COPADATA
do it your way

zenon Tutorial

First steps

v.7.20

Contents

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. First steps

Welcome and thank you for choosing zenon. We want to make the introduction to the use of zenon as simple and pleasant as possible. The first step in this direction is made with this tutorial. Nevertheless, an introducing tutorial cannot replace proper training.

This tutorial will introduce you to the basic operation of zenon. Here, you will learn how to create a project and how to draw screens that display the values from your PLC in dynamic elements.

3. The task

The following task must be carried out for this tutorial:

▶ Water is pumped from a stream with the help of a motor.

▶ This water is stored in two storage tanks of 20,000 liters each.

▶ The temperature of the water is monitored with temperature sensors.

▶ An action is executed in the event that there is a risk of freezing.

▶ An action is executed in the event that a defined fill level is exceeded.

▶ An action is executed in the event that a defined fill level is gone below.

THE FOLLOWING WILL BE CARRIED OUT IN THIS TUTORIAL:

▶ Two tanks and their inflow and outflow pipes will be visualized

▶ Limit values and alarms for variables will be defined

▶ In zenon, available elements are presented, discussed and configured individually

▶ Alarms and system messages are displayed in clear lists

▶ Implemented in the Runtime menu and button control

▶ Several steps combined in scripts

▶ Environment-dependent actions (such as tank full, temperature too low) presented, discussed and configured.

YOU WILL LEARN HOW YOU ...

▶ Get to know the Editor as a graphic user interface to create and maintain projects. In doing so, you will find out how to design your working environment in such a way that it is possible to work efficiently.

▶ You will be able to create workspaces in which you administer your projects. You will also get to know different types of projects and the how to administer properties such as color palettes, fonts, etc. centrally.

▶ You will be able to establish communication to your hardware or other data sources with the help of drivers. Furthermore, you will find out how to create variables as the central interface to driver objects and data types, and how you can create different data types yourself.

▶ You will create templates as the basis for your screens and get to know the advantages of template systems.

▶ You will be able to design screens with different elements and to display the variable values in these. You will group symbols and reuse these effectively.

▶ You will be able to implement the operation of your project with the help of predefined functions.

4. Working with zenon

In the course of this tutorial you will get a feeling for working with zenon. You will notice that different challenges will require a similar approach.

Highly simplified, working with zenon can be described as follows:

▶ You create an object.
It can be a driver, a variable, a screen, a screen element, a schedule, a user ...

▶ In the next step, you make your settings for this object.
You parameterize your object instead of programming.

▶ With the help of functions working steps are defined and actions initiated.
These are pre-defined macros that are easy to use and parameterize.

▶ By applying and combining these elements complex challenges can be carried out with only few mouse clicks and without time-consuming programming.

> 💡 **Information**
>
> An advantage of this procedure is that it is possible to re-use already configured steps.

5. A new project

In this step, you will learn what workspaces and projects are.

A neutral working area with a project contained therein is created.

TAGs	Description
Workspace	Is an administrative unit in the zenon Editor, in which projects are grouped. Only projects that are inserted into a workspace can be edited in the Editor.
Project	Summary of settings, screens, functions, variables, recipes etc. for the display of a visualization task. The projects are created in the Editor. The Editor data is saved in part in an SQL database and as binary data. The data can be stored either locally on the Editor computer or on a central computer (multi-project compatible). A backup of projects can be made at any time and this can be read back on the same computer or a different computer.

5.1 Creating a new workspace

At the start of the work with zenon, there is the workspace. Projects are created and administered there. As requirements increase, several projects can be compiled into one workspace and configured.

STEPS TO CREATE A NEW WORKSPACE

▸ Select the entry `Workspace new` in the menu *File*.

▸ Enter the folder `C:\` and enter `Basic_Tutorial` as the name of the workspace.

▸ Note: If it does not exist, the directory `C:\Basic Tutorial` is created and the file `Basic Tutorial.wsp6` is created in it once the dialog box has been confirmed with `OK`.

🔆 Information

The directory name and the name of the workspace file are issued automatically by zenon, but they can be changed later.

5.1.1　Tips for the workspace

You should organize your workspace according to your working environment. A well structured workspace increases clarity and efficiency

The following list serves as a guideline

▶　For integrators:
One workspace for each client

▶　For engineers:
One workspace for each production location

Note: Select a public folder as storage location of your workspace. The authorization of a workspace is thus not bound to the creator.

5.2　Creating a new project

To create a new project:

▶　In the `File` menu, select the `New project` command.
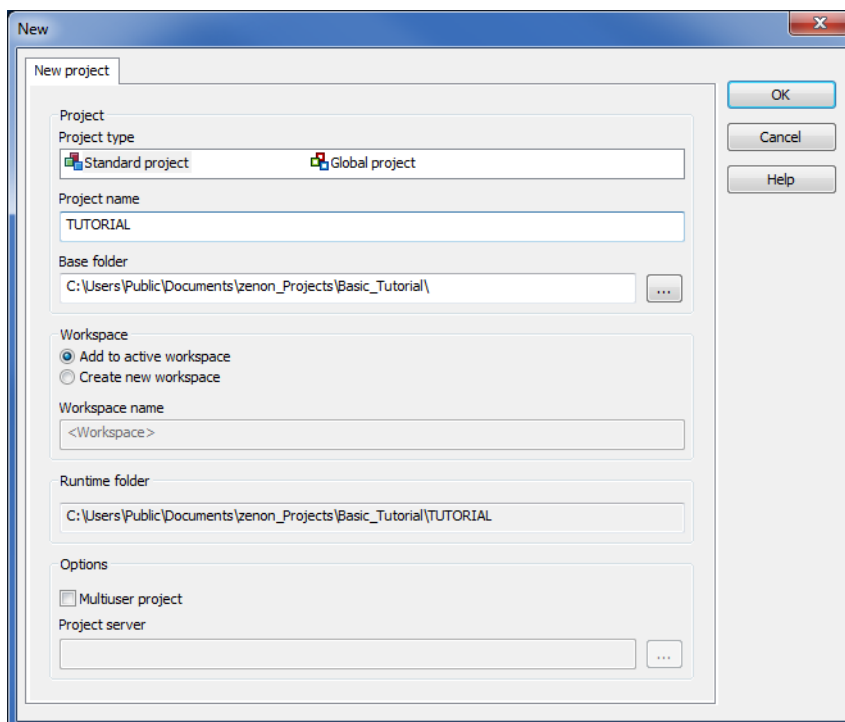
▶　Enter *'TUTORIAL'* as the project name and accept the proposed path.

> 💡 **Information**
>
> *If VBA is activated in your Editor (default setting), the selection dialog for executing a wizard will open now. Wizards are VBA macros, with which you can automate the work in the Editor. No wizard is used in this example. Close the dialog with* `Close` *and confirm the subsequent dialog "Would you like to save your settings?" with* `No`*.*

5.2.1 Distinction standard project and global project.

This tutorial works with a standard project. In addition, zenon offers the following project types:

- ▶ A **Global project** only has limited functionality. The objects created in a global project (templates, fonts, colors, etc.) are also available in every other project of the workspace. For instance, the templates of the global project can be selected under the name 'g_Name' in the other projects. In contrast, other settings are only configurable in global projects. For instance, modelling a plant in the module plant modelling is only possible in global projects.

- ▶ The option **Multi-user project** allows to create a project which can be edited by various engineers at the same time, with the control system monitoring that an object cannot be edited by two engineers at the same time.

Note: Both options are not used in this tutorial project.

5.2.2 Possibilities for application

You have different options depending on the task. In the process, note the multiple uses and thus save yourself additional work.

GLOBAL PROJECT

The global project is available to configure settings that can be reused in the projects. Settings of a global project are available to all projects in a workspace.

Examples:

- ▶ Design of font lists
- ▶ Definition of colors in the color palettes
- ▶ Inclusion of general language files - for example zenon system texts

> 💡 **Information**
>
> Steps that have already been configured can be reused in zenon.

MULTI-USER PROJECT

A multi-user project is appropriate if several people with different core tasks are working on one project.

Example of project distribution:

- ▶ The screen designer is responsible for the appearance and usability. To do this, they create font lists, screens, buttons, graphics and color palettes, which are not dependent on the technical configuration.

- ▶ Project engineer creates driver and variables

- ▶ Technical editor writes documentation and text documents for the project

- ▶ The special solution developer takes care of the special configuration of he project, for example the configuration of a reaction matrix

- ▶ The controlling employee configures the SAP interface for their cost invoicing

- ▶ ...

> 💡 **Information**
>
> *Multi-user project allow location-independent project cooperation.*

5.2.3   Creating a project with the wizard

In this tutorial, all necessary configurations is shown step-by-step. The wizard is intentionally not used.

The wizard can be started afterwards. This is also possible if you have decided not to use automatic call-up after the creation of a project In this case, your previous project configuration may be overwritten.

START WIZARDS

To start a wizard:

- ▶ Select, in the `File` drop-down list, `Wizards ...`

    or

▶ press the short cut `Alt+F12`
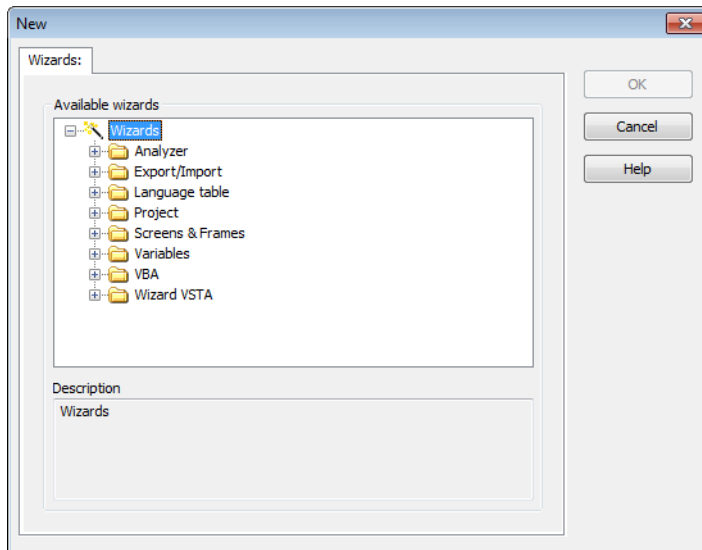
The wizard for project creation is automatically offered when a new project is created.

> 💡 **Information**
>
> *VBA must be activated in your Editor in order to be able to use the project wizard (standard setting).*

VBA ACTIVATION IN THE EDITOR

For wizards to be displayed, the settings for VBA and/or VSTA must be set correctly in file `zenon6.ini`*:*

`[VBA]`

`EIN=1`

`[VSTA]`

`ON=1`

If VSTA wizards are not displayed although the settings are correct, set entry `LOADED=` to `1` in area `[VSTA]`.

5.3 Engineering of the project

The properties of the `TUTORIAL` project are displayed in the properties window and can be changed here.

▶ Select the workspace in the project manager.

▶ In the detail view you now see a list of the projects belonging to this workspace.

▶ Select the TUTORIAL project in the detail view.

The properties window is still used often in this tutorial. Generally, it shows the properties of the object, which is selected in the detail view of the project manager. The icons on the top border of the properties window allow you to show the properties in different views:

💡 **Information**

*With the context menu of the properties window you can define which properties should be displayed in the favorites. Favorites for the properties window are only available in the **grouped view**.*

5.3.1 Functionality of the properties window

The individual symbols of the properties window offer the following functionality:

Symbol	Meaning
`Grouped`	The properties are combined into logical groups.
`All properties`	All properties are shown in a row.
`Dialog view`	The properties are displayed as dialog boxes.
`Show/hide favorites`	In the favorites, you can put together the most frequently used properties by using the context menu of the property window. Here, you can show or hide the favorites.
`Show/hide all properties`	If the favorites are shown, you can hide all other properties with this symbol for a better overview.
`Sorted logically`	With this icon, the displayed properties are sorted according to their logical connectedness.
`Sort ascending`	With this icon, the displayed properties are sorted in alphabetically ascending order.
`Sort descending`	With this icon, the displayed properties are sorted in alphabetically descending order.
`Expand all`	By clicking on the '+' on the left border of the property window, you can open a closed node. This icon automatically expands all closed nodes.
`Reduce all`	By clicking on the '-' on the left border of the property window, you can close an expanded node. This icon automatically closes all expanded nodes.
`Display properties help`	A window with the description of the selected property is displayed. To close the help window, click on the **x** on the top right-hand corner of the window.

5.3.2   Favorites in the properties window

For virtually every configuration in zenon, the settings are changed in the properties window. You can group often-used settings concisely by defining favorites.

To do this, carry out the following steps:

▸   In the properties window, change to the `Grouped` view.
Note: If the view is empty, select `Show/hide all properties`.

▸   Add to the various node items and subpoints with the + symbol.
Alternatively, select the `Expand all` button from the properties tool bar.

▸   Add the `Graphic design` node and the **General Runtime** node in it.

▸ Select the **Runtime title** entry that you want to add to your favorites.

▸ In the context menu (right mouse click) select `Add 'Runtime title' to favorites`.
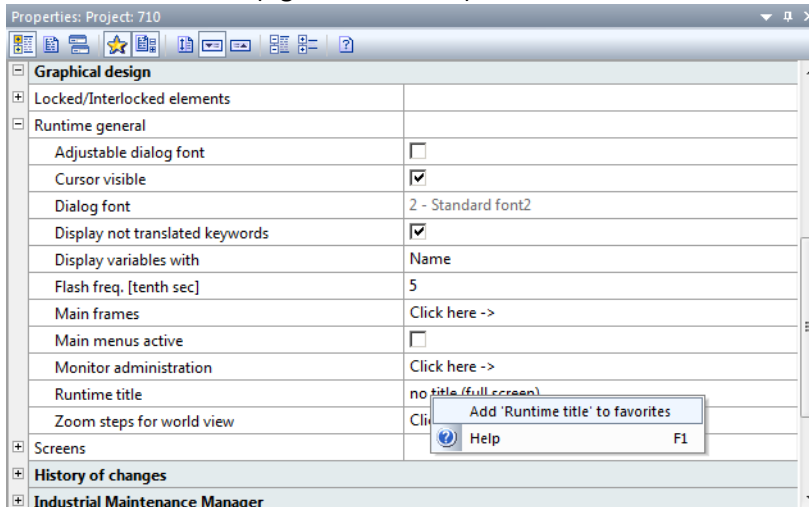
▸ Your favorites are the first entry in the grouped view.

Note: You can show or hide the favorites by clicking on the 'Show/hide favorites' symbol. in your properties display

---

💡 **Information**

You can also set favorites by dragging & dropping. To do this, select the desired property field and drag it to the desired location in the 'Favorites' node.

5.3.3   Runtime view

To display the main window with out a title in the Runtime:

▸ Switch to the grouped view in the zenon Editor.

▶   Open the **Graphical design** group.
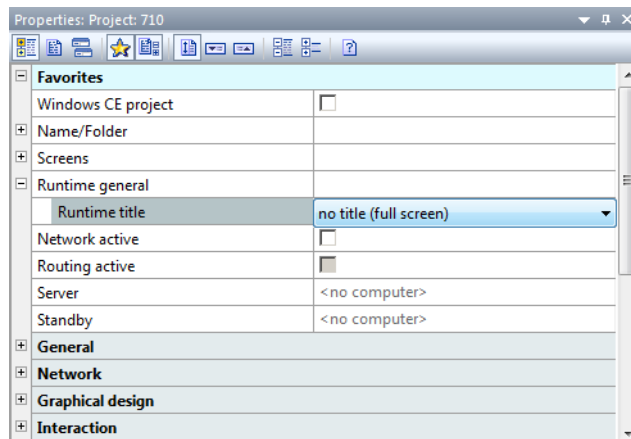
▶   Change the 'Runtime title' property to `No title (full screen)`

Note: This means the program window of the Runtime will be displayed without a title bar.

5.3.4   Project description

Personalize your project configuration:

▶   To do this, select the `General` section in the properties window.

▶   Select the entry 'Project description'.

▶   Select `Click here ->`

A dialog opens:
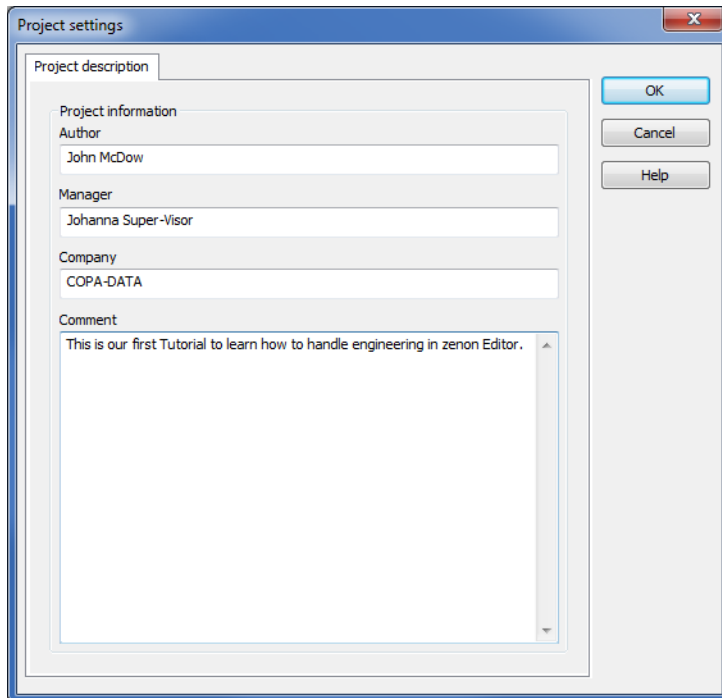
Enter your data into the input fields. Although all inputs are optional, it is recommended that you make detailed (and most of all correct) entries.

5.4 Several projects in one workspace

If you are using several projects, one of the projects is emphasized in bold in the workspace window. This marks the "start project". The start project is used in online mode and simulation mode. It is always the start project that is displayed in Runtime. Only one project can be stipulated as the start project.

To change the start project:

- ▶ Select a project in the workspace window
- ▶ Select the "**Activate project**" command in the context menu (right mouse click).
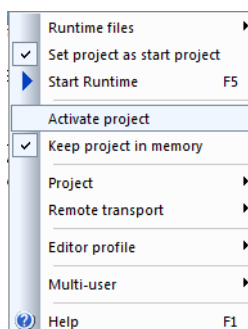
KEEP PROJECT IN MEMORY

If you are using several projects, you see the structure tree of the currently-active project. The deactivated projects do not have a structure tree in their respective view.

In order to be able to not have to access the structure of non-active projects, select `Keep project in memory` in the context menu (`right mouse click`) on a deactivated project.
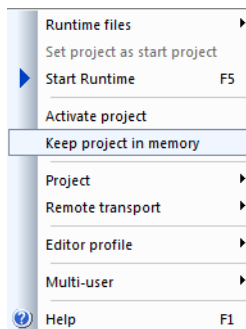
5.5   Practical tips for projects

▶ If you are using several projects in a workspace, take this into account when naming your configuration. Place a code for your project in front of each function, variable etc.

▶ If a new directory is created with each new workspace, all important data for this workspace is available together.

▶ Projects within a workspace can be moved by dragging & dropping

▶ Configurations of a project can be reused in any desired project of a workspace.

5.6   Introductory information on this tutorial

A project will be covered step-by-step in this tutorial.

The following information will be described briefly in advance. This configuration will be referred to during the course of the project:

5.6.1   Color palettes

Color palettes make it possible to compile individual colors into color palettes. You can define color set such as this, that can be easily edited, both in the editor and in Runtime. All colors that are defined with the help of palette colors can be easily changed over with this.

The uniform design of Company Designs (CD) can be completed very quickly in this way. If necessary, the design can be centrally changed in full (change palette) or only individual colors can be changed (change color in palette).

Colors that are configured in color palettes are available in zenon. You will now use the following configuration as the project progresses through this tutorial.

This is how you configure the color palette

▶   In the project manager, open the `Screens` node.

▶   Open Activate the context menu of the `Color palettes` entry with the right mouse button.

▶   Select the `New color palette` entry.
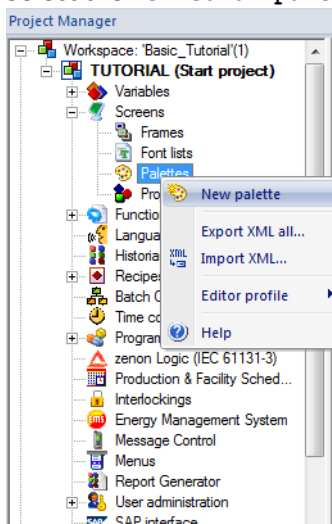
▶   Select `New color` in the detail view of the project manager.

•   For the **description**, enter *Freezing*.

•   Select *#0000FF* as the **color**

▶   Define four other colors in accordance with the requirements below:

No.	Identification	Freezing	
0	Freezing		#0000FF
1	Danger of frost		#C0FFFF
2	Heat warning		#FFAD5E
3	Overheating		#FF0000
4	Normal temperature		#3CB371

▶   Configure three other colors as limit value colors for the fill level of your water tank:

5	Fill level high		#A52A00
6	Fill level low		#A52A00
7	Fill level normal		#005500

> **Information**
>
> *If a color in a color palette is changed, this color changes wherever it was used in your configuration.*

If you configure the same color to be used for several fill levels, you should nevertheless configure two colors. They thus remain flexible for subsequent changes.

Other colors in this tutorial

You will use colors for different topics in this tutorial. The exact application of these colors is described in more detail in the respective chapter.

Prepare this already at this stage in accordance with the following:

No.	Identification	Freezing
0	Freezing	#0000FF
1	Danger of frost	#C0FFFF
2	Heat warning	#FFAD5B
3	Overheating	#FF0000
4	Normal temperature	#3CB371
5	Fill level high	#A52A00
6	Fill level low	#A52A00
7	Fill level normal	#005500
8	nav_background_navi	#C0C0C0
9	nav_background_main	#FFFFED
10	nav_button_background_1	#4B0082
11	nav_vutton_background_2	#98FB98
12	nav_button_typo1	#FFFFFF
13	nav_button_typo2	#480048
14	typo_header	#550F21

> **Information**
>
> *The colors of the images are proposals. You can amend these as you wish.*

A second color palette

Once you have configured the colors of your tutorial project with the help of color palettes, you now have an additional option to get to know advance color administration.

You can configure alternative colors for your configured colors. It is possible to select between the two color worlds using a function.

For the configuration of other color palettes:

▶ In the project manager, open the `Screens` node.

▶ Open Activate the context menu of the `Color palettes` entry with the right mouse button.

▶ Select the `New color palette` entry.

No.	Identification	Freezing	
0	Freezing	■ #0000FF	
1	Danger of frost	□ #C0FF	New palette
2	Heat warning	■ #FFAD	Create standard function
3	Overheating	■ #FF00	Delete palette
4	Normal temperature	■ #3CB:	New color...
5	Fill level high	■ #A52A	Delete color
6	Fill level low	■ #A52A	
7	Fill level normal	■ #0055	Copy
8	nav_background_navi	■ #C0C0	Paste
9	nav_background_main	□ #FFFF	
10	nav_button_background_1	■ #4B00	Save
11	nav_vutton_background_2	■ #98FB	Export XML selected...
12	nav_button_typo1	□ #FFFF	Import XML...
13	nav_button_typo2	■ #4800	Properties...
14	typo_header	■ #550F2I	Help F1

▶ Change to the **Color** group in the properties window.

▶ Select, with . . . , a new color, for example: *#C0C0C0*
Note: All colors are automatically highlighted once a color palette has been created. As a result of this, a new color is automatically assigned to the complete new color palette. If this highlighting is not active, you can get this back by clicking on the heading 'Color palette 1'. You can of course assign each color to your own color palette. However in this tutorial, the color palette 1 is assigned the same for all colors.

Color switching function

To be able to use your different color palettes, configure a corresponding function.

To do this, carry out the following steps:

▶ Create a new function

▶ In the **Function selection** dialog that now opens, open the `Application` folder by clicking on `[+]`.

▶ Select the **Switch color palette** function

▶ In the **Switch color palette** dialog that opens, select `Color palette 0` from the drop-down list.

Switch palette

Palette
Palette 0

☐ Show this dialog in the Runtime

OK   Cancel   Help
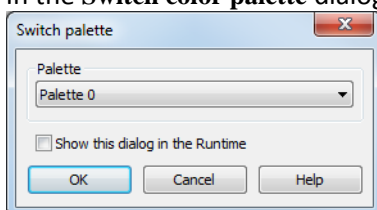
▶ Close the dialog with oκ and switch to the **General** group in the properties window.

▶ In the `Name` properties field, enter *switch_color_table_1_color* as the function name.

> 💡 **Information**
>
> *Select "Offer this dialog in Runtime" in order to give Runtime users the possibility to select the color palettes.*

Configure another color switching function with **Color palette 1** as an objective and *switch_color_table_1_black* as the name. There is the option to can activate **Offer this dialog in Runtime** here.

> 💡 **Information**
>
> *This configuration will be actively used during the progress of this project.*

5.6.2   Global symbol library

You will also use graphics for visualization during the course of this tutorial. You can of course use your own graphics files.

However in this tutorial you will only use graphics included in the zenon global symbol library.

> 💡 **Information**
>
> The global symbol library is a selection of graphics available for your configuration.

6. The Editor - a description

The zenon Editor is the user interface where your projects are "made" and administered. In it, you can image all process steps and make them controllable.

The person is the main focus with zenon. For this reason, the editor also offers much help that will make day-to-day work easier for you.

6.1 The window of the zenon Editor

The zenon Editor is the central element of your project configuration. The Editor window and its functions will be explained here once again in order to aid understanding of the tutorial:

The Editor´s user interface consists of different areas:

Element	Contents
01 - Toolbars:	A collection of Tool bars for the Editor and its modules. They are available for: ▸ Dockable windows ▸ Alignment ▸ Screens / Symbols ▸ Editor profiles ▸ Elements ▸ Menu bar ▸ Production & Facility Scheduler ▸ Remote Transport ▸ Report Generator ▸ Runtime Files ▸ Visibility levels ▸ VBA macros
02 - Project Manager:	Tree view of the Project Manager.
03 - Detail view of the Project Manager:	Details for the module selected in the Project Manager.
04 - Main window:	Main work space; here documents such as screens or reports are displayed.
05 - Toolbar elements:	Toolbar for screen elements - vector elements and dynamic elements.
06 - Properties	Displays the properties of a selected object. Three modes are available and can be selected from the Properties toolbar.
07 - Properties help:	Displays short help messages for properties of screens, variables, functions and other elements which can be engineered.
08 - Output window	Messages are displayed here if projects are compiled and sent to Runtime.
09 - Status bar:	Shows status information for Editor readiness and screens.
10 - Toolbar properties:	Defines display and sorting options of the properties, shows Properties help.
11 - Cross-reference list:	

You choose which windows are shown:

▸ in the `Drop-down list` under `Options` or

The arrangement of the windows can be adjustedindividually. Just the main window cannot be hidden.

6.2　Editor settings

You set the initial settings for the appearance and behavior of the Editor in the basic settings.

This is how you change the basic settings of your Editor:

1. Open the `Settings` dialog from the `Options` menu line
2. Select your settings in the individual tabs

> ### 💡 Information
>
> *zenon offers different options to adapt your Editor to your requirements. It is advisable to make use of the possibilities offered by this adaptation in order to work efficiently.*

6.3　Incorporating menu tools

You can integrate programs on your computer directly into the menu of zenon with the help of the menu tools. The programs that are incorporated individually this way appear at the end of the `Options` menu.

This is how you incorporate tools into the zenon Editor menu:

▶　Open the **settings window** by selecting, in the `Options` menu bar, the `Settings` entry

▶ Go to the `Menu Tools` tab

▶ Add your application with `New...`
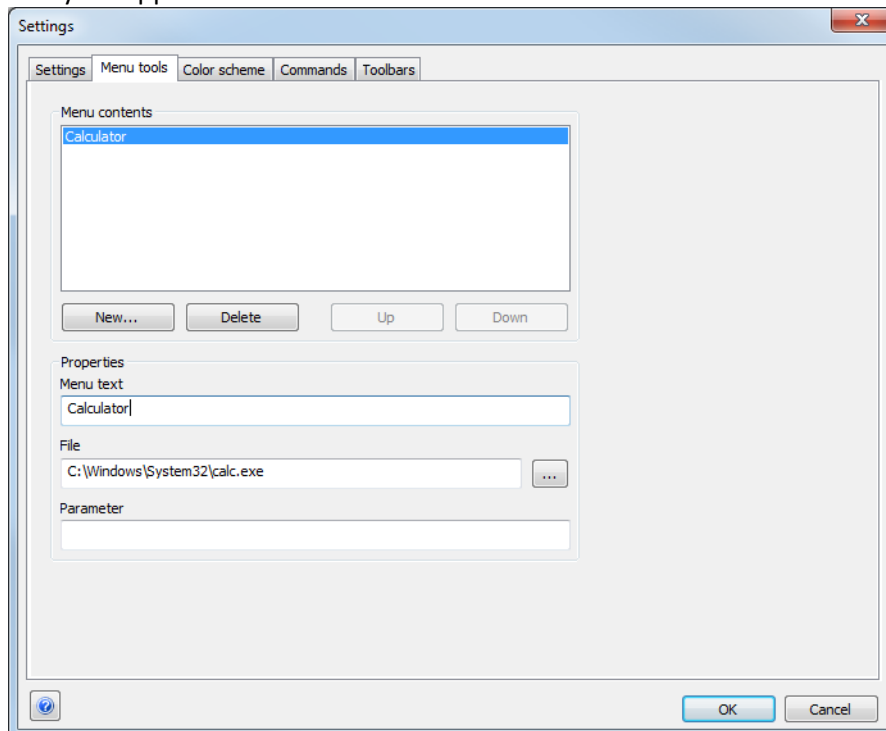
▶ Select the application on your computer - the calculator is added to the Editor in this tutorial. You can find this program on your computer: `C:\Windows\system32,` Filename: `calc.exe`

▶ Name the menu entry in the `Menu text` input field as *Calculator*

▶ Confirm the input with `OK`.

Check your `Tool configuration` menu by opening the `Options` menu bar. The `calc.exe` program with the **Calculator** entry is now at the end of the menu bar.
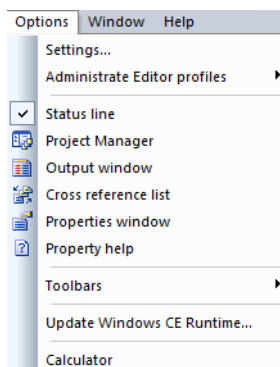
> ### 💡 Information
>
> *You personalize the menu bar of your Editor in order to call up programs quickly in zenon.*

6.4   Editing the window behavior

You can amend the window behavior of your zenon very easily.

It is merely important that you always call up the context menu (`mouse + right click`) directly in the bar with the window heading. In this example, it is on the left next to the `Project Manager` heading.
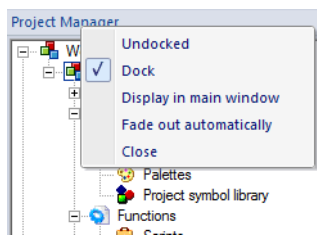
UNANCHORED

This selection opens the selected window in its own window.

DOCK (DEFAULT)

Positions the selected window in the position envisaged in zenon

SWITCH IN THE MAIN WINDOW

Displays the selected window in its own tab in the main window.

FADE OUT AUTOMATICALLY

Hides the selected window. If the mouse is moved over the tab of the hidden window, the window is shown automatically

6.5 Configuring the Editor profile

You can save the window layouts of your Editor permanently using the Editor profile.

THIS IS HOW YOU CONFIGURE YOUR EDITOR PROFILE

- ▶ Call up the Editor profile tool bar. If the toolbar is not displayed, it can be activated using the Menu *Options -> Toolbar*.

- ▶ Compile the window of your Editor according to your optimum requirements for each module. To do this, select a module in the project tree and arrange the window of your Editor as you want it.

- ▶ Select a level of the Editor profile for this view

- ▶ Name this level with `Assign individual names to Editor profile`

- ▶ Save each setting with `Save selected Editor profile`

- ▶ When a module is called up from the project manager node, your Editor automatically switches to the configured view.
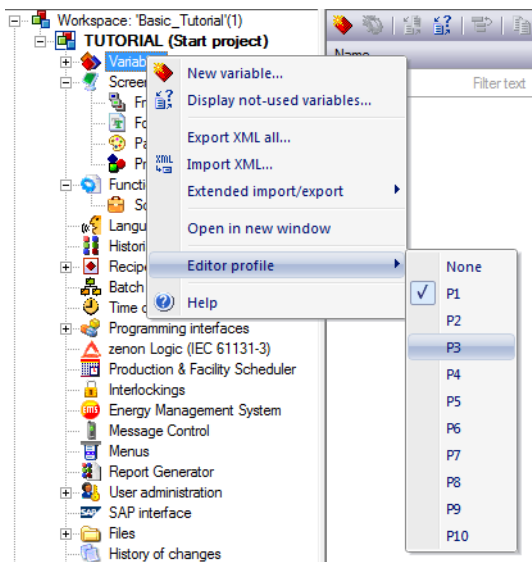
Note: If the automatic saving of the Editor profile function is activated, your settings are saved directly to your personalized profile. An extra save function is thus no longer necessary.

6.5.1 Customizing the Editor

Editor profiles are views of the Editor that can be applied to modules. When the respective module is called up from the project manager, the attendant window is shown in the respective Editor profile.

EXAMPLE:

This function is much clearer when explained using an example. Look at the interaction of the windows in zenon.

- ▶   To do this, call up the 'Variables' node in the project manager.

- ▶   In comparison to this, call up the 'Screens' node in the project manager.

