



COPADATA
do it your way

zenon driver manual

SYSDRV

v.7.11





©2014 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. Welcome to COPA-DATA help	5
2. SYSDRV	5
3. SYSDRV - Data sheet	6
4. Driver history	8
5. Configuration	8
5.1 Creating a driver	9
6. Creating variables	10
6.1 Creating system driver variables	11
6.2 General notes	15
6.3 Topic - Alarms	16
6.4 Topic - Historian	16
6.5 Topic - Batch Control	17
6.6 Topic - Command	19
6.7 Topic - User-defined	21
6.8 Topic - User administration	22
6.9 Topic - Printer	24
6.10 Topic - Hardware resources	25
6.11 Topic - Network	26
6.12 Topic - Folder	32
6.13 Topic - Performance Statistics Network	33
6.14 Topic - Performance Statistics Driver	36
6.15 Topic - Project information	38
6.16 Topic - standard recipes and Recipe Group Manager	41
6.17 Topic - System information	52
6.18 Creating variables by importing	53
6.18.1 XML import	54
6.18.2 DBF Import/Export	54
7. Driver-specific functions	61

8. Driver commands	61
9. Error analysis.....	63
9.1 Analysis tool	63

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. SYSDRV

zenon system driver for reading out internal project information, network information, hardware information of the PC etc.

The driver does not use up any licensed tags.

**Attention**

Each system driver variable can only be present once per project.

Which variables are present and how often these are present can be checked with the Diagnosis Viewer (on page 63) tool. The corresponding error message starts with:
Systemdriver variable is not unique!

3. SYSDRV - Data sheet

General:	
Driver file name	SYSDRV.exe
Driver name	Driver for system variables
PLC types	-
PLC manufacturer	zenon system driver; Internal driver;

Driver supports:	
Protocol	unknown;
Addressing: Address-based	-
Addressing: Name-based	x
Spontaneous communication	-
Polling communication	x
Online browsing	-
Offline browsing	-

Real-time capable	-
Blockwrite	-
Modem capable	-
Serial logging	-
RDA numerical	-
RDA String	-

Requirements:	
Hardware PC	-
Software PC	-
Hardware PLC	-
Software PLC	-
Requires v-dll	x

Platforms:	
Operating systems	Windows CE 6.0, Embedded Compact 7; Windows Vista, 7, 8, 8.1 Server 2008/R2, Server 2012/R2;
CE platforms	x86; ARM;

4. Driver history

Date	Driver version	Change
07.07.08	600	Created driver documentation

DRIVER VERSIONING

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

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

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



Example

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

5. Configuration

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



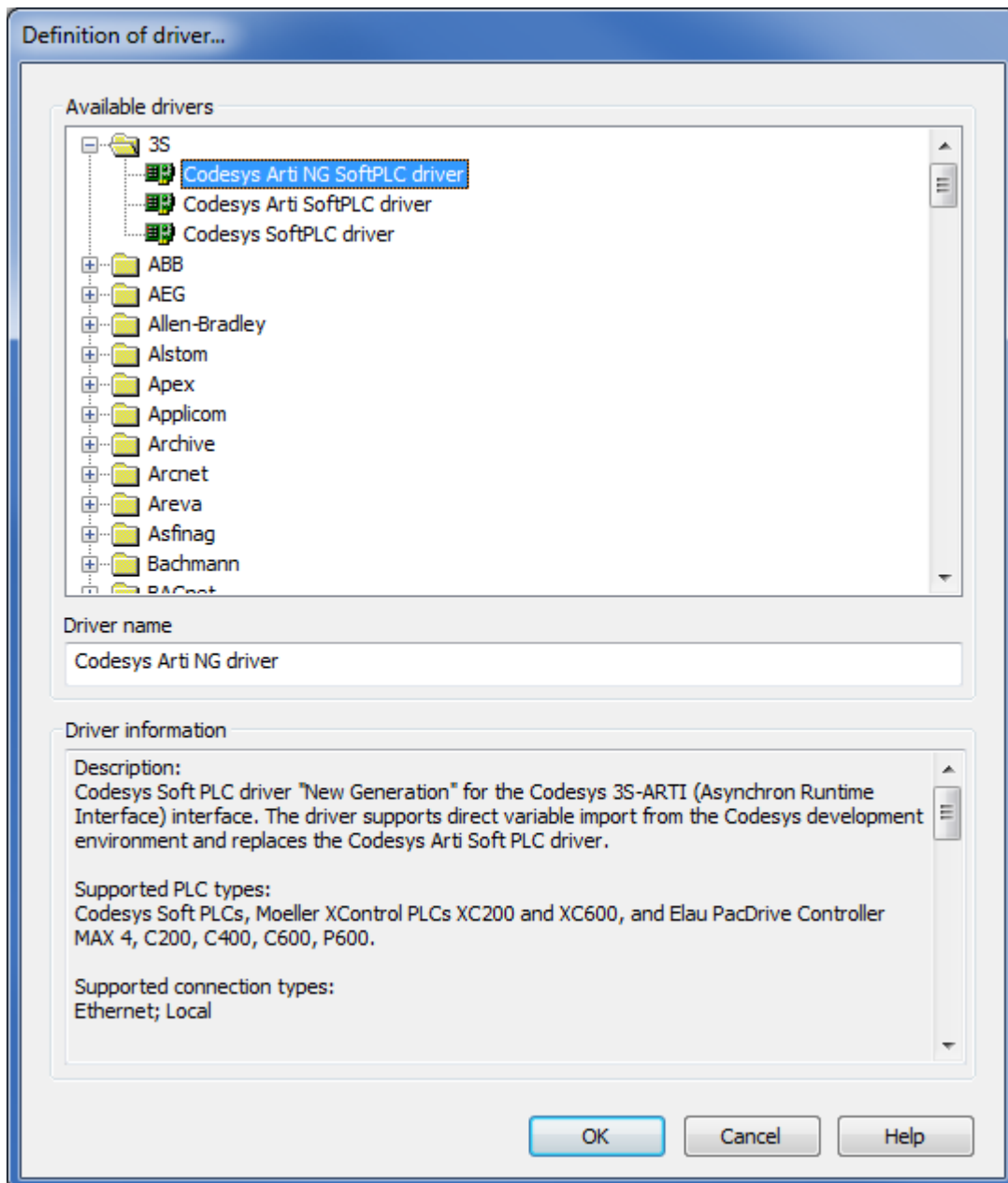
Information

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

5.1 Creating a driver

In order to create a new driver:

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



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

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

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



Information

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

- ▶ Internal
- ▶ MathDr32
- ▶ SysDrv.

