SIEMENS

Application Example• 06/2015

Configuration Examples for HMI Faceplates

SIMATIC Comfort Panels, Runtime Advanced and WinCC (TIA Portal)

https://support.industry.siemens.com/cs/ww/en/68014632

Warranty and Liability

Note

The application examples are not binding and do not claim to be complete with regard to configuration, equipment or any contingencies. The application examples do not represent customer-specific solutions. They are only intended to provide support for typical applications. You are responsible for the correct operation of the described products. These application examples do not relieve you of the responsibility of safely and professionally using, installing, operating and servicing equipment. When using these Application Examples, you recognize that we cannot be made liable for any damage/claims beyond the liability clause described. We reserve the right to make changes to these application examples at any time and without prior notice. If there are any deviations between the recommendations provided in this Application Example and other Siemens publications – e.g. Catalogs – the contents of the other documents shall have priority.

We do not accept any liability for the information contained in this document.

Any claims against us – based on whatever legal reason – resulting from the use of the examples, information, programs, engineering and performance data etc., described in this Application Example shall be excluded. Such an exclusion shall not apply in the case of mandatory liability, e.g. under the German Product Liability Act (“Produkthaftungsgesetz”), in case of intent, gross negligence, or injury of life, body or health, guarantee for the quality of a product, fraudulent concealment of a deficiency or breach of fundamental contractual obligations (“wesentliche Vertragspflichten”). The compensation for damages due to a breach of a fundamental contractual obligation is, however, limited to the foreseeable damage, typical for the type of contract, except in the event of intent or gross negligence or injury to life, body or health. The above provisions do not imply a change of the burden of proof to your detriment.

Any form of duplication or distribution of these application examples or excerpts hereof is prohibited without the expressed consent of Siemens AG.

Security information

Siemens provides products and solutions with industrial security functions that support the secure operation of plants, solutions, machines, equipment and/or networks. They are important components in a holistic industrial security concept. With this in mind, Siemens’ products and solutions undergo continuous development. Siemens recommends strongly that you regularly check for product updates.

For the secure operation of Siemens products and solutions, it is necessary to take suitable preventive action (e.g. cell protection concept) and integrate each component into a holistic, state-of-the-art industrial security concept. Third-party products that may be in use should also be considered. For more information about industrial security, visit http://www.siemens.com/industrialsecurity.

To stay informed about product updates as they occur, sign up for a product-specific newsletter. For more information, visit http://support.industry.siemens.com.

Table of Contents

Preface

Objective of this application

The objective of this application is to provide you with an aid for generating faceplates.

If you are not yet familiar with the topic of “Faceplates”, please view the documentation “Basic knowledge on the topic of HMI faceplates”.

The documentation “Basic knowledge on the topic of HMI faceplates” discusses the following topics.

- Releasing the faceplate
- Updating faceplates
- HMI user data type (HMI UDT)
- PLC data types

Main topics of this application

Using several practice-relevant examples, the individual configuration steps are described in detail. The examples are available as compressed download.

Validity

- Software version as of WinCC Comfort V13 SP1
- All SIMATIC HMI operator panels that support faceplates (see chapter 2.4 “Supported operator panels).

  Information on the function scope of the SIMATIC HMI operator panels is available in the Online Help of WinCC (TIA Portal) or FAQ http://support.automation.siemens.com/WW/view/en/40227286.

1 Task

1.1 Overview

Introduction

In plant and mechanical engineering in particular, the requirements for a uniform operator control and monitoring concept are high in order to guarantee short familiarization times or high operating security for the later operator of the plant.

A uniform operator control and monitoring concept is also important for the maintenance of plants. This can reduce down times during failures, maintenance works, or a plant expansion.

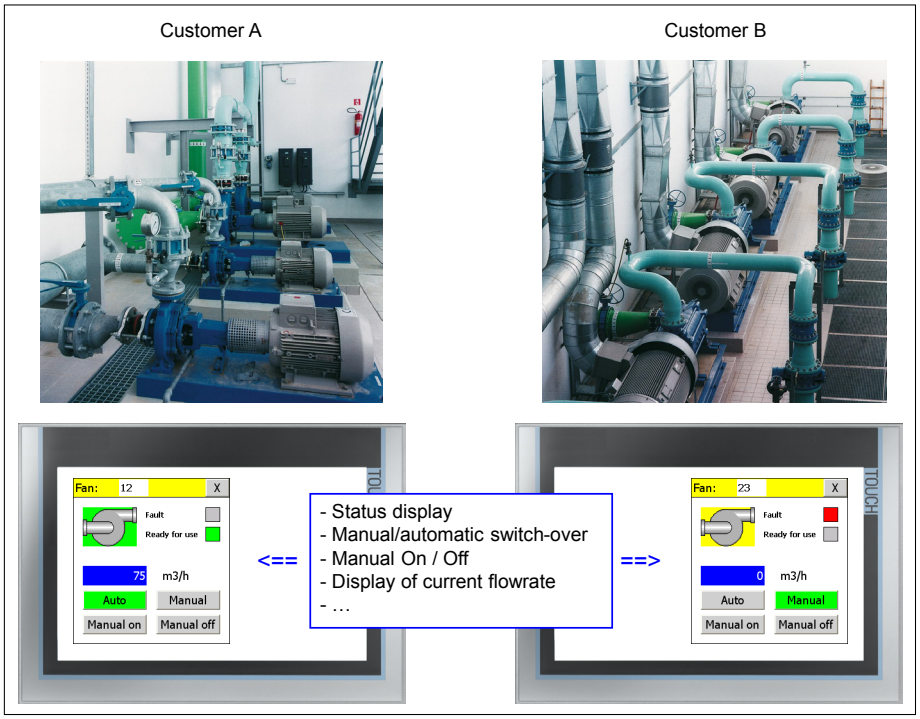
In serial machine construction, configured functions, such as drive on/off, switchover automatic/manual mode etc. are frequently used. These functions generally only need to be adjusted to the respective machine and its control tags. Using preprogrammed objects reduces engineering times and costs.

1.2 Overview of the automation task

The figure below provides an overview of the automation task.

You have different customers using the identical functions for controlling a pump or a drive.

Figure 1-1



Description of the automation task

You have several customers whose plant areas have similar functions.

The plant sections consist of several drives for pumps and fans. It shall be possible to operate each drive via one “control panel”.

The following functions shall be possible via the “control panel”.

- Output of the current status of the drive
- Switchover manual/automatic mode
- Manual mode on/off
- Output e.g. of the current flow rate

Due to the different requirements for the HMI operator panel, it must be possible to use HMI operator panels with different display sizes.

Problem description

The following requirements for the operating concept must be fulfilled:

- Uniform appearance for all plants.
- Can be used several times even for different customers.
- Central changeability of the control panel.
- Automatic update of the control panels used in the project after performed modifications or expansions.
- Created control panel shall be available as a “library” so it can be used by different persons in the engineering process.
- A connection to a data block for data exchange between the control panel and the connected PLC controller. The data block of the PLC controller has a respective given structure.

2 Solution

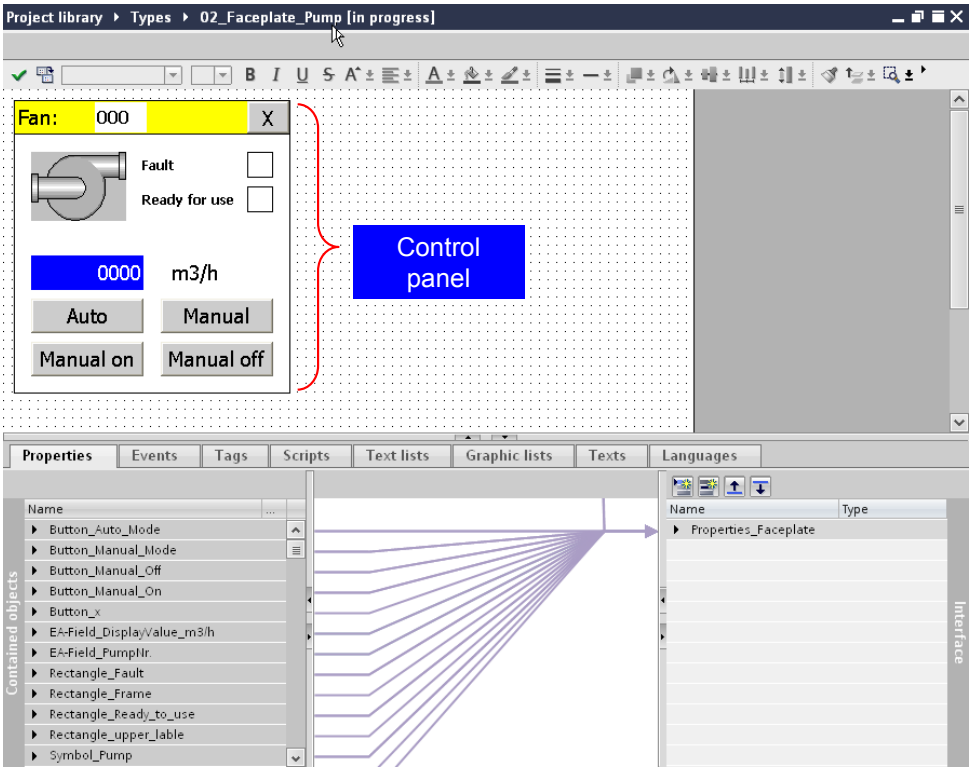
2.1 Solution overview

For realizing the tasks described in chapter 1 it is advisable to use “**faceplates**”.

Faceplates consist of a compilation of display and operator objects which you can manage and modify centrally in a library. Storing the faceplates in a library enables calling and using them easily in different projects.

The following figure shows the editor of a faceplate in progress. Don’t be irritated by the “Lines”, which in the beginning seem rather confusing. In the further course of the application, the correlations are explained step by step.

Figure 2-1



2.2 General notes on the application

Advantages

This application offers the following advantages:

- The step-by-step instructions for creating a faceplate and the practical application examples reduce the familiarization time.
- Faceplates increase the efficiency.
  Storing the faceplate in a library enables calling and using them easily in different projects.
- Configuration instruction and example projects
  Nine practice-relevant examples are described in the documentation. The described faceplates are available ready configured in the project on hand.
- Avoiding errors through a simple connection to the STEP 7 control program.

Topics not covered in this application

This application does not include a description of

- the used Comfort Panel. This document describes only the steps that are necessary for this application.
- how to create, for example, a network connection between operator panel and configuration computer or PLC controller.
- the installation of the SIMATIC software WinCC (TIA Portal).

Basic knowledge of these topics is assumed.

Assumed knowledge

- Basic knowledge how to work with and how to operate a Comfort Panel is expected.
- Skills for using WinCC (TIA Portal).

2.3 Hardware and software components used

The application was created with the following components:

Hardware components

Table 2-1

Component	Qty	Article number	Note
CPU 1515-2 PN	1	6ES7 515-2AM00-0AB0	**Optional:** the PLC is used for simulation purposes (data exchange between control panel and PLC).
TP1200 Comfort Panel	1	6AV2124-0MC01-0AX0	**Alternatively:** See Chapter 2.4.

Standard software components

Table -22

Component	No.	Article number	Note
WinCC Comfort V13 SP1	1	6AV2101-0AA01-0AA5	**Alternatively:** WinCC Advanced V13 SP1 or higher. WinCC Professional V13 SP1 or higher.
STEP 7 Professional V13 SP1	1	6ES7822-1AA01-0YA7	**Optional:** For the example configuration. Alternatively: higher version.

Sample files and projects

The following list includes all files and projects that are used in this example.

Table 2-3

Component	Note
68014632_Faceplates_instruction_CODE_v20.zip	This zip file contains the WinCC Comfort and STEP 7 project.
68014632_Faceplates_instruction_DOKU_v20_e.zip	Documentation

2.4 Supported operator panels and controllers

The following tables list those devices which support the faceplates.

Runtime

Table 2-4

No.	Operator panel
1.	WinCC Runtime Advanced
2.	WinCC Runtime Professional (this documentation only discusses the panels and WinCC Runtime Advanced)

Comfort Panels

Table 2-5

No.	Operator panel
3.	KP400 Comfort
4.	KTP400 Comfort
5.	KP700 Comfort
6.	TP700 Comfort
7.	KP900 Comfort
8.	TP900 Comfort
9.	KP1200 Comfort
10.	TP1200 Comfort
11.	KP1500 Comfort
12.	TP1500 Comfort
13.	TP1900 Comfort
14.	TP2200 Comfort

Panels

Table 2-6

No.	Operator panel
1.	TP 277
2.	OP 277

Mobile Panels

Table 2-7

No.	Operator panel
1.	Mobile Panels
2.	Mobile Panel 277
3.	Mobile Panel 277 IWLAN V2
4.	Mobile Panel 277F IWLAN V2
5.	Mobile Panel 277F IWLAN (RFID Tag)

Multi Panels

Table 2-8

No.	Operator panel
1.	MP 277
2.	MP 377

SIMATIC controllers

Table 2-9

No.	SIMATIC Controller
1.	S7-300/400 controllers
2.	S7-1200 controllers
3.	S7-1500 controllers

3 Configuration and Settings

In this chapter, details on the configuration of a faceplate are described.

The examples are based on one another.

3.1 Configuration examples

Using typical application examples, it is shown how to realize the described tasks with a faceplate.

The examples are based on one another. The function scope increases from example to example.

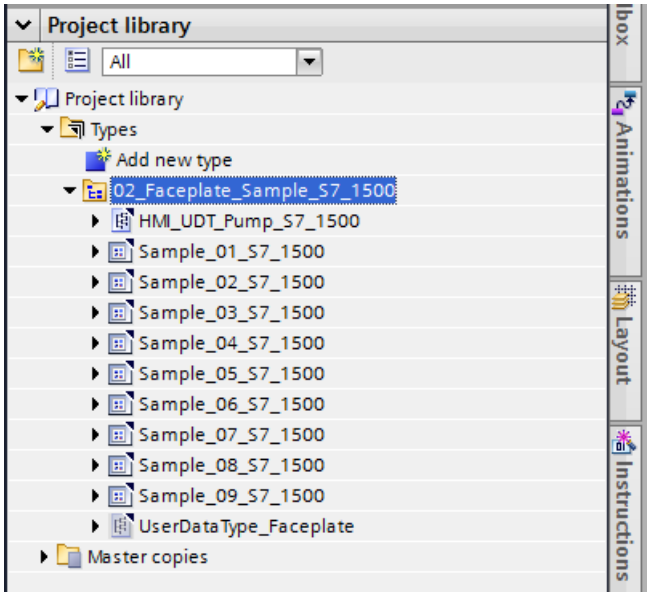
General notes on the example projects

An S7-1500 controller was used here as controller.

Apart from example “Sample_07” (faceplate with a PLC data type), the examples can also be used for an “S7-300/S7-400” controller.

The figure below shows the library with the stored faceplates.

Figure 3-1



3.1.1 Sample_01: assigning a configurable function to a button

Assigning a configurable function to a button

With two buttons it shall be possible for the user to assign any system function at the “Press” event, such as “SetBit” for example.

Figure 3-2

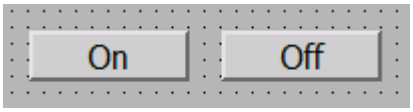
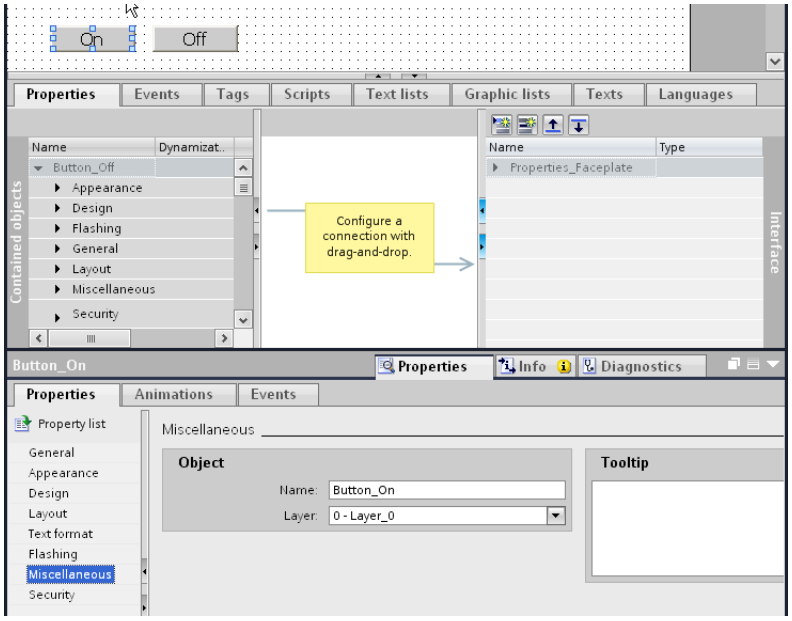
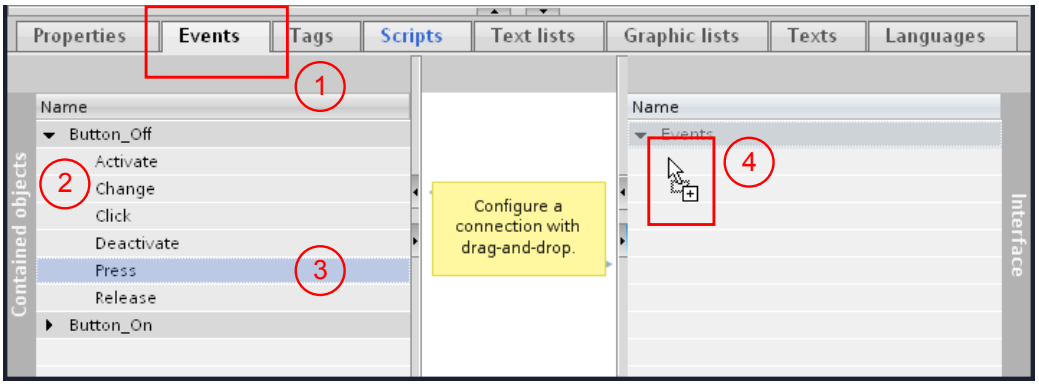
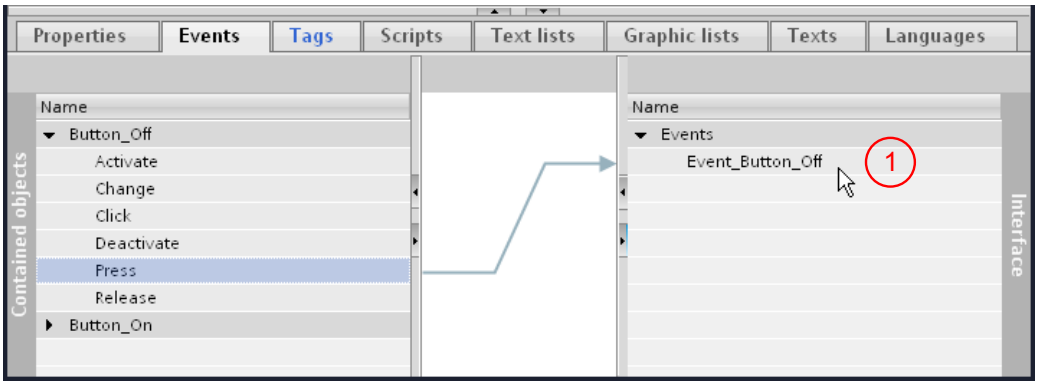
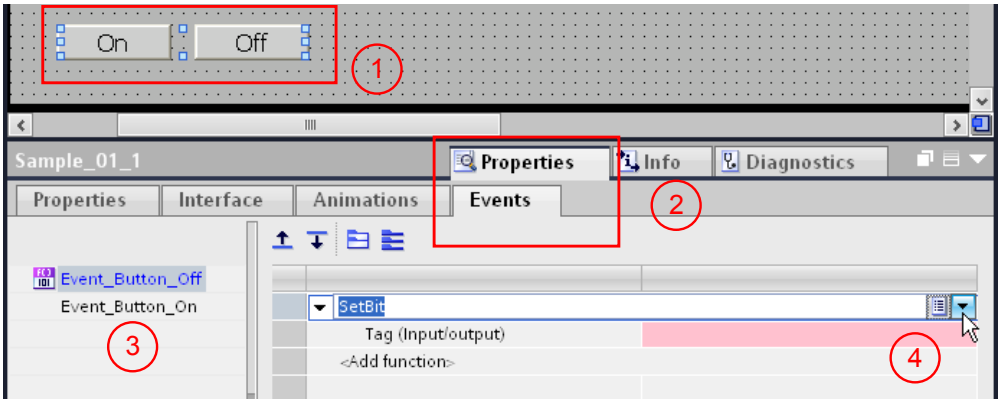


