SIEMENS

SIMATIC HMI

WinCC Unified Getting Started

System Manual

Online documentation

07/2021

Online documentation

Legal information

Warning notice system

This manual contains notices you have to observe in order to ensure your personal safety, as well as to prevent damage to property. The notices referring to your personal safety are highlighted in the manual by a safety alert symbol, notices referring only to property damage have no safety alert symbol. These notices shown below are graded according to the degree of danger.

⚠ DANGER

indicates that death or severe personal injury **will** result if proper precautions are not taken.

⚠ WARNING

indicates that death or severe personal injury **may** result if proper precautions are not taken.

⚠ CAUTION

indicates that minor personal injury can result if proper precautions are not taken.

NOTICE

indicates that property damage can result if proper precautions are not taken.

If more than one degree of danger is present, the warning notice representing the highest degree of danger will be used. A notice warning of injury to persons with a safety alert symbol may also include a warning relating to property damage.

Qualified Personnel

The product/system described in this documentation may be operated only by **personnel qualified** for the specific task in accordance with the relevant documentation, in particular its warning notices and safety instructions. Qualified personnel are those who, based on their training and experience, are capable of identifying risks and avoiding potential hazards when working with these products/systems.

Proper use of Siemens products

Note the following:

⚠ WARNING

Siemens products may only be used for the applications described in the catalog and in the relevant technical documentation. If products and components from other manufacturers are used, these must be recommended or approved by Siemens. Proper transport, storage, installation, assembly, commissioning, operation and maintenance are required to ensure that the products operate safely and without any problems. The permissible ambient conditions must be complied with. The information in the relevant documentation must be observed.

Trademarks

All names identified by ® are registered trademarks of Siemens AG. The remaining trademarks in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owner.

Disclaimer of Liability

We have reviewed the contents of this publication to ensure consistency with the hardware and software described. Since variance cannot be precluded entirely, we cannot guarantee full consistency. However, the information in this publication is reviewed regularly and any necessary corrections are included in subsequent editions.

Siemens AG
Digital Industries
Postfach 48 48
90026 NÜRNBERG
GERMANY

Document order number: Online documentation
Ⓟ 08/2021 Subject to change

Table of contents

Welcome

1

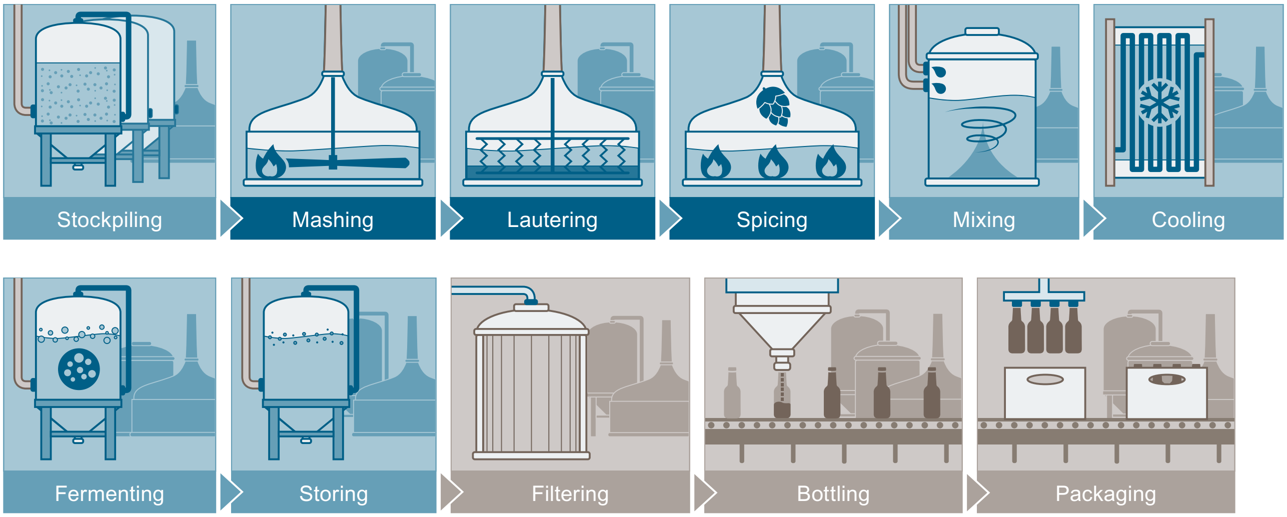
1.1 Welcome

Welcome to the Getting Started for "WinCC Runtime Unified". Using the example of a medium-sized brewery, we will show you how you can use WinCC to take the first step in configuring the operator control and monitoring solution for managing the brewery process.

Subject of the configuration

The configuration covers the following process steps from the brewery process shown below:

- Mashing
- Lautering
- Spicing



The brewery only processes raw materials from the region, which is why only one brewing kettle is used.

Configuration steps

In the Getting Started, you will get to know the following configuration steps:

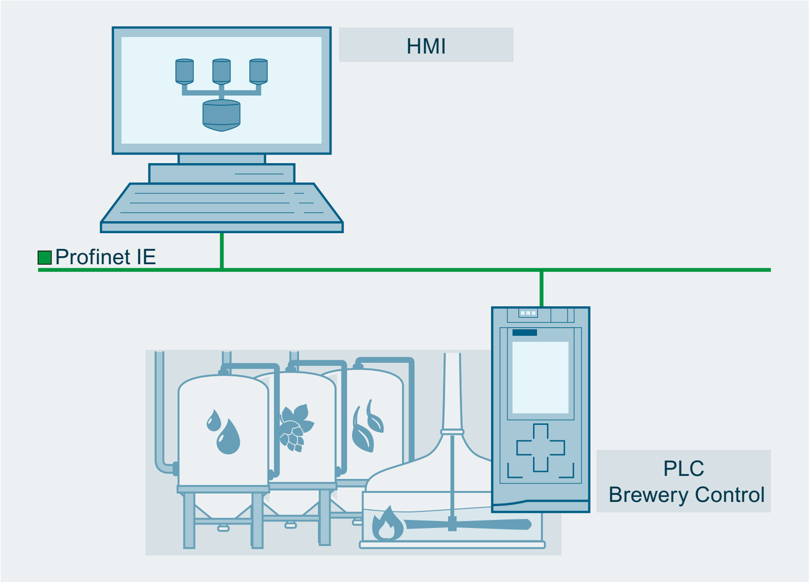
- Configuring the visualization of the brewing process
- Monitoring and logging the brewing process
- Configuring parameter sets (optional)
- Configuring user administration

- Configuring screen navigation
- Carrying out a function test in runtime

Sample project

If you simply want to view the configuration steps and results shown in this Getting Started, use the "Brewery" project. This project contains the complete, ready-to-run project in which all the configuration steps described in the Getting Started have already been completed.

The project consists of a controller and an HMI. The HMI is connected to the controller via Ethernet. The brewing process is operated and monitored via the HMI. The controller processes the parameters according to its programming and sends measurement data from the sensors to the HMI for the display.



See also

Sample project "Brewery" as download in the SIOS portal (https://support.industry.siemens.com/cs/ww/en/view/109757953)

Configuring the visualization of the brewery 2

2.1 Creating a project

2.1.1 Projects in WinCC V16

Basics

All the data for an HMI solution is stored with structure in a project: One or more HMI devices including process communication and configuration data for each operator panel.

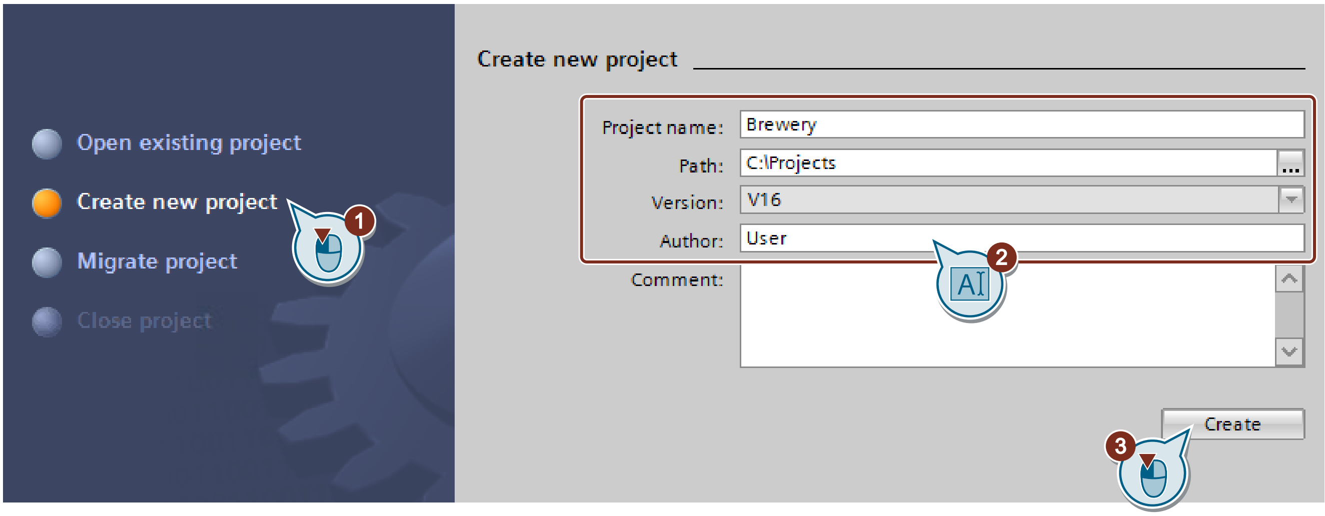
2.1.2 Creating a project

Introduction

A project is to be created for the brewery sample project.

Procedure

1. Create a new project



Create new project

Project name: Brewery

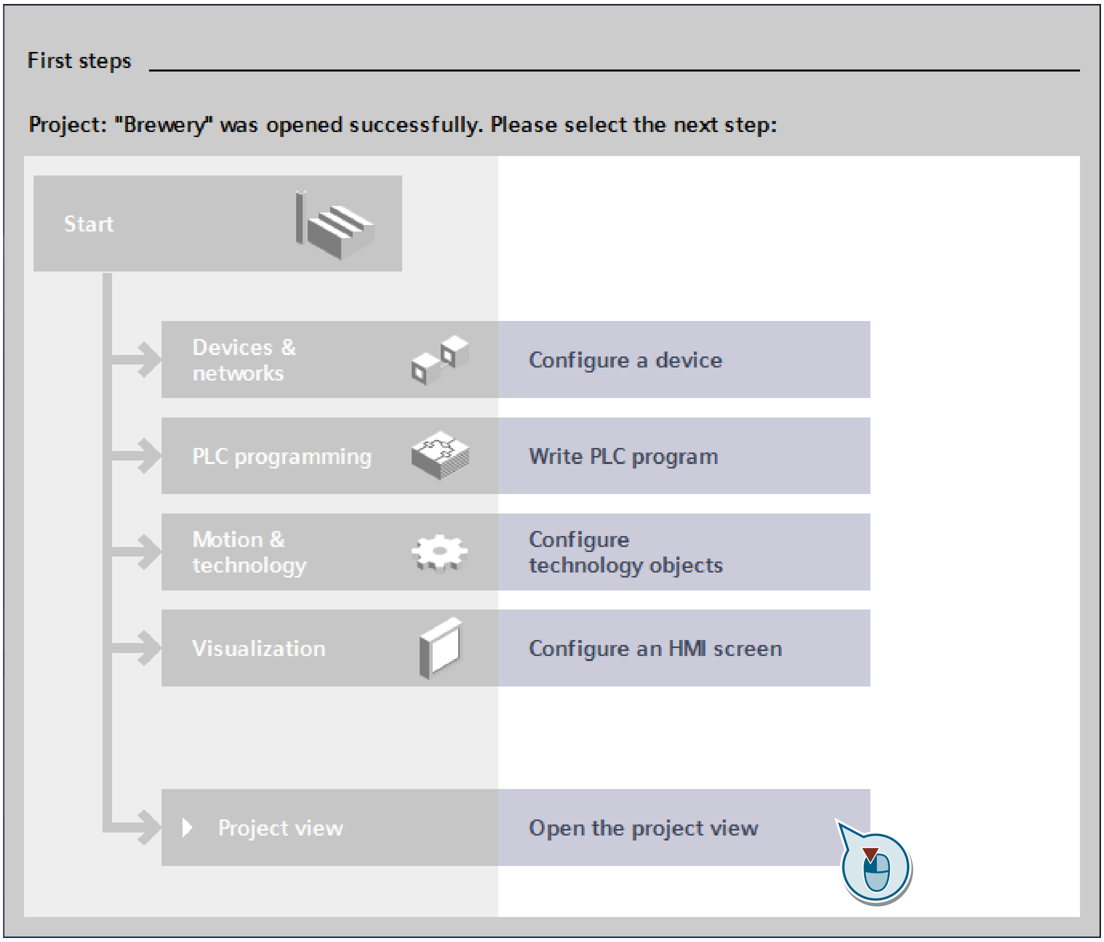
Path: C:\Projects

Version: V16

Author: User

Comment:

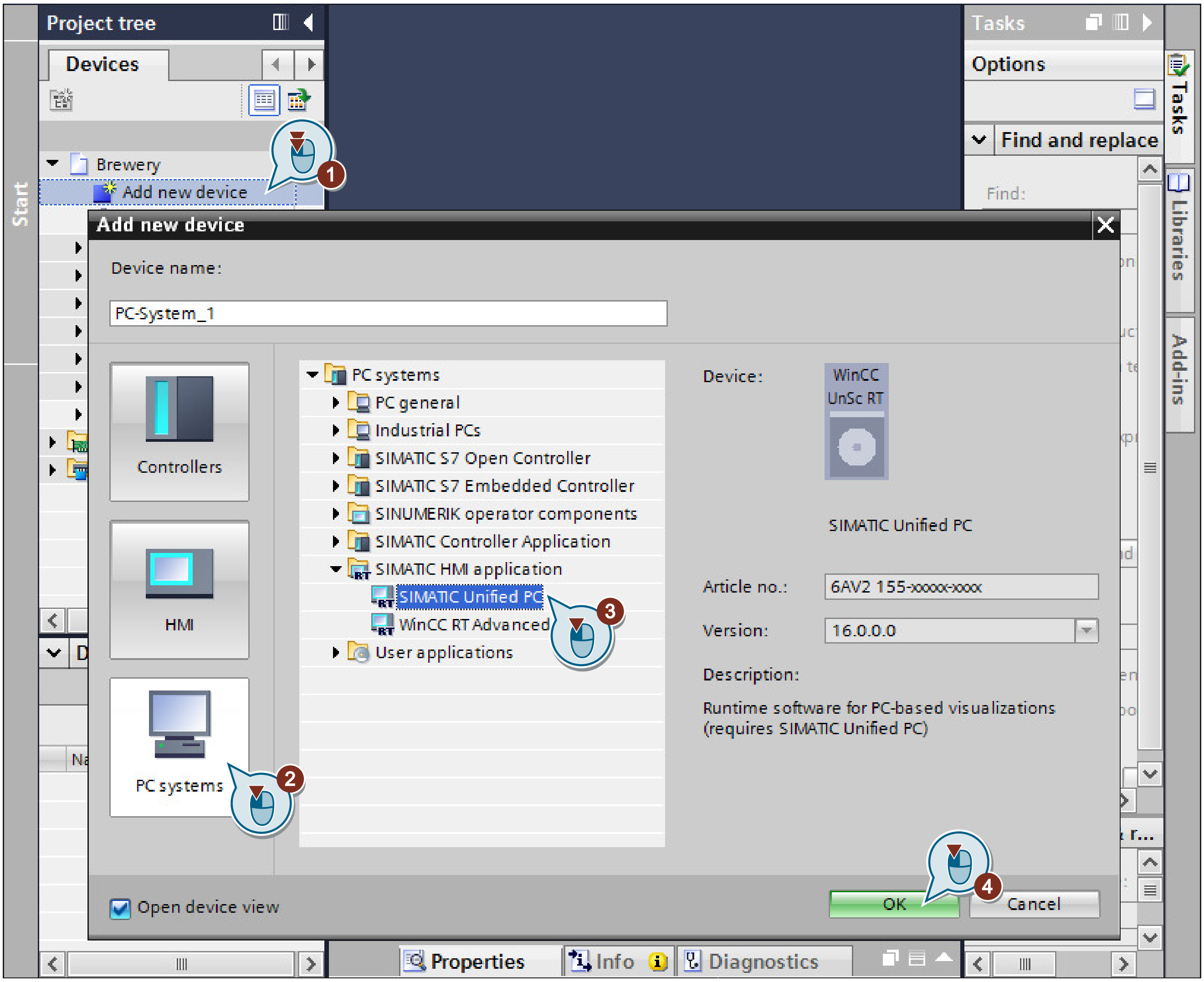
2. Switch to the project view



First steps

Project: "Brewery" was opened successfully. Please select the next step:

3. Add a new device



Add new device

Device name: PC-System_1

Device: SIMATIC Unified PC

Article no.: 6AV2 155-xxxxx-xxxx

Version: 16.0.0.0

Description:
Runtime software for PC-based visualizations (requires SIMATIC Unified PC)

☑ Open device view

2.2 Creating tags

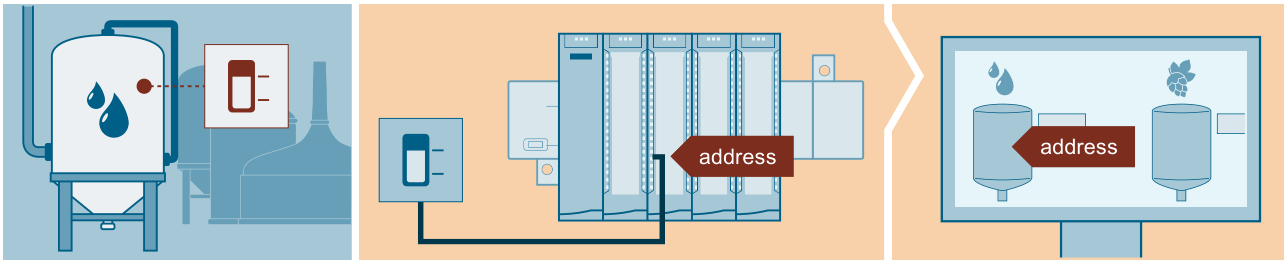
2.2.1 Tags

Basics

WinCC works with two types of tags:

- External tags
- Internal tags

The external tags form the link between WinCC and the automation systems. The values of external tags correspond to the process values from the memory of an automation system.



Internal tags do not have a process link and only convey values within WinCC.

2.2.2 Creating tags

Introduction

During the brewing process, individual process variables such as the fill levels in the storage tanks, or the temperature in the mash tun, are to be monitored. A tag must be created for each of these variables. In the following, the tags shown in the table are to be created and organized in tag tables.

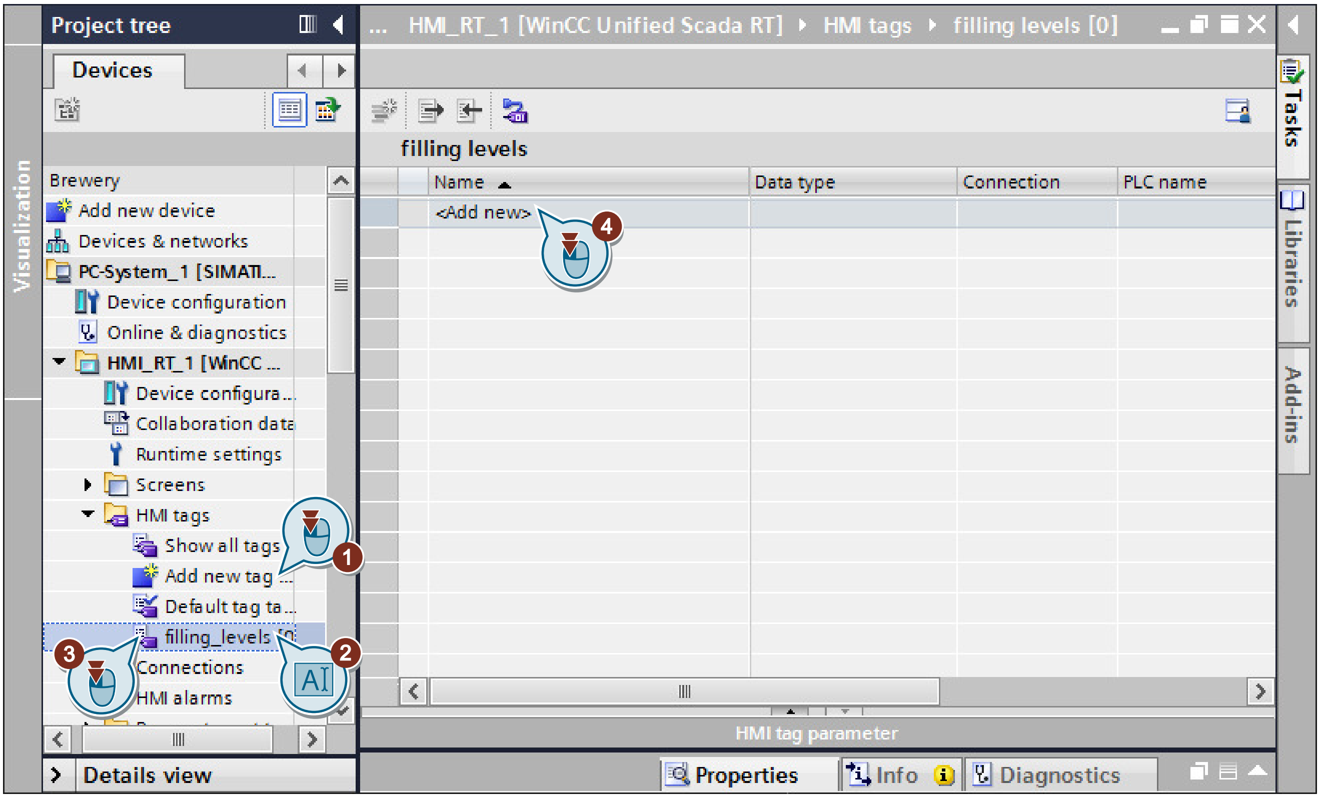
Name of the tag table	Use
filling_levels	Fill level of tanks
temperature	Temperature of tanks
pressure	Pressure of tanks
valve_status	Valve status
mash_tun	Mash tun states

Tag name	Use	Data type	Tag table
HMI_filling_level_water	Fill level of water tank	Int	filling_levels
HMI_filling_level_hop	Fill level of hops tank	Int	filling_levels
HMI_filling_level_malt	Fill level of malt tank	Int	filling_levels
HMI_valve_status_water	Valve status of water tank open/ closed	Bool	valve_status
HMI_valve_status_hop	Valve status of hops tank open/ closed	Bool	valve_status
HMI_valve_status_malt	Valve status of malt tank open/ closed	Bool	valve_status
HMI_filling_level_mash_tun	Fill level of mash tun	Int	mash_tun
HMI_temperature_mash_tun	Temperature of mash tun	Int	mash_tun
HMI_mash_tun_heating	Heater active/inactive	Bool	mash_tun
HMI_pressure_water	Pressure of water tank	Int	pressure
HMI_pressure_hop	Pressure of hops tank	Int	pressure
HMI_pressure_malt	Pressure of malt tank	Int	pressure
HMI_temperature_water	Temperature of water tank	Int	temperature

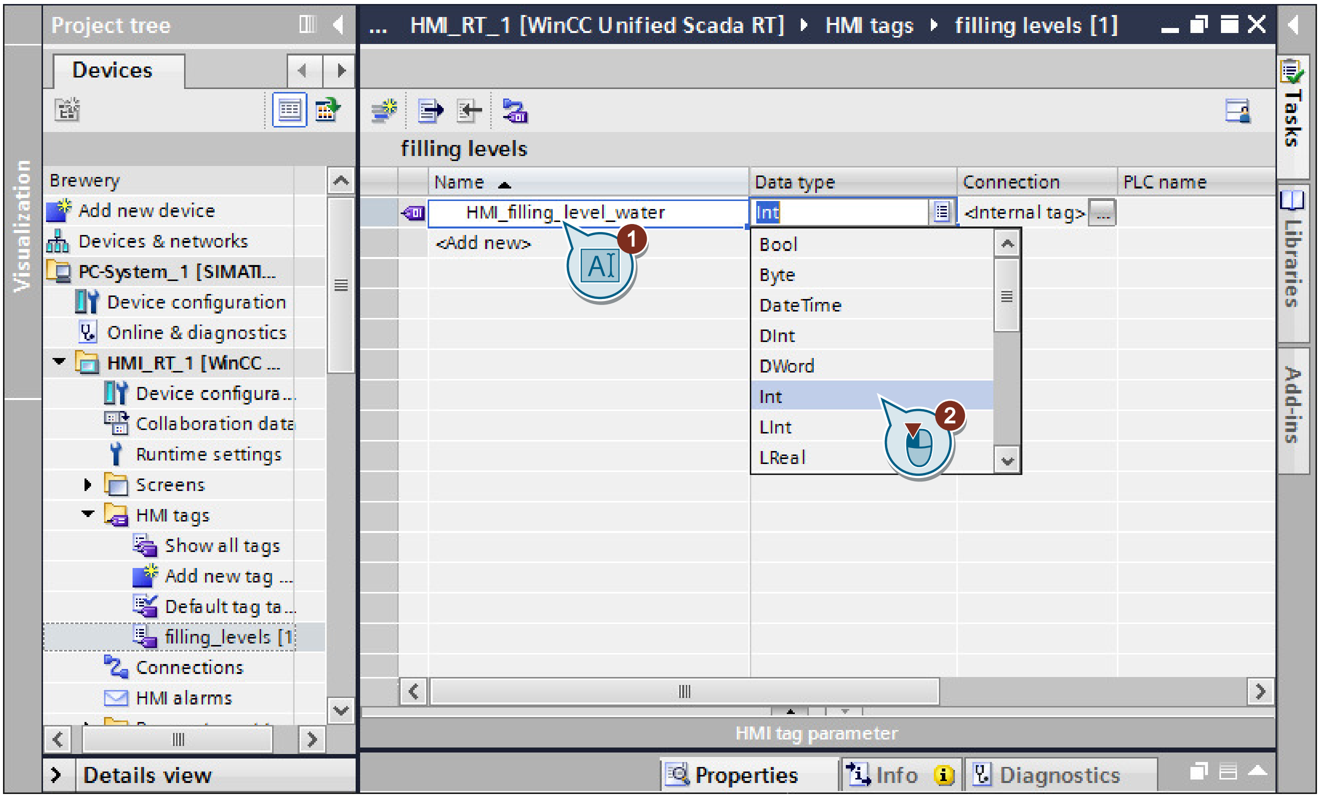
HMI_temperature_hop	Temperature of hops tank	Int	temperature
HMI_temperature_malt	Temperature of malt tank	Int	temperature

Procedure

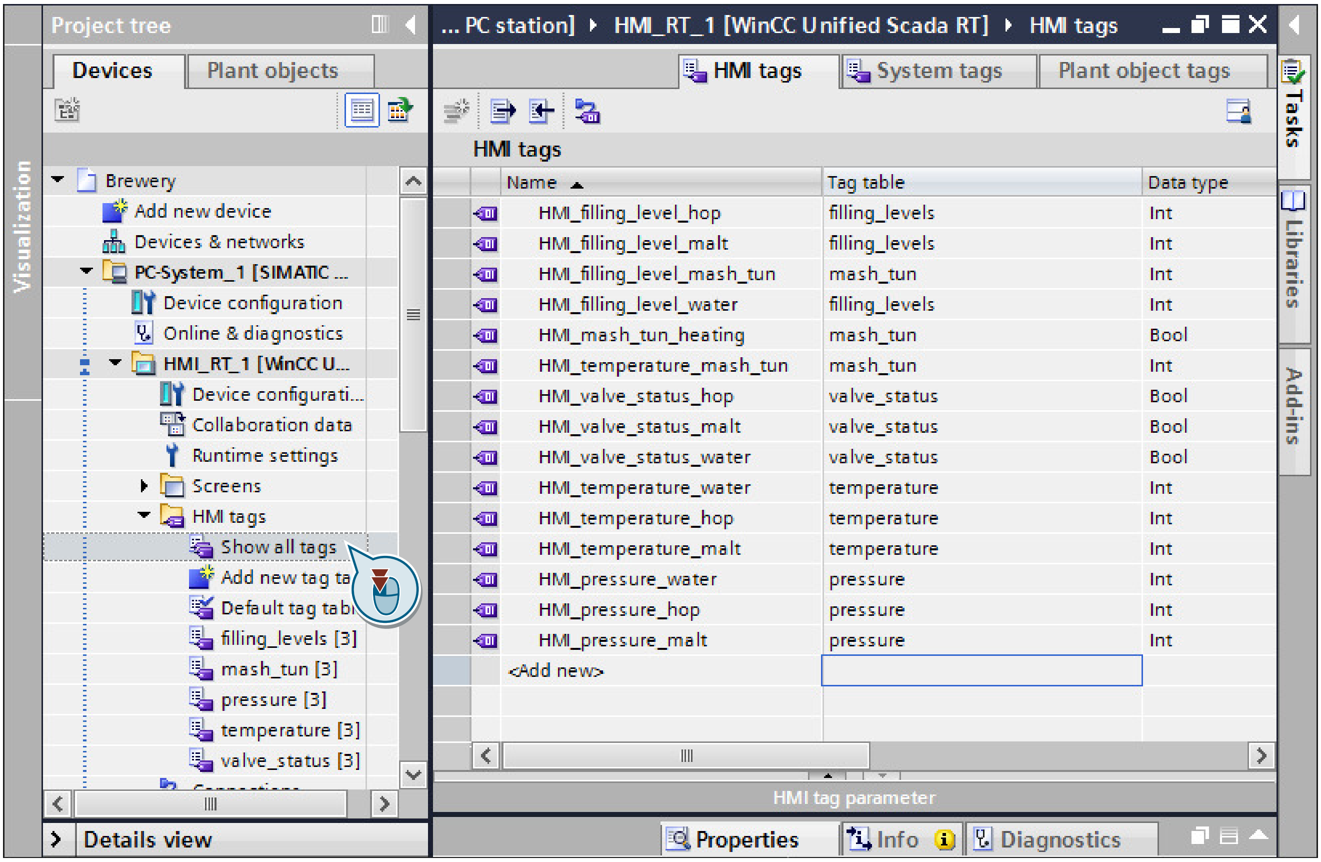
1. Create a tag table and add a tag to it.



2. Name the tag and choose the data type.



3. Create the rest of the tags in the same way.
   Check your results by viewing all tags.



HMI tags

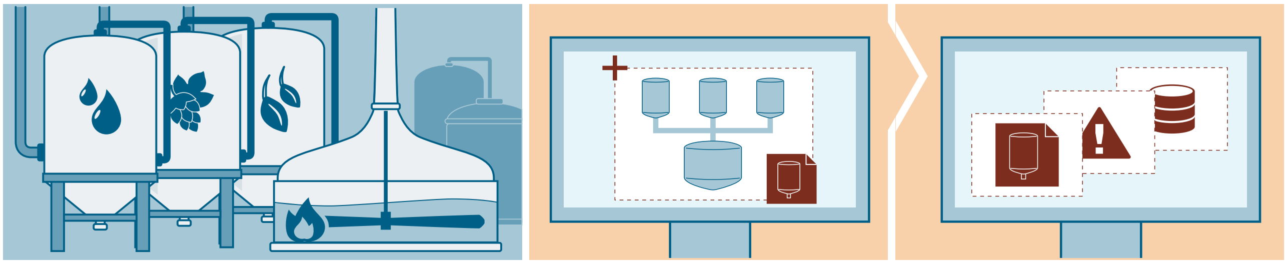
Name ▲	Tag table	Data type
HMI_filling_level_hop	filling_levels	Int
HMI_filling_level_malt	filling_levels	Int
HMI_filling_level_mash_tun	mash_tun	Int
HMI_filling_level_water	filling_levels	Int
HMI_mash_tun_heating	mash_tun	Bool
HMI_temperature_mash_tun	mash_tun	Int
HMI_valve_status_hop	valve_status	Bool
HMI_valve_status_malt	valve_status	Bool
HMI_valve_status_water	valve_status	Bool
HMI_temperature_water	temperature	Int
HMI_temperature_hop	temperature	Int
HMI_temperature_malt	temperature	Int
HMI_pressure_water	pressure	Int
HMI_pressure_hop	pressure	Int
HMI_pressure_malt	pressure	Int
<Add new>		

2.3 Configure the HMI screen

2.3.1 HMI screens

Basics

An HMI screen represents the user interface of an operator control and monitoring station. A system or a process is visualized with predefined screen objects in the HMI screen. The operator interacts with the process via the HMI screen.



A screen consists of static and dynamic screen objects:

- Static screen objects do not depend on the process. Static screen objects include, for example, labels and diagrams.
- Dynamic screen objects change in line with the process. Dynamic screen objects typically visualize process values such as fill levels. The operator uses dynamic screen objects to intervene in the process.

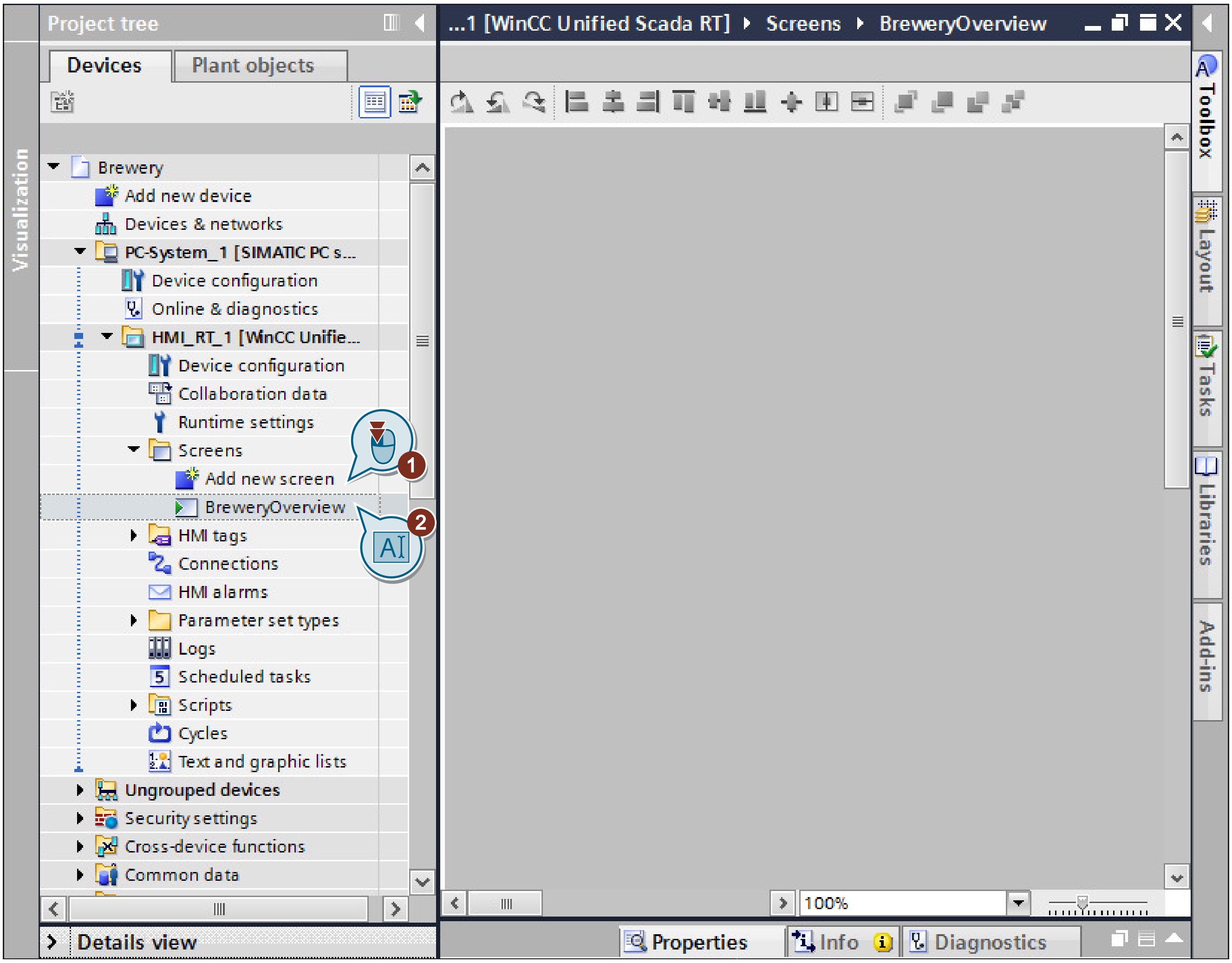
2.3.2 Configuring HMI screens for operating and monitoring

Introduction

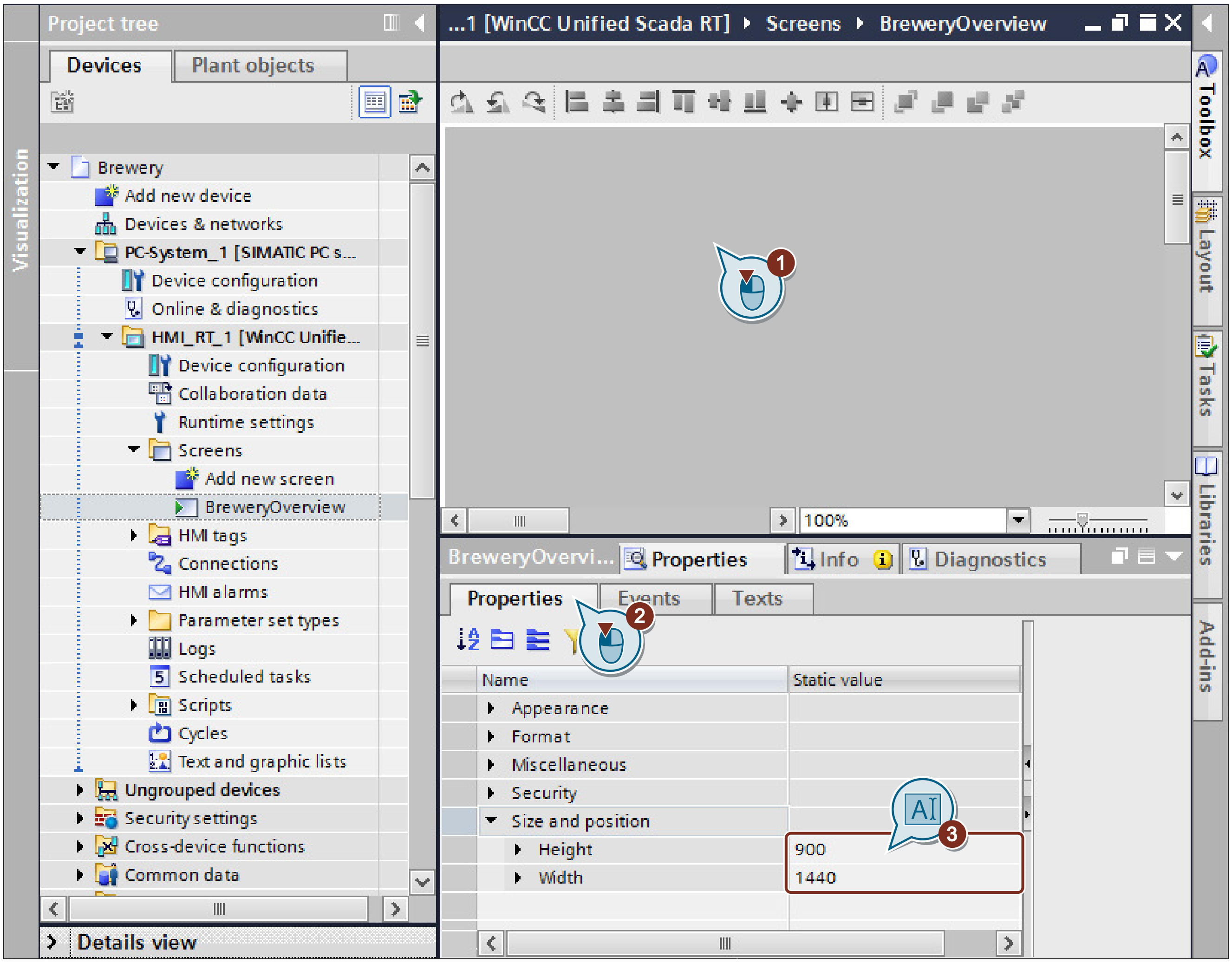
HMI screen name	Screen type	Use	Size in pixels
BreweryOverview	Start screen	Overview screen for operating and monitoring the brewing process	1440x900
BreweryProcess	Process control	Visualization of the brewing process	1440x800
Alarms	Process control	Overview of alarms from the process	
ProcessValues	Process control	Overview of logged process values	
ProductCtrl	Process control	Production control	
Header	Navigation	Screen for opening the navigation between process control screens	1440x100
Navigation	Navigation	Screen for navigating between process control screens	250x300

Procedure

1. Create the HMI screen to visualize the start screen.

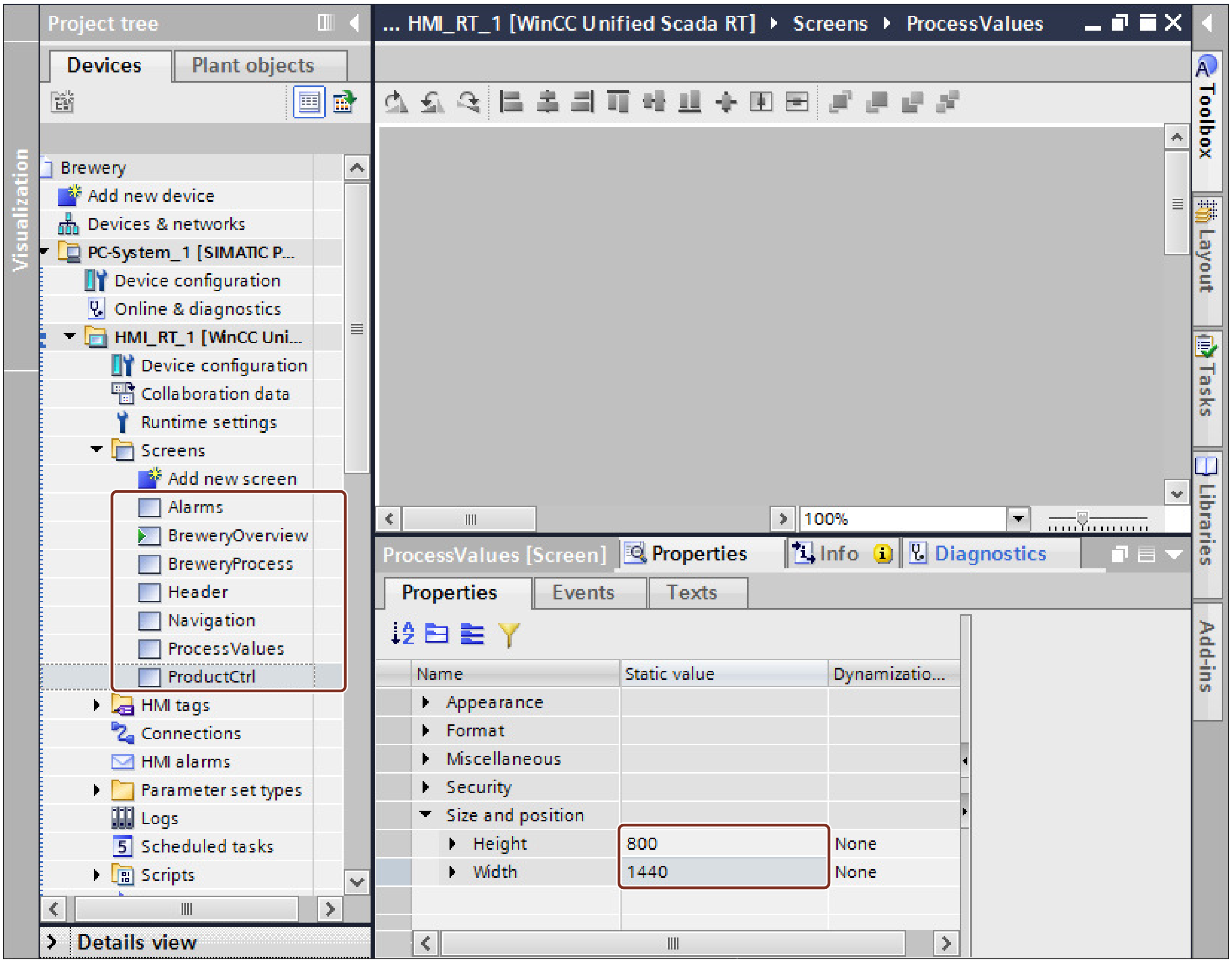


2. Adjust the size of the HMI screen.



Name	Static value
Height	900
Width	1440

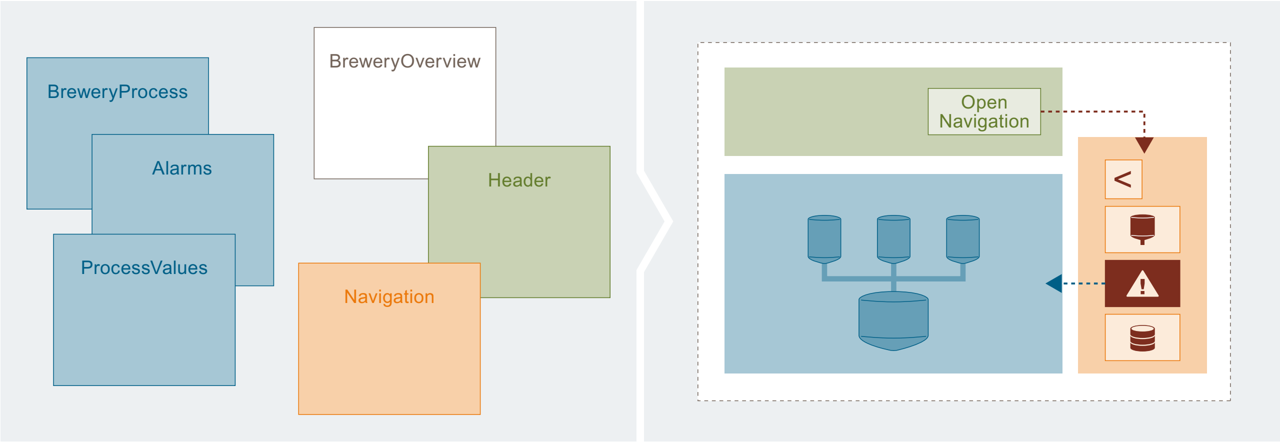
3. Create the other HMI screens in the same way and specify the screen size shown in the table above.



Name	Static value	Dynamizatio...
▸ Appearance		
▸ Format		
▸ Miscellaneous		
▸ Security		
▾ Size and position		
▸ Height	800	None
▸ Width	1440	None

2.3.3 Configuring screen navigation

Purpose



You implement the screen navigation using the screen window technology. In an overview screen, you insert the following screen windows, which fulfill different tasks:

- Header
  A navigation window is shown above the header.
- Navigation
  The operator uses the buttons in the navigation window to change the content in the content window.
  The navigation window is hidden by default.
- Content
  Information on the system is displayed in the content window. For example, the level of the tanks or alarms in the system are displayed.

You specify the content of the screen windows using HMI screens. In this section, you create all the HMI screens that you need to implement the screen navigation in this Getting Started.

Template Suite

Systems and machines are becoming more and more complex and the demands placed on the operator of a system are increasing. An intuitive and graphically appealing user interface as an interface between man and machine is therefore becoming increasingly important. The HMI Template Suite provides you with templates and ideas for configuring your HMI device in a clear and modern way.

The layout and design are designed for smooth operability, clarity and expandability. In this way, you can simplify the operation of your machine and reduce operating errors.

In the following application example, you will learn how to work with the HMI Template Suite:

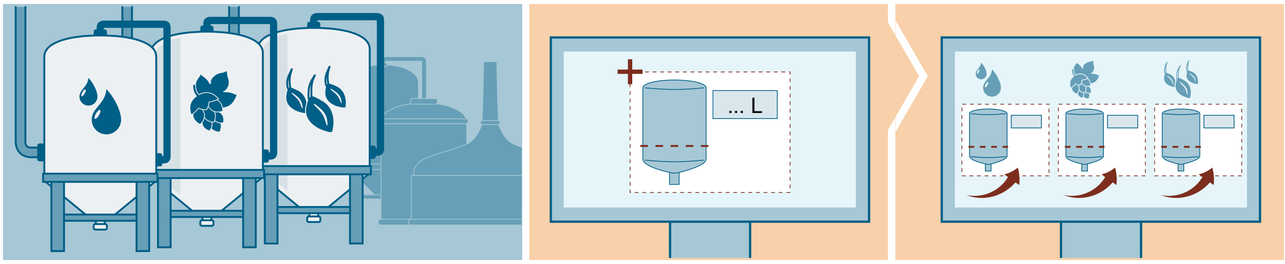
Application example "Template Suite" (https://support.industry.siemens.com/cs/us/en/view/91174767)

2.4 Configuring storage tanks with faceplates

2.4.1 Faceplates

Basics

Faceplates are user-defined groups of display and operating objects that are stored, managed and edited centrally in the project. All changes to the faceplate in the project can be changed centrally in the faceplate type. Ideally, you should use faceplates for plant objects or plant units that you use several times and that have identical data structures.



In order to support central changeability, faceplates are based on a type-instance model:

- You create properties for faceplate instances centrally in the faceplate type.
- The instances represent local points of use of the faceplate type.

2.4.2 Configuring a faceplate type

Introduction

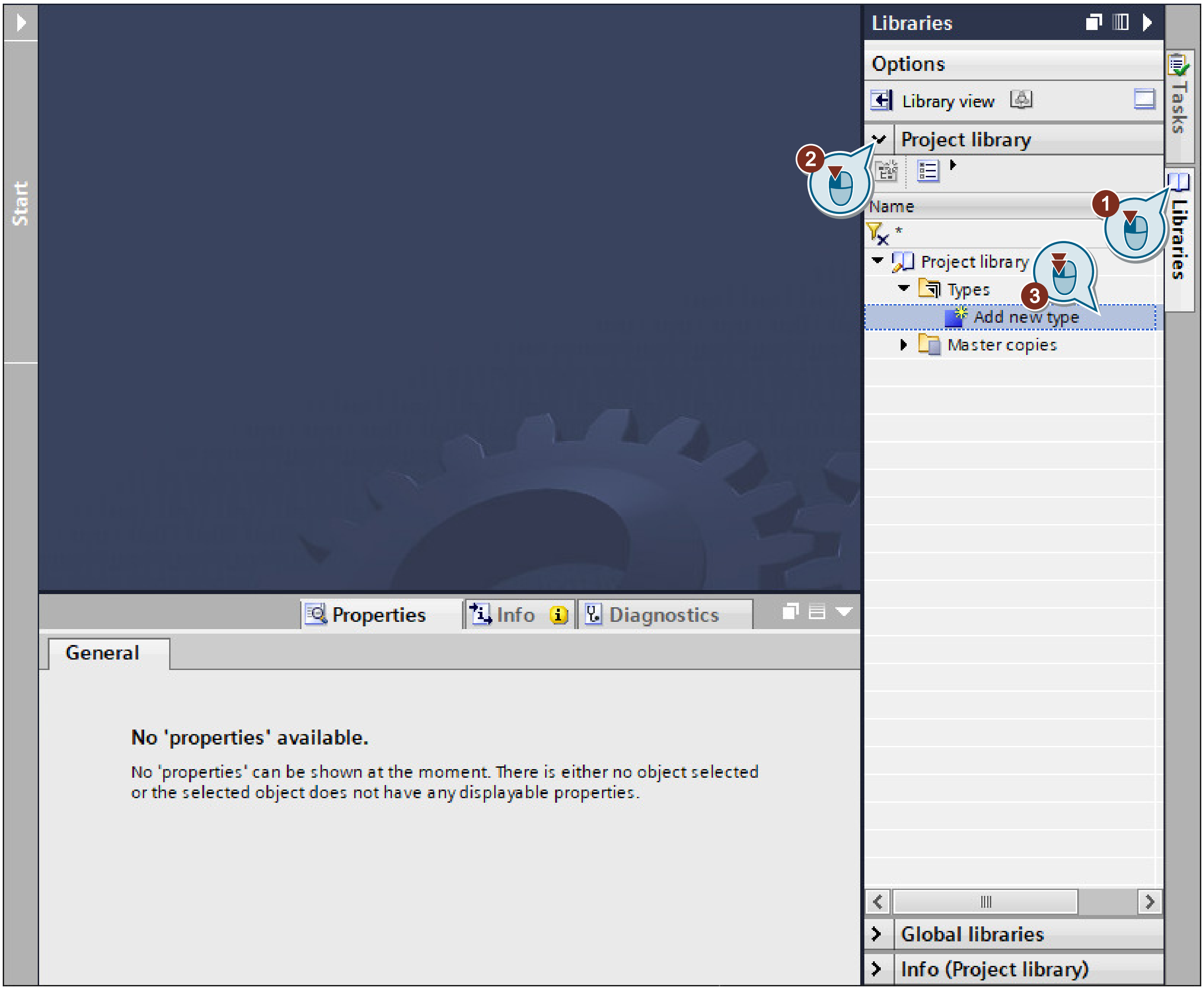
The basic ingredients water, hops and malt are stored in three identical storage tanks. A valve on each storage tank controls the transport into the mash tun. Because the same display and control elements are used for all tanks, use the faceplate technology to implement the visualization. You define properties on a faceplate type that are parameterized for later use.

You need the following properties to configure the faceplate type:

Property	Use	Data type
filling_level	Measured fill level in the tank	Int
valve_status	Valve status	Bool
temperature	Measured temperature in the tank	Int
pressure	Measured pressure in the tank	Int
name	Name of the ingredient	WString

Procedure

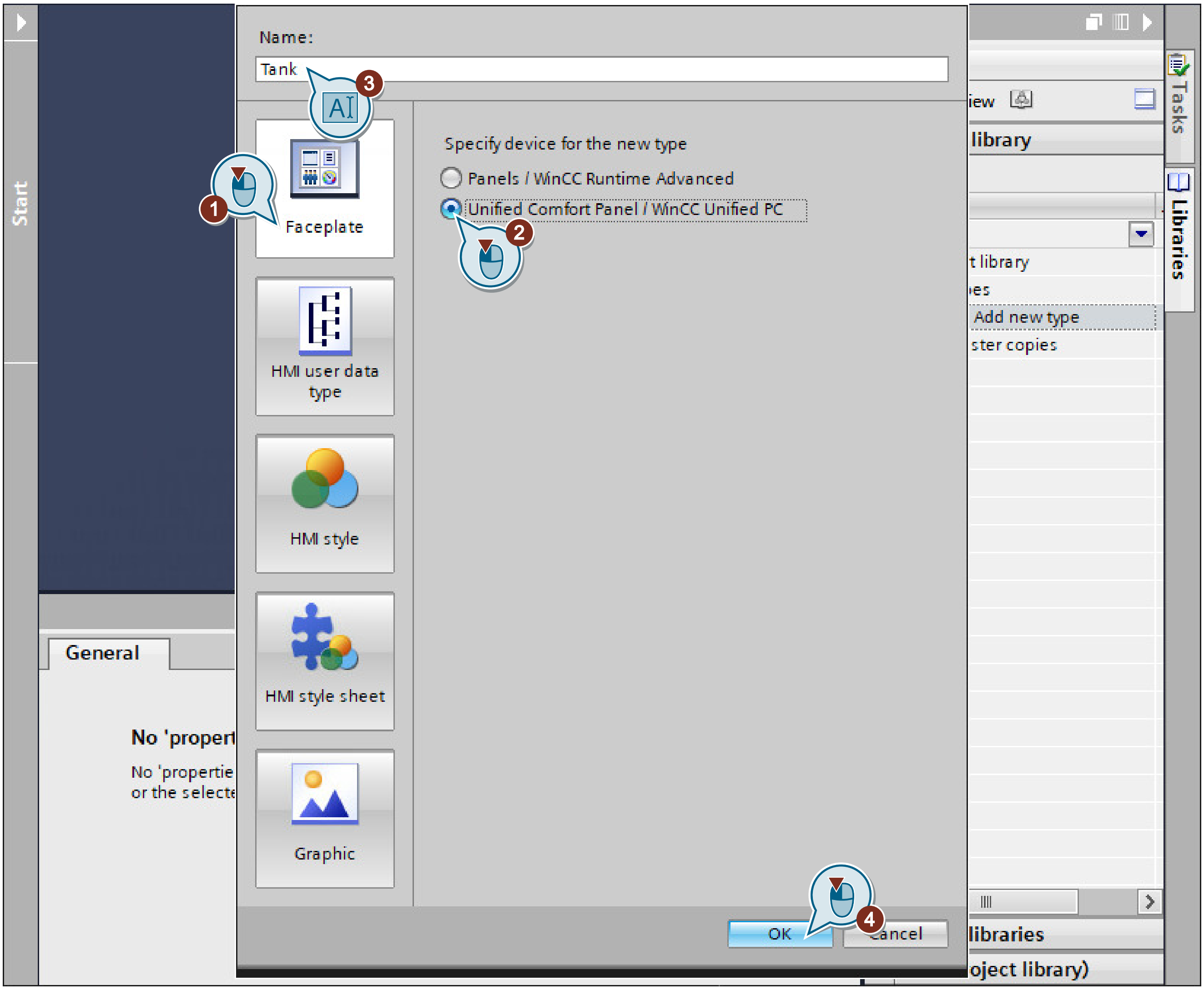
1. Create a new faceplate type in the library.



No 'properties' available.

No 'properties' can be shown at the moment. There is either no object selected or the selected object does not have any displayable properties.

2. Configure the new faceplate type.



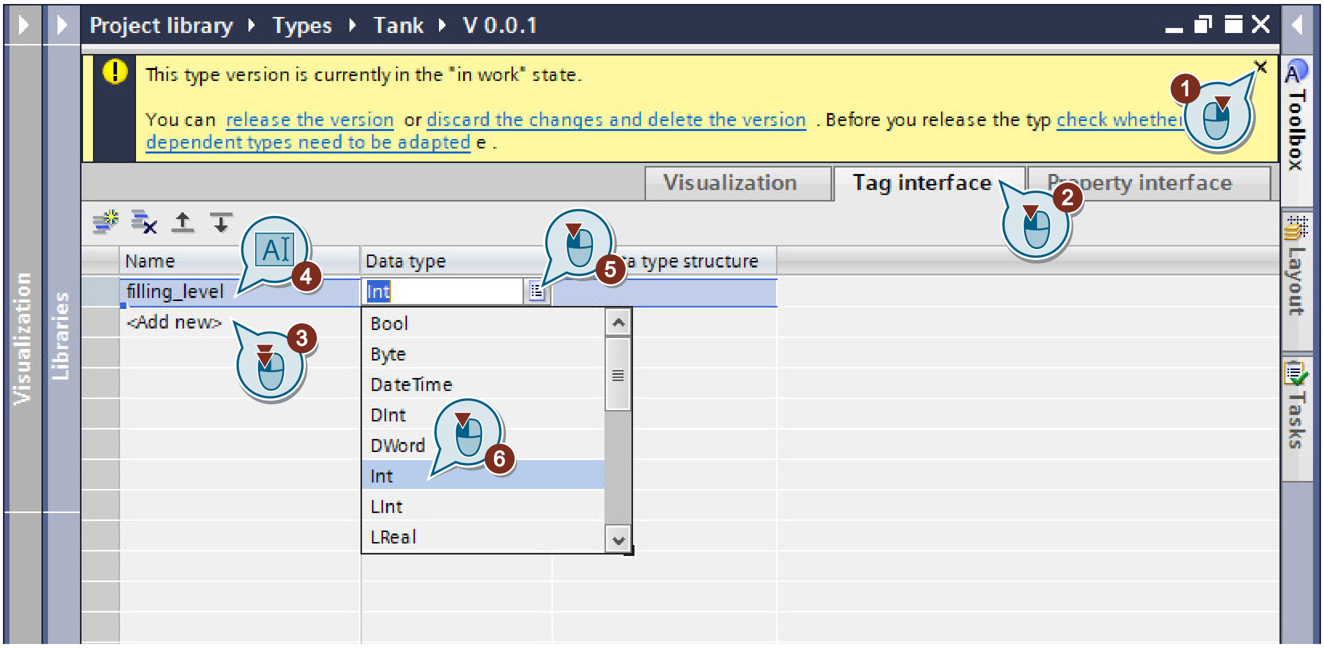
Name:
Tank

Specify device for the new type

○ Panels / WinCC Runtime Advanced

● Unified Comfort Panel / WinCC Unified PC

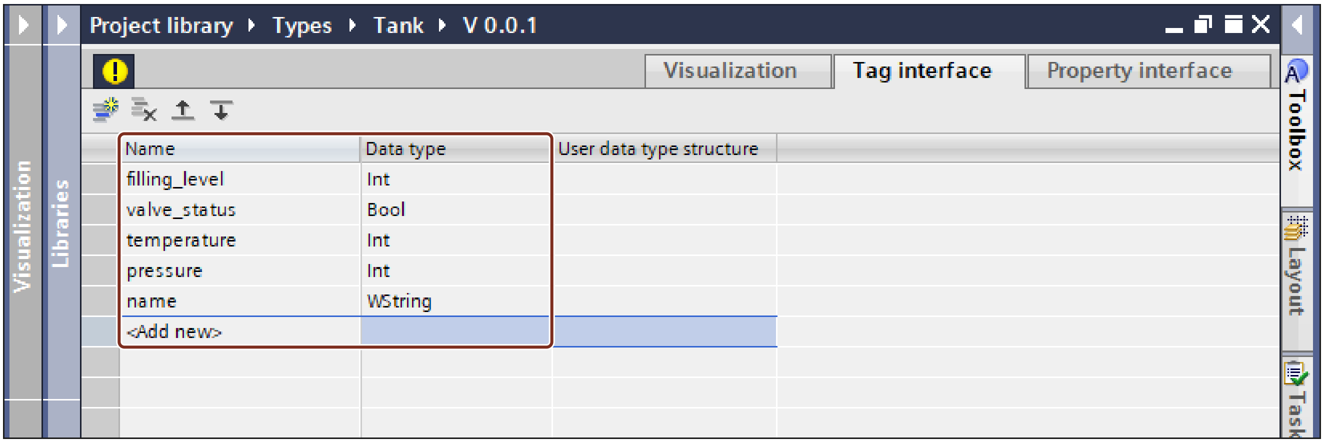
3. Create the "filling_level" property for the faceplate type.



This type version is currently in the "in work" state.

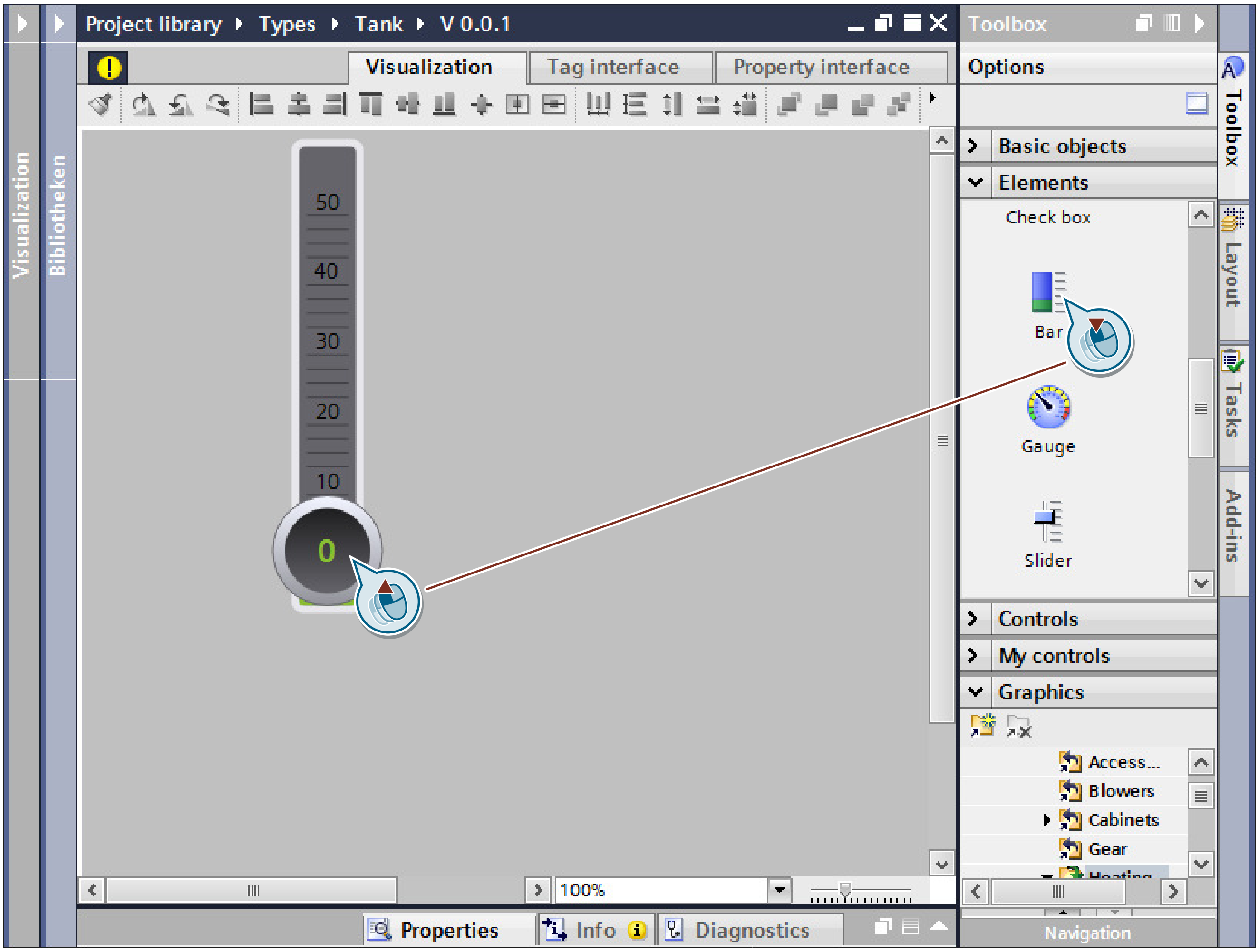
You can release the version or discard the changes and delete the version . Before you release the typ check whether dependent types need to be adapted e .

4. Create the remaining properties listed in the table for the faceplate type.

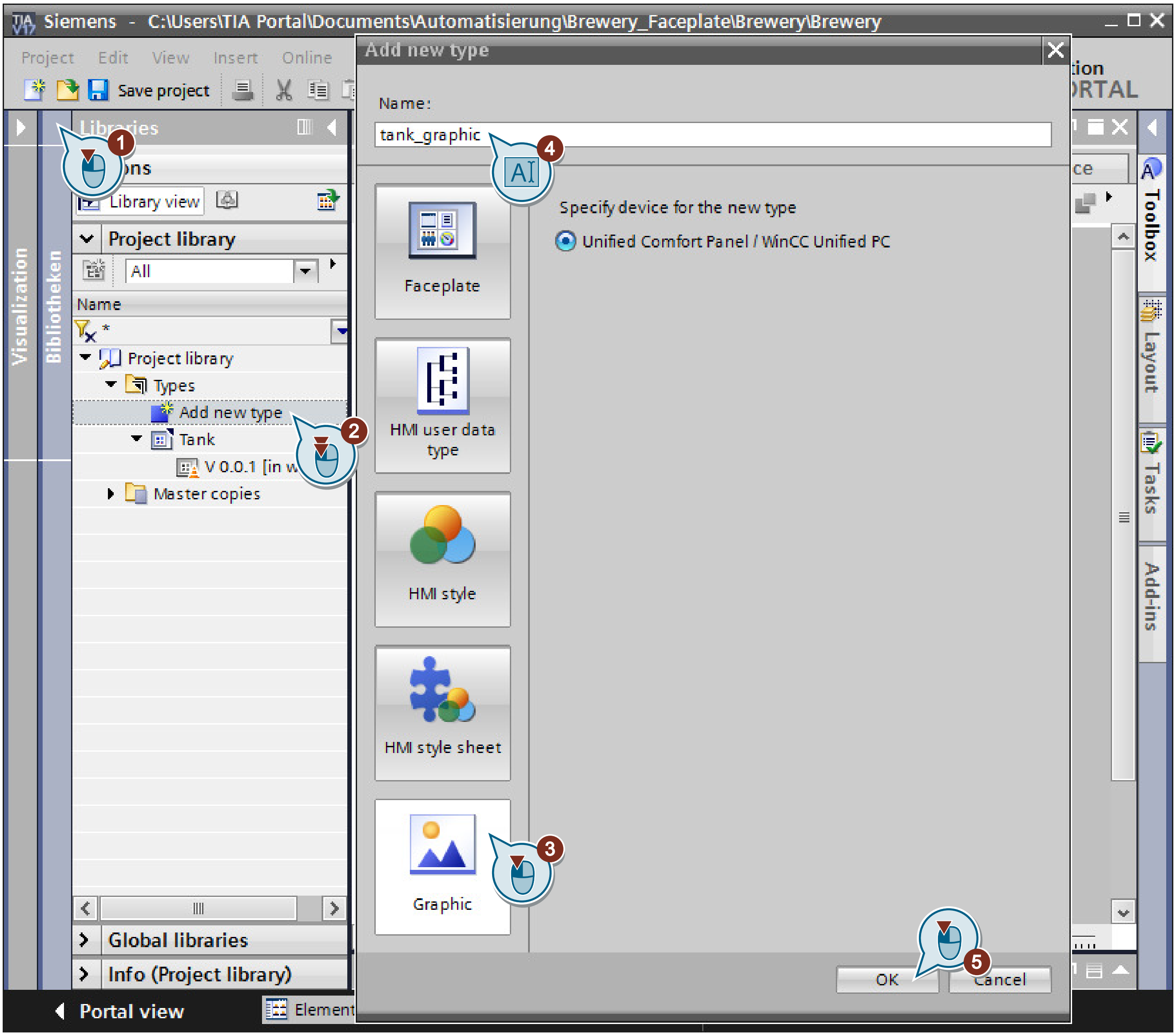


Name	Data type	User data type structure
filling_level	Int	
valve_status	Bool	
temperature	Int	
pressure	Int	
name	WString	
<Add new>		

5. Insert a bar graph for the level.



6. Create a new graphic type for visualizing the storage tank.



Siemens - C:\Users\TIA Portal\Documents\Automatisierung\Brewery_Faceplate\Brewery\Brewery

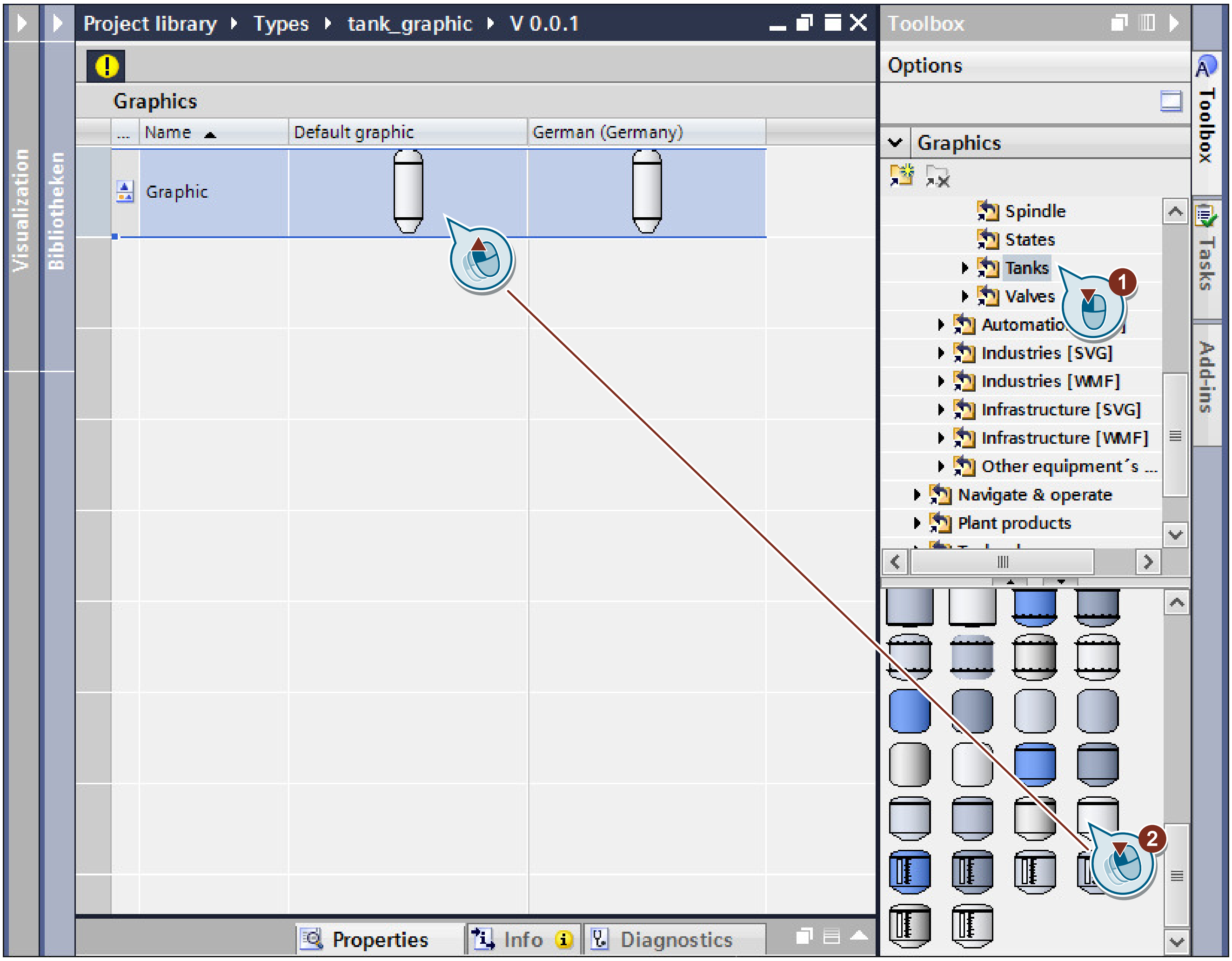
Add new type

Name: tank_graphic

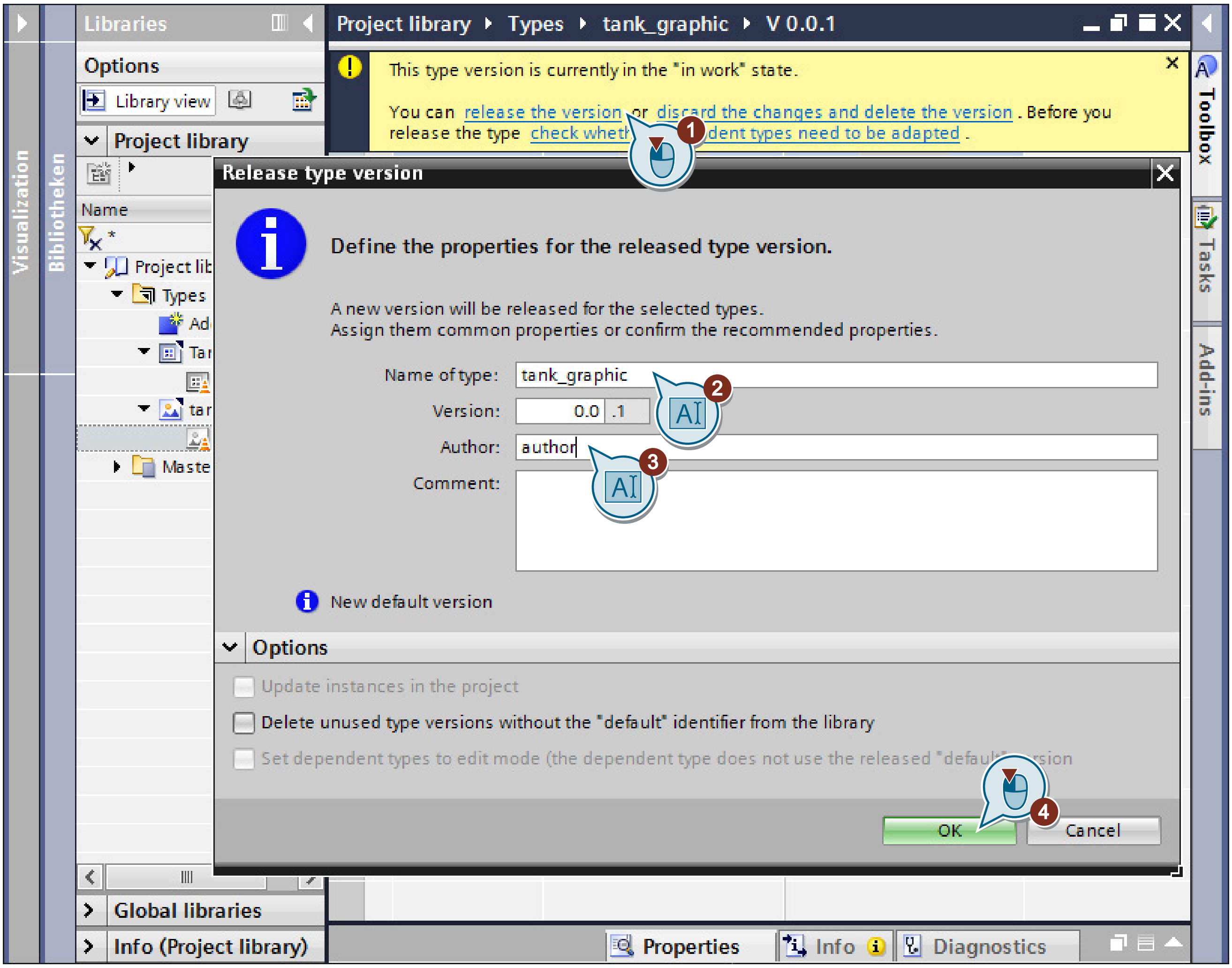
Specify device for the new type

● Unified Comfort Panel / WinCC Unified PC

7. Insert a tank as a standard graphic. You can find the tank in the toolbox under "Graphics > WinCC graphics folder > Equipment > Automation (EMF) > Tanks".



8. Release the version of the graphic type.



This type version is currently in the "in work" state.

You can release the version or discard the changes and delete the version . Before you release the type check wheth… …dent types need to be adapted .

Release type version

Define the properties for the released type version.

A new version will be released for the selected types.
Assign them common properties or confirm the recommended properties.

