



COPADATA
do it your way

zenon manual

zenon Logic

v.8.00





©2018 Ing. Punzenberger COPA-DATA GmbH

All rights reserved.

Distribution and/or reproduction of this document or parts thereof in any form are permitted solely with the written permission of the company COPA-DATA. Technical data is only used for product description and are not guaranteed qualities in the legal sense. Subject to change, technical or otherwise.

Contents

1. Welcome to COPA-DATA help	6
2. zenon Logic	6
3. Basics zenon Logic Workbench (IEC 61131-3)	7
3.1 General information on zenon Logic help	7
3.2 zenon Logic General	8
3.3 zenon Logic toolbar and context menu detail view	9
3.4 First steps with zenon Logic in zenon	9
3.4.1 Create a zenon Logic project in a zenon project	10
3.4.2 Editing a zenon Logic project	14
3.4.3 Deleting a zenon Logic project	14
3.4.4 Design of the Workbench	15
3.4.5 Show license information	16
3.5 Variable handling	17
3.5.1 Create O&M variables in zenon Logic	17
3.5.2 Create O&M variables in zenon	21
3.5.3 Using zenon variables in zenon Logic	22
3.5.4 Variable names	24
3.6 Arrays	25
3.7 Data structures	25
3.8 Distributed engineering (multiuser)	26
3.9 Reusing projects	27
4. zenon Logic Workbench	27
5. zenon Logic Runtime (Soft PLC)	27
5.1 User interface	29
5.1.1 Slider Allowed maximum	30
5.1.2 Settings	30
5.1.3 Messages	37
5.2 Licensing	37
5.2.1 TAG handling zenon - zenon Logic	38

5.2.2	Demo mode	39
5.3	Start the zenon Logic Runtime	39
5.4	zenon Logic Runtime cycle	46
5.4.1	Error message cycle time	48
5.5	Close the zenon Logic Runtime	49
5.6	Creating the hot restart file	50
5.7	Remote systems	50
5.7.1	zenon Runtime and zenon Logic Runtime on one remote PC	50
5.8	zenon Logic Runtime with Real Time Kernel (RTK)	51
5.8.1	Restrictions and requirements of the zenon Logic RTK	55
5.8.2	zenon Logic RTK configuration	56
5.8.3	Blue Screen Handling.....	63
5.8.4	Error codes at starting the zenon Logic RTK	64
5.9	Redundancy zenon Logic and zenon	67
5.9.1	Requirements	67
5.9.2	Configure redundancy	69
5.10	OEM specification	72
5.10.1	Support for the printf function.....	75
5.11	Forwarding time stamps and the variable status information.....	75
5.12	zenon Logic I/O driver support	76
5.13	zenon Logic I/O driver: Hot-Restart/Online Change	78
5.14	Error message	79
6.	zenon Logic Runtime Manager	82
6.1	Why should you use the zenon Logic Runtime Manager?	82
6.2	Main menu	83
6.3	Properties help.....	83
6.3.1	Configuration list	83
6.3.2	zenon Logic project folder	83
6.3.3	Name of the configuration	84
6.3.4	Port (Main)	84
6.3.5	Port (Binding)	84
6.3.6	Startup.....	85
6.3.7	Start in step mode	85
6.3.8	Windows real time priority.....	86
6.3.9	Hard real time.....	86

6.3.10	Report output.....	86
6.3.11	Auto start.....	87
6.3.12	Button "New"	87
6.3.13	Button "Edit"	87
6.3.14	Button "Save"	88
6.3.15	"Delete" button	88
6.3.16	Button "Start"	88
6.3.17	zenon Logic Runtime (settings)	88
6.3.18	Language selection (languages).....	88
6.3.19	Show splash screen (setting)	88

1. Welcome to COPA-DATA help

ZENON VIDEO-TUTORIALS

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

GENERAL HELP

If you cannot find any information you require in this help chapter or can think of anything that you would like added, please send an email to documentation@copadata.com.

PROJECT SUPPORT

You can receive support for any real project you may have from our Support Team, who you can contact via email at support@copadata.com.

LICENSES AND MODULES

If you find that you need other modules or licenses, our staff will be happy to help you. Email sales@copadata.com.

2. zenon Logic

zenon Logic is the programming environment integrated into zenon in accordance with IEC 61131. It is available as Editor (on page 7) and Runtime (Soft-PLC) (on page 27).

3. Basics zenon Logic Workbench (IEC 61131-3)

zenon Logic is an IEC 61131-3 programming environment for different target systems with zenon Logic Runtime kernel. In the case of zenon, the zenon Logic Runtime kernel is designed as a soft PLC for PC and CE platforms.

With the zenon Logic development environment - the Workbench - PLCs can be engineered and programmed in the five defined programming languages of IEC 61131-3.

PROJECT MANAGER CONTEXT MENU

Menu item	Action
New zenon Logic project	Opens the assistant to create a new zenon Logic project.
Import XML...	Imports entries from an XML file. <i>Attention:</i> At the XML import you must deactivate online change so that the reload of the zenon Logic variable works.
Import external zenon Logic project...	Opens the dialog to select a zenon Logic project folder.
Editor profiles	Opens the context menu with pre-defined Editor profiles.
Help	Opens online help.

3.1 General information on zenon Logic help

This help manual provides a quick introduction to the use of zenon Logic in conjunction with the integrated solution with zenon. This manual does not provide an instruction into computer languages, functions and function blocks. For this use the extensive help included directly in the zenon Logic Workbench> (Main.chm::/K5HELP.chm::/K5HELP.htm) (IEC 61131-3 programming environment).

The zenon Logic Workbench provides an extensive help functionality. In order to access this help function, open the entry **Help** in the menu of the zenon Logic Workbench or click here (Main.chm::/K5HELP.chm::/K5HELP.htm). In order to access a help chapter about a certain topic (e.g. function block), select it with a mouse click and then press **F1**.

The in zenon integrated zenon Logic Workbench has not TAG limitation independent of I/O or TAG extension of the zenon Editor.

3.2 zenon Logic General

zenon Logic is an IEC 61131-3 programming environment for different target systems with zenon Logic Runtime kernel. In the case of zenon, the zenon Logic Runtime kernel is designed as a soft PLC for PC and CE platforms.

ZENON LOGIC WORKBENCH

The zenon Logic Workbench is the tool for programming the zenon Logic PLC. The zenon Logic Workbench is started in the zenon Editor. All five IEC 61131-3 predefined languages are available:

Abbreviation	Long form of the language
SFC	Sequential Function Chart - AS
FBD	Function Block Diagram - FUP
LD	Ladder Diagram - KOP
ST	Structured Text - ST
IL	Instruction List - AWL

IEC 61131-3

Part three of the IEC 61131 describes the syntax and semantic of computer languages for PLCs which have been defined in part 1 of the IEC 61131.

ZENON LOGIC RUNTIME

The zenon Logic Runtime is the target system and executes the compiled code of the zenon Logic Workbench. There is the zenon Logic Soft PLC for the PC and for Windows CE platforms, as well as selected embedded platforms.



Information

You can find more information on platforms here:
www.copadata.com/en/products/zenon-logic.html
 (<https://www.copadata.com/en/hmi-scada-solutions/zenon-logic/>).

You can find more information about the operation of zenon Logic Runtime in the zenon Logic Runtime (soft PLC) (on page 27) manual.

STRATON RUNTIME

In addition to the zenon Logic soft PLC, there is also the hardware PLC straton, which has implemented the zenon Logic runtime kernel as OEM software. The manufacturers of these hardware PLCs can be found on www.straton-plc.com (<http://www.straton-plc.com>).

3.3 zenon Logic toolbar and context menu detail view



TOOL BAR

Parameters	Description
New zenon Logic project	Opens the dialog to create a new project (on page 10), create it in the list and opens the zenon Logic Workbench with the new project.
Open zenon Logic project in Workbench	Opens the selected project in zenon Logic Workbench.
Delete	Deletes selected entries after a confirmation from list.
Import external zenon Logic project	Opens dialog to select the zenon Logic project files to be imported.
Import XML	Opens dialog to select the XML file to be imported. <i>Attention:</i> At the XML import you must deactivate online change so that the reload of the zenon Logic variable works.
Remove all filters	Removes all filter settings.
Edit selected cell	Opens the selected cell for editing. The binocular symbol in the header shows which cell has been selected in a highlighted line. Only cells that can be edited can be selected.
Replace text in selected column	Opens the dialog for searching and replacing texts.
Properties	Opens the Properties window.
Help	Opens online help.

3.4 First steps with zenon Logic in zenon

Here you can find out how you create, configure and edit a zenon Logic project in the zenon Editor.

3.4.1 Create a zenon Logic project in a zenon project

To create a new zenon Logic project:

1. right click on node **zenon Logic (IEC 61131-3)** or in the detail view
2. select in the context menu or in the toolbar command **New zenon Logic project...**
3. The configuration dialog is opened
4. give it a valid name
5. select the desired driver
 - straton driver are automatically configured
 - for IEC driver the configuration dialog is opened after the dialog is closed
6. create the project by clicking on **Finish**



Information

You adjust the properties of the zenon Logic projects in the zenon project in the properties window of the zenon Editor. You will find a description of the single properties in the property help.

You can change the settings of the zenon Logic project in its project settings. In order to do this open the project settings in the Workbench under **Project/Settings....**

NOTES FOR PROJECT CONFIGURATION

Note when configuring the project:

General:

- ▶ In the zenon Editor, no zenon Logic projects with the name **__Global** can be created.
Note: Projects with the name **__Global** that already exist are not shown.
- ▶ zenon Logic Workbench creates, for each zenon Logic project, a project with the name **<Projectname>__Global**. These projects are not visible in the zenon Editor. They are also not deleted if the attendant zenon Logic project is deleted.

Distributed engineering:

- ▶ In the zenon Logic Workbench, lines can be created in the **All projects** node **MultiSpy** and the global **Binding** Editor can be called up. The following is applicable for these settings: They are
 - Not saved as multi-user
 - Not transferred to the project server

- Removed again in the next synchronization
- ▶ If new zenon Logic projects are created in the zenon Logic Workbench via File -> Add new project, the following is applicable: They are
 - Not transferred to the project server
 - Deleted again in the next synchronization

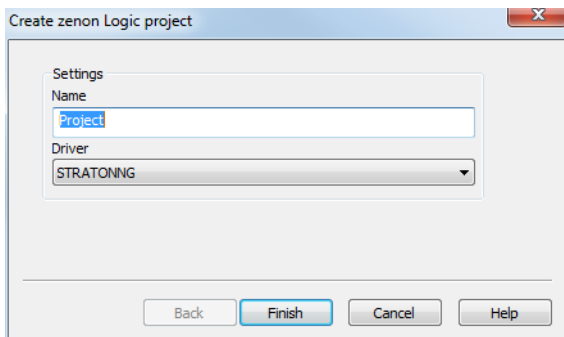
Attention: Multi-user zenon Logic projects can only be created using the zenon Editor.



Information

In general you should only create zenon Logic projects with the zenon Editor. This is because the attendant drivers are also automatically created in the zenon project.

NEW DIALOG PROJECT



Create zenon Logic project

Settings

Name
project

Driver
STRATONING

Back Finish Cancel Help

Parameter	Description
Name	<p>Freely definable name for the zenon Logic project.</p> <p>Note: The Name</p> <ul style="list-style-type: none"> ▸ consists of alphabetic characters ▸ must not exceed 15 characters ▸ must not be empty ▸ must only contain valid characters: invalid characters are: all numbers and special characters such as \ / , ; ß ú ý : * ? " < > () a dot may only be used within the name not at the beginning of the name <p>Button Finish is only active if a valid name was entered.</p>
Drivers	<p>Selection of the driver. You have the choice between:</p> <ul style="list-style-type: none"> ▸ stratonNG (Default) creates a new connection with the correct project name, the zenon Logic Runtime port and the IP address 127.0.0.1 ▸ straton32: creates a new connection with the zenon Logic Runtime port and the IP address 127.0.0.1 ▸ IEC870: opens configuration dialog ▸ IRC850: opens configuration dialog <p>This selection defines the eventual Runtime communication between zenon Logic and zenon. In addition the creation of the operating and monitoring (O and M) variables (on page 17) is based on this:</p> <ul style="list-style-type: none"> ▸ name based at the STRATON driver, ▸ address based at the IEC driver
Finish	Creates the new zenon Logic project.
Cancel	Cancels project creation
Help	Opens online help.

DRIVER SELECTION

Decision aid for the driver selection:

Drivers	Properties
stratonNG driver:	Default driver for zenon Logic projects: <ul style="list-style-type: none"> ▶ can be used cyclic or spontaneous (on change) ▶ supports complex data types (structure data types or arrays) ▶ enables several connections with one driver to several Runtimes at the same time (recommended for using with Windows CE))
straton32 driver:	Ethernet TCP/IP based. Note: Only included for compatibility reasons. Use of the stratonNG driver is recommended.
IEC870 driver:	It is mostly used in the infrastructure automation. Properties: <ul style="list-style-type: none"> ▶ can be operated serial (-101) or over Ethernet TCP/IP (-104) ▶ Data traffic is generally spontaneous (on change) ▶ complex data types are not supported
IEC850 driver:	Successor of the IEC60870 standard and is completely object-orientated.

For information about the configuration of the driver see the respective help of the driver:

- ▶ straton32 driver: straton32 (straton32.chm::/STRATON32.htm)
- ▶ stratonNG driver: stratonNG (stratonNG.chm::/stratonNG.htm)
- ▶ IEC870 driver: IEC870 (IEC870.chm::/IEC870.htm)
- ▶ IEC850 driver: IEC850 (IEC850.chm::/IEC850.htm)



Attention

If you have selected a driver once, you cannot change it afterwards.

Exception: Changing from the **straton32** to the **stratonNG** driver is not possible.

CHANGE DRIVER

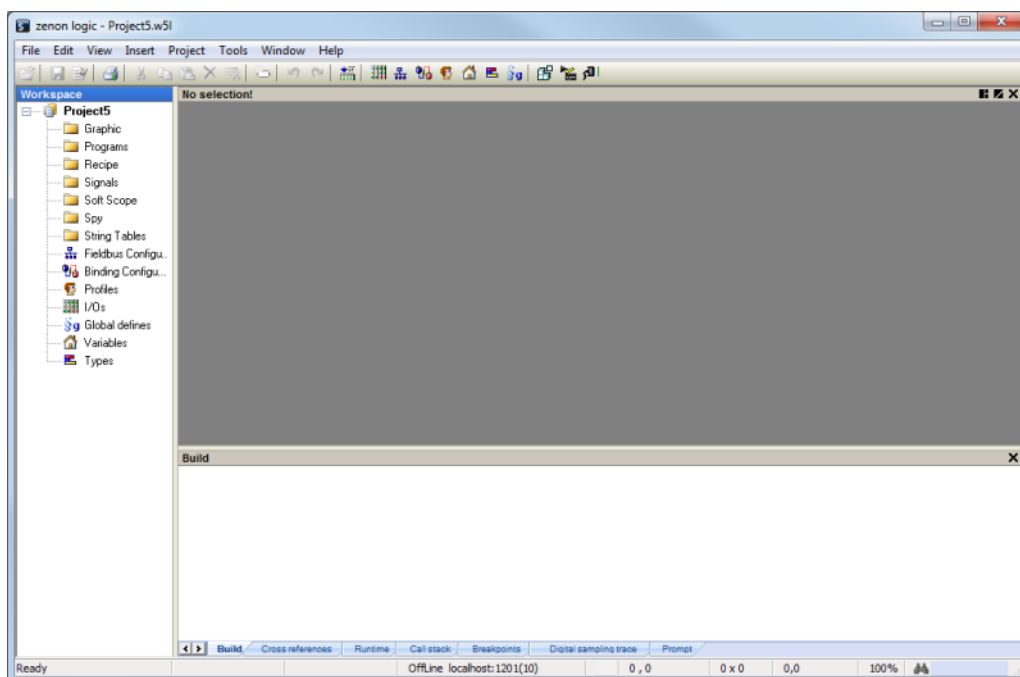
*To make a driver change from **straton32** to **stratonNG**:*

- ▶ In the zenon Editor, select the project for which you want to replace the **straton32** driver.
- ▶ In the detail view, go to the **Variables/Driver** node.
- ▶ Select the **Driver** entry there.
- ▶ In the detail view of the project manager, select the **straton32** driver.
- ▶ select **Change driver** in the context menu

- ▶ The dialog for selecting the driver is opened.
- ▶ Select the driver.
- ▶ Close the dialog.
The configuration is applied to the zenon Logic project.

3.4.2 Editing a zenon Logic project

With a double click on the project or the context menu, you can open an existing zenon Logic project for editing. Alternatively you can click the corresponding icon in the toolbar in the details window.