Table 3-1

No.	Action
1.	**Inserting objects** • Open the faceplate editor. • Add two buttons to the workspace. • Assign a label to the buttons in the inspector window in “Events > General”. **In this example:** Text -> On Text -> Off • Assign a name to the buttons in the inspector window in “Events > Miscellaneous”, which the button can be identified by. **In this example:** Button_1 -> Button_On Button_2 -> Button_Off 

No.	Action
2.	**Assigning properties** • Select the “Events” tab in the configuration area (1). • In the “Contained objects” list you open the properties of the button (2). • Select the event for which the function shall be performed (3). • Via drag&drop you configure a connection between the selected “Event” and the “Interface” list (4). **In this example:** Button_Off -> “Press”  • In the “Interface” list of the newly created property, you assign a name you can later use for identifying the property at the faceplate (1). **In this example:** Press -> Event_Button_Off  • Repeat the previously described steps for the second button. The required settings are thus completed.
3.	**Releasing the faceplate** For the performed configurations to become effective, the created faceplate must be released. The faceplate is updated.

No.	Action
4.	**Faceplate view: Assigning a function** Call up the created faceplate in a picture. The subsequent figure shows the newly created faceplate with the two buttons “**On**” and “**Off**” (1). In the “Properties > Events” tab of the faceplate (2), the two created properties “Event_Button_Off” and “Event_Button_On” are displayed (3). From the dropdown list (4), you can assign the familiar functions such as “SetBit” to the events “Event_Button_Off” and “Event_Button_On”. **Note:** The name of the properties was selected so they can easily be assigned to the buttons. 

3.1.2 Sample_02: creating a configurable “IO field”

Creating a configurable “IO field”

Process values shall be input and output at two IO fields.
It shall be possible to assign the HMI tags to the IO fields.

Figure 3-3

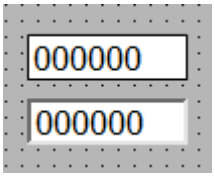
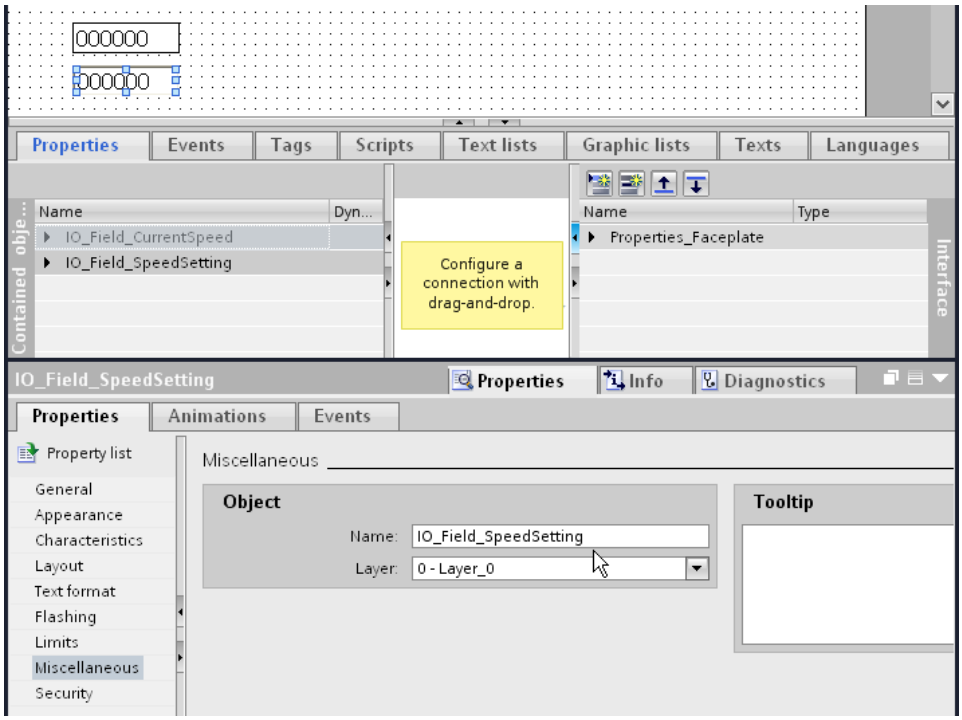
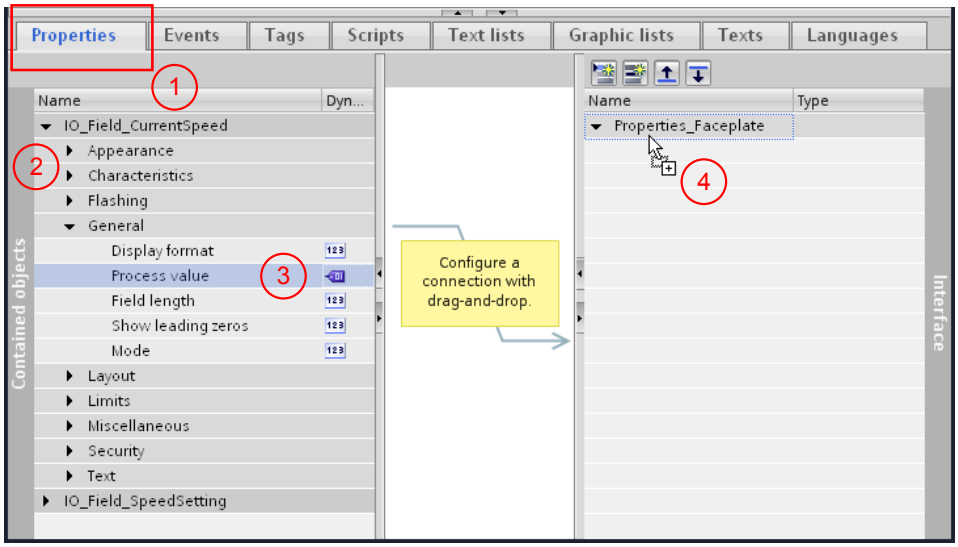
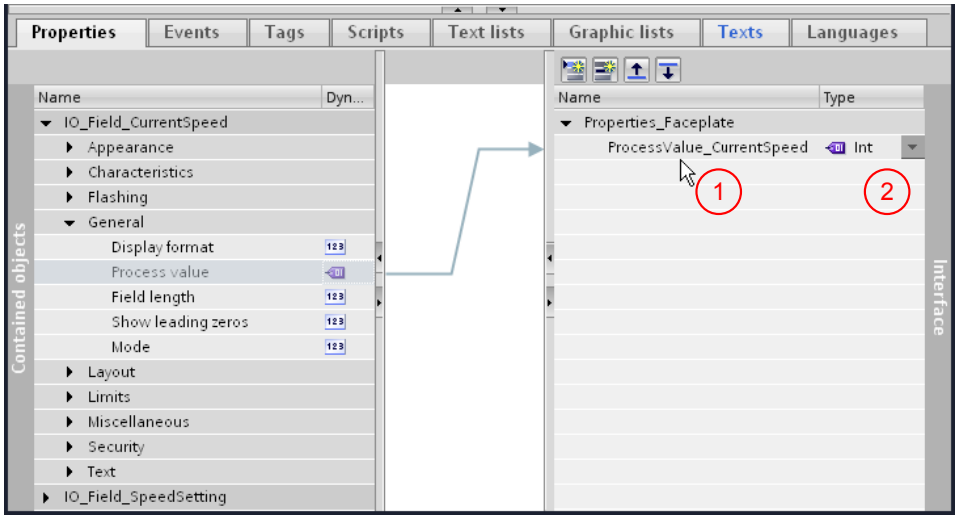
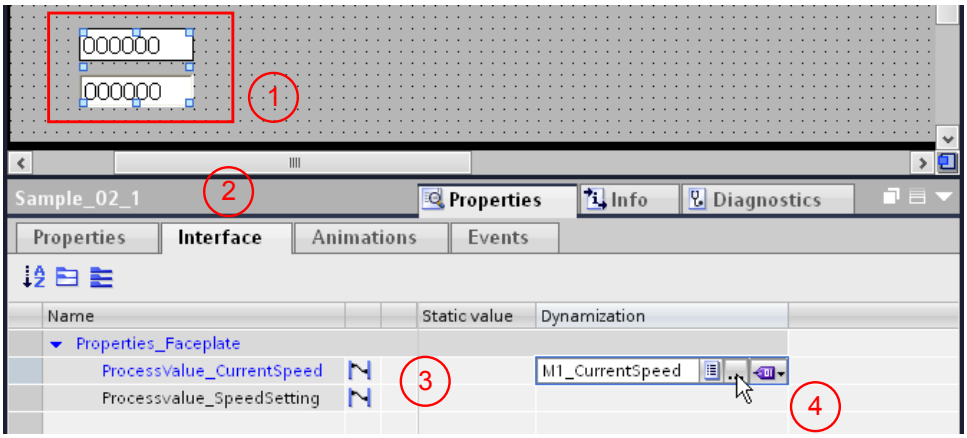


Table 3-2

No.	Action
1.	**Inserting objects** • Open the faceplate editor. • Add two IO fields to the workspace. • In the inspector window in “Events > Miscellaneous” you assign a name to the IO field which the button can be identified by. **In this example:** I/O field_1 -> IO_Field_CurrentSpeed I/O field_2 -> IO_Field_SpeedSetting • In the inspector window, you make further settings for the IO fields such as settings at the “Layout”. 

No.	Action
2.	**Assigning properties** • Select the “Properties” tab (1). • In the “Contained objects” list you open the properties of the IO field (2). • Open the “General” entry from the list and mark the “Process value” property (3). • Via drag&drop you configure a connection between the selected “Property” and the “Interface” list (4). Drag the mouse pointer to an existing category. (The name of the category can be specified/modified individually and may deviate from the names used in the picture). **In this example:** IO_Field_CurrentSpeed -> “Process value” ➔ “Properties_Faceplate”  • In the “Interface” list of the newly created property, you assign a name you can later use for identifying the property at the faceplate (1). **In this example:** Process_value -> ProcessValue_CurrentSpeed • Select the data type from the drop-down list (2). 

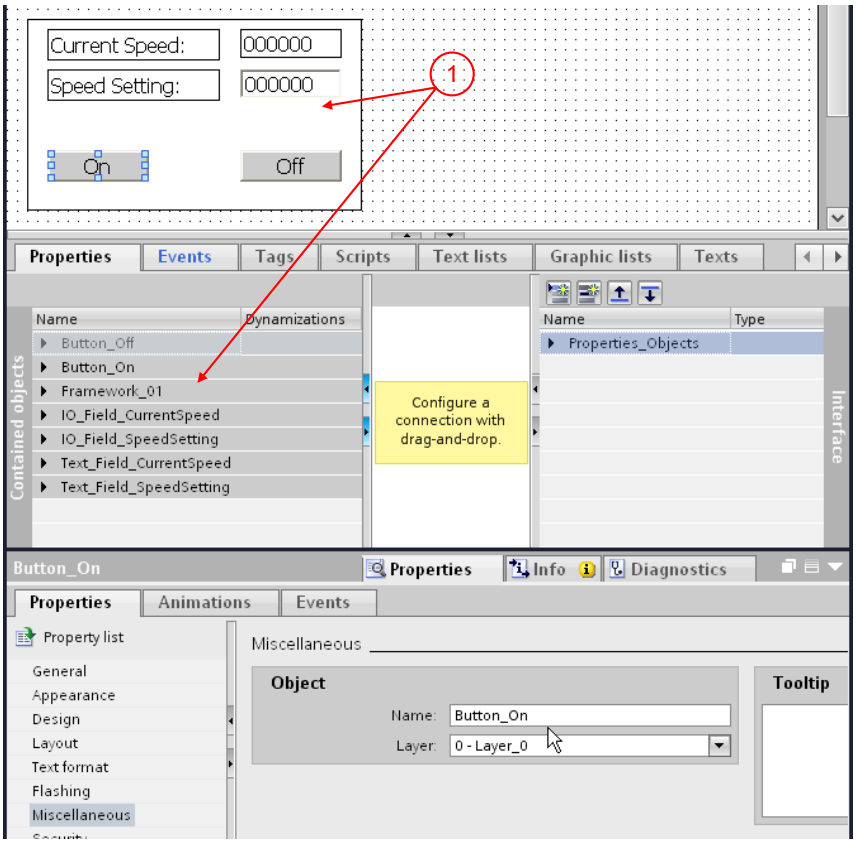
No.	Action
	• Repeat the previously described steps for the second IO field. The required settings are thus completed.
3.	**Releasing the faceplate** For the performed configurations to become effective, the created faceplate must be released. The faceplate is updated.
4.	**Faceplate view; assigning an HMI tag** Call up the created faceplate in a picture. The subsequent figure shows the created faceplate with the two IO fields (1). In the “Properties > Interface” tab of the faceplate (2), the two created properties “ProcessValue_CurrentSpeed” and “ProcessValue_SpeedSettings” are displayed (3). An HMI tag can be assigned to the “ProcessValue_CurrentSpeed” and “ProcessValue_SpeedSettings” properties via the drop-down list (4). **Note:** The name of the properties was selected so they can easily be assigned to the IO fields. 

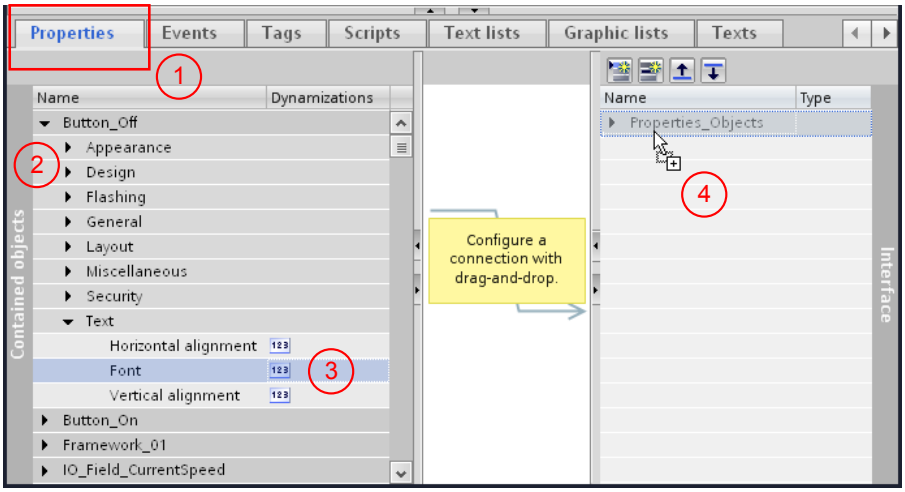
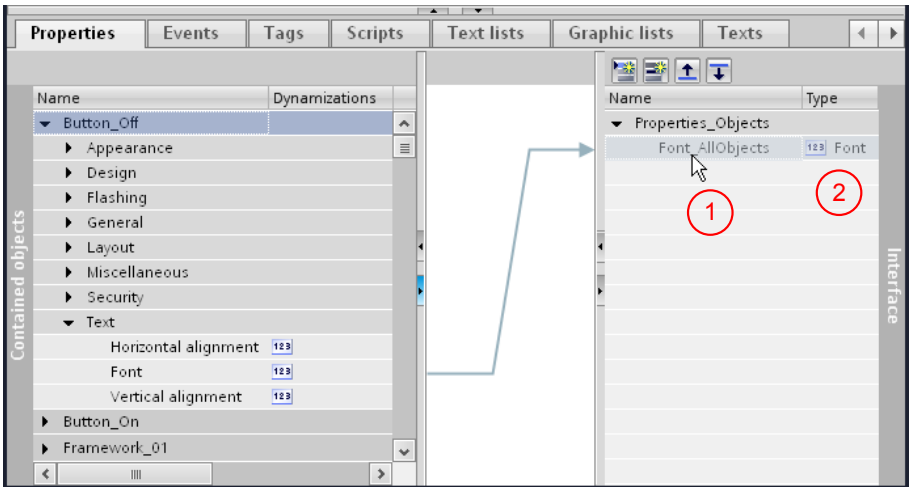
3.1.3 Sample_03: configuring font size at objects

