



FactoryTalk Optix Technical Training

Optix version 1.2.0.272



Table of contents

1. [Getting started](#)
2. [Be familiar with FTOptix Studio](#)
3. [How to Create a simple HMI project in few steps*](#)
4. [Define communication drivers and tags](#)
5. [Develop the user interface](#)
6. [Dynamic links](#)
7. [Animations and Events](#)
8. [Work with reusable objects](#)
9. [Getting a PRO license](#)
10. [Develop using version control and collaboration](#)
11. [Transfer and Execute the project](#)
12. [Save historical data with loggers](#)
13. [Represent data by Trends](#)
14. [Manage alarms](#)

* Wizards

Table of contents

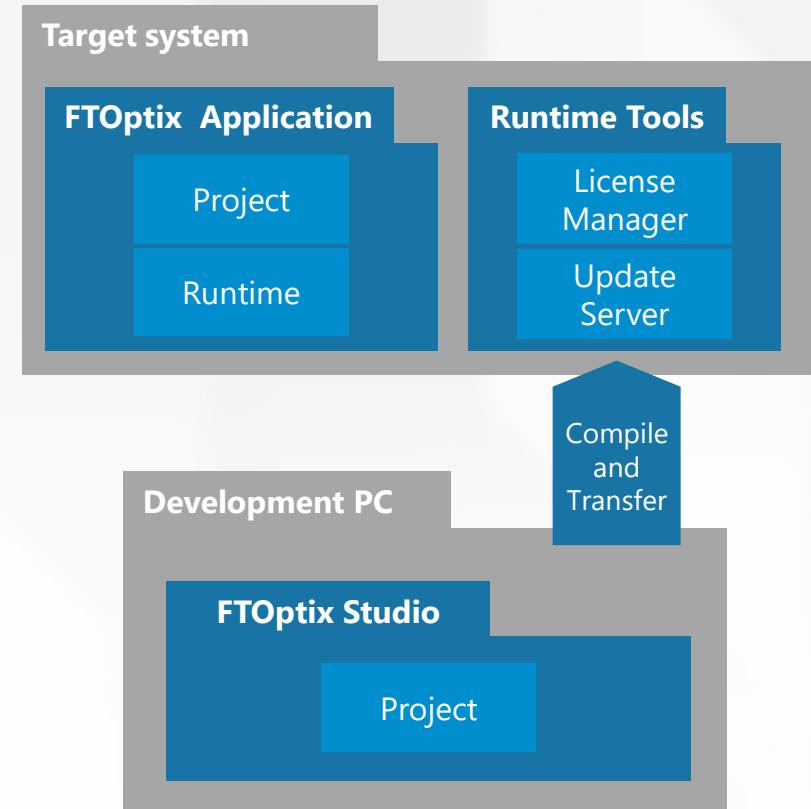
- [15. Use recipes](#)
- [16. Set Locale, Languages and Measurement System](#)
- [17. Define UI Session and Session variables](#)
- [18. Work with NetLogic scripts](#)
- [19. Configure Users and Groups](#)
- [20. Expose data as OPC-UA Server](#)
- [21. Exchange data via OPC-UA Client](#)
- [22. Define client-server architectures via OPC-UA](#)
- [23. Configure Reports](#)
- [24. Use Retentivity](#)
- [25. Connect to an External Database](#)
- [26. Appendix: see the light about OPC-UA Security](#)

Getting Started



Architecture

- **FTOptix Studio:** Integrated Development Environment (IDE) for designing and deploying HMI and IIOT applications
- **FTOptix Application:** compiled and "portable" application that combines the Project and the Runtime for the Target (HMI or iPC)
- Runtime Tools *:
• **Update Server:** Software to allow the deployment of Applications from Studio to the target
• **License Manager:** Software to manage the Optix license at runtime



Key features



EXTENSIBILITY options

Natively OPC UA

Libraries

Scripting



DESIGN options

Object-oriented

Software As
A Service

Version Control



GRAPHIC & DEPLOYMENT options

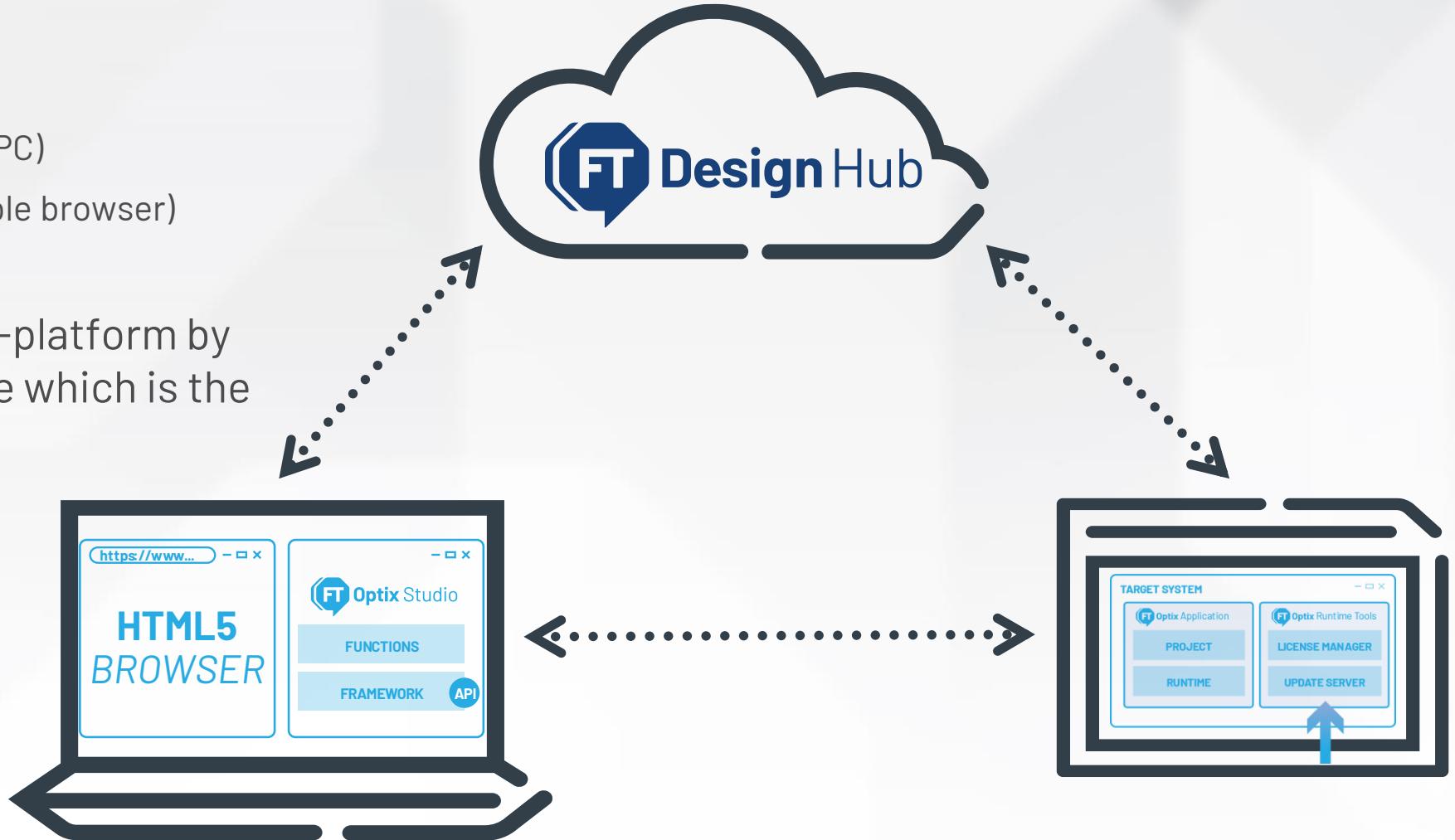
Cross-platform

Universal UI

Audit

FTOptix Studio

- Available both
 - «on premise» (setup on local PC)
 - «cloud» (any HTML5 compatible browser)
- All applications are cross-platform by design, no need to choose which is the target



FTOptix Studio flavours

STANDARD

- Install on your PC
- Design and Deploy application from your PC

FREE



PRO

STANDARD capabilities plus...

- Design application using **web-based** Studio
- Use the **Version Control** System (Project + Libraries) and multi-user **collaboration**

Annual subscription per User

FTOptix Runtime – Feature tokens overview

- Every **feature** used by the Runtime has a cost in **Tokens**
- License **size** defines the number of Tokens available

SIZE	RUNTIME FOR IPC
XS	5
S	8
M	11
L	15
XL	21
UNL	Unlimited



FTOptix Runtime – Feature tokens example

SMALL HMI

Native UI	1
Web UI (up to 1 clients)	1
Alarms + Alarms History	1
Data logger	1
Recipes	1
1x Comm Driver (multiple stations)	2
TOTAL	7

MEDIUM HMI

Native UI	1
Web UI (up to 3 clients)	2
Alarms + Alarms History	1
Data logger	1
Retentivity	1
Recipes	1
OPC UA Server (1 client)	1
1x Comm Driver (multiple station)	2
TOTAL	10



Getting Started

- Just need to create a «My Rockwell» account then...
- Optix Studio STANDARD can be downloaded for free from
<https://compatibility.rockwellautomation.com/>
- Optix Studio PRO is available as 90 Days Trial at
<https://home.cloud.rockwellautomation.com>

A large, abstract graphic of blue dots forms a wave-like pattern across the background, starting from the bottom left and curving upwards towards the top right.

Be familiar with
FTOptix Studio



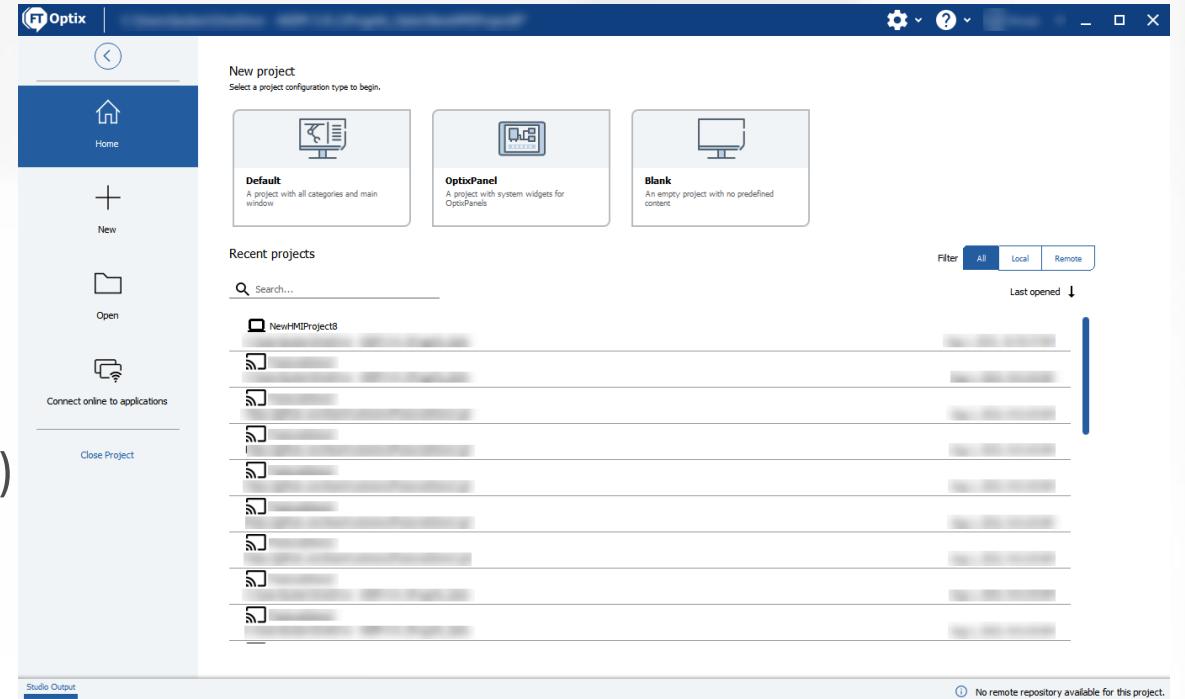
Development environment: new project wizard

Choice between

- Default: project with default folders/resource structure (recommended)
- OptixPanel: project with Optix Panel's dedicated widgets (network, screen settings, etc)
- Blank: empty project, without folders/resources structure (only if «advanced mode» is enabled)

Project configuration

- Name, Location
- Use version control: allow the project versioning and collaboration between developers (GIT)
- Encrypt secrets: passwords are encrypted via a private key stored in the FT HUB domain
- Default screens resolution (can be changed later)



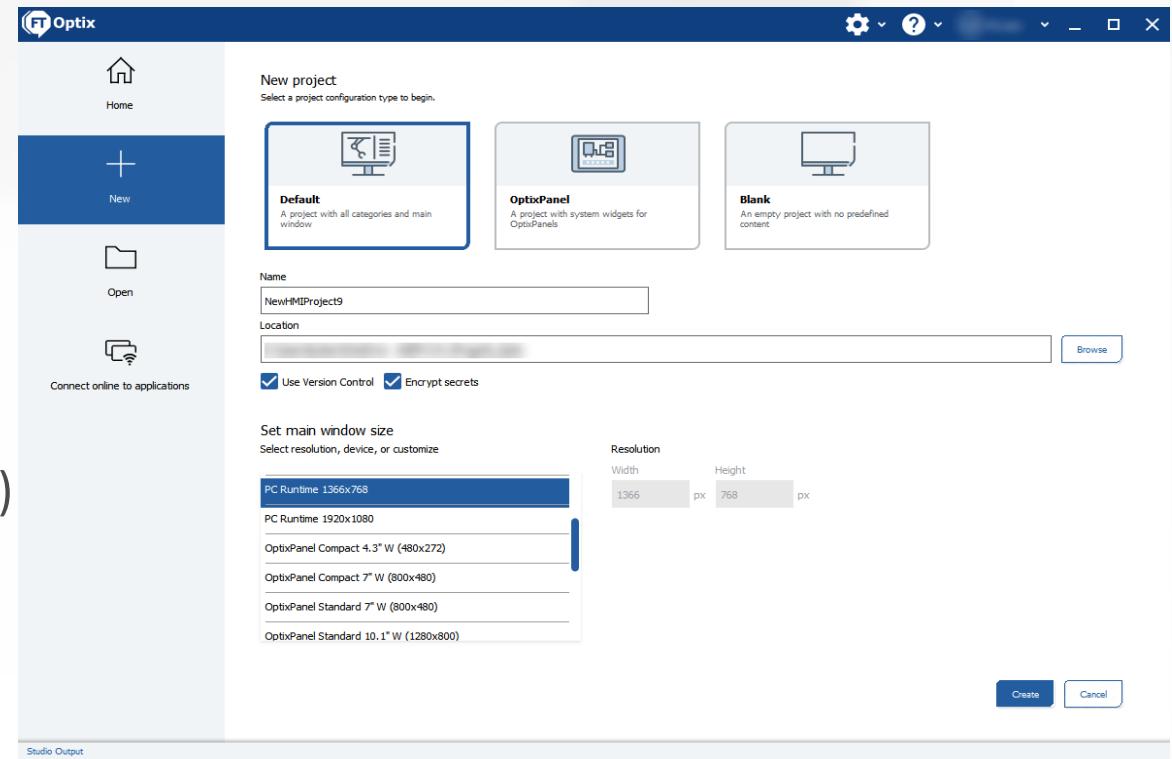
Development environment: new project wizard

Choice between

- Default: project with default folders/resource structure (recommended)
- OptixPanel: project with Optix Panel's dedicated widgets (network, screen settings, etc)
- Blank: empty project, without folders/resources structure (only if «advanced mode» is enabled)

Project configuration

- Name, Location
- Use version control: allow the project versioning and collaboration between developers (GIT)
- Encrypt secrets: passwords are encrypted via a private key stored in the FT HUB domain
- Default screens resolution (can be changed later)



Development environment

1. Wizards

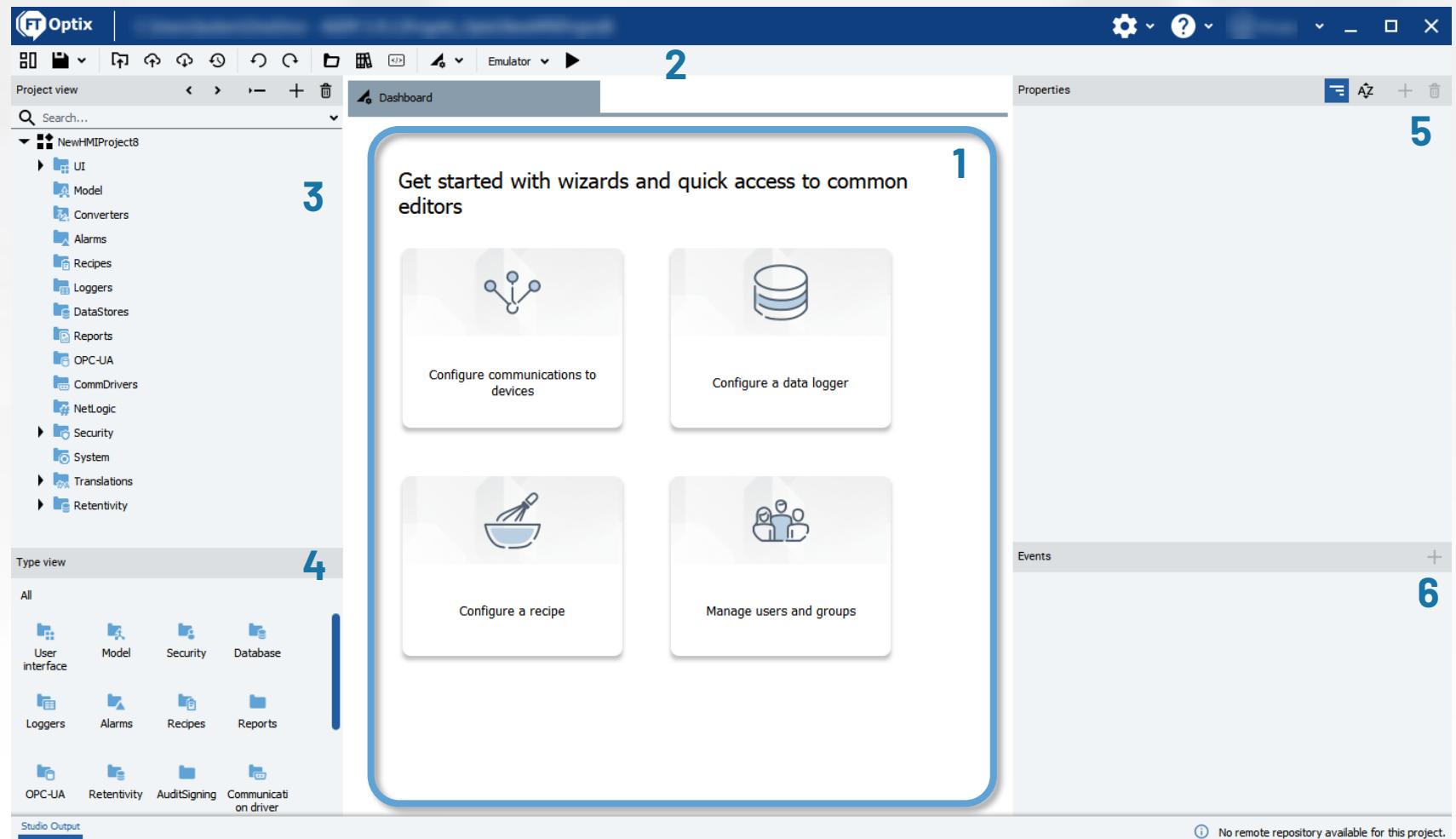
2. Main Toolbar

3. Project View panel

4. Types View panel

5. Properties panel

6. Events panel



Development environment

1. Wizards

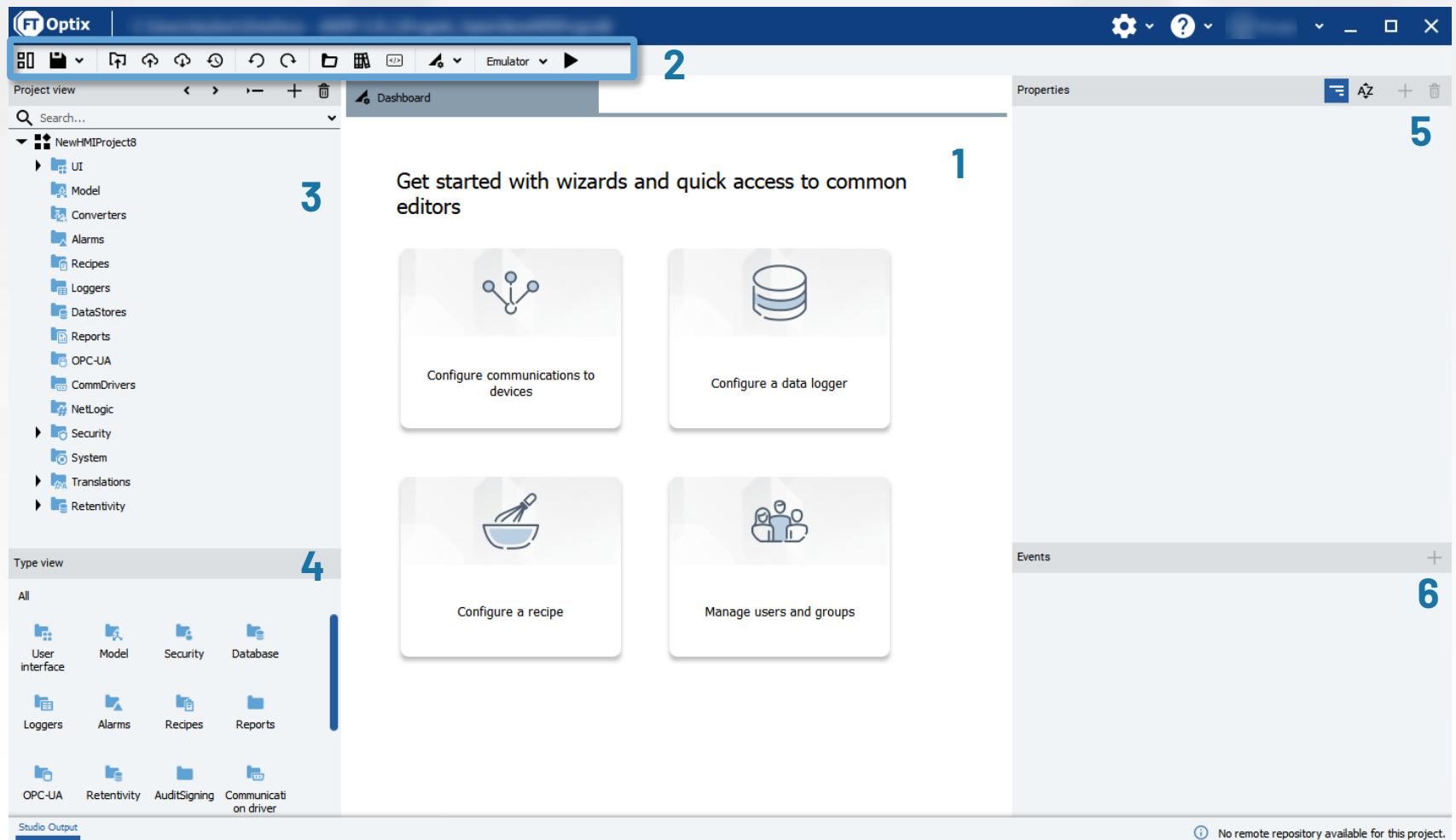
2. Main Toolbar

3. Project View panel

4. Types View panel

5. Properties panel

6. Events panel



Development environment

1. Wizards

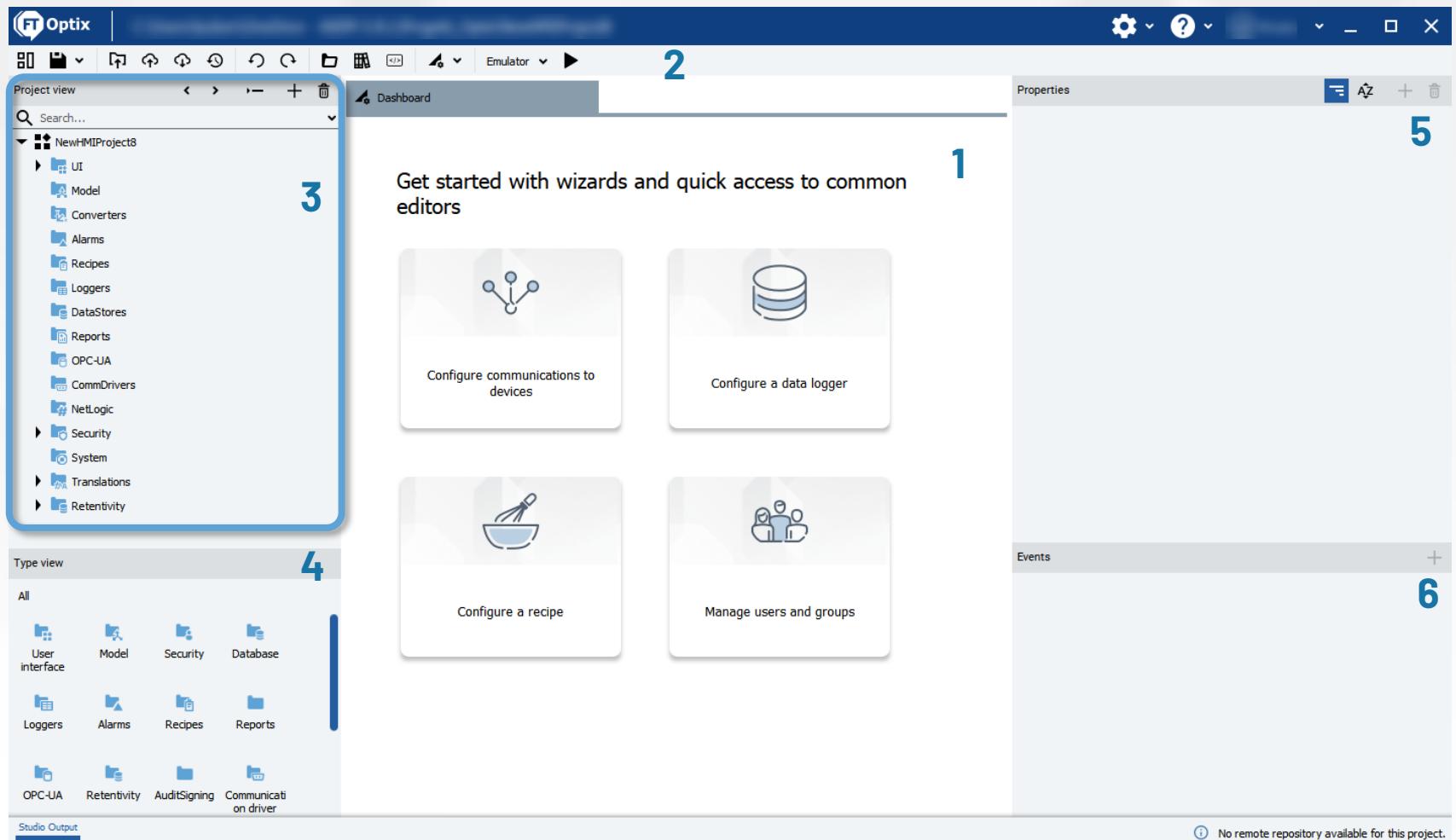
2. Main Toolbar

3. Project View panel

4. Types View panel

5. Properties panel

6. Events panel



Development environment

1. Wizards

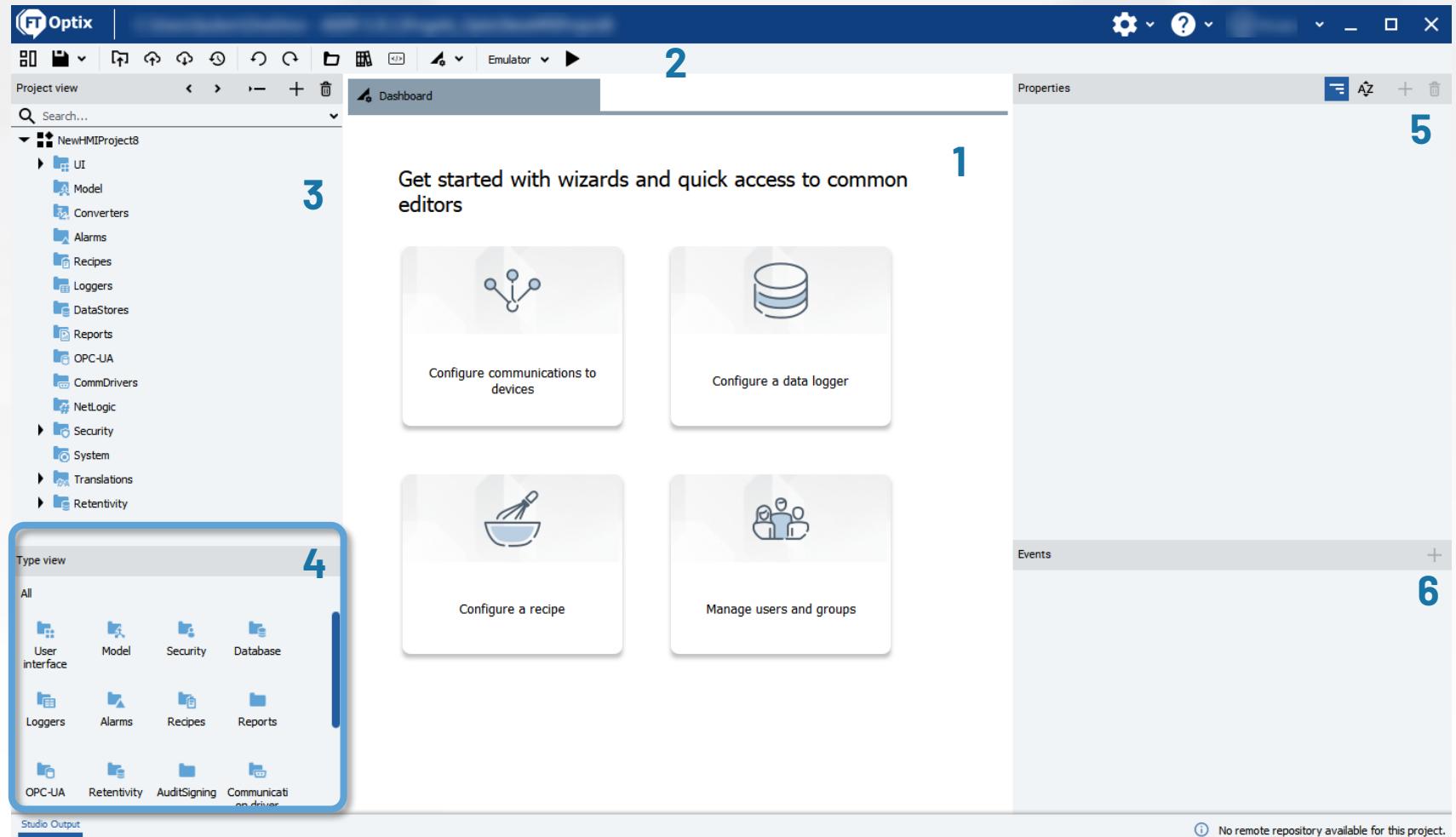
2. Main Toolbar

3. Project View panel

4. Types View panel

5. Properties panel

6. Events panel



Development environment

1. Wizards

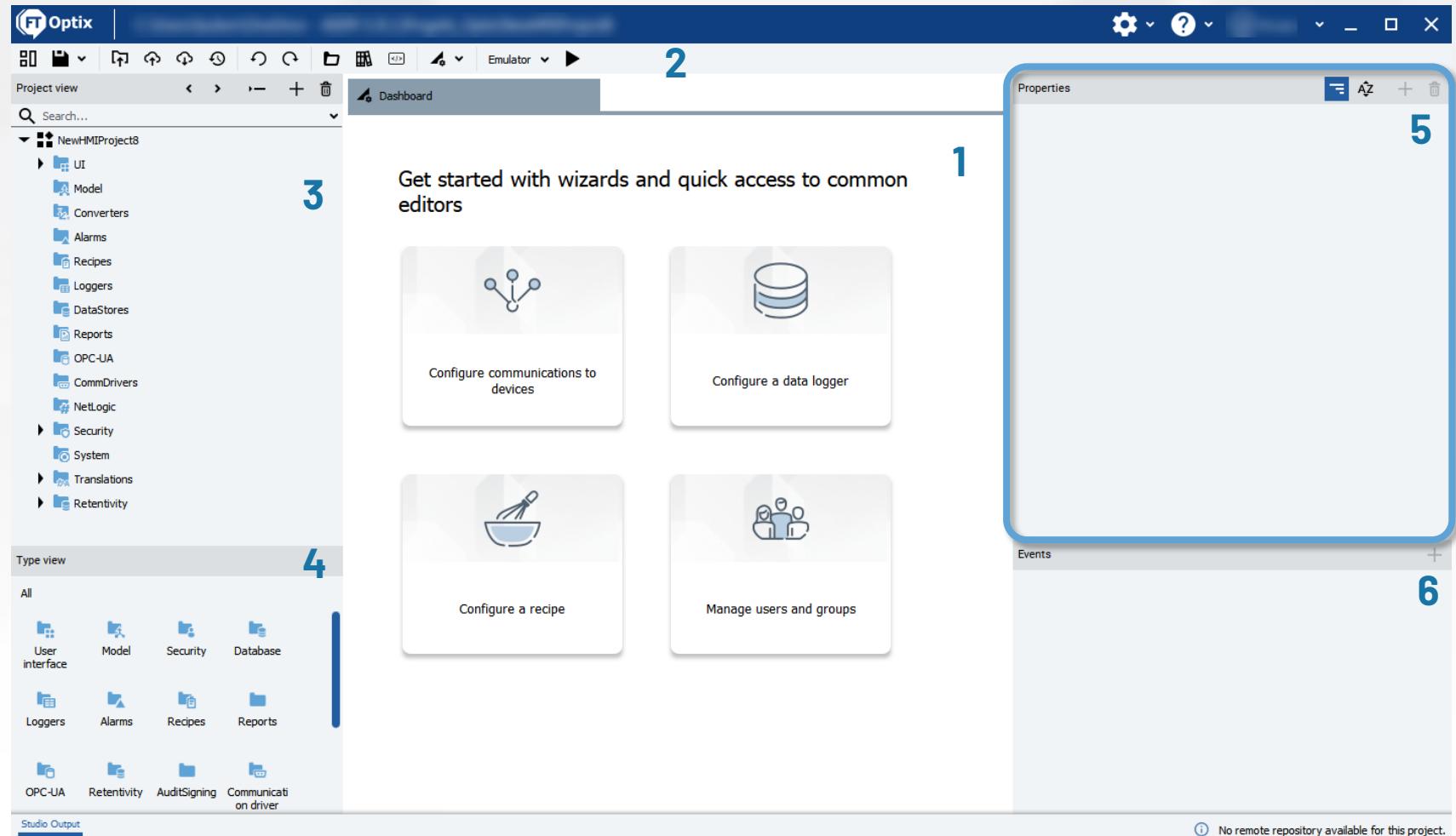
2. Main Toolbar

3. Project View panel

4. Types View panel

5. Properties panel

6. Events panel



Development environment

1. Wizards

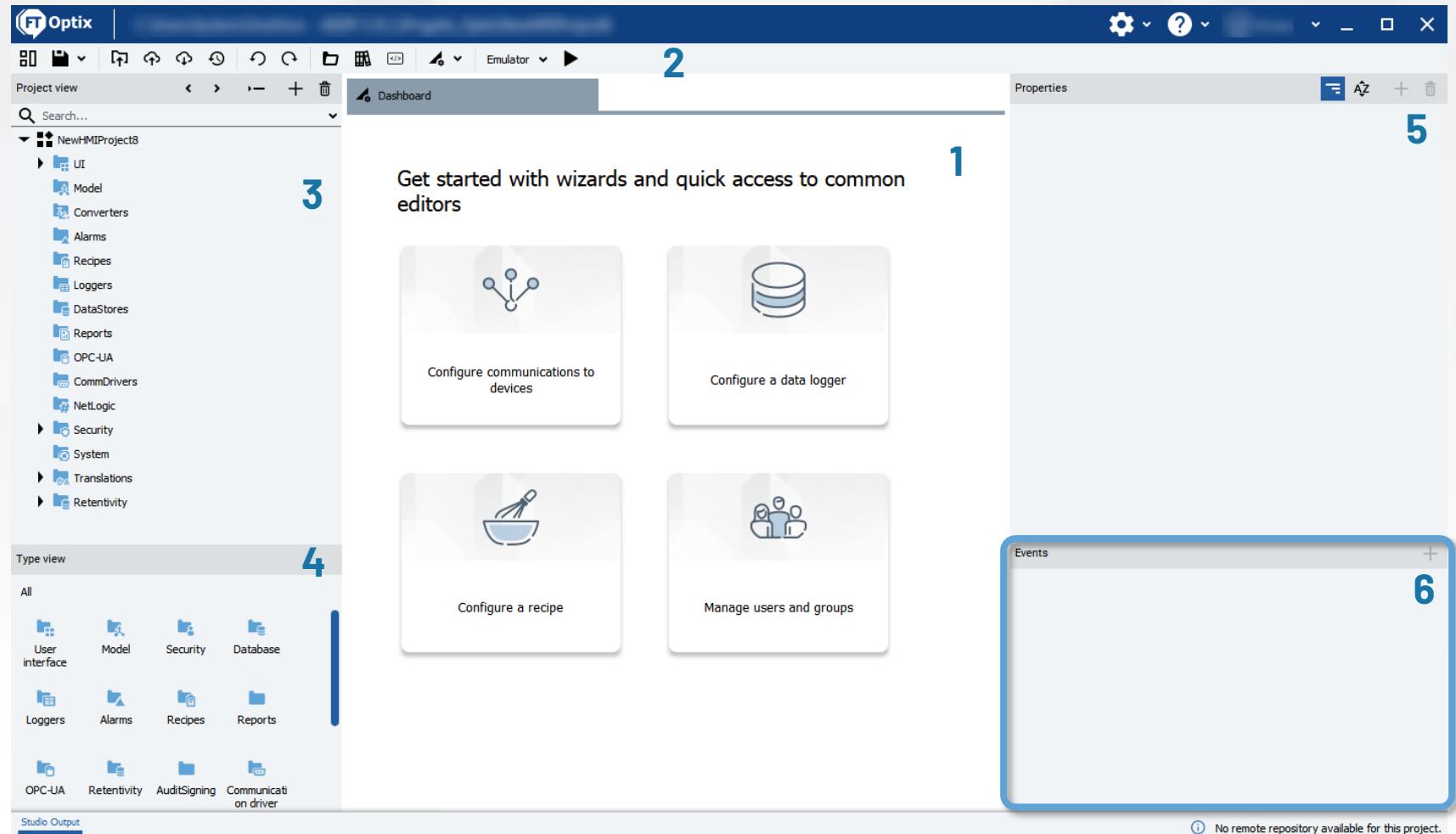
2. Main Toolbar

3. Project View panel

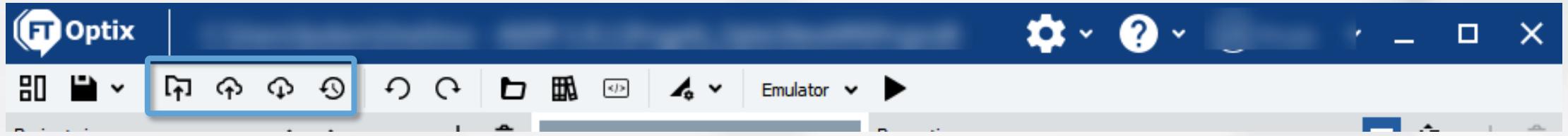
4. Types View panel

5. Properties panel

6. Events panel

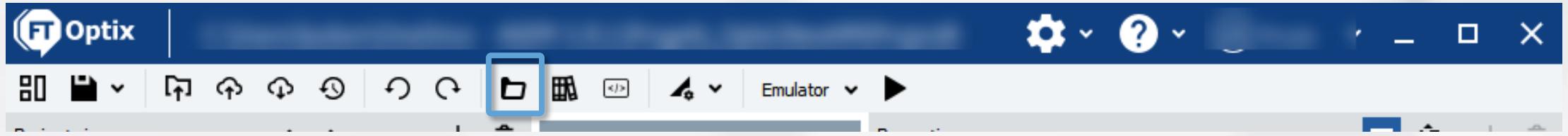


Main toolbar



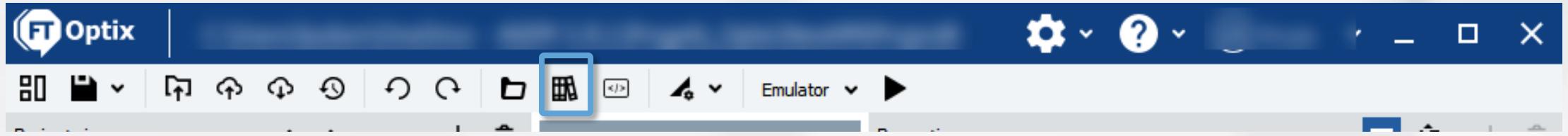
- **Collaboration:** Save & Commit/Push/Pull/History
- **Project Files**
- **Libraries** (%UserProfile%\Documents\Rockwell Automation\FactoryTalk Optix\Libraries)
- **NetSolution**
- **Wizards**
- **Deploy options:** Run the project within Emulator or transfer and run on a Target
- **Studio Options**

Main toolbar



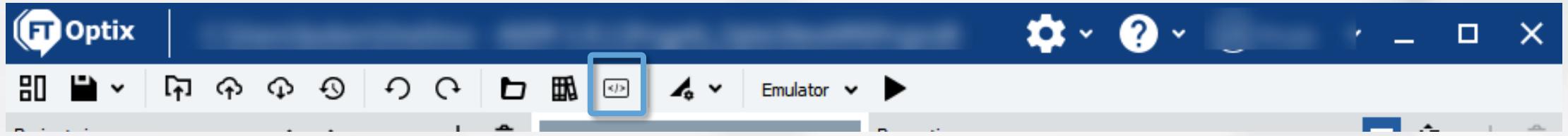
- **Collaboration:** Save & Commit/Push/Pull/History
- **Project Files**
- **Libraries** (%UserProfile%\Documents\Rockwell Automation\FactoryTalk Optix\Libraries)
- **NetSolution**
- **Wizards**
- **Deploy options:** Run the project within Emulator or transfer and run on a Target
- **Studio Options**

Main toolbar



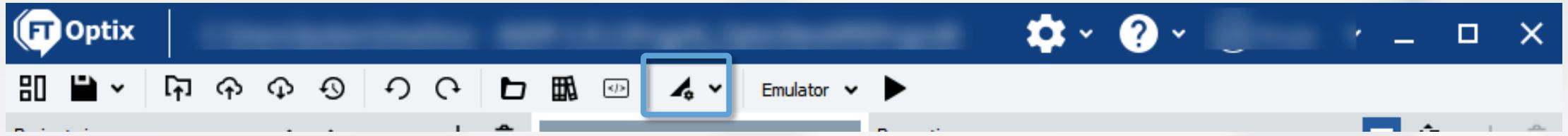
- **Collaboration:** Save & Commit/Push/Pull/History
- **Project Files**
- **Libraries** (%UserProfile%\Documents\Rockwell Automation\FactoryTalk Optix\Libraries)
- **NetSolution**
- **Wizards**
- **Deploy options:** Run the project within Emulator or transfer and run on a Target
- **Studio Options**

Main toolbar



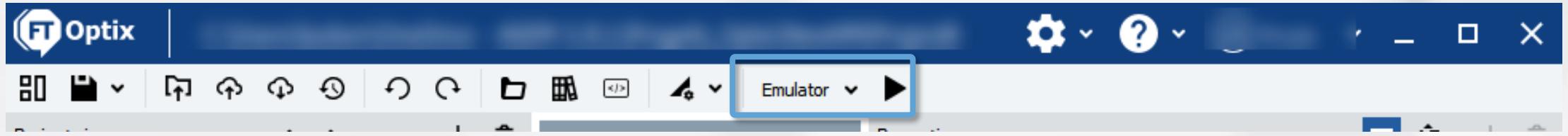
- **Collaboration:** Save & Commit/Push/Pull/History
- **Project Files**
- **Libraries** (%UserProfile%\Documents\Rockwell Automation\FactoryTalk Optix\Libraries)
- **NetSolution**
- **Wizards**
- **Deploy options:** Run the project within Emulator or transfer and run on a Target
- **Studio Options**

Main toolbar



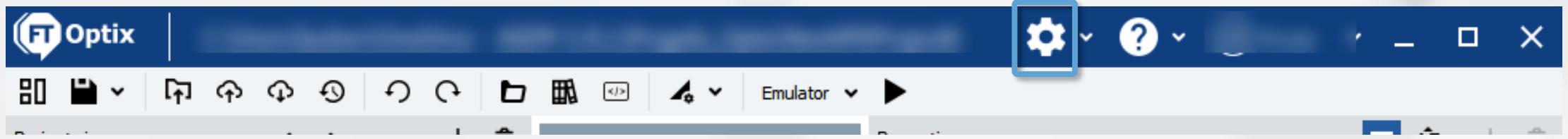
- **Collaboration:** Save & Commit/Push/Pull/History
- **Project Files**
- **Libraries** (%UserProfile%\Documents\Rockwell Automation\FactoryTalk Optix\Libraries)
- **NetSolution**
- **Wizards**
- **Deploy options:** Run the project within Emulator or transfer and run on a Target
- **Studio Options**

Main toolbar



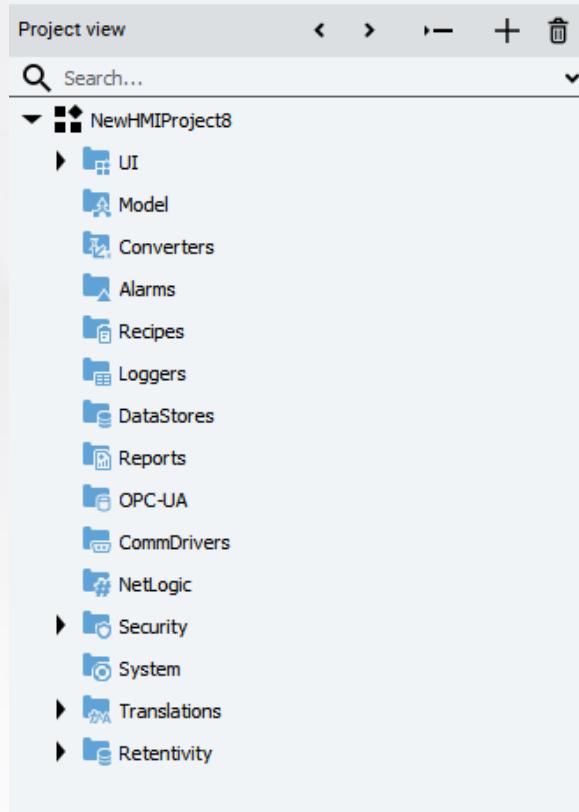
- **Collaboration:** Save & Commit/Push/Pull/History
- **Project Files**
- **Libraries** (%UserProfile%\Documents\Rockwell Automation\FactoryTalk Optix\Libraries)
- **NetSolution**
- **Wizards**
- **Deploy options:** Run the project within Emulator or transfer and run on a Target
- **Studio Options**

Main toolbar



- **Collaboration:** Save & Commit/Push/Pull/History
- **Project Files**
- **Libraries** (%UserProfile%\Documents\Rockwell Automation\FactoryTalk Optix\Libraries)
- **NetSolution**
- **Wizards**
- **Deploy options:** Run the project within Emulator or transfer and run on a Target
- **Studio Options**

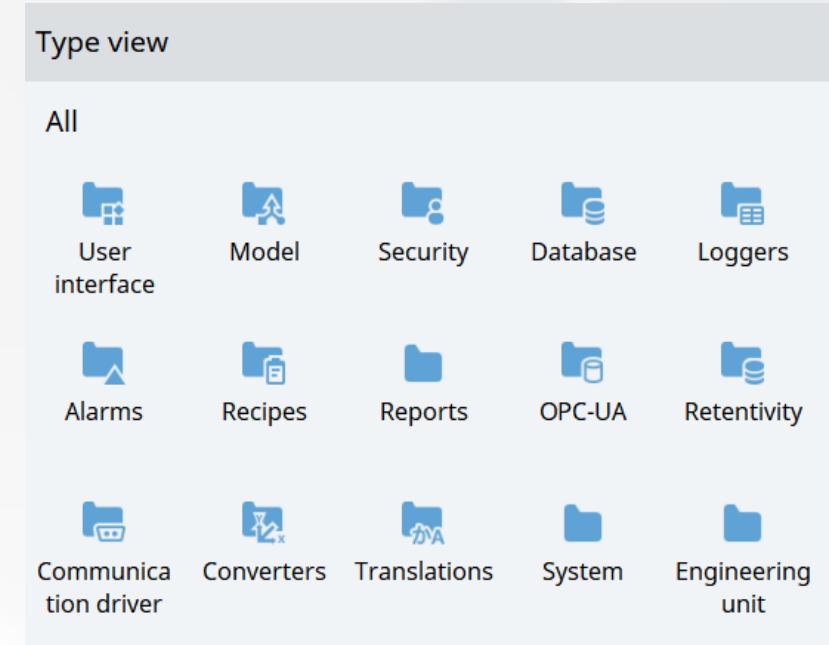
Project view



- The Project view shows, in a **typical tree architecture**, the project folders
- Folder structure constitutes the foundation of a project and are organized by type and functionality
- The Project view tree supports:
 - Context commands with the mouse (right-click)
 - Multiple selections
 - Drag&Drop
 - Copy/Paste
 - Search and filter
 - Detach (coming soon)

Type view

- Collection of **Native Types**, grouped in folders according to their purpose:
 - Basic User Control: to develop the User Interface
 - Resource objects to be added to the project folders
 - Companion specifications
- Object from Type view can be added into the project Tree by Drag&Drop
- Can be also the repository of **Custom Types**
- Here you can find the **Companion Specifications** types too



Property panel

- This pane shows properties of the selected node
 - Project
 - Resource
 - Objects
- Depending on the type of property, the editing can be done through:
 - Direct editing by typing
 - by Selecting the pencil icon 
 - Drop-down menu
 - by Selecting the DynamicLink icon 
- Allow editing “Multiple selections”

Properties	
Name	MyProject
Type	Project folder
Locales	en-US
Translation fallback locales	en-US
Branching enabled	False
Measurement systems map	<i>Default mapping</i> 
Authentication mode	Model only
Default user folder	Users 
>Password policy	
Maximum password age	0
Enforce password history	1
Minimum password age	0
Minimum password length	8

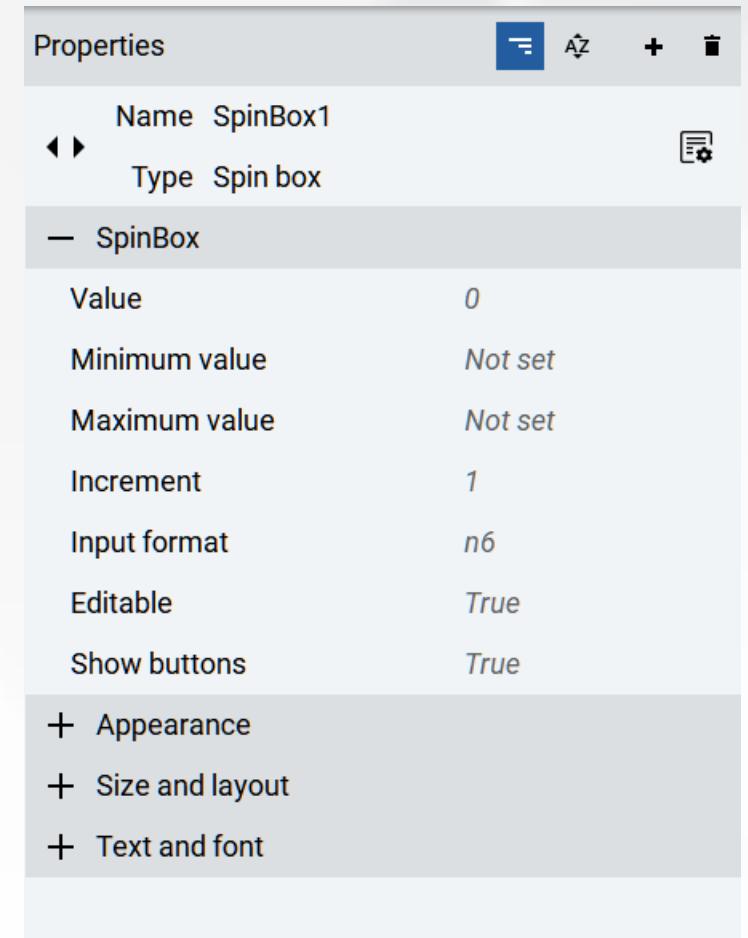
Property panel

- This pane shows properties of the selected node
 - Project
 - Resource
 - Objects
- Depending on the type of property, the editing can be done through:
 - Direct editing by typing
 - by Selecting the pencil icon 
 - Drop-down menu
 - by Selecting the DynamicLink icon 
- Allow editing “Multiple selections”

Properties	
Name	Screen1 (type)
Type	Panel
Visible	True
Enabled	True
Opacity	100
Rotation	0
Hit test visible	False
— Size and layout	
Horizontal alignment	Left
Vertical alignment	Top
Width	300
Height	300
Left margin	0
Top margin	0

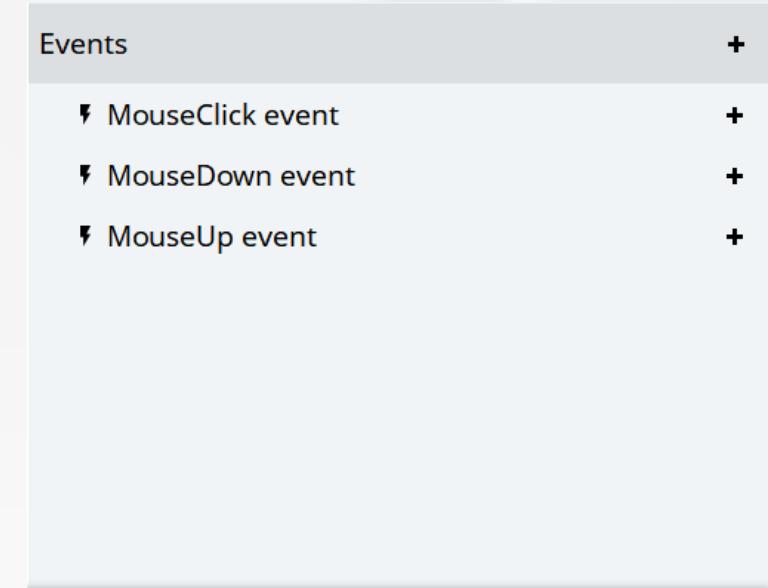
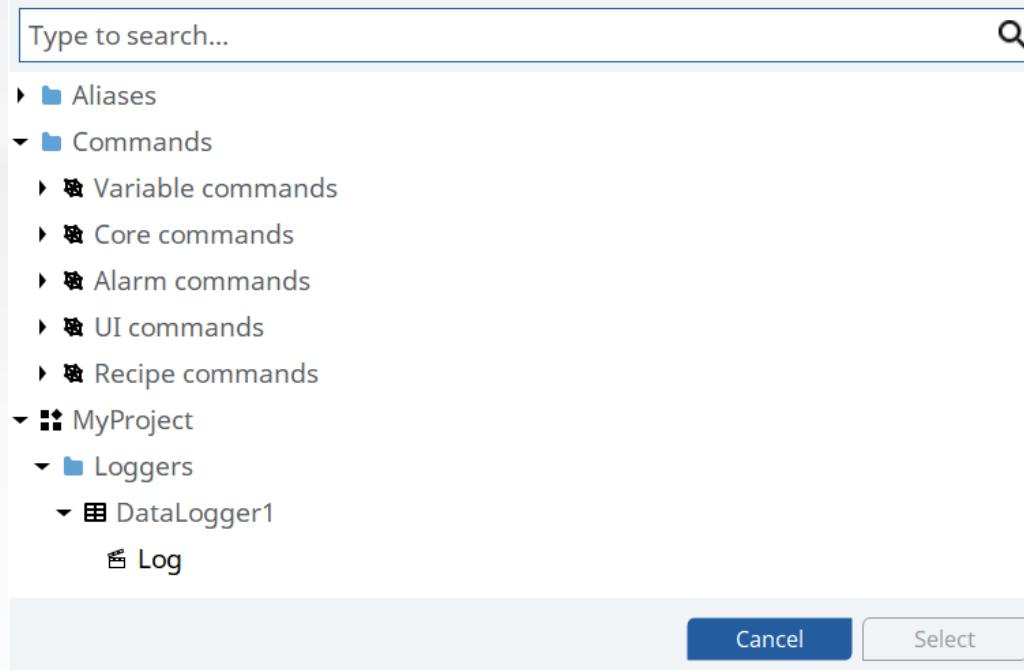
Property panel

- This pane shows properties of the selected node
 - Project
 - Resource
 - Objects
- Depending on the type of property, the editing can be done through:
 - Direct editing by typing
 - by Selecting the pencil icon 
 - Drop-down menu
 - by Selecting the DynamicLink icon 
- Allow editing “Multiple selections”



Event panel

- Allows defining Commands or Methods to be executed when events are triggered by the selected object
- Different objects have different events



How to create a simple HMI project in few steps



Create a simple HMI project in few steps

1. Configure connected device

Dashboard

Get started with wizards and quick access to common editors

Configure communications to devices

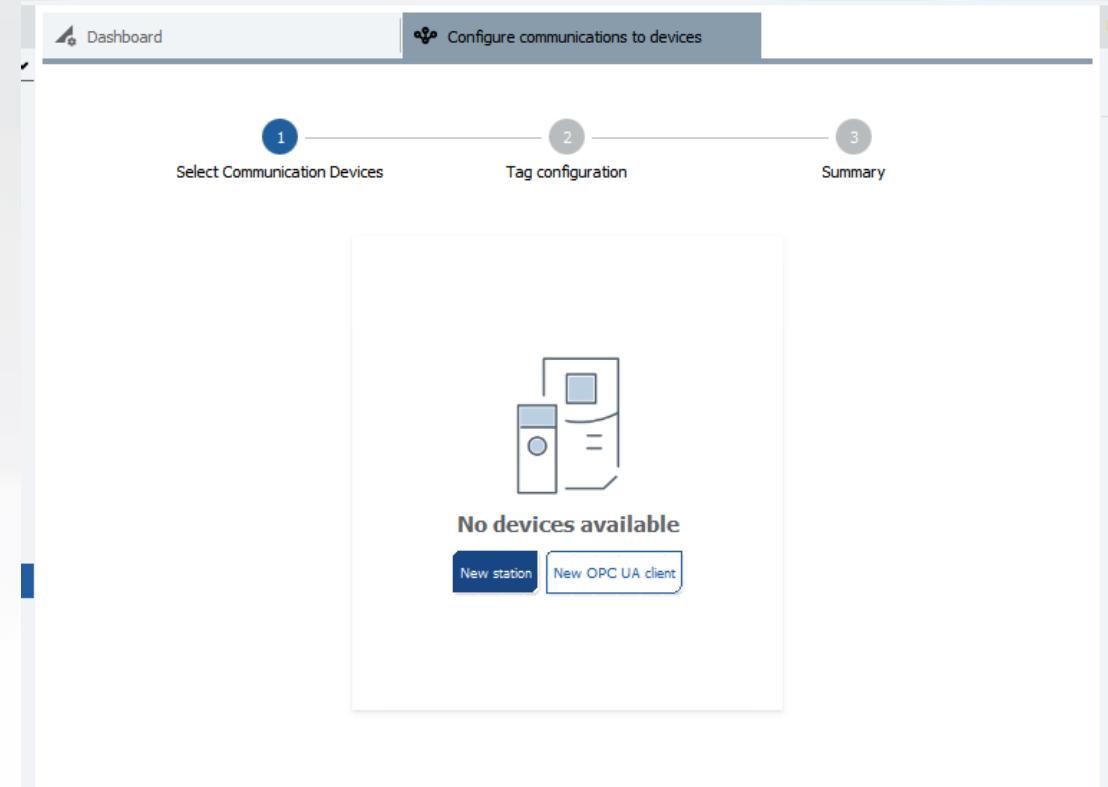
Configure a data logger

Configure a recipe

Manage users and groups

Create a simple HMI project in few steps

1. Configure connected device



Create a simple HMI project in few steps

1. Configure connected device

The screenshot shows a software interface for configuring communication devices. At the top, there's a navigation bar with 'Dashboard' and 'Configure communications to devices'. Below the navigation bar, a progress bar indicates three steps: 'Select Communication Devices' (step 1, highlighted in blue), 'Tag configuration' (step 2), and 'Summary' (step 3). The main area is titled 'Select protocol' and shows two options: 'CommDriver' (selected) and 'OPC UA Client'. A section titled 'New station' contains fields for 'Name' (RAEtherNet_IPStation1) and 'Type'. Under 'Type', several options are listed as radio buttons, with 'RA EtherNet/IP Station' selected.

Dashboard

Configure communications to devices

1 Select Communication Devices 2 Tag configuration 3 Summary

Select protocol

CommDriver OPC UA Client

New station

Name

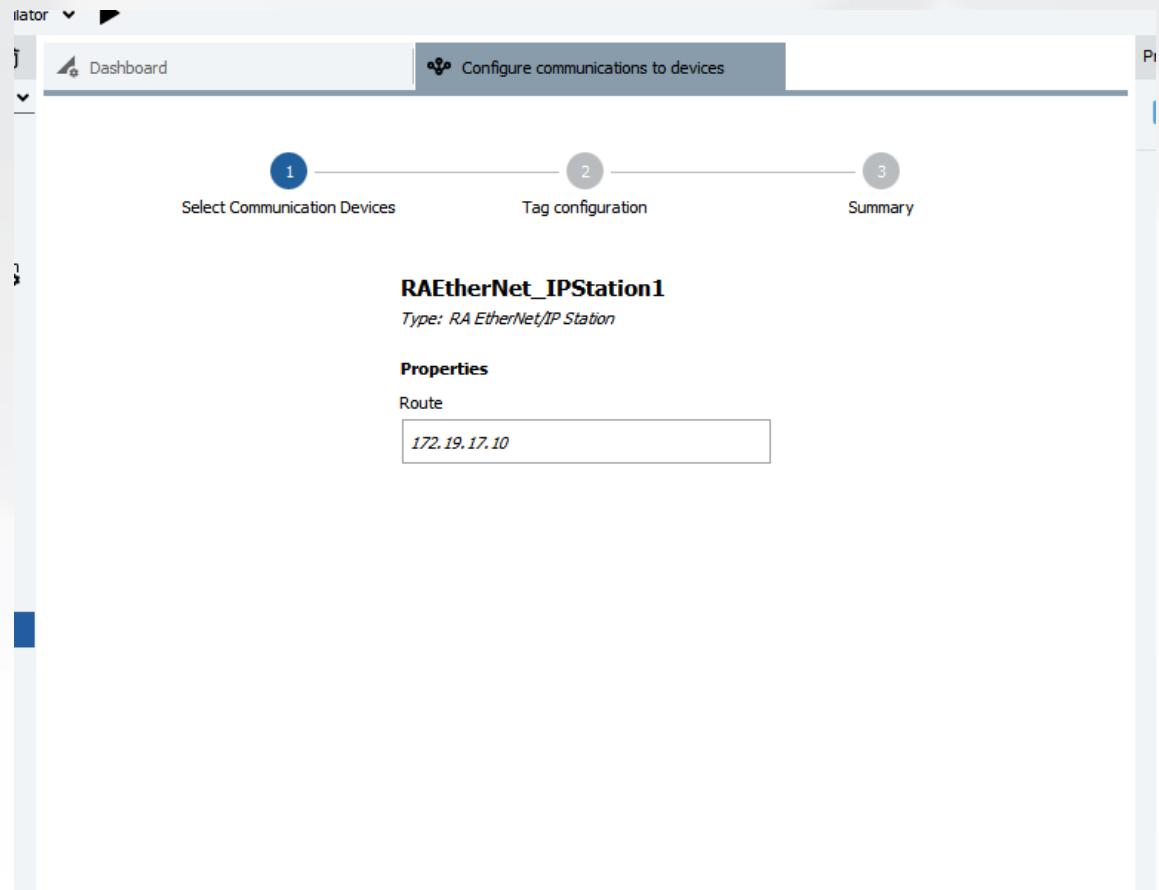
RAEtherNet_IPStation1

Type

CODESYS Station
 MELSEC FX3U Station
 MELSEC Q station
 Modbus Station
 OMRON EtherNet/IP station
 OMRON Fins Station
 RA EtherNet/IP Station
 S7TCP Station
 S7 TIA PROFINET station
 TwinCAT station

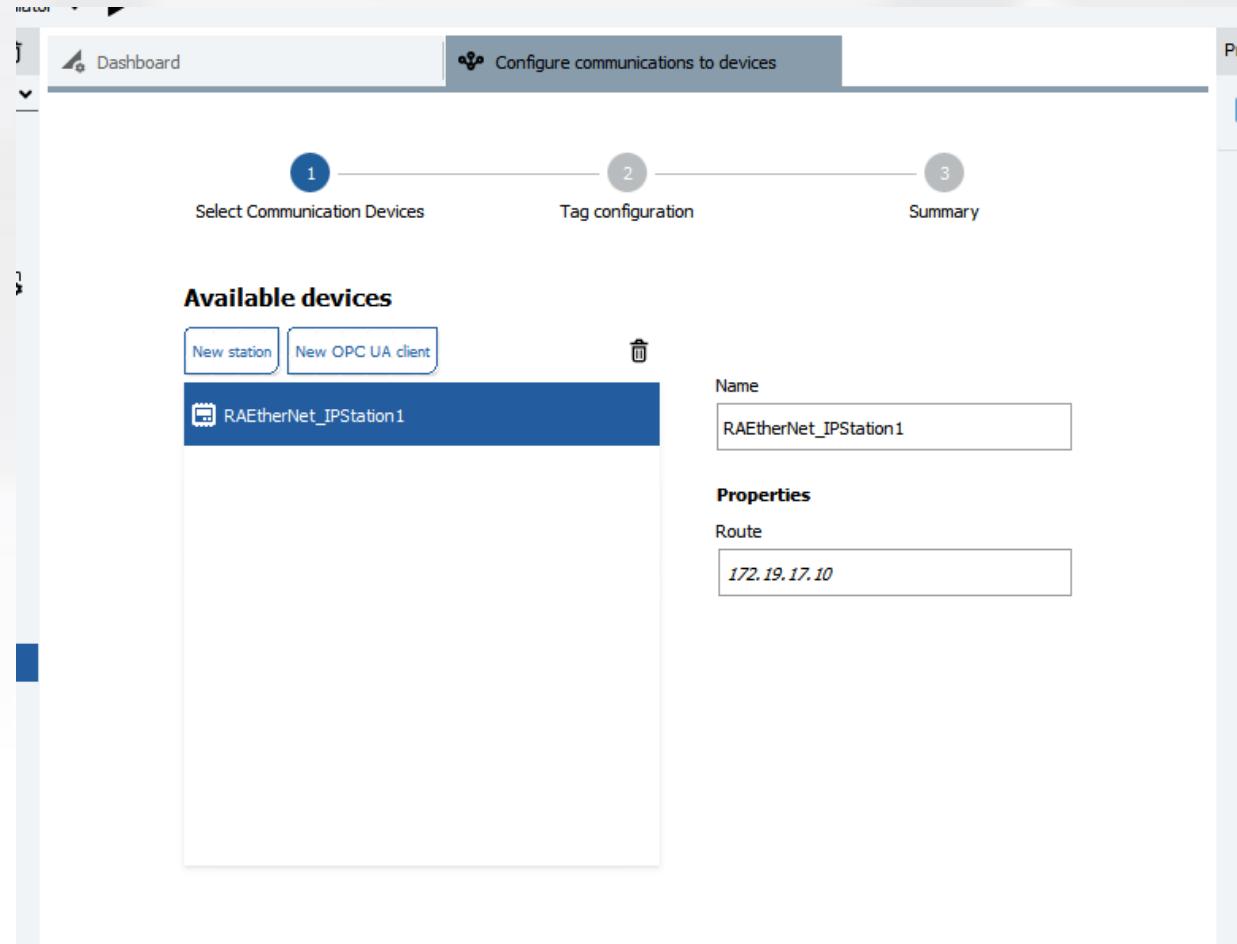
Create a simple HMI project in few steps

1. Configure connected device



Create a simple HMI project in few steps

1. Configure connected device



Create a simple HMI project in few steps

1. Configure connected device

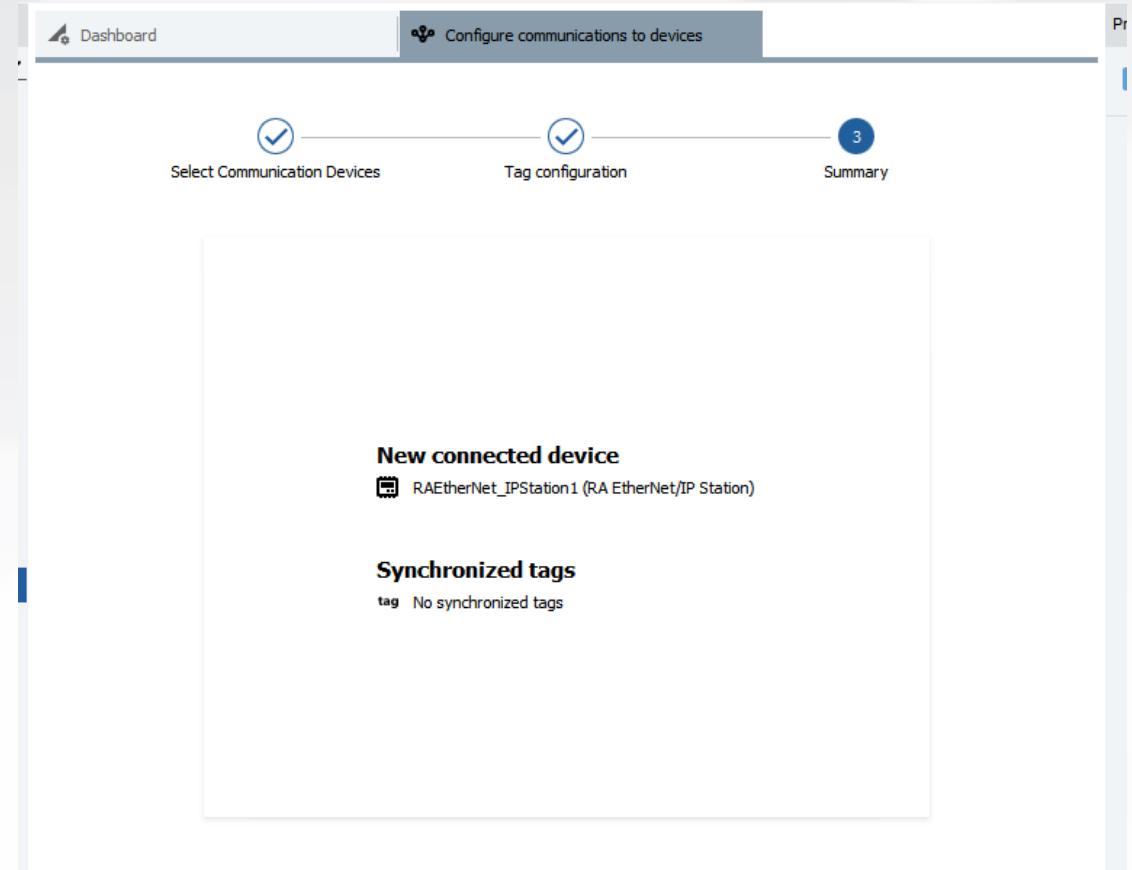
The screenshot shows a software interface for configuring communication devices. At the top, there are tabs for 'Dashboard' and 'Configure communications to devices'. Below these are three numbered steps: 1. Select Communication Devices (marked with a checkmark), 2. Tag configuration, and 3. Summary. Step 1 is currently active. The main area displays a list of tags from a device named 'RAEtherNet_IPStation1'. The list includes:

Tag Name	Type	Description
tag Motors	Motor	Motor[5]
tag Tank1	TankUDT	TankUDT
tag Tank2	TankUDT	TankUDT
tag TestBoolTag	RA EtherNet/IP Tag	Boolean
tag TestIncrementTag	RA EtherNet/IP Tag	Int32

Below the tag list, there is a folder named 'Program:AOI_LocalTagsProgram'. On the left, there is a status indicator showing 'Online' with a green circle and a search bar with a magnifying glass icon.

Create a simple HMI project in few steps

1. Configure connected device



Create a simple HMI project in few steps

2. Configure a datalogger

Let's begin building your user interface



I want to configure connected devices



I want to configure a data logger



I want to configure a recipe



I want to manage users and groups

Create a simple HMI project in few steps

2. Configure a datalogger

Configure data Logger

1 Select data logger 2 Select database 3 Select variables 4 Summary



No data loggers

New data loggers

Exit Next

Create a simple HMI project in few steps

2. Configure a datalogger

Configure data Logger

1 Select data logger 2 Select database 3 Select variables 4 Summary

New data logger

Name

Properties

Sampling mode

Sampling period

Buttons

Exit Back Next

Create a simple HMI project in few steps

2. Configure a datalogger

Configure data Logger

1 Select data logger 2 Select database 3 Select variables 4 Summary

Available data loggers

New data logger X

DataLogger1	Name DataLogger1
-------------	---------------------

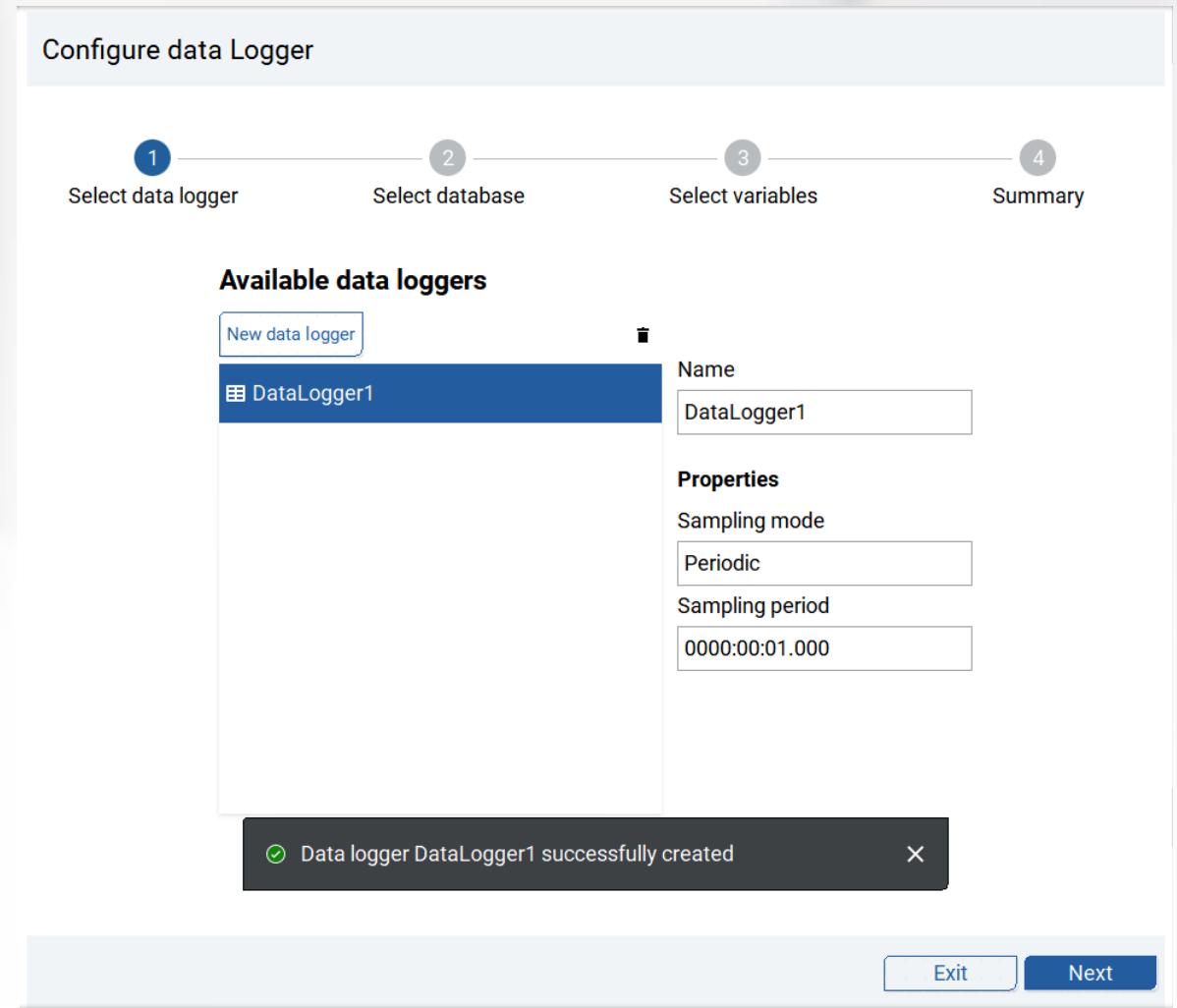
Properties

Sampling mode
Periodic

Sampling period
0000:00:01.000

✓ Data logger DataLogger1 successfully created X

Exit Next



Create a simple HMI project in few steps

2. Configure a datalogger

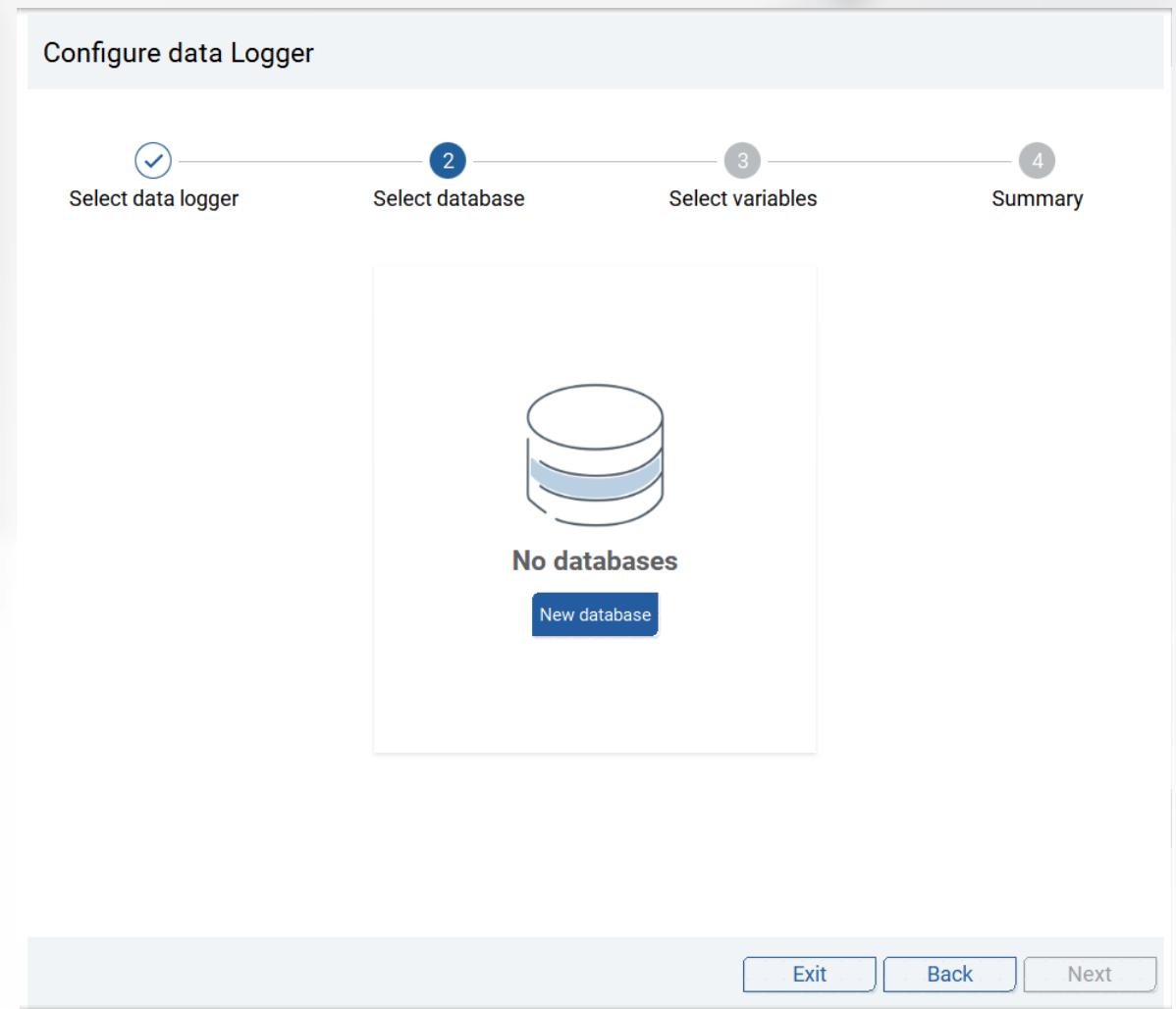
Configure data Logger

1 Select data logger 2 Select database 3 Select variables 4 Summary

No databases

New database

Exit Back Next



Create a simple HMI project in few steps

2. Configure a datalogger

Configure data Logger

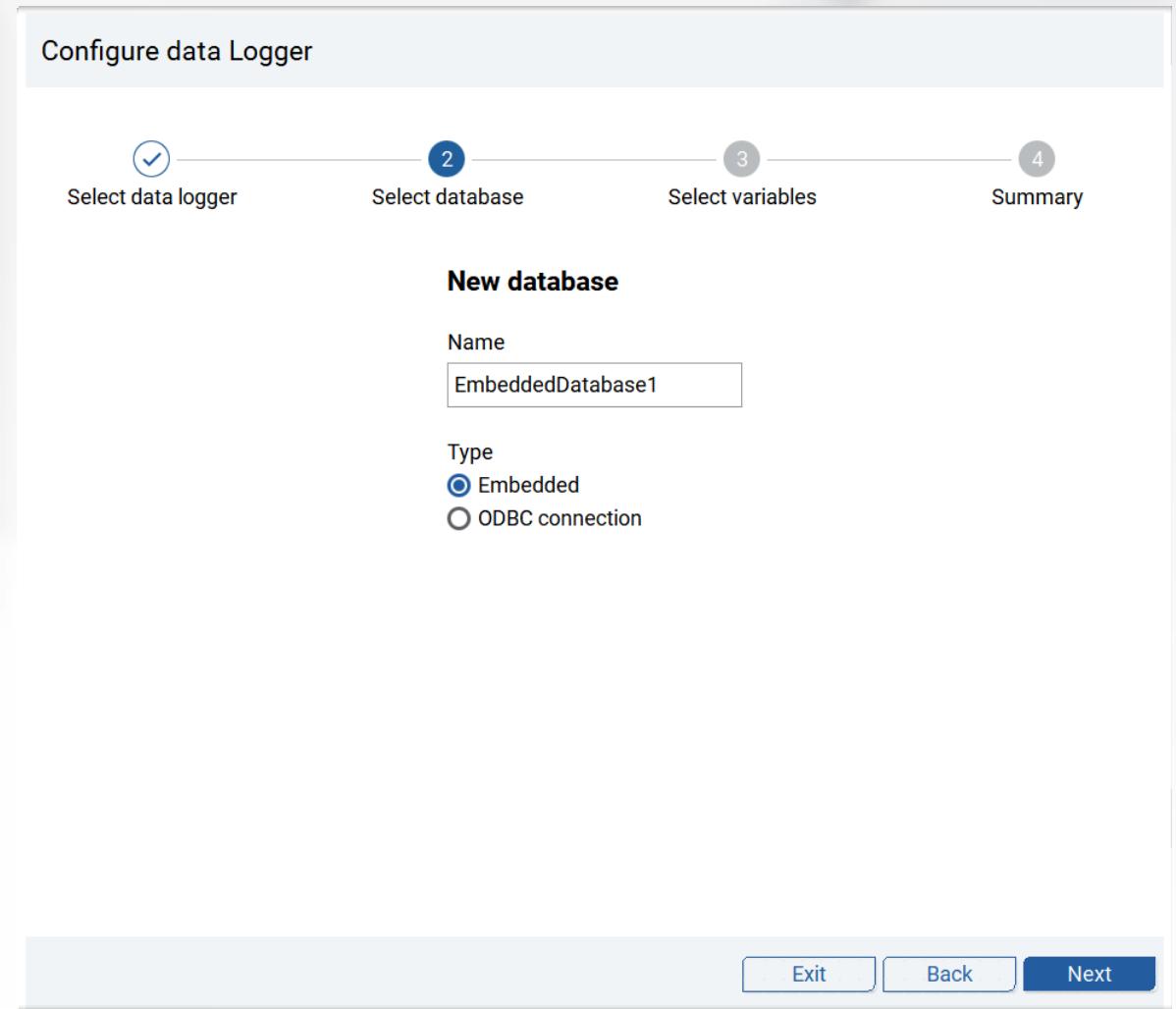
1 Select data logger 2 Select database 3 Select variables 4 Summary

New database

Name
EmbeddedDatabase1

Type
 Embedded
 ODBC connection

Exit Back Next



Create a simple HMI project in few steps

2. Configure a datalogger

- In memory: data are put in RAM, content of the DB is lost if the application is restarted

Configure data Logger

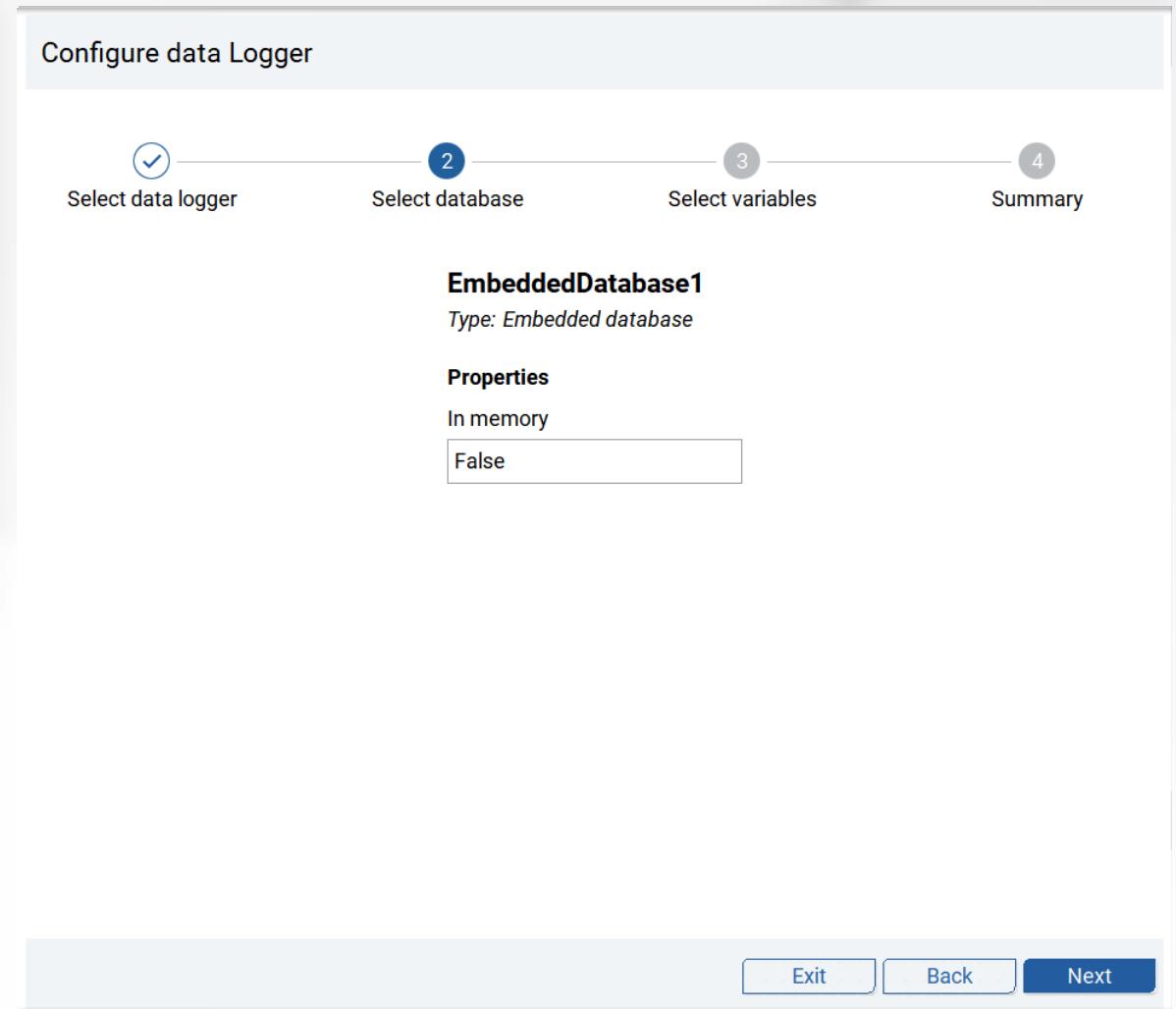
1 Select data logger 2 Select database 3 Select variables 4 Summary

EmbeddedDatabase1
Type: *Embedded database*

Properties

In memory
 False

Exit Back Next



Create a simple HMI project in few steps

2. Configure a datalogger

Configure data Logger

1 Select data logger 2 Select database 3 Select variables 4 Summary

Available databases

New database Delete

EmbeddedDatabase1

Name: EmbeddedDatabase1
Type: Embedded database

Properties

In memory: False

✓ Database EmbeddedDatabase1 successfully added X

Exit Back Next

Create a simple HMI project in few steps

2. Configure a datalogger

Configure data Logger

1 Select data logger 2 Select database 3 Select variables 4 Summary

Select variables

Type to search... 🔍 ✖

- ▼ 📁 Tank2
 - ▶ tag ValveInStatus
 - ▶ tag ValveInCmd
 - ▶ tag ValveOutStatus
 - ▶ tag ValveOutCmd
 - ▶ tag Level
 - ▶ tag SetPoint
- ▼ 📁 Tank1
 - ▶ tag ValveInStatus
 - ▶ tag ValveInCmd
 - ▶ tag ValveOutStatus
 - ▶ tag ValveOutCmd
 - ▶ tag Level

Exit Back Next

Create a simple HMI project in few steps

2. Configure a datalogger

Configure data Logger

1 Select data logger 2 Select database 3 Select variables 4 Summary

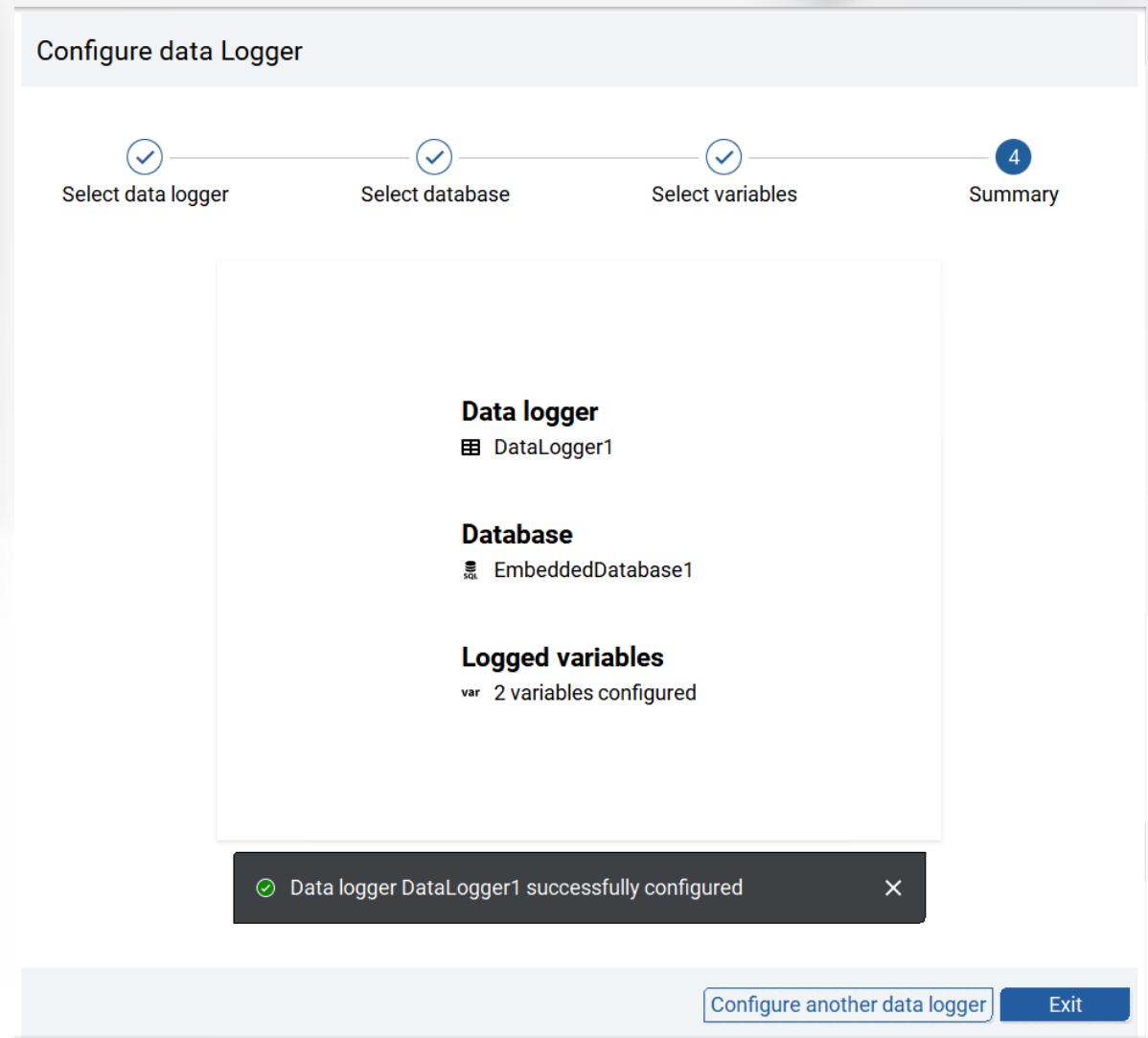
Data logger
DataLogger1

Database
EmbeddedDatabase1

Logged variables
2 variables configured

✓ Data logger DataLogger1 successfully configured

Configure another data logger Exit



Create a simple HMI project in few steps

3. Configure the user interface

Let's begin building your user interface



I want to configure
connected devices



I want to configure a data
logger



Preview

Configure my UI displays
starting with main window



I want to manage users and
groups

Create a simple HMI project in few steps

3. Configure the user interface

Preview

1 Configure UI
Select screen resolution, orientation and template

2 Customize
Select widgets, add title and style sheets

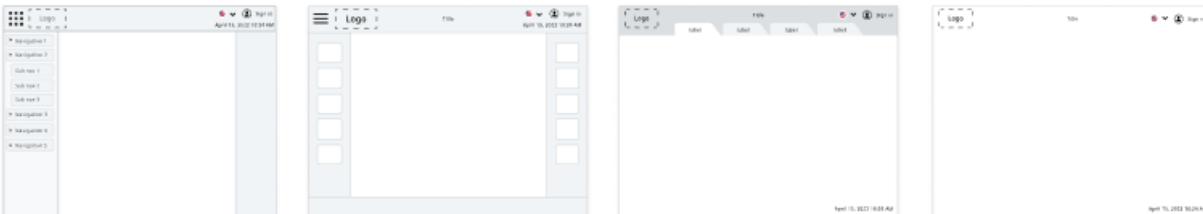
3 Preview & finish
Review and publish

What screen resolution do you want for your main window?
Responsive behavior is supported for scalability

3440 x 1440 1920 x 1080 1366 x 768 1024 x 768 Custom
Width 1920 px Height 1080 px

Orientation

What navigation style do you want?
Templates are editable



How many parent pages would you like to create?
Three children pages are included with each parent page.

