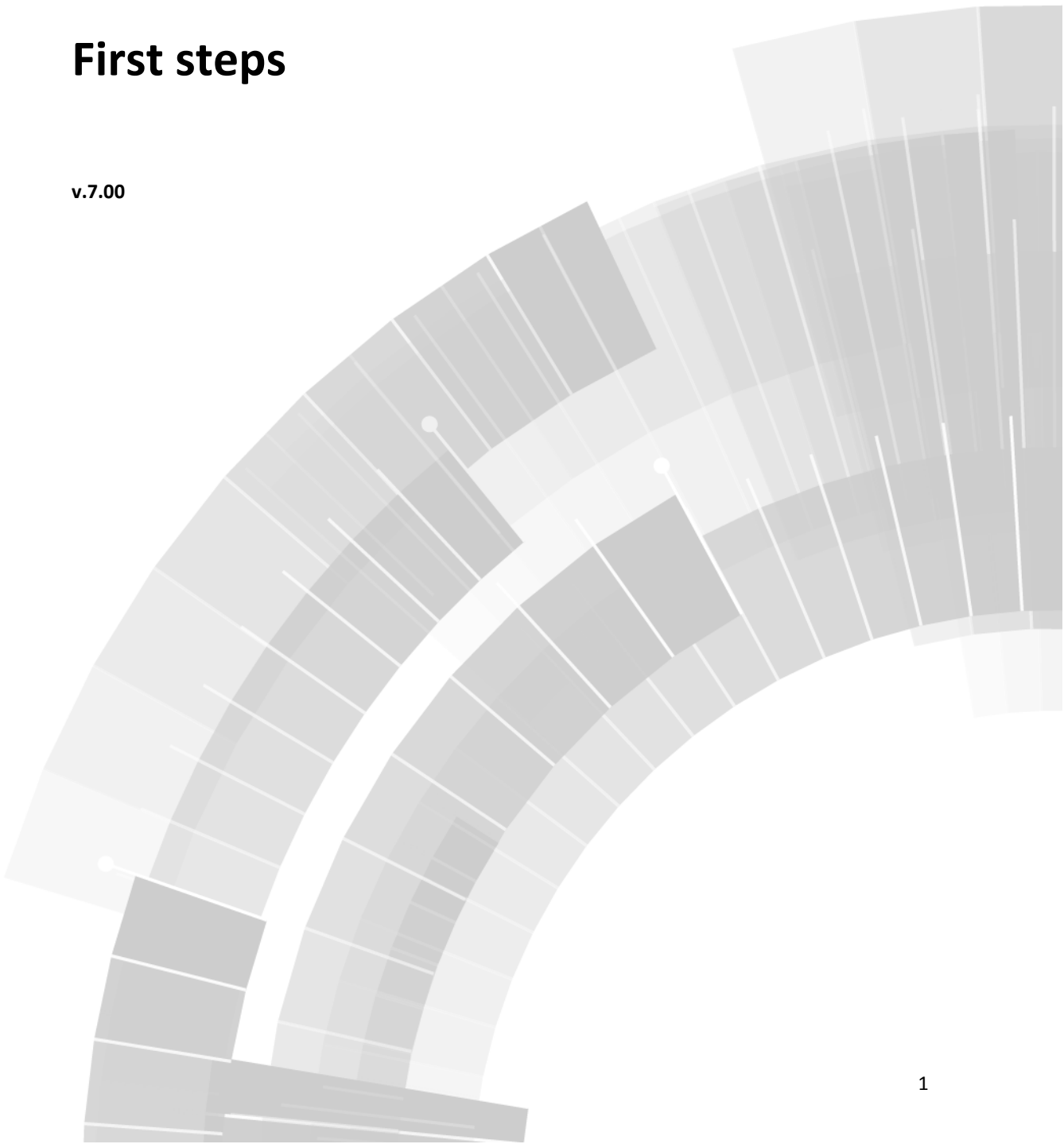


# zenon tutorial

## First steps

v.7.00





© 2012 Ing. Punzenberger COPA-DATA GmbH

All rights reserved.

Distribution and/or reproduction of this document or parts thereof in any form are permitted solely with the written permission of the company COPA-DATA. The technical data contained herein has been provided solely for informational purposes and is not legally binding. Subject to change, technical or otherwise.

# Contents

<b>1. Welcome to COPA-DATA help .....</b>	<b>4</b>
<b>2. First steps.....</b>	<b>4</b>
<b>3. A new project.....</b>	<b>5</b>
3.1 Creating a new workspace .....	5
3.2 Creating a new project.....	6
3.3 Configuration of the project .....	6
<b>4. Variables.....</b>	<b>9</b>
4.1 Drivers.....	9
4.2 Data types.....	11
4.2.1 Creating a new simple data type.....	11
4.2.2 Creating a new structure datatype.....	13
4.3 Variables.....	15
4.3.1 Creating a new simple variable .....	16
4.3.2 Creating a new structure variable .....	18
<b>5. Screens and functions.....</b>	<b>19</b>
5.1 Frames.....	19
5.1.1 Create new frame.....	20
5.2 Screens.....	21
5.2.1 Screen new .....	22
5.3 Screen elements.....	23
5.3.1 Vector elements .....	23
5.3.2 Dynamic elements .....	26
<b>6. Runtime (Online operation) .....</b>	<b>33</b>
6.1 Starting the Runtime .....	33
6.2 Stopping the Runtime .....	35

# 1. Welcome to COPA-DATA help

## GENERAL HELP

If you miss any information in this help chapter or have any suggestions for additions, please feel free to contact us via e-mail: [documentation@copadata.com](mailto:documentation@copadata.com) (<mailto:documentation@copadata.com>).

## PROJECT SUPPORT

If you have concrete questions relating to your project, please feel free to contact the support team via e-mail: [support@copadata.com](mailto:support@copadata.com) (<mailto:support@copadata.com>)

## LICENSES AND MODULES

If you realize that you need additional licenses or modules, please feel free to contact the sales team via e-mail: [sales@copadata.com](mailto:sales@copadata.com) (<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. You will learn here how to create a project and how to draw screens, which display the values from your PLC in dynamic elements.

## 3. A new project

In this step, you will learn what workspaces and projects are. We will create a workspace and, within that, an example project.