3.4.3 Deleting a zenon Logic project

To delete a zenon Logic project:

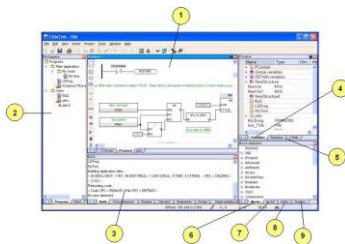
1. select menu item delete in the context menu of the project or in the toolbar
(Multi-select is possible)
2. confirm the confirmation message whether you really want to delete the project
3. select whether you want to delete the driver of the project
(When simultaneously deleting several projects, the deleting the driver is prompted for each project.)

**Attention**

When you delete the driver, all variables linked to the driver in zenon are also deleted.

3.4.4 Design of the Workbench

The zenon Logic Workbench can be divided in the following areas:



Range	Description
1	In this section documents are displayed and edited (programs, watch lists, ...).
2	The workspace: Documents are created and opened in this list.
3	The output window displays messages and provides diagnosis tools.
4	In the variable editor variables and instances are declared.
5	The properties tab is used for the configuration of the graphics documents.
6	The tab Libraries lists all available functions and function blocks
7	The tab Spy list displays the online values of selected variables during the debugging.
8	The tab Definitions displays all definitions.
9	The tab Graphics lists all graphical objects.

Each section (except section 1) can be shown and hidden with commands from the menu **Views**.

In the central area you will find the documents of the workspace. If several documents are opened at the same time, you can switch between the different documents using the tab at the bottom. With button **X** in the header the active document can be closed.

The variable editor and the document area in the center can be maximized at any time with the corresponding button (blue symbol with white line) or a double-click on the header.

If several documents are open in the central area, one of these can be fixed in the top or left part. In order to do this right click the corresponding tab and select **Fix**. With the same menu you can unfix the document at any time and fix another one:

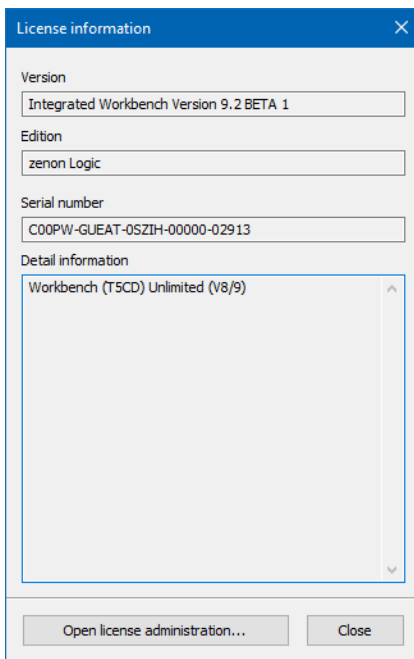
3.4.5 Show license information

To display license information in zenon Logic Workbench:

1. Open the menu **Help**.
2. Click on **About...** .

The license information dialog is opened.

LICENSE INFORMATION DIALOG



Parameters/buttons	Description
Version	Version number of the integrated workbench.
Edition	Edition used.
Serial number	License serial number.
Detail information	Shows details on the license.
Open license administration	Opens the License Manager . Licenses can be activated, returned and managed with this tool. The serial number is needed for this.
Close	Closes the dialog.

3.5 Variable handling

The integration of zenon Logic into zenon offers considerable advantages in the configuration of PLC and SCADA projects. The O&M variables (operating and monitoring) from the PLC system are available in zenon with just a few mouse clicks. When changing the variable (name, datatype, ...) these changes are immediately passed on the other system.

The advantage of the integration:

- ▶ Easy handling of O&M variables
- ▶ Immediate alignment of changes
- ▶ Saving of labor because of omission of import/export
- ▶ Increased quality by minimizing causes for errors



Attention

Instances of UDFBs are not available as B&B variables.

3.5.1 Create O&M variables in zenon Logic

All variables created in the zenon Logic Workbench can be marked as O&M (operating and monitoring) variables:

- ▶ Local program variables
- ▶ Global variables
- ▶ Retain variables (=remanent variables)
- ▶ Array variables
- ▶ Structure variables

Note: Variables of data type `TIME` cannot be declared as O&M variables.

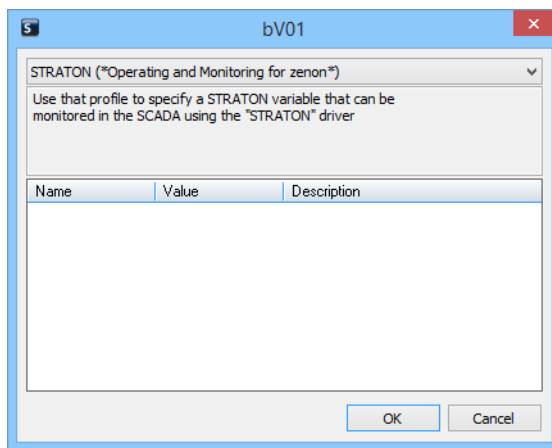
HIGHLIGHT VARIABLES AS O&M

In order to do this the following steps are necessary:

(Refer to the online help of the zenon Logic Workbench for the exact steps to create a variable in zenon Logic.)

1. Right click on a variable in the variable list.
2. Select **Properties** from the context menu.

3. In the drop-down list, select STRATON (*Operating and Monitoring for zenon*).
4. Click on **OK**.
A request for confirmation is opened.
5. Confirm the dialog by clicking on **Yes**.
6. Activate the checkbox of the variable in the `Symb` column.



Now the O&M variable has been entered in the zenon variable list.



Attention

As soon as a zenon Logic variable is declared as O&M variable, you must not edit it as text in the Workbench!

This leads to the loss of variable declarations.

The O&M variable is displayed with a prefix in zenon. The prefix has the following structure:

[zenon Logic project name >/zenon Logic program name or Global or Retain]/[variable name]

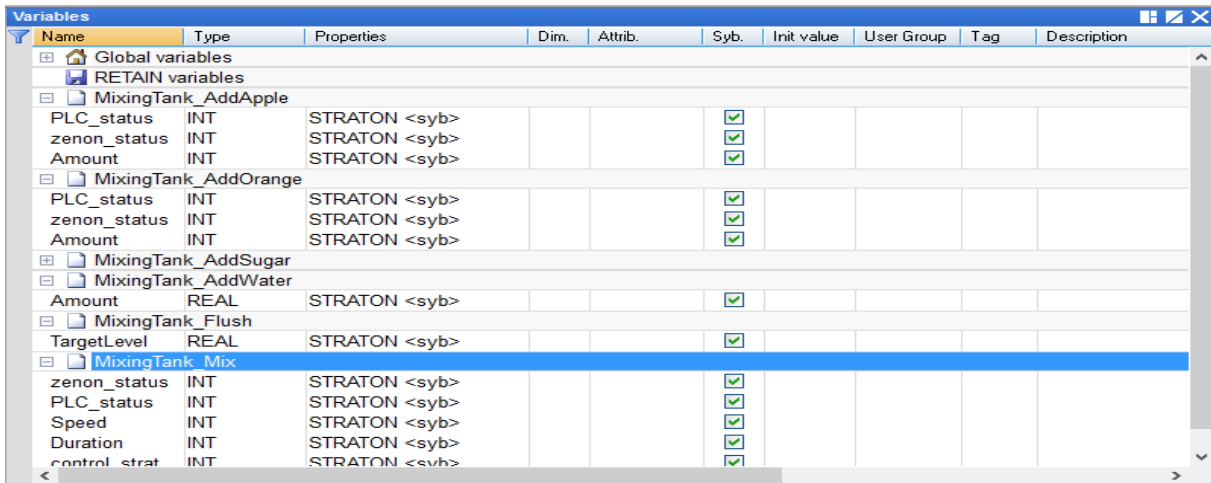


Attention

Variables have a character length limitation. Including prefix the variable name must not be longer than 128 characters.

ACTIVATE IN VARIABLE LIST

You can update the profile and the checkbox directly in the variable list of zenon Logic so you do not have to go to the context menu and the profile dialog.



Name	Type	Properties	Dim.	Attrib.	Syb.	Init value	User Group	Tag	Description
Global variables									
RETAIN variables									
MixingTank_AddApple									
PLC_status	INT	STRATON <syb>			<input checked="" type="checkbox"/>				
zenon_status	INT	STRATON <syb>			<input checked="" type="checkbox"/>				
Amount	INT	STRATON <syb>			<input checked="" type="checkbox"/>				
MixingTank_AddOrange									
PLC_status	INT	STRATON <syb>			<input checked="" type="checkbox"/>				
zenon_status	INT	STRATON <syb>			<input checked="" type="checkbox"/>				
Amount	INT	STRATON <syb>			<input checked="" type="checkbox"/>				
MixingTank_AddSugar									
MixingTank_AddWater									
Amount	REAL	STRATON <syb>			<input checked="" type="checkbox"/>				
MixingTank_Flush									
TargetLevel	REAL	STRATON <syb>			<input checked="" type="checkbox"/>				
MixingTank_Mix									
zenon_status	INT	STRATON <syb>			<input checked="" type="checkbox"/>				
PLC_status	INT	STRATON <syb>			<input checked="" type="checkbox"/>				
Speed	INT	STRATON <syb>			<input checked="" type="checkbox"/>				
Duration	INT	STRATON <syb>			<input checked="" type="checkbox"/>				
control_strat	INT	STRATON <syb>			<input checked="" type="checkbox"/>				

Mark the line **Properties** of the corresponding variable. Press **Ctrl+F12**. Thus the profile zenon Logic is displayed in this line. In addition you must activate the checkbox for embedding the symbol via double click or **Enter** in column **Symb.**.



Attention

The checkbox **Embed symbol** must be activated. Variables for which this checkbox is not active are not able to communicate with zenon using the **STARTON** driver and will not supply any values.

ADDITIONAL ATTRIBUTES OF THE VARIABLE

The additional attributes **Identification** and **Description** of a variable are also displayed in zenon when you select the profile **STRATON** (*Operating and Monitoring for zenon*). You can find the content of the identification and description in zenon in the **Identification** and **Resources label** variable properties.

ARRAYS AND STRUCTURE VARIABLES

If arrays or structure variables are marked with the profile **STRATON** (*Operating and Monitoring for zenon*), they are also displayed in the variable list in zenon. You must however activate the desired elements of the variable.



Attention

If a O&M variable is an array, you must take care when setting initial values.

Recommendation: Arrays with initial values should not be declared as O&M! If it is inevitable, the string describing the initial value must not be longer than 1024 characters.

UDFB INSTANCES

1. Input variables at UDFB instances are not available via the integrated solution in zenon.
2. In zenon variables of an UDFB instance cannot be created. This is only possible if an instance of the UDFB is created in zenon Logic and the zenon Logic profile is set.

UDFB instances which are marked with profile zenon Logic (*Operating and Monitoring for zenon*), they are not automatically available in zenon. In case you want them to be available, you must activate this functionality manually with the help of the following entry in file K5DBXS.INI. In file K5DBXS.INI scroll to section [XS] and enter ShareUDFBDatatype=1 .

If you deactivate this entry, existing UDFB data types in a project are still modified. However no new UDFBs are created.

HARDWARE IO DECLARATIONS

zenon Logic IO variables can be used zenon if in the zenon Logic Workbench they:

- ▶ have a profile
- ▶ have an alias name; this name is used in zenon instead of %IX0.0 (board type, board index, bit number)
An alias is not necessary if the zenon Logic profile is used and property **Embed symbol** was activated. In this case the variable is also available in zenon (Editor and Runtime).

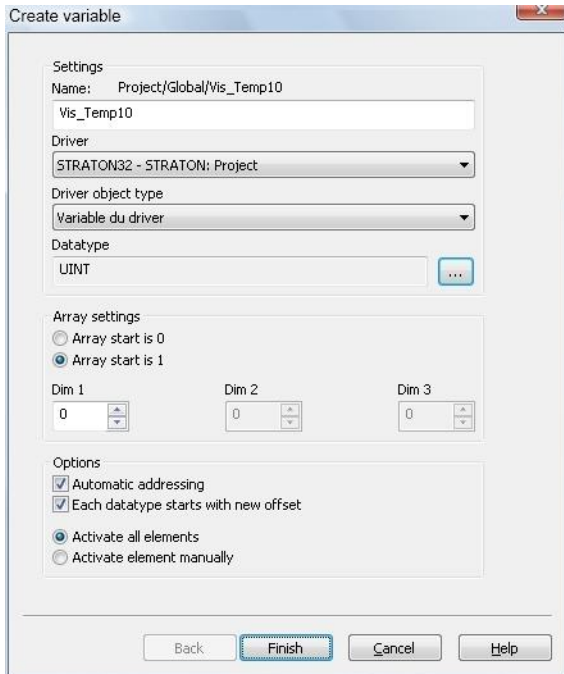
Note: If an I/O group is deleted in zenon Logic, the variables of the I/O board with zenon Logic profile are also deleted in zenon.

Attention:

- ▶ IO variables cannot be renamed in zenon.
- ▶ IO variables for zenon Logic IO cards cannot be created in zenon.
- ▶ The data type of IO variables should not be changed in zenon: Changes are not adopted in the zenon Logic Workbench.
- ▶ Changes to the hardware IO declaration should always be carried out in zenon Logic. In zenon no new variable for the straton32 driver can be created as hardware I/O variable for zenon Logic.

3.5.2 Create O&M variables in zenon

O&M (operating and monitoring) variables for the soft PLC zenon Logic can also be declared in zenon. In order to do this create a variable in zenon as usual:




Attention

Take care that the naming convention is IEC 61131-3 compatible (no special characters, ...); otherwise the button **Finish** will not be available in the dialog. See also: Variable names (on page 24)



Information

The variables declared in zenon are always entered in the Global area of the variable list in zenon Logic.

CHANGES TO VARIABLES OF A ZENON LOGIC DRIVER IN ZENON

If variables of a driver which is linked to a zenon Logic project are modified:

- ▶ a change recognition in the zenon Logic project is carried out
- ▶ the zenon Logic Workbench connection to the Runtime asks whether the project should be recompiled

This behavior is regardless of whether the changes affects the zenon Logic project.

3.5.3 Using zenon variables in zenon Logic

It is very easy to further process variables from PLC systems such as e.g. Siemens S7 or Beckhoff TwinCAT in zenon Logic, variables which were read in by the according zenon drivers in zenon.

There are two possibilities available for this:

- ▶ **Allocation:**
The variable from the PLC (e.g. S7) is linked to a zenon Logic variable with an allocation. In this case only a unidirectional communication is possible.
- ▶ **Logic to SCADA connection:**
The PLC variables (e.g. from a S7) are exchanged between zenon and zenon Logic using a **Logic to SCADA connection**. In order to switch the PLC variables in zenon Logic Workbench to visible, you must activate the **Externally visible** property in the properties of the variables. The variable is thus included in the variable list of zenon Logic Workbench.
Advantage: This process works in two directions. It can also be configured very quickly and easily.
Note: Such a connection only works locally on a PC or CE device. Here it is not possible to communicate with a zenon Logic Runtime on another device.

LOGIC TO SCADA CONNECTION

A **Logic to SCADA connection** variable is specially marked in zenon Logic, because it can no longer be changed in zenon Logic. Changing the variable here would lead to problems, as the source of the variable is the PLC programming environment in the external PLC (e.g. Step7).

In order for communication with Runtime for **Logic to SCADA connection** variables to also take place, an I/O driver - the **Logic to SCADA connection** driver - must also be created in zenon Logic.

Configuration:

1. Opens the field bus configuration in zenon Logic Workbench.
To do this, double click on the **field bus configuration** module or click on the symbol in the toolbar.
2. Right-click in the I/O driver area.
3. Select **Add configuration** in the context menu.
4. In the **All** group, select the **Logic to SCADA connection** entry.
This configuration is entered and displayed.

It is thus ensured that the externally-visible variables from zenon communicate with zenon Logic.



Attention

Note when choosing the variable name:

The name of the variable from an external PLC must comply with IEC 61131-3 with regard to how it is written.

Allowed are:

- ▶ Numbers
- ▶ Letters from A to Z (no umlauts)
- ▶ Underscores

In addition the name must not start with a number.
(for details, see also variable names (on page 24).)

Tip: To use a variable with non-permitted characters in zenon Logic, the name can be set in the code directly in curly brackets.

Example: The variable **myArray[3]** is called up in the program with **{myArray[3]}**.

POSSIBLE INTERACTIONS

*Note for **Logic to SCADA connections**, possible interactions with the **Hysteresis** property.*

*If a **Hysteresis** was configured for the zenon variable, the value in zenon Logic does not match the value in the PLC. It is also possible that it does not match the value in zenon.*

The following rules apply:

- ▶ The value changes in the PLC and within the configured hysteresis bandwidth:
The value is not communicated to zenon Runtime and therefore also not to zenon Logic.
- ▶ The value is changed by the user in zenon (by means of a function for example) and the value remains within the configured hysteresis bandwidth:
The value remains the same in zenon and in zenon Logic but it changes in the PLC.
- ▶ The value is changed in zenon Logic and remains within the configured hysteresis:
The value changes in the PLC and in zenon Logic, but remains unchanged in zenon.
If in this situation the value changes in the PLC and stays within the hysteresis, three different values exist.
One in the PLC, one in zenon and one in zenon Logic.