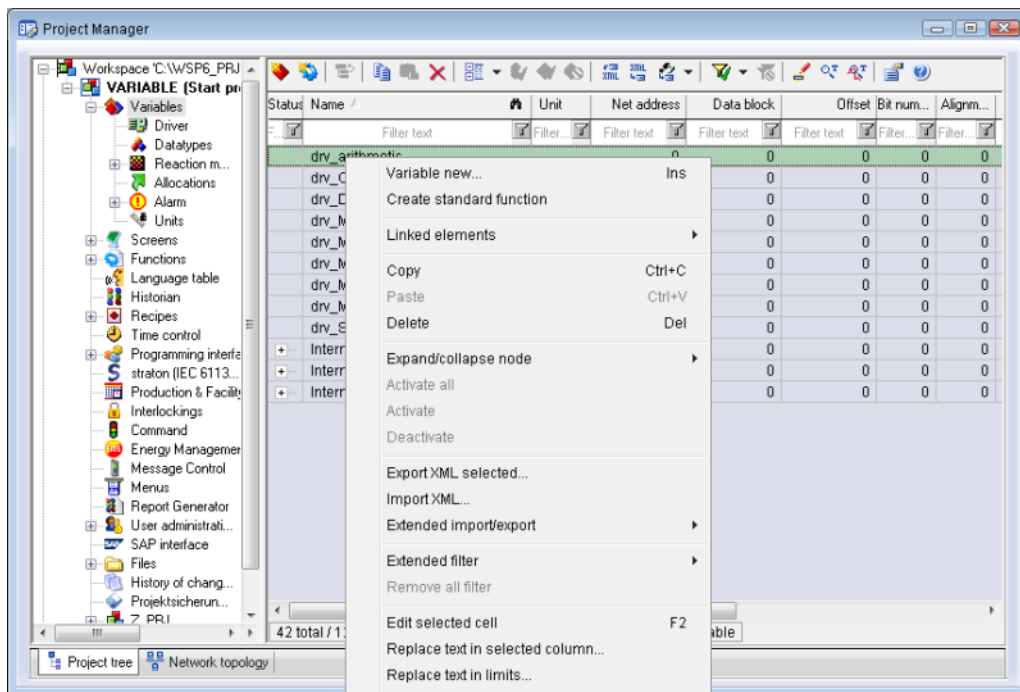
▶

6. Creating variables

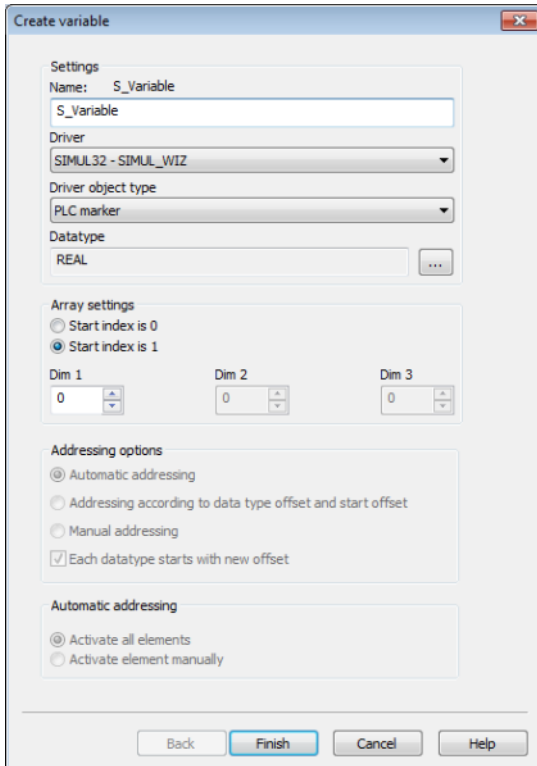
This is how you can create variables in the zenon Editor:

6.1 Creating system driver variables

In the Project manager right-click on **variable** and select **New variable...** in the context menu.



In the following dialog select as Driver SYSDRV – Driver for system variables and as Driver object type System driver variable.

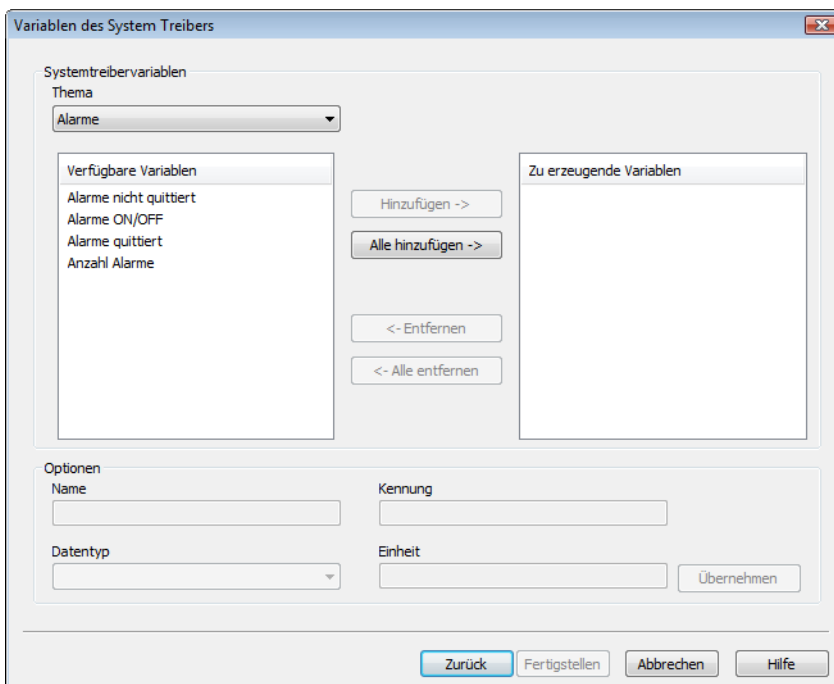


The 'Create variable' dialog box is shown with the following settings:

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

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

Click on **Next**.



The 'Variablen des System Treibers' dialog box is shown with the following settings:

- Systemtreibervariablen:**
 - Thema: Alarme
- Verfügbare Variablen:**
 - Alarme nicht quittiert
 - Alarme ON/OFF
 - Alarme quittiert
 - Anzahl Alarme
- Zu erzeugende Variablen:** (Empty list)
- Buttons:** Hinzufügen ->, Alle hinzufügen ->, <- Entfernen, <- Alle entfernen
- Optionen:**
 - Name: (Empty field)
 - Kennung: (Empty field)
 - Datentyp: (Empty dropdown)
 - Einheit: (Empty field)
 - Übernehmen: (Button)

Buttons at the bottom: Zurück, Fertigstellen, Abbrechen, Hilfe.

The following settings are available.

Parameters	Description
Theme	Select the subject area from which you want to create system driver variables. As long as you edit an existing variable, the selection is disabled.
Available variables	All system driver variables which are still available for the selected subject area. Already created variables or variables which are displayed in list Variables to be created are not displayed.
Add	Adds the currently selected variables to the list of the variables to be created.
Add all	Adds all variables which are displayed in list Available variables to the list of the variables to be created.
Delete	Removes all selected variables from the list of the variables to be created.
Remove all	Removes all variables from the list of the variables to be created.
Variables to be created	All system driver variables which you have selected and created with the help of button Finish . If you select one variable, you can edit it under Options .
Tools	
Name	Edit the name of the selected variable. Note: Only possible if you have selected only one variable. This variable must be a user-defined variable.
Identification	Edit the identification of the selected variable. Note: Only possible if you have selected only one variable.
Data type	Edit the data type of the selected variable. Note: Only possible if you have selected only one variable. This variable must be a user-defined variable.
Measuring unit	Edit the measuring unit of the selected variable. Note: Only possible if you have selected only one variable.
Apply	Left-click this button in order to apply the changed settings to the selected variable.

	<p>Note: If you select another variable before you click on Apply, all changes are canceled. Only valid changes can be applied.</p>
--	--

If you have selected and edited all system driver variables which you want to create, you can create them by left-clicking **Finish**.



Info

*In the lists **Available variables** and **Variables to be created** several entries can be selected at once. This happens with:*

- ▶ **Ctrl+left-click:** By pressing and holding **Ctrl** you can select a number of entries.
- ▶ **Shift+left-click:** All entries that are between the two selected entries are selected.
- ▶ If **Ctrl** and **Shift** are held down at the same time, all entries between the two are selected. Entries already selected are retained.
- ▶ **Ctrl+A:** All entries are selected.

CHANGING THE NAME OF A SYSTEM DRIVER VARIABLE

If you want to change the pre-defined names of the system driver variables, you can do this as follows: Export the desired variables to XML, change the name of the variables in the XML file (property **Name**) and import the file.