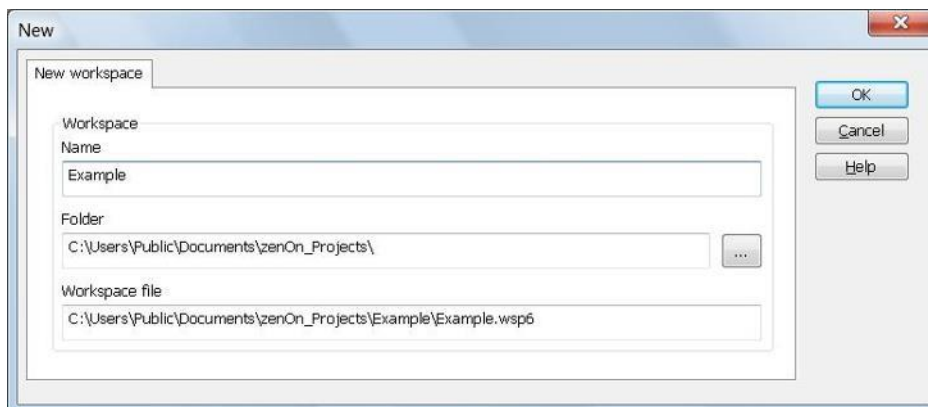
### 3.1 Creating a new workspace



#### Info

*We recommend to create a new subdirectory for each new workspace. This makes sure that all important files are in one directory.*

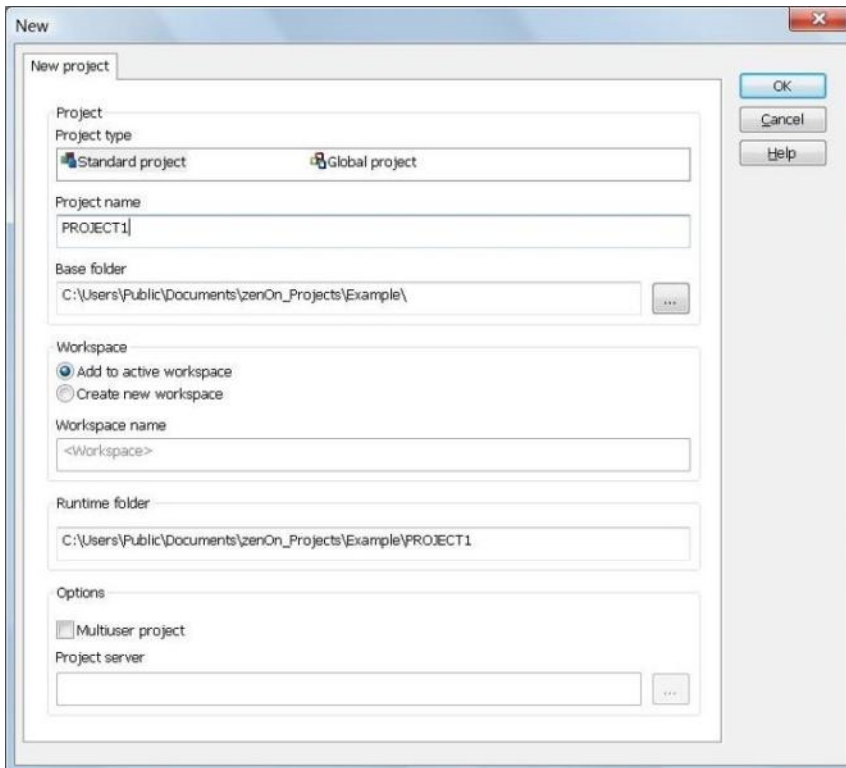
- ▶ Select the entry **Workspace new** in the menu *File*.
- ▶ Enter `C:\` as path and then enter the name 'EXAMPLE' for the workspace. If it does not exist, the directory `C:\EXAMPLE` is created after confirming the entries of the dialogbox with `OK` and the file `EXAMPLE.WSP6` is written into the directory.



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

## 3.2 Creating a new project

- ▶ Select the entry **Project new** in the menu File.



- ▶ Enter the project name 'PROJECT1' and accept the default path.

### Info

*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. In this example we will not use a wizard, so we close the dialog with **Cancel**.*

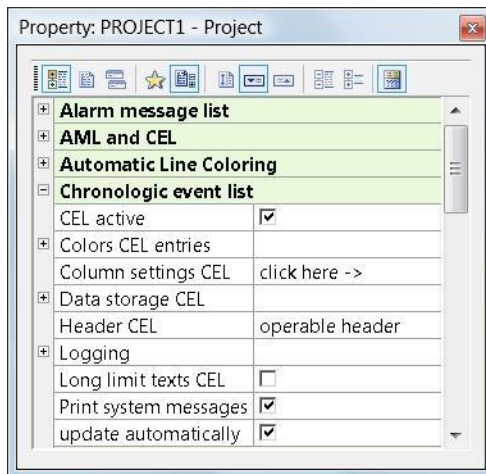
## 3.3 Configuration of the project

- ▶ 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 project **PROJECT1** in the detail view.

Now the properties of the project **PROJECT1** are displayed in the properties window and can be edited there.

We will use this properties window rather often in this training. 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 property windows allow you to show the properties in different views:



The individual symbols offer the following functions:

Symbol	Meaning
<b>Grouped</b>	The properties are combined into logical groups.
<b>All properties</b>	All properties are shown in a row.
<b>Dialog view</b>	The properties are displayed as dialog boxes.
<b>Show/hide favorites</b>	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.
<b>Show/hide all properties</b>	If the favorites are shown, you can hide all other properties with this symbol for a better overview.
<b>Sorted logically</b>	With this icon, the displayed properties are sorted according to their logical connectedness.
<b>Sort ascending</b>	With this icon, the displayed properties are sorted in alphabetically ascending order.
<b>Sort descending</b>	With this icon, the displayed properties are sorted in alphabetically descending order.
<b>Expand all</b>	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.
<b>Reduce all</b>	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.
<b>Display properties help</b>	A window with the description of the selected property is displayed. To close the help window, click on the <b>x</b> on the top right-hand corner of the window.



### Info

*With the context menu of the properties window you can define which properties should be displayed in the favorites.*

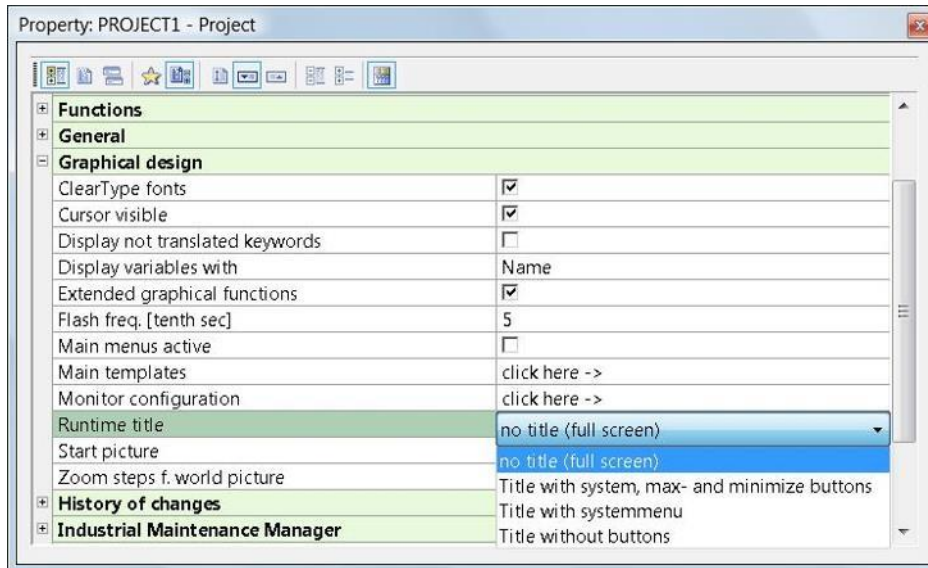
## RUNTIME VIEW

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

- ▶ Switch to the grouped view.



- ▶ Open the section **Graphical design**.



