

BECKHOFF New Automation Technology

Manual | EN

TF8550

TwinCAT 3 | Plastic HMI Framework

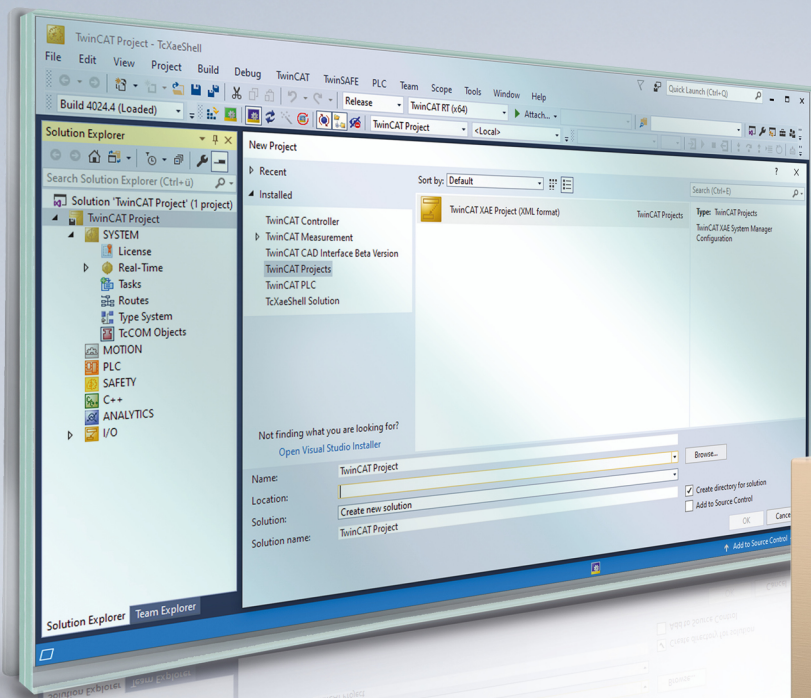


Table of contents

1 Foreword	7
1.1 Notes on the documentation	7
1.2 For your safety	7
1.3 Notes on information security.....	9
2 Overview	10
3 Installation	11
3.1 System requirements	11
3.2 Licensing	11
3.3 Installation	13
4 Controls	14
4.1 ArrowMotionGraph	16
4.1.1 Attributes	16
4.1.2 Functions.....	22
4.1.3 User Interactions	24
4.2 BlowPressureChart	24
4.2.1 Attributes	25
4.3 CamControl	29
4.3.1 Attributes	30
4.3.2 Functions.....	33
4.3.3 Events	34
4.4 Configurator	34
4.4.1 Attributes	35
4.4.2 Unit_Config.json.....	40
4.5 CurveEditor	42
4.5.1 Attributes	43
4.5.2 Functions.....	59
4.5.3 Events	65
4.5.4 User Interactions	66
4.5.5 Step by step	68
4.5.6 Using Recipe Management with the Curve Editor	69
4.6 FavoriteBarControl	71
4.6.1 Attributes	71
4.6.2 User Interactions	75
4.7 InputBox	76
4.7.1 Attributes	76
4.7.2 Functions.....	81
4.7.3 Events	82
4.7.4 Step by step	83
4.8 ManualOperation.....	85
4.8.1 Attributes	86
4.9 MeasurementUnitSelector.....	95
4.9.1 Attributes	95
4.9.2 User Interactions	98

4.10	PfwSingleTempControl	98
4.10.1	Attributes	99
4.11	PfwTempParameters	102
4.11.1	Attributes	104
4.12	ProcessScheduler	105
4.12.1	Attributes	105
4.12.2	Functions	107
4.12.3	Events	108
4.12.4	Dialogs	109
4.13	StateIndicator	110
4.13.1	Attributes	110
4.14	Table	112
4.14.1	Attributes	113
4.14.2	Events	120
4.14.3	Step by step	120
4.15	TimerControl	123
4.15.1	Attributes	124
4.16	VariableViewControl	126
4.16.1	Attributes	127
4.16.2	Dialogs	135
4.17	Monitoring	135
4.17.1	Attributes	137
4.18	Functions	143
4.18.1	UnitConversion	143
5	RecipeHelper	149
5.1	Concept	149
5.1.1	Generated recipe types and recipes	149
5.1.2	Temporary recipes	150
5.1.3	Automatic mapping of symbols and adding to recipe types	153
5.1.4	Handshake with PLC	153
5.1.5	On screen error messages	156
5.2	Functions	157
5.2.1	SaveActiveRecipe	157
5.2.2	RcpAutoUpdate	158
5.2.3	ListActDataTypeRecipes	158
5.2.4	LoadProductRecipe	159
5.2.5	DeleteActRcp	160
5.2.6	ExportRecipe	161
5.2.7	ImportRecipe	161
5.3	Using RecipeHelper with HMI controls	162
5.3.1	Listing Recipes on a data grid	162
5.3.2	Display active recipe with unsaved changes	163
6	Functions	164
6.1	Axes	164
6.1.1	ChangelnputBoxUnit	164

6.1.2	UpdateAxesHomingUnits	164
6.2	CodeBehind	165
6.2.1	DisableContextMenu	165
6.2.2	SlidingFunction	166
6.3	Frame	166
6.3.1	HideSliderArea	166
6.3.2	ShowSliderArea	166
6.3.3	ToggleSliderArea	167
6.3.4	TakeScreenShot	167
6.3.5	UpdateSliderContentRegion	168
6.3.6	UpdateSliderContentRegionEx	170
6.4	General	171
6.4.1	CreateBinding	171
6.4.2	SetSymbolArrayIdx	171
6.5	Parameters	172
6.5.1	UpdateParametersSettingsControls	172
6.5.2	UpdateParametersSettingsControlsEx	175
6.5.3	UpdateTemperatureParametersTableNumber	177
6.5.4	UpdateTemperatureParametersTableNumberEx	177
6.5.5	FillParamTable	178
6.6	Scope	178
6.6.1	UpdateScope	178
6.7	Trend	179
6.7.1	ResetTrendAxisColors	179
6.7.2	UpdateTrend	180
6.7.3	UpdateTrendAxisColors	180
7	Images	182
8	Localizations	187
9	Themes	188
10	Temperature	189
10.1	ZoneGrouping	189
10.1.1	Attributes	189
10.1.2	Functions	191
10.1.3	User Interactions	192
10.1.4	Dialogs	193
10.2	ZoneImageLayout	194
10.2.1	Attributes	196
10.2.2	Functions	199
10.2.3	User Interactions	200
10.2.4	Dialogs	206
10.2.5	Events	207
10.2.6	Step by step	208
10.3	ZoneConfiguration	208
10.3.1	Attributes	210
10.3.2	User Interactions	212

1 Foreword

1.1 Notes on the documentation

This description is intended exclusively for trained specialists in control and automation technology who are familiar with the applicable national standards.

For installation and commissioning of the components, it is absolutely necessary to observe the documentation and the following notes and explanations.

The qualified personnel is obliged to always use the currently valid documentation.

The responsible staff must ensure that the application or use of the products described satisfies all requirements for safety, including all the relevant laws, regulations, guidelines, and standards.

Disclaimer

The documentation has been prepared with care. The products described are, however, constantly under development.

We reserve the right to revise and change the documentation at any time and without notice.

No claims to modify products that have already been supplied may be made on the basis of the data, diagrams, and descriptions in this documentation.

Trademarks

Beckhoff®, TwinCAT®, TwinCAT/BSD®, TC/BSD®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® and XPlanar® are registered and licensed trademarks of Beckhoff Automation GmbH.

If third parties make use of designations or trademarks used in this publication for their own purposes, this could infringe upon the rights of the owners of the said designations.

Patents

The EtherCAT Technology is covered, including but not limited to the following patent applications and patents:

EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702
and similar applications and registrations in several other countries.

EtherCAT®

EtherCAT® is registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany

Copyright

© Beckhoff Automation GmbH & Co. KG, Germany.

The distribution and reproduction of this document as well as the use and communication of its contents without express authorization are prohibited.

Offenders will be held liable for the payment of damages. All rights reserved in the event that a patent, utility model, or design are registered.

1.2 For your safety

Safety regulations

Read the following explanations for your safety.

Always observe and follow product-specific safety instructions, which you may find at the appropriate places in this document.

Exclusion of liability

All the components are supplied in particular hardware and software configurations which are appropriate for the application. Modifications to hardware or software configurations other than those described in the documentation are not permitted, and nullify the liability of Beckhoff Automation GmbH & Co. KG.

Personnel qualification

This description is only intended for trained specialists in control, automation, and drive technology who are familiar with the applicable national standards.

Signal words

The signal words used in the documentation are classified below. In order to prevent injury and damage to persons and property, read and follow the safety and warning notices.

Personal injury warnings**⚠ DANGER**

Hazard with high risk of death or serious injury.

⚠ WARNING

Hazard with medium risk of death or serious injury.

⚠ CAUTION

There is a low-risk hazard that could result in medium or minor injury.

Warning of damage to property or environment**NOTICE**

The environment, equipment, or data may be damaged.

Information on handling the product

This information includes, for example:
recommendations for action, assistance or further information on the product.

1.3 Notes on information security

The products of Beckhoff Automation GmbH & Co. KG (Beckhoff), insofar as they can be accessed online, are equipped with security functions that support the secure operation of plants, systems, machines and networks. Despite the security functions, the creation, implementation and constant updating of a holistic security concept for the operation are necessary to protect the respective plant, system, machine and networks against cyber threats. The products sold by Beckhoff are only part of the overall security concept. The customer is responsible for preventing unauthorized access by third parties to its equipment, systems, machines and networks. The latter should be connected to the corporate network or the Internet only if appropriate protective measures have been set up.

In addition, the recommendations from Beckhoff regarding appropriate protective measures should be observed. Further information regarding information security and industrial security can be found in our <https://www.beckhoff.com/secguide>.

Beckhoff products and solutions undergo continuous further development. This also applies to security functions. In light of this continuous further development, Beckhoff expressly recommends that the products are kept up to date at all times and that updates are installed for the products once they have been made available. Using outdated or unsupported product versions can increase the risk of cyber threats.

To stay informed about information security for Beckhoff products, subscribe to the RSS feed at <https://www.beckhoff.com/secinfo>.

2 Overview

The TF8550 Plastic TC HMI controls product, along with its corresponding NuGet packages, offers a range of new controls, functions, code behind scripts, images, and a specialized theme that are tailored to the specific needs of plastic machines HMI projects.

These additions are designed to simplify and accelerate the process of setting up an HMI project for plastic applications and making it easier for users to create customized and effective solutions.

NuGet packages of TF8550

Package Name	Description
Beckhoff.TwinCAT.HMI.Plastic.Controls [► 14]	This package provides new controls for TwinCAT HMI projects.
Beckhoff.TwinCAT.HMI.Plastic.Functions [► 164]	This package provides functions that are used in TF85XX Plastic Application HMI projects.
Beckhoff.TwinCAT.HMI.Plastic.Images [► 182]	This package adds icons and images that can be used for plastic HMI applications.
Beckhoff.TwinCAT.HMI.Plastic.Localizations [► 187]	This package adds localized symbols that can be used in plastic HMI applications.
Beckhoff.TwinCAT.HMI.Plastic.RecipeHelper [► 149]	This package provides functions and scripts that add a preconfigured recipe structure and allow automated saving of values from the controls in the recipe.
Beckhoff.TwinCAT.HMI.Plastic.Temperature [► 14]	This package provides special controls and functions for temperature related visualization in plastic HMI applications.
Beckhoff.TwinCAT.HMI.Plastic.Themes [► 188]	This package adds a preconfigured plastic theme to style standard TwinCAT HMI controls in the project.

3 Installation

3.1 System requirements

The package Beckhoff.TwinCAT.HMI.Plastic is available as a NuGet package and can be used with a TC3 TwinCAT HMI projects created with TE2000 TC3 HMI Engineering.

Minimum version of TE2000 TC3 HMI Engineering is 1.12.742.x

See here for more information of [minimum system requirements for TE2000 TC3 HMI Engineering](#).

3.2 Licensing

The TwinCAT 3 function can be activated as a full version or as a 7-day test version. Both license types can be activated via the TwinCAT 3 development environment (XAE).

Licensing the full version of a TwinCAT 3 Function

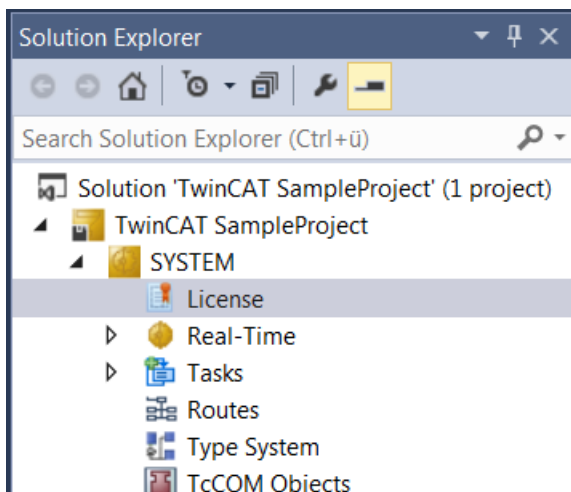
A description of the procedure to license a full version can be found in the Beckhoff Information System in the documentation "[TwinCAT 3 Licensing](#)".

Licensing the 7-day test version of a TwinCAT 3 Function



A 7-day test version cannot be enabled for a [TwinCAT 3 license dongle](#).

1. Start the TwinCAT 3 development environment (XAE).
2. Open an existing TwinCAT 3 project or create a new project.
3. If you want to activate the license for a remote device, set the desired target system. To do this, select the target system from the **Choose Target System** drop-down list in the toolbar.
 - ⇒ The licensing settings always refer to the selected target system. When the project is activated on the target system, the corresponding TwinCAT 3 licenses are automatically copied to this system.
4. In the **Solution Explorer**, double-click **License** in the **SYSTEM** subtree.



⇒ The TwinCAT 3 license manager opens.

- Open the **Manage Licenses** tab. In the **Add License** column, check the check box for the license you want to add to your project (e.g. "TF4100 TC3 Controller Toolbox").

Order No	License	Add License
TF3601	TC3 Condition Monitoring Level 2	<input type="checkbox"/> cpu license
TF3650	TC3 Power Monitoring	<input type="checkbox"/> cpu license
TF3680	TC3 Filter	<input type="checkbox"/> cpu license
TF3800	TC3 Machine Learning Inference Engine	<input type="checkbox"/> cpu license
TF3810	TC3 Neural Network Inference Engine	<input type="checkbox"/> cpu license
TF3900	TC3 Solar-Position-Algorithm	<input type="checkbox"/> cpu license
TF4100	TC3 Controller Toolbox	<input checked="" type="checkbox"/> cpu license
TF4110	TC3 Temperature-Controller	<input type="checkbox"/> cpu license
TF4500	TC3 Speech	<input type="checkbox"/> cpu license

- Open the **Order Information (Runtime)** tab.
 - ⇒ In the tabular overview of licenses, the previously selected license is displayed with the status "missing".
- Click **7-Day Trial License...** to activate the 7-day trial license.

Order Information (Runtime) | Manage Licenses | Project Licenses | Online Licenses

License Device: Target (Hardware Id) [Add...]

System Id: 2DB25408-B4CD-81DF-5488-6A3D9B49EF19 | Platform: other (91)

License Request

Provider: Beckhoff Automation [Generate File...]

License Id: [] | Customer Id: []

Comment: []

License Activation

7 Days Trial License... | License Response File...

- ⇒ A dialog box opens, prompting you to enter the security code displayed in the dialog.

Enter Security Code [X]

Please type the following 5 characters:

Kg8T4

[] | []

OK | Cancel

- Enter the code exactly as it is displayed and confirm the entry.
- Confirm the subsequent dialog, which indicates the successful activation.
 - ⇒ In the tabular overview of licenses, the license status now indicates the expiry date of the license.

10. Restart the TwinCAT system.

⇒ The 7-day trial version is enabled.

3.3 Installation

1. Beckhoff.TwinCAT.HMI.Plastic package is available as a NuGet package file.
2. Check for more information about NuGet [package management](#) in TwinCAT HMI projects.
3. Copy the Beckhoff.TwinCAT.HMI.Plastic.nuget file to one of the configured NuGet repository.
4. This will make the package available for installation in any TwinCAT HMI project. Install the package using NuGet package manager.
5. Once package is installed, all controls, function and other resources will be available in your project.

4 Controls

NuGet package: Beckhoff.TwinCAT.HMI.Plastic.Controls

Available: since version 12.6.0

The NuGet package **Beckhoff.TwinCAT.HMI.Plastic.Controls** can be installed in the TwinCAT HMI project via the package management system NuGet.

Name	Description
ArrowMotionGraph [▶ 16]	ArrowMotionGraph control allows the user to configure segments of an axis move. Positions and velocities for different segments are visualized as arrows on the control.
BlowPressureChart [▶ 24]	BlowPressureChart is a special chart type control that can visualize the air blowing process in a blow molding machine.
CamControl [▶ 29]	CamControl is used to visualize and configure an array of cam switches. Each cam switch is defined by hysteresis and threshold values.
Configurator [▶ 34]	The Configurator control stores user settings for the default input panel that appears when the user enters a new value on controls provided in the Beckhoff.TwinCAT.HMI.Plastic.Controls package. It also stores measurement unit configuration, supplies functions to access this configuration and perform unit conversion calculations.
CurveEditor [▶ 42]	The CurveEditor control provides users with the ability to add and manipulate a list of points on their HMI screen. Then the control can calculate an interpolated curve that passes through each point.
FavoriteBarControl [▶ 71]	The FavoriteBarControl can be used with the Responsive Navigation Bar control in TwinCAT HMI. The purpose of this control is to allow users to easily set any content as a favorite with creating a shortcut that is always available on the FavoriteBarControl.
InputBox [▶ 76]	InputBox is a control that allows the user to input numerical or Boolean values in the TwinCAT HMI project.
ManualOperation [▶ 85]	ManualOperation control can be used as on-screen touch buttons for moving an axes manually.
MeasurementUnitSelector [▶ 95]	With using the MeasurementUnitSelector control the user can access the unit configuration configured in the Unit_Config.json file and change the display unit settings on the client.
Monitoring [▶ 135]	Monitoring control is used to visualize and configure multiple FB_Monitoring objects in the PLC from a single framework control.
PfwSingleTempControl [▶ 98]	PfwSingleTempControl allows the user to easily display and control a temperature zone in TwinCAT HMI.
ProcessScheduler [▶ 105]	The ProcessScheduler control will allow the user to visualize and edit weekly schedules that can be used by the PLC to execute any process.
StateIndicator [▶ 110]	StateIndicator control is showing different states and can be used as an LED indicator
Table [▶ 112]	Table control can be used to visualize multiple PLC variables in a paginated table format.
TimerControl [▶ 123]	The TimerControl can be used to visualize and configure an FB_Timer symbol.
VariableViewControl [▶ 126]	VariableViewControl can be used as a dashboard to visualize live values from multiple variables on the screen.

NuGet package: Beckhoff.TwinCAT.HMI.Plastic.Temperature

Note: Available since version 12.6.0

The NuGet package **Beckhoff.TwinCAT.HMI.Plastic.Temperature** can be installed in the TwinCAT HMI project via the package management system NuGet.

Name	Description
ZoneConfiguration [▶ 208]	ZoneConfiguration control is used to visualize and allow users to change different operational parameters of the temperature groups and zones configured in the PLC.
ZonedImageLayout [▶ 194]	The ZonedImageLayout framework control allows users to present temperature zones over an image background in a layout that mimics the physical arrangement on the machine.
ZoneGrouping [▶ 189]	ZoneGrouping control is used to visualize and allow users to change temperature group configuration in the Tc3 Plastic Application project.

4.1 ArrowMotionGraph

NuGet package: Beckhoff.TwinCAT.HMI.Plastic.Controls

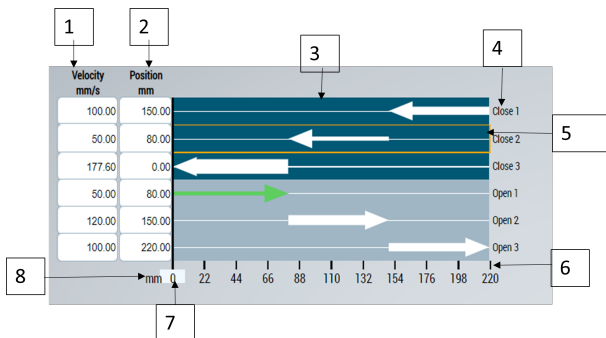
Available: since version 12.6.0

With the ArrowMotionGraph control users can access configurations of different segments of axis moves. Positions and velocities for different segments in every move are visualized as arrows on the control; the width of every arrow is proportional to the velocity of its segment.

● ActivateConfig action



Any new changes to the control must be activated using the [ActivateConfig](#) [▶ 22] action to set new values to the PLC.



1. Sets the velocity for the axis segments
2. Sets the position for the axis segments
3. Axis movements segments visualization
4. Segment names
5. Selected segment highlighted
6. Linear scale
7. Actual axis position
8. Scale unit

4.1.1 Attributes

Category: Configuration

Name	Description
Axis [▶ 17]	Symbol for axis fb.
ColumnWidths [▶ 19]	Width for each of 4 columns.
RowHeight [▶ 20]	Row height of each displayed row.
SegmentNames [▶ 20]	Segment label text for each move segment.
FirstMoveDirection [▶ 20]	Sets the axis movement direction for the first move.
Activation Required [▶ 17]	Is set to <code>TRUE</code> if user makes new changes that are not activated.
AxisUnitTypes [▶ 19]	List of unit types that can be used in the control.
RowHeight Unit [▶ 20]	Sets the unit of the RowHeight property always in pixel.
InvertScale [▶ 22]	Sets direction of scale on the control.
CurrentValue [▶ 22]	The actual position of the linked axis.
DigitAfterDecimal [▶ 22]	Decimal precision of values displayed on the control.

Category: Selection

Name	Description
SelectedRowSegFunction [▶ 21]	Segment function property for the selected segment.
DisplaySubPage [▶ 19]	Allows switching between the Target , Dynamic or Limiting subpages.
SelectedMoveStartFunction [▶ 21]	Moves start function property of selected move.
SelectedMoveEndFunction [▶ 21]	Moves end function property of selected move.
SelectedMovIndex [▶ 21]	The selected move index.
SelectedSegIndex [▶ 21]	The selected segment index.

4.1.1.1 Activation Required

Is set to `TRUE` if user makes new changes that are not activated.

Schema: tchmi:general##/definitions/Boolean

Attribute getter: getActivationRequired

Available since version 12.6.0

4.1.1.2 Axis

Symbol of fbAxis that will be displayed and configured using this control. The axis symbol binding at this attribute must have a variable structure as explained below.

The top level structure must have the properties listed below:

Name	Type	Description
ActPosition	Number	Current position of the axis
ActiveMove	Integer	Index of the current move
ActiveSegment	Integer	Index of the segment that is active
MaxAcceleration	Number	Maximum acceleration
MaxDeceleration	Number	Maximum deceleration
MaxPosition	Number	Position max limit
MaxVelocity	Number	Velocity max limit
MinPosition	Number	Position lower limit
MinVelocity	Number	Velocity lower limit
fbAxisMove	ARRAY [1..2] of 'Move Config'	Array of moves

PLC1.MAIN.fb_AMG	PLC1.FB_AMGTest	<input checked="" type="checkbox"/>	PLC1::MAIN::fb_AMG
ActPosition	PLC1.PROPERTY-GET-SET-LREAL	<input checked="" type="checkbox"/>	ActPosition
ActiveMove	PLC1.PROPERTY-GET-SET-INT	<input checked="" type="checkbox"/>	ActiveMove
ActiveSegment	PLC1.PROPERTY-GET-SET-INT	<input checked="" type="checkbox"/>	ActiveSegment
MaxAcceleration	PLC1.PROPERTY-GET-SET-LREAL	<input checked="" type="checkbox"/>	MaxAcceleration
MaxDeceleration	PLC1.PROPERTY-GET-SET-LREAL	<input checked="" type="checkbox"/>	MaxDeceleration
MaxPosition	PLC1.PROPERTY-GET-SET-LREAL	<input checked="" type="checkbox"/>	MaxPosition
MaxVelocity	PLC1.PROPERTY-GET-SET-LREAL	<input checked="" type="checkbox"/>	MaxVelocity
MinPosition	PLC1.PROPERTY-GET-SET-LREAL	<input checked="" type="checkbox"/>	MinPosition
MinVelocity	PLC1.PROPERTY-GET-SET-LREAL	<input checked="" type="checkbox"/>	MinVelocity
SetPosition	PLC1.PROPERTY-GET-SET-LREAL	<input checked="" type="checkbox"/>	SetPosition
fbAxisMove	PLC1.FB_AxisMoveHmi	<input checked="" type="checkbox"/>	fbAxisMove

Each move configuration structure or function block must have the properties listed below:

Name	Type	Description
EndFunction	Integer	The move end function value.
StartFunction	Integer	The move start function value.
fbSegments	ARRAY [1..5] of 'Segment Config'	The array of move segments.

fbAxisMove	PLC1.FB_AxisMoveHmi	<input checked="" type="checkbox"/>	fbAxisMove
fbMoves	PLC1.ARRAY_1..2_OF-FB_MoveHmiCfg	<input checked="" type="checkbox"/>	fbMoves
[0]	PLC1.FB_MoveHmiCfg	<input checked="" type="checkbox"/>	[0]
EndFunction	PLC1.PROPERTY-GET-SET-INT	<input checked="" type="checkbox"/>	EndFunction
StartFunction	PLC1.PROPERTY-GET-SET-INT	<input checked="" type="checkbox"/>	StartFunction
fbSegments	PLC1.ARRAY_1..5_OF-FB_SegHmiCfg	<input checked="" type="checkbox"/>	fbSegments
[1]	PLC1.FB_MoveHmiCfg	<input checked="" type="checkbox"/>	[1]

Each segment configuration structure or function block must have the properties listed below:

Name	Type	Description
Acceleration	Number	Set acceleration value for the segment
Deceleration	Integer	Set deceleration value for the segment
InUse	BOOL	This property is set to FALSE if any segment is not used by the current configuration.
Position	Number	Set position value for the segment
SegFunction	Integer	Segment function value for the segment
Velocity	Number	Set velocity value for the segment

fbAxisMove	PLC1.FB_AxisMoveHmi	<input checked="" type="checkbox"/>	fbAxisMove
fbMoves	PLC1.ARRAY_1..2_OF-FB_MoveHmiCfg	<input checked="" type="checkbox"/>	fbMoves
[0]	PLC1.FB_MoveHmiCfg	<input checked="" type="checkbox"/>	[0]
EndFunction	PLC1.PROPERTY-GET-SET-INT	<input checked="" type="checkbox"/>	EndFunction
StartFunction	PLC1.PROPERTY-GET-SET-INT	<input checked="" type="checkbox"/>	StartFunction
fbSegments	PLC1.ARRAY_1..5_OF-FB_SegHmiCfg	<input checked="" type="checkbox"/>	fbSegments
[0]	PLC1.FB_SegHmiCfg	<input checked="" type="checkbox"/>	[0]
Acceleration	PLC1.PROPERTY-GET-SET-LREAL	<input checked="" type="checkbox"/>	Acceleration
Deceleration	PLC1.PROPERTY-GET-SET-LREAL	<input checked="" type="checkbox"/>	Deceleration
InUse	PLC1.PROPERTY-GET-SET-BOOL	<input checked="" type="checkbox"/>	InUse
Position	PLC1.PROPERTY-GET-SET-LREAL	<input checked="" type="checkbox"/>	Position
SegFunction	PLC1.PROPERTY-GET-SET-INT	<input checked="" type="checkbox"/>	SegFunction
Velocity	PLC1.PROPERTY-GET-SET-LREAL	<input checked="" type="checkbox"/>	Velocity
[1]	PLC1.FB_SegHmiCfg	<input checked="" type="checkbox"/>	[1]
[2]	PLC1.FB_SegHmiCfg	<input checked="" type="checkbox"/>	[2]
[3]	PLC1.FB_SegHmiCfg	<input checked="" type="checkbox"/>	[3]
[4]	PLC1.FB_SegHmiCfg	<input checked="" type="checkbox"/>	[4]

Schema: tchmi:framework#/definitions/Symbol

Attribute getter: getAxis

Attribute setter: setAxis

Available since version 12.6.0

4.1.1.3 AxisUnitTypes

List of unit types that can be used in the control.

Schema: tchmi:framework#/definitions/AMGAxisUnits

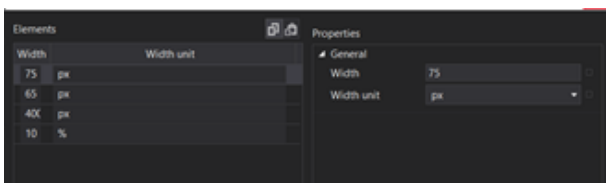
Attribute getter: getAxisUnitTypes

Attribute setter: setAxisUnitTypes

Available since version 12.6.0

4.1.1.4 ColumnWidths

Width for each of the 4 columns. The widths can be set in pixels or percent unit.



Schema: tchmi: framework#/definitions/AMG_ColWidthList

Attribute getter: getColumnWidths

Attribute setter: setColumnWidths

Available since version 12.6.0

4.1.1.5 DisplaySubPage

This attribute allows the user to switch between the **Target**, **Dynamic** or **Limiting** subpage of the control.

- The **Target** subpage shows velocity and position values.
- The **Dynamic** subpage shows acceleration and deceleration values for each segment.
- In **Limiting** subpage user can see jerk and limiting values for each segment.

Schema: tchmi:general#/definitions/DisplaySubPage

Attribute getter: getDisplaySubPage

Attribute setter: setDisplaySubPage

Available since version 12.6.0

4.1.1.6 FirstMoveDirection

This attribute sets the direction of the first move on the control. There are two directions that can be set in this attribute:

- In the direction **TowardsZero**, the first move arrows will start from the higher limit of the axis.
- In the direction **AwayFromZero**, the first move arrows will start from the minimum value and point towards the higher limit of the axis.

Schema: tchmi:framework#/definitions/AMGMoveDirection

Attribute getter: getFirstMoveDirection

Attribute setter: setFirstMoveDirection

Available since version 12.6.0

4.1.1.7 RowHeight

Row height of each displayed row.

Schema: tchmi:general#/definitions/MeasurementValue

Attribute getter: getRowHeight

Attribute setter: setRowHeight

Available since version 12.6.0

4.1.1.8 RowHeight Unit

Sets the unit of the RowHeight property always in pixel.

Schema: tchmi:framework#/definitions/PixelUnit

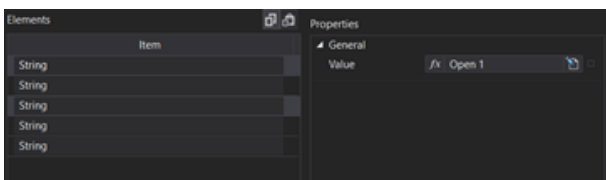
Attribute getter: getRowHeightUnit

Attribute setter: setRowHeightUnit

Available since version 12.6.0

4.1.1.9 SegmentNames

Segment label text for each move segment. The array or string can be configured with localization keys.



Schema: tchmi:framework#/definitions/SegNames

Attribute getter: getSegmentNames

Attribute setter: setSegmentNames

Available since version 12.6.0

4.1.1.10 SelectedMoveEndFunction

Shows the **EndFunction** property for the selected axis move.

If a non-zero value is set, a circular element is added at the end point of the move.



Schema: tchmi:general#/definitions/INT

Attribute getter: getSelectedMoveEndFunction

Attribute setter: setSelectedMoveEndFunction

Available since version 12.6.0

4.1.1.11 SelectedMoveStartFunction

Shows the **StartFunction** property for the selected axis move.

If a non-zero value is set, a circular element is added at the start of the move.



Schema: tchmi:general#/definitions/INT

Attribute getter: getSelectedMoveStartFunction

Attribute setter: setSelectedMoveStartFunction

Available since version 12.6.0

4.1.1.12 SelectedMovIndex

The selected move index.

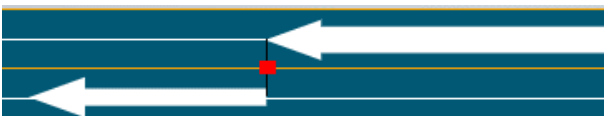
Schema: tchmi:general#/definitions/UINT

Attribute getter: getSelectedMovIndex

Available since version 12.6.0

4.1.1.13 SelectedRowSegFunction

The segment function property for the selected move segment. The maximum value of the segment function is 10. A segment with a non-zero segment function value will show a square element at the segment position.



Schema: tchmi:general#/definitions/INT

Attribute getter: getSelectedRowSegFunction

Attribute setter: setSelectedRowSegFunction

Available since version 12.6.0

4.1.1.14 SelectedSegIndex

The selected segment index.

Schema: tchmi:general#/definitions/UINT

Attribute getter: getSelectedSegIndex

Available since version 12.6.0

4.1.1.15 InvertScale

Scale at the bottom of control will be inverted, it will show max position to the left end and minimum position will be shown at the right end.

Schema: tchmi:general#/definitions/Boolean

Attribute getter: getInvertScale

Attribute setter: setInvertScale

Available: since version 12.8.0

4.1.1.16 DigitAfterDecimal

Decimal precision of all numeric values displayed on the control.

Schema: tchmi:general#/definitions/UINT

Attribute getter: getDigitsAfterDecimal

Attribute setter: setDigitsAfterDecimal

Available: since version 12.8.0

4.1.1.17 CurrentValue

Current position of the axis.

Schema: tchmi:general#/definitions/Number

Attribute getter: getCurrentValue

Attribute setter: setCurrentValue

Available: since version 12.8.0

4.1.2 Functions

Name	Description
ActivateConfig [▶ 22]	This function will write the current changes of the control to the PLC.
ResetConfigData [▶ 23]	This function will revert any user changes by reloading the last activated configuration on control.
AddNewAxisSegment [▶ 23]	This function adds a new axis segment to the control.
RemoveAxisSegment [▶ 23]	This function removes the selected axis segment from the control.

4.1.2.1 ActivateConfig

By calling this function all user changes will be activated and written to PLC symbols.

Parameter

Name	Type	Description
-	-	-

Return value

Type	Description
void	No return value.

Available: since version 12.6.0

4.1.2.2 ResetConfigData

This function will revert any user changes by reloading the last activated configuration.

Parameter

Name	Type	Description
-	-	-

Return value

Type	Description
void	No return value.

Available since version 12.6.0

4.1.2.3 AddNewAxisSegment

This function adds a new axis segment to the control at the selected MoveIndex and selected SegmentIndex.

Parameter

Name	Type	Description
-	-	-

Return value

Type	Description
void	No return value.

Available since version 12.6.0

4.1.2.4 RemoveAxisSegment

This function removes the selected axis segment from the control. If a segment followed by other segments is removed, the segments after the removed segments are moved up one level so that there is no gap in the segment list.

Parameter

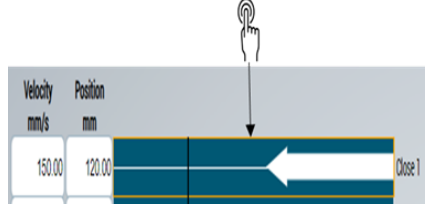
Name	Type	Description
-	-	-

Return value

Type	Description
void	No return value.

Available since version 12.6.0

4.1.3 User Interactions

Name	Description	Result on the control
Single Tap	Select a segment by tapping/ clicking on it.	

Available since version 12.6.0

4.2 BlowPressureChart

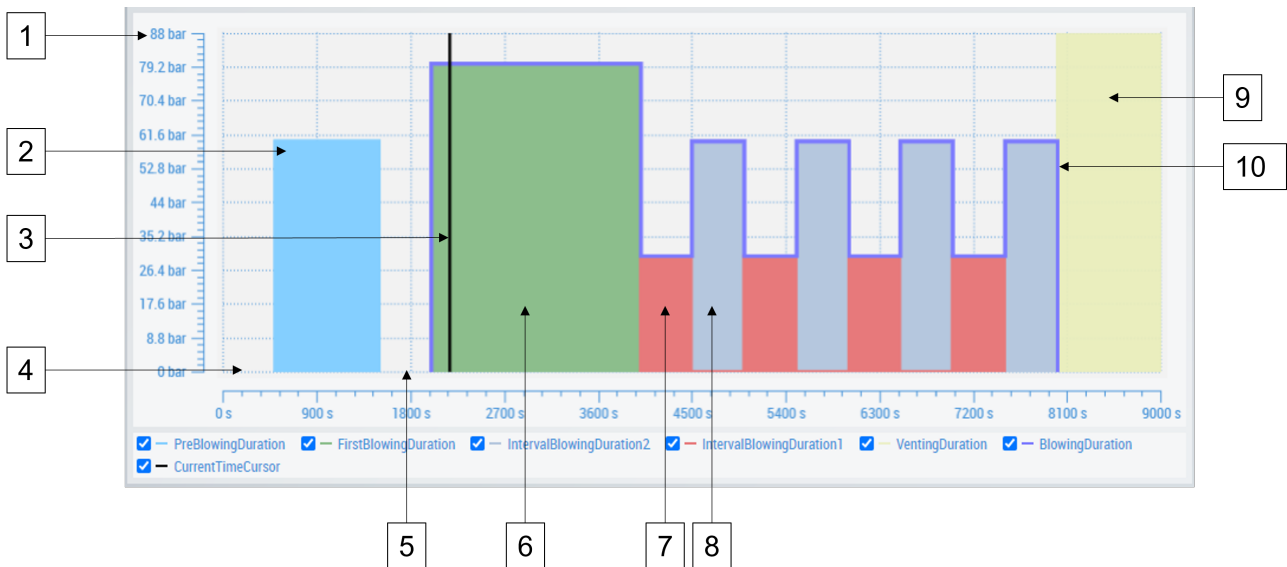
NuGet package: Beckhoff.TwinCAT.HMI.Plastic.Controls

Available: since version 12.6.0

BlowPressureChart is a special chart type framework control that can visualize the air blowing process in a blow molding machine.

Usually, the air blowing process is divided into different sections, each sections blows air for a fixed duration at a specific pressure.

The Pressure vs. Time chart of the BlowPressureChart allows the user to see the effects of different settings on the screen, a cursor line shows the current phase of process when it is active.



1. Pressure axis
2. Pre-blow duration and pressure
3. Current time cursor
4. Pre blow delay
5. Blow delay
6. First blowing duration and pressure
7. Interval blowing duration 1 and pressure
8. Interval blowing duration 2 and pressure
9. Venting duration
10. Total blowing duration.

4.2.1 Attributes

Category: Colors

Name	Description
PreBlowingDurationColor [▶ 25]	Sets the color for the pre blowing duration curve.
BlowingDurationColor [▶ 26]	Sets the color for the blowing duration curve.
FirstBlowingDurationColor [▶ 26]	Sets the color of the first blowing duration curve.
IntervalBlowingDuration1Color [▶ 26]	Sets the color of the interval blowing duration1 curve.
IntervalBlowingDuration2Color [▶ 26]	Sets the color of the interval blowing duration 2 curve.
CurrentTimeCursorlineColor [▶ 26]	Sets the color of the current time cursor line.
VentingDurationColor [▶ 26]	Sets the color for the venting duration.

Category: Time

Name	Description
DelayPreBlowing [▶ 27]	Sets the delay for the pre blowing value.
DurationPreBlowing [▶ 27]	Sets the duration for the pre blowing value.
DelayBlowing [▶ 27]	Sets the delay for the blowing value.
BlowingDuration [▶ 27]	Sets the total blowing duration.
FirstBlowingDuration [▶ 27]	Sets the first blowing interval value.
IntervalBlowingDuration1 [▶ 28]	Sets the duration of the first interval blowing duration value.
IntervalBlowingDuration2 [▶ 28]	Sets the duration of the second interval blowing duration value.
VentingDuration [▶ 28]	Sets the venting duration value.
CurrentTimeCursor [▶ 28]	Current time cursor of the blow pressure chart.

Category: Pressure

Name	Description
PreBlowPressure [▶ 28]	Sets the pre blow pressure value.
BlowPressure [▶ 28]	Sets the first blow pressure value.
BlowPressValue1 [▶ 29]	Sets the blow pressure value for the interval 1 blowing.
BlowPressValue2 [▶ 29]	Sets the blow pressure value for the interval 2 blowing.

Category: Common

Name	Description
FillChart [▶ 29]	If set to <code>TRUE</code> the area under the chart will be highlighted

4.2.1.1 PreBlowingDuration Color

Sets the color for the pre blowing duration curve.

Schema: `tchmi:framework#/definitions/Color`

Attribute getter: `getPreBlowingDurationColor`

Attribute setter: `setPreBlowingDurationColor`

Available: since version 12.6.0

4.2.1.2 BlowingDuration Color

Sets the color for the blowing duration curve.

Schema: tchmi:framework#/definitions/Color

Attribute getter: getBlowingDurationColor

Attribute setter: setBlowingDurationColor

Available: since version 12.6.0

4.2.1.3 FirstBlowingDuration Color

Sets the color of the first blowing duration curve.

Schema: tchmi:framework#/definitions/Color

Attribute getter: getFirstBlowingDurationColor

Attribute setter: setFirstBlowingDurationColor

Available: since version 12.6.0

4.2.1.4 IntervalBlowingDuration1 Color

Sets the color of the interval blowing duration1 curve.

Schema: tchmi:framework#/definitions/Color

Attribute getter: getIntervalBlowingDuration1Color

Attribute setter: setIntervalBlowingDuration1Color

Available: since version 12.6.0

4.2.1.5 IntervalBlowingDuration2 Color

Sets the color of the interval blowing duration2 curve.

Schema: tchmi:framework#/definitions/Color

Attribute getter: getIntervalBlowingDuration2Color

Attribute setter: setIntervalBlowingDuration2Color

Available: since version 12.6.0

4.2.1.6 CurrentTimeCursorline Color

Sets the color of the current time cursor line.

Schema: tchmi:framework#/definitions/Color

Attribute getter: getCurrentTimeCursorlineColor

Attribute setter: setCurrentTimeCursorlineColor

Available: since version 12.6.0

4.2.1.7 VentingDuration Color

Sets the color for the venting duration.

Schema: tchmi:framework#/definitions/Color

Attribute getter: getVentingDurationColor

Attribute setter: setVentingDurationColor

Available: since version 12.6.0

4.2.1.8 DelayPreBlowing

Sets the delay of the pre blowing value.

Schema: tchmi:general#/definitions/Number

Attribute getter: getDelayPreBlowing

Attribute setter: setDelayPreBlowing

Available: since version 12.6.0

4.2.1.9 DurationPreBlowing

Sets the duration for the pre blowing value.

Schema: tchmi:general#/definitions/Number

Attribute getter: getDurationPreBlowing

Attribute setter: setDurationPreBlowing

Available: since version 12.6.0

4.2.1.10 DelayBlowing

Sets the delay blowing value.

Schema: tchmi:general#/definitions/Number

Attribute getter: getDelayBlowing

Attribute setter: setDelayBlowing

Available: since version 12.6.0

4.2.1.11 BlowingDuration

Sets the total blowing duration.

Schema: tchmi:general#/definitions/Number

Attribute getter: getBlowingDuration

Attribute setter: setBlowingDuration

Available: since version 12.6.0

4.2.1.12 FirstBlowingDuration

Sets the first blowing interval value.

Schema: tchmi:general#/definitions/Number

Attribute getter: getFirstBlowingDuration

Attribute setter: setFirstBlowingDuration

Available: since version 12.6.0

4.2.1.13 IntervalBlowingDuration1

Sets the duration of the first interval blowing duration value.

Schema: tchmi:general#/definitions/Number

Attribute getter: getIntervalBlowingDuration1

Attribute setter: setIntervalBlowingDuration1

Available: since version 12.6.0

4.2.1.14 IntervalBlowingDuration2

Sets the duration of the second interval blowing duration value.

Schema: tchmi:general#/definitions/Number

Attribute getter: getIntervalBlowingDuration2

Attribute setter: setIntervalBlowingDuration2

Available: since version 12.6.0

4.2.1.15 VentingDuration

Sets the venting duration value.

Schema: tchmi:general#/definitions/Number

Attribute getter: setVentingDurationColor

Attribute setter: getVentingDurationColor

Available: since version 12.6.0

4.2.1.16 CurrentTimeCursor

Current time cursor of the blow pressure chart.

Schema: tchmi:general#/definitions/Number

Attribute getter: getCurrentTimeCursor

Attribute setter: setCurrentTimeCursor

Available: since version 12.6.0

4.2.1.17 PreBlowPressure

Sets the pre blow pressure value.

Schema: tchmi:general#/definitions/Number

Attribute getter: getPreBlowPressure

Attribute setter: setPreBlowPressure

4.2.1.18 BlowPressure

Sets the first blow pressure value.

Schema: tchmi:general#/definitions/Number

Attribute getter: getBlowPressure

Attribute setter: setBlowPressure

Available: since version 12.6.0

4.2.1.19 BlowPressValue1

Sets the blow pressure value for the interval 1 blowing.

Schema: tchmi:general#/definitions/Number

Attribute getter: setBlowPressValue1

Attribute setter: getBlowPressValue1

Available: since version 12.6.0

4.2.1.20 BlowPressValue2

Sets the blow pressure value for the interval 2 blowing.

Schema: tchmi:general#/definitions/Number

Attribute getter: getBlowPressValue2

Attribute setter: setBlowPressValue2

Available: since version 12.6.0

4.2.1.21 FillChart

If set to `TRUE` the area under the chart will be highlighted.

Schema: tchmi:general#/definitions/BOOL

Attribute getter: getFillChart

Attribute setter: setFillChart

Available: since version 12.6.0

4.3 CamControl

NuGet package: Beckhoff.TwinCAT.HMI.Plastic.Controls

Available: since version 12.6.0

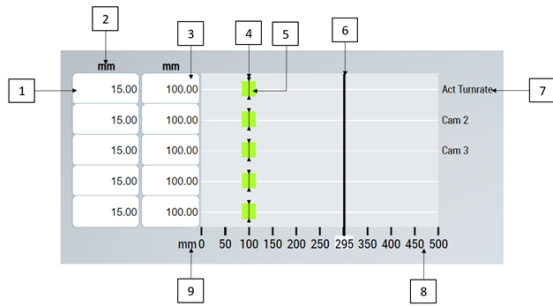
CamControl is used to visualize and configure array of cam switches. Each cam switch is defined by hysteresis and threshold values.

Threshold defines position of the cam switch on an axis from zero.

Hysteresis is the length added and subtracted from threshold to mark a range on the axis. If the axis position is in this range, the related cam should be switched on.

● Activate action

i Any new changes on the control must be activated using [ActivateConfig \[► 33\]](#) action to set new values to PLC.



1. Hysteresis value
2. Display unit of values
3. Threshold value
4. Threshold value visualization
5. Hysteresis zone visualization
6. Current axis position
7. Descriptor text for each cam switch
8. Axis scale
9. Scale unit

4.3.1 Attributes

Category: Configuration

Name	Description
MinValue [► 32]	Minimum value of the scale.
MaxValue [► 32]	Maximum value of the scale.
CamConfig [► 31]	List of cam switch configurations.
ColumnWidths [► 32]	Width for each of 4 columns.
RowHeight [► 32]	Height of each cam switch row in pixel.
RowHeight Unit [► 20]	Sets the Unit of the RowHeight property always in pixel.
CamDescriptor [► 31]	List of descriptor strings that will be displayed next to each cam switch row.
AxisUnitTypes [► 19]	List of unit types that can be used in the control.
InvertScale [► 33]	Sets direction of the scale displayed on the control.
DigitAfterDecimal [► 33]	Decimal precision of all numeric values displayed on the control.

Category: State

Name	Description
CurrentValue [► 32]	Current position of the axis.

Category: Common

Name	Description
ActivationRequired [► 30]	Is set to <code>TRUE</code> if user makes new changes that are not activated.

4.3.1.1 ActivationRequired

This read only attribute is set to `TRUE`, if a user makes new changes that are not activated.

This value can be used by other controls such as a TcHmiButton to change its appearance and inform the user that new changes must be activated.

Schema: tchmi:general#/definitions/Boolean

Attribute getter: getActivationRequired

Available: since version 12.6.0

4.3.1.2 AxisUnitTypes

List of unit types that can be used in the control.

Schema: tchmi:framework#/definitions/AMGAxisUnits

Attribute getter: getAxisUnitTypes

Attribute setter: setAxisUnitTypes

Available: since version 12.6.0

4.3.1.3 CamConfig

This attribute should be linked to a server symbol that represents the array of cam configuration objects (structures or function blocks) in the PLC. The linked symbol must be of the type array with minimum length of 1 and maximum length of 5.

Each element in this array must have the following properties.

Name	Type	Description
CamActive	Boolean	Is set to TRUE by the PLC for a cam switch when that cam switch is active.
Hysteresis	Number	Hysteresis property of a cam switch.
Threshold	Number	Threshold property of a cam switch.

Symbol	Value	Property	Value
PLC1.ARRAY_1..5_OF-FB_CamHmiNewCfg	✓	PLC1::MAIN::fbCamTest::fbCamHmiCfg	
PLC1.FB_CamHmiNewCfg [0]	✓	[0]	
PLC1.PROPERTY-GET-SET-BOOL	✓	CamActive	
PLC1.PROPERTY-GET-SET-LREAL	✓	Hysteresis	
PLC1.PROPERTY-GET-SET-LREAL	✓	Threshold	
PLC1.FB_CamHmiNewCfg [1]	✓	[1]	
PLC1.FB_CamHmiNewCfg [2]	✓	[2]	
PLC1.FB_CamHmiNewCfg [3]	✓	[3]	
PLC1.FB_CamHmiNewCfg [4]	✓	[4]	

Schema: tchmi:framework#/definitions/Symbol

Attribute getter: getCamConfig

Attribute setter: setCamConfig

Available: since version 12.6.0

4.3.1.4 CamDescriptor

List of descriptor strings that will be displayed next to every cam switch row.

Schema: tchmi:framework#/definitions/CamNameList

Attribute getter: getCamDescriptor

Attribute setter: setCamDescriptor

Available: since version 12.6.0

4.3.1.5 ColumnWidths

The width of each column can be configured through this attribute.

Schema: tchmi:framework#/definitions/AMG_ColWidthList

Attribute getter: getColumnWidths

Attribute setter: setColumnWidths

Available: since version 12.6.0

4.3.1.6 CurrentValue

Current position of the axis.

Schema: tchmi:general#/definitions/Number

Attribute getter: getCurrentValue

Attribute setter: setCurrentValue

Available: since version 12.6.0

4.3.1.7 MaxValue

It is the maximum value of the scale in system unit.

Schema: tchmi:general#/definitions/Number

Attribute getter: getMaxValue

Attribute setter: setMaxValue

Available: since version 12.6.0

4.3.1.8 MinValue

It is the minimum value of the scale in system unit.

Schema: tchmi:general#/definitions/Number

Attribute getter: getMinValue

Attribute setter: setMinValue

Available: since version 12.6.0

4.3.1.9 RowHeight

Height of each cam switch row in pixel.

Schema: tchmi:framework#/definitions/MeasurementValue

Attribute getter: getRowHeight

Attribute setter: setRowHeight

Available: since version 12.6.0

4.3.1.10 RowHeightUnit

Sets the unit of the RowHeight property always in pixel.

Schema: tchmi:framework#/definitions/PixelUnit

Attribute getter: getRowHeightUnit

Attribute setter: setRowHeightUnit

Available: since version 12.6.0

4.3.1.11 DigitAfterDecimal

Decimal precision of all numeric values displayed on the control.

Schema: tchmi:general#/definitions/UINT

Attribute getter: getDigitsAfterDecimal

Attribute setter: setDigitsAfterDecimal

Available: since version 12.8.0

4.3.1.12 InvertScale

Scale at the bottom of control will be inverted, it will show max position to the left end and minimum position will be shown at the right end.

Schema: tchmi:general#/definitions/Boolean

Attribute getter: getInvertScale

Attribute setter: setInvertScale

Available: since version 12.8.0

4.3.2 Functions

Category: Actions

Name	Description
ActivateConfig [▶ 33]	This function will write down the current changes by the user to PLC.
ResetConfigdata [▶ 33]	This function will revert all user changes and reload last activated configuration from PLC.

4.3.2.1 ActivateConfig

By calling this function all user changes will be activated and written to PLC symbols.

Parameter

Name	Type	Description
-	-	-

Return value

Type	Description
void	No return value.

Available: since version 12.6.0

4.3.2.2 ResetConfigdata

This function will revert any user changes by reloading the last activated configuration.

Parameter

Name	Type	Description
-	-	-

Return value

Type	Description
void	No return value.

Available: since version 12.6.0

4.3.3 Events

Category: Actions

Name	Description
onUserConfigActivated [▶ 34]	This event is generated when a new configuration is activated on the control.
onEditingCanceled [▶ 34]	This event is generated when user changes are canceled, and the old configuration is reloaded on the control.

4.3.3.1 onUserConfigActivated

This event is generated when a new configuration is activated on the control.

Available: since version 12.6.0

4.3.3.2 onEditingCanceled

This event is generated when user changes are canceled and the old configuration is reloaded on the control.

