoff-Settings     Image: Configuration file     Image: Configuration file </th <th>figuration eneral Configurati</th> <th>on file Beckhoff-Settings</th> <th>Browse</th> <th>_</th> <th></th>	figuration eneral Configurati	on file Beckhoff-Settings	Browse	_	
Timeout: 60000   ms   Selected Runtime system:   Port:   801 \ NetID   Apply     New   Edit   Delete   INI-File:   beckh_32_BECKH_32.txt			Diowse		
Port: 801 \ NetID: 0.0.0.0.0 Runtime systems: Port NetID New Edit Delete INI-File: beckh_32_BECKH_32 txt	Timeout: 6000	)0 ms			Carter
Runtime systems:       Apply         Port       NetD         New       Edit         Edit       Delete         INI-File:       beckh_32_BECKH_32.txt         direct access via variable name	Selected Runtim	ie system:			
Port       Net/D         New       Edit         Edit       Delete         INI-File:       beckh_32_BECKH_32.txt         direct access via variable name	Port: 801 \ Net	D: 0.0.0.0.0.0			
New       Edit       Delete	Runtime system:	x			
Edit Delete INI-File: beckh_32_BECKH_32.txt	Port	NetID		Apply	
Edit Delete INI-File: beckh_32_BECKH_32.txt					
INI-File: beckh_32_BECKH_32.txt				New	
INI-File: beckh_32_BECKH_32.txt				E dit	
INI-File: beckh_32_BECKH_32.txt					
direct access via variable name				Delete	
direct access via variable name	INI-File: bec	kh 32 BECKH 32.txt			
Infect access via variable name (only on local Computer)					
	only on loca	; via variable name   Computer)			

Due to compatibility reasons with older projects, this option can be selected.

### 6.2.4 Driver dialog Browse

#### PLC VARIABLES

In this list all variables defined in the soft-PLC are displayed.

#### BROWSE

With pressing this button all variables are downloaded from the TwinCAT server and displayed in the window "PLC-Variables". This requires that the symbols entries are created, as shown in the screenshot from TwinCat PLC control.

#### **BROWSING STRUCTURE VARIABLES**

The following settings are required, to enable the support of structure variable browsing by TwinCAT.

Under project -> options

Select the entry TwinCAT in the list on the left side of this dialog window. For the browsing of structure variables, the option box "Symbol download " is of importance. By default, the setting "Dynamic



symbols" is used. Set it to "Static symbols" and activate the check box "Global main" and "Global sub", as you can see in the screenshot.

#### ADD

The selected symbols in the list "PLC-Variables" are displayed in the list "Added PLC-Variables". Already existing entries are updated, if the symbols in the PLC have been changed since the last transfer.

On adding the added symbols get an internal identification number, that is used as an address for creating the zenon variables.

If symbols are deleted in the lower list and newly added then, they get a new identification number. zenon variables already created with the original number have to be edited then!

#### ADDED PLC-VARIABLES

In this list all variables are displayed, that were transferred from the list "PLC variables".

In order to add a variable, click on it in the list "PLC-Varaible" and press the button "Add" then. Now the variable is transferred to the lower list. These variables can be added to the zenon variables with the button "Select" lateron.

#### **USER DEFINED**

In order to define a variable by hand, click on the button "User-defined" and enter the variable data in the following dialog.

#### PROPERTIES

Single variable data can be edited with the button "Properties".



Delete	To delete a variable simply click on the button "Delete" (after having selected the variable).
Save	Before leaving the dialog the BEC file has to be saved. To do this simply press the button "Save" and the BEC file is saved.

If the dialog is closed without saving, all changes are lost. So do not forget to save the changes!

