



COPADATA
do it your way

zenon manual

zenon Logic

v.7.20





©2015 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 detail view toolbar and context menu | 8 |
| 3.4 First steps with zenon Logic in zenon | 9 |
| 3.4.1 Create a zenon Logic project in a zenon project | 9 |
| 3.4.2 Editing a zenon Logic project | 16 |
| 3.4.3 Deleting a zenon Logic project | 16 |
| 3.4.4 Design of the Workbench | 17 |
| 3.5 Variable handling | 17 |
| 3.5.1 Create O and M variables in zenon Logic | 18 |
| 3.5.2 Create O and M variables in zenon | 22 |
| 3.5.3 Using zenon variables in zenon Logic | 23 |
| 3.5.4 Variable names | 27 |
| 3.6 Arrays | 29 |
| 3.7 Data structure | 29 |
| 3.8 Distributed engineering (multiuser) | 30 |
| 3.9 Reusing projects | 31 |
| 4. zenon Logic Workbench | 31 |
| 5. zenon Logic Runtime (Soft PLC) | 31 |
| 5.1 User interface | 32 |
| 5.1.1 Slider Allowed maximum | 33 |
| 5.1.2 Settings | 33 |
| 5.1.3 Messages | 40 |
| 5.2 Licensing | 41 |
| 5.2.1 TAG handling zenon - zenon Logic | 42 |
| 5.2.2 Demo mode | 42 |

| | | |
|-----------|--|-----------|
| 5.3 | Start the zenon Logic Runtime | 43 |
| 5.4 | zenon Logic Runtime cycle | 48 |
| 5.4.1 | Error messages cycle time | 50 |
| 5.5 | Close the zenon Logic Runtime | 50 |
| 5.6 | Creating the hot restart file | 51 |
| 5.7 | Remote systems | 52 |
| 5.7.1 | zenon Runtime and zenon Logic Runtime on one remote PC | 52 |
| 5.7.2 | zenon Logic Runtime autarkic (PC) | 53 |
| 5.8 | zenon Logic Runtime with Real Time Kernel (RTK) | 54 |
| 5.8.1 | Restrictions and requirements of the zenon Logic RTK | 57 |
| 5.8.2 | zenon Logic RTK configuration | 59 |
| 5.8.3 | Blue Screen Handling..... | 66 |
| 5.8.4 | Error codes at starting the zenon Logic RTK | 67 |
| 5.9 | Redundancy zenon Logic and zenon | 70 |
| 5.9.1 | Requirements | 70 |
| 5.9.2 | Configure redundancy | 72 |
| 5.10 | OEM specification | 75 |
| 5.11 | Forwarding time stamps and variable status information | 78 |
| 5.12 | zenon Logic I/O driver support | 79 |
| 5.13 | zenon Logic I/O driver: Hot-Restart/Online Change | 81 |
| 5.14 | Error messages..... | 82 |
| 6. | zenon Logic Runtime Manager | 86 |
| 6.1 | Why should you use the zenon Logic Runtime Manager? | 87 |
| 6.2 | Main menu | 87 |
| 6.3 | Properties help..... | 87 |
| 6.3.1 | Configuration list | 87 |
| 6.3.2 | zenon Logic project folder | 88 |
| 6.3.3 | Name of the configuration | 88 |
| 6.3.4 | Port (Main) | 88 |
| 6.3.5 | Port (Binding) | 89 |
| 6.3.6 | Startup..... | 89 |
| 6.3.7 | Start in step mode | 89 |
| 6.3.8 | Windows real time priority..... | 90 |
| 6.3.9 | Hard real time..... | 90 |
| 6.3.10 | Report output..... | 90 |

| | | |
|--------|--------------------------------------|----|
| 6.3.11 | Auto start..... | 91 |
| 6.3.12 | Button "New" | 91 |
| 6.3.13 | Button "Edit" | 91 |
| 6.3.14 | Button "Save" | 92 |
| 6.3.15 | Button "Delete" | 92 |
| 6.3.16 | Button "Start" | 92 |
| 6.3.17 | zenon Logic Runtime (settings) | 92 |
| 6.3.18 | Language selection (languages)..... | 92 |
| 6.3.19 | Show splash screen (setting) | 92 |

1. Welcome to COPA-DATA help

GENERAL HELP

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

PROJECT SUPPORT

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

LICENSES AND MODULES

If you find that you need other modules or licenses, our staff will be happy to help you. Email sales@copadata.com (<mailto:sales@copadata.com>).

2. zenon Logic

zenon Logic is the programming environment in accordance with IEC 61131 included in zenon. It is available as Editor (on page 7) and Runtime (Soft-PLC) (on page 31).



License information

Must be licensed in Editor and Runtime.

Note: 8 TAGs are included in the standard license. For details about licensing see chapter Licensing (on page 41).

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.

CONTEXT MENU PROJECT MANAGER

| 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. Attention: 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 |
| FBD | Function Block Diagram |
| LD | Ladder Diagram |
| ST | ST – Structured Text |
| IL | Intstruction List |

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. The zenon Logic soft PLC is available for the PC and for Windows® CE platforms. Beside the soft PLC there are also hard PLCs, which have implemented the zenon Logic Runtime kernel as OEM software. The manufacturers of these hardware PLCs can be found on www.copalp.com (<http://www.copalp.com>).

You can find more information about zenon Logic here: zenon Logic Runtime (Soft PLC) (on page 31)

3.3 zenon Logic detail view toolbar and context menu



TOOL BAR

| Parameters | Description |
|---------------------------------------|---|
| New zenon Logic project | Opens the dialog to create a new project (on page 9), 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. Attention: At the XML import you must deactivate online change so that the reload of the zenon Logic variable works. |
| Remove all filters | |
| 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 for the selected entry. |
| Import XML | Imports entries from an XML file. |
| Remove all filters | Removes all filter settings. |
| Help | Opens online help. |

3.4 First steps with zenon Logic in zenon

Here the first steps in order to create a zenon Logic project in zenon are explained. In addition information is given on how to configure and edit the project in zenon. If you work for the first time with zenon Logic in zenon, we recommend you follow this manual.

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 tool bar 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: You will
 - Not transferred to the project server

- Deleted again in the next synchronization

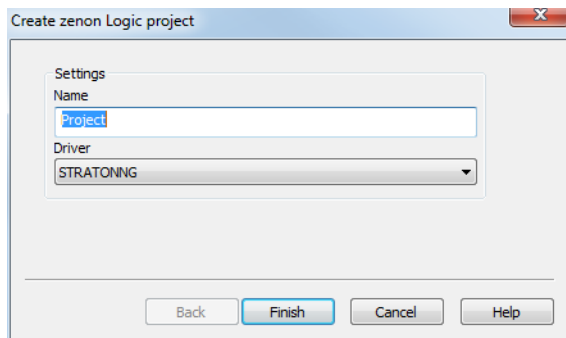
Attention: Multi-user zenon Logic projects can only be created using the zenon Editor.



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



| Parameters | 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: ▸ all numbers and special characters such as \ / , ; ß ä ö ü : * ? " < > () ▸ a dot may only be used within the name not at the beginning of the name ▸ Button Finish is only active if an valid name was entered. |
| Driver | <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 18) 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 help for the driver selection:

| Driver | Properties |
|--------------------------|--|
| straton32 driver: | <p>Ethernet TCP/IP based.</p> <p>Advantages:</p> <ul style="list-style-type: none"> ▶ can be used cyclic or spontaneous (on change) ▶ supports complex data types (structure data types or arrays) |
| stratonNG driver: | <p>Default driver for zenon Logic projects. Fulfills all requirements just like straton32.</p> <p>In addition:</p> <ul style="list-style-type: none"> ▶ enables several connections with one driver to several Runtimes at the same time (recommended for using with Windows CE) ▶ works more performant than the straton32 driver |
| IEC870 driver: | <p>It is mostly used in the infrastructure automation.</p> <p>Properties:</p> <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.

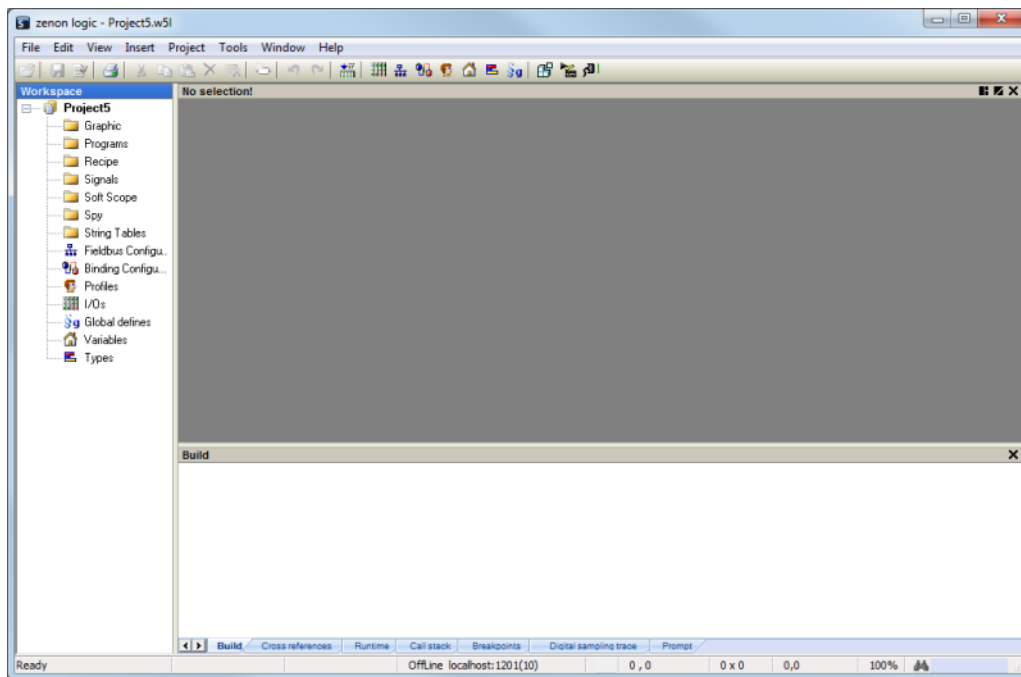
CHANGE DRIVER

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

- ▶ select the detail view of node **Variables/Driver** of the zenon Logic project with the **straton32** driver
- ▶ select **Change driver** in the context menu
- ▶ the dialog for the selection of the driver is opened
- ▶ you can only select the **stratonNG** driver
- ▶ select the driver
- ▶ close the dialog
- ▶ the configuration is taken over in 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 tool bar
(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.)