Available: since version 12.6.0

4.4 Configurator

NuGet package: Beckhoff.TwinCAT.HMI.Plastic.Controls

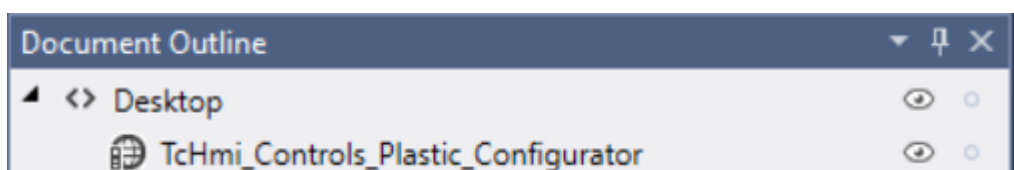
Available: since version 12.6.0

The configurator control stores user settings for the default input panel that appears when the user enters new value on controls provided in Beckhoff.TwinCAT.HMI.Plastic.Controls package.

It also stores measurement unit configuration, supplies functions to access this configuration and perform unit conversion calculations.

● The position of the control

i To allow all other controls to access the configurator control, it is necessary to include this control at the uppermost position in the control hierarchy. Setting it as the first control in the view is recommended.

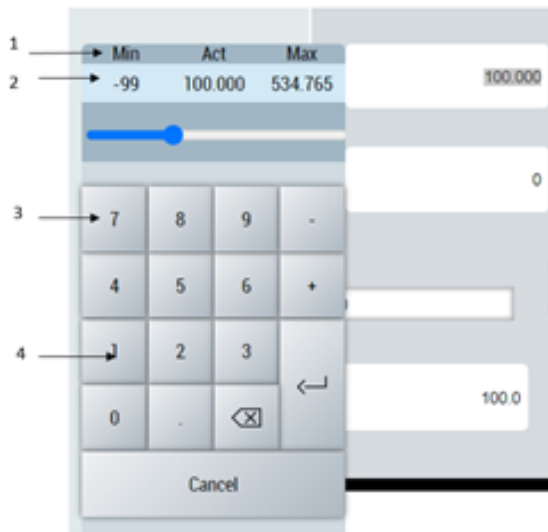


Input Panel

Colors	
Panel Background Colour	#FFA0B6C4
Text Background Color	#FFD3E9F6
Key Background Color	Theme
Key Label Color	#FF000000

Attributes to configure the InputPanel:

1. Panel Background Color
2. Text Background Color
3. Key Background Color
4. Key Label Color



4.4.1 Attributes

Category: Colors

Name	Description
PanelBackgroundColor [▶ 37]	Sets the background color for the input panel that will be available for editing InputBox controls.
TextBackgroundColor [▶ 38]	Sets the background color of the descriptor text displayed on the input panel.
KeyBackgroundColor [▶ 36]	Sets the background color of the keyboard keys used in the input panel.
KeyLabelColor [▶ 37]	Sets the label color of keyboard keys used in the input panel.

Category: InputPanel

Name	Description
KeyBoardLayout [▶ 37]	Sets the layout of the keyboard used in the input panel.
KeyBoardHeight [▶ 37]	Sets the height of the keyboard in pixels.
KeyBoardWidth [▶ 37]	Sets the width of the keyboard in pixels.

Category: Unit Conversion

Name	Description
UnitConfig [▶ 38]	Sets the layout of the keyboard used in the input panel.
DisplayUnitsChanged [▶ 36]	If display unit property of any unit group is changed, this attribute will change.
SystemUnitsChanged [▶ 38]	If system unit property of any unit group is changed, this attribute will change.
ConfiguredPercentUnitID [▶ 36]	Returns ID of <i>PercentUnit</i> configured in the Unit_Config.json file.
UnitGroups [▶ 38]	Returns a list of unit group objects configured in the Unit_Config.json file.
AllUnits [▶ 36]	Returns a list of units configured in Unit_Config.json file.

Category: Navigation

Name	Description

4.4.1.1 AllUnits

Returns a list of units configured in [Unit_Config.json](#) [[▶ 40](#)] file.

Schema: tchmi:general#/definitions/Array

Attribute getter: getAllUnits

Available: since version 12.6.0

4.4.1.2 ConfiguredPercentUnitID

Returns the unit ID of [PercentUnit](#) [[▶ 40](#)] configured in the JSON file.

Schema: tchmi:general#/definitions/String

Attribute getter: getConfiguredPercentUnitID

Available: since version 12.6.0

4.4.1.3 DisplayUnitsChanged

This attribute will change if any display unit property of any unit group is changed by a user in the client.

This attribute can be used as an event for other controls to recalculate unit-dependent properties in the project.

Schema: tchmi:general#/definitions/BOOL

Attribute getter: getDisplayUnitsChanged

Available: since version 12.6.0

4.4.1.4 KeyBackgroundColor

Sets the background color of the keyboard keys used in the input panel.

Schema: tchmi:framework#/definitions/Color

Attribute getter: getKeyBackground

Attribute setter: setKeyBackground

Available: since version 12.6.0

4.4.1.5 **KeyBoardHeight**

Sets the height of the keyboard in pixels.

Schema: tchmi:general#/definitions/Number

Attribute getter: getKeyBoardHeight

Attribute setter: setKeyBoardHeight

Available: since version 12.6.0

4.4.1.6 **KeyBoardLayout**

Sets the layout of the keyboard used in the input panel.

Default layout is a pre-configured numpad with escape key, user can choose a different layout to change it.

Schema: tchmi:framework#/definitions/TchMi.Controls.Beckhoff.TchMiKeyboard.ProjectKeyboardLayout

Attribute getter: getKeyBoardLayout

Attribute setter: setKeyBoardLayout

Available: since version 12.6.0

4.4.1.7 **KeyBoardWidth**

Sets the width of the keyboard in pixels.

Schema: tchmi:general#/definitions/Number

Attribute getter: getKeyBoardWidth

Attribute setter: setKeyBoardWidth

Available: since version 12.6.0

4.4.1.8 **KeyLabelColor**

Sets the label color of the keyboard keys used in the input panel.

Schema: tchmi:framework#/definitions/Color

Attribute getter: getKeyLabelColor

Attribute setter: setKeyLabelColor

Available: since version 12.6.0

4.4.1.9 **PanelBackgroundColor**

Sets the background color for the input panel that will be available for editing InputBox controls.

Schema: tchmi:framework#/definitions/Color

Attribute getter: getPanelBackgroundColour

Attribute setter: setPanelBackgroundColour

Available: since version 12.6.0

4.4.1.10 SystemUnitsChanged

If any system unit property of any unit group is changed, this attribute will change.

This attribute can be used as an event for other controls to recalculate unit-dependent properties in the project.

Schema: tchmi:general#/definitions/BOOL

Attribute getter: getSystemUnitsChanged

Available: since version 12.6.0

4.4.1.11 TextBackgroundColor

Sets the background color of the descriptor text displayed on the input panel.

Schema: tchmi:framework#/definitions/Color

Attribute getter: getTextBackgroundColor

Attribute setter: setTextBackgroundColor

Available: since version 12.6.0

4.4.1.12 UnitConfig

Path to Unit_Config.json file that configures different unit groups, units, and conversion functions.

Default Unit_Config.json file included in the package and its path is set as attribute default.

To add any new unit groups, or to change existing units

1. User should create a copy of the default file from path: `Beckhoff.TwinCAT.HMI.Plastic/InputGroup/Configuration/Unit_Config.json` into the project
2. Edit the new copy file
3. Set path of the new file as attribute value

Schema: tchmi:framework#/definitions/Path

Attribute getter: getUnitConfig

Attribute setter: setUnitConfig

Available: since version 12.6.0

4.4.1.13 UnitGroups

Returns a list of unit group objects configured in the Unit_Config.json file.

Schema: tchmi:general#/definitions/Array

Attribute getter: getUnitGroups

Available: since version 12.6.0

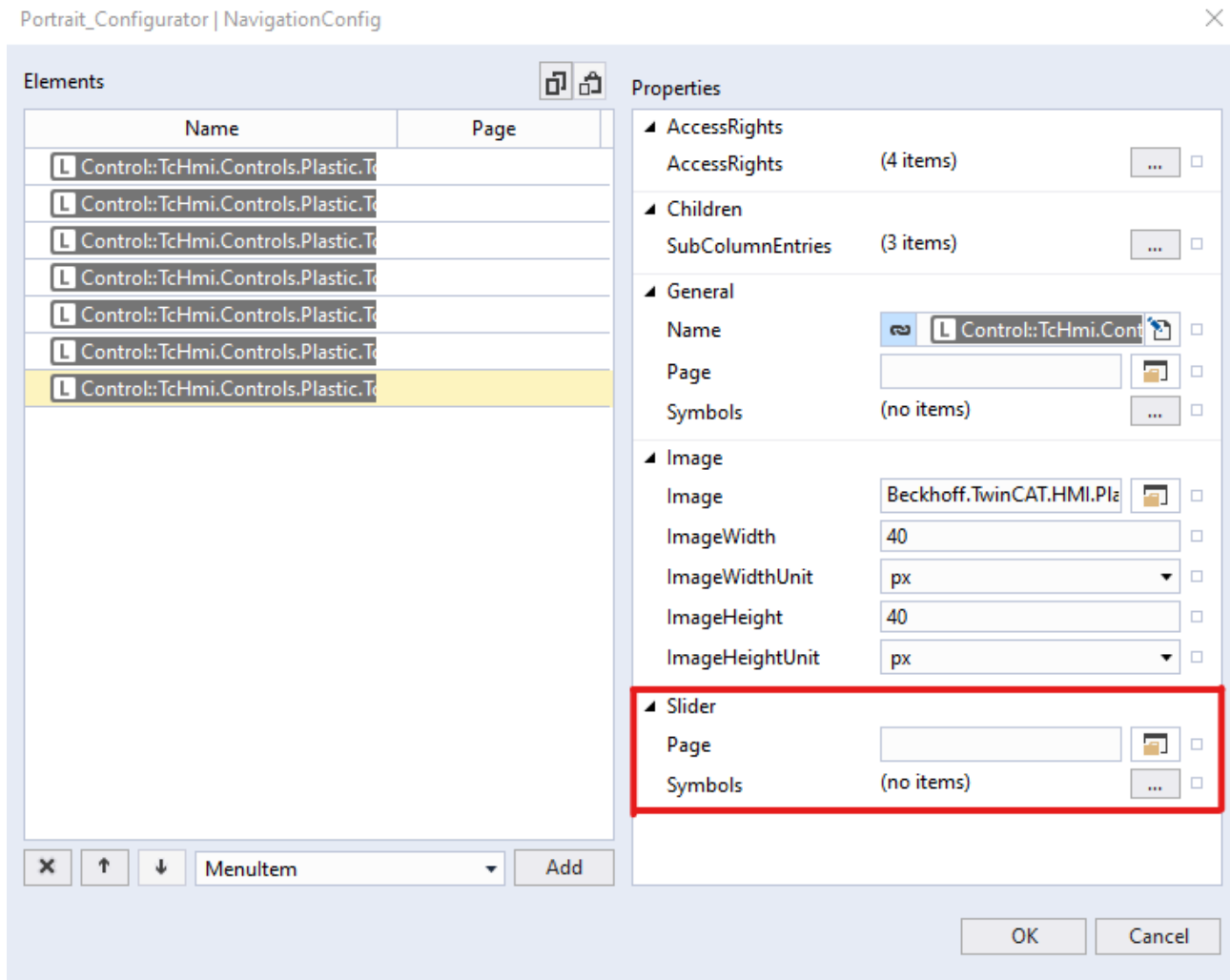
4.4.1.14 NavigationConfig

In Tc3_PlasticAppHmi projects a TcHmiNavigationbar control from package Beckhoff.TwinCAT.HMI.ResponsiveNavigation is used to configure and allow users to navigate to different contents.

Every content displayed on the main region can have related settings content displayed in the slider region in this application.

This attribute allows a user to configure the complete navigation tree and add information regarding any settings content needs to be displayed in the slider region when the target content is shown on the main region of the application. The configurator control watches any changes of content on the main region, and if configuration requires loads the slider content in the slider region.

More details on Navigation Bar Control and its MenuSrcData is available at [this](#) link.



For every content that is configured with this attribute, two new properties are available in comparison to the MenuSrcData attribute of the Navigation Bar control.

1. Slider Content: This is the path of content that should be displayed in the slider region when the Page is displayed in the main region.
2. Slider Symbols: If the content displayed in slider area is 'Parameters_Settings.content', the symbols property can be used to configure which parameters will be displayed on the content. A user can link a symbol of type 'fbParamHandle' to this property.

Schema: tchmi:framework#/definitions/PlasticNavigation

Attribute getter: getNavigationConfig

Attribute setter: setNavigationConfig

Available: since version 12.8.0

4.4.1.15 NavBarMenuData

The navigation configuration configured with NavigationConfig [▶ 38] attribute cannot be used directly with the TcHmiNavigationBar control.

NavBarMenuData attribute provides a derived navigation configuration that matches to the schema of MenuSrcData attribute of the TcHmiNavigationBar control.

This is an read only attribute, users should bind this to the MenuSrcData attribute of the TcHmiNavigationBar control.

Schema: tchmi:general#/definitions/Any

Attribute getter: getNavBarMenuData

Available: since version 12.8.0

4.4.2 Unit_Config.json

With the Unit_Config.json file, the user can configure which unit groups, unit IDs and unit conversion functions are used in any TwinCAT HMI project.

A default Unit_Config.json file is included in the package and can be found at <PackageInstallPath>/runtimes/native1.12-tchmi/InputGroup/Configuration/Unit_Config.json.

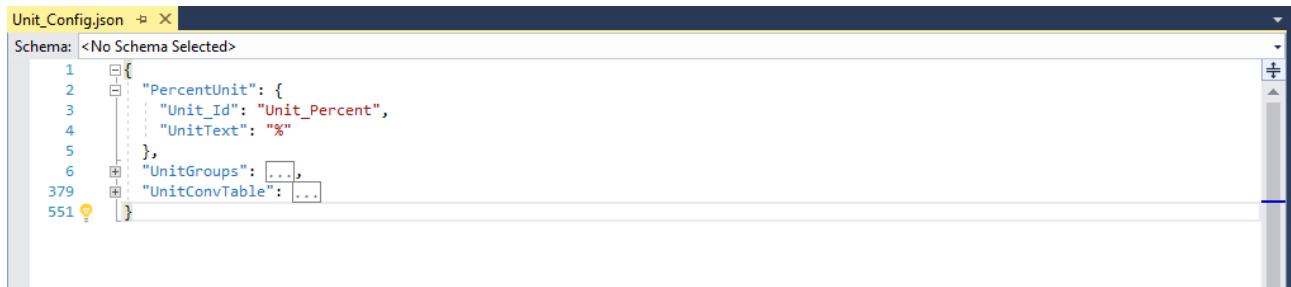
This file can be changed or a copy of it can be added to the project.

To set another file than the default Unit_Config.json, use the [UnitConfig \[▶ 38\]](#) attribute of the Configurator control.

Description of JSON structure of the Unit_Config.json is explained below.

The mandatory properties are:

- [PercentUnit \[▶ 40\]](#)
- [UnitGroups \[▶ 40\]](#)
- [UnitConvTable \[▶ 41\]](#)



4.4.2.1 PercentUnit

There can be only one PercentUnit in a configuration and it can be used by all unit groups if the value is to be converted to/from percent.

```
"PercentUnit": {
  "Unit_Id": "Unit_Percent",
  "UnitText": "%"
},
```

4.4.2.2 UnitGroups

The UnitGroups property is a list of unit group objects. The name of each object can be used as the [UnitGroup \[▶ 81\]](#) attribute for the InputBox control.

Each unit group object should have below explained properties.

- **Mandatory properties:**
- Units: list of unit objects.
- SystemUnit: Id of the unit that acts as the default system unit for the unit group.

- `DisplayUnit`: Id of the unit that acts as the default display unit for the unit group.
- **Optional properties:**
- `LocalizedKey`: User can add localized keys to newly added unit group, the key can be a localized symbol in the project.
- `LinkedDependentGroups`: This is an array of unit groups that are linked to this unit group and follow this unit groups changes. Every unit group listed in the list must have this unit group identifier mentioned as `LinkedSourceGroup` property.
- `LinkedSourceGroup`: This is group identifier marks the unit group that is watched for changes, any changes in configuration of the `LinkedSourceGroup` will be followed by this group.

Notice Any unit group can have only either of `LinkedDependentGroups` or `LinkedSourceGroup` property.

For example, Length and Velocity UnitGroups are shown below.

```
"UnitGroups": [
  {
    {
      "Length": {
        "DisplayUnit": "Unit_Length_mm",
        "SystemUnit": "Unit_Length_mm",
        "LinkedDependentGroups": [
          "Velocity",
        ],
        "Units": [
          {
            "Unit_Id": "Unit_Length_mm",
            "UnitText": "mm"
          },
          {
            "Unit_Id": "Unit_Length_inch",
            "UnitText": "in"
          }
        ]
      },
      {
        "Velocity": {
          "DisplayUnit": "Unit_Velocity_mm_s",
          "LinkedSourceGroup": "Length",
          "SystemUnit": "Unit_Velocity_mm_s",
          "Units": [
            {
              "LinkedSourceUnit": "Unit_Length_mm",
              "Unit_Id": "Unit_Velocity_mm_s",
              "UnitText": "mm/s"
            },
            {
              "LinkedSourceUnit": "Unit_Length_inch",
              "Unit_Id": "Unit_Velocity_in_s",
              "UnitText": "in/s"
            }
          ]
        }
      }
    }
  ]
}]
```

Each unit object in the units list should have the properties listed below:

- **Mandatory properties:**
- `Unit_Id`: Unique ID of the unit.
- `UnitText`: Text to be displayed on the screen when that unit is set as the display unit of its group.
- **Optional properties:**
- `LinkedSourceUnit`: This property gives the unit id of the `LinkedSourceGroup`, that is linked to this unit. For more detail use of units and unit groups linking, please check the [Linking of Units \[► 42\]](#).

4.4.2.3 UnitConvTable

The `UnitConvTable` property is a list of unit conversion objects.

Each unit conversion object must have three properties:

- `SourceUnit_Id`: Id of the source unit for conversion.
- `TargetUnit_Id`: Id of the target unit for conversion.
- `ConvFunction`: Conversion function that is applied to any value to convert it from source unit to target unit. It must be declared as a function in string format, where 'x' will be the input for the function, and it must return a value.

```
"UnitConvTable": [  
  {  
    "SourceUnit_Id": "Unit_Length_mm",  
    "TargetUnit_Id": "Unit_Length_inch",  
    "ConvFunction": "function (x) {return x*0.03937;}"  
  },  
]
```

4.4.2.4 Linking of unit groups

If we look at the example unit groups and units configured in [Unit Groups \[▶ 40\]](#) chapter, the Length is a base unit and Velocity is derived from it.

Ideally, we want a behavior such that, when the user changes display unit of Length group from "mm" to "inch" the linked group of Velocity should also switch its display unit from "mm/s" to "inch/s".

To achieve this, in the configuration Velocity is marked as one of the `LinkedDependentGroups` of Length And similarly, Length group is set as the `LinkedSourceGroup` in the Velocity group.

With this configuration if the user makes any changes in Length group it will also trigger changes in Velocity group.

Now to identify which unit in the Velocity group should be set as `DisplayUnit` unit the `LinkedSourceUnit` property will be used. In this example if "inch" is set as display unit for Length group, it will also set "inch/s" as display unit for Velocity group. As `Unit_Length_inch` from the Length group is set as `LinkedSourceUnit` property for `Unit_Velocity_in_s` unit in Velocity group.

Any changes done in a `LinkedDependentGroups` such as of Velocity will not trigger any changes in its `LinkedSourceGroup` that is the Length group. The linking is only one directional.

Feature of Linking of unit groups is available since version 12.8.0

4.5 CurveEditor

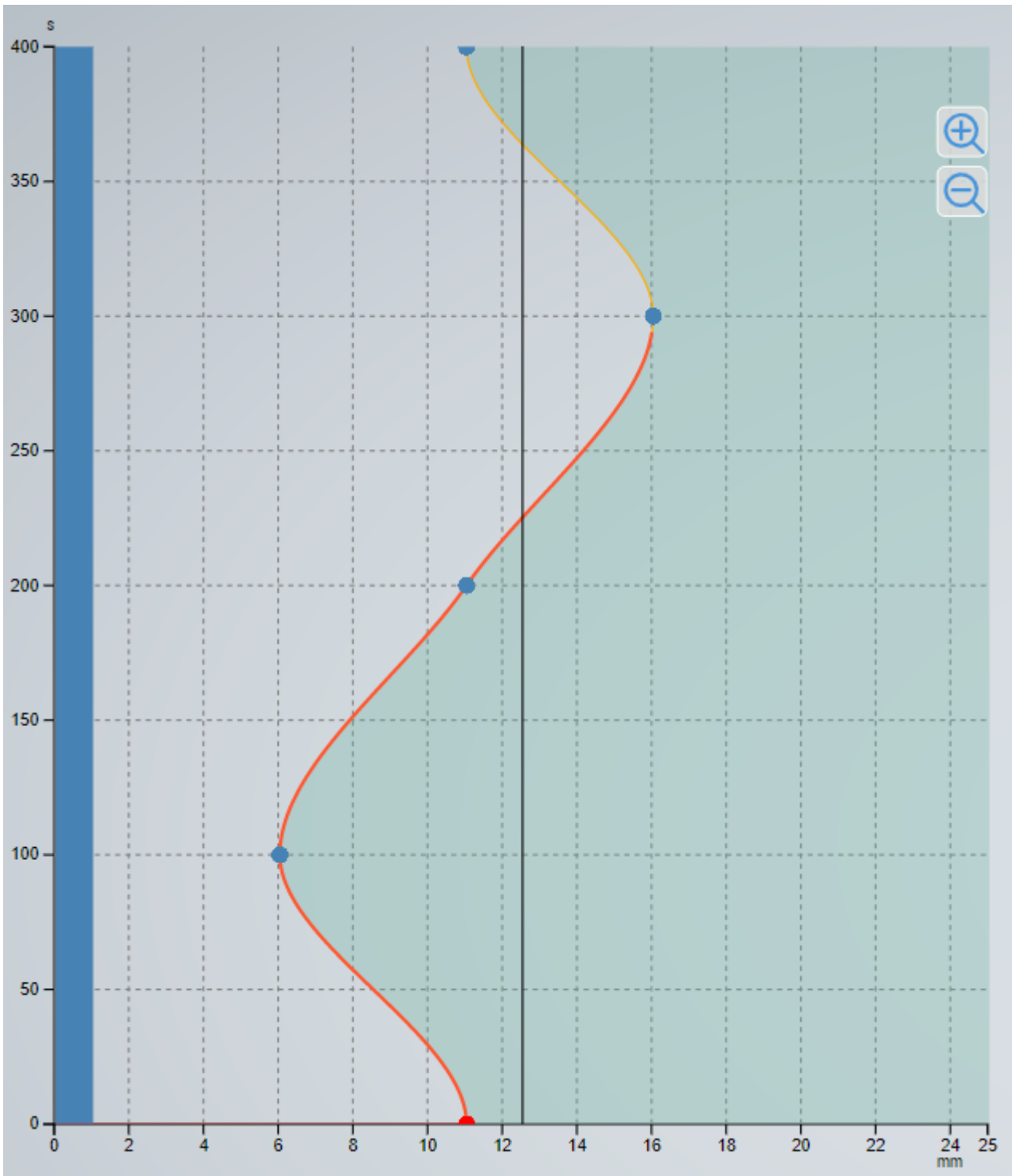
NuGet package: Beckhoff.TwinCAT.HMI.Plastic.Controls

Available: since version 12.6.0

The `CurveEditor` control provides users with the ability to add and manipulate a list of points on their HMI screen. In this way, the control is able to generate an interpolated curve that passes through each of these fixed points.

Users can use a range of touch gestures and attributes to define the specific parameters of the interpolation process, allowing for a high degree of customization and control.

The `CurveEditor` control is particularly well-suited for applications such as parison wall thickness control in blow molding machines, where precise and consistent control of the thickness of the material being molded is critical to the quality and performance of the final product.



4.5.1 Attributes

Category: Curve Editor

Name	Description
EditorProfiles [▶ 49]	List of configuration profiles.
ActiveProfileIndex [▶ 45]	Index of the config profile from the Editor Profiles [▶ 49] list that is currently displayed on the control.
ActualCurve [▶ 46]	List of fixed points representing actual curve values from the target.
ActualCurveIndex [▶ 46]	Last index of the values in ActualCurve [▶ 46] to be displayed on the control.

Category: Curve Data

Name	Description
BaseComponent [▶ 47]	Base component added to all set points.
CursorBase [▶ 48]	Cursor base position.
CursorValue [▶ 49]	Cursor value position.
Interpolation Modes [▶ 50]	Interpolation mode for curve calculations.
TransitionModes [▶ 56]	Transition mode between last and first point for the curve.
CalculateSlope [▶ 48]	If <code>TRUE</code> , the interpolation between two points will calculate the slope considering one point before and after these points, allowing a more continuous smooth curve.
AvoidNegativeVal [▶ 47]	Enables/disables negative values in generated setpoints.
DispCurrentProfile [▶ 49]	If <code>TRUE</code> , the curve on the actual curve attribute will be displayed.
ShowCentreLine [▶ 55]	Show/hide a line at the center of the value axis.
StepSize [▶ 56]	Step size for any incremental changes in other attributes.
ActivationRequired [▶ 45]	Is set to <code>TRUE</code> if the user makes new changes that are not activated.
ProfileComponent [▶ 52]	Highest value in the current fixed-point list.
FixedPoints [▶ 50]	Sets the fixed points to be interpolated.
SetPoints [▶ 52]	Result of interpolation of the fixed points as an output curve.

Category: Axes Data

Name	Description
BaseMax [▶ 47]	Base axis maximum.
ValueMin [▶ 58]	Value axis minimum.
ValueMax [▶ 58]	Value axis maximum.
BaseAxisPosition [▶ 47]	The base axis can be positioned horizontally or vertically.
ValueAxisUnitGroup [▶ 58]	Sets the value axis unit group.
BaseAxisUnit [▶ 47]	Unit text to be shown on the base axis.
DisplayValueinPercent [▶ 49]	If set to <code>TRUE</code> all the value of the graph will be displayed in percentage.

Category: Marker

Name	Description
ActiveMarkers [▶ 46]	If <code>TRUE</code> , activates the markers from the marker list.
MarkerList [▶ 51]	Shows the selected marker points list.
MarkerWidth [▶ 52]	Sets the marker width that is related to the base axis.
MarkerLength [▶ 51]	Sets the marker width that is related to the value axis.

Category: Shift Mode

Name	Description
ShiftMode [▶ 52]	Sets the control to shift mode, which allows one or more fixed points to be shifted in different directions.
AllowShiftVertically [▶ 46]	If set to <code>TRUE</code> , points can be shifted along the vertical axis in shift mode.
AllowShiftHorizontally [▶ 46]	If set to <code>TRUE</code> , points can be shifted along the horizontal axis in shift mode.
ShiftSelectionMode [▶ 55]	Selection mode to select fixed points for shifting.

Category: Layout

Name	Description
ViewBoxWidth [▶ 59]	Sets the horizontal resolution of the graph in pixels.
ViewBoxHeight [▶ 59]	Sets the vertical resolution of the graph in pixels.
Fixed Point Radius [▶ 50]	Sets the radius of fixed points displayed on the control in pixels.

4.5.1.1 ActivationRequired

This read only attribute is set to `TRUE`, if a user makes new changes that are not activated.

This value can be used by other controls such as a `TchHmiButton` to change its appearance and inform the user that new changes must be activated.

Schema: `tchmi:general#/definitions/Boolean`

Attribute getter: `getActivationRequired`

Available: since version 12.6.0

4.5.1.2 ActiveProfileIndex

Index of the config profile from the editor profiles list that is currently displayed on the control.

By changing this index, the user can select different profiles from the editor profiles list to be displayed on the control. To activate the displayed profile, the `ActivateConfig` [▶ 63] function must be called on the control.

On activation, the selected profile will be added to the top of the list, and the active profile index attribute will return 0.

Schema: `tchmi:general#/definitions/Number`

Attribute getter: `getActiveProfileIndex`

Attribute setter: `setActiveProfileIndex`

Available: since version 12.6.0

4.5.1.3 ActiveMarkers

If `TRUE`, markers will be displayed from the [Marker List \[► 51\]](#) and generated setpoints will include marker bumps as per the marker profile set with the [Marker Length \[► 51\]](#) and [Marker Width \[► 52\]](#) attributes.

Schema: tchmi:general#/definitions/Boolean

Attribute getter: getActiveMarkers

Attribute setter: setActiveMarkers

Available: since version 12.6.0

4.5.1.4 ActualCurve

List of fixed points representing actual curve values from the target.

The user can compare the calculated set points curve with real values from the target by displaying the actual curve.

It is advisable to use binding interval of 50 milliseconds to match the animation transition time for the smooth display of a changing curve on the control.

Schema: tchmi:framework#/definitions/FixedPointList

Attribute getter: getActCurve

Attribute setter: setActCurve

Available: since version 12.6.0

4.5.1.5 ActualCurveIndex

Last index of the values in [ActualCurve \[► 46\]](#) to be displayed on the control.

To display only a part of the actual curve on the control, set the index of the last valid point to this attribute.

To show the full curve, the total length of the curve array should be set.

Schema: tchmi:general#/definitions/Number

Attribute getter: getActCurveIndex

Attribute setter: setActCurveIndex

Available: since version 12.6.0

4.5.1.6 AllowShiftHorizontally

If set to `TRUE` fixed points can be shifted along the horizontal axis in shift mode.

Schema: tchmi:general#/definitions/Boolean

Attribute getter: getAllowShiftInHorizontalDirection

Attribute setter: setAllowShiftInHorizontalDirection

Available: since version 12.6.0

4.5.1.7 AllowShiftVertically

If set to `TRUE`, fixed points can be shifted along the vertical axis in shift mode.

Schema: tchmi:general#/definitions/Boolean

Attribute getter: getAllowShiftInVerticalDirection

Attribute setter: setAllowShiftInVerticalDirection

Available: since version 12.6.0

4.5.1.8 AvoidNegativeVal

Enables/disables negative values in generated setpoints. If set to `TRUE`, generated setpoints will be floored at minimum of 0, there will be no negative output.

Schema: tchmi:general#/definitions/Boolean

Attribute getter: getAvoidNegativeValues

Attribute setter: setAvoidNegativeValues

Available: since version 12.6.0

4.5.1.9 BaseAxisPosition

The base axis can be positioned horizontally or vertically.

Schema: tchmi:framework#/definitions/BaseAxisPosition

Attribute getter: getBaseAxisPosition

Attribute setter: setBaseAxisPosition

Available: since version 12.6.0

4.5.1.10 BaseAxisUnit

Unit text to be shown on the base axis.

Schema: tchmi:general#/definitions/String

Attribute getter: getBaseAxisUnit

Attribute setter: setBaseAxisUnit

Available: since version 12.6.0

4.5.1.11 BaseComponent

Base component added to all set points. It can be changed by setting this attribute to the required value.

It is also possible to increase or decrease the base component by executing the [IncreaseBaseComponent \[▶ 61\]](#) or [DecreaseBaseComponent \[▶ 61\]](#) functions respectively.

Schema: tchmi:general#/definitions/Number

Attribute getter: getBaseComponent

Attribute setter: setBaseComponent

Available: since version 12.6.0

4.5.1.12 BaseMax

Base axis maximum.

Schema: tchmi:general#/definitions/Number

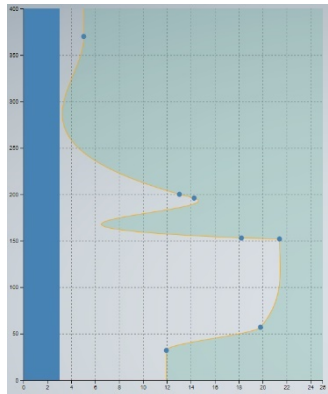

Attribute getter: getBaseMax

Attribute setter: setBaseMax

Available: since version 12.6.0

4.5.1.13 CalculateSlope

If the **PolynomialDegreeof3** interpolation is used, the curve interpolation can be calculated in two different ways using the Calculate Slope attribute:

Calculate slope property	Description	Example
TRUE	The upward gradient of a meant linear connection between the base before and after the regarded point is calculated and a weighting for the upward gradient at the regarded point is made as a function of the distance between the previous and next points.	
FALSE	The gradient at the fixpoints is 0. The interpolation will achieve a horizontal progression through the fixed points.	

Schema: tchmi:general#/definitions/Boolean

Attribute getter: getCalculateSlopes

Attribute setter: setCalculateSlopes

Available: since version 12.6.0

4.5.1.14 CursorBase

This is the base property of the current set point selected by the cursor. It can be used both ways.

1. If the user changes the cursor position by interacting with the curve or using the [IncreaseCursorBase \[▶_62\]](#) or [DecreaseCursorBase \[▶_62\]](#) functions, the updated cursor base can be read from this attribute.
2. If the user writes the desired cursor base to this attribute, the cursor will move to highlight the desired position.

Schema: tchmi:general#/definitions/Number

Attribute getter: getCursorBase

Attribute setter: setCursorBase

Available: since version 12.6.0

4.5.1.15 CursorValue

This is the value property of the point selected by the cursor.

1. If the user changes the cursor position or executes [IncreaseValueAtCursor \[► 63\]](#) or [DecreaseValueAtCursor \[► 63\]](#) functions, the updated Value property of the point currently selected by the cursor can be read from this attribute.
2. If the user writes the desired value to this attribute, the curve will create a new fixed point at the selected base position.
3. If the cursor is already pointing to an existing fixed point, it will result in a change of the value property of the fixed point.

Schema: tchmi:general#/definitions/Number

Attribute getter: getCursorValue

Attribute setter: setCursorValue

Available: since version 12.6.0

4.5.1.16 DispCurrentProfile

If `TRUE`, the curve on [Actual Curve \[► 46\]](#) attribute will be displayed.

Schema: tchmi:general#/definitions/Boolean

Attribute getter: getDisplayCurrentProfile

Attribute setter: setDisplayCurrentProfile

Available: since version 12.6.0

4.5.1.17 DisplayValueInPercent

If set to `TRUE`, all the values of the graph will be displayed in percentage.

Schema: tchmi:general#/definitions/Boolean

Attribute getter: getDispValueInPercent

Attribute setter: setDispValueInPercent

Available: since version 12.6.0

4.5.1.18 EditorProfiles

List of configuration profiles.

The user can iterate through the list of profiles using [ActiveProfileIndex \[► 45\]](#) attribute and activate any profile from the list.

On activation, the latest profile will be added at the start of the profile list. A total of 10 profiles can be stored in this list. By canceling the editing with the [ResetConfigdata \[► 63\]](#) function, the control will display the last activated profile.

Schema: tchmi:framework#/definitions/CurveEditorConfigList

Attribute getter: getEditorProfiles

Attribute setter: setEditorProfiles

Available: since version 12.6.0

4.5.1.19 FixedPointRadius

Set the radius of fixed points displayed on the control in pixels.

Schema: tchmi:general#/definitions/Number

Attribute getter: getFixedPointRadius

Attribute setter: setFixedPointRadius

Available: since version 12.6.0

4.5.1.20 FixedPoints

Sets the fixed points to be interpolated.

Schema: tchmi:framework#/definitions/FixedPointList

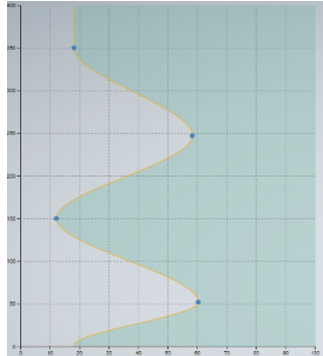
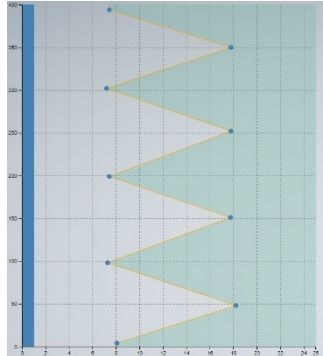
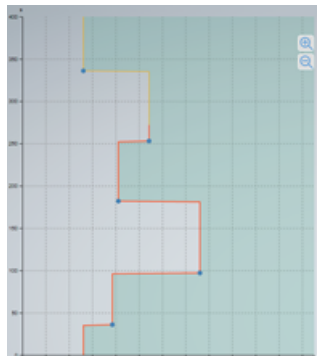
Attribute getter: getFixedPoints

Attribute setter: setFixedPoints

Available: since version 12.6.0

4.5.1.21 InterpolationModes

The interpolation mode for the set point curve can be set with this attribute.

Interpolation mode	Description	Example
PolynomialOfDegree3	The used interpolation is PolynomialOfDegree3; the interpolation can be influenced using the Calculate Slope [▶ 48] attribute.	
Linear	Linear interpolation between two fixed points is calculated for the curve.	
StepAfter	The curve changes its value only at fixed points and maintains the last fixed-point value between two points.	

Schema: tchmi:framework#/definitions/InterpolationMode

Attribute getter: getInterpolationMode

Attribute setter: setInterpolationMode

Available: since version 12.6.0

4.5.1.22 MarkerLength

Sets the length of each marker bump related to the value axis.

Schema: tchmi:general#/definitions/Number

Attribute getter: getMarkerLength

Attribute setter: setMarkerLength

Available: since version 12.6.0

4.5.1.23 MarkerList

Shows the selected marker points list.

To add or remove a marker from the marker list:

1. Place the cursor at the position where the marker point should be set/reset using single touch or by setting the cursor base attribute.
2. Click *AddNewMarker* or *RemoveSelectedMarker*.
⇒ A new marker point will be set or an active marker point will be deleted.

Schema: tchmi:framework#/definitions/FixedPointList

Attribute getter: getMarkerList

Attribute setter: setMarkerList

Available: since version 12.6.0

4.5.1.24 MarkerWidth

Sets the width of the marker bump that is related to the base axis.

Schema: tchmi:general#/definitions/Number

Attribute getter: getMarkerWidth

Attribute setter: setMarkerWidth

Available: since version 12.6.0

4.5.1.25 ProfileComponent

Highest value in the generated set points list.

Schema: tchmi:general#/definitions/Number

Attribute getter: getProfileComponent

Attribute setter: setProfileComponent

Available: since version 12.6.0

4.5.1.26 SetPoints

Result of interpolation of the fixed points as an output curve.

Schema: tchmi:framework#/definitions/FixedPointList

Attribute getter: getSetPoints

Attribute setter: setSetPoints

Available: since version 12.6.0

4.5.1.27 ShiftMode

Sets the control to shift mode, which allows one or more fixed points to be shifted in different directions.

Once the shift mode is activated, a red border will appear around the control to show that shift mode is active.

No user interactions




During shift mode, the following user interactions are disabled: Two-finger zooming, dragging a single fixed point, double tap to add a new point and base component dragging.

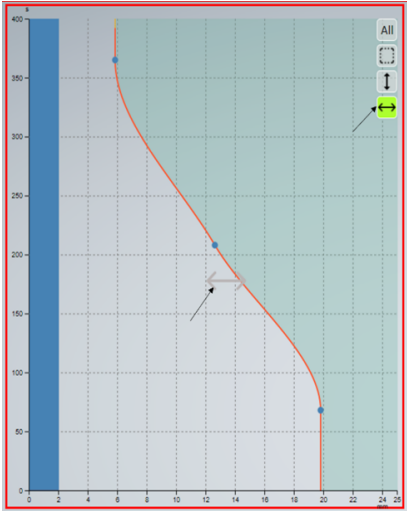
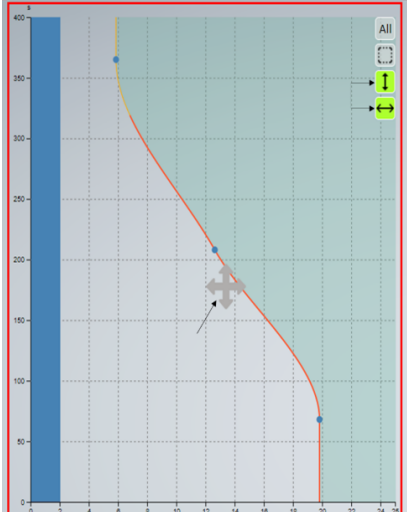
In this mode, all the selected fixed points can be dragged in three directions:

- Drag in base direction

- Drag in value axis direction
- Drag in both directions along the graph

The user can perform shift actions in shift mode using the buttons that appear in the top right corner.

Action	Description	Example
<p>Select all fixed points</p>	<p>If the first button is clicked once, all the fixed points that are available on the graph will be selected.</p>	 <p>The example shows a graph with a blue bar on the left and a curve on the right. Three red dots are placed on the curve at approximately x=6, x=12, and x=20. A toolbar in the top right corner has the 'All' button highlighted in green, indicating that all fixed points are selected.</p>
<p>Select fixed points using selection rectangle</p>	<p>The user can draw a rectangle on the screen using single-finger drag. All points covered by this rectangle and any point that is not selected but its base index falls between the first and last selected points will be selected at the end of the dragging gesture that draws the selection rectangle.</p>	 <p>The example shows the same graph as the first row. A dashed black rectangle is drawn around the first two red dots on the curve. The 'All' button in the toolbar is highlighted in green, indicating that the selection process is complete.</p>
<p>Drag fixed points in vertical direction</p>	<p>After selecting the desired fixed points, the user can select the drag direction using the button shown in the next figure.</p> <p>When the drag in the vertical direction is enabled, a drag handle will appear in the middle of the graph.</p> <p>The user can drag this handle along the base axis direction and all selected fixed points will move.</p>	 <p>The example shows the same graph. A vertical grey bar with a double-headed arrow is positioned on the curve between the first two red dots, indicating that vertical dragging is enabled. The 'All' button in the toolbar is highlighted in green.</p>

Action	Description	Example
<p>Drag fixed points in horizontal axis direction</p>	<p>After selecting the desired fixed points, the user can select the drag direction using the button shown in the next figure.</p> <p>When the drag in the horizontal axis direction is enabled, a drag handle will appear in the middle of the graph.</p> <p>The user can drag this handle along the base axis direction and all selected fixed points will move in the value axis directions.</p>	
<p>Drag fixed points in all directions</p>	<p>If base axis direction and value axis directions are enabled, the selected fixed points can be shifted in both directions simultaneously.</p>	

Schema: tchmi:genera##/definitions/Boolean

Attribute getter: getShiftMode

Attribute setter: setShiftMode

Available: since version 12.6.0

4.5.1.28 ShiftSelectionMode

Selection mode to select fixed points for shifting in shift mode.

Schema: tchmi:framework##/definitions/ShiftSelectionMode

Attribute getter: getShiftSelectionMode

Attribute setter: setShiftSelectionMode

Available: since version 12.6.0

4.5.1.29 ShowCentreLine

Shows/hides a line at the center of the value axis.

Schema: tchmi:genera##/definitions/Boolean

Attribute getter: getShowCentreLine

Attribute setter: setShowCentreLine

Available: since version 12.6.0

4.5.1.30 StepSize

Step size for any incremental changes in other attributes. This number attribute will affect the increment or decrement steps for the following functions:

- [IncreaseValueAtCursor \[▶ 63\]](#)
- [DecreaseValueAtCursor \[▶ 63\]](#)
- [IncreaseBaseComponent \[▶ 61\]](#)
- [DecreaseBaseComponent \[▶ 61\]](#)
- [IncreaseProfileComponent \[▶ 61\]](#)
- [DecreaseProfileComponent \[▶ 62\]](#)

Schema: tchmi:general#/definitions/Number

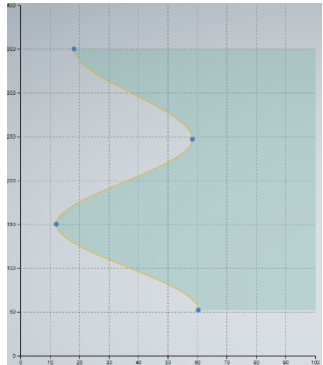
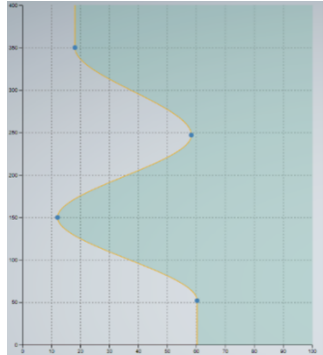
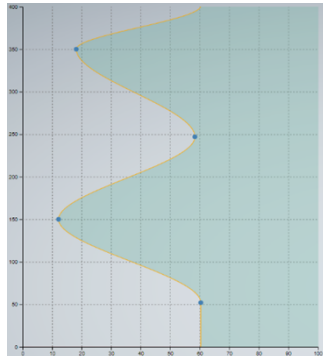
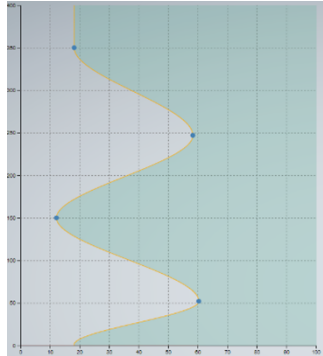
Attribute getter: getStepSize

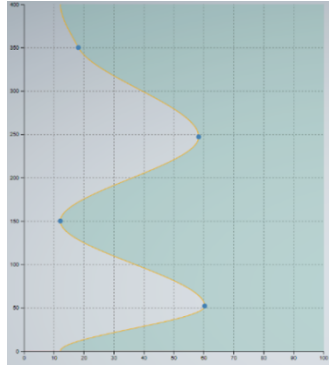
Attribute setter: setStepSize

Available: since version 12.6.0

4.5.1.31 TransitionModes

Transition between the last and first points of the curve.

Transition mode	Description	Example
none	There is no transition between the start and end points; the curve only covers the given fixed points.	 <p>The graph shows a coordinate system with a horizontal axis from 0 to 100 and a vertical axis from 0 to 60. Four blue dots representing fixed points are located at approximately (20, 55), (10, 15), (60, 25), and (60, 5). A yellow curve connects these points. At the start point (20, 55), the curve has a sharp corner, and at the end point (60, 5), it also has a sharp corner, indicating no transition.</p>
StartAndEndRepeat	If there are no fixed points placed on the first and last positions of the curve, the first and the last set points take the value of the nearest fixed points.	 <p>The graph is identical to the 'none' mode. However, the yellow curve starts at the first fixed point (20, 55) and ends at the last fixed point (60, 5), with a smooth transition at both ends. The start and end points of the curve are repeated to the values of the nearest fixed points.</p>
EndTransition	If there are no fixed points placed on the first and last positions of the curve, the first and the last set points take the value of the fixed points with the lower base.	 <p>The graph is identical to the 'none' mode. The yellow curve starts at the first fixed point (20, 55) and ends at the last fixed point (60, 5). At the start point, the curve smoothly transitions to the lower base value of the first fixed point (20). At the end point, it smoothly transitions to the lower base value of the last fixed point (60).</p>
StartTransition	If there are no fixed points placed on the first and last positions of the curve, the first and the last profile points take the value of the fixed points with the higher base.	 <p>The graph is identical to the 'none' mode. The yellow curve starts at the first fixed point (20, 55) and ends at the last fixed point (60, 5). At the start point, the curve smoothly transitions to the higher base value of the first fixed point (20). At the end point, it smoothly transitions to the higher base value of the last fixed point (60).</p>

Transition mode	Description	Example
StartAndEndThinnestPoint	If there are no fixed points placed on the first and last positions of the curve, the first and the last profile points take the value of the fixed points with the smallest value.	

Schema: tchmi:framework#/definitions/TransitionMode

Attribute getter: getTransitionMode

Attribute setter: setTransitionMode

Available: since version 12.6.0

4.5.1.32 ValueAxisUnitGroup

The unit group of the value axis. This group must be configured beforehand in the [unit config file \[▶ 40\]](#) of the [Configurator \[▶ 34\]](#) control in the project. The axis and values of the curve will be changed automatically by changing the display unit of the given group. The unit text of the display unit group will be displayed as the axis text.

Schema: tchmi:general#/definitions/String

Attribute getter: getValueAxisUnitGroup

Attribute setter: setValueAxisUnitGroup

Available: since version 12.6.0

4.5.1.33 ValueMax

Value axis maximum.

Schema: tchmi:general#/definitions/Number

Attribute getter: getValueMax

Attribute setter: setValueMax

Available: since version 12.6.0

4.5.1.34 ValueMin

Value axis minimum; it can be set to a negative value to allow curve values of less than 0 to be shown.

Schema: tchmi:general#/definitions/Number

Attribute getter: getValueMin

Attribute setter: setValueMin

Available: since version 12.6.0

4.5.1.35 **ViewBoxHeight**

Sets the vertical resolution of the graph in pixels.

Schema: tchmi:general#/definitions/Number

Attribute getter: getViewBoxHeight

Attribute setter: setViewBoxHeight

Available: since version 12.6.0

4.5.1.36 **ViewBoxWidth**

Sets the horizontal resolution of the graph in pixels.

Schema: tchmi:general#/definitions/Number

Attribute getter: getViewBoxWidth

Attribute setter: setViewBoxWidth

Available: since version 12.6.0

4.5.2 **Functions**

Category: Actions

Name	Description
selectNextFixedPoint [► 60]	This function selects the next available fixed point on the curve.
selectPreviousFixedPoint [► 61]	This function selects the previously available fixed point on the curve.
IncreaseBaseComponent [► 61]	This function increases the base component value by the step size.
DecreaseBaseComponent [► 61]	This function decreases the base component value by the step size.
IncreaseProfileComponent [► 61]	This function increases the profile component by the step size.
DecreaseProfileComponent [► 62]	This function decreases the profile component by the step size.
IncreaseCursorBase [► 62]	This function increases the cursor position on the base axis by 1. If shift mode is active, it will increment the base of all selected fixed points by 1.
DecreaseCursorBase [► 62]	This function decreases the cursor position on the base axis by 1. If shift mode is active, it will decrement the base of all selected fixed points by 1.
IncreaseValueAtCursor [► 63]	This function increases the value of the point which is currently selected by the cursor by the step size in normal mode. If shift mode is active, it will increment the value of all selected fixed points.
DecreaseValueAtCursor [► 63]	This function decreases the value of the point which is currently selected by the cursor by the step size in normal mode. If shift mode is active, it will decrement the value of all selected fixed points.
ActivateConfig [► 63]	Adds the current configuration to the editor profiles list and generates a setpoint for the current data.
ResetConfigdata [► 63]	This function removes all user changes and reloads the active configuration values.
AddNewFixPoint [► 64]	This function adds a new fixed point at the point selected by the cursor.
RemoveSelectedFixPoint [► 64]	This function removes the fixed point currently selected by the cursor; it will not change anything if no fixed point is selected.
AddNewMarker [► 64]	This function adds a new marker at the base location selected by the cursor.
RemoveSelectedMarker [► 65]	This function removes the marker currently selected by the cursor; it will not change anything if no marker is selected.

4.5.2.1 selectNextFixedPoint

This function selects the next available fixed point on the curve.

Parameter

Name	Type	Description
-	-	-

Return value

Type	Description
void	No return value.

Available: since version 12.6.0

4.5.2.2 selectPreviousFixedPoint

This function selects the previously available fixed point on the curve.

Parameter

Name	Type	Description
-	-	-

Return value

Type	Description
void	No return value.

Available: since version 12.6.0

4.5.2.3 IncreaseBaseComponent

This function increases the base component value by the step size [[▶ 56](#)].

Parameter

Name	Type	Description
-	-	-

Return value

Type	Description
void	No return value.

Available: since version 12.6.0

4.5.2.4 DecreaseBaseComponent

This function decreases the base component value by the step size.

Parameter

Name	Type	Description
-	-	-

Return value

Type	Description
void	No return value.

Available: since version 12.6.0

4.5.2.5 IncreaseProfileComponent

This function increases the profile component by the step size.

Parameter

Name	Type	Description
-	-	-

Return value

Type	Description
void	No return value.

Available: since version 12.6.0

4.5.2.6 DecreaseProfileComponent

This function decreases the profile component by the step size.

Parameter

Name	Type	Description
-	-	-

Return value

Type	Description
void	No return value.

Available: since version 12.6.0

4.5.2.7 IncreaseCursorBase

This function increases the cursor position on the base axis by 1. If shift mode is active, it will increment the base of all selected fixed points by 1.

Parameter

Name	Type	Description
-	-	-

Return value

Type	Description
void	No return value.

Available: since version 12.6.0

4.5.2.8 DecreaseCursorBase

This function decreases the cursor position on the base axis by 1. If shift mode is active, it will decrement the base of all selected fixed points by 1.

Parameter

Name	Type	Description
-	-	-

Return value

Type	Description
void	No return value.

Available: since version 12.6.0

4.5.2.9 IncreaseValueAtCursor

This function increases the value of the point which is currently selected by the cursor by the step size in normal mode. If shift mode is turned on, this function increases the value of all selected fixed points by an amount equal to the set step size.

Parameter

Name	Type	Description
-	-	-

Return value

Type	Description
void	No return value.