Configuration				×
General Configuration file Beckhoff-Se	ettings Browse			
PLC-Variable:				ОК
Variable name	Data type Length Index	Offset Comment		<u>C</u> ancel
Search:	(Variable nam	ie)	Browse	
			Add	
Applied PLC-Variable:		l		
Variable name Lengt	h Data type Index O	ifset NetID Port Co	on Save	
			Properties	
			User defined	
			Delete	

# 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

Workspace 'C:\WSP6_PRJ A	•	🌢 🖹	🖻 🖷 🗙   🎛 🗸	* * 6	XML 4a 4a	🏹 🕶 📆	🏅 🔍 👫 👔	10	
🖃 🌗 Variables	Status	Name 🛆		nnit Unit	Net address	Data block	Offset Bi	t num	Alignm
Driver			Filter text	Filter	Filter text 🗹 F	ilter text 📘	Filter text 📝 Fi	lter 🖬 F	Filter 🖬
A Datatypes     Beaction m		drv_a****			n	0	0	0	0
Allocations		drv_C	Variable new		Ins	0	0	0	0
Alarm		drv_E	Create standard fu	nction		0	0	0	0
Units		drv_N				0	0	0	0
🕀 🖅 Screens		drv_N	Linked elements		•	0	0	0	0
Functions		drv_N	Copy		Ctrl+C	0	0	0	0
		drv_N	Paste		Ctrl+V	0	0	0	0
Historian		drv_N				0	0	0	0
Time control		drv_S	Delete		Del	0	0	0	0
e e Programming interfa	+-	Interr	Expand/collapse n	odo		0	0	0	0
S straton (IEC 6113	+-	Interr		oue	,	0	0	0	0
- Production & Facilit	•-	Interr	Activate all			0	0	0	0
- Interlockings			Activate						
- E Command			Deactivate						
- 🚇 Energy Managemer									
- 🔋 Message Control			Export XML selecte	d					
- 2   Report Generator			Import XML						
State administrati			Extended import/ex	mort	,				
SAP interface				4					
🗄 🧰 Files			Extended filter		•				
			Remove all filter						
- Projektsicherun		_							
	·	otal / 1:	Edit selected cell		F2	able			

- 2. The dialog for configuring variables is opened
- 3. configure the variable



ate variable			×
Settings Name: S. Variable			
S_Variable			
Driver			
SIMUL32 - SIMUL_WIZ		-	
Driver object type			
PLC marker		•	
Datatype			
REAL			
Array settings			
<ul> <li>Start index is 0</li> </ul>			
Dim 1	Dim 2	Dim 3	
0	0	0	
Addressing options			
Automatic addressing			
Addressing according to	data type offset and sta	rt offset	
Manual addressing			
Each datatype starts wit	th new offset		
Automatic addressing			
Activate all elements			
<ul> <li>Activate element manual</li> </ul>	lly		
Back	Finish	Cancel He	elp

# 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. <b>Attention:</b> The <b>#</b> character is not permitted in variable names. If non- permitted characters are used, creation of variables cannot be completed and the <b>Finish</b> button remains inactive.
Drivers	Select the desired driver from the drop-down list. <b>Note:</b> 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

As of version 6 it is possible to directly import the TwinCat variable in the Editor online. In order to do this, yoou have to first browse the variables in the driver configuration and then add them to the configuration file.



# SPLITTING UP INDEX GROUPS IN TWINCAT IN ORDER TO GET NET ADDRESSES IN ZENON VARIABLES

Net address of the variable in zenon	Indes group of the TwinCat variable
0	16384
1	16385
2	16400
3	16401
4	16416
5	16417
6	16432
7	16433
8	61472
9	61473
10	61488
11	61489

The index group identifies a certain data type (Bool, Byte ...) in TwinCat

# 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
PLC marker	8	R / W	LREAL, BOOL, REAL, DINT, UDINT, INT, UINT, STRING, USINT, SINT	
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 31)