6.2 General notes

VARIABLES IN THE NETWORK

Variables that contain a reference (global/local) at the end of the name can be created in either the global or local version. All other variables are always local.

- ▶ All local variables are saved locally on the respective workstation (client/server/standby). The behavior of the variable can change, according to the position of the computer in the network topology.
- ▶ All global variables are depicted on the server (standby) and distributed to the network (clients). Changes to these variable values are always applicable throughout the network.

RELOAD

The values of all system driver variables remain unchanged after reloading.

6.3 Topic - Alarms

The following system driver variables are available for this subject area:

Name	Data type	Comment
Alarms not acknowledged	UDINT	Shows the number of not acknowledged alarms in the memory
Alarm Message List active/inactive	BOOL	Displays the status of the Alarm Message List: <ul style="list-style-type: none"> ▶ 0: inactive ▶ 1: active
Alarms acknowledged	UDINT	Shows the number of acknowledged alarms in the memory
Number of alarms	UDINT	Shows the number of alarms in memory.

6.4 Topic - Historian

The following system driver variables are available for this subject area:

Name	Data type	Comment
Memory for Historian	UDINT	Shows the amount of memory reserved for archive data processing (e.g. in the Extended Trend).
Last lot selected	STRING	Contains the names last selected in the online lot selection.
	local	

6.5 Topic - Batch Control

The following system driver variables are available for this subject area:

Name	Data type	Comment
Recipe terminated because of an error	BOOL	Is set to 1 if a recipe cannot be restarted. Before creating an image, the variable is set to 0. This variable is distributed in the network.
Input set value: Measuring unit	STRING	Unit of the variable.
Input set value: Parameter Maximum	STRING	Parameter Maximum The variable is filled with the maximum of the parameter when edited. If the minimum or the maximum of the parameter is edited, the variable with the maximum of the variables is filled. If the "Amendable in the control recipe" bit is edited, it receives the value 1.
Input set value: Parameter Minimum	STRING	Parameter Minimum The variable is filled with the minimum of the parameter when edited. If the minimum or the maximum of the parameter is edited, the variable with the minimum of the variables is filled. If the "Amendable in the control recipe" bit is edited, it receives the value 0.
Input set value: Parameter description	STRING	Description of the tag.
Input set value: Parameter property	STRING	Property that is edited at the parameter.
Input set value: Parameter name	STRING	Name of the parameter whose property is being edited.
Input set value: String length	DINT	String length of the variable.
Input set value: Variable maximum	STRING	Maximum of variables linked to the parameter.
Input set value: Variable minimum	STRING	Minimum of variables linked to the parameter.

6.6 Topic - Command

The following system driver variables are available for this subject area:



Information

The system driver variables of the subject area Command are only available in the Energy Edition.

Name	Data type	Comment
Screen name	STRING	Name of the current screen of the type Command
Screen step	DINT	Number of the current step of the command 0 = Initialization 1 = Step 1 2 = Unlock 3 = Step 2 4 = Lock 5 = Execution
Identification of the action variable	DINT	Identifies the type of the currently active action 0 = Single command 1 = Double command 2 = Setpoint input 3 = Status input 4 = Replace 5 = Manual correction 6 = Block 7 = Release 8 = Lock 9 = Revision 1000 = Not defined
Name of the action variable	STRING	Name of the data point on which the current action is executed.
Name of the response variable	STRING	Shows the name of the response variable.
Name of the interlocking	STRING	Shows the name of the command group.
Parameter of the action	UDINT	Parameter of the current action. In case of a single command, the command status will be displayed; otherwise, the current switching direction will be displayed.
Update	BOOL	This variable is set to 1 if the listed variables are consistent.
Interlocking code	UDINT	Status of the current interlocking. 0 = Not checked 1 = No active interlocking 2 = Value already set 3 = Not used 4 = Not used 5 = Write failure 6 = Internal Error 7 = Not used

		8 = Data point type invalid 9 = Not used 10 = Command interlocking does not exist 11 = Action does not exist 12 = Not used 13 = Not used 14 = Value missing 15 = Invalid value 16 = Topology unlockable 17 = Topology not unlockable 18 = Not used 19 = Not used 20 = Interlocking of the action unlockable 21 = Interlocking of the action not unlockable 22 = Not used 23 = Locked against command 24 = Locking administration invalid 25 = SBO expired 26 = Timeout for SBO activation 27 = Timeout for SBO Deactivation 28 = Timeout for Action execution 29 = SBO expired
Interlocking message	STRING	Shows the text of the active interlocking message.

6.7 Topic - User-defined

The following system driver variables are available for this subject area:

Name	Data type	Comment
User-defined1000	BOOL, INT, LREAL, REAL, SINT, STRING, UDINT, UINT, USINT	These are local variables that are not updated in the network. You can enter any name, identification, data type and unit. Attention: These variables have been obsolete since the introduction of the internal driver and should no longer be used.

6.8 Topic - User administration

The following system driver variables are available for this subject area:

Name	Data type	Comment
Current authorization group 1	UDINT	<p>Available authorization level for the user currently logged in to zenon. This information is displayed according to the bit encoding system of the user groups.</p> <p>Authorization level 0 corresponds to decimal value $2^0 = 1$ Authorization level 1 corresponds to decimal value $2^1 = 2$ Authorization level 0 and 1 corresponds to decimal value 2^0 and $2^1 = 3$ etc.</p> <p>The 128 possible authorization levels are distributed to 4 double words (authorization groups 1-4) with increasing value.</p> <p>Indicates authorization level 0 - 31.</p>
Current authorization group 2	UDINT	<p>Available authorization level for the user currently logged in to zenon. This information is displayed according to the bit encoding system of the user groups.</p> <p>Authorization level 0 corresponds to decimal value $2^0 = 1$ Authorization level 1 corresponds to decimal value $2^1 = 2$ Authorization level 0 and 1 corresponds to decimal value 2^0 and $2^1 = 3$ etc.</p> <p>The 128 possible authorization levels are distributed to 4 double words (authorization groups 1-4) with increasing value.</p> <p>Indicates authorization level 32 - 63.</p>
Current authorization group 3	UDINT	<p>Available authorization level for the user currently logged in to zenon. This information is displayed according to the bit encoding system of the user groups.</p> <p>Authorization level 0 corresponds to decimal value $2^0 = 1$ Authorization level 1 corresponds to decimal value $2^1 = 2$ Authorization level 0 and 1 corresponds to decimal value 2^0 and $2^1 = 3$ etc.</p> <p>The 128 possible authorization levels are distributed to 4 double words (authorization groups 1-4) with increasing value.</p> <p>Indicates authorization level 64 - 95.</p>
Current authorization group 4	UDINT	<p>Available authorization level for the user currently logged in to zenon. This information is displayed according to the bit encoding system of the user groups.</p> <p>Authorization level 0 corresponds to decimal value $2^0 = 1$ Authorization level 1 corresponds to decimal value $2^1 = 2$ Authorization level 0 and 1 corresponds to decimal value 2^0 and $2^1 = 3$ etc.</p>

		<p>The 128 possible authorization levels are distributed to 4 double words (authorization groups 1-4) with increasing value.</p> <p>Indicates authorization level 96 - 127.</p>
User - full name	STRING	Displays the full name of the user currently logged in in zenon.
User name	STRING	Displays the user name of the user currently logged in in zenon.
Days until password expires	DINT	<p>Displays the number of days until the password of the user currently logged in becomes invalid.</p> <p>The variable has days as a time format. The time duration can be defined in zenon with the Password - period of validity [d] property.</p> <p>Note: If the Password never expires option is activated in Active Directory or, in zenon, the Password - period of validity [d] property is set to 0, then the value 2147483647 is written to the variable.</p>
Invalid user name	BOOL	If the value of this variable is 1, the login failed either because of wrong user name or because of a wrong password.

