

MANUAL

# PB610 Panel Builder 600

Programming software for  
CP600 control panels



# Contents

<b>1 Getting started</b> .....	<b>1</b>	Changing multiple widgets properties .....	66
Assumptions .....	2	Changing fill color property according to tag values .....	67
What's new .....	2	<b>6 Pages</b> .....	<b>69</b>
Information security precautions .....	3	Unified pages .....	70
Precautions before starting .....	4	Project Validator .....	70
Installing the application .....	5	Differentiated pages .....	73
<b>2 Runtime</b> .....	<b>9</b>	<b>7 Project properties</b> .....	<b>79</b>
HMI device basic settings .....	10	Runtime .....	80
Context menu options .....	10	Plug-in .....	86
Built-in SNTIP service .....	13	Project .....	87
<b>3 Runtime on PC</b> .....	<b>15</b>	Web .....	91
Typical installation problems .....	17	Events .....	93
<b>4 My first project</b> .....	<b>21</b>	Regional Settings .....	93
The workspace .....	22	<b>8 The HMI simulator</b> .....	<b>95</b>
Creating a project .....	23	Simulator settings .....	96
Designing a page .....	26	Tags Simulation .....	96
The Widget Gallery .....	27	Data simulation methods .....	99
Label widget .....	29	Launching and stopping the simulator .....	99
Data field widget .....	36	<b>9 Transferring the project to HMI device</b> .....	<b>101</b>
Message widget .....	39	Download to HMI device .....	102
Attaching widget to tags .....	42	Update package .....	104
Dialog pages .....	44	The Runtime loader .....	107
<b>5 Programming concepts</b> .....	<b>47</b>	Upload projects .....	108
Data types .....	48	Runtime dynamic files handling .....	109
"Attach to" parameters .....	49	Userdata .....	110
Formula .....	55	<b>10 Tag editor</b> .....	<b>113</b>
Events .....	59	Communication protocols .....	114
Widgets positioning .....	62	Adding tags .....	115
Managing overlapping widgets .....	63	Exporting tags .....	118
Grouping widgets .....	64		

Importing tags .....	119	Tag Import .....	160
Tag find and rename .....	123	Default variables .....	162
Tag find and replace .....	125	Retentive Memory variables .....	177
<b>11 Indexed addressing .....</b>	<b>129</b>	Services variables .....	182
Creating an indexed addressing set .....	130	NFC variables .....	184
Using indexed tag set in pages .....	135	PLCM09 variables .....	186
<b>12 Tag cross reference .....</b>	<b>137</b>	JSmart variables .....	190
Updating data in the Tag Cross Reference pane .....	141	<b>15 Actions .....</b>	<b>193</b>
<b>13 System Variables (Attach To) .....</b>	<b>143</b>	Alarm actions .....	194
Alarms variables .....	145	Database actions .....	195
Buzzer variables .....	145	Event actions .....	199
Communication variables .....	146	MultiLanguage actions .....	199
Daylight Saving Time variables .....	146	Keyboard actions .....	199
Device variables .....	147	Media Player actions .....	201
Dump information variables .....	149	FTP actions .....	202
FTP client variables .....	150	Page actions .....	204
Keypad variables .....	150	Print actions .....	211
Language variables .....	150	Recipe actions .....	213
Network variables .....	151	Remote Client actions .....	217
Printing variables .....	151	ScreenSaver actions .....	218
Remote Client variables .....	152	System actions .....	218
Version variables .....	153	Tag actions .....	230
Screen variables .....	153	Trend actions .....	232
SD card variables .....	153	Text Editor actions .....	236
Server variables .....	154	User management actions .....	236
Time variables .....	154	Widget actions .....	239
Touch screen variables .....	155	<b>16 The HMI Client .....</b>	<b>245</b>
USB drive variables .....	156	Client application on PC .....	246
User management variables .....	156	Client application on HMI .....	247
<b>14 System Variables (Protocol) .....</b>	<b>159</b>	Settings and time zone options .....	248
Client System Variables .....	160	<b>17 Using the integrated FTP server .....</b>	<b>253</b>
Protocol Editor Settings .....	160	FTP settings .....	253
		<b>18 Using VNC for remote access .....</b>	<b>255</b>

Starting VNC server on WinCE devices .....	256	Data transfer editor .....	322
Starting VNC server on Linux devices .....	257	Exporting data to .csv files .....	324
Starting VNC viewer .....	257	Data transfer limitations and suggestions .....	324
<b>19 Alarms .....</b>	<b>259</b>	<b>24 Offline node management .....</b>	<b>327</b>
Alarms Editor .....	260	Offline node management process .....	328
Remote alarms acknowledge .....	263	Manual offline node management process .....	328
Alarm state machine .....	264	Manual offline configuration .....	328
Setting events .....	264	Automatic offline node detection .....	329
Active Alarms widget .....	267	<b>25 Multi-language .....</b>	<b>331</b>
Alarms History widget .....	272	The Multi-language editor .....	332
Additional Alarms widgets .....	272	Changing language .....	334
Managing alarms at runtime .....	273	Multi-language widgets .....	334
Enable/disable alarms at runtime .....	274	Exporting/importing multi-language strings .....	336
Displaying live alarm data .....	274	Changing language at runtime .....	338
Exporting alarm buffers to .csv files .....	276	Limitations in Unicode support .....	338
Exporting alarm configuration .....	276	<b>26 Scheduler .....</b>	<b>341</b>
<b>20 Events Buffer .....</b>	<b>281</b>	Creating a schedule .....	342
<b>21 Recipes .....</b>	<b>283</b>	HighResolution schedule .....	343
Managing recipes .....	283	Recurring schedule .....	343
Configuring a recipe widget .....	286	Configuring location for schedules .....	345
Recipe status .....	287	Configuring the Scheduler widget .....	345
Uploading/downloading a recipe .....	288	Scheduling events at runtime .....	346
Backup and restore recipes data .....	289	<b>27 21 CFR Part 11 Compliance .....</b>	<b>349</b>
<b>22 Trends .....</b>	<b>291</b>	x.509 Certificate .....	354
Data logging .....	292	Signed CSV files .....	358
Exporting trend buffer data .....	297	Signed PDF files .....	359
Realtime trend widget .....	299	Compliant applications .....	362
History trend widget .....	300	<b>28 User management and passwords .....</b>	<b>363</b>
Scatter diagram widget .....	303	Enable/disable security management .....	364
Scatter chart widget .....	304	Configuring groups and authorizations .....	364
Trend widget tips .....	309	Modifying access permissions .....	365
Table trend widget .....	315	Assigning widget permissions from page view .....	371
<b>23 Data transfer .....</b>	<b>321</b>		





Configuring users .....	371	Custom tables .....	417
Default user .....	373	DB table data source .....	417
Managing users at runtime .....	373	Database Configuration .....	419
Force remote login .....	374	SQL4Automation (legacy) .....	427
<b>29 Audit trails .....</b>	<b>375</b>	Installing SQL4Automation .....	427
Enable/disable audit trail .....	376	Configuring SQL4Automation .....	428
Electronic Signature .....	377	Connection Limits .....	430
Table audit widget .....	379	<b>36 OPC UA Server .....</b>	<b>431</b>
Exporting audit trail as .csv files .....	381	Features .....	431
<b>30 Reports .....</b>	<b>383</b>	Network .....	431
Adding a report .....	384	Authentication .....	431
Configuring text reports .....	384	Using x.509 Certificates .....	432
Configuring graphic reports .....	385	Using self-signed certificates .....	436
Print triggering events .....	386	Using external certificates .....	438
Default printer .....	387	Alarm map .....	440
<b>31 Screen saver .....</b>	<b>391</b>	<b>37 MQTT Interface .....</b>	<b>441</b>
<b>32 Backup/restore of Runtime and project .....</b>	<b>393</b>	Tags configuration .....	441
<b>33 Keypads .....</b>	<b>395</b>	Tags interface .....	443
Creating and using custom keypads .....	396	Settings .....	444
Deleting or renaming custom keypads .....	399	Topic and Payload .....	444
Keypad type .....	400	MQTT Broker Settings .....	447
Keypad position .....	400	<b>38 Special widgets .....</b>	<b>453</b>
<b>34 External keyboards .....</b>	<b>403</b>	BACnet widget .....	454
Search and filter .....	405	Canvas Widget .....	454
Displayed keys .....	405	Combo Box widget .....	457
Removing action associations .....	405	Consumption Meter widget .....	461
Keyboard layout .....	406	Control list widget .....	463
Enable/disable keyboard .....	406	TabBar widget .....	466
Associating actions to keys .....	406	ToolBar widget .....	468
<b>35 Storing data to external databases .....</b>	<b>409</b>	Stack widget .....	468
Configuring the HMI project .....	411	DateTime widget .....	471
Transfer data with JavaScript .....	412	Gesture area widget .....	473
Database tables .....	416	JavaScript function block widget .....	477

Multistate Image widget .....	479	Page object .....	554
Multistate Image Multilayer widget .....	479	Page object properties .....	554
Network Adapters widget .....	481	Page object methods .....	555
RSS Feed widget .....	481	Project object .....	557
Scrolling RSS Feed widget .....	482	Project object properties .....	557
Media Player widgets .....	483	Project object methods .....	558
Browser widget .....	486	Project object widgets .....	567
IP Camera widgets .....	487	Print reports object .....	568
Table widget .....	490	Group object .....	571
Grid Layout widget .....	507	Group object methods .....	571
TextEditor widget .....	514	State object .....	572
Variables widget .....	516	State object methods .....	572
QR Code widget .....	518	Keywords .....	573
<b>39 Custom widgets .....</b>	<b>521</b>	Global functions .....	574
Creating a custom widget .....	522	Handling read/write files .....	575
Adding properties to a custom widget .....	524	Sign in from JavaScript .....	578
Using structured tags .....	527	Limitations in working with widgets in JavaScript .....	580
JavaScript in custom widgets .....	529	Debugging of JavaScript .....	580
User's Gallery .....	532	<b>42 Handling Gestures .....</b>	<b>585</b>
<b>40 Sending an email message .....</b>	<b>535</b>	<b>43 Web access .....</b>	<b>587</b>
Configuring the email server .....	536	Supported platforms and browsers .....	588
Configure emails .....	536	Web pages .....	588
<b>41 JavaScript .....</b>	<b>539</b>	Testing the Web project .....	590
JavaScript editor .....	541	Downloading the Web project .....	591
Execution of JavaScript functions .....	541	Web connectivity issues .....	591
Events .....	543	Web supported features .....	592
Widget events .....	544	Troubleshooting and FAQ .....	595
Page events .....	546	Privacy .....	596
System events .....	547	<b>44 Protecting access to HMI devices .....</b>	<b>597</b>
Objects .....	549	Changing password on HMI device .....	598
Widget class objects .....	549	Ports and firewalls .....	598
Widget properties .....	550	Project Files Encryption .....	599
Widget methods .....	553		

Project Signature .....	602	<b>49 Communication protocols .....</b>	<b>659</b>
<b>45 Tips and tricks to improve performance</b>	<b>605</b>	ABB CODESYS Ethernet .....	661
Static Optimization .....	606	ABB CODESYS Serial .....	671
FAQ on Static Optimization .....	609	ABB IRC5 .....	677
Page caching .....	609	ABB IRC5 Linux .....	682
Image DB .....	609	ABB Modbus RTU .....	692
Precaching .....	610	ABB Modbus TCP .....	699
FAQ on precaching .....	610	ABB Pluto .....	705
<b>46 Functional specifications and compatibility .....</b>	<b>611</b>	ABB Mint Controller HCP .....	711
Table of functions and limits .....	612	BACnet .....	718
HMI devices capabilities .....	613	Client System Variables .....	760
Compatibility .....	614	CODESYS V2 ETH .....	762
Converting projects between different HMI devices .....	615	CODESYS V3 ETH .....	775
<b>47 System Settings (Linux Devices) .....</b>	<b>617</b>	CODESYS V3 Handler .....	786
Runtime Installation .....	617	Environment Variables .....	797
System Settings .....	622	Ethernet/IP CIP .....	798
Update System Components .....	638	Modbus RTU .....	823
Touchscreen calibration .....	640	Modbus RTU Server .....	839
Password protection .....	641	Modbus TCP .....	854
Backup and Restore .....	642	Modbus TCP Server .....	872
Recovery Mode .....	643	OPC UA Client .....	883
<b>47 System Settings (WinCE Devices) .....</b>	<b>645</b>	Simatic S7 ETH .....	897
Runtime Installation .....	645	System Variables .....	936
System Settings .....	647	Variables .....	938
Update System Components .....	650		
Touchscreen calibration .....	652		
Password protection .....	653		
Factory restore .....	653		
<b>48 Updating system components in HMI devices .....</b>	<b>655</b>		
Update of system components from the application .....	656		

# 1 Getting started

---

PB610 Panel Builder 600 is a software application designed to create graphical HMI pages. PB610 Panel Builder 600 has a drag-and-drop interface that makes it easy to create complex pages. Many of the features found in common Windows applications are also available in PB610 Panel Builder 600.

This document is divided into chapters that describe the key functions of PB610 Panel Builder 600 and explain how to use them. Each chapter is presented in a standalone manner, allowing you to jump from chapter to chapter, depending on the task at hand.

---

<b>Assumptions</b> .....	<b>2</b>
<b>What's new</b> .....	<b>2</b>
<b>Information security precautions</b> .....	<b>3</b>
<b>Precautions before starting</b> .....	<b>4</b>
<b>Installing the application</b> .....	<b>5</b>

# Assumptions

We assume that readers have a basic understanding of computers, Microsoft Windows, and the specific network environment where the application will run.

## What's new

### What's new in v4.5

- New gallery with refreshed style for widgets. New user interface with icons, preview, and search feature. (Ref.: ["The Widget Gallery" on page 27](#))
- HMI device can be configured to accept only signed projects. (Ref.: ["Project Signature" on page 602](#))
- Projects can be encrypted to secure intellectual property and not be readable or editable by unauthorized users. (Ref.: ["Project Files Encryption" on page 599](#))
- The HMI device can be configured to "Allow only Secure HTTPS connections". (Ref.: ["Web Server" on page 635](#))
- When you create a new page, you have the option to associate a dashboard to the page. (Ref.: ["Designing a page" on page 26](#))
- Enhancement of the HMI Simulator with data watch/modify window. (Ref.: ["Tags Simulation" on page 96](#))
- Support connection to external databases using macros or SQL commands. (Ref.: ["Storing data to external databases" on page 409](#))
- HTML5 compatible browser widget, based WebEngine. (Ref.: ["Browser widget" on page 486](#))
- QR Code widget. (Ref.: ["QR Code widget" on page 518](#))
- Tab and toolbar widgets to create easily inline buttons with embedded gestures. (Ref.: ["TabBar widget" on page 466](#))
- Stack widget to display group of widgets in Z axes layers both from design time and from runtime. (Ref.: ["Stack widget" on page 468](#))
- Scatter chart widget. (Ref.: ["Scatter chart widget" on page 304](#))
- Client system variables protocol. Defines a set of system variables that take the values from the active client and not of the server where runtime is running. (Ref.: ["Client System Variables" on page 160](#))
- Environment variables protocol to get access to environment variables of the hosting Operating System. (See communication protocols manual)
- Added support for major communication protocols to keep tags synchronized with the dictionary. (Ref.: ["Importing tags" on page 119](#))
- Trends autofill, import/export, and copy/paste functionalities to maximize productivity. (Ref.: ["Data logging" on page 292](#))
- Quick change project type from tight click on Project properties. (Ref.: ["Changing the device model" on page 23](#))
- Quick link to open an uploaded project. (Ref.: ["Upload projects" on page 108](#))
- Support for dynamic keyboards depending on language. (Ref.: ["Language Keyboards" on page 333](#))
- TLS support in sending emails. (Ref.: ["Configuring the email server" on page 536](#))
- The gesture area widget is supported on web clients (Ref.: ["Gesture area widget" on page 473](#))
- Indexed Tag Sets import/export and copy/paste functionalities to maximize productivity. (Ref.: ["Toolbar" on page 134](#))
- Support for variable widget variables in Indexed Tag sets for independent sessions on the Set. (Ref.: ["How to create an indexed tag set" on page 130](#))

- Added delay between bad passwords to prevent brute force attacks. (Ref.: ["Miscellaneous settings" on page 369](#))
- Allow MQTT to put many tag values into payload. (Ref.: ["Multiple Tags" on page 446](#))
- Autocomplete functionality has been added to the message widget with the ability to quickly choose a previously used message. (Ref. ["Auto-completing dialog" on page 42](#))
- Porting of Scheduler Widget to new table format.
- Unicode support for DB connectors.

## Breaking Changes

- PB610 Panel Builder 600 4.5 requires Linux BSP versions equal to or greater than 1.3.
- HMI devices based on Windows CE operating system are no longer supported. This means, for example, that the new functions have only been developed for the HMI devices based on the Linux platform. (See: ["HMI devices capabilities" on page 613](#))
- The old widget gallery has been replaced with the new gallery. See the chapter ["Switch to old widget gallery" on page 29](#) if you need to return to the old gallery
- By default, when a new project is created the FTP service will accept only encrypted connections. See ["FTP authorizations" on page 367](#) if you need to use an old FTP client that does not support the encrypted FTP mode.

# Information security precautions

When use this product you might receive damage as listed below.

1. Information leakage or outflow through this product
2. Fraudulent operation of this product by a malicious third party
3. Obstructing or stopping this product by a malicious third party

Sufficient measures, including the following measures, should be taken at your own risk to prevent such damages.

## Data storage

- Do not storage of personal information on this product.

## Password

- Please be sure to change the password since it is set to the default value at the time of purchase.
- Do not use the default password.
- Please be responsible for managing your password so that it is not known to any third party and do not forget it.
- If you forget your internal password, there is no way for you to reset it. You must return the product to us and we will return it to you in its factory condition.
- Please make sure that your password is at least 8 characters long and contains a combination of upper and lower case letters, numbers and symbols so that a third party cannot guess your password.
- Do not use the same password as your user name. Do not use the same password as the one you are using elsewhere.
- Please change your password on a regular basis.

## Networks

- Use this product on a network where safety is secured by using a firewall.
- When using this product on a system where a PC is connected, make sure that checking and cleaning of infection by computer virus or malicious program is performed periodically.
- It has the ability to use unencrypted communications. (FTP, HTTP, SMTP, PLC communications, etc.).

- Please make sure that a third party cannot easily connect to the network used by this product.
- Use HTTPS when accessing this product through a web browser.
- Also, be sure to close all browsers after accessing it.
- Do disable the functions of the services you don't use. (SNMP, NTP, VNC, DHCP, etc.)
- Be sure to log out when you have completed the necessary setup operations.
- Do not install this product or cables in a place where they can be easily destroyed or altered.
- SNMP is assumed to be used for testing purposes. It should be disabled during operation.
- Furthermore, it is recommended that the product be used in an environment that has VPN (Virtual Private Network) or leased line network.

### **Transfer/ Disposal/ Repair**

- If the product is to be disposed of, transferred, repaired, or otherwise transferred to a third party, important information may also be recorded on the product and on the external recording media used. At customer's risk, please handle it with care, such as erasing it.

### **Installation**

- On the back of the unit, there are interfaces that affect its operation, such as power supply, external storage media, and communication connectors. Please make sure to install the unit in a manner that does not allow unauthorized parties to touch it.

### **Exemption**

The Company shall not be responsible for any information security problems or damages that may occur to you in the event that you fail to comply with the above precautions in using this product.

### **OPEN SOURCE SOFTWARE**

This product uses software including open source software. For license information of open source software, refer to [Help] > [Legal Notices...] in the menu of PB610 Panel Builder 600.

### **Acknowledgement:**

- This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit. (<http://www.openssl.org/>)
- This product includes cryptographic software written by Eric Young ([eyay@cryptsoft.com](mailto:eyay@cryptsoft.com))
- This product includes software written by Tim Hudson ([tjh@cryptsoft.com](mailto:tjh@cryptsoft.com))

## **Precautions before starting**

### **Cautions on handling a USB memory**

- Do not remove the USB memory while the power is on. Communication may be stopped. It is necessary to restart the power supply to recover.
- The data saved in the USB memory may be lost in the following cases.

We assume no responsibility whatsoever for the lost of saved data.

1. When a user or third party used the USB memory incorrectly.
2. When the USB memory was affected by any static electricity or electrical noise.

3. When the USB memory was removed or the power supply of the HMI unit was turned off while data was being read, written or deleted to/from the USB memory.

- It is recommended to save important data in another media for backup.

### Cautions on handling a SD memory card

- Do not remove the SD memory card while the power is on. Communication may be stopped. It is necessary to restart the power supply to recover.
- The data saved in the SD memory card may be lost in the following cases.

We assume no responsibility whatsoever for the lost of saved data.

1. When a user or third party used the SD memory card incorrectly.
2. When the SD memory card was affected by any static electricity or electrical noise.
3. When the SD memory card was removed or the power supply of the HMI unit was turned off while data was being read, written or deleted to/from the SD memory card.

- It is recommended to save important data in another media for backup.

## Installing the application

PB610 Panel Builder 600 installation contains:

- PB610 Panel Builder 600: an application for designing custom HMI projects in a user-friendly manner, along with a variety of objects in its built-in library, the Widget Gallery.
- HMI Client: a light-weight application that can be used on Windows computers to remotely view and manage a project running on an HMI device.
- HMI Runtime: a standalone application that runs on the HMI devices. The HMI Runtime is installed via PB610 Panel Builder 600.
- PB610 PC Runtime: a standalone application that runs on Win32/Win64 platforms (computers instead of HMI devices).

### PB610 Panel Builder 600 system requirements

PB610 Panel Builder 600 has the following system requirements:

<b>Operating System</b>	Windows Embedded Standard (WES 2009) Windows Server 2003 Windows Vista Business/Ultimate Windows 7 Professional Windows Embedded Standard 7 Windows 8 Windows 10 Windows 11
<b>Storage</b>	500 MB Minimum
<b>RAM</b>	512 MB
<b>Other</b>	One Ethernet connection



## Installing multiple versions of PB610 Panel Builder 600

You may install different instances of PB610 Panel Builder 600 on the same computer. Each installation has its own settings and can be uninstalled individually.

Three installation scenarios are possible:

Installation scenario	Results
<b>First installation of PB610 Panel Builder 600 in the system</b>	Software is installed in the specified destination folder
<b>System with only one instance of PB610 Panel Builder 600 already installed</b>	Current version can be replaced or maintained.
<b>System with multiple instances of PB610 Panel Builder 600 already installed</b>	Last version installed can be replaced or maintained.

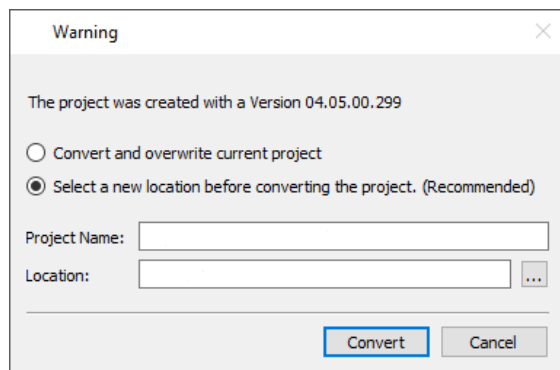
If you try to install a second instance of an already installed version of PB610 Panel Builder 600, a warning message is displayed.

Multiple PB610 Panel Builder 600 installations share a common workspace folder, each sub-folder includes the version number, for example *C:\Program Files\ABB\Panel Builder 600 Suite 4.0*. Each installed version has its ID and can therefore be removed individually.

Each installation is listed separately in the Windows **Start** menu.

## Opening older projects

When opening a PB610 Panel Builder 600 project (.jpr file) created with an older version of the software PB610 Panel Builder 600 asks to convert the project to the current version:



Option	Description
<b>Convert and overwrite current project</b>	The project is converted without a backup copy of the original version
<b>Select a new location before converting the project</b>	The project is copied inside the specified folder and then converted.



**WARNING: Do not edit projects with a version of PB610 Panel Builder 600 older than the version used to create them. This will damage the project and may cause runtime instability.**

## Multilanguage for PB610 Panel Builder 600

PB610 Panel Builder 600 is available in multiple languages. All languages are installed by default as part of PB610 Panel Builder 600.

The default language is English. To change it go to **Help > Change Language**.

## Crash reports

A crash report dialog appears whenever PB610 Panel Builder 600 freezes or crashes.



**Important: Always save crash report files since they may contain useful information for technical support.**



# 2 Runtime

---

HMI Runtime is designed to support different platforms and different operating systems.

---

<b>HMI device basic settings</b> .....	<b>10</b>
<b>Context menu options</b> .....	<b>10</b>
<b>Built-in SNTP service</b> .....	<b>13</b>

# HMI device basic settings

HMI devices are delivered from factory without Runtime. If no Runtime is installed on the device, see ["The Runtime loader" on page 107](#) for details.

## Runtime modes

The HMI Runtime is composed of two logic units:

- **Server:** runs communication protocols, collects data, monitors alarms, drives trend buffer sampling.
- **Client:** displays data collected by server.

The server unit is responsible for handling the HMI services such as the communication protocols, performing data acquisition, driving trend buffer sampling activities, monitoring alarms, and so on.

The client unit is the part which is responsible for the visualization process: use the data collected by the server to render it on the display as graphical information.

The server unit works in two operating modes:

- **Configuration mode:** server is idle (for example when no project is loaded on the device or some system files are missing).
- **Operation mode:** server is operating according to the settings defined by the system files and by the loaded application project.



Note: Data on client may be displayed even if no activity is running on the server.

## Context menu options

Some system operations are possible through the context menu. The context menu can be configured to be shown by pressing and holding an empty area of the screen for a few seconds or through the use of an action.



**For security reasons, it is advisable to configure the Context Menu to be accessible only using a "macro". In this way, it is possible to configure the use of the macro to authorized users only. (See ["Runtime" on page 80](#) for details)**

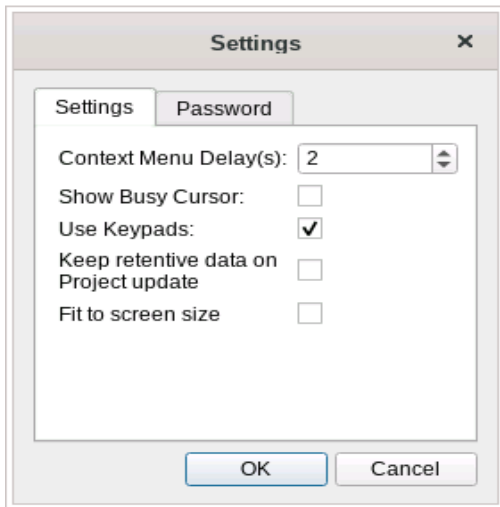
## Zoom In/Out

Select view size at runtime

## Pan Mode

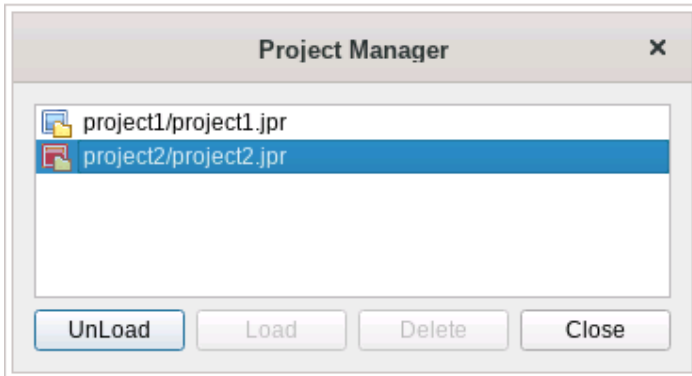
Enables/disables pan mode after a zoom in

## Settings



Main parameters	Description
<b>Context Menu Delay (s)</b>	Context menu activation delay. Range: 1–60 seconds.
<b>Show Busy Cursor</b>	Display an hourglass when the system is busy
<b>Use keypads</b>	Display keypads when user touches a data entry field. Set to <b>disable</b> when an external USB keyboard is connected to the device.
<b>Keep retentive data on project update</b>	Preserve the content of the retentive data at project download or update.
<b>Fit to screen size</b>	Adapts the view to the screen size
<b>Password</b>	Define password protected operations amongst the following: <ul style="list-style-type: none"> <li>• Download Project/Runtime</li> <li>• Upload project</li> <li>• Board management (BSP Update)</li> </ul> See " <a href="#">Protecting access to HMI devices</a> " on page 597 for details.

## Project Manager



This tool allows you to:

- unload the current project
- load another project
- delete a project.

When you load a new project, the current project is automatically unloaded. You must unload a project before you can delete it.

## Update

This function loads update packages from an external USB drive. See ["Update package" on page 104](#) for details.

## Select Certificate

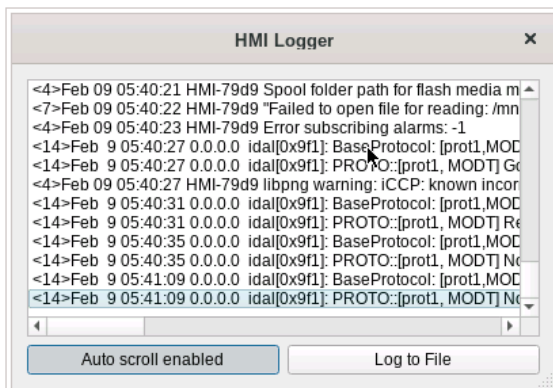
Select a certificate to use to verify signed projects (available only on PB610 PC Runtime)

## Backup

You can create a backup copy of the Runtime and of the project.

## Logging

This function displays a log of system operations.



Click **Log to file** to save data: a logger.txt file is saved to the ...\*var*\log folder.

This file can be retrieved using an FTP Client and forwarded to technical support.



Note: Once enabled, logging is maintained after power cycles and must be manually disabled.

## Show log at boot

This function enables the logger at start up. If the **Log to file** option has been enabled, log files are saved from startup.

## Logout

Logs off the current user.

## Show system settings

Allow the HMI settings and the management of system components. See "[System Settings \(Linux Devices\)](#)" on [page 617](#) for details.

## Developer tools

Utility functions for debugging at runtime. It is visible only if enabled in the Project Properties (see "[Developer tools](#)" on [page 83](#) for details).

## Register

Enter license keys.

## About

This function shows information about the Runtime version and the device IP Address.



**WARNING: Context Menu action has no effect if executed from a dialog page.**

## Exit

Exit from the HMI Runtime.

The command is only available during the development activity when the HMI Runtime has been manually activated. The command is not available during the normal operation of the operator panel.

## Built-in SNTP service

The HMI device features an integrated SNTP that synchronizes the internal real-time clock panel whenever the predefined server is available.

Use HMI device "[System Settings](#)" on [page 622](#) to configure the service.



*On WinCE devices, the SNTP service is available from the BSP v1.76 ARM / 2.79 MIPS or higher*





# 3 Runtime on PC

---

PB610 PC Runtime for Windows is an HMI platform that combines advanced HMI features and vector graphics with powerful web technologies. You can choose this platform to monitor and control your equipment with tags, alarms, schedulers, recipes, trends, JavaScript logic and events.

PB610 PC Runtime provides connectivity with factory and building automation protocols, based on Ethernet and serial interfaces.

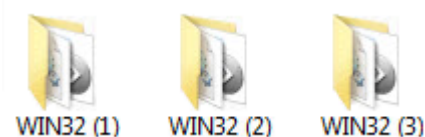
## PB610 PC Runtime system requirements

PB610 PC Runtime as the following minimum system requirements:

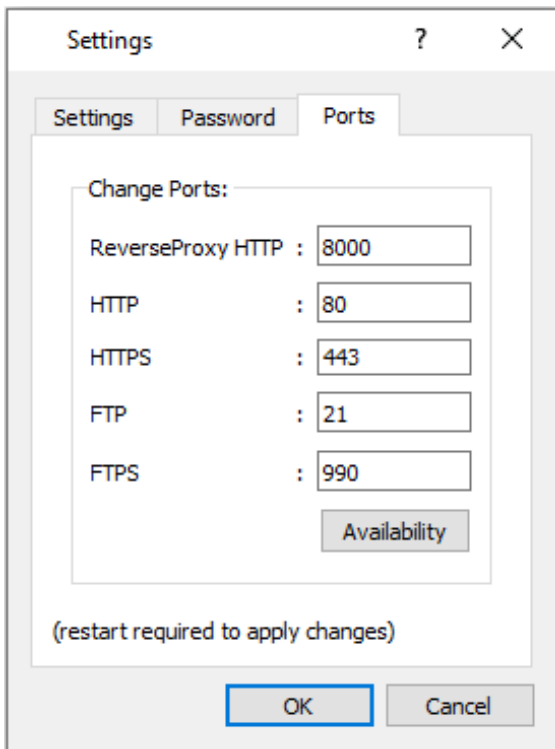
<b>Operating System</b>	Windows Embedded Standard (WES 2009) Windows Server 2003 Windows Vista Business/Ultimate Windows 7 Professional Windows Embedded Standard 7 Windows 8 Windows 10 Windows 11
<b>Storage</b>	256 MB Min
<b>RAM</b>	512 MB
<b>CPU</b>	min. 300 MHz Pentium III or similar processors with 500 MHz.
<b>Graphic</b>	min. SVGA
<b>Other</b>	One Ethernet connection

## Multiple instances of PB610 PC Runtime

PB610 PC Runtime can run in multiple instances. Copy the installation folder to a writable location and double-click on the HMI application in each folder to start it.



The port used by PB610 PC Runtime can be changed from the **Settings** dialog. Restart the application to apply the port change.



## Limitations

The following features are not supported in PB610 PC Runtime:

Function	Feature NOT supported
Manage Target	Board section
System Mode/ User Mode	Tap sequence and rotating menu
VNC/PDF readers	Non-standard computer software
Backup/Restore	Backup and restore functions. Standard computer software can be used for the purpose.
Protocols	Serial protocols requiring special hardware.

See ["Functional specifications and compatibility" on page 611](#) for more details.

## Fullscreen mode

PB610 PC Runtime can start in fullscreen mode or in a window.

To switch to full screen:

1. Right click in the PB610 PC Runtime main window to display the context menu.
2. Choose **Full Screen**.

## The workspace folder

When using PB610 PC Runtime, project files are stored in a workspace folder in:

`%appdata%\ABB\[build number]\server\workspace`

where [build number] is a folder named as build number (for example, 01.90.00.608).

## Typical installation problems



**Important: Make sure that ports 80/HTTP and 21/FTP are not blocked by the firewall.**

If a port is in use and a conflict is detected a dialog is displayed to allow the user to change the default ports.

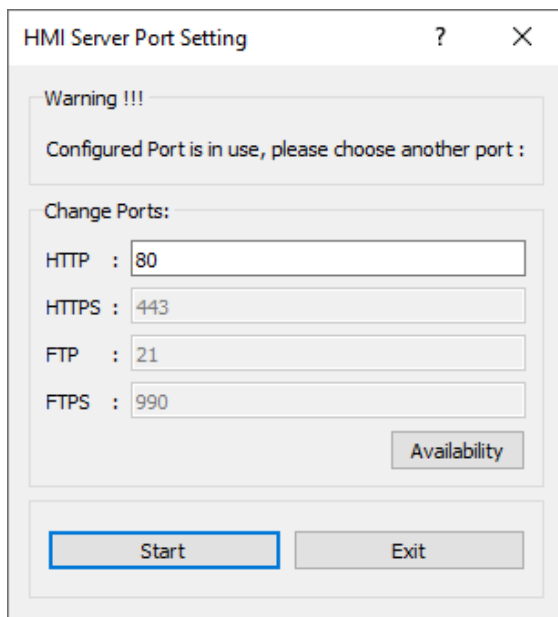
See "[Protecting access to HMI devices](#)" on page 597 for details.

In some conditions PB610 PC Runtime cannot detect all services running in ports like 80/HTTP and 21/FTP, this forces PB610 PC Runtime to be closed automatically. This happens, for example, when IIS or MS SQL Server or other windows services are running on these ports. In these cases, disable window services

If the project download to PB610 PC Runtime fails, try one of the following procedures.

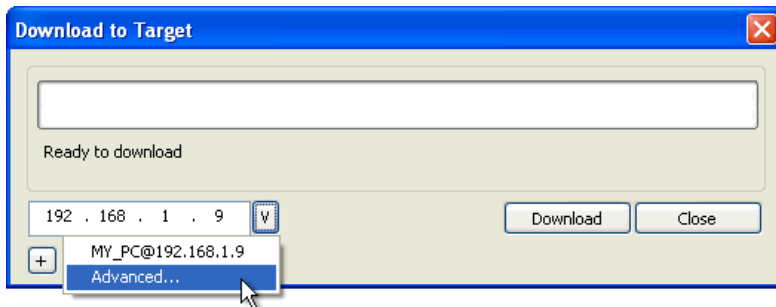
### Issues with port numbers

PB610 PC Runtime uses ports 80 and 21 by default. If at least one is occupied a warning message is displayed:

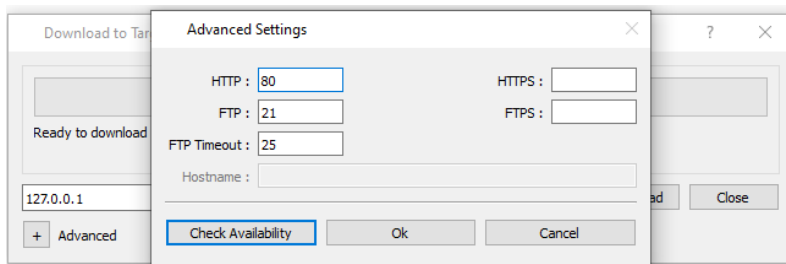


Make sure that when you change this port you also change the port used for download to HMI device in PB610 Panel Builder 600.

1. From the **Download to Target** dialog select **Advanced**.



2. Modify the port number to match that set on PB610 PC Runtime.



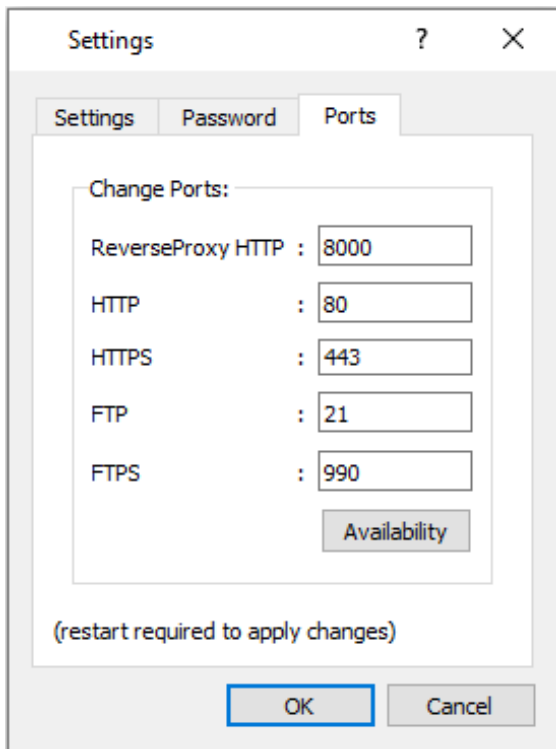
3. Click **OK** to confirm: you can now download you project to the PB610 PC Runtime.

## Restoring port information

If information about changes made on PB610 PC Runtime listening ports has been lost, the following error message is returned:

*Impossible to establish communication with Runtime. Please check connection settings and verify the Runtime is properly running on HMI device.*

The port used by PB610 PC Runtime can be changed from the **Settings** dialog. Restart the application to apply the port change.



## Bypassing firewall or antivirus blocks

If PB610 Panel Builder 600 is running on the same machine as the PB610 PC Runtime, your firewall or antivirus may block the connection from PB610 Panel Builder 600 to PB610 PC Runtime.

1. From the **Download to Target** dialog manually type-in the localhost IP address 127.0.0.1.
2. Click **Download**.



# 4 My first project

---

This section describes how to create a simple PB610 Panel Builder 600 project.