Number of pages
- 000 +

How many tabs would you like to create?
Tabs are editable

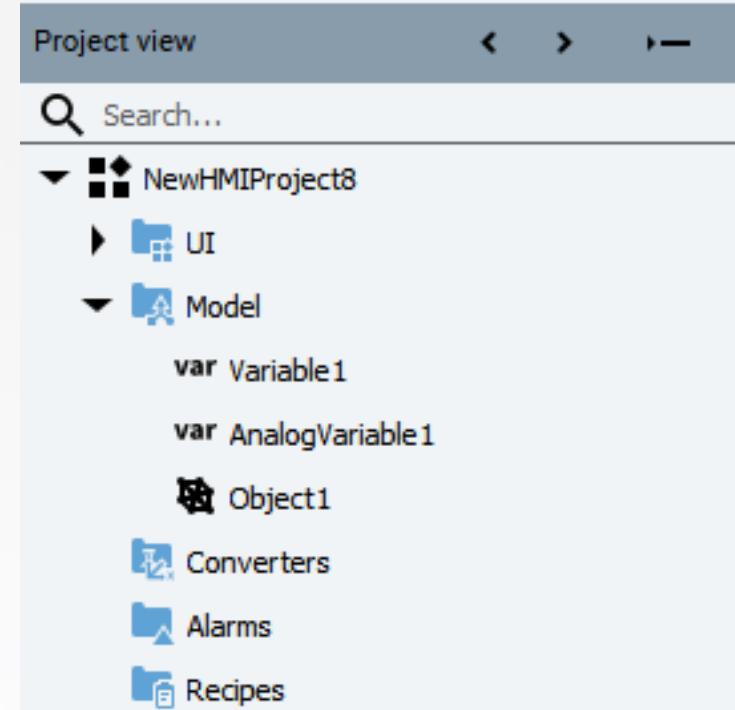
Number of tabs
- 000 +

Variables and tags



Model's variables

- What are the purposes of these Tags?
 - as "**internal variables**" used only on the HMI project (such as parameters or global variables)
 - as "**bridge**" between the Tag connected to the Comm Driver and the User Interface (such as using same UI for different PLC vendors)
- Under the "Model" node, it's possible to define different types of tags:
 - **Variable**: Standard Tag with desired data type
 - **Analog Variable**: Tag with Engineering Unit (used in combination with Measurement System)
 - **Object**: Structured Tag (a.k.a. User Defined Type)



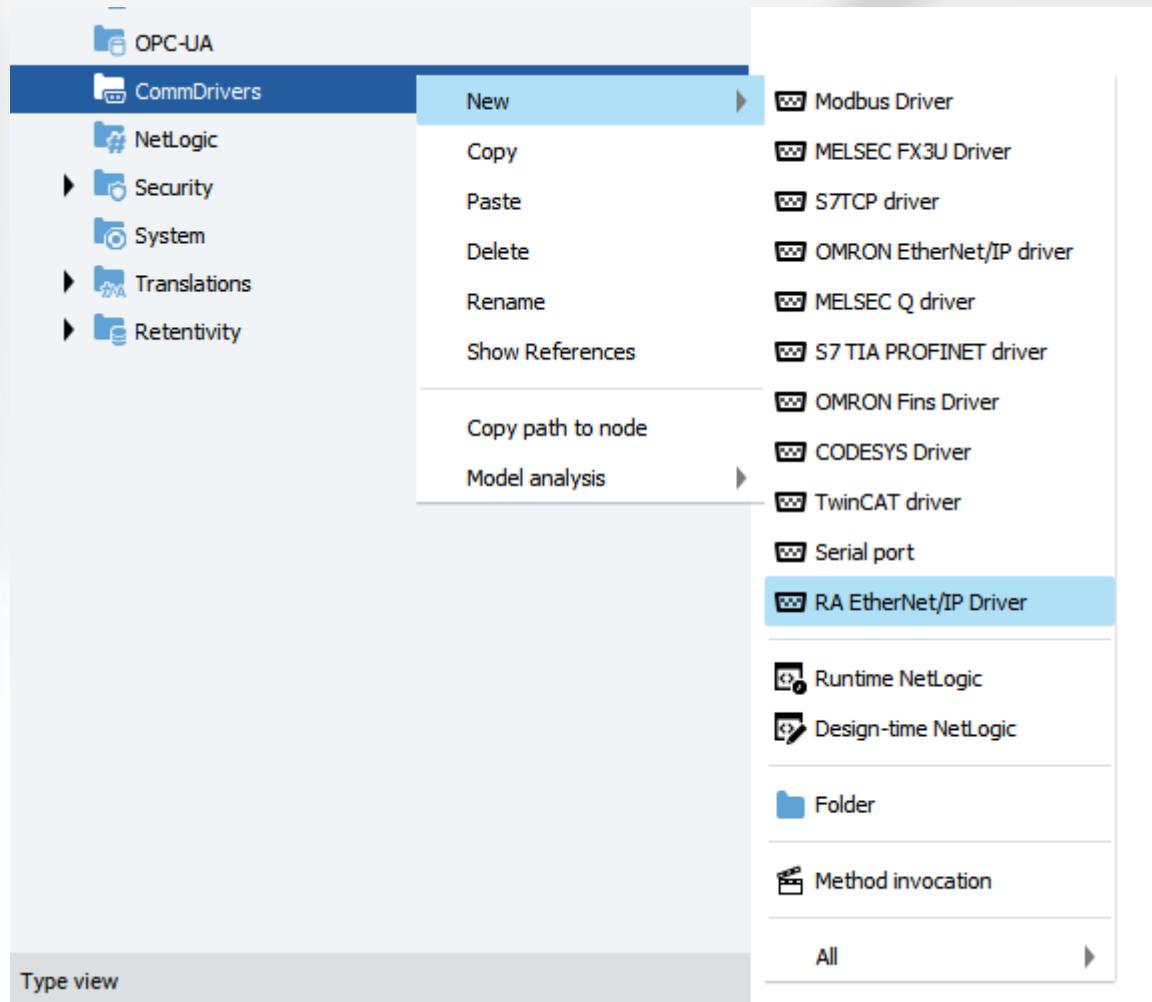
Communication drivers

- They allow to:
 - communicate with a PLC exchanging Tag values
 - import Tags from the PLC application (if supported by the protocol)
- To configure the communication,
 1. Add a Communication Driver object that represents the communication protocol. Some drivers (like Modbus) have a choice between serial and eth.
 2. Add one (or more) Station object that represents PLC and configure communication parameters

Communication driver	Online tags import	Offline tags import
Modbus Driver		
MELSEC FX3U Driver		✓
S7TCP driver	✓	✓
OMRON EtherNet_IP driver		✓
MELSEC Q driver		✓
S7 TIA PROFINET driver	✓	✓
OMRON Fins Driver		✓
CODESYS Driver	✓	✓
TwinCAT driver	✓	✓
Serial port		
RA EtherNet_IP Driver	✓	✓

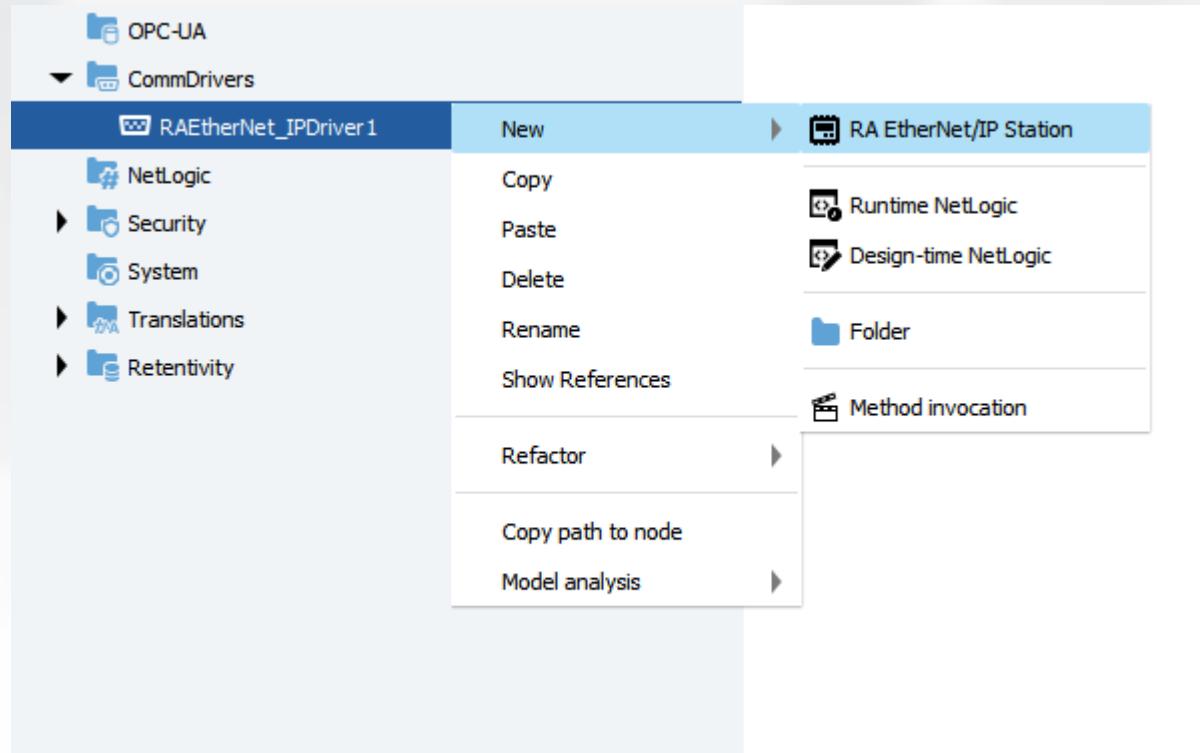
Communication drivers

- They allow to:
 - communicate with a PLC exchanging Tag values
 - import Tags from the PLC application (if supported by the protocol)
- To configure the communication,
 1. Add a Communication Driver object that represents the communication protocol. Some drivers (like Modbus) have a choice between serial and eth.
 2. Add one (or more) Station object that represents PLC and configure communication parameters



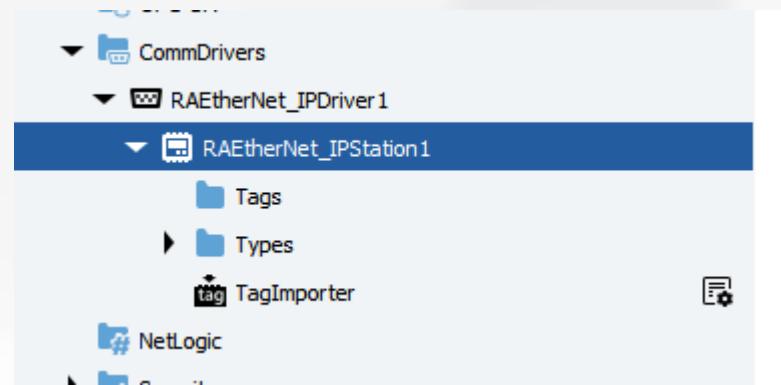
Communication drivers

- They allow to:
 - communicate with a PLC exchanging Tag values
 - import Tags from the PLC application (if supported by the protocol)
- To configure the communication,
 1. Add a Communication Driver object that represents the communication protocol. Some drivers (like Modbus) have a choice between serial and eth.
 2. Add one (or more) Station object that represents PLC and configure communication parameters



Communication drivers

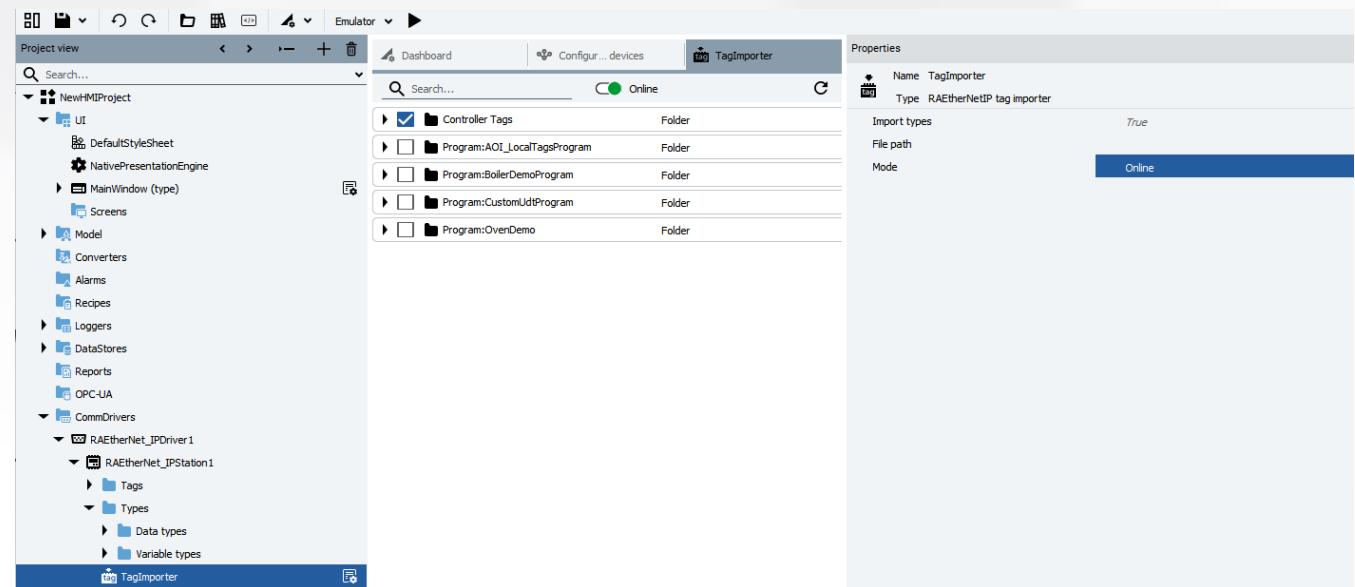
- They allow to:
 - communicate with a PLC exchanging Tag values
 - import Tags from the PLC application (if supported by the protocol)
- To configure the communication,
 1. Add a Communication Driver object that represents the communication protocol. Some drivers (like Modbus) have a choice between serial and eth.
 2. Add one (or more) Station object that represents PLC and configure communication parameters



Add tags exchanged with plc: tag importer

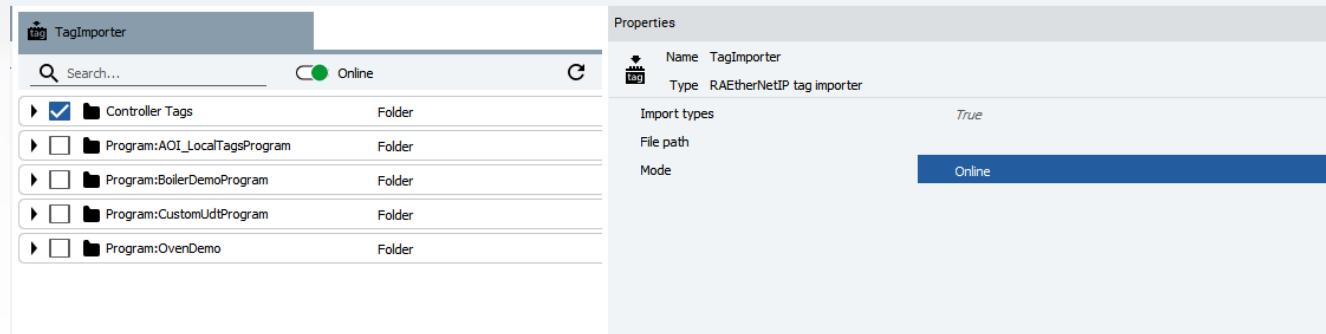
- Tag Importer
 - *Offline*: from an export file generated by PLC programming software
 - *Online*: reading tags directly from an online PLC (if supported by the comm driver)

- Remind!
 - the number of tags exchanged with PLC are not counted by the license manager



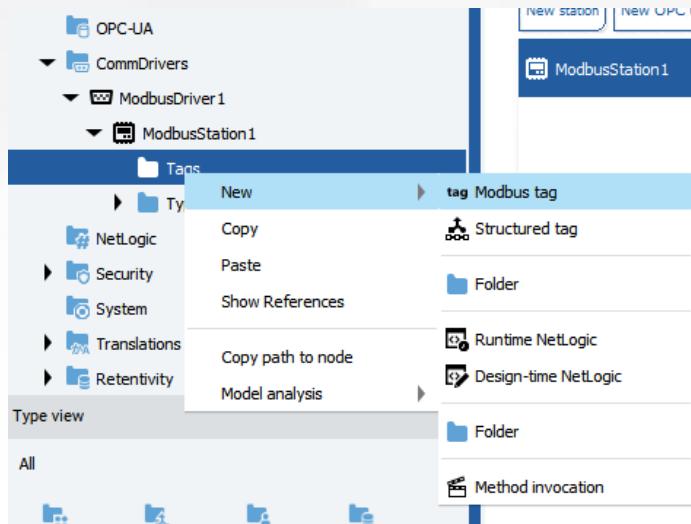
Add tags exchanged with plc: tag importer

- Tag Importer
 - *Offline*: from an export file generated by PLC programming software
 - *Online*: reading tags directly from an online PLC (if supported by the comm driver)
- Remind!
 - the number of tags exchanged with PLC are not counted by the license manager



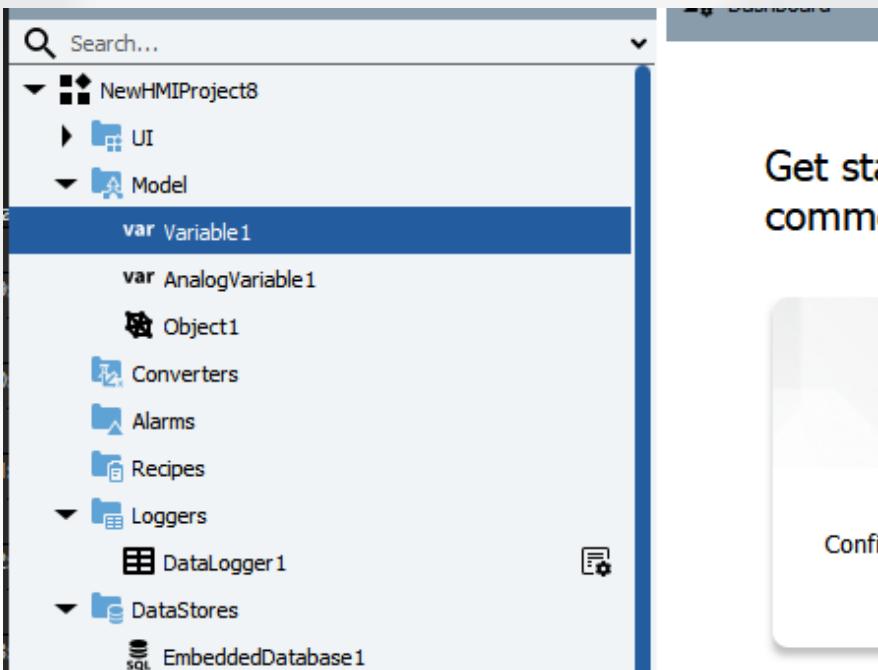
Add tags exchanged with plc: manually

- for Communication Drivers not using symbolic names (i.e. Modbus TCP)
- tags can be created “manually”
 - Right-click on the Tags node and select the "New Variable" command
 - on the Properties panel, configure the tag



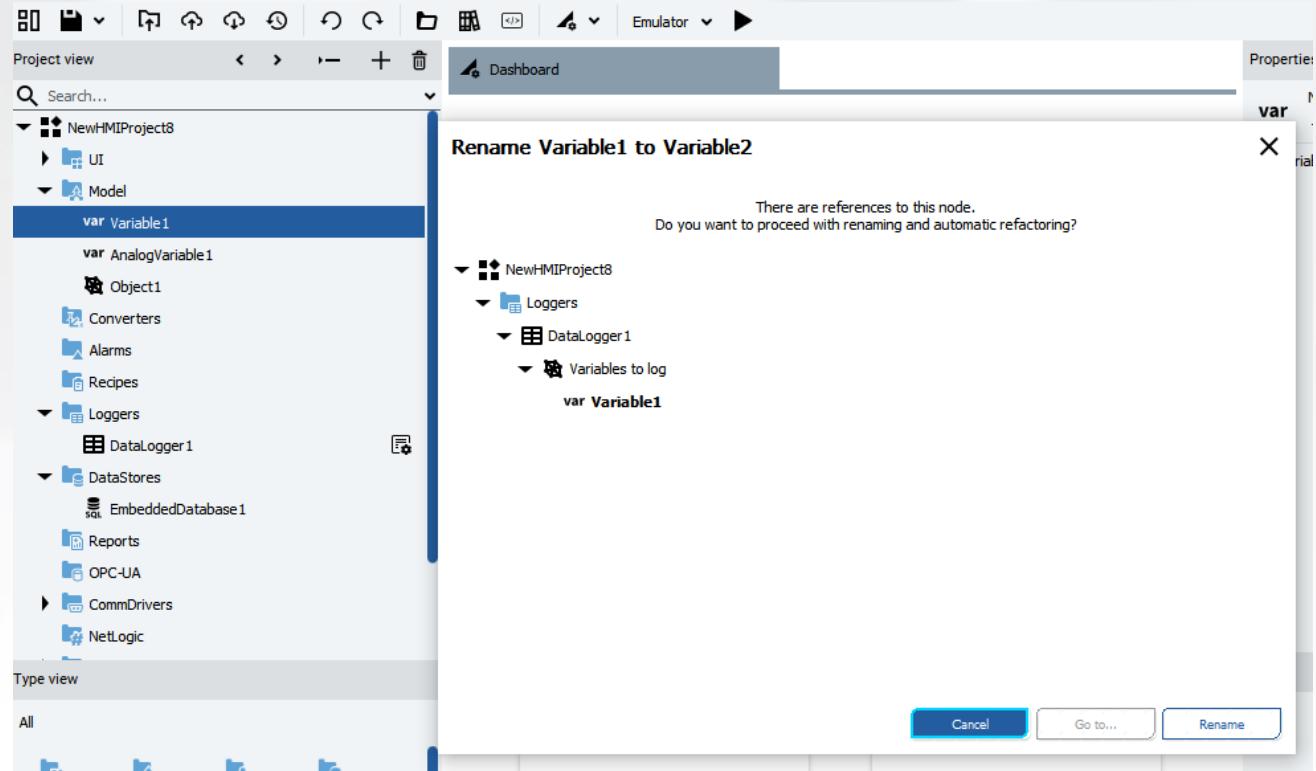
Tag rename

- Tags can be renamed
 - Applies to Model's Tags
 - Applies to Tags exchanged with PLC created manually and also imported!
 - If the Tag is referenced by a resource that requires refactoring (example: a datalogger), a popup will appear automatically
 - If it has been imported, the Tag will be marked as "Renamed"



Tag rename

- Tags can be renamed
 - Applies to Model's Tags
 - Applies to Tags exchanged with PLC created manually and also imported!
 - If the Tag is referenced by a resource that requires refactoring (example: a datalogger), a popup will appear automatically
 - If it has been imported, the Tag will be marked as "Renamed"



Hands-on session

- Create some Model tags
- Add the RA Ethernet/IP communication driver
- Import tags from FactoryTalk Logix Echo



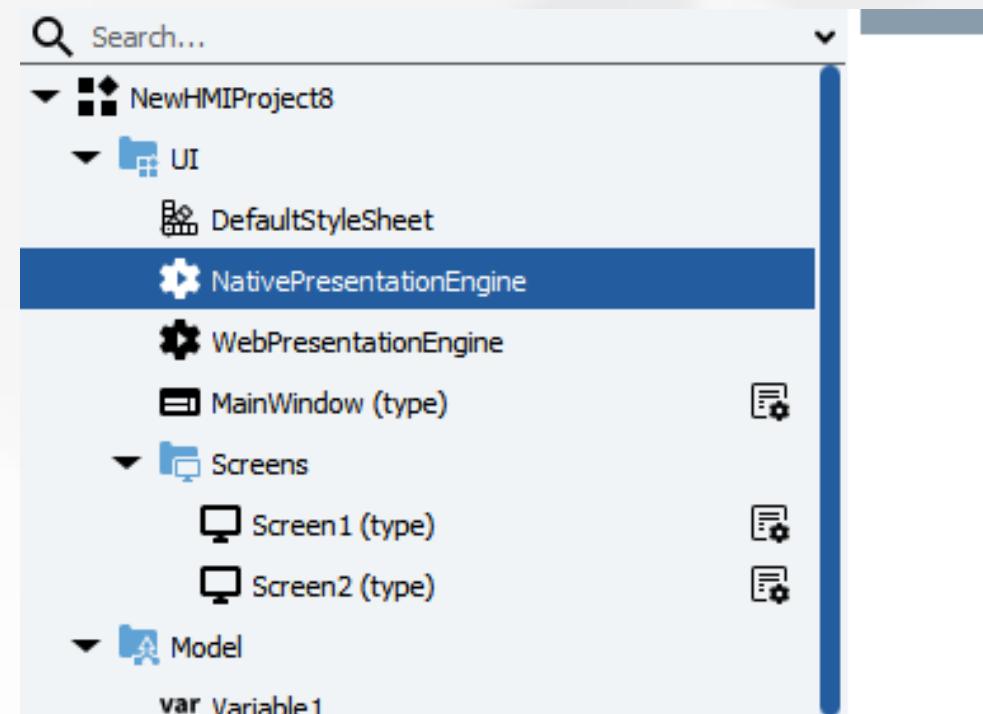


Develop the user interface



User interface (UI)

- Default Style Sheet
 - makes it possible to globally set some style properties of all graphical objects in the project
- Presentation Engine
 - **Native** Presentation Engine it's the object that create the user interface of the Application
 - **Web** Presentation Engine, it's the object that create the user interface for Web access
- Main Window
 - This object is the root container of graphical objects



Default style sheet

- Properties

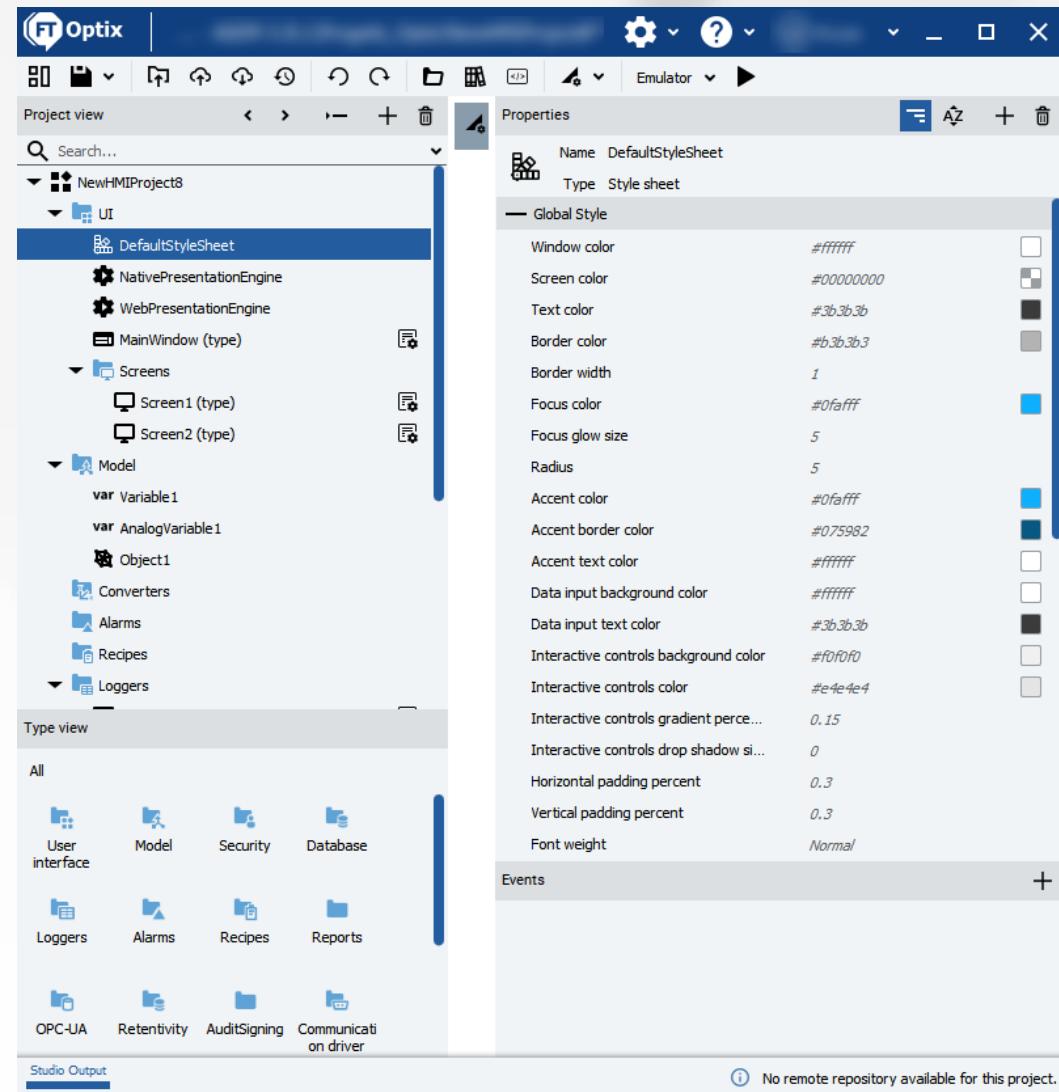
- Global Style: style properties in common with all objects (like radius)
- Control Styles: style properties for a specific type of object (Button, Switch...)

- Multiple style sheets

- into the same project
can be defined multiple style sheets
for example, to allow to switch
between a light/dark tone interface
at Runtime

- Template Libraries

- contain some pre-defined and
"ready to use" Style sheets



Default style sheet

- Properties

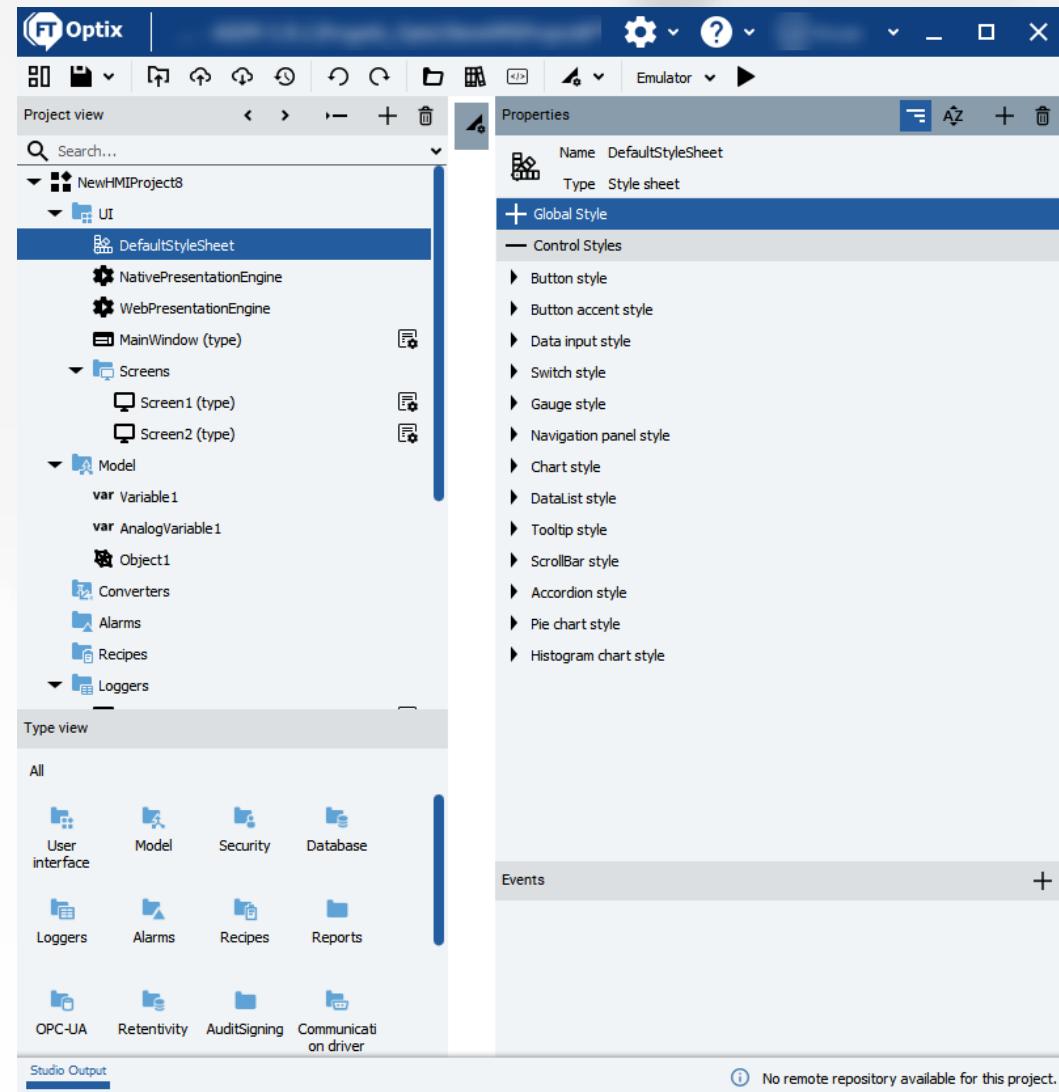
- Global Style: style properties in common with all objects (like radius)
- Control Styles: style properties for a specific type of object (Button, Switch...)

- Multiple style sheets

- into the same project
can be defined multiple style sheets
for example, to allow to switch
between a light/dark tone interface
at Runtime

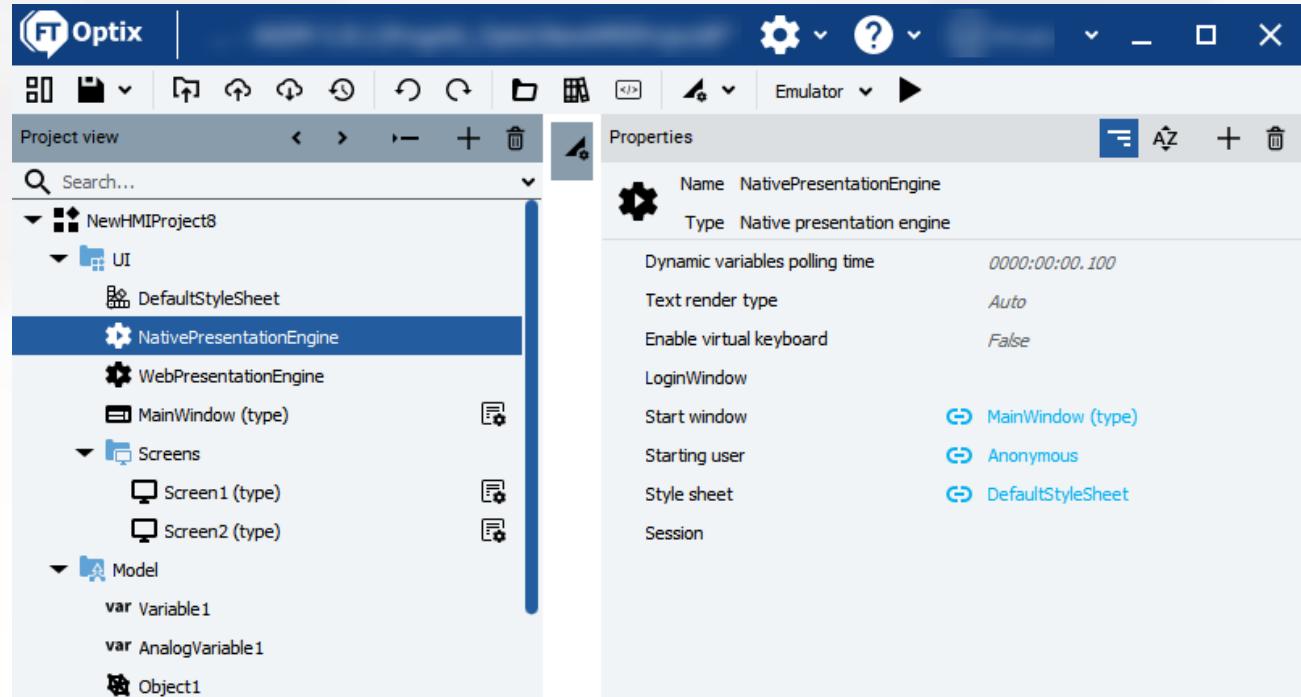
- Template Libraries

- contain some pre-defined and
"ready to use" Style sheets



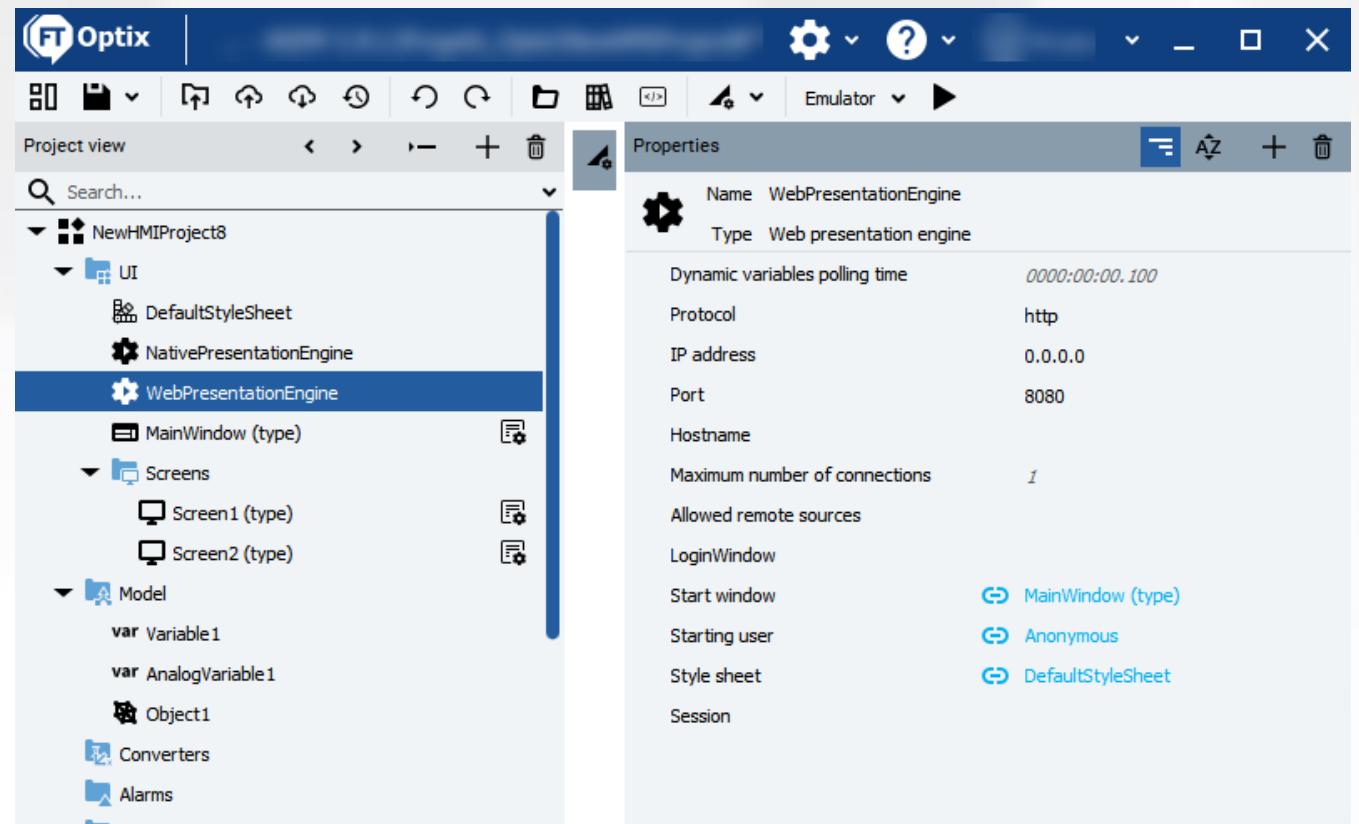
Native presentation engine

- Properties
 - Dynamic Variables polling time: used to refresh Tags used in Presentation Engine.
NOTE: this will also set the polling time used by comm drivers (for active tags)
- Start Window
- Starting User
- Style Sheet
- Enable Virtual Keyboard



Web presentation engine

- Allow publishing the UI through the integrated Web Server
- Properties:
 - Start Window, Starting User, Style Sheet...
 - **Protocol:** http/https
 - **IP Address:** this will be the IP address used by Web Server
0.0.0.0 = listen on all addresses
 - **Port:** define the listening port
 - Hostname
 - Maximum Number of connections
 - Allowed remote sources: external sources that can be embedded (websites)

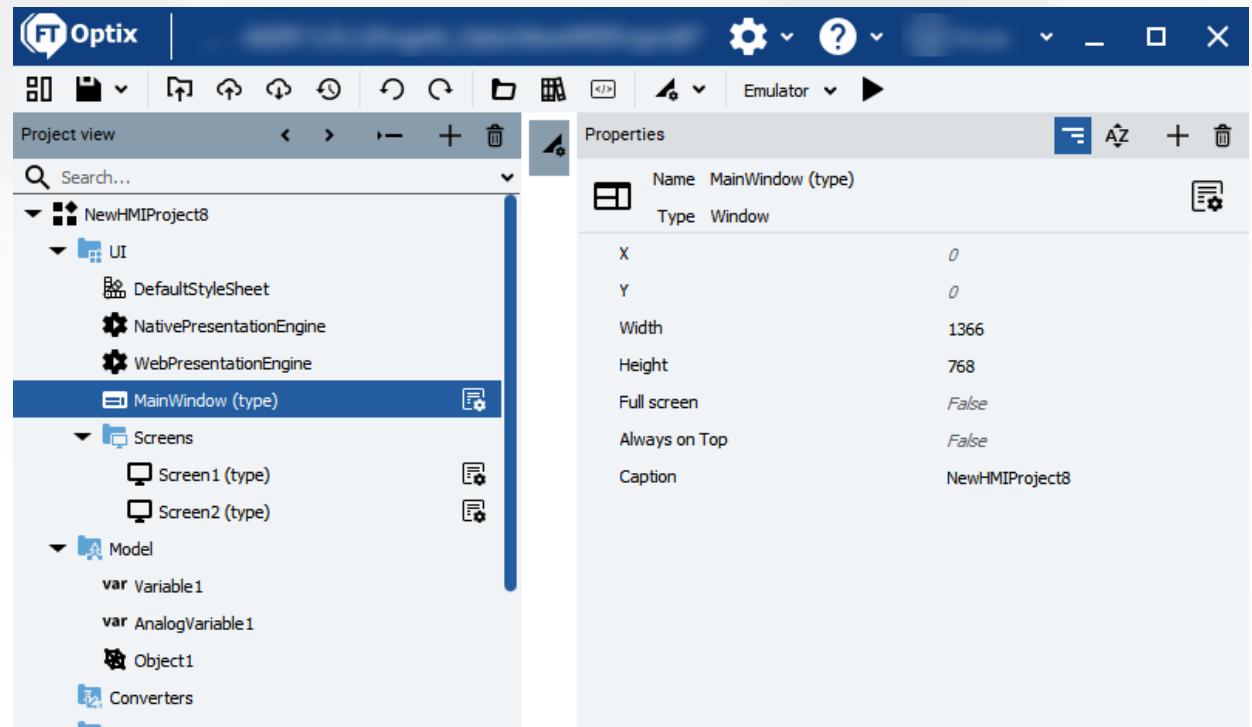


Main window

- Define the window where all graphical objects are showed
 - width/height: defines (manually) the size of the entire User Interface
 - Full Screen: set the size of UI equal to the current screen resolution hiding the title bar
 - Always on Top: force the window to be always on top of other applications
 - Caption: title of the window

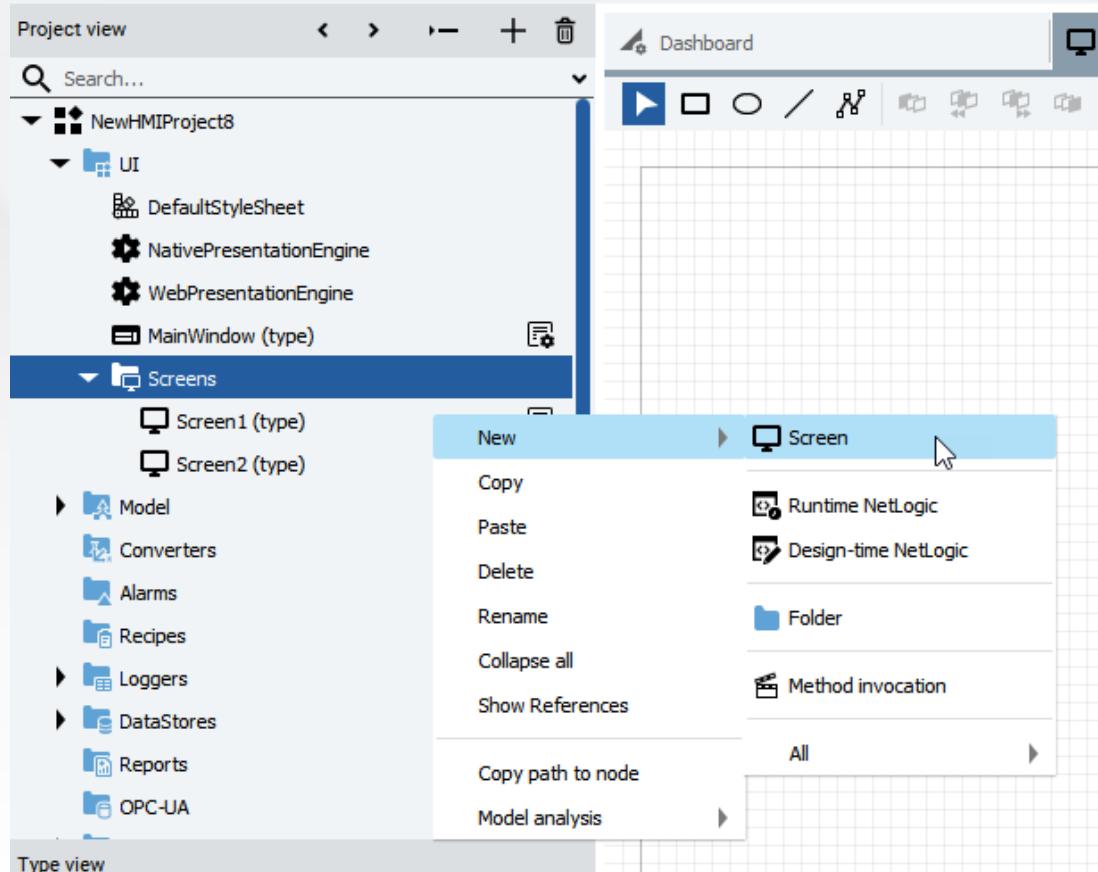
- NOTE:

- Application is a single-window application
- Switching between a page to another will not close/open a window, but will only change its content (panels)



Screens

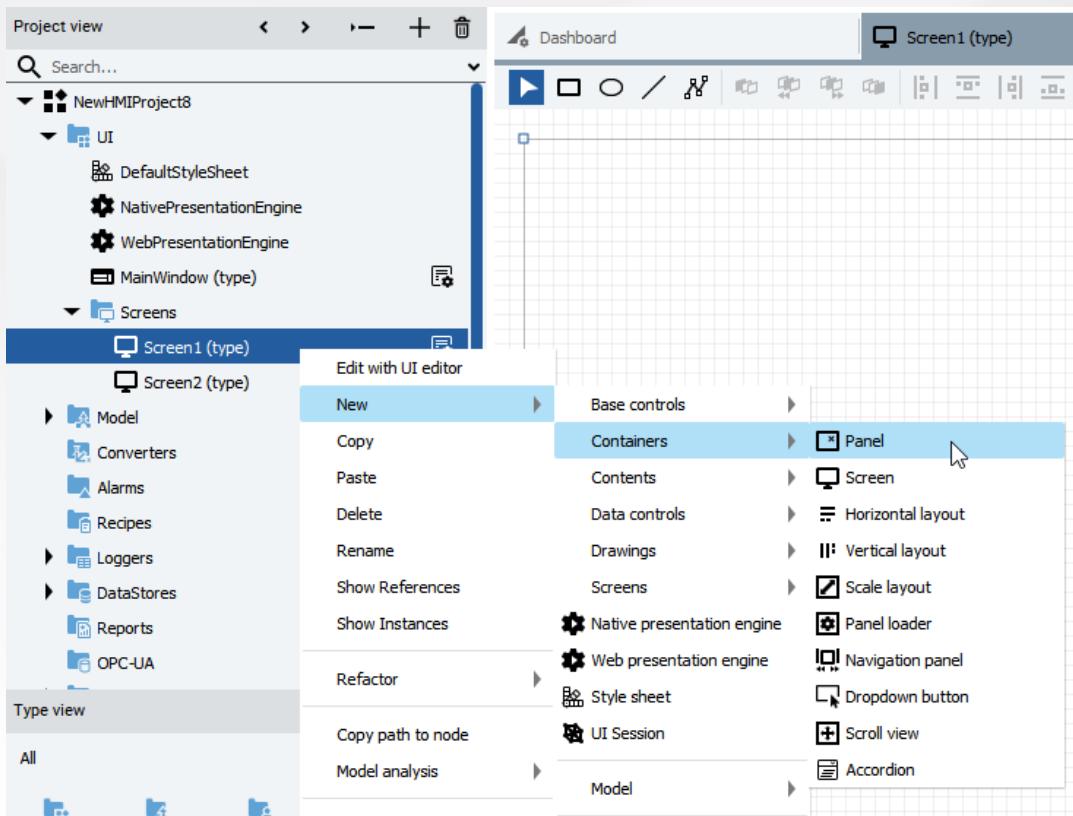
- Screens are the pages that are loaded to the running applications



- Screen sizes are stretched to the parent container (PanelLoader or NavigationPanel)
- Background can be customized with StyleSheet

Panels

- Panel is a Container that aggregates graphical objects (i.e. widgets)



- create a new Panel:
 - Right-click on the Page
 - select New > Containers > Panel

Navigation

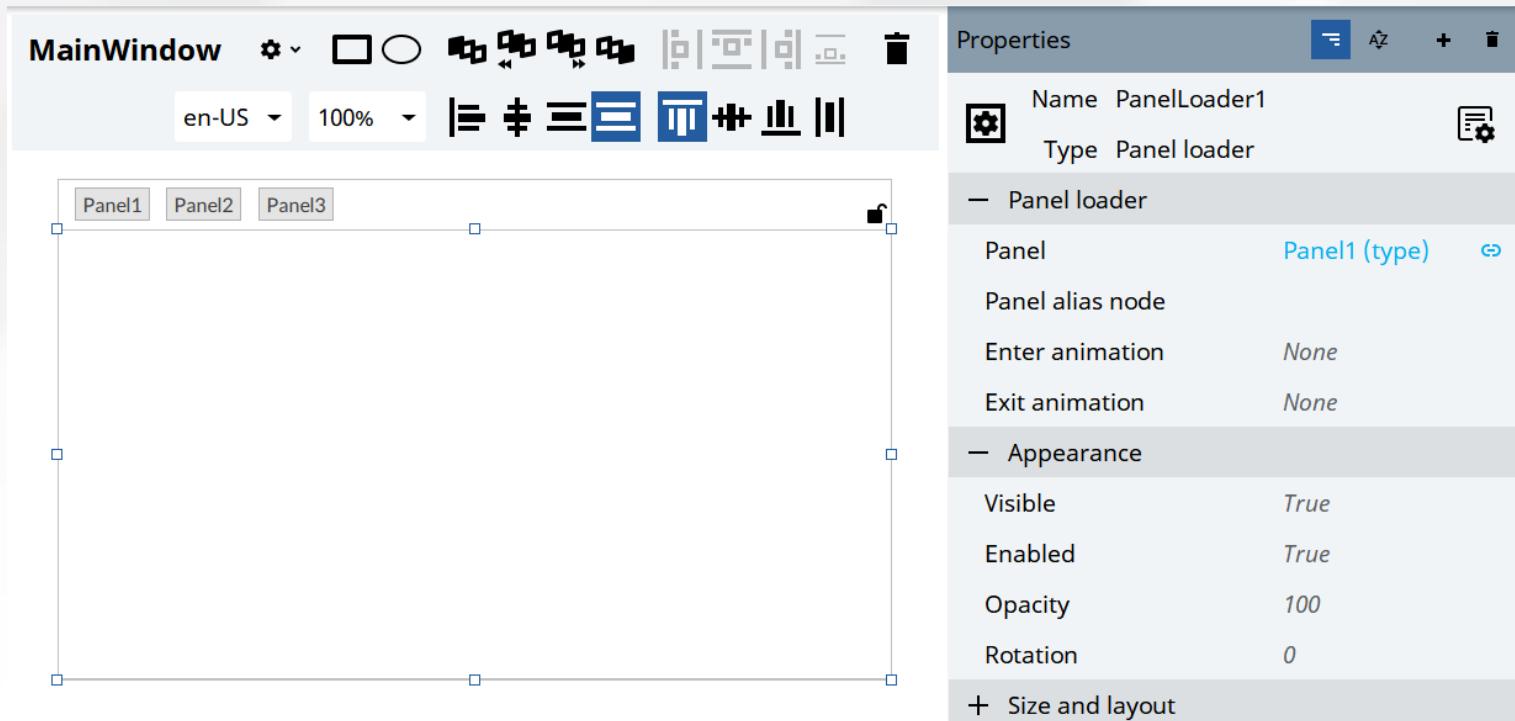
- Navigation can take place using two different ways

1. Panel Loader

- Container to display panels or screens
- "Change panel" method allow to load a different panel

2. Navigation Panel

- Container that automatically organizes Panels into navigable tabs.
- Just Drag&Drop Screens into the object
- Tabs can be hidden or disabled



Navigation

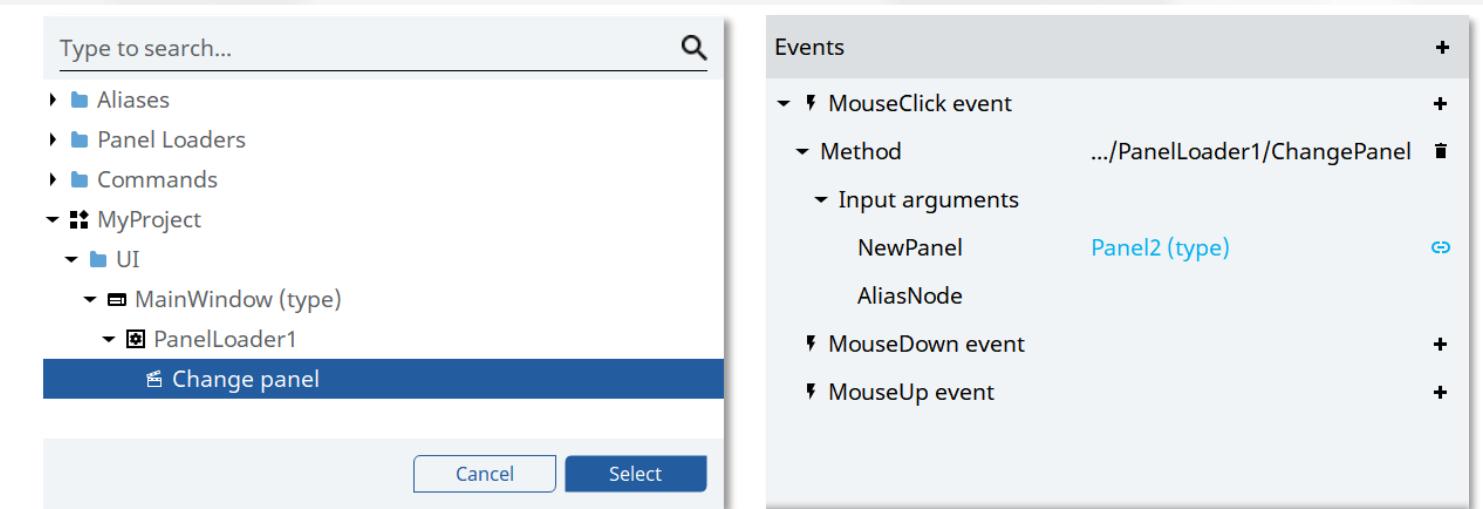
- Navigation can take place using two different ways

1. Panel Loader

- Container to display panels or screens
- "Change panel" method allow to load a different panel

2. Navigation Panel

- Container that automatically organizes Panels into navigable tabs.
- Just Drag&Drop Screens into the object
- Tabs can be hidden or disabled



Navigation

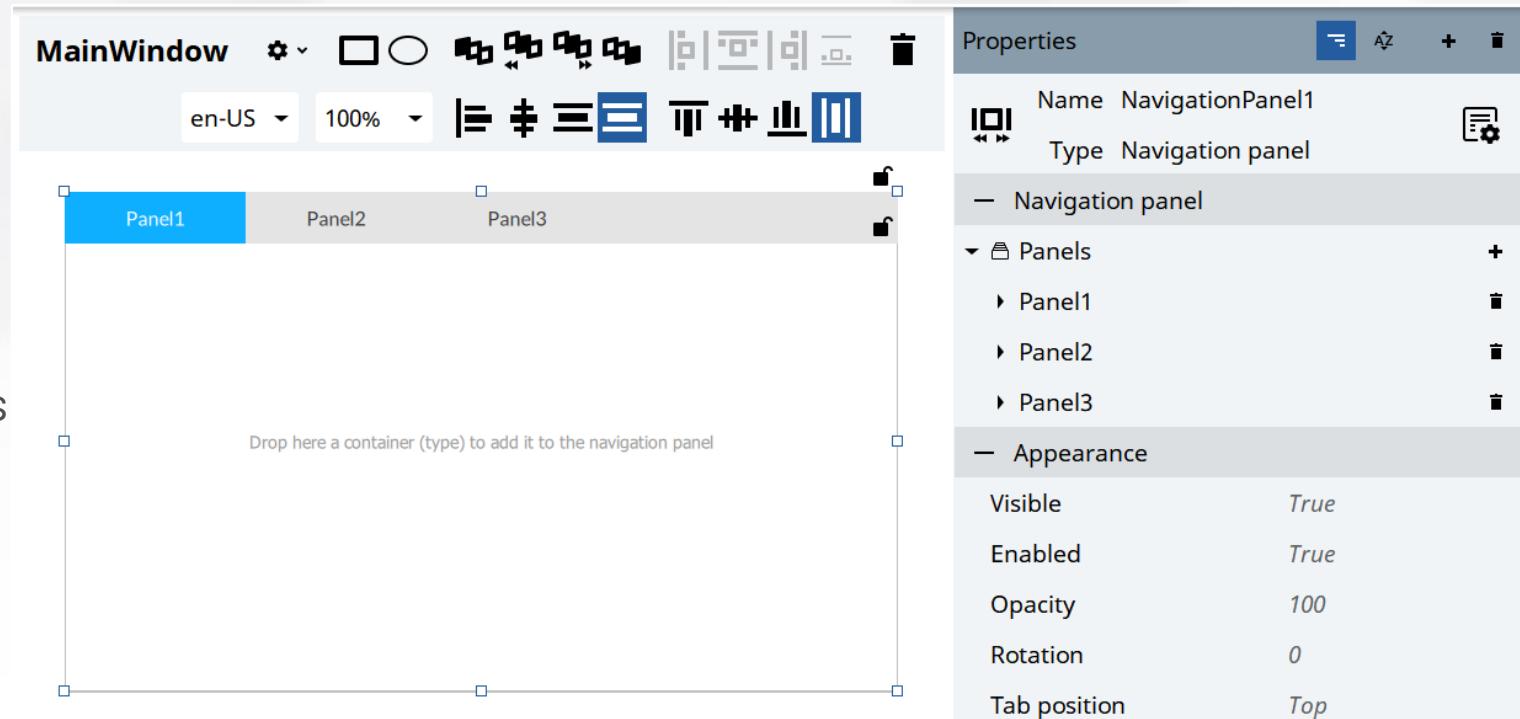
- Navigation can take place using two different ways

1. Panel Loader

- Container to display panels or screens
- "Change panel" method allow to load a different panel

2. Navigation Panel

- Container that automatically organizes Panels into navigable tabs.
- Just Drag&Drop Screens into the object
- Tabs can be hidden or disabled



Responsive design

- This can be achieved by using the **relative positioning** of objects or **scaling** them to be bigger or smaller

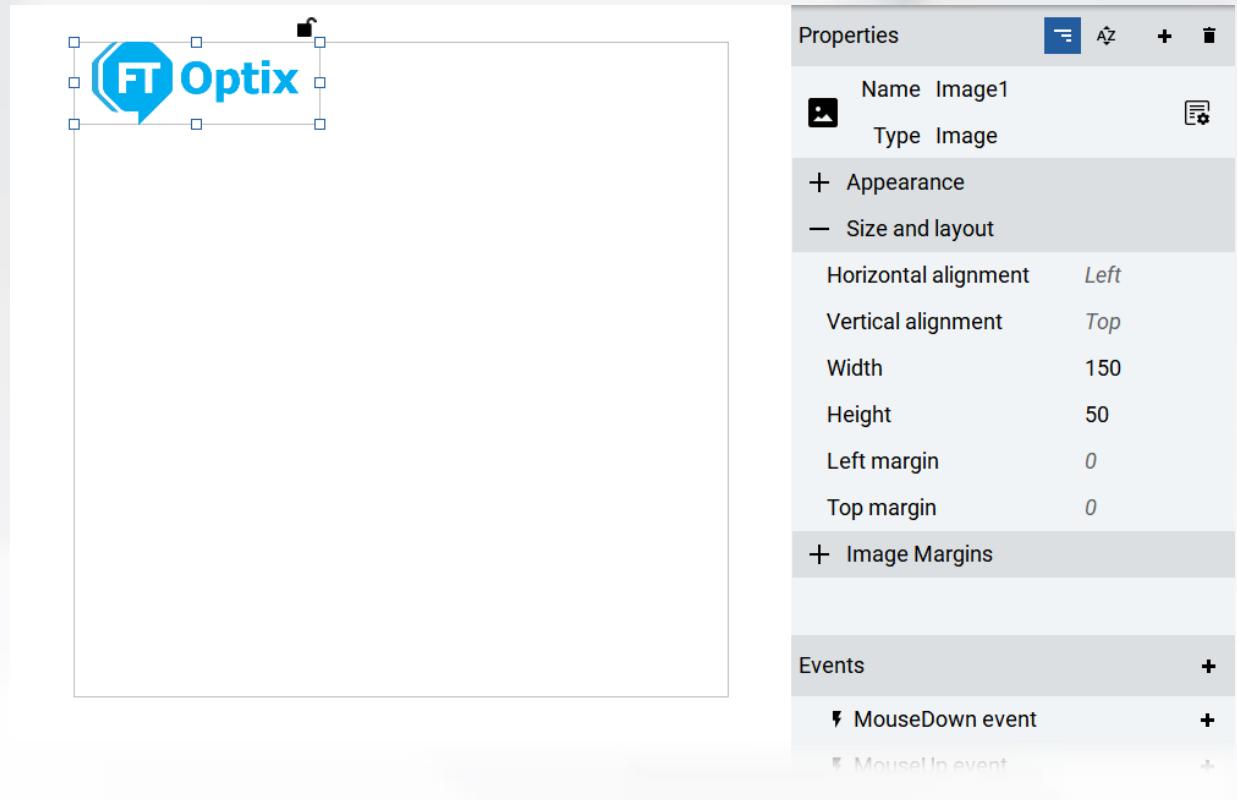
1. Alignment *: manage the reference

- Left/Right and Top/Bottom
- Center
- Stretch

2. Margins: manage the distance from the reference

- Horizontal Layout and Vertical Layout can be used for structured layouts

* Remember: to set Alignment on objects and also on all its containers: Panel, Navigation Panel...



Responsive design

- This can be achieved by using the **relative positioning** of objects or **scaling** them to be bigger or smaller

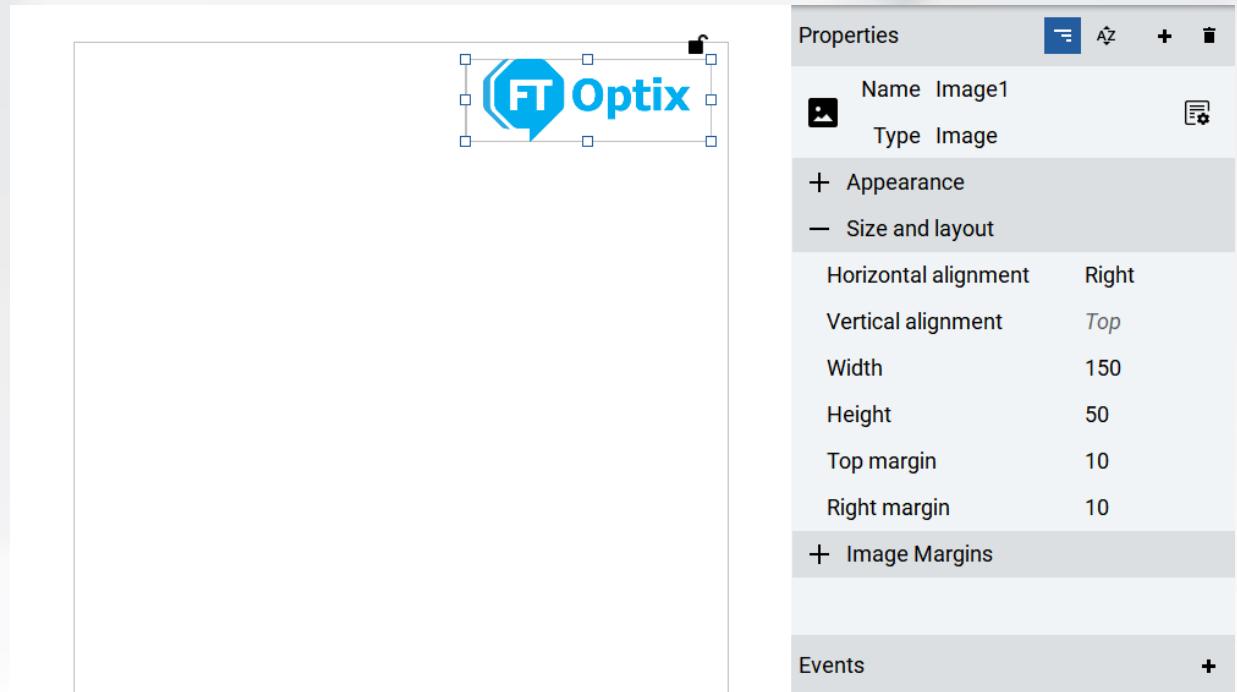
1. Alignment *: manage the reference

- Left/Right and Top/Bottom
- Center
- Stretch

2. Margins: manage the distance from the reference

- Horizontal Layout and Vertical Layout can be used for structured layouts

* Remember: to set Alignment on objects and also on all its containers: Panel, Navigation Panel...



Responsive design

- This can be achieved by using the **relative positioning** of objects or **scaling** them to be bigger or smaller

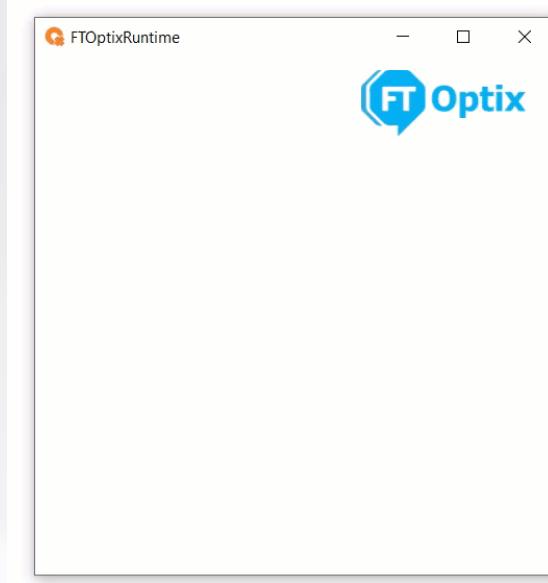
1. Alignment *: manage the reference

- Left/Right and Top/Bottom
- Center
- Stretch

2. Margins: manage the distance from the reference

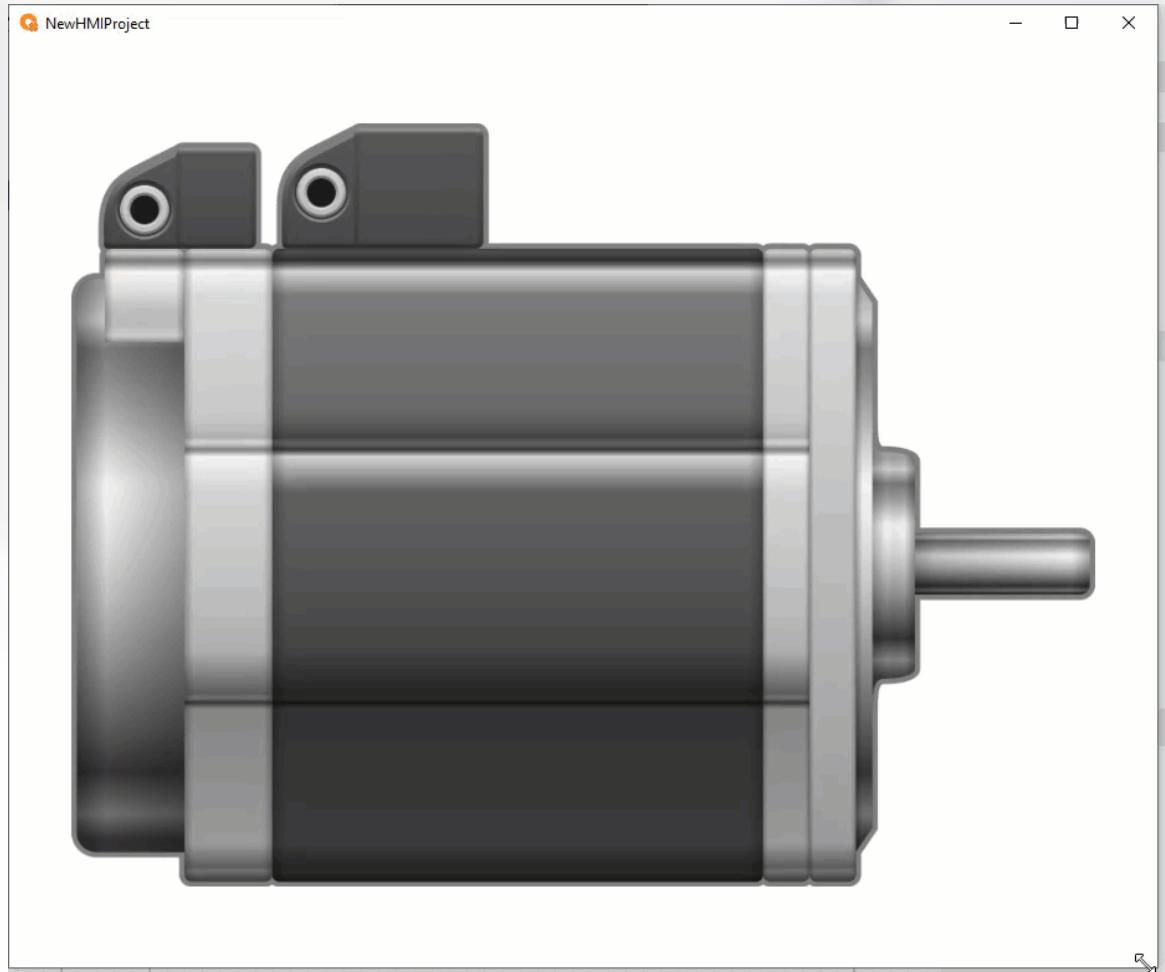
- Horizontal Layout and Vertical Layout can be used for structured layouts

* Remember: to set Alignment on objects and also on all its containers: Panel, Navigation Panel...



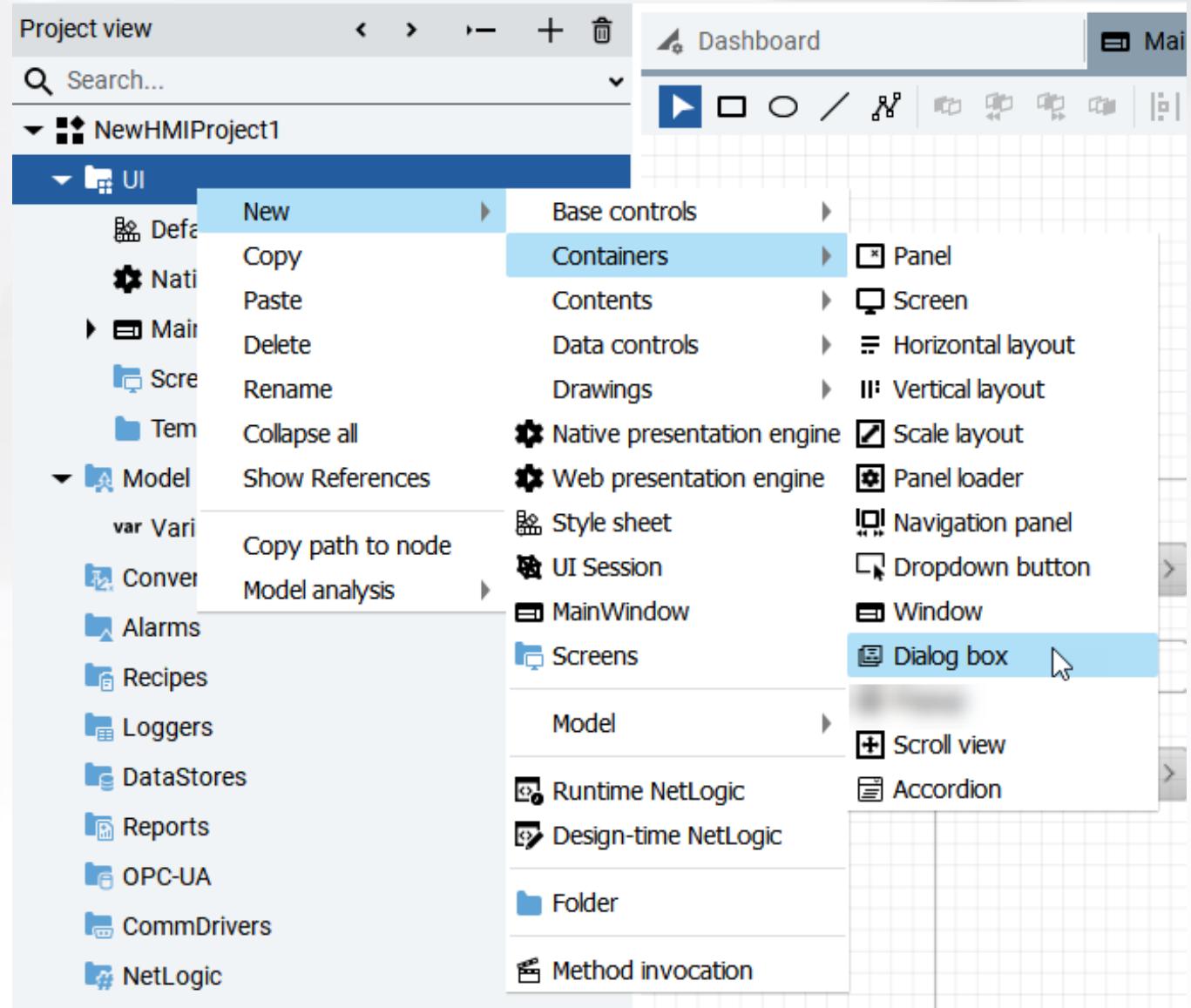
Scaling

- Objects can be adapted to the size of the container using a **ScaleLayout**
- Scaling ratio can be set to:
 - Fit: original ratio is maintained, empty bars can appear around the element if the container's ratio is different from the object's ratio
 - Stretch: object is stretch to the container size, ratio is ignored, graphics may get distorted



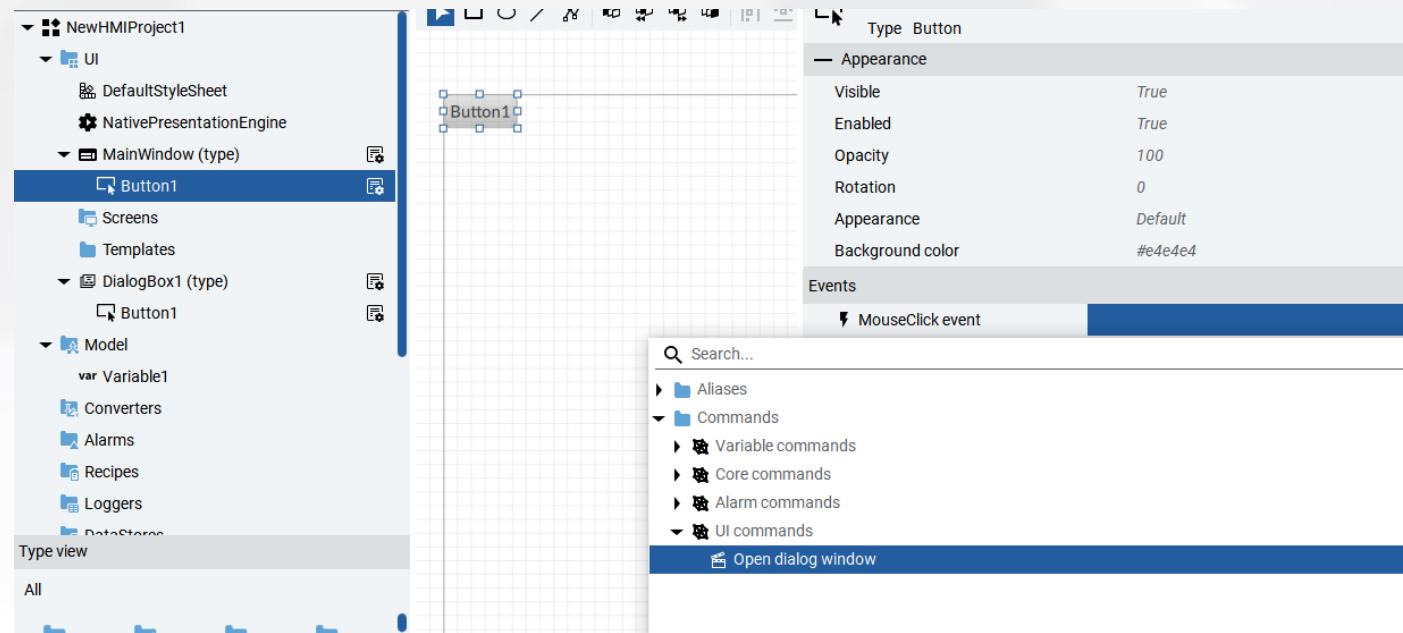
DialogBox

- Popup displays in Optix are called «DialogBox»
- DialogBox must be a type
- DialogBoxes are transparent by default
- **Note:** DialogBox does not open a new window on the device, it is a container on the highest Z-level of the MainWindow
- The «Close» method is exposed by the DialogBox itself



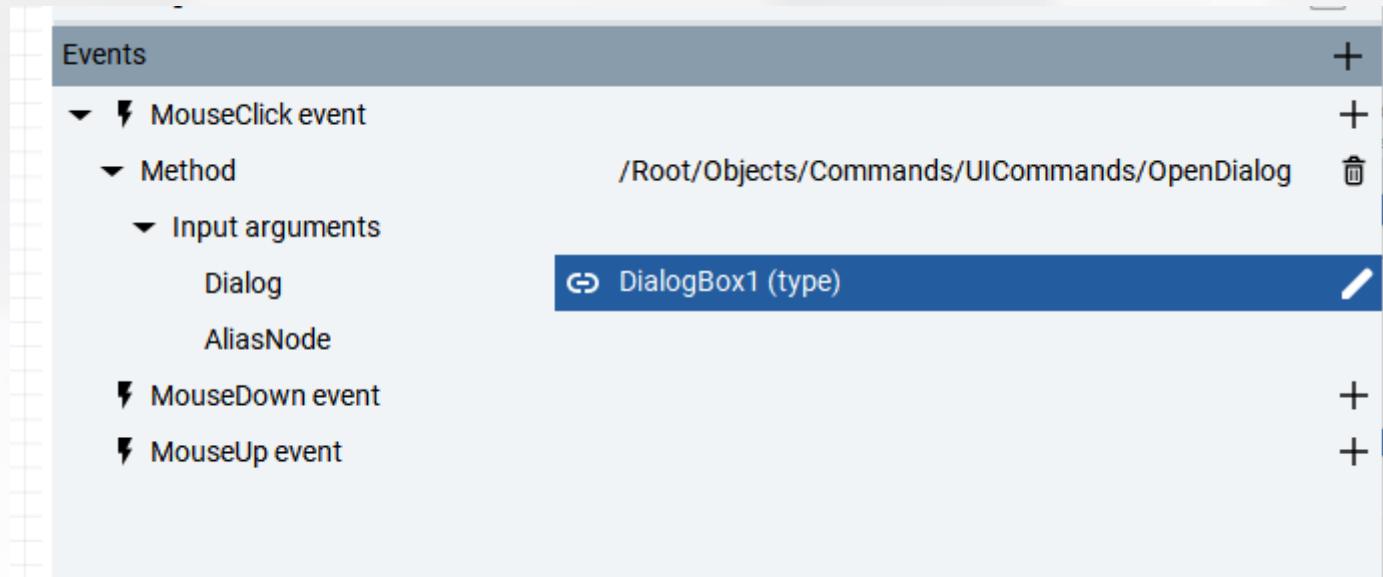
DialogBox

- Popup displays in Optix are called «DialogBox»
- DialogBox must be a type
- DialogBoxes are transparent by default
- **Note:** DialogBox does not open a new window on the device, it is a container on the highest Z-level of the MainWindow
- The «Close» method is exposed by the DialogBox itself



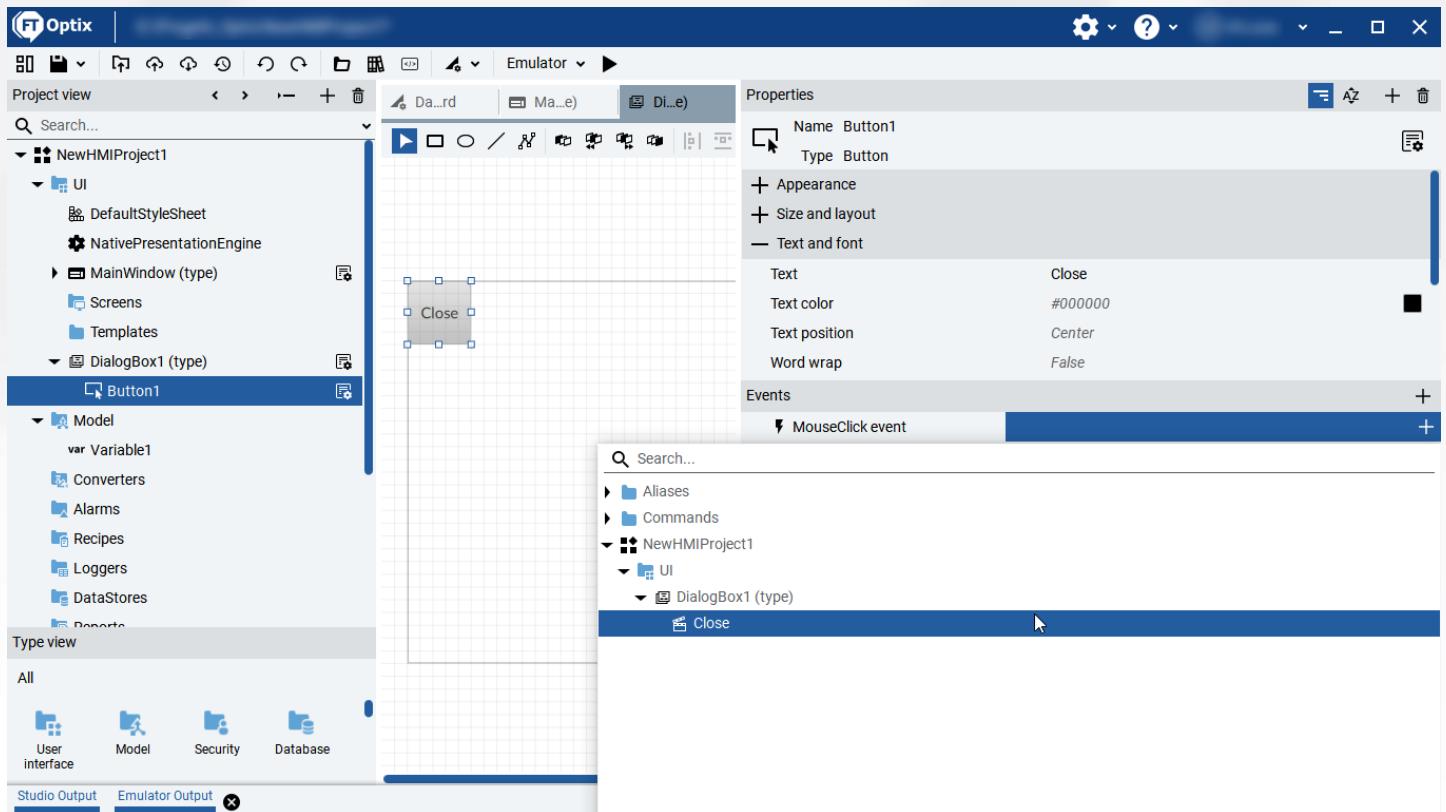
DialogBox

- Popup displays in Optix are called «DialogBox»
- DialogBox must be a type
- DialogBoxes are transparent by default
- **Note:** DialogBox does not open a new window on the device, it is a container on the highest Z-level of the MainWindow
- The «Close» method is exposed by the DialogBox itself



DialogBox

- Popup displays in Optix are called «DialogBox»
- DialogBox must be a type
- DialogBoxes are transparent by default
- **Note:** DialogBox does not open a new window on the device, it is a container on the highest Z-level of the MainWindow
- The «Close» method is exposed by the DialogBox itself



Hands-on session

- Create a simple HMI with two or three screens and a Navigation Panel
- Add a web presentation engine
and connect to the emulator using a browser
- Try to make the graphics responsive

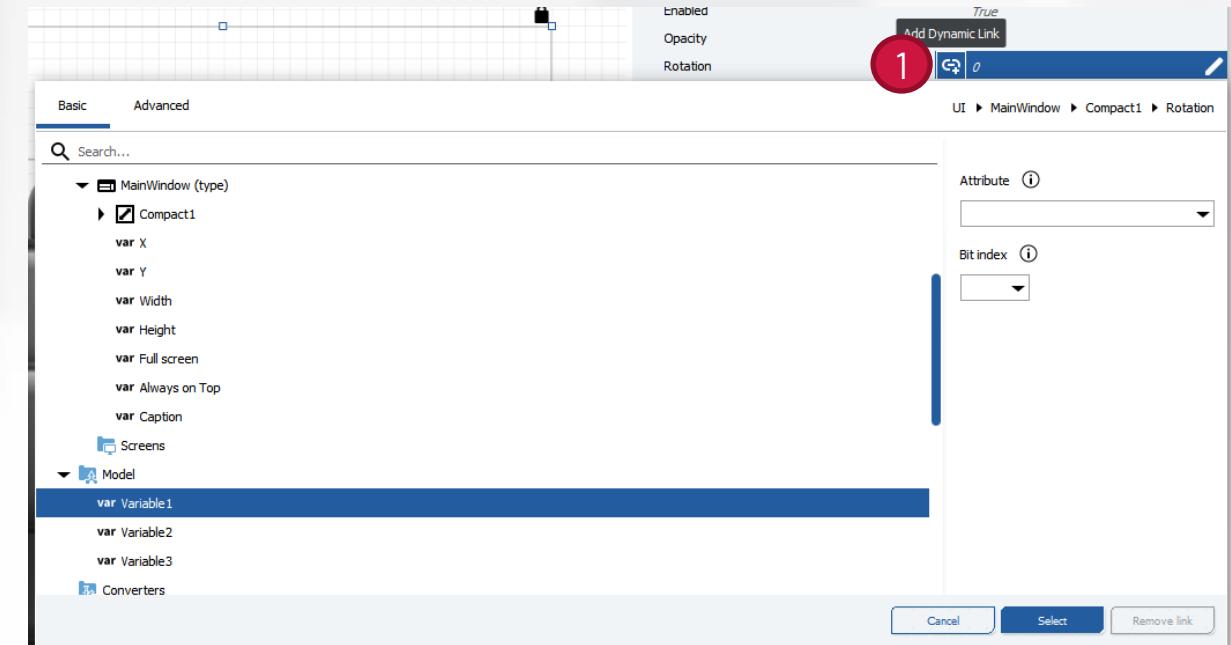


Dynamic links



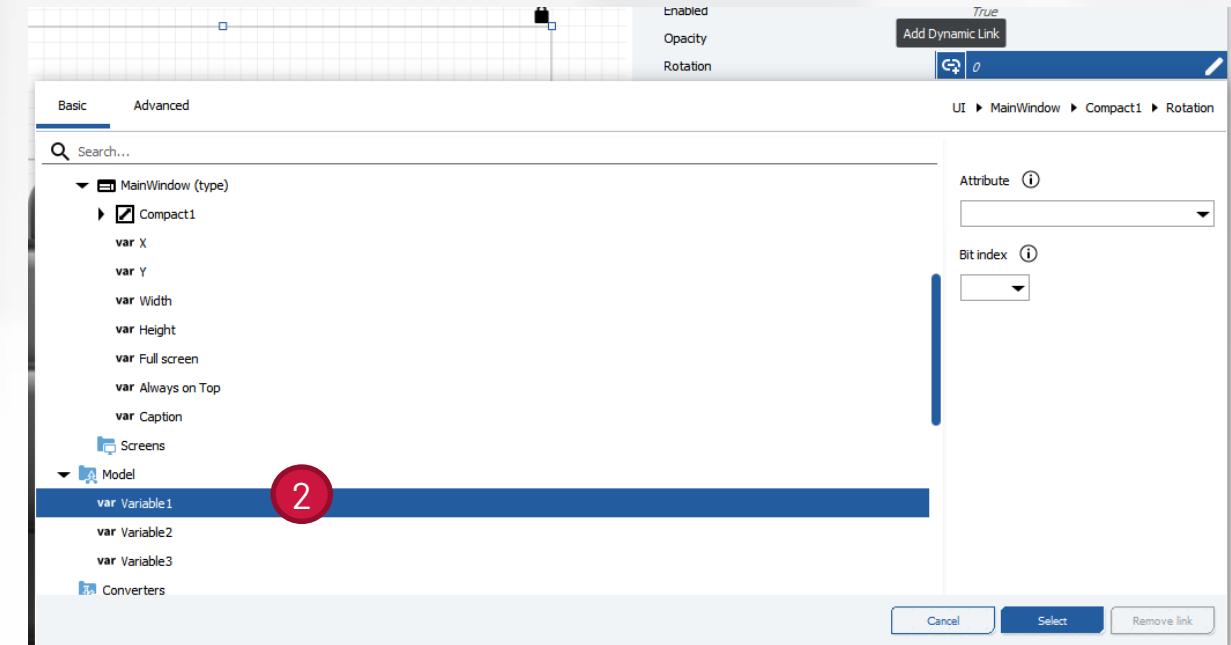
Dynamic links

- Dynamic Link is a connection between two nodes
 - Property connected to a Variable
absolute link = "Variable1"
 - Property connected to a Property of the same object
relative link = "../Enabled"
 - Property connected to a Property of another object
relative link = "../../Switch1/Checked"
- Can be created in different ways
 - Manual selection using the dynamic link window
 - Dragging the variable on a property
 - Dragging the variable on the graphical object
in this case, Studio sets the dynamic link
on the most suitable property, based on heuristics.



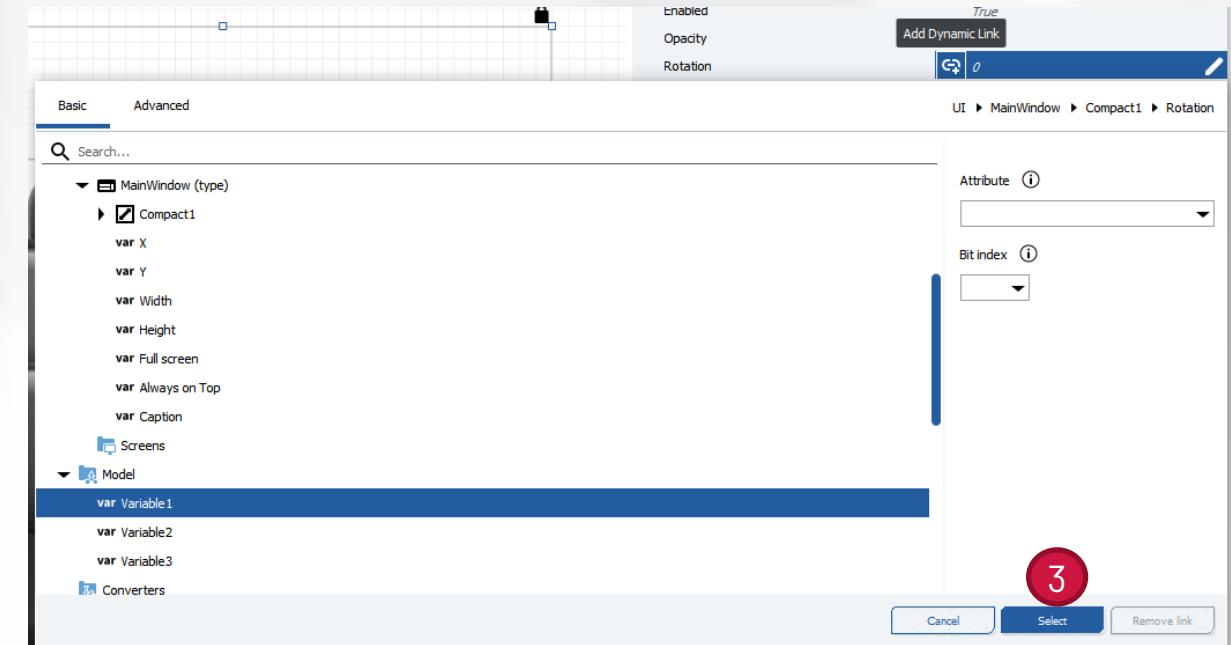
Dynamic links

- Dynamic Link is a connection between two nodes
 - Property connected to a Variable
absolute link = "Variable1"
 - Property connected to a Property of the same object
relative link = "../Enabled"
 - Property connected to a Property of another object
relative link = "../../Switch1/Checked"
- Can be created in different ways
 - Manual selection using the dynamic link window
 - Dragging the variable on a property
 - Dragging the variable on the graphical object
in this case, Studio sets the dynamic link
on the most suitable property, based on heuristics.



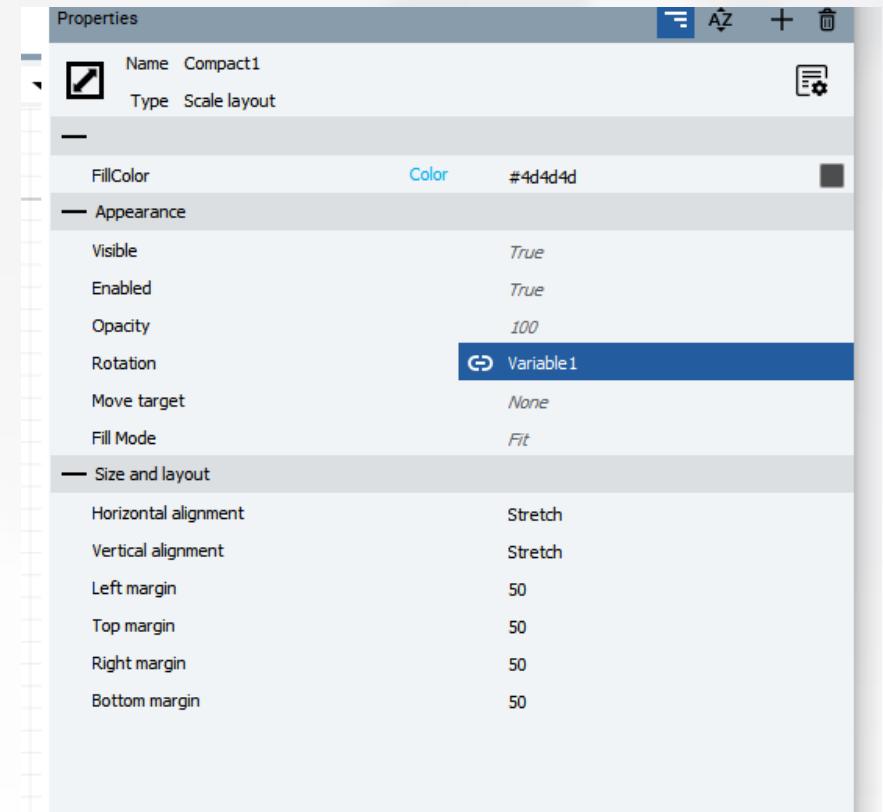
Dynamic links

- Dynamic Link is a connection between two nodes
 - Property connected to a Variable
absolute link = "Variable1"
 - Property connected to a Property of the same object
relative link = "../Enabled"
 - Property connected to a Property of another object
relative link = "../../Switch1/Checked"
- Can be created in different ways
 - Manual selection using the dynamic link window
 - Dragging the variable on a property
 - Dragging the variable on the graphical object
in this case, Studio sets the dynamic link
on the most suitable property, based on heuristics.



Dynamic links

- Dynamic Link is a connection between two nodes
 - Property connected to a Variable
absolute link = "Variable1"
 - Property connected to a Property of the same object
relative link = ".../Enabled"
 - Property connected to a Property of another object
relative link = ".../Switch1/Checked"
- Can be created in different ways
 - Manual selection using the dynamic link window
 - Dragging the variable on a property
 - Dragging the variable on the graphical object
in this case, Studio sets the dynamic link
on the most suitable property, based on heuristics.



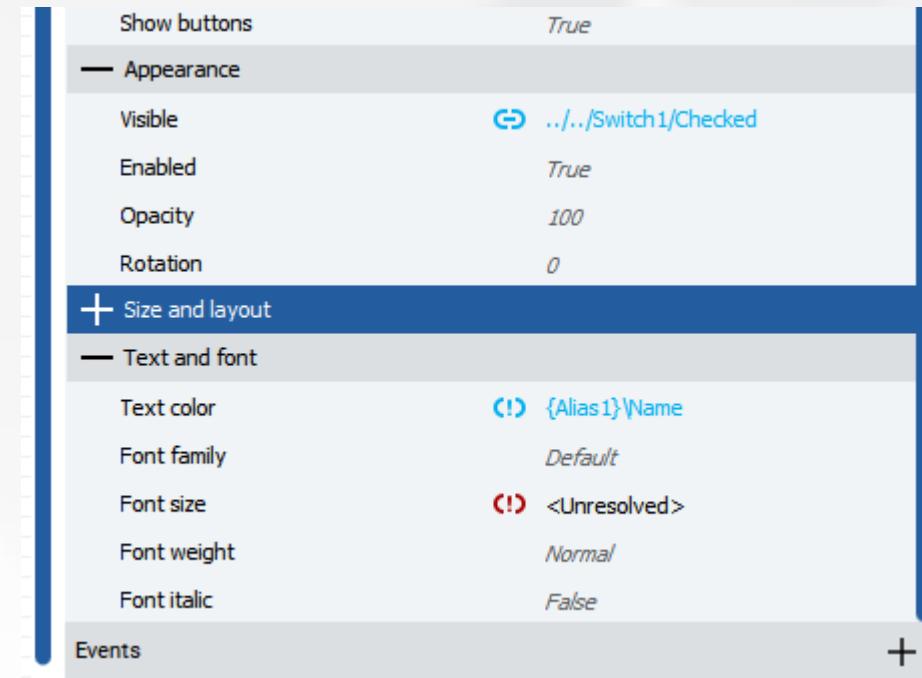
Dynamic links

- Dynamic Link is a connection between two nodes
 - Property connected to a Variable
absolute link = "Variable1"
 - Property connected to a Property of the same object
relative link = "../Enabled"
 - Property connected to a Property of another object
relative link = ".../Switch1/Checked"
- Can be created in different ways
 - Manual selection using the dynamic link window
 - Dragging the variable on a property
 - Dragging the variable on the graphical object
in this case, Studio sets the dynamic link
on the most suitable property, based on heuristics.