You will notice that an "additional" window is necessary for appropriate configuration of screens.
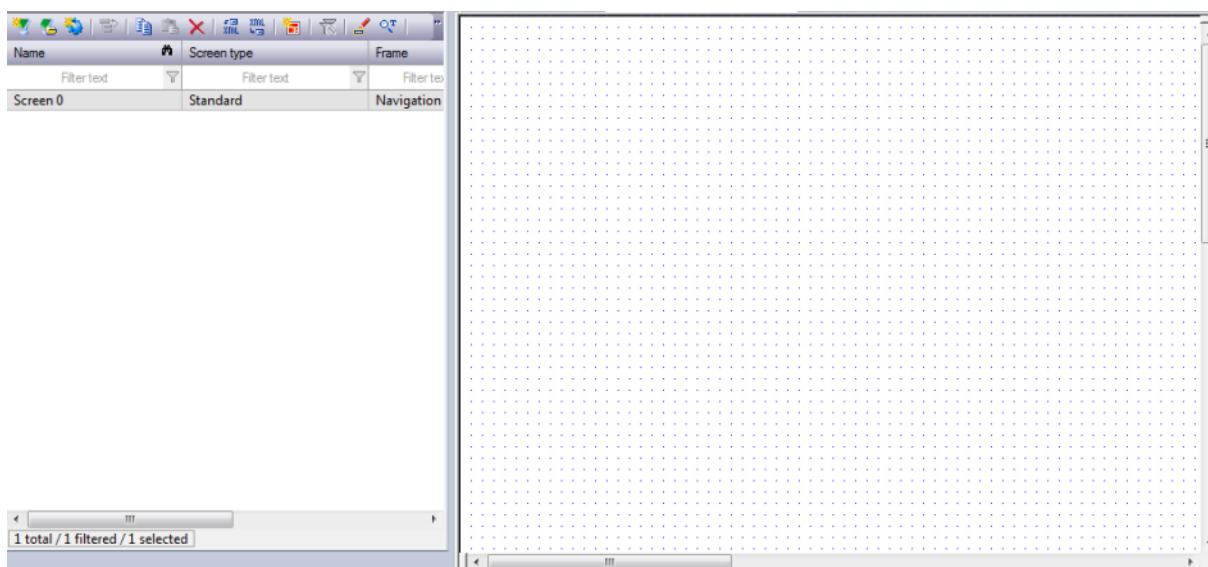
Because this window is not needed for variables, it takes up space as a result. With the help of the Editor profiles in zenon, you can create views depending on program modules.

> 💡 **Information**
>
> You return to the default view with the `Load standard profile` view.

7. Backup

Everybody is familiar with the frustration when the work you have been doing in the past few hours or even days is suddenly lost. A backup can help you here.

Timely and regular updates can save time and avoid frustration. For this reason, this topic is covered at the start of this tutorial.

The following two options are available:

▶   Backup of the workspace

▶   Backup of a project

> 💡   **Information**
>
> Back up your work regularly. Even if you are facing a new challenge, a backup can help you return to a functional state.

7.1   Backup of the workspace

The backup of a workspace includes all projects contained therein. Backups of workspaces are saved as `*.wsb` files. The name can freely be defined.

7.1.1   Create backup

To create a backup of the whole workspace:

▶   Select the entry **Workspace** -> `Create backup` from the context menu of the workspace.
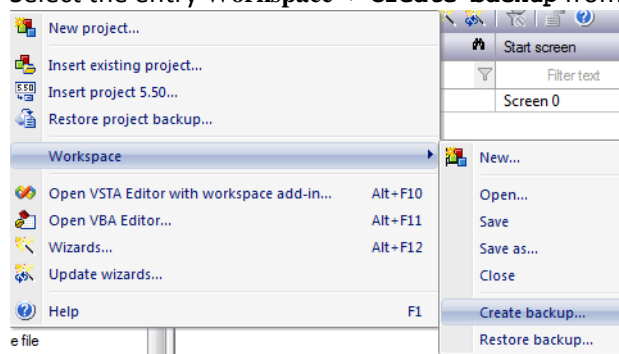
7.1.2   Rollback

With opened/closed workspace:

▶   Select the **Workspace ->Restore backup** entry from the context menu of the workspace.
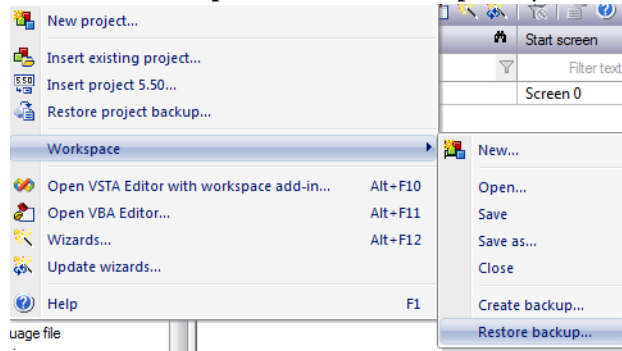
You now can select a backup and define the folder, to which the workspace should be restored.

7.2   Backup of the project

In contrast to backing up workspaces, the project backup options always apply to the selected project only. Other projects in the workspace are not backed up at the same time.

This affects the **General symbol library** most of all, as well as **global projects**.

7.2.1   Versioning of project backups

Project backups can also be versioned. Versions are identified and sorted according to numbering and date/time of their creation.

To activate versioning:

▶   navigate to the `General` node in the project properties

▶   Navigate to the **Versioning** area

▶   Activate the **Versioning** property.

▶   Assign a main number in the **Main version:** property
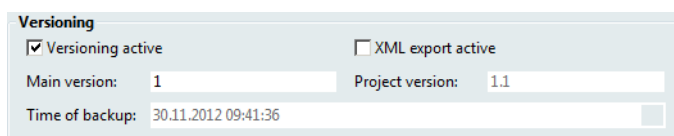
With this for each project backup a number for the project version is created. This consists of the main version and an internally-assigned detail number, separated by a decimal point. This detail number is incremented for each project backup. The version of a project backup is displayed in:

▶ the detail view of node project backup

▶ in the dialog for restoring a backup in the workspace

7.2.2 Project backups

zenon offers its own node in the project tree for making project backups.

You can find these in the project tree at the last position:
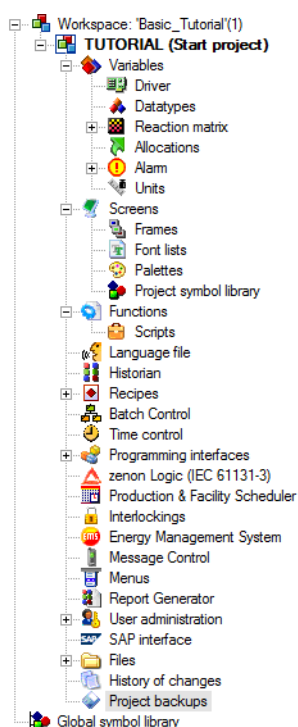
7.2.3 Creating a project backup

In order to make project backups as simple as possible, use the internal project backup in zenon. With this, you do not need to worry about SQL database migrations, issuing permissions and changes to MS-SQL.

To prepare a project backup:

- ▶ In the project manager, change to the 'Project backups' node.

- ▶ With the context menu `(mouse + right click)`, select `Create backup...` .
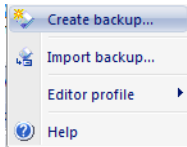
- ▶ In the **backup** dialog that opens, enter `Name` and `Description`.

The project backups that you have previously saved are displayed and administered in the detail view:

File name	♠ Size	Time	Project...	Server	Description
Filter text	Filter...	Filter text	Filter...	Filter...	Filter text
2013-03-22_09-49-04 tutorial.zip	393 KB	22.03.2013 09:4...		☐	erste_sicherung
2013-03-31_15-13-11 tutorial.zip	385 KB	31.03.2013 15:1...		☐	second Backup

7.2.4   Restoring a project

Proceed as follows to restore your project:

- ▶ In the project manager, switch to the 'Project backups' node.

- ▶ Select the desired backup in the detail list.

- ▶ Select, using the context menu or symbol:

  - • Restore backup...

  - • Restore backup as new project...

    > 💡 **Information**
    >
    > *If the backup to be loaded does not appear in your project backup list in the detail window, you must first make this available by importing the backup.*
    >
    > *To do this, select `Import backups...` in the context menu of the project backups*

7.2.5   Forwarding projects

A project backup is also best suited for the forwarding of your project to a project partner or customer.

In this case, carry out the following steps:

- ▶ Select a project backup from the detail list

- ▶ Select the `Export backup...` context menu or symbol

▶ In the subsequent dialog, select the save location of your project backup.

> 💡 **Information**
>
> *Your project backup is automatically saved as a .zip file.*

READING IN A PROJECT

In the event that you are sent a project configuration for further work, the procedure is the reverse:

▶ First import the project backup.

▶ Then select the project backup that has just been imported in the detail list.

▶ The procedure is the same if you read back a backup

7.2.6   save project copy

The `Save project as ...` command creates a copy and automatically inserts this to the current workspace. You can also use this command to create a copy of your configuration and to then forward this.

To create a project backup:

▶ Select the project in the project tree.

▶ In the `Project manager`, right-click on **Project** and select `Save as....` .

▶ The dialog for project backup is opened:
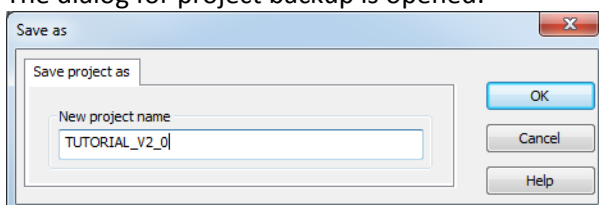
> 💡 **Information**
>
> At the backup of the workspace, all projects of the workspace are backed up, too.
> They are however not contained in the project back up and cannot be restored
> individually.

7.3 Backup Comparison Wizard

If you want to compare two backups with one another, you use the corresponding wizard.

zenon provides the `Backup Comparison Wizard` in the Project folder. This compares two the desired content in two backups.

This is how you call up the Backup Comparison Wizard:

▶ In the menu bar, select `File -> Wizards`

▶ Open the Project folder and select the **Backup Comparison Wizard**
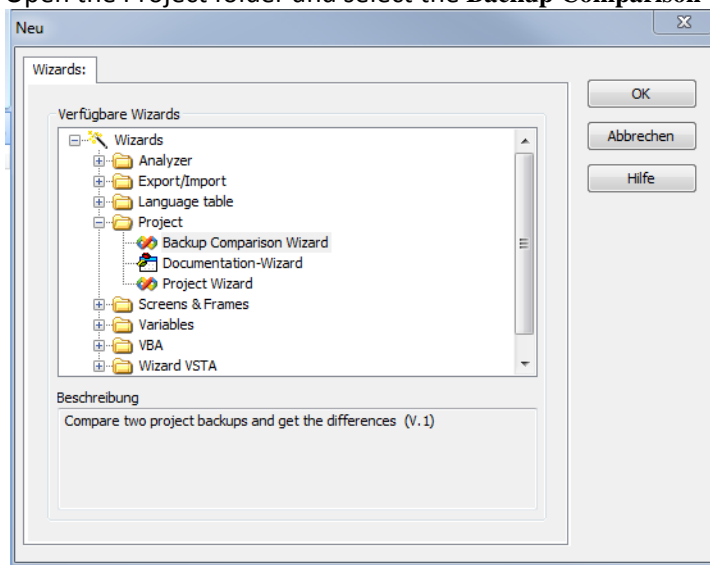
In the 'Settings' tab, select:

▶   The backups that you want to compare (latest version / older version)

▶   The contents of the module that you want to compare.
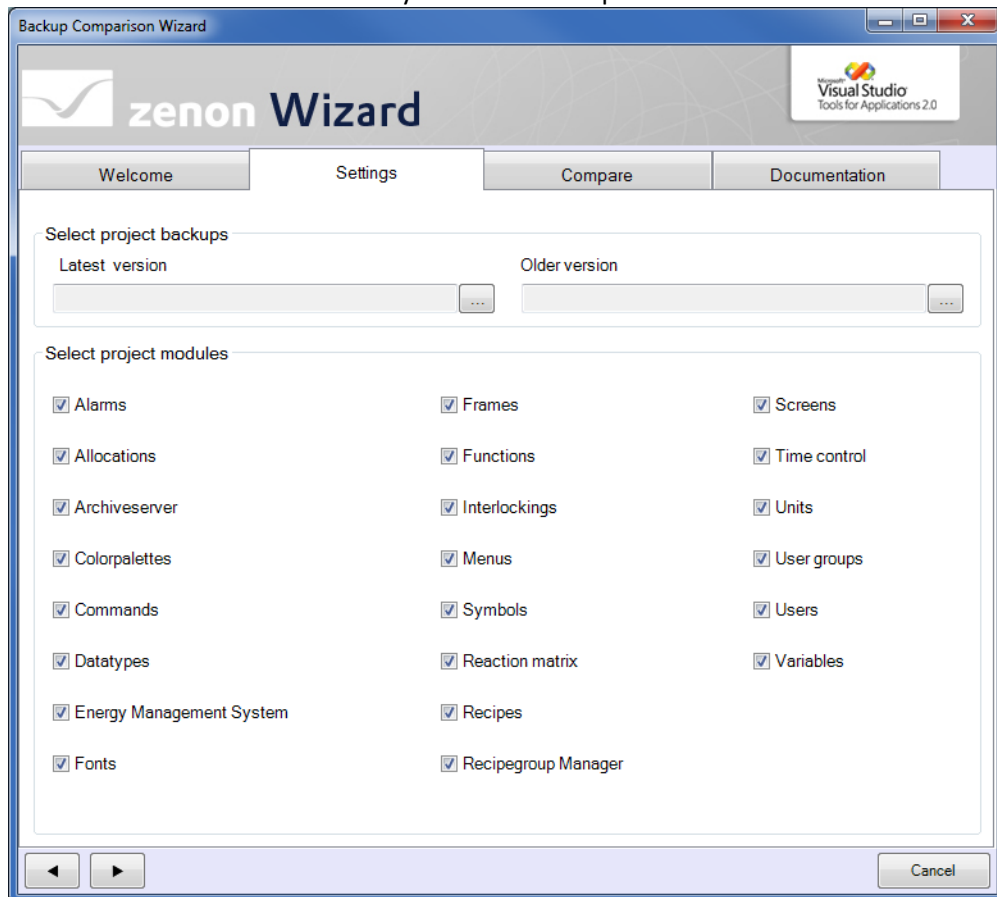
8. Variables

In this step, you will learn how   zenon connects to a process and how it receives values from the PLC.

To do this:

▶   A driver is created for communication with the PLC.

▶   Simple and structured data types are used and configured

▶   Simple and structures variables are configured, based on data types and drivers

A driver is created in the next tutorial step. The corresponding variables are then defined.

> **Information**
>
> *Communication to a PLC is enabled with the help of drivers.*
>
> *Data types define the logic.*
> *Variables interact using this method of communication and define control processes.*

8.1 Drivers

To communicate with a data source (PLC field bus, etc.) it is necessary to connect to a driver. Drivers are the interface between zenon Runtime as the HMI and the PLC as the point of execution. Different PLCs for different equipment can be addressed with the respective driver in zenon.

At the start of the project, the desired drivers must be defined (depending on the PLC, etc.) and their process variables must be created.

> **Information**
>
> In Hardware mode, zenon would immediately attempt to establish a connection to the PLC when Runtime is started. Because there is no PLC available in this tutorial, zenon would display all values as invalid.

8.1.1 Configuration of a new driver

To create a new driver:

▸ Open the node `Variables` in the project manager.

▸ Open Activate the context menu of the `Drivers` entry with the right mouse button.

▸ Select the menu entry `Driver new`.
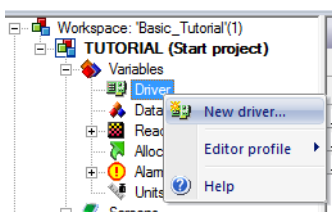
The dialog box to select a driver opens.

In the selection for available drivers, scroll until the Siemens folder appears.

Note: entering the first letter moves the selection to the folders with the corresponding letters.

Example: Entering `si` jumps to the `Siemens` folder.

▶ Open the tree view by clicking on `[+]`.

▶ Choose the driver S7 TCP-IP. The entry is put in the field Driver name as shown in the dialog box **Definition of driver**. Leave the dialog box by confirming the selection with **OK**.
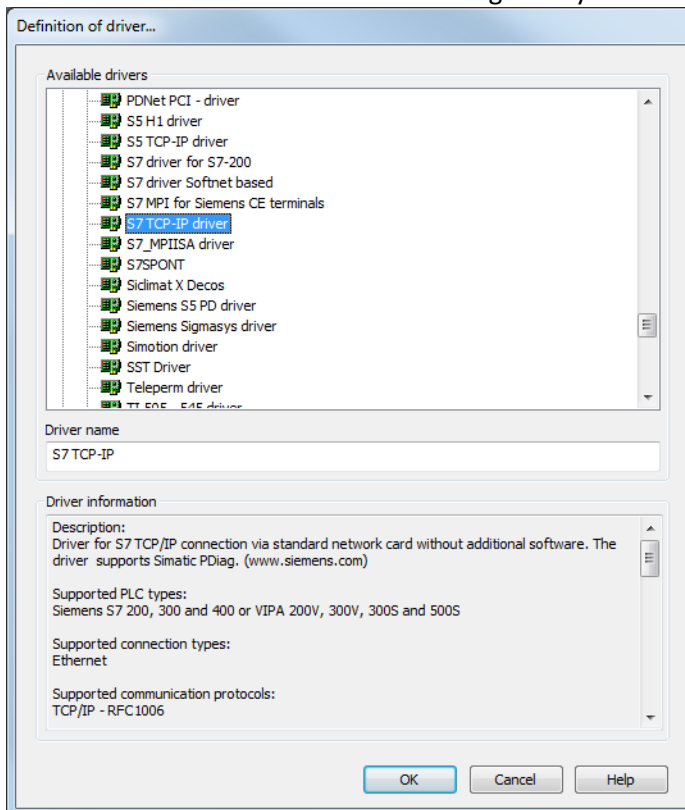
Now the dialog for the configuration of the driver opens.

💡 **Information**

Drivers for internal variables, math variables and system variables are automatically preinstalled when a project is created.

8.1.2 Driver configuration

Now configure you driver:
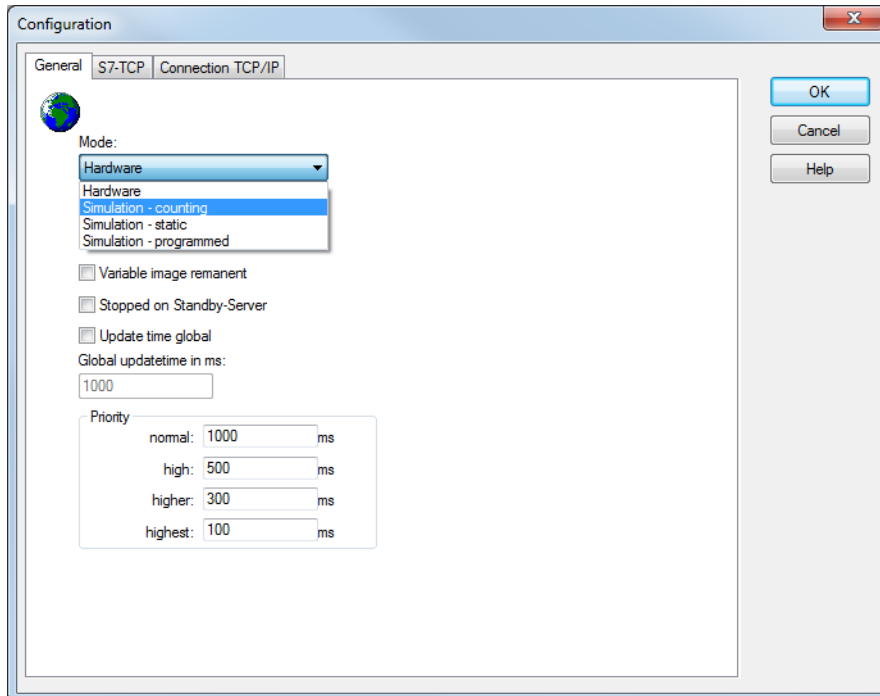
▶ Set the mode to Simulation (counting).
Background: In `Hardware` mode, zenon would immediately try to connect to a real PLC when Runtime is started. However because this is very probably not available, zenon would display all values as invalid if the `Hardware` mode is selected.

▶ Switch off global update time.
Background: With this setting, there is the possibility to assign one of four different update times to each individual variable when defining variables. These four update times can be set in the lower part of this dialog, under `Priority`.

The other tabs of this dialog are driver-specific, i.e. different according to the selected driver. These other settings can be disregarded in this tutorial, because no real PLC is being addressed.

8.2 Data types

A data type is a variable template without connection to the process. This connection to the process is only established with the driver object type for the variable.

In addition to the pre-defined data types, you can create your own data types. Two possibilities are available here: Simple data types and structure data types.

In this step, you will learn how to create simple and structure datatypes.

> ### 💡 Information
>
> The properties are each referenced by the data type on which they are based. If a property of the data type is changed, it is also changed for all variables that have been created with this data type. As a result of this, changes to a data type do not need to be "changed" individually for each variable.

8.2.1 Pre-installed data types

These are supplied with zenon as standard and are immediately available when a new variable is created. They conform to the standardized simple data types of IEC 61131-3 such as   INT, USINT etc. They define a set value range (fixed upper and lower limit) with a defined number of values. That is why real numbers can only be depicted as float-point numbers with a certain accuracy.

> ### 💡 Information
>
> The properties of the simple data types can be changed. The names of these data types always match with the names of the basic IEC data types, these and the IEC data types cannot be deleted or renamed.

State	Name	Description	IEC data type
F... ▽	Filter text ▽	Filter text ▽	Filter text ▽
	UDINT	Simple datatype	UDINT
	INT	Simple datatype	INT
	WORD	Simple datatype	WORD
	LINT	Simple datatype	LINT
	DINT	Simple datatype	DINT
	UINT	Simple datatype	UINT
	DWORD	Simple datatype	DWORD
	WSTRING	Simple datatype	WSTRING
	USINT	Simple datatype	USINT
	STRING	Simple datatype	STRING
	BYTE	Simple datatype	BYTE
	TOD	Simple datatype	TOD
	SINT	Simple datatype	SINT
	TIME	Simple datatype	TIME
	DATE_AND_TIME	Simple datatype	DATE_AND_TI...
	REAL	Simple datatype	REAL
	BOOL	Simple datatype	BOOL
	ULINT	Simple datatype	ULINT
	DATE	Simple datatype	DATE
	LREAL	Simple datatype	LREAL

The data types only become visible in zenon, once a driver is created. Each driver opens a list of data types it supports.

⚠ **Attention**

Changes in the properties of the data types affect all linked variables and thus also affect all linked structure data types.

8.2.2 Overview of the data types supported by zenon:

Short name	Long name	Comment / value range	Number of bits
BOOL	Boolean	Bit: 0/1	1
BYTE			
SINT	Short integer	Signed byte: -128 to 127	8
USINT	Unsigned short integer	Byte: 0 to 255	8
INT	Integer	Signed word: -32768 to 32767	16
UINT	Unsigned integer	Word: 0 to 65535	16
DINT	Double integer	Signed double: -2147483648 to 2147483647	32
UDINT	Unsigned double integer	Double: 0 to 4294967295	32
LINT	Long Integer	Because zenon cannot hold the value range of a 64 bit number, the actual value range is restricted to 52 bits. Numbers with signs are possible from –2251799813685248 to 2251799813685247. Numbers outside of this range cause a overflow or underflow, according to a 52-bit integer. The following components work with a full 64 bit resolution in the zenon Runtime: 1. The shared memory VBA Interface, when the value was entered in signal resolution. 2. The shared memory zenon Logic interface. 3. The driver kit works for the whole range of 64 bits, and so do all the drivers, provided the driver supports this IEC data type. At the moment, these are `Sample32`, `Internal` and `STRATON32`. RGM and recipes cannot save even a 52 bit value with full accuracy.	64 or 52

ULINT	Unsigned long integer	Because zenon cannot hold the value range of a 64 bit number, the actual value range is restricted to 52 bits. Numbers without signs are possible from 0 to 4503599627370496. Numbers outside of this range cause a overflow or underflow, according to a 52-bit integer. The following components work with a full 64 bit resolution in the zenon Runtime: 1. The shared memory VBA Interface, when the value was entered in signal resolution. 2. The shared memory zenon Logic interface. 3. The driver kit works for the whole range of 64 bits, and so do all the drivers, provided the driver supports this IEC data type. At the moment, these are Sample32, Internal and STRATON32. RGM and recipes cannot save even a 52 bit value with full accuracy.	64 or 52
REAL	Real numbers	Real numbers	32
LREAL	Real numbers	Real numbers	64
STRING	Variable-length single byte character string	ASCII string (the max. string length depends on the driver). The string length is set to 5 characters by default. The correct length is to be given during configuration if necessary.	>= 8 x string length
WSTRING	Variable length multi-byte character string	Contains multi-byte strings, for example strings in Unicode coding (UTF-8 etc). The string length is set to 5 characters by default. The correct length is to be given during configuration if necessary.	>= N x string length
DATE	Date	IEC date in steps of 1 day	16
TIME	Duration	Duration in the IEC format. IEC time in steps of 1 ms, signed integer	32
TOD	Time of day	Time of day in steps of 1 ms	32
DATE_AND_TIME	Date and time of day	Defines a period of time with 64 bit and is saved in a binary-coded decimal format.	64

> ⚠️ **Attention**
>
> A subsequent change of the IEC data type explicitly is not recommended!
> A number of settings depend on the IEC data type and have to be re-configured by hand if the
> IEC data type is changed. (Example: Signal resolution, measuring range, hysteresis, set value
> limits, non-linear value adjustment, etc.).

8.2.3   Creating a new simple data type

You can install the your own data types based on pre-installed data types. These each relate to the
respective basic data type.

The advantage: If a basic or structure data type is amended, these changes are forwarded to all related
projects.

For the creation of a simple data type:

- ▶ In the project manager, open the `Variables` node.

- ▶ Open the context menu of the `Datatypes` entry with the right mouse button.

- ▶ Select the menu entry `Structure datatype new`.
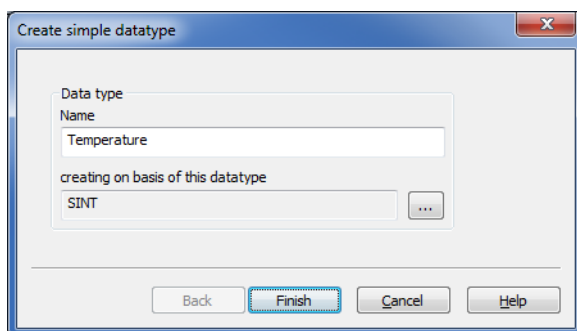
- ▶ Enter `Temperature` as a name for the new simple data type.

- ▶ Select `SINT` as the basic data type.

By clicking on `Finish`, the datatype is created and will then be available in the list of datatypes.

> 💡 **Information**
>
> *Values of -128 to 127 can be edited by selecting the basic data type.*

Configuration of the data type

Once a data type has been created, the parameters are set in the properties window.

Carry out the following steps to amend them to the requirements of this tutorial:

▶ Select the `Temperature` data type from the list.

The properties of the `Temperature` data type are now shown in the property window and can be configured there.

▶ Open the **General** group.
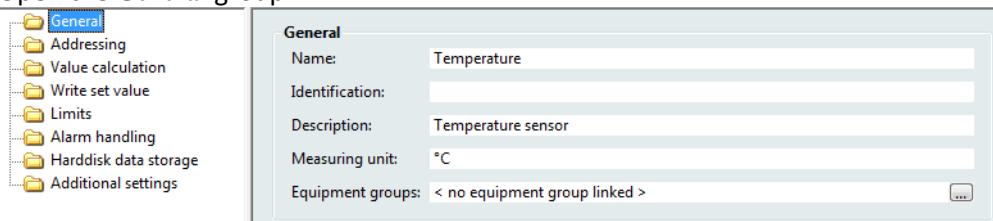
▶ Open the **Value calculation** group.
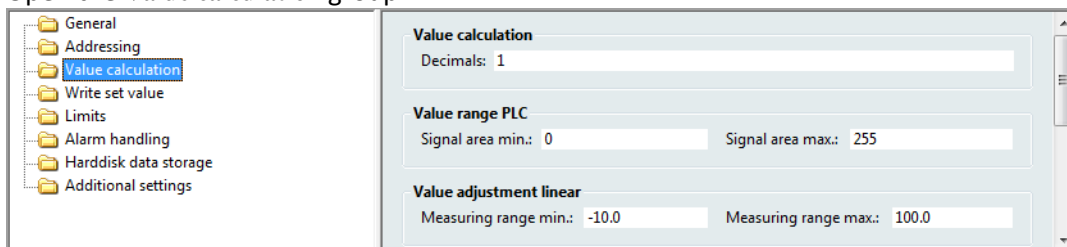
▶ Change the properties `Unit`, `Decimals`, `Linear value adjustment` and `Value range PLC` of the datatype as shown in the illustrations above.

Value adjustment linear	Numerical value of temperature sensor; In this example, it defines the lowest and the highest temperature that is measured. In this example tutorial, this is a minimum of -10.0 °C and a maximum of +100 °C.

Setting limit values

Limits have the task to trigger a reaction, as soon as a limit or a status is violated or reached. The limit is entered in the Editor and is fixed for this tutorial.

Limit values can be configured individually per variable or structured in data types.

The kinds of reaction can be manifold.
For example: Color changes, alarms, CEL entries, printouts, function calls, etc.
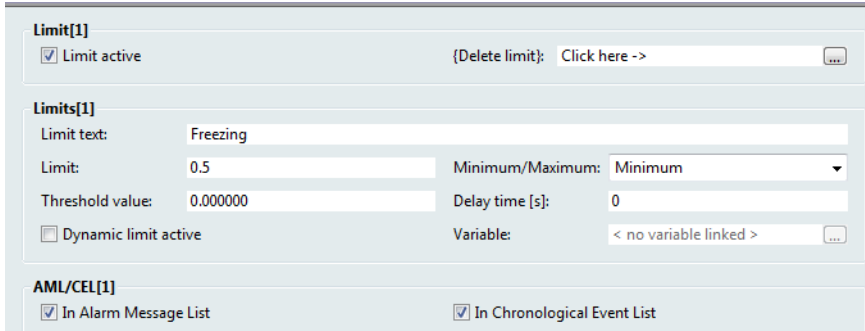
To create a limit value:

▶ Select the **Temperature** data type from the list of data types.

▶ Open the section `Limits` in the properties window.

▶ Select the `<New limit value>` entry.

A new section with the name `Limit[1]` is created.

▶ Open the section Limit[1].
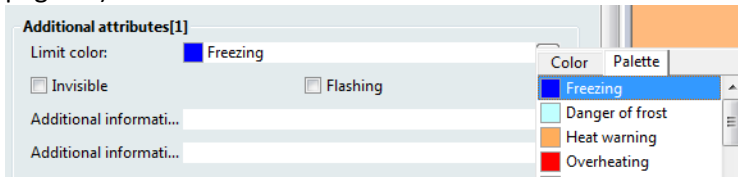
▶ Set two further limit values in accordance with the following:

Limit	Name
5 °C (minimum)	Risk of freezing
75 ° C (maximum)	Heat warning
90 °C (maximum)	Overheating

▶ In the **Additional attribute[1]** property field, select the limit value color from the color palette (on page 22).

Repeat this parameter setting for all limit values.

💡 **Information**

The configuration of limit values defines the behavior of numerical variables.

8.2.4 Creating a new structure datatype

Structured data types have, in contrast to simple data types, more complex structures, because several data types can be combined into one. A structure is actually just a hull that bears the name of the

structure but does not have any properties of its own. These only come with the structure elements, which you simply add to the structure data type (cf. creating a structure element).

▶ Open the entry `Variables` in the Project Manager.

▶ Activate the context menu of the entry `Datatypes` with the right mouse button.

▶ Select the menu entry `Structure datatype new`.
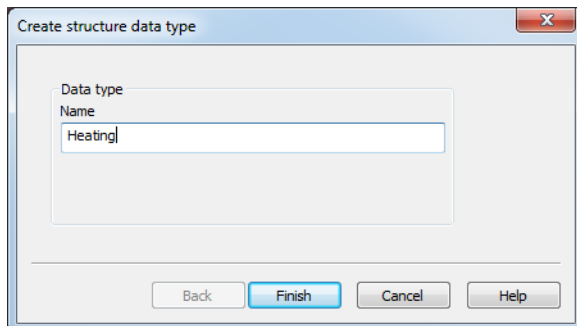
▶ Enter `Heating` as the name of the new structure data type.

CREATING A STRUCTURE ELEMENT

After you are done with the structure datatype a dialogbox will open, in which you can create the first structure element of this data type.
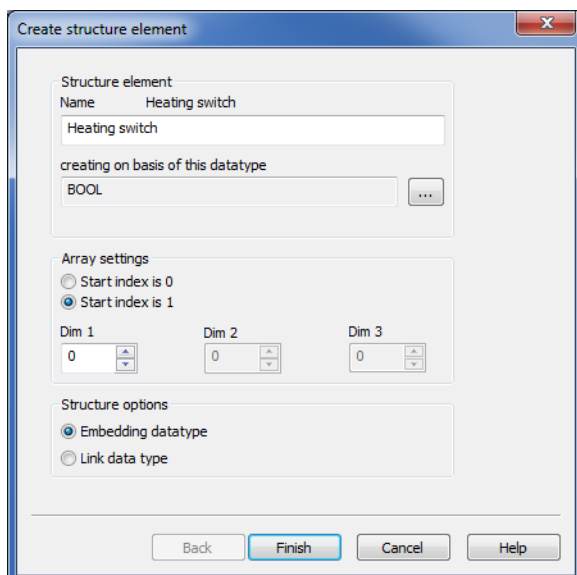
▶ Enter `Heating switch` as the name of the new structure element.

