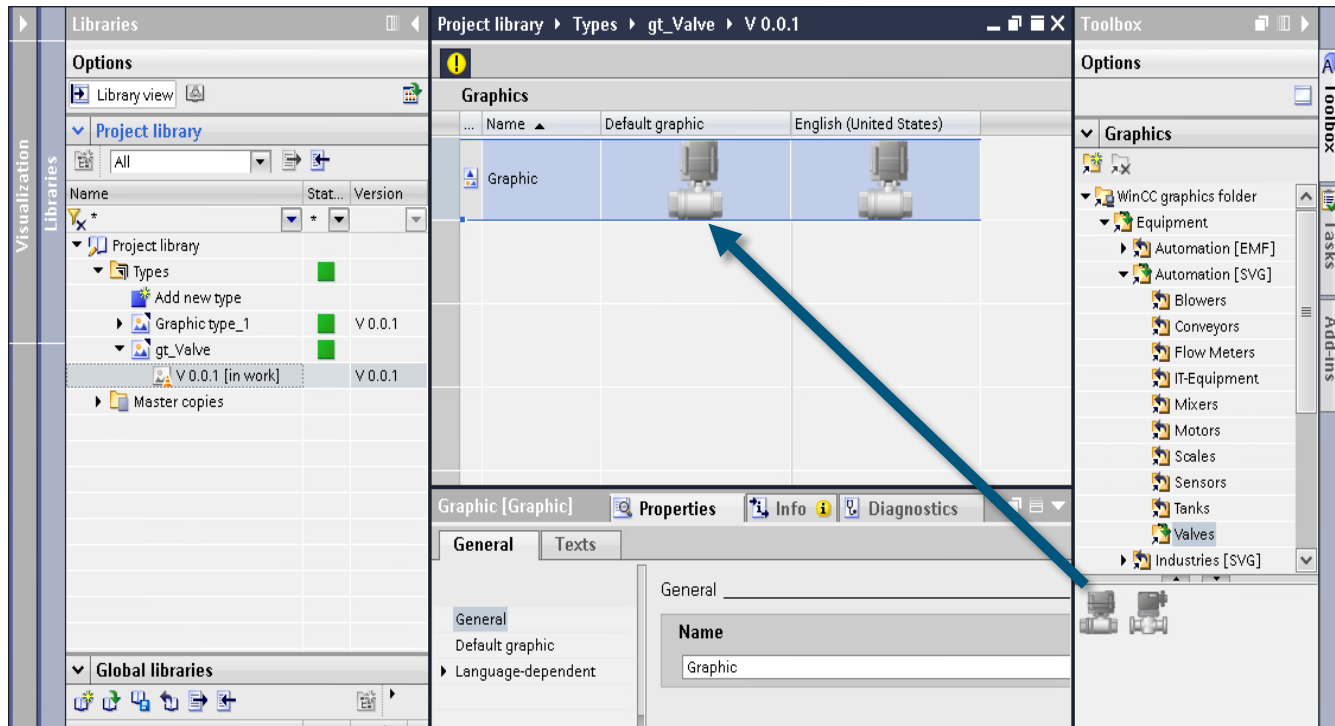




WinCC Unified Faceplates

Hands On: Create a Graphic type

H A N D S O N

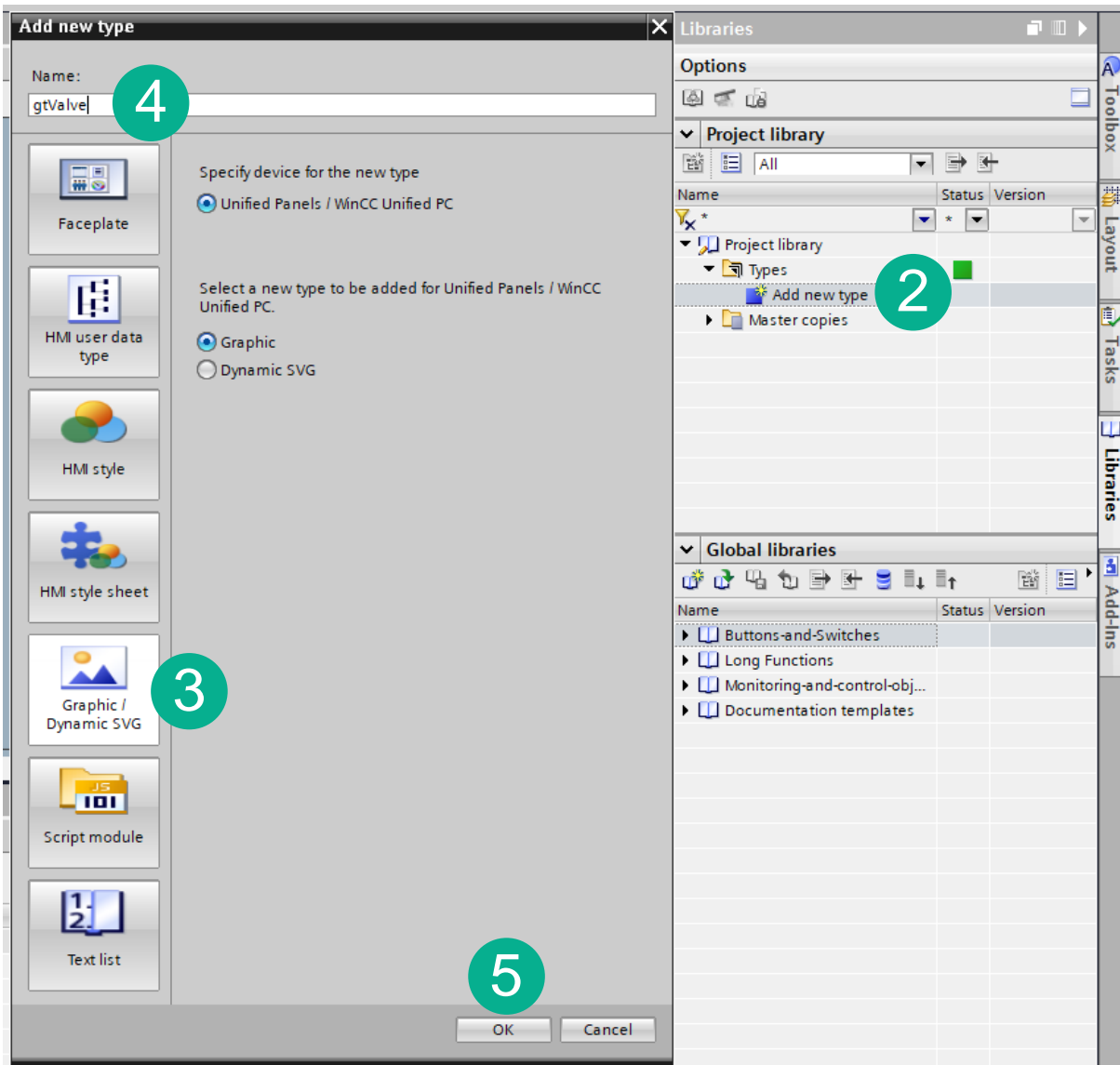


Create a new Graphic type with a valve graphic

1. Create the Graphic type “gtValve”
2. Use drag&drop to replace the Default graphic with a valve graphic (SVG) from the toolbox
3. Release the version

Hands On: Create a Graphic type

Content page – Create Type



1. Select Libraries from Task Card
2. Add a new Type to the Project Library
3. Select Graphic/Dynamic SVG
4. Rename to gtValve
5. Press OK

Hands On: Create a Graphic type

Content page – Add Graphic

The screenshot shows the Siemens WinCC Graphics Designer interface. At the top, a yellow warning banner contains a green circle with the number '3' and the text: "The 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, you can [check whether dependent types need to be adapted](#)." Below this, the main workspace is a grid with columns for 'Default graphic' and 'English (United States)'. A valve graphic