- ▶ Change the property `Runtime title` to `not titel (full screen`.

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

## 4. Variables

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

For that, we are going to create a driver and then define variables for this driver.

### 4.1 Drivers

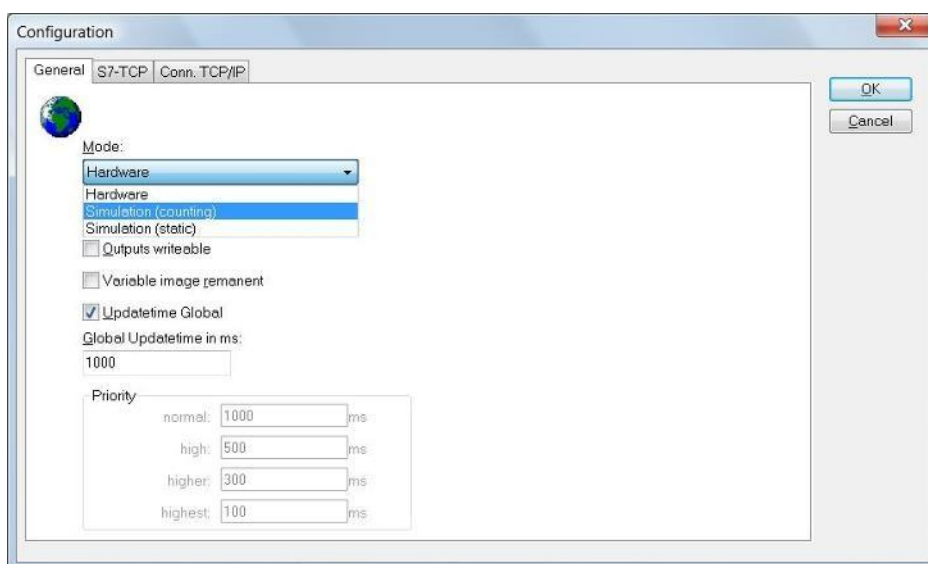
To communicate with a data source (PLC field bus, etc.) it is necessary to connect to a driver. Depending on the needs of the project, the necessary drivers (depending on the PLC) and their process variables have to be created.

- ▶ Open the node `variables` in the project manager.
- ▶ Activate the context menu of the entry `Drivers` with the right mouse button.
- ▶ Select the menu entry `Driver new`.

The driver selection dialog opens now.

- ▶ Scroll down in the list of Available drivers until you reach the folder Siemens.
- ▶ Choose the driver S7 TCP-IP. The entry is put in the field Driver name as shown in the dialog box **Definition of driver**. Close the dialog by confirming the selection with **OK**.

Now the dialog for the configuration of the driver opens.



- ▶ Set the Mode to Simulation (counting).

In the mode `Hardware`, zenon would immediately try to connect to the PLC when the Runtime is started. As we do not have a PLC at the moment, zenon would show errors for all values if we chose the `Hardware` mode.

- ▶ Switch off Update time global.

By setting this, we can assign one of four different update times to every single variable during the definition of variables. These four update times can be set in the lower part of this dialog, under **Priority**

All other tabs of this dialog are driver-specific, i.e. different according to the chosen driver. As we do not have a PLC at the moment, we do not have to regard these settings in this example.

## 4.2 Data types

Variable on the one hand are based on driver object types and on the other hand on data types.

The driver object types depend on the selected driver.

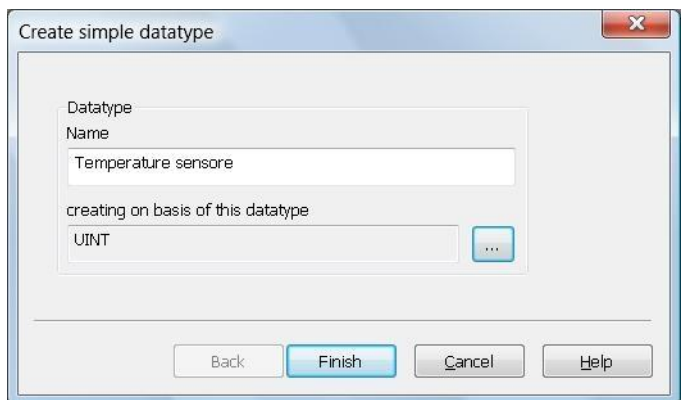
Generally speaking, data types are independent of the driver. But not all drivers support object types of all data types. When you are asked to select a data type, only the data types supported by your driver will be available.

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.

### 4.2.1 Creating a new simple data type

- ▶ Open the node `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`.



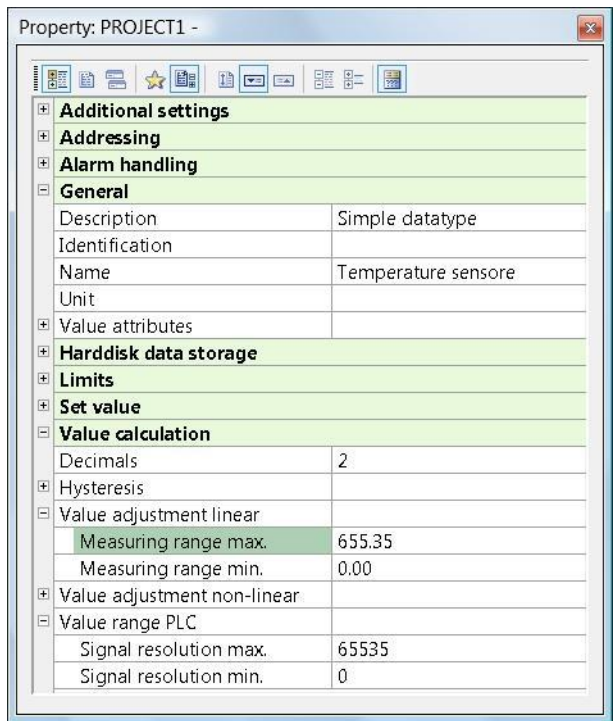
- ▶ As a name for the new simple datatype enter `Temperature sensor`.
- ▶ Select `UINT` as the basic datatype.

With `Finish` the datatype is created and now is available in the list of datatypes.

- ▶ Select the datatype `Temperature sensor` in the list.

In the property window, the properties of the data type `Temperature sensor` are now displayed and we can make all the necessary changes.

- Open the sections `General` and `Value calculation`.



- Change the properties `Unit`, `Decimals`, `Linear value adjustment` and `Value range PLC` of the datatype as shown in the illustration above.

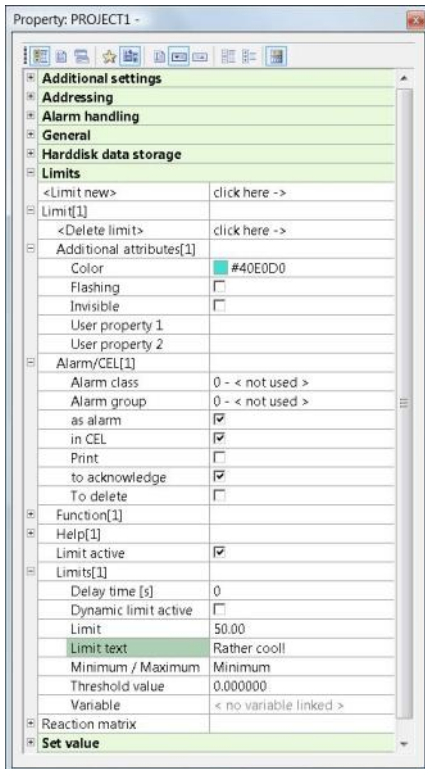
Value range PLC	digital value from the temperature sensor, in this case 16 bit resolution in the PLC
Value adjustment linear	analog value from temperature sensor, according to our setting, this corresponds to Min. 0,00 °C / Max. 655,35 °C in the project.