▶ Select `BOOL` as the basic data type. The data type should be embedded.

> 💡 **Information**
>
> *If a basic data type is embedded in a structure data type, the properties of that data type can be changed in the structure element independently from the basic data type. There is no "inheritance".*

After finishing the structure element the structure data type is available in the list.

Now create other structure elements in this data type:

▸   Open the context menu of the 'Heating' structure data type with the right mouse button.

▸   Select the `New structure element`… menu entry.

TAGs	Description
Name	Temperature sensor
Data type basis	Temperature
Structure options	Link data type

Project configuration for the datatypes of this tutorial

In addition to the previously-configured heating structure data type, configure the following structure data types:

TANK

Name	Data type	Description
Inflow - switch	Boolean	embedded
Outflow - switch	Boolean	embedded
Heating element	Heating structure data type (already-configured)	Linked

> **Information**
>
> *By adding the heating structure data type, all structure elements contained therein are accepted in the tank structure data type.*
>
> *In this tutorial, the temperature sensor contained in the heating structure element is used for both the display of the water temperature and for the automatic control of the heating.*

MOTOR

Name	Data type	Description
Fuel tank	USINT	embedded
Speed	UDINT	embedded
Motor temperature	Temperature (structure element)	embedded

The temperature data type is embedded in this example. Independent limit values can be configured as a result of this.

Differentiation: Embedded - linked

You have used two types of inclusion when creating the **structured data types** for *heating*: linked and embedded. A short explanation of the difference:

LINKED:

The linked data type is only a reference to an pre-existing data type. This means you take over all properties from that data type into your structure, except the name.

The advantage: If your structure element has many properties of an already existing data type, you can simply reuse that data type, again applying the central approach. If the basic data type is changed, all linked elements are changed in the same way!

> ### 💡 Information
>
> *Of course, you can still change each of the variable properties or separate them from the data type. However, if you want to ensure that the data type is only used for this structure and you want to be able to make changes to the addressing settings, choose your own embedded data type. This creates a new data type that can only be used within this structure.*

EMBEDDED:

Embedded elements can have properties that differ from their original data types. The properties are defined individually for each structure element and are saved there.

If the original data type is changed, the embedded variables do not inherit these changes.

8.2.5   Tips for data types

▶ Whenever possible, create your data type as a structure element.

▶ Note the naming here. Differentiation: Tank < = > storage tank

- Tank is more general, and thus can be used more variably

- Storage tank is specific and thus easier to identify, however not to be used for oil tank
  Recommendation: Use general descriptions for structure data types.

▶ Create overlaps when configuring your project in order to allow re-use

▶ Use unique object names which differ from each other not only by case sensitivity.

> ### 💡 Information
>
> *zenon provides an initial setup for common data types for basic installations. These basic data types are in line with the IEC 61131-3 standard.*

8.3   Variables

A process variable is the interface between the data source (PLC, field bus, etc.) and zenon. For correct detection and open-loop/closed control of a process, it is necessary to have an exchange of process data

on the one hand and the input of setpoint values and commands on the other hand. Process data and settings are defined for single process variables and the parameters are entered and changed in the variable list.

HOW VARIABLES WORK

Each variable is based on an data type (IEC data type or your own structure data types). On creating a new variable a data type has to be selected. The data type has the same properties as the variable itself (unit, signal resolution, limits, etc.).

All properties of the data type are inherited to the variable. A reference to the properties of the data type is established, i.e. if a property of the data type is changed, it is also changed in all variables linked to that data type.

In this step, you will learn how to create single variables and variable arrays, which are based on simple or structure data types.

> **Information**
>
> *Many zenon drivers offer you the possibility to import variables from the PLC program or directly from the PLC, which means that you do not have to create the variables manually.*
>
> *You will find the necessary information for the variable import in the according driver documentation.*
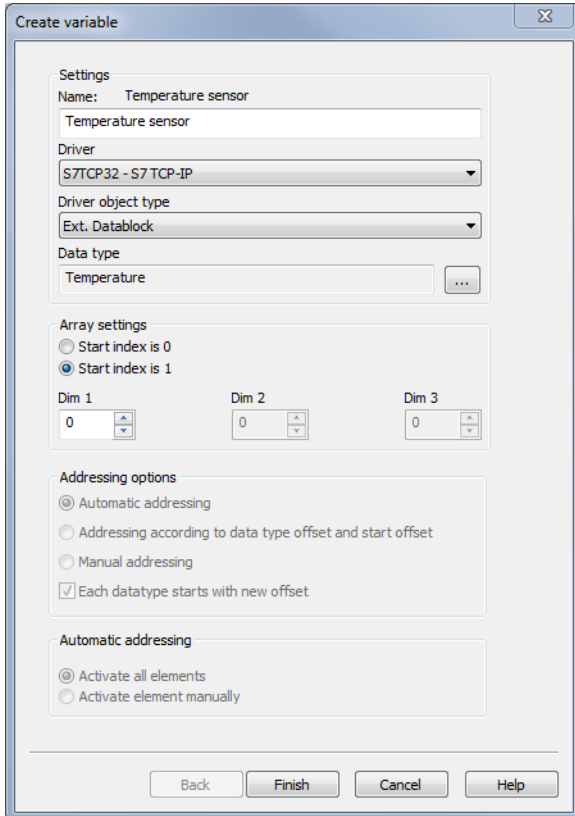
Because you have neither a PLC nor an S7 project in this example project, the variables will be created manually.

8.3.1 Creating a new simple variable

Carry out the following steps to create a variable:

- ▶ In the project manager, activate the context menu of the **Variables** entry by right clicking.
- ▶ Select the menu entry `Variable new`.

The dialog box for creating a variable opens.

- ► Enter `Temperature sensor` as the name.
- ► Choose the driver `S7TCP32 – S7 TCP-IP`
- ► Select **Ext.** as the driver object type **Data block**.
- ► Select `Temperature` as the data type.
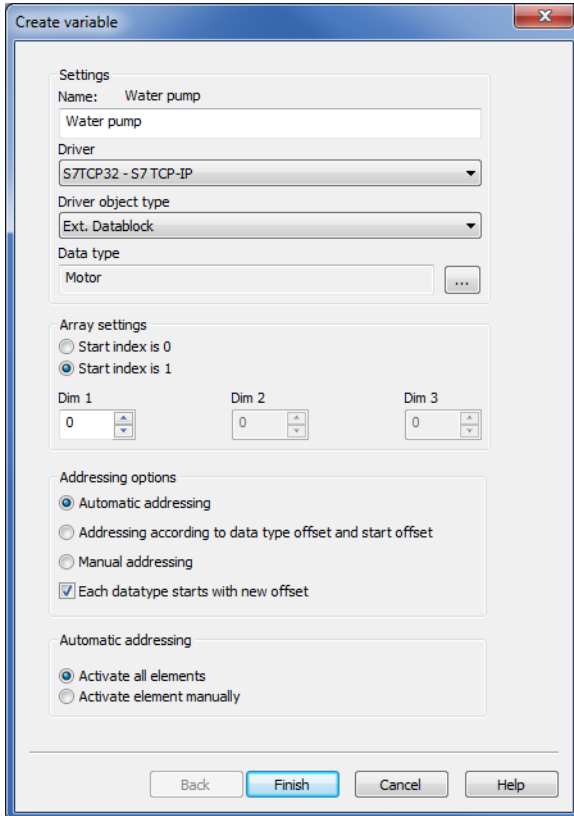
> 💡 **Information**
>
> *Always use logical names for the process variables in order to keep a better overview.*

8.3.2   Creating a new structure variable

You will now create other variables, but this time for a structure data type. Therefore proceed as described above.

- ► Enter `Water pump` as a name.
- ► Select the datatype `Motor`.

▸ Under 'Array', set `Start index is 1` settings.

▸ Activate the option `Automatic addressing`.

▸ Leave the `Each data type starts with a new offset` and `Activate all elements` settings as proposed by the system.

▸ Select `Ext.` as the driver object type `Data block`.

▸ Set the start offset to 21.
Note: You set the start offset in the **Addressing** property group in the `Start Offset:` property field

Variables with arrays

You can create multiple variables of the same type by changing the corresponding settings in the **array settings** area.

Imagine the following task:

In this tutorial project, a route of approximately 1,000 meters is to be laid between a stream and a storage tank. The temperature is to be measured every 200 meters.

For this task, you will configure the four heating stations in just one work step.

Carry out the following steps:

▶ In the project manager, activate the context menu of the **Variables** entry by right clicking.

▶ Select the menu entry `Variable new`.

▶ Name this/these variable(s) *Heating line*.

▶ Select *Heating* as the data type (the previously configured data type).

▶ Set the **Dim 1** array to *4*.
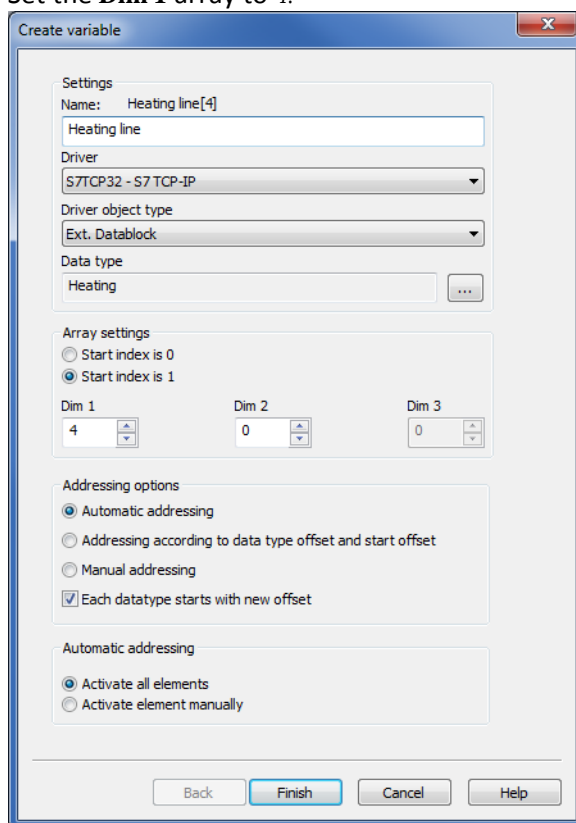
▶ Set the settings you require according to the image.

▶ Confirm the settings with `Finish`.

The variables now are added to the variable list in the detail view of the project manager. All properties of the *Heating* structured data type are thus carried over into the variables. The properties of the selected variable can be checked and changed in the properties window.

Four heating units, including the variables contained in these are created by changing the array dimension. These variables are automatically addressed due to the settings, i.e. you only need to issue one **Start Offset**. The other addresses are calculated automatically by zenon.

▶ In the 'Addressing' property group, set the **Start offset** property to `50`.

Configuration of variables

Then create a variable with two arrays for your storage tank:

- ▶ Create a new variable.
- ▶ Name this variable *Storage tank*.
- ▶ Select *Tank* as the data type (the already-configured data type).
- ▶ Set the **Dim 1** array to 2.
- ▶ Confirm the settings with `Finish`.
- ▶ In the 'Addressing' property group, set the **Start offset** property to `100`.

The variable list in the detail view of the project manager now should have the following entries.

Name	Identification	Measur...	Driver	Data block	Net address	Offset
Filter text	Filter text	Filter...	Filter text	Filter text	Filter text	Filter text
Temperature sensor		°C	S7TCP32 - S7 TCP-IP	0	0	0
Water pump			S7TCP32 - S7 TCP-IP	0	0	0
Water pump.Fuel tank			S7TCP32 - S7 TCP-IP	0	0	0
Water pump.Speed			S7TCP32 - S7 TCP-IP	0	0	1
Water pump.Motor temperature		°C	S7TCP32 - S7 TCP-IP	0	0	5
Heating line			S7TCP32 - S7 TCP-IP	0	0	0
Heating line[1].Heating switch			S7TCP32 - S7 TCP-IP	0	0	0
Heating line[2].Heating switch			S7TCP32 - S7 TCP-IP	0	0	1
Heating line[3].Heating switch			S7TCP32 - S7 TCP-IP	0	0	2
Heating line[4].Heating switch			S7TCP32 - S7 TCP-IP	0	0	3
Storage tank			S7TCP32 - S7 TCP-IP	0	0	0
Storage tank[1].Inflow - switch			S7TCP32 - S7 TCP-IP	0	0	100
Storage tank[1].Outflow - switch			S7TCP32 - S7 TCP-IP	0	0	100
Storage tank[1].Heating element						
Storage tank[1].Heating element.Heating swi...			S7TCP32 - S7 TCP-IP	0	0	101
Storage tank[2].Inflow - switch			S7TCP32 - S7 TCP-IP	0	0	102
Storage tank[2].Outflow - switch			S7TCP32 - S7 TCP-IP	0	0	102
Storage tank[2].Heating element						
Storage tank[2].Heating element.Heating swi...			S7TCP32 - S7 TCP-IP	0	0	103

Configuring the view

If you do not have all the described columns displayed in the overview of the detail list for the variables of a water tank that have been created, you can display these.

Carry out the following steps to display columns in the detail list:

Name	Identification	Measur...	Driver	Data block	Net address	Offset	Bit r
Filter text	Filter t			Filter text	Filter text	Filter text	Filte
		Remove all filters					
Temperature sensor			S7 TCP-IP	0	0	0	
Water pump		Group on this column	S7 TCP-IP	0	0	0	
Water pump.Fuel tank		Resolve grouping	S7 TCP-IP	0	0	0	
Water pump.Speed			S7 TCP-IP	0	0	1	
Water pump.Motor temperature		Column selection...	S7 TCP-IP	0	0	5	
Heating line		Format columns...	S7 TCP-IP	0	0	0	
Heating line[1].Heating switch		Adjust size to window width	S7 TCP-IP	0	0	0	
Heating line[2].Heating switch		Remove selected columns	S7 TCP-IP	0	0	1	
Heating line[3].Heating switch			S7 TCP-IP	0	0	2	
Heating line[4].Heating switch		↑ Sort ascending	S7 TCP-IP	0	0	3	
Storage tank		↓ Sort descending	S7 TCP-IP	0	0	0	
Storage tank[1].Inflow - switch		Help	S7 TCP-IP	0	0	100	
Storage tank[1].Outflow - switch			S7TCP32 - S7 TCP-IP	0	0	100	

▶ Place your mouse cursor over the heading column of the variables in the detail view

▶ Select **Select column...** from the context menu.

▶ Select `Offset` from the context menu in the left list of available columns and add this, with `Add ->`, to the `Selected columns:` view.
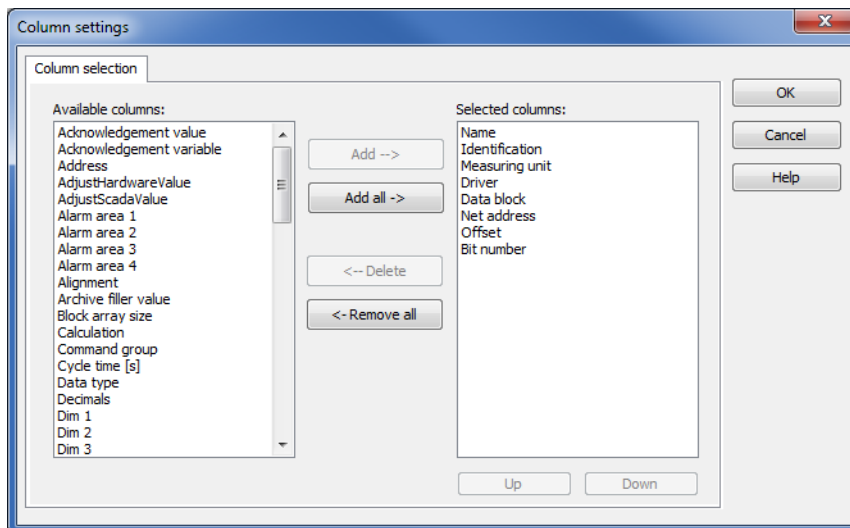
9. Functions

Working steps are shown in zenon with functions.

▶ Functions are available, sorted into groups.

▶ Often-used functions can be efficiently managed via 'Favorites'.

▶ Several functions can be combined into a function using scripts.

9.1 Configuring functions

To call up functions, proceed as follows:

▶ In the project manager, open the node `Functions`.

▶ Open Activate the context menu of the `New functions` menu with the right mouse button.

▶ Select the `End Runtime` menu entry.
This function is in the 'Application' function group.

▶   Name your newly-created function *RT-quit*.

▶   Create another "**reload**" function and name this *RT-reload*.
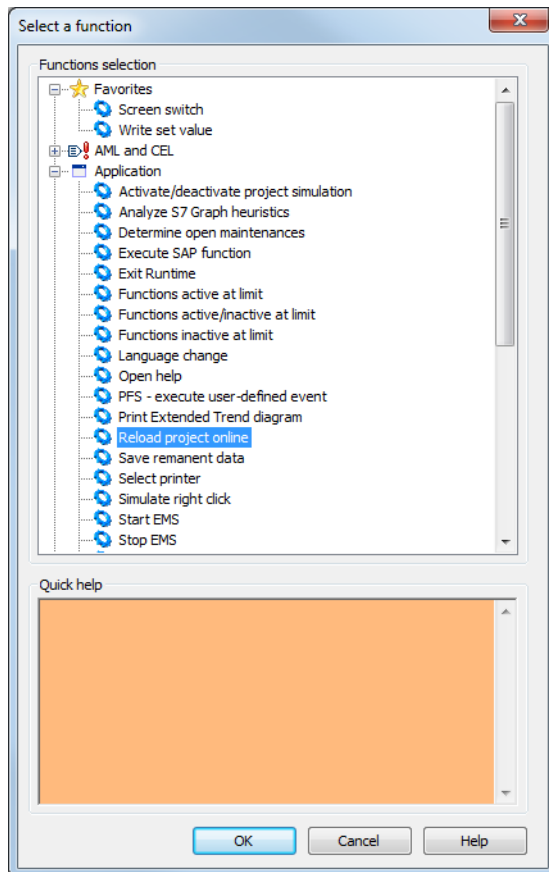
9.2   Function favorites

You will use functions frequently. You can use 'Favorites' to carry out your work more efficiently.

This is how you create function favorites:

▶   In the project manager, open the node `Functions`.

▶   Open Activate the context menu of the `New functions` menu with the right mouse button.

▶   Select the desired function and move this with *the mouse button held down* to the first **Favorites** folder.

▶ Alternatively, press the left mouse button and select `Add to favorites` in the context menu.
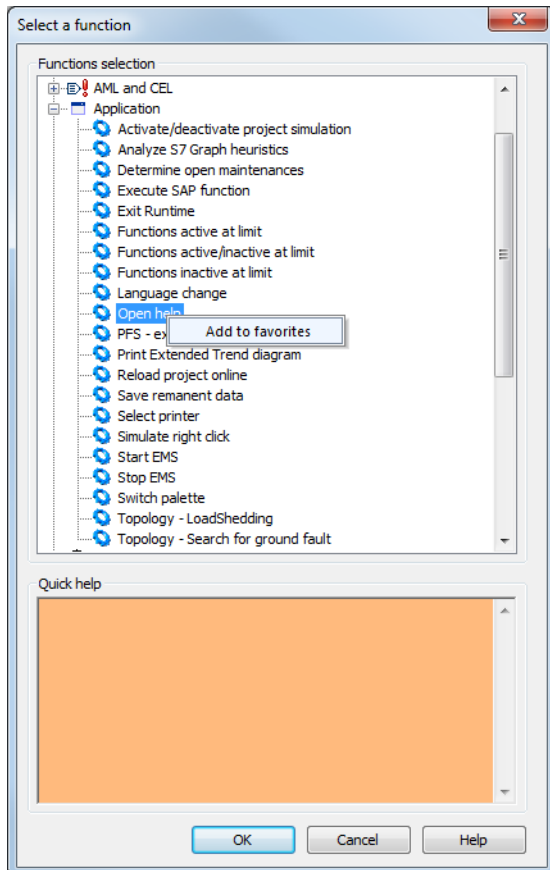
10. Frames and screens

In this step, you will learn how to create screens in zenon, allowing you to visualize and operate facilities.

> 💡 **Information**
>
> *Screens define the look of your project configuration.*

10.1 Distinction screens - templates

Two types are distinguished for visualizing your configuration:

FRAMES

Frames form the basis for the layout of the window and the screens displayed during the Runtime. You structure the display on the screen and determine the position. Each screen is assigned to a frame and displayed in the screen area defined by the frame. Templates determine general window properties such as position, size and appearance.

SCREENS

Screens are central elements of a project. They display the configured equipment, inform, and provide user elements.

> 💡 **Information**
>
> *Templates define the frame.*
>
> *Screens define the contents of these frames.*

10.2   Frames

A frame is the basis for the window technique. This is where general settings are made, like:

-   ▶   Size of frame = Size of screen

-   ▶   Position of the screen on the screen

-   ▶   Breakdown of the screens if several monitors are used ("What is shown and where?")

This offers the following advantages:

-   ▶   A consistent structure of screens throughout the project.

-   ▶   Size and position only has to be set once.

-   ▶   Changing a frame will change all the screens based on it.

-   ▶   Functions can especially refer to one frame (Alarms: Ackn. flashing, Hardcopy, etc.)

-   ▶   Screens rarely have to be closed by hand, as there can only be one screen of a frame open at a time.

DEFAULT TEMPLATES

When creating a new project, the template for alarm messages is configured automatically; template name: *ALARM STATUS LINE*.

Note: When using the project wizard, the templates *MAIN*, *NAVIGATION* and *SUBNAVIGATION* are created.

10.2.1 Create new frames

At least one frame must be created before the first screen can be configured.

Carry out the following steps to create a template:

▶ Open the node `Screens` in the Project Manager and select the entry `Frames`.

▶ Create a new frame with `New frame` in the context menu.
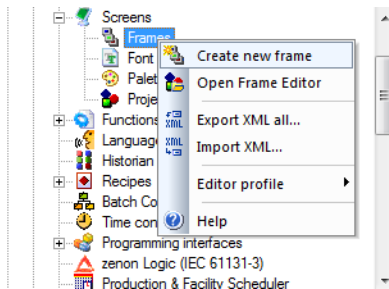
a frame with the name `Frame 1` is automatically calculated.
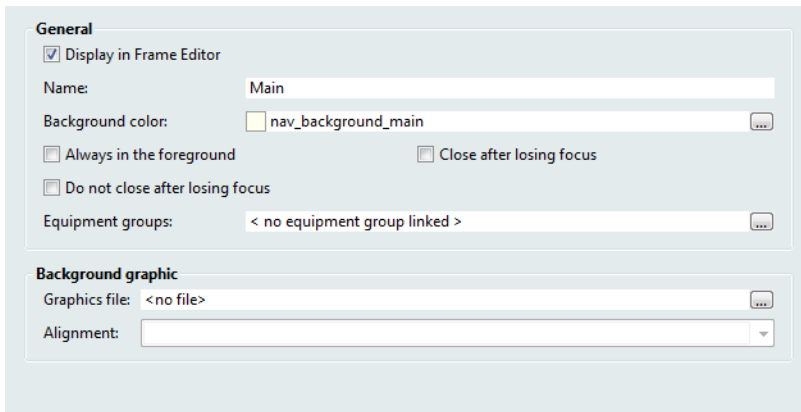
> 💡 **Information**
>
> *If there is no frame when creating a screen, then zenon automatically creates a default frame that fills the whole screen in runtime.*

Content template

The parameters for these are set in the following step once the first template has been created. These settings are - as usual - set up in the properties window. The size of the frame is set to the current screen resolution by default.

In this example, the template that has just been created is to contain the content area. This should be displayed over the whole area of the screen.

▶ Change the properties of the frame as shown in the illustration above.

Navigation template(s)

Create an additional template for the navigation area. In this example, a navigation bar with button control is to be configured in the lower corner of the screen.

Carry out the following steps for this positioning:

▶ Open the node `Screens` in the Project Manager and select the entry `Frames`.

▶ With `Frame new` in the context menu you create a new frame.

▶ Name the new template *Navigation* and assign it a color from the color palette as a background color, for example *nav_background_navi*.

Positioning a frame

There are two possibilities to position a frame:

SETTING THE PARAMETERS USING THE PROPERTIES WINDOW:

To do this, enter the following in the properties window in the **Position** group:
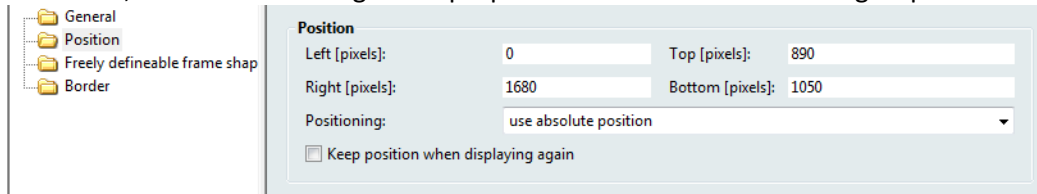
SETTING THE PARAMETERS USING THE FRAME EDITOR:

The frame editor offers the possibility to set up the sizes of frames graphically.

▶ Highlight the 'Navigation' frame in the detail view of the frames

▶ Open the context menu of the `Open frame editor` entry with the right mouse button.
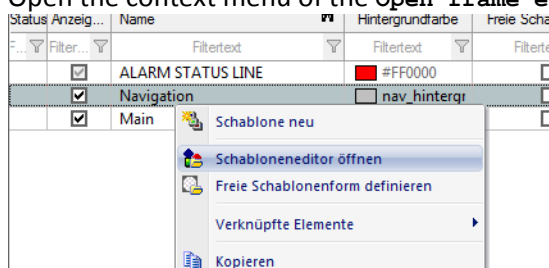
▶ The main window opens

▶ You can select a frame by clicking the mouse in this main window.

▶ The size can be changed with the mouse by moving the anchor points.
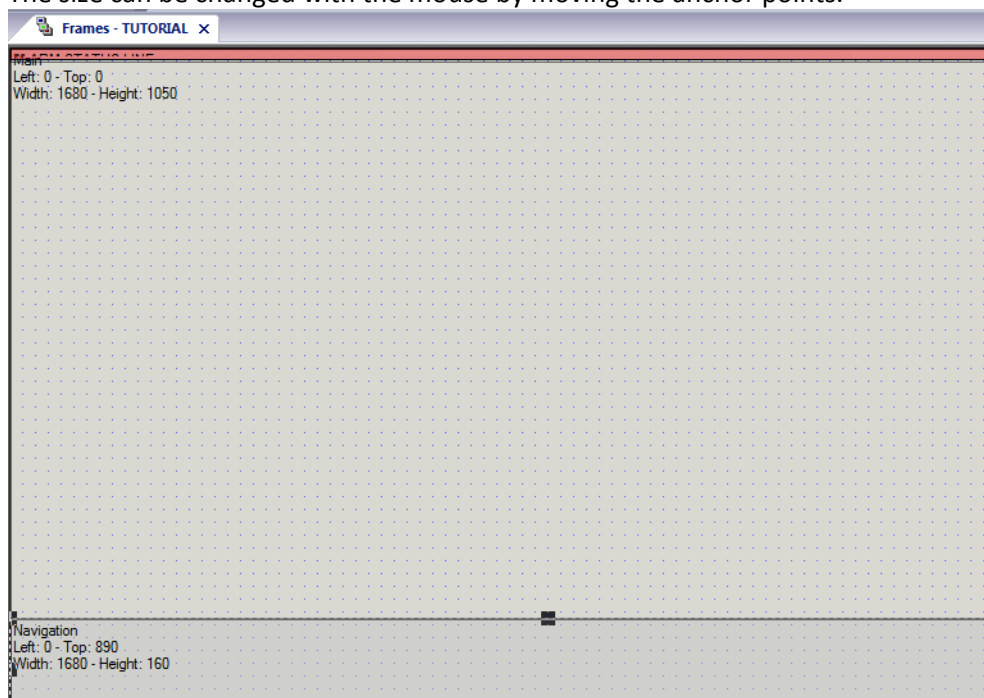
> **💡 Information**
>
> *Hide the other frames for a better overview in the frame editor. To do this, deactivate the checkbox in the 'Display' column in the detail list.*

Configuration of the second frame for subnavigation

Now configure a second frame for subnavigation:

- ▶ In the detail list of the frames, use the `Copy` context menu to copy the existing `Navigation` frame.
- ▶ Create a copy using the context menu and `Insert`.
- ▶ In the properties window, in the **General** group, rename the copy of the `Navigation frame` to *Sub_Navigation*.

10.3 Screens

Screens are central elements of a project. They display the configured equipment, inform, and provide user elements.

- ▶ A screen is a window with special predefined properties.
- ▶ Screens consist of vector elements and/or dynamic elements.
- ▶ These are linked with variables or functions.
- ▶ Screens are configured in the main window of the Editor.
- ▶ Each screen must be assigned a frame.

  > **💡 Information**
  >
  > You can also create a screen in a new project without having created a frame before.
  >
  > *In this case however, zenon automatically creates a frame with the default settings in the background.*

10.3.1 New screen

To configure a new screen, carry out the following steps:

▶ In the project manager, highlight the node `screens`.

▶ With `New screen` in the context menu you create a new screen.
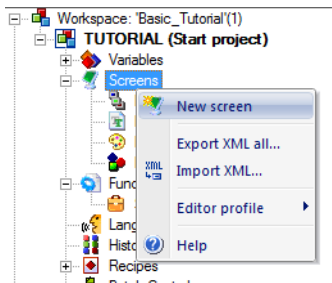
A screen with the name "*Screen 0*" is automatically created. You can change the properties of the screen in the property window.

▶ In the **General** group, enter the name *Complete overview* screen name and ensure that the screen is a `standard` screen type.

▶ In the **Frame** section, ensure that the screen is assigned to the **Main** frame.
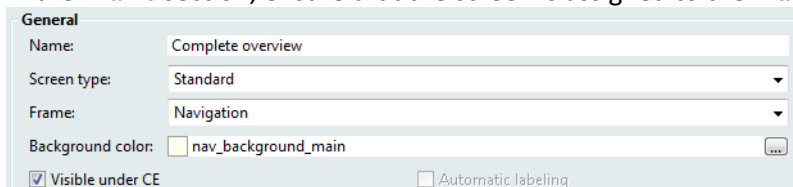
▶ Create a screen named "*Navigation bar*". Assign the **Navigation** frame to this screen.

> 💡 **Information**
>
> *The background color is automatically referenced by the assigned frame.*

10.3.2 Configuring the start screen

The following settings are necessary so that Runtime knows which screen is to be loaded when starting up:

▶ Select your project, `TUTORIAL`, in the project manager.

▶ Select the 'Graphic design' group in the properties window

▶ In the **Runtime general** group there, in **Main templates:**, click on the `Click here-> ...` button.

▶ Select, in the **Templates** window, `Frame_Main`.
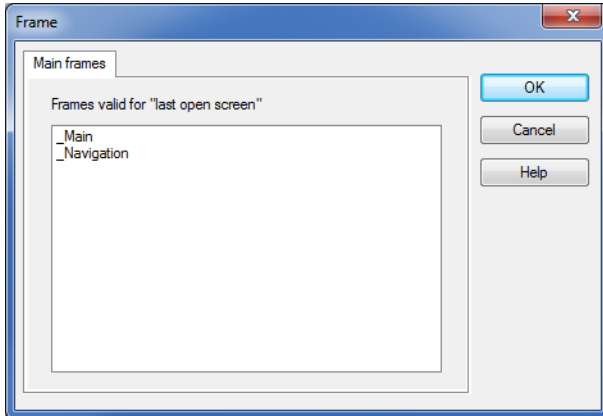
💡 **Information**

The first screen configured is automatically entered in the project properties under the `Start screen` *property in the* `Graphic design` *group. This screen will be opened automatically when the Runtime is started.*

10.3.3 Configuration of tutorial screens

Now configure other screens that will be filled with content during the course of this tutorial. Ensure that 'Main' (already configured) is always given as a frame:

▶ Line overview

▶ Storage tank 1

Note: An image is intentionally not created for the planned second storage tank. This second tank is substituted in a subsequent chapter.

10.3.4 Practical tips for screens

▶ Configure an empty standard screen for tests. Give this a name, such as *Test* or *Temporary*. In this screen, you can try out configurations, test partial configurations place symbols for editing ...

▶ use unique screen names

▶ If you are using global projects or several projects in one workspace, put a short description in front: for example water_storage_tank, motor_fuel_tank

11. Screen elements

Generally speaking, there are two different types of elements that can be used in a screen.

Vector elements	The appearance of these elements in the Runtime always stays the same.
Dynamic elements	These elements change their appearance in the Runtime (usually depending on the value of a varia

11.1 Vector elements

Vector elements, in contrast to dynamic elements, which we will be covered later in this tutorial, do not change their appearance in Runtime. They are used for graphical background information.

11.1.1 Static text

Start by placing some text in your start screen. Ensure that you have displayed the elements window.
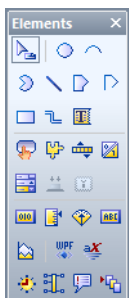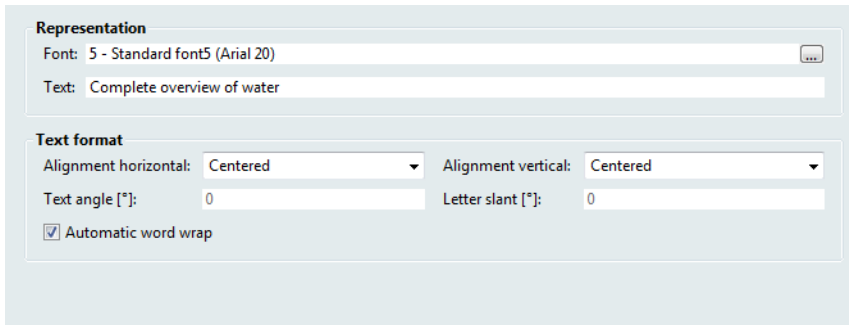
If this element is hidden, you can show this with *Options => Toolbars => Elements* .