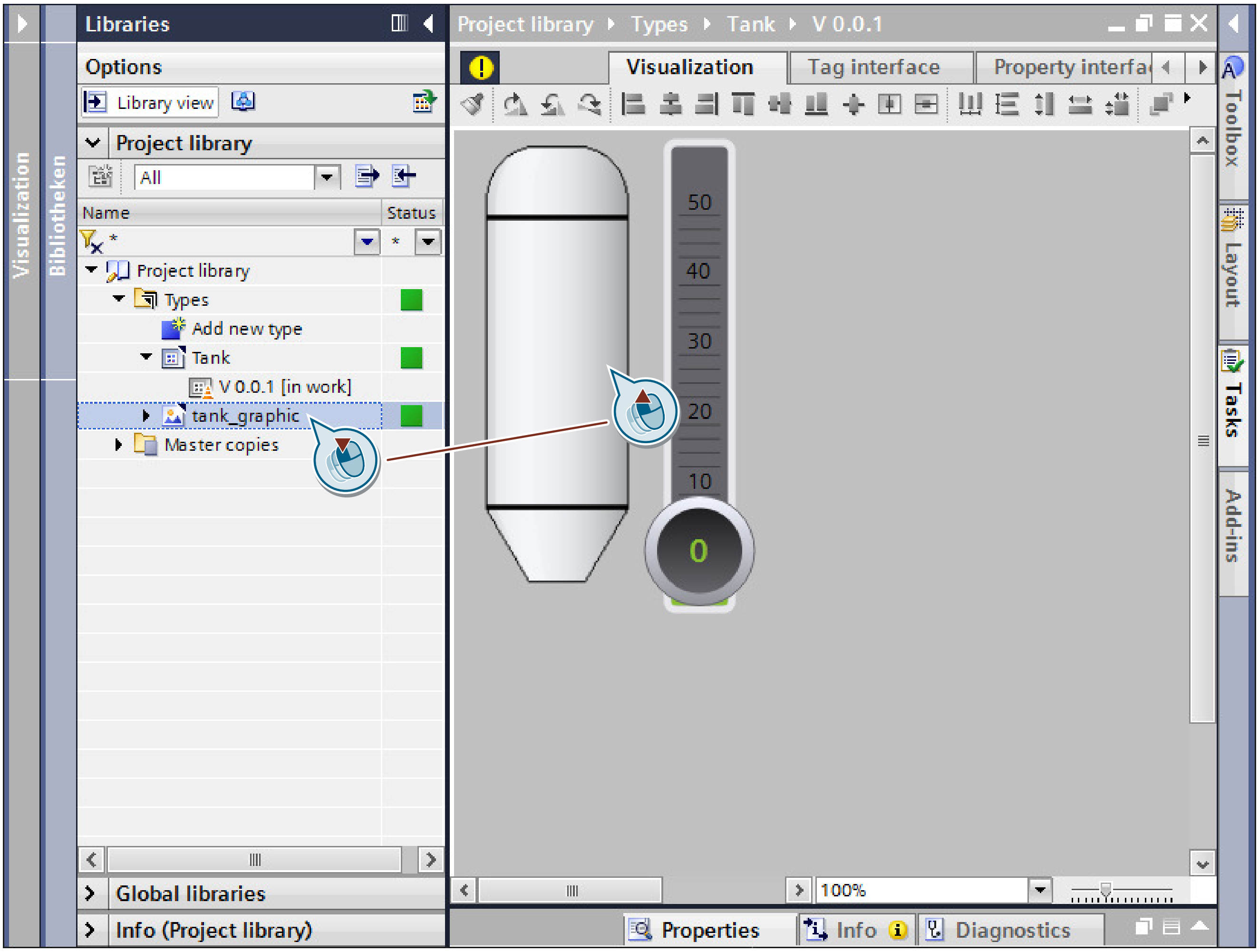
Name of type: tank_graphic
Version: 0.0 .1
Author: author
Comment:

New default version

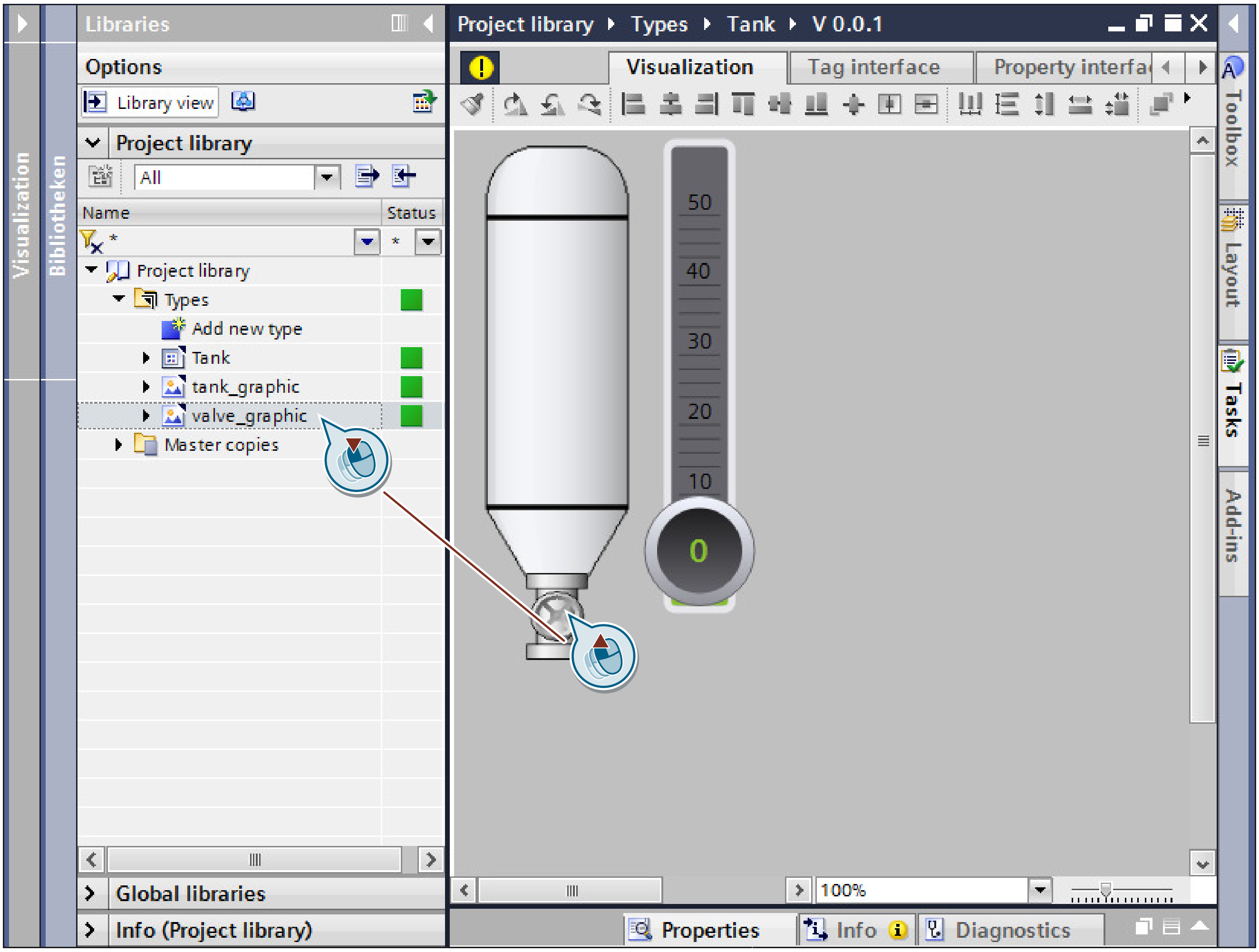
Options

□ Update instances in the project
□ Delete unused type versions without the "default" identifier from the library
□ Set dependent types to edit mode (the dependent type does not use the released "defaul… …rsion

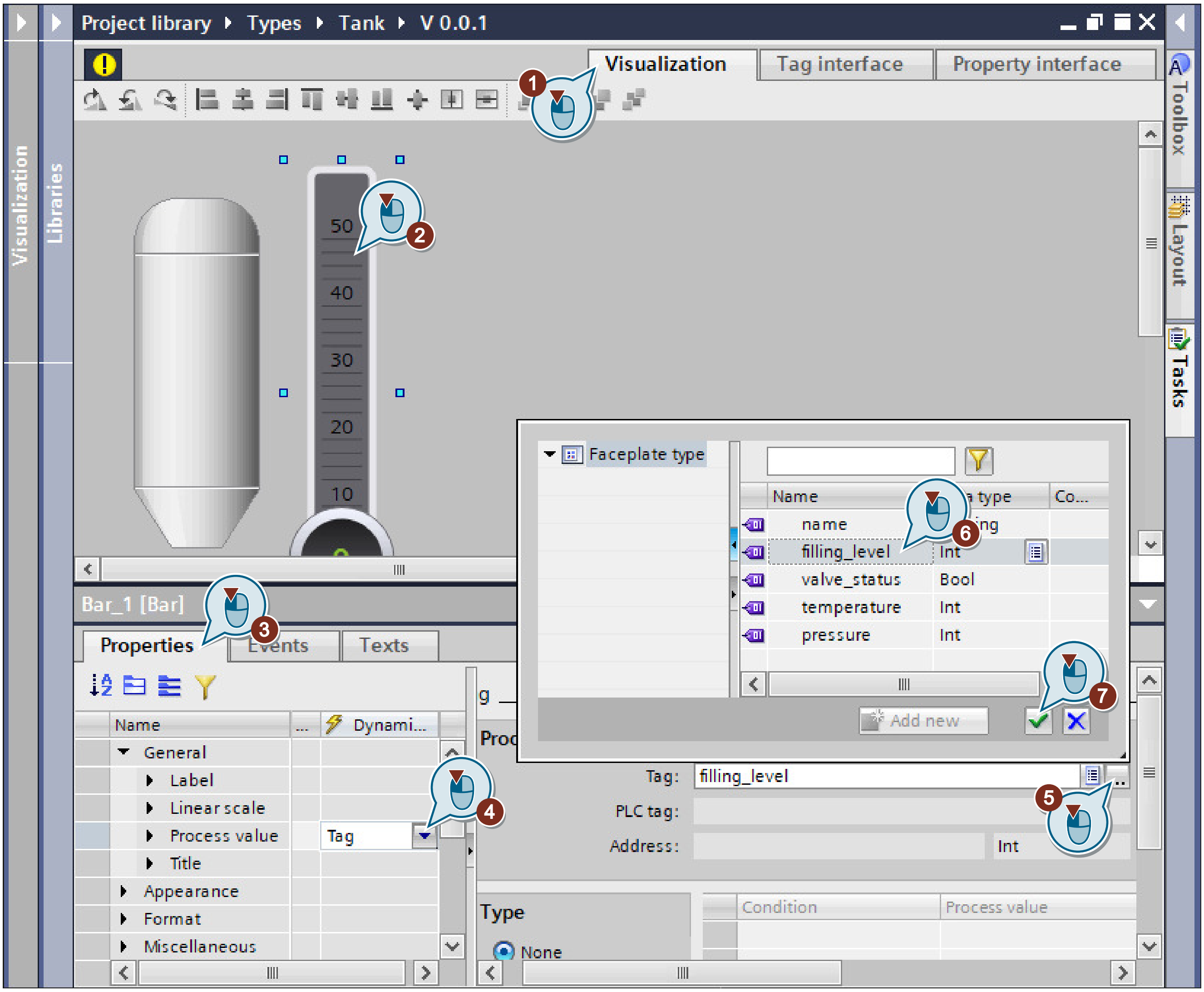
9. Insert the graphic type into the screen.



10. Create a graphic type for the valve in the same way and insert it into the screen. You can find the valve in the toolbox under "Graphics > WinCC graphics folder > Equipment > Automation (EMF) > Valves".



11. Link the property for displaying the fill level with the bar graph.



Name	...a type	Co...
name	...ing	
filling_level	Int	
valve_status	Bool	
temperature	Int	
pressure	Int	

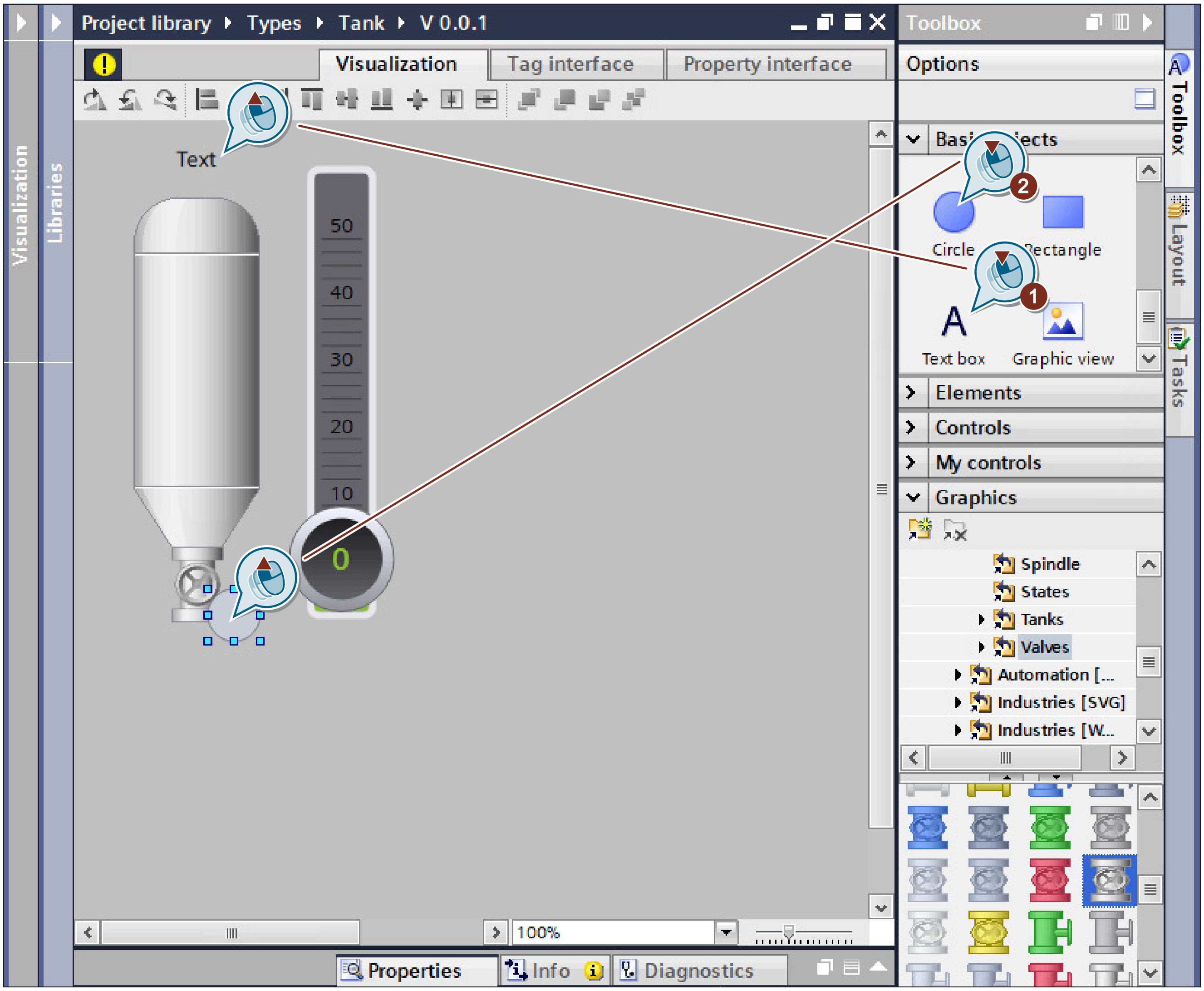
Name	...	Dynami...
General		
Label		
Linear scale		
Process value		Tag
Title		
Appearance		
Format		
Miscellaneous		

Tag: filling_level

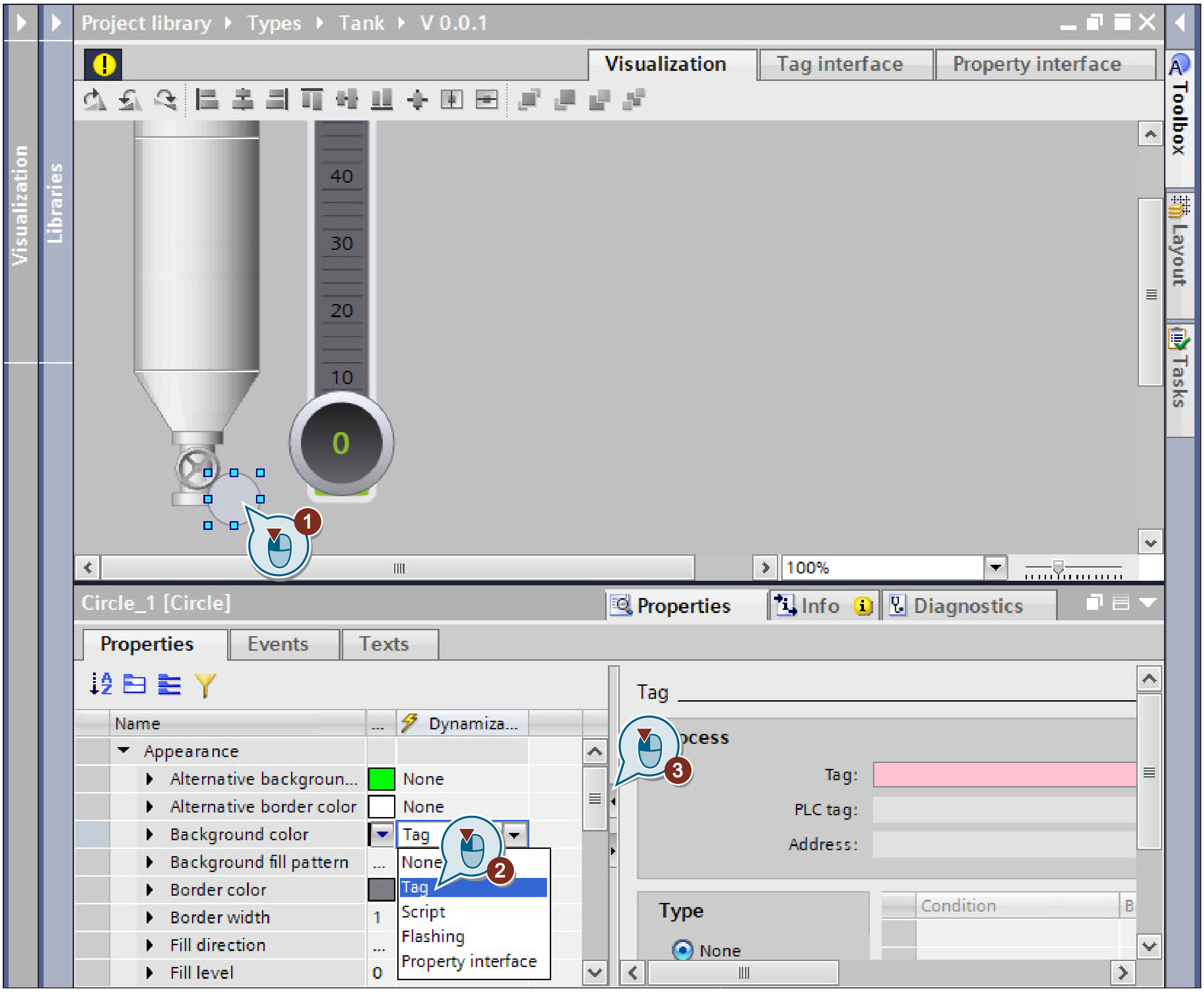
PLC tag:

Address: Int

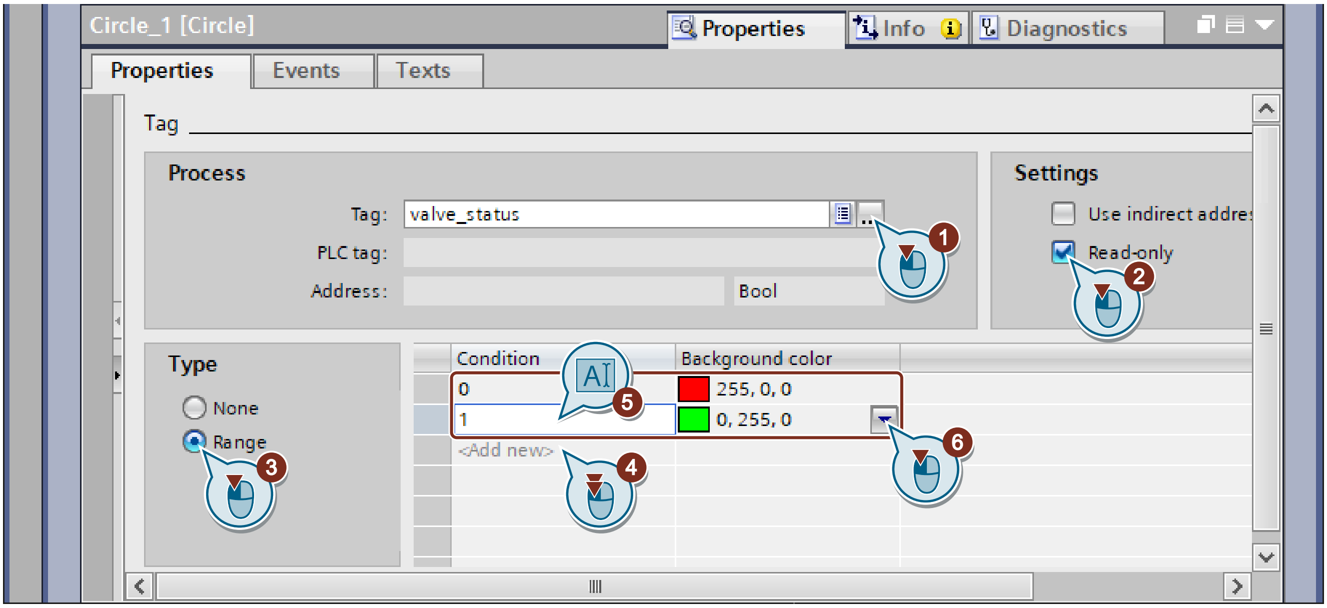
12. Insert a text field for the designation and a circle. The text field is not yet interconnected in this Getting Started.



13. Configure the circle so that its background color can be changed using a tag (1, 2). Finally, show the field for the parameter assignment of the tags (3).



14. Specify the valve status as a tag for the color and determine which color is applied for which state of the tag.



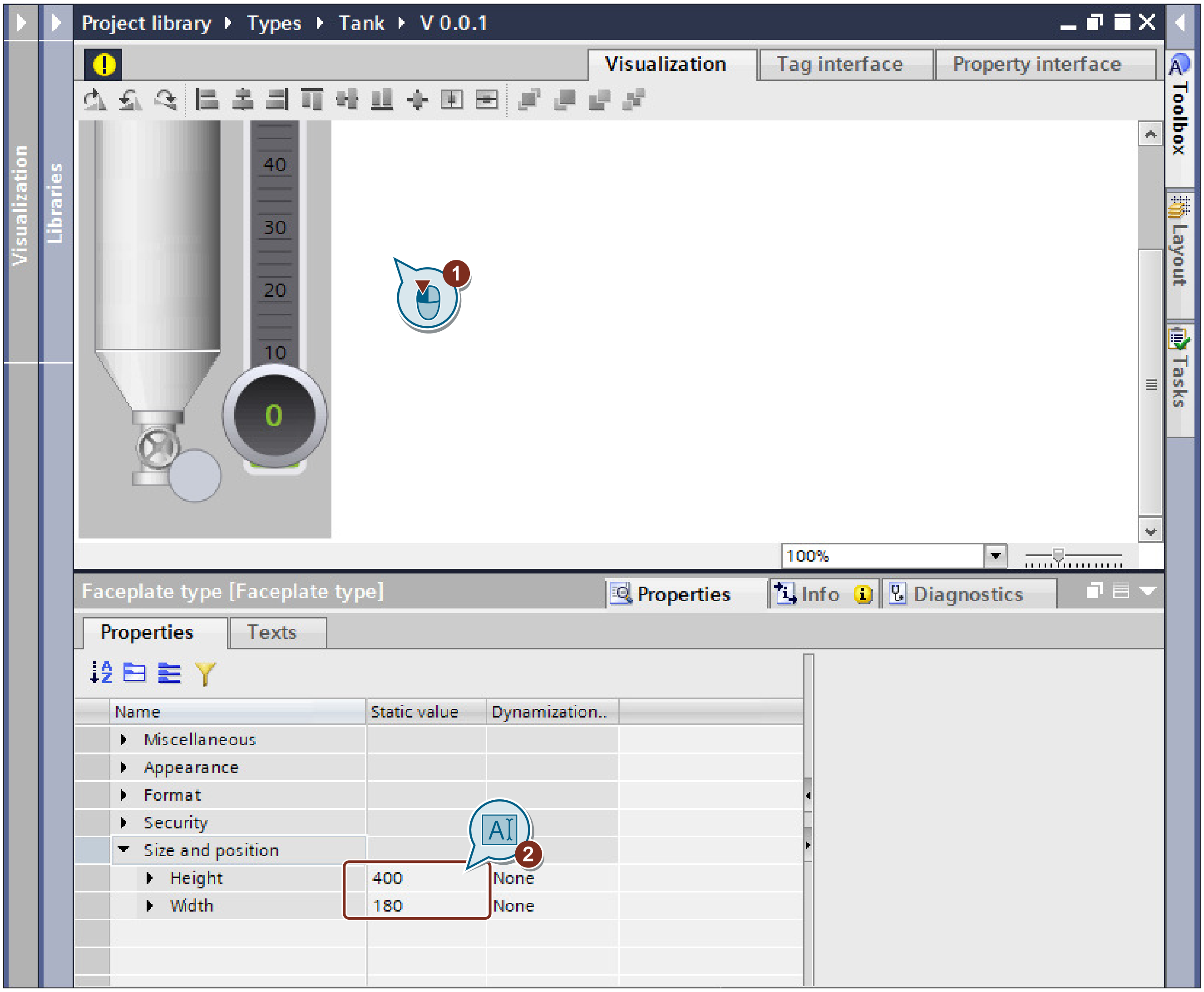
Tag: valve_status
PLC tag:
Address: Bool

□ Use indirect addres
☑ Read-only

○ None
● Range

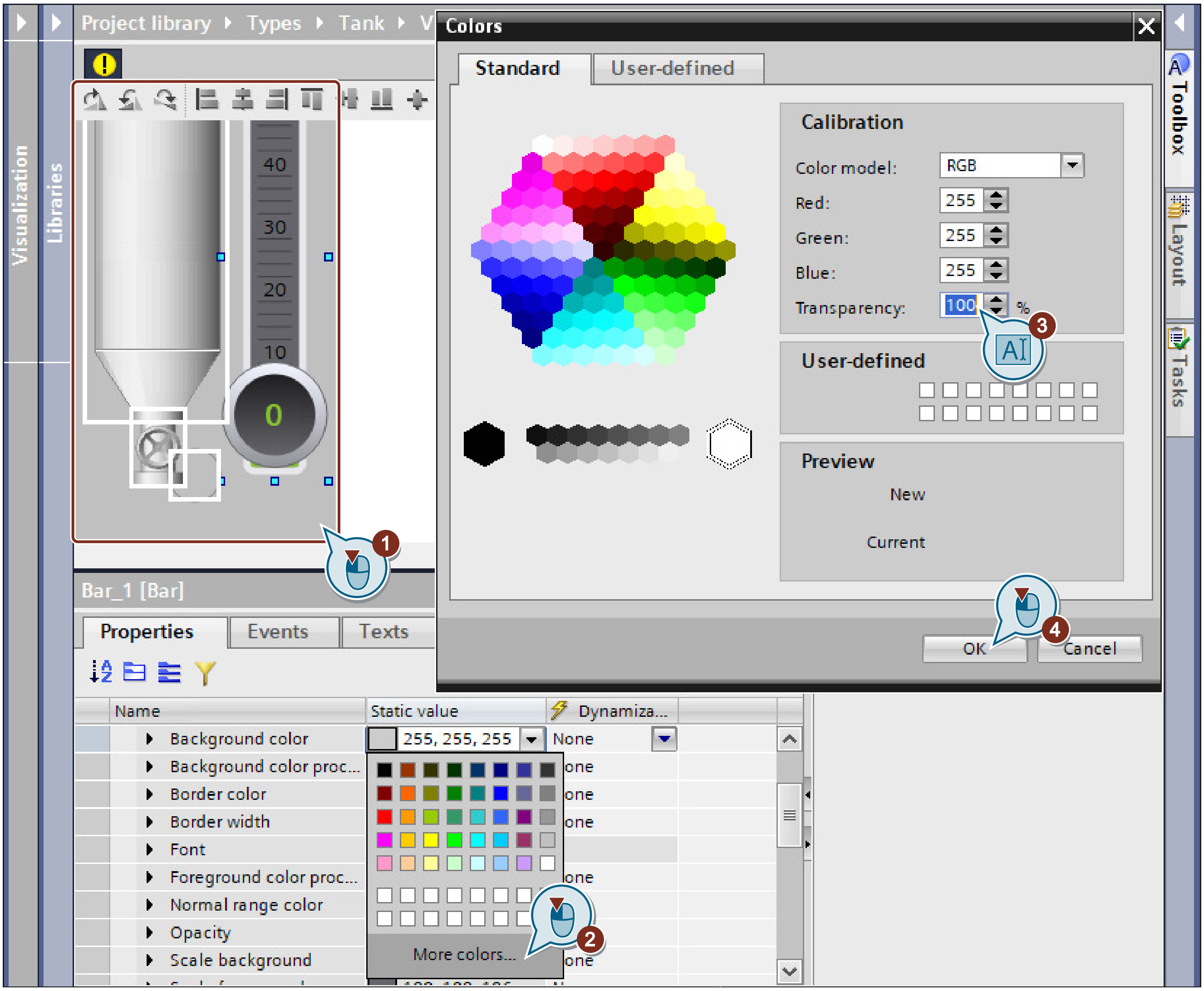
Condition	Background color
0	🟥 255, 0, 0
1	🟩 0, 255, 0
<Add new>	

15. Adjust the size of the faceplate.

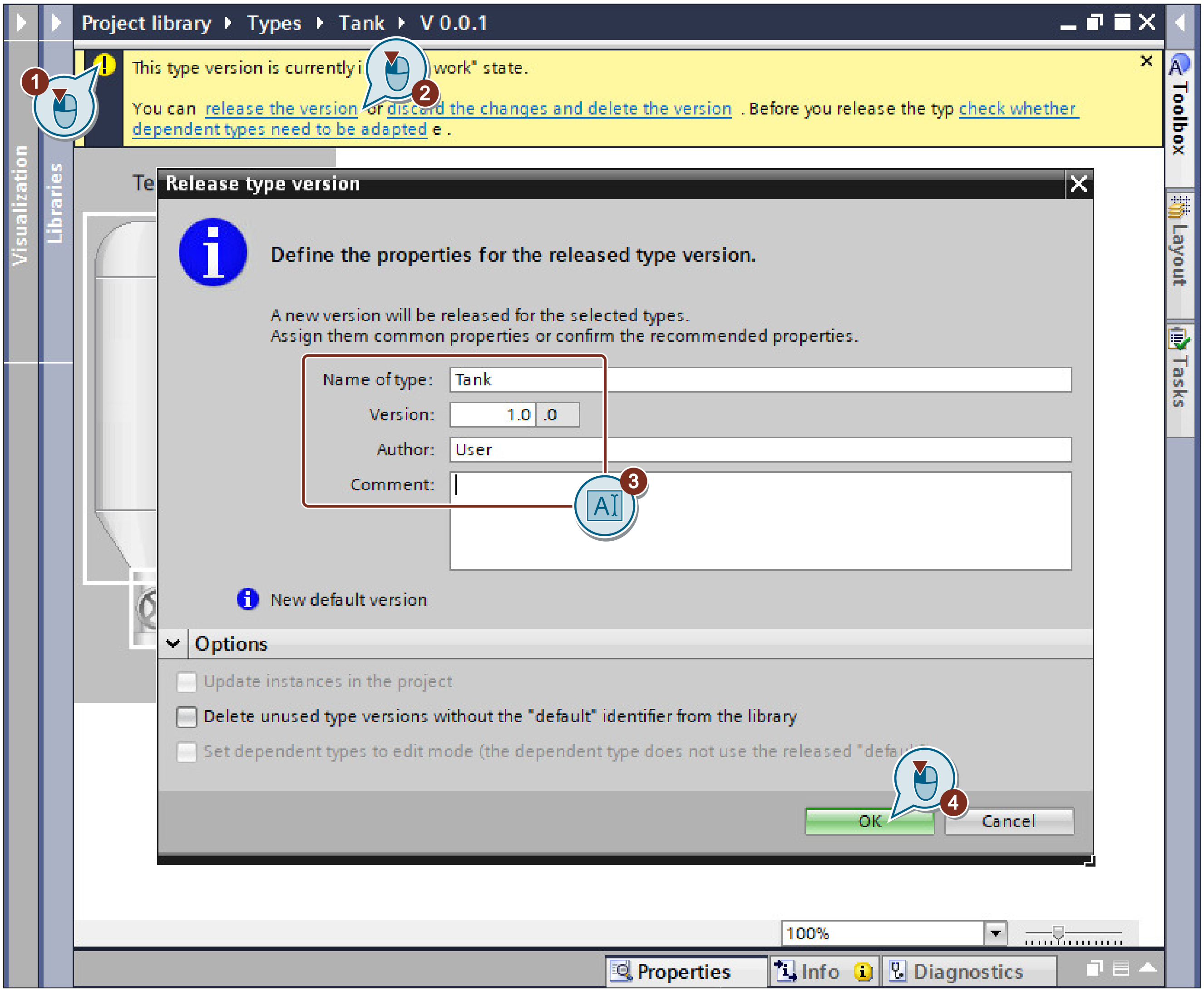


Name	Static value	Dynamization..
▸ Miscellaneous		
▸ Appearance		
▸ Format		
▸ Security		
▾ Size and position		
▸ Height	400	None
▸ Width	180	None

16. Then select with <Shift+Click> the objects "Tank", "Valve" and "Level indicator" and set the transparency to "100%".



17. Share the faceplate type in the library.



This type version is currently i[n "in] work" state.

You can release the version or discard the changes and delete the version . Before you release the typ check whether dependent types need to be adapted e .

Release type version

Define the properties for the released type version.

A new version will be released for the selected types.
Assign them common properties or confirm the recommended properties.

Name of type:	Tank
Version:	1.0 .0
Author:	User
Comment:	

New default version

Options

□ Update instances in the project

□ Delete unused type versions without the "default" identifier from the library

□ Set dependent types to edit mode (the dependent type does not use the released "defa[ult"]

OK Cancel

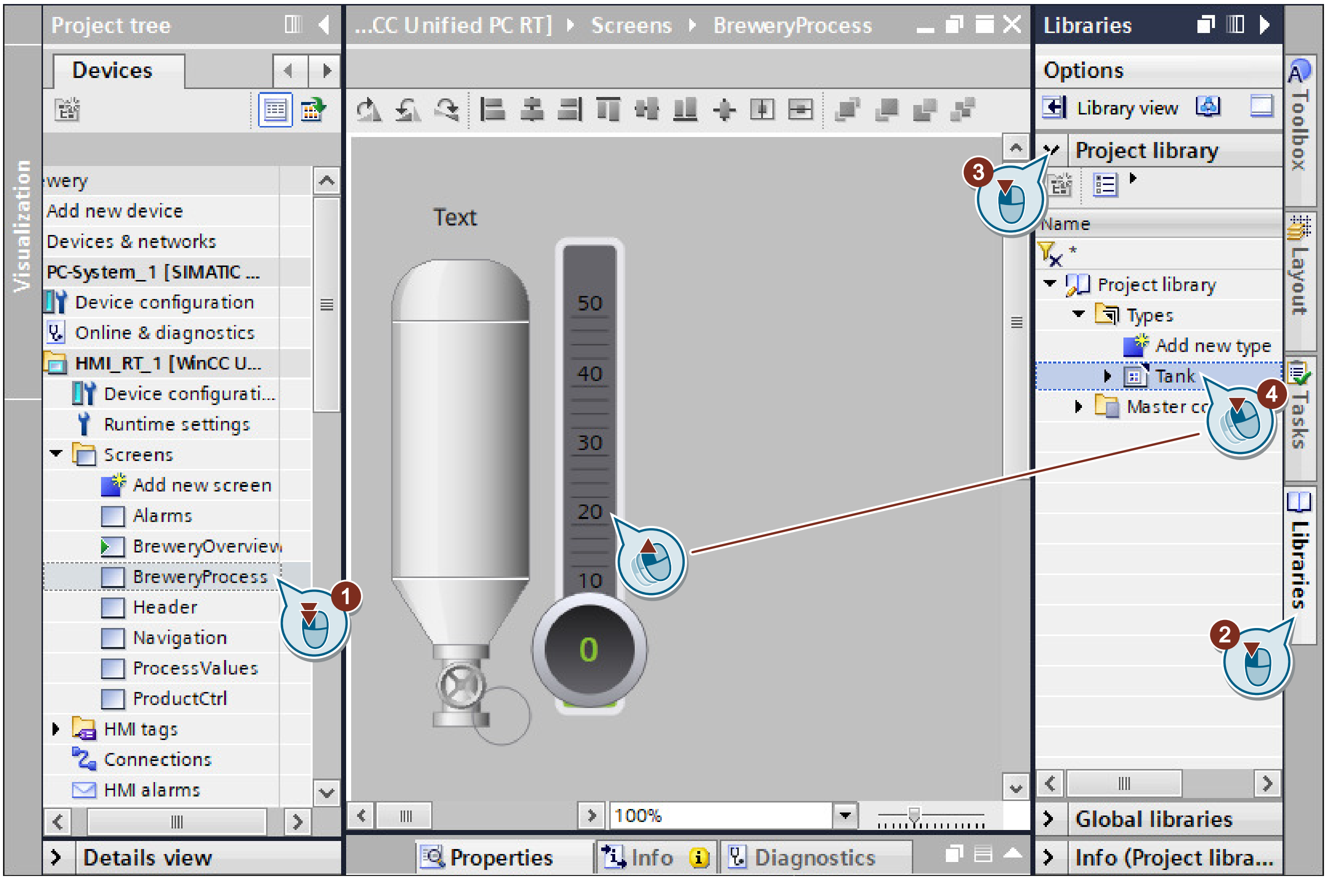
2.4.3 Configuring faceplate instances

Introduction

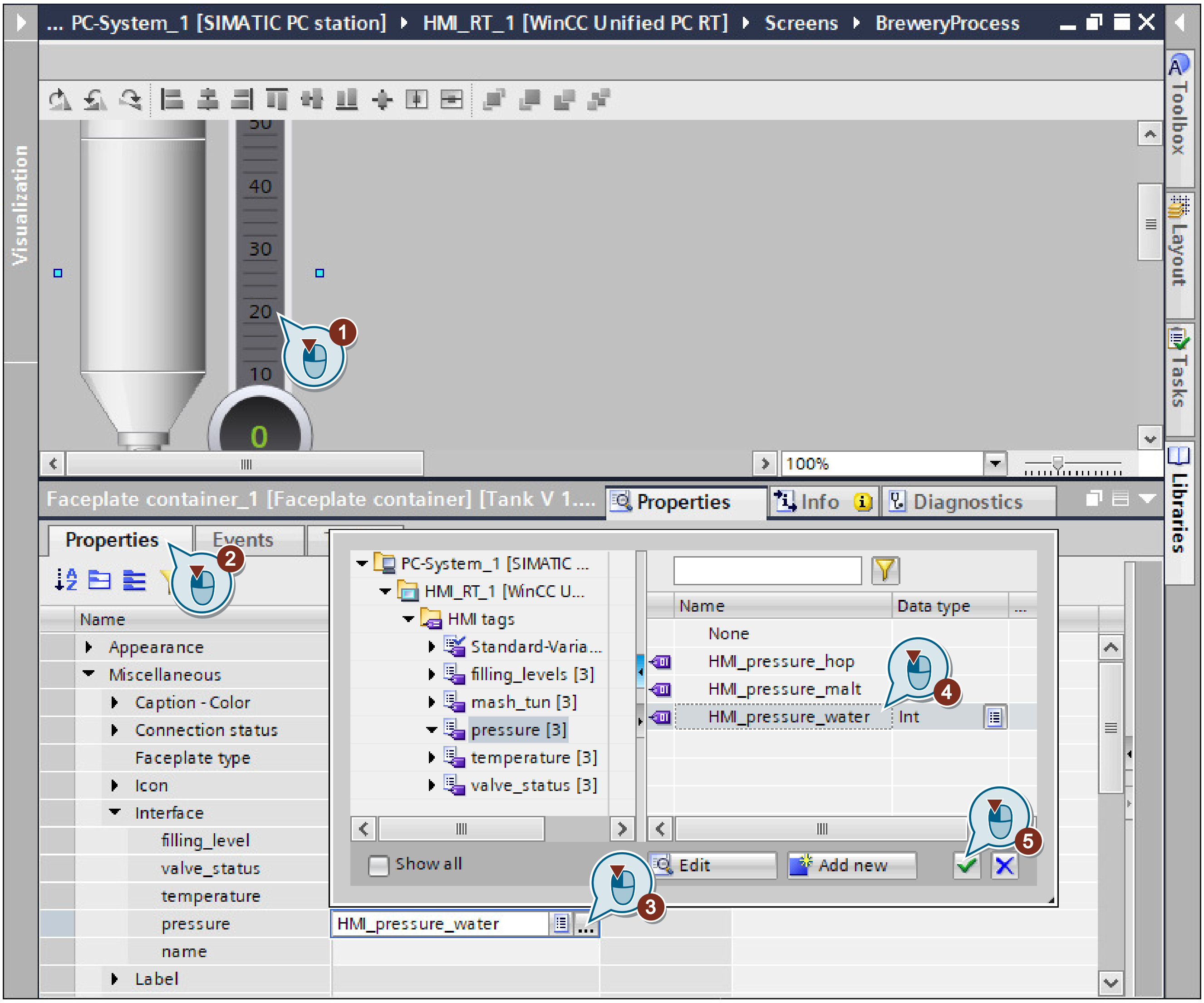
The storage tanks for the beer ingredients should now be visualized on the basis of the created faceplate type. For each storage tank, insert an instance of the faceplate type and parameterize the properties with the corresponding tags.

Procedure

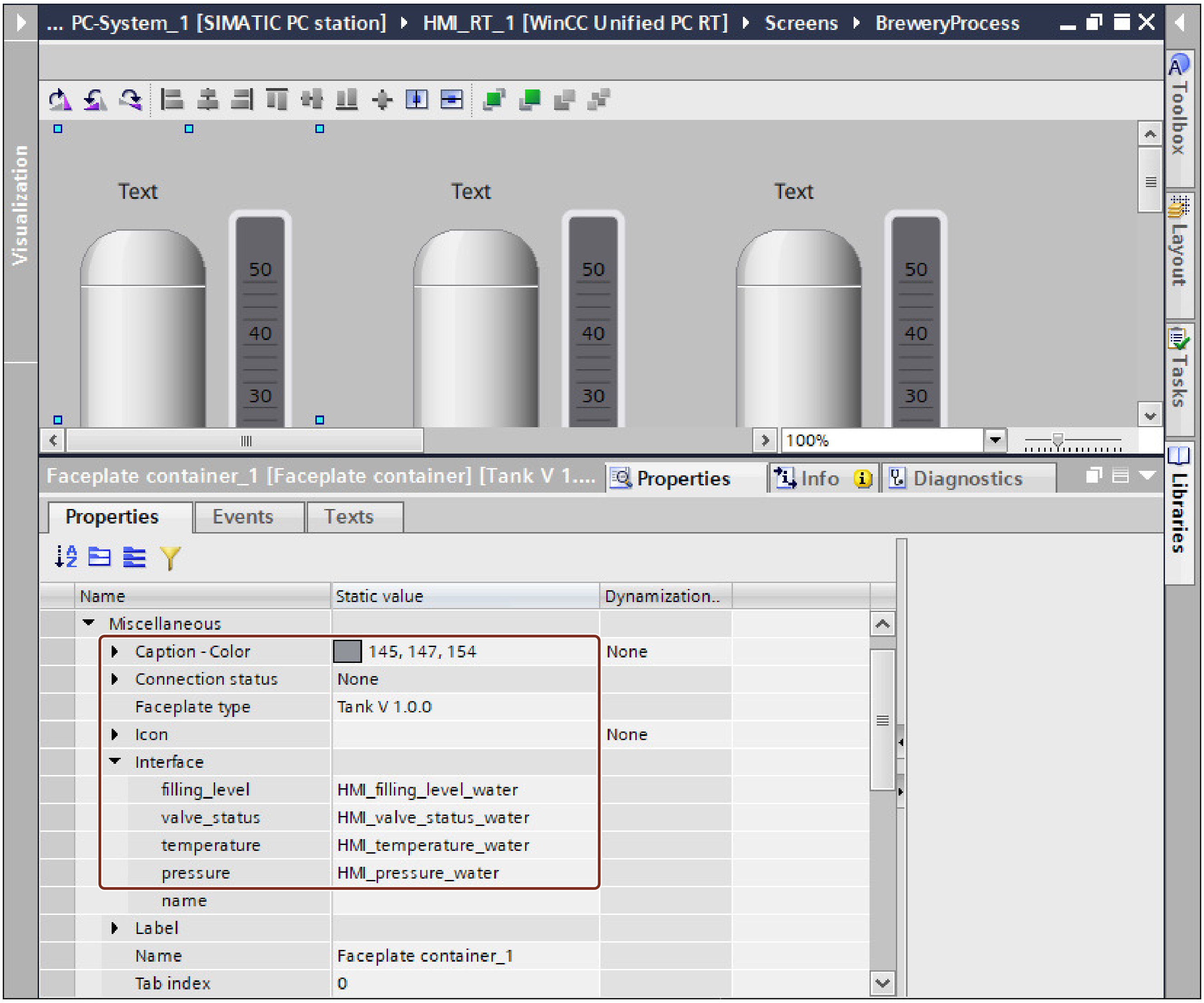
1. Open the HMI screen "BreweryProcess" and use the faceplate type to create the instance for the storage tank that contains water.



2. Assign parameters for the "pressure" properties of the faceplate instance with the "HMI_pressure_water" tag.


... PC-System_1 [SIMATIC PC station] ▸ HMI_RT_1 [WinCC Unified PC RT] ▸ Screens ▸ BreweryProcess
Visualization
Toolbox
Layout
Tasks
Libraries
50
40
30
20
10
0
1
100%
Faceplate container_1 [Faceplate container] [Tank V 1....
Properties
Info
Diagnostics
Properties
Events
2
Name
Appearance
Miscellaneous
Caption - Color
Connection status
Faceplate type
Icon
Interface
filling_level
valve_status
temperature
pressure
name
Label
PC-System_1 [SIMATIC ...
HMI_RT_1 [WinCC U...
HMI tags
Standard-Varia...
filling_levels [3]
mash_tun [3]
pressure [3]
temperature [3]
valve_status [3]
Show all
Name
Data type
None
HMI_pressure_hop
HMI_pressure_malt
HMI_pressure_water
Int
4
Edit
Add new
5
3
HMI_pressure_water

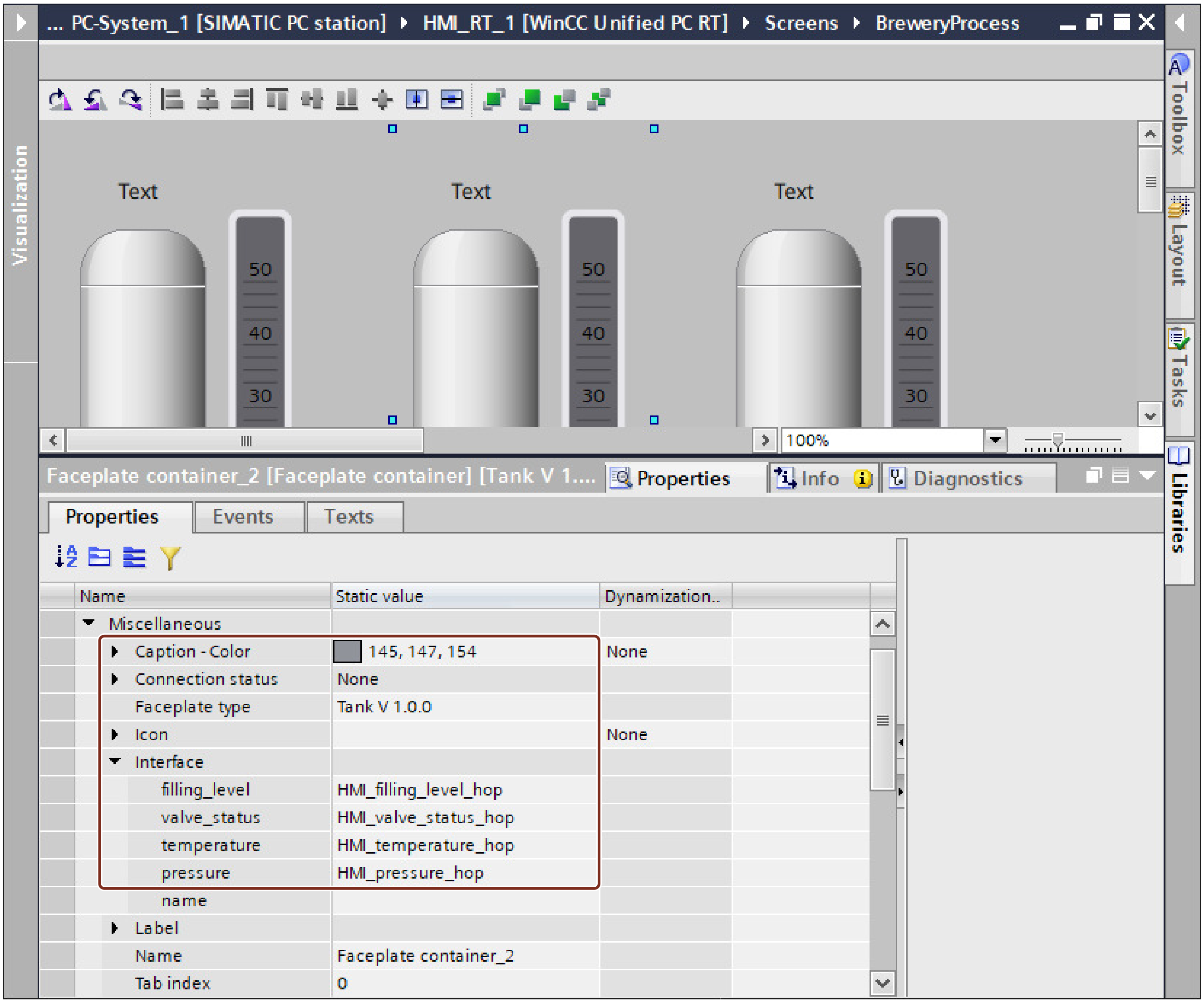
3. Assign parameters for the remaining properties of the faceplate instance in the same way.



... PC-System_1 [SIMATIC PC station] ▸ HMI_RT_1 [WinCC Unified PC RT] ▸ Screens ▸ BreweryProcess
Visualization
Text
50
40
30
Text
50
40
30
Text
50
40
30
100%
Toolbox
Layout
Tasks
Libraries
Faceplate container_1 [Faceplate container] [Tank V 1....
Properties
Info
Diagnostics
Properties
Events
Texts

Name	Static value	Dynamization..
▾ Miscellaneous		
▸ Caption - Color	145, 147, 154	None
▸ Connection status	None	
Faceplate type	Tank V 1.0.0	
▸ Icon		None
▾ Interface		
filling_level	HMI_filling_level_water	
valve_status	HMI_valve_status_water	
temperature	HMI_temperature_water	
pressure	HMI_pressure_water	
name		
▸ Label		
Name	Faceplate container_1	
Tab index	0	

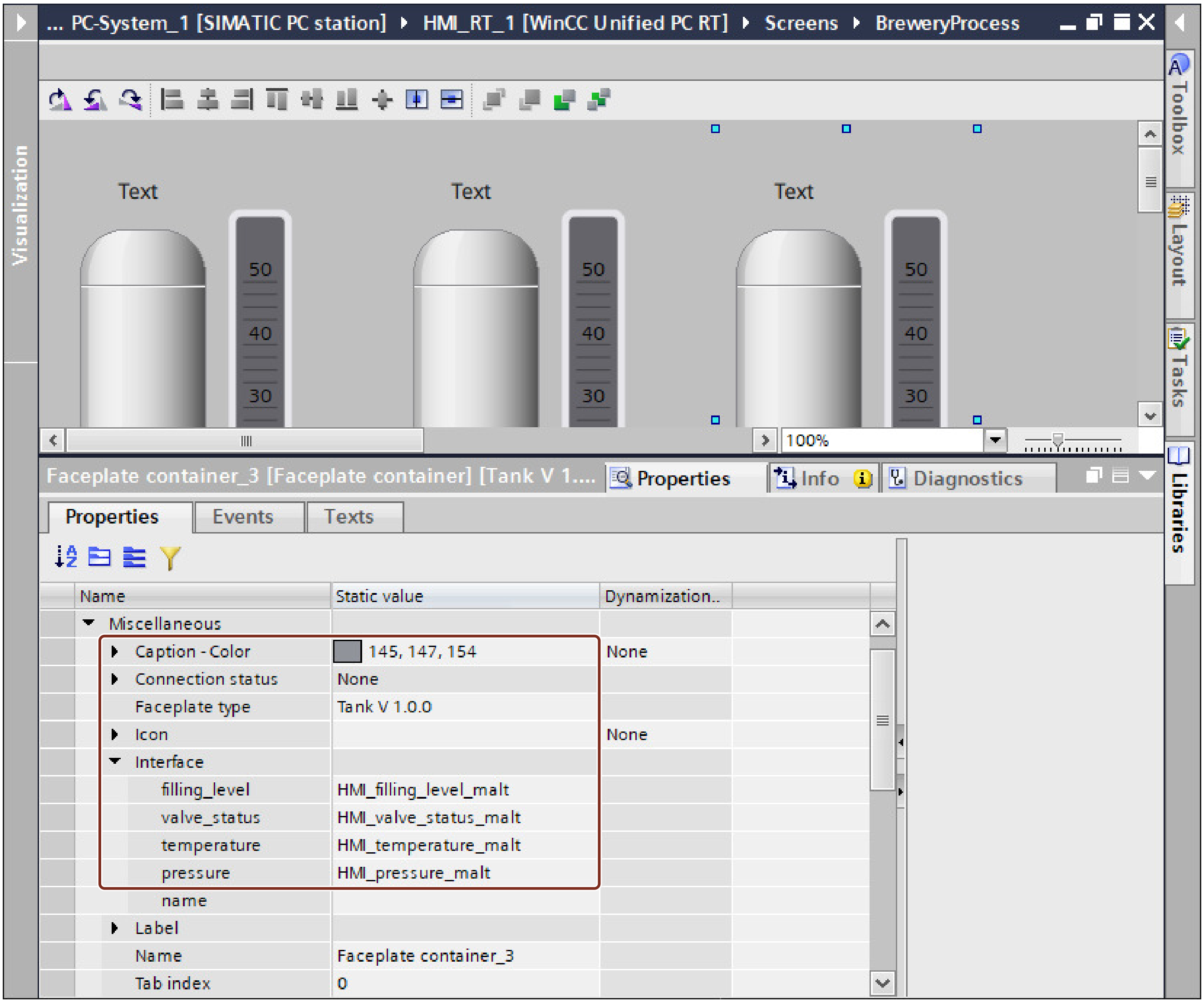
4. Visualize and assign parameters for the storage tank for "Hops" in the same way.



Faceplate container_2 [Faceplate container] [Tank V 1....

Name	Static value	Dynamization..
Miscellaneous		
Caption - Color	145, 147, 154	None
Connection status	None	
Faceplate type	Tank V 1.0.0	
Icon		None
Interface		
filling_level	HMI_filling_level_hop	
valve_status	HMI_valve_status_hop	
temperature	HMI_temperature_hop	
pressure	HMI_pressure_hop	
name		
Label		
Name	Faceplate container_2	
Tab index	0	

5. Visualize and assign parameters for the storage tank for "Malt" in the same way.


... PC-System_1 [SIMATIC PC station] ▸ HMI_RT_1 [WinCC Unified PC RT] ▸ Screens ▸ BreweryProcess
Visualization
Toolbox
Layout
Tasks
Libraries
Text
Text
Text
50
40
30
50
40
30
50
40
30
100%
Faceplate container_3 [Faceplate container] [Tank V 1....
Properties
Info
Diagnostics
Properties
Events
Texts

Name	Static value	Dynamization..
Miscellaneous		
Caption - Color	▨ 145, 147, 154	None
Connection status	None	
Faceplate type	Tank V 1.0.0	
Icon		None
Interface		
filling_level	HMI_filling_level_malt	
valve_status	HMI_valve_status_malt	
temperature	HMI_temperature_malt	
pressure	HMI_pressure_malt	
name		
Label		
Name	Faceplate container_3	
Tab index	0	

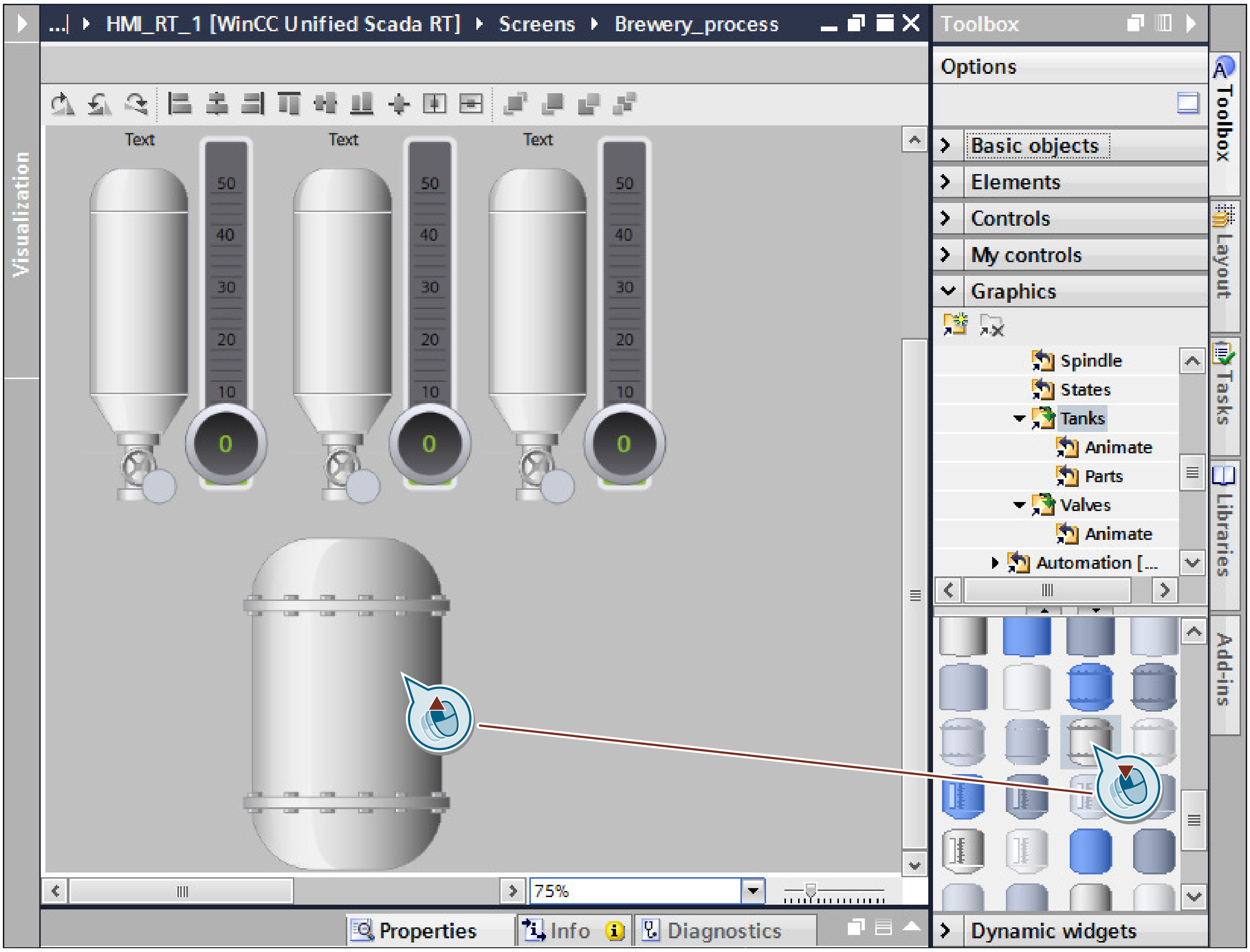
2.4.4 Configure the HMI screen

Introduction

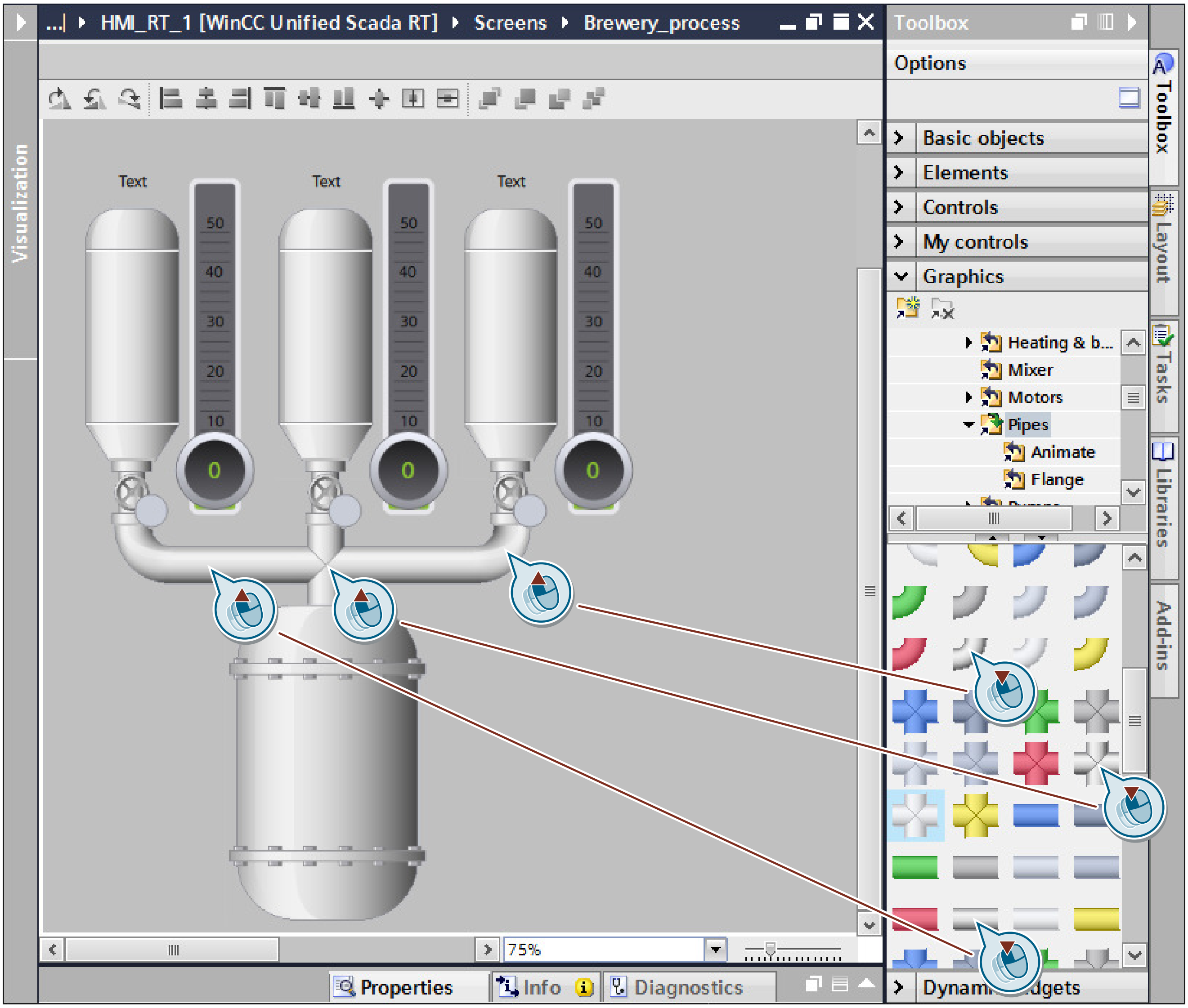
In the following, the already created process screen, which contains the three storage tanks for water, hops and malt, is to be expanded to include the remaining components of the brewing process. For this purpose, a mash tun is inserted and connected to the storage tanks via pipes. A heat source that heats the mash tun is inserted. In addition to the graphic elements, display elements for the process values of the mash tun are to be added.

Procedure

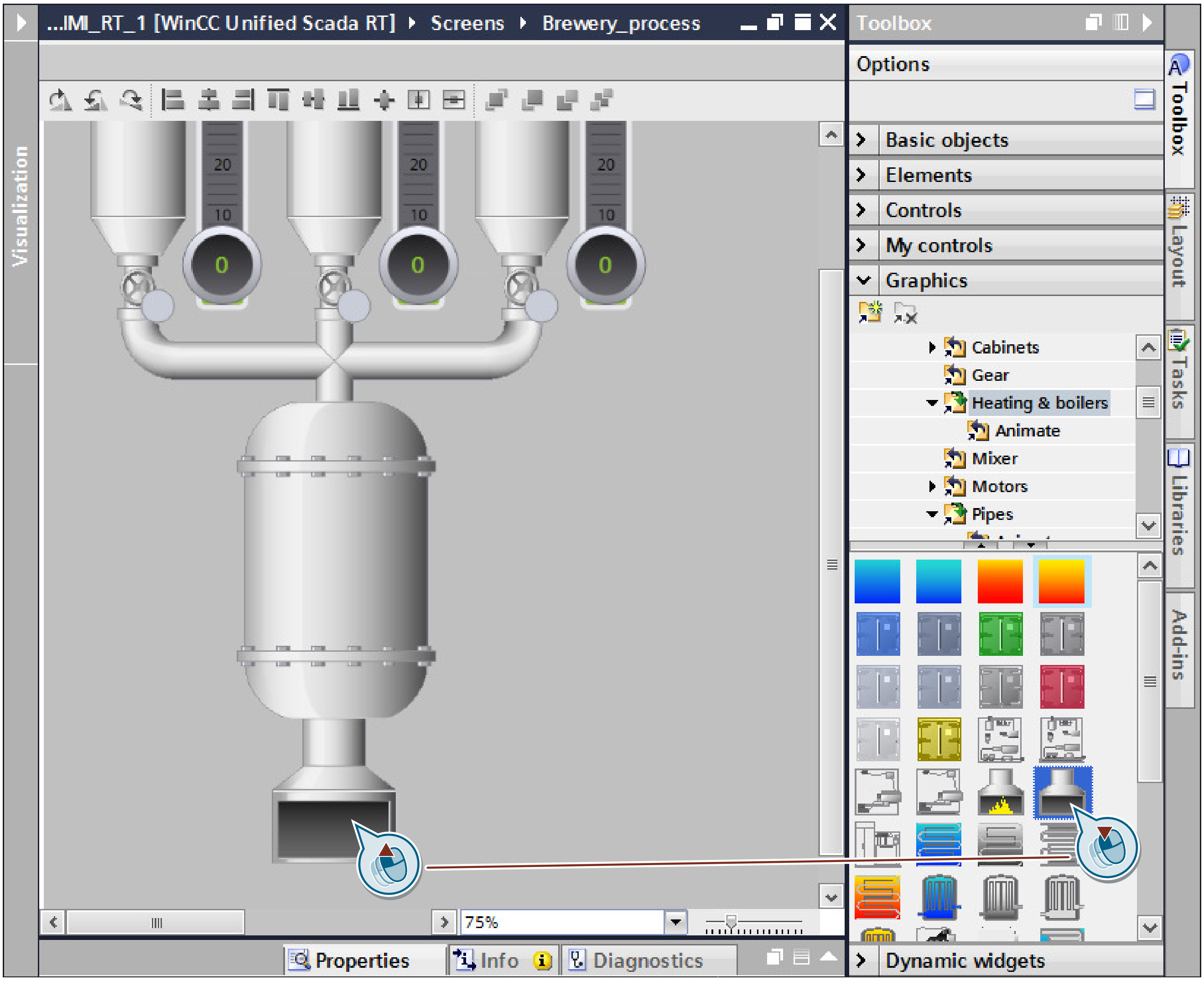
1. Insert the displayed tank, which represents the mash tun.



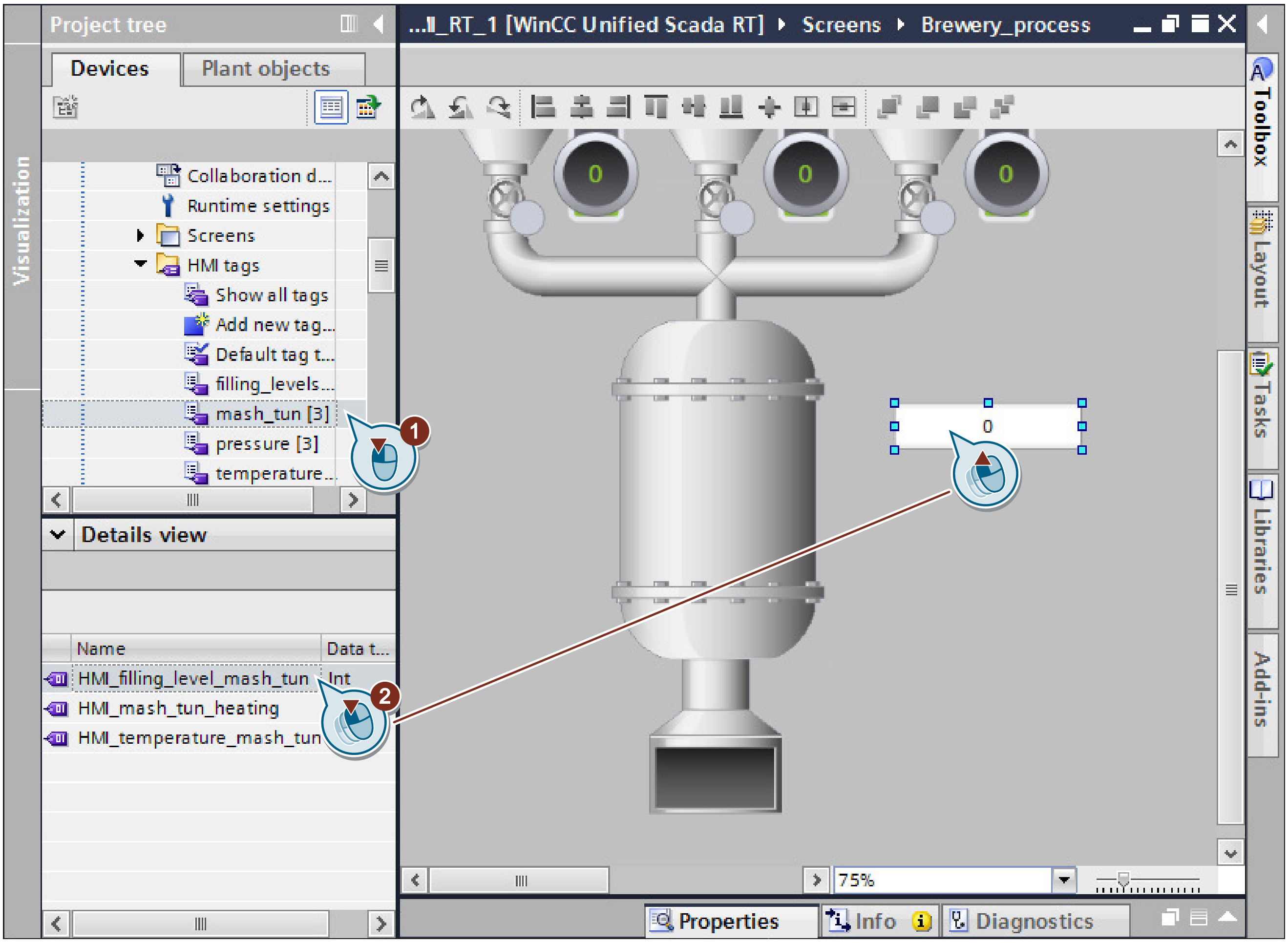
2. Use pipe displays to complete the HMI screen.


...| ▸ HMI_RT_1 [WinCC Unified Scada RT] ▸ Screens ▸ Brewery_process
Visualization
Text
Text
Text
50
40
30
20
10
0
75%
Properties
Info
Diagnostics
Toolbox
Options
Basic objects
Elements
Controls
My controls
Graphics
Heating & b...
Mixer
Motors
Pipes
Animate
Flange
Dynamic widgets
Toolbox
Layout
Tasks
Libraries
Add-ins

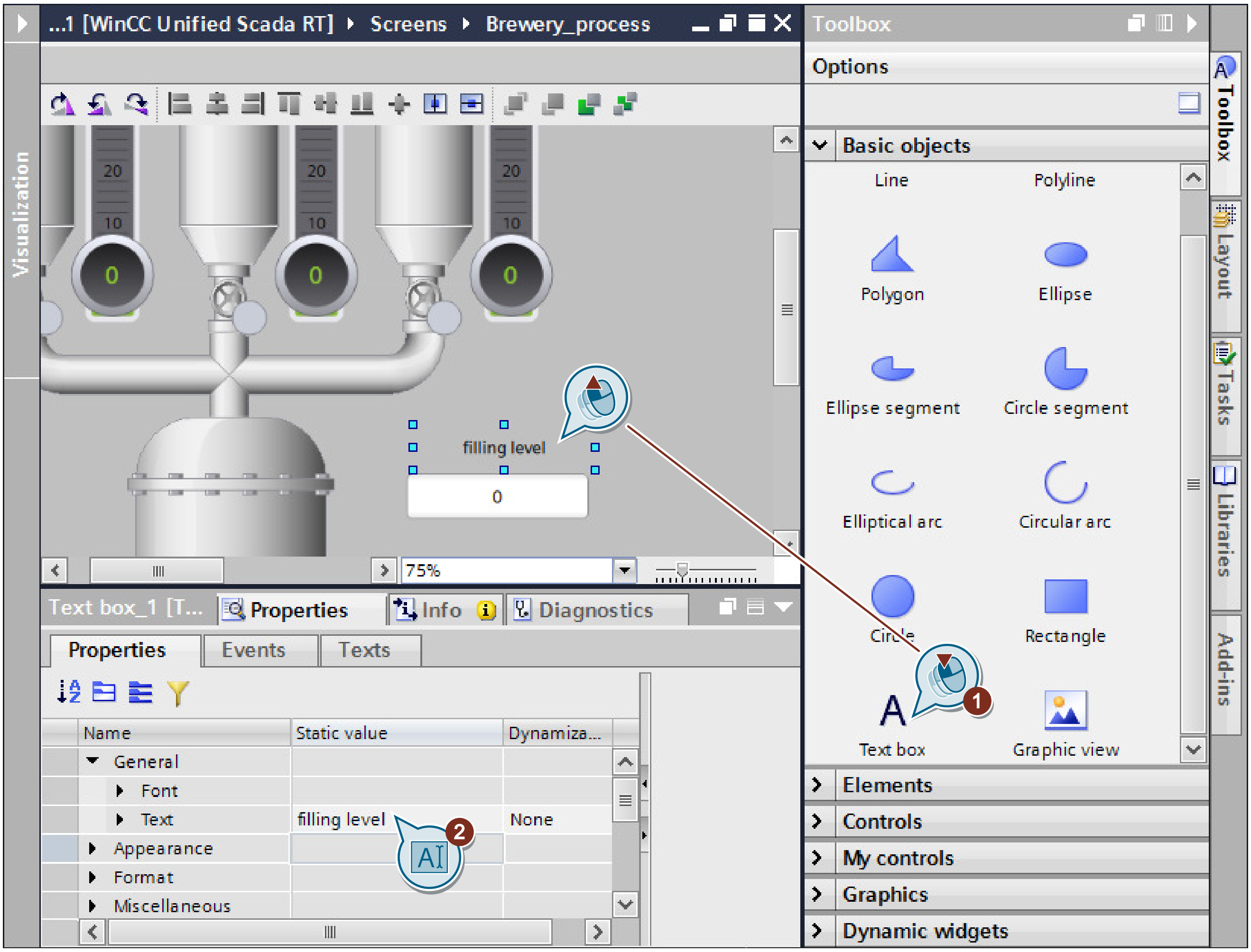
3. Insert the graphic of the displayed heat source.



4. Insert a value display for the fill level tag "HMI_filling_level_mash_tun" of the tank. When you drag a tag into the HMI screen, an I/O field is automatically created that is linked to this tag.



5. Insert a text field to label the tag.



Name	Static value	Dynamiza...
General		
Font		
Text	filling level	None
Appearance		
Format		
Miscellaneous		

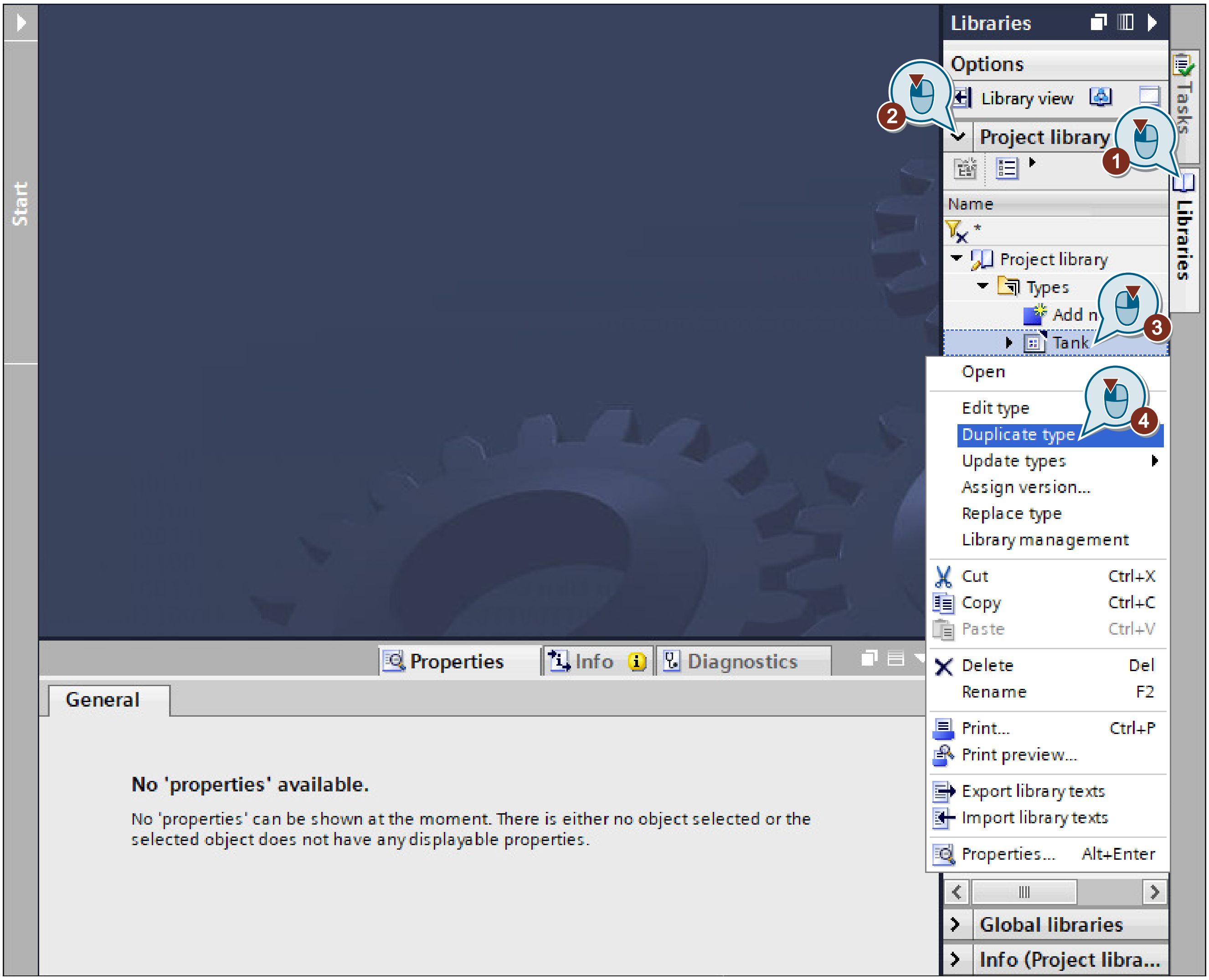
2.4.5 Configuring pop-up windows for faceplates

Introduction

In the following, a pop-up window for faceplates is to be configured in which additional information on the fill level, temperature and pressure of the respective storage tank is displayed when you click on one of the three storage tanks.