FillColor	Color	#4d4d4d
Appearance		
Visible		True
Enabled		True
Opacity		../../Width
Rotation		Variable1
Move target		None
Fill Mode		Fit
Size and layout		
Horizontal alignment		Stretch

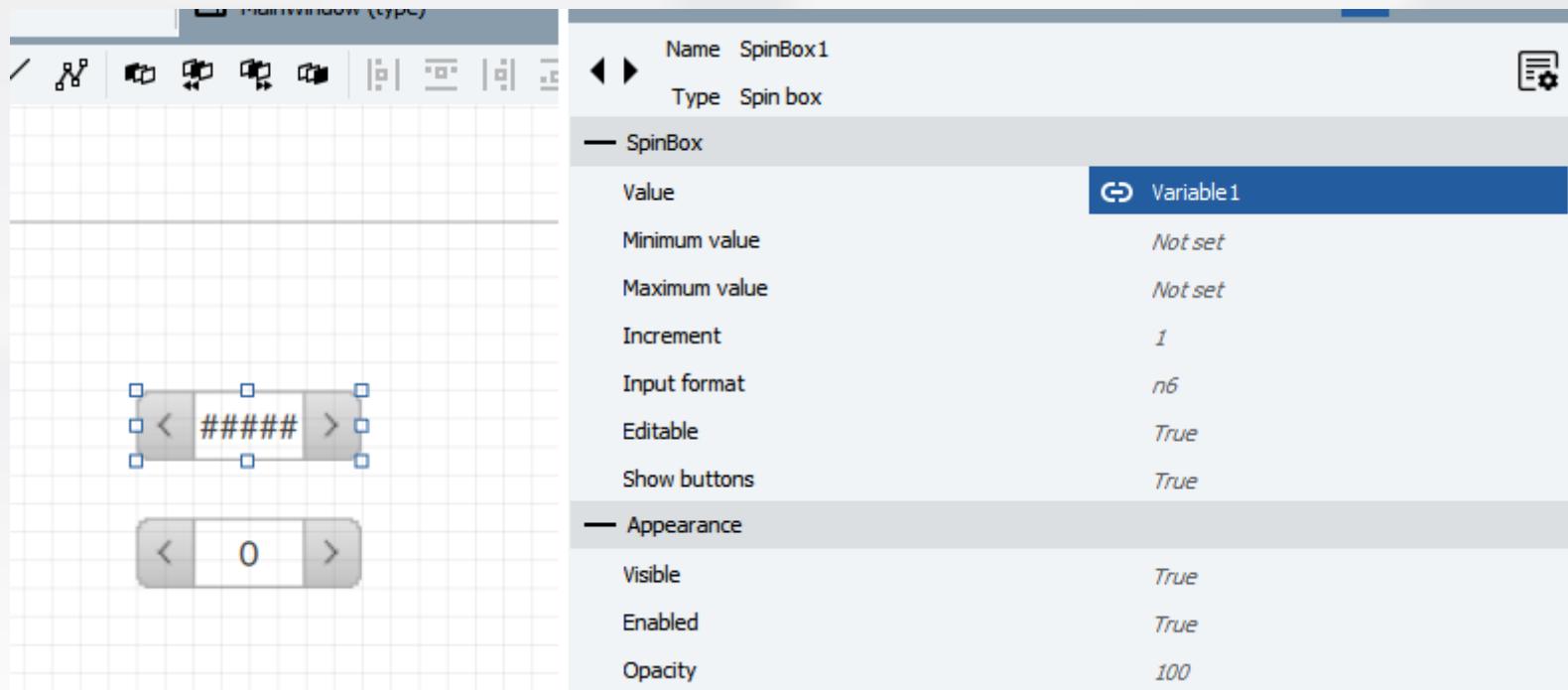
Dynamic links indicators

- Dynamic links can be
 - **Valid**: solid blue indicator, target variable is reachable
 - **Unknown**: blue exclamation mark, the target variable is not reachable at DesignTime (an Alias for example) but can be good at Runtime
 - **Unresolved**: the target variable does not exist, DynamicLink is most likely broken



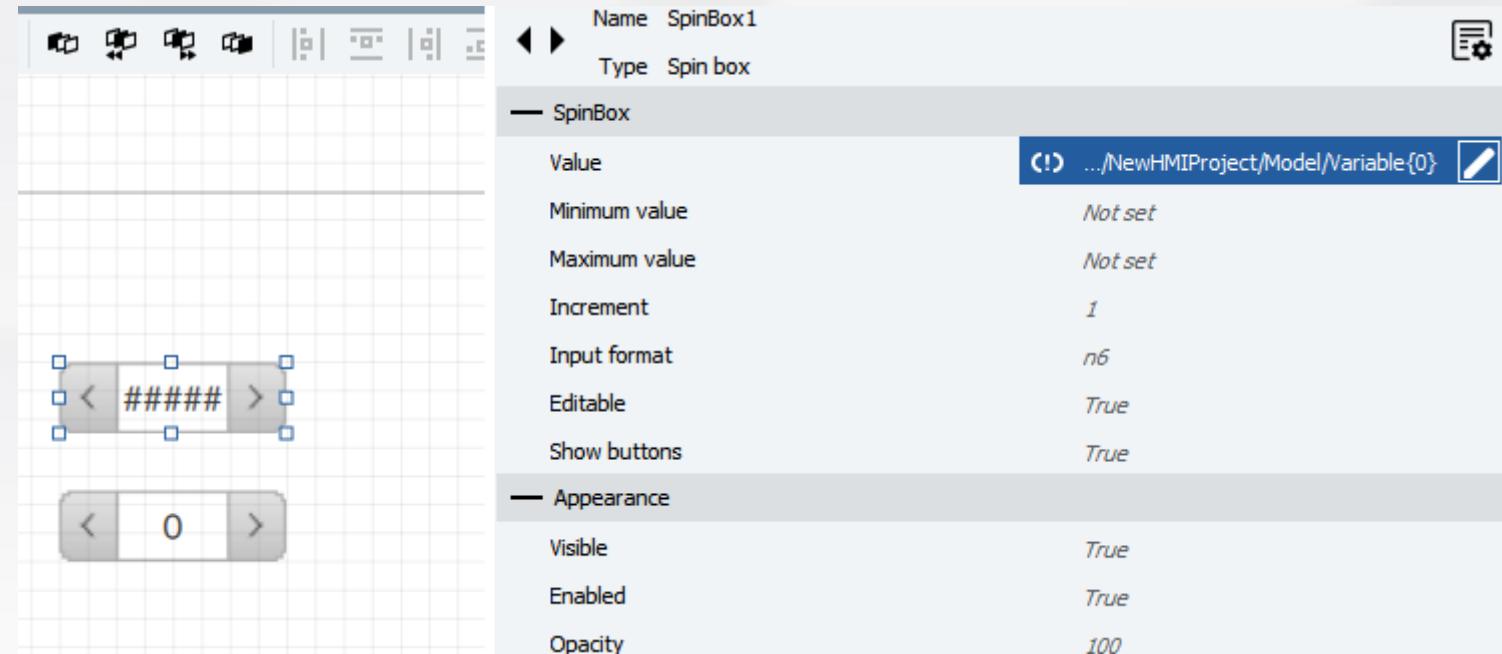
Advanced dynamic links: indexing

- Variables can be indexed by using {0} placeholder



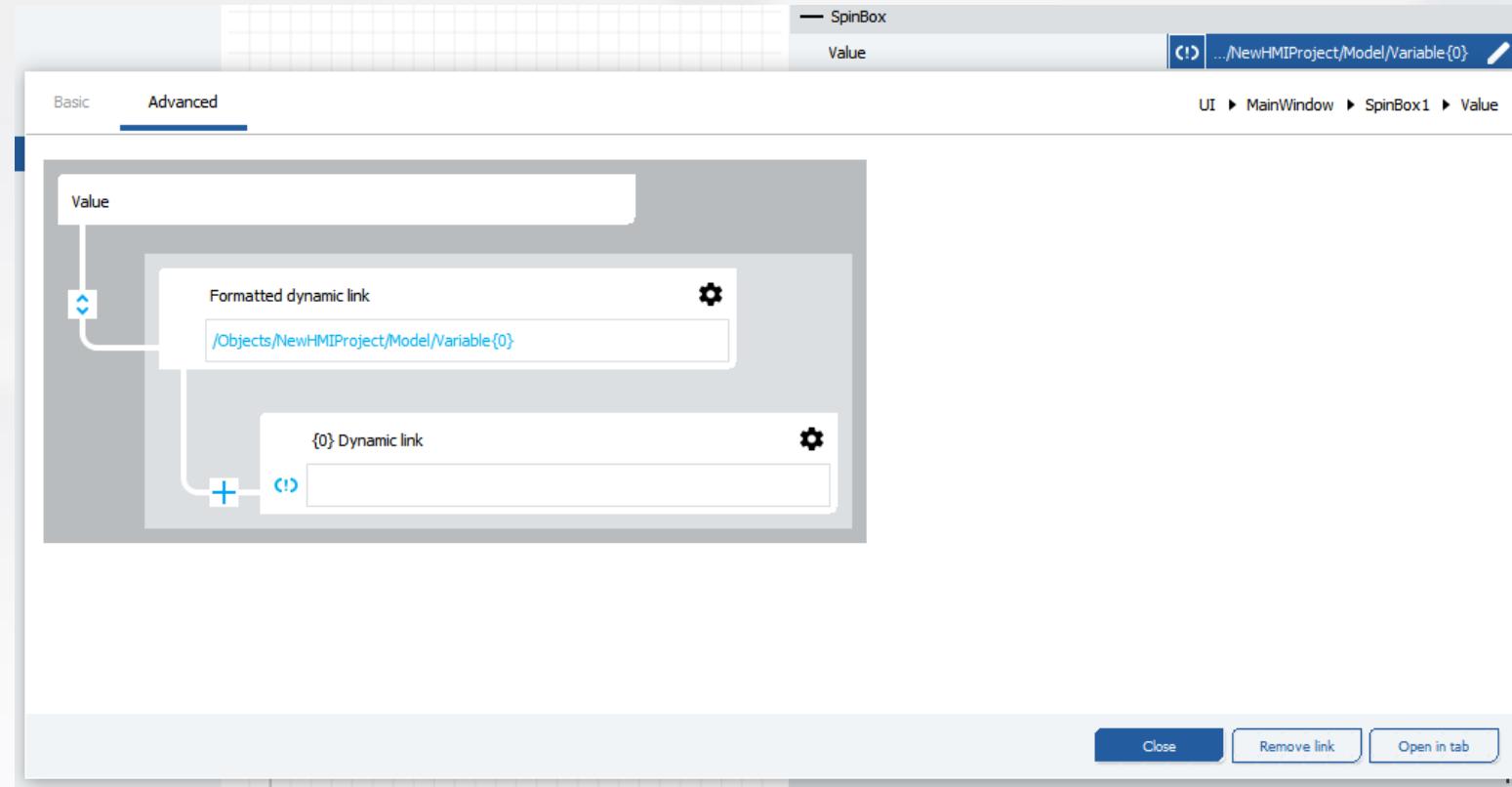
Advanced dynamic links: indexing

- Variables can be indexed by using {0} placeholder



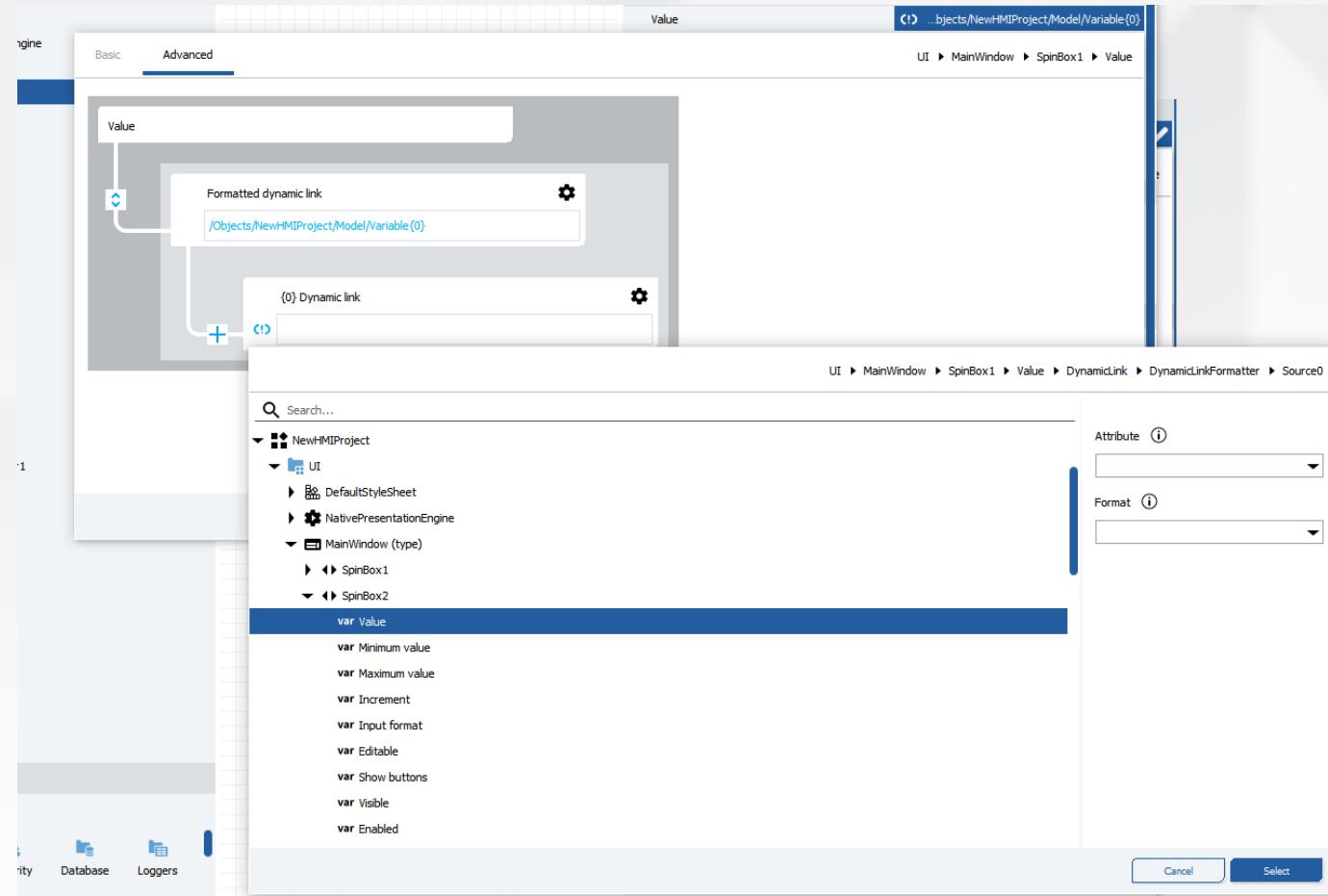
Advanced dynamic links: indexing

- Variables can be indexed by using {0} placeholder



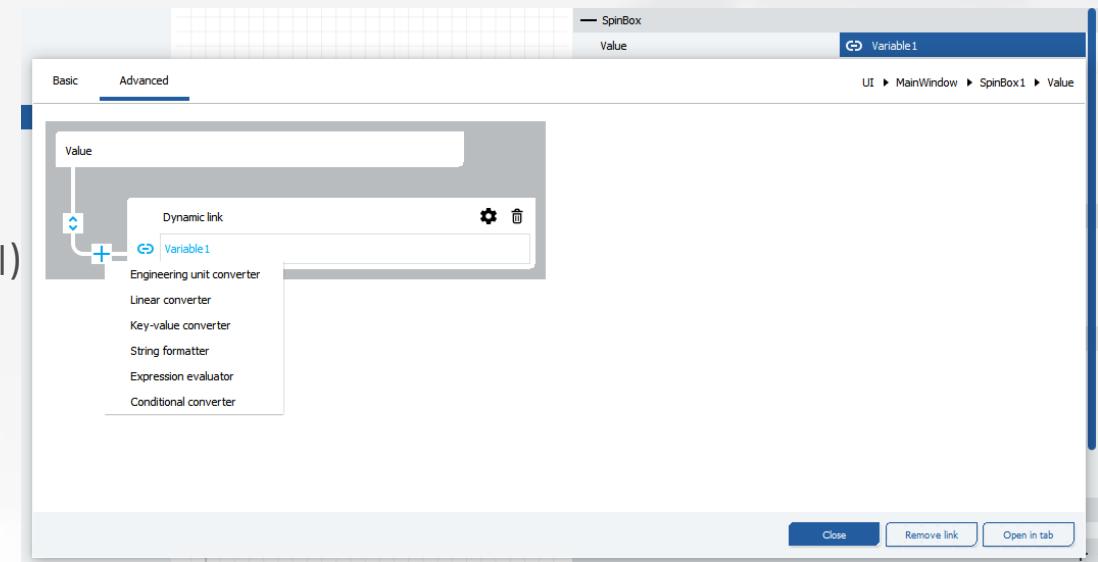
Advanced dynamic links: indexing

- Variables can be indexed by using {0} placeholder



Advanced dynamic links: converters

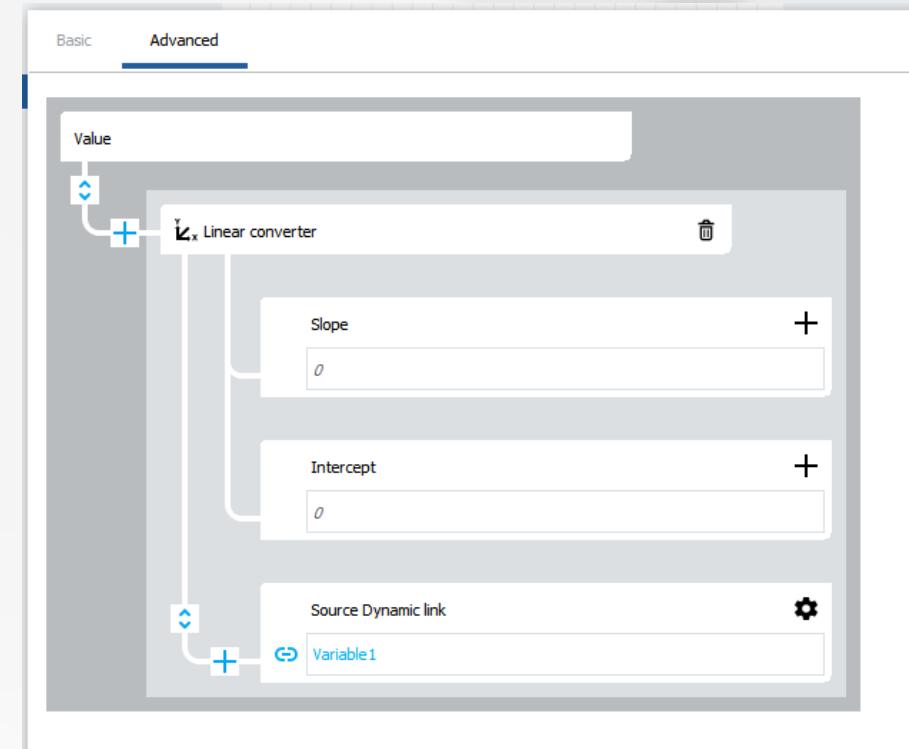
- By [Advanced...] button on Dynamic Link window, the value of a Property can be set through a Converter
 - **Engineering Unit converter**: it's a linear conversion defined by Raw range (PLC) and Scaled range (HMI)
 - **Linear converter**: it's a linear conversion defined by equation
$$y = mx + q$$
 - **Key-Value converter** * : converts the value of the source variable according to a table of key-value pairs.
 - **String formatter**: modify the format of a Numeric, Date/Time, String, SQL Query
 - **Expression evaluator**: performs arithmetic calculations with one or more input values
 - **Conditional converter**: writes two different values according to a boolean Condition DynamicLink



* Key-Value converter table can be exported to csv file and imported back, using a Template Library script

Advanced dynamic links: converters

- By [Advanced...] button on Dynamic Link window, the value of a Property can be set through a Converter
 - **Engineering Unit converter**: it's a linear conversion defined by Raw range (PLC) and Scaled range (HMI)
 - **Linear converter**: it's a linear conversion defined by equation
$$y = mx + q$$
 - **Key-Value converter** * : converts the value of the source variable according to a table of key-value pairs.
 - **String formatter**: modify the format of a Numeric, Date/Time, String, SQL Query
 - **Expression evaluator**: performs arithmetic calculations with one or more input values
 - **Conditional converter**: writes two different values according to a boolean Condition DynamicLink



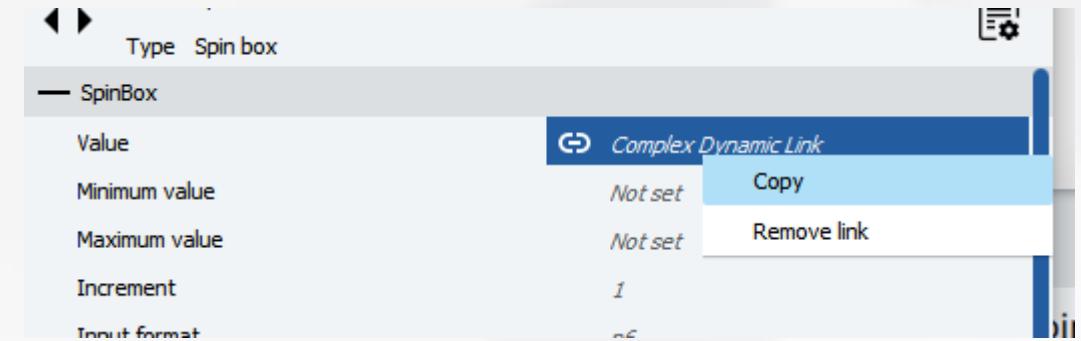
* Key-Value converter table can be exported to csv file and imported back, using a Template Library script

Advanced dynamic links: reusable converters

- Converters can be reused

- Using **copy/paste**

- Copy the converter from the property with "Complex Dynamic Link"
- Paste to another property



- Creating a **Converter as a Type**

- Create a Converter into Converters node
- The Converter will be automatically listed on the Advanced Dynamic Link configuration

Advanced dynamic links: reusable converters

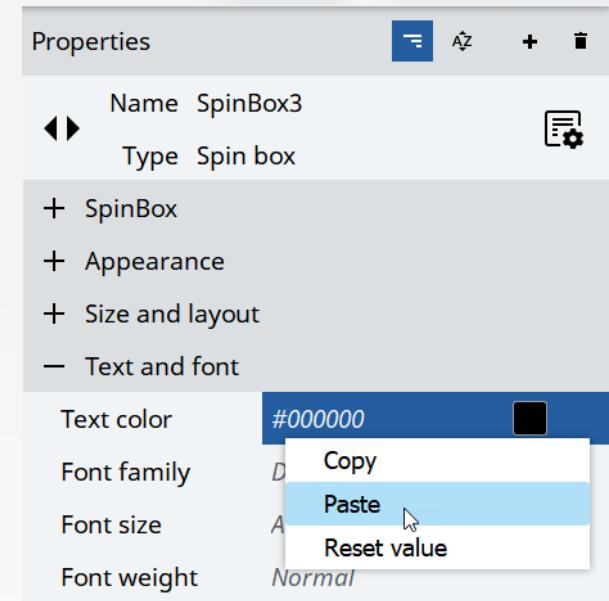
- Converters can be reused

- Using **copy/paste**

- Copy the converter from the property with "Complex Dynamic Link"
- Paste to another property

- Creating a **Converter as a Type**

- Create a Converter into Converters node
- The Converter will be automatically listed on the Advanced Dynamic Link configuration



Advanced dynamic links: reusable converters

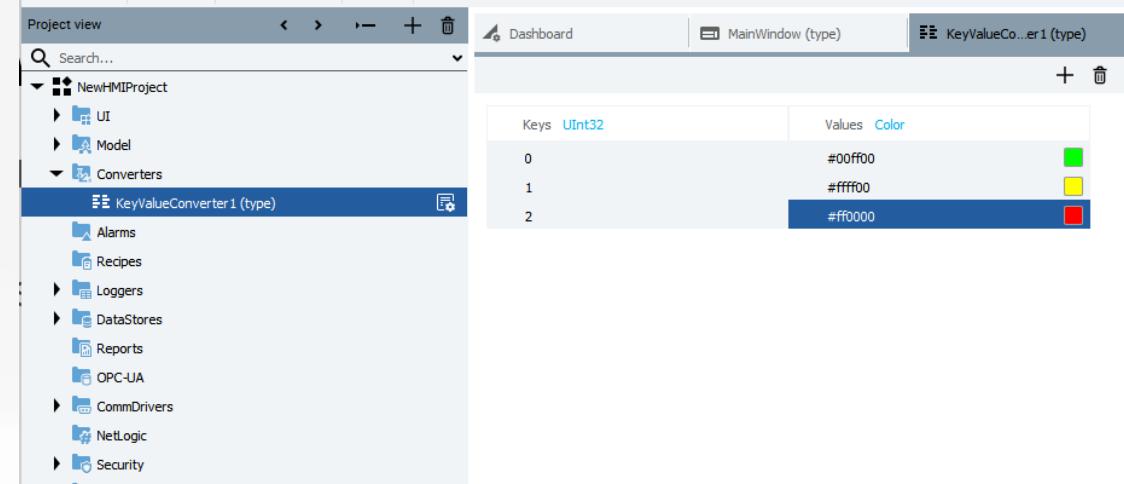
- Converters can be reused

- Using **copy/paste**

- Copy the converter from the property with "*Complex Dynamic Link*"
- Paste to another property

- Creating a **Converter as a Type**

- Create a Converter into Converters node
- The Converter will be automatically listed on the Advanced Dynamic Link configuration



Advanced dynamic links: reusable converters

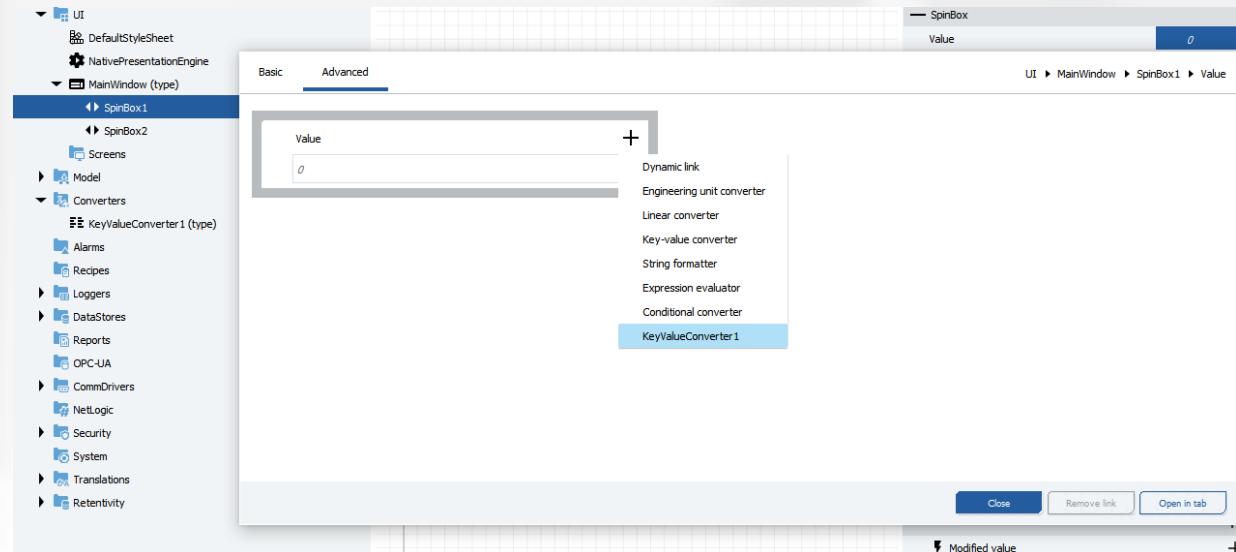
- Converters can be reused

- Using **copy/paste**

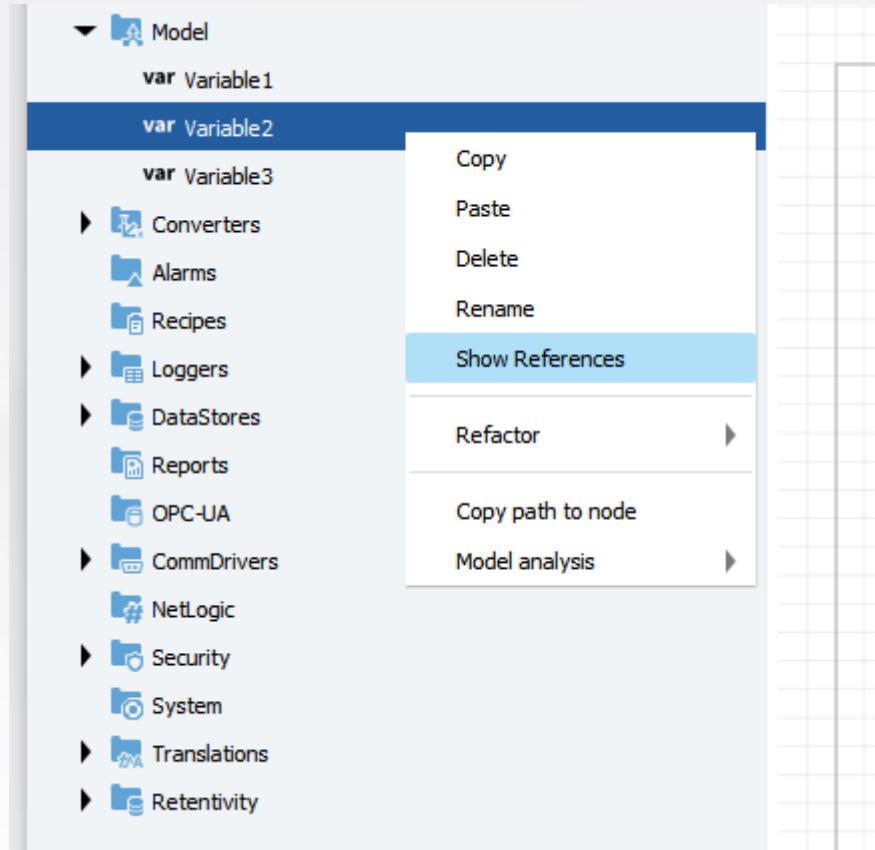
- Copy the converter from the property with "Complex Dynamic Link"
- Paste to another property

- Creating a **Converter as a Type**

- Create a Converter into Converters node
- The Converter will be automatically listed on the Advanced Dynamic Link configuration



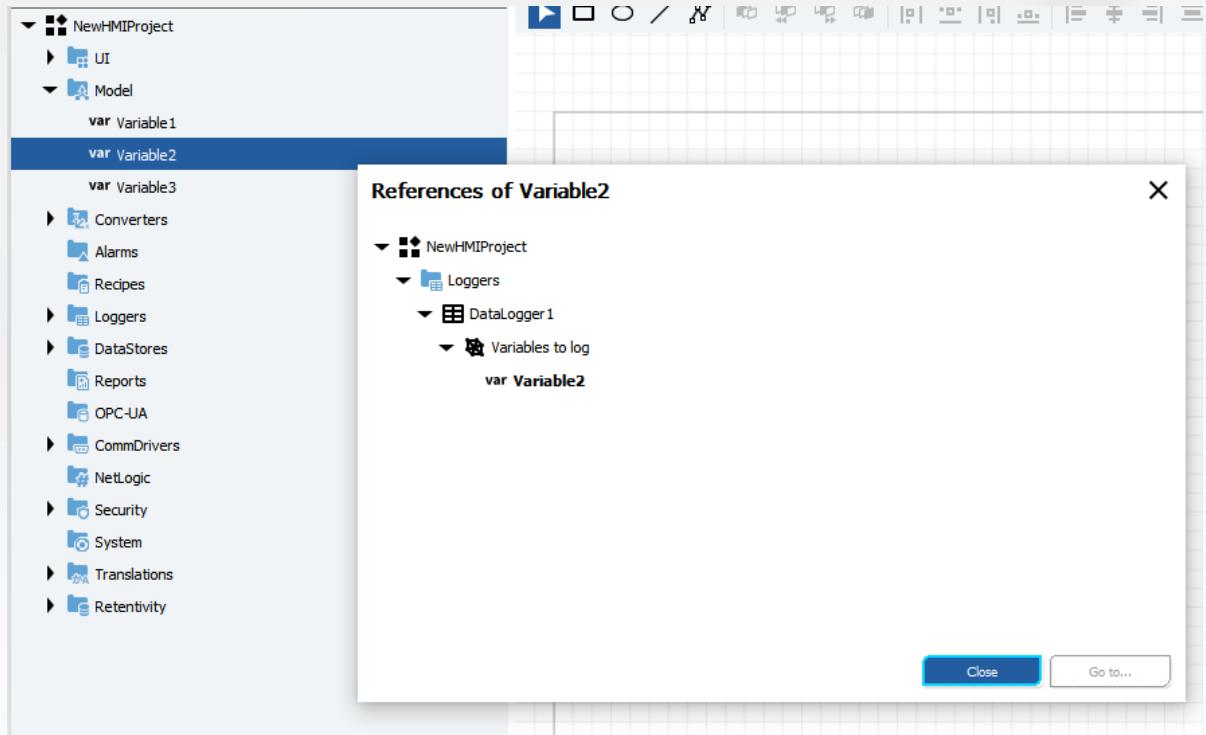
Show reference



- Answer the question:
"who refers to this object ?"

- Can be used mainly for Tags, but also, for other objects like
 - Datastore,
 - Scripts,
 - ...

Show reference



- Answer the question:
"who refers to this object ?"

- Can be used mainly for Tags, but also, for other objects like
 - Datastore,
 - Scripts,
 - ...

Hands-on session

- Define an "indexed dynamic link" to show the value of Variable1, Variable2, Variable3 using a switch object
- Use a Key-Value converter to change the background color of a rectangle based on the value of a variable
- Use a Key-Value converter to change the string of a label based on the value of a variable
- Use a Linear convert to show a scaled value



Animations and events



Commands, methods and events

- Commands/Methods can be executed on Events triggered by objects



- Event Types:

- **Native Event** (Event View): like "MouseClick", "MouseDown", "MouseUP" of a button

Events

- MouseClick event
- MouseDown event
- MouseUp event

- **Changed Event, Transitioned Event, Range Transitioned Event** (Property Panel): custom event

- Method Types:

- Global Methods or **Commands**: like "Change User", or "Open Dialog"...
 - Local **Methods exposed by objects**: like "Log" of Datalogger, or "Refresh" of Datalogger viewer...

Commands, methods and events

- Commands/Methods can be executed on Events triggered by objects

- Event Types:

- Native Event** (Event View): like "MouseClick", "MouseDown", "MouseUP" of a button

- Changed Event, Transitioned Event, Range Transitioned Event** (Property Panel): custom event

- Method Types:

- Global Methods or **Commands**: like "Change User", or "Open Dialog"...

- Local **Methods exposed by objects**: like "Log" of Datalogger, or "Refresh" of Datalogger viewer...



A screenshot of the 'Properties' panel for a variable named 'Variable1'. The variable is of type 'Int32' with a value of 0. A context menu is open over the variable name, with 'New' selected. The menu also includes 'Delete', 'Copy', and 'Show References'. Below the variable list, there are three items: 'Changed event', 'Transitioned event', and 'Range transitioned event', each preceded by a lightning bolt icon.

Commands, methods and events

- Commands/Methods can be executed on Events triggered by objects



- Event Types:

- Native Event** (Event View): like "MouseClick", "MouseDown", "MouseUP" of a button

- Changed Event, Transitioned Event, Range Transitioned Event** (Property Panel): custom event

- Method Types:

- Global Methods or **Commands**: like "Change User", or "Open Dialog"...

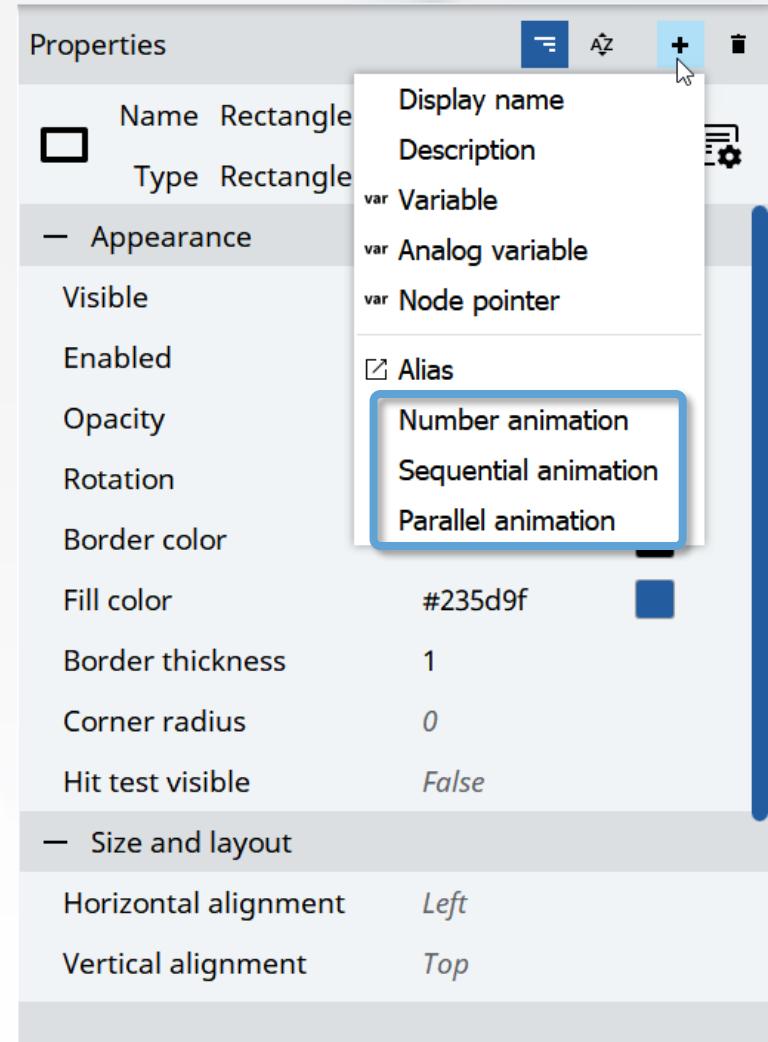
- Local **Methods exposed by objects**: like "Log" of Datalogger, or "Refresh" of Datalogger viewer...

The Properties panel shows the configuration for 'TextBox1'. The 'Visible' property is set to 'True'. In the dropdown menu for 'Event', 'Changed event' is selected. Other options in the dropdown include 'New', 'Delete', and 'Transitioned event'. The 'Border color' is set to '#000000'. Under 'Size and layout', the 'Horizontal alignment' is 'Left' and the 'Vertical alignment' is 'Top'. The 'Width' is set to 110 and the 'Height' is set to Auto.

Property	Value
Name	TextBox1
Type	Text box
Visible	True
Event	Changed event
Border color	#000000
Horizontal alignment	Left
Vertical alignment	Top
Width	110
Height	Auto

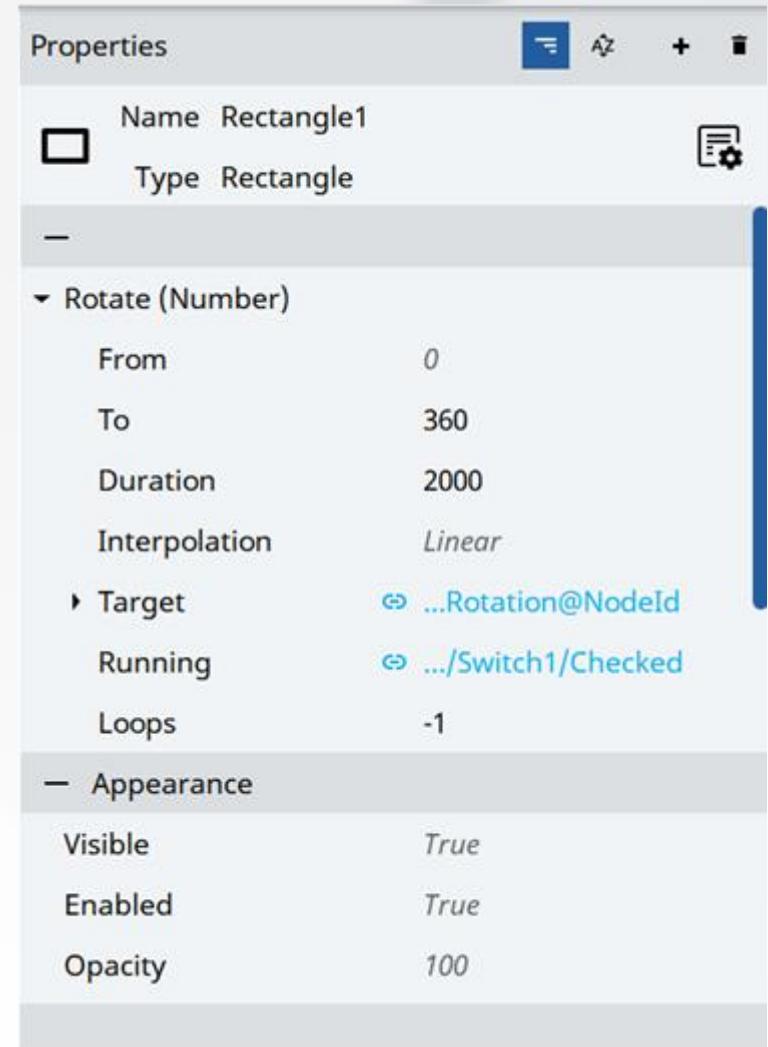
Animations

- Using Dynamic Links some basic animations can be defined as **just linking a Variable to a property**
 - Visibility, Enabling, Resizing, Change color...
- For "**eye-catching**" Animations:
 - Number animation: single animation
 - Sequential animation: container of animations executed in sequence
 - Parallel animation: container of animations executed in parallel(at the same time)
 - Animation of Behavior: allow to animate a property linked to a Variable



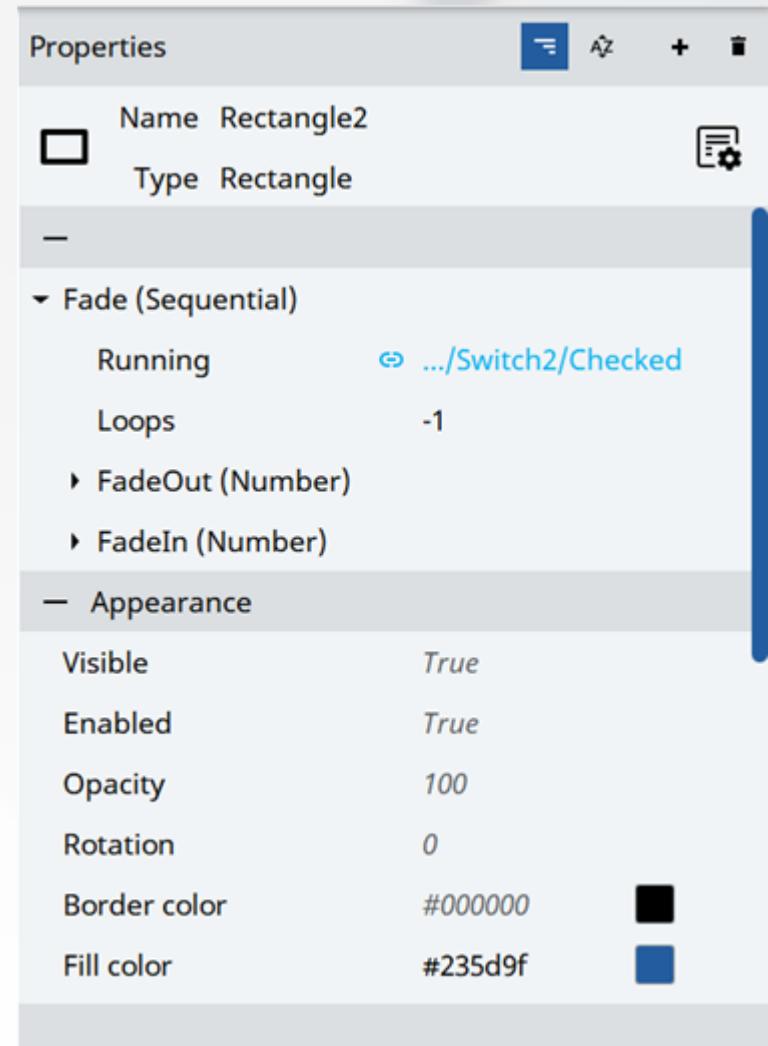
Animations

- Using Dynamic Links some basic animations can be defined as **just linking a Variable to a property**
 - Visibility, Enabling, Resizing, Change color...
- For "**eye-catching**" Animations:
 - Number animation: single animation
 - Sequential animation: container of animations executed in sequence
 - Parallel animation: container of animations executed in parallel(at the same time)
 - Animation of Behavior: allow to animate a property linked to a Variable



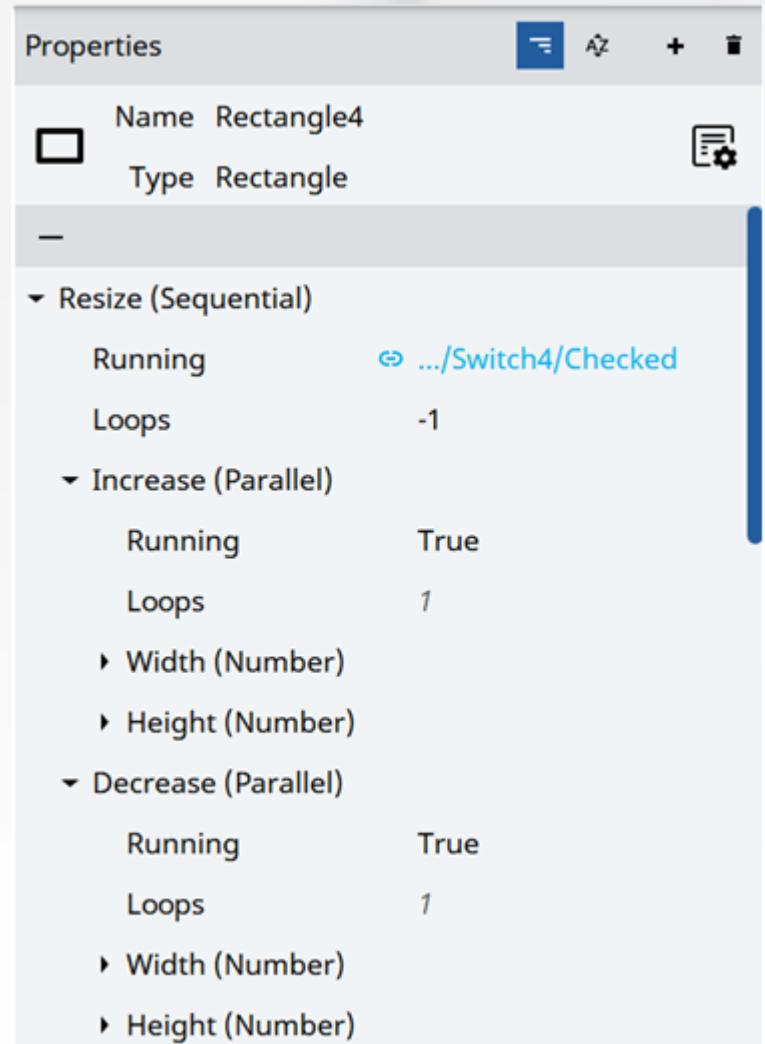
Animations

- Using Dynamic Links some basic animations can be defined as **just linking a Variable to a property**
 - Visibility, Enabling, Resizing, Change color...
- For "**eye-catching**" Animations:
 - Number animation: single animation
 - Sequential animation: container of animations executed in sequence
 - Parallel animation: container of animations executed in parallel(at the same time)
 - Animation of Behavior: allow to animate a property linked to a Variable



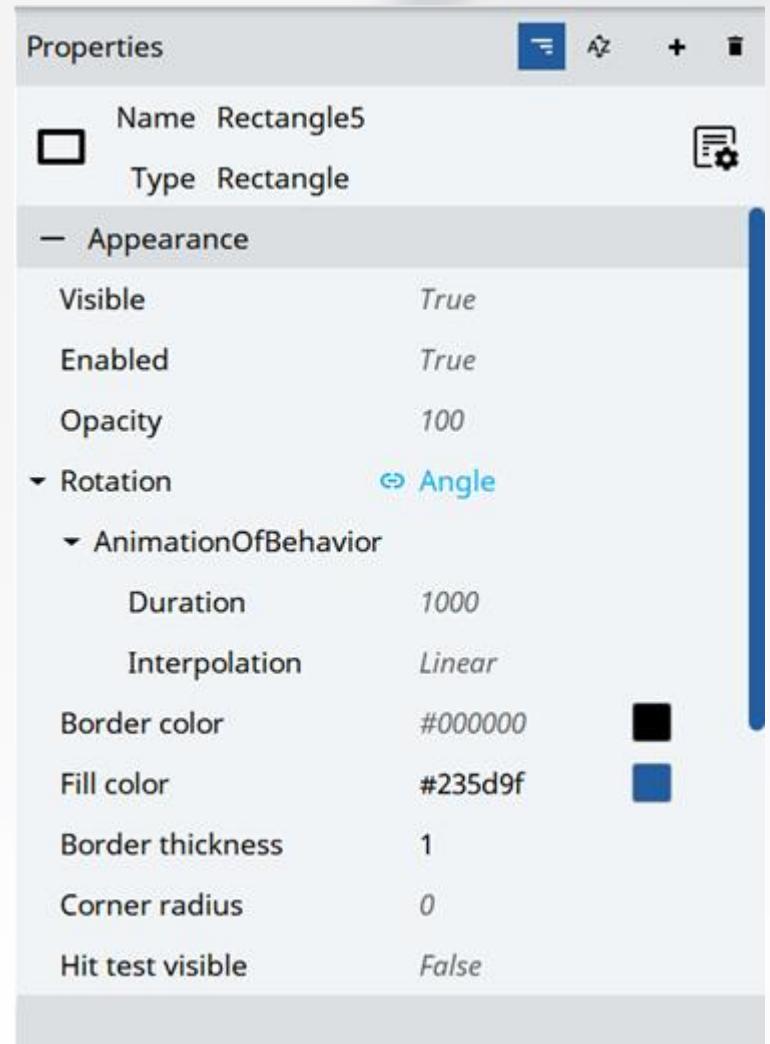
Animations

- Using Dynamic Links some basic animations can be defined as **just linking a Variable to a property**
 - Visibility, Enabling, Resizing, Change color...
- For "**eye-catching**" Animations:
 - Number animation: single animation
 - Sequential animation: container of animations executed in sequence
 - Parallel animation: container of animations executed in parallel(at the same time)
 - Animation of Behavior: allow to animate a property linked to a Variable



Animations

- Using Dynamic Links some basic animations can be defined as **just linking a Variable to a property**
 - Visibility, Enabling, Resizing, Change color...
- For "**eye-catching**" Animations:
 - Number animation: single animation
 - Sequential animation: container of animations executed in sequence
 - Parallel animation: container of animations executed in parallel (at the same time)
 - Animation of Behavior: allow to animate a property linked to a Variable



Hands-on session

- Add a Panel Loader object and add one or more buttons that call the method "Change Panel" to change the panel loaded
- Create a button object that calls the method "Open Dialog" to open a Dialog Box
- Define a Transitioned Event that executes a method like "Variable commands > Set Variable Value" or "Core commands > Close"



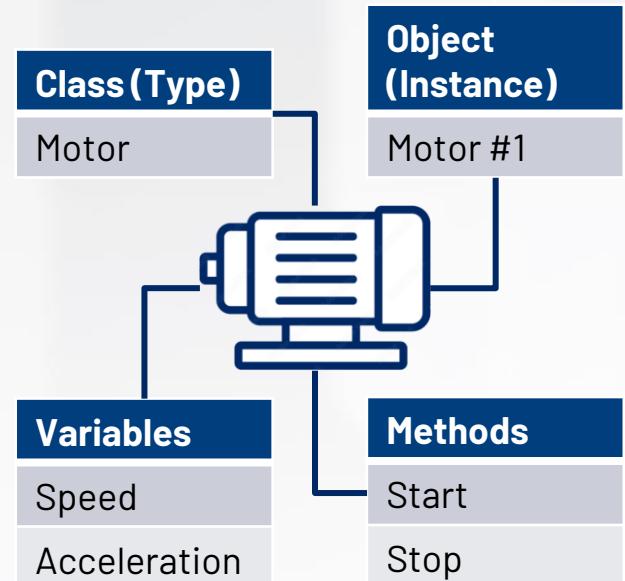
Work with reusable objects

Global objects, UDTs, Widgets



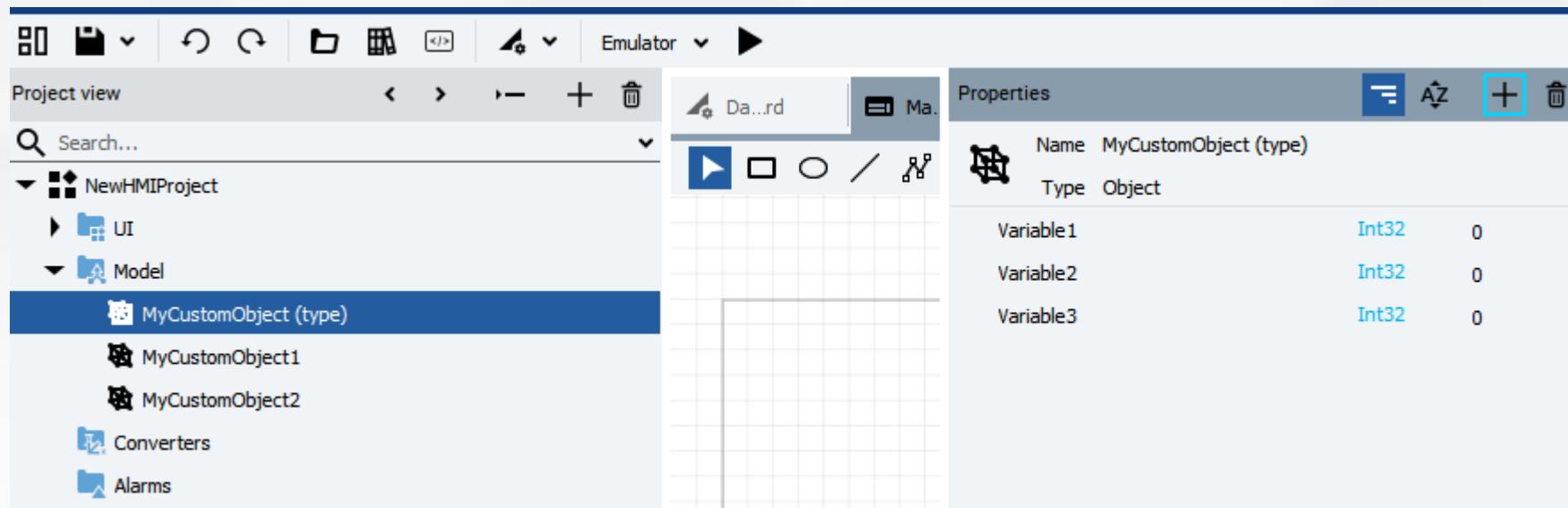
Object-oriented programming (OOP)

- Elements of OOP:
 - **Class:** as a core element, it's a collection of methods and variables and generally, it is regarded as a **type for instance**.
 - **Object:** as a basic element, it is regarded as an **instance** of a class
- Capabilities of OOP:
 - Encapsulation, Abstraction, Inheritance, Polymorphism
- Object-Oriented Programming in Optix means
 - **Native Types** already seen →
 - **Custom Types**



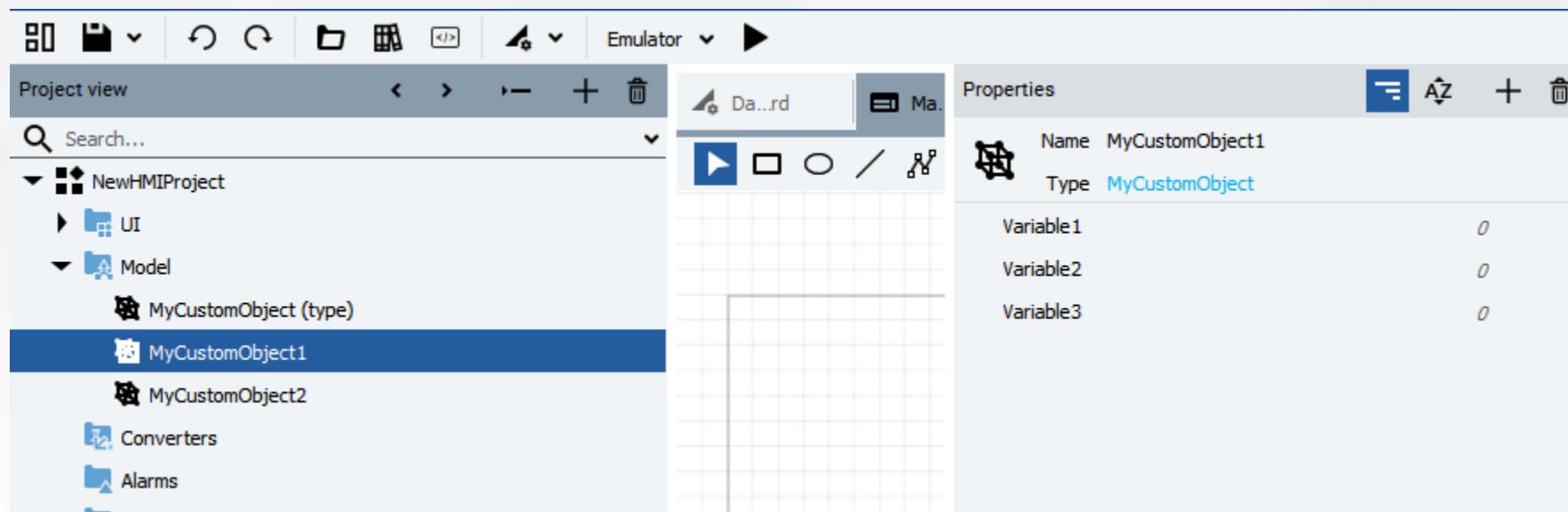
Custom types: reusable tags

- Object-Oriented Programming paradigm can be applied to Model Tags
 - "MyMotor(type)" = Class = Type
 - "MyMotor1", "MyMotor2" = Instances



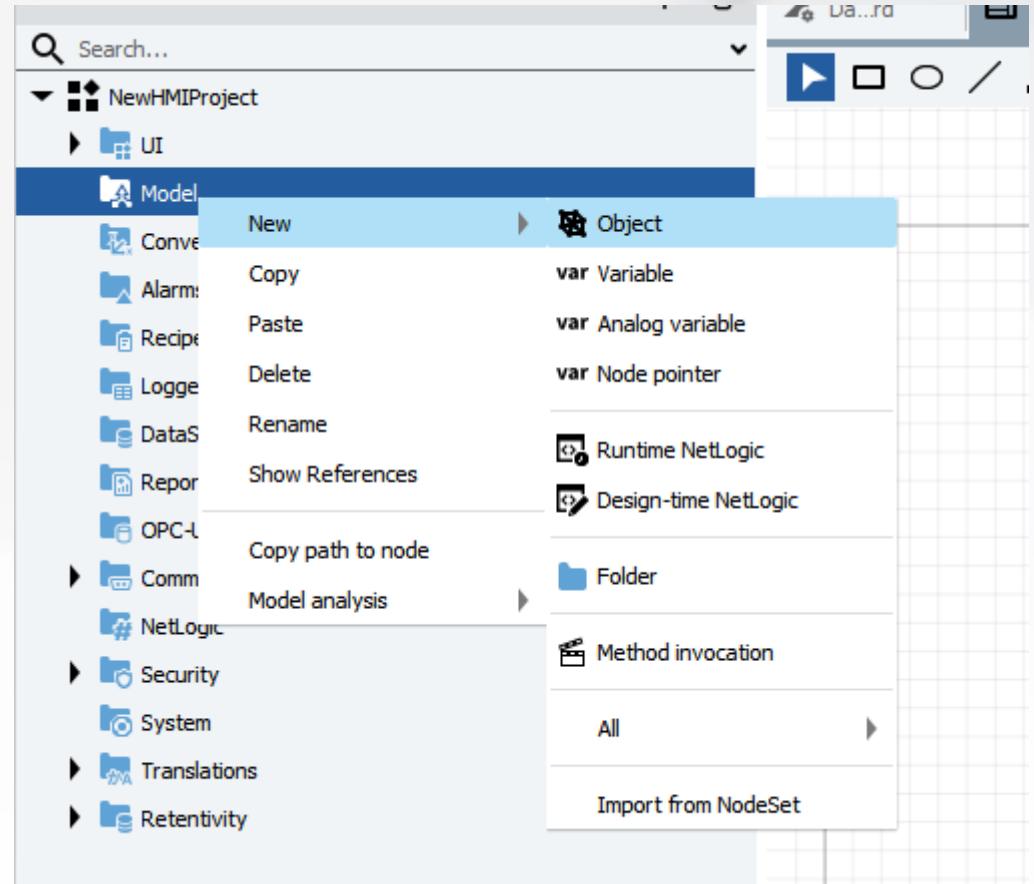
Custom types: reusable tags

- Object-Oriented Programming paradigm can be applied to Model Tags
 - "MyMotor(type)" = Class = Type
 - "MyMotor1", "MyMotor2" = Instances



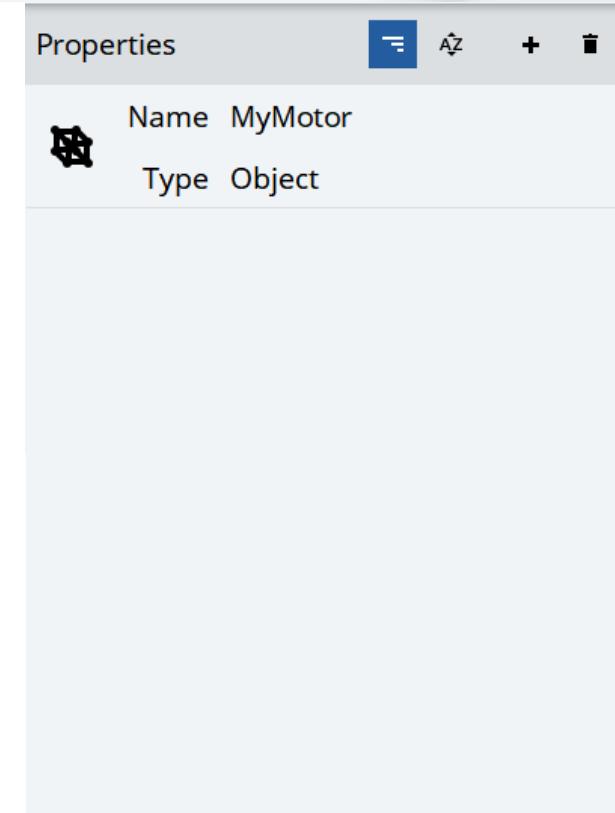
Custom types: reusable tags example

- Example: a Structured Tag into Model folder
- Created as a brand **new object** that becomes a **new type**
- How to make your object a Type:
 - Option 1: Drag&Drop into Types View panel
 - Option 2: in "Advanced Mode",
Right click > Refactor > Transform instance to type



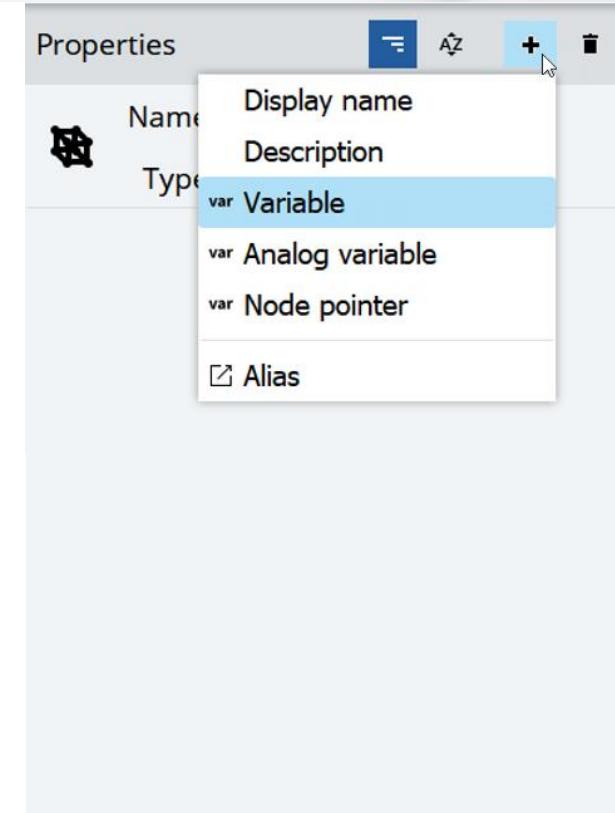
Custom types: reusable tags example

- Example: a Structured Tag into Model folder
- Created as a brand **new object** that becomes a **new type**
- How to make your object a Type:
 - Option 1: Drag&Drop into Types View panel
 - Option 2: in "Advanced Mode", Right click > Refactor > Transform instance to type



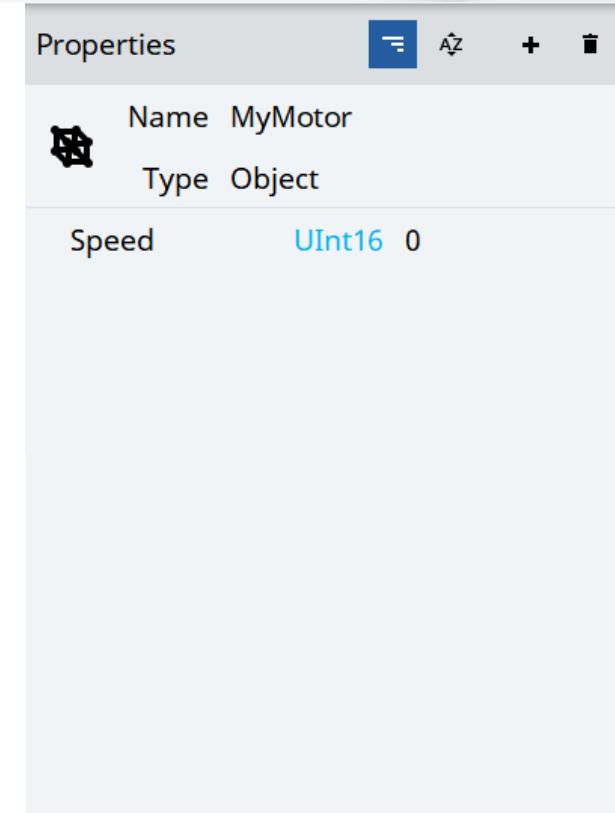
Custom types: reusable tags example

- Example: a Structured Tag into Model folder
- Created as a brand **new object** that becomes a **new type**
- How to make your object a Type:
 - Option 1: Drag&Drop into Types View panel
 - Option 2: in "Advanced Mode", Right click > Refactor > Transform instance to type



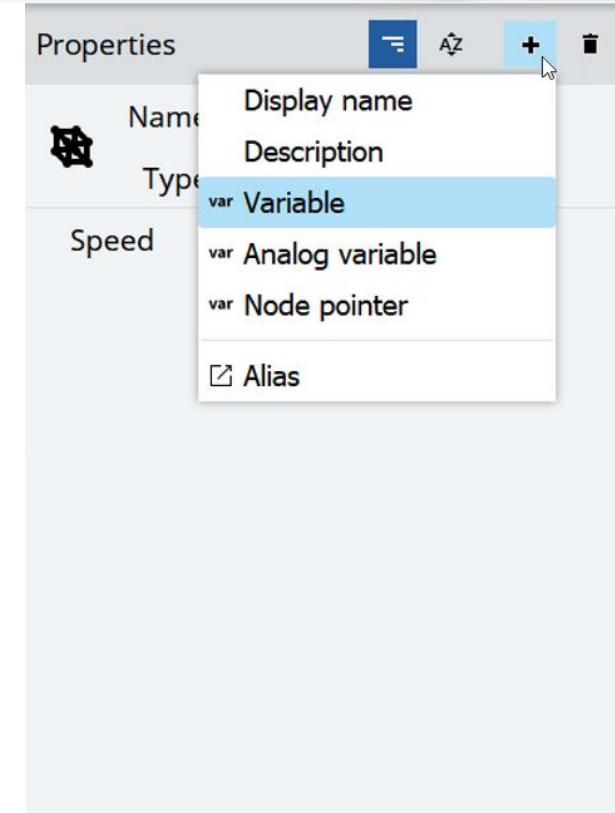
Custom types: reusable tags example

- Example: a Structured Tag into Model folder
- Created as a brand **new object** that becomes a **new type**
- How to make your object a Type:
 - Option 1: Drag&Drop into Types View panel
 - Option 2: in "Advanced Mode", Right click > Refactor > Transform instance to type



Custom types: reusable tags example

- Example: a Structured Tag into Model folder
- Created as a brand **new object** that becomes a **new type**
- How to make your object a Type:
 - Option 1: Drag&Drop into Types View panel
 - Option 2: in "Advanced Mode", Right click > Refactor > Transform instance to type



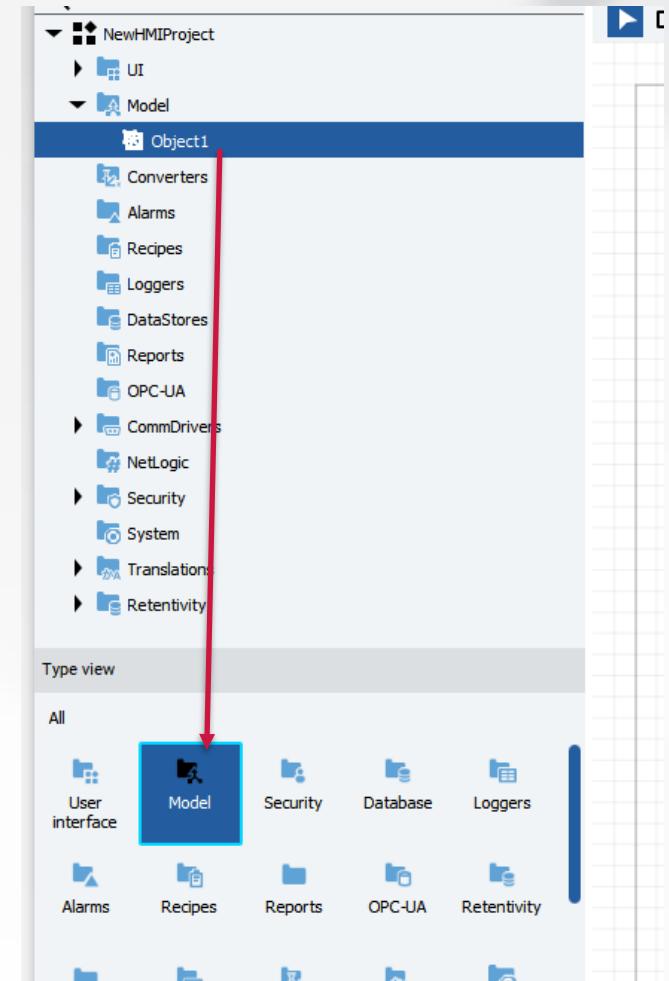
Custom types: reusable tags example

- Example: a Structured Tag into Model folder
- Created as a brand **new object** that becomes a **new type**
- How to make your object a Type:
 - Option 1: Drag&Drop into Types View panel
 - Option 2: in "Advanced Mode", Right click > Refactor > Transform instance to type

Properties	
Name	MyMotor
Type	Object
Speed	UInt16 0
Acceleration	UInt16 0

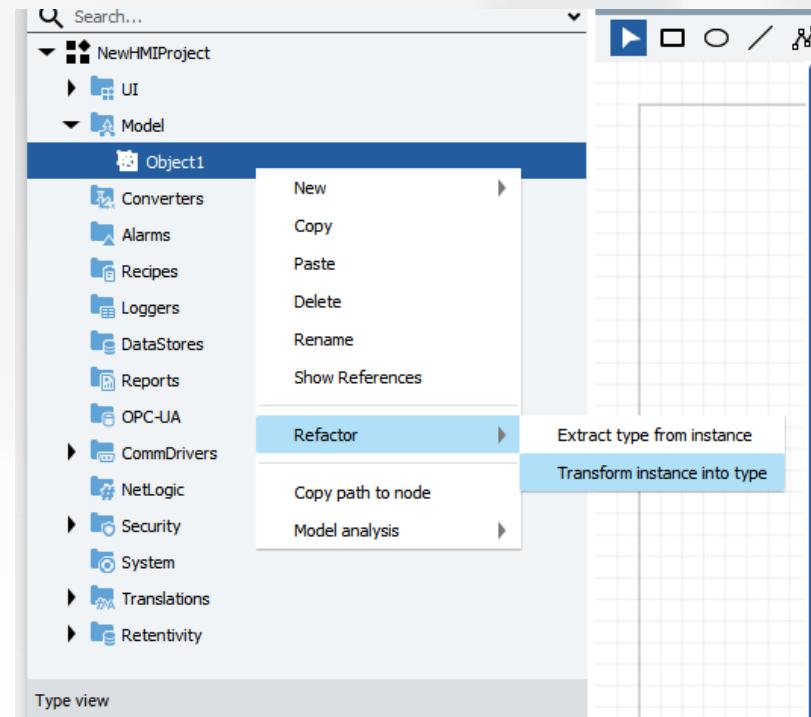
Custom types: reusable tags example

- Example: a Structured Tag into Model folder
- Created as a brand **new object** that becomes a **new type**
- How to make your object a Type:
 - Option 1: Drag&Drop into Types View panel
 - Option 2: in "Advanced Mode", Right click > Refactor > Transform instance to type



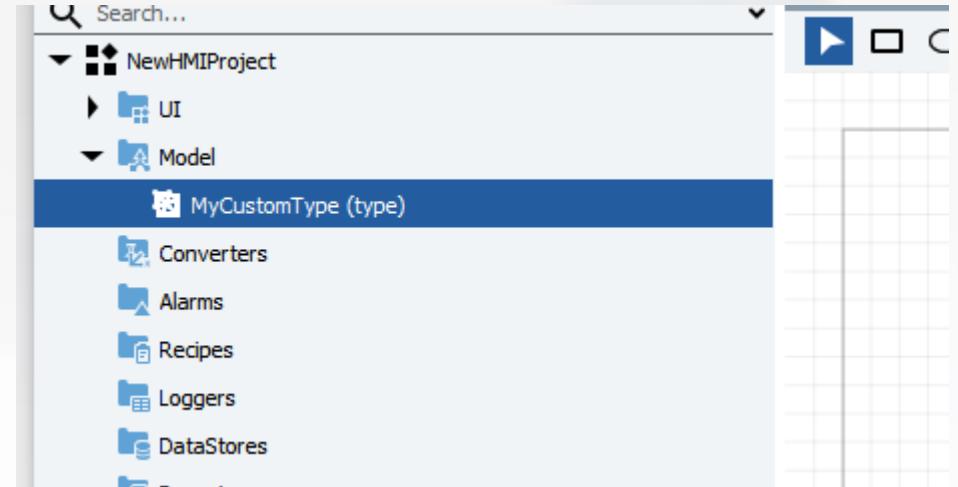
Custom types: reusable tags example

- Example: a Structured Tag into Model folder
- Created as a brand **new object** that becomes a **new type**
- How to make your object a Type:
 - Option 1: Drag&Drop into Types View panel
 - Option 2: in "Advanced Mode", Right click > Refactor > Transform instance to type



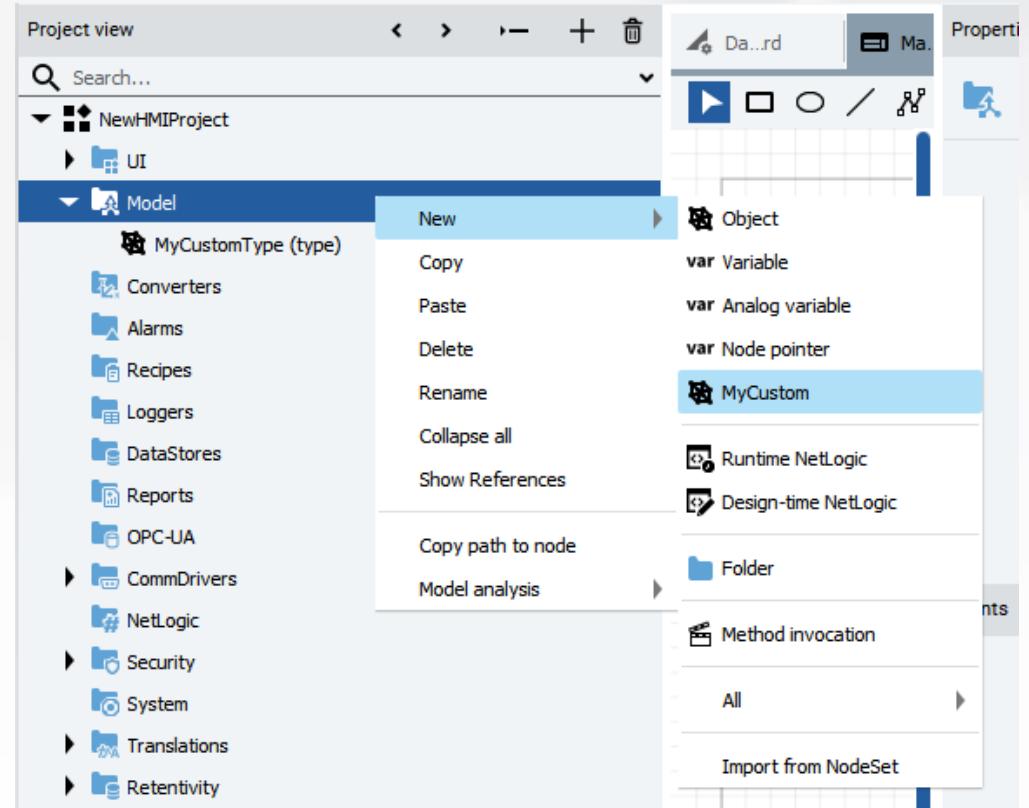
Custom types: reusable tags example

- Example: a Structured Tag into Model folder
- Created as a brand **new object** that becomes a **new type**
- How to make your object a Type:
 - Option 1: Drag&Drop into Types View panel
 - Option 2: in "Advanced Mode", Right click > Refactor > Transform instance to type



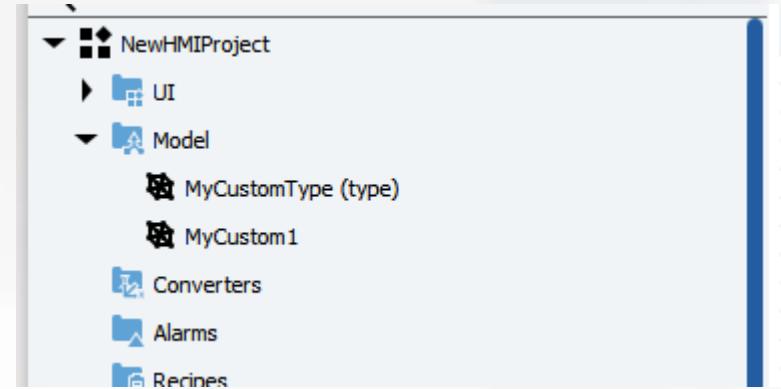
Custom types: reusable tags example

- Example: a Structured Tag into Model folder
- Created as a brand **new object** that becomes a **new type**
- How to make your object a Type:
 - Option 1: Drag&Drop into Types View panel
 - Option 2: in "Advanced Mode", Right click > Refactor > Transform instance to type



Custom types: reusable tags example

- Example: a Structured Tag into Model folder
- Created as a brand **new object** that becomes a **new type**
- How to make your object a Type:
 - Option 1: Drag&Drop into Types View panel
 - Option 2: in "Advanced Mode", Right click > Refactor > Transform instance to type



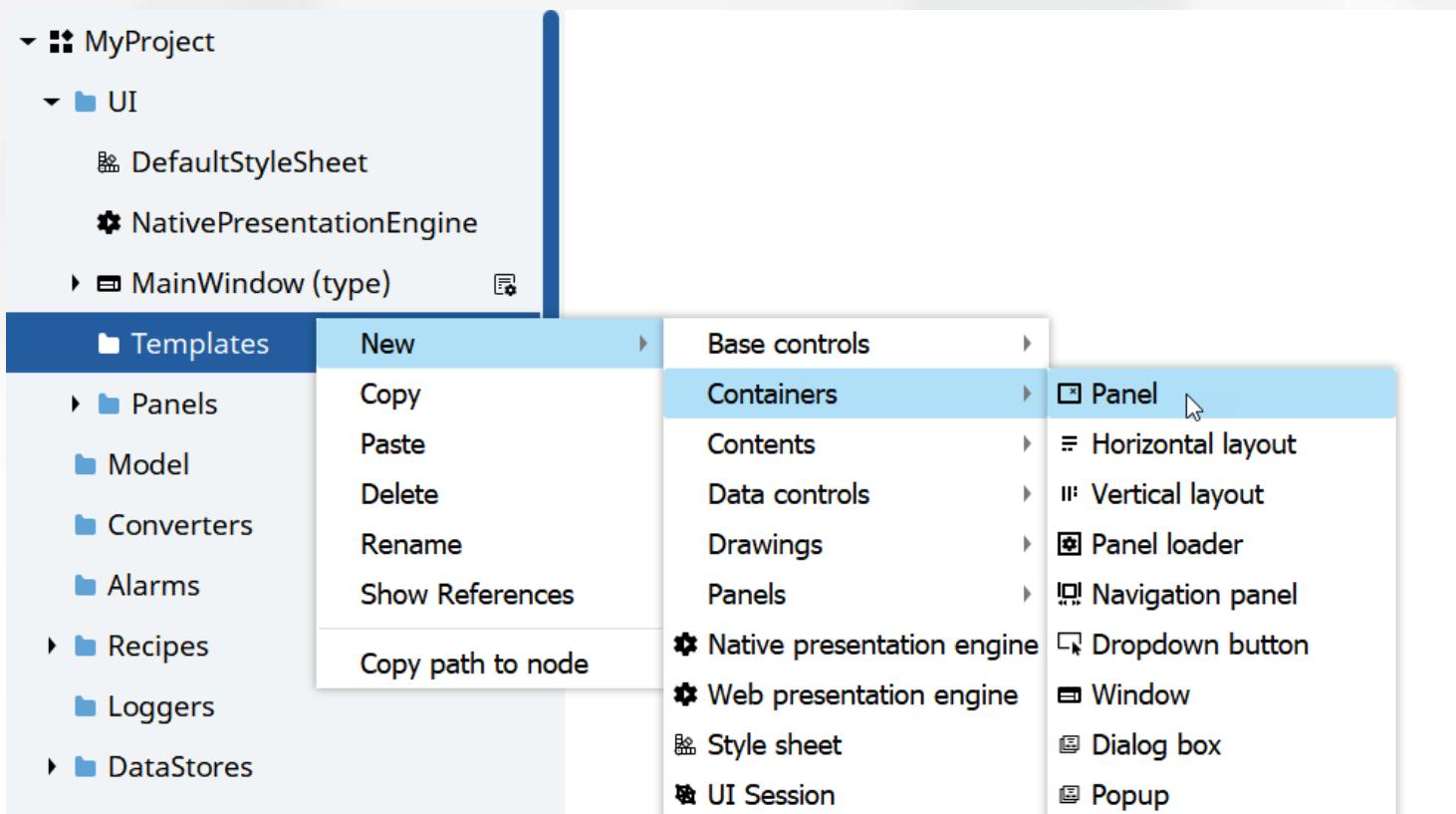
Hands-on session

- Create a Motor Type variable into the Model folder with these elements
 - Power (Float)
 - Speed (Int16)
- Create 3 instances of the Motor Type
 - Motor1, Motor2, Motor3...



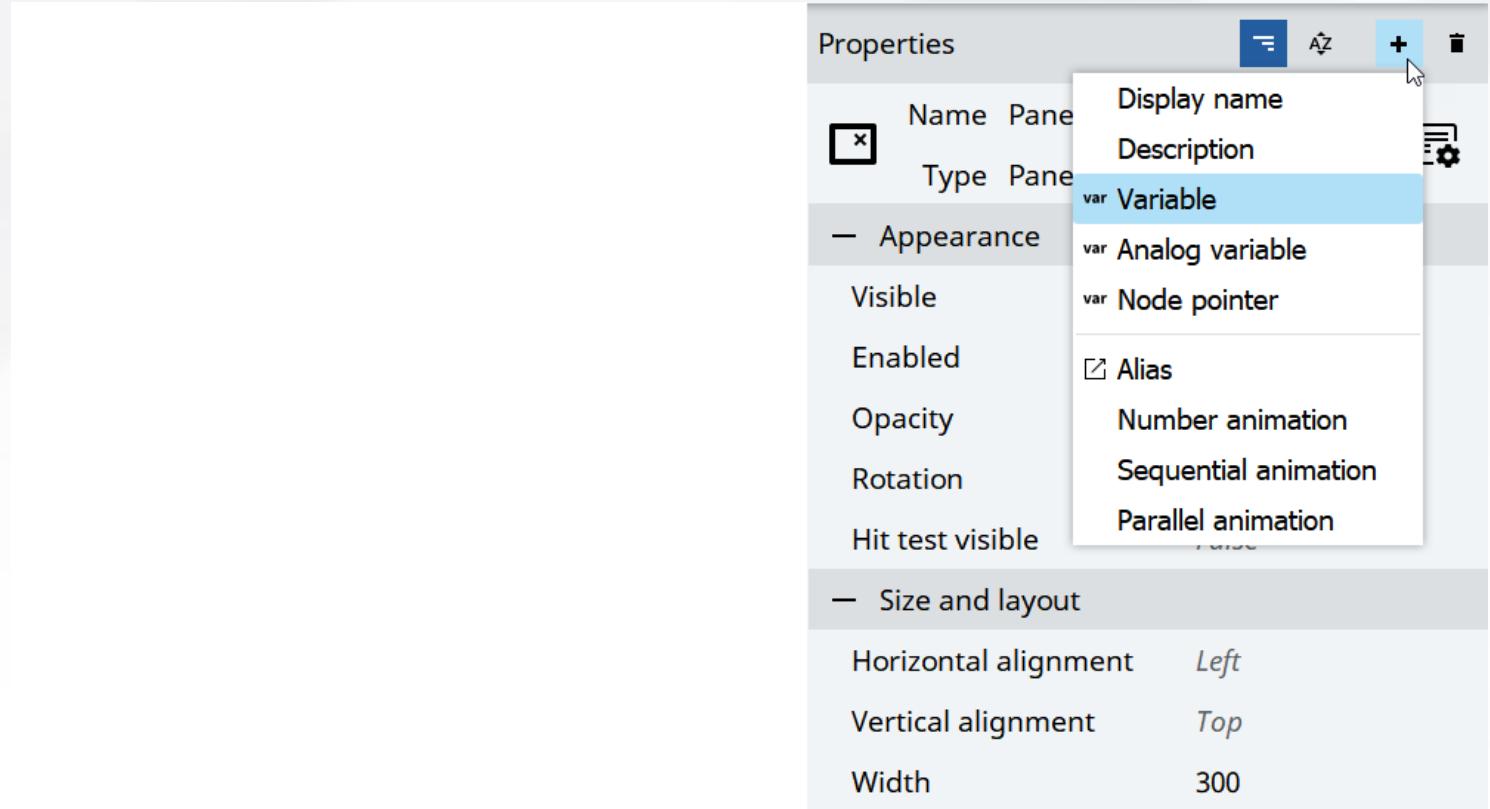
Custom types: reusable graphics with a "local variable"

- Object-Oriented Programming paradigm can be also applied to Graphics
 - "Panel_with_BG(type)" = Class = Type
 - "Panel1" = Instance
- Reusable Graphics can be
 - Widgets
 - Dialogs
 - Panels
- Example: a customized Panel with a rectangle used to set the background color



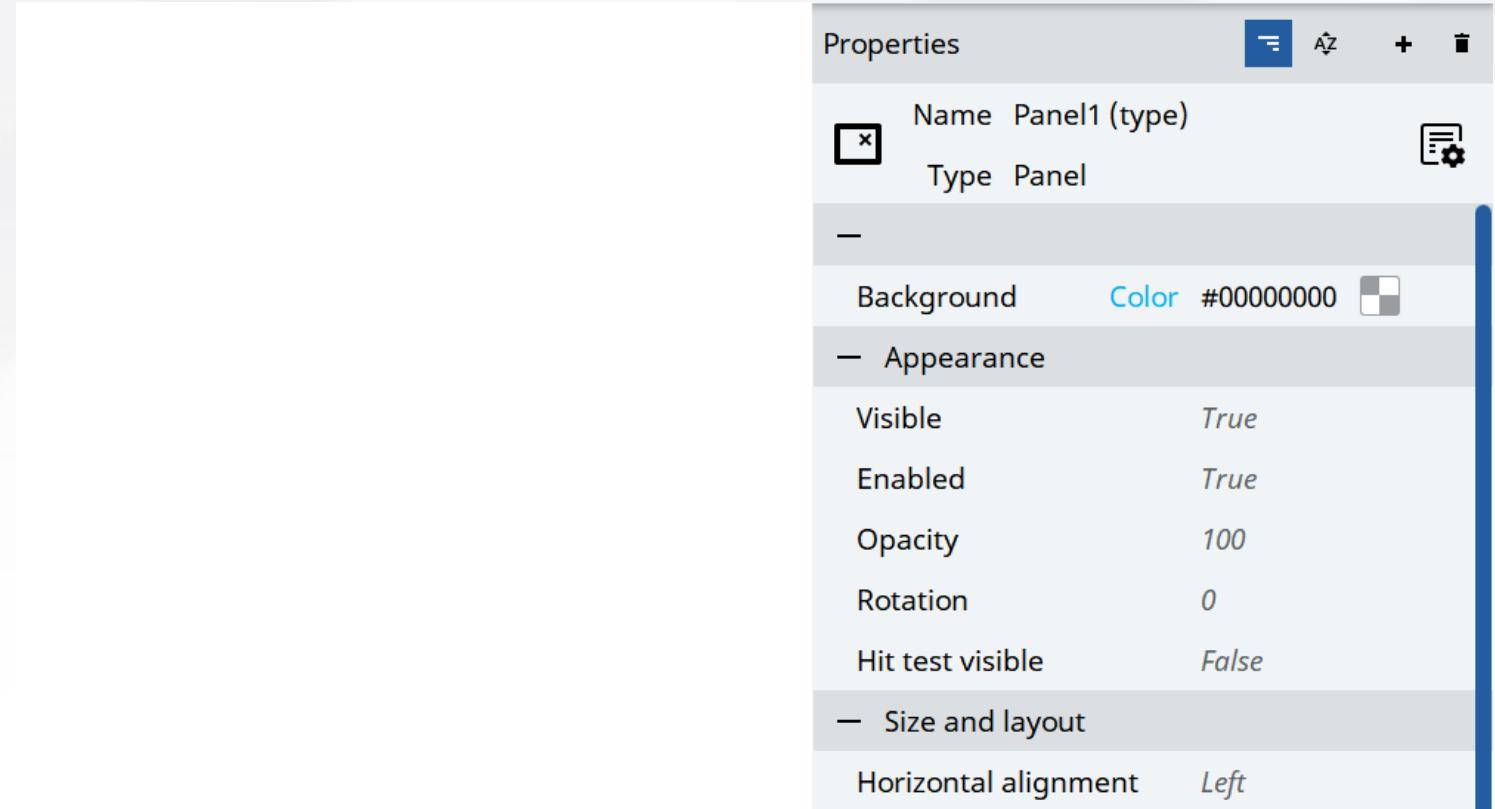
Custom types: reusable graphics with a "local variable"

- Object-Oriented Programming paradigm can be also applied to Graphics
 - "Panel_with_BG(type)" = Class = Type
 - "Panel1" = Instance
- Reusable Graphics can be
 - Widgets
 - Dialogs
 - Panels
- Example: a customized Panel with a rectangle used to set the background color



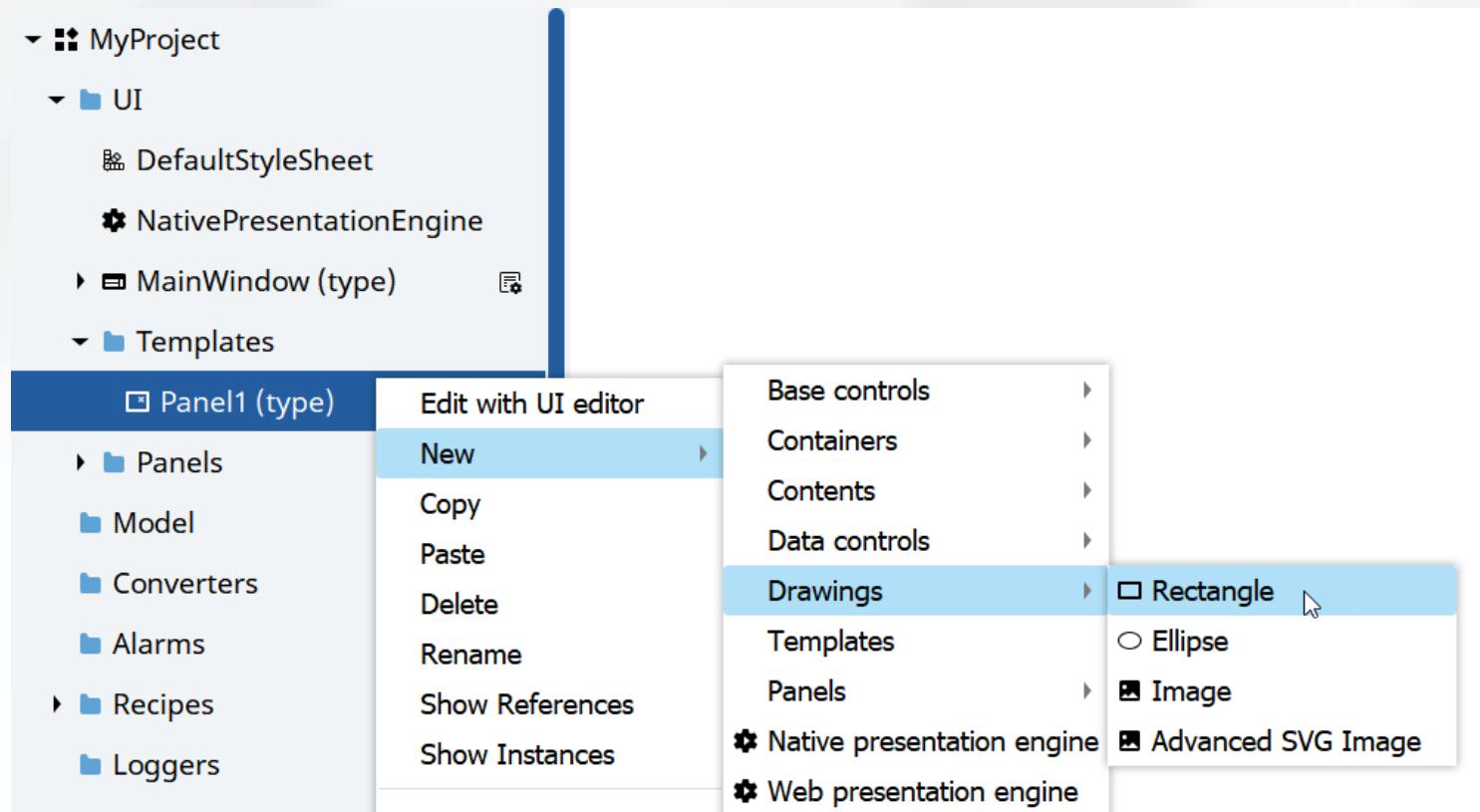
Custom types: reusable graphics with a "local variable"

- Object-Oriented Programming paradigm can be also applied to Graphics
 - "Panel_with_BG(type)" = Class = Type
 - "Panel1" = Instance
- Reusable Graphics can be
 - Widgets
 - Dialogs
 - Panels
- Example: a customized Panel with a rectangle used to set the background color



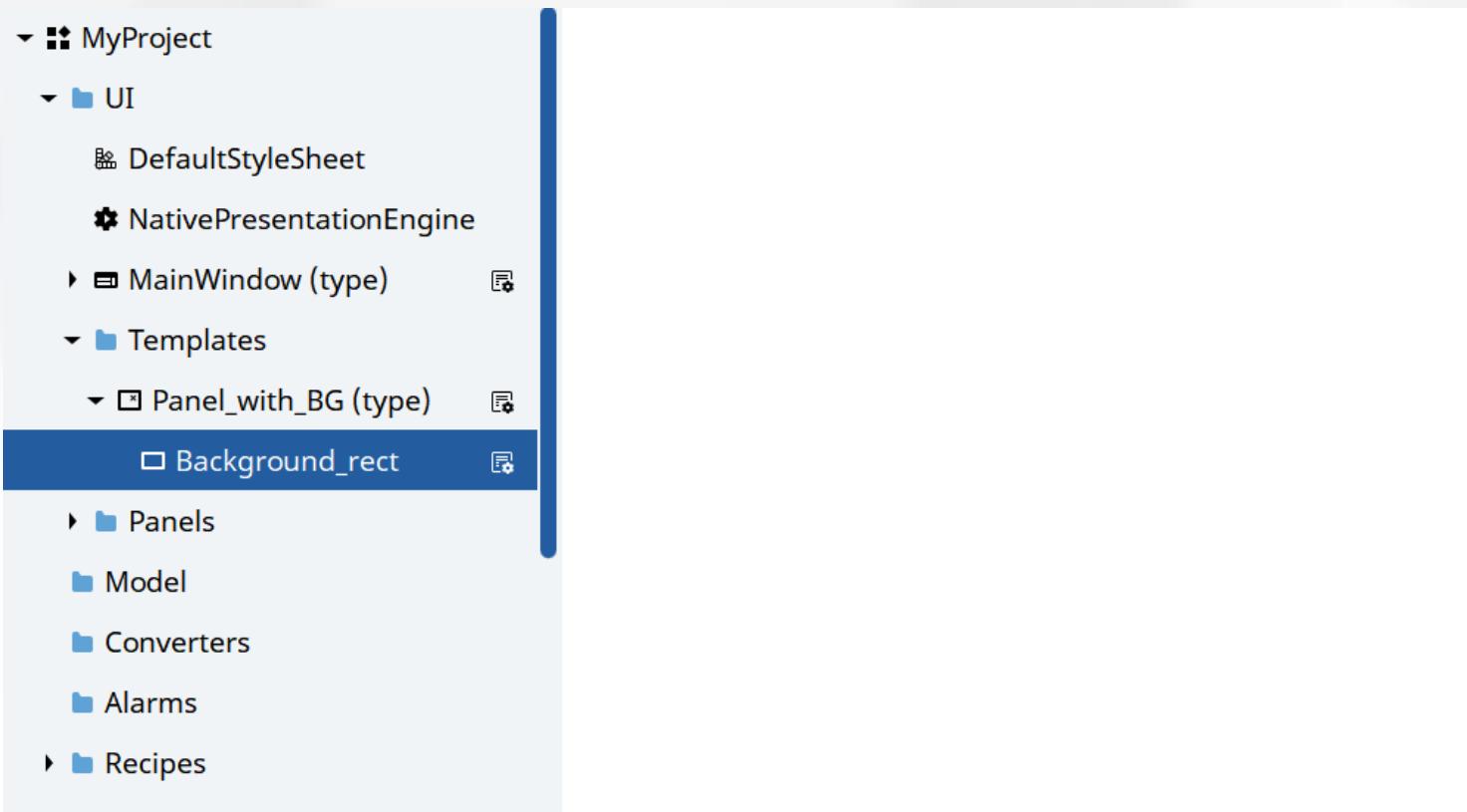
Custom types: reusable graphics with a "local variable"

- Object-Oriented Programming paradigm can be also applied to Graphics
 - "Panel_with_BG(type)" = Class = Type
 - "Panel1" = Instance
- Reusable Graphics can be
 - Widgets
 - Dialogs
 - Panels
- Example: a customized Panel with a rectangle used to set the background color



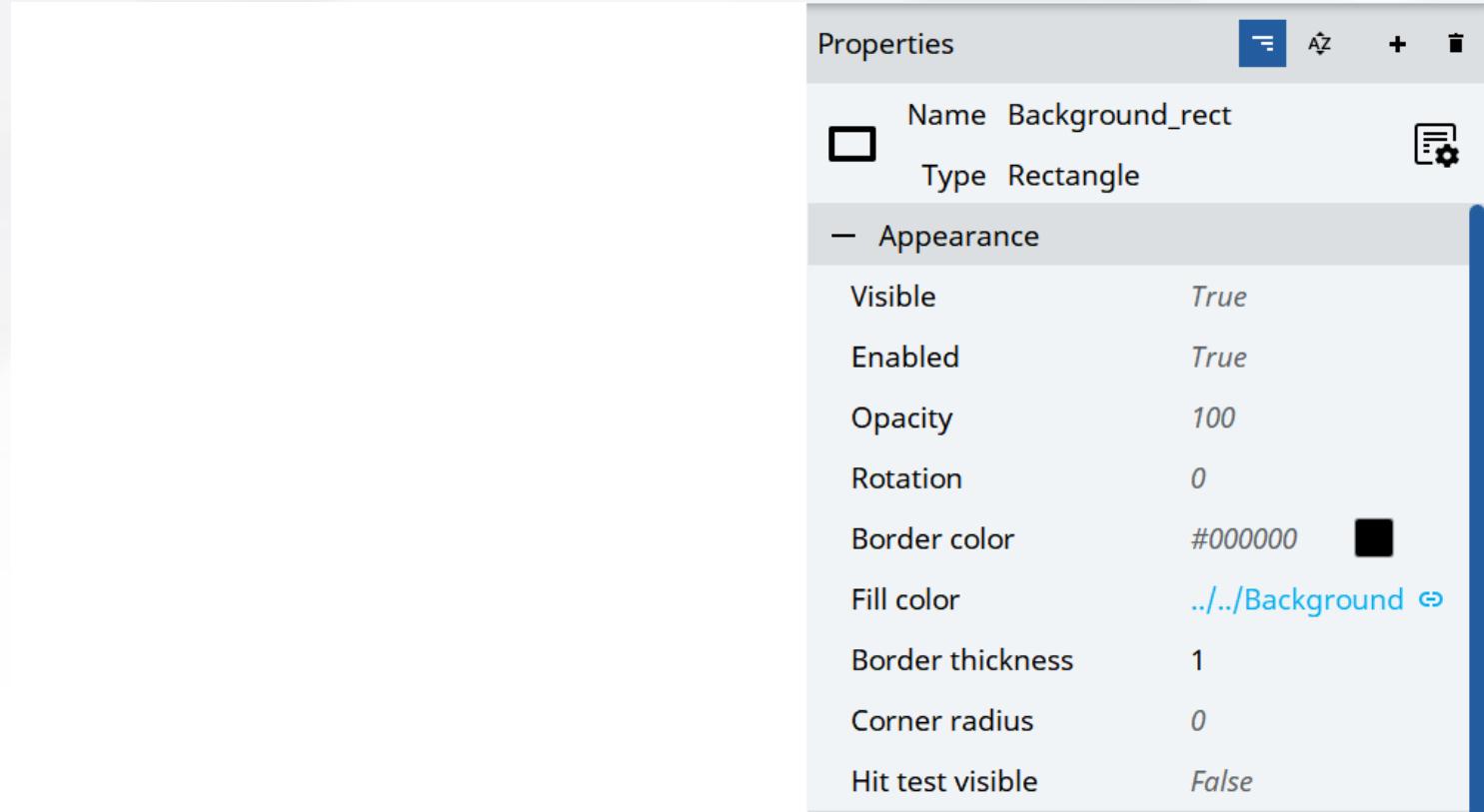
Custom types: reusable graphics with a "local variable"

- Object-Oriented Programming paradigm can be also applied to Graphics
 - "Panel_with_BG(type)" = Class = Type
 - "Panel1" = Instance
- Reusable Graphics can be
 - Widgets
 - Dialogs
 - Panels
- Example: a customized Panel with a rectangle used to set the background color



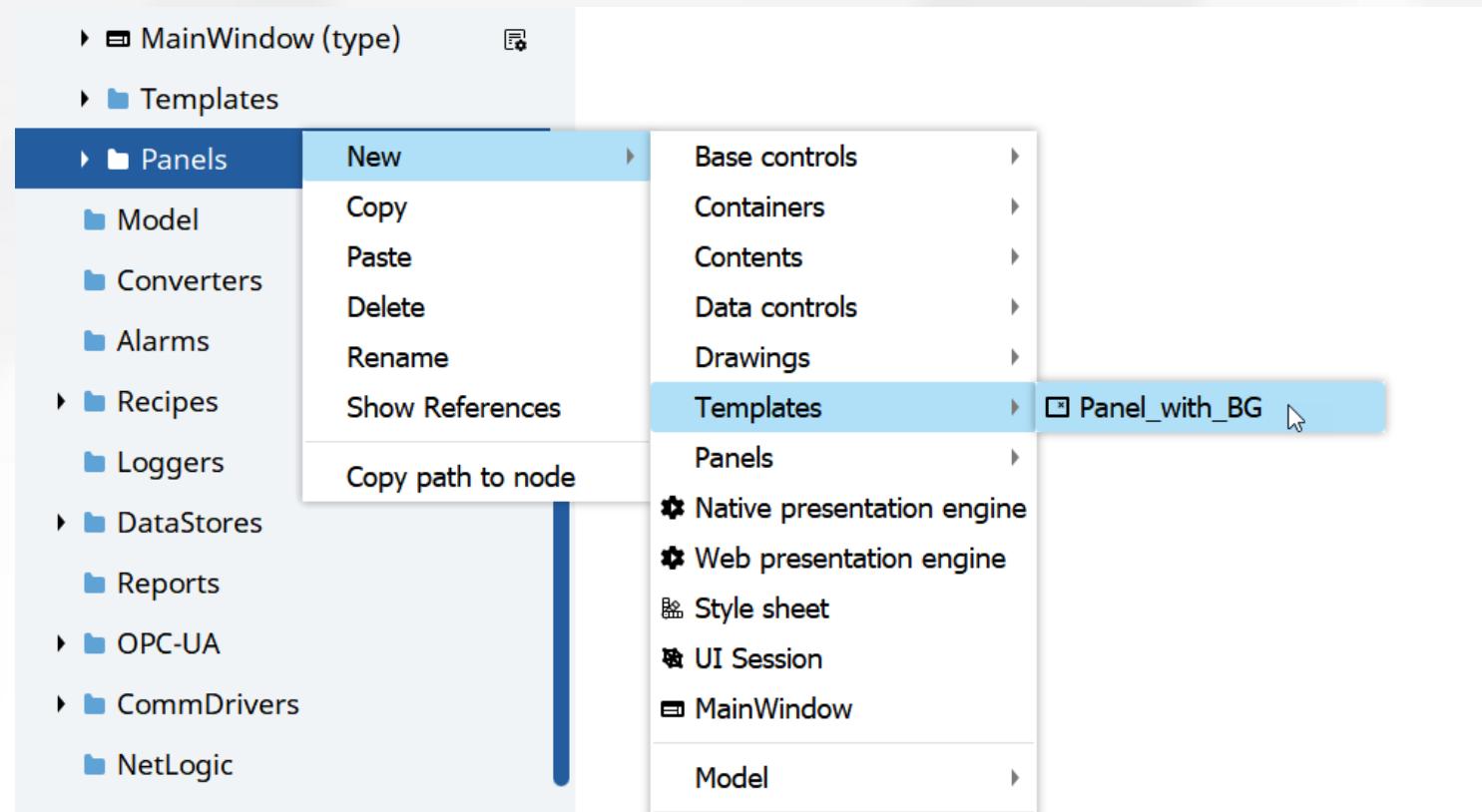
Custom types: reusable graphics with a "local variable"

- Object-Oriented Programming paradigm can be also applied to Graphics
 - "Panel_with_BG(type)" = Class = Type
 - "Panel1" = Instance
- Reusable Graphics can be
 - Widgets
 - Dialogs
 - Panels
- Example: a customized Panel with a rectangle used to set the background color



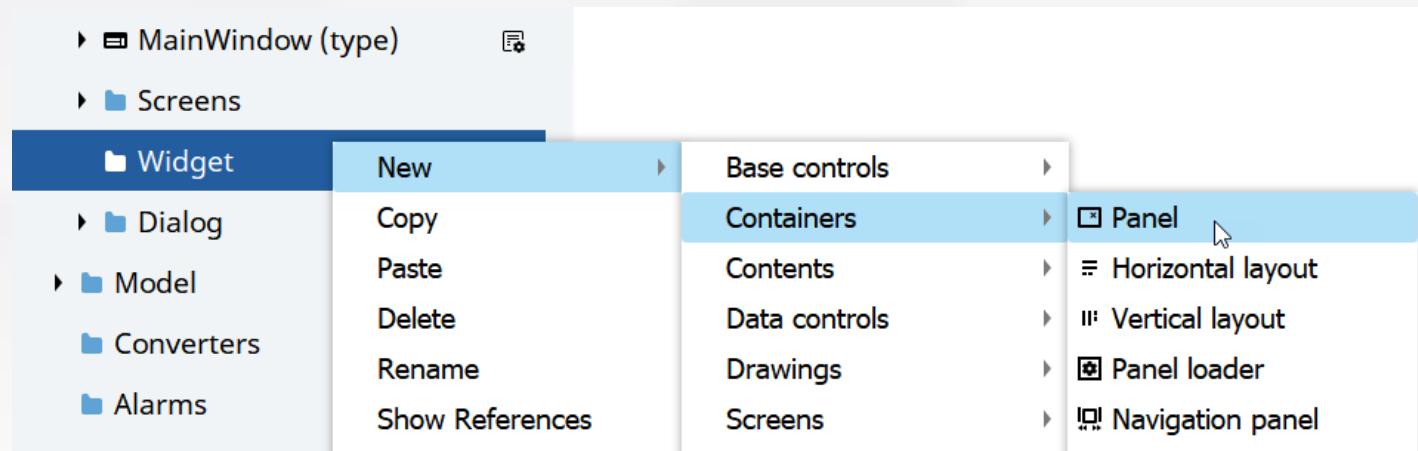
Custom types: reusable graphics with a "local variable"

- Object-Oriented Programming paradigm can be also applied to Graphics
 - "Panel_with_BG(type)" = Class = Type
 - "Panel1" = Instance
- Reusable Graphics can be
 - Widgets
 - Dialogs
 - Panels
- Example: a customized Panel with a rectangle used to set the background color



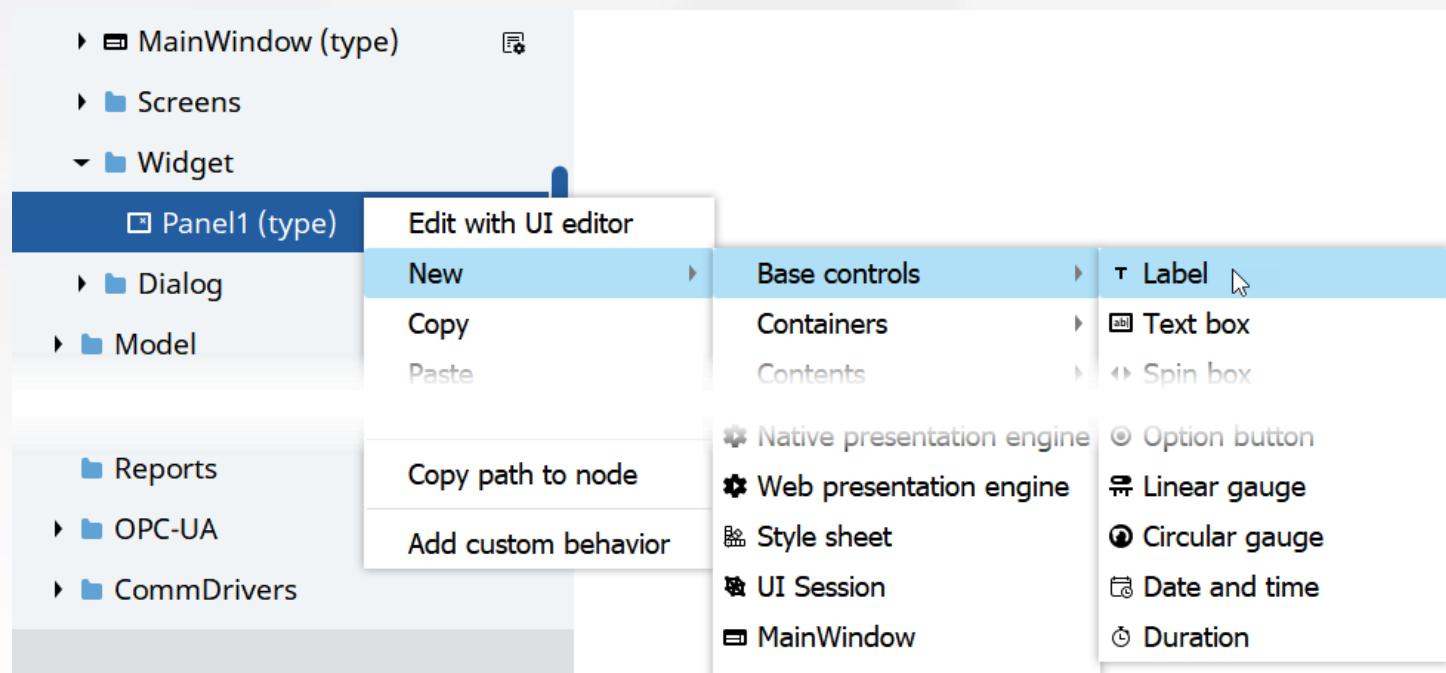
Custom types: reusable graphics with an alias

- Example: a motor widget that shows the values of different instances of the motor



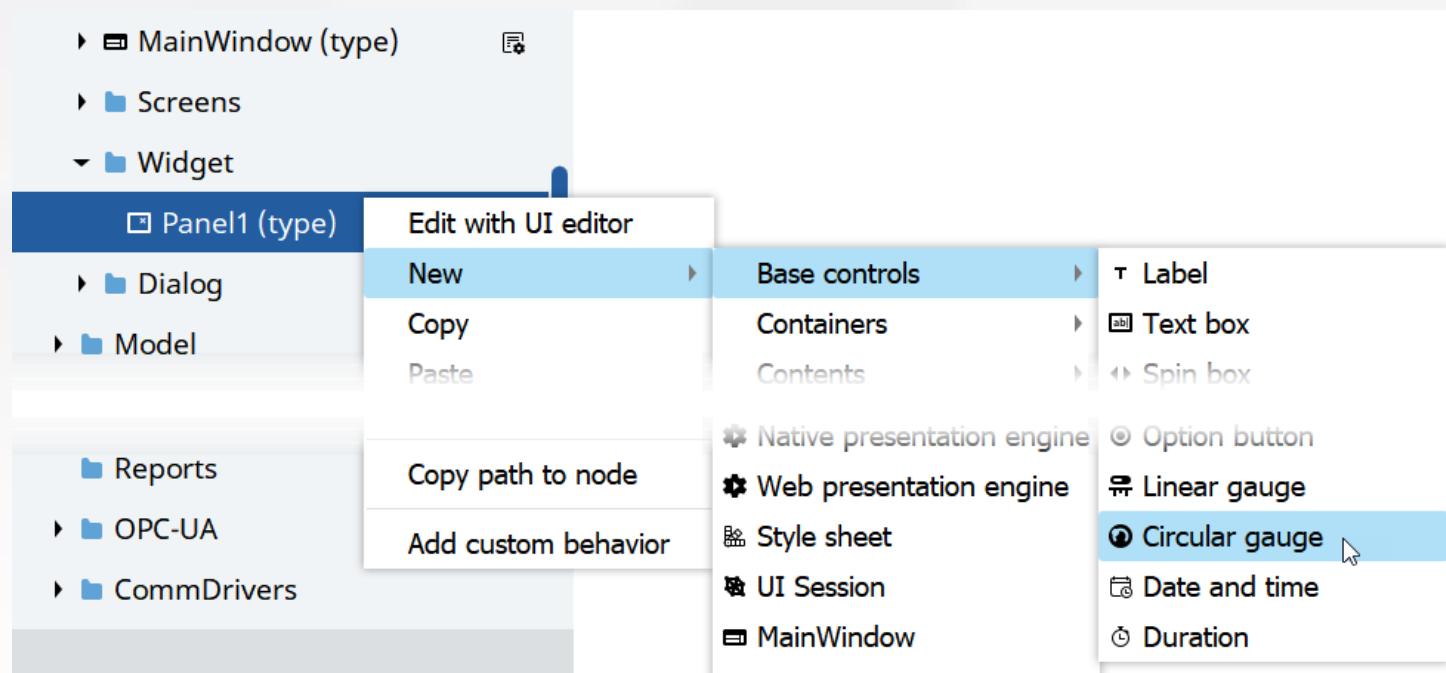
Custom types: reusable graphics with an alias

- Example: a motor widget that shows the values of different instances of the motor



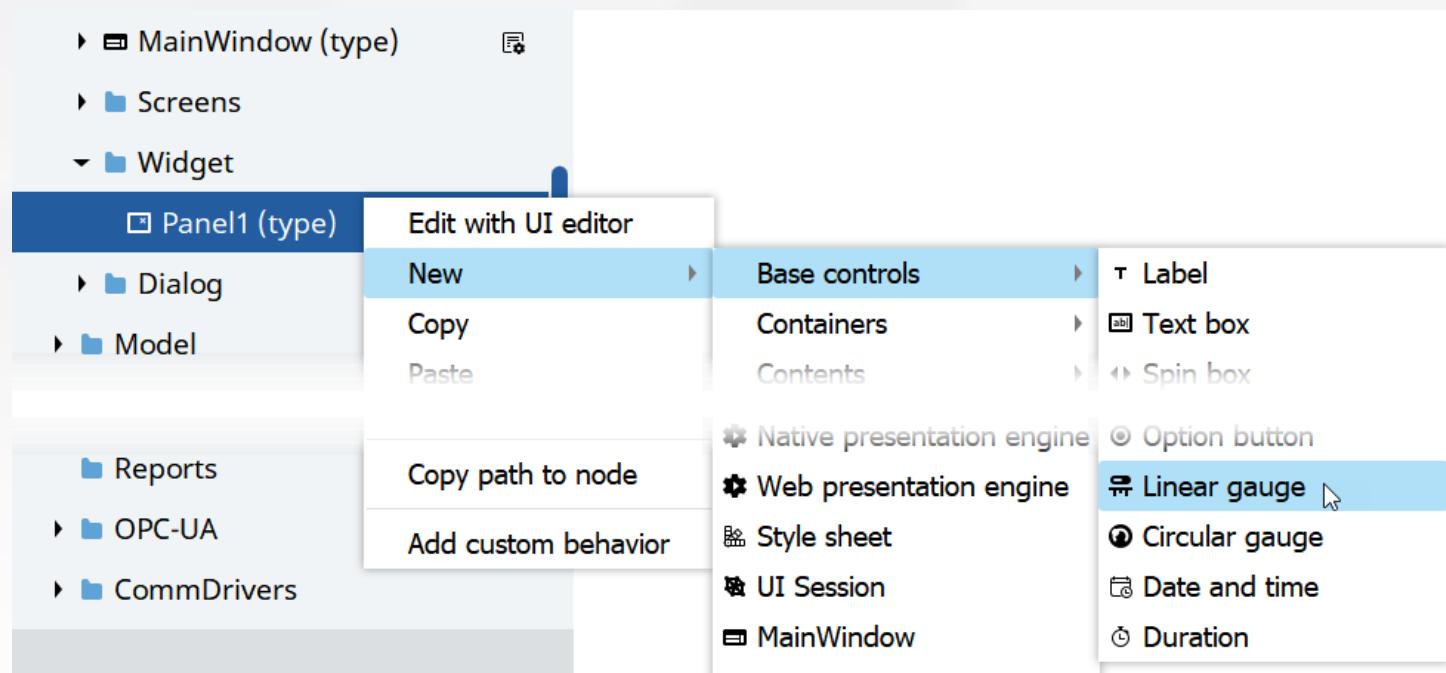
Custom types: reusable graphics with an alias