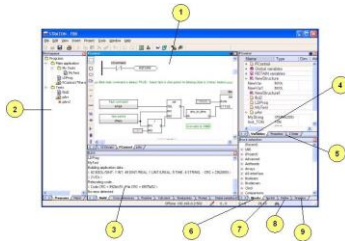


Attention

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 property tab and the Z arrangement are used for the engineering 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:

3.5 Variable handling

The intelligent integration of zenon Logic in zenon offers considerable advantages in engineering 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

External UDFBs from other zenon Logic projects are not part of the zenon Logic project and therefore not available in zenon.

3.5.1 Create O and 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
- ▶ UDFB instances (User Defined Function Block)

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



Attention

We recommend that when using UDFB instances as O&M variables to administrate only one zenon Logic project in a zenon project.

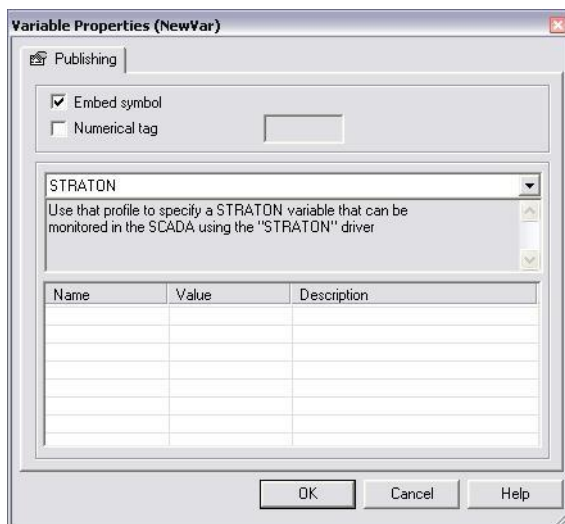
If two or more zenon Logic projects are administrated in zenon, the as O&M marked UDFBs appear as structure data types.

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. Activate checkbox **Embed symbol** in the following dialog.
4. Change the profile in the combo box from **None** to **STRATON** (*Operating and Monitoring for zenon*).
5. Click on OK.