6.9 Topic - Printer

The following system driver variables are available for this subject area:

Name	Data type	Comment
List printer - number of jobs	UDINT	Current number of waiting print jobs for the configured list printer.
List printer - Name	STRING	Name of the currently set list printer.
List printer - Status	UDINT	Shows the current status of the list printer. Printer_Status_Paused = 0x00000001, Printer_Status_Pending_Deletion=0x00000004
Offline-printer AML/CEL - number of jobs	UDINT	Current number of print jobs waiting for the configured offline printer AML/CEL.
Offline-printer AML/CEL - name	STRING	Name of the currently configured offline printer AML/CEL.
Offline-printer AML/CEL - status	UDINT	Shows the current status of the offline printer AML/CEL. Printer_Status_Paused = 0x00000001, Printer_Status_Pending_Deletion=0x00000004
Switch online-printer on/off	BOOL	Status of the online print administration of the Alarm Message List or the Chronological Event List. 0= off 1 = on
Screenshot printer - number of jobs		Current number of waiting print jobs for the configured screenshot printer.
Screenshot printer - name		Name of the currently configured screenshot printer.
Screenshot printer - status		Shows the current status of the screenshot printer. Printer_Status_Paused = 0x00000001, Printer_Status_Pending_Deletion=0x00000004
Who is printing online? AML/CEL	BOOL	Shows for which module the online print is active. 0 = Alarm Message List 1 = Chronological Event List

6.10 Topic - Hardware resources

The following system driver variables are available for this subject area:

Name	Data type	Comment
Free memory (%)	UDINT	Shows the available memory of the current workstation in %.
Free memory (%) - SB	UDINT	Shows the available memory of the standby server in %.
Free memory (kB)	UDINT	Shows the available memory of the current workstation in kilobytes.
Free memory (kB) - SB	UDINT	Shows the available memory of the standby server in kilobytes.
Free disk space - database (%)	UDINT	Shows the available disk space of the current workstation in %. This value always refers to the partition on which the Runtime folder of the project is located.
Free disk space - database (%) - SB	UDINT	Shows the available disk space of the standby server in %. This value always refers to the partition on which the Runtime folder of the project is located.
Free disk space - database (kB)	UDINT	Shows the available disk space of the current workstation in kilobytes. This value always refers to the partition on which the Runtime folder of the project is located.
Free disk space - database (kB) - SB	UDINT	Shows the available disk space of the standby server in kilobytes. This value always refers to the partition on which the Runtime folder of the project is located.
Free disk space - Export (%)	UDINT	Shows the amount of free disk space of the export folder configured in the project configuration in %.
Free disk space - export (%) - SB	UDINT	Shows the amount of free disk space of the export folder of the standby server configured in the project configuration in %.
Free disk space - export (kB)	UDINT	Shows the amount of free disk space of the export folder configured in the project configuration in kilobytes.
Free disk space - export (kB) - SB	UDINT	Shows the amount of free disk space of the export folder of the standby server configured in the project configuration in kilobytes.

6.11 Topic - Network

The following system driver variables are available for this subject area:

Name	Data type	Comment
Current primary server	STRING local	Computer name of the server currently handling processes If the name was acquired from the hosts file, this will be the name used there. For DNS, this is the Fully Qualified Domain Name. Note: If the network is deactivated, the variable sends the status INVALID. The Current Standby Server remains empty in contrast.
Current Primary Server	STRING local	Computer name of the server which is currently not handling processes. If the name was acquired from the hosts file, this is the name entered there. For DNS, this is the Fully Qualified Domain Name.
Number of connected Clients	UDINT	Delivers the number of clients currently connected to the server. This number also includes the standby server, if there is one.
Authorization: not granted	BOOL	Shows whether a requested authorization is denied in the network. The value of this variable is changed only for a short time and then changed back to the initial state. ▶ 0 = operating authorization request accepted ▶ 1 = operating authorization request declined
Authorization: Present here	BOOL	Shows whether there is operating authorization for the current project on the local computer. ▶ 0 = No ▶ 1 = Yes
Authorization: Computer that owns it.	STRING	Shows the name of the computer that has the authorization for the currently loaded project.
Result of evaluation, Server 1	UDINT	In the event of changes to a variable from the evaluation matrix, this value is written to the corresponding system driver variable for Server 1 and Server 2 after calculation of the new result of the evaluation. The values are equal to one another (server <-> standby), so that the current value is always displayed on both sides. However, after the other side has a failure, this value remains for the attendant variable and only updates itself once it reconnects.

Result of evaluation, Server 2	UDINT	In the event of changes to a variable from the evaluation matrix, this value is written to the corresponding system driver variable for Server 1 and Server 2 after calculation of the new result of the evaluation. The values are equal to one another (server <-> standby), so that the current value is always displayed on both sides. However, after the other side has a failure, this value remains for the attendant variable and only updates itself once it reconnects.
Names of connected clients	STRING	Delivers the names of the clients currently connected to the server. The standby server, if there is one, is also included.
Network timeout [milliseconds]	UDINT	Shows the timeout in milliseconds for the zenon network as configured in the project configuration.
Primary Server <-> Standby Server data synchronization		
\$\$ Redundanz-Umschaltung	BOOL	<p>A binary variable that takes the value 1 for a short time when the system performs a redundancy switch between server and standby server.</p> <ul style="list-style-type: none"> ▶ 0 = No redundancy switch over ▶ 1 = Redundancy switch over
Primary server failure	BOOL local	<p>Indicates that the connection to the process handling server was lost.</p> <p>Depending on the network position of the computer, this means:</p> <ul style="list-style-type: none"> ▶ Dominant Server: While it is not yet the process handling server, the value changes to TRUE if the connection to the process handling server is lost. Always FALSE after synchronization. ▶ Non-dominant Server: Changes to TRUE if the connection to the dominant server, which was the process handling server, is lost. Changes back to FALSE if the Standby Server was promoted to be the Primary Server. EVALUATION: Preferably via a REMA, as the Alarm Management is also swapped and taken over by the SB at that time. The Online Container is also not suitable because the variables are re-initialized during redundancy switching. ▶ Client: Changes to TRUE if the connection to the process handling server is lost. Changes back to FALSE if the client connects to the SB computer that is now the process handling

		server.
--	--	---------

Primary server shut down	BOOL local	<p>Indicates the regular stop of the process handling server. The value changes to TRUE if the process handling server was stopped properly. FALSE if there is a process handling server in the net.</p> <p>Depending on the network position of the computer, this means:</p> <ul style="list-style-type: none"> ▶ • Dominant Server: While it is not yet the process handling server, the value changes to TRUE if the process handling server has stopped. ▶ • Non-dominant Server: Changes to TRUE if the dominant server, which was the process handling server, has stopped. Changes back to FALSE if the StandBy was promoted to be the process handling server. ▶ EVALUATION: Preferably via a REMA, as the Alarm Management is also swapped and taken over by the SB at that time. The Online Container is also not suitable because the variables are re-initialized during redundancy switching. ▶ • Client: Changes to TRUE if the dominant server has stopped. Changes back to FALSE if the client connects to the SB computer that is now the process handling server. Is also TRUE while the process handling non-dominant server changes back to be the non-process handling server.
Standalone/Primary Server/Standby Server/Client	DINT	<p>Shows the type of the local computer in the network.</p> <ul style="list-style-type: none"> ▶ -1 = Standalone ▶ 0 = Client ▶ 1 = Primary Server ▶ 2 = Standby Server
\$\$ Server-Standby im Dateiabgleich	BOOL	<p>A binary variable that takes on the value 1 if the server and the standby server are synchronizing files after a redundancy switch.</p> <p>0 = no file sync 1 = file sync active</p>
Standby Server failure	BOOL	<p>Changes to TRUE if the connection to the currently non-process handling server is terminated unexpectedly. If</p>

