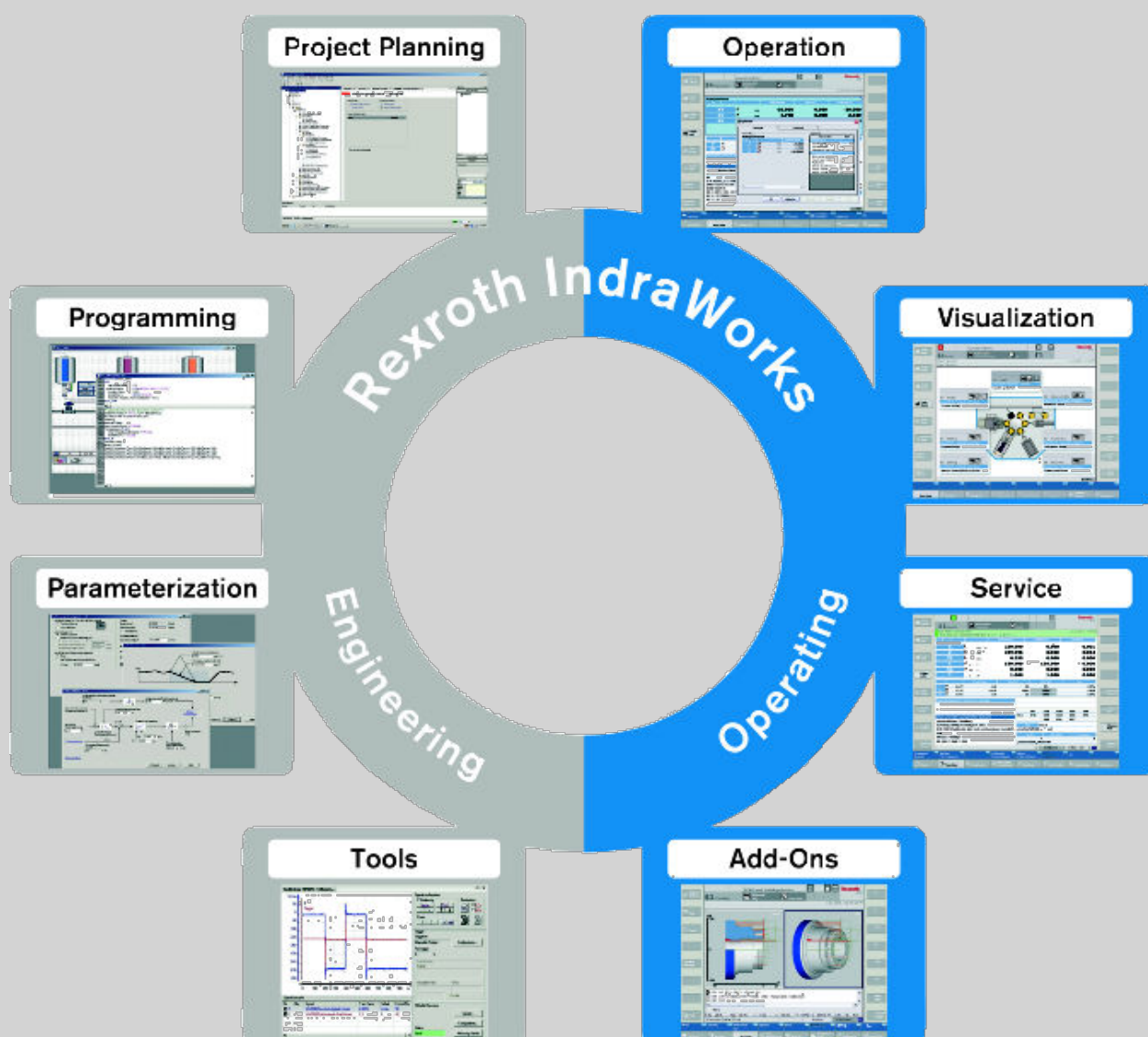


IndraWorks 14VRS

WebComposer Functional Description

Reference Book
R911376836

Edition 04



Title	IndraWorks 14VRS WebComposer Functional Description
Type of Documentation	Reference Book
Document Typecode	DOK-IWORKS-WEBCOMP*V14-RE04-EN-E
Internal File Reference	RS-7ac1f0760569e3120a347ea50054191f-4-en-US-3
Change Record	Edition 04, 2020-02 See tab. 1-1 "Change Record" on page 1
Copyright	© Bosch Rexroth AG 2020 All rights reserved, also regarding any disposal, exploitation, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.
Liability	The specified data is intended for product description purposes only and shall not be deemed to be a guaranteed characteristic unless expressly stipulated in the contract. All rights are reserved with respect to the content of this documentation and the availability of the product.
Editorial Department	Engineering Software and Control Firmware - Visualization and Diagnostics, UwLi (MS/MePe)

Table of Contents

	Page
1 About this documentation.....	1
1.1 Validity of the documentation.....	1
1.2 Required and supplementing documentation IndraWorks.....	2
1.2.1 Engineering.....	2
1.2.2 Engineering & visualization.....	4
1.2.3 Visualization.....	4
1.3 Using safety instructions.....	4
1.3.1 Structure of the safety instructions.....	4
1.3.2 Explaining signal words and safety alert symbol.....	4
1.3.3 Symbols used.....	6
1.3.4 Explaining the signal alert symbol on the device.....	6
1.4 Names and abbreviations.....	6
1.5 Customer feedback.....	6
 2 Important instructions on use.....	 7
2.1 Intended use.....	7
2.1.1 Introduction.....	7
2.1.2 Areas of use and application.....	7
2.1.3 Unintended use.....	7
 3 User interface.....	 9
3.1 Overview.....	9
3.2 Project elements and their functions.....	9
3.2.1 WebComposer commissioning visualization.....	9
3.2.2 Global data.....	9
3.2.3 Variables and VarDef.....	10
3.2.4 Variables and TextDef.....	10
3.2.5 Scripts, main functions and functions.....	10
3.2.6 Themes.....	11
3.2.7 Images.....	12
3.2.8 Screen.....	12
3.2.9 Script.....	13
3.2.10 Variables.....	13
3.2.11 Folder.....	13
3.2.12 Templates.....	14
3.3 Editors.....	14
3.3.1 Screen editor.....	14
3.3.2 Script editor.....	16
3.3.3 Text editor.....	17
3.3.4 Theme editor.....	17
3.3.5 Event editor.....	18
3.4 WebComposer widget library.....	18
3.5 WebComposer widget attribute.....	19

Table of Contents

	Page
3.5.1 General.....	21
3.5.2 Position.....	21
3.5.3 Background style.....	22
3.5.4 Communication settings.....	22
3.5.5 Grid.....	22
3.5.6 Look and feel.....	23
3.5.7 Text properties.....	24
3.5.8 Dynamic fill color.....	25
3.5.9 Dynamic motion.....	26
3.5.10 Dynamic size.....	27
3.5.11 Dynamic rotation.....	27
3.5.12 Dynamic screen.....	27
3.5.13 Events.....	28
3.5.14 Dynamic attribute.....	28
3.5.15 Widget attribute.....	28
3.6 Process variables.....	28
3.7 Data and functions.....	30
3.7.1 Global variables.....	30
3.7.2 Global main functions.....	30
3.7.3 Global functions.....	31
3.7.4 Screen variables.....	32
3.7.5 Screen scripts.....	32
3.7.6 Internal variable.....	32
3.7.7 Find & Replace function.....	33
3.7.8 Validation function.....	36
3.7.9 Import/export function.....	38
3.7.10 Selecting customized screens.....	38
3.7.11 3S visualization import.....	40
3.8 Preview.....	40
4 Use cases.....	41
4.1 Creating a WebComposer commissioning visualization.....	41
4.2 Configuring the connection to the control.....	43
4.3 Position widgets on a screen.....	43
4.4 Using screens in screens.....	45
4.4.1 Referencing screens.....	45
4.4.2 Editing referenced screens.....	46
4.4.3 User example axis overview.....	46
4.4.4 User example PLC array structure element.....	48
4.5 Linking widget attributes to variables.....	50
4.6 Visualizing data in the internal or external web browser.....	52
4.7 Triggering of events.....	52
4.7.1 Writing variables about events.....	53
4.7.2 Switching screens via events.....	55
4.7.3 Starting scripts via events.....	57
4.8 Creating custom screens.....	58

	Page
4.8.1 General.....	58
4.8.2 Configuration in IndraWorks Engineering.....	59
4.8.3 WebAssistant.....	62
4.9 Creating global, localized texts.....	64
4.10 Formatting global, localized texts.....	65
4.11 Using global, localized texts in widgets.....	65
4.12 Translating global texts externally.....	69
4.13 Debugging JavaScript code.....	69
4.14 3S visualization import.....	69
4.15 Creating a new theme.....	71
4.16 Taking a widget from a theme.....	74
4.17 Checking changes immediately.....	75
5 FAQ.....	77
5.1 The control does not return any data - What can I do?.....	77
5.2 Why are SVG graphics not displayed on customized pages?.....	77
5.3 Why is a transparent screen not displayed as transparent?.....	77
5.4 Why are WebComposer images empty upon opening (white screen)?.....	77
5.5 Why are the referenced screens not working?.....	78
5.6 Why is a valid variable identified as "BadNodeID"?.....	78
5.7 How can I can access the value of a PLC variable in the JavaScript code of a screen?.....	78
6 Attachment: Widget attribute.....	79
6.1 Bit list.....	79
6.1.1 General.....	79
6.1.2 Position.....	79
6.1.3 Style.....	79
6.1.4 Text properties.....	80
6.1.5 Dynamic fill color.....	81
6.1.6 Dynamic movements.....	82
6.1.7 Dynamic size.....	84
6.1.8 Dynamic rotation.....	86
6.1.9 Events.....	86
6.2 Button.....	88
6.2.1 General.....	88
6.2.2 Position.....	88
6.2.3 Style.....	89
6.2.4 Text properties.....	90
6.2.5 Dynamic fill color.....	92
6.2.6 Dynamic movements.....	93
6.2.7 Dynamic size.....	95
6.2.8 Dynamic rotation.....	97
6.2.9 Events.....	97
6.3 Chart.....	99
6.3.1 General.....	99

Table of Contents

	Page
6.3.2	Position..... 99
6.3.3	Style..... 100
6.3.4	Text properties..... 100
6.3.5	Dynamic fill color..... 101
6.3.6	Dynamic movements..... 102
6.3.7	Dynamic size..... 103
6.3.8	Dynamic rotation..... 104
6.3.9	Events..... 104
6.4	Checkbox..... 106
6.4.1	General..... 106
6.4.2	Position..... 106
6.4.3	Style..... 107
6.4.4	Text properties..... 108
6.4.5	Dynamic fill color..... 110
6.4.6	Dynamic movements..... 111
6.4.7	Dynamic size..... 113
6.4.8	Dynamic rotation..... 115
6.4.9	Events..... 115
6.5	Diagnosis events..... 117
6.5.1	General..... 117
6.5.2	Position..... 117
6.5.3	Style..... 118
6.5.4	Text properties..... 119
6.5.5	Dynamic fill color..... 120
6.5.6	Dynamic movements..... 121
6.5.7	Dynamic size..... 122
6.5.8	Dynamic rotation..... 123
6.5.9	Events..... 123
6.6	Ellipse..... 125
6.6.1	General..... 125
6.6.2	Position..... 125
6.6.3	Style..... 126
6.6.4	Text properties..... 127
6.6.5	Dynamic fill color..... 129
6.6.6	Dynamic movements..... 130
6.6.7	Dynamic size..... 132
6.6.8	Dynamic rotation..... 134
6.6.9	Events..... 134
6.7	Frame..... 136
6.7.1	General..... 136
6.7.2	Position..... 136
6.7.3	Style..... 137
6.7.4	Text properties..... 137
6.7.5	Dynamic fill color..... 138
6.7.6	Dynamic movements..... 138
6.7.7	Dynamic size..... 139

	Page
6.7.8 Dynamic rotation.....	140
6.7.9 Events.....	140
6.8 Image.....	142
6.8.1 General.....	142
6.8.2 Position.....	142
6.8.3 Style.....	143
6.8.4 Text properties.....	143
6.8.5 Dynamic fill color.....	143
6.8.6 Dynamic movements.....	144
6.8.7 Dynamic size.....	144
6.8.8 Dynamic rotation.....	145
6.8.9 Events.....	145
6.9 Input field.....	147
6.9.1 General.....	147
6.9.2 Position.....	147
6.9.3 Style.....	148
6.9.4 Text properties.....	149
6.9.5 Dynamic fill color.....	151
6.9.6 Dynamic movements.....	152
6.9.7 Dynamic size.....	154
6.9.8 Dynamic rotation.....	156
6.9.9 Events.....	156
6.10 Input field extended.....	158
6.10.1 General.....	158
6.10.2 Position.....	158
6.10.3 Style.....	159
6.10.4 Text properties.....	160
6.10.5 Dynamic fill color.....	162
6.10.6 Dynamic movements.....	163
6.10.7 Dynamic size.....	165
6.10.8 Dynamic rotation.....	167
6.10.9 Events.....	167
6.11 Line.....	169
6.11.1 General.....	169
6.11.2 Position.....	169
6.11.3 Style.....	170
6.11.4 Text properties.....	170
6.11.5 Dynamic fill color.....	171
6.11.6 Dynamic movements.....	172
6.11.7 Dynamic size.....	173
6.11.8 Dynamic rotation.....	174
6.11.9 Events.....	174
6.12 Polygon.....	176
6.12.1 General.....	176
6.12.2 Position.....	176
6.12.3 Style.....	177

Table of Contents

	Page
6.12.4 Text properties.....	177
6.12.5 Dynamic fill color.....	178
6.12.6 Dynamic movements.....	179
6.12.7 Dynamic size.....	180
6.12.8 Dynamic rotation.....	181
6.12.9 Events.....	181
6.13 Polyline.....	183
6.13.1 General.....	183
6.13.2 Position.....	183
6.13.3 Style.....	184
6.13.4 Text properties.....	184
6.13.5 Dynamic fill color.....	185
6.13.6 Dynamic movements.....	186
6.13.7 Dynamic size.....	187
6.13.8 Dynamic rotation.....	188
6.13.9 Events.....	188
6.14 Progress bar.....	190
6.14.1 General.....	190
6.14.2 Position.....	190
6.14.3 Style.....	191
6.14.4 Text properties.....	192
6.14.5 Dynamic fill color.....	194
6.14.6 Dynamic movements.....	195
6.14.7 Dynamic size.....	197
6.14.8 Dynamic rotation.....	199
6.14.9 Events.....	199
6.15 Radial gauge.....	201
6.15.1 General.....	201
6.15.2 Position.....	201
6.15.3 Style.....	202
6.15.4 Text properties.....	203
6.15.5 Dynamic fill color.....	205
6.15.6 Dynamic movements.....	206
6.15.7 Dynamic size.....	208
6.15.8 Dynamic rotation.....	210
6.15.9 Events.....	210
6.16 Radial progress bar.....	212
6.16.1 General.....	212
6.16.2 Position.....	212
6.16.3 Style.....	213
6.16.4 Text properties.....	214
6.16.5 Dynamic fill color.....	216
6.16.6 Dynamic movements.....	217
6.16.7 Dynamic size.....	219
6.16.8 Dynamic rotation.....	221
6.16.9 Events.....	221

	Page
6.17	Rectangle..... 223
6.17.1	General..... 223
6.17.2	Position..... 223
6.17.3	Style..... 224
6.17.4	Text properties..... 225
6.17.5	Dynamic fill color..... 227
6.17.6	Dynamic movements..... 228
6.17.7	Dynamic size..... 230
6.17.8	Dynamic rotation..... 232
6.17.9	Events..... 232
6.18	Rotary switch..... 234
6.18.1	General..... 234
6.18.2	Position..... 234
6.18.3	Style..... 235
6.18.4	Text properties..... 236
6.18.5	Dynamic fill color..... 238
6.18.6	Dynamic movements..... 239
6.18.7	Dynamic size..... 240
6.18.8	Dynamic rotation..... 241
6.18.9	Events..... 241
6.19	Slider..... 243
6.19.1	General..... 243
6.19.2	Position..... 243
6.19.3	Style..... 244
6.19.4	Text properties..... 245
6.19.5	Dynamic fill color..... 247
6.19.6	Dynamic movements..... 248
6.19.7	Dynamic size..... 250
6.19.8	Dynamic rotation..... 252
6.19.9	Events..... 252
6.20	Table..... 254
6.20.1	General..... 254
6.20.2	Position..... 254
6.20.3	Style..... 255
6.20.4	Text properties..... 255
6.20.5	Dynamic fill color..... 257
6.20.6	Dynamic movements..... 258
6.20.7	Dynamic size..... 260
6.20.8	Dynamic rotation..... 262
6.20.9	Events..... 262
6.21	Text field..... 264
6.21.1	General..... 264
6.21.2	Position..... 264
6.21.3	Style..... 265
6.21.4	Text properties..... 266
6.21.5	Dynamic fill color..... 268

Table of Contents

	Page
6.21.6 Dynamic movements.....	269
6.21.7 Dynamic size.....	271
6.21.8 Dynamic rotation.....	273
6.21.9 Events.....	273
6.22 Text box.....	275
6.22.1 General.....	275
6.22.2 Position.....	275
6.22.3 Style.....	276
6.22.4 Text properties.....	277
6.22.5 Dynamic fill color.....	279
6.22.6 Dynamic movements.....	280
6.22.7 Dynamic size.....	282
6.22.8 Dynamic rotation.....	284
6.22.9 Events.....	284
6.23 Toggle button.....	286
6.23.1 General.....	286
6.23.2 Position.....	286
6.23.3 Style.....	287
6.23.4 Text properties.....	287
6.23.5 Dynamic fill color.....	289
6.23.6 Dynamic movements.....	290
6.23.7 Dynamic size.....	292
6.23.8 Dynamic rotation.....	294
6.23.9 Events.....	294
6.24 Triangle.....	296
6.24.1 General.....	296
6.24.2 Position.....	296
6.24.3 Style.....	297
6.24.4 Text properties.....	298
6.24.5 Dynamic fill color.....	300
6.24.6 Dynamic movements.....	301
6.24.7 Dynamic size.....	303
6.24.8 Dynamic rotation.....	305
6.24.9 Events.....	305
7 Service and support.....	309
Index.....	311

1 About this documentation

Editions of this documentation

Edition	Release Date	Notes
01	2017-01	First edition
02	2017-07	Extended by the transfer functionality of customized websites for the WebAssistant
03	2018-09	Additions in chapter "User interface", "Use cases", "FAQ" New chapter "Creating customized pages" and "Widget attribute"
04	2020-02	Chapter "Creating customized pages" included in the chapter "Use cases" Additions in chapter "User interface", "Use cases", "FAQ"

Tab. 1-1: Change Record

1.1 Validity of the documentation

Target group This documentation describes the WebComposer functions for IndraWorks. The basic procedure to create WebComposer configurations and applications as description and handling instruction is explained.

Overview on target groups and product phases In the following illustration, the framed activities, product phases and target groups refer to the present documentation.

Example: In the product phase "Engineering", the target group "programmer" can execute the activity "programming" using this documentation.

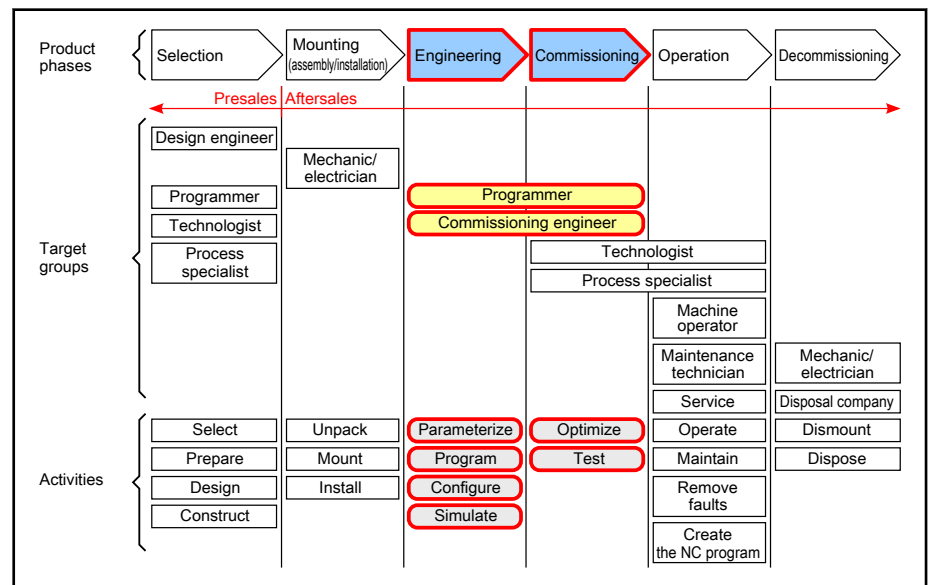


Fig. 1-1: Assigning the present documentation to the target groups, product phases and activities of the target group

1.2 Required and supplementing documentation IndraWorks

1.2.1 Engineering

Documentation titles with type codes and part numbers

IndraWorks 14VRS Software Installation DOK-IWORKS-SOFTINS*V14-CORS-EN-P, R911344286 This documentation describes the IndraWorks installation.
IndraWorks 14VRS Engineering DOK-IWORKS-ENGINEE*V14-APRS-EN-P, R911343566 This documentation describes the application of IndraWorks in which the Rexroth Engineering tools are integrated. It includes instructions on how to work with IndraWorks and how to operate the oscilloscope function.
IndraWorks 14VRS PLC Programming System IndraLogic 2G DOK-IWORKS-IL2GPRO*V14-APRS-EN-P, R911343571 This documentation describes the PLC programming tool IndraLogic 2G and its use. It includes the basic use, first steps, visualization, menu items and editors.
IndraWorks 14VRS Basic Libraries IndraLogic 2G DOK-IL*2G*-BASLIB**V14-LIRS-EN-P, R911343920 This documentation describes the system-comprehensive PLC libraries.
IndraWorks 14VRS WinStudio 7.4 DOK-IWORKS-WINSTUD*V14-APRS-EN-P, R911341585 This "User Manual and Technical Reference Book" facilitates working with the "Rexroth WinStudio"™ software for optimal results. This document provides technical information and step-by-step instructions to create web-enabled HMI/SCADA programs.
Rexroth IndraWorks 13VRS CamBuilder DOK-IWORKS-CAMBUIL*V13-APRS-EN-P, R911336291 This documentation describes the basic principles and operation of the CamBuilder, the cam editing tool.
IndraWorks 14VRS Field Buses DOK-IWORKS-FB*****V14-APRS-EN-P, R911341485 This documentation describes the field bus and local periphery connections supported by the IndraLogic XLC, IndraMotion MLC and IndraMotion MTX systems. The focus of this documentation is on the configuration, parameterization, commissioning and the diagnostics of the different periphery connections.
WebAssistant DOK-GENERL-WEB*ASSIST*-APRS-EN-P, R911381469 This documentation describes the WebAssistant. The WebAssistant is a webbased diagnostic tool used to access a control system via an Ethernet highspeed connection. The WebAssistant allows OEMs, end users and service engineers to access and to remotely diagnose a system. A PC with Internet Explorer version 11 or higher, Firefox version 45 or above or Google Chrome version 54 or above is required. The following control variants are supported: • IndraMotion MLC L45/L65/L75/XM21/XM22/XM42/VPx • IndraLogic XLC L45/L65/L75/VEP

IndraWorks IndraMotion Service Tool DOK-IWORKS-IMST*****-APRS-EN-P, R911341383 This documentation describes the IndraMotion Service Tool (IMST). IMST is a web-based diagnostic tool used to access a control system via an Ethernet high-speed connection. The IMST allows OEMs, end users and service engineers to access and remotely diagnose a system. The PC has to use at least Internet Explorer 8, Firefox 3.5 or a higher version. The following control variants are supported: • IndraMotion MLC L45/L65/L75/XM21/XM22/XM42/VPx • IndraLogic XLC L45/L65/L75
Rexroth IndraDrive Service Tools IMST/IDST DOK-IM*MLD-IMSTIDSTV13-RERS-EN-P, R911342652 This documentation describes the IndraMotion Service Tools (IMST and IDST). It is a web-based diagnostic tool used to access a drive system via a high-speed Ethernet connection. Using these service tools OEMs, end users and service engineers can access and remotely diagnose a system from anywhere.
IndraWorks SafeLogic 14VRS First Steps DOK-IWORKS-SL*STEP*V14-CORS-EN-P, R911341520 This documentation provides information to facilitate an easy start in the safety control engineering SafeLogic. By using project examples, information about the installation, configuration, commissioning, troubleshooting and diagnostics is provided to the user.
Rexroth IndraWorks 12VRS FDT Container DOK-IWORKS-FDT*CON*V12-APRS-EN-P, R911334398 This documentation describes the IndraWorks FDT Container functionality. It includes the activation of the functionality in the project and working with DTMs.
IndraWorks 14VRS Field Buses Libraries DOK-IWORKS-FB*LIB**V14-LIRS-EN-P, R911343575 This documentation describes field bus libraries for the IndraLogic XLC, IndraMotion MLC and IndraMotion MTX systems.
IndraWorks 14VRS Remote Support DOK-IWORKS-REMOTE**V14-APRS-EN-E, R911382628 This documentation supports the user while installing and using a remote support software compatible to the current operating systems from Win7 - Win10.
Rexroth IndraWorks Remote Condition Monitoring RMB_TechRCM DOK-IWORKS-TEC*RCM*V14-LIRS-EN-E, R911343671 This documentation describes the individual functions of the "Remote Condition Monitoring" systems as well as the most important use cases for a quick status assessment of the connected machines/systems.
IndraWorks 14VRS SafeLogic - Project Configuration, Version 02V08 DOK-IWORKS-SL**PRJ*V14-APRS-EN-P, R911341694 This documentation describes the creation and programming of SafeLogic projects in IndraWorks Engineering.

Tab. 1-2: IndraWorks documentation overview - Engineering

1.2.2 Engineering & visualization

Documentation titles with type codes and part numbers

IndraWorks 14VRS Energy Efficiency Management

DOK-IWORKS-4EE*****V14-APRS-EN-P, R911339229

This document describes the use of the Energy Efficiency Management function within the MLC, XLC and MTX systems. Using example applications, this documentation provides information about the project planning, configuration and visualization of the system.

Tab. 1-3: *IndraWorks documentation overview - Engineering & visualization*

1.2.3 Visualization

Documentation titles with type codes and part numbers

IndraWorks 14VRS HMI

DOK-IWORKS-HMI*****V14-APRS-EN-P, R911343569

This documentation describes the HMI operating interface IndraWorks Operation, its operation and functions as well as the configuration in IndraWorks Engineering.

Tab. 1-4: *IndraWorks documentation overview - Visualization*

1.3 Using safety instructions

1.3.1 Structure of the safety instructions

The safety instructions are structured as follows:

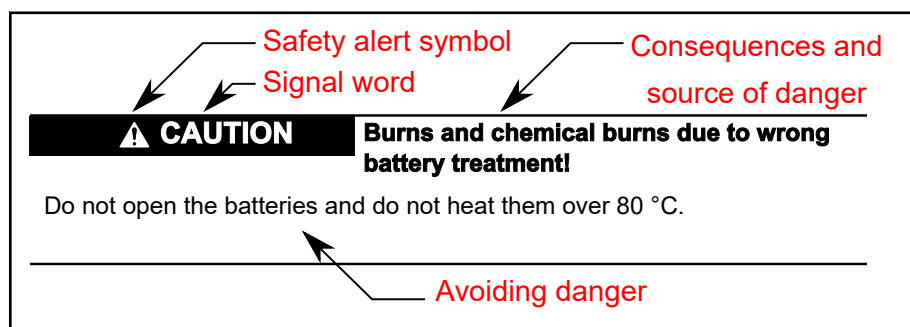


Fig. 1-2: *Structure of the safety instructions*

1.3.2 Explaining signal words and safety alert symbol

The safety instructions in this documentation contain specific signal words (danger, warning, caution, notice) and, if necessary, a safety alert symbol (according to ANSI Z535.6-2006).

The signal word draws attention to the safety instruction and indicates the risk potential.

The safety alert symbol (triangular safety reflector with exclamation marks), preceding the signal words Danger, Warning, Caution indicates hazards for persons.



In case of non-compliance with this safety instruction, death or serious injury will occur.

** WARNING**

In case of non-compliance with this safety instruction, death or serious injury **can** occur.

** CAUTION**

In case of non-compliance with this safety instruction, minor or moderate injury can occur.

NOTICE

In case of non-compliance with this safety instruction, material damage can occur.

1.3.3 Symbols used

Pointers are displayed as follows:



This is a note.

Tips are displayed as follows:



This is a tip.

1.3.4 Explaining the signal alert symbol on the device



If this symbol is on your device, you have to observe the documentation on the device. The respective documentation informs on the type of hazard as well as the steps required to avoid this hazard.

1.4 Names and abbreviations

Term	Explanation
Checkout	Term in connection with Version Control Changes possible
Hijack	Term in connection with Version Control (Local) changes possible
IWE	IndraWorks Engineering
IWO	IndraWorks Operation
NC	Numerical Control
OEM	Original Equipment Manufacturer
VCS	Version Control System

Tab. 1-5: Names and abbreviations used

1.5 Customer feedback

Customer requests, comments or suggestions for improvement are of great importance to us. Please email your feedback on the documentations to Feedback.Documentation@boschrexroth.de. Directly insert comments in the electronic PDF document and send the PDF file to Bosch Rexroth.

2 Important instructions on use

2.1 Intended use

2.1.1 Introduction

Bosch Rexroth products represent state-of-the-art developments and manufacturing. The products are tested prior to delivery to ensure operating safety and reliability.

The products may only be used as intended. If they are not used as intended, situations occur that result in damage to property or personal injury.



Bosch Rexroth shall not assume any warranty, liability or payment of damages in case of damage resulting from a non-intended use of the products; the use shall solely bear all risks from unintended use of the products.

Before using Rexroth products, ensure that all the prerequisites for an intended use of the products are met:

- Anybody handling Bosch Rexroth products in any way is obliged to read and consent to the relevant safety instructions and the intended use.
- The original condition of hardware products may not be altered; in other words, no structural modifications are permitted. It is not permitted to decompile software products or alter source codes.
- Do not install damaged or defective products or use them in operation.
- It has to be ensured that the products have been installed as described in the relevant documentation.

2.1.2 Areas of use and application

IndraWorks Engineering is used for configuration, parameterization, programming and diagnostics.

IndraWorks Engineering is used for selected Bosch Rexroth drive and control systems.

Additionally, the specifications given in the "Areas of Use and Application" for the control and drive systems used with the IndraWorks Engineering do also apply.



IndraWorks Engineering is designed for a screen setting of 96 DPI. The use of other settings is considered as unintended and can cause an incomplete or incorrect display of screen elements.

2.1.3 Unintended use

Any use of IndraWorks Engineering in applications other than those specified above or under operating conditions and other than those described in the documentation and the technical specifications is considered as "unintended".

Furthermore, IndraWorks Engineering must not be used in any applications not expressly approved by Bosch Rexroth.

3 User interface

3.1 Overview

The user interface of the WebComposer is part of IndraWorks Engineering. All application elements of the WebComposer are integrated in the IndraWorks user interface.

3.2 Project elements and their functions

The WebComposer has many static project elements that are automatically created when creating a WebComposer visualization. Additionally, the user has the option to add more project elements to the Project Explorer.

In contrast to static project elements, the recently added project elements can be deleted by the user at any time.

All project elements have general as well as specific functions that can be called via the context menu of the element or via the main menu.

3.2.1 WebComposer commissioning visualization

For the "WebComposer commissioning visualization", go to the "Visualization" library.

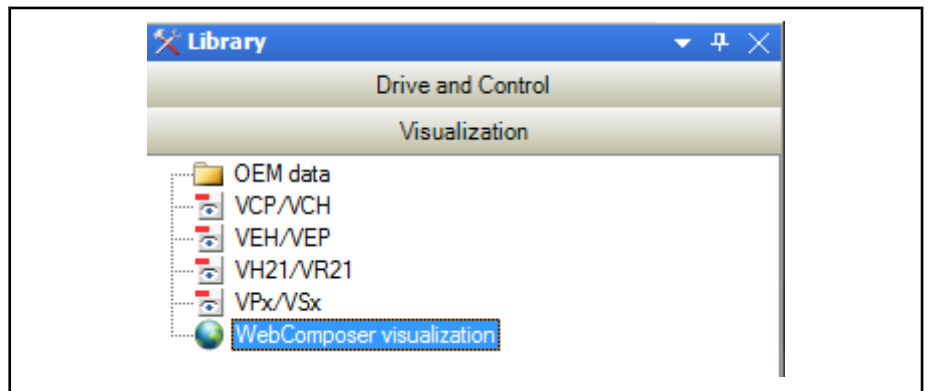


Fig. 3-1: Library with WebComposer commissioning visualization

Use "drag-and-drop" to create the "Visualization" node below a project. Optionally, a WebComposer commissioning visualization can also be created via the context menu of the project node.

Below the visualization node, folders or screens can be created. The number of folders and screens is not limited.

The user can delete or rename the visualization. When renaming the visualization, make sure, that the visualization name is unique within a project.

Apart from these general functions, the visualization provides additional functions:

- [chapter 3.7.7 "Find & Replace function" on page 33](#)
- [chapter 3.7.8 "Validation function" on page 36](#)
- [chapter 3.7.10 "Selecting customized screens" on page 38](#)
- [chapter 3.7.11 "3S visualization import" on page 40](#)

3.2.2 Global data

"Global data" is a static project element that is automatically generated when generating a visualization. This element is used to structure and summarize similar elements in the project tree.

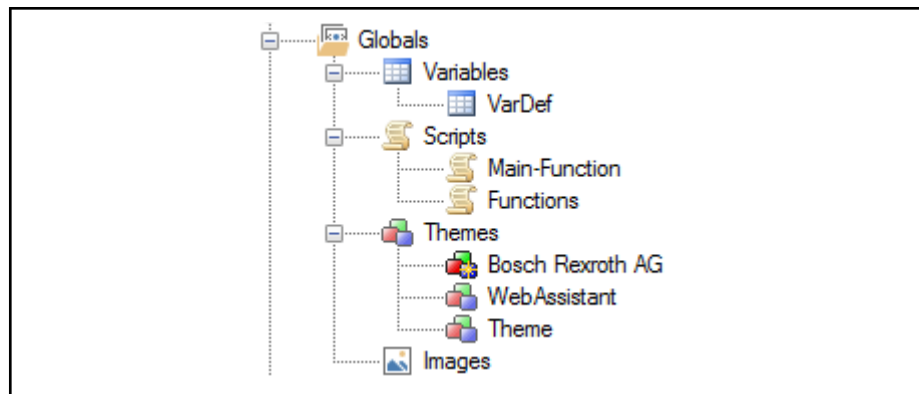


Fig. 3-2: Project element "Global data" and its static subelements

3.2.3 Variables and VarDef

"Variables" is a subelement of "Global data". Global Variables are a static project element that is automatically generated when generating a visualization and which cannot be deleted.

"VarDef" – a subelement of "Variables" – is also created automatically and cannot be deleted. Via "VarDef", it is possible to open the script editor (see [chapter 3.3.2 "Script editor" on page 16](#)) to create and edit "Global variables" (see [chapter 3.7.1 "Global variables" on page 30](#)).

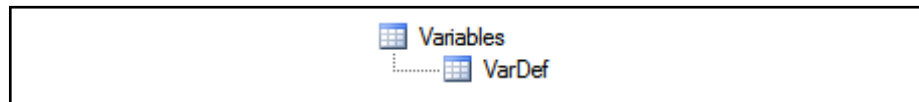


Fig. 3-3: "Variables" and "VarDef" project elements

3.2.4 Variables and TextDef

"Variables" is a subelement of "Global data". Global Variables are a static project element that is automatically generated when generating a visualization and which cannot be deleted.

"TextDef" – a subelement of "Variables" – is also created automatically and cannot be deleted. Via "TextDef", it is possible to open the text editor (see [chapter 3.3.3 "Text editor" on page 17](#)) to create and edit "Global texts". The text editor also supports text localization. These texts are available in the WebComposer device for all components.

3.2.5 Scripts, main functions and functions

"Scripts" is a subelement of "Global data" (see [chapter 3.2.2 "Global data" on page 9](#)). Global Variables are a static project element that is automatically generated when generating a visualization. This element is used to structure all global scripts.

"Main functions" - a subelement of "Scripts" is automatically generated. Via the project element "Main functions", it is possible to open the script editor (see [chapter 3.3.2 "Script editor" on page 16](#)) to create and edit the global main functions (see [chapter 3.7.2 "Global main functions" on page 30](#)).

"Functions" - a subelement of "Scripts" is automatically generated. Via the project element "Functions", it is possible to open the script editor (see [chapter 3.3.2 "Script editor" on page 16](#)) to create and edit the global functions (see [chapter 3.7.3 "Global functions" on page 31](#)).

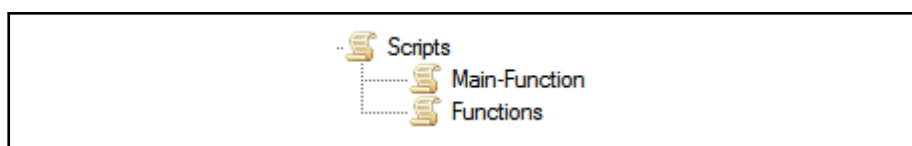


Fig. 3-4: Project element "Scripts" and its static subelements

There are default main functions which are used on the entire device. It is not possible to add main functions and the default main functions should not be deleted.

The functions, however, can be defined and called by the main functions and the screen script.

3.2.6 Themes

"Themes" is a subelement of "Global data" (see [chapter 3.2.2 "Global data" on page 9](#)). Global Variables are a static project element that is automatically generated when generating a visualization. This element is a collective element for all themes and cannot be deleted. The user can create user-defined themes.

The "Bosch Rexroth AG" theme - a subelement of the "Themes" project element is automatically created and cannot be deleted. This theme contains a Bosch default configuration that cannot be changed by the user!

The "WebAssistant" theme - a subelement of the "Themes" project element is automatically created, but can be renamed and deleted by the user. This theme contains a Bosch configuration for customized screens to be displayed by the WebAssistant. The theme can be modified by the user!

The "Theme" theme - a subelement of the "Theme" project element is automatically created, but can be renamed and deleted by the user. This theme contains a Bosch default configuration that cannot be changed by the user.

All newly created themes are visualized below the "Themes" project element. The "Themes" editor (see [chapter 3.3.4 "Theme editor" on page 17](#)) can be opened for each individual theme. Edit theme configurations in this editor.

Function: "Activate theme"

All themes below the project element "Themes" can be set as active theme. This done via the context menu of the corresponding theme.

There is only one active theme within a WebComposer commissioning visualization. As soon as another theme is selected as active theme, the original active theme loses its status.

The active theme is visualized using the icon .

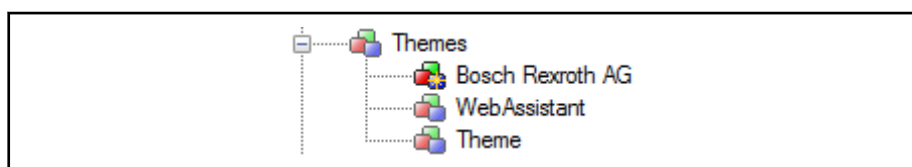


Fig. 3-5: Project element "Themes" with active "Bosch Rexroth AG" theme

The status specifies which theme is used to represent all screens and graphic objects (widgets).

If the active theme is deleted, the status automatically changes via the "Bosch Rexroth AG" theme.



Individual widgets can be taken from the active theme (see [chapter 4 "Use cases" on page 41](#)).

- "Export" function** Provides the option to export themes to an external file to subsequently be able to import them into another project.
In the first dialog, a description can be added to the export file.
In the last dialog, an export summary is displayed. This shows whether the export was completed successfully.
- "Import" function** Provides the option to import themes from an external file back to the project.
In the first dialog (Windows default dialog Open), an already existing export file has to be selected for import.
The last dialog displays an import summary. This shows whether the import was completed successfully.

3.2.7 Images

"Graphics" is a subelement of "Global data" (see [chapter 3.2.2 "Global data" on page 9](#)). Global Variables are a static project element that is automatically generated when generating a visualization.

This element is a collective folder for all graphics and cannot be deleted. The user can assign graphics to the project.

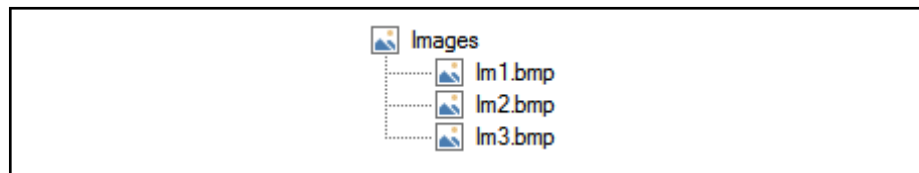


Fig. 3-6: Project element "Graphics" with three added graphics

Function: "Add graphic"

Using this function, the user can assign different graphics such as bit maps and/or icons to the project using a dialog box.

In the screen configuration, all graphics created in the project can be assigned to the widgets or the screen itself as background image.

- "Export" function** Provides the option to export elements to an external file.
In the first dialog, a description can be added to the export file.
In the last dialog, an export summary is displayed. This shows whether the export was completed successfully.
- "Import" function** Provides the option to import graphics from an external export file back to the project.
In the first dialog (Windows default dialog Open), an already existing export file has to be selected for import.
The last dialog displays an import summary. This shows whether the import was completed successfully.

3.2.8 Screen

A "Screen" is an element which can be created below the visualization in subfolders.

Exactly one script element and one variable element can be added to each screen.

Via a "Screen", the screen editor (see [chapter 3.3.1 "Screen editor" on page 14](#)) can be opened in which the screen is configured.

A screen and potentially existing subelements (scripts and variables) can be exported and imported.

Each screen can be deleted or renamed by the user. When renaming the screen, make sure, that the screen name is unique within a project. When deleting a screen, all subordinate data is also deleted.

3.2.9 Script

To be able to create a screen script (or also local script), explicitly add the element "Script" to the Project Explorer below a screen.

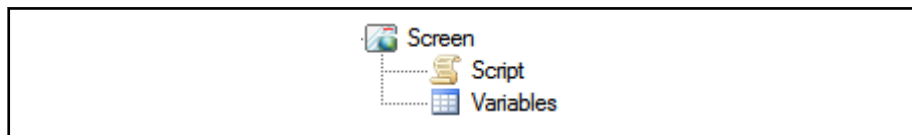


Fig. 3-7: Screen with screen script

A local screen script is edited using the script editor (see [chapter 3.3.2 "Script editor" on page 16](#)) and can also be deleted again by the user.

The local screen script is only available in the context of the superordinate screen.

Other screens cannot access this script.

3.2.10 Variables

To be able to create variables (or also local variables), explicitly add the element "Variable" to the Project Explorer below a screen.

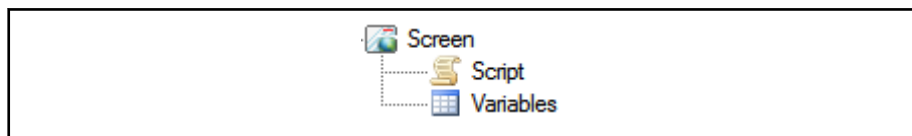


Fig. 3-8: Screen with screen variables

Local screen variables are edited using the script editor (see [chapter 3.3.2 "Script editor" on page 16](#)) and can also be deleted again by the user.

The local screen variables are only available in the context of the superordinate screen.

Other screens cannot access these variables.

3.2.11 Folder

"Folders" are only structuring elements for the user. These folders make the structure of large projects clearer.

Folders can be created below the visualization or below a folder.

Creating folders below the "Global data" elements or below a "Screen" is not possible..

Folders can be renamed and deleted. Copying and/or pasting folders is also possible.

"Export" function Provides the option to export folders to an external file to subsequently be able to import them into another project.

In the first dialog, a description can be added to the export file.

In the last dialog, an export summary is displayed. This shows whether the export was completed successfully.

"Import" function Provides the option to import folders from an external file back to the project. In the first dialog (Windows default dialog Open), an already existing export file has to be selected for import.

The last dialog displays an import summary. This shows whether the import was completed successfully.

3.2.12 Templates

Via the context menu ("Add/Templates") of the WebComposer commissioning visualization, templates can be added to the project. These templates are provided by IndraWorks. The templates are screens with widgets that can be changed/edited by the user after they have been created in the project tree. They can be used as references to other screens (see [chapter 4.4.2 "Editing referenced screens" on page 46](#)).



From IndraWorks 15V08, the user can also create templates.

There are two options to create templates. One option is to go to the context menu of a screen. Select "Create template" to create a template from the selected screen. This template is available in all WebComposer projects in this IndraWorks installation. Select "Create template" to open a dialog to specify a name for the template.

For the other option to create templates, go to the context menu of the WebComposer main node. There is the menu item "Manage templates" and two commands to create or delete templates. Select "Create template" and a dialog opens to create a template. In contrast to the first option, multiple screens can be selected. All selected screens are part of the template.

Each user-created template can also be deleted. Default templates from the installation cannot be deleted.

NOTICE

The names of the screens, from which the templates consist of, are used again when instantiating a template. If there is already a screen with the same name in the project, it is replaced by the screen from the template.

3.3 Editors

All WebComposer editors are displayed in the working area of the IndraWorks user interface.

Different editors are available for different tasks.

3.3.1 Screen editor

To configure a screen, different tool windows are used apart from the editor in the working area.

Toolbar



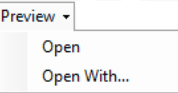
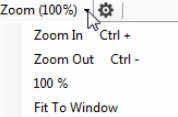
Fig. 3-9: Toolbar



Some functions are only active if one or several widgets are selected.

Toolbar functions

Icon	Name	Description
	Undo	Undoes changes step-by-step
	Repeat	Restores undone changes
	To the foreground	Brings a selected widget to the front
	In the background	Brings a selected widget to the back
	Bring to foreground	Brings a selected widget to the front step-by-step
	Bring to background	Brings a selected widget to the back step-by-step
	Align at grid	Aligns a widget at the specified grid
	Group	Groups several widgets to one object Grouped widget can simultaneously be edited
	Resolve grouping	Separates the selected grouping in its individual components Widgets can now be edited individually and independently
	Align bottom edge	Aligns all selected widgets at the bottom edge of the last selected widget
	Align top edge	Aligns all selected widgets at the top edge of the last selected widget
	Align right edge	Aligns all selected widgets at the right edge of the last selected widget
	Align left edge	Aligns all selected widgets at the left edge of the last selected widget
	Center alignment horizontally	Aligns all selected widgets horizontally to the last selected widget
	Center alignment vertically	Aligns all selected widgets vertically to the last selected widget
	Center distance horizontally	Aligns the distances of all selected widgets horizontally
	Center distance vertically	Vertically aligns the distances of all selected widgets
	Adjust height	Adjust the height of all selected widgets to the height of the last selected widget
	Adjust width	Adjust the width of all selected widgets to the width of the last selected widget
	Current mode	Editing mode: Screen can be adjusted; widgets can be added, deleted or edited
		RUN mode: Screen is shown in the preview (browser)

Icon	Name	Description
	Refresh	Refreshes the desktop following a change in the screen Changes to the screen are displayed in the preview if the preview is opened Refreshes the frame state if changes were made
	Status	Status Connected: No connection to the control established All further information is output in the message window
		Status Not Connected: No connection to the control established All further information is output in the message window
	Ignore incorrect communication variables	If the switch is active, all incorrect communication variables in the screen in preview mode are ignored and are exempt from the communication. If the switch is inactive and if there is at least one incorrect variable in the screen, the entire communication of the screen does not work in preview mode.
	Preview	"Open": Opens the preview of the current screen via the default browser Use "Refresh" to be able to view the current screen status "Open with": Opens the preview of the current screen via the a browser selected by the user Use "Refresh" to be able to view the current screen status
	Zoom	"Zoom In": Zooms in the screen step-by-step "Zoom out": Zooms out the screen step-by-step "100%": Unscaled screen "Fit To Window": Automatically adjust the screen size to the display screen
	Show DevTools	Opens the DevTools via a separate window For more information about the DevTool, refer to https://developers.google.com/chrome-developer-tools/?PHPSES-SID=ebd55af1156a549f3503274c51dd2b8f

Tab. 3-1: Symbols of the symbol bar of the screen editor



Several widgets can be selected by holding <Ctrl> when selecting the widget or by drawing a frame around the widget.