► Change to the 'Screens' node in the project manager.

► You open the screen in the main window by double clicking on `Complete overview` in the detail view

► Select the **static text** element.

► In the complete overview screen, press the left mouse button on the desired location and hold it down while pulling up a square.

In the property window, the properties of the elements can now be amended. If the properties of an element are subsequently changed, you select the corresponding element in the screen by clicking on it with the left mouse button. The properties are then available in the properties window again.

This procedure is the same for all elements.

Representation

Font: 5 - Standard font5 (Arial 20)

Text: Complete overview of water

Text format

Alignment horizontal: Centered   Alignment vertical: Centered

Text angle [°]: 0   Letter slant [°]: 0

☑ Automatic word wrap

- ▶ Enter *Complete overview of water* in the 'Display/text' field.

- ▶ Assign your text the font **5 - Standardfont5**.
  To select the font, click on . . . and select the corresponding font style from the font list that opens.

- ▶ Place the text frame at the top left in your **Complete overview** screen.

11.1.2 Font lists

Just as with, for example, colors, data types or variables, format templates are also administered centrally for texts.

To configure font lists, proceed as follows:

- ▶ In the project manager, open the node`screens`.

- ▶ Open Activate the context menu of the `Font lists` entry with the right mouse button.

▶ Select the **New font** menu entry.

▶ Select **New font** in the detail view of the project manager,
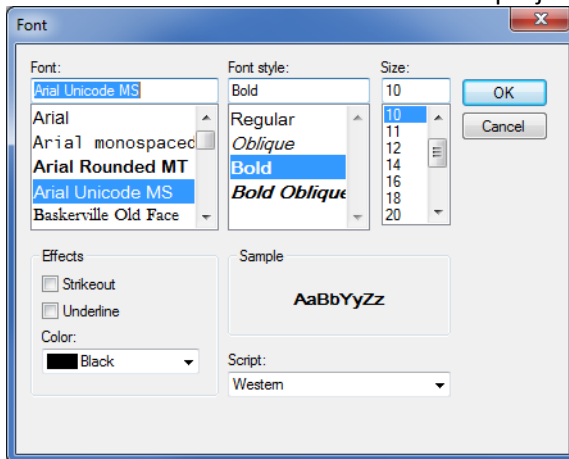
▶ Enter, as a **description**, *Buttontitle-black* in the properties window.
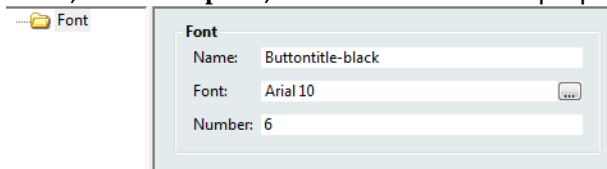
▶ Create a new font named *Buttontitle-white*.

- Select the same settings as for *Buttontitle-black* with the exception of the color. Select the color `white` for this font.

11.1.3 Tutorial configuration example

The next step is to place a heading for a storage tank.

▶ Change to the 'Screens' node in the project manager.

▶ Double clicking on `Storage tank 1` in the detail view opens the screen in the main window

▶ Select the **static text** element.

▶ In the complete overview screen, press the left mouse button on the desired location and hold it down while pulling up a square. Place this on the upper edge of the screen, in the center horizontally.

▶ Now write *Storage tank* in the **Display/text** properties field.

▶ Assign your text the font **5 - Standardfont5**.
To select the font, click on `...` and select the corresponding font style from the font list that opens.

CHANGING A FONT

In order to now amend the appearance of your headings, make the corresponding changes in the font list. This means you only have to change the font style once.

▶ In the project manager, switch to the `Font lists` in the **Screens** node.

▶ Select the `Standardfont5` font.

▶ Select, in the properties window, the `...` button in the `Font type:` properties field.

▶ The font type dialog opens.

▶ Change your font there:

• Font style: Black

• Font size: 36
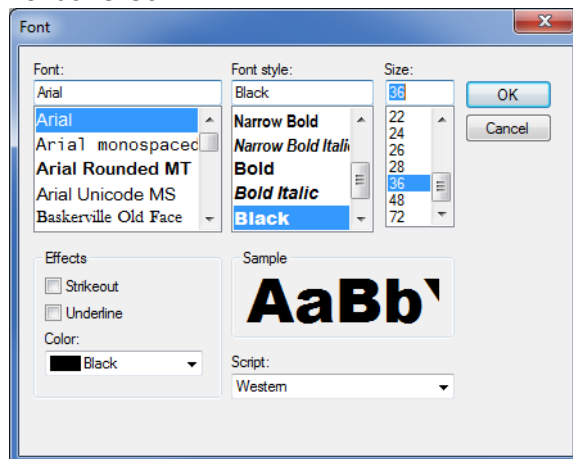
Note that the appearance of the heading in your 'Storage tank 1' screen and 'Complete overview' screens changes.

11.2   Dynamic elements

As opposed to vector elements, dynamic elements change their appearance in the Runtime.

Dynamic elements are used to display variable values or to execute functions.

11.2.1   Clock

The first dynamic element you will place is a clock display in the navigation bar.

Proceed in the following way:

▶ Open the `Navigation bar` screen.

▶ Select the **Clock** entry in the **Elements** tool bar.

▶ With the left mouse button pressed down you now drag the dynamic element to the navigation bar (approximately 40 pixels high and 200 pixels wide). Place your clock display at the right corner of your navigation bar screen.

▶ Now switch to the property window and set the parameters for your clock element:

Property group	Property field	Description
Representation	Font:	6 - Buttontitle-black
	Two-lined display	Inactive
Colors	Transparent	active
Runtime	Tooltip:	System time

11.2.2 Button

First you will create buttons for the navigation bar, with which

▶ Runtime can be ended

▶ The current project configuration will be loaded in Runtime.

Proceed in the following way:

▶ Open the `Navigation bar` screen.

▶ Select the **Button** entry in the **Elements** tool bar.

▶ Holding down the left mouse button, drag the dynamic element to the *Navigation bar* screen (150 pixels wide and 50 pixels high).
Position this button on the right edge of your navigation bar, to the left of your (already-configured) clock.

The dialog box for **selecting functions** opens automatically after you release the mouse button.
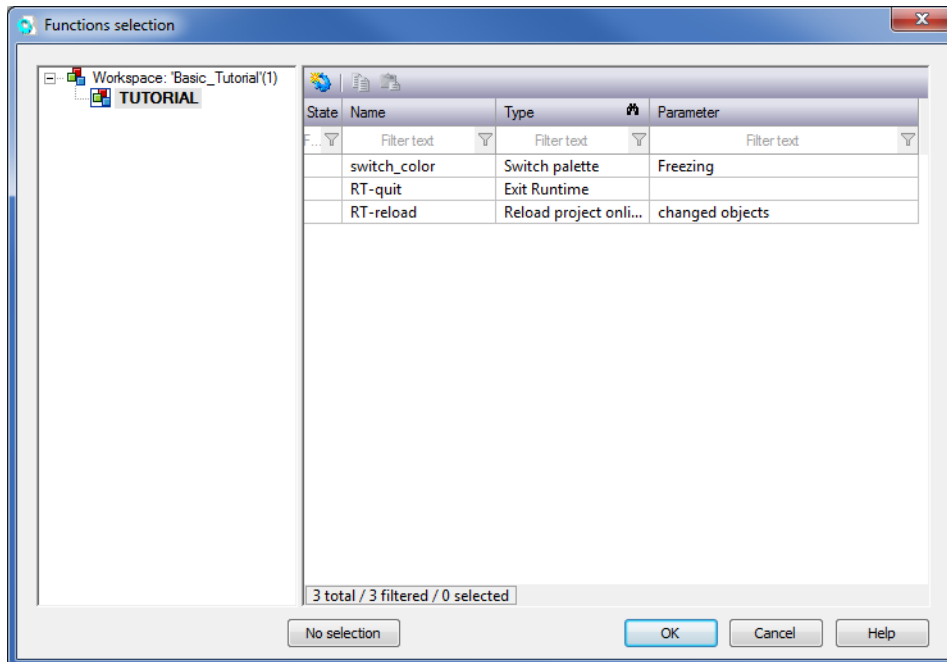
▶ Select the (already-configured). *RT-quit* function.

Setting the parameters of buttons

Now adjust your buttons once you have created them:

▶ Select the button.

▶ Open the section **Representation** in the properties window.

▶ For line 1, enter the text *End Runtime* and the text *Alt+Q* for line 2.

▶ Select **6 - Buttontitle-black** as the font, as already configured.

▶ Open the **Runtime** group in the properties window.

▶ In the `Keyboard operation` properties field, click on the `...` button .

The dialog for the definition of the key combination opens.
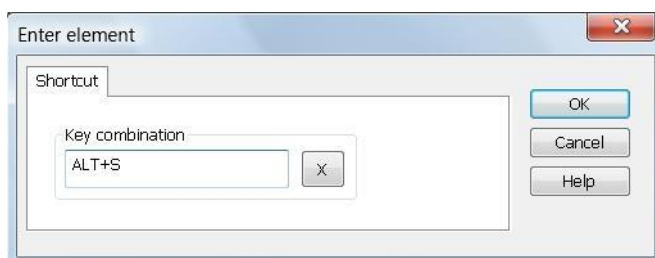
In the Runtime, you can now execute the function either by clicking the button or by pressing the key combination.

▸ Enter *Alt+Q* by clicking into the input area and pressing that key combination.

▸ Confirm the settings with oκ.

> 📄 **Example**
>
> Create, under the first button, another button for reloading the configuration with the label *Reload*and *Alt+R*, as well as the corresponding keyboard shortcut. Use RT-reload as the function.

> 💡 **Info**
>
> *We recommend to define a separate font for the text of the button and to use this font for all buttons in the project. If the font or the size is changed at a later point in time, all other buttons are amended automatically.*

Applying functions to buttons

After drawing a button, the **function selection** window is called up automatically. In this window, you can immediately assign your button a function when drawing it.

If you have not yet created a corresponding function, you can do this by calling up the context menu (*right mouse button*) `New function...` Subsequent action directly in the function selection window.
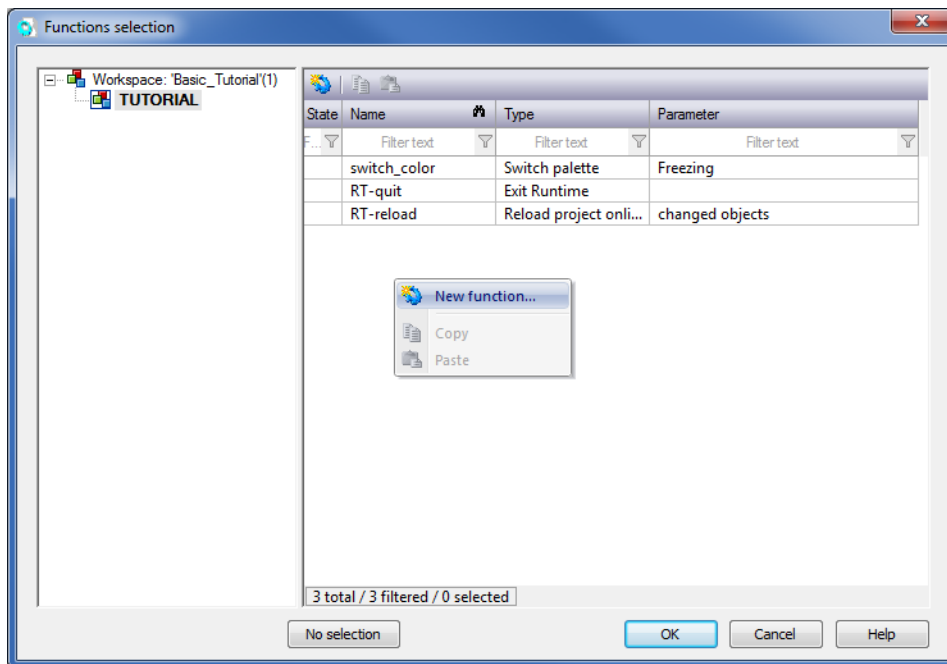
📑 **Example**

The button is not assigned a function (yet) by selecting the `Cancel` button. You can assign this function later either by *dragging and dropping* or by selecting the button and setting the parameters of the **Variable / Function** properties group at a subsequent point in time.

Configuring other functions

Configure the following screen switch functions for this tutorial:

Function name	Target screen (already configured)
go_to Storage tank 1	`Storage tank 1`
go_to Line overview	`Line overview`
go_to Complete overview	`Complete overview`

Note: The planned *Storage tank 2* screen has intentionally not yet been given a screen switch function.

Function via drag + drop

Another possibility to apply functions to buttons is to simply drag an existing function to a button. In the following steps, do not worry about the layout or exact size of the buttons.

Example:

▶ Draw a button with the labels "*Complete overview*" "*Alt + 0*".
In the **function selection** dialog that opens, select (the already-configured function) *go_to_complete_overview*.

▶ In the **function selection** dialog that opens, select `Cancel`.

▶ Hold down the mouse button and drag this function to the *Detail view of motor* button.

▶ Create a new button

▶ Label this button with *Detail view of storage tank 1* (Alt + 1)

▶ Drag and drop the (previously-configured) function *go_to storage tank 1* to the button that has just been created.

▶ Repeat these configuration steps again for your line overview. Labeling: "*Line overview*", "*Alt + 3*".

▶ Create a *Storage tank 2* button with   *Alt+2* as the keyboard shortcut and `Text line 2:`. This button is not assigned a function at this time.

Note: The assigned function becomes evident if you move the mouse over the corresponding button.
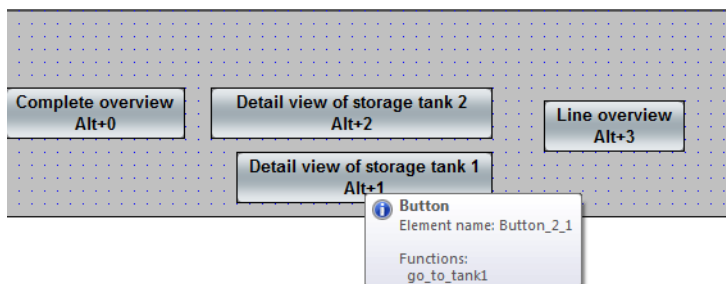
Further button settings

In addition to the labeling and allocation of a shortcut you can make the following settings for your buttons:

▶ **Tool tip**
Text appearing when the mouse is on the button.

▶ **Text when held:**
Text appearing when the button is held.

Configuring the tooltip

Carry out the following steps to configure the tooltip:

▶ Select the *Complete overview* button

▶ Switch to the **Runtime** group in the property window

▶ In the **Tooltip:** property, enter *Complete overview*.

▶ Repeat this setting for all buttons:
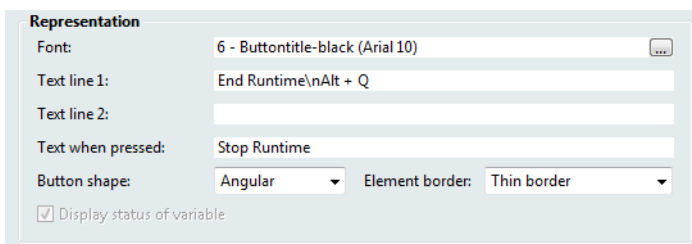
Button name	Tool tip
Reload	Load current configuration in Runtime
Stopping the Runtime	Ends Runtime
Detail view, motor	Detail view, motor
Storage tank 1	Detail view Storage tank 1
Storage tank 2	Detail view Storage tank 2
Line overview	Detail view inflow pipe

Text when pressed: with two-line button labeling

If the **Text line 2:** properties field is assigned for a button, **Text when pressed:** cannot be configured.

In order to combine two-line button labeling with **Text when pressed:**, carry out the following steps:

▶ Select the *End Runtime* button in the current project configuration.

▶ Switch to the **Display** group in the properties window.

▶ Delete the previous entry in **Text line 2:**.

▶ Add to the text in Text line 1:. \nAlt + Q
The following is then shown in the properties field: *End Runtime\nAlt + Q*

▶ Enter the following under 'Text when pressed': *Stop Runtime*

Note: A line break can be forced by entering the character string \n.
The prefixed & sign underlines the subsequent letter *Q*.

> ### 🔆 Information
>
> *The Text line 2 property must be empty to use \n.*

Orientating the buttons

After you have now carried out the previous step with the buttons,, you will no adjust the size and orientation of the buttons. zenon provides its own tool bar with the corresponding commands for this.

If you cannot see the menu bar orientation, display it the following way:

Select, in the menu bar, *Options => Tool bars=> Orientation*

The orientation tool bar appears:

ORIENTATION OF THE BUTTONS:

Now switch back to the `Navigation bar` screen.

- ▶ Set the orientation, size and color of any desired button
- ▶ Highlight the remaining buttons with `shift + mouse click`
- ▶ Now select, from the orientation tool bar:
  - Same size
  - Align up

Note: The requirements for size and orientation are always provided by the first-selected and highlighted button. It is therefore advisable to click the button with the longest text entry in order to set this as a reference.

11.2.3 Screen Functions

You can link functions to each screen, which will automatically be executed on opening or closing the screen. You will now use this functionality to open the button-bar with the start screen.

- ▶ In the Project Manager, select the `Screens` node.
- ▶ In the detail view of the Project Manager, highlight the `Complete overview` screen.
- ▶ Open the section `Execution` in the properties window.

▶ For start function, select the screen switch function *go_to_Navigation* from the list.

This means that in Runtime, whenever the *Complete overview* screen is opened, the `Navigation` will also be opened automatically.

> 💡 **Information**
>
> Another possibility is the configuration of start scripts.

11.2.4 Other screen elements

You will become familiar with the other screen elements for the creation of your **Storage tank1** screen with examples.

▶ In the project manager, highlight the node`screens`.

▶ Open your `Storage tank 1` screen by double clicking it.

Tips for editing screens

In addition to the settings of your Editor, you can also display the main window in full screen mode for drawing and configuration.

Proceed in the following way:

▶ Open, in the main window in the context menu (`right mouse click`) the **Full screen mode** entry.

▶ Your main window is displayed as a full screen.

To return to your editor view, select close full screen from the **Full screen** dialog window. This is automatically offered in full screen mode.
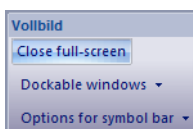
Context menu of the main window

When configuring graphics, if there is a screen open in the main window, there is a context menu (`mouse + right click`) available with the most important commands.

The display of the context menu depends on whether an element is selected or not. If no element is selected, only commands that do not relate to operations with elements are shown.

Example: Context menu for selected format
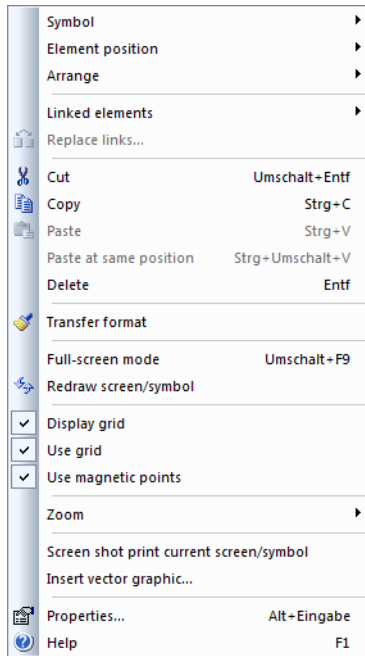
SHIFT KEY:

If the size of the graphics is changed using the corner points, the proportion of the screen is retained if the `shift` key is held down.

USE GRID

The grid makes it possible to align elements precisely on the screen. You can find the basic settings for this in the `Options -> Settings` menu.
The `Display grid` context menu entry shows the grid in the main window if it is activated. You thus do not see a grid if it is not activated.

Example illustration of the main window with 'Display grid' deactivated:
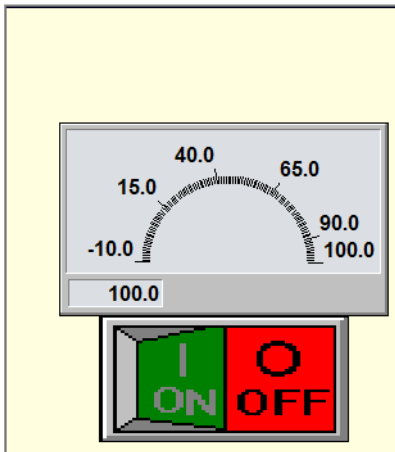
USE MAGNETIC POINTS

Symbols are automatically laid out on the grid if this is activated. Deactivate this option for precise positioning.

Printout of a screen configuration

To print a screenshot of your screen configuration, proceed as follows:

▶ Call up the desired screen.

▶ Select, in the main menu, via the context menu, `Print screenshot of current screen.`

SCREENSHOT OF THE CURRENT SCREEN DOES NOT REACT - HANDLING THE PROBLEM

If the above-described command does not result in a printout, it is probably because you have not assigned a printer yet.

To assign a printer, proceed as follows:

▶ Select, in the `File` menu, the `General configuration` -> `Standard...` entry
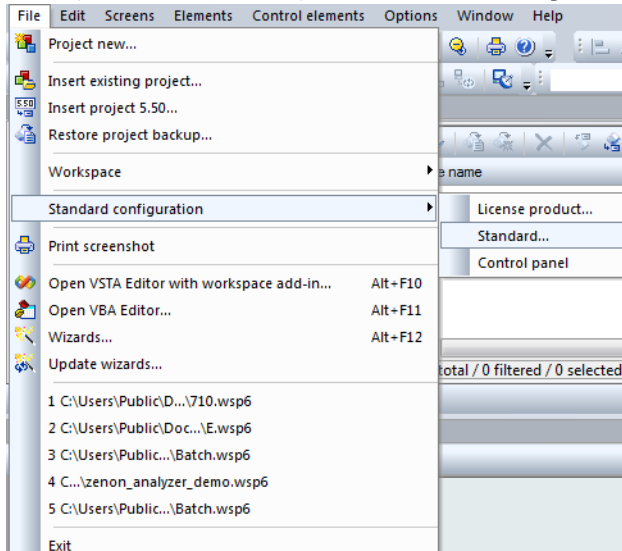
▶ In the `standard settings` dialog that opens, you can now assign your system printer the different functionalities. The `for screenshots:` dialog is relevant to the printout of the screen projection.
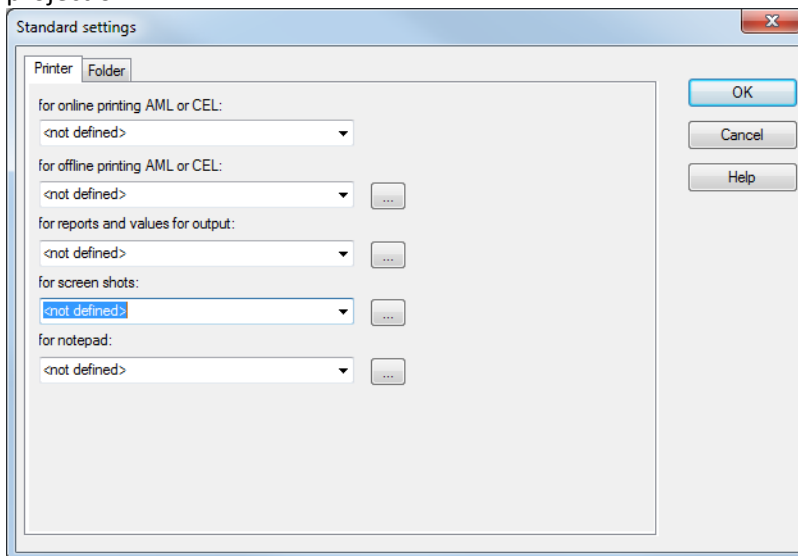
11.2.5   Numerical value

Numeric values display numerical values of a variable.

In this tutorial, the numeric value screen element is used to display the following values:

▶ Temperature in the tank

▶ Fill level of a tank (variable not yet configured)

Changing an existing structure data type

No variable has been created for the fill level with the previous project configuration (intentionally).

To change a structure type and add the fill level variable, carry out the following steps:

▶ In the project manager, highlight the `Variables` node.

▶ Switch to the `Data types`.

▶ Highlight the existing data type, *Tank*, by clicking on [+].

▶ Use `New structure element...` to add a new structure element
Note: You can only use `New structure element...` if the *Tank* structure data type has been selected.

▶ Give it the name *Fill level* with the embedded data type `UINT`.

▶ Switch back to the view of the variables.
Select the `Variables` node in the project manager.

ACTIVATION OF SUBSEQUENTLY-ADDED FILL LEVEL VARIABLE

Once a structure element has been added to an existing structure data type, this is still inactive in the detail list of the variables immediately afterwards and is grayed out.

Name	Mea...	Driver	Offset	Bit num...	Data type	Start offset
Filter text	Fil...	Filter text	Filter text	Filter...	Filter text	Filter text
Temperature sensor	°C	S7TCP32 - S7 TCP-IP	0	0	Temperature	0
Water pump		S7TCP32 - S7 TCP-IP	0	0	Motor	0
Water pump.Fuel tank		S7TCP32 - S7 TCP-IP	0	0	USINT/<e...	0
Water pump.Speed		S7TCP32 - S7 TCP-IP	1	0	UDINT/<e...	0
Water pump.Motor temperature	°C	S7TCP32 - S7 TCP-IP	5	0	Temperatu...	0
Heating line		S7TCP32 - S7 TCP-IP	0	0	Heating	0
Heating line[1].Heating switch		S7TCP32 - S7 TCP-IP	0	0	BOOL/<e...	0
Heating line[2].Heating switch		S7TCP32 - S7 TCP-IP	1	0	BOOL/<e...	0
Heating line[3].Heating switch		S7TCP32 - S7 TCP-IP	2	0	BOOL/<e...	0
Heating line[4].Heating switch		S7TCP32 - S7 TCP-IP	3	0	BOOL/<e...	0
Storage tank		S7TCP32 - S7 TCP-IP	0	0	Tank	100
Storage tank[1].Inflow - switch		S7TCP32 - S7 TCP-IP	100	0	BOOL/<e...	0
Storage tank[1].Outflow - switch		S7TCP32 - S7 TCP-IP	100	1	BOOL/<e...	0
Storage tank[1].Heating element						
Storage tank[1].Heating element.Heating swi...		S7TCP32 - S7 TCP-IP	101	0	BOOL/<e...	0
Storage tank[1].Fill level						
Storage tank[2].Inflow - switch		S7TCP32 - S7 TCP-IP	104	0	BOOL/<e...	0
Storage tank[2].Outflow - switch		S7TCP32 - S7 TCP-IP	104	1	BOOL/<e...	0
Storage tank[2].Heating element						
Storage tank[2].Heating element.Heating swi...		S7TCP32 - S7 TCP-IP	105	0	BOOL/<e...	0
Storage tank[2].Fill level						

Carry out the following steps to activate it:

▶ Select, in the variable view, the *Storage tank[1].Fill level* entry.

▶ Select **Activate** in the context menu.
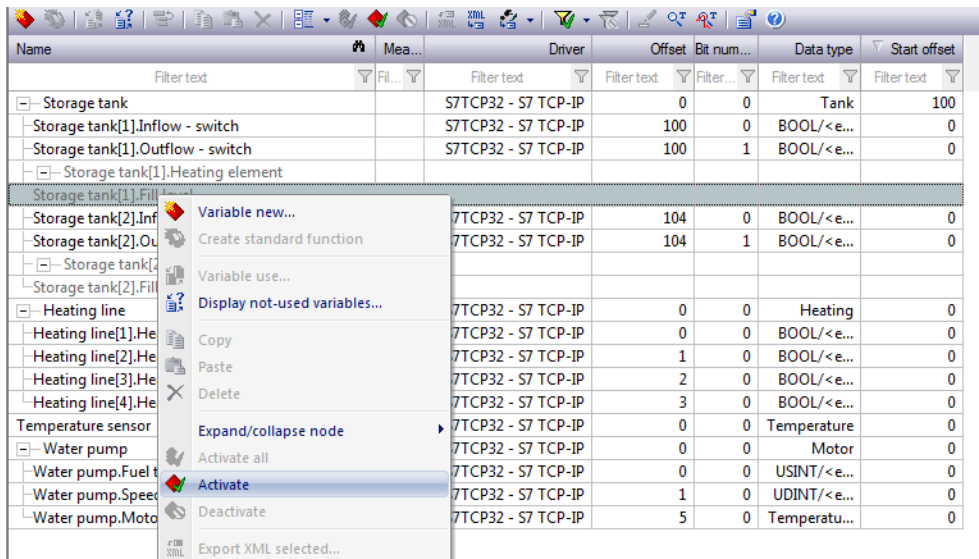
▶ The variable was activated and is no longer displayed in a light gray font

▶ repeat this activation process for *Storage tank[2].Fill level*.

> 💡 **Information**
>
> *Variables, data types and structure types can be expanded afterwards. The reconfiguration has to be activated.*

Other project configurations

Once you   have added the missing variable by means of the structure data type, set the parameters for this.

Carry out the following steps to configure the *Fill level* variable:

▶ In the project manager, highlight the **Variables** node.

▶ Switch to the **Data types**.

▶ Add to the existing data type, *Tank*, by clicking on [+].

▶ Highlight the *Tank* structure element.

▶ Enter, in the property window in the **General** group, under **Unit of measurement:** *Liter*.

▶ You configure the **Write set value** property group with *0* **Min. set value** and *20000* **Max. set value**.

▶ Create two new limit values in the property window:

- Limit text: Fill level low
  Limit value: 1000
  Minimum/Maximum: Minimum
  In alarm message list: Active
  In chronological event list: Active
  Limit value color: Fill_level_LOW (already configured)

- Limit text: Fill level high
  Limit value: 19000
  Minimum/Maximum: Maximum
  In alarm message list: Active
  In chronological event list: Active
  Limit value color: Fill_level_HIGH (already configured)

Configuring the element numerical value

▶ In order to integrate numerical values into your screen storage tank proceed as follows:

▶ Open the screen `Storage tank`.

▶ In the menu or rather the element symbol bar select the entry**Numerical value**.

▶ Hold the left mouse button down and create a frame for your numerical value in the screen *Storage tank 1* .

▶ The dialog variable selection opens:
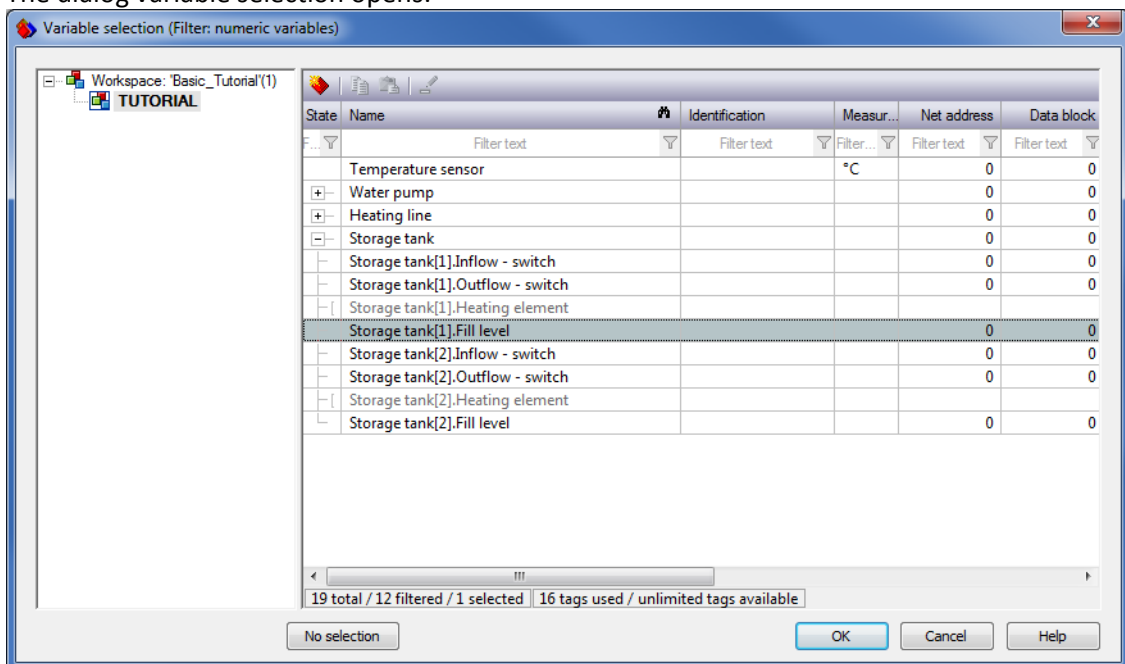
▶ Select `Storage tank[1].Filling level`.

▶ Repeat the configuration of a numerical value with the variable **Storage tank[1].Heating element.Temperature sensor**.
Note: If you cannot see the variable[1].Heating element.Temperature sensor extend the node Storage tank[1].Heating element by once clicking on the -[symbol in the column Status. Attention: click once only!

▶ In addition, select `4 - Standard font4` in the display properties **Font**of your numerical values. To do so, select both of your newly configurated numerical value elements and change into the properties group Display.

LABELING

▶ Place both numerical elements with a little distance below the headline.

▶ Label your numerical elements:

- From the elements symbol bar select `static text`.

- Place this element on the left next to the numerical element of your filling level.

- Enter the text *Filling level*

- Repeat these steps with the text *Water temperature*.

VIEW OF THE PRESENT CONFIGURATIONS

Tips for the numeric value element

▶ To change the size of an element on the screen, click on one of the corner points (the cursor changes to a double arrow), hold down the left mouse button and move the mouse to the desired size.