- Example: a motor widget that shows the values of different instances of the motor



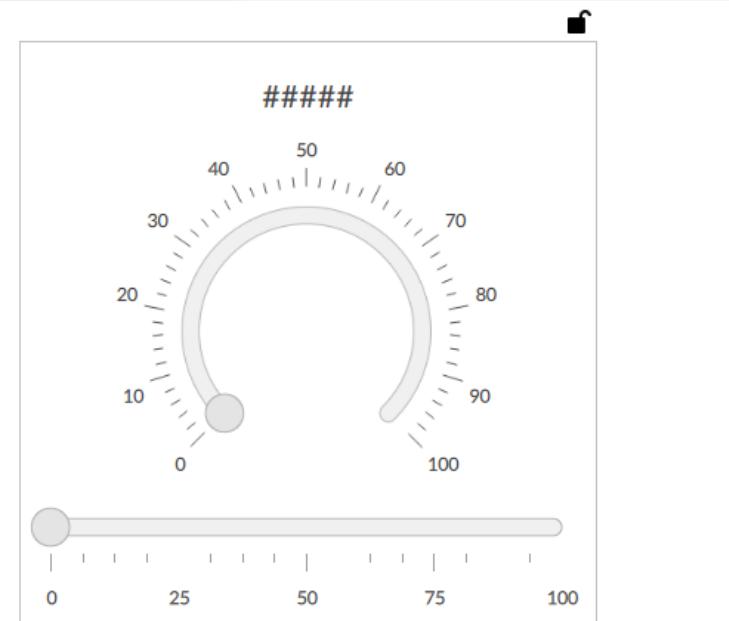
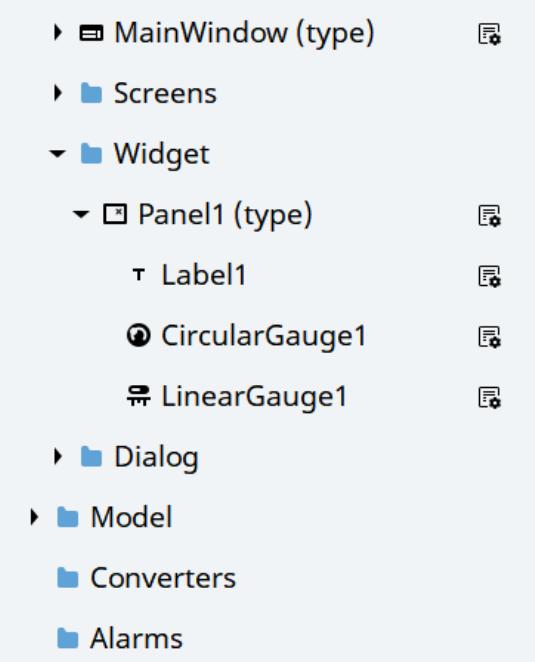
Custom types: reusable graphics with an alias

- Example: a motor widget that shows the values of different instances of the motor



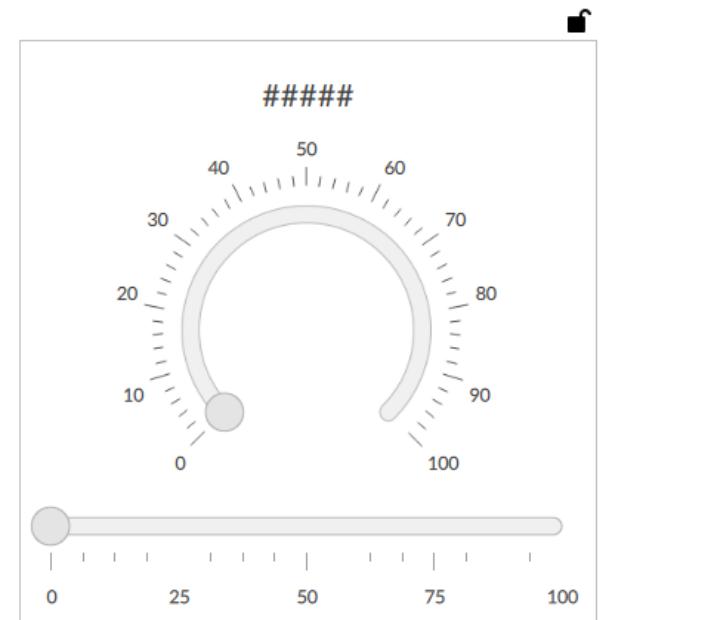
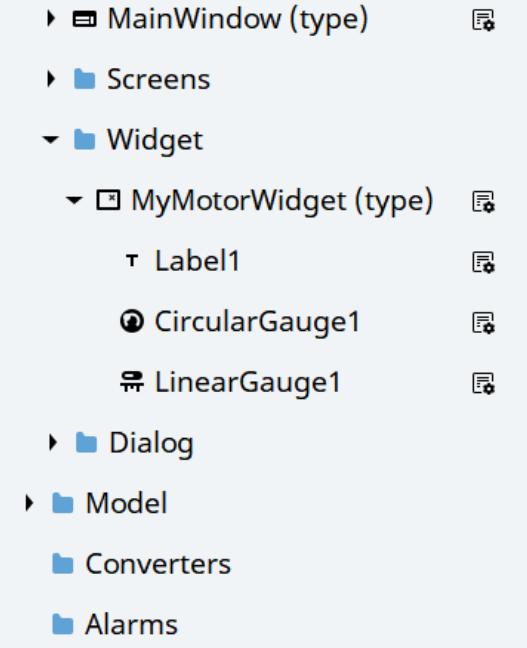
Custom types: reusable graphics with an alias

- Example: a motor widget that shows the values of different instances of the motor



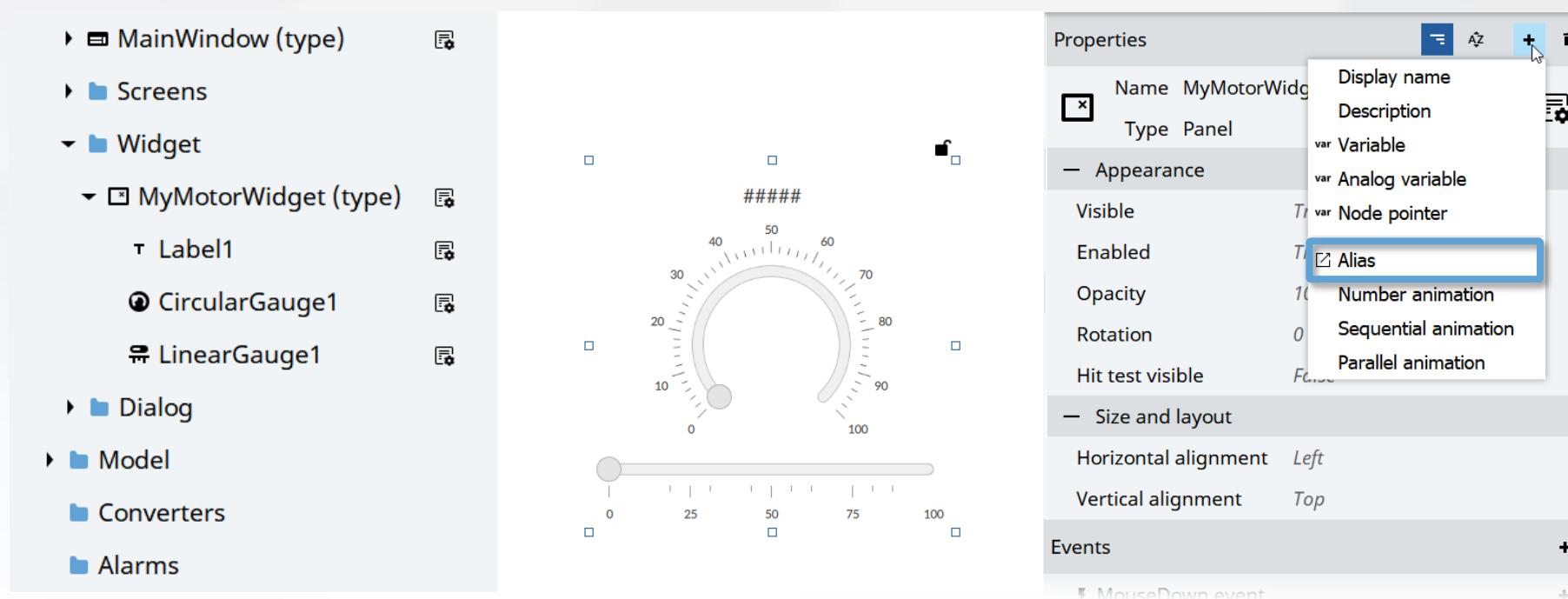
Custom types: reusable graphics with an alias

- Example: a motor widget that shows the values of different instances of the motor



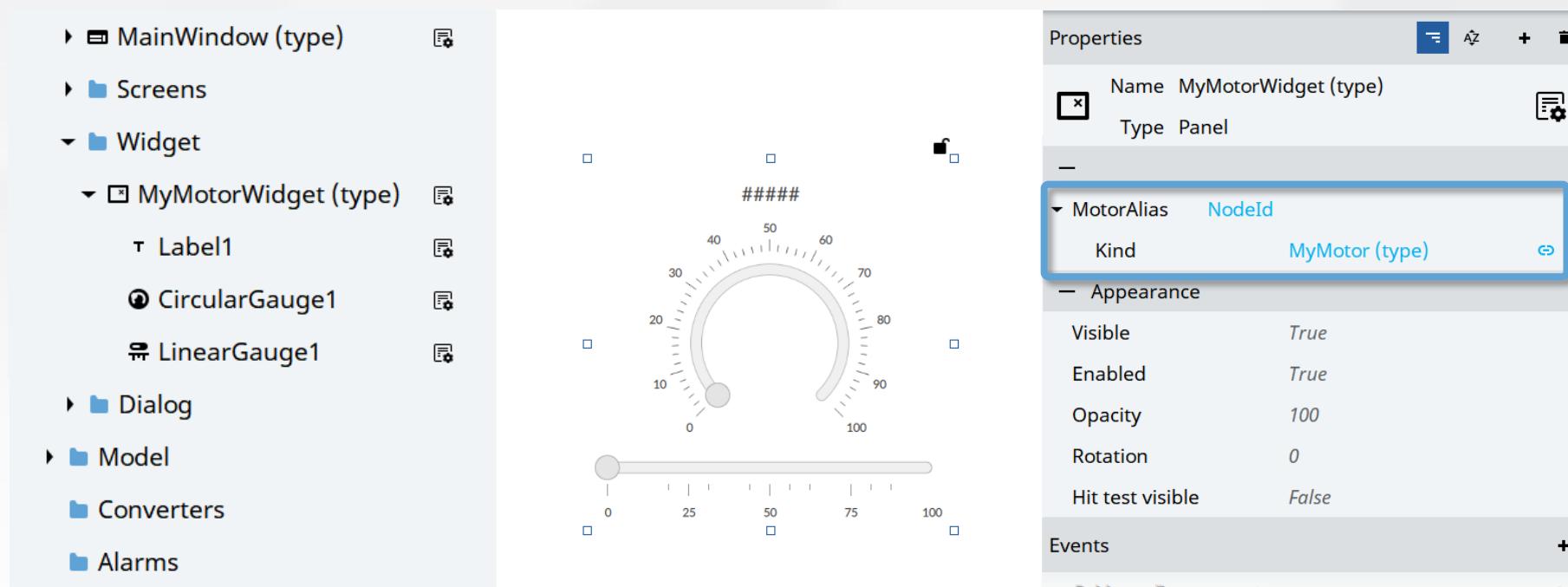
Custom types: reusable graphics with an alias

- Example: a motor widget that shows the values of different instances of the motor



Custom types: reusable graphics with an alias

- Example: a motor widget that shows the values of different instances of the motor



Custom types: reusable graphics with an alias

- Example: a motor widget that shows the values of different instances of the motor

The screenshot displays a software interface for creating graphical user interfaces (GUIs). On the left, a project tree lists several components:

- MainWindow (type)
- Screens
- Widget
 - MyMotorWidget (type)
 - Label1
 - CircularGauge1
 - LinearGauge1
- Dialog
- Model
- Converters
- Alarms

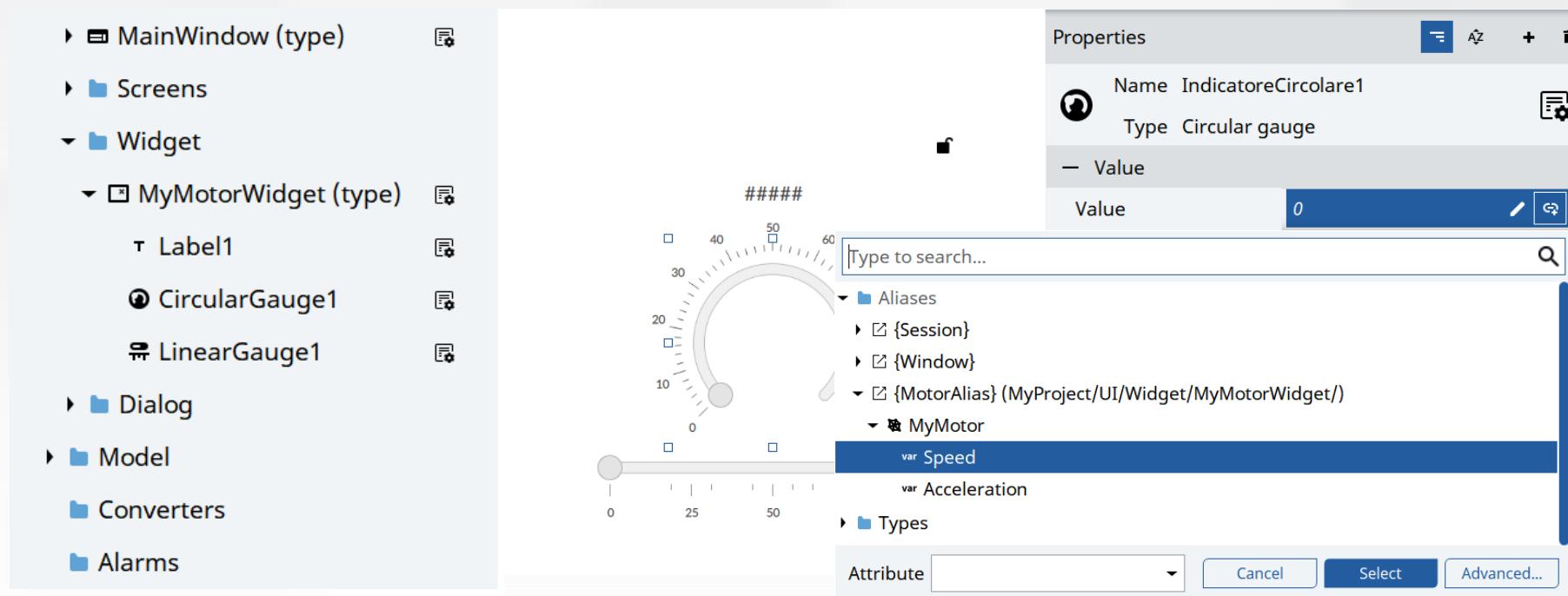
In the center, there is a circular gauge component with a scale from 0 to 100. The gauge has major tick marks at 0, 25, 50, 75, and 100, and minor tick marks every 10 units. The value is currently set to 0. Above the gauge, there is a placeholder text "#####".

On the right, the Properties panel is open for the "CircularGauge1" component. The properties are as follows:

Property	Value
Name	IndicatoreCircolare1
Type	Circular gauge
Value	0
Minimum value	0
Maximum value	100
Value change behavior	...ue change on edit finished
Gauge	
Start angle	-135
End angle	135
Editable	True
Events	+ ModifiedValueEvent

Custom types: reusable graphics with an alias

- Example: a motor widget that shows the values of different instances of the motor



Custom types: reusable graphics with an alias

- Example: a motor widget that shows the values of different instances of the motor

The screenshot displays a software interface for creating graphical user interfaces (GUIs). On the left, a project tree lists several components:

- MainWindow (type)
- Screens
- Widget
 - MyMotorWidget (type)
 - Label1
 - CircularGauge1
 - LinearGauge1
- Dialog
- Model
- Converters
- Alarms

In the center, there is a circular gauge component with a scale from 0 to 100. The gauge has major tick marks at 0, 25, 50, 75, and 100, and minor tick marks every 10 units. The value is currently set to 50. The gauge is labeled "IndicatoreCircolare1".

On the right, the "Properties" panel is open for the circular gauge:

- Name:** IndicatoreCircolare1
- Type:** Circular gauge
- Value**
 - Value:** {MotorAlias}/Speed
 - Minimum value:** 0
 - Maximum value:** 100
 - Value change behavior:** ...ue change on edit finished
- Gauge**
 - Start angle:** -135
 - End angle:** 135
 - Editable:** True
- Events**
 - + Modified value

Custom types: reusable graphics with an alias

- Example: a motor widget that shows the values of different instances of the motor

The screenshot displays a software interface for creating graphical user interfaces (GUIs). On the left, a project tree lists several components:

- MainWindow (type)
- Screens
- Widget
 - MyMotorWidget (type)
 - Label1
 - CircularGauge1
 - LinearGauge1
- Dialog
- Model
- Converters
- Alarms

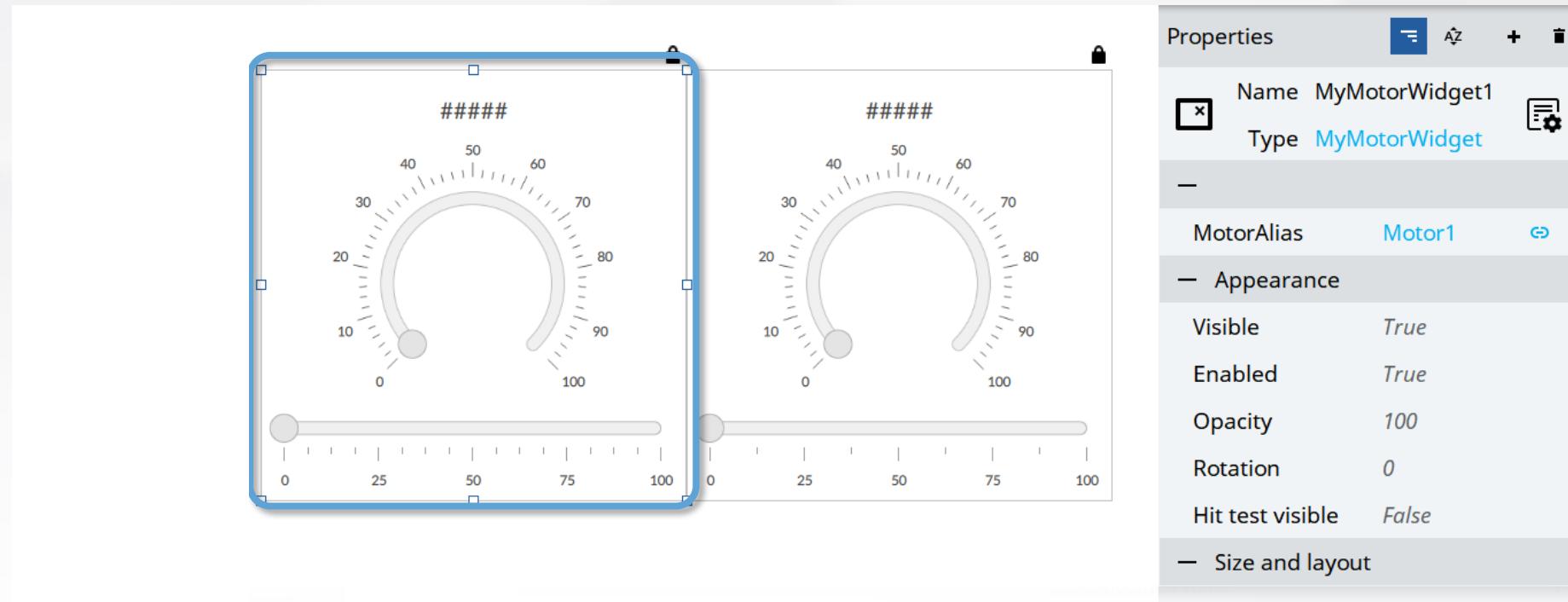
In the center, there is a circular gauge with a scale from 0 to 100. The needle is positioned at approximately 50. Above the gauge, the text "#####" is displayed. Below the gauge, a horizontal slider is shown with major ticks at 0, 25, 50, 75, and 100.

On the right, the "Properties" panel is open for the "IndicatoreLineare1" component, which is identified as a "Linear gauge". The properties are as follows:

- Name: IndicatoreLineare1
- Type: Linear gauge
- Value
 - Value: {MotorAlias}/Acceleration
 - Minimum value: 0
 - Maximum value: 100
 - Value change behavior: ...ue change on edit finished
- Gauge
 - Orientation: Horizontal
 - Editable: True
 - Major tick count: 5
- Events
 - Modified value

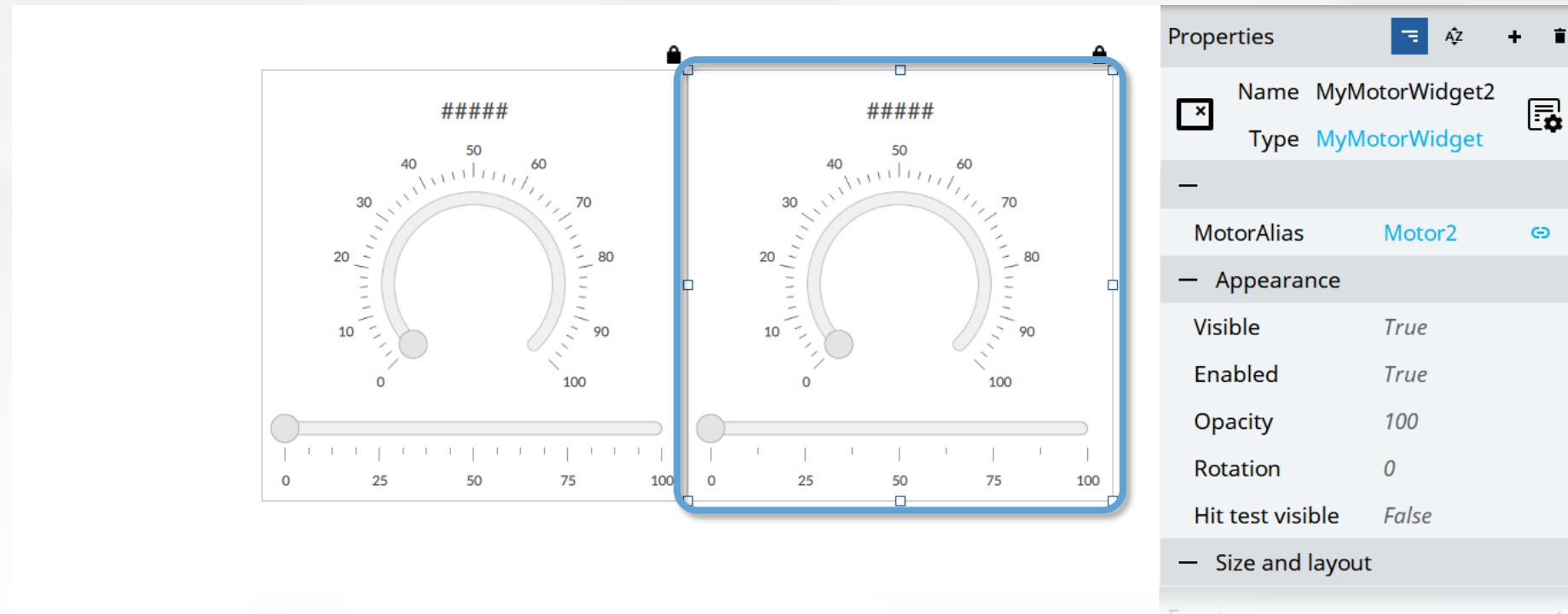
Custom types: reusable graphics with an alias

- Example: a motor widget that shows the values of different instances of the motor



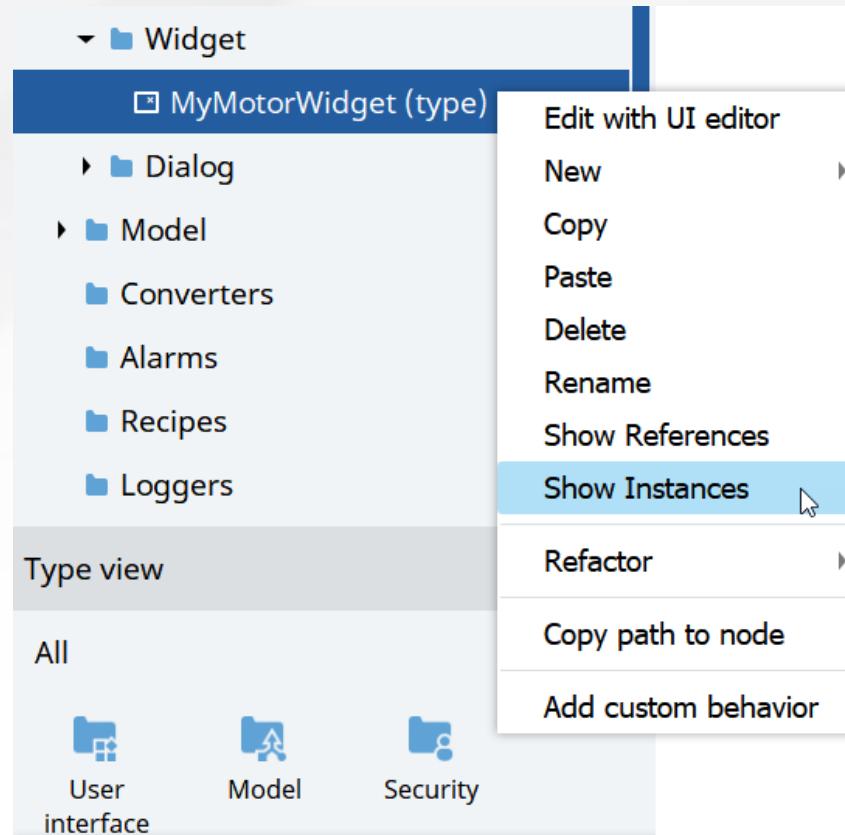
Custom types: reusable graphics with an alias

- Example: a motor widget that shows the values of different instances of the motor



Custom types: show instances

- Answer the question: "**where are the instances of this type?**"
- Can be used to find instances of
 - Widgets,
 - Panels,
 - Tags (Structure)
 - "typed" Alarm
 - ...

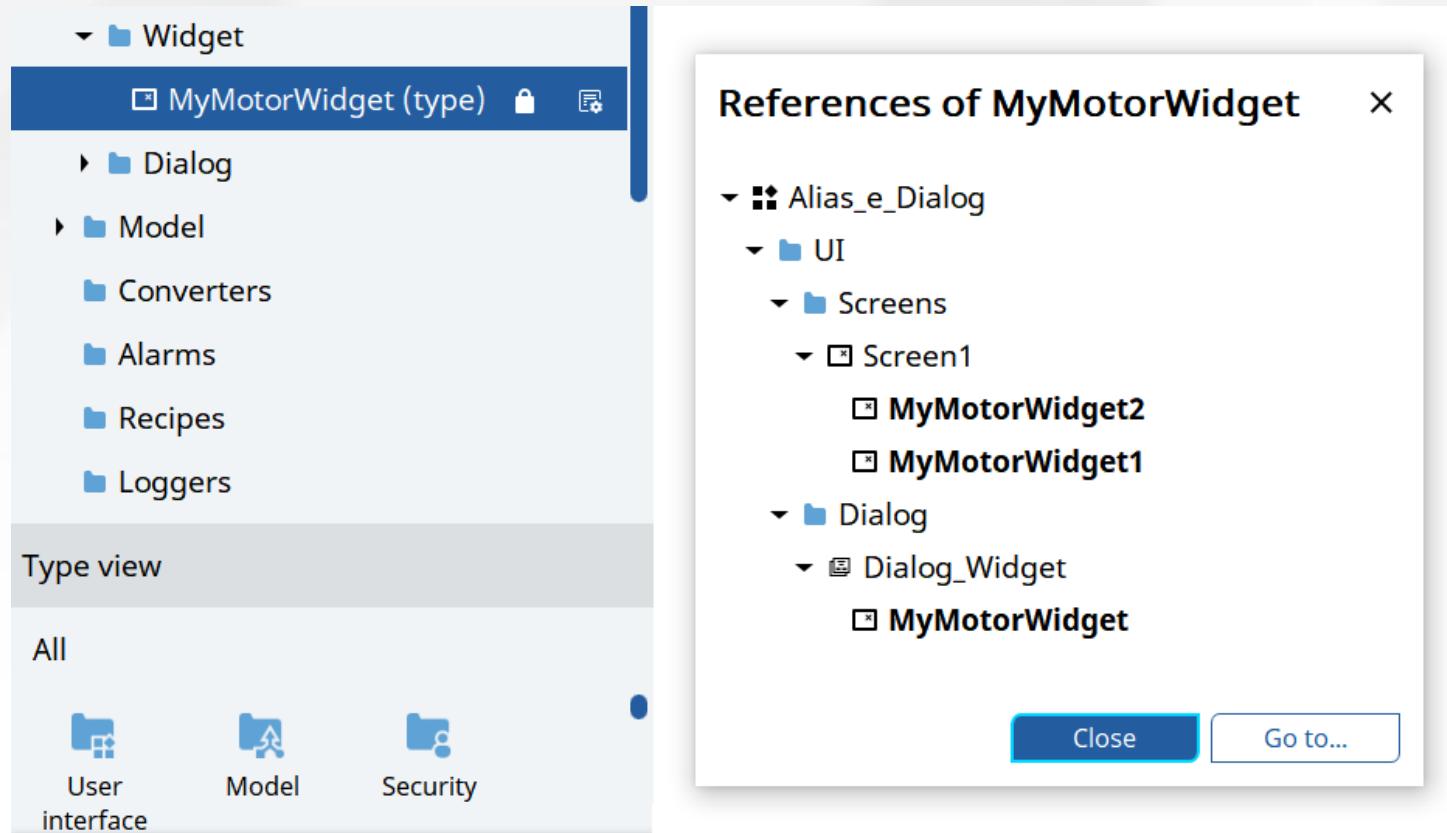


Custom types: show instances

- Answer the question: "where are the instances of this type?"

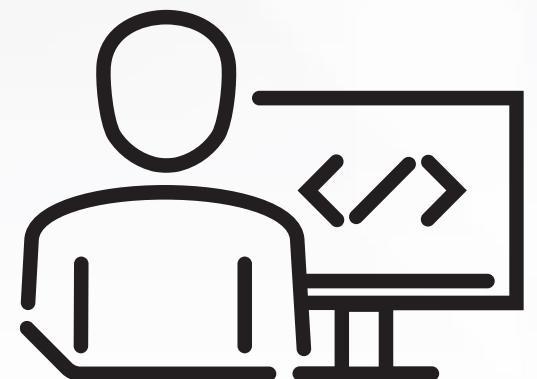
- Can be used to find instances of

- Widgets,
- Panels,
- Tags (Structure)
- "typed" Alarm
- ...



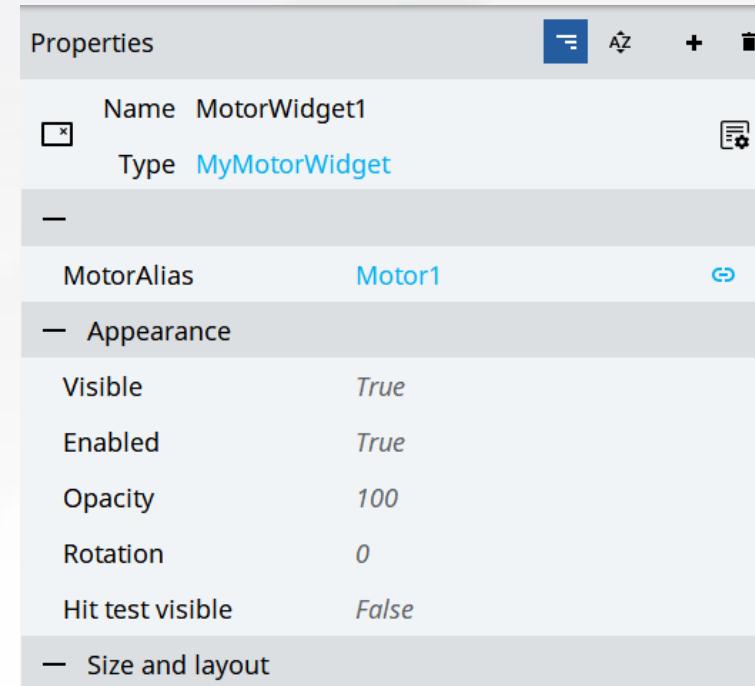
Hands-on session

- Create a widget called MotorWidget as a Type with an Alias to be used with the Motor variables
- Add 3 instances of the MotorWidget into a new Panel
- Specify, at design time, the alias on each MotorWidget instance associating to each Motor variable



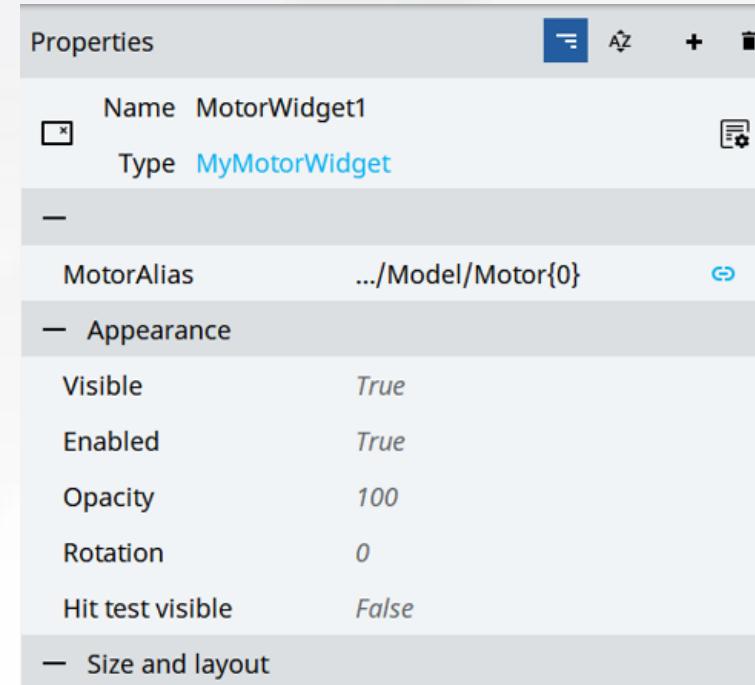
Reusable graphics: set aliases using indexing

- Alias variables can be indexed like simple variables
- Allow to easily change the alias of a widget



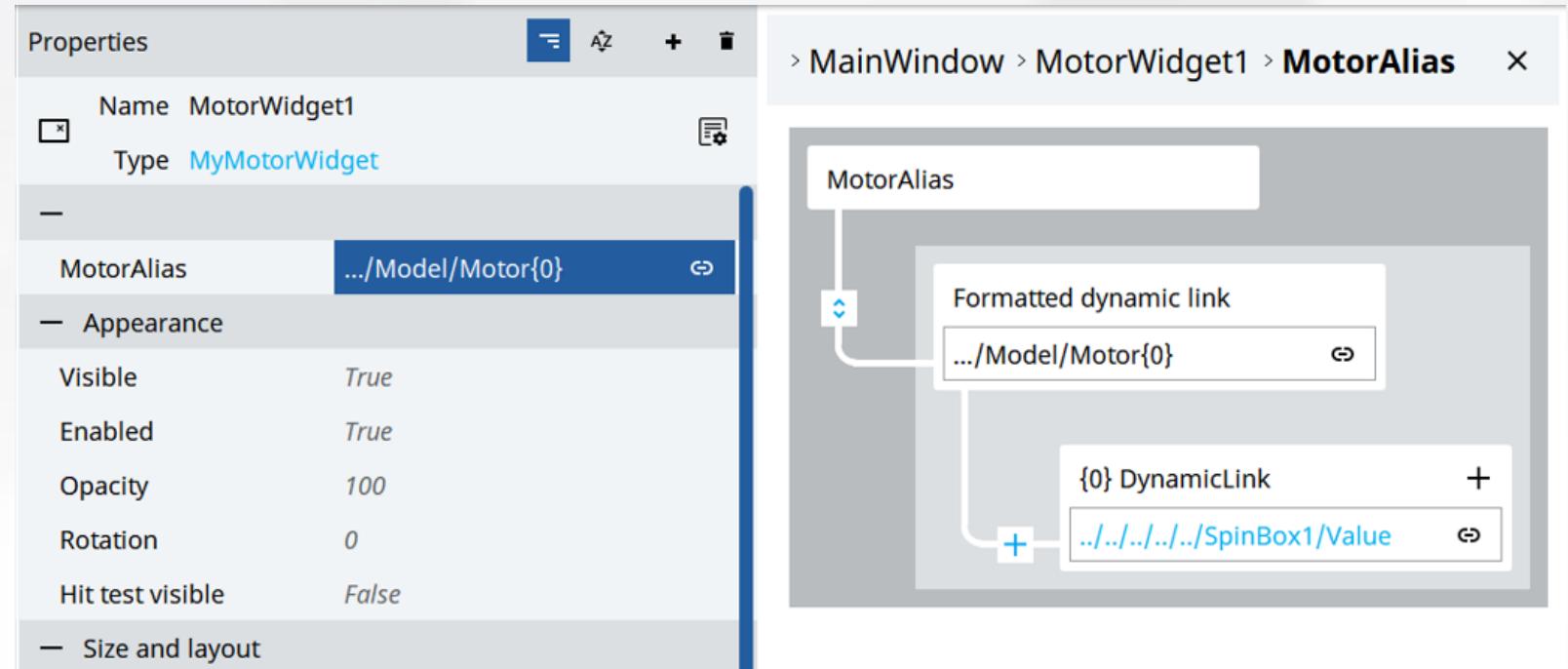
Reusable graphics: set aliases using indexing

- Alias variables can be indexed like simple variables
- Allow to easily change the alias of a widget



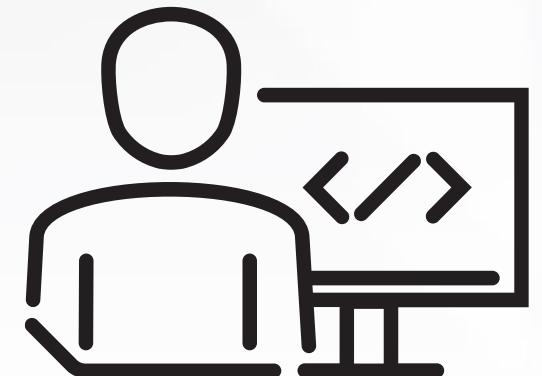
Reusable graphics: set aliases using indexing

- Alias variables can be indexed like simple variables
- Allow to easily change the alias of a widget



Hands-on session

- Add just only one instance of the MotorWidget into a new Panel
- add also a Switch object in the Panel
- use the "indexing" method to define the Alias on the MotorWidget instance, associating the index to the Switch value



Reusable graphics: set aliases at runtime

- For Dialog and Panels,
the Alias at Runtime can be set on:

- Open Dialog command
- Change Panel command
- Navigation Panel properties
- Panel Loader properties

Events		
⚡ MouseClick event		+
⚡ MouseDown event		+
▼ Method	.../UICommands/OpenDialog	■
▼ Input arguments		
Dialog	Dialog_with_Alias (type)	🔗
AliasNode	Motor1	🔗
⚡ MouseUp event		+

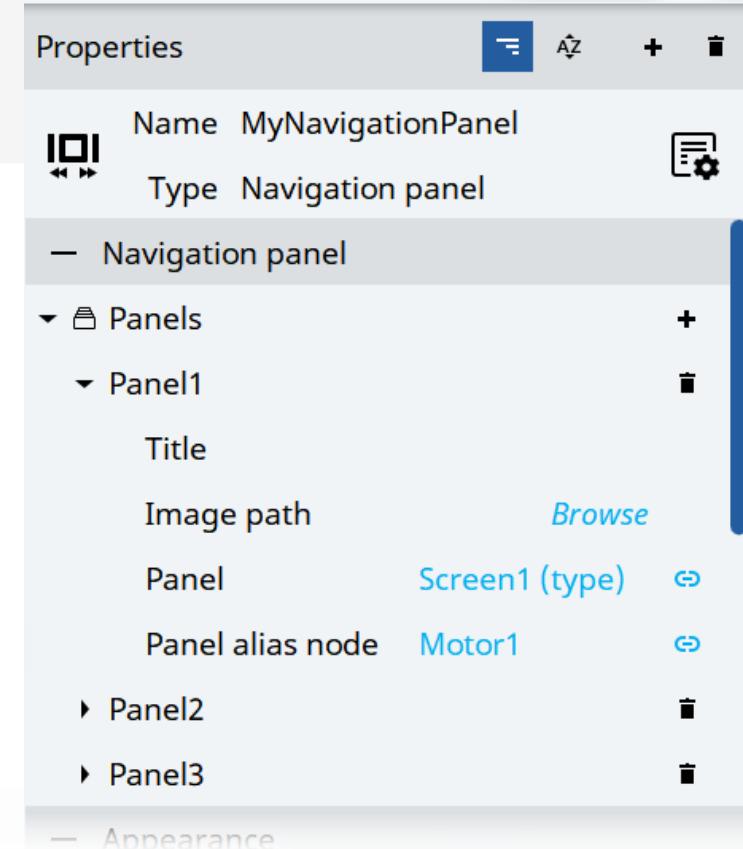
Reusable graphics: set aliases at runtime

- For Dialog and Panels,
the Alias at Runtime can be set on:
 - Open Dialog command
 - Change Panel command
 - Navigation Panel properties
 - Panel Loader properties

Events		+
▼	⚡ MouseClick event	+
▼	Method	.../PanelLoader/ChangePanel
▼	Input arguments	
NewPanel	Screen1 (type)	⊖
AliasNode	Motor1	⊖
⚡ MouseDown event		+
⚡ MouseUp event		+

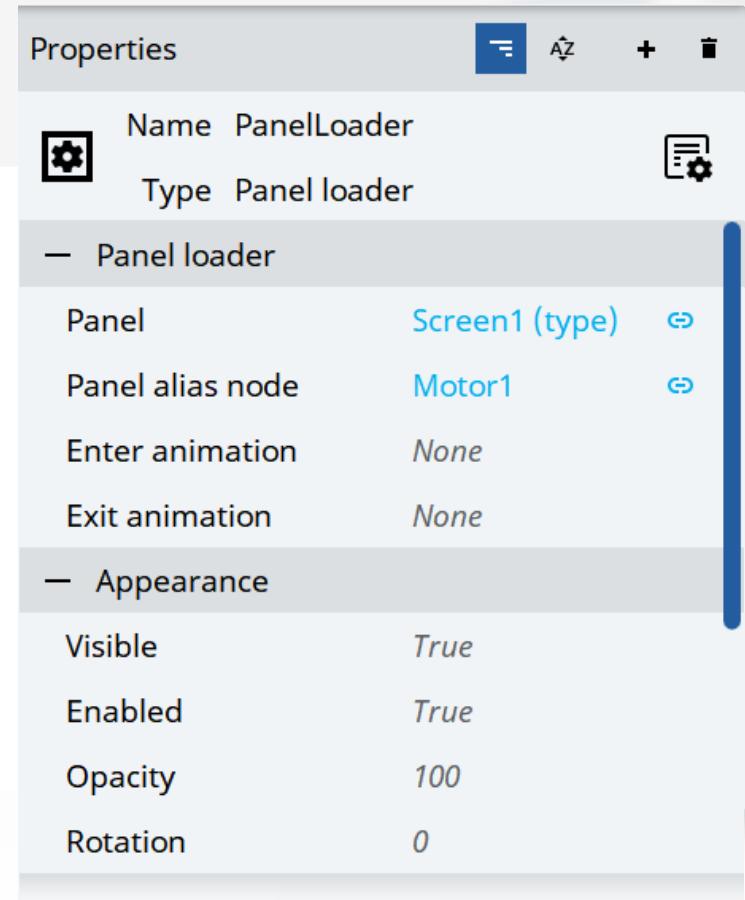
Reusable graphics: set aliases at runtime

- For Dialog and Panels,
the Alias at Runtime can be set on:
 - Open Dialog command
 - Change Panel command
 - Navigation Panel properties
 - Panel Loader properties



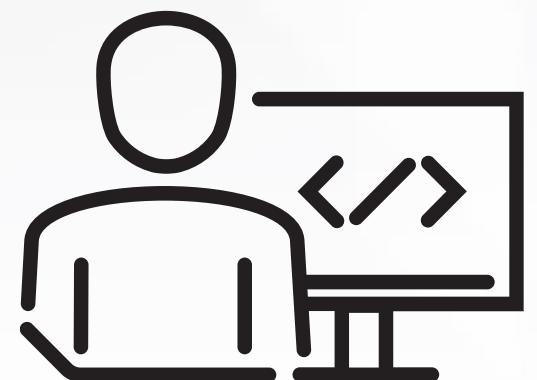
Reusable graphics: set aliases at runtime

- For Dialog and Panels,
the Alias at Runtime can be set on:
 - Open Dialog command
 - Change Panel command
 - Navigation Panel properties
 - Panel Loader properties



Hands-on session

- Add 3 Dropdown buttons (it's under the "container" group) into a new Panel
- Configure each button, with the following properties:
 - Panel = MotorWidget type
 - Panel alias node = the Alias of a Motor variable
- NOTE: you "probably" need to add a background to the MotorWidget type



Getting a PRO license



How to get a PRO Demo license

- Login to Rockwell Commerce portal
- Search for «FactoryTalk Optix DEMO»
- Add it to the cart and process the 0\$ payment

- License request should be handled in 24h
- This can only be requested by RA employee



Rockwell commerce link

A screenshot of a web page from the Rockwell Automation commerce site. On the left, there's a graphic showing two overlapping screens displaying industrial monitoring and control software interfaces. To the right of the graphic, the product title "FactoryTalk Optix DEMO" is displayed in red. Below the title, a detailed description explains that the demo entitlement is for authorized distributors and RA employees, is free, and requires logging in at FactoryTalkHub.com. It also mentions that the license includes a FactoryTalk Studio Pro entitlement. A "Show more" link is visible. At the bottom right, a red "CONTINUE" button is present.

How to buy a PRO license

- Login to Rockwell Commerce portal
- Search for «FactoryTalk Optix»
- Add it to the cart and process the payment
- This is for Rockwell customers



Rockwell commerce link

The screenshot shows a product listing for 'FactoryTalk Optix'. On the left, there's a large image of two computer monitors displaying complex industrial control dashboards with various graphs, charts, and status indicators. To the right of the image, the product name 'FactoryTalk Optix' is displayed in red text. Below the name is a detailed product description in black text. At the bottom right of the page, there is a red button labeled 'CONTINUE'.

FactoryTalk Optix

FactoryTalk Optix is a cloud-enabled HMI platform that enables design, test and deployment of applications directly from a web browser, connecting through FactoryTalk Hub. This modular platform delivers new options for design, deployment and graphics design, with extensibility options that make it flexible enough for any application. FactoryTalk Optix is truly visualization for visionaries. An open, extensible architecture, along with responsive graphics, version control, C# scripting, remote application management, and full integration with the OPC UA specification make

[Show more](#)

[CONTINUE](#)

Activate a PRO license

- Optix license is called «entitlement»
- Entitlements are assigned to the user that bought the license through the RA Commerce portal
- Entitlements have to be assigned to a FT Hub organization
- Users of an organization can self-assign any available entitlement
- Entitlements can be released once they are not needed anymore and used by a different user in the same organization

Entitlement details:

Entitlement Key	User	Organization
Contract Number	Catalog Number	Purchase Date
Expiration Date	Activation Date	Entitlement optix-ide-pro
Duration 365 days		

Full entitlements tutorial →



A large, abstract graphic of blue dots forms a winding, undulating line that spans the width of the slide, starting from the bottom left and curving upwards towards the top right.

Develop using version
control and collaboration



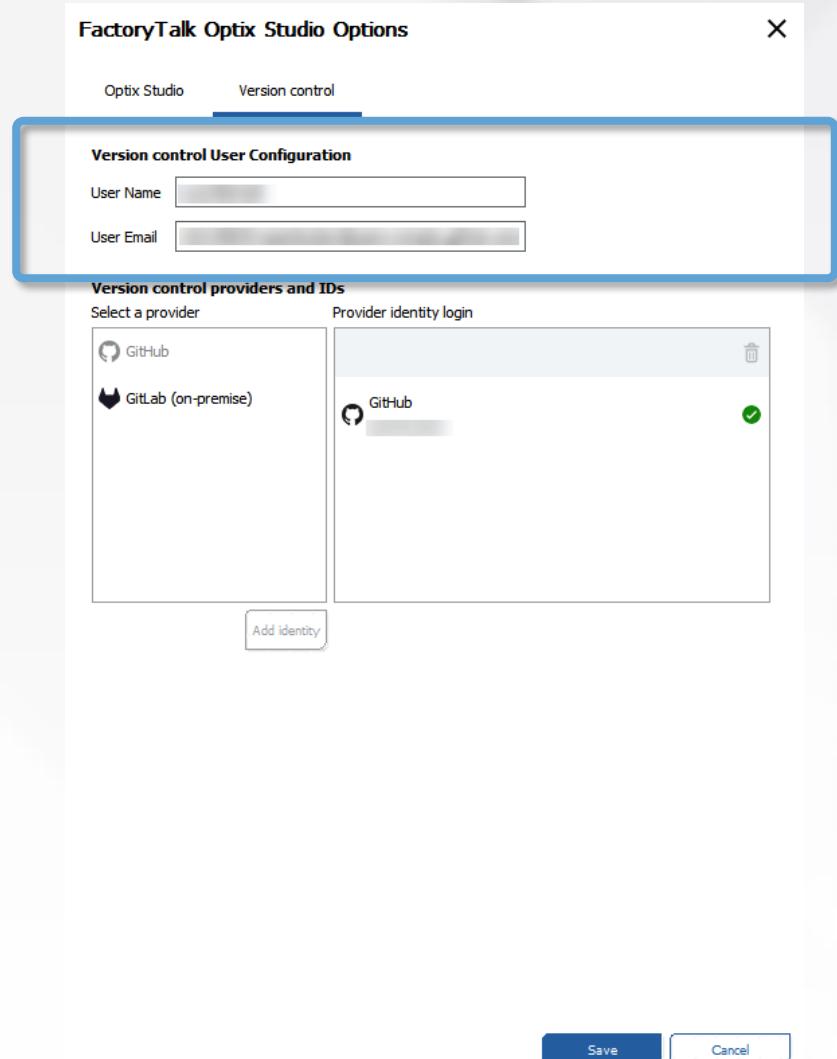
Version control and Collaboration

• Version Control

- used for keeping track of incrementally-different versions
- work also locally (offline)

• Collaboration

- distributed version control
- form of version control in which the full history, is mirrored on every developer's computer
- enables automatic management of merges,
- does not rely on a single location for backups
- Git is a distributed version control system
- GitHub is an Internet hosting service using Git



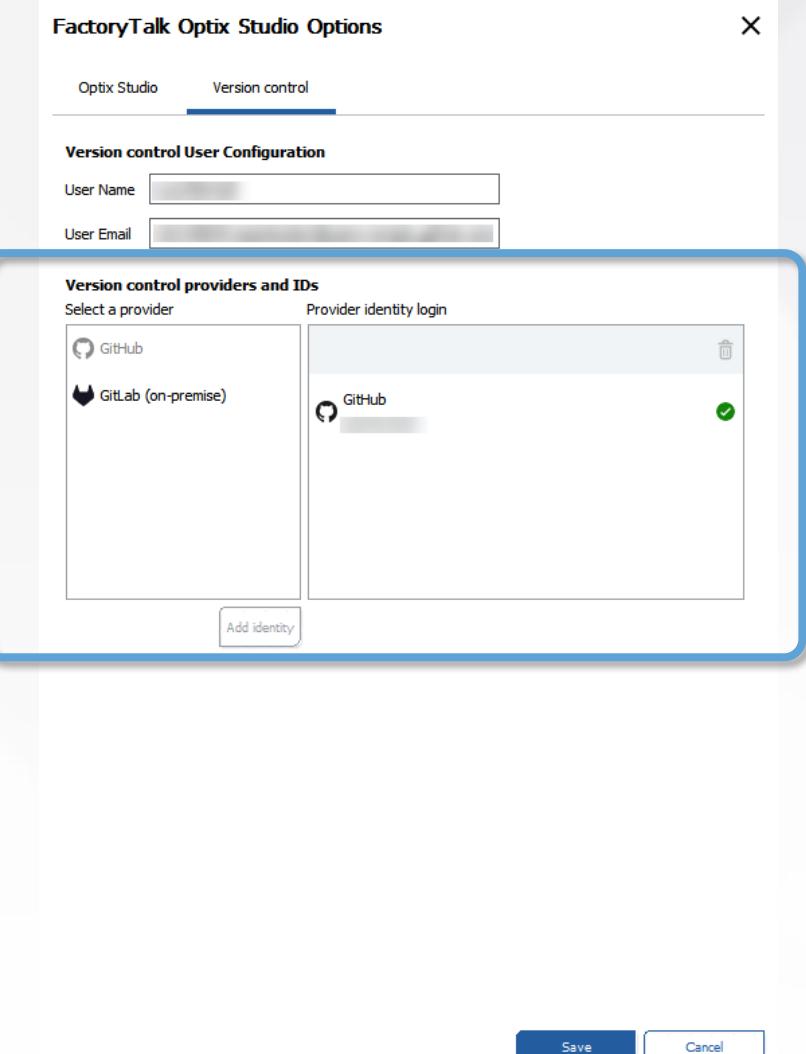
Version control and Collaboration

• Version Control

- used for keeping track of incrementally-different versions
- work also locally (offline)

• Collaboration

- distributed version control
- form of version control in which the full history, is mirrored on every developer's computer
- enables automatic management of merges,
- does not rely on a single location for backups
- Git is a distributed version control system
- GitHub is an Internet hosting service using Git



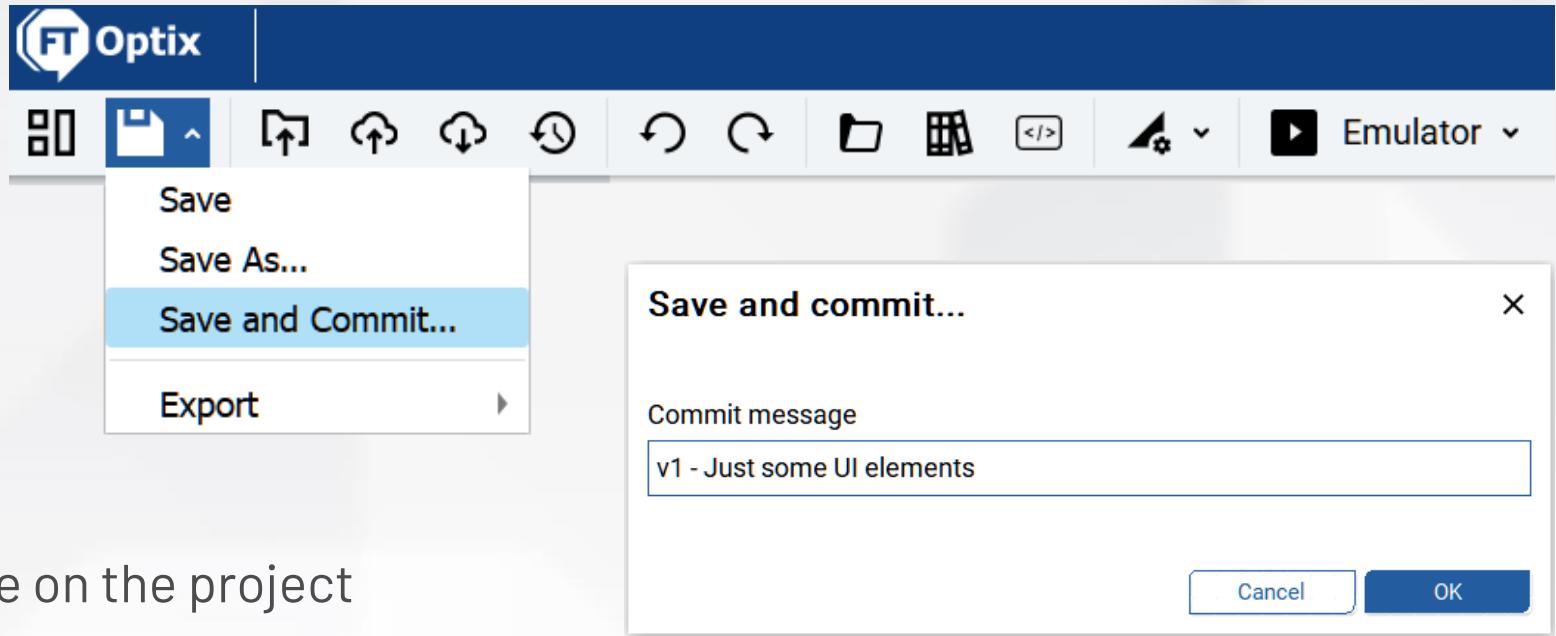
Version control: save and commit locally

- Save and Commit,
allow saving the project
keeping track of changes
- Works "Local"
- Ideal for those who work alone
with no colleagues that collaborate on the project
- No more needs to make zip files named v1, v2...
- The history button show differences,
and allow restoring a previous version



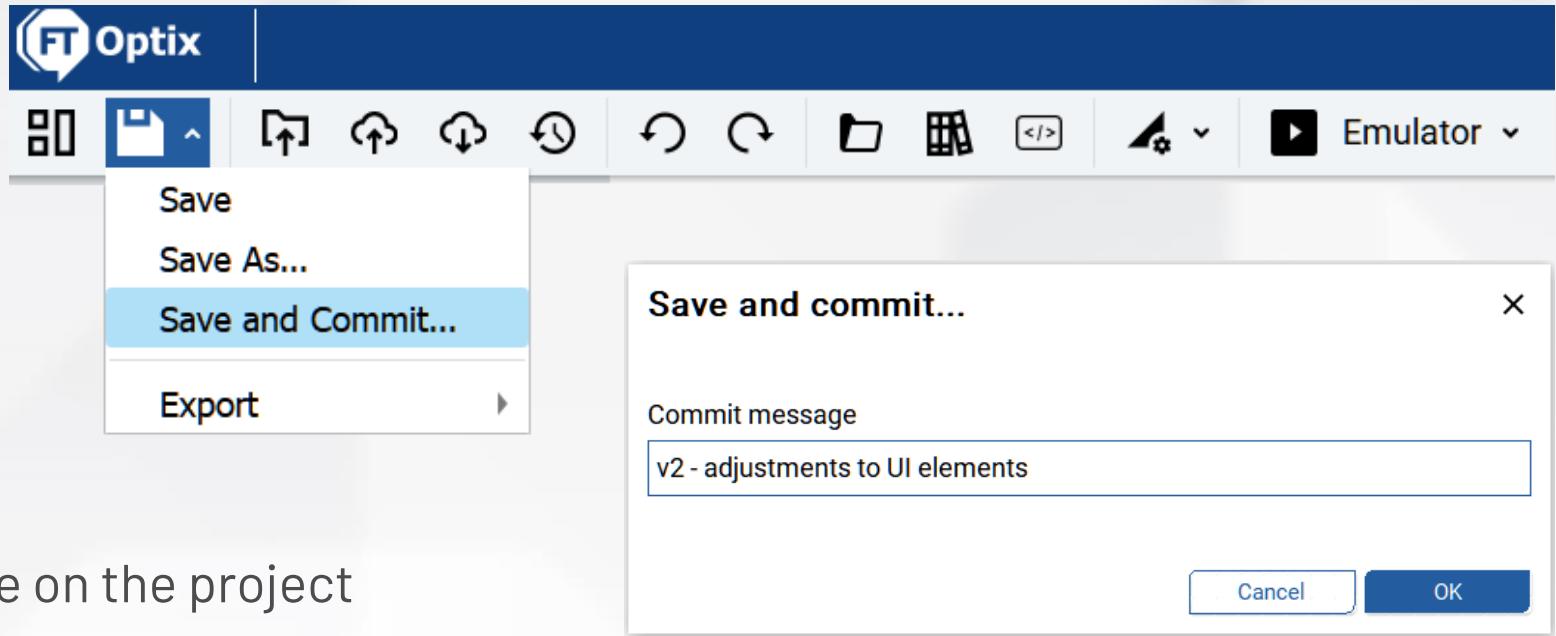
Version control: save and commit locally

- Save and Commit,
allow saving the project
keeping track of changes
- Works "Local"
- Ideal for those who work alone
with no colleagues that collaborate on the project
- No more needs to make zip files named v1, v2...
- The history button show differences,
and allow restoring a previous version



Version control: save and commit locally

- Save and Commit,
allow saving the project
keeping track of changes
- Works "Local"
- Ideal for those who work alone
with no colleagues that collaborate on the project
- No more needs to make zip files named v1, v2...
- The history button show differences,
and allow restoring a previous version



Version control: save and commit locally

- Save and Commit,
allow saving the project
keeping track of changes
- Works "Local"
- Ideal for those who work alone
with no colleagues that collaborate on the project
- No more needs to make zip files named v1, v2...
- The history button show differences,
and allow restoring a previous version

The screenshot shows a software interface for managing project versions and configurations. On the left, there is a 'View history' section displaying two commits:

- v2 - adjustments to UI elements (checked, Thu Sep 22 09:33:47 2022 +0200 by LucBeg)
- v1 - Just some UI elements (checked, Thu Sep 22 09:24:35 2022 +0200 by LucBeg)

On the right, the 'MyProject' tree view shows the structure:

- MyProject
 - UI
 - MainWindow (type)
 - SpinBox1
 - var LeftMargin: 130 ▶ 260
 - var TopMargin: 78 ▶ 80
 - SpinBox2
 - var Value
 - DynamicLink: {NodId:ns=81;g=ee385914...}
 - SpinBox3

A 'Filter by' dropdown at the top right includes 'All' (selected) and 'Updates'. A 'Restore' button is located at the bottom right of the configuration pane.

Version control: save and commit locally

- Save and Commit,
allow saving the project
keeping track of changes
- Works "Local"
- Ideal for those who work alone
with no colleagues that collaborate on the project
- No more needs to make zip files named v1, v2...
- The history button show differences,
and allow restoring a previous version

The screenshot displays a software interface for version control and project management. On the left, a vertical timeline shows two commits: 'v2 - adjustments to UI elements' (unchecked) and 'v1 - Just some UI elements' (checked). The checked commit is highlighted in blue. On the right, the project structure 'MyProject' is shown, containing various components like BranchingEnabled, PasswordPolicy, Locales, LocaleFallbackList, UI, Model, Converters, Alarms, Recipes, Loggers, DataStores, Reports, OPC-UA, and CommDrivers. A warning message at the bottom states: '⚠️ Commits that occurred after a restored version will be removed permanently.' with a 'Restore' button.

View history

Filter by: All Updates ↓ ↑

v2 - adjustments to UI elements
Thu Sep 22 09:33:47 2022 +0200 by LucBeg

v1 - Just some UI elements
Thu Sep 22 09:24:35 2022 +0200 by LucBeg

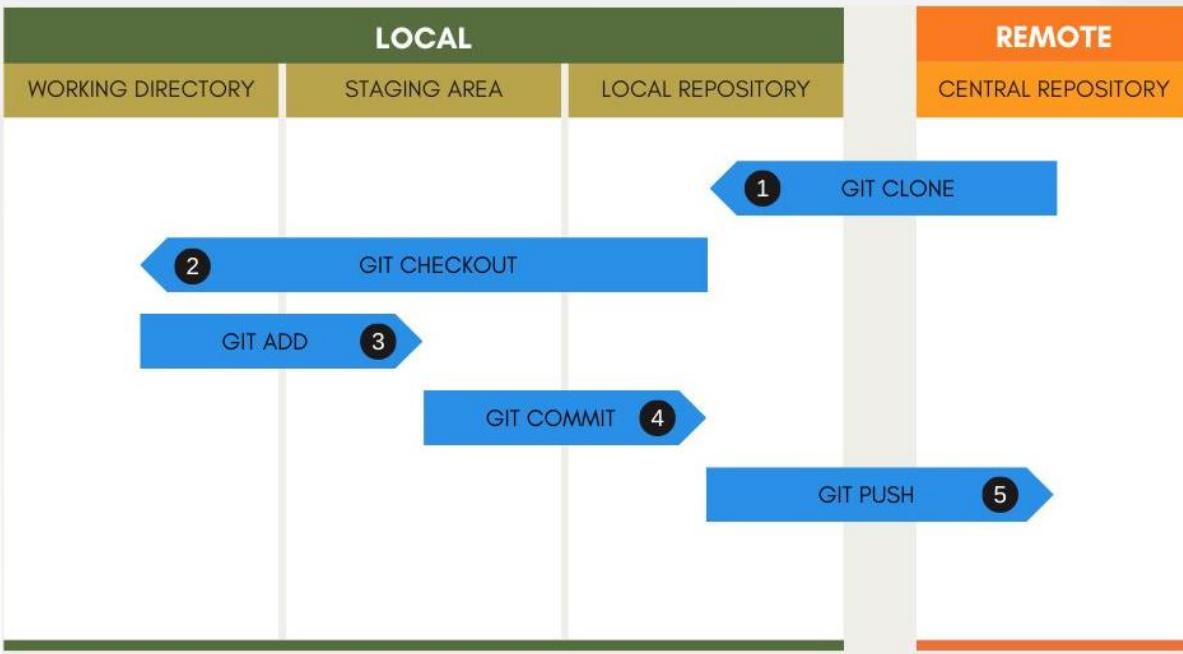
MyProject

- var BranchingEnabled 0
- >PasswordPolicy
- var Locales {en-US}
- var LocaleFallbackList {en-US}
- UI
- Model
- Converters
- Alarms
- Recipes
- Loggers
- DataStores
- Reports
- OPC-UA
- CommDrivers

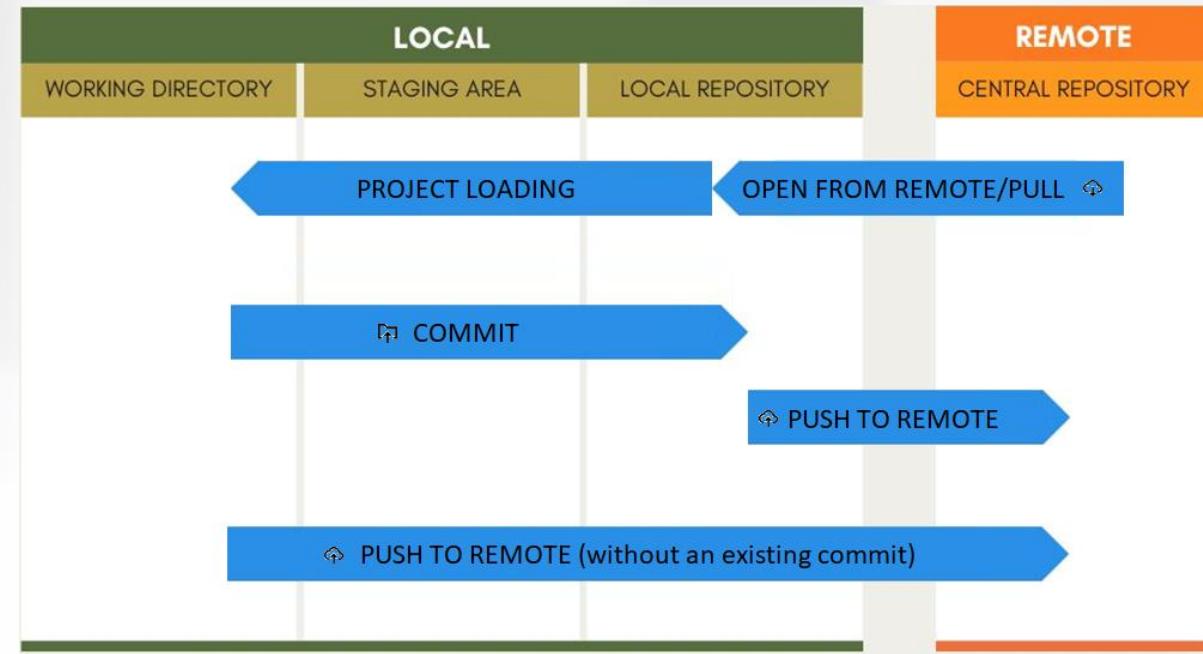
⚠️ Commits that occurred after a restored version will be removed permanently. Restore

GIT workflow vs. Optix workflow

GIT Workflow



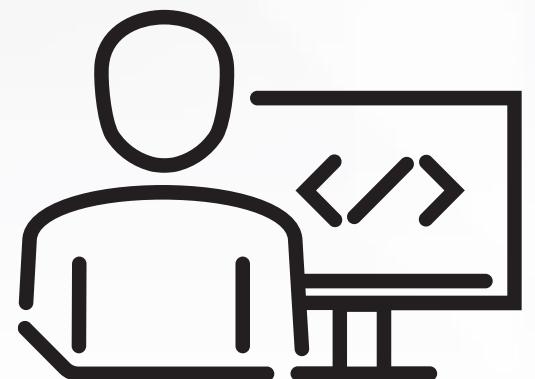
Optix Workflow



- Optix «hides» the GIT commands to improve the user experience
- User does not need to know anything about GIT commands and workflow

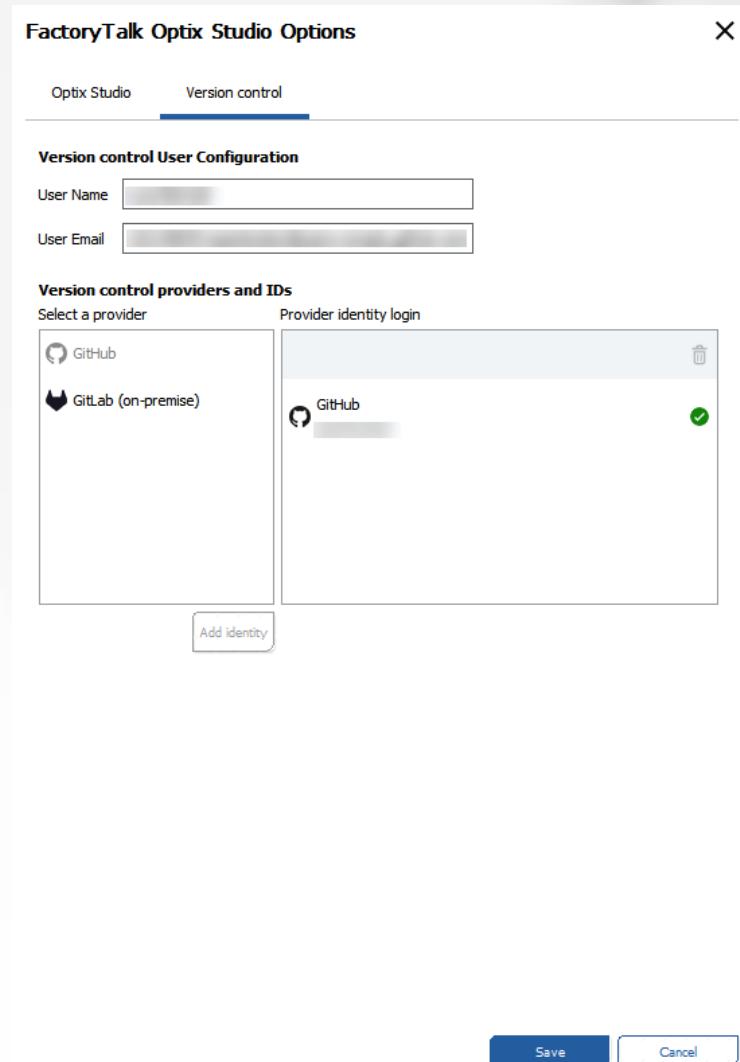
Hands-on session

- Execute some Save and commit
- Explore the version history
- Try to restore to a previous state



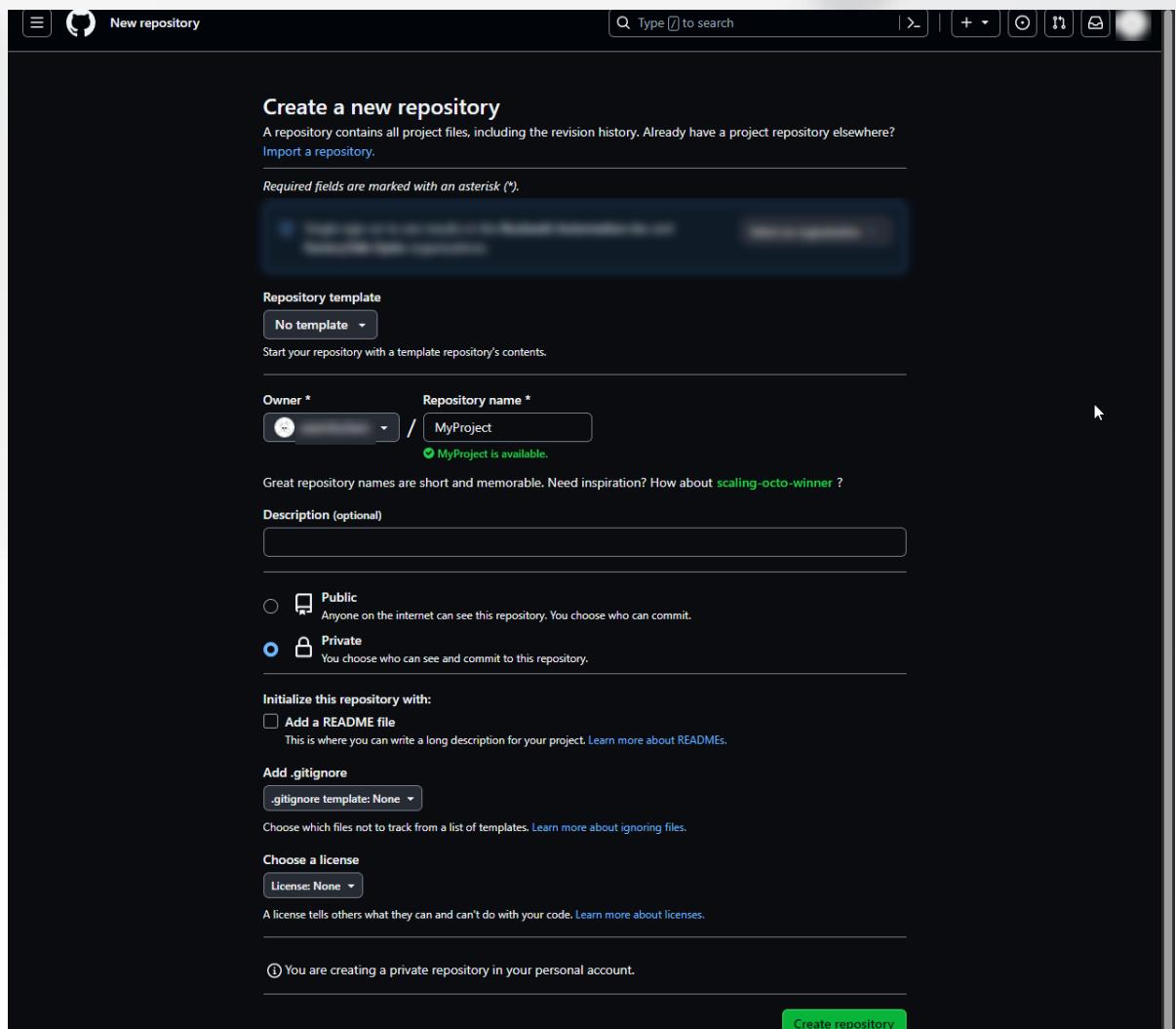
Collaboration: save and commit on GitHub

- Configure/verify the GitHub access from Studio options
- Access GitHub and create a Repository
 - repository name **MUST** have the same name of Optix project
 - advice: set the repository to be Private, then share with colleagues
- Copy the repository URL



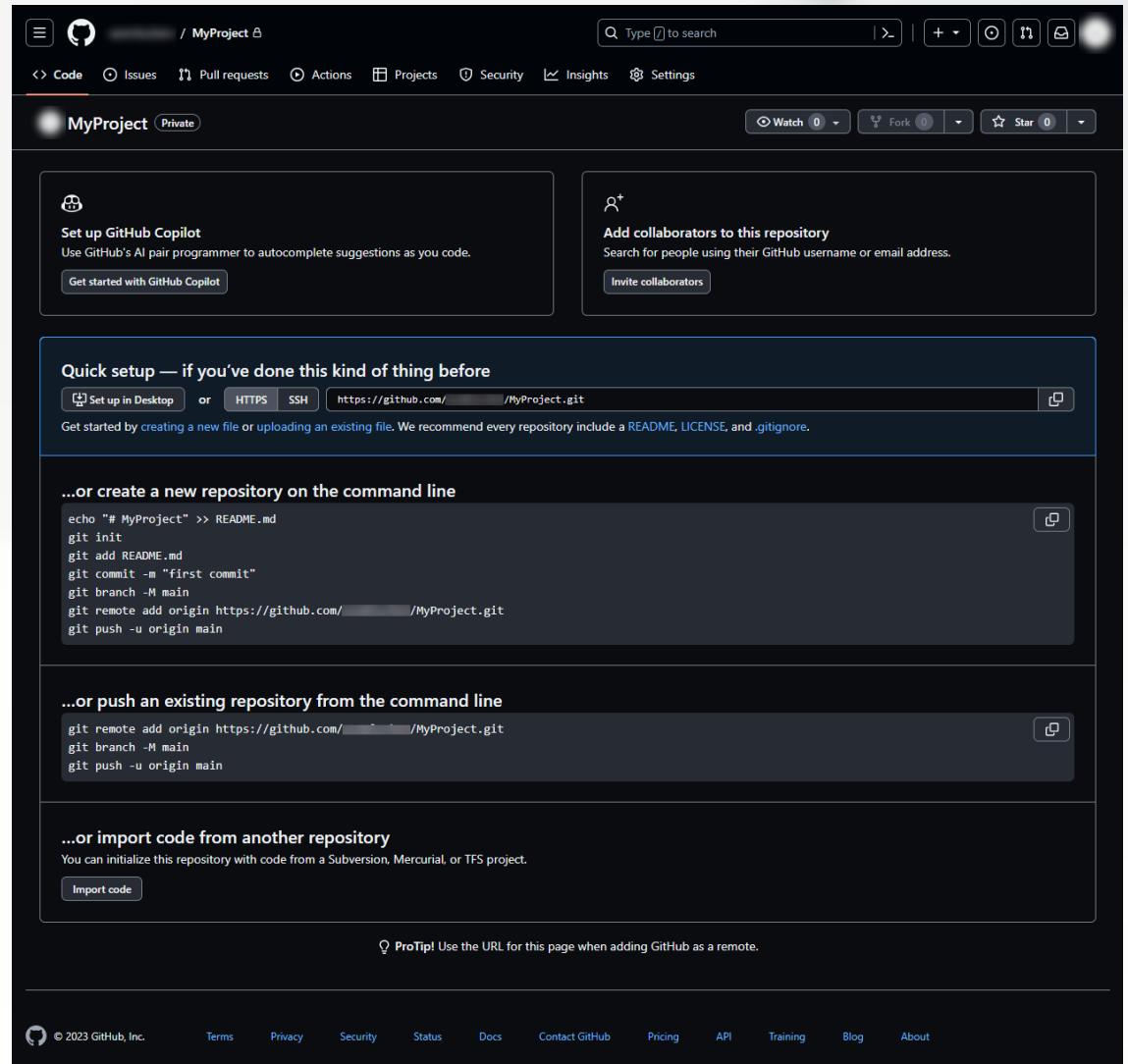
Collaboration: save and commit on GitHub

- Configure/verify the GitHub access from Studio options
- Access GitHub and create a Repository
 - repository name **MUST** have the same name of Optix project
 - advice: set the repository to be Private, then share with colleagues
- Copy the repository URL
- Do not tick anything under “Initialize this repository with”!



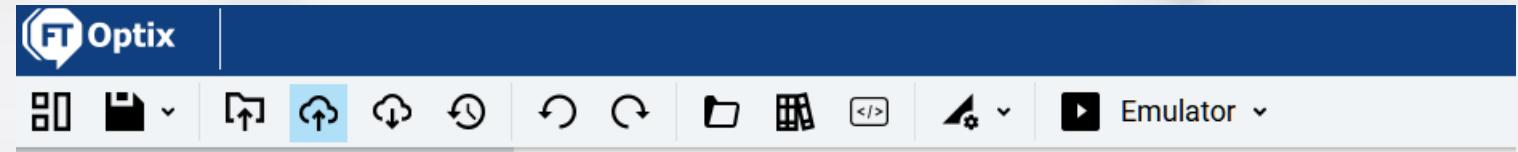
Collaboration: save and commit on GitHub

- Configure/verify the GitHub access from Studio options
- Access GitHub and create a Repository
 - repository name **MUST** have the same name of Optix project
 - advice: set the repository to be Private, then share with colleagues
- Copy the repository URL



Collaboration: save and commit on GitHub

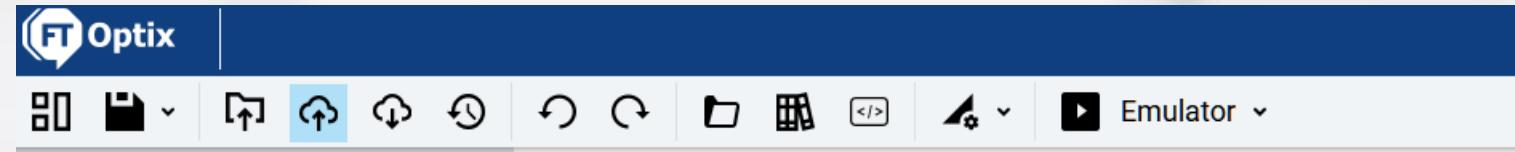
- Click on
the "Push to remote" icon



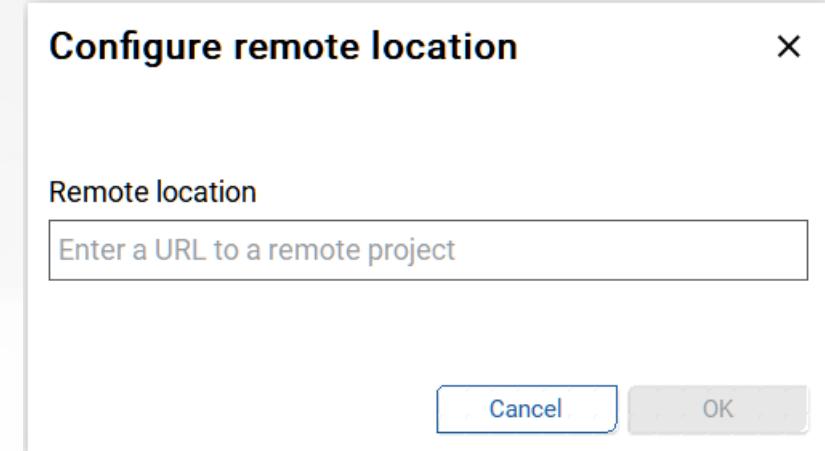
- Paste the repository URL
- Then "Project is synchronized"
- From now on, doing a "Save and commit",
the Studio warns that local changes
are available to push to GitHub

Collaboration: save and commit on GitHub

- Click on
the "Push to remote" icon



- Paste the repository URL

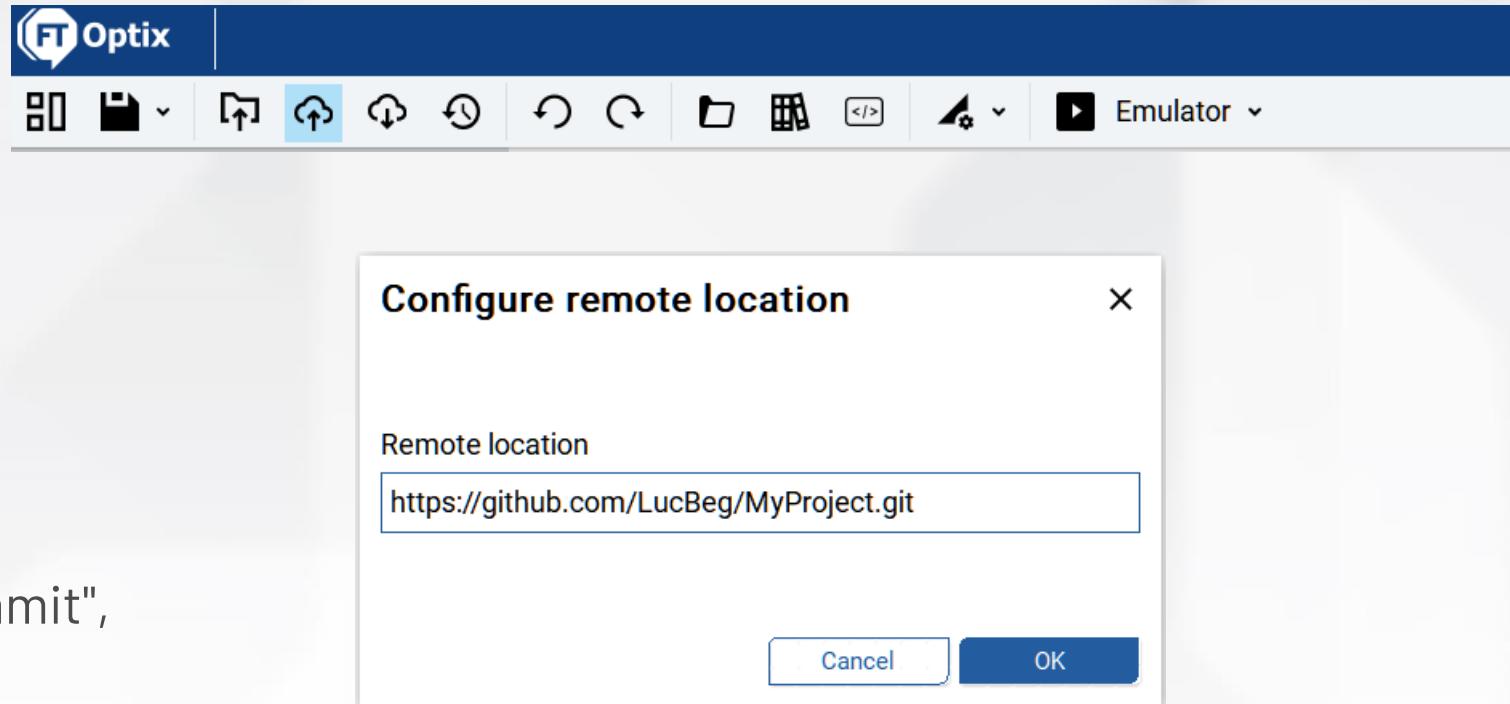


- Then "Project is synchronized"

- From now on, doing a "Save and commit",
the Studio warns that local changes
are available to push to GitHub

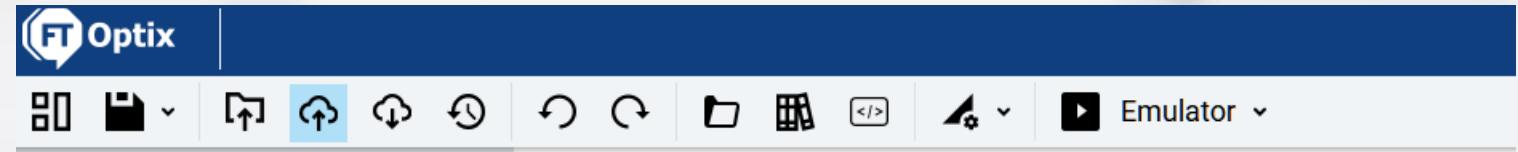
Collaboration: save and commit on GitHub

- Click on
the "Push to remote" icon
- Paste the repository URL
- Then "Project is synchronized"
- From now on, doing a "Save and commit",
the Studio warns that local changes
are available to push to GitHub



Collaboration: save and commit on GitHub

- Click on
the "Push to remote" icon



- Paste the repository URL

- Then "Project is synchronized"

- From now on, doing a "Save and commit",
the Studio warns that local changes
are available to push to GitHub



Collaboration: save and commit on GitHub

- Click on
the "Push to remote" icon



- Paste the repository URL

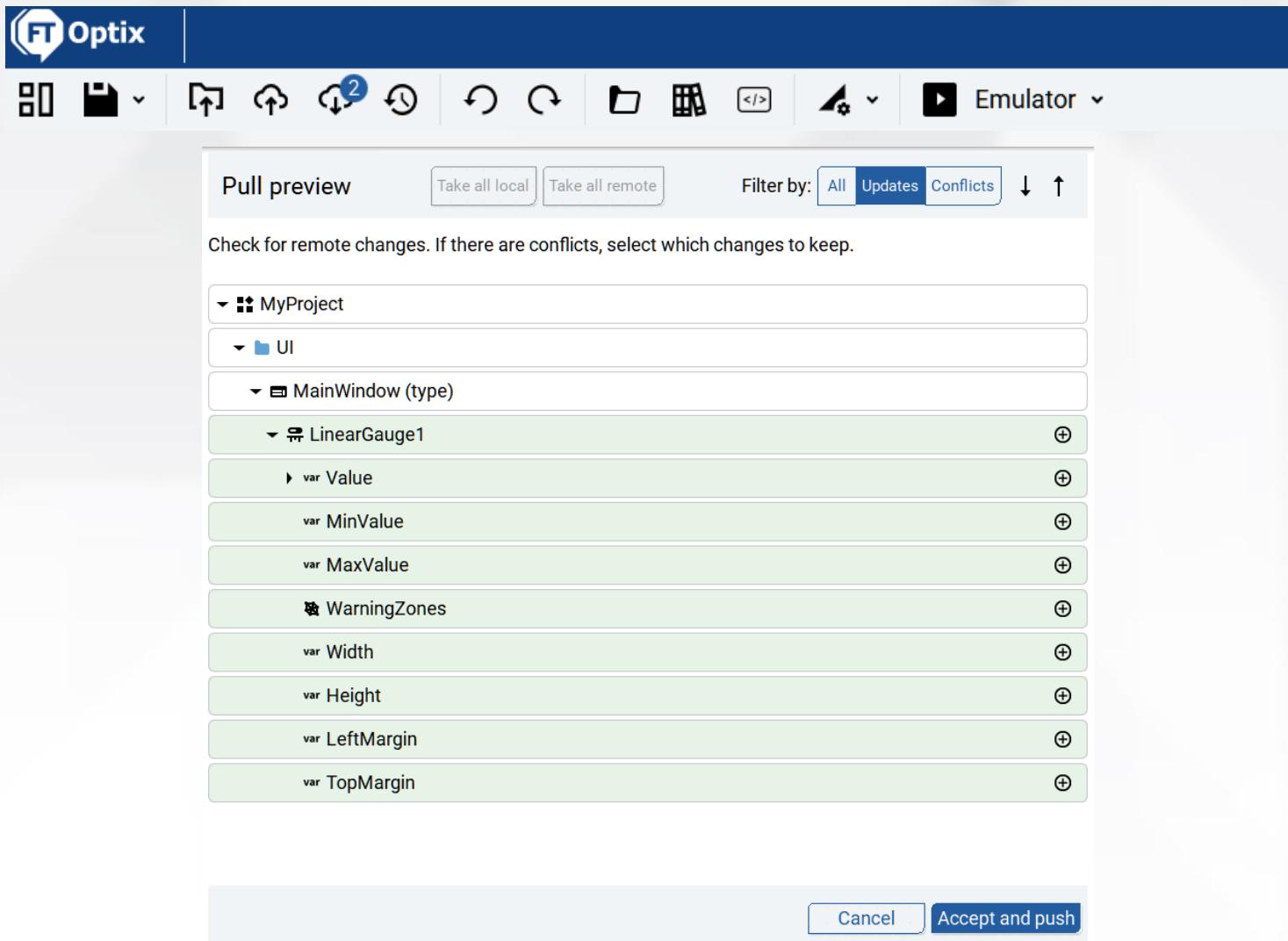
- Then "Project is synchronized"

- From now on, doing a "Save and commit",
the Studio warns that local changes
are available to push to GitHub



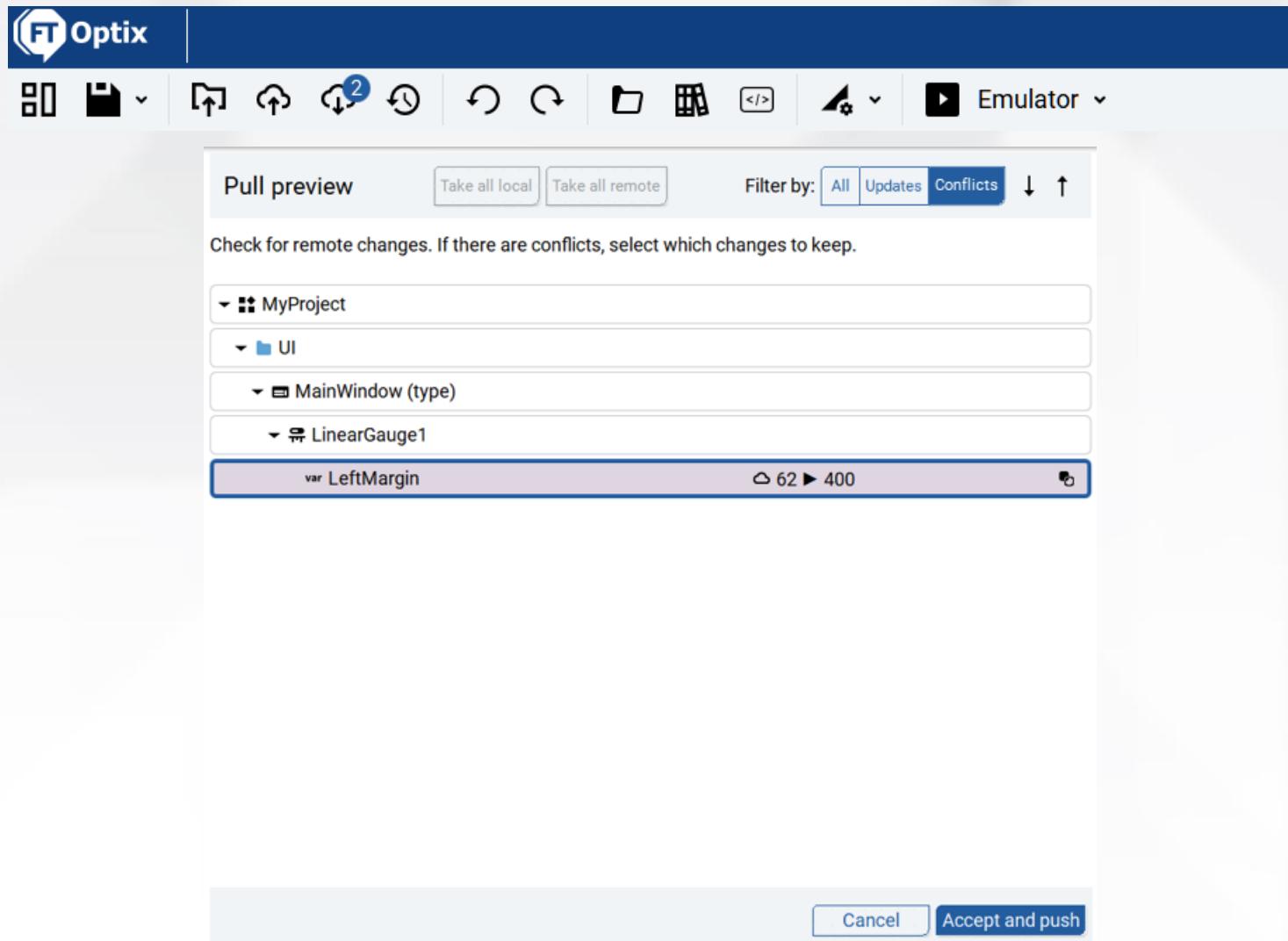
Collaboration: pull, conflicts, merge

- When someone else pushes changes on the remote repository, other users get notified (within a few minutes)
- In case of conflicts, can be decided what to take local or remote or accept all and merge



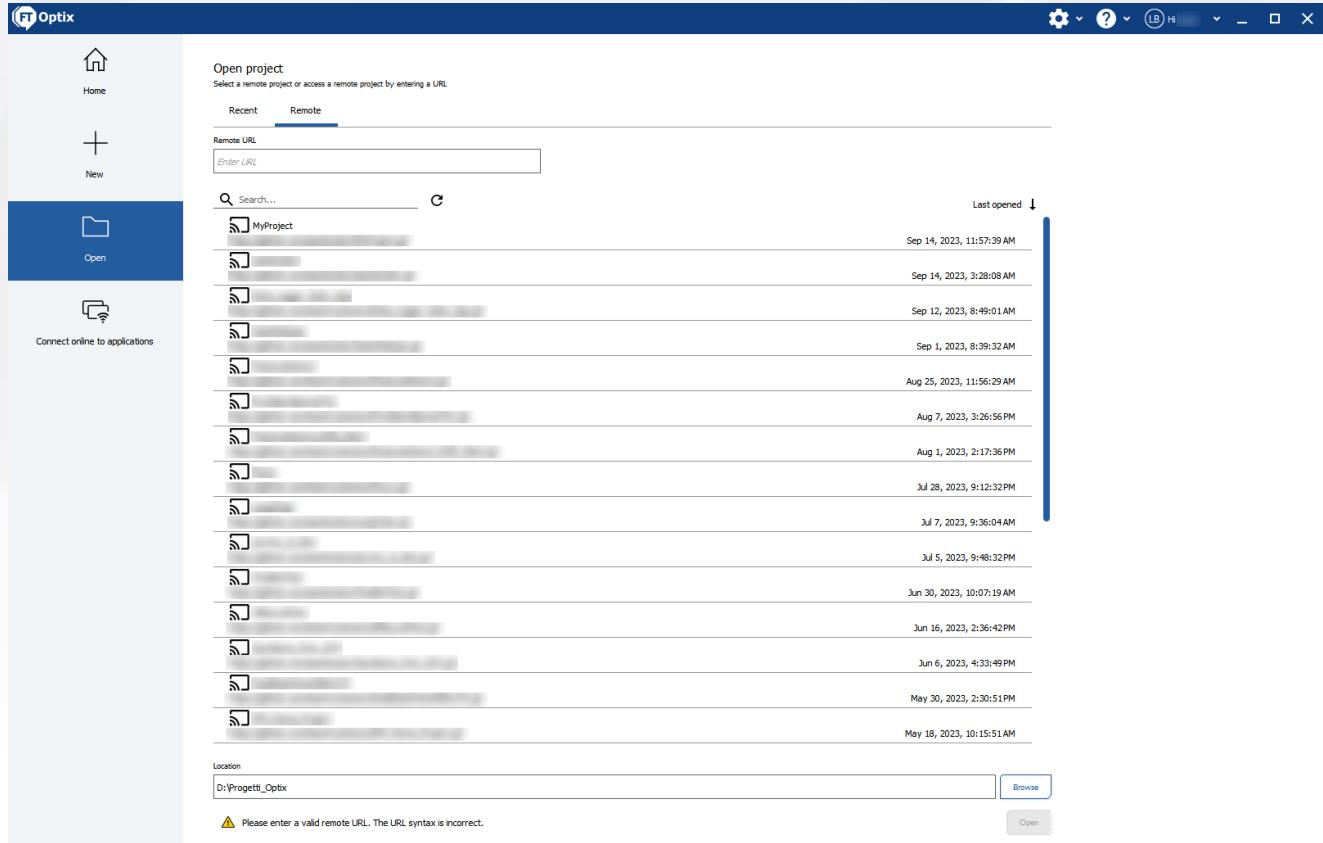
Collaboration: pull, conflicts, merge

- When someone else pushes changes on the remote repository, other users get notified (within a few minutes)
- In case of conflicts, can be decided what to take local or remote or accept all and merge



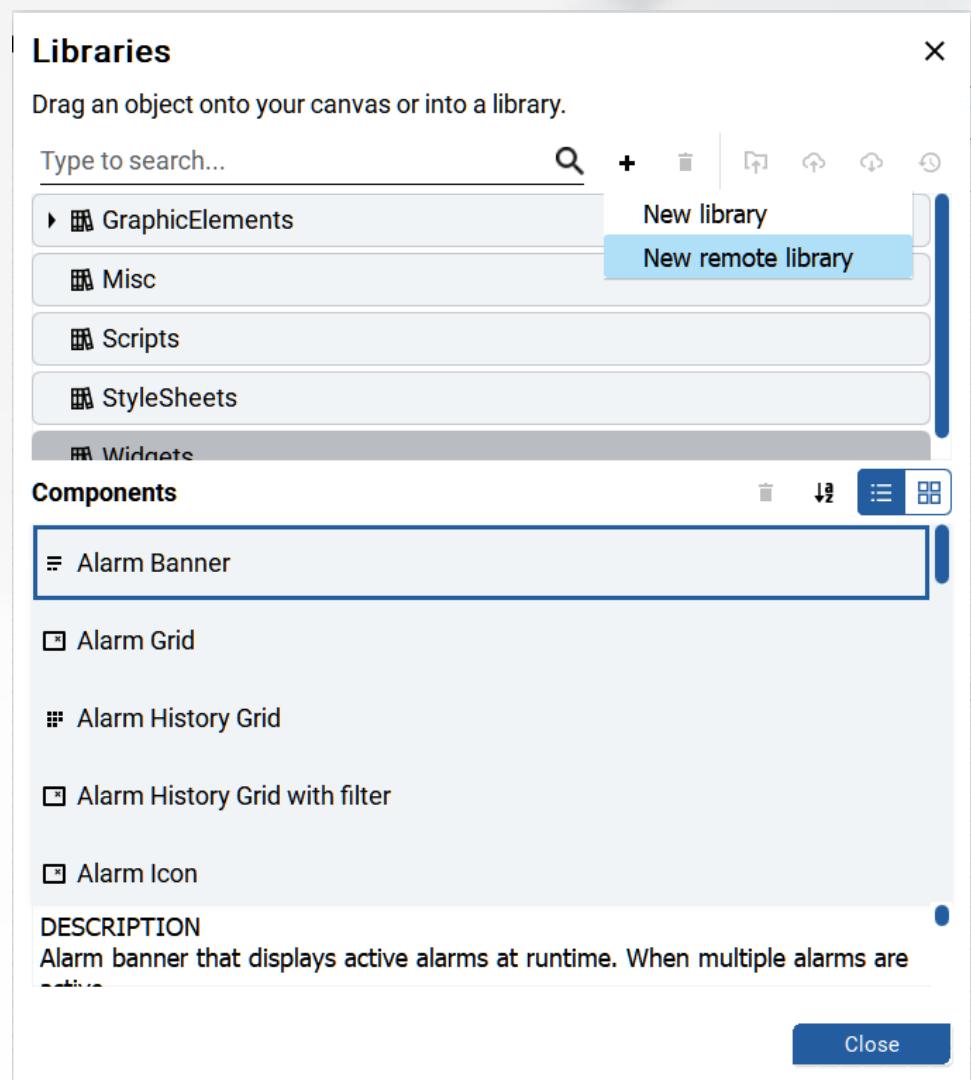
Collaboration: opening a remote project

- A new project developer, that does not have the project sources, can open the project from a Remote
- The project will be downloaded locally and from now on can start to collaborate with save and commit, push and pull