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>
```

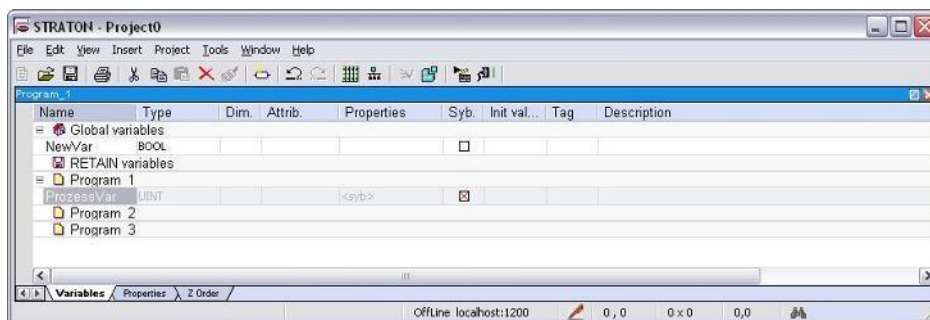


Attention

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

ACTIVATE IN VARIABLE LIST

You can set the profile and the flag directly in the variable list of zenon Logic. Thus you do not have to open the context menu and the dialog.



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*). In zenon the contents of identification and description can be found in the variable properties **Identification** and **Resource label**.

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 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).

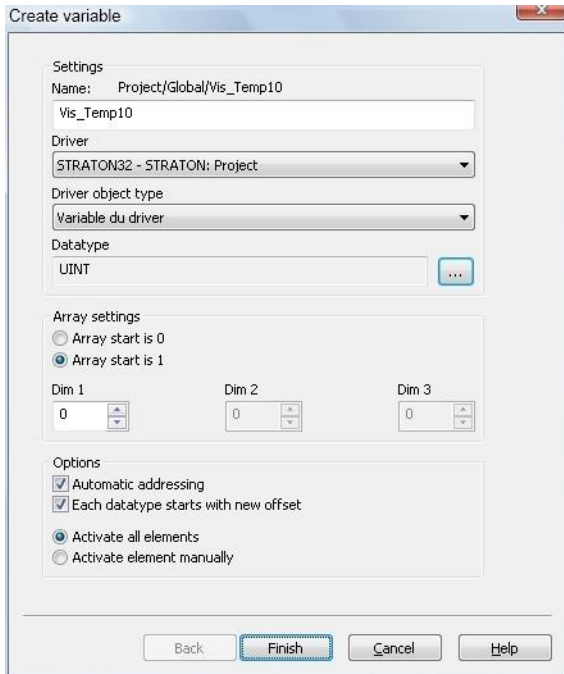
Note: If an I/O group is deleted in zenon Logic, the variables of the I/O board with zenon Logic profile are also deleted in zenon.

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 and 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 27)



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 deleted:

- ▶ 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

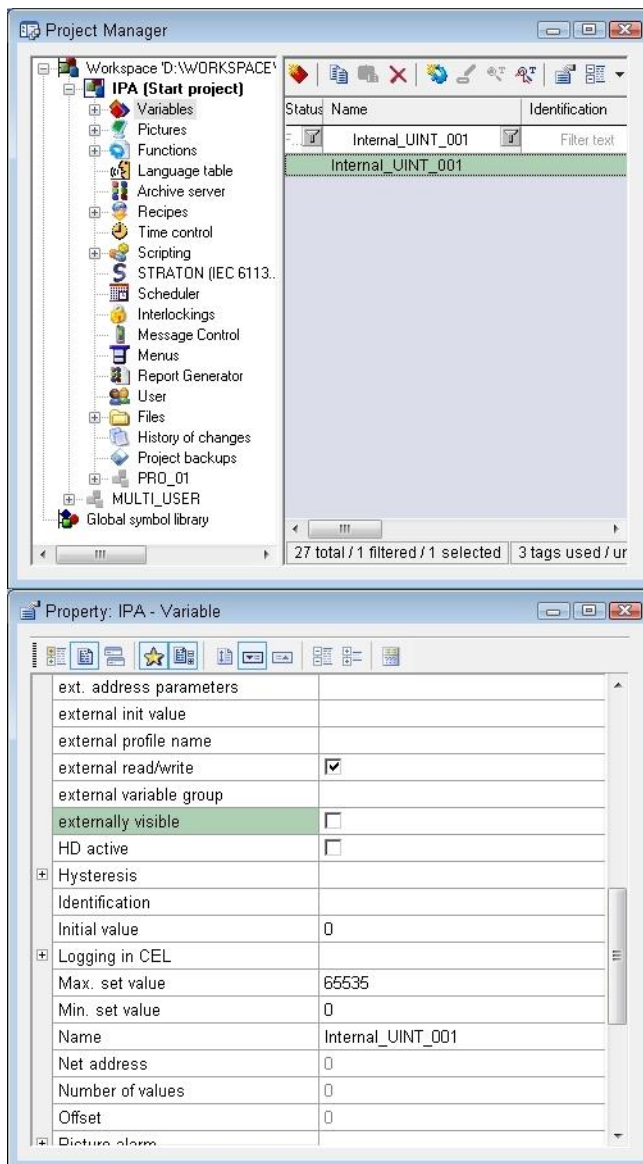
This behavior is regardless of whether the changes affects the zenon Logic project.

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.

One possibility to realize this functionality is the allocations (see Allocations). 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.

PLC variables (e.g. from a S7) can be exchanged between zenon and zenon Logic via a shared memory connection bidirectionally (!), easier and quicker . In the properties of the PLC variable the flag **Externally visible** has to be set to make the variable available in the zenon Logic Workbench.



As soon as this flag is set, the variable is displayed in the variable list of the zenon Logic Workbench.

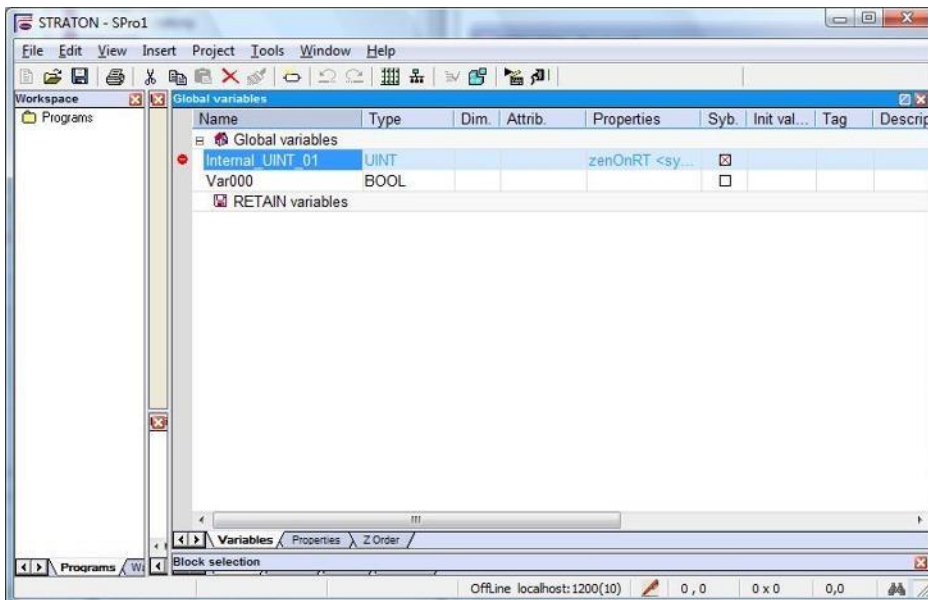


Attention

Regard interactions with property **Hysteresis**:

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:

- ▶ If the value changes in the PLC and stays within the configured hysteresis, it is not communicated to the zenon Runtime and therefore also not to zenon Logic.
- ▶ If the value is changed by the user in zenon (e.g. via a function) and the value stays within the configured hysteresis, the value remains the same in zenon and in zenon Logic but it changes in the PLC.
- ▶ If the value is changed in zenon Logic and stays within the hysteresis, the value in the PLC and in zenon Logic changes. It remains however the same 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.



Such a variable is especially marked in zenon Logic, as it cannot 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. S7).



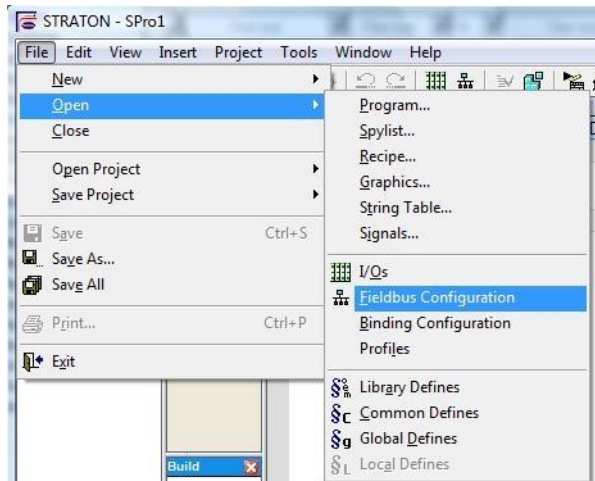
Attention

Be careful with the variable name! The name of the variable from an external PLC has to be IEC 61131-3 compatible. Also see *Variable names* (on page 27).

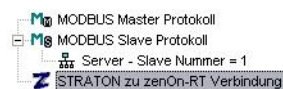
In zenon Logic a so-called I/O driver - shared memory driver - must be loaded in order to make the communication for such variable possible in the Runtime.

In order to do this open the fieldbus configuration in the zenon Logic Workbench with File -> Open.... -> Fieldbus configuration.

The following dialog opens with the menu Insert -> Insert configuration...:



Here select the configuration `straton to zenon-RT connection`. So this configuration is inserted in the tree of the fieldbus configuration:



Now it is guaranteed that the externally visible variables from zenon communicate with zenon Logic.



Information

Please be aware, that such a connection only works locally on a PC or a CE device. Here it is not possible to communicate with a zenon Logic Runtime on another device.

ARRAYS AND STRUCTURE VARIABLES

With the help of the zenrt driver complex variables (arrays and structure variables) are resolved into a flat structure. The values of variables with the same name in zenon Logic and zenon are exchanged via the shared memory driver.



Information

The functions for checking the status of a variable and for setting Dirty are not supported at the moment.

The status information (time stamp and status bits) are not supported for complex data types.

If you marked an element of an array or a structure variable as externally visible, a simple variable with the same name is created in the zenon Logic Workbench. If you marked the array or the structure variable itself as externally visible, this complex variable is created in the workbench.



Information

If you mark an array or a structure variable as externally visible, the individual elements are not created in the Workbench anymore.

You can create variables in the workbench, which contain characters '.' or '[' in their name. Therefore you can create variables which have the same name as variables which derive from an array or a structure variable. In this case the communication with the zenrt driver is not possible as no distinct allocation can be made.



Attention

Array variables in zenon which are based in the straton32 or stratonNG driver must not start with 1. A zenon array with dimensions 1, 2, 2 is created as a simple variable in zenon Logic.

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).

For variables of other drivers (e.g. S7 TCO/IP driver), the engineer has to check that the variable name fulfills the regulations of the IEC 61131-3 (communication takes place using shared memory).

EXAMPLES FOR THE IEC 61131-3 NOTATION

| Permitted name | Impermissible name | Invalid because: |
|----------------|--------------------|--------------------------|
| _XYZ | __XYZ | Two underscores. |
| xy_z | xy__z | Underlined twice. |
| 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 following 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 18) and Creating O and M variables in zenon (on page 22).



Attention

Array variables in zenon which are based in the straton32 or stratonNG driver must not start with 1. A zenon array with dimensions 1, 2, 2 is created as a simple variable in zenon Logic.

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 1024 characters.

3.7 Data structure

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 27). Here the spelling is not checked by zenon!



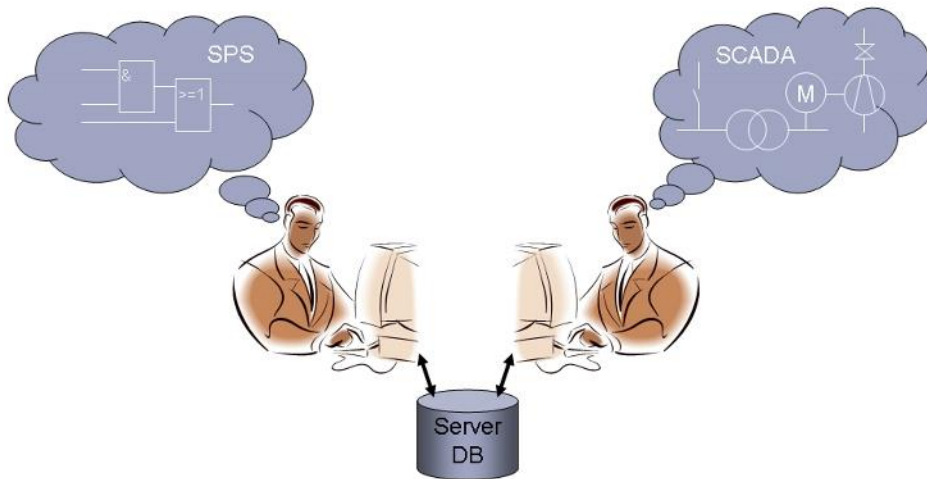
Information

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 a embedded data type.

3.8 Distributed engineering (multiuser)

It is possible that more than one engineer are working together on one zenon project. Thus for example one 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 zenon Logic (*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.