		<p>there is a connection, the value is FALSE.</p> <p>Depending on the network position of the computer, this means:</p> <ul style="list-style-type: none"> ▶ Dominant Server: The variable only acts as described from the time when the standby became the server handling the process. ▶ Non-dominant Server: If, during file synchronization, the connection to a server that is dominant but is not handling the process is interrupted, the value changes to TRUE. Always FALSE if not the server handling the process. ▶ Client: As per server handling the process.
Standby Server shut down	BOOL	<p>Is TRUE on the process handling server, if the non-process handling server was stopped properly and if there is no connection anymore. Changes to FALSE if the non-process handling server has registered at the process handling server.</p> <p>Depending on the network position of the computer, this means:</p> <ul style="list-style-type: none"> ▶ Dominant Server: Only from the time when the standby became the server handling the process does the variable act as described. ▶ Non-dominant Server: If this is ended during file synchronization with a server that is dominant but is not handling the process, the value changes to TRUE. Always FALSE if not the server handling the process. ▶ Client: As per server handling the process.
Standby Server started:	BOOL	<p>TRUE if the non-process handling server has registered at the process-handling server and if the data update was performed and the connection between the two computers is active.</p> <p>Depending on the network position of the computer, this means:</p> <ul style="list-style-type: none"> ▶ Dominant Server: Only from the time when the standby became the server handling the process does the variable act as described. ▶ Non-dominant Server: Becomes TRUE if the dominant server not handling the process starts. Changes to FALSE if the

		<p>computer is the server handling the process.</p> <ul style="list-style-type: none"> ▶ Client: As per server handling the process.
Switch from Primary Server to Standby Server	BOOL	<p>A binary variable that takes on the value 1 if the server becomes the standby server during a redundancy switch.</p> <ul style="list-style-type: none"> ▶ 0 = registered server is available as server in the network. ▶ 1 = registered server is available as standby server in the network.
Switch from Standby Server to Primary Server	BOOL	<p>A binary variable that takes on the value 1 if the standby server becomes the server during a redundancy switch.</p> <ul style="list-style-type: none"> ▶ 0 = registered Standby Server is available as standby server in the network. ▶ 1 = registered Standby Server is available as server in the network.

6.12 Topic - Folder

The following system driver variables are available for this subject area:

Name	Data type	Comment
Bitmap folder	STRING	Shows the folder for the graphics of the active project that were inserted via the project tree. You can edit the folder settings in the project administration.
Database folder	STRING	Shows the working folder of the active project.
List folder	STRING	Shows the folder for the lists of the active project that were inserted via the project tree.
Report folder	STRING	Shows the folder for the report files created in zenon.
Video folder	STRING	Shows the folder for the multimedia files of the active project that were inserted via the project tree.

6.13 Topic - Performance Statistics Network

The following system driver variables are available for this subject area:

Name	Data type	Comment
Pending messages (total)	UDINT	Shows the currently pending messages in the Windows Message Queue for the total application.
Pending messages (project)	UDINT	Shows the currently pending messages in the Windows Message Queue for the current project.
Received packets count	UDINT	Cumulated number of received network packets. Write set value initializes the counter.
Received packets count / s current	UDINT	Current number of received network packets per second. Set value function initializes Min and Max.
Received packets count / s current local	UDINT	Current number of locally received network packets per second. Set value function initializes Min and Max.
Received packets count / s maximum	UDINT	Current maximum number of received network packets per second.
Received packets count / s maximum local	UDINT	Current maximum number of locally received network packets per second.
Received packets count / s minimum	UDINT	Current minimum number of received network packets per second.
Received packets count / s minimum local	UDINT	Current minimum number of locally received network packets per second.
Received packets count local	UDINT	Cumulated number of locally received network packets. Write set value initializes the counter.
Received packets processing duration maximum [µs]	UDINT	Maximum processing duration for received network packets.
Received packets processing duration maximum [µs] local	UDINT	Maximum processing duration for locally received network packets.
Received packets processing duration minimum [µs]	UDINT	Minimum of the processing duration for received network packets.
Received packets processing duration minimum [µs] local	UDINT	Minimum of the processing duration for locally received network packets.
Received packets processing duration average [µs]	UDINT	Average of the processing duration for received network packets. Set value function initializes Counter, Min and Max.

Received packets processing duration average [μs] local	UDINT	Average of the processing duration for locally received network packets. Set value function initializes Counter, Min and Max.
Received packets size / s current [byte]	UDINT	Current size of received network packets per second. Set value function initializes Min and Max.
Received packets size / s current [byte] local	UDINT	Current size of locally received network packets per second. Set value function initializes Min and Max.
Received packets size / s maximum [byte]	UDINT	Current maximum size of received network packets per second.
Received packets size / s maximum [byte] local	UDINT	Current maximum size of locally received network packets per second.
Received packets size / s minimum [byte]	UDINT	Current minimum size of received network packets per second.
Received packets size / s minimum [byte] local	UDINT	Current minimum size of locally received network packets per second.
Received packets size [byte]	UDINT	Accumulated size of received network packets. Set value function initializes the counter.
Received packets size [byte] local	UDINT	Accumulated size of locally received network packets. Set value function initializes the counter.
Sent packets count	UDINT	Accumulated number of sent network packets. Set value function initializes the counter.
Sent packets count / s current	UDINT	Current number of sent network packets per second. Set value function initializes Min and Max.
Sent packets count / s current local	UDINT	Current number of locally sent network packets per second. Set value function initializes Min and Max.
Sent packets count / s maximum	UDINT	Current maximum number of sent network packets per second.
Sent packets count / s maximum local	UDINT	Current maximum number of locally sent network packets per second.
Sent packets count / s minimum	UDINT	Current minimum number of sent network packets per second.
Sent packets count / s minimum local	UDINT	Current minimum number of locally sent network packets per second.
Sent packets count local	UDINT	Accumulated number of locally sent network packets.

		Set value function initializes the counter.
Sent packets size / s current [byte]	UDINT	Current size of sent network packets per second. Set value function initializes Min and Max.
Sent packets size / s current [byte] local	UDINT	Current size of locally sent network packets per second. Set value function initializes Min and Max.
Sent packets size / s maximum [byte]	UDINT	Current maximum size of sent network packets per second.
Sent packets size / s maximum [byte] local	UDINT	Current maximum size of locally sent network packets per second.
Sent packets size / s minimum [byte]	UDINT	Current minimum size of sent network packets per second.
Sent packets size / s minimum [byte] local	UDINT	Current minimum size of locally sent network packets per second.
Sent packets size [byte]	UDINT	Accumulated size of sent network packets. Set value function initializes the counter.
Sent packets size [byte] local	UDINT	Accumulated size of locally sent network packets. Set value function initializes the counter.