is being dragged from the 'Valves' folder in the 'Graphics' toolbox (indicated by a green circle with '1') to the 'Default graphic' column (indicated by a green circle with '2'). At the bottom of the workspace, there are tabs for 'Properties', 'Info', and 'Diagnostics'. A green circle with the number '3' is also present in the top-left corner of the workspace area.

1. From the Graphics Toolbox, highlight Valves as shown
2. Drag the Valve from the details view and drop it on the Default Graphic column
3. Release the version

Hands On: Create a Graphic type Content page – Release version

Release type version

i 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:

Version:

Author:

Comment:

i 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 "default" version)

1

OK Cancel

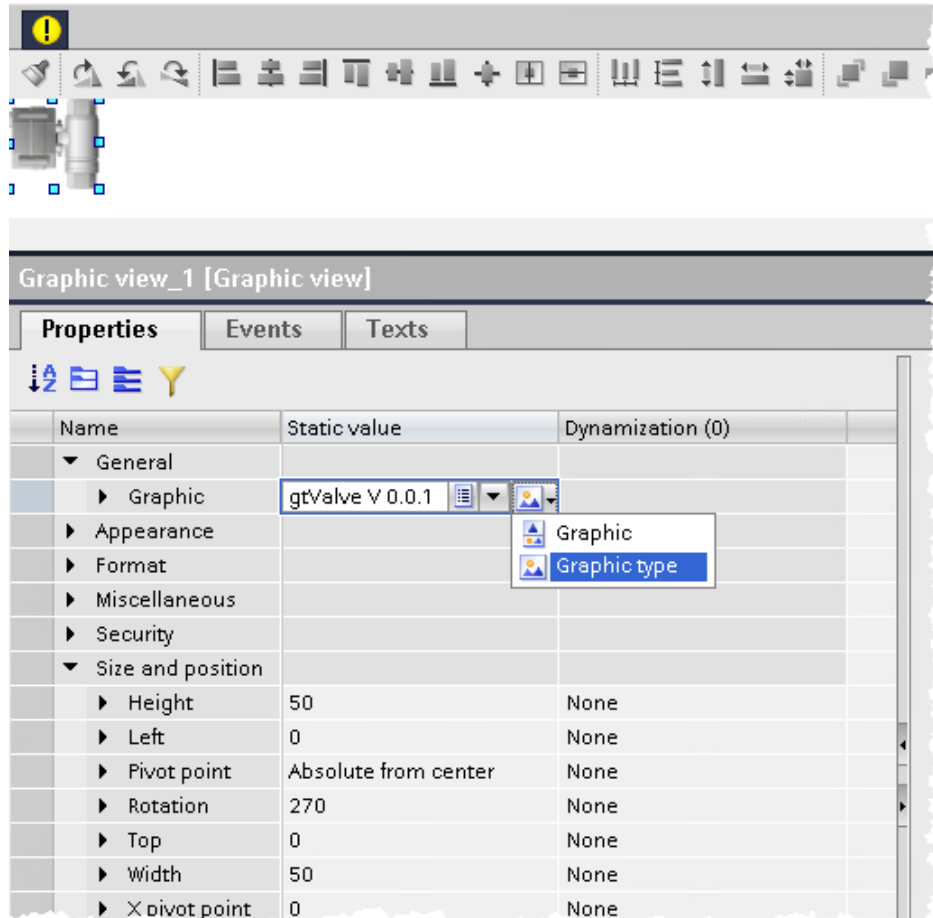
1. Press OK to release this version.

Note: It is a good idea to Delete unused type versions to keep the project from getting cluttered.

Note: You can change Major and Minor versions from this screen.