Available: since version 12.6.0

4.5.2.10 DecreaseValueAtCursor

This function decreases the value of the point which is currently selected by the cursor by the step size.

If shift mode is turned on, this function decreases the value of all selected fixed points by an amount equal to the set step size.

Parameter

Name	Type	Description
-	-	-

Return value

Type	Description
void	No return value.

Available: since version 12.6.0

4.5.2.11 ActivateConfig

Adds the current configuration to the editor profiles list and generates a setpoint for the current data.

Parameter

Name	Type	Description
-	-	-

Return value

Type	Description
void	No return value.

Available: since version 12.6.0

4.5.2.12 ResetConfigdata

This function will revert any user changes by reloading the last activated configuration.

Parameter

Name	Type	Description
-	-	-

Return value

Type	Description
void	No return value.

Available: since version 12.6.0

4.5.2.13 AddNewFixPoint

This function adds a new fix point at the point selected by the cursor.

Parameter

Name	Type	Description
-	-	-

Return value

Type	Description
void	No return value.

Available: since version 12.6.0

4.5.2.14 RemoveSelectedFixPoint

This function removes the fixed point which is currently selected by the cursor; it will not change anything if no fixed point is selected.

Parameter

Name	Type	Description
-	-	-

Return value

Type	Description
void	No return value.

Available: since version 12.6.0

4.5.2.15 AddNewMarker

This function adds a new marker at the base location selected by the cursor.

Parameter

Name	Type	Description
-	-	-

Return value

Type	Description
void	No return value.

Available: since version 12.6.0

4.5.2.16 RemoveSelectedMarker

This function removes the marker selected by the cursor. User first needs to select the marker by touching on the screen area around it, and then call this function.

Parameter

Name	Type	Description
-	-	-

Return value

Type	Description
void	No return value.

Available: since version 12.6.0

4.5.3 Events

Name	Description
onNewSetPointsGenerated [▶ 65]	This event is fired when the setpoint output is changed.
onUserConfigActivated [▶ 65]	This event is generated when a new configuration is activated on the control.
onEditingCancelled [▶ 65]	This event is generated when user changes are canceled and the old configuration is reactivated.

4.5.3.1 onNewSetPointsGenerated

This event is fired when the setpoint output is changed.

Available: since version 12.6.0

4.5.3.2 onUserConfigActivated

This event is generated when a new configuration is activated on the control.

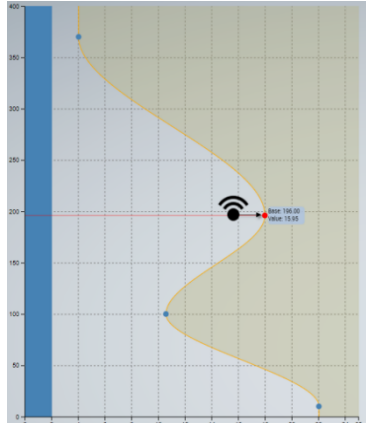
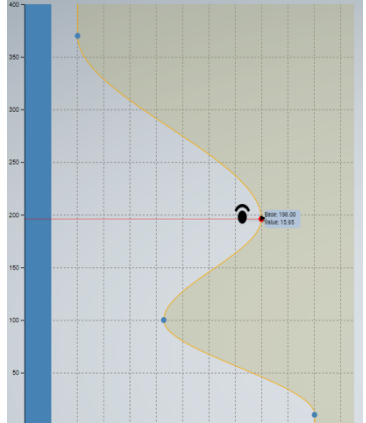
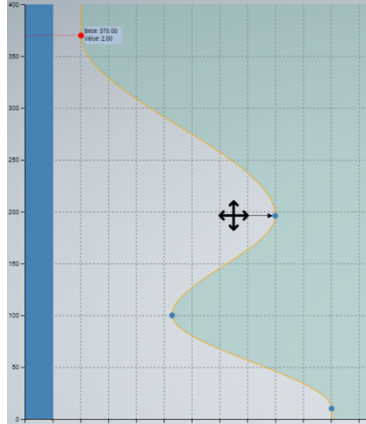
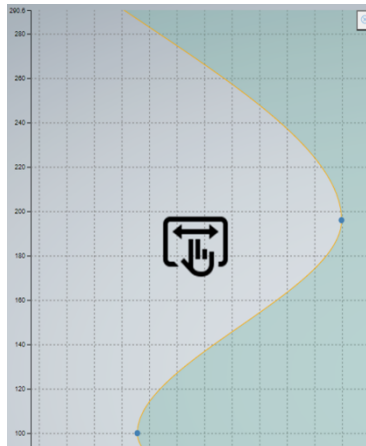
Available: since version 12.6.0

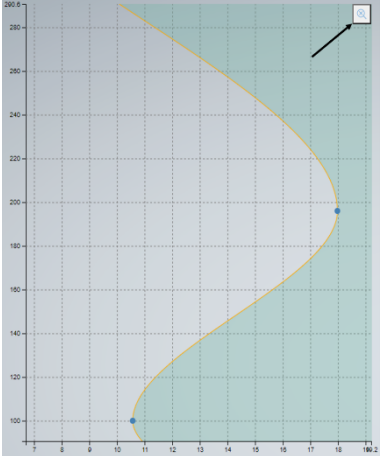
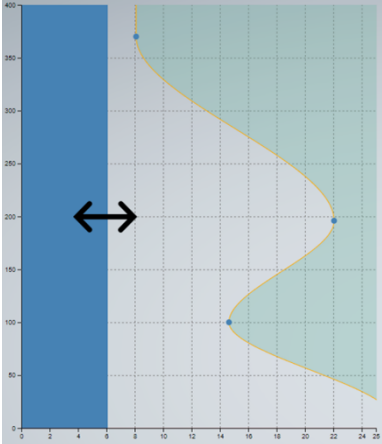
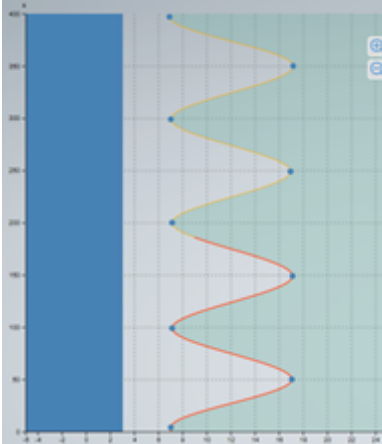
4.5.3.3 onEditingCancelled

This event is generated when user changes are canceled and the old configuration is reactivated.

Available: since version 12.6.0

4.5.4 User Interactions

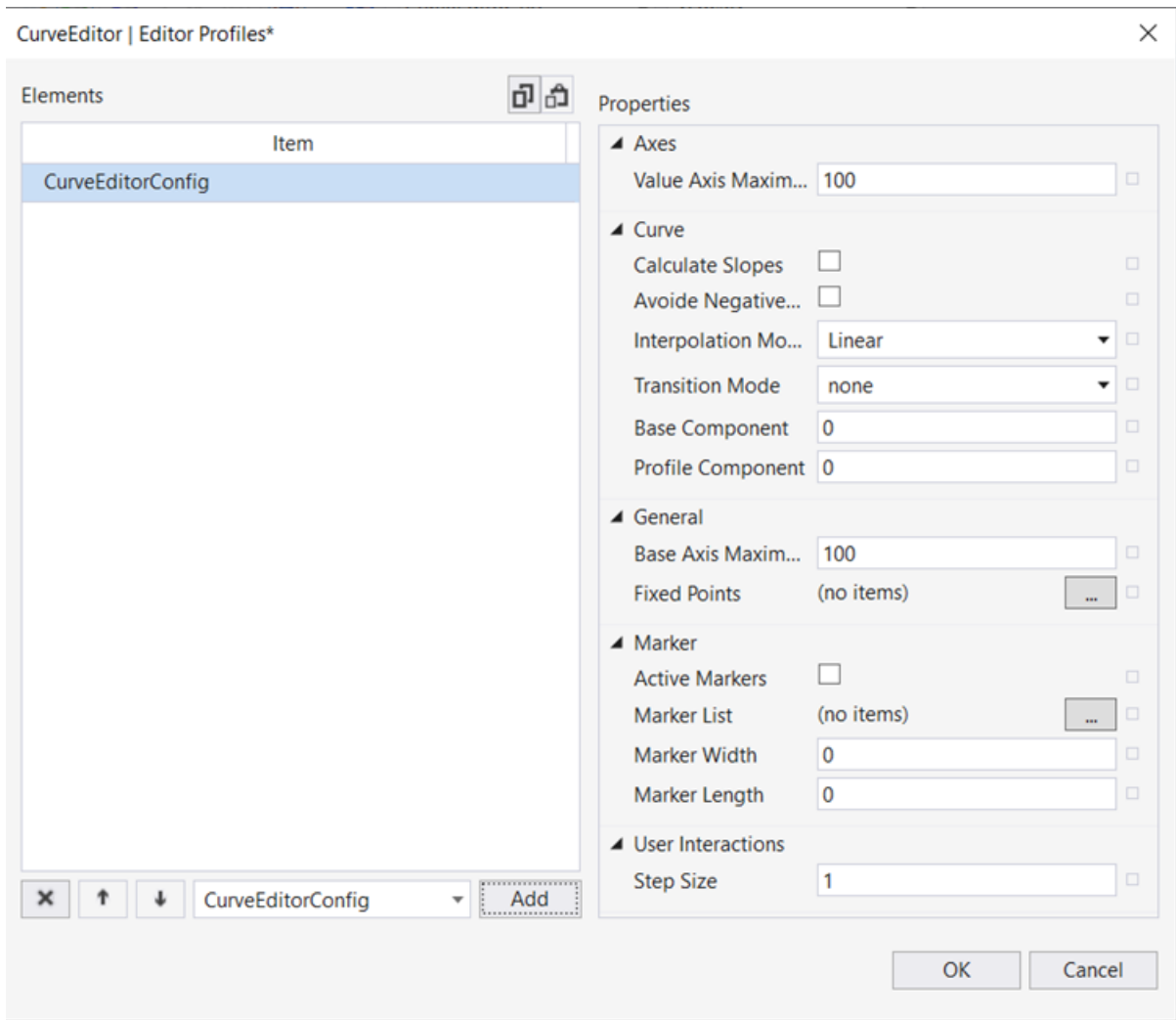
Name	Description	Result on the control
Double Tap	Once double tapped on the graph, a new fixed point will be added, which can be applied in the configuration by activating or resetting the edited configuration.	
Single Tap	When a single tap is done on a fixed point, the cursor value is updated and the updated cursor value of the fixed point is shown in the tool tip.	
Drag FixedPoint	Once a single fixed point is selected, it can be dragged up or down within the graph area.	
2-finger zoom in graph	Once the graph is configured, it can be zoomed in to check the minor curve settings. The zoom is done using two fingers to zoom in the graph. The graph shown is completely zoomed in.	

Name	Description	Result on the control
Zoom out Graph	<p>Once the graph is zoomed to get the graph in its default state, a zoom out button will appear in the top right corner.</p> <p>The graph will zoom out to the default position with a nice transaction animation effect.</p>	 <p>The image shows a graph with a yellow curve on a grid. The y-axis ranges from 100 to 300, and the x-axis ranges from 7 to 19.2. A small square button with a magnifying glass icon is located in the top right corner, with an arrow pointing to it.</p>
Base Component bar drag	<p>The base component bar can be dragged along the value axis. The base component will change accordingly and be added to all fixed points.</p>	 <p>The image shows the same graph as above, but with a blue vertical bar on the left side of the x-axis. A double-headed arrow indicates that this bar can be dragged horizontally along the x-axis.</p>
Zoom in / Zoom out using buttons	<p>The two buttons placed at the top right corner of the graph are used to zoom in and zoom out the graph manually.</p> <p>By clicking on the zoom in button, the graph will be zoomed in one step. Maximum 4x zoom is divided into 8 steps.</p> <p>By clicking on the zoom out button the graph will be zoomed out in one step.</p>	 <p>The image shows the graph with the blue bar and two buttons in the top right corner: a magnifying glass icon for zooming in and a square with a magnifying glass icon for zooming out.</p>

Available: since version 12.6.0

4.5.5 Step by step

1. Drag and drop the curve editor control from the toolbox window to your content.
2. Setup the editor profiles attribute.
 - Editor profiles has the highest priority of all attributes, it will be loaded first when the control initializes.
 - The active profile will overwrite other attributes set in it when loaded.
 - Editor profiles is an array of profiles; at least one profile should be configured to use the control.
 - If you want the curve editor to start every time with a static profile, you can program the object in the TwinCAT HMI object editor window.



3. Setting the base axis position.

- The base axis position attribute can be set to vertical or horizontal. This will define how the control looks and behaves.
- To provide unit conversion, display unit detection and to display unit text on the value axis, the curve editor control relies on unit conversion functions provided by the `Beckhoff.TwinCAT.HMI.Plastic.Controls` packages and the configurator control.

4. Set value axis data type.

- You can configure the `unit_config.json` file to include the unit group in the file with the related display and system unit configuration.
- The configurator control must be instantiated with the default ID at uppermost level in the project. (First control in `Desktop.view`)

i It is possible to use the curve editor without the unit conversion provided by the Plastic package.

Set an empty string as the value axis data type. This way the control will continue to function without any unit conversion.

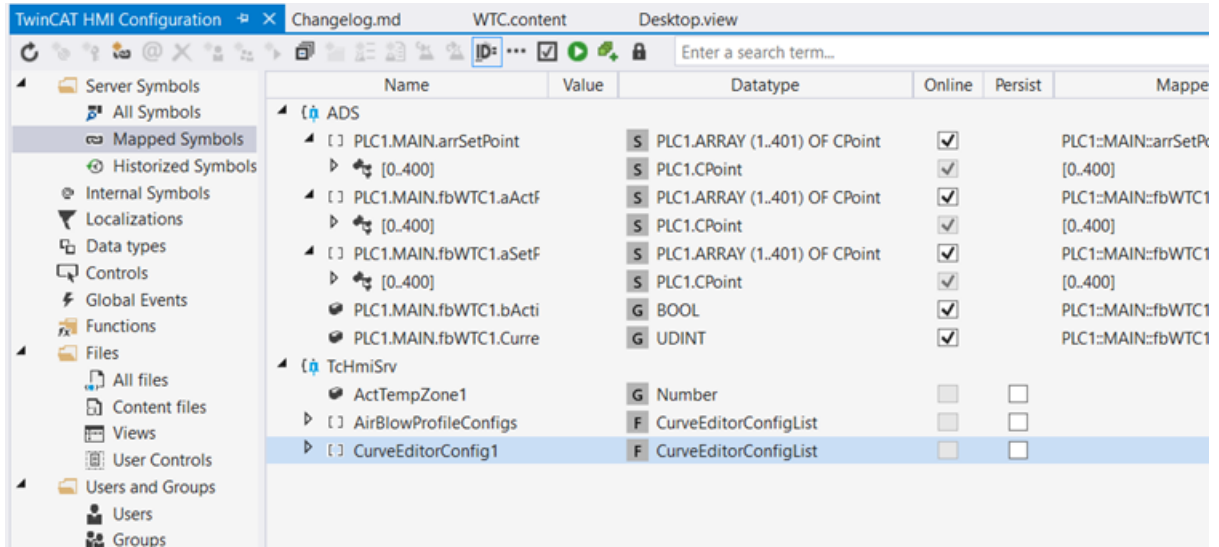
5. Set all other attributes or bind other attributes to other controls or server symbols.

4.5.6 Using Recipe Management with the Curve Editor

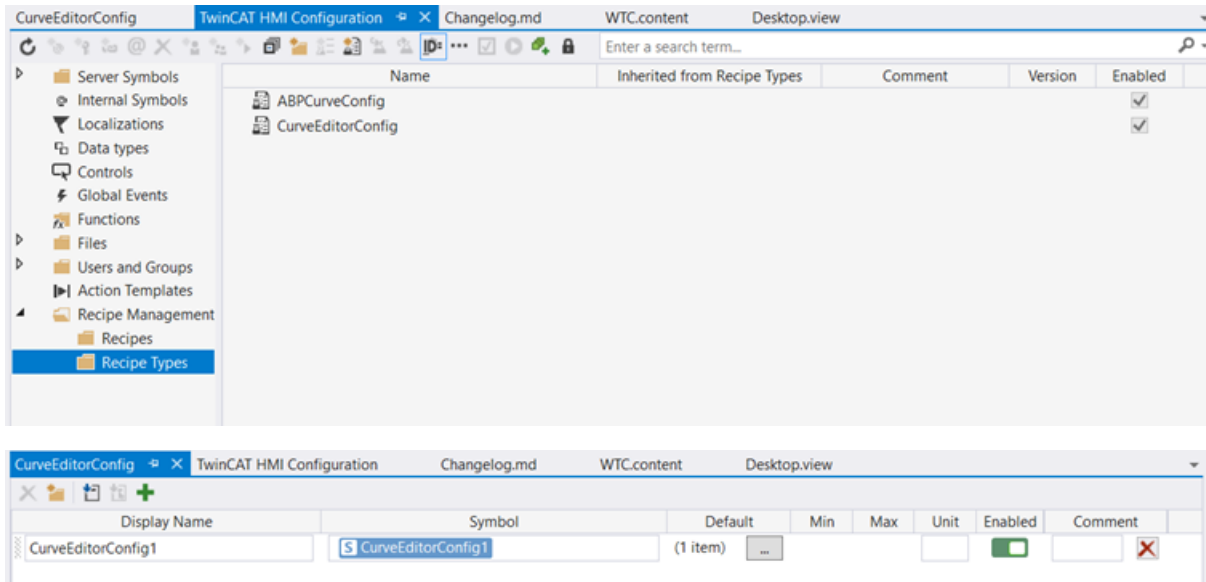
The curve editor allows multiple profiles to be saved to a list of profiles. This profile list can be stored in recipes.

The following steps should be taken to use the TwinCAT Recipe Management feature with the curve editor.

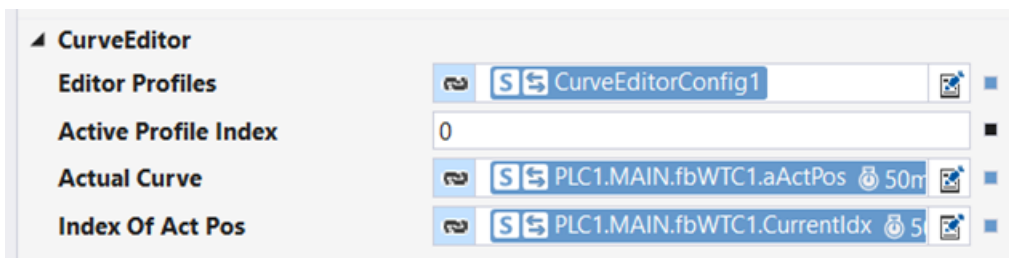
1. Create a server symbol, which will be used as the source of the editor profiles list for the controller. Each curve editor control will require one symbol of the type CurveEditorConfigList.



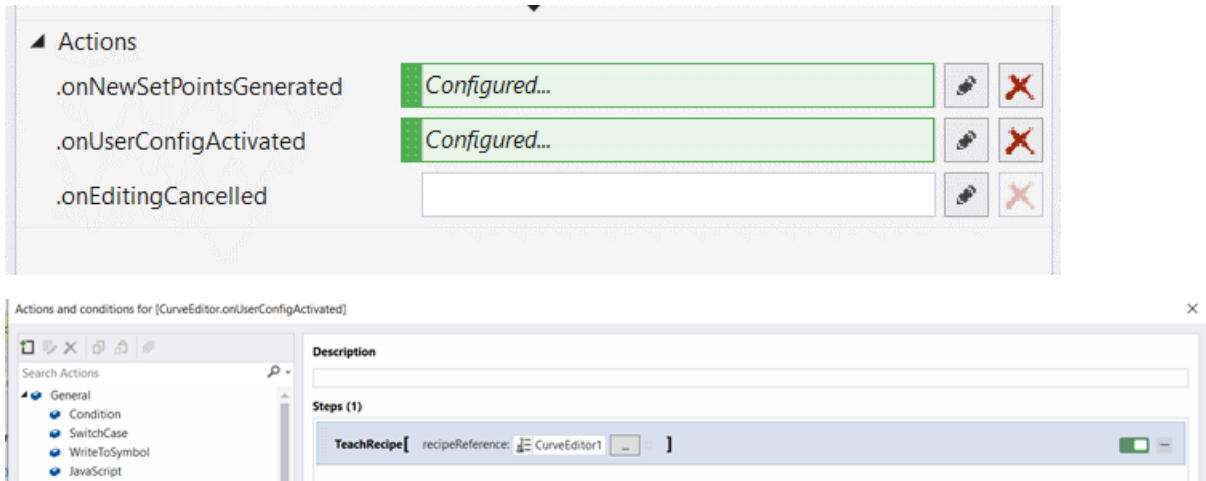
2. Add this server symbol to a recipe type. It can be an individual recipe type or grouped with other symbols in the product recipe type.



3. Use this server symbol as the configuration symbol for the Editor Profiles [▶ 49] property. Having two-way binding is the easiest way to update the symbol if there is any change in the configuration list.



- With two-way binding, the server symbol will be updated on every change. And the recipe teach function can be called on `.onUserConfigActivated` event.



4.6 FavoriteBarControl

NuGet package: Beckhoff.TwinCAT.HMI.Plastic.Controls

Available: since version 12.6.0

The FavoriteBarControl can be used with the [Responsive Navigation Bar](#) control in TwinCAT HMI. The purpose of this control is to allow users to easily set any content as a favorite, creating a shortcut that is always available on the FavoriteBarControl.

The FavoriteBarControl will work by allowing users to select current content displayed on the target host as a favorite by clicking on the **Add to Favorites** button which is displayed with a star symbol. Once added, the content will appear as a shortcut on the FavoriteBarControl, allowing the user to quickly navigate to it at any time.

The FavoriteBarControl is highly customizable, with options to show detail paths or icons for each selected content.

It will help to improve the usability and accessibility of TwinCAT HMI, allowing users to easily access their favorite content and streamline their workflow.



4.6.1 Attributes

Category: Common

Name	Description
FavContentList [▶ 73]	Default list of contents that are displayed on the control.
RegionControlSymbol [▶ 75]	Target region on which a favorite content should be displayed.
MenuDataSrc [▶ 74]	MenuDataSrc of a ResponsiveNavigation.TcHmiNavigationBar control where more information of favorite control such as icon, descriptor etc. is available.
DisplayContentPath [▶ 72]	User can select if the complete path of each favorite content is displayed on the control.
FavButtonPosition [▶ 72]	Position of Add to Favorites and Go Back buttons.

Category: Text

Name	Description
TextFontSize [▶ 75]	Font size for the text displayed.
FontSizeUnit [▶ 75]	Sets the font size unit.

Category: Image

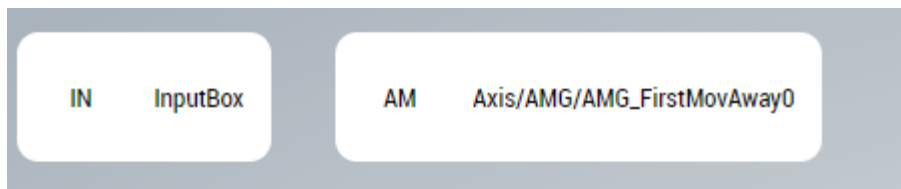
Name	Description
ImageHeight [▶ 74]	Height of any image or icon displayed on a favorite button.
ImageHeightUnit [▶ 74]	Unit for image height.
ImageWidth [▶ 74]	Width of any image or icon displayed on a favorite button.
ImageWidthUnit [▶ 74]	Unit for image width

4.6.1.1 DisplayContentPath

When a user adds a new favorite item, the control can check if an image has been configured for that content in [MenuSrcData \[▶ 74\]](#). If an image is available, the control will display that image on the favorite button. If no image has been configured, the control will display two-letter initials to provide a visual representation of the content.

Additionally, the `DisplayContentPath` attribute is a useful option for users who want to see the complete path configured for their content in the [MenuSrcData \[▶ 74\]](#).

When this attribute is set to `TRUE`, the full path will be displayed on the favorite button alongside the icon or initials.

Favorite buttons with `DisplayContentPath` set to `TRUE`:

Schema: `tchmi:framework#/definitions/TchHmi.Controls.ResponsiveNavigation.TchHmiNavigationBar.MenuItemLitst`

Attribute setter: `setDisplayContentPath`

Attribute getter: `getDisplayContentPath`

Available: since version 12.6.0

4.6.1.2 FavButtonPosition

User can select if the 'Add To Favorite' and 'Go Back' buttons are positioned to the left or the right end of the control.

Schema: `tchmi:framework#/definitions/FavButtonPosition`

Attribute setter: `setFavButtonPosition`

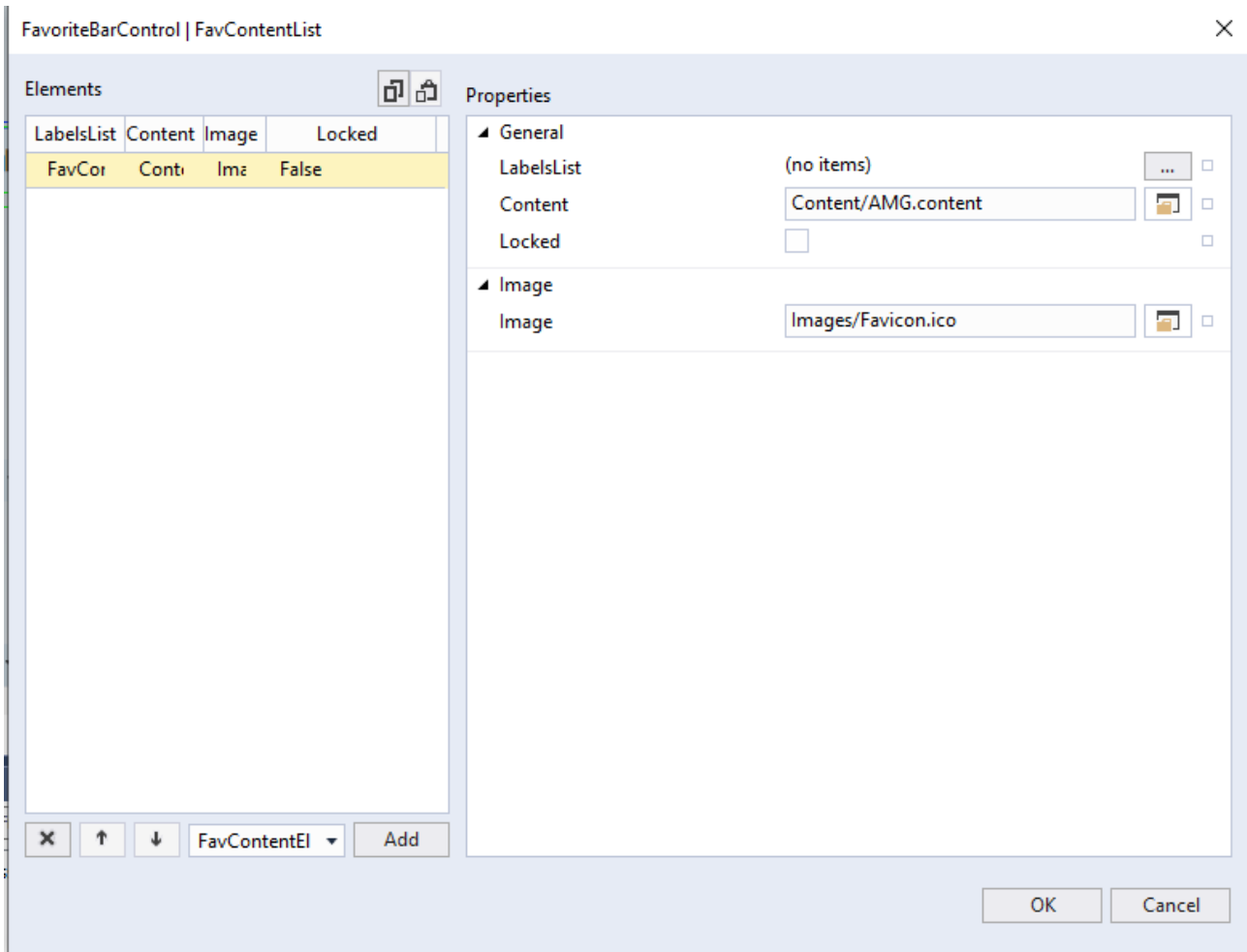
Attribute getter: `getFavButtonPosition`

Available: since version 12.6.0

4.6.1.3 FavContentList

The FavContentList attribute will allow users to configure a default list of contents that will be displayed on the FavoriteBarControl. This means that when the HMI is launched, the user will see a pre-defined set of favorite shortcuts that they can use to quickly navigate to their most commonly accessed content.

A dialog box can be used to configure default list of favorites:



Configuring each favorite element:

Name	Description
LabelsList	If a content path has to be displayed on the favorite button it can be configured as a list of strings. All strings will be joined together with '/' characters between them to form the content path.
Content	The path to the content file
Locked	If a favorite element is locked it cannot be removed from the favorite list by user clicking on 'Add to Favorite' button.
Image	Path to an image to display it as an icon of the favorite button.

Schema: tchmi:framework#/definitions/FavContentElementList

Attribute setter: setFavContentList

Attribute getter: getFavContentList

Available: since version 12.6.0

4.6.1.4 ImageHeight

The height of the images that are displayed on the favorite buttons is set with this attribute.

Schema: tchmi:framework#/definitions/MeasurementValue

Attribute setter: setImageHeight

Attribute getter: getImageHeight

Available: since version 12.6.0

4.6.1.5 ImageHeightUnit

The unit of the height of the images that are displayed on the favorite buttons is set with this attribute.

Schema: tchmi:framework#/definitions/MeasurementValue

Attribute setter: setImageHeightUnit

Attribute getter: getImageHeightUnit

Available: since version 12.6.0

4.6.1.6 ImageWidth

The width of the images that are displayed on the favorite buttons is set with this attribute.

Schema: tchmi:framework#/definitions/MeasurementValue

Attribute setter: setImageWidth

Attribute getter: getImageWidth

Available: since version 12.6.0

4.6.1.7 ImageWidthUnit

The unit of the width of the images that are displayed on the favorite buttons is set with this attribute.

Schema: tchmi:framework#/definitions/MeasurementValueUnit

Attribute setter: setImageWidthUnit

Attribute getter: getImageWidthUnit

Available: since version 12.6.0

4.6.1.8 MenuDataSrc

If a user adds a new favorite content from the screen, the favorite bar needs more information about the content such as label descriptors, image path etc.

By linking the navigation bar's menu source to the favorite bar control, this information can be made available to the favorite bar control.

The benefit of this approach is that the user can configure all their navigation options in the TcHmiNavigationBar control, and the FavoriteBarControl can read that configuration when adding a new favorite content.

Schema: tchmi:framework#/definitions/
TcHmi.Controls.ResponsiveNavigation.TcHmiNavigationBar.MenuItemLitst

Attribute setter: setMenuDataSrc

Attribute getter: getMenuDataSrc

Available: since version 12.6.0

4.6.1.9 RegionControlSymbol

The control symbol of the target region where the selected content should be displayed.

Schema: tchmi:framework#/definitions/TcHmi.Controls.ResponsiveNavigation.TcHmiNavigationBar.TargetRegionControlSymbol

Attribute setter: setFavContentList

Attribute getter: getFavContentList

Available: since version 12.6.0

4.6.1.10 TextFontSize

Sets the font size of the text displayed.

Schema: tchmi:general#/definitions/UINT

Attribute getter: getTextFontSize

Attribute setter: setTextFontSize

Available: since version 12.6.0

4.6.1.11 FontSizeUnit

Sets the font size unit.

Schema: tchmi:framework#/definitions/MeasurementUnit

Attribute getter: getFontSizeUnit

Attribute setter: setFontSizeUnit

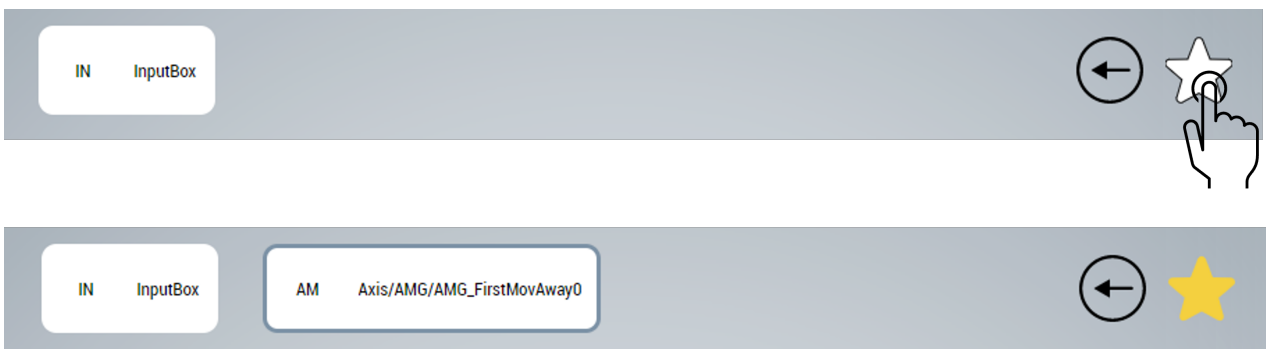
Available: since version 12.6.0

4.6.2 User Interactions

4.6.2.1 Add or remove a content as favorite on a favorite bar

Once a user navigates to a content on the target region control, the same user can add this content to the favorite bar by clicking on the 'Add to Favorite' button which is shown as a star.

This will add the current content to the favorite bar and change the appearance of the **Add to Favorite** button from a white star to a yellow filled star.



If a user adds a content as a favorite, every time the same user navigates to that content the **Add to Favorite** button will be displayed as a yellow filled star. With this feedback the user knows that the content is marked as a favorite.

The process to remove a content from the favorite bar is similar. If the user navigates to a content that was previously added to the favorite bar and clicks on the **Add to Favorite** button then this content will be removed from the favorite bar.

4.6.2.2 Go back to last content

The FavoriteBarControl also provides the users the ability to reload the last visited contents. To go back to the last navigated content the user can click on **Go Back** button on the control.

The user can navigate back to up to 9 visited contents.

4.7 InputBox

NuGet package: Beckhoff.TwinCAT.HMI.Plastic.Controls

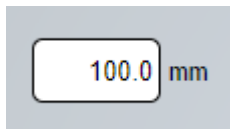
Available: since version 12.6.0

InputBox is a control that allows the user to input numerical or boolean values in the TwinCAT HMI project.

Key Features

- Range validation
- Unit conversion of physical quantities
- Visualization according to data type (BOOL, INT, FLOAT)
- Input filter that only allows numerical input

InputBox for FLOAT values:



InputBox for BOOL values:



4.7.1 Attributes

Category: Colors

Name	Description
TextColor [▶ 80]	The font color of the text displayed in the normal state.

Category: Common

Name	Description
UnitText [▶ 81]	Returns the unit text or an abbreviation string for the current display unit.

Category: Inputbox

Name	Description
DataType [▶ 78]	The data type of the control value.
IsReadOnly [▶ 79]	If <code>TRUE</code> the control is read only.
UnitGroup [▶ 81]	The unit group of the control value.
StepSize [▶ 80]	The step size to increment and decrement the value using IncreaseValueByStepSize [▶ 81] and DecreaseValueByStepSize [▶ 82] functions with the buttons on the numpad.

Category: Text

Name	Description
TextFontSize [▶ 80]	The font size of the displayed text.
DigitsAfterDecimal [▶ 78]	The displayed number of digits after the decimal point for the <code>FloatDataType</code> .
DisplayUnitText [▶ 78]	Displays the configured display unit text next to the value.
TextHorizontalAlignment [▶ 81]	Horizontal alignment for the TextVal [▶ 81] attribute.
ContentPadding [▶ 77]	Padding for the TextVal [▶ 81] attribute.

Category: Recipe management

Name	Description
RecipeType [▶ 79]	Adds the TextVal [▶ 81] symbol to the recipe type if the <code>RecipeHelper</code> NuGet package is used and the value is set to 'R'.
ReadbackGroup [▶ 79]	Enter if the control belongs to a group managed by recipe management.

Category: Common

Name	Description
MinVal [▶ 79]	Minimum value of the TextVal [▶ 81] attribute.
MaxVal [▶ 79]	Maximum value of the TextVal [▶ 81] attribute.
RefVal [▶ 80]	Reference value of the TextVal [▶ 81] attribute which is used for the display unit %.
TextVal [▶ 81]	The value to be displayed in the control.

Category: Fixed-Units

Name	Description
DisplayUnit [▶ 78]	Sets a fixed unit as the display unit for the control. The UnitGroup [▶ 81] attribute must be an empty string to use this. Only applicable if the DataType [▶ 78] attribute is set to <code>FloatDataType</code> .
SystemUnit [▶ 80]	Sets a fixed unit as the system unit for the control. The UnitGroup [▶ 81] attribute must be empty string to use this. Only applicable if the DataType [▶ 78] attribute is set to <code>FloatDataType</code> .

4.7.1.1 ContentPadding

This attribute sets the padding of the text value inside the input box. Padding is the distance from the text to the border, it can be set individually on all 4 sides of the text.

Schema: `tchmi:framework#/definitions/Padding`

Attribute getter: getContentPadding

Attribute setter: setContentPadding

Available: since version 12.6.0

4.7.1.2 **DataType**

The desired data type of the control: `BoolDataType`, `IntDataType`, `FloatDataType` or `TimeDataType`.

Schema: tchmi:framework#/definitions/FBValue_Type

Attribute getter: getDataType

Attribute setter: setDataType

Available: since version 12.6.0

4.7.1.3 **DigitsAfterDecimal**

Number of digits displayed after the decimal point for the `FloatDataType`.

Schema: tchmi:general#/definitions/UINT

Attribute getter: getDigitsAfterDecimal

Attribute setter: setDigitsAfterDecimal

Available: since version 12.6.0

4.7.1.4 **DisplayUnit**

Sets a fixed unit as the display unit for the control. The [UnitGroup \[▶ 81\]](#) attribute must be an empty string to use this. Only applicable if the [DataType \[▶ 78\]](#) attribute is set to `FloatDataType`.

Schema: tchmi:general#/definitions/String

Attribute getter: getDispUnit

Attribute setter: setDispUnit

Available: since version 12.6.0

4.7.1.5 **DisplayUnitText**

If this attribute is set to `TRUE`, the unit text of the current variable unit will be displayed next to the input box.

Schema: tchmi:general#/definitions/BOOL

Attribute getter: getDisplayUnitText

Attribute setter: setDisplayUnitText

Available: since version 12.6.0

4.7.1.6 **FontSizeUnit**

Sets the font size unit.

Schema: tchmi:framework#/definitions/MeasurementUnit

Attribute getter: getFontSizeUnit

Attribute setter: setFontSizeUnit

Available: since version 12.6.0

4.7.1.7 **IsReadOnly**

When `TRUE`, the control does not allow user interactions.

Schema: tchmi:general#/definitions/BOOL

Attribute getter: getReadOnly

Attribute setter: setReadOnly

Available: since version 12.6.0

4.7.1.8 **MaxVal**

The maximum limit of the text value that can be entered.

Schema: tchmi:framework#/definitions/ValueAttribute

Attribute getter: getMaxVal

Attribute setter: setMaxVal

Available: since version 12.6.0

4.7.1.9 **MinVal**

The minimum limit of the text value that can be entered.

Schema: tchmi:framework#/definitions/ValueAttribute

Attribute getter: getMinVal

Attribute setter: setMinVal

Available: since version 12.6.0

4.7.1.10 **ReadbackGroup**

Enter if the control belongs to a group managed by recipe management.

Schema: tchmi:general#/definitions/String

Attribute getter: getReadbackGroup

Attribute setter: setReadbackGroup

Available: since version 12.6.0

4.7.1.11 **RecipeType**

If this attribute is set to 'R', the text value symbol is added to the recipe type, when it is used in a project where the `Beckhoff.TwinCAT.HMI.Plastic.RecipeHelper` NuGet package is installed.

Schema: tchmi:framework#/definitions/RecipeType

Attribute getter: getRecipeType

Attribute setter: setRecipeType

Available: since version 12.6.0

4.7.1.12 RefVal

The reference value is required if the display unit is percent [%]. This attribute value will be used to calculate 100% reference.

Schema: tchmi:framework#/definitions/ValueAttribute

Attribute getter: getRefVal

Attribute setter: setRefVal

Available: since version 12.6.0

4.7.1.13 StepSize

The step size to increment and decrement the value using [IncreaseValueByStepSize \[▶ 81\]](#) and [DecreaseValueByStepSize \[▶ 82\]](#) functions with the buttons on the numpad.

Schema: tchmi:general#/definitions/Number

Attribute getter: getStepSize

Attribute setter: setStepSize

Available: since version 12.6.0

4.7.1.14 SystemUnit

Sets a fixed unit as the system unit for the control. The [UnitGroup \[▶ 81\]](#) attribute must be empty string to use this. Only applicable if the [DataType \[▶ 78\]](#) attribute is set to `FloatDataType`.

Schema: tchmi:general#/definitions/String

Attribute getter: getSysUnit

Attribute setter: setSysUnit

Available: since version 12.6.0

4.7.1.15 TextColor

The text color in the normal state of the text type control which does not affect the check type of the control.

Schema: tchmi:framework#/definitions/SolidColor

Attribute getter: getTextColor

Attribute setter: setTextColor

Available: since version 12.6.0

4.7.1.16 TextFontSize

Sets the font size of the value.

Schema: tchmi:general#/definitions/UINT

Attribute getter: getTextFontSize

Attribute setter: setTextFontSize

Available: since version 12.6.0

4.7.1.17 TextHorizontalAlignment

This attribute sets the horizontal alignment of the value text. The text can be placed in the center, left or right within the input box.

Schema: tchmi:framework#/definitions/HorizontalAlignment

Attribute getter: getTextHorizontalAlignment

Attribute setter: setTextHorizontalAlignment

Available: since version 12.6.0

4.7.1.18 TextVal

The text value of a text type control and the checked state value for a bool type input control.

Schema: tchmi:framework#/definitions/ValueAttribute

Attribute getter: getTextVal

Attribute setter: setTextVal

Available: since version 12.6.0

4.7.1.19 UnitText

This read only attribute returns a unit text or an abbreviation string for the current display unit of the control.

Schema: tchmi:general#/definitions/Any

Attribute getter: getUnitsText

Available: since version 12.6.0

4.7.1.20 UnitGroup

The unit group of a `FloatDataType` value. The string value for the unit group must be configured in the [Unit_Config.json](#) [▶ 40] file. The values will be displayed in the display unit and changes will be written to the PLC in the system unit.

Schema: tchmi:general#/definitions/String

Attribute getter: getUnitGroup

Attribute setter: setUnitGroup

Available: since version 12.6.0

4.7.2 Functions

Name	Description
IncreaseValueByStepSize [▶ 81]	This function increases and updates the text value by the given step size.
DecreaseValueByStepSize [▶ 82]	This function decreases and updates the text value by the given step size.

4.7.2.1 IncreaseValueByStepSize

This function increases and updates the text value by the given step size.

Parameter

Name	Type	Description
-	-	-

Return value

Type	Description
void	No return value.

Available: since version 12.6.0

4.7.2.2 DecreaseValueByStepSize

This function decreases and updates the text value by the given step size.

Parameter

Name	Type	Description
-	-	-

Return value

Type	Description
void	No return value.

Available: since version 12.6.0

4.7.3 Events

Category: Control

Name	Description
onUserInteractionAction [▶ 82]	This event is fired when the text is changed due to a user interaction.

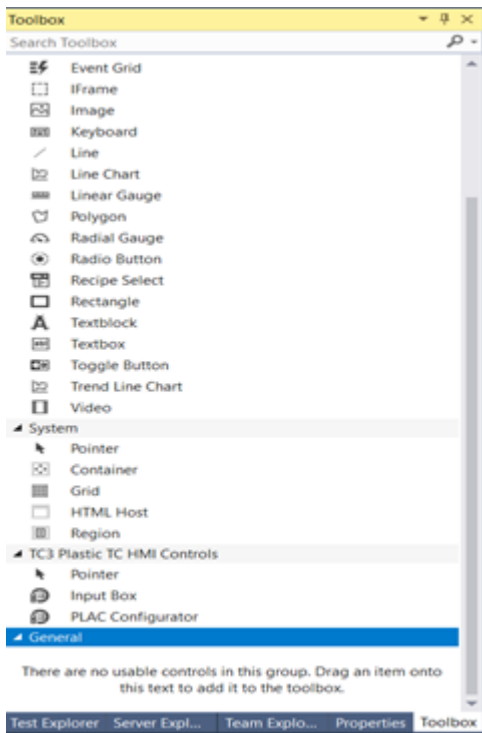
4.7.3.1 onUserInteractionAction

This event is fired when the [TextVal](#) [[▶ 81](#)] attribute is changed due to a user interaction.

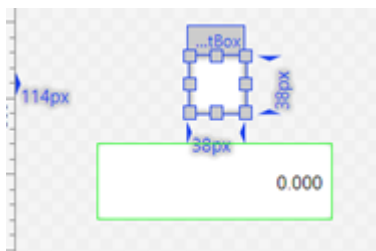
Available: since version 12.6.0

4.7.4 Step by step

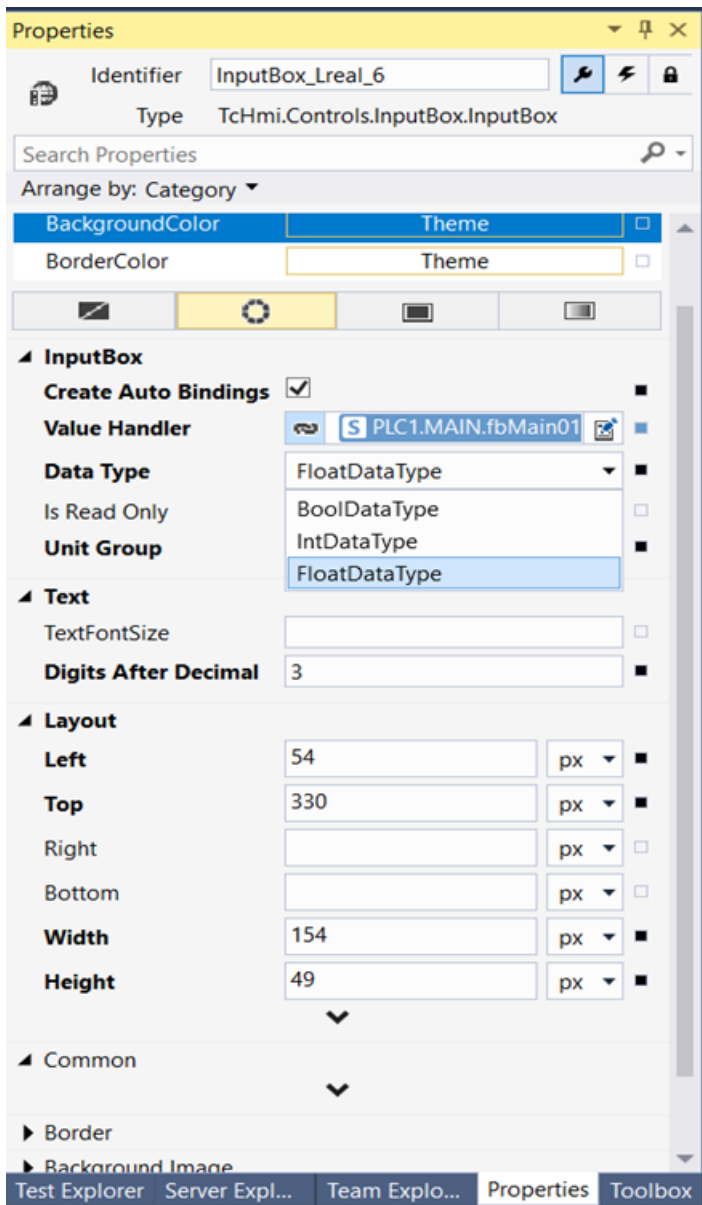
1. Drag and drop the InputBox from the Toolbox window onto the content or view file.



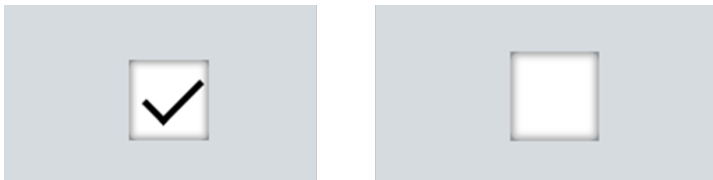
2. If the control `DataType [▶ 78]` attribute is set to `BoolDataType`, its width and height must be equal for proper display.



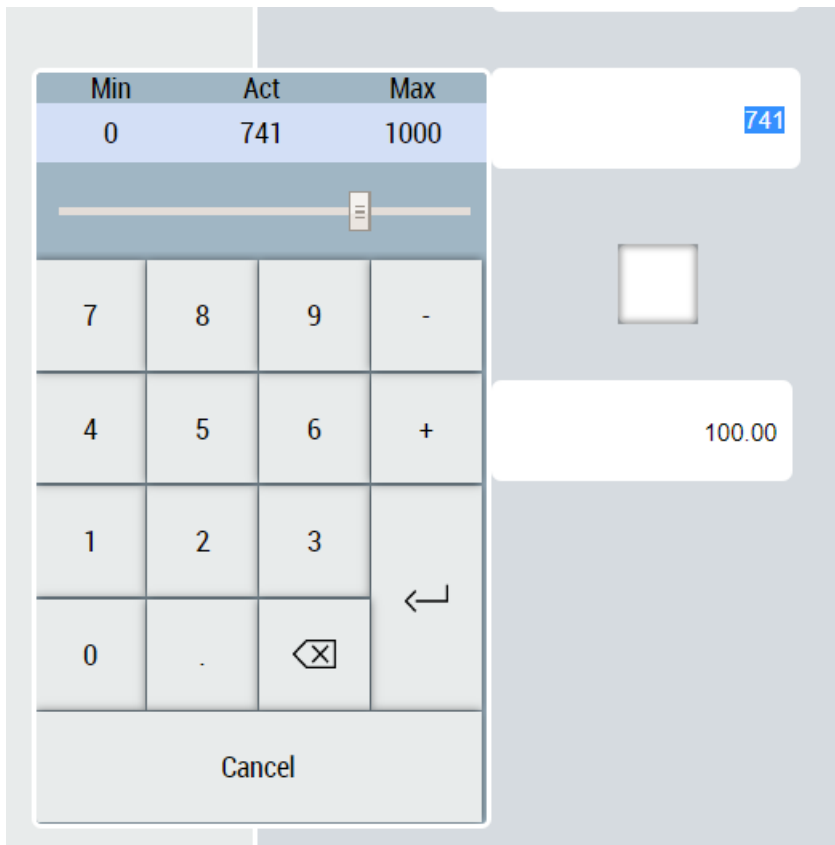
3. Set the `DataType` [▶ 78] attribute of the type `FBValue_DataType`, it will set how the control looks and functions.



If the Data Type is `BoolDataType` the control will form a Checkbox, with a checkmark matching the boolean state of the related `TextVal` [▶ 81] attribute.



For `IntDataType`, `FloatDataType` or `TimeDataType` the control will show as a textbox, with numbers from its `TextVal` [▶ 81] attribute displayed in decimal format. The `InputPanel` will appear when the user clicks/taps on the control.

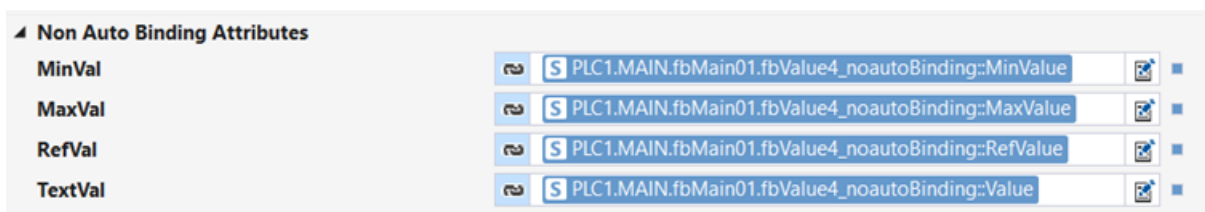


If data type is set as either `FloatDataType` or `TimeDataType`, numbers will be displayed in the display unit of the given data type.

The unit text is displayed next to the control if the `DisplayUnitText` [▶ 78] attribute is set to `TRUE`.



- The attributes `TextVal` [▶ 81], `RefVal` [▶ 80], `MinVal` [▶ 79] and `MaxVal` [▶ 79] can be mapped to any number or boolean symbol in the project.