DRIVER

zenon Logic 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 STARTON driver (**enable changes**).

ZENON LOGIC PROJECTS

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

DELETE ZENON LOGIC PROJECT

zenon Logic projects can also be deleted:

- ▶ if they are checked out (**enable changes**)
- ▶ in Offline mode

3.9 Reusing projects

zenon Logic projects which were created with the **stand-alone** Workbench of COPA-DATA can be integrated easily in a zenon project.

Select in the context menu of the detail view of the module **zenon Logic (IEC 61131-3)** menu item **External zenon Logic Import project...** With this command you can indicate an existing zenon Logic project (mark folder) and integrate it in the existing zenon project.

Another possibility is the XML import. In the context menu of the detail view of the module **<CA_PRODUCTNAME> (IEC 61131-3)** you will find the command **Import XML**. Thus zenon Logic projects which were first exported in XML can easily be imported in the zenon project.

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)

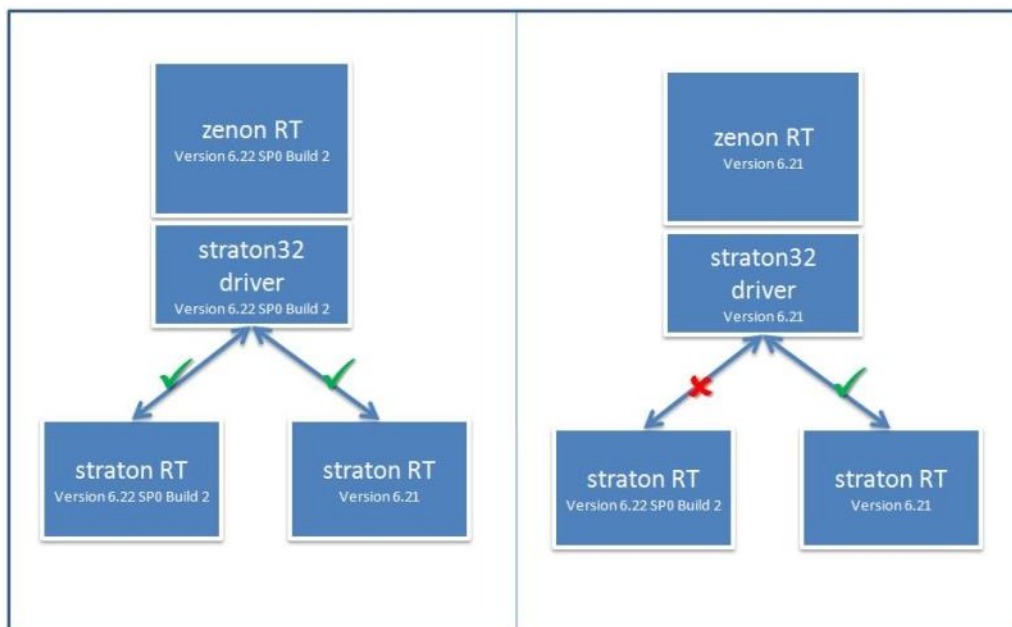
The zenon Logic Runtime interprets the compiled PLC Code of the zenon Logic Workbench. The Runtime runs on Windows 7 and Windows 8 Embedded PCs, as well as Windows CE platforms.

This chapter gives an overview of the use of the zenon Logic Runtime and the configuration tool zenon Logic Runtime Manager.



Attention

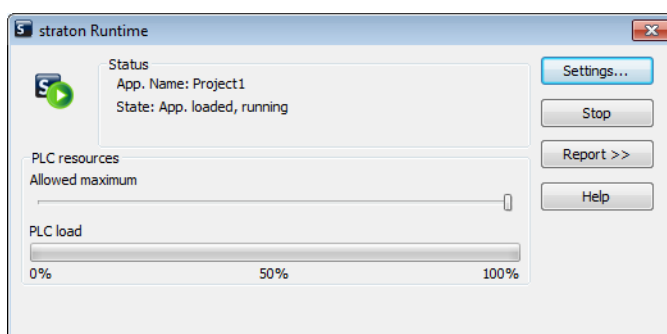
Due to performance issues the zenon Logic Runtime Version 6.22 SP0 Build 2 needs the straton driver for the same or earlier zenon versions. Earlier zenon Logic Runtime versions still work with all straton drivers.



5.1 User interface

Usually the zenon Logic Runtime can only be found as a symbol in the system tray. A double-click on the symbol with the green arrow or on menu item *Show* in the context menu opens the zenon Logic Runtime user interface.

Depending on the current status, the user interface displays different states (running, step mode, break point(s), stopped).



A bar chart 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 33) button : Makes configuration possible via three tabs
 - General settings (on page 34)
 - Redundancy (on page 37)
 - Advanced users (on page 38)
- ▶ **Start/Stop** button: opens the dialog to start Runtime or stops Runtime
- ▶ **Messages** (on page 40) 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.

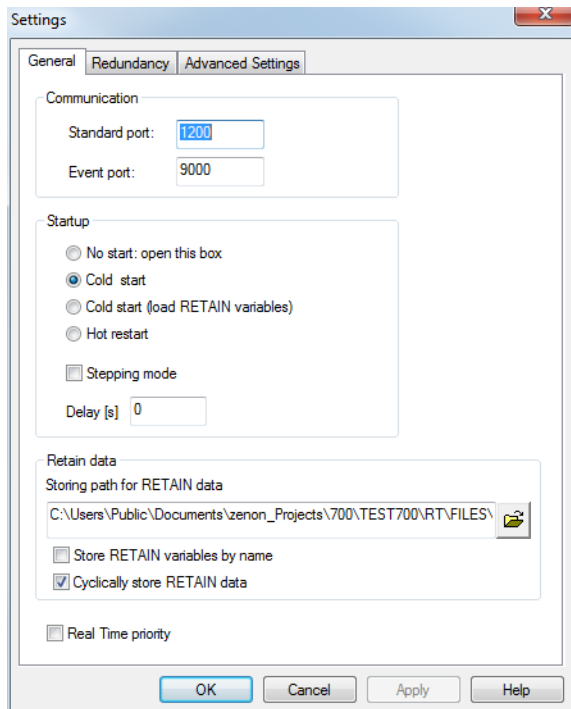
5.1.2 Settings

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:



| Parameters | 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 the zenon Logic Runtime for the on change data transfer (e.g. CA_PRODUCTNAME 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 whether 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 <i>embedded</i> in the <CA:PRODUCTNAME> Workbench. If you do not activate the check box, 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 <i>Realtime</i> . If the flag is not set, it means priority <i>Normal</i> . |



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.*

Hint: 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.

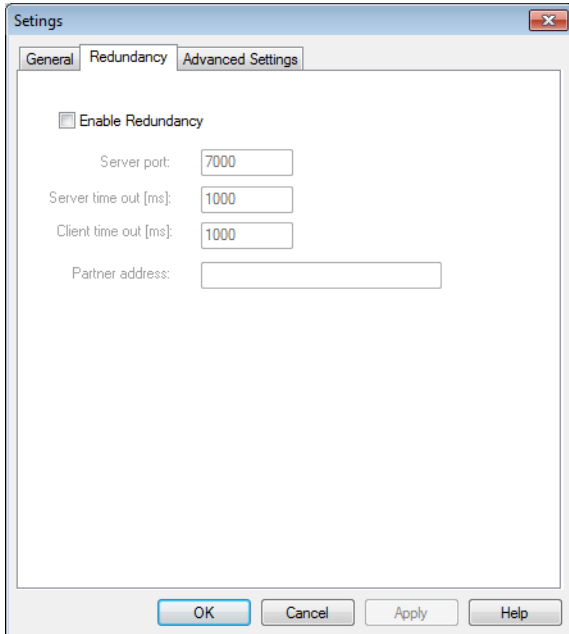
Please make the changes as usual in the zenon Logic project properties of the zenon Editors or in the zenon Logic Runtime Manager.

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 Time-out [ms] | Time-out in order to determine if the passive client is available. After the time-out expires, the replication is stopped. |
| Client Time-out [ms] | Time-out in order to determine that the active server is no longer available. After the time-out expires, the passive client becomes the active server. |
| Partner address: | IP address of the redundancy partner. |



Information

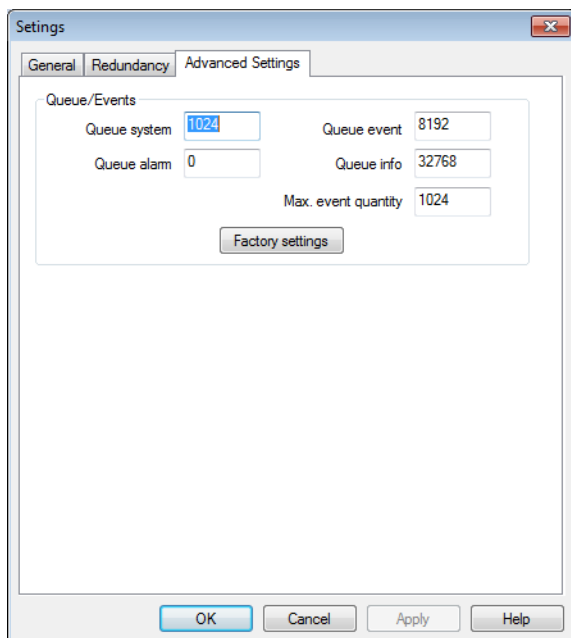
You can find details on configuring the redundancy of zenon Logic and zenon in chapter Redundancy zenon Logic and zenon (on page 70).

LIMITATIONS

Redundancy is not supported in zenon Logic RTK Runtime.

Enhanced 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 |
| max. connections | Number of possible connections (cannot be changed) | 4 ... 65535 Default: 256 |
| Events | Length of the event queue: In this queue the events of the zenon Logic Runtime event servers are written. 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 much many variables are handled by the event server per cycle. | 128 ... 65535 Default: 1024 |
| Reset | Restores the default settings | |



Information

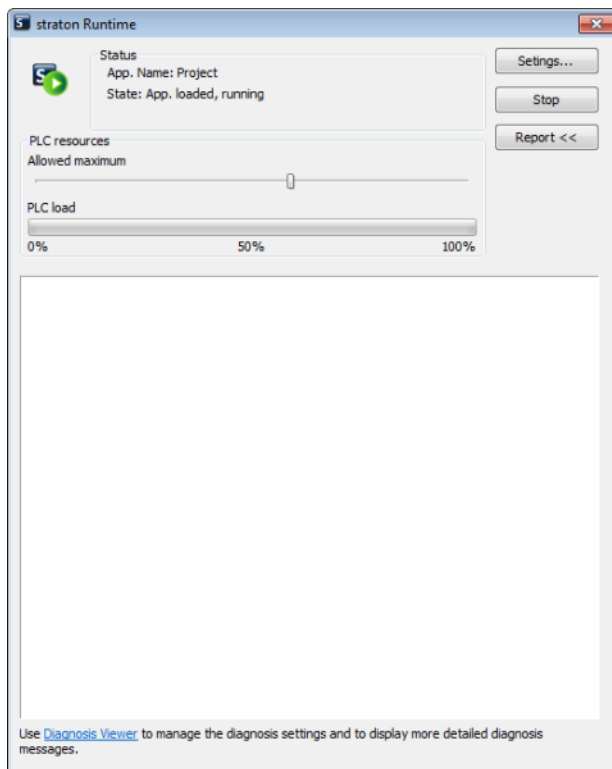
*If in large projects more than 1024 variables are transferred between zenon Logic and zenon with the `straton32.exe` driver, the parameter **Max. buffer** must be increased accordingly. An easy method to find out the number of **Max. buffer** is the value for the compiled I/Os in the output window of 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 OEM specific runtime: T5CD
< 132 BOOL/SINT; 1 INT; 11 DINT/REAL; 1 LINT/LREAL; 1 TIME; 1 STRING; -CRC = 2cfd5f84 >
< 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. In doing so, each zenon Logic Runtime and each bus driver receives their own connection to the diagnosis server as a diagnosis client. This does not include obsolete drivers, for example those from a third-party manufacturer.

5.2 Licensing

INSTALLATION WITH ZENON (ZENON RUNTIME AND ZENON LOGIC RUNTIME ON ONE PC)

The zenon Logic Runtime is automatically licenced with the zenon serial number. During the installation enter the serial number and the activation number in the according dialog. If no licence number is available, click on **Demo Version** (on page 42).

AUTARCHIC INSTALLATION (ZENON LOGIC RUNTIME WITHOUT ZENON RUNTIME ON ONE PC)

For an autarkic installation (zenon Logic runs on a PC without the zenon Runtime) you can again enter the serial number and the activation number directly in the installation dialog.

LATER LICENSING WITH LICENCE ORDER

You also can do the licensing later with the Licence order. Open the license order in the Windows start menu under Programs -> COPA-DATA -> Licensing. For later licensing the serial number and the activation number are entered in this dialog. The procedure is finished with OK.

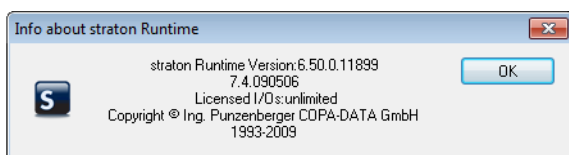
For soft-licensing the so-called licence number must be read from the hardware configuration of your PC. Click the button for soft licensing and follow the requests in the following dialogs.