---

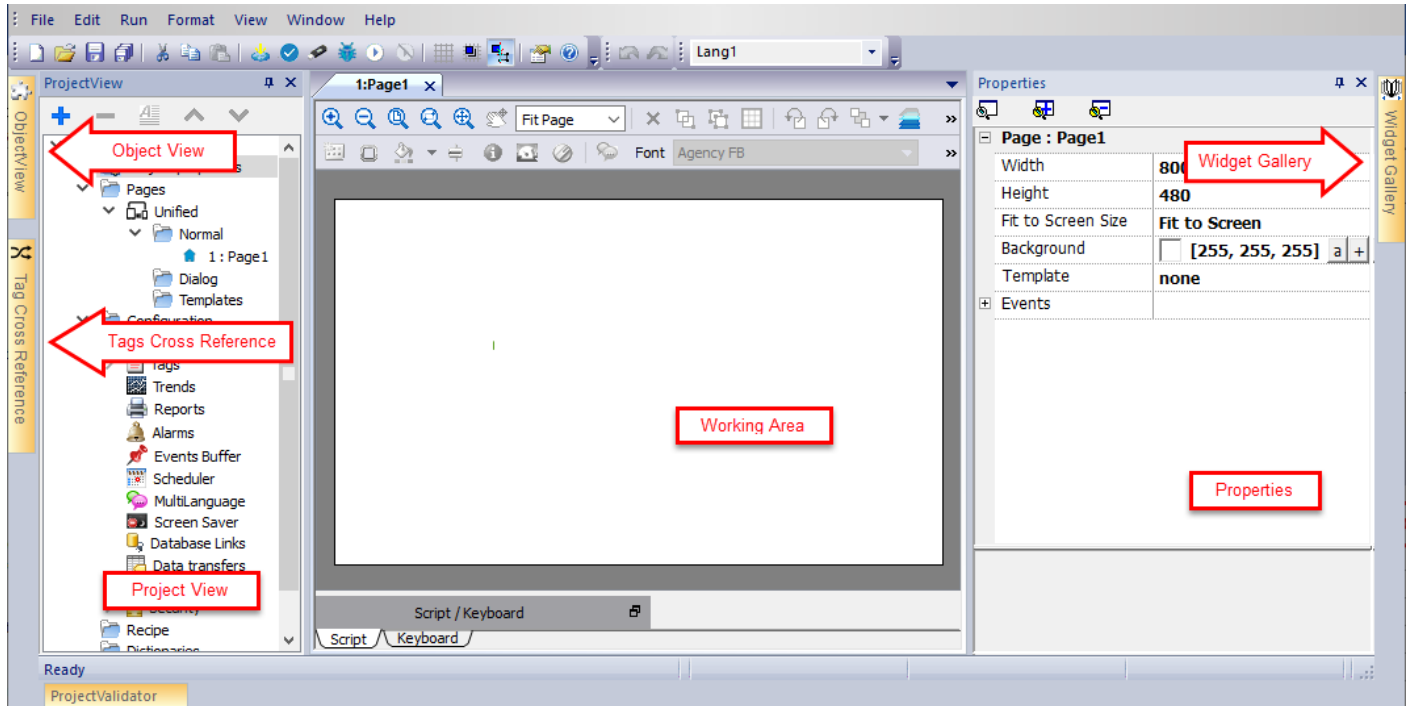
<b>The workspace</b> .....	<b>22</b>
<b>Creating a project</b> .....	<b>23</b>
<b>Designing a page</b> .....	<b>26</b>
<b>The Widget Gallery</b> .....	<b>27</b>
<b>Label widget</b> .....	<b>29</b>
<b>Data field widget</b> .....	<b>36</b>
<b>Message widget</b> .....	<b>39</b>
<b>Attaching widget to tags</b> .....	<b>42</b>
<b>Dialog pages</b> .....	<b>44</b>




# The workspace

## Workspace areas

PB610 Panel Builder 600 workspace is divided into the following main areas:



Area	Description
<b>Project View</b>	Project elements in hierarchical project tree.
<b>Object View</b>	Tree view of widgets organized by page.
<b>Working Area</b>	Space where pages are edited. Tabs at the top of the area show all open pages.
<b>Properties</b>	Properties of selected object.
<b>Widget Gallery</b>	Library of graphic objects and symbols.
<b>Tag cross reference</b>	List of locations where a given tag is referenced.
<b>Project Validator</b>	Area used from the Project Validator to list warning messages related to the project

 Note: The workspace layout can be changed at any time, changes are saved and maintained through working sessions.

## Resetting the workspace layout

To restore the default layout, use the **File > Reset and Restart** function.

## Creating a project

Path: **File** > **New Project**

1. In the **Project Wizard** dialog enter a name for the project and the storage location.
2. Click **Next**: the HMI device selection dialog is displayed.
3. Choose one device from the list of the available models.
4. Choose device orientation.
5. Choose the project template to create.
5. Click **Finish** to complete the Wizard.

### Portrait rotation exceptions

The following elements are not rotated in portrait mode.

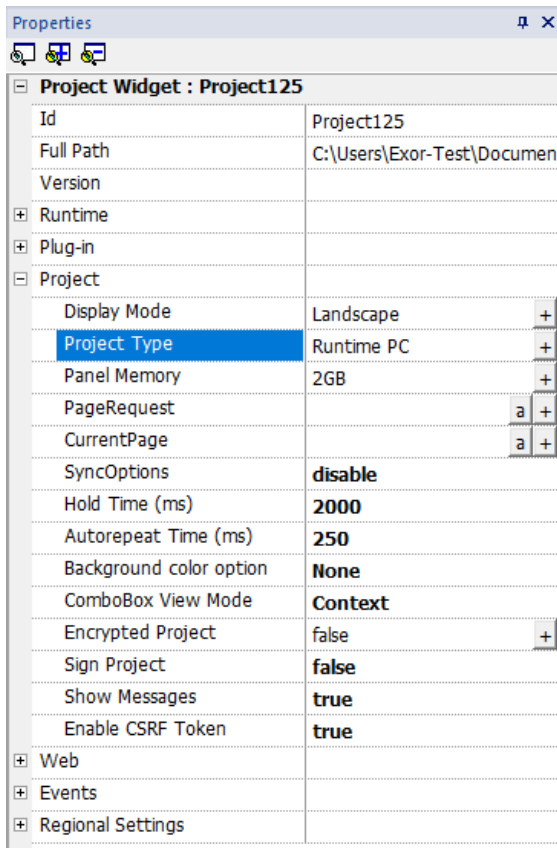
Element	Description
<b>Operating system dialogs</b>	System settings and system dialog
<b>ContextMenu and related dialogs</b>	Project Manager, About, Settings, Logging, Backup
<b>Video</b>	IPCamera, MediaPlayer
<b>JavaScript</b>	Alert and Print function
<b>Dialog pages</b>	"Title" of dialog pages
<b>Scheduler</b>	Dialogs for data entry
<b>Macro</b>	ShowMessage, LunchApplication, LunchBrowser
<b>External applications</b>	PDF Reader, VNC



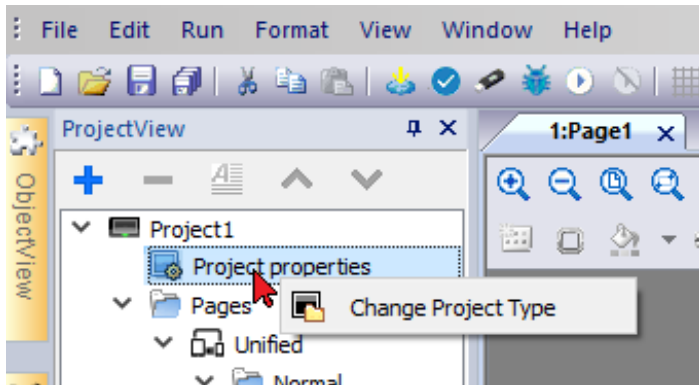
HMI devices based on Linux platform can be rotated from the BSP (see "*Displays*" tab from the "[System Settings](#)" on page 622)

### Changing the device model

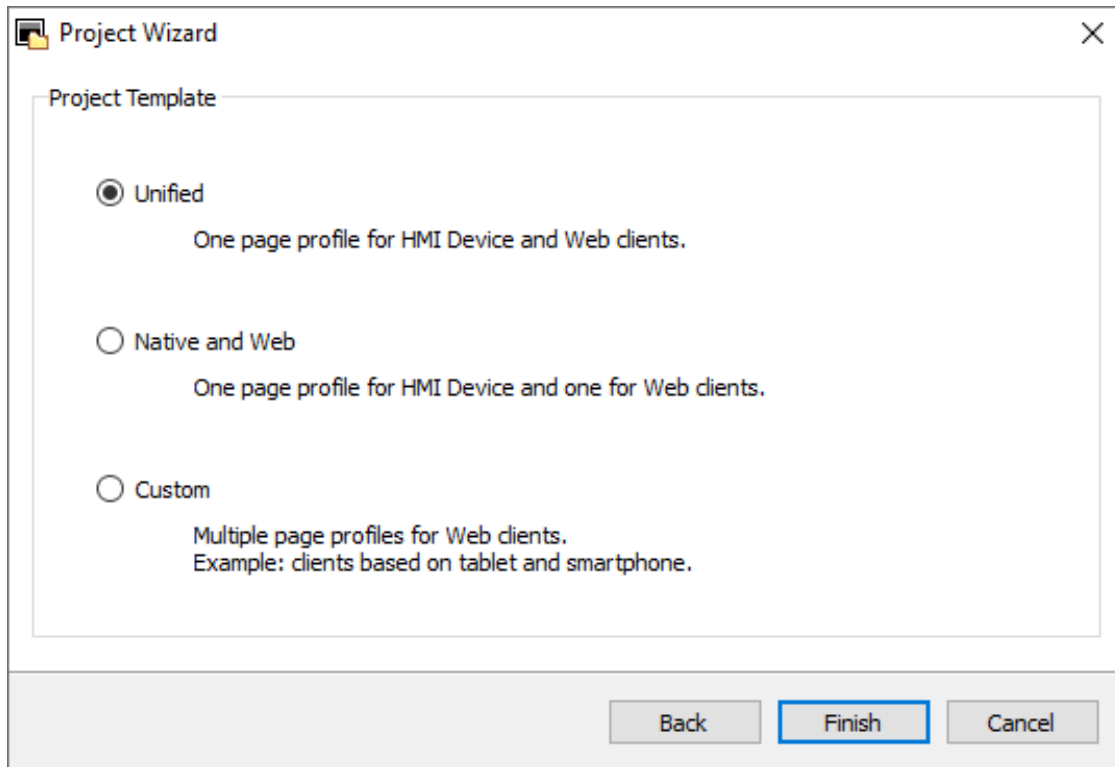
Once you have developed your project you can still change the device model, from the Project Properties pane (Ref.: "[Project](#)" on page 87). This will not resize the widgets, but will relocate them on the screen. A warning will be displayed if some objects cannot be relocated.



A shortcut to quickly access the device model dialog is available on the Project View panel



## Project Template



The "project template" proposes and then creates, the most common folder structures for the project's pages. Later, you can always modify the structure at any time.

The propose structure are:

Element	Description
<b>Unified</b>	Create a unique folder that will contain all the pages of your project. The same pages could be used on HMI Device, on remote clients and on Web Clients. Use this choice if you want to have the same pages on all platforms.
<b>Native and Web</b>	Create two folders, one to contain the pages of your project that will be used on the HMI device and another one for the pages that will be used on Web clients. Use this choice if you want to have different pages on Web clients.
<b>Custom</b>	Give the possibility to create different folders to contain the pages to use on HMI device, Web client, Table client, and Smartphone client.

## Copying, moving, renaming a project

PB610 Panel Builder 600 projects folder contain all the files of the project: to move, copy or backup a project, move or copy the project folder to the desired location.

To rename a project use the **File > Save Project As** function: this operation might take a few minutes.



**WARNING: Do not rename the project folders manually.**

# Designing a page

Path: **ProjectView** > **Pages**

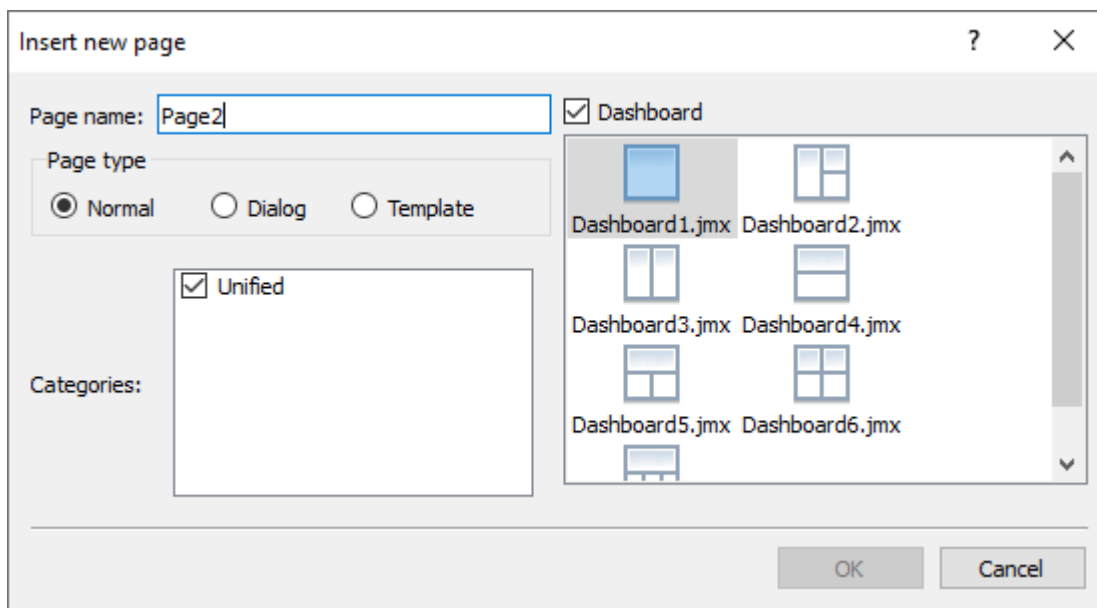
When a project is created, the first page is automatically added and shown in the **Page Editor**.

## Adding objects to a page

Drag and drop objects from **Widget Gallery** to the page.

## Adding a page

1. Right click the **Pages** node from the project tree and select **Insert new page**.
2. Type a name for the new page.



### Page Type

Define the page type that can be:

Normal	Normal page
Dialog	Page with a smaller size that can be popup over a normal page
Template	A page that can be used as a background of a normal page

### Category

If you are developing different pages for different devices types, select the "Category" folder that identifies the devices that will display the page. See "Differentiated pages " on page 1 for additional details.

### Dashboard

When you create a new page, you have the option to associate a dashboard with the page. The dashboard adds the ability to configure spatial relationships between page widgets. If you choose to have a dashboard it is recommended to

define the desired grid before placing the widgets in the grid. You can refer to chapter "[Grid Layout widget](#)" on page 507 for details of how to configure the grid.

## Importing a page

When importing a page PB610 Panel Builder 600 will import the page layout and the page widgets without importing the actions and data links attached to widgets. You can choose between two different behavior:

- importing only the pages and the widgets: in this case all actions and data link have to be defined
- importing pages with references to actions and data links: used tags must be present in the project for these elements to work properly



Note: Page import can only be performed between projects made using the same software version. Save the older project as the newer version, then try again.

1. Right click the **Pages** node from the project tree and select **Import page**.
2. Choose the page to be imported from the desired project then click **OK**: a warning message is displayed.
3. Click **Yes** to remove all the links to data and actions. Click **No** to maintain the reference to data links and actions. Tags need to be available in the new project.

## Group of pages

You can group similar pages for easier maintenance. Grouping pages does not affect how pages appears at runtime. To create a group of pages:

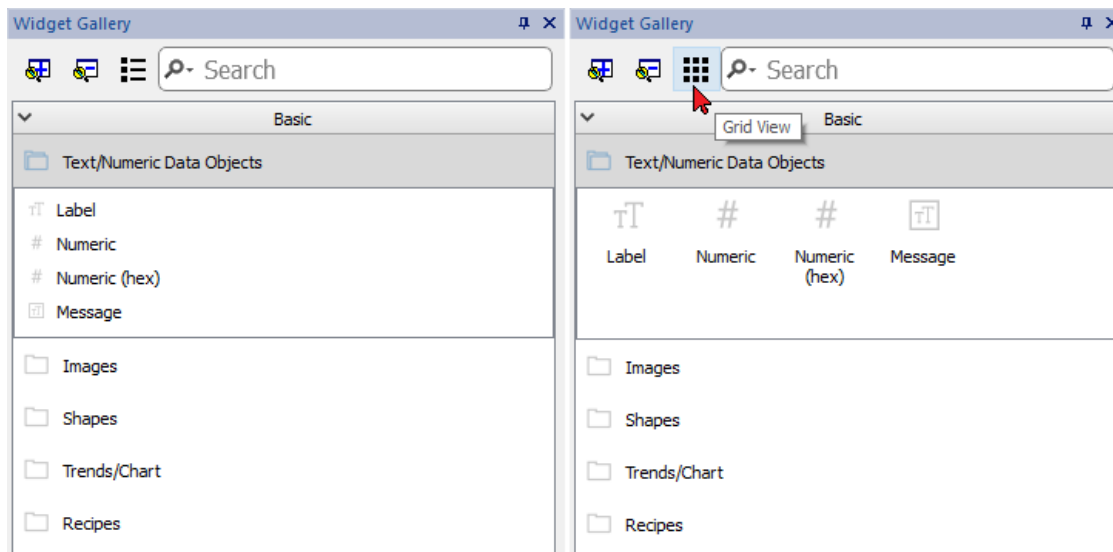
1. In **ProjectView** right click **Pages** node and select **Create Group**: a new folder is added
2. To move a page to a group, right click a page and select **Groups > groupName**.

# The Widget Gallery

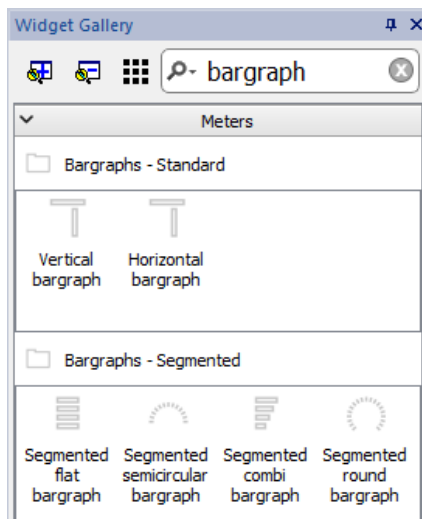
*Path: View > Toolbars and Docking Windows > Widget Gallery*

HMI objects required to build an application are available in the **Widget Gallery**. The gallery is divided into several categories, each containing a collection of widgets.

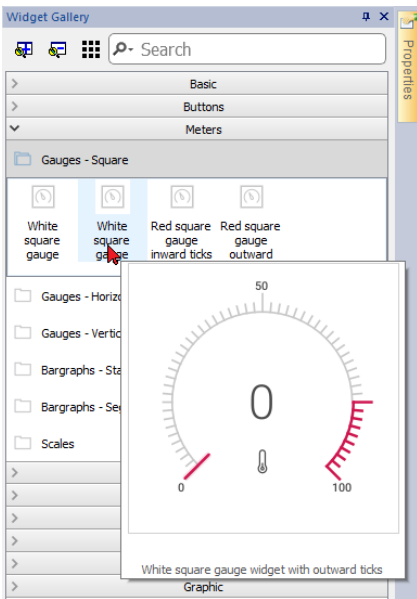
List view and Grid view are available:



You can type text in the search bar to easily find the widget that you are searching:



Note that when the mouse has placed over a widget the preview of the widget will be displayed



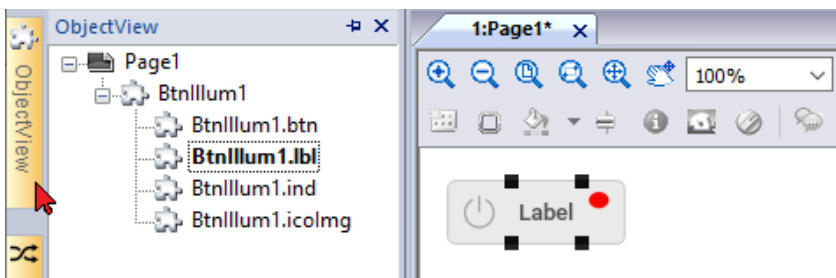
## Adding a widget to a page

1. Select the widget from the **Widget Gallery**.
2. Drag and drop it on the page.

## Complex widgets

Some widgets are composed of many sub widgets. For example, a rectangle button with a LED is a widget composed of a button, a label, an icon, and a LED. The structure of widgets can be seen in the **ObjectView** when the widget is selected.

You can select a sub-widget, such as the label in a button, from the **ObjectView** and modify it without ungrouping the whole widget.



## Switch to old widget gallery

The current version of PB610 Panel Builder 600 contains a completely revised new widget gallery. If you need to use the old gallery, you can switch back to the old library using the "Use Old Gallery" command available from the View menu on the main toolbar.

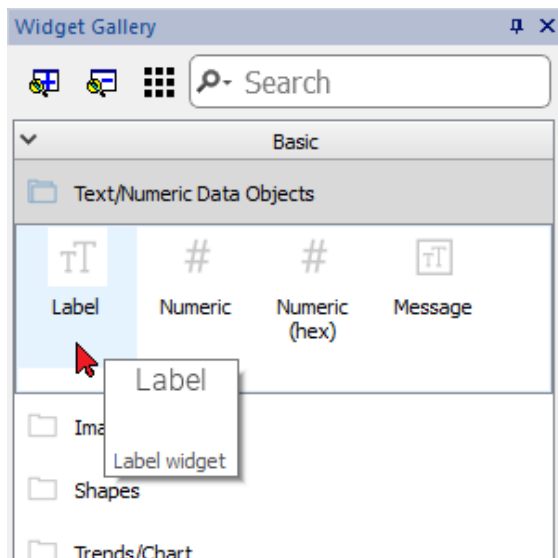
**Path:** *View > Use Old gallery*

## Label widget

The label widget gives the possibility to display text and tags values.



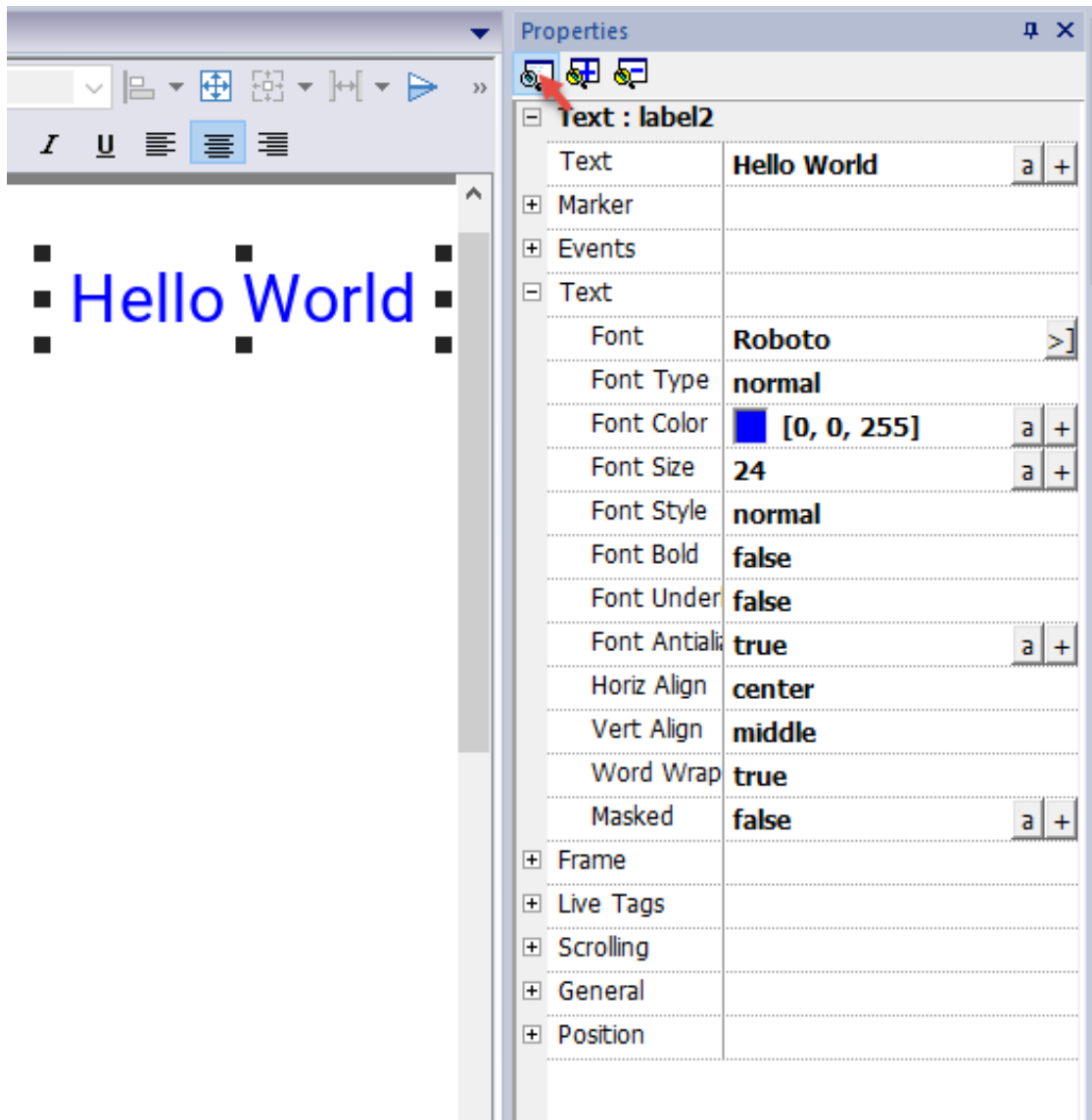
**Path: View > Toolbars and Docking Windows > Widget Gallery**



## Label properties

Drag and drop the widget inside the page and select the widget to open the properties dialog of the widget.

Note that some properties are visible only when the "Show Advanced Properties" button is selected.





Property	Description
<b>Text</b>	The string to display. String can be static or retrieved from a TAG. See " <a href="#">Attaching widget to tags</a> " on page 42
<b>Marker</b>	Enable a Marker around the widget (It is visible only inside PB610 Panel Builder 600)
<b>Events</b>	Action that will be executed if widget contents change. See " <a href="#">Events</a> " on page 59
<b>Text</b>	Text properties
<b>Frame</b>	Parameters to enable and configure a frame of the widget and/or a color for the background

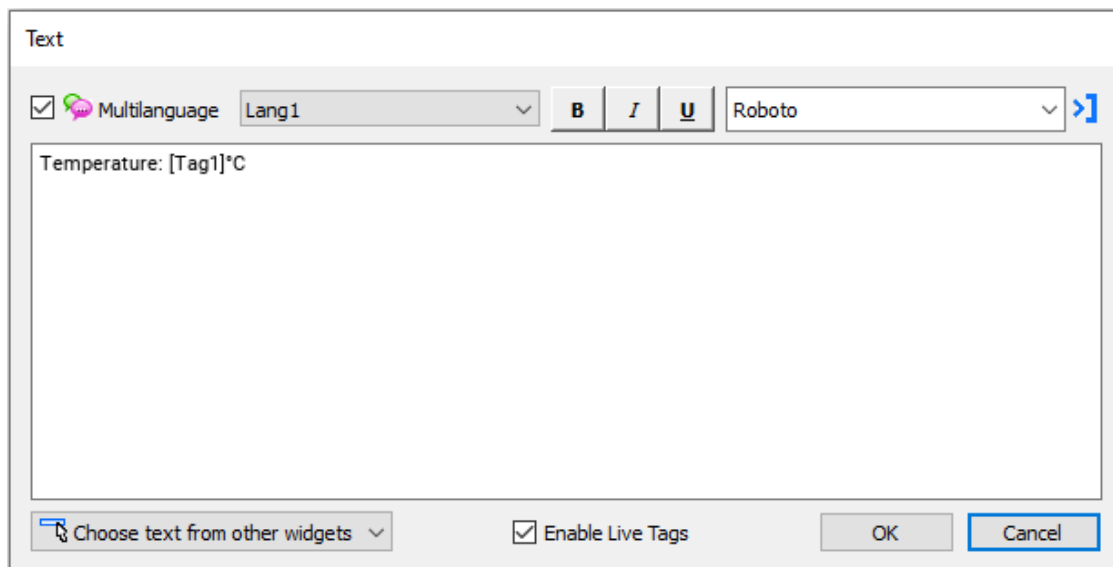
Property	Description
<b>Live Tags</b>	<p>Enable to use tags values inside the text message. See <a href="#">"Live Tags" on page 34</a></p> <ul style="list-style-type: none"> <li>- Enable Live Tag            Enable live tags placeholder</li> <li>- Dynamic Subscription    When true, only the tags that are visible are retrieved from the communication protocol. When false, all tags are kept continuously updated even they are not visible.</li> </ul>
<b>Scrolling</b>	<p>Parameters to enable and configure the scrolling of the text message</p> <ul style="list-style-type: none"> <li>• None</li> <li>• Slow</li> <li>• Normal</li> <li>• Fast</li> <li>• Custom</li> </ul> <p>When the custom mode is selected, the below parameters can be defined:</p> <ul style="list-style-type: none"> <li>- Scroll type            For each timeout, the text is scrolled of a custom amount of characters or pixels. <ul style="list-style-type: none"> <li>• Characters</li> <li>• Pixels</li> </ul> </li> <li>- Scroll delay            The timeout after which label effectively start to scroll (mSec)</li> <li>- Scroll timer            The timeout which defines each scroll step (mSec)</li> <li>- Scroll dots or - Scroll characters    The number of pixels scrolled for each timer timeout or The number of character scrolled for each timer timeout</li> <li>- Scroll behavior        OnlyOnce Text scrolling stops after the first complete.  LoopWithDelay Text scrolling restart after each complete cycle, waiting for the delay.  Loop Text scrolling never stop</li> </ul>

Property	Description
<b>General</b>	<p>General properties</p> <ul style="list-style-type: none"> <li>- Id                      Widget identifier. You can leave the default value or rename it to have a more appropriate name</li> <li>- Visible                When false the widget is not visible</li> <li>- Opacity                The opacity-level describes the transparency-level, where 1 is not transparent at all, 0.5 is 50% see-through, and 0 is completely transparent.</li> <li>- Blink                    The text will blink</li> <li>- Lock                    When True, the widget cannot be selected and moved from the PB610 Panel Builder 600 page editor</li> <li>- Static Optimization    Normal PB610 Panel Builder 600 will decide the best optimization mode to use</li> <li>                              Static PB610 Panel Builder 600 optimize the widget assuming it will never be modified by the runtime</li> <li>                              Dynamic PB610 Panel Builder 600 will not add additional optimizations</li> </ul>
<b>Position</b>	<p>The widget position on the display. See "<a href="#">Widget position on the display</a>" on page 35</p>

Some properties have a couple of buttons:

-  Enter edit mode:  
you can directly type the tag name to use
-  Attach to tag:  
the dialog where select the tag to use will be opened

A double clicks over the label widget will open the edit dialog box where you can enter the text to display and set the main text properties.



## Live Tags

"Enable Live Tags" is enabled, text between square brackets are managed as tags place holders and will be rendered, from the runtime, using the tag value.

For example, the text label "*Temperature: [Tag1] °C*" will be rendered as:

**Temperature: 18 °C**

where "18" is the value contained inside Tag1

### Tags

- [TagName]  
The tag value is read and continuously updated



Use '\' before '[' if you want to show the '[' in the description string, for example: `\[Tag\1\]` will display the string "[Tag1]".

Use '\', even when the tag label contains square brackets. For example, to display the live tag value of tag "TAG]3" or "TAG[3]" use:

- TAG]3 = [TAG]3
- TAG[3] = [TAG[3]]

### Array Tags

To reference the entire array (all elements will be shown):

- [TagName]  
All array elements will be displayed using a comma separate list.
- [TagName[-1]]  
All array elements will be displayed using a comma separate list.

To reference an element of the array:

- [TagName.Index]  
Example: [MyARRAY.5] will display the sixth element of the MyARRAY
- [TagName[TagIndex]]  
Example: [MyARRAY[TagIndex]] will display the sixth element of the MyARRAY when TagIndex is 5

### Data Formats

Placeholder characters can be used to control how to display the tag value (see "Custom Formats" on page 38)

- [TagName|format("###")]

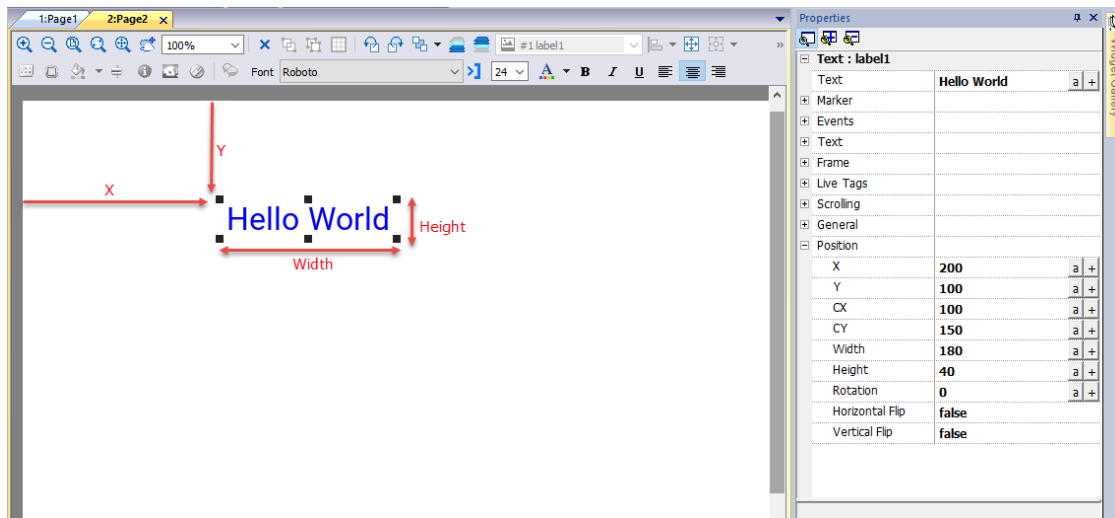
Example:

Live: [fCounter|format("#.00")] - Triggered: [!fCounter|format("#.00")]



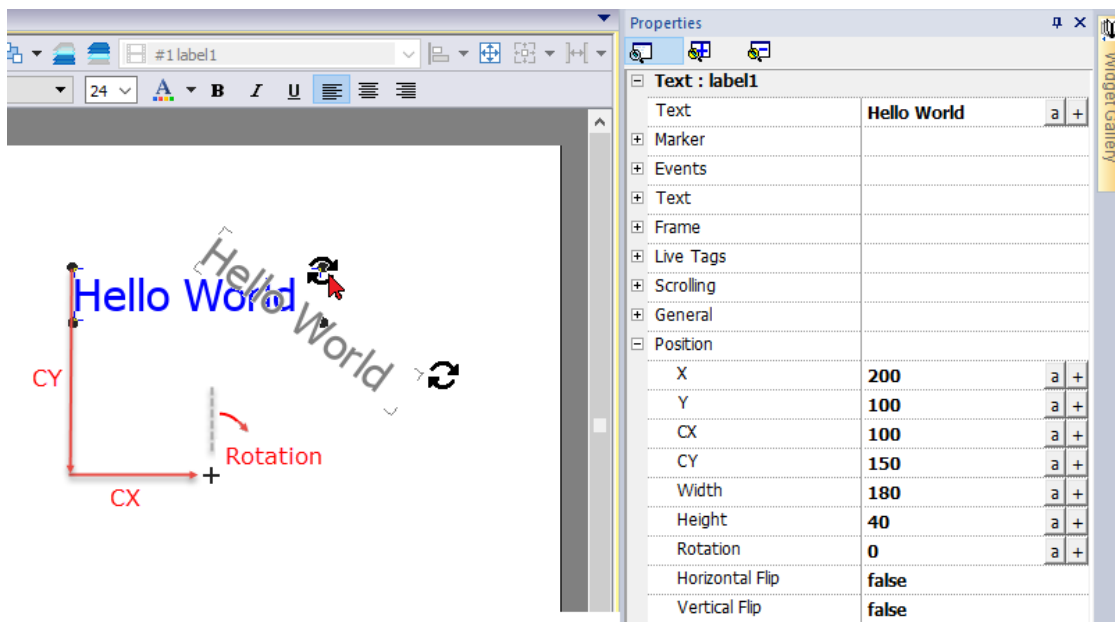
Note that by default, all tags are displayed as an integer. If you want to display a float number, you have to specify how to show the number adding the decimal digits.

### Widget position on the display



### Rotation

To rotate a widget, click two times the widget. After the first click, the markers will become square, after the second click will become circles. Now click the mouse over a circle marker and drag and drop to rotate the widget. The rotation center is identified by the CX and CY parameters.



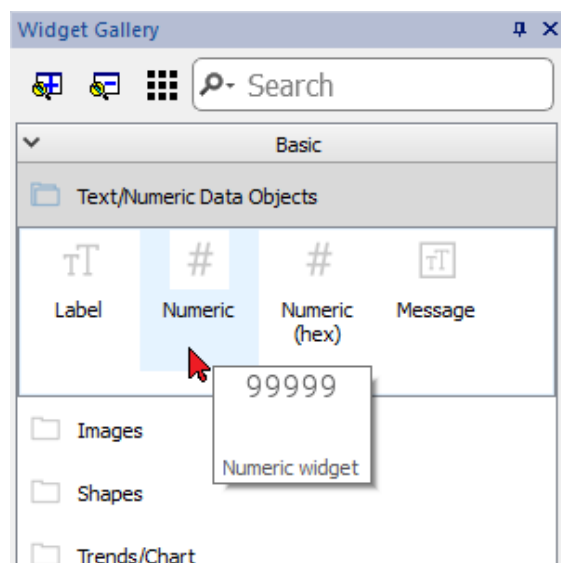


Note that all "Position" properties can be attached to tags and can be modified dynamically at runtime to move the widget.

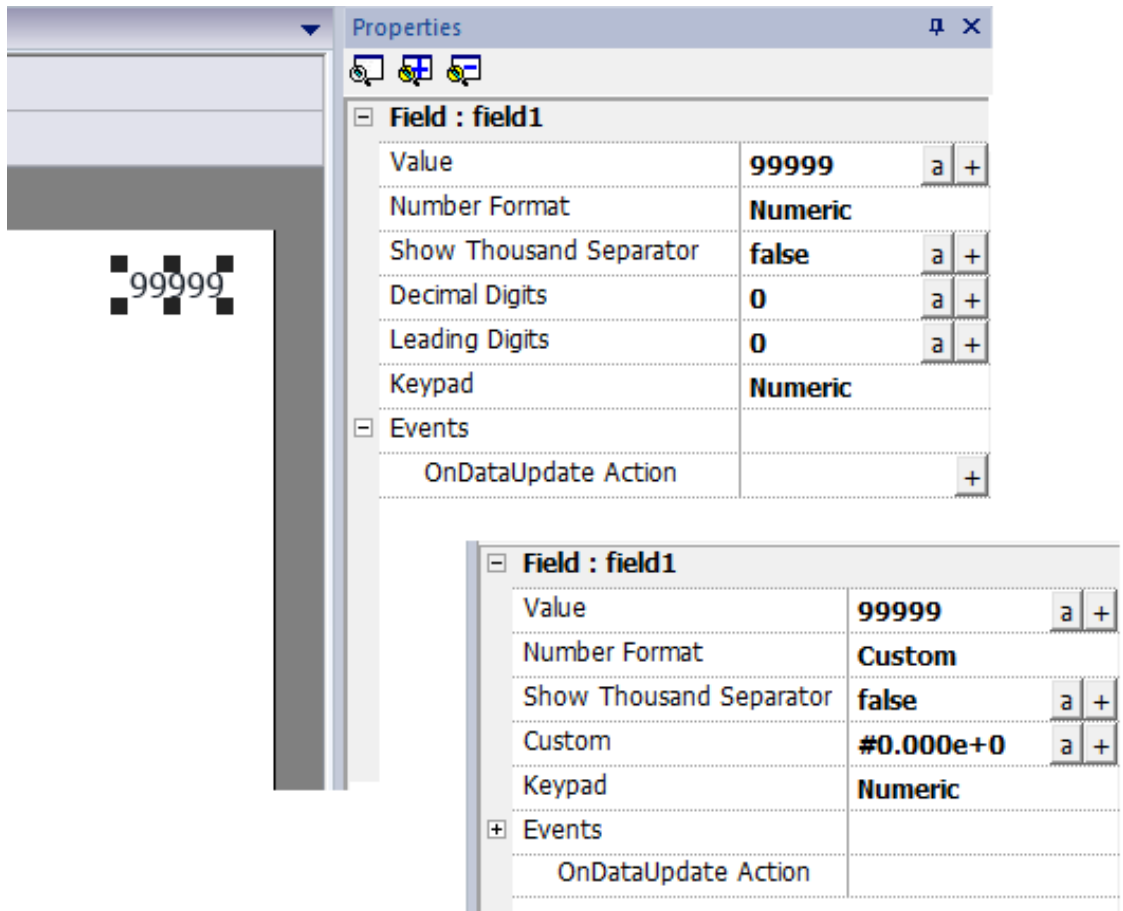
## Data field widget

The most common widget is the data field widget that give the possibility to display value of tags. (See ["Tag editor" on page 113](#)).