To be able to move widgets quicker, widgets can be moved using "Alt + cursor".

3.3.2 Script editor

For editing variables (global and local variables) as well as editing scripts (main functions, global and local functions), the script editor is used.

This is an editor that supports the Java script syntax (syntax highlighting), checks the inputs for syntactic errors and displays these errors.

- See [chapter 3.7.1 "Global variables" on page 30](#)
- See [chapter 3.7.4 "Screen variables" on page 32](#)
- See [chapter 3.7.2 "Global main functions" on page 30](#)
- See [chapter 3.7.3 "Global functions" on page 31](#)

- See [chapter 3.7.5 "Screen scripts" on page 32](#)

3.3.3 Text editor

The text editor is used to edit global texts and global text lists as well as to localize these texts.

This is an editor that supports entering and editing of global texts in different languages and which checks the used identifiers for syntactic errors.

- See [chapter 3.2.4 "Variables and TextDef" on page 10](#)
- See [chapter 4.9 "Creating global, localized texts" on page 64](#)
- See [chapter 4.10 "Formatting global, localized texts" on page 65](#)
- See [chapter 4.11 "Using global, localized texts in widgets" on page 65](#)

3.3.4 Theme editor

Specify and edit visual properties of all screens and screen objects with the themes editor.

Use "Themes" in the project tree to call the dialog.

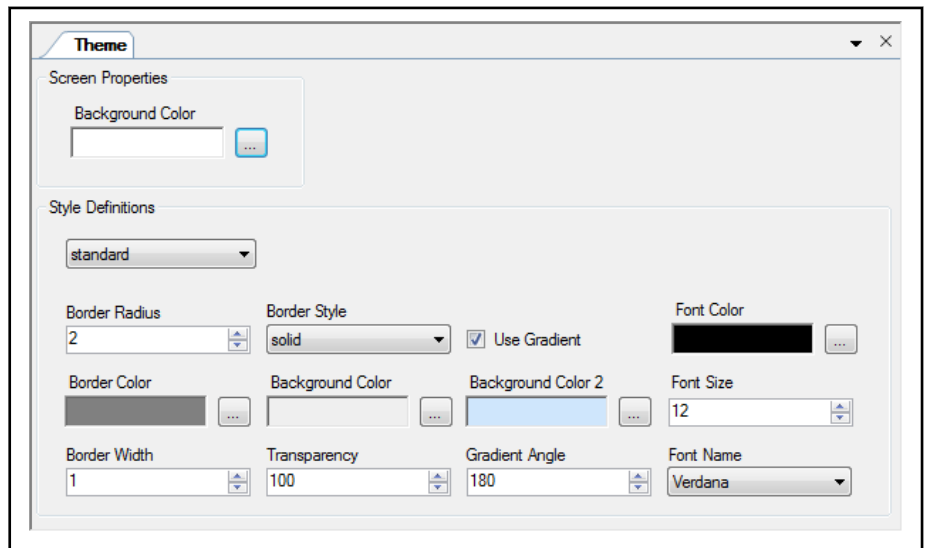


Fig. 3-10: Theme editor

Each theme contains three style types ("standard", "light" und "dark"). The style type contains a theme feature and can be assigned to each widget individually.

The three style types can be edited and adjusted by the user.

The following properties can be defined and edited:

- Border radius
- Border style
- Color gradient
- Font color
- Border color
- Background colors
- Font size
- Border width
- Transparency

- Gradient angle
- Font name

3.3.5 Event editor

The event editor is a tool to configure dynamic properties in screen objects. Using the editor, functions can be configured that are triggered when an event occurs (e.g. mouse-click).

The event editor is called via the event attributes of a widget in the "WebComposer Properties" dialog.

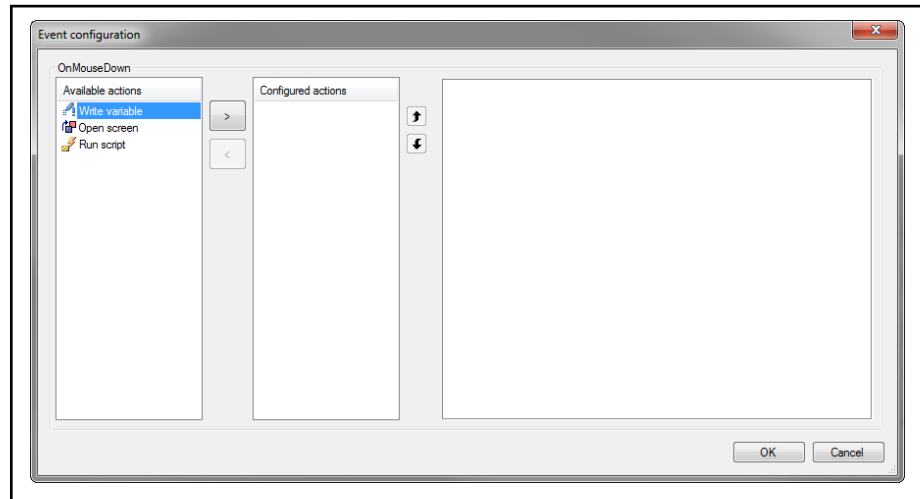


Fig. 3-11: Event editor

- Write variable** Facilitates selecting a variable and entering a value. If the event is executed, the WebComposer sets the entered value.
- Open screen** Facilitates selecting a variable a screen. If the event is executed, the WebComposer opens the selected screen.
- Start script** Facilitates selecting writing a script. If the event is executed, the WebComposer starts the written script.

3.4 WebComposer widget library

This library is displayed in the tool window of IndraWorks Engineering. All widgets and templates available for the WebComposer is contained in the library.

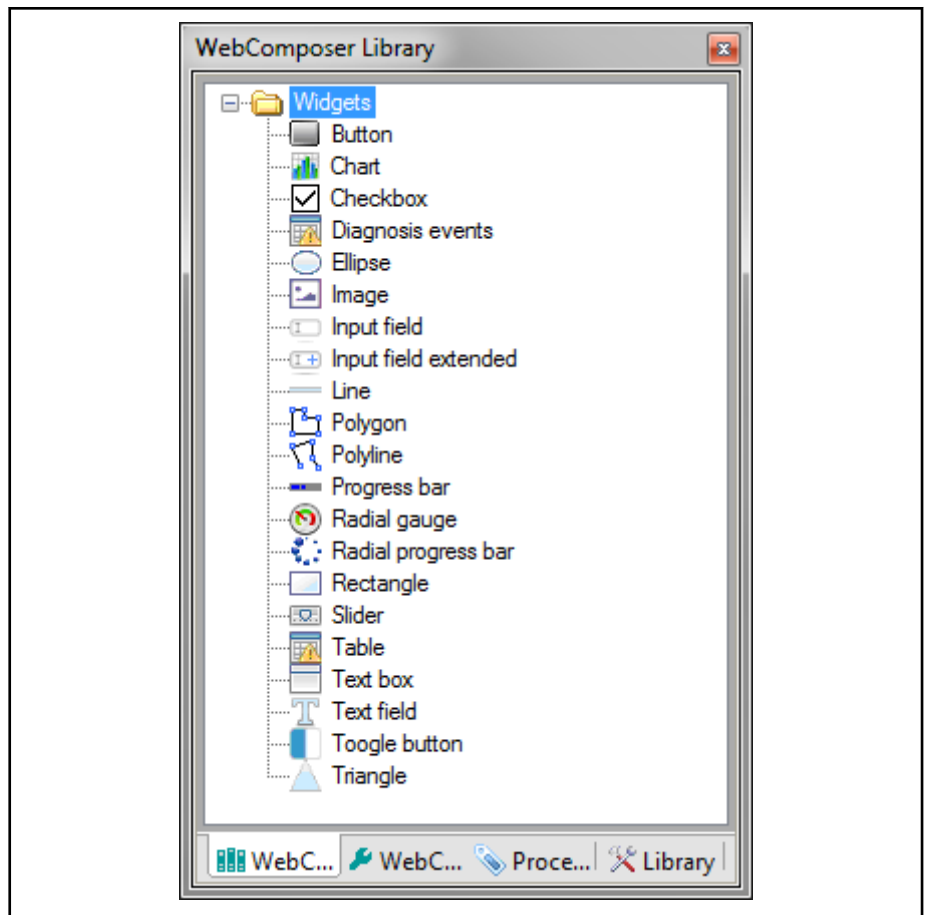


Fig. 3-12: WebComposer library

Use "drag-and-drop" to drag the widgets and templates to a screen.

This library is initially loaded in the tool window upon opening the first screen. When closing the screen, the library is retained in the tool window.

Double-click on the screen background to bring the library automatically to the foreground.

All library elements (widgets) available to the WebComposer are displayed in alphabetical order in the "WebComposer library".

Widget properties:

- Widgets are not limited in their number
- Widgets can be positioned on any position in the screen using "drag-and-drop"
- Double-click on a widget in the screen to open the corresponding tool window properties
- Widgets can be adjusted and changed via the WebComposer properties
- The widgets available in a screen can be deleted at any time

3.5 WebComposer widget attribute

This library is displayed in the tool window of IndraWorks Engineering. All static and dynamic properties available for a widget are contained in the library.



Not all attributes are available to all widgets.

The following chapters are a general description of all widget attributes.

For a detailed description of the widget attributes, refer to [chapter 6 "Attachment: Widget attribute" on page 79](#).

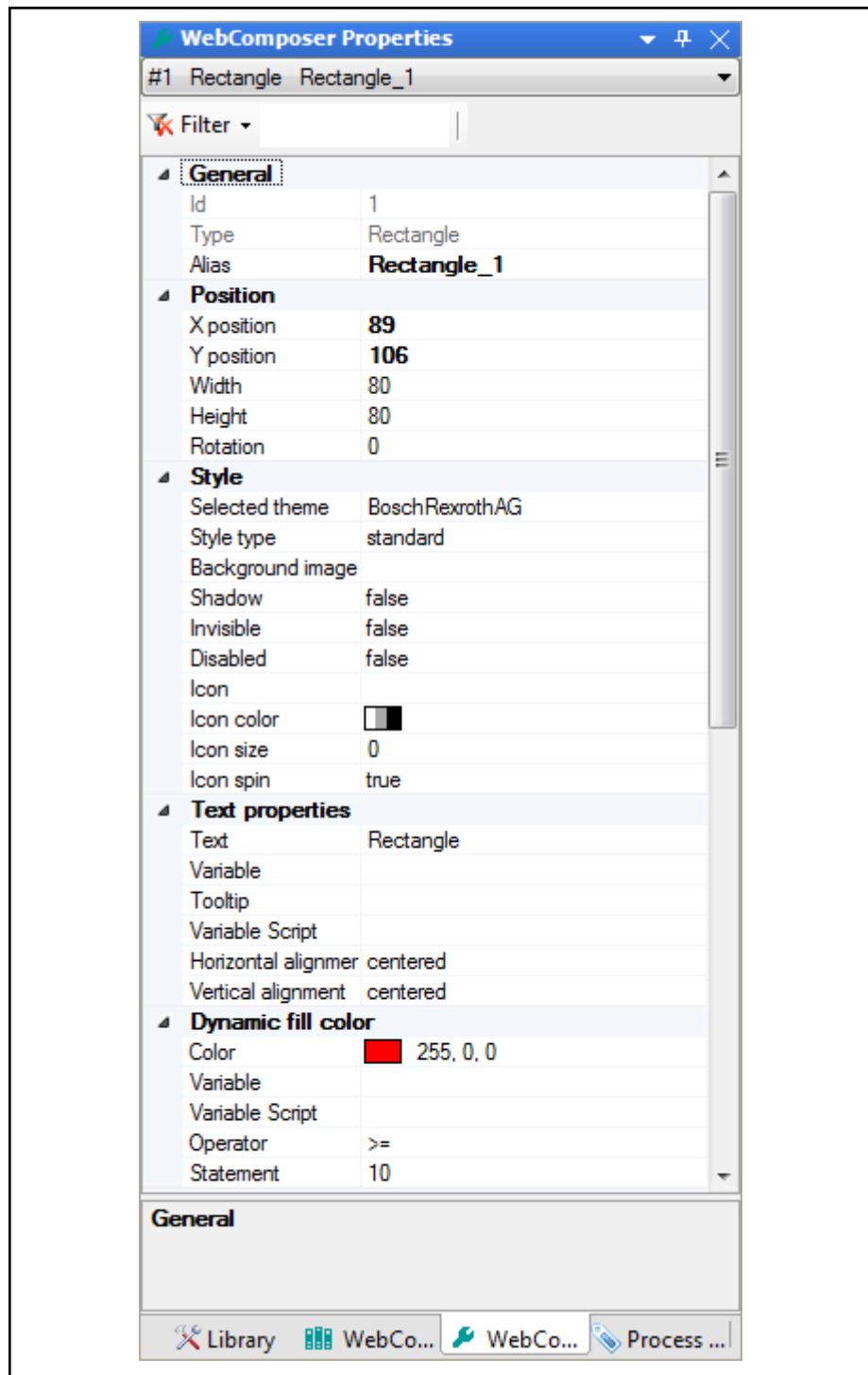


Fig. 3-13: WebComposer widget attribute

General information

The user can change the static and dynamic properties.

The Property window is loaded in the tool window when double-clicking on a widget or when the window is shown via the "View" menu in IndraWorks. When closing the screen, the Property window is retained in the tool window.

If the user clicks on a widget, the focus changes to the Property window.

In the Property window, the properties of the selected widget is displayed. If the background of a screen (no widget) is selected, the screen attributes are displayed.

Menu bar The menu bar of the Property window provides the following options:

1. **Filter**  **Filter** ▾ : Filter settings selection to adjust the property list to the user requirements.

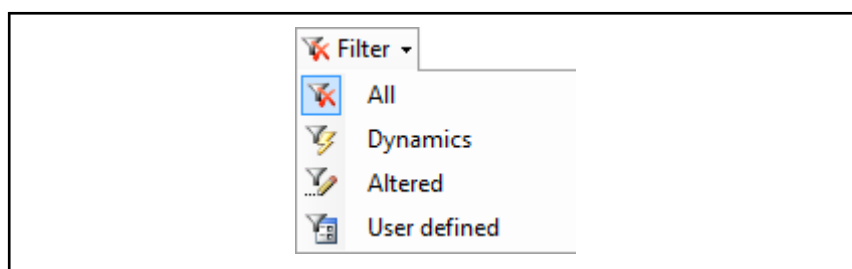


Fig. 3-14: Filter settings

- **All:** All categories with the corresponding properties
- **Dynamics:** Only properties used for dynamics
- **Changed:** Only properties that have been edited
- **User defined:** User-specific summary of the visible properties of all used widgets. The desired properties can be selected under

 **Configuration...**

2. **Search function:** Facilitates input of the individual search text
Subsequently, only properties containing entered text are displayed.

3.5.1 General

The "Properties" group "General" shows basic details of the selected widgets.

Alias Facilitates the assignment of an individual name. The default name can be changed by the user.

ID Visualizes the unique identification number of the widget. This identification number cannot be modified by the user!

Type Visualizes the widget type. This type cannot be modified by the user!

3.5.2 Position

The "Properties" group "Position" facilitates an adjustment of the position, size and rotation of a widget.

X- and Y-position Facilitates individual adjustment of a widget position.

The position can be changed as follows:

- Input of values in "Properties" field
- Move the position in the screen by using the mouse
- Move the position in the screen by using the keyboard cursor

The x-values and the y-values are determined by using the distance from the left (x-position) or the upper (y-position) border in pixels.

- Width and height** Facilitates individual adjustment of the widget size. The size can be changed directly in the screen by entering values in the "Properties" field or by zooming in or out using the mouse. The measurement unit of a widget is pixel, including the border.
- Rotation** Facilitates individual adjustment of the widget rotation. The rotation can be changed directly in the screen by entering values in the "Properties" field or by manually rotating in the screen using the mouse. The rotation value unit of a widget is degree around its center.

3.5.3 Background style

- The "Properties" group "Background style" facilitates individual adjustment of the visual background style of the screen editor.
- Background** Facilitates individual adjustment of the visual background style of the screen editor.
- This property contains the following options that can be selected via the drop-down menu .
- **FillColor:** Facilitates setting of a fill color. By selecting "FillColor", the specific property "Fill color" is available that can be used to select the color. Use the drop-down menu to select a theme. Use the  symbol to open the drop-down menu..
 - **Image:** Facilitates setting of a screen as background. By selecting "Image", the specific property "Background image" is provided. The user can select an image that has already been added to the project tree. Use the drop-down menu to select a theme. Use the  symbol to open the drop-down menu..
 - **LinearGradient:** Facilitates setting of a color gradient as background. If "LinearGradient" is selected, the specific properties "FillColor1", "FillColor2" and "Gradient angle" are provided. Use these properties to create an individual color gradient.

3.5.4 Communication settings

- The "Properties" group "Communication settings" allows to individually assign a screen to a control.
- IP address of the screen** Allows the adaptation of the IP address of the control used in the screen.
- Using screen IP address** Specifies whether the IP address of the screen is used instead of the globally set IP address of the visualization device.
- Checking UA variables** Specifies whether subscriptions should only be created for valid variables.

3.5.5 Grid

- The "Properties" group "Grid" facilitates a customized design as well as enabling and disabling the screen editor grid.
- Grid stance and internal/external margin color** The "Grid distance" property facilitates adjusting the grid distance by entering values.
- The "Internal margin color" and "External margin color" property facilitates the specification of the respective colors. Data is entered via the drop-down menu. Use the  symbol to open the drop-down menu..
- Grid active** Facilitates enabling and disabling of the grid. Data is entered using the values "true" (enabled) and "false" (disabled) via the drop-down menu. Use the  symbol to open the drop-down menu..

3.5.6 Look and feel

	The "Properties" group "Look and feel" facilitates a customized adjustment of the design features of the selected widgets in a screen.
Selected theme and style type	The property "Selected theme" facilitates the selection of a theme. Use the drop-down menu to select a theme. Use the  symbol to open the drop-down menu.. All available themes are displayed. If the user sets the theme to "none", new properties are displayed. Consequently, more specific specifications can be made (see chapter 4.16 "Taking a widget from a theme" on page 74). The "Style type" property facilitates adjusting of the design template features.
Line type	Facilitates specification of a line type. The user can select between "continuous line" or "dashed line". Use the drop-down menu to select a theme. Use the  symbol to open the drop-down menu..
Background screen and screen	The "background screen" allows specifying of a graphics used for the widget. The "Screen" property facilitates specifying a graphic that displays the widget.
Color of progress bar and background color	The "Color of progress bar" property facilitates specifying a color in which the progress bar is to be displayed. Use the drop-down menu to select a theme. Use the  symbol to open the drop-down menu.. The "Background color" property facilitates specifying a color in which the widget is to be displayed. Use the drop-down menu to select a theme. Use the  symbol to open the drop-down menu..
Minimum and maximum	Facilitates specifying the smallest or largest value. For example, if the "Speedo" widget is to display the values 100 to 500, the value 100 has to be specified in the "Minimum" "Properties" field and the value 500 has to be specified in the "Maximum" "Properties" field.
Main markings and secondary markings	Facilitate specifying the number of main and secondary markings of a "Speedo" widget.
Pointer color, point color bottom, point color center, point color top	Facilitate the graphic customization of the widget by using colors. Use the drop-down menu to select a theme. Use the  symbol to open the drop-down menu..
Transparency	Facilitates the specification of the transparency by entering a value between 0 and 100. Value 0 results in the object being completely transparent, value 100 results in the object overlapping the background completely. The value is entered directly in the "Properties" field.
Shadows, invisible and disabled	Can be controlled by assigning "true" or "false". The "Shadow" property facilitates enabling or disabling of an umbra shadow to achieve a spatial effect. The "Invisible" property facilitates displaying and hiding individual widgets. Linking to a variable is also possible. The "Disabled" property facilitates the inactive widget representation. Subsequently, the widget cannot be actively operated anymore. Linking to a variable is also possible.
Icon, icon color, icon size, rotate icon	Facilitates the icon representation on the basis of a free WEB library font-awesome (see http://fontawesome.io/icons/) as well as the graphic feature configuration.

The "Icon" property facilitates specifying the font-awesome icon class as text (example: fa-folder).

The "Icon color" property facilitates specifying a color to be used for the icon.

The color is selected via the drop-down menu. Use the  symbol to open the drop-down menu..

The "Icon size" property facilitates specifying the icon size in five steps (0 to 5x).

The "Rotate icon" facilitate a continuous rotating of the icon. This property can be used in combination with "fa-spinner" to visualize loading.

3.5.7 Text properties

The "Text properties" "Properties" group facilitates customizing of used texts.

Text and tooltip The "Text" property facilitates direct text input in the "Properties" field. The entered text is displayed in the preview and the browser. The text supports the "Printf" functionality.

Example:

Printf functionality formatting

"Float number: %3.2f"

Current value: 3.14159

Output:

Float number: 3.14



For more information on formatting with "Printf", refer to the info area in the WebComposer properties by clicking on the respective property.

Furthermore, the "Text" attribute supports the use of HTML elements to format the text.

Example:

Formatting using HTML elements

"first line
 second line"

Output:

first line

second line



For more information about formatting the text output with HTML elements, refer to http://www.w3schools.com/html/html_formatting.asp.

For the following widgets, text output with HTML elements cannot be used:

- Input field
- Input field extended
- Progress bar
- Triangle
- Speedo
- Switch

Minimum and maximum

The "Tooltip" property facilitates direct text input in the property field. The entered text is displayed in the preview and the browser if the mouse hovers over the widget.

Facilitates specification of a minimum or maximum value. Comply with the following rules regarding the format: Time=hh:mm:ss, date=yyyy-mm-dd, number=integer number. Input text using the keyboard in the "Properties" field.

Type

Facilitates specifying the type of the entered values. There are the following options: "text", "number", "time", "date", "email" or "password".

Use the drop-down menu to select a theme. Use the  symbol to open the drop-down menu..

Activation interval

Facilitates specification of a refresh interval. Possible options: "none", "slowest", "slow", "default", "fast" or "fastest".

Use the drop-down menu to select a theme. Use the  symbol to open the drop-down menu..

Variables and variable script

The "Variable" property facilitates linking to a variable. Via the  symbol, the process variable window is called. In this window, a variable can be selected from all global, internal and screen variables. The variable can also be entered manually into the field. The variable value is enter in the field manually.

The "Variable script" property facilitates editing of the read variables using JavaScript code. The script is executed when the variable values change. The current value is accessed via "value". Enter the script directly into the "Properties" field.



If a variable or a script is saved in both properties, the "Variable script" property has priority.

Horizontal and vertical alignment

The "Horizontal alignment" property facilitates individual adjustment of the widget content alignment in horizontal direction.

The "Vertical alignment" property facilitates individual adjustment of the widget content alignment in vertical direction.

The options "left", "centered" or "right" via the respective drop-down menu via the  symbol.

3.5.8 Dynamic fill color

The "Dynamic fill color" "Properties" group facilitates customizing of the fill colors of individual widgets.

Color Facilitates specifying a color to be used. Use the drop-down menu to select a theme. Use the  symbol to open the drop-down menu..

Variables and variable script The "Variable" property facilitates linking to a variable. Via the  symbol, the process variable window is called. In this window, a variable can be selected from all global, internal and screen variables. The variable can also be entered manually into the field.

The "Variable script" property facilitates editing of the read variables using JavaScript code. The script is executed when the variable values change. The current value is accessed via "value". Enter the script directly into the "Properties" field.



If a variable or a script is saved in both properties, the "Variable script" property has priority.

Operator and condition The "Operator" property facilitates specifying a comparison operator. The following options are available: "Value is greater than or equal to condition", "value is smaller than or equal to condition", "value is equal to condition" or "value is unequal to condition". Use the drop-down menu to select a theme. Use the  symbol to open the drop-down menu..

The "Condition" property facilitates entering a value to be compared to the value of the linked variable. Use the keyboard in the "Properties" field to select a value.

Widgets get be actuated by using an input field so that the fill color changes when reaching certain values.

3.5.9 Dynamic motion

The "Dynamic motion" "Properties" group facilitates controlling the individual widget motions.

Position X and position Y In this property, an "absolute" or "relative" motion in x-direction or y-direction can be selected.

Axis point 0/0 of the screen editor is used as scale in case of an absolute motion, the initial point of the widget is the scale in case of a relative motion.

Variable X and variable Y The "Variable X" property facilitates linking to a variable. Via the  symbol, the process variable window is called. In this window, a variable can be selected from all global, internal and screen variables. The variable can also be entered manually into the field.

The linked variable controls the element motion in x-direction.

The "Variable Y" property facilitates linking to a variable. Via the  symbol, the process variable window is called. In this window, a variable can be selected from all global, internal and screen variables. The variable can also be entered manually into the field.

The linked variable controls the element motion in y-direction

Widgets get be actuated by using an input field so that the widgets move into x-direction or y-direction, depending on the value.

Variables X script and variables Y script Facilitates editing of the read variables using JavaScript code. The script is executed when the variable values change. The current value is accessed via "value". Enter the script directly into the "Properties" field.



If a variable or a script is saved in both properties (variable and variable script), the "Variable script" property has priority.

3.5.10 Dynamic size

Variable width and height

The "Dynamic size" "Properties" group facilitates controlling the widget size.

The property "Variable width" facilitates linking with a variable. Via the  symbol, the process variable window is called. In this window, a variable can be selected from all global, internal and screen variables. The variable can also be entered manually into the field.

The linked variable controls the widget width.

The "Variable height" property facilitates linking to a variable. Via the  symbol, the process variable window is called. In this window, a variable can be selected from all global, internal and screen variables. The variable can also be entered manually into the field.

The linked variable controls the widget height.

Thus, widgets can be actuated by using an input field so that the width and height changes, depending on the value.

Variable width of the script and variable height of the script

Facilitates editing of the read variables using JavaScript code. The script is executed when the variable values change. The current value is accessed via "value". Enter the script directly into the "Properties" field.



If a variable or a script is saved in both properties (variable and variable script), the "Variable script" property has priority.

3.5.11 Dynamic rotation

Variables and variable script

The "Dynamic rotation" "Properties" group facilitates controlling the widget rotation.

The "Variable" property facilitates linking to a variable. Via the  symbol, the process variable window is called. In this window, a variable can be selected from all global, internal and screen variables. The variable can also be entered manually into the field.

The linked variable controls the widget rotation.

The "Variable script" property facilitates editing of the read variables using JavaScript code. The script is executed when the variable values change. The current value is accessed via "value". The script is directly entered in the attribute field.

Widgets get be actuated by using an input field so that the rotation changes, depending on the value.



If a variable or a script is saved in both properties, the "Variable script" property has priority.

Center of rotation

Facilitates the specification of a center of rotation. Use the drop-down menu to select a theme. Use the  symbol to open the drop-down menu..

3.5.12 Dynamic screen

The "Dynamic screen" "Properties" group facilitates controlling the screen widgets.

Screen

Facilitates specifying the screen to be selected. Use the drop-down menu to select a theme. Use the  symbol to open the drop-down menu.. The Web-Composer displays all created graphics of the current project tree.

Variables and variable script

The "Variable" property facilitates linking to a variable. Via the  symbol, the process variable window is called. In this window, a variable can be selected from all global, internal and screen variables. The variable can also be entered manually into the field.

The "Variable script" property facilitates editing of the read variables using JavaScript code. The script is executed when the variable values change. The current value is accessed via "value". The script is directly entered in the attribute field.

Thus, widgets can be actuated by using an input field so that the graphic changes, depending on the value.



If a variable or a script is saved in both properties (variable and variable script), the "Variable script" property has priority.

Operator and condition

The "Operator" property facilitates specifying a comparison operator. The following options are available: "Value is greater than or equal to condition", "value is smaller than or equal to condition", "value is equal to condition" or "value is unequal to condition". Use the drop-down menu to select a theme.

Use the  symbol to open the drop-down menu..

The "Condition" property facilitates entering a value to be compared to the value of the linked variable. Use the keyboard in the attribute field to select the property.

3.5.13 Events

See [chapter 4.7 "Triggering of events" on page 52](#)

3.5.14 Dynamic attribute

Dynamic attribute of a "Diagram" widget

The "Dynamic attribute" properties shows all widget-specific properties that cannot be assigned to any other "Properties" area.

Amongst others, facilitates selecting a diagram type (e.g. line diagram, bar diagram, etc.) or assigning a name to the x-axes and y-axes.

Dynamic attribute of a round "progress bar" widget and a "progress bar" widget

The "Maximum" property facilitates specifying a value that specifies when the bar is to reach 100%.

Dynamic attributes of a "Frame" widget

Amongst others, facilitates specifying a screen that is to be referenced and displayed.

Dynamic attributes of a "Switch" widget

Amongst others, facilitates linking of a variable, specifying a refresh interval or labeling the on/off switch.

Dynamic attributes of a "Table" widget

Amongst others, facilitates linking of a variable, the graphic table customization or specifying the number of columns and lines.

Dynamic attributes of a "Text box" widget

Amongst others, facilitates linking of a variable, specifying a refresh interval and entering a header.

3.5.15 Widget attribute

The "Widget attribute" properties shows all widget-specific properties that cannot be assigned to any other "Properties" area.

Widget attribute of an "Input field extended" widget

Amongst other, facilitates specifying labels or units.

3.6 Process variables

This process variable browser is displayed in the tool window of IndraWorks Engineering. The displayed variables are part of the currently opened screen.

The screen provides all local and global visualization variables for selection. This display changes dynamically depending on the selected screen.

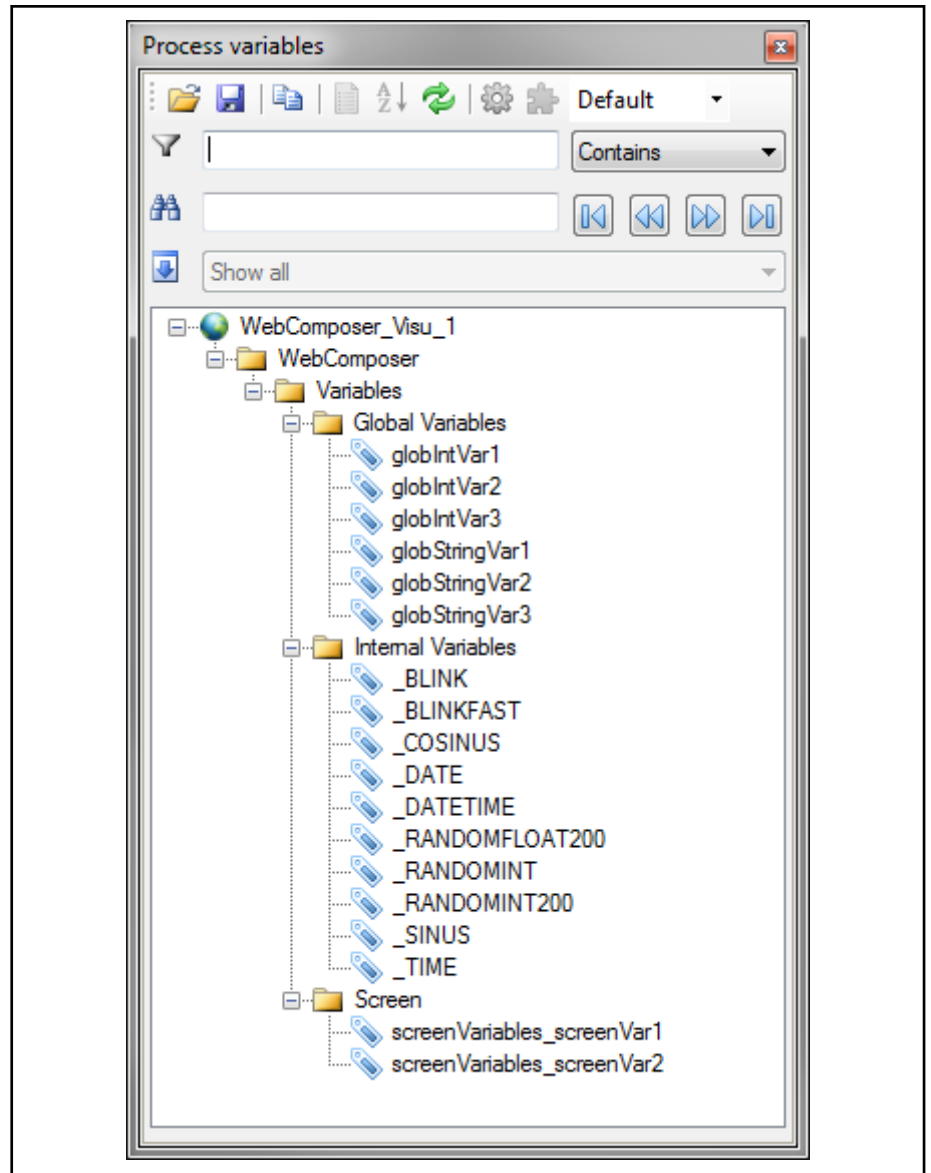


Fig. 3-15: Process variables

- **Global variables**
All variables created under "Global data/Variables/VarDef".
If no global variables have been created, the "Global Variables" branch is not displayed in the process variable browser.
- **Internal variable**
The internal variable are static and always available.
- **Screen variables**
The screen variables are variables of the currently displayed screen.
If no screen variable has been created, the branch in the project variable browser is not displayed.

The process variable screen changes dynamically if another screen is brought to the foreground.

3.7 Data and functions

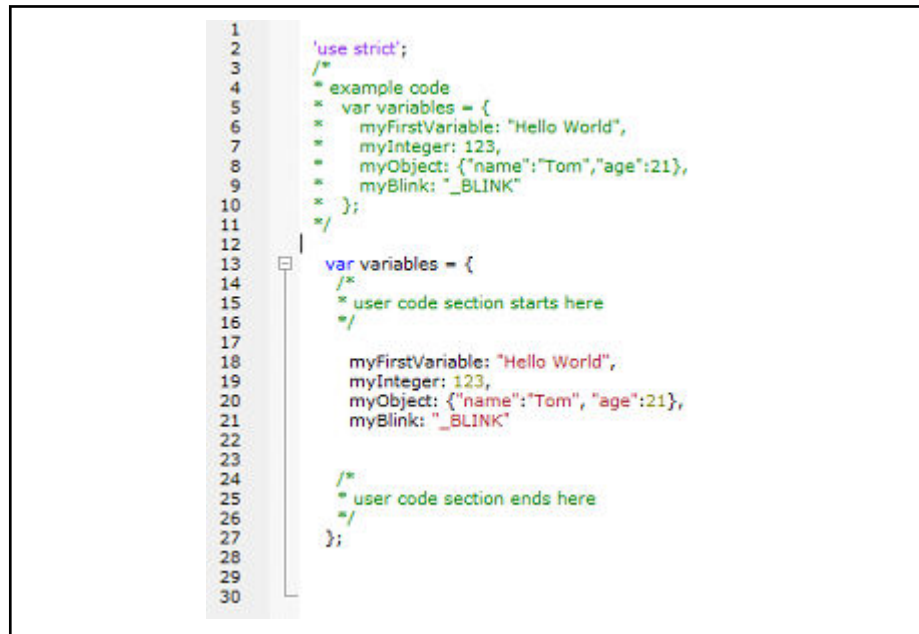
3.7.1 Global variables

The template for global variables is created automatically when creating a project and cannot be deleted by the user.

The global variables can be found in the project tree under **Global data ▶ Variables ▶ VarDef** and can be edited using the script editor.

Global variables can be defined using this editor.

These variables are available to all components in the entire project.



```

1      'use strict';
2      /*
3      */
4      /* example code
5      * var variables = {
6      *   myFirstVariable: "Hello World",
7      *   myInteger: 123,
8      *   myObject: {"name":"Tom","age":21},
9      *   myBlink: "_BLINK"
10     * };
11     */
12
13     var variables = {
14     /*
15     * user code section starts here
16     */
17
18     myFirstVariable: "Hello World",
19     myInteger: 123,
20     myObject: {"name":"Tom","age":21},
21     myBlink: "_BLINK"
22
23
24     /*
25     * user code section ends here
26     */
27     };
28
29
30

```

Fig. 3-16: Global variables

3.7.2 Global main functions

The template for global main functions is created automatically when creating a project and cannot be deleted by the user.

The global main functions can be found in the project tree under **Global data ▶ Scripts ▶ Main functions** and can be edited using the script editor.

The global main functions are defined and should not be deleted.

Additional global main functions cannot be created.

Global main functions are available to all components in the entire project.

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47

'use strict';
/*
 * accessing the global functions and variables:
 * globalFunctions.<name_of_function>
 * globalVariables.<name_of_variable>
 */
var mainScript = {
  /* this function will automatically called on page load */
  onApplicationLoad: function() {
    /* start of function implementation */

    /* end of function implementation */
  },

  /* adjust cycle time (ms)*/
  /* set value to -1 or 'none' to disable the cyclical execution */
  whileApplicationRunTimer: 1000,

  /* this function will automatically called while the application is running */
  whileApplicationRun: function () {
    /* start of function implementation */

    /* end of function implementation */
  },

  /* this function will automatically called on page unload */
  onApplicationUnload: function () {
    /* start of function implementation */

    /* end of function implementation */
  }
};

```

Fig. 3-17: Global main functions

3.7.3 Global functions

The template for global functions is created automatically when creating a project and cannot be deleted by the user.

The global functions can be found in the project tree under **Global data ▶ Scripts ▶ Functions** and can be edited using the script editor.

At this point, all global functions required for the application have to be implemented.

Global functions are available to all components in the entire project.