Attention

Events are only triggered in the event of value changes. In special groups of applications, there may be inconsistencies between values.

This can be avoided very easily in that different variables are used for reading and writing.

3.5.4 Variable names

In zenon Logic variable names must be in accordance with the IEC 61131-3 regulations. The correctness of the entry is checked by zenon as well as by zenon Logic (communication take place using the STRATON driver).

In contrast, for variables of other drivers (e.g. S7 TCP/IP driver), the person configuring the project has to ensure that the variable name corresponds to the conventions of IEC 61131-3 (communication using **Logic to SCADA connection**).

EXAMPLES OF PERMITTED AND NON-PERMITTED IEC 61131-3 WAYS OF WRITING

Permitted name	Impermissible name	Invalid because:
_XYZ	__XYZ	Two underscores.
xy_z	xy__z	Two underscores.
xyz_	xyz__	Two underscores.
MotorOn	Motor On	Space.
Motor_On	Motor/On	Slash.
mw10	1mw10	Number at the beginning.
xyz	#xyz	Hash.
_Motor1	_Motor 1	Space.
Stoerung	Störung	Umlaut.
Ueberbruecken	Überbrücken	Umlaut.
Mo_tor	Mo-tor	Hyphen.

From version 6.22 SP1 on it is possible to use variables which are not conform with the IEC 61131-3 regulations (free variable names). Take care when using free variable names.

Follow these rules when using free variable names:

- ▶ The following characters are not allowed to be part of a variable name: '(*, *)', '//', '{' and '}'
- ▶ Variable names are not allowed to start with '__' (two underscores)
- ▶ If free variable names are used when using programming language ST or IL, they must be between curly brackets {}.
- ▶ If free variable names are used with structure of array, they must be between curly brackets {}.
- ▶ Variables with free variable names with a basis data type can be used without curly braces {} in programming languages FBD and LD. They must be declared as **global** variables.
- ▶ Variables with the same name once with {} and once without {} are regarded as two different variables.



Information

Structure variables and arrays of structure variables with dot (".") in the name are displayed correctly. Standard arrays with slash ("/") or dot (".") in the name do not display values in the **spy list**.

3.6 Arrays

Arrays can be created in zenon and zenon Logic for both systems. The declaration as O&M variable takes place as described in chapters Creating O and M variables in zenon Logic (on page 17) and Creating O and M variables in zenon (on page 21). Take care when creating arrays in zenon Logic that you must activate the single elements in zenon.

When creating arrays in zenon, set the array start to 0. Thus the arrays have the same indices in both systems.



Attention

Arrays with initial values should not be declared as O&M! If it is inevitable, the string describing the initial value must not be longer than 255 characters.

3.7 Data structures

Structure datatypes can be created in zenon or in the zenon Logic Workbench.



Information

When creating structure data types in zenon the IEC 61131-3 naming convention has to be used otherwise the structure data type is not available in zenon Logic. Also see Variable names (on page 24). Here the spelling is not checked by zenon!



Information

If you create structure data types in zenon Logic, they are always declared as linked in zenon.

Therefore it is better to create a structure data type in zenon because then you can decide whether you want a linked or an embedded data type.

3.8 Distributed engineering (multiuser)

Several people configuring a project can work together on one zenon project. Thus for example one project engineer works on a zenon Logic project and another works on the zenon project.

VARIABLES

In distributed engineering O&M variables (operating and monitoring) are handled separately by the zenon Logic project.

You can create variables in zenon Logic as usual. If the variables should also be available in zenon, they have to be equipped with the corresponding property (element group, profile **STRATON** (*Operating and Monitoring for **zenon***)). After that the variable is created in zenon. You can check it in for other engineers (**apply changes**). You cannot edit a checked in variable in the zenon Logic Workbench. It is marked accordingly in the zenon Logic variable list.

DRIVERS

STRATON drivers are attended separately in distributed engineering by the zenon Logic project. That means if you want to make changes, you have to check out the STRATON driver (**enable changes**).

ZENON LOGIC PROJECTS

In order to edit a zenon Logic project in a multi-user zenon project, the zenon Logic project must be checked out (**Check out**). After the engineering, the changes can be accepted or undone.

DELETE ZENON LOGIC PROJECT

In order to delete zenon Logic projects, it must be checked out (**check out**) or edited in offline mode.

3.9 Reusing projects

zenon Logic projects can easily be integrated into a zenon project:

1. Select, in the context menu of the detail view of the **zenon Logic (IEC 61131-3)** module, **External zenon Logic Import project....**
2. Enter an existing zenon Logic project (highlight folder).
3. Click on **OK** to integrate the existing zenon Logic project into the zenon project.
4. Define a name for the imported project.
5. Select a driver for the integrated solution.

Another possibility is the XML import:

1. In the context menu of the detail view of the **zenon Logic (IEC 61131-3)** module, click on **Import XML**.
2. Select the desired XML file.
3. Click on **OK** to import the zenon Logic project.
4. Create a fixed name for the imported project.
5. Select a driver for the integrated solution.

Note: At the XML import you must deactivate **online change** so that the reload of the zenon Logic variable works.

4. zenon Logic Workbench

In this area you can find the manual for the zenon Logic Workbench. It is only available as online help.

5. zenon Logic Runtime (Soft PLC)

zenon Logic Runtime interprets the compiled PLC Code of the zenon Logic Workbench. This manual gives you an overview of the use of zenon Logic Runtime.

SUPPORTED SYSTEMS

Overview of supported servers and desktop operating systems up to Windows 8.1

	Windows	Windows Embedded	Windows 7 SP1/Windows 8 and 8.1/
--	---------	------------------	----------------------------------

	Embedded 7/8 Standard	8.1 Pro/Industry	Server 2008 (R2) SP1, 2012 and 2012 R2
Editor	--	X	X
Runtime	X	X	X
Runtime for Windows CE	--	--	--
Web Server	X	X	X
Web Client	X	X	X
HTML Web Engine	X	X	X

Key:

- ▶ **X**: supported
- ▶ --: not supported

Attention: Windows CE is no longer supported from version 7.50.

zenon CE version 7.20 is installed. To use this, the Runtime files for version 7.20 must be created. Overview of supported desktop operating systems for Windows 10

Windows version	zenon Supervisor / Operator	Everywhere by zenon	zenon Logic Runtime
Windows 10 Home	X	X (PC)	X
Windows 10 Mobile	--	X (Mobile)	--
Windows 10 Pro	X	X (PC)	X
Windows 10 Enterprise	X	X (PC)	X
Windows 10 Education	X	X (PC)	X
Windows 10 Mobile Enterprise	--	X (Mobile)	--
Windows 10 IoT Core	--	--	X
Windows 10 IoT Enterprise (Windows 10 Enterprise LTSC)	X	X (PC)	X

Key:

- ▶ **X**: available
- ▶ --: Cannot run

ZENON WEB CLIENT

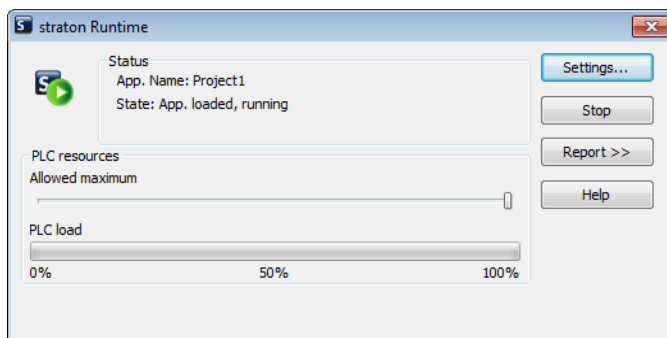
zenon Logic is not supported by zenon Web Client.

5.1 User interface

zenon Logic Runtime is available as a symbol in the system tray. This shows the current status of Runtime. A double-click on the symbol or on the **Show** item in the context menu opens the zenon Logic Runtime user interface.

The user interface shows a different status depending on the current status:

- ▶ runs
- ▶ Step mode
- ▶ Breakpoints set
- ▶ stopped



A progress bar shows the PLC load.

In addition to information about the status of the zenon Logic project running, the following actions can be executed in the zenon Logic Runtime user interface:

- ▶ Slider **Allowed maximum** of the **PLC resources**: set allowed maximum.
- ▶ **Settings...** (on page 30) button : Makes configuration possible via three tabs
 - General settings (on page 31)
 - Redundancy (on page 34)
 - Advanced users (on page 35)
- ▶ **Start/Stop** button: opens the dialog to start Runtime or stops Runtime
- ▶ **Messages** (on page 37) button: opens and closes the message window

5.1.1 Slider Allowed maximum

The slider **Allowed maximum** is used to guarantee other applications which run on the same device - such as zenon - system resources.

With this slider the maximal allowed execution time related to the cycle time can be set. If with a set cycle time of 10 ms the slider is set to 50% the PLC cycle may take a maximum of 5ms. If the PLC cycle takes longer than 5 ms, the remaining calculation is done in the next cycle. So one cycle has to be omitted from the PLC. This results in a cycle overflow.

The slider can be set between 10% and 100% steplessly.

The bar graph (PLC load) shows, how high the PLC load is in relation to the setting of the slider. From this you can conclude how to set the slider for an optimal tuning.

PLC load high (Slider between 50 and 100%): Set slider towards 100% or increase cycle time.

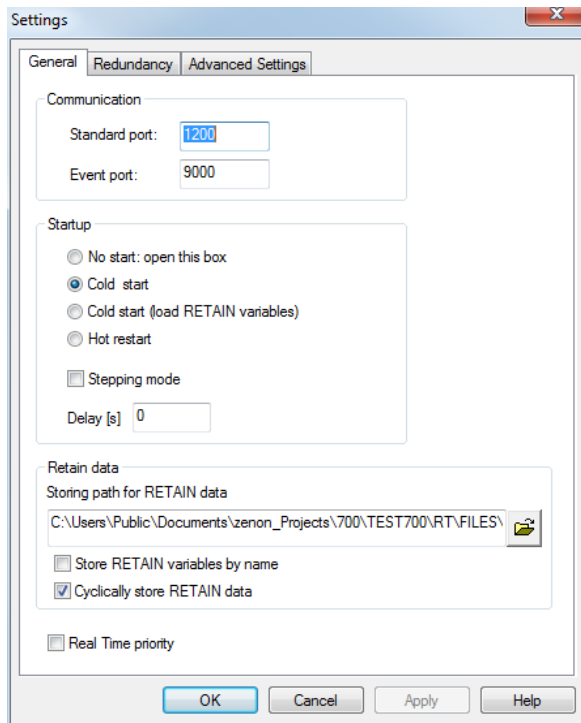
5.1.2 Settings

The **settings** make it possible to configure the zenon Logic Runtime with the help of three tabs:

- ▶ General settings
- ▶ Redundancy
- ▶ Advanced users

General

In the tab **General** the different parameter of the zenon Logic Runtime can be changed:



Parameter	Effect
Communication	
Port (Main):	IP port of the zenon Logic Runtime for the cyclic data transfer (e.g. zenon Logic Workbench).
Port (Binding):	IP port of zenon Logic Runtime for spontaneous data transfer (e.g. zenon Logic Workbench or binding).
Run-up	
No start. Open this box:	When the zenon Logic Runtime is started, first this box is opened.
Cold start:	Initialized run-up. Also retain variables are started initialized.
Cold start (loading of RETAIN variables):	Initialized run-up with current values of the retain variables. They only contain values, no states (see Info box).
Warm start:	Restart with all variable values from the last stop.
Start in step mode	Starts the zenon Logic Runtime and immediately stops it in debug mode.
Delay [s]	Delays the start by the set number of seconds. Default: 0
Retain Data	
Storage path Retain data	Here the file with the retain variables is stored. The default setting depends on how the zenon Logic Runtime is started.
Save retain variables with their names	Activate this checkbox if you want to save retain data in the format with the variable names. With this the values of the retain variables can also be saved if you add or delete a retain variable. Note: To save a variable, it must be highlighted as embedded in the <CA:PRODUCTNAME> Workbench. If you do not activate the checkbox, the retain data are saved the conventional way. After changing the retain variable, the Runtime starts with the initial values for all retain variables. Default: deactivated
Cyclic saving of retain variables	If you deactivate this check box, the retain data are no longer saved cyclically but only when the Runtime is closed. They are still saved after a change but not more frequent than about every 10 ms. Default: activated
Real-time priority	Sets the zenon Logic Runtime process to the windows priority Realtime. If the flag is not set, it means priority Normal.



Information

*If you activated option **Save retain variables with names** in zenon and also want to use it with manual Runtime start, you must make sure that in file K5DBXS.INI in area [CMD] option RETAINBYNAME is set. With this all retain variables are highlighted as embedded when the zenon Logic Workbench is started.*

Note: The INI file is only checked at the start of the zenon Logic Workbench. Changes during the Workbench runs have no effect.



Information

Retain variables

Retain data contain only the value of the zenon Logic variables not their status. This means for the start:

- ▶ Warm start: The status which was set for a variable is restored - regardless of whether it is a retain variable or not.
- ▶ Cold start with retain variables: Only the value of the retain variable in zenon Logic is restored, not the status.



Information

If the zenon Logic Runtime has been started with the zenon Runtime or with the zenon Logic Runtime Manager, changes done in this user interface are not effective after a restart of the zenon Logic Runtime.

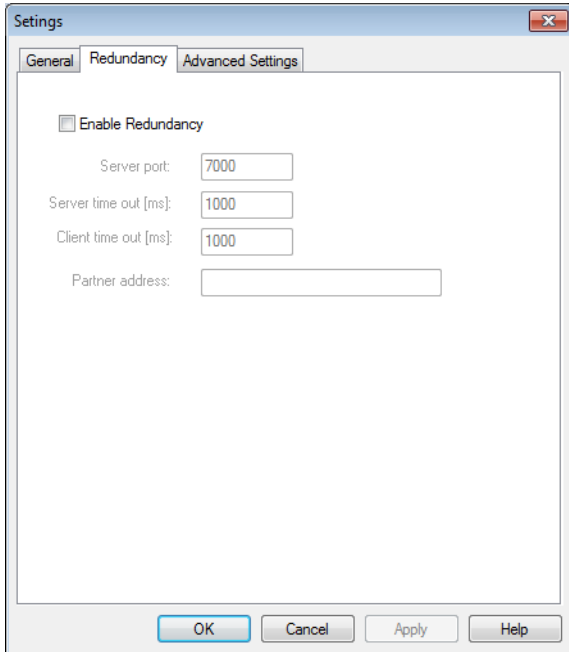
Please make the changes as usual in the zenon Logic project properties of the zenon Editors or in the zenon Logic Runtime Manager.

In such cases the dialog helps to determine which parameters were forwarded to the zenon Logic Runtime.

The changes to the settings which were done in this dialog are only effective if the zenon Logic Runtime is started directly (e.g. using the Windows Explorer).

Redundancy

In tab **Redundancy** you can see the redundancy settings for the zenon Logic Runtime.



You can check or enter the following entries:

Entry	Meaning
Activate redundancy	Runtime runs in redundancy mode
Server Port:	Port for replication connection
Server Timeout [ms]	Timeout in order to determine if the passive client is available. After the timeout expires, the replication is stopped.
Client Timeout [ms]	Timeout in order to determine that the active server is no longer available. After the timeout expires, the passive client becomes the active server.
Partner address	IP address of the redundancy partner.



Information

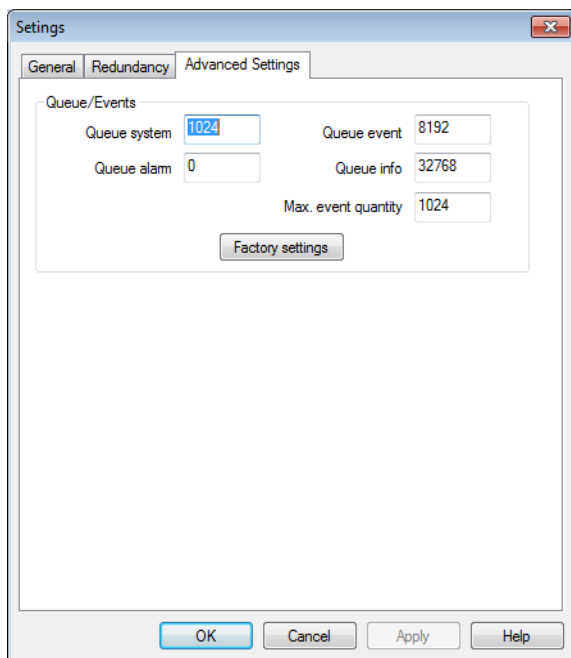
You can find details on configuring the redundancy of zenon Logic and zenon in chapter Redundancy zenon Logic and zenon (on page 67).