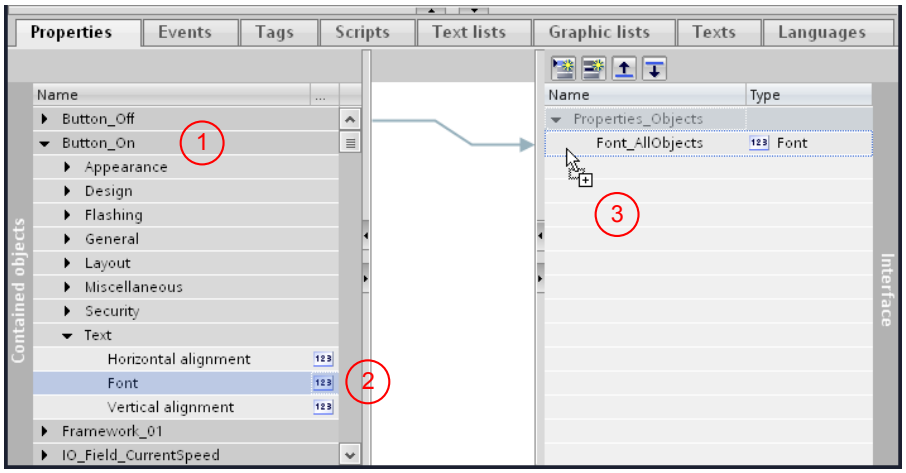
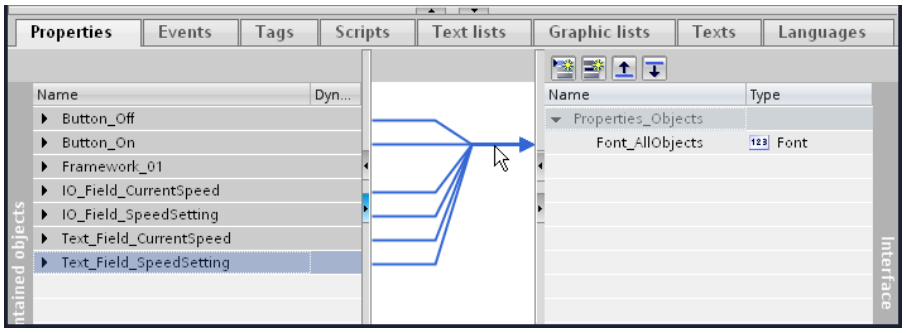
Configurable font size at the object

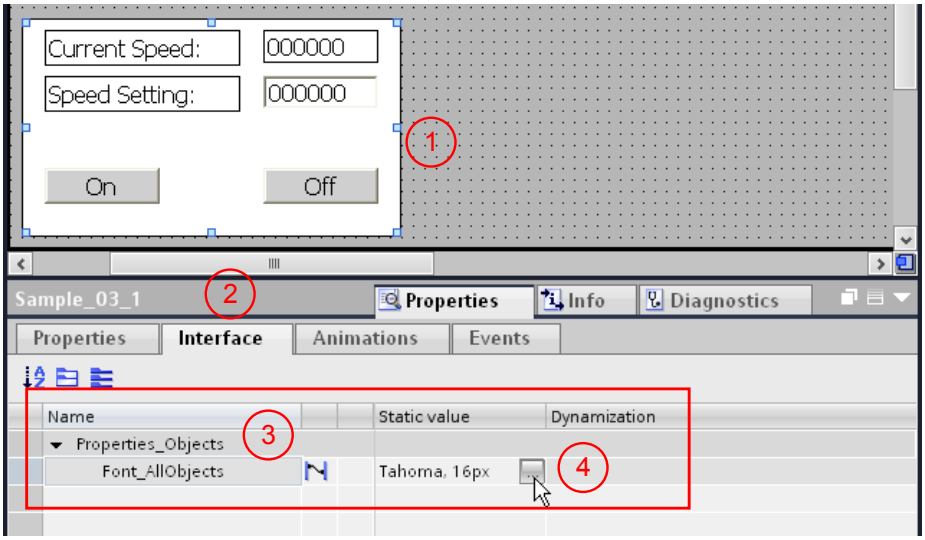
A faceplate shall be used for operator panels of different sizes. The font size of the objects used in the faceplate (two IO fields, two buttons and texts) shall be specified here.

Table 3-3

No.	Action
1.	**Inserting objects** • Open the faceplate editor. • Enter the required objects such as buttons, IO fields etc., into the workspace. For this purpose, look at the table section 1 in chapters 3.1.1 and 3.1.2. • Additionally, add two text fields into the workspace. • In the inspector window in “Events > Miscellaneous” you assign a name to the text field which the button can be identified by. **In this example:** Text field_1 -> Text_Field_CurrentSpeed Text field_2 -> Text_Field_SpeedSetting • In the inspector window, you make further settings for the individual objects, if necessary such as settings at the “Layout”. • For optical reasons, the objects are integrated into a frame (1). 

No.	Action
2.	**Assigning a property for the first object** • Select the “Properties” tab (1). • In the “Contained objects” list you open the properties of the first object (2). • Open the “Text” entry from the list and mark the “Font” property (3). • Configure a connection between the selected “Property” and the “Interface” list via drag&drop (4). Drag the mouse pointer to an existing category. (The name of the category can be specified/modified individually and may deviate from the names used in the picture). **In this example:** Button_Off -> “Text” -> “Font type” ➔ “Properties_Objects”  • In the “Interface” list of the newly created property, you assign a name you can later use for identifying the property at the faceplate (1). **In this example:** Font -> Font_AllObjects • The data type is given automatically (2).  The required settings for the first “object” are thus completed.

No.	Action
3.	**Assigning further properties to the “Interface” list** **Note:** You can assign **several** properties from the “Contained objects” list to **one** property from the “Interface” list. • In the “Contained objects” list you open the properties of the next object (1). • Open the “Text” entry from the list and mark the “Font” property (2). • Configure a connection between the selected “Property” and the “Interface” list via drag&drop. Drag the mouse pointer to the already existing property (3). **In this example:** Button_On -> “Text” -> “Font type” ➔ “Font_AllObjects”  Repeat the previously described steps for all listed elements. The picture below shows the complete assignment of the properties.  The required settings are thus completed.

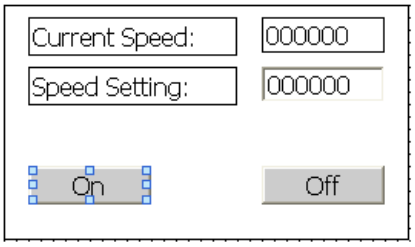
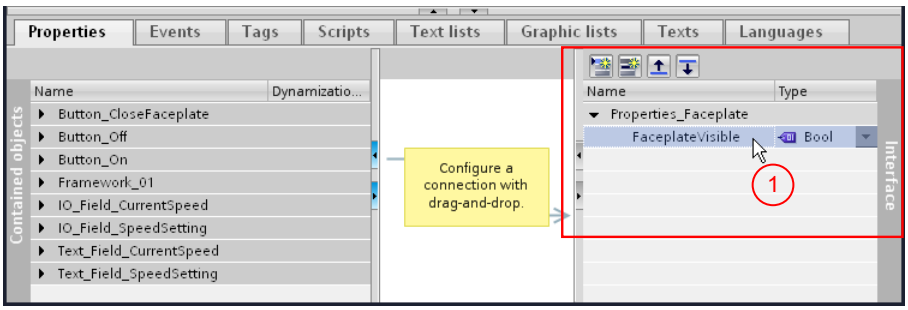
No.	Action
4.	**Releasing the faceplate** For the performed configurations to become effective, the created faceplate must be released. The faceplate is updated.
5.	**Faceplate view; assigning the text format** Call up the created faceplate in a picture. The subsequent figure shows the created faceplate (1). In the “Properties > Interface” tab of the faceplate (2) the created property “Font_AllObjects” is displayed (3). The font can be assigned to the “Font_AllObjects” property via the selection box (4). **In this example:** Font: Tahoma 16px **Note:** The name of the tags was selected so the stored function can be easily recognized. 

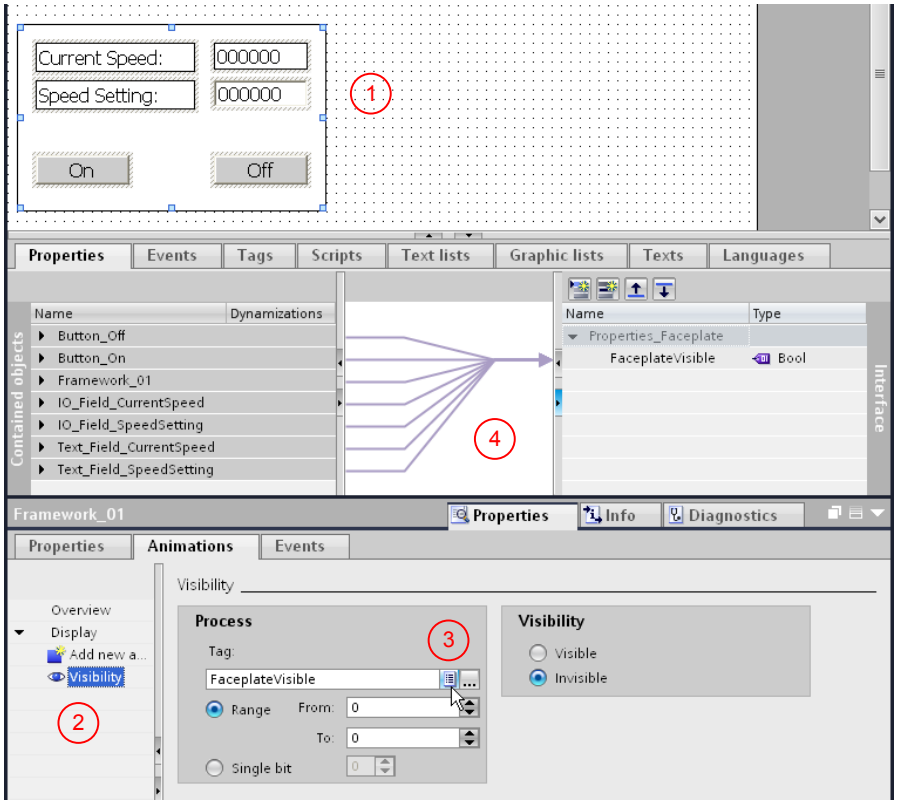
3.1.4 Sample_04: Switching objects invisible

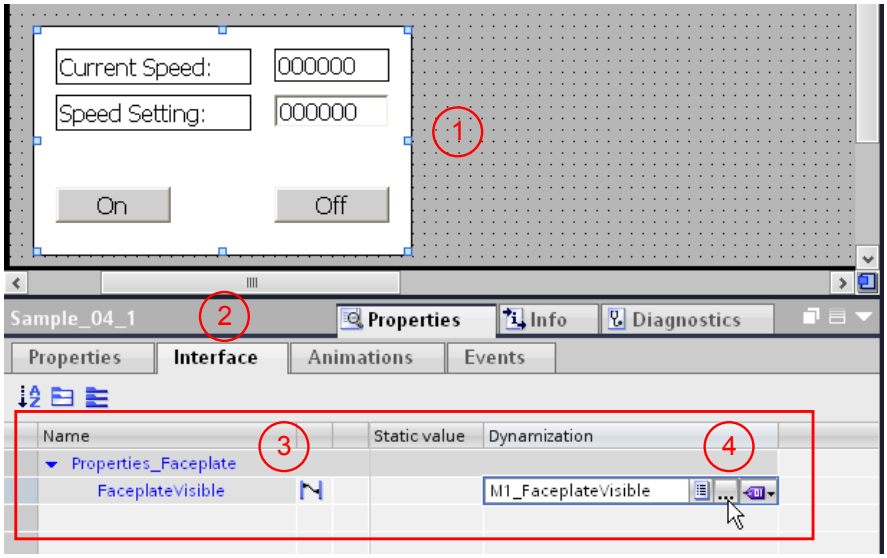
Switching objects invisible

A faceplate cannot be switched invisible directly via a system function. It shall be possible to switch all objects in a faceplate invisible via a function/tag.

Table 3-4

No.	Action
1.	**Inserting objects** • Open the faceplate editor. • Enter all objects such as buttons, IO fields, text fields etc., into the workspace you require. The structure of the faceplate from chapter 3.1.3 is used as a template for this example. 
2.	**Creating tags** • In the “Interface” list of the configuration area, you add a new tag and specify data type “Bool” (1) in the “Type” column via the dropdown menu. **In this example:** Name of the property: FaceplateVisible. Type: Bool. 

No.	Action
3.	**Creating an animation** • In the workspace you select all objects (1). • In the inspector window you open the “Properties > Animations” tab. • Add the “Visibility” animation in the area navigation at “Display” (2). • Selecting a tag: The properties/tags from the “Interface” list are displayed to you via the  symbol (3). Select the “FaceplateVisible” tag from the list. • Selecting the tag automatically connects all marked object with the “Visibility” animation. You can see this in the “graphic” area at the newly created connections (4). • Specify the area in which the faceplate shall be visible. **In this example:** Area: 0 to 0. Visibility: invisible.  The required settings are thus completed.
4.	**Releasing the faceplate** For the performed configurations to become effective, the created faceplate must be released. The faceplate is updated.

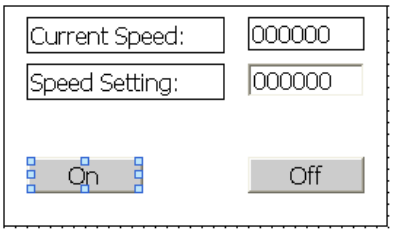
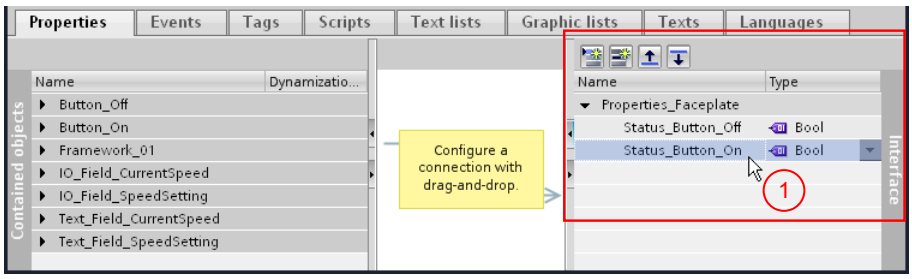
No.	Action
5.	**Faceplate view; tag assignment** Call up the created faceplate in a picture. The subsequent figure shows the created faceplate (1). In the “Properties > Interface” tab of the faceplate (2) the created tag “FaceplateVisible” is displayed (3). Via the dropdown menu you can assign an HMI tag to the “FaceplateVisible” tag via which the faceplate can be switched invisible (4). **Note:** The name of the tags was selected so the stored function can be easily recognized. 

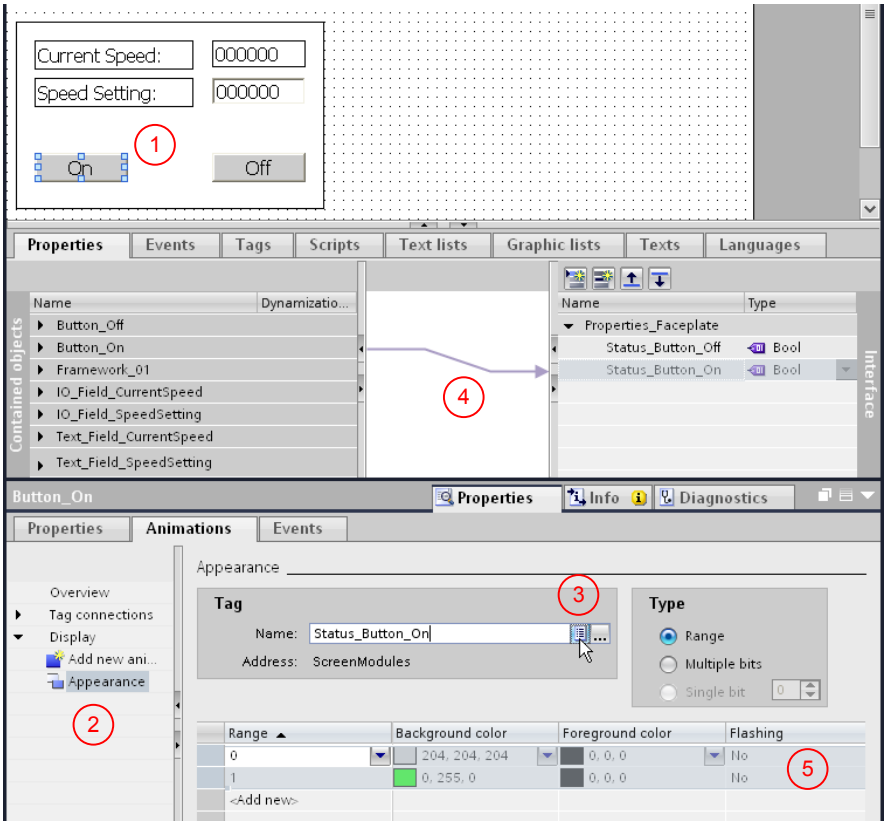
3.1.5 Sample_05: Configuring a color change at an object

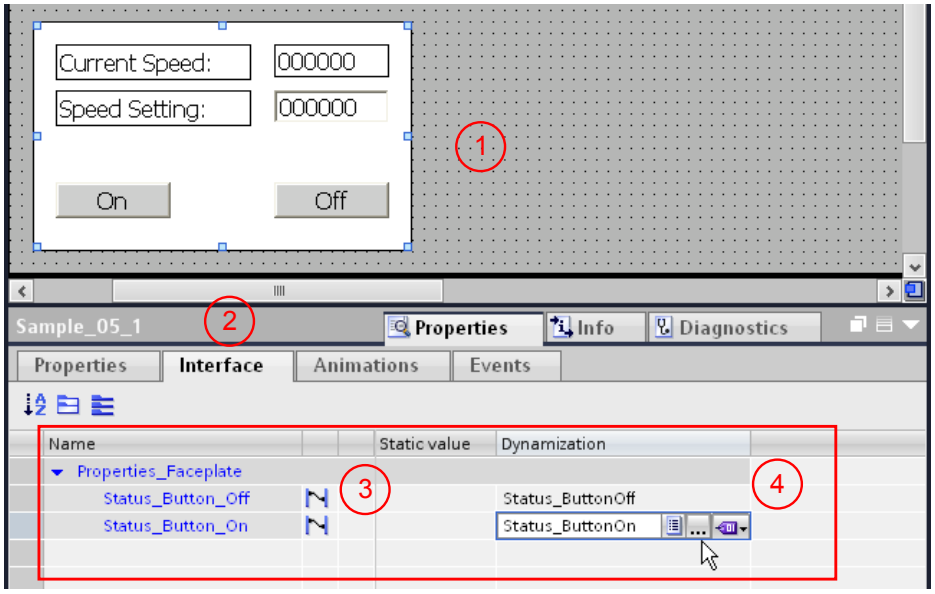
Configuring a color change at an object

Pressing a button shall change the background color of the pressed button.