4.8 ManualOperation

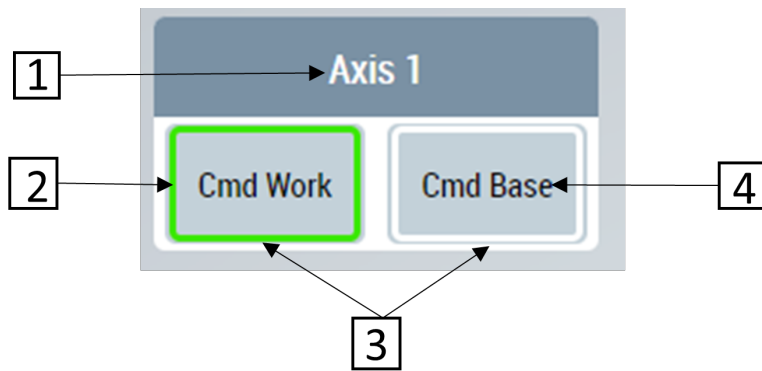
NuGet package: Beckhoff.TwinCAT.HMI.Plastic.Controls

Available: since version 12.6.0

ManualOperation control can be used as on-screen touch buttons for moving axes manually. The axes can be moved in two directions:

- Towards base position
- Towards work position

The control provides two sets of buttons and status indicators for each direction.



1. Name of the axis that can be configured using the attribute
2. The light ring around the button will illuminate in proper color
3. User can click on buttons to send command to move in either direction
4. Text displayed in each button

4.8.1 Attributes

Category: Parameters

Name	Description
ManualOperationSymbol [▶ 92]	Symbol of type FB_ManualOperation.
AxisName [▶ 87]	Axis name displayed on the control.
BtnWorkText [▶ 91]	Text property of the button that is used to send command towards work position.
BtnBaseText [▶ 90]	Text property of the button that is used to send command towards base position.
SingleMode [▶ 94]	To set the control in a single direction mode.

Category: Button Configuration

Name	Description
BorderRadius [▶ 90]	Sets the border radius for buttons on the control.
ButtonOrientation [▶ 91]	Places buttons horizontally or vertically next to each other.

Category: Button Work Background Image

Name	Description
BackgroundImageWork [▶ 90]	Sets the background image for the work direction button.
BackgroundImageWidthWork [▶ 90]	Sets the image width for the work direction button.
BackgroundImageWidthWorkUnit [▶ 90]	Sets the background image width unit for the work button.
BackgroundImageHeightWork [▶ 88]	Sets the image height for the work direction button.
BackgroundImageHeightWorkUnit [▶ 88]	Sets the background image height unit for the work button.
BackgroundImageHorizontalAlignWork [▶ 88]	Sets the horizontal alignment of the image for the work direction button.
BackgroundImageVerticalAlignWork [▶ 89]	Sets the vertical alignment of the image for the work direction button.
BackgroundImagePaddingWork [▶ 89]	Sets the image padding for the work direction button.

Category: Button Base Background Image

Name	Description
BackgroundImageBase [► 87]	Sets the background image for the base direction button.
BackgroundImageWidthBase [► 89]	Sets the image width for the base direction button.
BackgroundImageWidthBaseUnit [► 90]	Sets the background image width unit for the base button.
BackgroundImageHeightBase [► 87]	Sets the image height for the base direction button.
BackgroundImageHeightBaseUnit [► 88]	Sets the background image height unit for the base button.
BackgroundImageHorizontalAlignBase [► 88]	Sets the horizontal alignment of the image for the base direction button.
BackgroundImageVerticalAlignBase [► 89]	Sets the vertical alignment of the image for the base direction button.
BackgroundImagePaddingBase [► 89]	Sets the image padding for the base direction button.

Category: Colors

Name	Description
FaultColor [► 92]	Light ring color when FB_ManualOperation status is 'in fault'.
InPosColor [► 92]	Light ring color when FB_ManualOperation status is 'in position'.
MovingToPosColor [► 93]	Light ring color when FB_ManualOperation status is 'moving to position'.

4.8.1.1 AxisName

This attribute sets the axis name to the control.

Schema: tchmi:general#/definitions/String

Attribute getter: getAxisName

Attribute setter: setAxisName

Available: since version 12.6.0

4.8.1.2 BackgroundImageBase

Path to an image to be set as a background image for a button to move the axis in the base direction.

Schema: tchmi:framework#/definitions/Path

Attribute getter: getBackgroundImageBase

Attribute setter: setBackgroundImageBase

Available: since version 12.6.0

4.8.1.3 BackgroundImageHeightBase

Definition of the height of the background image for a button to move the axis in the base direction.

Schema: tchmi:framework#/definitions/MeasurementValue

Attribute getter: getBackgroundImageHeightBase

Attribute setter: setBackgroundImageHeightBase

Available: since version 12.6.0

4.8.1.4 BackgroundImageHeightBaseUnit

Sets the background image height unit for the base button.

Schema: tchmi:framework#/definitions/MeasurementUnit

Attribute getter: getBackgroundImageHeightBaseUnit

Attribute setter: setBackgroundImageHeightBaseUnit

Available: since version 12.6.0

4.8.1.5 BackgroundImageHeightWork

Definition of the height of the background image for a button to move the axis in the work direction.

Schema: tchmi:framework#/definitions/MeasurementValue

Attribute getter: getBackgroundImageHeightWork

Attribute setter: setBackgroundImageHeightWork

Available: since version 12.6.0

4.8.1.6 BackgroundImageHeightWorkUnit

Sets the background image height unit for the work button.

Schema: tchmi:framework#/definitions/MeasurementUnit

Attribute getter: getBackgroundImageHeightWorkUnit

Attribute setter: setBackgroundImageHeightWorkUnit

Available: since version 12.6.0

4.8.1.7 BackgroundImageHorizontalAlignBase

Definition of the horizontal alignment of the background image for a button to move the axis in the base direction.

Schema: tchmi:framework#/definitions/HorizontalAlignment

Attribute getter: getBackgroundImageHorizontalAlgBase

Attribute setter: setBackgroundImageHorizontalAlgBase

Available: since version 12.6.0

4.8.1.8 BackgroundImageHorizontalAlignWork

Definition of the horizontal alignment of the background image for a button to move the axis in the work direction.

Schema: tchmi:framework#/definitions/HorizontalAlignment

Attribute getter: getBackgroundImageHorizontalAlgWork

Attribute setter: setBackgroundImageHorizontalAlgWork

Available: since version 12.6.0

4.8.1.9 BackgroundImagePadding Base

Definition of an additional distance between background image and the border of the button to move the axis in the base direction for the four sides.

Schema: tchmi:framework#/definitions/Padding

Attribute getter: getBackgroundImagePaddingBase

Attribute setter: setBackgroundImagePaddingBase

Available: since version 12.6.0

4.8.1.10 BackgroundImagePadding Work

Definition of an additional distance between background image and the border of the button to move the axis in the work direction for the four sides.

Schema: tchmi:framework#/definitions/Padding

Attribute getter: getBackgroundImagePaddingWork

Attribute setter: setBackgroundImagePaddingWork

Available: since version 12.6.0

4.8.1.11 BackgroundImageVerticalAlignBase

Definition of the vertical alignment of the background image for a button to move the axis in the base direction.

Schema: tchmi:framework#/definitions/VerticalAlignment

Attribute getter: getBackgroundImageVerticalAlgBase

Attribute setter: setBackgroundImageVerticalAlgBase

Available: since version 12.6.0

4.8.1.12 BackgroundImageVerticalAlignWork

Definition of the vertical alignment of the background image for a button to move the axis in the work direction.

Schema: tchmi:framework#/definitions/VerticalAlignment

Attribute getter: getBackgroundImageVerticalAlgWork

Attribute setter: setBackgroundImageVerticalAlgWork

Available: since version 12.6.0

4.8.1.13 BackgroundImageWidthBase

Definition of the width of the background image for a button to move the axis in the base direction.

Schema: tchmi:framework#/definitions/MeasurementValue

Attribute getter: getBackgroundImageWidthBase

Attribute setter: setBackgroundImageWidthBase

Available: since version 12.6.0

4.8.1.14 BackgroundImageWidthBaseUnit

Sets the background image width unit for the base button.

Schema: tchmi:framework#/definitions/MeasurementUnit

Attribute getter: getBackgroundImageWidthBaseUnit

Attribute setter: setBackgroundImageWidthBaseUnit

Available: since version 12.6.0

4.8.1.15 BackgroundImageWidthWork

Definition of the width of the background image for a button to move the axis in the work direction.

Schema: tchmi:framework#/definitions/MeasurementValue

Attribute getter: getBackgroundImageWidthWork

Attribute setter: setBackgroundImageWidthWork

Available: since version 12.6.0

4.8.1.16 BackgroundImageWidthWorkUnit

Sets the background image width unit for the work button.

Schema: tchmi:framework#/definitions/MeasurementUnit

Attribute getter: getBackgroundImageWidthWorkUnit

Attribute setter: setBackgroundImageWidthWorkUnit

Available: since version 12.6.0

4.8.1.17 BackgroundImageWork

Path to an image to be set as a background image for a button to move the axis in the work direction.

Schema: tchmi:framework#/definitions/Path

Attribute getter: getBackgroundImageWork

Attribute setter: setBackgroundImageWork

Available: since version 12.6.0

4.8.1.18 BorderRadius

This attribute sets the border radius of the buttons on the control.

Schema: tchmi:framework#/definitions/BorderRadius

Attribute getter: getBorderRadius

Attribute setter: setBorderRadius

Available: since version 12.6.0

4.8.1.19 BtnBaseText

This will set a text to the button for moving the axis in the base direction.

Schema: tchmi:general#/definitions/String

Attribute getter: getBtnBaseText

Attribute setter: setBtnBaseText

Available: since version 12.6.0

4.8.1.20 BtnWorkText

This will set a text to the button for moving the axis in the work direction.

Schema: tchmi:general#/definitions/String

Attribute getter: getBtnWorkText

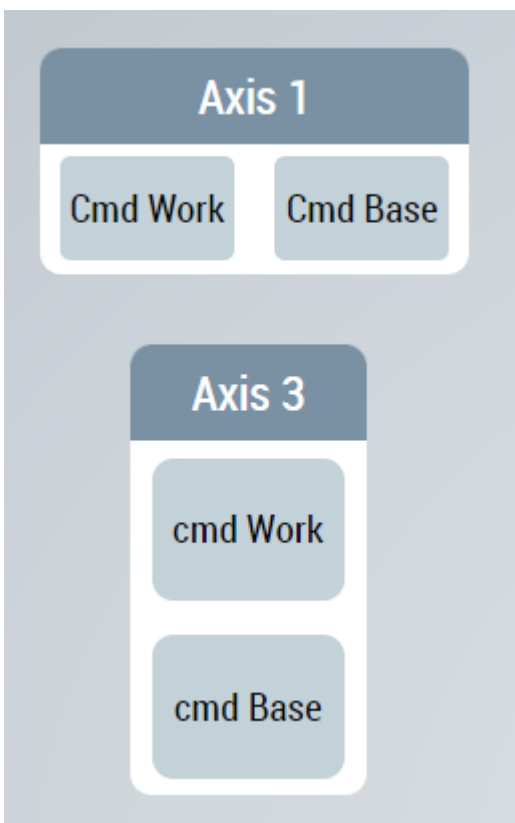
Attribute setter: setBtnWorkText

Available: since version 12.6.0

4.8.1.21 ButtonOrientation

This attribute sets the orientation of the buttons to display them side by side or on top of each other. There are two orientations which can be used:

- Horizontal: Side by side
- Vertical: On top of each other



Schema: tchmi:framework#/definitions/AxisBtnOrientation

Attribute getter: getAxisBtnOrientation

Attribute setter: setAxisBtnOrientation

Available: since version 12.6.0

4.8.1.22 FaultColor

Color of the light ring around the buttons when the status of the configured FB_ManualOperation symbol represent fault in the axis movement.

Schema: tchmi:framework#/definitions/Color

Attribute getter: getFaultColor

Attribute setter: setFaultColor

Available: since version 12.6.0

4.8.1.23 InPosColor

Color of the light ring around the buttons when the status of the configured FB_ManualOperation symbol represent axis has finished the movement into the commanded direction.

Schema: tchmi:framework#/definitions/Color

Attribute getter: getInPosColor

Attribute setter: setInPosColor

Available: since version 12.6.0

4.8.1.24 ManualOperationSymbol

Symbol of FB_ManualOperation to which the control will write the commands and read the status from. The symbol binding at this attribute must have a variable structure.

The top-level structure must have following properties:

Name	Type	Description
fbCmd	FB_CmdManualFunction	Function block or structure that stores command variables from operator.
fbState	FB_StateManualFunction	Function block or structure that stores status variables read by the control.

Each FB_CmdManualFunction function block or structure type must have following properties:

Name	Type	Description
ToBasePos	BOOL	Control will write <code>TRUE</code> to this variable when the operator presses the button to move toward base direction. And will write <code>FALSE</code> as soon as the button is depressed.
ToWorkPos	BOOL	Control will write <code>TRUE</code> value to this variable when the operator presses the button to move toward work direction. And will write <code>FALSE</code> as the button is depressed.

Each FB_StateManualFunction function block or structure type must have following properties:

Name	Type	Description
EnableBasePos	BOOL	If TRUE the ToBasePos button will be enabled, and the operator can press it.
EnableWorkPos	BOOL	If TRUE the ToWorkPos button will be enabled, and the operator can press it.
FaultBasePos	BOOL	If TRUE it stands for fault to move in base direction.
FaultWorkPos	BOOL	If TRUE it stands for fault to move in work direction.
InBasePos	BOOL	If TRUE it stands for the axis has finished moving towards base position.
InWorkPos	BOOL	If TRUE it stands for the axis has finished moving towards work position.
MovingToBasePos	BOOL	If TRUE it stands for the axis is now moving towards the base position.
MovingToWorkPos	BOOL	If TRUE it stands for the axis is now moving towards work position.

PLC1.MAIN.fbManuel	S	PLC1.FB_ManualFunctionHmi	<input checked="" type="checkbox"/>
fbCmd	S	PLC1.FB_CmdManualFunctionHmi	<input checked="" type="checkbox"/>
ToBasePos	S	PLC1.PROPERTY-GET-SET-BOOL	<input checked="" type="checkbox"/>
ToWorkPos	S	PLC1.PROPERTY-GET-SET-BOOL	<input checked="" type="checkbox"/>
fbState	S	PLC1.FB_StateManualFunctionHmi	<input checked="" type="checkbox"/>
EnableBasePos	S	PLC1.PROPERTY-GET-SET-BOOL	<input checked="" type="checkbox"/>
EnableWorkPos	S	PLC1.PROPERTY-GET-SET-BOOL	<input checked="" type="checkbox"/>
FaultBasePos	S	PLC1.PROPERTY-GET-SET-BOOL	<input checked="" type="checkbox"/>
FaultWorkPos	S	PLC1.PROPERTY-GET-SET-BOOL	<input checked="" type="checkbox"/>
InBasePos	S	PLC1.PROPERTY-GET-SET-BOOL	<input checked="" type="checkbox"/>
InWorkPos	S	PLC1.PROPERTY-GET-SET-BOOL	<input checked="" type="checkbox"/>
MovingToBasePos	S	PLC1.PROPERTY-GET-SET-BOOL	<input checked="" type="checkbox"/>
MovingToWorkPos	S	PLC1.PROPERTY-GET-SET-BOOL	<input checked="" type="checkbox"/>

Schema: tchmi:framework#/definitions/Symbol

Attribute getter: getFB_ManualOperation

Attribute setter: setFB_ManualOperation

Available: since version 12.6.0

4.8.1.25 MovingToPosColor

Color of the light ring around the buttons when the status of the configured FB_ManualOperation symbol represent that the axis is moving towards the commanded direction.

Schema: tchmi:framework#/definitions/Color

Attribute getter: getMovingToPosColor

Attribute setter: setMovingToPosColor

Available: since version 12.6.0

4.8.1.26 SingleMode

This attribute allows the control to be configured to work in the single mode. In single mode only the button towards the work direction will be available for the operator. This can be used for example for ON/OFF operations from HMI.

Schema: tchmi:general#/definitions/BOOL

Attribute getter: getSingleMode

Attribute setter: setSingleMode

Available: since version 12.6.0

4.8.1.27 HeaderBackgroundColor

Definition of the background color of the header.

Schema: tchmi:framework#/definitions/Color

Attribute getter: getHeaderBackgroundColor

Attribute setter: setHeaderBackgroundColor

Available: since version 12.8.0

4.8.1.28 HeaderFontColor

Definition of the text color of the header.

Schema: tchmi:framework#/definitions/Color

Attribute getter: getHeaderFontColor

Attribute setter: setHeaderFontColor

Available: since version 12.8.0

4.8.1.29 HeaderHeight

Definition of the height of the header.

Schema: tchmi:framework#/definitions/MeasurementValue

Attribute getter: getHeaderHeight

Attribute setter: setHeaderHeight

Available: since version 12.8.0

4.8.1.30 HeaderHeightUnit

Defines the unit of the height of the header:

- Pixels for an absolute size
- Percent for a relative size

Schema: tchmi:framework#/definitions/MeasurementUnit

Attribute getter: getHeaderHeightUnit

Attribute setter: setHeaderHeightUnit

Available: since version 12.8.0

4.8.1.31 HeaderFontSize

The font size. If the percent is specified as the unit, this is relative to the font size of the parent element.

Schema: tchmi:framework#/definitions/MeasurementValue

Attribute getter: getHeaderFontSize

Attribute setter: setHeaderFontSize

Available: since version 12.8.0

4.8.1.32 HeaderFontSizeUnit

Defines the unit of the height of the header:

- Pixels for an absolute size
- Percent for a relative size

Schema: tchmi:framework#/definitions/MeasurementUnit

Attribute getter: getHeaderHeightUnit

Attribute setter: setHeaderHeightUnit

Available: since version 12.8.0

4.9 MeasurementUnitSelector

NuGet package: Beckhoff.TwinCAT.HMI.Plastic.Controls

Available: since version 12.6.0

With the MeasurementUnitSelector control the user can access the unit configuration configured in the [Unit_Config.json \[▶ 40\]](#) file and change the display unit settings on the client.

GroupName	DisplayUnit	SystemUnit
Acceleration	<input checked="" type="checkbox"/> mm/s ²	mm/s ²
	<input type="checkbox"/> in/s ²	
	<input type="checkbox"/> %	
AngularAcceleration	<input checked="" type="checkbox"/> rad/s ²	rad/s ²
	<input type="checkbox"/> °/s ²	
	<input type="checkbox"/> %	
ElectricCurrent	<input checked="" type="checkbox"/> A	A
	<input type="checkbox"/> mA	
	<input type="checkbox"/> %	
ElectricPotential	<input checked="" type="checkbox"/> V	V
	<input type="checkbox"/> mV	
	<input type="checkbox"/> %	
ElectricResistance	<input checked="" type="checkbox"/> Ω	Ω
	<input type="checkbox"/> %	
	<input type="checkbox"/> %	
Force	<input checked="" type="checkbox"/> N	N
	<input type="checkbox"/> gf	
	<input type="checkbox"/> %	
Frequency	<input checked="" type="checkbox"/> Hz	Hz
	<input type="checkbox"/> %	

- Once the control is dragged from the toolbox and placed in the content, all the Unit Groups present in the configurator will be displayed
- There should be a [Configurator \[▶ 34\]](#) control added to the view to use this control

4.9.1 Attributes

Category: Common

Name	Description
GroupFilter [▶ 96]	It is a filter for all the group names, which will only display group names that are present in this filter.
RowHeight [▶ 96]	Sets the row height for the control.
RowHeightUnit [▶ 96]	Sets the unit of the Row Height [▶ 96] attribute always in Pixel.
TextFontSize [▶ 97]	Sets the font size for the control.
FontSizeUnit [▶ 96]	Sets the font size unit for the control.

4.9.1.1 FontSizeUnit

Sets the font size unit for the control.

Schema: tchmi:framework#/definitions/MeasurementUnit

Attribute getter: getFontSizeUnit

Attribute setter: setFontSizeUnit

Available: since version 12.6.0

4.9.1.2 GroupFilter

This attribute allows filtering the groups listed on the control.

Once active only the groups that are listed in this list will be displayed. If the filter is set to empty or if no filter is added, all the unit groups configured in [unit_config.json](#) [[▶ 40](#)] file are available on the control.

Schema: tchmi:framework#/definitions/GroupFilterList

Attribute getter: getGroupFilter

Attribute setter: setGroupFilter

Available: since version 12.6.0

4.9.1.3 RowHeight

Sets the row height for the control.

Schema: tchmi:framework#/definitions/MeasurementValue

Attribute getter: getRowHeight

Attribute setter: setRowHeight

Available: since version 12.6.0

4.9.1.4 RowHeightUnit

Sets the unit for the [Row Height](#) [[▶ 96](#)] attribute always in Pixel.

Schema: tchmi:framework#/definitions/PixelUnit

Attribute getter: getRowHeightUnit

Attribute setter: setRowHeightUnit

Available: since version 12.6.0

4.9.1.5 TextFontSize

Sets the font size for the control.

Schema: tchmi:framework#/definitions/MeasurementValue

Attribute getter: getFontSize

Attribute setter: setFontSize

Available: since version 12.6.0

4.9.1.6 HeaderBackgroundColor

Definition of the background color of the header.

Schema: tchmi:framework#/definitions/Color

Attribute getter: getHeaderBackgroundColor

Attribute setter: setHeaderBackgroundColor

Available: since version 12.8.0

4.9.1.7 HeaderFontColor

Definition of the text color of the header.

Schema: tchmi:framework#/definitions/Color

Attribute getter: getHeaderFontColor

Attribute setter: setHeaderFontColor

Available: since version 12.8.0

4.9.1.8 HeaderHeight

Definition of the height of the header.

Schema: tchmi:framework#/definitions/MeasurementValue

Attribute getter: getHeaderHeight

Attribute setter: setHeaderHeight

Available: since version 12.8.0

4.9.1.9 HeaderHeightUnit

Defines the unit of the height of the header:

- Pixels for an absolute size
- Percent for a relative size

Schema: tchmi:framework#/definitions/MeasurementUnit

Attribute getter: getHeaderHeightUnit

Attribute setter: setHeaderHeightUnit

Available: since version 12.8.0

4.9.1.10 HeaderFontSize

The font size. If the percent is specified as the unit, this is relative to the font size of the parent element.

Schema: tchmi:framework#/definitions/MeasurementValue

Attribute getter: getHeaderFontSize

Attribute setter: setHeaderFontSize

Available: since version 12.8.0

4.9.1.11 HeaderFontSizeUnit

Defines the unit of the height of the header:

- Pixels for an absolute size
- Percent for a relative size


Schema: tchmi:framework#/definitions/MeasurementUnit

Attribute getter: getHeaderHeightUnit

Attribute setter: setHeaderHeightUnit

Available: since version 12.8.0

4.9.2 User Interactions

Name	Description	Result on the control
Single Tap	With a tap the user can select a unit inside a unit group that acts as display unit for that group.	 <p>The screenshot shows a table with three columns: GroupName, DisplayUnit, and SystemUnit. The 'Acceleration' group has a dropdown menu open, showing options for 'mm/s²', 'in/s²', and '%'. The 'AngularAcceleration' group has a dropdown menu showing 'rad/s²', '°/s²', and '%'. The 'ElectricCurrent' group has a dropdown menu showing 'A', 'mA', and '%'. A blue checkmark is visible in the 'DisplayUnit' column for each group, indicating the selected unit.</p>

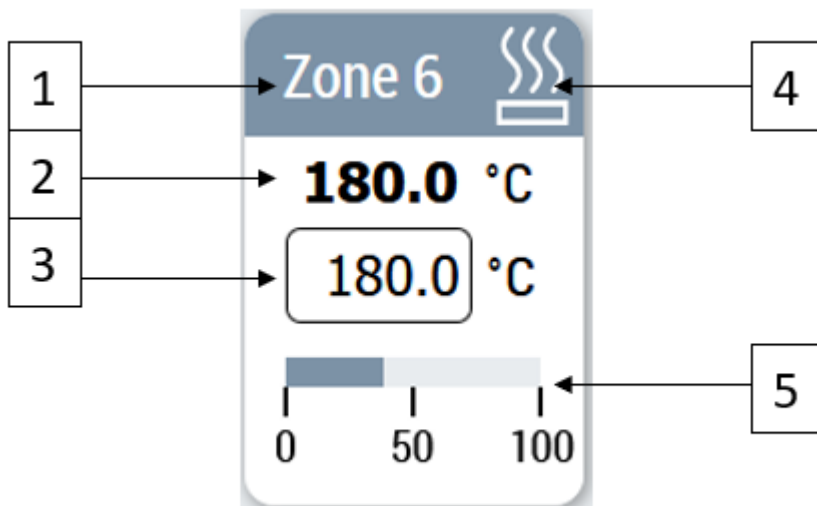
Available: since version 12.6.0

4.10 PfwSingleTempControl

NuGet package: Beckhoff.TwinCAT.HMI.Plastic.Controls

Available: since version 12.6.0

PfwSingleTempControl allows user to easily display and control a temperature zone in TwinCAT HMI.



- 1. Zone name display
- 2. Actual temperature value of the zone
- 3. Set temperature value of the zone
- 4. Status icon that can show Cooling, Heating or Error state
- 5. Power level of the zone

4.10.1 Attributes

Category: Colors

Name	Description
TopBarBackgroundColor [▶ 100]	Background color of the top bar.

Category: Common

Name	Description
fbTempZone [▶ 99]	Symbol of FB_TempZone.
Minimode [▶ 100]	If set to TRUE, the control will be displayed in mini mode.
ZoneName-FontSize [▶ 100]	Font size of the zone name displayed in the top bar of the control.
ZoneName-FontSizeUnit [▶ 101]	Unit of the font size for the zone name text.
ZoneName [▶ 100]	Text to be displayed as the zone name.
DigitsAfterDecimal [▶ 102]	Number of digits displayed after decimal points for float values.

4.10.1.1 fbTempZone

Symbol of the type FB_TempZone should be linked to this attribute. The control will resolve the individual variable paths inside the control and fetch necessary data.

The structure of FB_TempZone:

Name	Type	Description
ActualTemperature	Number	Actual temperature value.
SetpointTemperature	Number	Setpoint value of the zone.
Error	Bool	If zone is in error.
Heating	Bool	If zone is heating.
Cooling	Bool	If zone is cooling.
PowerLevel	Number	Output power level of the zone.

Schema: tchmi:framework#/definitions/Symbol

Attribute setter: setfbTempZone

Attribute getter: getfbTempZone

Available: since version 12.6.0

4.10.1.2 MiniMode

With Mini Mode attribute the user can define if the control should be displayed in minimized mode that is more suitable for a smaller screen.

In mini mode the power level progress bar will not be displayed.

Schema: tchmi:general#/definitions/Bool

Attribute setter: setMiniMode

Attribute getter: getMiniMode

Available: since version 12.6.0

4.10.1.3 TopBarBackgroundColor

Background color of the top bar area of the control.

Schema: tchmi:framework#/definitions/Color

Attribute setter: setTopBar_BackgroundColour

Attribute getter: getTopBar_BackgroundColour

Available: since version 12.6.0

4.10.1.4 ZoneName

Text to be displayed as the zone name on the control.

Schema: tchmi:general#/definitions/string

Attribute setter: setZoneName

Attribute getter: getZoneName

Available: since version 12.6.0

4.10.1.5 ZoneName-FontSize

Font size of the zone name text that is displayed on the control.

Schema: tchmi:framework#/definitions/MeasurementValue

Attribute setter: setZoneNameFontSize

Attribute getter: getZoneNameFontSize

Available: since version 12.6.0

4.10.1.6 ZoneName-FontSizeUnit

Unit of the font size set through `ZoneName-FontSize [1_100]` attribute. It can be 'px' or '%'.

Schema: tchmi:framework#/definitions/MeasurementUnit

Attribute setter: setZoneNameFontSizeUnit

Attribute getter: getZoneNameFontSizeUnit

Available: since version 12.6.0

4.10.1.7 HeaderBackgroundColor

Definition of the background color of the header.

Schema: tchmi:framework#/definitions/Color

Attribute getter: getHeaderBackgroundColor

Attribute setter: setHeaderBackgroundColor

Available: since version 12.8.0

4.10.1.8 HeaderFontColor

Definition of the text color of the header.

Schema: tchmi:framework#/definitions/Color

Attribute getter: getHeaderFontColor

Attribute setter: setHeaderFontColor

Available: since version 12.8.0

4.10.1.9 HeaderHeight

Definition of the height of the header.

Schema: tchmi:framework#/definitions/MeasurementValue

Attribute getter: getHeaderHeight

Attribute setter: setHeaderHeight

Available: since version 12.8.0

4.10.1.10 HeaderHeightUnit

Defines the unit of the height of the header:

- Pixels for an absolute size
- Percent for a relative size

Schema: tchmi:framework#/definitions/MeasurementUnit

Attribute getter: getHeaderHeightUnit

Attribute setter: setHeaderHeightUnit

Available: since version 12.8.0

4.10.1.11 HeaderFontSize

The font size. If the percent is specified as the unit, this is relative to the font size of the parent element.

Schema: tchmi:framework#/definitions/MeasurementValue

Attribute getter: getHeaderFontSize

Attribute setter: setHeaderFontSize

Available: since version 12.8.0

4.10.1.12 HeaderFontSizeUnit

Defines the unit of the height of the header:

- Pixels for an absolute size
- Percent for a relative size

Schema: tchmi:framework#/definitions/MeasurementUnit

Attribute getter: getHeaderHeightUnit

Attribute setter: setHeaderHeightUnit

Available: since version 12.8.0

4.10.1.13 DigitsAfterDecimal

The resolution of individual number values displayed on the control can be updated with this attribute.

Schema: tchmi:general#/definitions/Number

Attribute getter: getDigitsAfterDecimal

Attribute setter: setDigitsAfterDecimal

Available: since version 12.8.0

4.11 PfwTempParameters

NuGet package: Beckhoff.TwinCAT.HMI.Plastic.Temperature

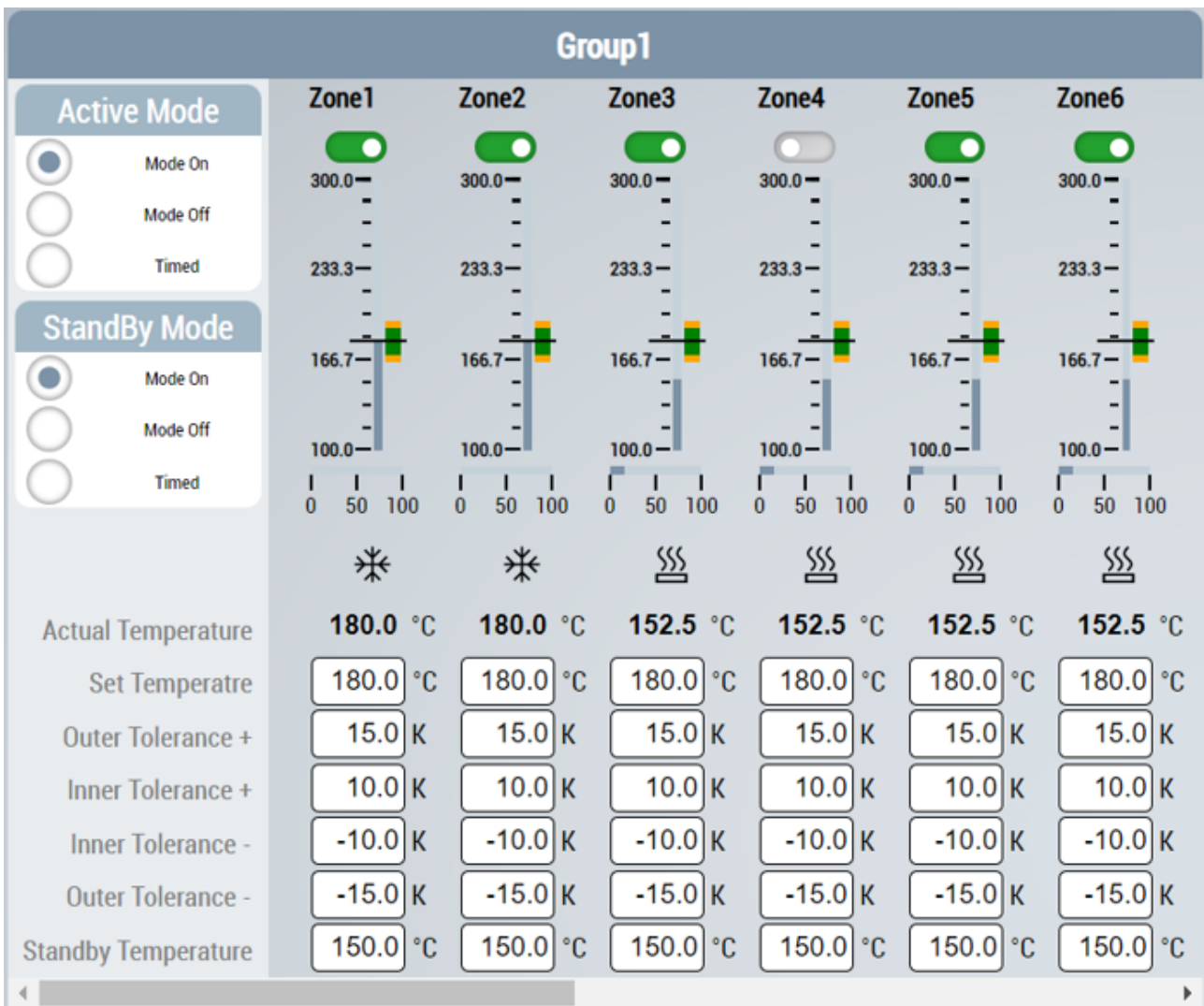
Available: in version 12.6.0

PfwTempParameters control is used to visualize and allow users to change different operational parameters of temperature groups and zones configured in the PLC.

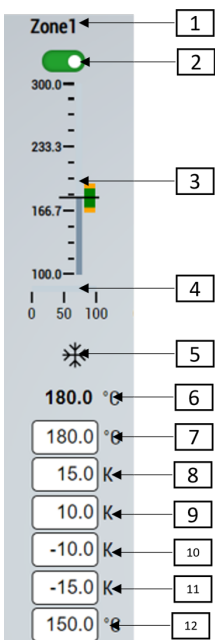
NOTICE

Deprecated: This control is no longer part of Beckhoff.TwinCAT.HMI.Plastic.Temperature package from 12.8.0 version onwards.

Please use [ZoneConfiguration \[▶_208\]](#) control instead.



Every temperature zone in the selected temperature zone group is visualized using different input and graphical controls.



1. Temperature zones ZoneName property
2. User can enable/disable a zone with this control.

3. A linear gauge will visualize the actual temperature, the set temperature, and the inner and outer tolerance bands on a scale.
4. A gauge will display the power level property of the zone.
5. Temperature zones cooling/heating status will be displayed with an appropriate icon.
6. Actual temperature of the zone
7. User can enter the set temperature for the given zone.
8. Outer tolerance in positive direction
9. Inner tolerance in positive direction
10. Inner tolerance in negative direction
11. Outer tolerance in negative direction
12. Standby temperature of the zone

4.11.1 Attributes

Category: Config

Name	Description
ActualZoneGroupId [▶ 104]	Index of the zone group displayed on the control.
DisplayOperationModes [▶ 104]	With this attribute the operation controls can be added or removed from the control.
DigitsAfterDecimal [▶ 104]	Individual values display resolution.

4.11.1.1 ActualZoneGroupId

With this attribute the user can change the index of the configured temperature zone group index that is displayed on the control. By changing this value the user can visualize and edit all possible temperature zone groups in the project.

Schema: tchmi:general#/definitions/Number

Attribute getter: getActualZoneGroupId

Attribute setter: setActualZoneGroupId

Available: since version 12.6.0

4.11.1.2 DigitsAfterDecimal

The resolution of individual number values displayed on the control can be updated with this attribute.

Schema: tchmi:general#/definitions/Number

Attribute getter: getDigitsAfterDecimal

Attribute setter: setDigitsAfterDecimal

Available: since version 12.6.0

4.11.1.3 DisplayOperationModes

By default, the operation controls that set the active mode and standby mode of the selected temperature zone group are not displayed on the control. With this attribute the controls can be made available for the user.

Schema: tchmi:general#/definitions/Boolean

Attribute getter: getDisplayOperationModes

Attribute setter: setDisplayOperationModes

Available: since version 12.6.0

4.12 ProcessScheduler

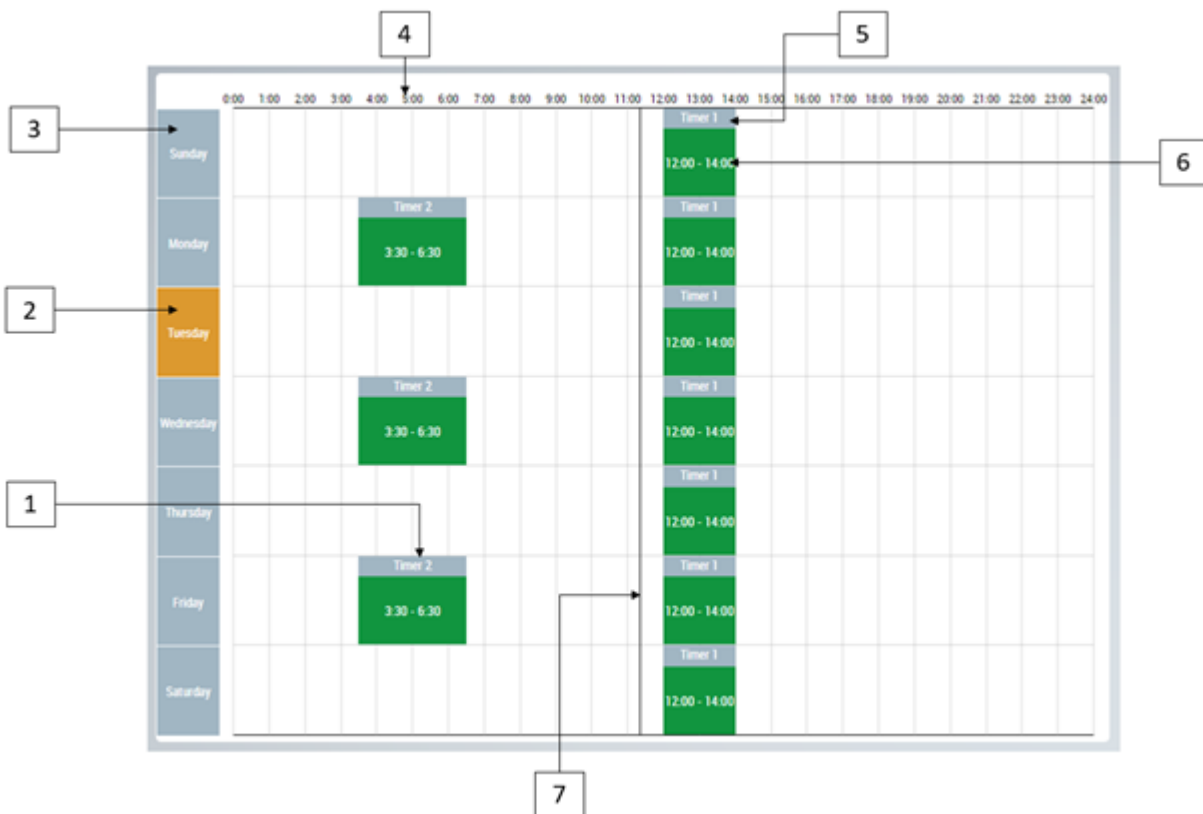
NuGet package: Beckhoff.TwinCAT.HMI.Plastic.Controls

Available: since version 12.6.0

The ProcessScheduler control will allow user to visualize and edit weekly schedules that can be used by the PLC to execute any process.

● Activate action

i Any new changes on the control must be activated using [ActivateConfig \[▶ 107\]](#) action to set new values to the PLC.



1. Single schedule object
2. Current day of the week
3. Weekdays
4. Hours
5. Title of the schedule
6. Time duration of the schedule
7. Current time indicator

4.12.1 Attributes

Category: Scheduler

Name	Description
TimerList [► 106]	Symbol of array with FB_TimerWeekdayHmi type.
TimerLabels [► 106]	Array of labels for each schedule object.
SelectedTimerIndex [► 106]	Index of selected schedule from the TimerList [► 106] array.

Category: Common

Name	Description
ActivationRequired [► 106]	Is set to <code>TRUE</code> if the user makes new changes that are not activated.

4.12.1.1 ActivationRequired

This read only attribute is set to `TRUE`, if a user makes new changes that are not activated.

This value can be used by other controls such as a `TcHmiButton` to change its appearance and inform the user that new changes must be activated.

Schema: tchmi:general#/definitions/Boolean

Attribute getter: getActivationRequired

Available: since version 12.6.0

4.12.1.2 SelectedTimerIndex

This attribute will reflect the array index of the schedule that the user has selected on the screen. By selecting a schedule object the user can edit or remove that schedule from the array.

Schema: tchmi:general#/definitions/Number

Attribute getter: getSelectedTimerIndex

Attribute setter: setSelectedTimerIndex

Available: since version 12.6.0

4.12.1.3 TimerLabels

Set of `TimerLabels` which is an array of labels. User can give a simple text or use localization symbols to set the title for the schedules.

Schema: tchmi:framework#/definitions/TimerLabelList

Attribute getter: getTimerLabels

Attribute setter: setTimerLabels

Available: since version 12.6.0

4.12.1.4 TimerList

This attribute should be bind to an array symbol on the PLC where each element stores the configuration of one schedule.

Properties of each element in this array:

Name	Type	Description
StartTime	LREAL	Start time in seconds.
EndTimer	LREAL	End time in seconds.
Weekday	BYTE	A BYTE that is used to select the weekdays for the schedule. The 1 st bit (LSB) is set to show that the schedule is used on Monday. Similarly the 7 th bit is set to show that the schedule is used on Sunday. For example a schedule that is used on Monday, Tuesday and Sunday will have the value that is equal to '2#0100 0011'.

Schema: tchmi:general#/definitions/Symbol

Attribute getter: getTimerList

Attribute setter: setTimerList

Available: since version 12.6.0

4.12.2 Functions

Category: Actions

Name	Description
ActivateConfig [▶ 107]	This event is generated when a new configuration is activated on the control.
ResetConfigdata [▶ 107]	This function will revert all user changes and reload the last activated configuration from the PLC.
AddScheduleData [▶ 108]	This function with access right will show a pop up on the control that allows creating a new schedule.
EditScheduleData [▶ 108]	This function with access right will show a pop up on the control that allows editing the selected schedule.

4.12.2.1 ActivateConfig

By calling this function all user changes will be activated and written to PLC symbols.

Parameter

Name	Type	Description
-	-	-

Return value

Type	Description
void	No return value.

Available: since version 12.6.0

4.12.2.2 ResetConfigdata

This function will revert any user changes by reloading the last activated configuration.

Parameter

Name	Type	Description
-	-	-

Return value

Type	Description
void	No return value.

Available: since version 12.6.0

4.12.2.3 AddScheduleData

This function will show a pop up on the control that allows creating a new schedule.

User needs the **Operate** access right to call this function.

Parameter

Name	Type	Description
-	-	-

Return value

Type	Description
void	No return value.

Available: since version 12.6.0

4.12.2.4 EditScheduleData

This function will show a pop up on the control that allows editing the selected schedule. The user must first select an existing schedule on the control and then call this function.

User needs the **Operate** access to execute this function.

Parameter

Name	Type	Description
-	-	-

Return value

Type	Description
void	No return value.

Available: since version 12.6.0

4.12.3 Events**Category: Actions**

Name	Description
onUserConfigActivated [►_109]	This event is generated when a new configuration is activated on the control.
onEditingCanceled [►_109]	This event is generated when user changes are canceled and the old configuration is reloaded on the control.

4.12.3.1 onUserConfigActivated

This event is generated when a new configuration is activated on the control.

Available: since version 12.6.0

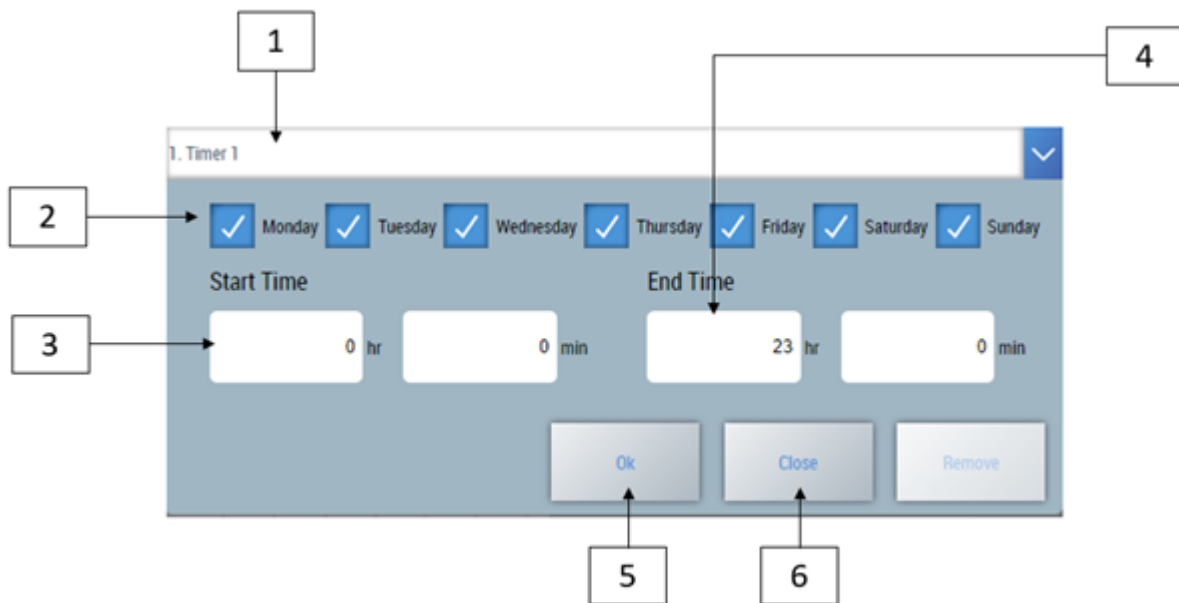
4.12.3.2 onEditingCanceled

This event is generated when user changes are canceled and the old configuration is reloaded on the control.

Available: since version 12.6.0

4.12.4 Dialogs

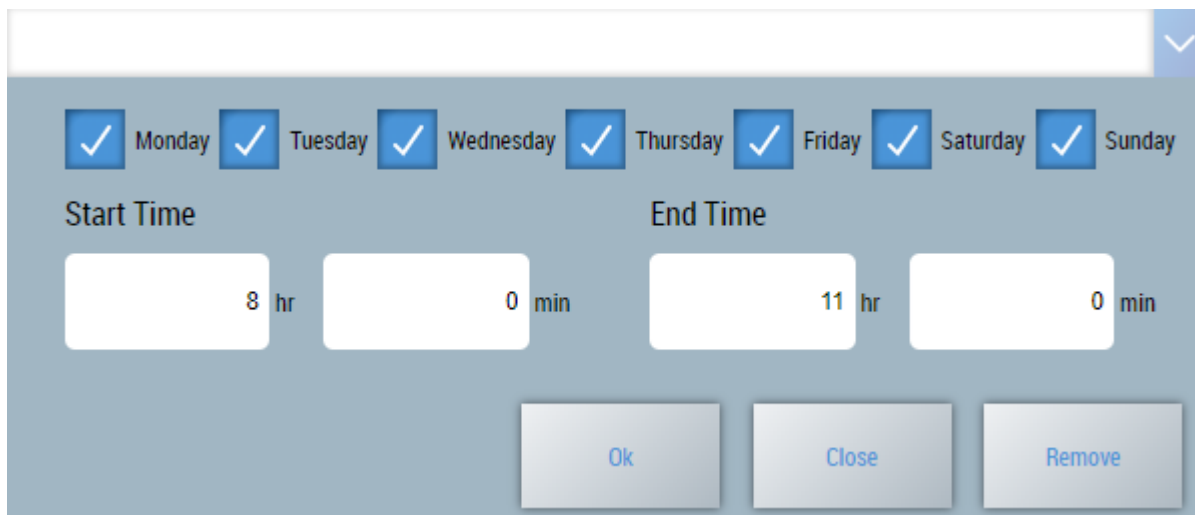
4.12.4.1 Add new schedule



1. Select the array index of the schedule to be added.
2. According to the checkboxes, the schedule will be configured with the selected days.
3. Select the start time of the schedule.
4. Select the end time of the schedule.
5. Add a new schedule and close the dialog.
6. Cancel changes and close the dialog.
7. New configuration must be activated by executing [ActivateConfig \[▶_107\]](#) function.

Available: since version 12.6.0

4.12.4.2 Edit or remove a schedule



1. By tapping or clicking on it select a schedule to be edited..
2. Call the [EditScheduleData](#) [▶ 108] function, to see the editing dialog
3. Change parameters or click on the remove button to remove a schedule from the list.
4. New configuration must be activated by executing the [ActivateConfig](#) [▶ 107] function.

Available: since version 12.6.0

4.13 StateIndicator

NuGet package: Beckhoff.TwinCAT.HMI.Plastic.Controls

Available: since version 12.6.0



StateIndicator control is showing different states and can be used as an LED indicator in TwinCAT HMI projects.

Its fill color will toggle between three possible colors based on the current state.

The state can be one of three possible states “Error state”, “Warning state” or ”Normal state” and can be changed using attributes.

4.13.1 Attributes

Category: Indicator State

Name	Description
IndicatorState [▶ 111]	Indicator status symbol of BYTE data type. The three least significant bits are used to calculate the state.
NormalState [▶ 112]	TRUE value to this attribute will set the state of the control to “Normal state”.
WarningState [▶ 112]	TRUE value to this attribute will set the state of the control to “Warning state”.
ErrorState [▶ 111]	TRUE value to this attribute will set the state of the control to “Error state”.

Category: Colors

Name	Description
NormalStateColor [▶ 112]	Sets the “Normal state” color for the control. The default color is Green.
WarningStateColor [▶ 112]	Sets the “Warning state” color for the control. The default color is Orange.
ErrorStateColor [▶ 111]	Sets the “Error state” color for the control. The default color is Red.

4.13.1.1 ErrorState

TRUE value to this attribute will set state of the control to “Error state”, and the control will be filled with the “Error state” color.

Schema: tchmi:general#/definitions/Boolean

Attribute getter: getErrorState

Attribute setter: setErrorState

Available: since version 12.6.0

4.13.1.2 ErrorStateColor

Sets the “Error state” color for the control. The default color is Red.

Schema: tchmi:framework#/definitions/Color

Attribute getter: getErrorStateColor

Attribute setter: setErrorStateColor

Available: since version 12.6.0

4.13.1.3 IndicatorState

Indicator status symbol of BYTE data type. The three least significant bits are used to calculate the state. The table below explains bit positions and their meaning for the control.

Bit position	Name	Meaning
0	Normal state bit	If this bit is set to TRUE, the control will have “Normal state” as its active state.
1	Warning state bit	If this bit is set to TRUE, the control will have “Warning state” as its active state. If set to TRUE, this bit overrides the state set by bit 0.
2	Error state bit	If this bit is set to TRUE, the control will have “Error state” as its active state. If set to TRUE, this bit overrides the states set by bit 1 and bit 0.
3..7	Not used	These bits carry no information for the state of the control.

Schema: tchmi:general#/definitions/BYTE

Attribute getter: getIndicatorState

Attribute setter: setIndicatorState

Available: since version 12.6.0

4.13.1.4 NormalState

TRUE value to this attribute will set the state of the control to “Normal state”, and the control will be filled with the “Normal state” color.

Schema: tchmi:general#/definitions/Boolean

Attribute getter: getNormalState

Attribute setter: setNormalState

Available: since version 12.6.0

4.13.1.5 NormalStateColor

Sets the “Normal state” color for the control. The default color is Green.

Schema: tchmi:framework#/definitions/Color

Attribute getter: getNormalStateColor

Attribute setter: setNormalStateColor

Available: since version 12.6.0

4.13.1.6 WarningState

TRUE value to this attribute will set the state of the control to “Warning state”, and the control will be filled with the “Warning state” color.

Schema: tchmi:general#/definitions/Boolean

Attribute getter: getWarningState

Attribute setter: setWarningState

Available: since version 12.6.0

4.13.1.7 WarningStateColor

Sets the “Warning state” color for the control. The default color is Orange.