```

1  'use strict';
2  /*
3   * accessing the global variables:
4   * globalVariables.<name_of_variable>
5   */
6
7
8  /*
9   * function example code
10   * var functions = {
11   *   myFirstFunction:function(argument1,argument2,argument3)
12   *   {
13   *     return argument1 + argument2 + argument3;
14   *   },
15   *   mySecondFunction:function()
16   *   {
17   *     alert('test');
18   *   }
19   * }
20   * return functions;
21   */
22  var functions = {
23    /*
24     * user code section starts here
25     */
26
27
28    /*
29     * user code section ends here
30     */
31  };
32
33
34

```

Fig. 3-18: Global functions

3.7.4 Screen variables

A template for the screen variables can be created under each screen and can be deleted again if required.

The screen variables are edited using the script editor (see [chapter 3.3.2 "Script editor" on page 16](#)).

The variables are only available for the superordinate screen. Other components (screen, scripts,...) cannot access these variables.

If the superordinate screen is deleted, the screen variables are also automatically deleted.

The screen variable syntax corresponds to the syntax of the Global Variables (see [chapter 3.7.1 "Global variables" on page 30](#)).

3.7.5 Screen scripts

A template for the screen script can be created under each screen and can be deleted again if required.

The screen script is edited using the script editor (see [chapter 3.3.2 "Script editor" on page 16](#)).

A screen script is only available for the superordinate screen. Other components (screen, scripts,...) cannot access this script.

If the superordinate screen is deleted, the screen script is also automatically deleted.

The screen script syntax corresponds to the syntax of the Global Variables (see [chapter 3.7.3 "Global functions" on page 31](#)).

3.7.6 Internal variable

Internal variables are a fixed set of variables provided by the WebComposer.

These variables can be used by all components.

There is **no** option available to edit internal variables.

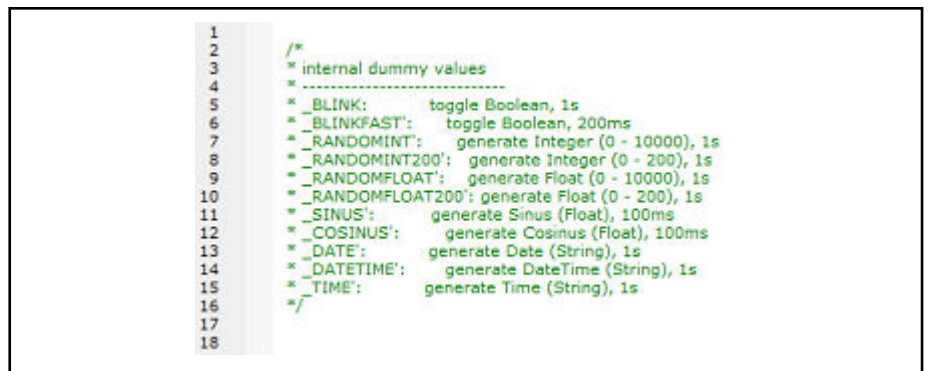


Fig. 3-19: Internal variable

3.7.7 Find & Replace function

By using the "Find & Replace" functionality, script files can be searched for texts or in screen files, property values can be searched and replaced.

Find "Right-click" on the WebComposer commissioning visualization in the project tree.

Select "Find...". The "Find" function is opened.

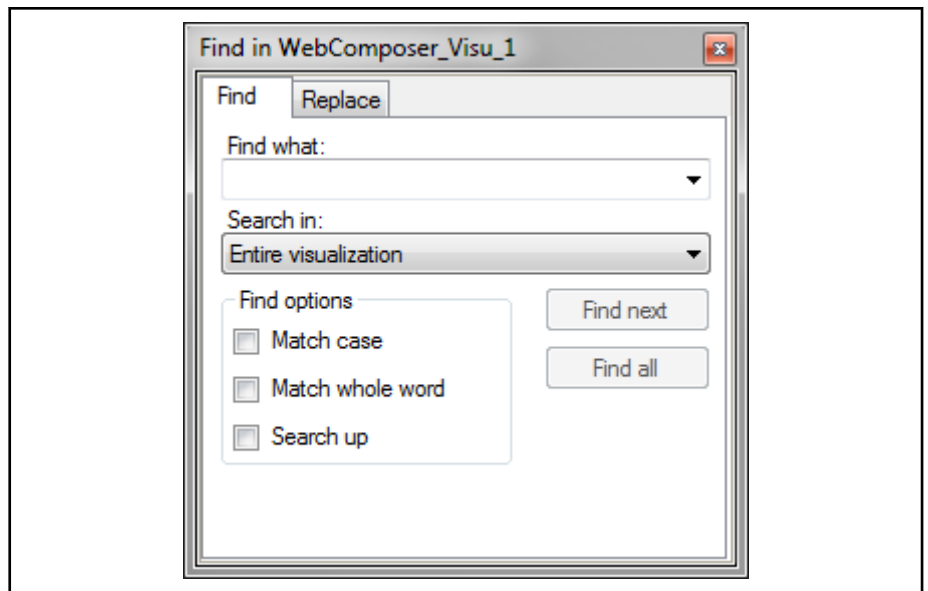


Fig. 3-20: "Find" window

The text or numbers to be searched for can be entered under "Find what".

Furthermore, there are several options to define the search.

Options for a more detailed definition:

- **Upper and lower case:** Upper and lower case is taken into consideration for the search
- **Match whole word:** Only whole words are searched for
- **Search up:** Reverse search

Option of the "Search in" function

- **Active editor:** The search is carried out only in the active opened editor
- **All open editors:** All opened editors are searched
- **Entire visualization:** The entire visualization is searched

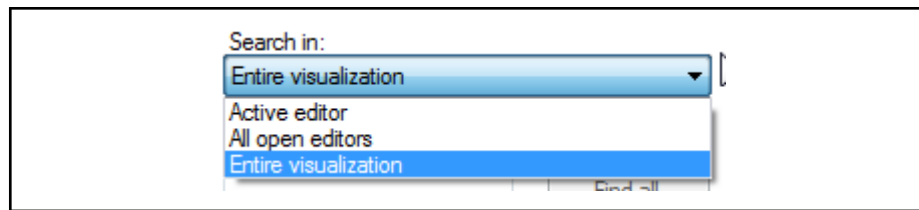


Fig. 3-21: "Search in" function

Search options:

- **Find next:** Only the next occurrence is searched for. The editor is opened and the text or number is highlighted. By "clicking" on the function, the next occurrence is displayed.
- **Find all:** Each occurrence of the searched content - depending on the selected search options - is listed in the message window.

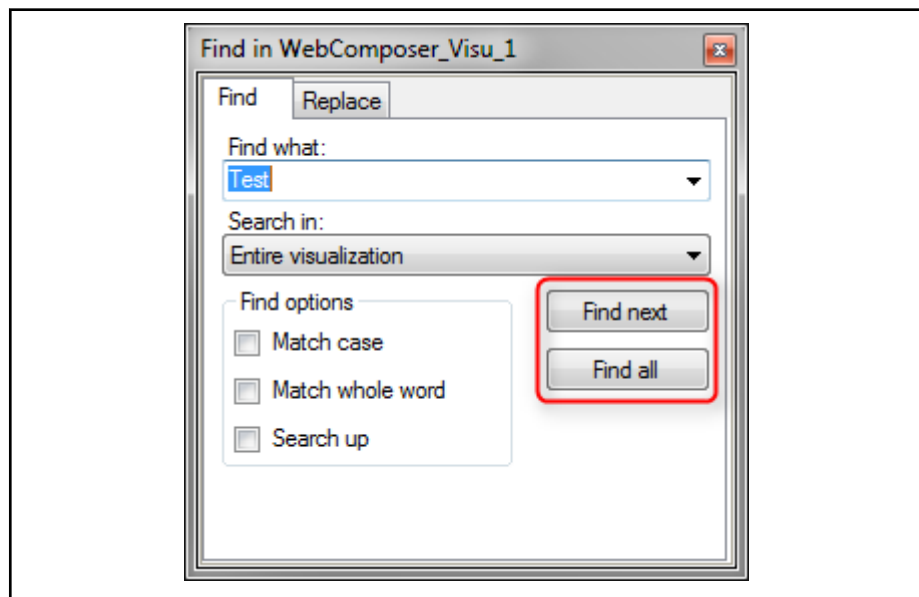


Fig. 3-22: Option of the Find in function

Replace Right-click on the WebComposer commissioning visualization in the project tree to select the "Replace..." function in the context menu.

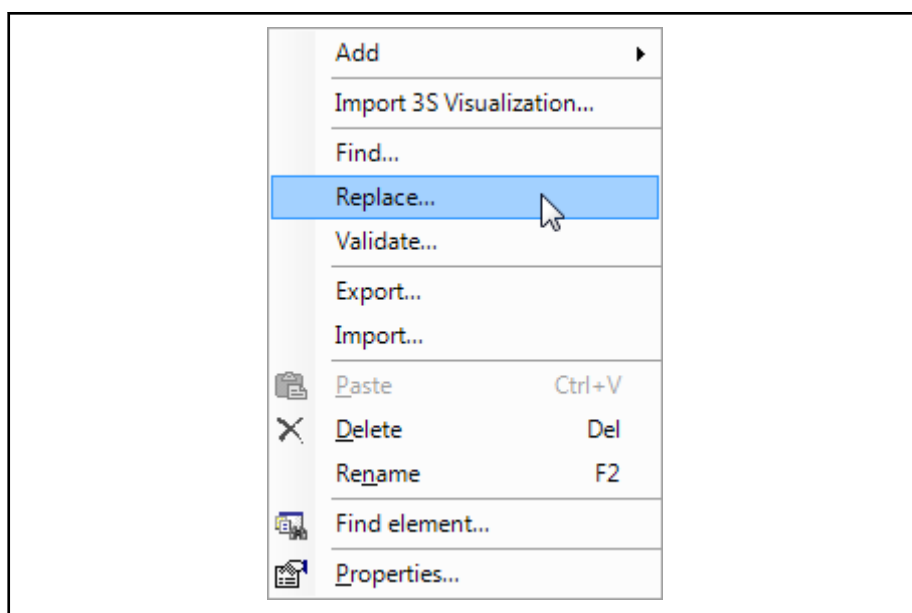


Fig. 3-23: Replacing via the context menu

The "Replace" window opens.

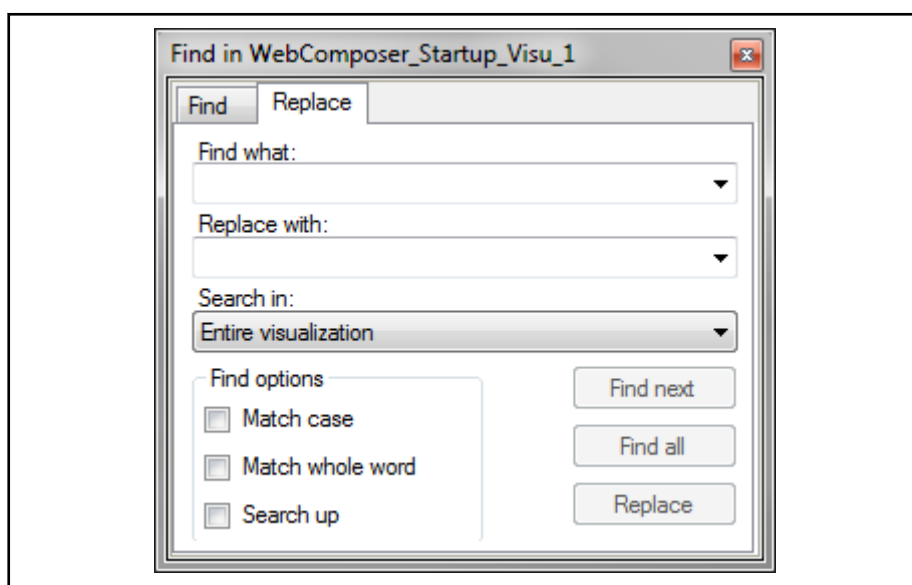


Fig. 3-24: "Replace" window

The text or numbers to be searched for can be entered under "Find what".
Enter text or numbers to replace a word with in the "Replace" window.



If the "Replace with" is left blank, the characters are deleted.

Furthermore, the following options are available to further define the search:

- **Upper and lower case:** Upper and lower case is taken into consideration for the search
- **Match whole word:** Only whole words are searched for
- **Search up:** Reverse search

Under "Find what", the following options can be selected:

- **Active editor:** The search is carried out only in the active opened editor
- **Open all editors:** All opened editors are searched
- **Entire visualization:** The entire visualization is searched

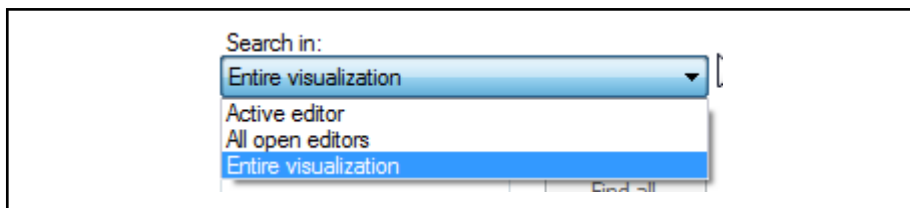


Fig. 3-25: "Search in" function

More options:

- **Replace:** The currently displayed occurrence is displayed
- **Find next:** Only the next occurrence is searched for. The editor is opened and the text or number is highlighted. By "clicking" on the function, the next occurrence is displayed.
- **Find all:** Each occurrence of the searched content - depending on the selected search options - is listed in the message window.



All found occurrences are displayed in the message window by specifying the ID, the description, the element, the position and the path.

3.7.8 Validation function

This function executes a complete check of the selected project element. The result of the validation is displayed in the message window.

"Double-click" on a message text in the message window to open and visualize the corresponding script or screen in the respective editor.

Dependencies/references between the configured data within the entire device are checked.

When the following conditions are fulfilled, data consistency is ensured:

- All referenced variables exist and are valid
- All referenced graphics exist and are valid
- All referenced graphics exist and are valid
- All referenced script functions exist and are valid
- The script syntax is correct

To check for errors in the WebComposer data project planning, the **Validate** function can be executed at any time via the WebComposer commissioning visualization context menu in the project tree.

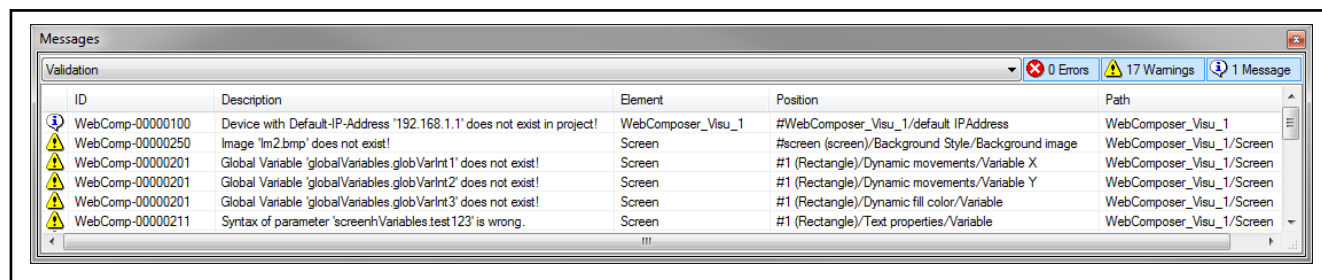


Fig. 3-26: Output of messages when validating WebComposer data

Output of rule violations in message window

All detected problems are output in the message box. The following information is displayed in each message:

- **Type:** This information classifies the message in errors, warnings and information. Errors can result in incorrect or unpredictable behavior in the browser and should be fixed before using the screens. Warnings refer to problems that do not impair the browser but individual functions cannot be executed correctly. Information references to further inconsistencies that do not have any effect on the browser.
- **ID:** Error number used to retrieve further information on cause and fixing in the [tab. 3-2 "Overview on the error IDs of the validation messages" on page 37](#) table
- **Description:** Short description of the message
- **Element:** Name of element containing the position the problem occurred in
- **Position:** Additional information on where the problem occurs within the element (e.g. line number in the script, widget attributes)
- **Path:** Storage location of the element in the WebComposer device

Double-click on the message line to quickly navigate to the position the problem occurred in. The editor opens in which the problem can be fixed. In some cases, no editor is available. In this case, a dialog containing more information on error recovery is displayed. After fixing one or several problems, call the **Validate** function to validate the correction.

Overview on error numbers

Each detected problem during the validation of the visualization data is assigned an ID.

Use this ID to find information on cause and remedy in the following table:

Type	ID	Description	Cause	Recovery
Warning	Web-Comp-00000100	The referenced variable "{0}" is invalid	In the screen/widget properties, a variable that does not exist is referenced.	Use the process variable browser to select a valid variable.
Warning	Web-Comp-00000101	The referenced global variable "{0}" is invalid.	In the screen/widget properties, a global variable that does not exist is referenced.	Use the process variable browser to select a valid variable or open the global variable using the script editor and create the variable.
Warning	Web-Comp-00000102	The referenced internal variable "{0}" is invalid.	In the screen/widget properties, an internal variable that does not exist is referenced.	Use the process variable browser to select a valid internal variable.
Warning	Web-Comp-00000103	The referenced screen variable "{0}" is invalid.	In the screen/widget properties, a screen variable that does not exist is referenced.	Use the process variable browser to select a valid variable or open the local screen variable using the script editor and create the variable.
Warning	Web-Comp-00000104	The referenced PLC variable "{0}" is invalid.	In the screen/widget properties, a PLC variable that does not exist is referenced.	Use the process variable browser to select a valid PLC variable.

Type	ID	Description	Cause	Recovery
Warning	Web-Comp-00000110	The referenced graphics "{0}" is invalid.	In the screen/widget properties, a graphics that does not exist is referenced.	Select an existing graphics via the context menu of the property or insert a new graphic under the specified name via the context menu of the "Graphics" node.
Warning	Web-Comp-00000120	The referenced screen "{0}" is invalid.	In the widget properties, a screen that does not exist is referenced.	Select an existing screen via the context menu of the property.
Warning	Web-Comp-00000130	The referenced theme "{0}" is invalid.	In the screen/widget properties, a theme that does not exist is referenced.	Select an existing theme via the context menu of the property.
Error	Web-Comp-00000200	Syntax error in script: "{0}"	Syntax error in the specified line/column of the specified script.	Fix the syntax error in the script.

Tab. 3-2: Overview on the error IDs of the validation messages

3.7.9 Import/export function

By using this function, the entire project or only certain project elements can be imported or exported.

3.7.10 Selecting customized screens

Use this function to select one or multiple screens from the current visualization to create a customized package including all required data for display in the WebAssistant.

In the package, all global data is also always automatically applied in addition to the selected screens (refer to [chapter 3.2.2 "Global data" on page 9](#)). If screen variables (see [chapter 3.7.4 "Screen variables" on page 32](#)) or screen scripts (see [chapter 3.7.5 "Screen scripts" on page 32](#)) are added, they are also included in the package.

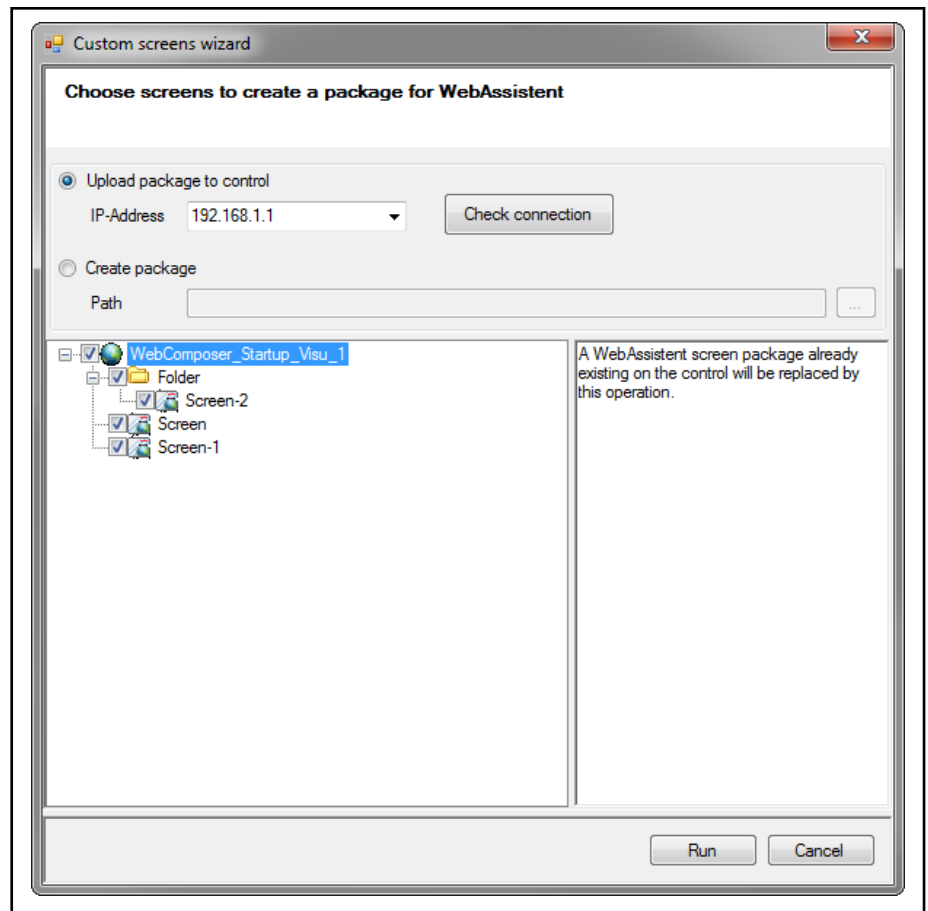


Fig. 3-27: Wizard to create customized screens

Specifying target for package

Specify the storage location of the package to be created in the upper third of the dialog. There are two alternatives:

- **Upload package to control**

In this case, the package is transferred to the control with the specified address via FTP/SFTP. A WebAssistant has to be installed on the control. In all other cases, or if not related to a control or if the IP address cannot be reached, an error message is output.

Press the button "Check connection" to check the connection to the selected IP address in advance. A green tick shows a valid connection. A red cross shows that there is no connection.

- **Create package**

In this case, the package is created and stored as specified in the "path". The user can also transfer this package to the control using the "Import" function of the WebAssistant.

Selecting screens

All folders and screens of the selected visualization device are displayed in the tree view of the dialog. Global data and existing screen variables and/or screen scripts are not displayed. Select all required screens for the package via the selection fields now. If a parent element is selected in the tree, all child elements are also selected. The same happens when the selection is removed. All child objects are also deselected at the same time.

Actions

- **Run**

Press this button to start the creation and provision of the package as per the settings made (see preceding). A progress bar is displayed while

creating the package that can be aborted via "Cancel". If no errors occur, the package is stored under the defined path. The executed actions are summarized in the dialog.



Before the package is created, the interdependencies of the screens and their references are checked.

If all screens with references to other screens were selected and if these are not part of the selection, a message is issued. All missing screens can now be applied automatically.

- **Cancel**

Press this button to cancel the complete process and to close the dialog.

3.7.11 3S visualization import

Use this function to import visualizations below the logic node into the WebComposer. For more information, see [chapter 4.14 "3S visualization import" on page 69](#)

3.8 Preview

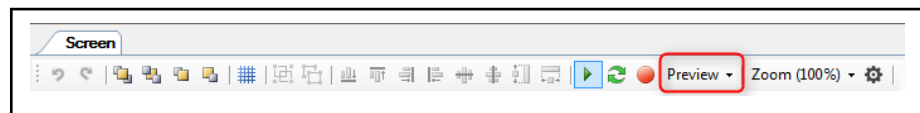


Fig. 3-28: Preview function in the symbol bar

This function opens the current project status in a browser. The user can select between "Open preview" or "Open preview with".

- "Open": The default browser is used
- "Open with": Any browser can be selected

In both cases, the current screen status is displayed.

While the preview is displayed, the screen can still be configured.

Use the "Refresh function"  (see: [tab. 3-1 "Symbols of the symbol bar of the screen editor" on page 15](#)), the preview status is adjusted to the current screen status in the WebComposer.

4 Use cases

4.1 Creating a WebComposer commissioning visualization



IndraWorks Engineering provides two options to create a WebComposer commissioning visualization:

- Via the main menu and the context menu
- Via "drag and drop"

Via the menu and the context menu

1. In the main menu, select "Project" or right-click on "Project" in the Project Explorer.
2. Select "WebComposer visualization" via the subitem "Add".

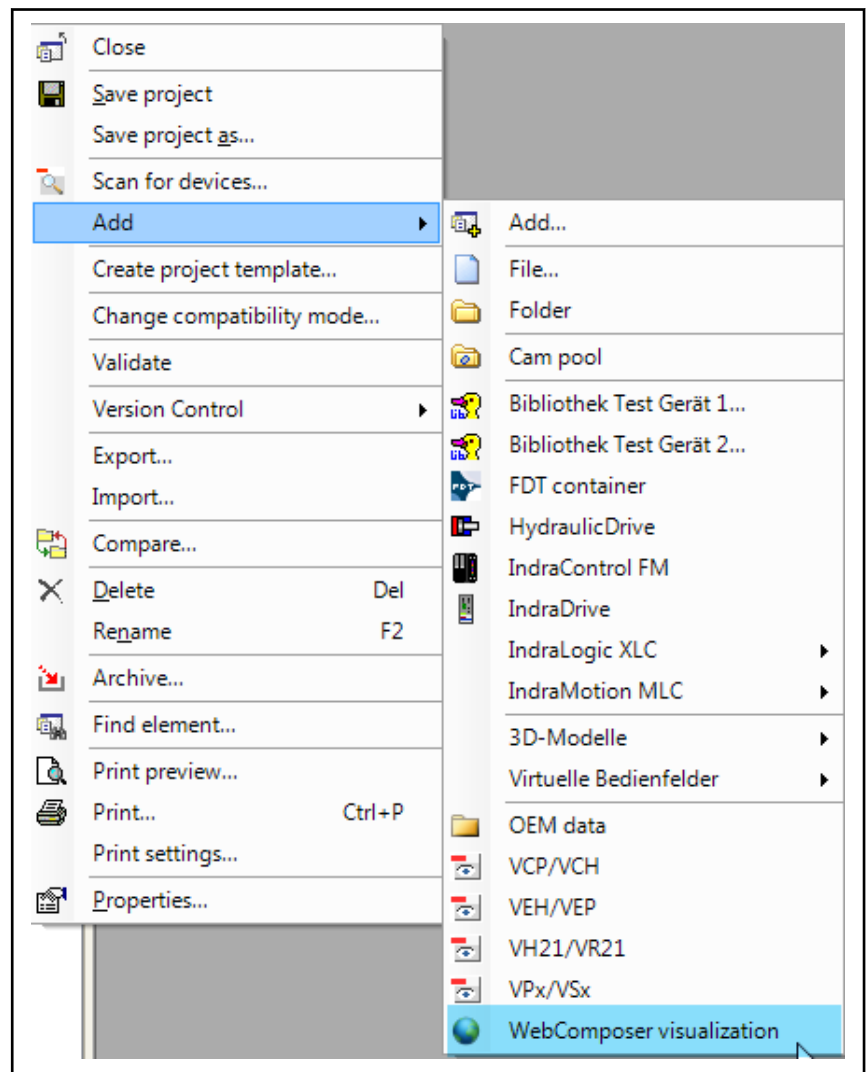


Fig. 4-1: Adding WebComposer commissioning visualization via the main menu

Use cases

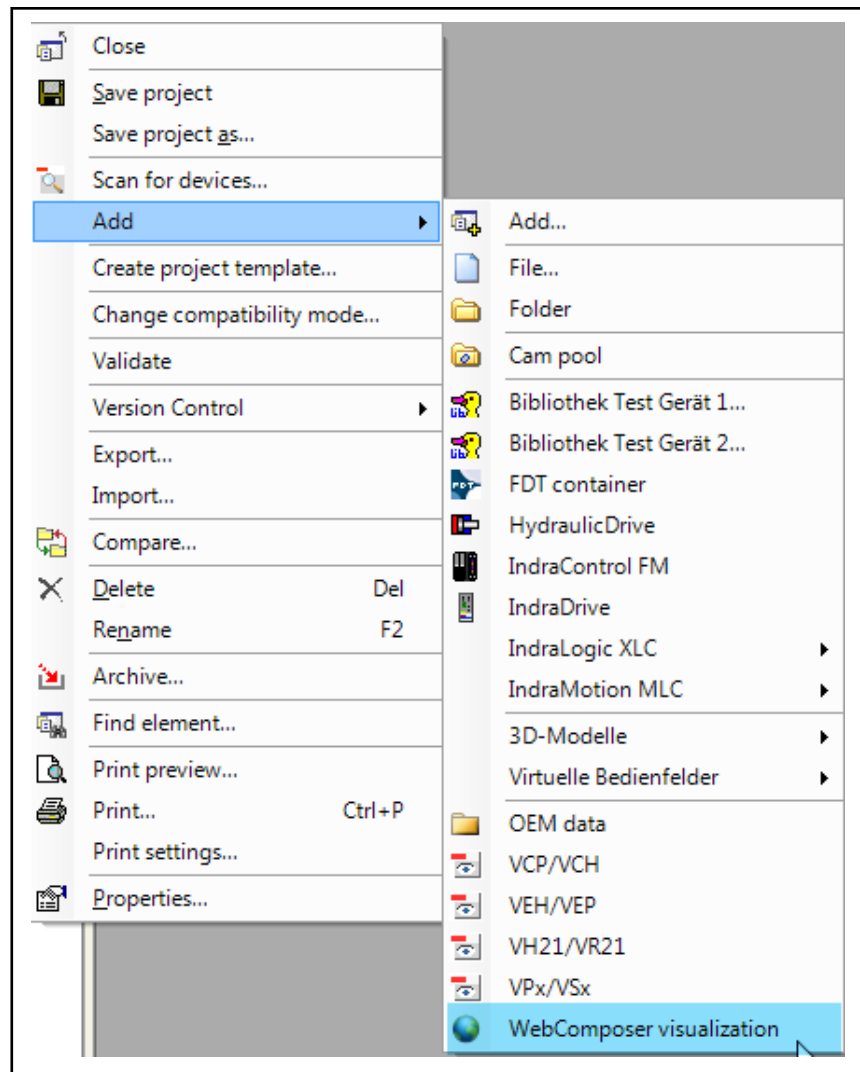


Fig. 4-2: Adding WebComposer commissioning visualization via the context menu

- Via "drag and drop" 1. In the "Library" tool window, open "Visualization".

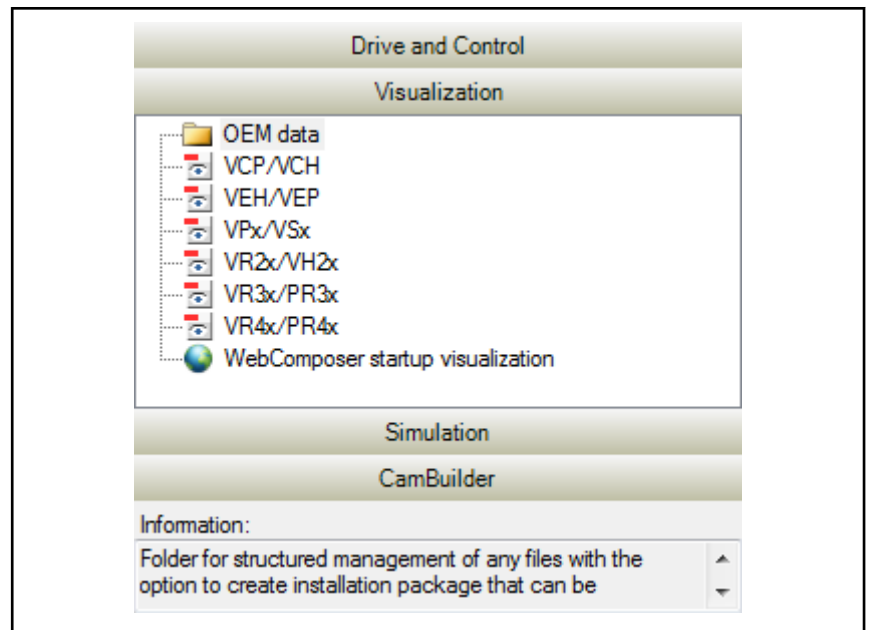


Fig. 4-3: Area "Visualization" in library

2. Drag the "WebComposer visualization" element to the project in the project tree using "drag-and-drop".



- The "Global data" subfolder is automatically added.
- The number of WebComposer commissioning visualizations in an IndraWorks project is not limited.

4.2 Configuring the connection to the control

The connection to a control is established via the IP address of the control. The IP address of the control is entered in the Properties dialog when creating the WebComposer device. This Property dialog can be called via the context menu of the WebComposer device and the IP address can be entered or changed.

By default, the set IP address applies to all screens created in the WebComposer device.

Configuring the connection to the control for individual screens

In the WebComposer, the IP address of the control can be entered for an individual screen. The default IP address is thus overwritten. This screen-specific IP address is entered in the attribute window of the screen in the communication settings area. The screen attributes are displayed when clicking on the screen background.

4.3 Position widgets on a screen



Widgets can only be positioned on a screen in "Design mode" (see [tab. 3-1 "Symbols of the symbol bar of the screen editor" on page 15](#)).

1. Create a new screen in the project tree and open it.
2. Bring the WebComposer library to the foreground
 - Double-click on the screen
 - Mouse-click on the "WebComposer library" tab in the tool window

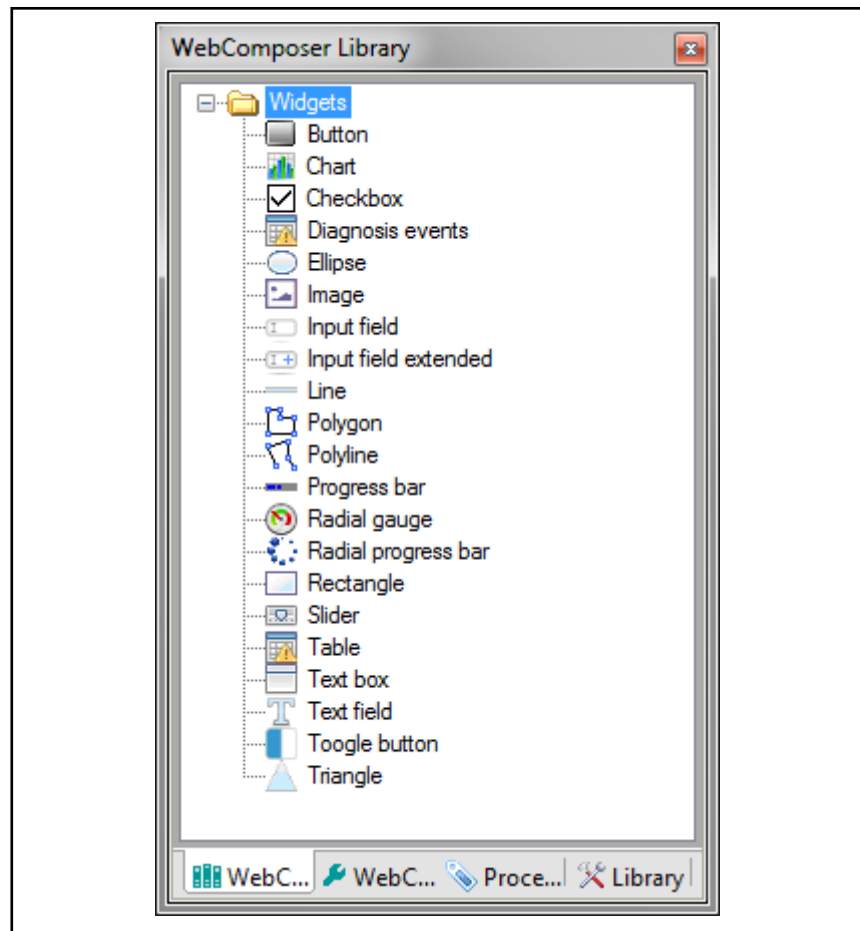


Fig. 4-4: WebComposer library

3. Drag the widget to the screen via "drag and drop".
4. There are several options to position the widget in the screen:
 - Use the mouse pointer - click on the widget and drag it to the location on the screen
 - Via the "WebComposer properties" in the tool window, change the "XPosition" and "YPosition" property to the desired values under "Position"
 - Via the keyboard cursor

A widget can be moved faster using the shortcut cursor + "Alt"



- The widget library is only displayed when the screen is opened for the first time.
- Widgets can be deleted, moved or edited in the screen at any time



The number of widgets on a screen is not limited.

4.4 Using screens in screens

4.4.1 Referencing screens

Via drag and drop Screens can be displayed or referenced in another screen using drag-and-drop in the WebComposer.



Steps 1-3 only have to be complied with if no screen has been created.

If the screen on which the frame is to be generated is already created and opened, directly proceed to step 4.

1. In the context menu, right-click on the WebComposer commissioning visualization or on a previously added folder.
2. Select "Screen" via the subitem "Add".

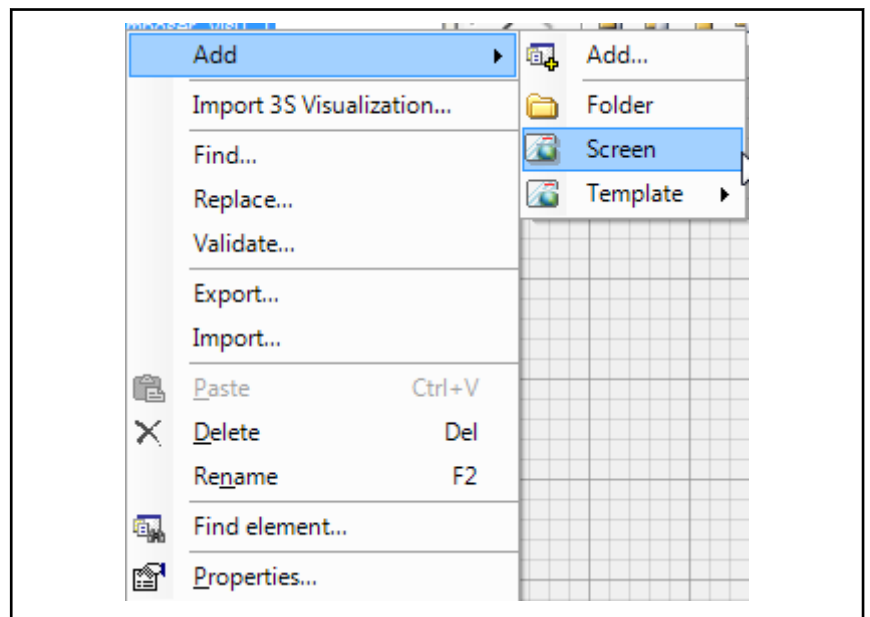


Fig. 4-5: Add screen via context menu

The WebComposer now added the screen to the project tree.

A template can also be added to the project tree. The template is automatically converted to a screen element and can subsequently be used as regular screen.

3. Open the created screen in which another screen is to be inserted and referenced.
4. Drag the screen to be inserted and referenced from the project tree to the opened screen by pressing the left mouse key and via "drag and drop". A "Frame widget" is automatically generated in the screen, the screen is referenced and displayed.

Via the WebComposer properties

1. Drag a frame widget to the desired screen (as described above).
2. Double-click on the frame widget to display the frame widget properties
3. In the drop-down box of the "Screen name" property under "Frame attributes", select the screen to be referenced.

Thus, the referenced screen or template is displayed in the selected frame widget.

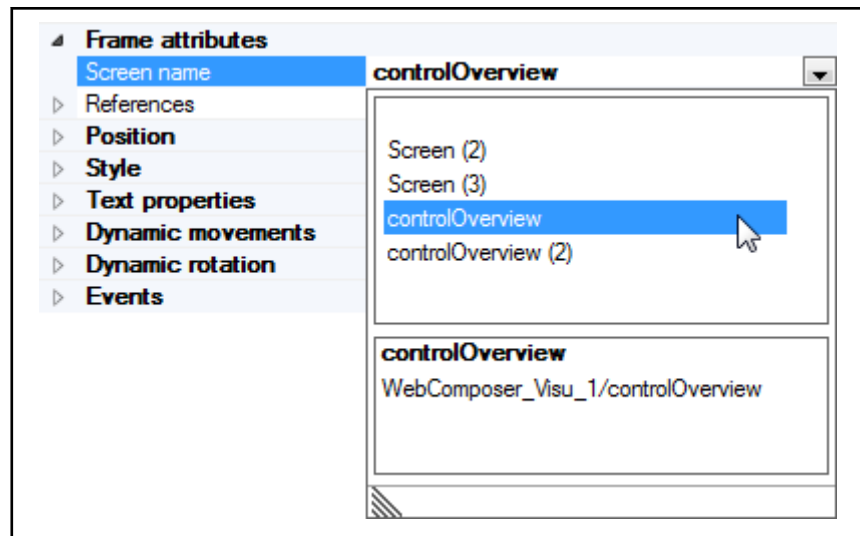


Fig. 4-6: Add screen via the WebComposer properties



The drop-down box of the "Screen name" property contains a list with all screens created in the current project.

4.4.2 Editing referenced screens

1. Double-click on the screen to be edited or right-click on the screen in the context menu and select "Open".
2. Edit the screen (deleting and adding widgets, repositioning widgets, etc.) - see [chapter 4.3 "Position widgets on a screen" on page 43](#).
3. Save the changes.
4. Open the screen representing the template.
5. • If the screen containing the reference to the template is opened,  has to be refreshed via the symbol in the menu bar.
The changes to the screen are now applied to all screens representing the template.
- If the screen containing the reference to the template is closed, the changes are automatically refreshed when opening the screen.

4.4.3 User example axis overview

Using the example of the "axis overview", it is explained how axis data can be generated in a screen once and can subsequently be used multiple times via an index (axis address).

1. Creating a new screen containing axis data.

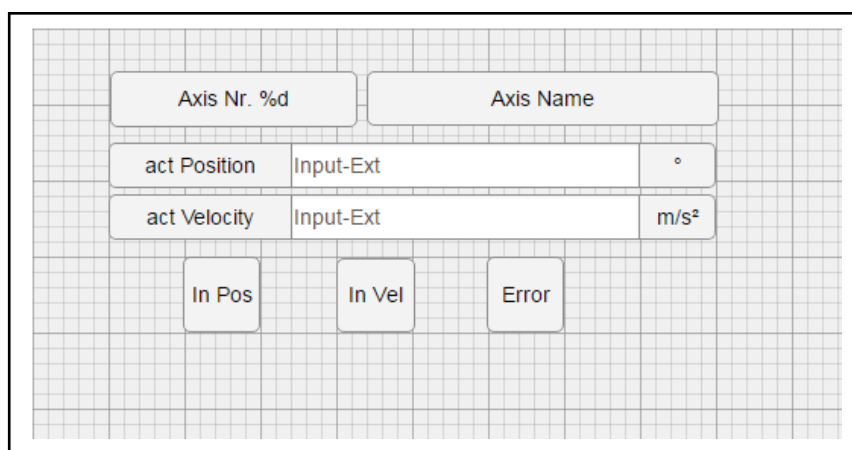


Fig. 4-7: Screen to display axis data

2. Creating new screen variables "axisNr" and "axisName"

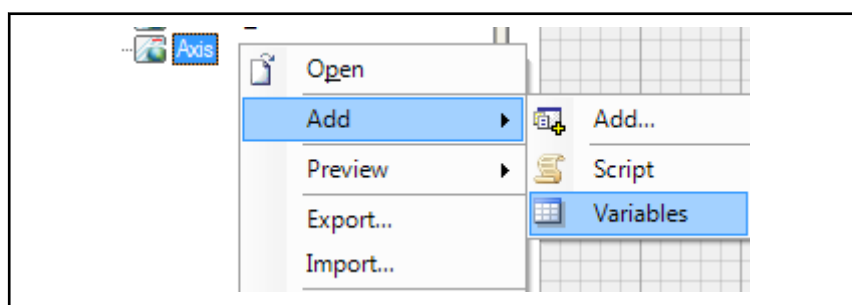


Fig. 4-8: Adding screen variables

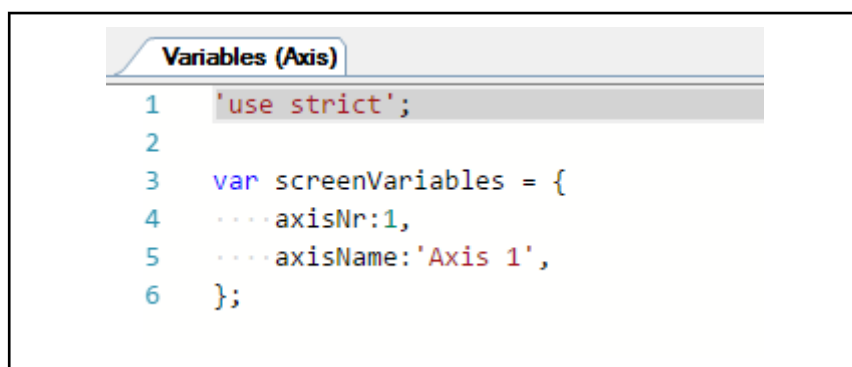


Fig. 4-9: Configuring screen variables

3. Connecting the axis data via the screen variable "axisNr."

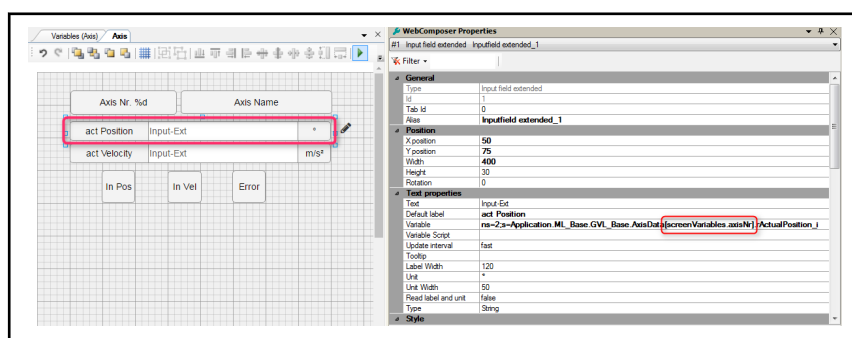


Fig. 4-10: Connecting the axis data

4. Creating and configuring the axis overview
 - Adding a new screen "Axis overview"

- Opening the "Axis overview" screen
- Adding two instances of the axis data screen using drag-and-drop
- Setting the frame attribute "axisNr" and "axisName"

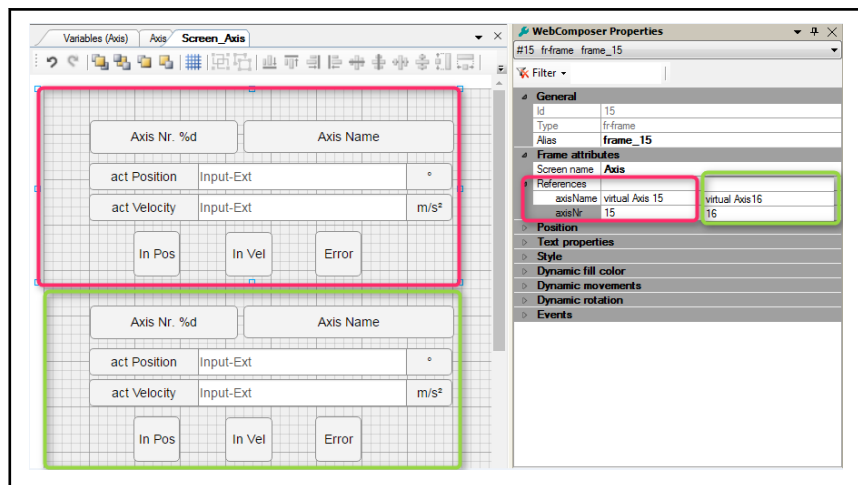


Fig. 4-11: Configuring the referenced frame attributes

5. Displaying the axis data in run mode



Fig. 4-12: Displaying the axis data in runtime mode

4.4.4 User example PLC array structure element

Using the example, it is explained how structure elements of an array can be generated in a screen once and can subsequently be used multiple times via the array index.

1. Creating a new screen containing structure data.

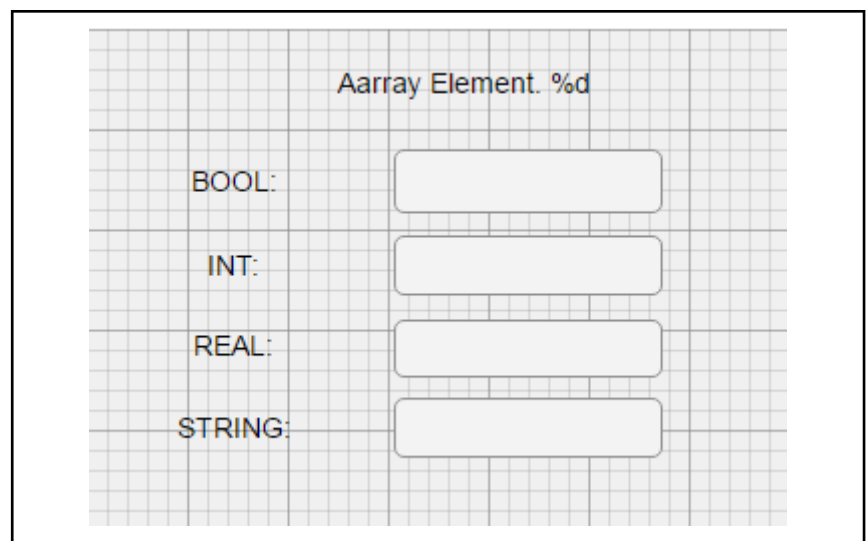


Fig. 4-13: Structure data display

Creating a structure in the PLC program

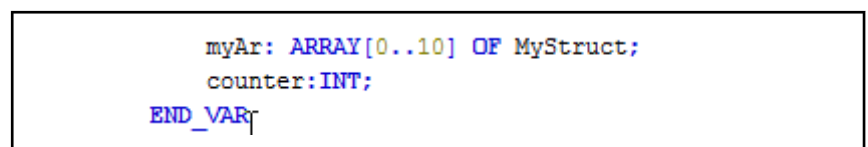


Fig. 4-14: Array with structure element

2. Creating a new screen variable "arIndex"

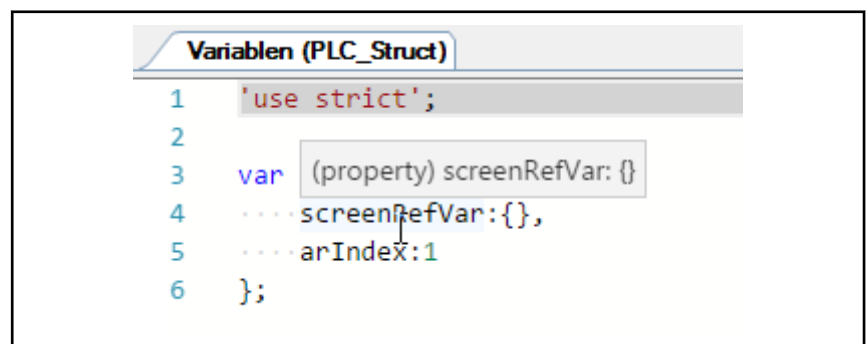


Fig. 4-15: Configuring screen variables