Table 3-5

No.	Action
1.	**Inserting objects** • Open the faceplate editor. • Enter all objects such as buttons, IO fields, text fields etc., into the workspace you require. The structure of the faceplate from chapter 3.1.4 is used as a template for this example. 
2.	**Creating tags** • In the “Interface” list of the configuration area you add two new tags and via the dropdown menu you specify data type “Bool” in the “Type” column (1). **In this example:** Name of the property: Status_Button_On Status_Button_Off Type: Bool. 

No.	Action
3.	**Creating an animation** • Select the “On” button in the workspace (1). • In the inspector window you open the “Properties > Animations” tab. • Add the “Appearance” animation in the navigation in “Display” (2). • Selecting a tag: The properties/tags from the “Interface” list are displayed to you via the  symbol (3). Select the “Status_Button_On” tag from the list. • Selecting the tag automatically creates a connection between the property of the “On” button (Button_On) in “Contained objects” list and the “Status_Button_On” tag in the “Interface” list. You can see this in the “graphic” area (4). • Specify the area and the background color (5). **In this example:** Value 0: gray Value 1: green  • Repeat the steps for the second “Off” button. The required settings are thus completed.

No.	Action
4.	**Releasing the faceplate** For the performed configurations to become effective, the created faceplate must be released. The faceplate is updated.
5.	**Faceplate view; tag assignment** Call up the created faceplate in a picture. The subsequent figure shows the created faceplate (1). In the “Properties > Interface” tab of the faceplate (2) the created tags “Status_Button_Off” and “Status_Button_On” are displayed (3). Via the dropdown list you can assign an HMI tag to the “Status_Button_Off” and “Status_Button_On”. Depending on the value of the tag, the background color of both buttons changes (4). **Note:** The name of the tags was selected so the stored function can be easily recognized. 

3.1.6 Sample_06: faceplate with user data type (HMI UDT)

Faceplate with a user data type (HMI UDT)

Note A description on the subject of “HMI UDT” is given in the document “Basic knowledge on the topic of HMI faceplates”.

Task

The drive for a fan shall be operated and monitored via a configured control panel. Due to the number of functions, the tag is connected via an HMI UDT.

The following functions and status information shall be executed and evaluated.

Status information

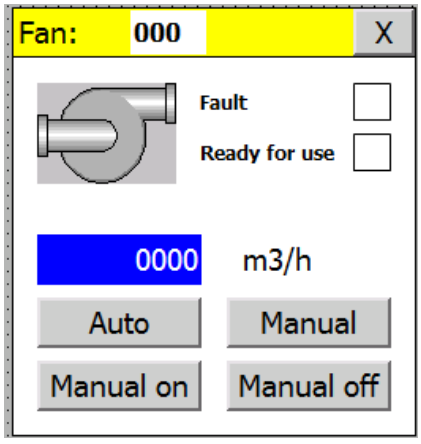
- “Automatic mode” state active
- “Manual mode” state active
- “Manual mode ON” state active
- “Manual mode OFF” state active
- “Ready to use” state
- “Error” state
- “OK” state

Functions

- Output of fan number
- Output value of the current flow rate
- Switchover to automatic mode
- Switchover to manual mode
- Manual mode “On”
- Manual mode “Off”

Faceplate view

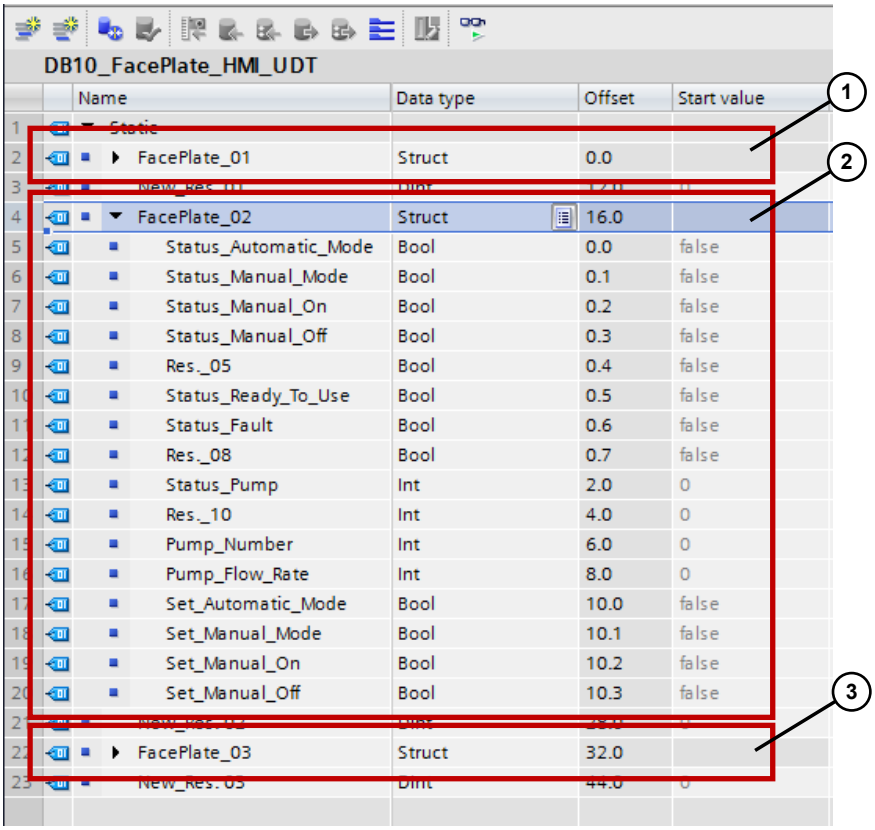
Figure 3-4



Note As of WinCC (TIA Portal) V13 SP1 and when using an S7-1200 or S7-1500 controller, you do no longer need to use a PLC data type.

Creating data block, user data type and HMI tag for the data exchange

Table 3-6

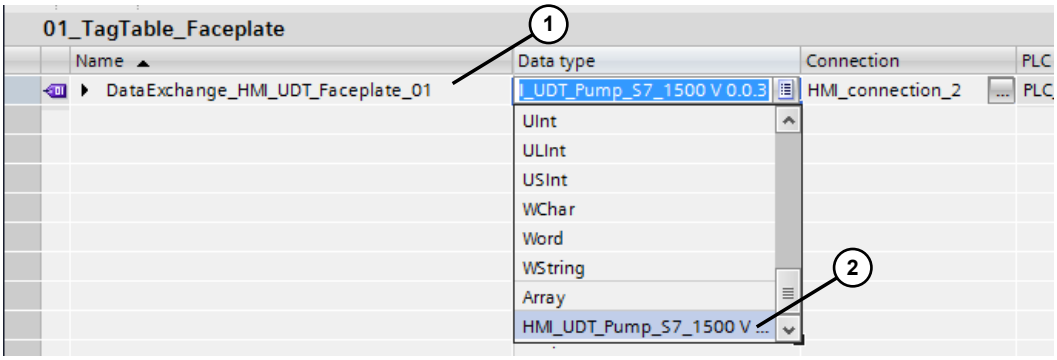
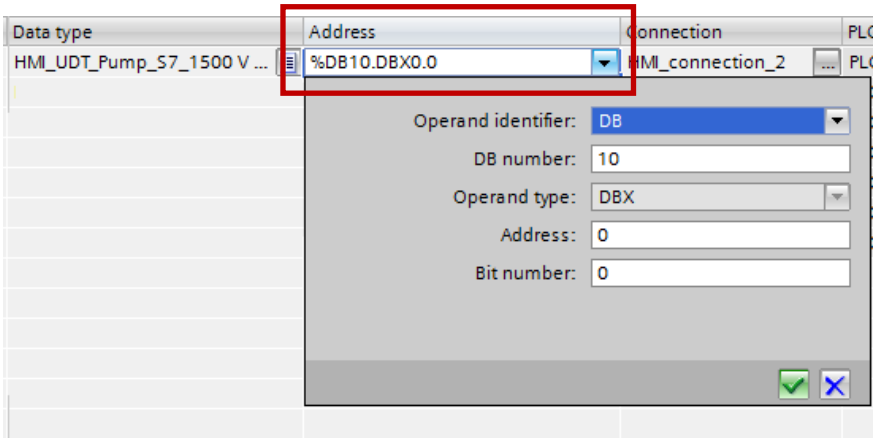
No.	Action
1.	**Creating a data block** For realizing the described Sample_06, the following data block “DB10” was created. The displayed data block contains the structure for three faceplates.  (1) Structure for the first faceplate (2) Structure (opened) for the second faceplate (3) Structure for a third faceplate **Note:** The data block can be extended at will.

No.	Action
2.	**Configuring a user data type** Document “Basic knowledge on the topic of HMI faceplates” describes how to configure a user data type. The figure below shows the created user data type with the respective tags. The structure (structure) corresponds to that of the “DB10” data block. Please ensure, that the user data block is “released” after creating it. Name of the user data type: HMI_UDT_Pump

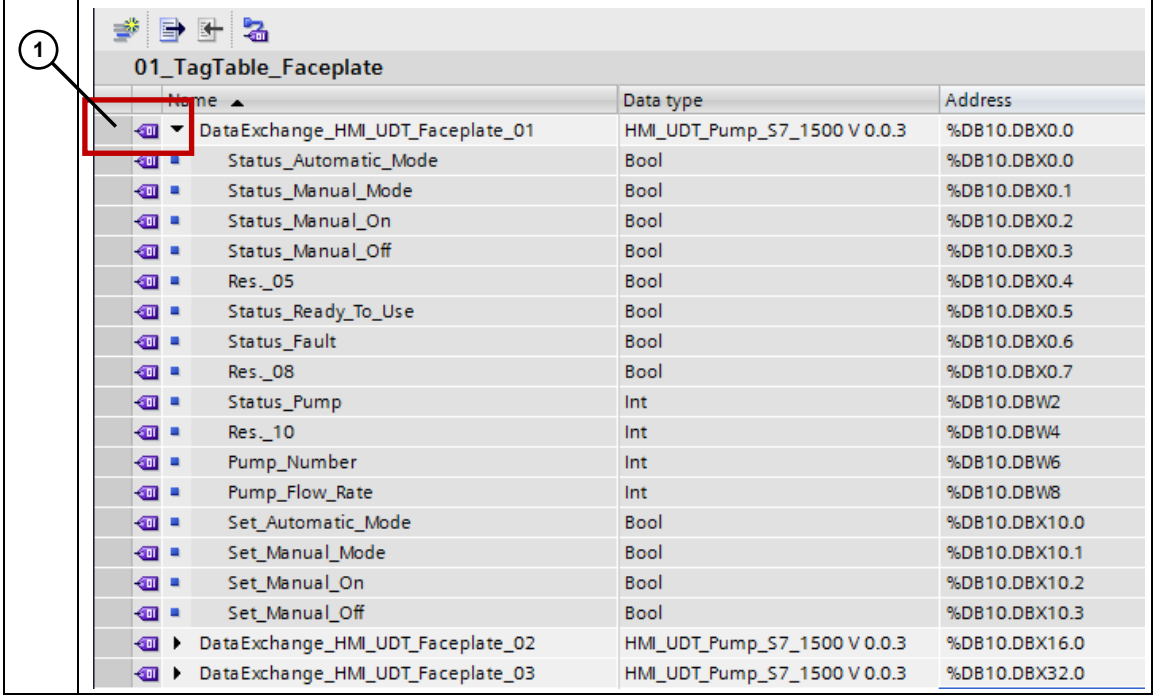
HMI user data types

Communication driver: SIMATIC S7 1500 Device family: Panels / WinCC Runtime Advanced

Name	Communication driver	Data type	Length	Start value	Offset	Bit offset
Status_Automatic_Mode	SIMATIC S7 1500	Bool	1		0	0
Status_Manual_Mode	SIMATIC S7 1500	Bool	1		0	1
Status_Manual_On	SIMATIC S7 1500	Bool	1		0	2
Status_Manual_Off	SIMATIC S7 1500	Bool	1		0	3
Res._05	SIMATIC S7 1500	Bool	1		0	4
Status_Ready_To_Use	SIMATIC S7 1500	Bool	1		0	5
Status_Fault	SIMATIC S7 1500	Bool	1		0	6
Res._08	SIMATIC S7 1500	Bool	1		0	7
Status_Pump	SIMATIC S7 1500	Int	2		2	0
Res._10	SIMATIC S7 1500	Int	2		4	0
Pump_Number	SIMATIC S7 1500	Int	2		6	0
Pump_Flow_Rate	SIMATIC S7 1500	Int	2		8	0
Set_Automatic_Mode	SIMATIC S7 1500	Bool	1		10	0
Set_Manual_Mode	SIMATIC S7 1500	Bool	1		10	1
Set_Manual_On	SIMATIC S7 1500	Bool	1		10	2
Set_Manual_Off	SIMATIC S7 1500	Bool	1		10	3

No.	Action
3.	**Creating an HMI tag** The data exchange between the PLC and the user data type used in the faceplate requires an HMI tag. • Create an HMI tag with a controller connection. The name can be selected by the user (1). • Select the user data type used in the faceplate as the data type (2). In this case “HMI_UDT_Pump_S7_1500”.  **Assigning a start address to the HMI tag** The “DB10” data block displayed in the table section “1” contains the structure for three faceplates. - The start address for the first faceplate is “DB10.DBX**0**.0” - The start address for the second faceplate is “DB10.DBX**16**.0” - The start address for a third faceplate is “DB10.DBX**32**.0”. 

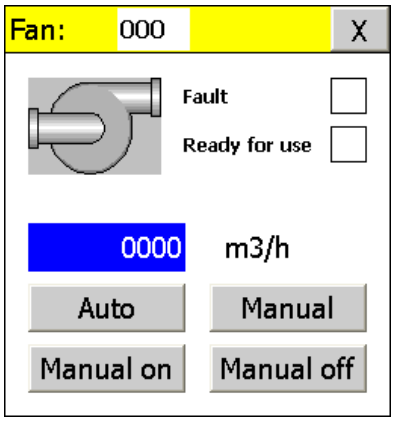
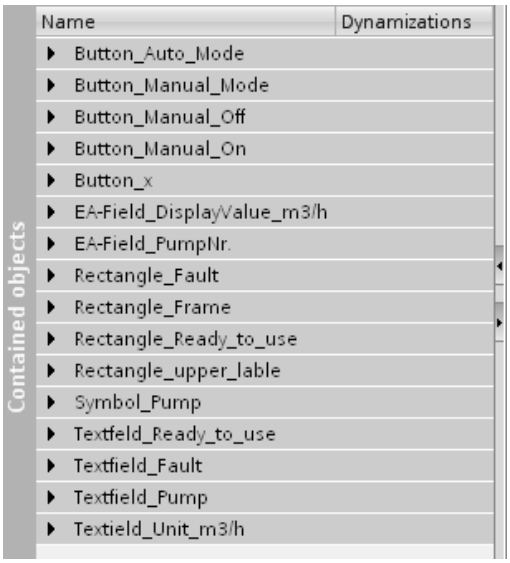
No.	Action
4.	**View of the HMI tag with assigned user data type** The symbol for reducing / expanding the visible entries enables viewing the complete structure of the created HMI tags (1). The structure of the assigned user data type was automatically assigned to HMI tag “DataExchange_HMI_UDT_Faceplate_01”.