LIMITATIONS

Redundancy is not supported in zenon Logic RTK Runtime.

Extended settings

On this tab you carry out settings for the on change communication with external applications (Binding, zenon driver, ...).



The following settings are available:

Entry	Meaning	Range of values
System	Max. length of the system queue. In this queue e.g. overflows of the event queue are buffered.	1024 ... 65520 Default: 1024
Alarms	Length of the alarm queue (not used at the moment)	0 ... 65520 Default: 0
Events	Length of the event queue: The events of the zenon Logic Runtime event sever are written in this queue. From there they are sent via TCP. If this queue overflows, the overflow messages are written in the system queue.	0 ... 65535 Default: 8192
Messages	Length of the info queue. In this queue the messages to the zenon Logic Workbench are buffered.	1024 ... 65520 Default: 32768
max. puffer	Buffer size of the events per cycle. This parameter defines how many variables can be processed by the event server per cycle.	128 ... 65535 Default: 1024
Reset to default settings	Restores the default settings	



Information

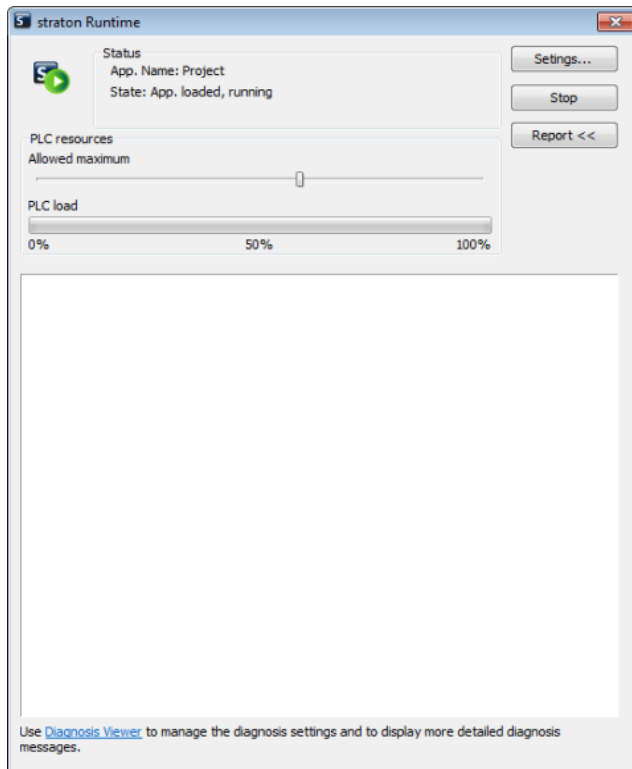
*If in large projects more than 1024 variables are transferred between zenon Logic and zenon with the `straton32.exe` driver, the parameter **Max. buffer** must be increased accordingly. An easy method to find out the number of **Max. buffer** is the value for the compiled I/Os in the output window of zenon Logic compilers. See illustration below.*

```
Build in progress... Please Wait...
>> Complex variables stored in a separate segment
>> Option not supported by runtime version 2.20 or older

Loading application symbols...
test
CounterX01
CounterX02
CounterX03
CounterX04
CounterX05
CounterX06
CounterX07
CounterX08
CounterX09
Building application data...
Warning: Compiled for DEM specific runtime: TSCD
< 132 BOOL/SINT; 1 INT; 11 DINT/REAL; 1 LINT/LREAL; 1 TIME; 1 STRING; - CRC = 2cfd6f84 >
< CT Segment = 1 byte[s] >
< 129 I/Os >
Relocating code...
< Code CRC=a53de7dd - File CRC=2fe496f0 - Size=190568>
No error detected
```

5.1.3 Messages

Runtime messages can be displayed in the zenon Logic Runtime user interface. Click on the **Messages>>** button to activate the view. The message display opens. Click on the **Messages<<** button to close the message display.



For actions in the message display, click in the message display with the left or right mouse button. You obtain a context menu:

Menu entry	Description
Display time stamp of messages	Adds a time stamp to each message.
Copy messages to the clipboard	Copies selected entries to the clipboard.
Delete all messages	Deletes all messages from the message window.

You administer the settings for zenon Logic logging and the drivers with the Diagnosis Viewer.

5.2 Licensing

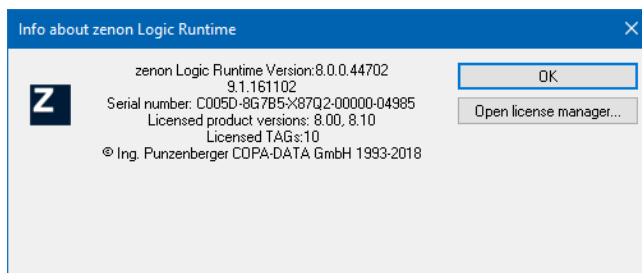
zenon Logic Runtime needs a valid license to start. Otherwise the 30 minute demo mode (on page 39) is started. Licensing is carried out by means of **COPA-DATA license administration**.

SHOW INFORMATION ABOUT THE ZENON LOGIC RUNTIME LICENSE

To view the license information:

1. With Runtime running, double-click on the symbol for zenon Logic in the system tray.
The dialog with the status information is opened.
2. In the top left corner, click on the zenon Logic logo.
3. In the drop-down list, select **About zenon Logic Runtime....**
The dialog with the license information is opened.

LICENSE INFORMATION DIALOG

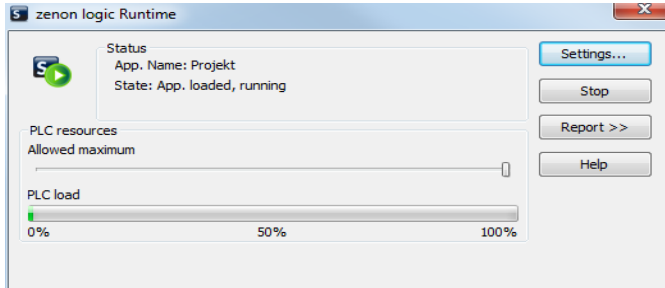


Option	Description
License information	Shows information about the license currently being used.
OK	Closes the dialog.
Open License Administration	Opens the License Manager . Licenses can be activated, returned and managed with this tool. The serial number is needed for this.

5.2.1 TAG handling zenon - zenon Logic

The number of the TAGs used in zenon Logic is the number of I/Os in zenon Logic minus the number of the used I/Os with a zenon or zenon Logic profile. These I/Os are already considered at the TAG calculation in zenon.

This makes sure that no I/Os are counted twice.



The I/Os used are displayed in the zenon Logic Runtime user interface.

5.2.2 Demo mode

If, when starting zenon Logic Runtime, no valid serial number is found, zenon Logic Runtime starts in demo mode. The Demo Mode stops the zenon Logic Runtime after 10 minutes. With a special demo serial number, zenon Logic Runtime runs continuously for 30 days. After these 30 days then the maximal time of the zenon Logic Runtime is again 10 minutes.

ZENON LOGIC MINI

You did not buy the zenon Logic Runtime license but you own a valid zenon Runtime license. Then a mini version of the zenon Logic Runtime is licensed automatically.

With the mini version up to eight TAGs (=variables) can be used by zenon Logic via input/output mechanisms. TAGs are variables of field bus drivers and variables with profiles (e.g. O&M profile zenon Logic). Excluded from the calculation are the zenon internal profiles **STRATON** and **zenOnRT** for zenon-internal communication.

Calculation example:

- ▶ Four variables for communication with zenon (are not calculated)
- ▶ four variables for the communication with Profibus
- ▶ four variables for the communication with the **IEC 61850** Client.

This equals the maximum number of eight TAGs ($0 + 4 + 4 = 8$).

5.3 Start the zenon Logic Runtime

The zenon Logic Runtime supports multi-instances and therefore needs parameters (transfer parameters), when it is started.

There are the following possibilities for starting zenon Logic Runtime:

- ▶ with the help of the zenon Runtime
- ▶ with the help of the zenon Logic Runtime Manager
- ▶ With a BAT file
- ▶ Using the command line

START WITH THE HELP OF THE ZENON RUNTIME

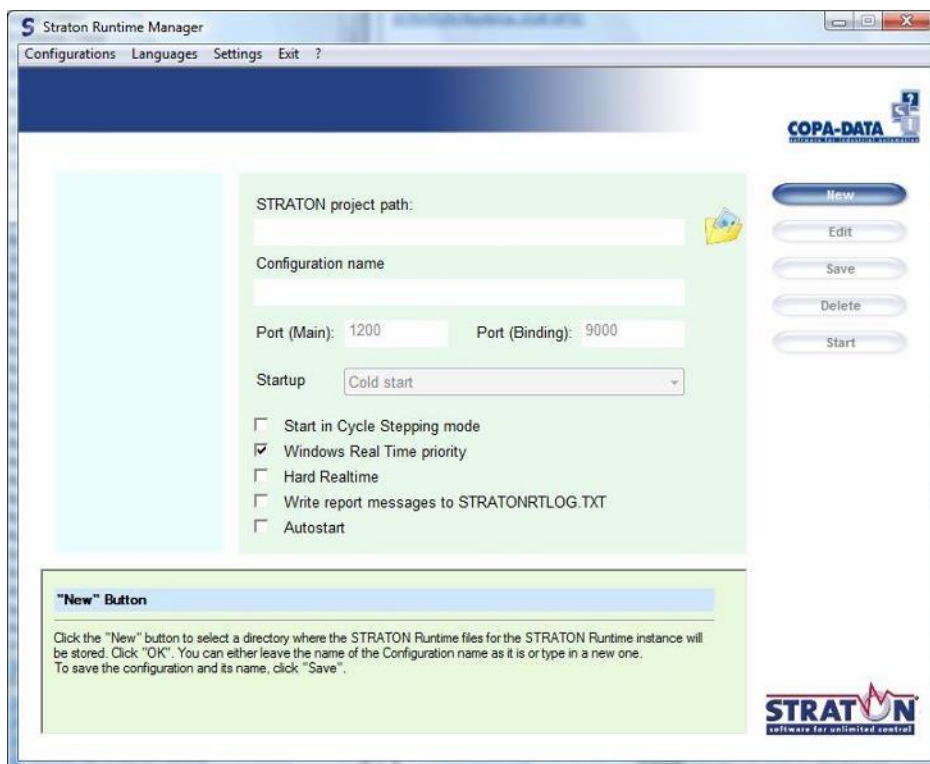
In order to start the zenon Logic Runtime with the zenon Runtime:

- ▶ Select the zenon Logic project in the zenon Editor.
- ▶ Change the `Start type` (**Runtime**) as you wish. The properties of **Start type** are described in the embedded help in zenon.

Note: If the **Start place** property is set to `Server 1` and `Server 2`, then `localhost` has to be used in the **Host** as a value. Otherwise zenon Logic Runtime can no longer start.

START WITH THE HELP OF THE ZENON LOGIC RUNTIME MANAGER

You can use the zenon Logic Runtime Manager to start the zenon Logic Runtime manually. You can find the zenon Logic Runtime Manager in the COPA-DATA area in the Windows® start menu. The zenon Logic Runtime manager is used to start zenon Logic Runtime instances on a PC. You can for example chose hard real time here in order to start the zenon Logic real time version (zenon Logic RTK) or to start a zenon Logic Runtime instance during the booting of the system ("Auto start").



You receive accurate information about the individual elements of the zenon Logic Runtime manager by moving the mouse pointer over them. The corresponding help is displayed in the property help.

You receive the complete help about this application by clicking the ? in the menu bar.



Attention

Runtime Manager must be started with administrator rights.

START IN A DOS SHELL

zenon Logic Runtime can be started using the command line interface or a BAT file. This start process is also valid for starting under Windows CE. If the path under CE contains a space, the path parameter must be defined between individual quote marks. Example: -PATH="\hard disk\project1\'

For the command the following is true:

- ▶ all commands must be entered in capitals
- ▶ the following signs are allowed:
 - -
 - / (must not be used as a last character for the value of **PATH**!)

The following handover parameters are available:

MAIN PARAMETER

Settings as under **Settings -> General (on page 31)** in the user interface of zenon Logic Runtime.

Parameter	Description
PORT=<Zahl>	Communication port. If no parameter is set: 1200 (Default)
PORTTRACE=<Zahl>	Event port. If no parameter is set: PORT+7800
PATH=<DB-Path>	Storage area of the zenon Logic application code. If no parameter is set: zenon Logic Runtime folder. Note: <ul style="list-style-type: none"> ▶ Must be set. ▶ Path must not end with a Backslash (\). ▶ For error detection, the driver checks the project name at the end of the path. ▶ zenon Runtime discovers, when starting, zenon Logic Runtimes that are already running by comparing the complete path.
ZENPATH=<Pfad>	zenon path
ZENPROJECT=<Name>	zenon project name. Note: Must be set.
DONTHIDE	Do not hide the Runtime after starting (it is the only parameter without the = character)
STEP=<0 1>	Activation/deactivation of the stepping mode: <ul style="list-style-type: none"> ▶ 1: Start in stepping mode ▶ 0: no stepping mode ▶ Without parameters: no stepping mode
START=<0 1 2 3>	Type of Runtime start: <ul style="list-style-type: none"> ▶ 0: Display setup dialog ▶ 1: Coldstart ▶ 2: Warm start ▶ 3: Hot start ▶ Without parameters: Warm start
LOWPRIORITY=<0 1>	Priority: <ul style="list-style-type: none"> ▶ 0: low priority ▶ 1: Windows real time priority ▶ Without parameters: low priority
WRITELOG=<0 1>	Setting for logging:

	<ul style="list-style-type: none"> ▶ 0: Do not create LOG ▶ 1: Create LOG ▶ Without parameters: Create LOG <p>The log file that is stated in the PATH parameter is created. Name: STRATONRTLOG.TXT</p>
--	---

ADDITIONAL PARAMETERS

Parameter	Description
REMPATH=<Pfad>	Path for remanent data. If no parameter is set: PATH folder
DELAY=<Zahl>	Start delay in [s]: Delayed start of the zenon Logic Runtime. If no parameter is set: 0
MAXUTILPRCT=<Zahl>	Position of the slider for maximum PLC workload in %. If no parameter is set: 100

EVENT PARAMETER

Settings as under **Settings -> Advanced settings** (on page 35) in the user interface of zenon Logic Runtime for the queue/events.

Parameter	Description
QSSYSTEM=<Zahl>	Size of system queue. If no parameter is set: 1024
QSALARM=<Zahl>	Size of alarm queue. If no parameter is set: 0
QSEVENT=<Zahl>	Size of event queue. If no parameter is set: 8192
QSINFO=<Zahl>	Size of info queue. If no parameter is set: 32768
NBEVENT=<Zahl>	Maximum number of recordable events. If no parameter is set: 1024
EVENTCON=<Zahl>	Number of possible event connections. If no parameter is set: 256

REDUNDANCY PARAMETER

Settings as under **Settings -> Redundancy** (on page 34) in the user interface of zenon Logic Runtime.

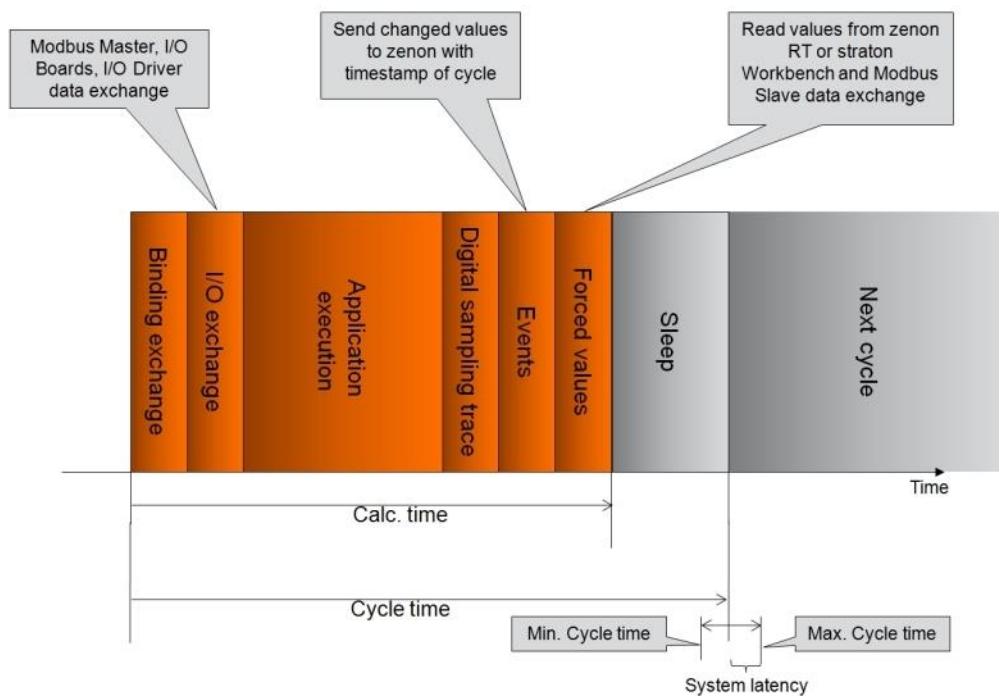
Parameter	Description
REDENABLE=<0,1>	Redundancy on/off. Without parameters: 0
REDADDR=<IP>	Redundancy partner address.
REDPORT=<Zahl>	Redundancy port for replication.
REDCLITIMEOUT=<Zahl>	Redundancy client time out [ms]; time out which is used to determine when a passive client becomes an active server. If no parameter is set: 1000
REDSRVTIMEOUT=<Zahl>	Server Timeout redundancy [ms]. Time-out for ending the replication. If no parameter is set: 1000
REDZENON=<0,1>	If set to 1, the partner address is read from the PROJECT.INI . If no parameter is set: 0

5.4 zenon Logic Runtime cycle

The zenon Logic Runtime cycle consists of:

- ▶ **Calc. time:** Time for data exchange, calculation, save calculated values, read and write modified values, idle time.
- ▶ **Cycle time:** Complete cycle time determined by system latency.
- ▶ **System latency:** Response time of the system.