We now will define limits for the datatype `Temperature sensor`.

- Open the section `Limits` in the properties window.
- Select the entry `<Limit new>`.

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

- ▶ Open the section Limit[1].



- ▶ Change the properties of the limit as shown in the illustration above.

### Example

Create three more limits for the datatype Temperature sensor:

Limit 2: 100; MIN; dark blue; no alarm

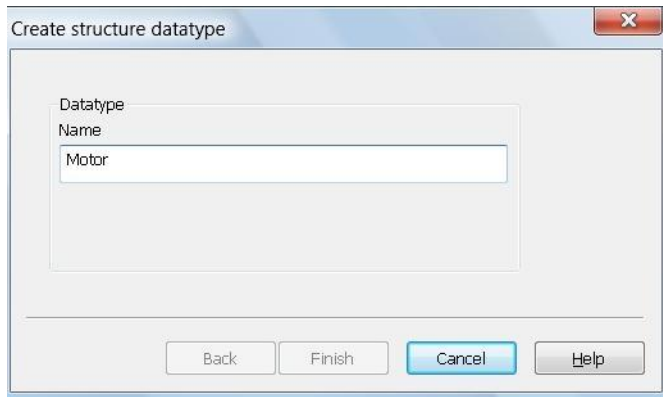
Limit 3: 400; MAX; dark red; no alarm

Limit 4: 500; MAX; light red; alarm

## 4.2.2 Creating a new structure datatype

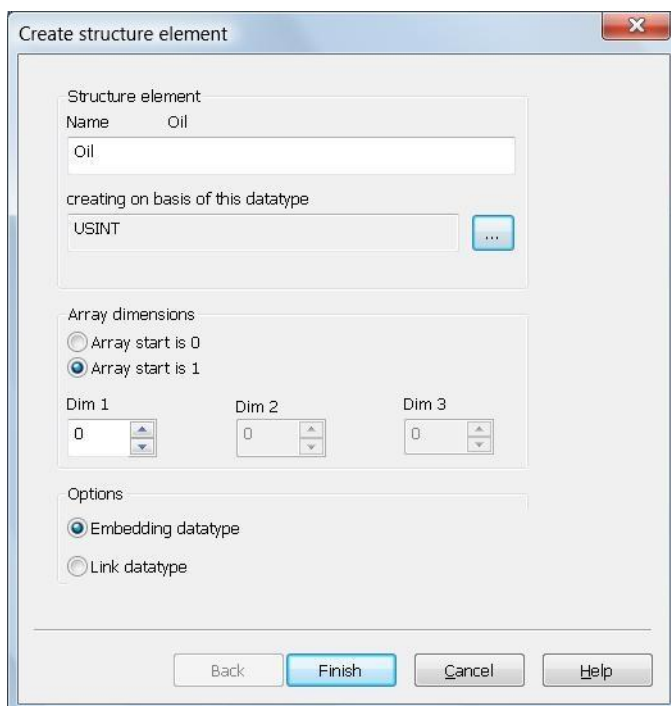
- ▶ Open the entry `variables` in the Project Manager.

- ▶ Activate the context menu of the entry `Datatypes` with the right mouse button.
- ▶ Select the menu entry `Structure datatype new`.



- ▶ As a name for the new structure datatype enter `Motor`.

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.



- ▶ As a name for the new structure element enter `Oil`.
- ▶ Select `USINT` as the basic datatype. The data type should be embedded.

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.

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

Now we will create further structure elements for this data type.

- ▶ Activate the context menu of the structure data type "Motor" with the right mouse button.
- ▶ Select the menu entry **Structure element new...**

#### Example

Create another structure element for the structure datatype Motor:

*On/off: BOOL; embedded*

*Temperature: Temperature sensor; linked*

## 4.3 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.

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

#### Info

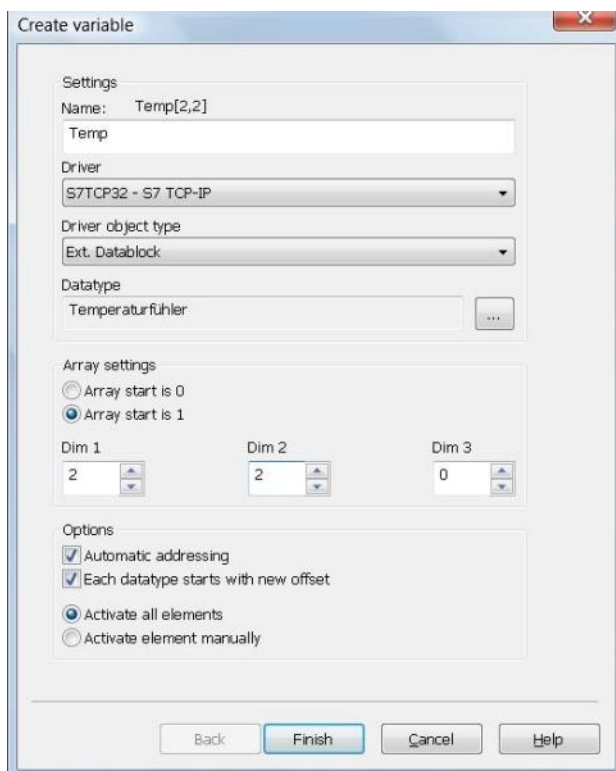
*Many zenon drivers allow the import of variables from the PLC program or directly from the PLC, which means that you do not have to create the variables manually. You will find the necessary information for the variable import in the according driver documentation.*

Because we have neither a PLC nor an S7 project in our example project, we will create the variables manually.

### 4.3.1 Creating a new simple variable

- ▶ In the project manager, activate the context menu of the entry Variables by a right click.
- ▶ Select the menu entry **variable new**.

The dialog for creating variables opens now.



- ▶ As a name enter Temp.
- ▶ Choose the driver S7TCP32 - S7 TCP-IP
- ▶ Select Ext. datablock as the driver object type.
- ▶ Select the datatype Temperature sensor.

With these settings, you would create a single variable. For creating multiple variables of the same type, some changes in the **Array Settings** are necessary.

- ▶ Set the array dimensions 1 and 2 both to the value 2.



- ▶ 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 our datatype `Temperature sensor` are pre-defined in the variables. The properties of the selected variable can be checked and changed in the properties window.

As we changed the array dimensions, not only one but four variables have been created. With our settings, these variables automatically get their addresses, i.e. we only have to define a `start offset`. zenon calculates the remaining addresses automatically.