6.14 Topic - Performance Statistics Driver

The following system driver variables are available for this subject area:

Name	Data type	Comment
Received values count / s current	UDINT	Current number of received values per second. Set value function initializes Min and Max.
Received values count / s current local	UDINT	Current number of locally received values per second. Set value function initializes Min and Max.
Received values count / s maximum	UDINT	Current maximum number of received values per second.
Received values count / s maximum local	UDINT	Current maximum number of locally received values per second.
Received values count / s minimum	UDINT	Current minimum number of received values per second.
Received values count / s minimum local	UDINT	Current minimum number of locally received values per second.
Received values duration maximum [µs]	UDINT	Maximum value of processing duration for received values.
Received values duration maximum [µs] local	UDINT	Maximum value of processing duration for locally received values.
Received values duration minimum [µs]	UDINT	Minimum value of processing duration for received values.
Received values duration minimum [µs] local	UDINT	Minimum value of processing duration for locally received values.
Received values duration average [µs]	UDINT	Average value of the editing time for received values. Writing the set value initializes counter, minimum and maximum.
Received values duration average [µs] local	UDINT	Average value of the editing time for local received values. Writing the set value initializes counter, minimum and maximum.
Received value lists count / s current	UDINT	Current number of received value lists per second. Set value function initializes Min and Max.
Received value lists count / s current local	UDINT	Current number of locally received value lists per second. Set value function initializes Min and Max.
Received value lists count / s maximum	UDINT	Current maximum number of received value lists per second.
Received value lists count / s maximum local	UDINT	Current maximum number of locally received value lists per second.

Received value lists count / s minimum	UDINT	Current minimum number of received value lists per second.
Received value lists count / s minimum local	UDINT	Current minimum number of locally received value lists per second.
Received value lists duration maximum [µs]	UDINT	Maximum value of the processing duration for received value lists.
Received value lists duration maximum [µs] local	UDINT	Maximum value of processing duration for locally received value lists.
Received value lists duration minimum [µs]	UDINT	Minimum value of the processing duration for received values.
Received value lists duration minimum [µs] local	UDINT	Minimum value of processing duration for locally received value lists.
Received value lists duration average [µs]	UDINT	Average value of the editing time for received value lists. Writing the set value initializes counter, minimum and maximum.
Received value lists duration average [µs] local	UDINT	Average value of the editing time for local received value lists. Writing the set value initializes counter, minimum and maximum.
Number of sent requests	UDINT	Cumulated number of sent requests. Set value function initializes the counter.
Number of sent requests local	UDINT	Cumulated number of locally sent requests. Set value function initializes the counter.
Sent set-values count	UDINT	Accumulated number of sent set-values. Set value function initializes the counter.
Sent set-values count local	UDINT	Cumulated number of sent set-values. Set value function initializes the counter.

6.15 Topic - Project information

The following system driver variables are available for this subject area:

Name	Data type	Comment
Number of screen-back actions	UDINT	Displays the number of possible „Screen back” actions. These actions can be configured in the zenon Editor.
Number of reports in memory	UDINT	Shows how many reports are currently executed in the memory of zenon. The reports are initiated via the function administration.
Save resolution-dependent screens active/inactive	BOOL	Status of the option SERIALIZE in the zenon6.ini file 0 = Entry SERIALIZE = 0 1 = Entry SERIALIZE = 1
Flash rate [milliseconds]	UDINT	Shows the currently configured time for the blink frequency for displaying limit violations.
wrong value for set value	BOOL	Display whether the set value input fails due to an invalid value. 0 = Set value executed 1 = Set value failed The change to 1 only happens for a short moment and can be used for a reaction matrix, as a trigger for a function, to trigger an alarm or to color certain elements. The value changes if a value higher or lower than the one set by an element or a screen of type Keyboard is entered. If the value is changed successfully by dialog Write set value , no value change takes place and the user is informed about the invalid value change via a popup.
Functions on/off	BOOL	Shows the current status of the zenon function administration. 0 = Function administration inactive 1 = Function administration active
Function logging active/inactive	BOOL	Displays whether function logging is activated for the current project. All locally executed functions are written into a log file. 0 = Function logging deactivated 1 = Function logging activated
no authorization to execute function	BOOL	Indicates whether the execution of an action (Set value, Execute function) is tried for which the current user does not have the required authorization. 0 = Action execution allowed 1 = Action execution not allowed
Type of control system (SICAM 230/zenon)	UDINT	Shows the currently used zenon version. 3 = zenon

		1 = SICAM 230
Project name	STRING	Shows the name of the active project in zenon.
Project version	STRING	Displays project version (main.chm::/32632.htm).
Send Message active	BOOL	Displays whether function Send message is active. 0 = Function Send message is inactive 1 = Function Send message is active
Input set value: Current set point input	STRING	Displays the current value of the set value input in screen Keyboard. With this you can check while screen Keyboard is open whether the entered value lies within the limits or whether the value is correct which was entered via screen specific function Display value as text or the function Set point input for keyboard screen.
Set value: Limit for maximum	STRING	Upper limit for set value input for the currently selected variable
Set value: Limit for minimum	STRING	Lower limit for set value input for the currently selected variable.
Serial Number	STRING	Shows the zenon serial number.
Driver queue overflow	BOOL	Displays whether a queue overflow occurs for a driver which is available in zenon. 0 = No queue overflow occurred 1 = Queue overflow occurred
Driver queue overflow (name)	STRING	Displays the name of the driver for which the queue overflow occurred. Variable is updated when system variable causes "driver queue overflow".
Variable for last set value input	STRING	Displays the name of the variable with the last successful set value action.
Variable for set value	STRING	Displays the name of the variable that is currently open for the set value action.
Version of the Runtime files	STRING	Shows the zenon version, for which the current project was created.
Screen opened last	STRING	Shows the name of the last open screen in zenon. The selection of the frames which are considered at this takes place in property Main frames in the project manager (Graphical design, Runtime general).

6.16 Topic - standard recipes and Recipe Group Manager

The following system driver variables are available for this subject area:

RGM GENERAL INFORMATION

Variable	Data type	Description
RGM recipe function in progress (global/local)	DINT	<p><i>States that an RGM function (including RGM screen functions) is being executed or has been ended:</i></p> <ul style="list-style-type: none"> ▶ -1: at work ▶ 0 Concluded successfully ▶ 1: Error: no user rights ▶ 2: Error: No authorization in the network ▶ 3: Error: Cancellation by the user ▶ 4: Error: undefined ▶ 5: Error: Error when saving the recipe file ▶ 6: Function cancelled via VBA
Last used filter (global/local)	STRING	Saves the last filter used for the recipe value value list.
XML import result (global/local)	DINT	<p>Displays if the import is currently active and whether the import was concluded with or without errors.</p> <p>The variable is assigned at the start and the end of the import. In the event of an error occurring, the error number of the first error that occurred is set after the import has ended. In addition, all errors that have occurred can be viewed in the CEL with additional information, such as the recipe concerned.</p> <p>These error numbers also apply for the RGM feedback variable <code>Result of the import</code>. The values are written to the defined variables at the start and end of the import of the respective recipe groups.</p> <p><u>General messages:</u></p> <ul style="list-style-type: none"> ▶ 0: Undefined status (no import has been carried out yet) ▶ 1: Import was started and is currently in progress ▶ 2: Import was concluded without an error <p><u>General error:</u></p>

		<ul style="list-style-type: none"> ▶ 3: No file selected. ▶ 4: Unknown XML-structure ▶ 5: Import file is not present. <p>(6 - 19: not in use)</p> <p><u>RGM-specific error:</u></p> <ul style="list-style-type: none"> ▶ 20: A new recipe could not be created. ▶ 21: The settings of a recipe cannot be imported (for example, time-out for synchronous writing, ...). ▶ 22: The RGM data could not be accessed. ▶ 23: A new recipe group could not be created. ▶ 24: The variables or recipes of a recipe group cannot be imported correctly. ▶ 25: Group selection is missing for the import of an individual recipe. ▶ 26: The recipe group selected for the import of individual recipes is not present. ▶ 27: Error when creating variables in the group. ▶ 28: Error when importing the recipe group settings (comments, for example). ▶ 29: Error when importing the variable settings (the minimum and maximum value, for example). ▶ 30: The recipe group could not be imported. ▶ 31: The data of the selected recipe group for the import of individual recipes could not be loaded.
--	--	---

LAST RECIPE WRITTEN

In addition to the normal RGM actions, there are separate system driver variables for reading and writing for the graphical recipe variables (main.chm::/32418.htm). These system variables contain

identical values to `last_recipe_written` (write shadow variable) and `last_recipe_read` (read graphical recipe variable).