▶ To move a dynamic element, click on the middle of the element (an arrow cross appears), hold down the left mouse button and drag the element to its new position.

▶ Click on the icon Save screen or select the entry `save screen` from the context menu of the screen.

THE EASIEST WAY TO CREATE SEVERAL COUNTER VALUES ON A SCREEN:

Highlight the element and copy it to the clipboard with `Ctrl+C`*, then paste it with* `Ctrl+V`*. Now you can drag the desired variables on the elements with the mouse.*

ORIENTATION OF YOUR PROJECT CONFIGURATION

The functionality of the 'Orientation' tool bar is naturally also available for the numeric value element to the full extent. use this to amend the size and orientation of your project configuration.

LABELING OF THE NUMERIC VALUES

To enable quick orientation, provide your numeric values with a key. To do this, simply use the 'static text' screen element.

Alternatively, you can also give numeric values a tooltip. The parameters for this are set in the **Runtime** property group under **Tooltip:** .

USE OF FONTS FROM THE FONT LIST

It is recommended that your own font is configured in the font list for labeling and keys. This tutorial uses a configured font key with Arial, bold, 12 pt.

Activation of automatic variable selection

If the variable selection does not open automatically, perform the following actions:

▶ Open the menu **Option**.

▶ Select the command **Settings**.

The following dialog opens:
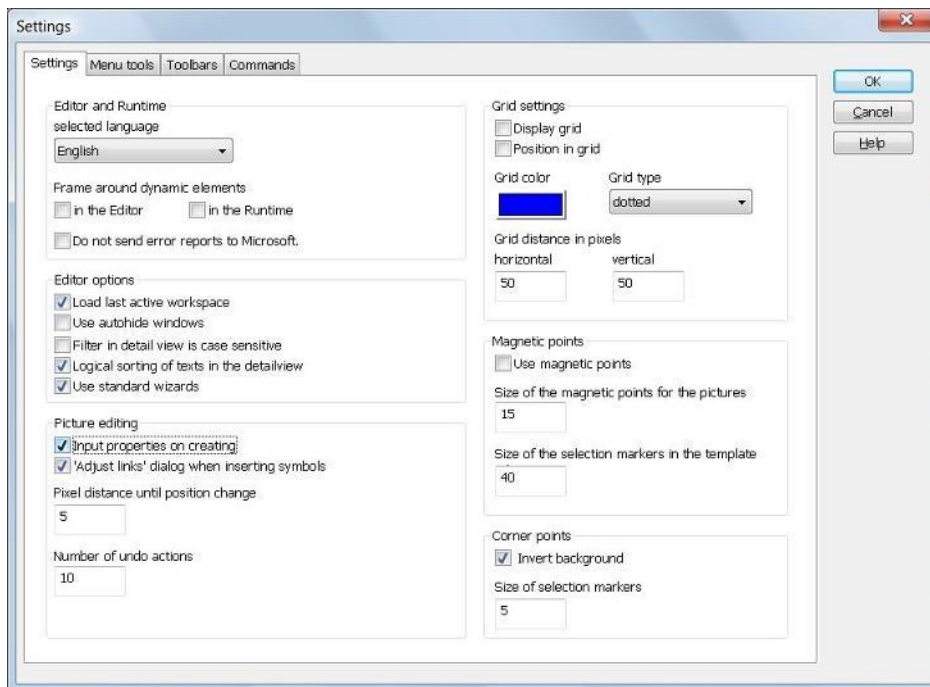
▶ Activate the **Input properties after creating objects** option in the **Element editing** group and then press the **OK** button.

> 💡 **Information**
>
> *You can also open a dialog box to select the variable by double clicking on the empty element frame.*

11.2.6 Project configuration with symbols

Make use of the global symbol library in zenon to visualize your storage tank. It contains graphic symbols classified according to topic.

Proceed as follows to use symbols for visualization:

▶ In the project manager, open the **Global symbol library** node.

▶ Enter *tank* in the default window in the element name filter.

▶ Open the folder by clicking on [+].

▶ Select a symbol and move this by dragging & dropping in the main window to your open screen, *Storage tank 1*.

▶ Select a template from the global symbol library.

▶ Enter *valve* as the filter text and select *Throughflow-Meter2*

▶ Place this symbol on the previously-placed tank graphics by dragging & dropping.

▶ Enlarge both symbols:

View of the previous configuration of your *Storage tank 1* screen in full screen mode:
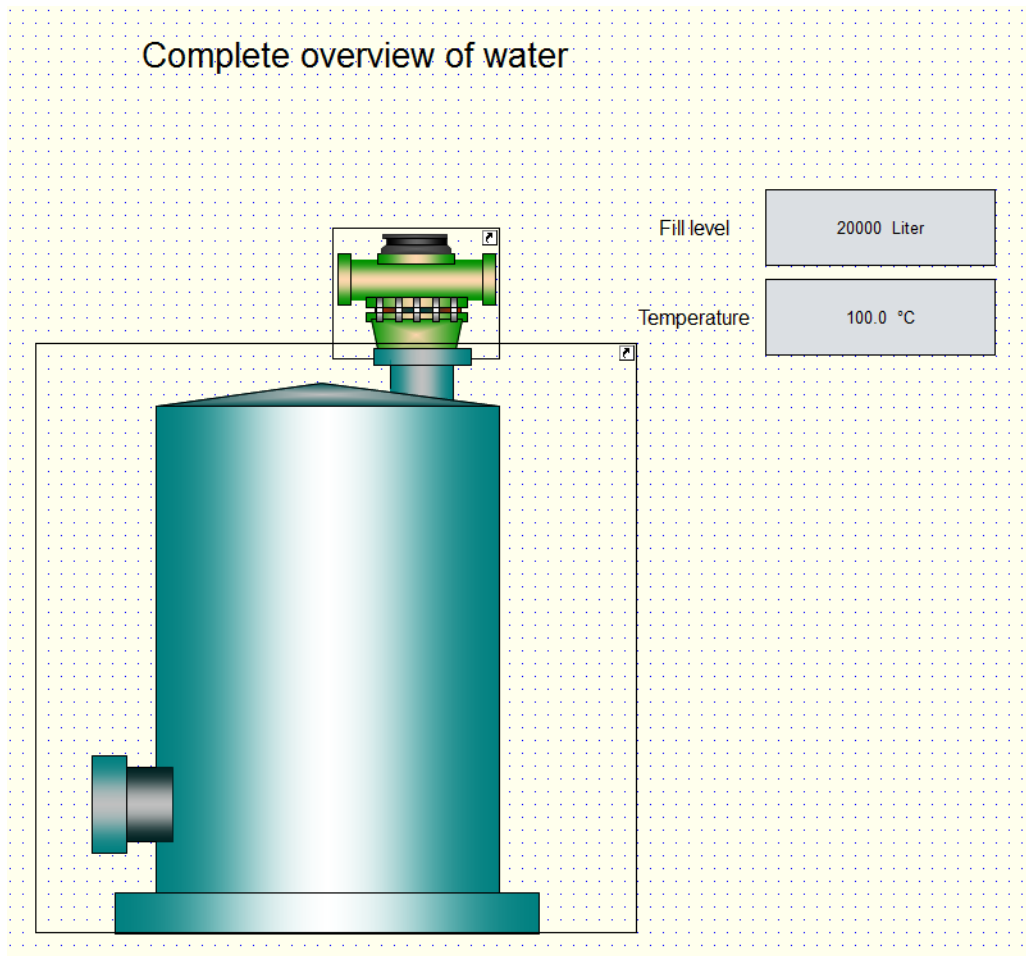
> ### ℹ Information
>
> *The symbols in your symbol library depend on your zenon Edition license.*

11.2.7 Bar display

You show the numerical size of a signal with the Bar graph dynamic element. The length of the bar changes with the signal size in runtime.

To create Bar Graph in a screen:

▶ Open the `Storage tank 1` screen.

▶ Select the `Bar Graph` command from the Elements menu.

▶ With `the left mouse button held down`, drag the bar graph element to the *Storage tank 1* screen to the water tank symbol.
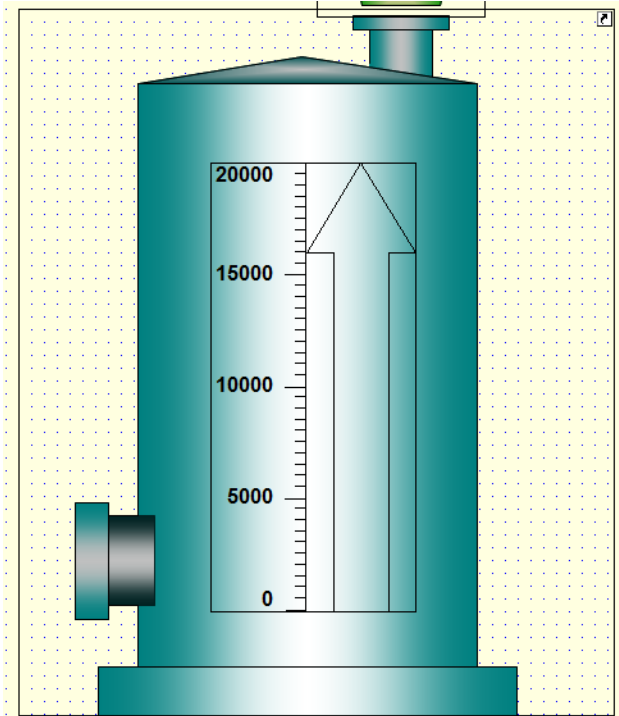
▶ In the `Variable selection` dialog, move the bar to the **Storage tank[1].Fill level** variable and then press the ᴏᴋ button.

▶ Make the following settings in the properties window:

Property group	Property field	Description
Representation	Font:	8 - keys
Colors	Transparent	active
	Bar color	FillLevel_Normal
Runtime	Tooltip:	Storage tank fill level

Color-coded display of the limit violation

At limit violations the color from the variable can color the whole area or only the area starting from the limit violation.

This functionality is controlled with the 'Explicit' property in the 'Display' group.
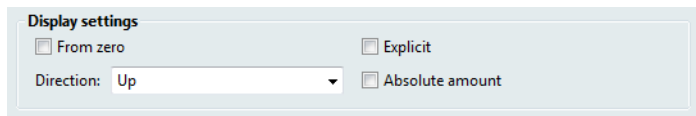
EXPLICIT ACTIVE

If a limit value of the linked variable is breached, only the part of the display that goes into the limit value breach is shown in the color of the respective limit value.
If the variable has several defined limit values, the corresponding areas of the display are each shown in the color of the limit value.

Sample view:
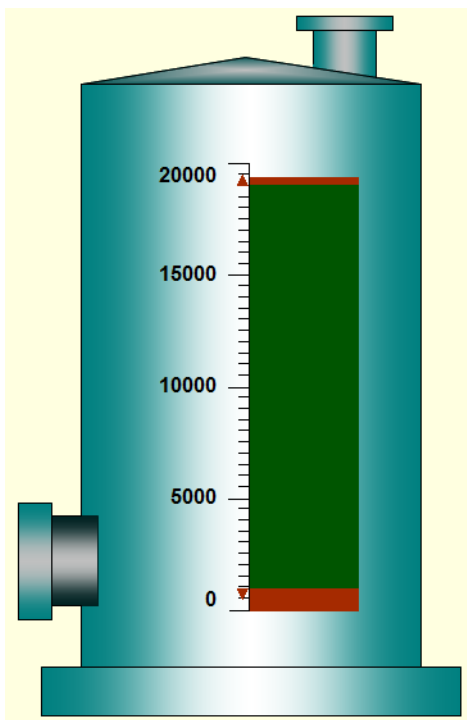
EXPLICIT INACTIVE

If a limit of the linked variable is violated, the complete display is displayed in the color of the violated limit.

Sample view:
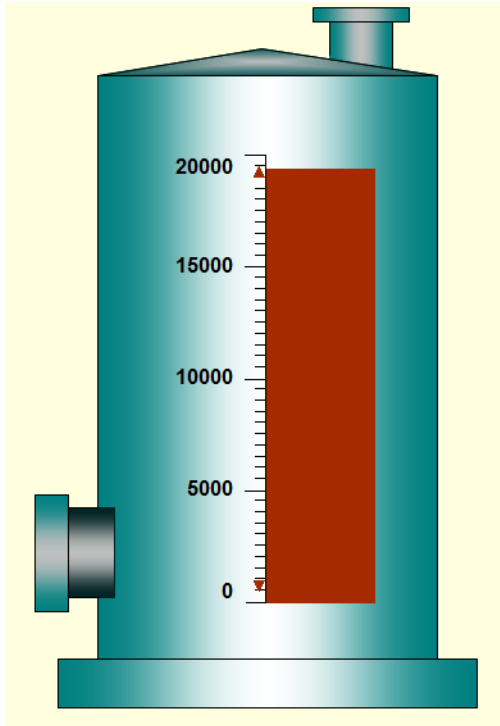
11.2.8   Pointer instrument

You display the indicating instrument with the **Indicating instrument** dynamic element.

▶   Open the `Storage tank 1` screen.

▶   Select the **Indicating instrument** entry in the menu or the tool bar.

Hold down the left mouse button and drag the frame for the indicating instrument in the *Storage tank 1* screen. Position the indicting element in your text block next to the water temperature.

▶   Select *Storage tank[1].Heating element.Temperature sensor* in the dialog that opens.

▶   Set the parameters of the indicating instrument in the property window:

Property group	Property field	Description
Representation	Font:	8 - keys
	Display measuring unit	active
	Arc of a circle	active
	Explicit	inactive
	Main ticks scale	25000
	Subsidiary ticks scale	10
Colors	Use limit color	active
Runtime	Tooltip:	Storage tank fill level

DETAILS VIEW:
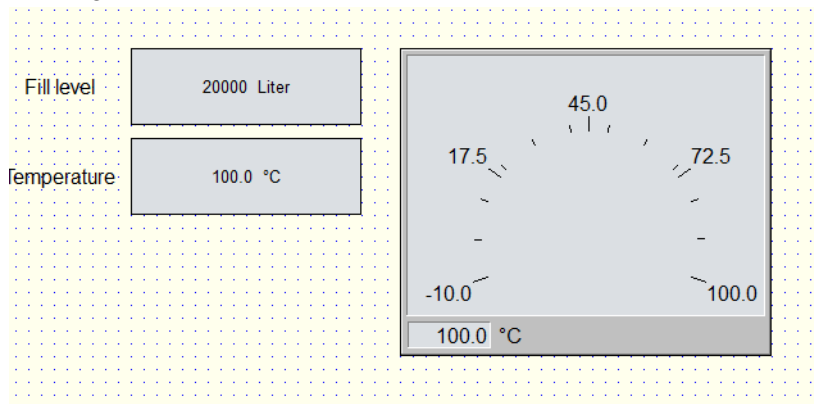
11.2.9   Universal slider

With the **Universal slider**   dynamic element, you show variables in the form of slider controllers, bar graphs, LED bar displays or any other sliders you wish.

Proceed as follows to configure a **universal slider**:

▶   Open the `Storage tank 1` screen.

▶   Select the **Universal slider** entry in the menu or the tool bar of the element.

▶   With the `left mouse button held down` in the *storage tank* screen, drag the frame for your slider.
Position the universal slider in your text block under the already-configured pointer instrument.

▶   In the dialog that has now opened, select *Storage tank[1]/.Temperature sensor*

▶   Set the **main ticks** to `25` and the **subsidiary ticks** to `10`.

▶   Make the following settings in the properties window:
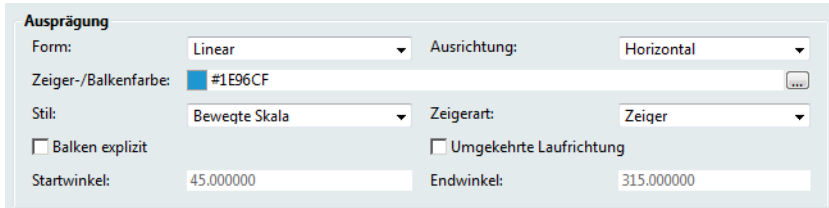
Property group	Property field	Description
Representation	Font:	8 - keys
	Main ticks:	10
	Subsidiary ticks:	10
Universal slider	Style:	Moving Scale
	Pointer type:	Pointer
	Bar explicit	active
	Display title	active
	Title text:	Water temperature
	Display drag indicator:	active
Runtime	Tooltip:	Water temperature

EXAMPLE VIEW:
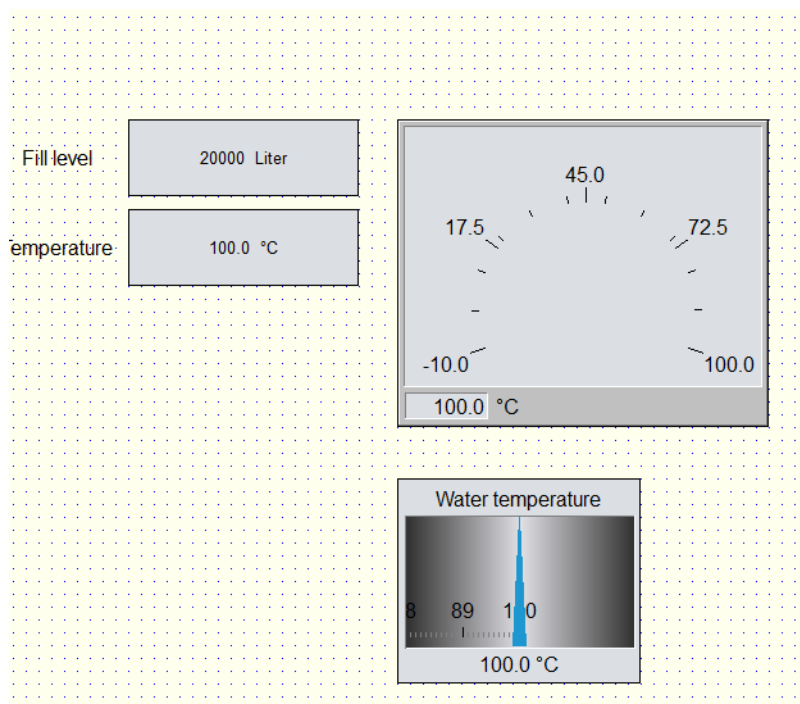
Experiment with the appearance of your universal slider.

Change the style in the **Universal slider** property group:

11.2.10 Trend element

With the **Trend Element** dynamic element, you display all values in Runtime in the form of trend curves, whereby several process variables can be displayed at the same time.

Proceed as follows to configure a **trend element**:

▶ Open the screen `Storage tank`.

▶ Select the **Trend element** entry in the menu or tool bar.

▶ With the `left mouse button held down` in the *Storage tank 1* screen, drag the frame for the trend element.
Position the trend element between the tank graphics and your text block.

▶ In the variable selection dialog that opens, select *Storage tank[1].Temperature sensor* and *Storage tank[1].Fill level*.

▶ Set the **main ticks** to `25` and the **subsidiary ticks** to `10`.

▶ Make the following settings in the properties window:

Property group	Property field	Description
Runtime	Tooltip:	Graphical overview of fill level and temperature of the last 10 minutes.
Curves	In this property group, you set the parameters for the color and appearance of your trend curves. In this example, two curves are offered, because two variables were selected. Select different colors and line types for each curve.	

EXAMPLE VIEW WITH TWO VARIABLES DISPLAYED IN THE TREND ELEMENT:

▶
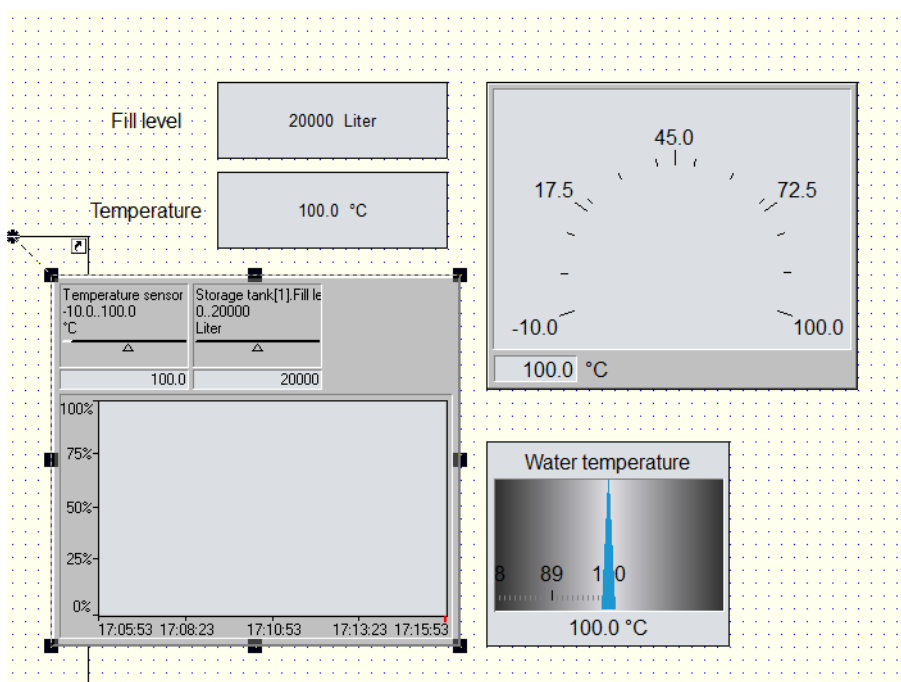
11.2.11 Dynamic Text

With the **Dynamic Text** dynamic element, you display the current limit value text in the event of the limit being breached or display the value of a string variable in alphanumeric form. If there is no limit violation, the text defined in the Editor is displayed.

Proceed as follows to configure a **dynamic text** element:

▶   Open the screen `Storage tank`.

▶   Select the **Dynamic element** entry In the menu or the tool bar.

▶   With `the left mouse button held down`, drag the template for your **dynamic text** element to the *Storage tank* screen.
    Position this element in your text block to the left of the previously-configured numeric value for the fill level.

▶   In the dialog that has now opened, select *Storage tank [1].Fill level*

▶   Make the following settings in the properties window:

Property group	Property field	Description
Representation	Font:	8-key
	Text:	No limit value breach
Colors	Transparent	active
	Text/line color	Storage tank [1].Fill level
	Fill color:	Storage tank [1].Fill level
Runtime	Tooltip:	Fill level status

Dynamic text - other configurations

Now configure a second **dynamic text** element for the temperature.

▶   Select the **Dynamic text** entry In the menu or the tool bar.

▶   With `the left mouse button held down`, drag the template for your **dynamic text** element to the *Storage tank* screen.
    Position this template in your text block to the left of the previously-configured numeric value for the temperature.

▶   In the dialog that has now opened, select *Storage tank[1]/.Temperature sensor*

▶   Make the following settings in the properties window:

Property group	Property field	Description
Representation	Font:	8-key
	Text:	No limit value breach
	Fill color:	Storage tank [1].Temperature sensor
Colors	Transparent	active
Runtime	Tooltip:	Water temperature status

EXAMPLE SHOWN WITH DYNAMIC TEXT ELEMENTS (DETAIL):
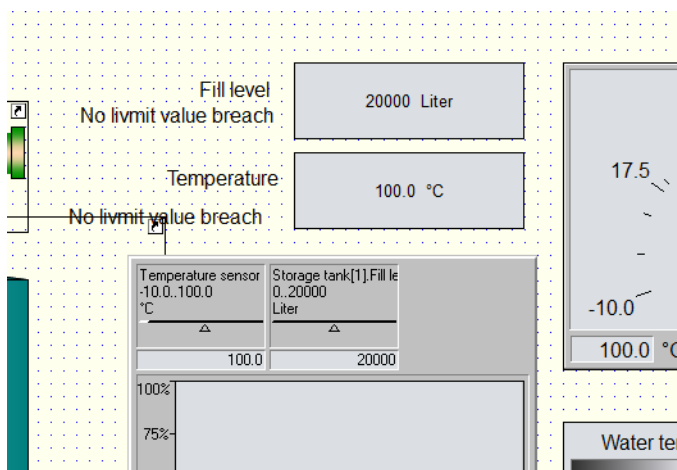
> 💡 **Information**
>
> *Note the different text colors. These are due to the two different parameters set in the text color properties field.*

11.2.12 Combined element

The combined element is a dynamic element that can be used universally and can adopt the most varied graphic characteristics.

The form of the display is defined through statuses. Variable values of the main value, values of further values and status information from all linked variables can all serve as statuses. Any desired links can be created using formulas.

Configuring a combined element

This tutorial will cover the option to switch some heating on and off. The heating should only be visualized if it has been activated. If the heating has been deactivated, no visualization should be active. The **combined element** is the solution for this.

Carry out the following steps to configure a **combined element**:

▶ Open the screen `Storage tank`.

▶ Select the **Combined element** entry in the menu or tool bar.

▶ With the `left mouse button held down` in the *storage tank* screen, drag the frame for your **combined element**.
Position this element precisely underneath the symbol of your tank.

▶ In the dialog that now opens, select *Storage tank[1].Heating element.Switch on/off*.

▶ Make the following settings in the properties window:

Property group	Property field	Description
Runtime	Tooltip:	Visualization of heating

The dialog for configuration opens once the element has been positioned.

Setting the parameters for a combined element

The dialog for configuration opens after the combined element is positioned.

you can also get to this dialog for subsequent changes via the properties window. To do this, press `Click here ->` in the **Display** properties group in the **Configuration and test:** property field
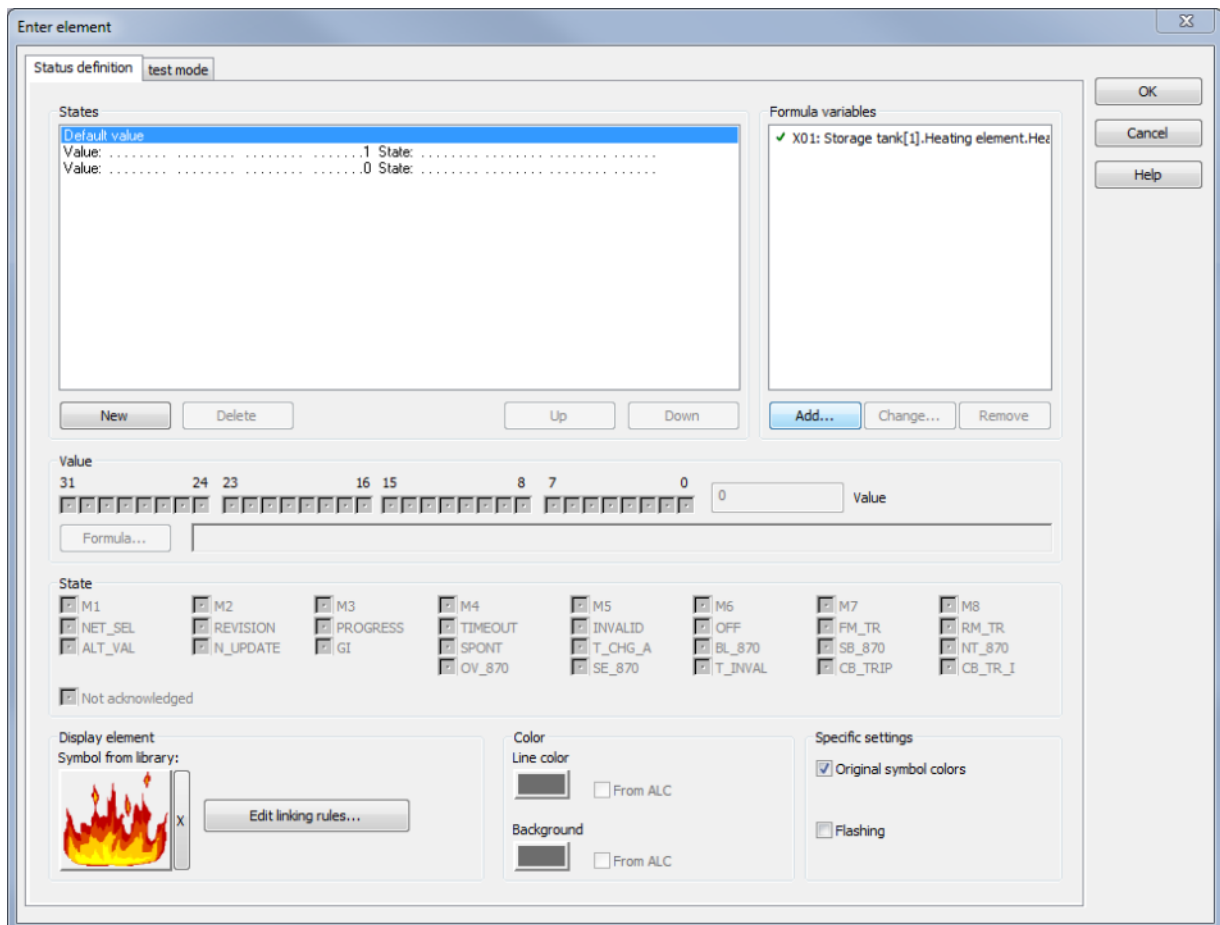
Carry out the following steps to connect values to graphics:

▶ Create a new value with the `New` button.

▶ Write the value to the value bar.

▶ Assign a display element to your value by clicking on `Empty corner` under **`Symbol from library:`** .
A display element from the global symbol library has been selected for this tutorial. You can find this fire symbol in the *International Symbols, fire* symbol group.

▶ Set the `Original symbol color` to active.

Repeat these steps for each value If you do not want to use graphics for visualization of a value, simply write the value and do not assign a display element to it.

If you no longer wish to display a previously-assigned display element, remove this by clicking on X.
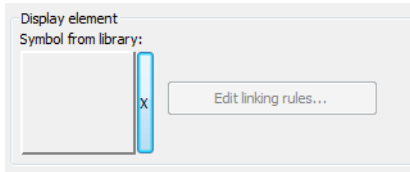
11.2.13 Switch

You show the value of a binary variable in graphic form and modify it with the **Switch** dynamic element.

Proceed as follows to configure a **switch**:

▶   Open the screen `Storage tank`.

▶   In the menu or the element symbol bar, select the **Switch** entry.

▶   With the `left mouse button held down` in the *storage tank* screen, drag the frame for your **switch** element.
      Position this element precisely underneath the previously-inserted symbol for the throughflow meter.

▶   In the dialog that now opens, select *Storage tank[1].Inflow - Switch*.

▶   Make the following settings in the properties window:

Property group	Property field	Description
Representation	Predefined graphics:	DIP
Runtime	Tooltip:	Water inflow on/off

Repeat these steps for the outflow:

▶   When selecting the variable, use *Storage tank[1].Outflow - Switch*.

▶   Enter *Water outflow on/off* in the properties of the   **Runtime** group as a **tooltip**.

Switch with buttons

The 'Switch' element can also be used as a switch for one-stage unsecured command processing input.

In this example tutorial, a switch for the heating is to be configured. In addition, an on/off switch is configured for the heating as a button. The heating is active as long as this switch or button is pressed. The minimum running time of the heating process should be 5 seconds.

Carry out the following steps to configure the task:

▶ Open the screen `Storage tank`.

▶ In the menu or the element symbol bar, select the **Switch** entry.

▶ With the `left mouse button held down` in the *Storage tank screen*, move the frame for your **Switch** element. Position this switch under the right-hand corner of your storage tank.
In the dialog that now opens, select *Storage tank[1].Heating element.Switch on/off*.

▶ Make the following settings in the properties window:

| Property group	Property field	Description
Display	Predefined graphics:	ELECTRICITY
Runtime	Tool tip:	Tank heating on/off

Repeat these steps for the switch:

▶ In the menu or the element symbol bar, select the **Switch** entry.

▶ Position this element below the tank symbol, on the left next to the already-configured combined element of the heating.

▶ When selecting the variable, use *Storage tank[1].Outflow - Switch* again.

▶ As **Pre-defined graphics:** Select *BUTTON*.

▶ Enter *Tank heating pulse switch* in the properties of the **Runtime** group as a **tool tip**.

In order to create a button from the switch, change to the **Write set value** group in the properties window.

Property field	Setting parameters	Description
Momentary switch	active	Defines switch as a Momentary switch.
Momentary switch On	active	Determines the direction. If this property is active, the button is set to off by default. The switch is set to on by pressing this button.
Holding time [ms]:	5000	This results in the button being active for at least 5 seconds.