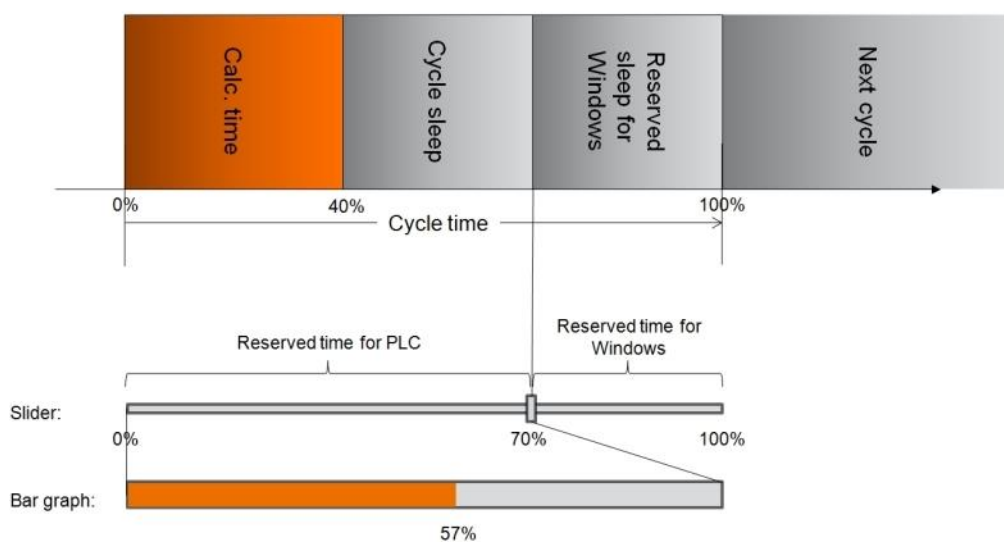
These values can be displayed in the zenon Logic user interface. Use the shortcut **Alt+S** or **Shift+Alt+A**.



Property	Description
Binding exchange	Reading of the Binding values which are published in the network and which were configured by the application (if they were configured).
I/O exchange	Data exchange between the I/O drivers (except Modbus slave/server) and the I/O boards (if they were configured).
Application executione	Execution of the programs.
Digital sampling trace	Execution of the digital sampling (if it was configured).
Events	Sending of the events (Binding or straton32.exe driver of zenon) (if they were configured).
Forced values	Reading of the forced values of zenon from the zenon Logic Workbench (online debugging) and data exchange of the Modbus slave/server (if they were configured).
Sleep	<p>Wait the remaining time until the cycle time is over.</p> <p>When cycle setting is <i>As fast as possible (free run)</i>, the time is always 1 ms.</p> <p>The Sleep time is used in order to execute tasks of the operation system and of other windows applications.</p>

TIME ALLOCATION PLC AND WINDOWS

With the help of the slider in the zenon Logic Runtime user interface, you can determine how much time is reserved for the PLC and for Windows. The value displayed in the bar display always refers to the ratio set in the slider.



BEHAVIOR WITH CONFIGURED FREE RUNNING MODE

If, in the zenon Logic development environment, Free running (= as fast as possible) is set up, this configuration is not supported in zenon Logic Runtime. In this case, a cycle time of 1 millisecond is used in Runtime.

Note: Free running (as fast as possible) is configured in zenon Logic in the **Options** dialog. To do this, call up the **Project** menu entry in zenon Logic. Select the **Parameter...** entry there. Select the **cycle time** property in the **Options** dialog in the **Runtime** area. The **Cycle time** dialog with the corresponding options opens when double clicking on the property.

5.4.1 Error message cycle time

If the **cycle time** is exceeded, messages are displayed in the zenon Logic Runtime or in the Workbench and corresponding entries are written to the LOG file.

Error message user interface	Entry log file	Meaning
Timing configuration error. (x) cycles lost.	Timing configuration error. (x1) cycles lost. Configured cycle time = (x2) us, maximum execution time = (x3) us, maximum PLC utilisation = (x4) .	Number (x) of not executed cycles due to cycle time out. The execution time is higher than the set cycle time or there are not enough CPU resources available.
Timing configuration error. PLC utilization peaked at (x)%.	Timing configuration error. PLC utilization exceeds 80%. Configured cycle time = (x1) us, maximum execution time = (x2) us, maximum PLC utilisation = (x3) % ."	The execution time is higher than 80% of the set cycle time; actually (x)%.

TROUBLESHOOTING

You can rectify the **cycle time** being exceeded by:

- ▶ **cycle time** increasing
- ▶ decreasing execution time, e.g.:
 - use a system with more CPU resources
 - Use code compiled with C-Compiler
 - reduce the program

5.5 Close the zenon Logic Runtime

The zenon Logic Runtime can be closed manually either by using the context menu of the symbol in the system tray or by using the open window of the zenon Logic Runtime:

CONTEXT MENU SYMBOL

Right clicking the icon in the system tray opens the following context menu:



1. A dialog is opened if **Close** is clicked on.

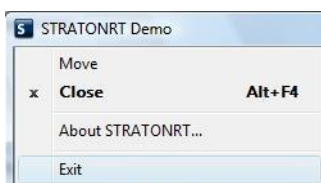


2. Clicking on Yes stops and closes Runtime.

IN THE WINDOWS OF THE ZENON LOGIC RUNTIME

Another possibility to close the Runtime is in the open window of the zenon Logic Runtime.

The following menu opens after a click to the upper, left corner of the window:



1. A dialog is opened if **Close** is clicked on.
2. If you acknowledge the dialog, the zenon Logic Runtime is closed immediately.



Information

If the zenon Logic Runtime was started together with the zenon Runtime and for the zenon Logic project the start type 'autom. start/stop' was selected in the zenon Editor, the zenon Logic Runtime is automatically closed when the zenon Runtime is closed.

5.6 Creating the hot restart file

When you close the zenon Logic Runtime, a `Hot Restart` file is created automatically. You can trigger the writing of the `WriteHot` file via function `Hot Restart`.

Every time a `Hot Restart` file is created, a backup file is also created for which the last character of the file name is replaced with a '_'. In addition the files are created with a header which contains the `Checksum` of the data. When reading the data, header and `Checksum` are checked. If an error occurs, the backup file is used.

Function `WriteHot` has the following input parameters:

Parameter	Description
I	Trigger; if this parameter is <code>TRUE</code> the function is executed.
SYNC	If this parameter is <code>TRUE</code> , the cycle is hold for as long as the <code>Hot Restart</code> file is written. If the parameter is <code>FALSE</code> , the function is carried out asynchronous.
BUSY	This parameter is <code>TRUE</code> if the function is executed.



Information

If you call up the function several times in a cycle, only the last call is considered.

5.7 Remote systems

This chapter deals with target systems which are not on the local computer. The target system is accessed via Ethernet TCP/IP.

5.7.1 zenon Runtime and zenon Logic Runtime on one remote PC

INSTALLATION ON THE TARGET SYSTEM

First you install and license the zenon Runtime on the remote PC. The zenon Logic Runtime is automatically installed and licensed.

DOWNLOAD THE PROJECT DATA WITH ZENON LOGIC RUNTIME, START TYPE "AUTOM. START/STOP" OR "AUTOM. START"

In order to download the zenon and zenon Logic Runtime files, use the Remote Transport function of zenon. You will find more information in the chapter Remote Transport.

The zenon Logic Runtime is started and closed together with the zenon Runtime automatically.

DOWNLOAD OF THE APPLICATION DATA WITH THE ZENON LOGIC RUNTIME START TYPE "START MANUALLY"

If the zenon Logic Runtime is started manually on the target PC (e.g. during boot up of the operating system), the PLC code has to be loaded manually to the target platform with the zenon Logic Workbench. For that

- ▶ enter the correct parameters in the property dialog **Workbench** in the zenon Editor: **Target host** and **Target port**
- ▶ in the zenon Logic Workbench click on the button **Online**
- ▶ with this the zenon Logic Workbench establishes a connection to the target system.
- ▶ now you can transfer the PLC code to the target system via button **Download**; the progress of the download is displayed with the help of a progress bar
- ▶ after the download the PLC starts automatically with the new application.
- ▶ If the download was successful and the PLC program runs on the target system faultless, **RUN** is displayed in the toolbar

5.8 zenon Logic Runtime with Real Time Kernel (RTK)

The zenon Logic RTK (Realtime Kernel) is the zenon Logic Runtime system for hard realtime. In contrast to the "normal" zenon Logic Runtime, this runs at the same time and independently of the operating system. The highest availability and security against failure is thus guaranteed. zenon Logic is thus also still available if the operating system is blocked or no longer available.

Requirements:

- ▶ zenon Logic RTK is available for the following Windows operating systems:
 - Windows 7, 32-bit version
 - Windows 8 and 8.1, 32-bit version
 - Windows 10, 32-bit version

Attention: zenon Logic Runtime with RTK can only run on 32-bit operating systems.

- ▶ zenon Logic RTK can be started with one of the following possibilities:
 - zenon Logic Runtime Manager (**hard real time** (on page 86) option must be set)

- By means of the command line
- With a BAT file.



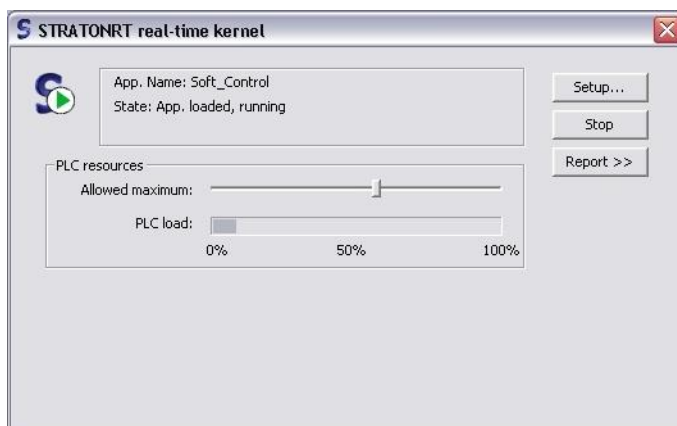
Attention

*Do not operate two real-time extensions in parallel on one computer.
This means:*

- ▶ If another real-time extension is already active on the computer, do not start the zenon Logic RTK.
- ▶ Only one instance of the zenon Logic RTK can be started at a time.

The Realtime Kernel is designed to trigger the zenon Logic Runtime cycle independent of the operation system. So the zenon Logic Runtime cycle is guaranteed to be executed.

The PC's part of the calculating capacity can be set with a slider.



Depending on the size of the PLC application and the set cycle time, the slider has to be adjusted to the PLC load. The slider can be set between 10% and 85%. The current load is visualized as a bar and indicates whether the permitted maximum for executing the PLC cycle is sufficient.

If the display reaches the 100% mark, there is a danger of cycle overflows! In this case set the slider higher (towards 85%). If this is not possible due to programs in Windows (e.g. long screen opening times of zenon), the cycle time of the zenon Logic application has to be increased.



Attention

If the bar reaches the 100% mark, there is a danger of cycle overflows!



Example

Slider at 10%, cycle time 10ms:

The zenon Logic RTK must process the PLC program in 1 ms which equals 10% of the set cycle time of 10 ms.

Slider at 50%, cycle time 5ms:

For executing the PLC program 2.5ms are available for the zenon Logic RTK.

Slider at 85%, cycle time 20ms:

For executing the PLC program 17ms are available for the zenon Logic RTK.

Formula

To ensure that the slider setting does not lead to cycle overflows in conjunction with the cycle time, the following formula can be applied:

$$\text{cycle time} > (100 / \text{Sliderposition} [\%]) * \text{calculating time}$$

In order to find out the calculating time of the PLC cycle, click Shift+Alt+A and note the value Max. Cal. . time [μs].

Assumption:

Max. cal. Time: 274μs

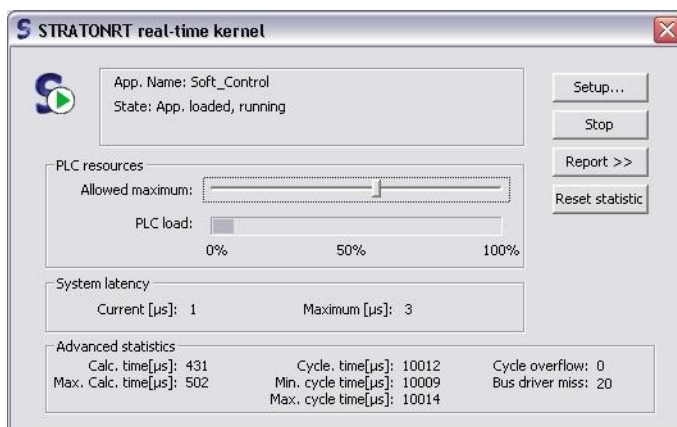
Slider position: 50%

Set cycle time: 1ms (1000μs)

Calculation:

$$(100/50) * 274 = 548\mu s$$

This means that the defined cycle time of 1 ms is sufficient.



**Attention**

Note: If you do not stick to the described formula, the determinism which is based in the cycle time is not given anymore!

CYCLE TIME

Cycle times from 100 µs or higher can be set.

*Note: The actual use of the set cycle time depends on the system and the selected HAL Timer. Information on the current system load is provided by the system load time (keyboard shortcut: **ALT+S**) and **Advanced statistics** (keyboard shortcut: **SHIFT+ALT+A**).*

Description of the measured values displayed:

System latency (Alt+S)	
Current [µs]	Current system latency (see also Runtime cycle (on page 46))
Maximum [µs]	Maximum system latency (since the start of the Runtime)

Extended statistic (Shift+Alt+A)	
Cal. time [µs].	Current calculation time of the last cycle
Max. cal. time [µs].	Maximal calculation time for a PLC cycle (since the start of the Runtime)
Cycle time [µs]	Current cycle time
Min. cycle time [µs]	Minimal cycle time (since the start of the Runtime)
Max. cycle time [µs]	Maximal cycle time (since the start of the Runtime)
Cycle overflow [1]	Counter for cycle time exceedance (since the start of the Runtime)
Driver overflow [1]	Counter for failed PLC cycles of Windows (non-realtime) drivers (since the start of the Runtime)

The value of the cycle time can always only be a multiple of the HAL Timerperiod (time basis). The time basis depends on the selected timer and its time basis (see also zenon Logic RTK configuration (on page 56)).

That means: A set cycle time of 10 ms of a hardware with a time basis of 122 µs is not executed with exactly 10,000 ms but with 10,004 ms.

5.8.1 Restrictions and requirements of the zenon Logic RTK

HARDWARE REQUIREMENTS FOR THE ZENON LOGIC RTK

ACPI:

The hardware (PC or IPC) and the operating system must support ACPI:

- ▶ ACPI-Multiprocessor-P
- ▶ ACPI-PC (Advanced Configuration and Power Interface)
- ▶ ACPI uniprocessor PC

You can check whether your PC supports ACPI in the system control (system/device manager/computer).

CPUS:

The zenon Logic RTK is supported from the following processor generations on:

- ▶ Pentium II or younger
- ▶ AMD Athlon XP or younger

LIMITATIONS

The zenon Logic Real Time Kernel (RTK) runs as a separate process at hardware level independently of the operating system. In contrast, zenon Logic Runtime runs on Windows®. The real-time-enabled zenon Logic RTK has a few restrictions compared to the zenon Logic Runtime.

- ▶ Virtual Machines are not supported.
- ▶ zenon Logic redundancy is not supported
- ▶ Not all function blocks are supported.
- ▶ Not all I/O drivers are supported.
- ▶ Some real-time drivers must be registered/deregistered manually.