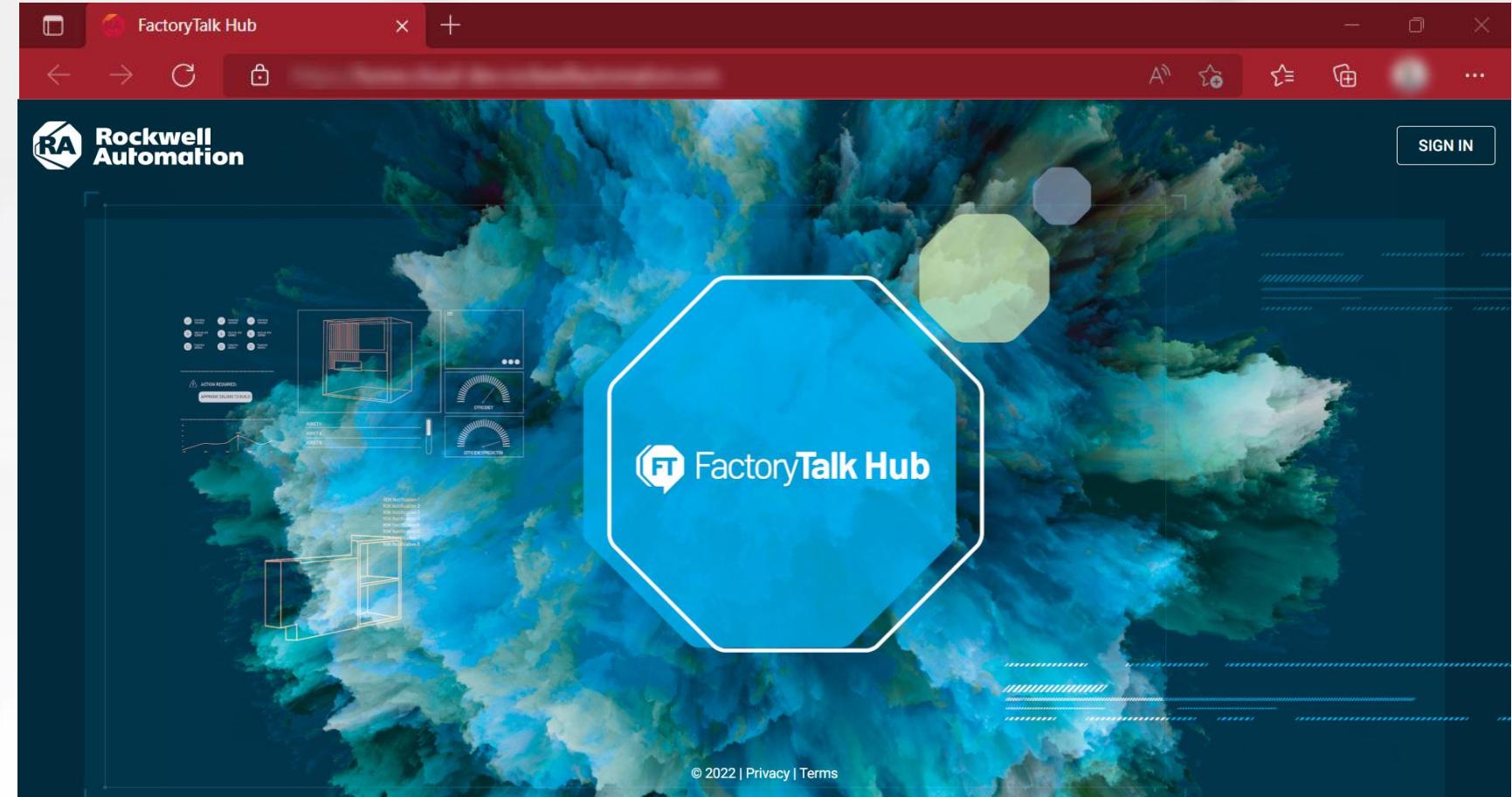
Shared libraries

- a Custom Library can be shared as per the project
- a Shared Library is useful when the company has built his own graphical or widget library in the way all projects reuse the same basic components



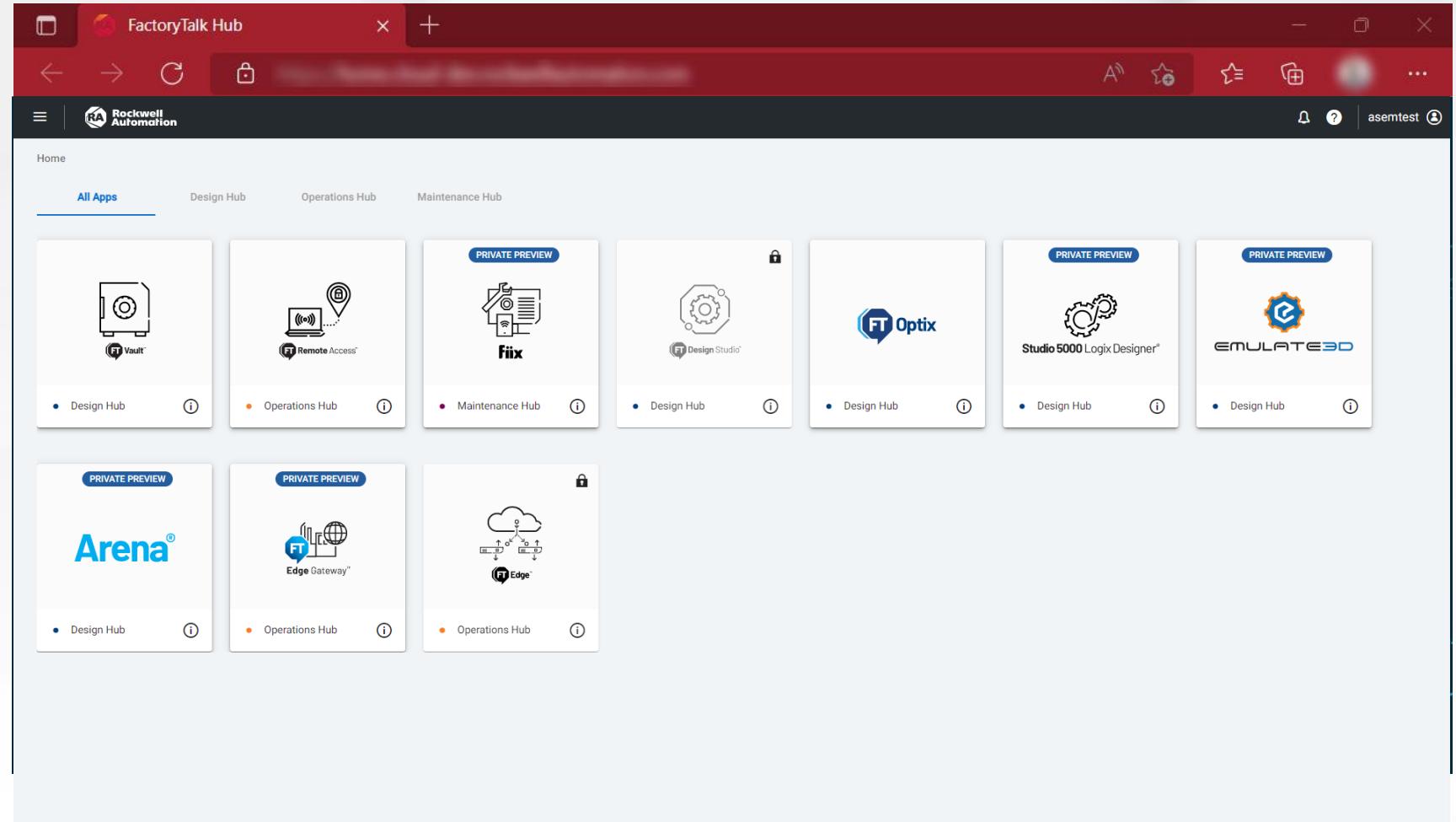
Web IDE using FT Hub

- Sign in to FactoryTalk Hub
- Click on the FTOptix tile
- Select which build to launch on Web IDE or continue with the last used version



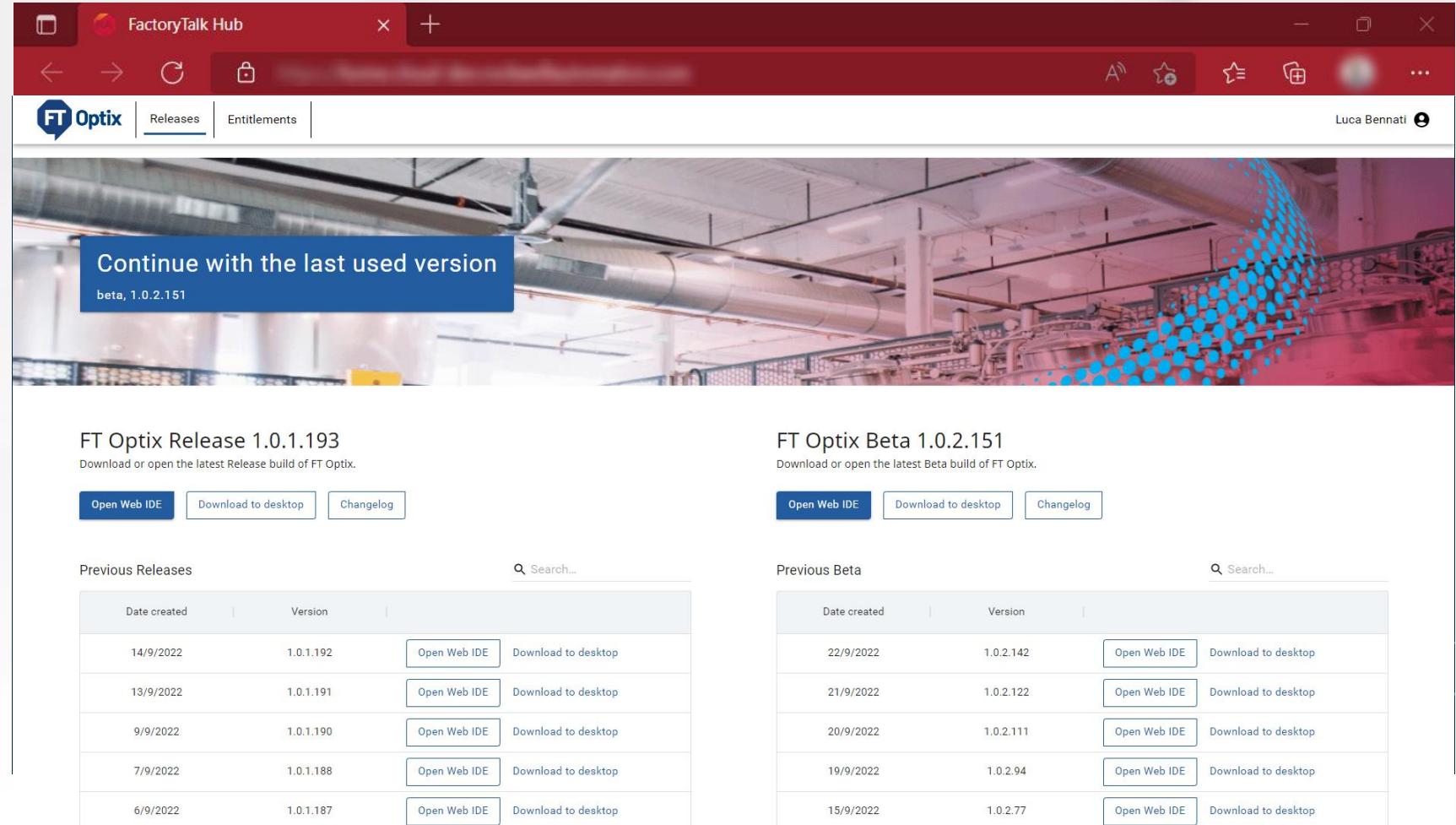
Web IDE using FT Hub

- Sign in to FactoryTalk Hub
- Click on the FTOptix tile
- Select which build to launch on Web IDE or continue with the last used version



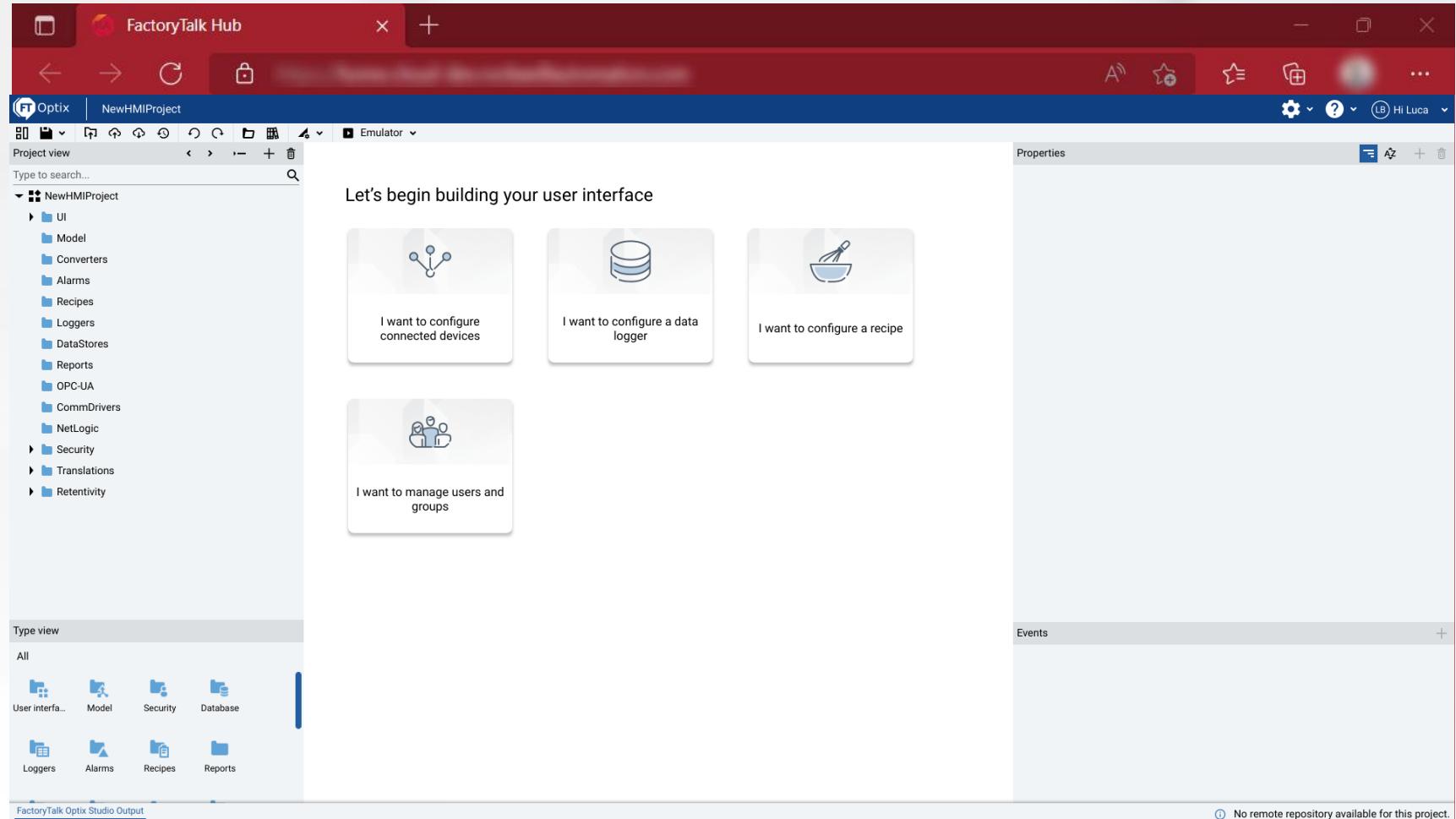
Web IDE using FT Hub

- Sign in to FactoryTalk Hub
- Click on the FTOptix tile
- Select which build to launch on Web IDE or continue with the last used version



Web IDE using FT Hub

- Sign in to FactoryTalk Hub
- Click on the FTOptix tile
- Select which build to launch on Web IDE or continue with the last used version





Transfer and execute
the project



Project compatibility and backup

- **Upgrade:**

- Projects created with an old version of Studio **can be opened with later versions**
- Studio automatically updates the project and all its modules to the current version

- **Downgrade:**

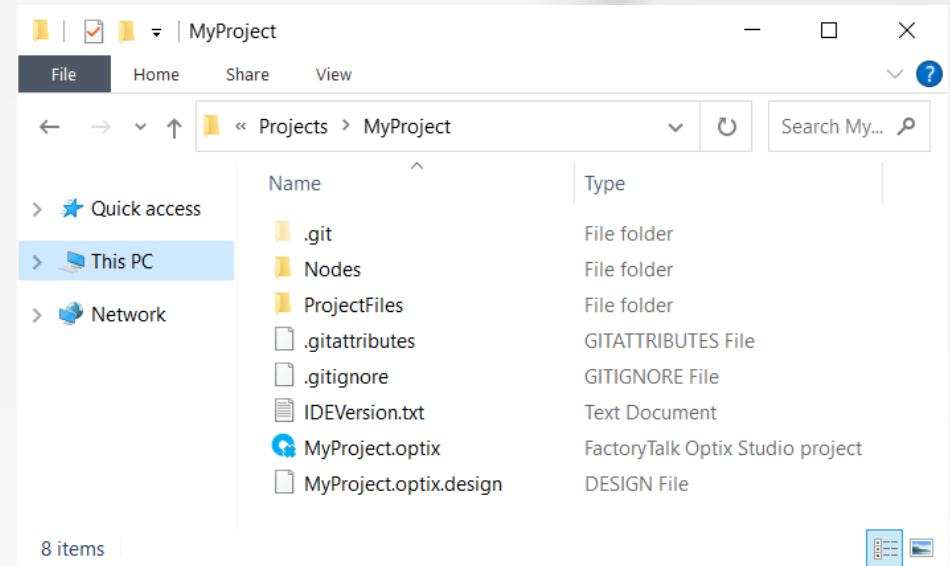
- Projects created with newer versions of Studio **cannot be opened with older versions**

- **Backups:** %localappdata%\Rockwell Automation\FactoryTalk Optix\FTOptixStudio\Backups\

- Automatic: contain the last three backups, performed when saving the project
- Upgrade: performed when updating the project to make it compatible with a newer version
- Recovery: performed when a corrupt project is opened, and repair is attempted

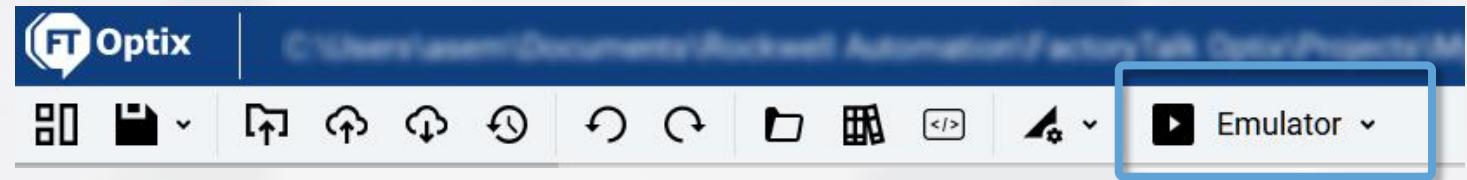
Project structure

- Contents of the project folder
 - **.optix** file: information on the project's elements
 - **.optix.design** file: information on accessory elements only required at design time.
 - **Nodes** folder: contains a YAML file for each node that describes its related information model
 - **ProjectFiles** folder: containing all files used in the project, including images, fonts and documents.
 - NetSolution subfolder contains all files used to create customized C# scripts
 - PKI subfolder: Public Key Infrastructure folder, used to manage security certificates for OPC-UA



Execute the project with emulator

- Allow the execution of the project from Studio.
- Emulator folder: %localappdata%\Rockwell Automation\FactoryTalk Optix\Emulator\
- Emulator output is useful to check license token in use by the Runtime



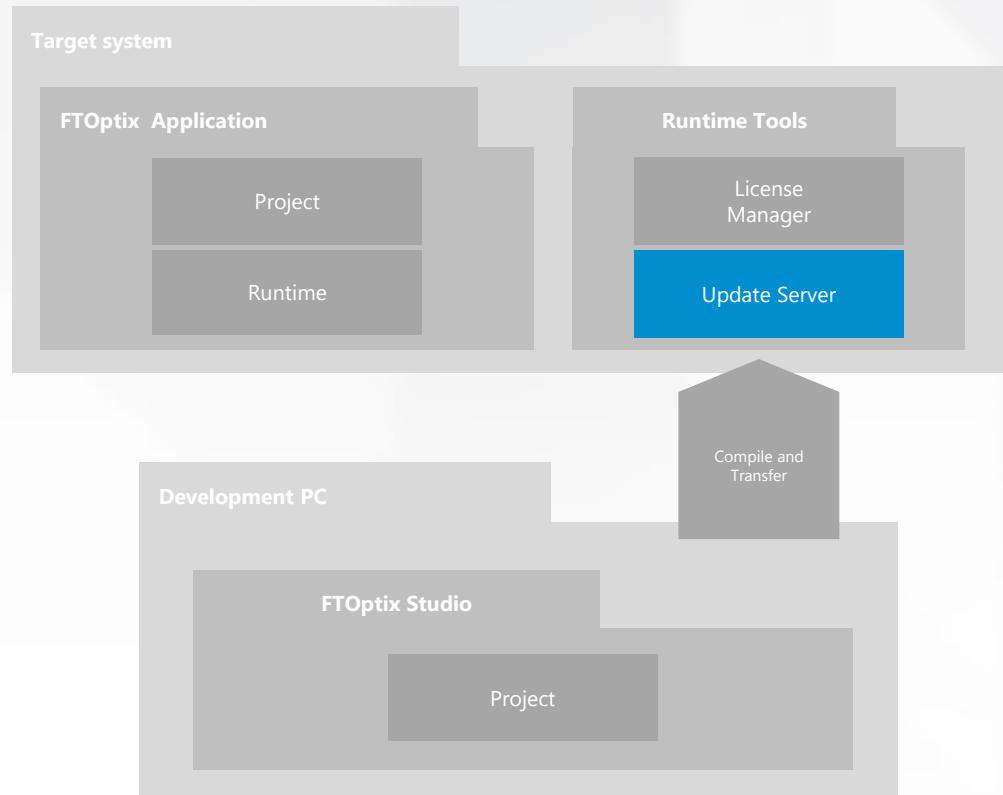
Code	Timestamp	Category	Object	Message
①	2022-09-27 18:19:56.572	FTOptixRuntime		Starting project MyProject
①	2022-09-27 18:19:56.572	FTOptixRuntime		FTOptixRuntime is currently using 3 license token(s) ↴ Component: Native pres... ↴ Component: Retentivity storage consu...
①	2022-09-27 18:19:56.572	FTOptixRuntime		No license tokens found ↴ FTOptixRuntime will be closed in: 120 minutes

Transfer the project to local target

1. Target must have "Runtime Tools" installed and "Update Server" running

- **Optix Panel:** Runtime Tools are already installed in production and Update the Server running as a service

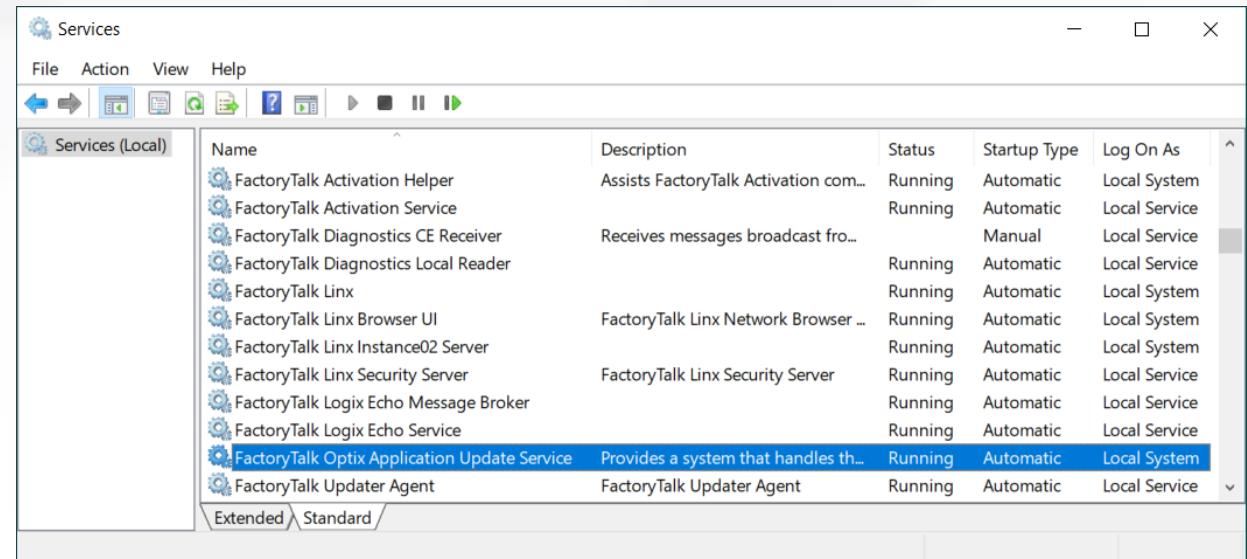
- **iPC:** Runtime Tools must be installed manually. Update Server will be installed as Service and set to automatically startup when Windows starts



Transfer the project to local target

1. Target must have "Runtime Tools" installed and "Update Server" running

- **Optix Panel:** Runtime Tools are already installed in production and Update the Server running as a service
- **iPC:** Runtime Tools must be installed manually. Update Server will be installed as Service and set to automatically startup when Windows starts



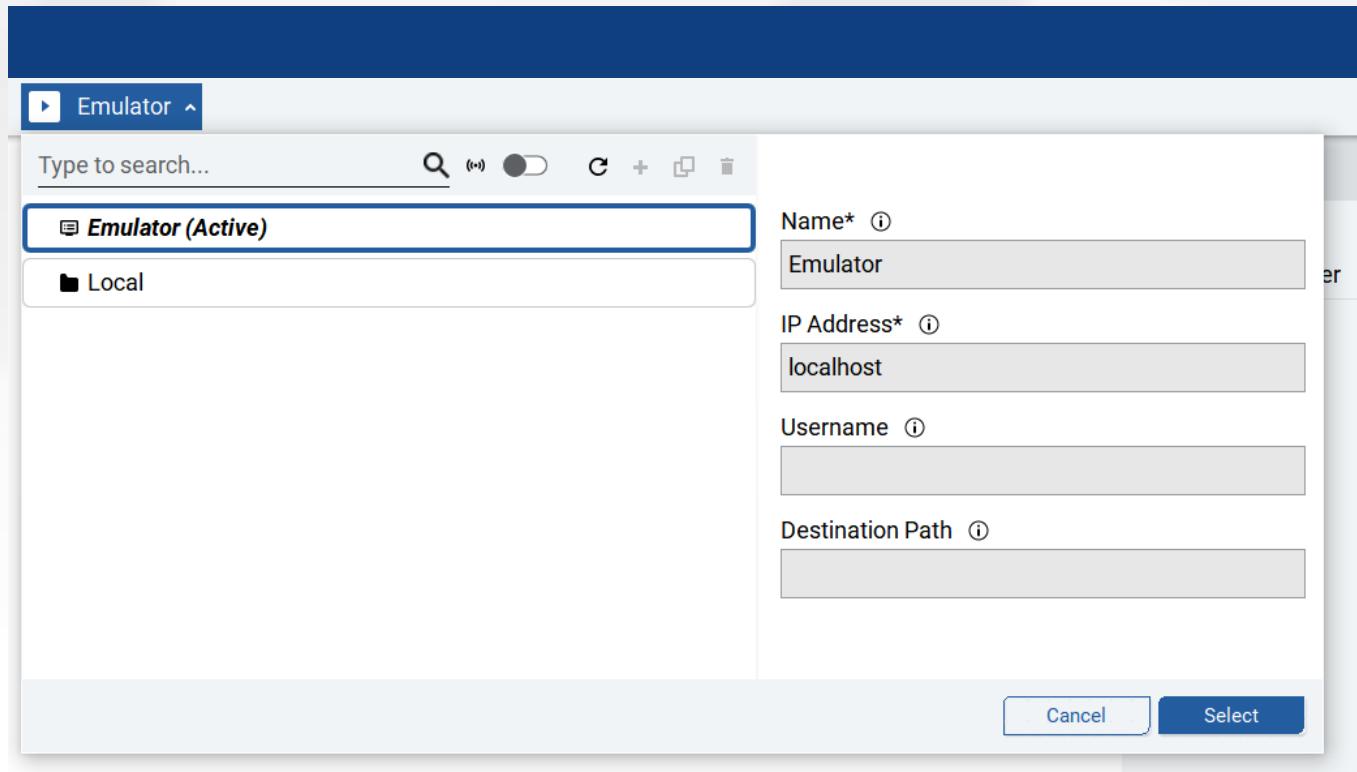
The screenshot shows the Windows Services snap-in window. The title bar reads "Services". The menu bar includes File, Action, View, and Help. Below the menu is a toolbar with various icons. The main area is divided into two panes: "Services (Local)" on the left and a list of services on the right. The list includes the following services:

Name	Description	Status	Startup Type	Log On As
FactoryTalk Activation Helper	Assists FactoryTalk Activation com...	Running	Automatic	Local System
FactoryTalk Activation Service		Running	Automatic	Local Service
FactoryTalk Diagnostics CE Receiver	Receives messages broadcast fro...	Running	Manual	Local Service
FactoryTalk Diagnostics Local Reader		Running	Automatic	Local Service
FactoryTalk Linx		Running	Automatic	Local System
FactoryTalk Linx Browser UI	FactoryTalk Linx Network Brower ...	Running	Automatic	Local System
FactoryTalk Linx Instance02 Server		Running	Automatic	Local System
FactoryTalk Linx Security Server	FactoryTalk Linx Security Server	Running	Automatic	Local Service
FactoryTalk Logix Echo Message Broker		Running	Automatic	Local Service
FactoryTalk Logix Echo Service		Running	Automatic	Local Service
FactoryTalk Optix Application Update Service	Provides a system that handles th...	Running	Automatic	Local System
FactoryTalk Updater Agent	FactoryTalk Updater Agent	Running	Automatic	Local Service

Transfer the project to local target

2. Configure Target device in Studio

- Name,
- IP Address,
- Username,
 - for Optix Panels, it's the same admin user that access the System Manager
 - for iPC, it's a Windows user with rights to access OS folders
- Destination Path: optional



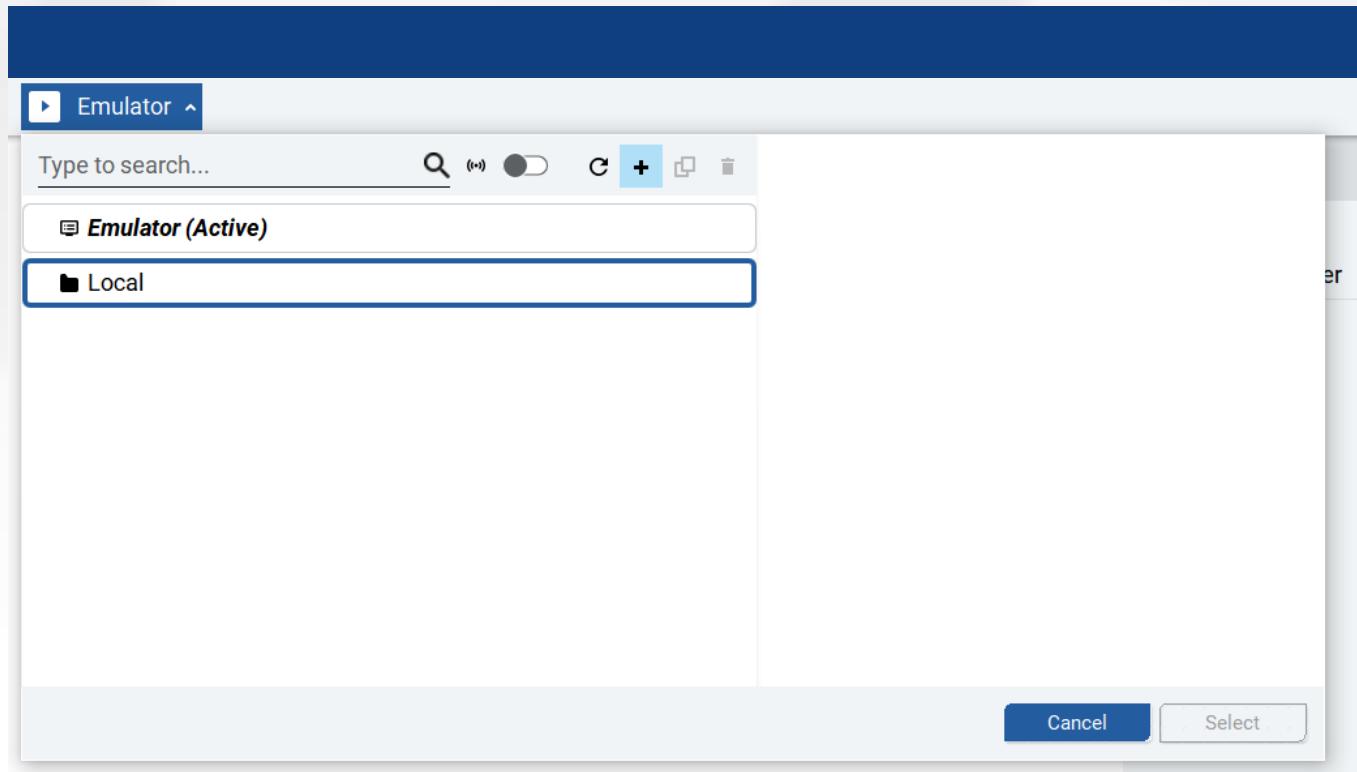
3. Transfer from Studio to Target

- set the Active path
- click the Play button

Transfer the project to local target

2. Configure Target device in Studio

- Name,
- IP Address,
- Username,
 - for Optix Panels, it's the same admin user that access the System Manager
 - for iPC, it's a Windows user with rights to access OS folders
- Destination Path: optional



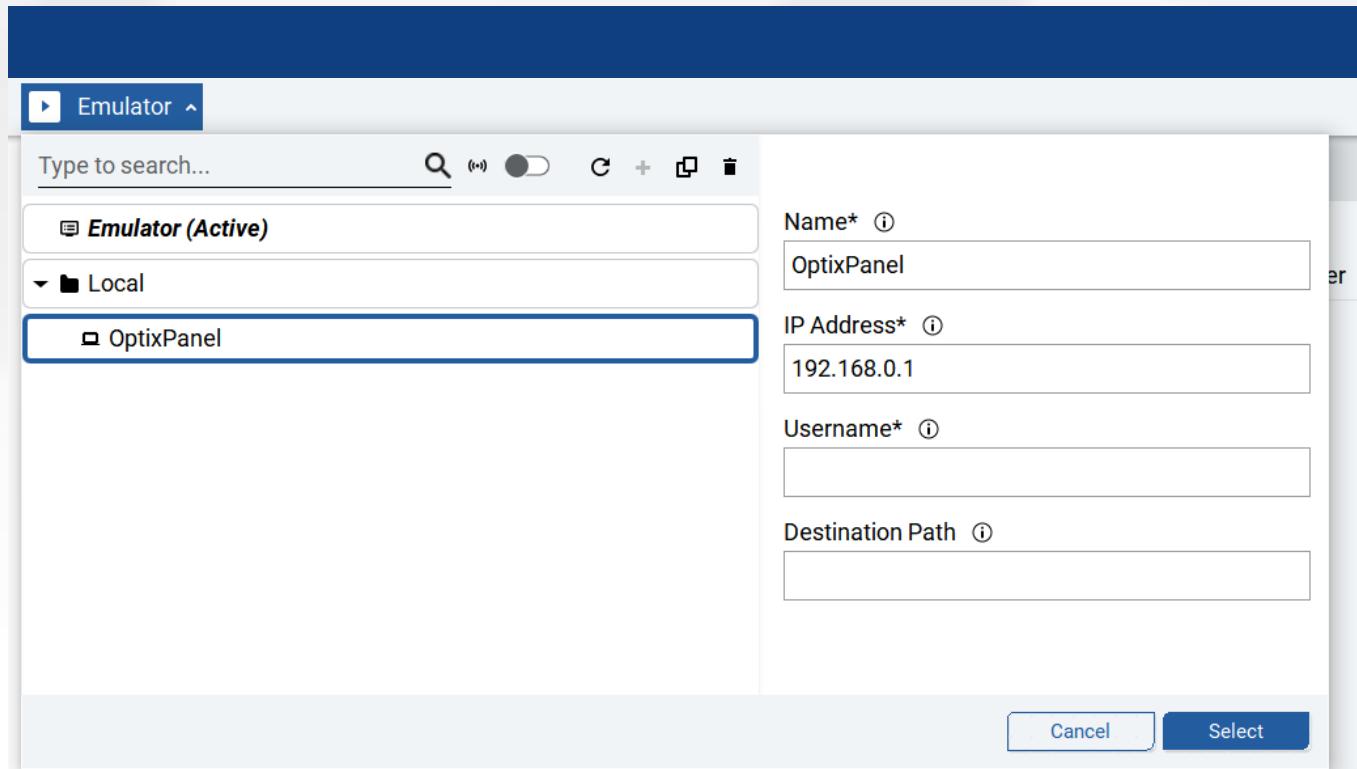
3. Transfer from Studio to Target

- set the Active path
- click the Play button

Transfer the project to local target

2. Configure Target device in Studio

- Name,
- IP Address,
- Username,
 - for Optix Panels, it's the same admin user that access the System Manager
 - for iPC, it's a Windows user with rights to access OS folders
- Destination Path: optional



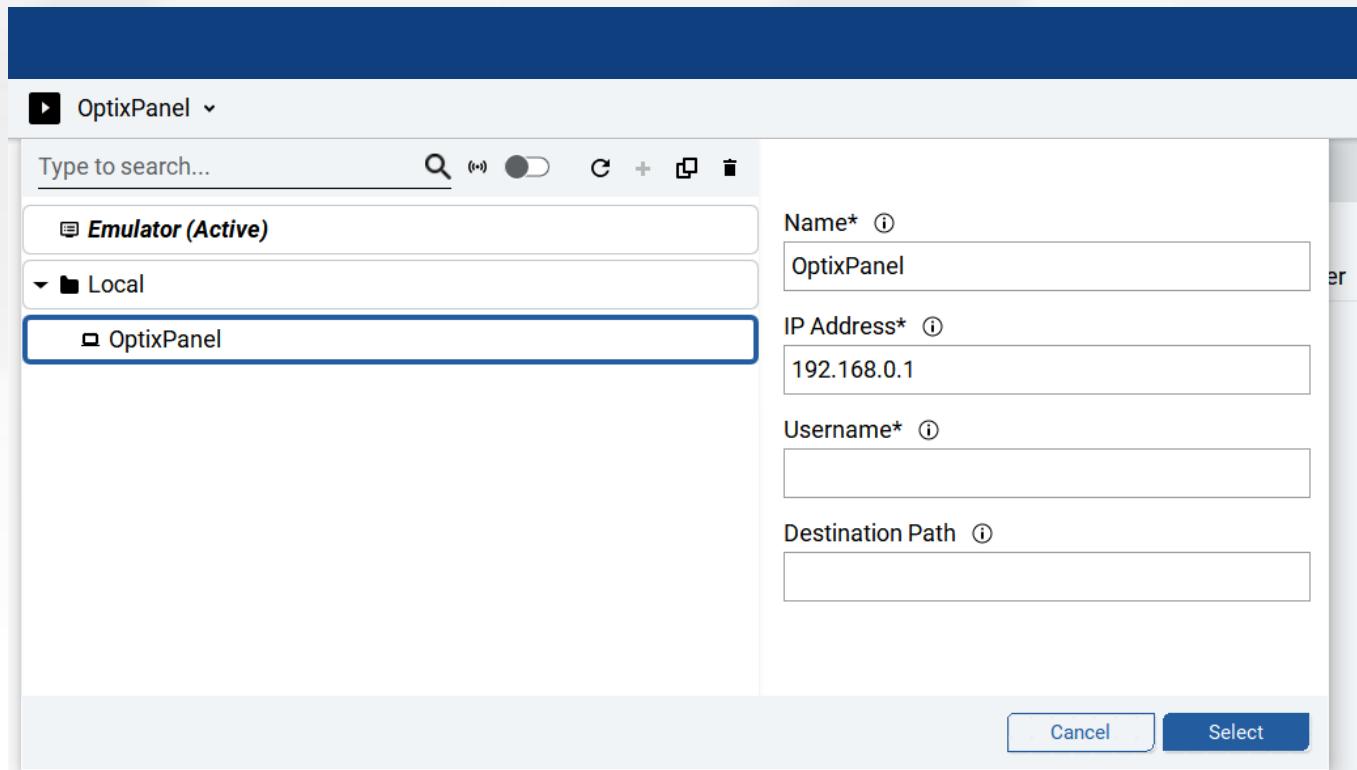
3. Transfer from Studio to Target

- set the Active path
- click the Play button

Transfer the project to local target

2. Configure Target device in Studio

- Name,
- IP Address,
- Username,
 - for Optix Panels, it's the same admin user that access the System Manager
 - for iPC, it's a Windows user with rights to access OS folders
- Destination Path: optional

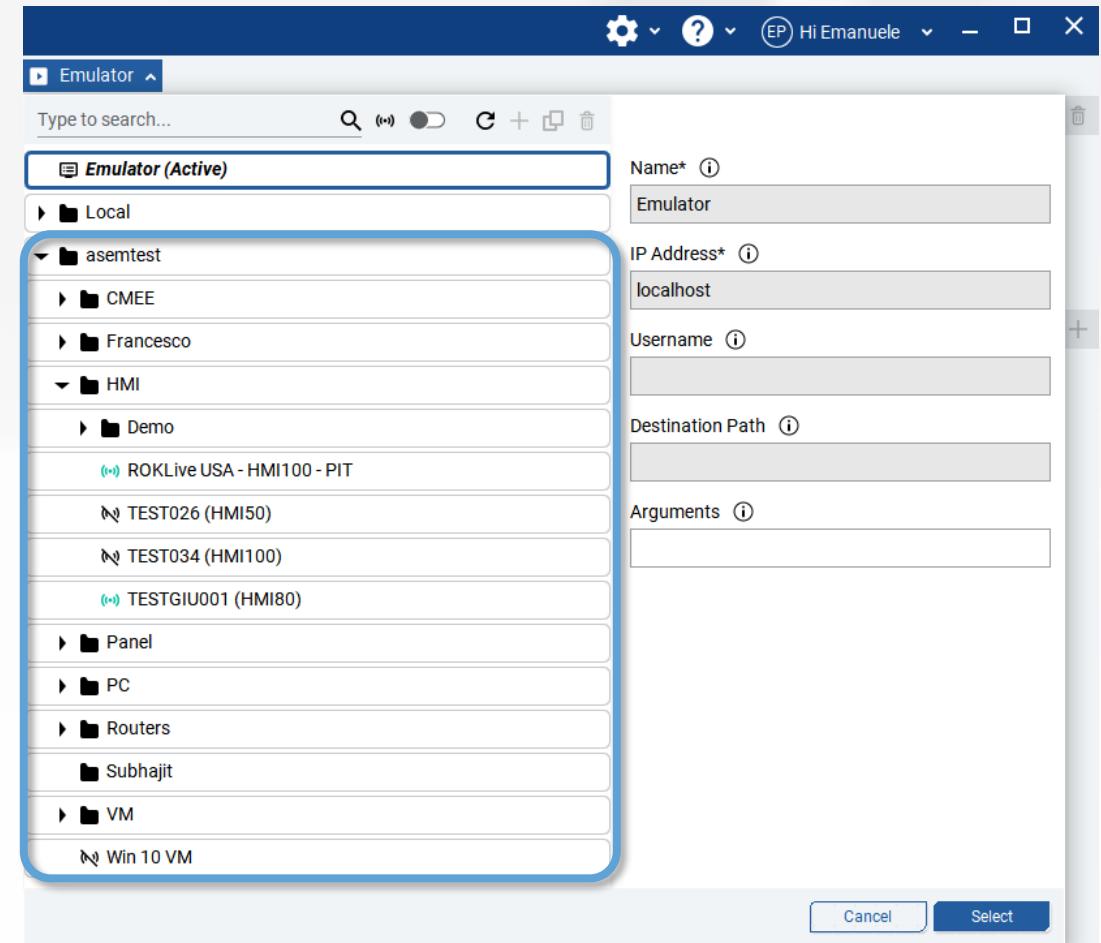


3. Transfer from Studio to Target

- set the Active path
- click the Play button

Transfer the project to remote target (via FT Remote Access)

- If you are signed in FT Hub and you have also access to a FactoryTalk Remote Access domain
- FT Optix Studio, show you the devices available through FT RA
- Then you will be able to:
 - activate a VPN connection to the device and transfer the project
 - open the FTRA screen mirroring feature (VNC-like)



Execute the project on target

4. Application is launched by Studio after download

- **HMI:** Application is also set to be executed automatically at HMI startup
- **iPC:** Application is also set to be automatically executed at Windows startup by using the shortcut of

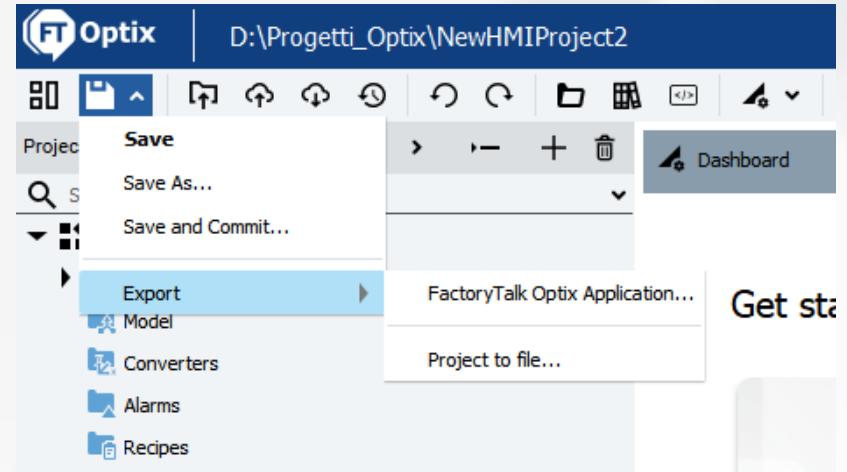
```
%localappdata%\Rockwell Automation\FactoryTalk  
Optix\Application\FTOptixRuntime.exe
```

into:

```
shell:startup
```

Transfer project to target manually

- Application can be transferred manually
 - "**Export FactoryTalk Optix Application**" : exports the (semi) compiled Application into a folder that needs to be manually transferred and executed on Target



Platform	HMI Model
Optix Panel	Optix Panel Standard or Compact
Windows 64-bit	iPC Windows
Ubuntu 64-bit	iPC Linux Debian

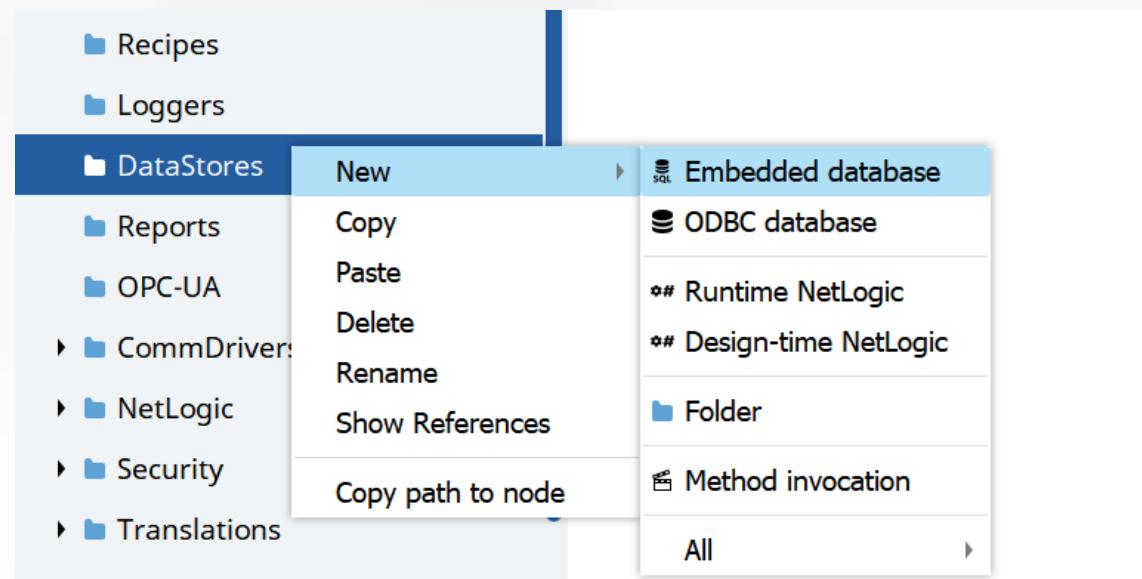


Save historical data
with loggers



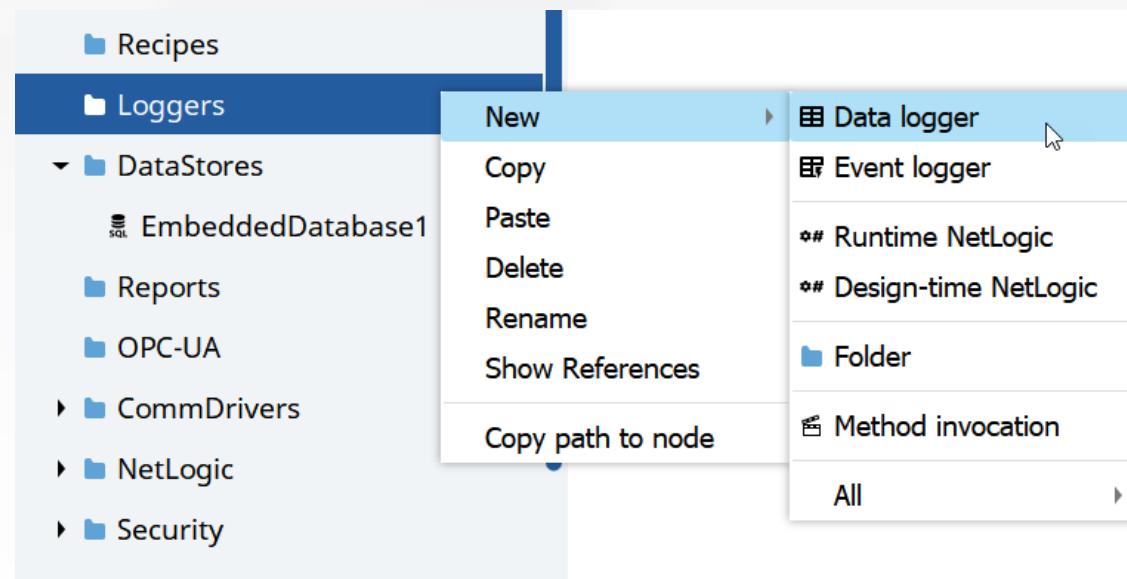
Loggers

- A logger is an object that records data and stores them in a database
- Types of logger
 - **Data logger:** Records values of one or more variables at the same time
 - **Event logger:** Records the properties of one or more events



Loggers

- A logger is an object that records data and stores them in a database
- Types of logger
 - **Data logger:** Records values of one or more variables at the same time
 - **Event logger:** Records the properties of one or more events



Data logger

The screenshot shows the Rockwell Studio interface. On the left, the project tree is visible under 'MyProject': UI, Model, Converters, Alarms, Recipes, Loggers, DataStores, Reports, OPC-UA, CommDrivers, ModbusDriver1, ModbusStation1, Tags (with three checked checkboxes for ModbusTag1, ModbusTag2, and ModbusTag3), and Types. At the bottom, there's a 'NetLogic' folder. On the right, a configuration window for 'DataLogger1' is open, showing a tree structure with 'Loggers' selected.

- At every sampling event the Datalogger create a new record into a table with the following structure:

Date/Time	Variable 1	Variable 2	...	Variable N
hh:mm:ss	value	value	...	value
hh:mm:ss	value	value	...	value

- Select Variables to record
 - Model tags, CommDrivers Tag, Objects properties...
- Configure Datalogger properties
 - Sampling Mode: Periodic/Change in Value
 - Sampling Period: dd:hh:mm:ss.fff
 - Table name (optional)
 - Store: DataStore used to save historical data

Data logger

The screenshot shows the 'Properties' configuration screen for a 'Data logger'. At the top, there are buttons for sorting, filtering, adding, and deleting items. The main area contains the following settings:

- Name:** DataLogger1
- Type:** Data logger
- Sampling mode:** Periodic
- Sampling period:** 0000:00:01.000
- Log operation code...**: False
- Log timestamp...**: False
- Log local time**: True
- Table name**: (empty)
- Variables to log**:
 - ModbusTag1: .../Tags/ModbusTag1
 - ModbusTag2: .../Tags/ModbusTag2
 - ModbusTag3: .../Tags/ModbusTag3
- Store**: EmbeddedDatabase1

- At every sampling event the Datalogger create a new record into a table with the following structure:

Date/Time	Variable 1	Variable 2	...	Variable N
hh:mm:ss	value	value	...	value
hh:mm:ss	value	value	...	value

- Select Variables to record
 - Model tags, CommDrivers Tag, Objects properties...
- Configure Datalogger properties
 - Sampling Mode: Periodic/Change in Value
 - Sampling Period: dd:hh:mm:ss.fff
 - Table name (optional)
 - Store: DataStore used to save historical data

Data logger "viewer"

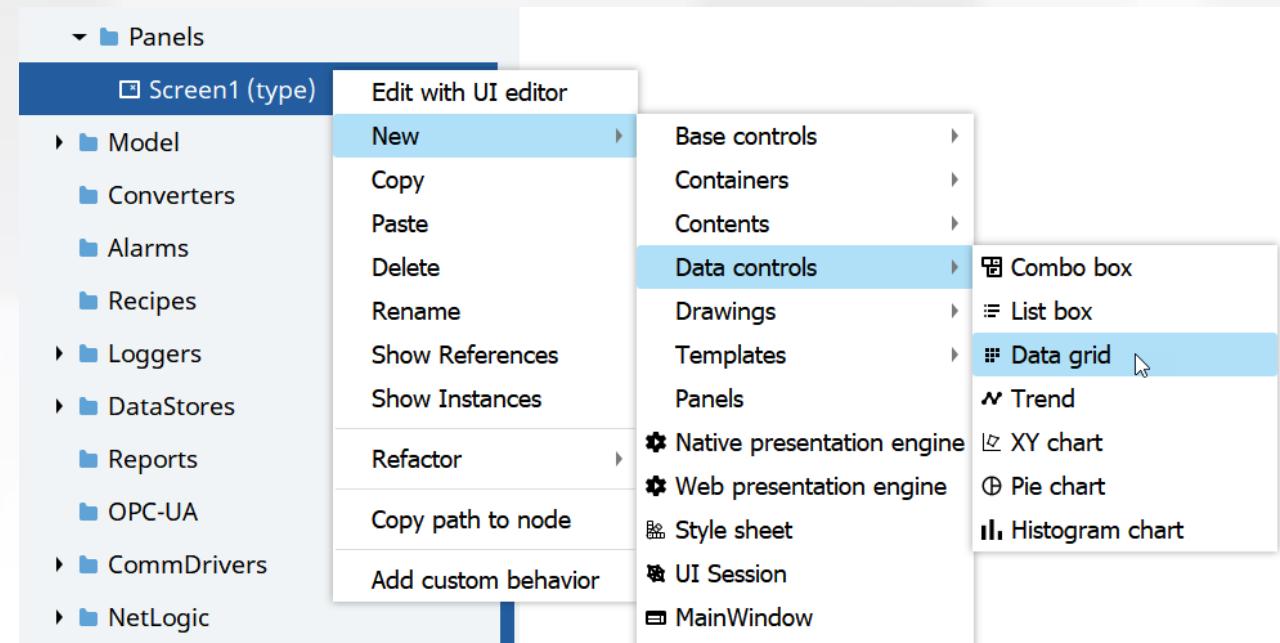
1. Add a "Data grid" object by New > Data controls > Data grid
2. Drag&Drop the Datalogger to the Data grid object

- The object can be customized through Properties pane

- Query
- Columns format
- Sorting

- Export

- Data logger can be exported using the script "Data Logger exporter" available into Template Library. This script can be associated with a button event



Data logger "viewer"

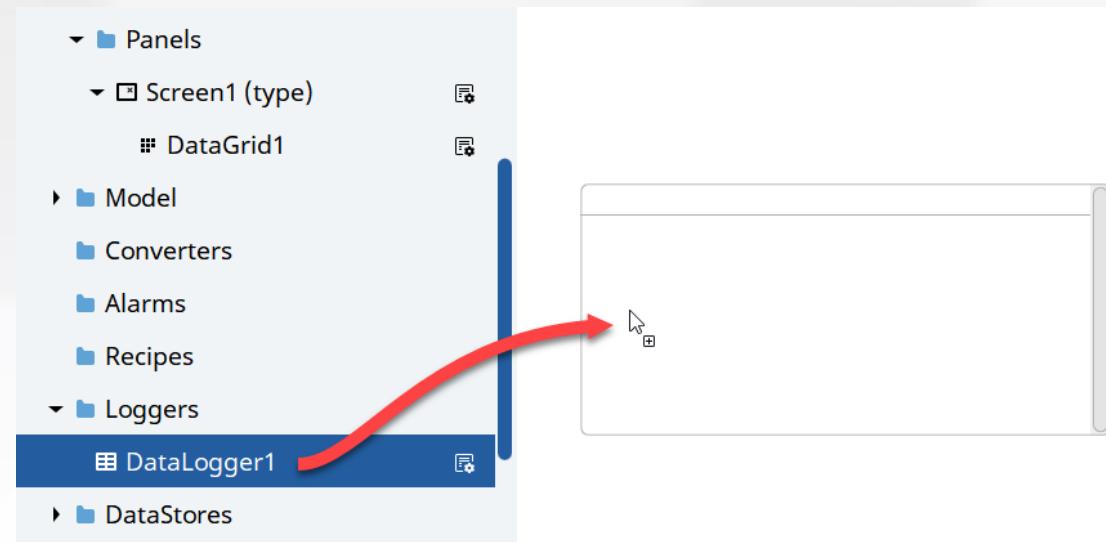
1. Add a "Data grid" object by New > Data controls > Data grid
2. Drag&Drop the Datalogger to the Data grid object

- The object can be customized through Properties pane

- Query
- Columns format
- Sorting

- Export

- Data logger can be exported using the script "Data Logger exporter" available into Template Library. This script can be associated with a button event



Data logger "viewer"

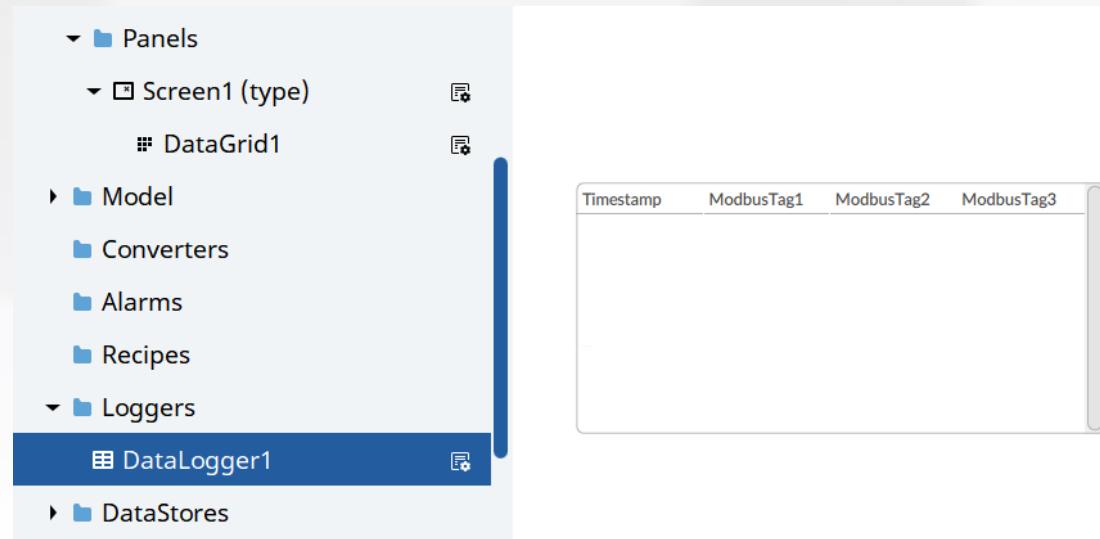
1. Add a "Data grid" object by New > Data controls > Data grid
2. Drag&Drop the Datalogger to the Data grid object

- The object can be customized through Properties pane

- Query
- Columns format
- Sorting

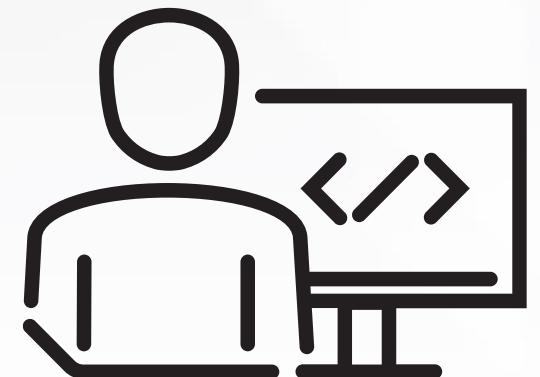
- Export

- Data logger can be exported using the script "Data Logger exporter" available into Template Library. This script can be associated with a button event



Hands-on session

- Define a new Datalogger with a periodic sampling of 5 seconds
- Configure a Grid to show the datalogger table stored on the database
- Add a button to manually trigger a new record
- Add a button to refresh the grid
- Add a switch to start/stop the sampling



Event logger

The screenshot shows the configuration interface for an 'Event logger' named 'EventLogger1'. The interface includes a toolbar with search, sort, and add/delete buttons, and a properties table.

Property	Value
Name	EventLogger1
Type	Event logger
Enabled	True
Locales	
Event configurations	
EventConfiguration1	
Event source	Tags
Listen subtree	True
Event type	AuditWriteUpdateEventType
Non-interactive s...	<i>False</i>
Event fields to log	
Database	EmbeddedDatabase1

- At every event the Event Logger creates a new record into the table with structure defined into "Event fields to log"

Time	Client User ID	Source Node	Old Value	New Value
hh:mm:ss	Guest	Variable1	0	5
hh:mm:ss	Admin	Variable2	2	8

- Configure the Source and Event to log
- Select Event fields to log
- Define the Datastore

Event logger

Loggers > EventLogger1

BaseEventType	
<input type="checkbox"/> var	EventId
<input type="checkbox"/> var	SourceName
<input checked="" type="checkbox"/> var	SourceNode
<input checked="" type="checkbox"/> var	Time
AuditEventType	
<input type="checkbox"/> var	ActionTimeStamp
<input type="checkbox"/> var	ClientAuditEntryId
<input checked="" type="checkbox"/> var	ClientUserId
<input type="checkbox"/> var	ServerId
<input type="checkbox"/> var	Status
AuditUpdateEventType	
AuditWriteUpdateEventType	
<input type="checkbox"/> var	AttributeId
<input type="checkbox"/> var	IndexRange
<input checked="" type="checkbox"/> var	NewValue
<input checked="" type="checkbox"/> var	OldValue

- At every event the Event Logger creates a new record into the table with structure defined into "Event fields to log"

Time	Client User ID	Source Node	Old Value	New Value
hh:mm:ss	Guest	Variable1	0	5
hh:mm:ss	Admin	Variable2	2	8

- Configure the Source and Event to log
- Select Event fields to log
- Define the Datastore

Event logger

The screenshot shows the configuration interface for an 'Event logger' named 'EventLogger1'. The interface includes a 'Properties' tab with general settings like 'Name' and 'Type', and a detailed 'Event configurations' section. Under 'Event configurations', there is one entry: 'EventConfiguration1' which has 'Event source' set to 'Tags', 'Listen subtree' to 'True', 'Event type' to 'AuditWriteUpdateEventType', and 'Non-interactive s...' to 'False'. Below this, the 'Event fields to log' section lists several fields with their corresponding Xpath expressions: Time (UtcTime/EventArgs/Time), ClientUserId (String/EventArgs/ClientUserId), SourceNode (String/EventArgs/SourceNode@BrowseName), OldValue (Int32/EventArgs/OldValue@Value), NewValue (Int32/EventArgs/NewValue@Value), and Database (EmbeddedDatabase1). Each field has a small edit icon next to it.

- At every event the Event Logger creates a new record into the table with structure defined into "Event fields to log"

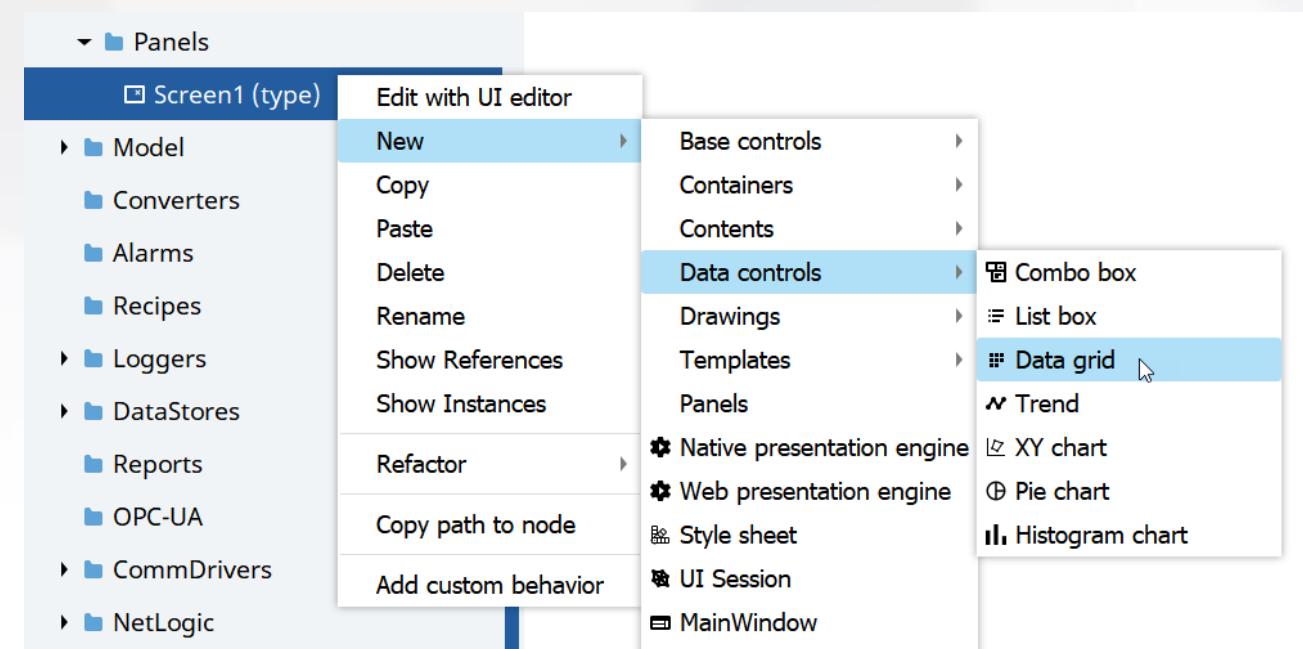
Time	Client User ID	Source Node	Old Value	New Value
hh:mm:ss	Guest	Variable1	0	5
hh:mm:ss	Admin	Variable2	2	8

- Configure the Source and Event to log
- Select Event fields to log
- Define the Datastore

Event logger "viewer"

1. Add a "Data grid" object by New > Data controls > Data grid
2. Drag&Drop the Event Logger to the Data grid object

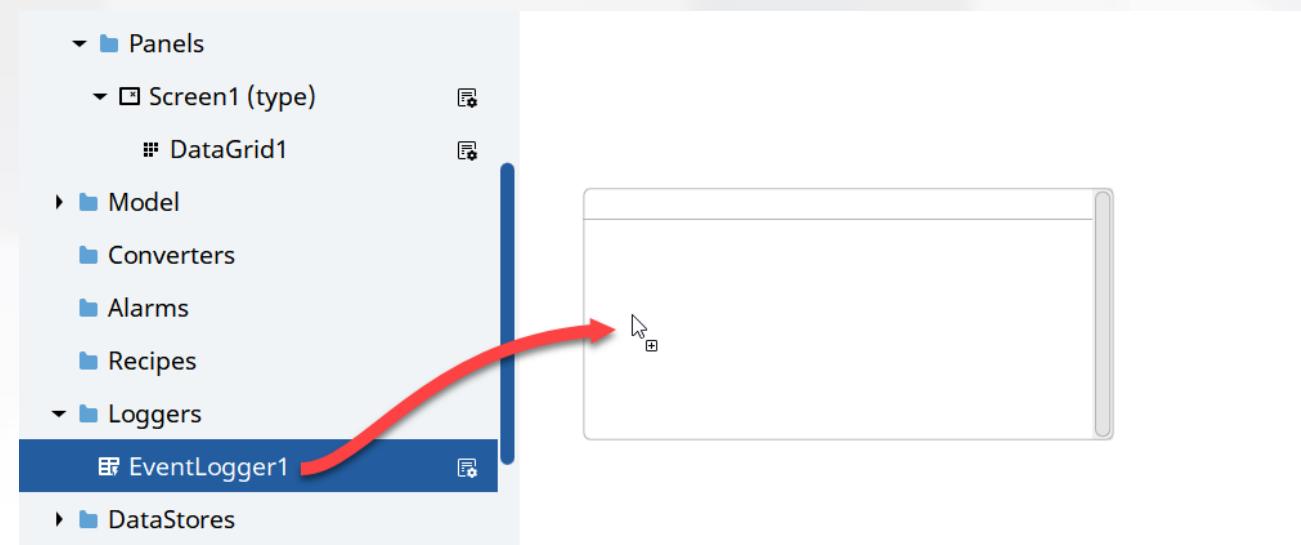
- The object can be customized through Properties pane
 - Query
 - Columns format
 - Sorting



Event logger "viewer"

1. Add a "Data grid" object by New > Data controls > Data grid
2. Drag&Drop the Event Logger to the Data grid object

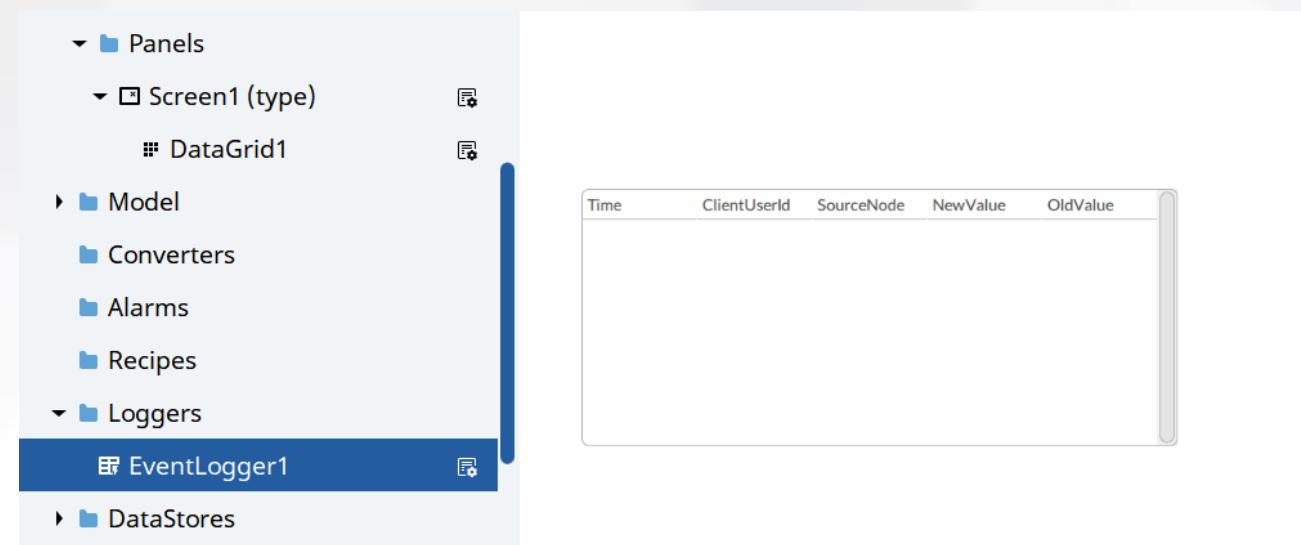
- The object can be customized through Properties pane
 - Query
 - Columns format
 - Sorting



Event logger "viewer"

1. Add a "Data grid" object by New > Data controls > Data grid
2. Drag&Drop the Event Logger to the Data grid object

- The object can be customized through Properties pane
 - Query
 - Columns format
 - Sorting



Hands-on session

- Configure an Event Logger to trace changes made to variables
- Add a Grid to show the Event Logger table
- Add a button to refresh the Grid



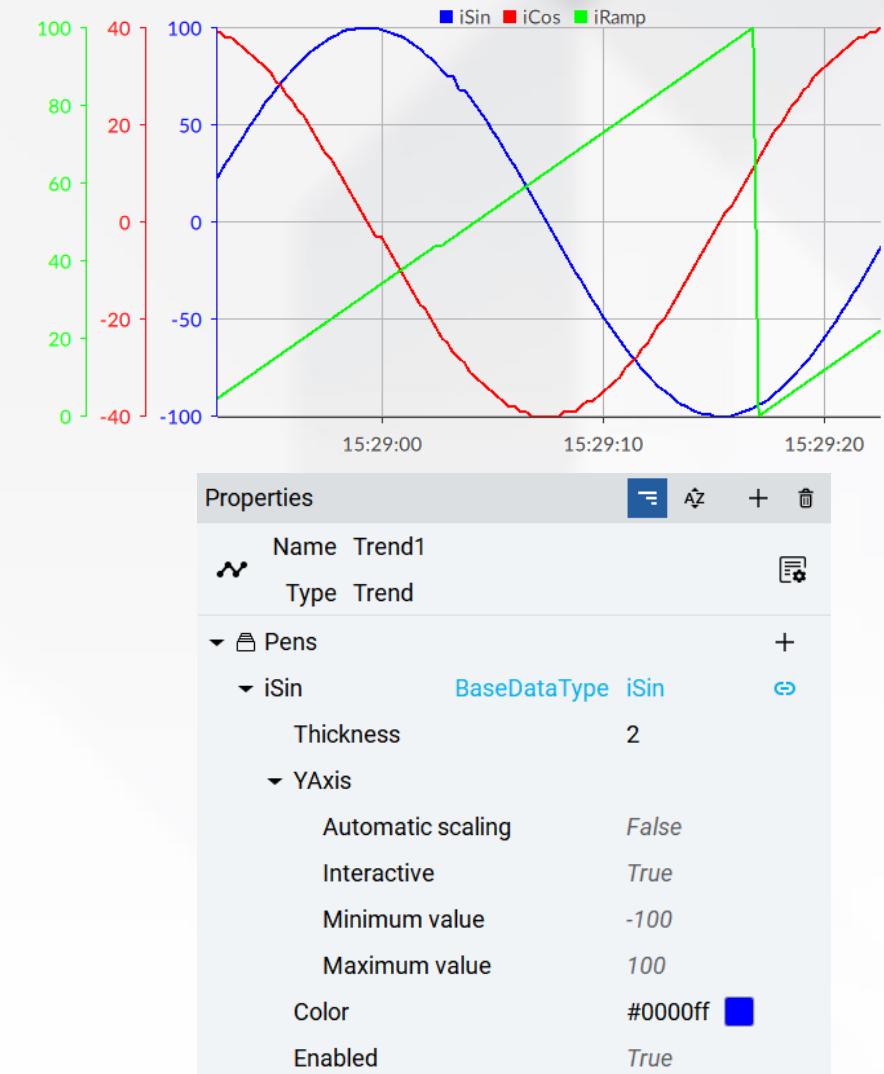


Represent data by Trends



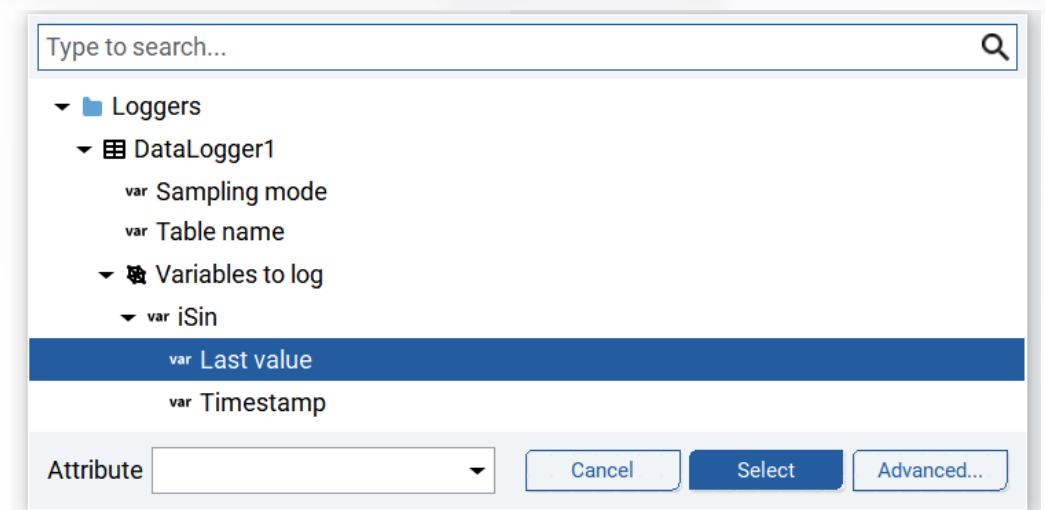
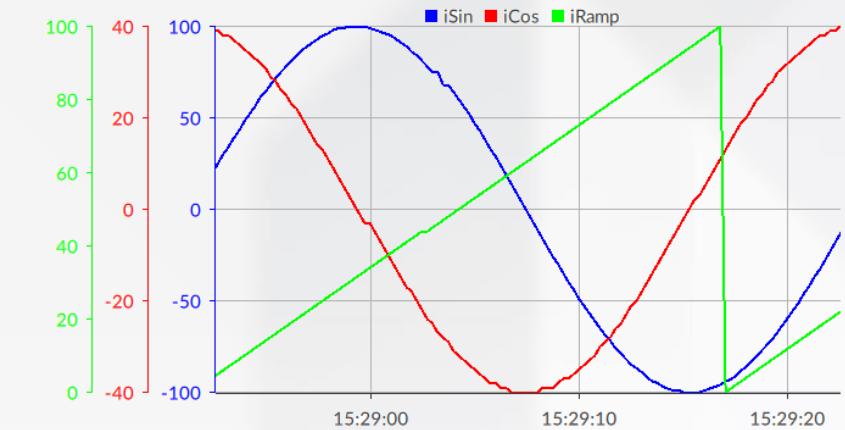
Real-time Trends

- Trend widget is available under the "Data Controls" group
- Real-time Trend = data showed in the Trend window, will be purged on the exit
- To configure a Trend as a Real-time one, is enough to associate a Pen with a Variable
- Optionally, a Y-Axis for each Pen can be configured



Historical Trends

- Historical Trends = when the Trend window is loaded, a query is executed to retrieve historical data
- To configure a Trend as an Historical one:
 - a DataLogger is needed to save historical data
 - Pens have to be associated to
DataLogger > Variables to Log > {name} > Last Value



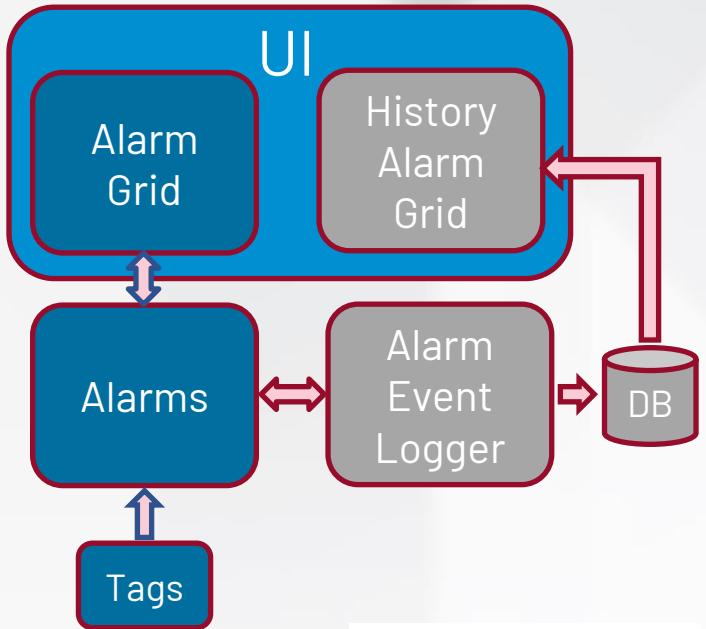


Manage alarms



Alarms

- Alarm functional module allows the definition of alarm events connected to Tags and manage them at runtime
- Active Alarms
 - just define alarms
 - drag&drop Alarm Grid widget from Template Library to UI
- History of Alarms
 - add a new datastore (Embedded database or ODBC)
 - drag&drop Alarm Event Logger object from Template Library, into Loggers node
 - drag&drop History Alarm Grid widget from Template Library to UI



OPC/UA Alarms
specifications

Alarm types

- Types of alarms:
 - *Digital*: "two-state" alarm, typically used to monitor Boolean tags
 - *Level*: an alarm that activates when the monitored variable exceed the range "Low – High"
 - *Deviation*: alarms that activate when the monitored variable exceed the range defined by a Setpoint and Low/High deviation limit
 - *Rate of Change*: like Level Alarm but evaluated with specific polling time (i.e. to filter spurious spikes)
- Exclusive/Non-Exclusive (when all four limits are used)
 - Exclusive: exceeding High and High-high limits, only High-high alarms will be shown
 - Non-Exclusive: exceeding High and High-high limits, both alarms will be shown

Properties	
I/O	Name
	DigitalAlarm1
	Type
	Digital alarm
	Input variable
	0
	Normal state value
	0
	Enabled
	<i>True</i>
	Auto acknowledge
	<i>False</i>
	Auto confirm
	<i>False</i>
	Message
	Severity
	1

Alarm types

- Types of alarms:
 - *Digital*: "two-state" alarm, typically used to monitor Boolean tags
 - *Level*: an alarm that activates when the monitored variable exceed the range "Low – High"
 - *Deviation*: alarms that activate when the monitored variable exceed the range defined by a Setpoint and Low/High deviation limit
 - *Rate of Change*: like Level Alarm but evaluated with specific polling time (i.e. to filter spurious spikes)
- Exclusive/Non-Exclusive (when all four limits are used)
 - Exclusive: exceeding High and High-high limits, only High-high alarms will be shown
 - Non-Exclusive: exceeding High and High-high limits, both alarms will be shown

Properties	
   	
Name	ExclusiveLevelAlarm1
Type	Exclusive level alarm
Input variable	0
High-high limit	0
High limit	0
Low limit	0
Low-low limit	0
Enabled	<i>True</i>
Auto acknowledge	<i>False</i>
Auto confirm	<i>False</i>
Message	
Severity	1

Alarm types

- Types of alarms:
 - *Digital*: "two-state" alarm, typically used to monitor Boolean tags
 - *Level*: an alarm that activates when the monitored variable exceed the range "Low – High"
 - *Deviation*: alarms that activate when the monitored variable exceed the range defined by a Setpoint and Low/High deviation limit
 - *Rate of Change*: like Level Alarm but evaluated with specific polling time (i.e. to filter spurious spikes)
- Exclusive/Non-Exclusive (when all four limits are used)
 - Exclusive: exceeding High and High-high limits, only High-high alarms will be shown
 - Non-Exclusive: exceeding High and High-high limits, both alarms will be shown

Properties	
Name	ExclusiveDeviationAlarm1
Type	Exclusive deviation alarm
Input variable	0
Set point value	0
High-high limit	0
High limit	0
Low limit	0
Low-low limit	0
Enabled	<i>True</i>
Auto acknowledge	<i>False</i>
Auto confirm	<i>False</i>
Message	
Severity	1

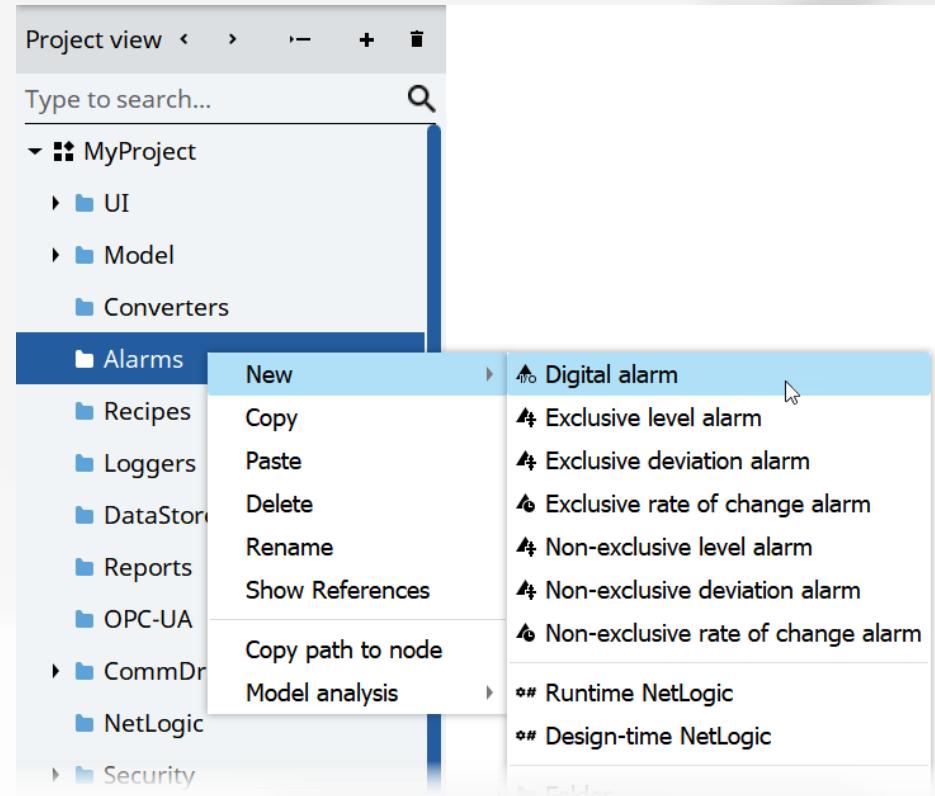
Alarm types

- Types of alarms:
 - *Digital*: "two-state" alarm, typically used to monitor Boolean tags
 - *Level*: an alarm that activates when the monitored variable exceed the range "Low – High"
 - *Deviation*: alarms that activate when the monitored variable exceed the range defined by a Setpoint and Low/High deviation limit
 - *Rate of Change*: like Level Alarm but evaluated with specific polling time (i.e. to filter spurious spikes)
- Exclusive/Non-Exclusive (when all four limits are used)
 - Exclusive: exceeding High and High-high limits, only High-high alarms will be shown
 - Non-Exclusive: exceeding High and High-high limits, both alarms will be shown

Properties	
   	
Name	ExclusiveRateOfChangeAlarm1
Type	Exclusive rate of change alarm
Input variable	0
Polling time	0000:00:00.000
High-high limit	0
High limit	0
Low limit	0
Low-low limit	0
Enabled	<i>True</i>
Auto acknowledge	<i>False</i>
Auto confirm	<i>False</i>
Message	
Severity	1

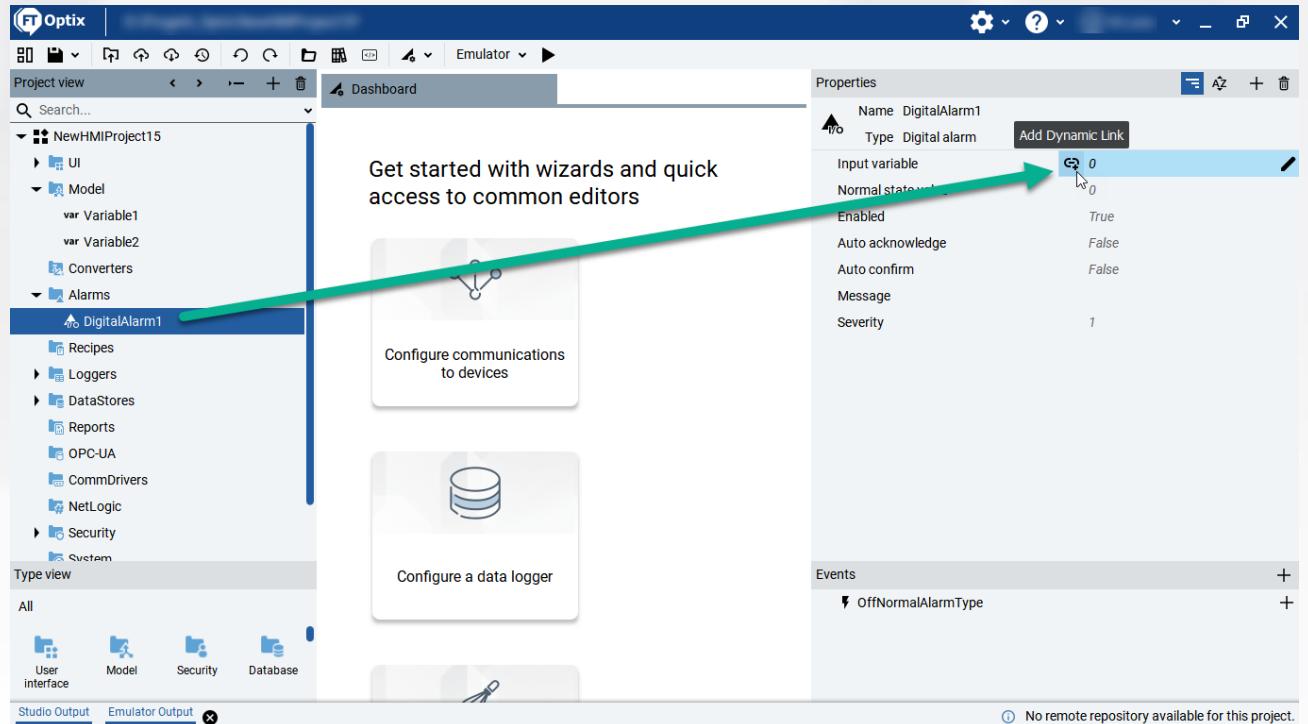
Alarm definition

- How to define alarms
 1. Choose the alarm type
 2. Configure Input Variable and properties
- For Tags where every bit bring alarm information
 1. Select the single tag
 2. Select the bit index
- Import/Export
 - Alarms Definition can be imported/exported to CSV file using a Template Library script
 - The path of the CSV file can be relative using the %PROJECTDIR% placeholder



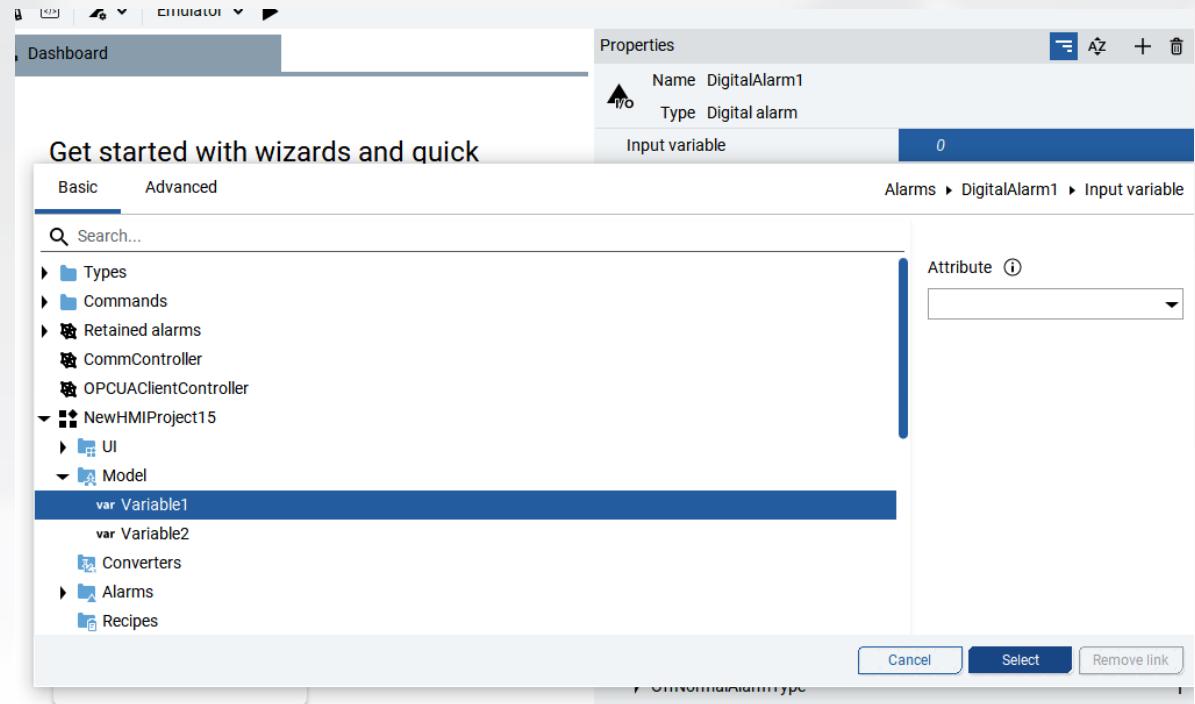
Alarm definition

- How to define alarms
 1. Choose the alarm type
 2. Configure Input Variable and properties
- For Tags where every bit bring alarm information
 1. Select the single tag
 2. Select the bit index
- Import/Export
 - Alarms Definition can be imported/exported to CSV file using a Template Library script
 - The path of the CSV file can be relative using the %PROJECTDIR% placeholder



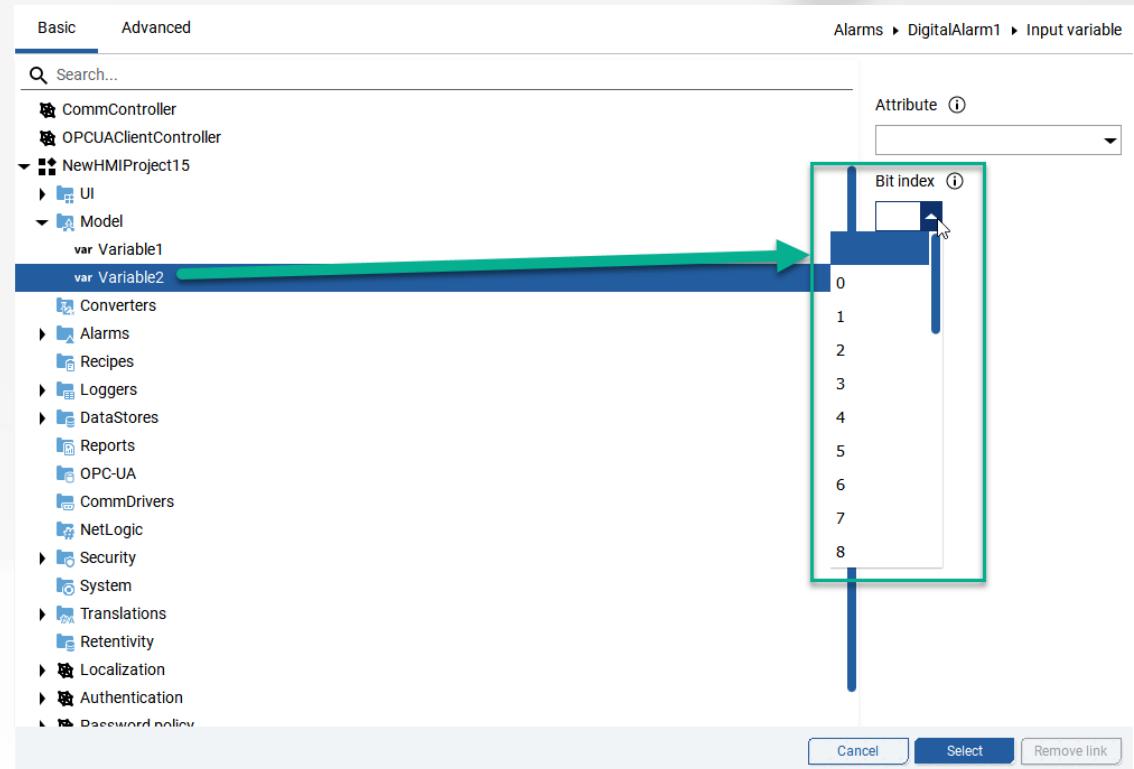
Alarm definition

- How to define alarms
 1. Choose the alarm type
 2. Configure Input Variable and properties
- For Tags where every bit bring alarm information
 1. Select the single tag
 2. Select the bit index
- Import/Export
 - Alarms Definition can be imported/exported to CSV file using a Template Library script
 - The path of the CSV file can be relative using the %PROJECTDIR% placeholder



Alarm definition

- How to define alarms
 1. Choose the alarm type
 2. Configure Input Variable and properties
- For Tags where every bit bring alarm information
 1. Select the single tag
 2. Select the bit index
- Import/Export
 - Alarms Definition can be imported/exported to CSV file using a Template Library script
 - The path of the CSV file can be relative using the %PROJECTDIR% placeholder



Show alarms

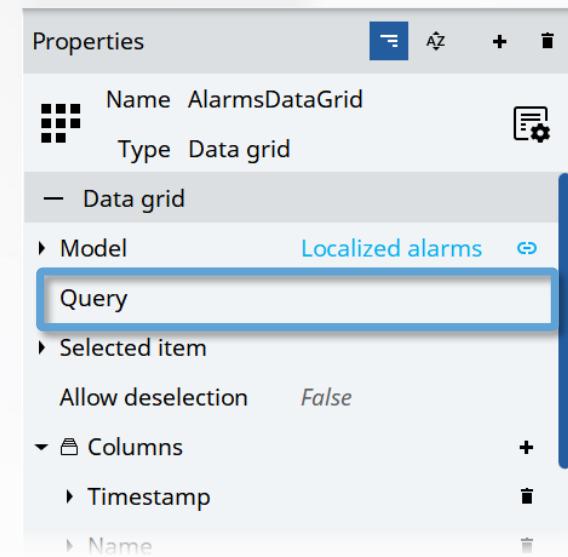
- Available Widgets

- into Template Library
 - Alarm Grid,
 - Alarm Banner,
 - Alarm Icon

Timestamp	Name	Source Variable	Message	Active	Acked	Confirmed	Severity
Feb 10, 2022, 3:44:05 PM	DigitalAlarm1.1	ModbusTag1	Alarm on Tag1 bit 1	●	✗	✓	1
Feb 10, 2022, 3:44:04 PM	DigitalAlarm1.0	ModbusTag1	Alarm on Tag1 bit 0	●	✗	✓	1

Acknowledge **Acknowledge All** **Confirm** **Confirm All**

- customizable
 - columns,
 - buttons,
 - the "Query" parameter allow to filter alarms of Alarm Grid
- support Messages in different languages



Show alarms history

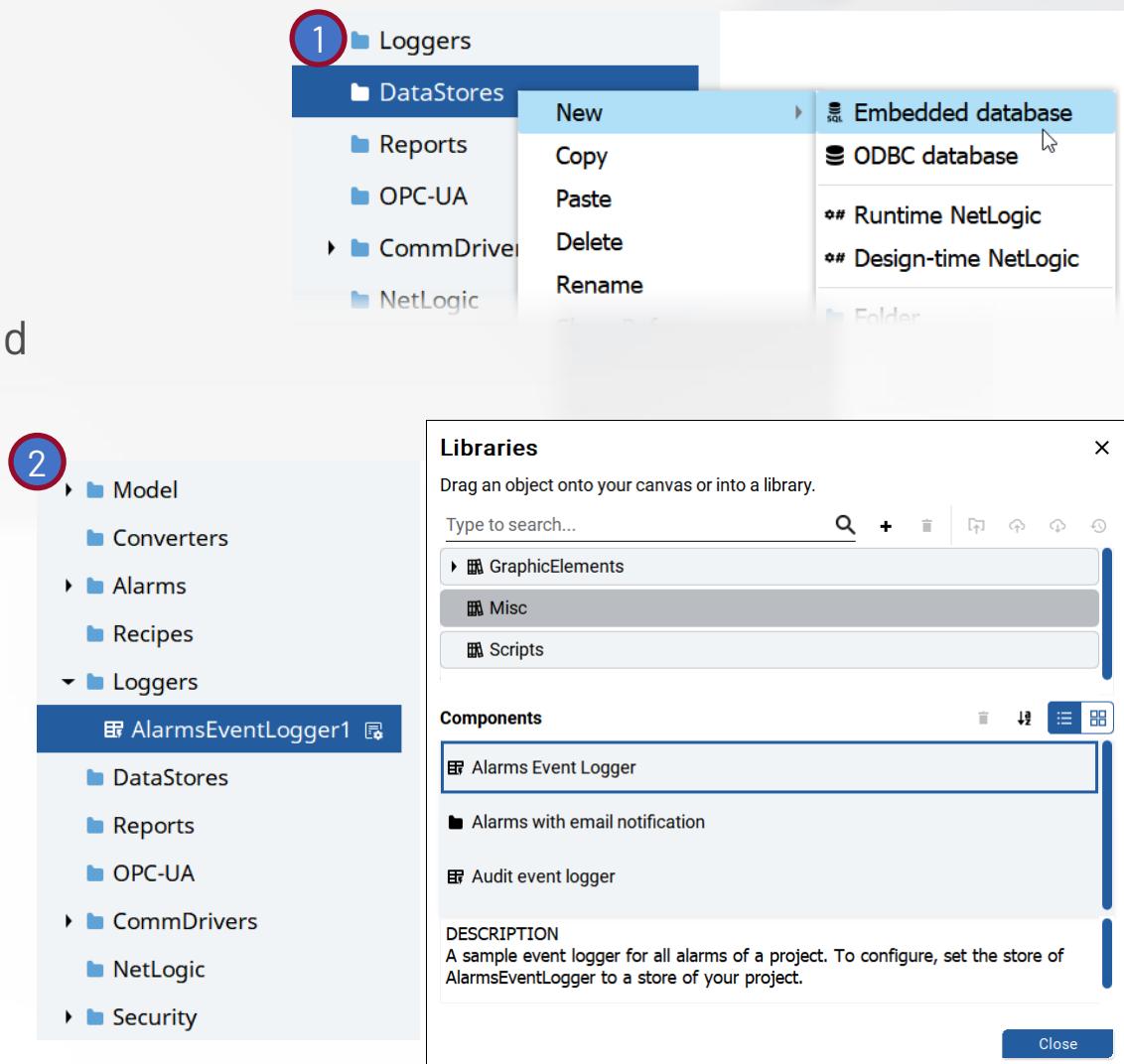
- To record alarms history some additional resources are needed
 1. Datastore resource (if not already present)
 2. Alarm event logger (from Template Library) just drag&drop and configure datastore to be used

- Available Widgets

- Alarm History grid
- Alarm History grid with filter

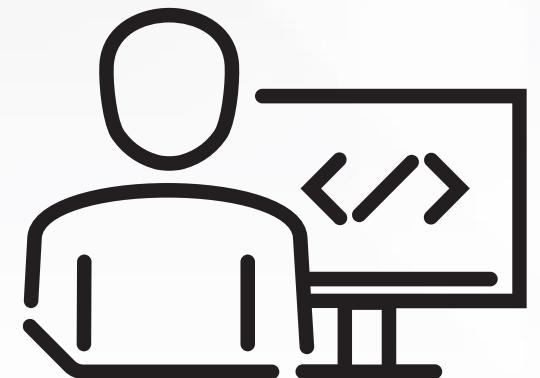
- Export

- Alarm History can be exported with script "Alarms History Exporter" available in Template Library



Hands-on session

- Define some digital alarms configuring the input variables
- Add the Alarm Grid widget to see alarms at runtime with the emulator
- Add the Alarm Event Logger to log historical alarm events
- Add the Alarm History Grid widget to see historical



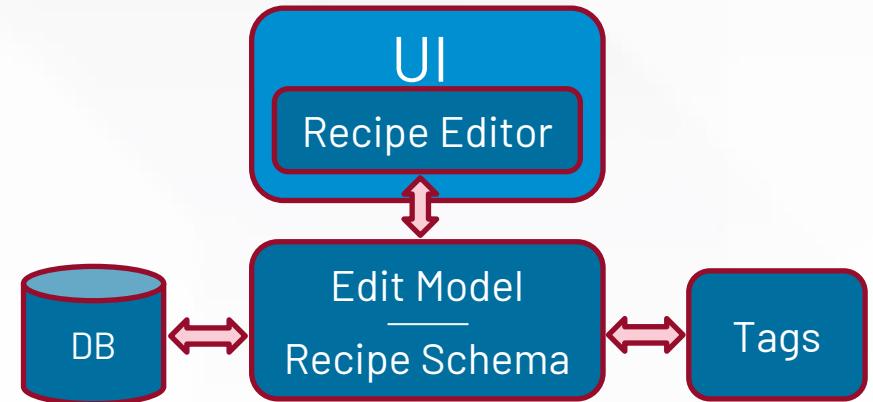
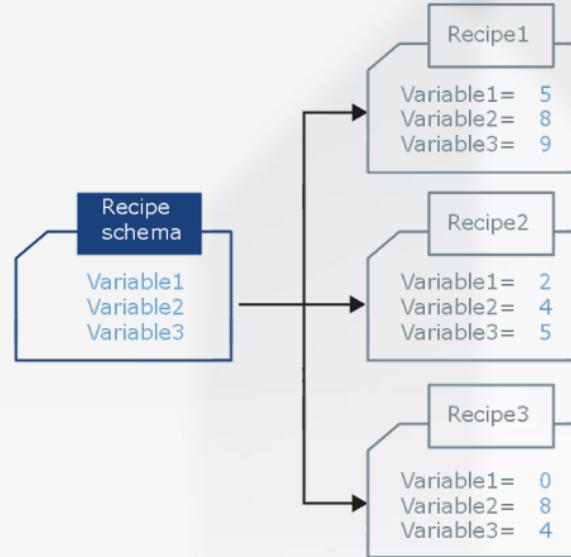