**Path: View > Toolbars and Docking Windows > Widget Gallery**




## Field properties





Property	Description
<b>Value</b>	Tag that contain the information to display
<b>Number Format</b>	<p>Display format</p> <ul style="list-style-type: none"> <li>• None No restrictions (system decide the format to use)</li> <li>• Numeric Numerical format. Decimal digits and Leading digits can be used to better define the number format</li> <li>• hex Hexadecimal format. Leading digits can be used to better define the number format</li> <li>• HEX The same of "hex" format but using uppercase</li> <li>• scientific Scientific format. Decimal digits can be used to better define the number format</li> <li>• SCIENTIFIC The same of "scientific" format but using uppercase</li> </ul>



Property	Description
	<ul style="list-style-type: none"> <li>Custom Use the additional "Custom" parameter to better define the format to use (see the below table)</li> </ul>
<b>Show Thousand Separator</b>	To show/hidden the thousands separator
<b>Decimal Digits</b>	Number of decimal digits to show (not available on all format types)
<b>Leading Digits</b>	Number of leading digits to show (not available on all format types)
<b>Custom</b>	String that define the number format to use (available only when selected Number Format is CUSTOM)
<b>Keypad</b>	Keypad type to pop up to edit the tag value <ul style="list-style-type: none"> <li>None</li> <li>Alphabetic, Numeric, Etc. Pop up a predefined keypad or a user keypad (see "<a href="#">Keypads</a>" on page 395)</li> <li>Wheel Keypad will not be displayed . Wheel can be used to increment/decrement the numeric value</li> <li>Macro Keypad will not be displayed . Keyboard macro can be used to enter keys (see "<a href="#">Keyboard actions</a>" on page 199)</li> </ul>
<b>Events</b>	
<b>OnDataUpdate Action</b>	Commands list to execute any time the tag value changes (See " <a href="#">Actions</a> " on page 193 for the available commands)

 The character used as thousand separators (point) and the character used as decimal separator (comma) can be modified from the global Project Property. See "[Regional Settings](#)" on page 93

Some properties have a couple of buttons:

-  Enter edit mode:  
you can directly type the tag name to use
-  Attach to tag:  
the dialog where select the tag to use will be opened

### Custom Formats

In custom property, the allowed chars are "#" "." "0" "h" "H" "e" "E"

Use the place holder characters to control the display of digits before and after the decimal place. Use the number sign (#) if you want to display only the significant digits in a number. This sign does not allow the display non-significant zeros. Use the numerical character for zero (0) if you want to display non-significant zeros when a number might have fewer digits than have been specified in the format code.

If a number has more digits to the left of the decimal point than there are placeholders in the format code, the extra digits are displayed. However, if a number has more digits to the right of the decimal point than there are placeholders in the format code, the number is rounded off to the same number of decimal places as there are placeholders.

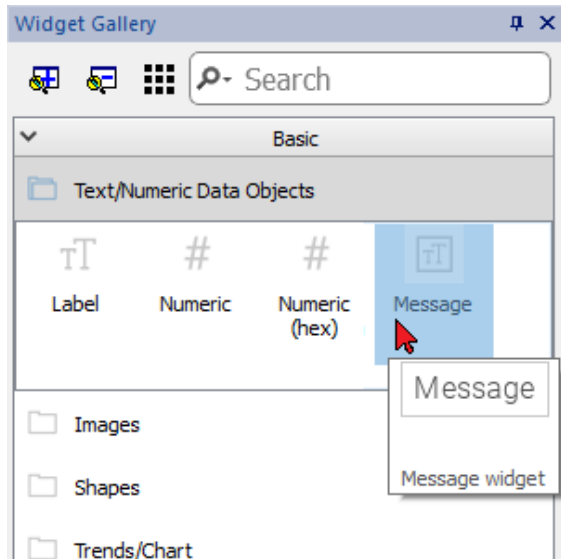
**Examples**

To display	As	Place Holder
123	0123	000#
1500	5DC	#H
1500	5dc	#h
1500	05DC	000#H
123.456	123.46	#.##
123.456	000123.456000	00000#.000000
12,200,000	1.22E+07	#0.00E+00
12,200,000	12.2E+6	#0.0E+0

## Message widget

The message widget gives the possibility to display text a message indexed from a tag value.

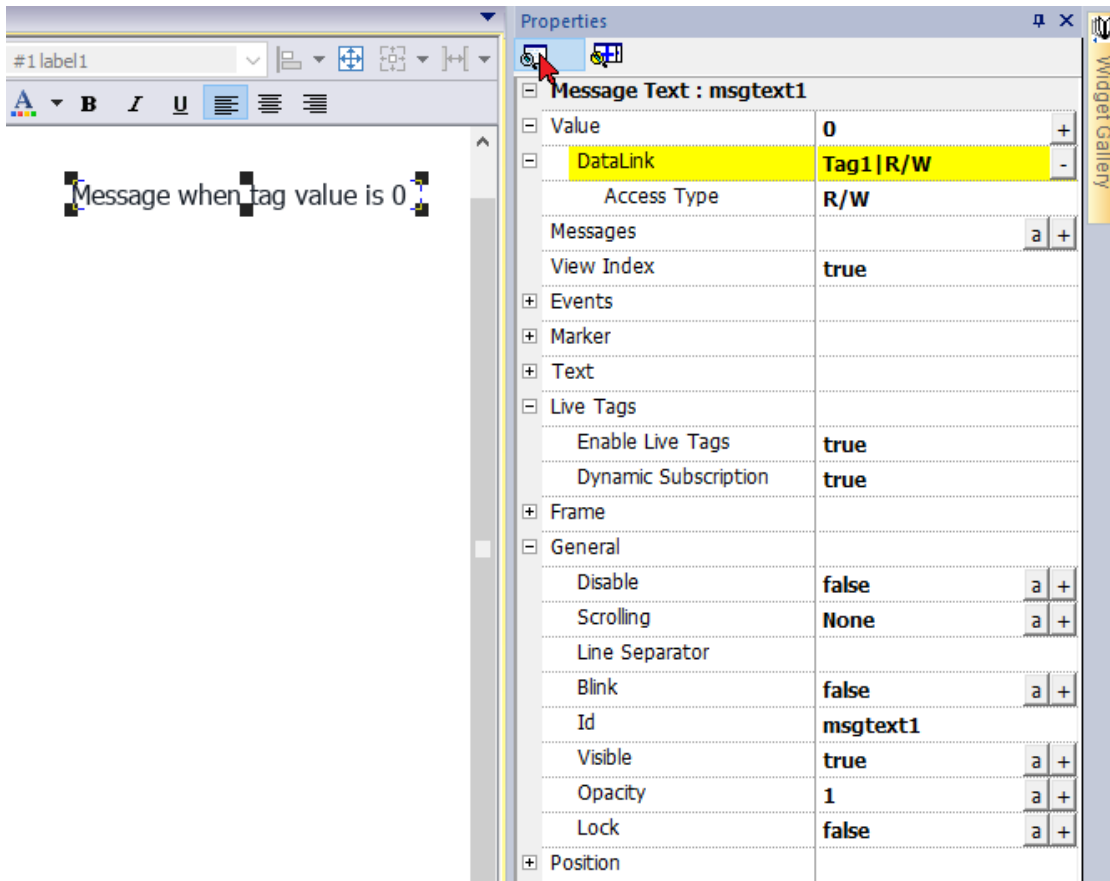
**Path:** *View > Toolbars and Docking Windows > Widget Gallery*




### Message properties

Drag and drop the widget inside the page and select the widget to open the properties dialog of the widget.



Note that some properties are visible only when the "Show Advanced Properties" button is selected.



Property	Description
<b>Value</b>	The tag name to used to dynamically select the message to display. See <a href="#">"Attaching widget to tags "</a> on page 42  When the "Access Type" is R/W, the value of the attached tag can be changed by clicking on the message. As a result, the message will be updated to be aligned with the new index value.
<b>Messages</b>	The list of messages to display. Click + to open a dialog where you enter messages and the associated index
<b>View Index</b>	If true, when the widget is editable (Access Type = R/W) the selection dialog will also show the index number associated with each message.
<b>Events</b>	Action that will be executed if widget contents change. See <a href="#">"Events"</a> on page 59
<b>Marker</b>	Enable a Marker around the widget (It is visible only inside PB610 Panel Builder 600)
<b>Text</b>	Text properties (font, color, size, etc.)

Property	Description
<b>Live Tags</b>	<p>Enable to use tags values inside the text message. See <a href="#">"Message widget" on page 39</a></p> <ul style="list-style-type: none"> <li>- Enable Live Tag            Enable live tags placeholder</li> <li>- Dynamic Subscription    When true, only the tags that are visible are retrieved from the communication protocol. When false, all tags are kept continuously updated even they are not visible.</li> </ul>
<b>Frame</b>	Parameters to enable and configure a frame of the widget and/or a color for the background
<b>General</b>	<p>General properties</p> <ul style="list-style-type: none"> <li>- Disable                    Disable user inputs on the widget</li> <li> <div style="display: flex; align-items: center;">  <p>Attaching a tag at this property is possible to enable/disable the possibility to modify the value at runtime</p> </div> </li> <li>- Scrolling                 Parameters to enable and configure the scrolling of the text message</li> <li> <p>When enabled, all messages are linked together and displayed in scrolling mode.</p> </li> <li>- Line Separator         Characters to insert between messages when shown in scrolling mode</li> <li>- Blink                      The text will blink</li> <li>- Id                          Widget identifier. You can leave the default value or rename it to have a more appropriate name</li> <li>- Visible                    When false the widget is not visible</li> <li>- Opacity                    The opacity-level describes the transparency-level, where 1 is not transparent at all, 0.5 is 50% see-through, and 0 is completely transparent.</li> <li>- Lock                        When True, the widget cannot be selected and moved from the PB610 Panel Builder 600 page editor</li> </ul>
<b>Position</b>	The widget position on the display. See <a href="#">"Widget position on the display" on page 35</a>

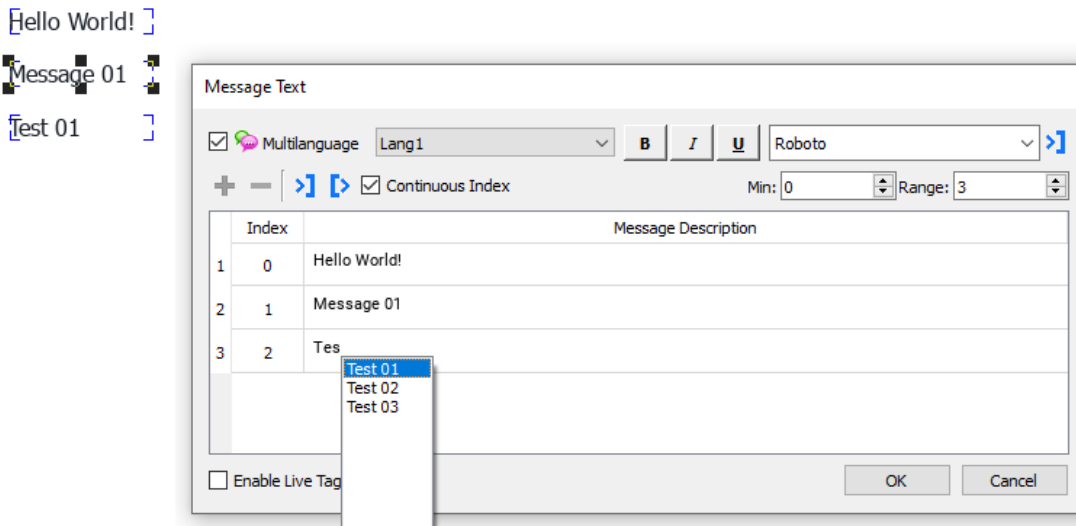
Some properties have a couple of buttons:

-  Enter edit mode:  
you can directly type the tag name to use
-  Attach to tag:  
the dialog where select the tag to use will be opened

A double clicks over the label widget will open the edit dialog box where you can enter the text to display and set the main text properties.


### Auto-completing dialog

When you start to type the message the auto-completing dialog will propose the available messages already typed.



### Import/Export

The import/export buttons, give you the possibility to import or export the entire messages list inside a .xml file that can be edit/modify using external tools.

 Export messages inside an editable .xml file

 Import messages from a .xml file



If you need to use the same message widget in different places, to save maintenance time you can create and duplicate a custom widget. When a custom widget is configured with "Only Logic" or "Full" inheritance mode, the modify (e.g. add, remove or change messages list) of a single widget will be propagated to all widgets. See ["Creating a custom widget" on page 522](#)



## Attaching widget to tags

To control a widget and animate it through live data it is possible to bind a specific property to different data sources. For example it is possible to bind the gauge **Value** property to a probe temperature tag, or the **Display** property to a recipe

data

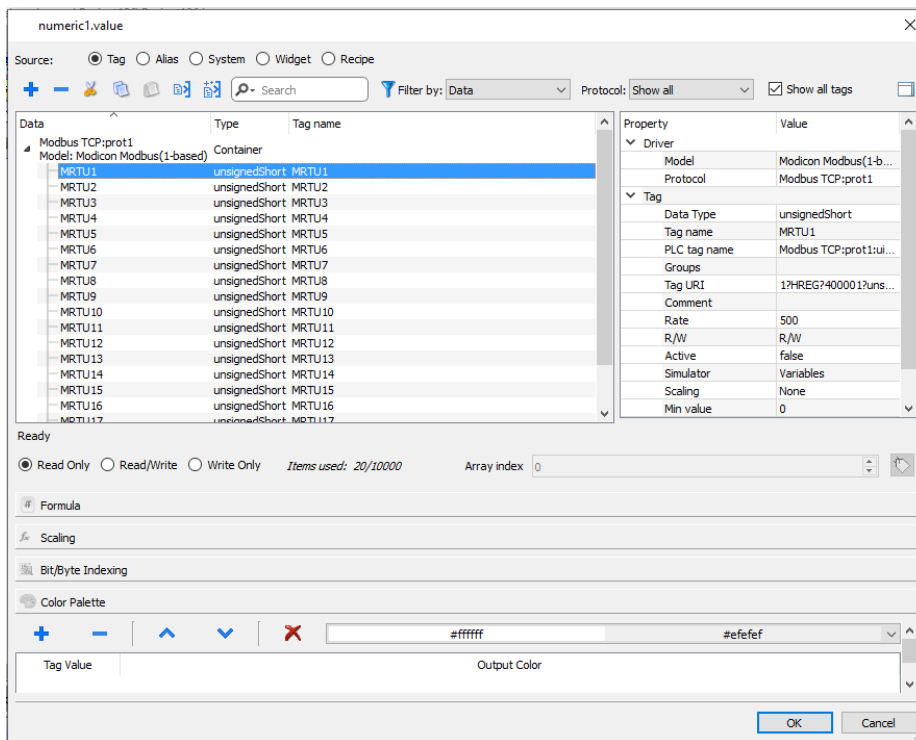
## Data sources

Elements to which an object property can be attached:

Data source	Description
Tag	Tag defined in the Tag Editor
Alias	Indexed tag address
System	Predefined system tags (see <a href="#">"System Variables (Attach To)" on page 143</a> )
Widget	Connect to a widget property (for example, value of a slider widget)
Recipe	Data from the Recipe Manager (see <a href="#">"Recipes" on page 283</a> )

## Attaching a property to a tag

1. Click **+** in the **Properties** pane.
2. In **Source** choose the data source, in the list choose a protocol and the tag. Use the **Search** box to filter tags.



3. Set the access type (for example **Read Only**). The **Array Index** field appears when the selected tag is an array to identify the element of the array to use. The indirect index mode, through an additional tag, is supported.
4. Click **OK** to confirm.

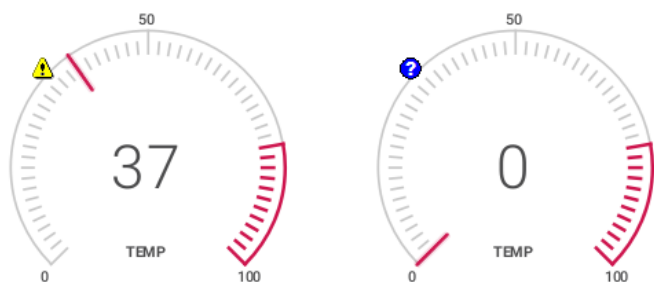
The icons adjacent to the tag name highlight when a definition does not match the tag definition in the dictionary, or when missing. If the **Show all tags** is selected, all the dictionary tags are shown also if not imported within the application. A double-click will import the tags from the dictionary.

See [""Attach to" parameters" on page 49](#) for details.

## Communication Error

Two icons may appear close to widgets that have an attached tag.

⚠5535 ?5535



Communication error



Data not yet available (slow communication protocol)

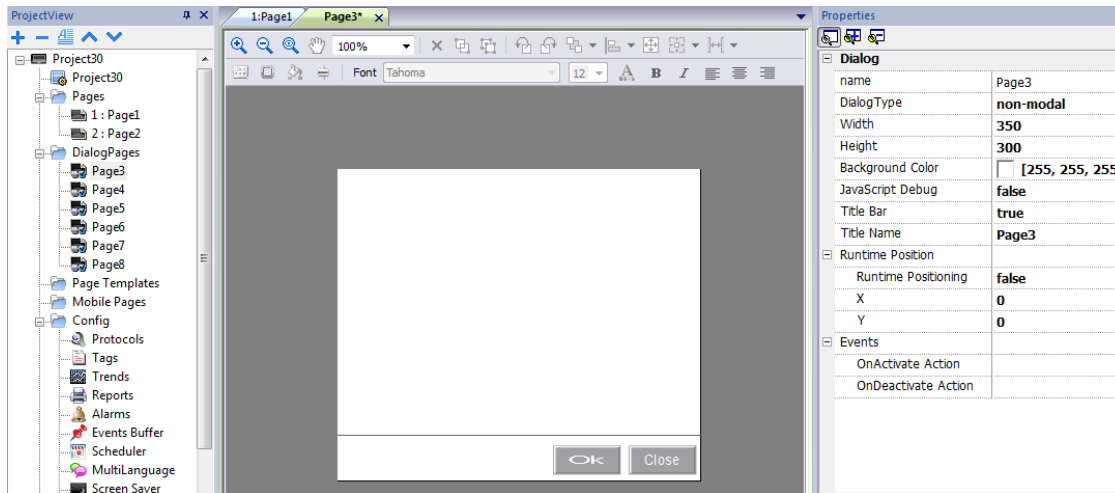
## Dialog pages

*Path: ProjectView > Dialogs*

Dialog pages are opened at runtime on top of the current page on project request. They are used to notify alarms, errors or to require user action.

### Main dialog properties

Property	Description
<b>Dialog Type</b>	<p><b>modal</b> = user cannot return to main project window/page until dialog is closed.</p> <p><b>non-modal</b> = user can continue to use main project window (or other non-modal dialogs) while a dialog is shown on top of it.</p>
<b>Title Bar</b>	<p><b>true</b> = dialog title displayed</p> <p><b>false</b> = no dialog title displayed</p>
<b>Title Name</b>	Dialog title. Only if <b>Title Bar</b> =true.
<b>Runtime Position</b>	<p>Dialog fixed position</p> <p><b>false</b> = Dialog will be placed centered on the screen</p> <p><b>true</b> = Dialog will be placed with upper-left corner at position X and Y</p>



### Maximum number of dialogs

Maximum number of open dialogs is defined in ["Functional specifications and compatibility"](#) on page 611.

When the maximum number of open dialogs is reached, the oldest dialog is closed to open the new one.





# 5 Programming concepts

---

Programming for PB610 Panel Builder 600 is based on a few basic concepts and behaviors.

---

<b>Data types</b> .....	<b>48</b>
<b>"Attach to" parameters</b> .....	<b>49</b>
<b>Formula</b> .....	<b>55</b>
<b>Events</b> .....	<b>59</b>
<b>Widgets positioning</b> .....	<b>62</b>
<b>Managing overlapping widgets</b> .....	<b>63</b>
<b>Grouping widgets</b> .....	<b>64</b>
<b>Changing multiple widgets properties</b> .....	<b>66</b>
<b>Changing fill color property according to tag values</b> .....	<b>67</b>

# Data types

When creating a tag you have to specify its properties. Data type are specific to PB610 Panel Builder 600, memory type are specific to the selected protocol. Choose the value according to the internal representation you need for the selected controller address.



Note: arrays type use the same data type followed by "[]" (i.e.: boolean [])



On some PLCs (e.g. CODESYS) arrays start from index 1 while on PB610 Panel Builder 600 the arrays start from index 0. This means that arrayTAG[1] on PLC is the arrayTAG[0] on PB610 Panel Builder 600.

Data Type	Memory Space	Limits
boolean	1-bit data	0 ... 1
byte	8-bit data	-128 ... 127
short	16-bit data	-32768 ... 32767
int	32-bit data	-2.1e9 ... 2.1e9
int64	64-bit data	-9.2e18 ... 9.2e18
unsignedByte	8-bit data	0 ... 255
unsignedShort	16-bit data	0 ... 65535
unsignedInt	32-bit data	0 ... 4.2e9
uint64	64-bit data	0 ... 1.8e19
float	IEEE single-precision 32-bit floating point type	1.17e-38 ... 3.4e38
double	IEEE double-precision 64-bit floating point type	2.2e-308 ... 1.79e308
string	Array of elements containing character code defined by selected encoding	

## System Time

Format of System Time inside the HMI Device is the Unix time (also known as Epoch time). It is the number of seconds that have elapsed since the Unix epoch, that is the time 00:00:00 UTC on 1 January 1970.

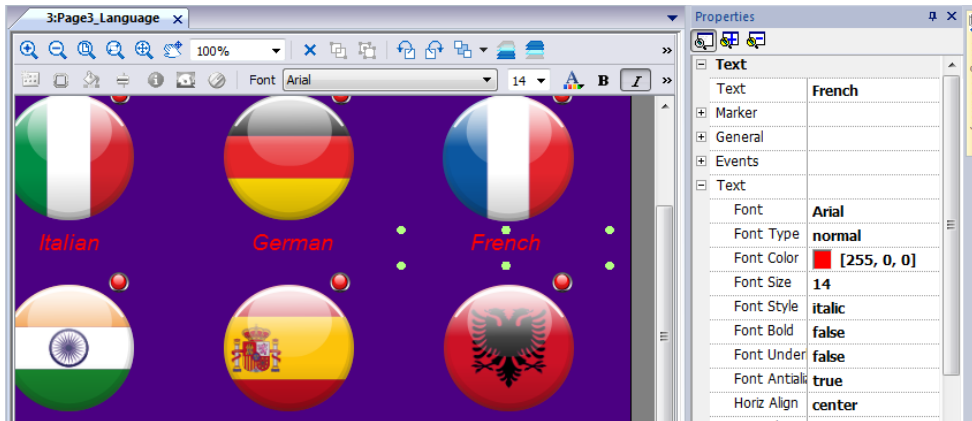
Example:

Tag Value	System Time	ISO 8601
0	01/01/1970 – 01:00:00	1970-01-01T00:00:00+00:00
1	01/01/1970 – 01:00:01	1970-01-01T00:00:01+00:00
60	01/01/1970 – 01:01:00	1970-01-01T00:01:00+00:00

## "Attach to" parameters

### Object properties

In PB610 Panel Builder 600 the properties of an object placed on a page can be set at programming time or configured to be dynamic. To change a property at programming time use the page toolbar or the property pane. Select the object first to see its properties displayed.

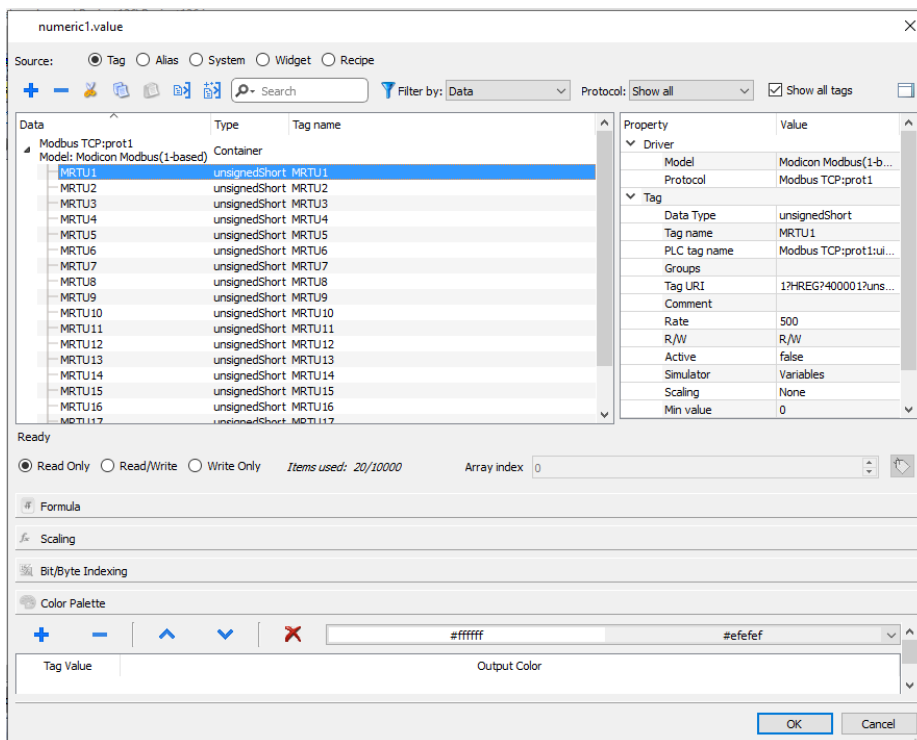


The page toolbar shows only the most common object properties, while the property pane show all the properties in a basic or advanced view.

To change a property value dynamically you can attach it to tags or variables.

### Attaching a property to a tag

1. Click **+** in the **Properties** pane.
2. In **Source** choose the data source, in the list choose a protocol and the tag. Use the **Search** box to filter tags.



3. Set the access type (for example **Read Only**). The **Array Index** field appears when the selected tag is an array to identify the element of the array to use. The indirect index mode, through an additional tag, is supported.
4. Click **OK** to confirm.

The icons adjacent to the tag name highlight when a definition does not match the tag definition in the dictionary, or when missing. If the **Show all tags** is selected, all the dictionary tags are shown also if not imported within the application. A double-click will import the tags from the dictionary.

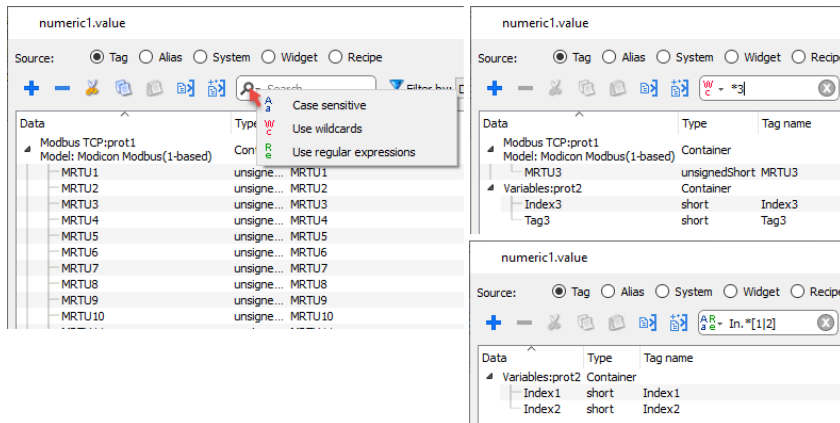
## Data sources

Elements to which an object property can be attached:

Data source	Description
<b>Tag</b>	Tag defined in the Tag Editor
<b>Alias</b>	Indexed tag address
<b>System</b>	Predefined system tags (see <a href="#">"System Variables (Attach To)" on page 143</a> )
<b>Widget</b>	Connect to a widget property (for example, value of a slider widget)
<b>Recipe</b>	Data from the Recipe Manager (see <a href="#">"Recipes" on page 283</a> )

## Advanced search

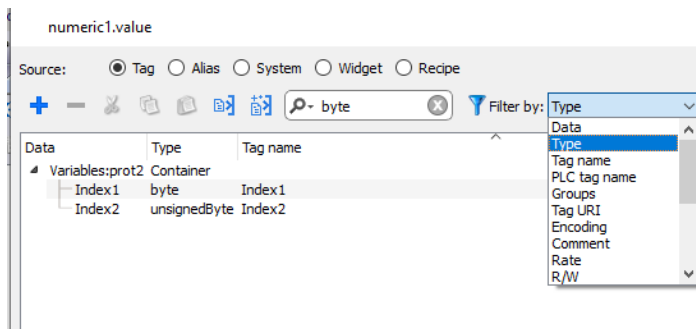
Various syntax options can be applied to search box:



Main options	Function
<b>Wildcards</b>	Search using simple wildcards matching . Character '?': matches any single character. Character '*': matches zero or more of any characters. "[...]": sets of characters can be represented in square brackets.
<b>Regular Expression</b>	Describes character pattern. See <a href="https://en.wikipedia.org/wiki/Regular_expression">https://en.wikipedia.org/wiki/Regular_expression</a> for additional details regarding regular expressions.

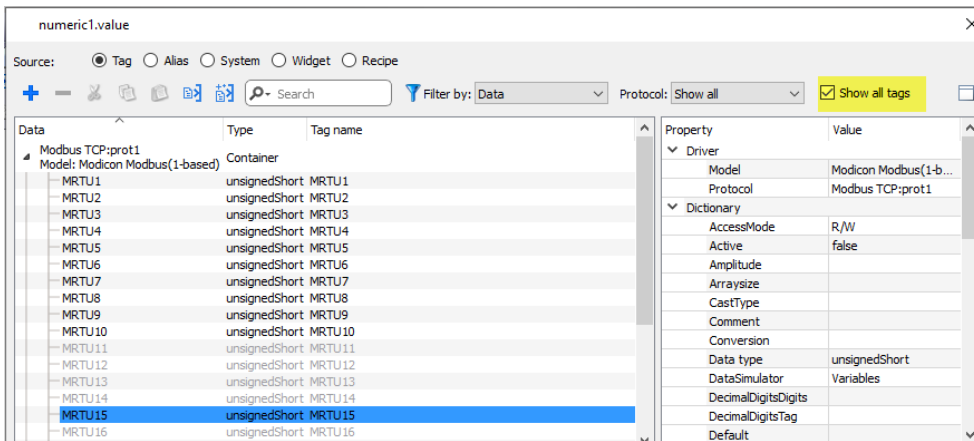
## Filtering tags

Choose various tag filter criteria:

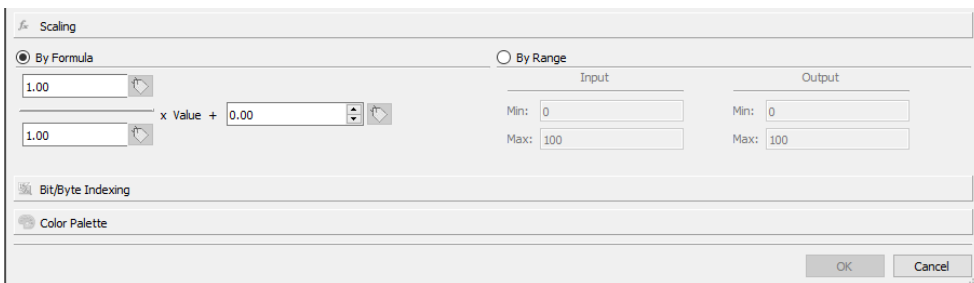


## Showing dictionary tags

When **Show all tags** is checked, tags that belong to one dictionary but have not been imported yet, appear in blue color. You can select and double-click a tag to import it into the project.



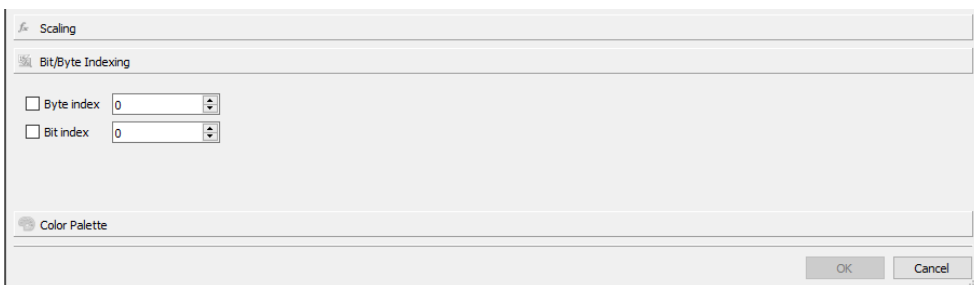
## Converting tag value



**Scaling** tab converts the tag value. In **By Range** section set the input and output range: the system will automatically calculate the scaling factors.

## Extract tag bit/byte based on index

Allows extracting a single bit or byte content from a word depending on the specified bit or byte number

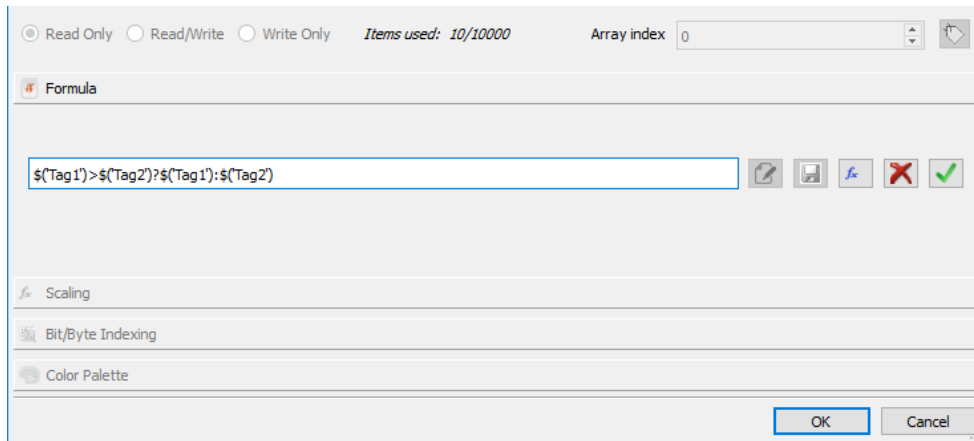


## Use a formula to calculate the value to use

Allow to use a formula to calculate the value to use. See ["Formula"](#) on page 55 chapter for additional details.

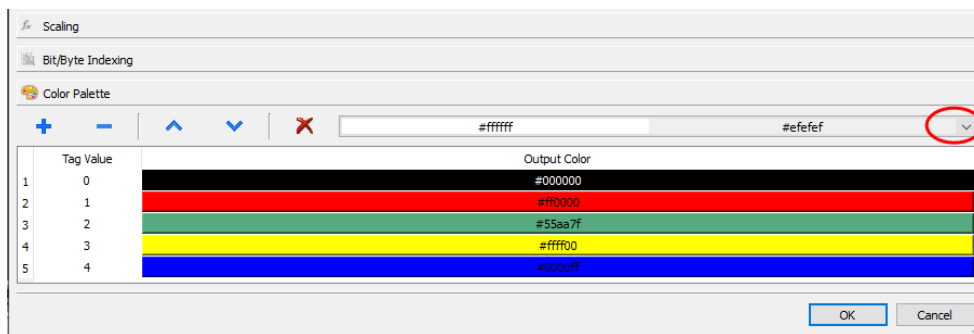


Note that using a formula the datalink will be ReadOnly



## Mapping tag values to color

Allows to mapping numeric or string tag values to colors. For example, this option can be used to change the color of a button.



Section	Function
	From the toolbar add/remove or move up/down the colors lines. The tag value is editable and you can modify the sequence values.
	Last defined color combination is saved automatically and can be retrieved from the color toolbar.

Tag value could be a range of values separate by a comma, examples:

- 5, 10-15, 20
- A, AB, C



It is responsibility of the application's developer define all items correctly to cover all possible application's values, we could have unexpected color when the value is not defined inside the defined colors palette.



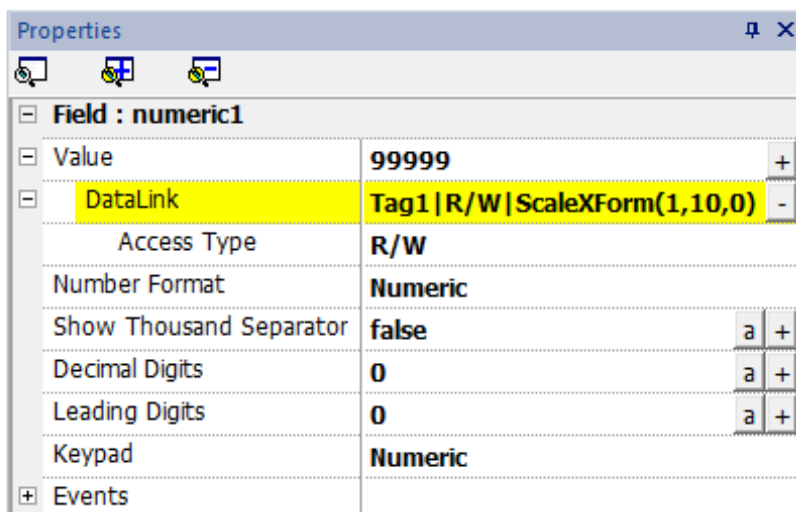
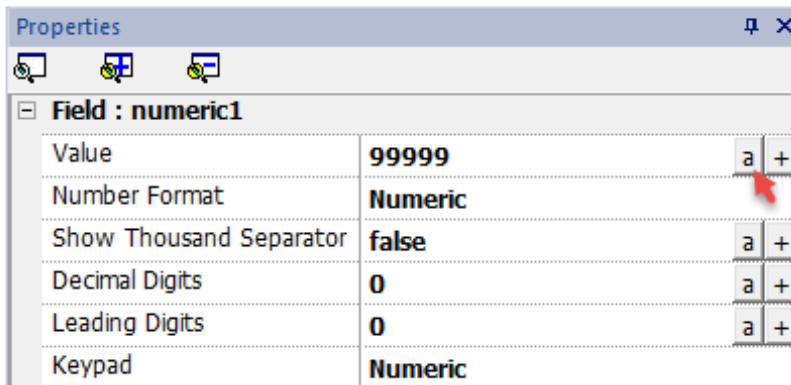
**Note that the mapping tag value to color will return a string data type (e.g. “#FF0000”)**

## Datalink Serialization

Instead of use the above “Attach to...” dialog box, datalinks can be entered, or modified, manually.



Click a button in the **Properties** pane and enter the text that describe the datalink



The data link format is:

```
Tagname [index] | [Attribute] | [XForm] | [XForm] | ...
```

on in case of formula:

```
= <formula>
```

Example:

- arrayTag[2]
- Tag[0]index]
- Alarm triggered: \_SysPropMgr
- Tag|R/W|ScaleXForm(1,10,0)
- Tag|R/W|ScaleXForm(1,10,0)|ByteIndexXForm(1)|ColorPaletteCustomXForm(0#00aa7f,1#ff0000)
- =\$('Tag1')>\$('Tag2')?\$('Tag1'):\$('Tag2')
- =\$Contains(\$('Tag4'),\$('Tag3'))
- =\$Pow(2,\$('Tag2'))

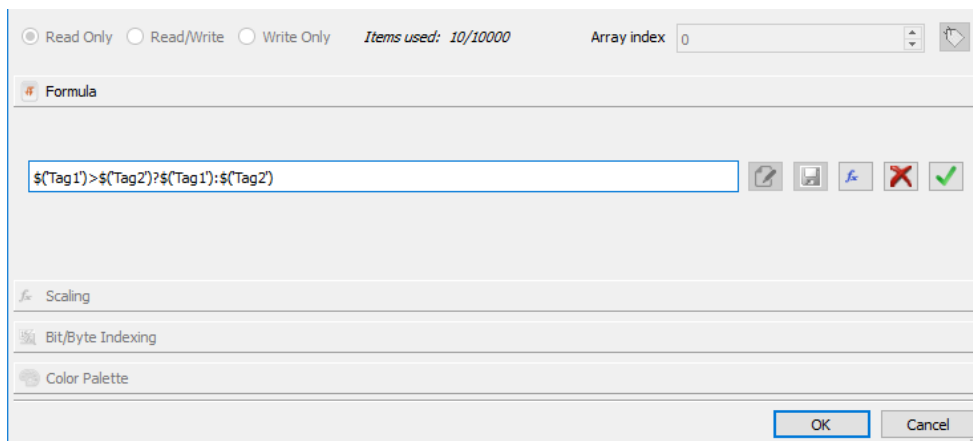
# Formula

A formula is an expression made of:

- Operators: can be the basic mathematics operations, logic operators, compare operators or basic string operators.
- Operands: can be literals (numbers and strings used as constants) and references to tags.

Round brackets are supported as priority operators. The operator \$ will be used to call functions and, in particular, to referring to a tag (see below for examples).

**The attach to dialog allow to use a formula to calculate the value to return.**



## Commands



Enter edit mode



Save the entered formula inside the formulas' library to have the possibility to reuse the same formula inside other places of the project.



Open the formulas' library to select an already defined formula.



Removing the entered formula





Confirming the entered formula

When you are in edit mode you can simple edit the formula and double click tags or functions from the library to add them inside the formula.