Variable	Data type	Description
Standard recipe RGM completely written (global/local) ----- ----- RGM write graphical recipe variable - result	UDINT	Variable receives the value 0 as long as a recipe is written locally. Value becomes 1 if the recipe has been completely written. (After starting the Runtime, the variable also has the value 0) Values for write operation. <ul style="list-style-type: none"> ▶ 0: Send initialization value before the recipe. ▶ 1: Writing was completed successfully and has ended. ▶ 2: Writing was not carried out due to a general error (parameter). ▶ 3: Writing was completed successfully and has ended. ▶ 4: Timeout occurred. ▶ 5: Write terminated, for example because RT is being ended. ▶ 6: time expired (time-out).
Last standard recipe/ RGM recipe name (global/local) ----- ----- RGM write graphical recipe variable - recipe name	STRING	Displays the name of the last standard recipe/RGM recipe sent globally or locally.
Last recipe group sent (globally/locally) ----- ----- RGM write graphical recipe variable - recipe group name	STRING	Displays the name of the last recipe group sent of the last recipe sent.
Last RGM recipe number sent (globally/locally) ----- ----- RGM write graphical recipe variable - recipe number	UDINT	Shows the recipe number of the last RGM recipe sent, globally or locally
RGM last recipe sent -	DINT	Authorization level of the last recipe written.

authorization (global/local) - - - - - - - - - - RGM write graphical recipe variable - authorization		
RGM last recipe sent - last user change (global/local) - - - - - - - - - - RGM write graphical recipe variable - operator last change	STRING	Name of the user who was logged in when the last change was made to the last recipe written.
RGM last recipe sent - comment 1 to 8 (global/local) - - - - - - - - - - RGM write graphical recipe variables - comment 1 to 8	STRING	Eight comments can be added to a recipe. The variable contains the comment of the respective number for the last recipe sent.
RGM last recipe sent - recipe status (global/local) - - - - - - - - - - RGM write graphical recipe variable - recipe state	DINT	Status of the last recipe written as a number.
RGM last recipe sent - recipe status text (global/local) - - - - - - - - - - RGM write graphical recipe variable - recipe state text	STRING	Status of the last recipe sent as text in 1 - @Text format.
RGM last recipe sent - recipe version (global/local)	DINT	Version of the last recipe written.

<pre> - - - - - - - - - - RGM write graphical recipe variable - recipe version </pre>		
<pre> RGM last recipe sent - time of last change (global/local) </pre>	STRING	Time of the last recipe change of the last recipe written.

LAST RECIPE READ

In addition to the normal RGM actions, there are separate system driver variables for reading and writing for the graphical recipe variables (main.chm::/32418.htm). These system variables contain identical values to `last recipe written` (write shadow variable) and `last recipe read` (read graphical recipe variable).

Variable	Data type	Description
<pre> RGM recipe of all values that have been read completely (global/local) - - - - - - - - - - RGM read graphical recipe variable - result </pre>	UDINT	<p>After the recipe has been written, this variable contains the result of the writing operation. Possible variable values:</p> <ul style="list-style-type: none"> ▶ 0: Set before the reading and only changes when the reading process is done. ▶ 1: Finished reading successfully. ▶ 2: During reading an error not defined in greater detail has occurred. ▶ 3: When reading, at least one variable has status INVALID (main.chm::/24148.htm). ▶ 4: At least one value is not within the min-max limits. ▶ 5: During reading a timeout (30000 + 100*VarCount in [ms]) occurred.
<pre> RGM last recipe read - authorization (global/local) - - - - - - - - - - RGM read graphical recipe variable - authorization </pre>	DINT	Level of authorization of the last recipe read.
<pre> RGM last recipe read - </pre>	STRING	Name of the user who was logged in when the last change

last user change (global/local) - - - - - - - - - - RGM read graphical recipe variable - operator last change		was made to the last recipe read.
RGM last recipe read - comment 1 to 8 (global/local) - - - - - - - - - - RGM read graphical recipe variables - comment 1 to 8	STRING	Eight comments can be added to a recipe. The variable contains the comment of the respective number for the last recipe read.
RGM last recipe read - recipe group name (global/local) - - - - - - - - - - RGM read graphical recipe variable - recipe group name	STRING	Name of the recipe group name of the recipe read last.
RGM last recipe read - recipe name (global/local) - - - - - - - - - - RGM read graphical recipe variable - recipe name	STRING	Name of the recipe read last.
RGM last recipe read - recipe number (global/local) - - - - - - - - - - RGM read graphical recipe variable - recipe number	DINT	Number of the recipe read last.

RGM last recipe read - recipe status (global/local) - - - - - - - - - - RGM read graphical recipe variable - recipe status	DINT	Status of the last recipe read as a number.
RGM last recipe read - recipe status text (global/local) - - - - - - - - - - RGM read graphical recipe variable - recipe state text	STRING	Status of the last recipe read as text in 1- Text format.
RGM last recipe read - recipe version (global/local) - - - - - - - - - - RGM read graphical recipe variable - recipe version	DINT	Version of the last recipe read.
RGM last recipe read - time of last change (global/local) - - - - - - - - - - RGM read graphical recipe variable - time last change	STRING	Time of the last recipe change of the last recipe read.

LAST RECIPE SELECTED

If several RGM screens are opened at the same time, the values of the last recipe selection are always taken.

Variable	Data type	Description
RGM last selected recipe - authorization	DINT	Contains the level of authorization of the last selected recipe as a number.
RGM last selected recipe - last user change	STRING	Contains the name of the user that made the last change to the last selected recipe.
RGM last selected recipe - group name	STRING	Contains the name of the group recipe of the last selected recipe.
RGM last selected recipe - comment 1 to 8	STRING	Eight comments can be added to a recipe. The variable contains the comment of the respective number for the last recipe selected.
RGM last selected recipe - recipe name	STRING	Contains the name of the last selected recipe.
RGM last selected recipe - recipe number	DINT	Contains the recipe number of the last selected recipe.
RGM last selected recipe - recipe status	DINT	Contains the status of the last selected recipe as a number.
RGM last selected recipe - recipe status text	STRING	Contains the status of the last selected recipe as text. Example: 1 - newly created
RGM last selected recipe - recipe version	DINT	Contains the recipe version of the last selected recipe.
RGM last selected recipe - time of last change	STRING	Contains the time at which the last selected recipe was last changed.