Hands On: Create Faceplate type and instance

H A N D S O N

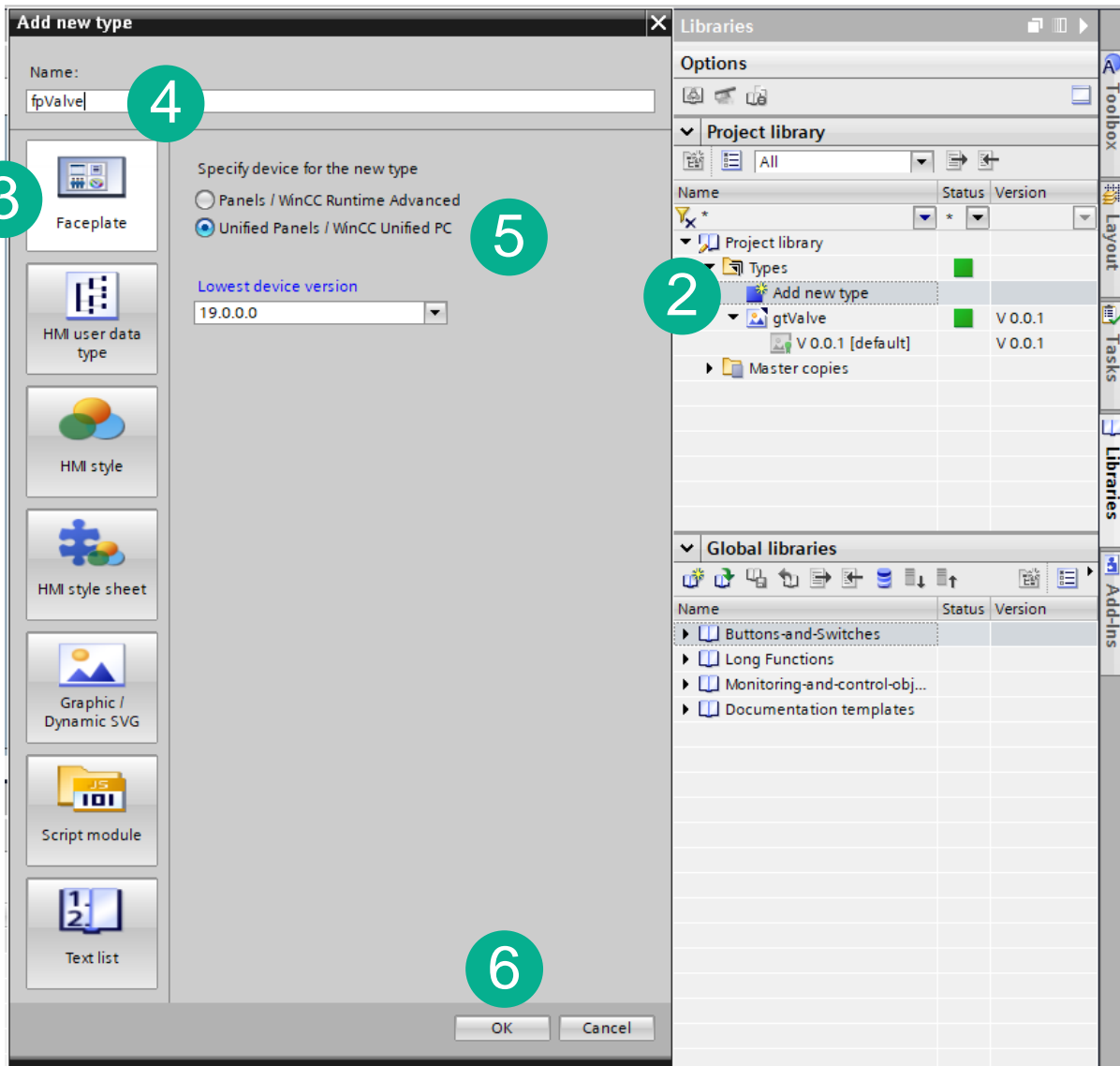


Goal: create a new Faceplate for a valve

2. Create new faceplate type “fpValve”
3. Change size 250 x 150 pixels
4. Change Background fill pattern to “Transparent”
5. In the Visualization tab add a Graphic view:
connect it to Graphic type “gtValve”
size 50 x 50 pixel
rotate by 270 degree
Change Background fill pattern to “Transparent”

Hands On: Create Faceplate type and instance

Content page – Hands On



1. Select Libraries from Task Card
2. Add a new Type to the Project Library
3. Select Faceplate
4. Rename to fpValve
5. Specify the Unified device type
6. Press OK

Hands On: Create Faceplate type and instance

Content page – Hands On

Project library ▸ Types ▸ fpValve ▸ V 0.0.1

! 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, you can

fpValve_V_0_0_1 [Faceplate type]

Properties Events Texts Expressions

📏 📏 📏 📏 📏 📏


Name	Static value	Dynamization (0)
▶ Appearance		
▶ Format		
▶ Miscellaneous		
▶ Security		
▼ Size and position		
▶ Size - height	250	None
▶ Size - width	150	None

1. Change screen size as shown

Hands On: Create Faceplate type and instance

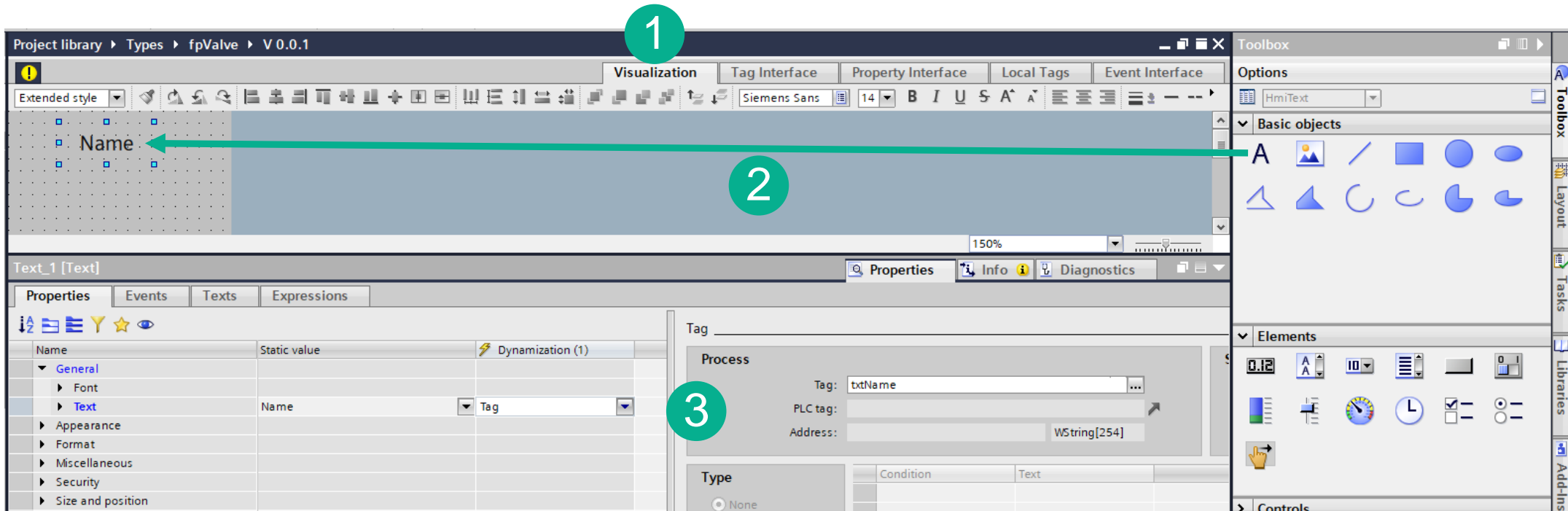
Content page – Hands On

Name	Data type	User data type structure
txtName	WString	
Status	Int	
RunPB	Bool	
<Add new>		