EXAMPLE ILLUSTRATION OF YOUR PREVIOUS CONFIGURATION (DETAIL WITH HEATING ACTIVATED):
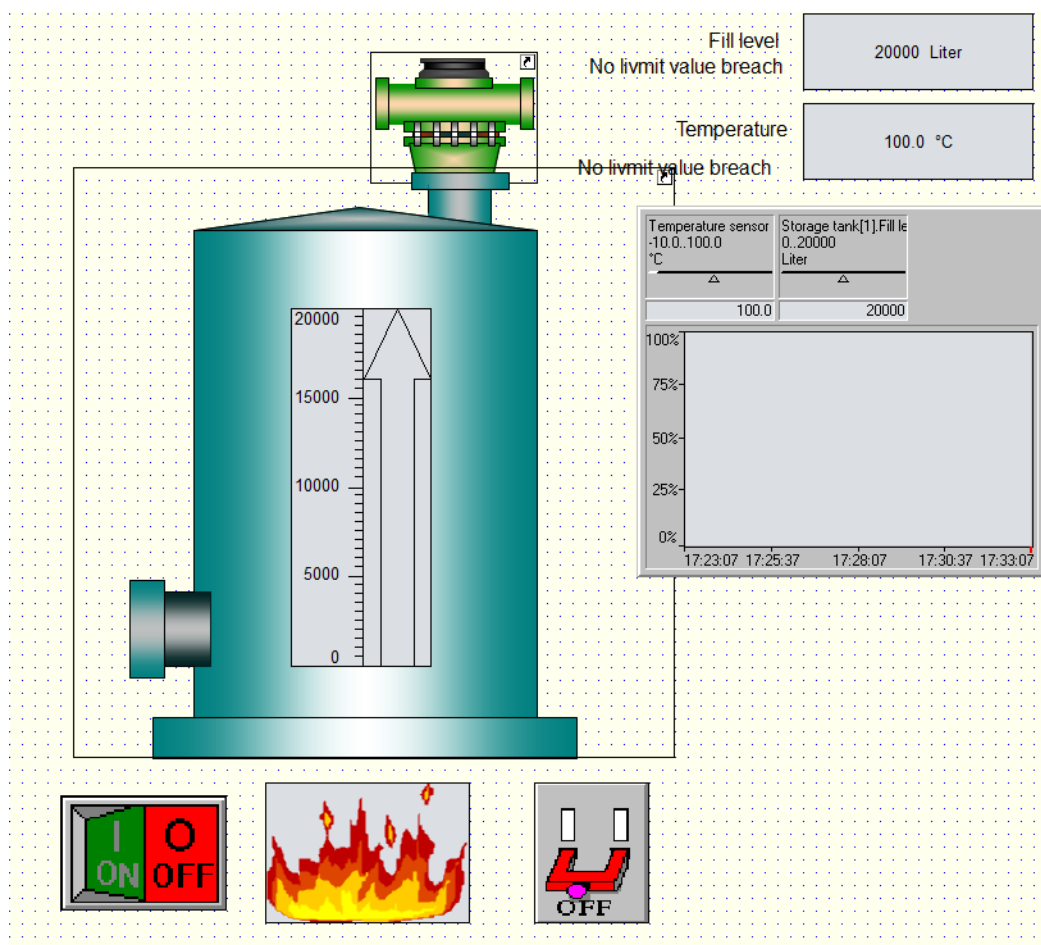
> 🔘 **Information**
>
> The properties are only available for the button if the screen element is a switch.

11.2.14 Status Element

You control a screen symbol with a **Status Element** dynamic element. In doing so, properties of a linked variable such as colors or flashing for example are transferred to a symbol. You can therefore also show statistical limit value texts.

Proceed as follows to configure a **status element**:

- ▶ Open the screen `Storage tank`.

- ▶ Select the **Status element** element in the menu or the tool bar.

- ▶ With the `left mouse button held down` in the *Storage tank* screen, drag the frame for the status element.
  Position this element to the right of the (already-configured) universal slider for the water temperature.

- ▶ In the dialog that has now opened, select *Storage tank[1]/.Temperature sensor*

- ▶ Select the `shutter_red_2` symbol from the global symbol library and position this close to the status element.
  Note: This symbol is in the **interface_buttons_16_9** group.

- ▶ Set the parameters of your status element in the property window:

Property group	Property field	Description
Representation	Font:	8 - keys
	Symbol:	shutter_red_2
	Display limit	active
Colors	Text/line color:	Storage tank [1].Temperature sensor
Visibility	Variable:	Storage tank [1].Temperature sensor
	take from limit	Active
Send value to hardware	Setting values active	Inactive
Variable / function	Variable:	Storage tank [1].Temperature sensor

There is no need to set the parameters for the symbol in this case, because visibility and coloring of this symbol can also be controlled by the status element. In addition, this type of configuration also has the effect of the limit value text being displayed as a text in the corresponding limit value color.

If you only want to display status with graphics, it is sufficient to configure a symbol with the corresponding visibility settings. In this case, the `Display limit value text` display setting is configured as **Display** `Inactive` in the properties group.

Note: This project configuration for visibility is carried out in the status element.

11.2.15 Combo-/Listbox

You link values of variables with texts using the combo/list box dynamic element. You use this in two ways:

▸   If a text is selected in runtime, the value of the variable changes to the value that is linked to this text.

▸   If the value reaches the value of the defined value, the linked text is displayed in the combo/list box with the current value.

In this tutorial, the functionality of this element is to be set to fill the fill level of the storage tank with the corresponding values from a combo box. As a second application, a list box is configured that transfers pre-defined temperatures to the temperature sensor.

Static combo box

To configure a **combo box**, proceed as follows:

▶ Open the screen `Storage tank`.

▶ Select the **Combo-/Listbox** entry in the menu or the tool bar.

▶ With the `left mouse button held down` in the *storage tank* screen, drag the frame for your element.
Position this element above the (already-configured) pointer instrument for the water temperature. Ensure that you configure a height of at least 150 pixels.

▶ In the dialog that has now opened, select *Storage tank [1].Fill level*

▶ The dialog to set the parameters of your combo box opens:
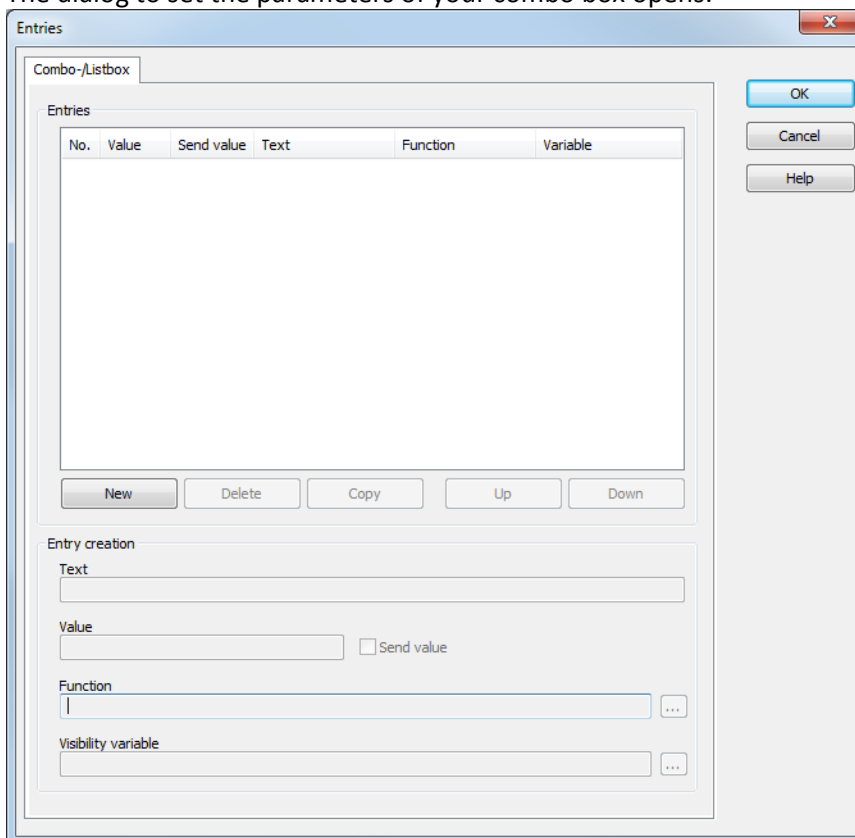
Note: if the 'Entries' dialog is not automatically opened, switch to the **Display** group of properties. Click in the 'Entries' properties field: `Click here ->`

▶ A new value is configured in the entries list by clicking on 'New'.

• Values can be written to variables can be using 'Send value active'

- A function call is carried out by selecting ... with a function
  Note: Value and function can be configured independently from one another.

▶ Enter the following values:

Value	Text
100	100 liters
5000	5,000 liters
10000	10,000 liters
15,000	15,000 liters
19500	19,500 liters

▶ Project configuration content remaining the same:

- Enter Storage tank [1].Fill level as a visibility variable for each.
  Send value is active

For (subsequent) changes to your configuration:
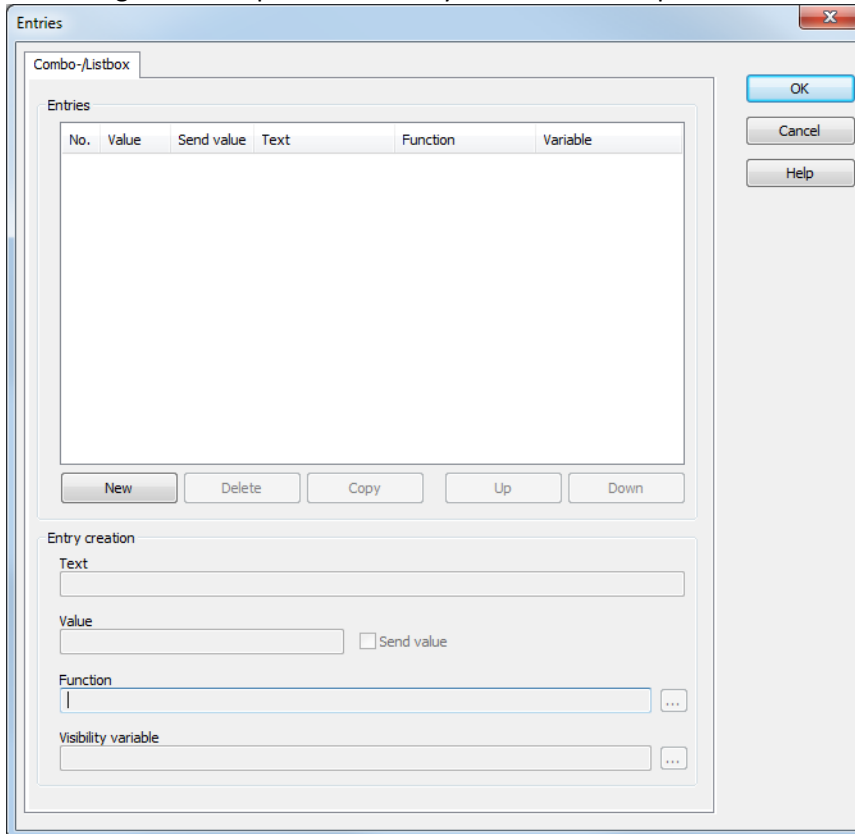
▶ Select the (already-configured) combo box / list box element

▶ The dialog for setting parameters in the **Display** property group is opened via **Click here->** in the `Entries:` property field.

Static list box

To configure a **list box**, proceed as follows:

▶ Open the screen `Storage tank`.

▶ Select the **Combo-/Listbox** entry in the menu or the tool bar.

▶ With the `left mouse button held down` in the *storage tank* screen, drag the frame for your element.
Position this element to the left of your symbol for the storage tank. Ensure that you configure a height of 60 pixels.

▶ Select *Storage tank[1].Heating elements.Temperature sensor* in the dialog that opens.

▶ The dialog to set the parameters for your lit box now opens:

▶ Enter the following values:

Value	Text
0.5	Freezing (0, 5°)
2.5	Risk of freezing (2.5°)
16	Normal (16 °)
55	Danger of overheating (55 °)
99	Evaporation (99 °)

Visibility variable and function are not configured in this example. Once you have entered your values, your configuration should look as follows:
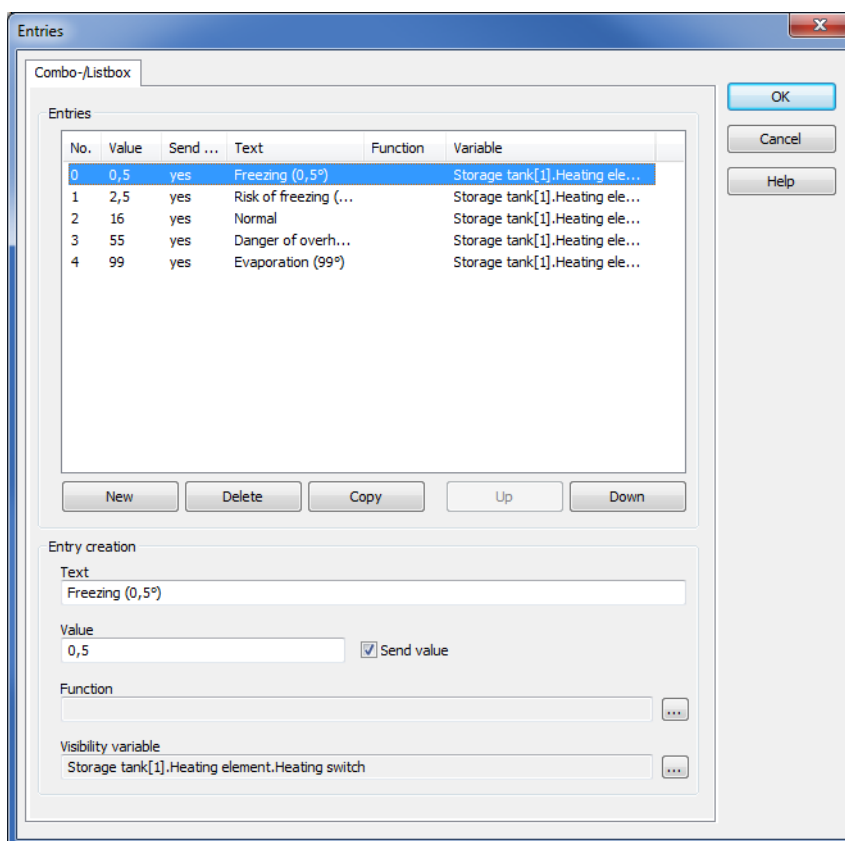
Text color with combo boxes / list boxes

The text color of the combo box /list box entries can be configured depending on limit value colors.

These settings are configured in the properties window:

▶ Select your list box for the water temperature

▶ Switch to the **Colors** group in the properties window

▸ Assign, under 'Dynamic colors' in **Text/line color:** in the properties field, *Storage tank[1].Temperature sensor* by selecting the . . . button .

PROJECT CONFIGURATION VIEW:
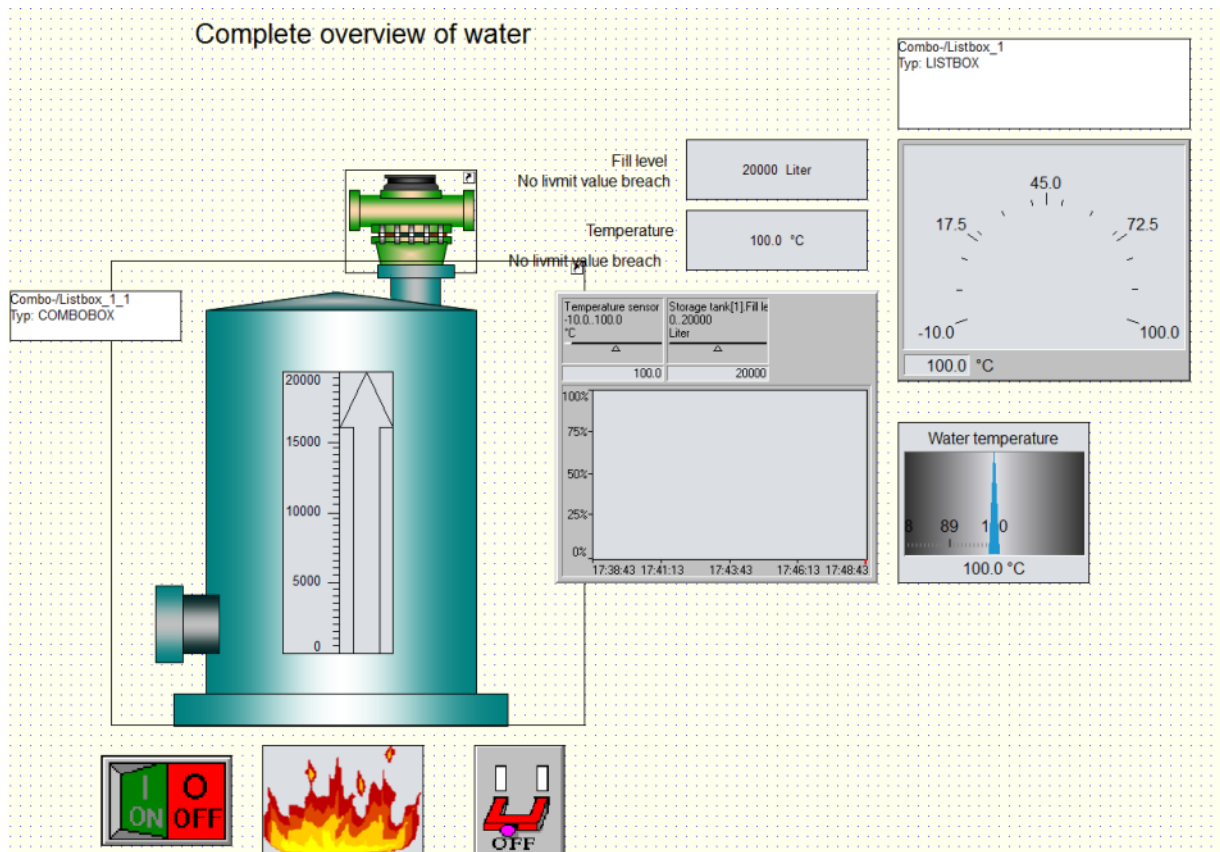
Note: As a result of the low height, not all entries in this list box are visible in Runtime. In this case, there is the possibility to scroll in Runtime.

11.2.16 Multibin

You can define several states with the multi-binary dynamic element:

▸ display graphics depending on variable values

▸ Color symbols

▸ Issue status text

▸ link numerical values and binary values

▸ display a switch or key for the first variable, provided the first variable is a binary variable

CONFIGURATION EXAMPLE:

To illustrate the functionality of the dynamic screen element, the following problem is posed:

For your project configuration, a pipe is to be routed from a stream to the storage tank. This route is 1,000 meters long. A heating station is to be installed at a distance of 200 meters, which activates the heating in the event of a drop in temperature. This activation is possible by means of a switch An overview is to be configured, in which that status of the individual switches is also output as a text. If a heating station is active, this is to be shown graphically.

Preparatory project configuration

You will get to know other functions of zenon through the task at hand.

The focus is on

- ▶ Editing of symbols
- ▶ Reusability
- ▶ Multiple use of existing project configurations

NECESSARY CONFIGURATION STEPS:

The following steps are necessary before you can configure the multi-binary element dynamic element in this tutorial:

- ▶ Creation of a new screen (already configured).
- ▶ Creation of a button in the navigation bar (already configured).
- ▶ Creation of a function to control this screen (already configured).
- ▶ Configuration of a symbol template and repeated use of a symbol in a screen.
- ▶ Configuring the multi-binary element

Symbol editing on the screen

For the planned configuration, only one symbol, which in turn consists of several symbols, is used more than once. There is the possibility to draw a sample symbol group and then `copy and paste` it into the screen.

However, your project configuration is more efficient and most of all less prone to errors if you first configure a new symbol from different symbols and then use this from the symbol library in the future. You will also become familiar with the substitution function of zenon.

First create a new symbol from the existing symbols. To do this, carry out the following steps:

Create a new empty screen and give this a name, for example *Temporary*.

▶ Configure a switch

• Select `Heating Line[1].Heating switch` as a variable

▶ Configure a pointer instrument

• Select `Heating Line[1].Temperature sensor` as a variable

▶ Configure a combined element

• Use the *fire* symbol for the bit value 1 You can find this fire symbol in the *International Symbols* symbol group.

• Default status and bit value 0 do not have a symbol

Arrange your project configuration according to the picture below. The overall size of this symbol group should be approximately 350 pixels high and 250 pixels wide:
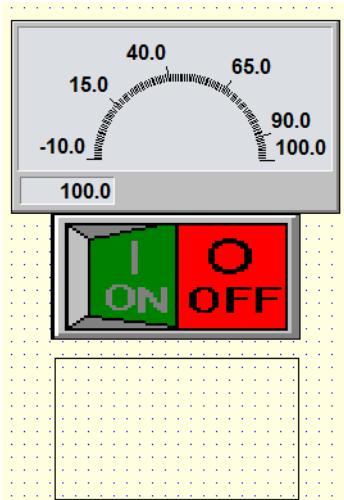
Note: The fire symbol is not visible with the combined element, because the default value is not assigned a symbol.

ADDING TO AN EXISTING SYMBOL LIBRARY

▶ Highlight all three symbols by dragging a frame around the three elements with the mouse button held down.

▶ Select **Symbol** -> **Create element group** from the context menu.

▶ If you now call up the context menu, you can select the "new" **Add to symbol library ...** command.

Symbol	▶		Create element group
Element Position	▶		Resolve
Arrange	▶		Switch to single edit mode
Replace links...			Insert in existing element group...
Cut	Umschalt+Entf		Insert into symbol library...
Copy	Strg+C		
Paste	Strg+V		
Paste at same position	Strg+Umschalt+V		
Delete	Entf		
Transfer format			
Full-screen mode	Umschalt+F9		
Redraw screen/symbol			
Display grid			
Use grid			
Use magnetic points			
Zoom	▶		
Export screen/symbol as a graphic file...			
Insert vector graphic...			
Properties...	Alt+Eingabe		
Help	F1		

▶ Then select the save location of your "new" symbol in the dialog that opens. Select the project library for this tutorial.
The element group is inserted in the symbol library and converted into a symbol.

> 💡 **Information**
>
> *Once your symbol has been added to a symbol library, the screen template* Temporary *can also be deleted. There is no actual link between screen and symbol in zenon.*
> *However you can use this screen for later symbol configurations.*

Multiple use by means of linking rules

You now have a self-created symbol. This symbol contains three different screen elements and two variables.

▶ Now open your (previously-configured) line overview screen.

▶ In the project manager, open the **Screens** node and the project symbol library there.
Note: You should see the symbol you created beforehand in the project symbol library.

In the next step, you will apply this symbol to a screen a total of nine times.

The symbol you have configured in the symbol serves as a sample for this. All elements, variables and functions contained in this symbol are automatically taken on during project configuration.

Continue with the project configuration:

▶ `Drag and drop` your symbol to the screen open `line overview` screen.

▶ The **`Edit linking rule`** dialog opens automatically.
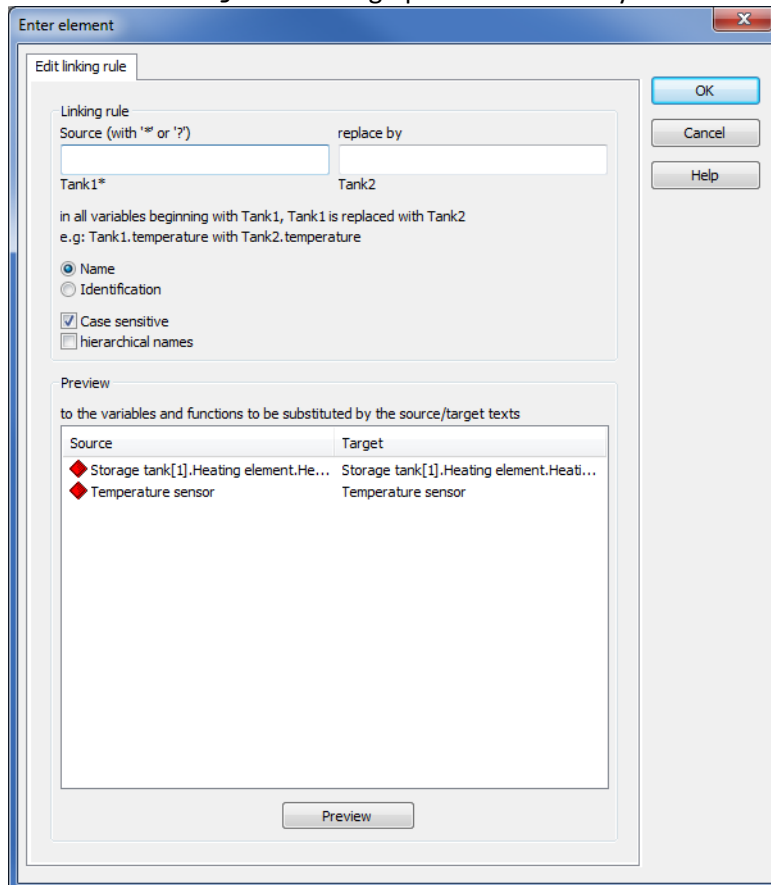
▶ Confirm this dialog for your first symbol with **OK**.

Substitution of symbols

This use of an element or symbol from a symbol library leads to each positioning of your symbol being created with the variables configured in the template. You have configured nine switches and the temperature sensor for the first heating station.

A short-term solution to configure the symbol from the symbol library without variables and functions and then assigning these then in the screen. However this workaround very quickly reaches its limits with more complex configurations. Subsequent changes must also be repeated individually for each symbol.

For your first symbol, the variables stored for the symbol are still correct.
However, in order to also be able to subsequently assign the correct variables, carry out the following steps:

▶ `Drag and drop` the sample symbol in your line overview screen again.

▶ However, for this second symbol, enter **Source** *Heating line[1 on* in the input field.

▶ In the **replace by**, enter *Heating Line [2*.

▶ Repeat this step again twice. As a replacement rule, select:

  • *Heating Line[3* for the third symbol group

  • *Heating Line[4* for the fourth symbol group

Use the `Direction` menu bar to orientate the first line of your project configuration. You should now have your symbols arranged next to each other four times in the upper half of your screen.
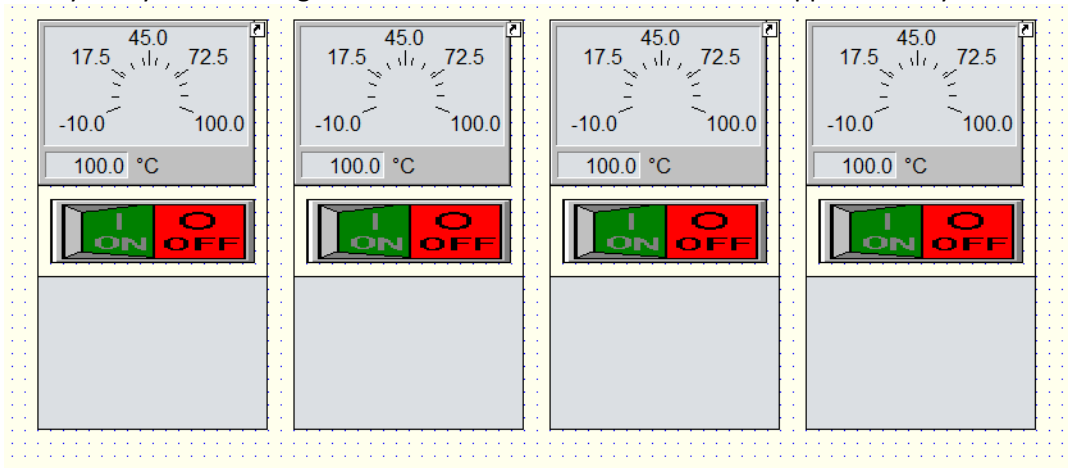
Release properties from links

For optimum use of symbols, it is sometimes necessary to assign individual values to individual properties from system elements.

You can release the **Tooltip** property with this function for example. If the symbol is dragged onto the screen, the 'Tooltip' property can be changed individually for each motor. This function is available for all properties.

Carry out the following steps to release the 'Tooltip' of your element group:

▶ In the project manager change to the node `Screens`.

▶ Select, from the `Project symbol library` your elementgroup in the detail view.

▶ You open the element group with a double click.
Note: With element groups, ensure that you select the main element.

▶ Select the switch symbol in the main window.

▶ Select the property group in general and give it the following **element name:** *Heating stations tooltip.*

▶ In the properties window, select **Tooltip:** for *Heating switch station*.

▶ Now move the word `Tooltip:` from the properties window (Runtime properties group) via Drag&Drop into the bar of the main window under the symbol that has been opened.
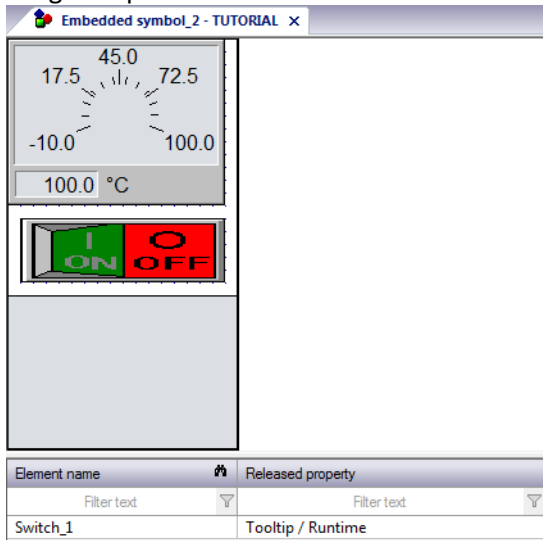
▶ Close and save the symbol.

▶ Change to the *Line overview* screen

▶ Select a (compiled) symbol and change to the `Runtime` properties group in the properties window.
Note: At the place where the tooltip is normally entered, the corresponding properties field is missing. Your released element is at the end of the properties help for this.
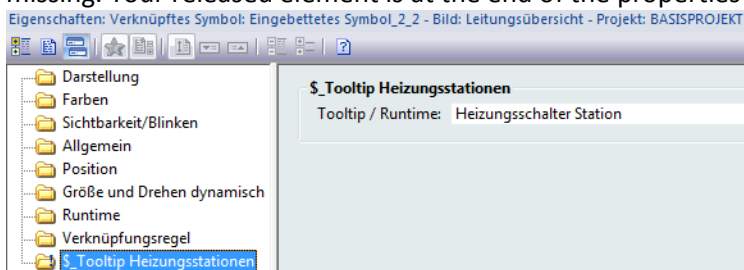
> ### 💡 Information
>
> *If you configure a neutral text for the tooltip before release, this is automatically applied to the previous configuration.*

TO BLOCK A RELEASED ELEMENT AGAIN:

▶ Select, in the main window, the property name or the released property.

▶ In the context menu, select `Remove selected element`.
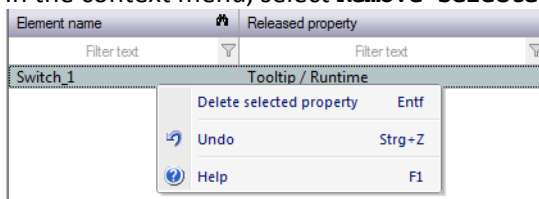
Linked Symbol Wizard

The Linked Symbol Wizard is available in order for you to be able to edit approved properties easily and with a clear overview: double click on a symbol that:

▶ Is linked in the screen and

▶ has released properties

The Linked Symbol Wizard opens
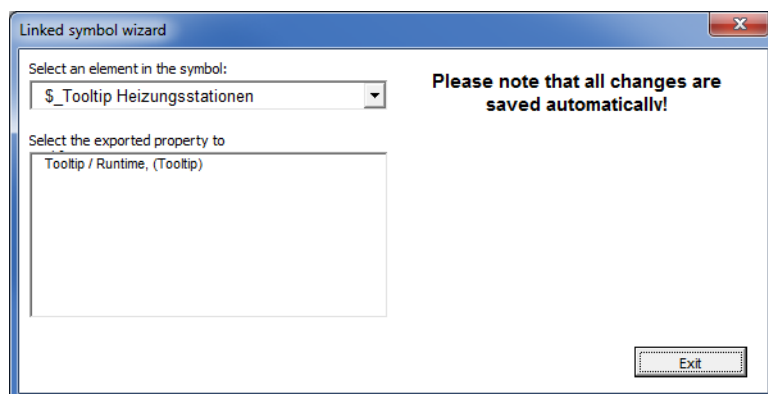
▶ Select the property to be changed

▶ Enter the new value**Note:** Each change is accepted immediately!

▶ Close the wizard by clicking on `Exit`

Visibility levels

Similar to CAD programs, you can define visibility levels in the Editor and allocate them with unique elements. That way you can easily hide unused elements in complex images and show them again anytime. 15 visibility levels are available. Levels are allocated using the 'Visibility' property in the 'General' node. Individual levels can be activated or deactivated using the buttons of the Visibility levels toolbar:

First show the `Visibilty levels` tool bar.

To do this, go to the *Options => Toolbars=> Visibility levels* menu bar.
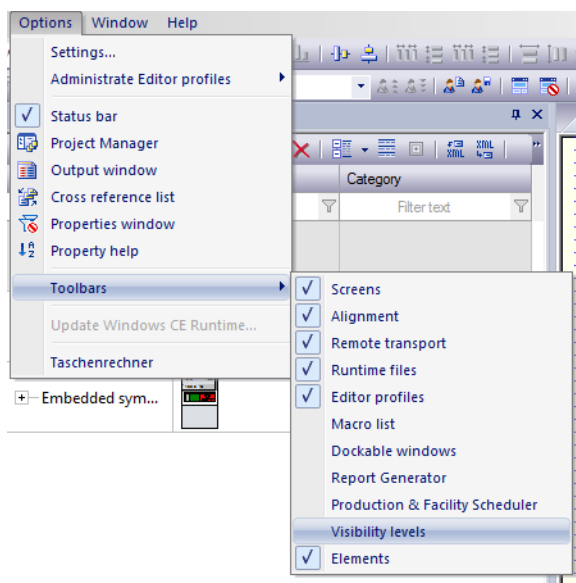
💡 **Information**

Visibility levels apply graphics configuration to the selected level.