Procedure

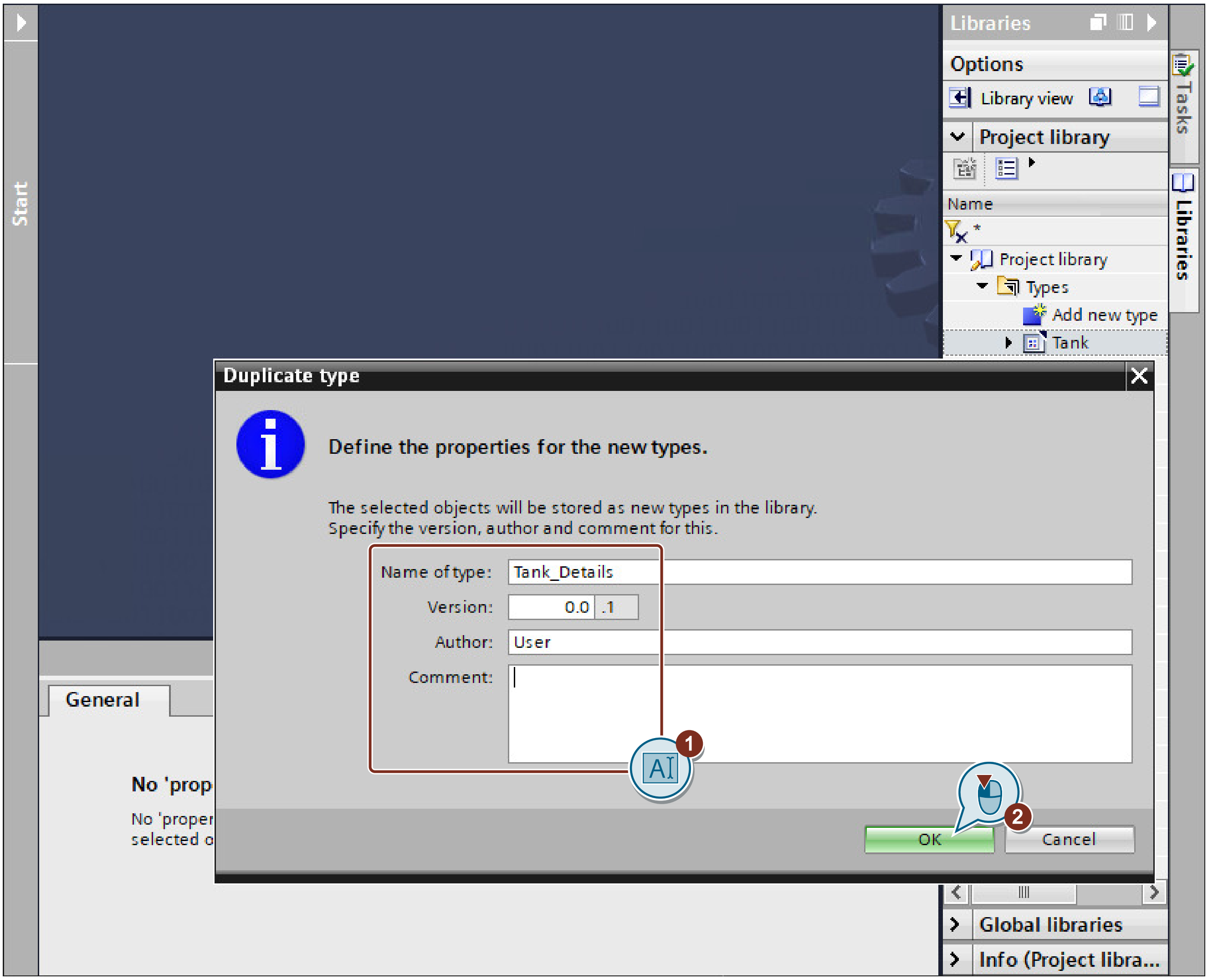
1. Duplicate the faceplate for the tank in the library. All interface tags of the "Tank" faceplate type are transferred to the new faceplate.



No 'properties' available.

No 'properties' can be shown at the moment. There is either no object selected or the selected object does not have any displayable properties.

2. Specify the properties of the new faceplate type.



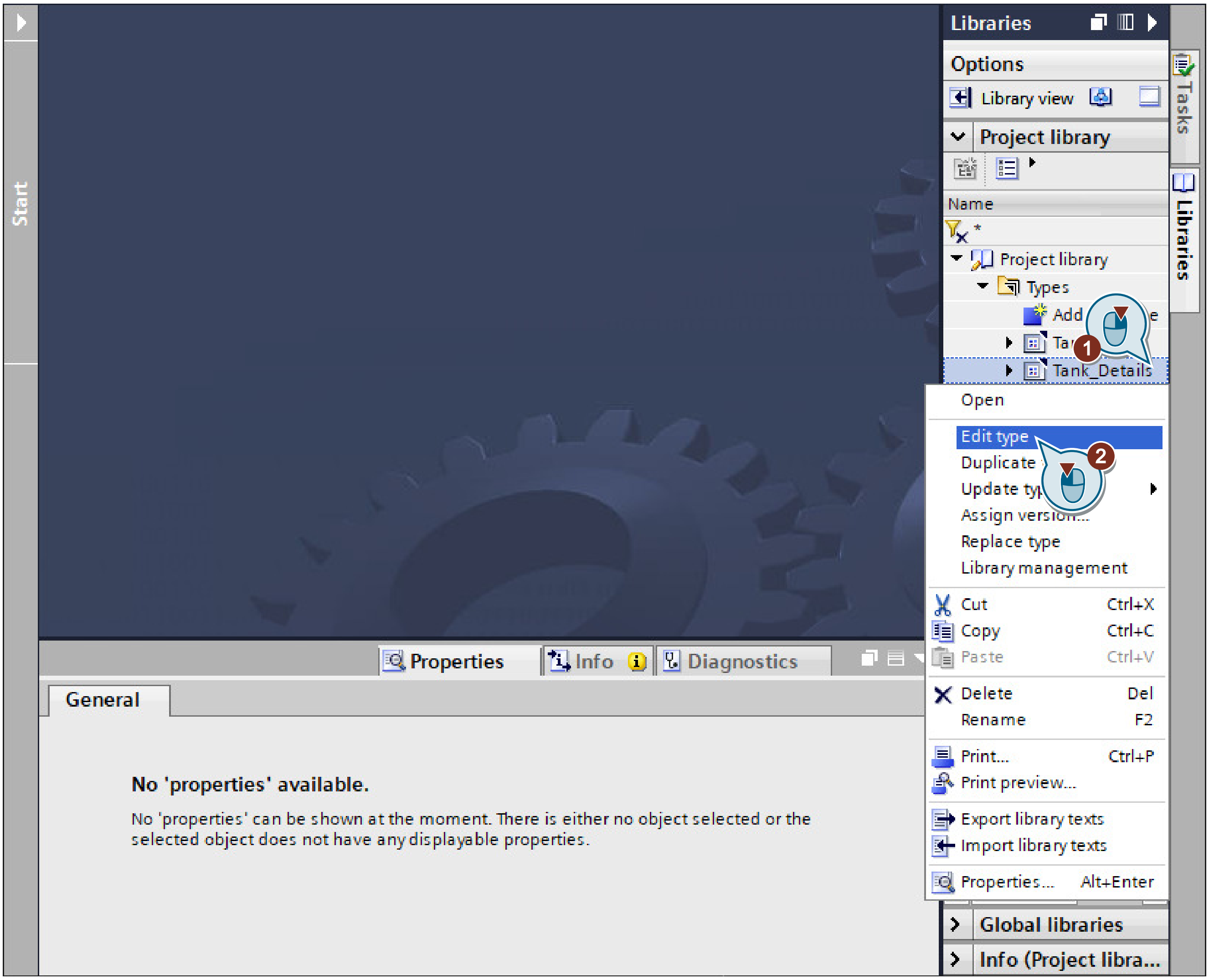
Duplicate type

Define the properties for the new types.

The selected objects will be stored as new types in the library.
Specify the version, author and comment for this.

Name of type: Tank_Details
Version: 0.0 .1
Author: User
Comment:

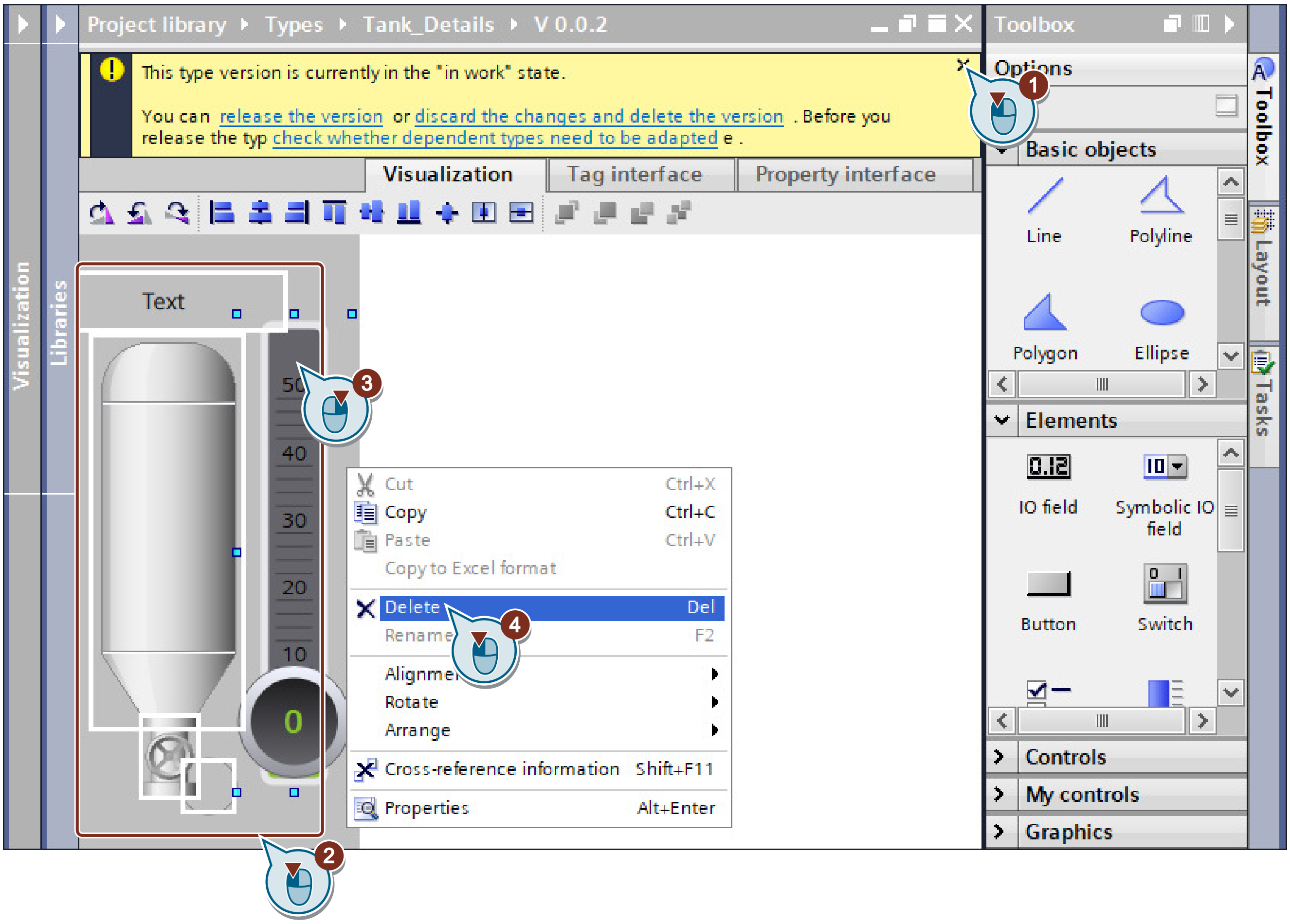
3. Open the new faceplate type for editing.



No 'properties' available.

No 'properties' can be shown at the moment. There is either no object selected or the selected object does not have any displayable properties.

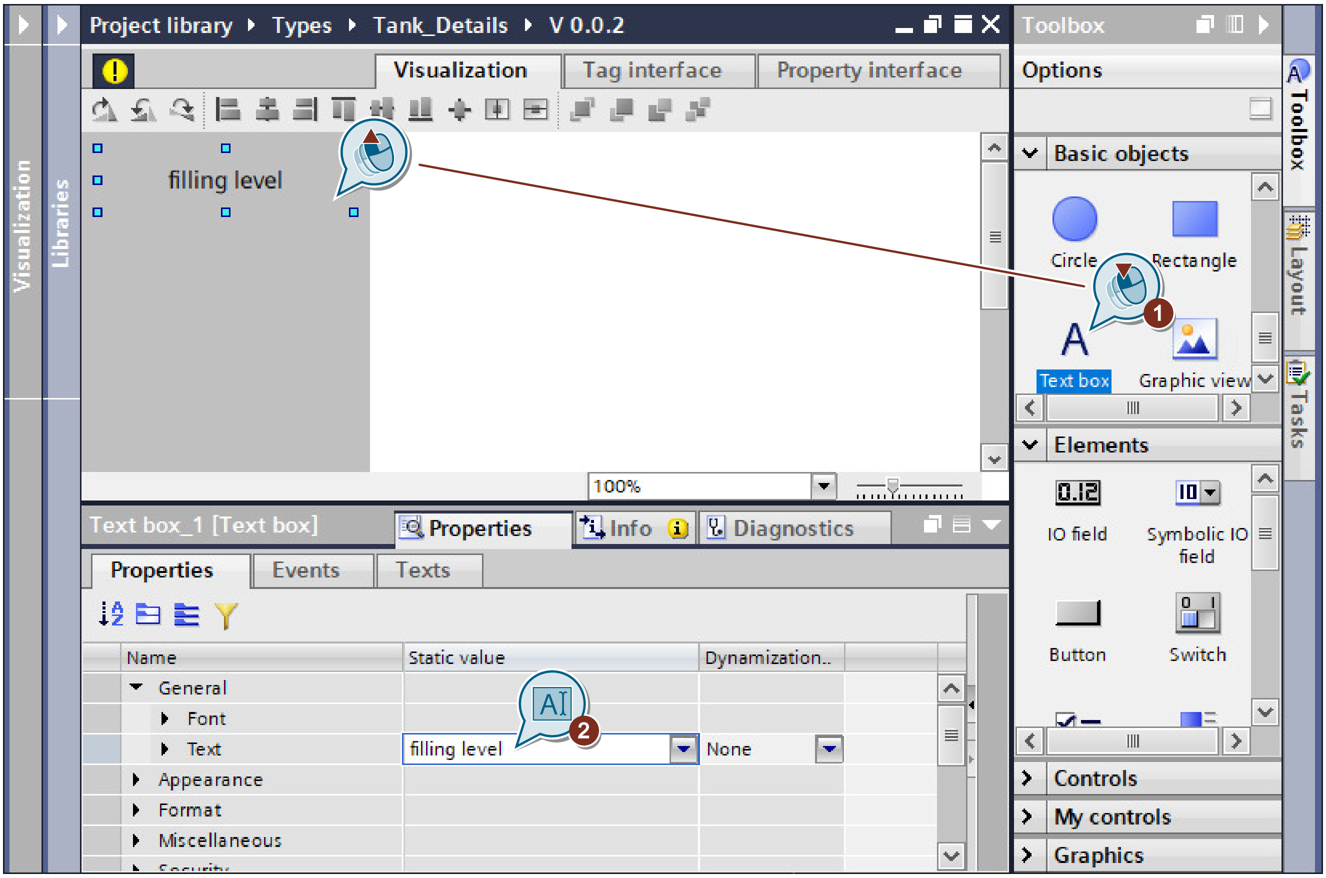
4. Delete the graphic elements of the copied faceplate.



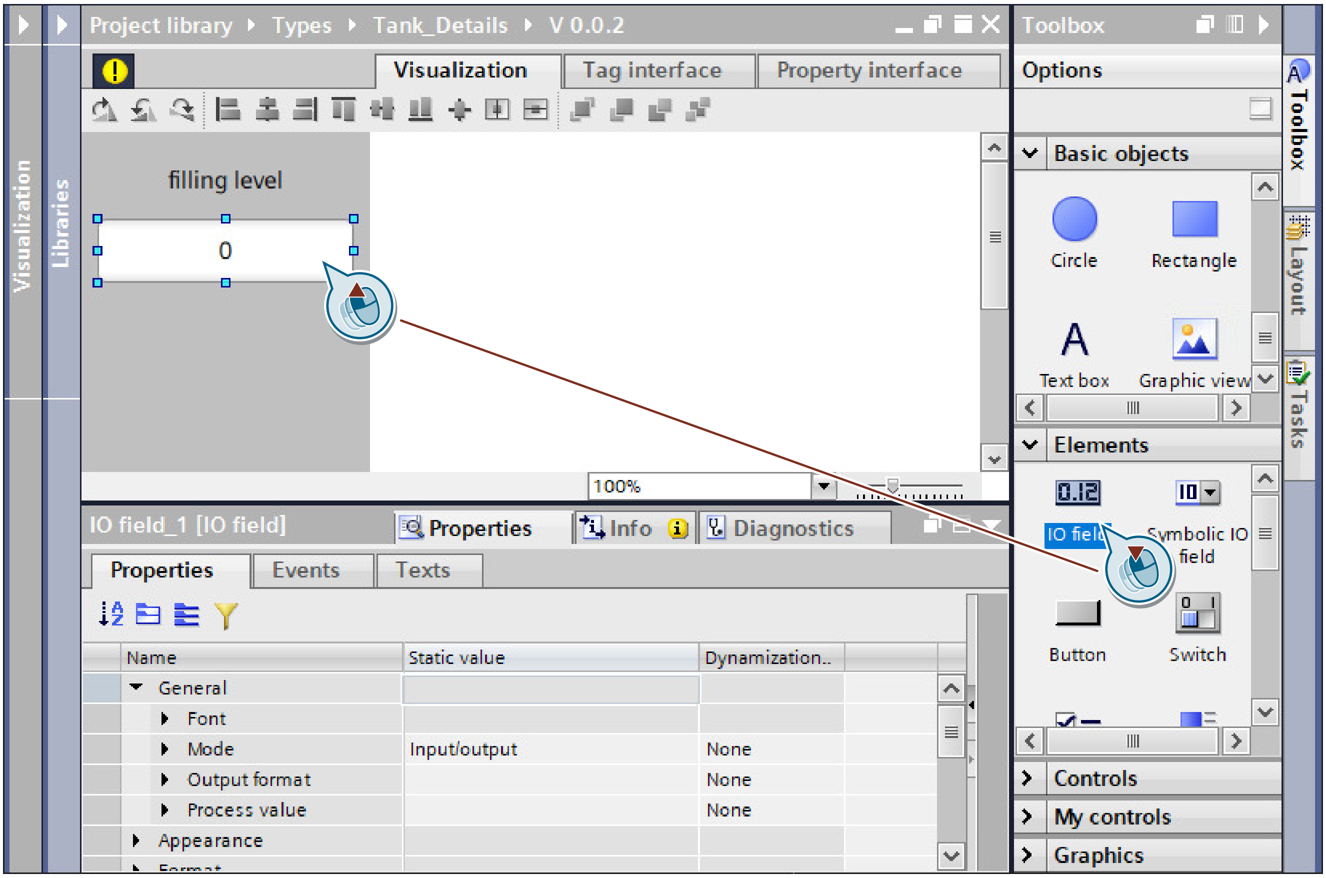
This type version is currently in the "in work" state.

You can release the version or discard the changes and delete the version . Before you release the typ check whether dependent types need to be adapted e .

5. Insert the text field for the fill level.

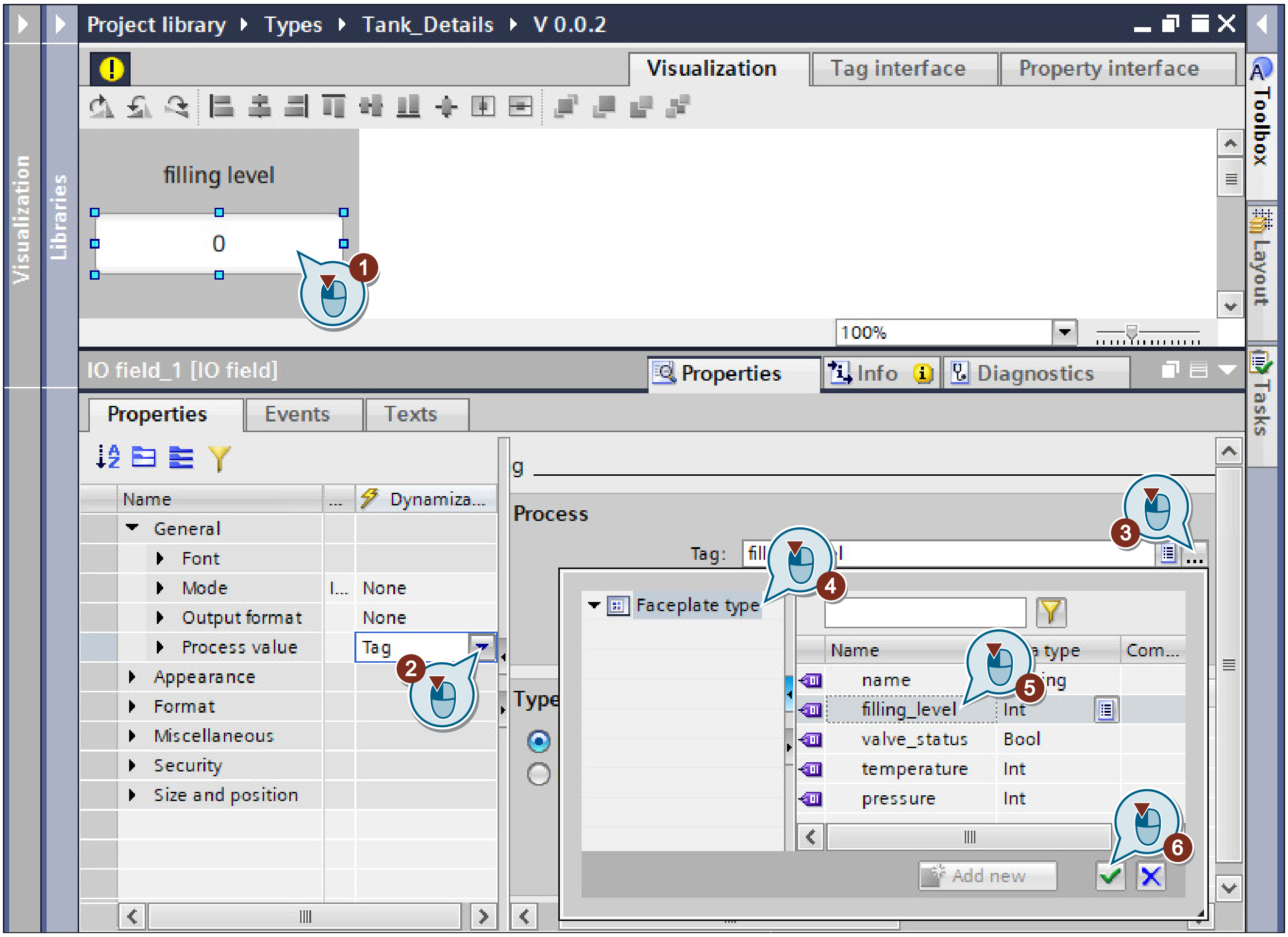


6. Insert the output field for the fill level.



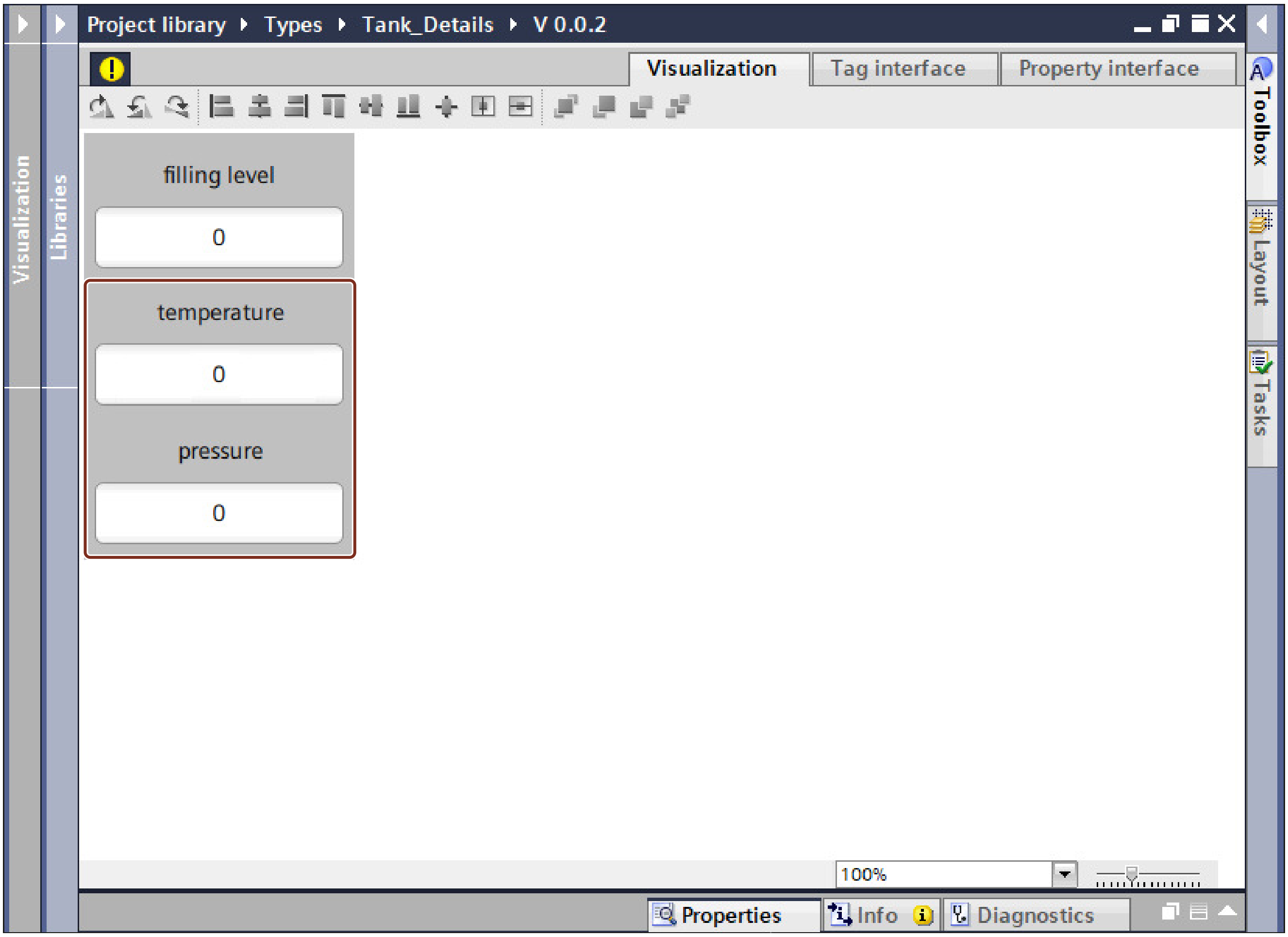
Name	Static value	Dynamization..
▾ General		
▸ Font		
▸ Mode	Input/output	None
▸ Output format		None
▸ Process value		None
▸ Appearance		
▸ Format		

7. Connect the output field with the interface tag "filling level".

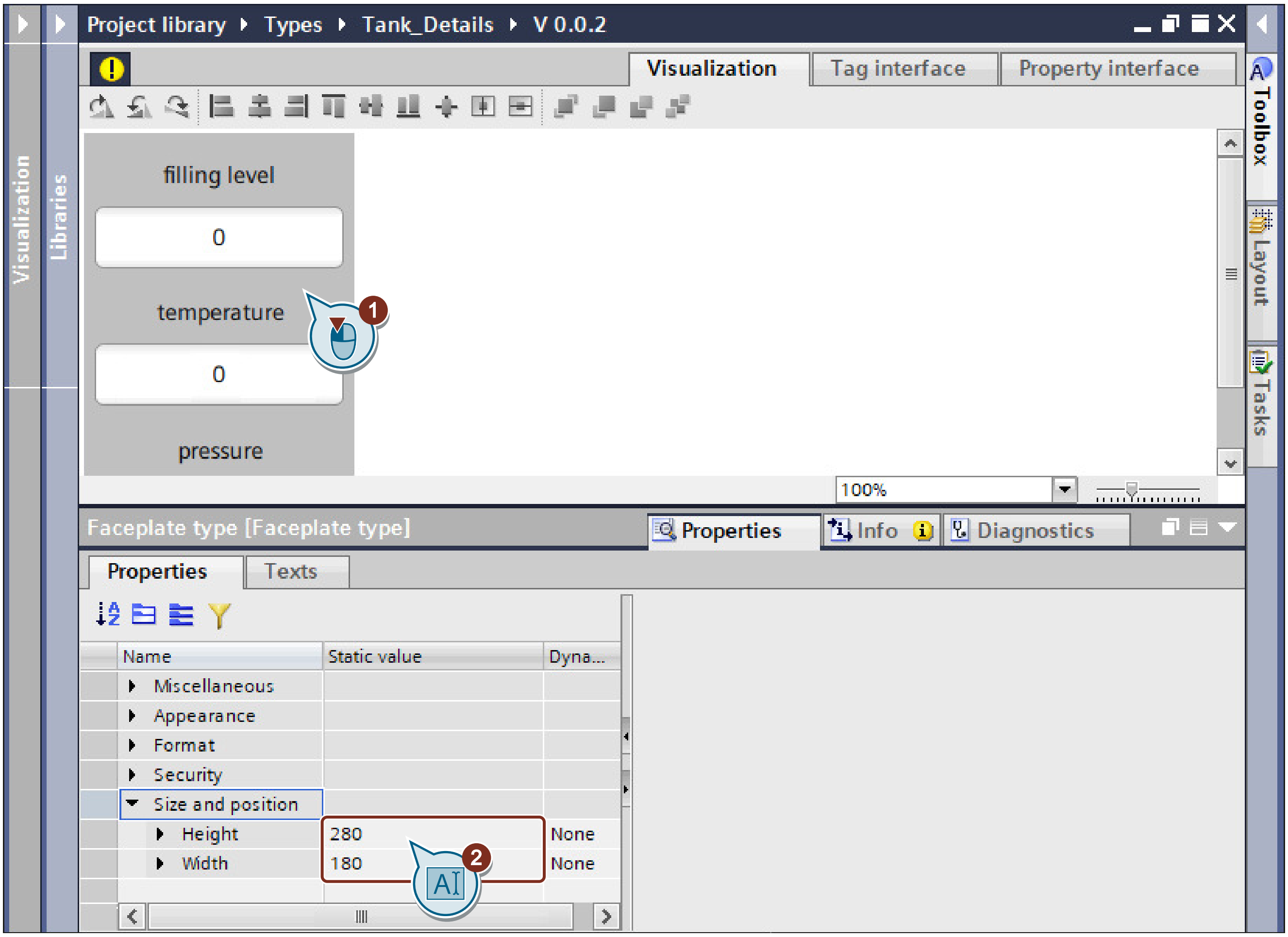


Name	Data type
name	...ing
filling_level	Int
valve_status	Bool
temperature	Int
pressure	Int

8. Configure the text fields and output fields for temperature and pressure and connect the output fields with the interface tags "temperature" and "pressure".

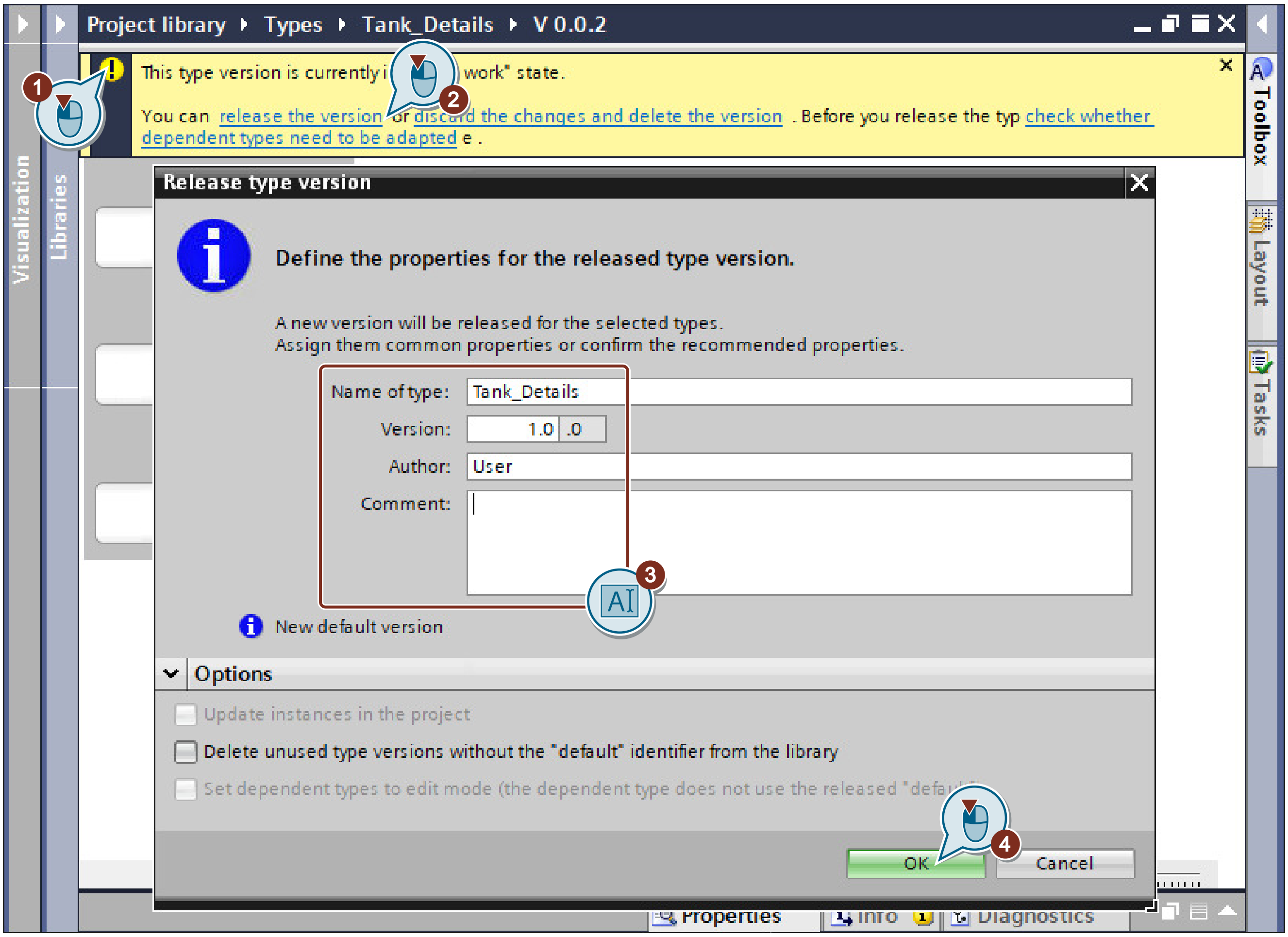


9. Adjust the size of the faceplate.



Name	Static value	Dyna...
Miscellaneous		
Appearance		
Format		
Security		
Size and position		
Height	280	None
Width	180	None

10. Share the faceplate pop-up in the library.



Project library ▸ Types ▸ Tank_Details ▸ V 0.0.2

This type version is currently i[...] work" state.

You can release the version or discard the changes and delete the version . Before you release the typ check whether dependent types need to be adapted e .

Release type version

Define the properties for the released type version.

A new version will be released for the selected types.
Assign them common properties or confirm the recommended properties.

Name of type:	Tank_Details
Version:	1.0 .0
Author:	User
Comment:	

New default version

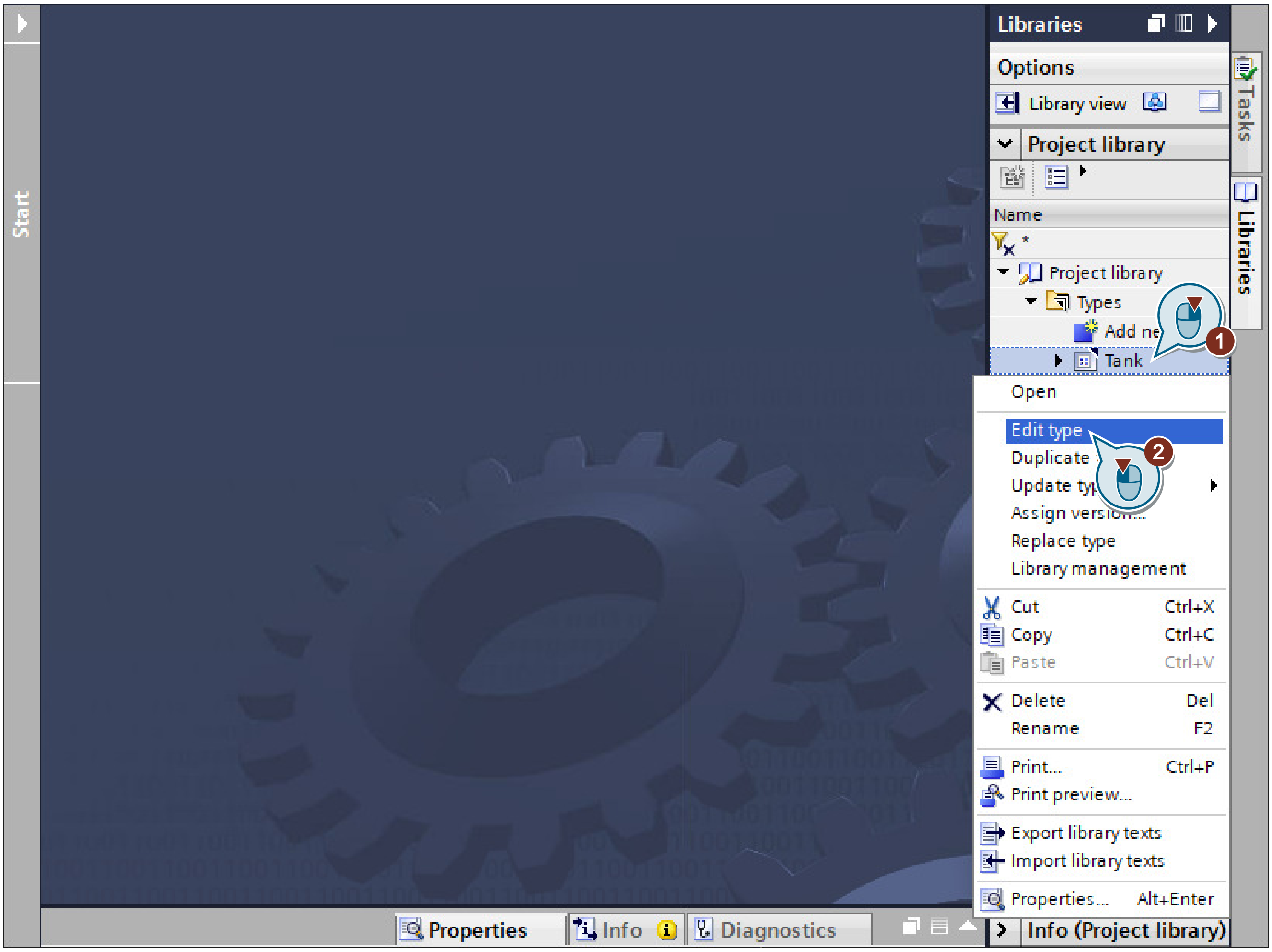
Options

□ Update instances in the project

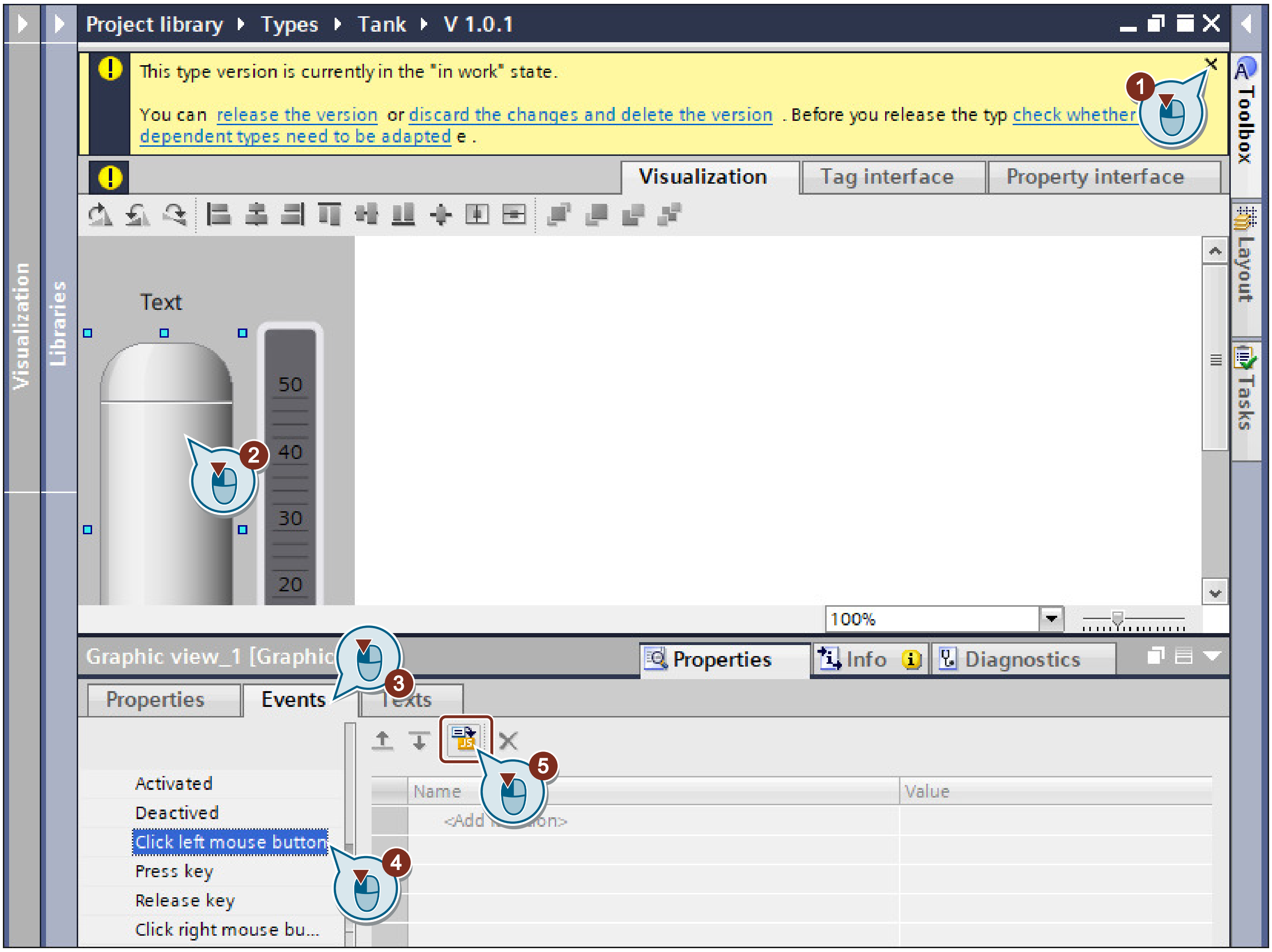
□ Delete unused type versions without the "default" identifier from the library

□ Set dependent types to edit mode (the dependent type does not use the released "defa[...]

11. Open the faceplate type "Tank" in the library for editing.



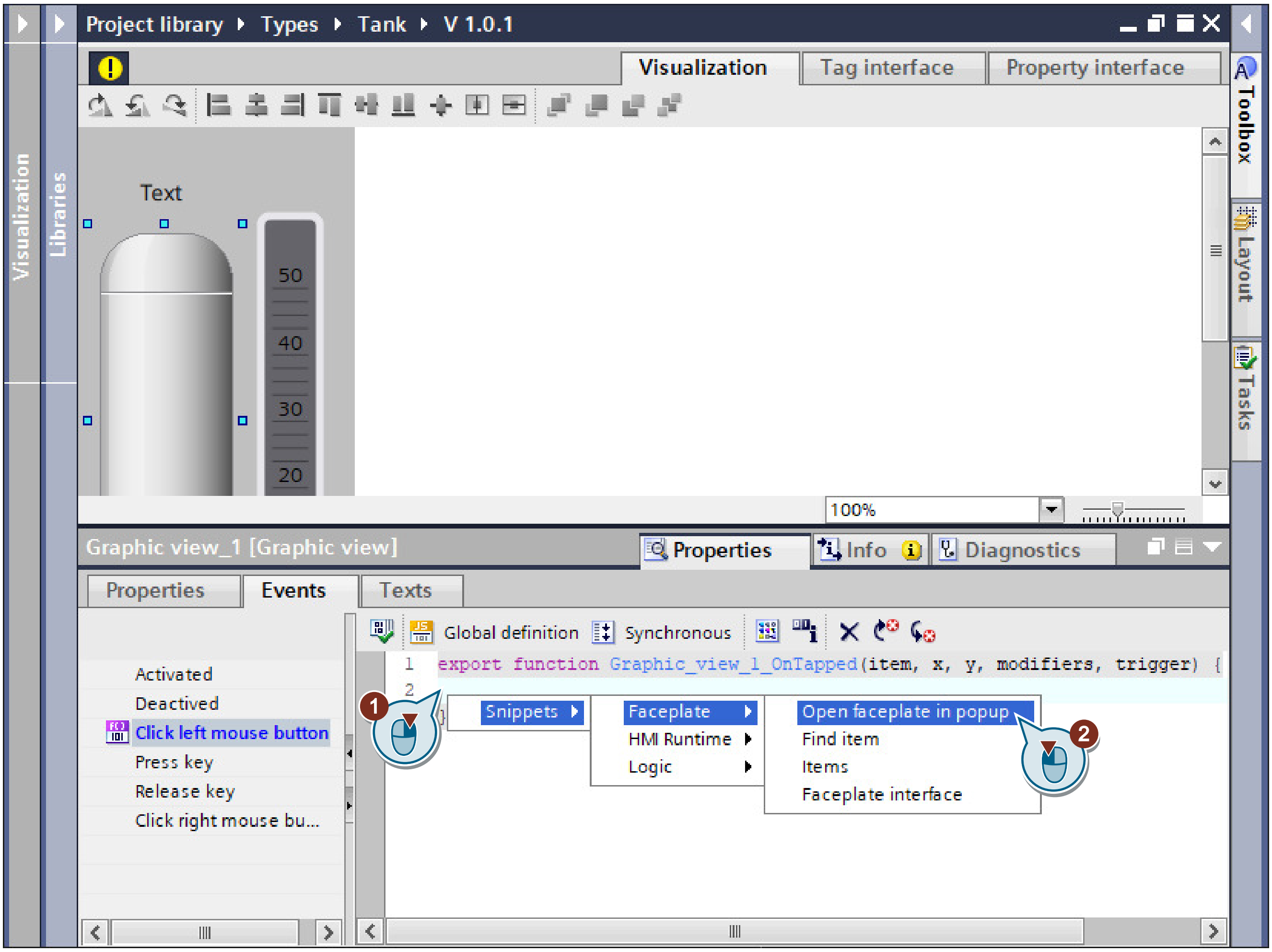
12. Create an event that is triggered by left-clicking on the tank.



This type version is currently in the "in work" state.

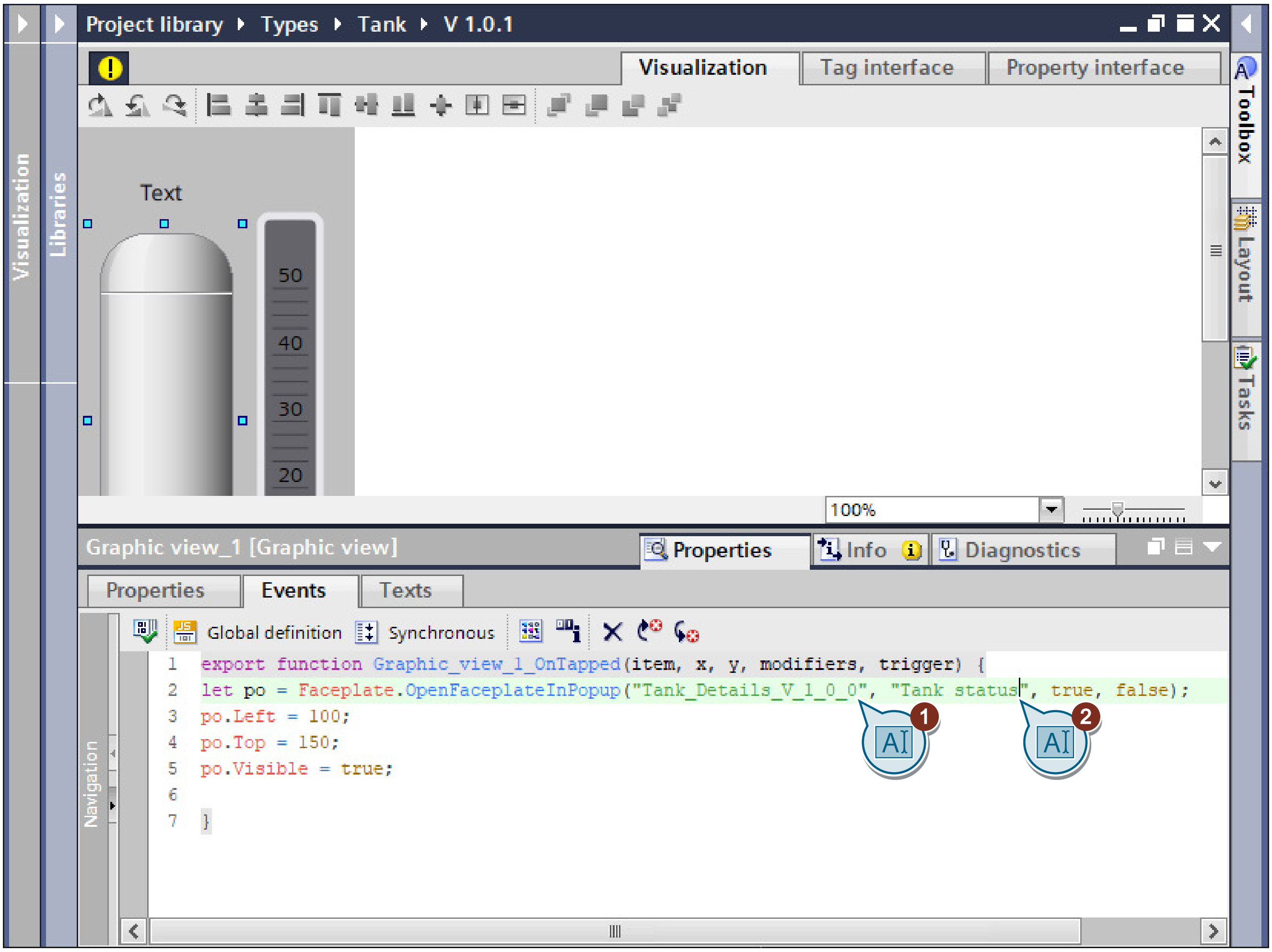
You can release the version or discard the changes and delete the version . Before you release the typ check whether dependent types need to be adapted e .

13. Insert the script template for opening a faceplate as a pop-up.



```
export function Graphic_view_1_OnTapped(item, x, y, modifiers, trigger) {
```

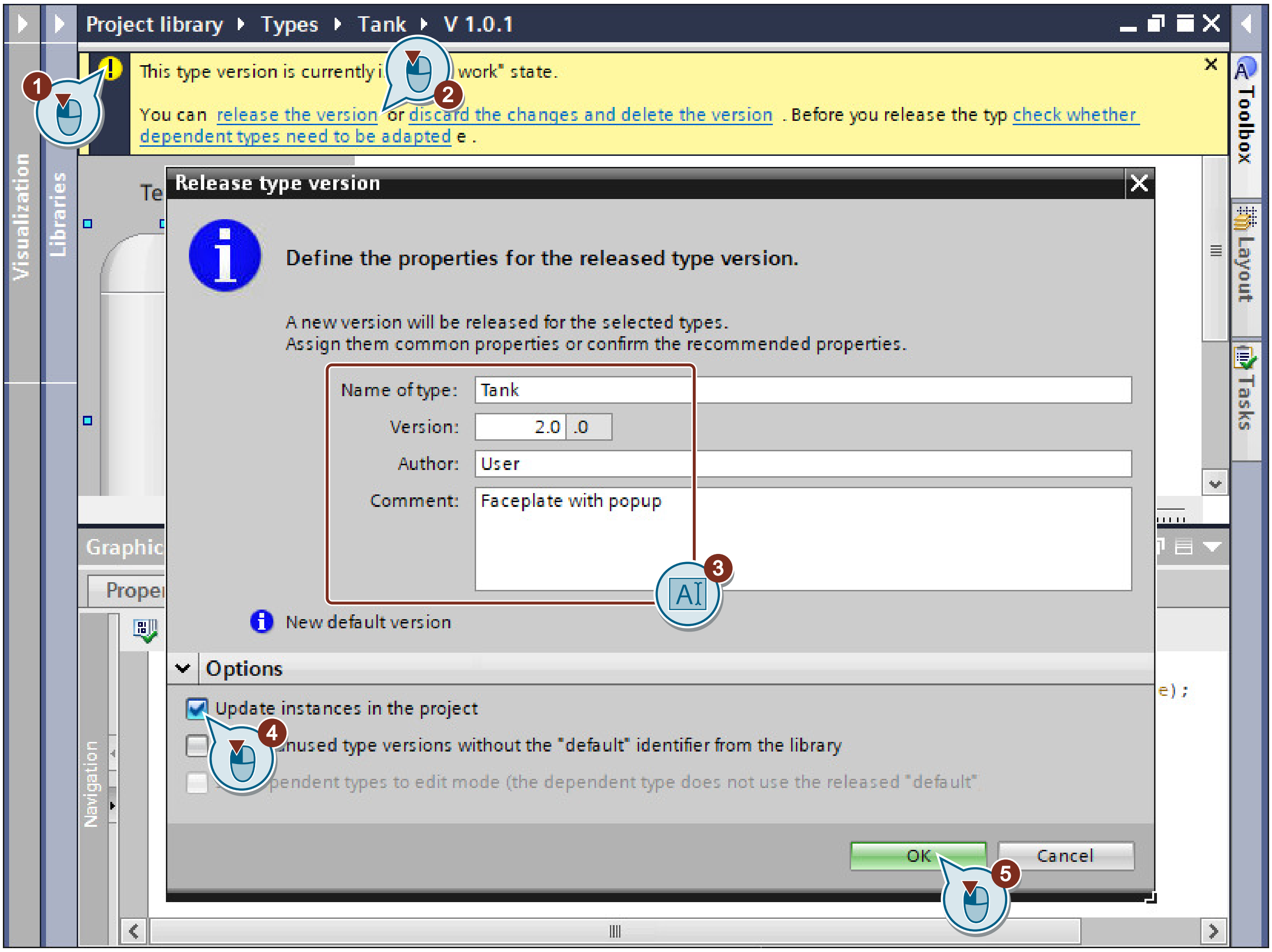
14. Add the name of the newly created faceplate and a designation to the script.



```
export function Graphic_view_1_OnTapped(item, x, y, modifiers, trigger) {
let po = Faceplate.OpenFaceplateInPopup("Tank_Details_V_1_0_0", "Tank status", true, false);
po.Left = 100;
po.Top = 150;
po.Visible = true;

}
```

15. Share the faceplate type in the library and specify that the instances are updated in the project.



This type version is currently i work" state.

You can release the version or discard the changes and delete the version . Before you release the typ check whether dependent types need to be adapted e .

Release type version

Define the properties for the released type version.

A new version will be released for the selected types.
Assign them common properties or confirm the recommended properties.

Name of type: Tank
Version: 2.0 .0
Author: User
Comment: Faceplate with popup

New default version

Options

☑ Update instances in the project
☐ unused type versions without the "default" identifier from the library
☐ pendent types to edit mode (the dependent type does not use the released "default"

Result

The pop-up window for faceplates has been created and is displayed in Runtime when you click one of the storage tanks with the configured information.

Monitoring and logging the brewing process 3

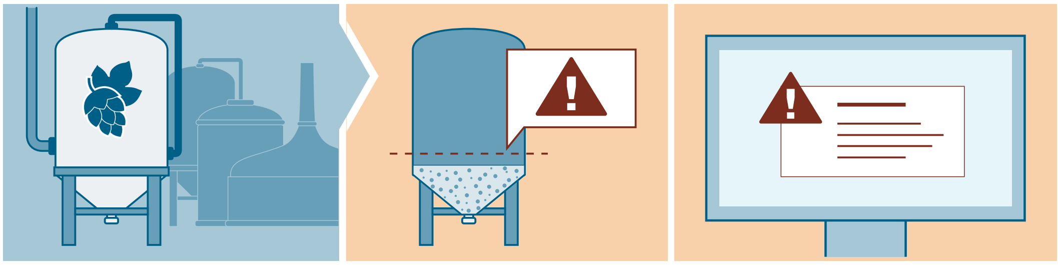
3.1 Configuring alarms

3.1.1 Alarms

Basics

WinCC ensures that all events in the course of the process are recorded chronologically. Deviations from the process are visualized as alarms to the system operator. WinCC supports complete traceability when processing the cause of an error.

An alarm contains the traceable status of the error processing with time stamp. An example of an alarm is when the level of a storage tank falls below the limit value.



3.1.2 Configuring alarms

Introduction

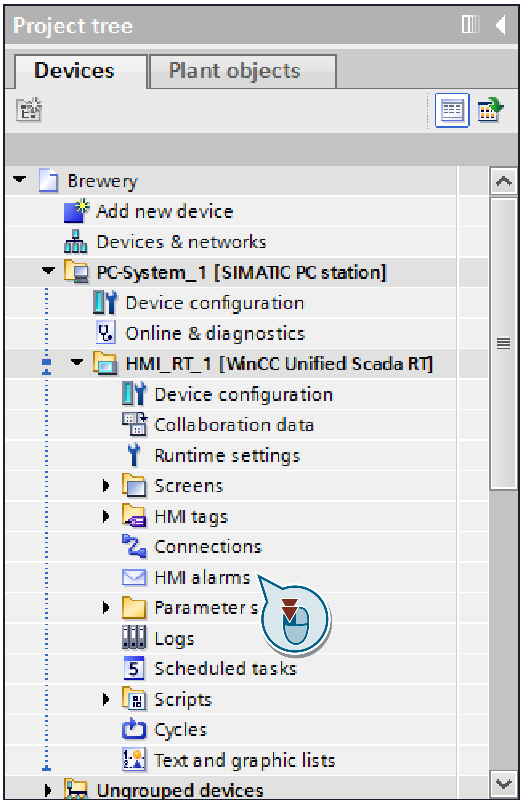
In the following, an alarm is to be configured that is triggered when the fill level in one of the storage tanks falls below a defined limit value.

The following alarms are to be configured for the storage tanks of the ingredients.

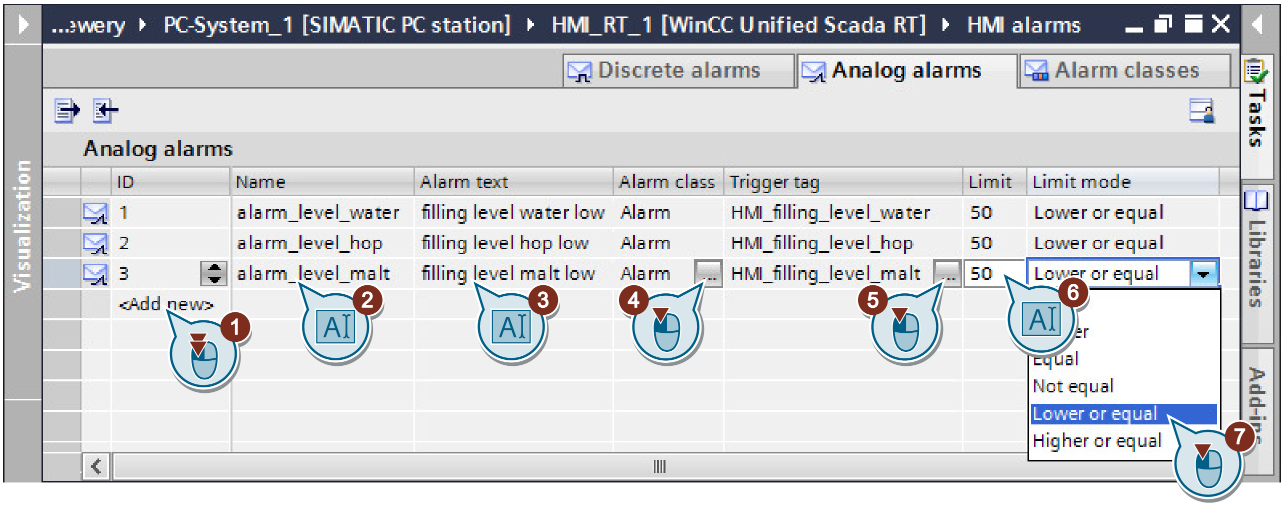
Name	Alarm text	Supervised tag	Alarm class	Limit
alarm_level_water	"filling level water low"	HMI_filling_level_water	Alarm	50 l
alarm_level_malt	"filling level malt low"	HMI_filling_level_malt	Alarm	50 l
alarm_level_hop	"filling level hop low"	HMI_filling_level_hop	Alarm	50 l

Procedure

1. Open the alarm overview.



2. Create alarms for limit violations of the fill level. Create the remaining alarms in the same way.



Analog alarms

ID	Name	Alarm text	Alarm class	Trigger tag	Limit	Limit mode
1	alarm_level_water	filling level water low	Alarm	HMI_filling_level_water	50	Lower or equal
2	alarm_level_hop	filling level hop low	Alarm	HMI_filling_level_hop	50	Lower or equal
3	alarm_level_malt	filling level malt low	Alarm	HMI_filling_level_malt	50	Lower or equal
<Add new>						

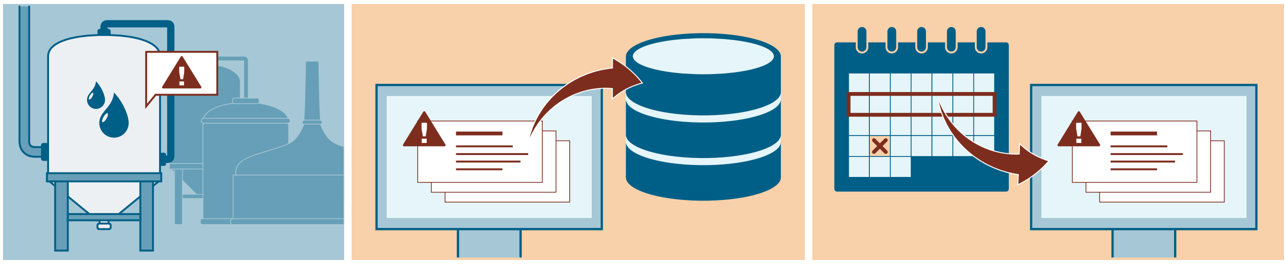
3.2 Logging alarms and process values

3.2.1 Logging

Basics

Logging is a method for analyzing error states and for process documentation. By evaluating the logged data, you can derive measures to optimize maintenance cycles or increase product quality.

In WinCC, during configuration, you select alarm classes and tags that are to be logged in the process mode. For later evaluation, select the time period from which you want to display alarms, for example.



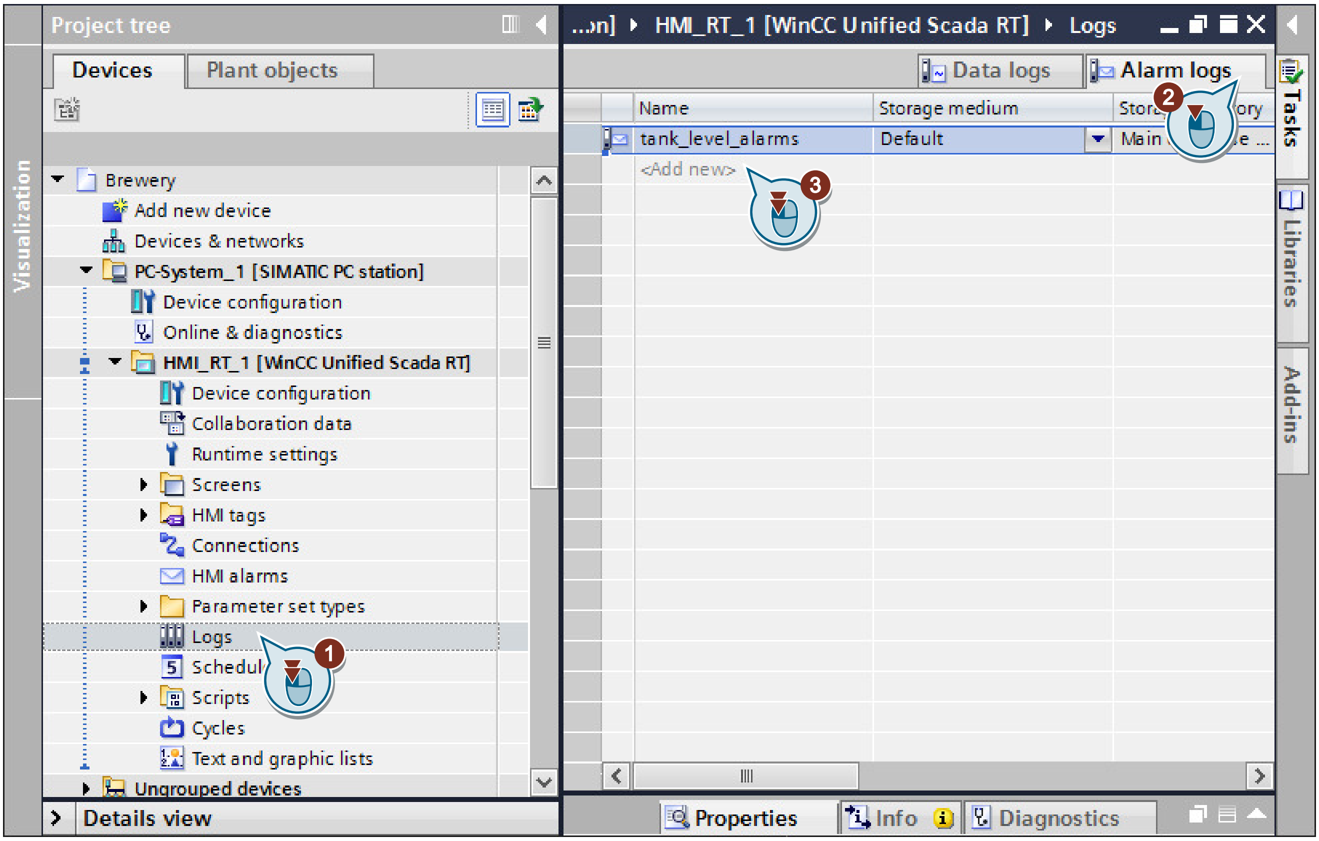
3.2.2 Logging alarms

Introduction

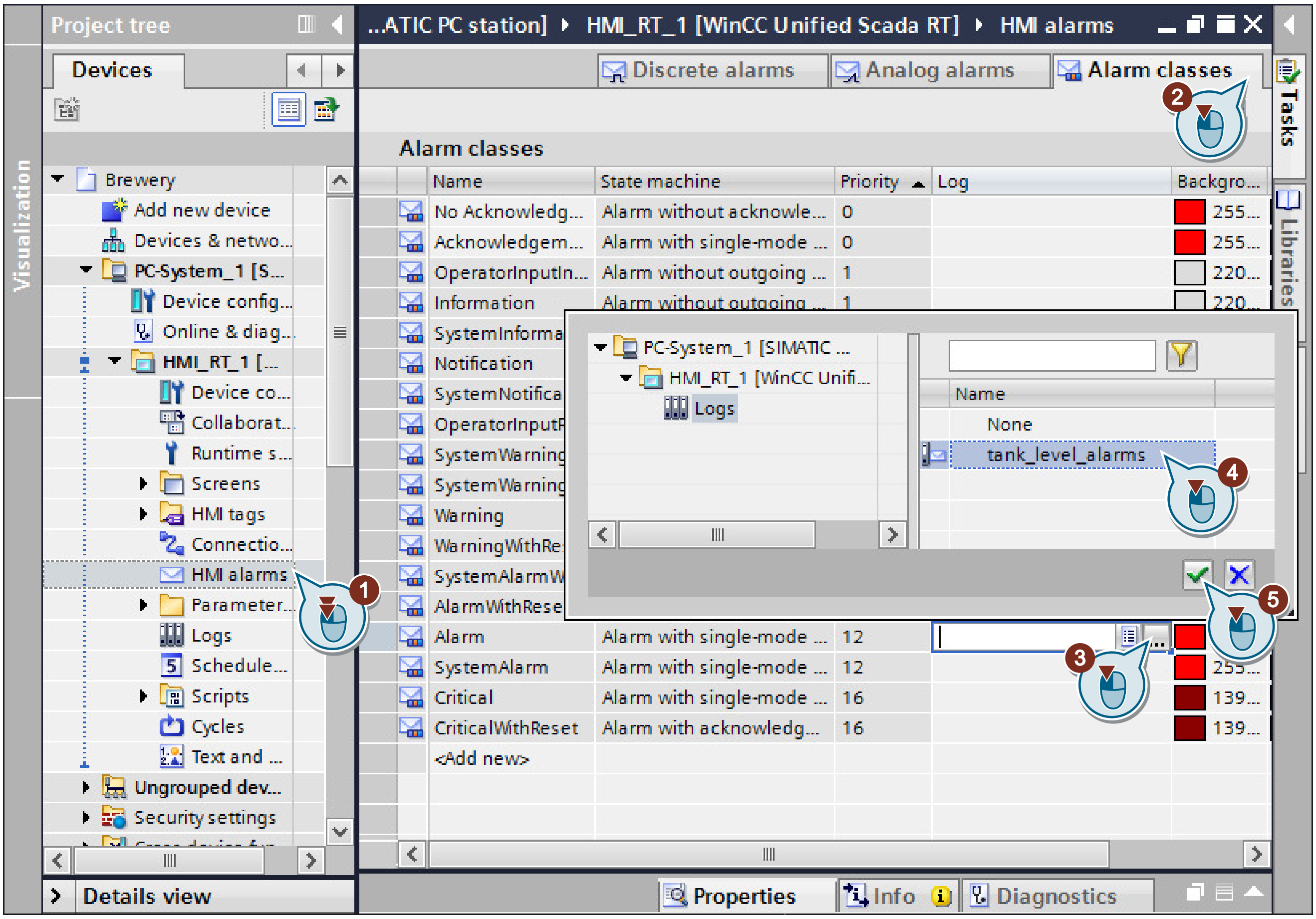
The previously configured level alarms with the alarm class "Alarm" should now be logged. An alarm log is created for the logging and then stored in the "Alarm" alarm class. All alarms of the alarm class are logged in this alarm log.

Procedure

1. Create a new alarm log in which the alarms for limit value violations are to be logged.



2. Store the alarm log with the alarm class.



Alarm classes

	Name	State machine	Priority	Log	Backgro...
	No Acknowledg...	Alarm without acknowle...	0		🟥 255...
	Acknowledgem...	Alarm with single-mode ...	0		🟥 255...
	OperatorInputIn...	Alarm without outgoing ...	1		220...
	Information	Alarm without outgoing ...	1		220...
	SystemInforma				
	Notification				
	SystemNotifica				
	OperatorInputF				
	SystemWarning				
	SystemWarning				
	Warning				
	WarningWithRe				
	SystemAlarmW				
	AlarmWithRese				
	Alarm	Alarm with single-mode ...	12		🟥
	SystemAlarm	Alarm with single-mode ...	12		🟥 255...
	Critical	Alarm with single-mode ...	16		🟥 139...
	CriticalWithReset	Alarm with acknowledg...	16		🟥 139...
	<Add new>				

PC-System_1 [SIMATIC ...
HMI_RT_1 [WinCC Unifi...
Logs

Name
None
tank_level_alarms

3.2.3 Logging process values

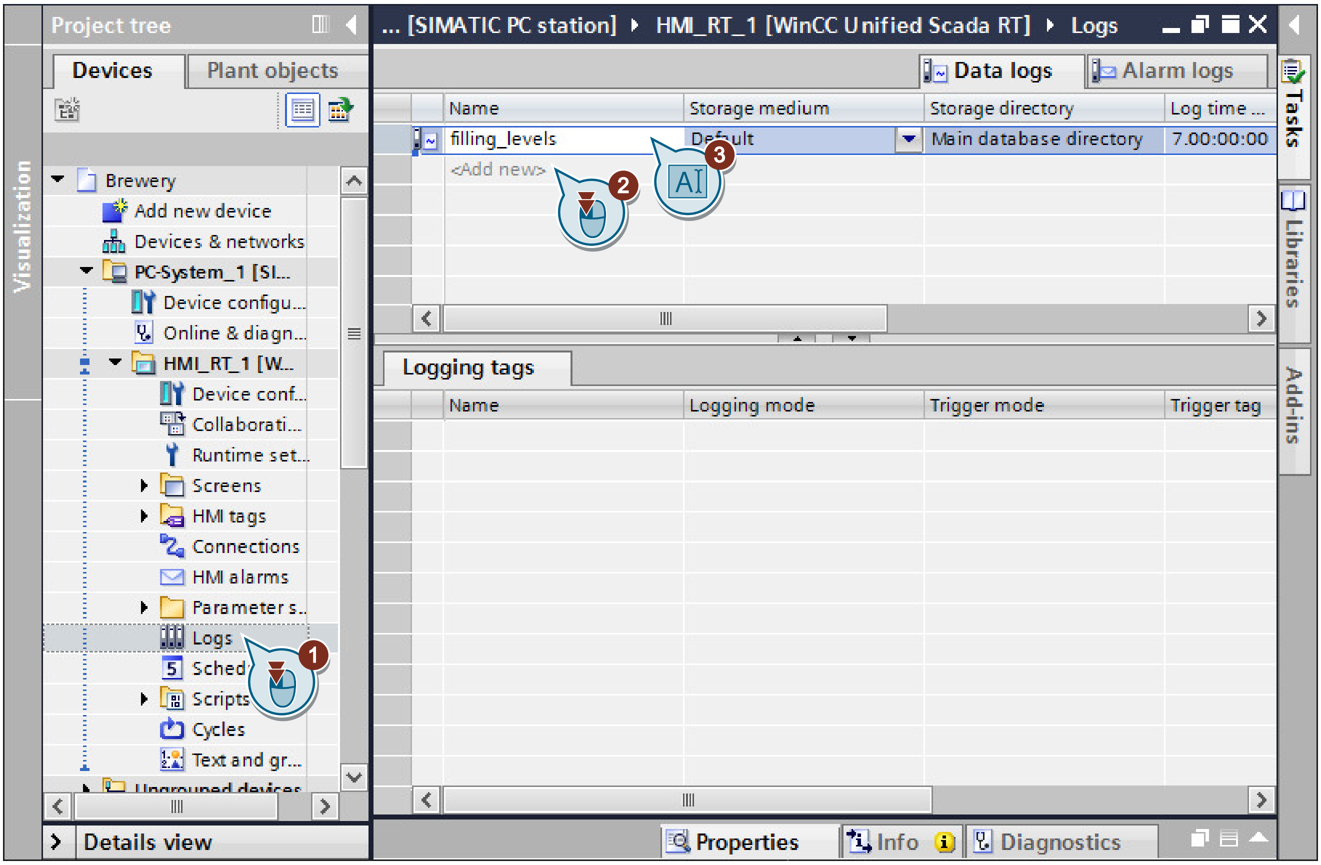
Introduction

The process values of the fill levels in the storage tanks and in the mash tun should be logged. To do this, a logging tag must be created for each tag and then assigned to a data log.

Logging tag name	Logging of process values from tag	Data type	Data log
log_water_level	HMI_filling_level_water	Int	filling_levels
log_hop_level	HMI_filling_level_hop	Int	filling_levels
log_malt_level	HMI_filling_level_malt	Int	filling_levels
log_mash_tun_level	HMI_filling_level_mash_tun	Int	filling_levels
log_status_water	HMI_valve_status_water	Bool	valve_status
log_status_hop	HMI_valve_status_hop	Bool	valve_status
log_status_malt	HMI_valve_status_malt	Bool	valve_status

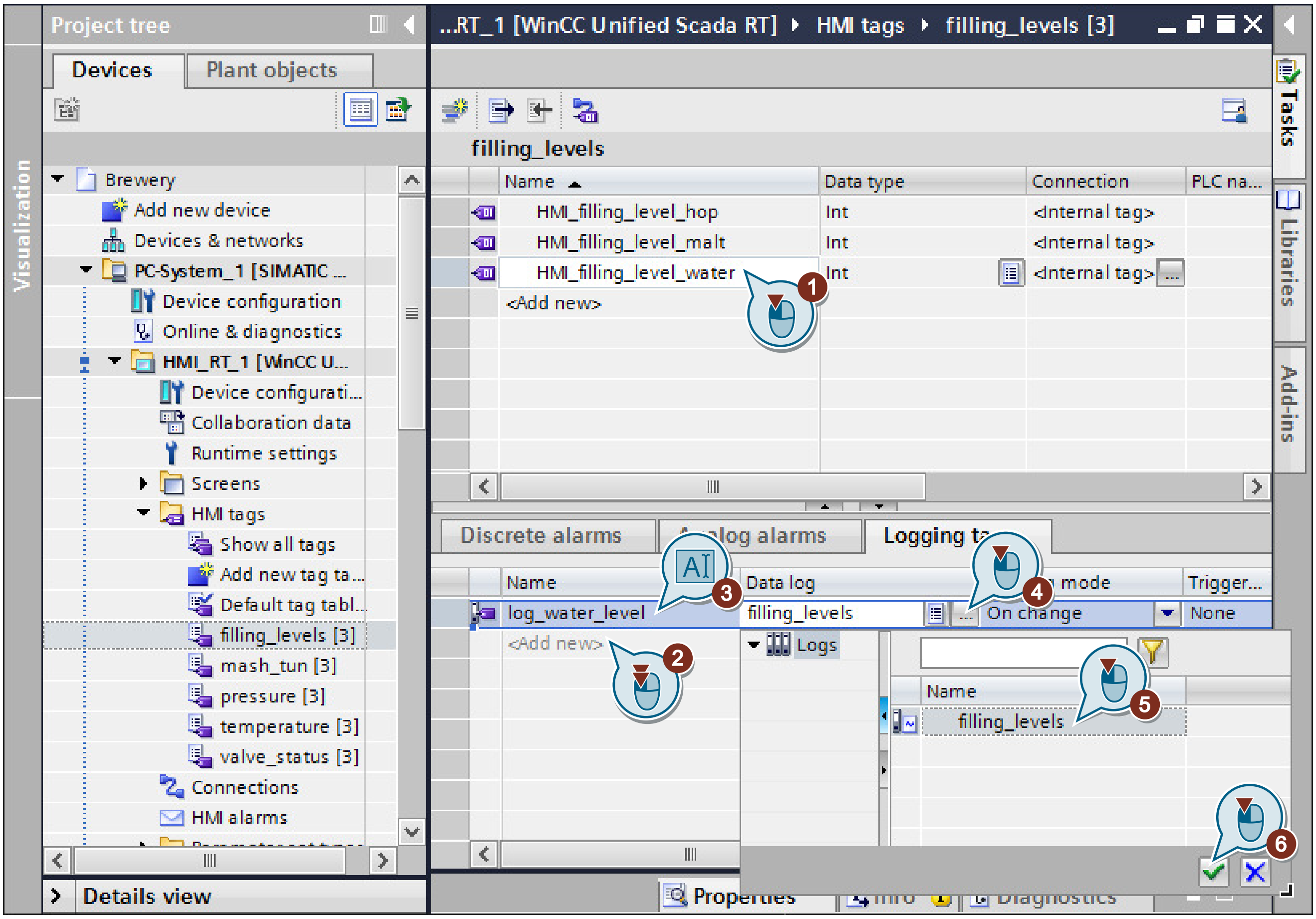
Procedure

1. Create a data log with the name "filling_levels".



Name	Storage medium	Storage directory	Log time ...
filling_levels	Default	Main database directory	7.00:00:00
<Add new>			

2. Open the tag table "filling_levels" and create a logging tag for each of the tags mentioned above. Assign the logging tags to the data log "filling_levels".