1. Press  to minimize info window
2. Select Tag Interface
3. Add tags as shown, **please note data types**

Hands On: Create Faceplate type and instance

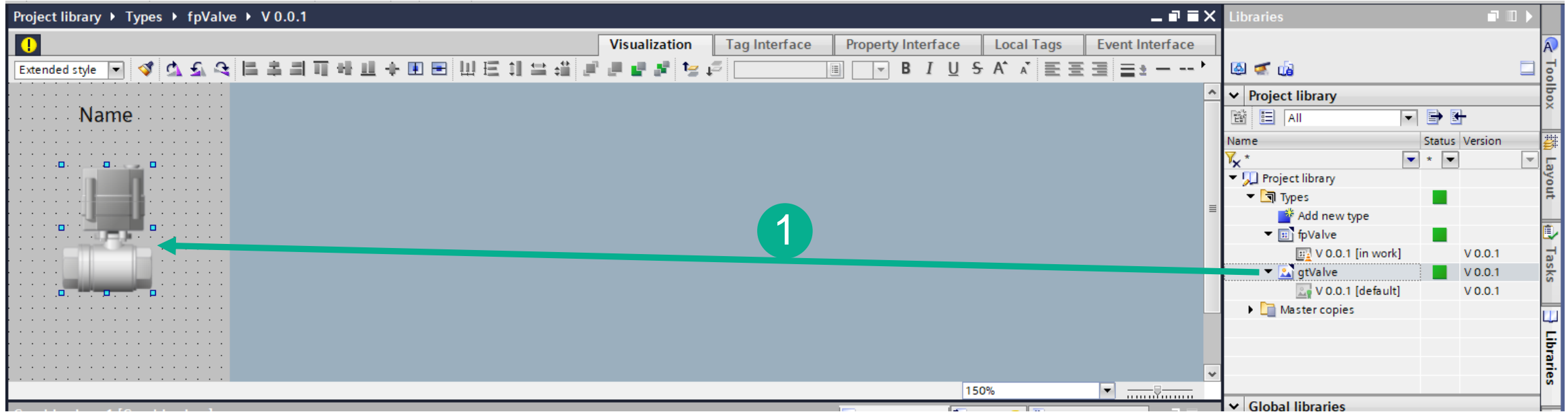
Content page – Hands On



1. Switch back to the Visualization tab
2. From the Basic Objects Toolbox, drag a Text Field to the faceplate
3. Change the Dynamization to Tag and assign the process tag to txtName

Hands On: Create Faceplate type and instance

Content page – Hands On



1. From the Project Library, drag the gtValve type to the faceplate as resize as shown

Hands On: Create Faceplate type and instance

Content page – Hands On

1. From the Project Library, drag a circle as resize as shown

2. Change Background Dynamization to Tag – Assign RunPB tag

3. Add a Type Range with (2) conditions

4. Condition 0 – Select More Colors, Transparency = 100%

5. Condition 1 – Green color

Property	Value	Unit
Appearance - style item	HmiCircle	
Background - alternative color	0, 255, 0	None
Background - color	200, 205, 215	Tag
Background - fill pattern	Solid	None
Border - alternative color	255, 255, 255	None
Border - color	125, 125, 133	None
Border - width	1	None
Fill direction	Bottom to top	None
Fill level	0	None
Fill level - show		None
Focus - show visual		None
Line - type	Solid	None
Opacity	1	None
Miscellaneous		
Security		

Condition	Background - color	Flash
0	200, 205, 215	No
1		No

Hands On: Create Faceplate type and instance

Content page – Hands On

Project library ▸ Types ▸ fpValve ▸ V 0.0.1

Visualization Tag Interface Property Interface Local Tags Event Interface

Extended style

Name

1

2

3

Circle_2 [Circle]

Properties Events Texts Expressions

Name	Static value	Dynamization (1)
Appearance		
Appearance - style item	HmiCircle	
Background - alternative color	0, 255, 0	None
Background - color	200, 205, 215	Tag
Background - fill pattern	Solid	None
Border - alternative color	255, 255, 255	None
Border - color	125, 125, 133	None
Border - width	1	None
Fill direction	Bottom to top	None
Fill level	0	None
Fill level - show		<input type="checkbox"/> None
Focus - show visual		<input checked="" type="checkbox"/> None
Line - type	Solid	None

Tag

Process

Tag: Status

PLC tag:

Address: Int

Type

None

Range

Multiple bits

Single bit 0

Condition	Background - color	Flashing	Alternating
0	200, 205, 215	No	255
1	0, 255, 0	No	255
2	255, 0, 0	No	255
<Add new>			

1. From the Project Library, drag a circle as resize as shown
2. Change Background Dynamization to Tag – Assign Status tag
3. Add a Type Range with (3) conditions, set conditions as shown