Example of formulas are:

- \$('Tag1')+\$('Tag2')
- \$('Tag1')&\$('Tag2')
- \$('Tag1')>\$('Tag2')?\$('Tag1'):\$('Tag2')
- \$Pow(2,\$('Tag2'))
- \$Contains(\$('Tag1'),\$('Tag2'))

## Syntax for formula

Basic Operations	
'Text'	String literal
NUMBER	Number literal, e.g. 169857 or 13.547
String( ... )	Cast to string (note there is not \$)
Number( ... )	Cast to number (note there is not \$)
\$FuncName( param1, param2, ... )	General function call. (Both default and user ones)
\$('TagName')	Tag, or widget property, or recipe, etc.  <b>Note that tag name must be string literal</b>
\$('TagName')[index]	Element of a array tag.  <b>Note that tag name must be string literal</b>
exp1 ? exp2 : exp3	Ternary expression. If exp1 is true, then is taken exp2, otherwise is taken exp3. This is like using if/then/else statement

Math Operators	
+	Addition
-	Subtraction
*	Multiplication
/	Division
%	Module

Bitwise Operators	
&	Sets each bit to 1 if both bits are 1
	Sets each bit to 1 if one of two bits is 1
~	Inverts all the bits
^	Sets each bit to 1 if only one of two bits is 1
<<	Shifts left by pushing zeros in from the right and let the leftmost bits fall off
>>	Shifts right by pushing copies of the leftmost bit in from the left, and let the rightmost bits fall off
>>>	Shifts right by pushing zeros in from the left, and let the rightmost bits fall off

### Logical Operators

&&	AND
	OR
!	NOT

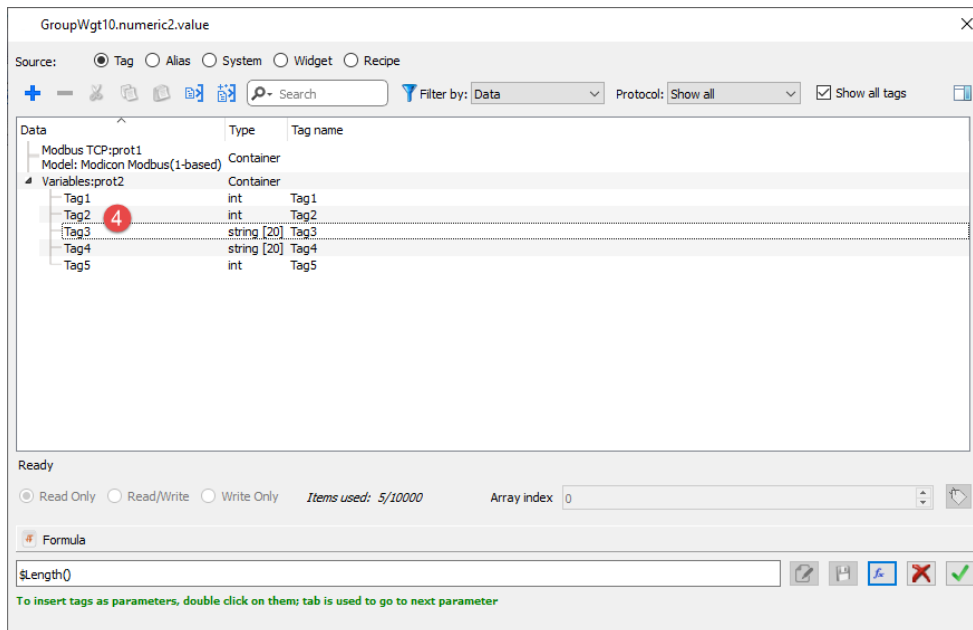
### Compare Operators

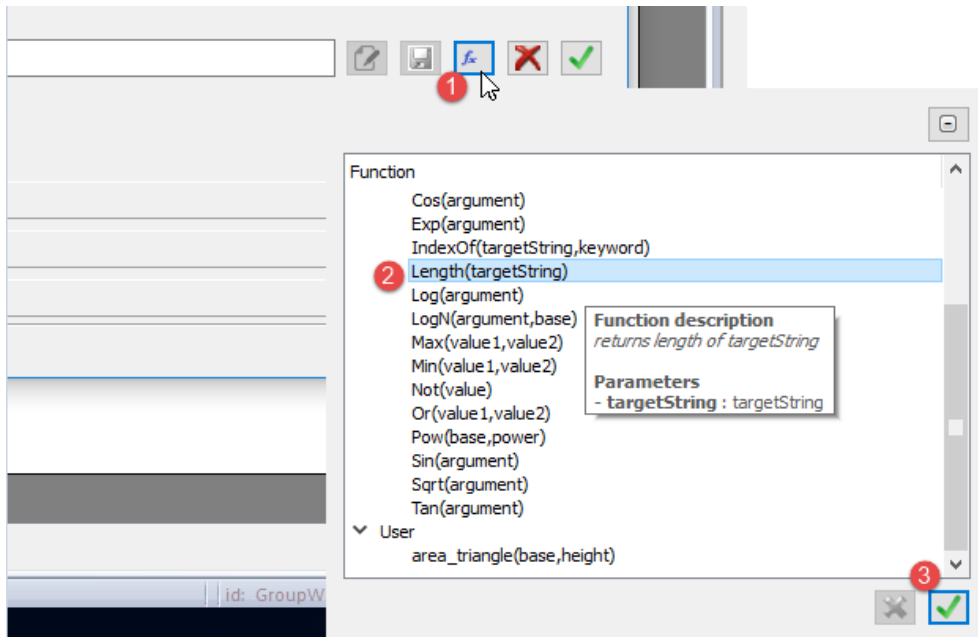
<	Less than
<=	Less than or equal to
>	Greater than
>=	Greater than or equal to
==	Equal to
!=	Not equal to

## Use predefined formula from the library

To insert a project or a predefined formula:

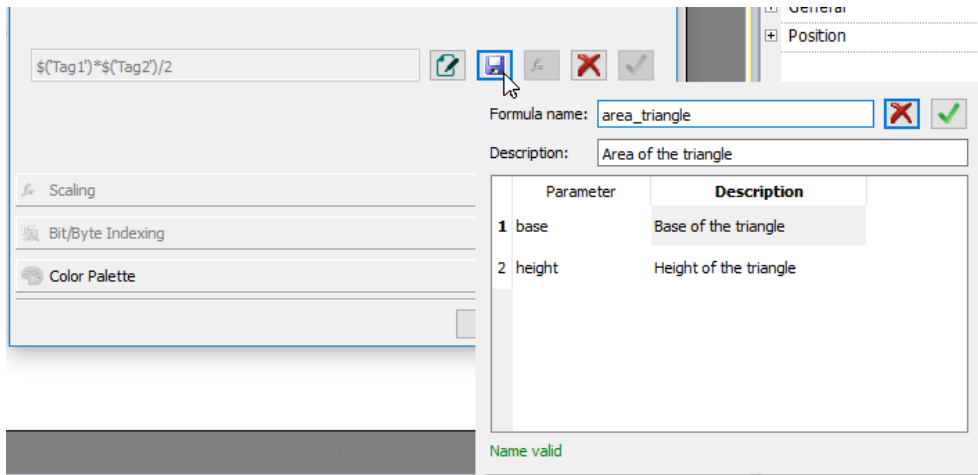
1. Open the formulas' library
2. Select the formula to use
3. Confirm the selected formula
4. Enter the arguments required from the selected formula



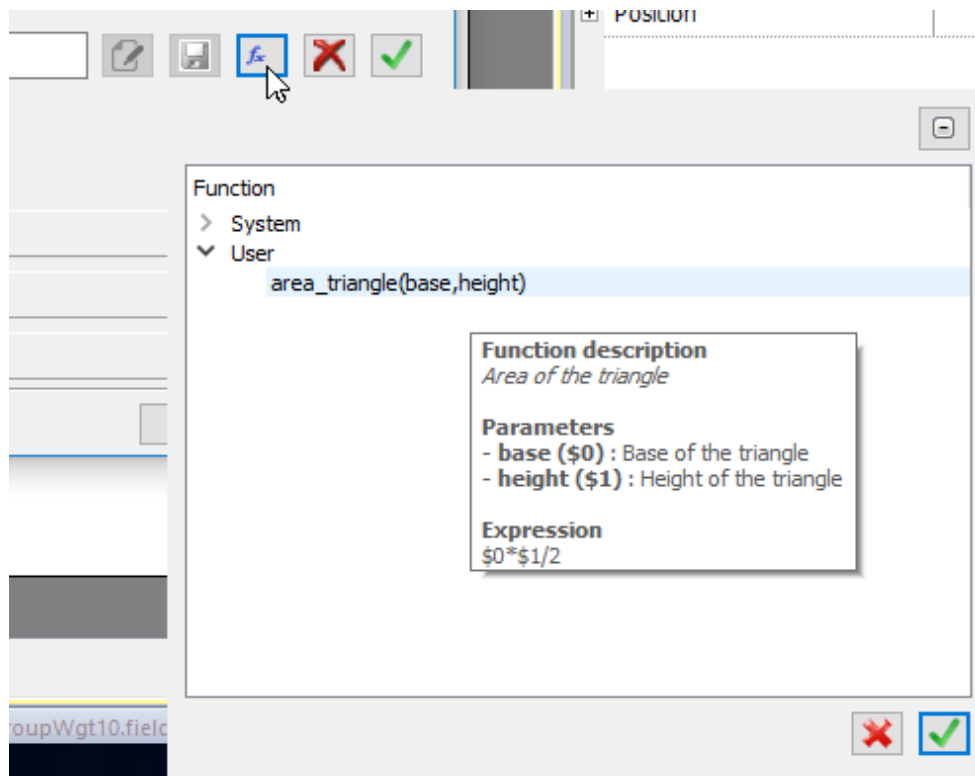


### Add user formulas into the library

After entering a new formula, using SAVE button is possible to store the new formula inside the project folder to make it available from the formulas' library.



A user formula could be retrieved from the formulas' library as for the other predefined formulas.



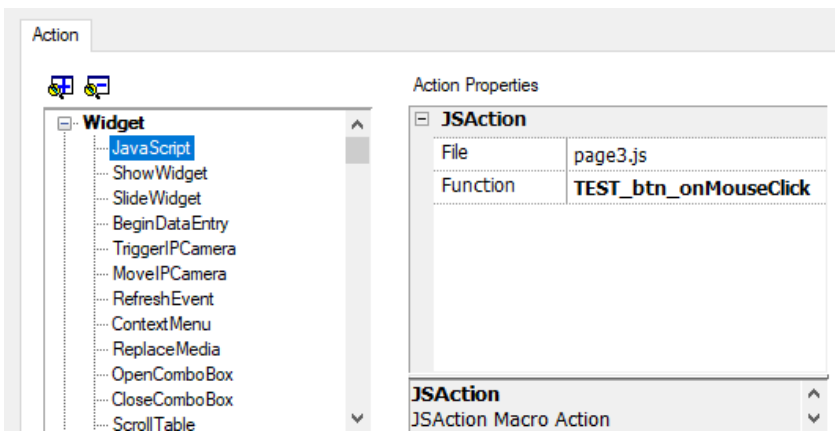
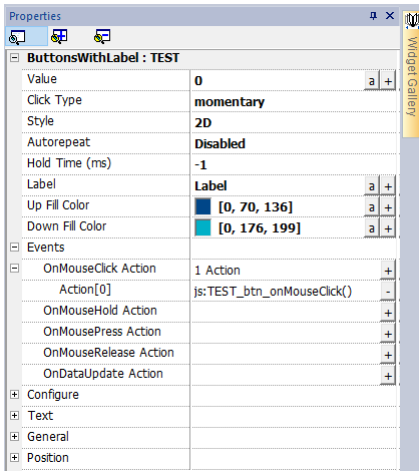
## Events

Events are used to trigger actions at project level and can be associated to:

- buttons / touch (click, press, release)
- mouse wheel
- external input devices like keyboards and mouse (click, press, hold, release, wheel)
- data changes (OnDataUpdate)
- switch of pages (OnActivate, OnDeactivate)
- alarms
- scheduler

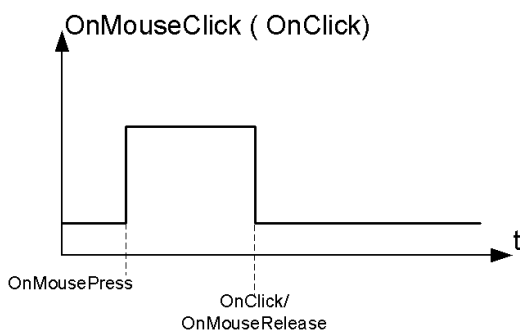
You can attach one or more actions to an event, so that they will be executed whenever the event occurs.

This example shows a JavaScript action activated by pressing a button.



## OnClick / OnMouseClick

Triggers the event when the button/key is pressed and released quickly.



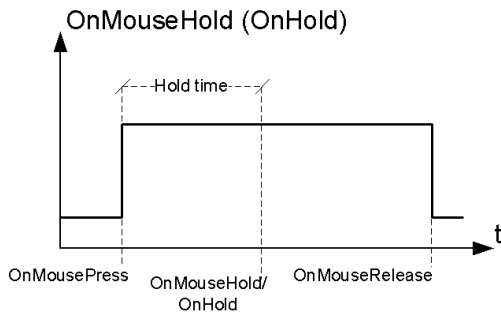
## OnHold/OnMouseHold

Triggers the event when the button/key is pressed and held pressed for a certain time set as **Hold Time** in the widget properties. Actions programmed for this event will be executed only after the hold time has expired.

The default **Hold Time** is configured in Project properties but can be redefined for each button/key. See "[Project properties](#)" on page 79.



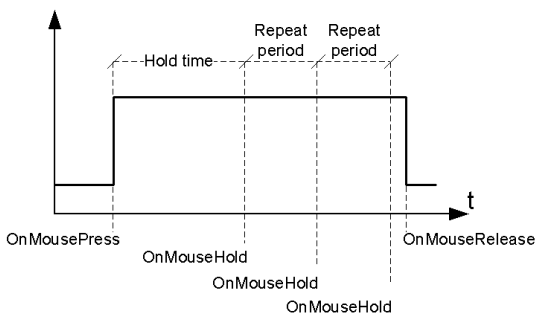
Note: If **Hold Time** is set to -1 for the widget, the project **Hold Time** value will be used.



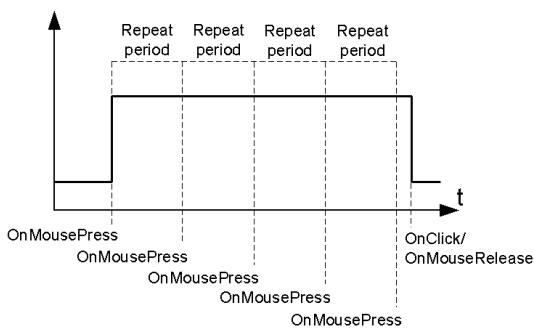
## Autorepeat

Enables auto repeat for a press or hold event of button or key. **Autorepeat Time** is specified in the Project properties but can also be redefined for each button or key

OnMouseHold (OnHold) and Autorepeat



OnMousePress and Autorepeat



## OnWheel

Triggers the event when a wheel (for example a USB mouse wheel) value changes. A wheel usually is used to increase/decrease values in a text box or attached to a tag.


## OnActivate

Triggers the event when a page is loaded. The event starts before widgets in the page are initialized.



## OnDataUpdate

Triggers the event when the tag value changes. The update moment depend on the time needed by the protocol to finish the update process. For example the **OnDataUpdate** event can be triggered or not, depending on whether data becomes available from protocol respectively after or before widgets being initialized for the first time. In particular, page change notifications are more likely to happen with slow protocols and remote clients.

 Note: The value read during **OnActivate** can be the same obtained from a subsequent **OnDataUpdate** event, since **OnDataUpdate** notifications are sent asynchronously.

## Widgets positioning

You can position widgets in the page using two methods:

- Snap to Grid
- Snap to Object

To display the grid, on the **View** menu, click **Show Grid**.

### Snap to Grid

*Path: View > Snap to Grid*

When you move or re-size an object, its top left corner will align with the nearest intersection of lines in the grid, even if the grid is not visible.

### Setting grid properties

*Path: View > Properties*

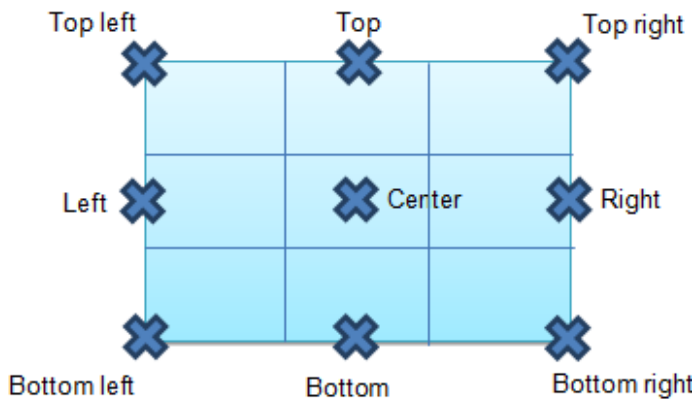
Parameter	Description
Spacing X	Space in pixel between two lines/dots on the X axis
Spacing Y	Space in pixel between two lines/dots on the Y axis
Type	Grid type (dot or line)
Color	Grid color

### Snap to Object

*Path: View > Snap to Object*

When you move an object, it will align with other objects on the page.

When you select an object, one of the following hot points is selected as the source of the snap point, depending on the area you pressed: top, top left, top right, bottom, bottom left, bottom right, left, right, center:

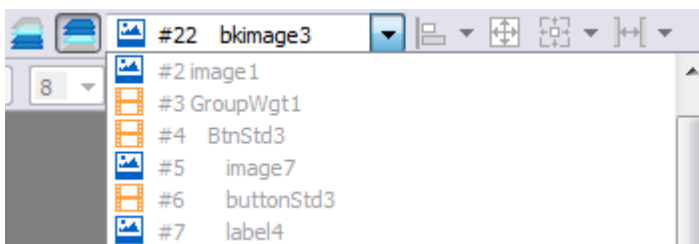


An algorithm finds a matching hot point among the near widgets hot points matching either the x or the y coordinates of the source snap point. For line widgets, the source snap points are the terminal points of the line.

## Managing overlapping widgets

When one or more widgets on the page overlap, you can manage their order so that one is displayed on top of the other.

The order of the widget on the page is shown in the combo box. A widget with greater z-order number is in front of an element with a lower z-order number. A picture icon identifies static objects, a movie frame icon identifies dynamic objects.



**Important: Correct ordering of widgets is essential for runtime performance since overlapping dynamic widgets can invalidate static optimization and reduce performance of HMI applications.**

## Hiding/showing widget on z-order

To hide widgets above a selected widget:

- On the toolbar click  and select a widget: all widgets above this one are hidden

To hide widgets below a selected widget:

- On the toolbar click  and select a widget: all widgets below this one are hidden

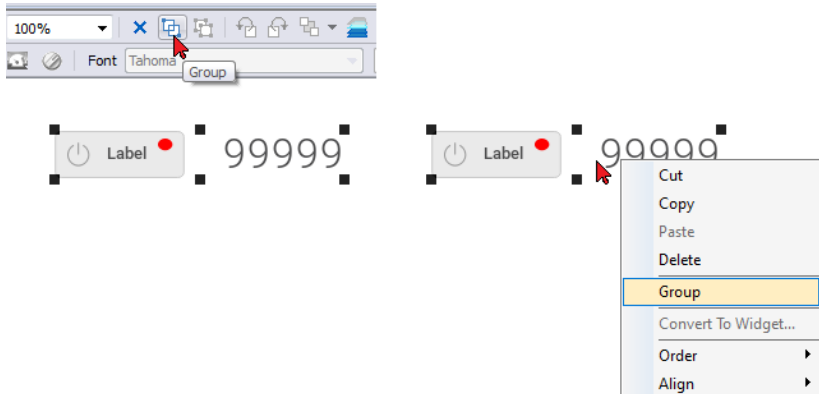
The toolbar allows to:

- hide widgets stacked above and/or below selected widgets
- work on different widgets using the combo box which lists all the widgets in their z-order.

# Grouping widgets

To group widgets:

1. Select all the widgets to group.
2. Right-click and then click **Group**.

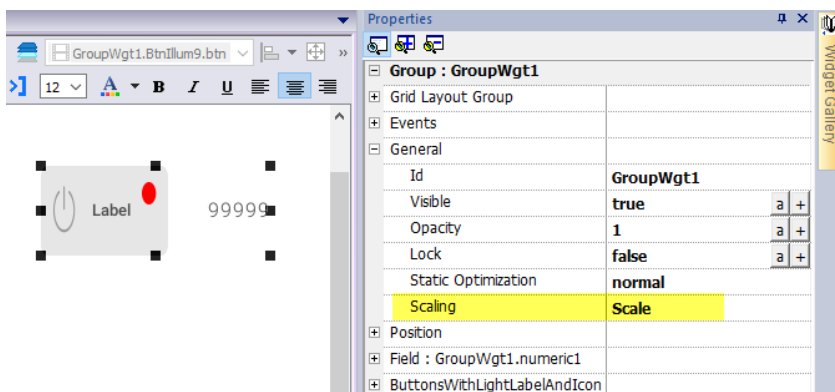


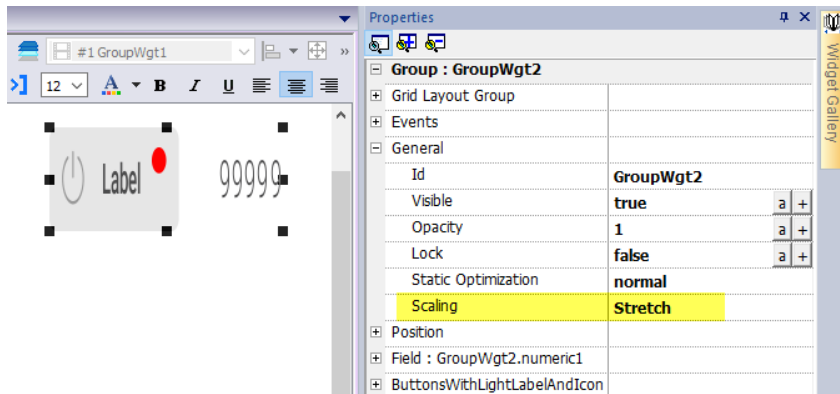
Tip: Double click to enter the group editing mode. In group mode only the group widgets are editable and selectable. All other widgets are partially hidden

## Resizing grouped widgets

You can define how object reacts when re-sized. Use the **Scaling** property in **General** section:

- **Scale**: object and text are not re-sized proportionally
- **Stretch**: object and text are re-sized proportionally



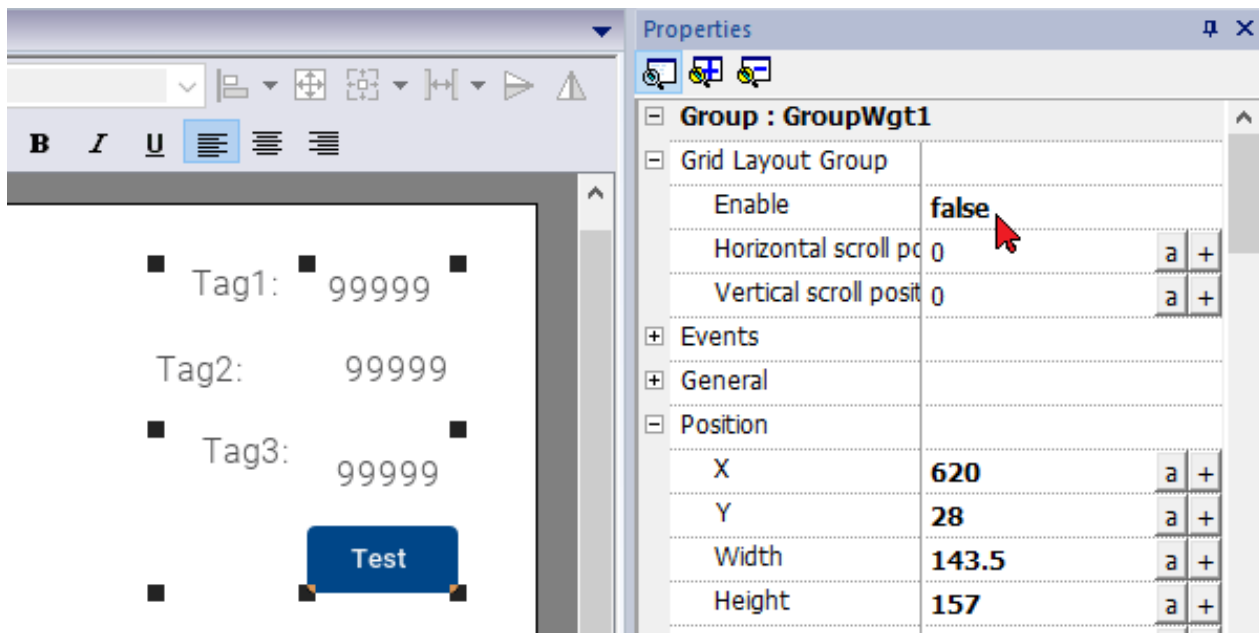


## Grid Layout Group

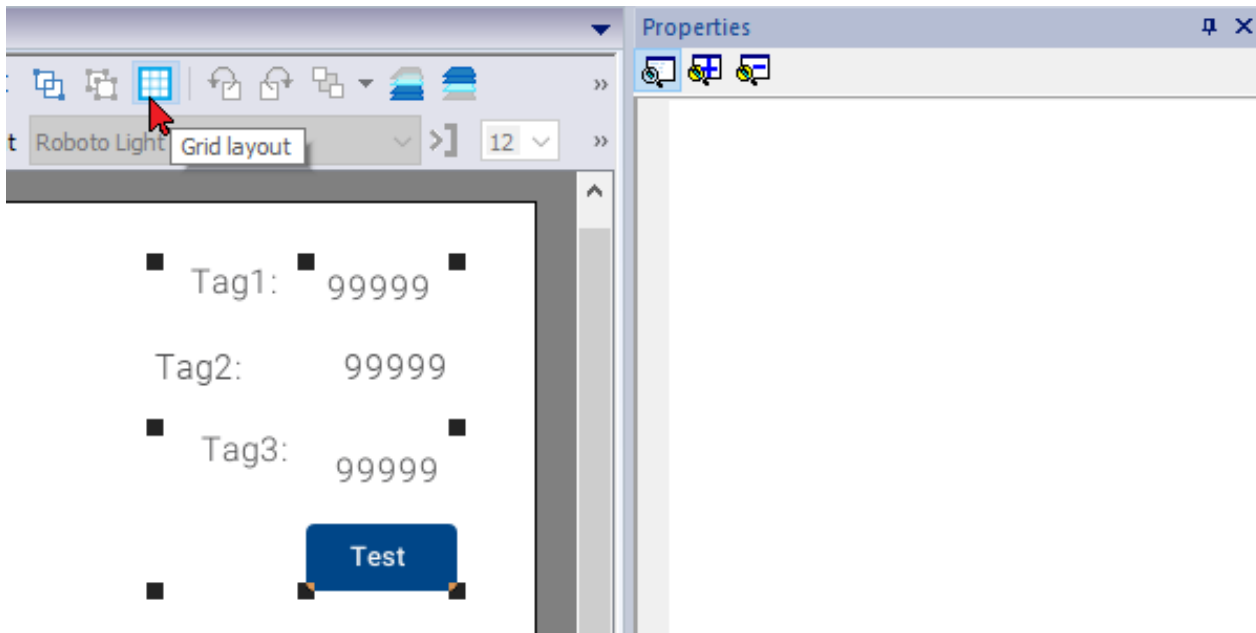
The grid layout add the possibility to configure the spatial relationships among the widgets of the group.

To create a grid layout:

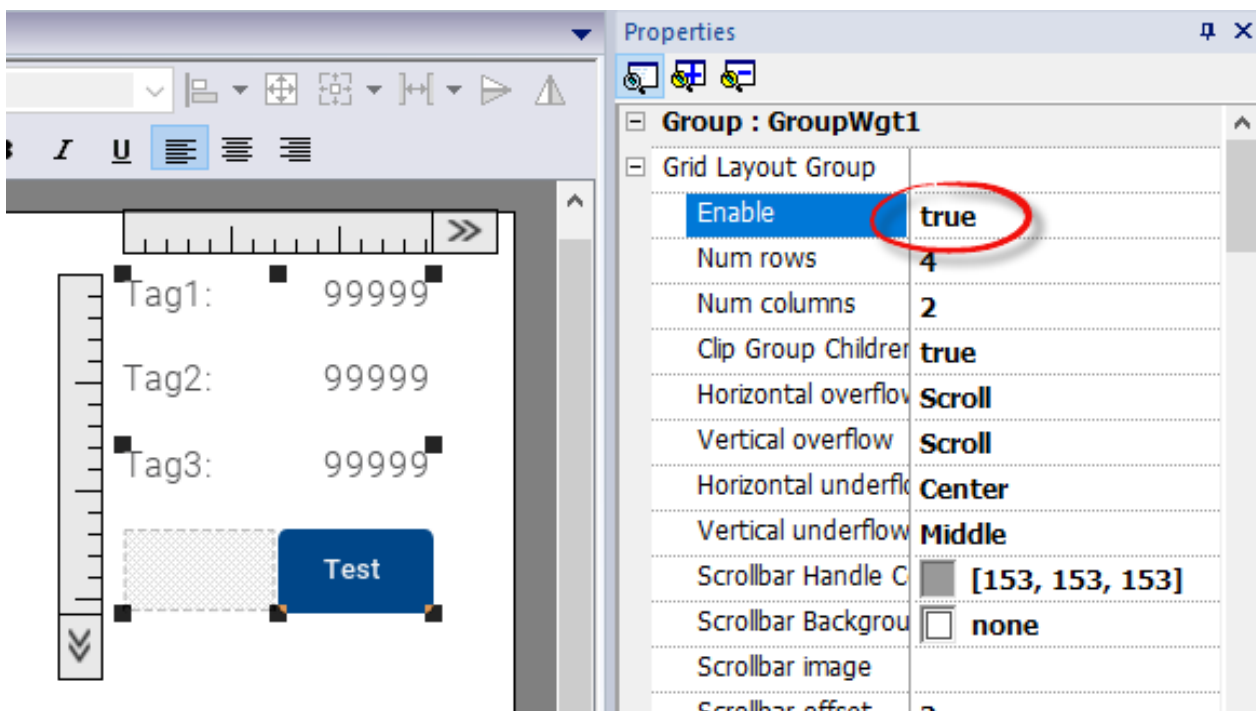
Enable the "Grid Layout" parameter of the group of widgets.



or select the widgets that will be inside the table and click the "Grid Layout" button on page toolbar. This command will create a new group with the grid layout already enabled.



The selected widgets will be aligned and collected inside a group with the grid layout property enabled.



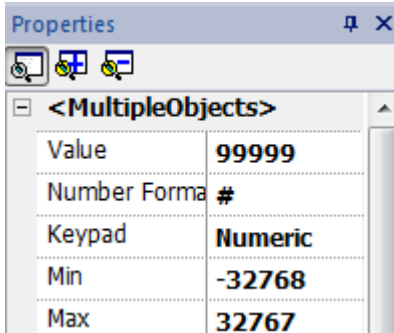
There are several elements associated with the grid layout that can be configured to match your needs, see the "[Grid Layout widget](#)" on page 507 to details.

## Changing multiple widgets properties

You can set the properties of more widgets of the same type all at once.

To change properties:

1. Select widgets.
2. Set common properties from **Properties** pane.
3. When multiple widgets are selected, the Properties pane title changes to **<MultipleObjects>**: all changes will be applied to all selected widgets.



Note: Not all properties can be modified for multiple widgets simultaneously and must therefore be modified individually.

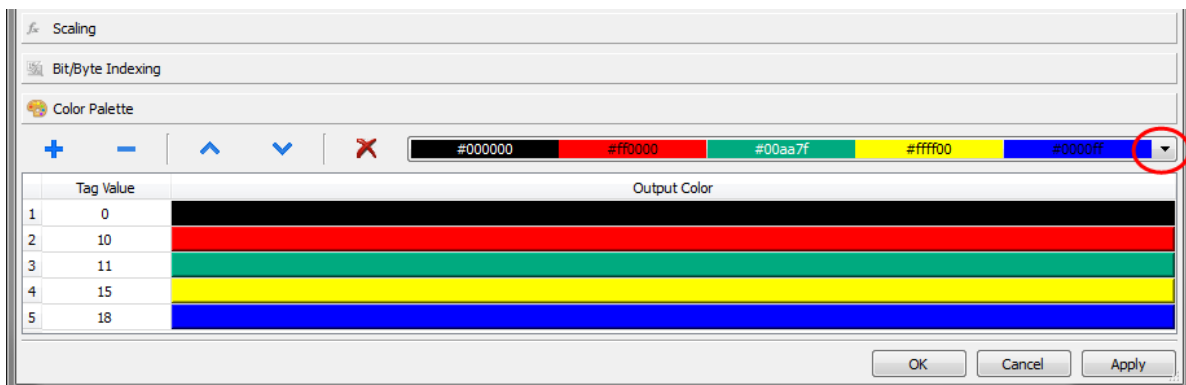
## Changing fill color property according to tag values

PB610 Panel Builder 600 allows to change the color property of a widget dynamically, based on tag values in two ways:

- Using ColorPalette
- Connecting the Color property to a String type tag

### Changing color property using ColorPalette

1. Create the tag (internal or PLC) that you want to refer to for color management. The tag can be of any data type. On the basis of the value of this tag, the color will change.
2. Attach this tag to the **Fill Color** property of an object (for example, a button).
3. In the same dialog select the **ColorPalette** tab and add the colors that will be used for the object according to the tag value.



Note: The last used colors' tables are saved and can be reused selecting them from the colors list box on the toolbar.

## Changing color property connecting Color property to a String type tag

1. Create the tag (internal or PLC) that you want to refer to for color management. On the basis of the value of this tag, the color will change. The tag must be of String type and the **Arraysize** property of the tag must be big enough to contain the string formatted as explained here.
2. Attach this tag to the **Fill Color** property of an object (for example, a button).
3. Write in the **String** tag the RGB color code of the required color. Use one of these formats:
  - **#XXYYZZ**, Where XX, YY and ZZ are the RGB components of the needed color expressed in Hexadecimal format (range 00–FF).
  - **rgb(XXX,YYY,ZZZ)**, where XXX, YYY and ZZZ are the RGB components of the needed colors expressed in Decimal format (range 0–255).



Note: This feature can be applied to all the objects available in the Widget gallery that have a color property. The runtime change of the color is possible only thanks to the properties of the SVGs that are composing the object. This feature can not be applied to other image formats such as JPEG or BMP files.

# 6 Pages

---

This section describes how pages are organized. You can have the same pages shown inside all clients (default mode) or you can customize the pages to better adapt them to each different client.

---

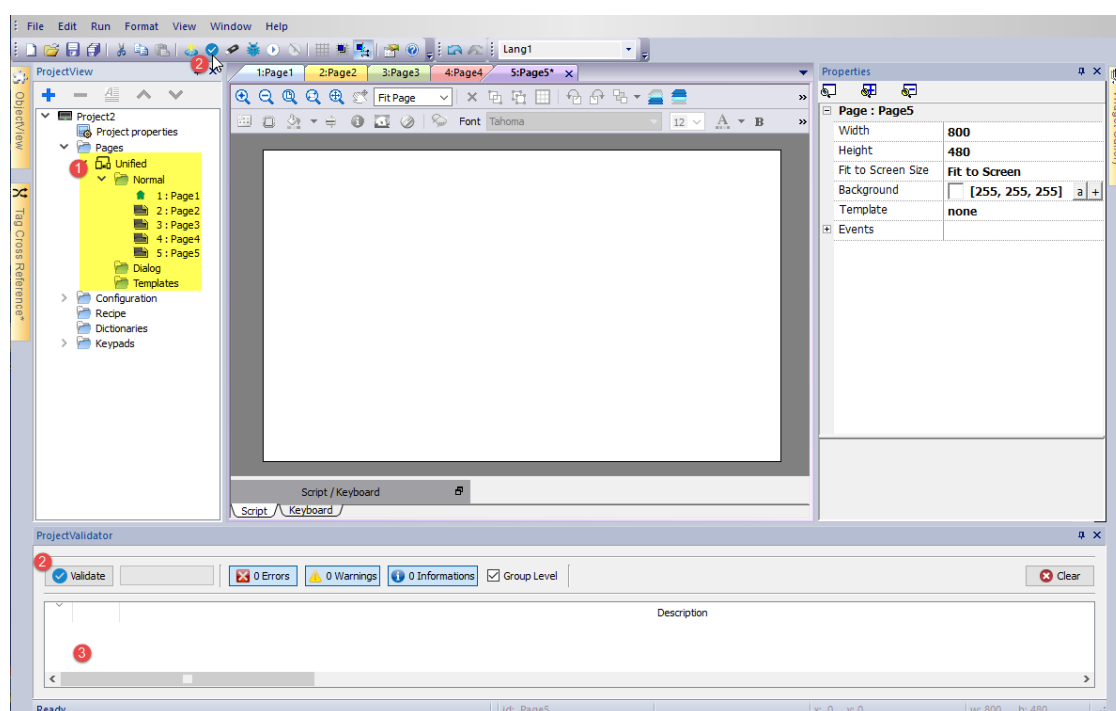
<b>Unified pages</b> .....	<b>70</b>
<b>Project Validator</b> .....	<b>70</b>
<b>Differentiated pages</b> .....	<b>73</b>



## Unified pages

Starting from PB610 Panel Builder 600 v4.0 there is no longer a need to create the pages for HMI device and Web client differently. The same pages can be rendered indifferently on the HMI device or on Web clients. Since some properties or some widgets could be not supported on Web client, some pages could be render differently. The "Project Validator" tool can be used to check if some pages contain widgets that will be rendered differently into Web client.

1. Pages
2. Project Validator button
3. Project Validator output messages

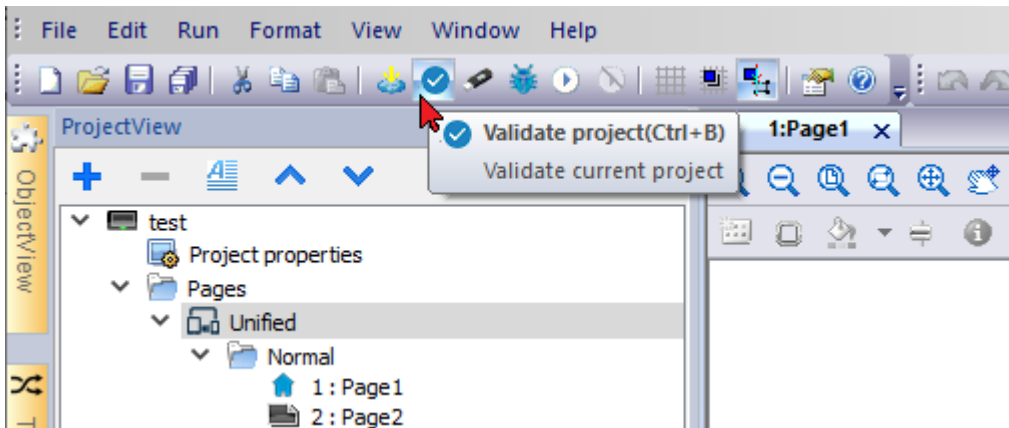


## Project Validator

The "Project Validator" tool check and list the widgets that will be rendered differently into Web client. User can double click each warning message reported from the Project Validator to open the pages that contain the reported widgets to take the appropriate action. However, user action is not mandatory, the project can be downloaded anyway and the unsupported property will not be managed from the Web Client.



**Note that the current version of the Project Validator checks the widget's web compatibility. It is not checking the entire project (e.g. missing tags or Javascript errors)**



### Example

The "Release on disabled" property is not supported on the Web client.

- When the property is set to "False", HMI device and Web client will work in the same way and project validator will not report any message.
- When the property is set to "True", the Project Validator will report the warning message. In the case that the project will download to the HMI device, the Web client simply will not manage the "Release on disabled" property.

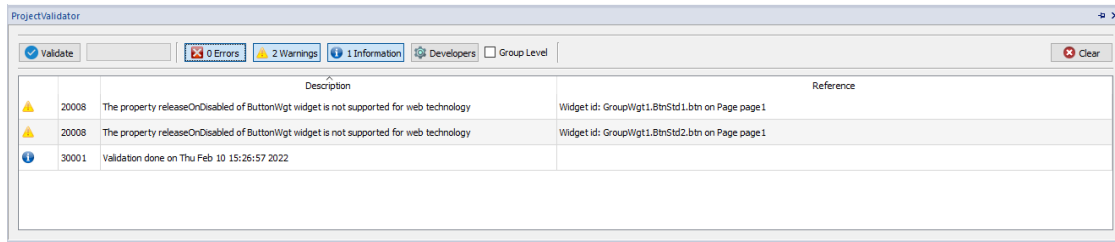
ButtonsWithLabel : BtnStd2	
Value	0
Click Type	momentary
Style	2D
Autorepeat	Disabled
Hold Time (ms)	-1
Label	Label
Up Fill Color	[0, 70, 136]
Down Fill Color	[0, 176, 199]
+ Events	
+ Configure	
+ Text	
- General	
Disable	false
Release on disabled	false
Blink	false
Id	BtnStd2
Visible	true
Opacity	1
Lock	false
Static Optimization	normal
+ Position	

### Group Level

When the "Group Level" is checked, the Project Validator will report the group name that contains one or more widgets with the unsupported properties. A double click will select the grouped widget.

ID	Description	Reference
20010	GroupWgt1 contains widgets with properties not supported for web technology	Widget id: GroupWgt1, on page: page1
30001	Validation done on Thu Feb 10 15:18:47 2022	

When the "Group Level" is not checked, the Project Validator will report the list of the not supported properties. A double click will select the widget that have the unsupported property.



## Differentiated pages

If a project needs to have different pages for the HMI device, web client, tablet client, etc., there is the possibility to add different folder to contain the pages to use on the different clients. Right click on the page folder to add a new category of pages. For each category, you have to define the below properties where Technology, User Agent and Min/Max are filter parameters to define the web clients that belong to the category.