Schema: tchmi:framework#/definitions/Color

Attribute getter: getWarningStateColor

Attribute setter: setWarningStateColor

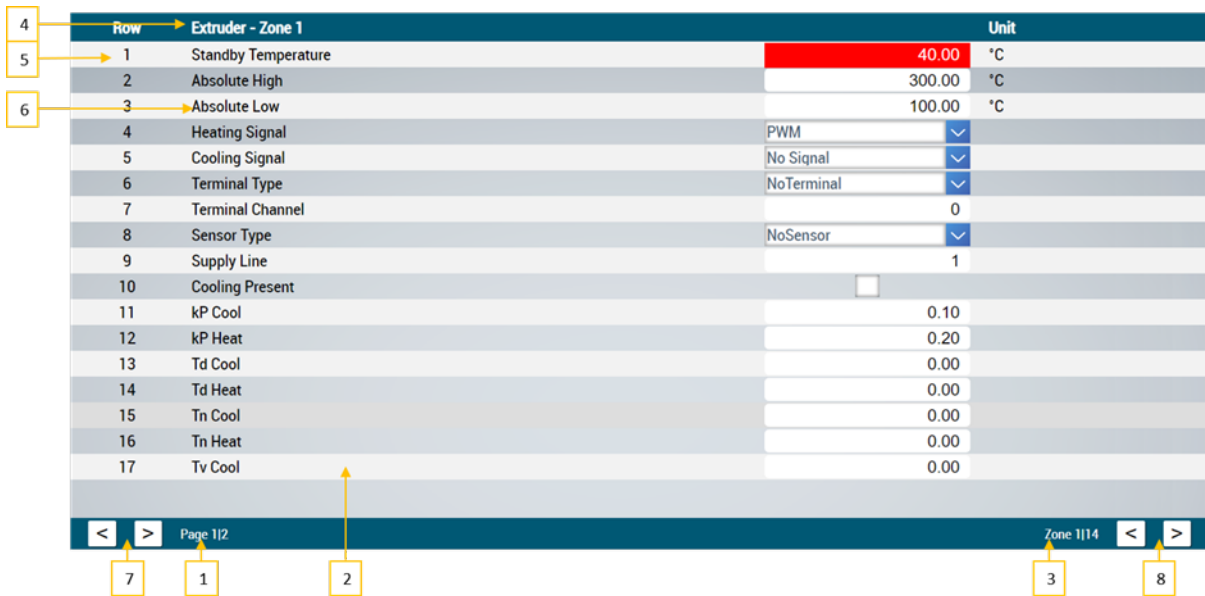
Available: since version 12.6.0

4.14 Table

NuGet package: Beckhoff.TwinCAT.HMI.Plastic.Controls

Available: since version 12.6.0

Table control can be used to visualize multiple PLC variables in paginated table format. Details about the configuration can be found in the configuration chapter.



1. Text displayed at the subpage selector
2. Number of rows to be displayed on each sub page
3. Text displayed at the table selector
4. Table title
5. Row number
6. Row description
7. Buttons to navigate between subpages of a table
8. Buttons to navigate between different tables

4.14.1 Attributes

Category: Table Config

Name	Description
TableRowCount [► 117]	Number of rows.
SubTableText [► 117]	Text to show for the table selector.
FirstTable [► 114]	The first table configuration.
CreateMultipleTables [► 114]	Allow to create multiple tables in the control.
Tables [► 117]	List of all tables.
RowHeight [► 117]	Sets the row height.
RowHeightUnit [► 117]	Sets the Row HeightUnit always in Pixel.

Category: Selection

Name	Description
ActiveSubPageIndex [► 113]	Index of displayed subpage.
ActiveTableIndex [► 114]	Index of displayed table.

4.14.1.1 ActiveSubPageIndex

Sets the index of the active subpage that is displayed on the control.

If the index is set out of range, it will be reset to default value (0).

Schema: tchmi:general##/definitions/INT

Attribute getter: getActSubPage

Attribute setter: setActSubPage

Available: since version 12.6.0

4.14.1.2 **ActiveTableIndex**

Sets the index of the sub-table that is displayed on the control.

If the index is set out of range, it will be reset to default value (0).

Schema: tchmi:general#/definitions/INT

Attribute getter: getActSubTable

Attribute setter: setActSubTable

Available: since version 12.6.0

4.14.1.3 **CreateMutipleTables**

If set to `TRUE`, multiple tables as per [Tables \[▶_117\]](#) attribute are created.

Schema: tchmi:general#/definitions/BOOL

Attribute getter: getCreateMultipleTables

Attribute setter: setCreateMultipleTables

Available: since version 12.6.0

4.14.1.4 **FirstTable**

The first table row elements are configured with attribute.

TplConfigTable_2 | First Table

FBValueDataType	RowNumber	LocalizationKey
FloatDataType	1	L Row_Locale_1
BoolDataType	2	L Row_Locale_2
IntDataType	3	L Row_Locale_3
noneDataType	4	L Row_Locale_4
IntDataType_E	5	L Row_Locale_5
FloatDataType	6	L Row_Locale_6
BoolDataType	7	L Row_Locale_7
IntDataType	8	L Row_Locale_8
IntDataType	9	L Row_Locale_9
noneDataType	10	L Row_Locale_10
noneDataType	11	L Row_Locale_11
FloatDataType	12	L Row_Locale_12
FloatDataType	13	L Row_Locale_12
FloatDataType	14	L Row_Locale_13
FloatDataType	15	L Row_Locale_13
IntDataType_E	16	L Row_Locale_5

Properties

- Access Rights
 - Access Right (no items)
- ControlConfig
 - FBValue_DataType FloatDataType
 - UnitGroup fx Length
 - MaxValue S PLC1.MAIN.fbmain
 - MinValue S PLC1.MAIN.fbmain
 - RefValue S PLC1.MAIN.fbmain
 - Value S PLC1.MAIN.fbmain
 - ComboBox SrcD... (4 items)
 - isReadOnly
 - Recipe Type None
- General
 - isDescriptor
 - RowNumber 1
 - LocalizationKey L Row_Locale_1

TableRow Add

OK Cancel

First Table array element properties:

Property	Type	Description
Access Right	tchmi:framework#/definitions/AccessConfig	Access rights to every row element can be listed, it allows individual access configuration for each row.
FBValue_DataType	tchmi:framework#/definitions/RowDataType	<ul style="list-style-type: none"> Select the data type from the enumeration for formatting and displaying the value, the user can select <code>noneDataType</code>, <code>BoolDataType</code>, <code>IntDataType</code>, <code>IntDataType_Enum</code>, <code>FloatDataType</code> or <code>TimeDataType</code> as data types. Based on this selection the row will create a control to display the value and allow user input. For <code>BoolDataType</code> a Checkbox will appear as user interaction control. If <code>IntDataType_Enum</code> is set, a combo box control will appear. Which will be configured by the <code>ComboBoxSrcData</code> property. No input control will be shown if <code>noneDataType</code> data type is selected. For the rest options an <code>InputBox</code> with appropriate configurations will appear.
UnitGroup	string	Sets the unit group for the input box control for this row.
MaxValue	tchmi:framework#/definitions/ValueAttribute	Sets the maximum value the user can enter.
MinValue	tchmi:framework#/definitions/ValueAttribute	Sets the minimum value the user can enter.
RefValue	tchmi:framework#/definitions/ValueAttribute	Sets the reference value which is needed to calculate percentage if the unit group is percentage.
Value	tchmi:framework#/definitions/ValueAttribute	Value symbol path of the value that will be displayed and updated on a user input.
ComboBox SrcData	tchmi:framework#/definitions/TcHmi.Controls.Beckhoff.TcHmiCombobox.ListItemList	With this property you can configure a combo box.
isReadOnly	boolean	Set it to <code>TRUE</code> if the value shown on the row should not be editable by the user.
Recipe Type	tchmi:framework#/definitions/RecipeType	This is the recipe type attribute for the input box control.
isDescriptor	boolean	Set it to <code>TRUE</code> to use this row only as a descriptor without any input option.
RowNumber	number	Display row number of the row.
LocalizationKey	string	This is the text that will be shown as a description of each row.

Schema: tchmi:framework#/definitions/TableRowsList

Attribute getter: getReftable

Attribute setter: setReftable

Available: since version 12.6.0

4.14.1.5 RowHeight

Sets the height for each row in the control.

Schema: tchmi:framework#/definitions/MeasurementValue

Attribute getter: getRowHeight

Attribute setter: setRowHeight

Available: since version 12.6.0

4.14.1.6 RowHeightUnit

Sets the unit of the row height always in pixel.

Schema: tchmi:framework#/definitions/PixelUnit

Attribute getter: getRowHeightUnit

Attribute setter: setRowHeightUnit

Available: since version 12.6.0

4.14.1.7 SubTableText

The text that is displayed at the table selector.

Schema: tchmi:general#/definitions/String

Attribute getter: getSubTableText

Attribute setter: setSubTableText

Available: since version 12.6.0

4.14.1.8 TableRowCount

Number of rows displayed on one page. Total rows in each table will be divided by this attribute to create subpages.

Schema: tchmi:general#/definitions/Number

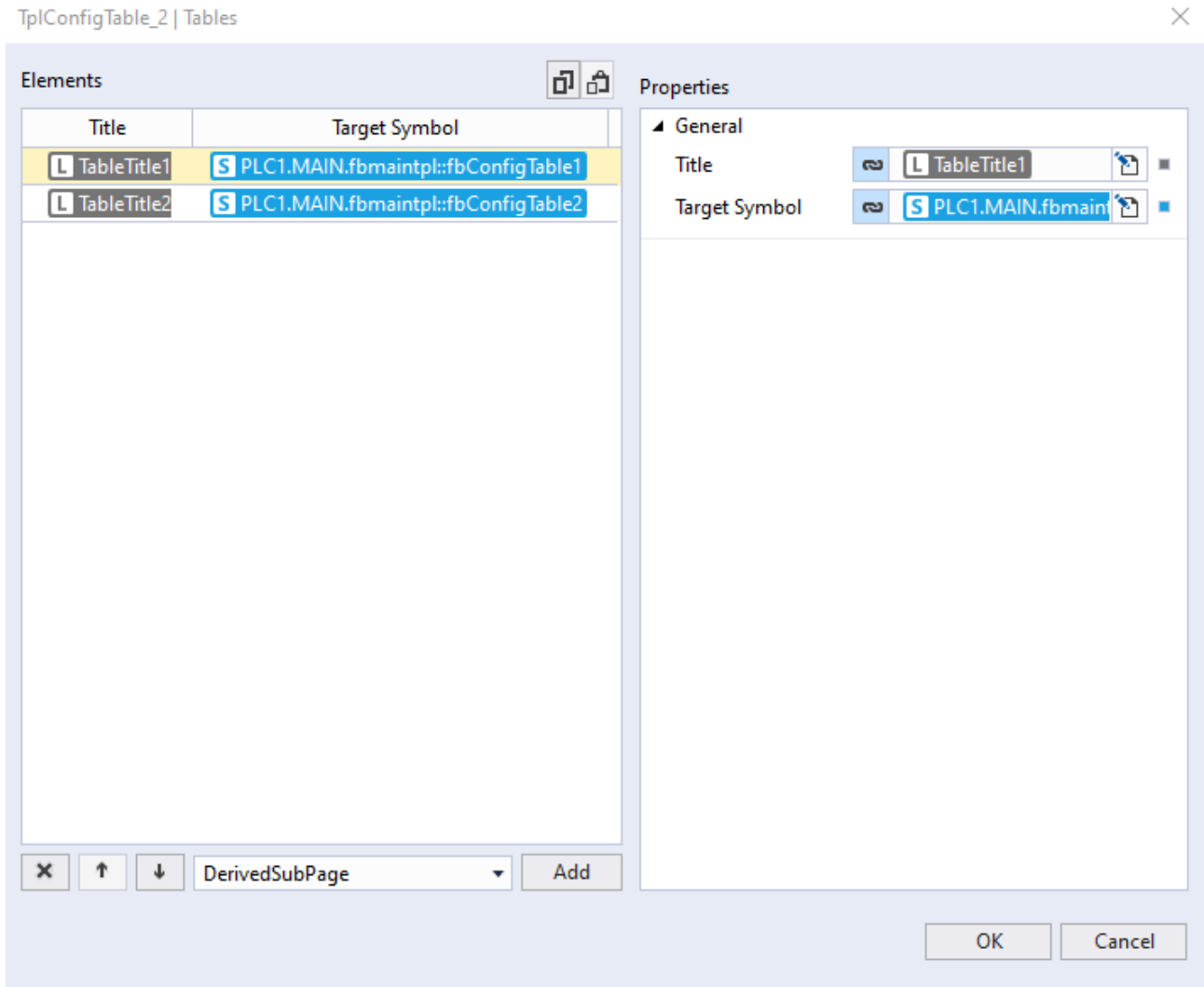
Attribute getter: getTableRowCount

Attribute setter: setTableRowCount

Available: since version 12.6.0

4.14.1.9 Tables

After configuring the first table, it is possible to add more similar tables to the control.



Properties of the table elements:

Name	Description
Table Title	This is a string that is shown on the top line of the table.
Target Symbol	This is a top-level symbol that covers all sub symbols in the row.

There must be at least one element in the table list, and the first element must have the target symbol of the FB whose direct children are configured in all first table rows.

Please check out [step by step guide \[▶ 120\]](#) to get more information about configuring tables attribute to show multiple sub tables on the control.

Schema: tchmi:framework#/definitions/DerivedSubPageList

Attribute getter: getTableList

Attribute setter: setTableList

Available: since version 12.6.0

4.14.1.10 HeaderBackgroundColor

Definition of the background color of the header.

Schema: tchmi:framework#/definitions/Color

Attribute getter: getHeaderBackgroundColor

Attribute setter: setHeaderBackgroundColor

Available: since version 12.8.0

4.14.1.11 HeaderFontColor

Definition of the text color of the header.

Schema: tchmi:framework#/definitions/Color

Attribute getter: getHeaderFontColor

Attribute setter: setHeaderFontColor

Available: since version 12.8.0

4.14.1.12 HeaderHeight

Definition of the height of the header.

Schema: tchmi:framework#/definitions/MeasurementValue

Attribute getter: getHeaderHeight

Attribute setter: setHeaderHeight

Available: since version 12.8.0

4.14.1.13 HeaderHeightUnit

Defines the unit of the height of the header:

- Pixels for an absolute size
- Percent for a relative size

Schema: tchmi:framework#/definitions/MeasurementUnit

Attribute getter: getHeaderHeightUnit

Attribute setter: setHeaderHeightUnit

Available: since version 12.8.0

4.14.1.14 HeaderFontSize

The font size. If the percent is specified as the unit, this is relative to the font size of the parent element.

Schema: tchmi:framework#/definitions/MeasurementValue

Attribute getter: getHeaderFontSize

Attribute setter: setHeaderFontSize

Available: since version 12.8.0

4.14.1.15 HeaderFontSizeUnit

Defines the unit of the height of the header:

- Pixels for an absolute size
- Percent for a relative size

Schema: tchmi:framework#/definitions/MeasurementUnit

Attribute getter: getHeaderHeightUnit

Attribute setter: setHeaderHeightUnit

Available: since version 12.8.0

4.14.2 Events

Name	Description
onTableDataChanged [▶ 120]	This event is fired when a user changes a value by editing the input control of any row.

4.14.2.1 onTableDataChanged

This event is raised, when a user changes a value by editing the input control of any row.

Available: since version 12.6.0

4.14.3 Step by step

4.14.3.1 Configure table control to show single table

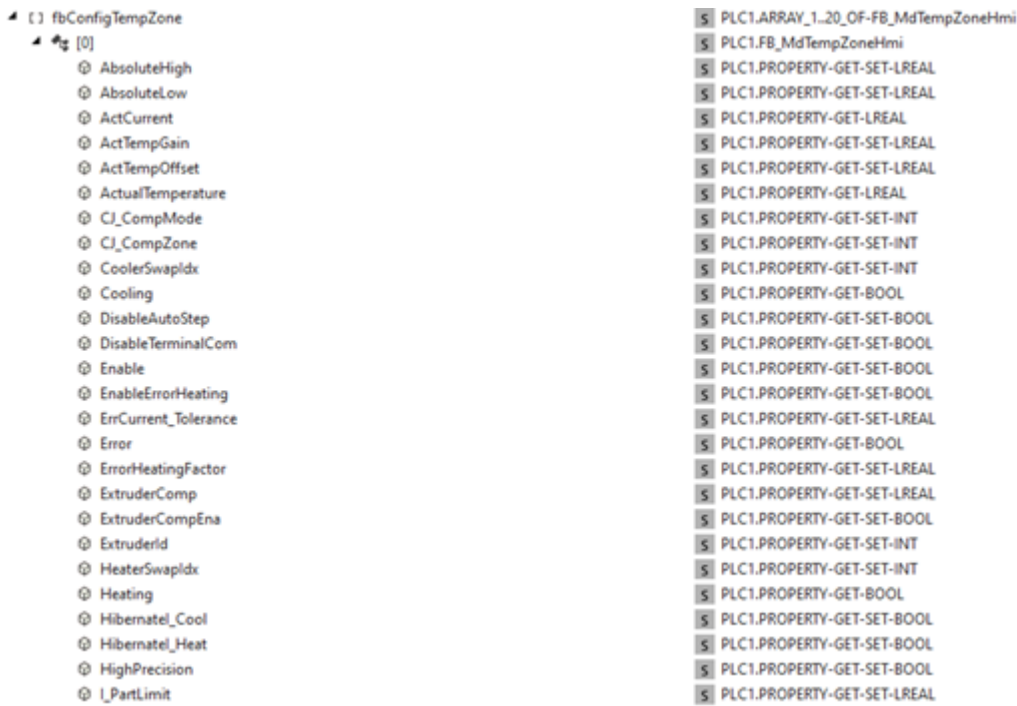
1. Drag and drop a table control from the toolbox to the content.
2. Set `FALSE` value to [CreateMutipleTables](#) [[▶ 114](#)] attribute.
3. Edit the [First Table](#) [[▶ 114](#)] attribute with the object editor dialog that appears when clicked on the edit button.
4. Add as many numbers of rows you need, edit each row configuration and making appropriate binding to the PLC symbols.

4.14.3.2 Configure Table control to show multiple tables

Table control can be used to show multiple tables, multiple similar function blocks can be listed as sub tables and the user can navigate through each sub table using on control navigation buttons.

The instructions explain how a user can visualize multiple function blocks of the same type with an example to configure many temperature zone function blocks, by configuring [First Table](#) [[▶ 114](#)] and [Tables](#) [[▶ 117](#)] attributes.

Structure of a temperature zone function block:



Each property in this function block should be accessible to the user on the table control as a row variable. First Table [▶ 114] attribute can be configured for this purpose.

There are multiple instances of same temperature zone’s function block in the project, and they can be added to the table control using Tables attribute.



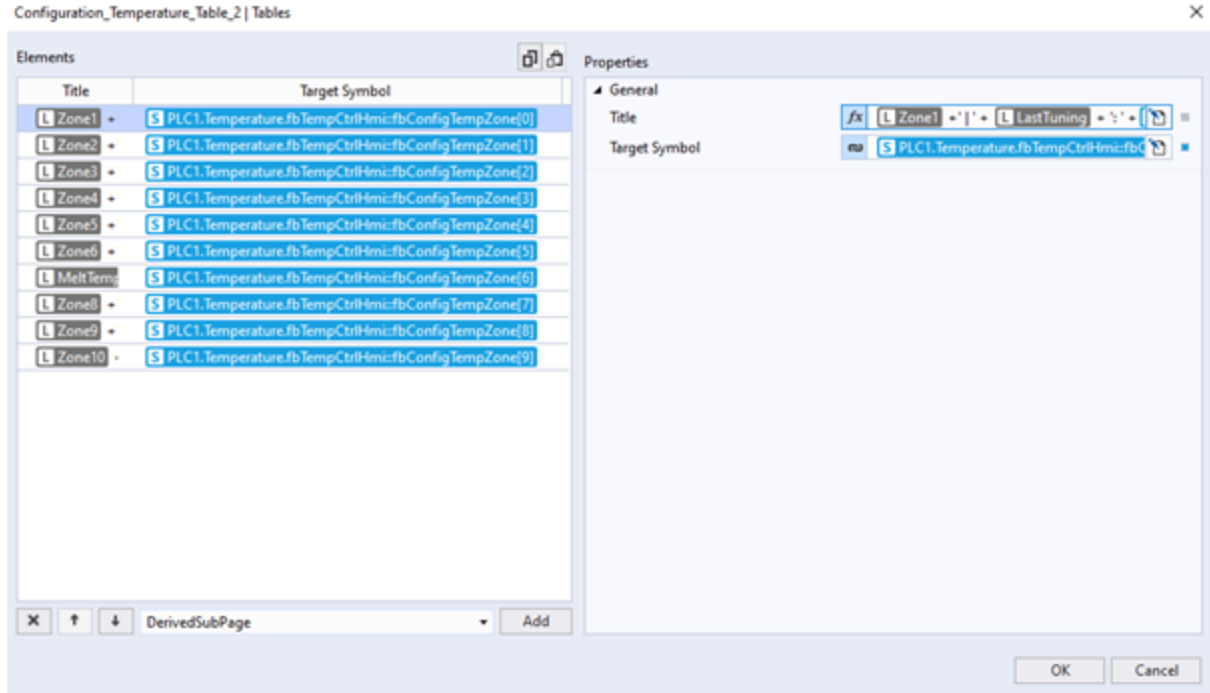
Follow these steps to configure the Table control:

1. Drag and drop table control from tools to content.
2. Configure the First Table [▶ 114] attribute to list properties of fbConfigTempZone[0] symbol for each row.
 - Add required number of rows in the object editor dialogue that appears when you edit the attribute.
 - Each row will have appropriate configuration regarding Data Type, Unit group, Max value, Min value etc.
 - Every row value property is bound to a first level variable in fbConfigTempZone[0] for example in this case AbsoluteHigh, AbsoluteLow etc.
 - Now fbConfigTempZone[0] will act as the reference symbol and each row is bound to the first level variable inside the fbConfigTempZone[0].
3. Set CreateMultipleTables [▶ 114] attribute to TRUE.
4. Configure tables attribute to list number of function blocks instances that have exactly same variables structure as the reference symbol used while configuring the FirstTable [▶ 114] attribute.

- Add multiple table elements in the editor array.
- Configure the first element in the array, the **Title** property should be set to appropriate string value.
- The Target Symbol property of the first element must be bound to the reference symbol whose first child variables are configured for each row of [FirstTable \[▸ 114\]](#) attribute. In this example this property will be bound to the symbol fbConfigTempZone[0].

The Target Symbol property of the first element in the [Tables \[▸ 117\]](#) array must be bound to the same symbol whose sub-symbols are configured as a value for every row in [First Table \[▸ 114\]](#) array.

- Set the required title for each table and set the target symbol to its respective function block symbol.



5. As a result, all 10 temperature zones are available on the control and the user can navigate to the related zone with using the sub-table navigation buttons.

Row		Unit
1	Save zone name	<input type="checkbox"/>
2	Module ID	<input type="text" value="0"/>
3	Zone ID	<input type="text" value="0"/>
4	Supply ID	<input type="text" value="0"/>
5	Extruder ID	<input type="text" value="0"/>
6	Select cooling output	NoSignal <input type="button" value="v"/>
7	Select heating output	NoSignal <input type="button" value="v"/>
8	Temperature sensor terminal	NoTerminal <input type="button" value="v"/>
9	Sensor type	NoSensor <input type="button" value="v"/>
10	Terminal channel	<input type="text" value="0"/>
11	In use	<input type="checkbox"/>
12	Enable	<input type="checkbox"/>
13	Use cooling	<input type="checkbox"/>
14	Tune cooling	<input type="checkbox"/>
15	Forced cooling enabled	<input type="checkbox"/>

Page 1|6 Zone 1|10

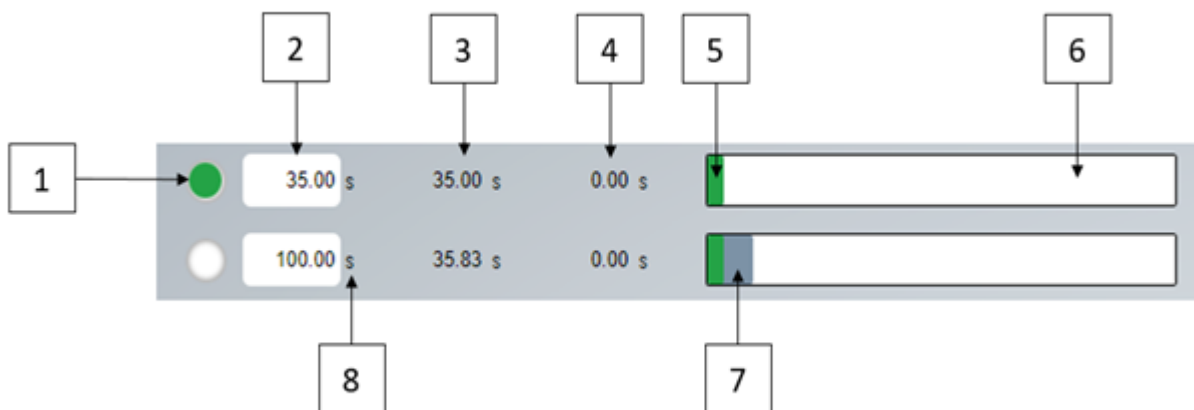
4.15 TimerControl

NuGet package: Beckhoff.TwinCAT.HMI.Plastic.Controls

Available: since version 12.6.0

The TimerControl can be used to visualize and configure an FB_Timer symbol. This symbol resembles the operation of a timer in the PLC.

On the control we can see various numerical and graphical components providing more information about actual status. User can enter SetValue property of the timer function block from the control.



1. Elapsed indication
2. Threshold InputBox

3. Current time (read only)
4. Latched time (read only)
5. Current value progress
6. Maximum duration
7. Threshold value progress
8. Unit text

4.15.1 Attributes

Category: Colors

Name	Description
Elapsed Indicator Color [▶ 124]	Fill color of the indicator when the timer elapses .

Category: Common

Name	Description
FB Timer [▶ 124]	Symbol from type FB Timer.
Progress Bar Max [▶ 125]	Largest value visible on the progress bar.
Show Elapsed [▶ 125]	Show the elapsed indicator.
Show Latched [▶ 125]	Show the latched time.
Show Progress [▶ 125]	Show the progress bar.
Threshold As Progress Max [▶ 126]	If <code>TRUE</code> , the threshold value will function as the maximum value for the progress bar visualization.

Category: Time Unit Group

Name	Description
Time Unit Group [▶ 126]	Sets the unit group for the input boxes.

4.15.1.1 ElapsedIndicatorColor

Fill color of indicator when timer elapses.

Schema: tchmi:framework#/definitions/Color

Attribute getter: getElapsedIndicatorColor

Attribute setter: setElapsedIndicatorColor

Available: since version 12.6.0

4.15.1.2 FBTimer

This attribute should be linked to a server symbol (function block or structure) that has member properties as explained below.

Name	Type	Description
ActualValue	LREAL	Actual elapsed time in LREAL.
LatchedValue	LREAL	Last latched time in LREAL.
Out	BOOL	Set to <code>TRUE</code> when timer elapsed.
SetValue	LREAL	Set time from user. This property should not be read only.

fbHmi	S	PLC1.FB_TimerHmi	✓	fbHmi
ActualValue	S	PLC1.PROPERTY-GET-LREAL	✓	ActualValue
LatchedValue	S	PLC1.PROPERTY-GET-LREAL	✓	LatchedValue
Out	S	PLC1.PROPERTY-GET-BOOL	✓	Out
SetValue	S	PLC1.PROPERTY-GET-SET-LREAL	✓	SetValue

Schema: tchmi:framework#/definitions/Symbol

Attribute getter: getTimer

Attribute setter: setTimer

Available: since version 12.6.0

4.15.1.3 ProgressBarMax

This will define the maximum threshold value a user can enter.

It will also function as the progress bar maximum limit if [ThresholdAsProgressMax \[►_126\]](#) attribute is set to FALSE.

Schema: tchmi:general#/definitions/Number

Attribute getter: getProgressBarMax

Attribute setter: setProgressBarMax

Available: since version 12.6.0

4.15.1.4 ShowElapsed

The elapsed indicator will be displayed on the control.

Schema: tchmi:general#/definitions/BOOL

Attribute getter: getShowElapsed

Attribute setter: setShowElapsed

Available: since version 12.6.0

4.15.1.5 ShowLatched

The latched time value will be displayed on the control.

Schema: tchmi:general#/definitions/BOOL

Attribute getter: getShowLatched

Attribute setter: setShowLatched

Available: since version 12.6.0

4.15.1.6 ShowProgress

A progress bar showing the current and the set time will be displayed.

Schema: tchmi:general#/definitions/BOOL

Attribute getter: getShowProgress

Attribute setter: setShowProgress

Available: since version 12.6.0

4.15.1.7 ThresholdAsProgressMax

If `TRUE`, the threshold value will function as the maximum value for the progress bar visualization:

Schema: tchmi:general#/definitions/BOOL

Attribute getter: getThresholdAsProgressMax

Attribute setter: setThresholdAsProgressMax

Available: since version 12.6.0

4.15.1.8 TimeUnitGroup

The unit group for the input boxes.

Schema: tchmi:general#/definitions/String

Attribute getter: getTimeUnitGroup

Attribute setter: setTimeUnitGroup

Available: since version 12.6.0

4.16 VariableViewControl

NuGet package: Beckhoff.TwinCAT.HMI.Plastic.Controls

Available: since version 12.6.0

The `VariableViewControl` can be used as a dashboard to visualize live values from multiple variables on the screen.

User can change the configuration during live session directly on the client screen and decide which variables will be displayed.

It also allows individual positioning of each value to allow better organizing variables in groups.



4.16.1 Attributes

Category: Colors

Name	Description
TileBackgroundColor [►_130]	Sets the tile background color for each tile.
LabelBackgroundColor [►_129]	Sets every tile's label background color.
LabelColor [►_130]	Sets the label text color for each tile.
BorderColor [►_128]	Color for the tile border.

Category: TileDesign

Name	Description
LabelFontSize [▶ 130]	Sets the label font size for every tile.
LabelFontSizeUnit [▶ 130]	The unit for the LabelFontSize [▶ 130] attribute.
ValueFontSize [▶ 131]	Sets the font size of the value displayed inside each tile.
ValueFontSizeUnit [▶ 132]	Value font size unit for each tile.
TileWidth [▶ 131]	Sets the tile width for each tile in the control.
TileWidthUnit [▶ 131]	Tile width unit.
TileHeight [▶ 130]	Sets the tile height for each tile in the control.
TileHeightUnit [▶ 131]	Tile height unit.
TileSpacing [▶ 131]	Sets the space between tiles.
TileSpacingUnit [▶ 131]	Unit of tile spacing.
BorderRadius [▶ 129]	Sets the border radius to every tile in the control.
BorderWidth [▶ 129]	Width of the border for each tile.
BorderStyle [▶ 129]	Defines the style of border for each tile. Valid values are <code>Solid</code> , <code>Dashed</code> , <code>Dotted</code> and <code>None</code> .

Category: ConfigPopupDesign

Name	Description
Config PopUp Font Size [▶ 129]	Sets the font size to the group and variable list which is displayed inside the configuration pop up window.
Config PopUp Font SizeUnit [▶ 129]	Unit of Config Popup Font Size [▶ 129] attribute.

Category: Common

Name	Description
Variable View Mode [▶ 134]	Single row or multi row view of the control.
Variable Collection [▶ 132]	List of all variables that can be selected by the user to show on screen.
Active Config [▶ 128]	List of variables shown on the control at startup.

4.16.1.1 ActiveConfig

List of variables shown on the control at startup.

Schema: `tchmi:framework#/definitions/VariableConfigList`

Attribute getter: `getActiveConfig`

Attribute setter: `setActiveConfig`

Available: since version 12.6.0

4.16.1.2 BorderColor

Color for the tile border.

Schema: `tchmi:framework#/definitions/Color`

Attribute getter: `getBorderColor`

Attribute setter: `setBorderColor`

Available: since version 12.6.0

4.16.1.3 BorderStyle

Defines the style of the border for each tile. Valid values are `Solid`, `Dashed`, `Dotted` and `None`.

Schema: `tchmi:framework#/definitions/BorderStyle`

Attribute getter: `getBorderStyle`

Attribute setter: `setBorderStyle`

Available: since version 12.6.0

4.16.1.4 BorderRadius

Sets the border radius to every tile in the control.

Schema: `tchmi:framework#/definitions/BorderRadius`

Attribute getter: `getBorderRadius`

Attribute setter: `setBorderRadius`

Available: since version 12.6.0

4.16.1.5 BorderWidth

Width of the border for each tile.

Schema: `tchmi:framework#/definitions/BorderWidth`

Attribute getter: `getBorderWidth`

Attribute setter: `setBorderWidth`

Available: since version 12.6.0

4.16.1.6 ConfigPopUpFontSize

Sets the font size to the group and variable list which is displayed inside the configuration pop up window.

Schema: `tchmi:framework#/definitions/MeasurementValue`

Attribute getter: `getConfigPopUpFontSize`

Attribute setter: `setConfigPopUpFontSize`

Available: since version 12.6.0

4.16.1.7 ConfigPopUpFontSizeUnit

Unit of the font size for the configuration pop up window.

Schema: `tchmi:framework#/definitions/MeasurementUnit`

Attribute getter: `getConfigPopUpFontSizeUnit`

Attribute setter: `setConfigPopUpFontSizeUnit`

Available: since version 12.6.0

4.16.1.8 LabelBackgroundColor

Sets every tile's label background color.

Schema: `tchmi:framework#/definitions/Color`

Attribute getter: getLabelBackgroundColor

Attribute setter: setLabelBackgroundColor

Available: since version 12.6.0

4.16.1.9 LabelColor

Sets the label text color for each tile.

Schema: tchmi:framework#/definitions/Color

Attribute getter: getLabelColor

Attribute setter: setLabelColor

Available: since version 12.6.0

4.16.1.10 LabelFontSize

Sets the label font size for every tile.

Schema: tchmi:framework#/definitions/MeasurementValue

Attribute getter: getLabelFontSize

Attribute setter: setLabelFontSize

Available: since version 12.6.0

4.16.1.11 LabelFontSizeUnit

The unit for the label font size.

Schema: tchmi:framework#/definitions/MeasurementUnit

Attribute getter: getLabelFontSizeUnit

Attribute setter: setLabelFontSizeUnit

Available: since version 12.6.0

4.16.1.12 TileBackgroundColor

Sets the tile background color for each tile.

Schema: tchmi:framework#/definitions/Color

Attribute getter: getTileBackgroundColor

Attribute setter: setTileBackgroundColor

Available: since version 12.6.0

4.16.1.13 TileHeight

Sets the tile height for each tile in the control.

Schema: tchmi:framework#/definitions/MeasurementValue

Attribute getter: getTileHeight

Attribute setter: setTileHeight

Available: since version 12.6.0

4.16.1.14 TileHeightUnit

Tile height unit.

Schema: tchmi:framework#/definitions/MeasurementUnit

Attribute getter: getTileHeightUnit

Attribute setter: setTileHeightUnit

Available: since version 12.6.0

4.16.1.15 TileSpacing

Sets the space between the tiles.

Schema: tchmi:framework#/definitions/MeasurementValue

Attribute getter: getTileSpacing

Attribute setter: setTileSpacing

Available: since version 12.6.0

4.16.1.16 TileSpacingUnit

Unit of the tile spacing.

Schema: tchmi:framework#/definitions/MeasurementUnit

Attribute getter: getTileSpacingUnit

Attribute setter: setTileSpacingUnit

Available: since version 12.6.0

4.16.1.17 TileWidth

Sets the tile width for each tile in the control.

Schema: tchmi:framework#/definitions/MeasurementValue

Attribute getter: getTileWidth

Attribute setter: setTileWidth

Available: since version 12.6.0

4.16.1.18 TileWidthUnit

Tile width unit.

Schema: tchmi:framework#/definitions/MeasurementUnit

Attribute getter: getTileWidthUnit

Attribute setter: setTileWidthUnit

Available: since version 12.6.0

4.16.1.19 ValueFont Size

Sets the font size of the value displayed inside of each tile.

Schema: tchmi:framework#/definitions/MeasurementValue

Attribute getter: getValueFontSize

Attribute setter: setValueFontSize

Available: since version 12.6.0

4.16.1.20 ValueFontSizeUnit

Value font size unit for each tile.

Schema: tchmi:framework#/definitions/MeasurementUnit

Attribute getter: getValueFontSizeUnit

Attribute setter: setValueFontSizeUnit

Available: since version 12.6.0

4.16.1.21 VariableCollection

If a user wants to add a new variable to be monitored on the VariableViewControl, the control needs the following information for the new variable:

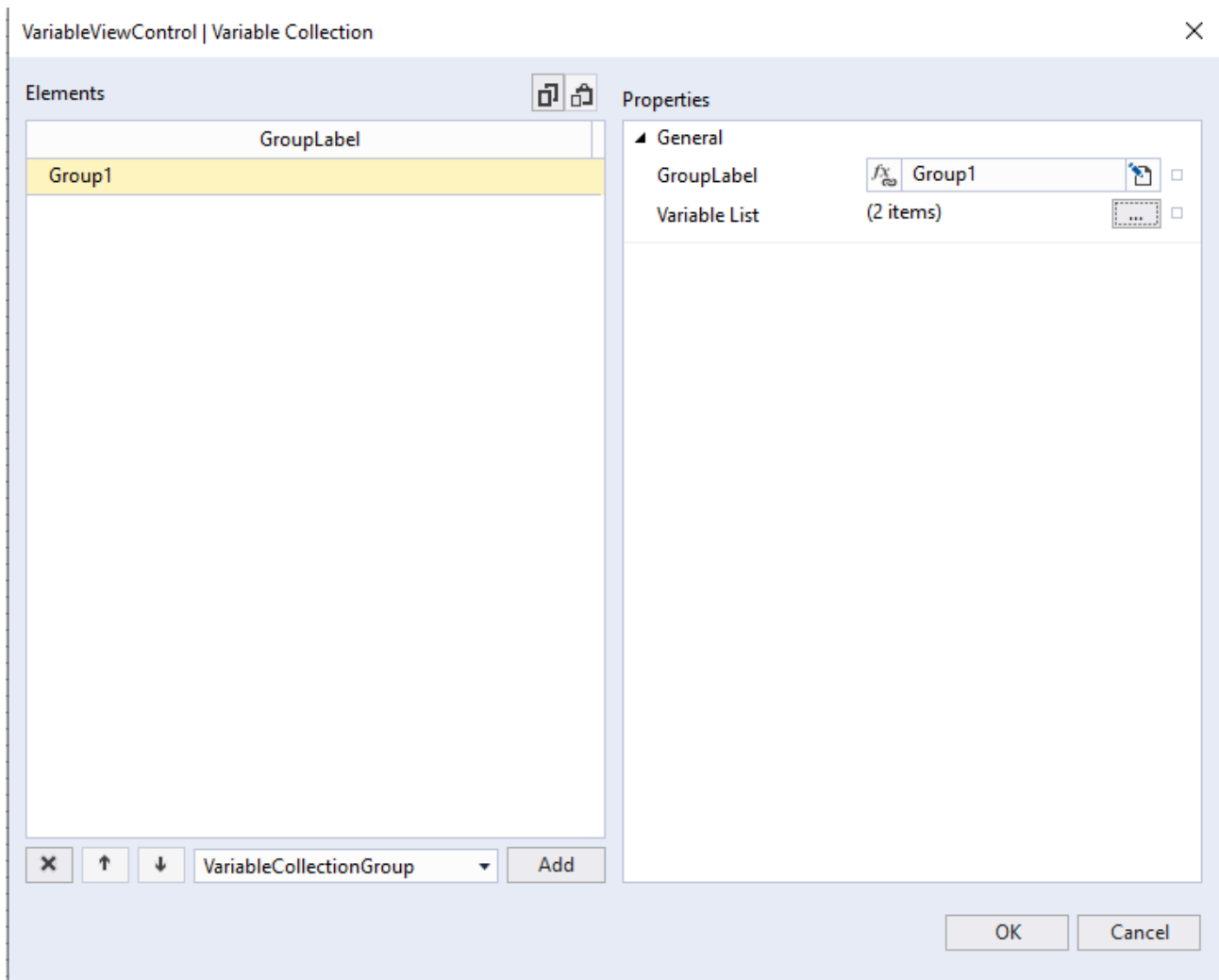
- Path of the symbol to read the value.
- Label text which can be a localized symbol or a function binding.
- Unit group of that variable to allow unit conversion for proper display of values.

This information can be prepared using the Variable Collection attribute.

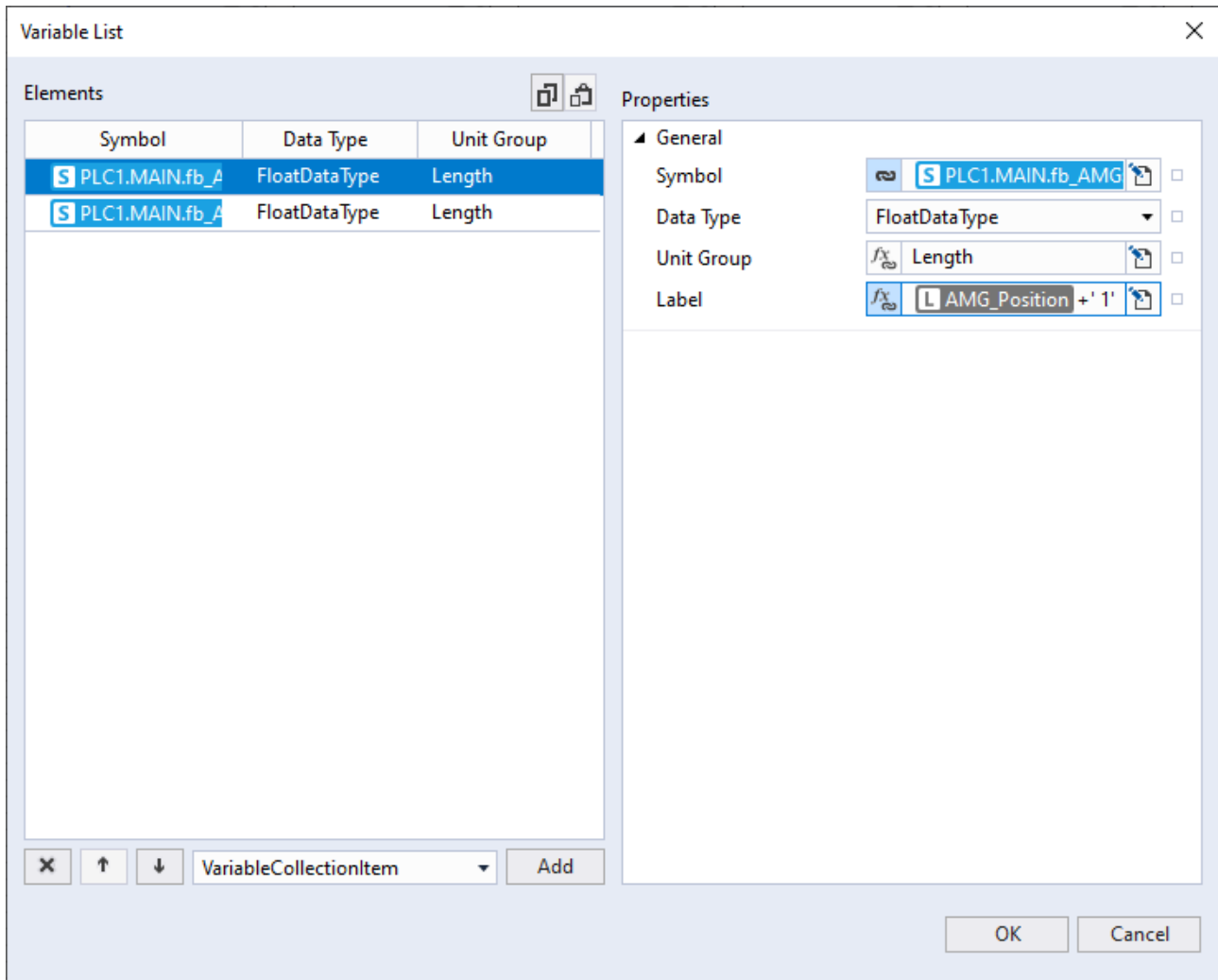
Variable Collection attribute is also used by the configuration dialog that appears on the control.

To make it easier for user to select an appropriate variable, all the variables are divided into multiple groups.

The name for each group is configured using the GroupLabel attribute in the object editor dialog in engineering environment by the user.



Every group of variables will have Group Label and Variable List.



Every variable list can have multiple variables, each variable has below properties:

- Symbol: Binding path to the symbol to display on screen.
- Data Type: For a better display the user can define the data type as `FloatDataType`, `IntDataType` or `BoolDataType`.
- Unit Group: Unit group of the displayed symbol.
- Label: Text to be displayed as tile identifier.

List of all variables that can be selected by the user to show on screen.

Schema: `tchmi:framework#/definitions/VariableCollection`

Attribute getter: `getVariableCollection`

Attribute setter: `setVariableCollection`

Available: since version 12.6.0

4.16.1.22 VariableViewMode

Single row or multi row view of the control.

Schema: `tchmi:framework#/definitions/VariableViewMode`

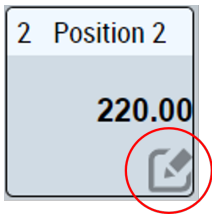
Attribute getter: `getVariableViewMode`

Attribute setter: `setVariableViewMode`

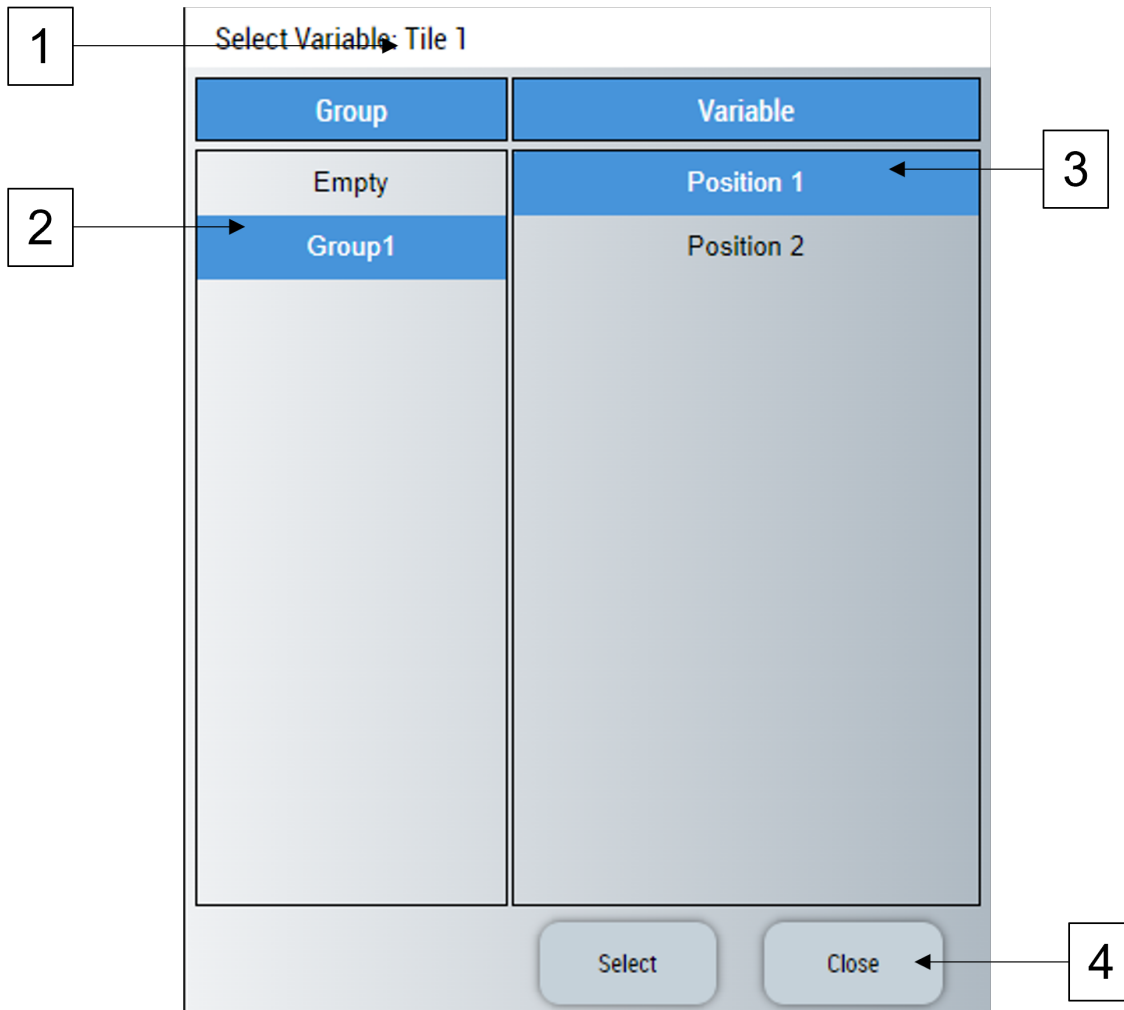
Available: since version 12.6.0

4.16.2 Dialogs

4.16.2.1 Change displayed variable of a tile



By clicking on the edit button on any of the tile of the control, the user can trigger a configuration pop up. On this pop up the user can change the variable to be displayed on the selected tile.



1. The tile number of the tile which is configured.
2. Group can be selected as per configured in the [Variable Collection \[►_132\]](#) attribute.
3. Symbol can be selected by selecting the descriptor text associated with the symbol.
4. User can click on select or cancel buttons to confirm or cancel changes.

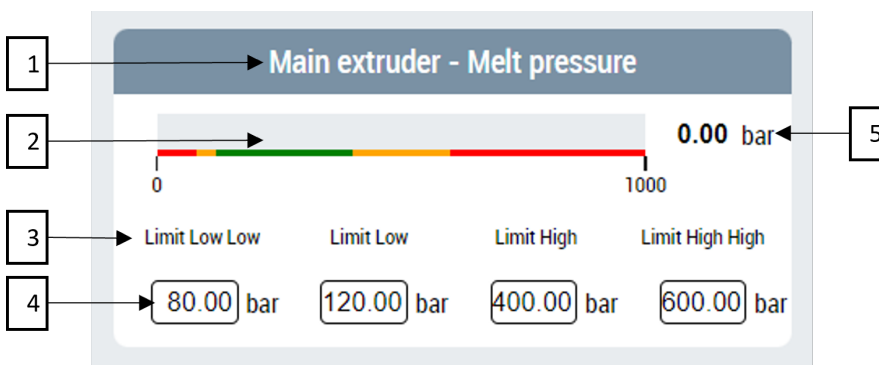
4.17 Monitoring

Monitoring control is used to visualize and configure multiple FB_Monitoring objects in the PLC from a single framework control.

The image shows how seven different instances of FB_Monitoring are visualized on this control.



Every individual monitoring object will be represented with its own element.



Components of every monitoring element:

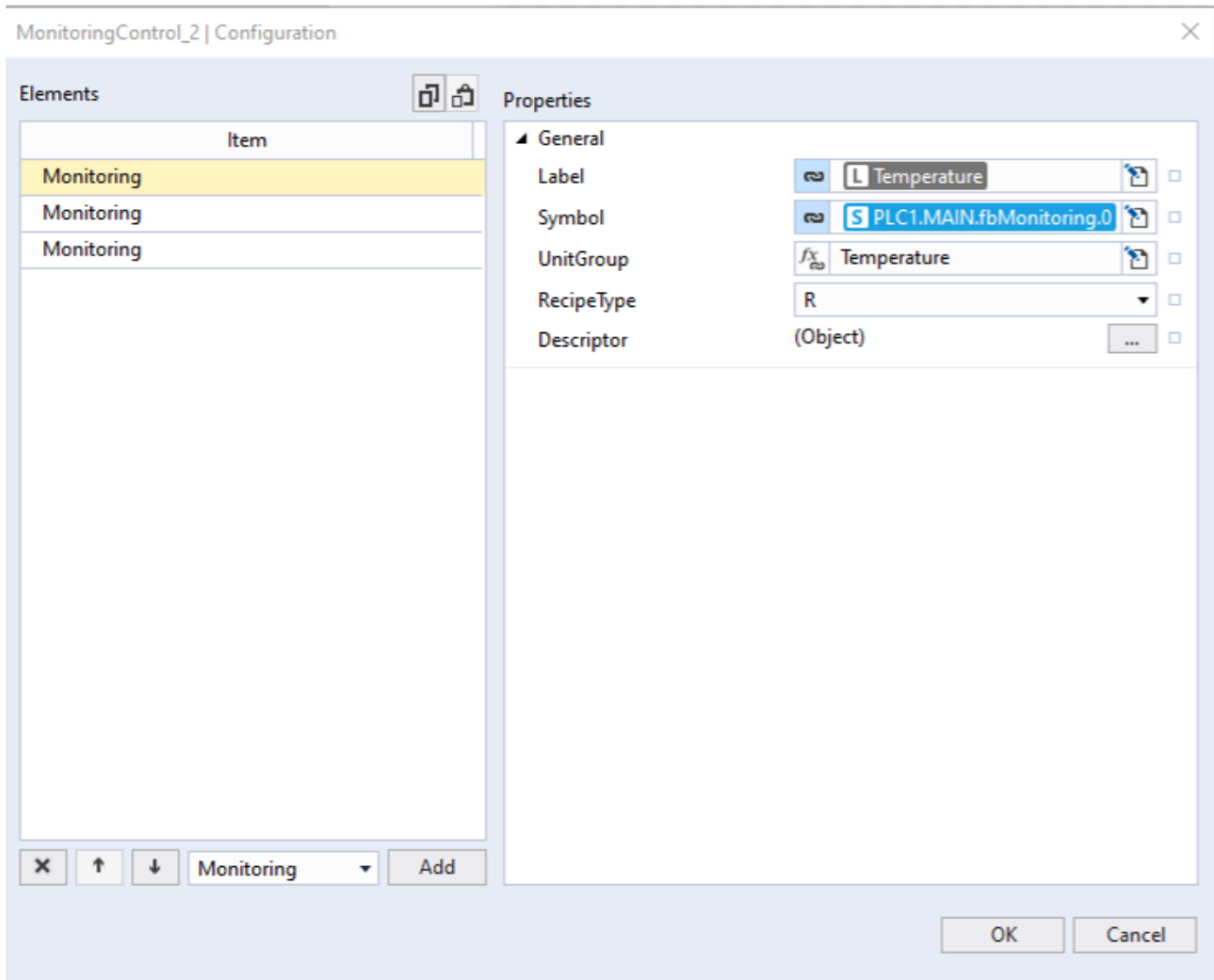
1. Header text or label
2. A linear gauge showing the actual value against colored ranges that highlight different limits.
3. Descriptor text for each input field.
4. Input controls allowing user to enter Limit Low Low, Limit Low, Limit High and Limit High High settings.