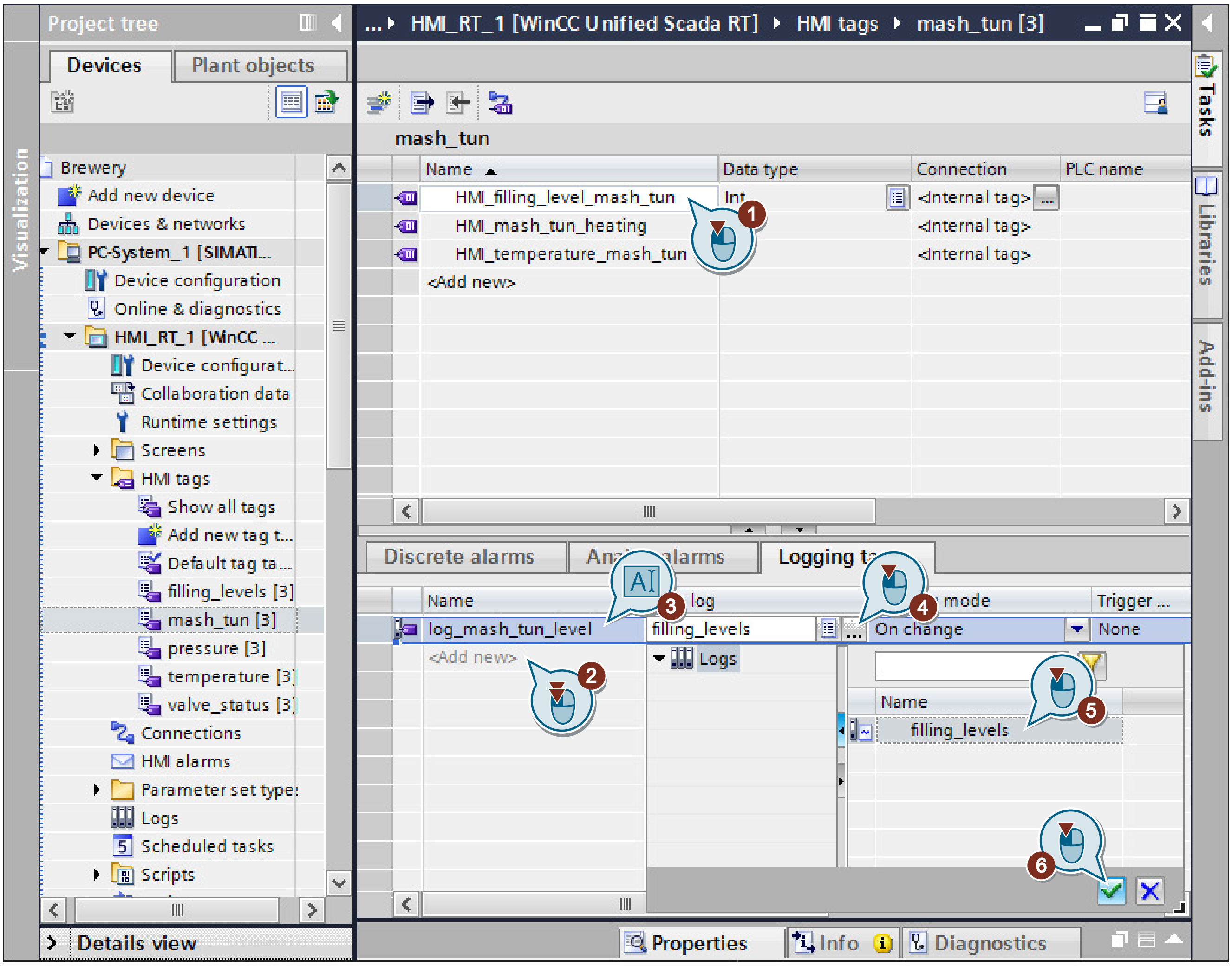
filling_levels

Name	Data type	Connection	PLC na...
HMI_filling_level_hop	Int	<Internal tag>	
HMI_filling_level_malt	Int	<Internal tag>	
HMI_filling_level_water	Int	<Internal tag>	
<Add new>			

Name	Data log	Logging mode	Trigger...
log_water_level	filling_levels	On change	None
<Add new>			

Name
filling_levels

3. Open the tag table "mash_tun" and create a logging tag for the tag mentioned above. Assign the logging tags to the data log "filling_levels".

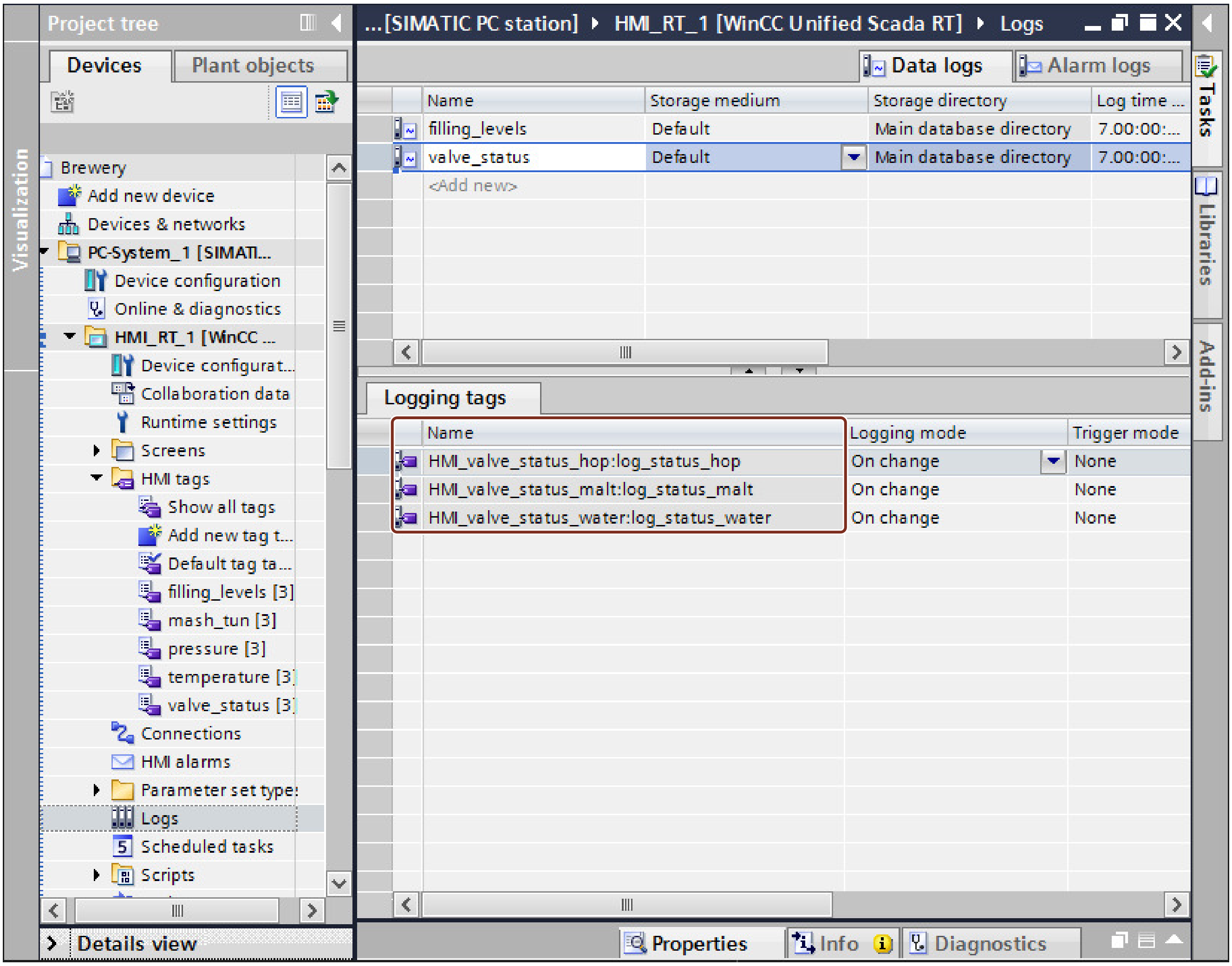


mash_tun

Name	Data type	Connection	PLC name
HMI_filling_level_mash_tun	Int	<Internal tag>	
HMI_mash_tun_heating		<Internal tag>	
HMI_temperature_mash_tun		<Internal tag>	
<Add new>			

Name	log	mode	Trigger ...
log_mash_tun_level	filling_levels	On change	None
<Add new>			

4. Create a "valve_status" data log in the same way.
5. Open the tag table "valve_status" and create a logging tag in the same way for each of the tags mentioned above. Assign the logging tags to the data log "valve_status".



Name	Storage medium	Storage directory	Log time ...
filling_levels	Default	Main database directory	7.00:00:...
valve_status	Default	Main database directory	7.00:00:...
<Add new>			

Name	Logging mode	Trigger mode
HMI_valve_status_hop:log_status_hop	On change	None
HMI_valve_status_malt:log_status_malt	On change	None
HMI_valve_status_water:log_status_water	On change	None

3.3 Visualizing alarms and process values

3.3.1 Visualizing alarms and process values

Visualization of alarms

Recorded alarms can be shown in an alarm control in Runtime. In this way, you can see immediately whether there are problems or errors in the system. For example, it is indicated if the fill level of a storage tank falls below a limit value and has to be refilled.

Visualization of process values

Current process values can be shown in a trend control in Runtime. For example, you can display the fill level of the three storage tanks in a separate trend for each. As a result, you can always see the current fill level clearly and know when new supplies have to be ordered and refilled. You can also display the valve status in a trend control. In this way, you can see which valve is open and when.

3.3.2 Configuring an alarm control

Introduction

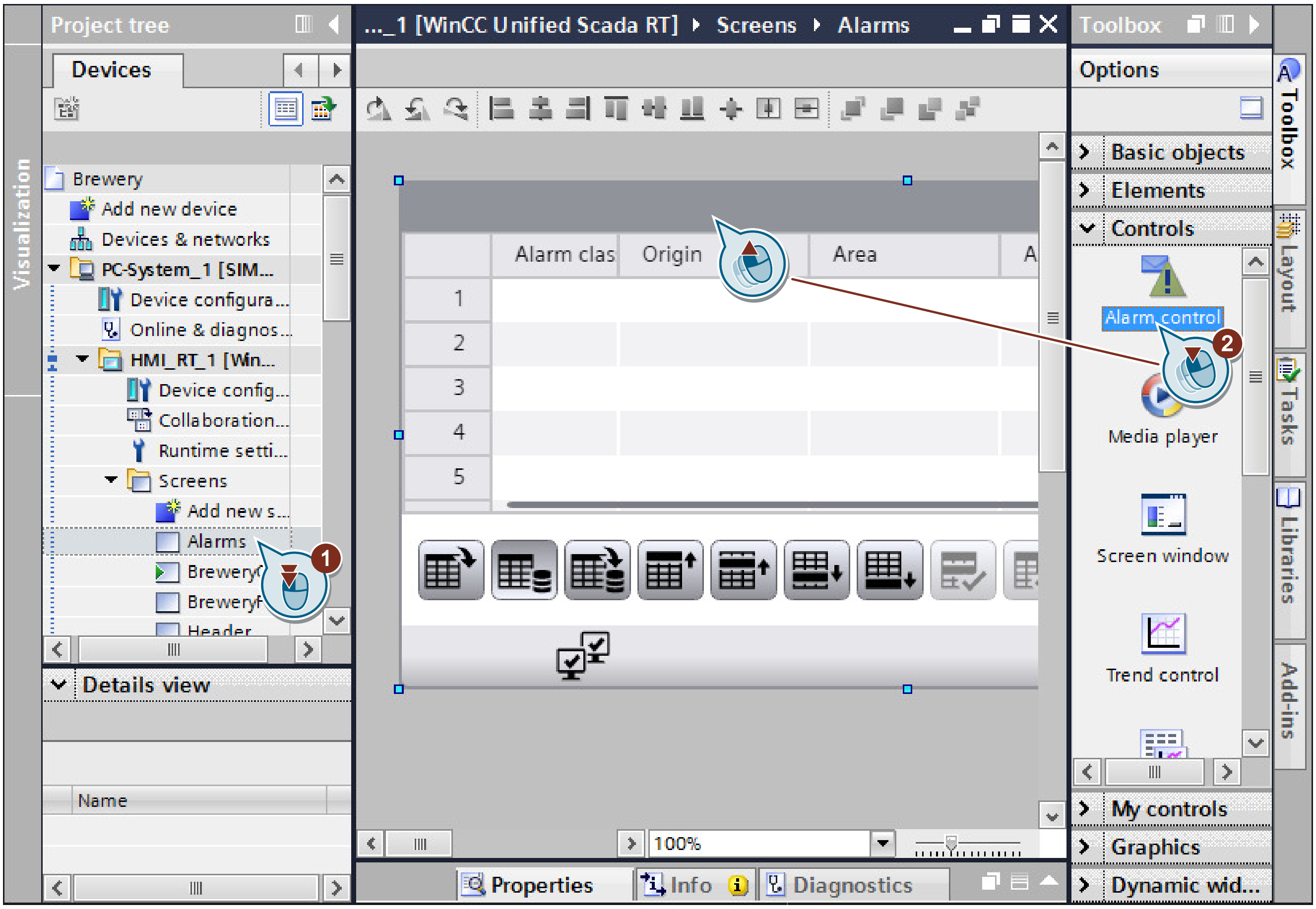
In the following, you will learn how to configure an alarm control to display the logged alarms.

Requirement

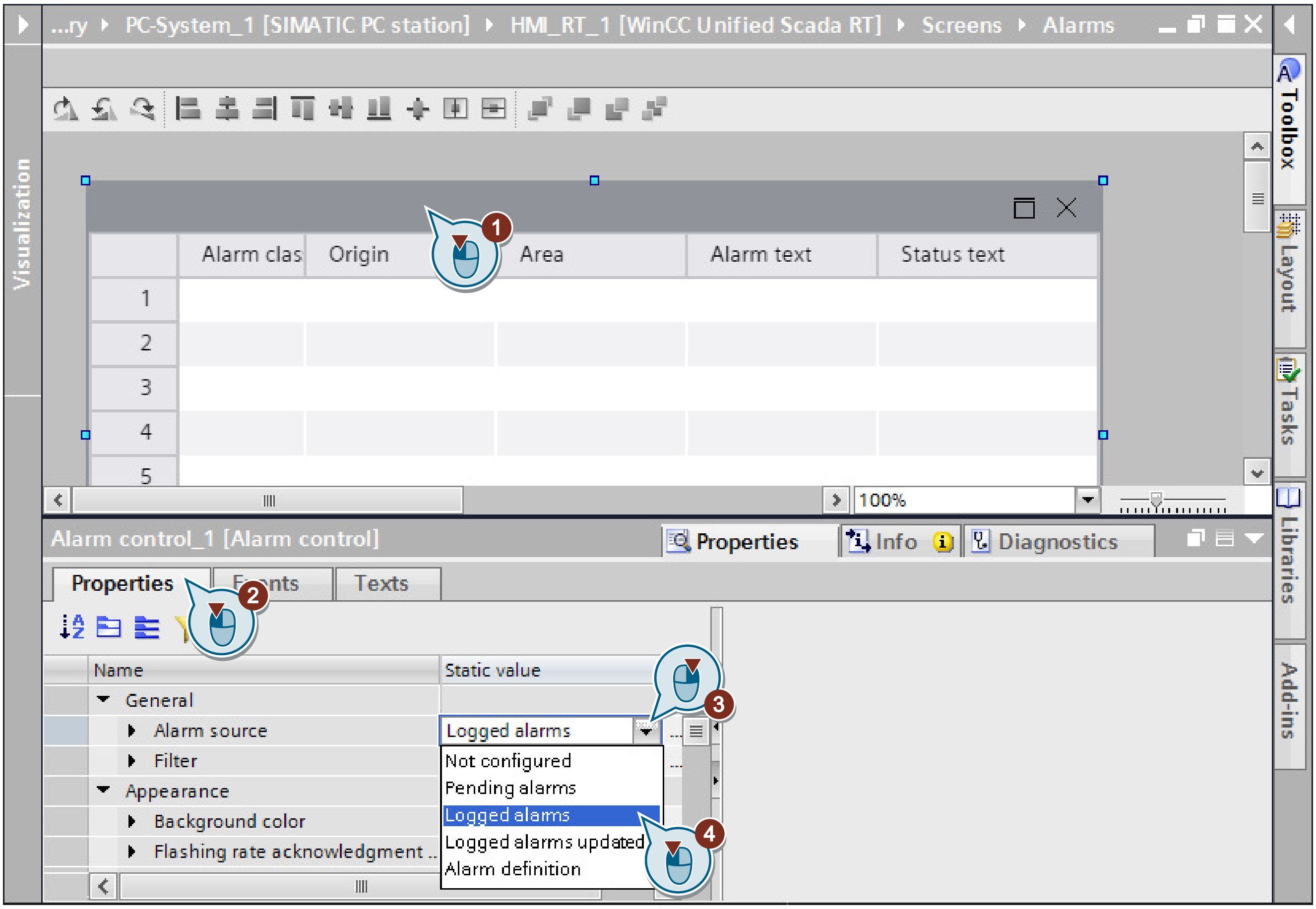
The alarm log is created.

Procedure

1. Open the "Alarms" screen and insert an alarm control.



2. Configure the alarm control so that the logged alarms are displayed:



Result

The alarm control to display the logged alarms is configured.

3.3.3 Visualizing process values as trends

Introduction

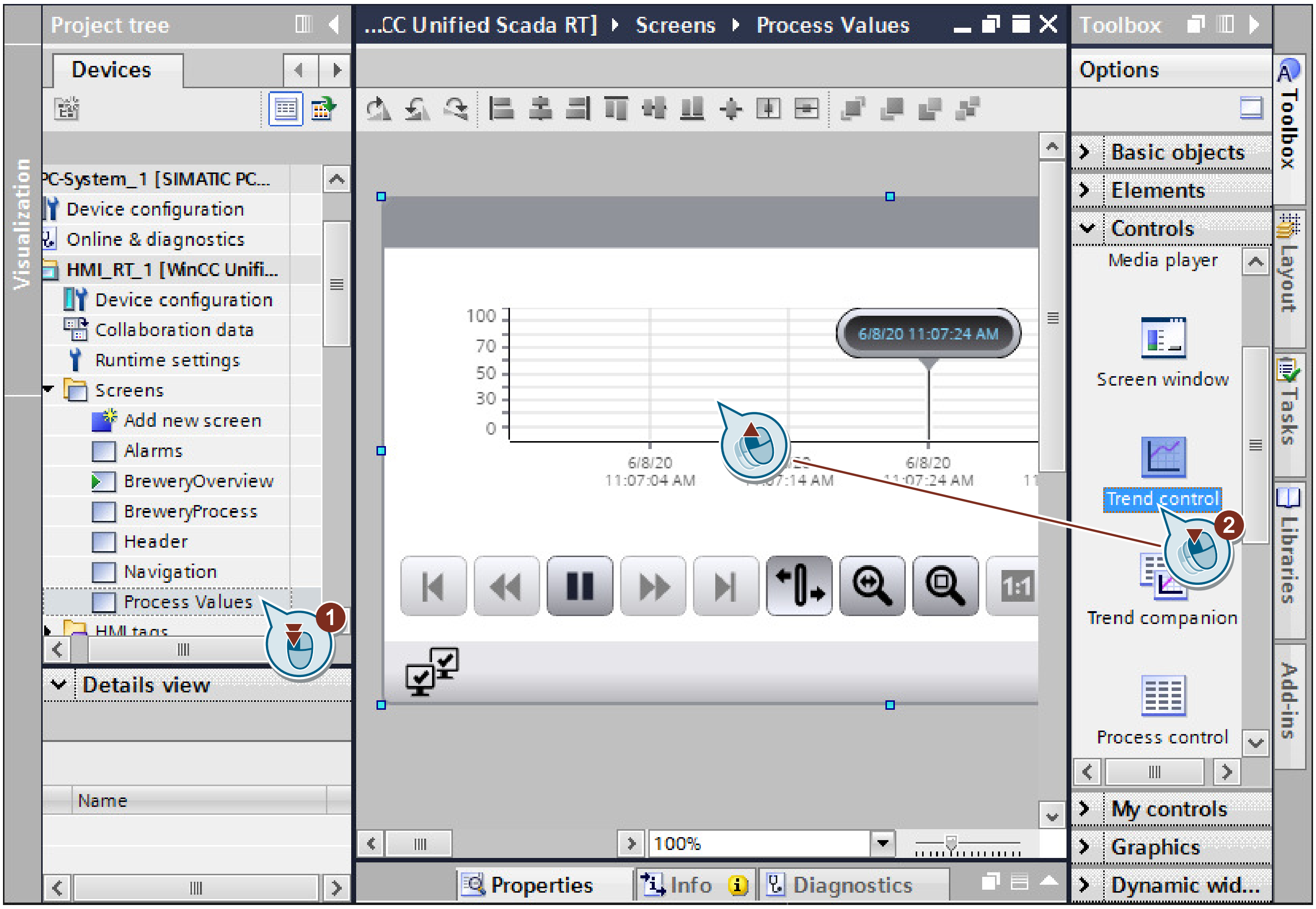
The fill levels of the three tanks and the status of the three valves should be displayed as a trend on the HMI device.

Requirement

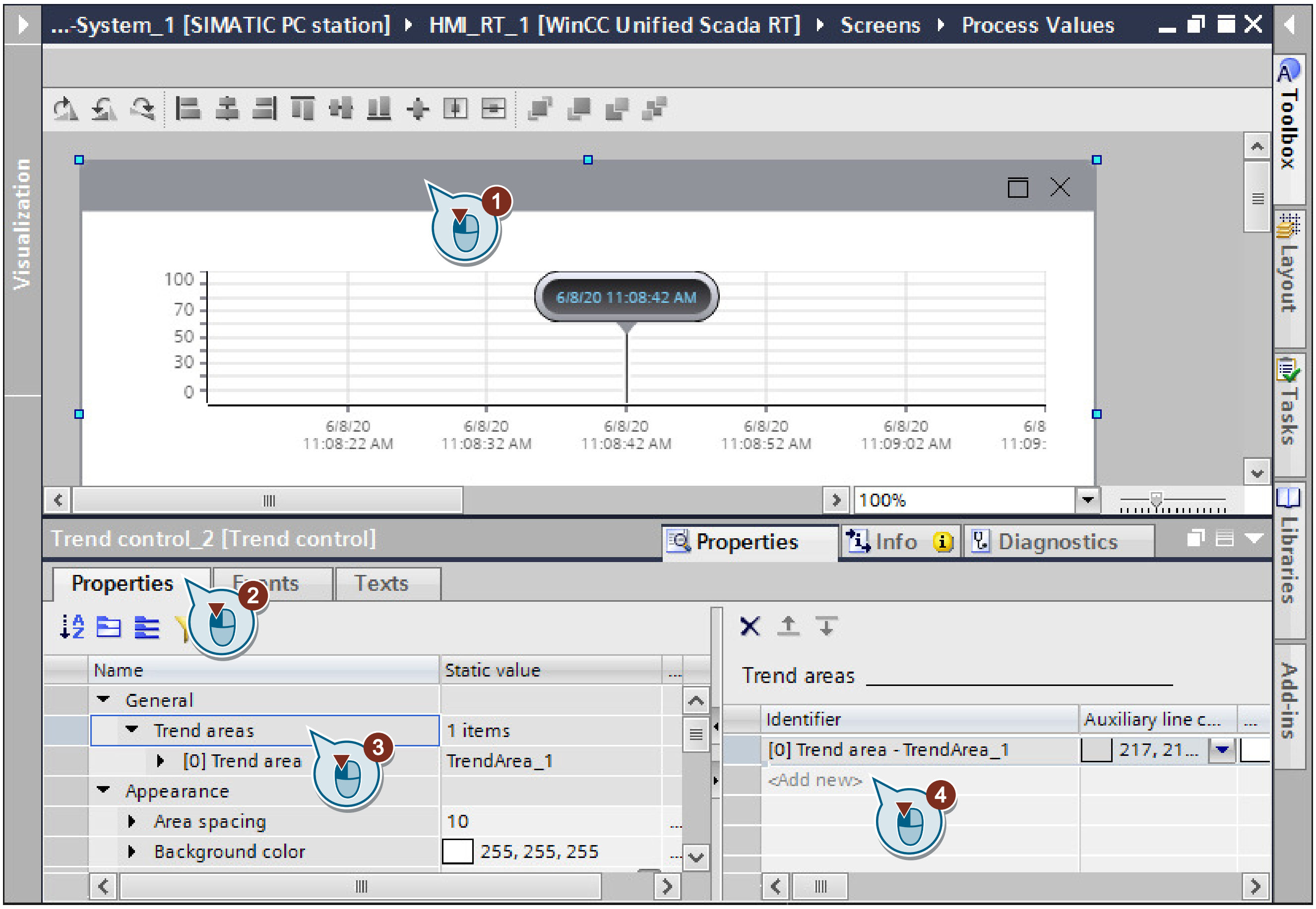
- The fill levels of the 3 tanks are created as logged tags "log_hop_level", "log_malt_level" and "log_water_level".
- The status of the 3 valves are created as logged tags "log_status_hop", "log_status_malt" and "log_status_water".

Procedure

1. Open the "ProcessValues" screen and insert a trend view.



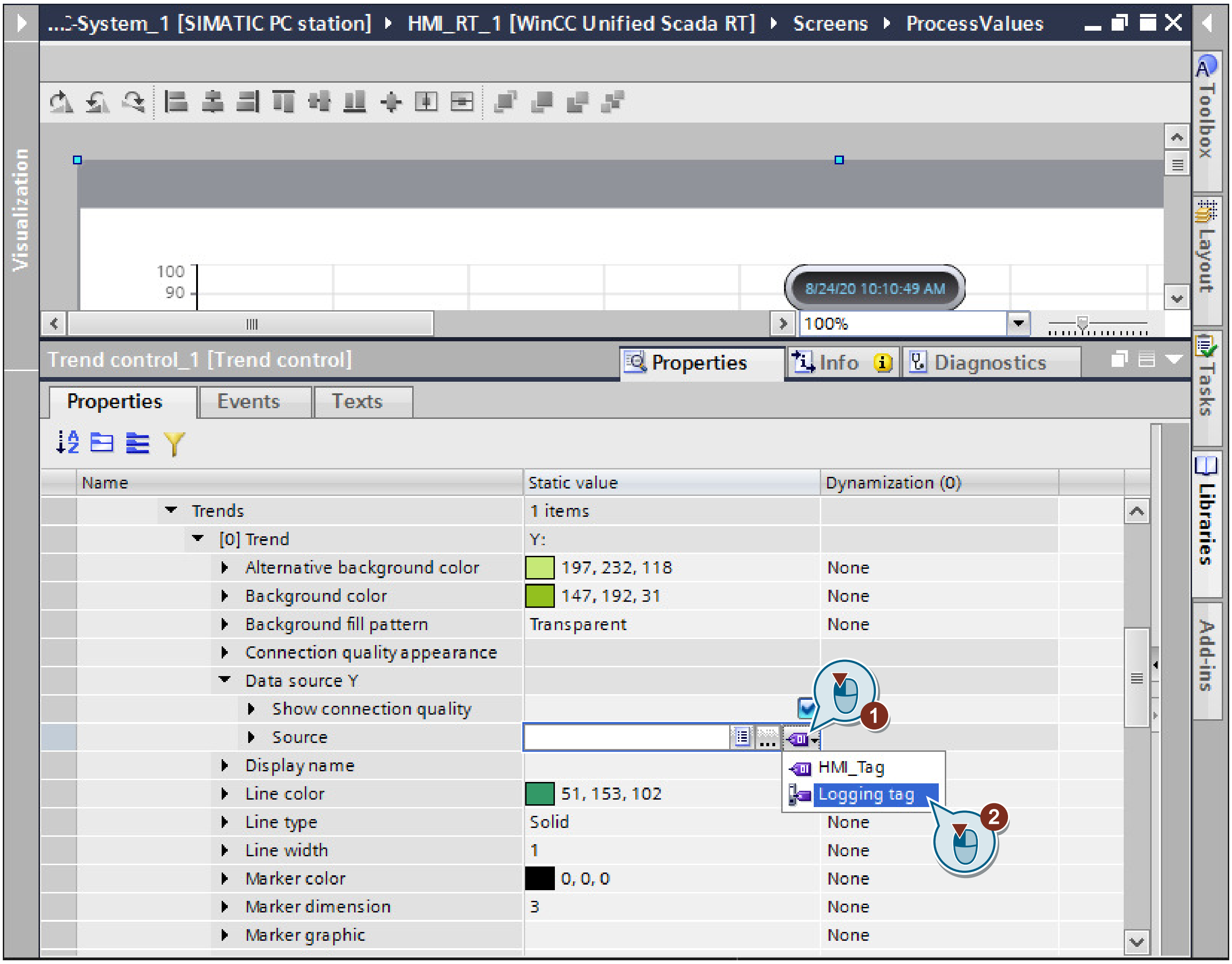
2. Add two trend areas to the trend view.



Name	Static value
General	
Trend areas	1 items
[0] Trend area	TrendArea_1
Appearance	
Area spacing	10
Background color	255, 255, 255

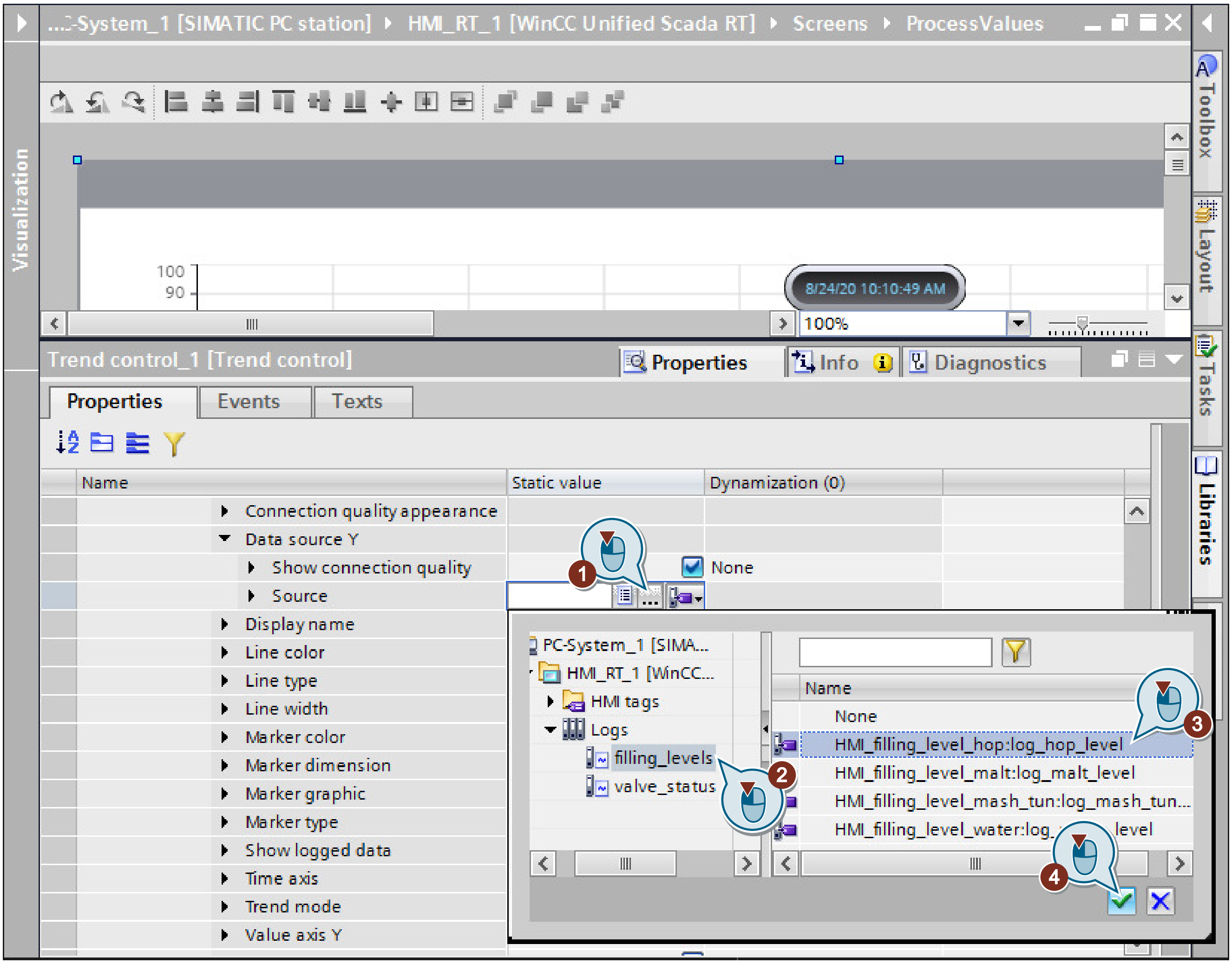
Identifier	Auxiliary line c...
[0] Trend area - TrendArea_1	217, 21...

3. Configure the first trend view with the fill levels:
   – Specify that the first trend is supplied from logged tags.

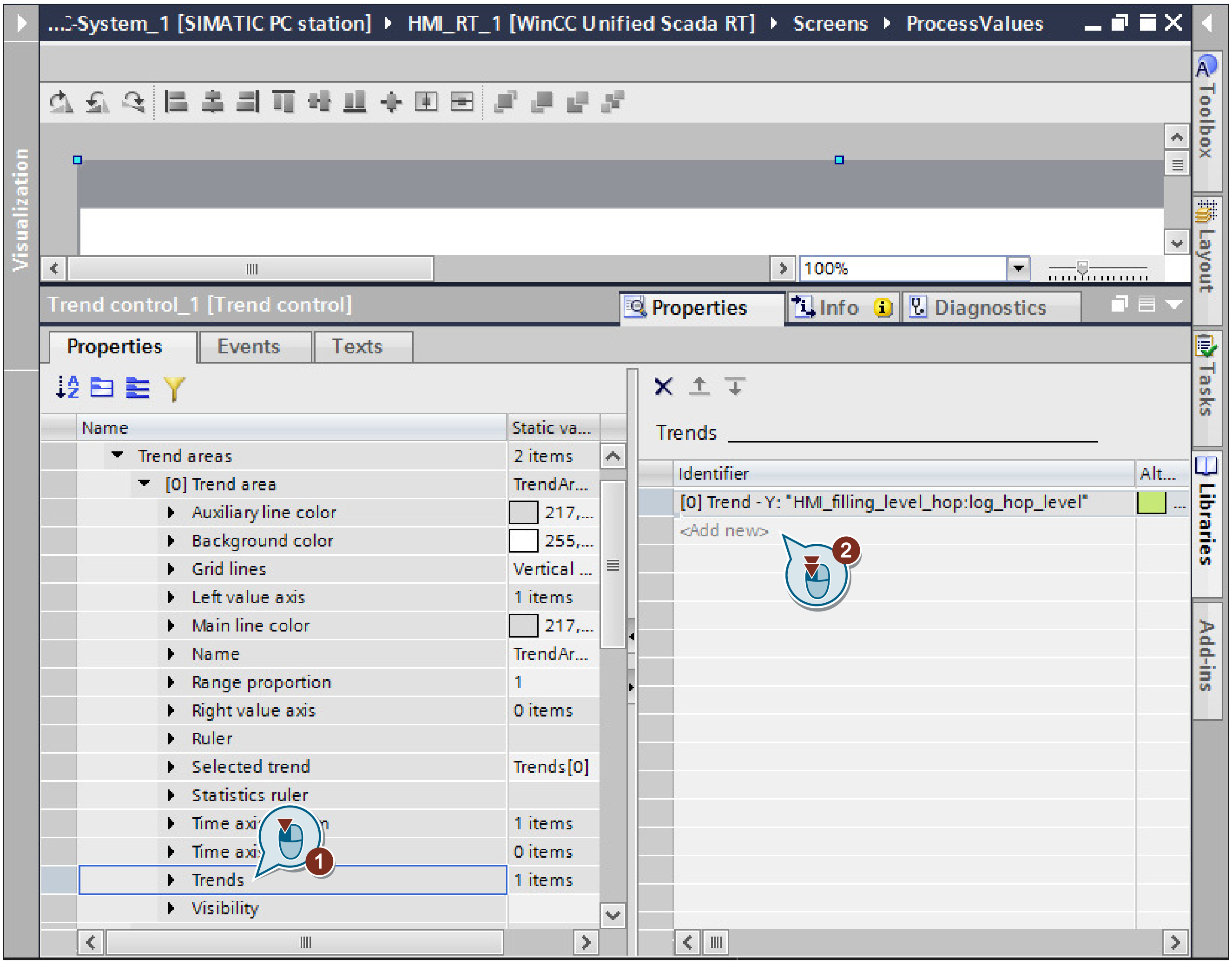


Name	Static value	Dynamization (0)
Trends	1 items	
[0] Trend	Y:	
Alternative background color	🟩 197, 232, 118	None
Background color	🟩 147, 192, 31	None
Background fill pattern	Transparent	None
Connection quality appearance		
Data source Y		
Show connection quality		
Source		
Display name		
Line color	🟩 51, 153, 102	
Line type	Solid	None
Line width	1	None
Marker color	⬛ 0, 0, 0	None
Marker dimension	3	None
Marker graphic		None

– Specify that the first trend is supplied with values from the logged tag "log_hop_level".



– Add two more trends to the first trend view.



...-System_1 [SIMATIC PC station] ▸ HMI_RT_1 [WinCC Unified Scada RT] ▸ Screens ▸ ProcessValues

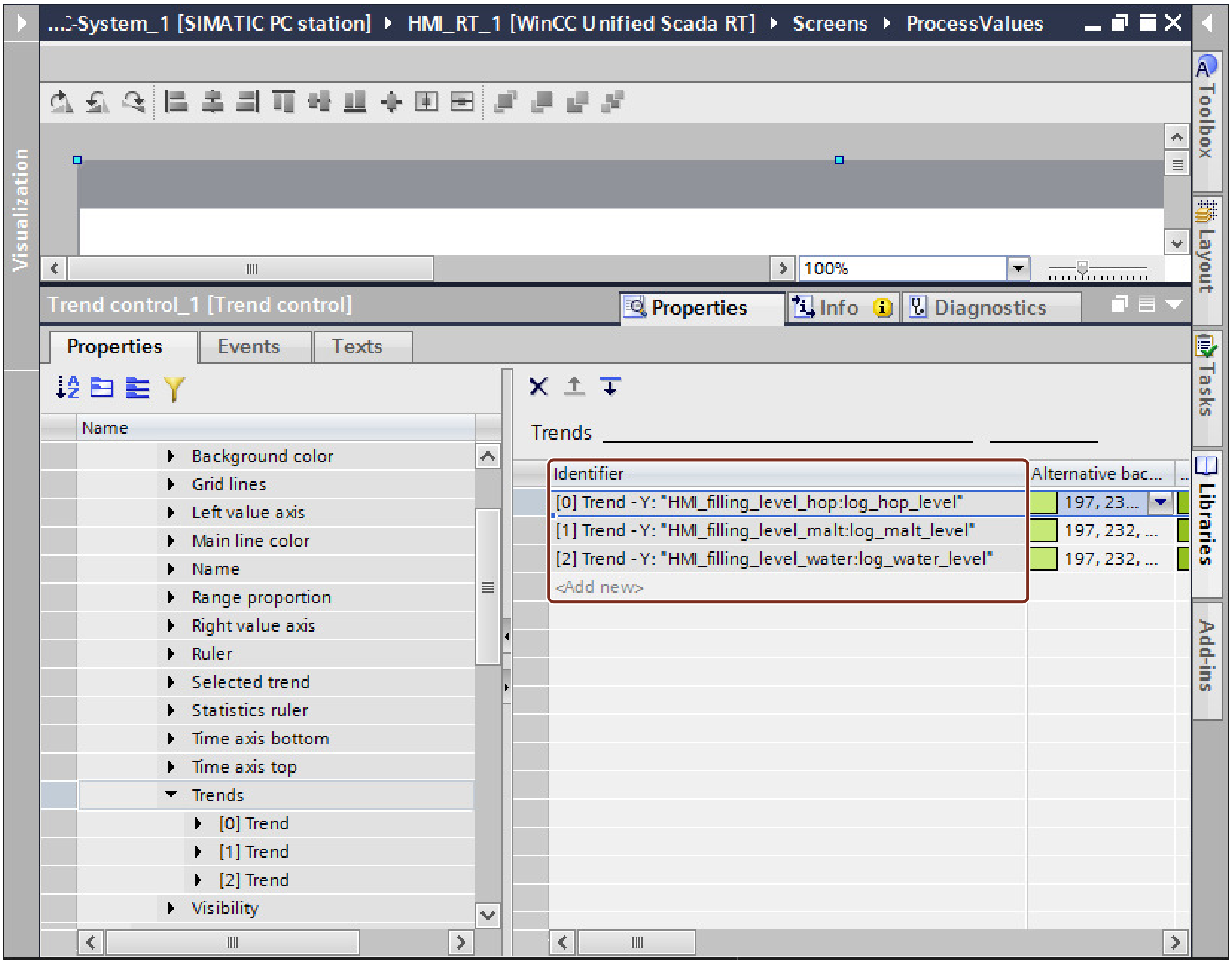
Trend control_1 [Trend control]

Name	Static va...
▾ Trend areas	2 items
▾ [0] Trend area	TrendAr...
▸ Auxiliary line color	217,...
▸ Background color	255,...
▸ Grid lines	Vertical ...
▸ Left value axis	1 items
▸ Main line color	217,...
▸ Name	TrendAr...
▸ Range proportion	1
▸ Right value axis	0 items
▸ Ruler	
▸ Selected trend	Trends[0]
▸ Statistics ruler	
▸ Time axi...n	1 items
▸ Time axi...	0 items
▸ Trends	1 items
▸ Visibility	

Trends

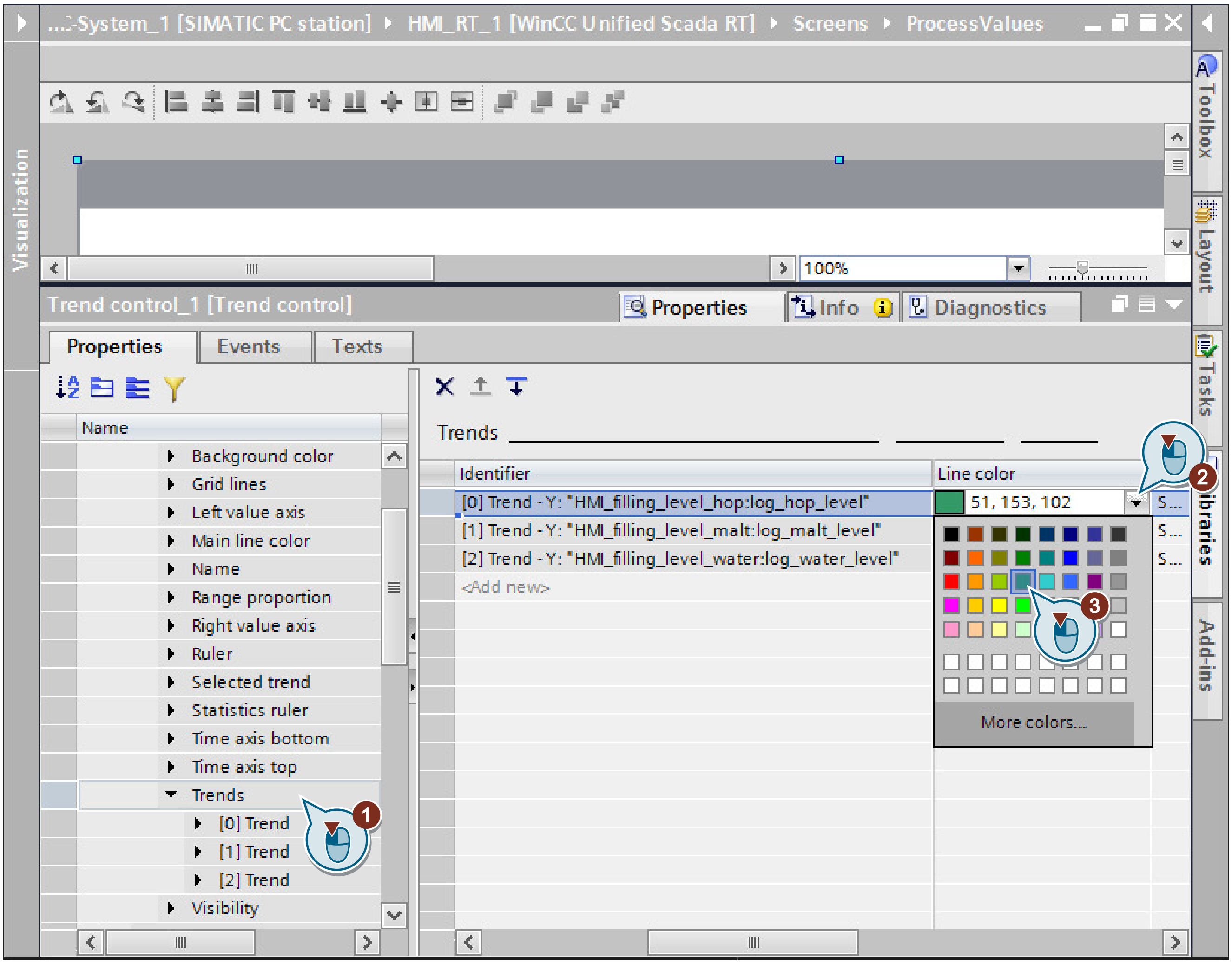
Identifier	Alt...
[0] Trend - Y: "HMI_filling_level_hop:log_hop_level"	🟩 ...
<Add new>	

– Specify that both trends are supplied with values from the logged tags "log_malt_level" and "log_water_level".



Identifier	Alternative bac...
[0] Trend - Y: "HMI_filling_level_hop:log_hop_level"	197, 23...
[1] Trend - Y: "HMI_filling_level_malt:log_malt_level"	197, 232, ...
[2] Trend - Y: "HMI_filling_level_water:log_water_level"	197, 232, ...
<Add new>	

– Assign the desired line color to the first trend.

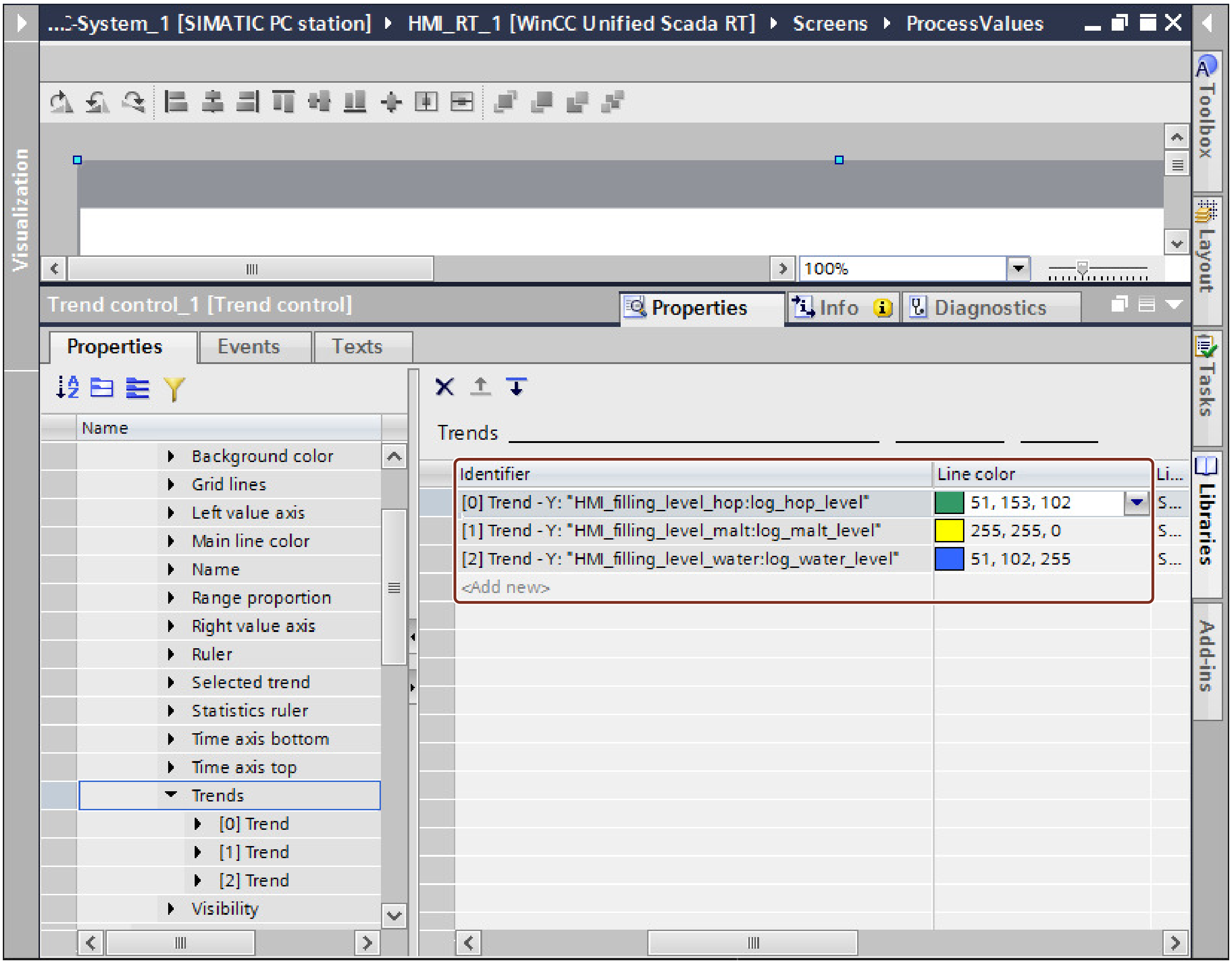


...-System_1 [SIMATIC PC station] ▸ HMI_RT_1 [WinCC Unified Scada RT] ▸ Screens ▸ ProcessValues

Trend control_1 [Trend control]

Identifier	Line color
[0] Trend - Y: "HMI_filling_level_hop:log_hop_level"	51, 153, 102
[1] Trend - Y: "HMI_filling_level_malt:log_malt_level"	
[2] Trend - Y: "HMI_filling_level_water:log_water_level"	
<Add new>	

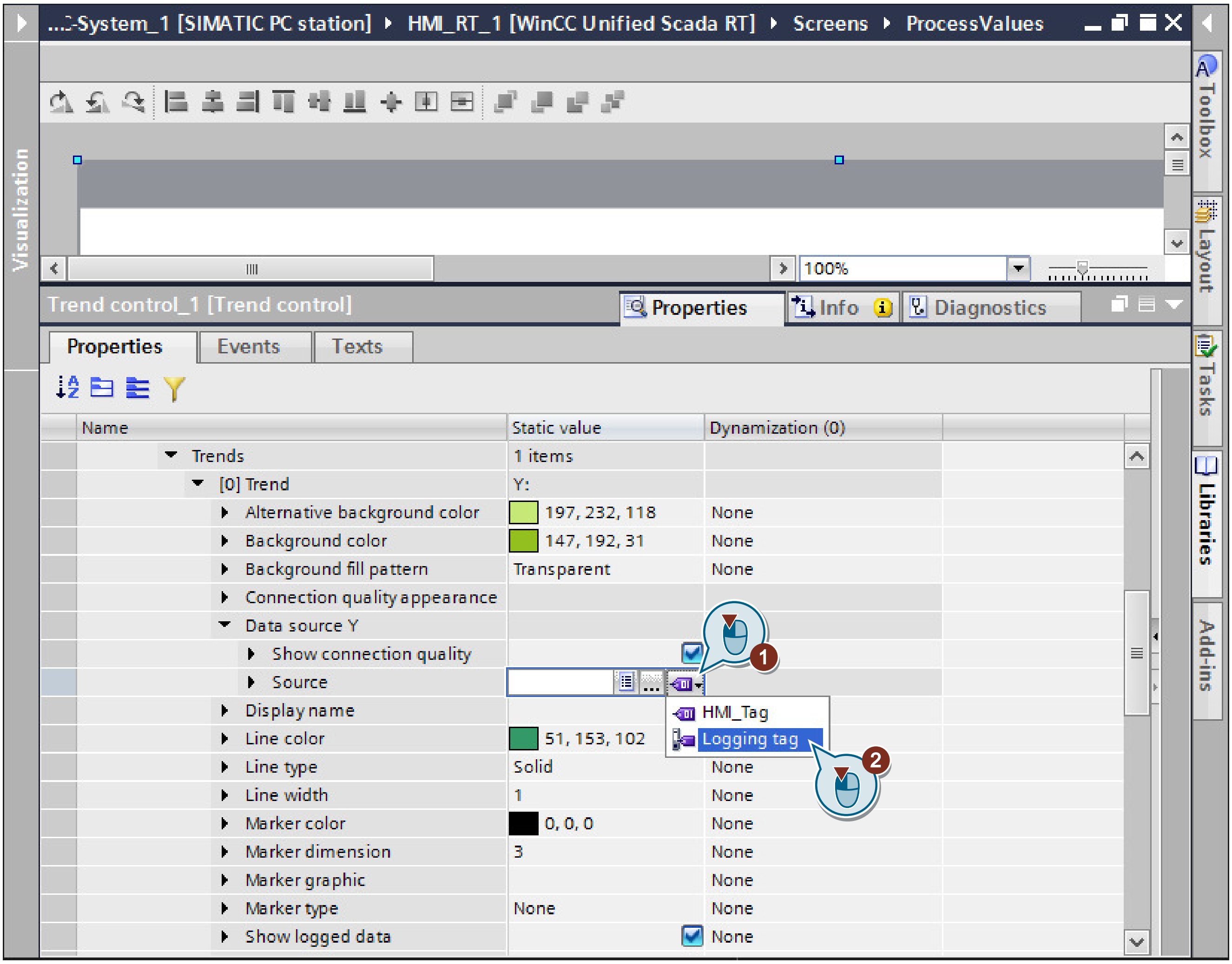
– Assign the desired line colors to the other two trends.



Identifier	Line color	Li...
[0] Trend - Y: "HMI_filling_level_hop:log_hop_level"	🟩 51, 153, 102	S...
[1] Trend - Y: "HMI_filling_level_malt:log_malt_level"	🟨 255, 255, 0	S...
[2] Trend - Y: "HMI_filling_level_water:log_water_level"	🟦 51, 102, 255	S...
<Add new>		

4. Configure the second trend view with the valve status:

– Specify that the first trend is supplied from logged tags.

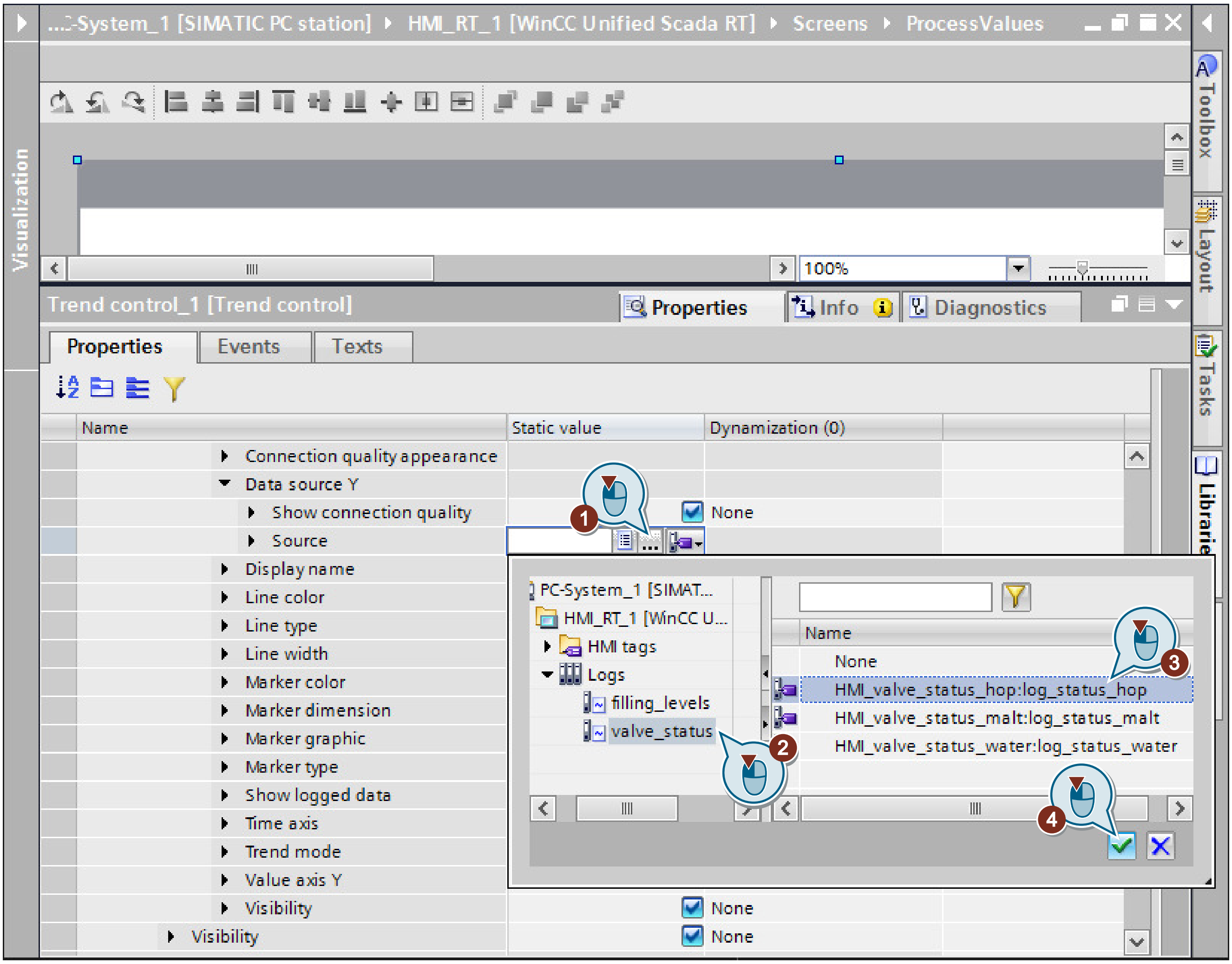


...-System_1 [SIMATIC PC station] ▸ HMI_RT_1 [WinCC Unified Scada RT] ▸ Screens ▸ ProcessValues

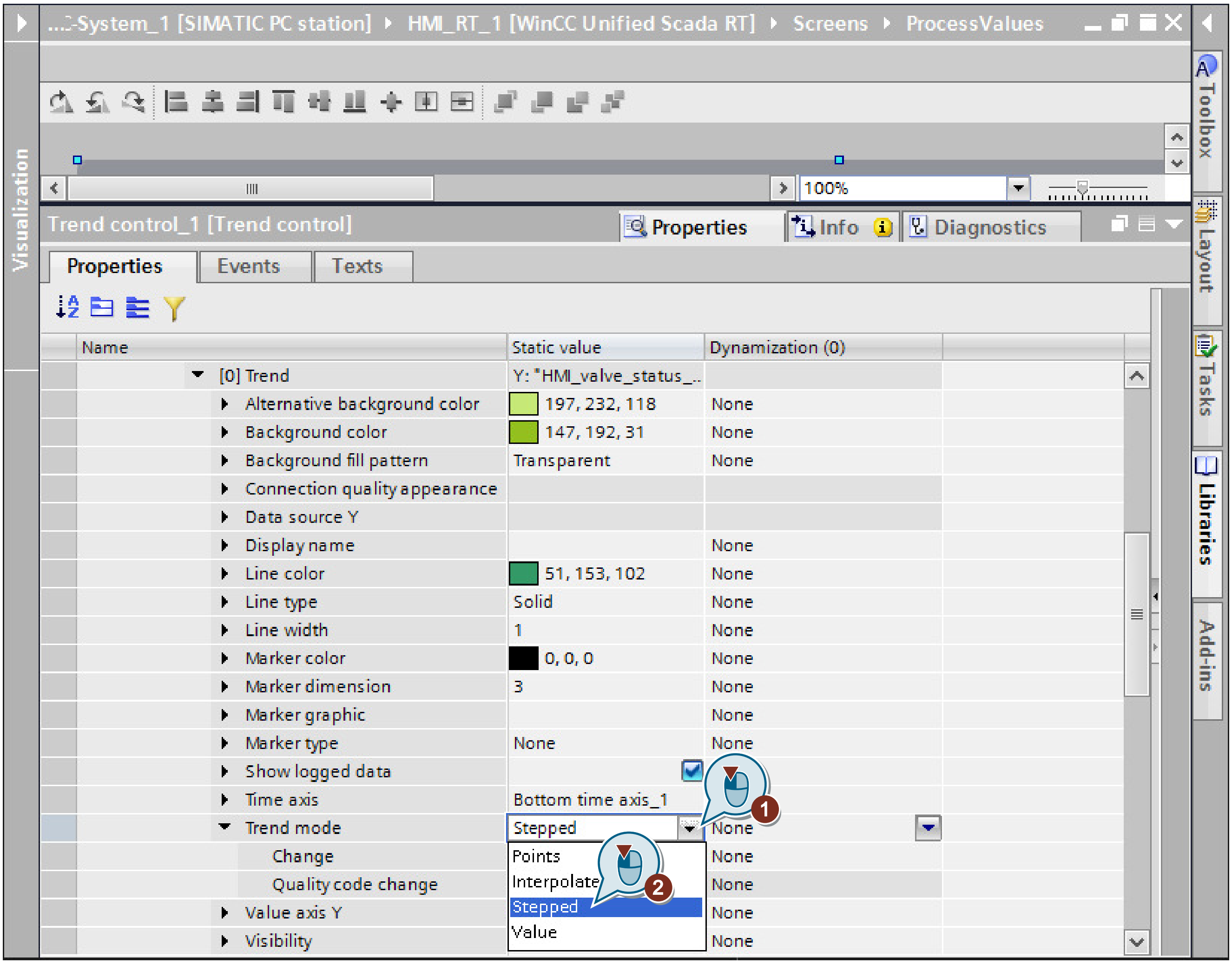
Trend control_1 [Trend control]

Name	Static value	Dynamization (0)
Trends	1 items	
[0] Trend	Y:	
Alternative background color	🟩 197, 232, 118	None
Background color	🟩 147, 192, 31	None
Background fill pattern	Transparent	None
Connection quality appearance		
Data source Y		
Show connection quality	☑	
Source		
Display name		
Line color	🟩 51, 153, 102	None
Line type	Solid	None
Line width	1	None
Marker color	⬛ 0, 0, 0	None
Marker dimension	3	None
Marker graphic		None
Marker type	None	None
Show logged data	☑	None

– Specify that the first trend is supplied with values from the logged tag "log_status_hop".



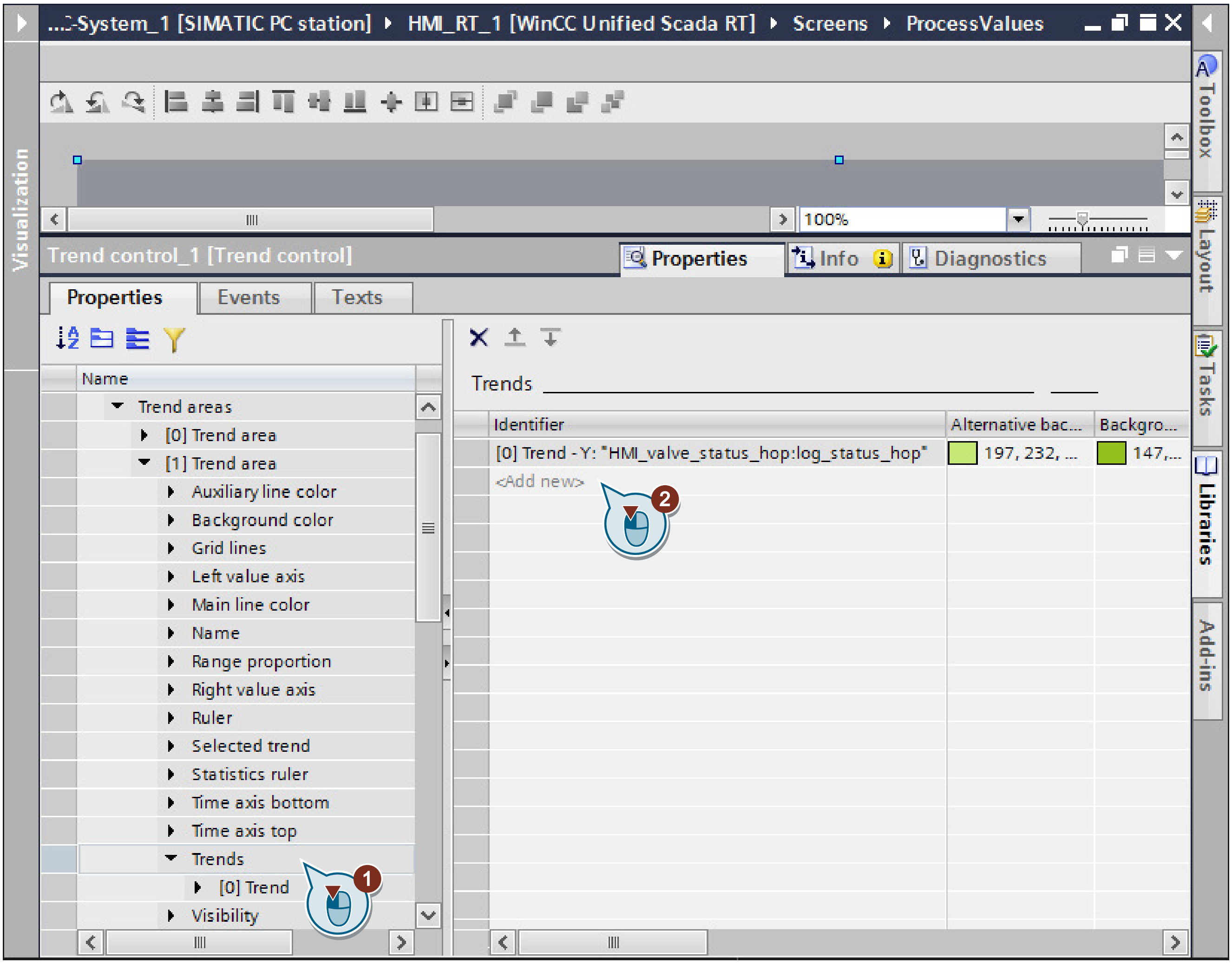
– Set the trend mode for the first trend "Stepped".



...-System_1 [SIMATIC PC station] ▸ HMI_RT_1 [WinCC Unified Scada RT] ▸ Screens ▸ ProcessValues
100%
Trend control_1 [Trend control]
Properties
Info
Diagnostics
Properties
Events
Texts
Visualization
Toolbox
Layout
Tasks
Libraries
Add-ins
1
2

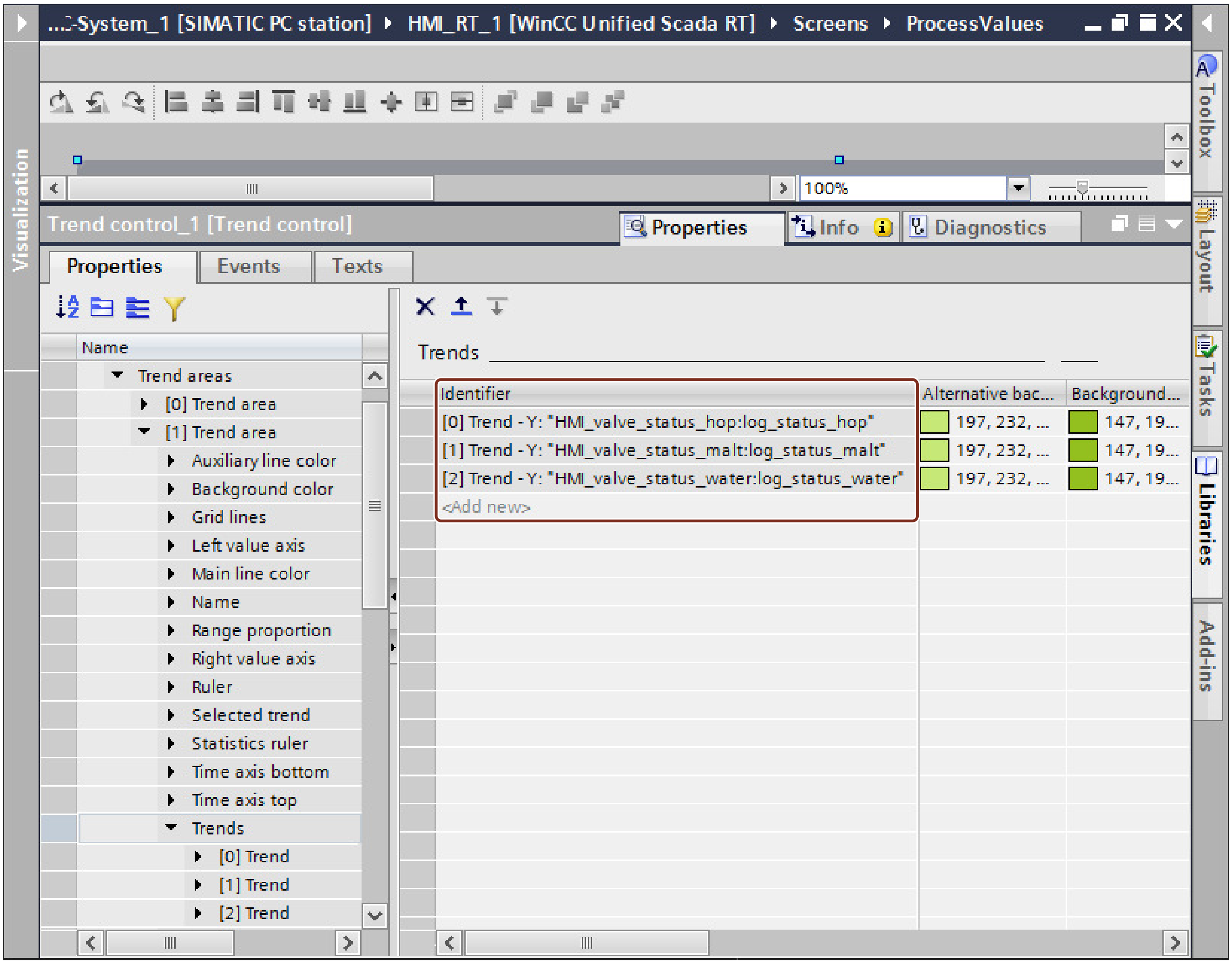
Name	Static value	Dynamization (0)
▾ [0] Trend	Y: "HMI_valve_status_...	
▸ Alternative background color	🟩 197, 232, 118	None
▸ Background color	🟩 147, 192, 31	None
▸ Background fill pattern	Transparent	None
▸ Connection quality appearance		
▸ Data source Y		
▸ Display name		None
▸ Line color	🟩 51, 153, 102	None
▸ Line type	Solid	None
▸ Line width	1	None
▸ Marker color	⬛ 0, 0, 0	None
▸ Marker dimension	3	None
▸ Marker graphic		None
▸ Marker type	None	None
▸ Show logged data	☑	
▸ Time axis	Bottom time axis_1	
▾ Trend mode	Stepped	None
Change	Points	None
Quality code change	Interpolate	None
▸ Value axis Y	Stepped	None
▸ Visibility	Value	None

– Add two more trends to the second trend view.



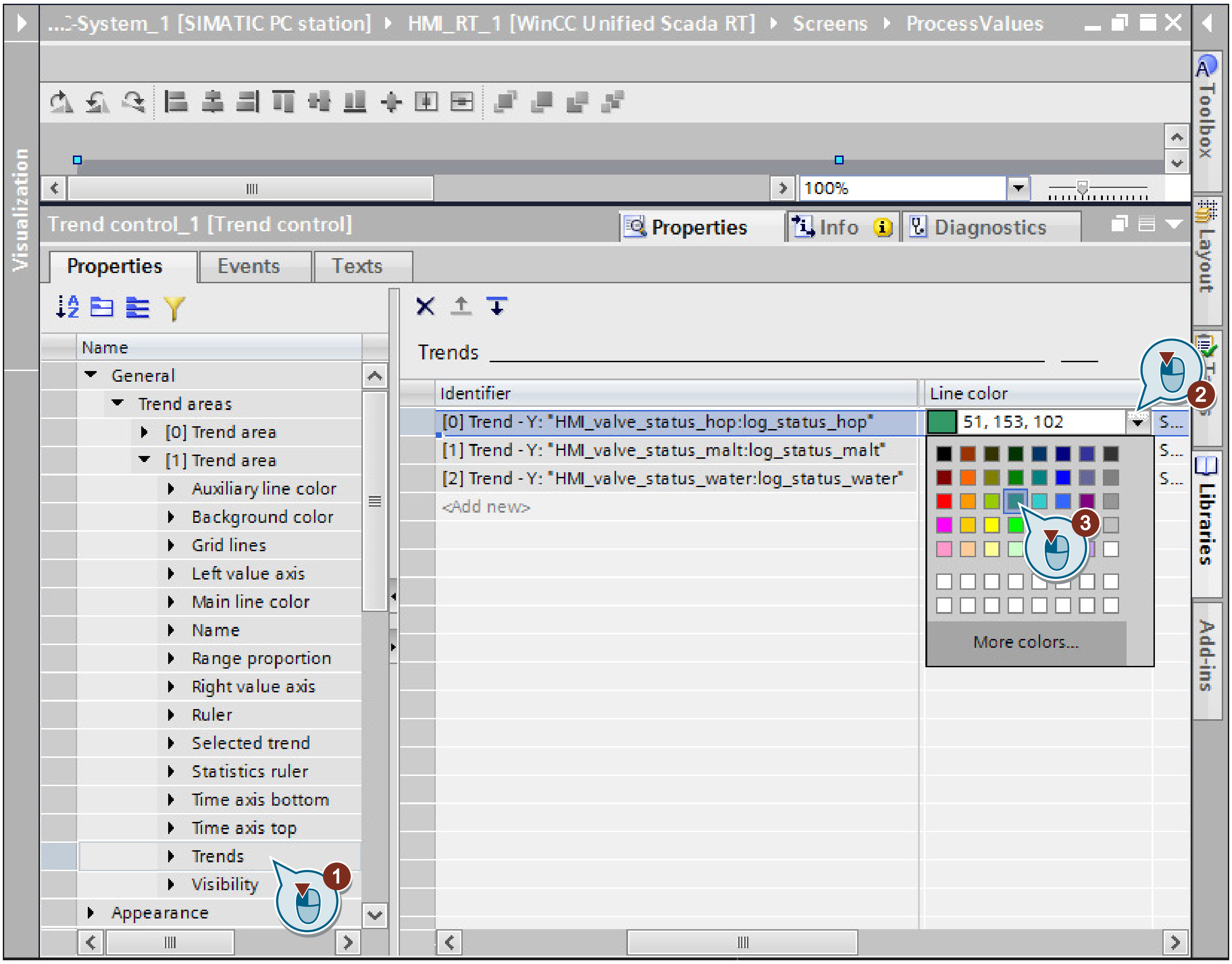
Identifier	Alternative bac...	Backgro...
[0] Trend - Y: "HMI_valve_status_hop:log_status_hop"	🟩 197, 232, ...	🟩 147,...
<Add new>		

– Specify that both trends are supplied with values from the logged tags "log_status_malt" and "log_status_water", and specify the trend mode "Stepped" for the two trends.



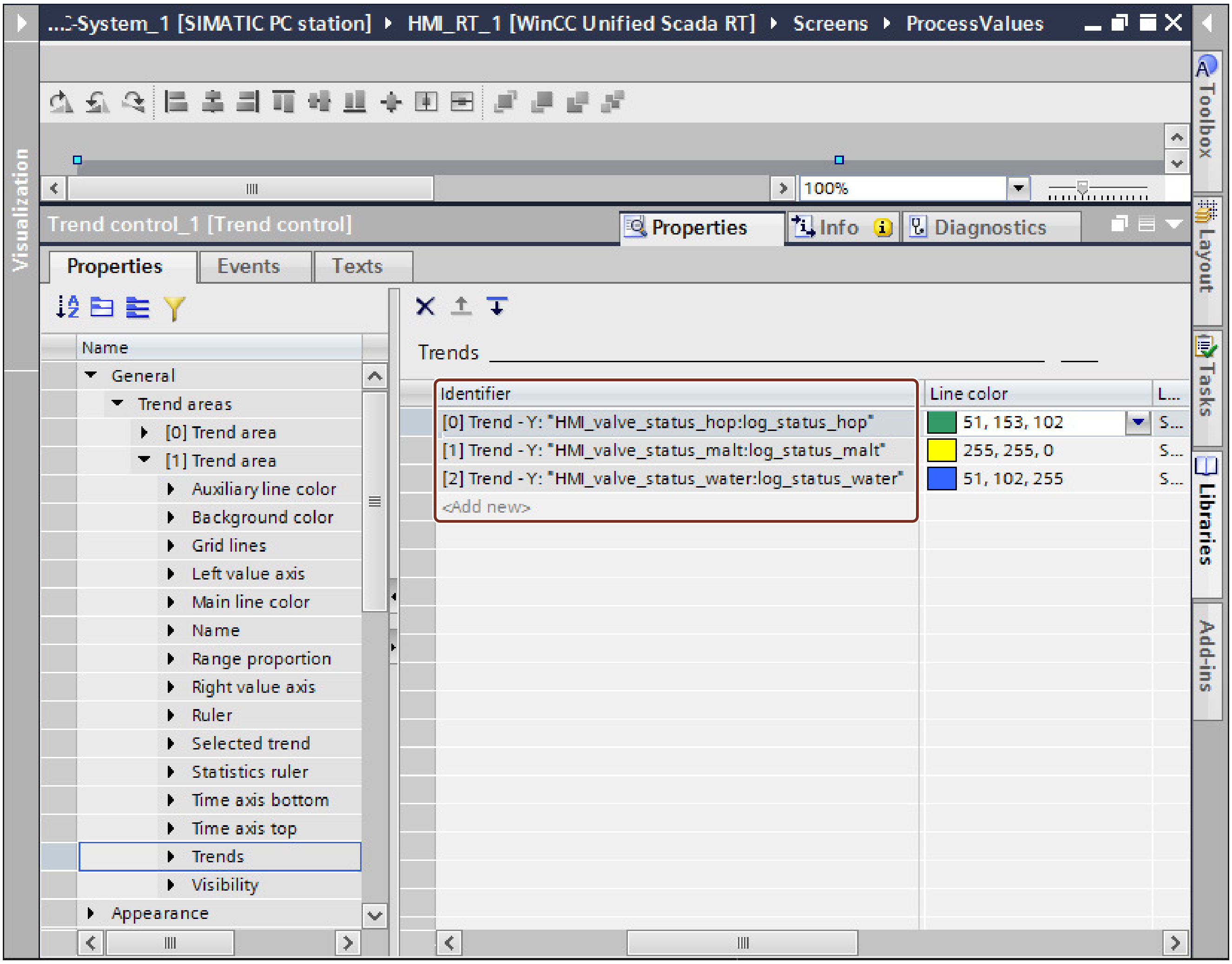
Identifier	Alternative bac...	Background...
[0] Trend - Y: "HMI_valve_status_hop:log_status_hop"	🟩 197, 232, ...	🟩 147, 19...
[1] Trend - Y: "HMI_valve_status_malt:log_status_malt"	🟩 197, 232, ...	🟩 147, 19...
[2] Trend - Y: "HMI_valve_status_water:log_status_water"	🟩 197, 232, ...	🟩 147, 19...
<Add new>		

– Assign the desired line color to the first trend.



Identifier	Line color	
[0] Trend - Y: "HMI_valve_status_hop:log_status_hop"	🟩 51, 153, 102	S...
[1] Trend - Y: "HMI_valve_status_malt:log_status_malt"		S...
[2] Trend - Y: "HMI_valve_status_water:log_status_water"		S...
<Add new>		

– Assign the desired line colors to the other two trends.



Identifier	Line color	L...
[0] Trend - Y: "HMI_valve_status_hop:log_status_hop"	🟩 51, 153, 102	S...
[1] Trend - Y: "HMI_valve_status_malt:log_status_malt"	🟨 255, 255, 0	S...
[2] Trend - Y: "HMI_valve_status_water:log_status_water"	🟦 51, 102, 255	S...
<Add new>		

Result

The trend view for displaying the fill levels and the valve status is configured.