Property	Description						
<b>Name</b>	The category name						
<b>Width, Height</b>	The default size used when create a new page						
<b>Technology</b>	Identify the clients that can use these pages. It can be a combination of: <ul style="list-style-type: none"> <li>Local HMI Device</li> <li>Remote HMI Client</li> <li>Web Clients (PC, Tablet, Smart Phones, etc.)</li> </ul>						
<b>User Agent</b>	It is a regular expression that identifies the web browsers that can display the pages of the category. The user-agent of the web client has to match with this parameter. Example: <table border="0" style="margin-left: 20px;"> <tr> <td style="padding-right: 20px;">.*</td> <td>Anything (all web clients)</td> </tr> <tr> <td>Android</td> <td>Only Android web clients</td> </tr> <tr> <td>Android iPhone</td> <td>Only Android or iPhone web clients</td> </tr> </table>	.*	Anything (all web clients)	Android	Only Android web clients	Android iPhone	Only Android or iPhone web clients
.*	Anything (all web clients)						
Android	Only Android web clients						
Android iPhone	Only Android or iPhone web clients						
<b>Min Width</b> <b>Min Height</b> <b>Max Width</b> <b>Max Height</b>	Defines the size of the display of the Web browser that has to show the pages of this category.  The default, Min=0 and Max=-1, is meaning any size.						



If the definition of a Web client belongs more than one category, are choices the pages that are available inside the closest category.

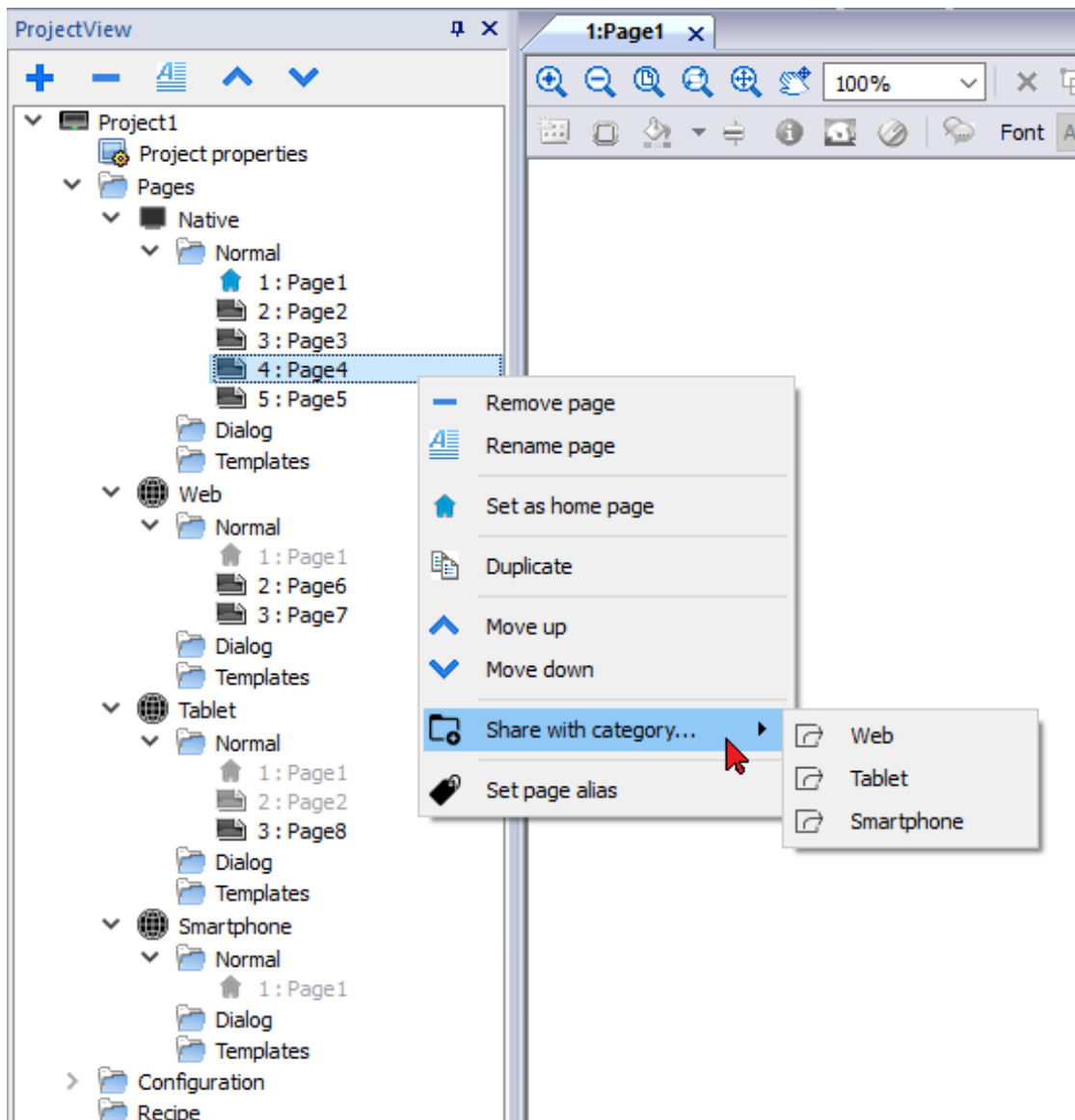
The screenshot displays the ProjectView window on the left and two Properties windows on the right. The ProjectView window shows a hierarchical tree structure for 'Project1' with categories: Native, Web, Tablet, and Smartphone. Each category has a 'Normal' sub-category containing pages (e.g., 1: Page1 to 5: Page5 for Native). The Properties windows show the configuration for 'Page Category : Native' and 'Page Category : Tablet'. The Native category has a width of 1024 and height of 768, with technology 'HMI|RemoteClient'. The Tablet category has a width of 1024 and height of 768, with technology 'Web'. The Tablet category also includes a 'Web Only' section with user agent 'iPad|RIM Tablet OS' and min/max width/height settings of 0 and -1 respectively.

Name	Native
Width	1024
Height	768
Technology	HMI RemoteClient

Name	Tablet
Width	1024
Height	768
Technology	Web
Web Only	
User Agent	iPad RIM Tablet OS
Min Width	0
Min Height	0
Max Width	-1
Max Height	-1

## Shared pages

Pages can be shared between the categories. Shared pages are highlighted in gray color and can be opened indifferently from each category.



### Home Page

From the context menu of the page is possible to define the Home page of the category. The Home Page is the first page that is displayed in the browser type defined in the category and defines the starting point for your web project. The pages you can access from the home page depend on how other pages are linked in the project.

When security is enabled, you can specify a different homepage for each group of users. In this case, this setting will be overwritten. See ["User management and passwords"](#) on page 363 for details.

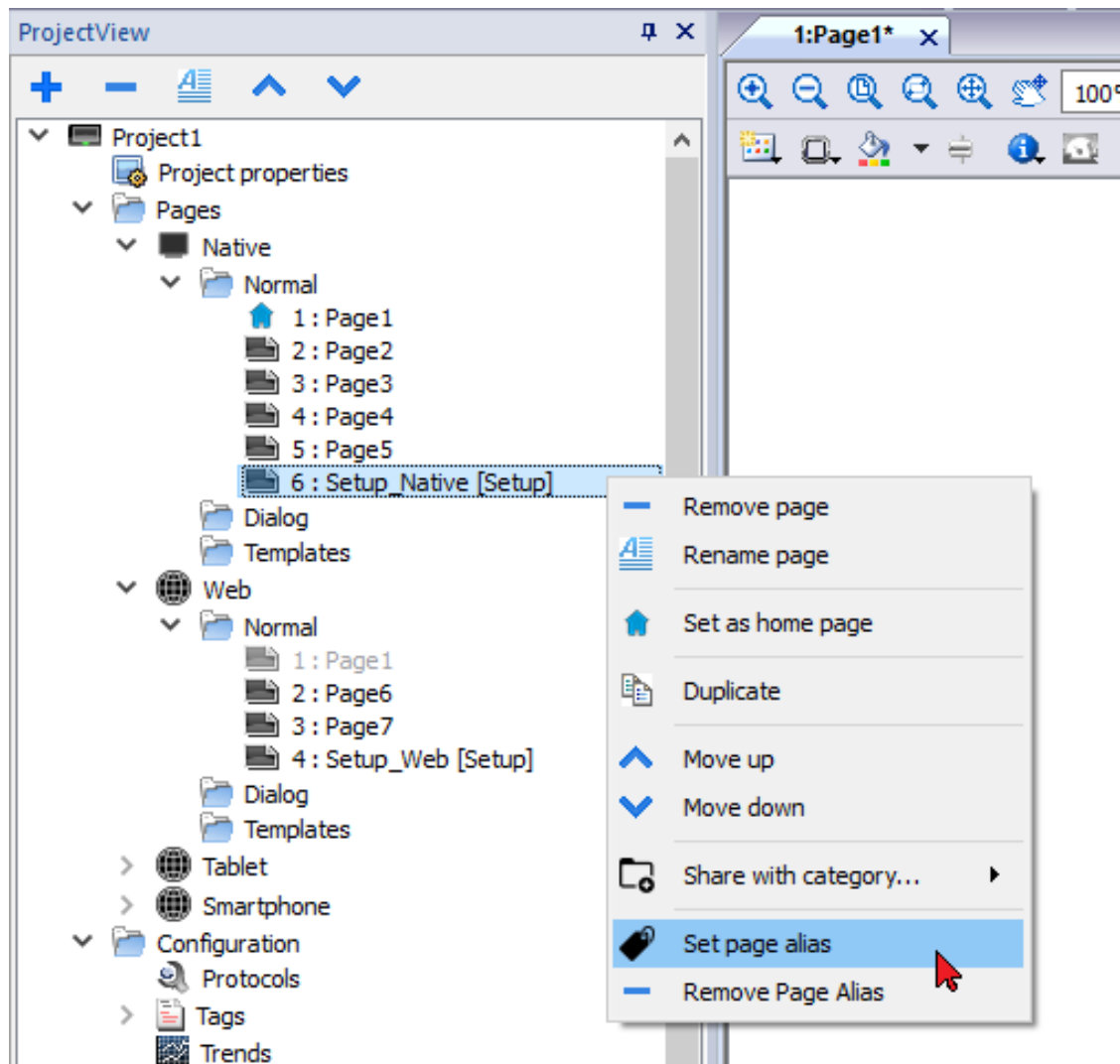
### Alias pages

Using pages shared between categories could be useful the alias page parameter to load the appropriate customized page.

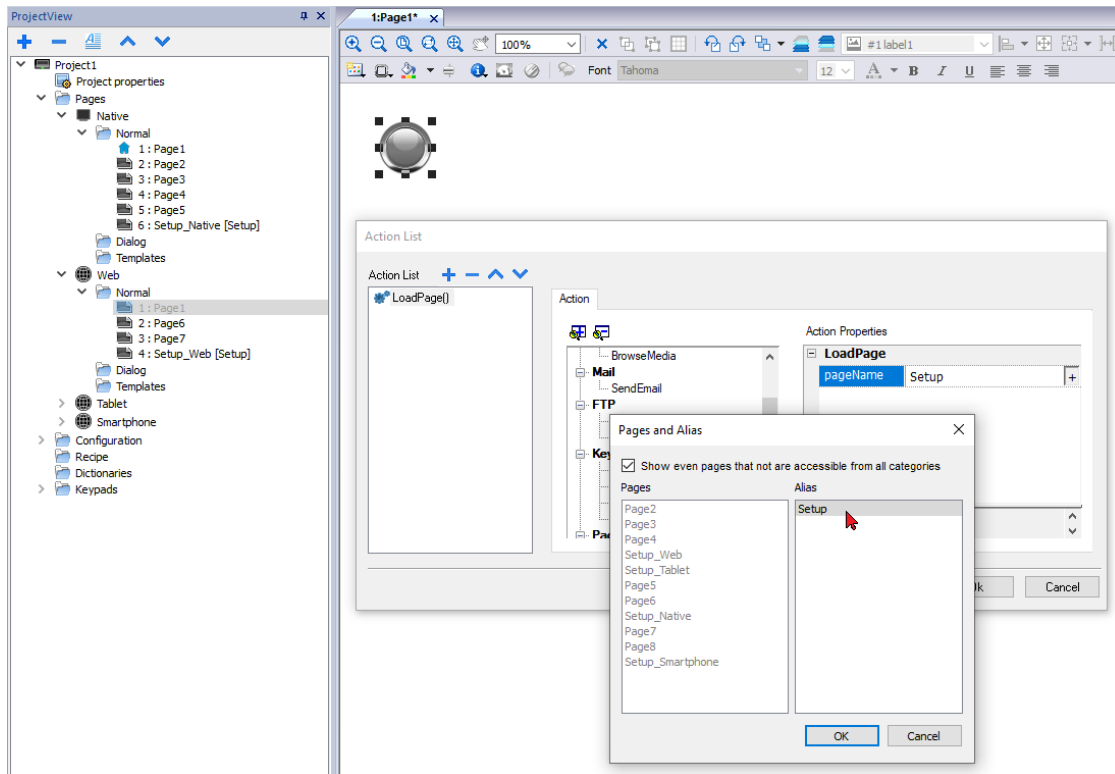
For example, you can have a shared "Page1" common to all categories. Page1 will be shown on both the HMI device and on Web Client, but from this page, you need to add a macro to load a customize setup page. This means a macro that load the page "Setup\_Native" on HMI device or a different page "Setup\_Web" on a web client.

To load a different page depending on the client used, you can add the same alias to both "Setup\_Native" and "Setup\_Web" pages and use the alias name in the LoadPage macro.

Set the alias page:



Use the alias page:





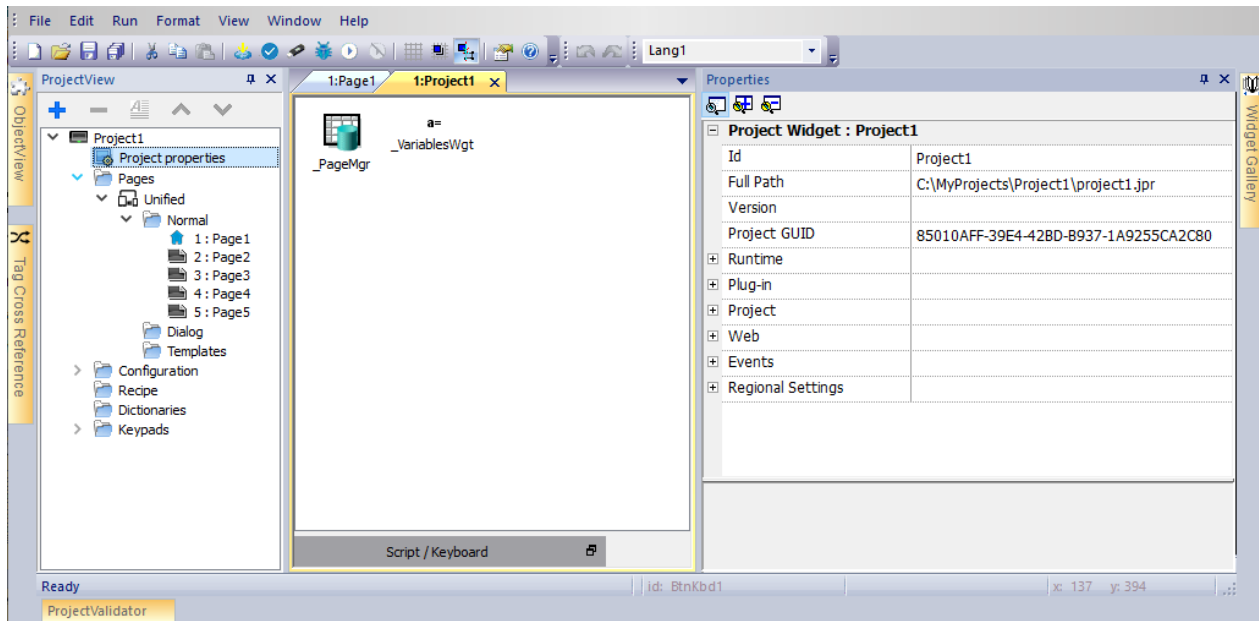


# 7 Project properties

Project properties contain settings for the project.

**Path:** *ProjectView*> double-click *Project properties*> *Properties* pane

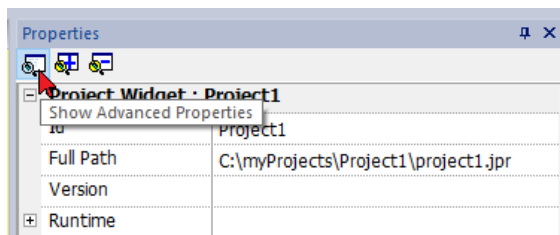
The project **Properties** pane contains a list of project level user-configurable data.



## Basic and advanced properties

Some properties are displayed only in advanced mode. To view all project properties:

- Click **Show Advanced Properties** button to expand the property view in the **Properties** pane.



## Available properties

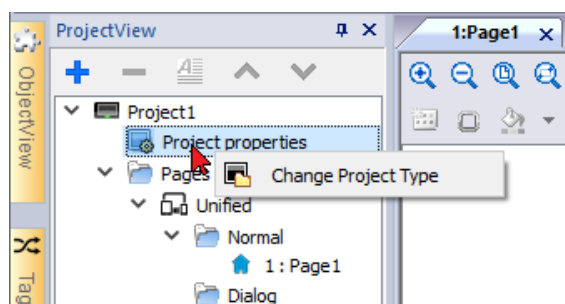
Property	Description
<b>Id</b>	Project name (read only)
<b>Full Path</b>	Project path (read only)
<b>Project GUID</b>	Project unique identifier (read only)

Property	Description
<b>Version</b>	The Version field is available for users to report the project version
<b>+ Runtime</b>	Properties related with the application runtime. See <a href="#">"Runtime" below</a> for details
<b>+ Plug-In</b>	Optional modules. See <a href="#">"Plug-in" on page 86</a> for details
<b>+ Project</b>	Properties related with the project. See <a href="#">"Project" on page 87</a> for details
<b>+ Web</b>	Properties related with the web interface. See <a href="#">"Web" on page 91</a> for details
<b>+ Events</b>	Global events. See <a href="#">"Events" on page 93</a> for details
<b>+ Regional Settings</b>	Definition of date format, list separator, thousand and decimal symbol of number. See <a href="#">"Regional Settings" on page 93</a> for details

Project ID, Project GUI and Project Version are available from system variables. See ["Default variables" on page 162](#) for details.

### Change Project Type






Right-click on project properties to quickly open the dialog to change the project type.





## Runtime

Path: **ProjectView**> double-click **Project properties**> **Properties** pane

Property	Description
<b>Context Menu</b>	<p>Define how context menu should appear in the HMI project.</p> <p><b>on delay</b> = context menu appears touching/pressing and holding for a few seconds an empty area of the runtime screen, or via <b>Context menu</b> action</p> <p><b>on action</b> = context menu appears only via <b>Context menu</b> action.</p> <p>See <a href="#">"ContextMenu" on page 240</a> for details.</p>
<b>Developer Tool</b>	Enable/disables a collection of runtime debugging utility tools.
<b>Buzzer on Touch</b>	<p>Enables buzzer when touching a widget on HMI device screen.</p> <p>Supported widgets:</p> <ul style="list-style-type: none"> <li>• buttons</li> </ul>

Property	Description
	<ul style="list-style-type: none"> <li>• hotspots</li> <li>• needles</li> <li>• fields</li> <li>• external keys</li> <li>• combo boxes</li> <li>• tables items</li> <li>• control list items</li> </ul>  <i>On Windows CE panels, available for from v1.76 ARM / 2.79 MIPS.</i>
<b>Buzzer duration</b>	Default 200 ms
<b>Keyboard</b>	Enables the use of keyboard macros at runtime when using external keyboards.
<b>JavaScript Debug</b>	Enables the JavaScript debugger at runtime for the current project.
<b>Allow JS Remote Debugger</b>	Enables JavaScript remote debugger for current project.  Remote debugging not supported on HMI Client.
<b>Image DB enable</b>	Activates an engine used by the Runtime to optimize project performance.  <b>WARNING: This property should only be disabled by technical support for debugging purposes since this might reduce performance at runtime.</b>
<b>FreeType Font Rendering</b>	Switches to FreeType the font rendering used by PB610 Panel Builder 600 and runtime.  The main reason for using the FreeType is that native WCE engine does not support very well Asian fonts. The second, not less important, reason is that we need the same engine in all devices to avoid different rendering, in particular if static optimization is involved.  <b>Available only on Linux devices (See: "HMI devices capabilities" on page 613)</b>
<b>Communication icon delay (ms)</b>	Delay before display the communication error icon (default is 0 mSec) The special value -1 is meaning always disabled

Property	Description
<b>Fast Boot</b>	<p>When fast boot is enabled and the User Interface is started before the background server</p> <ul style="list-style-type: none"> <li>• Default: User Interface is loaded after the background server is ready to use</li> <li>• Fast UI: User Interface is loaded before loading the background server</li> </ul> <p> <b>Available only on Linux devices (See: "HMI devices capabilities" on page 613)</b></p>
<b>Waiting period for storage devices</b>	<p>Set initial waiting period (in seconds) for storage device in case of fastboot</p> <p> <b>In the case of fastboot, the flash is temporarily mounted as read-only and then remounted as read/write at a later time. If you get an error message stating that the storage device is not working properly, you can configure the system to delay this check. The problem can occur occasionally when fastboot is enabled and the device is overloaded with very intense communication.</b></p>

## Fast Boot

When fast boot is enabled, the HMI device will provide the welcome screen as fast as possible after the power up. In this mode, only the minimum necessary features are loaded before starting the User Interface. Loading of protocols, events, trends, alarms, actions are postponed after loading the User Interface.

There are two flags to set:

- The "Fast Boot" flag available inside the advanced project properties
- The "Fast Boot" flag available inside the Services page of the BSP System Settings tool (see "[System Settings](#)" on page 622)

When fast boot is enabled and the User Interface is started before the background server the JavaScript event `project.onServerReady` can be used to get server synchronization.

Example:

```

if (!project.serverIsReady) {
    // Set the callback to wait for server ready
    project.onServerReady = onServerReady;
} else {
    // Server is ready, call it now
    onServerReady();
}

function onServerReady()
{
    project.setTag("Tag1", 1);
    project.showMessage("Server is ready, tags can be used: " + project.getTag("Tag1"))
}

```



*This is an advance feature available only on Linux platforms*

## Developer tools

Collection of runtime debugging functions that can be enabled or disabled.

1. In **Properties** pane, set **Developer Tools** to **true**.
2. Download the project.
3. Open context menu.
4. Select **Developer tools**.

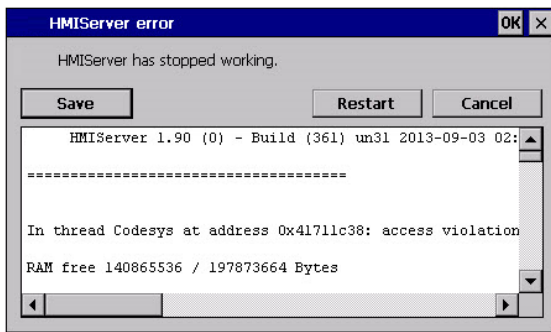
### Developer tools list

Tool	Description
<b>Show/Hide all</b>	Shows a dialog containing information about device status like CPU load, memory usage, event queues.
<b>CPU statistics</b>	Shows information on CPU load. See <a href="#">"CPU Statistics" on the next page</a> .
<b>Memory statistics</b>	Shows information about system RAM . A negative value indicates that free memory is decreasing.
<b>Event queues</b>	Shows information on event queues (size, maximum achieved size, number of processed events, last and maximum processing time). Timing statistics are only available for non-UI queue.
<b>Timelog summary</b>	Show page loading time.
<b>Embed window</b>	Allows embedding in runtime the scene or leave the developer tool window as a standalone window (dialog).
<b>Reset queue stats</b>	Resets statistical information on event queues.
<b>Disable watchdog</b>	Disable the watchdog function and prevents system restart in case of freeze or crash of services.
<b>Ignore exceptions</b>	Disables crash report function, exceptions are not saved in the crash report window.
<b>Launch VNC</b>	Launches the VNC server if available in runtime. VNC server is available as a plugin for Windows CE runtime only.
<b>Profiling</b>	Measures the time spent for loading/rendering the active page. See <a href="#">"Profiling" on the next page</a>

### Watchdog

This feature allows you to disable the watchdog. This way you can avoid system restart in case of a runtime crash and have the time to save the crash report or check system status information (for example, memory available, CPU load, events queue size and so on).

The crash report dialog is displayed automatically in case of a system freeze or crash allowing users to save a log file of crash.



**Important: Save this file for technical support.**

## CPU Statistics

```

2014-04-25 23:02:48, up: 0:08:27, idle: 24 %
Period 2110 ms (overhead 69ms)
  Thread      ID Prio    ms kernel/  user
*          59637774  3    697    0/   697
  Codesys    78839810  0     8     0/     8
Other threads < 5ms
RAM free 125833216 / 194211840 Bytes (diff: 0)
ImageDB size ~2MB, free 44MB / RAMSIZE-76MB)
Page Preload 56MB free / RAMSIZE-64MB)
Page Cache 80MB free / RAMSIZE-40MB)
Storage free 45 / 92 MB

  EvQueue  Size  MaxSize    Evts    ms  max(ms)
  EvMgr    0      0         0      0      0
  ActionMgr 0      1        61    22    189
  AlarmMgr 0      0         0      0      0
  MODR     0      0       122    11    15
  UI       0      11       270    --    --

Timelog is disabled!
(Tap-tap to change position)

```

On the top row the current machine time is shown along with the total device uptime.

CPU statistics are collected with a frequency of 2000 milliseconds. The actual period and the overhead required to collect and visualize statistics are displayed as well. The more the actual period is far from the nominal 2000 milliseconds the higher is the system load. CPU consumption of threads is listed reporting the name of the thread (if available, main thread is marked with a \*), the thread ID, the thread priority and CPU time spent during the 2000 milliseconds period, divided in user and kernel time.

## Profiling

Profiling allows you to check time spent for loading/rendering the active page. Profiling will start from the next page load and will be active only for the first painting of the page to the screen (the configuration is retained).


```

2014-04-25 23:27:19, up: 0:32:58, idle: 36
Period 2053 ms (overhead 47ms)

Page "Alarms.jmx":
START      dT (ms/cpuMs)
Time parsing   : +    6    45/   45
Time unloading : +   54    6/    6
Time lst update : +  195    3/    0
Time gfx creation: +  198  300/  133
      OnLoad :      241/   94
Time rendering : +  535  390/  387
ImageDB cache 15 hit/0 miss(0 ms, cpu: 0 ms)

Page "TemplatePage1.jmx":
Time init/start : +   60  133/   66
Time lst update : +  195    2/    0
Time gfx creation: +  459  27/   27
      OnLoad :     9/    9
ImageDB cache 28 hit/0 miss(0 ms, cpu: 0 ms)

(Tap-tap to change position)
    
```

Profiling option	Description
<b>Enable timelog</b>	Enable timelog capture. Timing will be visible inside the “Timelog summary” window.
<b>Save timelog to file</b>	<p>Saves a report of profile details and the time spent loading a project and its pages into a timelog.txt file. This file can be exported and shared for further analysis.</p> <p> <b>Important: The execution of this function may reduce page change performance.</b></p>
<b>Overlay OnLoad times</b> <b>Overlay Rendering times</b>	<p>This view allows displaying time spent on single widgets and is available only for the rendering and OnLoad steps. The view gives an immediate feeling of where time is spent. Red zones represent the most time critical zones. Detailed widget times are visualized by a tooltip window (on Windows platform attached to mouse over event, on Windows CE press drag and release over the region of interest). In case of out-of-the-scene widgets some arrows allow to navigate to these areas and hovering on them the tooltip will show the area summary</p>
<b>Select overlay color</b>	Select the overlay color to use

**Timelog data**

Data	Description
<b>Time parsing</b>	Time spent parsing current page. Depends on page complexity/number of widgets.
<b>Time gfx creation</b>	Time spent for image rendering. Mainly related to the <i>Onload</i> method.
<b>Time rendering</b>	Time spent rendering the page.
<b>Time unloading</b>	Time spent unloading the page, if current page depends from another page.



Times are provided in couples: wall time/CPU time. Wall time is the absolute time required by this part which can be higher than the actual CPU time required since higher priority threads are also running (for instance protocols). The start time column refers to the page load start time. It can be used to track the actual time required to load a page, since partial times only refer to the most time critical functions and do not include other times that often contribute significantly to the total time.

For example, the actual total wall time required to load a page is rendering (which is the last step) start time + rendering wall time.

## FreeType font rendering

New projects use the FreeType font engine as default. Projects created with older versions of PB610 Panel Builder 600 could use an older font engine also after project conversion to avoid any backward compatibility issue.



Switch to FreeType whenever possible for better page rendering.

Once you have switched to the new font rendering, save the project and verify that all texts are displayed correctly in all project pages.

### Font rendering issues

When switching to the FreeType font engine a project created with the older font engine, you may experience the following problems:

- text requires more/less pixels for rendering thus changing text layout
- widgets are resized to accommodate text
- better rendering can be obtained using antialiasing (antialiasing is a text widget property)

## Plug-in

You can choose which software modules are downloaded to the runtime with the project. Software plug-in has been designed to reduce memory requirements for the HMI application in HMI devices where storage is limited. This option is not supported in Windows platform.

Path: **ProjectView**> double-click **Project properties**> **Properties** pane

Property	Description
<b>Browser</b>	Module required by WebBrowser widget
<b>TextEditor</b>	Module required by TextEditor widget
<b>PDF Reader</b>	PDF Reader (Available only on Windows CE platforms)
<b>VNC Server</b>	VNC Server (Available only on Windows CE platforms)

Once enabled, software plug-in become part of the runtime. Use PB610 Panel Builder 600 to install it using one of the following procedures:

- install Runtime/update Runtime
- update package

To remove plug-ins from runtime use one of the following functions in System Mode:

- format flash
- restore factory settings



**Important:** The system cannot detect automatically which software plug-ins are required by the HMI application, make sure you select them all in the Project Properties.











Note: Software plug-in support has been designed for embedded HMI devices where storage is limited. This option is not supported in Windows platform.



## Project

These properties define various elements of page behavior.

Path: **ProjectView**> double-click **Project properties**> **Properties** pane

Property	Description
<b>Display Mode</b>	Defines HMI device orientation.
<b>Project Type</b>	<p>Defines HMI device type for the project. According to the model, some project features and properties are automatically adjusted.</p> <p> <b>WARNING: Starting from v2, the HMI Runtime will check if the selected project type is matching with the HMI device model and will advise with a message when the selected type is not matching: "HMI Type mismatch. Convert project and download again."</b></p>
<b>Panel Memory</b>	Size of the available internal panel memory.
<b>PageRequest CurrentPage SyncOptions</b>	<p>You can synchronize pages shown on the HMI Runtime and HMI Client from a controller such as a PLC.</p> <p>Attached tag must contain an integer value within the range of the available project pages and must be available at least as a Read resource.</p> <p> See the <a href="#">"Web" on page 91</a> for the Web Browser support</p>
<b>Hold Time Autorepeat Time</b>	<p>Defines the values for hold time and auto repeat time for buttons and external keyboards.</p> <p> Note: These properties can be redefined for each button or key in their widget property table.</p>
<b>Hide Project Loading at boot</b>	<p>When hidden, the splash screen stay on the screen until the application is ready to run.</p> <p> <b>Available only on Linux devices (See: <a href="#">"HMI devices capabilities" on page 613</a>)</b></p>
<b>Target Zoom Factor</b>	It is the zoom factor of the HMI device that will be applied when project is loaded at runtime.

Property	Description						
	<table border="1"> <tr> <td><b>Range</b></td> <td>0.2 - 3.0</td> </tr> <tr> <td><b>Fit to screen</b></td> <td>-1 = Fit to screen size    Fit to screen maintains the aspect ratio. It find the scaling factor, i.e. scale for width and height, then take the smallest.</td> </tr> <tr> <td><b>Default value</b></td> <td>1 = no zoom</td> </tr> </table>	<b>Range</b>	0.2 - 3.0	<b>Fit to screen</b>	-1 = Fit to screen size   Fit to screen maintains the aspect ratio. It find the scaling factor, i.e. scale for width and height, then take the smallest.	<b>Default value</b>	1 = no zoom
<b>Range</b>	0.2 - 3.0						
<b>Fit to screen</b>	-1 = Fit to screen size   Fit to screen maintains the aspect ratio. It find the scaling factor, i.e. scale for width and height, then take the smallest.						
<b>Default value</b>	1 = no zoom						
<b>Background color option</b>	<p>When the defined page is smaller of the entire display area, colorize the area that is not covered from the page (for example when page is Zoom Out)</p> <table border="1"> <tr> <td><b>None</b></td> <td>Old mode, color is white (default)</td> </tr> <tr> <td><b>Selected color</b></td> <td>Color to use</td> </tr> <tr> <td><b>Page background</b></td> <td>Auto adjust color based on background of template or of page</td> </tr> </table>	<b>None</b>	Old mode, color is white (default)	<b>Selected color</b>	Color to use	<b>Page background</b>	Auto adjust color based on background of template or of page
<b>None</b>	Old mode, color is white (default)						
<b>Selected color</b>	Color to use						
<b>Page background</b>	Auto adjust color based on background of template or of page						
<b>Signature</b>	<p>Algorithm to use to signing</p> <ul style="list-style-type: none"> <li>• sha256</li> <li>• sha1</li> </ul>						
<b>Gesture Passthru Enabled</b>	<p>Enable the possibility to pass gesture events to underlying widgets after a configurable delay. User has to keep pressed the finger and then execute the gesture.</p>						
<b>Gesture Passthru Delay (ms)</b>	<p>When enabled, the gesture events are passed to underlying widgets after this delay (see <a href="#">"Gesture events pass thru" on page 474</a> for details)</p>						
<b>Gesture Multitouch</b>	<p>Enable multi touch gestures</p> <ul style="list-style-type: none"> <li>• false</li> <li>• true (default)</li> </ul> <p> This property give the possibility to disable the multi touch gestures. This could be useful to avoid problems with old projects that were not designed to manage the multi touch gestures.</p>						
<b>On Access Denied</b>	<p>When user try to use a widget that is locked from the security configuration to read-only (e.g. a field or a button), a padlock icon is shown for a couple of seconds to highlight that the widget is not accessible.</p> <ul style="list-style-type: none"> <li>• None</li> <li>• Show Icon</li> </ul>						
<b>ComboBox View Mode</b>	<p>Select the visualization mode of all the Combo Box widgets of the project (see <a href="#">"Combo Box widget "full screen" mode with images" on page 459</a> for details)</p> <ul style="list-style-type: none"> <li>• Context Classic view with drop-down menus</li> <li>• Full screen</li> </ul>						

Property	Description
	Enhanced view with configurable texts and images that will pop up in the middle of the screen for easy scroll and selection.
<b>Encrypted Project</b>	Encrypt or decrypt the project to protect intellectual property and not be readable or editable by unauthorized users (see <a href="#">"Project Files Encryption" on page 599</a> for details)
<b>Sign Project</b>	If true, the project will be signed before being downloaded to the HMI device.
<b>Certificate</b>	Select, from the list of certificates you have installed on your PC, the certificate to use to sign the project. Be sure to install the same certificate (the public key) on the HMI device (see <a href="#">"Project Signature" on page 602</a> for details).  This parameter is available only when <i>"Sign Project"</i> = true
<b>Show Messages</b>	Avoid popup errors or warning messages. Messages will be logged to /var/log/popup_messages.log (max size of the log file is 256Kb)   <b>Available only on Linux devices (See: <a href="#">"HMI devices capabilities" on page 613</a>)</b>
<b>Enable CSRF Token</b>	Enable CSRF Token for web security <ul style="list-style-type: none"> <li>• When CSRF Token is disabled, we can pass cgi multiple commands from same browser session.</li> <li>• When CSRF Token is enabled, then we can not pass multiple cgi requests even on same browser session.</li> </ul>  <b>For security reasons, this flag should always be set to true. For backward compatibility, projects developed with versions prior to 4.05 will be converted with this flag set to false.</b>

## PageRequest, CurrentPage and SyncOptions

It is possible to have HMI Runtime exchange devices information on the page shown by the HMI. You can synchronize pages shown on the HMI device and on HMI Client or to control an HMI project from a controller such as a PLC.

The following properties can be customized:

Property	Description
<b>PageRequest</b>	Page to be shown on the HMI device and on HMI Client. Attached tag must contain an integer value within the range of the available project pages and must be available at least as a Read resource.
<b>CurrentPage</b>	Page number displayed on the HMI device or on HMI Client or on both. Attached tag must be available at least as a Write resource and must have integer data type.
<b>SyncOptions</b>	Synchronization of project pages with the value contained into the <b>CurrentPage</b> property. Options can be: <ul style="list-style-type: none"> <li>• <b>disable</b>: page number value is ignored,</li> </ul>

Property	Description
	<ul style="list-style-type: none"> <li>• <b>local</b>: page number displayed on HMI,</li> <li>• <b>remote</b> : page number displayed on HMI Client.</li> <li>• <b>local + remote</b>: page number displayed on HMI and on HMI Client, if different pages are displayed the last page loaded is considered.</li> </ul>



- The PageRequest is opened at the project startup.
- When the user login, or switch user, the user's home page or the user's last page has priority over the PageRequest

### Example: forced page change from controller/PLC to HMI device and HMI Client

Set properties as follows:

<b>PageRequest</b>	attached to tag "A"
<b>CurrentPage</b>	empty
<b>SyncOptions</b>	disable

Set value of tag "A" to display the requested page on HMI device and HMI Client.

### Example: forced page change from controller/PLC to HMI and HMI Client. Read current page loaded on HMI

Set properties as follows:

<b>PageRequest</b>	attached to tag "A"
<b>CurrentPage</b>	attached to a tag "B" as read/write
<b>SyncOptions</b>	local

Set value of tag "A" to display the requested page on HMI device and HMI Client. Tag "B" will contain the number of page currently shown by the device.

### Example: forced page change from controller/PLC to HMI device and HMI Client. Read current page loaded on HMI Client.

Set properties as follows:

<b>PageRequest</b>	attached to tag "A"
<b>CurrentPage</b>	attached to a tag "B" as read/write
<b>SyncOptions</b>	remote

Set value of tag "A" to display the requested page on HMI and HMI Client. Tag "B" will contain the number of page currently shown by HMI Client.

**Example: forced page change from controller/PLC to HMI device and HMI Client. Force HMI Client page synchronization with HMI device (not vice versa).**

Set properties as follows:

<b>PageRequest</b>	attached to a tag "A" as Read/Write
<b>CurrentPage</b>	attached to the same tag "A" as per <b>PageRequest</b>
<b>SyncOptions</b>	local

Set value of tag "A" to display the requested page on HMI and HMI Client. Change page on HMI to display the same page on HMI Client.

**Example: forced page change from controller/PLC to HMI device and HMI Client. Force HMI page synchronization with HMI Client (not vice-versa).**

Set properties as follows:

<b>PageRequest</b>	attached to a tag "A" as read/write
<b>CurrentPage</b>	attached to the same tag "A" as per <b>PageRequest</b>
<b>SyncOptions</b>	remote

Change value of tag "A" to display the requested page on HMI and HMI Client. Change page on HMI Client to display the same page on HMI.

**Example: synchronize displayed page between HMI device and on HMI Client**

Set properties as follows:

<b>PageRequest</b>	attached to a tag "A" as read/write
<b>CurrentPage</b>	attached to the same tag "A" as per <b>PageRequest</b>
<b>SyncOptions</b>	local+remote

Changing page on HMI device, same page will be shown on HMI Client and vice-versa.

## Web

Path: **ProjectView**> double-click **Project properties**> **Properties** pane

Property	Description
<b>Show Runtime Errors</b>	<b>true</b>   Display a message when a runtime error occurs
	<b>false</b>   Disabled
<b>Web Inactivity Timeout</b>	Defines a timeout for PB4Web client. When the timeout expires without any activity the current user is logged out.