5. Value being monitored in digits.

Available: since version 12.8.0

4.17.1 Attributes

4.17.1.1 Configuration



Configuration is an array of objects. Every property in these objects is explained below.

Name	Type	Description
Label	string	This is the text that will be displayed on the header area of the respective monitoring element.
Symbol	tchmi:framework#/definitions/Symbol	Symbol of the linked FB_Monitoring function block variable.
UnitGroup	string	Unit group used for unit conversion of all numerical values.
RecipeType	tchmi:framework#/definitions/RecipeType	If the inputs from the user are to be saved in recipe, the target recipe type can be defined here.
Descriptor	tchmi:framework#/definitions/Descriptor	With this object the user can set text values that will be displayed as descriptor text for different limit values in the control. If this object is null, default values will be set as descriptors.

Every FB_Monitoring object in PLC must have properties as explained below:

Name	Type	Description
ActualValue	Float	Actual value to be monitored.
LimitHH	Float	Upper critical limit.
LimitH	Float	Upper warning limit.
LimitLL	Float	Lower critical limit.
LimitL	Float	Lower warning limit.
ScalingValueMin	Float	Minimum value acts as lower limit of linear gauge.
ScalingValueMax	Float	Maximum value acts as upper limit of the linear gauge.

Schema: tchmi:framework#/definitions/MonitoringConfig

Attribute getter: getConfiguration

Attribute setter: setConfiguration

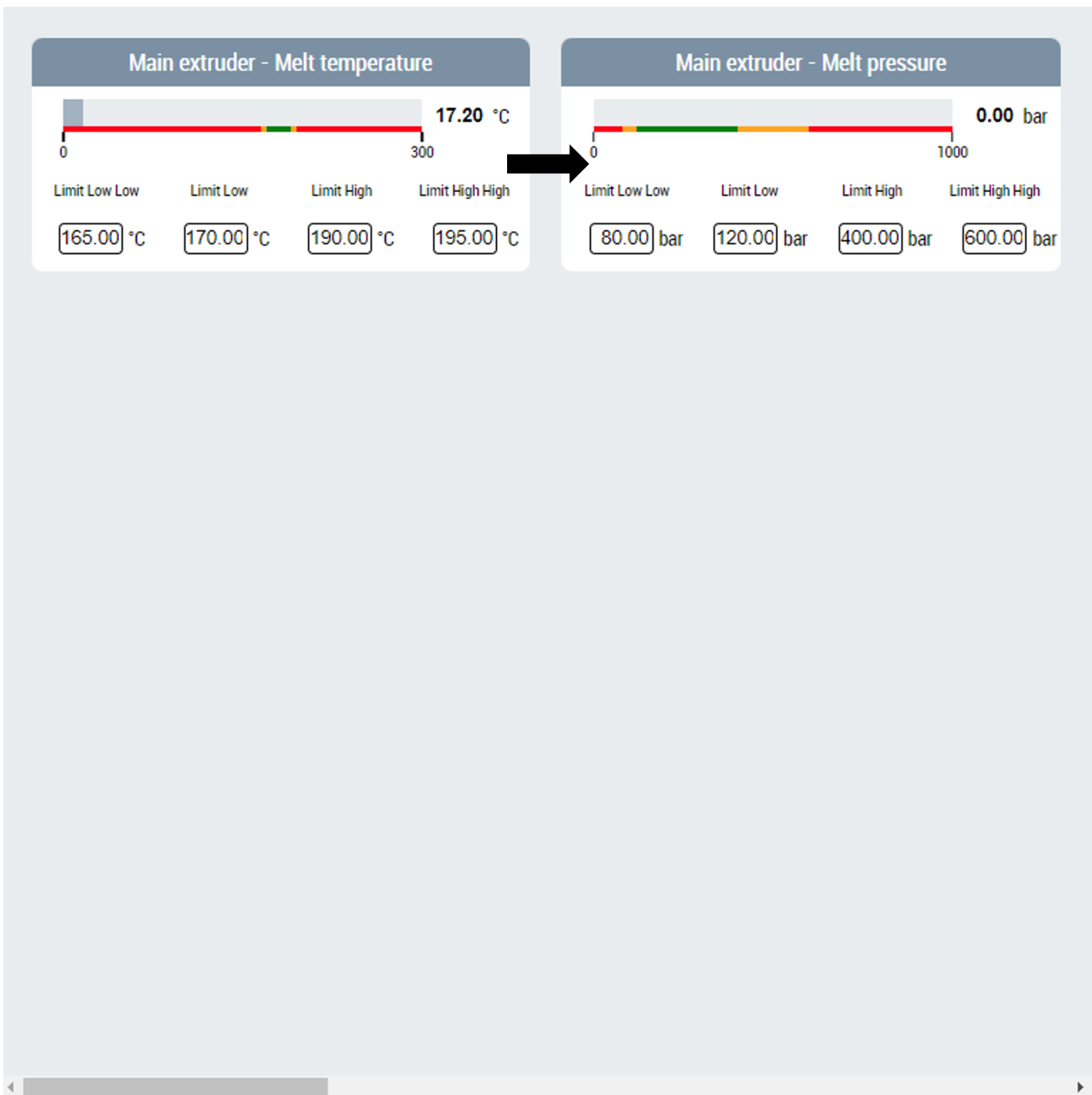
Available: since version 12.8.0

4.17.1.2 Orientation

With monitoring control the user can visualize multiple monitoring objects in one control. The Orientation attribute can be used to select the direction in which these multiple monitoring elements will be arranged on the screen.

Two possible values of this attribute are:

Horizontal: In horizontal orientation all the elements will be arranged horizontally next to each other.



Vertical: In vertical orientation elements will be arranged vertically to each other.



Schema: tchmi:framework#/definitions/Orientation

Attribute getter: getOrientation

Attribute setter: setOrientation

Available: since version 12.8.0

4.17.1.3 Wrap

With wrap attribute the user could choose if the elements organized on the control should be wrapped.

In horizontal orientation wrapping will wrap new elements into next rows.



In vertical orientation wrapping will add new elements into next column.



Schema: tchmi:general#/definitions/Boolean

Attribute getter: getWrap

Attribute setter: setWrap

Available: since version 12.8.0

4.17.1.4 RowGap

Row gap is the vertical gap between two monitoring elements.

Schema: tchmi:framework#/definitions/MeasurementValue

Attribute getter: getRowGap

Attribute setter: setRowGap

Available: since version 12.8.0

4.17.1.5 RowGapUnit

Unit for the row gap attribute.

Schema: tchmi:framework#/definitions/MeasurementValue

Attribute getter: getRowGapUnit

Attribute setter: setRowGapUnit

Available: since version 12.8.0

4.17.1.6 ColumnGap

Column gap is the horizontal gap between two monitoring elements.

Schema: tchmi:framework#/definitions/MeasurementValue

Attribute getter: getColumnGap

Attribute setter: setColumnGap

Available: since version 12.8.0

4.17.1.7 ColumnGapUnit

Unit for column gap attribute.

Schema: tchmi:framework#/definitions/MeasurementValue

Attribute getter: getColumnGapUnit

Attribute setter: setColumnGapUnit

Available: since version 12.8.0

4.18 Functions

4.18.1 UnitConversion

4.18.1.1 ConvToDisplayUnit

```
function ConvToDisplayUnit(GroupID: string, SrcVal: number, RefVal: number): number
```

Converts given SrcVal value in system unit to right display unit value, using provided group ID and reference value.

Parameter

Name	Type	Description
GroupID	string	The unit group of the source value.
SrcVal	number	The value to be converted.
RefVal	number	Reference value resulting in 100% if display or system unit is Percent.

Return value

Type	Description
number	Result in display unit of given unit group.

NuGet package: Beckhoff.TwinCAT.HMI.Plastic.Controls

Available: since version 12.6.0

4.18.1.2 ConvToSystemUnit

```
function ConvToSystemUnit(GroupID: string, SrcVal: number, RefVal: number): number
```

Converts given value in display unit to appropriate system unit value, using provided GroupID and RefVal parameters.

Parameter

Name	Type	Description
GroupID	string	The unit group of the source value.
SrcVal	number	The value to be converted.
RefVal	number	Reference value resulting in 100% if display or system unit is Percent.

Return value

Type	Description
number	Result in system unit of given unit group.

NuGet package: Beckhoff.TwinCAT.HMI.Plastic.Controls

Available: since version 12.6.0

4.18.1.3 GetConfiguredPercentUnitID

```
GetConfiguredPercentUnitID(): string
```

Returns configured PercentUnit ID.

Parameter

Name	Type	Description
-	-	-

Return Value

Type	Description
string	Unit_Id property of the PercentUnit configured in unit_config.json [► 40]

NuGet package: Beckhoff.TwinCAT.HMI.Plastic.Controls

Available: since version 12.6.0

4.18.1.4 GetDisplayUnitForUnitGroup

```
function GetDisplayUnitForUnitGroup(ctx: Context, GroupID: string)
```

Gets configured display unit ID for given unit group, waits for Configurator control to load unit config from JSON file.

Parameter

Name	Type	Description
ctx	TcHmi.Context	<ul style="list-style-type: none"> Context object Does not need to be specified by the user when used in a trigger or function expression. In this case the transfer and the evaluation is done by the corresponding parser. Is used by the caller to indicate success or failure to the framework during execution. If the function is used directly in JavaScript/TypeScript, the context object must be passed in the call and evaluated if necessary.
GroupID	string	The unit group ID.

Return value

Type	Description
void	No return value.

NuGet package: Beckhoff.TwinCAT.HMI.Plastic.Controls

Available: since version 12.6.0

4.18.1.5 GetDispUnitText

```
function GetDispUnitText(GroupID: string): string
```

Returns UnitText property of the unit that is configured as display unit for given unit group parameter.

Parameter

Name	Type	Description
GroupID	string	The unit group ID.

Return value

Type	Description
string	UnitText property of unit that is set as display unit for given unit group.

NuGet package: Beckhoff.TwinCAT.HMI.Plastic.Controls

Available: since version 12.6.0

4.18.1.6 GetSystemUnitForUnitGroup

```
function GetSystemUnitForUnitGroup (ctx: Context, GroupID: string)
```

Gets configured system unit ID for given unit group, waits for Configurator control to load unit config from JSON file.

Parameter

Name	Type	Description
ctx	TcHmi.Context	<ul style="list-style-type: none"> Context object Does not need to be specified by the user when used in a trigger or function expression. In this case the transfer and the evaluation is done by the corresponding parser. Is used by the caller to indicate success or failure to the framework during execution. If the function is used directly in JavaScript/TypeScript, the context object must be passed in the call and evaluated if necessary.
GroupID	string	The unit group ID.

Return value

Type	Description
void	No return value.

NuGet package: Beckhoff.TwinCAT.HMI.Plastic.Controls

Available: since version 12.6.0

4.18.1.7 GetUnitConverted

```
function GetUnitConverted(SrcVal: number, SrcUnitID: string, TargetUnitID: string, RefVal: number): number
```

Converts value in source measurement unit to target measurement unit. If no valid conversion formula is configured in [unit_config.json](#) [► 40], the function will return source value without changing it.

Parameter

Name	Type	Description
SrcVal	number	Value to be converted.
SrcUnitID	string	Unit group ID in which source value is calculated.
TargetUnitID	string	Unit group ID into which result should be calculated.
RefVal	number	Reference value resulting in 100% if display or system unit is Percent.

Return value

Type	Description
number	Result of unit conversion.

NuGet package: Beckhoff.TwinCAT.HMI.Plastic.Controls

Available: since version 12.6.0

4.18.1.8 GetUnitTextForUnit

```
function GetUnitTextForUnit(ctx: Context, UnitID: string)
```

Gets unit text property of unit identified by UnitID parameter, waits for Configurator control to load unit config from `unit_config.json` [▶ 40] file.

Parameter

Name	Type	Description
ctx	<u>TcHmi.Context</u>	<ul style="list-style-type: none"> Context object Does not need to be specified by the user when used in a trigger or function expression. In this case the transfer and the evaluation is done by the corresponding parser. Is used by the caller to indicate success or failure to the framework during execution. If the function is used directly in JavaScript/TypeScript, the context object must be passed in the call and evaluated if necessary.
UnitID	string	Unit ID of the unit whose text property is to be read.

Return value

Type	Description
void	No return value.

NuGet package: Beckhoff.TwinCAT.HMI.Plastic.Controls

Available: since version 12.6.0

4.18.1.9 SetDisplayUnitForUnitGroup

`SetDisplayUnitForUnitGroup(ctx: Context | null, GroupID: string, UnitID: string)`

Sets unit identified by UnitID as display unit for unit group identified by GroupID.

Parameter

Name	Type	Description
ctx	<u>TcHmi.Context</u>	<ul style="list-style-type: none"> Context object Does not need to be specified by the user when used in a trigger or function expression. In this case the transfer and the evaluation is done by the corresponding parser. Is used by the caller to indicate success or failure to the framework during execution. If the function is used directly in JavaScript/TypeScript, the context object must be passed in the call and evaluated if necessary.
GroupID	string	The unit group ID.
UnitID	string	The unit ID to set as display unit.

Return value

Type	Description
void	No return value.

NuGet package: Beckhoff.TwinCAT.HMI.Plastic.Controls

Available: since version 12.6.0

4.18.1.10 SetSystemUnitForUnitGroup

`SetSystemUnitForUnitGroup (ctx: Context | null, GroupID: string, UnitID: string)`

Sets given unit identified by parameter UnitID as system unit for unit group identified by parameter GroupID.

Parameter

Name	Type	Description
ctx	<u>TcHmi.Context</u>	<ul style="list-style-type: none"> Context object Does not need to be specified by the user when used in a trigger or function expression. In this case the transfer and the evaluation is done by the corresponding parser. Is used by the caller to indicate success or failure to the framework during execution. If the function is used directly in JavaScript/TypeScript, the context object must be passed in the call and evaluated if necessary.
GroupID	string	The unit group ID.
UnitID	string	The unit ID to set as system unit.

Return value

Type	Description
void	No return value.

NuGet package: Beckhoff.TwinCAT.HMI.Plastic.Controls

Available: since version 12.6.0

5 RecipeHelper

NuGet package: Beckhoff.TwinCAT.HMI.Plastic.RecipeHelper

Available: from version 12.6.0

Beckhoff.TwinCAT.HMI.Plastic.RecipeHelper is a NuGet package that can be used in a TwinCAT HMI project in addition with Beckhoff.TwinCAT.HMI.Plastic.Controls package. The RecipeHelper package adds additional functionalities to TwinCAT HMI Recipe Management Extension such as

- Easy setup of recipe structure for application.
- Automatically adding set value symbols to recipes that are used with Plastic controls.
- Temporary recipe storage and allowing option to undo any changes to original recipe.
- PLC handshake for safely changing recipe on a machine.
- Handshake with PLC to reactivate the last recipe on a startup.

The RecipeHelper package provides a set of JavaScript functions that works with the existing TwinCAT HMI Recipe Management Extension, please check more information at: [Twincat HMI recipe management](#).

Access to recipe config

If user authentication is active, you must explicitly enable the access to the recipe configuration for the desired user groups. Please check the [link](#) for more information about user access to RecipeManagement.Config.



Without read-write access to the RecipeManagement.Config symbol the RecipeHelper functions will not work as expected.

5.1 Concept

5.1.1 Generated recipe types and recipes

When the RecipeHelper NuGet package is included in a TwinCAT HMI project during the first initialization it will automatically create objects such as recipe types, recipes and server symbols that are needed in the project for its functions.

The newly added objects are listed below.

Server symbols:

1. `LastActivatedRecipe`: This is a server symbol of type String, it will store the name of the recipe that was last activated on the server.
2. `RecipeLastUpdatedTime`: This symbol stores date and time in a string format when any recipe is changed, this symbol will be added to each product recipe.
3. `UnsavedChangesInActRecipe`: This boolean symbol will be TRUE if there are latest changes in the active temporary recipe that are still not saved in the original recipe.
4. `PLC_HMI_Handshake_State`: This integer symbol will be updated when the RecipeHelper state machine is processing the handshake communication with the PLC. The state of communication is essential the information to avoid any wrong data entered to the recipe during an invalid state.

Recipe types and recipes:

1. `ActRcpDataType`: This is the recipe type for default product recipes. At the time of creation of this recipe type it will only have one variable i.e. `RecipeLastUpdatedTime` as its member.
2. The RecipeHelper will also create one recipe instance of this type named `ActRcpData`, this will function as the recipe that will always work as the current recipe and store all the changes to the original recipe.

5.1.2 Temporary recipes

The concept of using a temporary recipe is to have a single recipe that is always active on the control. Any latest changes are stored in this active temporary recipe until the user explicitly calls the `SaveActiveRecipe` function to save the recipe. This gives the user an opportunity to reload the original recipe to undo any latest changes.

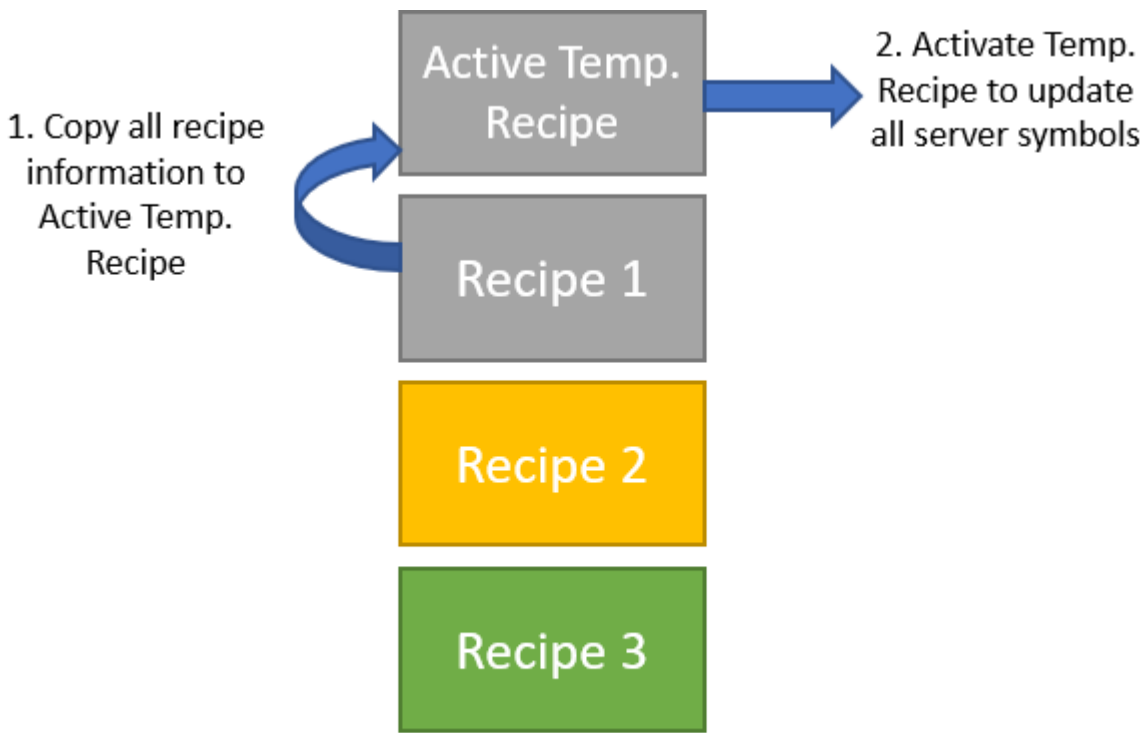
Since the active temporary changes are stored in a recipe, those changes are available on the controller even after a restart.

Different recipe operations performed in this configuration are explained below.

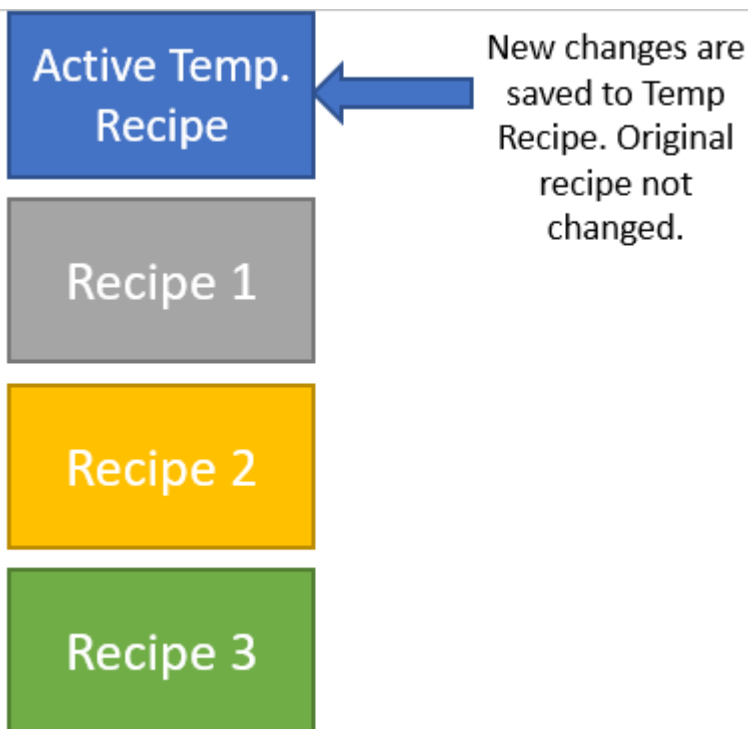
Recipe Structure: One autogenerated recipe `ActRcpData` will function as an active recipe, and there can be several recipes defined by users.



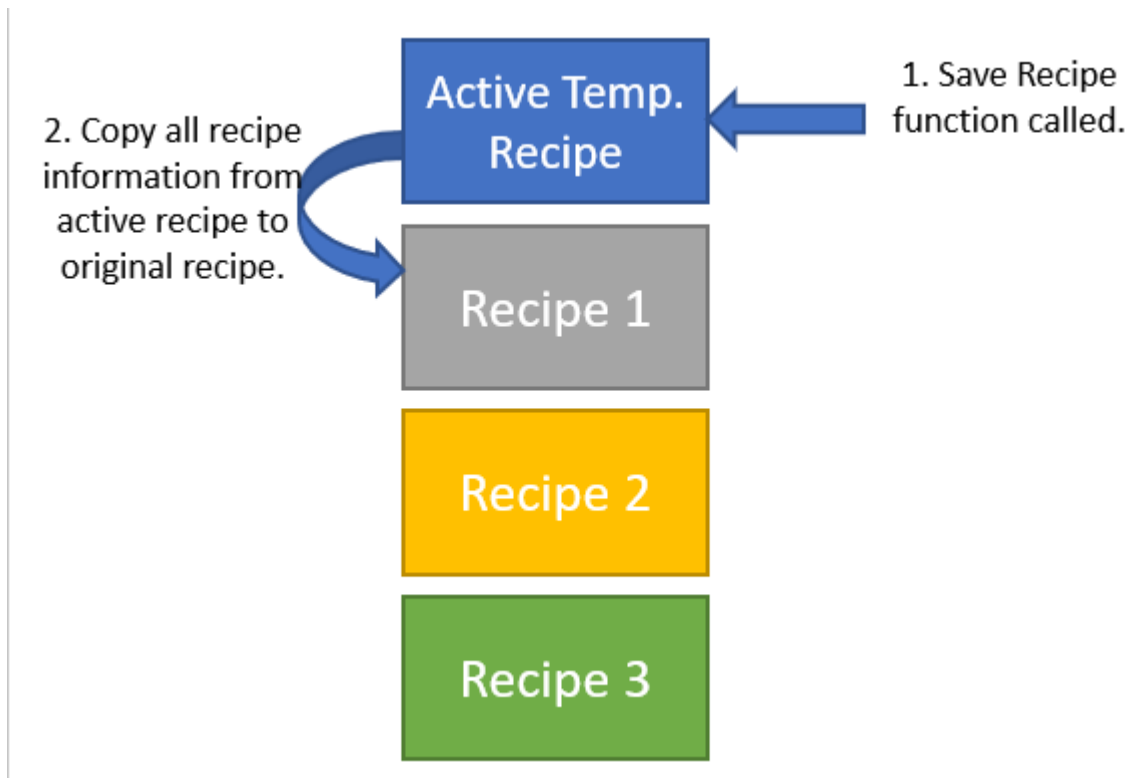
Activating a recipe: The function `LoadProductRecipe` needs to be called to load and activate a recipe. The recipe will first copy all the recipe information into the temporary recipe `ActRcpData` and after that this temporary recipe will be activated.



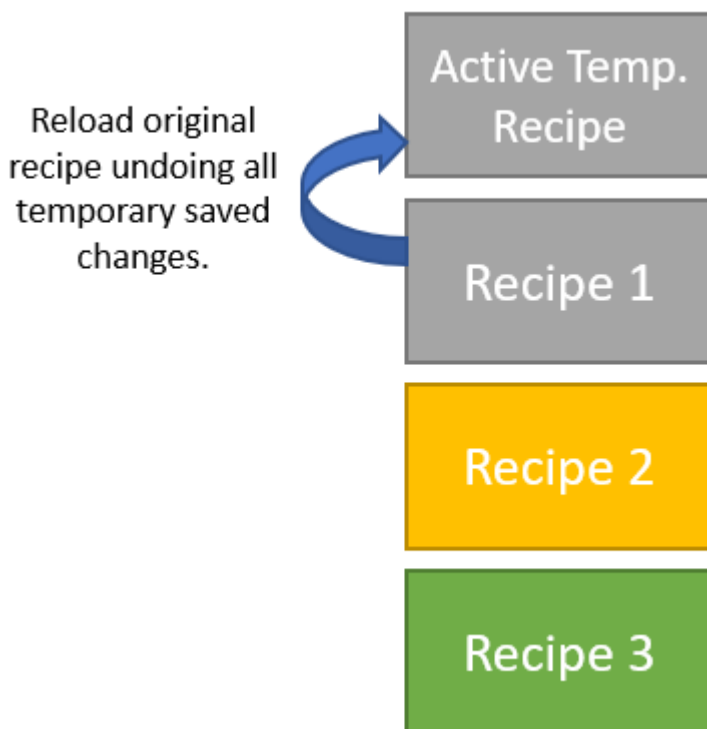
New changes by the user: All the new changes made on the controls provided in Beckhoff.TwinCAT.HMI.Plastic package will be automatically saved in the active temporary recipe without affecting the original recipe.



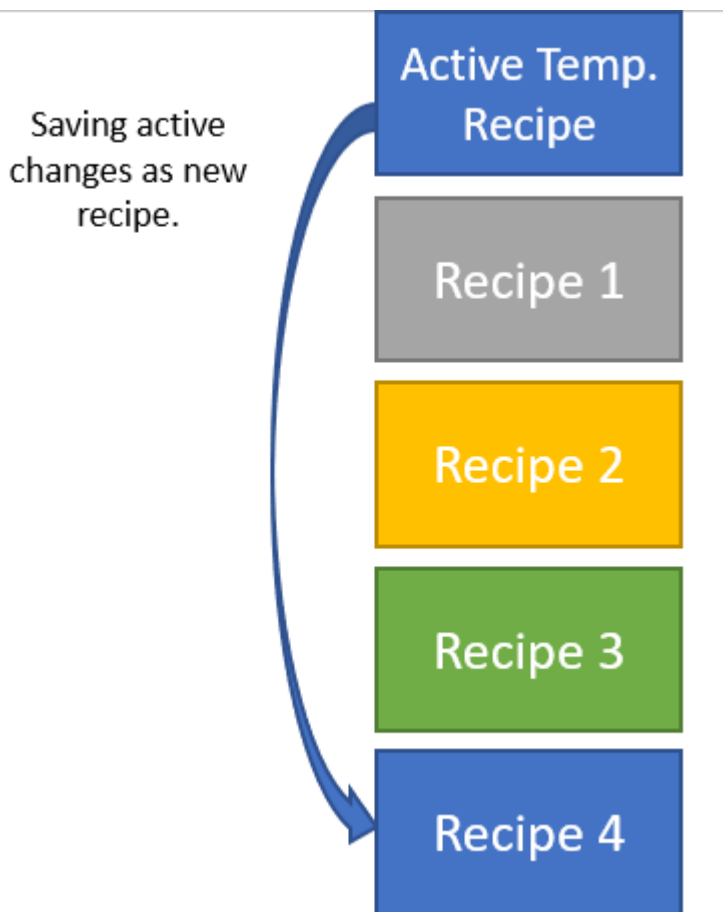
Saving a recipe: By calling the `SaveActiveRecipe` function with the target name same as the last activated recipe name the user can save all the content from active temporary recipe into its original recipe.



Undoing current changes and reloading original recipe: By calling `LoadProductRecipe` with the name of the recipe that was last activated the user can overwrite the active temporary recipe with undoing all the unsaved changes.



Saving current changes as a new recipe: Using `SaveActiveRecipe` with a new target recipe name all the active temporary recipe will be saved as a new recipe, without affecting the original recipe that was loaded before.



5.1.3 Automatic mapping of symbols and adding to recipe types

As the recipe type `ActRcpDataType` is created by the RecipeHelper package, it does not have any member variables. The user can add the members of the recipe type. Some controls provided by the Beckhoff.TwinCAT.HMI.Plastic package can automatically add symbols to this recipe type and recipes that are based on this type with using the JavaScript API functions.

For example, if a `FB_Axis` symbol for PTP motion is bound to a `FB_Axis` attribute of an `ArrowMotionGraph` control, the control on initialization will add all user configurable variables under given `FB_Axis` symbol to the `ActRcpDataType` type. Any changes by the user on the control from a browser will automatically be updated and saved into the active temporary recipe.



Adding new symbols to recipe types is only possible in the engineering environment, i.e. while editing contents and in live view.



If any content has not initialized in live view or any member symbol of the recipe is changed or removed in the PLC this must be checked in engineering before publishing the project.



After publishing the project to the target device the package will not add new symbols to the recipe. If any of the recipe symbols are not found on the target this will cause an error and recipe operations will not perform.

5.1.4 Handshake with PLC

Handshake communication with PLC is an essential feature of the RecipeHelper package. For a safe operation, the JavaScript code will execute a state machine with requests signals to PLC and waiting proper response signals. Only after a successful handshake the RecipeHelper will perform the below operations.

- changing recipe
- saving new data from PLC into recipe
- activation on startup

5.1.4.1 FB_PlcStateToHmi

PLC variables for the handshake are available in the `FB_PlcStateToHmi` function block. The below image shows properties/variables inside this function block.

ADS	object
PLC1	object
Constants	#/definit
Global_Version	#/definit
GVL	#/definit
HmiCommunication	#/definit
fbPlcStateToHmi	PLC1.HmiCommunication.fbPlcStateToHmi
AxesInitialized	PLC1.HmiCommunication.fbPlcStateToHmi:AxesInitialized
BlockedAxisDataPlc	PLC1.HmiCommunication.fbPlcStateToHmi:BlockedAxisDataPlc
BlockedTempDataPlc	PLC1.HmiCommunication.fbPlcStateToHmi:BlockedTempDataPlc
ClientID	PLC1.HmiCommunication.fbPlcStateToHmi:ClientID
DataReqFailed	PLC1.HmiCommunication.fbPlcStateToHmi:DataReqFailed
DataRequestPlc	PLC1.HmiCommunication.fbPlcStateToHmi:DataRequestPlc
DataRequestPlcActiveHmi	PLC1.HmiCommunication.fbPlcStateToHmi:DataRequestPlcActiveHmi
DataRequestPlcQuitHmi	PLC1.HmiCommunication.fbPlcStateToHmi:DataRequestPlcQuitHmi
DataValidPlc	PLC1.HmiCommunication.fbPlcStateToHmi:DataValidPlc
LiveSignHMI	PLC1.HmiCommunication.fbPlcStateToHmi:LiveSignHMI
ProductChangeConfirmPlc	PLC1.HmiCommunication.fbPlcStateToHmi:ProductChangeConfirmPlc
ProductChangeEnable	PLC1.HmiCommunication.fbPlcStateToHmi:ProductChangeEnable
ProductChangeRequestHmi	PLC1.HmiCommunication.fbPlcStateToHmi:ProductChangeRequestHmi
Reset	PLC1.HmiCommunication.fbPlcStateToHmi:Reset
SaveDataQuitPlc	PLC1.HmiCommunication.fbPlcStateToHmi:SaveDataQuitPlc
SaveDataRequestPlc	PLC1.HmiCommunication.fbPlcStateToHmi:SaveDataRequestPlc

Name	Type	Description
AxesInitialized	Bool	Used by the PLC.
BlockedAxisDataPlc	Bool	Used by the PLC.
BlockedTempDataPlc	Bool	Used by the PLC.
ClientID	UINT	The client ID of the client instance that manages the handshake communication.
DataReqFailed	Bool	Indication from the PLC that the data request has failed, need to reset the handshake state machine from the HMI.
DataRequestPlc	Bool	PLC requesting data from the active recipe.
DataRequestPlcActiveHmi	Bool	Is set to TRUE from HMI to inform that the HMI has received the request and is processing the data download to the PLC.
DataRequestPlcQuitHmi	Bool	Is set to TRUE from HMI when all data download is finished.
DataValidPlc	Bool	This is set from PLC to TRUE to indicate that the PLC has received and validated the data from the HMI and it is safe to allow user operations.
LiveSignHMI	UINT	This will be updated from the client that is overseeing the handshake operation, no live sign within a timespan indicates a connection loss.
ProductChangeEnable	Bool	Product change is allowed by the PLC.
ProductChangeRequestHmi	Bool	Request to perform a change of a recipe from the HMI.
ProductChangeConfirmPlc	Bool	Allowing a change of a recipe from the PLC.
Reset	Bool	Reset request from the HMI if the handshake failed and the DataReqFailed flag is set.
SaveDataRequestPlc	Bool	The PLC requests the HMI to save data from the PLC to the recipe.
SaveDataQuitPlc	Bool	HMI sets this variable to indicate the save data request received and completed.

5.1.4.2 Working with multiple clients

As the handshake between the PLC and the HMI, the client follows a sequential state machine. It is necessary that only one client initiates the communication and oversees it through all states.

The ClientID variable of the `FB_PlcStateToHmi` is set to a unique ID by the first client that gets access to this variable. Only the client identified by this ID will process communication requests, to inform the PLC about the connection status and it will update the `LiveSignHMI` variable every five hundred milliseconds.

If there is no change in the `LiveSignHMI` variable for a more than 5 seconds period, the PLC should remove this client by writing 'zero' to the `ClientID` variable.

If there are other clients currently active, one of them will get the access by writing its own ID into the `ClientID` variable and show its active status by updating the `LiveSignHMI` variable.

5.1.4.3 Startup state-machine

Following steps are taken to complete the communication and download the last recipe data to the PLC variables on the machine startup:

1. On the startup the PLC does not have any recipe data because the HMI may start a session later. The PLC will set the `DataRequestPLC` flag to TRUE and the `DataValidPlc` flag to FALSE.
2. The HMI client will read `DataRequestPLC` variable on initialization and set `DataRequestPLCActiveHmi` to TRUE to indicate it has received the request.
3. The recipe that is saved as the active temporary recipe will be activated from the HMI and will set the values to all the related PLC variables.
4. The HMI client will write FALSE to `DataRequestPLCActiveHmi` and TRUE to `DataRequestPLCQuitHMI` to indicate the complete activation.
5. The PLC will remove the request by resetting `DataRequestPlc` to FALSE.

5.1.4.4 Save a recipe

If there are recent changes in the PLC variables which the PLC wants to save into the active recipe, following steps will be taken to complete the process.

1. PLC raises save request by setting the `SaveDataRequestPlc` variable to TRUE.
2. HMI client reads this and saves all changes in an active temporary recipe.
3. HMI signals end of save operation by writing TRUE value to the `SaveDataQuitePlc` variable.
4. PLC recognizes this and removes the request.

5.1.4.5 Load a recipe

This handshake state sequence assures that the PLC allows a safe change of recipe and is updated when new recipe values are downloaded.

The steps for this handshake are:

1. PLC must allow a change of a recipe by setting `ProductChangeEnable` to TRUE.
2. HMI client requests a product change by setting `ProductChangeRequestHmi` to TRUE.
3. When it is safe, the PLC responds to the request by setting `ProductChangeConfirmPlc` to TRUE.
4. HMI copies the new recipe to the temporary recipe `ActRcpData` but does not activate it.
5. HMI removes the change request by resetting `ProductChangeRequestHmi` to FALSE.
6. PLC now requests the new recipe to be activated and symbol values to be downloaded by setting `DataRequestPLC` to TRUE.
7. This triggers a handshake sequence same as [Startup state-machine \[► 156\]](#) and the new recipe data will be downloaded to PLC.

5.1.4.6 DataReqFailed and Reset

1. During any state of the state machine, if the state fails to execute within a duration the PLC can inform the HMI about this failure by setting the `DataReqFailed` variable to TRUE.
2. In response the HMI will set and reset the `Reset` variable by creating a rising edge.
3. The PLC will reset its states and starts the request again from the first step.

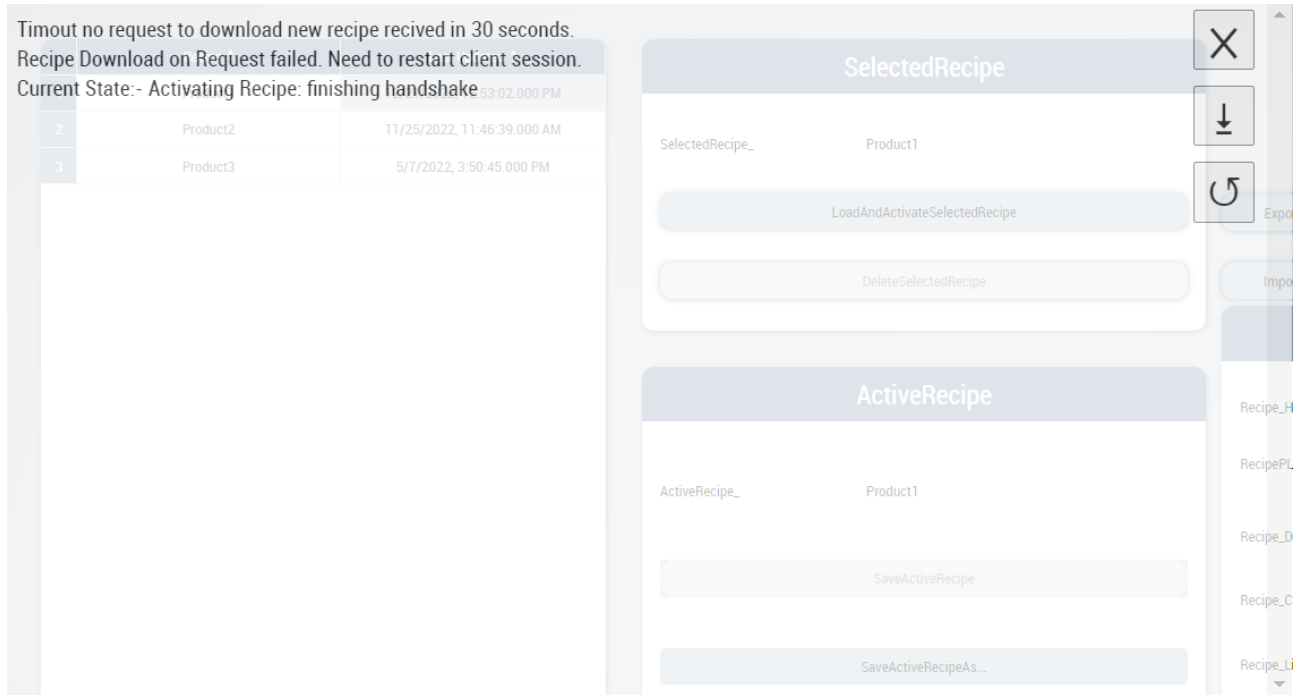
5.1.5 On screen error messages

All the operations and functions of the RecipeHelper package are using RecipeManagement APIs to access the server.

Any errors occurring during the execution of JavaScript are in client environments and are not logged into the TwinCAT Event Logger. The user cannot see these errors in HMI and must open the browser log window to see them.

The RecipeHelper package provides a utility that displays any errors in a special pop up screen on the HMI. The user can perform the following actions on this error pop up:

1. Ignore and close the pop up.
2. Download this error message for logging and contacting support about it.
3. Reset the communication handshake to retry any operation that failed.



5.2 Functions

5.2.1 SaveActiveRecipe

```
function SaveActiveRecipe (ctx: Context, TargetRecipe: string)
```

The function SaveActiveRecipe will save the active temporary recipe ActRcpData into the recipe name set by the TargetRecipe parameter.

Parameter

Name	Type	Description
ctx	Required<TcHmi.Context>	Context object. Does not need to be specified by the user when used in a trigger or function expression. In this case the transfer and the evaluation are done by the corresponding parser. Is used by the caller to indicate success or failure to the framework during execution. If the function is used directly in JavaScript/TypeScript, the context object must be passed in the call and evaluated if necessary.
TargetRecipe	string	Target recipe path.

Return value

Type	Description
void	No return value.

5.2.2 RcpAutoUpdate

```
function RcpAutoUpdate(symName: string, symNewValue: any, symRM: string, symDefValue: any)
```

The function RcpAutoUpdate will teach the active temporary recipe ActRcpData. The function will verify if DataValidPlc variable in FB_PlcStateToHmi function block is TRUE to make sure only a valid state is stored in the recipe.

Parameter

Name	Type	Description
symName	string	Symbol name of the symbol to update in the recipe. Irrespective of the symbol name passed in this parameter, the complete ActRcpData recipe will be updated from the PLC values.
symNewValue	any	The value passed in this parameter has no effect because the recipe will directly read the value from the PLC.
symRM	string	This must be 'R' and no other types of recipes are supported.
symDefValue	any	Default value if the symbol is added to the recipe type. This is only used in the engineering environment and is not possible after publishing the project.

Return value

Type	Description
void	No return value.

5.2.3 ListActDataTypeRecipes

```
function ListActDataTypeRecipes(ctx: TcHmi.Context, RecipeList?: TcHmi.Server.RecipeManagement.FolderRecipe)
```

Parameter

Name	Type	Description
ctx	Required<TcHmi.Context>	Context object. Does not need to be specified by the user when it is used in a trigger or function expression. In this case the transfer and the evaluation are done by the corresponding parser. Is used by the caller to indicate success or failure to the framework during execution. If the function is used directly in JavaScript/TypeScript, the context object must be passed in the call and evaluated if necessary.
recipeList	tchmi:server#/definitions/ folderRecipe	The reference to all recipes. The list of all recipes is the <code>recipeList</code> symbol in the RecipeManagement extension.

Returns a list of all recipes of the type `ActRcpDataType` in an array of object with each object having two properties

- `RecipeName`: Name of the recipe listed.
- `RecipeLastUpdatedTime`: Last updated time for the recipe in string form.

The function is used on the `SrcData` attribute of the data grid.

Return value

Type	Description
void	No return value.

5.2.4 LoadProductRecipe

```
function LoadProductReciepe(ctx: TcHmi.Context, NewActRecipeName: string)
```

The function `LoadProductRecipe` will load a recipe and it will follow the PLC handshake communication as described earlier in this document.

Parameter

Name	Type	Description
ctx	Required<TcHmi.Context>	Context object. Does not need to be specified by the user when it is used in a trigger or function expression. In this case the transfer and the evaluation are done by the corresponding parser. Is used by the caller to indicate success or failure to the framework during execution. If the function is used directly in JavaScript/TypeScript, the context object must be passed in the call and evaluated if necessary.
recipeList	tchmi:server#/definitions/ folderRecipe	The reference to all recipes. The list of all recipes is the <code>recipeList</code> symbol in the RecipeManagement extension.

Return value

Type	Description
void	No return value.

5.2.5 DeleteActRcp

```
function DeleteActRcp(ctx: TcHmi.Context, RecipeName: string)
```

This function allows deleting a recipe from the server and as a safety it will not allow to delete the currently activated recipe from the server.

Parameter

Name	Type	Description
ctx	Required<TcHmi.Context>	Context object. Does not need to be specified by the user when used in a trigger or function expression. In this case the transfer and the evaluation are done by the corresponding parser. Is used by the caller to indicate success or failure to the framework during execution. If the function is used directly in JavaScript/TypeScript, the context object must be passed in the call and evaluated if necessary.
RecipeName	string	Name of recipe to delete.

Return value

Type	Description
void	No return value.

5.2.6 ExportRecipe

```
function ExportRecipe(ctx: Context, recipeReference: string)
```

This function will export a recipe into a JSON file and download it on the client system.

Parameter

Name	Type	Description
ctx	Required<TcHmi.Context>	Context object. Does not need to be specified by the user when used in a trigger or function expression. In this case the transfer and the evaluation are done by the corresponding parser. Is used by the caller to indicate success or failure to the framework during execution. If the function is used directly in JavaScript/TypeScript, the context object must be passed in the call and evaluated if necessary.
recipeReference	tchmi:framework##/definitions/RecipeReference	The reference to the recipe that should be downloaded. The reference consists of the recipe path (with folder structure).

Return value

Type	Description
void	No return value.

5.2.7 ImportRecipe

```
function ImportRecipe(ctx: TcHmi.Context)
```

This function allows importing of multiple JSON files which were exported with the `ExportRecipe` function to be imported and added to the server.

Parameter

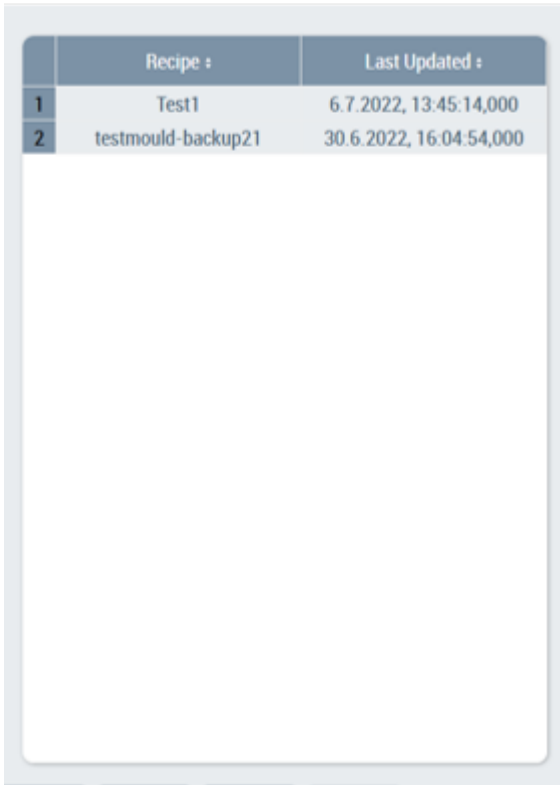
Name	Type	Description
ctx	Required<TcHmi.Context>	Context object. Does not need to be specified by the user when used in a trigger or function expression. In this case the transfer and the evaluation are done by the corresponding parser. Is used by the caller to indicate success or failure to the framework during execution. If the function is used directly in JavaScript/TypeScript, the context object must be passed in the call and evaluated if necessary.

Return value

Type	Description
void	No return value.

5.3 Using RecipeHelper with HMI controls

5.3.1 Listing Recipes on a data grid



	Recipe :	Last Updated :
1	Test1	6.7.2022, 13:45:14,000
2	testmould-backup21	30.6.2022, 16:04:54,000

1. Create an instance of a TcHmiDataGrid control.
2. Set the below JSON object as its Columns attribute.

```
[
  {
    "name": "RecipeName",
    "label": "%1%Recipe%/1%",
    "control": "TextBlock",
    "width": 50,
    "widthUnit": "%",
    "resize": false,
    "sortable": true,
    "minWidth": 100,
    "minWidthUnit": "px",
    "cellBackground": null,
    "textColor": null,
    "editable": false,
    "horizontalAlignment": "Center",
    "verticalAlignment": "Center",
    "headerHorizontalAlignment": "Center",
    "headerVerticalAlignment": "Center"
  },
  {
    "name": "RecipeLastUpdatedTime",
    "label": "Last Updated",
    "control": "TextBlock",
    "width": 50,
    "widthUnit": "%",
    "resize": false,
    "sortable": true,
    "minWidth": 100,
    "minWidthUnit": "px",
    "cellBackground": null,

```

```

"textColor": null,
"editable": false,
"horizontalAlignment": "Center",
"verticalAlignment": "Center",
"headerHorizontalAlignment": "Center",
"headerVerticalAlignment": "Center",
"format": {
"objectType": "Function",
"active": true,
"fn": "Tchmi.Functions.Beckhoff.ToDateTimeString",
"fnParams": [
{
"objectType": "StaticValue",
"valueType": "tchmi:general#/definitions/Locale",
"value": null
}
]
}
}
]

```

3. Set the function binding as below as its `SrcData` attribute.

```

"%f%Tchmi.Functions.Plastic.ListActDataTypeRecipes(%s%TchmiRecipeManagement.Config::recipeList%/s)%/f%"

```

5.3.2 Display active recipe with unsaved changes

1. Bind a text block to the server symbol `LastActivatedRecipe`. This will display the last activated recipe.
2. The symbol `UnsavedChangesInActRcpData` indicates that there are unsaved changes when it is `TRUE` and can be used to add an asterisk character after the active recipe name.

Product2*

6 Functions

NuGet package: Beckhoff.TwinCAT.HMI.Plastic.Functions

Available: from version 12.6.0

The NuGet package **Beckhoff.TwinCAT.HMI.Plastic.Functions** can be installed in the TwinCAT HMI project via the NuGet package management system.

6.1 Axes

6.1.1 ChangeInputBoxUnit

```
function ChangeInputBoxUnit (InputBox: TcHmi.Controls.Plastic.InputBox): void
```

The `ChangeInputBoxUnit` function changes the unit group of an `InputBox` control into an angular unit group, if the symbol in the `TextVal` property of the `InputBox` control is a value from a `FB_Axis` function block of the `Tc3_PlasticBaseApplication` library and the axis is a transformation axis.

Unit group	Angular unit group
Length	Angular
Velocity	AngularVelocity
Acceleration	AngularAcceleration
Jerk	AngularJerk

By default, this function is only used by the `UpdateAxesHomingUnits` [▶ 164] function

Parameter

Name	Type	Description
InputBox	TcHmi.Controls.Plastic.InputBox	The control whose UnitGroup is to be changed.

Return value

Type	Description
void	no return value

NuGet package: Beckhoff.TwinCAT.HMI.Plastic.Functions

Available from version 12.6.0

Usage: TC3 Plastic Application HMI projects

6.1.2 UpdateAxesHomingUnits

```
function UpdateAxesHomingUnits (ContentId: string): void
```

The `UpdateAxesHomingUnits` function changes the unit groups of `InputBox` controls to an angular unit group if the symbol in the `TextVal` property of the `InputBox` control is a value from a `FB_Axis` function block of the `Tc3_PlasticBaseApplication` library and the axis is a transformation axis. The function is intended for use on a TwinCAT HMI page where the homing parameters of an axis are connected to `InputBox` controls. The symbols used can be found in the `FB_Homing` function block of the `Tc3_PlasticBaseApplication` library.

`InputBox` controls with the following IDs are processed by the function, where `ContentId` is the input parameter of the function:

- `ContentId + _HomingBlock_Velocity_InputBox`
- `ContentId + _HomingBlock_Acceleration_InputBox`

- ContentId + _HomingBlock_Deceleration_InputBox
- ContentId + _HomingBlock_Jerk_InputBox
- ContentId + _HomingBlock_DistanceLimit_InputBox
- ContentId + _HomingBlock_SetPosition_InputBox
- ContentId + _HomingBlockDetect_Velocity_InputBox
- ContentId + _HomingBlock_DetectionVelocityLimit_InputBox
- ContentId + _HomingBlockDetect_Acceleration_InputBox
- ContentId + _HomingBlockDetect_Deceleration_InputBox
- ContentId + _HomingBlockDetect_Jerk_InputBox
- ContentId + _HomingBlockDetect_DistanceLimit_InputBox
- ContentId + _HomingBlockDetect_RecordedPosition_InputBox
- ContentId + _HomingBlockDetect_DetectionVelocityLimit_InputBox
- ContentId + _HomingFinish_Distance_InputBox
- ContentId + _HomingFinish_Velocity_InputBox
- ContentId + _HomingFinish_Acceleration_InputBox
- ContentId + _HomingFinish_Deceleration_InputBox

Unit group	Angular unit group
Length	Angular
Velocity	AngularVelocity
Acceleration	AngularAcceleration
Jerk	AngularJerk

Parameter

Name	Type	Description
ContentId	string	The text with which the IDs of the InputBox controls begin.

Return value

Type	Description
void	no return value

NuGet package: Beckhoff.TwinCAT.HMI.Plastic.Functions

Available from version 12.6.0

Usage: TC3 Plastic Application HMI projects

6.2 CodeBehind

6.2.1 DisableContextMenu

The CodeBehind function DisableContextMenu disables the context menu so that no right-click works.