Hands On: Create Faceplate type and instance

Content page – Hands On

1. Add a Button from the Elements Toolbox, resize as shown

2. Change text as shown

1

2

Project library > Types > fpValve > V 0.0.1

Visualization Tag Interface Property Interface Local Tags Event Interface

Extended style

Name

Start/Stop

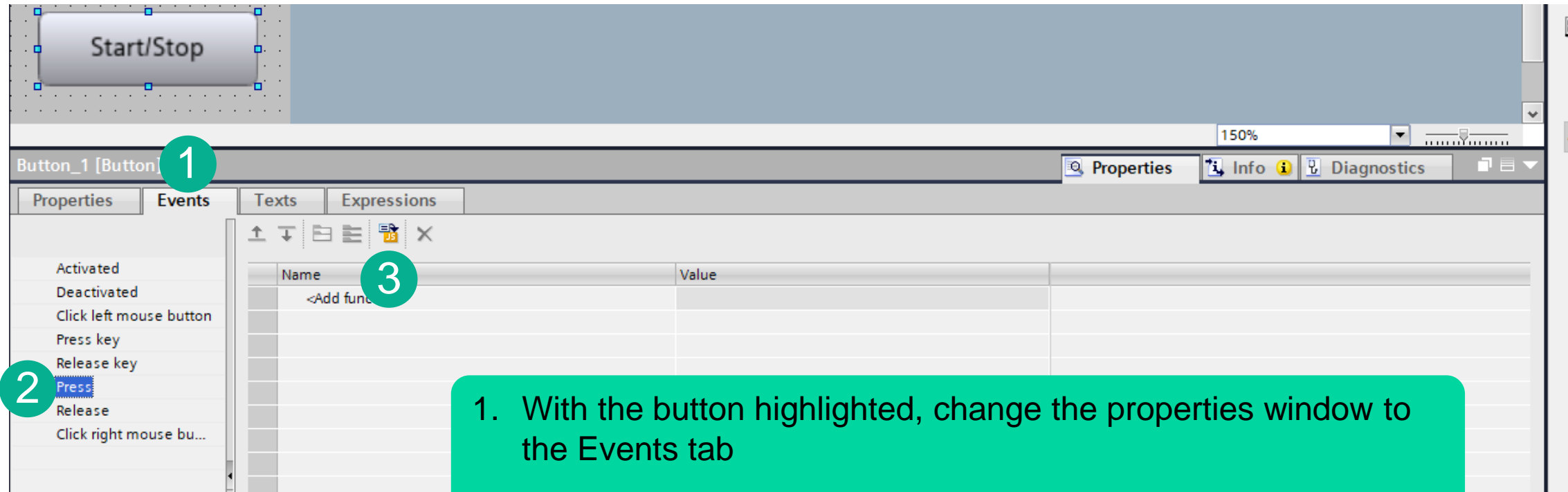
Button_1 [Button]

Properties Events Texts Expressions

Name	Static value	Dynamization (0)
General		
Content		
Graphic		None
Graphic - pressed button		None
Text	Start/Stop	None
Text - pressed button		None

Hands On: Create Faceplate type and instance

Content page – Hands On



1. With the button highlighted, change the properties window to the Events tab
2. Select the Press event
3. Press the Java Scripts button

Note: The Java Script view will allow for advanced scripting.

Hands On: Create Faceplate type and instance

Content page – Hands On

```
let tag1 = Tags("RunPB");           //Creates an object to represent RunPB
let tagValue1 = tag1.Read();        //Reads the value of RunPB
HMIRuntime.Trace("Value of MyTag1: " + tagValue1); //Enters the value of RunPB into the external Trace Viewer for diagnostics

if (tagValue1) {                    //If condition (RunPB = 1)
    tag1.Write(0);                   //Set RunPB = 0
    let screenItem = Faceplate.FindItem("Button_1"); //Defines the pushbutton as a screen object
    screenItem.Text = 'Run'          //Changes text on the button to Run
} else {                             //Else condition (RunPB = 0)
    tag1.Write(1);                   //Set RunPB = 1
    let screenItem = Faceplate.FindItem("Button_1"); //Defines the pushbutton as a screen object
    screenItem.Text = 'Stop'         //Changes text on the button to Stop
}
```