Property	Description	
	<b>Range</b>	1–86400 s (form 1 s to 24 h)
	<b>Default value</b>	600 s
	<b>Values</b>	0 = disabled
<b>Web Icon</b>	The favorite icon associate at the web pages	
<b>Refresh Time</b>	Defines the refresh time for the communication between the runtime and PB4Web clients.	
	<b>Range</b>	50–10000 ms
	<b>Default value</b>	100 ms
<b>Use browser render API</b>	<b>true</b>	Use the browser API requestAnimationFrame to manage graphic update
	<b>false</b>	Disable this properties for old browsers that not support the web engine optimization
<b>Force browser layers</b>	<b>true</b>	Force browser to use graphic layers when widgets are attached to tags
	<b>false</b>	Disable this properties for old browsers that not support the web engine optimization
<b>Enable Global JavaScript for remote</b>	Define if the JavaScript code defined inside the Project Properties, general triggered from Alarms and Schedulers events, have to run only on local HMI device or even on remote clients.	
	<b>None</b>	Will not be executed on remote clients (run only inside the local HMI device)
	<b>Client</b>	Will be executed on HMI Client
	<b>Web</b>	Will be executed on Web client
	<b>Both</b>	Will be executed on both HMI Client and Web clients
<b>Max Bandwidth (Kbs)</b>	Limit for maximum data sent by server (useful for old slow browsers). Set to 0 to use all the available bandwidth (default)	
<b>Web clients connection mode</b>	<b>Auto</b>	The connection mode is selected by the client (default)
	<b>SSE</b>	Force the Server-Sent Events mode
	<b>Long Polling</b>	Force the Long-polling mode
<b>WebPageRequest</b>	You can synchronize pages shown on the PB4Web Clients from a controller such	


Property	Description
	as a PLC. Page to be shown on the PB4Web Client. Attached tag must contain an integer value within the range of the available project pages and must be available at least as a Read resource.
<b>Web Communication icon delay (ms)</b>	Delay before display the communication error icon (default is 0 mSec) The special value -1 is meaning always disabled
<b>Enable the change page loader animation</b>	If set to "true", an animated icon is displayed while the page is loading.
<b>Enable change page animation for cached pages</b>	If set to "false", the animated icon is not displayed when loading pages that are in the cache. Generally, the loading of these pages is very fast.



The `project.getClientType()` can be used to retrieve the running client type. See "[Project object](#)" on page 557 inside JavaScript chapter for additional details.

## Events

Path: **ProjectView**> double-click **Project properties**> **Properties** pane

Property	Description
<b>OnWheel</b>	Used only in conjunction with wheel input devices. Normally the wheel is used to increase/decrease the value of a tag without an external keyboard device. Attach this property to a change of wheel event and use an action like <b>BiStep</b> to increase/decrease a tag value.  <b>The project's OnWheel Action is executed only when the OnWheel Action will not overwritten from the loaded page.</b>

## Regional Settings

Path: **ProjectView**> double-click **Project properties**> **Properties** pane

Property	Description
<b>Short date format</b>	The date format to use when user select SHORT-DATE in the date format of the widget
<b>Long date format</b>	The date format to use when user select LONG-DATE in the date format of the widget
<b>List separator</b>	List separator character to use inside the dumped files.



---

Property	Description
<b>Decimal symbol</b>	Character to use in numeric widgets to separate the integer part from the fractional part (it is visible only when user configure the widget to show the fractional part)
<b>Thousand symbol</b>	Character to use in numeric widgets to separate the thousands (it is visible only when user configure the widget to show the thousand character)



You can use placeholders to freely define the Time and Date format (see "[Time and Date placeholders](#)" on [page 472](#))

# 8 The HMI simulator

---

HMI simulator allows you testing projects before downloading it to the HMI device. It may be used to test the project when no HMI device is available and to speed up development and debugging activities.


On the HMI simulator, you can choose to use the real protocols to update the values of the tags (supported only for Ethernet or serial protocols) or simulate the communication and update the values of the tags manually.

---

<b>Simulator settings</b> .....	<b>96</b>
<b>Tags Simulation</b> .....	<b>96</b>
<b>Data simulation methods</b> .....	<b>99</b>
<b>Launching and stopping the simulator</b> .....	<b>99</b>

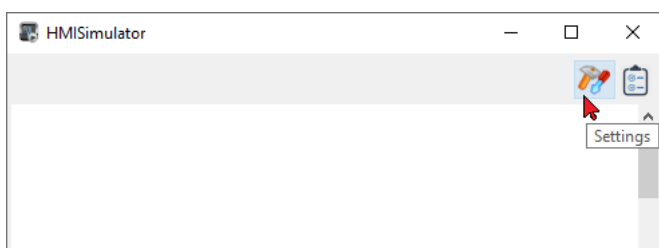
# Simulator settings

The Simulator works by default with simulated protocols. It can also work with real protocols (Ethernet or serial protocols)

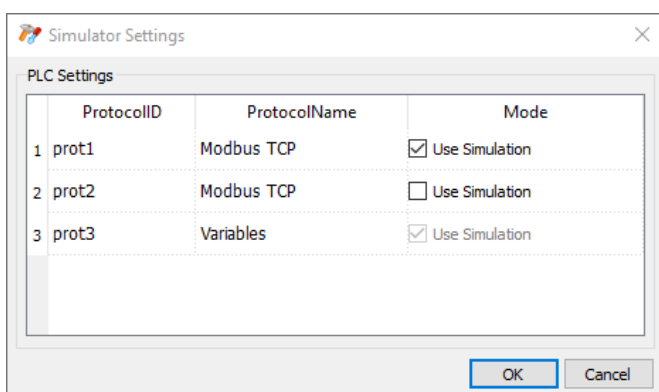
 Note: For protocols not supporting communication with external devices, such as the Variables protocol, this option is always disabled.

## Changing simulated protocols

1. Click the simulator **Settings** icon.



2. Select **Use Simulation** to use simulated protocols, otherwise real protocols will be used for communication with external devices.



## Tags Simulation

Through the "Tag Simulation" dialog it is possible to interact with the tag values in order to simulate the operation of the application.

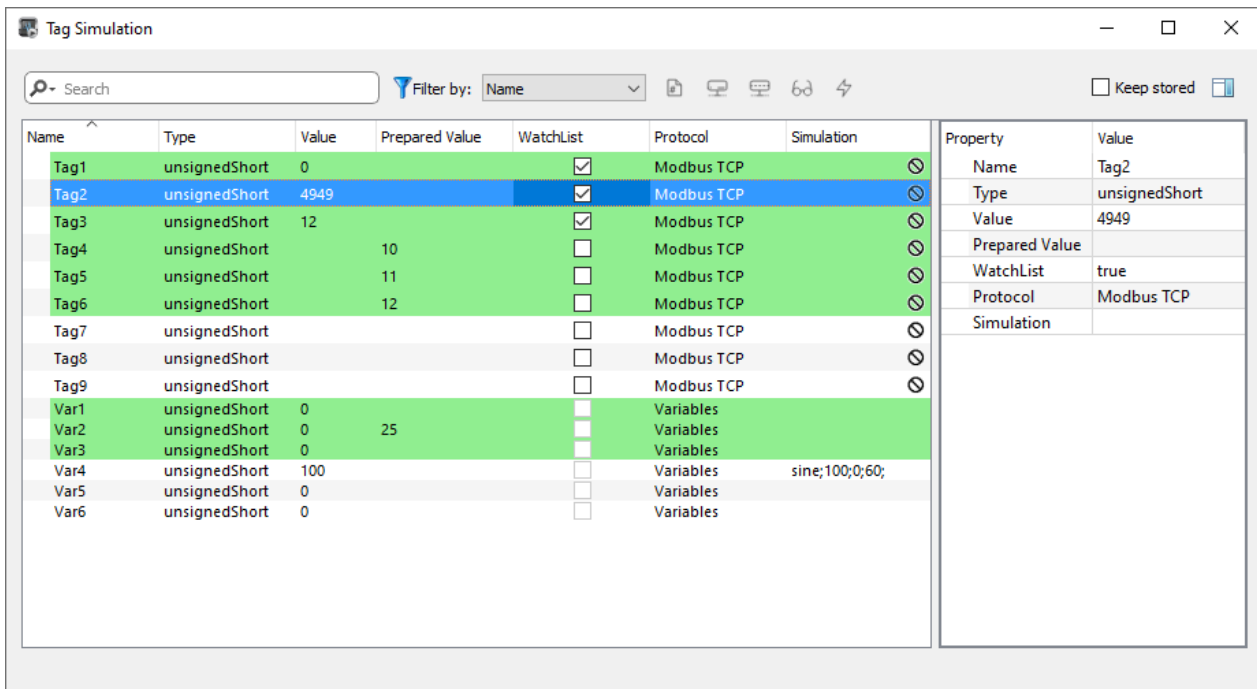


**Available only on Linux devices (See: ["HMI devices capabilities" on page 613](#))**

To activate the "Tag Simulation" dialog click the simulator **Tag Simulation** icon:



The "Tag Simulation" dialog will be shown



**Colors:**

- The rows with green backgrounds highlight the tags that are used in the active page.
- The rows with blue backgrounds highlight the selected tags.

The property panel on the right is showing the properties of the selected tag. Note that you can select multiple tags and modify the same property on all selected tags.

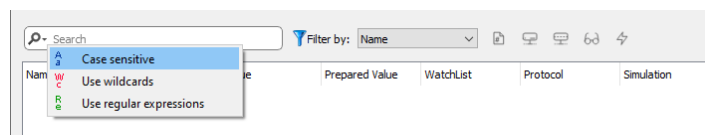
**Columns available inside the "Tag Simulation" dialog**

Column	Description
<b>Name</b>	Tag's name
<b>Type</b>	Tag's type
<b>Value</b>	Tag's value. It is shown only when the tag is related to a simulated protocol or when the "watch list" parameter is checked. If the tag is related to a simulated protocol a double click gives you the possibility to modify the value.
<b>Prepared Value</b>	Tag values that are not simulated but are related to a working protocol cannot be written directly. You can prepare the tag values you want to modify within the "Prepared values"

Column	Description
	column and then press the "Write prepared values" button on the toolbar to write all the prepared values at the same time.
<b>Watch List</b>	Select the tags you want to be read to be able to see their values in the "Value" column. This function is not available when a protocol is in simulation mode.
<b>Protocol</b>	The protocol associated with the tag
<b>Simulation</b>	Triggers an automatic change of the tag value. See the available functions on: " <a href="#">Data simulation methods</a> " on the facing page. This feature is available only on simulated protocols.  When you exit and restart the simulator this field will be reset with the values defined within the project. If you prefer to keep the values entered with this field, select the "Keep Archived" on the toolbar.





## Filters available on the toolbar

In the main filter, available on the toolbar, you can enter the filter text, select the column in which to search, and the search mode.



Main options	Function
<b>Wildcards</b>	Search using simple wildcards matching . Character '?': matches any single character. Character '*': matches zero or more of any characters. "[...]": sets of characters can be represented in square brackets.
<b>Regular Expression</b>	Describes character pattern.  See <a href="https://en.wikipedia.org/wiki/Regular_expression">https://en.wikipedia.org/wiki/Regular_expression</a> for additional details regarding regular expressions.

Using the buttons on the right you can activate the following filters:

-  Show only current page tags
-  Hide online tags (the tags managed from a physical protocol)
-  Hide simulation tags (the tags where the protocol is simulated)
-  Hide tags not in watch list

The last button will write the "Prepared Value" inside each selected tag, then the prepared values are cleared. Note this write will be executed on all selected tags even they are not on the Watchlist.



Write prepared values

## Data simulation methods

Set tag simulation behavior in the **Simulator** field of Tag Editor.

Method	Description
<b>Variables</b>	Data is stored in a simulator variable. This variable holds the value of the tag so you can read and write the value.
<b>SawTooth</b>	A count value is incremented from <b>Offset</b> to <b>Amplitude + Offset</b> value with a <b>Period</b> of 60..3600 seconds. When the counter reaches <b>Amplitude + Offset</b> , the value is reset to <b>Offset</b> and the counter restarts.
<b>Sine Wave</b>	A sine wave value is generated and written to the tag value. <b>Min</b> , <b>Max</b> and <b>Period</b> values can be defined for each tag.
<b>Triangle Wave</b>	A triangle wave value is generated and written to the tag value. <b>Min</b> , <b>Max</b> and <b>Period</b> values can be defined for each tag.
<b>Square Wave</b>	A square wave value is generated and written to the tag value. <b>Min</b> , <b>Max</b> and <b>Period</b> values can be defined for each tag.

See ["Adding tags" on page 115](#) for details.

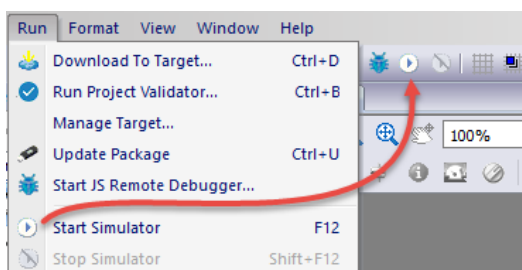
### Keep Stored

When the "Keep Stored" flag on the toolbar is set to true, the configured simulations will be kept even after closing and reopening the simulator.

## Launching and stopping the simulator

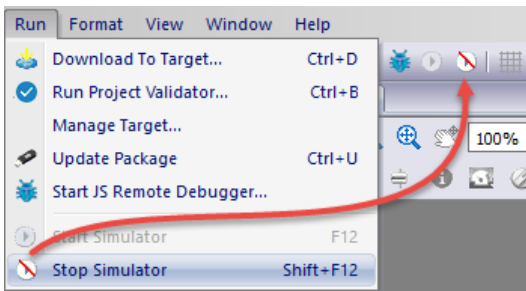
To launch the simulator:

1. On the **Run** menu, click **Start Simulator**: the Simulator runs on the computer in the same way as the server would run on the HMI device.



To stop the simulator:

1. On the **Run** menu, click **Stop Simulator** or on the simulated page double-click the **Exit** button.



# 9 Transferring the project to HMI device

---

To transfer the PB610 Panel Builder 600 project to the target HMI device you can use:

- function **Run > Download to Target**
- function **Run > Update Package** with the use of a USB device

---


<b>Download to HMI device</b> .....	<b>102</b>
<b>Update package</b> .....	<b>104</b>
<b>The Runtime loader</b> .....	<b>107</b>
<b>Upload projects</b> .....	<b>108</b>
<b>Runtime dynamic files handling</b> .....	<b>109</b>
<b>Userdata</b> .....	<b>110</b>



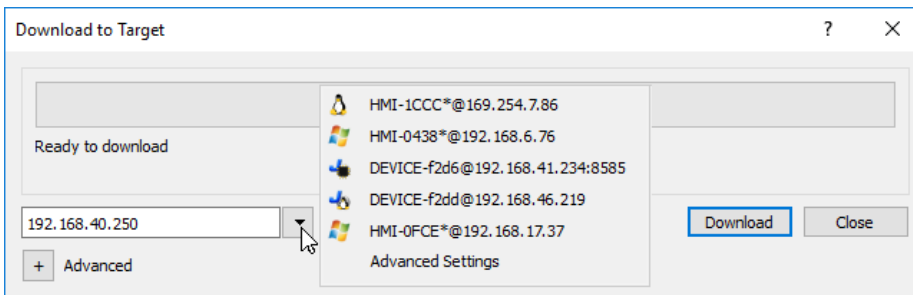
# Download to HMI device

## Path: Run > Download to Target

This function transfers project and HMI Runtime via Ethernet .

 Note: The HMI device must have a valid IP address. See ["HMI device basic settings"](#) on page 10 for details on how to assign an IP address.

1. Click the discovery button: a list of the detected IP addresses is displayed.
2. Select the HMI device IP address.

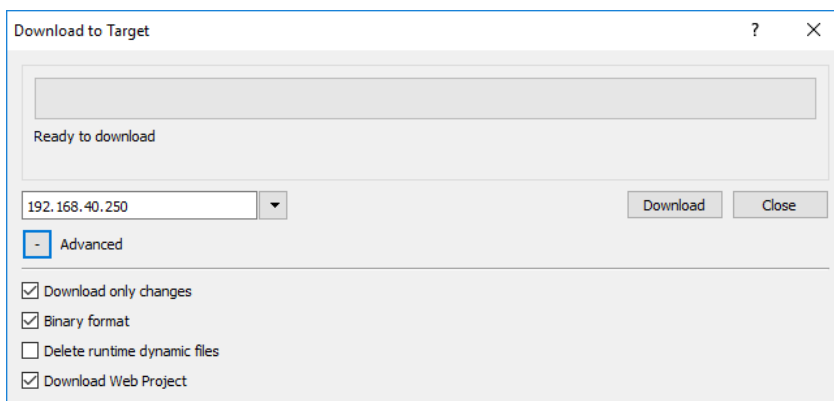


You can even enter the IP address manually or, if available, the host name provided by a DNS server. Using a service tool like Bonjour, Linux-based HMI devices can be discovered using their hostname (e.g HMI-0d37.local). Bonjour is a trademark of Apple inc.



3. Click **Download**: PB610 Panel Builder 600 will switch the HMI device to Configuration Mode and transfer the files.

When the download operation is completed, the HMI device automatically switched back to Operation Mode and the project is started.

## Advanced options



Option	Description
<b>Download only changes</b>	Transfers to the HMI device only the modified project files.
<b>Binary format</b>	Download files using binary format.

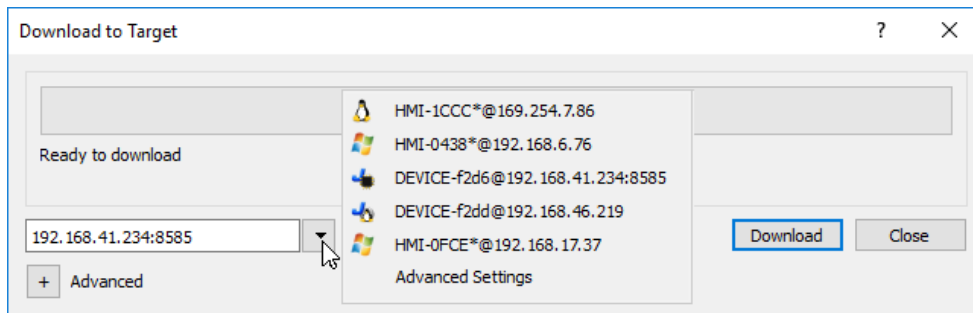
Option	Description
<b>Delete runtime dynamic files</b>	<p>Modified configuration of recipes, users, schedulers, etc. done at runtime will be deleted and overwritten by the configuration defined in the project.</p> <p> <b>CAUTION: This operation cannot be undone, deleted dynamic files cannot be restored.</b></p> <p> <b>CAUTION: Dynamic files are not deleted if stored on external devices (USB or SD Cards).</b></p>
<b>Download Web Project</b>	<p>Download the PB4Web pages to HMI device.</p>

When transferring a project, PB610 Panel Builder 600 uses a combination of HTTP and FTP connections:

- HTTP connection - issues the commands to switch to transfer mode or to unload running project,
- FTP session - transfers the files to the flash memory in the HMI device.

### Advanced Settings

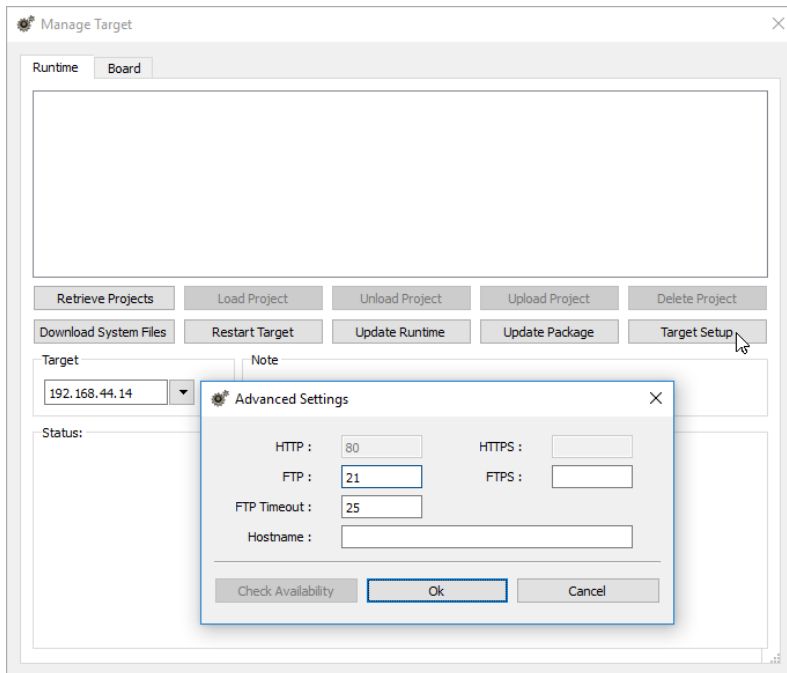
Using the “Advanced Settings” option, you can define the ports to use, but generally, you do not need to enter this information because HMI devices will provide the ports to use inside the panel list.



## Changing HMI device connection settings

**Path:** Run > Manage Target

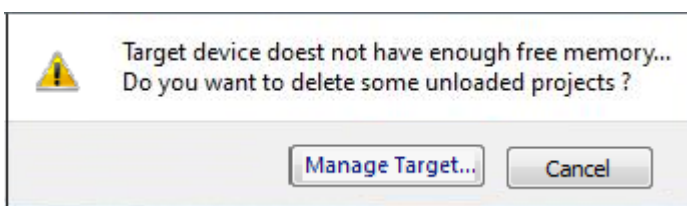
1. Click **Target Setup**: the **Advanced Settings** dialog is displayed. Default port for HTTP connections on the HMI device is port 80.



2. Set correct HTTP, FTP or HTTPS, FTPS ports for the HMI device. (These are the ports used by the system to connect to the HMI device and may need to be modified when default ports are used by other services or applications or if the local network requires specific settings.)
3. Specify **Hostname** to easily identify each device in a network where multiple devices are available. The default hostname is "HMI" for all devices.
4. Click **Download System Files**. At the next download the new ports will be used in the HMI device and new hostname will appear in the drop-down list

## Managing big projects

For successful download the project size should be at least 2 MB smaller than the available memory. If not, you run out of flash memory in the HMI device and a warning message is displayed.



To free more memory:

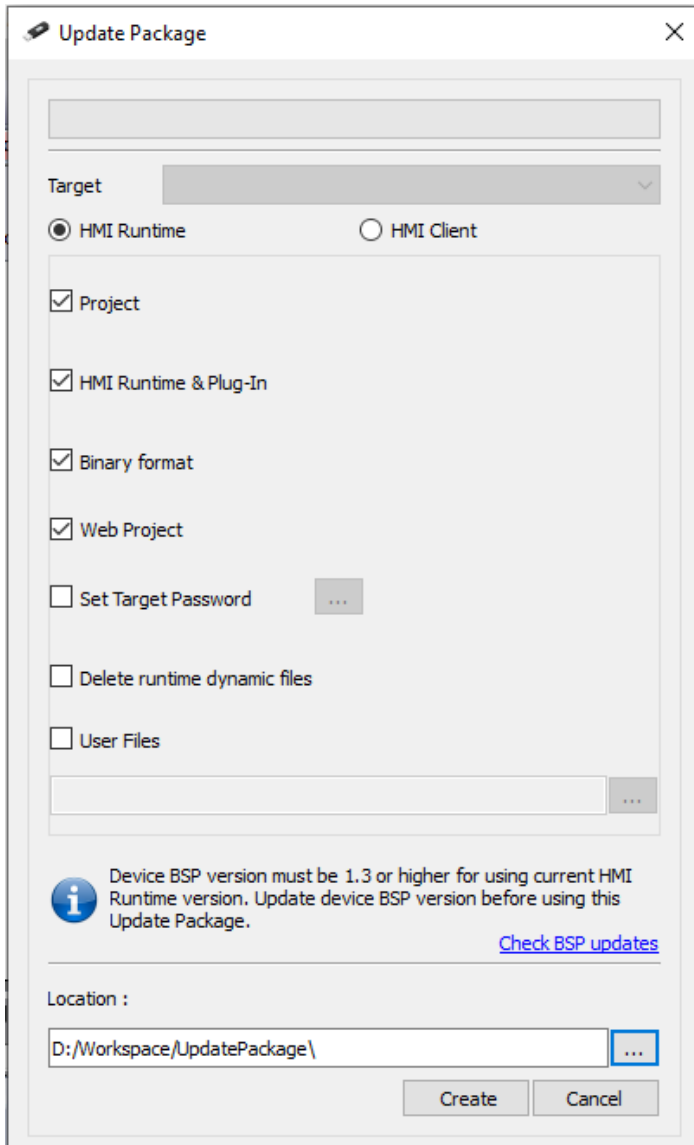
1. Click **Manage Target**.
2. Delete the projects you no longer need to make more memory available.

## Update package

The Update Package create a UpdatePackage.zip file to install or update the application inside the HMI device using an USB memory key.

## Creating an update package

Path: **Run > Update Package**



Option	Description
<b>Target</b>	HMI device type. Selected automatically if the project is open.
<b>Application Selector</b>	Select the application to insert inside the UpdatePackage.zip <ul style="list-style-type: none"> <li>• HMI Runtime</li> <li>• HMI Client (Available only on Linux devices)</li> </ul>
<b>Project</b>	Adds open project to update package.
<b>HMI Runtime &amp; Plug-In</b>	HMI Runtime is added to the update package. If the project is open the required plugins are also added to update package.

Option	Description
<b>Binary Format</b>	Download files using binary format.
<b>Web Project</b>	Download the PB4Web pages to HMI device.
<b>Set Target Password</b>	Sets password to perform critical tasks (for example, project download/upload , board management)  See <a href="#">"Protecting access to HMI devices" on page 597.</a>
<b>Delete runtime dynamic files</b>	When checked, all dynamic files will be deleted and the FRAM will be clean up.
<b>User Files</b>	Selects files to be copied to the QTHM folder of HMI device. Max size 5 MB
<b>Encrypted</b>	On WCE devices, enables the encryption of the update package so that it can only be unzipped by the HMI Runtime.  On Linux devices, this functionality has been removed, use project encryption instead. See: <a href="#">"Project Files Encryption" on page 599.</a>
<b>Location</b>	Location of update package.



**Important:** When create a package with the HMI Runtime application, always include both project and the runtime. If you need to use an old project with the latest Runtime version, convert the project first. See ["Installing the application" on page 5](#) for details.

### Example of user's file location

Computer:

*C:\Users\Username\Desktop\myFolder*

- *subFolder1/file1*
- *subFolder1/file2*
- *file3*
- *file4*

WinCE devices:

*/Flash/QtHmi*

- *subFolder1/file1*
- *subFolder1/file2*
- *file3*
- *file4*

Linux devices:

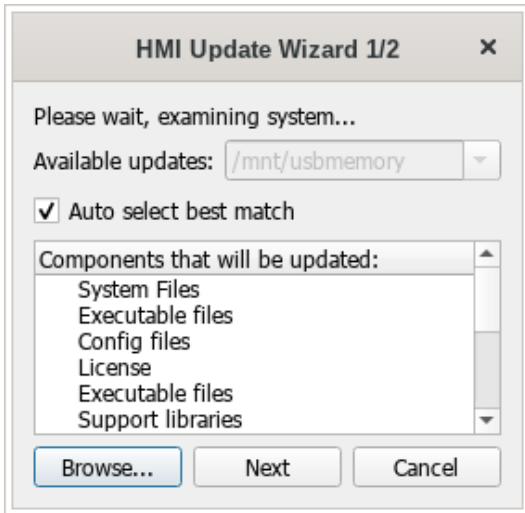
*/mnt/data/hmi/qthmi*

- *subFolder1/file1*
- *subFolder1/file2*
- *file3*
- *file4*

## Loading an update package

*Path: from the context menu > Update*

1. Assuming you have stored the package in the root folder of a USB drive, remove the drive from the computer, plug it in the HMI device, display the context menu by holding your finger for a few seconds on the screen and select **Update**.
2. The system will check for the presence of the update package in the USB drive root and ask confirmation to proceed with the update.



3. Select **Auto select best match** and click **Next**: the procedure is completed automatically. Alternatively use the browser button to select the file to use.

## The Runtime loader

HMI devices are delivered from factory without Runtime.

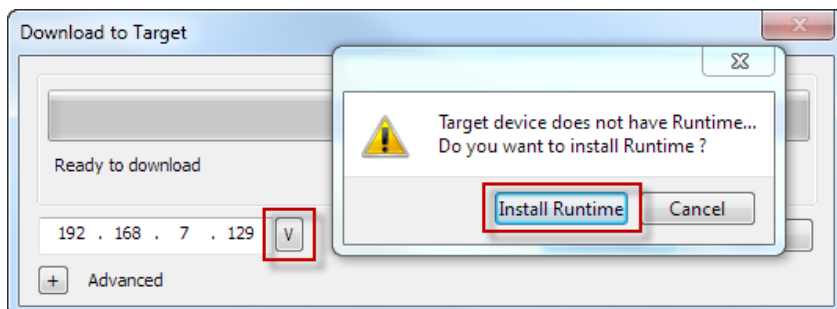
When you power up the device for the first time, the Runtime Loader window is displayed (see ["Runtime Installation" on page 617](#) for details)



*The Runtime Loader presence depends on the device Operating System and may not be available on all the units. Old versions of HMI devices may not include the Runtime Loader. Contact technical support if you need further information.*

## Installing Runtime from PB610 Panel Builder 600

When you download a project the Runtime is automatically installed if needed.



See ["Transferring the project to HMI device" on page 101](#) for details.

1. Click **Install Runtime**: the procedure is run automatically.

## Installing Runtime from a USB drive

1. Prepare the Update Package as described in ["Update package" on page 104](#)
2. Plug the USB drive in the device and follow the instructions for the type of device (see ["Install Runtime via USB Memory" on page 618](#) for details)



*Note: Old versions of HMI devices may not support automatic installation of Runtime. Contact technical support for more information.*

## Upload projects

**Path: Run > Manage Target**

You can copy a project from the Runtime to the computer where PB610 Panel Builder 600 is running.

1. In the **Runtime** tab, select the IP address of the device from the drop-down list **Target**.

2. Click **Retrieve Projects**: a list of all the projects available is displayed.
3. Select project to upload
4. Click **Upload Project**



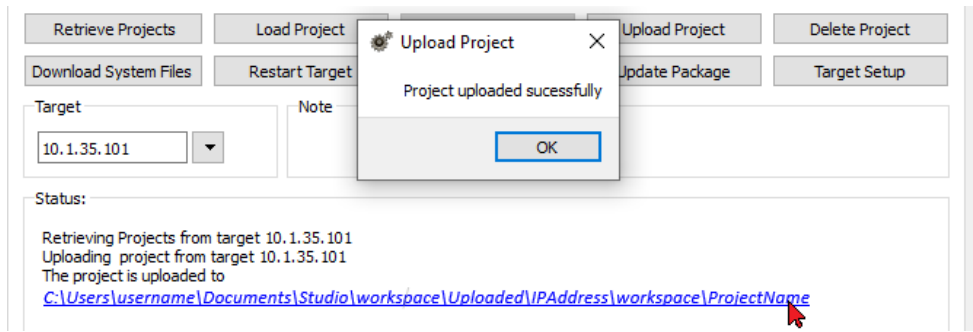
Upload could be password protected. See ["Protecting access to HMI devices" on page 597](#) for details.

5. If required, enter password. The upload process starts.

A copy of the project is saved in:

*C:\Users\username\Documents\PB610 Panel Builder 600\workspace\Uploaded\Runtime\IPAddress\workspace\ProjectName*

In the status area a clickable link will be proposed to be able to quickly open the loaded project in PB610 Panel Builder 600



Note: If the upload operation fails, check firewall settings the computer where PB610 Panel Builder 600 is running.

## Runtime dynamic files handling

HMI Runtime uses Runtime Dynamic files to store information at runtime, these information are:

- User modified Security Settings (changes to existing users, added users)
- User modified Recipe data
- User modified Schedulers settings
- Event buffers (Alarms, Audit Trail)
- Trends sampled data

All these information are project depending, the system relies on the project name, if the project name is changed (for example after a save-as operation or after a conversion to a newer runtime version) and downloaded in the target, all the existing Runtime Dynamic files will be removed and the system will create new Runtime Dynamic files for the actual application. The previous existing information will be lost.

The Runtime Dynamic files will be also deleted by selecting the apposite advanced option “Delete Runtime Dynamic Files” when downloading an application in the target.



Delete Runtime Dynamic File option is not effective if the above mentioned information are stored into an external memory location (USB or SD Card)

The below table shows the behavior of the system in the possible use cases.

Operation	Runtime Dynamic Files behavior
Application download via Ethernet, project not renamed	Maintained
Application download via Ethernet, project renamed or different project	Deleted
Application download via Ethernet, Delete Runtime Dynamic files option selected	Deleted
Application download via Update Package, project not renamed	Maintained
Application download via Update Package, project renamed or different project	Deleted
Application download via Update Package, Delete Runtime Dynamic files option selected	Deleted



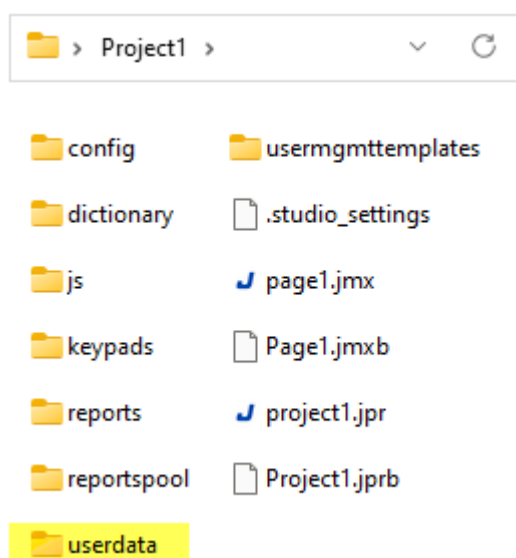
Operation	Runtime Dynamic Files behavior
Update Runtime only via Ethernet	Maintained
Update Runtime only via Update Package	Maintained
Update Runtime and application via Ethernet, project not renamed	Maintained
Update Runtime and application via Ethernet, project renamed or different project	Deleted
Update Runtime and application via Ethernet, Delete Runtime Dynamic files option selected	Deleted
Update Runtime and application via Update Package, project not renamed	Maintained
Update Runtime and application via Update Package, project renamed or different project	Deleted
Update Runtime and application via Update Package, Delete Runtime Dynamic files option selected	Deleted

## Userdata

Sometimes we need to download, with projects, additional resource files (e.g. some protocols needed additional .csv files). The "userdata" is a special folder that you can add inside your project folder to upload your additional files into the HMI device.



**Note that any other folders with different names will not be downloaded to the HMI device. Only the "userdata" folder and its contents (both files and subfolders) will be downloaded.**



Inside the HMI device, the "userdata" folder will be available on the below path:

```
/mnt/data/hmi/qthmi/deploy/workspace/project1/userdata/
```



Note that "project1" will be the name of your project



# 10 Tag editor

---

A tag is a friendly name used to identify the memory location of a device. Tags can be read or write from an external device through communication protocols.

From the Tags Editor, you can configure the protocols and the list of tags to use.

---

<b>Communication protocols</b> .....	<b>114</b>
<b>Adding tags</b> .....	<b>115</b>
<b>Exporting tags</b> .....	<b>118</b>
<b>Importing tags</b> .....	<b>119</b>
<b>Tag find and rename</b> .....	<b>123</b>
<b>Tag find and replace</b> .....	<b>125</b>

# Communication protocols

**Path: ProjectView > Config > Protocols**

Device communication drivers are configured in the **Protocol Editor**. You can add up to the maximum number of protocols as specified in Table of functions and limits. Variable and System Variables are not counted as protocols.

See "[Communication protocols](#)" on page 659 for more details.



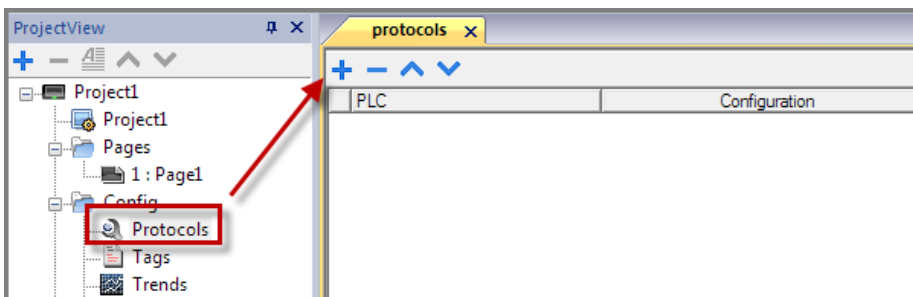
Note: you can run different Ethernet protocols over the same physical Ethernet port, but you cannot run different serial protocols using the same serial port. Some serial protocols support access to multiple controllers, but this option is set within the protocol itself which is still counted as one protocol.

## Adding a protocol



Note: Refer to CP600 operating instructions manual in case you need cables information.

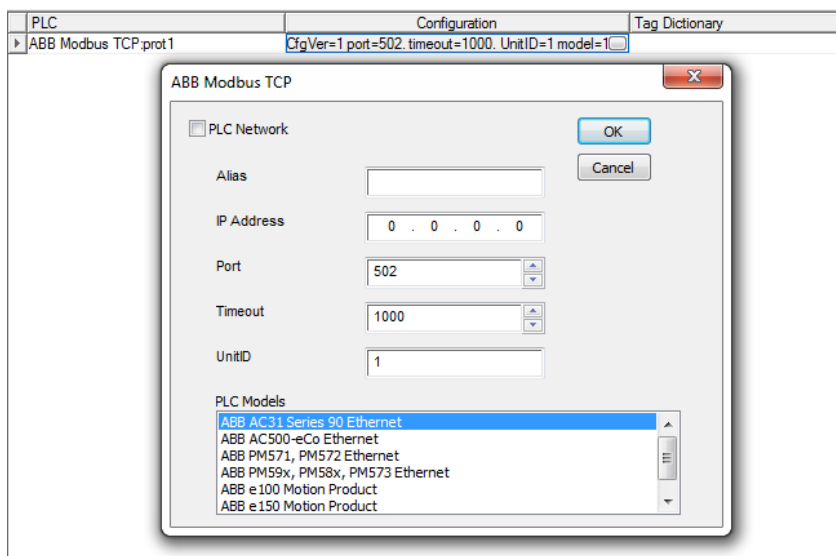
1. Click +.



2. Select the protocol from the **PLC** list and enter the required values.

## Changing protocol settings

To change configuration parameters, click the browse button in the **Configuration** column.



## Protocol parameters

Click **Show Advanced Properties** icon to see all parameters.

Parameter	Description
<b>Dictionaries</b>	Tags imported for the protocol. See <a href="#">"Importing tags" on page 119</a> for details.
<b>Enable Offline Algorithm Offline Retry Timeout</b>	See <a href="#">"Automatic offline node detection" on page 329</a> for details.
<b>Version</b>	Protocol version available in PB610 Panel Builder 600 for selected HMI device.

## Adding tags

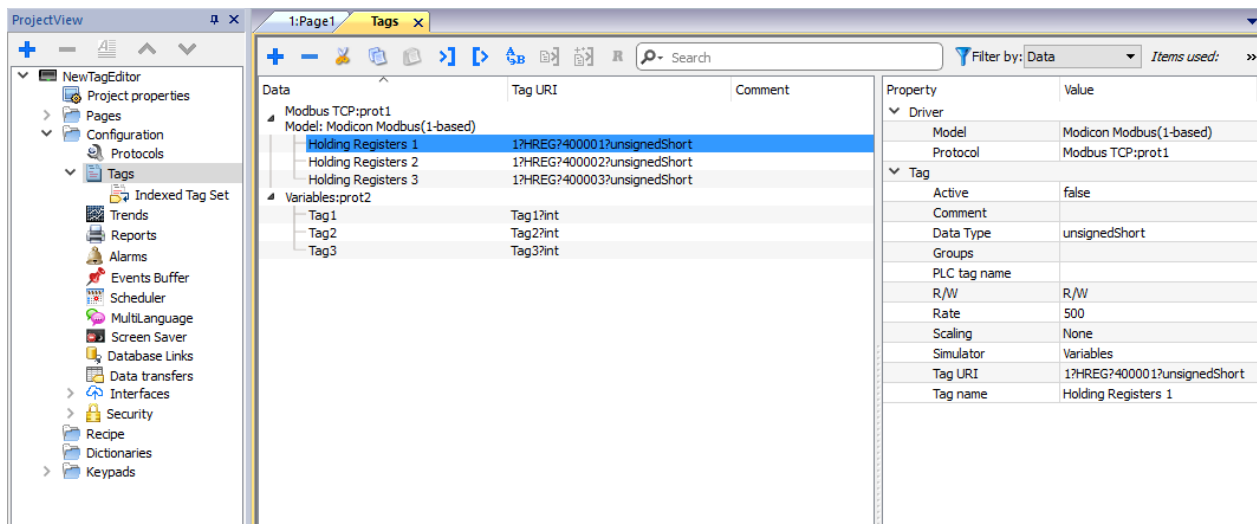
PB610 Panel Builder 600 uses tag names to access all device data. All fields and reference locations in the device need to be assigned a tag name to be used in the HMI project.

Tag Editor can be used to create and manage tags. After the tags have been defined, they can be used in the project by attaching them to widgets' properties.

See [""Attach to" parameters" on page 49](#) for details.

## Tag editor

Path: **ProjectView > Tags**



## Adding a tag




1. Click **+** and enter the required data.
2. Select the Address from the communication protocol address dialog
3. Click on the fields that are inside the property dialog if something is to change (e.g. tag name)




Note that if a tag is selected, the add tag command + will create a new tag using the property of the selected tag.

## Tag properties

Some properties depend from the protocol used. See specific protocol documentation for details.