AFTERWARDS LICENSING USING THE ZENON EDITOR

In the zenon Editor you can open the dialog for licensing using File -> Standard configuration -> License product.... You can enter the serial number and the activation key here.

INFORMATION ABOUT THE LICENSE OF THE ZENON LOGIC RUNTIME

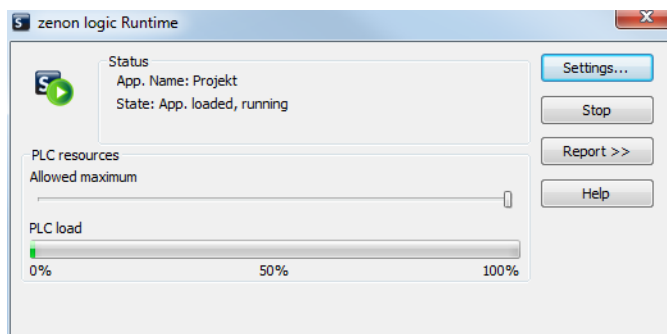
Open the user interface of the zenon Logic Runtime by clicking on the zenon Logic Runtime icon in the system tray. Then click on the top left corner and select **About STARTONRT...** A window appears with information about version and license of the zenon Logic Runtime.



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 used I/Os are displayed in the UI of the <CA_PRODUCTNAME> Runtime.

5.2.2 Demo mode

zenon Logic Runtime starts in Demo Mode if you enter a false (or no) serial number or a false (or no) activation key or if no dongle is connected. The Demo Mode stops the zenon Logic Runtime after 15 minutes. With a special demo serial number the zenon Logic Runtime runs continuously for 30 days. After these 30 days then the maximal time of the zenon Logic Runtime is again 15 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. B&B profile zenon Logic).

Calculation example:

- ▶ four variables for the communication zenon Logic <=> zenon
- ▶ four variables for the communication zenon Logic <=> Profibus
- ▶ This equals the maximum number of eight TAGs ($4 + 4 = 8$).

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 three possibilities in order to start the zenon Logic Runtime:

1. with the help of the zenon Runtime
2. with the help of the zenon Logic Runtime Manager
3. in a DOS Shell/BAT-file

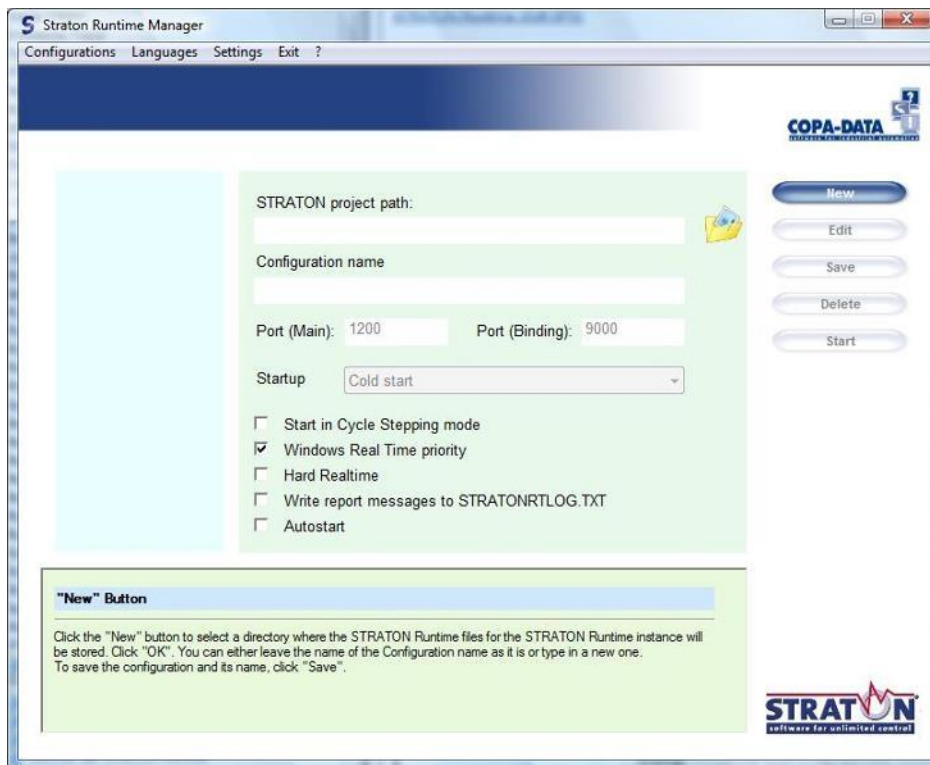
START WITH THE HELP OF THE ZENON RUNTIME

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.

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 will find the zenon Logic Runtime Manager in the folder COPA-DATA 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 ? in the menu bar.



Attention

Runtime Manager must be started with administrator rights.

START IN A DOS SHELL

The zenon Logic Runtime can be started in a DOS Shell or something similar (.bat file). For the command the following is true:

- all commands must be entered in capitals

- ▶ the following signs are allowed:

-

/

no sign

The following handover parameters are available:

MAIN PARAMETER:

| Parameters | Description |
|-----------------------|--|
| PORT=<number> | Communication port. Without parameters: 1200 |
| PORTTRACE=<number> | Event server point, Binding port. Without parameters: PORT+7800 |
| PATH=<DB-Path> | Storage area of the zenon Logic application code. Without parameters: zenon Logic Runtime directory |
| ZENPATH=<Pfad> | zenon path (without meaning) |
| ZENPROJECT=<name> | zenon project name (without meaning) |
| DONTHIDE | Do not hide the Runtime after the start (it is the only parameter without '=') |
| STEP=<0 1> | Without parameters: no stepping mode. 0: no stepping mode 1: Start in stepping mode |
| START=<0 1 2 3> | Without parameters: Warm start 0: Display setup dialog 1: Cold start 2: Warm start 3: Hot start |
| LOWPRIORITY=<0 1> | Without parameters: low priority 0: low priority 1: Windows real time priority |
| WRITELOG=<0 1> | Without parameters: Create log 0: Do not create log 1: Create log The log-file is created in the PATH folder as STRATONRTLOG.TXT. |

ADDITIONAL PARAMETERS:

| Parameters | Description |
|--------------------------|--|
| REMPATH=<Path> | Path for remanent data. Without parameters: PATH directory |
| DELAY=<number> | Start delay in [s]: Delayed start of the zenon Logic Runtime. Without parameters: 0 |
| MAXUTILPRCT=<number > | Position of the slider for maximum PLC workload in %. Without parameters: 100 |

EVENT SERVER:

Settings equal to the settings under **Settings/Advanced** for the event server.

| Parameters | Description |
|-------------------|--|
| QSSYSTEM=<number> | Size of system queue. Without parameters: 1024 |
| QSALARM=<number> | Size of alarm queue. Without parameters: 0 |
| QSEVENT=<number> | Size of event queue. Without parameters: 8192 |
| QSINFO=<number> | Size of info queue. Without parameters: 32768 |
| NBEVENT=<number> | Maximum number of recordable events. Without parameters: 1024 |
| EVENTCON=<number> | Number of possible event connections. Without parameters: 256 |

REDUNDANCY SETTINGS:

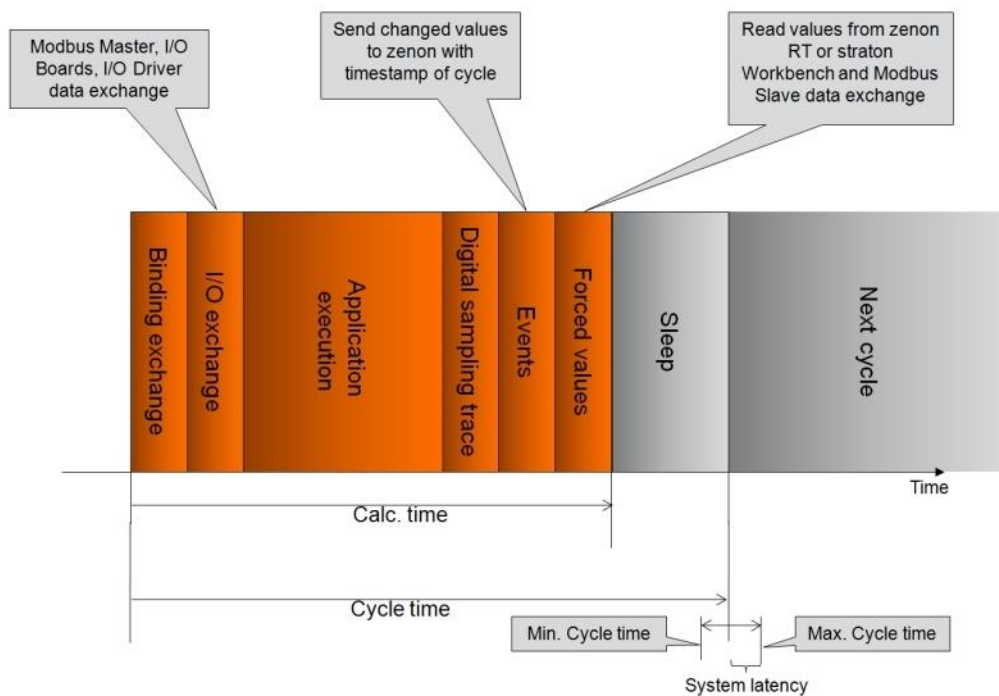
| Parameters | Description |
|------------------------|--|
| REDENABLE=<0,1> | Redundancy on/off. Without parameters: 0 |
| REDADDR=<IP> | Redundancy partner address. |
| REDPORT=<number> | Redundancy port for replication. |
| REDCLITIMEOUT=<number> | Redundancy client time out [ms]; time out which is used to determine when a passive client becomes an active server. Without parameters: 1000 |
| REDSRVTIMEOUT=<number> | Redundancy server time out [ms]; time out which is used to determine when the replication is stopped. Without parameters: 1000 |
| REDZENON=<0,1> | If set to 1, the partner address is read from <code>PROJECT.INI</code> . Without parameters: 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.

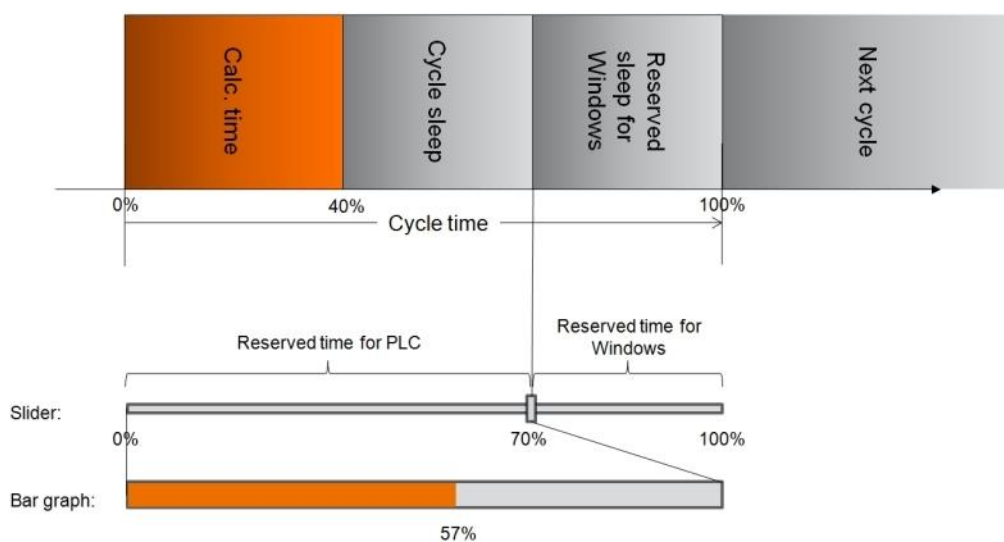
These values can be displayed in the zenon Logic user interface. Use the shortcut **Alt+S** or **Shift+Alt+A**.



| 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 execution | 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 graph always refers to the ratio set in the slide control.



5.4.1 Error messages cycle time

When 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 fix exceeding the cycle time by:

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



- ▶ With a click on **End**, you open a dialog

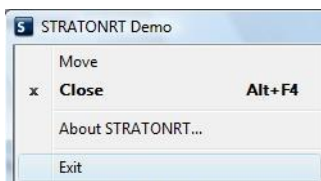


- ▶ Clicking on **Yes** stops and closes the Runtime.

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:



- With a click on **End**, you open a dialog
- 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 hot restart file via function WriteHot.

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:

| Parameters | Description |
|------------|---|
| I | Trigger; if this parameter is TRUE; the function is executed. |
| SYNC | If this parameter is TRUE, the cycle is hold for as long as the hot restart file is written. If the parameter is FALSE, the function is carried out asynchronous. |
| BUSY | This parameter is TRUE 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 tool bar

5.7.2 zenon Logic Runtime autarkic (PC)

INSTALLATION ON THE TARGET SYSTEM:

Install the zenon Logic on the target system with the autarkic installation.