Working with visibility levels

Lastly, you configure the pipeline in your line overview. Your pipe should look something like this when finished:
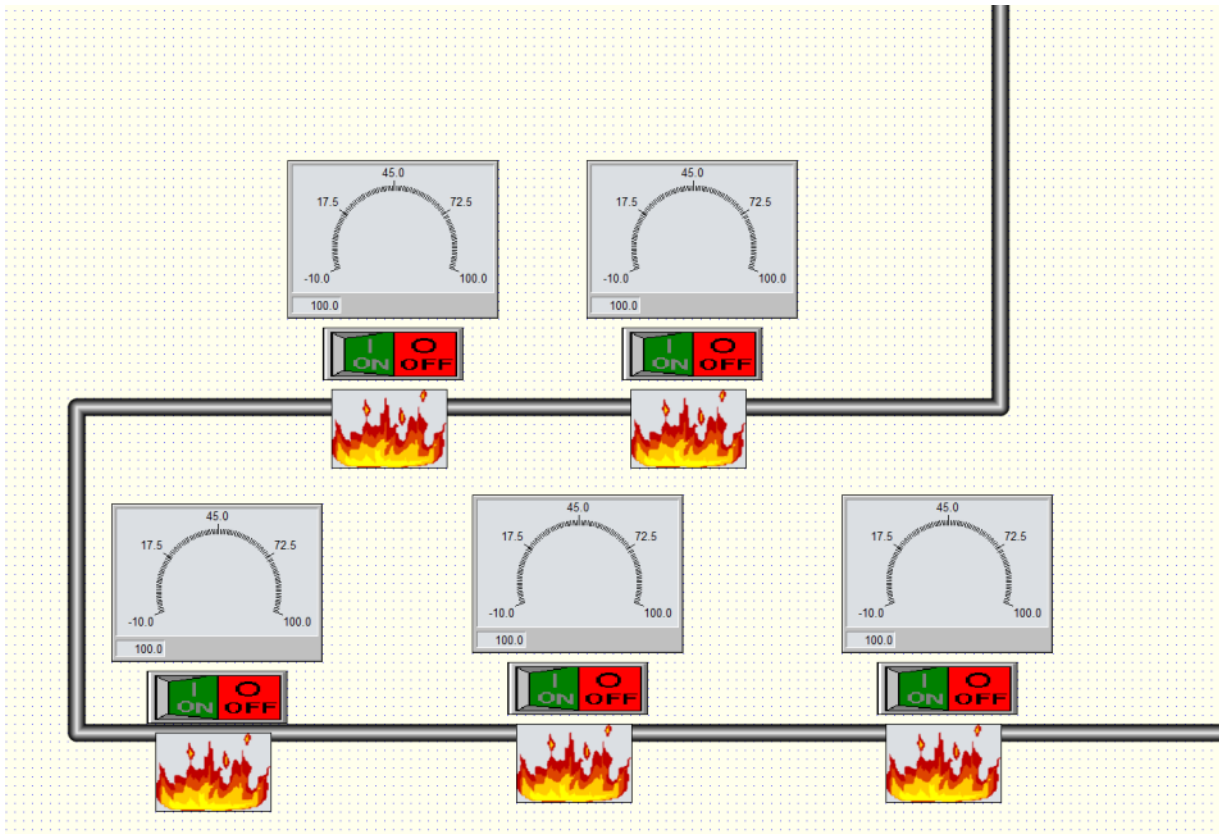
carry out the following steps to draw a pipeline with the help of visibility levels:

▶ Open your *Line overview* screen in the main window.

▶ Switch this screen to full screen mode.

▶ select the Tube symbol in the Elements tool bar or in the Elements drop-down list

⌐

- In the main window, select the start point by clicking the left mouse button

- pull the first section

- place an initial corner point with a simple mouse click (left)

- draw another line and place the corner points

- set the end point by double clicking or pressing the `esc` key

- Highlight your pipe and select `Background` from the `Element Position` context menu.

- **Hint:**
  The shape, size and position can be changed at any time by pulling the vertexes or moving

the element Hold down the `shift` key when pulling; this way, only horizontal or vertical movements away from the original position are possible

▶ In the properties window, in the **General** property group, select **Visibility level:** *1*.

DISPLAY OF VISIBILITY LEVELS

Now look at your screen configuration again.

If you, in the tool bar for the visibility levels, first click on the button for `Make all visibility levels invisible` and then activate level `1` by clicking the mouse on the `1` button of the tool bar, you see your pipeline only.

However, if you select level `0` active and level `1` not active, you see your heating station without a pipeline.

> 💡 **Information**
>
> *Visibility levels have no effect on the view in Runtime.*

Practical tips for visibility levels

ATTENTION:

If you hide visibility levels, do not forget to show these again. If you, for example, hide all visibility levels in the `Line overview` screen up to level 2, level 2 is only now visible in the `storage tank` screen. In this example tutorial, this would lead to you not seeing any more content in the storage tank screen. Only once you show all levels again in the **Visibility levels** toolbar via `Make all visibility levels visible`, is all previous configuration content visible.

Configuring the multi-binary element

Carry out the following steps to configure a **multi-binary element**:

▶ Open the `Line overview` screen.

▶ Select **Multi-binary element** in the menu or the tool bar of the element.

▶ With the `left mouse button held down`, in the `Line overview` screen, drag the frame for your **multi-binary element**.
Position this element above the first row with the four symbols.

▶ In the dialog that has now opened, select *Heating line[1]/.Heating switch*

▶ The dialog to set the parameters of the multi-binary element opens
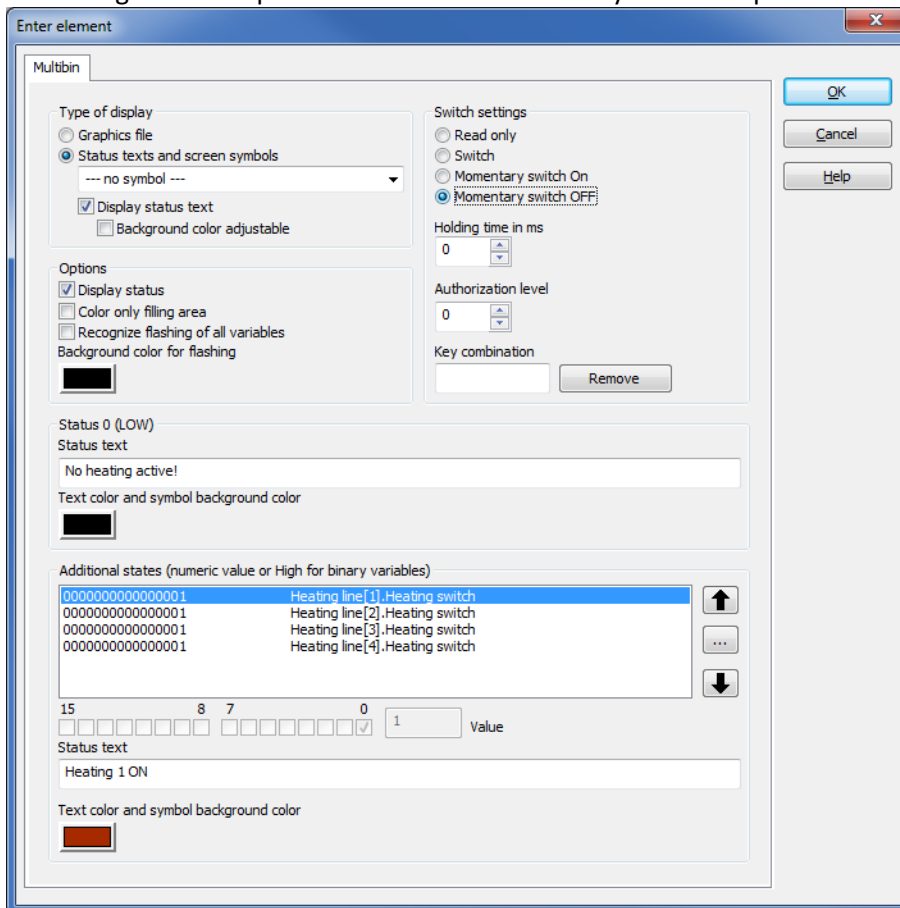
Setting parameters for the multi-binary element

When setting parameters for the multi-binary element, first define a default status and then step-by-step states.

Proceed in the following way:

▶ Highlight your (already) configured multi-binary element

▶ Switch to the **Display** properties group in the properties window.

▶ Select `8 – key` as the font

▶ Open the configuration dialog by clicking on `. . .` in the **Configuration** property

▶ Set the parameters for your multi-binary element in accordance with the following requirements:

- `Status text and screen symbol` as **display type**

- `Display status text` active

- `Display status` active

- `No heating active!` As **status text** in the **Status 0 (LOW)** input field

▶ For setting the parameters of other states:

- Clicking on the `...` button opens the variable selection.
  Note: You can select several variables when pressing the `Ctrl` key. Click on the `...` button again to add further variables.

- Set the parameters for your individual variables by highlighting these, entering `Heizung n` each time as the respective status text and selecting a vibrant red as the text color for each.
  Configuration example:
  Highlight the first variable
  Add *Heating 1 active* in the **Status text** input field
  Select a vibrant red as the **text color and symbol fill color**

  Highlight the second variable
  Highlight the first variable
  Add *Heating 2 active* into the **Status text** input field
  Select a vibrant red as the **text color and symbol fill color**

...
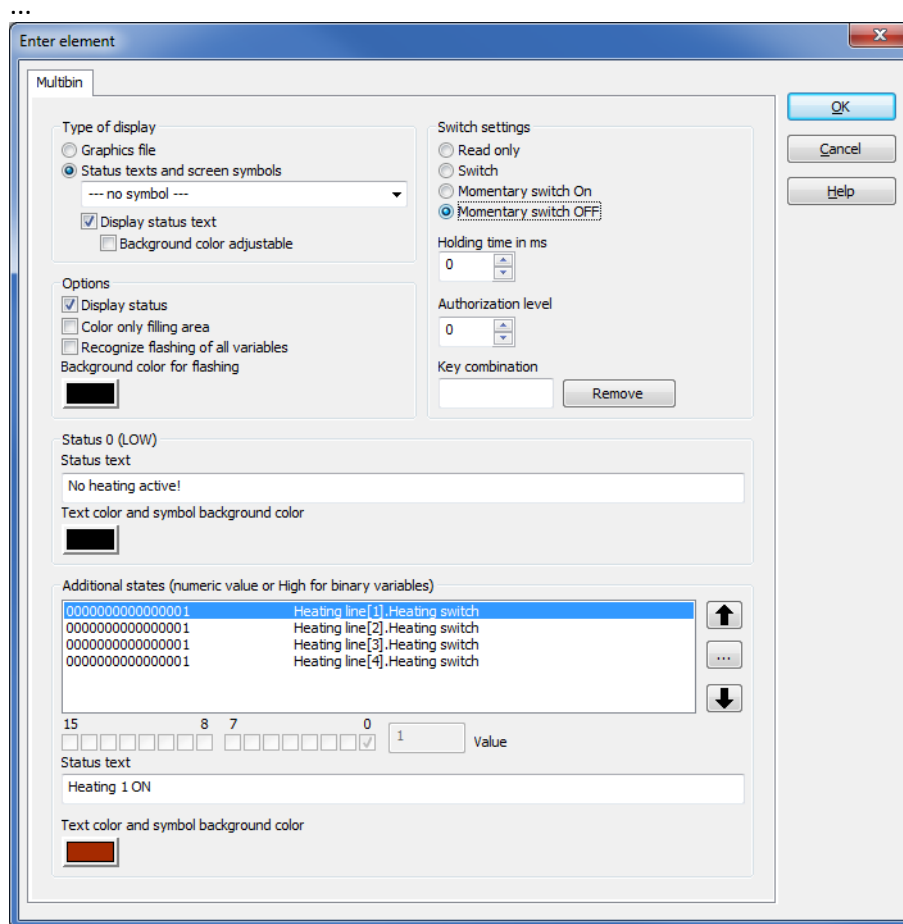
▶ Then configure a tooltip in the **Runtime** properties group with the `Visualization of status of pipe heating` entry.

12. Multiple use by means of substitution

You have often read about a second storage tank in this tutorial. However this has not been created as a screen previously nor have the configuration steps been carried out for this second tank.

The same structures do not need to be configured multiple times thanks to substitution. You configure just one function and apply this to an existing screen. zenon detects this "multiple use" and automatically offers a filter dialog when the function is being created. The parameters for changes or substitutions can then be set in this dialog.

SUBSTITUTION:

▶ Configure a new **screen switching** function.

▶ Select, in the `screen selection` dialog, the (already-configured) *Storage tank 1* screen.

▶ The `Filter` dialog is opened

▶ Now set the parameters for the substitution rules.
In this example, the complete contents of storage tank 1 are to be substituted for storage tank 2:

   • Enter *Storage tank[1]* in the source field (=Tank1*)

   • Enter *Storage tank[2]* in the target field (= Tank2)

▶ Confirm your project configuration by clicking on the `OK` button

▶ Confirm the subsequent dialog
"`Source variables have not yet been replaced by the target variables! Substitute now?`"
with the `Yes` button.

Lastly, apply the function that has just been created to the (already-configured) 'Storage tank 2' button.

▶ To do this, switch to the `screens`.

▶ Select the *Navigation bar* screen and

▶ Apply the *go_to_storage_tank_2* function to the button
(by dragging & dropping or using the properties of the button)

12.1   Hints for Substitution

To be able to apply the possibility to apply the substitution optimally, note the following:

▶ Use structure data types.

▶ Ensure that your variables are given a short and concise name.

▶ Take the possibility of substitution into account at the naming stage. This naming should be unique and easily-substitutable.

▶ Configure a screen as a template first, which can be reused as often as you like once it has been completed by means of a function.

13. Screen types

zenon offers a range of special screen types for special tasks. After selecting a special screen type, all user elements for the respective screen type are available in the Control elements menu. Menu item

add template opens a selection dialog for adding pre-defined layouts with certain control elements at pre-defined locations.
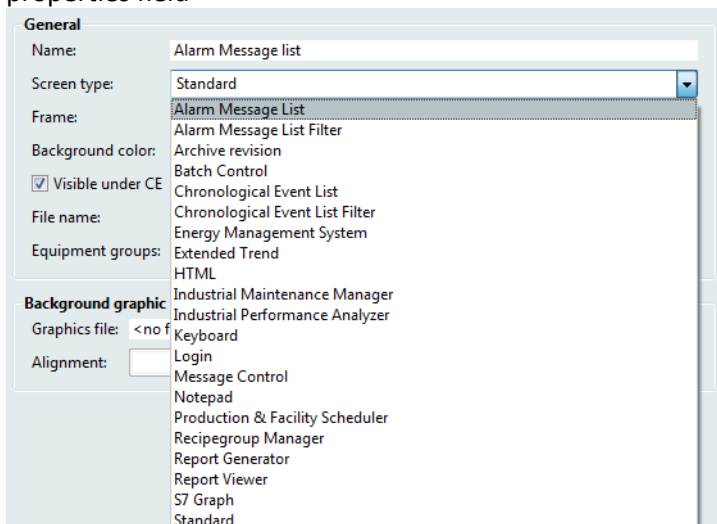
⚠ **Attention**

If a type is changed from a standard type to a special type, the new screen type is fixed with the screen and can no longer be changed.

13.1 Configuration of special screen types

Carry out the following steps to create a new screen type:

▶ Create a new screen

▶ Select the corresponding screen type in the **General** properties group, in the **Screen type:** properties field

Note: Alternatively you can use the drop-down list, in the detail list, in the `Screen type` column.
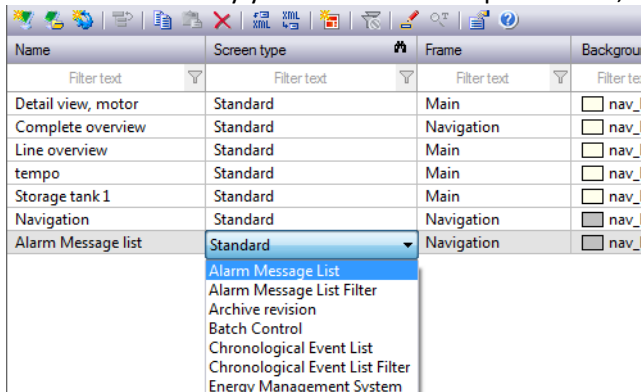
▶ Configure new screens in accordance with the following requirements:
Note: All screens to be configured use the *Main* frame.
The Tooltip column is for subsequent configuration of the buttons in the navigation bar.

Name	Screen type	Tooltip:
Alarm Message List	Alarm Message List	Alarm Message List
CEL	Chronological Event List	Chronological Event List
Diagnosis of variables	Variable diagnosis	Variable diagnosis

OTHER CONFIGURATION STEPS

Switch back to the (already-configured) *Navigation* screen and create the corresponding buttons for the newly-created screens.

13.2   Content of special screen types

Once special screen types have been created, you will now find out how to fill these with content in the next step. Now fill your variable diagnosis, CEL and alarm message list screens by selecting, in the *Menu bar=> Control elements=> Add template... .*

To do this, carry out the following steps:

▶ In the project manager, highlight the `Screens` node.

▶   Open the corresponding screen in the detail list with a *double click*.

▶   In the menu bar, select `Control elements -> Add template`
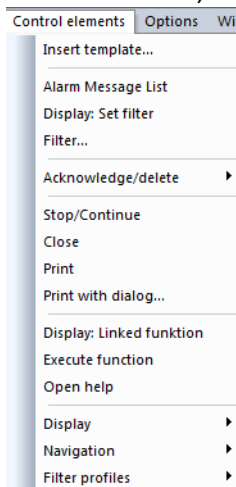
▶   Select the corresponding template in the `Select template` dialog.
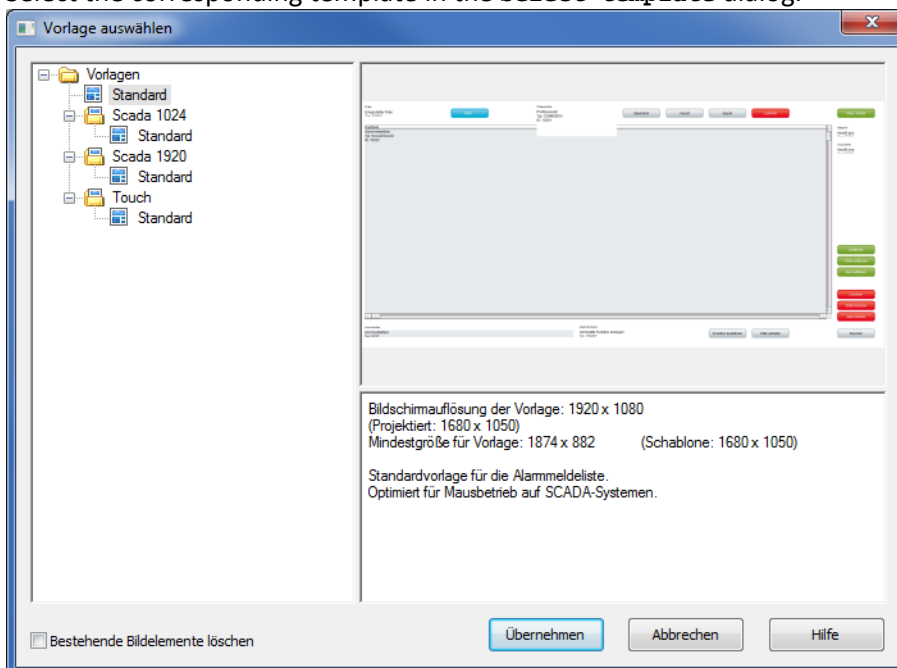
Note: Alternatively, you can also add individual modules from this pre-made template. These are listed in the selection list under `Insert template....`

THE CORRECT TIME

Insertion of templates always relates to the content that is configured at the time of insertion. Configuration that is added later such as variables is not automatically added later. It is therefore recommended that the content of special screen types is created as late as possible during configuration.

13.3   Screen switch to screen types function

The screen filter dialog is displayed using the screen switch function on an Alarm Message List, Chronological Event List or Time Filter screen, .

With the screen filters, you have the possibility to comfortably create individual filter pages. You do not have to use the pre-defined zenon dialogs for this; you can compose your own filter dialogs instead. You can do this according to aspects such as touch operation, filtering of special areas only, ability to switch languages, etc.

> ### 💡 Information
>
> *Special screen types need only be created once, even when used multiple times. They can be opened with different data by means of the screen switching function.*

13.3.1   Switch screen to special screen types function

The screen switching function is configured in the same way as standard screen types. zenon automatically recognizes that a special screen type is given as an objective. A filter dialog is automatically called up in the Editor when the function is configured.
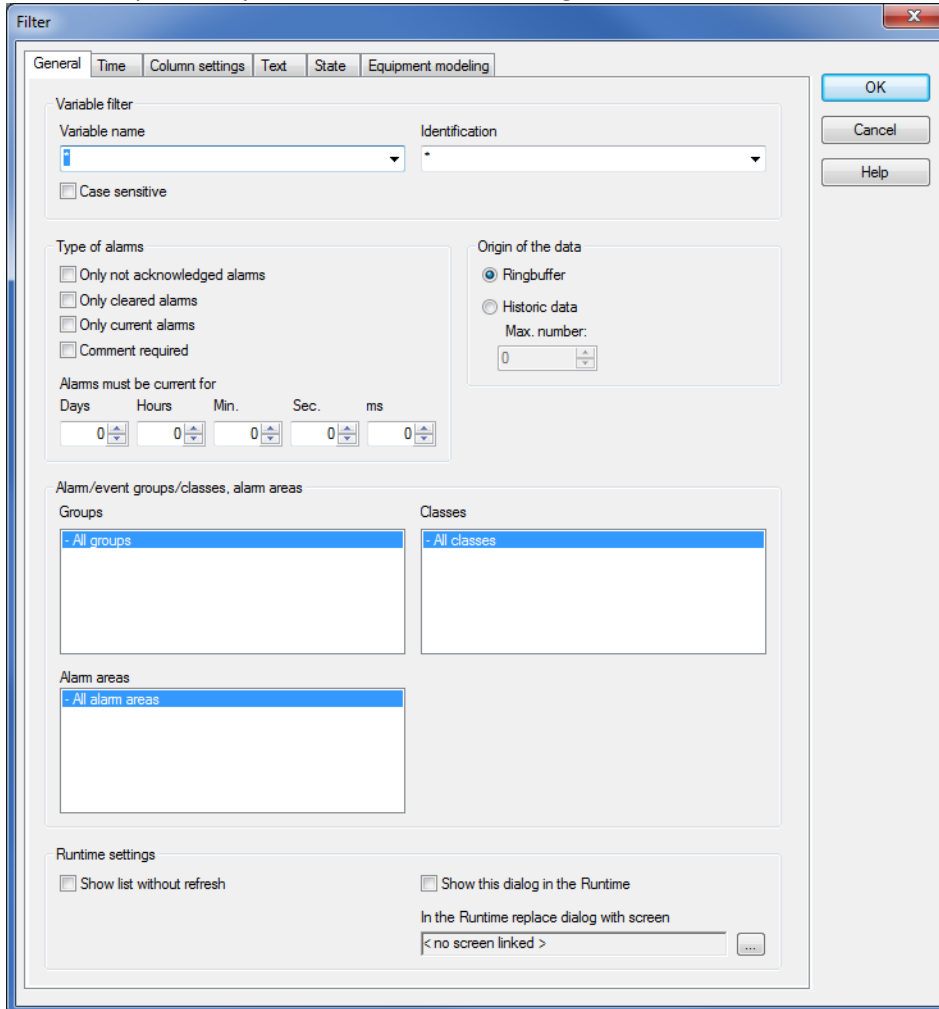
Carry out the following steps:

▶ Create a screen switch function for your (previously configured) screen types

   • Alarm message list
      Function name: *switch-screen_Alarm-Message-List*

   • CEL
      Function name: *switch-screen_CEL*

- Variable diagnosis
Function name: *switch-screen_Variable-Diagnosis*

▸ In the dialog that opens automatically, activate the respective **Offer this dialog in Runtime**. This allows the possibility to amend the filter settings in Runtime.

▸ If this dialog deactivated in Runtime, screen switching is carried out immediately in Runtime with the filter settings set (in the Editor) for the respective screen.

Note: If the filter dialog is not opened automatically, you can call this up again in the properties window in the **General** properties group by pressing the `...` button of the `Parameter:` property field again. The filter dialog appears again after the variable selection dialog.

13.3.2 Applying the function

Apply the functions that have just been created to the buttons in your navigation bar:

▸ Open the `Screens` node in the project manager.

▶ A *double click* opens the `navigation bar` screen in the main window.

▶ Move the corresponding function to the respective button by `dragging + dropping`.

13.4   Your own screen types

Creation of your own screen types is simple. Proceed in the following way:

▶ Configure and set the parameters for a screen according to your requirements.

▶ Once your screen has been finalized, select `Create template for screen type...` in the context menu

▶ You can now save your screen as a screen type in the dialog that now opens.

CONFIGURATION OF YOUR OWN SCREEN TYPE

If you construct a screen based on your own screen type, proceed as follows:

▶ Create a new screen.

▶ Retain the proposed `standard` screen type.

▶ Open your screen in the main window by double clicking the screen name in the detail view.

▶ Insert your own content using the `Control elements` and `Insert template...` menu items.

▶ Select the corresponding template in the dialog that now opens.
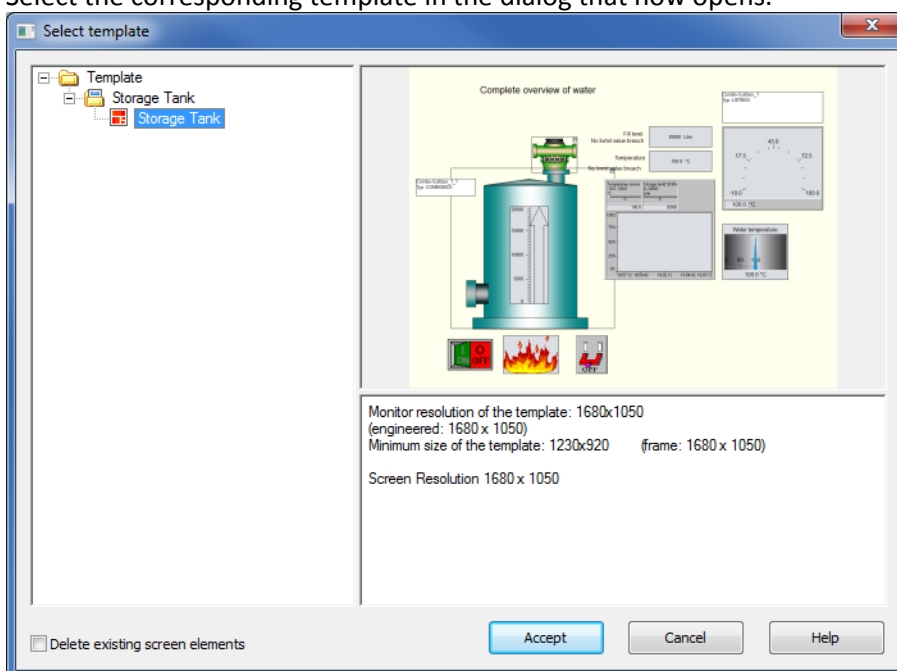
▶ The template is applied to your new screen.

> 💡 **Information**
>
> *You can save several visualizations for each screen type. Use this possibility to optimize it for various systems and screen resolutions by offering several solutions.*

14. Runtime (Online operation)

zenon consists of

- ▶ The configuration environment, the Editor,
- ▶ and Runtime

Projects are created in the Editor and operated and monitored in Runtime.

Runtime / runtime environment allows projects to run regardless of platforms, i.e. regardless of the hardware and operating system used. A Runtime environment is the 'platform' in which the projects run. Runtime does not need a development environment in order to run.

14.1 Start Runtime

Runtime can be started in several ways:

1. By clicking on the `Start Runtime` button in your Editor
2. In the Start menu in the program group *Programs/COPA-DATA/zenon***720** with the entry `zenon Logic Runtime`
3. By pressing the `Runtime` button in the zenon Startup Tool
4. With the `F5` key in the Editor

> ⚠ **Attention**
>
> *If you start Runtime from the Editor, do not forget to save your Runtime data `[F7]`.*

If you carry out configuration in the Editor whilst you have opened Runtime, update your project configuration by pressing the (already-configured) reload button.

14.2   Working in the Runtime

After starting the Runtime two images are loaded.

First, the overal summary (since this was automatically entered as `Start screen` during the project configuration) and then the image *Navigation bar* (since it was executed with the **Start function** of the start screen).

First, you will be shown how to orient yourself in an existing Runtime project:

- ▶ Right-click anywhere in an empty area of the screen.
  If you hold the right mouse button the name of the screen will appear at the mouse pointer.

- ▶ Move the mouse over a button or a screen element.
  The tool tip is displayed behind the mouse pointer.

- ▶ Now click with the `right mouse button` on the numerical value element for the operating mode.
  In the left corner above the element the name of the linked variable will be displayed.

- ▶ In the navigation bar, click the button *end Runtime* and hold it: the button labeling will change.

14.3   Stopping the Runtime

To end Runtime:

- ▶ Press the **End Runtime** button to close Runtime.

- ▶ Pres `Alt + Q`
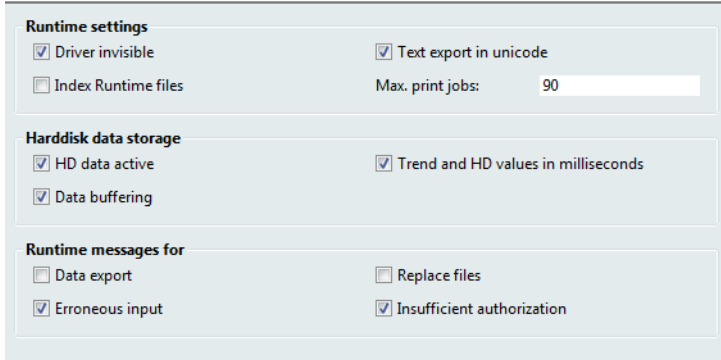
  > 💡 **Information**
  >
  > *If you have not defined a button for stopping the Runtime and if the Windows title bar is not available, the Runtime can also be closed with the key combination Alt+F4.*

14.4   Configuring Runtime

The settings for Runtime are made during configuration in the Editor. To do this, carry out the following steps:

- ▶ Highlight the project name in the the project manager.

▶   The following properties field is shown once the Runtime Settings properties group is called up:

Runtime settings
☑ Driver invisible   ☑ Text export in unicode
☐ Index Runtime files   Max. print jobs: 90

Harddisk data storage
☑ HD data active   ☑ Trend and HD values in milliseconds
☑ Data buffering

Runtime messages for
☐ Data export   ☐ Replace files
☑ Erroneous input   ☑ Insufficient authorization

💡 **Information**

If the Runtime setting **Block system keys** is active, the system keys are also no longer available in the other applications on the computer in question.
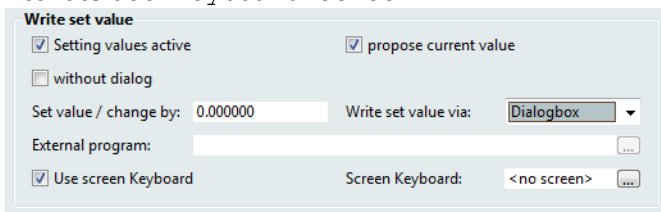
14.4.1   Use screen Keyboard

When configuring a project, always bear in mind that the operator on site often does not have hardware equipment like your project configuration computer.

The hardware on which Runtime runs is not always a desktop computer in the conventional sense. If Windows CE or Windows Embedded is used on the Runtime system, there is often no keyboard available for operation. Allow for this situation by offering a virtual keyboard.

To automatically switch on a virtual keyboard in Runtime:

▶   Select the corresponding element in your project configuration.

▶   Switch to the **Write set value** group in the properties window.

▶   Activate `Use keyboard screen`.

Write set value
☑ Setting values active   ☑ propose current value
☐ without dialog
Set value / change by: 0.000000   Write set value via: Dialogbox ▼
External program: [...]
☑ Use screen Keyboard   Screen Keyboard: <no screen> [...]

If you now change a field value in Runtime the keyboard is called up automatically.
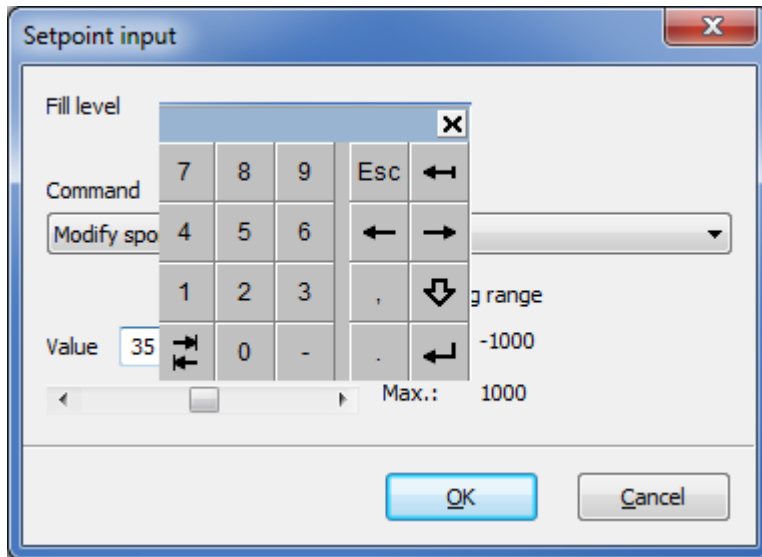
HARDWARE KEYBOARDS AND VIRTUAL KEYBOARDS USED TOGETHER

Virtual keyboards act like hardware keyboards in the operating system. If virtual keyboards are used together with hardware keyboards, they affect each other. Settings on the hardware keyboards - e.g. determine whether Shift is used - affect the virtual keyboard and vice versa.

Examples:

If Capps lock is pressed on the hardware keyboard, it is also true for the virtual keyboard. In this case you cannot enter numbers via the automatic keyboard.

If the Num key on the hardware keyboard is pressed in order to switch of the number pad, the number pad on the virtual keyboard also does not work.

14.4.2   Keyblock Runtime

Keyblock Runtime Start is a program with which both zenon Runtime is started and all Windows system keys can be blocked. Keyboard shortcuts such as the Windows key or `Ctrl+Alt+Del` no longer have an

effect.
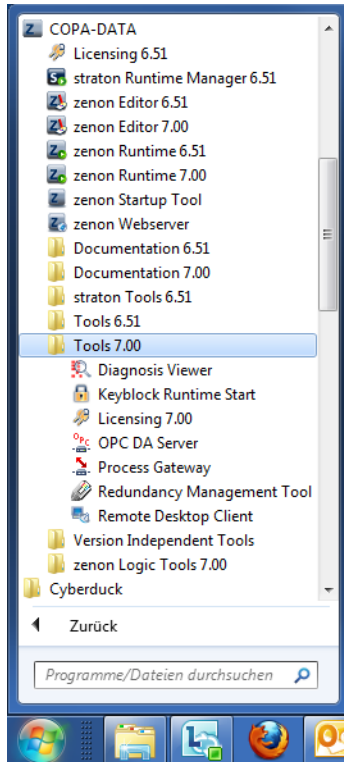
To use Keyblock Runtime Start:

▶ In the Windows start folder, open zenon Tools in the COPA-DATA folder

▶ Select Keyblock Runtime Start

▶ the program starts and automatically starts the Runtime

▶ The program blocks all access to the operating system.