Property	Description
<b>Active</b>	<p>Update mode.</p> <ul style="list-style-type: none"> <li>• <b>false</b> = tags are read from controller only when required by the HMI device.</li> <li>• <b>true</b> = tags are continuously read even if not required by the displayed page.</li> </ul> <p> <b>Important: Leave this value set to false for higher communication performance.</b></p>
<b>Description</b>	Tag description
<b>Encoding</b>	Encoding type for string data type (UTF-8, Latin1, UTF-2 and UTF-16)
<b>Groups</b>	Group names associated to a tag
<b>PLC tag name</b>	Original PLC tag name, used to match tags used by HMI application (Tag Name) and tags exported from PLC
<b>R/W</b>	<p>R/W tag attribute (R/W, R or W).</p> <p> Note: The content of Write Only tags is always written and never read. When communication is not active, the content of these tags may not be available in widgets.</p>
<b>Rate</b>	<p>Tag refresh time. Default: 500ms (except SystemVariable which is 1000ms).</p> <p>When the refresh rate is set to “Manual”, the HMI device will not read the tag from the remote device automatically in background. Tag is read and refreshed into the database only by explicitly required from the “ForceReadTag” action or using the forceRefresh option into the JavaScript getTag().</p> <p> <b>WARNING: Tags refresh rate is the maximum refresh rate. Actual refresh rate depends on: communication type (serial, fieldbus, Ethernet), protocol, amount of data exchanged.</b></p>
<b>Scaling</b>	<p>Conversion applied to tag before database storage.</p> <ul style="list-style-type: none"> <li>• <b>By Formula</b> = defined as a linear transformation.</li> <li>• <b>By Range</b> = defined as a range conversion.</li> <li>• <b>Fixed Point</b> = fixed point scaling</li> </ul>
<b>Simulator</b>	Tag behavior during simulation. Several profiles are available.

Property	Description
Tag URI	Controller memory address. To edit click on the right side of the column to get the dialog box where you can enter the address information.
Tag name	Unique tag name at project level. Primary key to identify information in the runtime tag database.  <b>WARNING: Duplicate tag names are not allowed.</b>



Note that is allowed to select multiple tags in Tag Editor and to change the same property to all (e.g. to change refresh time in 10 tags to 500 without change it in all tags one by one).

## Managing tag names

Tag names must be unique at project level. If the same tags, from the same symbol file have to be used for two different controllers, use the "Alias" feature to add a prefix to the imported tags and make them unique at project level.



Note: Not all protocols support the "Alias" feature.

See "[Communication protocols](#)" on page 659 for details.

## Managing tag groups

Tags used in each page are identified as part of a group, so that requests made by the communication protocol to the connected controller(s) can be processed faster: only the tags included in the displayed page are polled from the controller.

## Scaling

Using the tag scaling function it is possible to resize the tag values that will be visible from the HMI application.

There is the possibility to configure

- Linear transformation, using the "By Formula" or the "By Range" mode
- Fixed Point transformation


Generally, the data type used inside the HMI is the same data type inherited from the PLC device. When a transformation is used, considerate the possibility to change the HMI's data type to not lose precision.

### Example

If your PLC manages value with two decimal digits using an integer in fixed point, you can configure the scaling transformation as the below picture where the value read from the PLC will be divided by 100 and stored inside a float data type. E.g. PLC integer value 12345 will become the float value 123.45 inside the HMI device.



Fixed Point

Number of decimal digits:  

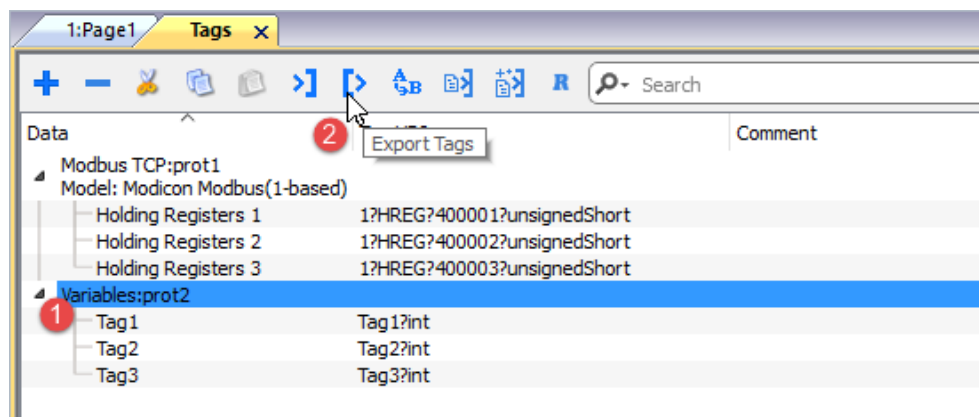
---

Converted:

HMI data type:

## Exporting tags

Path: *ProjectView* > *Tags*



1. Select the protocol for the tags you want to export.
2. Click the **Export Tags** button: all the tags configurations for the selected protocols are exported into an .xml file.

You can edit the resulting .xml file using third part tools (for example, Microsoft Excel) and then re-import the modified file (see "[Importing tags](#)" on the facing page for details).

# Importing tags

## Introduction

Some protocols allow you to import tags stored in a comma separated file (.csv or other formats). Refer to the Tag Import section of each protocol for details (see ["Communication protocols" on page 659](#)).

Importing is a two step process:

1. Import of the tag definition into a dictionary
2. Import tags from the dictionary to the project

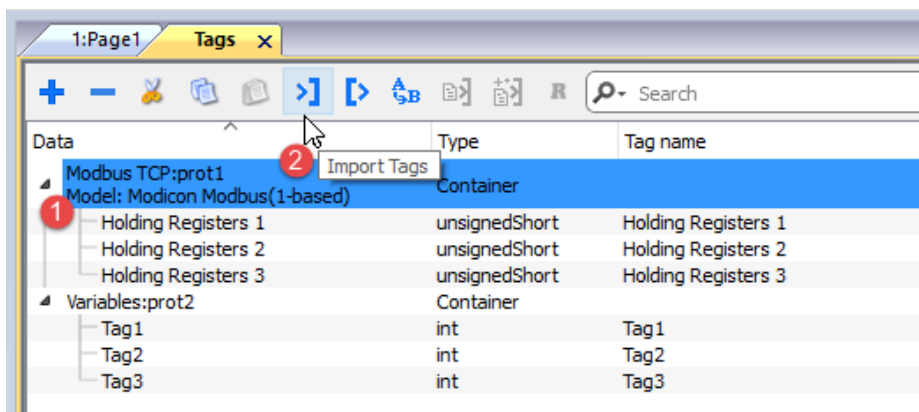


**WARNING: Special characters in tag names, such as "&" character, that can cause communication errors will be substituted with the underscore "\_" character when imported. See ["Limitations in Unicode support" on page 338](#)**

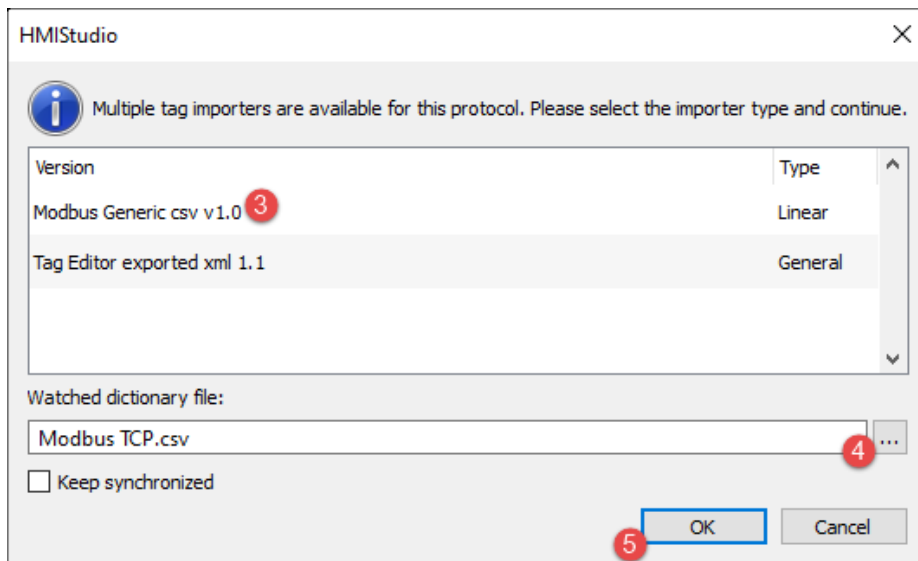
## Importing tags

To import tags from an external file:

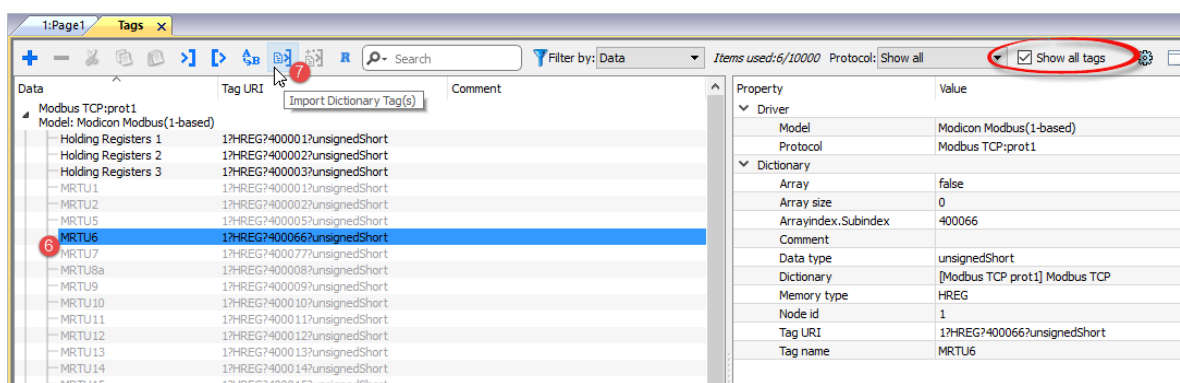
1. In **ProjectView, Tags** select the protocol from the filter list.



2. Click the **Import Tags** button: the dialog to choose the importer type appears. The list of the supported importers is depended from the selected protocol.



3. Select the importer type to use
4. Select the dictionary file
5. Press OK to attach the dictionary file to project file. The tags available within the Dictionary but not imported into the project are gray and are visible only when the "Show all tags" check box is selected.

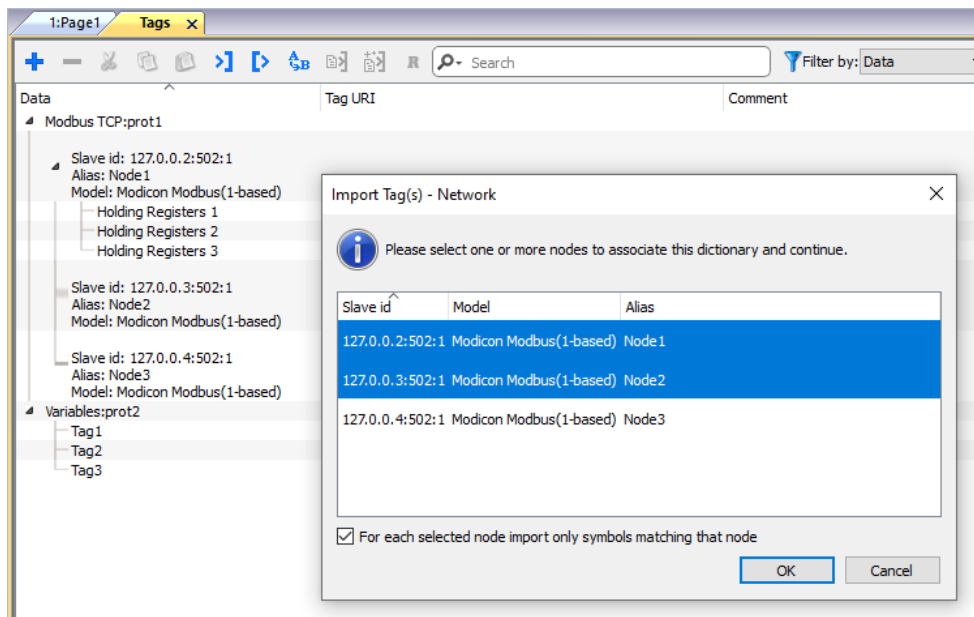


6. To import tags, select one or more tags or a node (hierarchical view only)
7. Click the **Import tag** button: tags are imported to the project and listed in black color.

When the project is configured to use a protocol network you must also select the protocol node where tags are to be imported. You can import the same tags on multiple protocols.

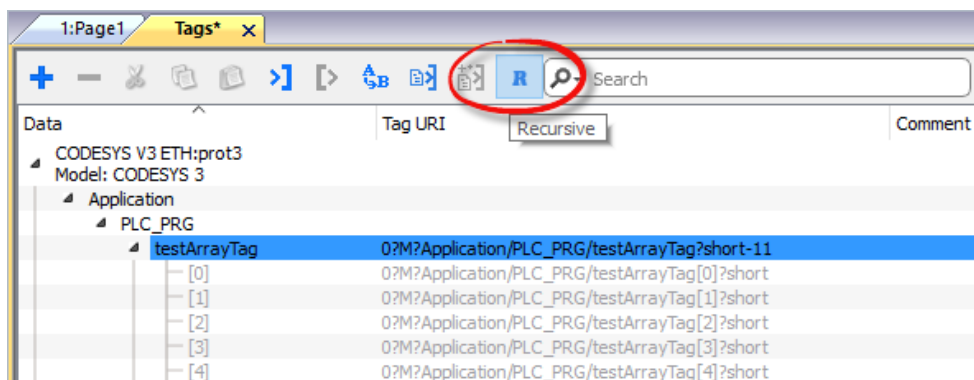
#### For each selected node import only symbols matching that node

When the tags file contains the node information, you can choose to use the information to filter the tags and import only those matching with the selected nodes.



### Recursive

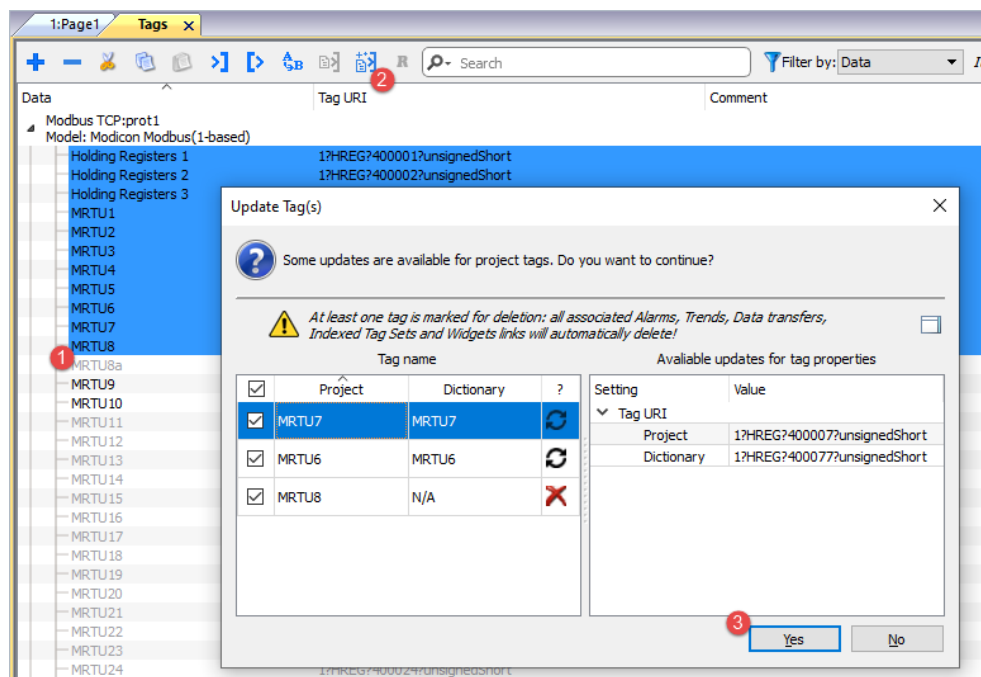
Recursive is a toggle button. When selected, when an array tag is imported even all array elements are imported into separate tags.





### Updating the imported tags

To check the dictionary file and update the imported tags:

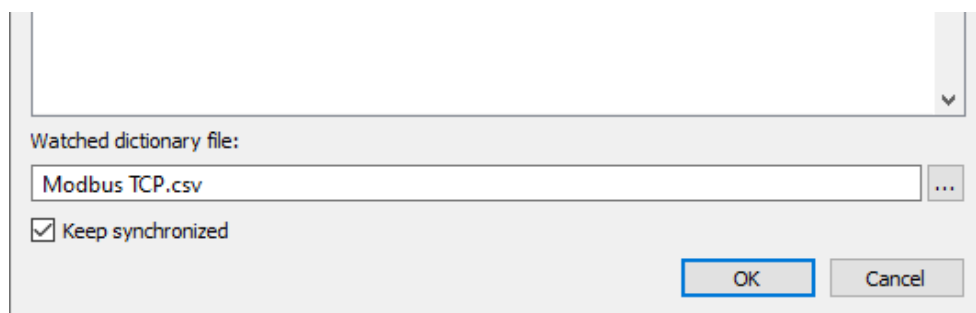
1. Select the tags that you want to check
2. If some change is found the update icon will be enabled, click the icon and the "Update Tag(s)" dialog with the list of found differences is showed
3. Unchecked the tags that you do not want yo update and click OK to confirm



-  These tags need to be updated. The list of differences between project and dictionary is displayed.
-  These tags are no longer available in the dictionary. If updated, these tags will be removed from the project.

### Keep Synchronized

Check the "Keep Synchronized" check box if you want that PB610 Panel Builder 600 checks and update the tags from file dictionary automatically without user intervention.

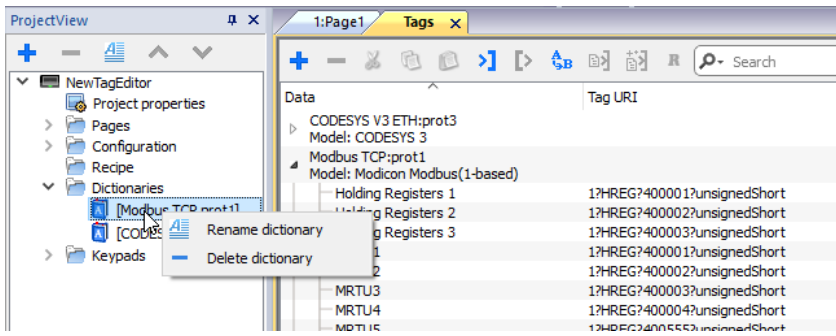


### Dictionaries

**Path:** *ProjectView > Dictionaries*

A dictionary is a list of tags imported in the Tag Editor for a specific protocol. Depending on the protocol type, tags are shown in linear view or in hierarchical view.

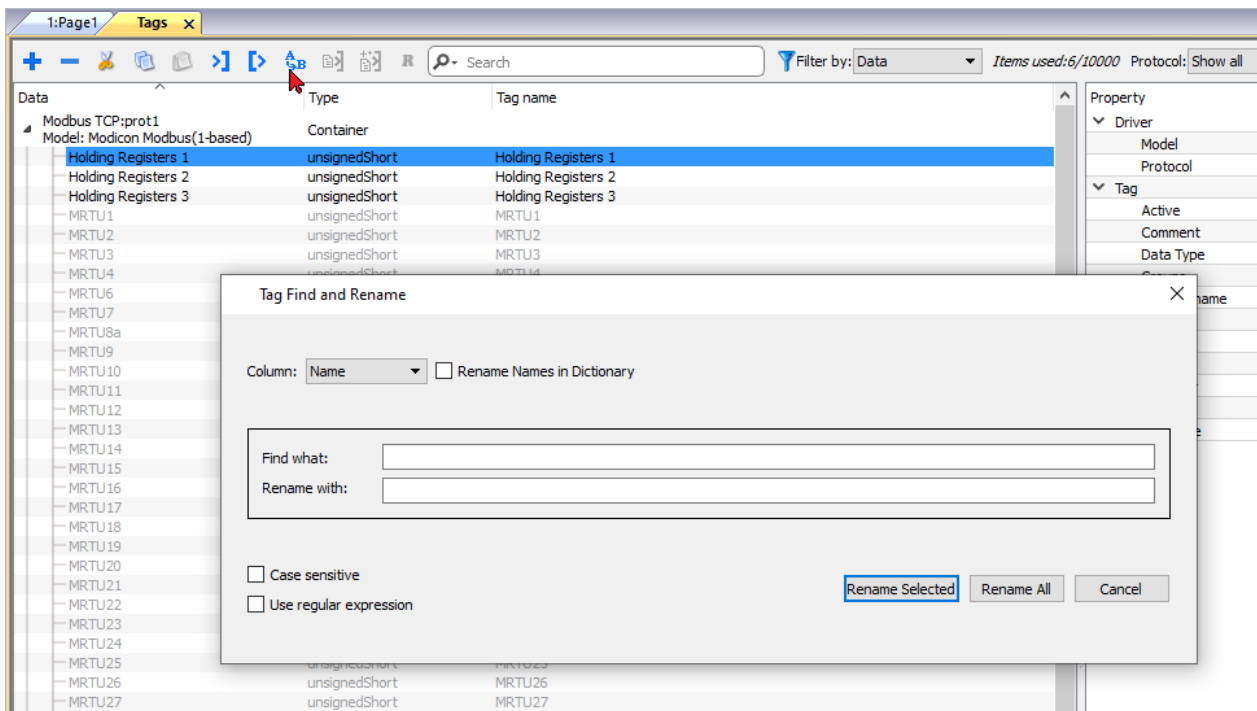
To remove a dictionary, right-click the dictionary name.




## Tag find and rename



Tag find and rename feature will rename a tag reference inside the entire project.

Note this feature can be used not only to rename tags, but even to change values from each columns of the tags editor



### Parameters

<b>Column</b>	Select the column to modify with the find and rename operation
<b>Rename Names in Dictionary</b>	The tags' names rename will be extend to rename even to the internal dictionary tags' names. This parameter is useful when you have to substitute the dictionary with another dictionary that contains renamed tags
	 This parameter is available only when the selected column is

Parameters	
	 "Name"
<b>Find what</b>	String to search
<b>Rename with</b>	String to replace
<b>Case sensitive</b>	Takes account of upper and lower case letters
<b>Use regular expression</b>	Enable regular expression in search/replace pattern See <a href="https://en.wikipedia.org/wiki/Regular_expression">https://en.wikipedia.org/wiki/Regular_expression</a> for additional details regarding regular expressions.  When regular expression is enabled, the "Find what" parameter will not offer predefined values but only free text handling.

## RENAME SELECTED

Execute the rename only for the selected tags

## RENAME ALL

Execute the rename for the entire tags database



**References used in Java script and within custom widgets will not update.  
Undo is not supported for this command**

## Regular expression example

Using the tags list of the above picture.

If you want add a prefix to all tags you don't need to use regular expression:

Find what:           **MRTU**  
 Rename with       **PLC01\_MRTU**

But if you want add a postfix, you need to use a regular expression:

Find what:           **MRTU(.\*)**  
 Rename with       **MRTU\1\_PLC01**

Where

- (.\*) is meaning any sequence of characters
- \1 is a copy of the first sequence of characters enclosed by (...) found inside the search string

## Tag find and replace

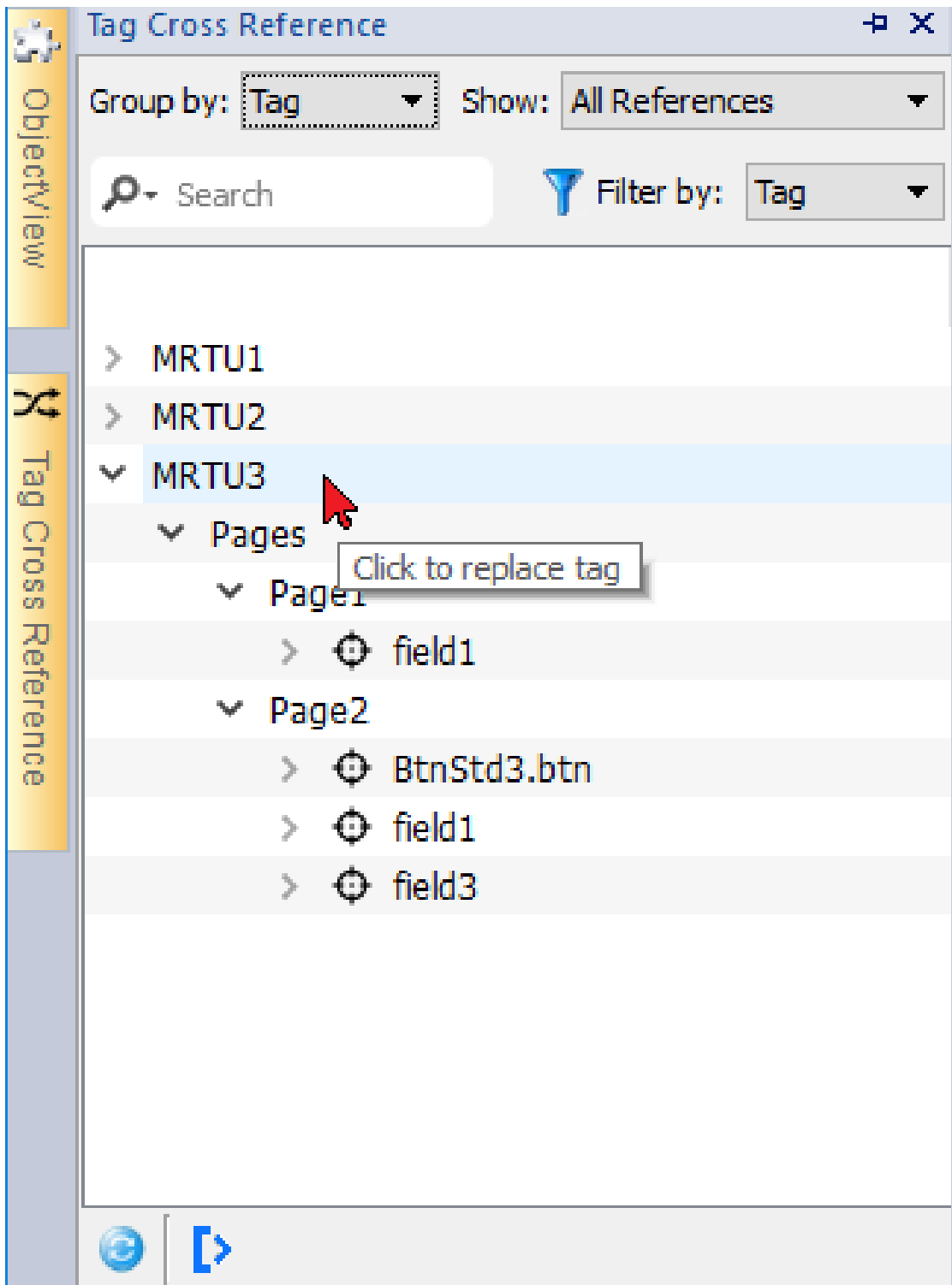
Using this feature you can search all occurrence of a tag inside the project and replace it with another tag.



Tag replace is only applicable for Protocol tags which are shown in black color and not for System Variable, Alias and Recipe tags. (See "[Opening the Tag Cross Reference pane](#)" on page 138 for the different colors meaning)

From the Tag Cross Reference view, click the tag that you want replace





Using the check boxes select where you want apply the replace,

then click the “Select new tag” button to replace the data links of the selected objects or press “Cancel” to abort the operation.

Tag Cross Reference

Group by: Tag Show: All References

Search Filter by: Tag

- > MRTU1
- > MRTU2
- ▼ MRTU3
  - ▼  Pages
    - ▼  Page1
      - >  field1
    - ▼  Page2
      - >  BtnStd3.btn
      - >  field1
      - >  field3

Select new tag Cancel



References used in the Java script and within custom widgets may not be listed.  
Undo is not supported for this command



# 11 Indexed addressing

---

Indexed addressing allows you to select a set of tags depending on the value of another tag. This is very useful, for example, to use the same graphics to visualize a set of data coming from different sources, all the user has to do is pick the source to monitor from a list.

---

<b>Creating an indexed addressing set .....</b>	<b>130</b>
<b>Using indexed tag set in pages .....</b>	<b>135</b>

# Creating an indexed addressing set

## Scenario

In this scenario, environment data is collected from with four rooms, each equipped with temperature, pressure, and humidity sensors. Data is available as follows:

Room Number	Temperature	Pressure	Humidity
1	Room1-Temperature	Room1-Pressure	Room1-Humidity
2	Room2-Temperature	Room2-Pressure	Room2-Humidity
3	Room3-Temperature	Room3-Pressure	Room3-Humidity
4	Room4-Temperature	Room4-Pressure	Room4-Humidity

Using the indexed addressing feature, you can use a single table format to arrange all data in the HMI device.

Data from the three different sensors can be displayed in a single page where the room number is used as a selector (combo box) to pick the correct set of tags.

Room 1 ▾

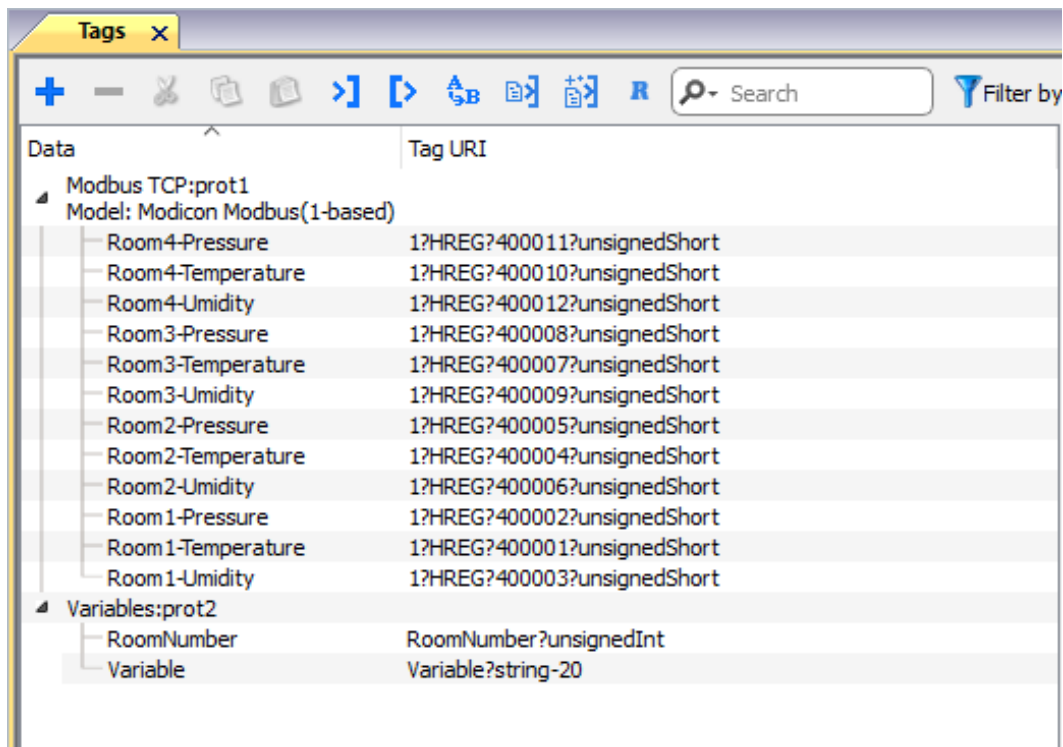
Temperature (°C)	21
Pressure	1
Umidity (%)	75

## How to create an indexed tag set

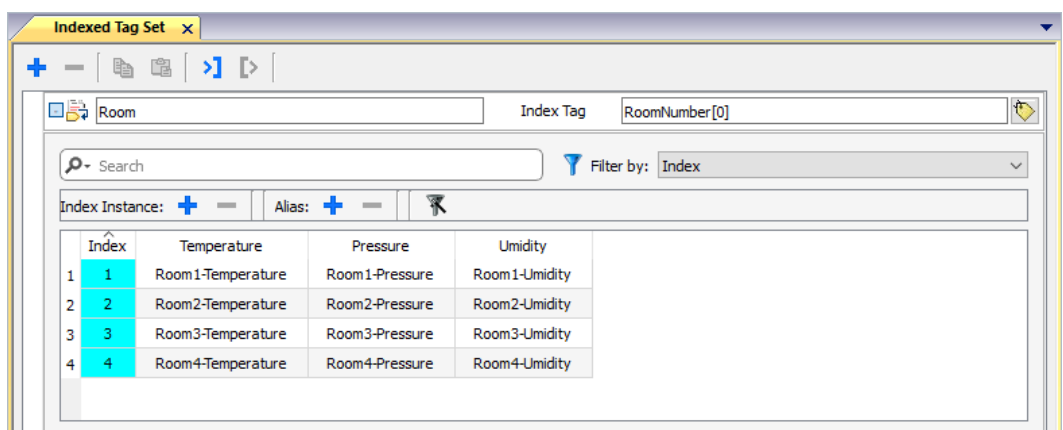
**Path:** *ProjectView* > *Tags*

To do this you need to create an indexed tag set.

1. In the Tag Editor, define protocols and tag. Define a tag for each data to be indexed, in this example you must create a tag for each sensor in each room.



2. Create a tag to be used as index tag. In this example you create a "RoomNumber" tag that could be of type UnsignedInt using Variable protocol.
3. From **ProjectView**, select **Config> Tags**, double-click **Indexed Tag Set**: the Indexed Tag Set editor is displayed.
4. Click + to add an Indexed Tag Set. In this example you will call it "Room".
5. Select the tag "RoomNumber" to use as a selector for the room number.
6. Create an **Index Instance** for each set of data. In this example, one for each room.
7. Create an **Alias** for each type of data and rename the table columns appropriately. In this example "Temperature", "Pressure" and "Humidity".
8. Double-click on each cell to associate the correct tag.



Note: The Index Tag datatype can be a number, a string or any type of simple data types.



Note: To reference an array data type use the array index = -1

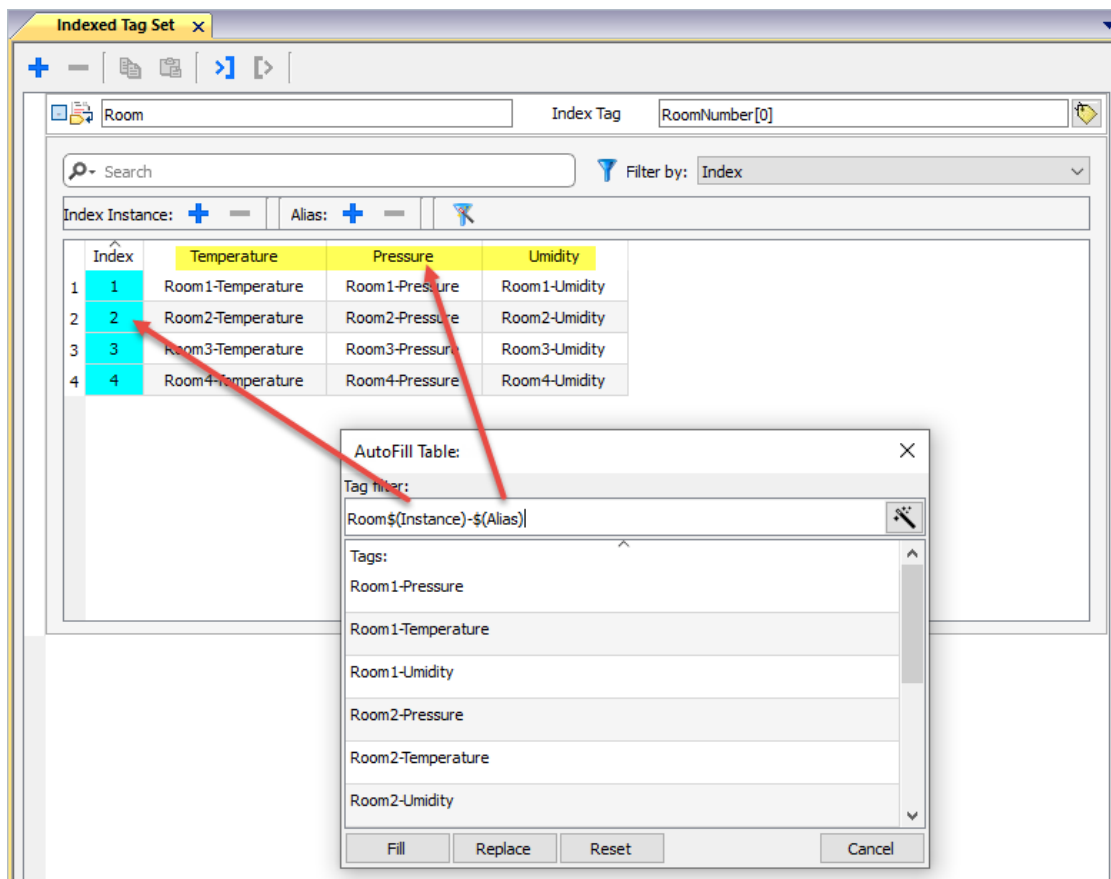
### Index Tag

The "Index Tag" used to select the instance to use can be a Tag or an element of the Global \_VariablesWgt widget (Ref.: "Global Variable Widget" on page 517). Note that using a Tag the selection will be global for all clients while using the \_VariablesWgt widget the selection will be local and any client can have its own selection.

### Autofill function

An Indexed Tag Set table may become very complex and filling it may be an error prone procedure. Enable the Autofill feature to make sure aliases are entered correctly.

Click  to enable the Autofill feature: the **Autofill Table** is displayed.



This function uses regular expression for populating the table with tags trying to match the filter where the keyword \$(Instance) will be replaced with the defined Index values and the keyword \$(Alias) with the defined alias labels.

See [https://en.wikipedia.org/wiki/Regular\\_expression](https://en.wikipedia.org/wiki/Regular_expression) for additional details regarding regular expressions.

### Autofill example

“Room\$(Instance)-\$(Alias)” will match all tag names:

Room1-Temperature,

Room1-Pressure,

Room1-Humidity,

Room2-Temperature,

...

“Room0\*\$(Instance)-\$(Alias)” will match all tag names:

Room1-Temperature,

Room01-Pressure,

Room001-Humidity,


Room2-Temperature,

Room02-Pressure,

Room002-Humidity,

...

## Autofill table elements

Element	Description
<b>Fill</b>	Fills in missing entries in the tag table using the set filter (if any). For example, when new instances or new aliases are added you can use this option to fill in the new entries.
<b>Replace</b>	Replace all table entries with those provided by the Autofill table.
<b>Reset</b>	Resets the tag filter to empty, no automatic fill is done.
	Suggests a valid filter expression for your project.

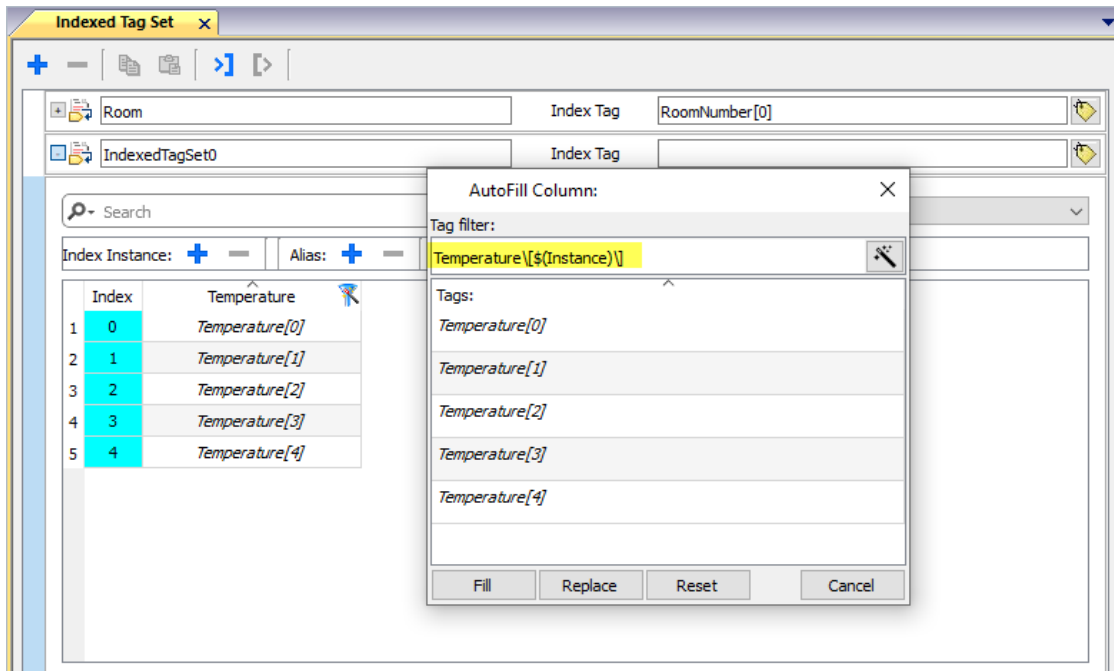


Note: Filters are saved as project preferences and can be set for the entire table or for a column. Once a filter is set for a column, the table filter is ignored. You can therefore selectively change the filter for handling a particular alias only.









Note: To reference the elements of an array use the \ character to disable the regular expression interpretation of the square brackets (array tags are differentiated by Italic).