START:

Before downloading the zenon Logic project, start it. The best way is to use the zenon Logic Runtime Manager. With this tool you can also carry out the settings so that the zenon Logic Runtime starts together with the operating system.

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 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 tool bar

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. With that highest determinism and failure safety towards the operating system are guaranteed.

Requirements:

- ▶ The zenon Logic RTK is only available for operating systems Windows 7 in the 32 bit version.
- ▶ The zenon Logic RTK can only be started with the zenon Logic Runtime Manager (flag `Hard realtime` has to be activated) or or in a DOS-Shell/BAT-file.



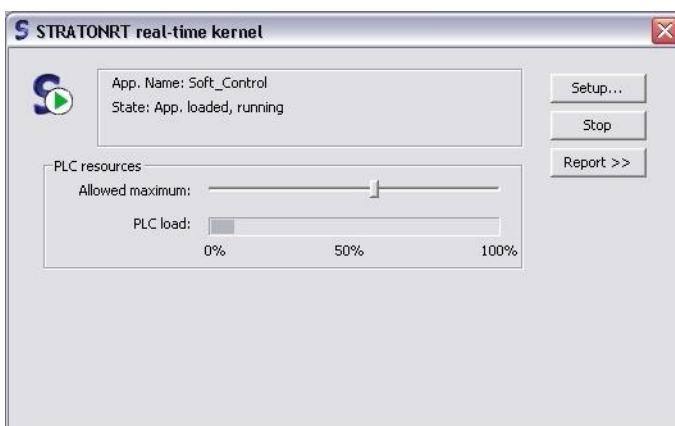
Attention

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, so that it triggers the zenon Logic Runtime cycle independent of Windows. So the zenon Logic Runtime cycle is guaranteed to be executed deterministically. Cycle times must be 100 µs or higher.

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



Depending on the size of the PLC application and the defined cycle time the slider has to be adjusted to the PLC load. The slider can be set between 10% and 85%. The current load (bar) indicates whether the allowed maximum for executing the PLC cycle is sufficient.

If the bar is near 100%, there is a danger of cycle overflows! In this case set the slider higher (towards 85%). If this is not possible because of the applications in Windows (e.g. bad 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

The following formula can be used to find out, if the setting of the slider does not lead to cycle overflows in combination with the cycle time:

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

*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

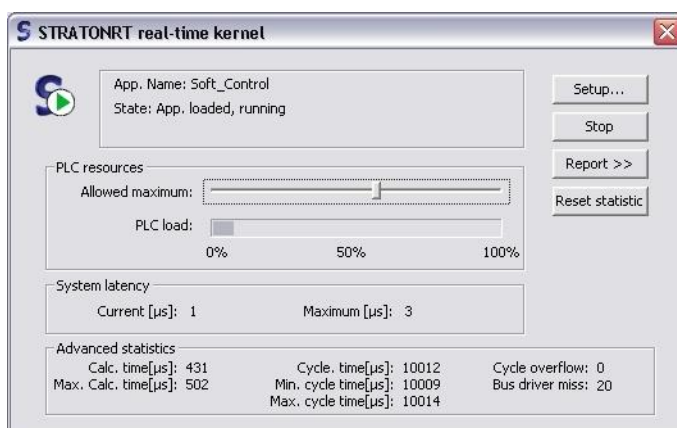
Slider position: 50%

Set cycle time: 1ms (1000μs)

Calculation:

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

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



**Attention**

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

DESCRIPTION OF THE DISPLAYED MEASURING VALUES (SHOW WITH ALT+S OR SHIFT+ALT+A):

| System latency (Alt+S) | |
|-------------------------------|--|
| Current [µs] | Current system latency (see also Runtime cycle (on page 48)) |
| 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 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 Timer Periode (time basis). The time basis depends on the selected timer and its time basis (see also zenon Logic RTK configuration (on page 59)). 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 (Windows 7) 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.

- ▶ 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

Note: The zenon Logic real time components are only available for Windows 7 and Windows 8 under 32-bit.

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

*In order to determine exactly which function blocks the zenon Logic standard library supports, use the feature of the configuration of the zenon Logic Workbench (main menu: *Project/Configuration/Upload*).*

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 79)

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>: `C:\Program Files\COPA-DATA\zenon720`

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.



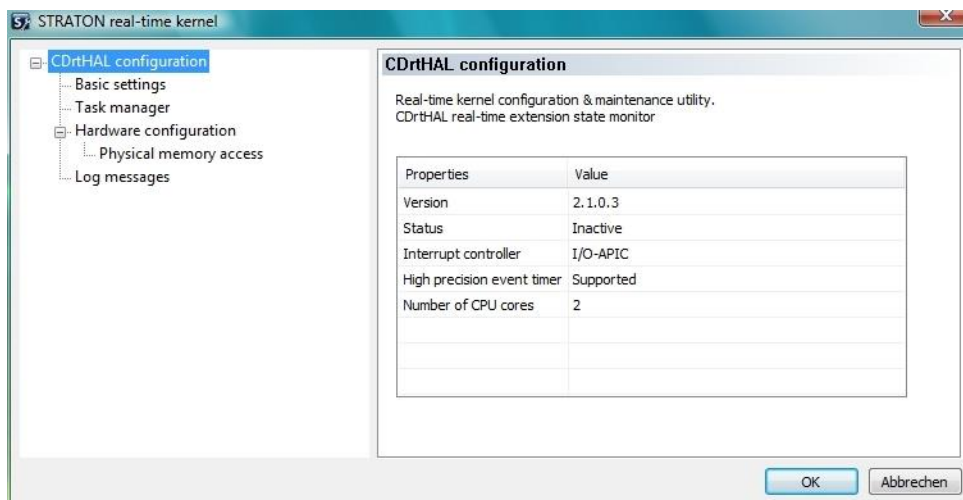
The window for editing the configuration opens. In the left window you see a menu tree with the following options:

- ▶ CDrtHAL configuration
- ▶ Basic 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:



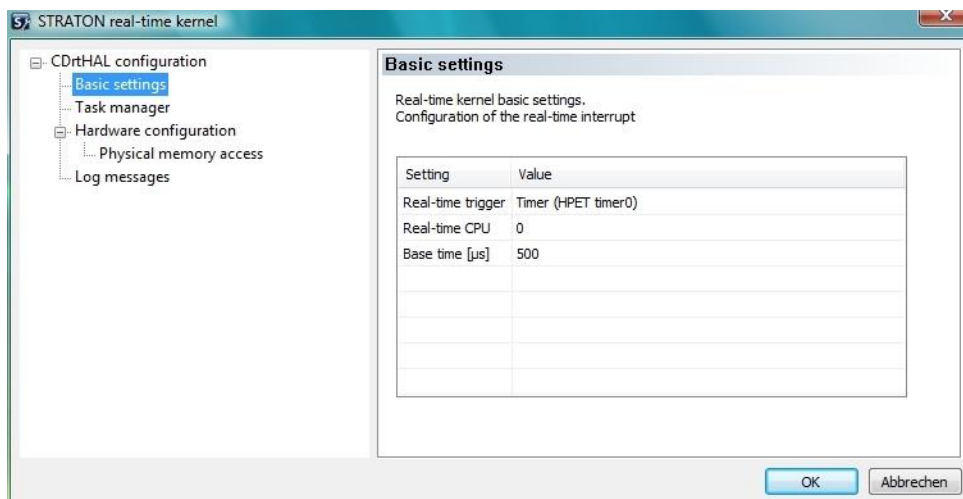
| Property | Information |
|-----------------------------------|--|
| Version | Version of the real time driver (CDrtHAL). |
| Status | RTK is active or inactive |
| Interrupt controller | Type of interrupt controller. |
| High precision event timer | Support depends on hardware: |
| Process cores |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

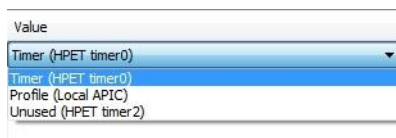
- ▶ Interrupt-Controller APIC
- ▶ Interrupt-Controller PIC
- ▶ HPET (High precision event timer)

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.



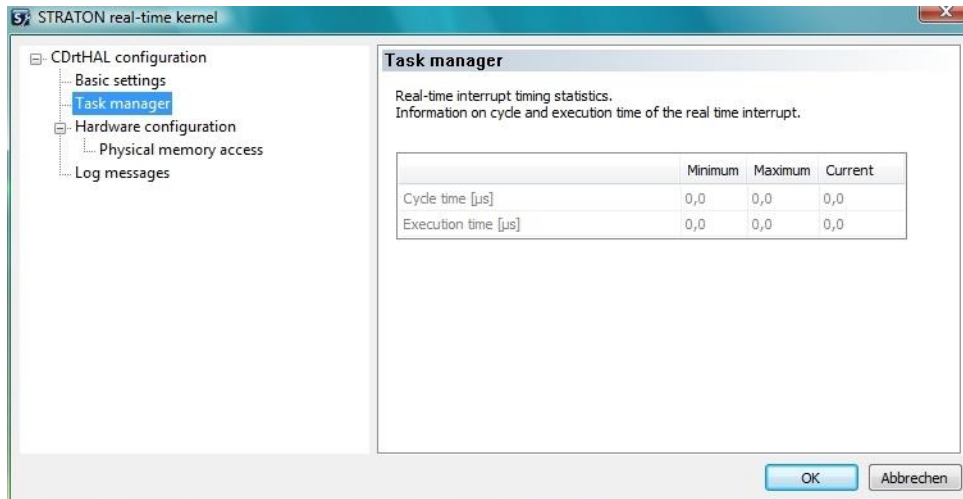
Attention

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.

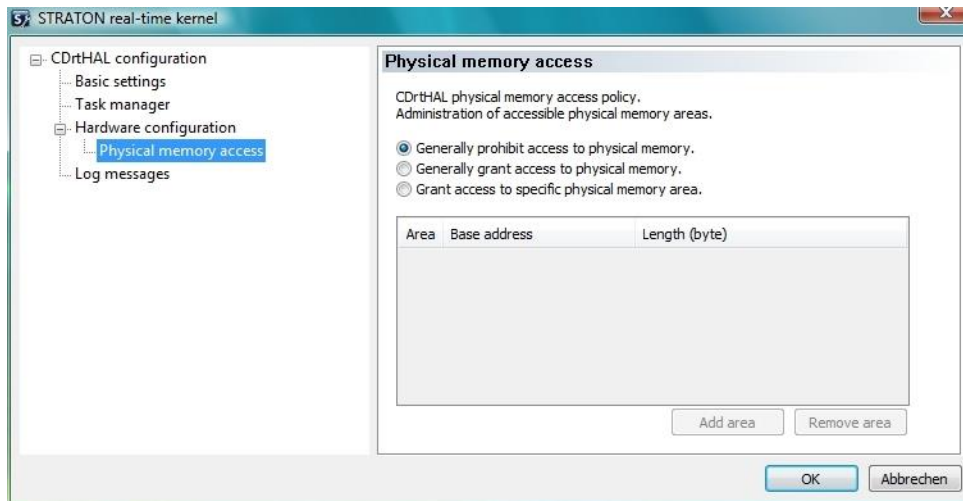


| Property | Information |
|---|--|
| Cycle time (in microseconds) | Minimal, maximal and current value. Cycle refers to the cycles of the Timerticks based on the basic time. |
| Execution time (in microseconds) | Minimal, maximal and current value. |

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.

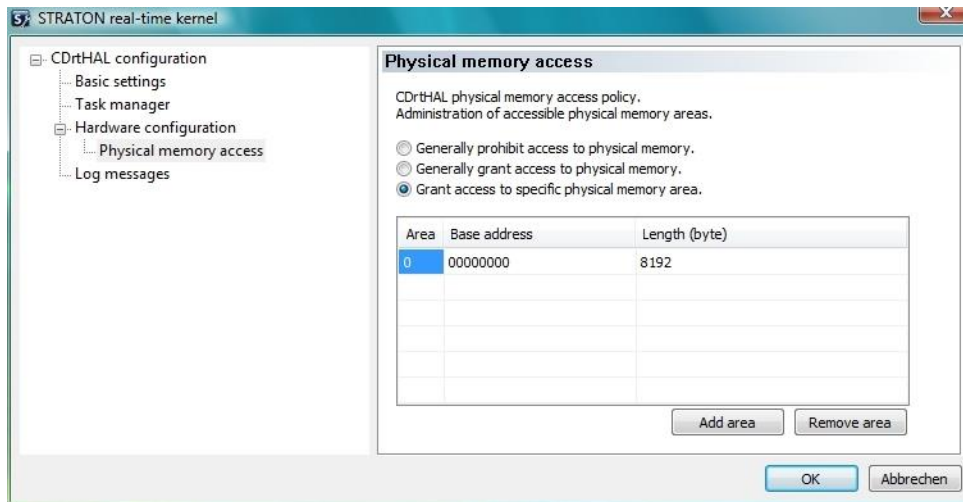
The configuration offers three settings:



| Options | 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 prohibited in the shipping state. Basic setting. |
| Generally allow access to the physical memory. | Allows the communication to use the whole memory, 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

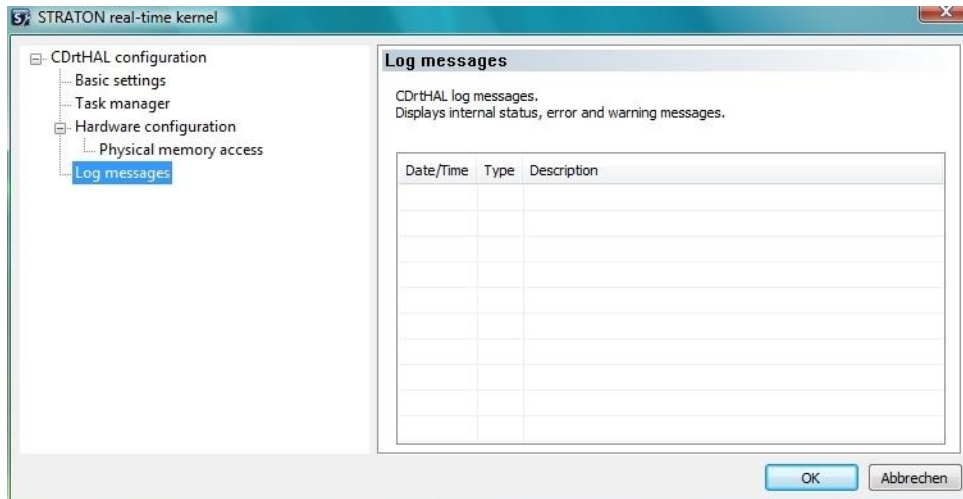
In order to allocate a fixed memory area for zenon Logic and I/O driver, click on **Access to the physical memory** in the left window and then on **Allow access to defined physical memory area** in the right selection window.



| 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

Messages from the RTK are displayed in the area **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 reboot, 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 probable 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 theoretically next cycle.

If the zenon Logic RTK is configured to continue running, the next cycle is executed after a short delay of ca. 10 ms (switching the graphics mode).

In order to achieve this, the zenon Logic function **RTK_OnBugCheck** of the zenon Logic standard library must be set to TRUE in the zenon Logic application.



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, no real time field bus driver, system clock functions, RETAIN variables, writing log messages (changes reserved by product version cycles).

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

SYSTEM REQUIREMENTS FOR SMOOTH CONTINUOUS RUNNING AFTER A BSOD:

- ▶ Windows® 7 (32 Bit)

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 you restart STRATONRTK. |
| 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 (old) 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