Configuring parameter sets (optional) 4

4.1 Parameter sets

Parameter set type

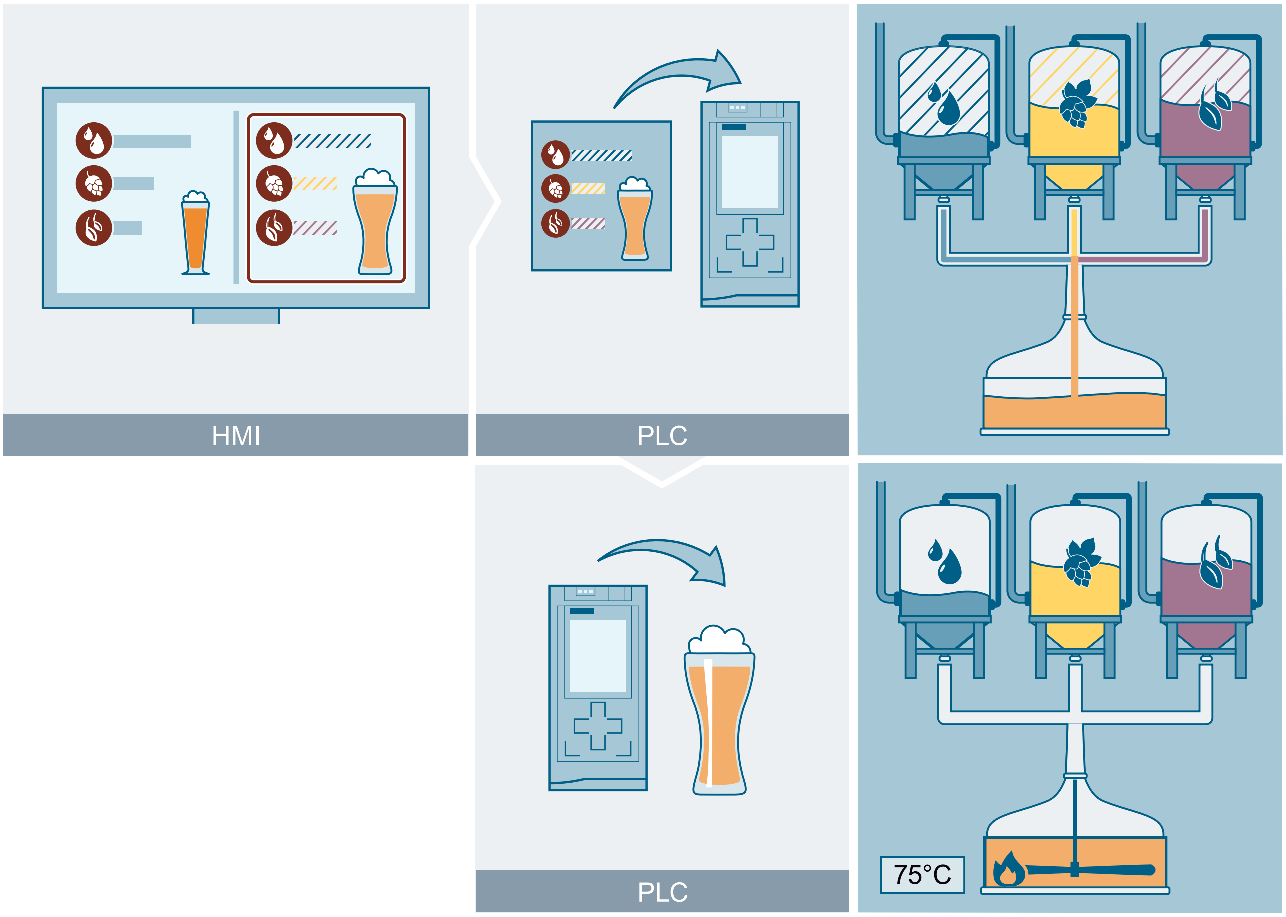
The parameter set type specifies the parameters and their properties according to which a product is manufactured. A parameter set type is a template for different products with the same basic ingredients.

No specific parameter values are stored in the parameter set type. For this purpose, a parameter set is created based on the parameter set type for each product to be manufactured, for example, Hefeweizen or Pilsner.

Parameter sets should be used when frequently changing between products with similar composition. In the process, you can quickly switch to another product. Basic changes to the structure are carried out centrally on the parameter set type.

Parameter sets in the brewing process

The brewmaster creates or imports one or more parameter sets. The shift manager selects the parameter set for the product to be manufactured. The parameter set is transferred from the HMI to the controller and the brewing process starts. The controller runs the user program and uses the parameter values of the basic ingredients for production. The individual ingredients are fed into the mash tun. The brewing process continues according to the user program.



4.2 Using parameter sets

Introduction

In the following, a parameter set type is to be configured. To do this you have to configure a controller. If you want to test your configuration afterwards, either load the project onto a real controller or use "PLCSim" to simulate a controller.

Requirement

Configuration requirement

To configure a controller, you need a "STEP 7" license.

Requirements for testing the configuration

To test the configuration of the parameter sets with "PLCSim", you need an installed and configured version of "PLCSim". For information on the installation and configuration, refer to the "PLCSim" function manual.

Procedure

For communication between the HMI and the controller, perform the following steps on the controller:

1. Add a controller with which the HMI can communicate.
2. Create a user data type (UDT).
3. Create a program block and store the UDT as data type.

On the HMI side, connect the HMI to the controller and then configure the parameter set type. To do this, carry out the following steps:

1. Link the connection of HMI and controller via an HMI tag with the program block.
2. Create the parameter set type.
3. Insert the parameter set control into the control panel of the HMI.

4.3 Configuring a controller

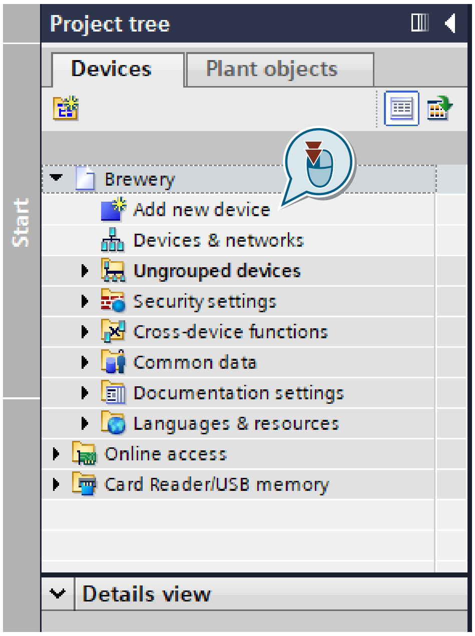
4.3.1 Adding and networking a controller

Introduction

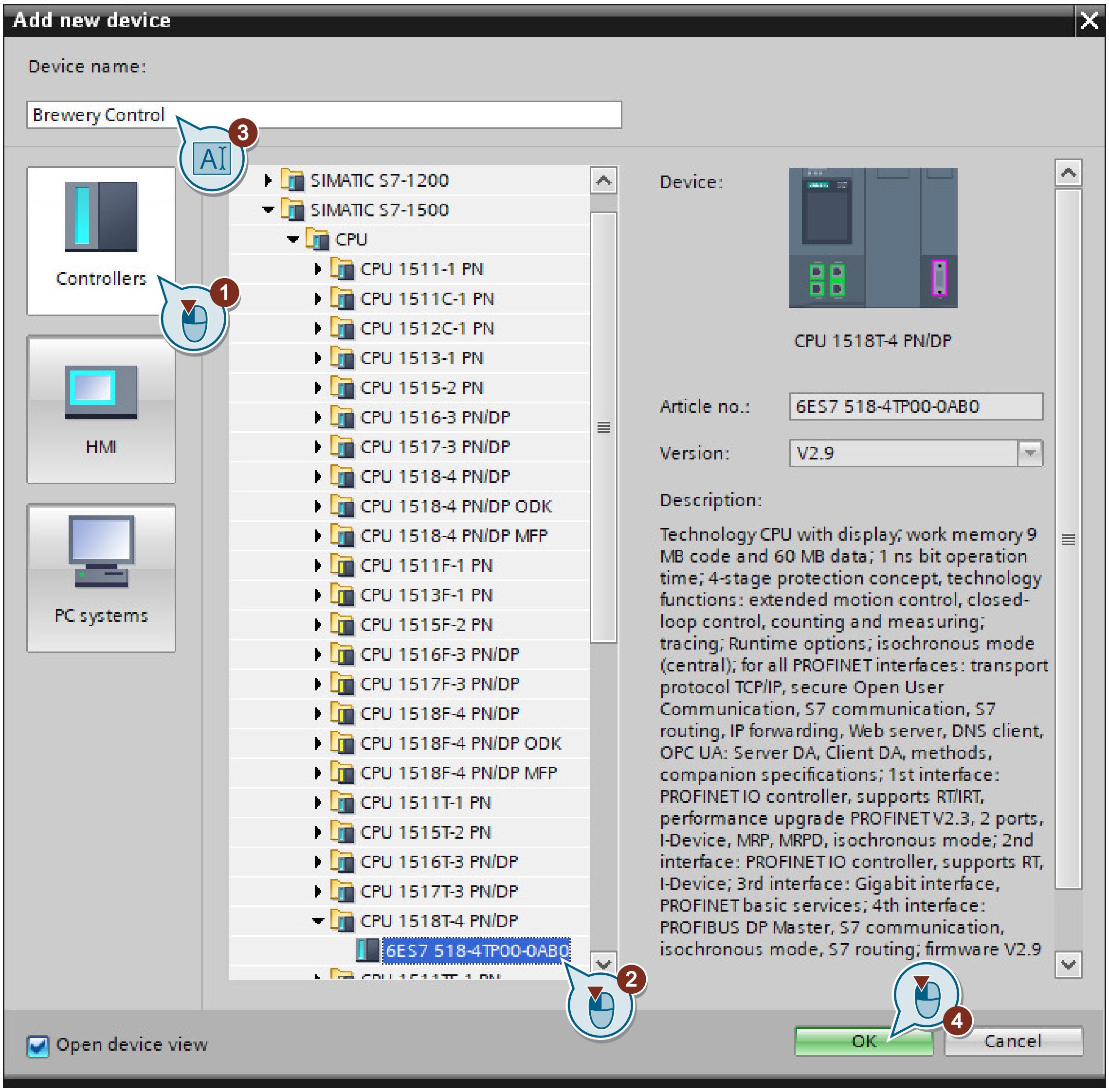
To transfer parameter sets, first configure a controller and then the connection to the HMI.

Procedure

1. Add a new device.



2. Select the "1518T-4 PN/DP" controller and rename it to "Brewery Control".



Add new device

Device name:

Brewery Control

- SIMATIC S7-1200
- SIMATIC S7-1500
  - CPU
    - CPU 1511-1 PN
    - CPU 1511C-1 PN
    - CPU 1512C-1 PN
    - CPU 1513-1 PN
    - CPU 1515-2 PN
    - CPU 1516-3 PN/DP
    - CPU 1517-3 PN/DP
    - CPU 1518-4 PN/DP
    - CPU 1518-4 PN/DP ODK
    - CPU 1518-4 PN/DP MFP
    - CPU 1511F-1 PN
    - CPU 1513F-1 PN
    - CPU 1515F-2 PN
    - CPU 1516F-3 PN/DP
    - CPU 1517F-3 PN/DP
    - CPU 1518F-4 PN/DP
    - CPU 1518F-4 PN/DP ODK
    - CPU 1518F-4 PN/DP MFP
    - CPU 1511T-1 PN
    - CPU 1515T-2 PN
    - CPU 1516T-3 PN/DP
    - CPU 1517T-3 PN/DP
    - CPU 1518T-4 PN/DP
      - 6ES7 518-4TP00-0AB0

Device:

CPU 1518T-4 PN/DP

Article no.: 6ES7 518-4TP00-0AB0

Version: V2.9

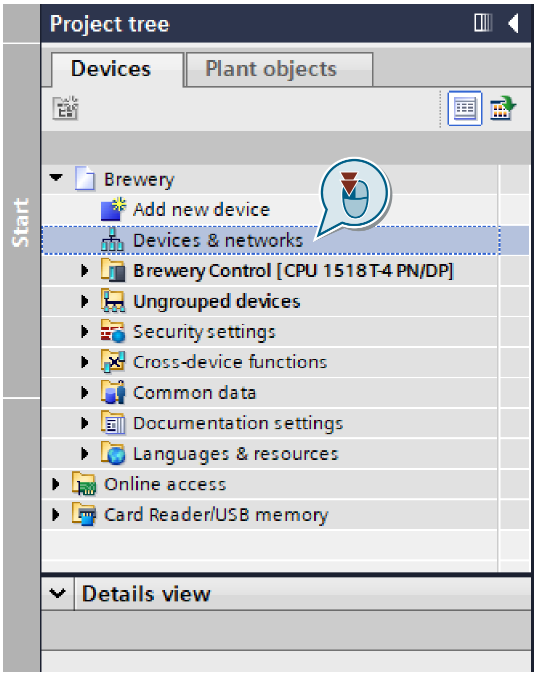
Description:

Technology CPU with display; work memory 9 MB code and 60 MB data; 1 ns bit operation time; 4-stage protection concept, technology functions: extended motion control, closed-loop control, counting and measuring; tracing; Runtime options; isochronous mode (central); for all PROFINET interfaces: transport protocol TCP/IP, secure Open User Communication, S7 communication, S7 routing, IP forwarding, Web server, DNS client, OPC UA: Server DA, Client DA, methods, companion specifications; 1st interface: PROFINET IO controller, supports RT/IRT, performance upgrade PROFINET V2.3, 2 ports, I-Device, MRP, MRPD, isochronous mode; 2nd interface: PROFINET IO controller, supports RT, I-Device; 3rd interface: Gigabit interface, PROFINET basic services; 4th interface: PROFIBUS DP Master, S7 communication, isochronous mode, S7 routing; firmware V2.9

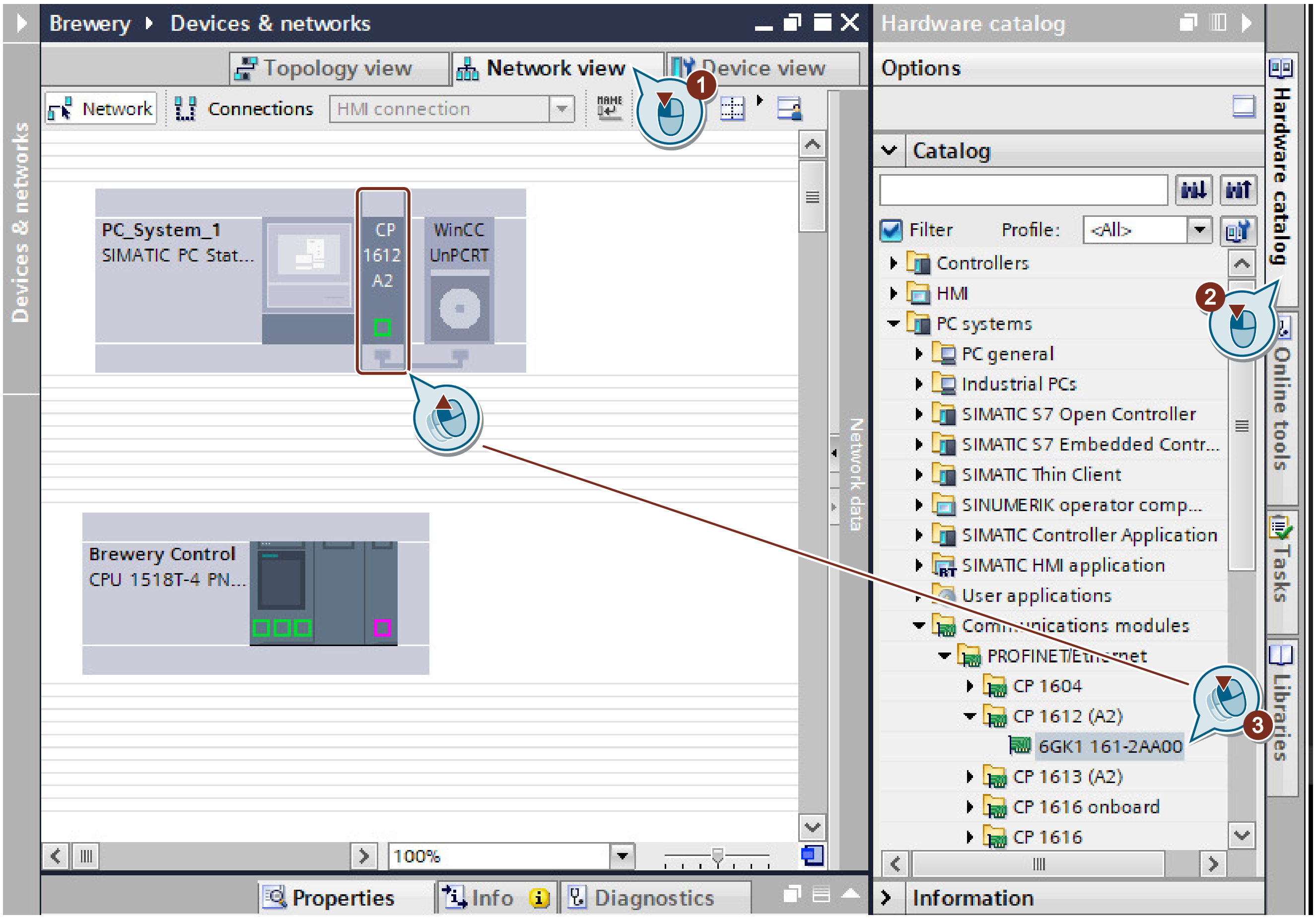
☑ Open device view

OK Cancel

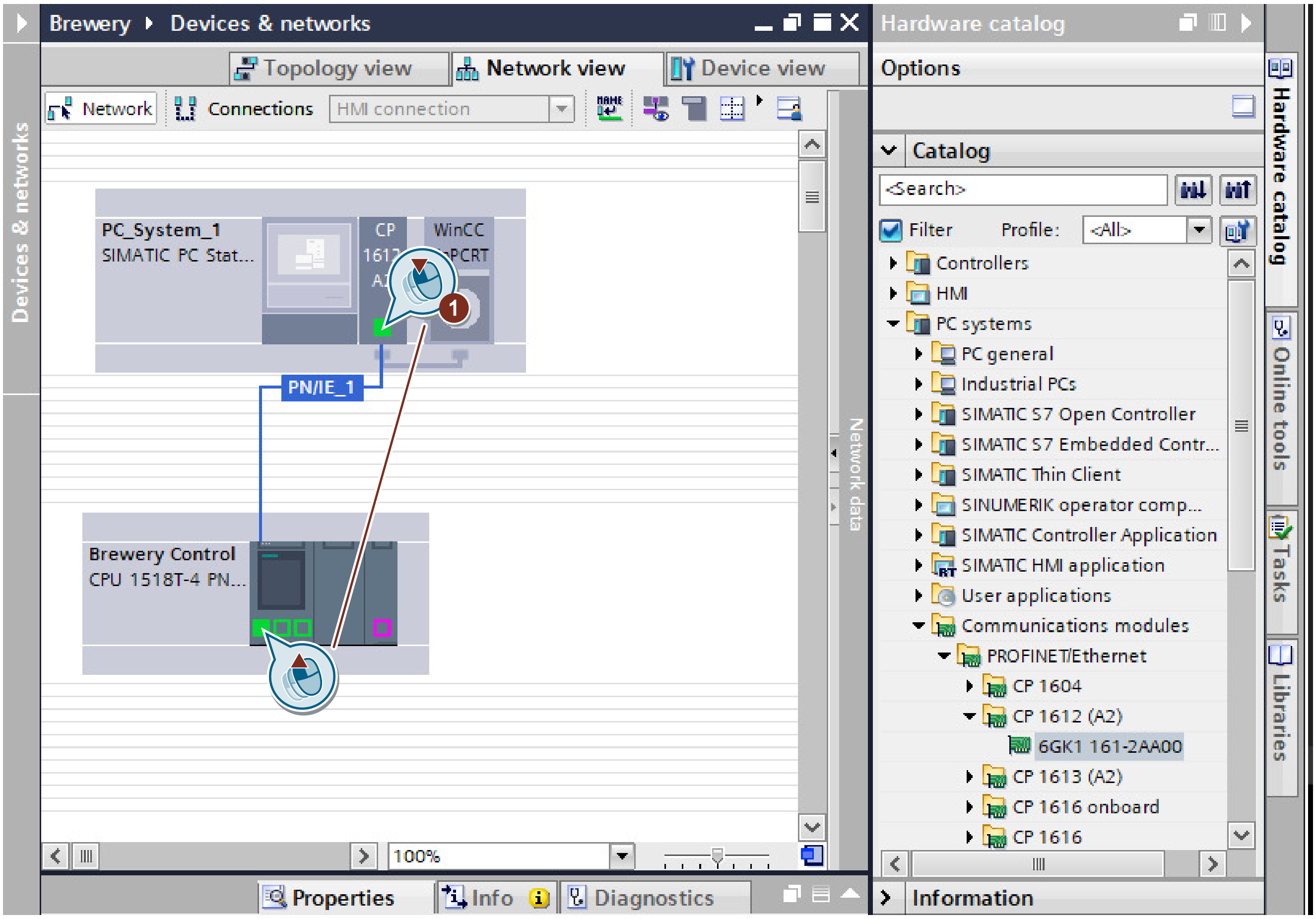
3. Switch to "Devices & Networks".



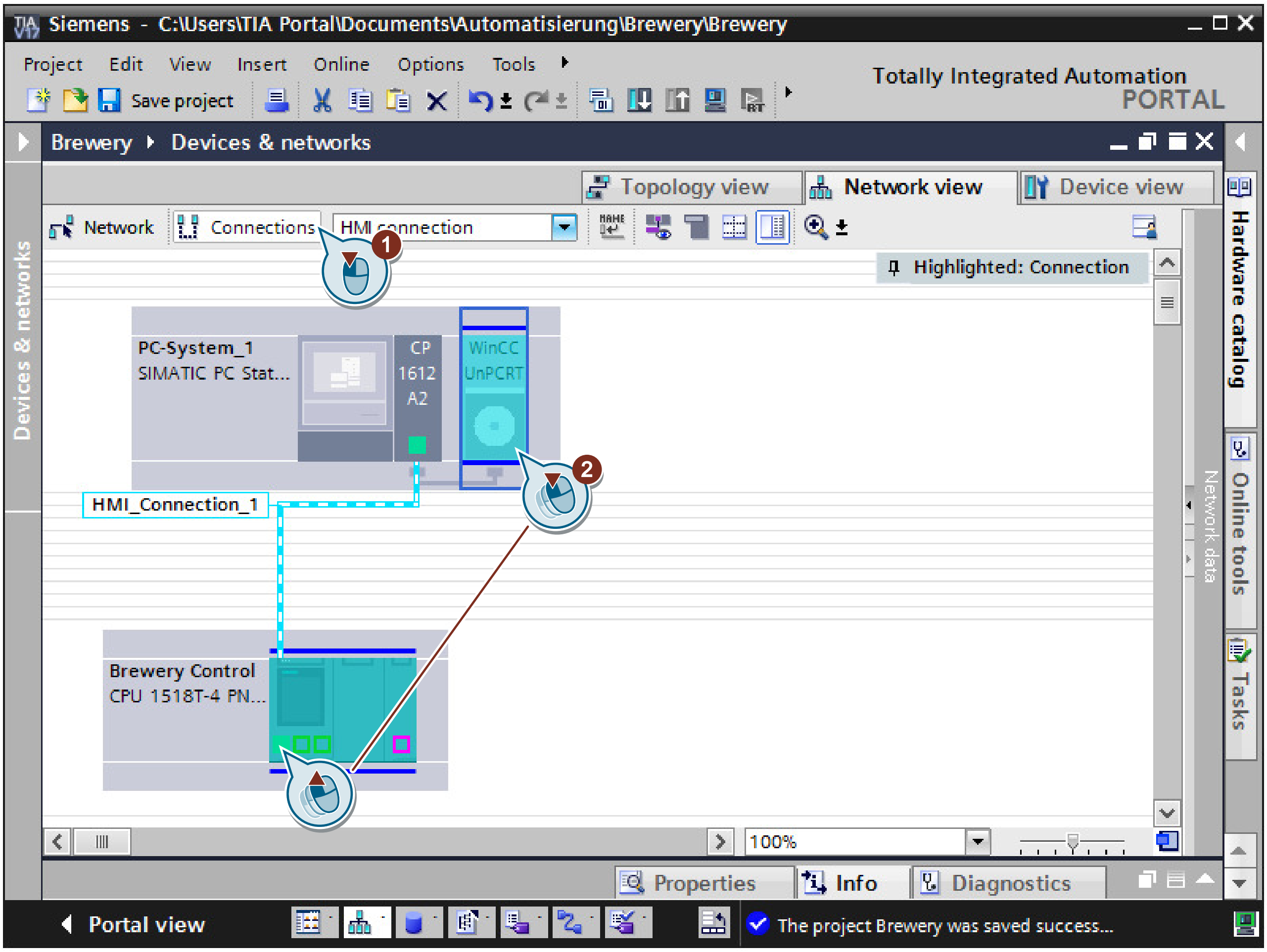
4. Add the "CP 1612 (A2)" communication module to the HMI.



5. Network the controller with the new HMI interface.



6. Create the HMI connection between the HMI and controller.



4.3.2 Create PLC data type and add to the library

Introduction

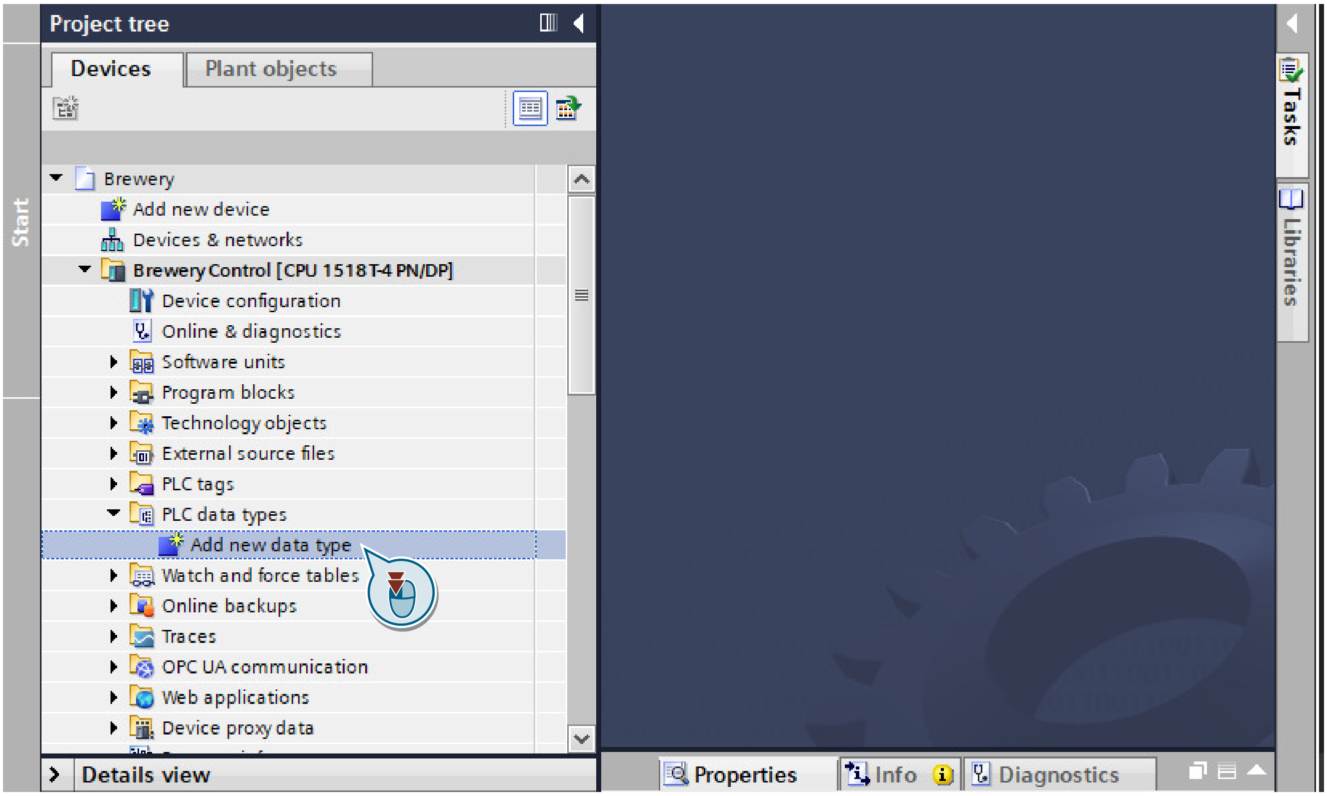
Here's how to create a user data type on the controller and add it to the project library. The user data type is later used as the basis for the parameter set type.

The following elements should be created in the user data type:

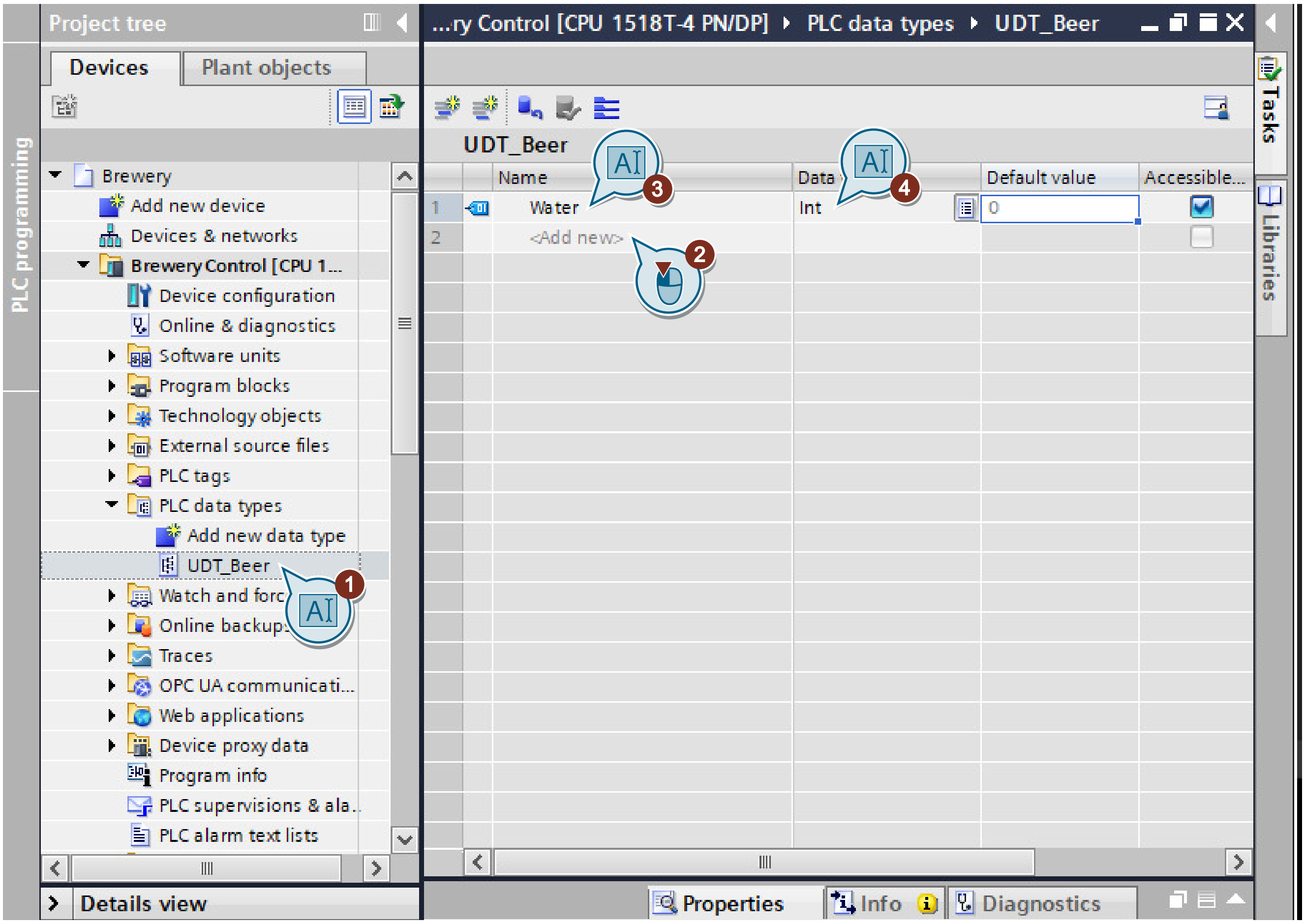
Element name	Use	Data type
water	Quantity of water	Int
hop	Quantity of hops	Int
malt	Quantity of malt	Int

Create user data type and add to the library

1. Add a new data type.

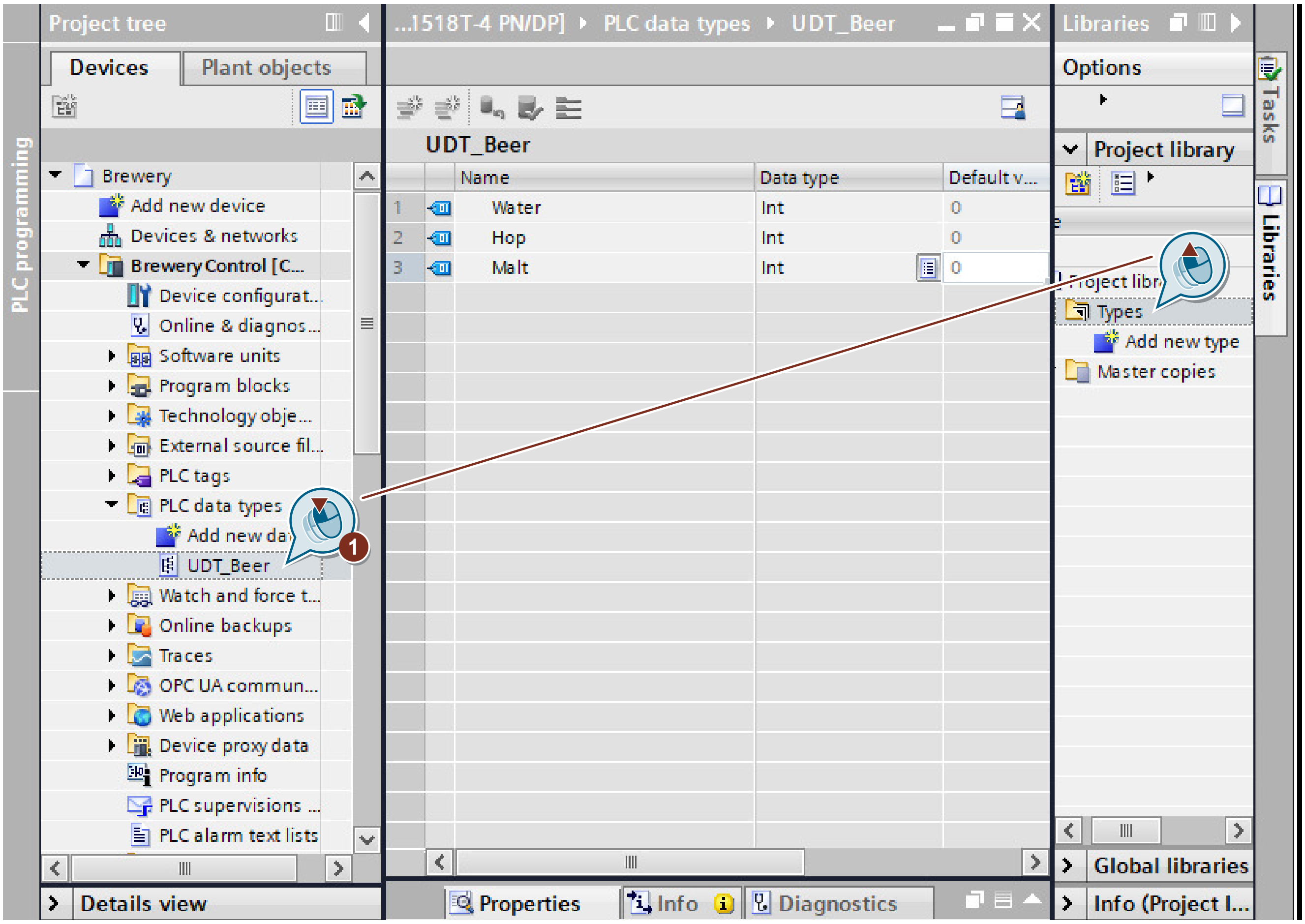


2. Rename the user data type in "UDT_Beer" and add the elements "Water", "Hop" and "Malt". Select the "Int" data type for all elements.



	Name	Data	Default value	Accessible...
1	Water	Int	0	☑
2	<Add new>			□

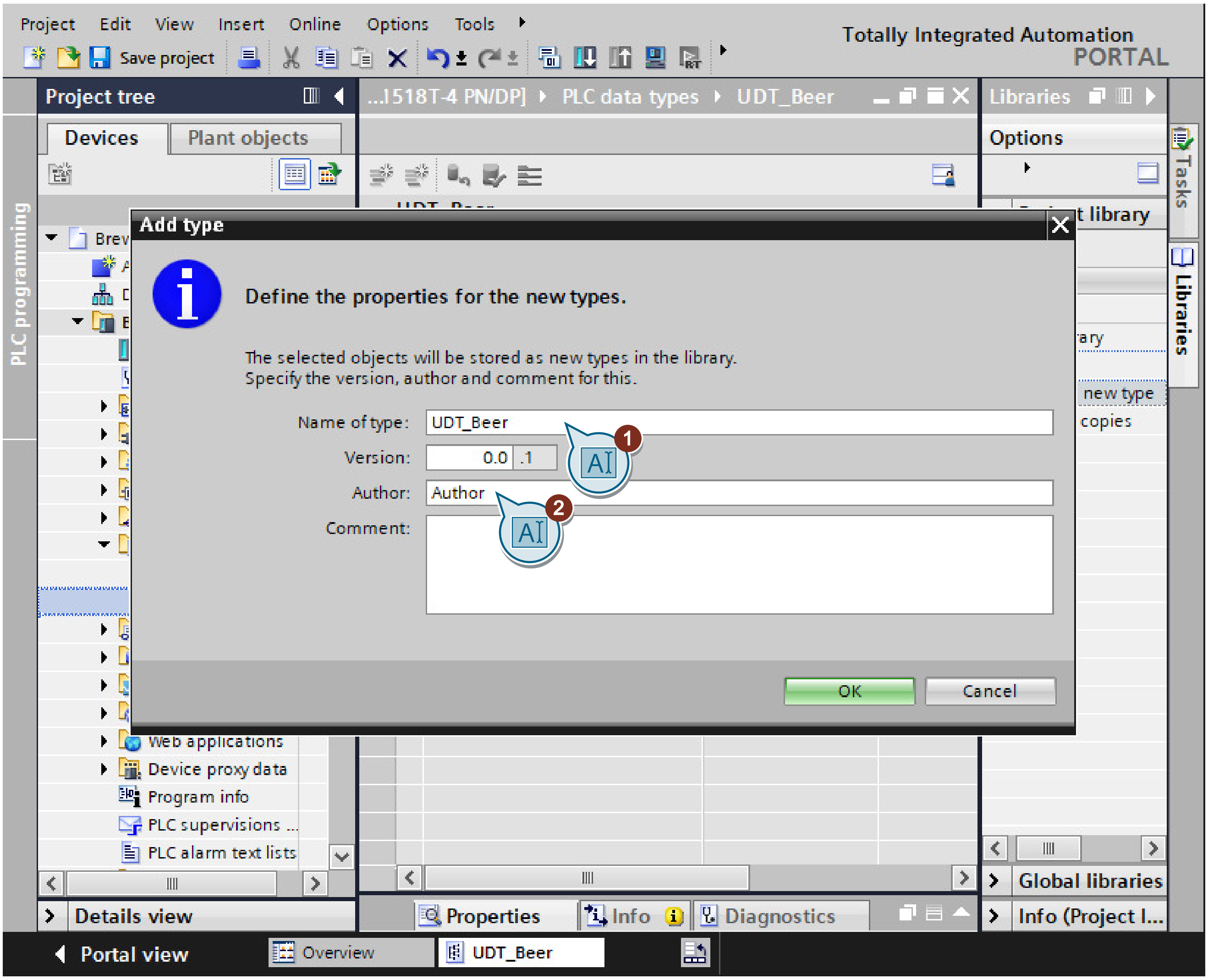
3. Add the new user data type to the project library.



UDT_Beer

	Name	Data type	Default v...
1	Water	Int	0
2	Hop	Int	0
3	Malt	Int	0

4. Select the user data type in the library with "UDT_Beer".



Add type

Define the properties for the new types.

The selected objects will be stored as new types in the library.
Specify the version, author and comment for this.

Name of type: UDT_Beer

Version: 0.0 .1

Author: Author

Comment:

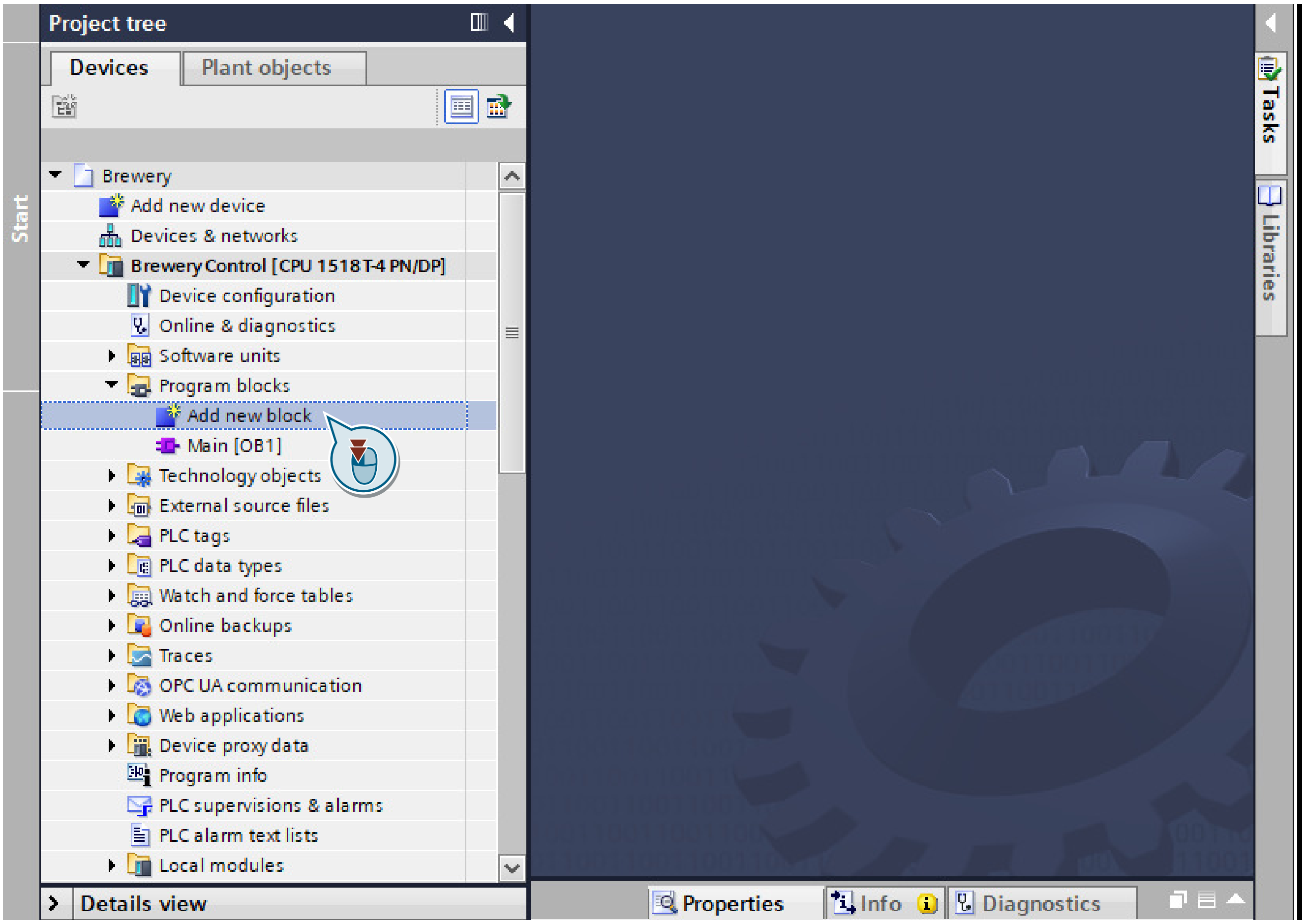
4.3.3 Creating a data block

Introduction

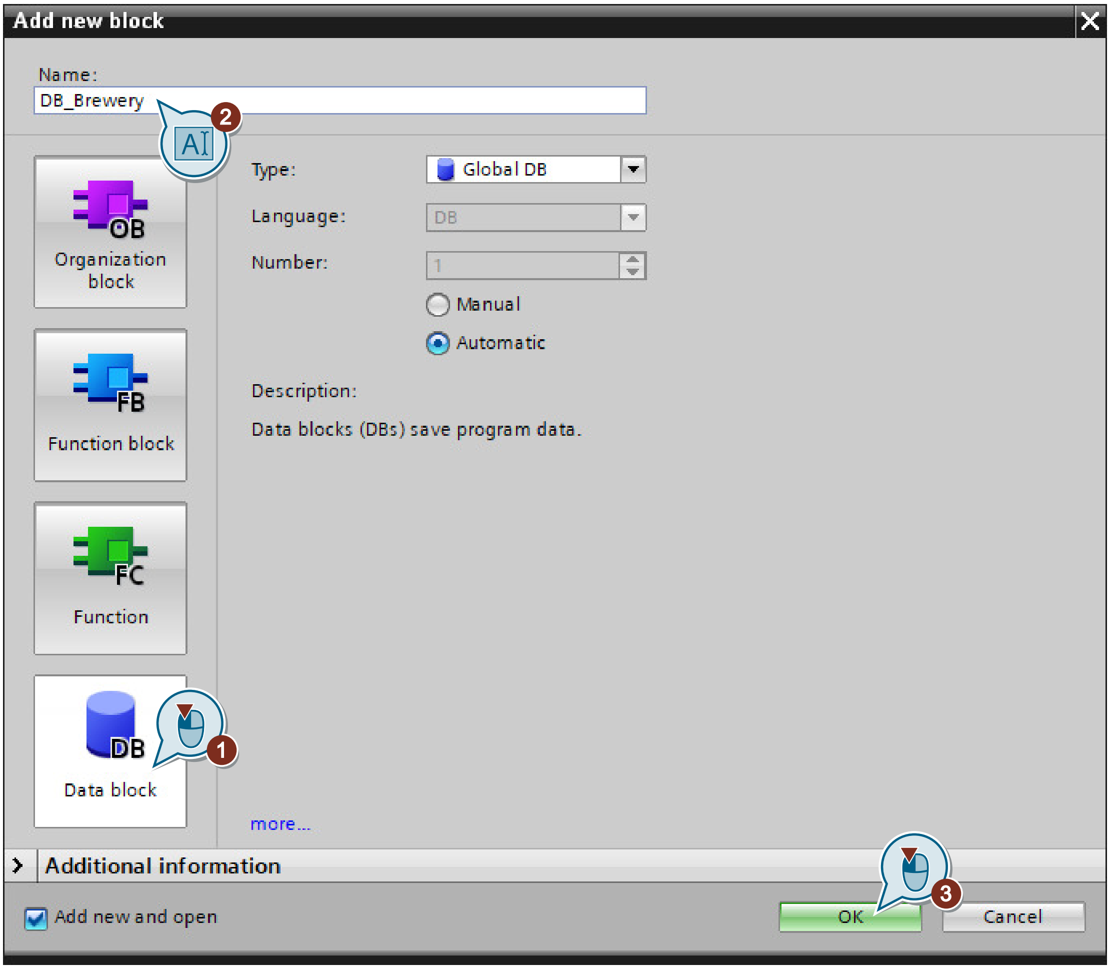
The values of the elements "Water", "Hop" and "Malt" are stored in the program block. In the following, you insert a data block and link it to the user data type "UDT_Beer".

Procedure

1. Add a new program block.



2. Rename the new data block to "DB_Brewery".



Add new block

Name: DB_Brewery

Type: Global DB

Language: DB

Number: 1

○ Manual

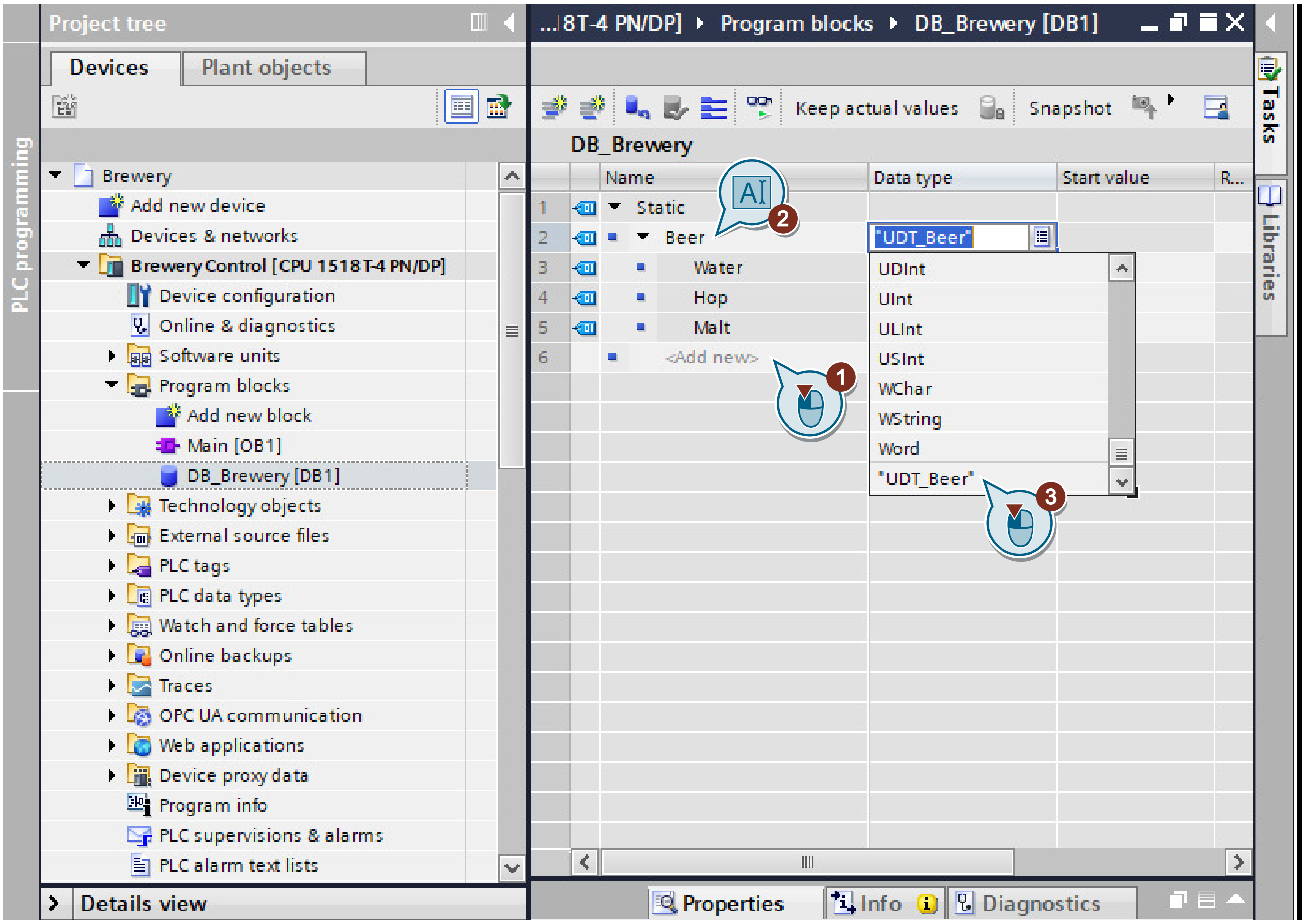
● Automatic

Description:

Data blocks (DBs) save program data.

☑ Add new and open

3. Create a new element "Beer" and use the user data type as data type. The parameters of the user data type are added automatically.



4.4 Configuring an HMI

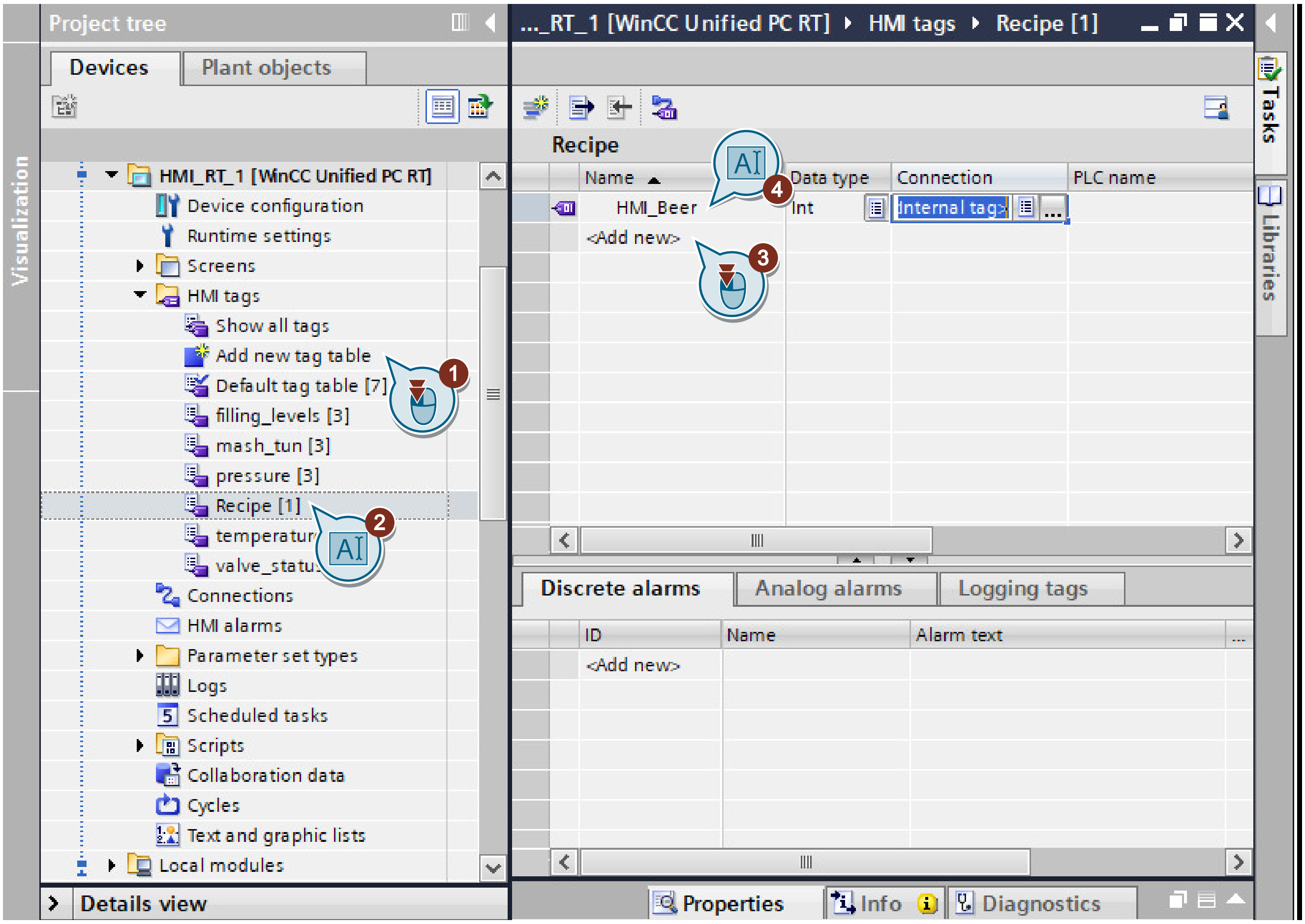
4.4.1 Link HMI connection with program block

Introduction

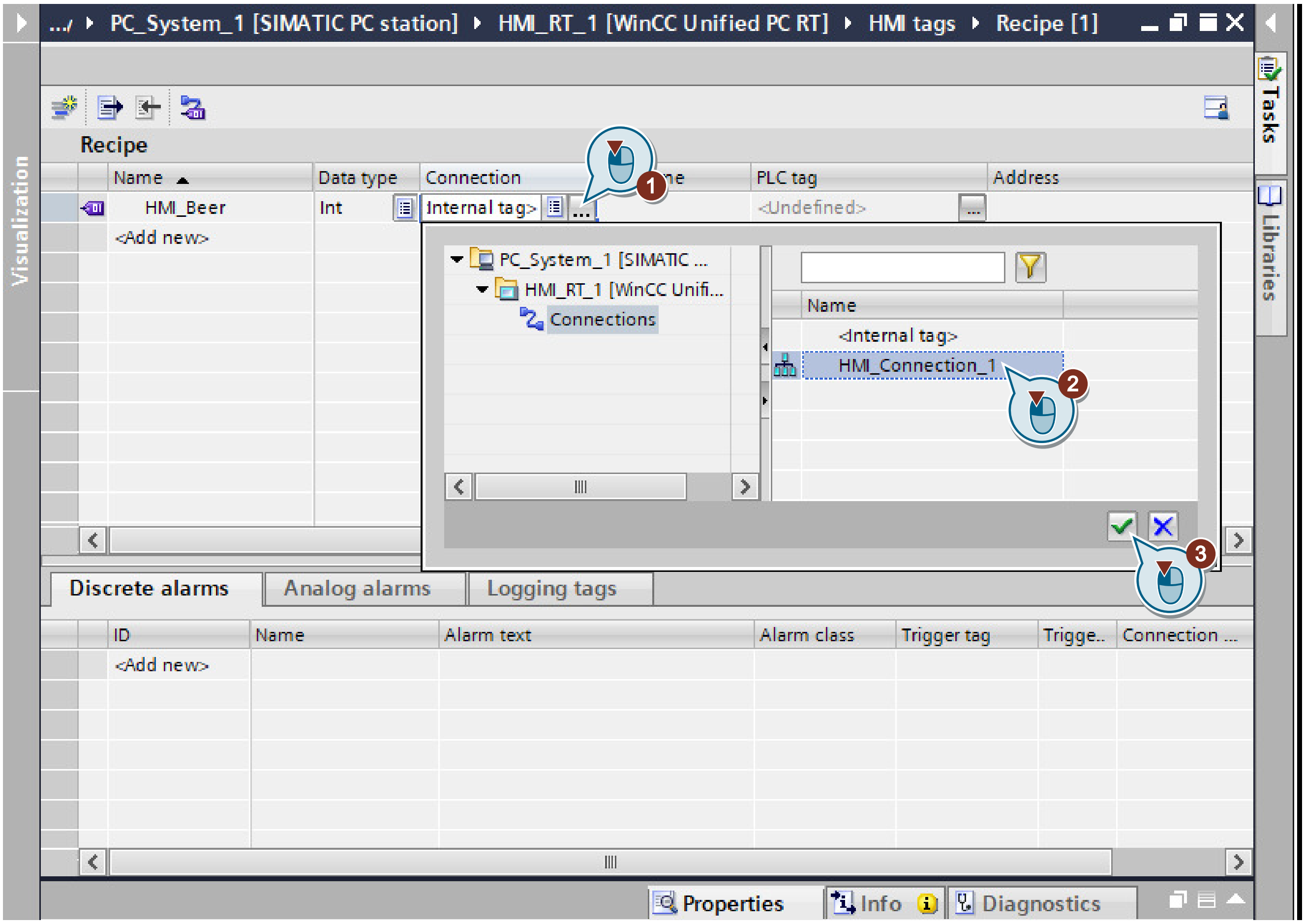
In the following, you configure the data exchange between controller and HMI device.

Link HMI connection with program block

1. Add a new tag table "Recipe" and create a parameter "HMI_Beer".



2. Select the HMI connection.

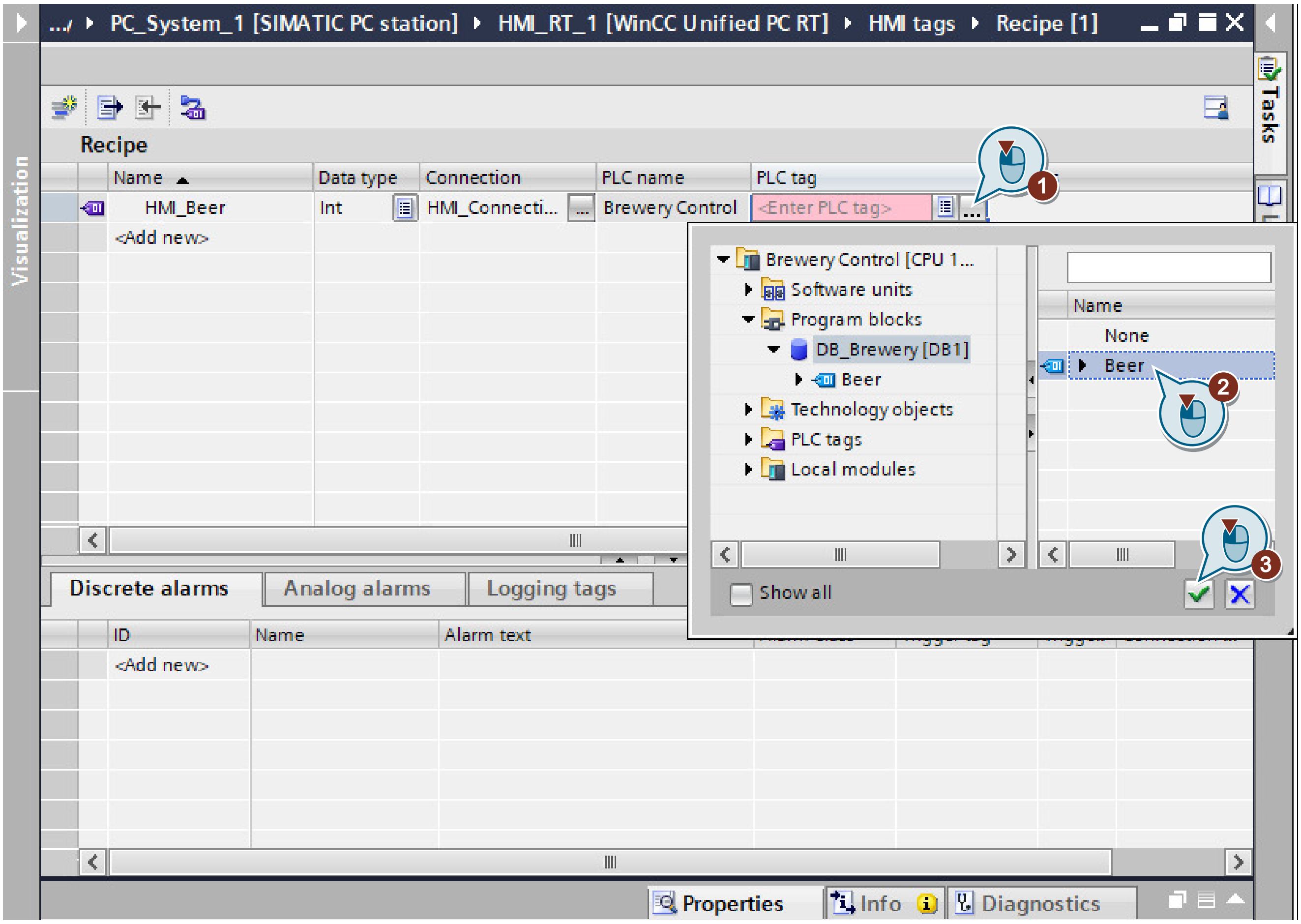


.../ ▸ PC_System_1 [SIMATIC PC station] ▸ HMI_RT_1 [WinCC Unified PC RT] ▸ HMI tags ▸ Recipe [1]
Visualization
Tasks
Libraries
Recipe

Name ▲	Data type	Connection	...ne	PLC tag	Address
HMI_Beer	Int	Internal tag>		<Undefined>	
<Add new>					

ID	Name	Alarm text	Alarm class	Trigger tag	Trigge..	Connection ...
<Add new>						

3. Select the PLC data type "Beer" previously created for the data block.


.../ ▸ PC_System_1 [SIMATIC PC station] ▸ HMI_RT_1 [WinCC Unified PC RT] ▸ HMI tags ▸ Recipe [1]
Visualization
Tasks
Recipe
Name | Data type | Connection | PLC name | PLC tag
HMI_Beer | Int | HMI_Connecti... | Brewery Control | <Enter PLC tag>
<Add new>
Brewery Control [CPU 1...
Software units
Program blocks
DB_Brewery [DB1]
Beer
Technology objects
PLC tags
Local modules
Name
None
Beer
Show all
1
2
3
Discrete alarms
Analog alarms
Logging tags
ID | Name | Alarm text
<Add new>
Properties
Info
Diagnostics

4.4.2 Creating a parameter set type and configuring a parameter control

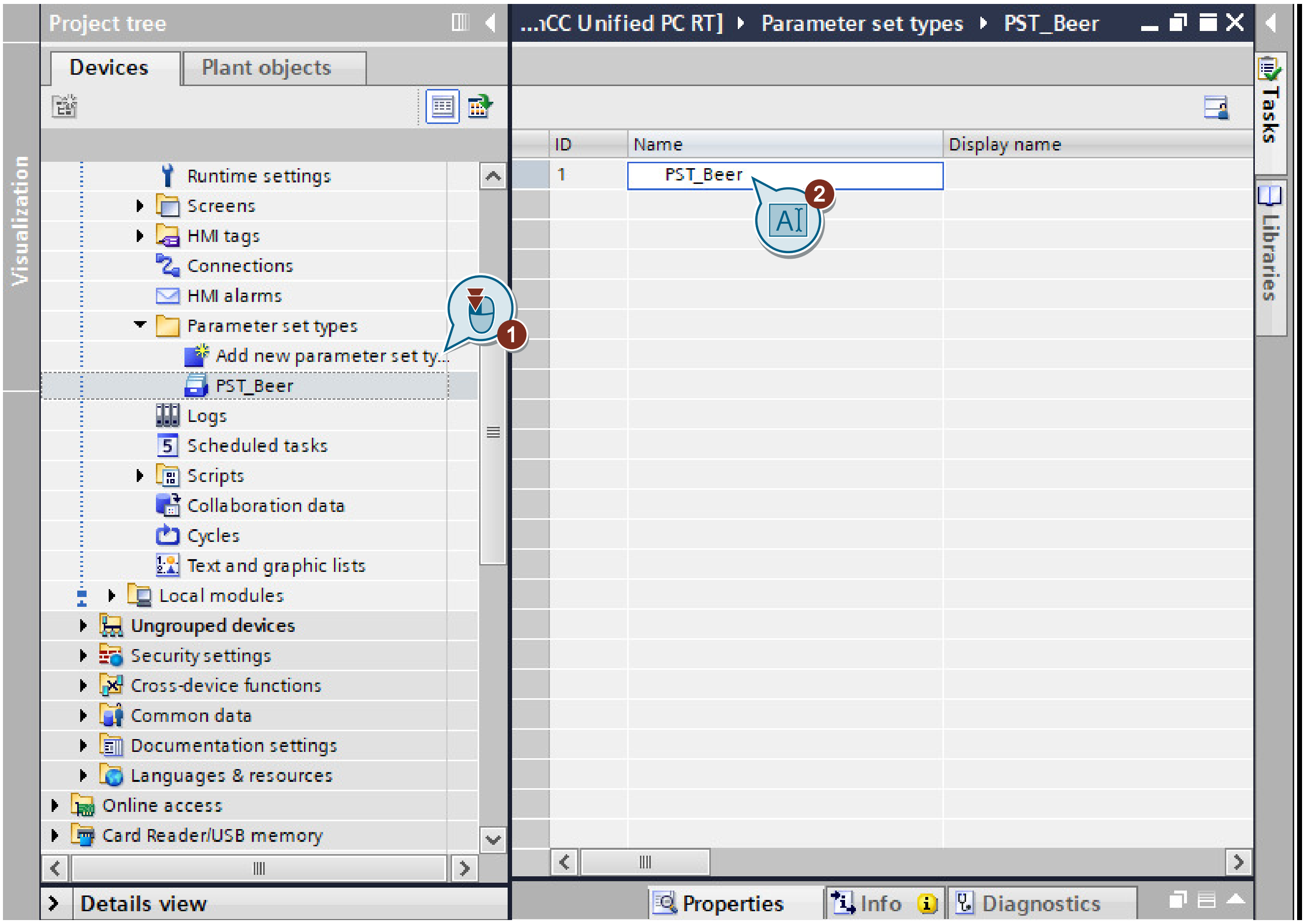
Introduction

In the following, you create a parameter set type for several beer types. The parameter set is to receive its structure through the parameter set type "PST_Beer". The parameter set type consists of the elements "water", "hop" and "malt".

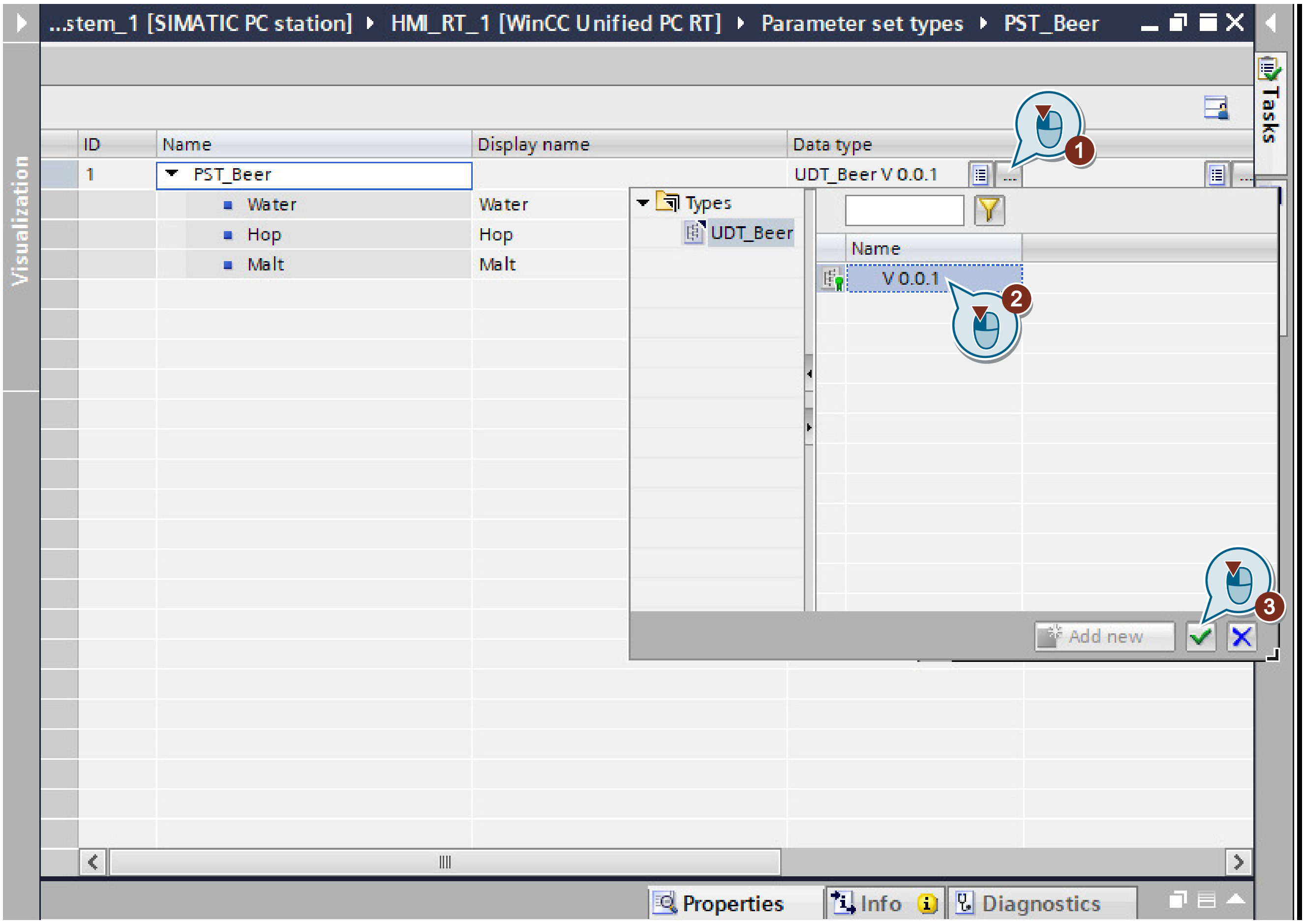
Element name	Use	Data type	Unit of measurement
water	Quantity of water	Int	l
hop	Quantity of hops	Int	kg
malt	Quantity of malt	Int	kg

Procedure

1. Add a new parameter set type and rename it "PST_Beer".



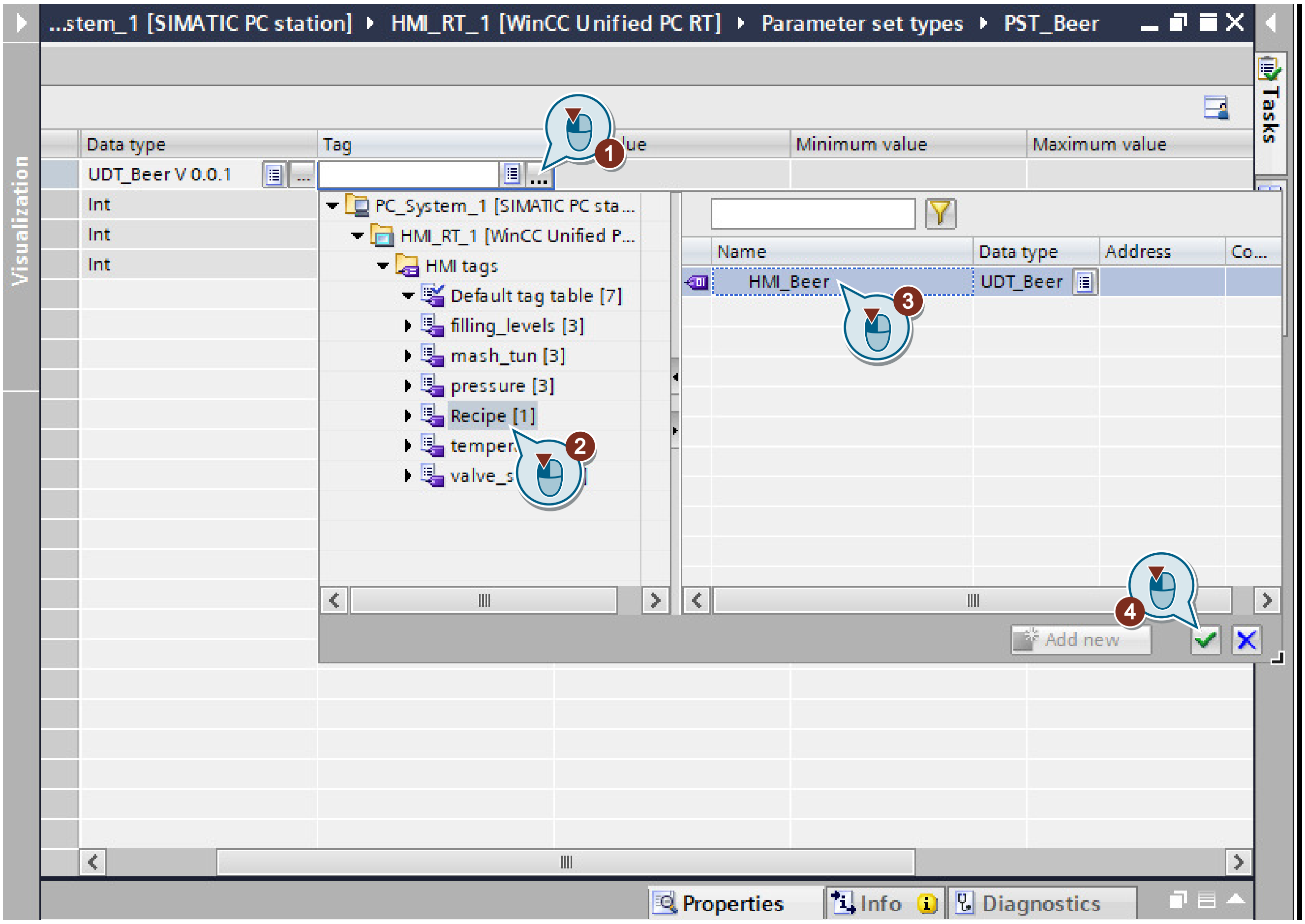
2. Select the user data type "UDT_Beer" as the data type. The structure is created automatically based on the user data type.