### **OBJECTS FOR PROCESS VARIABLES IN ZENON**

Object	Read	Write	Comment
Configuratio n			
Bit marker	Y	Y	Address range (Offset) from 0 - 65535, bit 0 - 7
Byte marker	Y	Y	Address range (Offset) from 0 - 65535
Word marker	Y	Y	Address range (Offset) from 0 - 65535
Input bit	Y	Y	Address range (Offset) from 0 - 65535, bit 0 - 7
InputByte	Y	Y	Address range (Offset) from 0 - 65535
InputWord	Y	Y	Address range (Offset) from 0 - 65535
Output Bit	Y	Y	Address range (Offset) from 0 - 65535, bit 0 - 7
Output byte	Y	Y	Address range (Offset) from 0 - 65535
Output Word	Y	Y	Address range (Offset) from 0 - 65535

# 7.3.2 Mapping of the data types

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



PLC	zenon
IX 1.0	Input bit offset 1 bit 1
I1	Input byte offset 1 low-order
MB 100	Byte marker offset 100 low-order
MX 100.0	Bit marker offset 100 bit 0
MW 100	Word marker offset 100
Q1	Output byte offset 1 low-order
QW 60000	Output word offset 60000
QX 1.1	Output bit offset 1 bit 1

#### **EXAMPLE FOR ALL POSSIBLE ZENON DATA TYPES**

**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 INTwith 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 $\ensuremath{\mathbbm 1}$ 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



BITADR	N	2	For bit variables: bit address For byte variables: 0=lower, 8=higher byte For string variables: Length of string (max. 63 characters)	
ARRAYSIZE	N	16	Number of variables in the array for index variables ATTENTION: Only the first variable is fully available. All others are only available for VBA or the 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	
SIGMIN	Float	16	Non-linearized signal - minimum (signal resolution)	
SIGMAX	F	16	Non-linearized signal - maximum (signal resolution)	
ANZMIN	F	16	Technical value - minimum (measuring range)	
ANZMAX	F	16	Technical value - maximum (measuring range)	
ANZKOMMA	N	1	Number of decimal places for the display of the values (measuring range)	
UPDATERATE	F	19	Update rate for mathematics variables (in sec, one decimal possible) not used for all other variables	
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	C	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 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 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 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:

#### **ZENON6.INI ENTRIES**

[BECKHOFF]



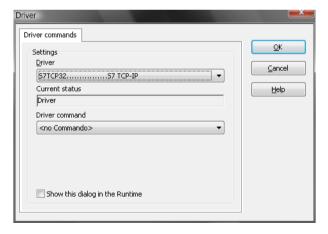
SIMUL=0 0....Hardware mode 1....Simulation mode

# 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



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.
<ul> <li>Start driver (online mode)</li> </ul>	Driver is reinitialized and started.
• Stop driver (offline	Driver is stopped. No new data is accepted.



mode)	Note: If the driver is in offline mode, all variables that were created for this driver receive the status switched off (OFF; Bit 20).
<ul> <li>Driver in simulation mode</li> </ul>	Driver is set into simulation mode. The values of all variables of the driver are simulated by the driver. No values from the connected hardware (e.g. PLC, bus system,) are displayed.
<ul> <li>Driver in hardware mode</li> </ul>	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.
<ul> <li>Driver-specific command</li> </ul>	Enter driver-specific commands. Opens input field in order to enter a command.
Activate driver write set value	e Write set value to a driver is allowed.
<ul> <li>Deactivate driver write set value</li> </ul>	Write set value to a driver is prohibited.
<ul> <li>Establish connection with modem</li> </ul>	Establish connection (for modem drivers) Opens the input fields for the hardware address and for the telephone number.
Disconnect from moder	n 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. Error analysis**

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

# 10.1 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 Log 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 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

#### 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.

### 💡 Info

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

# 10.2 Check list

Are the used variables correctly defined in the PLC?

Is the 'default.bec' configuration file of the driver located on the CE device?

Have you analyzed the error text file (which errors did occur)?

Send the zenon project to support@copadata.com (mailto:support@copadata.com)