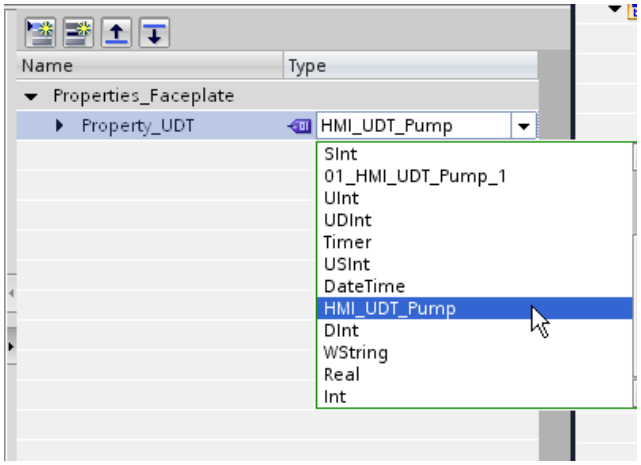
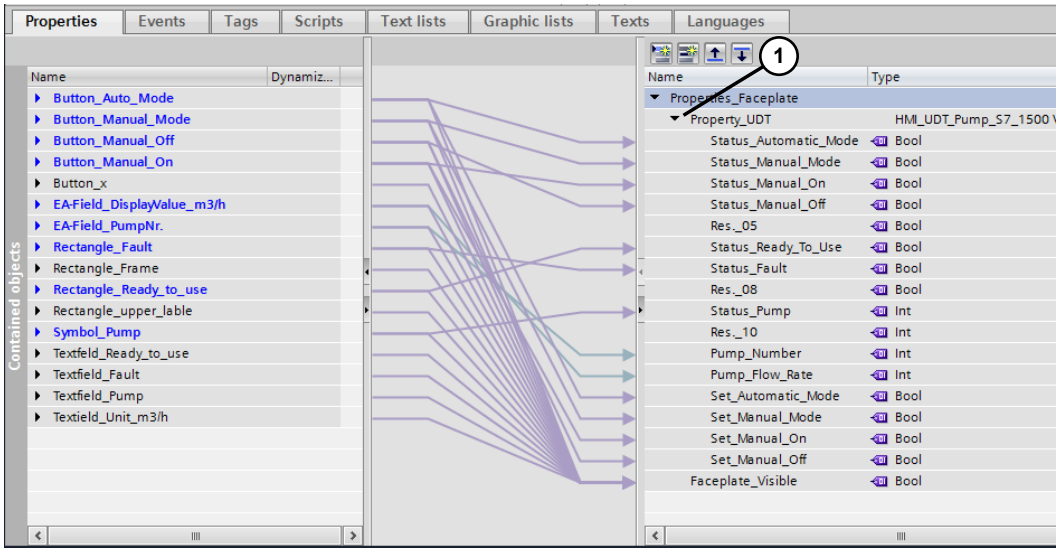


Name	Data type	Address
DataExchange_HMI_UDT_Faceplate_01	HMI_UDT_Pump_S7_1500 V 0.0.3	%DB10.DBX0.0
Status_Automatic_Mode	Bool	%DB10.DBX0.0
Status_Manual_Mode	Bool	%DB10.DBX0.1
Status_Manual_On	Bool	%DB10.DBX0.2
Status_Manual_Off	Bool	%DB10.DBX0.3
Res._05	Bool	%DB10.DBX0.4
Status_Ready_To_Use	Bool	%DB10.DBX0.5
Status_Fault	Bool	%DB10.DBX0.6
Res._08	Bool	%DB10.DBX0.7
Status_Pump	Int	%DB10.DBW2
Res._10	Int	%DB10.DBW4
Pump_Number	Int	%DB10.DBW6
Pump_Flow_Rate	Int	%DB10.DBW8
Set_Automatic_Mode	Bool	%DB10.DBX10.0
Set_Manual_Mode	Bool	%DB10.DBX10.1
Set_Manual_On	Bool	%DB10.DBX10.2
Set_Manual_Off	Bool	%DB10.DBX10.3
DataExchange_HMI_UDT_Faceplate_02	HMI_UDT_Pump_S7_1500 V 0.0.3	%DB10.DBX16.0
DataExchange_HMI_UDT_Faceplate_03	HMI_UDT_Pump_S7_1500 V 0.0.3	%DB10.DBX32.0

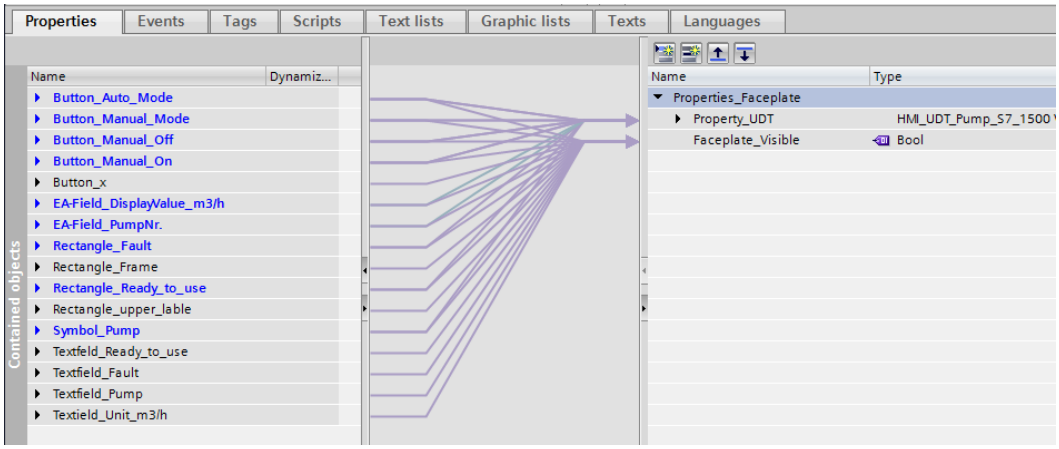
Integrating a user data type (HMI UDT) into a faceplate

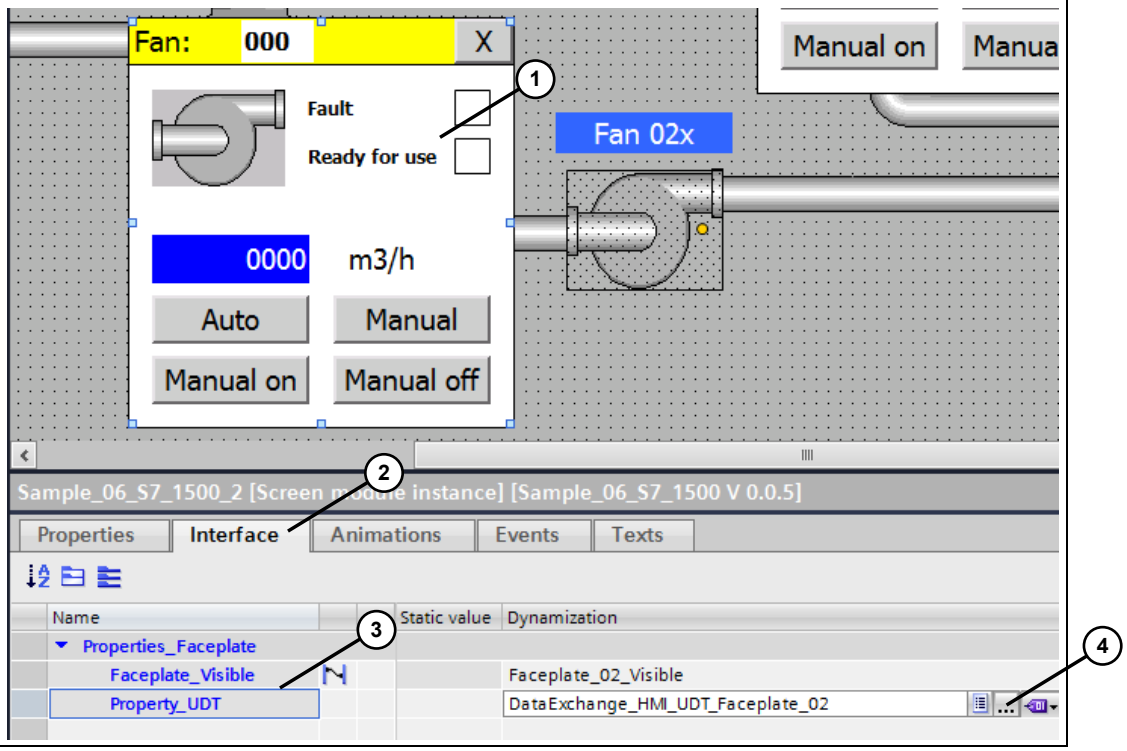
Table 3-7

No.	Action
1.	**Inserting objects** • Open the faceplate editor. • Enter all objects such as buttons, IO fields, text fields etc., into the workspace required according to the plant description. The picture shows the completed faceplate. 
2.	**Adjusting the names of the inserted objects** Adjust the names of the used objects in the "**Properties > Contained objects**" tab so you can easily identify these objects. For this purpose, look at the previously described chapters 3.1.1 to 3.1.5. The figure below shows the edited names in the "Contained objects" list. 

No.	Action
3.	**Using the user data type in the faceplate** In the "**Properties > Interface**" tab of the configuration area you add a tag and select the created user data type as a "Type" via the dropdown menu. **In this example:** Name of the user data type: HMI_UDT_Pump 
4.	**View of the configured faceplate** After assigning the user data type in the "Interface" list, the structure of the user data type is displayed to you via the created tag. The symbol for reducing / expanding the visible entries enables viewing the complete structure of the created tags (1). The figure below shows the already created configurations between the "Contained objects" list and the "Interface" list.  The details for assigning the individual configurations are described below. The procedure is principally the same as described previously for chapters 3.1.1 to 3.1.5.

No.	Action
5.	**Configuring a color change at objects** A color change has been stored at the following objects (animation). • Button: Button_Auto_Mode • Button: Button_Manual_Mode • Button: Button_Manual_On • Button: Button_Manual_Off • Rectangle: Rectangle_Ready_to_use • Rectangle: Rectangle_Fault • Display: Symbol_Pump The procedure is described in chapter 3.1.5. The allocation can be controlled by means of the graphic representation.
6.	**Creating configurable IO fields** The following IO fields can be configured. • Input/output field: IO-Field_DisplayValue_m3/h • Output field: IO-Field_PumpNr The procedure is described in the chapter 3.1.2. The assignment via drag&drop in this case occurs directly to the respective tag at the interface. **Example:** Input/output field: IO-Field_DisplayValue_m3/h
7.	**Assigning an event to the buttons** An event has been configured at the following objects. • Button: Button_Auto_Mode • Button: Button_Manual_Mode • Button: Button_Manual_On • Button: Button_Manual_Off • Button: Button_x The procedure is described in the chapter.
8.	**Switching objects invisible** For the enclosed application example, there is the option to switch the used faceplate invisible. The procedure is described in the chapter 3.1.4.

No.	Action
9.	**View of the completed faceplate in the faceplate editor** The “Interface” list only contains the tags for the user data type and the tag for switching the faceplate invisible. **In this example:** Property_UDT -> HMI_UDT_Pump_S7_1500 Faceplate_Visible -> Bool The “Connections” shown in the graphic area are now more extensive in this example application. The reference behind it should now no longer be “confusing” for you. 
10.	**Releasing the faceplate** For the performed configurations to become effective, the created faceplate must be released. The faceplate is updated.

No.	Action
11.	**Faceplate view, assigning tags to the interface** Call up the created faceplate in a picture. The subsequent figure shows the created faceplate (1). Using the “User data type” reduces the number of tags to be configured to a minimum. In the “Properties > Interface” tab of the faceplate (2), the two created tags “Property_UDT“ and “Faceplate_Visible” are displayed (3). An HMI tag can be assigned to both tags via the dropdown list (4). Assign the previously described HMI tag “DataExchange_HMI_UDT_Faceplate_02” to the “Property_UDT” tag. **Note:** The name of the tags was selected so the stored function can be easily recognized. 

3.1.7 Sample_07: Faceplate with a PLC data type

Note A description on the subject of “PLC data types” is given in the document “Basic knowledge on the topic of HMI faceplates”.

Task Description

The task corresponds to the example from the previous chapter 3.1.6.

The drive for a fan shall be operated and monitored via a configured control panel. Due to the number of functions, the tag is connected via a PLC data type.

Status information

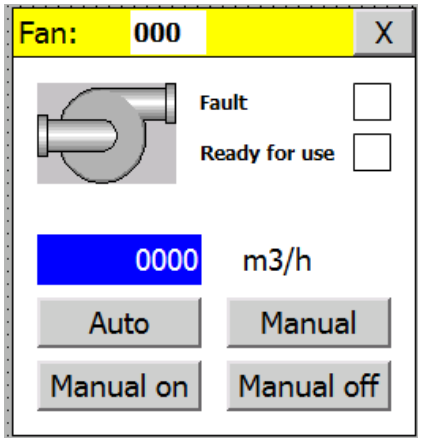
- “Automatic mode” state active
- “Manual mode” state active
- “Manual mode ON” state active
- “Manual mode OFF” state active
- “Ready to use” state
- “Error” state
- “OK” state

Functions

- Output of fan number
- Output value of the current flow rate
- Switchover to automatic mode
- Switchover to manual mode
- Manual mode “On”
- Manual mode “Off”

Faceplate view

Figure 3-5



Creating PLC data types

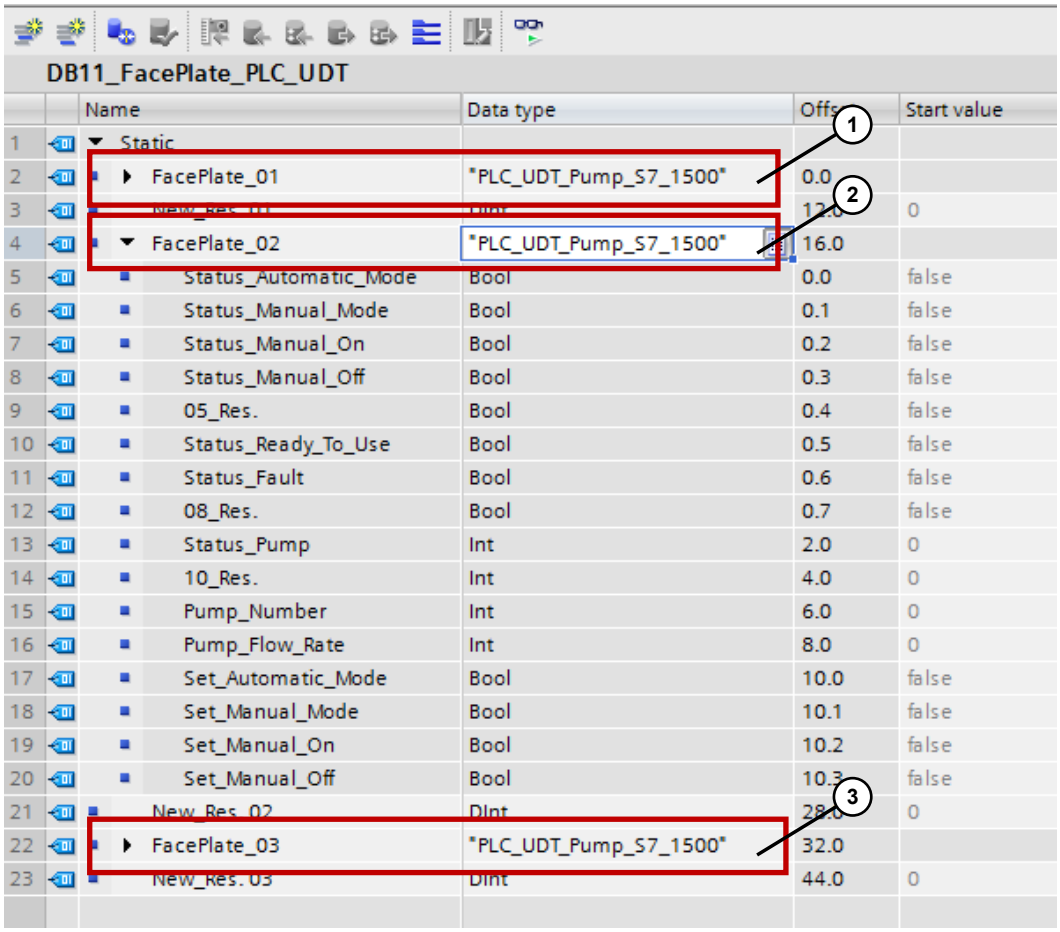
Table 3-8

No.	Action
1.	**Creating PLC data types** The following structure was used for realizing the task. The structure is the same for all faceplates. Document “Basic knowledge on the topic of HMI faceplates” describes how to configure a PLC data type. **Notes** • Please note that you are using an S7-1200 or S7-1500 controller. • The PLC data type can be extended at will. PLC_UDT_Pump_S7_1500

	Name	Data type	Default value	Accessible f...	Visible in ...
1	Status_Automatic_Mode	Bool	false	☑	☑
2	Status_Manual_Mode	Bool	false	☑	☑
3	Status_Manual_On	Bool	false	☑	☑
4	Status_Manual_Off	Bool	false	☑	☑
5	05_Res.	Bool	false	☑	☑
6	Status_Ready_To_Use	Bool	false	☑	☑
7	Status_Fault	Bool	false	☑	☑
8	08_Res.	Bool	false	☑	☑
9	Status_Pump	Int	0	☑	☑
10	10_Res.	Int	0	☑	☑
11	Pump_Number	Int	0	☑	☑
12	Pump_Flow_Rate	Int	0	☑	☑
13	Set_Automatic_Mode	Bool	false	☑	☑
14	Set_Manual_Mode	Bool	false	☑	☑
15	Set_Manual_On	Bool	false	☑	☑
16	Set_Manual_Off	Bool	false	☑	☑

Integrating a PLC data type into a data block

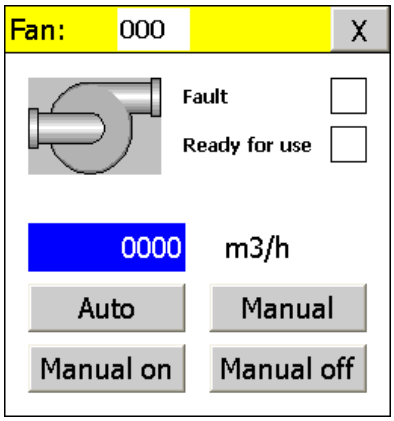
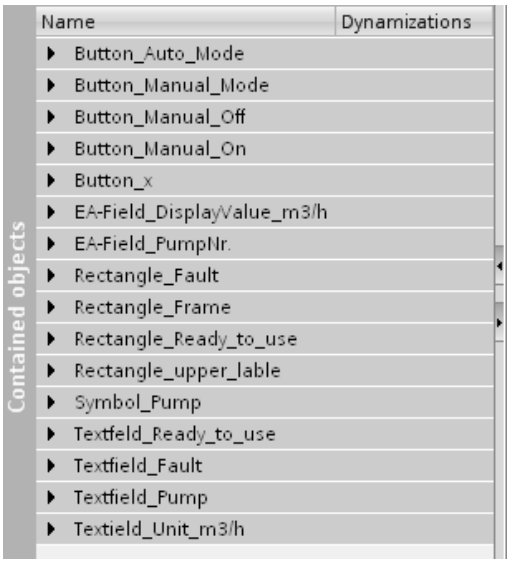
Table 3-9

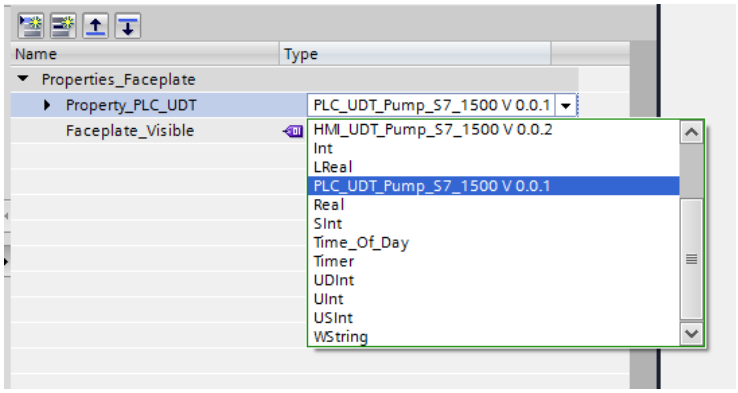
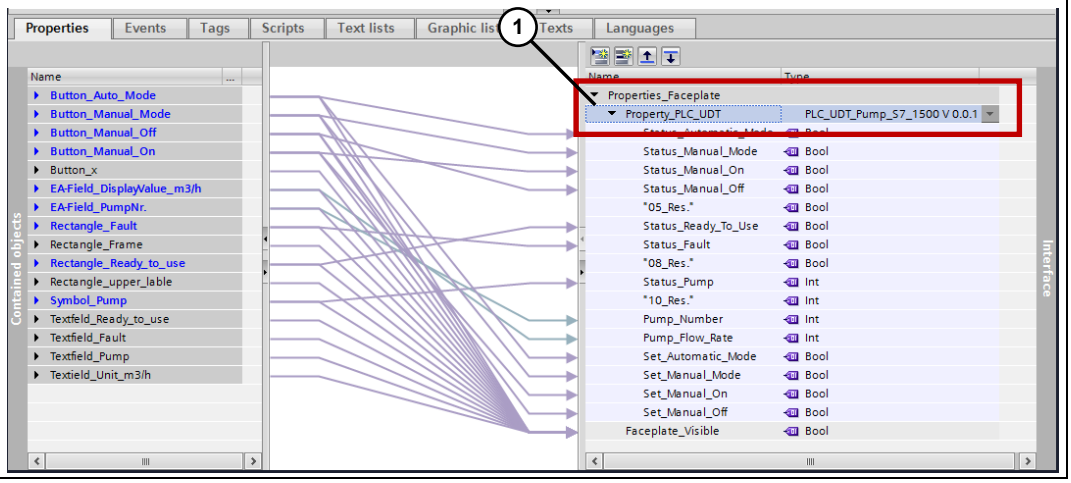
No.	Action
1.	**Creating a data block** For realizing the described task, the following data block “DB11” was created. The data block contains the data for three faceplates. For each faceplate, a tag was stored and then assigned to the “PLC data type” tag.  DB11_FacePlate_PLC_UDT (1) (2) (3) (1) Structure for the first faceplate (2) Structure (opened) for the second faceplate (3) Structure for a third faceplate **Note:** The data block can be extended at will.