ID	Name	Display name	Data type
1	PST_Beer		UDT_Beer V 0.0.1
	Water	Water	
	Hop	Hop	
	Malt	Malt	

...stem_1 [SIMATIC PC station] ▸ HMI_RT_1 [WinCC Unified PC RT] ▸ Parameter set types ▸ PST_Beer
Visualization
Tasks
Types
UDT_Beer
Name
V 0.0.1
Add new
1
2
3
Properties
Info
Diagnostics

3. Select the tag "Beer".

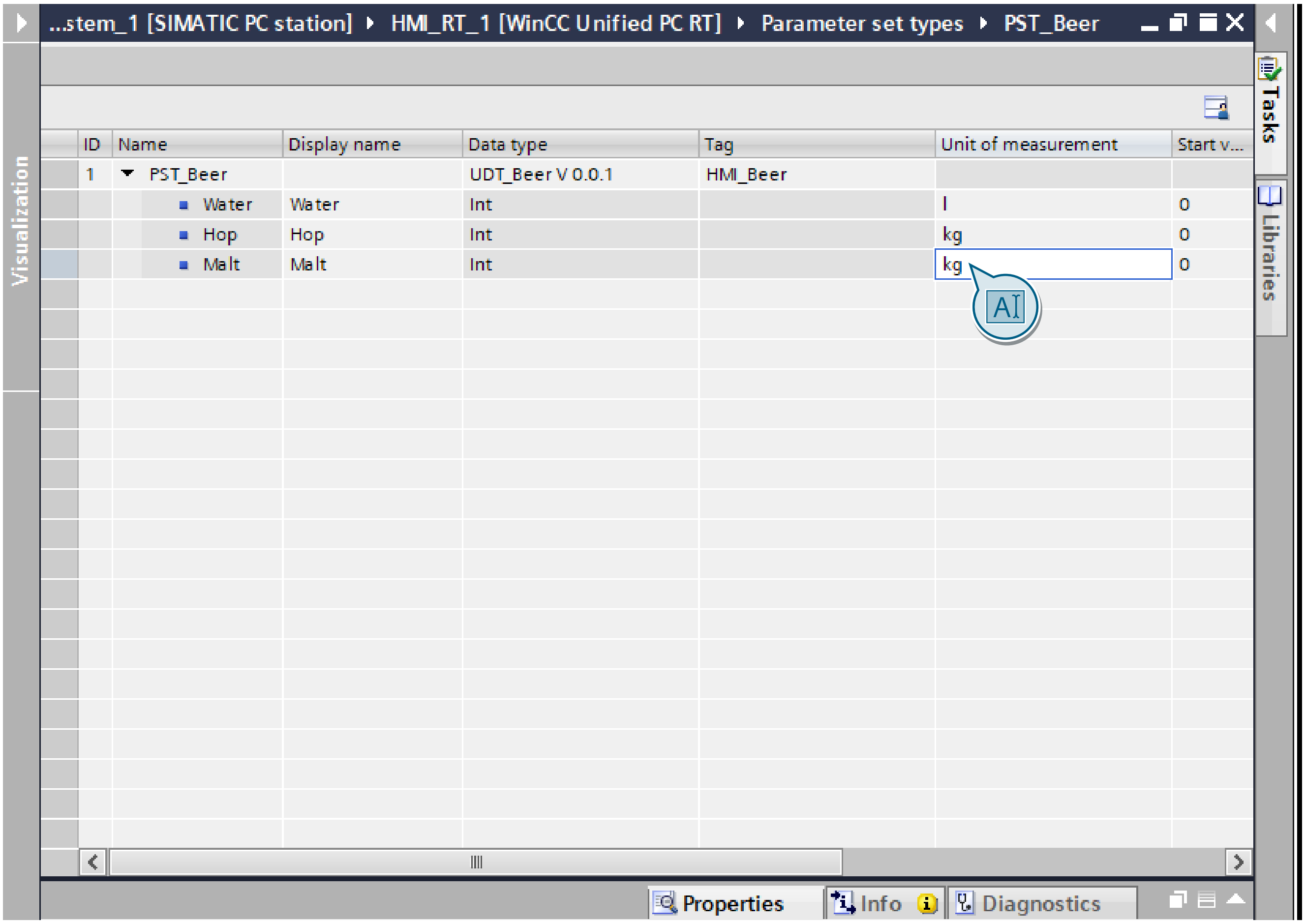


...stem_1 [SIMATIC PC station] ▸ HMI_RT_1 [WinCC Unified PC RT] ▸ Parameter set types ▸ PST_Beer

Data type	Tag	...lue	Minimum value	Maximum value
UDT_Beer V 0.0.1				
Int				
Int				
Int				

Name	Data type	Address	Co...
HMI_Beer	UDT_Beer		

4. Specify the units of measurement for the data types.



...stem_1 [SIMATIC PC station] ▸ HMI_RT_1 [WinCC Unified PC RT] ▸ Parameter set types ▸ PST_Beer
Visualization
Tasks
Libraries
Properties
Info
Diagnostics

ID	Name	Display name	Data type	Tag	Unit of measurement	Start v...
1	▼ PST_Beer		UDT_Beer V 0.0.1	HMI_Beer		
	■ Water	Water	Int		l	0
	■ Hop	Hop	Int		kg	0
	■ Malt	Malt	Int		kg	0

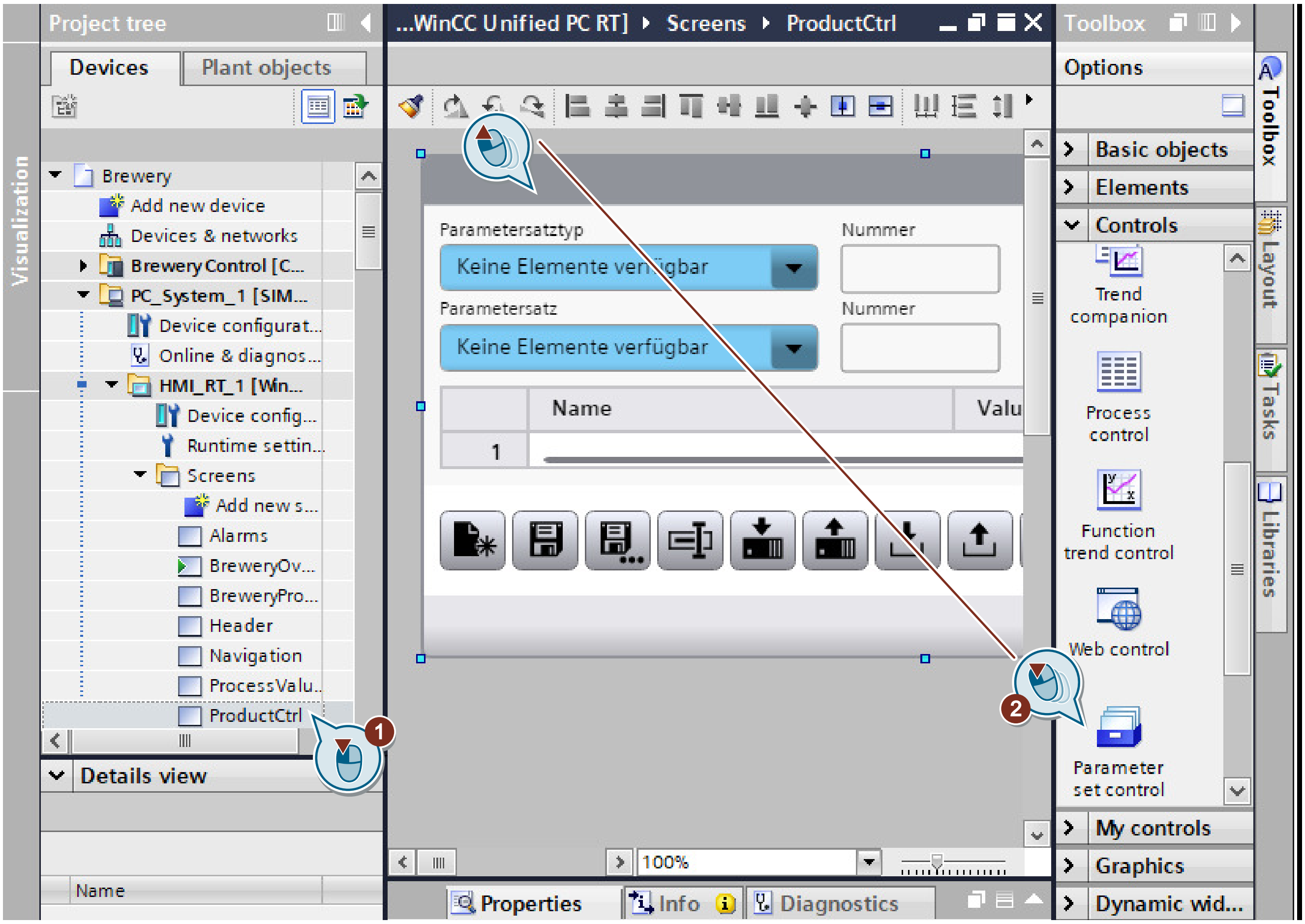
4.4.3 Inserting and configuring a parameter set control

Introduction

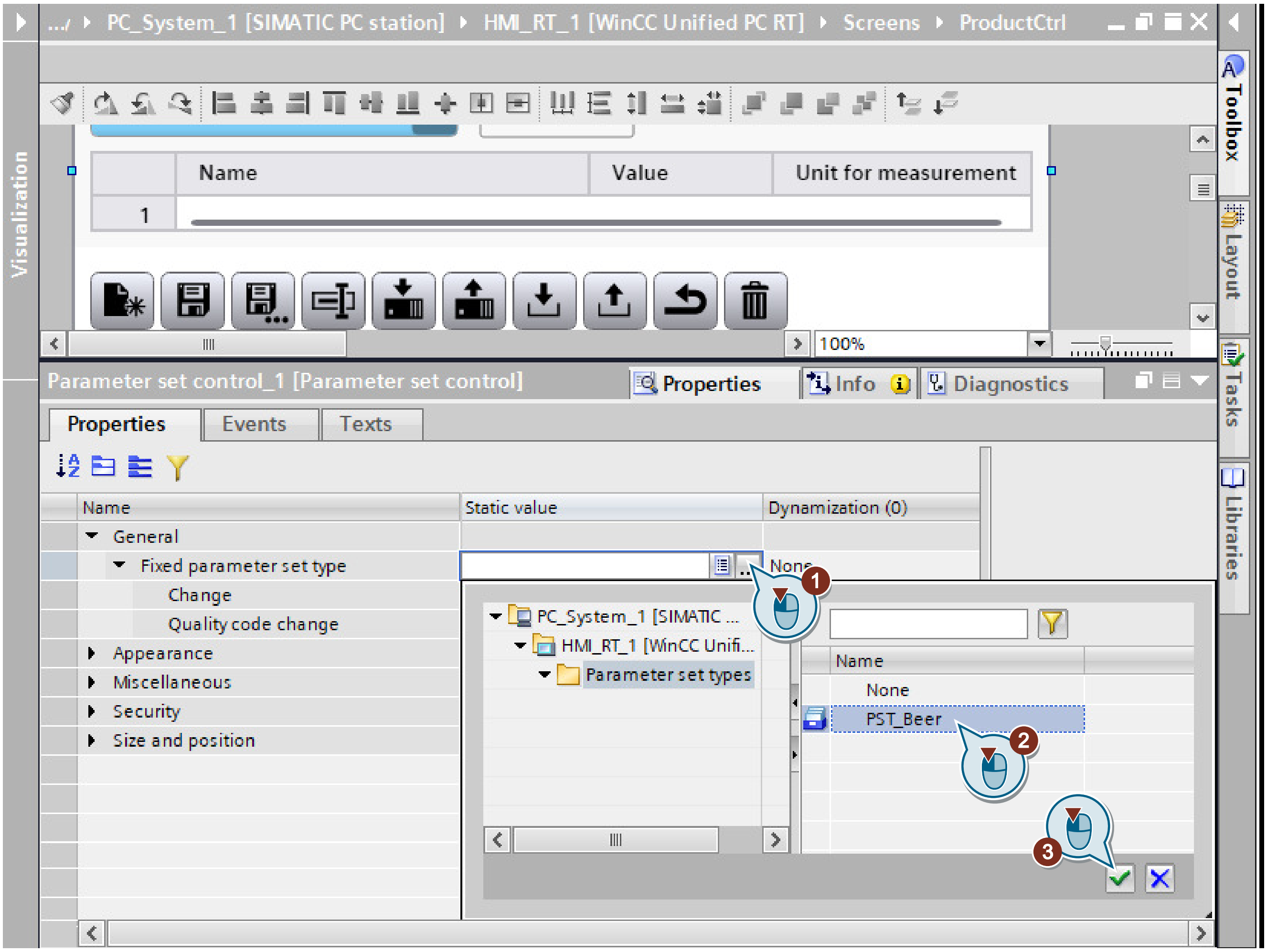
In the following you configure the parameter set control. The parameter sets are created and edited in runtime via the parameter set control.

Procedure

1. Insert the parameter set control in the HMI screen "ProductCtrl".



2. Connect the parameter set type to the parameter set control.



4.5 Setting up a connection with PLCSim

Introduction

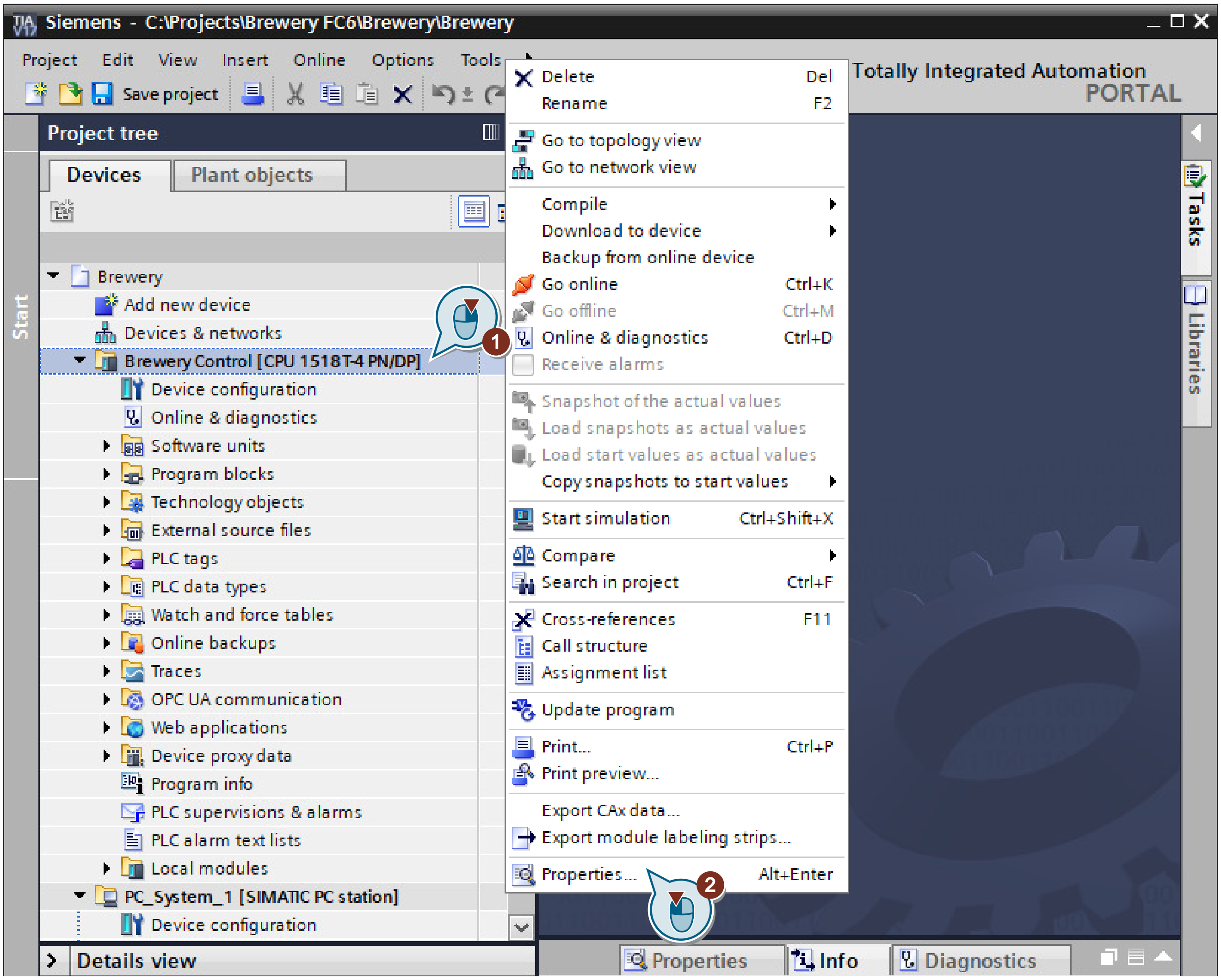
To establish the connection with PLCSim, define the access rights for the PLC before setting up the connection.

When you start the simulation, the PLCSim control window opens. You can control the simulation using the Run and Stop buttons. If you follow the explained procedure, no operation of the control window is required.

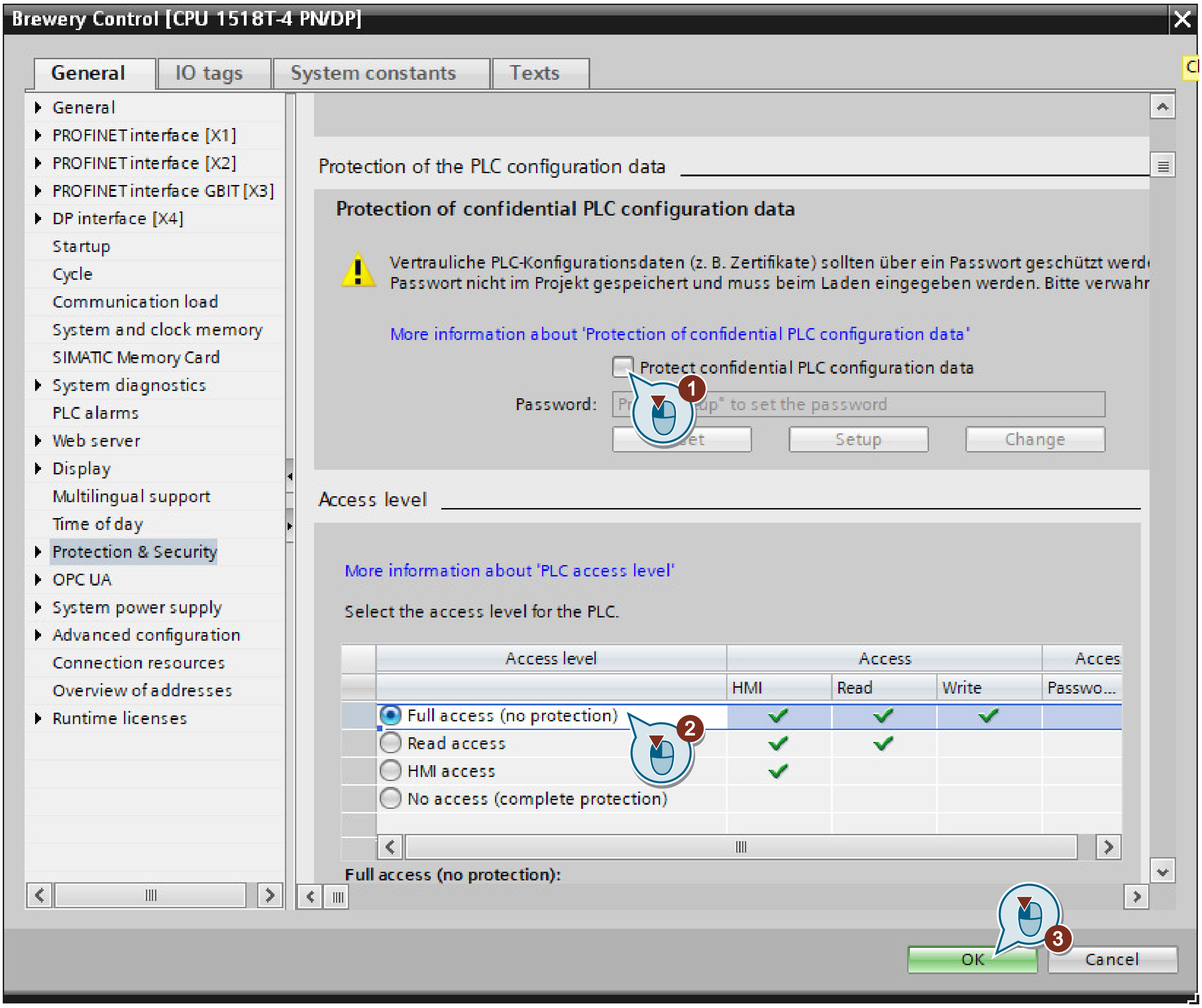
During the procedure, deactivate functions for the protection of confidential data and restrictions on operation. These functions are deactivated to simplify the simulation. In the real process environment, this access must be restricted. With automation systems, the unconditional maintenance of operational safety and the protection of life and limb are paramount. The decisive requirement for this is maintaining the availability of the system and thus unrestricted control over the process.

Procedure

1. Open the properties of the "Brewery Control" controller.



2. Deactivate the function for protecting confidential PLC configuration data and select the "Full access" access level for the controller.



Protection of the PLC configuration data

Protection of confidential PLC configuration data

Vertrauliche PLC-Konfigurationsdaten (z. B. Zertifikate) sollten über ein Passwort geschützt werd
Passwort nicht im Projekt gespeichert und muss beim Laden eingegeben werden. Bitte verwahr

More information about 'Protection of confidential PLC configuration data'

□ Protect confidential PLC configuration data

Password: Pr up" to set the password

Access level

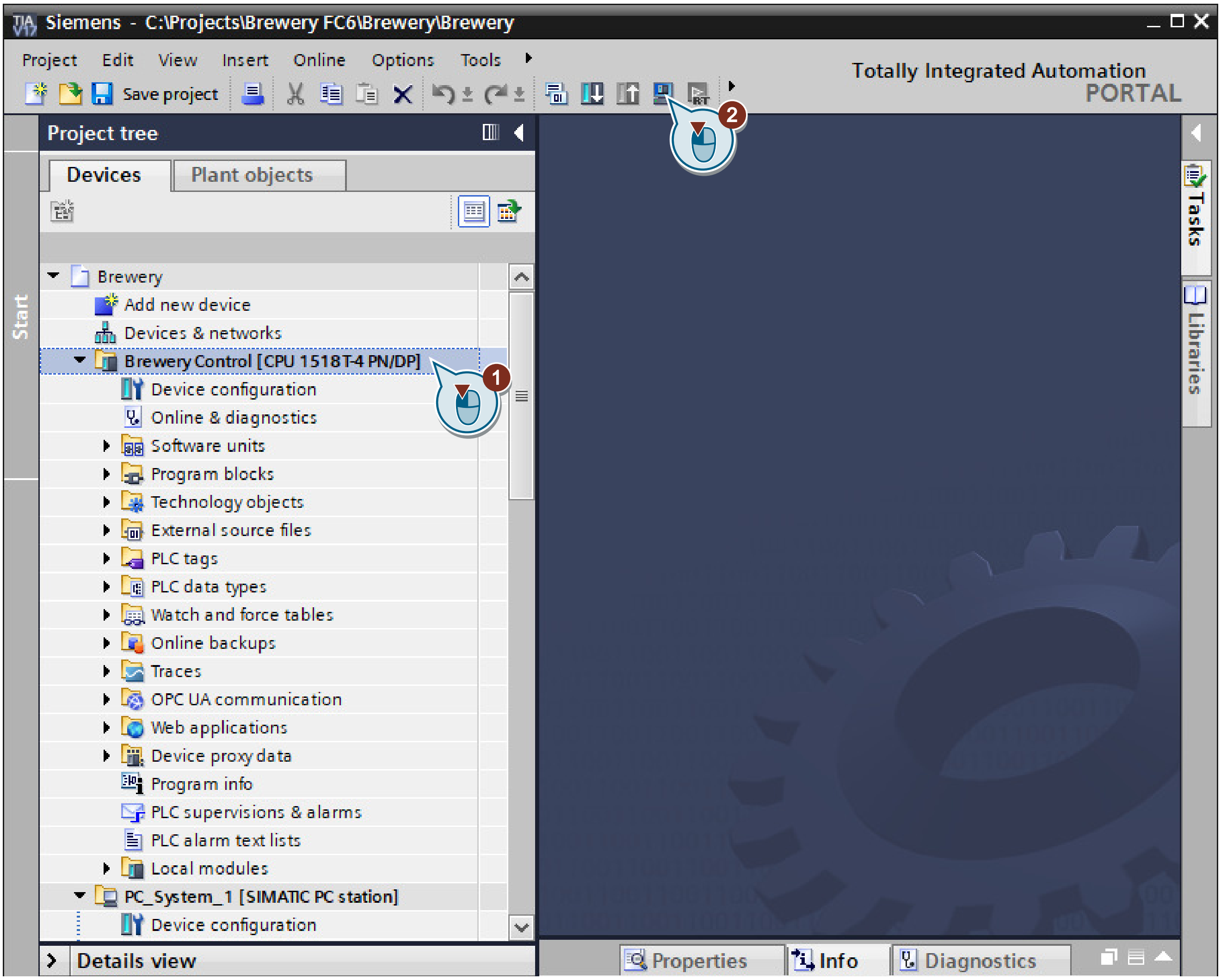
More information about 'PLC access level'

Select the access level for the PLC.

Access level	Access			Acces
	HMI	Read	Write	Passwo...
● Full access (no protection)	✓	✓	✓	
○ Read access	✓	✓		
○ HMI access	✓			
○ No access (complete protection)				

Full access (no protection):

3. Start the simulation.



Siemens - C:\Projects\Brewery FC6\Brewery\Brewery

Project Edit View Insert Online Options Tools

Save project

Totally Integrated Automation
PORTAL

Project tree

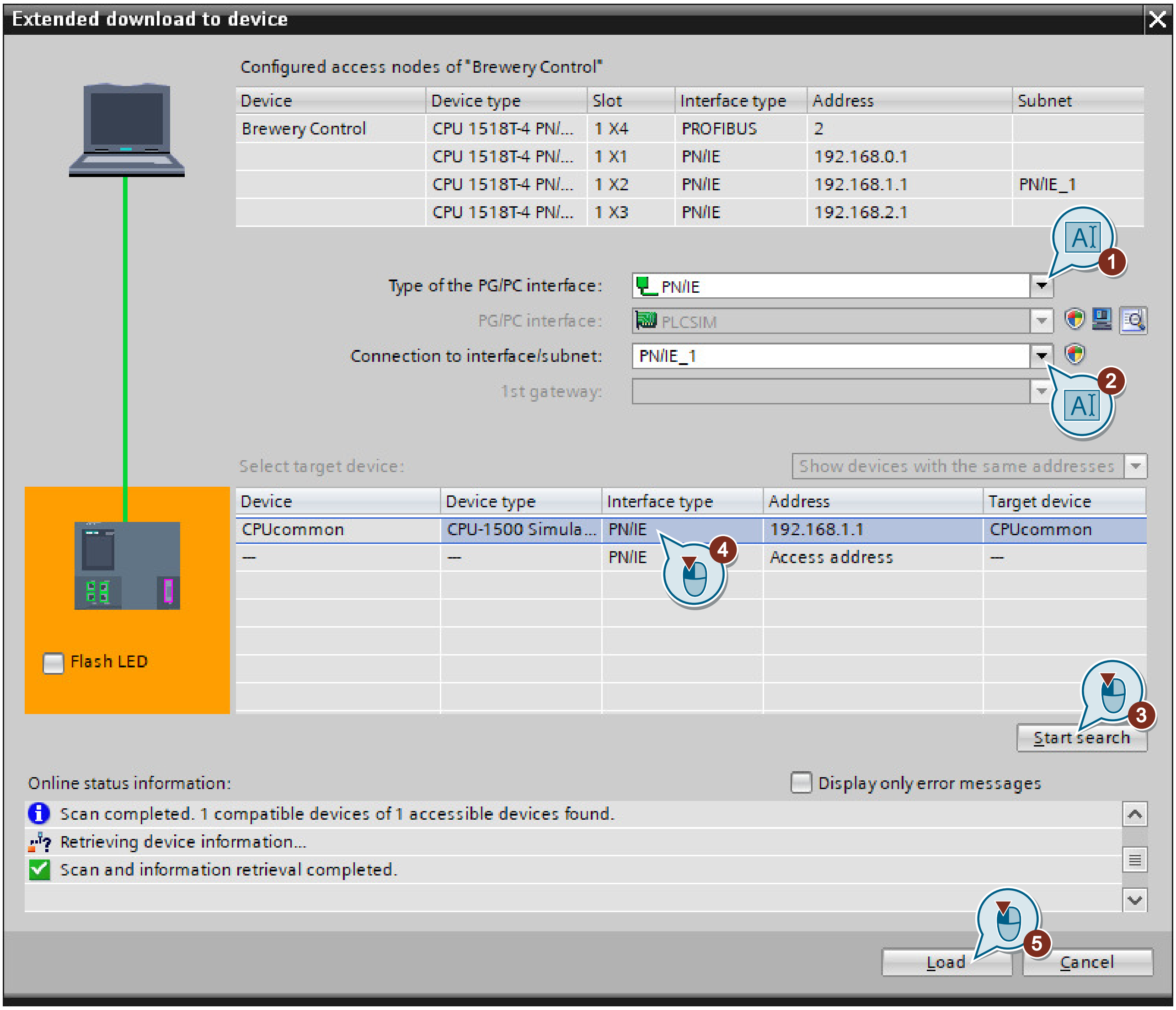
Devices | Plant objects

- Brewery
  - Add new device
  - Devices & networks
  - Brewery Control [CPU 1518T-4 PN/DP]
    - Device configuration
    - Online & diagnostics
    - Software units
    - Program blocks
    - Technology objects
    - External source files
    - PLC tags
    - PLC data types
    - Watch and force tables
    - Online backups
    - Traces
    - OPC UA communication
    - Web applications
    - Device proxy data
    - Program info
    - PLC supervisions & alarms
    - PLC alarm text lists
    - Local modules
  - PC_System_1 [SIMATIC PC station]
    - Device configuration

Details view

Properties | Info | Diagnostics

4. Select your target device.



Extended download to device

Configured access nodes of "Brewery Control"

Device	Device type	Slot	Interface type	Address	Subnet
Brewery Control	CPU 1518T-4 PN/...	1 X4	PROFIBUS	2	
	CPU 1518T-4 PN/...	1 X1	PN/IE	192.168.0.1	
	CPU 1518T-4 PN/...	1 X2	PN/IE	192.168.1.1	PN/IE_1
	CPU 1518T-4 PN/...	1 X3	PN/IE	192.168.2.1	

Type of the PG/PC interface: PN/IE

PG/PC interface: PLCSIM

Connection to interface/subnet: PN/IE_1

1st gateway:

Select target device: Show devices with the same addresses

Device	Device type	Interface type	Address	Target device
CPUcommon	CPU-1500 Simula...	PN/IE	192.168.1.1	CPUcommon
—	—	PN/IE	Access address	—

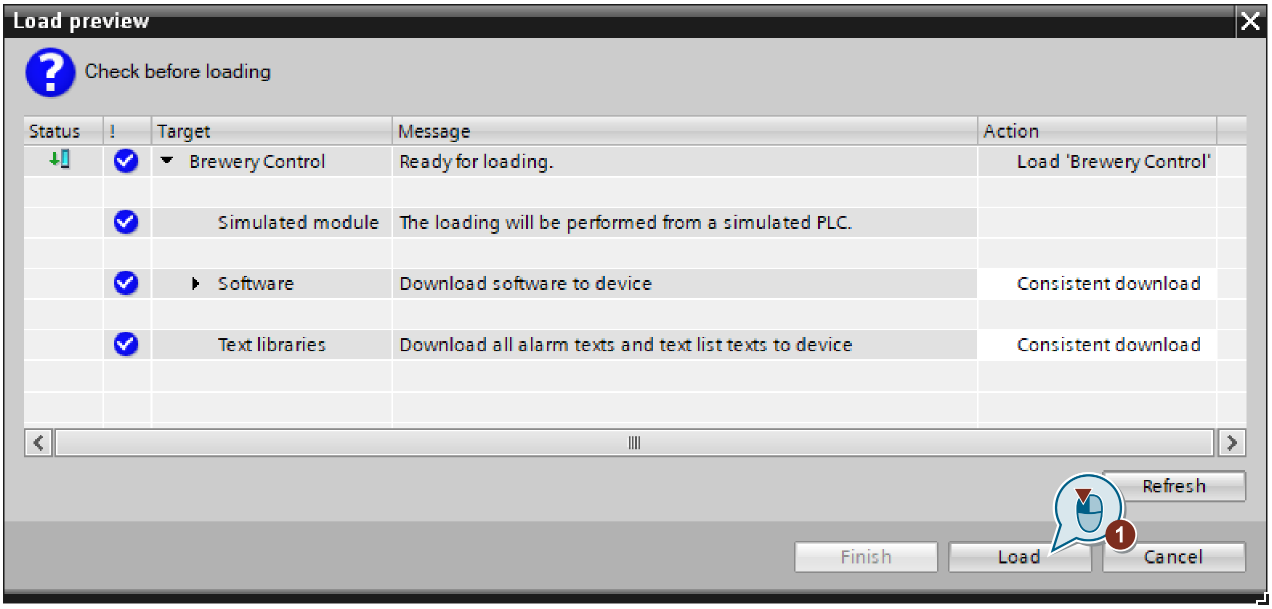
□ Flash LED

Online status information:

□ Display only error messages

Scan completed. 1 compatible devices of 1 accessible devices found.
Retrieving device information...
Scan and information retrieval completed.

5. Check the settings and load the data.

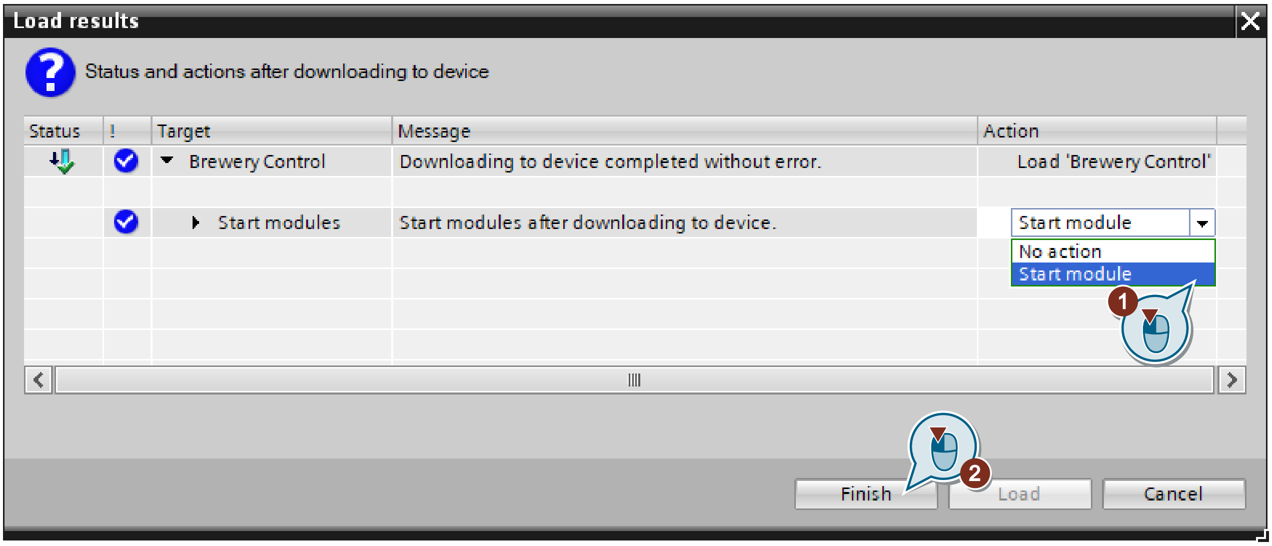


Load preview

Check before loading

Status	!	Target	Message	Action
	✓	Brewery Control	Ready for loading.	Load 'Brewery Control'
	✓	Simulated module	The loading will be performed from a simulated PLC.	
	✓	Software	Download software to device	Consistent download
	✓	Text libraries	Download all alarm texts and text list texts to device	Consistent download

6. Complete the loading process.

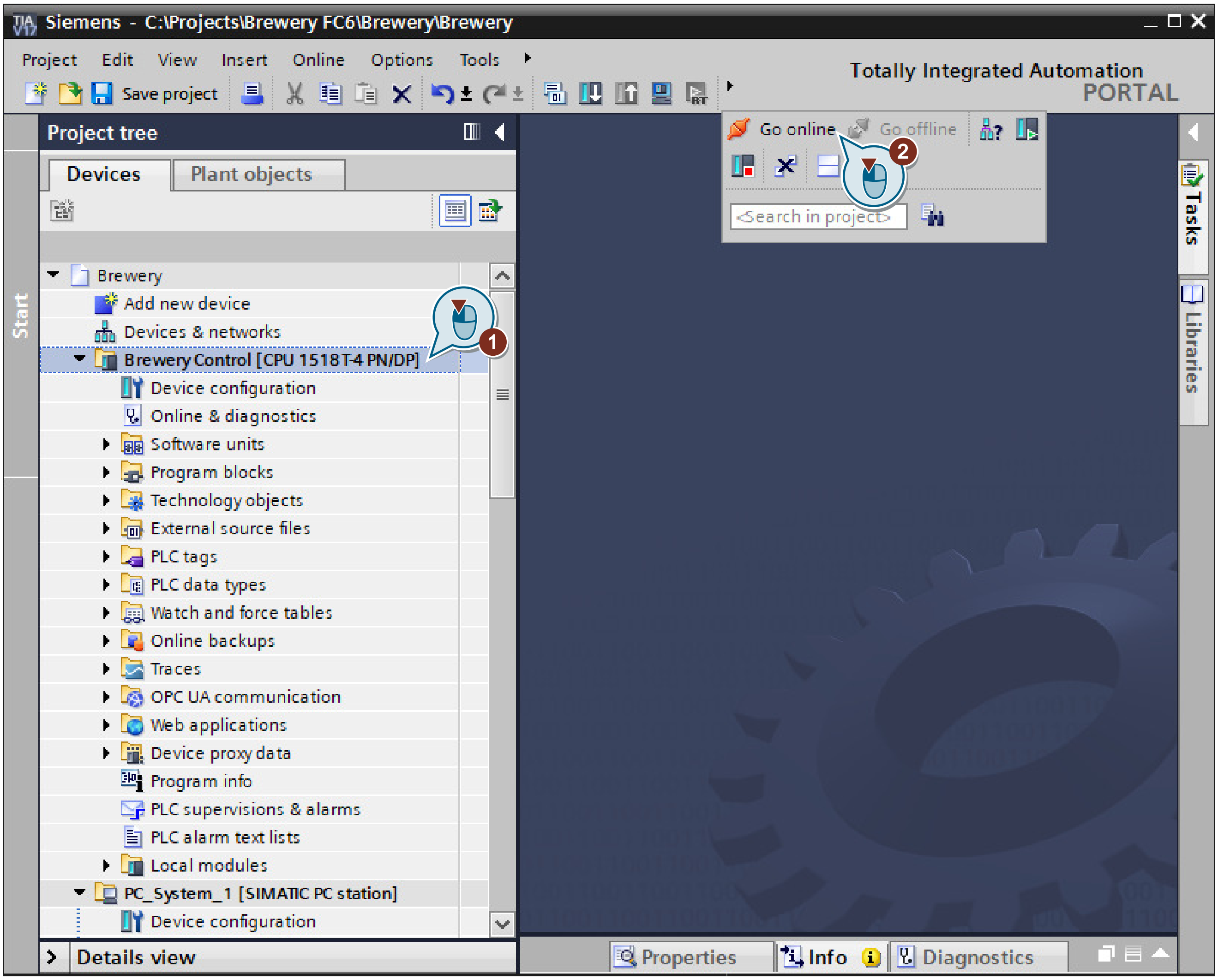


Load results

Status and actions after downloading to device

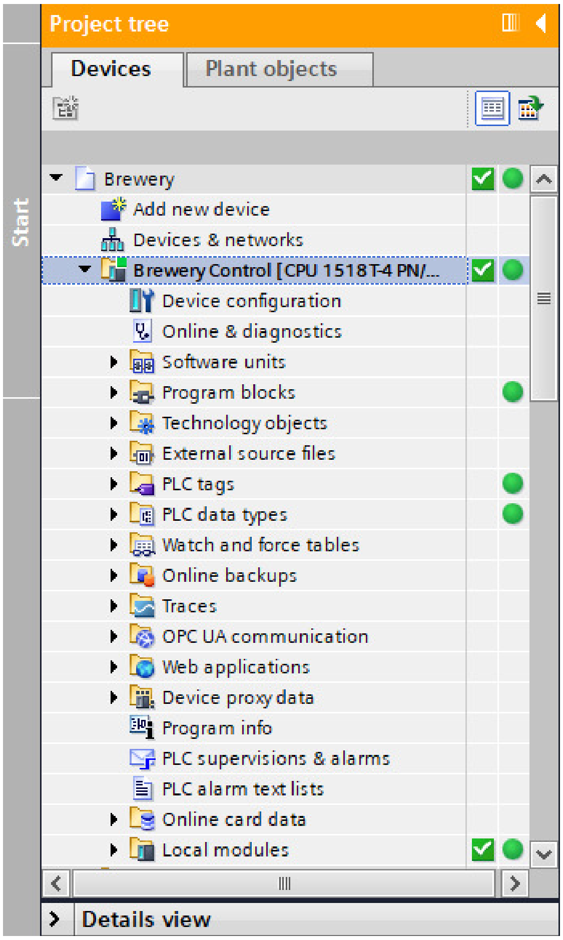
Status	!	Target	Message	Action
	✓	Brewery Control	Downloading to device completed without error.	Load 'Brewery Control'
	✓	Start modules	Start modules after downloading to device.	Start module

7. Establish an "online connection".


Siemens - C:\Projects\Brewery FC6\Brewery\Brewery
Project Edit View Insert Online Options Tools
Save project
Totally Integrated Automation
PORTAL
Project tree
Devices
Plant objects
Go online
Go offline
<Search in project>
Brewery
Add new device
Devices & networks
BreweryControl [CPU 1518T-4 PN/DP]
Device configuration
Online & diagnostics
Software units
Program blocks
Technology objects
External source files
PLC tags
PLC data types
Watch and force tables
Online backups
Traces
OPC UA communication
Web applications
Device proxy data
Program info
PLC supervisions & alarms
PLC alarm text lists
Local modules
PC_System_1 [SIMATIC PC station]
Device configuration
Details view
Properties
Info
Diagnostics
Start
Tasks
Libraries
1
2

Result

The connection has been established successfully.



Configuring user administration 5

5.1 User administration in WinCC

Basics

WinCC allows you to restrict security-relevant operations to specific user groups and thereby protect Runtime data and functions against unauthorized access.

- Set up users and assign authorizations to these users. Configure the authorizations required for operation of safety-related objects.
- In Runtime, users must log on with their user name and password. The requested action is executed provided the user has the required authorization.

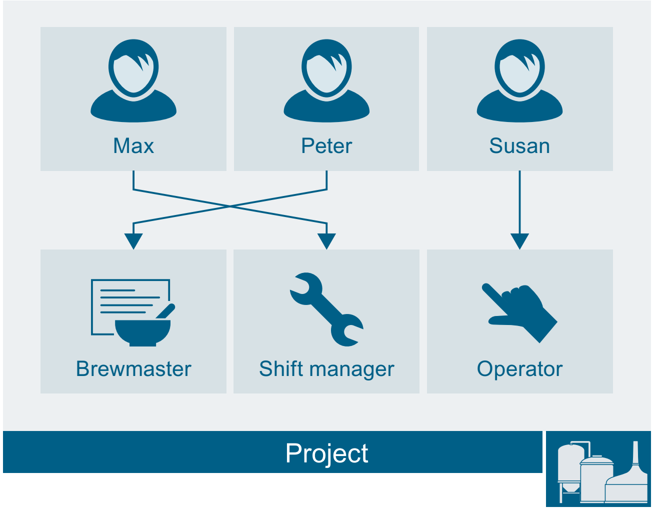
When a user is assigned a role, she/he is given the rights associated with it. Roles with associated rights may look like the following for a brewing process:

	Brewmaster	Shift manager	Operator
May monitor the process	x	x	x
May operate the process	x	x	
May edit the recipes	x		

User management is either global or local.

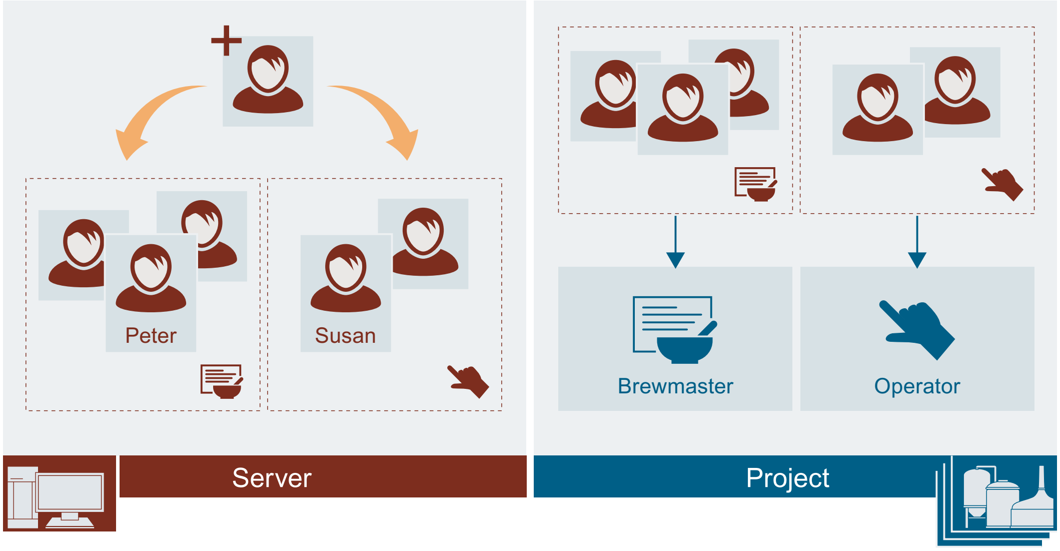
Global and local user management

Local user management is performed in the project. The users are created by the engineer and exist only in this project. The engineer defines the rights of the individual roles and assigns users their roles.



If a new user is to be added or removed subsequently, the project must be changed. Local user management is useful when only one project is created and managed. For example, a single brewery.

In global user management, users are decoupled from the role in the project. Users are managed centrally on an external server. Users are created on the external server and organized into groups that correspond to their subsequent use. The engineer does not create users in the project. The engineer imports the groups from the external server and assigns the corresponding roles to the groups.



To add users later, they are created on the external server and assigned to an existing group. This automatically gives them the role assigned to the group in the respective project. No change to the project is required.

The global user management is particularly suitable if multiple projects are to be maintained centrally. This is the case for example, when several different breweries are managed by the same company, or when new employees are added frequently.

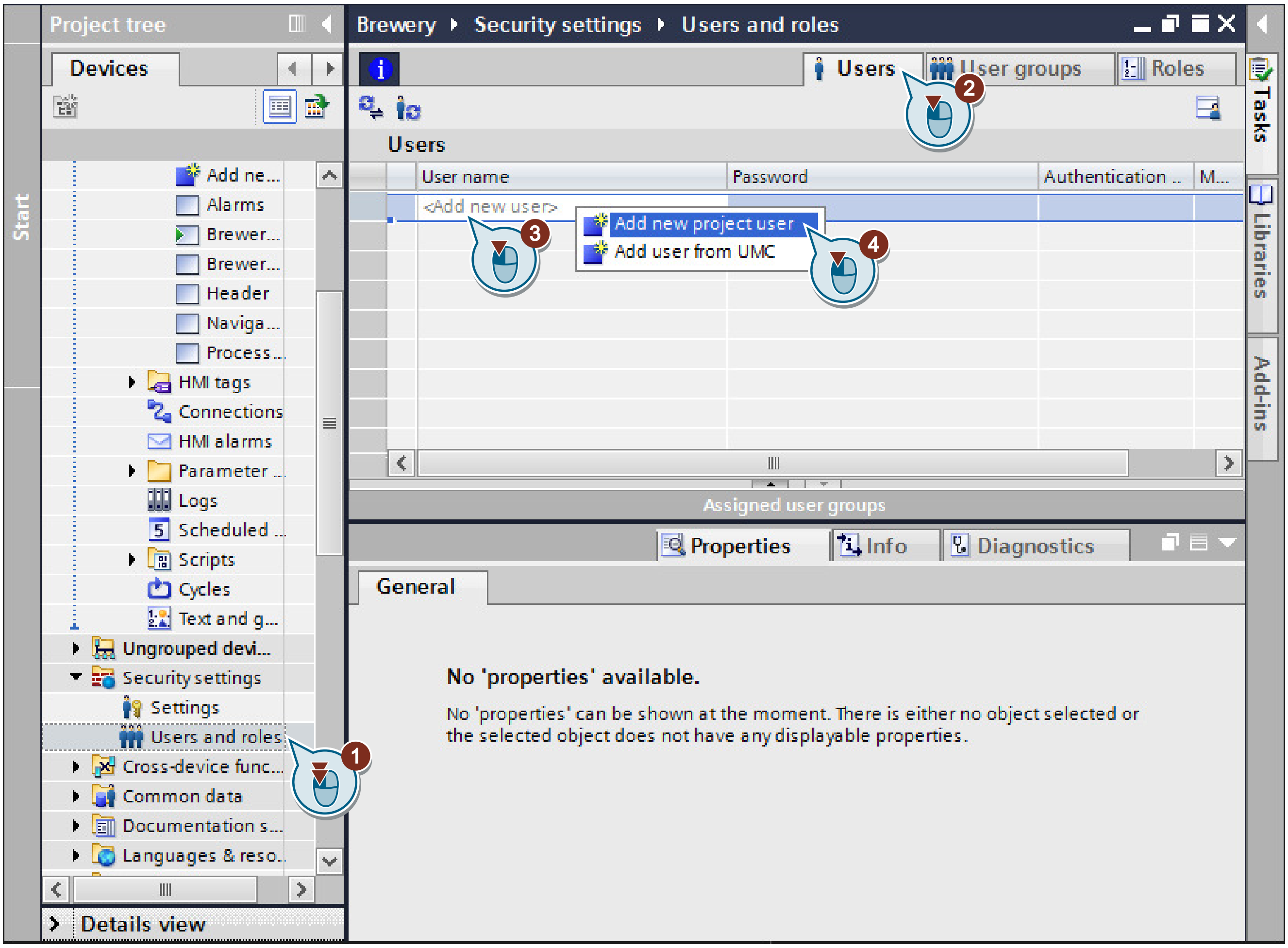
5.2 Configuring local user administration

Introduction

In the following, you will learn how to create a local user management.

Creating local users

1. Create a new local user.



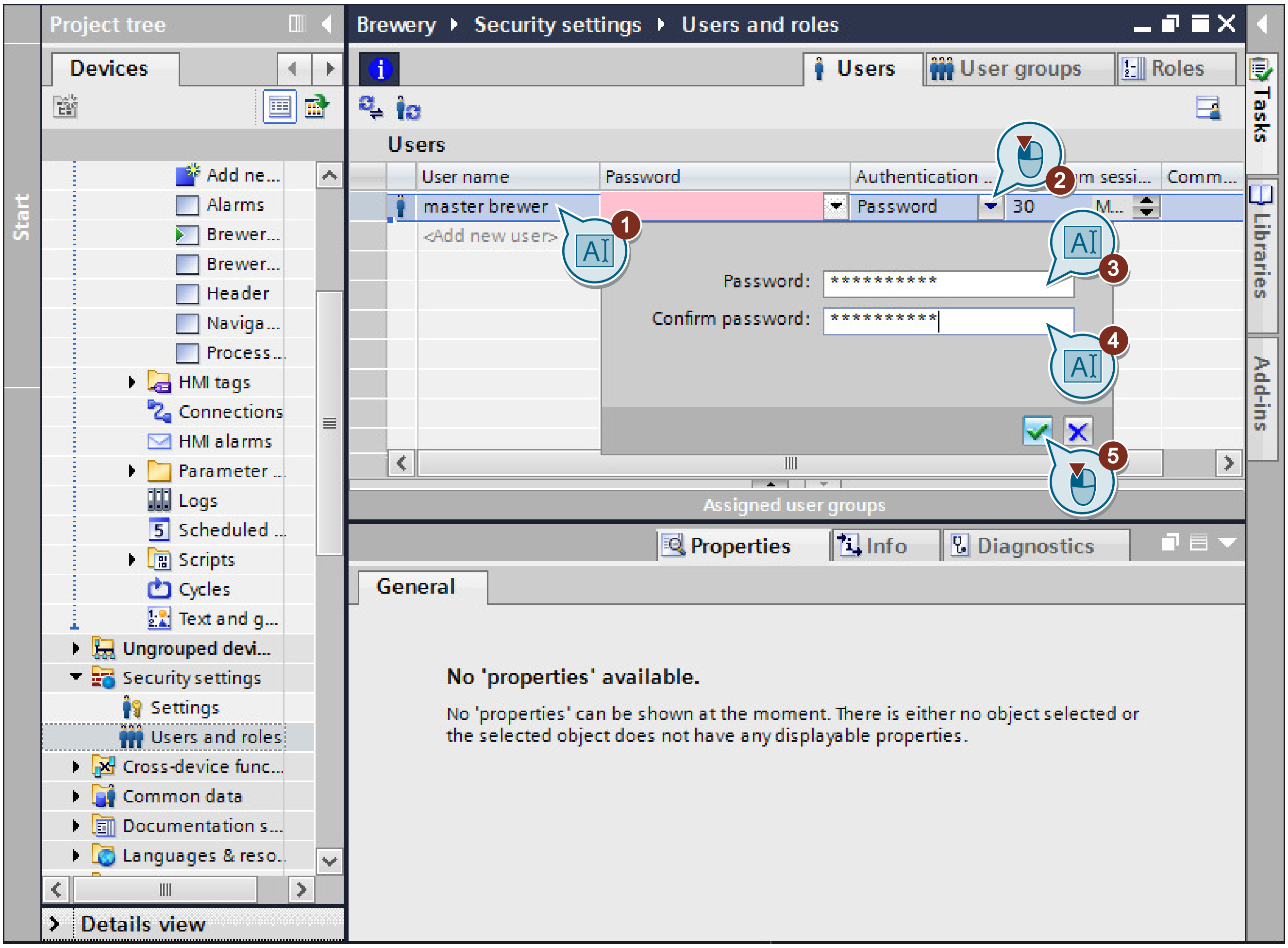
No 'properties' available.

No 'properties' can be shown at the moment. There is either no object selected or the selected object does not have any displayable properties.

2. Name the new local user and define a password.

Note

User names may not contain white spaces



Users

User name	Password	Authentication ...	...m sessi...	Comm...
master brewer		Password	30	M...
<Add new user>				

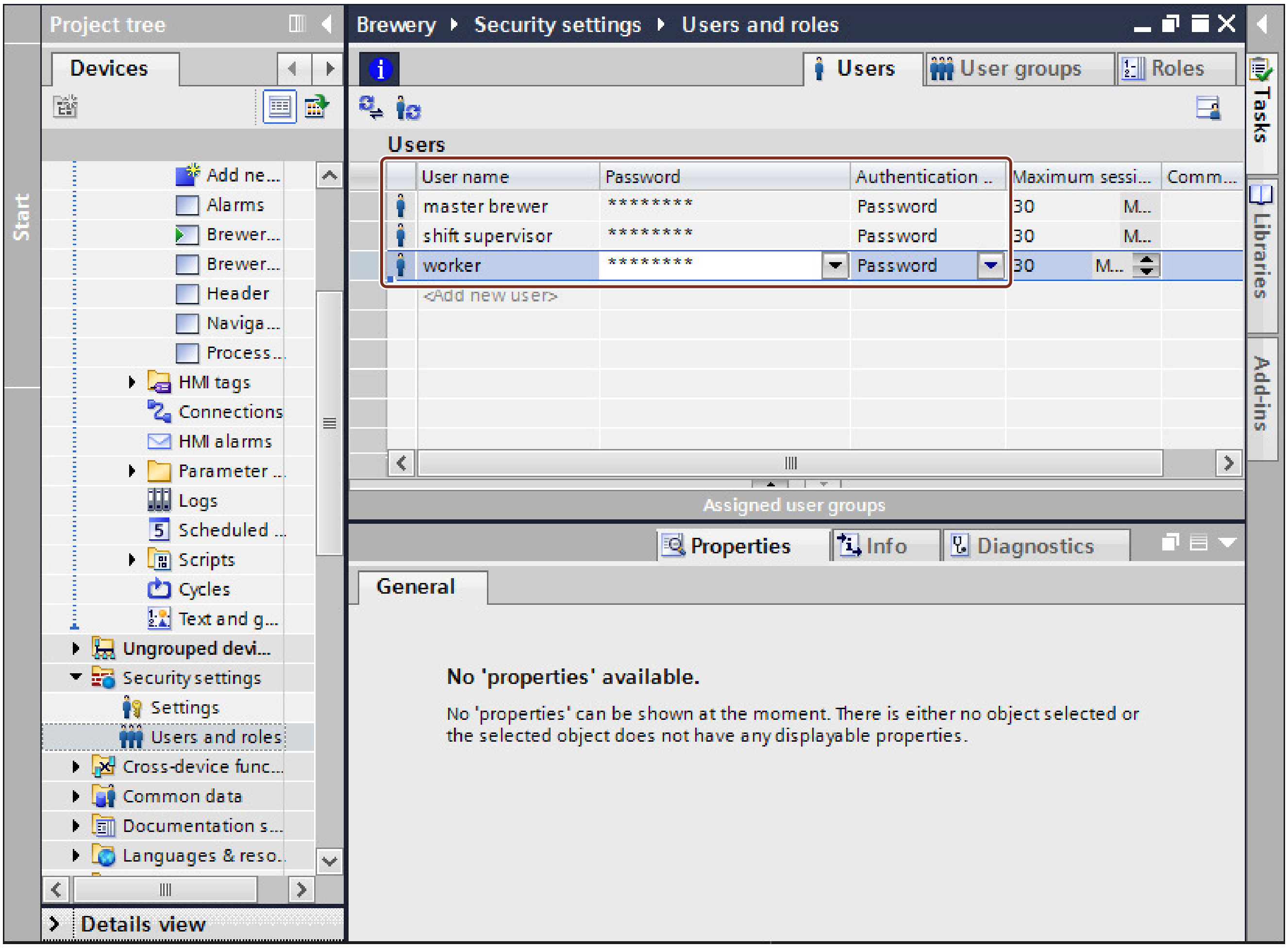
Password: **********

Confirm password: **********

No 'properties' available.

No 'properties' can be shown at the moment. There is either no object selected or the selected object does not have any displayable properties.

3. Create two more users and define a password for each user.



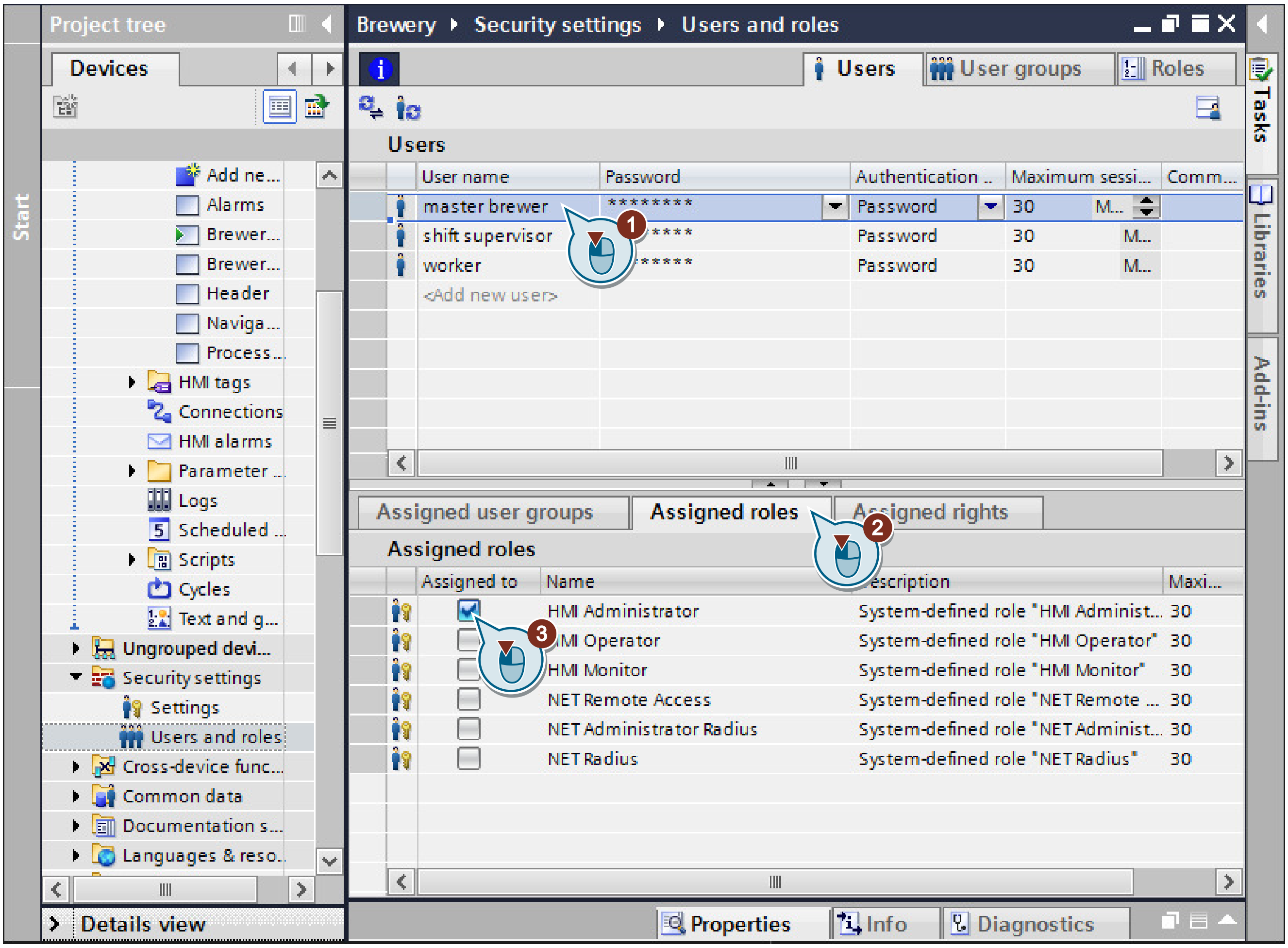
Users

	User name	Password	Authentication ...	Maximum sessi...	Comm...
	master brewer	********	Password	30 M...	
	shift supervisor	********	Password	30 M...	
	worker	********	Password	30 M...	
	<Add new user>				

No 'properties' available.

No 'properties' can be shown at the moment. There is either no object selected or the selected object does not have any displayable properties.

4. Assign the role "masterbrewer" to the user "HMI Administrator".



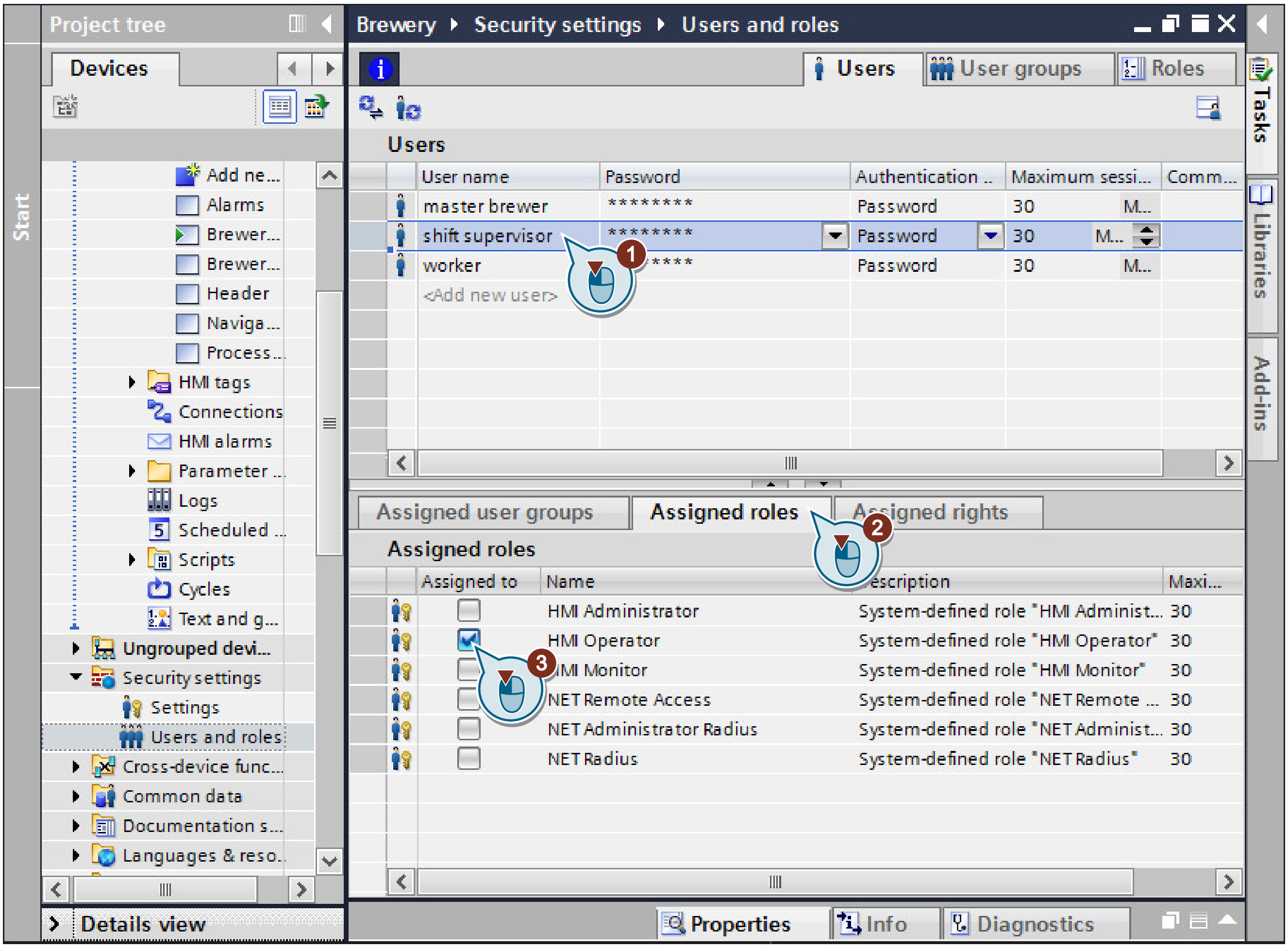
Users

User name	Password	Authentication ...	Maximum sessi...	Comm...
master brewer	********	Password	30 M...	
shift supervisor	****	Password	30 M...	
worker	******	Password	30 M...	
<Add new user>				

Assigned roles

Assigned to	Name	Description	Maxi...
☑	HMI Administrator	System-defined role "HMI Administ...	30
☐	MI Operator	System-defined role "HMI Operator"	30
☐	HMI Monitor	System-defined role "HMI Monitor"	30
☐	NET Remote Access	System-defined role "NET Remote ...	30
☐	NET Administrator Radius	System-defined role "NET Administ...	30
☐	NET Radius	System-defined role "NET Radius"	30

5. Assign the role "shiftsupervisor" to the user "HMI Operator".



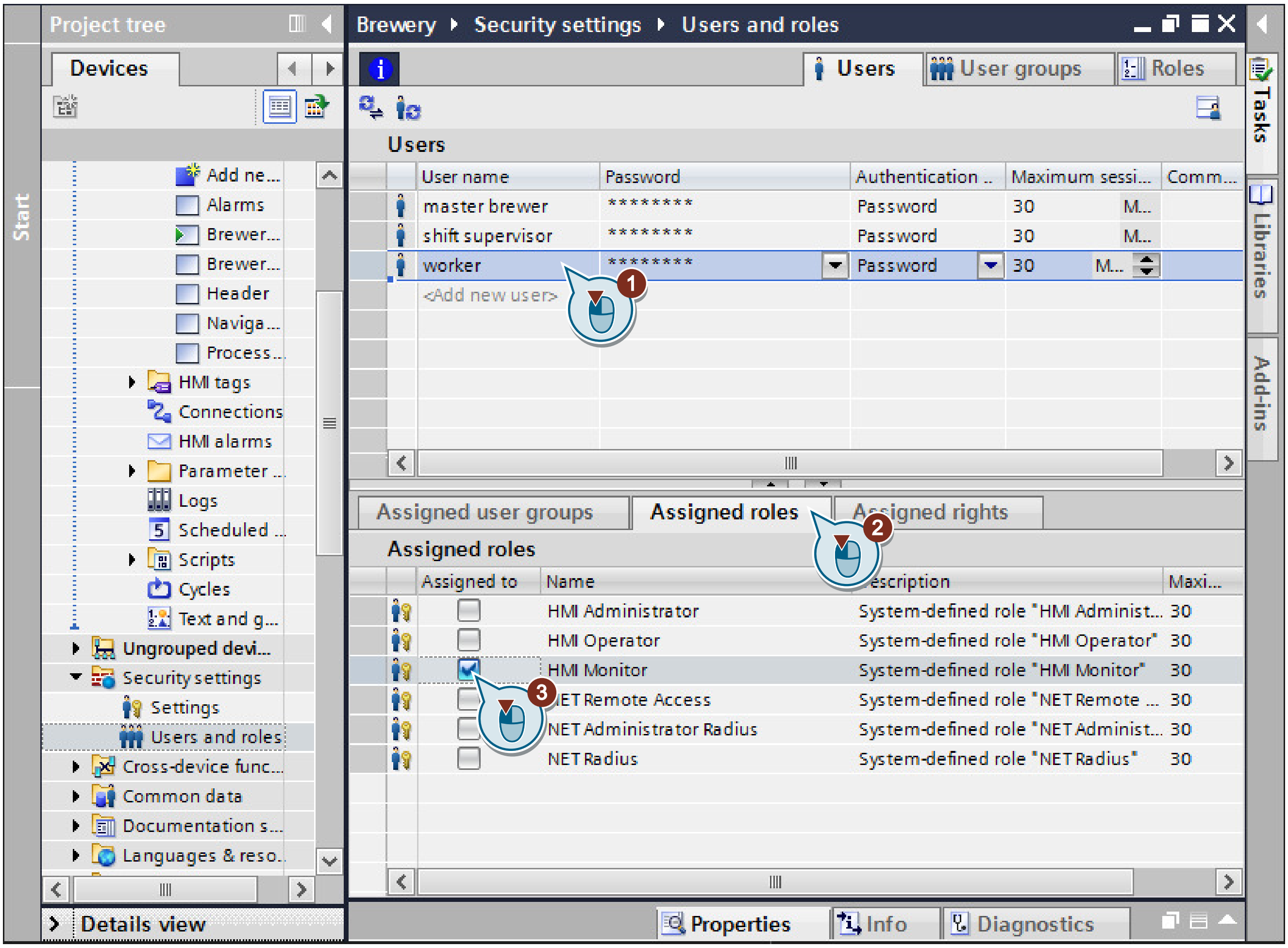
Users

User name	Password	Authentication ...	Maximum sessi...	Comm...
master brewer	********	Password	30 M...	
shift supervisor	********	Password	30 M...	
worker	****	Password	30 M...	
<Add new user>				

Assigned roles

Assigned to	Name	Description	Maxi...
☐	HMI Administrator	System-defined role "HMI Administ...	30
☑	HMI Operator	System-defined role "HMI Operator"	30
☐	HMI Monitor	System-defined role "HMI Monitor"	30
☐	NET Remote Access	System-defined role "NET Remote ...	30
☐	NET Administrator Radius	System-defined role "NET Administ...	30
☐	NET Radius	System-defined role "NET Radius"	30

6. Assign the role "worker" to the user "HMI Monitor".



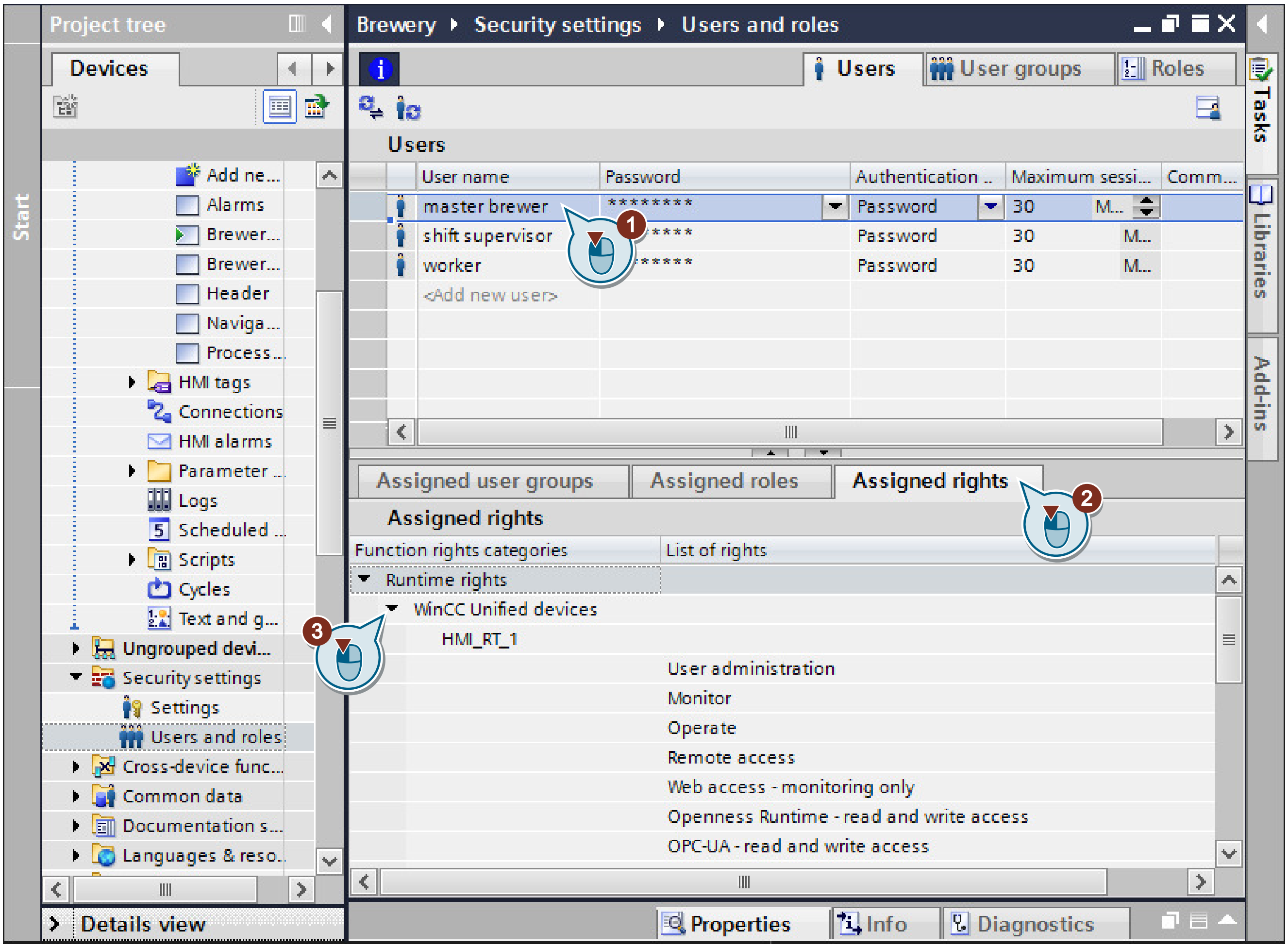
Users

User name	Password	Authentication ...	Maximum sessi...	Comm...
master brewer	********	Password	30 M...	
shift supervisor	********	Password	30 M...	
worker	********	Password	30 M...	
<Add new user>				

Assigned roles

Assigned to	Name	Description	Maxi...
☐	HMI Administrator	System-defined role "HMI Administ...	30
☐	HMI Operator	System-defined role "HMI Operator"	30
☑	HMI Monitor	System-defined role "HMI Monitor"	30
☐	NET Remote Access	System-defined role "NET Remote ...	30
☐	NET Administrator Radius	System-defined role "NET Administ...	30
☐	NET Radius	System-defined role "NET Radius"	30

7. Display and check the rights of a user.



Users

User name	Password	Authentication ...	Maximum sessi...	Comm...
master brewer	********	Password	30 M...	
shift supervisor	********	Password	30 M...	
worker	********	Password	30 M...	
<Add new user>				

Assigned rights

Function rights categories	List of rights
Runtime rights	
WinCC Unified devices	
HMI_RT_1	
	User administration
	Monitor
	Operate
	Remote access
	Web access - monitoring only
	Openness Runtime - read and write access
	OPC-UA - read and write access

If necessary, you can also define your own user roles.

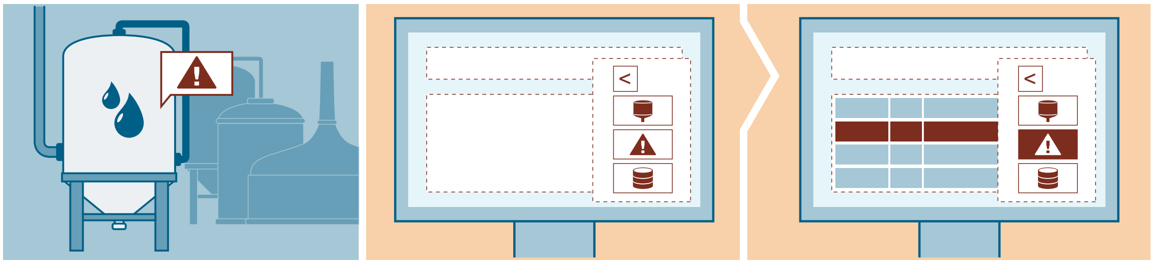
Configuring screen navigation

6

6.1 Screen navigation

Basics

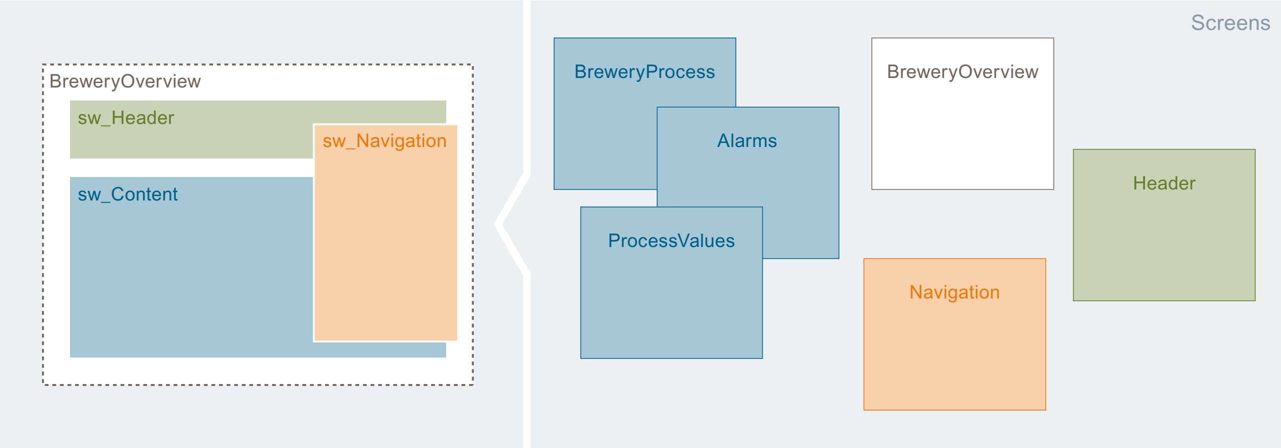
Process visualization is generally split between multiple screens, for example on the basis of functional or technological aspects. Changing between screens is referred to as screen navigation.



6.2 Screen window technology

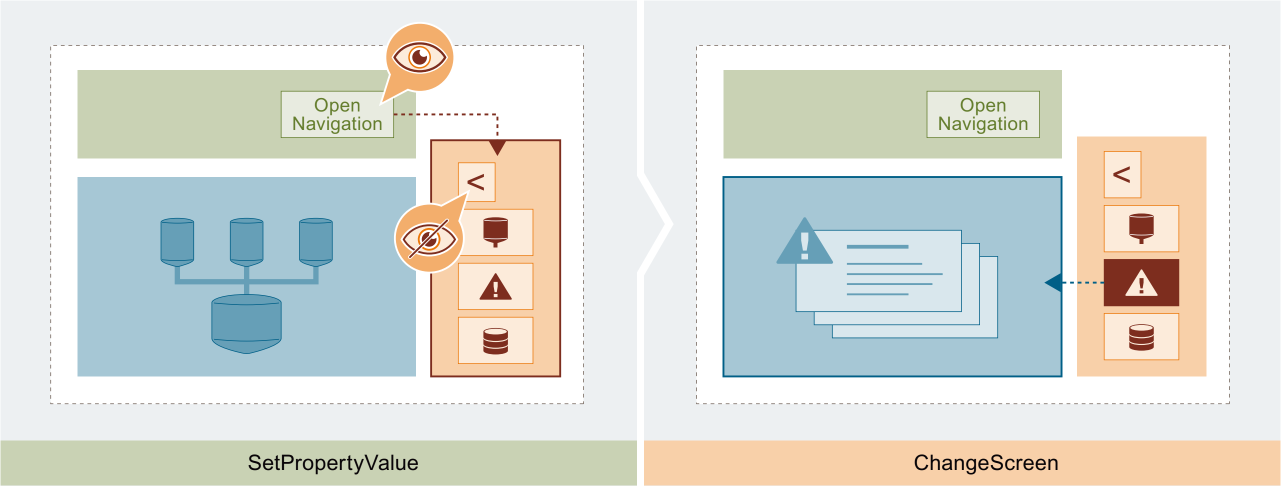
Basics

You use the screen window technology to implement the screen navigation. You divide the overview screen into three areas using screen windows and determine which HMI screens are displayed in the respective screen window.



Information on the system is displayed in the screen window "sw_Content", for example the brewing process, alarms or process values. The HMI screens that are used for navigation are displayed in the "sw_Header" and "sw_Navigation" screen windows.

Use the "Open Navigation" button in the HMI screen "Header" to show the HMI screen "Navigation". Using the buttons in the HMI screen "Navigation", the operator changes the screens in the screen window "sw_Content" in Runtime.