3. Connecting the relevant PLC data via the screen variable "arIndex".

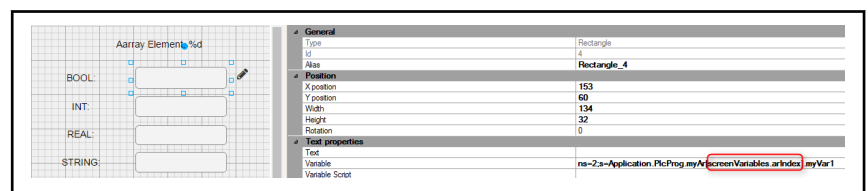


Fig. 4-16: Connecting the PLC structure data

4. Creating and configuring the PLC data overview screen
 - Adding and opening a new screen "PLC data overview"
 - Adding instances of the "PLC data screen" using drag-and-drop
 - Setting the frame attribute "arIndex"

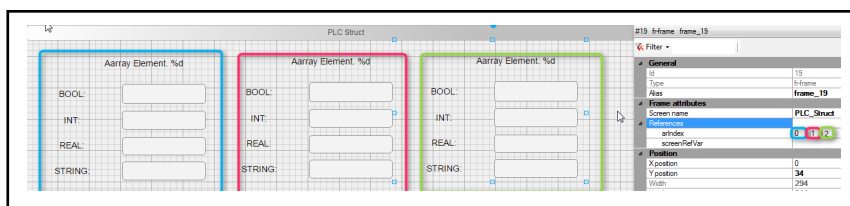


Fig. 4-17: Configuring the referenced frame attributes

5. Displaying the PLC data in run mode

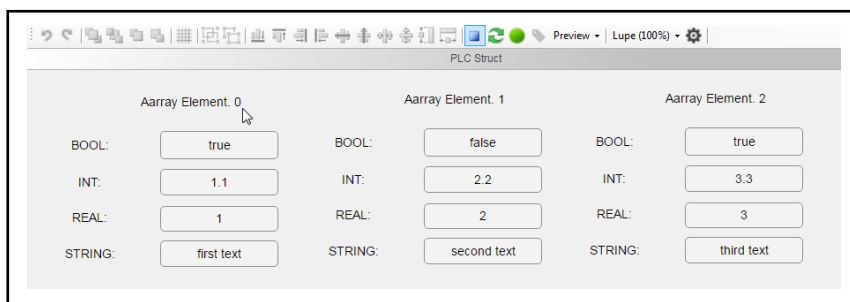


Fig. 4-18: Displaying the PLC data at runtime

myAr	ARRAY [0..10] OF MyStruct	
myAr[0]	MyStruct	
myVar1	BOOL	TRUE
myVar2	INT	1
myVar3	REAL	1.1
myVar4	STRING	'first text'
myAr[1]	MyStruct	
myVar1	BOOL	FALSE
myVar2	INT	2
myVar3	REAL	2.2
myVar4	STRING	'second text'
myAr[2]	MyStruct	
myVar1	BOOL	TRUE
myVar2	INT	3
myVar3	REAL	3.3
myVar4	STRING	'third text'

Fig. 4-19: PLC programming system

4.5 Linking widget attributes to variables



The WebComposer provides three options to link widget attributes to variables:

- Via the WebComposer properties
- Move to the widget via "drag and drop"
- Move to WebComposer properties via "drag and drop"

Via the WebComposer properties

1. Double-click on a widget to open the WebComposer properties.
2. Select the attribute to be linked to a variable.
3. Double-click on the symbol .

The process variable window opens. All internal variables and global variables created by the user are contained in this window.

If a variable template was assigned to the screen and variables are defined in the screen, they are also contained in the process variable window.

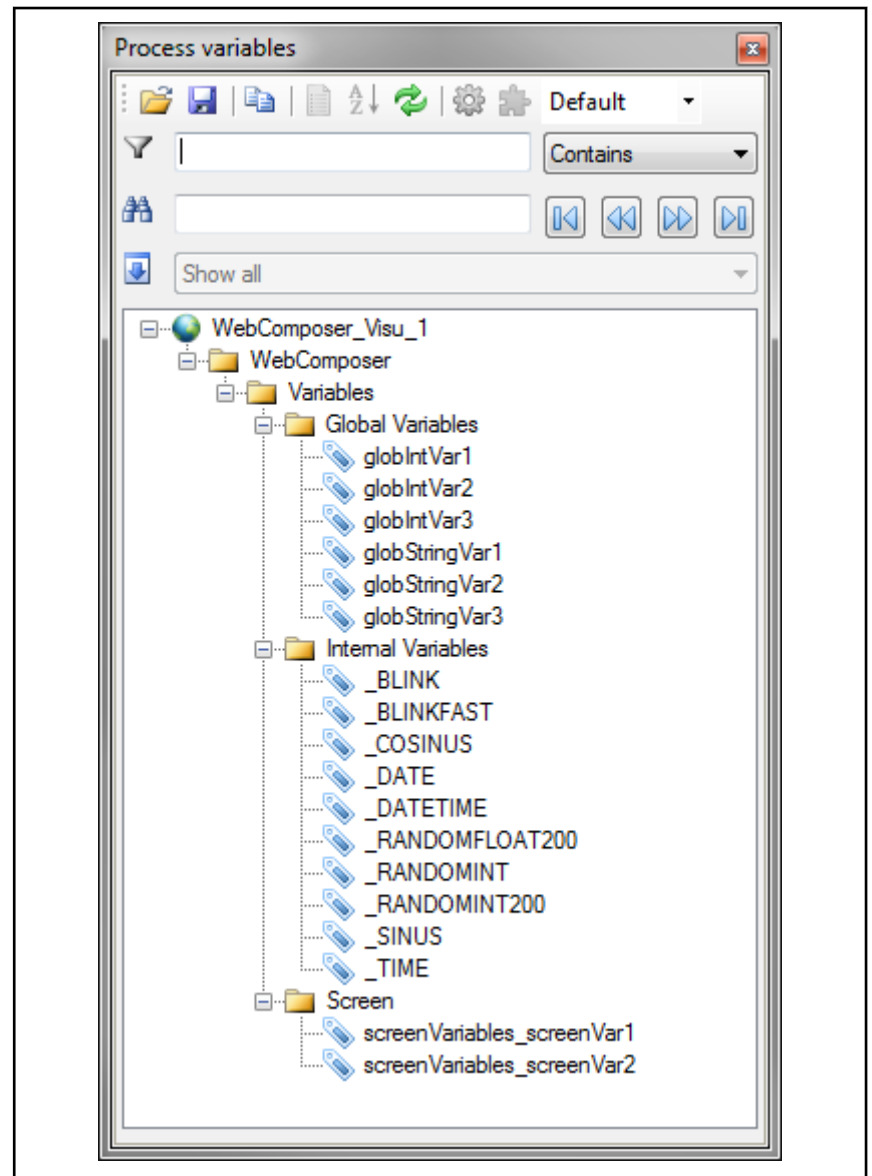


Fig. 4-20: Process variables

4. Select the desired variable and confirm with "OK".

The variable is not linked to the widget attribute.

Move to a widget via "drag and drop"

1. Open the "Process variable" tab.
2. Drag the variable to the widget in the screen via "drag and drop" that is to be linked to the variable.

The WebComposer now automatically links the "Text variable" attribute to the selected variable.



If the selected widget does not have a "Text variable" attribute, the variable is not linked to the widget.

Exception: In the "Screen widget", the "Dynamic screen variable" attribute is linked to the selected variable.

Move to WebComposer properties via "drag and drop"

1. Open the "Process variable" tab.
2. Open the "WebComposer properties" in a separate window.
3. Drag the variable to the attribute via "drag and drop".



Attributes that can be linked are visualized with the  symbol and attributes that cannot be linked are visualized with the  symbol when hovering over the attribute with the cursor.

4.6 Visualizing data in the internal or external web browser



The WebComposer provides two options to visualize data:

- Visualization in an internal browser
- Visualization in an external browser

Visualization in an internal browser

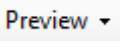
You are in "Editing mode"  (see [chapter 3.3.1 "Screen editor" on page 14](#)).

1. Open the editor if it has not yet been opened.
2. Double-click on the symbol "Current mode" .

The WebComposer now changes to RUN mode and visualizes the project in the internal browser.

Visualization in an external browser

You are in "Editing mode"  (see [chapter 3.3.1 "Screen editor" on page 14](#)).

1. Open the editor if it has not yet been opened.
2. Double-click on the symbol "Preview"  off.
3. Select "Open" or "Open with" (see [chapter 3.3.1 "Screen editor" on page 14](#)).
 - "Open": The default browser is used.
 - "Open with": Any browser can be selected.

The WebComposer now changes to RUN mode and visualizes the project in an external browser.

4.7 Triggering of events

All available events are contained under the WebComposer properties of the respective widget that can be opened by double-clicking on the widget.

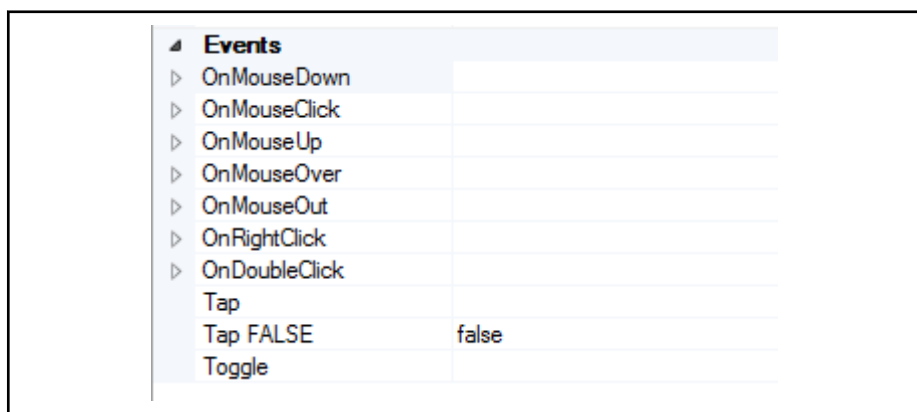


Fig. 4-21: Widget events

Events

Event	Description
OnMouseDown	Event is triggered when clicking on this element
OnMouseClicked	Event is triggered if a complete mouse-click was executed on the element; i.e. press mouse key and release
OnMouseUp	Event is triggered when releasing the mouse key on this element
OnMouseOver	Event is triggered when dragging the mouse pointer on this element
OnMouseOut	Event is triggered when the mouse pointer is removed from this element
OnRightClick	Event is triggered upon a right-click
OnDoubleClick	Event is triggered upon a double-click

Tab. 4-1: Events and their descriptions

4.7.1 Writing variables about events

1. Double-click on the symbol  of the event.

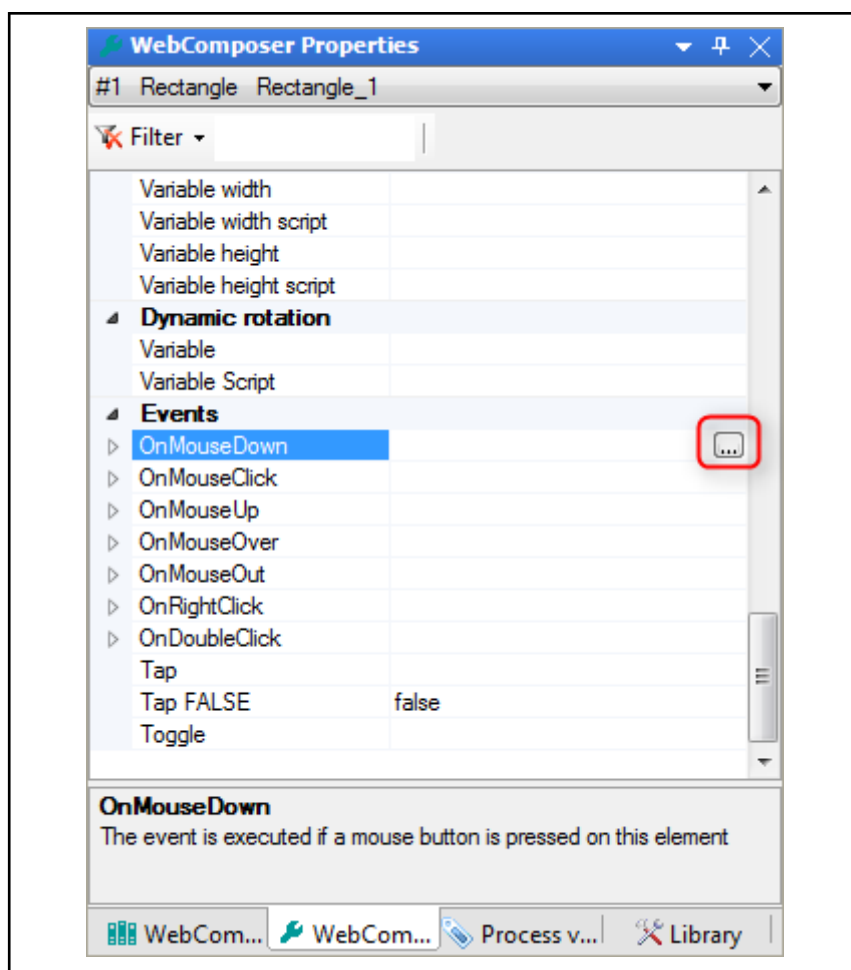


Fig. 4-22: Open the event configuration "OnMouseDown event"

The event configuration of the desired event opens.

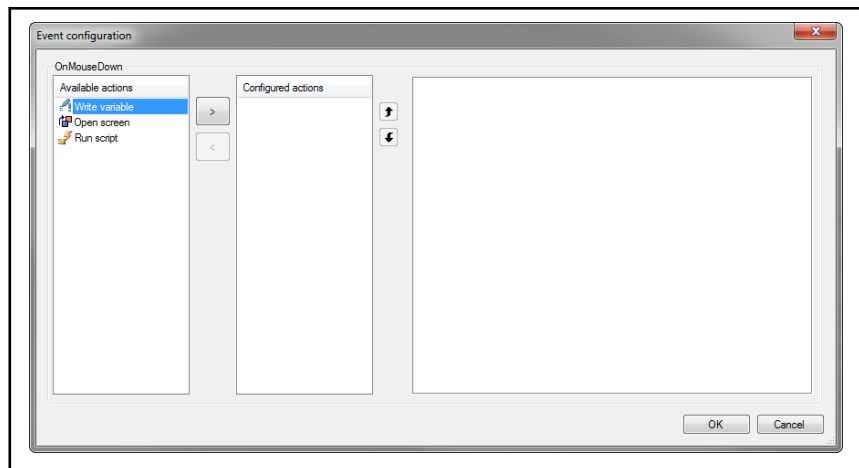


Fig. 4-23: Event configuration of a widget

2. Select "Write variable" in "Available actions".
3. Apply the "Write variable" function by clicking on  to the "Configured actions".

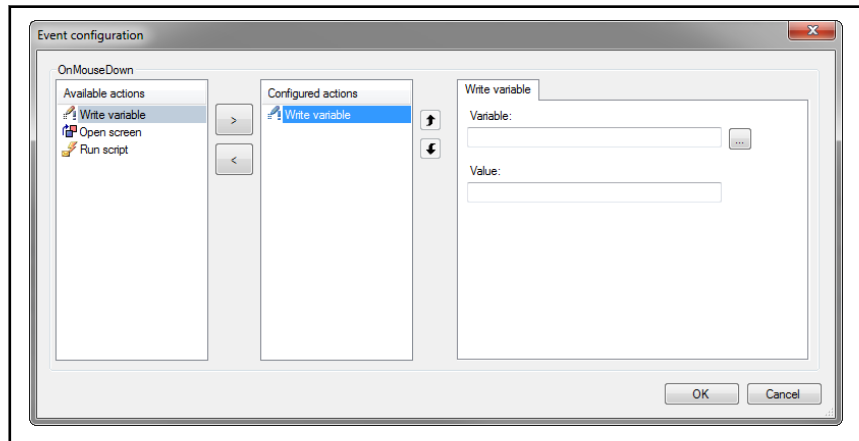


Fig. 4-24: Event configuration of a widget - Write variable

4. Double-click on the symbol . The "Process variable" window opens.

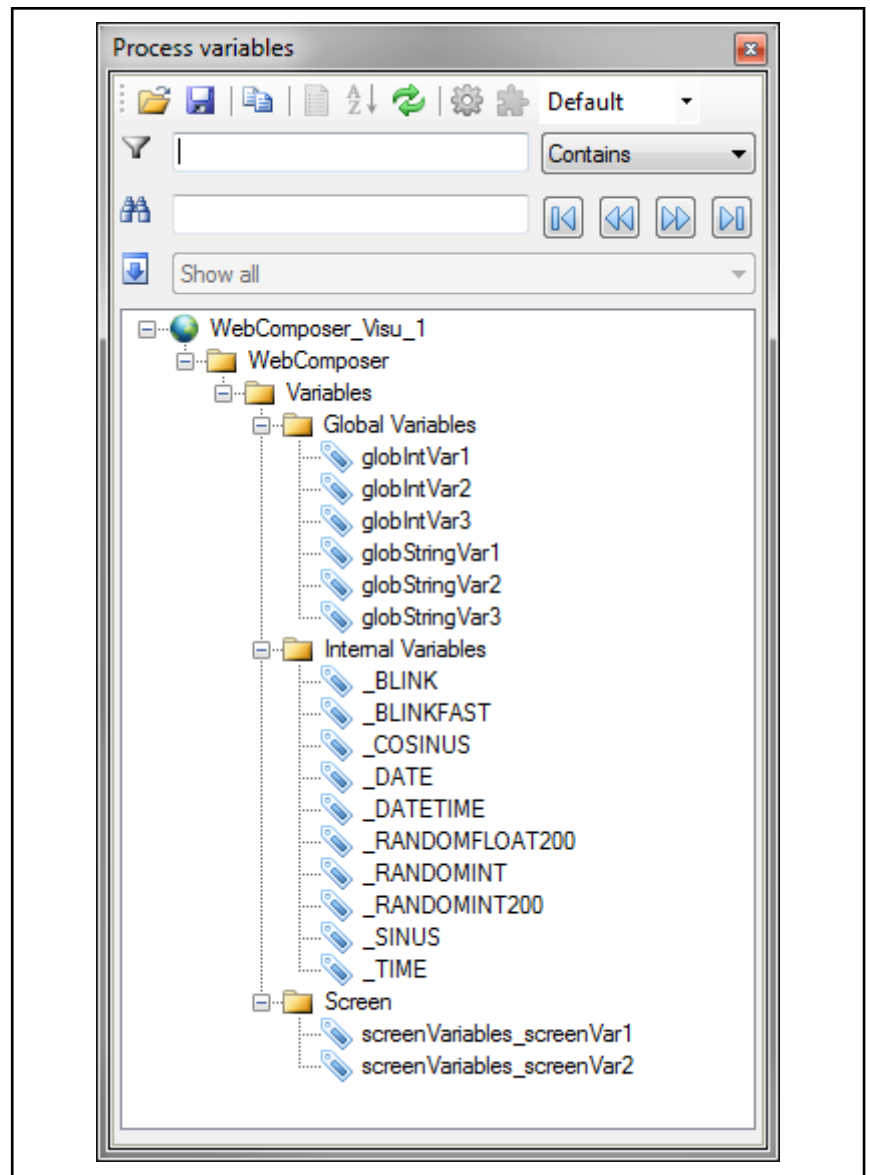


Fig. 4-25: "Process variables" window

5. Select the desired variable and confirm with "OK".
The "process variable" window is closed.
6. Enter a value in the "Value" input field in the "Event configurator" that is to be set when executing the event.
Confirm with "OK".

4.7.2 Switching screens via events

1. Double-click on the symbol  of the event.

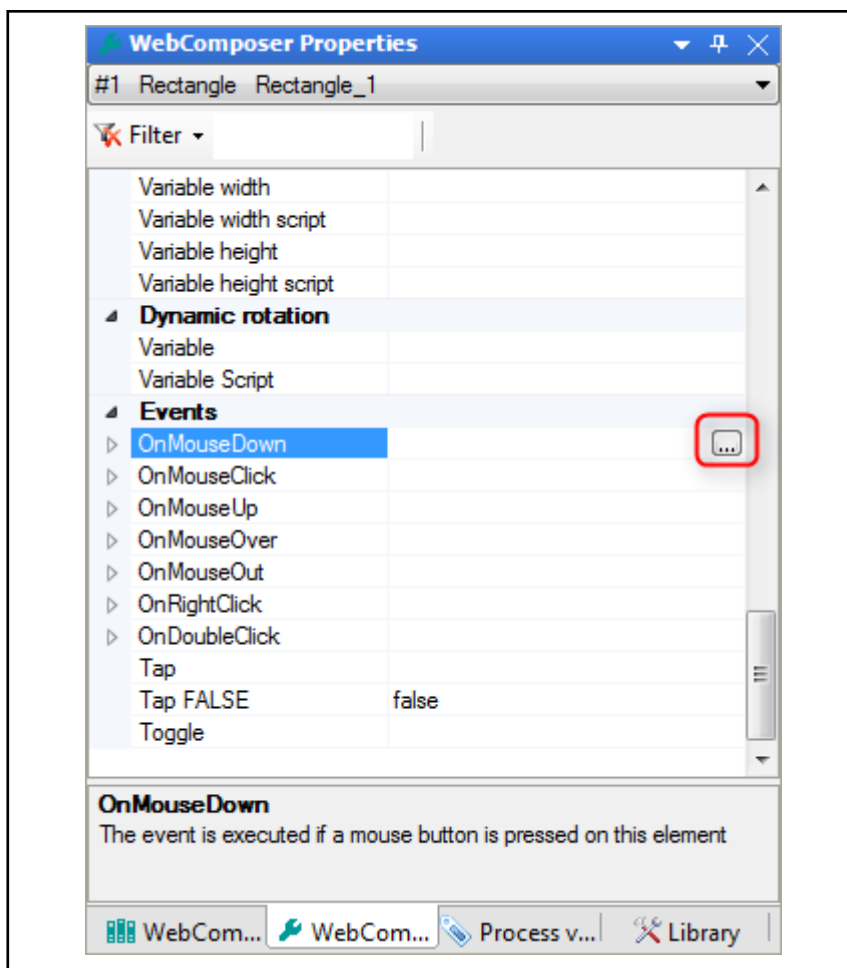


Fig. 4-26: Open the event configuration "OnMouseDown event"
The event configuration of the desired event opens.

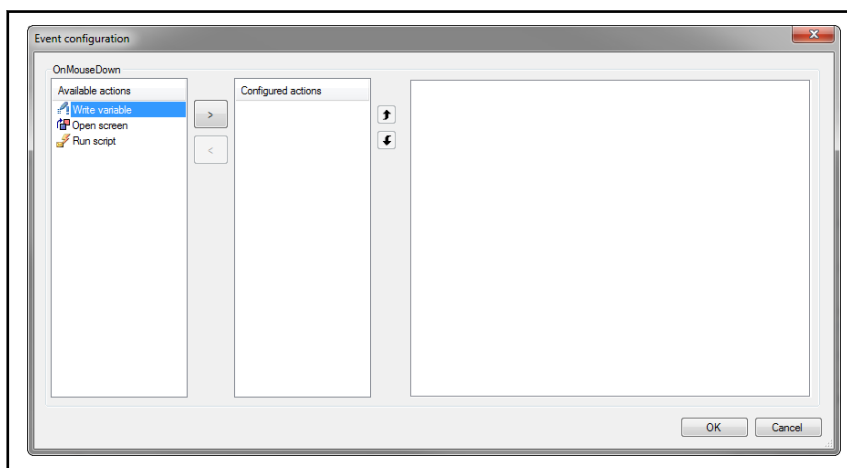


Fig. 4-27: Event configuration of a widget

2. Select "Open screen" in "Available actions".
3. Apply the "Open screen" function to the "Configured actions" by clicking on .
4. Select the screen and confirm with "OK".

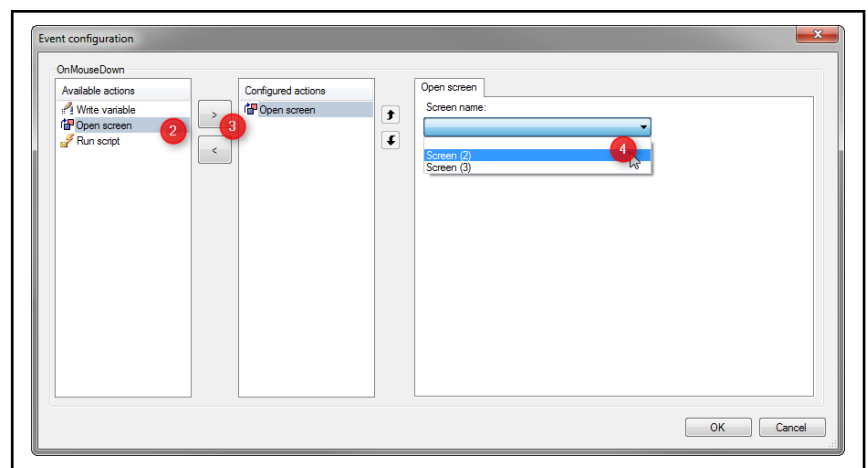


Fig. 4-28: Process to add screen to event configuration



These events only become active in "RUN mode" and can only be executed in "RUN mode".

4.7.3 Starting scripts via events

1. Double-click on the symbol  of the event.

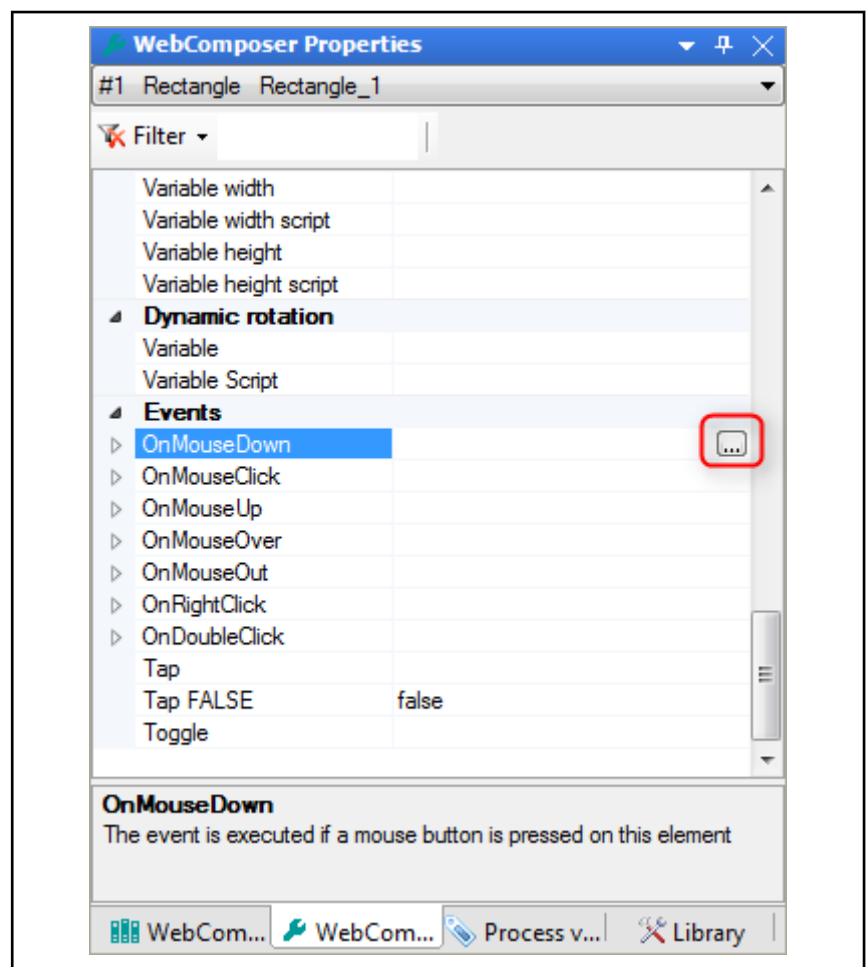


Fig. 4-29: Open the event configuration "OnMouseDown event"

The event configuration of the desired event opens.

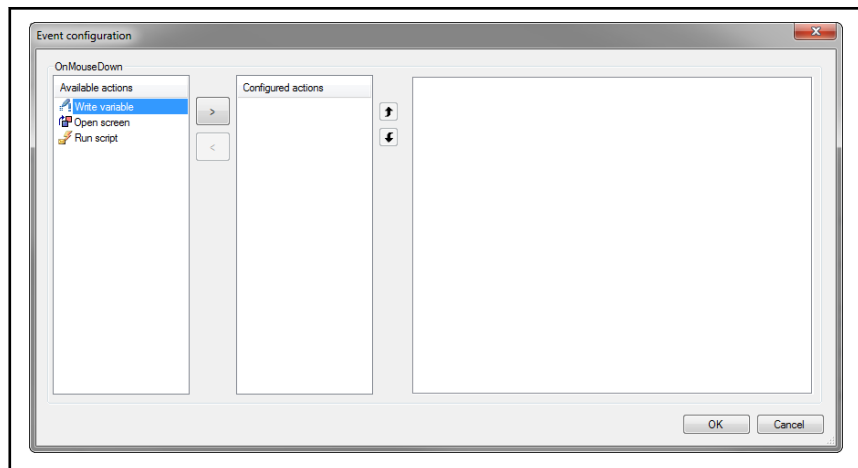


Fig. 4-30: Event configuration of a widget

2. Select "Start script" in "Available actions".
3. Apply the "Start script" function to the "Configured actions" by clicking on the  symbol.
4. Write a script in "Start script". The script is to be started when executing the event.

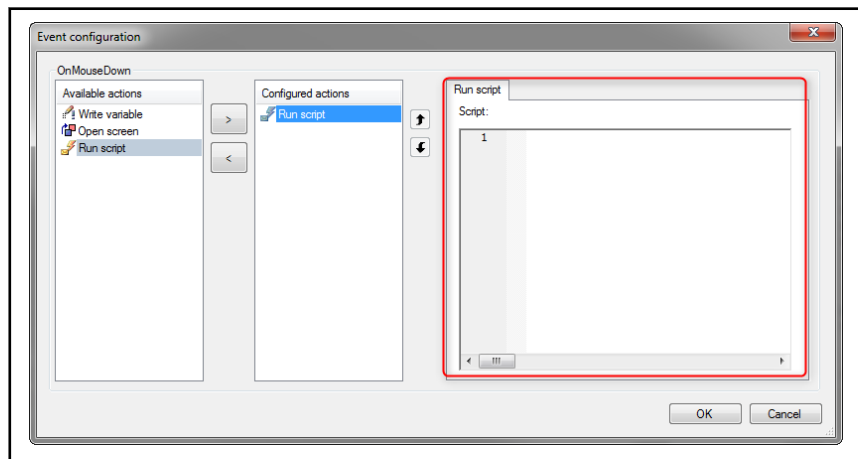


Fig. 4-31: "Start script" area

5. To confirm the selection, press "OK".

4.8 Creating custom screens

4.8.1 General

The user interface of the WebComposer in IndraWorks Engineering can be used to create and edit web-based custom screens. Multiple graphic objects (widgets) are provided. Drag them to the user screens and drop them. The created screens including their data and functions (variables and scripts) can be displayed either directly on the control via FTP or using the WebAssistant (web browser). They can be stored as screen package on a data carrier using the upload function in the WebAssistant.



The custom screens in the WebAssistant are exclusively to be used for servicing and commissioning purposes and not for machine operations! Use cases such as the setup mode or traversing an axis in jog mode is considered as unintended use case.

4.8.2 Configuration in IndraWorks Engineering

Licensing

Creating and transferring custom screens require a license. There are two license variants.

- Single Engineering license for WebComposer applications
- Multiple Engineering license (five times) for WebComposer applications

Creating custom screens



The following functionality is not supported for custom screens:

- Screen scripts in referenced screens
- Using main functions

Notation to set or compare with Boolean variables: "TRUE" or "true".

Selecting and transferring

To open the dialog for custom screens, go to the context menu of the WebComposer device.

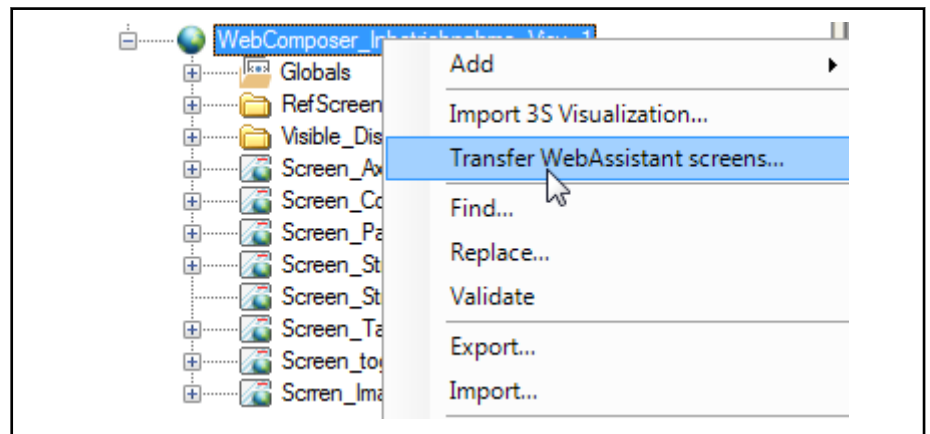


Fig. 4-32: Opening the dialog for custom screens

Use cases

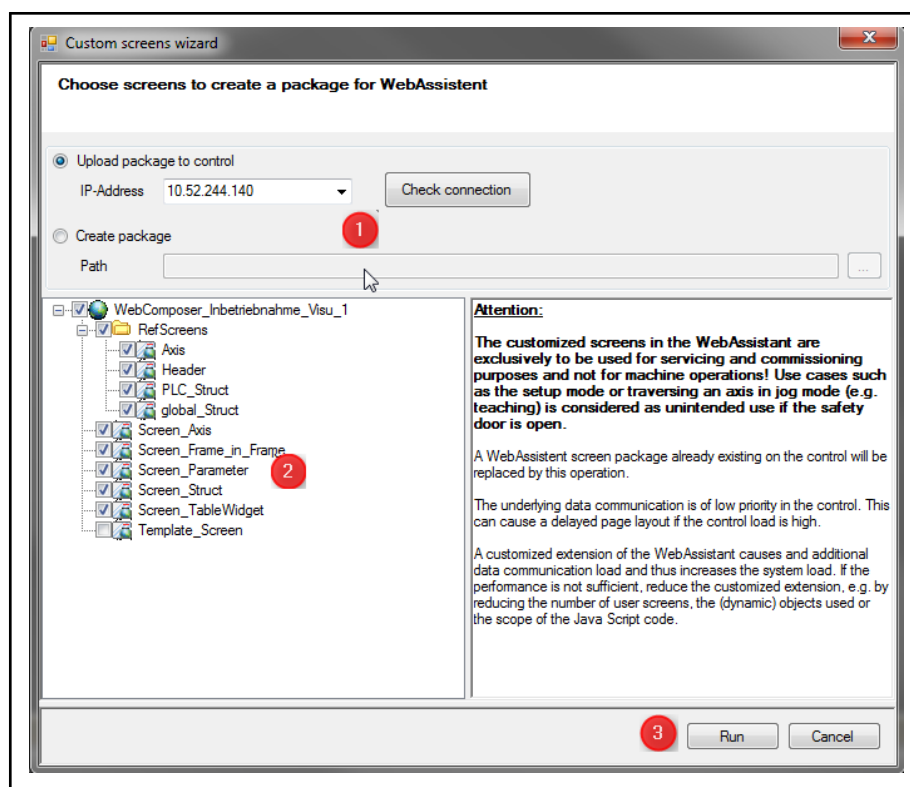


Fig. 4-33: Wizard for custom screens

Storage location of the screen package ①

- Upload package to control: Custom screens are directly transferred to the control via FTP



After entering the IP address, press "Check connection" to test the connection to the control. The test result is shown as symbol.

- Connection could be set up 
- Connection could not be set up 

Selecting custom screens ②

- Create package: The screen package is stored on a data carrier and can be uploaded and displayed using the "Upload" function in the WebAssistant.

All available WebComposer screens are displayed in the selection window and can be selected as custom screens. Selecting/deselecting the "Root" node selects/deselects all subnodes.



An already existing screen package on the control is deleted due to the transfer and replaced with the new screen package.

Executing des wizards ③

Press the "Run" button to create or transfer the screen package.

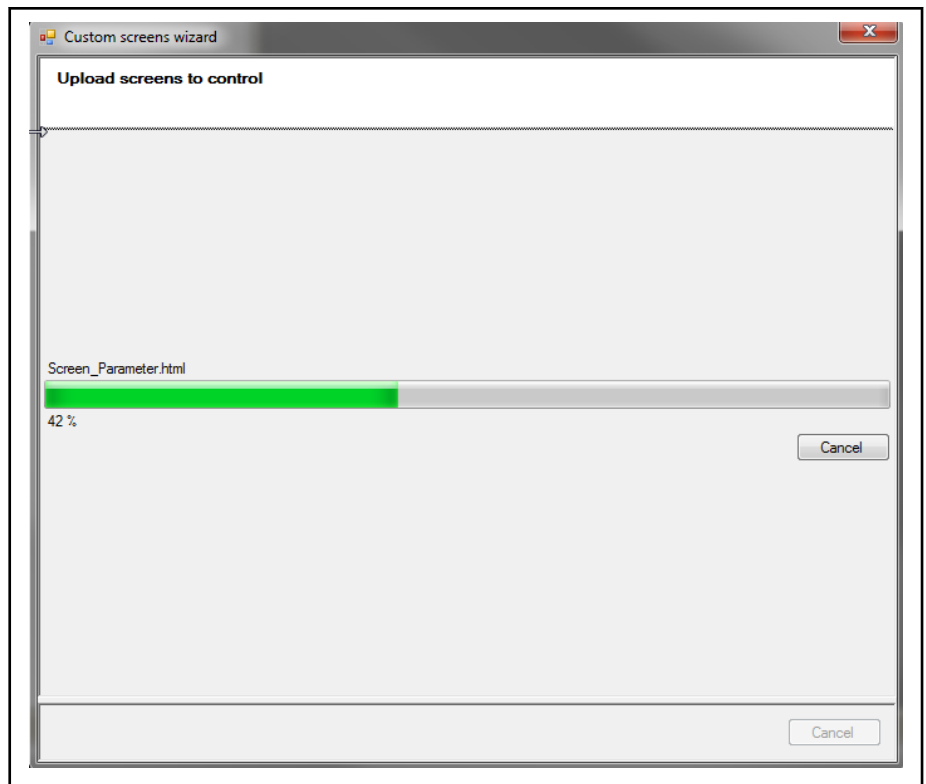


Fig. 4-34: Transferring custom screens



A customized extension of the WebAssistant causes an additional data communication load and thus increases the system load. If the performance is not sufficient, reduce the customized extension, e.g. by reducing the number of user screens, the dynamic objects used or the scope of the Java Script code.

A summary of the process is displayed in the last step of the wizard. The upper section of the summary shows whether the process was successful or whether errors were issued. The lower section of the summary visualizes the individual file operations.

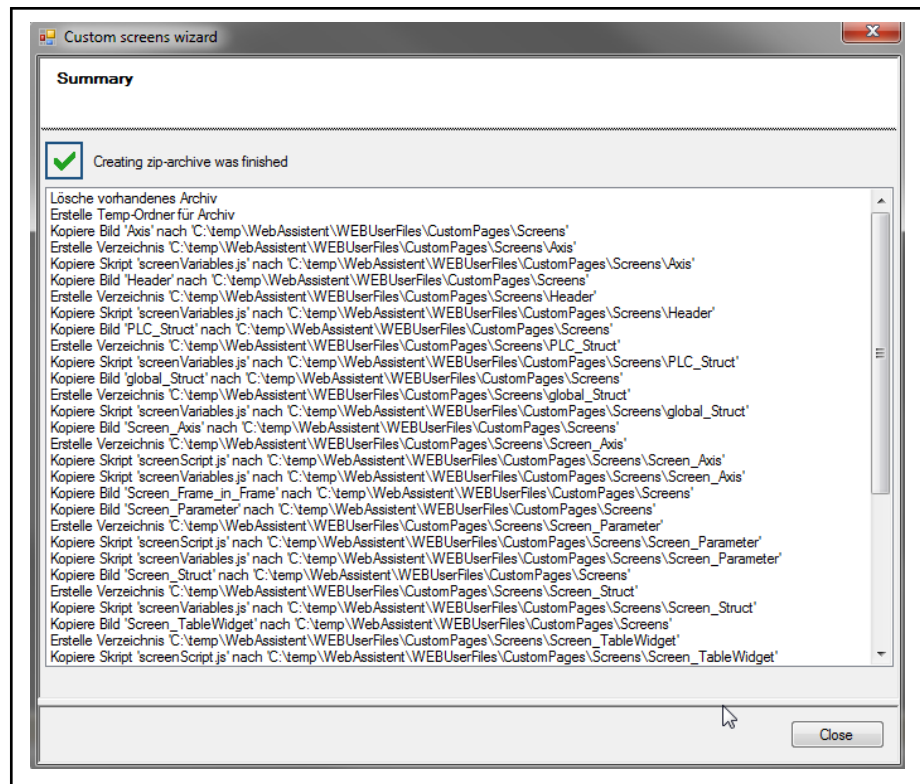


Fig. 4-35: Wizard summary

4.8.3 WebAssistant

Licensing the custom screens

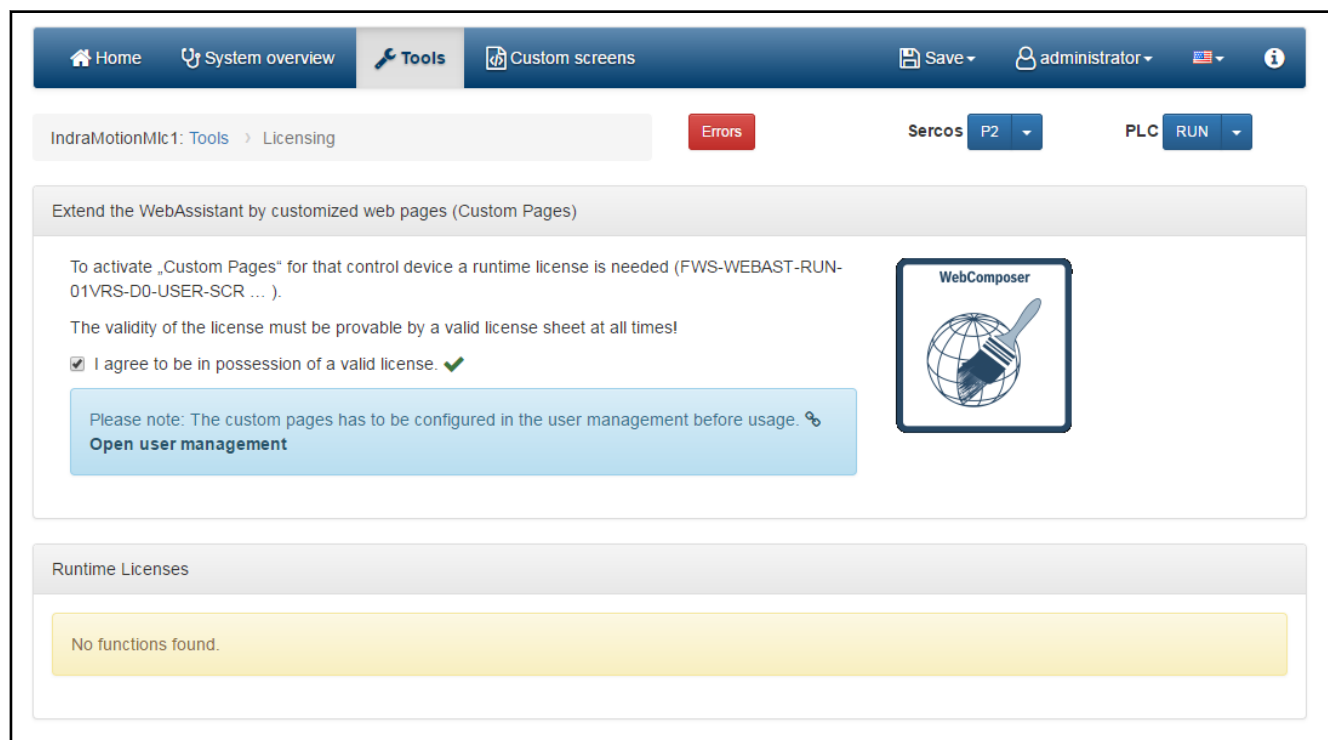


Fig. 4-36: Licensing site

Enhancement by custom screens

To display custom screens created with the "WebComposer" in the WebAssistant, confirm in this dialog that you have a valid license. The sites are then visible.

License runtimes

This section shows all firmware functions subject to license used in the current application.

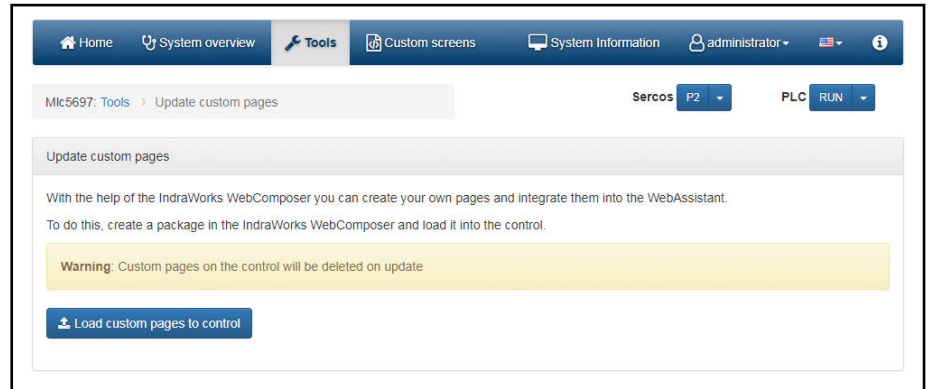
Refreshing custom screens

Fig. 4-37: Refreshing custom screens

Page can be created in the WebComposer which are then visible in the WebAssistant. To export these screens in the WebComposer, use the "Load custom screens...". The screens are then imported using this function in the WebAssistant.

To display the screens, the following conditions have to be met:

1. The license FWS-WEBAST-RUN-01VRS-D0-USER-SCR has to available
2. Confirm that you own the licensed mentioned above under Tools/ Licensing
3. Each single screen that should become visible for each user has to be enabled in the user management
4. After changing the user management, the WebAssistant has to be loaded again (e.g. with F5)

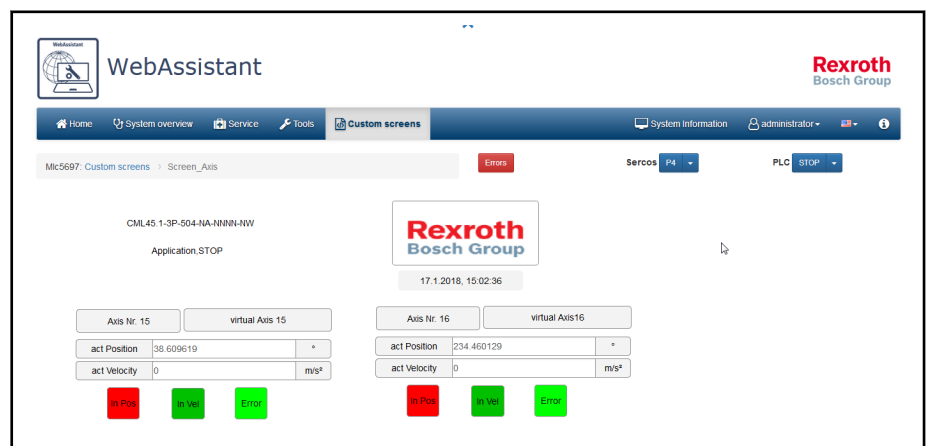
Displaying custom screens

Fig. 4-38: Displaying custom screens

4.9 Creating global, localized texts

Use the text editor [chapter 3.2 "Project elements and their functions" on page 9](#) in the WebComposer to edit texts available in all WebComposer components. This text can be entered in several languages and is subsequently linked in the "text" attribute of a widget via an identifier.

Localization/multilingual

Use the text editor to enter texts for different languages. Add the language to the project via the IndraWorks menu "Manage project-language-project language". All selected project languages are provided in the text editor and texts can be entered. The global text displayed on the screens at runtime is based on the currently set IndraWorks project language ("Select project-language-project language").

Text lists

Text lists are a special feature. The entered text can be displayed on the basis of a value of a defined process date (PLC variable, control and drive parameters, global or local script variables). In contrast to regular texts, a value can be saved for each text in the text list. At runtime, it is checked which value in the text list corresponds to the value of the variables linked in the widget and the corresponding text is displayed. If no variable value in the text list matches, the first text in the list is displayed.

The following rules apply to display texts in a widget:

- If the "Variable" attribute is empty, the content of the "Text" attribute is displayed
- If the "Variable" attribute is assigned (e.g. with a PLC variable), the content/variable value is displayed
- If the "Text" attribute contains a text with placeholder (e.g. %s) and if the "Variable" attribute is assigned (e.g. with a PLC variable), the content/variable value is displayed instead of a placeholder

These rules apply for texts directly entered in the "Text" attribute as well as for global texts defined in the text editor, if required also in multiple languages.

Placeholders that can be used:

- %b, results in an integer as binary number
- %c, results in an integer such as the character with the ASCII value
- %d or %i, results in an integer as signed decimal number
- %e, results in float value by scientific notation
- %u, results in an integer decimal number without sign
- %f, results in a float value; see accuracy example
- %o, results in an integer as octal number
- %s, results in a string
- %x, results in an integer as hexadecimal number (lower case)
- %X, results in an integer as hexadecimal number (upper case)
- %j, results in JavaScript object or array as JSON-decoded string

Placeholder format:

"%[width][.precision]specifier"

Example of text with placeholders

Content text attribute	Variable value	Displayed text
The color: %s	"blue"	The color: blue

First number: %d	12345	First number: 12345
Second number: %04d	25	Second number: 0025
Float number: %3.2f	3.14159	Float number: 3.14
Hexadecimal: %x	255	Hexadecimal: ff
Octal: %o	255	Octal: 377

Tab. 4-2:

4.10 Formatting global, localized texts

Texts directly entered in the "Text" attribute as well as for global texts can be formatted. HTML characters are used for formatting. The following formatters can be used:

- ""Bold
- "
"Line break
- ""Sorted list
- ""List element
- ""Important text

For more HTML formatters, browse the WWW/internet.

Example of text formatting

Content text attribute	Displayed text
Regular text and bold text.	Regular text and bold text .
First line Second line	First line Second line
First line - Second line - Third line	1. First line 2. Second line 3. Third line

Tab. 4-3:

4.11 Using global, localized texts in widgets

Execute the following steps if global texts are to be used in widgets:

1. Double-click on the "global data variable TextDef" project node to open the text editor
2. Add a new text or a new text list and specify a unique "identifier"
3. Enter the desired text, if required also for several languages.
4. Save the texts
5. Open the screen with the widget on which the text is to be displayed
6. Select the widget and select the "Text" attribute in the "WebComposer properties" window
7. Call the process variable browser via the button in the value field of the "Text" attribute and open the "Global texts" area.
8. Select the identifier. The identifier is applied to the value field

Use cases



The identifier is displayed in edit mode. In Run mode, the text linked to the identifier is displayed. The global text displayed on the screens at runtime (Run mode) is based on the currently set IndraWorks project language ("Select project-language-project language").

Content attribute "Text"	Content "global text"	Content attribute "Variable"	Variable value	Result 2)	Explanation
"This is my text"	-	-	-	"This is my text"	The "Text" attribute contains text and the "Variable" attribute is empty -> Text in "Text" attribute is displayed.
"This is my text"	-	ns=2;s=Application.HMI.strNewText	"Another text"	"Another text"	The "Text" attribute contains text and the "Variable" attribute contains a variable -> The variable value is displayed.
"This is my %s text"	-	ns=2;s=Application.HMI.arHMIAxes[1].strName	"new"	"This is my new text"	The "Text" attribute contains text with a placeholder (e.g. %s) and the "Variable" attribute contains a variable -> The variable value replaces the placeholder and the text including the variable value is displayed.
globalText.MyText	"This is my global text"	-	-	"This is my global text"	The "Text" attribute is linked to a global text and the "Variable" attribute is empty -> Text in the linked global text is displayed (in the selected IW project language).
globalText.MyTextExt	"This is my global text"	ns=2;s=Application.HMI.arHMIAxes[1].strName	"This text is more important!"	"This text is more important!"	The "Text" attribute is linked to a global text and the "Variable" attribute contains a variable -> The variable value is displayed.

Content attribute "Text"	Content "global text"	Content attribute "Variable"	Variable value	Result 2)	Explanation
globalText.MyTextExt	"This is my %s global text"	ns=2;s=Application.HMI.arHMIAxes[1].strName	"new"	"This is my new global text"	The "Text" attribute is linked to a global text with a placeholder (e.g. %s) and the "Variable" attribute contains a variable -> The variable value replaces the placeholder and the global text including the variable value is displayed (in the selected IW project language).

Use cases

Content attribute "Text"	Content "global text"	Content attribute "Variable"	Variable value	Result 2)	Explanation
globalText.MyTextList 1)	See below 2)	ns=2;s=Application.HMI.intValue	0	"My first text"	The "Text" attribute is linked to a global text list and the "Variable" attribute contains a variable -> If the variable value of the variables corresponds to a variable value of the text list, the text is displayed. If the variable value of the variable does not correspond to a variable value of the text list, the first text of the text list is displayed.
globalText.MyTextList 1)	See below 2)	ns=2;s=Application.HMI.intValue	2	"My third (2) text"	The "Text" attribute is linked to a global text list and the "Variable" attribute contains a variable -> If the variable value of the variables corresponds to a variable value of the text list, the text is displayed. Placeholders are replaced by the variable value. If the variable value of the variable does not correspond to a variable value of the text list, the first text of the text list is displayed.

1)
2)

"German" is set as project language in the examples
The text list required for the previous examples in the text editor has the following structure:

Identifier	Variable value	German	English
MyTextList	0	My first text	My first text
	1	My second text	My second text
	2	My third (%s) text	My third (%s) text

4.12 Translating global texts externally

If several languages are to be supported for texts in WebComposer screens, it can be selected directly in the text editor.

There is also the option to export all WebComposer texts in an Excel file and to e.g. send it to an external translation agency. Open "Export project language translation file..." in the IndraWorks menu. In this dialog, select the desired target language for the texts.



A translation file can only contain one target language; i.e. if the text is to be translated into two languages, two translation files have to be created.



When saving the Excel file, a prompt is displayed asking the user if the text is to be saved in unicode text format. Confirm this prompt with "Yes" as the Excel file cannot be imported anymore otherwise.

4.13 Debugging JavaScript code

Call the browser debugger using "Show DevTools" in the screen editor menu. More information about the debugger can be found under "Chrome DevTools".

4.14 3S visualization import

Use this function to import visualizations below the logic node into the WebComposer. Prior to the import, ensure that you are not logged in to the control and that the PLC project is compiled successfully.



A complete import is not possible as not all elements and their attributes can be applied. For elements and attributes that are not imported, a node in the device tree is created in the WebComposer device. Double-click on the node, the entries in the message window are listed.

The import wizard creates a POU "WebComposer" in a PLC task to recover the process communication. This POU is only used for communication and is not required for a standard WebComposer application (only the used PLC variables have to be contained in the symbol file).

After the import, translation error can occur in the WebComposer POU of the PLC. In the following a list of the known errors and bugfixings.

- Error C0032

Example:

```
Error;;C0032: Cannot convert type 'INT' to type 'TIME';WebComposer
[IndraMotionMlc1: Logic: Application] Project: ISI_Pathfinder_GAT_MLCx5_13V06;Line 772, Column 1 (Impl);WebComposer [IndraMotionMlc1: Logic: Application] Project: ISI_Pathfinder_GAT_MLCx5_13V06 arKinCtrl_gb[VisuKinematicsNo].Admin.ConveyorBelt[1].ErrorReactionDesyncTime:=x728_7_32_text_variableW;
```

Reason:

WebComposer always declares INT type for write variables. To clear this error, declare the variable x728_7_32_text_variableW (source is

INT) to the destination type arKinCtrl_gb[VisuKinematicsNo].Admin.ConveyorBelt[1].ErrorReactionDesyncTime (destination is TIME).

Recovery:

Declare x728_7_32_text_variableW as TIME by changing

```
x728_7_32_text_variableW:INT;
to
x728_7_32_text_variableW:TIME;
```

- Error C0037

Example:

Error;;C0037: 'stCyclesTime' is no input of 'TE_AxisInterface-Type01Prog';WebComposer [IndraMotionMlc1: Logic: Application] Project: Project53;Line 1466, Column 1 (Impl);WebComposer [IndraMotionMlc1: Logic: Application] Project: Project53

Reason:

3S Visualization element is using a variable in Execute ST Code or Write Variable which is not a type of VAR_INPUT.

Recovery:

Declare 'stCyclesTime' within VAR_INPUT:

```
VAR_INPUT
stCyclesTime : TE_PLC_CYCLES_TIME;
END_VAR
```

- "Paths" are incomplete

Example:

```
PROGRAM PlcProg VAR Index:int; arrayStrings:ARRAY[0..10] OF
STRING; END_VAR
```

Reason:

PLC variables including the local variables can be specified in 3S as "incomplete". The following example is valid in 3S, but "index" cannot be evaluated in the WebComposer: PlcProg.arrayStrings[index]

Recovery:

The statement has to be rewritten to