```

#define T5RET_OK 0
#define T5RET_ERROR 1
#define T5RET_BADCODE 2 /* invalid app code - bad header */
#define T5RET_BADCODETARGET 3 /* invalid code - bad target id */
#define T5RET_BADCODEVERSION 4 /* invalid code - bad version */
#define T5RET_EXECSIZE 5 /* exec buffer overflow */
#define T5RET_BADREQUEST 6 /* unknown comm request */
#define T5RET_SERVERREAD 7 /* cannot process read request */
#define T5RET_UNKNOWNTIC 8 /* unknown tic code */
#define T5RET_DATA8OVERFLOW 9 /* too many DATA8 used */
#define T5RET_DATA16OVERFLOW 10 /* too many DATA16 used */
#define T5RET_DATA32OVERFLOW 11 /* too many DATA32 used */
#define T5RET_DIVZERO 12 /* division by 0 in TIC (32bit) */
#define T5RET_TIMEOVERFLOW 13 /* too many TIME variables */
#define T5RET_CANTACTIME 14 /* active timers not supported */
#define T5RET_SAFETICLOOP 15 /* awful loop in TIC (safe mode) */
#define T5RET_OUTOFTIC 16 /* TIC code ointer out of range */
#define T5RET_UNKNOWNSTF 17 /* not supported std function */
#define T5RET_LDIVZERO 18 /* division by 0 in TIC (64bit) */
#define T5RET_DATA64OVERFLOW 19 /* too many DATA64 used */
#define T5RET_STRINGOVERFLOW 20 /* too many strings */
#define T5RET_SDIVZERO 21 /* division by zero in TIC (8 bit) */
#define T5RET_WDIVZERO 22 /* division by zero in TIC (16 bit) */
#define T5RET_HOTRESOURCES 23 /* HOT restart: system resources used */
#define T5RET_HOTMAP 24 /* invalid var map in hot restart */
#define T5RET_HOTACTIME 25 /* change of active timer map */
#define T5RET_HOTSTAMP 26 /* bad build stamp for hot restart */
#define T5RET_HOTSFC 27 /* SFC changed - no hot restart! */
#define T5RET_HOTRETAIN 28 /* retain def has changed - no hot */
#define T5RET_HOTFORMAT 29 /* bad DB format - no hot restart! */
#define T5RET_CYCLEOVERFLOW 30 /* cycle time overflow */
#define T5RET_RETAINSTORING 31 /* cant store retain variables */
#define T5RET_ARRAYBOUND 32 /* array index out of bounds */
#define T5RET_RECURSE 33 /* recursive call of sub-program */
#define T5RET_XVOVERFLOW 34 /* too many external variables used */
#define T5RET_NOBSAMPLING 35 /* sampling trace not supported */
#define T5RET_TASKOVERFLOW 36 /* too many tasks */
#define T5RET_HOTCALLSTACK 37 /* call stack overflow (hot) */

```