Configuration

In this section, you will learn the following:

- Inserting and configuring screen windows
- Configuring buttons for showing and hiding screen windows using the system function "SetPropertyValue"
- Configuring buttons for changing the content of a screen window using the system function "ChangeScreen"

6.3 Configuring screen window for navigation

Introduction

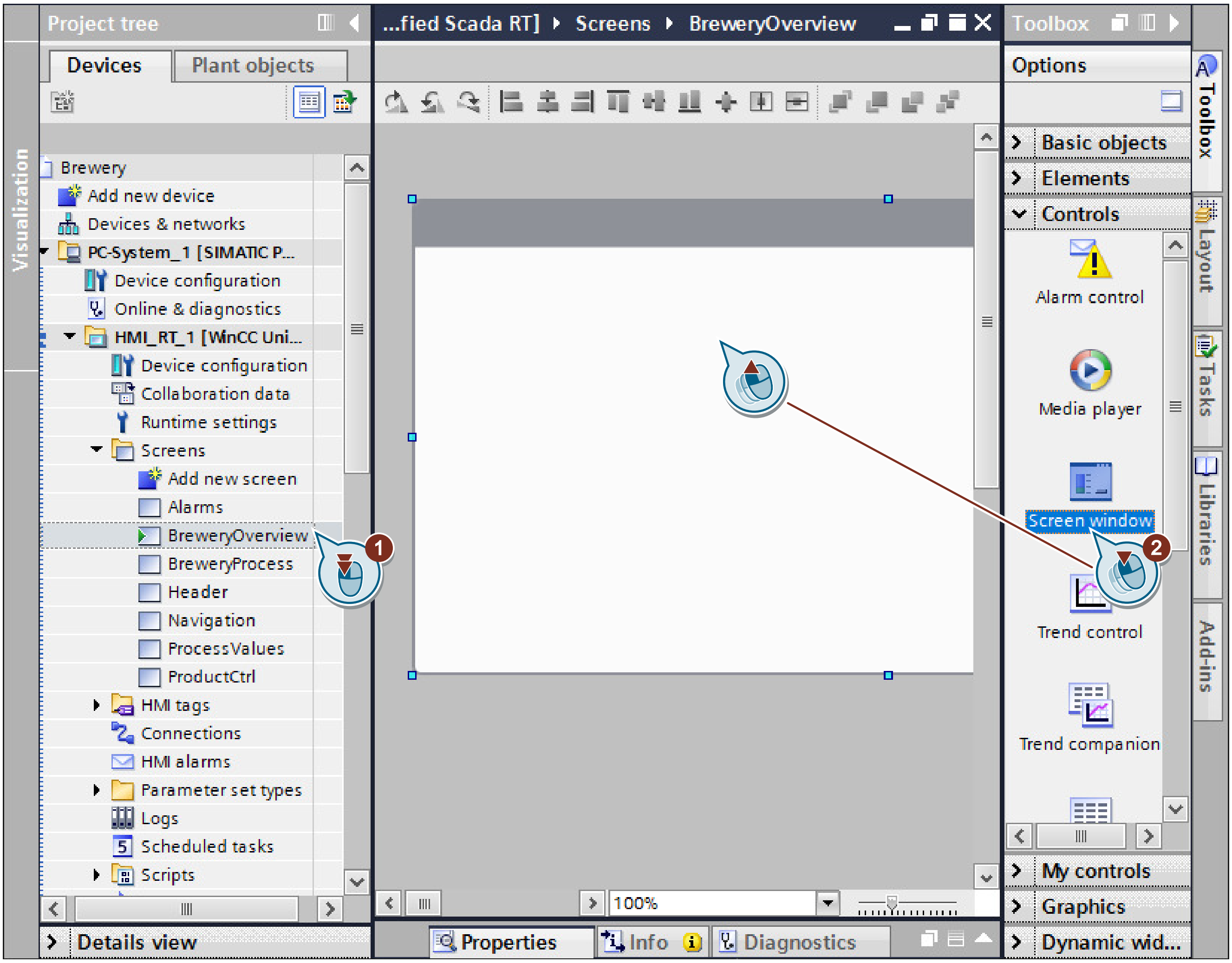
You implement the screen navigation using the "screen window technology". With the screen window technology, you implement picture-in-picture view: The screen changes are executed in the screen window. The root screen with the screen windows is always visible.

Object	Type	Label	Use
BreweryOverview	HMI screen	-	Overview screen for operating and monitoring the brewing process
sw_Content	Screen window	-	Screen for displaying the screens for process control
sw_Header	Screen window	-	Screen for opening the navigation between process control screens

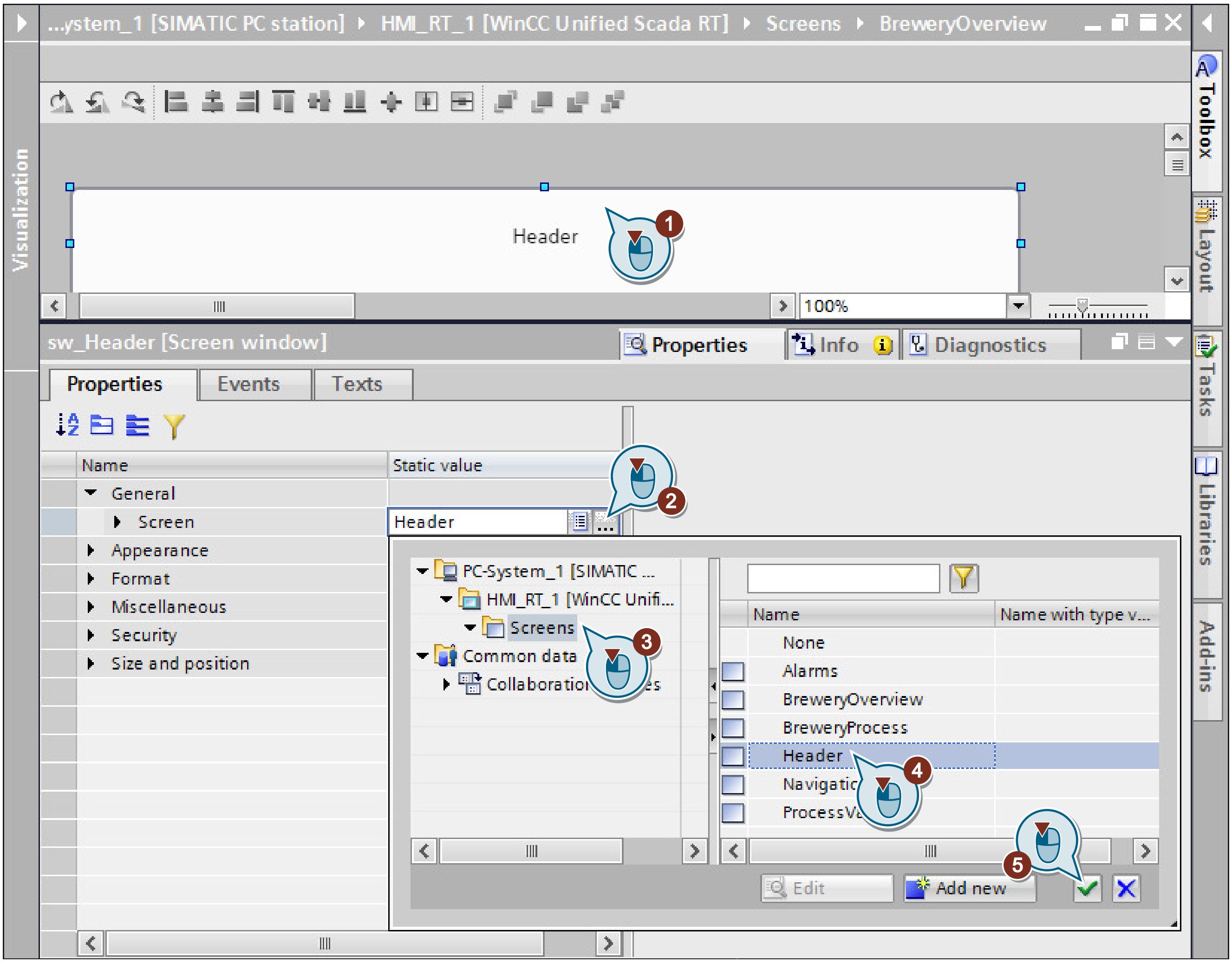
Object	Type	Label	Use
sw_Navigation	Screen window	-	Screen for navigating between process control screens
btn_ShowNavigation	Button	Open Navigation	Shows the screen window "sw_Navigation"
btn_HideNavigation	Button	<	Hides the screen window "sw_Navigation"
btn_ProcessBrewery	Button	Brewery	Screen change for the brewing process
btn_ProcessValues	Button	Process Values	Screen change for process values from the brewing process
btn_Alarms	Button	Alarms	Screen change for alarms from the brewing process
btn_ProductCtrl	Button	Production	Screen change for production control

Procedure

1. Open the HMI screen "BreweryOverview" and insert a screen window to display the Headers.

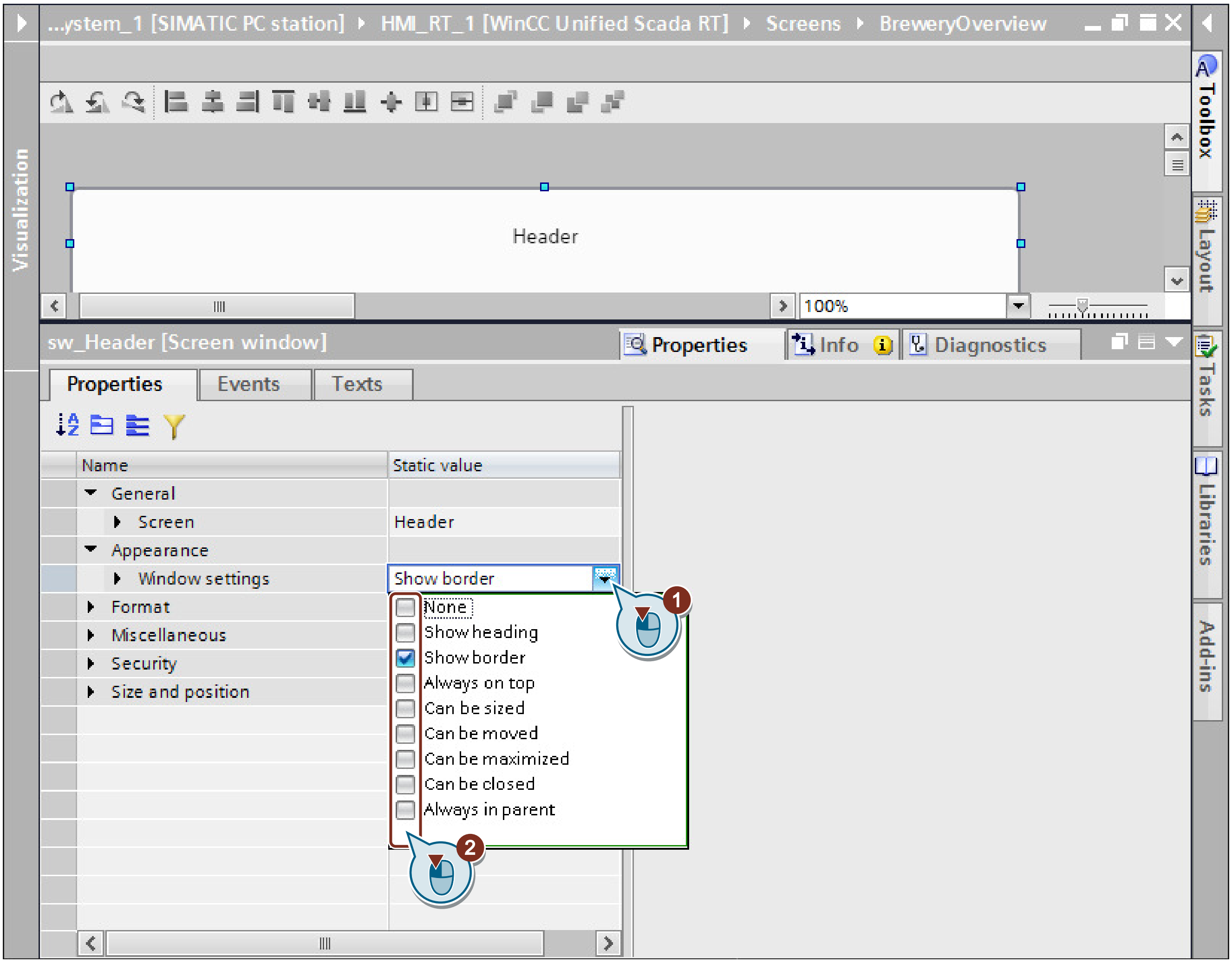


2. Specify that the HMI screen "Header" is displayed in the screen window.



...ystem_1 [SIMATIC PC station] ▸ HMI_RT_1 [WinCC Unified Scada RT] ▸ Screens ▸ BreweryOverview
Visualization
Toolbox
Layout
Tasks
Libraries
Add-ins
Header
1
100%
sw_Header [Screen window]
Properties
Info
Diagnostics
Properties
Events
Texts
Name
Static value
General
Screen
Appearance
Format
Miscellaneous
Security
Size and position
Header
2
PC-System_1 [SIMATIC ...
HMI_RT_1 [WinCC Unifi...
Screens
3
Common data
Collaboration...es
Name
Name with type v...
None
Alarms
BreweryOverview
BreweryProcess
Header
4
Navigatic
ProcessV
5
Edit
Add new

3. Configure the window settings for the screen window.



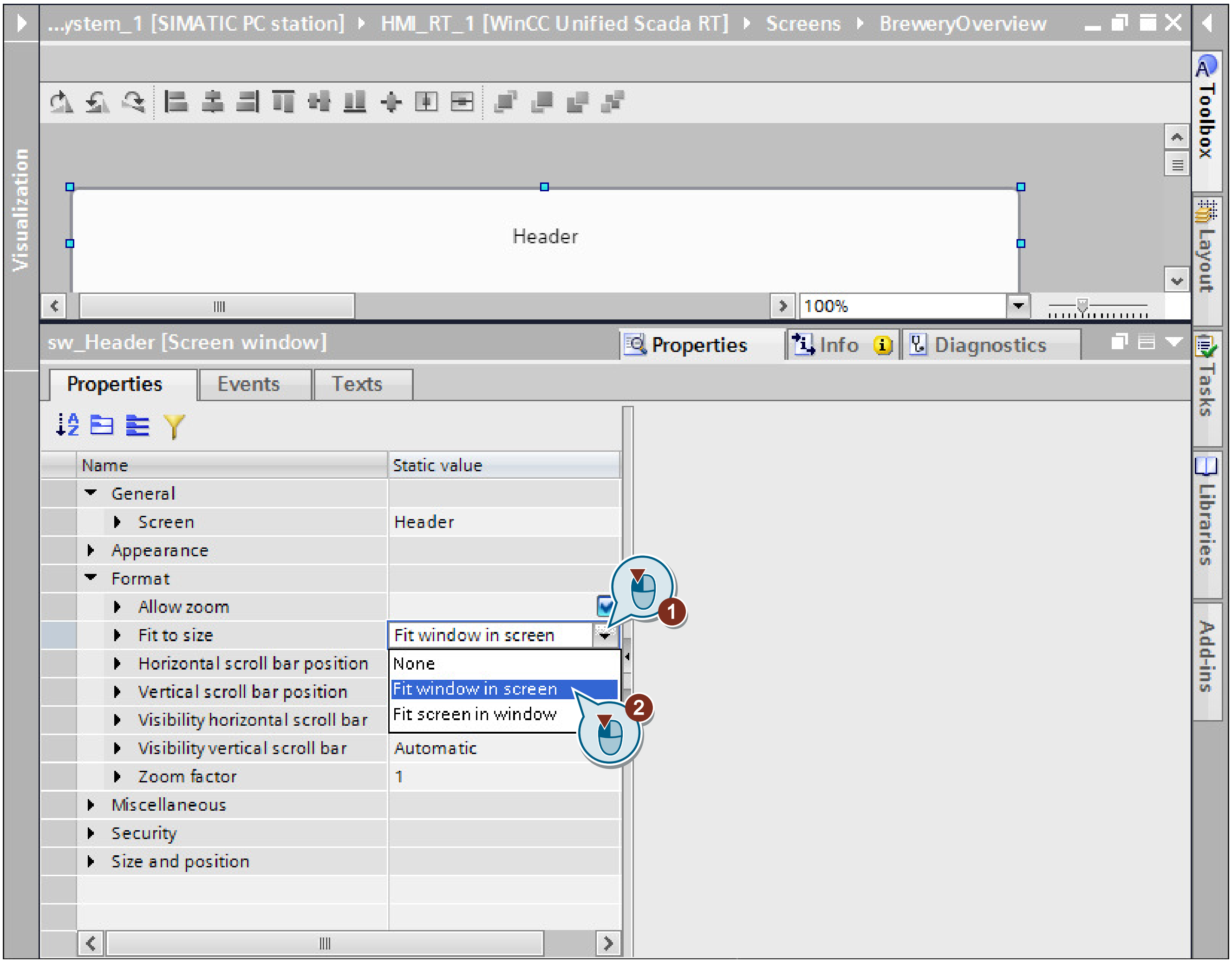
...ystem_1 [SIMATIC PC station] ▸ HMI_RT_1 [WinCC Unified Scada RT] ▸ Screens ▸ BreweryOverview
Visualization
Toolbox
Layout
Tasks
Libraries
Add-ins
Header
100%
Properties
Info
Diagnostics
Events
Texts
1
2

sw_Header [Screen window]

Name	Static value
General	
Screen	Header
Appearance	
Window settings	Show border
Format	
Miscellaneous	
Security	
Size and position	

- □ None
- □ Show heading
- ☑ Show border
- □ Always on top
- □ Can be sized
- □ Can be moved
- □ Can be maximized
- □ Can be closed
- □ Always in parent

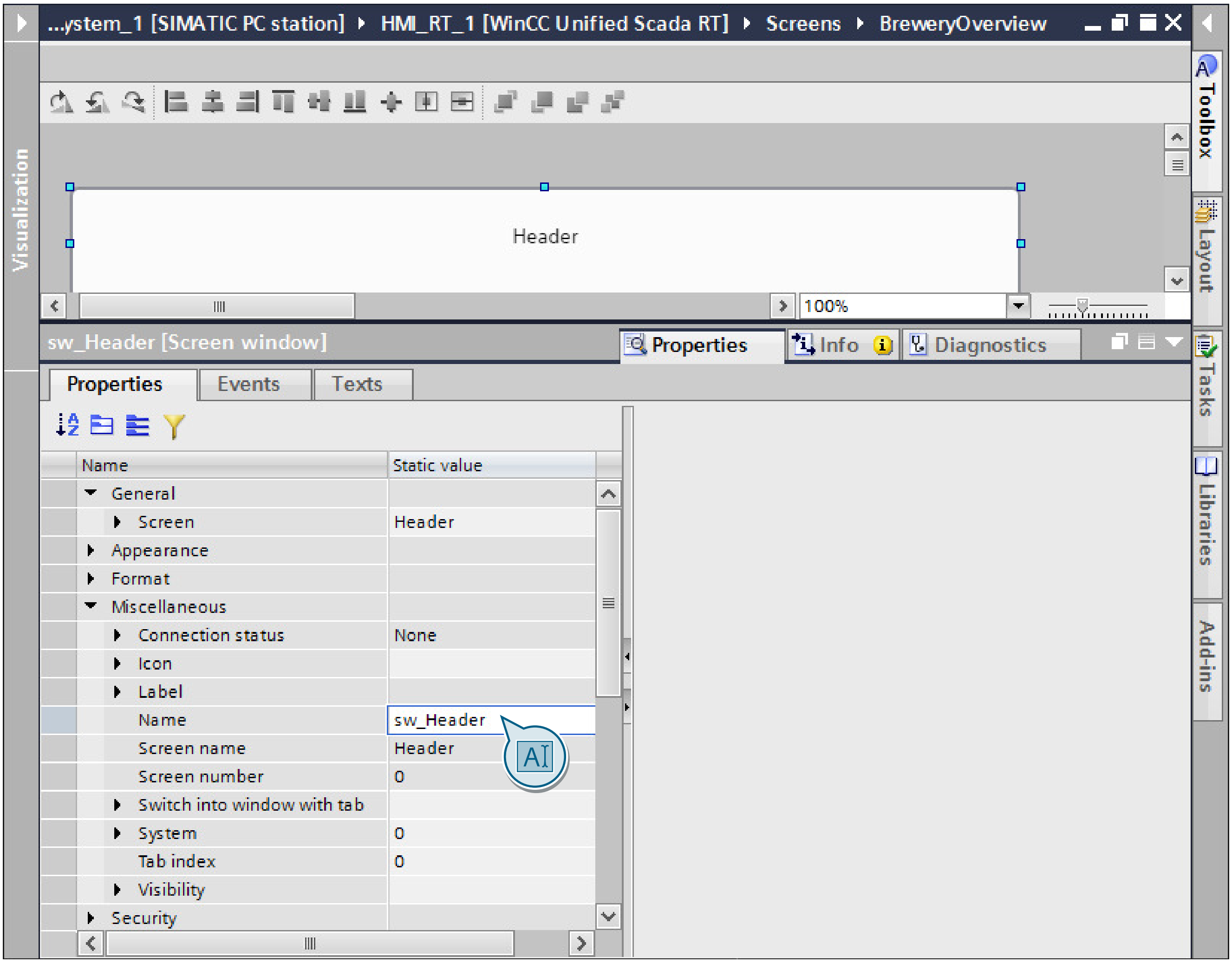
4. Configure the screen window so that it adapts to the size of the displayed HMI picture.



...ystem_1 [SIMATIC PC station] ▸ HMI_RT_1 [WinCC Unified Scada RT] ▸ Screens ▸ BreweryOverview
Visualization
Toolbox
Layout
Tasks
Libraries
Add-ins
Header
100%
sw_Header [Screen window]
Properties
Info
Diagnostics
Properties
Events
Texts
1
2

Name	Static value
General	
Screen	Header
Appearance	
Format	
Allow zoom	☑
Fit to size	Fit window in screen
Horizontal scroll bar position	None
Vertical scroll bar position	Fit window in screen
Visibility horizontal scroll bar	Fit screen in window
Visibility vertical scroll bar	Automatic
Zoom factor	1
Miscellaneous	
Security	
Size and position	

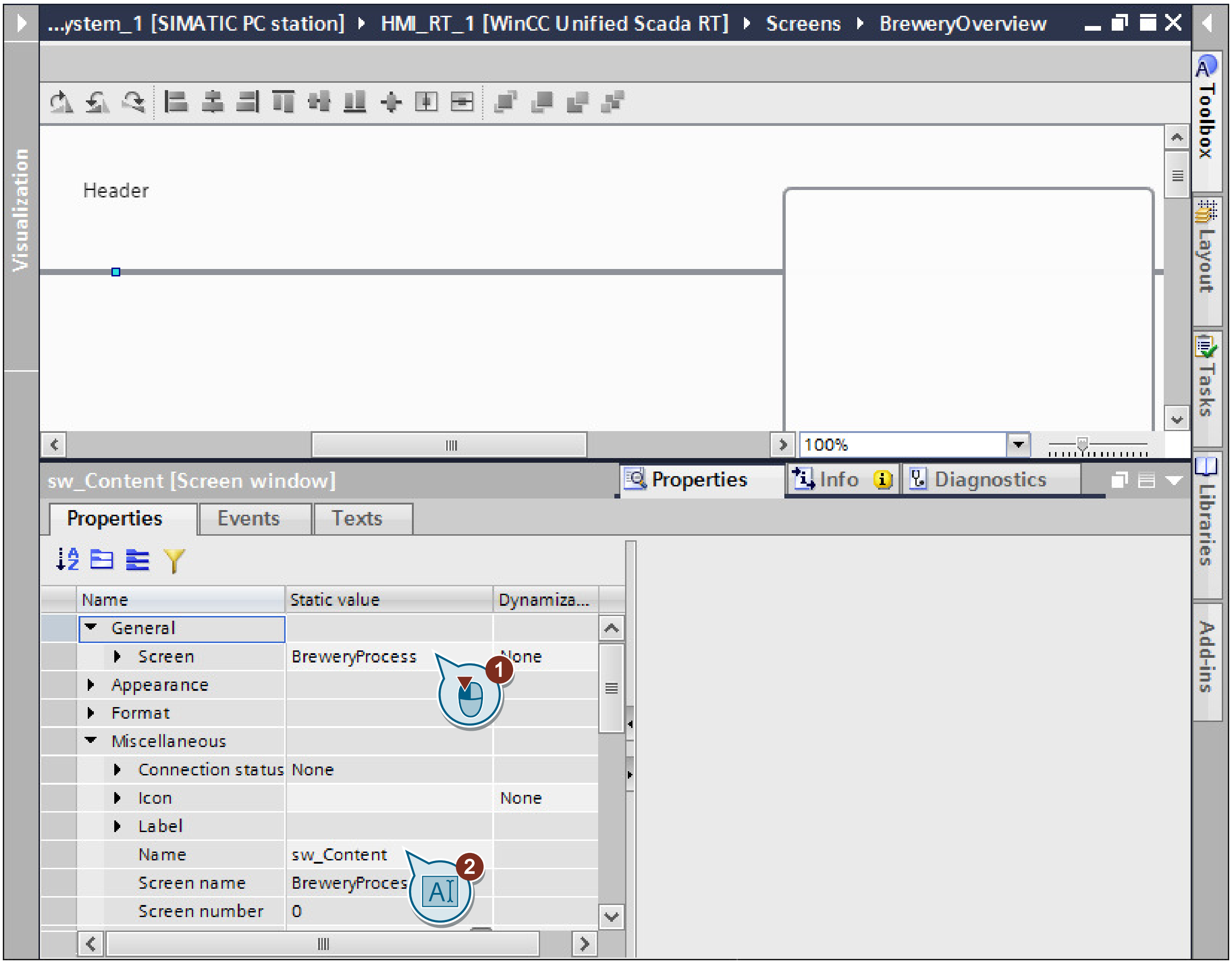
5. Specify the name of the screen window.



...ystem_1 [SIMATIC PC station] ▸ HMI_RT_1 [WinCC Unified Scada RT] ▸ Screens ▸ BreweryOverview
Visualization
Toolbox
Layout
Tasks
Libraries
Add-ins
Header
100%
sw_Header [Screen window]
Properties
Info
Diagnostics
Properties
Events
Texts
AI

Name	Static value
General	
Screen	Header
Appearance	
Format	
Miscellaneous	
Connection status	None
Icon	
Label	
Name	sw_Header
Screen name	Header
Screen number	0
Switch into window with tab	
System	0
Tab index	0
Visibility	
Security	

6. Configure the screen window "sw_Content":
   - Copy the screen window "sw_Header".
   - Rename the copied screen window to "sw_Content".
   - Specify that the HMI screen "BreweryProcess" is displayed in the screen window.

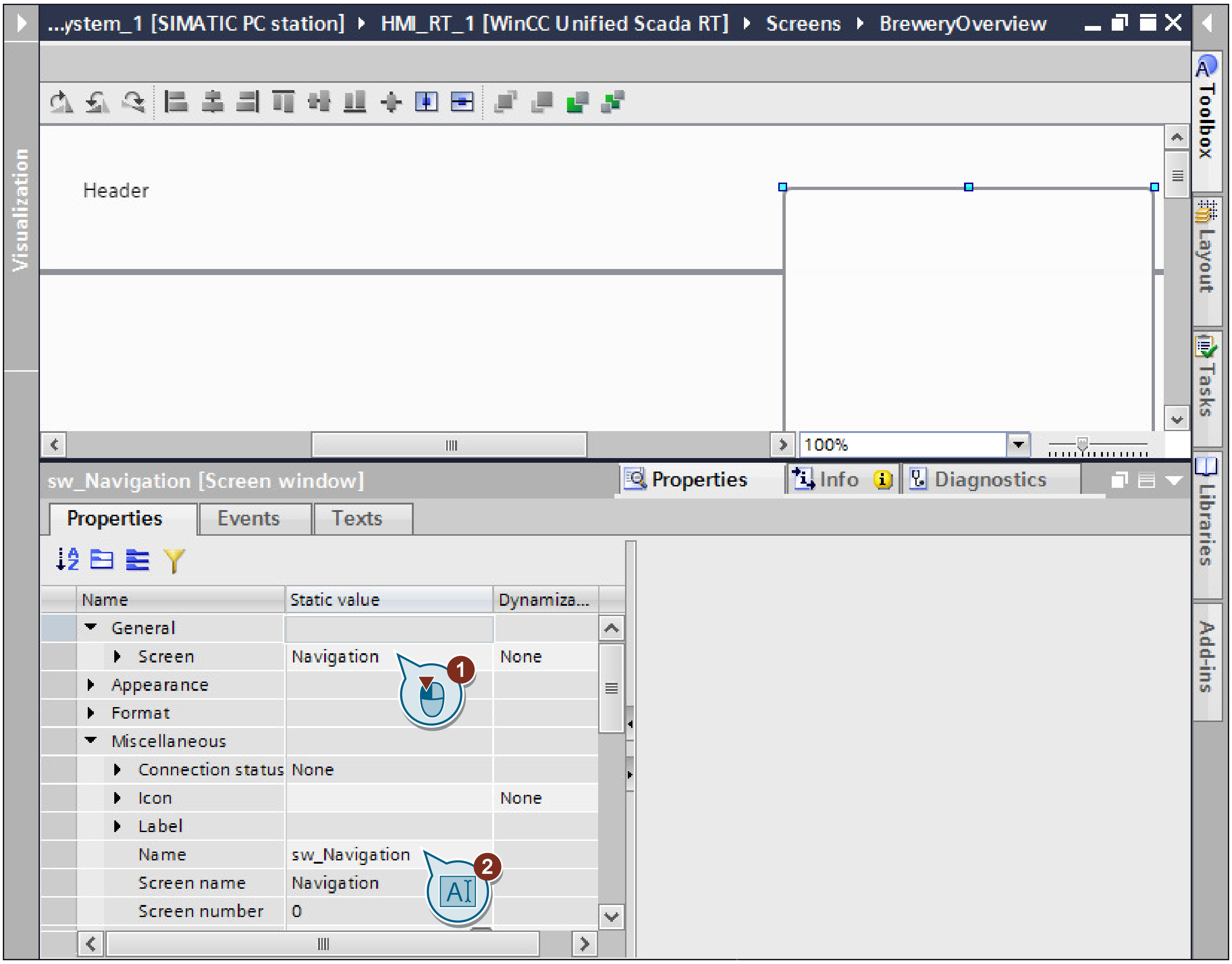


...ystem_1 [SIMATIC PC station] ▸ HMI_RT_1 [WinCC Unified Scada RT] ▸ Screens ▸ BreweryOverview

sw_Content [Screen window]

Name	Static value	Dynamiza...
General		
Screen	BreweryProcess	None
Appearance		
Format		
Miscellaneous		
Connection status	None	
Icon		None
Label		
Name	sw_Content	
Screen name	BreweryProces	
Screen number	0	

7. Configure the screen window "sw_Navigation":
   - Copy the screen window "sw_Header".
   - Rename the copied screen window to "sw_Navigation".
   - Specify that the HMI screen "Navigation" is displayed in the screen window.

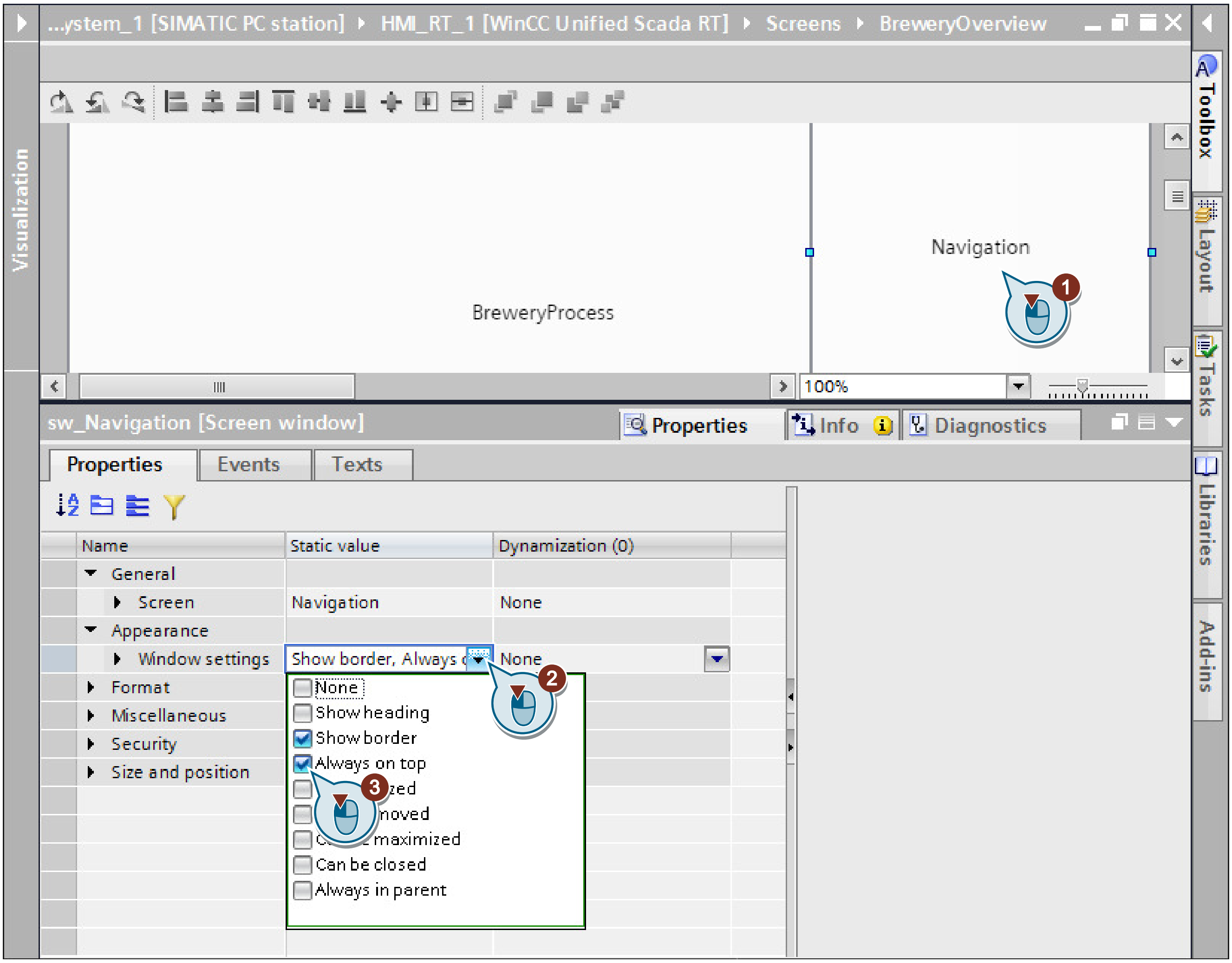


...ystem_1 [SIMATIC PC station] ▸ HMI_RT_1 [WinCC Unified Scada RT] ▸ Screens ▸ BreweryOverview

sw_Navigation [Screen window]

Name	Static value	Dynamiza...
General		
Screen	Navigation	None
Appearance		
Format		
Miscellaneous		
Connection status	None	
Icon		None
Label		
Name	sw_Navigation	
Screen name	Navigation	
Screen number	0	

   - Specify that the screen window "sw_Navigation" is always displayed in the foreground.



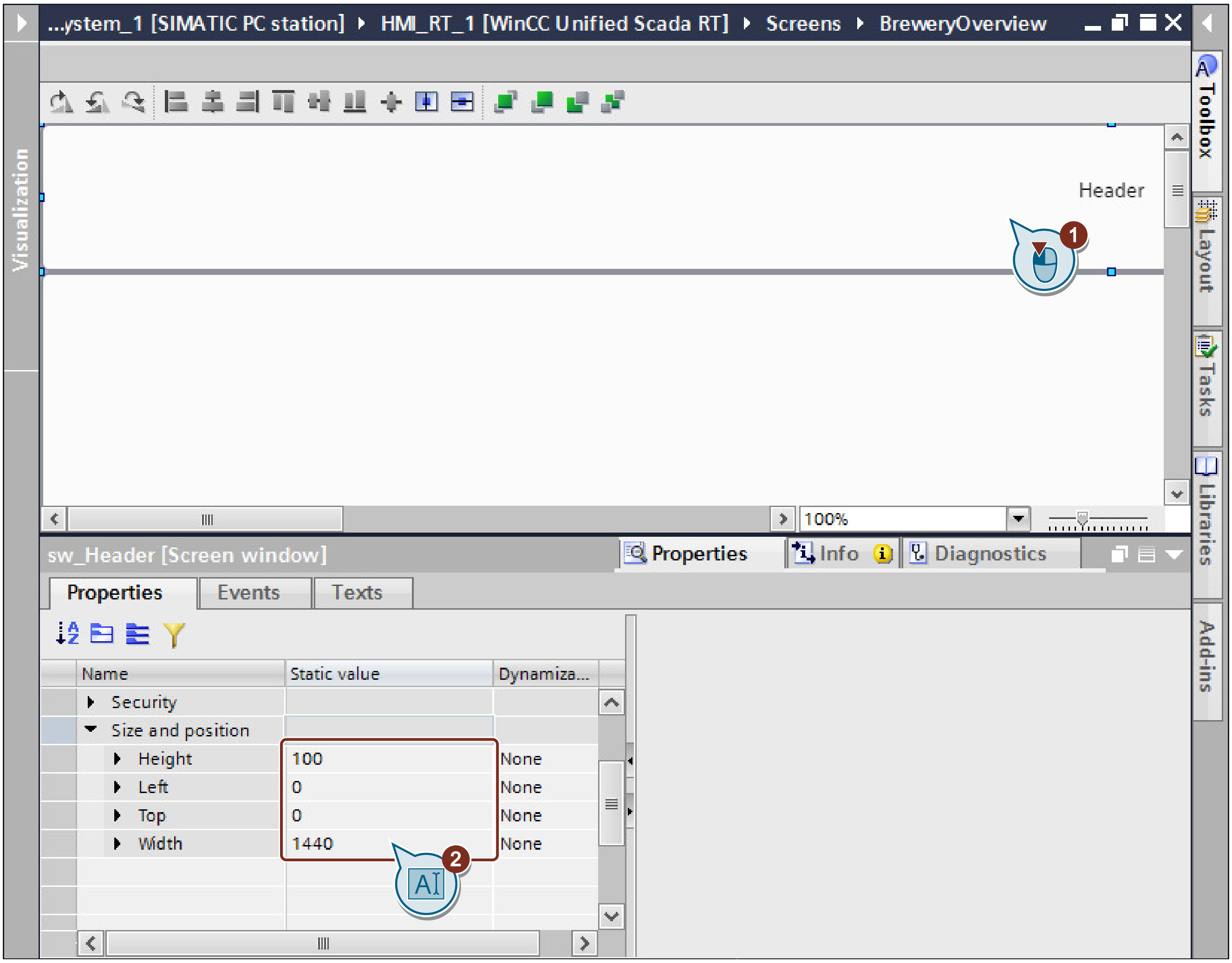
sw_Navigation [Screen window]

Name	Static value	Dynamization (0)
▾ General		
▸ Screen	Navigation	None
▾ Appearance		
▸ Window settings	Show border, Always o	None
▸ Format		
▸ Miscellaneous		
▸ Security		
▸ Size and position		

- ☐ None
- ☐ Show heading
- ☑ Show border
- ☑ Always on top
- ☐ zed
- ☐ noved
- ☐ C maximized
- ☐ Can be closed
- ☐ Always in parent

8. Position the three screen windows in the HMI screen:

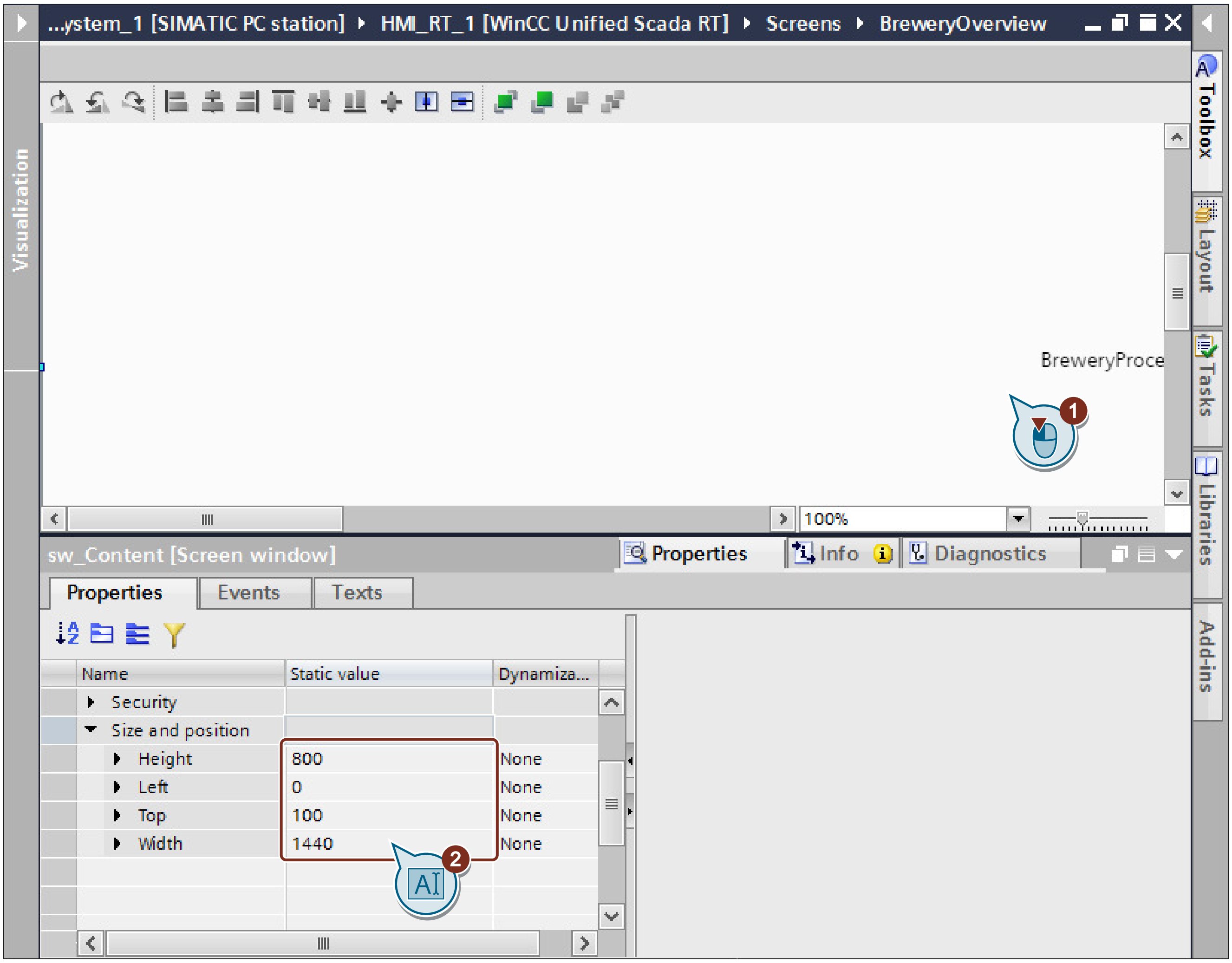
– Specify the position for the screen window "sw_Header".



...ystem_1 [SIMATIC PC station] ▸ HMI_RT_1 [WinCC Unified Scada RT] ▸ Screens ▸ BreweryOverview
Visualization
Header
1
100%
Toolbox
Layout
Tasks
Libraries
Add-ins
sw_Header [Screen window]
Properties
Info
Diagnostics
Properties
Events
Texts
2

Name	Static value	Dynamiza...
Security		
Size and position		
Height	100	None
Left	0	None
Top	0	None
Width	1440	None

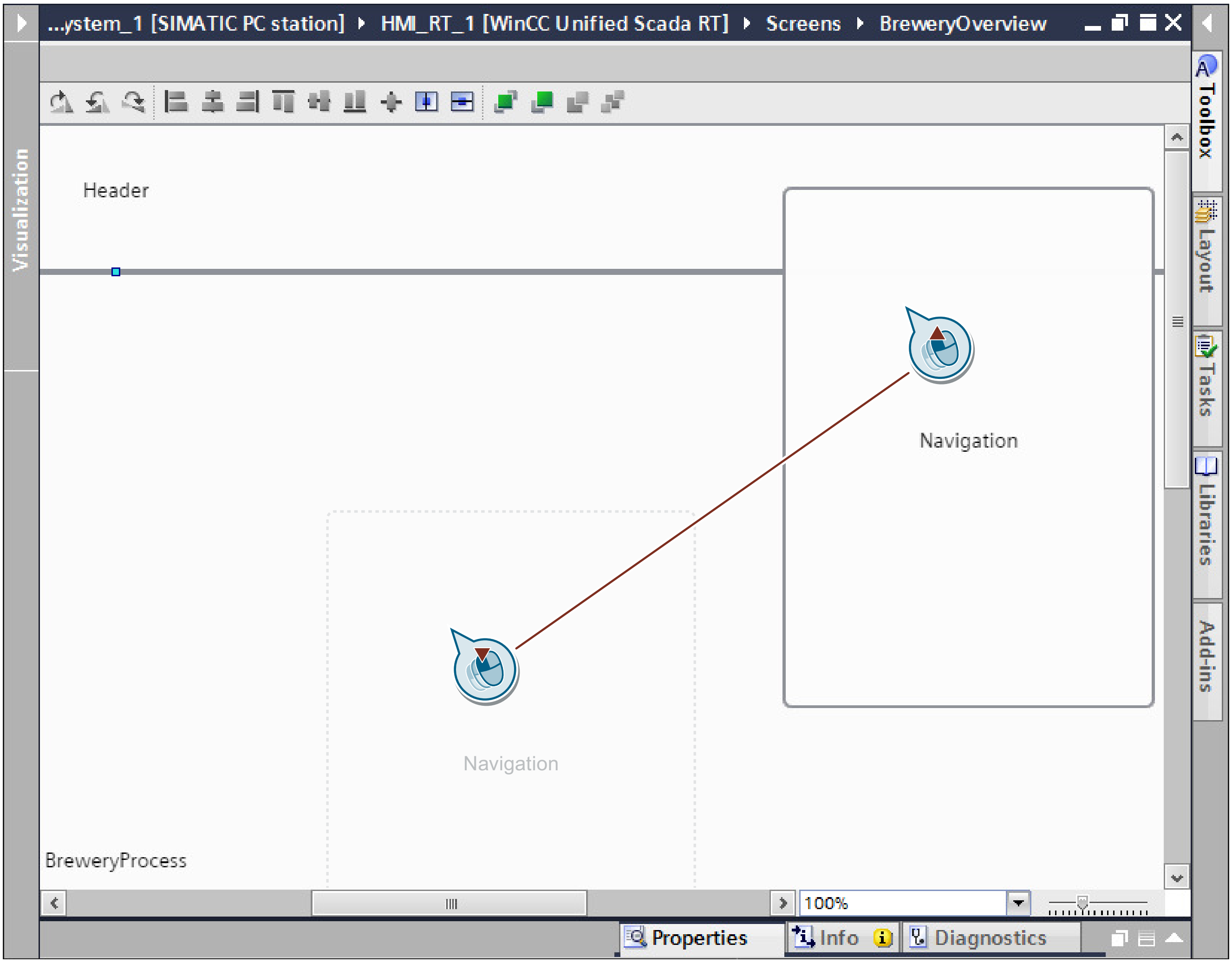
– Specify the position for the screen window "sw_Content".



...ystem_1 [SIMATIC PC station] ▸ HMI_RT_1 [WinCC Unified Scada RT] ▸ Screens ▸ BreweryOverview
Visualization
Toolbox
Layout
Tasks
Libraries
Add-ins
BreweryProce
1
100%
sw_Content [Screen window]
Properties
Info
Diagnostics
Properties
Events
Texts
2

Name	Static value	Dynamiza...
▸ Security		
▾ Size and position		
▸ Height	800	None
▸ Left	0	None
▸ Top	100	None
▸ Width	1440	None

– Using the mouse, position the screen window "sw_Navigation" over the other two screen windows.



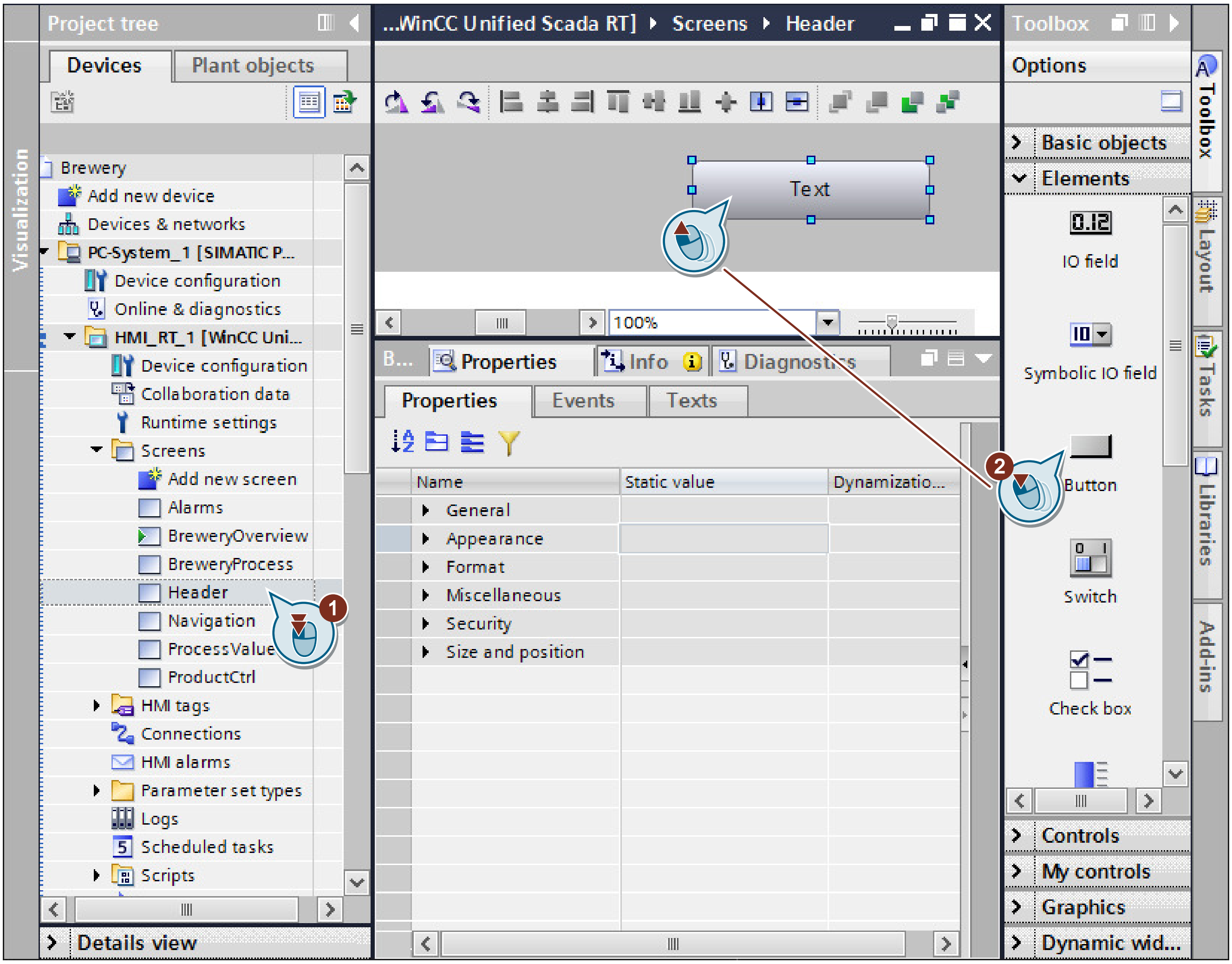
6.4 Configure the HMI screen to open the navigation window

Introduction

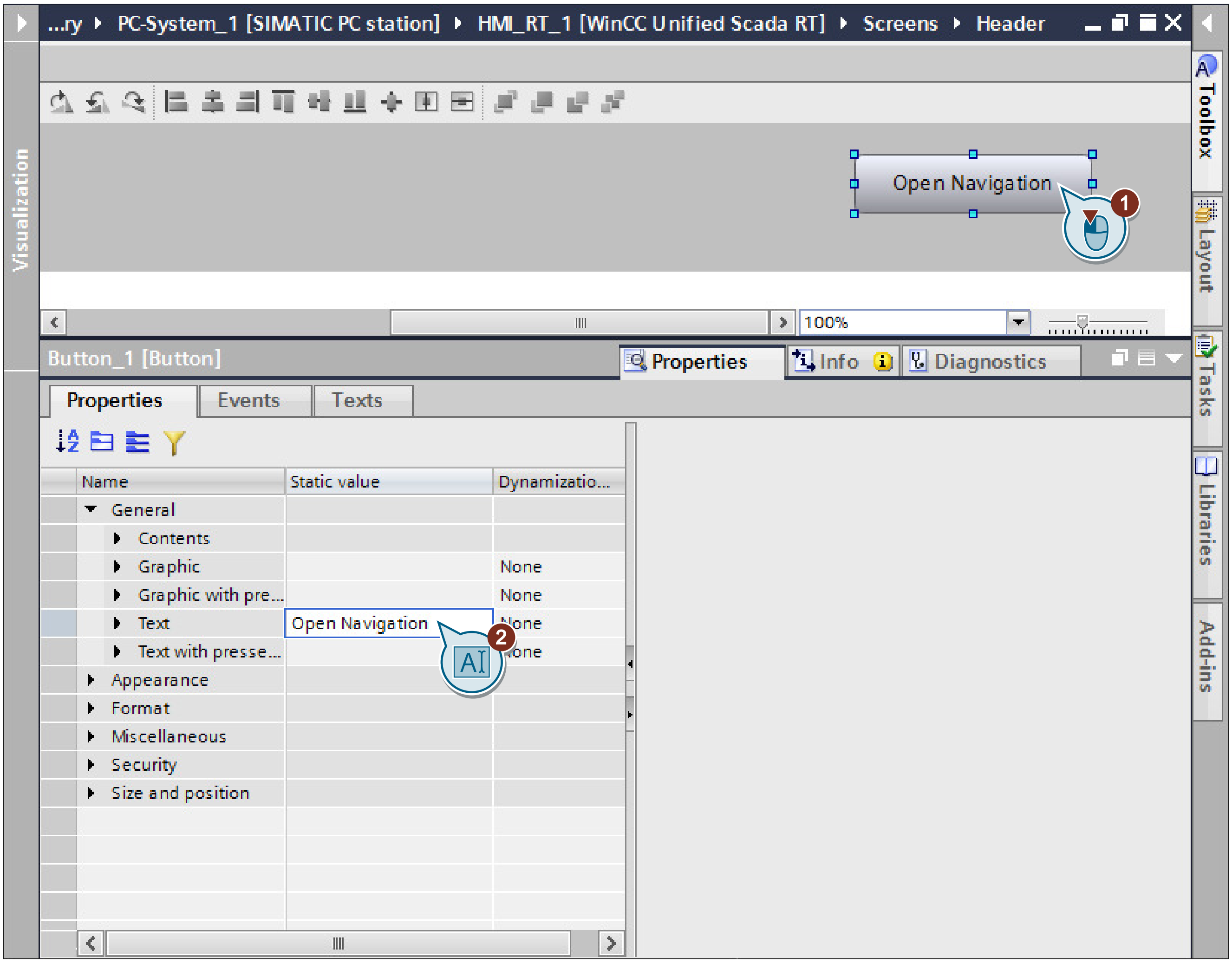
In the following, you configure the HMI screen "Header" which is displayed in the screen window "sw_Header". Use the button in the HMI screen "Header" to show the screen window "sw_Navigation" in which the operator changes the screen in the screen window "sw_Content".

Procedure

1. Open the HMI screen "Header" and insert the button that shows the screen window "sw_Navigation".



2. Define the label for the button.

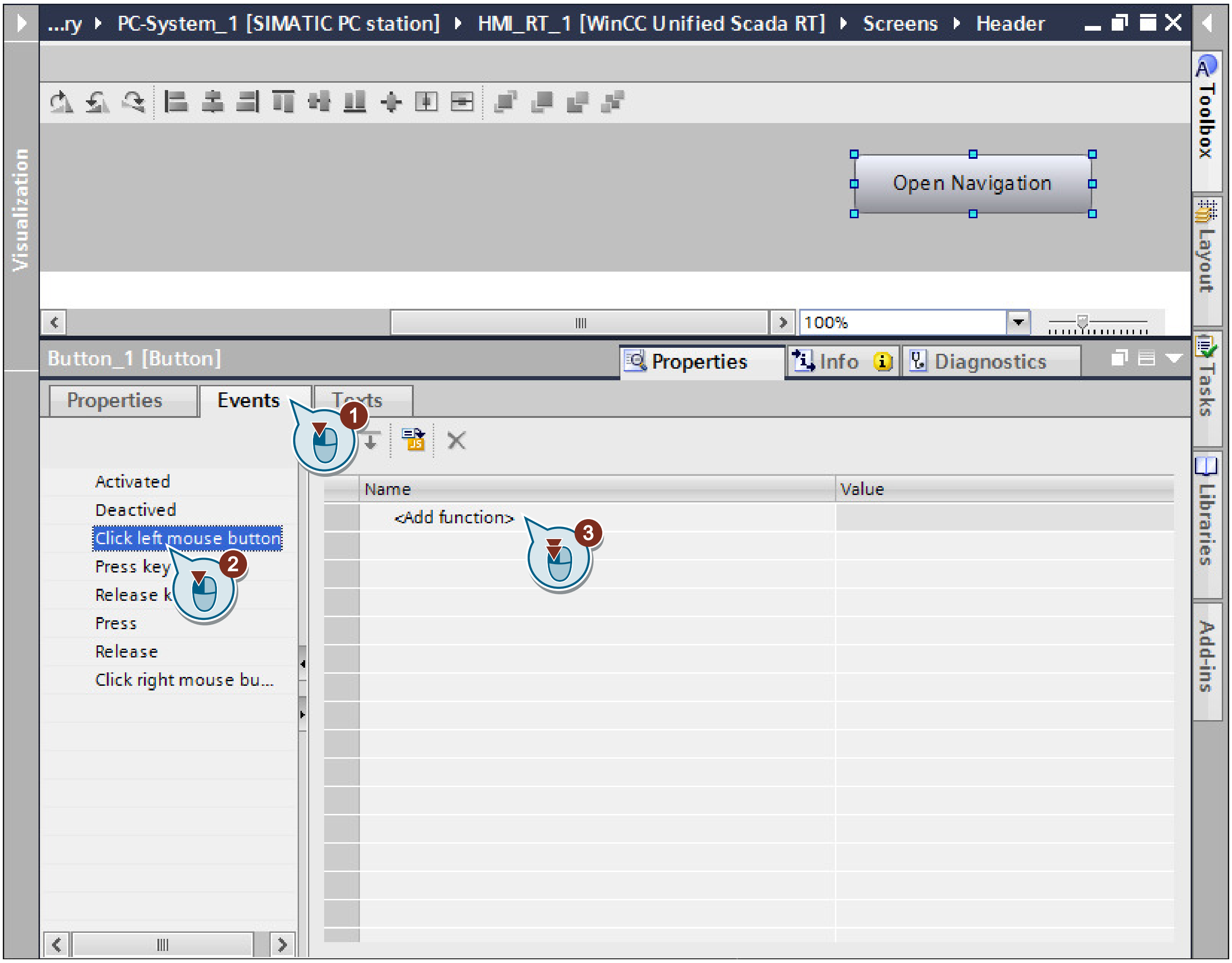


...ry ▸ PC-System_1 [SIMATIC PC station] ▸ HMI_RT_1 [WinCC Unified Scada RT] ▸ Screens ▸ Header

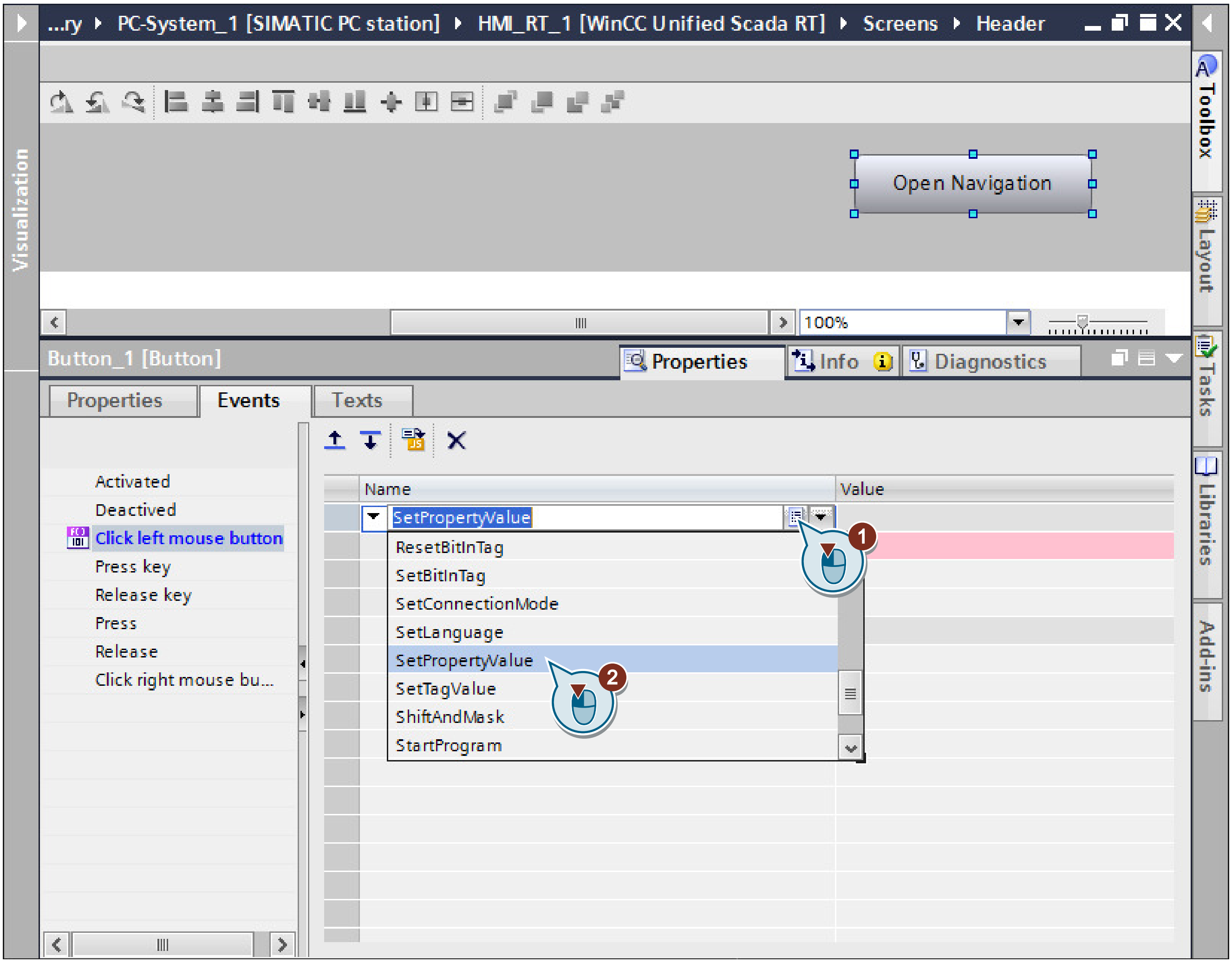
Button_1 [Button]

Name	Static value	Dynamizatio...
General		
Contents		
Graphic		None
Graphic with pre...		None
Text	Open Navigation	None
Text with presse...		None
Appearance		
Format		
Miscellaneous		
Security		
Size and position		

3. Add a system function to the event "Click left mouse button".



4. Select the system function for setting the property value.

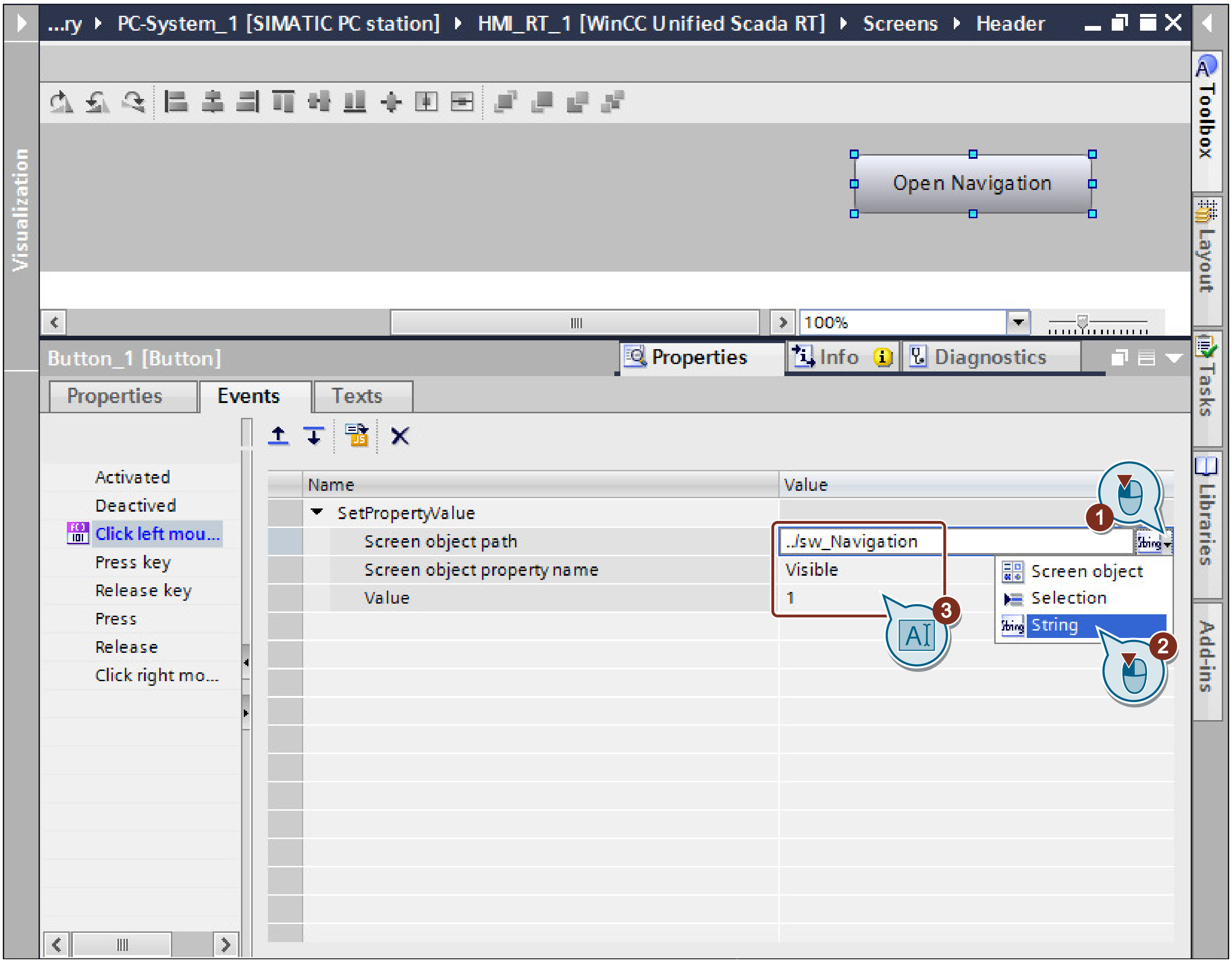


5. Specify the parameters of the system function "SetPropertyValue".

Note

Upper and lower case

Pay attention to the correct case when setting the parameters.



Name	Value
SetPropertyValue	
Screen object path	../sw_Navigation
Screen object property name	Visible
Value	1

6.5 Configuring the HMI screen to open the process screens

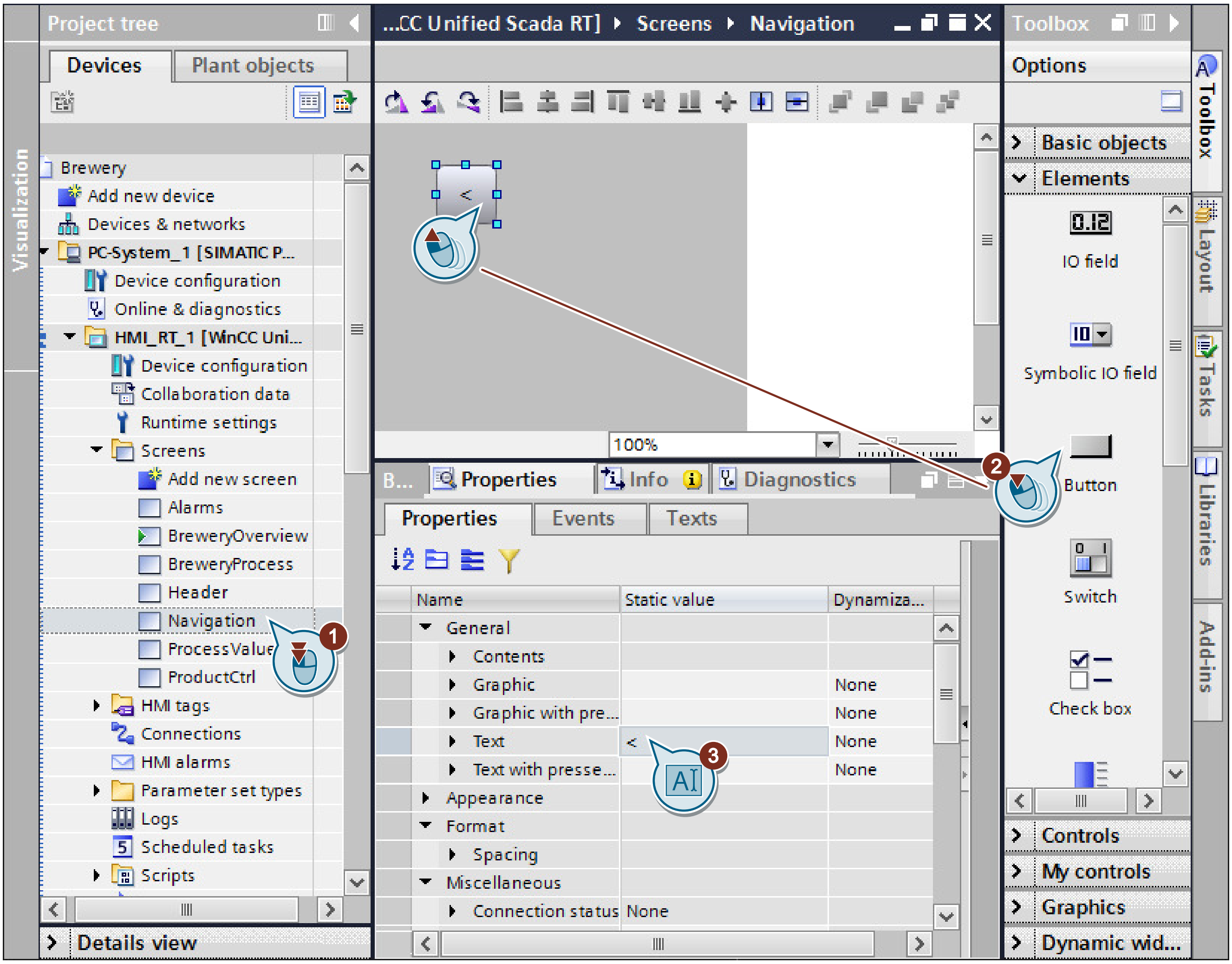
Introduction

In the following, you configure the HMI screen "Navigation" which is displayed in the screen window "sw_Navigation". Use the buttons in the HMI screen "Navigation" to change the displayed HMI screen in the screen window "sw_Content".

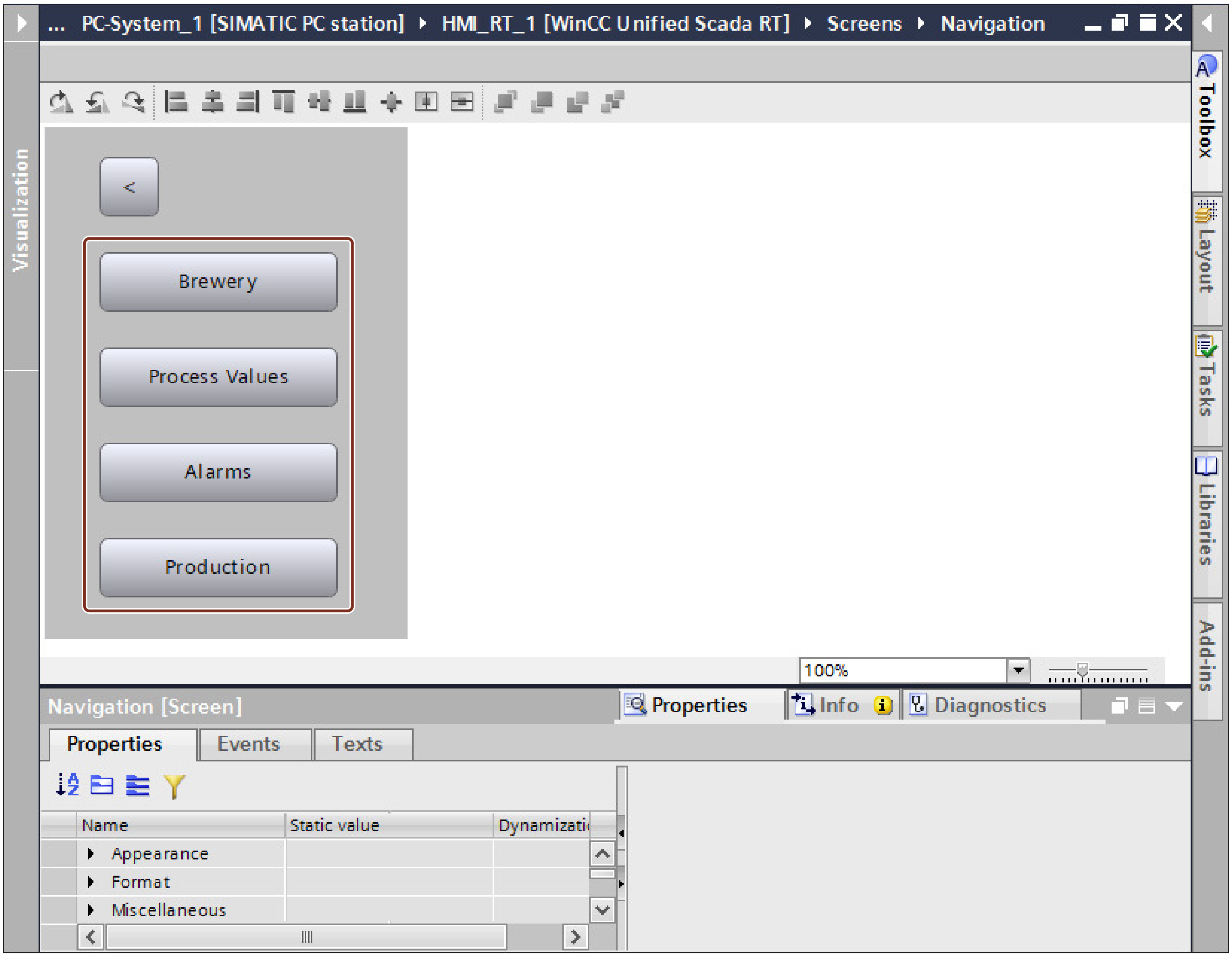
The screen window "sw_Navigation" is hidden by default and can be shown using the "Open Navigation" button in the "Header" and hidden again using the "<" button in the HMI screen "Navigation".

Procedure

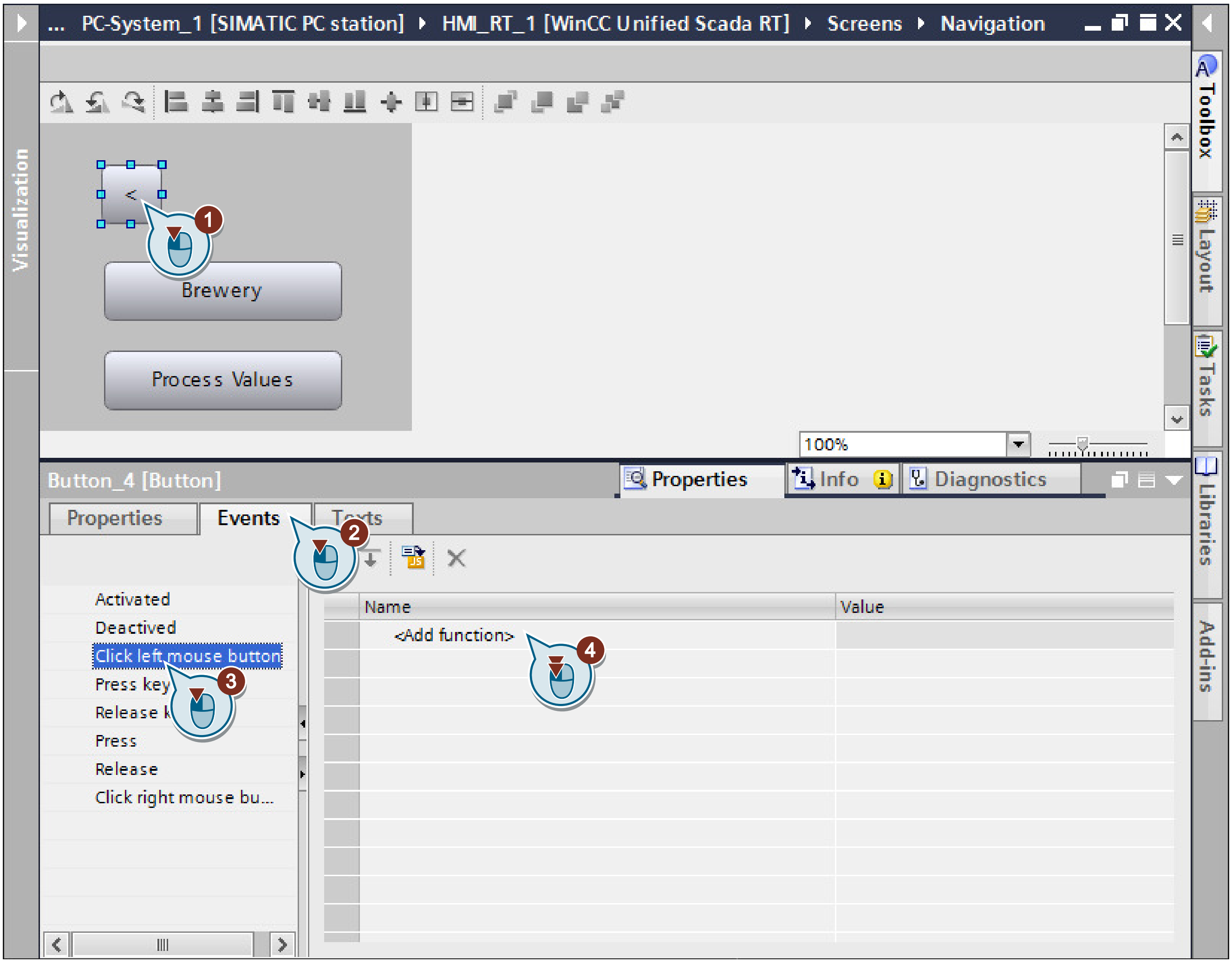
1. Open the HMI screen "Navigation" and insert the button that hides the screen window "sw_Navigation".