- ▶ Open the section **variables**.
- ▶ Select the entry **Temp**.
- ▶ In the section Addressing set the property `Start offset` to the value 11.

This results in the following addresses:

Temp[1,1]	Offset 11
Temp[1,2]	Offset 13
Temp[2,1]	Offset 15
Temp[2,2]	Offset 17

The property `last used offset` accordingly reads: 18/7 (Offset 18 / Bit 7).

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

Status	Name	Driver	Identification	Unit	Net address	Data block	Offset	Bit
[-]	Temp	S7TCP32 - ...			0	0	0	
[-]	Temp[1,1]	S7TCP32 - ...		°C	0	0	11	
[-]	Temp[1,2]	S7TCP32 - ...		°C	0	0	13	
[-]	Temp[2,1]	S7TCP32 - ...		°C	0	0	15	
[-]	Temp[2,2]	S7TCP32 - ...		°C	0	0	17	



### Info

*Always use logical names for the process variables in order to keep a better overview.*

### 4.3.2 Creating a new structure variable

Now we will create more variables, but this time for a structure data type. Therefore proceed as described above.

- ▶ As a name enter `Motor`.
- ▶ Select the datatype `Motor`.
- ▶ Set Dim 1 to the value 4 under Array settings.



- ▶ Activate the option Automatic addressing.
- ▶ Leave the options 'Each data type starts with new offset' and 'Activate all elements' as proposed by the default.
- ▶ Select an Ext. datablock as the driver object type.
- ▶ Set the Start offset to 21.

Now the variables for four motors are created (array dim 1=4), where each of the four motors consists of three variables.

Your variable list in the detail view of the project manager should look like the illustration below:

Status	Name	Driver	Identification	Unit	Net address	Data block	Offset	Bit
[-]	Temp	S7TCP32 - ...			0	0	0	
[-]	Temp[1,1]	S7TCP32 - ...		°C	0	0	11	
[-]	Temp[1,2]	S7TCP32 - ...		°C	0	0	13	
[-]	Temp[2,1]	S7TCP32 - ...		°C	0	0	15	
[-]	Temp[2,2]	S7TCP32 - ...		°C	0	0	17	
[-]	Motor	S7TCP32 - ...			0	0	0	
[-]	Motor[1].Oil	S7TCP32 - ...			0	0	21	
[-]	Motor[1].On...	S7TCP32 - ...			0	0	22	
[-]	Motor[1].Te...	S7TCP32 - ...		°C	0	0	23	
[-]	Motor[2].Oil	S7TCP32 - ...			0	0	25	
[-]	Motor[2].On...	S7TCP32 - ...			0	0	26	
[-]	Motor[2].Te...	S7TCP32 - ...		°C	0	0	27	
[-]	Motor[3].Oil	S7TCP32 - ...			0	0	29	
[-]	Motor[3].On...	S7TCP32 - ...			0	0	30	
[-]	Motor[3].Te...	S7TCP32 - ...		°C	0	0	31	
[-]	Motor[4].Oil	S7TCP32 - ...			0	0	33	
[-]	Motor[4].On...	S7TCP32 - ...			0	0	34	
[-]	Motor[4].Te...	S7TCP32 - ...		°C	0	0	35	

## 5. Screens and functions

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

### 5.1 Frames

A frame has to be created before we can draw our first screen. Every screen is based on a frame.

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 monitor

And so a general layout is created. This offers some advantages like:

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.



### Info

*Always create at least two frames.*

*1 frame for preprocess screens*

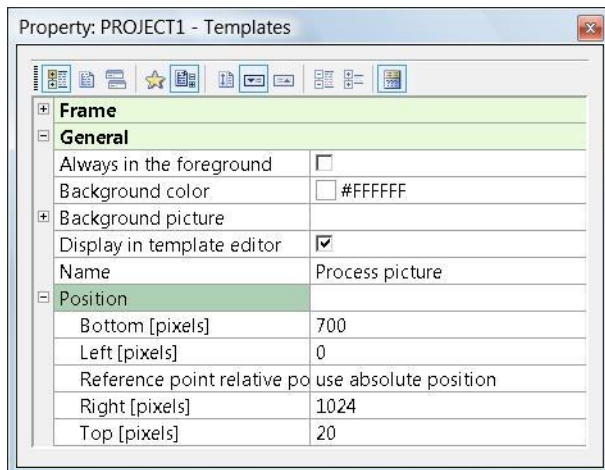
*1 frame for button bars*

## 5.1.1 Create new frame

- ▶ 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.

A frame with the name `Frame_0` is automatically created. You can change the properties of the frame in the property window.

The size of the frame is set to the current screen resolution by default. We now change these settings, so that some space is left over on the top and on the bottom of the screen. We need the space at the top for the status line and the one at the bottom for creating a button bar later. For our example, we use a screen resolution of 1024\*768.



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

### Example

Create a frame with the name *Button Bar* for the button bars. The size of the frame should range from left/top 0/700 to right/bottom 1024/768.

## 5.2 Screens

A screen is a window with special predefined properties. Each screen has to be based on a frame.

 **Info**

Nevertheless, you can 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.*

### 5.2.1 Screen new

- ▶ Select the branch **screens** in the project manager .
- ▶ With **Screen new** in the context menu you create a new screen.

A screen with the name Screen 0 will be created. You can change the properties of the screen in the property window.

- ▶ In the section **General** enter the screen name `Start Pic` and take care that the screen has the **Screen type Standard**.
- ▶ In the section **Frame** make sure, that the screen is linked to the frame **Process screen**.
- ▶ Open this new screen with a doubleclick on the screen name in the detail view of the project manager or by opening the context menu with a right-click and selecting the entry **Open screen**.

 **Info**

As this is the first screen that we have saved, the screen name will be added automatically to the project properties under the property `start pic` in the group **Graphical design**.

*This screen will be opened automatically when the Runtime is started.*

### Example

Now create a screen with the name Hall. This screen should also be linked to the frame Process screen.

*Now create a screen with the name BB-Start. Link the frame Button bar to this screen.*

## 5.3 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 variable).

### 5.3.1 Vector elements

Vector elements, as opposed to dynamic elements, which we will get to know later, do not change their appearance in the Runtime. They are used for graphical background information.

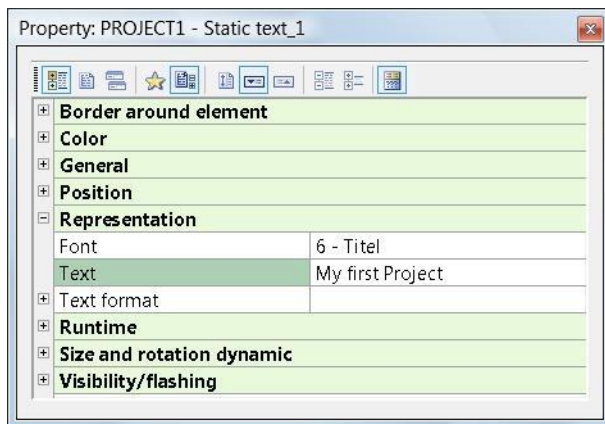
#### Static text

We are now going to enter text into our start screen.