Use recipes



Recipes

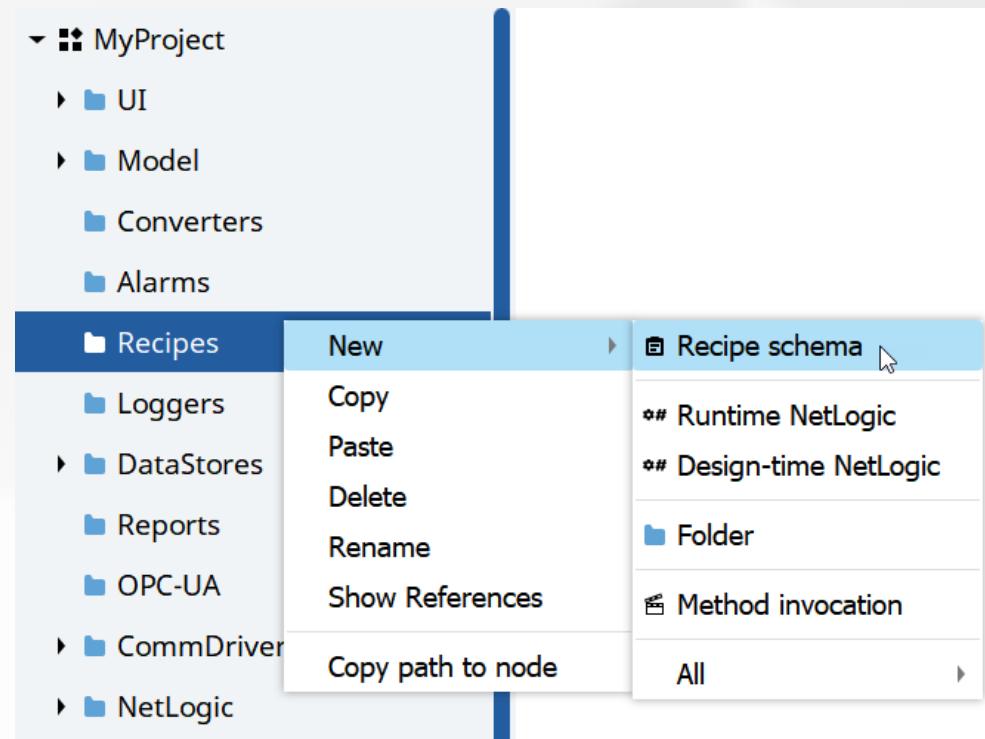
- Allows defining a group of values/set points stored in database archives (embedded or ODBC)
- in Studio
 - configure the Recipe Schema by defining datastore and selecting Tags
- in Runtime
 - User can create several recipes, by defining values manually or by reading values directly from PLC
 - User can store the recipe in the database for future use
 - User can retrieve the recipe from the database and transfer to the PLC



Configure recipe schema

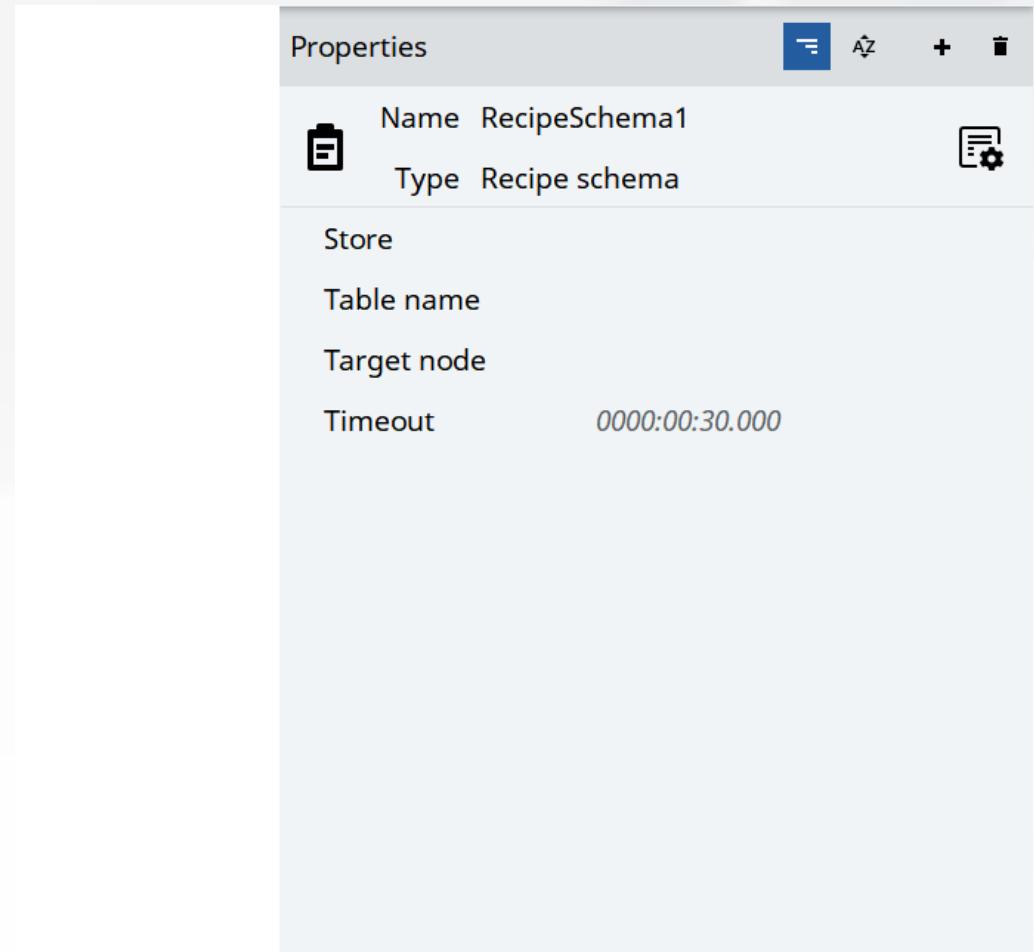
- Steps to configure Recipes

1. Add a Recipe Schema by
Recipes > New > Recipe Schema
2. Select the Datastore
(if not available add a new one)
3. Configure the Target Node
This is the project folder containing Variables
that will be used in the Recipe Schema
Example: Tags, Models...
4. Select which Variables add to the Schema
then confirm with "Apply"



Configure recipe schema

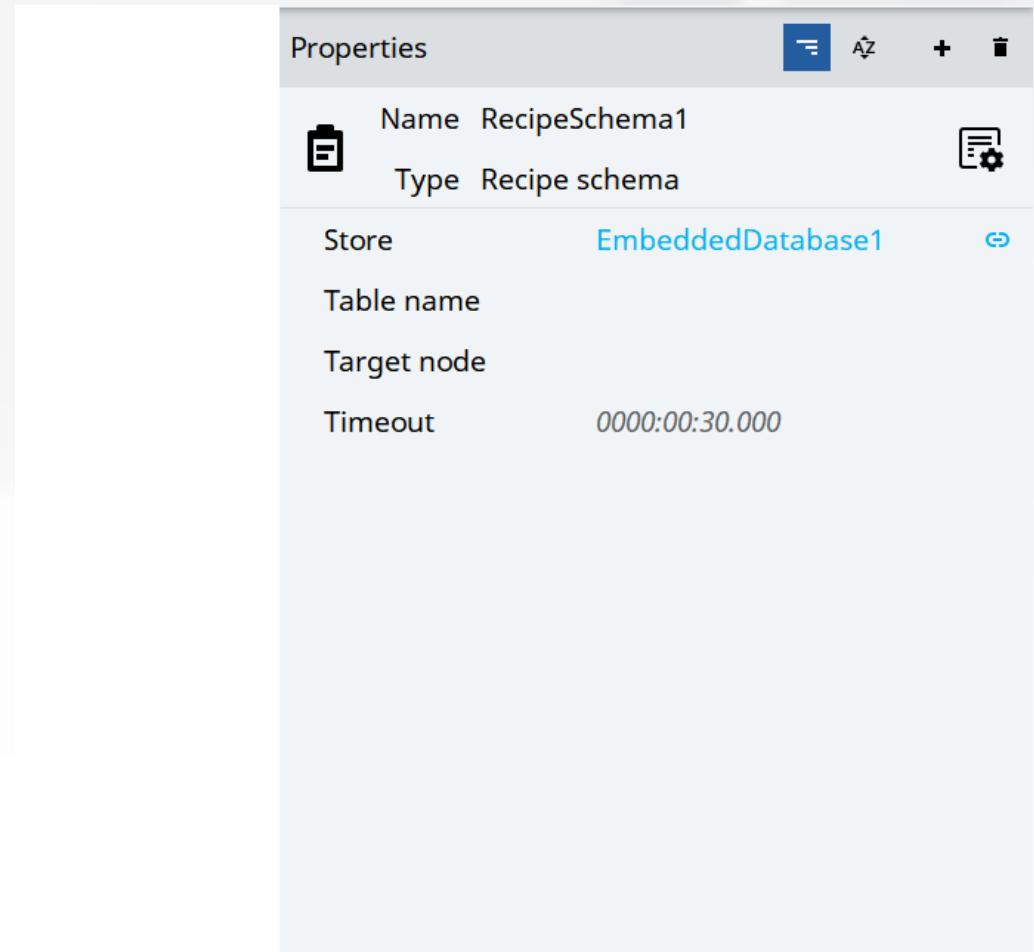
- Steps to configure Recipes
 1. Add a Recipe Schema by
Recipes > New > Recipe Schema
 2. Select the Datastore
(if not available add a new one)
 3. Configure the Target Node
This is the project folder containing Variables
that will be used in the Recipe Schema
Example: Tags, Models...
 4. Select which Variables add to the Schema
then confirm with "Apply"



Configure recipe schema

- Steps to configure Recipes

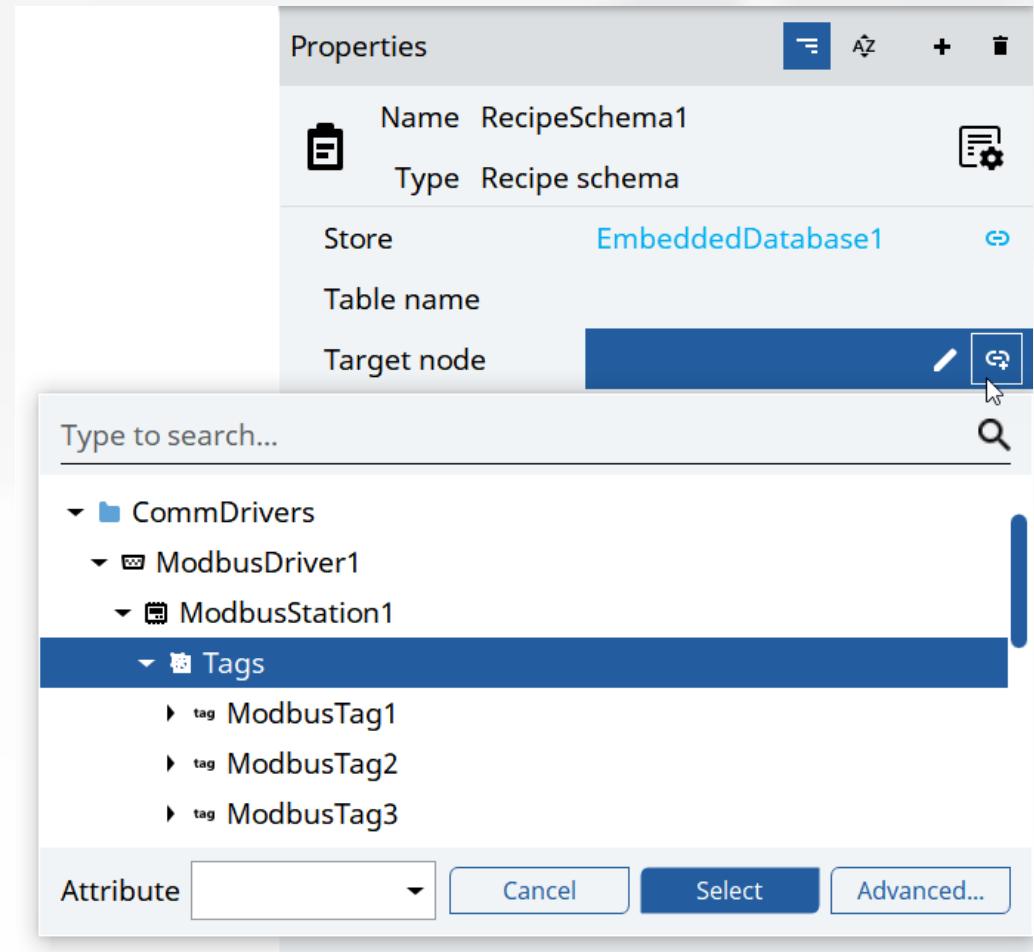
1. Add a Recipe Schema by
Recipes > New > Recipe Schema
2. Select the Datastore
(if not available add a new one)
3. Configure the Target Node
This is the project folder containing Variables
that will be used in the Recipe Schema
Example: Tags, Models...
4. Select which Variables add to the Schema
then confirm with "Apply"



Configure recipe schema

- Steps to configure Recipes

1. Add a Recipe Schema by
Recipes > New > Recipe Schema
2. Select the Datastore
(if not available add a new one)
3. Configure the Target Node
This is the project folder containing Variables
that will be used in the Recipe Schema
Example: Tags, Models...
4. Select which Variables add to the Schema
then confirm with "Apply"



Configure recipe schema

- Steps to configure Recipes

1. Add a Recipe Schema by
Recipes > New > Recipe Schema
2. Select the Datastore
(if not available add a new one)
3. Configure the Target Node
This is the project folder containing Variables
that will be used in the Recipe Schema
Example: Tags, Models...
4. Select which Variables add to the Schema
then confirm with "Apply"

Properties	
Name	RecipeSchema1
Type	Recipe schema
Store	EmbeddedDatabase1
Table name	
Target node	Tags
Timeout	0000:00:30.000

Configure recipe schema

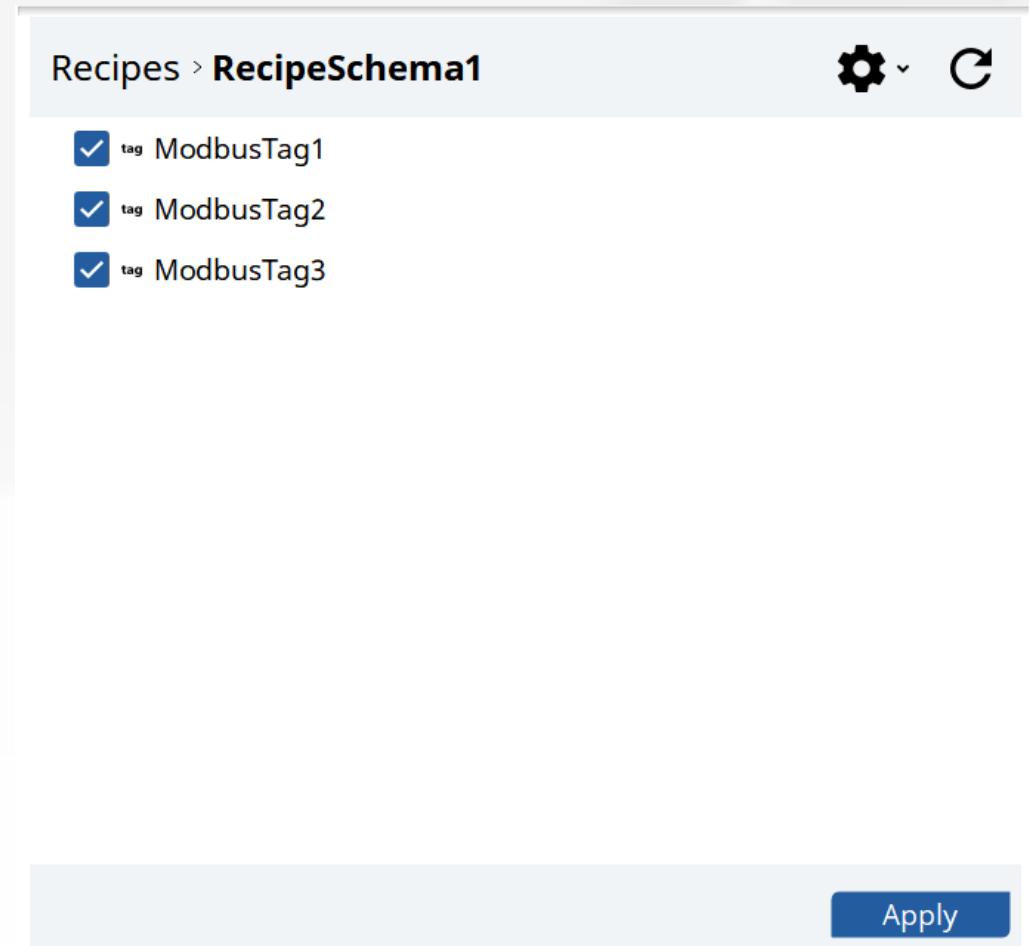
- Steps to configure Recipes

1. Add a Recipe Schema by
Recipes > New > Recipe Schema

2. Select the Datastore
(if not available add a new one)

3. Configure the Target Node
This is the project folder containing Variables
that will be used in the Recipe Schema
Example: Tags, Models...

4. Select which Variables add to the Schema
then confirm with "Apply"



Recipe editor widget

- Steps to configure the Recipe Editor widget

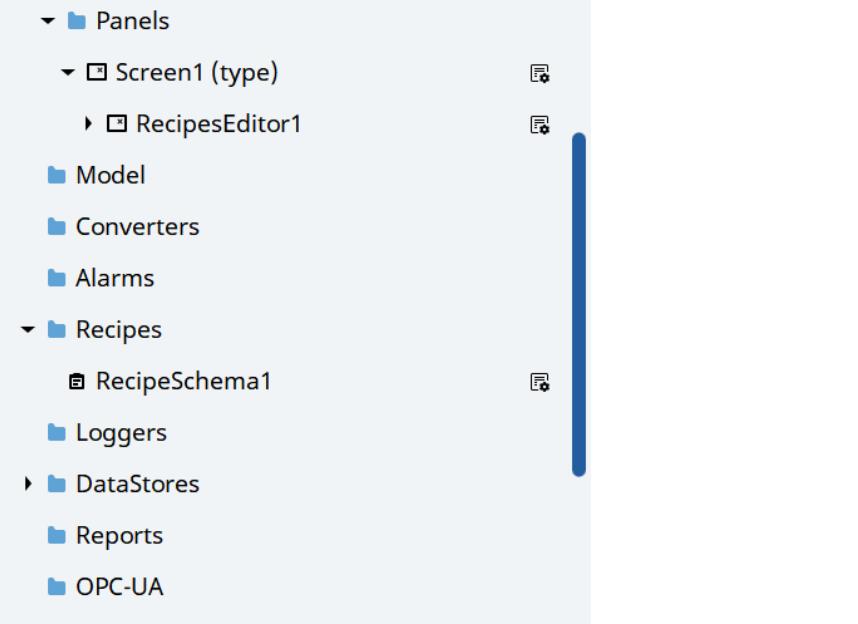
1. Drag&drop the widget from Template Library
2. Into Property pane,
select the recipe schema
3. Right-click on the "RecipeEditorUISetup"
and select "Execute Setup"

- Widget:

- Save: save/create the recipe into the database
- Delete: delete the recipe from the database
- Load: read Tag values and load them into recipe values
- Apply: write recipe values into Tags

- Import/Export

- Recipes can be exported/imported
through script available into Template Library



Recipe editor widget

- Steps to configure the Recipe Editor widget

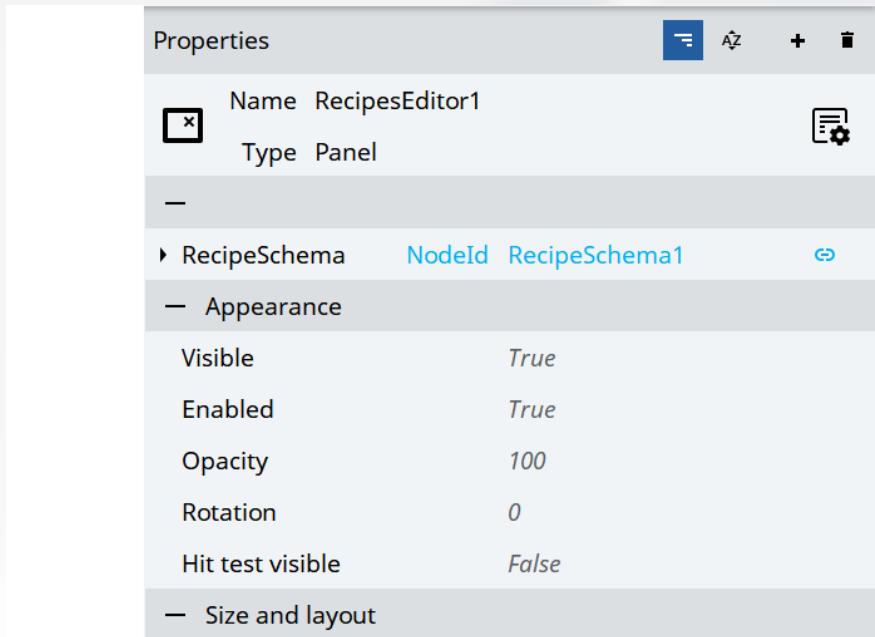
1. Drag&drop the widget from Template Library
2. Into Property pane,
select the recipe schema
3. Right-click on the "RecipeEditorUISetup"
and select "Execute Setup"

- Widget:

- Save: save/create the recipe into the database
- Delete: delete the recipe from the database
- Load: read Tag values and load them into recipe values
- Apply: write recipe values into Tags

- Import/Export

- Recipes can be exported/imported
through script available into Template Library



Recipe editor widget

- Steps to configure the Recipe Editor widget

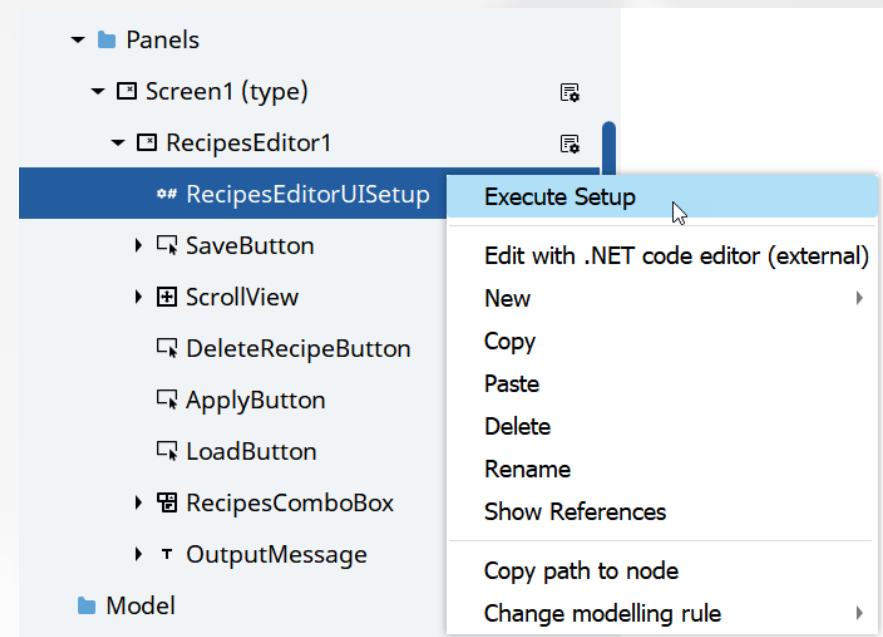
1. Drag&drop the widget from Template Library
2. Into Property pane,
select the recipe schema
3. Right-click on the "RecipeEditorUISetup"
and select "Execute Setup"

- Widget:

- Save: save/create the recipe into the database
- Delete: delete the recipe from the database
- Load: read Tag values and load them into recipe values
- Apply: write recipe values into Tags

- Import/Export

- Recipes can be exported/imported
through script available into Template Library



Recipe editor widget

- Steps to configure the Recipe Editor widget

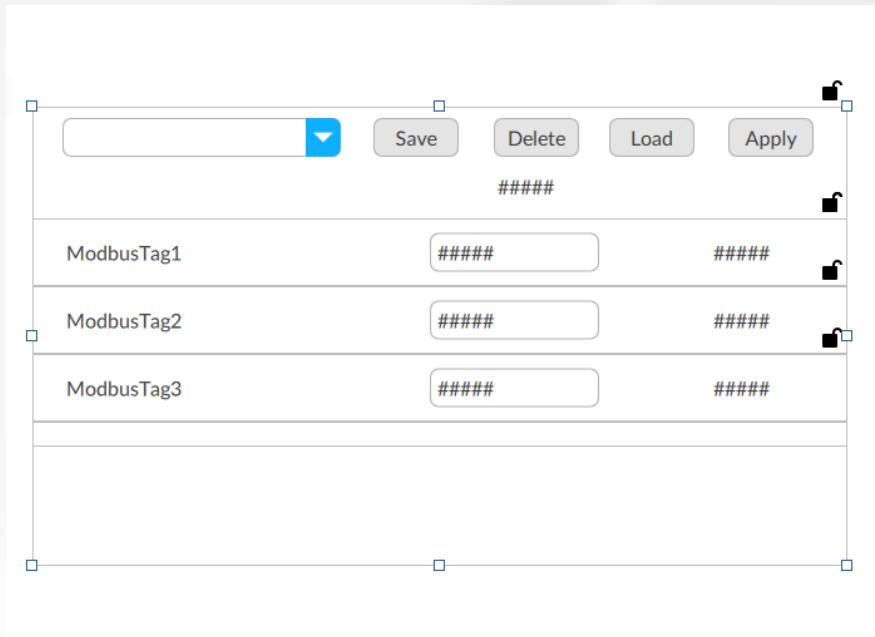
1. Drag&drop the widget from Template Library
2. Into Property pane,
select the recipe schema
3. Right-click on the "RecipeEditorUISetup"
and select "Execute Setup"

- Widget:

- Save: save/create the recipe into the database
- Delete: delete the recipe from the database
- Load: read Tag values and load them into recipe values
- Apply: write recipe values into Tags

- Import/Export

- Recipes can be exported/imported
through script available into Template Library



Recipe editor widget

- Steps to configure the Recipe Editor widget

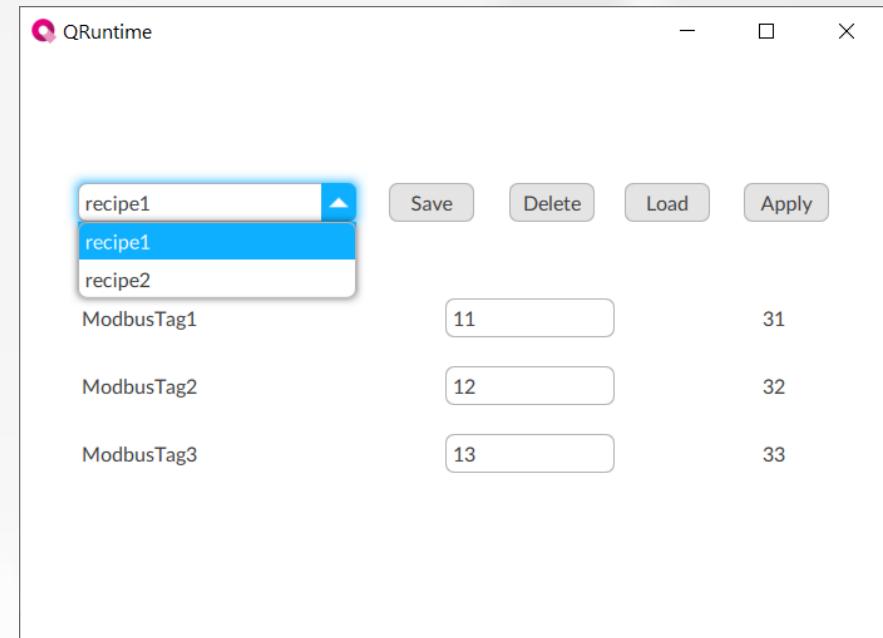
1. Drag&drop the widget from Template Library
2. Into Property pane,
select the recipe schema
3. Right-click on the "RecipeEditorUISetup"
and select "Execute Setup"

- Widget:

- Save: save/create the recipe into the database
- Delete: delete the recipe from the database
- Load: read Tag values and load them into recipe values
- Apply: write recipe values into Tags

- Import/Export

- Recipes can be exported/imported
through script available into Template Library



Recipe editor widget

- Steps to configure the Recipe Editor widget

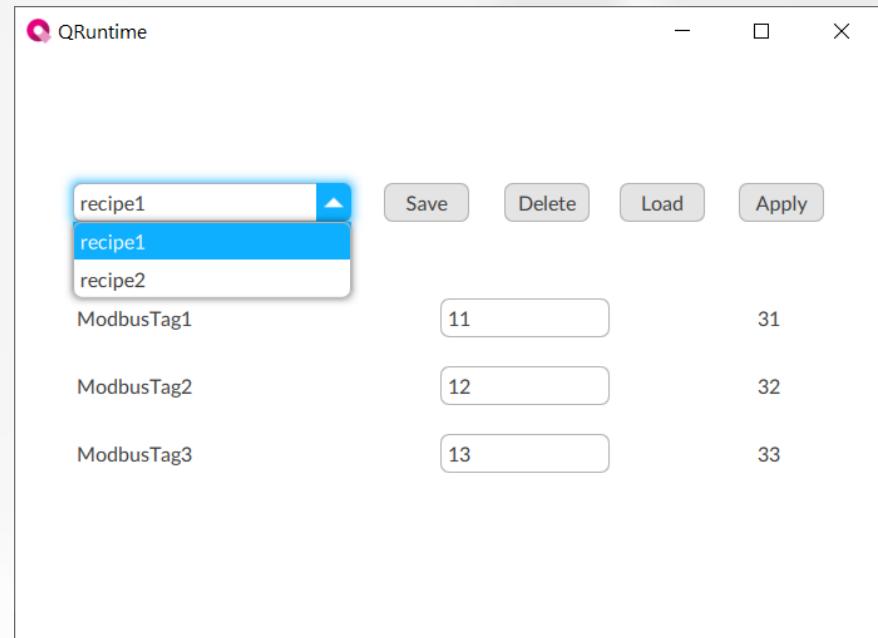
1. Drag&drop the widget from Template Library
2. Into Property pane,
select the recipe schema
3. Right-click on the "RecipeEditorUISetup"
and select "Execute Setup"

- Widget:

- Save: save/create the recipe into the database
- Delete: delete the recipe from the database
- Load: read Tag values and load them into recipe values
- Apply: write recipe values into Tags

- Import/Export

- Recipes can be exported/imported
through script available into Template Library



Hands-on session

- Create a recipe schema using tags coming from the communication driver
- Add the recipe widget to the project
- Execute the Recipe widget "setup"
- Play with the widget at runtime with Emulator



A large, abstract graphic of blue dots forms a winding, undulating line that spans the width of the slide, starting from the bottom left and curving upwards towards the top right.

Set Locale, Languages and Measurement System



Localization

- a Locale is **a set of User Interface settings** based on country

Locale	en-US	it-IT
Regional Settings	mm/dd/yyyy ... 10.5	dd/mm/yyyy ... 10,5
Language/Translations	Hello	Ciao
Measurement System	°F	°C

- Locale, Language and Measurement System are **related to the Session**
 - A session is a Runtime context in which a user executes operations
 - Examples: a session of Native Presentation Engine can look different from a session of Web Presentation Engine.
A session of User1 can look different from a session of User2

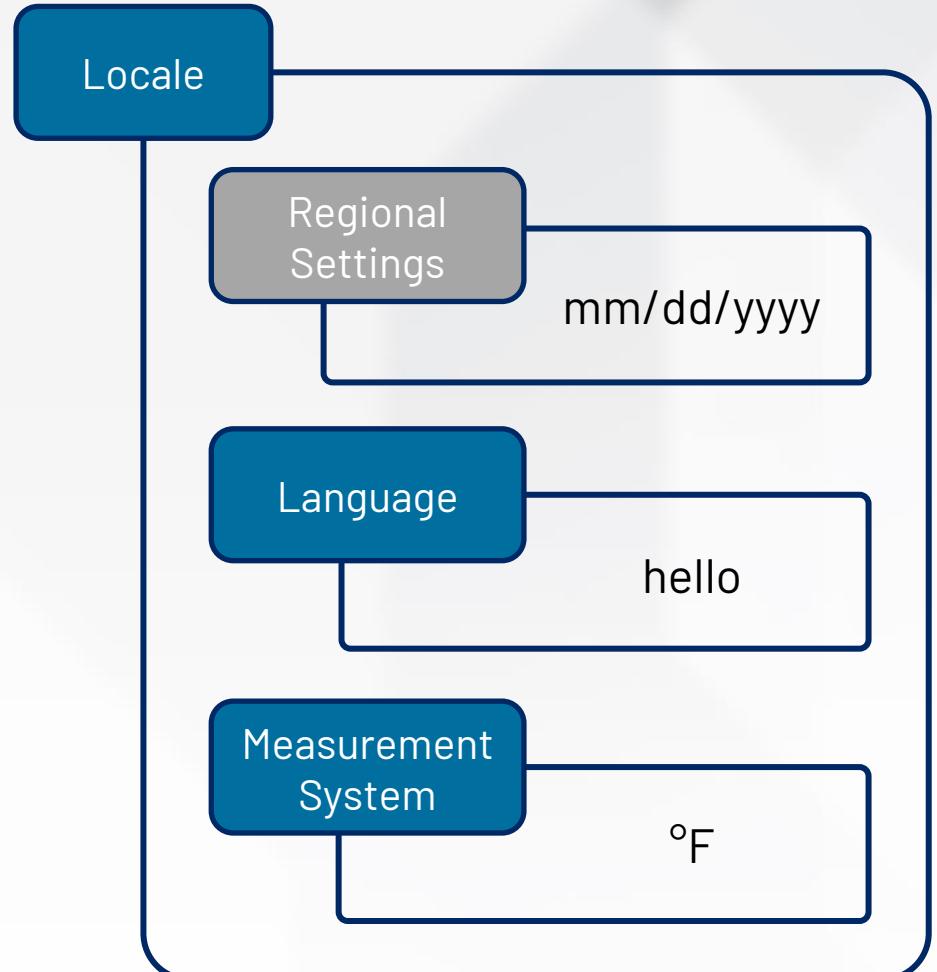
Locale

- Changing Locale will set altogether:

- Regional Settings
 - Language
 - Measurement System

- How to change the Locale at Runtime

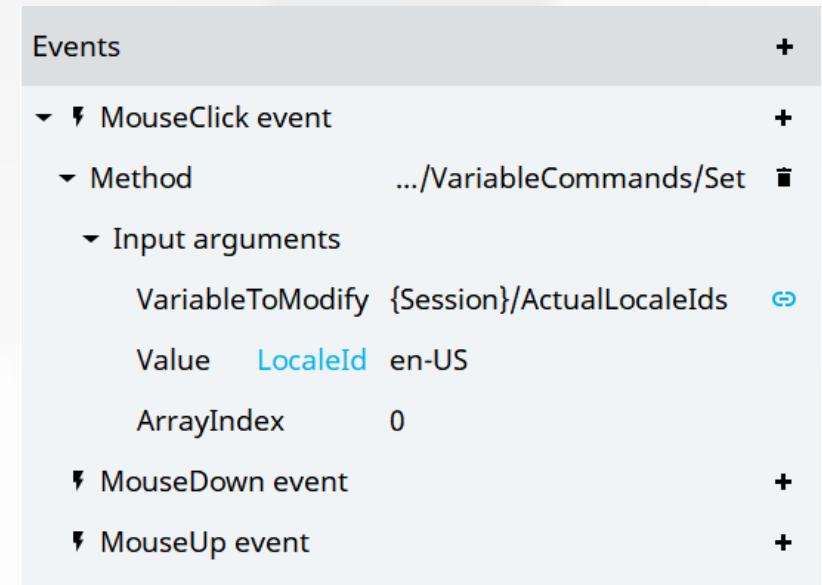
- Method
 - Command > Variable commands > Set variable value
 - Variable to Modify
 - Aliases > {Session} > Session > ActualLocaleId
 - Value
 - Select the Locale ISO code



- Language and Measurement System can be set independently from Locale

Locale

- Changing Locale will set altogether:
 - Regional Settings
 - Language
 - Measurement System
- How to change the Locale at Runtime
 - Method
 - Command > Variable commands > Set variable value
 - Variable to Modify
 - Aliases > {Session} > Session > ActualLocaleIds
 - Value
 - Select the Locale ISO code
- Language and Measurement System can be set independently from Locale



Languages - Translations

- The translation editor makes it possible to translate strings used in multilingual projects

- Translation Table**

- shows the translation of strings (Keys) in different languages (Locales)

- Translation Key References**

- shows all strings of the project that can be synched with the Translation Table (to become a key)

The screenshot shows the FactoryTalk Translation Table interface. On the left is a tree view of project components, with 'Translations' selected. The main area is a table titled 'Translation Table' with columns for 'Key' and 'en-US'. The table lists various project terms and their English equivalents.

Key	en-US
Acked	Acked
Acknowledge	Acknowledge
Acknowledge All	Acknowledge All
Active	Active
Apply	Apply
Confirm	Confirm
Confirm All	Confirm All
Confirmed	Confirmed
Delete	Delete
Failed	Failed
Load	Load
Message	Message
Name	Name
RecipeApply	Recipe applied

Languages - Translations

- The translation editor makes it possible to translate strings used in multilingual projects

• Translation Table

- shows the translation of strings (Keys) in different languages (Locales)

• Translation Key References

- shows all strings of the project that can be synched with the Translation Table (to become a key)

The screenshot shows the FactoryTalk Translation Key References interface. On the left is a 'Project view' sidebar with a search bar and a tree structure of project components: Model, Converters, Alarms, Recipes, Loggers, DataStores, Reports, OPC-UA, CommDrivers, NetLogic, Security, and Translations. Under Translations, 'LocalizationDictionary1' is selected. Below the tree are Retentivity and TranslationKeyReferences. The main area is titled 'Translation Key References' and contains a table with columns: String, Key, Path, and Synchronized. The table lists various keys and their corresponding paths and synchronization status.

String	Key	Path	Synchronized
Acked	Acked	MyProject/UI/panels/Screen2/...	<input checked="" type="checkbox"/>
Acknowledge	Acknowledge	MyProject/UI/panels/Screen2/...	<input checked="" type="checkbox"/>
Acknowledge All	Acknowledge All	MyProject/UI/panels/Screen2/...	<input checked="" type="checkbox"/>
Active	Active	MyProject/UI/panels/Screen2/...	<input checked="" type="checkbox"/>
Apply	Apply	MyProject/UI/panels/Screen1/...	<input checked="" type="checkbox"/>
Confirm	Confirm	MyProject/UI/panels/Screen2/...	<input checked="" type="checkbox"/>
Confirm All	Confirm All	MyProject/UI/panels/Screen2/...	<input checked="" type="checkbox"/>
Confirmed	Confirmed	MyProject/UI/panels/Screen2/...	<input checked="" type="checkbox"/>
Delete	Delete	MyProject/UI/panels/Screen1/...	<input checked="" type="checkbox"/>
Label1		MyProject/UI/panels/Screen3/...	<input type="checkbox"/>
Label2		MyProject/UI/panels/Screen3/...	<input type="checkbox"/>
Label3		MyProject/UI/panels/Screen3/...	<input type="checkbox"/>
Load	Load	MyProject/UI/panels/Screen1/...	<input checked="" type="checkbox"/>
Message	Message	MyProject/UI/panels/Screen2/...	<input checked="" type="checkbox"/>

Translation table

- Remove Translations

- to remove one or more keys from the table

- Add locale

1. Click to add a new language
2. Select the language in a standard Language-Country ISO code
3. Confirm adding the new language to the Project properties

- View Translation Reference

- switch to Key Reference table

Translation Table		Remove translations	Add locale	View Translation References
Key	↑ en-US			
Acked	Acked			
Acknowledge	Acknowledge			
Acknowledge All	Acknowledge All			
Active	Active			
Apply	Apply			
Confirm	Confirm			
Confirm All	Confirm All			
Confirmed	Confirmed			
Delete	Delete			

- Import/Export

- Translation Table can be exported and imported back using the "Translation Importer Exporter" script available in Template Library

Translation table

- Remove Translations

- to remove one or more keys from the table

- Add locale

1. Click to add a new language
2. Select the language in a standard Language-Country ISO code
3. Confirm adding the new language to the Project properties

- View Translation Reference

- switch to Key Reference table

The screenshot shows a 'Translation Table' interface. At the top right are buttons for 'Remove translations', 'Add locale', and 'View Translation References'. A dropdown menu titled 'Select the new locale...' is open, showing options: 'it-IT' (selected), 'it-CH', 'it-IT' (highlighted with a blue border), and 'iu-CA'. The main area is a table with columns 'Key' and 'en-US'. The table contains the following data:

Key	en-US
Acked	Acked
Acknowledge	Acknowledge
Acknowledge All	Acknowledge All
Active	Active
Apply	Apply
Confirm	Confirm
Confirm All	Confirm All
Confirmed	Confirmed
Delete	Delete

- Import/Export

- Translation Table can be exported and imported back using the "Translation Importer Exporter" script available in Template Library

Translation table

- Remove Translations

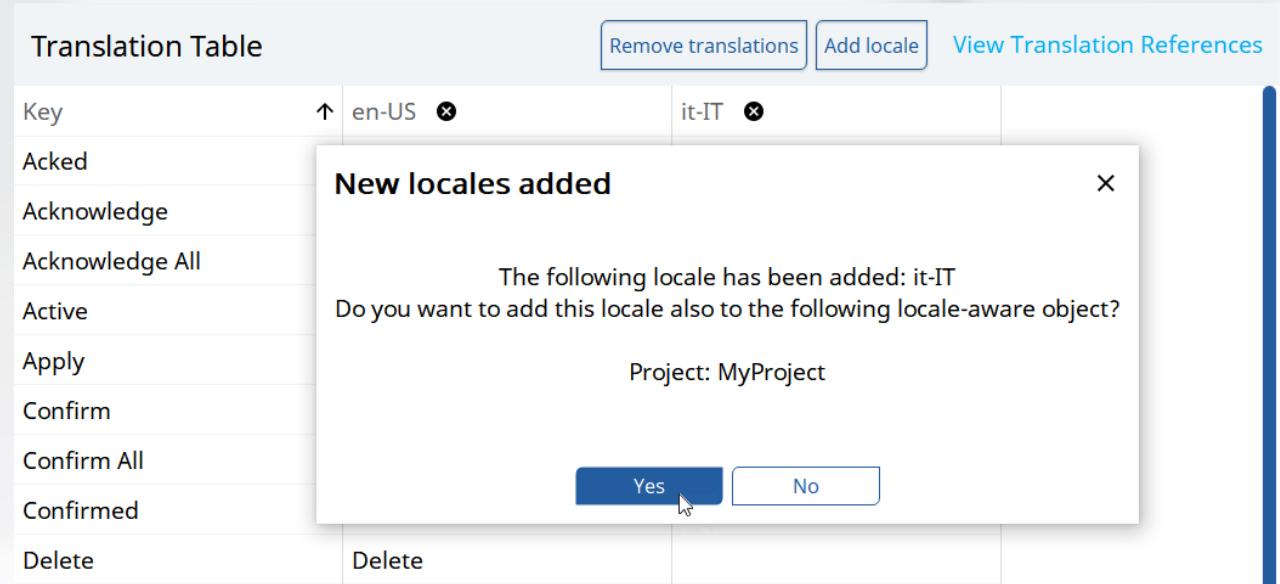
- to remove one or more keys from the table

- Add locale

1. Click to add a new language
2. Select the language in a standard Language-Country ISO code
3. Confirm adding the new language to the Project properties

- View Translation Reference

- switch to Key Reference table



- Import/Export

- Translation Table can be exported and imported back using the "Translation Importer Exporter" script available in Template Library

Translation key references

- Key Reference table

- shows all strings of the project

- Setting "Synchronized" to true

- creates a key into Translation Table with the same text

- copy the key text into the default language (en-US)

Translation Key References			
String	↑ Key	Path	Synchronized
Acked	Acked	MyProject/UI/Panels/Screen2/...	<input checked="" type="checkbox"/>
Acknowledge	Acknowledge	MyProject/UI/Panels/Screen2/...	<input checked="" type="checkbox"/>
Acknowledge All	Acknowledge All	MyProject/UI/Panels/Screen2/...	<input checked="" type="checkbox"/>
Active	Active	MyProject/UI/Panels/Screen2/...	<input checked="" type="checkbox"/>
Apply	Apply	MyProject/UI/Panels/Screen1/...	<input checked="" type="checkbox"/>
Confirm	Confirm	MyProject/UI/Panels/Screen2/...	<input checked="" type="checkbox"/>
Confirm All	Confirm All	MyProject/UI/Panels/Screen2/...	<input checked="" type="checkbox"/>
Confirmed	Confirmed	MyProject/UI/Panels/Screen2/...	<input checked="" type="checkbox"/>
Delete	Delete	MyProject/UI/Panels/Screen1/...	<input checked="" type="checkbox"/>
Label1		MyProject/UI/Panels/Screen3/...	<input type="checkbox"/>
Label2		MyProject/UI/Panels/Screen3/...	<input type="checkbox"/>
Label3		MyProject/UI/Panels/Screen3/...	<input type="checkbox"/>
Load	Load	MyProject/UI/Panels/Screen1/...	<input checked="" type="checkbox"/>
Message	Message	MyProject/UI/Panels/Screen2/...	<input checked="" type="checkbox"/>

Translation key references

- Key Reference table

- shows all strings of the project

- Setting "Synchronized" to true

- creates a key into Translation Table with the same text

- copy the key text into the default language (en-US)

Translation Key References			
String	↑ Key	Path	Synchronized
Acked	Acked	MyProject/UI/Panels/Screen2/...	<input checked="" type="checkbox"/>
Acknowledge	Acknowledge	MyProject/UI/Panels/Screen2/...	<input checked="" type="checkbox"/>
Acknowledge All	Acknowledge All	MyProject/UI/Panels/Screen2/...	<input checked="" type="checkbox"/>
Active	Active	MyProject/UI/Panels/Screen2/...	<input checked="" type="checkbox"/>
Apply	Apply	MyProject/UI/Panels/Screen1/...	<input checked="" type="checkbox"/>
Confirm	Confirm	MyProject/UI/Panels/Screen2/...	<input checked="" type="checkbox"/>
Confirm All	Confirm All	MyProject/UI/Panels/Screen2/...	<input checked="" type="checkbox"/>
Confirmed	Confirmed	MyProject/UI/Panels/Screen2/...	<input checked="" type="checkbox"/>
Delete	Delete	MyProject/UI/Panels/Screen1/...	<input checked="" type="checkbox"/>
Label1	Label1	MyProject/UI/Panels/Screen3/...	<input checked="" type="checkbox"/>
Label2		MyProject/UI/Panels/Screen3/...	<input type="checkbox"/>
Label3		MyProject/UI/Panels/Screen3/...	<input type="checkbox"/>
Load	Load	MyProject/UI/Panels/Screen1/...	<input checked="" type="checkbox"/>
Message	Message	MyProject/UI/Panels/Screen2/...	<input checked="" type="checkbox"/>

Translation usage

- Two ways to set the "starting" language:

1. Language or Locale Id of the "Starting user"
2. First language listed in the "Locales" project property

- How to change the language at Runtime

- Method
 - Command > Variable commands > Set variable value
- Variable to Modify
 - Aliases > {Session} > Session > ActualLanguages
- Value
 - Select the language ISO code

Properties	
Name	NativePresentationEngine
Type	Native presentation engine
Dynamic variables polling time	0000:00:00.100
Text render type	Auto
LoginWindow	
Start window	MainWindow (type) 🔗
Starting user	Anonymous 🔗
Style sheet	DefaultStyleSheet 🔗

Translation usage

- Two ways to set the "starting" language:

1. Language or Locale Id of the "Starting user"
2. First language listed in the "Locales" project property

- How to change the language at Runtime

- Method
 - Command > Variable commands > Set variable value
- Variable to Modify
 - Aliases > {Session} > Session > ActualLanguages
- Value
 - Select the language ISO code

Properties	
Name	NativePresentationEngine
Type	Native presentation engine
Dynamic variables polling time	0000:00:00.100
Text render type	Auto
LoginWindow	
Start window	MainWindow (type) 🔗
Starting user	User1 🔗
Style sheet	DefaultStyleSheet 🔗

Translation usage

- Two ways to set the "starting" language:
 1. Language or Locale Id of the "Starting user"
 2. First language listed in the "Locales" project property

Properties	
Name	User1
Type	User
Password	
LocaleIds	it-IT
Languages	
Measurement system	
Domain	

- How to change the language at Runtime
 - Method
 - Command > Variable commands > Set variable value
- Variable to Modify
 - Aliases > {Session} > Session > ActualLanguages
- Value
 - Select the language ISO code

Translation usage

- Two ways to set the "starting" language:
 1. Language or Locale Id of the "Starting user"
 2. First language listed in the "Locales" project property
- How to change the language at Runtime
 - Method
 - Command > Variable commands > Set variable value
 - Variable to Modify
 - Aliases > {Session} > Session > ActualLanguages
 - Value
 - Select the language ISO code

Properties	
Name	MyProject
Type	Project folder
Locales	it-IT;en-US
Translation fallback locales	en-US
Branching enabled	False
Measurement systems map	<i>Default mapping</i> 
Authentication mode	<i>Model only</i>
Default user folder	Users 

Translation usage

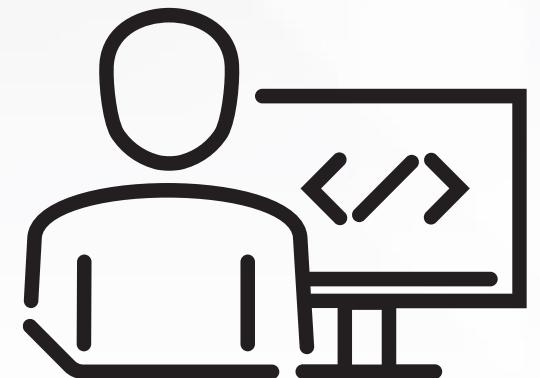
- Two ways to set the "starting" language:
 1. Language or Locale Id of the "Starting user"
 2. First language listed in the "Locales" project property
- How to change the language at Runtime
 - Method
 - Command > Variable commands > Set variable value
 - Variable to Modify
 - Aliases > {Session} > Session > ActualLanguages
 - Value
 - Select the language ISO code

Properties	
Name	MyProject
Type	Project folder
Locales	it-IT;en-US
Translation fallback locales	en-US
Branching enabled	False
Measurement systems map	Default mapping
Authentication mode	Model only
Default user folder	Users

Events	
▼	MouseClick event
▼	Method .../VariableCommands/Set
▼	Input arguments
VariableToModify	{Session}/ActualLanguages
Value	LocaleId en-US
ArrayIndex	0
▼	MouseDown event
▼	MouseUp event

Hands-on session

- Use some strings to label objects around the project
- Import strings into the Translation Table using the Key Reference Table then edit the translation manually
- Change the language at Runtime with the Emulator
- Optionally
 - try to export the Translation table,
 - modify with an external editor,
 - import back in Optix Studio



Measurement system

- Choose between:
 - International System of Units (SI)
 - US Customary measurement system
 - British Imperial units
- is enough to set the Unit of Measure on the Variable, then at Runtime, the conversion will be automatically applied depending on the Measurement System chosen



liter / quart



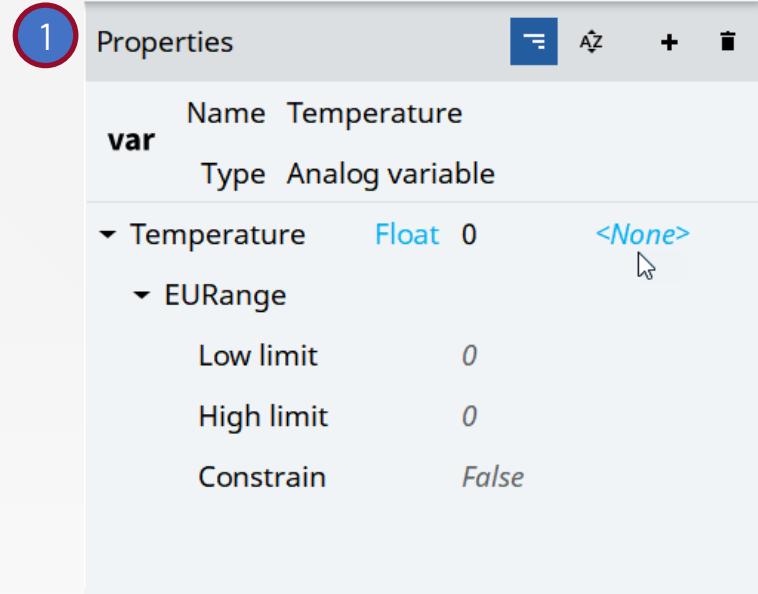
°C / °F



meter / yard

Localize an Analog Variable

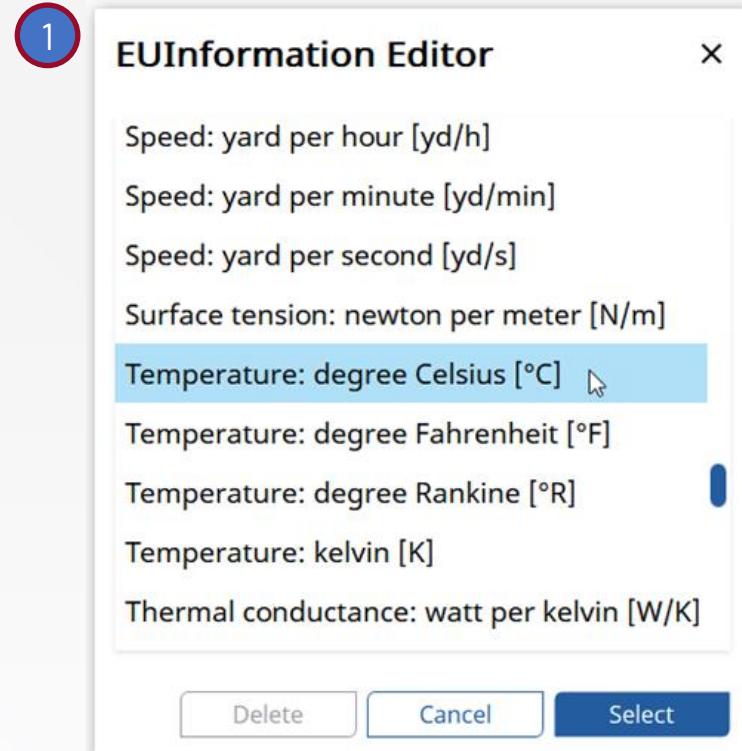
1. Define a new Analog Variable with Engineering Unit into the Model folder
 2. Add an Analog Variable to the object
 3. Into the object's Analog Variable set an Advanced Dynamic Link using "Binding mode between Source and Parent engineering unit"
 4. Link the object's value to the object's Analog Variable
- Unit of measure can be defined directly within a Comm.Driver Tag as well



Properties			
var	Name	Temperature	
	Type	Analog variable	
Temperature	Float	0	<None>
EURange			
Low limit		0	
High limit		0	
Constrain		False	

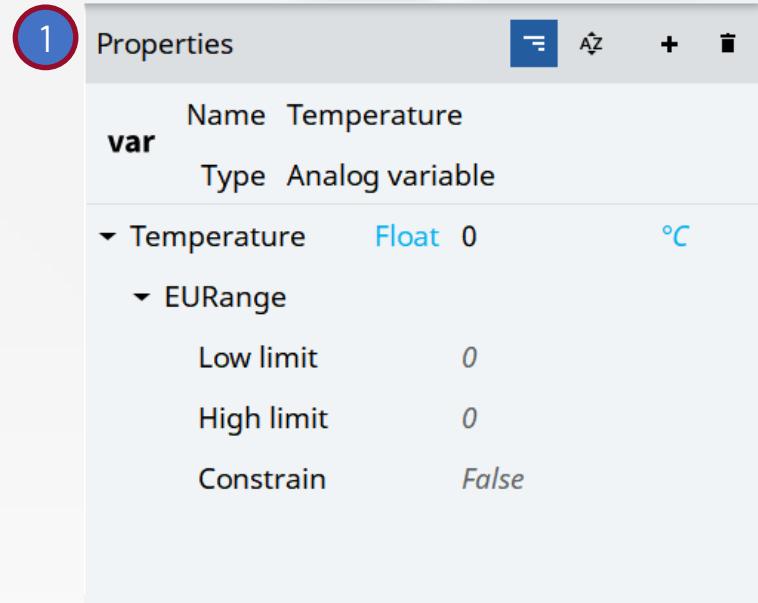
Localize an Analog Variable

1. Define a new Analog Variable with Engineering Unit into the Model folder
 2. Add an Analog Variable to the object
 3. Into the object's Analog Variable set an Advanced Dynamic Link using "Binding mode between Source and Parent engineering unit"
 4. Link the object's value to the object's Analog Variable
- Unit of measure can be defined directly within a Comm.Driver Tag as well



Localize an Analog Variable

1. Define a new Analog Variable with Engineering Unit into the Model folder
 2. Add an Analog Variable to the object
 3. Into the object's Analog Variable set an Advanced Dynamic Link using "Binding mode between Source and Parent engineering unit"
 4. Link the object's value to the object's Analog Variable
-
- Unit of measure can be defined directly within a Comm.Driver Tag as well



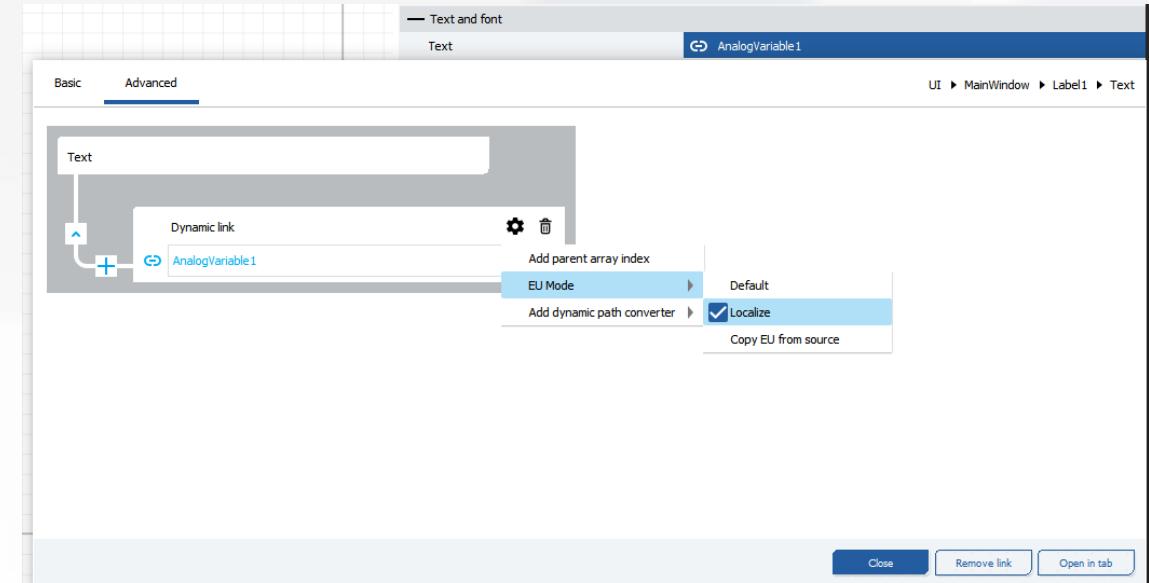
The screenshot shows the 'Properties' dialog box with one item selected, indicated by a red circle with the number '1'. The dialog has tabs for 'Properties', 'Script', 'Script Editor', and 'Script Log'. The 'Properties' tab is active, displaying the following information:

Name	Type	Value	Unit
var	Name	Temperature	
	Type	Analog variable	
Temperature	Float	0	°C
EURange			
Low limit		0	
High limit		0	
Constrain		False	

Localize an Analog Variable

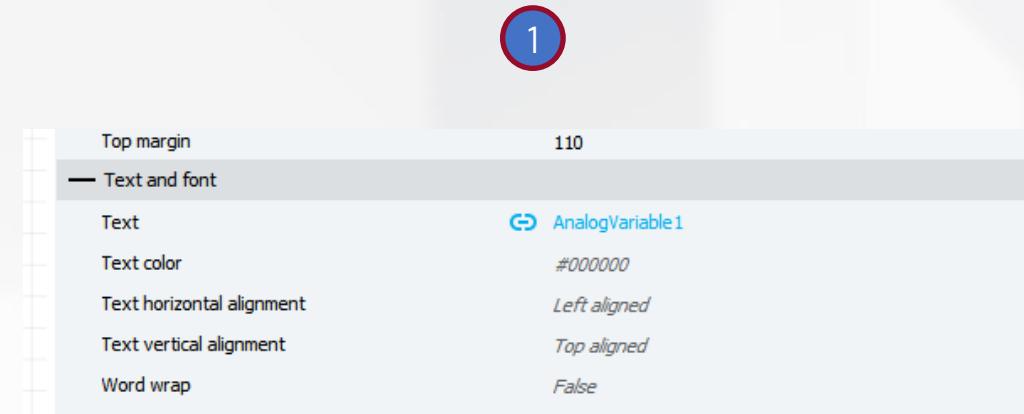
1. Define a new Analog Variable with Engineering Unit into the Model folder
 2. Add an Analog Variable to the object
 3. Into the object's Analog Variable set an Advanced Dynamic Link using "Binding mode between Source and Parent engineering unit"
 4. Link the object's value to the object's Analog Variable
-
- Unit of measure can be defined directly within a Comm.Driver Tag as well

2



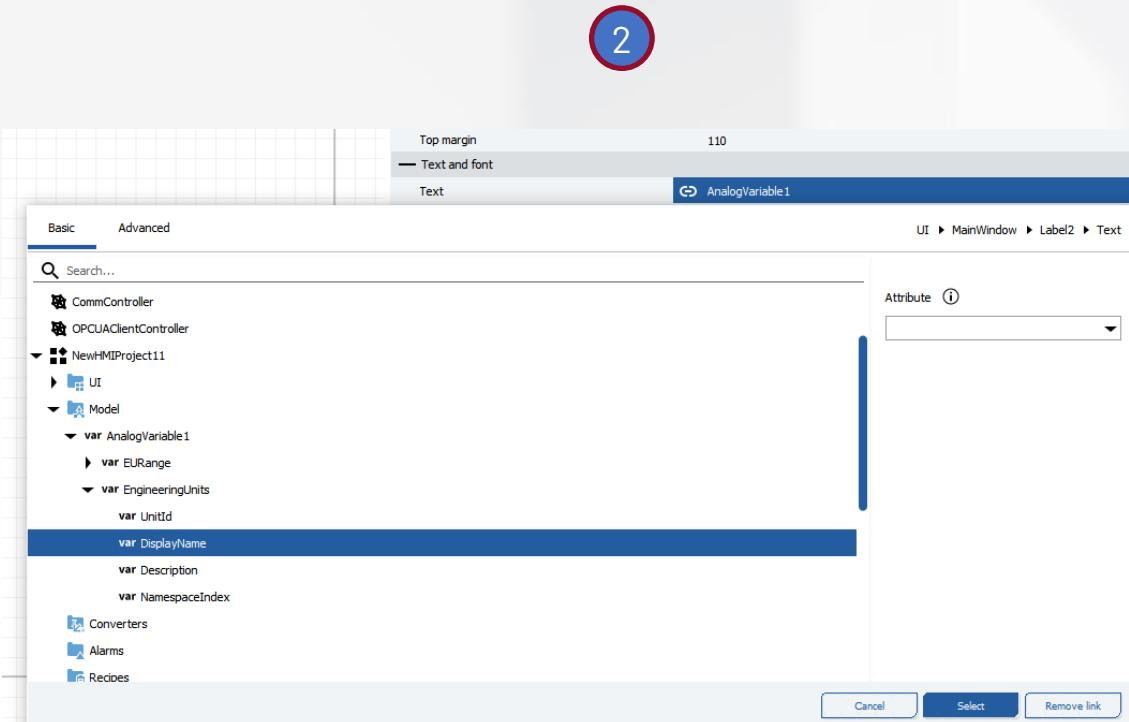
Display and localize the Engineering Unit

1. Add a label to the screen and create a DynamicLink from the Analog Variable to the Text property of the label
2. Open the DynamicLink popup, expand the Analog Variable and select Engineering Unit > Display Name
3. Reopen the DynamicLink popup, move to the Advanced tab and localize the Engineering Unit



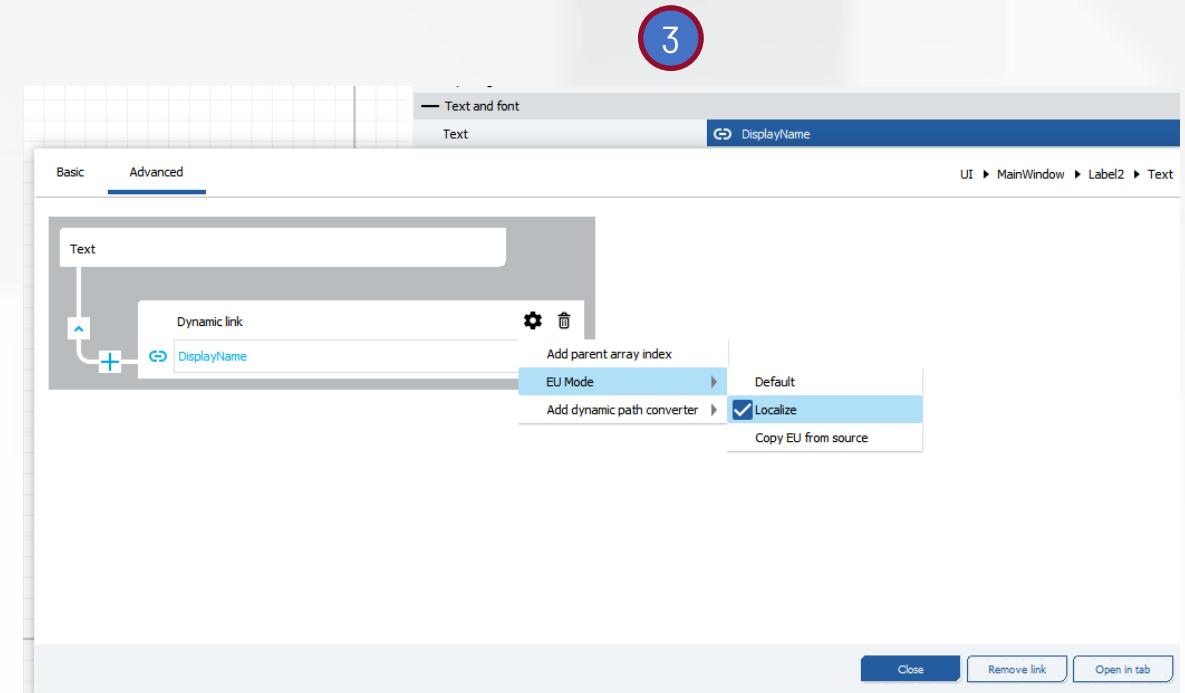
Display and localize the Engineering Unit

1. Add a label to the screen and create a DynamicLink from the Analog Variable to the Text property of the label
2. Open the DynamicLink popup, expand the Analog Variable and select Engineering Unit > Display Name
3. Reopen the DynamicLink popup, move to the Advanced tab and localize the Engineering Unit



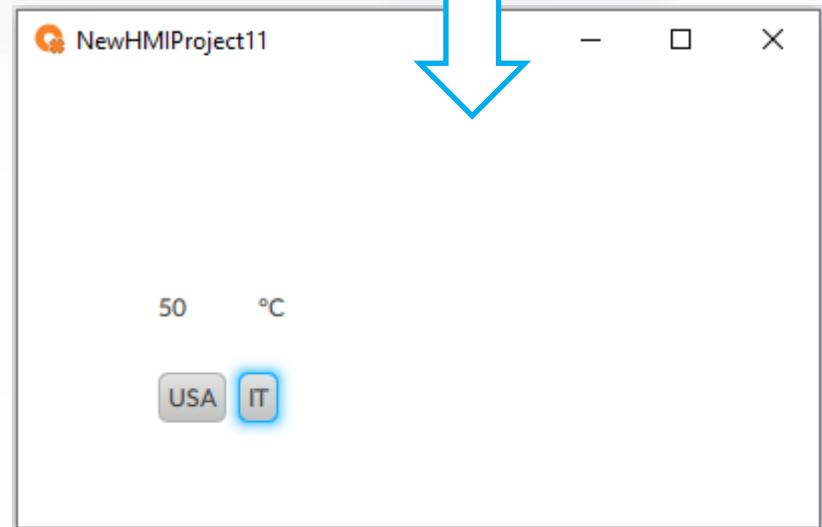
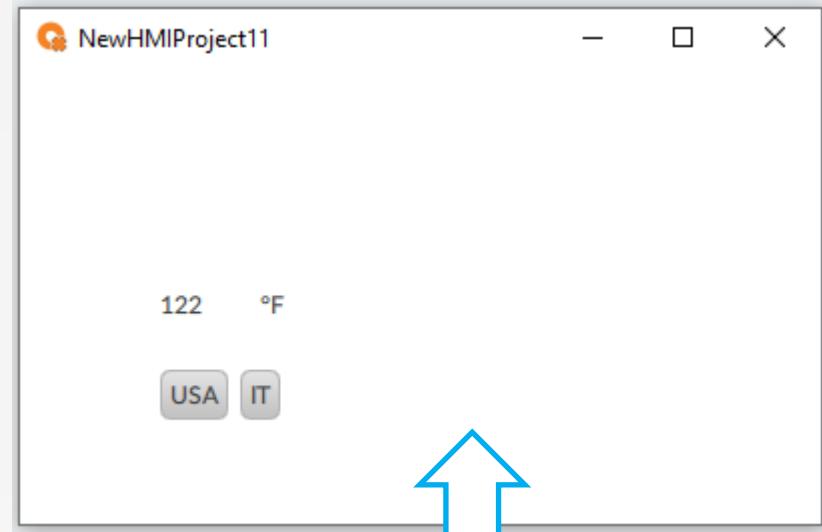
Display and localize the Engineering Unit

1. Add a label to the screen and create a DynamicLink from the Analog Variable to the Text property of the label
2. Open the DynamicLink popup, expand the Analog Variable and select Engineering Unit > Display Name
3. Reopen the DynamicLink popup, move to the Advanced tab and localize the Engineering Unit



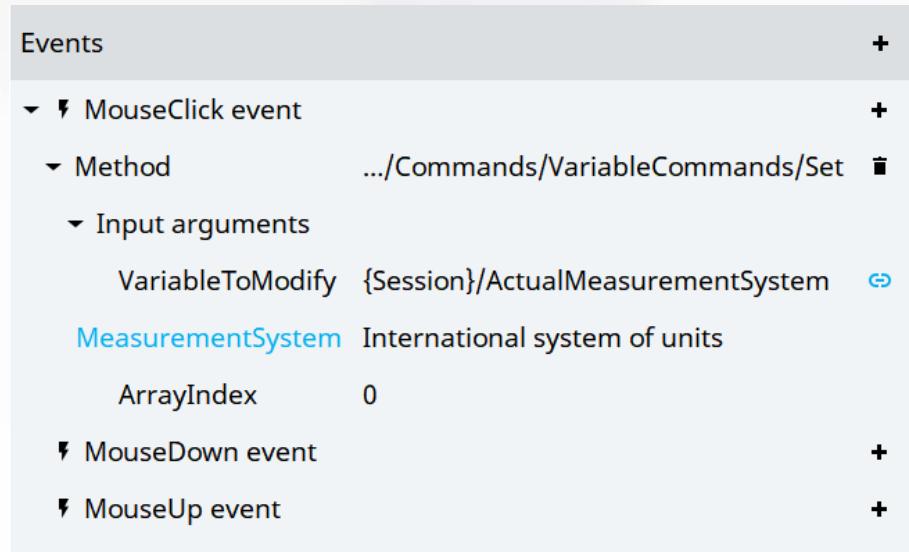
Test the conversion

1. Add two buttons to change the Measurement System
2. Execute the runtime and check the conversion



Set a measurement system

- How to change the Measurement System at Runtime
 - Method
 - Command > Variable commands > Set variable value
 - Variable To Modify
 - Aliases > {Session} > Session > ActualMeasurementSystem
 - Value
 - Select one of the available Measurement Systems



Hands-on session

- Set one or more Units of Measure to one or more variables
- Add a button to change the Measurement System (or the Locale)
- Check the conversion is applied



A large, abstract graphic consisting of a grid of blue dots arranged in a wave-like, undulating pattern across the entire background of the slide.

Work with NetLogic scripts



.NET

• What is .NET?

- .NET is a free, open-source development platform

• What is .NET Standard?

- .NET Standard is an API specification that lets you develop class libraries for multiple implementations of .NET

• What are .NET Framework, Mono and .NET Core?

- They are formally known as *implementations* of .NET Standard
- .NET Framework is the original implementation of .NET (runs only on Windows)
- Mono is used when a small runtime is required (formerly used by UNIQOO)
- .NET Core is the latest implementation and runs on any platform (used by OPTIX)

• What is C#?

- it's one of the programming languages supported by .NET

• What .NET version does Optix support?

- Optix 1.2.X uses supports .NET 6 (latest available LTS)
- Additional packages or libraries must be compatible with .NET 6 (NuGet packages for example)



Version	Start Date	End Date
.NET 8 LTS	Nov 14, 2023	2026
.NET 7	Nov 8, 2022	May 14, 2024
.NET 6.0 LTS	Nov 8, 2021	Nov 12, 2024
.NET 5.0	Nov 10, 2020	May 10, 2022
.NET 3.1 LTS	Dec 3, 2019	Dec 13, 2022
.NET 3.0	Sep 23, 2019	Mar 3, 2020
.NET 2.2	Dec 4, 2018	Dec 23, 2019
.NET 2.1 LTS	May 30, 2018	Aug 21, 2021
.NET 2.0	Aug 14, 2017	Oct 1, 2018
.NET 1.1	Nov 16, 2016	Jun 27, 2019
.NET 1.0	Jun 27, 2016	Jun 27, 2019

NetLogic

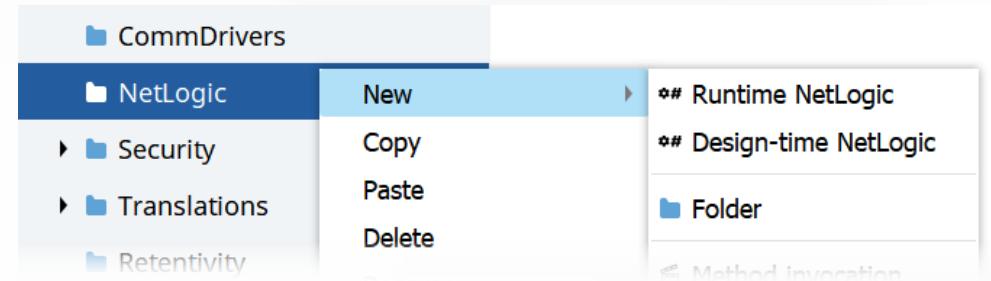
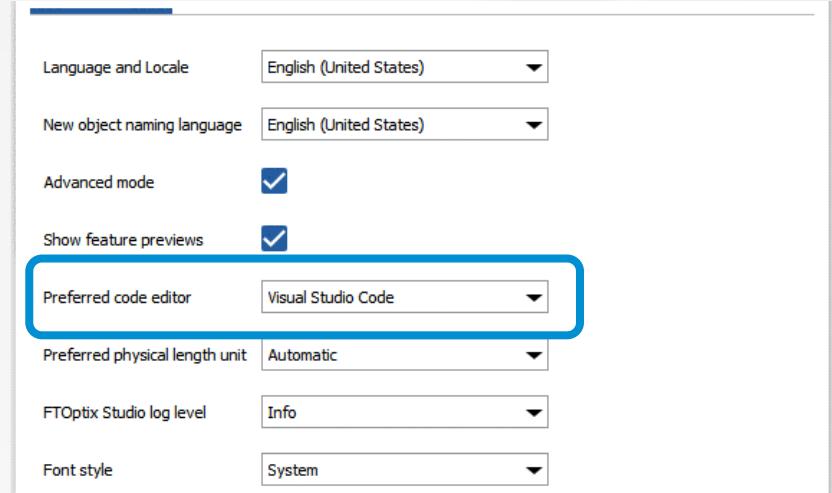
- it's an object that contains **C#** code compatible with **.NET Standard 6.0**

- Supported NetLogic Editors:

- Microsoft Visual Studio Code ([Download](#))
- Microsoft Visual Studio 2022 (no previous versions)

- Types of NetLogic:

- **Runtime**: to be executed at Runtime
- **Design-time**: to be executed in IDE



Design-time NetLogic

- **Where** can be placed?

- Anywhere!
But usually stay in NetLogic folder

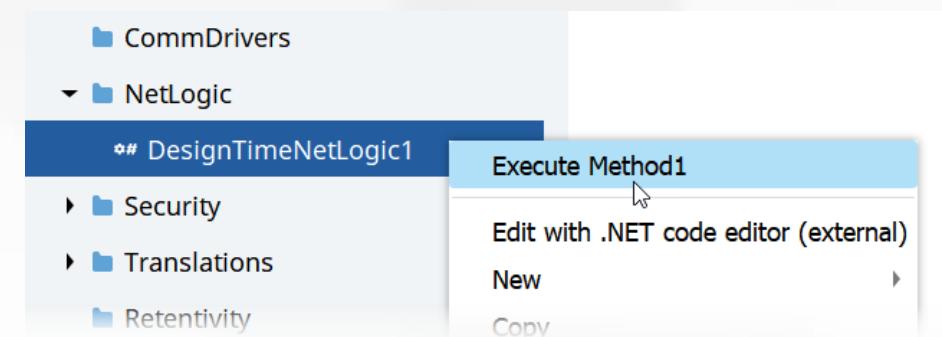
- **When** is the method executed?

- When the "exported method"
is executed/called from Studio

- Notes on "[ExportMethod]":

- allow exposing the method outside the script
- must be used before every method
that need to be exposed
- exported methods, must be "public"

```
public class DesignTimeNetLogic1 : BaseNetLogic
{
    [ExportMethod]
    public void Method1()
    {
        // Insert code to be executed by the method
    }
}
```



Design-time NetLogic: create simple variables

```
[ExportMethod]
public void CreateSimpleVariables()
{
    var myFolder = Project.Current.Get<Folder>("Model/Variables");
    if (myFolder!=null)
    {
        myFolder.Delete();
    }
    myFolder = InformationModel.Make<Folder>("Variables");
    Project.Current.Get("Model").Add(myFolder);
    for (int i=1; i<=3; i++)
    {
        var myVar = InformationModel.MakeVariable("Variable" + i,OpcUa.DataTypes.UInt16);
        myFolder.Add(myVar);
    }
}
```

Design-time NetLogic: create Motor Type and instances

```
[ExportMethod]
public void CreateMotorType()
{
    var motorType = InformationModel.MakeObjectType("MotorType");
    var speed = InformationModel.MakeVariable("Speed", OpcUa.DataTypes.UInt16);
    var running = InformationModel.MakeVariable("Running", OpcUa.DataTypes.Boolean);
    motorType.Add(speed);
    motorType.Add(running);
    Project.Current.Get("Model").Add(motorType);
}

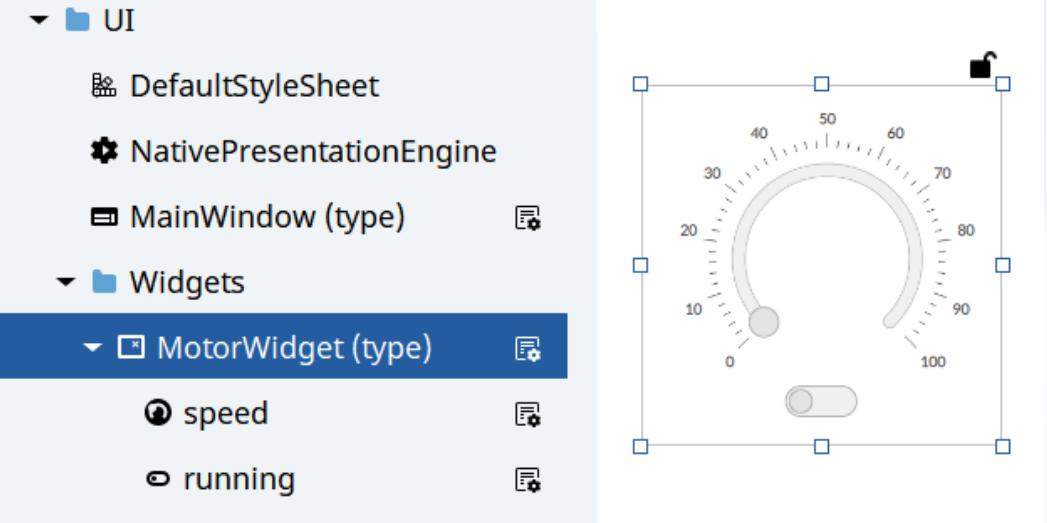
[ExportMethod]
public void CreateMotorInstances()
{
    var motorInstancesFolder = InformationModel.Make<Folder>("MotorInstances");
    Project.Current.Get("Model").Add(motorInstancesFolder);
    for (int i = 1; i <= 3; i++)
    {
        var motorType = Project.Current.Get("Model/MotorType");
        var motorInstance = InformationModel.MakeObject("Motor" + i, motorType.NodeId);
        motorInstancesFolder.Add(motorInstance);
    }
}
```

Design-time NetLogic: create widget instances

```
[ExportMethod]
public void CreateMotorWidgetInstances()
{
    // NOTE: the "MotorWidget" type with "MotorAlias" alias,
    // must be already available at UI/Widgets

    for (int i = 1; i <= 3; i++)
    {
        var motorWidgetType = Project.Current.Get("UI/Widgets/MotorWidget");
        var motorWidgetInstance = InformationModel.MakeObject("MotorWidget" + i, motorWidgetType.NodeId);
        Project.Current.Get("UI/MainWindow").Children.Add(motorWidgetInstance);

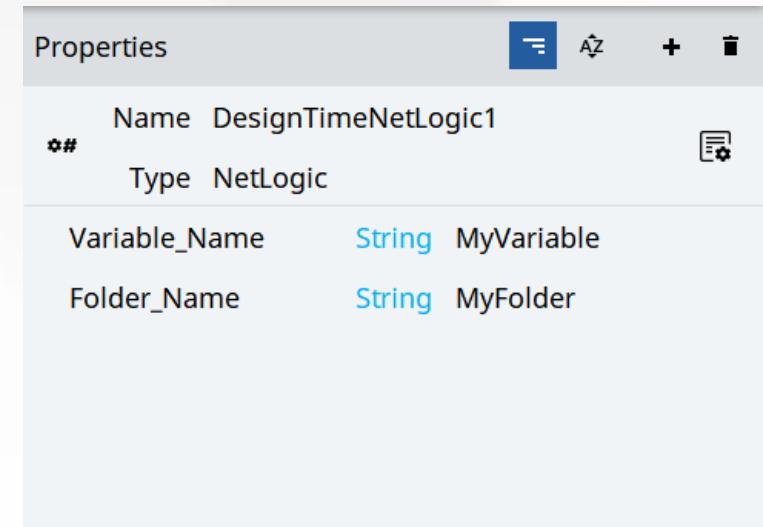
        var motorWidget = Project.Current.Get<Panel>("UI/MainWindow/MotorWidget"+i);
        motorWidget.SetAlias("MotorAlias", Project.Current.Get("Model/MotorInstances/Motor" + i));
        motorWidget.LeftMargin = (motorWidget.Width + 50) * (i - 1);
    }
}
```



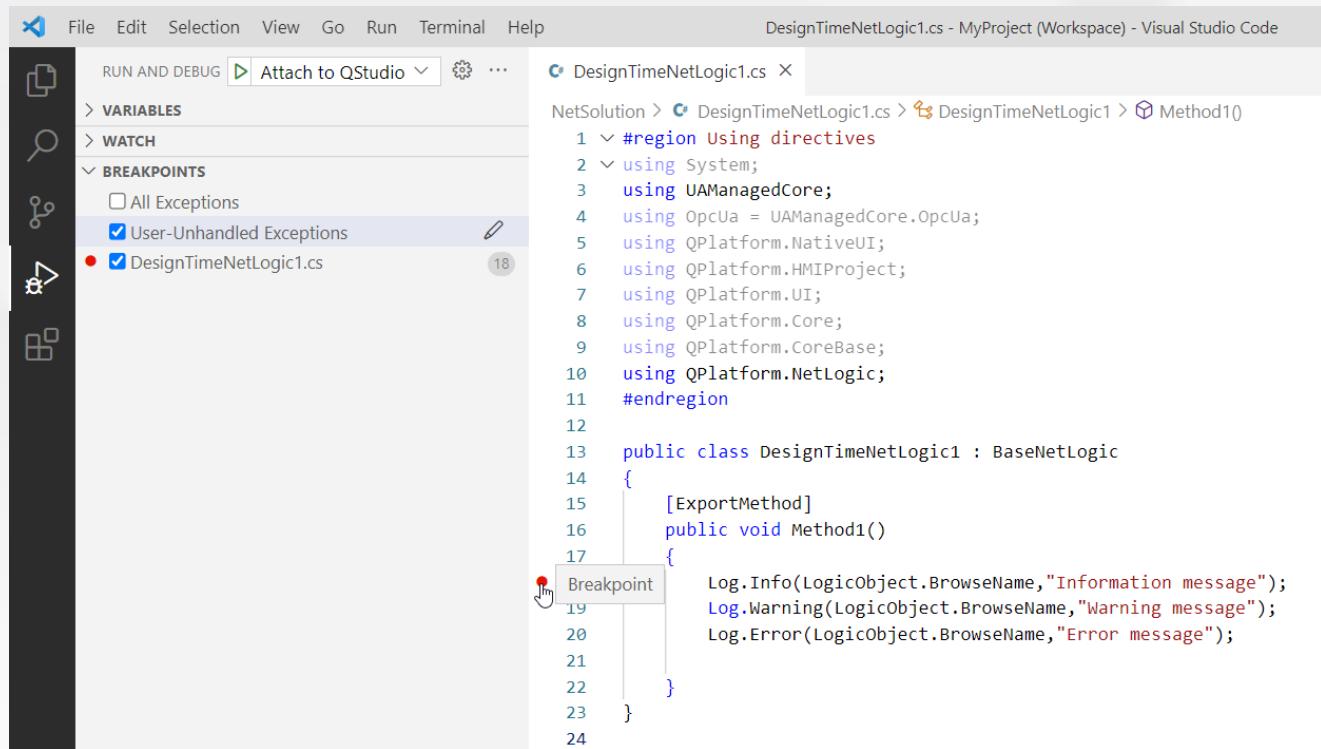
Design-time NetLogic: script parameters

- Variables added to the NetLogic object behaves like script parameters
- Value of these variables can be read using LogicObject.GetVariable("{variable}").Value

```
[ExportMethod]
public void ReadParameters()
{
    string varName = LogicObject.GetVariable("Variable_Name").Value;
    string folderName = LogicObject.GetVariable("Folder_Name").Value;
    Log.Info(LogicObject.BrowseName, "Variable_Name = " + varName);
    Log.Info(LogicObject.BrowseName, "Folder_Name = " + folderName);
}
```



Debug with visual studio



The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** DesignTimeNetLogic1.cs - MyProject (Workspace) - Visual Studio Code.
- Sidebar:** RUN AND DEBUG, Attach to QStudio, Variables, Watch, Breakpoints. Under Breakpoints, "User-Unhandled Exceptions" and "DesignTimeNetLogic1.cs" are selected.
- Code Editor:** The code for DesignTimeNetLogic1.cs is displayed. A red circular breakpoint icon is visible on the left margin at line 19, indicating a breakpoint is set there.
- Code:**

```
1  #region Using directives
2  using System;
3  using UAManagedCore;
4  using OpcUa = UAManagedCore.OpcUa;
5  using QPlatform.NativeUI;
6  using QPlatform.HMIPrject;
7  using QPlatform.UI;
8  using QPlatform.Core;
9  using QPlatform.CoreBase;
10 using QPlatform.NetLogic;
11 #endregion
12
13 public class DesignTimeNetLogic1 : BaseNetLogic
14 {
15     [ExportMethod]
16     public void Method1()
17     {
18         Log.Info(LogicObject.BrowseName, "Information message");
19         Log.Warning(LogicObject.BrowseName, "Warning message");
20         Log.Error(LogicObject.BrowseName, "Error message");
21     }
22 }
23
24 }
```

- Set a breakpoint by clicking to the left of the row number
- Select
 - Attach to Studio for Design time NetLogic
 - attach to Runtime for Runtime NetLogic
- Click "Run and Debug" or F5
- Execute the script
- Execute step-by-step
 - Step Over (F10)
 - Step Into (F11)

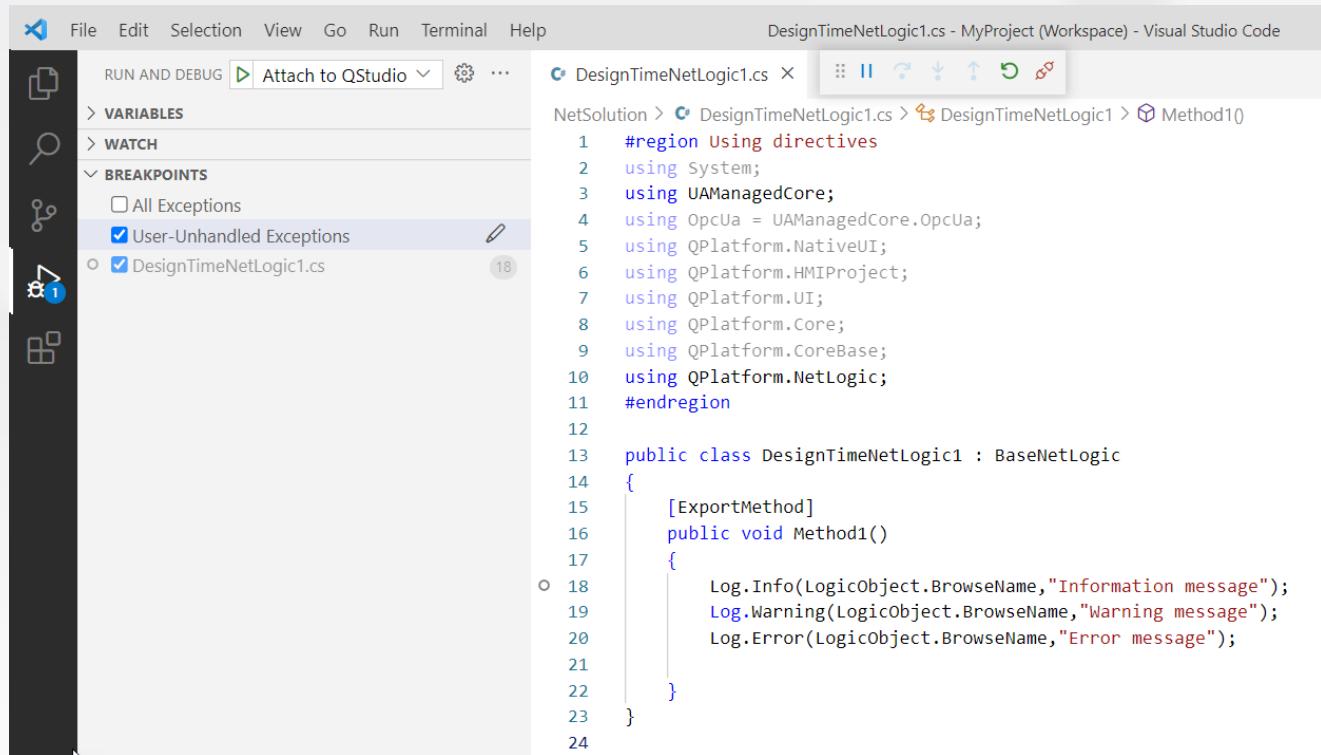
Debug with visual studio

```
DesignTimeNetLogic1.cs - MyProject (Workspace) - Visual Studio Code
DesignTimeNetLogic1.cs

1  #region Using directives
2  using System;
3  using UAManagedCore;
4  using OpcUa = UAManagedCore.OpcUa;
5  using QPlatform.NativeUI;
6  using QPlatform.HMIPrject;
7  using QPlatform.UI;
8  using QPlatform.Core;
9  using QPlatform.CoreBase;
10 using QPlatform.NetLogic;
11 #endregion
12
13 public class DesignTimeNetLogic1 : BaseNetLogic
14 {
15     [ExportMethod]
16     public void Method1()
17     {
18         Log.Info(LogicObject.BrowseName, "Information message");
19         Log.Warning(LogicObject.BrowseName, "Warning message");
20         Log.Error(LogicObject.BrowseName, "Error message");
21     }
22 }
23
24 }
```

- Set a breakpoint by clicking to the left of the row number
- Select
 - Attach to Studio for Design time NetLogic
 - attach to Runtime for Runtime NetLogic
- Click "Run and Debug" or F5
- Execute the script
- Execute step-by-step
 - Step Over (F10)
 - Step Into (F11)

Debug with visual studio



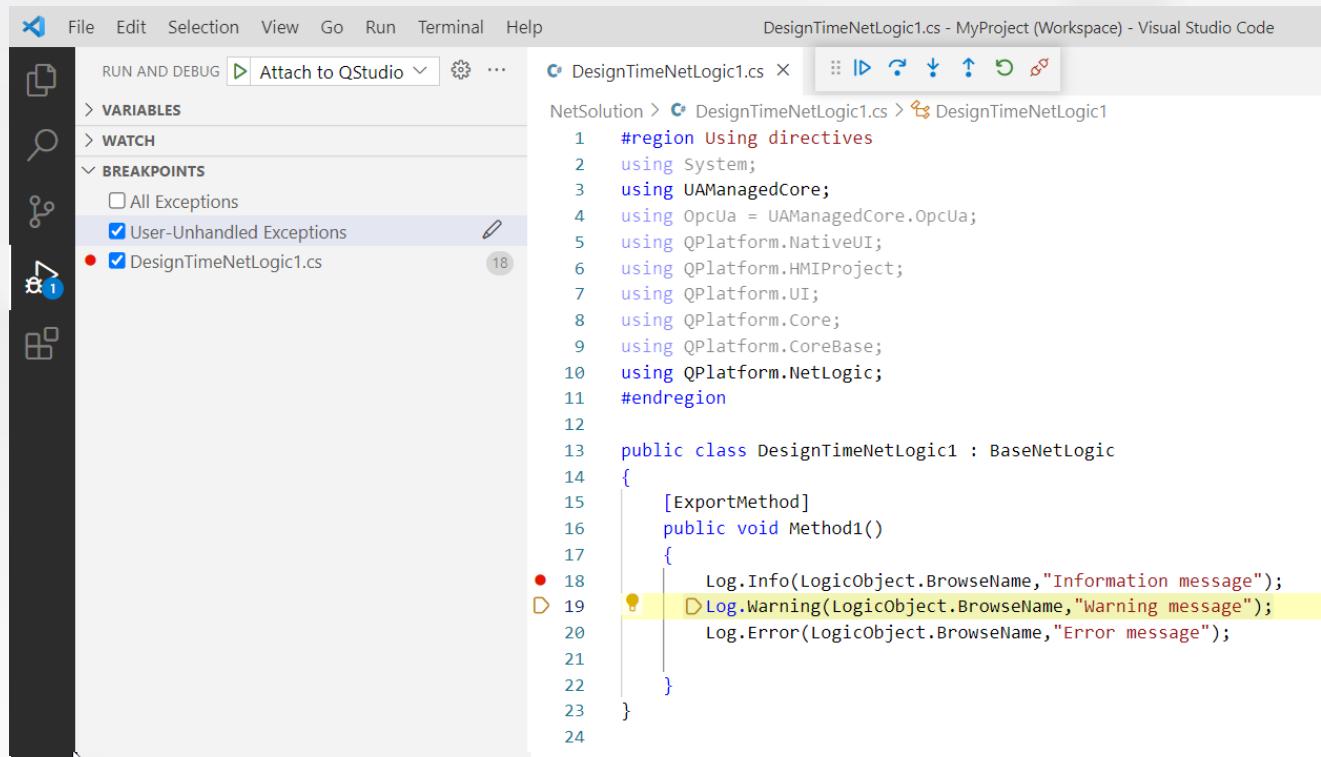
The screenshot shows the Visual Studio Code interface with the following details:

- Toolbar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Breakpoints:** A dropdown menu showing "RUN AND DEBUG" and "Attach to QStudio".
- Breakpoints Tool Bar:** Variables, Watch, Breakpoints. Under Breakpoints, "User-Unhandled Exceptions" is selected.
- Code Editor:** File: DesignTimeNetLogic1.cs - MyProject (Workspace) - Visual Studio Code. The code is as follows:

```
1  #region Using directives
2  using System;
3  using UAManagedCore;
4  using OpcUa = UAManagedCore.OpcUa;
5  using QPlatform.NativeUI;
6  using QPlatform.HMIProject;
7  using QPlatform.UI;
8  using QPlatform.Core;
9  using QPlatform.CoreBase;
10 using QPlatform.NetLogic;
11 #endregion
12
13 public class DesignTimeNetLogic1 : BaseNetLogic
14 {
15     [ExportMethod]
16     public void Method1()
17     {
18         Log.Info(LogicObject.BrowseName, "Information message");
19         Log.Warning(LogicObject.BrowseName, "Warning message");
20         Log.Error(LogicObject.BrowseName, "Error message");
21     }
22 }
23
24 }
```

- Set a breakpoint by clicking to the left of the row number
- Select
 - Attach to Studio for Design time NetLogic
 - attach to Runtime for Runtime NetLogic
- Click "Run and Debug" or F5
- Execute the script
- Execute step-by-step
 - Step Over (F10)
 - Step Into (F11)

Debug with visual studio



The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** DesignTimeNetLogic1.cs - MyProject (Workspace) - Visual Studio Code.
- Sidebar:** RUN AND DEBUG, Attach to QStudio, Variables, Watch, BREAKPOINTS (selected), All Exceptions, User-Unhandled Exceptions, DesignTimeNetLogic1.cs (with a red dot).
- Code Editor:** DesignTimeNetLogic1.cs file content:

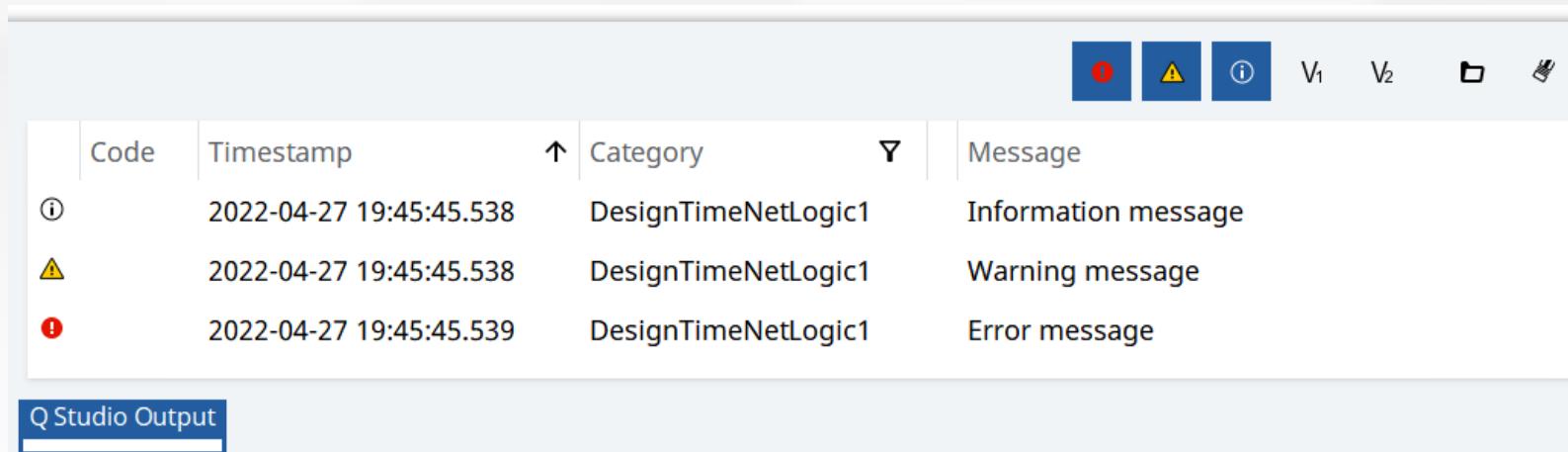
```
1  #region Using directives
2  using System;
3  using UAManagedCore;
4  using OpcUa = UAManagedCore.OpcUa;
5  using QPlatform.NativeUI;
6  using QPlatform.HMIPrject;
7  using QPlatform.UI;
8  using QPlatform.Core;
9  using QPlatform.CoreBase;
10 using QPlatform.NetLogic;
11 #endregion
12
13 public class DesignTimeNetLogic1 : BaseNetLogic
14 {
15     [ExportMethod]
16     public void Method1()
17     {
18         Log.Info(LogicObject.BrowseName,"Information message");
19         Log.Warning(LogicObject.BrowseName,"Warning message");
20         Log.Error(LogicObject.BrowseName,"Error message");
21     }
22 }
23
24 }
```

A yellow rectangle highlights the line `Log.Warning(LogicObject.BrowseName,"Warning message");` which contains a yellow breakpoint icon.

- Set a breakpoint by clicking to the left of the row number
- Select
 - Attach to Studio for Design time NetLogic
 - attach to Runtime for Runtime NetLogic
- Click "Run and Debug" or F5
- Execute the script
- Execute step-by-step
 - Step Over (F10)
 - Step Into (F11)

Debug by logging to output

- Log.Info(LogicObject.BrowseName,"Information message");
- Log.Warning(LogicObject.BrowseName,"Warning message");
- Log.Error(LogicObject.BrowseName,"Error message");



The screenshot shows the 'Studio Output' window with the following data:

Code	Timestamp	Category	Message
ⓘ	2022-04-27 19:45:45.538	DesignTimeNetLogic1	Information message
⚠	2022-04-27 19:45:45.538	DesignTimeNetLogic1	Warning message
❗	2022-04-27 19:45:45.539	DesignTimeNetLogic1	Error message

NetLogic don'ts !

The screenshot shows a Visual Studio Code interface. On the left is the Explorer sidebar with a tree view of a project named 'MYPROJECT (WORKSPACE)'. Items listed include 'NetSolution', '.vscode', 'bin', 'obj', 'Private', 'DesignTimeNetLogic1.cs' (which is selected and highlighted with a red circle containing the number '2'), 'MyProject.code-workspace', 'MyProject.csproj', 'MyProject.sln', and 'RuntimeNetLogic1.cs'. In the center is the code editor window titled 'DesignTimeNetLogic1.cs'. The code is as follows:

```
1  #region Using directives
2  using System;
3  using UAManagerCore;
4  using OpcUa = UAManagerCore.OpcUa;
5  using QPlatform.NativeUI;
6  using QPlatform.HMIPrjct;
7  using QPlatform.UI;
8  using QPlatform.Core;
9  using QPlatform.CoreBase;
10 using QPlatform.NetLogic;
11 #endregion
12 
13 public class DesignTimeNetLogic1 : BaseNetLogic
14 {
15     [ExportMethod]
16     public void Method1()
17     {
18         // Insert code to be executed by the method
19     }
20 }
21
```

Annotations are present: a red circle with the number '1' highlights the 'DesignTimeNetLogic1' class name in line 13; another red circle with the number '2' highlights the 'DesignTimeNetLogic1.cs' file in the Explorer sidebar.

1. NetLogic class name is the same of the NetLogic object name
 - **Don't change the name from the source code!**
 - Rename it from Studio instead!
2. NetLogic objects are also browsable from the NetSolution
 - **Don't delete a NetLogic from the solution!**
 - Remove it from Studio instead!