WINDOWS KEY

When locking the system keys, the normal operation of the scroll bars with the mouse in the Runtime is also blocked. You can work around the blocking with the help of the context menu.

HIDING THE CONTROL PANEL IN THE START MENU

▶ Locking the tool bar for operation

▶ Prevents

• Changing passwords

• Closing Windows

• Logout

- Locking the computer

- User change

- Hiding all element in the task manager

> ### 💡 Information
>
> *If Keyblock Runtime Start is started using the startup process of the operating system:*
>
> ▸ take care that the auto start folder is user-specific: If an other user is logged in, the program is not executed.
>
> ▸ Processing the startup process programs can be prevented by pressing the Shift key when the operating system is booting.

This locking cannot be bypassed during Runtime. When the Runtime is closed normally, the system restrictions are canceled. If the Runtime should be operable without these limitations, the Runtime must be started instead of Keyblock Runtime Start.

> ### ⚠ Attention
>
> *Ensure that you configure a possibility to close Runtime in your project, such s a button, for example.*

14.5   Practical tips for Runtime

Also bear in mind that Runtime is the user interface of your project configuration.

▸ When configuring, ensure that there is a clear overview and that it is easy to use

▸ Create different authorization levels

▸ Write set values for all inputs in Runtime when configuring in order to avoid incorrect entries.

▸ Use the functions and scripts in order to make use of as much automation as possible.
  Note: each entry entered by a human entails a certain amount of risk.

▸ If you use Runtime on an ongoing basis to check individual project configuration steps, do not forget

- to save the Runtime data and

- to load the current project configuration data in Runtime.

▸ Configure meaningful tooltips. They offer a good first opportunity for orientation in Runtime.

14.6   Changing the simulation mode

Your configured driver has up to now been configured with the status simulation - counting. This means that your project continuously counts up. However, with Boolean variables in particular, this can mean that they often switch on and off.

In order to observe the behavior of Boolean variables correctly, it makes sense in this tutorial to change the driver mode from `counting` to `static`.

Proceed in the following way:

 ▸ In the project manager, switch to the **Variables** node.

 ▸ Select **Driver** -> `S7 TCP-IP driver`

 ▸ Open the driver configuration dialog in the **General** properties group by clicking on `...` the `Configuration` property.

 ▸ Change the **Mode** to `Simulation - static`

Note: If you make this change when Runtime is running, do not forget to back up your Runtime data in the Editor and to reload it into Runtime.

> 💡 **Information**
>
> *You can reuse all configuration steps that have already been carried out when changing your driver configuration. Reconfiguration is not necessary.*

14.7   Changing values in Runtime

Because the other variables (such as fill level, temperature) can now keep a constant value, you will use setpoint input. To do this, click on the corresponding configuration in Runtime and xx modified spontaneous value.

▸ Click on the numerical value element for the variable with the `left mouse button`. *Storage tank[1].Fill level*.

▸ The standard dialog box for setting values opens and allows you to change the value of this variable.
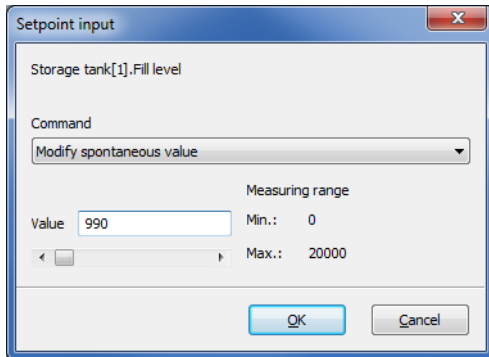
▸ Set the value of the variable *Storage tank[1].Fill level* to *990* and close the dialog box with OK.

▸ Set the value of the variable *Storage tank[1].Heating elements.Temperature sensor* to *0* and close the dialog box with OK.

> 💡 **Information**
>
> *The driver is set to the counting simulation mode and the bar of the bar graph is moving.*

▸ You can switch between the screen with the configured buttons in the button bar.

15. Automated function call

Until now, you have always triggered functions by means of actively instigating it. You have learned, how to ...

▸ Change values with the input of set values,

▸ Call up functions with buttons.

You will later configure how certain states (such as fill level or temperature) execute functions automatically.

In this part of the tutorial, you will learn:

▸ Calling up a function by means of a reaction matrix
Example: If a value of a given fill state is gone below or exceeded, the corresponding valve is opened or closed.

▸ Calling up a function by mens of a limit value breach
Example: If a given temperature is gone below, the heating of the water pipe is automatically

activated. The heating is automatically deactivated again once a certain temperature has been reached.

15.1   Calling up a function by means of a reaction matrix

The reaction matrix allows function calls, regardless of variables. In doing so, logical states in the reaction matrix are configured and the limit value of a variable is set as related to the parameter.

In this tutorial, the inflow valve of the water tank is opened if the limit value of 1,000 liters is gone below and closed if the limit value of 19,000 liters is exceeded.

Proceed in the following way:

- ▶ Configure the corresponding functions
- ▶ Configure a reaction matrix
- ▶ Assign your variable a function related to the limit value

15.1.1   Reaction matrix - preparing functions

The inflow valve is to be controlled in the event of a value being exceeded or gone below. First configure your functions for this:

- ▶ Create a **Write set value** function.
- ▶ Select the variable *Storage tank[1].Inflow - Switch*
- ▶ Set value = *On (1)*
- ▶ Select **Direct to hardware** *active* in the options
- ▶ Name this function *set_tankinflow_ON*
- ▶ Configure a second function with the set value as *Off (0)* and the name *set_tankinflow_OFF*.
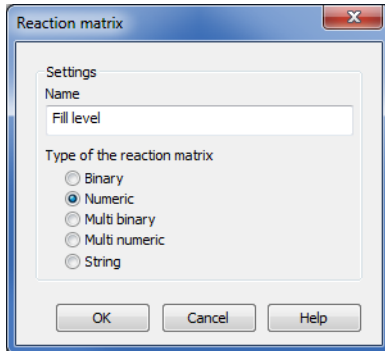
15.1.2   Reaction matrix - configuring the reaction matrix

To engineer a **reaction matrix**, proceed as follows:

- ▶ In the project manager, change to the **Variables** node.

▶ Open this node with **[+]**.

▶ Activate the 'Reaction matrix' node and select, via the context menu, `(mouse + right click)` **New reaction matrix... .**
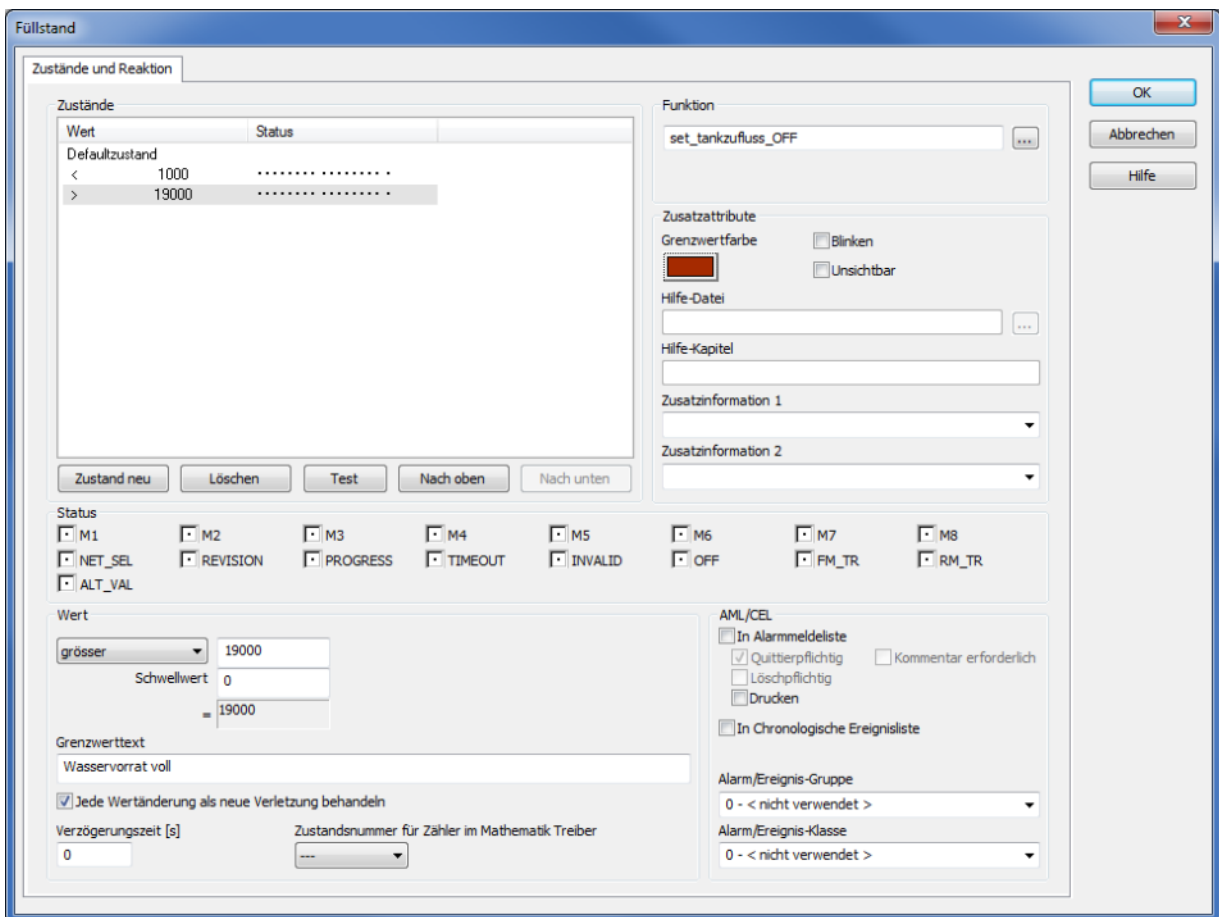
▶ In the **reaction matrix** dialog that opens, enter `Fill level` as the `name` and *Numerical* as the `type`.

The **States and reactions** dialog opens. You set the parameters for states, limit value texts and functions here.

▶ Use the **New status** button to create a new value.

- Create a value entry for <1000

- Assign this status the (already-configured) set_tankinflow_ON function

- In the additional attributes, set the (already-configured) color fill level low from the color palette

- Enter *Low water supply* in the **Limit value text** field

- **Treat each change of value as new limit violation:** must be set as active

▶ Configure another status with the following requirements:

| Parameters	Description
Value	> 19000
Function	set_tankinflow_OFF
Limit color	Fill level high (dark red from color palette)
Limit text	Water supply full
Treat each change of value as new limit violation	active

15.1.3   Reaction matrix - assigning limit values

In order for your reaction matrix to be triggered with the states and functions contained therein, assign your matrix limit values from variables.

To do this, carry out the following steps:

- ▶ In the project manager, change to the **Variables** node.

- ▶ Open this node with `[+]`.

- ▶ Switch to the **Data types** view

- ▶ Select, in the detail view, the `Storage tank` structure data type and open this with `[+]`.

- ▶ Select the *Fill level* structure element there.

- ▶ Select the **Limit values** properties group in the properties window.

- ▶ In the **Reaction matrix** property field, select the (previously-configured) *Fill level* reaction matrix from the drop-down list.

No other parameters need to be set for the other limit value entries.
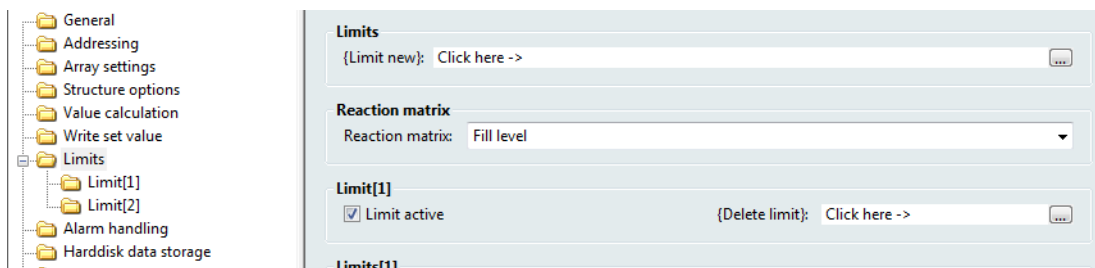
EFFECTS IN RUNTIME

If you test this configuration in Runtime, note the switch for the tank inflow:

- ▶ If the fill level < 1,000 liters, the switch is automatically 'On'.

- ▶ If the fill level > 19,000 liters, the switch is automatically 'Off'.

Use either the set value input or the (already configured) static drop-down list to change the fill level.

15.2   Function call when limit value breached

You will become familiar with the function call in the event of a limit using the (already-configured) *Line overview* screen.

In doing so, the heating should be activated if the lower water temperature limit is breached. As soon as the water temperature gets back to a normal value, the pipe heating is to be switched off accordingly.

Start this configuration by creating the attendant functions:

Note: This function is unfortunately not replaceable.

To do this, carry out the following steps:

▶ In the project manager, change to the **Variables** node.

▶ Open this node with `[+]`.

▶ Switch to the **Data types** view

▶ Select, in the detail view, the `tank` structure data type and open this with `[+]`.

▶ Select the *Fill level* structure element there.

▶ Select the **Limit values** properties group in the properties window.

▶ In the **Reaction matrix** property field, select the (previously-configured) *Fill level* reaction matrix from the drop-down list.

No other parameters need to be set for the other limit value entries.

16. Menus

In the previous project configuration of this tutorial, you have only implemented navigation using screens and buttons. In this section of the tutorial, you will learn another possibility, by configuring menus and context menus.

The following steps are necessary to implement menus:

▶ Configuring the functions

▶ Configuration of the menus

▶ Setting the parameters for the menus

▶ Setting the parameters for the project settings

> 💡 **Information**
>
> *The functions do not need to be reconfigured. You use the already-used and thus already-configured functions.*

16.1 Creating menus

In this example, you will configure menus for navigation between your (already-configured) screens.

To do this, carry out the following steps:

- ▶ In the project manager, switch to the `Screens` node.

- ▶ In the detail view, expand the **Main menu view** by clicking on `[+]`.

- ▶ Select `New main menu` via the tool bar or the context menu.

- ▶ Give this menu a name by entering, in the **Name:** properties field, the name *Navigation*.

- ▶ You can now see a cell at the top left of the cross-reference list.

- ▶ Double click in this cell and write *Home* in this cell.
  Alternatively, enter the text in the properties window, in the **Display / type** group in the **Text:** properties field .
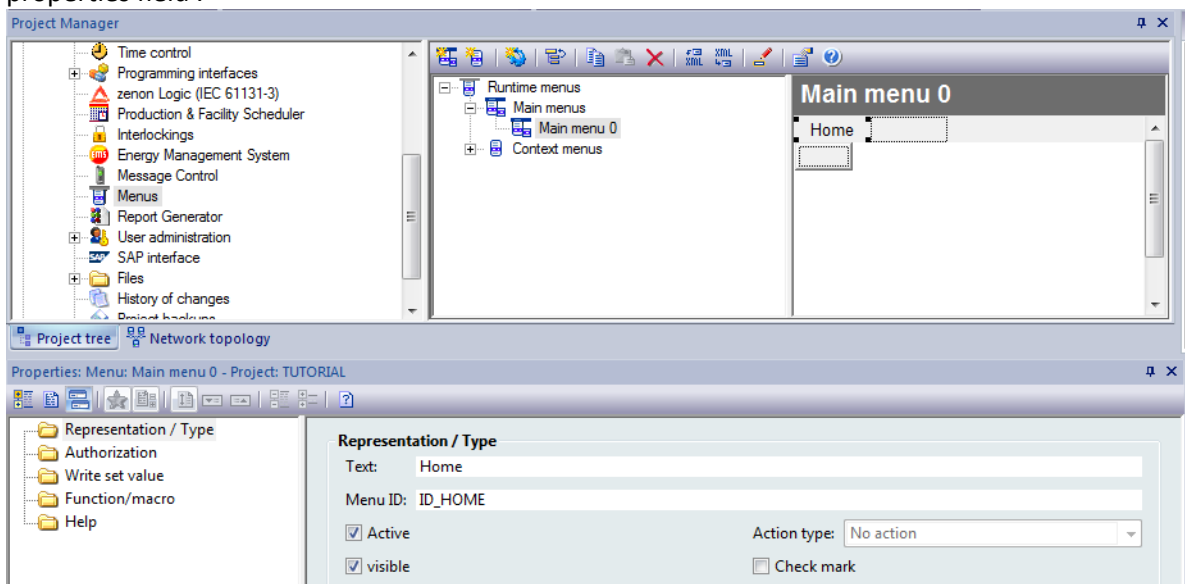
- ▶ If you now click on the *Home* entry, an empty cell is offered under the menu text.
  You can configure the contents of the menu by clicking in this cell.
  Note: Each entry made then results in an empty cell underneath. The same applies for vertical menu items. It is possible to move them by dragging and dropping.
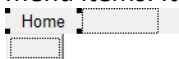
16.1.1 Properties of the menu configuration

You can set the parameters for special menu-related content in the properties menu of your respective menu entry.

- ▶ Separator active
  In the **Display / type** properties group, you can activate the `Separator active` properties field

instead of a menu entry. This results in you placing a separator between your menu entries.
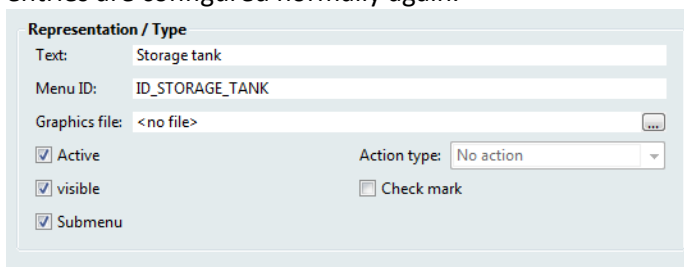
▶ Submenu
In the properties group **Display / type**, a subordinate branch opens if `Submenu` is activated.
Note: This property is activated for the superordinate menu entry. The respective submenu entries are configured normally again.

▶ Action type:
You can also find the `Action type:` property field in the **Display / type** submenu. You can assign you menu entry with an action in this list.
If you want to, for example, call up a function with your menu entry, this `action type` is to be set to *function*.

16.1.2 Configuration of menus

Complete the configuration of your menu in accordance with the following requirements. The naming of the assigned functions follows the configuration progress of this tutorial:

HOME MENU ENTRIES:

▶ Reloading Runtime

▶ Stopping the Runtime

▶ *Separator*

▶ COLOR view

▶ B/W view

NAVIGATION MENU ENTRIES

▶ Complete overview

▶ *Separator*

▶ Storage tank

- Tank 1

- Tank 2
  Note: Tank 1 and Tank 2 are submenus. To be able to configure submenus, activate, in the **Display** group / type in the property window the check box for the property `Submenu.` In this example, the `Checkbox submenu` for the *Storage tank* menu item is active. This is visible in the Editor by means of a triangle next to the *Storage tank* entry.

▶ Line overview

▶ *Separator*

▶ CEL

▶ Alarm Message List

▶ Variable overview

16.1.3 Assigning a function

To assign individual menu functions, proceed as follows:

▶ Select the respective menu item.

▶ Select, in the **Display / type** group in the property window, the `Action type:` property field and select *Function* from the drop-down list.

▶ Switch to the **Function/Macro** properties group.

▶ Clicking on ... in the `Function` property window opens the dialog for selecting a function.

16.1.4 Application of menus

Your menus are configured in the cross-reference list and the parameters are set in the properties window. The application is called up with a function.

First create a function to do this:

▶ Switch to the 'Functions' node in the project manager.

▶ Create a `Display menu` function.
  Note: You can find this function in the **Screens** function group.

▶ In the dialog that opens, select the `Display menu` action from the drop-down list.
  As a **Menu**, select the (previously-configured) `Navigation` entry.
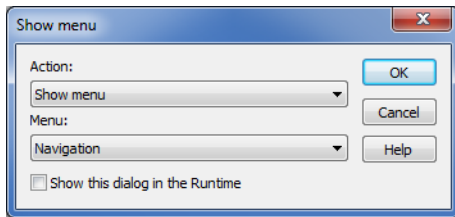
Note: If **Offer this dialog in Runtime** is active, this dialog is not executed automatically but is offered to the user for manual selection in Runtime. Your configured settings are given as a proposal in the process. If you want to offer a neutral menu selection to the Runtime user, `Action:` and `Menu:` can also be configured without selection.

▶ Name this function *load_menu*.

APPLYING THE MENU FUNCTION

The following step is necessary in order for menus to be displayed in Runtime:

▶ Select, in the project view, the workspace; it is *Tutorial* in this example.

▶ Select the **Graphic design** group in the properties.

▶ Set the `main menus active` property to active (active checkbox).

One possibility is applying the function on a button (such as *Show menus*).

However in this tutorial, the menu is to be displayed automatically when Runtime is started. Activating menus by means of start scripts is described precisely in the scripts chapter.

16.1.5   Practical tips for menus

▶ When configuring, use the multiple selection option with `shift + mouse click`. First create your menu entries, then highlight these by clicking the mouse and pressing the `shift key`. Then switch to the **Display / type** properties menu and select the corresponding `action type:`. Your configuration is applied to all selected menu entries as a result of this procedure.

▶ Use the option of `submenus` and `separators` to keep your menus simple and with a clear overview.

▶ Use already-configured functions, such as the screen switching functions already applied to buttons in this tutorial.

▶ If you notice that a function is missing when you are configuring menus, you do need need to switch to the `Functions` node.
   Note: When selecting functions whilst setting parameters, use the `New function` button in the

function selection dialog.
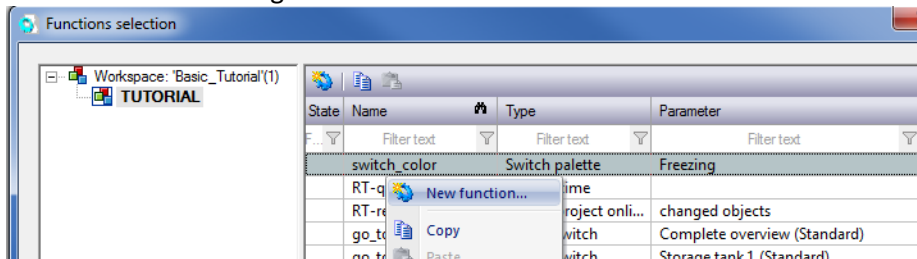
16.2 Creating context menus

As an example of the configuration of a context menu, prescribed temperatures are to be set in the overview of the storage tank using the context menu. In addition, the temperature is to be increased or reduced within the limits by 10 degrees. The temperature is to be reduced by 10 degrees with the click of a mouse.

To do this, proceed as follows:

▶ Configuring a context menu

▶ Configuring attendant functions

▶ Applying functions to a context menu

▶ Applying a context menu

16.2.1 Setting the parameters of a context menu

To configure context menus, proceed as follows:

▶ In the project manager, switch to the `Screens` node.

▶ In the detail view, expand the **Runtime menu** by clicking on `[+]`.

▶ Click on **context menu**.

▶ Select `New context menu` via the tool bar or the context menu.

▶ Give this menu a name by entering, in the **Name:** properties field, the name *Context_menu_storage_tank*.

▶ The procedure for entering the context menu entries is the same as with the menus:

• + 10 degrees

• - 10 degrees

• Enter temperature

16.2.2   Functions for context menus

Create the attendant functions for the context menu:

- ▶ Create a new `Write set value` function**.**
  Note: You can find this in the **Function favorites** or in the **Variable favorites group**.

- ▶ Link this function to the `Storage tank[1].Heating element.Temperature sensor` variable.

- ▶ Set the parameters for the 'Write set value' dialog according to the image below:
  It is important that the checkbox for `Propose current value` is activated!

- ▶ Name this function *+10*.

- ▶ Create a second function with the name *-10* and a change of `-10`.

ASSIGNING FUNCTIONS FOR CONTEXT MENUS

Proceed as follows to assign the functions that have just been created:

- ▶ Call up your (already-configured) *Context_menu_Storage_tank* context menu.

- ▶ Select the (already-configured) context menu entry *+10* degrees.

- ▶ In the properties, select in the **Display / type** group, the `Action type:` property *Function*.

- ▶ Now switch to the **Function/Macro** properties group.

- ▶ Select, by clicking on the `...` button in the `Function` property, the above-configured function *+10*.

- ▶ Repeat this step for the *-10 degrees* context menu and use the *-10* function in the process.

- ▶ For the 'Enter temperature' context menu entry select, in the `Action type:` property, *Function*.

16.2.3   Applying a context menu

Applying the context menu to an element

- ▶ In the project manager change to the node `Screens`.

- ▶ Open the (already-configured) `Storage tank 1` screen.

- ▶ Place a **bar graph element** under the list box, on the left next to the tank symbol.

- ▶ Select `Storage tank[1].Heating elements.Temperature sensor` in the dialog that opens to select variables.

▸ In the **Display** properties group, select `Direction:` in the properties field *Right* from the from the drop-down list.

▸ Assign the *Context_menu_Storage_tank* that has been created to the bar graph element.

• To do this, select the bar graph element

• Select, in the properties, the **Runtime** group and the corresponding context menu from the `Context menu` drop-down list (in this example: *Context_menu_Storage_tank*)

17. Scripts

Sometimes it is necessary to combine several functions into one work step.

Scripts combine several functions for this purpose. The place of execution depends on the settings of the **Execute script** function.

Simply put, the following steps are necessary for the use of scripts:

▸ Create a script

▸ Fill the script with functions

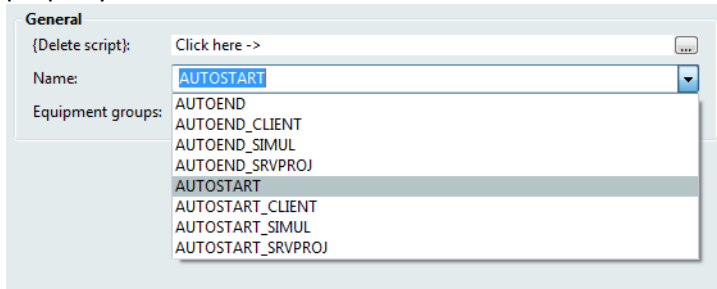▸ Configure function when the script is called up

> 💡 **Information**
>
> *The parameter settings of the script function overwrite the settings of the individual functions. When the functions are used individually, the parameter settings of the function remain.*

17.1 Creating scripts

Carry out the following steps to create a script:

▸ Switch to the 'Functions' node in the project manager.

▸ Open this with **[+]**.

▸ Select **New script** in the detail list or the context menu.

▸ Name this script in the property window by selecting *AUTOSTART* from the drop-down list.
Note: This drop-down list is called up by clicking on the arrow on the right on th 'Name:'

property field.

▶ Create two further scripts, not using the drop-down list when setting the parameters for the `Name` property field. Instead, give it two independent names; in this example: *Tank* and *Line*.

> 💡 **Information**
>
> *If a script is executed at a certain time, such as when Runtime is started, Runtime logs on to the server etc., the name is to be selected from the drop-down list. The name is then shown to represent the time of execution.*

17.2   Adding functions to scripts

A script serves functions as a container.

▶ Fill the (already-configured) *AUTOSTART* script with functions. To do this, with the script selected, select `Add functions...` in the tool bar of the detail view or via the context menu. In the **function selection** dialog that opens, add the existing function or remove this by selecting the corresponding `Add` or `Remove` button.

▶ For this tutorial, add the following functions:

• set_value_50
Create a new function **Write set value** with the name *set_value_50.*
To set the parameters of this function, use the variable *Storage tank[1].Heating elements.Temperature sensor*, and the set value *50*.
Note: Ensure that, in the `Write set value` dialog, `Direct to hardware` is activated.

• load_menu
This function displays (already-configured) menus when Runtime is called up.

17.3 Practical tips scripts

Scripts are an important instrument in order to be able to combine several functions. They can be a very powerful tool when used correctly.

▶ Sometimes it makes sense to create scripts even if no script content has been configured. The script execution function can already be configured in the ongoing project configuration process.

▶ Open project configuration steps can also be expedited with open questions with the help of scripts.

▶ Empty functions can also be added to scripts and the parameters for these can be set later.

▶ Scripts can contain scripts. Scripts cannot be integrated into scripts. However if you combine script functions into a combined script, the same functionality can be achieved.

18. Project documentation

Document your project configuration carefully! This documentation makes it easier for you to troubleshoot or cooperate with project partners in the event of subsequent expansions. Thanks to zenon, this usually very laborious process can be efficiently implemented in a few steps.

Use the integrated wizard for your project documentation:

▶ To call up the wizard, select, in the `File` menu bar, the `Wizards ...` entry
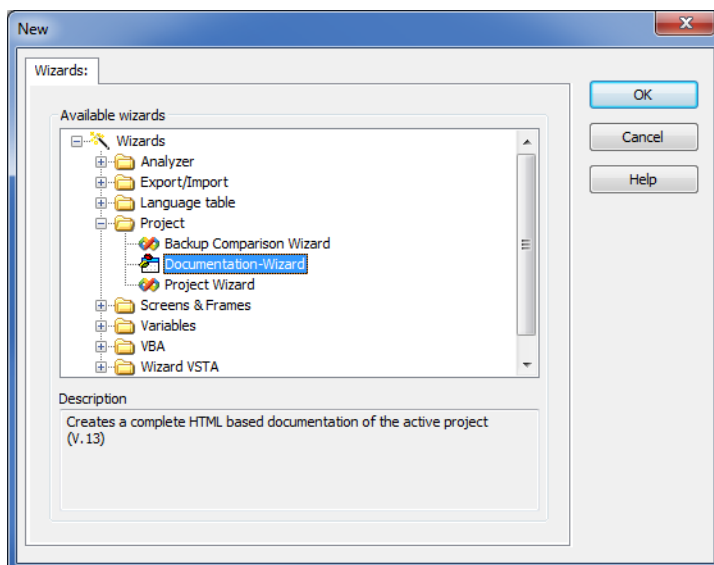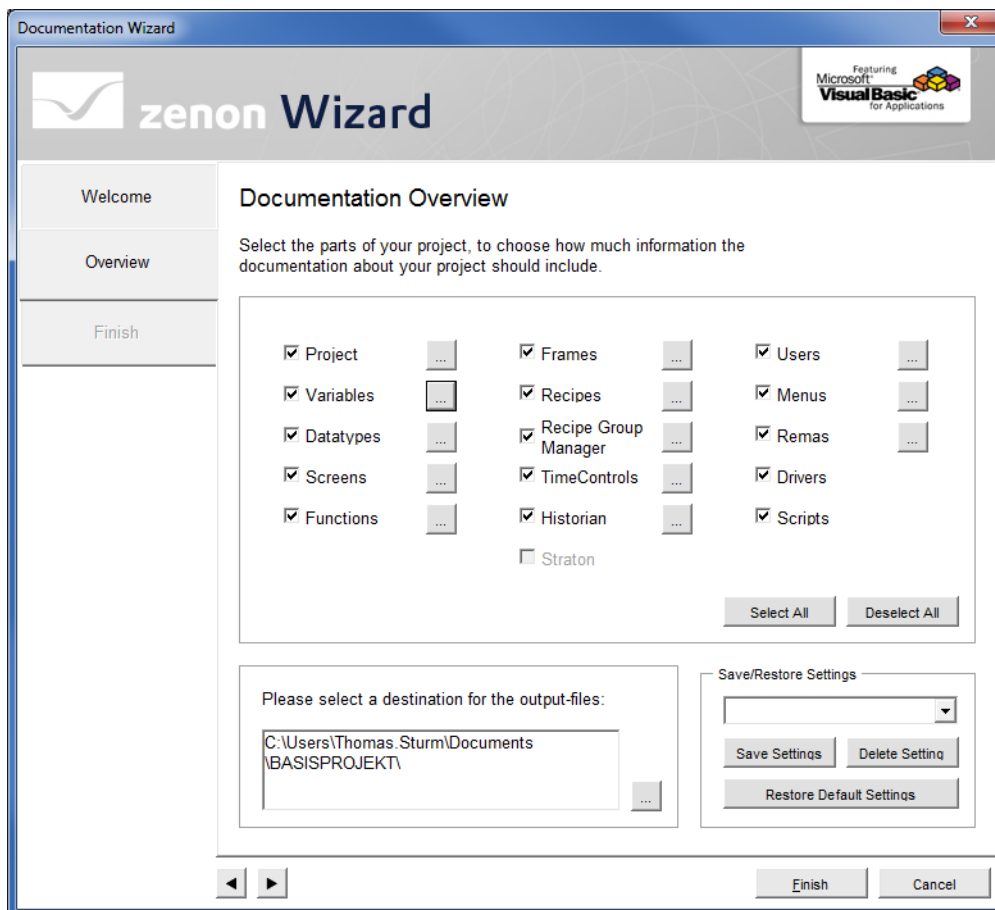
> **Information**
>
> *The wizard is only available in English.*

18.1 Selecting the documentation contents

This wizard leads you through the steps needed to create a HTML document of the active project. It is possible to define the modules that should be included in the documentation and those that should not.

The wizard can be changed:

▶ By deactivating the checkbox, it is possible to leave the complete contents of the project out of the documentation.

▶ Clicking the . . . button next to the contents of a project provides you with the respective detail view. Select further documentation contents here