FUNCTION BLOCKS

The following function blocks are not processed directly in the Kernel Mode, but only emulated:

- ▶ all file operations (category 'Files')
- ▶ all TCP/IP modules (category 'TCP-IP')

Following function blocks are not supported by the zenon Logic RTK:

- ▶ day_time_local (Category 'Clock')



Information

*The respective configuration of Runtime can be loaded in the zenon Logic workbench. To do this, select the **Configuration** entry in the Workbench in the context menu of the loaded project.*

After you have selected the configuration, the function blocks which are not supported are marked red in the library.

I/O DRIVER

The following zenon Logic I/O drivers are executed by the zenon Logic RTK in Kernel Mode (real time):

- ▶ Profibus (Hilscher CIF Profibus)
- ▶ Hilscher SYCON universal (Hilscher SYCON configuration)

You can find more information about supported I/O drivers here: zenon Logic I/O driver support (on page 76)

REGISTER AND DE-REGISTER REAL TIME DRIVERS

There is a small chance that the real time driver of the zenon Logic RTK must be registered and de-registered manually. In order to do this start the command prompt via:
Start > execute > cmd.

Use the following syntax:

- ▶ Register:
`<Folder of the zenon Logic RTK>:\stratonrtkvm -regsrv`

- ▶ De-register:
`Folder of the zenon Logic RTK>:\stratonrtkvm -unregsrv`

Example of the <folder of the zenon Logic RTK>: %Programme (x86) %\COPA-DATA\zenon 800

5.8.2 zenon Logic RTK configuration

The zenon Logic Real Time Kernel (RTK) runs as an own instance on hardware level and is independent of the operating system. The Timerticks which are used for control are directly obtained from the motherboard. Which timer are available depends on the used hardware. The configuration is carried out by the zenon Logic RTK configurator.

Start RTK configuration

Double click the button for zenon Logic RTK Configurator in the windows control panel.



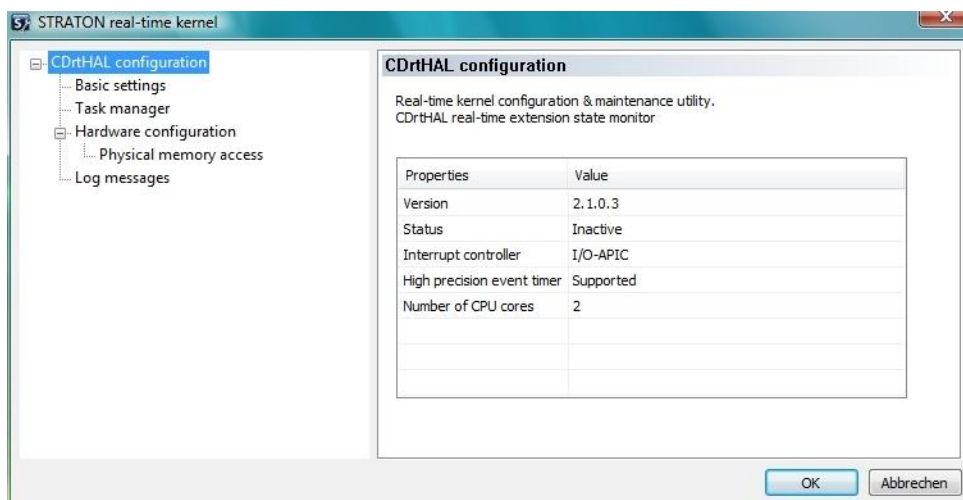
The window for editing the configuration opens. In the left window you see a menu tree with the following options:

- ▶ CDrthAL configuration
- ▶ Default settings
- ▶ Task manager
- ▶ Hardware configuration
- ▶ Access to the physical memory.
- ▶ LOG messages

In the right-hand window you can see information about the menu items and configure the RTK.

RTK basis information

After the start menu item 'CDrthAl configuration' is active in the left-hand window of the configuration software. In the right-hand window displays basic information about the configuration and the system:



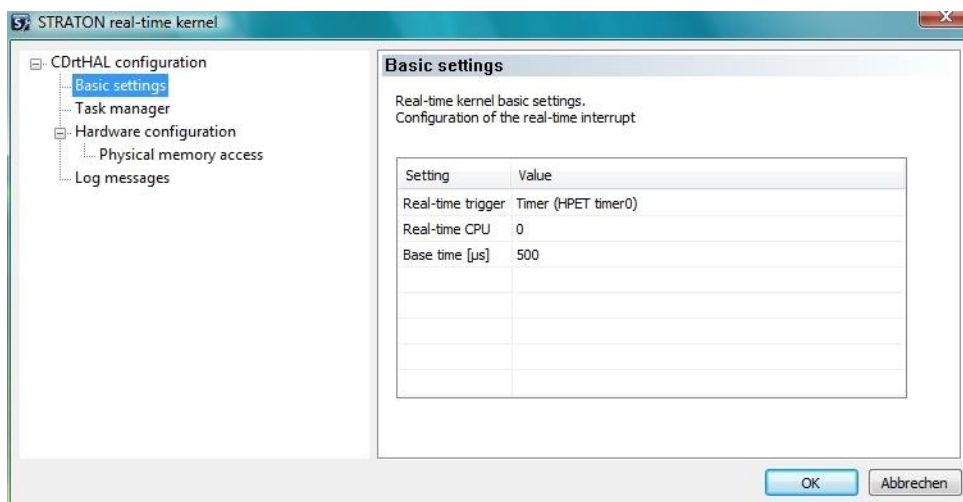
Property	Information
Version	Version of the real time driver (CDrtHAL).
Status	RTK is active or inactive
Interrupt controller	Type of interrupt controller.
High precision event timer	Support depends on hardware:
Process cores	Number of available process cores.

RTK basic settings

RTK BASIC SETTINGS

In the basic setting you define the following:

- ▶ Source for the Timertick
- ▶ CPU on which the RTK is carried out
- ▶ Basic time as basis for the multiplication factor for the calculation of the cycle time



TIMER FOR REAL-TIME SOURCE

Which timer is displayed on the selection list, depends on the used hardware. This information is also displayed on the start page of the RTK configuration. Available can be:

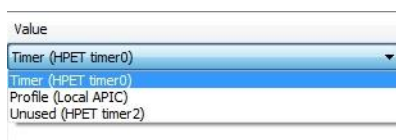
- ▶ Interrupt-Controller APIC
- ▶ Interrupt-Controller PIC
- ▶ HPET (High precision event timer)

The following possibilities arise from this:

1. PIC without HPET
 - Timer (8254 Timer), basic times: 334 μ s, 143 μ s, 112 μ s, 53 μ s
 - Profiles (real-time clock), basic times: 488 μ s, 244 μ s, 122 μ s
2. APIC without HPET
 - Profiles (real-time clock), basic times: 488 μ s, 244 μ s, 122 μ s
 - Profiles (local APIC), basic times: choose anywhere between 50 - 500 μ s
 - Unused (8254 Timer), basic times: choose anywhere between 50 - 500 μ s
3. APIC with HPET
 - Timer (HPET timer 0), basic times: 500 μ s, 250 μ s, 125 μ s
 - Profiles (local APIC), basic times: choose anywhere between 50 - 500 μ s
 - Unused (HPET timer 2), basic times: choose anywhere between 50 - 500 μ s

Whether and to what extent basic times are available depends on the hardware (real-time clock) and the operating system. When using the 'timer' real-time source, the source for the Windows Timer Interrupt is used for the real time execution. In this case the set basic time must be an integral factor of the Windows Timer Interrupt basic time.

real-time source: depends on the operating system Select using list field (drop down).



We recommend the following basic setting: Unused (HPET Timer2).

Real time CPU: Select the CPU on which the process of the RTK should be carried out.

Basic time: Basic time of the timerticks. Depending on the real time source, the basic time is selected from a list or entered freely. During operation the multiplication factor for the timer ticks refers to the entered basic time.



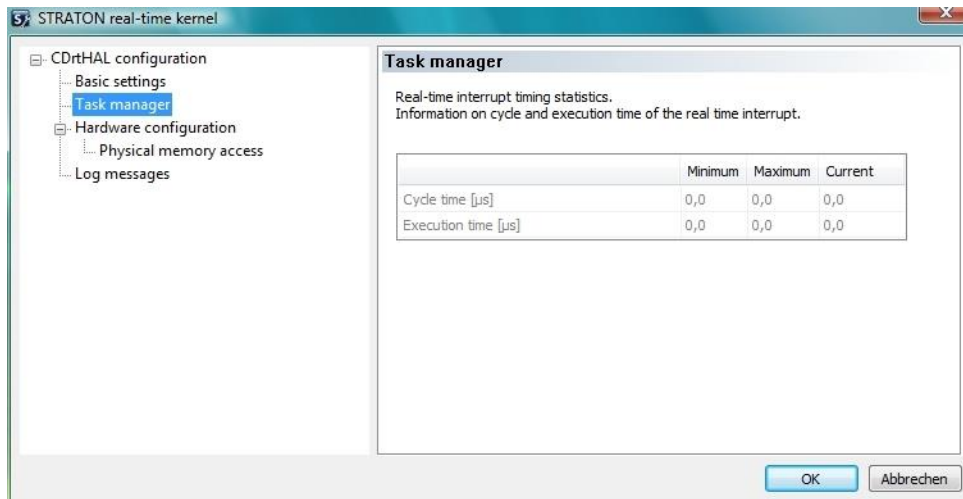
Attention

When using Profile: Real time source the kernel profiling must not be activated under any circumstances as long as a zenon Logic RTK application is executed. For more information about kernel profiling see:

*<http://www.microsoft.com/downloads/details.aspx?familyid=9d467a69-57ff-4ae7-96ee-b18c4790cffd&displaylang=en>
(<http://www.microsoft.com/downloads/details.aspx?familyid=9d467a69-57ff-4ae7-96ee-b18c4790cffd&displaylang=en>)*

RTK Taskmanager

The task manager displays cycle times and execution times.

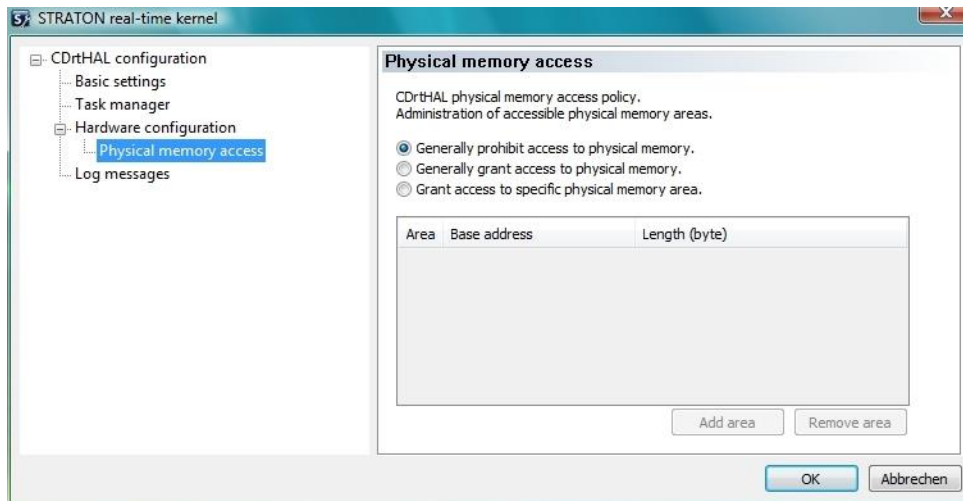


Property	Information
Cycle time (in microseconds)	Minimal, maximal and current value. Cycle refers to the cycles of the Timerticks based on the basic time.
Execution time (in microseconds)	Minimal, maximal and current value.

RTK hardware configuration

A separate reserved storage area is needed for the communication with the I/O drivers (e.g. Profibus) which are executed in the kernel mode. zenon Logic RTK and the I/O driver system agree on an area on which both have read and write access.

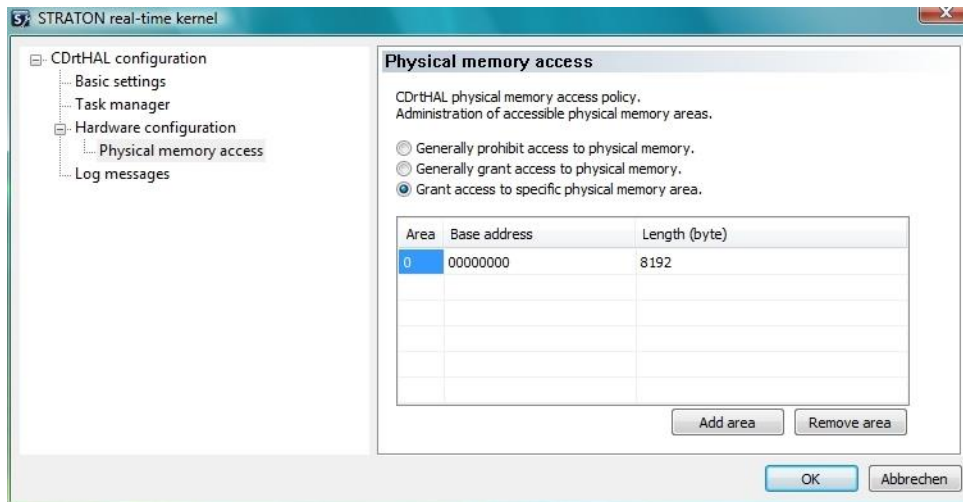
The configuration offers three settings:



Option	Effect
Generally prohibit access to the physical memory.	Free access memory areas are a safety risk with regards to malware. Thus access to the physical memory is denied in the shipping state. Default settings
Generally allow access to the physical memory.	Allows the communication to use the total available memory for communication, zenon Logic RTK und Profibus set their communication areas randomly. Unsafe setting which is not recommended.
Allow access to a defined physical memory area.	. zenon Logic RTK and I/O driver have a strictly allocated memory area. Recommended setting

Define memory area

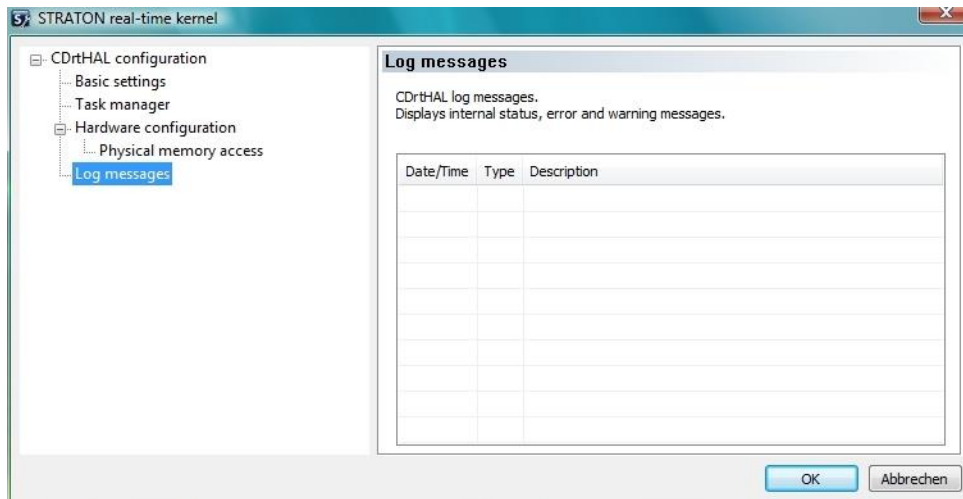
In order to allocate a fixed memory area for zenon Logic and I/O driver, click on **Access to the physical memory** in the left window and then on **Allow access to defined physical memory area** in the right selection window.



Property	Information
Area	The number is assigned by the configurator automatically.
Base address	Start address for the free memory. It can for example be determined with the help of the Hilscher Tool (included in the delivery of the card).
Length	Length of the free area. Attention: must be stated in Bytes.
Add area	Creates a new entry in the list.
Delete	Deletes an existing entry from the list.

RTK Logs

RTK messages are shown in the **Log messages**.



Property	Information
Date/Time	When did the event happen
Type	What type of message (status, error, ...)
Description	What has happened?

Close RTK configuration

By clicking button **OK** all changes are saved and the configurator is closed. If the system needs to be rebooted, a corresponding message is displayed.

If you close the configurator by clicking the close symbol (**X**) or by clicking **Cancel**, a dialog appears asking whether you want to save the changes.

5.8.3 Blue Screen Handling

A possible Bluescreen of Death (BSOD) is recognized by the zenon Logic RTK immediately.

- ▶ If the zenon Logic RTK is configured for stopping in case of a BSOD, the shut down program - if defined - is executed immediately, i.e. before the next cycle.
- ▶ If the zenon Logic RTK is configured to continue, the next cycle is executed after a minimum delay of approximately 10 [ms] (switching of the graphics mode).

To achieve this, in the zenon Logic application, the zenon Logic **RTK_OnBugCheck** function must be added to the zenon Logic standard library with `TRUE`.



Attention

In case of a BSOD the execution of the zenon Logic application can be limited.