- ▶ Open the menu **Vect. Elements**.
- ▶ Select the vector element **Static text**.
- ▶ In the screen Start pic, press the left mouse button on the desired location and hold it while pulling up a square.

You can now adjust the properties of the element in the property window. If you want to change the properties later on, you only have to select the element in the screen - by clicking on it with the left mouse button. Then the properties are again available in the properties window.

This procedure is the same for all elements.

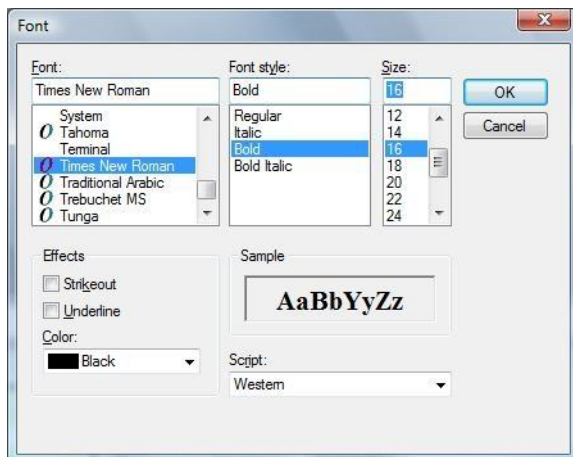


- ▶ Enter 'My first project' in the field Representation/Text.

Now we will create a new font for our text element.

- ▶ Select the **Screens/Fonts** in the Project Manager.
- ▶ Open the context menu in the detail view and select the entry **Font new**.

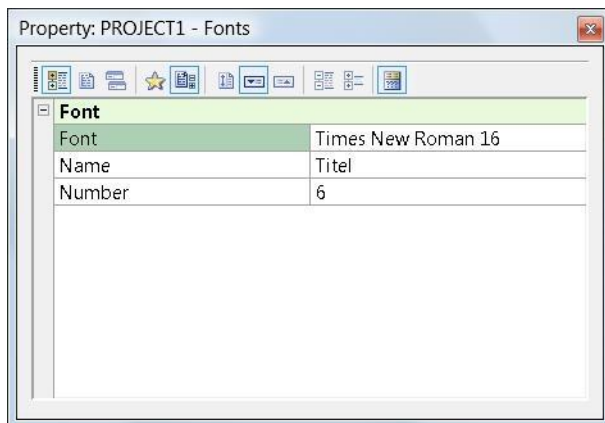
The Windows standard dialog for the definition of fonts opens now.



- ▶ Now select Times New Roman bold 36 and confirm the setting with **OK**.



The detail view of the Project Manager now lists New font 1 and the properties window shows the properties of the new font.



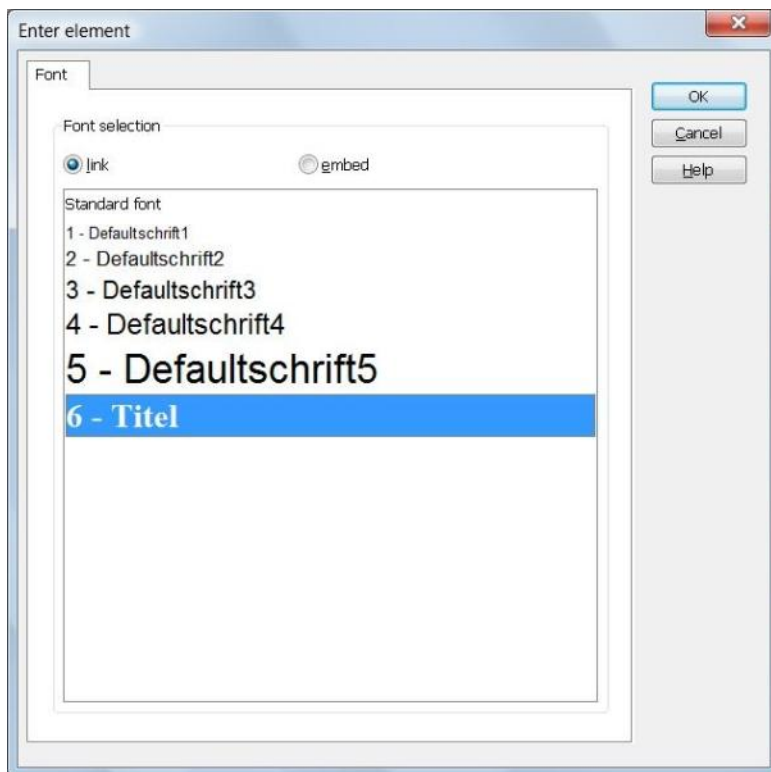
- ▶ Change the name of the font to `Titel`.

The fonts selected here can be used in all elements that include text.

We now will use this font in our text element.

- ▶ Select the text element in the starting screen and open the section `Font` in the properties window.
- ▶ Select the right column of the property `Font`.

The dialog for the font selection opens.



- ▶ Select the font Title.
- ▶ Click on the icon Save screen or select the entry **save screen** from the context menu of the screen.

### 5.3.2 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.

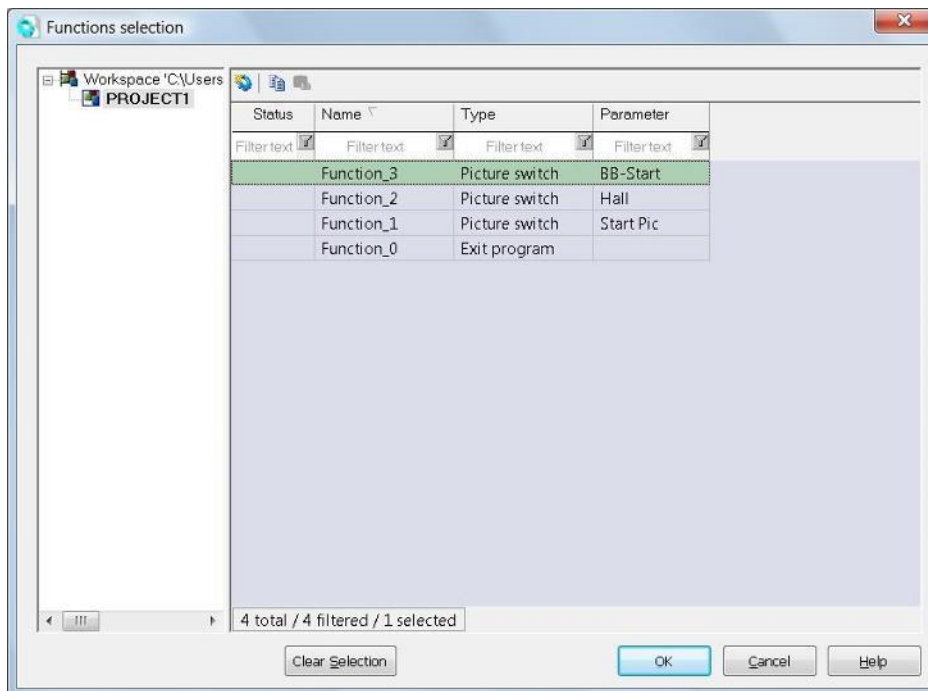
#### **Text button**

First we will create buttons for our button bar BB-Start to switch between the screens Start Pic and Hall.