NuGet package: Beckhoff.TwinCAT.HMI.Plastic.Functions

Available from version 12.6.0

Usage: TC3 Plastic Application HMI projects

6.2.2 SlidingFunction

The CodeBehind function `SlidingFunction` enables the sliding of contents that are located in a `TcHmiRegion` with the ID `Portrait_SliderContent_Region`. A `TcHmiGrid` control is required on this content, where the ID is composed of the ID of the content and the ending `_Grid`. The function changes the `Left` property of the `TcHmiGrid` control to 0, -1024 or -2048 depending on the sliding movement. Thus a sliding function for up to three subpages with a width of 1024px each can be realized. If three subpages are desired, the `TcHmiGrid` control must have a width of 3052px accordingly.

NuGet package: Beckhoff.TwinCAT.HMI.Plastic.Functions

Available: from version 12.8.0

Usage: TC3 Plastic Application HMI projects

6.3 Frame

6.3.1 HideSliderArea

```
function HideSliderArea(SliderContentRegion: TcHmi.Controls.System.TcHmiRegion, MainGrid:
TcHmi.Controls.System.TcHmiGrid, SliderTabContainer: TcHmi.Controls.System.TcHmiContainerControl):
void
```

The function `HideSliderArea` closes the slider area of a `.view` file if it was previously displayed.

Parameter

Name	Type	Description
SliderContentRegion	TcHmi.Controls.System.TcHmiRegion	The <code>TcHmiRegion</code> control, which serves as a slider area.
MainGrid	TcHmi.Controls.System.TcHmiGrid	The <code>TcHmiGrid</code> control, which contains the main region and the slider region.
SliderTabContainer	TcHmi.Controls.System.TcHmiContainerControl	The <code>TcHmiContainerControl</code> control that contains the tabs of the slider area that are used to switch the slider content.

Return value

Type	Description
void	no return value

NuGet package: Beckhoff.TwinCAT.HMI.Plastic.Functions

Available from version 12.6.0

Usage: TC3 Plastic Application HMI projects

6.3.2 ShowSliderArea

```
function ShowSliderArea(SliderContentRegion: TcHmi.Controls.System.TcHmiRegion, MainGrid:
TcHmi.Controls.System.TcHmiGrid, SliderTabContainer: TcHmi.Controls.System.TcHmiContainerControl):
void
```

The function `ShowSliderArea` displays the slider area of a `.view` file if it was previously closed.

Parameter

Name	Type	Description
SliderContentRegion	TcHmi.Controls.System.TcHmiRegion	The TcHmiRegion control, which serves as a slider area.
MainGrid	TcHmi.Controls.System.TcHmiGrid	The TcHmiGrid control, which contains the main region and the slider region.
SliderTabContainer	TcHmi.Controls.System.TcHmiContainer	The TcHmiContainer control that contains the tabs of the slider area that are used to switch the slider content.

Return value

Type	Description
void	no return value

NuGet package: Beckhoff.TwinCAT.HMI.Plastic.Functions

Available from version 12.6.0

Usage: TC3 Plastic Application HMI projects

6.3.3 ToggleSliderArea

```
function ToggleSliderArea(SliderContentRegion: TcHmi.Controls.System.TcHmiRegion, MainGrid: TcHmi.Controls.System.TcHmiGrid, SliderTabContainer: TcHmi.Controls.System.TcHmiContainerControl): void
```

The function ToggleSliderArea displays the slider area of a .view file if it was previously closed and closes it if it was previously displayed. The two functions [HideSliderArea \[▶ 166\]](#) and [ShowSliderArea \[▶ 166\]](#) are used for this purpose.

Parameter

Name	Type	Description
SliderContentRegion	TcHmi.Controls.System.TcHmiRegion	The TcHmiRegion control, which serves as a slider area.
MainGrid	TcHmi.Controls.System.TcHmiGrid	The TcHmiGrid control, which contains the main region and the slider region.
SliderTabContainer	TcHmi.Controls.System.TcHmiContainer	The TcHmiContainer control that contains the tabs of the slider area that are used to switch the slider content.

Return value

Type	Description
void	no return value

NuGet package: Beckhoff.TwinCAT.HMI.Plastic.Functions

Available from version 12.6.0

Usage: TC3 Plastic Application HMI projects

6.3.4 TakeScreenshot

```
function TakeScreenshot(ContentName: string): void
```

The TakeScreenShot function is used to create a screenshot. The transfer parameter is used for the specific naming of the screenshot file, but is not mandatory for the functionality.

Parameter

Name	Type	Description
ContentName	string	The path of the current TcHmiContent to be attached to the screenshot file name. Is not necessary.

Return value

Type	Description
void	no return value

NuGet package: Beckhoff.TwinCAT.HMI.Plastic.Functions

Available from version 12.6.0

Usage: TC3 Plastic Application HMI projects

6.3.5 UpdateSliderContentRegion

```
function UpdateSliderContentRegion(MainContentRegion: TcHmi.Controls.System.TcHmiRegion,
SliderContentRegion: TcHmi.Controls.System.TcHmiRegion, MainContentSpecificToggleButton:
TcHmi.Controls.Beckhoff.TcHmiToggleButton, NavigationToggleButton:
TcHmi.Controls.Beckhoff.TcHmiToggleButton): void
```

The UpdateSliderContentRegion function is used to load the appropriate TcHmiContent into the TcHmiRegion control for the slider content, depending on the displayed TcHmiContent in the TcHmiRegion control for the main content, provided the ToggleState property of the corresponding TcHmiToggleButton control (with the ID MainContentSpecificToggleButton) is set to Active. The TcHmiToggleButton control is additionally assigned an associated icon as BackgroundImage property from the NuGet package Beckhoff.TwinCAT.HMI.Plastic.Images.

Main content path	Slidercontent path	Icon path
Contents/Navigation/Axes/Blowpin/Blowpin.content	Contents/Slider/Axes/Blowpin/Blowpin_Settings.content	Beckhoff.TwinCAT.HMI.Plastic.Images/Axes/Blowpin/Blowpin.svg
Contents/Navigation/Axes/Blowpin/Blowpin_Homing.content	Contents/Slider/Axes/Blowpin/Blowpin_Homing_Settings.content	Beckhoff.TwinCAT.HMI.Plastic.Images/Axes/Blowpin/BlowpinHoming.svg
Contents/Navigation/Axes/Carriage/Carriage.content	Contents/Slider/Axes/Carriage/Carriage_Settings.content	Beckhoff.TwinCAT.HMI.Plastic.Images/Axes/Carriage/Carriage.svg
Contents/Navigation/Axes/Carriage/Carriage_Homing.content	Contents/Slider/Axes/Carriage/Carriage_Homing_Settings.content	Beckhoff.TwinCAT.HMI.Plastic.Images/Axes/Carriage/CarriageHoming.svg
Contents/Navigation/Axes/Clamp/Clamp.content	Contents/Slider/Axes/Clamp/Clamp_Settings.content	Beckhoff.TwinCAT.HMI.Plastic.Images/Axes/Clamp/Clamp.svg
Contents/Navigation/Axes/Clamp/Clamp_Homing.content	Contents/Slider/Axes/Clamp/Clamp_Homing_Settings.content	Beckhoff.TwinCAT.HMI.Plastic.Images/Axes/Clamp/ClampHoming.svg
Contents/Navigation/Extruder/Trend.content	Contents/Slider/Extruder/Extruder_Settings.content	Beckhoff.TwinCAT.HMI.Plastic.Images/Extruder/Extruder.svg
Contents/Navigation/Parameters/Parameters_Blowpin.content	Contents/Slider/Parameters/Parameters_Settings.content	Beckhoff.TwinCAT.HMI.Plastic.Images/Parameters/ParametersBlowpin.svg
Contents/Navigation/Parameters/Parameters_Carriage.content	Contents/Slider/Parameters/Parameters_Settings.content	Beckhoff.TwinCAT.HMI.Plastic.Images/Parameters/ParametersCarriage.svg
Contents/Navigation/Parameters/Parameters_Clamp.content	Contents/Slider/Parameters/Parameters_Settings.content	Beckhoff.TwinCAT.HMI.Plastic.Images/Parameters/ParametersClamp.svg
Contents/Navigation/Parameters/Parameters_CoExtruder.content	Contents/Slider/Parameters/Parameters_Settings.content	Beckhoff.TwinCAT.HMI.Plastic.Images/Parameters/ParametersCoExtruder.svg
Contents/Navigation/Parameters/Parameters_MainExtruder.content	Contents/Slider/Parameters/Parameters_Settings.content	Beckhoff.TwinCAT.HMI.Plastic.Images/Parameters/ParametersMainExtruder.svg
Contents/Navigation/Parameters/Parameters_Monitoring.content	Contents/Slider/Parameters/Parameters_Settings.content	Beckhoff.TwinCAT.HMI.Plastic.Images/Parameters/ParametersMonitoring.svg
Contents/Navigation/Parameters/Parameters_Setpoints.content	Contents/Slider/Parameters/Parameters_Settings.content	Beckhoff.TwinCAT.HMI.Plastic.Images/Parameters/ParametersSetpoints.svg
Contents/Navigation/Parameters/Parameters_Temperature.content	Contents/Slider/Parameters/Parameters_Settings.content	Beckhoff.TwinCAT.HMI.Plastic.Images/Parameters/ParametersTemperature.svg
Contents/Navigation/Parameters/Parameters_Wtc.content	Contents/Slider/Parameters/Parameters_Settings.content	Beckhoff.TwinCAT.HMI.Plastic.Images/Parameters/ParametersWtc.svg
Contents/Navigation/System/RecipeManagement.content	Contents/Slider/System/RecipeManagement_Settings.content	Beckhoff.TwinCAT.HMI.Plastic.Images/Parameters/ParametersRecipeManagement.svg
Contents/Navigation/Temperature/Temperature_Parameters.content	Contents/Slider/Temperature/Temperature_Parameters_Settings.content	Beckhoff.TwinCAT.HMI.Plastic.Images/Parameters/ParametersTemperature.svg
Contents/Navigation/Temperature/Temperature_TimeScheduling.content	Contents/Slider/Temperature/Temperature_TimeScheduling_Settings.content	Beckhoff.TwinCAT.HMI.Plastic.Images/Temperature/TemperatureTimeScheduling.svg
Contents/Navigation/Temperature/Temperature_Zones.content	Contents/Slider/Temperature/Temperature_Zones_Settings.content	Beckhoff.TwinCAT.HMI.Plastic.Images/Temperature/TemperatureTimeScheduling.svg

Main content path	Slidercontent path	Icon path
Contents/Navigation/Wtc/Wtc.content	Contents/Slider/Wtc/Wtc_Settings.content	Beckhoff.TwinCAT.HMI.Plastic.Images/Wtc/Wtc.svg
Contents/Navigation/Wtc/Wtc_Homing.content	Contents/Slider/Wtc/Wtc_Homing_Settings.content	Beckhoff.TwinCAT.HMI.Plastic.Images/Wtc/WtcHoming.svg

Parameter

Name	Type	Description
MainContentRegion	TcHmi.Controls.System.TcHmiRegion	The TcHmiRegion control where the main content is displayed.
SliderContentRegion	TcHmi.Controls.System.TcHmiRegion	The TcHmiRegion control in which the associated slider content of the main content is to be displayed.
MainContentSpecificToggleButton	TcHmi.Controls.Beckhoff.TcHmiToggleButton	The TcHmiToggleButton control, which is used to display the slider content in the slider area.
NavigationToggleButton	TcHmi.Controls.Beckhoff.TcHmiToggleButton	The TcHmiToggleButton control, which is used for displaying the navigation page in the slider area.

Return value

Type	Description
void	no return value

NuGet package: Beckhoff.TwinCAT.HMI.Plastic.Functions

Available from version 12.6.0

Usage: TC3 Plastic Application HMI projects

6.3.6 UpdateSliderContentRegionEx

```
function UpdateSliderContentRegionEx(MainContentRegion: TcHmi.Controls.System.TcHmiRegion,
SliderContentRegion: TcHmi.Controls.System.TcHmiRegion, MainContentSpecificToggleButton:
TcHmi.Controls.Beckhoff.TcHmiToggleButton, NavigationToggleButton:
TcHmi.Controls.Beckhoff.TcHmiToggleButton): void
```

The function `UpdateSliderContentRegionEx` is used to update the slider content using the set scheme of the `NavigationConfig` property of the `Configurator` control. Depending on the displayed `TcHmiContent` in the `TcHmiRegion` control for the main content, the appropriate `TcHmiContent` is loaded into the `TcHmiRegion` control for the slider content, provided the `ToggleState` property of the associated `TcHmiToggleButton` control (with the ID `MainContentSpecificToggleButton`) is set to `Active`. In addition, an icon is assigned to the `TcHmiToggleButton` control as `BackgroundImage` property, which is also configured via the `NavigationConfig` property.

Parameter

Name	Type	Description
MainContentRegion	TcHmi.Controls.System.TcHmiRegion	The TcHmiRegion control where the main content is displayed.
SliderContentRegion	TcHmi.Controls.System.TcHmiRegion	The TcHmiRegion control in which the associated slider content of the main content is to be displayed.
MainContentSpecificToggleButton	TcHmi.Controls.Beckhoff.TcHmiToggleButton	The TcHmiToggleButton control, which is used to display the slider content in the slider area.
NavigationToggleButton	TcHmi.Controls.Beckhoff.TcHmiToggleButton	The TcHmiToggleButton control, which is used to display the navigation page in the slider area.

Return value

Type	Description
void	No return value.

NuGet package: Beckhoff.TwinCAT.HMI.Plastic.Functions

Available: from version 12.8.0

Usage: TC3 Plastic Application HMI projects

6.4 General

6.4.1 CreateBinding

```
function CreateBinding(expression: string, propertyName: string, control: TcHmi.Controls.System.TcHmiControl): void
```

The CreateBinding function creates a binding between a symbol and a property of a control.

Parameter

Name	Type	Description
expression	string	The symbol expression with which a binding is to be created.
propertyName	string	The name of the property of the control with which a binding is to be created.
control	TcHmi.Controls.System.TcHmiControl	The control with which a binding is to be created.

Return value

Type	Description
void	No return value.

NuGet package: Beckhoff.TwinCAT.HMI.Plastic.Functions

Available: from version 12.8.0

Usage: TC3 Plastic Application HMI projects

6.4.2 SetSymbolArrayIdx

```
function SetSymbolArrayIdx(Symbol: symbol, Idx: number, TargetArray: number): void
```

The SetSymbolArrayIdx function replaces the zero at location [0] in any symbol expression with a desired Idx.

Parameter

Name	Type	Description
Symbol	symbol	Symbol expression to be changed.
Idx	number	The index to be inserted into the symbol expression.
TargetArray	number	The number of [0] digits viewed from the left to the digit to be replaced.

Return value

Type	Description
string	The symbol expression with the replaced index.

NuGet package: Beckhoff.TwinCAT.HMI.Plastic.Functions

Available: from version 12.8.0

Usage: TC3 Plastic Application HMI projects

6.5 Parameters

6.5.1 UpdateParametersSettingsControls

```
function UpdateParametersSettingsControls(): void
```

The function UpdateParametersSettingsControls assigns different symbols to the following controls depending on the TargetContent property of the TcHmiRegion control with the ID Portrait_MainContent_Region.

Control ID	Control type	Property
Parameters_Settings_Load_Button	TcHmi.Controls.Beckhoff.TcHmiButton	StateSymbol
Parameters_Settings_Save_Button	TcHmi.Controls.Beckhoff.TcHmiButton	StateSymbol
Parameters_Settings_CreationDateValue_Textblock	TcHmi.Controls.Beckhoff.TcHmiTextblock	Text
Parameters_Settings_StorageCountValue_Textblock	TcHmi.Controls.Beckhoff.TcHmiTextblock	Text
Parameters_Settings_StorageDateValue_Textblock	TcHmi.Controls.Beckhoff.TcHmiTextblock	Text
Parameters_Settings_Load_StateIndicator	TcHmi.Controls.Plastic.StateIndicator	IndicatorStatus
Parameters_Settings_Save_StateIndicator	TcHmi.Controls.Plastic.StateIndicator	IndicatorStatus

Contents/Navigation/Parameters/Parameters_Temperature.content

Control ID	PLC Symbol
Parameters_Settings_Load_Button	PLC1.Temperature.fbTempCtrlHmi::fbParamHandle::Load
Parameters_Settings_Save_Button	PLC1.Temperature.fbTempCtrlHmi::fbParamHandle::Save
Parameters_Settings_CreationDateValue_Textblock	PLC1.Temperature.fbTempCtrlHmi::fbParamHandle::FileCreationDate
Parameters_Settings_StoreCountValue_Textblock	PLC1.Temperature.fbTempCtrlHmi::fbParamHandle::FileStoreCount
Parameters_Settings_StoredDateValue_Textblock	PLC1.Temperature.fbTempCtrlHmi::fbParamHandle::FileStoreDate
Parameters_Settings_Load_StateIndicator	PLC1.Temperature.fbTempCtrlHmi::fbParamHandle::LoadingStatus
Parameters_Settings_Save_StateIndicator	PLC1.Temperature.fbTempCtrlHmi::fbParamHandle::SavingStatus

Contents/Navigation/Parameters/Parameters_MainExtruder.content

Control ID	PLC Symbol
Parameters_Settings_Load_Button	PLC1.Extruder.fbMainExtruderHmi::fbParamHandle::Load
Parameters_Settings_Save_Button	PLC1.Extruder.fbMainExtruderHmi::fbParamHandle::Save
Parameters_Settings_CreationDateValue_Textblock	PLC1.Extruder.fbMainExtruderHmi::fbParamHandle::FileCreationDate
Parameters_Settings_StoreCountValue_Textblock	PLC1.Extruder.fbMainExtruderHmi::fbParamHandle::FileStoreCount
Parameters_Settings_StoredDateValue_Textblock	PLC1.Extruder.fbMainExtruderHmi::fbParamHandle::FileStoreDate
Parameters_Settings_Load_StateIndicator	PLC1.Extruder.fbMainExtruderHmi::fbParamHandle::LoadingStatus
Parameters_Settings_Save_StateIndicator	PLC1.Extruder.fbMainExtruderHmi::fbParamHandle::SavingStatus

Contents/Navigation/Parameters/Parameters_CoExtruder.content

Control ID	PLC Symbol
Parameters_Settings_Load_Button	PLC1.Extruder.fbCoExtruderHmi::fbParamHandle::Load
Parameters_Settings_Save_Button	PLC1.Extruder.fbCoExtruderHmi::fbParamHandle::Save
Parameters_Settings_CreationDateValue_Textblock	PLC1.Extruder.fbCoExtruderHmi::fbParamHandle::FileCreationDate
Parameters_Settings_StoreCountValue_Textblock	PLC1.Extruder.fbCoExtruderHmi::fbParamHandle::FileStoreCount
Parameters_Settings_StoredDateValue_Textblock	PLC1.Extruder.fbCoExtruderHmi::fbParamHandle::FileStoreDate
Parameters_Settings_Load_StateIndicator	PLC1.Extruder.fbCoExtruderHmi::fbParamHandle::LoadingStatus
Parameters_Settings_Save_StateIndicator	PLC1.Extruder.fbCoExtruderHmi::fbParamHandle::SavingStatus

Contents/Navigation/Parameters/Parameters_Blowpin.content

Control ID	PLC Symbol
Parameters_Settings_Load_Button	PLC1.Blowpin.fbBlowpinHmi::fbParamHandle::Load
Parameters_Settings_Save_Button	PLC1.Blowpin.fbBlowpinHmi::fbParamHandle::Save
Parameters_Settings_CreationDateValue_Textblock	PLC1.Blowpin.fbBlowpinHmi::fbParamHandle::FileCr eationDate
Parameters_Settings_StoreCountValue_Textblock	PLC1.Blowpin.fbBlowpinHmi::fbParamHandle::FileSt oreCount
Parameters_Settings_StoredDateValue_Textblock	PLC1.Blowpin.fbBlowpinHmi::fbParamHandle::FileSt oreDate
Parameters_Settings_Load_StateIndicator	PLC1.Blowpin.fbBlowpinHmi::fbParamHandle::Loadin gStatus
Parameters_Settings_Save_StateIndicator	PLC1.Blowpin.fbBlowpinHmi::fbParamHandle::Savin gStatus

Contents/Navigation/Parameters/Parameters_Carriage.content

Control ID	PLC Symbol
Parameters_Settings_Load_Button	PLC1.Carriage.fbCarriageHmi::fbParamHandle::Load
Parameters_Settings_Save_Button	PLC1.Carriage.fbCarriageHmi::fbParamHandle::Save
Parameters_Settings_CreationDateValue_Textblock	PLC1.Carriage.fbCarriageHmi::fbParamHandle::FileC reationDate
Parameters_Settings_StoreCountValue_Textblock	PLC1.Carriage.fbCarriageHmi::fbParamHandle::FileS toreCount
Parameters_Settings_StoredDateValue_Textblock	PLC1.Carriage.fbCarriageHmi::fbParamHandle::FileS toreDate
Parameters_Settings_Load_StateIndicator	PLC1.Carriage.fbCarriageHmi::fbParamHandle::Load ingStatus
Parameters_Settings_Save_StateIndicator	PLC1.Carriage.fbCarriageHmi::fbParamHandle::Savi ngStatus

Contents/Navigation/Parameters/Parameters_Clamp.content

Control ID	PLC Symbol
Parameters_Settings_Load_Button	PLC1.Clamp.fbClampHmi::fbParamHandle::Load
Parameters_Settings_Save_Button	PLC1.Clamp.fbClampHmi::fbParamHandle::Save
Parameters_Settings_CreationDateValue_Textblock	PLC1.Clamp.fbClampHmi::fbParamHandle::FileCreat ionDate
Parameters_Settings_StoreCountValue_Textblock	PLC1.Clamp.fbClampHmi::fbParamHandle::FileStore Count
Parameters_Settings_StoredDateValue_Textblock	PLC1.Clamp.fbClampHmi::fbParamHandle::FileStore Date
Parameters_Settings_Load_StateIndicator	PLC1.Clamp.fbClampHmi::fbParamHandle::LoadingS tatus
Parameters_Settings_Save_StateIndicator	PLC1.Clamp.fbClampHmi::fbParamHandle::SavingSt atus

Contents/Navigation/Parameters/Parameters_Wtc.content

Control ID	PLC Symbol
Parameters_Settings_Load_Button	PLC1.Wtc.fbWtcHmi::fbParamHandle::Load
Parameters_Settings_Save_Button	PLC1.Wtc.fbWtcHmi::fbParamHandle::Save
Parameters_Settings_CreationDateValue_Textblock	PLC1.Wtc.fbWtcHmi::fbParamHandle::FileCreationDate
Parameters_Settings_StoreCountValue_Textblock	PLC1.Wtc.fbWtcHmi::fbParamHandle::FileStoreCount
Parameters_Settings_StoredDateValue_Textblock	PLC1.Wtc.fbWtcHmi::fbParamHandle::FileStoreDate
Parameters_Settings_Load_StateIndicator	PLC1.Wtc.fbWtcHmi::fbParamHandle::LoadingStatus
Parameters_Settings_Save_StateIndicator	PLC1.Wtc.fbWtcHmi::fbParamHandle::SavingStatus

Contents/Navigation/Parameters/Parameters_Monitoring.content

Control ID	PLC Symbol
Parameters_Settings_Load_Button	PLC1.Monitoring::fbParamHandle::Load
Parameters_Settings_Save_Button	PLC1.Monitoring::fbParamHandle::Save
Parameters_Settings_CreationDateValue_Textblock	PLC1.Monitoring::fbParamHandle::FileCreationDate
Parameters_Settings_StoreCountValue_Textblock	PLC1.Monitoring::fbParamHandle::FileStoreCount
Parameters_Settings_StoredDateValue_Textblock	PLC1.Monitoring::fbParamHandle::FileStoreDate
Parameters_Settings_Load_StateIndicator	PLC1.Monitoring::fbParamHandle::LoadingStatus
Parameters_Settings_Save_StateIndicator	PLC1.Monitoring::fbParamHandle::SavingStatus

Contents/Navigation/Parameters/Parameters_Setpoints.content

Control ID	PLC Symbol
Parameters_Settings_Load_Button	PLC1.Setpoints::fbParamHandle::Load
Parameters_Settings_Save_Button	PLC1.Setpoints::fbParamHandle::Save
Parameters_Settings_CreationDateValue_Textblock	PLC1.Setpoints::fbParamHandle::FileCreationDate
Parameters_Settings_StoreCountValue_Textblock	PLC1.Setpoints::fbParamHandle::FileStoreCount
Parameters_Settings_StoredDateValue_Textblock	PLC1.Setpoints::fbParamHandle::FileStoreDate
Parameters_Settings_Load_StateIndicator	PLC1.Setpoints::fbParamHandle::LoadingStatus
Parameters_Settings_Save_StateIndicator	PLC1.Setpoints::fbParamHandle::SavingStatus

Parameter

Name	Type	Description
-	-	-

Return value

Type	Description
void	no return value

NuGet package: Beckhoff.TwinCAT.HMI.Plastic.Functions

Available from version 12.6.0

Usage: TC3 Plastic Application HMI projects

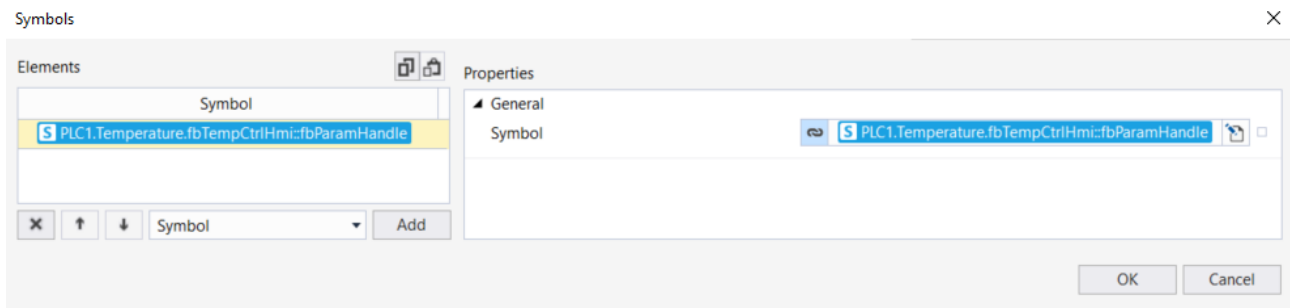
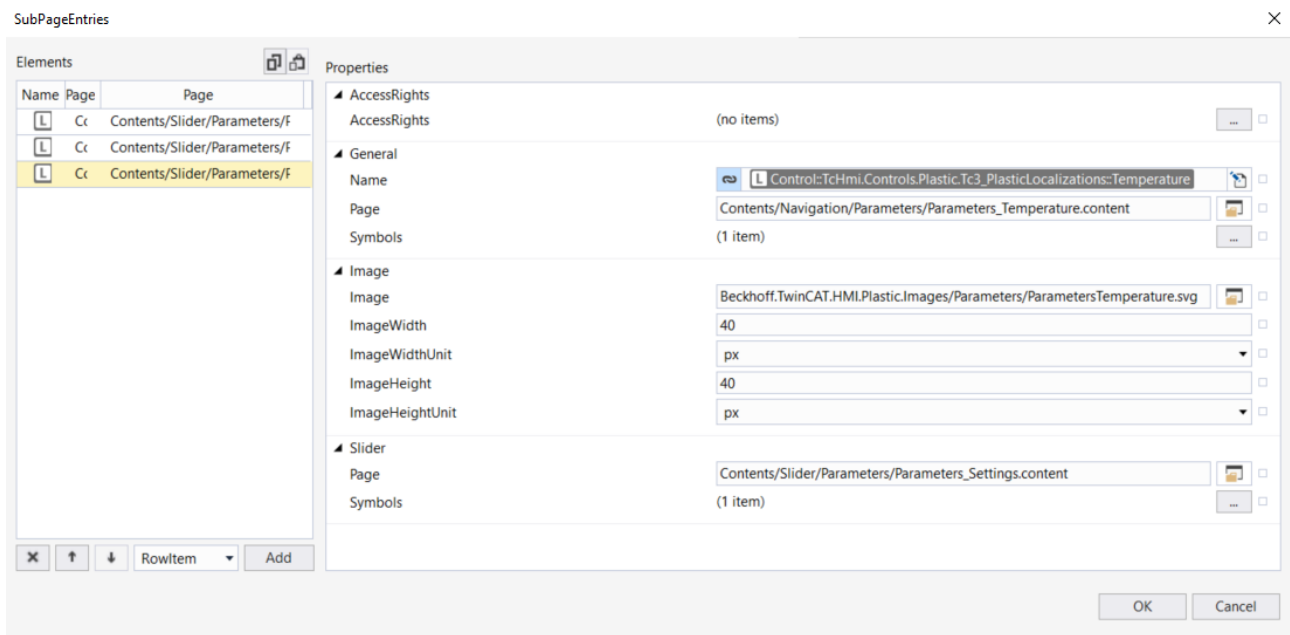
6.5.2 UpdateParametersSettingsControlsEx

```
function UpdateParametersSettingsControlsEx(): void
```

The function UpdateParametersSettingsControlsEx is used to assign the symbols of the set symbol of type FB_ParamHandle from the library Tc3_PlasticBaseApplication to the listed controls with the help of the set scheme of the NavigationConfig property of the Configurator control, depending on the TargetContent property of the TcHmiRegion control with the ID Portrait_MainContent_Region.

Control ID	Control type	Property
Parameters_Settings_Load_Button	TcHmi.Controls.Beckhoff.TcHmiButton	StateSymbol
Parameters_Settings_Save_Button	TcHmi.Controls.Beckhoff.TcHmiButton	StateSymbol
Parameters_Settings_CreationDateValue_Textblock	TcHmi.Controls.Beckhoff.TcHmiTextblock	Text
Parameters_Settings_StoreCountValue_Textblock	TcHmi.Controls.Beckhoff.TcHmiTextblock	Text
Parameters_Settings_StoredDateValue_Textblock	TcHmi.Controls.Beckhoff.TcHmiTextblock	Text
Parameters_Settings_Load_StateIndicator	TcHmi.Controls.Plastic.StateIndicator	IndicatorStatus
Parameters_Settings_Save_StateIndicator	TcHmi.Controls.Plastic.StateIndicator	IndicatorStatus

Example setting of the NavigationConfig property:



Parameter

Name	Type	Description
-	-	-

Return value

Type	Description
void	No return value.

NuGet package: Beckhoff.TwinCAT.HMI.Plastic.Functions

Available: from version 12.8.0

Usage: TC3 Plastic Application HMI projects

6.5.3 UpdateTemperatureParametersTableNumber

```
function UpdateTemperatureParametersTableNumber(): void
```

The function UpdateTemperatureParametersTableNumber adjusts the number of zones in the table control with the ID Parameters_Temperature_Table if the control is located in the TcHmiContent with path Contents/Navigation/Parameters/Parameters_Temperature.content and this TcHmiContent is loaded in the TcHmiRegion control with ID Portrait_MainContent_Region. The number of zones is determined by the symbol PLC1.Temperature.fbTempCtrlHmi::CountPfwZones.

Parameter

Name	Type	Description
-	-	-

Return value

Type	Description
void	no return value

NuGet package: Beckhoff.TwinCAT.HMI.Plastic.Functions

Available from version 12.6.0

Usage: TC3 Plastic Application HMI projects

6.5.4 UpdateTemperatureParametersTableNumberEx

```
function UpdateTemperatureParametersTableNumberEx(): void
```

The function UpdateTemperatureParametersTableNumberEx adjusts the number of temperature channels in the table control with the ID Parameters_Temperature_Table using the NavigationConfig property of the Configurator control if the control is located in the TcHmiContent with the path Contents/Navigation/Parameters/Parameters_Temperature.content and this TcHmiContent is loaded in the TcHmiRegion control with the ID Portrait_MainContent_Region. To do this, a symbol of type FB_TemperatureHmi must be passed to the NavigationConfig property of the Configurator control for the corresponding page of the Symbols property of the General category.

Parameter

Name	Type	Description
-	-	-

Return value

Type	Description
void	No return value.

NuGet package: Beckhoff.TwinCAT.HMI.Plastic.Functions

Available: from version 12.8.0

Usage: TC3 Plastic Application HMI projects

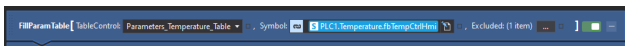
6.5.5 FillParamTable

```
function FillParamTable(TableControl, Symbol, Excluded): void
```

The function FillParamTable fills a table of the type `TcHmi.Controls.Plastic.Table` [▶ 112] with the machine data of a PLC symbol. All the structured subsymbols whose instance name starts with "fbParams" are taken into account. For each found instance of a fbParams, a subcategory is defined and the available subsymbols are inserted.

In addition to the inserted parameters, in the `TcHmi.Controls.Plastic.Table` [▶ 112] further own parameters can be defined under the attribute FirstTable. These are inserted after the generated table entries.

Sample call:



Sample result:

Parameter

Name	Type	Description
TableControl	<code>TcHmi.Controls.Plastic.Table</code> [▶ 112]	Table to be filled
Symbol	<code>TcHmi.Symbol</code>	Symbol to be used for filling the table
Excluded	<code>PlasticSymbols</code>	Subsymbols to be excluded when filling the table

Return value

Type	Description
void	no return value

NuGet package: Beckhoff.TwinCAT.HMI.Plastic.Functions

Available: from version 12.8.0

Usage: TC3 Plastic Application HMI projects

6.6 Scope

6.6.1 UpdateScope

```
function UpdateScope(ToggleButtonControl: TcHmi.Controls.Beckhoff.TcHmiToggleButton, AxisPath: string): boolean
```

The function UpdateScope updates a `TcHmiTrendLineChart` control with the ID `Scope_TrendLineChart` and displays axis-specific values depending on the axis type. The axis type is determined via the `AxisPath` property. The `AxisPath` symbol is checked with the `::fbSpecific::fbNc::Valid` extension. If this value is true, then this specifies that the axis is of type NC. The `TcHmiStateImage` control with the ID `Scope_StateImage` gets the ID of the `TcHmiToggleButton` control passed as parameter in this function for the `State` property. This allows a suitable image to be stored in the `TcHmiStateImage` control for each axis to be displayed, which is then switched over appropriately.

Parameter

Name	Type	Description
ToggleButtonControl	TcHmi.Controls.Beckhoff.TcHmiToggleButton	The TcHmiToggleButton control, which is used for switching the axes.
AxisPath	string	The axis path of the axis to be displayed.

Return value

Type	Description
boolean	Values: <ul style="list-style-type: none"> • true: The axis is of type NC. • false: The axis is not of type NC.

NuGet package: Beckhoff.TwinCAT.HMI.Plastic.Functions

Available from version 12.6.0

Usage: TC3 Plastic Application HMI projects

6.7 Trend

6.7.1 ResetTrendAxisColors

```
function ResetTrendAxisColors(): void
```

The function ResetTrendAxisColors resets the controls listed in the table. The TcHmiInput controls are assigned a hexadecimal color code and the corresponding TcHmiCheckbox controls are assigned the appropriate background color.

TcHmi.Controls.Beckhoff.TcHmi-Input Control ID	TcHmi.Controls.Beckhoff.TcHmi-Checkbox Control ID	Hexadecimal color code
Trend_Settings_1_Input	Trend_Settings_1_Rectangle	#E6194B
Trend_Settings_2_Input	Trend_Settings_2_Rectangle	#F58231
Trend_Settings_3_Input	Trend_Settings_3_Rectangle	#FFE119
Trend_Settings_4_Input	Trend_Settings_4_Rectangle	#BFEF45
Trend_Settings_5_Input	Trend_Settings_5_Rectangle	#3CB44B
Trend_Settings_6_Input	Trend_Settings_6_Rectangle	#469990
Trend_Settings_7_Input	Trend_Settings_7_Rectangle	#42D4F4
Trend_Settings_8_Input	Trend_Settings_8_Rectangle	#4363D8
Trend_Settings_9_Input	Trend_Settings_9_Rectangle	#911EB4
Trend_Settings_10_Input	Trend_Settings_10_Rectangle	#F032E6
Trend_Settings_11_Input	Trend_Settings_11_Rectangle	#469990
Trend_Settings_12_Input	Trend_Settings_12_Rectangle	#42D4F4

Parameter

Name	Type	Description
-	-	-

Return value

Type	Description
void	no return value

NuGet package: Beckhoff.TwinCAT.HMI.Plastic.Functions

Available from version 12.6.0

Usage: TC3 Plastic Application HMI projects

6.7.2 UpdateTrend

```
function UpdateTrend(FB_TrendHmi: TcHmi.Symbol): void
```

The function UpdateTrend updates a TcHmiTrendLineChart control with the ID `Trend_TrendLineChart` and displays values. The TcHmiRegion control with the ID `Portrait_MainContent_Region` is used to check whether the `TargetContent` property has the value `Contents/Home.content` or not, since the content on which the TcHmiTrendLineChart control exists can also be reused on the `Contents/Home.content` by another TcHmiRegion control. By using the function [UpdateTrendAxisColors \[► 180\]](#) the colors of the curves can be adjusted.

Parameter

Name	Type	Description
FB_TrendHmi	Symbol	The symbol of the FB of type <code>FB_TrendHmi</code> from the <code>Tc3_PlasticBaseApplication</code> library.

Return value

Type	Description
void	no return value

NuGet package: Beckhoff.TwinCAT.HMI.Plastic.Functions

Available from version 12.6.0

Usage: TC3 Plastic Application HMI projects

6.7.3 UpdateTrendAxisColors

```
function UpdateTrendAxisColors(): void
```

The function UpdateTrendAxisColors reads the `Text` property of the TcHmiInput control. If the read `Text` property is a string for describing colors in hexadecimal format (`#RRGGBB`), then this value is used to adjust the `BackgroundColor` property of the associated TcHmiCheckbox control to visually represent the color. In addition, the color value is stored in an array that serves as the return value of this function. The return value is used, for example, in the function [UpdateTrend \[► 180\]](#) to be able to adjust the colors of the curves. The following controls are considered by the function:

TcHmi.Controls.Beckhoff.TcHmiInput Control ID	TcHmi.Controls.Beckhoff.TcHmiCheckbox Control ID
Trend_Settings_1_Input	Trend_Settings_1_Rectangle
Trend_Settings_2_Input	Trend_Settings_2_Rectangle
Trend_Settings_3_Input	Trend_Settings_3_Rectangle
Trend_Settings_4_Input	Trend_Settings_4_Rectangle
Trend_Settings_5_Input	Trend_Settings_5_Rectangle
Trend_Settings_6_Input	Trend_Settings_6_Rectangle
Trend_Settings_7_Input	Trend_Settings_7_Rectangle
Trend_Settings_8_Input	Trend_Settings_8_Rectangle
Trend_Settings_9_Input	Trend_Settings_9_Rectangle
Trend_Settings_10_Input	Trend_Settings_10_Rectangle
Trend_Settings_11_Input	Trend_Settings_11_Rectangle
Trend_Settings_12_Input	Trend_Settings_12_Rectangle

Parameter

Name	Type	Description
-	-	-

Return value

Type	Description
string[]	Array with the colors in the syntax that can be used by a TcHmiTrendLineChart control.

NuGet package: Beckhoff.TwinCAT.HMI.Plastic.Functions

Available from version 12.6.0

Usage: TC3 Plastic Application HMI projects

7 Images

NuGet package: Beckhoff.TwinCAT.HMI.Plastic.Images

Available: from version 12.6.0

The NuGet package **Beckhoff.TwinCAT.HMI.Plastic.Images** can be installed in the TwinCAT HMI project via the NuGet package management system.

To use the icons from the package, the text must be inserted into the property of a control (e.g. BackgroundImage) according to the following scheme:

Beckhoff.TwinCAT.HMI.Plastic.Images/Folder/Icon.svg

In this example, the string references the icon named *Icon.svg*, which is located inside the package in the folder named "Folder".

Icon references

- Beckhoff.TwinCAT.HMI.Plastic.Images/Axes/Axes.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Axes/Scope.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Axes/Blowpin/Blowpin.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Axes/Blowpin/BlowpinDown.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Axes/Blowpin/BlowpinHoming.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Axes/Blowpin/BlowpinUp.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Axes/Blowpin/CoolingBlowpin.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Axes/Blowpin/CoolingBlowpinDown.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Axes/Blowpin/CoolingBlowpinUp.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Axes/Blowpin/SideBlowpin.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Axes/Blowpin/SideBlowpinBlow.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Axes/Carriage/Carriage.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Axes/Carriage/CarriageHoming.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Axes/Carriage/CarriageLeft.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Axes/Carriage/CarriageRight.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Axes/Clamp/Clamp.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Axes/Clamp/ClampClose.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Axes/Clamp/ClampHoming.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Axes/Clamp/ClampLeft.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Axes/Clamp/ClampOpen.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Axes/Clamp/ClampRight.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Extruder/CoExtruder.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Extruder/CoExtruderVelocityDown.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Extruder/CoExtruderVelocityUp.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Extruder/ExtruderDown.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Extruder/ExtruderHead.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Extruder/ExtruderHeadWhite.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Extruder/ExtruderUp.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Extruder/MainExtruder.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Extruder/MainExtruderVelocityDown.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Extruder/MainExtruderVelocityUp.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Extruder/Trend.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Extruder/ViewStripeExtruder.svg

- Beckhoff.TwinCAT.HMI.Plastic.Images/Extruder/ViewStripeExtruderVelocityDown.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Extruder/ViewStripeExtruderVelocityUp.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Frame/AdditionalPages.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Frame/Beckhoff_Logo.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Frame/Home.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Frame/Info.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Frame/Left.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Frame/ManualFunctions.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Frame/Navigation.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Frame/Pause.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Frame/Right.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Frame/Screenshot.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Frame/SliderDown.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Frame/SliderUp.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Frame/Start.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Frame/Stop.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Frame/ToBasePosition.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Frame/User.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Frame/Flags/br.png
- Beckhoff.TwinCAT.HMI.Plastic.Images/Frame/Flags/can.png
- Beckhoff.TwinCAT.HMI.Plastic.Images/Frame/Flags/cn.png
- Beckhoff.TwinCAT.HMI.Plastic.Images/Frame/Flags/de.png
- Beckhoff.TwinCAT.HMI.Plastic.Images/Frame/Flags/en_gb.png
- Beckhoff.TwinCAT.HMI.Plastic.Images/Frame/Flags/en_us.png
- Beckhoff.TwinCAT.HMI.Plastic.Images/Frame/Flags/eng.png
- Beckhoff.TwinCAT.HMI.Plastic.Images/Frame/Flags/es.png
- Beckhoff.TwinCAT.HMI.Plastic.Images/Frame/Flags/fr.png
- Beckhoff.TwinCAT.HMI.Plastic.Images/Frame/Flags/in.png
- Beckhoff.TwinCAT.HMI.Plastic.Images/Frame/Flags/is.png
- Beckhoff.TwinCAT.HMI.Plastic.Images/Frame/Flags/it.png
- Beckhoff.TwinCAT.HMI.Plastic.Images/Frame/Flags/jp.png
- Beckhoff.TwinCAT.HMI.Plastic.Images/Frame/Flags/nd.png
- Beckhoff.TwinCAT.HMI.Plastic.Images/Frame/Flags/pk.png
- Beckhoff.TwinCAT.HMI.Plastic.Images/Frame/Flags/pr.png
- Beckhoff.TwinCAT.HMI.Plastic.Images/Frame/Flags/ru.png
- Beckhoff.TwinCAT.HMI.Plastic.Images/Frame/Flags/sa.png
- Beckhoff.TwinCAT.HMI.Plastic.Images/Frame/Flags/tr.png
- Beckhoff.TwinCAT.HMI.Plastic.Images/Frame/Flags/uk.png
- Beckhoff.TwinCAT.HMI.Plastic.Images/Frame/Flags/us.png
- Beckhoff.TwinCAT.HMI.Plastic.Images/MachineRenderings/BlowMolding.png
- Beckhoff.TwinCAT.HMI.Plastic.Images/MachineRenderings/BlowMoldingLine.png
- Beckhoff.TwinCAT.HMI.Plastic.Images/MachineRenderings/CoExtruder.png
- Beckhoff.TwinCAT.HMI.Plastic.Images/MachineRenderings/ExtKopf.png
- Beckhoff.TwinCAT.HMI.Plastic.Images/MachineRenderings/Extruder.png
- Beckhoff.TwinCAT.HMI.Plastic.Images/MachineRenderings/ExtruderWith13Zones.png
- Beckhoff.TwinCAT.HMI.Plastic.Images/MachineRenderings/ExtruderWith5Zones.png

- Beckhoff.TwinCAT.HMI.Plastic.Images/MachineRenderings/Flansch.png
- Beckhoff.TwinCAT.HMI.Plastic.Images/Parameters/Parameters.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Parameters/ParametersBlowpin.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Parameters/ParametersCarriage.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Parameters/ParametersClamp.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Parameters/ParametersCoExtruder.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Parameters/ParametersMainExtruder.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Parameters/ParametersMonitoring.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Parameters/ParametersSetpoints.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Parameters/ParametersTemperature.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Parameters/ParametersWtc.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Process/Blowing.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Process/EnergyMeasurement.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Process/Monitoring.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Process/Process.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Process/Setpoints.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Process/Timer.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/System/Administration.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/System/AlarmColumns.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/System/AlarmHistory.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/System/Deselect.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/System/Events.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/System/EventsAlarms.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/System/Filter.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/System/Recipe.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/System/RecipeManagement.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/System/System.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Temperature/Cooling.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Temperature/Heating.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Temperature/Temperature.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Temperature/TemperatureControlCabinet.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Temperature/TemperatureOff.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Temperature/TemperatureOn.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Temperature/TemperatureParameters.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Temperature/TemperatureStandby.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Temperature/TemperatureTimed.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Temperature/TemperatureTimeScheduling.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Temperature/TemperatureZones.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Temperature/TemperatureZonesLocalization.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Temperature/TemperatureZonesLocalizationReverse.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Various/Automatic.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Various/Cut.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Various/Drive.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Various/Info.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Various/Lamp.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Various/Plus.svg

- Beckhoff.TwinCAT.HMI.Plastic.Images/Various/SemiAutomatic.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Various/BlowMolding/CanisterHandleCutter.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Various/BlowMolding/CanisterHandleCutterBackward.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Various/BlowMolding/CanisterHandleCutterForward.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Various/BlowMolding/Corepuller.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Various/BlowMolding/CorepullerClose.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Various/BlowMolding/CorepullerOpen.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Various/BlowMolding/Cutter.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Various/BlowMolding/CutterBackward.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Various/BlowMolding/CutterForward.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Various/BlowMolding/ExpandingMandrel.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Various/BlowMolding/ExpandingMandrelOut.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Various/BlowMolding/Gripper.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Various/BlowMolding/GripperBackward.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Various/BlowMolding/GripperClose.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Various/BlowMolding/GripperForward.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Various/BlowMolding/GripperOpen.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Various/BlowMolding/PartGripper.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Various/BlowMolding/PartGripperClose.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Various/BlowMolding/PartGripperOpen.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Various/Extruder/ExtruderAdministration.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Various/Extruder/ExtruderMachineOverview.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Various/Extruder/ExtruderOverviewWhite.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Various/Hydraulic/HydraulicPump.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Various/Hydraulic/HydraulicPumpTechnical.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Various/InjectionMolding/BarrelHeat.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Various/InjectionMolding/BarrelHeat2.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Various/InjectionMolding/Carriage.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Various/InjectionMolding/Carriage2.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Various/InjectionMolding/Clamp.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Various/InjectionMolding/ClampClose.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Various/InjectionMolding/ClampCooling.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Various/InjectionMolding/ClampLock.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Various/InjectionMolding/ClampOpen.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Various/InjectionMolding/Cloud.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Various/InjectionMolding/Configuration.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Various/InjectionMolding/Corepuller.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Various/InjectionMolding/DataLoadSave.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Various/InjectionMolding/Ejector.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Various/InjectionMolding/Injection.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Various/InjectionMolding/InjectionHoldingPressure.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Various/InjectionMolding/InjectionUnit.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Various/InjectionMolding/InjectionUnitConfiguration.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Various/InjectionMolding/InjectionUnitPressure.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Various/InjectionMolding/InjectionUnitTemperature.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Various/InjectionMolding/InjectionUnitTorque.svg

- Beckhoff.TwinCAT.HMI.Plastic.Images/Various/InjectionMolding/Metering.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Various/InjectionMolding/Pressure.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Various/InjectionMolding/Pump.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Various/InjectionMolding/QualityTable.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Various/InjectionMolding/ShotScope.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Various/InjectionMolding/Temperature.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Various/InjectionMolding/TimingScope.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Various/InjectionMolding/UserManagement.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Wtc/EditorProfileBackward.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Wtc/EditorProfileForward.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Wtc/EditorProfileUndo.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Wtc/Wtc.svg
- Beckhoff.TwinCAT.HMI.Plastic.Images/Wtc/WtcHoming.svg

8 Localizations

NuGet package: Beckhoff.TwinCAT.HMI.Plastic.Localizations

Available from version 12.6.0

The NuGet package **Beckhoff.TwinCAT.HMI.Plastic.Localizations** can be installed in the TwinCAT HMI project via the NuGet package management system.

It includes an English (en-US) and a German (de-DE) language file with language symbols that may be relevant for plastic user interfaces.

To use the language symbols from the package, the following text must be inserted into the property of a control (e.g. Text):

```
"%I%Control::TcHmi.Controls.Plastic.Tc3_PlasticLocalizations::Language%/I%"
```

In this sample, the string references the language key Language. The English translation is "Language" and the German "Sprache".

To display the list of all available language symbols or to change the translations of the languages English and German or to add further languages, the language files can be imported in the TwinCAT HMI project. To do this, the following steps must be performed:

1. Right-click on the folder **Localization** in the TwinCAT HMI project and add a new item.
2. Select the **Import Localization** option in the window that opens and then click the **Add** button.
3. In the TwinCAT HMI Wizard window open the **Control** tab, select the control with the name "Tc3_PlasticLocalizations" via the checkbox and confirm it by clicking the **OK** button. (If desired, only one of the two available languages can be selected)
4. The selected language files appear in the folder "Localizations" and can be edited using the localization editor.

9 Themes

NuGet package: Beckhoff.TwinCAT.HMI.Plastic.Themes

Available from version 12.6.0

The NuGet package Beckhoff.TwinCAT.HMI.Plastic.Themes can be installed in the TwinCAT HMI project via the NuGet package management system.

It includes design adaptations of existing controls and classes that can be used in the TwinCAT HMI project.

Use of the Plastic design

To be able to use the design from the NuGet package in a TwinCAT HMI project, a theme with the name "Plastic" must first be created. To do this, perform the following steps:

1. Right-click on the folder "Themes" in the TwinCAT HMI project and add a new item.
 2. Select the **Theme** option in the window that opens, assign the name "Plastic" and then click the **Add** button.
- ⇒ The theme named "Plastic" appears in the folder "Themes" and can be edited using the theme editor.

Use of class themes

To use class themes from the NuGet package, they must be inserted in the theme editor. This requires the following steps:

1. Open the Plastic.theme file in the theme editor.
 2. Right-click on the tab **Control Classes** and select **Create new Control Class**.
 3. Now add the following control classes to be able to use them in the project: inputbox_toggle_switch; SliderTab; Tiles; TilesHeadline
- ⇒ Now the control classes can be assigned to the controls via the property **ClassNames**.



The added control classes are grayed out in the theme editor because they do not contain a definition. The definition of the previously added control classes via the theme editor is not necessary by using the Beckhoff.TwinCAT.HMI.Plastic.Themes NuGet package, because the package contains the definition.

10 Temperature

10.1 ZoneGrouping

NuGet package: Beckhoff.TwinCAT.HMI.Plastic.Temperature

Available: since version 12.8.0

ZoneGrouping control is used to visualize and allow users to change temperature group configuration in the Tc3 Plastic Application project.



1. Temperature groups list.
2. Zones added to each group are displayed inside it.
3. A list of all available zones in the project.
4. Button to start zone names editing dialog.
5. Button to start group names editing dialog.

User can see all the temperature groups, with information about zones belonging to each of the group.

Users can add and remove zones from any group.

The buttons on the control start different dialogs that can be used to change group names and zones names respectively.



Activate action

Any new changes on the control must be activated using [ActivateConfig](#) [▶ 191] action to set new values to PLC.

10.1.1 Attributes

Category: Colors

Name	Description
HeaderBackgroundColor [► 190]	Definition of the background color of the header.
HeaderFontColor [► 190]	Definition of the text color of the header.

Category: Header Configuration

Name	Description
HeaderHeight [► 190]	Height of the control's header.
HeaderHeightUnit [► 190]	Unit of the HeaderHeight attribute.
HeaderFontSize [► 191]	Font size of the header.
HeaderFontSizeUnit [► 191]	Unit for header font size.

10.1.1.1 Activation Required

Is set to TRUE if user makes new changes that are not activated.

Schema: tchmi:general#/definitions/Boolean

Attribute getter: getActivationRequired

Available: since version 12.8.0

10.1.1.2 HeaderBackgroundColor

Definition of the background color of the header.