Cast vs Type parameter

- Use of a "**type parameter**"

- Type parameter can be used with "generic methods" that allows using different types
- a generic method can be called either without an argument
or by specifying the type of argument within angle brackets

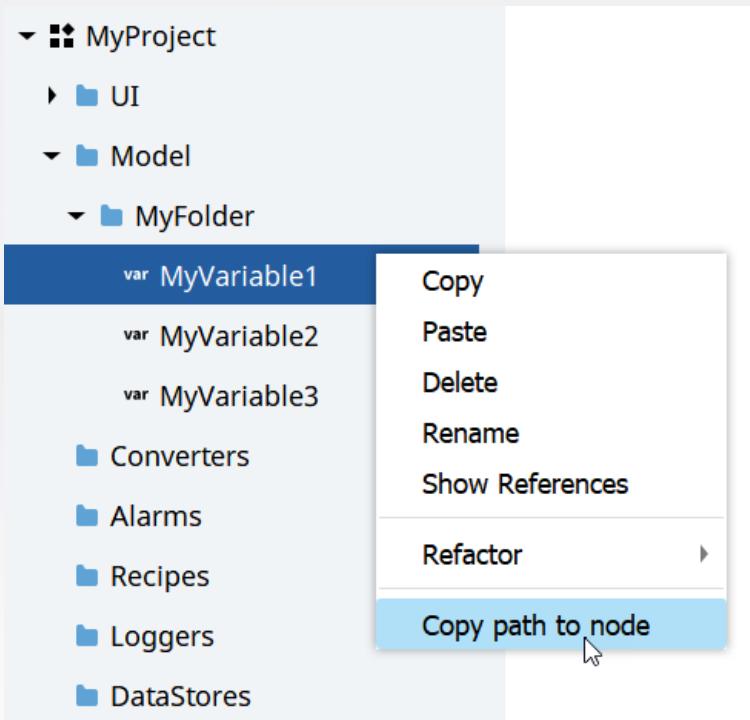
```
var myfolder = Project.Current.Get<Folder>("Model\MyFolder");
```

- Use of a "**cast**"

- a cast is an explicit conversion
- To perform a cast, specify the type that you are casting to in parentheses in front of the value or variable to be converted.

```
var myfolder = (Folder)Project.Current.Get("Model\MyFolder");
```

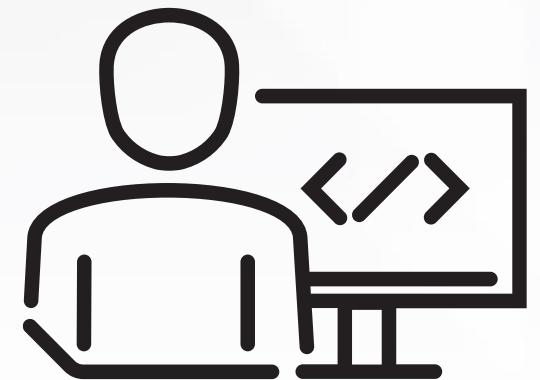
Copy path to node



- Can be useful when using absolute addressing with `Project.Current.Get`
- Provide the absolute path of the node into clipboard
 - "MyProject/Model/MyFolder/MyVariable1"
 - `Project.Current.GetVariable("Model/MyFolder/MyVariable1");`
- NOTE: remember to remove the project name

Hands-on session

- Write one or more Design-Time script
- Try to debug



Runtime Netlogic

- **Where** can be placed?

- into the NetLogic folder
- into the Main Window
- into a Panel
- into a graphical object (like a Button)

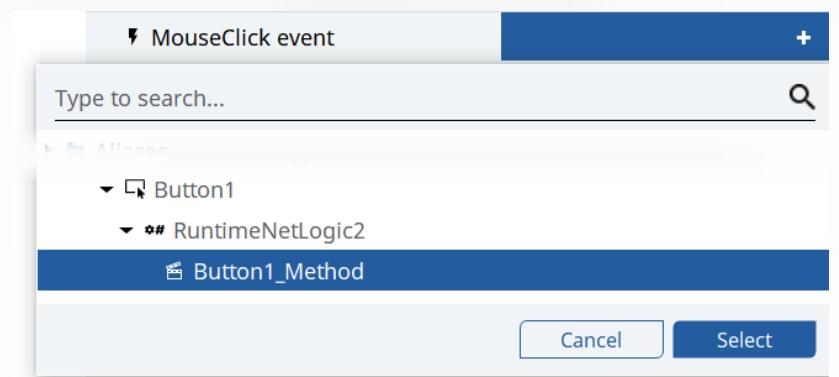
- **When** are methods executed?

- *Start*: automatically executed when the node that contain the NetLogic is created/opened
- *Stop*: automatically executed when the node that contain the NetLogic is destroyed/closed
- *Exported Method*: executed when called by an Event

```
public class RuntimeNetLogic1 : BaseNetLogic
{
    public override void Start()
    {
        // code executed when the logic is started
    }

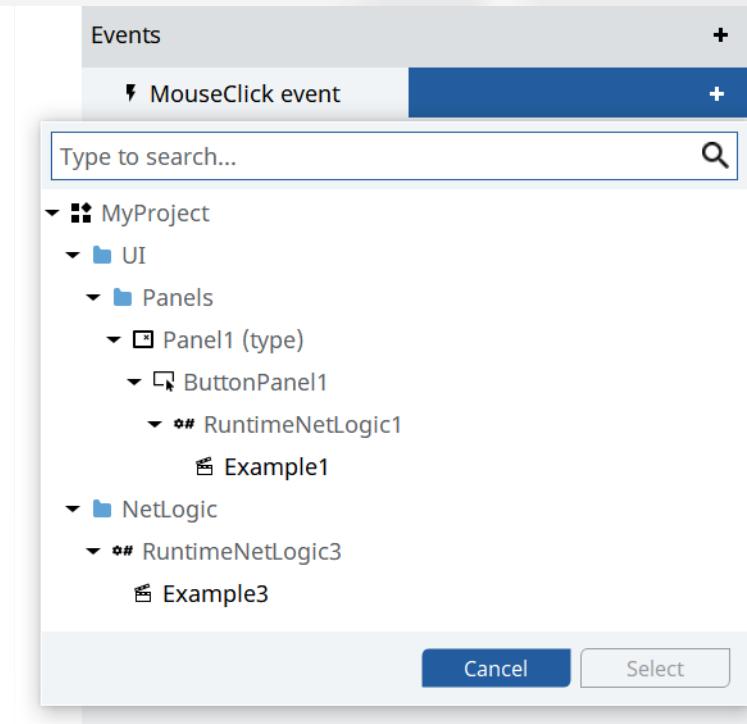
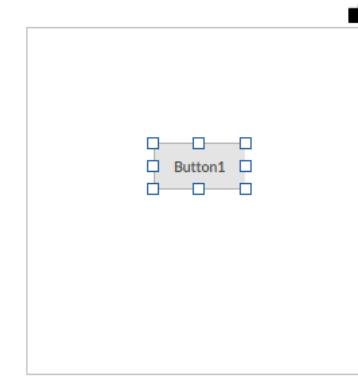
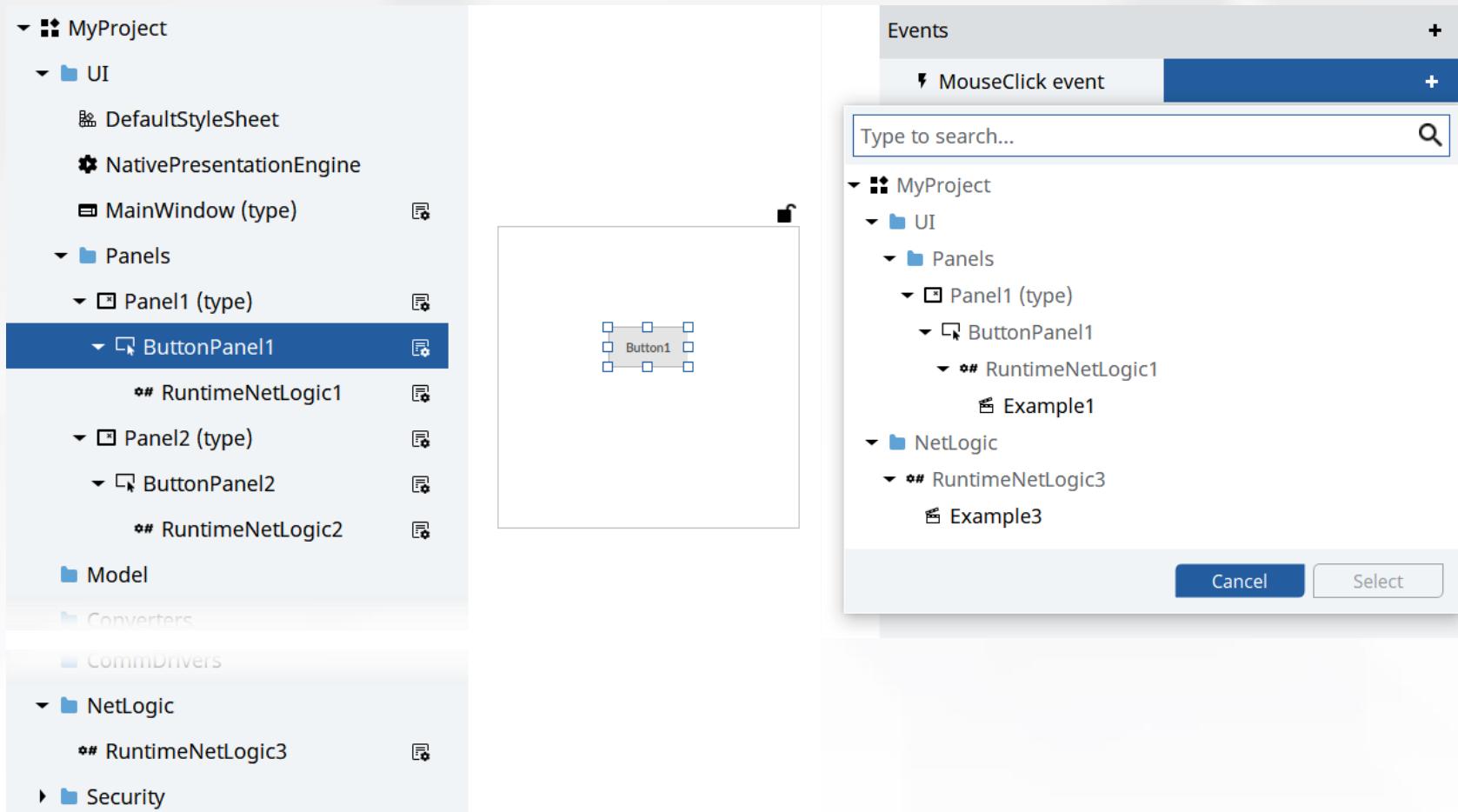
    public override void Stop()
    {
        // code executed when the logic is stopped
    }

    [ExportMethod]
    public void Button1_Method()
    {
        // code executed by the method
    }
}
```



Runtime NetLogic: scope

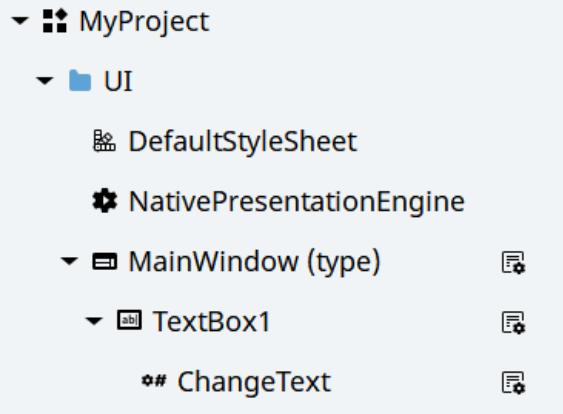
- Button1 of Panel1 cannot access to method exposed by RuntimeNetLogic2 because is related to Panel2.
- Panel2 and therefore RuntimeNetLogic2 do not exists when Button1 is loaded



Runtime NetLogic: usage of "owner"

```
public class ChangeText : BaseNetLogic
{
    [ExportMethod]
    public void ChangeTextRelativePath()
    {
        // WORKING AT RUNTIME!
        Owner.GetVariable("Text").Value = "Text changed using Owner";
    }

    [ExportMethod]
    public void ChangeTextAbsolutePath()
    {
        // NOT WORKING AT RUNTIME!
        Project.Current.Get<TextBox>("UI/MainWindow/TextBox1").Text = "Text changed using Project.Current";
    }
}
```



	Project.Current...	Owner...
Type of path	Absolute	Relative
Runtime usage	can be used to refer nodes <u>outside the UI</u> *	must be used to refer nodes <u>inside the UI</u> *

* This is due to the UI Sessions, dynamically created at Runtime

Runtime NetLogic: read/write variables value



```
[ExportMethod]
public void Sum()
{
    var addend1 = (Int32)Project.Current.GetVariable("Model/Variable1").Value;
    var addend2 = (Int32)Project.Current.GetVariable("Model/Variable2").Value;
    Project.Current.GetVariable("Model/Variable3").Value = addend1 + addend2;
}
```

Runtime NetLogic: subscribe to VariableChange event

```
private IUAVariable addend1;
private IUAVariable addend2;

public override void Start()
{
    addend1 = Project.Current.GetVariable("Model/Variable1");
    addend2 = Project.Current.GetVariable("Model/Variable2");
    addend1.VariableChange += addend1_VariableChange;
    addend2.VariableChange += addend2_VariableChange;
}

private void addend1_VariableChange(object sender, VariableChangeEventArgs e)
{
    Project.Current.GetVariable("Model/Variable3").Value = (Int32)e.NewValue + (Int32)addend2.Value;
}

private void addend2_VariableChange(object sender, VariableChangeEventArgs e)
{
    Project.Current.GetVariable("Model/Variable3").Value = (Int32)addend1.Value + (Int32)e.NewValue;
}

public override void Stop()
{
    addend1.VariableChange -= addend1_VariableChange;
    addend2.VariableChange -= addend2_VariableChange;
}
```

This example executes the Sum when an addend changes, without the need to click the Button

Runtime NetLogic: set a variable always in use

```
private IUAVariable addend1;
private IUAVariable addend2;
private RemoteVariableSynchronizer variableSynchronizer;

public override void Start()
{
    addend1 = Project.Current.GetVariable("Model/Variable1");
    addend2 = Project.Current.GetVariable("Model/Variable2");
    addend1.VariableChange += addend1_VariableChange;
    addend2.VariableChange += addend2_VariableChange;

    variableSynchronizer = new RemoteVariableSynchronizer();
    variableSynchronizer.Add(addend1);
    variableSynchronizer.Add(addend2);
}

private void addend1_VariableChange(object sender, VariableChangeEventArgs e)
{
    Project.Current.GetVariable("Model/Variable3").Value = (Int16)e.NewValue + (Int16)addend2.Value;
}

private void addend2_VariableChange(object sender, VariableChangeEventArgs e)
{
    Project.Current.GetVariable("Model/Variable3").Value = (Int16)addend1.Value + (Int16)e.NewValue;
}
```

Starting from the previous example, here Variable1 and Variable2 are associated with Modbus registers, but they are not in use (because not linked to any object on page) so they need to be added to a RemoteVariableSynchronizer

Runtime NetLogic: execute code in asynchronous mode *

```
private PeriodicTask myPeriodicTask;
private IUAVariable variable1;
private double dCounter;

public override void Start()
{
    variable1 = Project.Current.GetVariable("Model/Variable1");
    myPeriodicTask = new PeriodicTask(IncrementVariable, 250, LogicObject);
    myPeriodicTask.Start();
    dCounter = 0;
}

private void IncrementVariable(PeriodicTask task)
{
    dCounter = dCounter + 0.05;
    variable1.Value = 50 + Math.Sin(dCounter) * 50;
}

public override void Stop()
{
    myPeriodicTask.Dispose();
}
```

* The classes available are the following:

PeriodicTask to create a task that executes code at regular time intervals.

DelayedTask to create a task that executes code after a time delay.

LongRunningTask to create tasks that require significant time and/or CPU resources.

The “async” methods in C# cannot be used to access project nodes

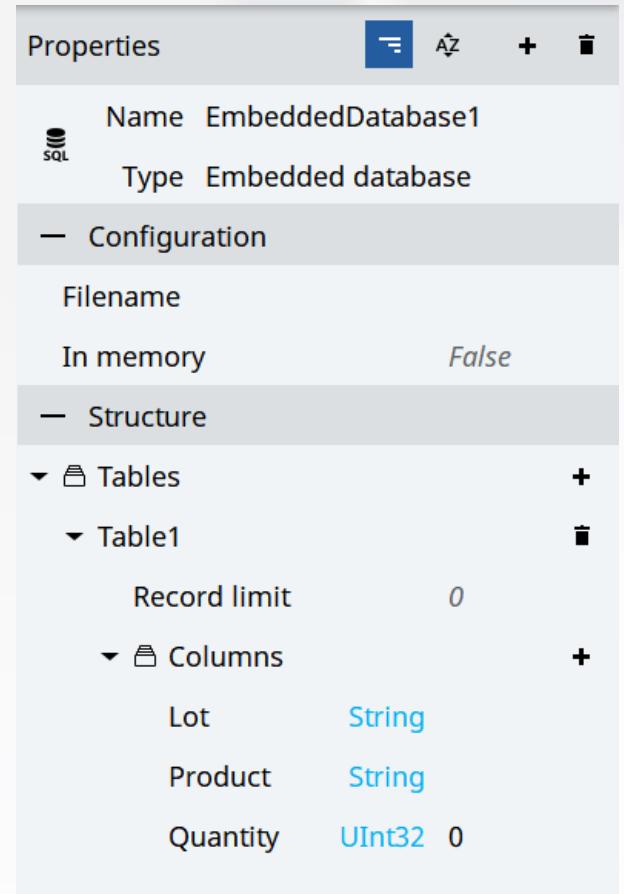
Runtime NetLogic: execute DB query

```
private Store myStore;
private Table myTable;
private object[,] resultSet;
private string[] header;

public override void Start()
{
    myStore = Project.Current.Get<Store>("DataStores/EmbeddedDatabase1");
    myTable = myStore.Tables.Get<Table>("Table1");
}

[ExportMethod]
public void Insert(string Lot, string Prod, uint Quantity)
{
    object[,] rawValues = new object [1,3];      // insert 1 row with 3 columns
    rawValues[0,0] = Lot;
    rawValues[0,1] = Prod;
    rawValues[0,2] = Quantity;

    string[] columns = new string[3] {"Lot", "Product", "Quantity"};
    myTable.Insert(columns, rawValues);
}
```



Runtime NetLogic: execute DB query

```
[ExportMethod]
public void Delete(string Lot)
{
    myStore.Query($"DELETE FROM Table1 WHERE Lot='{Lot}'", out header, out resultSet);
}

[ExportMethod]
public void Update(string Lot, uint Quantity)
{
    myStore.Query($"UPDATE Table1 SET Quantity='{Quantity}' WHERE Lot='{Lot}'", out header, out resultSet);
}

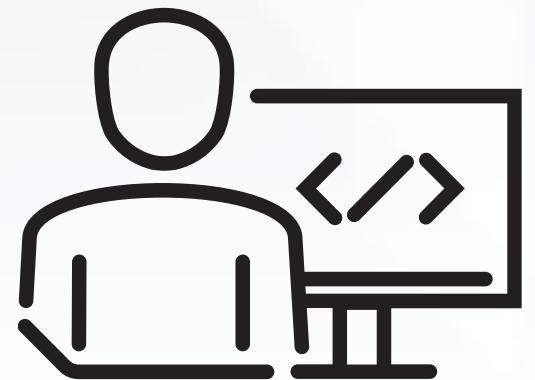
[ExportMethod]
public void Select(string Lot)
{
    myStore.Query($"SELECT Product, QUANTITY FROM Table1 WHERE Lot='{Lot}'", out header, out resultSet);
    for (int i = 0; i < resultSet.GetLength(0); i++)
    {
        Log.Info(LogicObject.BrowseName, $"Product = '{resultSet[i,0]}' - Quantity = '{resultSet[i,1]}'");
    }
}
```

Runtime NetLogic: look at some library scripts

- MQTT push-agent
- FTP Client/Server
- Import/Export Alarm
- Import/Export Translations

Hands-on session

- Write one or more Runtime script
- Try to debug



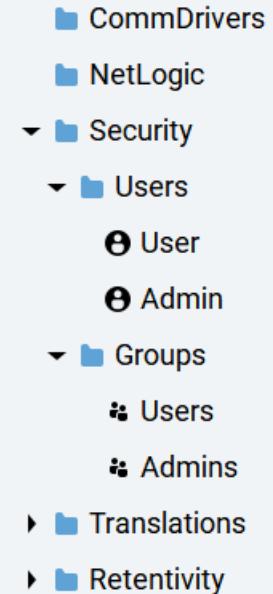


Configure users and groups



Users and groups

- Allow defining a level of Security for commands/UI objects
- Project Properties
 - Authentication Mode: Model = Project's users, Local = iPC Local users, Domain = Active Directory users
 - Default User Folder
 - Password Policy
- Presentation Engine properties
 - Login Window: on Web Presentation Engine only, allow to specify a Window that should contain a Login Form to be prompted at startup.
 - Starting user: "Anonymous" it's the user logged in at the startup (it's not visible in the Users folder)

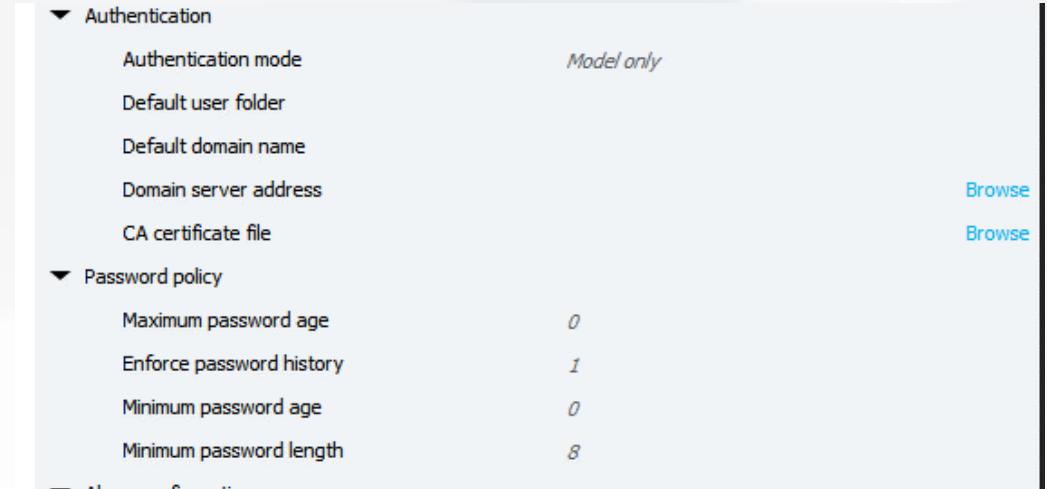


Users and groups

- Allow defining a level of Security for commands/UI objects

- Project Properties

- Authentication Mode: Model = Project's users,
Local = iPC Local users,
Domain = Active Directory users
- Default User Folder
- Password Policy



- Presentation Engine properties

- Login Window: on Web Presentation Engine only, allow to specify a Window that should contain a Login Form to be prompted at startup.
- Starting user: "Anonymous" it's the user logged in at the startup (it's not visible in the Users folder)

Users and groups

- Allow defining a level of Security for commands/UI objects

- Project Properties

- Authentication Mode:
 - Model = Project's users,
 - Local = iPC Local users,
 - Domain = Active Directory users
- Default User Folder
- Password Policy

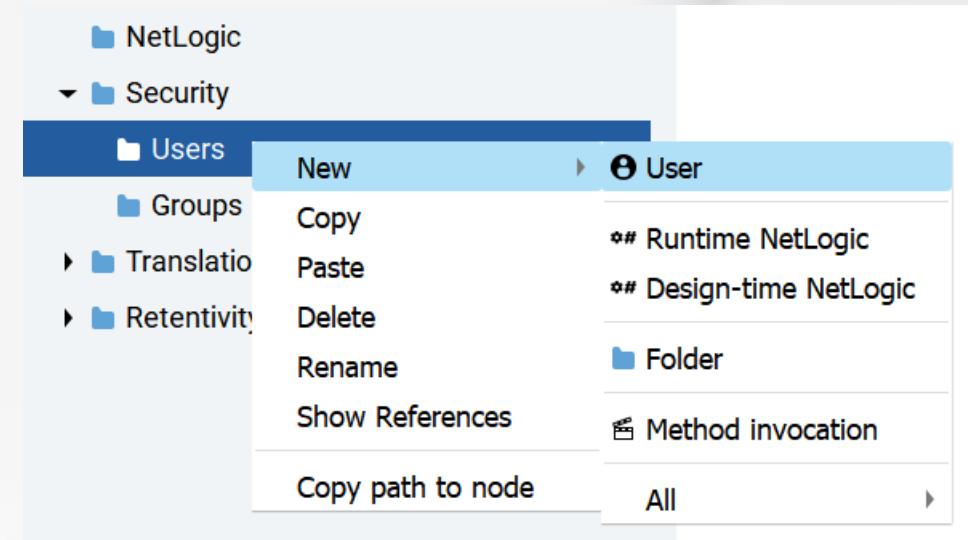
- Presentation Engine properties

- Login Window: on Web Presentation Engine only, allow to specify a Window that should contain a Login Form to be prompted at startup.
- Starting user: "Anonymous" it's the user logged in at the startup (it's not visible in the Users folder)

Properties	
	Name NativePresentationEngine
	Type Native presentation engine
	Dynamic variables polling time 0000:00:00.100
Text render type	Auto
LoginWindow	
Start window	MainWindow (type) ↗
Starting user	Anonymous ↗
Style sheet	DefaultStyleSheet ↗
Session	
Enable virtual keyboard	False

Create users and groups

- User/Group Creation:
 - Users and Groups can be created manually, by the right click on Users or Groups folders (except for Local/Domain users)
- User properties:
 - Password
 - LocaleId: a combination of Regional Settings, Language and Measurement System
 - Languages: Translation table column to be used (overrides the default language of the LocaleId)
 - Measurement system: International System, British imperial or US Customary (overrides the default Measurement system of the LocaleId)
 - Domain: Domain name used in case of authentication mode: Domain only, Domain and Local, or Any (it is automatically populated after a successful login) – blank by default



Create users and groups

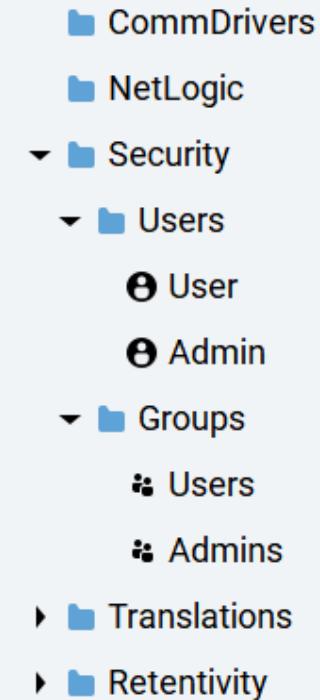
- User/Group Creation:
 - Users and Groups can be created manually, by the right click on Users or Groups folders (except for Local/Domain users)
- User properties:
 - Password
 - LocaleId: a combination of Regional Settings, Language and Measurement System
 - Languages: Translation table column to be used (overrides the default language of the LocaleId)
 - Measurement system: International System, British imperial or US Customary (overrides the default Measurement system of the LocaleId)
 - Domain: Domain name used in case of authentication mode: Domain only, Domain and Local, or Any (it is automatically populated after a successful login) – blank by default

The screenshot shows the FactoryTalk interface with the following details:

- Navigation:** NetLogic > Security > Users
- Context Menu:** A context menu is open over a user entry, with "User" selected.
- Menu Options:** New, Copy, Paste, Delete, Rename, Show References, Copy path to node, All.
- Properties Panel:** Shows the properties for the selected user:
 - Name: User
 - Type: User
 - Password
 - LocaleId
 - Languages
 - Measurement system
 - Domain

Assign users to groups (and viceversa)

- Assignment must be done through the wizard "manage users and groups"
- The wizard allows assigning users to groups but also groups to users



Assign users to groups (and viceversa)

- Assignment must be done through the wizard "manage users and groups"
- The wizard allows assigning users to groups but also groups to users



Assign users to groups (and viceversa)

- Assignment must be done through the wizard "manage users and groups"
- The wizard allows assigning users to groups but also groups to users

Manage Users and Groups

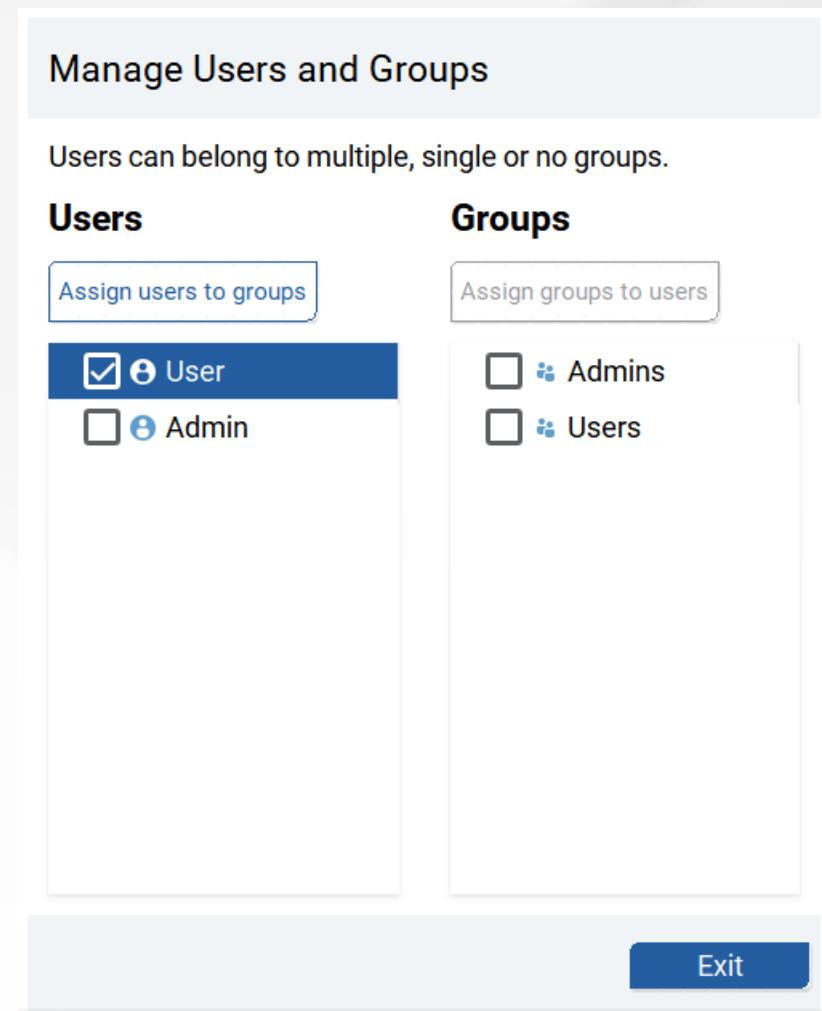
Users can belong to multiple, single or no groups.

Users	Groups
Assign users to groups	Assign groups to users
<input type="checkbox"/>  User	<input type="checkbox"/>  Admins
<input type="checkbox"/>  Admin	<input type="checkbox"/>  Users

Exit

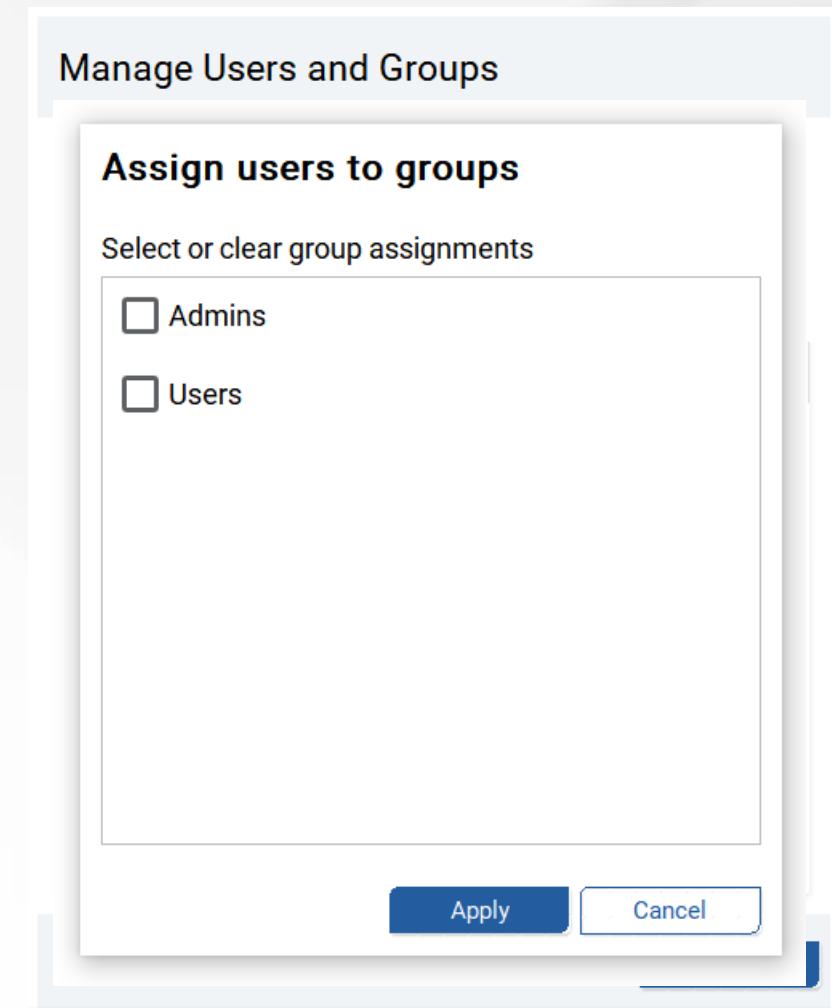
Assign users to groups (and viceversa)

- Assignment must be done through the wizard "manage users and groups"
- The wizard allows assigning users to groups but also groups to users



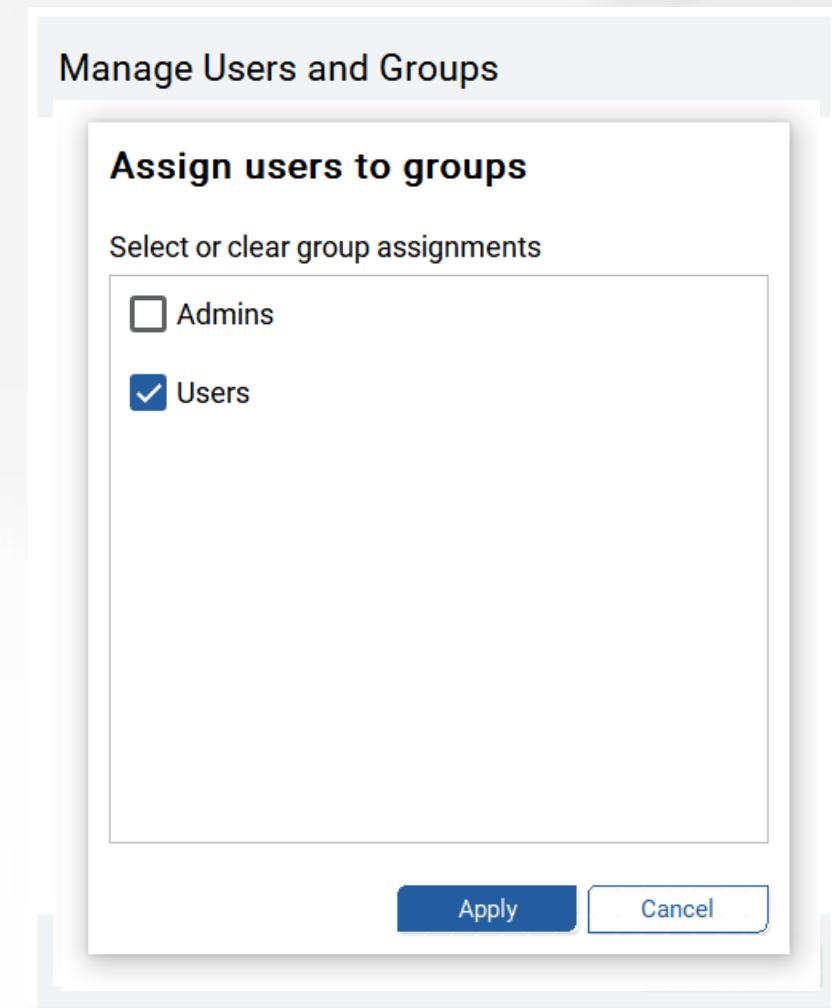
Assign users to groups (and viceversa)

- Assignment must be done through the wizard "manage users and groups"
- The wizard allows assigning users to groups but also groups to users



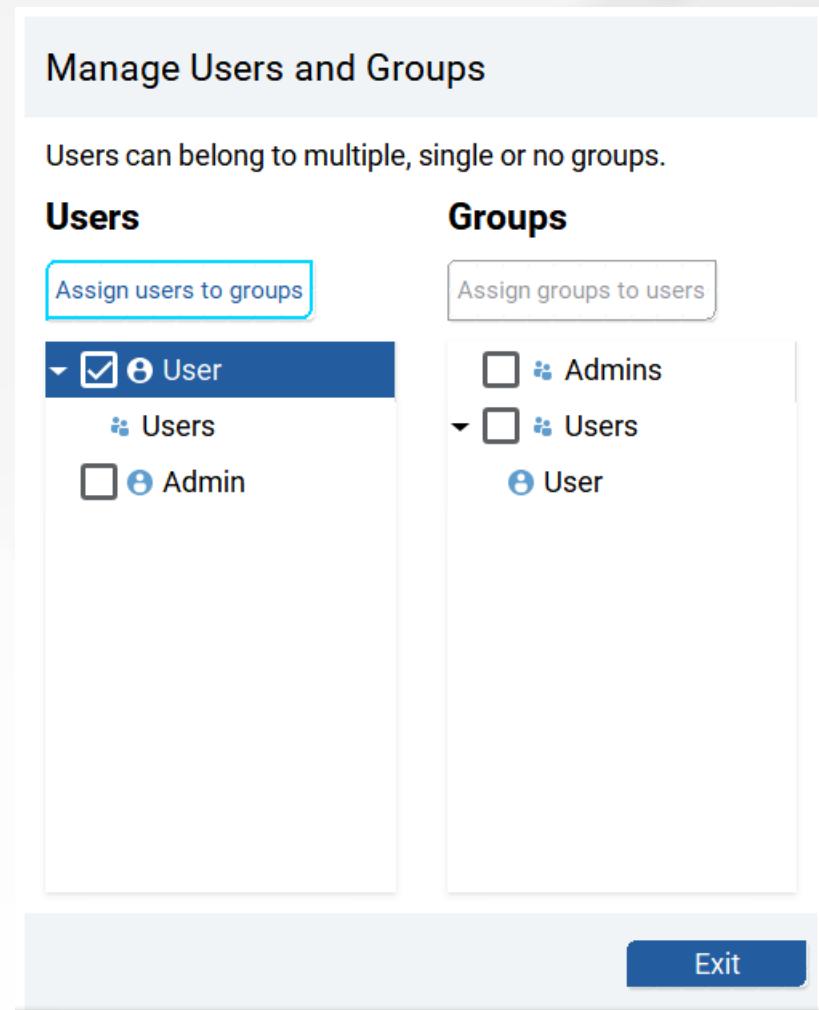
Assign users to groups (and viceversa)

- Assignment must be done through the wizard "manage users and groups"
- The wizard allows assigning users to groups but also groups to users



Assign users to groups (and viceversa)

- Assignment must be done through the wizard "manage users and groups"
- The wizard allows assigning users to groups but also groups to users



Assign users to groups (and viceversa)

- Assignment must be done through the wizard "manage users and groups"
- The wizard allows assigning users to groups but also groups to users

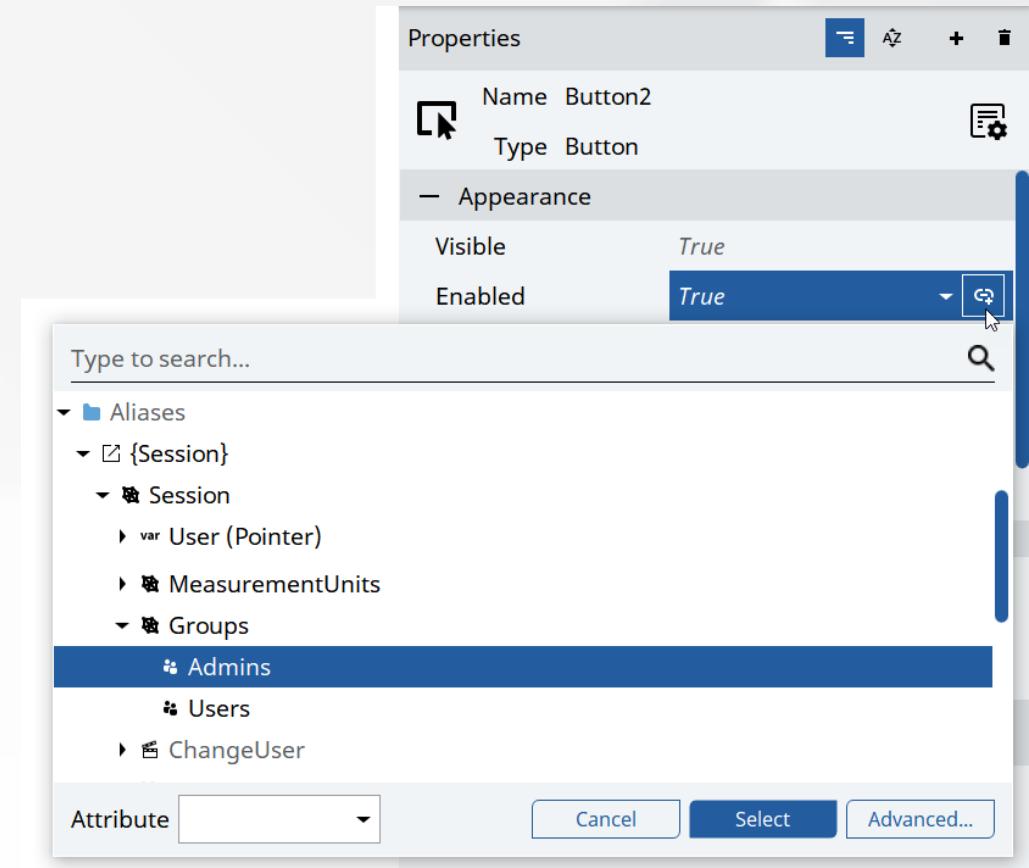
Manage Users and Groups

Users can belong to multiple, single or no groups.

Users	Groups
<input type="button" value="Assign users to groups"/>	<input type="button" value="Assign groups to users"/>
<ul style="list-style-type: none">- <input type="checkbox"/>  User<ul style="list-style-type: none"> Users- <input type="checkbox"/>  Admin<ul style="list-style-type: none"> Admins Users	<ul style="list-style-type: none">- <input type="checkbox"/>  Admins<ul style="list-style-type: none"> Admin- <input type="checkbox"/>  Users<ul style="list-style-type: none"> User Admin

Define security into UI

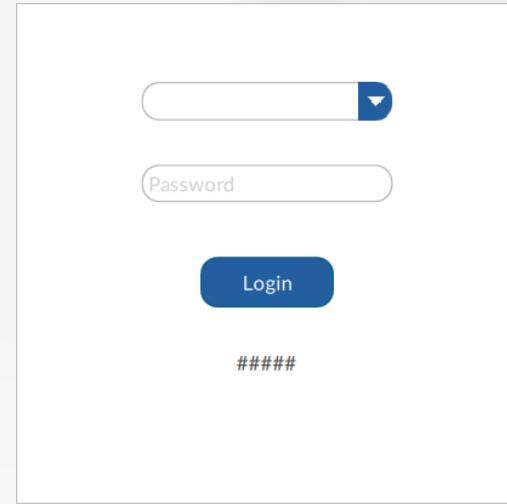
- It's possible to make an object
 - "Visible/Invisible"
 - "Enabled/Disabled"
- just linking a Group to the needed property
{Session} > Session > Groups > ...
- In this example "Button2" will be "Enabled" only for users of group "Admins"



Manage user access at runtime

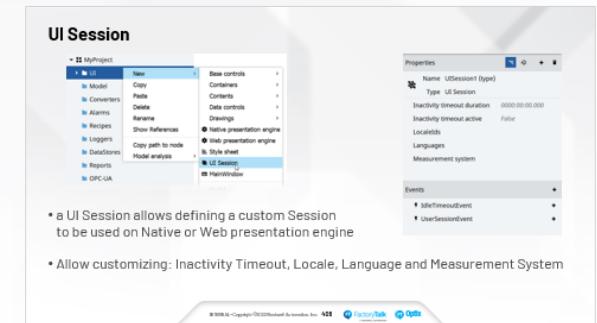
- Login options

- using the **widget "LoginForm"** of Template Library.
Widget comes as a folder that contains the "LoginForm" to be placed in a Panel or called by a Dropdown button
- using the **method "Aliases > {Session} > Session > Login"**
or "Aliases > {Session} > Session > Change User"
- using the **method "Commands > Core Commands > Change User"**



- Logout options

- using the **method "Aliases > {Session} > Session > Logout"**
- using an **UISession object** and use the `IdleTimeoutEvent` to call the method "UI Session > Logout" or "UI Session > Change User"
- using the **script "Idle Timeout"** available in Template Library, place into the MainWindow and associate a "Logout" command



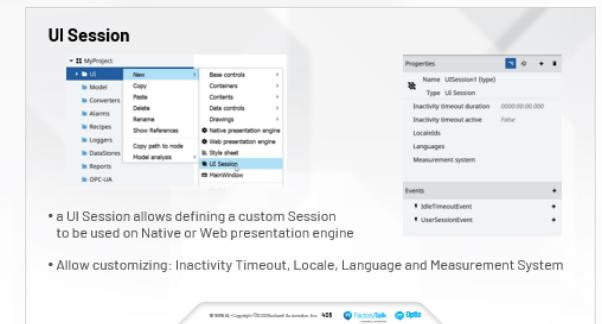
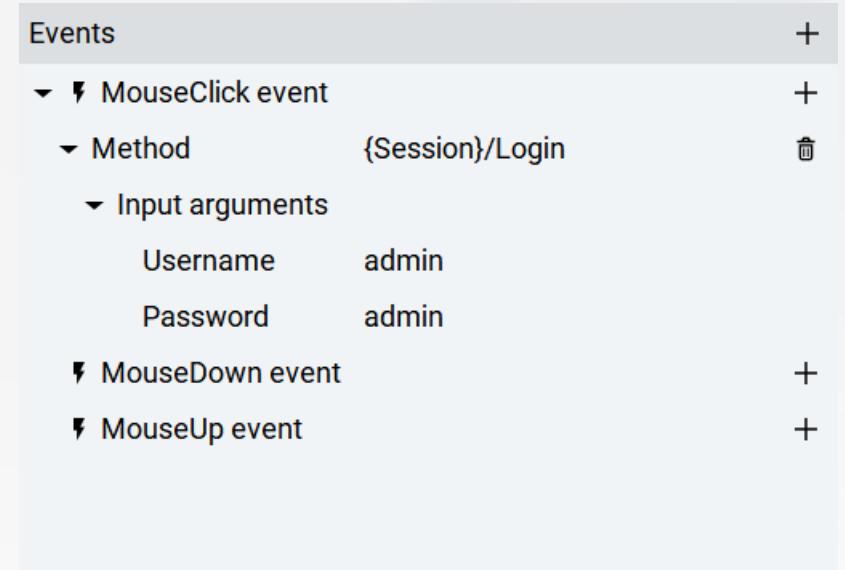
Manage user access at runtime

- Login options

- using the **widget "LoginForm"** of Template Library.
Widget comes as a folder that contains the "LoginForm" to be placed in a Panel or called by a Dropdown button
- using the **method "Aliases > {Session} > Session > Login"** or "Aliases > {Session} > Session > Change User"
- using the **method "Commands > Core Commands > Change User"**

- Logout options

- using the **method "Aliases > {Session} > Session > Logout"**
- using an **UISession object** and use the IdleTimeoutEvent to call the method "UI Session > Logout" or "UI Session > Change User"
- using the **script "Idle Timeout"** available in Template Library, place into the MainWindow and associate a "Logout" command



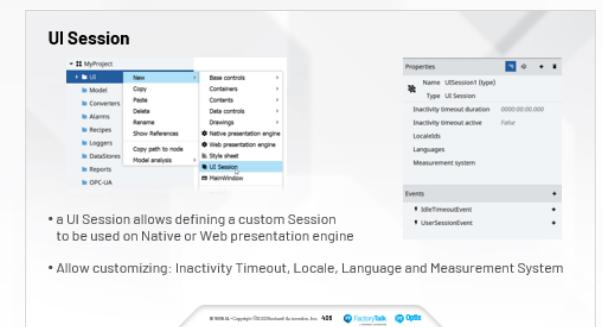
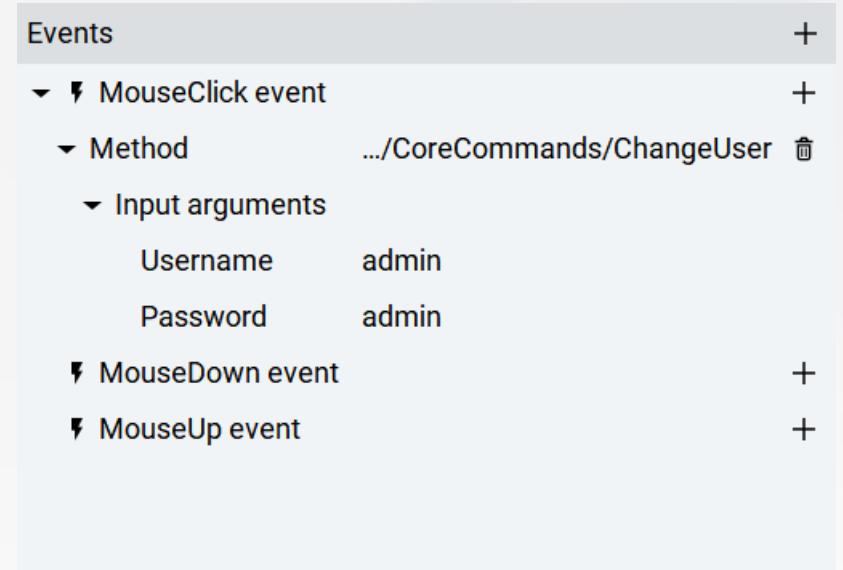
Manage user access at runtime

- Login options

- using the **widget "LoginForm"** of Template Library.
Widget comes as a folder that contains the "LoginForm" to be placed in a Panel or called by a Dropdown button
- using the **method "Aliases > {Session} > Session > Login"** or "Aliases > {Session} > Session > Change User"
- using the **method "Commands > Core Commands > Change User"**

- Logout options

- using the **method "Aliases > {Session} > Session > Logout"**
- using an **UISession object** and use the IdleTimeoutEvent to call the method "UI Session > Logout" or "UI Session > Change User"
- using the **script "Idle Timeout"** available in Template Library, place into the MainWindow and associate a "Logout" command



Edit users at runtime

- using the **widget "User Editor"** of Template Library
- Widget, comes as a folder that contains the "UserEditorOverview" to be placed in a Panel
- It is required to link Users and Groups folder to the widget

The screenshot shows the "User Editor Overview" interface. On the left, a sidebar titled "user" lists "admin". At the bottom are "Create" and "Delete" buttons. The main area has fields for "Name" (set to "user"), "Password" (empty), and "Locale" (empty). To the right is a "Groups" section with a list containing "users" (checked) and "admins" (unchecked). A large blue "Apply" button is at the bottom center.

Name	user
Password	<input type="text"/>
Locale:	<input type="text"/> ▾

Groups

- users
- admins

Apply

Hands-on session

- Create a couple of different users, and groups and make "user - group" association
- Define security at some UI objects (Visibility/Enabling)
- Add the Login form to a Panel
- Optional: call the Login form from a dropdown button
paying attention to the background...
- Try at runtime with different users



Expose data as OPC-UA Server



OPC-UA Server: basic configuration

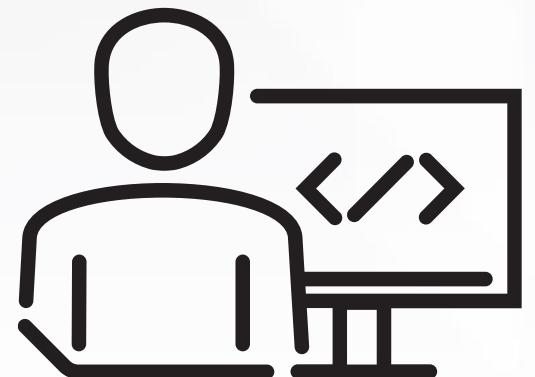
- *Endpoint URL*
 - protocol, IP/hostname, port used by Server
- *Nodes to publish*
 - by default the entire project will be published
- *Sampling Interval*
 - Defines a “best effort” cyclic rate that the Server uses to sample the item from its source.
i.e. could lead the comm.driver tag cyclic rate
- *Multiple connections*
 - Allow multiple OPC-UA Clients to connect
then OPC-UA Server cost 2 tokens instead of 1

The screenshot shows the FactoryTalk Optix interface with the following details:

- Top Navigation:** DataStores, Reports, OPC-UA (selected), CommDrivers, NetLogic.
- Context Menu:** New (highlighted), Copy, Paste, Delete. Sub-menu items: OPC UA server, OPC UA client, Runtime NetLogic.
- Properties Panel:**
 - Server:** Name: OPCUAServer1, Type: OPC UA server.
 - Endpoint URL:** opc.tcp://localhost:59100 (with a [Browse](#) link).
 - Nodes to publish:** (with a plus sign to add more).
 - Sampling interval:** 0000:00:00.100.
 - Multiple connection:** False.
 - Security:** Minimum message security mode: None, Minimum security policy: None.
 - Certificate Files:** Server certificate file (with a [Browse](#) link) and Server private key file (with a [Browse](#) link).

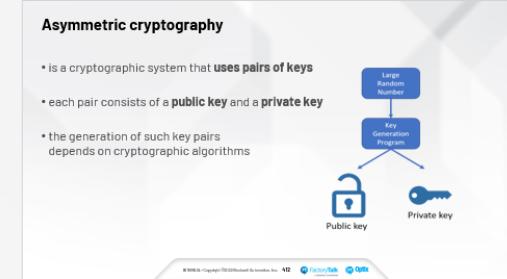
Hands-on session

- Add the OPC UA Server node to the project
- Configure the OPC UA Server to expose the entire project
- Launch the Emulator
- Launch the OPC UA Client from UA Expert and connect to the Emulator
- Browse some variables or call a method exposed like the "Log" method of a Datalogger



OPC-UA Server: security configuration

- Minimum message security mode
 - Security Profile: none, Sign, Sign+Encrypt
- Minimum security policy
- Server Certificate File
 - ".der" file stored in "...\\ProjectFiles\\PKI\\Own\\Server"
- Server Private key file
 - ".pem" file stored in "...\\ProjectFiles\\PKI\\Own\\Server"
- If we do not provide any Certificate file,
The runtime will automatically generate
a self-signed certificate



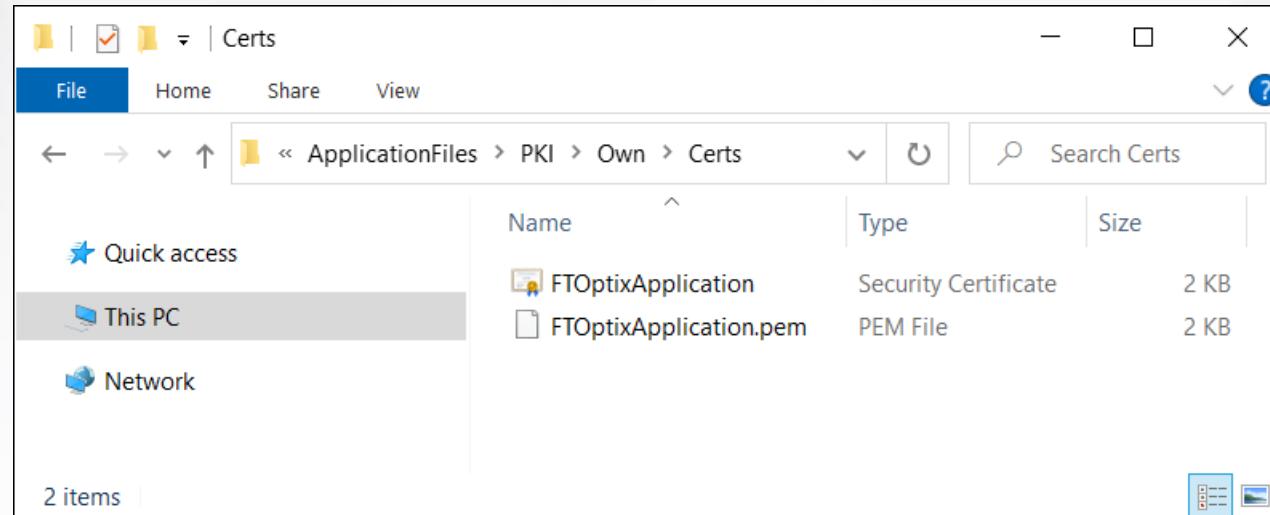
Properties

Name	OPCUAServer1
Type	OPC UA server
— Server	
Endpoint URL	opc.tcp://localhost:59100 Browse
Nodes to publish	+
Sampling interval	0000:00:00.100
Multiple connection	<i>False</i>
— Security	
Minimum message security mode	<i>None</i>
Minimum security policy	<i>None</i>
Server certificate file	Browse
Server private key file	Browse

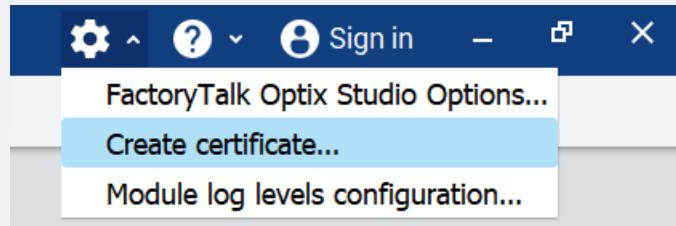
OPC-UA Server: security certificates

- If not provided, the Runtime will automatically generate the ".der" Certificate and the ".pem" Public Key file into folder: "...\\ApplicationFiles\\PKI\\Own\\Certs\\" of the Application.
- For a project executed within Emulator, you will find files into folder:

```
%localappdata%\Rockwell Automation\FactoryTalk Optix  
\Emulator\Projects\<Project>\ApplicationFiles\PKI\Own\Certs\
```



OPC-UA Server: security certificates



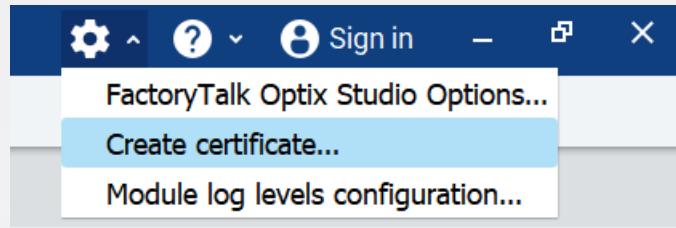
- **Optionally**

OPC-UA Server Certificate can be

- created manually,
- purchased from a certified authority
- In these cases files ".der" and ".pem" have to be imported into the OPC-UA Server configuration using Browse button

The dialog box is titled 'Create certificate'. It has two main sections: 'Subject' and 'OPC UA information'. The 'Subject' section contains fields for 'Common name*', 'Organization*', 'Organization unit', 'Locality', 'State', and 'Country'. The 'OPC UA information' section contains fields for 'Application URI*', 'Domain names', and 'IP addresses'. On the right side, there is a 'Certificate settings' group with dropdown menus for 'RSA key strength' (set to 2048 bits), 'Signature algorithm' (set to SHA-256), and 'Expiration date' (set to 09/21/2025 00:00:00). Below these settings are fields for 'Name' (containing 'optixhmi_cert') and 'Location' (containing 'C:/Users/Rockwell/Documents'). At the bottom left is a note '*Required field'. At the bottom right are 'Cancel' and 'Create' buttons.

OPC-UA Server: security certificates



- **Optionally**

OPC-UA Server Certificate can be

- created manually,
- purchased from a certified authority
- In these cases files ".der" and ".pem" have to be imported into the OPC-UA Server configuration using Browse button

The dialog box is titled 'Create certificate'. It has two main sections: 'Subject' and 'OPC UA information'.

Subject:

- Common name*: FTOptixApplication@Win10VM
- Organization*: Asem a Rockwell Automation company
- Organization unit: (empty)
- Locality: (empty)
- State: (empty)
- Country: (empty)

OPC UA information:

- Application URI*: urn:Win10VM:FactoryTalkOptixHMI:FTOptixApplication
- Domain names: DESKTOP-ET9DM06
- IP addresses: (empty)

Certificate settings:

- RSA key strength: 2048 bits
- Signature algorithm: SHA-256
- Expiration date: 09/21/2025 00:00:00

Other fields:

- Name: optixhmi_cert
- Location: C:/Users/Rockwell/Documents

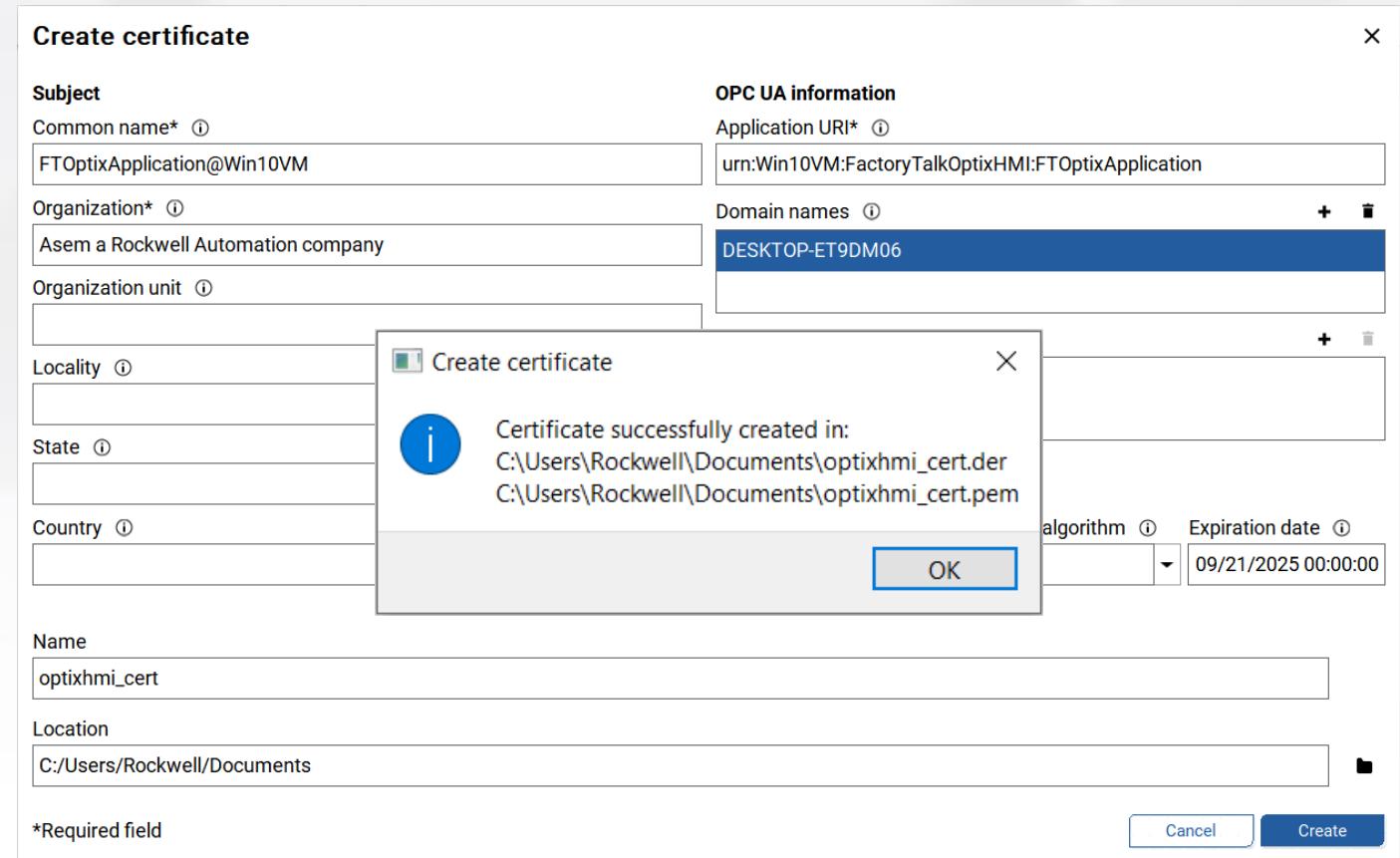
*Required field

Buttons at the bottom: Cancel and Create.

OPC-UA Server: security certificates

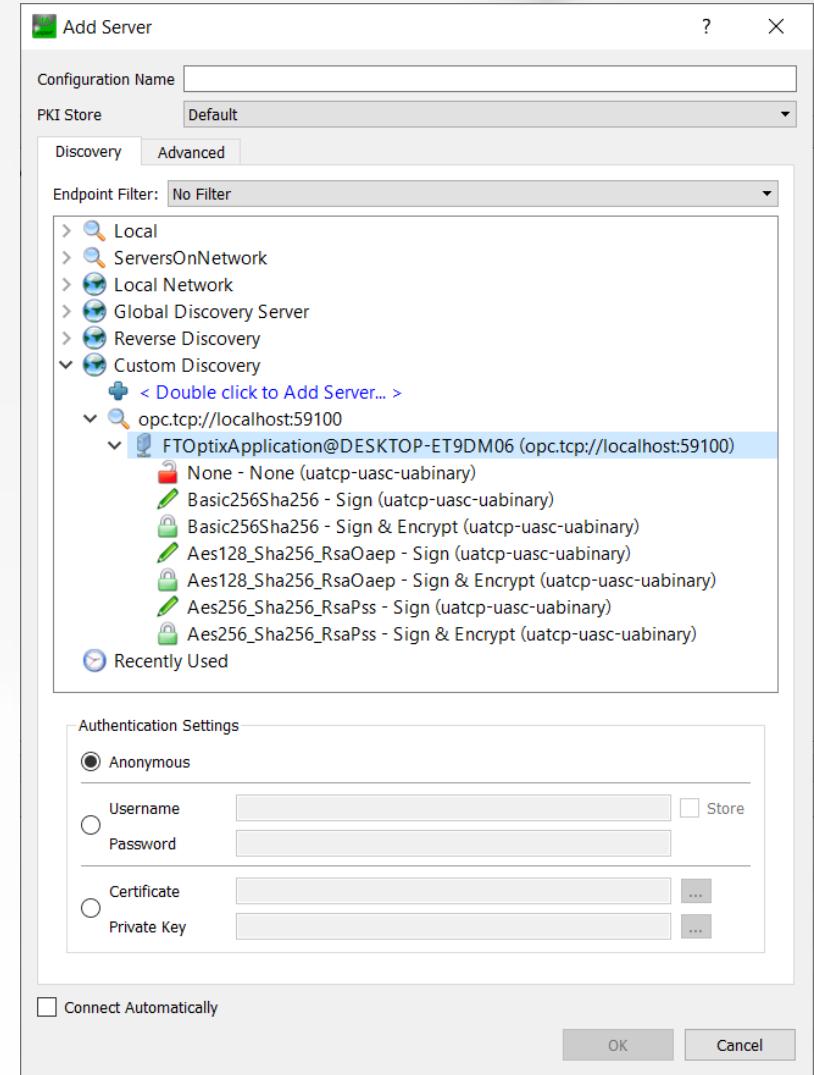


- **Optionally**
OPC-UA Server Certificate can be
 - created manually,
 - purchased from a certified authority
- In these cases files ".der" and ".pem" have to be imported into the OPC-UA Server configuration using Browse button



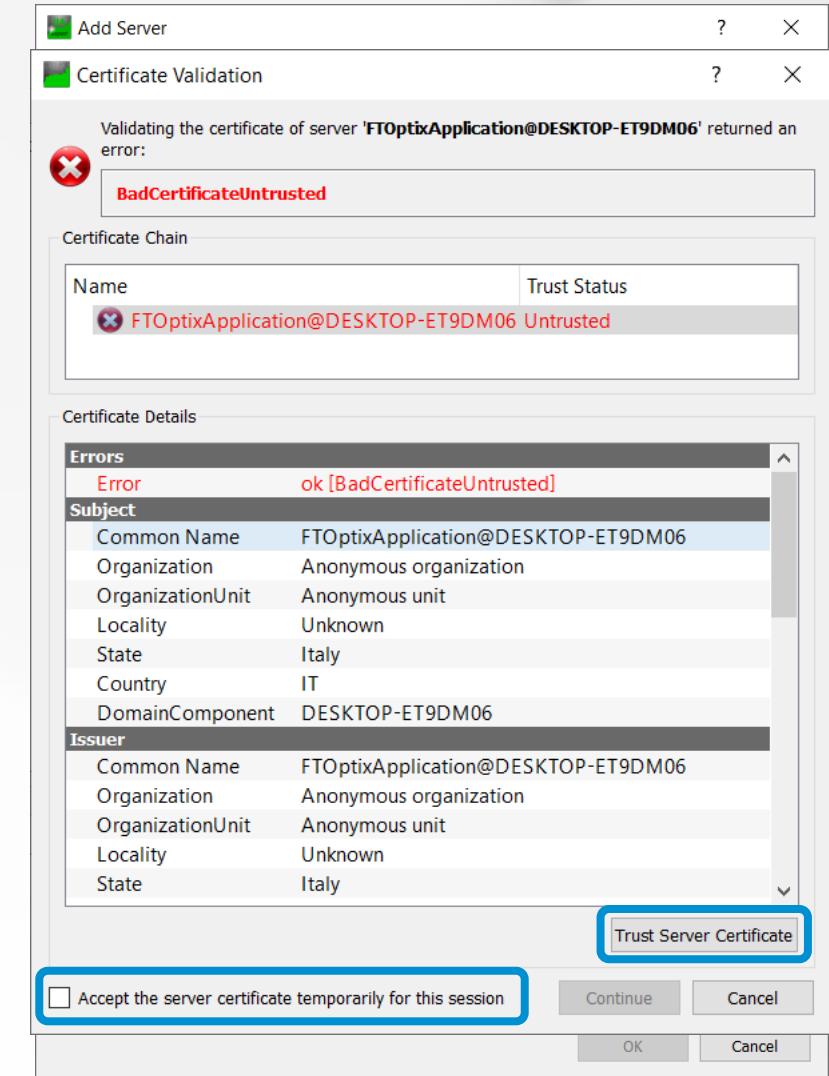
OPC-UA Server: connection using certificates

- Using a 3rd part OPC-UA Client, you will need to define the endpoint URL and then select one of the available combinations of Security Policies and Security Profiles.
- This 3rd part OPC-UA Client receives the Server Certificate, but given that is "untrusted" ask us if we want to:
 - A. Trust Server certificate
or
 - B. Accept the server certificate temporarily



OPC-UA Server: connection using certificates

- Using a 3rd part OPC-UA Client, you will need to define the endpoint URL and then select one of the available combinations of Security Policies and Security Profiles.
- This 3rd part OPC-UA Client receives the Server Certificate, but given that is "untrusted" ask us if we want to:
 - A. Trust Server certificate
or
 - B. Accept the server certificate temporarily



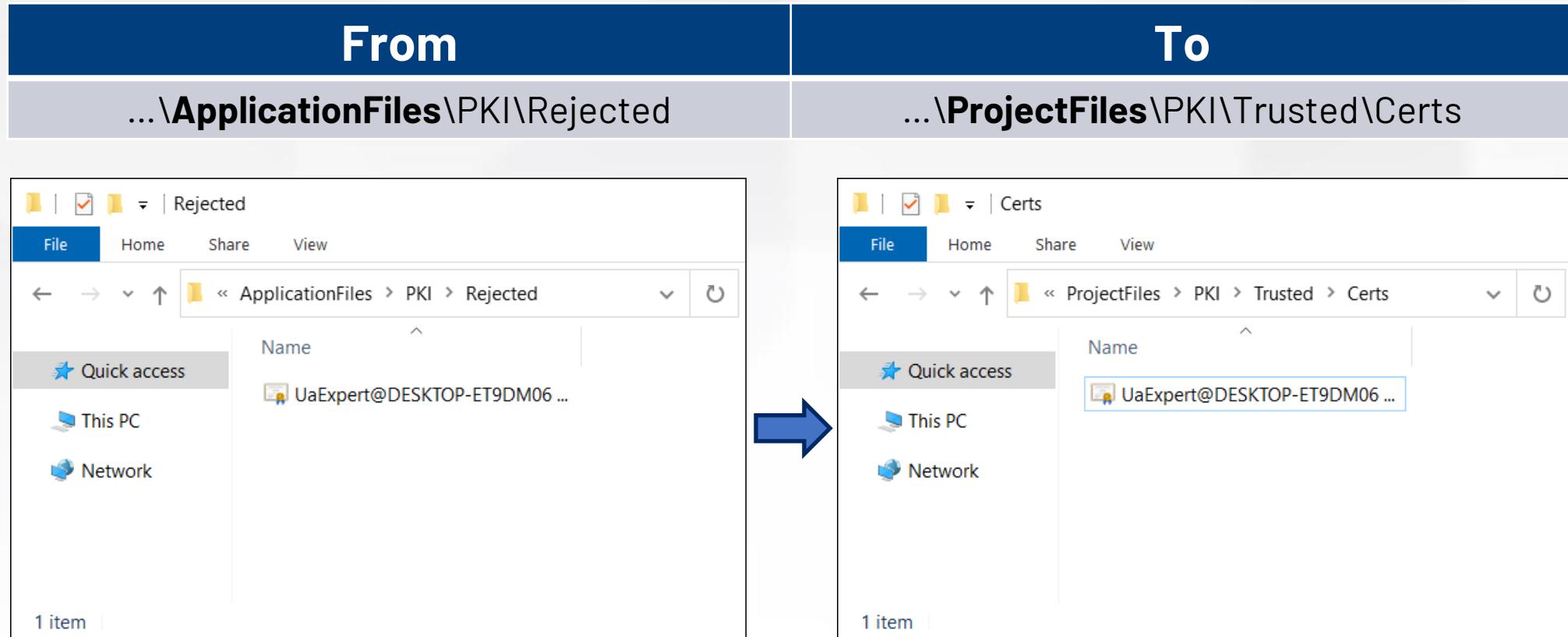
OPC-UA Server: connection using certificates

- If we do not provide the OPC-UA Client certificate to the Server (containing the public key), the result is "***BadSecurityCheckFailed***"

Timestamp	Source	Server	Message
9/22/2022 6:40:42.698 PM	Server Node	FTOptixApplication@DESKTOP-ET9DM06	Endpoint: 'opc.tcp://localhost:59100'
9/22/2022 6:40:42.698 PM	Server Node	FTOptixApplication@DESKTOP-ET9DM06	Security policy: 'http://opcfoundation.org/UA/SecurityPolicy#Basic256Sha256'
9/22/2022 6:40:42.698 PM	Server Node	FTOptixApplication@DESKTOP-ET9DM06	ApplicationUri: 'urn:DESKTOP-ET9DM06:FactoryTalkOptixHMI:FTOptixApplication'
9/22/2022 6:40:42.698 PM	Server Node	FTOptixApplication@DESKTOP-ET9DM06	Used UserTokenType: Anonymous
9/22/2022 6:40:44.918 PM	Server Node	FTOptixApplication@DESKTOP-ET9DM06	Error 'BadSecurityChecksFailed' was returned during OpenSecureChannel
9/22/2022 6:40:44.918 PM	Server Node	FTOptixApplication@DESKTOP-ET9DM06	Connection status of server 'FTOptixApplication@DESKTOP-ET9DM06' changed to 'Disconnected'.

OPC-UA Server: connection using certificates

- FTOptix Application with OPC-UA Server, by default, receive and reject the OPC-UA Client certificate, so we need to move it



OPC-UA Server: connection using certificates

- The folder: "...\\ProjectFiles\\PKI\\Trusted\\Certs"

- can be the folder inside of Emulator:

```
%localappdata%\Rockwell Automation\FactoryTalk Optix  
\Emulator\Projects\<Project>\ProjectFiles\PKI\Trusted\Certs
```

- but **better if it's the one where the project is stored:**

```
C:\<Project_Folder>\<Project_Name>\ProjectFiles\PKI\Trusted\Certs
```

in this way, when the project is transferred to HMI or iPC, the certificate will be transferred within the application

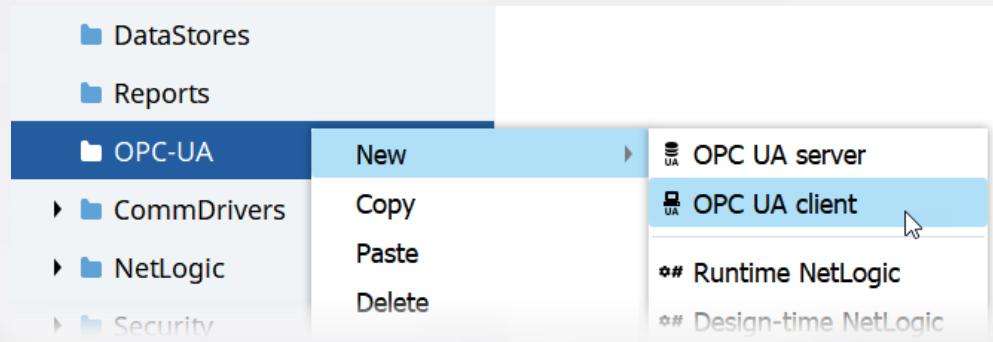
- After moving the OPC-UA Client certificate from Rejected to Trusted, the Client will be able to connect to the Server.



Exchange data via OPC-UA Client



OPC-UA Client: basic configuration



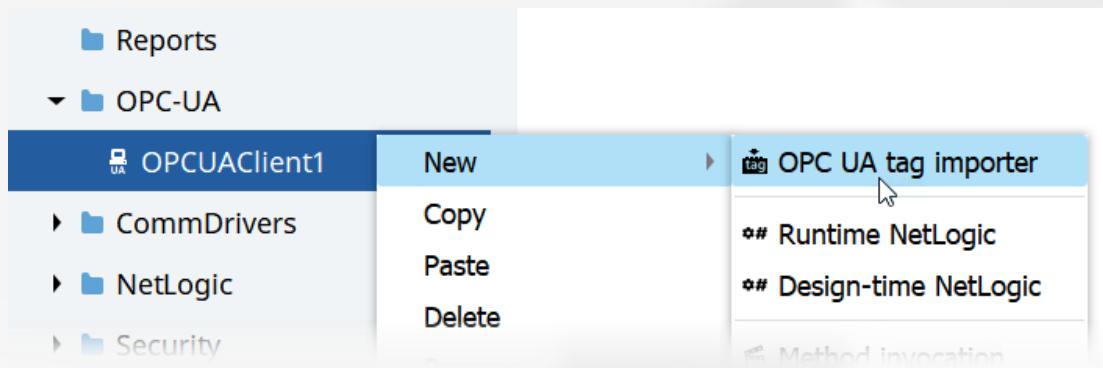
- *Events*: If not interested to read OPC-UA Server Events, set to None
- *Synchronize Node IDs on start*: In case node id of OPC-UA Server changes
- *Server Endpoint URL*: URL of OPC-UA Server to connect to
- *Requested publishing interval*: defines the rate that the OPC-UA Server returns data change notifications to the OPC UA client

A screenshot of a 'Properties' dialog box. At the top, it says 'Properties' and has icons for filter, sort, add, edit, and delete. The main area contains the following settings:

- Client**:
 - Events: All
 - Synchronize the Node IDs on start: False
- Server Connection**:
 - Server endpoint URL: opc.tcp://localhost:59100
 - Verify server identity: True
 - Requested publishing interval: 0000:00:00.000
- Runtime configurations**: A '+' sign icon.
- Security**: A greyed-out section.

OPC-UA Client: tag importer

- Like any other Tag Importer
- Allow to import/synchronize tags exposed by an OPC-UA Server



OPC-UA > OPCUAClient1 > OPCUATagImporter					Type to search...
	PLC Item	HMI Item	DataType	Status	
<input type="checkbox"/>	▶ Demo			New	
<input type="checkbox"/>	▶ DeviceSet			New	
<input type="checkbox"/>	▶ DeviceTopology			New	
<input type="checkbox"/>	NetworkSet			New	

Buttons at the bottom: Delete, Ignore, Synchronize.

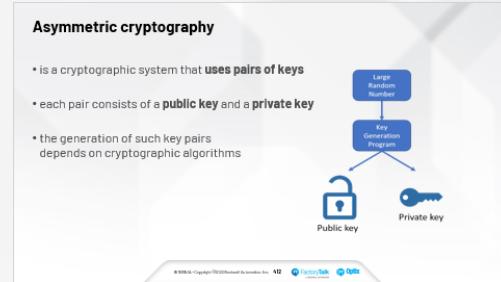
Hands-on session

- Launch the OPC UA Server (UaAnsiCServer)
- Add the OPC UA Client node to the project and configure the server endpoint
- Open the OPC UA tag importer and import the folder "Demo > BoilerDemo"
- Into the UI:
 - attach the variable "FillLevel" to a Text box
 - attach the variable "Temperature" to a Text box
 - add a button that calls the method "Fill"
setting the argument "SetPoint" as variable
 - add a button that calls the method "Heat"
setting the argument "SetPoint" as variable



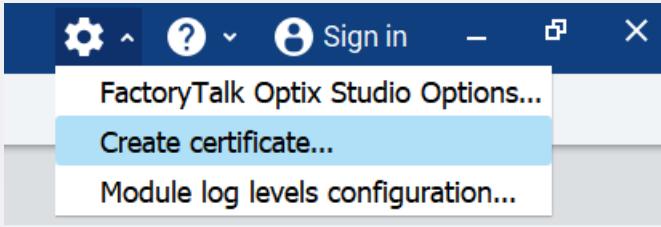
OPC-UA Client: security configuration

- Verify server identity
 - if True, accept only CA-issued certificates
 - if False, accept Self-signed certificates**
- Minimum message security mode
 - Security Profile: none, Sign, Sign+Encrypt
- Minimum security policy
- Client Certificate File
 - "der" file stored in "...\\ProjectFiles\\PKI\\Own\\Client"
- Client Private key file
 - ".pem" file stored in "...\\ProjectFiles\\PKI\\Own\\Client"
- Runtime, **will NOT automatically generate** a self-signed certificate for the Client so it's necessary to create one manually



— Client	
Events	All
Synchronize the Node IDs on start	<i>False</i>
— Server Connection	
Server endpoint URL	<i>opc.tcp://localhost:59100</i>
Verify server identity	<i>True</i>
Requested publishing interval	<i>0000:00:00.000</i>
Runtime configurations	+
— Security	
Minimum message security mode	<i>None</i>
Minimum security policy	<i>None</i>
Client certificate file	Browse
Client private key file	Browse

OPC-UA Client: create security certificates



- **is necessary to**
 - create a Self-Signed certificate
 - or purchase one from a certified authority
- Domain names: hostname of PC/HMI where the OPC-UA Client will run
- IP address: IP Address of PC/HMI where the OPC-UA Client will run

Create certificate

Subject

Common name* ⓘ

Organization* ⓘ

Organization unit ⓘ

Locality ⓘ

State ⓘ

Country ⓘ

OPC UA information

Application URI* ⓘ

Domain names ⓘ + ⚙️

IP addresses ⓘ + ⚙️

Certificate settings

RSA key strength ⓘ 2048 bits

Signature algorithm ⓘ SHA-256

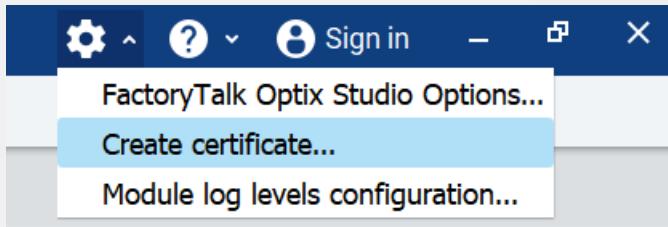
Expiration date ⓘ 09/21/2025 00:00:00

Name optixhmi_cert

Location C:/Users/Rockwell/Documents

*Required field

OPC-UA Client: create security certificates



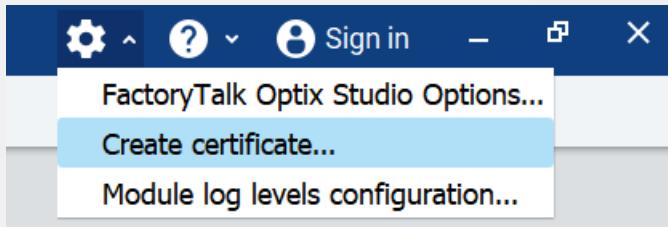
- **is necessary to**
 - create a Self-Signed certificate
 - or purchase one from a certified authority
- Domain names: hostname of PC/HMI where the OPC-UA Client will run
- IP address: IP Address of PC/HMI where the OPC-UA Client will run

The screenshot shows the 'Create certificate' dialog box. It is divided into several sections:

- Subject** (left column):
 - Common name*: FTOptixApplication@Win10VM
 - Organization*: Asem a Rockwell Automation company
 - Organization unit
 - Locality
 - State
 - Country
- OPC UA information** (right column):
 - Application URI*: urn:Win10VM:FactoryTalkOptixHMI:FTOptixApplication
 - Domain names:
 - DESKTOP-ET9DM06
 - IP addresses
- Certificate settings**:
 - RSA key strength: 2048 bits
 - Signature algorithm: SHA-256
 - Expiration date: 09/21/2025 00:00:00
- File settings**:
 - Name: optixhmi_cert
 - Location: C:/Users/Rockwell/Documents
- Buttons at the bottom: 'Cancel' and 'Create'.

A note at the bottom left of the dialog box says: *Required field.

OPC-UA Client: create security certificates



- is necessary to
 - create a Self-Signed certificate
 - or purchase one from a certified authority
- Domain names: hostname of PC/HMI where the OPC-UA Client will run
- IP address: IP Address of PC/HMI where the OPC-UA Client will run

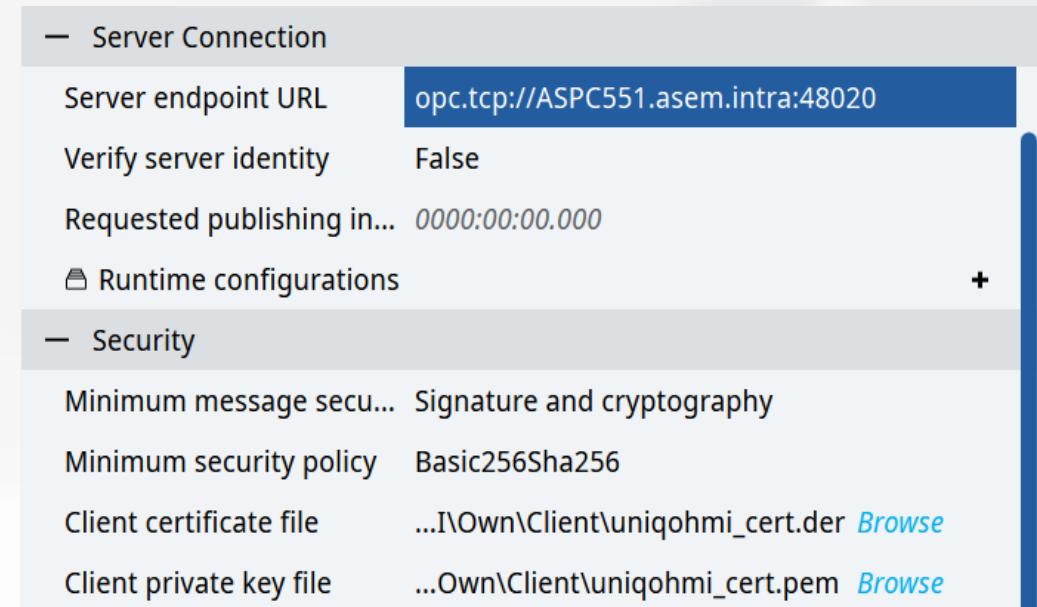
A screenshot of the 'Create certificate' dialog box. The 'Subject' section contains 'Common name*' (FTOptixApplication@Win10VM), 'Organization*' (Asem a Rockwell Automation company), and 'Domain names' (DESKTOP-ET9DM06). The 'OPC UA information' section shows 'Application URI*' (urn:Win10VM:FactoryTalkOptixHMI:FTOptixApplication). Below the main form, a smaller 'Create certificate' dialog box displays a success message: 'Certificate successfully created in: C:\Users\Rockwell\Documents\optixhmi_cert.der C:\Users\Rockwell\Documents\optixhmi_cert.pem'. The 'Name' field in the main dialog is set to 'optixhmi_cert' and the 'Location' field is 'C:/Users/Rockwell/Documents'. A note at the bottom says '*Required field'. At the bottom right of the main dialog are 'OK', 'Cancel', and 'Create' buttons.

OPC-UA Client: connection using certificates

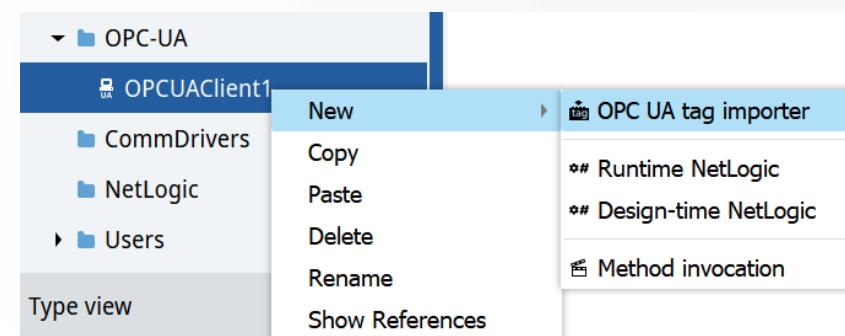
- Connection to a 3rd part OPC-UA Server

1. Configure OPC-UA Client Settings:

- Server endpoint URL
- Set "Verify server identity" to False
if OPC-UA Server uses a self-signed certificate
- Set Security Profile and Security Policy
- Import self-signed Client Certificate file
and Client Private Key file



2. Import tags adding the object OPC UA tag importer



OPC-UA Client: connection using certificates

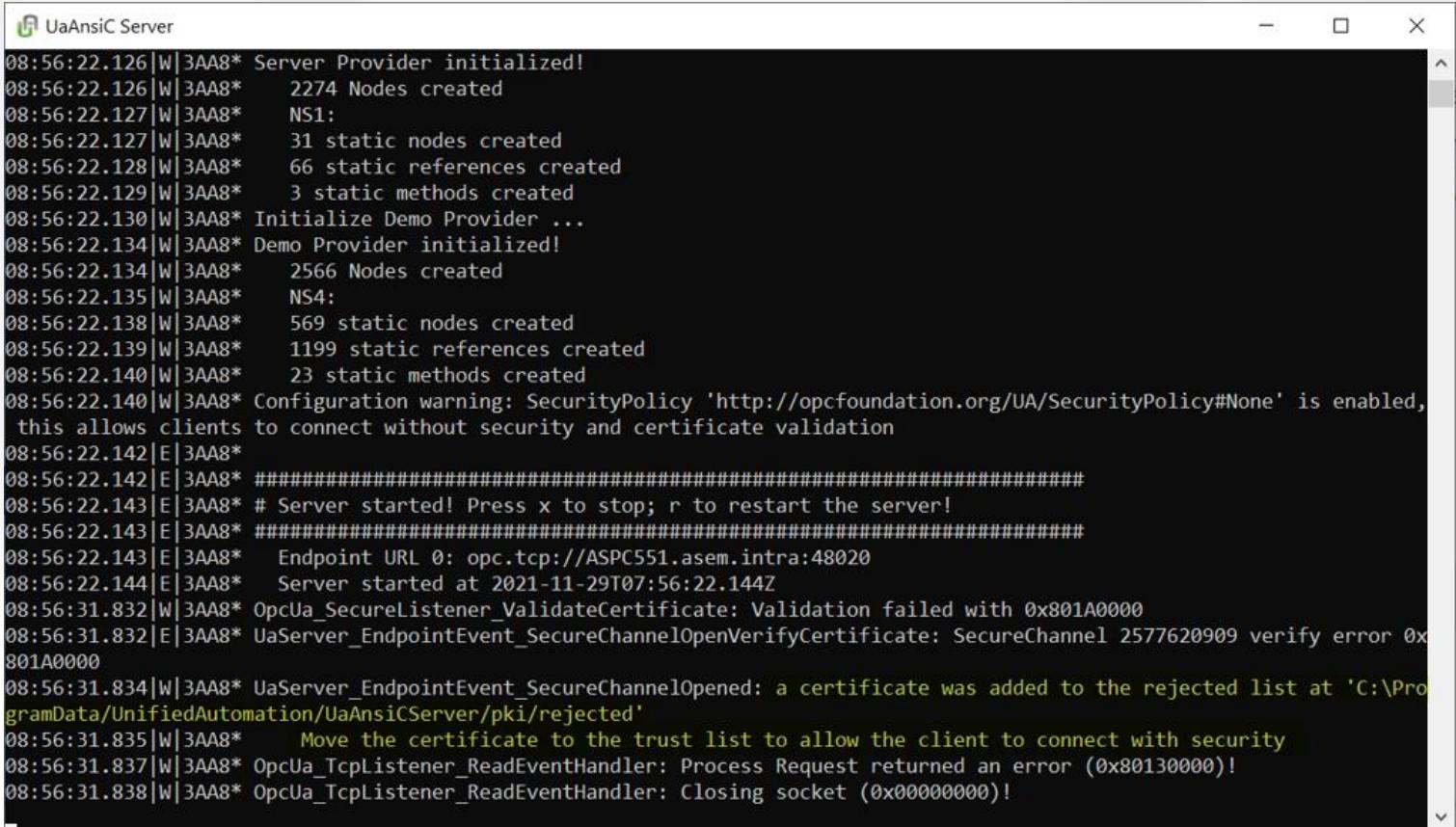
- OPC-UA Client sends his certificate to the Server,
but in this case, the Server automatically set the certificate into the Rejected folder so
"BadSecurityChecksFailed" is triggered

Code	Timestamp	Category	Object	Message
43	2021-11-29 08:54:31.967	TaskConsumer		Exception caught: Connection failed with error BadSecurityChecksFailed

Q Studio Output

OPC-UA Client: connection using certificates

- In this case, we need to go to the OPC-UA Server and manually move the Client certificate from Rejected to Trusted folder
- After this operation, the Client will be able to connect to Server to Browse tag for import and exchange data.



```
UaAnsiC Server
08:56:22.126|W|3AA8* Server Provider initialized!
08:56:22.126|W|3AA8* 2274 Nodes created
08:56:22.127|W|3AA8* NS1:
08:56:22.127|W|3AA8* 31 static nodes created
08:56:22.128|W|3AA8* 66 static references created
08:56:22.129|W|3AA8* 3 static methods created
08:56:22.130|W|3AA8* Initialize Demo Provider ...
08:56:22.134|W|3AA8* Demo Provider initialized!
08:56:22.134|W|3AA8* 2566 Nodes created
08:56:22.135|W|3AA8* NS4:
08:56:22.138|W|3AA8* 569 static nodes created
08:56:22.139|W|3AA8* 1199 static references created
08:56:22.140|W|3AA8* 23 static methods created
08:56:22.140|W|3AA8* Configuration warning: SecurityPolicy 'http://opcfoundation.org/UA/SecurityPolicy#None' is enabled,
this allows clients to connect without security and certificate validation
08:56:22.142|E|3AA8*
08:56:22.142|E|3AA8* #####
08:56:22.143|E|3AA8* # Server started! Press x to stop; r to restart the server!
08:56:22.143|E|3AA8* #####
08:56:22.143|E|3AA8* Endpoint URL 0: opc.tcp://ASPC551.asem.intra:48020
08:56:22.144|E|3AA8* Server started at 2021-11-29T07:56:22.144Z
08:56:31.832|W|3AA8* OpcUa_SecureListener_ValidateCertificate: Validation failed with 0x801A0000
08:56:31.832|E|3AA8* UaServer_EndpointEvent_SecureChannelOpenVerifyCertificate: SecureChannel 2577620909 verify error 0x
801A0000
08:56:31.834|W|3AA8* UaServer_EndpointEvent_SecureChannelOpened: a certificate was added to the rejected list at 'C:\Pro
gramData\UnifiedAutomation\UaAnsiCServer\pki/rejected'
08:56:31.835|W|3AA8* Move the certificate to the trust list to allow the client to connect with security
08:56:31.837|W|3AA8* OpcUa_TcpListener_ReadEventHandler: Process Request returned an error (0x80130000)!
08:56:31.838|W|3AA8* OpcUa_TcpListener_ReadEventHandler: Closing socket (0x00000000)!
```

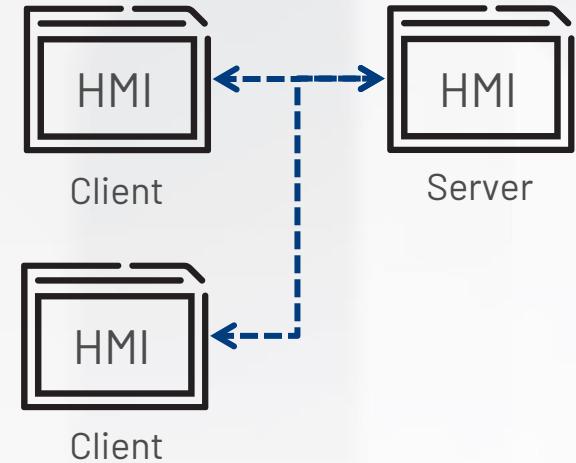
A large, abstract graphic of blue dots forms a wave-like pattern across the slide, starting from the bottom left and curving upwards towards the top right.

Define client-server architectures via OPC-UA



Define client-server architectures via OPC-UA

- Thanks to the Full-featured OPC-UA architecture of FT Optix
It's possible to define client-server architectures
- Clients will have access to:
 - Variables
 - Methods
 - Alarms and Historical Alarms
 - Datalogger and Eventlogger
 - Recipes
- Client: HMI project with OPC-UA Client (Full OPC-UA Client = 3 tokens)
- Server: HMI project with OPC-UA Server (1 client = 1 token, 2 or more clients = 3 tokens)



Configure the server

- On the HMI project, that is going to act as a Server, add the OPCUAServer node
- Endpoint can be manually changed
- Also Nodes to publish can be manually defined

The image shows two side-by-side windows. The left window is titled 'Project view' and displays a tree structure of a project named 'BoilerDemo'. The 'OPC-UA' folder is expanded, showing the 'OPCUAServer' node, which is highlighted with a blue selection bar. Other nodes visible include 'UI', 'Model', 'Converters', 'Alarms', 'Recipes', 'Loggers', 'DataStores', 'Reports', 'CommDrivers', and 'NetLogic'. A search bar at the top of the project view window contains the text 'Type to search...'. The right window is titled 'Properties' and shows configuration settings for the selected 'OPCUAServer' node. The 'Name' is set to 'OPCUAServer' and the 'Type' is 'OPC UA server'. Under the 'Server' section, the 'Endpoint URL' is set to 'opc.tcp://localhost:59100' with a 'Browse' link. The 'Nodes to publish' section lists '0000:00:00.100'. The 'Sampling interval' is set to '0000:00:00.100'. Under 'Security', 'Multiple connection' is set to 'False'. Under 'Information', the 'Product URI' is listed as 'nTalkOptivHMI-FTOptivApplication'.

Properties	
Name	OPCUAServer
Type	OPC UA server
— Server	
Endpoint URL	opc.tcp://localhost:59100 Browse
Nodes to publish	0000:00:00.100
Sampling interval	0000:00:00.100
Multiple connection	False
— Security	
Minimum message sec...	None
Minimum security policy	None
Server certificate file	Browse
Server private key file	Browse
— Information	
Product URI	nTalkOptivHMI-FTOptivApplication

Configure the client

- On the HMI project, that is going to act as a Client, add the OPCUAClient node
- Server endpoint must be the one defined into the Server project
- Events property should be set to "All" to be able to attach to Methods, Alarms, Loggers...

The screenshot shows two windows side-by-side. The left window is the 'Project view' showing a tree structure of a project named 'BoilerDemoClient'. Under 'OPC-UA', the node 'OPCUAClient1' is selected and highlighted in blue. The right window is the 'Properties' window for this node. It displays the following configuration:

Property	Value
Name	OPCUAClient1
Type	OPC UA client
Client	
Events	All
Synchronize the Node IDs on start	False
Server Connection	
Server endpoint URL	opc.tcp://localhost:59100
Verify server identity	True
Requested publishing interval	0000:00:00.000
Runtime configurations	
Security	
Minimum message security mode	None
Minimum security policy	None

Configure the client

- Using the OPCUA Tag importer, it's possible to import not only variables but also objects!
- For example, select the entire folders Model, Alarms, Loggers and DataStores
- Elements like Alarms, Datalogger and AlarmsEventLogger can be used as they were part of the Client project

OPC-UA > OPCUAClient1 > **OPCUATagImporter**

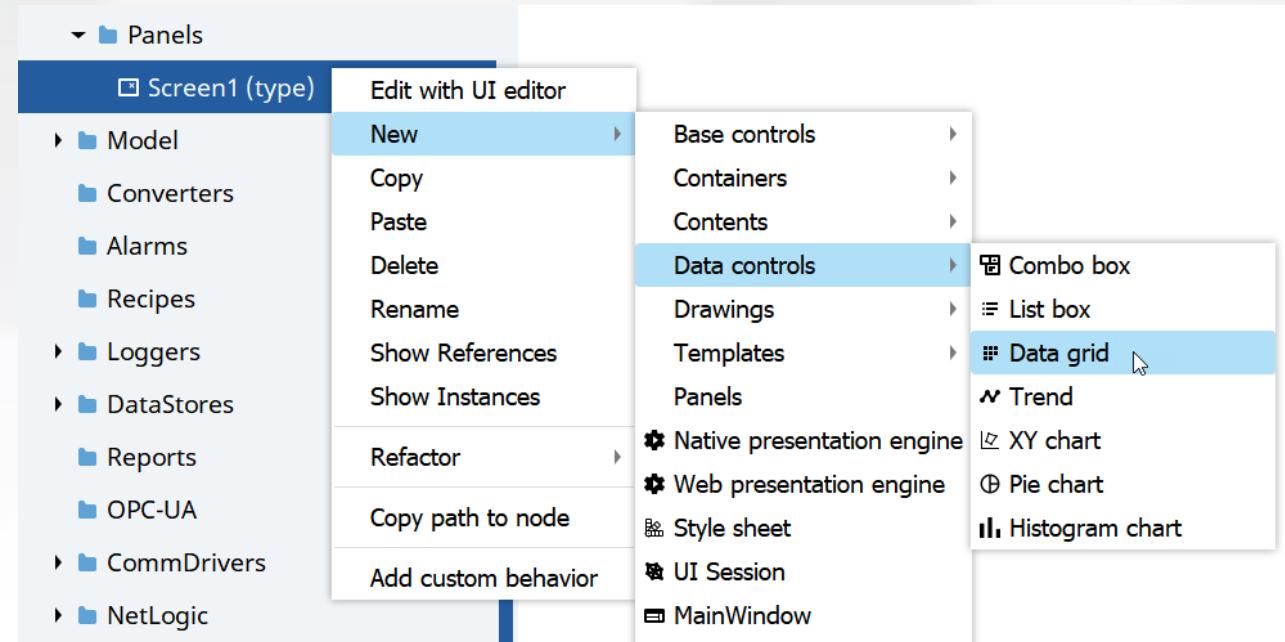
Online Type to search...  