- ▶ Open the screen BB-Start.

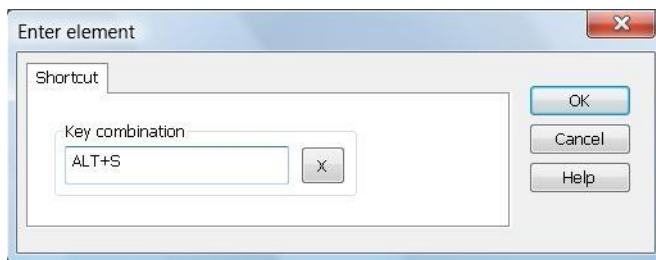
- ▶ Select the command **Text Button** from the menu Dyn. Elements.
- ▶ With the left mouse button pressed down you now draw a dynamic element on the screen BB-Start (ca. 1.5 cm high and 3 cm wide).

The dialog box for selecting functions opens automatically after you release the mouse button.



- ▶ Select the function Screen switch - Start Pic.
- ▶ Open the section Representation in the properties window.
- ▶ For Line 1 enter the text Start Pic and for Line 2 the text Alt+S.
- ▶ Click on the right column of the property *Runtime/Operation/Key combination*.

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-S by clicking into the input area and pressing that key combination.
- ▶ Confirm the settings with **OK**.

### Example

Create a corresponding button for the screen Hall with the text Hall 1 and Alt 1 and the according key combination.

*Create another button for the function `Exit program` with the text `Exit` and the key combination `Alt+E`.*

### 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 you want to change the size or any other attribute of the font at a later moment the changes will then be updated for all buttons automatically.*

## Screen Functions

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

- ▶ Select the node `screens` in the project manager.

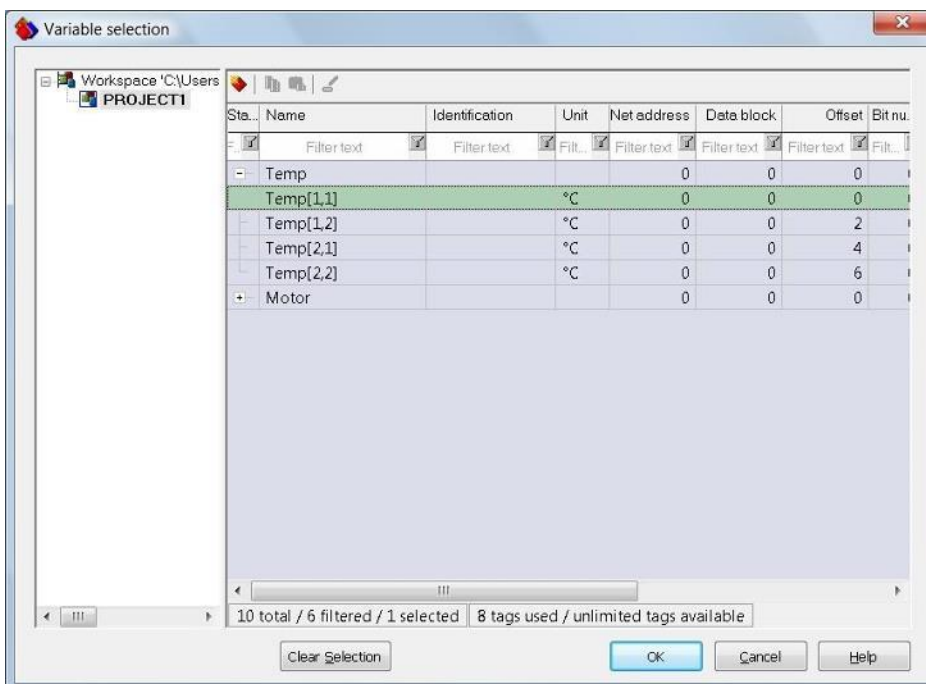
- ▶ In the detail view of the Project Manager select the screen `Start Pic`.
- ▶ Open the section **Execution** in the properties window.
- ▶ As Start function, select the function Screen switch: BB-Start from the list.

Now, whenever the screen `Start pic` is opened, the screen `BB-Start` will also be opened automatically.

## Numerical value

- ▶ Open the screen `Start Pic`.
- ▶ Select the entry Numerical value from the menu `Dyn. Elements`.
- ▶ With the left mouse button pressed down you now draw a dynamic element on the screen "HALL" (about 2 cm high and 5 cm wide).

Now the dialog for the variable selection opens.



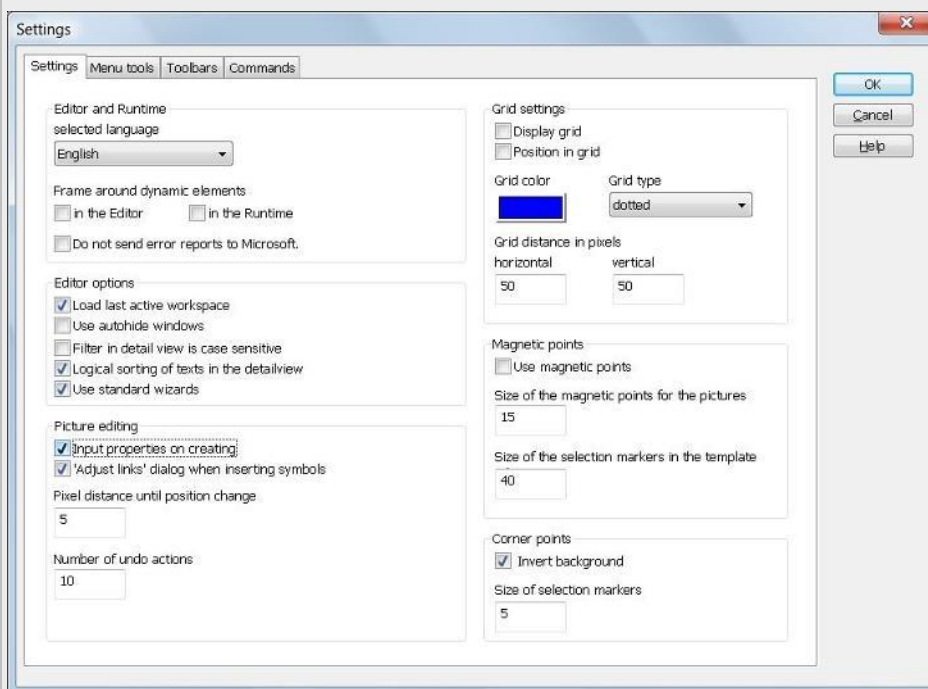
- ▶ In the dialog box `Variable selection`, highlight the variable `Temp[1.1]` and then press the button **OK**.

### Info

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 option **Input properties after creating objects** in the group **Element editing** and then press the button **OK**.

*You can also open the dialog with a doubleclick on the element.*

In order to change the size of a dynamic element you click on one of its markers (the cursor is a double arrow, you hold the left mouse button pressed down and by moving the mouse change the size of the element).

In order to move a dynamic element, click on it (the cursor is an arrow cross), hold the left mouse button pressed down 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.