1. Copy this script, it will be used on the next page

Hands On: Create Faceplate type and instance

Content page – Hands On

Button_1 [Button] Properties Info Diagnostics

Properties Events Texts Expressions

Global definition Synchronous

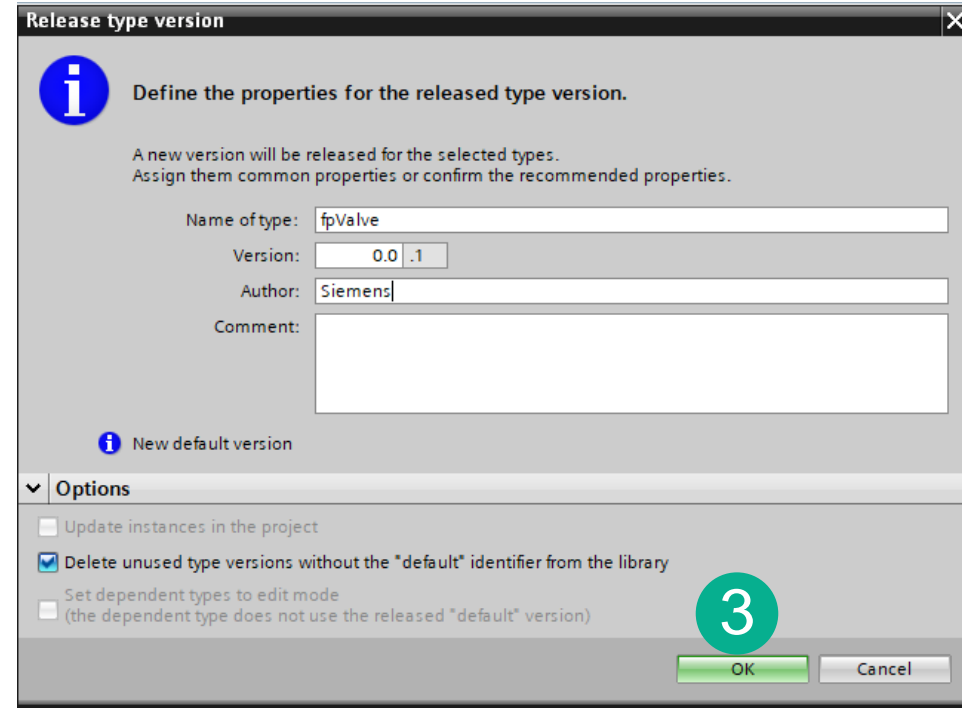
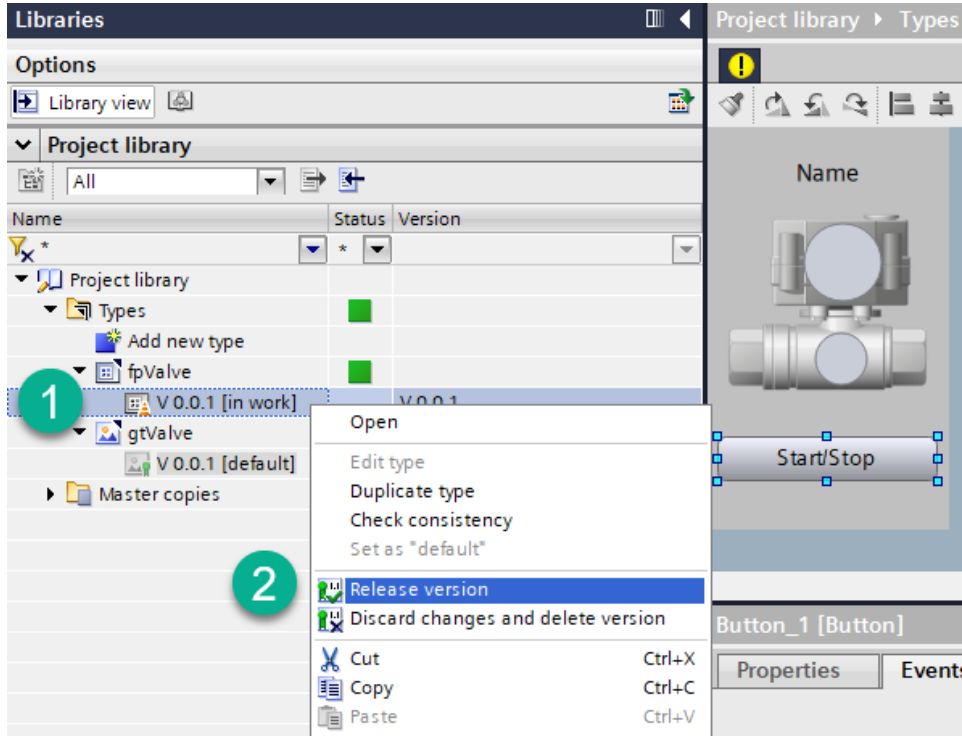
```
1 export function Button_1_OnDown(item, x, y, modifiers, trigger) {
2   let tag1 = Tags("RunPB"); //Creates an object to represent RunPB
3   let tagValue1 = tag1.Read(); //Reads the value of RunPB
4   HMIRuntime.Trace("Value of MyTag1: " + tagValue1); //Enters the value of RunPB into the external Trace Viewer for diagnostics
5
6   if (tagValue1) { //If condition (RunPB = 1)
7     tag1.Write(0); //Set RunPB = 0
8     let screenItem = Faceplate.FindItem("Button_1"); //Defines the pushbutton as a screen object
9     screenItem.Text = 'Run' //Changes text on the button to Run
10  }else { //Else condition (RunPB = 0)
11    tag1.Write(1); //Set RunPB = 1
12    let screenItem = Faceplate.FindItem("Button_1"); //Defines the pushbutton as a screen object
13    screenItem.Text = 'Stop' //Changes text on the button to Stop
14  }
15 }
```

Activated
Deactivated
Click left mouse button
Press key
Release key
Press
Release
Click right mouse bu...

1. Paste the script from the previous slide here
2. Press the syntax check button to check for errors

Hands On: Create Faceplate type and instance

Content page – Hands On



1. Right click on Faceplate Type
2. Select 'Release Version'
3. Select OK

Hands On: Create Faceplate type and instance

Content page – Hands On

The screenshot displays the Siemens SIMATIC Manager interface. On the left, the 'Project tree' shows the hierarchy: Workshop_Faceplate2 > HMI_1 [MTP700 Unified Basic] > HMI tags > New HMI Tags [6]. The main workspace shows the 'New HMI Tags' dialog with a table of tags. The 'Name_vlv1' tag is highlighted. Below the dialog, the 'Discrete alarms' tab is active, showing a table with columns for ID, Name, Alarm text, Alarm class, Trigger tag, Trigg..., Connection of t., Acknowledg..., Ackn..., and Ack. The 'Name_vlv1 [HMI_Tag] Properties' window is open, showing the 'Values' tab with a 'Start value' field set to 'Valve 1'. Red circles with numbers 1-4 indicate the steps: 1. 'Add new tag table' in the project tree; 2. 'New HMI Tags [6]' in the project tree; 3. The 'New HMI Tags' table; 4. The 'Start value' field in the 'Name_vlv1 [HMI_Tag] Properties' window.

Name	Data type	Connection
Name_vlv1	WString	<Internal tag>
Name_vlv2	WString	<Internal tag>
Status_vlv1	Int	<Internal tag>
Status_vlv2	Int	<Internal tag>
RunPB_vlv1	Bool	<Internal tag>
RunPB_vlv2	Bool	<Internal tag>
<Add new>		

ID	Name	Alarm text	Alarm class	Trigger tag	Trigg...	Connection of t.	Acknowledg...	Ackn...	Ack
<Add new>									

Properties	Events	Texts
General		
Settings		
Range		
Linear scaling		
Values		
Comment		

1. Expand HMI Tags, Add a new tag table
2. Rename tag table as shown
3. Add tags as shown, notice Data Type and they are all internal tags
4. Highlight Name_vlv1 and change start value as shown. Repeat for Name_vlv2

Hands On: Create Faceplate type and instance

Content page – Hands On

The screenshot shows the Siemens HMI design environment. The main workspace displays an 'Overview' screen with a faceplate instance. A red circle with the number '1' is placed over the faceplate, with a red arrow pointing to the 'fpValve V 0.0.1 [default]' entry in the Project Library. The Properties panel at the bottom left shows the configuration for the selected faceplate instance.