Schema: tchmi:framework#/definitions/Color

Attribute getter: getHeaderBackgroundColor

Attribute setter: setHeaderBackgroundColor

Available: since version 12.8.0

10.1.1.3 HeaderFontColor

Definition of the text color of the header.

Schema: tchmi:framework#/definitions/Color

Attribute getter: getHeaderFontColor

Attribute setter: setHeaderFontColor

Available: since version 12.8.0

10.1.1.4 HeaderHeight

Definition of the height of the header.

Schema: tchmi:framework#/definitions/MeasurementValue

Attribute getter: getHeaderHeight

Attribute setter: setHeaderHeight

Available: since version 12.8.0

10.1.1.5 HeaderHeightUnit

Defines the unit of the height of the header:

- Pixels for an absolute size
- Percent for a relative size

Schema: tchmi:framework#/definitions/MeasurementUnit

Attribute getter: getHeaderHeightUnit

Attribute setter: setHeaderHeightUnit

Available: since version 12.8.0

10.1.1.6 HeaderFontSize

The font size. If the percent is specified as the unit, this is relative to the font size of the parent element.

Schema: tchmi:framework#/definitions/MeasurementValue

Attribute getter: getHeaderFontSize

Attribute setter: setHeaderFontSize

Available: since version 12.8.0

10.1.1.7 HeaderFontSizeUnit

Pixels, or percent for relative sizes.

Schema: tchmi:framework#/definitions/MeasurementUnit

Attribute getter: getHeaderFontSizeUnit

Attribute setter: setHeaderFontSizeUnit

Available: since version 12.8.0

10.1.2 Functions

10.1.2.1 ActivateConfig

By calling this function all user changes will be activated and written to PLC symbols.

Parameter

Name	Type	Description
-	-	-

Return value

Type	Description
void	No return value.

Available: since version 12.8.0

10.1.2.2 ResetConfigData

This function will revert any user changes by reloading the last activated configuration.

Parameter

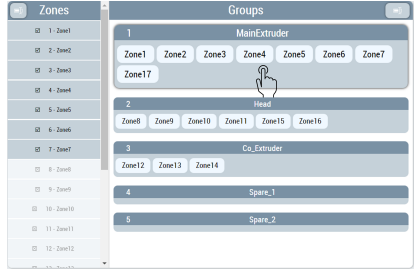
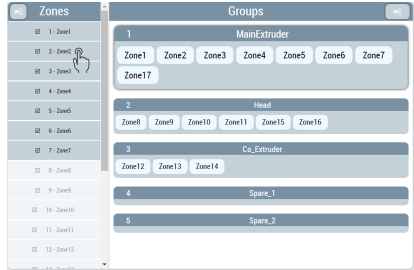
Name	Type	Description
-	-	-

Return value

Type	Description
void	No return value.

Available since version 12.8.0

10.1.3 User Interactions

Name	Description	Results on the control
Select a group.	By tapping or clicking on any group tile, will select that group.	
Add or remove zones from a group.	<p>Once a group is selected the temperature zones panel will show the possible options.</p> <p>Zones that are part of the group can be removed by clicking/tapping on it.</p> <p>Zones that are not part of any group can be added to selected group by clicking/tapping on it.</p> <p>Any zone that is already added to another group cannot be added to selected group and will be disabled for any click action.</p>	

10.1.4 Dialogs

10.1.4.1 Edit zone names

Edit Zone Names
✕

Index	Zone Name
1	Zone1
2	Zone2
3	Zone3
4	Zone4
5	Zone5
6	Zone6
7	Zone7
8	Zone8
9	Zone9
10	Zone10
11	Zone11
12	Zone12
13	Zone13

Ok
Cancel

User can edit zone name value for each channel in PLC from this dialog. After entering new values press **Ok** to accept new changes and **Cancel** to close the dialog.

10.1.4.2 Edit group names

Edit Group Names
✕

Index	Group Name
1	Group1
2	Group 2
3	Hydraulic
4	
5	
6	
7	
8	
9	
10	

Ok
Cancel

In this dialog group name values for each group are displayed. User can change the names and press **OK** to accept new changes and **Cancel** to close the dialog.

10.2 ZonelImageLayout

The ZonelImageLayout framework control allows users to present temperature zones over an image background in a layout that mimics the physical arrangement on the machine. All the changes can be done directly from the client thus eliminating the necessity of accessing the project through engineering.

Based on the standard TcHMI Control, the ZonelImageLayout control uses the standard background image attributes such as BackgroundImage to display an image in the backdrop. Different attributes and dialogs allow user to change the background image and related properties directly from the client.

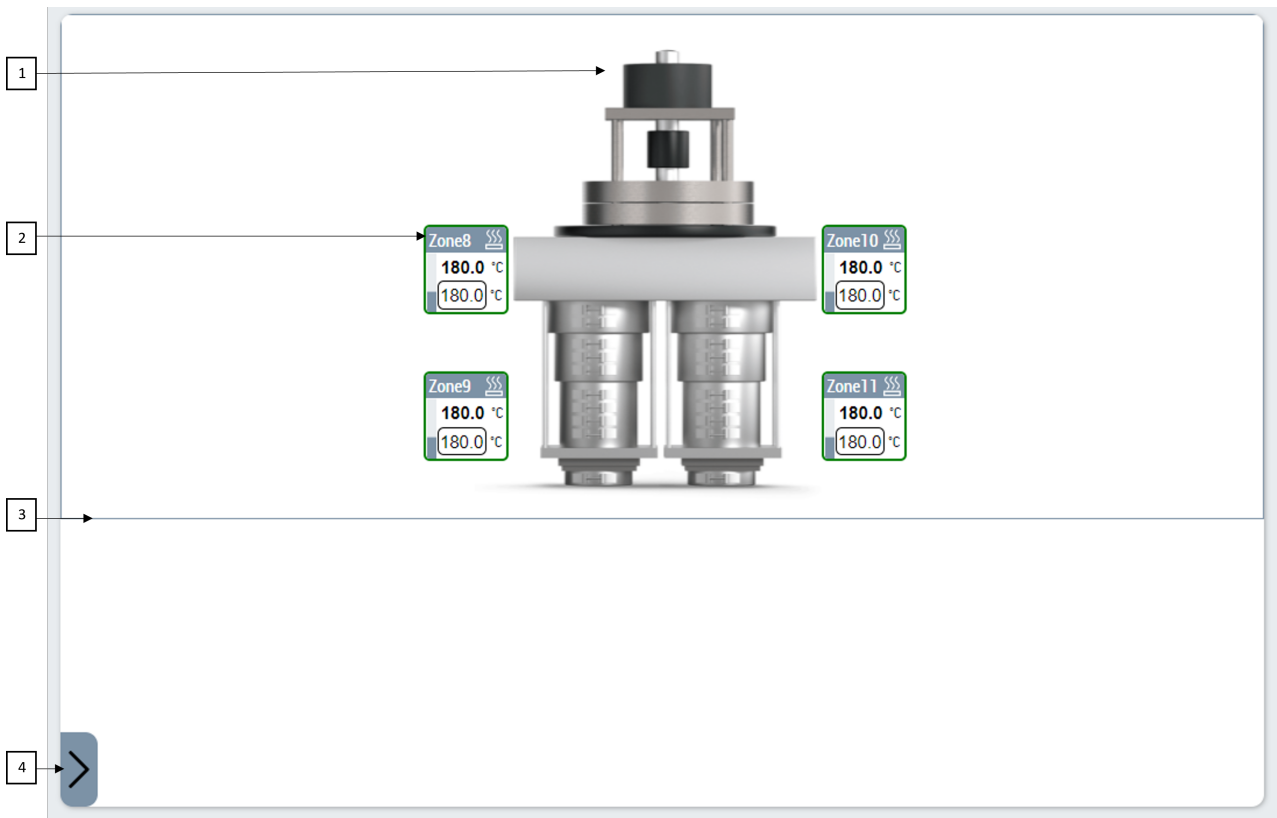
On top of the background the control portrays small tiles, with each tile presenting the actual status of a temperature zone.

User can use various attributes, dialogs, and touch/click interactions to move and arrange temperature zone tiles in different layouts.



ActivateConfig action

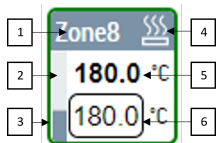
Any new changes to the control must be activated using the ActivateConfig [▶ 199] action to set new values to the PLC.



1. Background Image displayed on the control.
2. Temperature zones
3. Layout area to guide user.
4. Button to open zone selection dialog.

Tiles displaying zones

Each zone on the layout is represented with a visual element that gives more information of the linked temperature zone in the controller.



1. ZoneName
2. PowerLevel
3. Border color indicating zone status
4. Zone heating/cooling/error status
5. ActualTemperature of the zone
6. Input for user to write SetpointTemperature of the zone

Table below explains every possible border color and what it represents.

Border Color	Zone Status
Blue	Zone actual temperature is very low.
Green	Zones actual temperature value is within inner tolerances band.
Orange	Zones actual temperature value is out of inner tolerances but within outer tolerance bands.
Red	Zones actual temperature value is out of outer tolerances band.

10.2.1 Attributes

10.2.1.1 ZonelayoutConfig

With ZonelayoutConfig attribute, users can establish the default configuration to be shown on the control.

This attribute also enables binding to a server symbol, allowing users to capture and store new changes from the client side.

Editing object of type tchmi:framework#/definitions/ZonelayoutConfig

Properties

- General**
 - LayoutWidth: px
 - LayoutHeight: px
 - BackgroundPath:
 - BackgroundWidth: px
 - BackgroundHeight: px
 - Zones: (no items)
 - BackgroundPadding:
 - Left: px
 - Right: px
 - Top: px
 - Bottom: px
 - BackgroundHorizontalAlignment:
 - BackgroundVerticalAlignment:

The properties in the object are explained in table below.

Name	Type	Description
LayoutWidth	tchmi:framework#/definitions/ MeasurementValue	Width of the layout frame drawn on the control as a guidance area for user.
LayoutWidthUnit	tchmi:framework#/definitions/ MeasurementUnit	Unit of LayoutWidth.
LayoutHeight	tchmi:framework#/definitions/ MeasurementValue	Height of the layout frame.
LayoutHeightUnit	tchmi:framework#/definitions/ MeasurementUnit	Unit of LayoutHeight attribute.
BackgroundPath	tchmi:framework#/definitions/Path	Path to an image that will be shown in background of the control.
BackgroundWidth	tchmi:framework#/definitions/ MeasurementValue	Background image width.
BackgroundWidthUnit	tchmi:framework#/definitions/ MeasurementUnit	Unit of background image width
BackgroundHeight	tchmi:framework#/definitions/ MeasurementValue	Background image height.
BackgroundHeightUnit	tchmi:framework#/definitions/ MeasurementUnit	Unit of background image height
Zones	tchmi:framework#/definitions/ ZonelImageConfig	List of temperature zones that are displayed on the control. Each zone object contains 3 properties: X: Position of zone tile in the layout in X direction from top left. Y: Position of zone tile in the layout in Y direction from top left. ZoneIndex: Index of the zone object in aChannels array in PLC.
BackgroundPadding	tchmi:framework#/definitions/ Padding	Background image padding.
BackgroundHorizontalAlignment	tchmi:framework#/definitions/ HorizontalAlignment	Horizontal alignment of background image.
BackgroundVerticalAlignment	tchmi:framework#/definitions/ VerticalAlignment	Vertical alignment of background image.

Schema: tchmi:framework#/definitions/ZonelImageLayoutConfig

Attribute getter: getZonelImageLayoutConfig

Attribute setter: setZonelImageLayoutConfig

Available: since version 12.8.0

10.2.1.2 **Activation Required**

Is set to TRUE if the user makes new changes that are not activated.

Schema: tchmi:general#/definitions/Boolean

Attribute getter: getActivationRequired

Available: since version 12.8.0

10.2.1.3 **ShiftMode**

The ShiftMode is a binary attribute. Setting it to true is necessary to reorganize temperature zone tiles layout.

Once set, this mode alters the tiles to exclusively show zone index and name details. Users can then interactively drag, and position individual tiles as needed.

It is not possible to add or remove new zones to configuration when ShiftMode is set to true.

Schema: tchmi:general#/definitions/Boolean

Attribute getter: getShiftMode

Attribute setter: setShiftMode

Available: since version 12.8.0

10.2.1.4 **LayoutWidth**

Users might encounter scenarios where a single layout configuration appears on distinct controls with varying display dimensions. To aid users in constraining their layout designs to suitable sizes, a designated layout area is outlined on the control.

This rectangular frame serves as visual reference and doesn't impose limitations on element movement beyond its borders.

LayoutWidth attribute sets the width of the layout rectangle drawn on the control.

Schema: tchmi:framework#/definitions/MeasurementValue

Attribute getter: getLayoutWidth

Attribute setter: setLayoutWidth

Available: since version 12.8.0

10.2.1.5 **LayoutWidthUnit**

Unit of the LayoutWidth attribute.

Schema: tchmi:framework#/definitions/MeasurementUnit

Attribute getter: getLayoutWidthUnit

Attribute setter: setLayoutWidthUnit

Available: since version 12.8.0

10.2.1.6 **LayoutHeight**

Sets the height of layout rectangle outlined on the control.

Schema: tchmi:framework#/definitions/MeasurementValue

Attribute getter: getLayoutHeight

Attribute setter: setLayoutHeight

Available: since version 12.8.0

10.2.1.7 **LayoutHeightUnit**

Unit of the LayoutHeight attribute.

Schema: tchmi:framework#/definitions/MeasurementUnit

Attribute getter: getLayoutHeightUnit

Attribute setter: setLayoutHeightUnit

Available: since version 12.8.0

10.2.1.8 Mode

- In the View mode control only allows viewing the configured layout. It is not possible to add or remove temperature zones, shift zones in shift mode or change background image properties using dialogs in view mode.
- In the config mode user with operate access rights can change the configuration using various dialogs.

Schema: tchmi:framework#/definitions/ZoneImageLayout.Mode

Attribute getter: getLayoutHeightUnit

Attribute setter: setLayoutHeightUnit

Available: since version 12.8.0

10.2.2 Functions

10.2.2.1 ActivateConfig

By calling this function all user changes will be activated and written to ZoneImageLayoutConfig attribute.

Parameter

Name	Type	Description
-	-	-

Return value

Type	Description
void	No return value.

Available: since version 12.8.0

10.2.2.2 ResetConfigData

This function will revert any user changes by reloading the last activated configuration.

Parameter

Name	Type	Description
-	-	-

Return value

Type	Description
void	No return value.

Available since version 12.8.0

10.2.2.3 ChangeBackgroundImage

With operate access executing this function will start [Change background image \[▶ 206\]](#) dialog. And allow the user to change background image.

Parameter

Name	Type	Description
-	-	-

Return value

Type	Description
void	No return value.

Mode **attribute must be set to Config to use this feature. Please check [Mode \[▸ 199\]](#) for more information**

Available: since version 12.8.0

10.2.2.4 EditBackgroundImage

With operate access executing this function will start [Edit background image \[▸ 206\]](#) dialog. And allow user to change background image properties.

Parameter

Name	Type	Description
-	-	-

Return value

Type	Description
void	No return value.

Mode **attribute must be set to Config to use this feature. Please check [Mode \[▸ 199\]](#) for more information**

Available: since version 12.8.0

10.2.3 User Interactions**10.2.3.1 Open zones list panel**

To add new zones to the layout or to remove any zone already in layout, the user needs to access the list of temperature zones that are available.

To do this click on the button that opens the zones list.

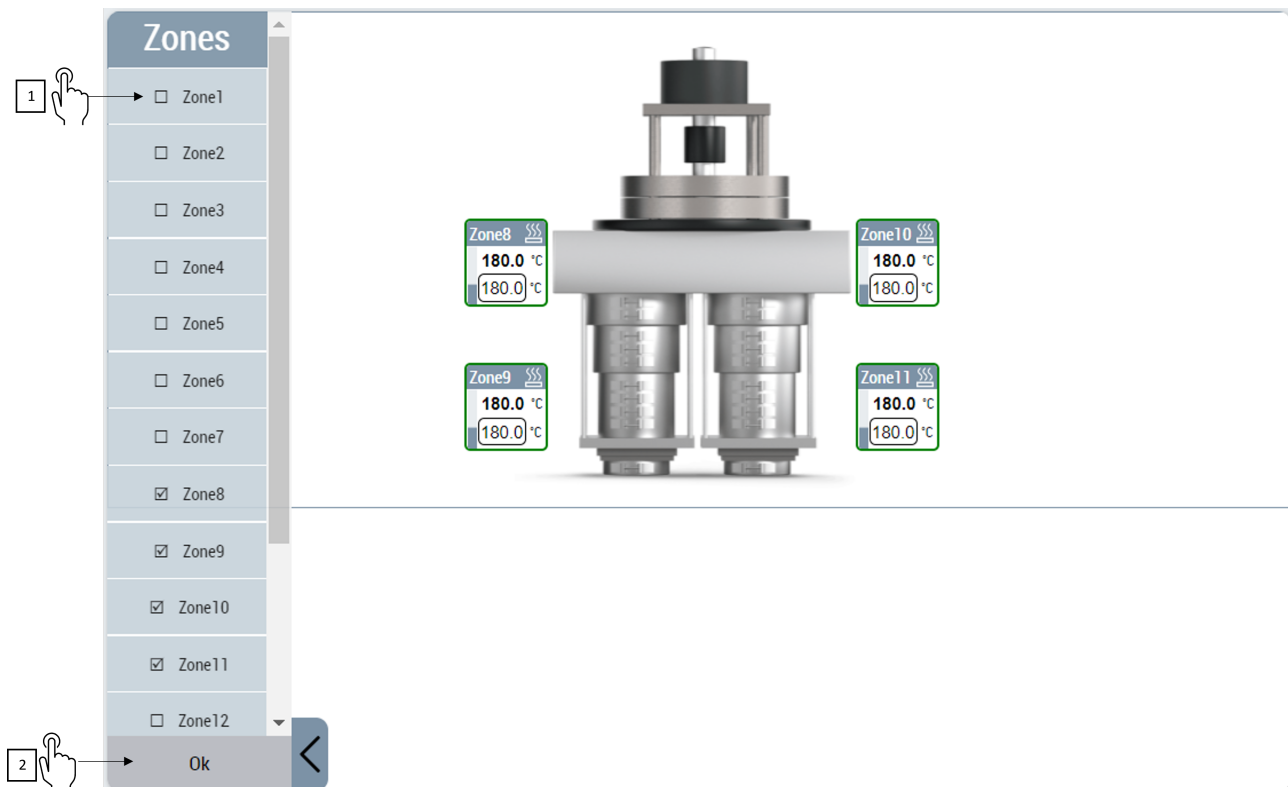


The zones list will list all the available zones in the project. Every temperature channel that is assigned to a temperature group will be listed.

Mode **attribute must be set to Config to use this feature. Please check [Mode \[► 199\]](#) for more information**

10.2.3.2 Add new temperature zones to layout

To add new temperature zones that are not already part of the layout, first open the zones list panel.

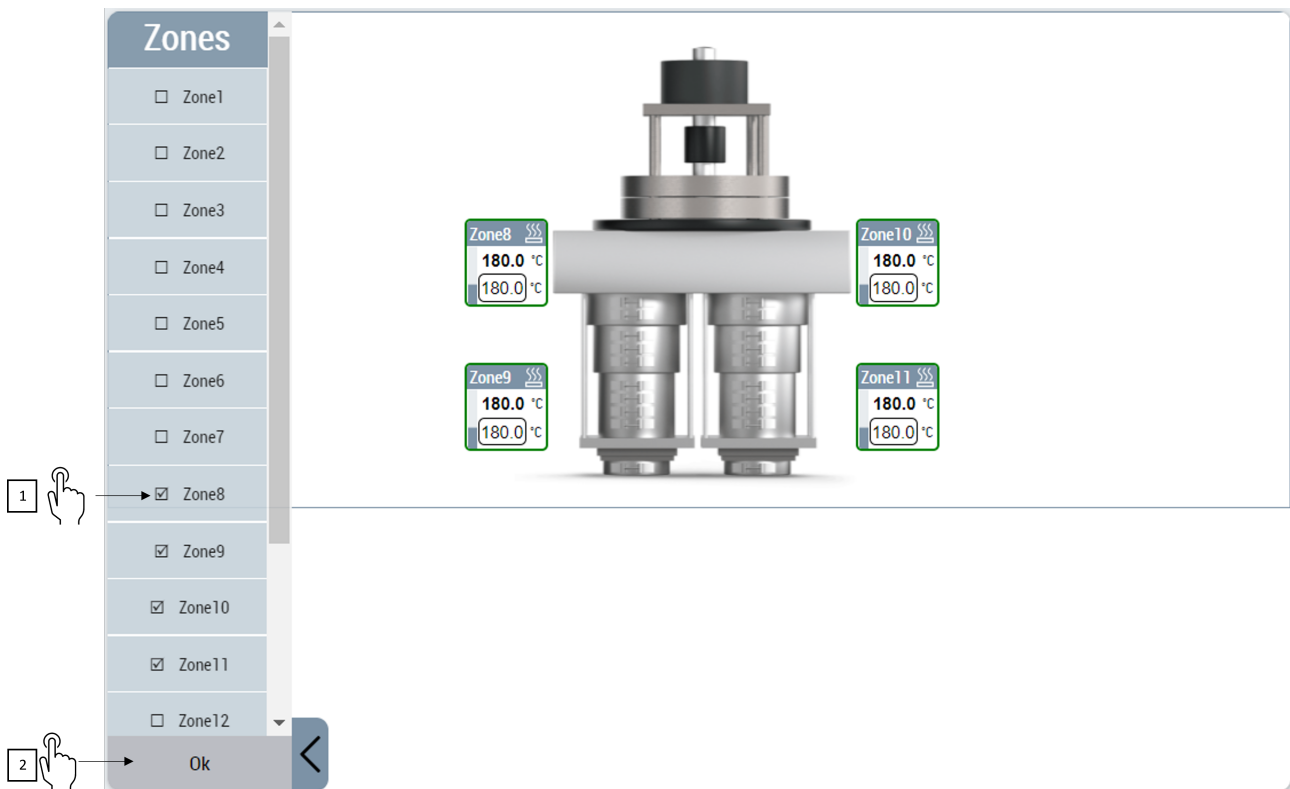


1. Click on temperature zone that is to be added. You can select multiple zones and add all of them together.
2. Click the **Ok** button at the bottom of zones list to accept your changes.
 - ⇒ The new zones will be added to the layout.

Mode **attribute must be set to Config to use this feature. Please check [Mode \[▶ 199\]](#) for more information**

10.2.3.3 Remove temperature zones from layout

To remove a temperature zone from the layout:



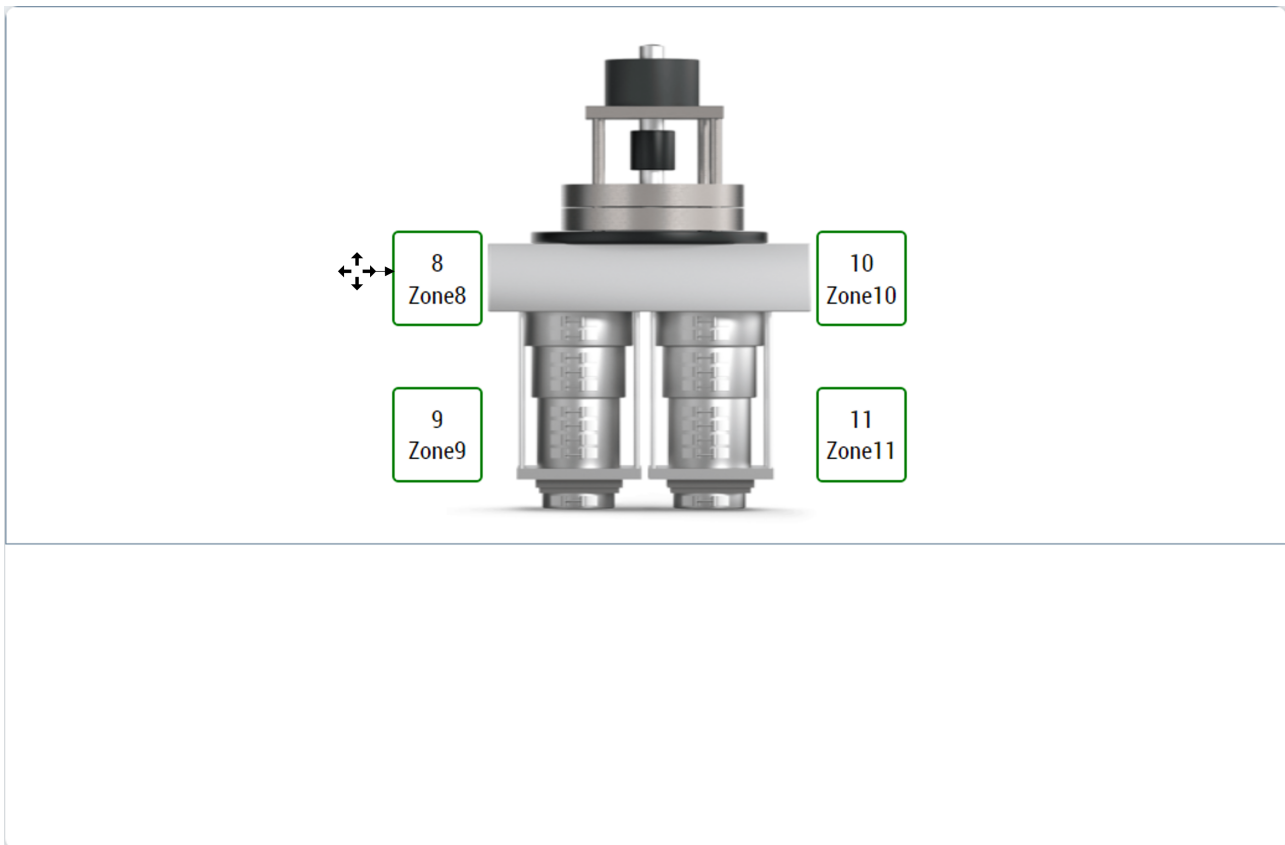
1. After opening the zones list panel, click on the temperature zone that is to be removed. You can select multiple zones to be added and removed simultaneously.
2. Click the **Ok** button at the bottom of zones list to accept your changes.
 - ⇒ The selected zones will be removed from the layout.

Mode **attribute must be set to Config to use this feature. Please check [Mode \[▶ 199\]](#) for more information**

10.2.3.4 Moving temperature zones in shift mode

To move temperature zone tiles in the layout area the controls [ShiftMode \[▶ 197\]](#) attribute must be set to TRUE.

Once in shift mode the user can start moving individual temperature zones using touch-drag action in any direction.



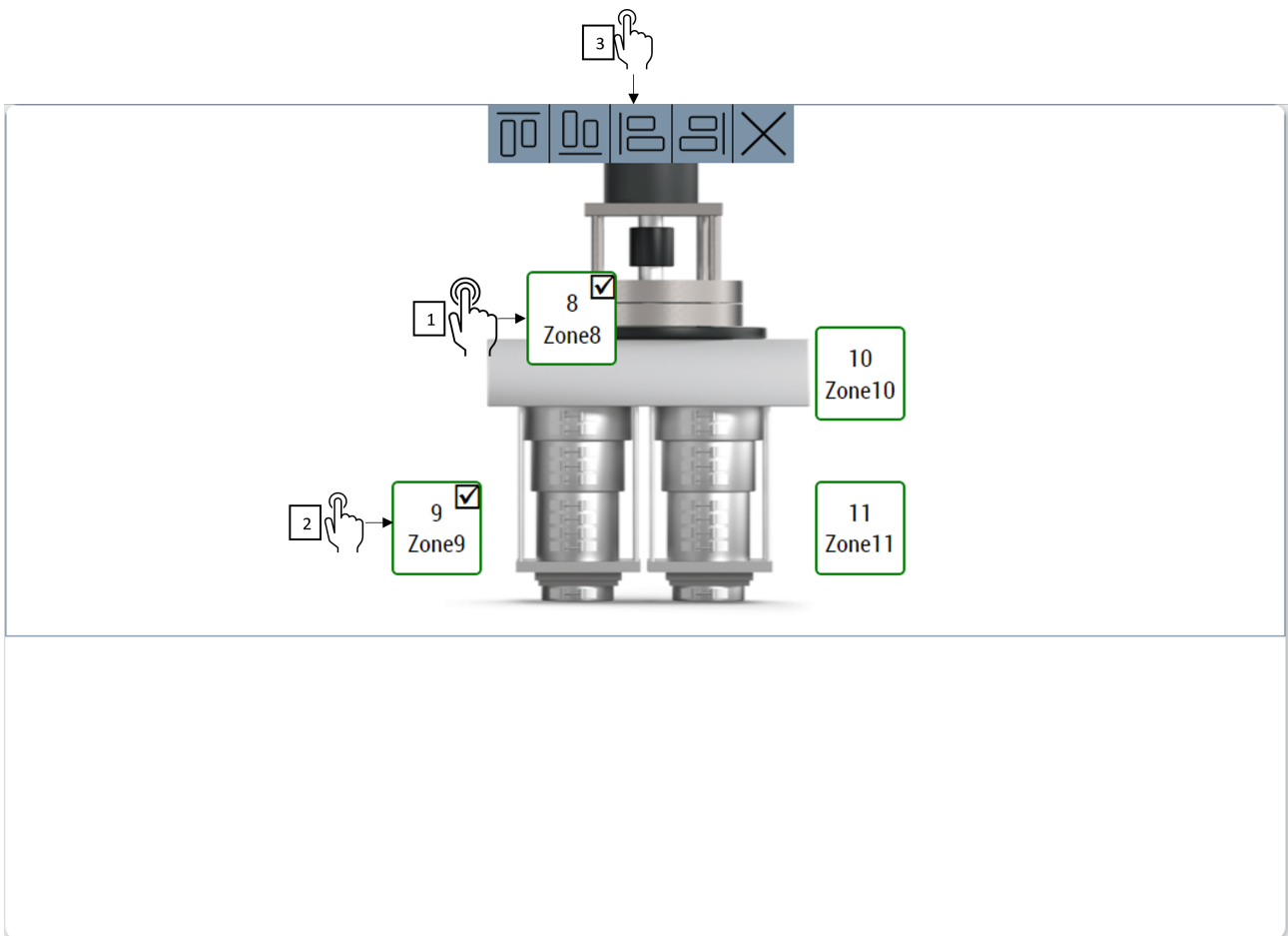
Mode attribute must be set to Config to use this feature. Please check [Mode \[▶ 199\]](#) for more information

10.2.3.4.1 Align multiple zones

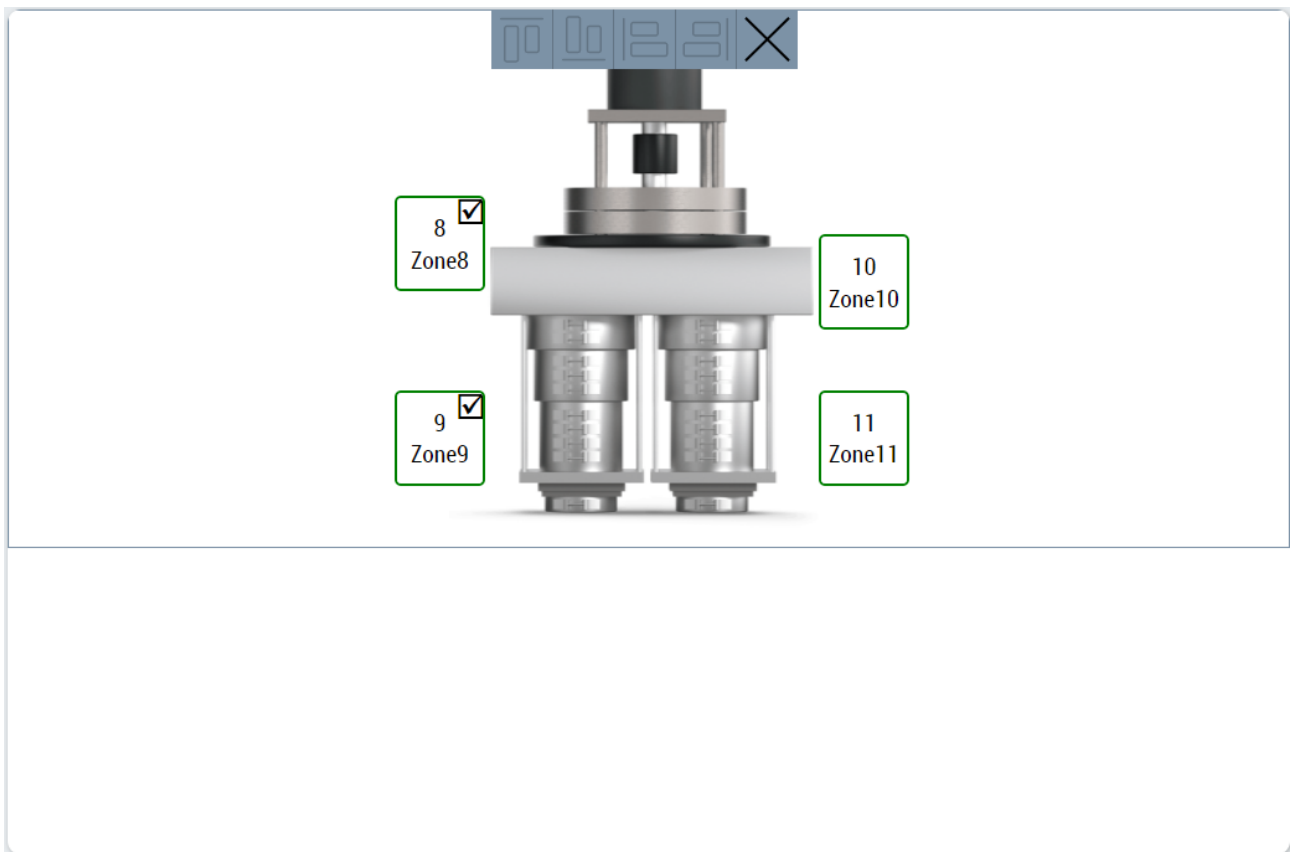
To move and organize multiple temperature zone tiles on the screen.

1. In shift mode, to select first zone double tap on it.
2. This will start selection and alignment mode; on the top side of the control an alignment bar will appear showing available aligning options for the selection.
3. Second zone can be added to the selection by tapping on it.
4. A zone that was added to selection can be removed by tapping on it again.
5. Once two or more zones are selected user can click on one of the below alignment options.
 - Align Left: All the selected zones will align with the left position of the zone that is located farthest to the left.
 - Align Right: All the selected zones will align with the right position of the zone that is located farthest to the right.
 - Align Top: All the selected zones will align with the top position of the zone that is located farthest to the top.
 - Align Bottom: All the selected zones will align with the top position of the zone that is located farthest to the bottom.

For example, temperature zones Zone8 and Zone9 were selected and aligned to left in image below.



The result of aligning will be as per the image below.



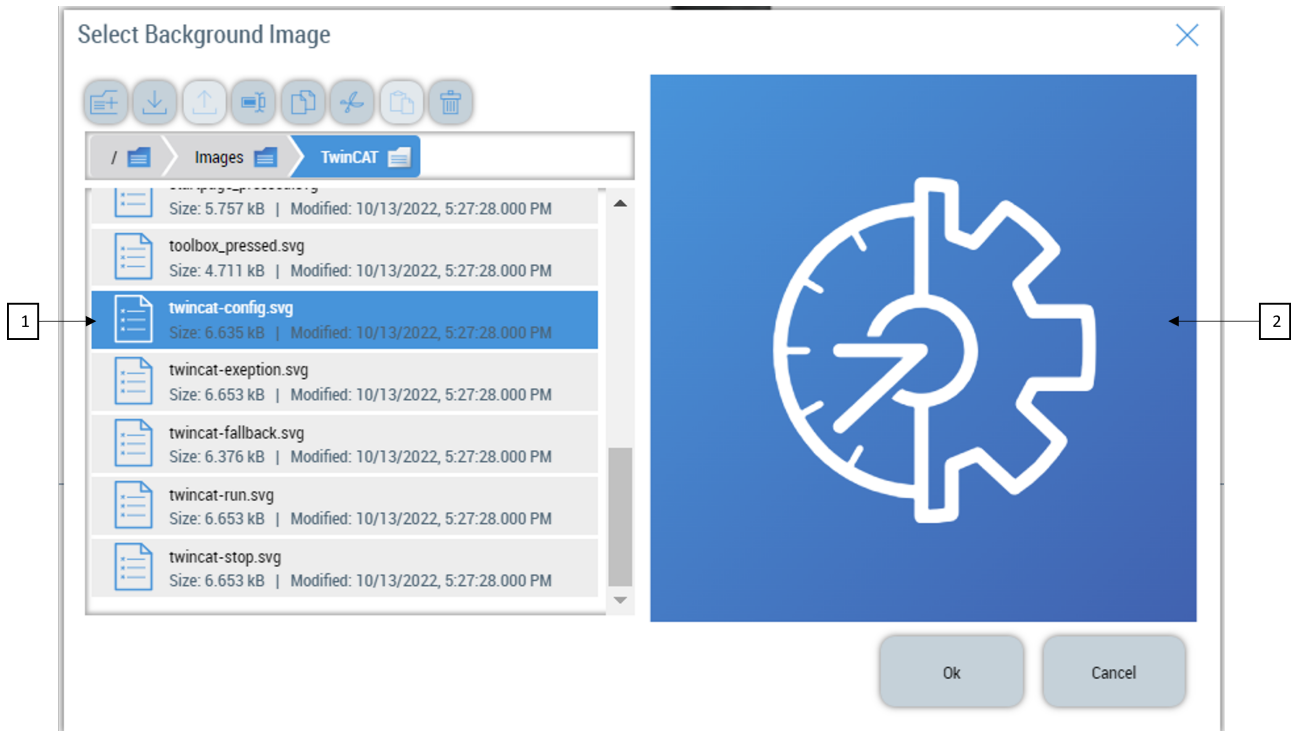
To end multiple selection and alignment mode user must click on close button on the alignment bar on top of the control.

Mode **attribute must be set to Config to use this feature. Please check [Mode \[▶ 199\]](#) for more information**

10.2.4 Dialogs

10.2.4.1 Change background image

The user can change the background image of the ZonelImageLayout control using this dialog.



The dialog pop up contains:

1. File Explorer control, this will allow the user to navigate to the right image file destination.
2. Image viewer shows the pre-view thumbnail of the selected image file.

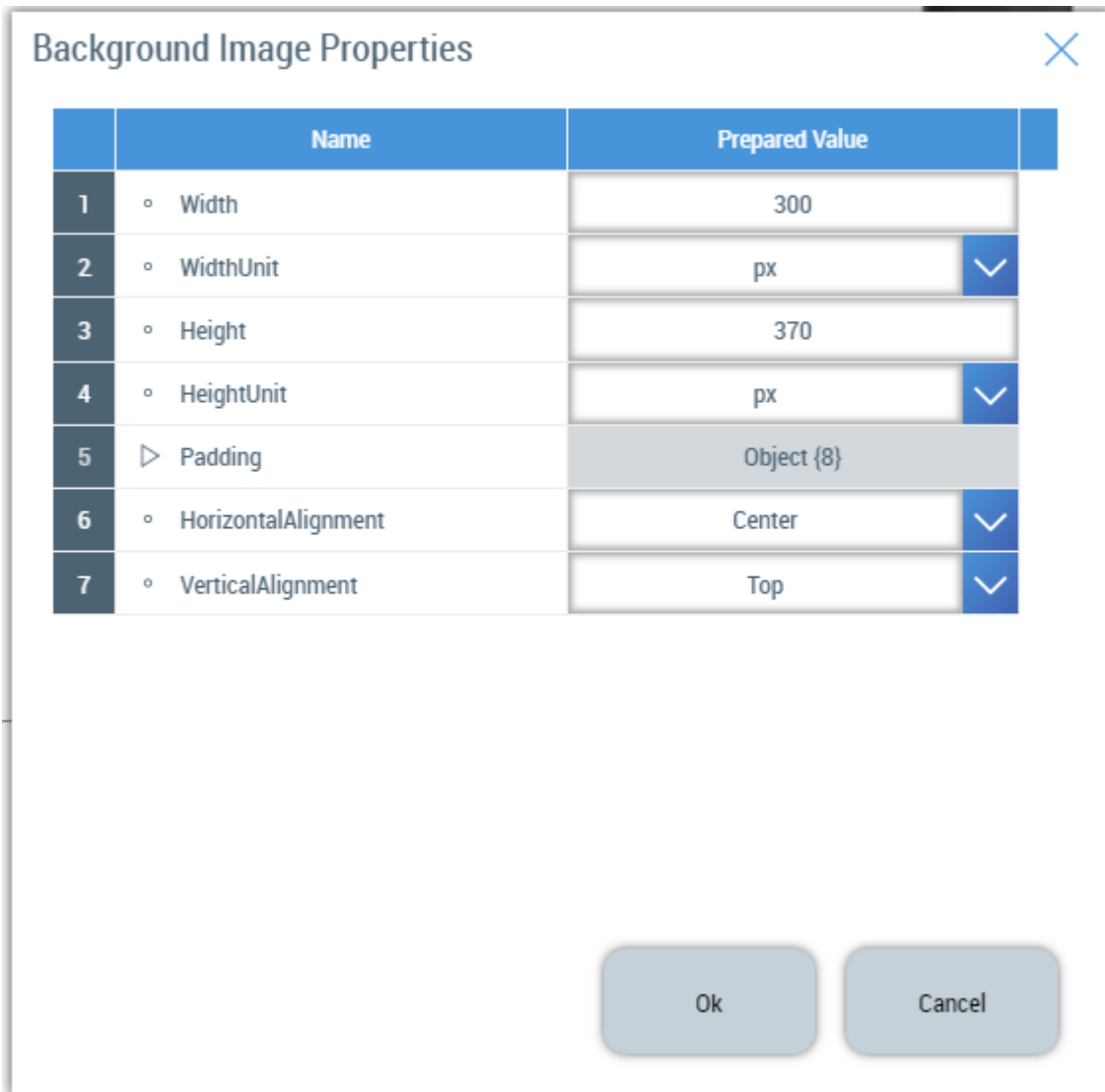
Once the right file is selected in the file explorer control, the user can press the **OK** button to accept the new change. This will apply the selected image as the background image of the control.

Or the user can press **Cancel** to avoid the changing of the background image.

Mode **attribute must be set to Config to use this feature. Please check [Mode \[▶ 199\]](#) for more information**

10.2.4.2 Edit background image

All the background image properties of the control are available for editing in this dialog.



The user can edit each property and confirm changes with the **Ok** or **Cancel** button.

Mode **attribute must be set to Config to use this feature. Please check [Mode \[▶ 199\]](#) for more information**

10.2.5 Events

Category: Actions

Name	Description
onUserConfigActivated [▶ 207]	This event is generated when a new configuration is activated on the control.
onEditingCanceled [▶ 208]	This event is generated when user changes are canceled, and the old configuration is reloaded on the control.

10.2.5.1 onUserConfigActivated

This event is generated when a new configuration is activated on the control.

Available: since version 12.8.0

10.2.5.2 onEditingCanceled

This event is generated when user changes are canceled, and the old configuration is reloaded on the control.

Available: since version 12.8.0

10.2.6 Step by step

10.2.6.1 Displaying a single layout configuration in different controls

It is possible to have the TcHMI project organized in such a way, that a user with special operation rights can only configure the layout displayed on the control.

This configured layout should be displayed in other contents using different instances of ZoneImageLayout controls that work in View mode.

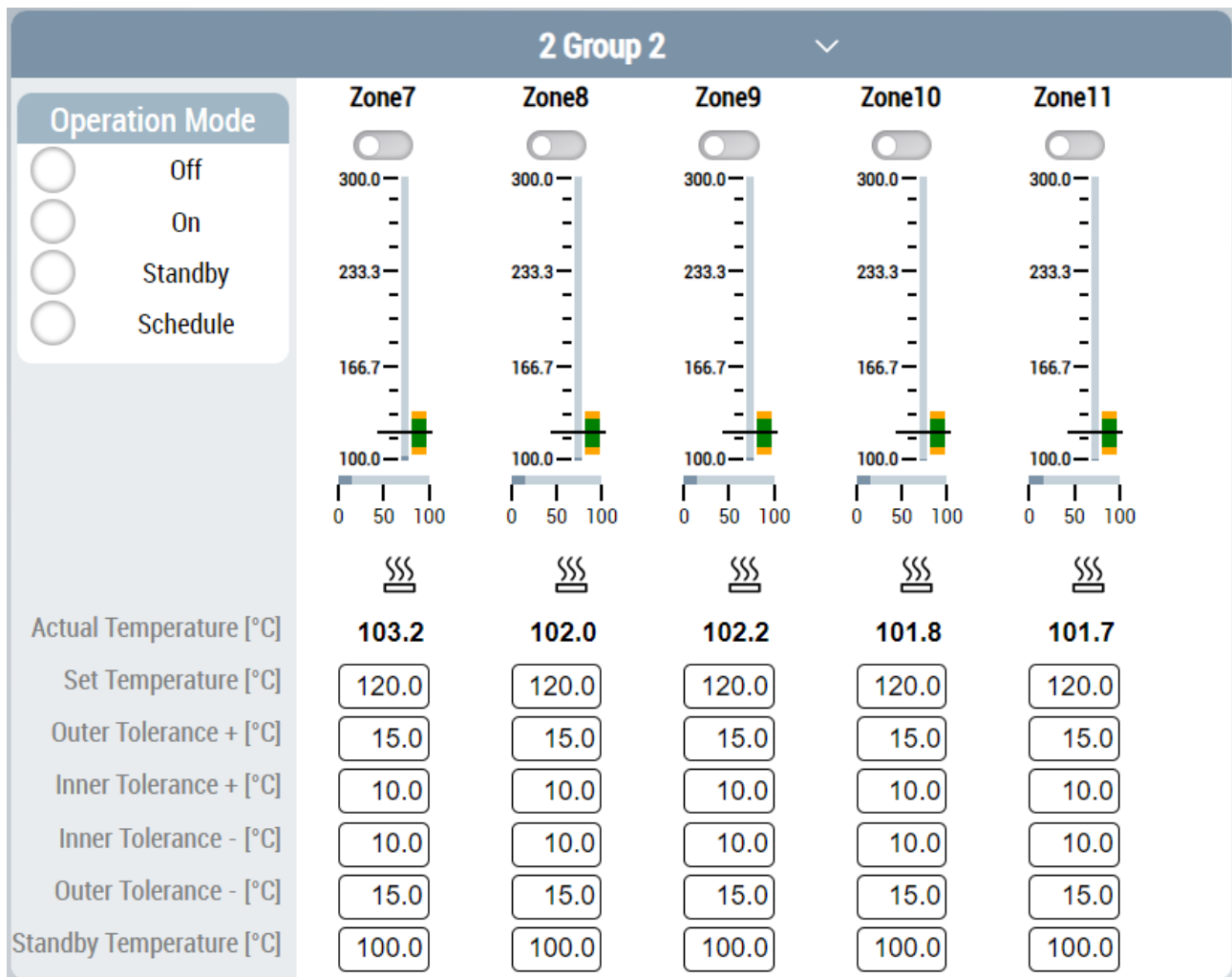
1. Create a server symbol of type `tchmi:framework#/definitions/ZoneImageLayoutConfig` and initialize its default values.
2. Create configuring control instance by drag and drop from toolbar. Bind this controls `ZoneImageLayoutConfig` attribute to the server symbol created in last step. Keep the binding two-way so changes will be stored back to the server symbol.
3. Set Mode attribute of this control to Config as this control will be used to edit the layout.
4. Create a new instance of control that will be used only for visualizing the layout. Set Mode attribute to View.
5. Bind the server symbol created in the 1st step to its `ZoneImageLayoutConfig` attribute. No need to set two-way binding mode as this control will only display the config and never change it.
6. This way same configuration can be displayed on different controls in the HMI.

10.3 ZoneConfiguration

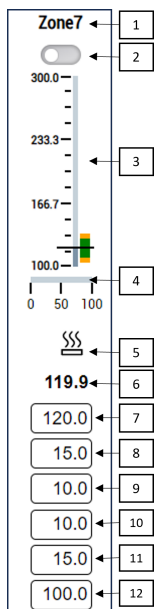
NuGet package: Beckhoff.TwinCAT.HMI.Plastic.Temperature

Available: since version 12.8.0

ZoneConfiguration control is used to visualize and allow users to change different operational parameters of temperature groups and zones configured in the PLC.



Every temperature zone in the selected temperature zone group is visualized using different input and graphical controls.



1. ZoneName of the channel.
2. The user can enable/disable a channel with this control.
3. A linear gauge will visualize the actual temperature, set temperature, inner and outer tolerance bands on a scale.
4. A gauge will visualize the PowerLevel property of the channel.

5. Temperature channels cooling/heating status will be displayed with an appropriate icon.
6. ActualTemperature of the channel.
7. SetpointTemperature input for the channel.
8. Outer tolerance in positive direction.
9. Inner tolerance in positive direction.
10. Inner tolerance in negative direction.
11. Outer tolerance in negative direction.
12. Standby temperature of the zone.

10.3.1 Attributes

Category: Colors

Name	Description
HeaderBackgroundColor	Definition of the background color of the header.
HeaderFontColor	Definition of the text color of the header.

Category: Header Configuration

Name	Description
HeaderHeight	Height of the control's header.
HeaderHeightUnit	Unit of the HeaderHeight attribute.
HeaderFontSize	Font size of the header.
HeaderFontSizeUnit	Unit for header font size.

Category: Config

Name	Description
ActualZoneGroupId [▶ 211]	Group Id of the temperature group displayed on the control.
DisplayOperationModes [▶ 211]	With this attribute the operation controls can be shown on the control.
DigitsAfterDecimal [▶ 211]	Individual values display resolution.

10.3.1.1 HeaderBackgroundColor

Definition of the background color of the header.

Schema: tchmi:framework#/definitions/Color

Attribute getter: getHeaderBackgroundColor

Attribute setter: setHeaderBackgroundColor

Available: since version 12.8.0

10.3.1.2 HeaderFontColor

Definition of the text color of the header.

Schema: tchmi:framework#/definitions/Color

Attribute getter: getHeaderFontColor

Attribute setter: setHeaderFontColor

Available: since version 12.8.0

10.3.1.3 ActualZoneGroupId

With this attribute the user can change the GroupId of the configured temperature zone group. By changing this attribute the user can select the temperature group that is visualized.

Schema: tchmi:general#/definitions/Number

Attribute getter: getActualZoneGroupId

Attribute setter: setActualZoneGroupId

Available: since version 12.8.0

10.3.1.4 DigitsAfterDecimal

The resolution of individual number values displayed on the control can be updated with this attribute.

Schema: tchmi:general#/definitions/Number

Attribute getter: getDigitsAfterDecimal

Attribute setter: setDigitsAfterDecimal

Available: since version 12.8.0

10.3.1.5 DisplayOperationModes

By default, the operation controls that set the active mode of the selected temperature zone group are not displayed on the control. With this attribute the controls can be made available for the user.

Schema: tchmi:general#/definitions/Boolean

Attribute getter: getDisplayOperationModes

Attribute setter: setDisplayOperationModes

Available: since version 12.8.0

10.3.1.6 HeaderHeight

Definition of the height of the header.

Schema: tchmi:framework#/definitions/MeasurementValue

Attribute getter: getHeaderHeight

Attribute setter: setHeaderHeight

Available: since version 12.8.0

10.3.1.7 HeaderHeightUnit

Defines the unit of the height of the header:

- Pixels for an absolute size
- Percent for a relative size

Schema: tchmi:framework#/definitions/MeasurementUnit

Attribute getter: getHeaderHeightUnit

Attribute setter: setHeaderHeightUnit

Available: since version 12.8.0

10.3.1.8 HeaderFontSize

The font size. If the percent is specified as the unit, this is relative to the font size of the parent element.

Schema: tchmi:framework#/definitions/MeasurementValue

Attribute getter: getHeaderFontSize

Attribute setter: setHeaderFontSize

Available: since version 12.8.0

10.3.1.9 HeaderFontSizeUnit

Defines the unit of the height of the header:

- Pixels for an absolute size
- Percent for a relative size

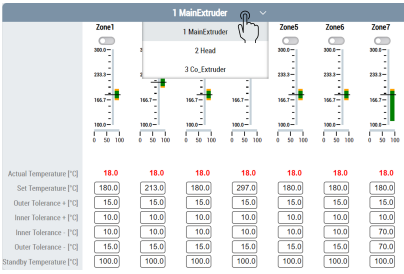
Schema: tchmi:framework#/definitions/MeasurementUnit

Attribute getter: getHeaderHeightUnit

Attribute setter: setHeaderHeightUnit

Available: since version 12.8.0

10.3.2 User Interactions

Name	Description	Result on the control
Single Tap on control header.	Select active group id in combo box drop down that appears if a user taps on the header area of control.	

More Information:
www.beckhoff.com/tf8550

Beckhoff Automation GmbH & Co. KG
Hülshorstweg 20
33415 Verl
Germany
Phone: +49 5246 9630
info@beckhoff.com
www.beckhoff.com