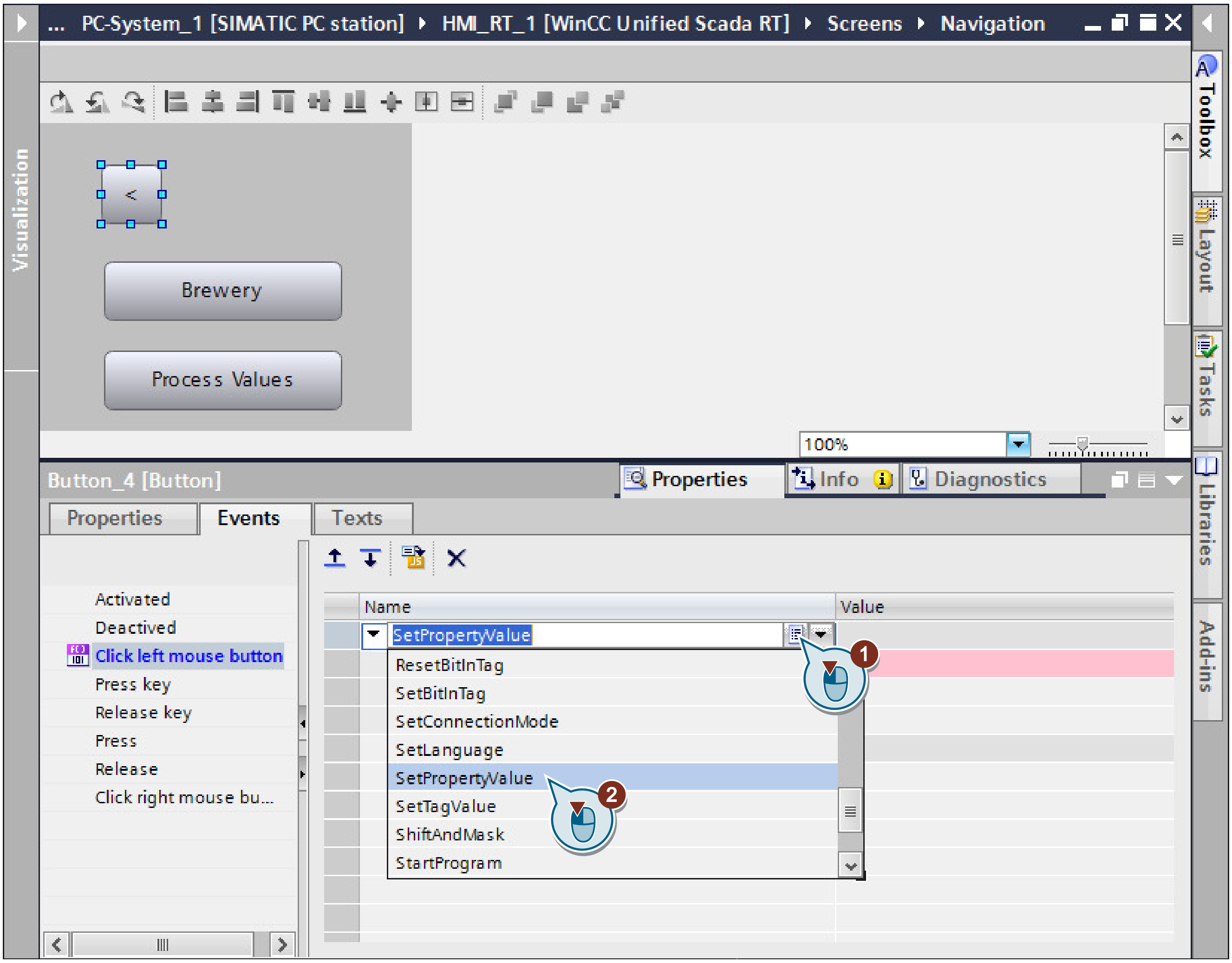
2. Insert the other buttons in the same way.


... PC-System_1 [SIMATIC PC station] ▸ HMI_RT_1 [WinCC Unified Scada RT] ▸ Screens ▸ Navigation
Visualization
Toolbox
Layout
Tasks
Libraries
Add-ins
<
Brewery
Process Values
Alarms
Production
100%
Navigation [Screen]
Properties
Info
Diagnostics
Properties
Events
Texts
Name
Static value
Dynamizati
Appearance
Format
Miscellaneous

3. Configure the button "<" that hides the screen window "sw_Navigation".
   – Add a system function to the event "Click left mouse button".



– Select the system function "SetPropertyValue".

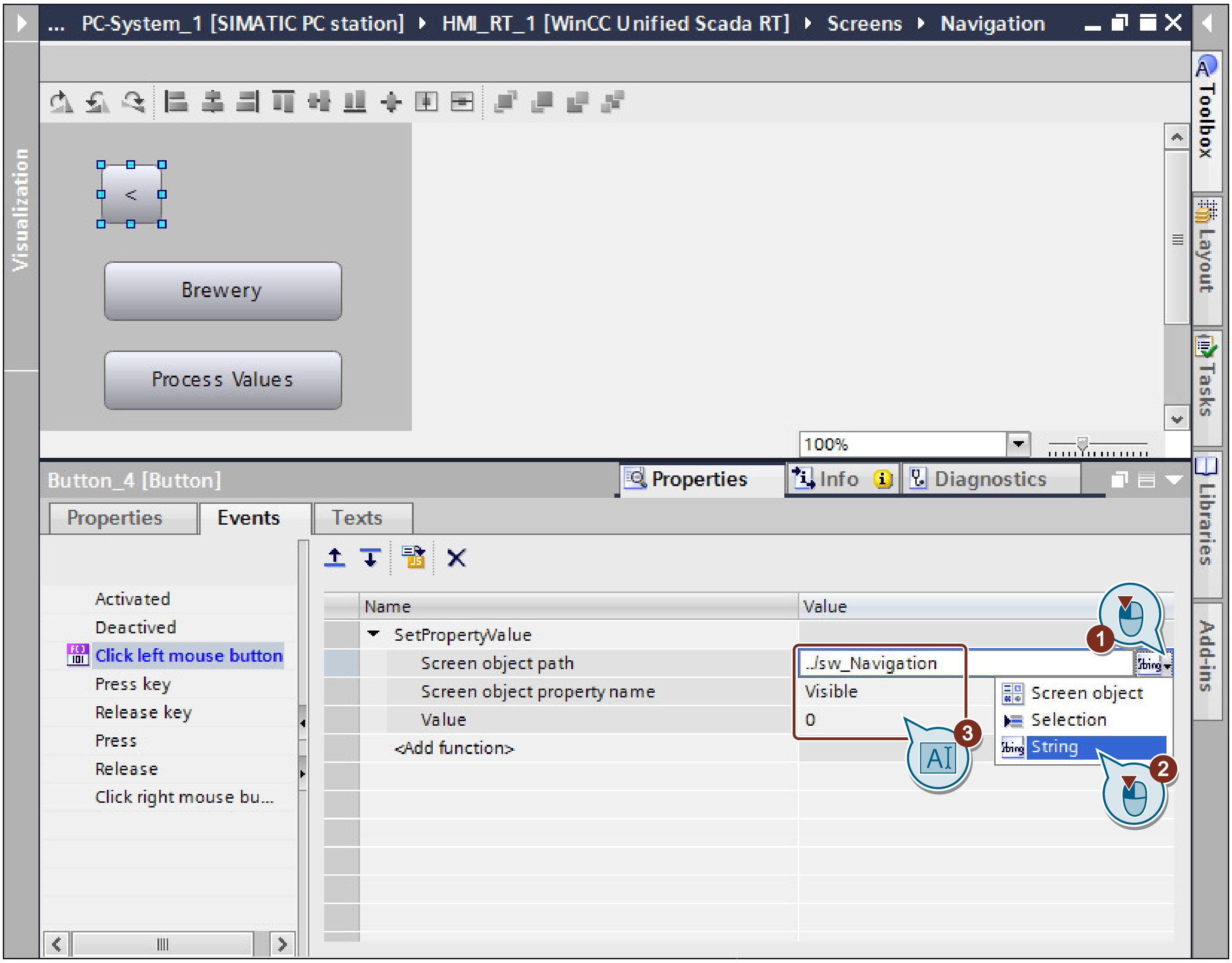


– Specify the parameters of the system function "SetPropertyValue".

Note

Upper and lower case

Pay attention to the correct case when setting the parameters.



... PC-System_1 [SIMATIC PC station] ▸ HMI_RT_1 [WinCC Unified Scada RT] ▸ Screens ▸ Navigation

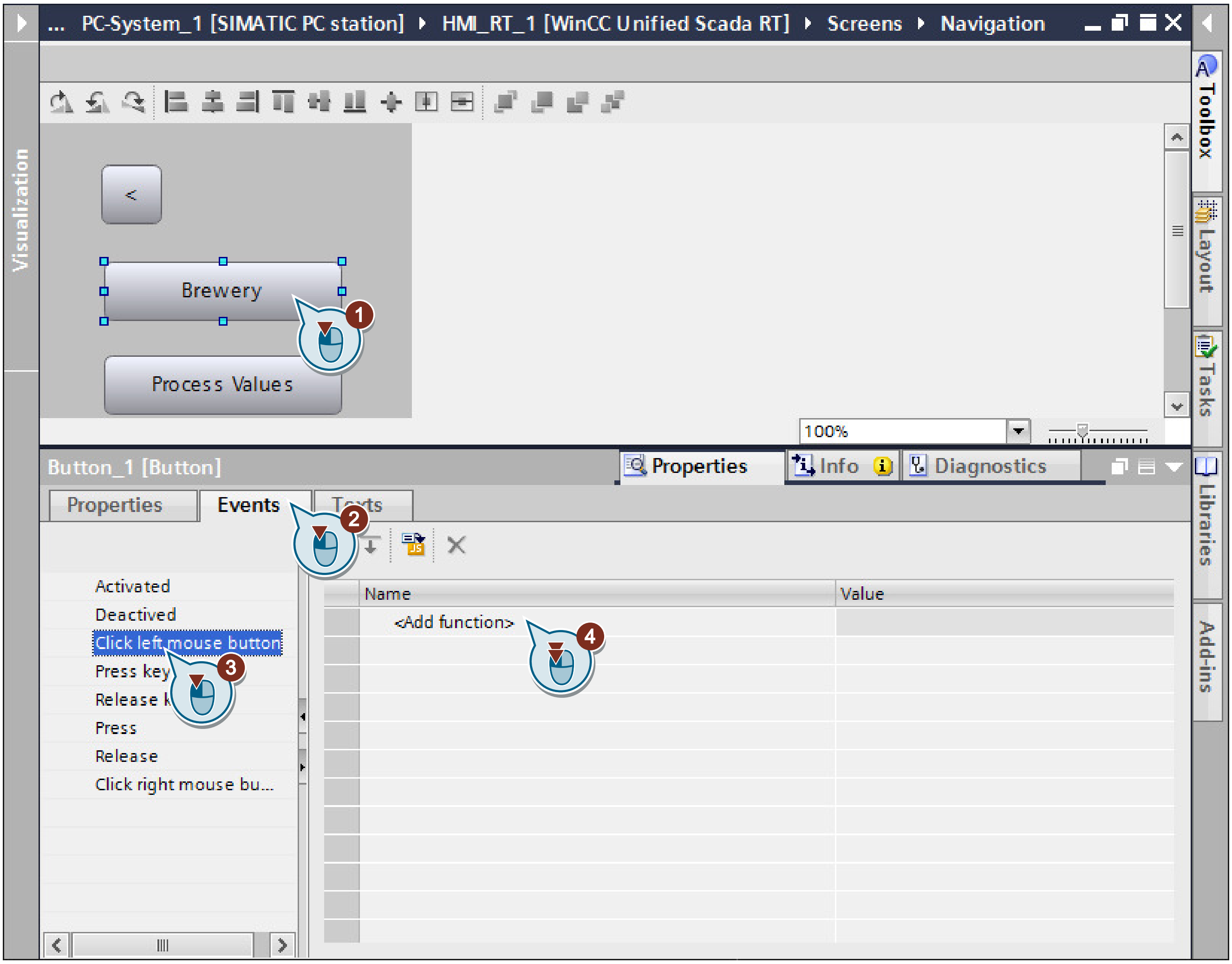
Button_4 [Button]

Properties | Events | Texts

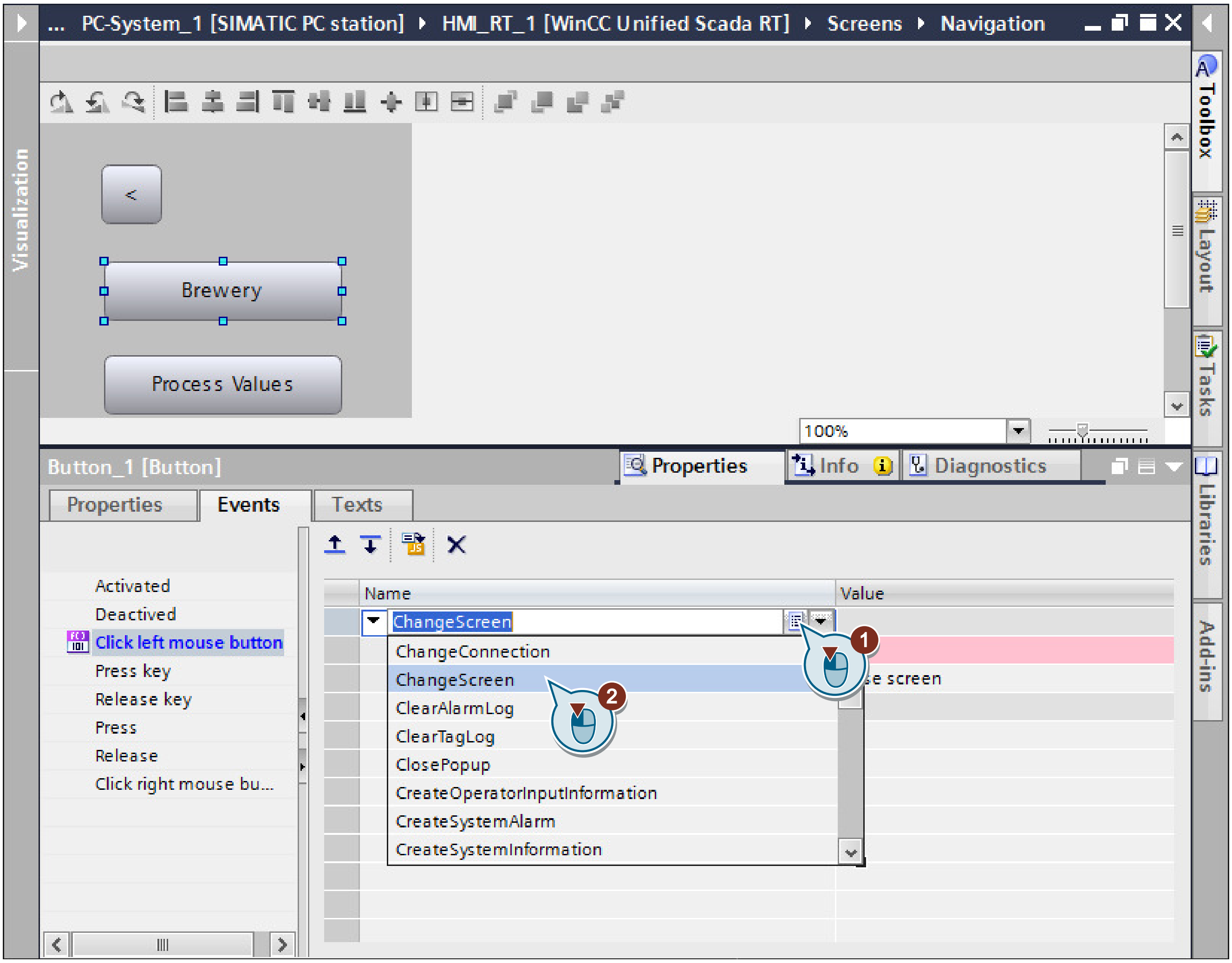
Events: Activated, Deactived, Click left mouse button, Press key, Release key, Press, Release, Click right mouse bu...

Name	Value
SetPropertyValue	
Screen object path	../sw_Navigation
Screen object property name	Visible
Value	0
<Add function>	

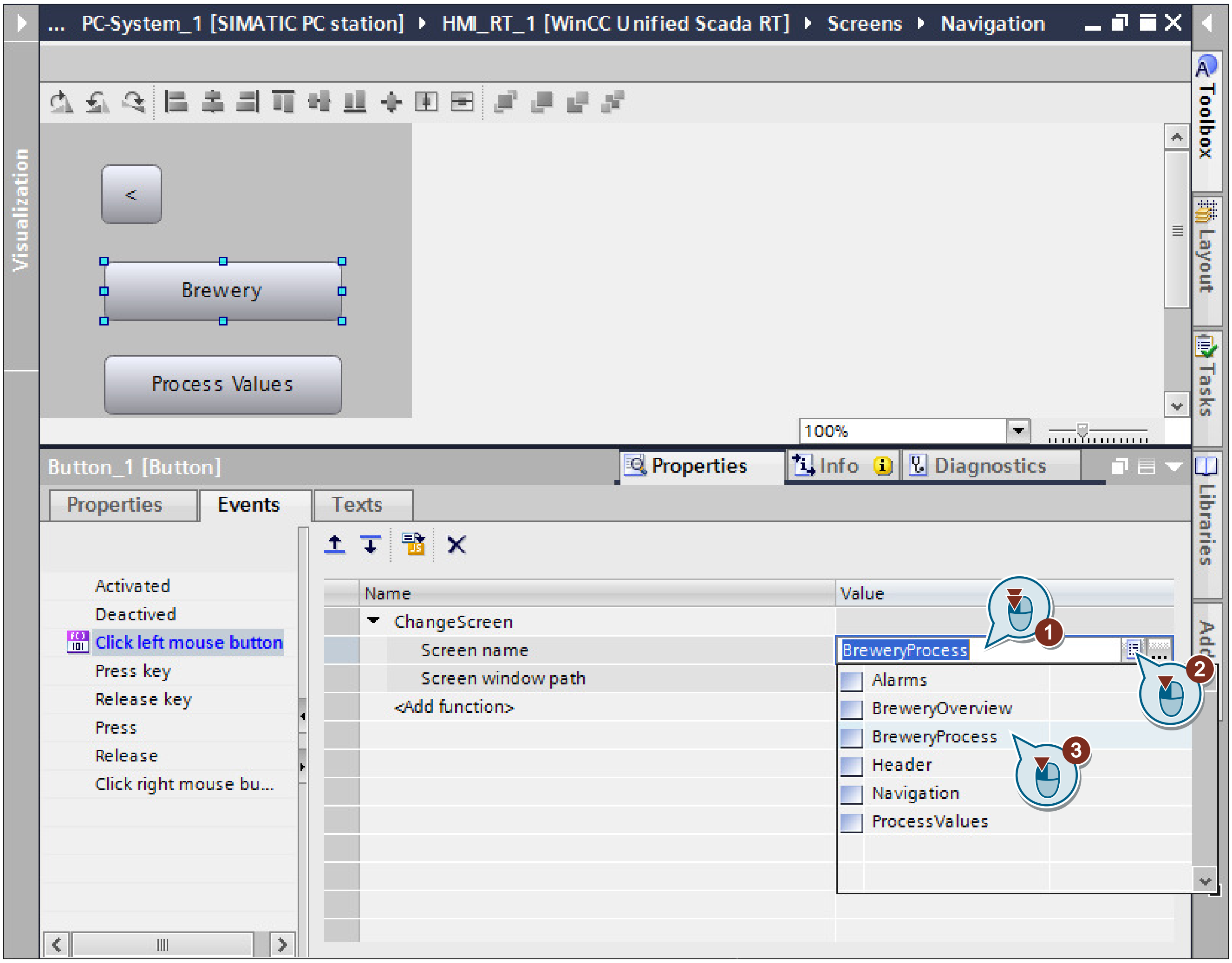
4. Configure the button "Brewery" which shows the brewing process in the screen window "sw_Content".
   - Insert a new event.



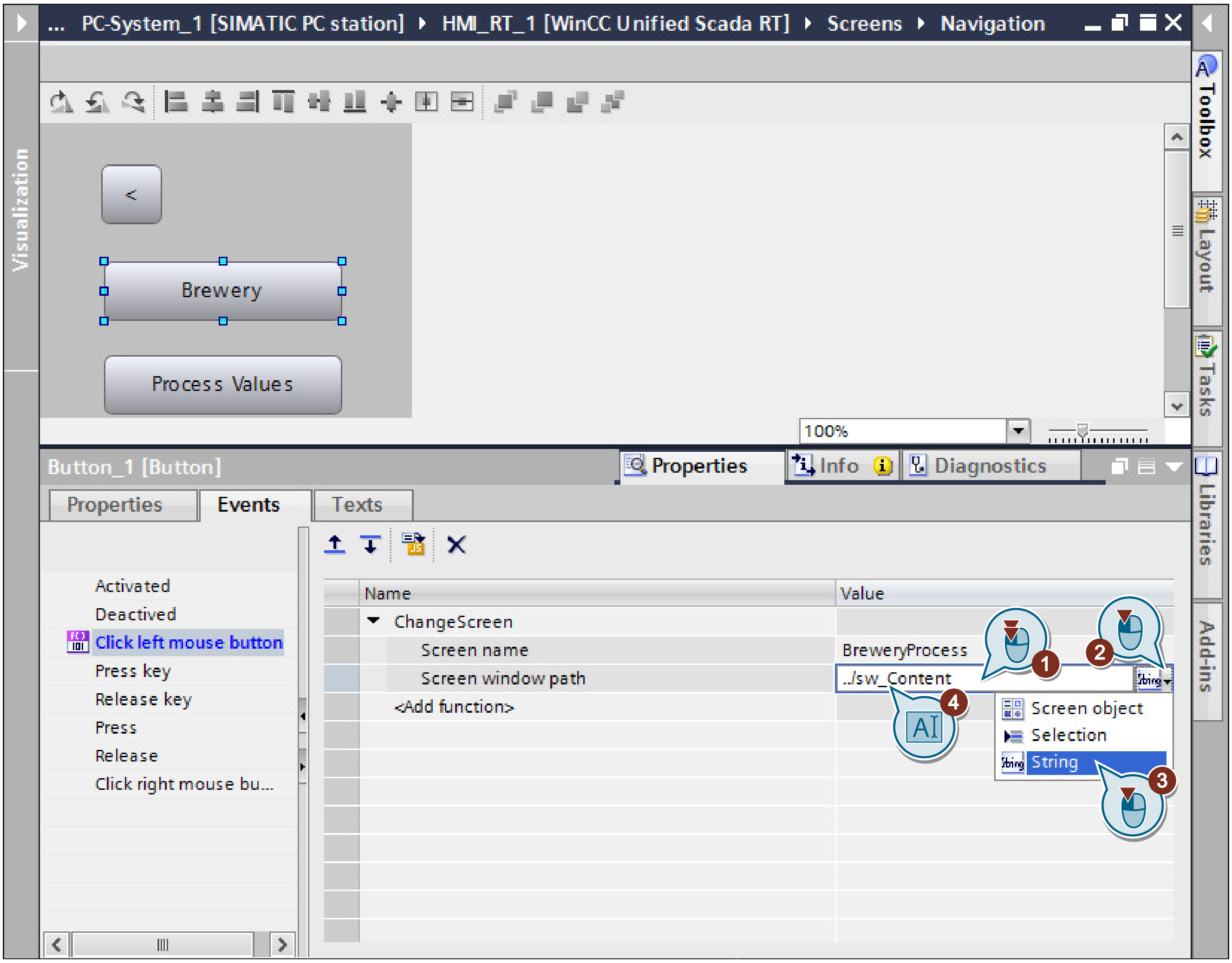
– Select the system function "ChangeScreen".



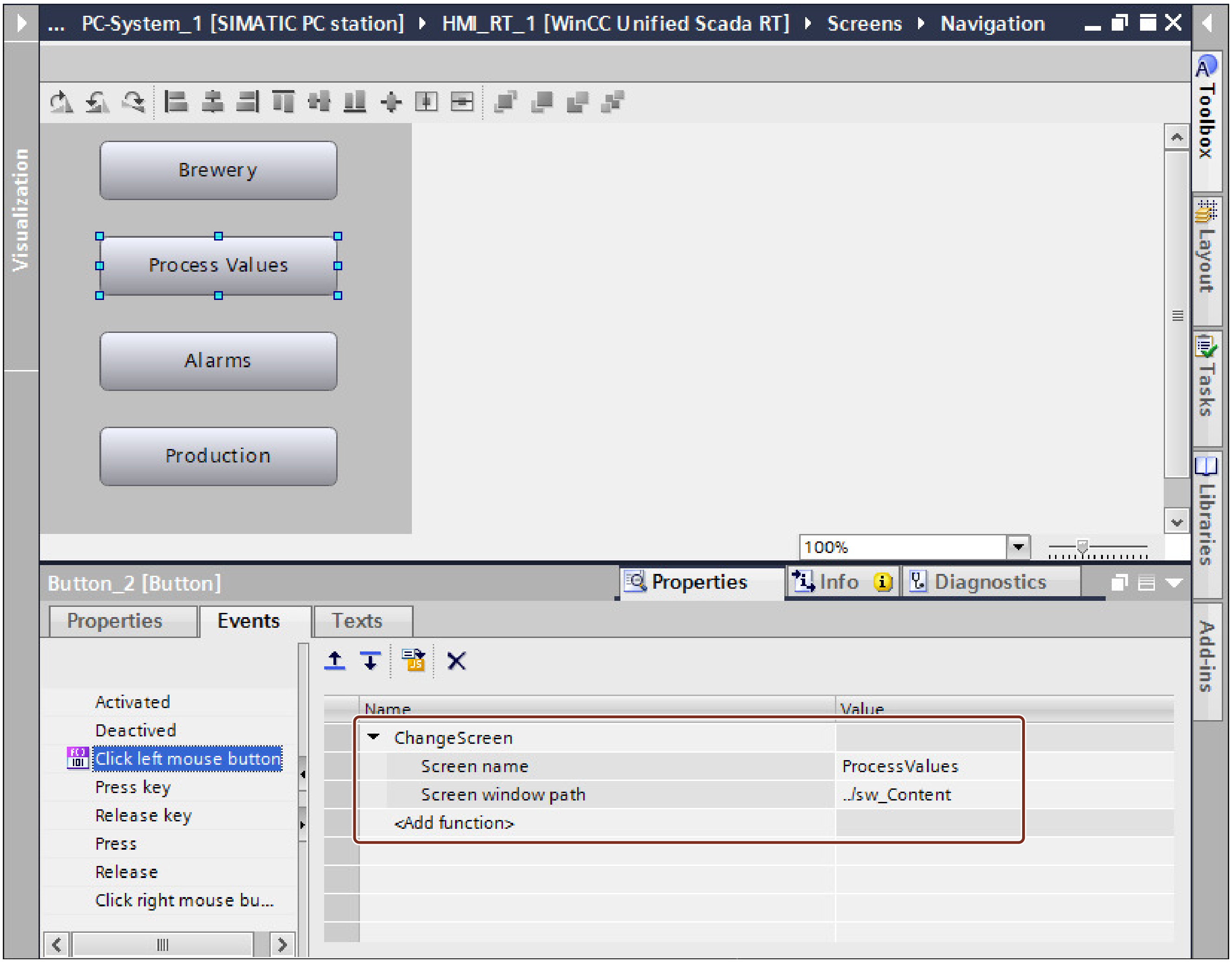
– Specify that the screen "BreweryProcess" is displayed in the screen window "sw_Content".



– Enter the relative path "../sw_Content" of the screen window "sw_Content".

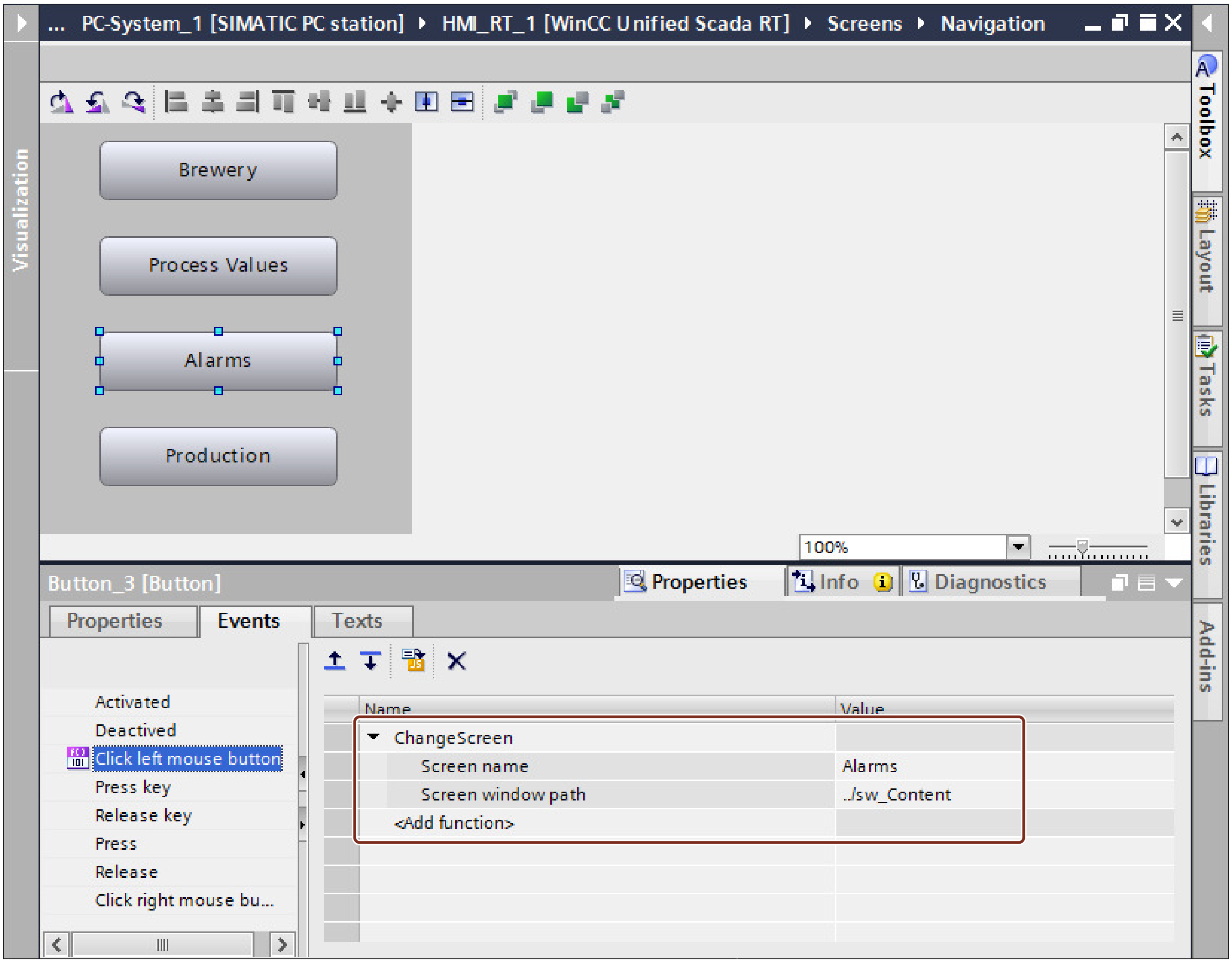


5. Configure the button "Process Values" in the same way as the button "Brewery" and specify the displayed parameters for the event "ChangeScreen".



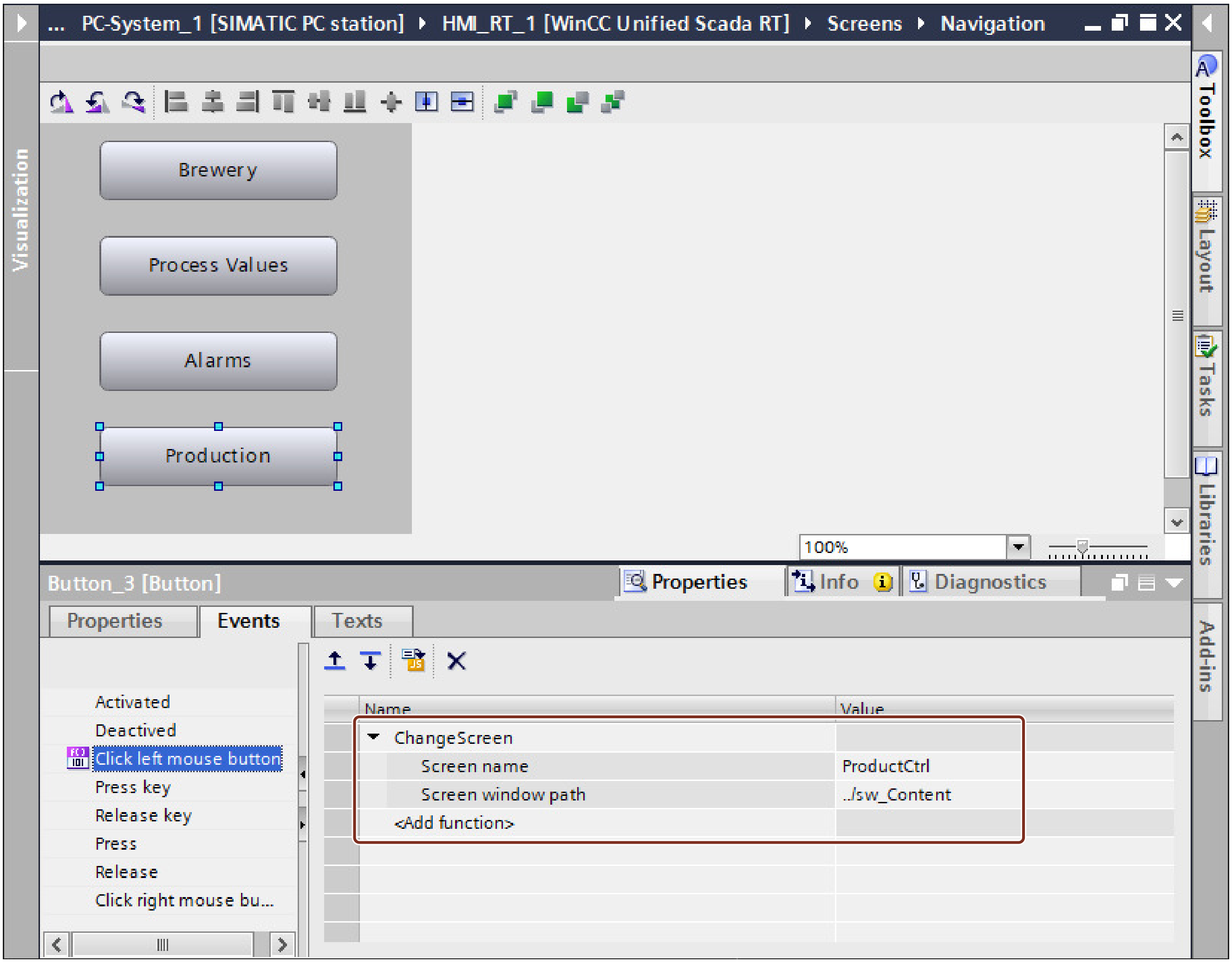
Name	Value
ChangeScreen	
Screen name	ProcessValues
Screen window path	../sw_Content
<Add function>	

6. Configure the button "Alarms" in the same way as the button "Brewery" and specify the displayed parameters for the event "ChangeScreen".



Name	Value
ChangeScreen	
Screen name	Alarms
Screen window path	../sw_Content
<Add function>	

7. Configure the button "Production" in the same way as the button "Brewery" and specify the displayed parameters for the event "ChangeScreen".



Name	Value
ChangeScreen	
Screen name	ProductCtrl
Screen window path	../sw_Content
<Add function>	

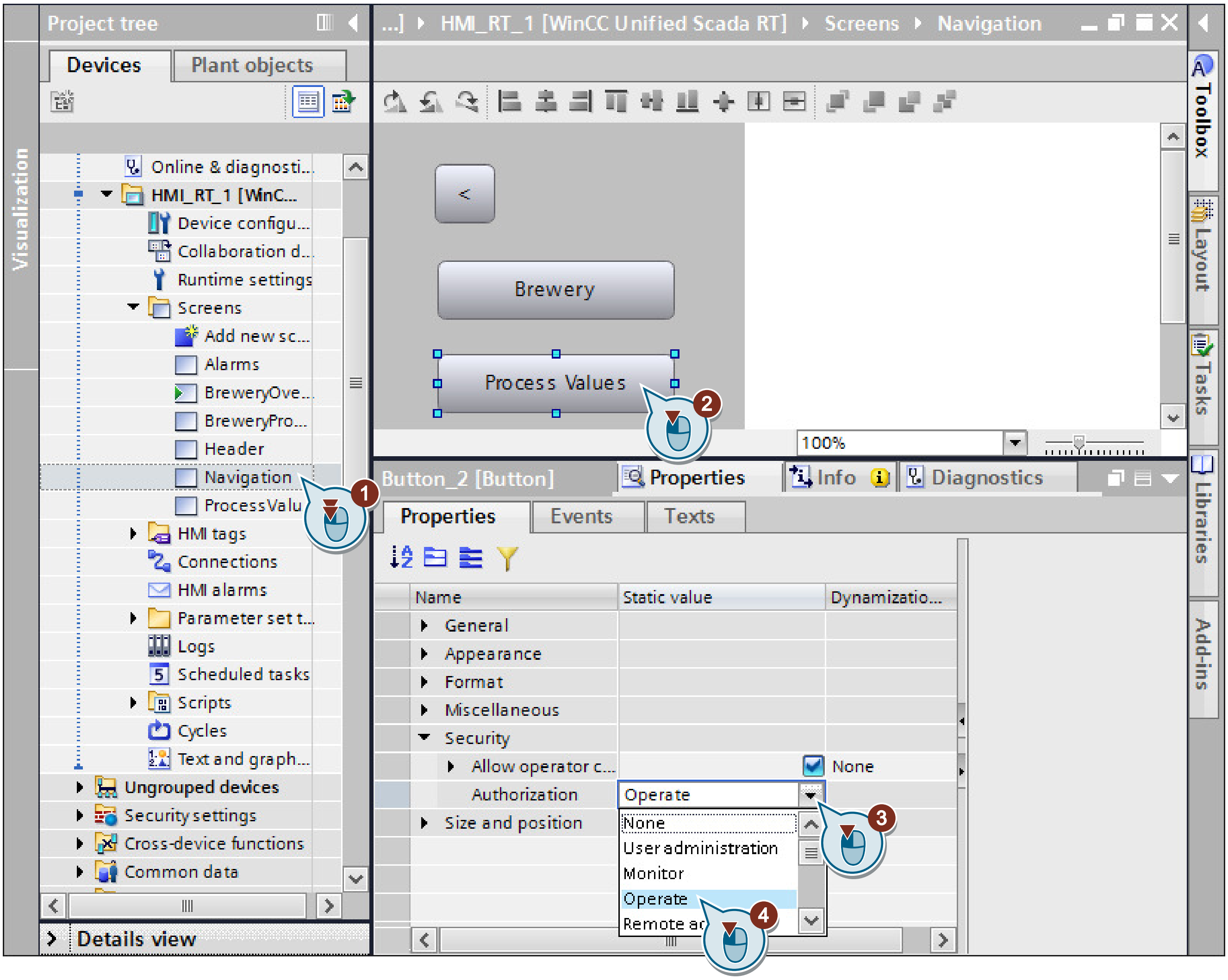
6.6 Configuring a button with access protection

Introduction

Employees of the user group "Worker" should not access the process values. To ensure this, configure an appropriate access protection for the button for calling the process values.

Procedure

1. Open the "Navigation" screen.
2. Select the "Process Values" button and assign the "Operate" access right to the button.

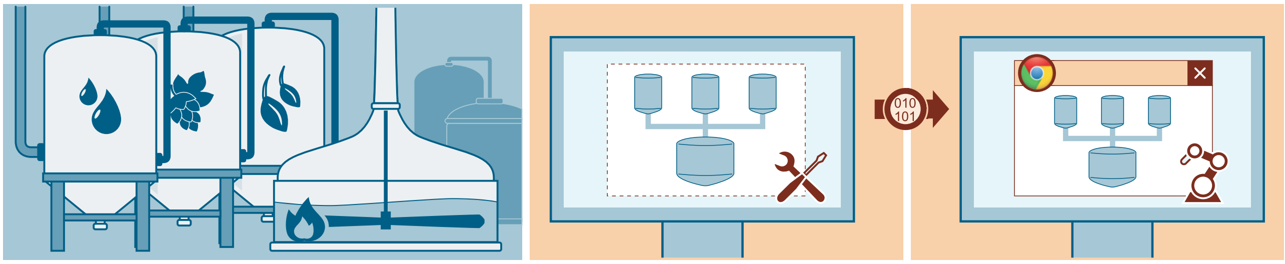


Runtime 7

7.1 Function test in Runtime

Basics

The environment for operating and monitoring the brewing process is called "Runtime". Runtime runs in the browser independently of the HMI device.



In this Getting Started, Runtime is run on the PC on which you configured it. For runtime to be started in the browser, the user logged on to Windows must have local admin rights.

7.2 Installing certificates

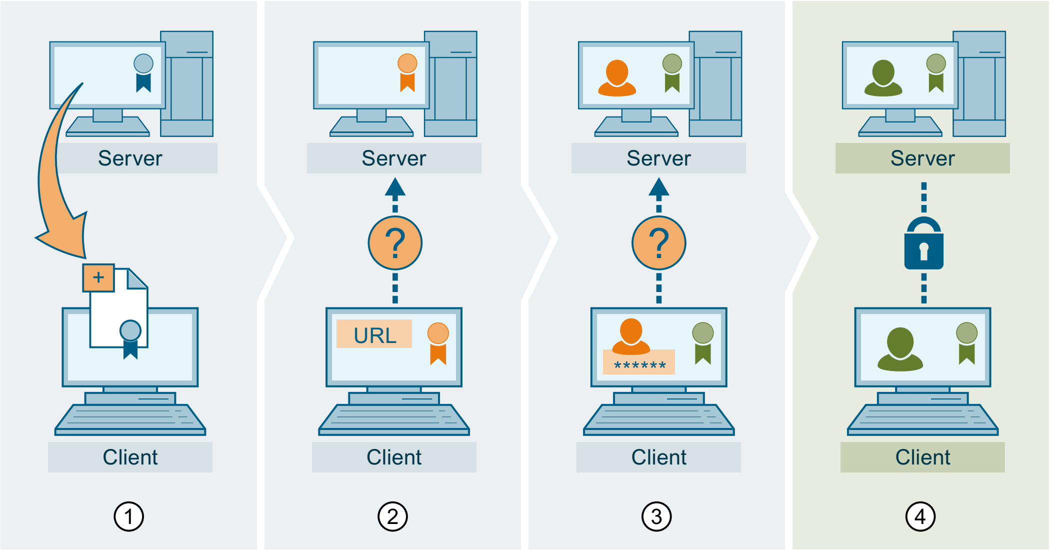
Introduction

Certificates are used to securely connect end devices to WinCC Unified Runtime. Certificates fulfill the following tasks:

- Confirm the identity of the communication partner
- Encrypt data exchange

Operating principle

With the WinCC Unified Configuration, you can transfer the certificate to the client PCs that access Runtime ①. When the operator enters the URL of the Runtime server, the identity of the client is checked using the certificate ②. After the check, the operator authenticates himself or herself with a user name and password ③. The Runtime server and client communicate encrypted using the HTTPS protocol ④.



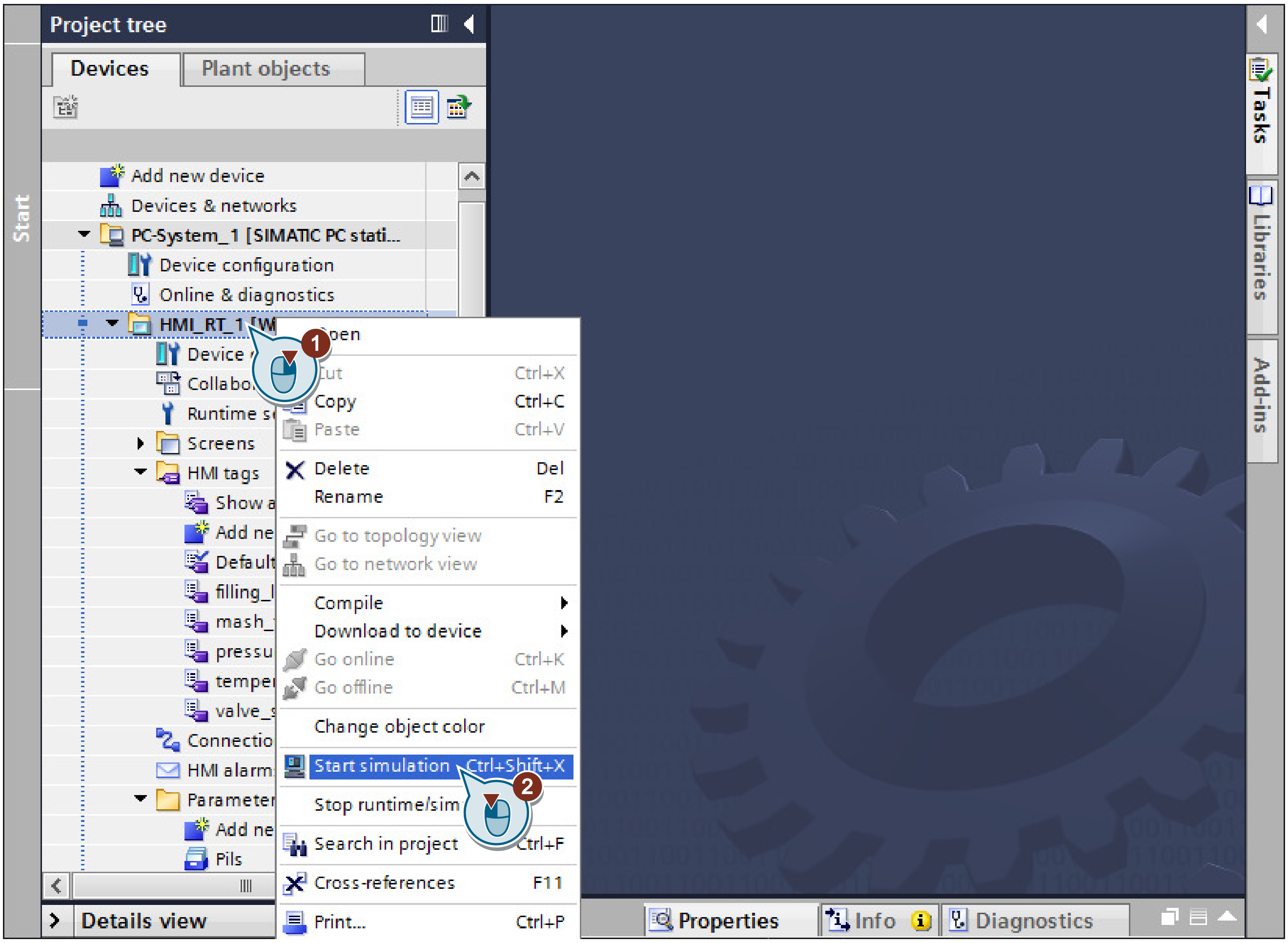
Setting up the certificate structure

WinCC Configuration Tool | FAQ (https://support.industry.siemens.com/cs/at/en/view/109777591)

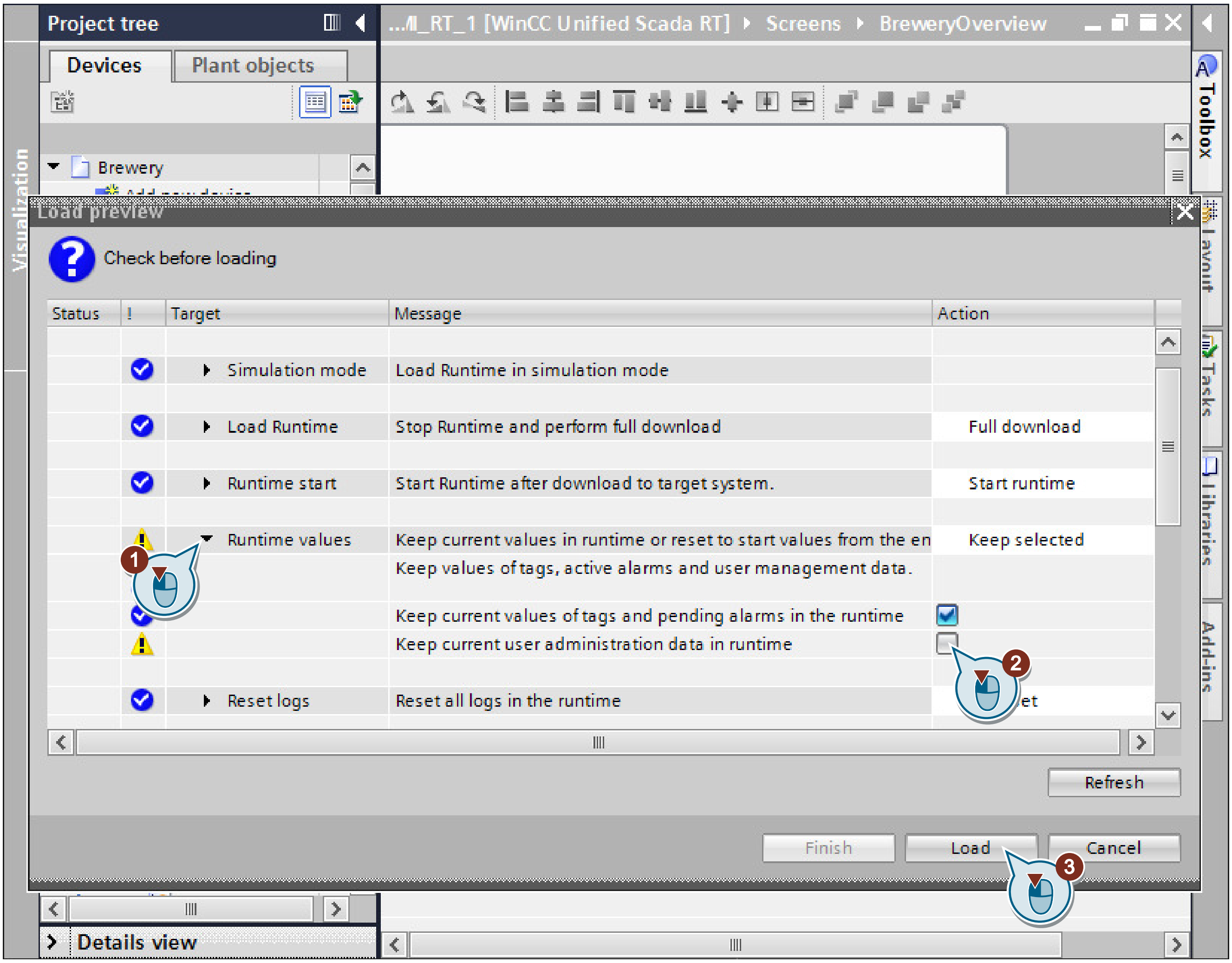
7.3 Testing the configuration in Runtime

Starting the simulation

1. First, start the simulation for the "BreweryControl" via the shortcut menu of the control. The "PLCSim" application is started automatically once the load process is complete.
2. Start the simulation for the HMI device.



3. Specify that the local user management is used and confirm the download of the configuration data to the HMI device.

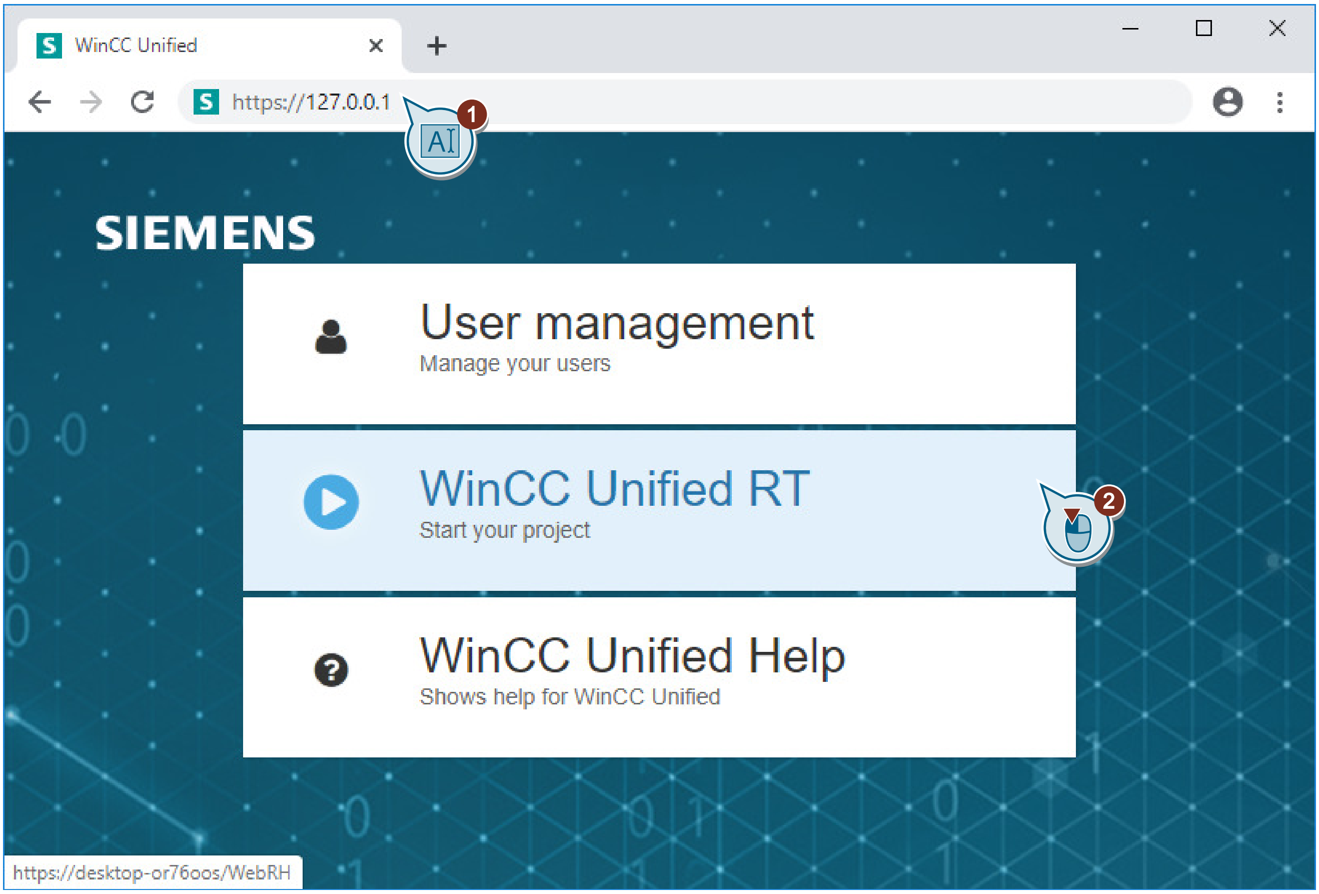


Load preview

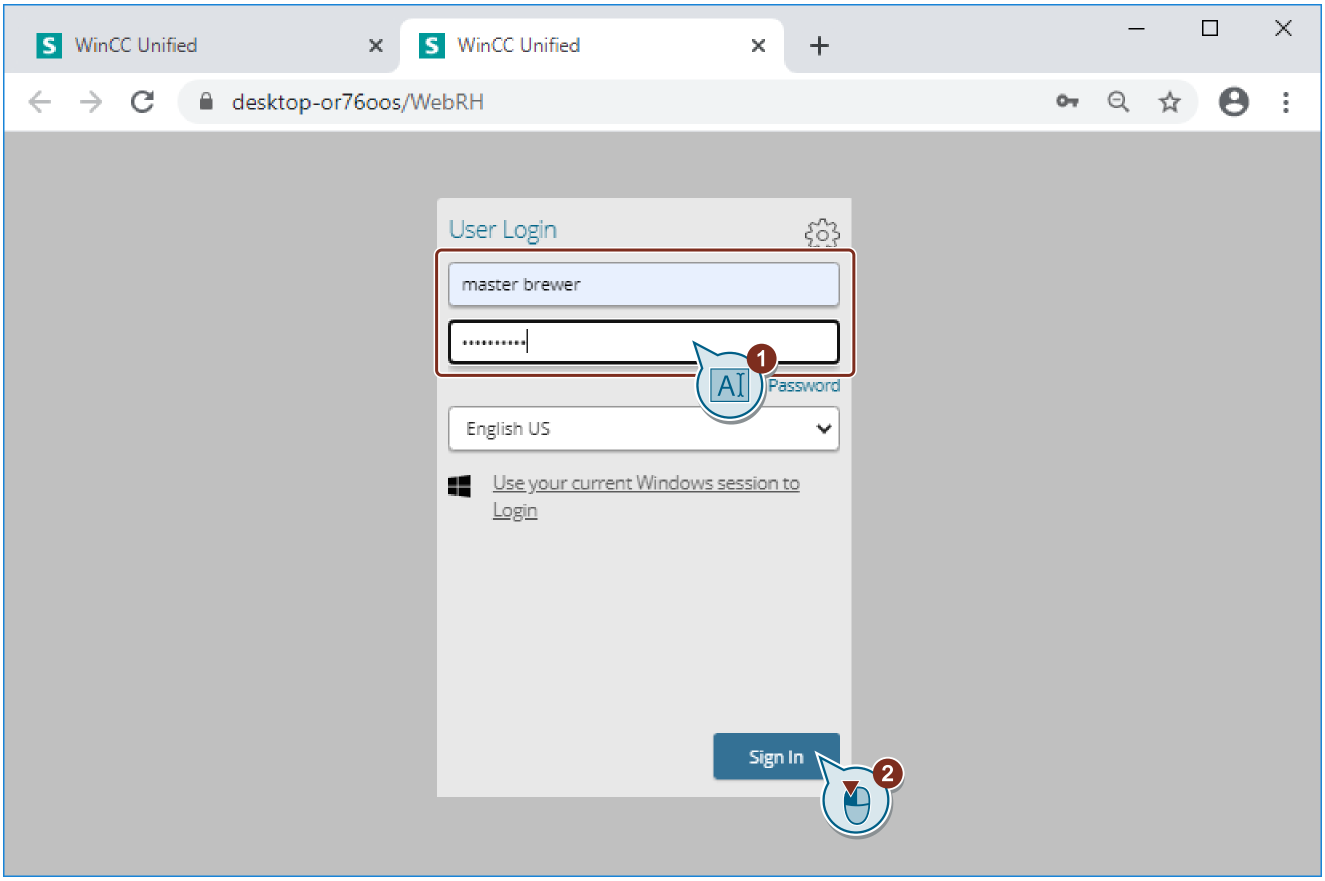
Check before loading

Status	!	Target	Message	Action
	✓	Simulation mode	Load Runtime in simulation mode	
	✓	Load Runtime	Stop Runtime and perform full download	Full download
	✓	Runtime start	Start Runtime after download to target system.	Start runtime
	⚠	Runtime values	Keep current values in runtime or reset to start values from the en	Keep selected
			Keep values of tags, active alarms and user management data.	
	✓		Keep current values of tags and pending alarms in the runtime	☑
	⚠		Keep current user administration data in runtime	□
	✓	Reset logs	Reset all logs in the runtime	et

4. Open the Chrome browser on the configuration computer and start Runtime.



5. Log on in Runtime.
   In this case, use the user "masterbrewer" which you created in the local user management.



master brewer

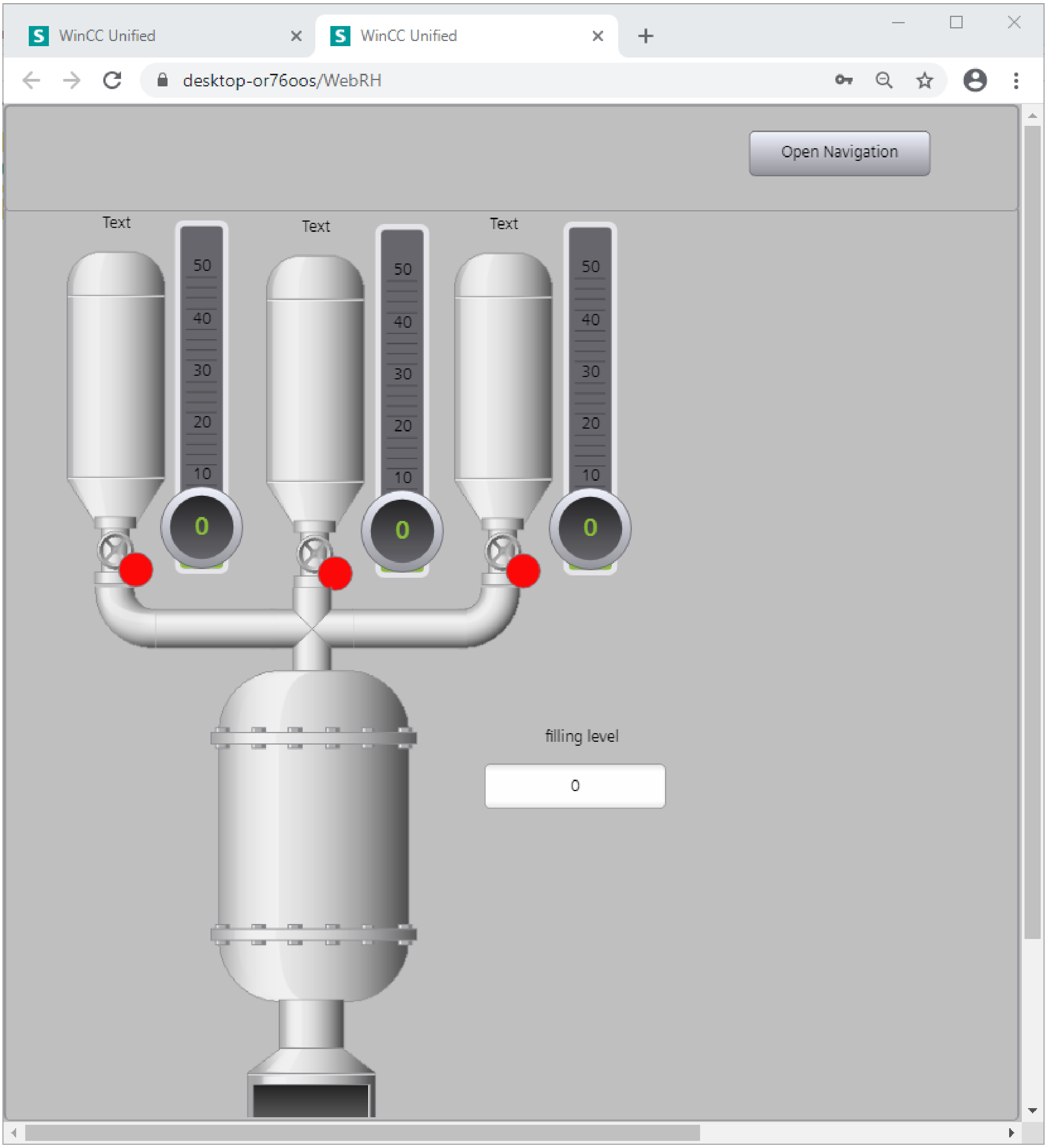
••••••••••

English US

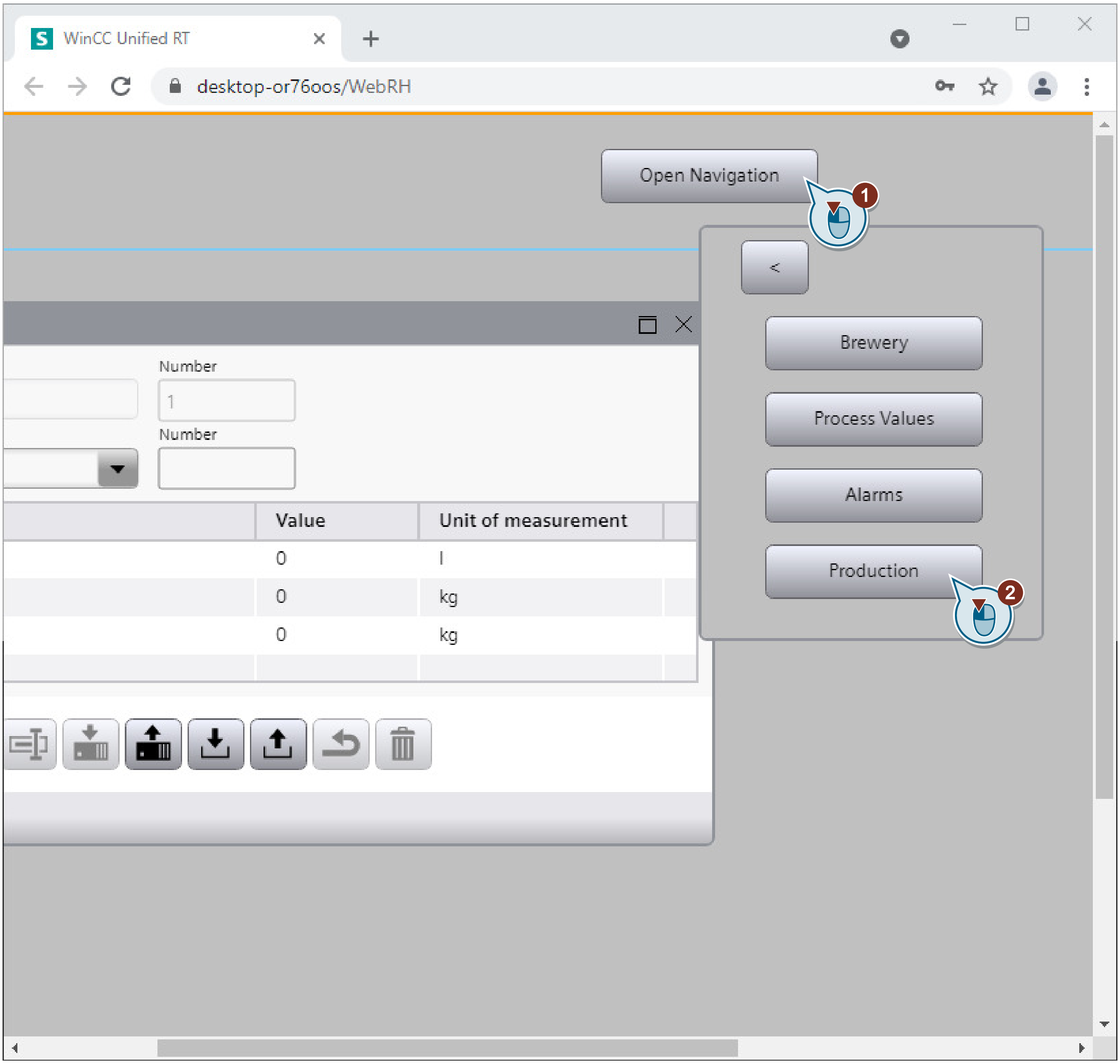
Use your current Windows session to Login

Testing the configuration

1. Check that all screen changes are working as intended.

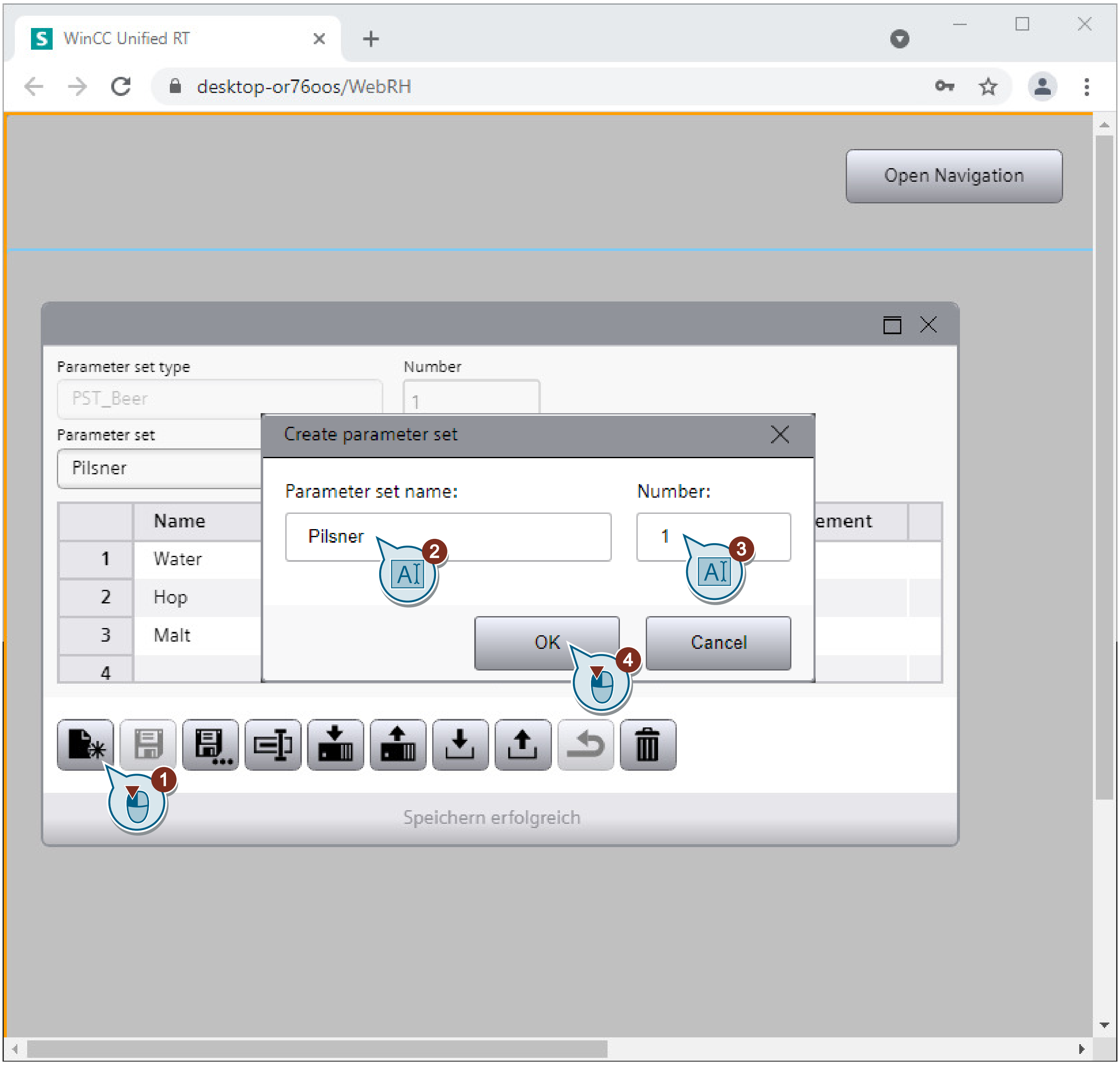


2. Navigate to the "ProductControl" screen.



	Value	Unit of measurement
	0	l
	0	kg
	0	kg

3. Create a parameter set for the "Pilsner" beer type.



Parameter set type: PST_Beer

Number: 1

Parameter set: Pilsner

	Name	ement
1	Water	
2	Hop	
3	Malt	
4		

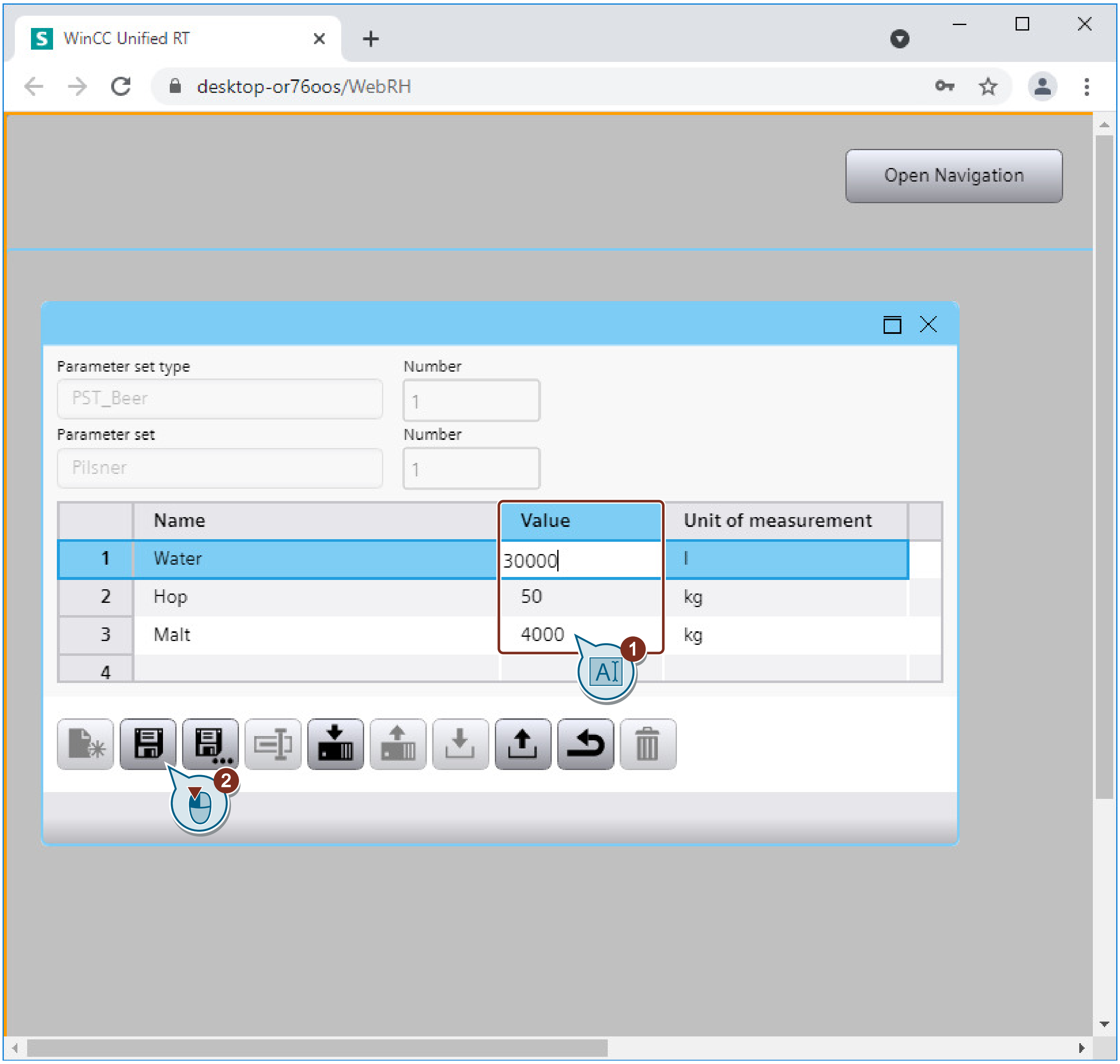
Create parameter set

Parameter set name: Pilsner

Number: 1

Speichern erfolgreich

4. Enter the desired quantities of ingredients and save the "Pilsner" parameter set.



Parameter set type: PST_Beer | Number: 1

Parameter set: Pilsner | Number: 1

	Name	Value	Unit of measurement
1	Water	30000	l
2	Hop	50	kg
3	Malt	4000	kg
4			

Congratulations!

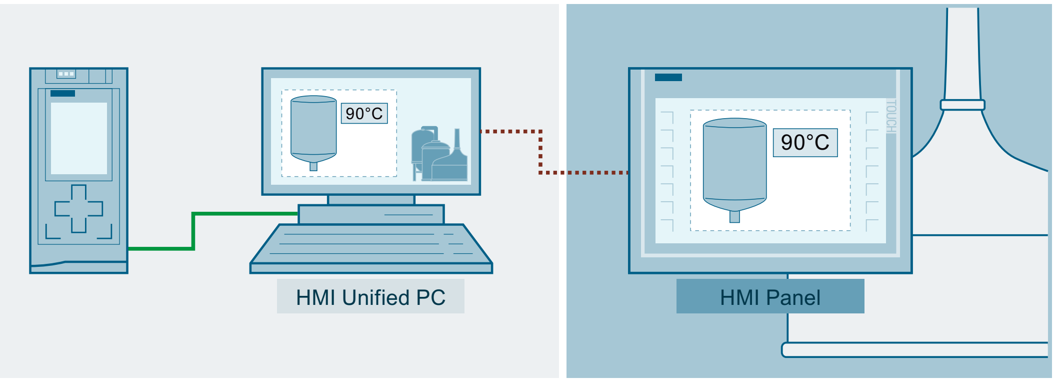
You have created your first project with WinCC Runtime Unified and successfully tested it in Runtime.

Additional functions 8

8.1 Runtime collaboration

Description

Runtime Collaboration refers to the exchange of data between several Unified Systems (Panels and PCs) during runtime.
Runtime Collaboration allows process pictures of other stations to be displayed and operated in a screen window. Thus, no double configuration effort is required.



To be able to transfer data, the collaboration settings must be specified in all collaborating systems. After the configuration, the Panel is shared for access. Access to the Panel is activated in the PC.

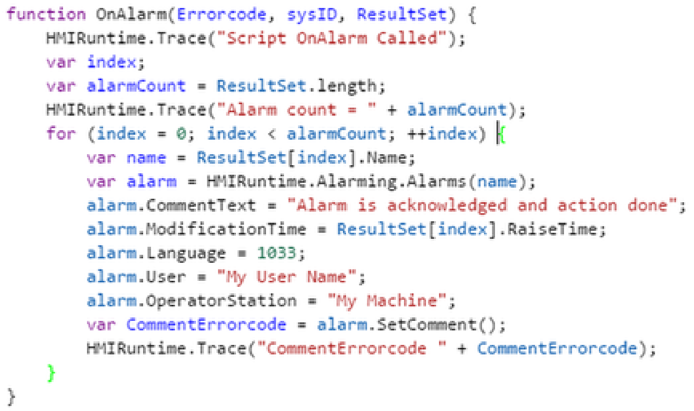
8.2 Scripting in WinCC

Introduction

In this Getting Started, you have configured functions at events such as "Activated" which, for example, trigger a screen change in Runtime. With this configuration, you have got to know a small part of the "scripting". WinCC Runtime Unified has a modern script environment that supports JavaScript as the programming language.

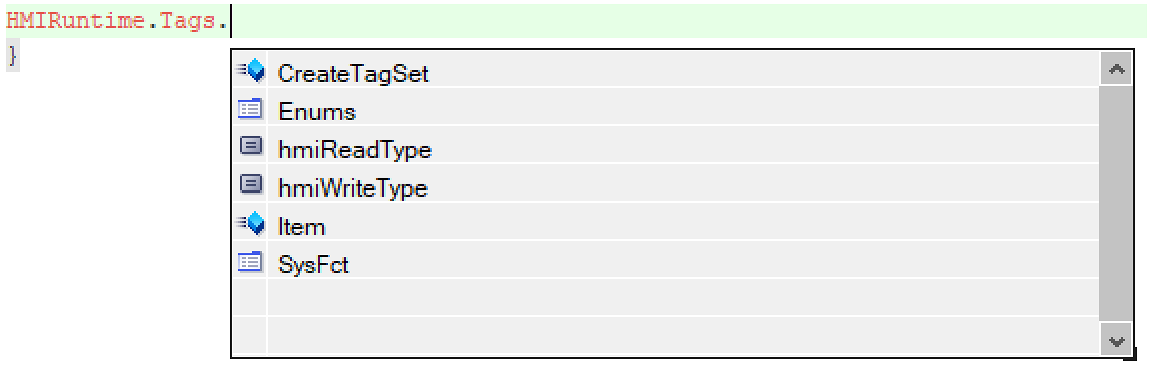
Programming environment

The programming environment is integrated into the familiar working environment and supports functions such as syntax highlighting:



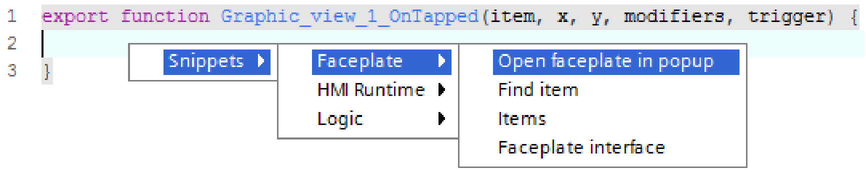
```
function OnAlarm(Errorcode, sysID, ResultSet) {
    HMIRuntime.Trace("Script OnAlarm Called");
    var index;
    var alarmCount = ResultSet.length;
    HMIRuntime.Trace("Alarm count = " + alarmCount);
    for (index = 0; index < alarmCount; ++index) {
        var name = ResultSet[index].Name;
        var alarm = HMIRuntime.Alarming.Alarms(name);
        alarm.CommentText = "Alarm is acknowledged and action done";
        alarm.ModificationTime = ResultSet[index].RaiseTime;
        alarm.Language = 1033;
        alarm.User = "My User Name";
        alarm.OperatorStation = "My Machine";
        var CommentErrorcode = alarm.SetComment();
        HMIRuntime.Trace("CommentErrorcode " + CommentErrorcode);
    }
}
```

JavaScript is based on an object model that you navigate through in the editor with IntelliSense:



```
HMIRuntime.Tags.
}
```

Code snippets make programming easier by providing frequently used instruction patterns:



```
export function Graphic_view_1_OnTapped(item, x, y, modifiers, trigger) {

}
```

Concepts

Scripting supports global and local scripts:

- You program global scripts centrally in the project. For example, you can call global scripts in local scripts, such as for unit of measure conversions.
- You program local scripts at the respective point of use, e.g. color change at an object.

You can use predefined system functions for frequently required programming tasks, for example to change values or to exit Runtime.

Typical application examples

In this Getting Started, you have used the following JavaScript functions:

- Code Snippets (Page 47)
- System functions (Page 153)

Other typical application examples of JavaScript are:

- Reading and writing of tags
- Changing object properties depending on other properties or values
- Access to alarms

Reference

You can find more information on the use of scripting in the SIOS Portal under entry ID 109758536.