<input checked="" type="checkbox"/>	BoilerDemo	Project folder
<input type="checkbox"/>	var BranchingEnabled	Variable Boolean
<input type="checkbox"/>	var Locales	Variable String[0]
<input type="checkbox"/>	var LocaleFallbackList	Variable String[0]
> <input type="checkbox"/>	UI	Folder
> <input checked="" type="checkbox"/>	Model	Folder
<input type="checkbox"/>	Converters	Folder
> <input checked="" type="checkbox"/>	Alarms	Folder
> <input type="checkbox"/>	Recipes	Folder
> <input checked="" type="checkbox"/>	Loggers	Folder
> <input checked="" type="checkbox"/>	DataStores	Folder
<input type="checkbox"/>	Reports	Folder
> <input type="checkbox"/>	OPC-UA	Folder
<input type="checkbox"/>	var MeasurementSystemMap	Variable MeasurementSyste...
> <input type="checkbox"/>	PasswordPolicy	Password policy



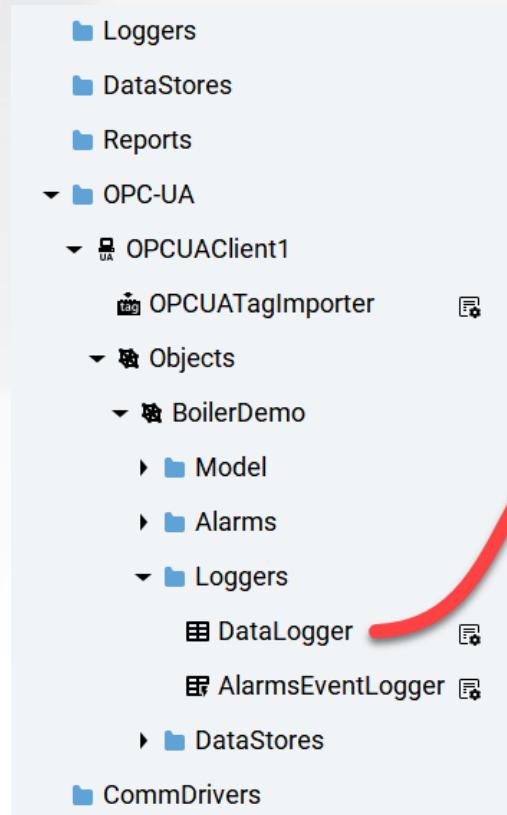
Configure the client

- Using the OPCUA Tag importer, it's possible to import not only variables but also objects!
- For example, select the entire folders Model, Alarms, Loggers and DataStores
- Elements like Alarms, Datalogger and AlarmsEventLogger can be used as they were part of the Client project



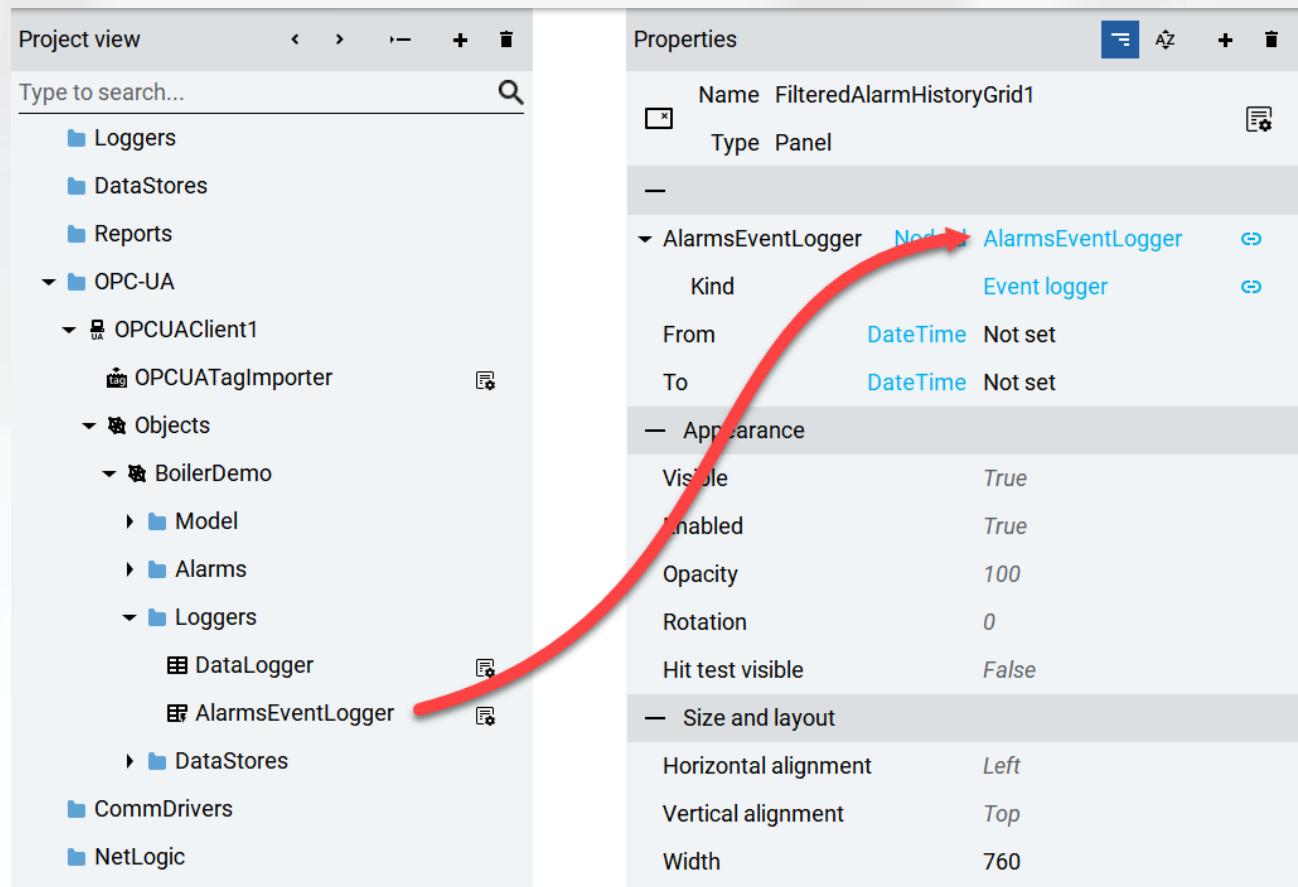
Configure the client

- Using the OPCUA Tag importer, it's possible to import not only variables but also objects!
- For example, select the entire folders Model, Alarms, Loggers and DataStores
- Elements like Alarms, Datalogger and AlarmsEventLogger can be used as they were part of the Client project



Configure the client

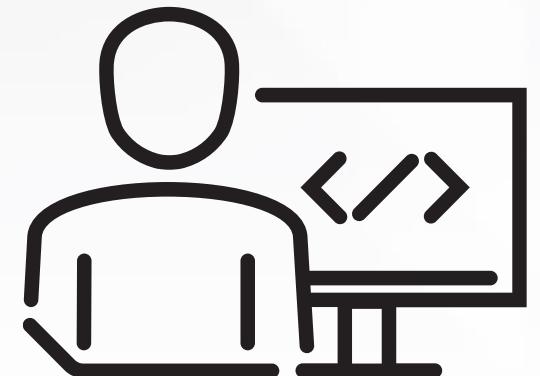
- Using the OPCUA Tag importer, it's possible to import not only variables but also objects!
- For example, select the entire folders Model, Alarms, Loggers and DataStores
- Elements like Alarms, Datalogger and AlarmsEventLogger can be used as they were part of the Client project



Properties	
Name	FilteredAlarmHistoryGrid1
Type	Panel
AlarmsEventLogger	<i>Node</i> → AlarmsEventLogger
Kind	Event logger
From	DateTime Not set
To	DateTime Not set
Appearance	
Visible	True
Enabled	True
Opacity	100
Rotation	0
Hit test visible	False
Size and layout	
Horizontal alignment	Left
Vertical alignment	Top
Width	760

Hands-on session

- Use the BoilerDemo project as Server
- Create a new project that acts as a Client
- Try to visualize and manage the Server's Alarms, Alarm History and Datalogger from the Client project

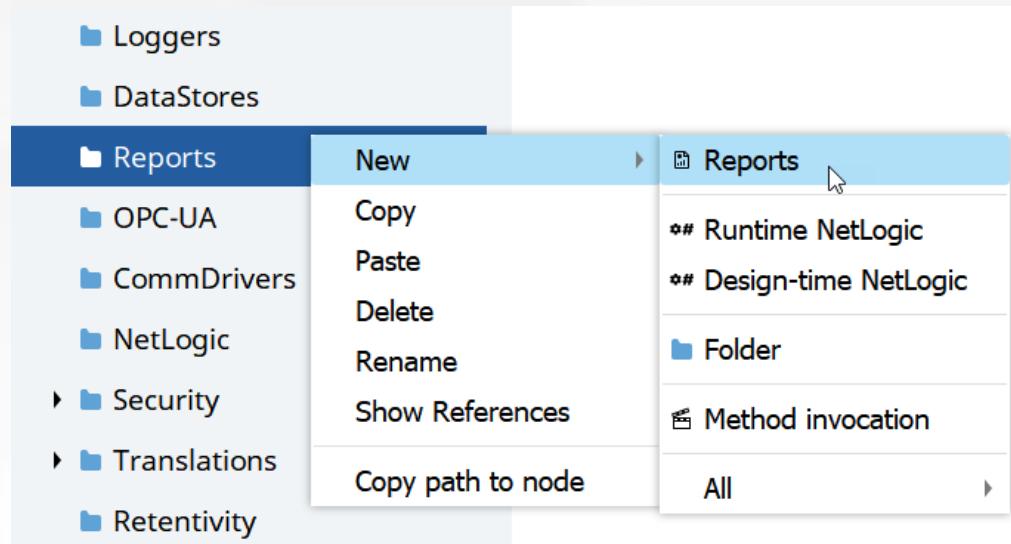


Configure Reports



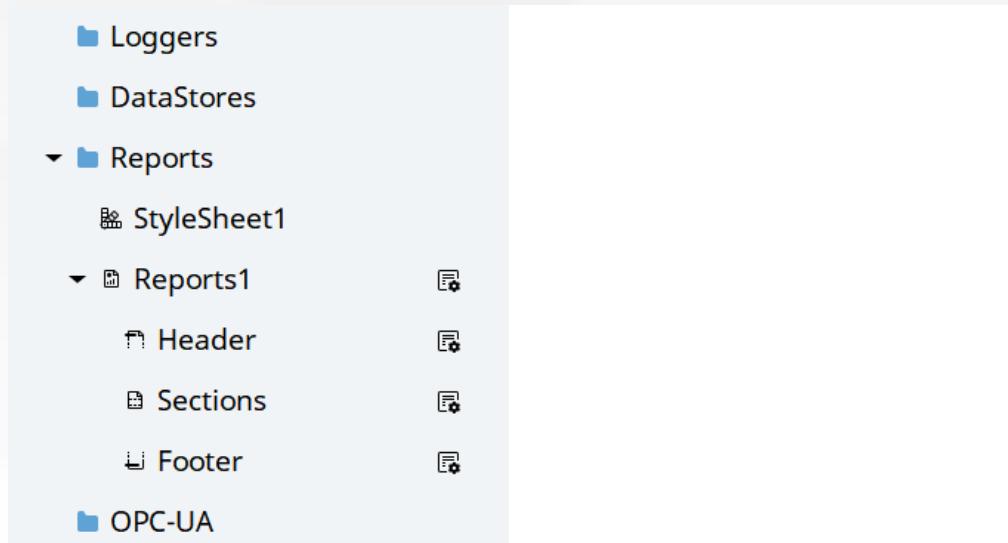
Reports

- Design and generate reports in PDF format
- Data coming from QApplication and/or from a database
- Components:
 - StyleSheet
 - Header
 - Sections
 - Footer



Reports

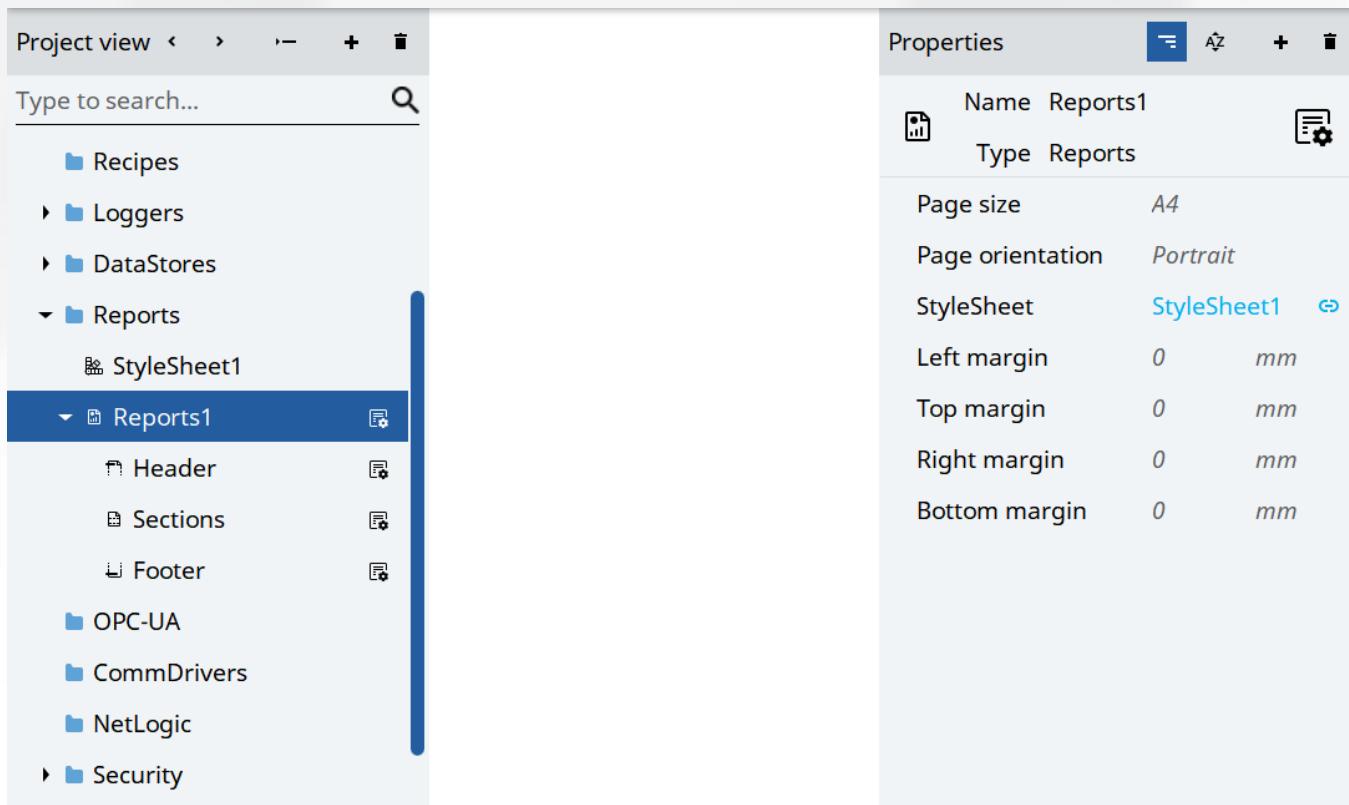
- Design and generate reports in PDF format
- Data coming from QApplication and/or from a database
- Components:
 - StyleSheet
 - Header
 - Sections
 - Footer



Report properties

- Allow setting the report's global layout properties, which apply to all pages of the final PDF document

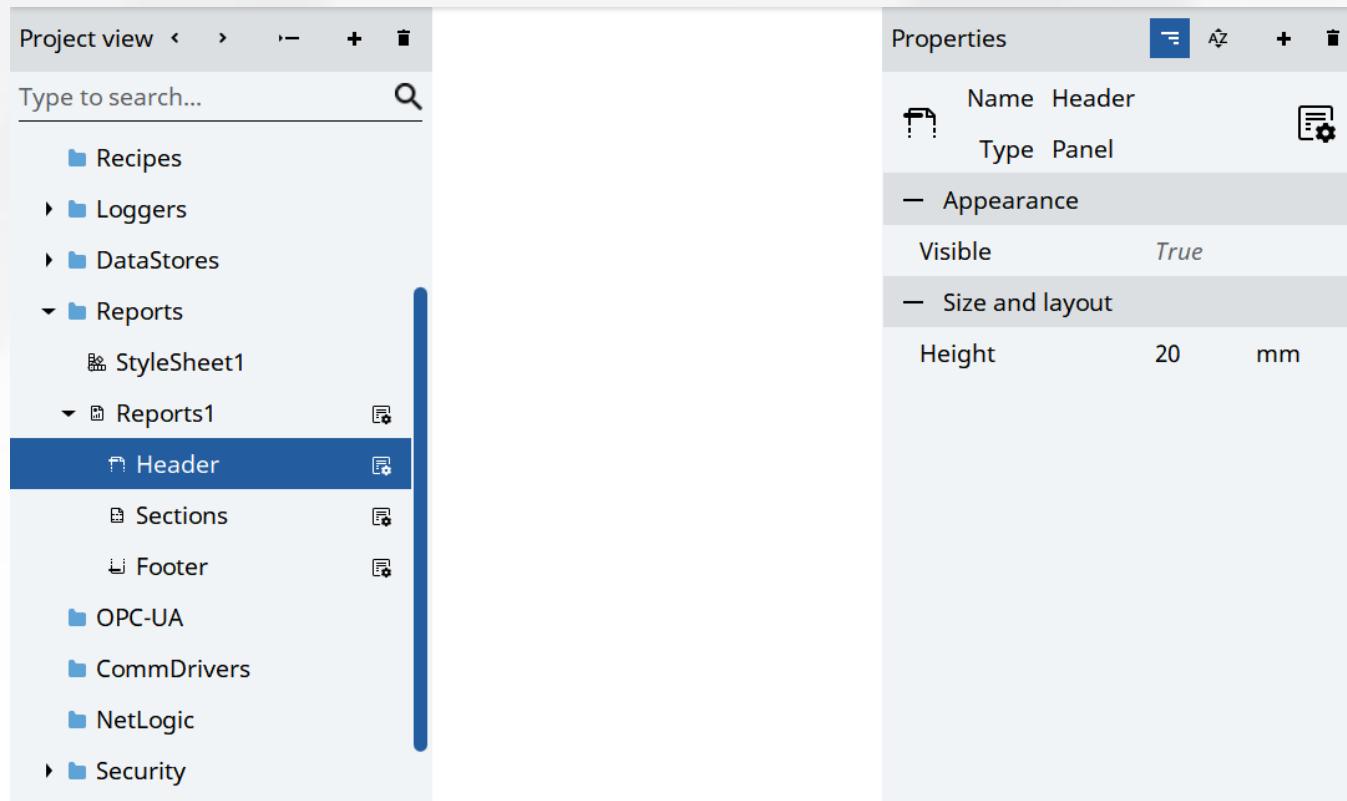
- Page size,
- Page orientation,
- Style sheet used,
- Page margins



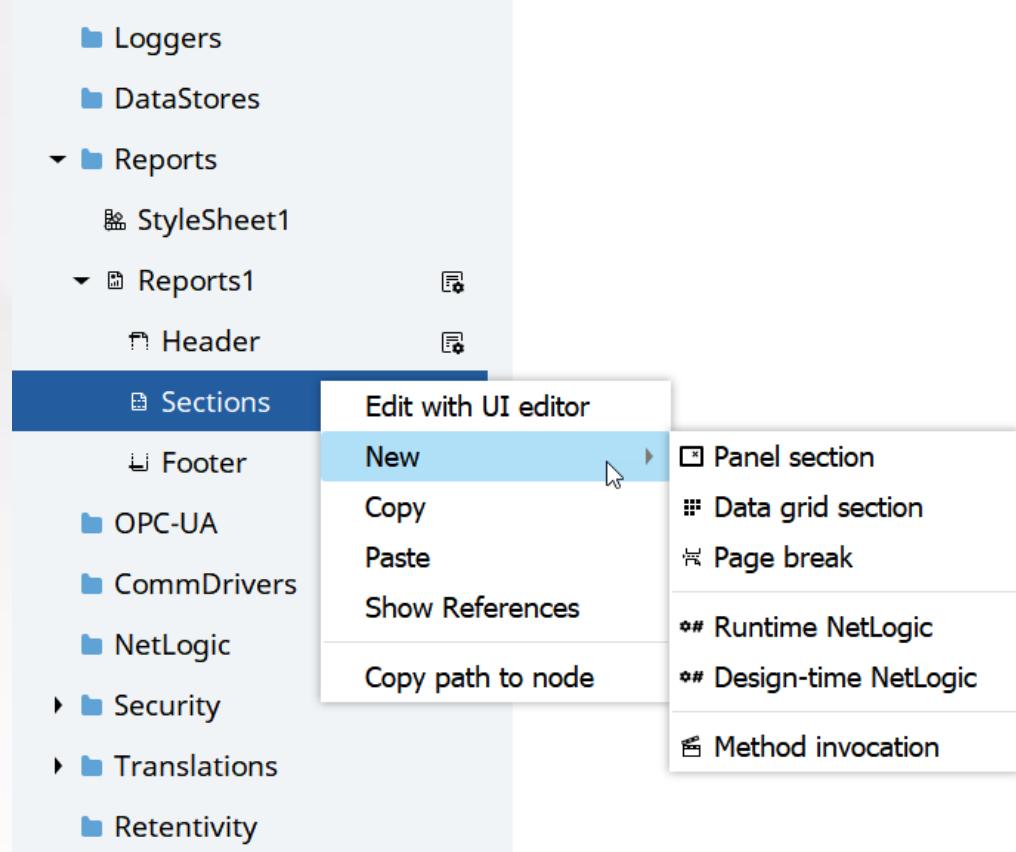
Report header and footer

- Header/Footer is content that appears at the top/bottom of a report page
- Properties:

- Visible
- Height

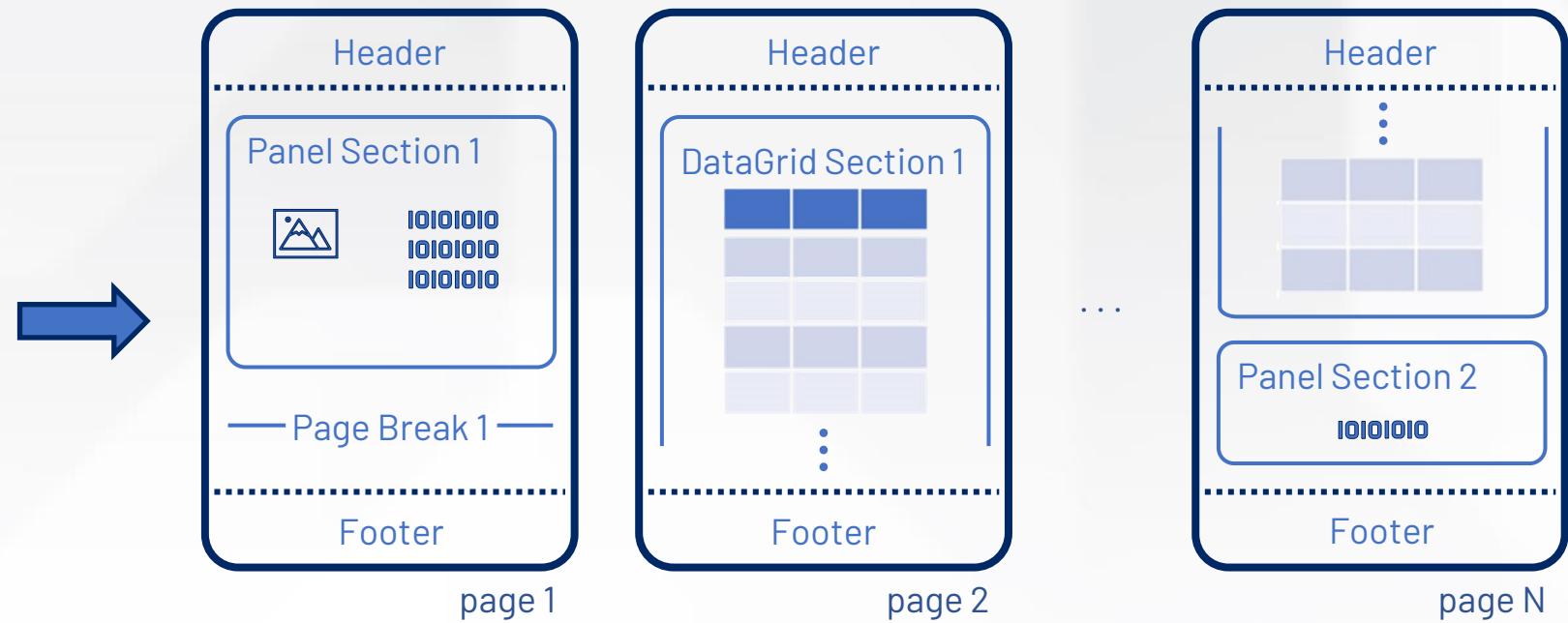
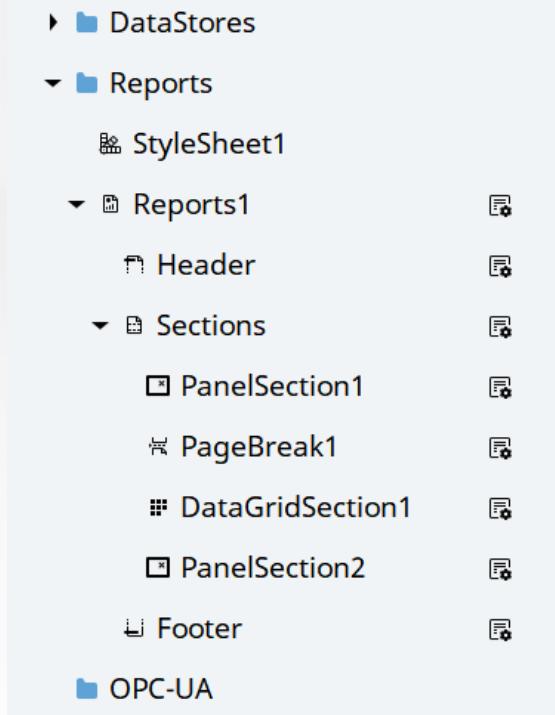


Report sections



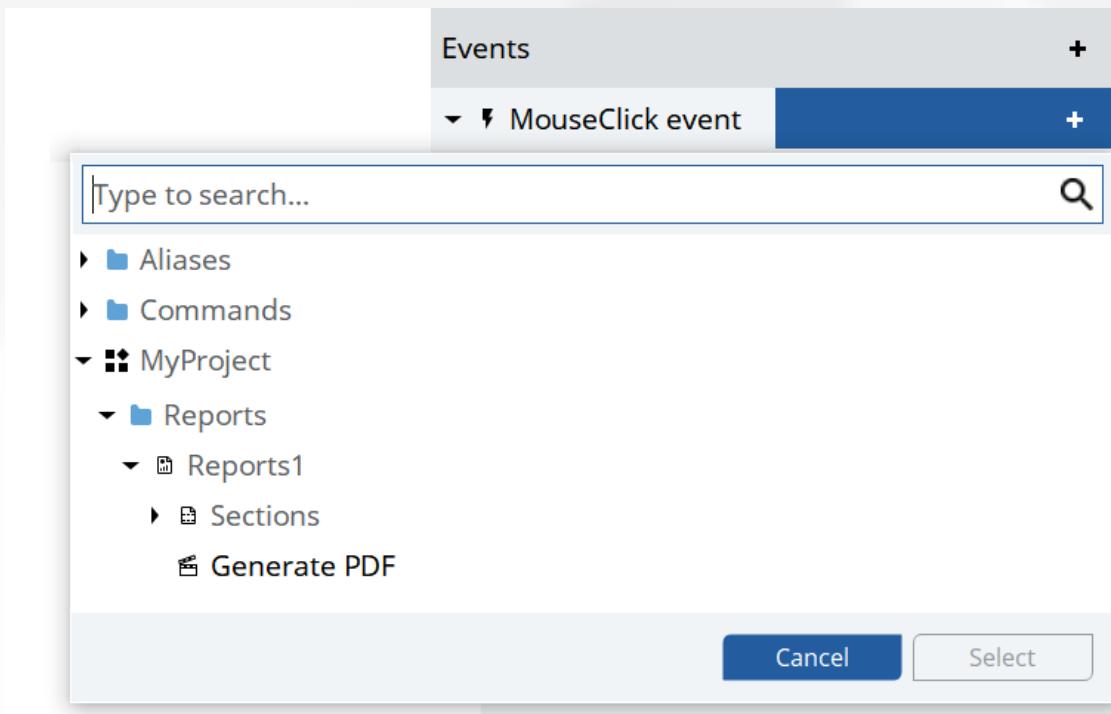
- **Panel:** container of graphical objects as Label, Rectangle, Panel, Image. Can be used to show variable values.
- **DataGrid:** container of data read from a database in table format. Can be used to show historical data collected by a Datalogger or Alarms.
- **Page break:** placeholder that forces the report to continue to another page.

Report sections example



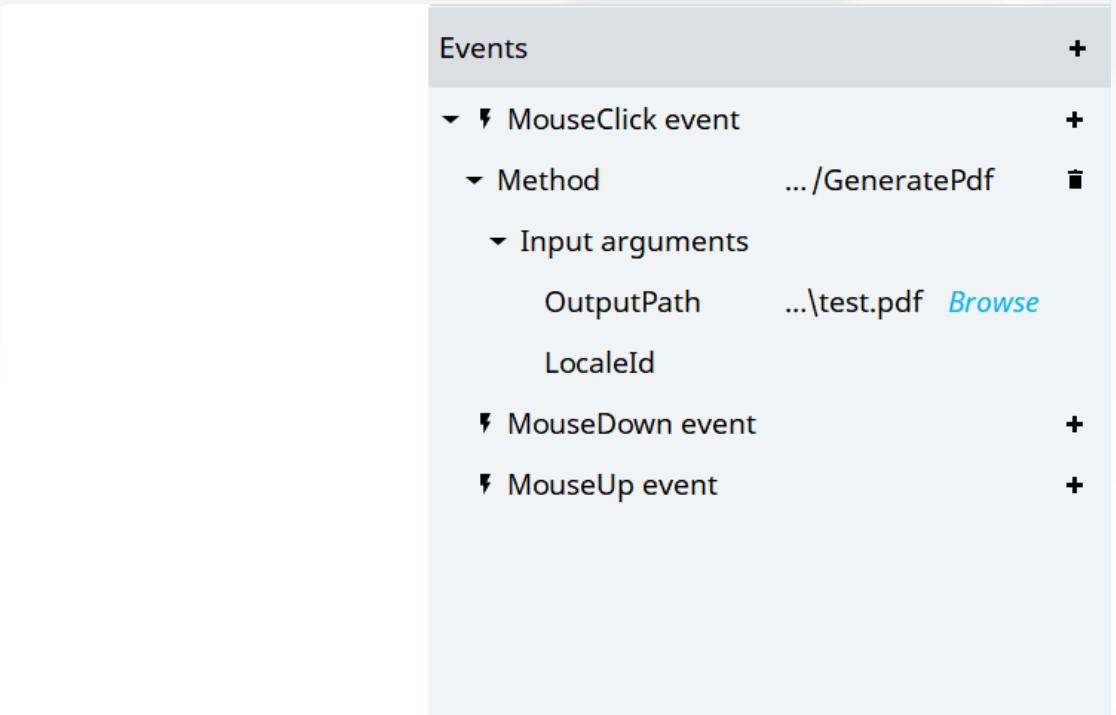
Generate report at runtime

- Report resource expose method "Generate PDF"
- Can be called for example on Mouse Click event
- Properties:
 - Output Path: the path of the PDF file
 - Locale Id (optional): language to be used inside report



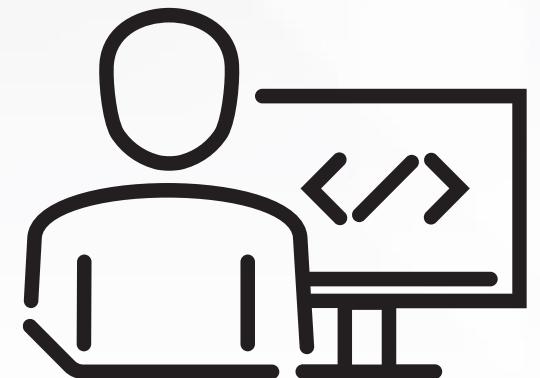
Generate report at runtime

- Report resource expose method "Generate PDF"
- Can be called for example on Mouse Click event
- Properties:
 - Output Path: the path of the PDF file
 - Locale Id (optional): language to be used inside report



Hands-on session

- Define a report that includes a Datagrid to show values from a datalogger table
- Add a button that calls the "Generate PDF" method



Use Retentivity



Retentivity usage

- Provide storage of changes made at Runtime

- Retentivity applies to:

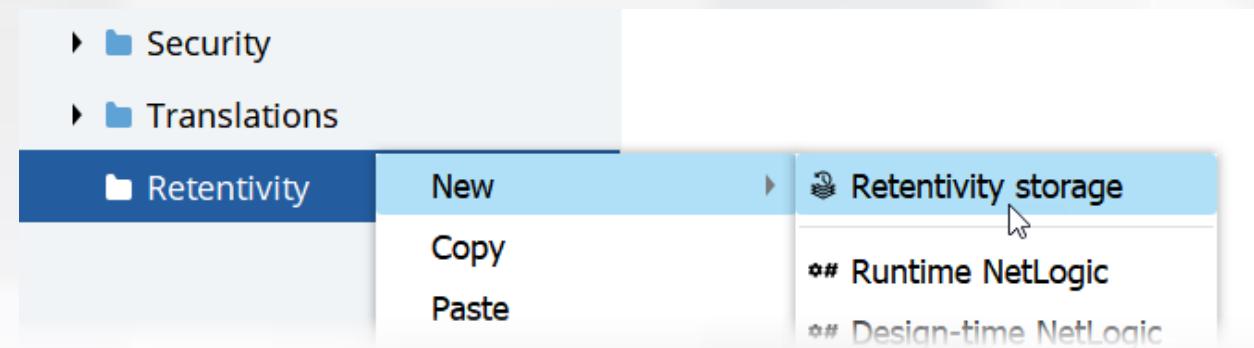
- **Variables**,

- **Users**,

- **Dictionary of Translations**,

- or every node of the project

- Every Retentivity storage object is related to an embedded database



Retentivity properties and location

- Properties

- Nodes:** Folder/node of the project to be retained.
It's possible to define several nodes on the same Retentivity Storage, or create separated storage
- Write delay:** define how often to save into the database
Zero means values are saved as soon as a change is detected
- Delta observer enabled:** allow to enable/disable the Retentivity

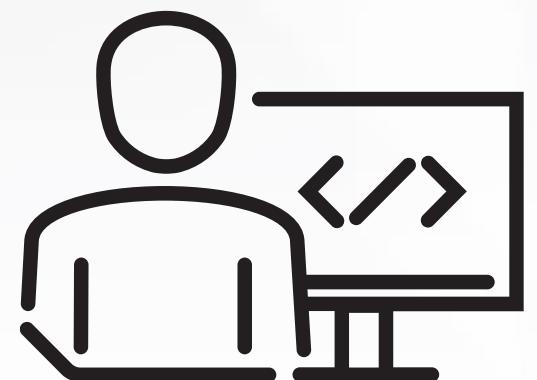
Properties	
Name	RetentivityStorage1
Type	Retentivity storage
Nodes	
Node1	NodeId Model
Node2	NodeId Security
Node3	NodeId LocalizationDictionary1
Write delay	0000:00:00.000
Delta observer enabled	<i>True</i>

- Location:

- Database file can be found in the ApplicationFiles folder of QApplication
- Example: %localappdata%\Rockwell Automation\FactoryTalk Optix\Emulator\Projects\<ProjectName>\ApplicationFiles

Hands-on session

- Add retentivity storage to the project
- Add a project's node to the retentivity (like the Model folder)
- Verify that changes made on Model's variable are reteined

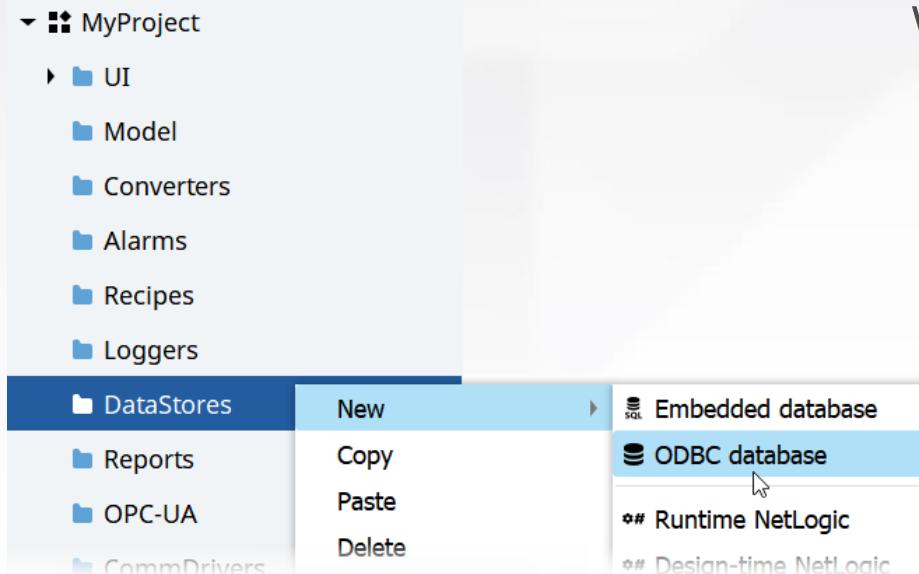


Connect to an External Database



ODBC Database

- It's used to query/populate a relational database external to the QApplication
- Supported DBMS:
 - SQL Server or SQL Server Express 2012 (or higher)
 - MySQL Server



- Objects that can exchange data with an ODBC database:

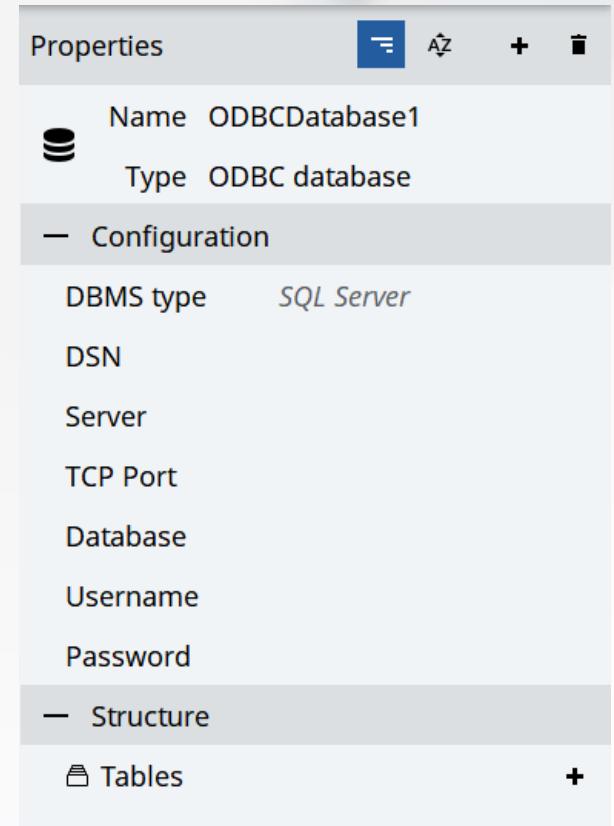
- *Data grid, List box, Combo box and Trend objects* query a table to display data at runtime

- *Data logger and Event logger objects* store data in a table

- *Recipe object* stores data in a table and execute a query to display data at runtime

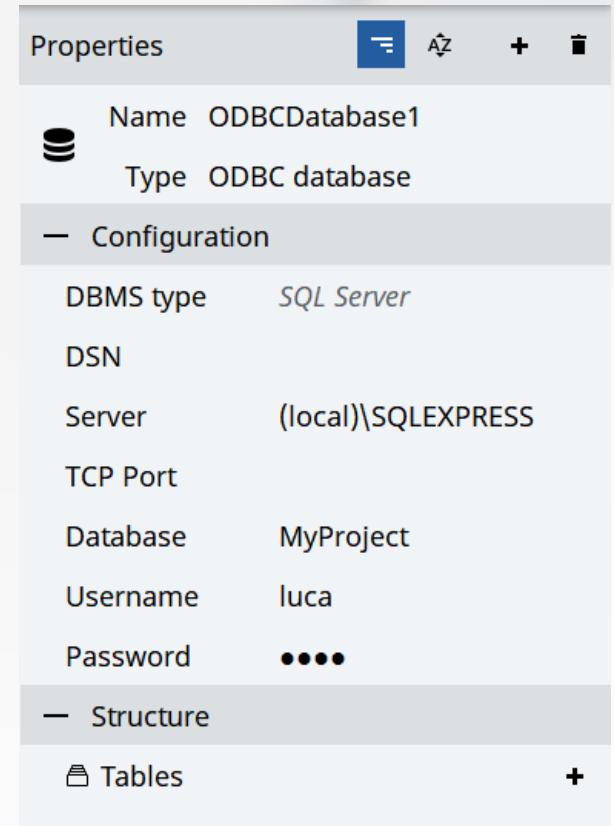
ODBC Database

- DBMS type: SQL Express or MySQL
- DSN (optional): only in case ODBC is used
- Server: can be IP Address, Hostname or "Hostname\Instance Name"
- TCP Port (optional): to be used if the port is different from the default
- Database: Name of the database
 - NOTE: database must be created manually**
- Username/Password: credential of the user, used to read/write into database tables

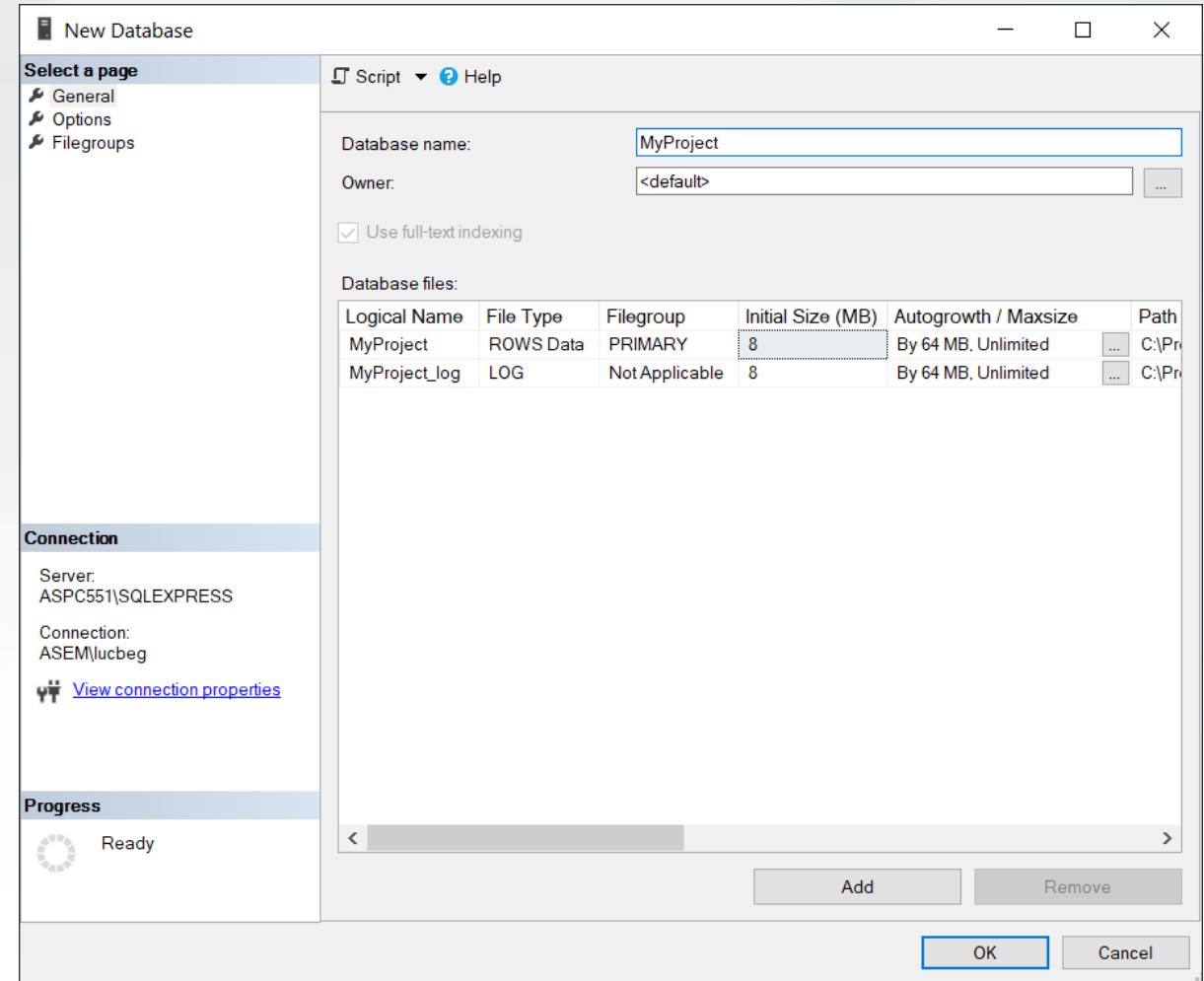
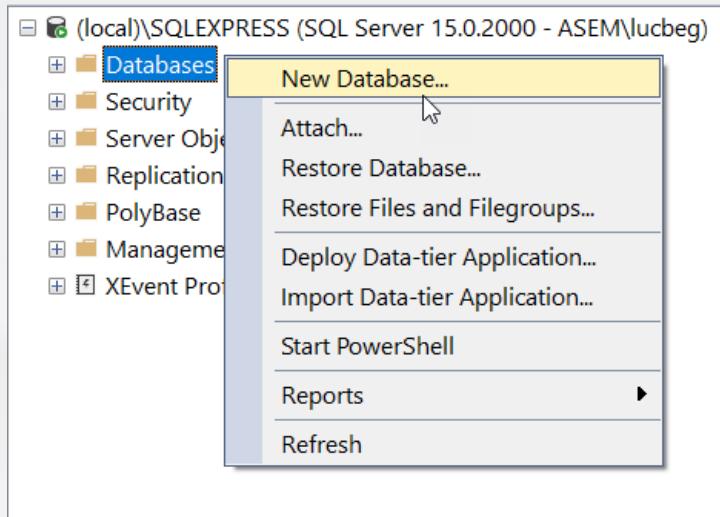


ODBC Database

- DBMS type: SQL Express or MySQL
- DSN (optional): only in case ODBC is used
- Server: can be IP Address, Hostname or "Hostname\Instance Name"
- TCP Port (optional): to be used if the port is different from the default
- Database: Name of the database
 - NOTE: database must be created manually**
- Username/Password: credential of the user, used to read/write into database tables



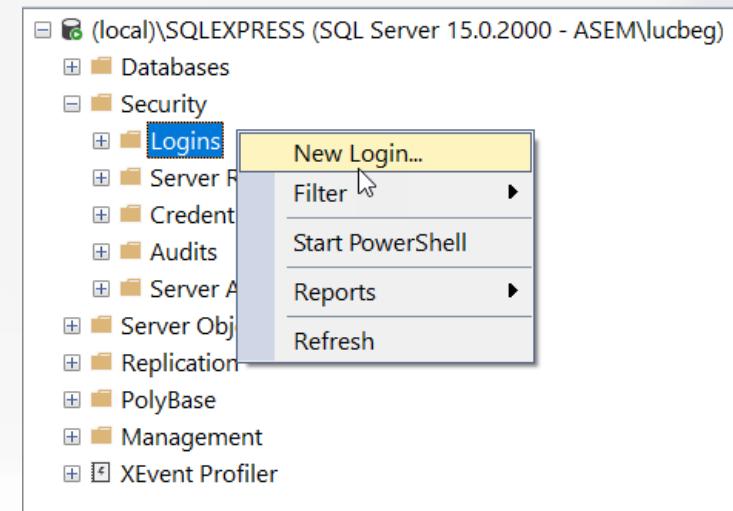
SQL Server: create database via SQL Server Management Studio



- Is suggested to create the database before the creation of the user
- To create a database
Databases > New Database...
and is enough to provide the name

SQL Server: create user via SQL Server Management Studio

1. Security > Logins > New Login...
2. Once created a User, it's necessary to assign a role for the needed database
3. User Mapping > Database > db_owner

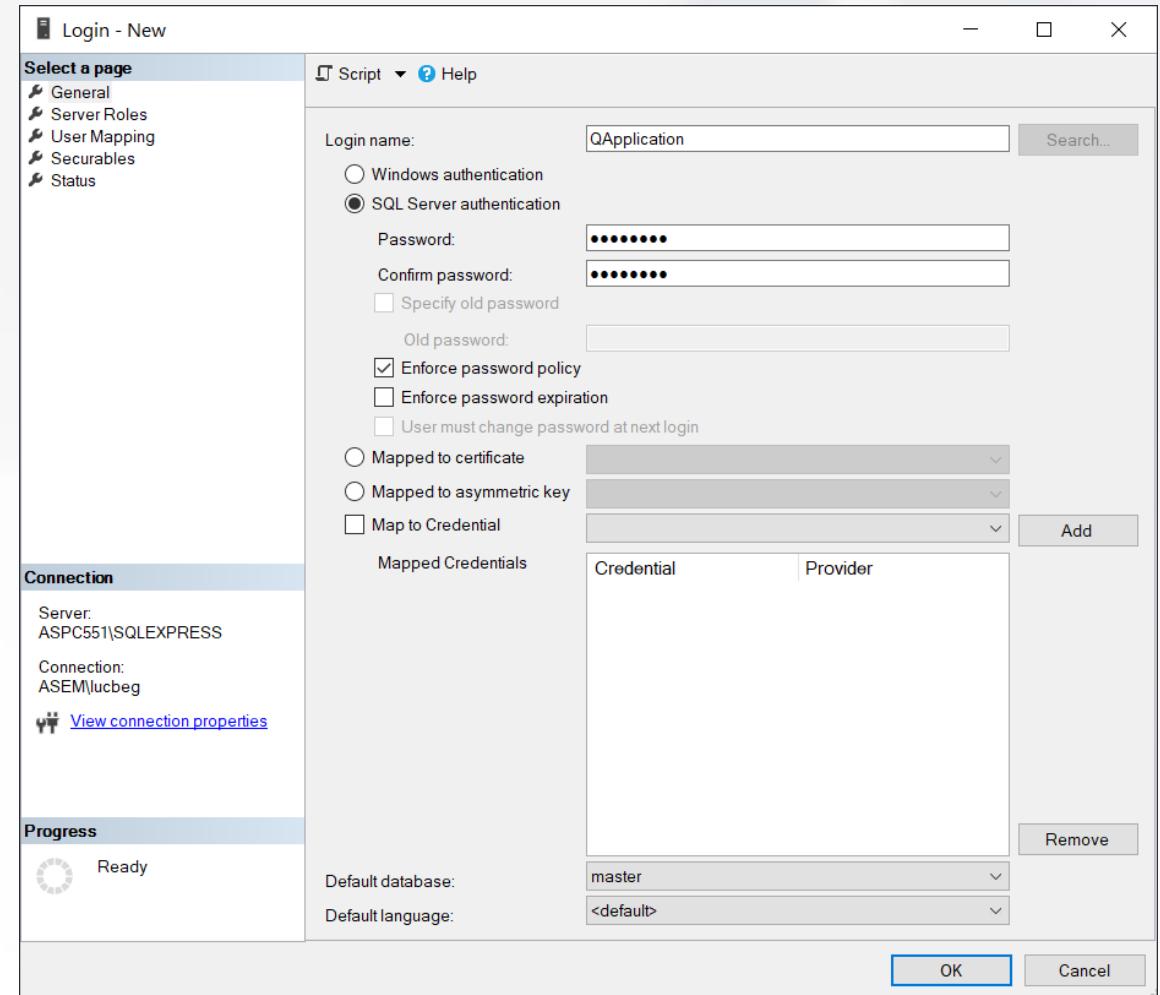


SQL Server: create user via SQL Server Management Studio

1. Security > Logins > New Login...

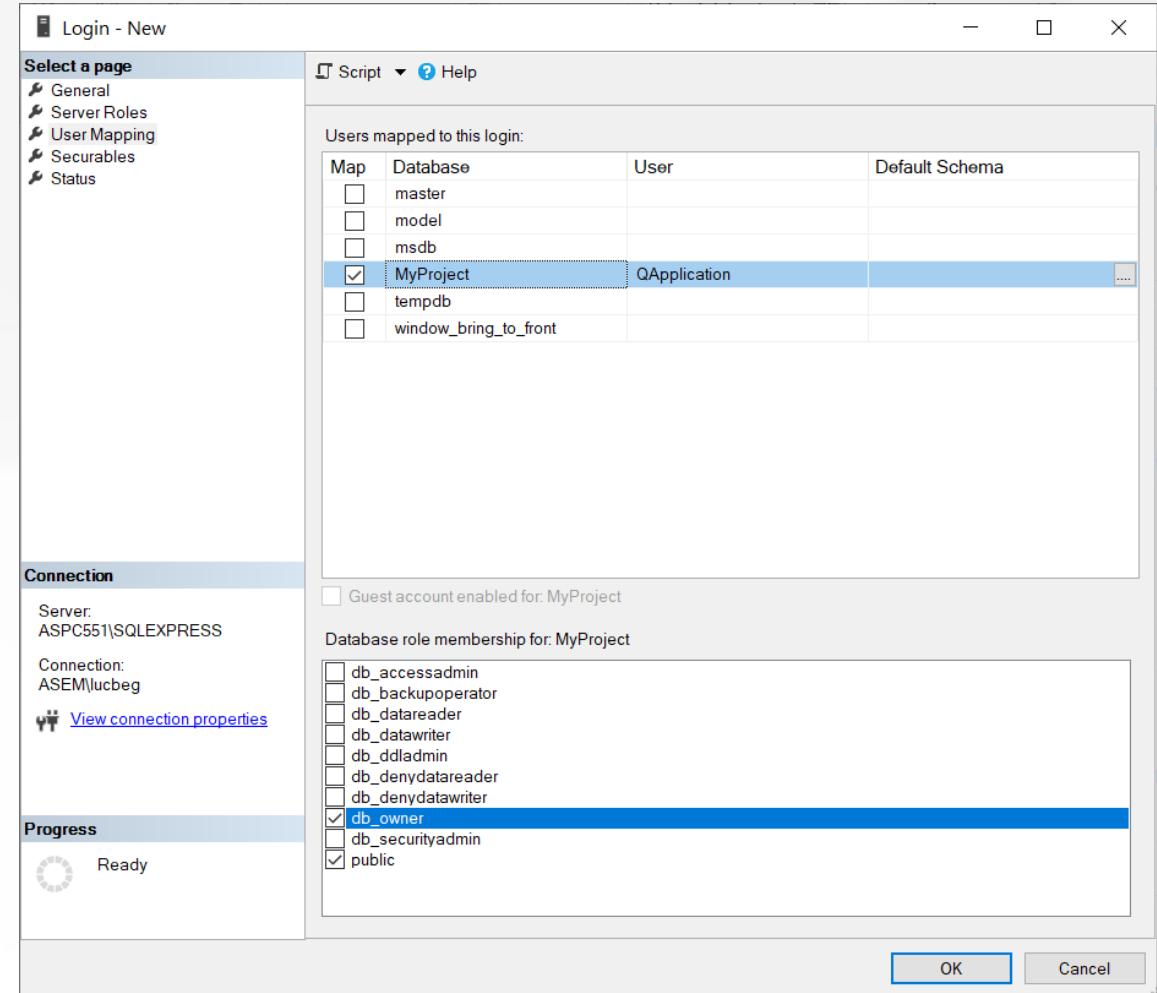
2. Once created a User, it's necessary
to assign a role for the needed database

3. User Mapping > Database > db_owner



SQL Server: create user via SQL Server Management Studio

1. Security > Logins > New Login...
2. Once created a User, it's necessary to assign a role for the needed database
3. User Mapping > Database > db_owner



Usage of ODBC Data Source (optional)

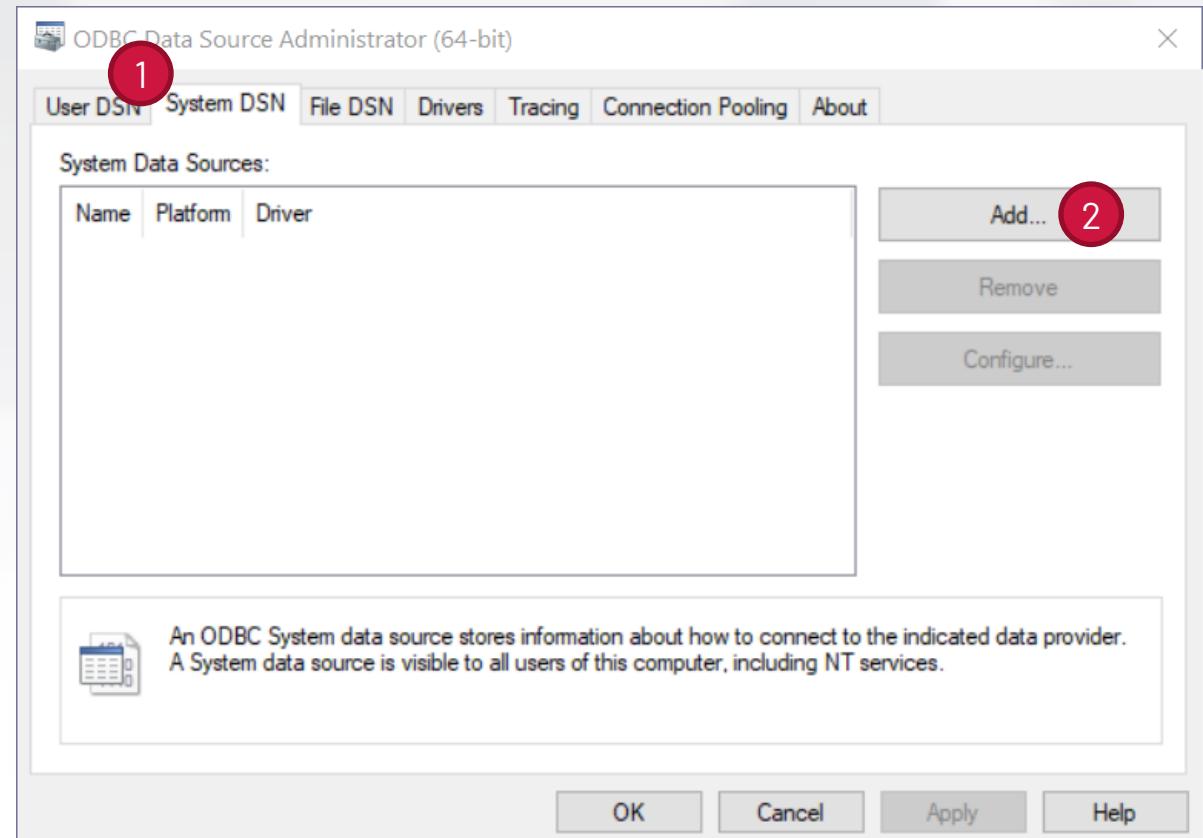
- To configure a DSN
use "ODBC Data Sources (64-bit)"

- ODBC Driver supported:

- SQL Server
- MySQL ANSI
- MySQL Unicode

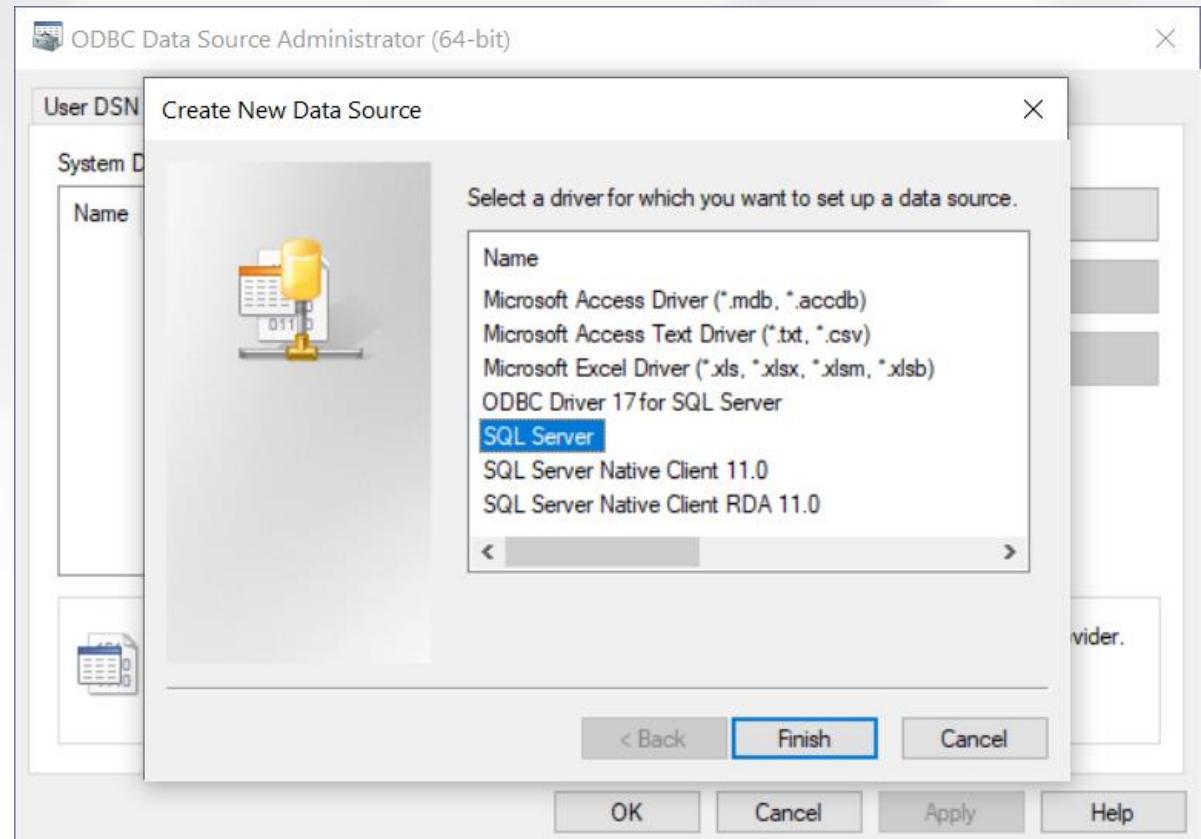
- Key settings:

- DSN Name: defined into ODBC DSN,
must be the same in FTOptix Studio
- Default Database: is the Database created
for the QApplication
- Username/Password: valid credential to access the Database



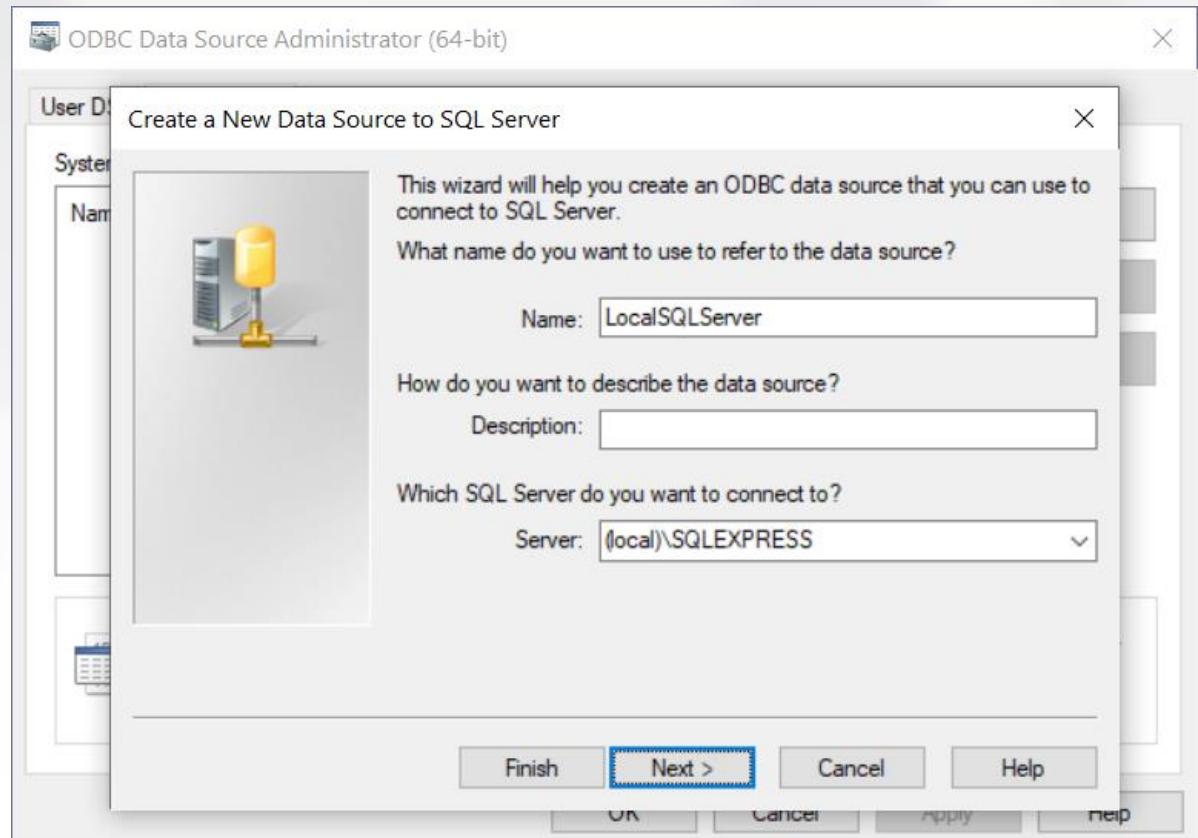
Usage of ODBC Data Source (optional)

- To configure a DSN
use "ODBC Data Sources (64-bit)"
- ODBC Driver supported:
 - SQL Server
 - MySQL ANSI
 - MySQL Unicode
- Key settings:
 - DSN Name: defined into ODBC DSN, must be the same in FTOptix Studio
 - Default Database: is the Database created for the QApplication
 - Username/Password: valid credential to access the Database



Usage of ODBC Data Source (optional)

- To configure a DSN
use "ODBC Data Sources (64-bit)"
- ODBC Driver supported:
 - SQL Server
 - MySQL ANSI
 - MySQL Unicode
- Key settings:
 - DSN Name: defined into ODBC DSN, must be the same in FTOptix Studio
 - Default Database: is the Database created for the QApplication
 - Username/Password: valid credential to access the Database



Usage of ODBC Data Source (optional)

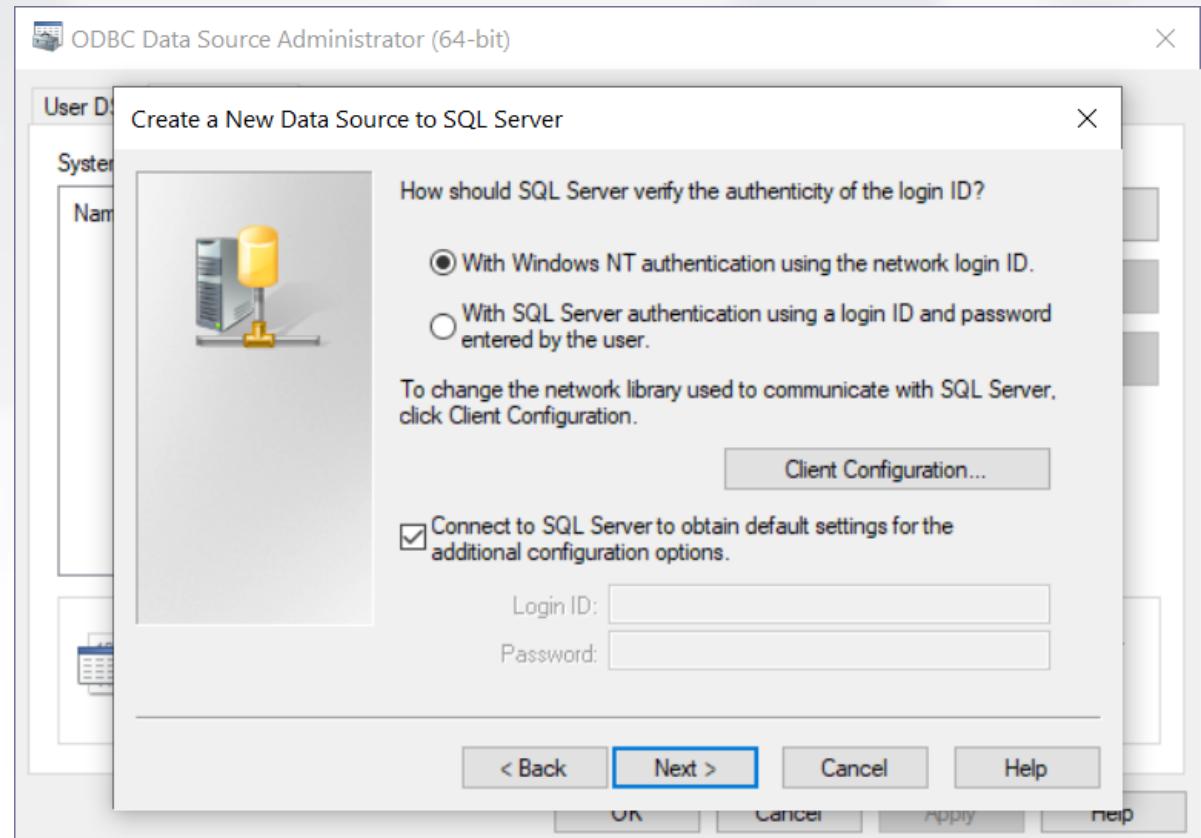
- To configure a DSN
use "ODBC Data Sources (64-bit)"

- ODBC Driver supported:

- SQL Server
- MySQL ANSI
- MySQL Unicode

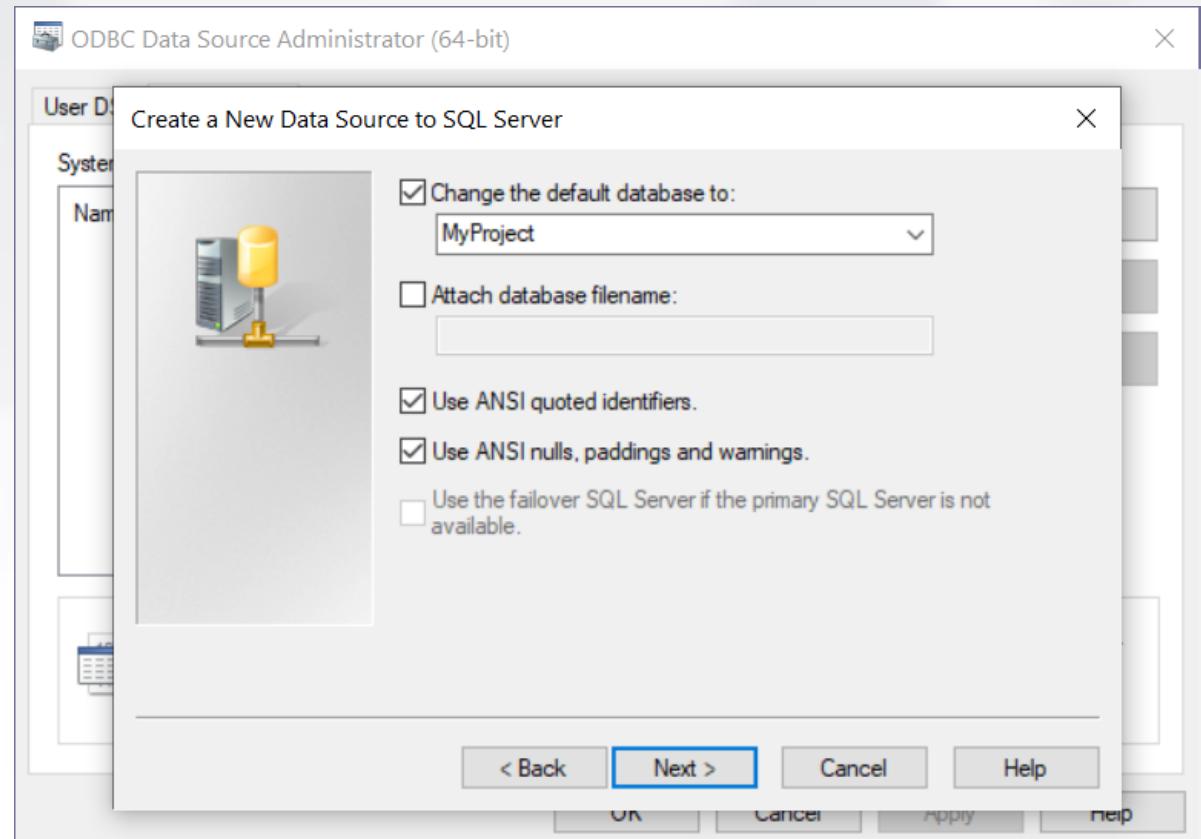
- Key settings:

- DSN Name: defined into ODBC DSN,
must be the same in FTOptix Studio
- Default Database: is the Database created
for the QApplication
- Username/Password: valid credential to access the Database



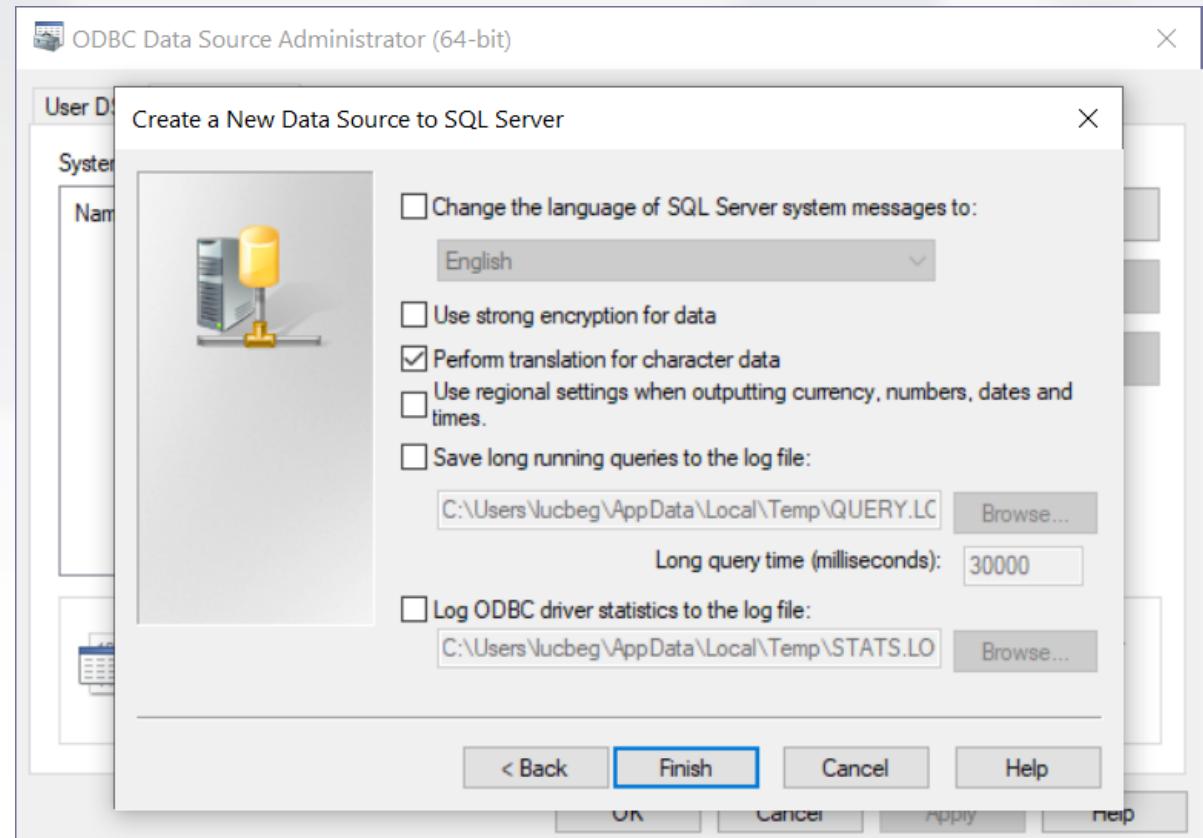
Usage of ODBC Data Source (optional)

- To configure a DSN
use "ODBC Data Sources (64-bit)"
- ODBC Driver supported:
 - SQL Server
 - MySQL ANSI
 - MySQL Unicode
- Key settings:
 - DSN Name: defined into ODBC DSN, must be the same in FTOptix Studio
 - Default Database: is the Database created for the QApplication
 - Username/Password: valid credential to access the Database



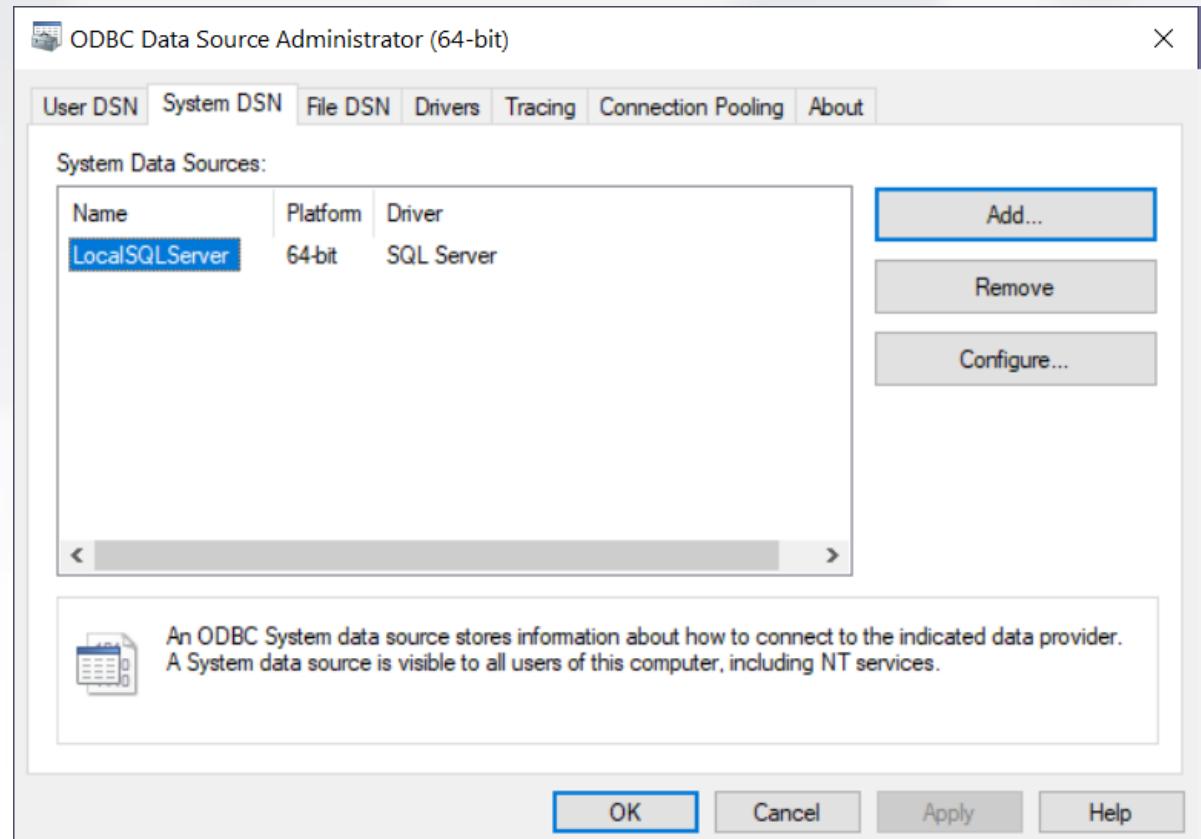
Usage of ODBC Data Source (optional)

- To configure a DSN
use "ODBC Data Sources (64-bit)"
- ODBC Driver supported:
 - SQL Server
 - MySQL ANSI
 - MySQL Unicode
- Key settings:
 - DSN Name: defined into ODBC DSN, must be the same in FTOptix Studio
 - Default Database: is the Database created for the QApplication
 - Username/Password: valid credential to access the Database



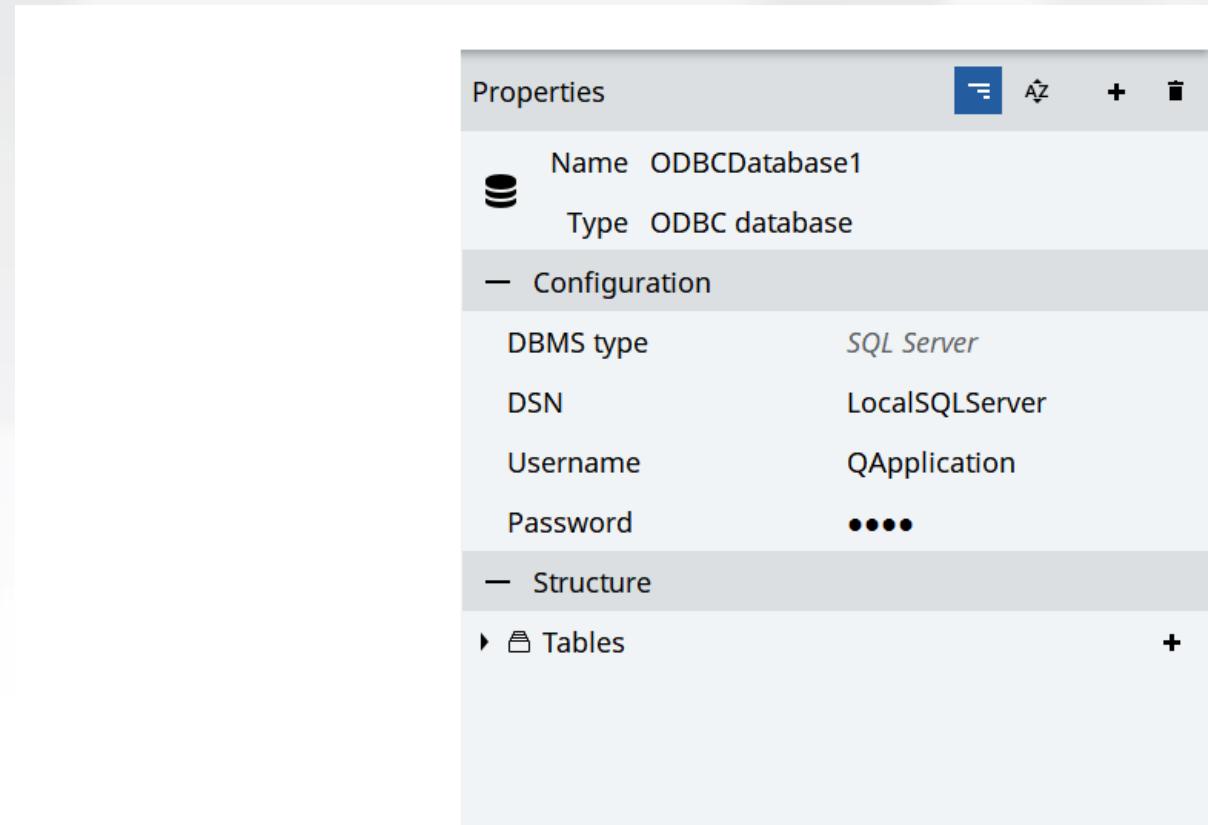
Usage of ODBC Data Source (optional)

- To configure a DSN
use "ODBC Data Sources (64-bit)"
- ODBC Driver supported:
 - SQL Server
 - MySQL ANSI
 - MySQL Unicode
- Key settings:
 - DSN Name: defined into ODBC DSN, must be the same in FTOptix Studio
 - Default Database: is the Database created for the QApplication
 - Username/Password: valid credential to access the Database



Usage of ODBC Data Source (optional)

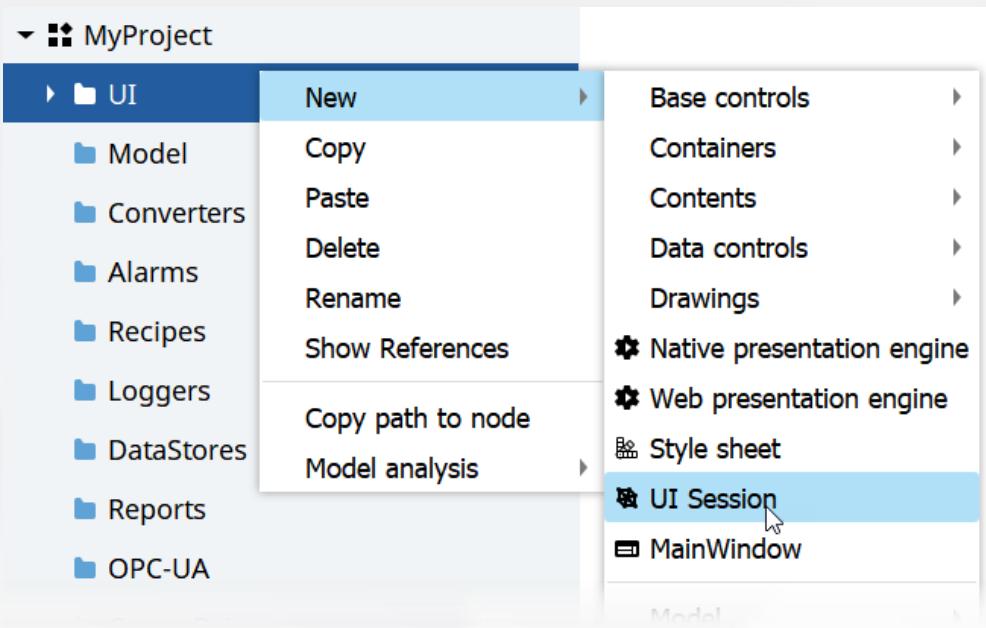
- To configure a DSN
use "ODBC Data Sources (64-bit)"
- ODBC Driver supported:
 - SQL Server
 - MySQL ANSI
 - MySQL Unicode
- Key settings:
 - DSN Name: defined into ODBC DSN, must be the same in FTOptix Studio
 - Default Database: is the Database created for the QApplication
 - Username/Password: valid credential to access the Database



Define UI Session and Session variables



UI Session



The screenshot shows the 'Properties' panel for a UI Session named 'UISession1 (type)'. The properties listed are:

- Name: UISession1 (type)
- Type: UI Session
- Inactivity timeout duration: 0000:00:00.000
- Inactivity timeout active: False
- LocaleIds
- Languages
- Measurement system

Under the 'Events' section, there are two entries:

- IdleTimeoutEvent
- UserSessionEvent

- a UI Session allows defining a custom Session to be used on Native or Web presentation engine
- Allow customizing: Inactivity Timeout, Locale, Language and Measurement System

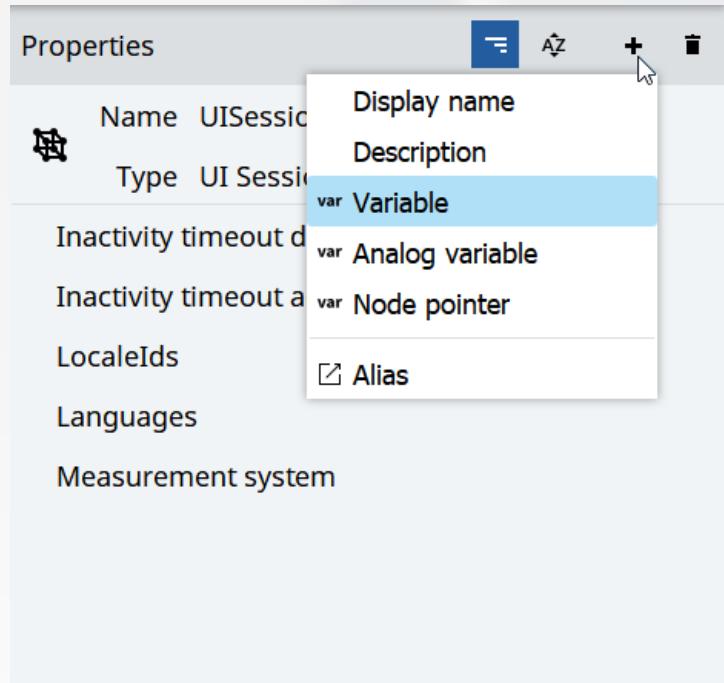
Session variables vs others

- Model/Comm. Driver variables
 - belongs to the Project context
 - display the same value regardless of sessions or panels
- Panel variables
 - belongs to the Panel context
 - different sessions show different values,
but when you change the panel the value is lost
- Session variables
 - belongs to the Session context
 - different sessions show different values,
but when you change the panel the value is retained

Properties	
Name	UISession1 (type)
Type	UI Session
Inactivity timeout duration	0000:00:00.000
Inactivity timeout active	<i>False</i>
LocaleIds	
Languages	
Measurement system	

Session variables vs others

- Model/Comm. Driver variables
 - belongs to the Project context
 - display the same value regardless of sessions or panels
- Panel variables
 - belongs to the Panel context
 - different sessions show different values,
but when you change the panel the value is lost
- Session variables
 - belongs to the Session context
 - different sessions show different values,
but when you change the panel the value is retained

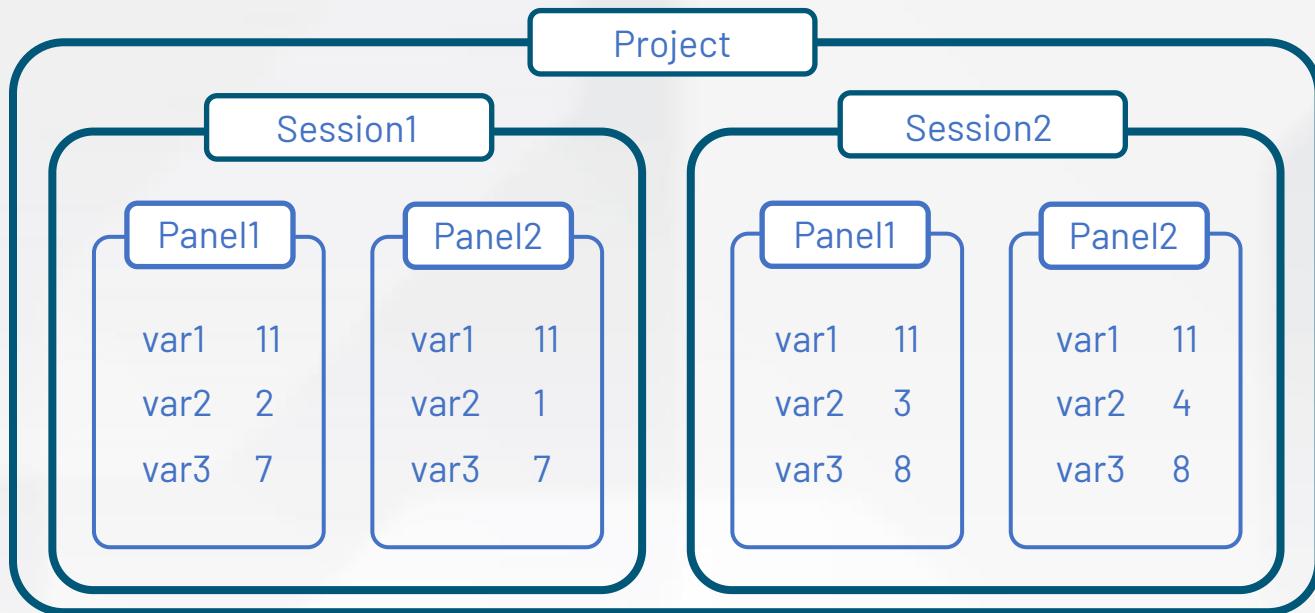


Session variables vs others

- Model/Comm. Driver variables
 - belongs to the Project context
 - display the same value regardless of sessions or panels
- Panel variables
 - belongs to the Panel context
 - different sessions show different values,
but when you change the panel the value is lost
- Session variables
 - belongs to the Session context
 - different sessions show different values,
but when you change the panel the value is retained

Properties	
Name	UISession1 (type)
Type	UI Session
Inactivity timeout duration	0000:00:00.000
Inactivity timeout active	<i>False</i>
LocaleIds	
Languages	
Measurement system	
Session_Variable1	<i>Int32</i> 0

Session variables example



var1 = Model variable
var2 = Panel variable
var3 = Session variable

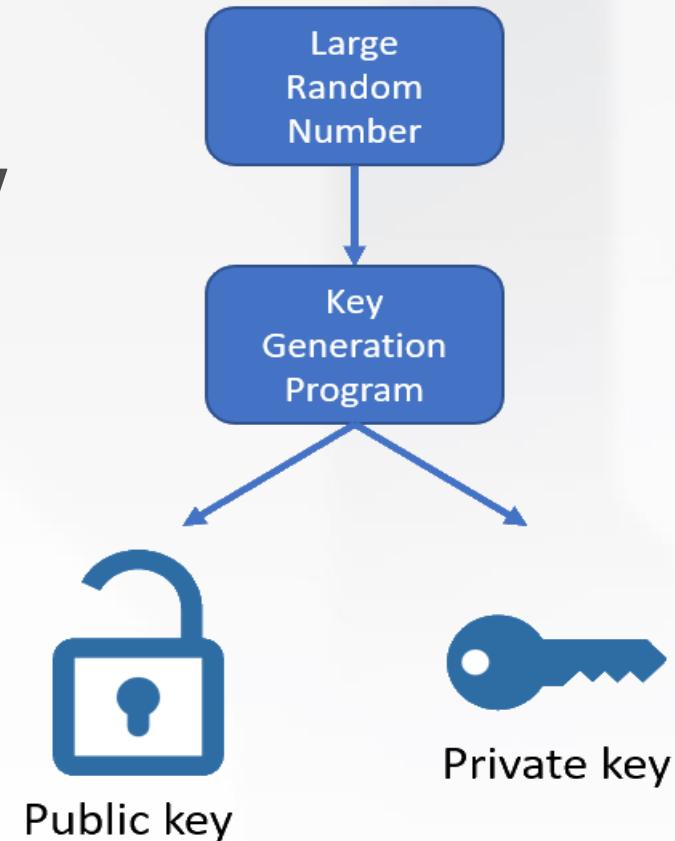
- Examples of different UI Sessions
 - Example1: one Session active on Native Presentation Engine, and another Session on the Web Presentation Engine
 - Example2: two browsers connected to the same Web Presentation Engine

Appendix: see the light about OPC-UA Security



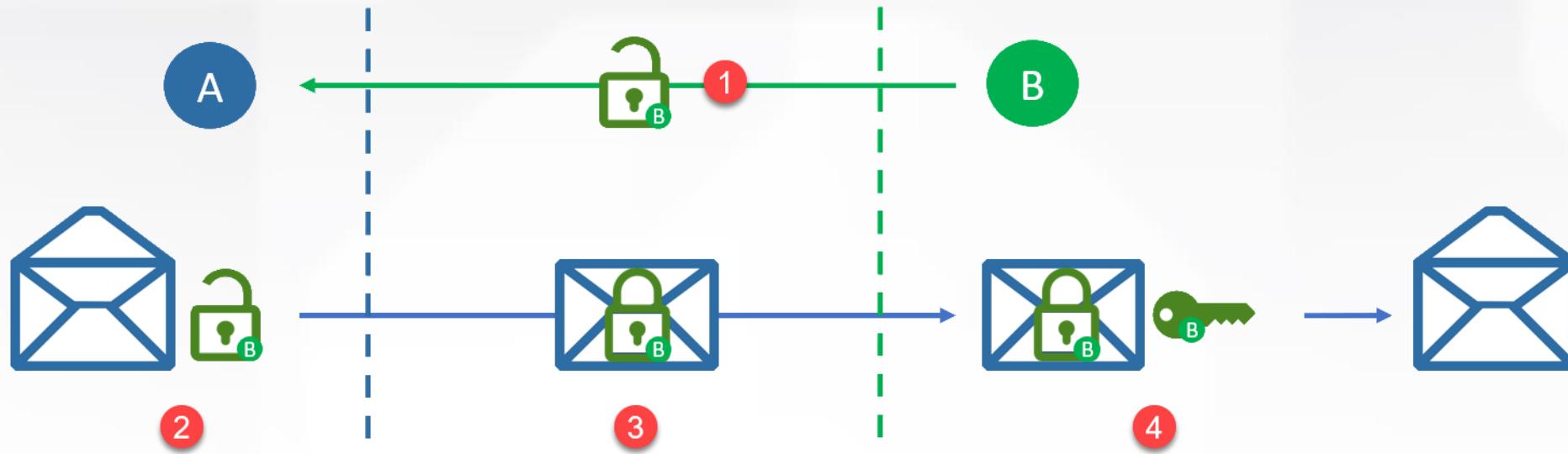
Asymmetric cryptography

- is a cryptographic system that **uses pairs of keys**
- each pair consists of a **public key** and a **private key**
- the generation of such key pairs depends on cryptographic algorithms



How does encryption work ?

1. Alice asks and receives Bob's padlock (Public Key exchange)
2. Alice uses the padlock to close the letter
3. Alice sends the locked letter to Bob
4. When Bob receives the letter, he can open it with his key (Private Key) of which he is the only owner.



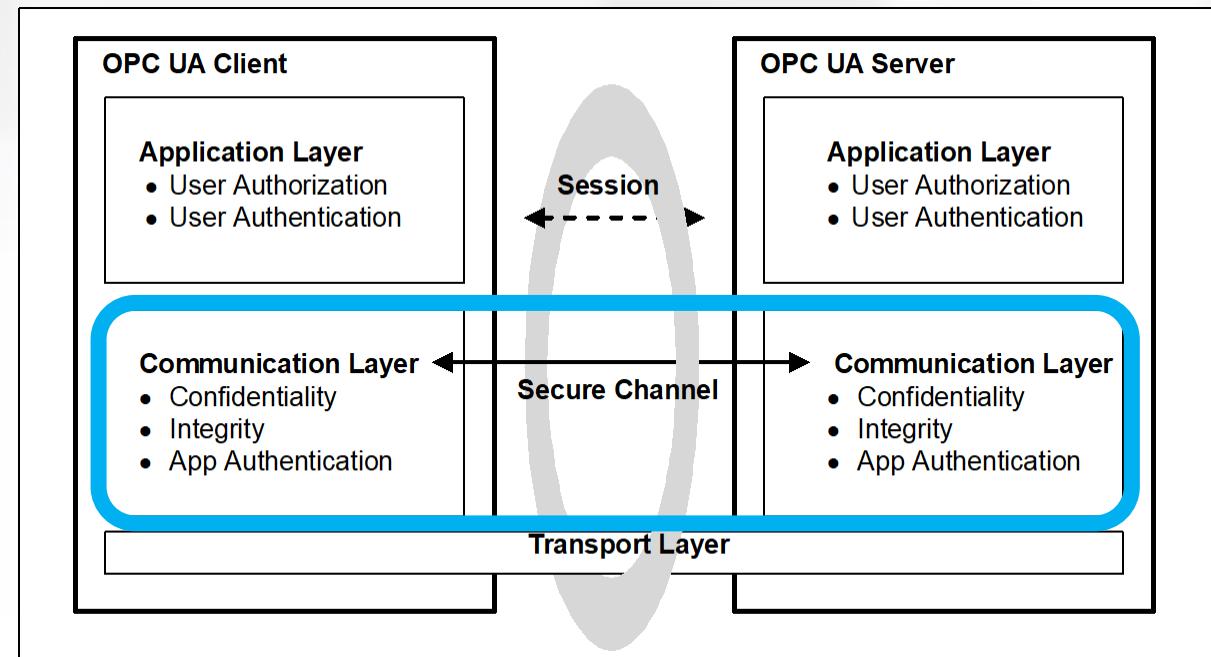
How does encryption apply to OPC-UA?

- When the OPC-UA Client connects to the OPC-UA Server:

- X.509 Certificates are exchanged so a **Secure Channel** is established

Certificates used here are called
"Application Instance Certificates"

These are the certificates exchanged when establishing a secure communication between OPC-UA Client and Server



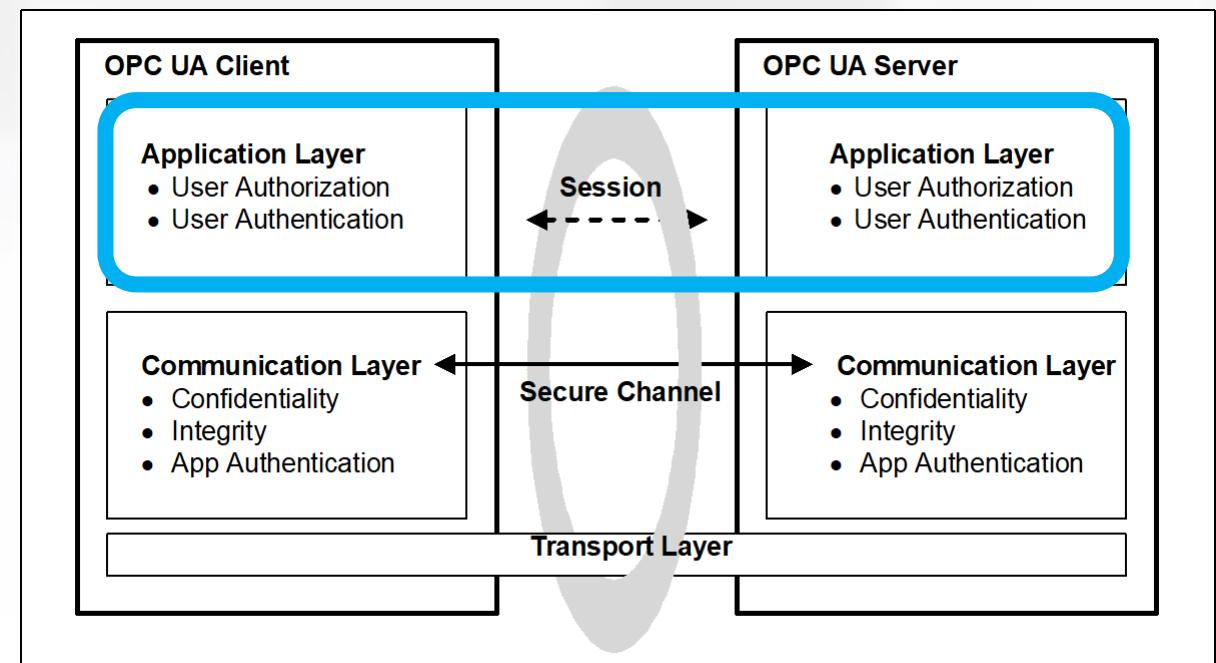
How does encryption apply to OPC UA?

- Over the Secure Channel, a "**Session**" at "Application Layer" is defined.

Here is where the OPC-UA Client application and OPC-UA Server application transmit information, settings, and commands.

at Application Layer
there is the User Authentication
and User Authorization

- anonymous,
- username/password
- X.509 Certificate
(User Instance Certificate)



Secure channel security

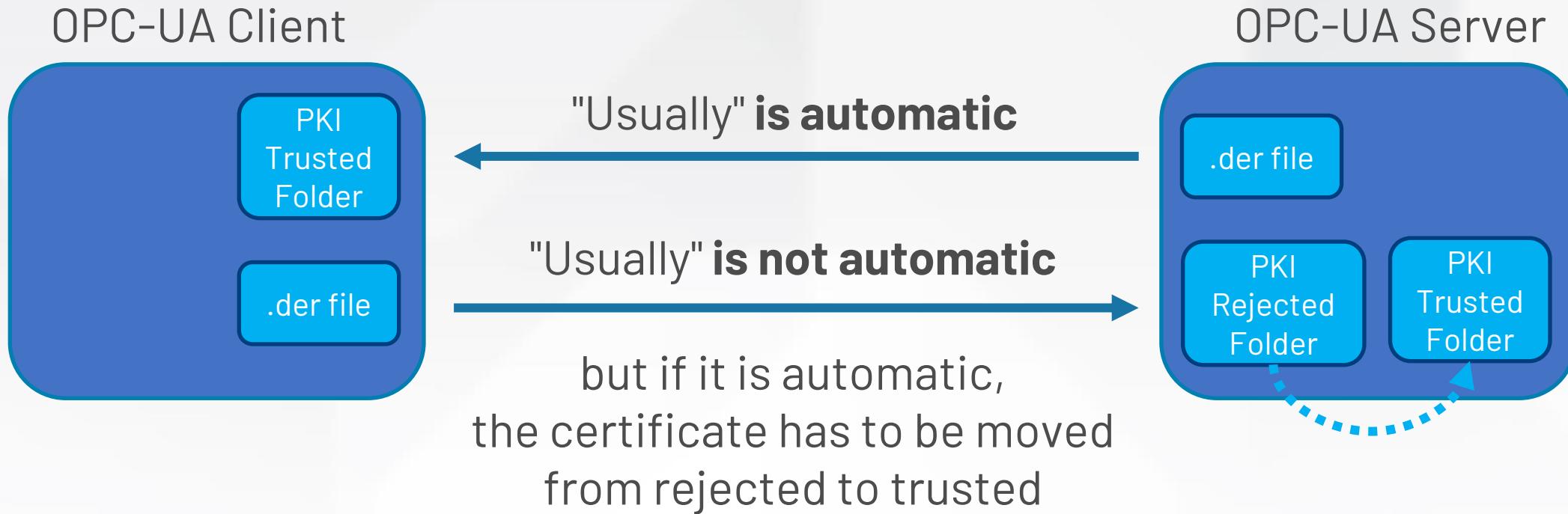
- Secure Channel security = Security Profile + Security Policy

Security Profiles	Description	Security Policies (length of the key and the algo)
None	No security	None
Sign	The sender uses <u>his private key</u> to digitally sign the data. The receiver verifies whether the data comes from the expected sender using the <u>sender's public key</u>	Basic128Rsa15 (deprecated) Basic256 (deprecated) Basic256Sha256
Sign+Encrypt	In addition to Sign, the sender encrypt data using <u>receiver's public key</u>	Aes128-Sha256-RsaOaep Aes256-Sha256-RsaPss

PKI and X.509 Certificates

- Public Key Infrastructure (PKI)
 - is a system for the creation, storage, and distribution of digital certificates
- X.509
 - is a standard defining the format of public key certificates
- In OPC-UA:
 - ".der" is the certificate file containing **public key**,
 - ".pem" is the **private key** file

OPC-UA certificate exchange





Factory**Talk**

by ROCKWELL AUTOMATION



THANK YOU!