CHECK RECIPE VALUES

Variable	Data type	Description
RGM recipe values check - deviations	STRING	List all variable differences in the following order:

(global/local)		<p>[Variable name; recipe value; variable value; unit]</p> <p>each entry is written in a new line.</p> <p>Requirement: Variable RGM recipe value check - result has the value 1.</p> <p>Attention: Variable can either be created and evaluated globally or locally.</p>
Check RGM recipe values - authorization (global/local)	DINT	Authorization level of the last recipe checked.
Check RGM recipe values - last user change (global/local)	STRING	Name of the user who was logged in when the last change was made to the last recipe checked.
RGM recipe values check - result (global/local)	DINT	<p>Result of the check:</p> <ul style="list-style-type: none"> ▶ 0: All values match. ▶ 1: At least one variable value deviates from the recipe value. ▶ 2: At least one variable is faulty (INVALID). <p>Error messages:</p> <ul style="list-style-type: none"> ▶ -1: An error while reading the variable value occurred. ▶ -2: The recipe group could not be opened. ▶ -3: The recipe could not be changed. <p>Attention: Variable can either be created and evaluated globally or locally.</p>
Check RGM recipe values - comment 1 to 8 (global/local)	STRING	Eight comments can be added to a recipe. The variable contains the comment of the respective number for the last recipe checked.
Check RGM recipe values - recipe group name (global/local)	STRING	<p>Name of the recipe group of the recipe checked last.</p> <p>Requirement: Variable RGM recipe value check - result has a value ≥ -1.</p> <p>Attention: Variable can either be created and evaluated globally or locally.</p>
RGM recipe values check - recipe name (global/local)	STRING	<p>Name of the recipe checked last.</p> <p>Requirement: Variable RGM recipe value check - result</p>

		has a value ≥ -1 . Attention: Variable can either be created and evaluated globally or locally.
RGM recipe values check - recipe number (global/local)	UDINT	Number of the recipe checked last. Requirement: Variable RGM recipe value check - result has a value ≥ -1 . Attention: Variable can either be created and evaluated globally or locally.
Check RGM recipe values - recipe status (global/local)	DINT	Status of the last recipe checked as a number.
Check RGM recipe values - recipe status text (global/local)	STRING	Status of the last recipe checked as text in the format: 1 - @Text
Check RGM recipe values - recipe version (global/local)	DINT	Version of the last recipe checked.
Check RGM recipe values - time of last user change (global/local)	STRING	Time of the last recipe change of the last recipe checked.

RGM BEHAVIOR

All variables are saved locally on the client and assigned data at the time a recipe is selected. If values are changed after values have been selected, for example the recipe number or a comment, then this data is not displayed for the variables. This means: The variables represent a snapshot at the time the recipe is selected. It is therefore possible to establish what was changed after saving.

6.17 Topic - System information

The following system driver variables are available for this subject area:

Name	Data type	Comment
Hard disk data storage active/inactive	BOOL	Display whether the hard disk data storage of the variables is active in zenon. 0 = HDD administration is inactive 1 = HDD administration is active
Local system variable: DWORD 1-4	UDINT	These are local variables that are not updated in the network. The values for these variables can be configured in the zenon6.ini file, which means they can be used as constants for the local computer
Computer name	STRING	Shows the name of the local computer.
Runtime folder not available	BOOL	Checks whether the data directory of the Runtime is available. Upon a negative result, the Runtime will no longer try to perform read or write access on the directory. The ring buffers of the AML/CEL/HDD management will not be closed. The Runtime will no longer check the availability of the folder. As soon as the directory is available again, the Runtime will have to be restarted again.
Runtime status (simulation)	DINT	Displays the current status of the Runtime. The following values are available for the variable: 0 : Process connection 1: Switch to process connection in simulation 2: Change to simulation with process connection 3: In simulation 4: Change the simulation image during simulation
Time without operation	UDINT	Displays for how long no operation has been carried out in the Runtime. The time is displayed in seconds. The variable is updated cyclically. You can define the cycle time in the file <code>project.ini</code> . For example in order to update the variable every 5 seconds add <code>UPDATE_SEK=5</code> in the are [Systemtreiber].

6.18 Creating variables by importing

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



Information

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

6.18.1 XML import

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

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



Attention

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

Example:

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

6.18.2 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. The length can be limited using the MAX_LAENGE entry in project.ini .
KANAL_R	C	128	The original name of a variable that is to be replaced by the new name entered under "VARIABLENNAME" (field/column must be entered manually). The length can be limited using the MAX_LAENGE entry in project.ini .
KANAL_D	Log	1	The variable is deleted with the 1 entry (field/column has to be created by hand).
TAGNR	C	128	Identification. The length can be limited using the MAX_LAENGE entry in project.ini .
Unit	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 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 zenon (only if supported by the driver)
OBJEKT	N	2	Driver-specific ID number of the primitive object comprises TREIBER-OBJEKTYP and DATENTYP
SIGMIN	Float	16	Non-linearized signal - minimum (signal resolution)
SIGMAX	F	16	Non-linearized signal - maximum (signal resolution)
ANZMIN	F	16	Technical value - minimum (measuring range)
ANZMAX	F	16	Technical value - maximum (measuring range)
ANZKOMMA	N	1	Number of decimal places for the display of the values (measuring range)
UPDATERATE	F	19	Update rate for mathematics variables (in sec, one decimal possible) not used for all other variables
MENTIEFE	N	7	Only for compatibility reasons
HDRATE	F	19	HD update rate for historical values (in sec, one decimal possible)
HDTIEFE	N	7	HD entry depth for historical values (number)
NACHSORT	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. 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	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	hnical 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	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 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 (if "-1" is entered here, the existing function is not overwritten during import)
A_GRUPPE1	N	10	Alarm/event group
A_KLASSE1	N	10	Alarm/event class
MIN_MAX1	C	3	Minimum, Maximum
FARBE1	N	10	Color as Windows coding
GRENZTXT1	C	66	Limit value text
A_DELAY1	N	10	Time delay
INVISIBLE1	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. Driver-specific functions

The driver supports the following functions:

ZENON6.INI ENTRIES

Parameters	Description
[SYSTEMTREIBER]	
UPDATE_SEK = ...	Cycle time of the system driver in seconds
[LOCAL_VAR]	
ID_DWORD_1 = <value> ID_DWORD_2 = <value> ID_DWORD_3 = <value> ID_DWORD_4 = <value>	<p>These entries deliver the values for the system driver variables "System info -> Local system variable: DWORD1-4".</p> <p>These variables are intended for the identification of a single computer in the network. This means the values are always managed locally in the zenon.ini file.</p> <p>By default, limits can be defined for each of these variables.</p> <p>If there are no entries in the zenon.ini file, the values of these variables will be set to 0 in zenon in Runtime.</p>

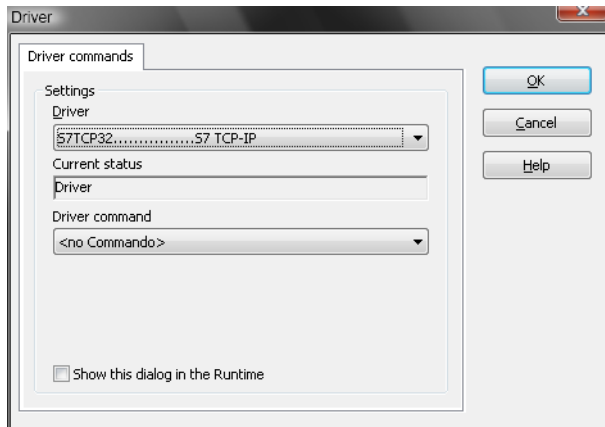
8. Driver commands

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

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

- create a new function

- ▶ select *Variables -> Driver commands*
- ▶ The dialog for configuration is opened



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. 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. The values of all variables of the driver are simulated by the driver. No values from the connected hardware (e.g. PLC, bus system, ...) are displayed.
▶ Driver in hardware mode	Driver is set into hardware mode. For the variables of the driver the values from the connected hardware (e.g. PLC, bus system, ...) are displayed.
▶ Driver-specific command	Enter driver-specific commands. Opens input field in order to enter a command.
▶ 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.

9. Error analysis

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

9.1 Analysis tool

All zenon modules such as Editor, Runtime, drivers, etc. write messages to a joint log file. To display them correctly and clearly, use the Diagnosis Viewer (main.chm::/12464.htm) program that was also installed with zenon. You can find it under *Start/All programs/zenon/Tools 7.11 -> 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\zenon7.11\LOG for zenon Version 7.11. 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**.



Information

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