Following functions are no longer supported when this occurs:

- ▶ File Operations
- ▶ TCP functions
- ▶ Functions of the serial interface
- ▶ Non real-time field bus driver
- ▶ System clock functions
- ▶ RETAIN variables
- ▶ Writing of log messages (the right to make changes due to product version cycles is reserved)

*In order to check in the zenon Logic application if a BSOD has occurred, use the zenon Logic function **RTK_IsBugCheck**.*

OPTIONS FOR PROFIBUS MASTER (HILSCHER CIF PROFIBUS):

No restart after stop (keeping outputs): If this option is active, the outputs of the Profibus I/Os remain set after the shut down. Otherwise the outputs would be set to the defined alternate values after the shut down sequence.

5.8.4 Error codes at starting the zenon Logic RTK

The displayed error code refers to starting the `STRATONRTKVM.exe`.

Error code	Meaning
0x00000000	If an error occurred in the zenon Logic RTK, usually the error is explained in a message: The log message with the error code 0x00000000 can be ignored.
0xFFFFFFFF	A timeout occurred on starting the STRATONRTKVM.exe. Possible reasons: STRATONRTKVM.exe does not exist, too high load of the system. Attention: After such an error make sure that the STRATONRTKVM.exe as well as the STRATONRTK.exe no longer run (task manager) before restarting the zenon Logic RTK.
0x00000001 – 0x0000FFFF	Is an error code of the zenon Logic virtual machine (COPALP error message).
0xFFFF01, 0xFFFF02	Error loading the non-realtime bus drivers! probable reason: incompatible (outdated) version of one or more bus driver(s).
0xFFFFF000	rTHAL error code (0xFFFFF000 is added to the rTHAL code)
0xFFFFF001	Make sure, that the rTHAL driver has been installed correctly. If the problem still exists, contact your zenon support and tell them the Windows version (service pack, hot fix, ...) and the used Windows HAL. The name of the used HAL is shown in the device manager under computer. System requirements: ACPI, Pentium II or AMD Athlon or higher.

RTHAL ERROR CODES:

0x0	No error
-----	----------

POSSIBLE ERROR CODES ON STARTING

Error code	Meaning
0x1	Error while loading the driver. Reason: Driver is not registered correctly.
0x2	START could not be executed. Reason: Wrong version of the driver is installed.

ERROR CODES DURING RUNTIME

Error code	Meaning
0x6	SUSPEND could not be executed.
0x7	STOP could not be executed.

INTERNAL ZENON LOGIC VIRTUAL MACHINE ERROR CODES

T5RET_OK 0
 T5RET_ERROR 1
 T5RET_BADCODE 2 invalid app code - bad header
 T5RET_BADCODETARGET 3 invalid code - bad target id
 T5RET_BADCODEVERSION 4 invalid code - bad version
 T5RET_EXECSIZE 5 exec buffer overflow
 T5RET_BADREQUEST 6 unknown comm request
 T5RET_SERVERREAD 7 cannot process read request
 T5RET_UNKNOWNTIC 8 unknown tic code
 T5RET_DATA8OVERFLOW 9 too many DATA8 used
 T5RET_DATA16OVERFLOW 10 too many DATA16 used
 T5RET_DATA32OVERFLOW 11 too many DATA32 used
 T5RET_DIVZERO 12 division by 0 in TIC (32bit)
 T5RET_TIMEOVERFLOW 13 too many TIME variables
 T5RET_CANTACTIME 14 active timers not supported
 T5RET_SAFETICLOOP 15 awful loop in TIC (safe mode)
 T5RET_OUTOFTIC 16 TIC code ointer out of range
 T5RET_UNKNOWNSTF 17 not supported std function
 T5RET_LDIVZERO 18 division by 0 in TIC (64bit)
 T5RET_DATA64OVERFLOW 19 too many DATA64 used
 T5RET_STRINGOVERFLOW 20 too many strings
 T5RET_SDIVZERO 21 division by zero in TIC (8 bit)
 T5RET_WDIVZERO 22 division by zero in TIC (16 bit)
 T5RET_HOTRESOURCES 23 HOT restart: system resources used
 T5RET_HOTMAP 24 invalid var map in hot restart
 T5RET_HOTACTIME 25 change of active timer map
 T5RET_HOTSTAMP 26 bad build stamp for hot restart
 T5RET_HOTSFC 27 SFC changed - no hot restart!
 T5RET_HOTRETAIN 28 retain def has changed - no hot
 T5RET_HOTFORMAT 29 bad DB format - no hot restart!
 T5RET_CYCLEOVERFLOW 30 cycle time overflow
 T5RET_RETAINSTORING 31 cant store retain variables
 T5RET_ARRAYBOUND 32 array index out of bounds
 T5RET_RECURSE 33 recursive call of sub-program

T5RET_XV_OVERFLOW 34 too many external variables used
T5RET_NOBSAMPLING 35 sampling trace not supported
T5RET_TASK_OVERFLOW 36 too many tasks
T5RET_HOTCALLSTACK 37 call stack overflow (hot)
T5RET_STACKOVER 38 stack overflow
T5RET_HOTASI 39 cannot hot restart ASi
T5RET_TOXC 40 OXC limit reached
T5RET_WOXC 41 OXC info

5.9 Redundancy zenon Logic and zenon

zenon Logic can be used redundantly and offers the possibility to combine the zenon redundancy with the zenon Logic redundancy.

In this chapter the configuration of the zenon/zenon Logic redundancy is described. The zenon/zenon Logic redundancy is referred to as SCADA logic redundancy from here on.

Further terms:

- ▶ zenon server = dominant zenon server
- ▶ zenon SB = zenon Standby Server
- ▶ Active zenon Logic Runtime = zenon Logic Runtime instance being executed
- ▶ Passive zenon Logic Runtime = backup system (Standby) of zenon Logic Runtime

OPERATING SYSTEMS

You can find information on supported operating systems in the **zenon Logic Runtime** (on page 27) chapter, **Supported systems** section.

LIMITATIONS

Redundancy is not supported in zenon Logic RTK Runtime.

5.9.1 Requirements

Two PCs are necessary for SCADA logic redundancy. The both PCs are connected via Ethernet TCP/IP (100 MBit, recommended 1 GBit).



Attention

The SCADA logic redundancy requires both systems, zenon and zenon Logic, to be installed on one PC for each server.

We recommend to set the cycle time of the zenon Logic project to twice the cycle time of a single project, but at least to 50 ms.

Since SCADA logic assumes that zenon Logic and zenon communicate with each other using the **Logic to SCADA** driver, it must be ensured that the zenon server and the active zenon Logic Runtime run on the same PC.

In order to prevent a one-sided switching from zenon Logic or zenon, a synchronization between the zenon Logic Runtime and the zenon Runtime must be engineered. A watchdog is used for this.

WATCHDOG

ZENON LOGIC

The status of the zenon project (standalone/server/standby server/client) is transferred via the **Logic to SCADA** connection and is evaluated by the redundancy function block. It has a freely definable time out value as input. As output binary values for redundancy switch and active, passive and failed Runtimes etc. are output. They determine the reaction of the zenon Logic Runtime.

ZENON

A watchdog generated by the zenon Runtime driver is transferred using the **Logic to SCADA** connection and evaluated in zenon. This evaluation takes place in accordance with the settings of properties **Failure recognition after [s]** and **Function for failure recognition**. They define a time out time and a function (normally **Exit Runtime**).

In order for the correct watchdog to be evaluated, it must be clear for several active zenon Logic projects from which zenon Logic project the watchdog originated. If zenon Logic is started manually, by means of a zenon function, batch file or command line, the zenon project name (**ZENPROJECT=<Name>**) must also be given as a transfer parameter (on page 39).



Attention

The start of the zenon Runtime must not be performed from the zenon Editor but with a direct call from the zenon Runtime!

The zenon redundancy must be set up. Details: See help chapter Network.

5.9.2 Configure redundancy

zenon Logic and zenon can control each other in redundant operation by means of a watchdog. If the zenon Logic Runtime fails:

- ▶ the local zenon Runtime is also closed
- ▶ both systems are switched to redundancy mode

CONFIGURATION IN ZENON

To activate redundancy:

- ▶ activate property **Redundant operation** in group **Runtime** of the properties of the zenon Logic project in zenon
- ▶ define
 - **Failure recognition after [s]**
 - **Function for failure recognition**

CONFIGURATION IN ZENON LOGIC

The settings for redundancy are made in tab Redundancy (on page 34) of the zenon Logic Runtime.

WATCHDOG

With the watchdog each side checks the other for changes to this counter. If the counter for a projected time is not changed, this is reported to the program:

- ▶ zenon: a function configured in an integrated project is carried out (**Function for failure recognition**)
- ▶ zenon Logic: A function block **Output** becomes active

The watchdog examination starts as soon as the other side has changed its watchdog for the first time. This is necessary because the time is needed to start or the zenon Logic Runtime does not necessarily need to start with the zenon Runtime.



Information

If zenon Logic Runtime is started manually, the following command line settings (on page 39) must be made:

- ▶ zenon path:
ZENPATH=<Pfad>
- ▶ Name zenon project:
ZENPROJECT=<Name>
- ▶ Information on zenon network project:
REDENABLE=<0,1>
REDZENON=<0,1>

WATCHDOG IN ZENON

zenon checks zenon Logic Runtime Watchdog if:

- ▶ zenon Logic Runtime executes code, timeout is $\neq 0$ and Runtime can be assigned to an integrated zenon Logic project
- ▶ zenon Logic Runtime is stopped
- ▶ zenon Logic Runtime is passive or started passively
- ▶ zenon Logic Runtime is ended and was previously passive or active

FUNCTION FOR RECOGNIZING FAILURES

With the **Function for failure recognition** property, a function is selected that is executed as soon as the timeout defined in **Failure recognition after [s]** for the watchdog has expired. The watchdog is only rechecked if Runtime is registered again.

If a function from another project is used, the project must already have been loaded when the project with the integrated zenon Logic project is loaded.

LOGGING

The zenon Logic zenRt driver connection can be logged with the Diagnosis Viewer. The module name is zenon Logic. Each message contains the name of the zenon project and the name of the zenon Logic project. The logging includes:

- ▶ Error
 - `zprj:<z-projekt> sprj:<s-projekt> Runtime < Runtime process ID> watchdog expired.`
Runtime watchdog was not operated by Runtime for too long.

- `zprj:<z-projekt> sprj: <s-projekt> Runtime: < Runtime process ID>`
Unknown zenon Logic project.
zenon Logic project cannot be assigned to an integrated zenon project.

► Warnings

- `vm: <Vm-Prozess-ID> zenon project '<Name>' unknown`
Name of the zenon Logic project not known in zenon.

WATCHDOG IN ZENON LOGIC RUNTIME

The following applies for zenon Logic watchdog:

- The zenon watchdog is checked regularly as long as the timeout of the zenon failure recognition is $< > 0$.
This check takes place if:
 - Code is executed
 - Runtime is active or passive.

FUNCTION BLOCK ZENRT_RED AND REDSWITCH

In zenon Logic it is possible, with the help of function blocks, to check whether there is a connection to a zenon server and if it is possible to switch the zenon Logic state if necessary.

The function block **zenrt_Red** checks via the timeout information from driver "**straton_to_zenon_Runtime_Connection**" whether zenon Logic is connected to a zenon Server. If this is not the case, you can switch the affected zenon Logic to *inactive* with the help of function block **RedSwitch** and switch the Standby to *active*.

For more information about function blocks see the zenon Logic online help.

TIME HANDLING IN FUNCTION BLOCK ZENRT_RED

The number of switch attempts increases at the cycle rate of the timeout. If the zenon Logic Runtime project cannot be assigned to a zenon network project, the network timeout time (**Timeout [s]** property in **Network** group) is added to the failure recognition time (**Failure recognition after [s]** property). The time that the **ERROR** output is active is not **MAX_SWITCH * cycle** but **MAX_SWITCH * timeout [+ network timeout]**.

The non-process-handling, non-dominant zenon computer is considered to be the Standby (Server 2). The non-process handling, dominant computer (server starts as standby) is not considered a standby.

5.10 OEM specification

In this OEM specification the possibilities of the COPA-DATA zenon Logic Runtime are described which are referred to in the zenon Logic Workbench help with "**see OEM specification**".

CONFIGURATION (PROJECT PROPERTIES)

As a standard the zenon Logic Workbench contains several features and an extensive library of functions and function blocks.. Depending on the Runtime configuration, some features or blocks may be not available. Therefore it is possible to select configurations for the Workbench in order to highlight the features which are not supported by the Runtime.

The respective configuration of Runtime can be loaded in the zenon Logic workbench. To do this, select the **Configuration** entry in the Workbench in the context menu of the loaded project.

FILE ADMINISTRATION FUNCTION (FUNCTION BLOCKS)

This function makes it possible to read and write files sequentially.

Possible paths depend on the operating system used. Some folders are write protected if UAC (User Account Control) is turned on.

The following functionality is supported by zenon Logic Runtime versions:

Function	CE	RT (PC)	RTK (PC)	Implementation
F_ROPEN	X	X	X	not blocking
F_WOPEN	X	X	X	not blocking
F_AOPEN	X	X	X	not blocking
F_CLOSE	X	X	X	not blocking
F_EOF	X	X	X	not blocking
FA_READ	X	X	X	not blocking
FA_WRITE	X	X	X	not blocking
FM_READ	X	X	X	not blocking
FM_WRITE	X	X	X	not blocking
FB_READ	X	X	X	not blocking
FB_WRITE	X	X	X	not blocking
F_EXIST	X	X	X	not blocking
F_GETSIZE	X	X	X	not blocking

F_COPY	X	X	X	not blocking
F_DELETE	X	X	X	not blocking
F_RENAME	X	X	X	not blocking
LogFileCSV	X	X	X	not blocking
F_EXIST	X	X	--	blocking
F_GETSIZE	X	X	--	blocking
F_COPY	X	X	--	blocking
F_DELETE	X	X	--	blocking
F_RENAME	X	X	--	blocking

Key:

X: supported

--: not supported

SFC EXECUTION

The execution of the SFC program equals that of a T5 standard target system und follows the information of the zenon Logic Workbench help.

The exclusiveness of transitions within a divergence is supported.

INTERLINK APPLICATIONS - BINDING

A maximum of 65535 Binding variables can be published in the network. However this value is reduced by every variable which is sent to zenon using the event service.

The standard IP port number of a zenon Logic Runtime is 9000.

PROJECT PROPERTIES - "C" COMPILER

This feature is supported by all zenon Logic target systems (RT, CE, RTK). All C compilers which are normally proposed by the zenon Logic Workbench can be used. **Attention:** Only use the default name of the created Runtime-DLL.

LOGFILECSV (FUNCTION BLOCK)

See first point: File administration function (function blocks).

ADDITIONAL OEM RELEVANT FEATURES AND FUNCTIONS:

Feature	RT (PC)	RTK (PC)	CE
DTAT (function block)	x	x	x
TCP/IP management functions	x	x	x
Dynamic memory allocation function	x	x	x
PrintF (Function)	x	x	x
Step by Step Debugging	x	x	x
Functions for real time clock administration	x	x	x
Digital sampling recording	x	x	x
UDP management function	x	x	x
T5 registry management functions	-	-	-
DTFORMAT (Function)	x	x	x
GETSYSINFO (function) - _SYSINFO_CYCLESTAMP_MS	x	x	x
Variable status bits	x	⁻¹	x
DAY_TIME (Function)	x	x	x
SET_DAY_TIME (Function) ²	x	-	x

Key:

- ▶ x: Supported
 - ¹: Driver which support the **variable status bits** are only emulated or not supported by the zenon Logic RTK. Therefore no access is granted to the status bits or the time stamps.
 - ²: zenon internally works with UNIX time. Therefore only system times within their validity are allowed (0-MAXDINT starting with 1.1.1970 0:00 GMT). Data which refer to an earlier point in time affect the functionality of the zenon Runtime.
- ▶ -: Not supported

RETAIN VARIABLES (REMANENT FLAGS)

Retain variables saved at the defined saving location - normally in the instance path (hand over parameter `PATH`) - when the Runtime is stopped or closed in a normal way. Depending on the system, it can be the hard disk or a flash card.

With function "WriteHot" retain data and hot restart data can be saved explicitly, e.g. after every change or cyclically.

Some zenon Logic I/O drivers support saving the RETAIN values to the SRAM (battery buffered RAM).

In order for the RETAIN values to be loaded when the PLC application is restarted, the zenon Logic Runtime must be started with the corresponding handover parameters/settings (cold start - load RETAIN variables).

You can find information regarding the configuration of Retain data with the zenon Logic Runtime in the chapter settings (on page 30)/general (on page 31).

I/O DRIVER

You can find more information about supported I/O drivers here: zenon Logic I/O driver support (on page 76)

You can find information about the support of Online Change or Hot Restart here: zenon Logic I/O driver: Hot-Restart/Online Change (on page 78)

5.10.1 Support for the printf function

zenon Logic Runtime supports the "**printf**" function to track messages whilst the program is being executed. These messages are displayed in the output window of the Workbench. To do this, the Workbench be online in zenon Logic Runtime.