1. From the Project Library drag fpValve onto the Overview screen

2. Change the Interface of the faceplate as shown

Name	Static value	Dynamization (0)
Appearance		
Format		
Miscellaneous		
Connection status	None	
Faceplate type	fpValve V 0.0.1	
Icon		None
Interface		
txtName	Name_vlv1	None
Status	Status_vlv1	None
RunPB	RunPB_vlv1	None
Label		

Hands On: Create Faceplate type and instance

Content page – Hands On

1. From the Elements Toolbox drag an IO Field as shown

2. Change the Process Value to 'Status_vlv1'

Note: This field will be used to test the "Status" of our valve

IO field_1 [IO field]

Name	Static value	Dynamization (1)
General		
Font		
Mode	Input/output	None
Output format		None
Process value		Tag
Appearance		

Tag: _____

Process

Tag: Status_vlv1

PLC tag: _____

Address: _____ Int

Hands On: Create Faceplate type and instance

Content page – Hands On

1. Save the project

2. Select HMI device in the Project Tree

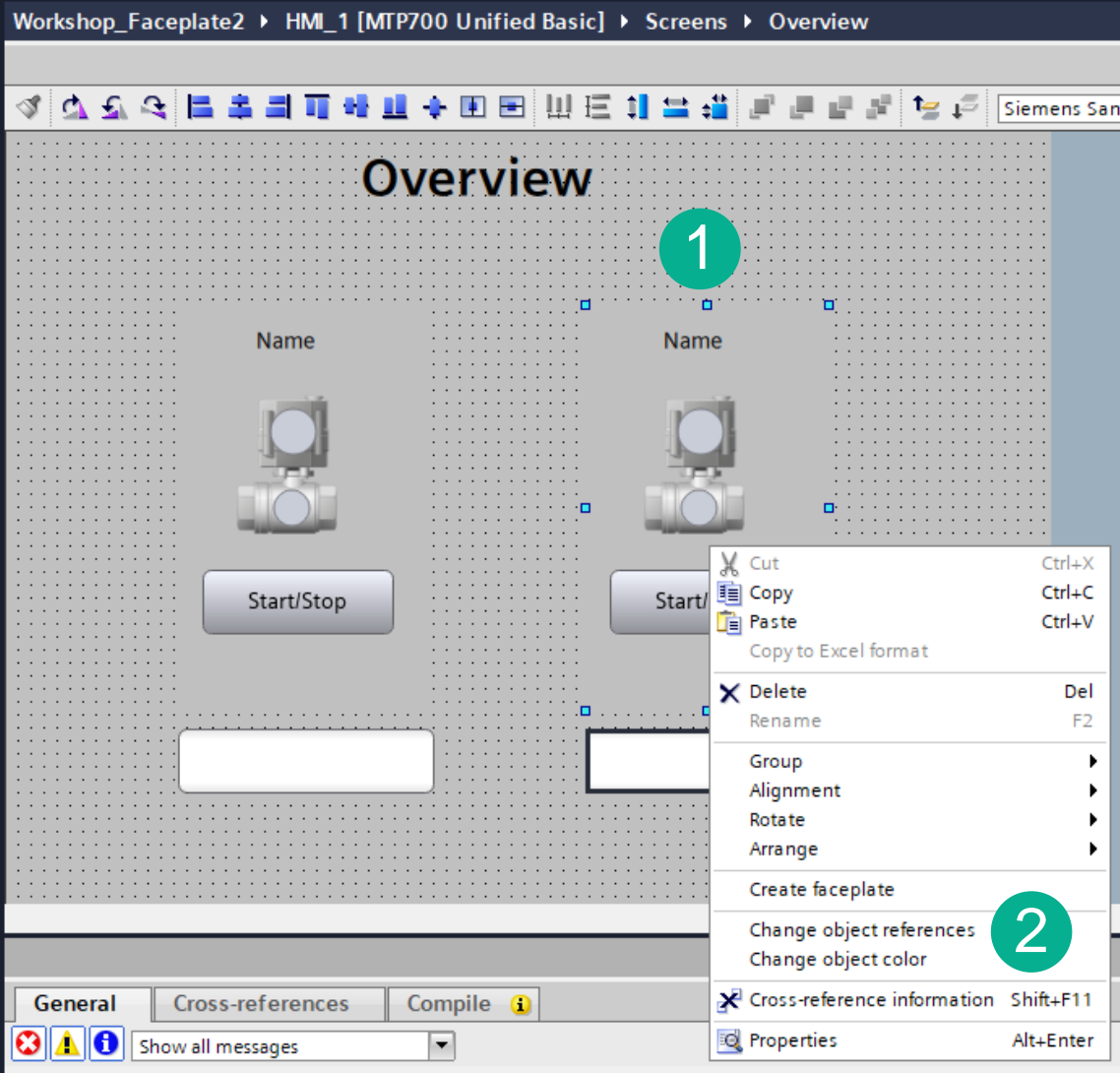
3. Press Start Simulation, Portal will open the default browser