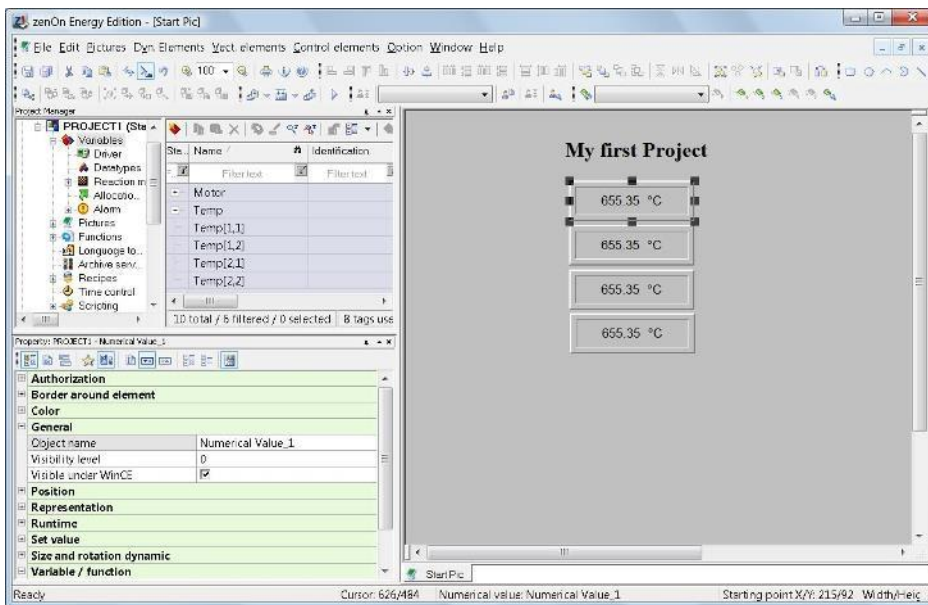
### Example

Now create further numerical value elements for the variables *Temp[1.2]*, *Temp[2.1]* and *Temp[2.2]*.

### Info

The simplest way to do this is as follows:

Select the element and copy it with *Ctrl-C* to the clipboard, then paste it three times with *Ctrl-V*. Now you can drag the desired variables on the elements with the mouse.



## Bar Graph

- ▶ Open the screen **Hall 1**.
- ▶ Select the command **Bar Graph** from the menu **Dyn. Elements**.

- ▶ With the left mouse button pressed down you now draw a dynamic element on the screen „HALL“ (ca.7 cm high and 3 cm wide).
- ▶ In the dialogbox **variable selection**, highlight the variable `Temp [ 1 , 1 ]` and then press the button **OK**.
- ▶ In the properties window under **Colors/Colors static** set the **Bar color** to **green**.

### Indicating Instrument

- ▶ Select the command **Indicating instrument** from the menu **Dyn. Elements**.
- ▶ With the left mouse button pressed down you now draw a dynamic element on the Start Pic (ca.15 cm high and 20 cm wide).
- ▶ Select the variable `Temp[1,2]` in the dialogbox **variable selection** and then press the button **OK**.
- ▶ In the properties window open the section **Representation** and then **Scale**.
- ▶ Set the **Main ticks** to 100 and the **Sub ticks** to 20.

So there will be a long tick with the value at every 100 and a short one with out value every 20.

### Trend Graph

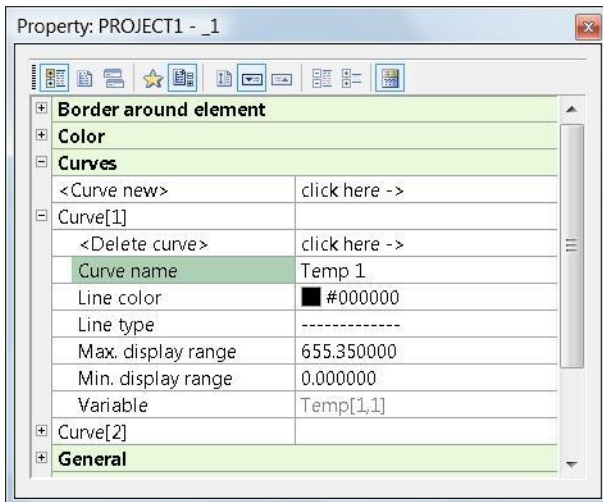
- ▶ Select the command **Trend curve** from the menu **Dyn. Elements**.
- ▶ With the left mouse button pressed down you now draw a dynamic element on the Start Pic (ca.15 cm high and 20 cm wide).

Now the dialog for the **variable selection** opens.

- ▶ Doubleclick the variables "`Temp[1,1]`" and "`Temp[1,2]`" in the dialogbox '**variable selection**'.
- ▶ Then press the button **OK**.



In the properties window you will now find the new section Curves.



- ▶ In the properties window, open the section Curves and then Curve[1].
- ▶ Change the Curve name to Temp 1.
- ▶ For Curve[2] change the text to Temp 2.
- ▶ Set the property Time of representation in the section Representation to 10 minutes.

## 6. Runtime (Online operation)

Now it is time to see our example project in the Runtime. In this step, you will learn how to start and stop the Runtime.

### 6.1 Starting the Runtime

The Runtime can be started in three ways:

1. By clicking on the button **Start Runtime**.

2. In the start menu in the folder *Programs/COPA-DATA/zenon700* with the entry **zenon Logic Runtime**.
3. With the key F5 from the Editor.

After the Runtime is started, two screens are loaded:

First the screen *Start* (because it was automatically entered in the configuration as the *Start* screen) and then the screen *BL-Start* (because it is opened via the *Start* function of the screen *Start*).

Now we will have a look at how to find your way in an existing Runtime project.

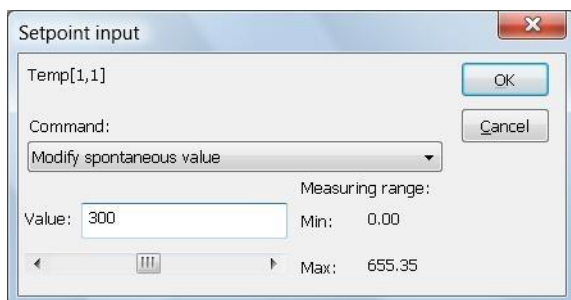
- ▶ Click into an empty area of the screen with the right mouse button.

If you keep the mouse button pressed long enough the name of the screen appears at the position of the mouse pointer.

- ▶ Now click on the numerical value element for the variable *Mode* with the right mouse button.

On the left side above the element, the name of the linked variable will now be displayed.

- ▶ Now click on the numerical value element for the variable *Temp[1,1]* with the left mouse button.



The standard dialog for settings values opens and allows you to change the value of this variable.

- ▶ Set the value of the variable *Temp[1,1]* to 300.00 and close the dialog with **OK**.
- ▶ Set the value of the variable *Temp[1,2]* to 350.00 and close the dialog with **OK**.

With the buttons *Start Pic* and *Hall 1* in the button bar you can switch between the two process screens.

- ▶ Switch to the screen *Hall 1* and watch the behaviour of the dynamic elements there.

The driver is set to the counting simulation mode and the bar of the bar graph is moving.

## 6.2 Stopping the Runtime

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



### Info

*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.*