However this only works if, in the Diagnosis Viewer, the logging level "**Messages**" is active for all modules on the Diagnosis Client. You can find details on these logging levels in the Tools manual - Diagnosis Viewer, in the Diagnosis Client chapter.

The **printf** command works with the simulator of the zenon Logic Workbench without additional configuration.

5.11 Forwarding time stamps and the variable status information

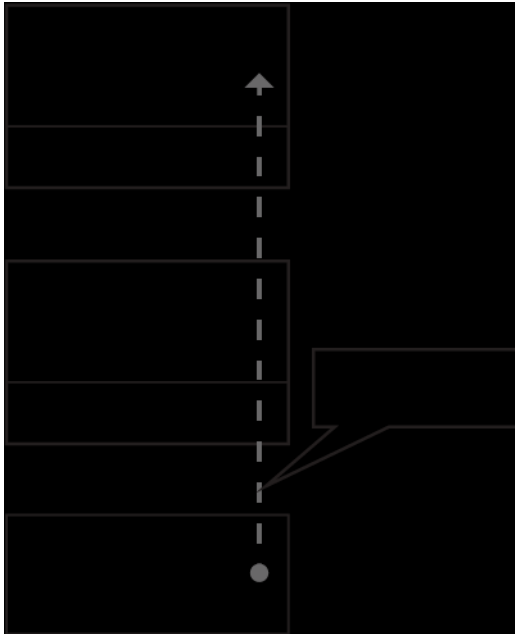
As of version 6.22 SP1 Build 3 the zenon Logic Runtime automatically forwards time stamp and variable status information from the zenon Logic I/O driver to the event handler of the zenon Logic Runtime and thus via **straton32** to zenon. Therefore in zenon (e.g. CEL) the time stamp of the source (I/O driver slave or server) are displayed and not the time stamp of the zenon Logic Runtime.

The following prerequisites are necessary:

- ▶ The variable must have a time stamp in the zenon Logic I/O driver. It depends on the protocol. Some drivers do not provide time stamps.

- The zenon Logic project must have been compiled and loaded with compiler flag `Create status bit for variable with profile` turned on.

Overview forwarding the time stamp:



5.12 zenon Logic I/O driver support

I/O drivers which are supported by different zenon Logic Runtime types:

Drivers	RT (PC)	RTK (PC)	CE
Advantech ADAM 5550	--	--	--
Anybus Konfiguration	X	E	--
AppicomIO Konfiguration	X	E	--
AS-interface	X	--	--
BECKHOFF CX1000	--	--	X
Brodersen IOTOOLS	X	E	--
CAN-bus	--	--	--
Hilscher CIF Profibus	X	X ¹	X
Hilscher SYCON Konfiguration	X	E	X ²
Hilscher SYCON Konfiguration [universal]	X	X	X
Hilscher SYCON.net Konfiguration	X	E	--
IEC 60870 Slave	X	--	X
IEC 61850 Client	X	E ¹	X
IEC 61850 Server	X	--	X
Interbus-S	X	E	--
MODBUS Master Protokoll	X	E	X
MODBUS Slave Protokoll	X	E	X
PROFINET IO	X	E	X
SoftNet ProfibusDP	X	E	--
Logic zu SCADA Verbindung	X	--	X
ThinkIO/IO System 758	--	--	X
Wago 750-860	--	--	--
Weihenstephan Standard - Client	X	E	X
Win32 Shared memory	X	--	--
XFlow	--	--	--

KEY

- ▶ X: Fully supported
- ▶ --: Not supported
- ▶ E: Carried out as emulation (performance loss possible, availability is limited in the PLC cycle, driver selection at a BSOD)
 - 1: The driver-specific function modules do not work with the zenon Logic RTK.

- 2: Hilscher DLLs must be available on the target system.

5.13 zenon Logic I/O driver: Hot-Restart/Online Change

The zenon Logic I/O drivers support the Hot-Restart or Online Change in different ways:

Drivers	Hot-Restart	Online Change	Online Change including change of the bus driver configuration
Anybus Konfiguration	X	X	X ¹
ApplicomIO Konfiguration	X	X	X ¹
AS-interface	X	X	X ¹
BECKHOFF CX1000	X	X	--
Brodersen IOTOOLS	X	X	X ¹
Hilscher CIF Profibus	X	X	--
Hilscher SYCON Konfiguration	X	X	X ¹
Hilscher SYCON Konfiguration [universal]	X	X	X ¹
Hilscher SYCON.net Konfiguration	X	X	X ¹
IEC 61850 Client	X	X	--
Interbus-S	X	X	--
MODBUS Master Protokoll	X	X	X ¹
MODBUS Slave Protokoll	X	X	X ¹
PROFINET IO	X	X	--
SoftNet ProfibusDP	X	X	X ¹
straton zu zenon Runtime Verbindung	X	X	-- ²
ThinkIO/IO System 758	X	X	--
Weihenstephan Standard - Client	X	X	X ¹
Win32 Shared memory	X	X	X ¹

LEGEND

- ▶ X: Functionality available
- ▶ --: Functionality not available

- 1: Online Change including the change of the bus driver configuration is always followed by the reinitialization of the I/O driver. Because of the reinitialization I/Os can fail for a short period!
- 2: No need for change in the bus driver configuration.

5.14 Error message

Errors during the execution are logged in the log file.

Error message	Description
General error.	Error which could not be specified any further.
Invalid application code - bad header.	STRATON.cod file is invalid. Header invalid.
Invalid application code - bad target ID.	STRATON.cod file is invalid. Target ID invalid.
Invalid application code - bad version.	STRATON.cod file is invalid. Version invalid.
Unknown communication request.	Host and Runtime not compatible (T5 protocol).
Cannot process read request.	Incoming communication request cannot be processed.
Unknown TIC code.	Invalid instruction in file straton.cod .
Too many DATA8 used.	Too many SINT/USINT variables used.
Too many DATA16 used.	Too many INT/UINT variables used.
Too many DATA32 used.	Too many DINT/UDINT/REAL variables used.
Division by 0 in TIC (32bit).	Division by 0 in the PLC logic of the user.
Too many TIME variables.	Memory overflow for online change.
Infinite loop in TIC (safe mode).	Not closed loop in the application code.
Standard function used, which is not supported by this runtime.	PLC logic of the user refers to a function which is not available in this version of the Runtime.
Division by 0 in TIC (64bit).	Division by 0 in the PLC logic of the user.
Too many DATA64 used.	Too many LINT/LREAL variables used.
Too many strings.	Memory overflow for online change.
Division by zero in TIC (8 bit).	Division by 0 in the PLC logic of the user.

Division by zero in TIC (16 bit).	Division by 0 in the PLC logic of the user.
HOT restart: System resources used.	Hot restart not possible.
Invalid variable map in hot restart.	Hot restart not possible.
Bad build stamp for hot restart.	Hot restart not possible. Invalid straton.cod version.
SFC changed - no hot restart!	Hot restart not possible.
Retain definition has changed - no hot restart.	Hot restart not possible.
Bad DB format - no hot restart!	Hot restart not possible.
Cycle time overflow.	Cycle time overflow. For details, see the Cycle time error messages section.
Can not store retain variables.	Not enough memory in order to save retain variable.
Array index out of bounds.	Access to invalid array index.
Recursive call of sub-program.	A sub-program calls up itself.
Too many external variables used.	Memory overflow for online change.
Too many tasks.	Memory overflow for online change.
Call stack overflow (hot).	Memory overflow for online change.
Stack overflow.	Too many call ups of intricate sub programs.
Can not hot restart (ASi).	Hot restart not possible (Asi bus driver).
Online change not supported.	Online change not possible
Online change failed - Bad code.	Online change not possible. Invalid straton.cod version.
Online change - Not the same application.	Online change not possible
Online change failed - IO drivers.	Online change not possible
Online change failed - CT segment changed.	Online change not possible
Online change failed - bus drivers.	Online change not possible. Bus driver does not permit online change.
Online change failed - variable map / profiles.	Online change not possible

Online change failed - tasks.	Online change not possible
Online change failed - MODBUS.	Online change not possible
Online change failed - CAN bus.	Online change not possible
Online change failed - event production.	Online change not possible
Online change failed - change D8 alloc.	Online change not possible
Online change failed - change D16 alloc.	Online change not possible
Online change failed - change D32 alloc.	Online change not possible
Online change failed - change D64 alloc.	Online change not possible
Online change failed - change TMR alloc.	Online change not possible
Online change failed - change STR alloc.	Online change not possible
Online change failed - change FBI alloc.	Online change not possible
Online change failed - change CFC alloc.	Online change not possible
Online change failed - change XV alloc.	Online change not possible
Online change failed - STR buff size.	Online change not possible
Online change failed - STR length.	Online change not possible
Online change failed - FBI buffer.	Online change not possible
Online change failed - FBI changed.	Online change not possible
Online change failed - Program style.	Online change not possible
Online change failed - Number of programs.	Online change not possible

ERROR MESSAGE CYCLE TIME

If the **cycle time** is exceeded, messages are displayed in the zenon Logic Runtime or in the Workbench and corresponding entries are written to the LOG file.

Error message user interface	Entry log file	Meaning
Timing configuration error. (x) cycles lost.	Timing configuration error. (x1) cycles lost. Configured cycle time = (x2) us, maximum execution time = (x3) us, maximum PLC utilisation = (x4) .	Number (x) of not executed cycles due to cycle time out. The execution time is higher than the set cycle time or there are not enough CPU resources available.
Timing configuration error. PLC utilization peaked at (x)%.	Timing configuration error. PLC utilization exceeds 80%. Configured cycle time = (x1) us, maximum execution time = (x2) us, maximum PLC utilisation = (x3) % ."	The execution time is higher than 80% of the set cycle time; actually (x)%.

TROUBLESHOOTING

You can rectify the **cycle time** being exceeded by:

- ▶ **cycle time** increasing
- ▶ decreasing execution time, e.g.:
 - use a system with more CPU resources
 - Use code compiled with C-Compiler

reduce the program

6. zenon Logic Runtime Manager

The zenon Logic Runtime Manager administrates all zenon Logic Runtime projects, which are stand alone or must be started manually, on your computer. It does not matter whether there is a zenon Runtime installed on the computer.

6.1 Why should you use the zenon Logic Runtime Manager?

- ▶ If you created your zenon Logic project in zenon and you set the zenon property of the zenon Logic project RT start type to `start manually`, the zenon Logic Runtime Manager is the ideal tool for organizing the parameters for the manual start.

- ▶ If you want to start your zenon Logic Runtime project during the booting of the Windows operating system, you can set this with one mouse click in the zenon Logic Runtime Manager.
- ▶ If you want to reorganize your zenon Logic project, you can do this with the zenon Logic Runtime Manager.
- ▶ If you want to run the zenon Logic Runtime autarchically (without the zenon Runtime), it is best to call up the zenon Logic Runtime instance using the zenon Logic Runtime Manager.

6.2 Main menu

- ▶ Configurations: There you can administrate the zenon Logic Runtime projects
- ▶ Languages: There you can switch between languages online
- ▶ Settings: Enter the path of the zenon Logic Runtime
- ▶ %: Open the help
- ▶ Exit: Exits the program - **Hint:** Do not forget to save the settings before you exit the zenon Logic Runtime Manager!

6.3 Properties help

Place the mouse cursor over an element in the zenon Logic Runtime Manager in order to see the corresponding help text in the property help.

In addition all elements of the zenon Logic Runtime Managers are described here:

6.3.1 Configuration list

This list contains all available zenon Logic Runtime configurations.

- ▶ Click on an entry in the list in order to display its parameter settings.
- ▶ After that click on **Edit** in order to change the parameters.
- ▶ Click on **Save** in order to save the changes.

6.3.2 zenon Logic project folder

The selected path of the zenon Logic Runtime project (storage location of the PLC code) is entered there automatically. You can change this entry manually or by left-clicking the folder symbol.



Information

*Before you are able to edit any setting which you have already saved, you must first select the concerning configuration in the list. After that click on button **Edit**. In order to save the changes, click on button **Save**.*

6.3.3 Name of the configuration

The name of the last path of the zenon Logic Runtime files is entered automatically if a new configuration is created. You can change this name freely.



Information

*Before you are able to edit any setting which you have already saved, you must first select the concerning configuration in the list. After that click on button **Edit**. In order to save the changes, click on button **Save**.*

6.3.4 Port (Main)

Enter the main port for the zenon Logic Runtime instance. This port defines the connection of the zenon Logic Workbench and a zenon driver. Per default this port is set to 1200.



Information

*Before you are able to edit any setting which you have already saved, you must first select the concerning configuration in the list. After that click on button **Edit**. In order to save the changes, click on button **Save**.*

6.3.5 Port (Binding)

Enter the binding port for the zenon Logic Runtime instance. The binding port defines the connection for spontaneous TCP/IP communication to other zenon Logic Runtimes. Per default this port is set to 7800 higher than the main port.



Information

*Before you are able to edit any setting which you have already saved, you must first select the concerning configuration in the list. After that click on button **Edit**. In order to save the changes, click on button **Save**.*

6.3.6 Startup

Define the startup behavior of the zenon Logic Runtime instance.

- ▶ Cold start
zenon Logic Runtime starts with initialized variables.
- ▶ Cold start (load RETAIN variables)
Die zenon Logic Runtime starts initialized with remanent markers.
- ▶ Hot restart
zenon Logic Runtime starts with the variable values from the time Runtime was stopped.



Information

*Before you are able to edit any setting which you have already saved, you must first select the concerning configuration in the list. After that click on button **Edit**. In order to save the changes, click on button **Save**.*

6.3.7 Start in step mode

In order to start the zenon Logic Runtime in step mode, activate this checkbox.



Information

*Before you are able to edit any setting which you have already saved, you must first select the concerning configuration in the list. After that click on button **Edit**. In order to save the changes, click on button **Save**.*

6.3.8 Windows real time priority

In order for the zenon Logic Runtime instance to run in the Windows real time priority, activate this check box.



Information

We recommend to always have the real time priority activated. First and foremost deactivating the real time priority is used for testing purposes (e.g. if there may be infinite loops).



Information

*Before you are able to edit any setting which you have already saved, you must first select the concerning configuration in the list. After that click on button **Edit**. In order to save the changes, click on button **Save**.*

6.3.9 Hard real time

If you activate this checkbox, the zenon Logic real time PLC (zenon Logic RTK) is called up when the configuration is started. As the zenon Logic real time PLC may only run once on the computer, it is validated that this checkbox is only activated for one configuration.



Information

*Before you are able to edit any setting which you have already saved, you must first select the concerning configuration in the list. After that click on button **Edit**. In order to save the changes, click on button **Save**.*

6.3.10 Report output

Activate this checkbox in order to activate the writing of messages of the Runtime in a LOG file.



Information

Writing a LOG file will influence the performance.



Information

The LOG file is named `zenon Logic RTLOG.txt` and is saved in the zenon Logic Runtime project folder.



Information

*Before you are able to edit any setting which you have already saved, you must first select the concerning configuration in the list. After that click on button **Edit**. In order to save the changes, click on button **Save**.*

6.3.11 Auto start

In order to start a zenon Logic Runtime instance directly with the booting of the Windows operating system, activate this checkbox.



Information

*Before you are able to edit any setting which you have already saved, you must first select the concerning configuration in the list. After that click on button **Edit**. In order to save the changes, click on button **Save**.*

6.3.12 Button "New"

If you want to create a configuration, click on button **New**. After that a dialog opens in which you can select the folder in which you want to save the zenon Logic Runtime files (the PLC code). Confirm your selection with **OK**. You can change the name of the configuration. In order to save the configuration and the name of the configuration, click on **Save**.

6.3.13 Button "Edit"

Before you can change the parameters of a configuration, you must click on button **Edit** after you have selected an entry from the configuration list. In order to save the changes, you must click on **Save**.

6.3.14 Button "Save"

Click on button **Save** in order to save all changes.

6.3.15 "Delete" button

To delete a configuration, select the entry from the configuration list and click on the **Delete** button.

6.3.16 Button "Start"

In order to start the zenon Logic Runtime with the respective configuration, select a configuration from the list and click on button **Start**. The zenon Logic Runtime starts immediately.



Information

If the folder of the configuration does not contain any zenon Logic Runtime files (PLC code), the zenon LogicRuntime starts in mode `No program`.

6.3.17 zenon Logic Runtime (settings)

Enter the folder of the zenon Logic Runtime (folder of `stratonRT.exe`). In order to browse through your hard disk, left-click the folder symbol.

6.3.18 Language selection (languages)

Select the desired language by clicking on the respective flag.

6.3.19 Show splash screen (setting)

If you activate this check box, the splash screen of the zenon Logic Runtime Manager is displayed at the start.