4. Test your work

5. When finished close the browser

Hands On: Create Faceplate type and instance

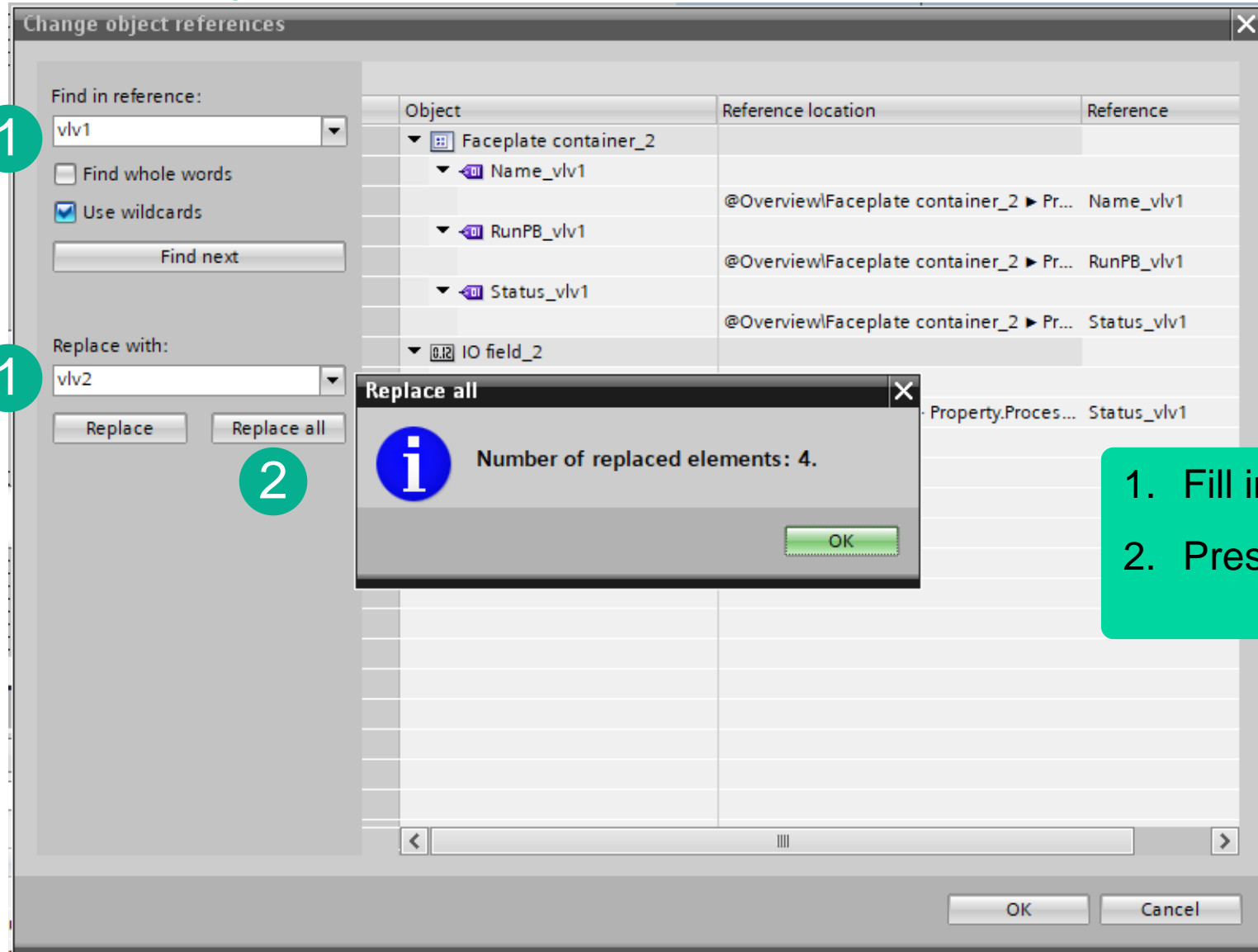
Content page – Hands On



1. Copy/Paste the Faceplate and IO Field
2. With the copy selected, right click and select Change Object References

Hands On: Create Faceplate type and instance

Content page – Hands On



1. Fill in Find and Replace fields as shown
2. Press Replace All

Hands On: Create Faceplate type and instance

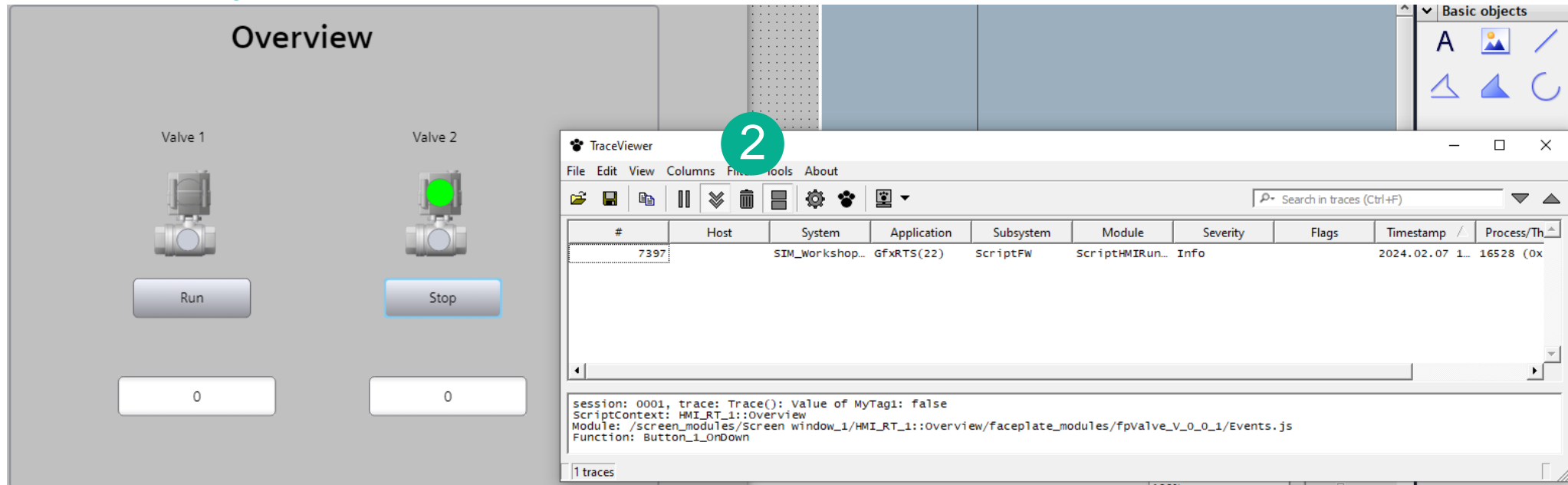
Content page – Hands On

The screenshot shows the Siemens SIMATIC Manager software interface. On the left is the Project Tree with 'HMI_1 [MTP700 Unified Basic]' selected. The top toolbar contains various icons, including 'Save project' (1) and 'Start Simulation' (3). The main area shows a browser window displaying an 'Overview' screen with two valve faceplates, 'Valve 1' and 'Valve 2'. 'Valve 1' has a red indicator light and a 'Run' button, while 'Valve 2' has a green indicator light and a 'Stop' button. A '4' callout points to the valve faceplates. A '5' callout points to the browser window's close button.

1. Save the project
2. Select HMI device in the Project Tree
3. Press Start Simulation, Portal will open the default browser
4. Test your work
5. When finished close the browser

Hands On: Create Faceplate type and instance

Optional – Using the Trace Viewer



1. The Trace Viewer can be launched from the following directory
 1. **C:\Program Files\Siemens\Automation\WinCCUnified\bin\RTILtraceViewer.exe**
2. Use the Trashcan to clear the log.
3. Test you screen and see the results in the Traceviewer

WinCC Unified Workshop



Siemens Industries Inc Digital Industries Factory Automation Visualization

Subject to changes and errors. The information given in this document only contains general descriptions and/or performance features which may not always specifically reflect those described, or which may undergo modification in the course of further development of the products. The requested performance features are binding only when they are expressly agreed upon in the concluded contract.

All product designations, product names, etc. may contain trademarks or other rights of Siemens, its affiliated companies or third parties. Their unauthorized use may infringe the rights of the respective owner.

usa.siemens.com