```
PlcProg.arrayStrings[PlcProg.index]
```

- Case sensitivity of variables is ignored

Reason:

Case sensitivity of variables is checked due to an internal error.

Recovery:

Adjust the variables so that they correspond exactly to the declaration in the PLC.



It is recommended to change the variable connection of an imported visualization to the standard communication and to delete the generated POU again.

Elements and attributes that cannot be imported

No all elements and their attributes can be applied. All elements that have not been applied are identified with a red "X" in the WebComposer screens. Additionally, an "Import Messages" file is created in the project tree. Double-

click to display the messages reported during the import in the message window. Double-click on the message to open the relevant screen.

Some examples of elements and attributes that cannot be imported:

- BarDisplay element
- ComboBoxInteger element
- Table element
- Slider element
- OnDialogClosed attribute
- OnShowDialog attribute
- OpenFileDialog attribute
- Absolute/relative motion X attribute
- Absolute/relative motion Y attribute
- Frame variables with a nesting depth > 1
- Enum

If accesses to local variables are used, they cannot be imported either. A compiler error is reported in the WebComposer POU. These local variables have to be declared as VAR INPUTS. Consequently, the POU can be compiled again and the values are displayed on the WebComposer screen.

4.15 Creating a new theme

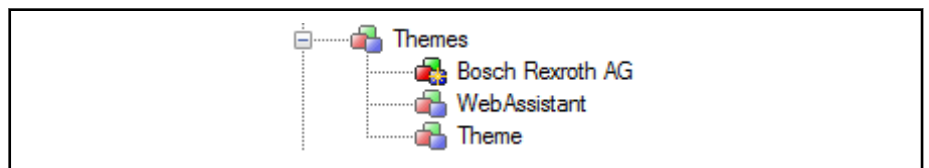


Fig. 4-39: Project element "Themes" with active "Bosch Rexroth AG" template "Themes" is a subelement of "Global data".



New themes can only be created below "Themes".

The WebComposer provides three options to create new themes:

- Via the context menu
- Via the "Copy" function
- Via the "Import" function

Creating objects via the context menu

1. Right-click on "Themes" in the Project Explorer.
2. Select "Theme" via the subitem "Add".

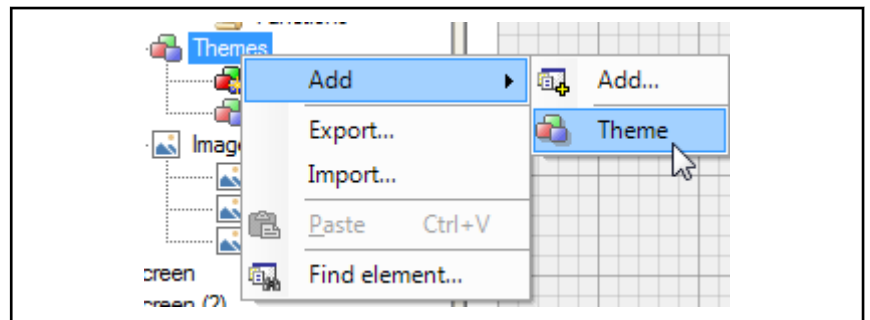


Fig. 4-40: Add a new theme in the Project Explorer

Subsequently, a new theme is added to the project tree.

Creating objects via the "Copy" function

3. Double-click on the recently added theme.
The "Theme editor" of the themes opens.
4. Adjust the properties.
1. Right-click on the theme in the Project Explorer.
2. Select "Copy".

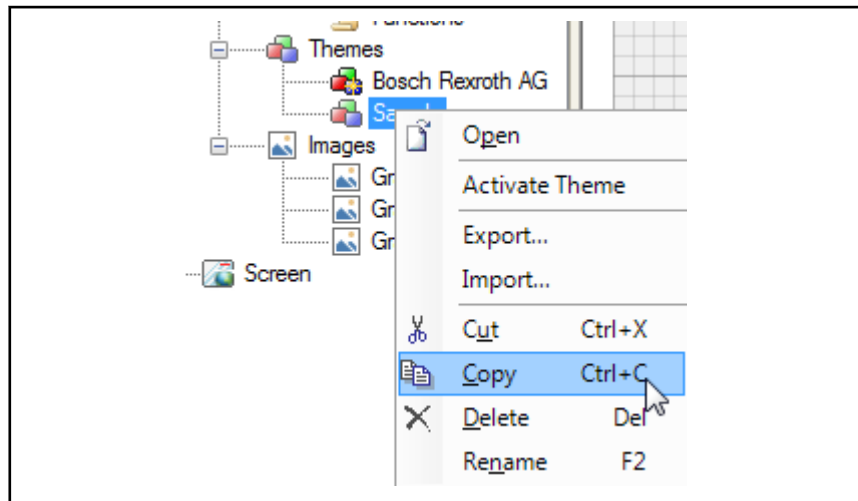


Fig. 4-41: Copy theme to clipboard

The WebComposer now copied the theme to the clipboard.

3. Right-click on "Themes" in the Project Explorer.
4. Select "Paste".
The previously copied theme is added to the list of themes. The name is automatically adjusted.
5. Double-click on the recently added theme.
The "Theme editor" of the themes opens.
6. Adjust the properties.

Creating objects via the "Import" function

- Prerequisite: A theme archive is already available.*
1. Right-click on "Themes" in the Project Explorer.
 2. Select "Import".

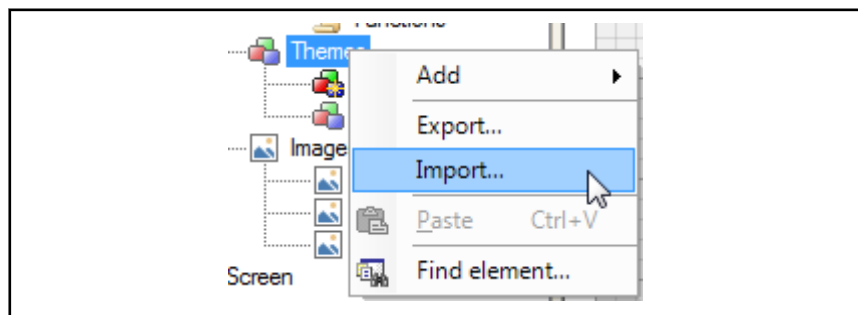


Fig. 4-42: Importing a theme archive in "Themes"

3. Select the theme archive to be imported and confirm with "OK".

The file to be imported is added to the menu item "Themes".

4. Double-click on the recently added theme.
The "Theme editor" of the themes opens.
5. Adjust the properties.

Customizing the theme properties

1. Double-click on the recently added theme.
The Theme editor opens.

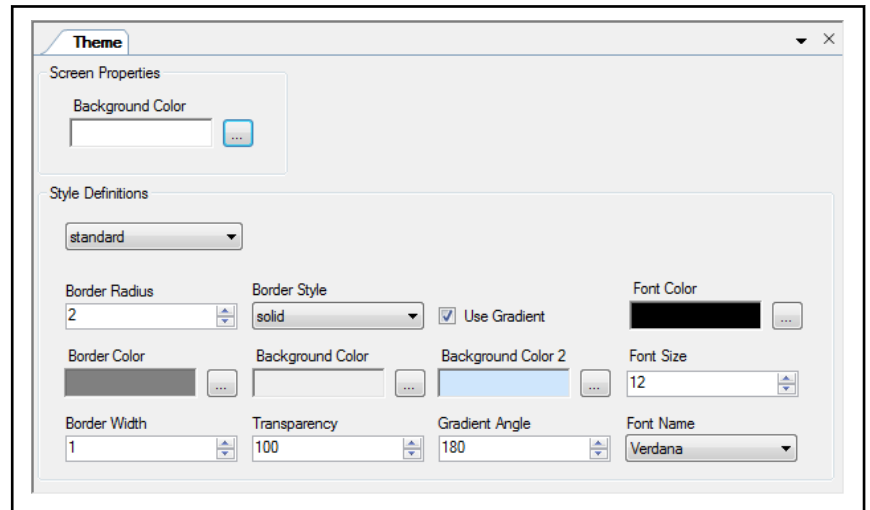


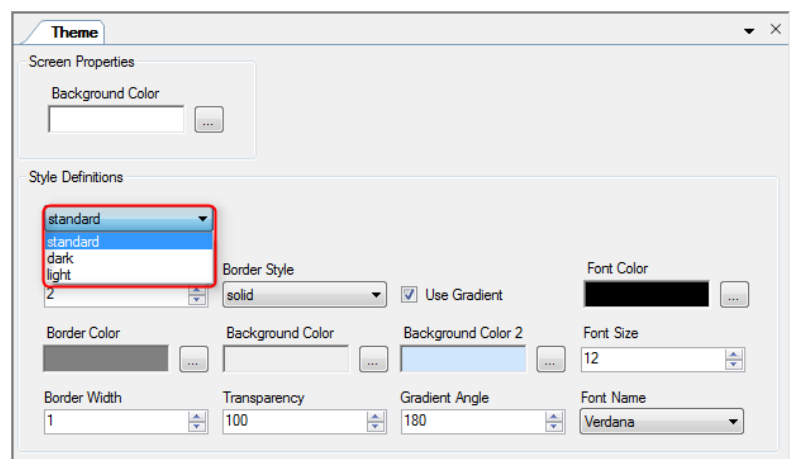
Fig. 4-43: Theme editor

2. Within the theme editor, the visual properties of this theme can be defined and edited. The visual properties are subdivided into "Screen properties" and "Style definitions".



Each theme has three style types (standard, dark, light).

These style types can also be customized via the "Style definitions".



Enabling the newly created theme

1. Right-click on the new theme in the Project Explorer.
2. Select "Activate theme".

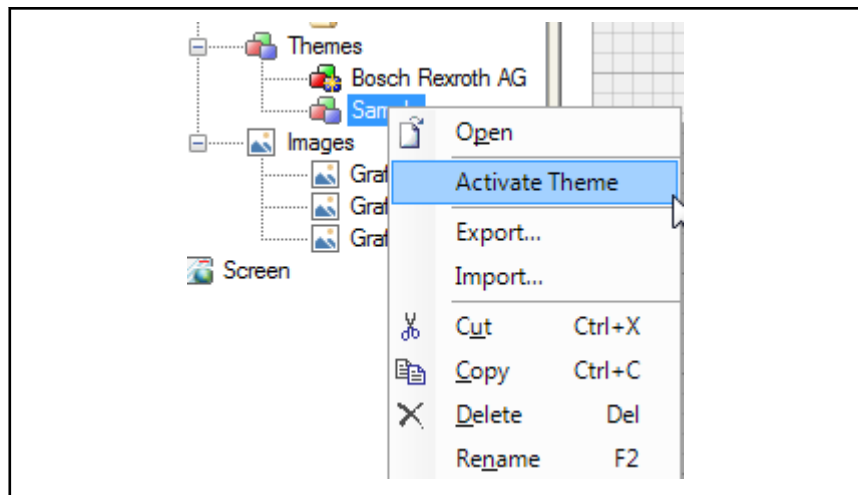


Fig. 4-44: Enabling the theme in the Project Explorer

The active theme is visualized in the Project Explorer by .



- The active theme is applied to the entire visualization.
- Only one theme can be set to active at a time.



- The newly created themes can be renamed and deleted.
- The "Bosch Rexroth AG" template can neither be deleted nor changed.



The number of themes is not limited.

4.16 Taking a widget from a theme

Individual widgets can be taken from the active theme to be able to configure them independently from the theme.

1. Open the "WebComposer properties" of the widget by double-clicking on the widget.
2. Set the "Selected theme" in the "Style" area to "none".

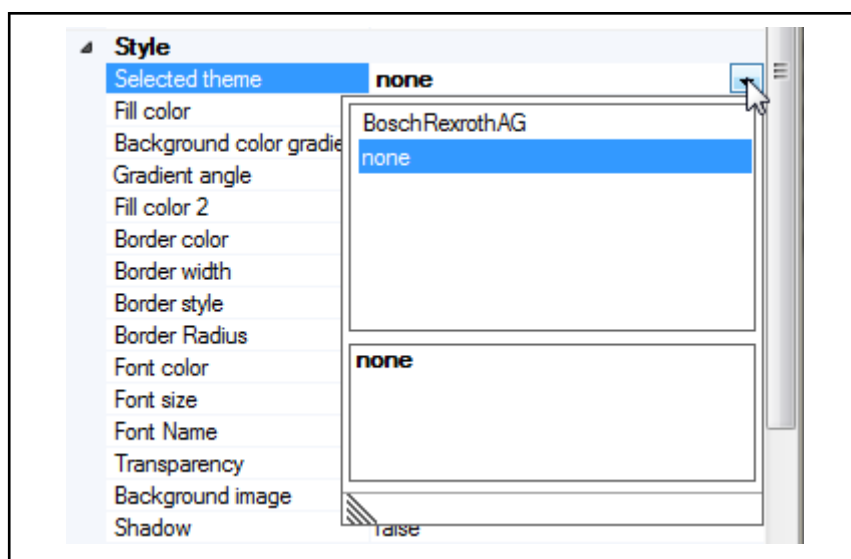


Fig. 4-45: Selected theme: none

New setting options open in the WebComposer properties under "Style" which can be used to set specific settings for the selected widget.

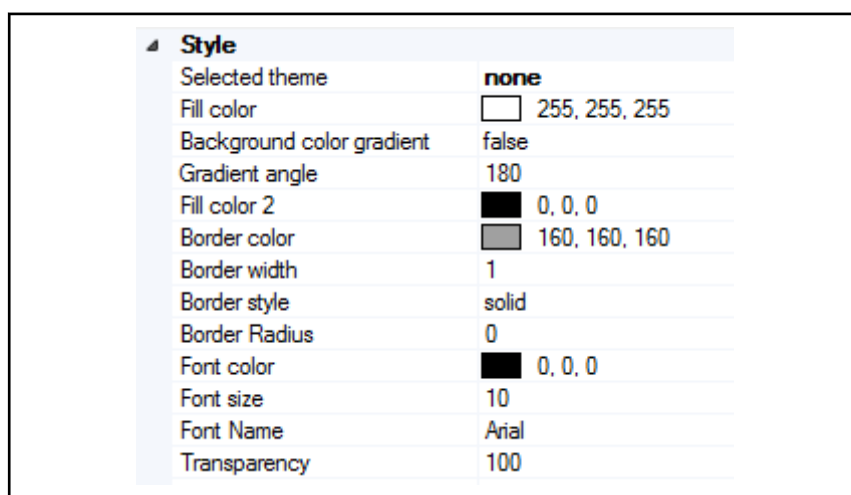


Fig. 4-46: Style properties of a widget can be taken from the active theme

3. Adjust the widget properties.



The adjustment does not have any influence on other widgets with an active theme.

4.17 Checking changes immediately

Often, only small changes such as changes in the widget position or size are made and the look-and-feel is to be tested immediately during runtime in run mode. When only using the editor and its run mode in IndraWorks, check the change upon each change to run mode and subsequently go editor mode again.

If the changes have to become visible immediately, the following procedure is recommended:

1. Open the relevant image via the preview button in the menu of the screen editor in the web browser.
2. Select the changes in the screen or at the widget.

3. Save the changes via the "Save all" button in the IndraWorks menu.
The screen is immediately refreshed in the web browser.



The update in the external web browser only works if the screen is in the selected/visible tab in the web browser.

No update is executed if the web browser is minimized.

5 FAQ

5.1 The control does not return any data - What can I do?

If data cannot be read by the control, it can have various reasons.

- An incorrect IP address is entered. Please check the IP address in the property dialog of the WebComposer device.
- An incorrect IP address is entered in an individual screen (valid for the entire screen). Please check the IP address in the attribute window of the screen in the "Communication settings" area. The screen attributes are displayed when clicking on the screen background.
- During the transition to Run Mode, communication problems are reported and the entire communication for the screen is stopped. Please check the screen. The affected widgets with communication problems are represented graphically. Please also check the "WebComposer" category for entries in the message window. Double-click on the entry to select the widget and the "Attribute" window automatically.

If none of the mentioned measures solves the problem, call the browser debugger using "Show DevTools".

5.2 Why are SVG graphics not displayed on customized pages?

The WebComposer also supports SVG graphics. However, if SVG graphics are used on customized pages of the WebAssistant, the SVG graphics are only supported from MLC firmware 14V22. SVG graphics cannot be displayed in the WebAssistant with older MLC firmware versions.

5.3 Why is a transparent screen not displayed as transparent?

If a transparent image (e.g. *.gif) is not displayed as transparent in the WebComposer, the image is most probably assigned to design template, setting the image background to a defined color.

Please ensure that the attribute "Selected theme" is assigned the value **none** in the "Look and feel" area.

5.4 Why are WebComposer images empty upon opening (white screen)?

Possible reasons:

- WebConnector service is not running
Please check the output in the IndraWorks message window. The entry "WebConnector is running successfully" should be available. If the WebConnector is not running, check the status of the "boschrexroth.webconnector" service under **System Control ► System and Security ► Administration ► Services**. If required, start the service manually.
- WebConnector is not installed or is installed incorrectly
If the WebConnector service is not available in the system or if the manual start does not fix the problem, check the WebConnector installation in the system under "Apps & Features". Exactly one "WebConnector" entry should be contained there. If two or more WebConnector installations exist, uninstall all installations and subsequently execute a repair installation in IndraWorks. If there is not WebConnector installation, execute the repair installation in IndraWorks.

- Java Runtime Environment (JRE) is not installed or is not up-to-date
WebComposer requires at least JRE version 1.8.0.
To verify the installed version, enter the Windows prompt (cmd.exe) and enter the command "java -showversion". If required, install JRE and update (download: <https://java.com/de/download/>).

5.5 Why are the referenced screens not working?

When configuring screen references, note that only constants or global variables can be used as index within a structure. Screen variables cannot be used as index.

5.6 Why is a valid variable identified as "BadNodeID"?

Reason: incorrect spelling of key words, as they are case sensitive.

Example: "globalVariables", in this case, "GlobalVariables" results in an error.

5.7 How can I can access the value of a PLC variable in the JavaScript code of a screen?

By using the widget attribute "Variable script" and a value assignment, the value of a linked PLC variable can be read and processed. Also refer to the description text and the application examples in the help window of the mentioned attribute.



Writing a value of a PLC variable in a script is currently not supported.

6 Attachment: Widget attribute

6.1 Bit list

6.1.1 General

Type	Type of the element
Id	Id of the element
Alias	Unique widget identifier

6.1.2 Position

X position	Distance from the left edge in pixels
Y position	Distance from the top edge in pixels
Width	Width of the element in pixel, including border
Height	Height of the element in pixel, including border
Rotation	Rotation of the element about its center in degrees

6.1.3 Style

Fill color	Background fill color
Border color	The border color, it can only be selected if no theme is selected.
Border width	Border width in pixel
Transparency	A value between 0 and 100, 0: the object appears completely transparent, 100: the object completely covers the background
Shadow	Show a drop shadow

6.1.4 Text properties

Variable	The linked variable, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.BLINK globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive paramters is normaly '0'
Font color	The color of the font, it can only be selected if no theme is selected.
Font size	Font size can only be selected if no theme is selected.
Font Name	Font Name can only be selected if no theme is selected.
Custom Filter	Name of filter function to format the displayed value.
Variable Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
Tooltip	Text that is displayed when the mouse pointer is over the item. Be aware that in case of a touch device the tooltip can't be displayed.
Update interval	Length of the update rate The following update rates are available: 'none': Value is only read once when screen is opened 'slowest': 10000ms 'slow': 5000ms 'default': 1000ms 'fast': 500ms 'fastest': 200ms Be aware that a fast update rate increases the performance load of the application and the control.

6.1.5 Dynamic fill color

Variable for Dynamic FillColor	The linked variable, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.BLINK globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive paramters is normaly '0'
Variable Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be checked against the operators and statments of the defined fill colors. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
No of Fill Color	Number of fill color for the widget. Minimum Value : 1 Maximum Value : 16 First match will be taken and rest is ignored. E.g. if "Operator" and "Statement" are matching for fill color 3, then the statements for all following defined fill colors (4,5,...) will be ignored. Hint: Only 'Range' operator (x to Y) is possible if number of fill colors is more then 1

6.1.6 Dynamic movements

Position X	Movement of widget along X - absolute: from left edge of the screen - relative: from the original placement of the widget
Variable X	Linked variable controls the movement of the element in X-direction, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive parameters is normally '0'
Variable X Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
Position Y	Movement of widget along Y - absolute: from left edge of the screen - relative: from the original placement of the widget

Variable Y	Linked variable controls the movement of the element in Y-direction, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive paramters is normaly '0'
Variable Y Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".

6.1.7 Dynamic size

Variable width	Linked variable controls the width of the object, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0047, Position command value' with '0'= control Id, '22'= drive address: 0,22,S-0-0047 Hint: Control Id for drive paramters is normaly '0'
Variable width script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".

Variable height	Linked variable controls the height of the object, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0047, Position command value' with '0'= control Id, '22'= drive address: 0,22,S-0-0047 Hint: Control Id for drive paramters is normaly '0'
Variable height script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".

6.1.8 Dynamic rotation

Variable for Dynamic Rotation	Linked variable controls the rotation of the element, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive paramters is normaly '0'
Variable Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
Rotation base	Specifies the point, where the rotation is based of

6.1.9 Events

OnClick	The event is executed if a full click is performed on the element, ie. a mouse press and a mouse release
OnMouseUp	The event is executed if a mouse button is released on this element
OnMouseDown	The event is executed if a mouse button is pressed on this element
OnMouseOver	The event is fires when when the mouse cursor enters the element
OnMouseOut	The event is fires when the mouse cursor leaves the element
OnValueChange	The event is fires when the assigned value has been changed
OnRightClick	This event fires when the right mouse button is pressed
OnBlur	This event is raised when the element loses focus
OnDoubleClick	This event fires on double mouse click
OnFocus	This event fires when the element gets focus

Push value	The linked variable, which is either entered or selected via the process variable browser, is set to TRUE (or FALSE) as long as the mouse is getting pressed. This is a convenience functionality and usually used with the "Button" widget and useful to set bool variables or single bits of an integer (use {bit} to set a single bit of integers). Depending on the 'Push value FALSE' attribute, either TRUE or FALSE is set when the button is pressed. Be aware that this feature used with active widgets (like "Input field" or "Chechbox") can cause problems when the same linked variable of the text properties is used because both try to write to the same variable independently! e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.myBool ns=2;s=Application.globalVAR.myInt{7}, with '{7}' = only this bit is toggled JavaScript variables: globalVariables.myBool screenVariables.myInt{14}, with '{14}' = only this bit is toggled Control-Parameter 'C-0-0080, User defined decimal number', with '{4}' = only this bit is used: C-0-0080{4} Axis-Parameter 'A-0-0048, Velocity polarities' with '0'= control Id, '3'= axis address, '{0}' = only this bit is toggled: 0,3,A-0-0048{0} Drive-Parameter 'P-0-0008, Activation E-Stop function' with '0'= control Id, '22'= drive address, '{1}' = only this bit is toggled: 0,22,P-0-0008{1} Hint: Control Id for drive paramters is usually '0'
------------	---

Attachment: Widget attribute

Push value FALSE	If set to 'true', the 'Push value' written to the linked variable as long as the mouse is getting pressed is FALSE. Attribute is 'false' (default) --> TRUE is written to the linked variable as long as the mouse is getting pressed Attribute is 'true' --> FALSE is written to the linked variable as long as the mouse is getting pressed
Toggle	The linked variable, which is either entered or selected via the process variable browser, is toggled between TRUE and FALSE when clicked. This is a convenience functionality and only useful to toggle bool variables or single bits of an integer (use {bit} to toggle a single bit of integers). Be aware that this feature used with active widgets (like "Input field" or "Checkbox") can cause problems when the same linked variable of the text properties is used because both try to write to the same variable independently! e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.myBool ns=2;s=Application.globalVAR.myInt{7}, with '{7}' = only this bit is toggled JavaScript variables: globalVariables.myBool screenVariables.myInt{14}, with '{14}' = only this bit is toggled Control-Parameter 'C-0-0080, User defined decimal number', with '{4}' = only this bit is used: C-0-0080{4} Axis-Parameter 'A-0-0048, Velocity polarities' with '0'= control Id, '3'= axis address, '{0}' = only this bit is toggled: 0,3,A-0-0048{0} Drive-Parameter 'P-0-0008, Activation E-Stop function' with '0'= control Id, '22'= drive address, '{1}' = only this bit is toggled: 0,22,P-0-0008{1}

6.2 Button

6.2.1 General

Type	Type of the element
Id	Id of the element
Tab Id	Sets the tab order
Alias	Unique widget identifier

6.2.2 Position

X position	Distance from the left edge in pixels
Y position	Distance from the top edge in pixels
Width	Width of the element in pixel, including border
Height	Height of the element in pixel, including border
Rotation	Rotation of the element about its center in degrees

6.2.3 Style

Fill color	Background fill color
Border color	The border color, it can only be selected if no theme is selected.
Border width	Border width in pixel
Transparency	A value between 0 and 100, 0: the object appears completely transparent, 100: the object completely covers the background
Shadow	Show a drop shadow

6.2.4 Text properties

Text	Entered text will be shown. If a variable is entered in the 'Variable' attribute of the 'Text properties', the value of the linked variable will be shown respectively inserted if one of the below explained formatters is used inside of the text. printf formatters, %[width][.precision]specifier. Specifier: %b, yields an integer as a binary number %c, yields an integer as the character with that ASCII value %d or %i, yields an integer as a signed decimal number %e, yields a float using scientific notation %u, yields an integer as an unsigned decimal number %f, yields a float as is; see example for precision %o, yields an integer as an octal number %s, yields a string as is %x, yields an integer as a hexadecimal number (lower-case) %X, yields an integer as a hexadecimal number (upper-case) %j, yields a JavaScript object or array as a JSON encoded string e.g. The color: %s, value=blue => The color: blue First number: %d, value=12345 => First number: 12345 Second number: %04d, value=25 => Second number: 0025 Third number: %i, value=1234 => Third number: 1234 Float number: %3.2f, value=3.14159 => Float number: 3.14 Hexadecimal: %x, value=255 => Hexadecimal: ff Octal: %o, value=255 => Octal: 377 Unsigned value: %u, value=150 => Unsigned value: 150 html tags can also be used for formatting purposes bold text
 line break ordered list list item important text e.g. This is normal text - and this is bold text, defines bold text areas My text in first line
 my text in second line, defines a single line break first line second line third line, defines an ordered list see also:
------	---

Variable	The linked variable, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.BLINK globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive paramters is normaly '0'
Font color	The color of the font, it can only be selected if no theme is selected.
Font size	Font size can only be selected if no theme is selected.
Font Name	Font Name can only be selected if no theme is selected.
Custom Filter	Name of filter function to format the displayed value.
Variable Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
Tooltip	Text that is displayed when the mouse pointer is over the item. Be aware that in case of a touch device the tooltip can't be displayed.
Horizontal alignment	Alignment of the contents horizontally (left, centered or right)
Update interval	Length of the update rate The following update rates are available: 'none': Value is only read once when screen is opened 'slowest': 10000ms 'slow': 5000ms 'default': 1000ms 'fast': 500ms 'fastest': 200ms Be aware that a fast update rate increases the performance load of the application and the control.

6.2.5 Dynamic fill color

Variable for Dynamic FillColor	The linked variable, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.BLINK globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive paramters is normaly '0'
Variable Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be checked against the operators and statments of the defined fill colors. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
No of Fill Color	Number of fill color for the widget. Minimum Value : 1 Maximum Value : 16 First match will be taken and rest is ignored. E.g. if "Operator" and "Statement" are matching for fill color 3, then the statements for all following defined fill colors (4,5,...) will be ignored. Hint: Only 'Range' operator (x to Y) is possible if number of fill colors is more then 1

6.2.6 Dynamic movements

Position X	Movement of widget along X - absolute: from left edge of the screen - relative: from the original placement of the widget
Variable X	Linked variable controls the movement of the element in X-direction, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive parameters is normally '0'
Variable X Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
Position Y	Movement of widget along Y - absolute: from left edge of the screen - relative: from the original placement of the widget

Attachment: Widget attribute

Variable Y	Linked variable controls the movement of the element in Y-direction, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive paramters is normaly '0'
Variable Y Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".

6.2.7 Dynamic size

Variable width	Linked variable controls the width of the object, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0047, Position command value' with '0'= control Id, '22'= drive address: 0,22,S-0-0047 Hint: Control Id for drive paramters is normaly '0'
Variable width script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".

Attachment: Widget attribute

Variable height	Linked variable controls the height of the object, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0047, Position command value' with '0'= control Id, '22'= drive address: 0,22,S-0-0047 Hint: Control Id for drive paramters is normaly '0'
Variable height script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".

6.2.8 Dynamic rotation

Variable for Dynamic Rotation	Linked variable controls the rotation of the element, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive paramters is normaly '0'
Variable Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
Rotation base	Specifies the point, where the rotation is based of

6.2.9 Events

OnMouseClicked	The event is executed if a full click is performed on the element, ie. a mouse press and a mouse release
OnMouseUp	The event is executed if a mouse button is released on this element
OnMouseDown	The event is executed if a mouse button is pressed on this element
OnMouseOver	The event is fires when when the mouse cursor enters the element
OnMouseOut	The event is fires when the mouse cursor leaves the element
OnRightClick	This event fires when the right mouse button is pressed
OnDoubleClick	This event fires on double mouse click

Attachment: Widget attribute

Push value	The linked variable, which is either entered or selected via the process variable browser, is set to TRUE (or FALSE) as long as the mouse is getting pressed. This is a convenience functionality and usually used with the "Button" widget and useful to set bool variables or single bits of an integer (use {bit} to set a single bit of integers). Depending on the 'Push value FALSE' attribute, either TRUE or FALSE is set when the button is pressed. Be aware that this feature used with active widgets (like "Input field" or "Checkbox") can cause problems when the same linked variable of the text properties is used because both try to write to the same variable independently! e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.myBool ns=2;s=Application.globalVAR.myInt{7}, with '{7}' = only this bit is toggled JavaScript variables: globalVariables.myBool screenVariables.myInt{14}, with '{14}' = only this bit is toggled Control-Parameter 'C-0-0080, User defined decimal number', with '{4}' = only this bit is used: C-0-0080{4} Axis-Parameter 'A-0-0048, Velocity polarities' with '0'= control Id, '3'= axis address, '{0}' = only this bit is toggled: 0,3,A-0-0048{0} Drive-Parameter 'P-0-0008, Activation E-Stop function' with '0'= control Id, '22'= drive address, '{1}' = only this bit is toggled: 0,22,P-0-0008{1} Hint: Control Id for drive parameters is usually '0'
------------	--

Push value FALSE	If set to 'true', the 'Push value' written to the linked variable as long as the mouse is getting pressed is FALSE. Attribute is 'false' (default) --> TRUE is written to the linked variable as long as the mouse is getting pressed Attribute is 'true' --> FALSE is written to the linked variable as long as the mouse is getting pressed
Toggle	The linked variable, which is either entered or selected via the process variable browser, is toggled between TRUE and FALSE when clicked. This is a convenience functionality and only useful to toggle bool variables or single bits of an integer (use {bit} to toggle a single bit of integers). Be aware that this feature used with active widgets (like "Input field" or "Checkbox") can cause problems when the same linked variable of the text properties is used because both try to write to the same variable independently! e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.myBool ns=2;s=Application.globalVAR.myInt{7}, with '{7}' = only this bit is toggled JavaScript variables: globalVariables.myBool screenVariables.myInt{14}, with '{14}' = only this bit is toggled Control-Parameter 'C-0-0080, User defined decimal number', with '{4}' = only this bit is used: C-0-0080{4} Axis-Parameter 'A-0-0048, Velocity polarities' with '0'= control Id, '3'= axis address, '{0}' = only this bit is toggled: 0,3,A-0-0048{0} Drive-Parameter 'P-0-0008, Activation E-Stop function' with '0'= control Id, '22'= drive address, '{1}' = only this bit is toggled: 0,22,P-0-0008{1}

6.3 Chart

6.3.1 General

Type	Type of the element
Id	Id of the element
Alias	Unique widget identifier

6.3.2 Position

X position	Distance from the left edge in pixels
Y position	Distance from the top edge in pixels
Width	Width of the element in pixel, including border
Height	Height of the element in pixel, including border
Rotation	Rotation of the element about its center in degrees

6.3.3 Style

Fill color	Background fill color
Border color	The border color, it can only be selected if no theme is selected.
Border width	Border width in pixel
Transparency	A value between 0 and 100, 0: the object appears completely transparent, 100: the object completely covers the background
Shadow	Show a drop shadow

6.3.4 Text properties

Font color	The color of the font, it can only be selected if no theme is selected.
Font size	Font size can only be selected if no theme is selected.
Font Name	Font Name can only be selected if no theme is selected.
Custom Filter	Name of filter function to format the displayed value.
Variable Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50 , if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset , if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
Tooltip	Text that is displayed when the mouse pointer is over the item. Be aware that in case of a touch device the tooltip can't be displayed.
Update interval	Length of the update rate The following update rates are available: 'none': Value is only read once when screen is opened 'slowest': 10000ms 'slow': 5000ms 'default': 1000ms 'fast': 500ms 'fastest': 200ms Be aware that a fast update rate increases the performance load of the application and the control.

6.3.5 Dynamic fill color

Variable for Dynamic FillColor	The linked variable, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.BLINK globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive paramters is normaly '0'
Variable Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be checked against the operators and statments of the defined fill colors. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
No of Fill Color	Number of fill color for the widget. Minimum Value : 1 Maximum Value : 16 First match will be taken and rest is ignored. E.g. if "Operator" and "Statement" are matching for fill color 3, then the statements for all following defined fill colors (4,5,...) will be ignored. Hint: Only 'Range' operator (x to Y) is possible if number of fill colors is more then 1

6.3.6 Dynamic movements

Position X	Movement of widget along X - absolute: from left edge of the screen - relative: from the original placement of the widget
Variable X Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
Position Y	Movement of widget along Y - absolute: from left edge of the screen - relative: from the original placement of the widget
Variable Y Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".

6.3.7 Dynamic size

Variable width	Linked variable controls the width of the object, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0047, Position command value' with '0'= control Id, '22'= drive address: 0,22,S-0-0047 Hint: Control Id for drive paramters is normaly '0'
Variable width script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".

Attachment: Widget attribute

Variable height	Linked variable controls the height of the object, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0047, Position command value' with '0'= control Id, '22'= drive address: 0,22,S-0-0047 Hint: Control Id for drive paramters is normaly '0'
Variable height script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".

6.3.8 Dynamic rotation

Variable Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
Rotation base	Specifies the point, where the rotation is based of

6.3.9 Events

OnClick	The event is executed if a full click is performed on the element, ie. a mouse press and a mouse release
OnMouseUp	The event is executed if a mouse button is released on this element
OnMouseDown	The event is executed if a mouse button is pressed on this element
OnMouseOver	The event is fires when when the mouse cursor enters the element
OnMouseOut	The event is fires when the mouse cursor leaves the element

OnRightClick	This event fires when the right mouse button is pressed
OnDoubleClick	This event fires on double mouse click
Push value	The linked variable, which is either entered or selected via the process variable browser, is set to TRUE (or FALSE) as long as the mouse is getting pressed. This is a convenience functionality and usually used with the "Button" widget and useful to set bool variables or single bits of an integer (use {bit} to set a single bit of integers). Depending on the 'Push value FALSE' attribute, either TRUE or FALSE is set when the button is pressed. Be aware that this feature used with active widgets (like "Input field" or "Chechbox") can cause problems when the same linked variable of the text properties is used because both try to write to the same variable independently! e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.myBool ns=2;s=Application.globalVAR.myInt{7}, with '{7}' = only this bit is toggled JavaScript variables: globalVariables.myBool screenVariables.myInt{14}, with '{14}' = only this bit is toggled Control-Parameter 'C-0-0080, User defined decimal number', with '{4}' = only this bit is used: C-0-0080{4} Axis-Parameter 'A-0-0048, Velocity polarities' with '0'= control Id, '3'= axis address, '{0}' = only this bit is toggled: 0,3,A-0-0048{0} Drive-Parameter 'P-0-0008, Activation E-Stop function' with '0'= control Id, '22'= drive address, '{1}' = only this bit is toggled: 0,22,P-0-0008{1} Hint: Control Id for drive paramters is usually '0'

Attachment: Widget attribute

Push value FALSE	If set to 'true', the 'Push value' written to the linked variable as long as the mouse is getting pressed is FALSE. Attribute is 'false' (default) --> TRUE is written to the linked variable as long as the mouse is getting pressed Attribute is 'true' --> FALSE is written to the linked variable as long as the mouse is getting pressed
Toggle	The linked variable, which is either entered or selected via the process variable browser, is toggled between TRUE and FALSE when clicked. This is a convenience functionality and only useful to toggle bool variables or single bits of an integer (use {bit} to toggle a single bit of integers). Be aware that this feature used with active widgets (like "Input field" or "Checkbox") can cause problems when the same linked variable of the text properties is used because both try to write to the same variable independently! e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.myBool ns=2;s=Application.globalVAR.myInt{7}, with '{7}' = only this bit is toggled JavaScript variables: globalVariables.myBool screenVariables.myInt{14}, with '{14}' = only this bit is toggled Control-Parameter 'C-0-0080, User defined decimal number', with '{4}' = only this bit is used: C-0-0080{4} Axis-Parameter 'A-0-0048, Velocity polarities' with '0'= control Id, '3'= axis address, '{0}' = only this bit is toggled: 0,3,A-0-0048{0} Drive-Parameter 'P-0-0008, Activation E-Stop function' with '0'= control Id, '22'= drive address, '{1}' = only this bit is toggled: 0,22,P-0-0008{1}

6.4 Checkbox

6.4.1 General

Type	Type of the element
Id	Id of the element
Tab Id	Sets the tab order
Alias	Unique widget identifier

6.4.2 Position

X position	Distance from the left edge in pixels
Y position	Distance from the top edge in pixels
Width	Width of the element in pixel, including border
Height	Height of the element in pixel, including border
Rotation	Rotation of the element about its center in degrees

6.4.3 Style

Fill color	Background fill color
Border color	The border color, it can only be selected if no theme is selected.
Border width	Border width in pixel
Transparency	A value between 0 and 100, 0: the object appears completely transparent, 100: the object completely covers the background
Shadow	Show a drop shadow

6.4.4 Text properties

Text	Entered text will be shown. If a variable is entered in the 'Variable' attribute of the 'Text properties', the value of the linked variable will be shown respectively inserted if one of the below explained formatters is used inside of the text. printf formatters, %[width][.precision]specifier. Specifier: %b, yields an integer as a binary number %c, yields an integer as the character with that ASCII value %d or %i, yields an integer as a signed decimal number %e, yields a float using scientific notation %u, yields an integer as an unsigned decimal number %f, yields a float as is; see example for precision %o, yields an integer as an octal number %s, yields a string as is %x, yields an integer as a hexadecimal number (lower-case) %X, yields an integer as a hexadecimal number (upper-case) %j, yields a JavaScript object or array as a JSON encoded string e.g. The color: %s, value=blue => The color: blue First number: %d, value=12345 => First number: 12345 Second number: %04d, value=25 => Second number: 0025 Third number: %i, value=1234 => Third number: 1234 Float number: %3.2f, value=3.14159 => Float number: 3.14 Hexadecimal: %x, value=255 => Hexadecimal: ff Octal: %o, value=255 => Octal: 377 Unsigned value: %u, value=150 => Unsigned value: 150 html tags can also be used for formatting purposes bold text
 line break ordered list list item important text e.g. This is normal text - and this is bold text, defines bold text areas My text in first line
 my text in second line, defines a single line break first line second line third line, defines an ordered list see also:
------	---

Variable	The linked variable, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.BLINK globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive paramters is normaly '0'
Font color	The color of the font, it can only be selected if no theme is selected.
Font size	Font size can only be selected if no theme is selected.
Font Name	Font Name can only be selected if no theme is selected.
Custom Filter	Name of filter function to format the displayed value.
Variable Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
Tooltip	Text that is displayed when the mouse pointer is over the item. Be aware that in case of a touch device the tooltip can't be displayed.
Update interval	Length of the update rate The following update rates are available: 'none': Value is only read once when screen is opened 'slowest': 10000ms 'slow': 5000ms 'default': 1000ms 'fast': 500ms 'fastest': 200ms Be aware that a fast update rate increases the performance load of the application and the control.

6.4.5 Dynamic fill color

Variable for Dynamic FillColor	The linked variable, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.BLINK globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive paramters is normaly '0'
Variable Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be checked against the operators and statments of the defined fill colors. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
No of Fill Color	Number of fill color for the widget. Minimum Value : 1 Maximum Value : 16 First match will be taken and rest is ignored. E.g. if "Operator" and "Statement" are matching for fill color 3, then the statements for all following defined fill colors (4,5,...) will be ignored. Hint: Only 'Range' operator (x to Y) is possible if number of fill colors is more then 1

6.4.6 Dynamic movements

Position X	Movement of widget along X - absolute: from left edge of the screen - relative: from the original placement of the widget
Variable X	Linked variable controls the movement of the element in X-direction, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive parameters is normally '0'
Variable X Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
Position Y	Movement of widget along Y - absolute: from left edge of the screen - relative: from the original placement of the widget

Attachment: Widget attribute

Variable Y	Linked variable controls the movement of the element in Y-direction, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive paramters is normaly '0'
Variable Y Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".

6.4.7 Dynamic size

Variable width	Linked variable controls the width of the object, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0047, Position command value' with '0'= control Id, '22'= drive address: 0,22,S-0-0047 Hint: Control Id for drive paramters is normaly '0'
Variable width script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".

Attachment: Widget attribute

Variable height	Linked variable controls the height of the object, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0047, Position command value' with '0'= control Id, '22'= drive address: 0,22,S-0-0047 Hint: Control Id for drive paramters is normaly '0'
Variable height script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".

6.4.8 Dynamic rotation

Variable for Dynamic Rotation	Linked variable controls the rotation of the element, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive paramters is normaly '0'
Variable Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
Rotation base	Specifies the point, where the rotation is based of

6.4.9 Events

OnMouseClicked	The event is executed if a full click is performed on the element, ie. a mouse press and a mouse release
OnMouseUp	The event is executed if a mouse button is released on this element
OnMouseDown	The event is executed if a mouse button is pressed on this element
OnMouseOver	The event is fires when when the mouse cursor enters the element
OnMouseOut	The event is fires when the mouse cursor leaves the element
OnRightClick	This event fires when the right mouse button is pressed
OnDoubleClick	This event fires on double mouse click

Attachment: Widget attribute

Push value	The linked variable, which is either entered or selected via the process variable browser, is set to TRUE (or FALSE) as long as the mouse is getting pressed. This is a convenience functionality and usually used with the "Button" widget and useful to set bool variables or single bits of an integer (use {bit} to set a single bit of integers). Depending on the 'Push value FALSE' attribute, either TRUE or FALSE is set when the button is pressed. Be aware that this feature used with active widgets (like "Input field" or "Checkbox") can cause problems when the same linked variable of the text properties is used because both try to write to the same variable independently! e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.myBool ns=2;s=Application.globalVAR.myInt{7}, with '{7}' = only this bit is toggled JavaScript variables: globalVariables.myBool screenVariables.myInt{14}, with '{14}' = only this bit is toggled Control-Parameter 'C-0-0080, User defined decimal number', with '{4}' = only this bit is used: C-0-0080{4} Axis-Parameter 'A-0-0048, Velocity polarities' with '0'= control Id, '3'= axis address, '{0}' = only this bit is toggled: 0,3,A-0-0048{0} Drive-Parameter 'P-0-0008, Activation E-Stop function' with '0'= control Id, '22'= drive address, '{1}' = only this bit is toggled: 0,22,P-0-0008{1} Hint: Control Id for drive parameters is usually '0'
------------	--

Push value FALSE	If set to 'true', the 'Push value' written to the linked variable as long as the mouse is getting pressed is FALSE. Attribute is 'false' (default) --> TRUE is written to the linked variable as long as the mouse is getting pressed Attribute is 'true' --> FALSE is written to the linked variable as long as the mouse is getting pressed
Toggle	The linked variable, which is either entered or selected via the process variable browser, is toggled between TRUE and FALSE when clicked. This is a convenience functionality and only useful to toggle bool variables or single bits of an integer (use {bit} to toggle a single bit of integers). Be aware that this feature used with active widgets (like "Input field" or "Checkbox") can cause problems when the same linked variable of the text properties is used because both try to write to the same variable independently! e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.myBool ns=2;s=Application.globalVAR.myInt{7}, with '{7}' = only this bit is toggled JavaScript variables: globalVariables.myBool screenVariables.myInt{14}, with '{14}' = only this bit is toggled Control-Parameter 'C-0-0080, User defined decimal number', with '{4}' = only this bit is used: C-0-0080{4} Axis-Parameter 'A-0-0048, Velocity polarities' with '0'= control Id, '3'= axis address, '{0}' = only this bit is toggled: 0,3,A-0-0048{0} Drive-Parameter 'P-0-0008, Activation E-Stop function' with '0'= control Id, '22'= drive address, '{1}' = only this bit is toggled: 0,22,P-0-0008{1}

6.5 Diagnosis events

6.5.1 General

Type	Type of the element
Id	Id of the element
Alias	Unique widget identifier

6.5.2 Position

X position	Distance from the left edge in pixels
Y position	Distance from the top edge in pixels
Width	Width of the element in pixel, including border
Height	Height of the element in pixel, including border
Rotation	Rotation of the element about its center in degrees

6.5.3 Style

Fill color	Background fill color
Border color	The border color, it can only be selected if no theme is selected.
Border width	Border width in pixel
Transparency	A value between 0 and 100, 0: the object appears completely transparent, 100: the object completely covers the background
Shadow	Show a drop shadow

6.5.4 Text properties

Text	Entered text will be shown. If a variable is entered in the 'Variable' attribute of the 'Text properties', the value of the linked variable will be shown respectively inserted if one of the below explained formatters is used inside of the text. printf formatters, %[width][.precision]specifier. Specifier: %b, yields an integer as a binary number %c, yields an integer as the character with that ASCII value %d or %i, yields an integer as a signed decimal number %e, yields a float using scientific notation %u, yields an integer as an unsigned decimal number %f, yields a float as is; see example for precision %o, yields an integer as an octal number %s, yields a string as is %x, yields an integer as a hexadecimal number (lower-case) %X, yields an integer as a hexadecimal number (upper-case) %j, yields a JavaScript object or array as a JSON encoded string e.g. The color: %s, value=blue => The color: blue First number: %d, value=12345 => First number: 12345 Second number: %04d, value=25 => Second number: 0025 Third number: %i, value=1234 => Third number: 1234 Float number: %3.2f, value=3.14159 => Float number: 3.14 Hexadecimal: %x, value=255 => Hexadecimal: ff Octal: %o, value=255 => Octal: 377 Unsigned value: %u, value=150 => Unsigned value: 150 html tags can also be used for formatting purposes bold text
 line break ordered list list item important text e.g This is normal text - and this is bold text, defines bold text areas My text in first line
 my text in second line, defines a single line break first line second line third line, defines an ordered list see also:
Font color	The color of the font, it can only be selected if no theme is selected.
Font size	Font size can only be selected if no theme is selected.
Font Name	Font Name can only be selected if no theme is selected.
Custom Filter	Name of filter function to format the displayed value.

Attachment: Widget attribute

Variable Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
Tooltip	Text that is displayed when the mouse pointer is over the item. Be aware that in case of a touch device the tooltip can't be displayed.

6.5.5 Dynamic fill color

Variable for Dynamic FillColor	The linked variable, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.BLINK globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive paramters is normaly '0'
Variable Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be checked against the operators and statments of the defined fill colors. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
No of Fill Color	Number of fill color for the widget. Minimum Value : 1 Maximum Value : 16 First match will be taken and rest is ignored. E.g. if "Operator" and "Statement" are matching for fill color 3, then the statements for all following defined fill colors (4,5,...) will be ignored. Hint: Only 'Range' operator (x to Y) is possible if number of fill colors is more then 1

6.5.6 Dynamic movements

Position X	Movement of widget along X - absolute: from left edge of the screen - relative: from the original placement of the widget
Variable X Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
Position Y	Movement of widget along Y - absolute: from left edge of the screen - relative: from the original placement of the widget
Variable Y Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".

6.5.7 Dynamic size

Variable width	Linked variable controls the width of the object, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0047, Position command value' with '0'= control Id, '22'= drive address: 0,22,S-0-0047 Hint: Control Id for drive paramters is normaly '0'
Variable width script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".

Variable height	Linked variable controls the height of the object, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0047, Position command value' with '0'= control Id, '22'= drive address: 0,22,S-0-0047 Hint: Control Id for drive paramters is normaly '0'
Variable height script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".

6.5.8 Dynamic rotation

Variable Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
Rotation base	Specifies the point, where the rotation is based of

6.5.9 Events

OnClick	The event is executed if a full click is performed on the element, ie. a mouse press and a mouse release
OnMouseUp	The event is executed if a mouse button is released on this element
OnMouseDown	The event is executed if a mouse button is pressed on this element
OnMouseOver	The event is fires when when the mouse cursor enters the element
OnMouseOut	The event is fires when the mouse cursor leaves the element

Attachment: Widget attribute

OnRightClick	This event fires when the right mouse button is pressed
OnDoubleClick	This event fires on double mouse click
Push value	The linked variable, which is either entered or selected via the process variable browser, is set to TRUE (or FALSE) as long as the mouse is getting pressed. This is a convenience functionality and usually used with the "Button" widget and useful to set bool variables or single bits of an integer (use {bit} to set a single bit of integers). Depending on the 'Push value FALSE' attribute, either TRUE or FALSE is set when the button is pressed. Be aware that this feature used with active widgets (like "Input field" or "Chechbox") can cause problems when the same linked variable of the text properties is used because both try to write to the same variable independently! e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.myBool ns=2;s=Application.globalVAR.myInt{7}, with '{7}' = only this bit is toggled JavaScript variables: globalVariables.myBool screenVariables.myInt{14}, with '{14}' = only this bit is toggled Control-Parameter 'C-0-0080, User defined decimal number', with '{4}' = only this bit is used: C-0-0080{4} Axis-Parameter 'A-0-0048, Velocity polarities' with '0'= control Id, '3'= axis address, '{0}' = only this bit is toggled: 0,3,A-0-0048{0} Drive-Parameter 'P-0-0008, Activation E-Stop function' with '0'= control Id, '22'= drive address, '{1}' = only this bit is toggled: 0,22,P-0-0008{1} Hint: Control Id for drive paramters is usually '0'

Push value FALSE	If set to 'true', the 'Push value' written to the linked variable as long as the mouse is getting pressed is FALSE. Attribute is 'false' (default) --> TRUE is written to the linked variable as long as the mouse is getting pressed Attribute is 'true' --> FALSE is written to the linked variable as long as the mouse is getting pressed
Toggle	The linked variable, which is either entered or selected via the process variable browser, is toggled between TRUE and FALSE when clicked. This is a convenience functionality and only useful to toggle bool variables or single bits of an integer (use {bit} to toggle a single bit of integers). Be aware that this feature used with active widgets (like "Input field" or "Checkbox") can cause problems when the same linked variable of the text properties is used because both try to write to the same variable independently! e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.myBool ns=2;s=Application.globalVAR.myInt{7}, with '{7}' = only this bit is toggled JavaScript variables: globalVariables.myBool screenVariables.myInt{14}, with '{14}' = only this bit is toggled Control-Parameter 'C-0-0080, User defined decimal number', with '{4}' = only this bit is used: C-0-0080{4} Axis-Parameter 'A-0-0048, Velocity polarities' with '0'= control Id, '3'= axis address, '{0}' = only this bit is toggled: 0,3,A-0-0048{0} Drive-Parameter 'P-0-0008, Activation E-Stop function' with '0'= control Id, '22'= drive address, '{1}' = only this bit is toggled: 0,22,P-0-0008{1}

6.6 Ellipse

6.6.1 General

Type	Type of the element
Id	Id of the element
Alias	Unique widget identifier

6.6.2 Position

X position	Distance from the left edge in pixels
Y position	Distance from the top edge in pixels
Width	Width of the element in pixel, including border
Height	Height of the element in pixel, including border
Rotation	Rotation of the element about its center in degrees

6.6.3 Style

Fill color	Background fill color
Border color	The border color, it can only be selected if no theme is selected.
Border width	Border width in pixel
Transparency	A value between 0 and 100, 0: the object appears completely transparent, 100: the object completely covers the background
Shadow	Show a drop shadow

6.6.4 Text properties

Text	Entered text will be shown. If a variable is entered in the 'Variable' attribute of the 'Text properties', the value of the linked variable will be shown respectively inserted if one of the below explained formatters is used inside of the text. printf formatters, %[width][.precision]specifier. Specifier: %b, yields an integer as a binary number %c, yields an integer as the character with that ASCII value %d or %i, yields an integer as a signed decimal number %e, yields a float using scientific notation %u, yields an integer as an unsigned decimal number %f, yields a float as is; see example for precision %o, yields an integer as an octal number %s, yields a string as is %x, yields an integer as a hexadecimal number (lower-case) %X, yields an integer as a hexadecimal number (upper-case) %j, yields a JavaScript object or array as a JSON encoded string e.g. The color: %s, value=blue => The color: blue First number: %d, value=12345 => First number: 12345 Second number: %04d, value=25 => Second number: 0025 Third number: %i, value=1234 => Third number: 1234 Float number: %3.2f, value=3.14159 => Float number: 3.14 Hexadecimal: %x, value=255 => Hexadecimal: ff Octal: %o, value=255 => Octal: 377 Unsigned value: %u, value=150 => Unsigned value: 150 html tags can also be used for formatting purposes bold text
 line break ordered list list item important text e.g This is normal text - and this is bold text, defines bold text areas My text in first line
 my text in second line, defines a single line break first line second line third line, defines an ordered list see also:
------	--

Attachment: Widget attribute

Variable	The linked variable, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.BLINK globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive paramters is normaly '0'
Font color	The color of the font, it can only be selected if no theme is selected.
Font size	Font size can only be selected if no theme is selected.
Font Name	Font Name can only be selected if no theme is selected.
Custom Filter	Name of filter function to format the displayed value.
Variable Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
Tooltip	Text that is displayed when the mouse pointer is over the item. Be aware that in case of a touch device the tooltip can't be displayed.
Horizontal alignment	Alignment of the contents horizontally (left, centered or right)
Vertical alignment	Alignment of the contents vertically (top, centered or bottom)
Update interval	Length of the update rate The following update rates are available: 'none': Value is only read once when screen is opened 'slowest': 10000ms 'slow': 5000ms 'default': 1000ms 'fast': 500ms 'fastest': 200ms Be aware that a fast update rate increases the performance load of the application and the control.

6.6.5 Dynamic fill color

Variable for Dynamic FillColor	The linked variable, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.BLINK globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive paramters is normaly '0'
Variable Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be checked against the operators and statments of the defined fill colors. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
No of Fill Color	Number of fill color for the widget. Minimum Value : 1 Maximum Value : 16 First match will be taken and rest is ignored. E.g. if "Operator" and "Statement" are matching for fill color 3, then the statements for all following defined fill colors (4,5,...) will be ignored. Hint: Only 'Range' operator (x to Y) is possible if number of fill colors is more then 1

6.6.6 Dynamic movements

Position X	Movement of widget along X - absolute: from left edge of the screen - relative: from the original placement of the widget
Variable X	Linked variable controls the movement of the element in X-direction, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive parameters is normally '0'
Variable X Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
Position Y	Movement of widget along Y - absolute: from left edge of the screen - relative: from the original placement of the widget

Variable Y	Linked variable controls the movement of the element in Y-direction, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive paramters is normaly '0'
Variable Y Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".

6.6.7 Dynamic size

Variable width	Linked variable controls the width of the object, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0047, Position command value' with '0'= control Id, '22'= drive address: 0,22,S-0-0047 Hint: Control Id for drive paramters is normaly '0'
Variable width script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".

Variable height	Linked variable controls the height of the object, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0047, Position command value' with '0'= control Id, '22'= drive address: 0,22,S-0-0047 Hint: Control Id for drive paramters is normaly '0'
Variable height script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".

6.6.8 Dynamic rotation

Variable for Dynamic Rotation	Linked variable controls the rotation of the element, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive paramters is normaly '0'
Variable Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
Rotation base	Specifies the point, where the rotation is based of

6.6.9 Events

OnClick	The event is executed if a full click is performed on the element, ie. a mouse press and a mouse release
OnMouseUp	The event is executed if a mouse button is released on this element
OnMouseDown	The event is executed if a mouse button is pressed on this element
OnMouseOver	The event is fires when when the mouse cursor enters the element
OnMouseOut	The event is fires when the mouse cursor leaves the element
OnRightClick	This event fires when the right mouse button is pressed
OnDoubleClick	This event fires on double mouse click

Push value	The linked variable, which is either entered or selected via the process variable browser, is set to TRUE (or FALSE) as long as the mouse is getting pressed. This is a convenience functionality and usually used with the "Button" widget and useful to set bool variables or single bits of an integer (use {bit} to set a single bit of integers). Depending on the 'Push value FALSE' attribute, either TRUE or FALSE is set when the button is pressed. Be aware that this feature used with active widgets (like "Input field" or "Chechbox") can cause problems when the same linked variable of the text properties is used because both try to write to the same variable independently! e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.myBool ns=2;s=Application.globalVAR.myInt{7}, with '{7}' = only this bit is toggled JavaScript variables: globalVariables.myBool screenVariables.myInt{14}, with '{14}' = only this bit is toggled Control-Parameter 'C-0-0080, User defined decimal number', with '{4}' = only this bit is used: C-0-0080{4} Axis-Parameter 'A-0-0048, Velocity polarities' with '0'= control Id, '3'= axis address, '{0}' = only this bit is toggled: 0,3,A-0-0048{0} Drive-Parameter 'P-0-0008, Activation E-Stop function' with '0'= control Id, '22'= drive address, '{1}' = only this bit is toggled: 0,22,P-0-0008{1} Hint: Control Id for drive paramters is usually '0'
------------	---

Attachment: Widget attribute

Push value FALSE	If set to 'true', the 'Push value' written to the linked variable as long as the mouse is getting pressed is FALSE. Attribute is 'false' (default) --> TRUE is written to the linked variable as long as the mouse is getting pressed Attribute is 'true' --> FALSE is written to the linked variable as long as the mouse is getting pressed
Toggle	The linked variable, which is either entered or selected via the process variable browser, is toggled between TRUE and FALSE when clicked. This is a convenience functionality and only useful to toggle bool variables or single bits of an integer (use {bit} to toggle a single bit of integers). Be aware that this feature used with active widgets (like "Input field" or "Checkbox") can cause problems when the same linked variable of the text properties is used because both try to write to the same variable independently! e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.myBool ns=2;s=Application.globalVAR.myInt{7}, with '{7}' = only this bit is toggled JavaScript variables: globalVariables.myBool screenVariables.myInt{14}, with '{14}' = only this bit is toggled Control-Parameter 'C-0-0080, User defined decimal number', with '{4}' = only this bit is used: C-0-0080{4} Axis-Parameter 'A-0-0048, Velocity polarities' with '0'= control Id, '3'= axis address, '{0}' = only this bit is toggled: 0,3,A-0-0048{0} Drive-Parameter 'P-0-0008, Activation E-Stop function' with '0'= control Id, '22'= drive address, '{1}' = only this bit is toggled: 0,22,P-0-0008{1}

6.7 Frame

6.7.1 General

Type	Type of the element
Id	Id of the element
Alias	Unique widget identifier

6.7.2 Position

X position	Distance from the left edge in pixels
Y position	Distance from the top edge in pixels
Width	Width of the element in pixel, including border
Height	Height of the element in pixel, including border
Rotation	Rotation of the element about its center in degrees

6.7.3 Style

Fill color	Background fill color
Border color	The border color, it can only be selected if no theme is selected.
Border width	Border width in pixel
Transparency	A value between 0 and 100, 0: the object appears completely transparent, 100: the object completely covers the background
Shadow	Show a drop shadow

6.7.4 Text properties

Custom Filter	Name of filter function to format the displayed value.
Tooltip	Text that is displayed when the mouse pointer is over the item. Be aware that in case of a touch device the tooltip can't be displayed.
Update interval	Length of the update rate The following update rates are available: 'none': Value is only read once when screen is opened 'slowest': 10000ms 'slow': 5000ms 'default': 1000ms 'fast': 500ms 'fastest': 200ms Be aware that a fast update rate increases the performance load of the application and the control.

6.7.5 Dynamic fill color

Variable for Dynamic FillColor	The linked variable, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.BLINK globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive paramters is normaly '0'
No of Fill Color	Number of fill color for the widget. Minimum Value : 1 Maximum Value : 16 First match will be taken and rest is ignored. E.g. if "Operator" and "Statement" are matching for fill color 3, then the statements for all following defined fill colors (4,5,...) will be ignored. Hint: Only 'Range' operator (x to Y) is possible if number of fill colors is more then 1

6.7.6 Dynamic movements

Position X	Movement of widget along X - absolute: from left edge of the screen - relative: from the original placement of the widget
Position Y	Movement of widget along Y - absolute: from left edge of the screen - relative: from the original placement of the widget

6.7.7 Dynamic size

Variable width	Linked variable controls the width of the object, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0047, Position command value' with '0'= control Id, '22'= drive address: 0,22,S-0-0047 Hint: Control Id for drive paramters is normaly '0'
Variable width script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".

Attachment: Widget attribute

Variable height	Linked variable controls the height of the object, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0047, Position command value' with '0'= control Id, '22'= drive address: 0,22,S-0-0047 Hint: Control Id for drive paramters is normaly '0'
Variable height script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".

6.7.8 Dynamic rotation

Rotation base	Specifies the point, where the rotation is based of
---------------	---

6.7.9 Events

OnClick	The event is executed if a full click is performed on the element, ie. a mouse press and a mouse release
OnMouseUp	The event is executed if a mouse button is released on this element
OnMouseDown	The event is executed if a mouse button is pressed on this element

Push value	The linked variable, which is either entered or selected via the process variable browser, is set to TRUE (or FALSE) as long as the mouse is getting pressed. This is a convenience functionality and usually used with the "Button" widget and useful to set bool variables or single bits of an integer (use {bit} to set a single bit of integers). Depending on the 'Push value FALSE' attribute, either TRUE or FALSE is set when the button is pressed. Be aware that this feature used with active widgets (like "Input field" or "Chechbox") can cause problems when the same linked variable of the text properties is used because both try to write to the same variable independently! e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.myBool ns=2;s=Application.globalVAR.myInt{7}, with '{7}' = only this bit is toggled JavaScript variables: globalVariables.myBool screenVariables.myInt{14}, with '{14}' = only this bit is toggled Control-Parameter 'C-0-0080, User defined decimal number', with '{4}' = only this bit is used: C-0-0080{4} Axis-Parameter 'A-0-0048, Velocity polarities' with '0'= control Id, '3'= axis address, '{0}' = only this bit is toggled: 0,3,A-0-0048{0} Drive-Parameter 'P-0-0008, Activation E-Stop function' with '0'= control Id, '22'= drive address, '{1}' = only this bit is toggled: 0,22,P-0-0008{1} Hint: Control Id for drive paramters is usually '0'
------------	---

Attachment: Widget attribute

Push value FALSE	If set to 'true', the 'Push value' written to the linked variable as long as the mouse is getting pressed is FALSE. Attribute is 'false' (default) --> TRUE is written to the linked variable as long as the mouse is getting pressed Attribute is 'true' --> FALSE is written to the linked variable as long as the mouse is getting pressed
Toggle	The linked variable, which is either entered or selected via the process variable browser, is toggled between TRUE and FALSE when clicked. This is a convenience functionality and only useful to toggle bool variables or single bits of an integer (use {bit} to toggle a single bit of integers). Be aware that this feature used with active widgets (like "Input field" or "Checkbox") can cause problems when the same linked variable of the text properties is used because both try to write to the same variable independently! e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.myBool ns=2;s=Application.globalVAR.myInt{7}, with '{7}' = only this bit is toggled JavaScript variables: globalVariables.myBool screenVariables.myInt{14}, with '{14}' = only this bit is toggled Control-Parameter 'C-0-0080, User defined decimal number', with '{4}' = only this bit is used: C-0-0080{4} Axis-Parameter 'A-0-0048, Velocity polarities' with '0'= control Id, '3'= axis address, '{0}' = only this bit is toggled: 0,3,A-0-0048{0} Drive-Parameter 'P-0-0008, Activation E-Stop function' with '0'= control Id, '22'= drive address, '{1}' = only this bit is toggled: 0,22,P-0-0008{1}

6.8 Image

6.8.1 General

Type	Type of the element
Id	Id of the element
Alias	Unique widget identifier

6.8.2 Position

X position	Distance from the left edge in pixels
Y position	Distance from the top edge in pixels
Width	Width of the element in pixel, including border
Height	Height of the element in pixel, including border
Rotation	Rotation of the element about its center in degrees

6.8.3 Style

Fill color	Background fill color
Border color	The border color, it can only be selected if no theme is selected.
Border width	Border width in pixel
Transparency	A value between 0 and 100, 0: the object appears completely transparent, 100: the object completely covers the background
Shadow	Show a drop shadow

6.8.4 Text properties

Custom Filter	Name of filter function to format the displayed value.
Tooltip	Text that is displayed when the mouse pointer is over the item. Be aware that in case of a touch device the tooltip can't be displayed.

6.8.5 Dynamic fill color

Variable for Dynamic FillColor	The linked variable, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.BLINK globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive paramters is normaly '0'
No of Fill Color	Number of fill color for the widget. Minimum Value : 1 Maximum Value : 16 First match will be taken and rest is ignored. E.g. if "Operator" and "Statement" are matching for fill color 3, then the statements for all following defined fill colors (4,5,...) will be ignored. Hint: Only 'Range' operator (x to Y) is possible if number of fill colors is more then 1

6.8.6 Dynamic movements

Position X	Movement of widget along X - absolute: from left edge of the screen - relative: from the original placement of the widget
Position Y	Movement of widget along Y - absolute: from left edge of the screen - relative: from the original placement of the widget

6.8.7 Dynamic size

Variable width	Linked variable controls the width of the object, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0047, Position command value' with '0'= control Id, '22'= drive address: 0,22,S-0-0047 Hint: Control Id for drive parameters is normally '0'
Variable width script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".

Variable height	Linked variable controls the height of the object, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0047, Position command value' with '0'= control Id, '22'= drive address: 0,22,S-0-0047 Hint: Control Id for drive paramters is normaly '0'
Variable height script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".

6.8.8 Dynamic rotation

Rotation base	Specifies the point, where the rotation is based of
---------------	---

6.8.9 Events

OnClick	The event is executed if a full click is performed on the element, ie. a mouse press and a mouse release
OnMouseUp	The event is executed if a mouse button is released on this element
OnMouseDown	The event is executed if a mouse button is pressed on this element
OnMouseOver	The event is fires when when the mouse cursor enters the element
OnMouseOut	The event is fires when the mouse cursor leaves the element
OnRightClick	This event fires when the right mouse button is pressed
OnDoubleClick	This event fires on double mouse click

Attachment: Widget attribute

Push value	The linked variable, which is either entered or selected via the process variable browser, is set to TRUE (or FALSE) as long as the mouse is getting pressed. This is a convenience functionality and usually used with the "Button" widget and useful to set bool variables or single bits of an integer (use {bit} to set a single bit of integers). Depending on the 'Push value FALSE' attribute, either TRUE or FALSE is set when the button is pressed. Be aware that this feature used with active widgets (like "Input field" or "Checkbox") can cause problems when the same linked variable of the text properties is used because both try to write to the same variable independently! e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.myBool ns=2;s=Application.globalVAR.myInt{7}, with '{7}' = only this bit is toggled JavaScript variables: globalVariables.myBool screenVariables.myInt{14}, with '{14}' = only this bit is toggled Control-Parameter 'C-0-0080, User defined decimal number', with '{4}' = only this bit is used: C-0-0080{4} Axis-Parameter 'A-0-0048, Velocity polarities' with '0'= control Id, '3'= axis address, '{0}' = only this bit is toggled: 0,3,A-0-0048{0} Drive-Parameter 'P-0-0008, Activation E-Stop function' with '0'= control Id, '22'= drive address, '{1}' = only this bit is toggled: 0,22,P-0-0008{1} Hint: Control Id for drive parameters is usually '0'