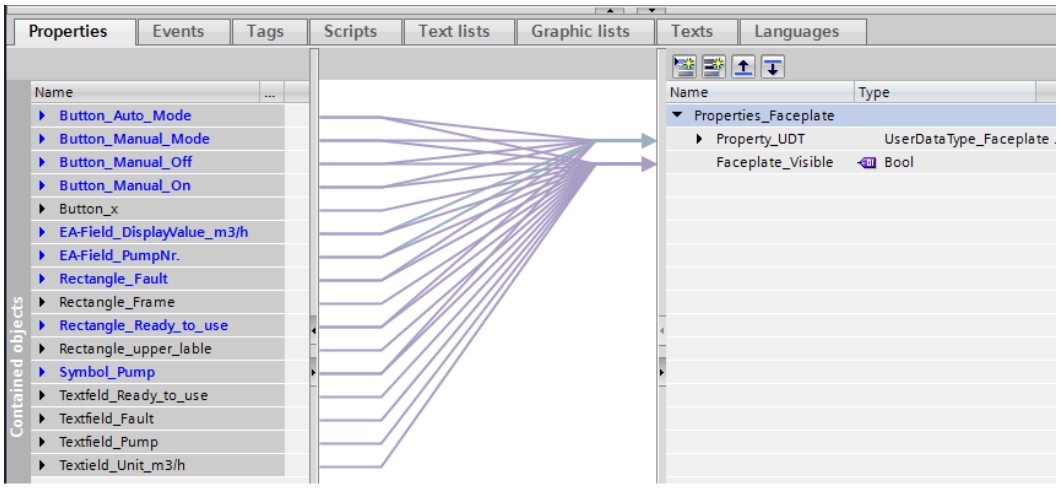
	Name	Data type	Offs	Start value
1	Static			
2	FacePlate_01	"PLC_UDT_Pump_S7_1500"	0.0	
3	New_Res_01	DInt	12.0	0
4	FacePlate_02	"PLC_UDT_Pump_S7_1500"	16.0	
5	Status_Automatic_Mode	Bool	0.0	false
6	Status_Manual_Mode	Bool	0.1	false
7	Status_Manual_On	Bool	0.2	false
8	Status_Manual_Off	Bool	0.3	false
9	05_Res.	Bool	0.4	false
10	Status_Ready_To_Use	Bool	0.5	false
11	Status_Fault	Bool	0.6	false
12	08_Res.	Bool	0.7	false
13	Status_Pump	Int	2.0	0
14	10_Res.	Int	4.0	0
15	Pump_Number	Int	6.0	0
16	Pump_Flow_Rate	Int	8.0	0
17	Set_Automatic_Mode	Bool	10.0	false
18	Set_Manual_Mode	Bool	10.1	false
19	Set_Manual_On	Bool	10.2	false
20	Set_Manual_Off	Bool	10.3	false
21	New_Res_02	DInt	28.0	0
22	FacePlate_03	"PLC_UDT_Pump_S7_1500"	32.0	
23	New_Res_03	DInt	44.0	0

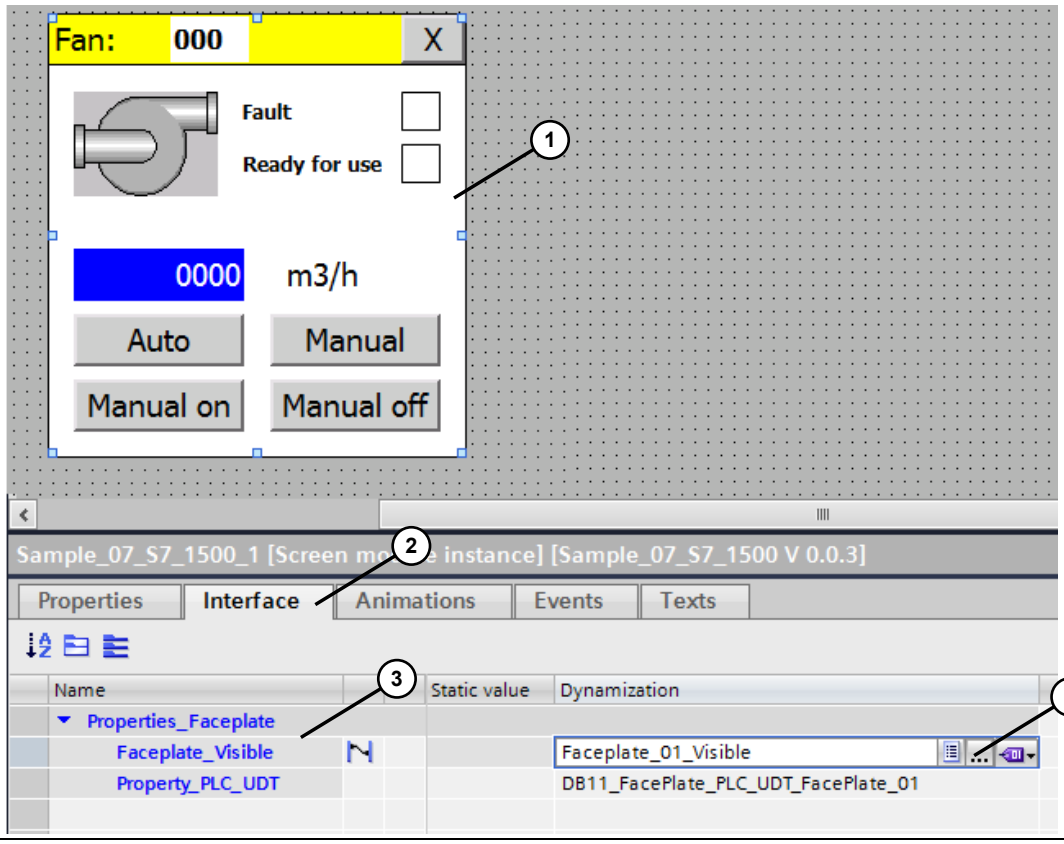
Integrating a PLC data type into a faceplate

Table 3-10

No.	Action
1.	**Inserting objects** • Open the faceplate editor. • Enter all objects such as buttons, IO fields, text fields etc., into the workspace required according to the plant description. The attached faceplate from example “Sample_06” serves as a template. 
2.	**Adjusting the names of the inserted objects** Adjust the names of the used objects in the “**Properties > Contained objects**” tab so you can easily identify these objects. For this purpose, look at the previously described chapters 3.1.1 to 3.1.5. The figure below shows the edited names in the “Contained objects” list. 

No.	Action
3.	**Using PLC data types in the faceplate** In the "**Properties > Interface**" tab of the configuration area you add a tag and select the created PLC data type as a "Type" via the dropdown menu. **In this example:** Name of the user data type PLC_UDT_Ump_S7_1500 
4.	**View of the configured faceplate** After assigning the user data type in the "Interface" list, the structure of the user data type is displayed to you via the created tag. The symbol for reducing / expanding the visible entries enables viewing the complete structure of the created tags (1). The figure below shows the already created configurations between the "Contained objects" list and the "Interface" list. 

No.	Action
5.	**Details for assigning the individual configurations** The configurations of the following points corresponds to those of the previous examples ("Sample_06") and are not further discussed here. • Configuring a color change at objects (Link) • Creating configurable IO fields (Link) • Assigning an event to the buttons (Link) • Switching objects invisible (Link)
6.	**View of the completed faceplate in the faceplate editor** The "Interface" list only contains the tags for the PLC data type and the tag for switching the faceplate invisible. **In this example:** Property_UDT -> PLC_UDT_Ump_S7_1500 Faceplate_Visible -> Bool The "Connections" shown in the graphic area are now more extensive in this example application. The reference behind it should now no longer be "confusing" for you. 
7.	**Releasing the faceplate** For the performed configurations to become effective, the created faceplate must be released. The faceplate is updated.

No.	Action
8.	**Faceplate view, assigning tags to the interface** Call up the created faceplate in a picture. The subsequent figure shows the created faceplate (1). Using the PLC data type reduces the number of tags to be configured to a minimum. • In the “Properties > Interface” tab of the faceplate (2), the two created tags “Property_UDT” and “Faceplate_Visible” are displayed (3). An HMI tag can be assigned to both tags via the dropdown list (4). • Assign HMI tag “DB11_FacePlate_PLC_UDT_PLC_FacePlate_01” to the “Property_PLC_UDT” tag. **Note:** The name of the tags was selected so the stored function can be easily recognized. 

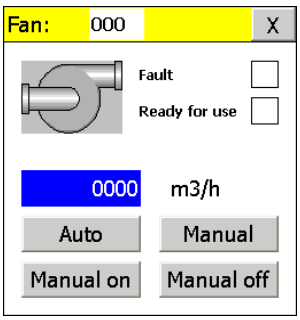
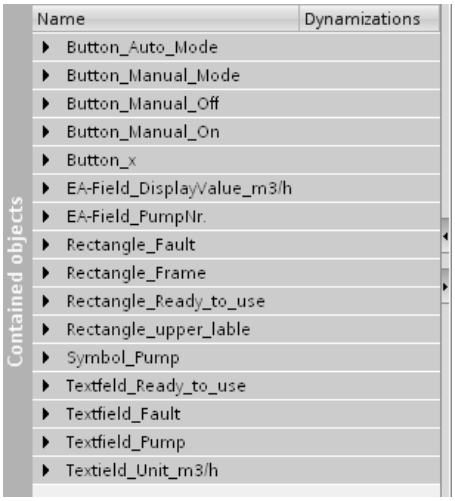
3.1.8 Sample_08: user authorizations at the faceplate

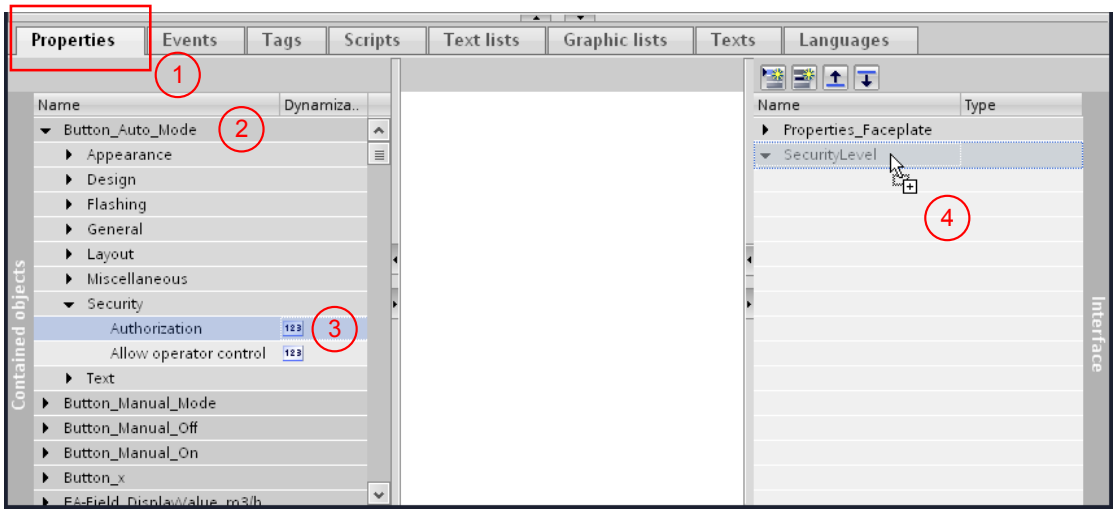
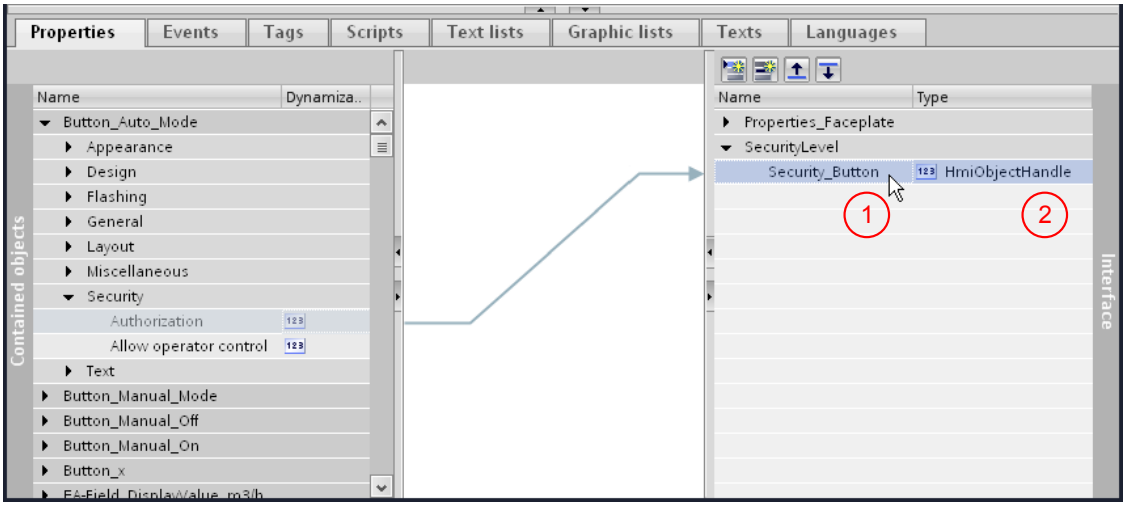
Assigning the user authorization to a faceplate

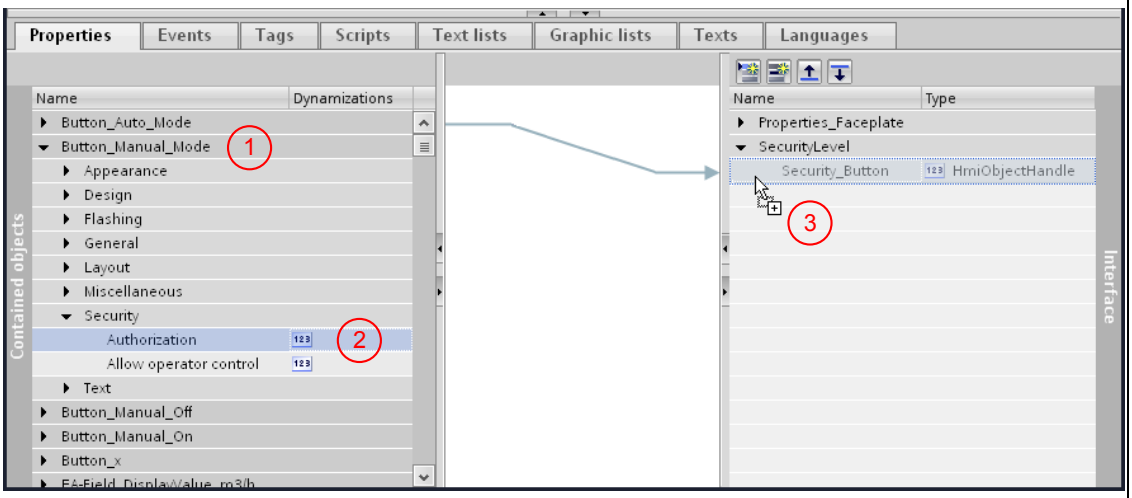
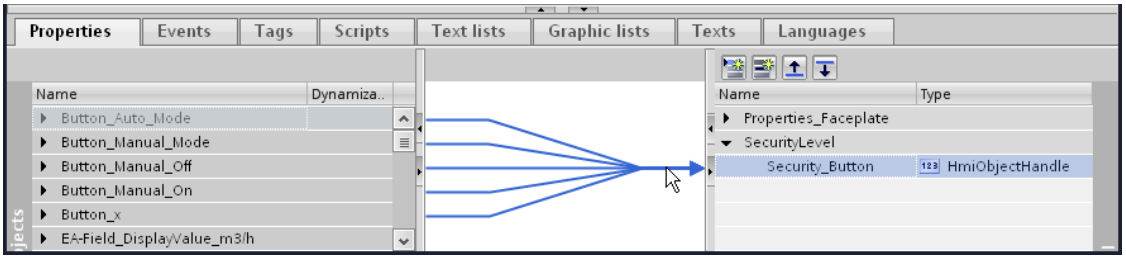
Switching from automatic mode to manual mode as well as the subsequent operation in manual mode shall only be performed by an authorized person (service staff).