```
#define T5RET_STACKOVER      38  /* stack overflow */
#define T5RET_HOTASI         39  /* cannot hot restart ASi */
#define T5RET_TOXC           40  /* OXC limit reached */
#define 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

Redundancy can be used with the following operating systems:

- ▶ Windows 7
- ▶ Windows 8
 - Note: Windows 8 RT is not supported for reasons relating to the system.
- ▶ Windows CE

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 demands that both systems, zenon and zenon Logic, are installed on one PC.

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 implies that zenon Logic and zenon communicate with each other using the shared memory driver (zenon Logic to zenonRT connection), it must be guaranteed 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 (stand alone/Server/Standby Server/client) is transferred via the shared memory command area 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.

To make it clear from which zenon project the watchdog or the status originates,

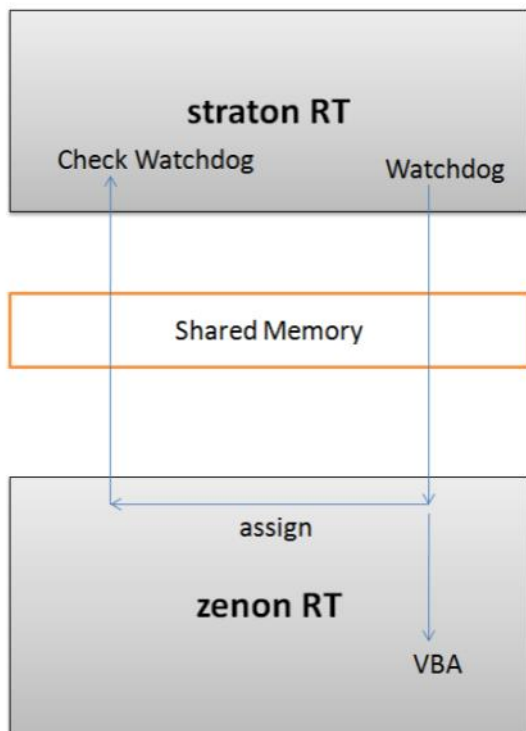
- ▶ the zenon Runtime driver or the shared memory area must have the zenon project GUID
- ▶ the zenon Runtime driver or the shared memory area must have the zenon Logic project name

ZENON

A watchdog generated by the zenon Runtime driver is transferred via the shared memory command area 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. The following must be the case for this:

- ▶ the zenon Runtime driver or the shared memory area must have the zenon Logic project name



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 <CAD_PRODUCTNAME>

The settings for redundancy are made in tab Redundancy (on page 37) of the straton 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 43) 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 <> 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 time-out 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-project> sprj:<s-project> Runtime <Runtime Process ID> watchdog expired.
Runtime watchdog was not operated by Runtime for too long.
 - zprj:<z-project> sprj: <s-project> Runtime: <Runtime process ID> Unknown straton project.
zenon Logic Project cannot be assigned to an integrated zenon project.
- ▶ Warnings
 - vm: <Vm process ID> zenon project '<Name>' unknown
Name of the zenon Logic projects is 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 >CD_PRODUCTNAME< 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 you can check with the help of function blocks whether a connection to a zenon Server exists and you can 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 count up in the time-out cycle. 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 now no longer `MAX_SWITCH * cycle` but `MAX_SWITCH * timeout [+ network timeout]`.

The non-process-handling, non-dominant zenon computer is considered a Standby Server.
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.

In the zenon Logic Workbench a list of this configurations is available under Project -> Configuration.... This can change with the product/version cycle.

FILE ADMINISTRATION FUNCTION (FUNCTION BLOCKS)

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

The following functions are supported by zenon Logic Runtime versions

- ▶ CE
- ▶ RT (PC)
- ▶ RTK (PC)

and are implemented not blocking:

F_OPEN, F_WOPEN, F_AOPEN, F_CLOSE, F_EOF, FA_READ, FA_WRITE, FM_READ, FM_WRITE, FB_READ, FB_WRITE, F_EXIST, F_GETSIZE, F_COPY, F_DELETE, F_RENAME, LogFileCSV

The following functions are only supported by zenon Logic Runtime versions

- ▶ CE
- ▶ RT (PC)

and are implemented **not blocking**:

F_EXIST, F_GETSIZE, F_COPY, F_DELETE, F_RENAME

The possible paths vary depending on the operation system. Note that some folders are write-protected if UAC (User Account Control) is activated.

SFC EXECUTION

The execution of the SFC program equals that of a T5 standard target system and 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), e.g. the ThinkIO/System758 driver.

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 33)/general (on page 34).

I/O DRIVER

You can find more information about supported I/O drivers here: zenon Logic I/O driver support (on page 79)

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 81)

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 Diagnosis Viewer manual, in the Diagnose Client ##12470 section

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

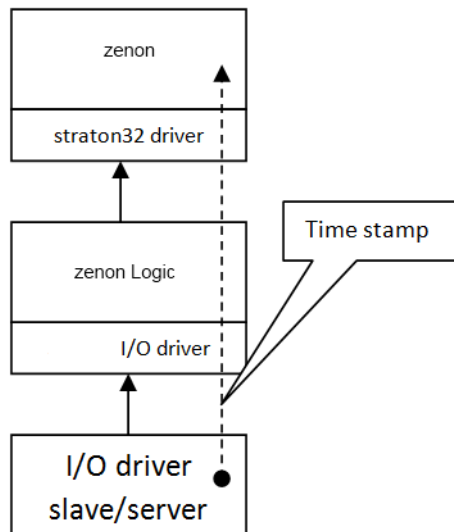
5.11 Forwarding time stamps and variable status information

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:



Note: The names **zenon** and **zenon Logic** in the graphic are just examples. They can be different depending on the configuration and correspond to zenon and zenon Logic.

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 Configuration | x | E | - |
| ApplicomIO Configuration | x | E | - |
| AS-interface | x | - | - |
| BECKHOFF CX1000 | - | - | x |
| Brodersen IOTOLS | x | E | - |
| CAN-bus | - | - | - |
| Hilscher CIF Profibus | x | x ¹ | x |
| Hilscher SYCON Configuration | x | E | x ² |
| Hilscher SYCON Configuration [universal] | x | x | x |
| Hilscher SYCON.net Configuration | 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 Protocol | x | E | x |
| MODBUS Slave Protocol | x | E | x |
| PROFINET IO | x | E | x |
| SoftNet ProfibusDP | x | E | - |
| straton to zenon Runtime connection | x | - | x |
| ThinkIO/IO System 758 | - | - | x |
| Wago 750-860 | - | - | - |
| Weihenstephan Standard - Client | x | E | x |
| Win32 Shared memory | x | - | - |
| XFlow | - | - | - |

LEGEND

x: Fully supported

-: Not supported

E: Carried out as emulation (performance loss possible, determinism is limited in the PLC cycle, driver selection at a BSOD)

- 1: The driver specific function blocks do not work under zenon Logic RTK.
- 2: Hilscher DLLs must be available on the target platform.

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 Configuration | x | x | x ¹ |
| ApplicomIO Configuration | x | x | x ¹ |
| AS-interface | x | x | x ¹ |
| BECKHOFF CX1000 | x | x | - |
| Brodersen IOTOLS | x | x | x ¹ |
| Hilscher CIF Profibus | x | x | - |
| Hilscher SYCON Configuration | x | x | x ¹ |
| Hilscher SYCON Configuration [universal] | x | x | x ¹ |
| Hilscher SYCON.net Configuration | x | x | x ¹ |
| IEC 61850 Client | x | x | - |
| Interbus-S | x | x | - |
| MODBUS Master Protocol | x | x | x ¹ |
| MODBUS Slave Protocol | x | x | x ¹ |
| PROFINET IO | x | x | - |
| SoftNet ProfibusDP | x | x | x ¹ |
| straton to zenon Runtime connection | 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 messages

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 <code>straton.cod</code> . |
| 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 <code>straton.cod</code> 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

When 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 fix exceeding the cycle time by:

- ▶ increasing cycle time
- ▶ decreasing execution time, e.g.:
 - use a system with more CPU resources
 - use C compiled code
 - 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 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 changes 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 drivers. 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 - The zenon Logic Runtime starts with initialized variables.
- ▶ Cold start (load RETAIN variables) - The zenon Logic Runtime starts initialized with remanent markers.
- ▶ Hot restart - The zenon Logic Runtime starts with the variable values from the time the 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 Button "Delete"

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

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 <CA_PRODUCTNAME >Runtime 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.