------------	--

Push value FALSE	If set to 'true', the 'Push value' written to the linked variable as long as the mouse is getting pressed is FALSE. Attribute is 'false' (default) --> TRUE is written to the linked variable as long as the mouse is getting pressed Attribute is 'true' --> FALSE is written to the linked variable as long as the mouse is getting pressed
Toggle	The linked variable, which is either entered or selected via the process variable browser, is toggled between TRUE and FALSE when clicked. This is a convenience functionality and only useful to toggle bool variables or single bits of an integer (use {bit} to toggle a single bit of integers). Be aware that this feature used with active widgets (like "Input field" or "Checkbox") can cause problems when the same linked variable of the text properties is used because both try to write to the same variable independently! e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.myBool ns=2;s=Application.globalVAR.myInt{7}, with '{7}' = only this bit is toggled JavaScript variables: globalVariables.myBool screenVariables.myInt{14}, with '{14}' = only this bit is toggled Control-Parameter 'C-0-0080, User defined decimal number', with '{4}' = only this bit is used: C-0-0080{4} Axis-Parameter 'A-0-0048, Velocity polarities' with '0'= control Id, '3'= axis address, '{0}' = only this bit is toggled: 0,3,A-0-0048{0} Drive-Parameter 'P-0-0008, Activation E-Stop function' with '0'= control Id, '22'= drive address, '{1}' = only this bit is toggled: 0,22,P-0-0008{1}

6.9 Input field

6.9.1 General

Type	Type of the element
Id	Id of the element
Tab Id	Sets the tab order
Alias	Unique widget identifier

6.9.2 Position

X position	Distance from the left edge in pixels
Y position	Distance from the top edge in pixels
Width	Width of the element in pixel, including border
Height	Height of the element in pixel, including border
Rotation	Rotation of the element about its center in degrees

6.9.3 Style

Fill color	Background fill color
Border color	The border color, it can only be selected if no theme is selected.
Border width	Border width in pixel
Transparency	A value between 0 and 100, 0: the object appears completely transparent, 100: the object completely covers the background
Shadow	Show a drop shadow

6.9.4 Text properties

Text	Entered text will be shown. If a variable is entered in the 'Variable' attribute of the 'Text properties', the value of the linked variable will be shown respectively inserted if one of the below explained formatters is used inside of the text. printf formatters, %[width][.precision]specifier. Specifier: %b, yields an integer as a binary number %c, yields an integer as the character with that ASCII value %d or %i, yields an integer as a signed decimal number %e, yields a float using scientific notation %u, yields an integer as an unsigned decimal number %f, yields a float as is; see example for precision %o, yields an integer as an octal number %s, yields a string as is %x, yields an integer as a hexadecimal number (lower-case) %X, yields an integer as a hexadecimal number (upper-case) %j, yields a JavaScript object or array as a JSON encoded string e.g. The color: %s, value=blue => The color: blue First number: %d, value=12345 => First number: 12345 Second number: %04d, value=25 => Second number: 0025 Third number: %i, value=1234 => Third number: 1234 Float number: %3.2f, value=3.14159 => Float number: 3.14 Hexadecimal: %x, value=255 => Hexadecimal: ff Octal: %o, value=255 => Octal: 377 Unsigned value: %u, value=150 => Unsigned value: 150 html tags can also be used for formatting purposes bold text
 line break ordered list list item important text e.g This is normal text - and this is bold text, defines bold text areas My text in first line
 my text in second line, defines a single line break first line second line third line, defines an ordered list see also:
------	--

Attachment: Widget attribute

Variable	The linked variable, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.BLINK globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive paramters is normaly '0'
Font size	Font size can only be selected if no theme is selected.
Font Name	Font Name can only be selected if no theme is selected.
Custom Filter	Name of filter function to format the displayed value.
Variable Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
Tooltip	Text that is displayed when the mouse pointer is over the item. Be aware that in case of a touch device the tooltip can't be displayed.
Update interval	Length of the update rate The following update rates are available: 'none': Value is only read once when screen is opened 'slowest': 10000ms 'slow': 5000ms 'default': 1000ms 'fast': 500ms 'fastest': 200ms Be aware that a fast update rate increases the performance load of the application and the control.

6.9.5 Dynamic fill color

Variable for Dynamic FillColor	The linked variable, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.BLINK globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive paramters is normaly '0'
Variable Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be checked against the operators and statments of the defined fill colors. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
No of Fill Color	Number of fill color for the widget. Minimum Value : 1 Maximum Value : 16 First match will be taken and rest is ignored. E.g. if "Operator" and "Statement" are matching for fill color 3, then the statements for all following defined fill colors (4,5,...) will be ignored. Hint: Only 'Range' operator (x to Y) is possible if number of fill colors is more then 1

6.9.6 Dynamic movements

Position X	Movement of widget along X - absolute: from left edge of the screen - relative: from the original placement of the widget
Variable X	Linked variable controls the movement of the element in X-direction, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive parameters is normally '0'
Variable X Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
Position Y	Movement of widget along Y - absolute: from left edge of the screen - relative: from the original placement of the widget

Variable Y	Linked variable controls the movement of the element in Y-direction, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive paramters is normaly '0'
Variable Y Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".

6.9.7 Dynamic size

Variable width	Linked variable controls the width of the object, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0047, Position command value' with '0'= control Id, '22'= drive address: 0,22,S-0-0047 Hint: Control Id for drive paramters is normaly '0'
Variable width script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".

Variable height	Linked variable controls the height of the object, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0047, Position command value' with '0'= control Id, '22'= drive address: 0,22,S-0-0047 Hint: Control Id for drive paramters is normaly '0'
Variable height script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".

6.9.8 Dynamic rotation

Variable for Dynamic Rotation	Linked variable controls the rotation of the element, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive paramters is normaly '0'
Variable Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
Rotation base	Specifies the point, where the rotation is based of

6.9.9 Events

OnClick	The event is executed if a full click is performed on the element, ie. a mouse press and a mouse release
OnMouseUp	The event is executed if a mouse button is released on this element
OnMouseDown	The event is executed if a mouse button is pressed on this element
OnMouseOver	The event is fires when when the mouse cursor enters the element
OnMouseOut	The event is fires when the mouse cursor leaves the element
OnValueChange	The event is fires when the assigned value has been changed
OnRightClick	This event fires when the right mouse button is pressed
OnBlur	This event is raised when the element loses focus
OnDoubleClick	This event fires on double mouse click
OnFocus	This event fires when the element gets focus

Push value	The linked variable, which is either entered or selected via the process variable browser, is set to TRUE (or FALSE) as long as the mouse is getting pressed. This is a convenience functionality and usually used with the "Button" widget and useful to set bool variables or single bits of an integer (use {bit} to set a single bit of integers). Depending on the 'Push value FALSE' attribute, either TRUE or FALSE is set when the button is pressed. Be aware that this feature used with active widgets (like "Input field" or "Chechbox") can cause problems when the same linked variable of the text properties is used because both try to write to the same variable independently! e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.myBool ns=2;s=Application.globalVAR.myInt{7}, with '{7}' = only this bit is toggled JavaScript variables: globalVariables.myBool screenVariables.myInt{14}, with '{14}' = only this bit is toggled Control-Parameter 'C-0-0080, User defined decimal number', with '{4}' = only this bit is used: C-0-0080{4} Axis-Parameter 'A-0-0048, Velocity polarities' with '0'= control Id, '3'= axis address, '{0}' = only this bit is toggled: 0,3,A-0-0048{0} Drive-Parameter 'P-0-0008, Activation E-Stop function' with '0'= control Id, '22'= drive address, '{1}' = only this bit is toggled: 0,22,P-0-0008{1} Hint: Control Id for drive paramters is usually '0'
------------	---

Attachment: Widget attribute

Push value FALSE	If set to 'true', the 'Push value' written to the linked variable as long as the mouse is getting pressed is FALSE. Attribute is 'false' (default) --> TRUE is written to the linked variable as long as the mouse is getting pressed Attribute is 'true' --> FALSE is written to the linked variable as long as the mouse is getting pressed
Toggle	The linked variable, which is either entered or selected via the process variable browser, is toggled between TRUE and FALSE when clicked. This is a convenience functionality and only useful to toggle bool variables or single bits of an integer (use {bit} to toggle a single bit of integers). Be aware that this feature used with active widgets (like "Input field" or "Checkbox") can cause problems when the same linked variable of the text properties is used because both try to write to the same variable independently! e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.myBool ns=2;s=Application.globalVAR.myInt{7}, with '{7}' = only this bit is toggled JavaScript variables: globalVariables.myBool screenVariables.myInt{14}, with '{14}' = only this bit is toggled Control-Parameter 'C-0-0080, User defined decimal number', with '{4}' = only this bit is used: C-0-0080{4} Axis-Parameter 'A-0-0048, Velocity polarities' with '0'= control Id, '3'= axis address, '{0}' = only this bit is toggled: 0,3,A-0-0048{0} Drive-Parameter 'P-0-0008, Activation E-Stop function' with '0'= control Id, '22'= drive address, '{1}' = only this bit is toggled: 0,22,P-0-0008{1}

6.10 Input field extended

6.10.1 General

Type	Type of the element
Id	Id of the element
Tab Id	Sets the tab order
Alias	Unique widget identifier

6.10.2 Position

X position	Distance from the left edge in pixels
Y position	Distance from the top edge in pixels
Width	Width of the element in pixel, including border
Height	Height of the element in pixel, including border
Rotation	Rotation of the element about its center in degrees

6.10.3 Style

Fill color	Background fill color
Border color	The border color, it can only be selected if no theme is selected.
Border width	Border width in pixel
Transparency	A value between 0 and 100, 0: the object appears completely transparent, 100: the object completely covers the background
Shadow	Show a drop shadow

6.10.4 Text properties

Text	Entered text will be shown. If a variable is entered in the 'Variable' attribute of the 'Text properties', the value of the linked variable will be shown respectively inserted if one of the below explained formatters is used inside of the text. printf formatters, %[width][.precision]specifier. Specifier: %b, yields an integer as a binary number %c, yields an integer as the character with that ASCII value %d or %i, yields an integer as a signed decimal number %e, yields a float using scientific notation %u, yields an integer as an unsigned decimal number %f, yields a float as is; see example for precision %o, yields an integer as an octal number %s, yields a string as is %x, yields an integer as a hexadecimal number (lower-case) %X, yields an integer as a hexadecimal number (upper-case) %j, yields a JavaScript object or array as a JSON encoded string e.g. The color: %s, value=blue => The color: blue First number: %d, value=12345 => First number: 12345 Second number: %04d, value=25 => Second number: 0025 Third number: %i, value=1234 => Third number: 1234 Float number: %3.2f, value=3.14159 => Float number: 3.14 Hexadecimal: %x, value=255 => Hexadecimal: ff Octal: %o, value=255 => Octal: 377 Unsigned value: %u, value=150 => Unsigned value: 150 html tags can also be used for formatting purposes bold text
 line break ordered list list item important text e.g. This is normal text - and this is bold text, defines bold text areas My text in first line
 my text in second line, defines a single line break first line second line third line, defines an ordered list see also:
------	---

Variable	The linked variable, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.BLINK globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive parameters is normally '0'
Font color	The color of the font, it can only be selected if no theme is selected.
Font size	Font size can only be selected if no theme is selected.
Font Name	Font Name can only be selected if no theme is selected.
Custom Filter	Name of filter function to format the displayed value.
Variable Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
Tooltip	Text that is displayed when the mouse pointer is over the item. Be aware that in case of a touch device the tooltip can't be displayed.
Update interval	Length of the update rate The following update rates are available: 'none': Value is only read once when screen is opened 'slowest': 10000ms 'slow': 5000ms 'default': 1000ms 'fast': 500ms 'fastest': 200ms Be aware that a fast update rate increases the performance load of the application and the control.

6.10.5 Dynamic fill color

Variable for Dynamic FillColor	The linked variable, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.BLINK globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive paramters is normaly '0'
Variable Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be checked against the operators and statments of the defined fill colors. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
No of Fill Color	Number of fill color for the widget. Minimum Value : 1 Maximum Value : 16 First match will be taken and rest is ignored. E.g. if "Operator" and "Statement" are matching for fill color 3, then the statements for all following defined fill colors (4,5,...) will be ignored. Hint: Only 'Range' operator (x to Y) is possible if number of fill colors is more then 1

6.10.6 Dynamic movements

Position X	Movement of widget along X - absolute: from left edge of the screen - relative: from the original placement of the widget
Variable X	Linked variable controls the movement of the element in X-direction, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive parameters is normally '0'
Variable X Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
Position Y	Movement of widget along Y - absolute: from left edge of the screen - relative: from the original placement of the widget

Attachment: Widget attribute

Variable Y	Linked variable controls the movement of the element in Y-direction, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive paramters is normaly '0'
Variable Y Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".

6.10.7 Dynamic size

Variable width	Linked variable controls the width of the object, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0047, Position command value' with '0'= control Id, '22'= drive address: 0,22,S-0-0047 Hint: Control Id for drive paramters is normaly '0'
Variable width script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".

Attachment: Widget attribute

Variable height	Linked variable controls the height of the object, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0047, Position command value' with '0'= control Id, '22'= drive address: 0,22,S-0-0047 Hint: Control Id for drive paramters is normaly '0'
Variable height script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".

6.10.8 Dynamic rotation

Variable for Dynamic Rotation	Linked variable controls the rotation of the element, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive paramters is normaly '0'
Variable Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
Rotation base	Specifies the point, where the rotation is based of

6.10.9 Events

OnClick	The event is executed if a full click is performed on the element, ie. a mouse press and a mouse release
OnMouseUp	The event is executed if a mouse button is released on this element
OnMouseDown	The event is executed if a mouse button is pressed on this element
OnMouseOver	The event is fires when when the mouse cursor enters the element
OnMouseOut	The event is fires when the mouse cursor leaves the element
OnValueChange	The event is fires when the assigned value has been changed
OnRightClick	This event fires when the right mouse button is pressed
OnBlur	This event is raised when the element loses focus
OnDoubleClick	This event fires on double mouse click
OnFocus	This event fires when the element gets focus

Attachment: Widget attribute

Push value	The linked variable, which is either entered or selected via the process variable browser, is set to TRUE (or FALSE) as long as the mouse is getting pressed. This is a convenience functionality and usually used with the "Button" widget and useful to set bool variables or single bits of an integer (use {bit} to set a single bit of integers). Depending on the 'Push value FALSE' attribute, either TRUE or FALSE is set when the button is pressed. Be aware that this feature used with active widgets (like "Input field" or "Checkbox") can cause problems when the same linked variable of the text properties is used because both try to write to the same variable independently! e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.myBool ns=2;s=Application.globalVAR.myInt{7}, with '{7}' = only this bit is toggled JavaScript variables: globalVariables.myBool screenVariables.myInt{14}, with '{14}' = only this bit is toggled Control-Parameter 'C-0-0080, User defined decimal number', with '{4}' = only this bit is used: C-0-0080{4} Axis-Parameter 'A-0-0048, Velocity polarities' with '0'= control Id, '3'= axis address, '{0}' = only this bit is toggled: 0,3,A-0-0048{0} Drive-Parameter 'P-0-0008, Activation E-Stop function' with '0'= control Id, '22'= drive address, '{1}' = only this bit is toggled: 0,22,P-0-0008{1} Hint: Control Id for drive parameters is usually '0'
------------	--

Push value FALSE	If set to 'true', the 'Push value' written to the linked variable as long as the mouse is getting pressed is FALSE. Attribute is 'false' (default) --> TRUE is written to the linked variable as long as the mouse is getting pressed Attribute is 'true' --> FALSE is written to the linked variable as long as the mouse is getting pressed
Toggle	The linked variable, which is either entered or selected via the process variable browser, is toggled between TRUE and FALSE when clicked. This is a convenience functionality and only useful to toggle bool variables or single bits of an integer (use {bit} to toggle a single bit of integers). Be aware that this feature used with active widgets (like "Input field" or "Checkbox") can cause problems when the same linked variable of the text properties is used because both try to write to the same variable independently! e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.myBool ns=2;s=Application.globalVAR.myInt{7}, with '{7}' = only this bit is toggled JavaScript variables: globalVariables.myBool screenVariables.myInt{14}, with '{14}' = only this bit is toggled Control-Parameter 'C-0-0080, User defined decimal number', with '{4}' = only this bit is used: C-0-0080{4} Axis-Parameter 'A-0-0048, Velocity polarities' with '0'= control Id, '3'= axis address, '{0}' = only this bit is toggled: 0,3,A-0-0048{0} Drive-Parameter 'P-0-0008, Activation E-Stop function' with '0'= control Id, '22'= drive address, '{1}' = only this bit is toggled: 0,22,P-0-0008{1}

6.11 Line

6.11.1 General

Type	Type of the element
Id	Id of the element
Alias	Unique widget identifier

6.11.2 Position

X position	Distance from the left edge in pixels
Y position	Distance from the top edge in pixels
Width	Width of the element in pixel, including border
Height	Height of the element in pixel, including border
Rotation	Rotation of the element about its center in degrees

6.11.3 Style

Fill color	Background fill color
Border color	The border color, it can only be selected if no theme is selected.
Border width	Border width in pixel
Transparency	A value between 0 and 100, 0: the object appears completely transparent, 100: the object completely covers the background
Shadow	Show a drop shadow

6.11.4 Text properties

Custom Filter	Name of filter function to format the displayed value.
Variable Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
Tooltip	Text that is displayed when the mouse pointer is over the item. Be aware that in case of a touch device the tooltip can't be displayed.

6.11.5 Dynamic fill color

Variable for Dynamic FillColor	The linked variable, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.BLINK globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive paramters is normaly '0'
Variable Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be checked against the operators and statments of the defined fill colors. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
No of Fill Color	Number of fill color for the widget. Minimum Value : 1 Maximum Value : 16 First match will be taken and rest is ignored. E.g. if "Operator" and "Statement" are matching for fill color 3, then the statements for all following defined fill colors (4,5,...) will be ignored. Hint: Only 'Range' operator (x to Y) is possible if number of fill colors is more then 1

6.11.6 Dynamic movements

Position X	Movement of widget along X - absolute: from left edge of the screen - relative: from the original placement of the widget
Variable X Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
Position Y	Movement of widget along Y - absolute: from left edge of the screen - relative: from the original placement of the widget
Variable Y Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".

6.11.7 Dynamic size

Variable width	Linked variable controls the width of the object, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0047, Position command value' with '0'= control Id, '22'= drive address: 0,22,S-0-0047 Hint: Control Id for drive paramters is normaly '0'
Variable width script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".

Attachment: Widget attribute

Variable height	Linked variable controls the height of the object, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0047, Position command value' with '0'= control Id, '22'= drive address: 0,22,S-0-0047 Hint: Control Id for drive paramters is normaly '0'
Variable height script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".

6.11.8 Dynamic rotation

Variable Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
Rotation base	Specifies the point, where the rotation is based of

6.11.9 Events

OnClick	The event is executed if a full click is performed on the element, ie. a mouse press and a mouse release
OnMouseUp	The event is executed if a mouse button is released on this element
OnMouseDown	The event is executed if a mouse button is pressed on this element
OnMouseOver	The event is fires when when the mouse cursor enters the element
OnMouseOut	The event is fires when the mouse cursor leaves the element

OnRightClick	This event fires when the right mouse button is pressed
OnDoubleClick	This event fires on double mouse click
Push value	The linked variable, which is either entered or selected via the process variable browser, is set to TRUE (or FALSE) as long as the mouse is getting pressed. This is a convenience functionality and usually used with the "Button" widget and useful to set bool variables or single bits of an integer (use {bit} to set a single bit of integers). Depending on the 'Push value FALSE' attribute, either TRUE or FALSE is set when the button is pressed. Be aware that this feature used with active widgets (like "Input field" or "Checkbox") can cause problems when the same linked variable of the text properties is used because both try to write to the same variable independently! e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.myBool ns=2;s=Application.globalVAR.myInt{7}, with '{7}' = only this bit is toggled JavaScript variables: globalVariables.myBool screenVariables.myInt{14}, with '{14}' = only this bit is toggled Control-Parameter 'C-0-0080, User defined decimal number', with '{4}' = only this bit is used: C-0-0080{4} Axis-Parameter 'A-0-0048, Velocity polarities' with '0'= control Id, '3'= axis address, '{0}' = only this bit is toggled: 0,3,A-0-0048{0} Drive-Parameter 'P-0-0008, Activation E-Stop function' with '0'= control Id, '22'= drive address, '{1}' = only this bit is toggled: 0,22,P-0-0008{1} Hint: Control Id for drive parameters is usually '0'

Attachment: Widget attribute

Push value FALSE	If set to 'true', the 'Push value' written to the linked variable as long as the mouse is getting pressed is FALSE. Attribute is 'false' (default) --> TRUE is written to the linked variable as long as the mouse is getting pressed Attribute is 'true' --> FALSE is written to the linked variable as long as the mouse is getting pressed
Toggle	The linked variable, which is either entered or selected via the process variable browser, is toggled between TRUE and FALSE when clicked. This is a convenience functionality and only useful to toggle bool variables or single bits of an integer (use {bit} to toggle a single bit of integers). Be aware that this feature used with active widgets (like "Input field" or "Checkbox") can cause problems when the same linked variable of the text properties is used because both try to write to the same variable independently! e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.myBool ns=2;s=Application.globalVAR.myInt{7}, with '{7}' = only this bit is toggled JavaScript variables: globalVariables.myBool screenVariables.myInt{14}, with '{14}' = only this bit is toggled Control-Parameter 'C-0-0080, User defined decimal number', with '{4}' = only this bit is used: C-0-0080{4} Axis-Parameter 'A-0-0048, Velocity polarities' with '0'= control Id, '3'= axis address, '{0}' = only this bit is toggled: 0,3,A-0-0048{0} Drive-Parameter 'P-0-0008, Activation E-Stop function' with '0'= control Id, '22'= drive address, '{1}' = only this bit is toggled: 0,22,P-0-0008{1}

6.12 Polygon

6.12.1 General

Type	Type of the element
Id	Id of the element
Alias	Unique widget identifier

6.12.2 Position

X position	Distance from the left edge in pixels
Y position	Distance from the top edge in pixels
Width	Width of the element in pixel, including border
Height	Height of the element in pixel, including border
Rotation	Rotation of the element about its center in degrees

6.12.3 Style

Fill color	Background fill color
Border color	The border color, it can only be selected if no theme is selected.
Border width	Border width in pixel
Transparency	A value between 0 and 100, 0: the object appears completely transparent, 100: the object completely covers the background
Shadow	Show a drop shadow

6.12.4 Text properties

Custom Filter	Name of filter function to format the displayed value.
Variable Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
Tooltip	Text that is displayed when the mouse pointer is over the item. Be aware that in case of a touch device the tooltip can't be displayed.

6.12.5 Dynamic fill color

Variable for Dynamic FillColor	The linked variable, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.BLINK globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive paramters is normaly '0'
Variable Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be checked against the operators and statments of the defined fill colors. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
No of Fill Color	Number of fill color for the widget. Minimum Value : 1 Maximum Value : 16 First match will be taken and rest is ignored. E.g. if "Operator" and "Statement" are matching for fill color 3, then the statements for all following defined fill colors (4,5,...) will be ignored. Hint: Only 'Range' operator (x to Y) is possible if number of fill colors is more then 1

6.12.6 Dynamic movements

Position X	Movement of widget along X - absolute: from left edge of the screen - relative: from the original placement of the widget
Variable X Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
Position Y	Movement of widget along Y - absolute: from left edge of the screen - relative: from the original placement of the widget
Variable Y Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".

6.12.7 Dynamic size

Variable width	Linked variable controls the width of the object, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0047, Position command value' with '0'= control Id, '22'= drive address: 0,22,S-0-0047 Hint: Control Id for drive paramters is normaly '0'
Variable width script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".

Variable height	Linked variable controls the height of the object, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0047, Position command value' with '0'= control Id, '22'= drive address: 0,22,S-0-0047 Hint: Control Id for drive paramters is normaly '0'
Variable height script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".

6.12.8 Dynamic rotation

Variable Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
Rotation base	Specifies the point, where the rotation is based of

6.12.9 Events

OnClick	The event is executed if a full click is performed on the element, ie. a mouse press and a mouse release
OnMouseUp	The event is executed if a mouse button is released on this element
OnMouseDown	The event is executed if a mouse button is pressed on this element
OnMouseOver	The event is fires when when the mouse cursor enters the element
OnMouseOut	The event is fires when the mouse cursor leaves the element

Attachment: Widget attribute

OnRightClick	This event fires when the right mouse button is pressed
OnDoubleClick	This event fires on double mouse click
Push value	The linked variable, which is either entered or selected via the process variable browser, is set to TRUE (or FALSE) as long as the mouse is getting pressed. This is a convenience functionality and usually used with the "Button" widget and useful to set bool variables or single bits of an integer (use {bit} to set a single bit of integers). Depending on the 'Push value FALSE' attribute, either TRUE or FALSE is set when the button is pressed. Be aware that this feature used with active widgets (like "Input field" or "Chechbox") can cause problems when the same linked variable of the text properties is used because both try to write to the same variable independently! e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.myBool ns=2;s=Application.globalVAR.myInt{7}, with '{7}' = only this bit is toggled JavaScript variables: globalVariables.myBool screenVariables.myInt{14}, with '{14}' = only this bit is toggled Control-Parameter 'C-0-0080, User defined decimal number', with '{4}' = only this bit is used: C-0-0080{4} Axis-Parameter 'A-0-0048, Velocity polarities' with '0'= control Id, '3'= axis address, '{0}' = only this bit is toggled: 0,3,A-0-0048{0} Drive-Parameter 'P-0-0008, Activation E-Stop function' with '0'= control Id, '22'= drive address, '{1}' = only this bit is toggled: 0,22,P-0-0008{1} Hint: Control Id for drive paramters is usually '0'

Push value FALSE	If set to 'true', the 'Push value' written to the linked variable as long as the mouse is getting pressed is FALSE. Attribute is 'false' (default) --> TRUE is written to the linked variable as long as the mouse is getting pressed Attribute is 'true' --> FALSE is written to the linked variable as long as the mouse is getting pressed
Toggle	The linked variable, which is either entered or selected via the process variable browser, is toggled between TRUE and FALSE when clicked. This is a convenience functionality and only useful to toggle bool variables or single bits of an integer (use {bit} to toggle a single bit of integers). Be aware that this feature used with active widgets (like "Input field" or "Checkbox") can cause problems when the same linked variable of the text properties is used because both try to write to the same variable independently! e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.myBool ns=2;s=Application.globalVAR.myInt{7}, with '{7}' = only this bit is toggled JavaScript variables: globalVariables.myBool screenVariables.myInt{14}, with '{14}' = only this bit is toggled Control-Parameter 'C-0-0080, User defined decimal number', with '{4}' = only this bit is used: C-0-0080{4} Axis-Parameter 'A-0-0048, Velocity polarities' with '0'= control Id, '3'= axis address, '{0}' = only this bit is toggled: 0,3,A-0-0048{0} Drive-Parameter 'P-0-0008, Activation E-Stop function' with '0'= control Id, '22'= drive address, '{1}' = only this bit is toggled: 0,22,P-0-0008{1}

6.13 Polyline

6.13.1 General

Type	Type of the element
Id	Id of the element
Alias	Unique widget identifier

6.13.2 Position

X position	Distance from the left edge in pixels
Y position	Distance from the top edge in pixels
Width	Width of the element in pixel, including border
Height	Height of the element in pixel, including border
Rotation	Rotation of the element about its center in degrees

6.13.3 Style

Fill color	Background fill color
Transparency	A value between 0 and 100, 0: the object appears completely transparent, 100: the object completely covers the background
Shadow	Show a drop shadow

6.13.4 Text properties

Custom Filter	Name of filter function to format the displayed value.
Variable Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
Tooltip	Text that is displayed when the mouse pointer is over the item. Be aware that in case of a touch device the tooltip can't be displayed.

6.13.5 Dynamic fill color

Variable for Dynamic FillColor	The linked variable, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.BLINK globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive paramters is normaly '0'
Variable Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be checked against the operators and statments of the defined fill colors. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
No of Fill Color	Number of fill color for the widget. Minimum Value : 1 Maximum Value : 16 First match will be taken and rest is ignored. E.g. if "Operator" and "Statement" are matching for fill color 3, then the statements for all following defined fill colors (4,5,...) will be ignored. Hint: Only 'Range' operator (x to Y) is possible if number of fill colors is more then 1

6.13.6 Dynamic movements

Position X	Movement of widget along X - absolute: from left edge of the screen - relative: from the original placement of the widget
Variable X Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
Position Y	Movement of widget along Y - absolute: from left edge of the screen - relative: from the original placement of the widget
Variable Y Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".

6.13.7 Dynamic size

Variable width	Linked variable controls the width of the object, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0047, Position command value' with '0'= control Id, '22'= drive address: 0,22,S-0-0047 Hint: Control Id for drive paramters is normaly '0'
Variable width script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".

Attachment: Widget attribute

Variable height	Linked variable controls the height of the object, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0047, Position command value' with '0'= control Id, '22'= drive address: 0,22,S-0-0047 Hint: Control Id for drive paramters is normaly '0'
Variable height script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".

6.13.8 Dynamic rotation

Variable Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
Rotation base	Specifies the point, where the rotation is based of

6.13.9 Events

OnClick	The event is executed if a full click is performed on the element, ie. a mouse press and a mouse release
OnMouseUp	The event is executed if a mouse button is released on this element
OnMouseDown	The event is executed if a mouse button is pressed on this element
OnMouseOver	The event is fires when when the mouse cursor enters the element
OnMouseOut	The event is fires when the mouse cursor leaves the element

OnRightClick	This event fires when the right mouse button is pressed
OnDoubleClick	This event fires on double mouse click
Push value	The linked variable, which is either entered or selected via the process variable browser, is set to TRUE (or FALSE) as long as the mouse is getting pressed. This is a convenience functionality and usually used with the "Button" widget and useful to set bool variables or single bits of an integer (use {bit} to set a single bit of integers). Depending on the 'Push value FALSE' attribute, either TRUE or FALSE is set when the button is pressed. Be aware that this feature used with active widgets (like "Input field" or "Checkbox") can cause problems when the same linked variable of the text properties is used because both try to write to the same variable independently! e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.myBool ns=2;s=Application.globalVAR.myInt{7}, with '{7}' = only this bit is toggled JavaScript variables: globalVariables.myBool screenVariables.myInt{14}, with '{14}' = only this bit is toggled Control-Parameter 'C-0-0080, User defined decimal number', with '{4}' = only this bit is used: C-0-0080{4} Axis-Parameter 'A-0-0048, Velocity polarities' with '0'= control Id, '3'= axis address, '{0}' = only this bit is toggled: 0,3,A-0-0048{0} Drive-Parameter 'P-0-0008, Activation E-Stop function' with '0'= control Id, '22'= drive address, '{1}' = only this bit is toggled: 0,22,P-0-0008{1} Hint: Control Id for drive parameters is usually '0'

Attachment: Widget attribute

Push value FALSE	If set to 'true', the 'Push value' written to the linked variable as long as the mouse is getting pressed is FALSE. Attribute is 'false' (default) --> TRUE is written to the linked variable as long as the mouse is getting pressed Attribute is 'true' --> FALSE is written to the linked variable as long as the mouse is getting pressed
Toggle	The linked variable, which is either entered or selected via the process variable browser, is toggled between TRUE and FALSE when clicked. This is a convenience functionality and only useful to toggle bool variables or single bits of an integer (use {bit} to toggle a single bit of integers). Be aware that this feature used with active widgets (like "Input field" or "Checkbox") can cause problems when the same linked variable of the text properties is used because both try to write to the same variable independently! e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.myBool ns=2;s=Application.globalVAR.myInt{7}, with '{7}' = only this bit is toggled JavaScript variables: globalVariables.myBool screenVariables.myInt{14}, with '{14}' = only this bit is toggled Control-Parameter 'C-0-0080, User defined decimal number', with '{4}' = only this bit is used: C-0-0080{4} Axis-Parameter 'A-0-0048, Velocity polarities' with '0'= control Id, '3'= axis address, '{0}' = only this bit is toggled: 0,3,A-0-0048{0} Drive-Parameter 'P-0-0008, Activation E-Stop function' with '0'= control Id, '22'= drive address, '{1}' = only this bit is toggled: 0,22,P-0-0008{1}

6.14 Progress bar

6.14.1 General

Type	Type of the element
Id	Id of the element
Alias	Unique widget identifier

6.14.2 Position

X position	Distance from the left edge in pixels
Y position	Distance from the top edge in pixels
Width	Width of the element in pixel, including border
Height	Height of the element in pixel, including border
Rotation	Rotation of the element about its center in degrees

6.14.3 Style

Fill color	Background fill color
Border color	The border color, it can only be selected if no theme is selected.
Border width	Border width in pixel
Transparency	A value between 0 and 100, 0: the object appears completely transparent, 100: the object completely covers the background
Shadow	Show a drop shadow

6.14.4 Text properties

Text	Entered text will be shown. If a variable is entered in the 'Variable' attribute of the 'Text properties', the value of the linked variable will be shown respectively inserted if one of the below explained formatters is used inside of the text. printf formatters, %[width][.precision]specifier. Specifier: %b, yields an integer as a binary number %c, yields an integer as the character with that ASCII value %d or %i, yields an integer as a signed decimal number %e, yields a float using scientific notation %u, yields an integer as an unsigned decimal number %f, yields a float as is; see example for precision %o, yields an integer as an octal number %s, yields a string as is %x, yields an integer as a hexadecimal number (lower-case) %X, yields an integer as a hexadecimal number (upper-case) %j, yields a JavaScript object or array as a JSON encoded string e.g. The color: %s, value=blue => The color: blue First number: %d, value=12345 => First number: 12345 Second number: %04d, value=25 => Second number: 0025 Third number: %i, value=1234 => Third number: 1234 Float number: %3.2f, value=3.14159 => Float number: 3.14 Hexadecimal: %x, value=255 => Hexadecimal: ff Octal: %o, value=255 => Octal: 377 Unsigned value: %u, value=150 => Unsigned value: 150 html tags can also be used for formatting purposes bold text
 line break ordered list list item important text e.g. This is normal text - and this is bold text, defines bold text areas My text in first line
 my text in second line, defines a single line break first line second line third line, defines an ordered list see also:
------	---

Variable	The linked variable, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.BLINK globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive paramters is normaly '0'
Font color	The color of the font, it can only be selected if no theme is selected.
Font size	Font size can only be selected if no theme is selected.
Font Name	Font Name can only be selected if no theme is selected.
Custom Filter	Name of filter function to format the displayed value.
Variable Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
Tooltip	Text that is displayed when the mouse pointer is over the item. Be aware that in case of a touch device the tooltip can't be displayed.
Update interval	Length of the update rate The following update rates are available: 'none': Value is only read once when screen is opened 'slowest': 10000ms 'slow': 5000ms 'default': 1000ms 'fast': 500ms 'fastest': 200ms Be aware that a fast update rate increases the performance load of the application and the control.

6.14.5 Dynamic fill color

Variable for Dynamic FillColor	The linked variable, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.BLINK globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive paramters is normaly '0'
Variable Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be checked against the operators and statments of the defined fill colors. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
No of Fill Color	Number of fill color for the widget. Minimum Value : 1 Maximum Value : 16 First match will be taken and rest is ignored. E.g. if "Operator" and "Statement" are matching for fill color 3, then the statements for all following defined fill colors (4,5,...) will be ignored. Hint: Only 'Range' operator (x to Y) is possible if number of fill colors is more then 1

6.14.6 Dynamic movements

Position X	Movement of widget along X - absolute: from left edge of the screen - relative: from the original placement of the widget
Variable X	Linked variable controls the movement of the element in X-direction, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive parameters is normally '0'
Variable X Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
Position Y	Movement of widget along Y - absolute: from left edge of the screen - relative: from the original placement of the widget

Attachment: Widget attribute

Variable Y	Linked variable controls the movement of the element in Y-direction, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive paramters is normaly '0'
Variable Y Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".

6.14.7 Dynamic size

Variable width	Linked variable controls the width of the object, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0047, Position command value' with '0'= control Id, '22'= drive address: 0,22,S-0-0047 Hint: Control Id for drive paramters is normaly '0'
Variable width script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".

Attachment: Widget attribute

Variable height	Linked variable controls the height of the object, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0047, Position command value' with '0'= control Id, '22'= drive address: 0,22,S-0-0047 Hint: Control Id for drive paramters is normaly '0'
Variable height script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".

6.14.8 Dynamic rotation

Variable for Dynamic Rotation	Linked variable controls the rotation of the element, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive paramters is normaly '0'
Variable Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
Rotation base	Specifies the point, where the rotation is based of

6.14.9 Events

OnMouseClicked	The event is executed if a full click is performed on the element, ie. a mouse press and a mouse release
OnMouseUp	The event is executed if a mouse button is released on this element
OnMouseDown	The event is executed if a mouse button is pressed on this element
OnMouseOver	The event is fires when when the mouse cursor enters the element
OnMouseOut	The event is fires when the mouse cursor leaves the element
OnRightClick	This event fires when the right mouse button is pressed
OnDoubleClick	This event fires on double mouse click

Attachment: Widget attribute

Push value	The linked variable, which is either entered or selected via the process variable browser, is set to TRUE (or FALSE) as long as the mouse is getting pressed. This is a convenience functionality and usually used with the "Button" widget and useful to set bool variables or single bits of an integer (use {bit} to set a single bit of integers). Depending on the 'Push value FALSE' attribute, either TRUE or FALSE is set when the button is pressed. Be aware that this feature used with active widgets (like "Input field" or "Checkbox") can cause problems when the same linked variable of the text properties is used because both try to write to the same variable independently! e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.myBool ns=2;s=Application.globalVAR.myInt{7}, with '{7}' = only this bit is toggled JavaScript variables: globalVariables.myBool screenVariables.myInt{14}, with '{14}' = only this bit is toggled Control-Parameter 'C-0-0080, User defined decimal number', with '{4}' = only this bit is used: C-0-0080{4} Axis-Parameter 'A-0-0048, Velocity polarities' with '0'= control Id, '3'= axis address, '{0}' = only this bit is toggled: 0,3,A-0-0048{0} Drive-Parameter 'P-0-0008, Activation E-Stop function' with '0'= control Id, '22'= drive address, '{1}' = only this bit is toggled: 0,22,P-0-0008{1} Hint: Control Id for drive parameters is usually '0'
------------	--

Push value FALSE	If set to 'true', the 'Push value' written to the linked variable as long as the mouse is getting pressed is FALSE. Attribute is 'false' (default) --> TRUE is written to the linked variable as long as the mouse is getting pressed Attribute is 'true' --> FALSE is written to the linked variable as long as the mouse is getting pressed
Toggle	The linked variable, which is either entered or selected via the process variable browser, is toggled between TRUE and FALSE when clicked. This is a convenience functionality and only useful to toggle bool variables or single bits of an integer (use {bit} to toggle a single bit of integers). Be aware that this feature used with active widgets (like "Input field" or "Checkbox") can cause problems when the same linked variable of the text properties is used because both try to write to the same variable independently! e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.myBool ns=2;s=Application.globalVAR.myInt{7}, with '{7}' = only this bit is toggled JavaScript variables: globalVariables.myBool screenVariables.myInt{14}, with '{14}' = only this bit is toggled Control-Parameter 'C-0-0080, User defined decimal number', with '{4}' = only this bit is used: C-0-0080{4} Axis-Parameter 'A-0-0048, Velocity polarities' with '0'= control Id, '3'= axis address, '{0}' = only this bit is toggled: 0,3,A-0-0048{0} Drive-Parameter 'P-0-0008, Activation E-Stop function' with '0'= control Id, '22'= drive address, '{1}' = only this bit is toggled: 0,22,P-0-0008{1}

6.15 Radial gauge

6.15.1 General

Type	Type of the element
Id	Id of the element
Alias	Unique widget identifier

6.15.2 Position

X position	Distance from the left edge in pixels
Y position	Distance from the top edge in pixels
Width	Width of the element in pixel, including border
Height	Height of the element in pixel, including border
Rotation	Rotation of the element about its center in degrees

6.15.3 Style

Fill color	Background fill color
Border color	The border color, it can only be selected if no theme is selected.
Border width	Border width in pixel
Transparency	A value between 0 and 100, 0: the object appears completely transparent, 100: the object completely covers the background
Shadow	Show a drop shadow

6.15.4 Text properties

Text	Entered text will be shown. If a variable is entered in the 'Variable' attribute of the 'Text properties', the value of the linked variable will be shown respectively inserted if one of the below explained formatters is used inside of the text. printf formatters, %[width][.precision]specifier. Specifier: %b, yields an integer as a binary number %c, yields an integer as the character with that ASCII value %d or %i, yields an integer as a signed decimal number %e, yields a float using scientific notation %u, yields an integer as an unsigned decimal number %f, yields a float as is; see example for precision %o, yields an integer as an octal number %s, yields a string as is %x, yields an integer as a hexadecimal number (lower-case) %X, yields an integer as a hexadecimal number (upper-case) %j, yields a JavaScript object or array as a JSON encoded string e.g. The color: %s, value=blue => The color: blue First number: %d, value=12345 => First number: 12345 Second number: %04d, value=25 => Second number: 0025 Third number: %i, value=1234 => Third number: 1234 Float number: %3.2f, value=3.14159 => Float number: 3.14 Hexadecimal: %x, value=255 => Hexadecimal: ff Octal: %o, value=255 => Octal: 377 Unsigned value: %u, value=150 => Unsigned value: 150 html tags can also be used for formatting purposes bold text
 line break ordered list list item important text e.g This is normal text - and this is bold text, defines bold text areas My text in first line
 my text in second line, defines a single line break first line second line third line, defines an ordered list see also:
------	--

Attachment: Widget attribute

Variable	The linked variable, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.BLINK globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive paramters is normaly '0'
Font color	The color of the font, it can only be selected if no theme is selected.
Font Name	Font Name can only be selected if no theme is selected.
Custom Filter	Name of filter function to format the displayed value.
Variable Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
Tooltip	Text that is displayed when the mouse pointer is over the item. Be aware that in case of a touch device the tooltip can't be displayed.
Update interval	Length of the update rate The following update rates are available: 'none': Value is only read once when screen is opened 'slowest': 10000ms 'slow': 5000ms 'default': 1000ms 'fast': 500ms 'fastest': 200ms Be aware that a fast update rate increases the performance load of the application and the control.

6.15.5 Dynamic fill color

Variable for Dynamic FillColor	The linked variable, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.BLINK globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive paramters is normaly '0'
Variable Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be checked against the operators and statments of the defined fill colors. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
No of Fill Color	Number of fill color for the widget. Minimum Value : 1 Maximum Value : 16 First match will be taken and rest is ignored. E.g. if "Operator" and "Statement" are matching for fill color 3, then the statements for all following defined fill colors (4,5,...) will be ignored. Hint: Only 'Range' operator (x to Y) is possible if number of fill colors is more then 1

6.15.6 Dynamic movements

Position X	Movement of widget along X - absolute: from left edge of the screen - relative: from the original placement of the widget
Variable X	Linked variable controls the movement of the element in X-direction, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive parameters is normally '0'
Variable X Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
Position Y	Movement of widget along Y - absolute: from left edge of the screen - relative: from the original placement of the widget

Variable Y	Linked variable controls the movement of the element in Y-direction, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive paramters is normaly '0'
Variable Y Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".

6.15.7 Dynamic size

Variable width	Linked variable controls the width of the object, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0047, Position command value' with '0'= control Id, '22'= drive address: 0,22,S-0-0047 Hint: Control Id for drive paramters is normaly '0'
Variable width script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".

Variable height	Linked variable controls the height of the object, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0047, Position command value' with '0'= control Id, '22'= drive address: 0,22,S-0-0047 Hint: Control Id for drive paramters is normaly '0'
Variable height script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".

6.15.8 Dynamic rotation

Variable for Dynamic Rotation	Linked variable controls the rotation of the element, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive paramters is normaly '0'
Variable Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
Rotation base	Specifies the point, where the rotation is based of

6.15.9 Events

OnClick	The event is executed if a full click is performed on the element, ie. a mouse press and a mouse release
OnMouseUp	The event is executed if a mouse button is released on this element
OnMouseDown	The event is executed if a mouse button is pressed on this element
OnMouseOver	The event is fires when when the mouse cursor enters the element
OnMouseOut	The event is fires when the mouse cursor leaves the element
OnRightClick	This event fires when the right mouse button is pressed
OnDoubleClick	This event fires on double mouse click

Push value	The linked variable, which is either entered or selected via the process variable browser, is set to TRUE (or FALSE) as long as the mouse is getting pressed. This is a convenience functionality and usually used with the "Button" widget and useful to set bool variables or single bits of an integer (use {bit} to set a single bit of integers). Depending on the 'Push value FALSE' attribute, either TRUE or FALSE is set when the button is pressed. Be aware that this feature used with active widgets (like "Input field" or "Chechbox") can cause problems when the same linked variable of the text properties is used because both try to write to the same variable independently! e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.myBool ns=2;s=Application.globalVAR.myInt{7}, with '{7}' = only this bit is toggled JavaScript variables: globalVariables.myBool screenVariables.myInt{14}, with '{14}' = only this bit is toggled Control-Parameter 'C-0-0080, User defined decimal number', with '{4}' = only this bit is used: C-0-0080{4} Axis-Parameter 'A-0-0048, Velocity polarities' with '0'= control Id, '3'= axis address, '{0}' = only this bit is toggled: 0,3,A-0-0048{0} Drive-Parameter 'P-0-0008, Activation E-Stop function' with '0'= control Id, '22'= drive address, '{1}' = only this bit is toggled: 0,22,P-0-0008{1} Hint: Control Id for drive paramters is usually '0'
------------	---

Attachment: Widget attribute

Push value FALSE	If set to 'true', the 'Push value' written to the linked variable as long as the mouse is getting pressed is FALSE. Attribute is 'false' (default) --> TRUE is written to the linked variable as long as the mouse is getting pressed Attribute is 'true' --> FALSE is written to the linked variable as long as the mouse is getting pressed
Toggle	The linked variable, which is either entered or selected via the process variable browser, is toggled between TRUE and FALSE when clicked. This is a convenience functionality and only useful to toggle bool variables or single bits of an integer (use {bit} to toggle a single bit of integers). Be aware that this feature used with active widgets (like "Input field" or "Checkbox") can cause problems when the same linked variable of the text properties is used because both try to write to the same variable independently! e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.myBool ns=2;s=Application.globalVAR.myInt{7}, with '{7}' = only this bit is toggled JavaScript variables: globalVariables.myBool screenVariables.myInt{14}, with '{14}' = only this bit is toggled Control-Parameter 'C-0-0080, User defined decimal number', with '{4}' = only this bit is used: C-0-0080{4} Axis-Parameter 'A-0-0048, Velocity polarities' with '0'= control Id, '3'= axis address, '{0}' = only this bit is toggled: 0,3,A-0-0048{0} Drive-Parameter 'P-0-0008, Activation E-Stop function' with '0'= control Id, '22'= drive address, '{1}' = only this bit is toggled: 0,22,P-0-0008{1}

6.16 Radial progress bar

6.16.1 General

Type	Type of the element
Id	Id of the element
Alias	Unique widget identifier

6.16.2 Position

X position	Distance from the left edge in pixels
Y position	Distance from the top edge in pixels
Width	Width of the element in pixel, including border
Height	Height of the element in pixel, including border
Rotation	Rotation of the element about its center in degrees

6.16.3 Style

Fill color	Background fill color
Border color	The border color, it can only be selected if no theme is selected.
Border width	Border width in pixel
Transparency	A value between 0 and 100, 0: the object appears completely transparent, 100: the object completely covers the background
Shadow	Show a drop shadow

6.16.4 Text properties

Text	Entered text will be shown. If a variable is entered in the 'Variable' attribute of the 'Text properties', the value of the linked variable will be shown respectively inserted if one of the below explained formatters is used inside of the text. printf formatters, %[width][.precision]specifier. Specifier: %b, yields an integer as a binary number %c, yields an integer as the character with that ASCII value %d or %i, yields an integer as a signed decimal number %e, yields a float using scientific notation %u, yields an integer as an unsigned decimal number %f, yields a float as is; see example for precision %o, yields an integer as an octal number %s, yields a string as is %x, yields an integer as a hexadecimal number (lower-case) %X, yields an integer as a hexadecimal number (upper-case) %j, yields a JavaScript object or array as a JSON encoded string e.g. The color: %s, value=blue => The color: blue First number: %d, value=12345 => First number: 12345 Second number: %04d, value=25 => Second number: 0025 Third number: %i, value=1234 => Third number: 1234 Float number: %3.2f, value=3.14159 => Float number: 3.14 Hexadecimal: %x, value=255 => Hexadecimal: ff Octal: %o, value=255 => Octal: 377 Unsigned value: %u, value=150 => Unsigned value: 150 html tags can also be used for formatting purposes bold text
 line break ordered list list item important text e.g. This is normal text - and this is bold text, defines bold text areas My text in first line
 my text in second line, defines a single line break first line second line third line, defines an ordered list see also:
------	---

Variable	The linked variable, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.BLINK globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive paramters is normaly '0'
Font color	The color of the font, it can only be selected if no theme is selected.
Font size	Font size can only be selected if no theme is selected.
Font Name	Font Name can only be selected if no theme is selected.
Custom Filter	Name of filter function to format the displayed value.
Variable Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
Tooltip	Text that is displayed when the mouse pointer is over the item. Be aware that in case of a touch device the tooltip can't be displayed.
Update interval	Length of the update rate The following update rates are available: 'none': Value is only read once when screen is opened 'slowest': 10000ms 'slow': 5000ms 'default': 1000ms 'fast': 500ms 'fastest': 200ms Be aware that a fast update rate increases the performance load of the application and the control.

6.16.5 Dynamic fill color

Variable for Dynamic FillColor	The linked variable, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.BLINK globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive paramters is normaly '0'
Variable Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be checked against the operators and statments of the defined fill colors. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
No of Fill Color	Number of fill color for the widget. Minimum Value : 1 Maximum Value : 16 First match will be taken and rest is ignored. E.g. if "Operator" and "Statement" are matching for fill color 3, then the statements for all following defined fill colors (4,5,...) will be ignored. Hint: Only 'Range' operator (x to Y) is possible if number of fill colors is more then 1

6.16.6 Dynamic movements

Position X	Movement of widget along X - absolute: from left edge of the screen - relative: from the original placement of the widget
Variable X	Linked variable controls the movement of the element in X-direction, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive parameters is normally '0'
Variable X Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
Position Y	Movement of widget along Y - absolute: from left edge of the screen - relative: from the original placement of the widget

Attachment: Widget attribute

Variable Y	Linked variable controls the movement of the element in Y-direction, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive paramters is normaly '0'
Variable Y Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".

6.16.7 Dynamic size

Variable width	Linked variable controls the width of the object, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0047, Position command value' with '0'= control Id, '22'= drive address: 0,22,S-0-0047 Hint: Control Id for drive paramters is normaly '0'
Variable width script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".

Attachment: Widget attribute

Variable height	Linked variable controls the height of the object, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0047, Position command value' with '0'= control Id, '22'= drive address: 0,22,S-0-0047 Hint: Control Id for drive paramters is normaly '0'
Variable height script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".

6.16.8 Dynamic rotation

Variable for Dynamic Rotation	Linked variable controls the rotation of the element, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive paramters is normaly '0'
Variable Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
Rotation base	Specifies the point, where the rotation is based of

6.16.9 Events

OnMouseClicked	The event is executed if a full click is performed on the element, ie. a mouse press and a mouse release
OnMouseUp	The event is executed if a mouse button is released on this element
OnMouseDown	The event is executed if a mouse button is pressed on this element
OnMouseOver	The event is fires when when the mouse cursor enters the element
OnMouseOut	The event is fires when the mouse cursor leaves the element
OnRightClick	This event fires when the right mouse button is pressed
OnDoubleClick	This event fires on double mouse click

Attachment: Widget attribute

Push value	The linked variable, which is either entered or selected via the process variable browser, is set to TRUE (or FALSE) as long as the mouse is getting pressed. This is a convenience functionality and usually used with the "Button" widget and useful to set bool variables or single bits of an integer (use {bit} to set a single bit of integers). Depending on the 'Push value FALSE' attribute, either TRUE or FALSE is set when the button is pressed. Be aware that this feature used with active widgets (like "Input field" or "Checkbox") can cause problems when the same linked variable of the text properties is used because both try to write to the same variable independently! e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.myBool ns=2;s=Application.globalVAR.myInt{7}, with '{7}' = only this bit is toggled JavaScript variables: globalVariables.myBool screenVariables.myInt{14}, with '{14}' = only this bit is toggled Control-Parameter 'C-0-0080, User defined decimal number', with '{4}' = only this bit is used: C-0-0080{4} Axis-Parameter 'A-0-0048, Velocity polarities' with '0'= control Id, '3'= axis address, '{0}' = only this bit is toggled: 0,3,A-0-0048{0} Drive-Parameter 'P-0-0008, Activation E-Stop function' with '0'= control Id, '22'= drive address, '{1}' = only this bit is toggled: 0,22,P-0-0008{1} Hint: Control Id for drive parameters is usually '0'
------------	--

Push value FALSE	If set to 'true', the 'Push value' written to the linked variable as long as the mouse is getting pressed is FALSE. Attribute is 'false' (default) --> TRUE is written to the linked variable as long as the mouse is getting pressed Attribute is 'true' --> FALSE is written to the linked variable as long as the mouse is getting pressed
Toggle	The linked variable, which is either entered or selected via the process variable browser, is toggled between TRUE and FALSE when clicked. This is a convenience functionality and only useful to toggle bool variables or single bits of an integer (use {bit} to toggle a single bit of integers). Be aware that this feature used with active widgets (like "Input field" or "Checkbox") can cause problems when the same linked variable of the text properties is used because both try to write to the same variable independently! e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.myBool ns=2;s=Application.globalVAR.myInt{7}, with '{7}' = only this bit is toggled JavaScript variables: globalVariables.myBool screenVariables.myInt{14}, with '{14}' = only this bit is toggled Control-Parameter 'C-0-0080, User defined decimal number', with '{4}' = only this bit is used: C-0-0080{4} Axis-Parameter 'A-0-0048, Velocity polarities' with '0'= control Id, '3'= axis address, '{0}' = only this bit is toggled: 0,3,A-0-0048{0} Drive-Parameter 'P-0-0008, Activation E-Stop function' with '0'= control Id, '22'= drive address, '{1}' = only this bit is toggled: 0,22,P-0-0008{1}

6.17 Rectangle

6.17.1 General

Type	Type of the element
Id	Id of the element
Alias	Unique widget identifier

6.17.2 Position

X position	Distance from the left edge in pixels
Y position	Distance from the top edge in pixels
Width	Width of the element in pixel, including border
Height	Height of the element in pixel, including border
Rotation	Rotation of the element about its center in degrees

6.17.3 Style

Fill color	Background fill color
Border color	The border color, it can only be selected if no theme is selected.
Border width	Border width in pixel
Transparency	A value between 0 and 100, 0: the object appears completely transparent, 100: the object completely covers the background
Shadow	Show a drop shadow

6.17.4 Text properties

Text	Entered text will be shown. If a variable is entered in the 'Variable' attribute of the 'Text properties', the value of the linked variable will be shown respectively inserted if one of the below explained formatters is used inside of the text. printf formatters, %[width][.precision]specifier. Specifier: %b, yields an integer as a binary number %c, yields an integer as the character with that ASCII value %d or %i, yields an integer as a signed decimal number %e, yields a float using scientific notation %u, yields an integer as an unsigned decimal number %f, yields a float as is; see example for precision %o, yields an integer as an octal number %s, yields a string as is %x, yields an integer as a hexadecimal number (lower-case) %X, yields an integer as a hexadecimal number (upper-case) %j, yields a JavaScript object or array as a JSON encoded string e.g. The color: %s, value=blue => The color: blue First number: %d, value=12345 => First number: 12345 Second number: %04d, value=25 => Second number: 0025 Third number: %i, value=1234 => Third number: 1234 Float number: %3.2f, value=3.14159 => Float number: 3.14 Hexadecimal: %x, value=255 => Hexadecimal: ff Octal: %o, value=255 => Octal: 377 Unsigned value: %u, value=150 => Unsigned value: 150 html tags can also be used for formatting purposes bold text
 line break ordered list list item important text e.g This is normal text - and this is bold text, defines bold text areas My text in first line
 my text in second line, defines a single line break first line second line third line, defines an ordered list see also:
------	--

Attachment: Widget attribute

Variable	The linked variable, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.BLINK globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive paramters is normaly '0'
Font color	The color of the font, it can only be selected if no theme is selected.
Font size	Font size can only be selected if no theme is selected.
Font Name	Font Name can only be selected if no theme is selected.
Custom Filter	Name of filter function to format the displayed value.
Variable Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
Tooltip	Text that is displayed when the mouse pointer is over the item. Be aware that in case of a touch device the tooltip can't be displayed.