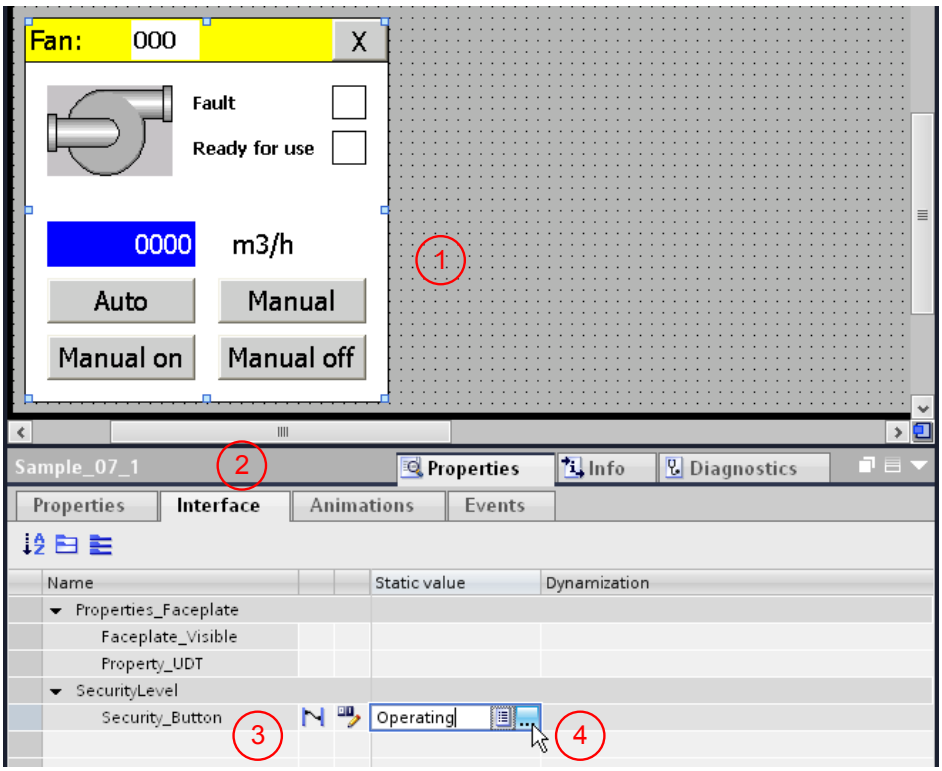
A user administration is required for this.

Table 3-11

No.	Action
1.	**Inserting objects** • Open the faceplate editor. • Enter all objects such as buttons, IO fields, text fields etc., into the workspace you require. The attached faceplate from example “Sample_07” serves as a template. 
2.	**Adjusting the names of the inserted objects** Adjust the names of the used objects in the “**Properties > Contained objects**” tab so you can easily identify these objects. For this purpose, look at the previously described chapters 3.1.1 to 3.1.5. The figure below shows the edited names in the “Contained objects” list. 

No.	Action
3.	**Assign a property to the first button in the “Interface” list** • Select the “Properties” tab (1). • In the “Contained objects” list you open the properties of the “Button_Auto_Mode” button (2). • Open the “Security” entry from the list and mark the “Authorization” property (3). • Via drag&drop you configure a connection between the selected “Property” and the “Interface” list (4). Drag the mouse pointer to an existing category. (In this example a further category named “SecurityLevel” was created).  • In the “Interface” list of the newly created property, you assign a name you can later use for identifying the property at the faceplate (1). **In this example:** Authorization -> Security_Button • The data type is given automatically (2).  The required settings for the first button are thus completed.

No.	Action
4.	**Assign further buttons in the “Interface” list** **Note:** You can assign **several** properties from the “Contained objects” list to **one** property from the “Interface” list. However, you can also store a separate “Property” in the “Interface” list for each button. In this example, the properties of all five buttons are assigned to a “Property” in “Interface” list. • In the “Contained objects” list you open the properties of the button (1). • Open the “Security” entry from the list and mark the “Authorization” property (2). • Configure a connection between the selected “Property” and the “Interface” list via drag&drop. Drag the mouse pointer to the already existing “Security_Button” property (3).  • Repeat the previously described steps for all listed elements. The picture below shows the complete assignment of the properties.  The required settings are thus completed.

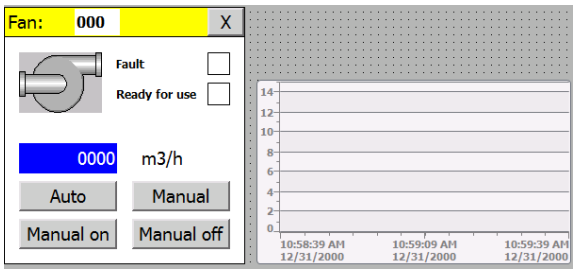
No.	Action
5.	**Releasing the faceplate** For the performed configurations to become effective, the created faceplate must be released. The faceplate is updated.
6.	**Faceplate view; assigning the authorization** Call up the created faceplate in a picture. The subsequent figure shows the created faceplate (1). In the “Properties > Interface” tab of the faceplate (2) the created property “Security_Button” is displayed (3). The respective authorization can be assigned to the “Security_Button” property via the selection box (4). **In this example:** Operating **Note:** The name of the tags was selected so the stored function can be easily recognized. 

3.1.9 Sample_09: Using a trend view in a faceplate

Using a trend view in a faceplate

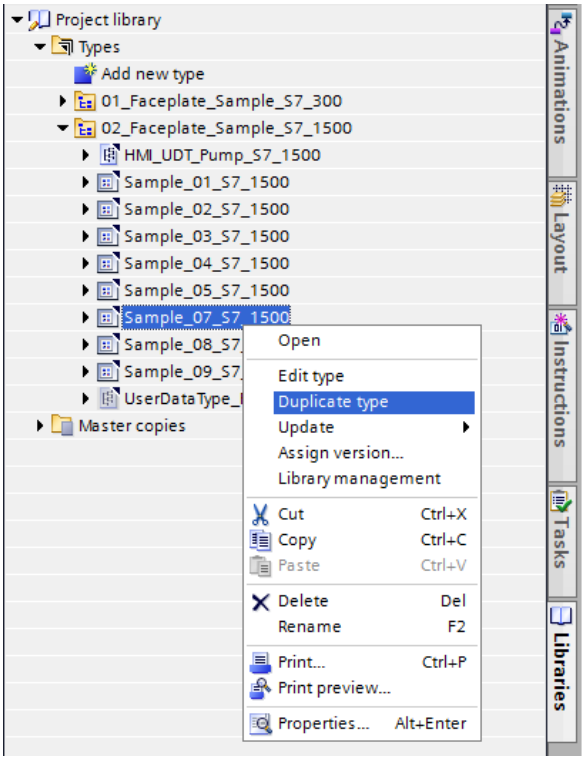
The current flowrate of air over a period of one minute shall additionally be output via the trend view in the control station.

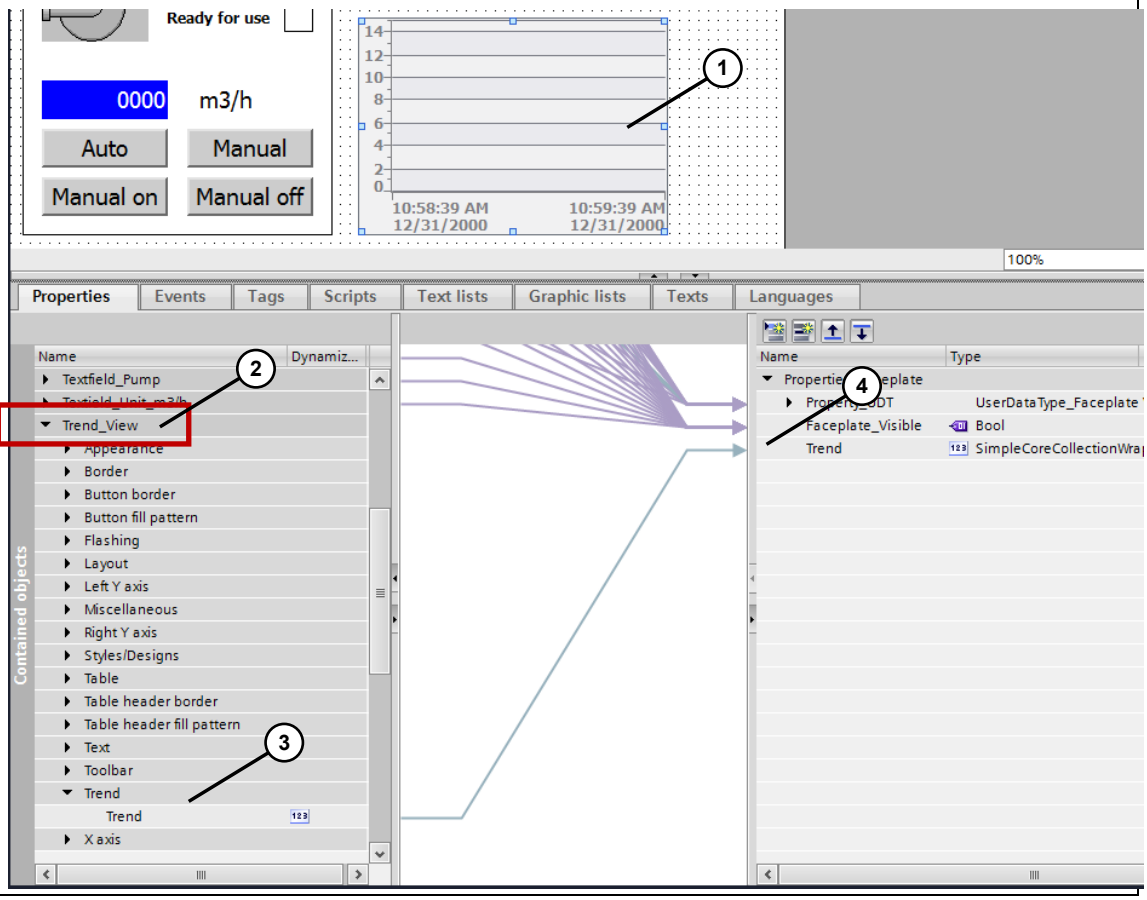
Figure 3-6

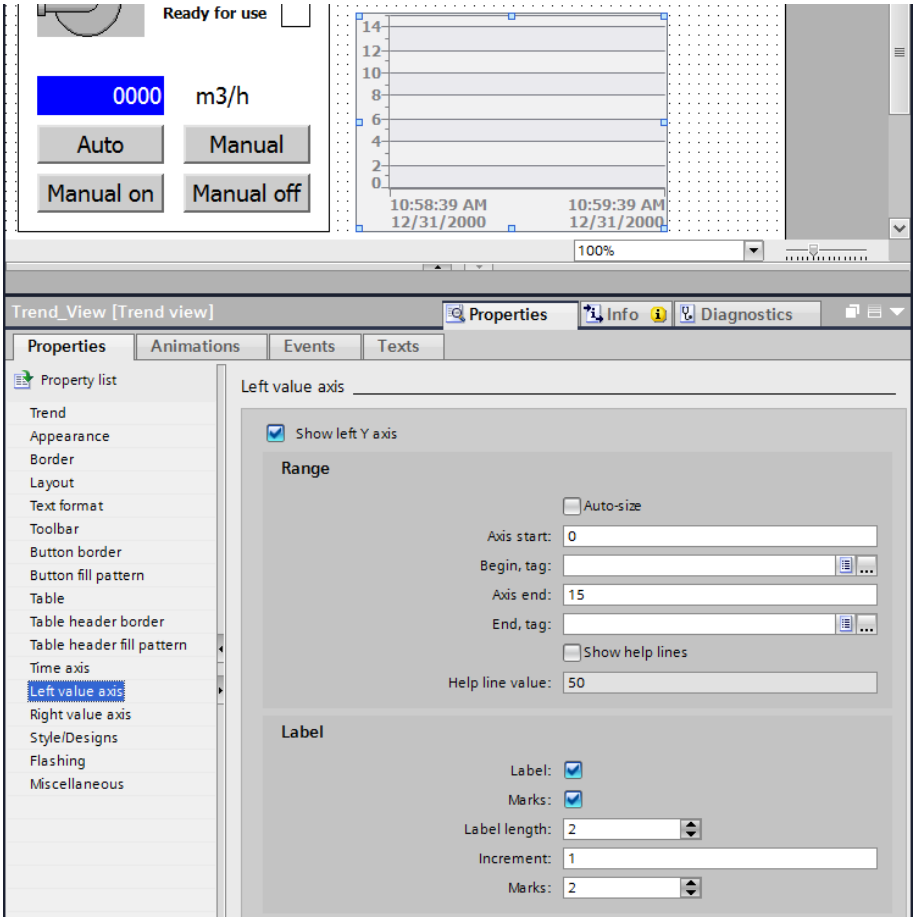


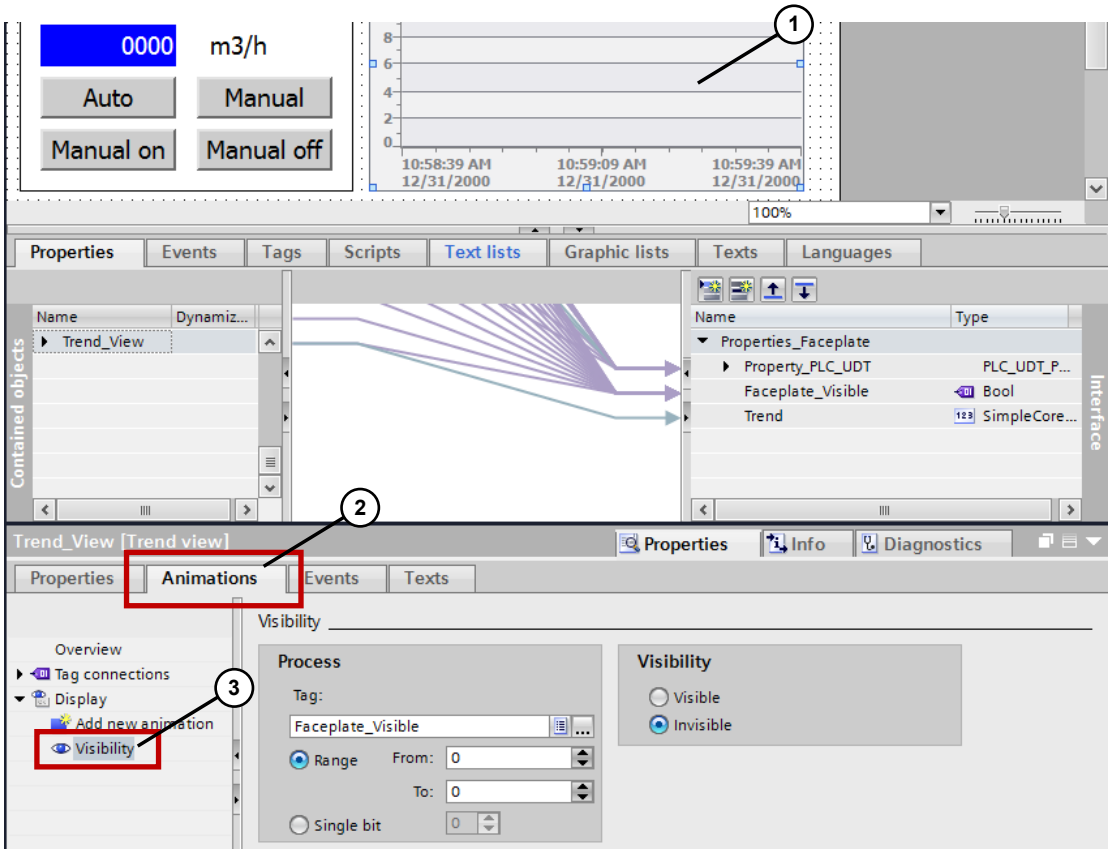
The created control station in “Sample_07” serves as a template.

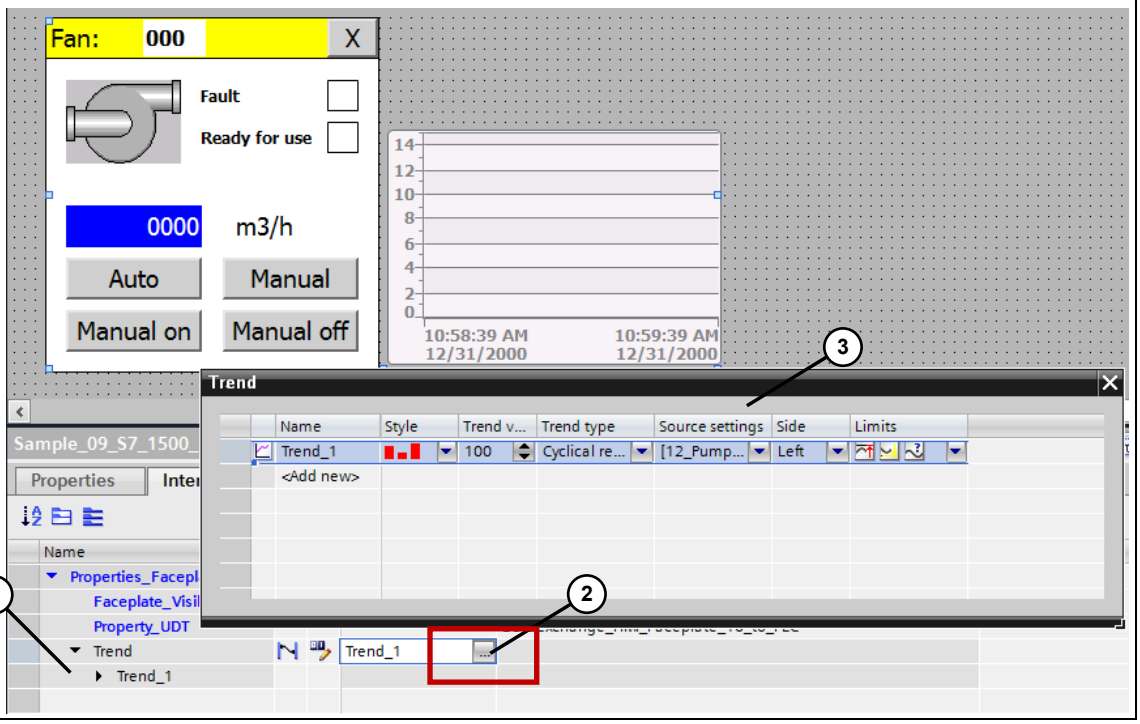
Table 3-12

No.	Action
1.	**Duplicate control stations** • Open the project library and right-click to select the faceplate you wish to duplicate. In this case “Sample_07”. • Select the “Duplicate type” item from the context menu. A window opens where you can specify the properties of the new faceplate, such as the new name, for example. Confirm the entries with “OK”. 