Horizontal alignment	Alignment of the contents horizontally (left, centered or right)
Vertical alignment	Alignment of the contents vertically (top, centered or bottom)
Update interval	Length of the update rate The following update rates are available: 'none': Value is only read once when screen is opened 'slowest': 10000ms 'slow': 5000ms 'default': 1000ms 'fast': 500ms 'fastest': 200ms Be aware that a fast update rate increases the performance load of the application and the control.

6.17.5 Dynamic fill color

Variable for Dynamic FillColor	The linked variable, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.BLINK globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive paramters is normaly '0'
Variable Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be checked against the operators and statments of the defined fill colors. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
No of Fill Color	Number of fill color for the widget. Minimum Value : 1 Maximum Value : 16 First match will be taken and rest is ignored. E.g. if "Operator" and "Statement" are matching for fill color 3, then the statements for all following defined fill colors (4,5,...) will be ignored. Hint: Only 'Range' operator (x to Y) is possible if number of fill colors is more then 1

6.17.6 Dynamic movements

Position X	Movement of widget along X - absolute: from left edge of the screen - relative: from the original placement of the widget
Variable X	Linked variable controls the movement of the element in X-direction, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive parameters is normally '0'
Variable X Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
Position Y	Movement of widget along Y - absolute: from left edge of the screen - relative: from the original placement of the widget

Variable Y	Linked variable controls the movement of the element in Y-direction, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive paramters is normaly '0'
Variable Y Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".

6.17.7 Dynamic size

Variable width	Linked variable controls the width of the object, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0047, Position command value' with '0'= control Id, '22'= drive address: 0,22,S-0-0047 Hint: Control Id for drive paramters is normaly '0'
Variable width script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".

Variable height	Linked variable controls the height of the object, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0047, Position command value' with '0'= control Id, '22'= drive address: 0,22,S-0-0047 Hint: Control Id for drive paramters is normaly '0'
Variable height script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".

6.17.8 Dynamic rotation

Variable for Dynamic Rotation	Linked variable controls the rotation of the element, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive paramters is normaly '0'
Variable Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
Rotation base	Specifies the point, where the rotation is based of

6.17.9 Events

OnClick	The event is executed if a full click is performed on the element, ie. a mouse press and a mouse release
OnMouseUp	The event is executed if a mouse button is released on this element
OnMouseDown	The event is executed if a mouse button is pressed on this element
OnMouseOver	The event is fires when when the mouse cursor enters the element
OnMouseOut	The event is fires when the mouse cursor leaves the element
OnRightClick	This event fires when the right mouse button is pressed
OnDoubleClick	This event fires on double mouse click

Push value	The linked variable, which is either entered or selected via the process variable browser, is set to TRUE (or FALSE) as long as the mouse is getting pressed. This is a convenience functionality and usually used with the "Button" widget and useful to set bool variables or single bits of an integer (use {bit} to set a single bit of integers). Depending on the 'Push value FALSE' attribute, either TRUE or FALSE is set when the button is pressed. Be aware that this feature used with active widgets (like "Input field" or "Chechbox") can cause problems when the same linked variable of the text properties is used because both try to write to the same variable independently! e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.myBool ns=2;s=Application.globalVAR.myInt{7}, with '{7}' = only this bit is toggled JavaScript variables: globalVariables.myBool screenVariables.myInt{14}, with '{14}' = only this bit is toggled Control-Parameter 'C-0-0080, User defined decimal number', with '{4}' = only this bit is used: C-0-0080{4} Axis-Parameter 'A-0-0048, Velocity polarities' with '0'= control Id, '3'= axis address, '{0}' = only this bit is toggled: 0,3,A-0-0048{0} Drive-Parameter 'P-0-0008, Activation E-Stop function' with '0'= control Id, '22'= drive address, '{1}' = only this bit is toggled: 0,22,P-0-0008{1} Hint: Control Id for drive paramters is usually '0'
------------	---

Attachment: Widget attribute

Push value FALSE	If set to 'true', the 'Push value' written to the linked variable as long as the mouse is getting pressed is FALSE. Attribute is 'false' (default) --> TRUE is written to the linked variable as long as the mouse is getting pressed Attribute is 'true' --> FALSE is written to the linked variable as long as the mouse is getting pressed
Toggle	The linked variable, which is either entered or selected via the process variable browser, is toggled between TRUE and FALSE when clicked. This is a convenience functionality and only useful to toggle bool variables or single bits of an integer (use {bit} to toggle a single bit of integers). Be aware that this feature used with active widgets (like "Input field" or "Checkbox") can cause problems when the same linked variable of the text properties is used because both try to write to the same variable independently! e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.myBool ns=2;s=Application.globalVAR.myInt{7}, with '{7}' = only this bit is toggled JavaScript variables: globalVariables.myBool screenVariables.myInt{14}, with '{14}' = only this bit is toggled Control-Parameter 'C-0-0080, User defined decimal number', with '{4}' = only this bit is used: C-0-0080{4} Axis-Parameter 'A-0-0048, Velocity polarities' with '0'= control Id, '3'= axis address, '{0}' = only this bit is toggled: 0,3,A-0-0048{0} Drive-Parameter 'P-0-0008, Activation E-Stop function' with '0'= control Id, '22'= drive address, '{1}' = only this bit is toggled: 0,22,P-0-0008{1}

6.18 Rotary switch

6.18.1 General

Type	Type of the element
Id	Id of the element
Alias	Unique widget identifier

6.18.2 Position

X position	Distance from the left edge in pixels
Y position	Distance from the top edge in pixels
Width	Width of the element in pixel, including border
Height	Height of the element in pixel, including border
Rotation	Rotation of the element about its center in degrees

6.18.3 Style

Fill color	Background fill color
Border color	The border color, it can only be selected if no theme is selected.
Border width	Border width in pixel
Transparency	A value between 0 and 100, 0: the object appears completely transparent, 100: the object completely covers the background
Shadow	Show a drop shadow

6.18.4 Text properties

Text	Entered text will be shown. If a variable is entered in the 'Variable' attribute of the 'Text properties', the value of the linked variable will be shown respectively inserted if one of the below explained formatters is used inside of the text. printf formatters, %[width][.precision]specifier. Specifier: %b, yields an integer as a binary number %c, yields an integer as the character with that ASCII value %d or %i, yields an integer as a signed decimal number %e, yields a float using scientific notation %u, yields an integer as an unsigned decimal number %f, yields a float as is; see example for precision %o, yields an integer as an octal number %s, yields a string as is %x, yields an integer as a hexadecimal number (lower-case) %X, yields an integer as a hexadecimal number (upper-case) %j, yields a JavaScript object or array as a JSON encoded string e.g. The color: %s, value=blue => The color: blue First number: %d, value=12345 => First number: 12345 Second number: %04d, value=25 => Second number: 0025 Third number: %i, value=1234 => Third number: 1234 Float number: %3.2f, value=3.14159 => Float number: 3.14 Hexadecimal: %x, value=255 => Hexadecimal: ff Octal: %o, value=255 => Octal: 377 Unsigned value: %u, value=150 => Unsigned value: 150 html tags can also be used for formatting purposes bold text
 line break ordered list list item important text e.g This is normal text - and this is bold text, defines bold text areas My text in first line
 my text in second line, defines a single line break first line second line third line, defines an ordered list see also:
Font color	The color of the font, it can only be selected if no theme is selected.
Font size	Font size can only be selected if no theme is selected.
Font Name	Font Name can only be selected if no theme is selected.
Custom Filter	Name of filter function to format the displayed value.

Variable Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
Tooltip	Text that is displayed when the mouse pointer is over the item. Be aware that in case of a touch device the tooltip can't be displayed.
Update interval	Length of the update rate The following update rates are available: 'none': Value is only read once when screen is opened 'slowest': 10000ms 'slow': 5000ms 'default': 1000ms 'fast': 500ms 'fastest': 200ms Be aware that a fast update rate increases the performance load of the application and the control.

6.18.5 Dynamic fill color

Variable for Dynamic FillColor	The linked variable, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.BLINK globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive paramters is normaly '0'
Variable Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be checked against the operators and statments of the defined fill colors. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
No of Fill Color	Number of fill color for the widget. Minimum Value : 1 Maximum Value : 16 First match will be taken and rest is ignored. E.g. if "Operator" and "Statement" are matching for fill color 3, then the statements for all following defined fill colors (4,5,...) will be ignored. Hint: Only 'Range' operator (x to Y) is possible if number of fill colors is more then 1

6.18.6 Dynamic movements

Position X	Movement of widget along X - absolute: from left edge of the screen - relative: from the original placement of the widget
Variable X Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
Position Y	Movement of widget along Y - absolute: from left edge of the screen - relative: from the original placement of the widget
Variable Y Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".

6.18.7 Dynamic size

Variable width	Linked variable controls the width of the object, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0047, Position command value' with '0'= control Id, '22'= drive address: 0,22,S-0-0047 Hint: Control Id for drive paramters is normaly '0'
Variable width script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".

Variable height	Linked variable controls the height of the object, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0047, Position command value' with '0'= control Id, '22'= drive address: 0,22,S-0-0047 Hint: Control Id for drive paramters is normaly '0'
Variable height script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".

6.18.8 Dynamic rotation

Variable Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
Rotation base	Specifies the point, where the rotation is based of

6.18.9 Events

OnClick	The event is executed if a full click is performed on the element, ie. a mouse press and a mouse release
OnMouseUp	The event is executed if a mouse button is released on this element
OnMouseDown	The event is executed if a mouse button is pressed on this element
OnMouseOver	The event is fires when when the mouse cursor enters the element
OnMouseOut	The event is fires when the mouse cursor leaves the element

Attachment: Widget attribute

OnRightClick	This event fires when the right mouse button is pressed
OnDoubleClick	This event fires on double mouse click
Push value	The linked variable, which is either entered or selected via the process variable browser, is set to TRUE (or FALSE) as long as the mouse is getting pressed. This is a convenience functionality and usually used with the "Button" widget and useful to set bool variables or single bits of an integer (use {bit} to set a single bit of integers). Depending on the 'Push value FALSE' attribute, either TRUE or FALSE is set when the button is pressed. Be aware that this feature used with active widgets (like "Input field" or "Checkbox") can cause problems when the same linked variable of the text properties is used because both try to write to the same variable independently! e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.myBool ns=2;s=Application.globalVAR.myInt{7}, with '{7}' = only this bit is toggled JavaScript variables: globalVariables.myBool screenVariables.myInt{14}, with '{14}' = only this bit is toggled Control-Parameter 'C-0-0080, User defined decimal number', with '{4}' = only this bit is used: C-0-0080{4} Axis-Parameter 'A-0-0048, Velocity polarities' with '0'= control Id, '3'= axis address, '{0}' = only this bit is toggled: 0,3,A-0-0048{0} Drive-Parameter 'P-0-0008, Activation E-Stop function' with '0'= control Id, '22'= drive address, '{1}' = only this bit is toggled: 0,22,P-0-0008{1} Hint: Control Id for drive parameters is usually '0'

Push value FALSE	If set to 'true', the 'Push value' written to the linked variable as long as the mouse is getting pressed is FALSE. Attribute is 'false' (default) --> TRUE is written to the linked variable as long as the mouse is getting pressed Attribute is 'true' --> FALSE is written to the linked variable as long as the mouse is getting pressed
Toggle	The linked variable, which is either entered or selected via the process variable browser, is toggled between TRUE and FALSE when clicked. This is a convenience functionality and only useful to toggle bool variables or single bits of an integer (use {bit} to toggle a single bit of integers). Be aware that this feature used with active widgets (like "Input field" or "Checkbox") can cause problems when the same linked variable of the text properties is used because both try to write to the same variable independently! e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.myBool ns=2;s=Application.globalVAR.myInt{7}, with '{7}' = only this bit is toggled JavaScript variables: globalVariables.myBool screenVariables.myInt{14}, with '{14}' = only this bit is toggled Control-Parameter 'C-0-0080, User defined decimal number', with '{4}' = only this bit is used: C-0-0080{4} Axis-Parameter 'A-0-0048, Velocity polarities' with '0'= control Id, '3'= axis address, '{0}' = only this bit is toggled: 0,3,A-0-0048{0} Drive-Parameter 'P-0-0008, Activation E-Stop function' with '0'= control Id, '22'= drive address, '{1}' = only this bit is toggled: 0,22,P-0-0008{1}

6.19 Slider

6.19.1 General

Type	Type of the element
Id	Id of the element
Tab Id	Sets the tab order
Alias	Unique widget identifier

6.19.2 Position

X position	Distance from the left edge in pixels
Y position	Distance from the top edge in pixels
Width	Width of the element in pixel, including border
Height	Height of the element in pixel, including border
Rotation	Rotation of the element about its center in degrees

6.19.3 Style

Fill color	Background fill color
Border color	The border color, it can only be selected if no theme is selected.
Border width	Border width in pixel
Transparency	A value between 0 and 100, 0: the object appears completely transparent, 100: the object completely covers the background
Shadow	Show a drop shadow

6.19.4 Text properties

Text	Entered text will be shown. If a variable is entered in the 'Variable' attribute of the 'Text properties', the value of the linked variable will be shown respectively inserted if one of the below explained formatters is used inside of the text. printf formatters, %[width][.precision]specifier. Specifier: %b, yields an integer as a binary number %c, yields an integer as the character with that ASCII value %d or %i, yields an integer as a signed decimal number %e, yields a float using scientific notation %u, yields an integer as an unsigned decimal number %f, yields a float as is; see example for precision %o, yields an integer as an octal number %s, yields a string as is %x, yields an integer as a hexadecimal number (lower-case) %X, yields an integer as a hexadecimal number (upper-case) %j, yields a JavaScript object or array as a JSON encoded string e.g. The color: %s, value=blue => The color: blue First number: %d, value=12345 => First number: 12345 Second number: %04d, value=25 => Second number: 0025 Third number: %i, value=1234 => Third number: 1234 Float number: %3.2f, value=3.14159 => Float number: 3.14 Hexadecimal: %x, value=255 => Hexadecimal: ff Octal: %o, value=255 => Octal: 377 Unsigned value: %u, value=150 => Unsigned value: 150 html tags can also be used for formatting purposes bold text
 line break ordered list list item important text e.g This is normal text - and this is bold text, defines bold text areas My text in first line
 my text in second line, defines a single line break first line second line third line, defines an ordered list see also:
------	--

Attachment: Widget attribute

Variable	The linked variable, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.BLINK globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive paramters is normaly '0'
Font color	The color of the font, it can only be selected if no theme is selected.
Font size	Font size can only be selected if no theme is selected.
Font Name	Font Name can only be selected if no theme is selected.
Custom Filter	Name of filter function to format the displayed value.
Variable Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
Tooltip	Text that is displayed when the mouse pointer is over the item. Be aware that in case of a touch device the tooltip can't be displayed.
Update interval	Length of the update rate The following update rates are available: 'none': Value is only read once when screen is opened 'slowest': 10000ms 'slow': 5000ms 'default': 1000ms 'fast': 500ms 'fastest': 200ms Be aware that a fast update rate increases the performance load of the application and the control.

6.19.5 Dynamic fill color

Variable for Dynamic FillColor	The linked variable, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.BLINK globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive paramters is normaly '0'
Variable Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be checked against the operators and statments of the defined fill colors. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
No of Fill Color	Number of fill color for the widget. Minimum Value : 1 Maximum Value : 16 First match will be taken and rest is ignored. E.g. if "Operator" and "Statement" are matching for fill color 3, then the statements for all following defined fill colors (4,5,...) will be ignored. Hint: Only 'Range' operator (x to Y) is possible if number of fill colors is more then 1

6.19.6 Dynamic movements

Position X	Movement of widget along X - absolute: from left edge of the screen - relative: from the original placement of the widget
Variable X	Linked variable controls the movement of the element in X-direction, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive parameters is normally '0'
Variable X Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
Position Y	Movement of widget along Y - absolute: from left edge of the screen - relative: from the original placement of the widget

Variable Y	Linked variable controls the movement of the element in Y-direction, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive paramters is normaly '0'
Variable Y Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".

6.19.7 Dynamic size

Variable width	Linked variable controls the width of the object, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0047, Position command value' with '0'= control Id, '22'= drive address: 0,22,S-0-0047 Hint: Control Id for drive paramters is normaly '0'
Variable width script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".

Variable height	Linked variable controls the height of the object, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0047, Position command value' with '0'= control Id, '22'= drive address: 0,22,S-0-0047 Hint: Control Id for drive paramters is normaly '0'
Variable height script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".

6.19.8 Dynamic rotation

Variable for Dynamic Rotation	Linked variable controls the rotation of the element, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive paramters is normaly '0'
Variable Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
Rotation base	Specifies the point, where the rotation is based of

6.19.9 Events

OnClick	The event is executed if a full click is performed on the element, ie. a mouse press and a mouse release
OnMouseUp	The event is executed if a mouse button is released on this element
OnMouseDown	The event is executed if a mouse button is pressed on this element
OnMouseOver	The event is fires when when the mouse cursor enters the element
OnMouseOut	The event is fires when the mouse cursor leaves the element
OnRightClick	This event fires when the right mouse button is pressed
OnDoubleClick	This event fires on double mouse click

Push value	The linked variable, which is either entered or selected via the process variable browser, is set to TRUE (or FALSE) as long as the mouse is getting pressed. This is a convenience functionality and usually used with the "Button" widget and useful to set bool variables or single bits of an integer (use {bit} to set a single bit of integers). Depending on the 'Push value FALSE' attribute, either TRUE or FALSE is set when the button is pressed. Be aware that this feature used with active widgets (like "Input field" or "Chechbox") can cause problems when the same linked variable of the text properties is used because both try to write to the same variable independently! e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.myBool ns=2;s=Application.globalVAR.myInt{7}, with '{7}' = only this bit is toggled JavaScript variables: globalVariables.myBool screenVariables.myInt{14}, with '{14}' = only this bit is toggled Control-Parameter 'C-0-0080, User defined decimal number', with '{4}' = only this bit is used: C-0-0080{4} Axis-Parameter 'A-0-0048, Velocity polarities' with '0'= control Id, '3'= axis address, '{0}' = only this bit is toggled: 0,3,A-0-0048{0} Drive-Parameter 'P-0-0008, Activation E-Stop function' with '0'= control Id, '22'= drive address, '{1}' = only this bit is toggled: 0,22,P-0-0008{1} Hint: Control Id for drive paramters is usually '0'
------------	---

Attachment: Widget attribute

Push value FALSE	If set to 'true', the 'Push value' written to the linked variable as long as the mouse is getting pressed is FALSE. Attribute is 'false' (default) --> TRUE is written to the linked variable as long as the mouse is getting pressed Attribute is 'true' --> FALSE is written to the linked variable as long as the mouse is getting pressed
Toggle	The linked variable, which is either entered or selected via the process variable browser, is toggled between TRUE and FALSE when clicked. This is a convenience functionality and only useful to toggle bool variables or single bits of an integer (use {bit} to toggle a single bit of integers). Be aware that this feature used with active widgets (like "Input field" or "Checkbox") can cause problems when the same linked variable of the text properties is used because both try to write to the same variable independently! e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.myBool ns=2;s=Application.globalVAR.myInt{7}, with '{7}' = only this bit is toggled JavaScript variables: globalVariables.myBool screenVariables.myInt{14}, with '{14}' = only this bit is toggled Control-Parameter 'C-0-0080, User defined decimal number', with '{4}' = only this bit is used: C-0-0080{4} Axis-Parameter 'A-0-0048, Velocity polarities' with '0'= control Id, '3'= axis address, '{0}' = only this bit is toggled: 0,3,A-0-0048{0} Drive-Parameter 'P-0-0008, Activation E-Stop function' with '0'= control Id, '22'= drive address, '{1}' = only this bit is toggled: 0,22,P-0-0008{1}

6.20 Table

6.20.1 General

Type	Type of the element
Id	Id of the element
Tab Id	Sets the tab order
Alias	Unique widget identifier

6.20.2 Position

X position	Distance from the left edge in pixels
Y position	Distance from the top edge in pixels
Width	Width of the element in pixel, including border
Height	Height of the element in pixel, including border
Rotation	Rotation of the element about its center in degrees

6.20.3 Style

Fill color	Background fill color
Border color	The border color, it can only be selected if no theme is selected.
Border width	Border width in pixel
Transparency	A value between 0 and 100, 0: the object appears completely transparent, 100: the object completely covers the background
Shadow	Show a drop shadow

6.20.4 Text properties

Variable	The linked variable, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.BLINK globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive paramters is normaly '0'
Font color	The color of the font, it can only be selected if no theme is selected.
Font size	Font size can only be selected if no theme is selected.
Font Name	Font Name can only be selected if no theme is selected.
Custom Filter	Name of filter function to format the displayed value.
Variable Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".

Attachment: Widget attribute

Tooltip	Text that is displayed when the mouse pointer is over the item. Be aware that in case of a touch device the tooltip can't be displayed.
Update interval	Length of the update rate The following update rates are available: 'none': Value is only read once when screen is opened 'slowest': 10000ms 'slow': 5000ms 'default': 1000ms 'fast': 500ms 'fastest': 200ms Be aware that a fast update rate increases the performance load of the application and the control.

6.20.5 Dynamic fill color

Variable for Dynamic FillColor	The linked variable, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.BLINK globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive paramters is normaly '0'
Variable Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be checked against the operators and statments of the defined fill colors. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
No of Fill Color	Number of fill color for the widget. Minimum Value : 1 Maximum Value : 16 First match will be taken and rest is ignored. E.g. if "Operator" and "Statement" are matching for fill color 3, then the statements for all following defined fill colors (4,5,...) will be ignored. Hint: Only 'Range' operator (x to Y) is possible if number of fill colors is more then 1

6.20.6 Dynamic movements

Position X	Movement of widget along X - absolute: from left edge of the screen - relative: from the original placement of the widget
Variable X	Linked variable controls the movement of the element in X-direction, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive parameters is normally '0'
Variable X Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
Position Y	Movement of widget along Y - absolute: from left edge of the screen - relative: from the original placement of the widget

Variable Y	Linked variable controls the movement of the element in Y-direction, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive paramters is normaly '0'
Variable Y Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".

6.20.7 Dynamic size

Variable width	Linked variable controls the width of the object, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0047, Position command value' with '0'= control Id, '22'= drive address: 0,22,S-0-0047 Hint: Control Id for drive paramters is normaly '0'
Variable width script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".

Variable height	Linked variable controls the height of the object, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0047, Position command value' with '0'= control Id, '22'= drive address: 0,22,S-0-0047 Hint: Control Id for drive paramters is normaly '0'
Variable height script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".

6.20.8 Dynamic rotation

Variable for Dynamic Rotation	Linked variable controls the rotation of the element, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive paramters is normaly '0'
Variable Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
Rotation base	Specifies the point, where the rotation is based of

6.20.9 Events

OnClick	The event is executed if a full click is performed on the element, ie. a mouse press and a mouse release
OnMouseUp	The event is executed if a mouse button is released on this element
OnMouseDown	The event is executed if a mouse button is pressed on this element
OnMouseOver	The event is fires when when the mouse cursor enters the element
OnMouseOut	The event is fires when the mouse cursor leaves the element
OnValueChange	The event is fires when the assigned value has been changed
OnRightClick	This event fires when the right mouse button is pressed
OnBlur	This event is raised when the element loses focus
OnDoubleClick	This event fires on double mouse click
OnFocus	This event fires when the element gets focus

Push value	The linked variable, which is either entered or selected via the process variable browser, is set to TRUE (or FALSE) as long as the mouse is getting pressed. This is a convenience functionality and usually used with the "Button" widget and useful to set bool variables or single bits of an integer (use {bit} to set a single bit of integers). Depending on the 'Push value FALSE' attribute, either TRUE or FALSE is set when the button is pressed. Be aware that this feature used with active widgets (like "Input field" or "Chechbox") can cause problems when the same linked variable of the text properties is used because both try to write to the same variable independently! e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.myBool ns=2;s=Application.globalVAR.myInt{7}, with '{7}' = only this bit is toggled JavaScript variables: globalVariables.myBool screenVariables.myInt{14}, with '{14}' = only this bit is toggled Control-Parameter 'C-0-0080, User defined decimal number', with '{4}' = only this bit is used: C-0-0080{4} Axis-Parameter 'A-0-0048, Velocity polarities' with '0'= control Id, '3'= axis address, '{0}' = only this bit is toggled: 0,3,A-0-0048{0} Drive-Parameter 'P-0-0008, Activation E-Stop function' with '0'= control Id, '22'= drive address, '{1}' = only this bit is toggled: 0,22,P-0-0008{1} Hint: Control Id for drive paramters is usually '0'
------------	---

Attachment: Widget attribute

Push value FALSE	If set to 'true', the 'Push value' written to the linked variable as long as the mouse is getting pressed is FALSE. Attribute is 'false' (default) --> TRUE is written to the linked variable as long as the mouse is getting pressed Attribute is 'true' --> FALSE is written to the linked variable as long as the mouse is getting pressed
Toggle	The linked variable, which is either entered or selected via the process variable browser, is toggled between TRUE and FALSE when clicked. This is a convenience functionality and only useful to toggle bool variables or single bits of an integer (use {bit} to toggle a single bit of integers). Be aware that this feature used with active widgets (like "Input field" or "Checkbox") can cause problems when the same linked variable of the text properties is used because both try to write to the same variable independently! e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.myBool ns=2;s=Application.globalVAR.myInt{7}, with '{7}' = only this bit is toggled JavaScript variables: globalVariables.myBool screenVariables.myInt{14}, with '{14}' = only this bit is toggled Control-Parameter 'C-0-0080, User defined decimal number', with '{4}' = only this bit is used: C-0-0080{4} Axis-Parameter 'A-0-0048, Velocity polarities' with '0'= control Id, '3'= axis address, '{0}' = only this bit is toggled: 0,3,A-0-0048{0} Drive-Parameter 'P-0-0008, Activation E-Stop function' with '0'= control Id, '22'= drive address, '{1}' = only this bit is toggled: 0,22,P-0-0008{1}

6.21 Text field

6.21.1 General

Type	Type of the element
Id	Id of the element
Alias	Unique widget identifier

6.21.2 Position

X position	Distance from the left edge in pixels
Y position	Distance from the top edge in pixels
Width	Width of the element in pixel, including border
Height	Height of the element in pixel, including border
Rotation	Rotation of the element about its center in degrees

6.21.3 Style

Fill color	Background fill color
Border color	The border color, it can only be selected if no theme is selected.
Border width	Border width in pixel
Transparency	A value between 0 and 100, 0: the object appears completely transparent, 100: the object completely covers the background
Shadow	Show a drop shadow

6.21.4 Text properties

Text	Entered text will be shown. If a variable is entered in the 'Variable' attribute of the 'Text properties', the value of the linked variable will be shown respectively inserted if one of the below explained formatters is used inside of the text. printf formatters, %[width][.precision]specifier. Specifier: %b, yields an integer as a binary number %c, yields an integer as the character with that ASCII value %d or %i, yields an integer as a signed decimal number %e, yields a float using scientific notation %u, yields an integer as an unsigned decimal number %f, yields a float as is; see example for precision %o, yields an integer as an octal number %s, yields a string as is %x, yields an integer as a hexadecimal number (lower-case) %X, yields an integer as a hexadecimal number (upper-case) %j, yields a JavaScript object or array as a JSON encoded string e.g. The color: %s, value=blue => The color: blue First number: %d, value=12345 => First number: 12345 Second number: %04d, value=25 => Second number: 0025 Third number: %i, value=1234 => Third number: 1234 Float number: %3.2f, value=3.14159 => Float number: 3.14 Hexadecimal: %x, value=255 => Hexadecimal: ff Octal: %o, value=255 => Octal: 377 Unsigned value: %u, value=150 => Unsigned value: 150 html tags can also be used for formatting purposes bold text
 line break ordered list list item important text e.g. This is normal text - and this is bold text, defines bold text areas My text in first line
 my text in second line, defines a single line break first line second line third line, defines an ordered list see also:
------	---

Variable	The linked variable, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.BLINK globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive paramters is normaly '0'
Font color	The color of the font, it can only be selected if no theme is selected.
Font size	Font size can only be selected if no theme is selected.
Font Name	Font Name can only be selected if no theme is selected.
Custom Filter	Name of filter function to format the displayed value.
Variable Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
Tooltip	Text that is displayed when the mouse pointer is over the item. Be aware that in case of a touch device the tooltip can't be displayed.
Horizontal alignment	Alignment of the contents horizontally (left, centered or right)
Vertical alignment	Alignment of the contents vertically (top, centered or bottom)
Update interval	Length of the update rate The following update rates are available: 'none': Value is only read once when screen is opened 'slowest': 10000ms 'slow': 5000ms 'default': 1000ms 'fast': 500ms 'fastest': 200ms Be aware that a fast update rate increases the performance load of the application and the control.

6.21.5 Dynamic fill color

Variable for Dynamic FillColor	The linked variable, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.BLINK globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive paramters is normaly '0'
Variable Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be checked against the operators and statments of the defined fill colors. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
No of Fill Color	Number of fill color for the widget. Minimum Value : 1 Maximum Value : 16 First match will be taken and rest is ignored. E.g. if "Operator" and "Statement" are matching for fill color 3, then the statements for all following defined fill colors (4,5,...) will be ignored. Hint: Only 'Range' operator (x to Y) is possible if number of fill colors is more then 1

6.21.6 Dynamic movements

Position X	Movement of widget along X - absolute: from left edge of the screen - relative: from the original placement of the widget
Variable X	Linked variable controls the movement of the element in X-direction, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive parameters is normally '0'
Variable X Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
Position Y	Movement of widget along Y - absolute: from left edge of the screen - relative: from the original placement of the widget

Attachment: Widget attribute

Variable Y	Linked variable controls the movement of the element in Y-direction, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive paramters is normaly '0'
Variable Y Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".

6.21.7 Dynamic size

Variable width	Linked variable controls the width of the object, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0047, Position command value' with '0'= control Id, '22'= drive address: 0,22,S-0-0047 Hint: Control Id for drive paramters is normaly '0'
Variable width script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".

Attachment: Widget attribute

Variable height	Linked variable controls the height of the object, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0047, Position command value' with '0'= control Id, '22'= drive address: 0,22,S-0-0047 Hint: Control Id for drive paramters is normaly '0'
Variable height script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".

6.21.8 Dynamic rotation

Variable for Dynamic Rotation	Linked variable controls the rotation of the element, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive paramters is normaly '0'
Variable Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
Rotation base	Specifies the point, where the rotation is based of

6.21.9 Events

OnMouseClicked	The event is executed if a full click is performed on the element, ie. a mouse press and a mouse release
OnMouseUp	The event is executed if a mouse button is released on this element
OnMouseDown	The event is executed if a mouse button is pressed on this element
OnMouseOver	The event is fires when when the mouse cursor enters the element
OnMouseOut	The event is fires when the mouse cursor leaves the element
OnRightClick	This event fires when the right mouse button is pressed
OnDoubleClick	This event fires on double mouse click

Attachment: Widget attribute

Push value	The linked variable, which is either entered or selected via the process variable browser, is set to TRUE (or FALSE) as long as the mouse is getting pressed. This is a convenience functionality and usually used with the "Button" widget and useful to set bool variables or single bits of an integer (use {bit} to set a single bit of integers). Depending on the 'Push value FALSE' attribute, either TRUE or FALSE is set when the button is pressed. Be aware that this feature used with active widgets (like "Input field" or "Checkbox") can cause problems when the same linked variable of the text properties is used because both try to write to the same variable independently! e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.myBool ns=2;s=Application.globalVAR.myInt{7}, with '{7}' = only this bit is toggled JavaScript variables: globalVariables.myBool screenVariables.myInt{14}, with '{14}' = only this bit is toggled Control-Parameter 'C-0-0080, User defined decimal number', with '{4}' = only this bit is used: C-0-0080{4} Axis-Parameter 'A-0-0048, Velocity polarities' with '0'= control Id, '3'= axis address, '{0}' = only this bit is toggled: 0,3,A-0-0048{0} Drive-Parameter 'P-0-0008, Activation E-Stop function' with '0'= control Id, '22'= drive address, '{1}' = only this bit is toggled: 0,22,P-0-0008{1} Hint: Control Id for drive parameters is usually '0'
------------	--

Push value FALSE	If set to 'true', the 'Push value' written to the linked variable as long as the mouse is getting pressed is FALSE. Attribute is 'false' (default) --> TRUE is written to the linked variable as long as the mouse is getting pressed Attribute is 'true' --> FALSE is written to the linked variable as long as the mouse is getting pressed
Toggle	The linked variable, which is either entered or selected via the process variable browser, is toggled between TRUE and FALSE when clicked. This is a convenience functionality and only useful to toggle bool variables or single bits of an integer (use {bit} to toggle a single bit of integers). Be aware that this feature used with active widgets (like "Input field" or "Checkbox") can cause problems when the same linked variable of the text properties is used because both try to write to the same variable independently! e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.myBool ns=2;s=Application.globalVAR.myInt{7}, with '{7}' = only this bit is toggled JavaScript variables: globalVariables.myBool screenVariables.myInt{14}, with '{14}' = only this bit is toggled Control-Parameter 'C-0-0080, User defined decimal number', with '{4}' = only this bit is used: C-0-0080{4} Axis-Parameter 'A-0-0048, Velocity polarities' with '0'= control Id, '3'= axis address, '{0}' = only this bit is toggled: 0,3,A-0-0048{0} Drive-Parameter 'P-0-0008, Activation E-Stop function' with '0'= control Id, '22'= drive address, '{1}' = only this bit is toggled: 0,22,P-0-0008{1}

6.22 Text box

6.22.1 General

Type	Type of the element
Id	Id of the element
Alias	Unique widget identifier

6.22.2 Position

X position	Distance from the left edge in pixels
Y position	Distance from the top edge in pixels
Width	Width of the element in pixel, including border
Height	Height of the element in pixel, including border
Rotation	Rotation of the element about its center in degrees

6.22.3 Style

Fill color	Background fill color
Border color	The border color, it can only be selected if no theme is selected.
Border width	Border width in pixel
Transparency	A value between 0 and 100, 0: the object appears completely transparent, 100: the object completely covers the background
Shadow	Show a drop shadow

6.22.4 Text properties

Text	Entered text will be shown. If a variable is entered in the 'Variable' attribute of the 'Text properties', the value of the linked variable will be shown respectively inserted if one of the below explained formatters is used inside of the text. printf formatters, %[width][.precision]specifier. Specifier: %b, yields an integer as a binary number %c, yields an integer as the character with that ASCII value %d or %i, yields an integer as a signed decimal number %e, yields a float using scientific notation %u, yields an integer as an unsigned decimal number %f, yields a float as is; see example for precision %o, yields an integer as an octal number %s, yields a string as is %x, yields an integer as a hexadecimal number (lower-case) %X, yields an integer as a hexadecimal number (upper-case) %j, yields a JavaScript object or array as a JSON encoded string e.g. The color: %s, value=blue => The color: blue First number: %d, value=12345 => First number: 12345 Second number: %04d, value=25 => Second number: 0025 Third number: %i, value=1234 => Third number: 1234 Float number: %3.2f, value=3.14159 => Float number: 3.14 Hexadecimal: %x, value=255 => Hexadecimal: ff Octal: %o, value=255 => Octal: 377 Unsigned value: %u, value=150 => Unsigned value: 150 html tags can also be used for formatting purposes bold text
 line break ordered list list item important text e.g This is normal text - and this is bold text, defines bold text areas My text in first line
 my text in second line, defines a single line break first line second line third line, defines an ordered list see also:
------	--

Attachment: Widget attribute

Variable	The linked variable, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.BLINK globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive paramters is normaly '0'
Font color	The color of the font, it can only be selected if no theme is selected.
Font size	Font size can only be selected if no theme is selected.
Font Name	Font Name can only be selected if no theme is selected.
Custom Filter	Name of filter function to format the displayed value.
Variable Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
Tooltip	Text that is displayed when the mouse pointer is over the item. Be aware that in case of a touch device the tooltip can't be displayed.
Horizontal alignment	Alignment of the contents horizontally (left, centered or right)
Update interval	Length of the update rate The following update rates are available: 'none': Value is only read once when screen is opened 'slowest': 10000ms 'slow': 5000ms 'default': 1000ms 'fast': 500ms 'fastest': 200ms Be aware that a fast update rate increases the performance load of the application and the control.

6.22.5 Dynamic fill color

Variable for Dynamic FillColor	The linked variable, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.BLINK globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive paramters is normaly '0'
Variable Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be checked against the operators and statments of the defined fill colors. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
No of Fill Color	Number of fill color for the widget. Minimum Value : 1 Maximum Value : 16 First match will be taken and rest is ignored. E.g. if "Operator" and "Statement" are matching for fill color 3, then the statements for all following defined fill colors (4,5,...) will be ignored. Hint: Only 'Range' operator (x to Y) is possible if number of fill colors is more then 1

6.22.6 Dynamic movements

Position X	Movement of widget along X - absolute: from left edge of the screen - relative: from the original placement of the widget
Variable X	Linked variable controls the movement of the element in X-direction, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive parameters is normally '0'
Variable X Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
Position Y	Movement of widget along Y - absolute: from left edge of the screen - relative: from the original placement of the widget

Variable Y	Linked variable controls the movement of the element in Y-direction, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive paramters is normaly '0'
Variable Y Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".

6.22.7 Dynamic size

Variable width	Linked variable controls the width of the object, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0047, Position command value' with '0'= control Id, '22'= drive address: 0,22,S-0-0047 Hint: Control Id for drive paramters is normaly '0'
Variable width script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".

Variable height	Linked variable controls the height of the object, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0047, Position command value' with '0'= control Id, '22'= drive address: 0,22,S-0-0047 Hint: Control Id for drive paramters is normaly '0'
Variable height script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".

6.22.8 Dynamic rotation

Variable for Dynamic Rotation	Linked variable controls the rotation of the element, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive paramters is normaly '0'
Variable Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
Rotation base	Specifies the point, where the rotation is based of

6.22.9 Events

OnClick	The event is executed if a full click is performed on the element, ie. a mouse press and a mouse release
OnMouseUp	The event is executed if a mouse button is released on this element
OnMouseDown	The event is executed if a mouse button is pressed on this element
OnMouseOver	The event is fires when when the mouse cursor enters the element
OnMouseOut	The event is fires when the mouse cursor leaves the element
OnRightClick	This event fires when the right mouse button is pressed
OnDoubleClick	This event fires on double mouse click

Push value	The linked variable, which is either entered or selected via the process variable browser, is set to TRUE (or FALSE) as long as the mouse is getting pressed. This is a convenience functionality and usually used with the "Button" widget and useful to set bool variables or single bits of an integer (use {bit} to set a single bit of integers). Depending on the 'Push value FALSE' attribute, either TRUE or FALSE is set when the button is pressed. Be aware that this feature used with active widgets (like "Input field" or "Chechbox") can cause problems when the same linked variable of the text properties is used because both try to write to the same variable independently! e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.myBool ns=2;s=Application.globalVAR.myInt{7}, with '{7}' = only this bit is toggled JavaScript variables: globalVariables.myBool screenVariables.myInt{14}, with '{14}' = only this bit is toggled Control-Parameter 'C-0-0080, User defined decimal number', with '{4}' = only this bit is used: C-0-0080{4} Axis-Parameter 'A-0-0048, Velocity polarities' with '0'= control Id, '3'= axis address, '{0}' = only this bit is toggled: 0,3,A-0-0048{0} Drive-Parameter 'P-0-0008, Activation E-Stop function' with '0'= control Id, '22'= drive address, '{1}' = only this bit is toggled: 0,22,P-0-0008{1} Hint: Control Id for drive paramters is usually '0'
------------	---

Attachment: Widget attribute

Push value FALSE	If set to 'true', the 'Push value' written to the linked variable as long as the mouse is getting pressed is FALSE. Attribute is 'false' (default) --> TRUE is written to the linked variable as long as the mouse is getting pressed Attribute is 'true' --> FALSE is written to the linked variable as long as the mouse is getting pressed
Toggle	The linked variable, which is either entered or selected via the process variable browser, is toggled between TRUE and FALSE when clicked. This is a convenience functionality and only useful to toggle bool variables or single bits of an integer (use {bit} to toggle a single bit of integers). Be aware that this feature used with active widgets (like "Input field" or "Checkbox") can cause problems when the same linked variable of the text properties is used because both try to write to the same variable independently! e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.myBool ns=2;s=Application.globalVAR.myInt{7}, with '{7}' = only this bit is toggled JavaScript variables: globalVariables.myBool screenVariables.myInt{14}, with '{14}' = only this bit is toggled Control-Parameter 'C-0-0080, User defined decimal number', with '{4}' = only this bit is used: C-0-0080{4} Axis-Parameter 'A-0-0048, Velocity polarities' with '0'= control Id, '3'= axis address, '{0}' = only this bit is toggled: 0,3,A-0-0048{0} Drive-Parameter 'P-0-0008, Activation E-Stop function' with '0'= control Id, '22'= drive address, '{1}' = only this bit is toggled: 0,22,P-0-0008{1}

6.23 Toogle button

6.23.1 General

Type	Type of the element
Id	Id of the element
Tab Id	Sets the tab order
Alias	Unique widget identifier

6.23.2 Position

X position	Distance from the left edge in pixels
Y position	Distance from the top edge in pixels
Width	Width of the element in pixel, including border
Height	Height of the element in pixel, including border
Rotation	Rotation of the element about its center in degrees

6.23.3 Style

Fill color	Background fill color
Border color	The border color, it can only be selected if no theme is selected.
Border width	Border width in pixel
Transparency	A value between 0 and 100, 0: the object appears completely transparent, 100: the object completely covers the background
Shadow	Show a drop shadow

6.23.4 Text properties

Variable	The linked variable, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.BLINK globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive paramters is normaly '0'
Font color	The color of the font, it can only be selected if no theme is selected.
Font size	Font size can only be selected if no theme is selected.
Font Name	Font Name can only be selected if no theme is selected.
Custom Filter	Name of filter function to format the displayed value.
Variable Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".

Attachment: Widget attribute

Tooltip	Text that is displayed when the mouse pointer is over the item. Be aware that in case of a touch device the tooltip can't be displayed.
Update interval	Length of the update rate The following update rates are available: 'none': Value is only read once when screen is opened 'slowest': 10000ms 'slow': 5000ms 'default': 1000ms 'fast': 500ms 'fastest': 200ms Be aware that a fast update rate increases the performance load of the application and the control.

6.23.5 Dynamic fill color

Variable for Dynamic FillColor	The linked variable, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.BLINK globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive paramters is normaly '0'
Variable Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be checked against the operators and statments of the defined fill colors. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
No of Fill Color	Number of fill color for the widget. Minimum Value : 1 Maximum Value : 16 First match will be taken and rest is ignored. E.g. if "Operator" and "Statement" are matching for fill color 3, then the statements for all following defined fill colors (4,5,...) will be ignored. Hint: Only 'Range' operator (x to Y) is possible if number of fill colors is more then 1

6.23.6 Dynamic movements

Position X	Movement of widget along X - absolute: from left edge of the screen - relative: from the original placement of the widget
Variable X	Linked variable controls the movement of the element in X-direction, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive parameters is normally '0'
Variable X Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
Position Y	Movement of widget along Y - absolute: from left edge of the screen - relative: from the original placement of the widget

Variable Y	Linked variable controls the movement of the element in Y-direction, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive paramters is normaly '0'
Variable Y Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".

6.23.7 Dynamic size

Variable width	Linked variable controls the width of the object, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0047, Position command value' with '0'= control Id, '22'= drive address: 0,22,S-0-0047 Hint: Control Id for drive paramters is normaly '0'
Variable width script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".

Variable height	Linked variable controls the height of the object, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0047, Position command value' with '0'= control Id, '22'= drive address: 0,22,S-0-0047 Hint: Control Id for drive paramters is normaly '0'
Variable height script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".

6.23.8 Dynamic rotation

Variable for Dynamic Rotation	Linked variable controls the rotation of the element, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive paramters is normaly '0'
Variable Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
Rotation base	Specifies the point, where the rotation is based of

6.23.9 Events

OnClick	The event is executed if a full click is performed on the element, ie. a mouse press and a mouse release
OnMouseUp	The event is executed if a mouse button is released on this element
OnMouseDown	The event is executed if a mouse button is pressed on this element
OnMouseOver	The event is fires when when the mouse cursor enters the element
OnMouseOut	The event is fires when the mouse cursor leaves the element
OnRightClick	This event fires when the right mouse button is pressed
OnDoubleClick	This event fires on double mouse click

Push value	The linked variable, which is either entered or selected via the process variable browser, is set to TRUE (or FALSE) as long as the mouse is getting pressed. This is a convenience functionality and usually used with the "Button" widget and useful to set bool variables or single bits of an integer (use {bit} to set a single bit of integers). Depending on the 'Push value FALSE' attribute, either TRUE or FALSE is set when the button is pressed. Be aware that this feature used with active widgets (like "Input field" or "Chechbox") can cause problems when the same linked variable of the text properties is used because both try to write to the same variable independently! e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.myBool ns=2;s=Application.globalVAR.myInt{7}, with '{7}' = only this bit is toggled JavaScript variables: globalVariables.myBool screenVariables.myInt{14}, with '{14}' = only this bit is toggled Control-Parameter 'C-0-0080, User defined decimal number', with '{4}' = only this bit is used: C-0-0080{4} Axis-Parameter 'A-0-0048, Velocity polarities' with '0'= control Id, '3'= axis address, '{0}' = only this bit is toggled: 0,3,A-0-0048{0} Drive-Parameter 'P-0-0008, Activation E-Stop function' with '0'= control Id, '22'= drive address, '{1}' = only this bit is toggled: 0,22,P-0-0008{1} Hint: Control Id for drive paramters is usually '0'
------------	---

Attachment: Widget attribute

Push value FALSE	If set to 'true', the 'Push value' written to the linked variable as long as the mouse is getting pressed is FALSE. Attribute is 'false' (default) --> TRUE is written to the linked variable as long as the mouse is getting pressed Attribute is 'true' --> FALSE is written to the linked variable as long as the mouse is getting pressed
Toggle	The linked variable, which is either entered or selected via the process variable browser, is toggled between TRUE and FALSE when clicked. This is a convenience functionality and only useful to toggle bool variables or single bits of an integer (use {bit} to toggle a single bit of integers). Be aware that this feature used with active widgets (like "Input field" or "Checkbox") can cause problems when the same linked variable of the text properties is used because both try to write to the same variable independently! e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.myBool ns=2;s=Application.globalVAR.myInt{7}, with '{7}' = only this bit is toggled JavaScript variables: globalVariables.myBool screenVariables.myInt{14}, with '{14}' = only this bit is toggled Control-Parameter 'C-0-0080, User defined decimal number', with '{4}' = only this bit is used: C-0-0080{4} Axis-Parameter 'A-0-0048, Velocity polarities' with '0'= control Id, '3'= axis address, '{0}' = only this bit is toggled: 0,3,A-0-0048{0} Drive-Parameter 'P-0-0008, Activation E-Stop function' with '0'= control Id, '22'= drive address, '{1}' = only this bit is toggled: 0,22,P-0-0008{1}

6.24 Triangle

6.24.1 General

Type	Type of the element
Id	Id of the element
Alias	Unique widget identifier

6.24.2 Position

X position	Distance from the left edge in pixels
Y position	Distance from the top edge in pixels
Width	Width of the element in pixel, including border
Height	Height of the element in pixel, including border
Rotation	Rotation of the element about its center in degrees

6.24.3 Style

Fill color	Background fill color
Border color	The border color, it can only be selected if no theme is selected.
Border width	Border width in pixel
Transparency	A value between 0 and 100, 0: the object appears completely transparent, 100: the object completely covers the background
Shadow	Show a drop shadow

6.24.4 Text properties

Text	Entered text will be shown. If a variable is entered in the 'Variable' attribute of the 'Text properties', the value of the linked variable will be shown respectively inserted if one of the below explained formatters is used inside of the text. printf formatters, %[width][.precision]specifier. Specifier: %b, yields an integer as a binary number %c, yields an integer as the character with that ASCII value %d or %i, yields an integer as a signed decimal number %e, yields a float using scientific notation %u, yields an integer as an unsigned decimal number %f, yields a float as is; see example for precision %o, yields an integer as an octal number %s, yields a string as is %x, yields an integer as a hexadecimal number (lower-case) %X, yields an integer as a hexadecimal number (upper-case) %j, yields a JavaScript object or array as a JSON encoded string e.g. The color: %s, value=blue => The color: blue First number: %d, value=12345 => First number: 12345 Second number: %04d, value=25 => Second number: 0025 Third number: %i, value=1234 => Third number: 1234 Float number: %3.2f, value=3.14159 => Float number: 3.14 Hexadecimal: %x, value=255 => Hexadecimal: ff Octal: %o, value=255 => Octal: 377 Unsigned value: %u, value=150 => Unsigned value: 150 html tags can also be used for formatting purposes bold text
 line break ordered list list item important text e.g. This is normal text - and this is bold text, defines bold text areas My text in first line
 my text in second line, defines a single line break first line second line third line, defines an ordered list see also:
------	---

Variable	The linked variable, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.BLINK globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive paramters is normaly '0'
Font color	The color of the font, it can only be selected if no theme is selected.
Font size	Font size can only be selected if no theme is selected.
Font Name	Font Name can only be selected if no theme is selected.
Custom Filter	Name of filter function to format the displayed value.
Variable Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
Tooltip	Text that is displayed when the mouse pointer is over the item. Be aware that in case of a touch device the tooltip can't be displayed.
Update interval	Length of the update rate The following update rates are available: 'none': Value is only read once when screen is opened 'slowest': 10000ms 'slow': 5000ms 'default': 1000ms 'fast': 500ms 'fastest': 200ms Be aware that a fast update rate increases the performance load of the application and the control.

6.24.5 Dynamic fill color

Variable for Dynamic FillColor	The linked variable, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.BLINK globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive paramters is normaly '0'
Variable Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be checked against the operators and statments of the defined fill colors. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
No of Fill Color	Number of fill color for the widget. Minimum Value : 1 Maximum Value : 16 First match will be taken and rest is ignored. E.g. if "Operator" and "Statement" are matching for fill color 3, then the statements for all following defined fill colors (4,5,...) will be ignored. Hint: Only 'Range' operator (x to Y) is possible if number of fill colors is more then 1

6.24.6 Dynamic movements

Position X	Movement of widget along X - absolute: from left edge of the screen - relative: from the original placement of the widget
Variable X	Linked variable controls the movement of the element in X-direction, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive parameters is normally '0'
Variable X Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
Position Y	Movement of widget along Y - absolute: from left edge of the screen - relative: from the original placement of the widget

Attachment: Widget attribute

Variable Y	Linked variable controls the movement of the element in Y-direction, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive paramters is normaly '0'
Variable Y Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".

6.24.7 Dynamic size

Variable width	Linked variable controls the width of the object, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0047, Position command value' with '0'= control Id, '22'= drive address: 0,22,S-0-0047 Hint: Control Id for drive paramters is normaly '0'
Variable width script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".

Attachment: Widget attribute

Variable height	Linked variable controls the height of the object, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0047, Position command value' with '0'= control Id, '22'= drive address: 0,22,S-0-0047 Hint: Control Id for drive paramters is normaly '0'
Variable height script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".

6.24.8 Dynamic rotation

Variable for Dynamic Rotation	Linked variable controls the rotation of the element, which is either entered or selected via the process variable browser. e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.NestColor[1] JavaScript variables: internalVariables.RANDOMFLOAT globalVariables.test screenVariables.test2 Control-Parameter 'C-0-0080, User defined decimal number': C-0-0080 Axis-Parameter 'A-0-0100, Actual position value' with '0'= control Id, '3'= axis address: 0,3,A-0-0100 Drive-Parameter 'S-0-0141, Motor type' with '0'= control Id, '22'= drive address: 0,22,S-0-0141 Hint: Control Id for drive paramters is normaly '0'
Variable Script	JavaScript code to process the value of the linked variable when it is read. Script execution starts when the read value changes! Use 'value' to access the actual value of the linked variable. e.g. value=value-50, if 100 is the read value of a PLC variable, then result is 50 --> 50 will be processed. screenVariables.Sum = value * 3.5 + screenVariables.Offset, if 20 is the read value of a control parameter and Offset is 10 then result is 80 --> 80 is stored in the screen variable "Sum".
Rotation base	Specifies the point, where the rotation is based of

6.24.9 Events

OnClick	The event is executed if a full click is performed on the element, ie. a mouse press and a mouse release
OnMouseUp	The event is executed if a mouse button is released on this element
OnMouseDown	The event is executed if a mouse button is pressed on this element
OnMouseOver	The event is fires when when the mouse cursor enters the element
OnMouseOut	The event is fires when the mouse cursor leaves the element
OnRightClick	This event fires when the right mouse button is pressed
OnDoubleClick	This event fires on double mouse click

Attachment: Widget attribute

Push value	The linked variable, which is either entered or selected via the process variable browser, is set to TRUE (or FALSE) as long as the mouse is getting pressed. This is a convenience functionality and usually used with the "Button" widget and useful to set bool variables or single bits of an integer (use {bit} to set a single bit of integers). Depending on the 'Push value FALSE' attribute, either TRUE or FALSE is set when the button is pressed. Be aware that this feature used with active widgets (like "Input field" or "Chechbox") can cause problems when the same linked variable of the text properties is used because both try to write to the same variable independently! e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.myBool ns=2;s=Application.globalVAR.myInt{7}, with '{7}' = only this bit is toggled JavaScript variables: globalVariables.myBool screenVariables.myInt{14}, with '{14}' = only this bit is toggled Control-Parameter 'C-0-0080, User defined decimal number', with '{4}' = only this bit is used: C-0-0080{4} Axis-Parameter 'A-0-0048, Velocity polarities' with '0'= control Id, '3'= axis address, '{0}' = only this bit is toggled: 0,3,A-0-0048{0} Drive-Parameter 'P-0-0008, Activation E-Stop function' with '0'= control Id, '22'= drive address, '{1}' = only this bit is toggled: 0,22,P-0-0008{1} Hint: Control Id for drive paramters is usually '0'
------------	---

Push value FALSE	If set to 'true', the 'Push value' written to the linked variable as long as the mouse is getting pressed is FALSE. Attribute is 'false' (default) --> TRUE is written to the linked variable as long as the mouse is getting pressed Attribute is 'true' --> FALSE is written to the linked variable as long as the mouse is getting pressed
Toggle	The linked variable, which is either entered or selected via the process variable browser, is toggled between TRUE and FALSE when clicked. This is a convenience functionality and only useful to toggle bool variables or single bits of an integer (use {bit} to toggle a single bit of integers). Be aware that this feature used with active widgets (like "Input field" or "Checkbox") can cause problems when the same linked variable of the text properties is used because both try to write to the same variable independently! e.g. OPC-UA Items like a PLC variable: ns=2;s=Application.globalVAR.myBool ns=2;s=Application.globalVAR.myInt{7}, with '{7}' = only this bit is toggled JavaScript variables: globalVariables.myBool screenVariables.myInt{14}, with '{14}' = only this bit is toggled Control-Parameter 'C-0-0080, User defined decimal number', with '{4}' = only this bit is used: C-0-0080{4} Axis-Parameter 'A-0-0048, Velocity polarities' with '0'= control Id, '3'= axis address, '{0}' = only this bit is toggled: 0,3,A-0-0048{0} Drive-Parameter 'P-0-0008, Activation E-Stop function' with '0'= control Id, '22'= drive address, '{1}' = only this bit is toggled: 0,22,P-0-0008{1}

7 Service and support

Our worldwide service network provides an optimized and efficient support. Our experts offer you advice and assistance should you have any queries. You can contact us **24/7**.

Service Germany Our technology-oriented Competence Center in Lohr, Germany, is responsible for all your service-related queries for electric drive and controls.

Contact the **Service Hotline** and **Service Helpdesk** under:

Phone:	+49 9352 40 5060
Fax:	+49 9352 18 4941
E-mail:	service.svc@boschrexroth.de
Internet:	http://www.boschrexroth.com

Additional information on service, repair (e.g. delivery addresses) and training can be found on our internet sites.

Service worldwide Outside Germany, please contact your local service office first. For hotline numbers, refer to the sales office addresses on the internet.

Preparing information To be able to help you more quickly and efficiently, please have the following information ready:

- Detailed description of malfunction and circumstances
- Type plate specifications of the affected products, in particular type codes and serial numbers
- Your contact data (phone and fax number as well as your e-mail address)

Index

A

About this documentation.....	1
Validity of the documentation.....	1
ANSI Z535.6-2006.....	4

C

Complaints.....	6
Criticism.....	6
Customer Feedback.....	6

D

Documentation	
Change record.....	1

F

Feedback.....	6
---------------	---

H

Hazard warnings.....	4
Helpdesk.....	309
Hotline.....	309

I

Important instructions on use.....	7
Intended use.....	7
Information representation	
Names and abbreviations.....	6
Intended use	
Areas of use and application.....	7
Introduction.....	7
Intended Use	
Areas of use and application.....	7

S

Safety instructions.....	4
Service hotline.....	309
Signal alert symbol.....	4
Signal words.....	4
Suggestions.....	6
Support.....	309
Symbols.....	6

T

Target groups.....	1
--------------------	---

U

Unintended use	
Introduction.....	7
Unintended Use	
Consequences, non-liability.....	7

W

Warnings.....	4
---------------	---

Notes

Notes

Bosch Rexroth AG

Electric Drives and Controls

P.O. Box 13 57

97803 Lohr, Germany

Bgm.-Dr.-Nebel-Str. 2

97816 Lohr, Germany

Phone +49 9352 18 0

Fax +49 9352 18 8400

www.boschrexroth.com/electrics



R911376836