No.	Action
2.	**Insert a trend view** • Open the duplicated faceplate with “Type edit”. • Insert the trend view next to the existing control station (1). • If necessary, adjust the name of the trend view in the configuration area in the “**Properties > Contained objects**” tab. In this example, “Trend_View” (2). • In the “Contained objects” list you open the properties of trend display “Trend_View” (2). • In the list you open the “Trend” entry and there, you select the “Trend” property (3). • Configure a connection between the selected “Property” and the “Interface” list via drag&drop. The data type is adopted automatically (4). **Note** The trends are later assigned at the faceplate via the configured “Trend” interface – irrespective of how many trends you assign to the trend view later on. 

No.	Action
3.	**Display of the trend view** Select die trend view and in the “Properties” of the inspector window, you create the view of the trend view as before. With regards to the example, the following points were adapted: - Table - Time axis - Value axis left/right 

No.	Action
4.	**Animation “Visibility”** Trend view shall be shown and hidden together with the other objects. To do this, the “Visibility” animation is used here. • Select the trend view in the workspace (1). • In the inspector window, you open the “Properties > Animations” tab (2). • Add the “Visibility” animation in the area navigation at “Display” (3). • Selecting a tag: The properties/tags from the “Interface” list are displayed to you via the symbol. Select the “Faceplate_Visible” tag from the list. • Specify the area in which the trend view shall be visible. **In this example:** Area: 0 to 0. Visibility: invisible.  The required settings are thus completed.

No.	Action
5.	**Releasing the faceplate** For the performed configurations to become effective, the created faceplate must be released. The faceplate is updated.
6.	**Faceplate view** **Configuring trends** Call up the created faceplate in a picture. The subsequent figure shows the new created faceplate. In the “Properties > Interface” tab of the faceplate, the created property “**Trend**” is displayed (1). • To configure the trend view, click on the selected button (2) in the “Static value” column. • The “Trend” window opens. In the window you perform the configuration of the trends as before. 

3.2 Application Example

3.2.1 Example 1

In picture “001_Application > Topic_001.1” of the attached HMI configuration, an example is listed of how the created faceplate can be integrated into a plant picture.

Task

In a ventilation system two fan drives shall be operated and monitored via the HMI operator panel. The control elements for controlling the fan drives shall only be visible or called on demand.

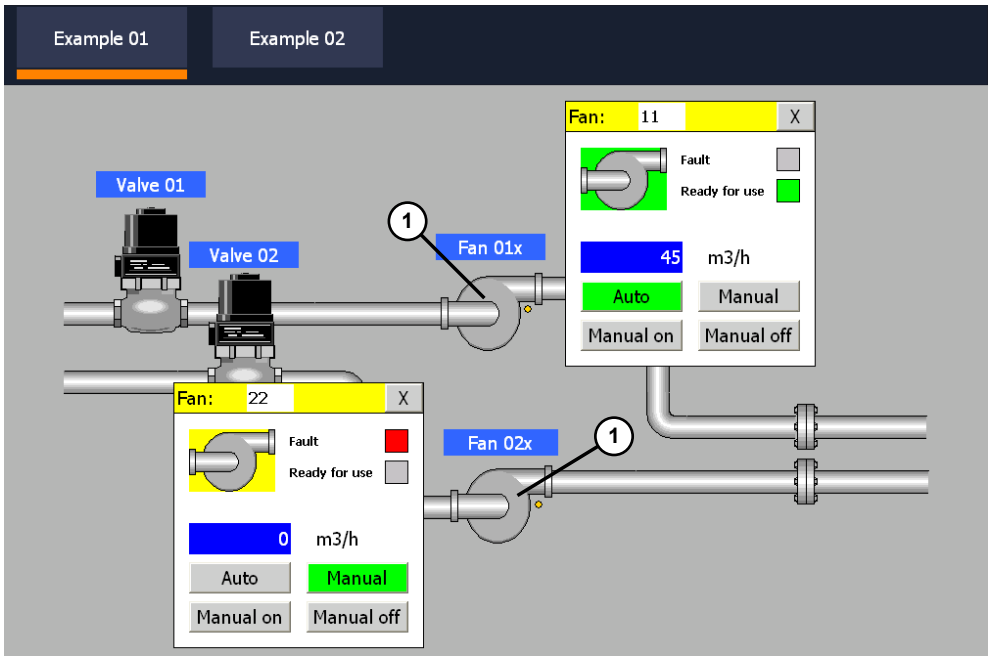
Solution

The following picture shows the control elements (faceplates) in a plant picture.

The used “functions” of the faceplates are described in detail for chapter 3.1.

Clicking on one of the “fan symbols” (1) calls the respective control element or closes it again.
The control element can also be closed again via the “**X**” button at the control element.

Figure 3-7



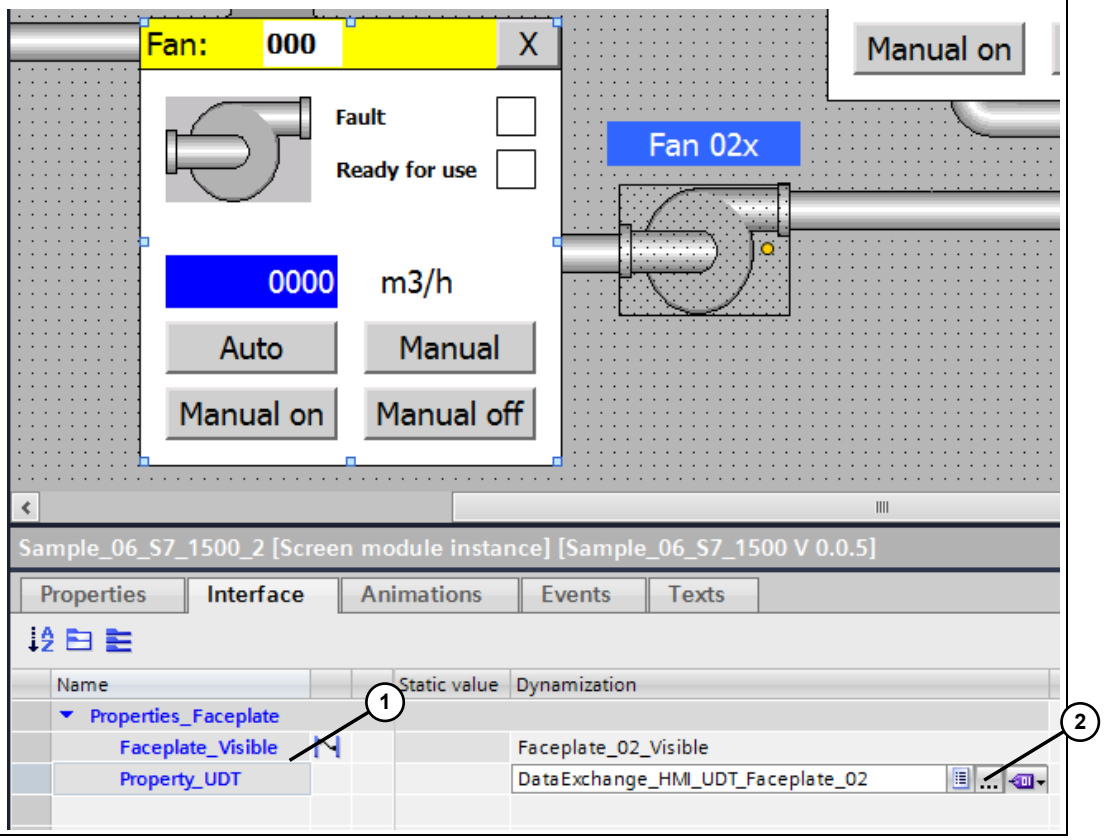
The call of the faceplates for the “Topic_001.1” picture is described on the following page.

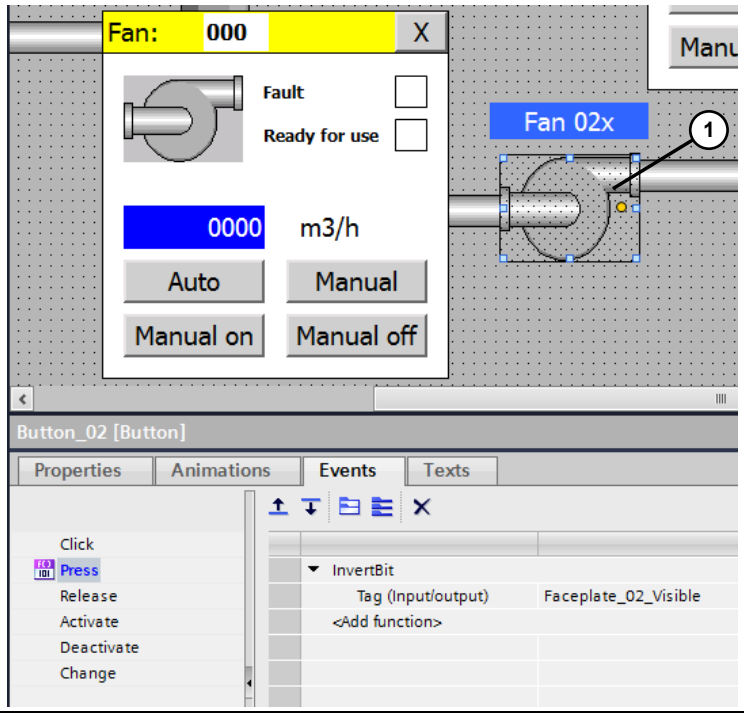
Configuration

In the attached HMI project, the used faceplate is stored as an example (Sample_06).

Faceplate as well as user data type are available in the “Libraries” task card.

Table 3-13

No.	Action
1.	**Configuring the faceplate** Add the “Sample_06” faceplate into the plant picture. In the “**Properties > Interface**” tab of the faceplate both tags “Faceplate_Visible” and “Property_UDT” are displayed (1). Assign the respective HMI tag to this tag (2). **In this example:** Faceplate_Visible -> Faceplate_02_Visible Property_UDT -> DataExchange_FP_02_PLC_HMI_UDT 

No.	Action
2.	**Invisible button** The invisible button shows or hides the faceplate next to it (1). The “InvertiereBit” function is stored in “**Properties > Events > Press**”. The “**Faceplate_02_Visible**” tag is used as the HMI tag. 

3.2.2 Example 2

The “001_Application > Topic_001.2” picture contains the second example.

In the second example, a faceplate is displayed in connection with a trend view. Faceplate “Sample_09” is used as a template.

A PLC program is stored for the example. The operation of the faceplate can be simulated via the PLC program.

The functions can be recreated with PLC Sim and the Runtime simulation.

Figure 3-8

4 Operating the Application

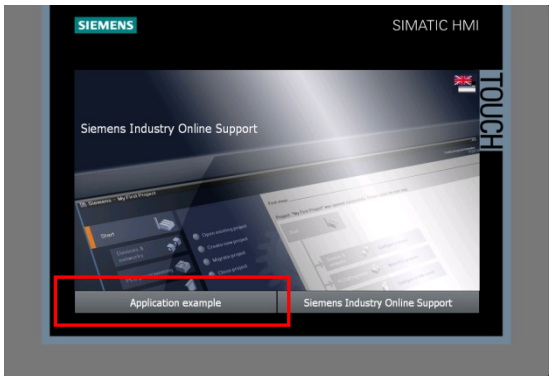
4.1 Overview

The enclosed example configuration helps you get an impression of how the faceplate can be applied.

For testing the example configuration, you can, whenever necessary, also use the simulation integrated in WinCC (TIA Portal).

Calling the example application

Table 4-1

No.	Action	Pictures
1.	**Calling the example application** After starting Runtime, the following screen will be displayed. Here you select "**concrete example**". This takes you to the "**Overview**" screen.	
2.	**"Overview" screen** The "Overview" screen enables you to navigate to the example application. To do this, click the button "**Example 1**" or "**Example 2**".	

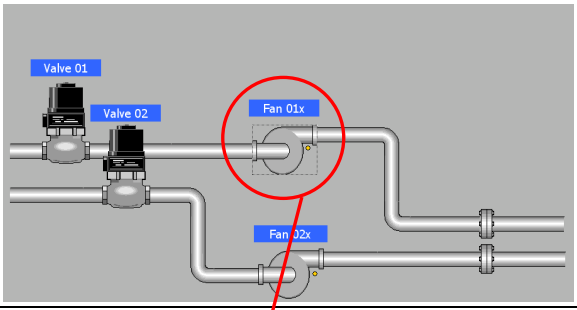
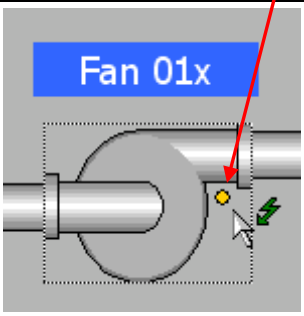
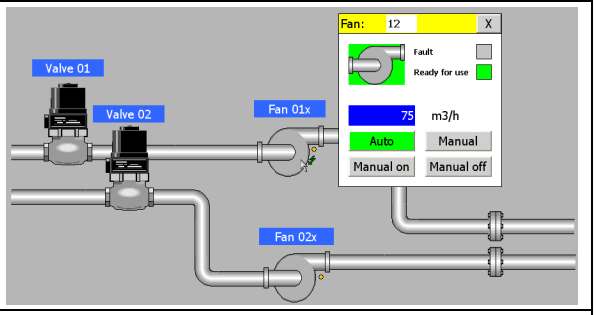
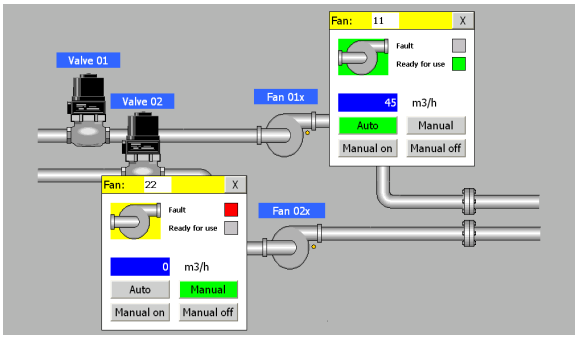
4.2 “Example 1” picture

Showing or hiding the faceplates

The picture shows an application case for using faceplates.

The control stations for the fan drives can alternatively be shown or hidden. The picture remains clearer and provides space for further objects.

Table 4-2

No.	Action	Pictures
1.	**Hiding faceplates** In this case, the two control stations are hidden.	
	Apart from the fan symbol, a yellow point is also displayed so the operator can see behind which object a there is a “function” (button). (Clearly apparent in Runtime).	
2.	**Faceplate shown** In this case, a control station has been called. The call is performed via the invisible button located above the fan symbol. Renewed actuation of the invisible button, or pressing the “**X**” button at the control station, hides the control station again.	
3.	**Both faceplates are shown** In this case, both control stations have been called. The call is performed via the invisible button located above the fan symbol. Renewed actuation of the invisible button, or pressing the “**X**” button at the control station, hides the control station again.	

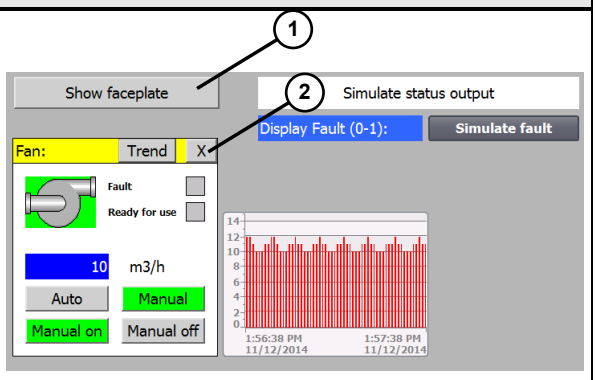
4.3 “Example 2” picture

Functional view of a faceplate

A PLC program is stored for example 2.

The following functions are evaluated or controlled via the PLC program.

Table 4-3

No.	Action	Pictures
1.	**Simulation of the faceplate** **Showing / hiding the faceplate** You can show or hide the button (1) via the faceplate. You can also hide the control station again via the “X” button at the control station. Buttons, fan icon and status display are represented with a background color that depends on the selected function.	

5 Further notes, tips and tricks

Tips and help for trouble-shooting

Why is the configured tag with data type “HMI UDT” displayed “red” at the interface of the faceplate?

- Check the used communication driver in the HMI UDT (e.g. S7 300/400 or S7 1500).
- Check the used connection to the HMI tag you assigned the “HMI UDT” as data type to.
- The communication driver at the “HMI UDT” and the used “connection” must match.

What do you need to observe when changing over from a “HMI UDT” to a PLC data type?

- When recreating the structure of the HMI UDT 1.1 in the PLC data type, the connections are adopted when “exchanging” the HMI UDT. Otherwise you need to recreate the connections.

How do you know the version number of the faceplate in the picture?

- Call the properties of the faceplate. The version number is displayed in the “header” of the properties.

Why are functions executed in the faceplate (for example, an object is “shown/hidden”) even though no parameter exists at the interface?

- Open the faceplate
- In the inspector window you view the properties of the individual objects.

Do all of the parameters of a “HMI UDT” or a PLC data type need to be switched?

- No, not all of the stored parameters need to be switched.
  You can, for example, create a “universal” UDT which covers all possible functions and can thus be used for different for faceplates.

What do you need to observe when editing/expanding a PLC data type retrospectively?

- After the changes have been performed, the project must first be restarted. Then, the faceplate to which the PLC data type is linked must be adjusted.

6 Links & Literature

Internet link specifications

This list is by no means complete and only presents a selection of suitable information.

Table 6-1

	Topic	Title
\1\	Link to this document	https://support.industry.siemens.com/cs/ww/en/view/68014632
\2\	Siemens Industry Online Support	Industry Online Support https://support.industry.siemens.com
\3\	FAQ	What are the functional differences between the different SIMATIC panels? https://support.industry.siemens.com/cs/ww/en/view/40227286
\4\	FAQ	How can you create faceplates in WinCC (TIA Portal) with user authorizations? https://support.industry.siemens.com/cs/ww/en/view/57434982
\5\	Application	Application: Sample blocks for STEP 7 and WinCC flexible https://support.industry.siemens.com/cs/ww/en/view/36435784
\6\	Application	Faceplates for the Visualization of Sentron PAC Power Meters. https://support.industry.siemens.com/cs/ww/en/view/67318600

7 History

Table 7-1

Version	Date	Modifications
V1.0	01/2013	First version
V2.0	06/2015	Update to WinCC V13 SP1 (new function – PLC data types)