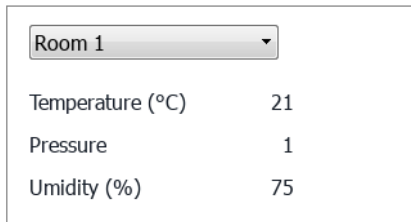
## Toolbar



Toolbar Element	Description
	Add a new Indexed Tag Set
	Remove the select Indexed Tag Sets
	Copy the selected Indexed Tag Sets
	Paste the copied Indexed Tag Sets
	Export selected Indexed Tag Sets to .xml file
	Import Indexed Tag Sets from .xml file

## Using indexed tag set in pages

Once an indexed tag set has been created, you can use it to create a page for the HMI device as in this example.

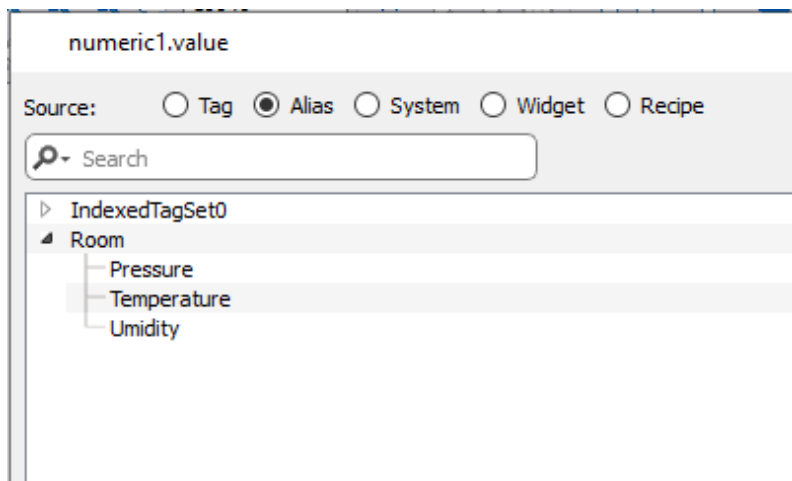


To create this page:

1. Create a page and add a combo box, three labels and three numeric fields.
2. Use the index tag created for the room number for the combo box, "RoomNumber" in this example. This will be the selector for the room number.
3. Create a list for the combo box. In this example use the following list.

Index	String List
0	Room Number
1	Room 1
2	Room 2
3	Room 3
4	Room 4

4. Attach to each numeric field value the corresponding Alias variable (**Room > Temperature**, **Room > Humidity**, **Room > Pressure**).





# 12 Tag cross reference

---

The **Tag Cross Reference** pane displays a list of tag names used in current project organized according to their location and use.

From this pane you can:

- verify where each tag is used (alarms, pages, recipes, schedulers, trends, and so on)
- identify invalid tag references (references to tags not defined in the tag editor)
- identify tags not used in the project



Note: The Tag Cross Reference pane may not be list all tags used in JavaScript code.

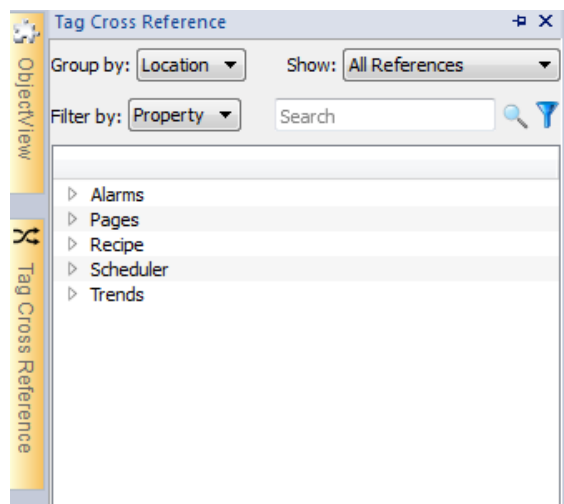
---

<b>Updating data in the Tag Cross Reference pane</b> .....	<b>141</b>
--	------------

## Opening the Tag Cross Reference pane

*Path: View > Toolbars and docking windows > Tag Cross Reference*

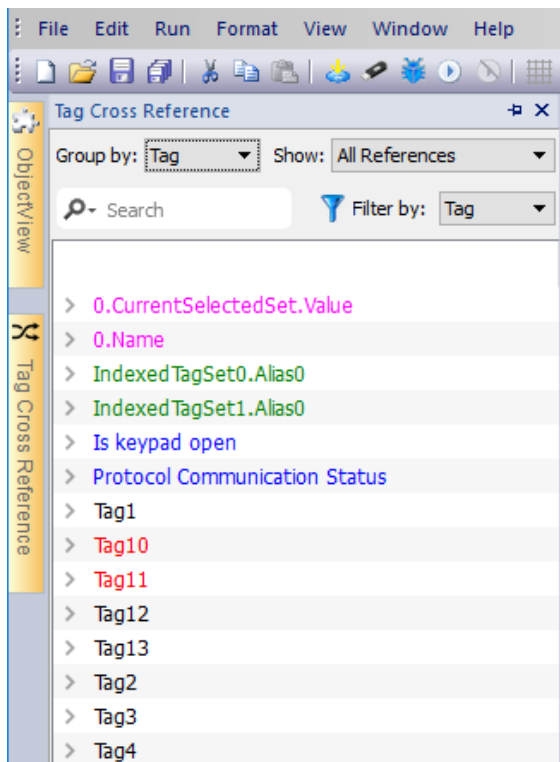
Click the **Tag Cross Reference** tab to open the Tag Cross Reference pane.



### Meaning of colors

Black	Protocol Tags
Magenta	Recipe Tags
Blue	System Variable Tags
Dark Green	Alias Tags
Red	Invalid Tags

Example:



## Working in the Tag Cross Reference pane

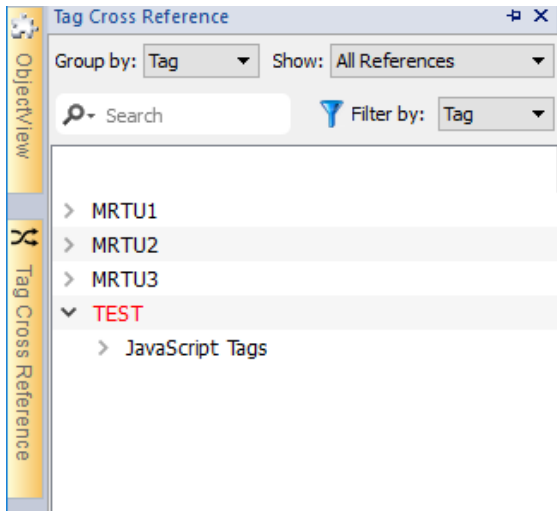
The Tag Cross Reference pane provides a set of standard functions.

Element	Function
<b>Group by</b>	Groups tags by <b>Location</b> (alarms, pages, trends and so on) or <b>Tag</b> name
<b>Show</b>	Filters tags and displays: <ul style="list-style-type: none"> <li>• <b>All Reference</b>: all tags</li> <li>• <b>Invalid Tag Reference</b>: tags not listed in the Tag Editor.</li> <li>• <b>Unused Tags</b>: tags listed in the Tag Editor but not used in project.</li> </ul>
<b>Search field</b>	Applies a filter to display a limited number of tags
<b>Filter by</b>	Filters tags by <b>Location</b> , <b>Tag</b> or <b>Property</b> .

Navigate the listed tags to find where they are used inside the project.

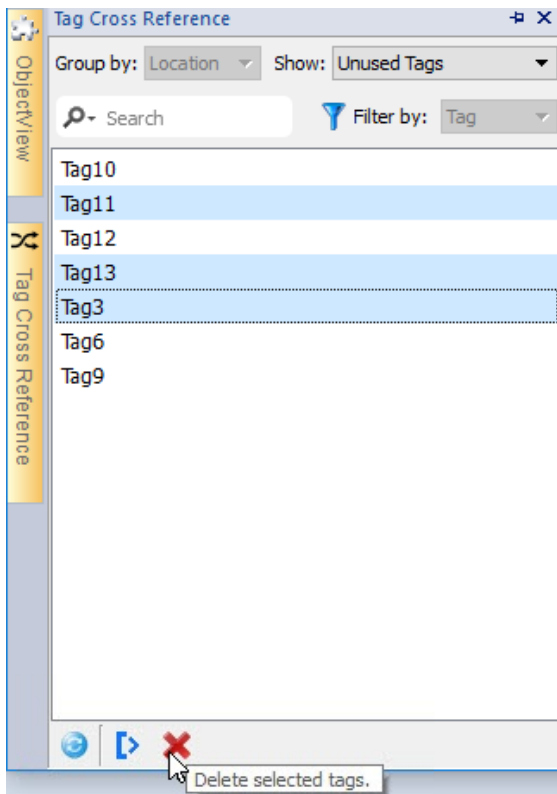
Double-click on a tag to open the editor or page where it is used.

**Invalid tag references will be listed in red color:**



### Delete unused tags


From the unused tags view, it is possible to select one or more tags and delete them from the tag editor. To select a tag, click a tag, to select multiple tags use SHIFT or CTRL keys.



**Be aware that eventually tags referenced inside JavaScript may not be found (depends on how the code was written). Even it is not a good practical to using tags' references inside custom widgets, even tags' referenced inside Custom Widgets may not be detected from the Tag Cross Reference engine.**

# Updating data in the Tag Cross Reference pane

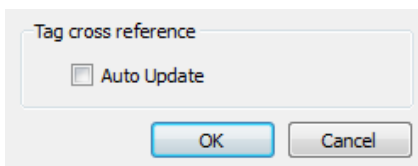
## Manual update

By default, the information displayed in the Tag Cross Reference pane must be updated manually. To do this, click the refresh button . A warning sign is displayed when a refresh is needed.

## Automatic update

*Path: View > Properties*

You enable the automatic update of the Tag Cross Reference pane from the PB610 Panel Builder 600 **Properties** page.



Select the **Auto Update** option.

## Exporting data

Data displayed in the Tag Cross Reference pane can be exported in .csv file.

Data is organized in the exported file according to how it was grouped in the pane.

Grouped by	File format
Location	RESOURCE, RESOURCE DESC, WIDGET-ID, ATTRIBUTE, TAG
Tag	TAG, RESOURCE, RESOURCE DESC, WIDGET-ID, ATTRIBUTE



Note: The separators used in export operation depends on regional settings of your computer.





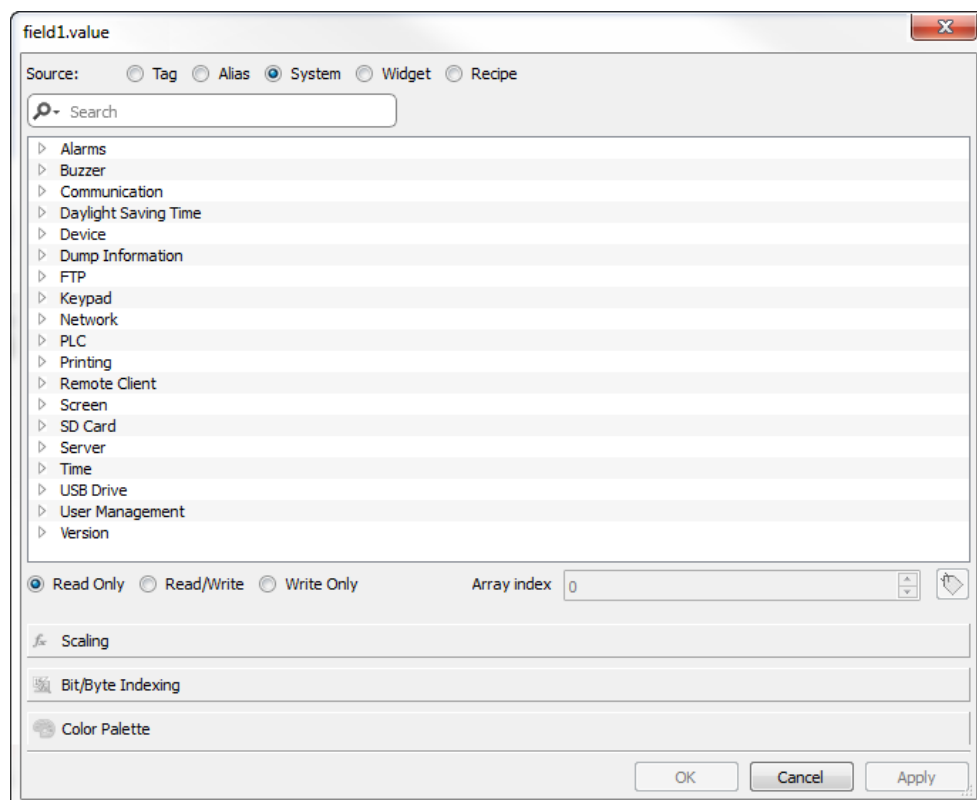
# 13 System Variables (Attach To)

Path: **Source**> **Attach to**

System variables are special tags containing information about the HMI runtime.



Note: System Variables are available also as a standard protocol in the Protocol Editor. Use System Variables as a protocol when you have to transfer data between system variables and tags from devices, or to select custom refresh rate for a system variable.



<b>Alarms variables</b> .....	<b>145</b>
<b>Buzzer variables</b> .....	<b>145</b>
<b>Communication variables</b> .....	<b>146</b>
<b>Daylight Saving Time variables</b> .....	<b>146</b>
<b>Device variables</b> .....	<b>147</b>
<b>Dump information variables</b> .....	<b>149</b>
<b>FTP client variables</b> .....	<b>150</b>
<b>Keypad variables</b> .....	<b>150</b>
<b>Language variables</b> .....	<b>150</b>
<b>Network variables</b> .....	<b>151</b>

---

---

<b>Printing variables</b> .....	<b>151</b>
<b>Remote Client variables</b> .....	<b>152</b>
<b>Version variables</b> .....	<b>153</b>
<b>Screen variables</b> .....	<b>153</b>
<b>SD card variables</b> .....	<b>153</b>
<b>Server variables</b> .....	<b>154</b>
<b>Time variables</b> .....	<b>154</b>
<b>Touch screen variables</b> .....	<b>155</b>
<b>USB drive variables</b> .....	<b>156</b>
<b>User management variables</b> .....	<b>156</b>

## Alarms variables

Number of alarms of the requested type.


Variable	Description	Data type
<b>Alarm not acknowledged</b>	True when alarms unacknowledged is pending (Not Triggered Not Acknowledged<>0) OR (Triggered Not Acknowledged<>0)	boolean read only
<b>Alarm triggered</b>	True when at least one alarm is triggered (Triggered Acknowledged<>0) OR (Triggered <>0) OR (Triggered Not Acknowledged<>0)	boolean read only
<b>Number of missed alarm events</b>	Alarms exceeding the event queue. Queue length is defined in the <i>engineconfig.xml</i> file.	int read only
<b>Number of not triggered acknowledged</b>	Alarm condition no longer active; alarms already acknowledged	int read only
<b>Number of not triggered not acknowledged</b>	Alarma condition no longer active; awaiting acknowledgment	int read only
<b>Number of triggered acknowledged</b>	Alarm condition active; alarms already acknowledged	int read only
<b>Number of triggered alarms</b>	Alarm active: acknowledgement not required	int read only
<b>Number of triggered not acknowledged</b>	Alarm condition active; awaiting acknowledgment	int read only



Note: For compatibility reasons, the older names are still valid but they usage is deprecated.

## Buzzer variables

Adjust buzzer behavior.

Variable	Description	Data type
<b>Buzzer Setup</b>	<p>0 = disabled            1 = enabled (buzzer sounds as audible on any touchscreen event)            2 = buzzer status controlled by <b>Buzzer Control</b> system variable or by <b>Buzzer on Touch</b> property inside the "<a href="#">Project properties</a>" on page 79</p> <p> <b>Buzzer on touchscreen (Setup=1) is not available on Linux platforms. See "Buzzer on Touch" property in alternative.</b></p>	int
<b>Buzzer Control</b>	<p>0 = buzzer off            1 = buzzer on            2 = buzzer blink</p>	int
<b>Buzzer Off Time</b>	Duration in milliseconds of off time when blink has been selected. Default = 1000. Range: 100–5000.	int
<b>Buzzer On Time</b>	Duration in milliseconds of on time when blink has been selected. Default = 1000. Range: 100–5000.	int

## Communication variables

Communication status between HMI device and controllers.

Variable	Description	Data type
<b>Protocol Communication Status</b>	<p>Summarize the status of the communication protocols.</p> <p>0 = No protocol running, protocol drivers might not have been properly downloaded to the HMI device.</p> <p>1 = Protocols loaded and started, no communication error.</p> <p>2 = At least one communication protocol is reporting an error.</p>	int Read only
<b>Protocol Error Message</b>	<p>Communication error with error source.</p> <p>For example: "[xxxx]" where "xxxx" is the protocol abbreviation, the error source.</p> <p>Multiple acronyms appear in case of multiple error sources. Blank when no errors are reported.</p>	ASCII string Read only
<b>Protocol Error Count</b>	Number of communication errors occurred since last reset. Reset value with Reset Protocol Error Count action, see " <a href="#">System actions</a> " on page 218.	int Read only

## Daylight Saving Time variables

Information on the system clock. The variables contain information on the "local" time. Standard Time (solar time) and Day Light Saving time (DST) are available.








Note: All variables are read only; you cannot use them to update the system clock.

Variable	Description
<b>Standard Offset</b>	Offset in minutes when standard time is set, with respect to GMT (for example: $-8 \times 60 = -480$ minutes).
<b>Standard Week</b>	Week in which the standard time starts (for example: First = 1).
<b>Standard Month</b>	Month in which the standard time starts. Range: 0–11. (for example: November = 10).
<b>Standard Day</b>	Day of week in which the standard time starts (for example: Sunday = 0).
<b>Standard Hour</b>	Hour in which the standard time starts (for example: 02 = 2).
<b>Standard Minute</b>	Minute in which the standard time starts (for example: 00 = 0).
<b>DST Offset</b>	Offset in minutes when DLS time is set, with respect to GMT
<b>DST Week</b>	Week in which the DLS time starts
<b>DST Month</b>	Month in which the DLS time starts. Range: 0–11.
<b>DST Day</b>	Day of week in which the DLS time starts
<b>DST Hour</b>	Hour in which the DLS time starts
<b>DST Minute</b>	Minute in which the DLS time starts

## Device variables

Device settings and operating status information.

Variable	Description	Data type
<b>Available System Memory</b>	Free available RAM memory in bytes.	uint64 read only
<b>Backlight Time</b>	Activation time in hours of the display backlight since production of the device.	unsignedInt read only
<b>Battery LED</b>	Enables/disables the low battery LED indicator (when available). <b>0</b> = disabled <b>1</b> = enabled  Not available on Linux platforms (find the platform of your device at " <a href="#">HMI devices capabilities</a> " on page 613)	int
<b>Battery</b>	Reserved	int

Variable	Description	Data type
<b>Timeout</b>	 Not available on Linux platforms (find the platform of your device at " <a href="#">HMI devices capabilities</a> " on page 613)	
<b>Display Brightness</b>	<p>Returns and adjusts brightness level.</p> <p>Even when set to 0, the backlight is still on and the <b>Backlight Time</b> counter increases.</p> <p>Range: 0–255</p> <p>On WinCE device only: When set to a low light level (0..3), the backlight stays lit to a higher level for 8 seconds to allow the user to make the adjustments and then is switched-off.</p>	int
<b>External Timeout</b>	<p>Non-operational time after which the display backlight is automatically turned off. The backlight is automatically turned on when the user touches the screen.</p> <p><b>-1 =</b> Switch off backlight and disable touch (switch display off). <b>Backlight Time</b> counter is stopped.</p> <p> On Linux devices requires BSP v1.0.324 or higher.</p> <p><b>-2 =</b> Switch off backlight but not disable touch. If touch is pressed, event is not passed to applications but screen saver exit and backlight return on.</p> <p> Available only on Linux devices. Requires BSP v1.0.324 or higher.</p> <p><b>0 =</b> Switch backlight on (switch display on)</p> <p><b>1..n =</b> Timeout, in seconds, for switch off backlight (screen saver timer)</p> <p> The timeout value is rounded to multiples of one minute (60, 120, 180, etc.) in all Linux devices.</p> <p>Find the platform of your device at "<a href="#">HMI devices capabilities</a>" on page 613</p>	int
<b>Flash Free Space</b>	Free space left in internal Flash memory.	uint64 read only
<b>Manufacturer Code</b>	Internal code that identify the HMI type	unsignedInt read only
<b>System Font List</b>	List of system fonts	string read only

Variable	Description	Data type
<b>System Mode</b>	Runtime operation status. <b>1</b> = booting <b>2</b> = configuration mode <b>3</b> = operating mode <b>4</b> = restart <b>5</b> = shutdown	int
<b>System UpTime</b>	Time the system has been powered since production of the unit (hours).	unsignedInt read only

## Dump information variables

Status of the copy process to external drives (USB or SD Card) for trend and event buffers.

Variable	Description	Data type
<b>Dump Error Message</b>	Return error message if any error occurs during the dump operation	string read only
<b>Dump Archive Status</b>	<b>0</b> = initial default state <b>1</b> = operation triggered <b>2</b> = operation complete successfully <b>3</b> = operation completed with errors	int read only
<b>Dump Recipe Status</b>	<b>0</b> = initial default state <b>1</b> = operation triggered <b>2</b> = operation complete successfully <b>3</b> = operation completed with errors	int read only
<b>Dump Trend Status</b>	<b>0</b> = initial default state <b>1</b> = operation triggered <b>2</b> = operation complete successfully <b>3</b> = operation completed with errors	int read only
<b>Reset Recipe Status</b>	<b>0</b> = initial default state <b>1</b> = operation triggered <b>2</b> = operation complete successfully <b>3</b> = operation completed with errors	int read only
<b>Restore Recipe Status</b>	<b>0</b> = initial default state <b>1</b> = operation triggered <b>2</b> = operation complete successfully <b>3</b> = operation completed with errors	int read only



## FTP client variables

The FTP client variables are updated when the FTP actions are used.

Variable	Description	Data type
<b>FTP Current Command</b>	Last FTP command	string read only
<b>FTP Error Message</b>	Last FTP error message	string read only
<b>FTP Progress</b>	Download/upload progress (0/100%)	short read only
<b>FTP Status</b>	Status of last FTP command: <ul style="list-style-type: none"> <li>• 0 = idle</li> <li>• 1 = active</li> <li>• 2 = done</li> <li>• 3 = error</li> </ul>	short read only

## Keypad variables

Keypad status.

Variable	Description	Data type
<b>Is keypad open</b>	0 = no keypad open 1 = keypad open	int read only

## Language variables

Keypad status.

Variable	Description	Data type
<b>Current Language Code</b>	Information on the active language	string read only
<b>Current Language Id</b>	Information on the active language	int read only
<b>Current Language Name</b>	Information on the active language	string read only

## Network variables

Device network parameters.

Variable	Description	Data type
<b>Adapters Parameters</b>	This is a JSON string that can be use to read or update the network adapters parameters	string
<b>Gateway</b>	Gateway address of the main Ethernet interface of device	string read only
<b>IP Address</b>	IP address of the main Ethernet interface of device	string read only
<b>Mac ID</b>	MAC ID of the main Ethernet interface of device	string read only
<b>Status</b>	Contains the result of the last operation required by writing inside the Adapter Parameters. It is updated after each write operation. <ul style="list-style-type: none"> <li>• Empty string is meaning no errors</li> <li>• Last error descriptions</li> </ul>	string read only
<b>Subnet Mask</b>	Subnet Mask of the main Ethernet interface of device	string read only

## Printing variables


Information on printing functions.

Variable	Description	Data type
<b>Completion percentage</b>	Percentage of completion of current print job. Range: 0–100	read only
<b>Current disk usage</b>	Folder size in bytes where PDF reports are stored. If <i>Flash</i> has been selected as <i>Spool media type</i> , this value corresponds to <i>reportspool</i> .	read only
<b>Current job</b>	Name of the report the job is processing. Current job is the following: <ul style="list-style-type: none"> <li>• [report name] for a <b>Graphic Report</b></li> </ul>	read only

Variable	Description	Data type
	<ul style="list-style-type: none"> <li>[first line of text] for a <b>Text Report</b></li> </ul>	
<b>Current RAM usage</b>	Size in bytes of the RAM used to process the current job	read only
<b>Disk quota</b>	Maximum size in bytes of the folder where PDF reports are stored	read only
<b>Graphic job queue size</b>	Number of available graphic jobs in the printing queue	read only
<b>Last error message</b>	Description of the last returned error	string read only
<b>RAM quota</b>	Maximum size in bytes of the RAM used to generate reports	read only
<b>Status</b>	Printing system status. Values: <ul style="list-style-type: none"> <li><b>idle</b></li> <li><b>error</b></li> <li><b>paused</b></li> <li><b>printing</b></li> </ul>	string read only
<b>Text job queue size</b>	Number of available text jobs in the printing queue	read only

## Remote Client variables

On remote clients, the below system variable can be used to know if the server (HMI device) is reachable.

Variable	Description	Data type
<b>Connection status</b>	<p><b>0</b> = client can not reach the server client. The connection with server is lost.</p> <p><b>1</b> = client can reach the server. The connection with server is active.</p> <p> This is only a client side variable. On HMI Runtime or PB610 PC Runtime it will be always 0.</p>	int (32 bit) read only

The following system variables are associated to the transferring files to a remote HMI device.

Variable	Description	Data type
<b>Download from HMI error message</b>	Error description	ASCII string read only
<b>Download from HMI percentage</b>	Download progress (0→100)	read only
<b>Download from HMI status</b>	<b>0</b> = idle, action is not in use or completed <b>1</b> = file download in progress <b>2</b> = error	int (32 bit) read only
<b>Upload to HMI error message</b>	Error description	ASCII string read only
<b>Upload to HMI percentage</b>	Upload progress (0→100)	read only
<b>Upload to HMI status</b>	<b>0</b> = idle, action is not in use or completed <b>1</b> = file upload in progress <b>2</b> = error	int (32 bit) read only

## Version variables

Operating System and runtime version.

Variable	Description	Data type
<b>Main OS Version</b>	Version of Main OS.	string
<b>Runtime Version</b>	Version of runtime.	string

## Screen variables

Screen status.

Variable	Description
<b>Time remaining to unlock</b>	Time remaining to unlock screen (see <b>LockScreen</b> action, " <a href="#">Page actions</a> " on page 204)
<b>X Screen resolution</b>	Display horizontal screen size in pixel
<b>Y Screen resolution</b>	Display vertical screen size in pixel

## SD card variables

Information on the external SD card.

Variable	Description	Data type
SD Card FreeSpace	Available space on card in bytes	uint64 read only
SD Card Name	Name of SD card	string read only
SD Card Size	Size in bytes of the card plugged in the slot	uint64 read only
SD Card Status	0 = SD card unplugged 1 = SD card plugged	int read only

## Server variables

Server status.



**Important: All variables refer to server, not to HMI Client.**

Variable	Description	Data type
Current page	Name of current page	string
Current project	Name of current project	string
Operating mode time	Seconds elapsed since device started operating mode as in <b>System Date</b> format (milliseconds).	uint64
Project load time	Date when the project was loaded on the HMI Runtime as in <b>System Date</b> format (milliseconds).	uint64

## Time variables

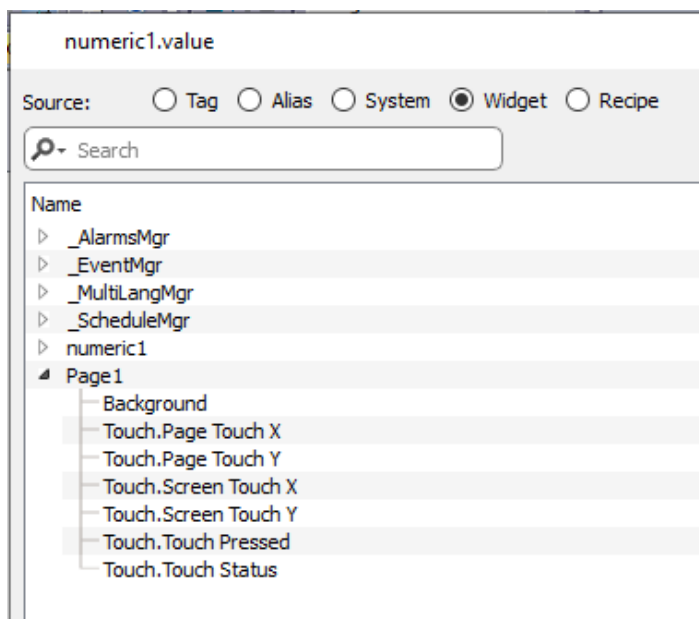
System time expressed in UTC format.

Variable	Description	Data type
Day Of Month	Range: 1–31	int
Day Of Week	Range: 0 = Sunday, .. , 6 = Saturday	int
Hour	Range: 0–23	int
Minute	Range: 0–59	int
Month	Range: 1–12	int

Variable	Description	Data type
<b>Second</b>	Range: 0–59	int
<b>System Time</b>	The same as UTC time. It can also be set as date/time for this variable.	unsignedInt
<b>Year</b>	Current Year	int

## Touch screen variables

Cursor status and position on the touchscreen. These are properties of the active page and can be selected in the **Widget** section.



Note: Page size can be different than HMI device display size.

Variable	Description	Java Script
<b>Page Touch X</b>	Cursor position related to page	page.primaryTouch.x
<b>Page Touch Y</b>		page.primaryTouch.y
<b>Screen Touch X</b>	Cursor position related touchscreen	page.primaryTouch.screenX
<b>Screen Touch Y</b>		page.primaryTouch.screenY

Variable	Description	Java Script
<b>Touch Press</b>	0 = screen not pressed 1 = screen pressed	page.primaryTouch.pressed
<b>Touch Status</b>	Generic touch screen changes. This variable contains the concatenation of <b>Screen Touch X</b> , <b>Screen Touch Y</b> and <b>Touch Press</b> values (for example, "924,129,0").  The main usage of this variable is to trigger an event, using the OnDataUpdate feature, when something (x, y or click) is changed.	page.primaryTouchStatus

## USB drive variables

Information on the external USB drive connected to the device.

Variable	Description	Data type
<b>USB Drive free space</b>	Available space in bytes	uint64 read only
<b>USB Drive Name</b>	Name of USB device	string read only
<b>USB Drive Size</b>	Size in bytes of the device plugged in the USB port	uint64 read only
<b>USB Drive Status</b>	0 = USB Drive unplugged 1 = USB Drive plugged	int read only

## User management variables

Information on users and groups.

Variable	Description	Data type
<b>This Client User-Name</b>	Name of the user logged to the client where the system variable is displayed.	string read only
<b>This Client Group-Name</b>	Group of currently logged user	string read only
<b>This Client ID</b>	Only for HMI Clients. Local and remote clients connected to the same server (for example, runtime) get a unique ID.	short

Variable	Description	Data type
		read only
<b>No Of Remote-Clients Alive</b>	Number of HMI Clients connected to the server	short read only

### JavaScript

From JavaScript, the variables can be accessed as properties of the `_SysPropMgr` object.

Example:

```
var sysVar = project.getWidget( "_SysPropMgr" );  
  
var UserName    = sysVar.getProperty("This Client User-Name");  
var UserGroup   = sysVar.getProperty("This Client Group-Name");  
var clientId    = sysVar.getProperty("This Client ID");  
var numClients  = sysVar.getProperty("No Of Remote-Clients Alive");
```





# 14 System Variables (Protocol)

---

System Variables communication driver allows to create Tags that point to system information.



System Variables communication driver is not counted as physical protocol.  
Refer to **Table of functions and limits** from main manual in "Number of physical protocols" line.

---

<b>Client System Variables</b> .....	<b>160</b>
<b>Protocol Editor Settings</b> .....	<b>160</b>
<b>Tag Import</b> .....	<b>160</b>
<b>Default variables</b> .....	<b>162</b>
<b>Retentive Memory variables</b> .....	<b>177</b>
<b>Services variables</b> .....	<b>182</b>
<b>NFC variables</b> .....	<b>184</b>
<b>PLCM09 variables</b> .....	<b>186</b>
<b>JSmart variables</b> .....	<b>190</b>

# Client System Variables

The "Client System Variables" protocol can be configured like the System Variables protocol, but there is an additional flag to indicate whether variables can only be written by the remote client. This means that each client can write their own values.

Tags defined using the Client System variables protocol will assume the value they have on the client instead of the value on the server. For example, "Page name", "IP Address", "Display Brightness", etc. will show different values on the server and on the remote clients.

When the "**Remote Only**" flag is set:

- Writing a tag of a "Client System Variables" protocol only affects the client running it.
- The JavaScript procedures associated with alarms and schedulers are performed on the clients and therefore any writes have an effect on the client variables (e.g. to turn the buzzer on or off from alarms).

For the descriptions of the variables refer to the documentation relating to the "[System Variables \(Protocol\)](#)" on the [previous page](#)



System Variables communication driver is not counted as physical protocol.  
Refer to **Table of functions and limits** from main manual in "Number of physical protocols" line.

## Protocol Editor Settings

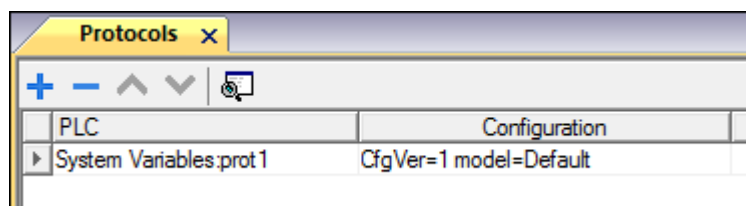
### Adding a protocol

To configure the protocol:

1. In **Config** node double-click **Protocols**.
2. To add a driver, click **+**: a new line is added.
3. Select the protocol from the **PLC** list.

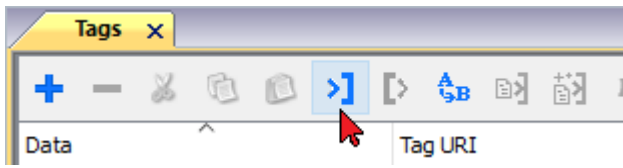
The protocol configuration dialog is displayed.

From PLC Model list select the specific System Variables type.



## Tag Import

Select the driver in Tag Editor and click on the **Import Tags** button to start the importer.

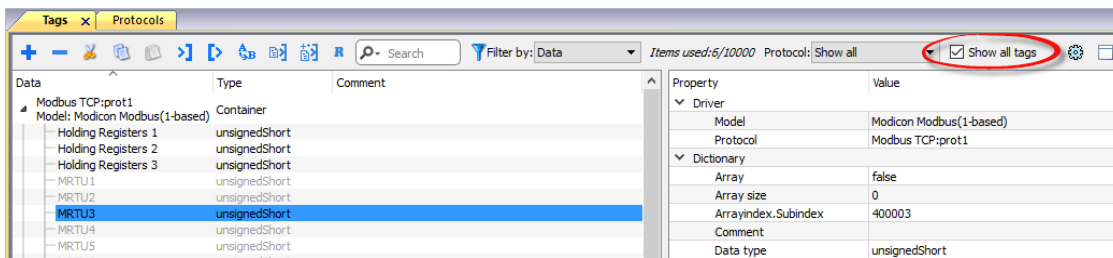





The system will require a generic XML file exported from Tag Editor by appropriate button.

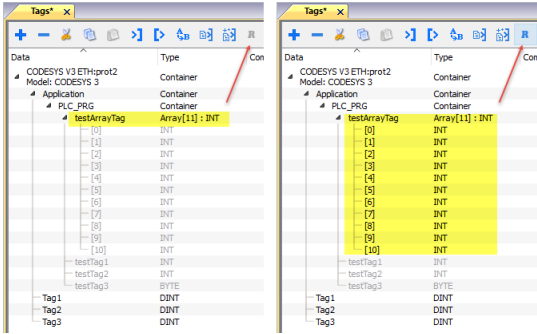



Once the importer has been selected, locate the symbol file and click **Open**.

The tags available within the Dictionary but not imported into the project are gray and are visible only when the "Show all tags" check box is selected.



Toolbar item	Description
	<p><b>Import Tag(s).</b></p> <p>Select tags to be imported and click on this icon to add tags from tag dictionary to the project</p>
	<p><b>Update Tag(s).</b></p> <p>Click on this icon to update the tags in the project, due a new dictionary import.</p>
	<p>Check this box to import all sub-elements of a tag.</p> <p>Example of both checked and unchecked result:</p>

Toolbar item	Description
	
	<p>Searches tags in the dictionary basing on filter combobox item selected.</p>

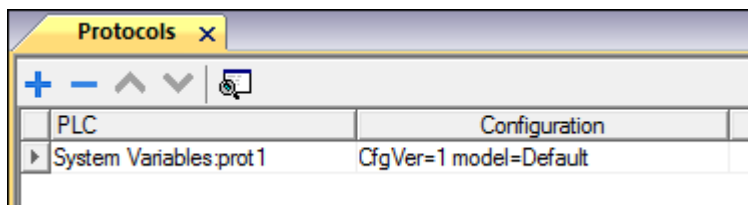
## Default variables

System Variables - Default protocol allows to create Tags that point to HMI system variables regarding:

- [Alarms](#)
- [Buzzer](#)
- [Communication](#)
- [Database](#)
- [Daylight Saving Time](#)
- [Device](#)
- [Dump information](#)
- [Network](#)
- [Screen](#)
- [SD Card](#)
- [Server](#)
- [Time](#)
- [USB Drive](#)
- [Version](#)
- [Virtual Com Switch](#)

### Protocol Editor Settings

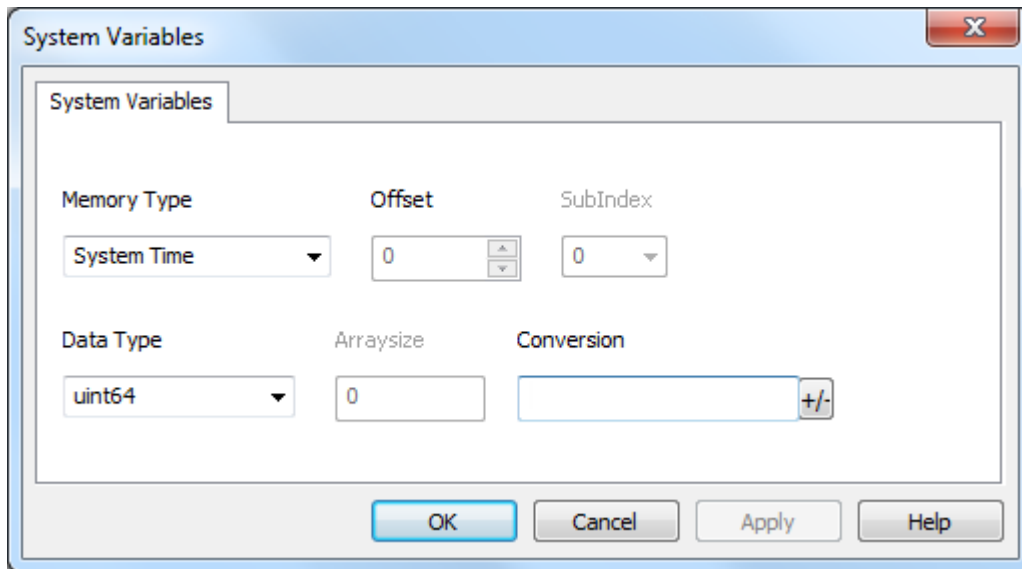
From PLC Model list of Protocol Editor dialog, select Default.




### Tag Editor Settings

Path: **ProjectView** > **Config** > double-click **Tags**

1. To add a tag, click **+**: a new line is added.
2. Select **System Variables** from the **Driver** list: tag definition dialog is displayed.







Element	Description																											
<b>Memory Type</b>	Represents the system variable to which the Tag refers to. The below section shows the full list of possible system variables, grouped by category.																											
	<b>Alarms Variables</b>																											
	<table border="1"> <thead> <tr> <th>Variable Name</th> <th>Description</th> <th>Data Type</th> </tr> </thead> <tbody> <tr> <td><b>Alarm not acknowledged</b></td> <td>True when alarms unacknowledged is pending (Not Triggered Not Acknowledged&lt;&gt;0) OR (Triggered Not Acknowledged&lt;&gt;0)</td> <td>boolean read only</td> </tr> <tr> <td><b>Alarm triggered</b></td> <td>True when at least one alarm is triggered (Triggered Acknowledged&lt;&gt;0) OR (Triggered &lt;&gt;0) OR (Triggered Not Acknowledged&lt;&gt;0)</td> <td>boolean read only</td> </tr> <tr> <td><b>Number of missed alarm events</b></td> <td>Alarms exceeding the event queue. Queue length is defined in the engineconfig.xml file.</td> <td>int read only</td> </tr> <tr> <td><b>Number of not triggered acknowledged</b></td> <td>Alarm condition no longer active; alarms already acknowledged</td> <td>int read only</td> </tr> <tr> <td><b>Number of not triggered not acknowledged</b></td> <td>Alarm condition no longer active; awaiting acknowledgment</td> <td>int read only</td> </tr> <tr> <td><b>Number of triggered acknowledged</b></td> <td>Alarm condition active; alarms already acknowledged</td> <td>int read only</td> </tr> <tr> <td><b>Number of triggered alarms</b></td> <td>Alarm active: acknowledgment not required</td> <td>int read only</td> </tr> <tr> <td><b>Number of triggered not acknowledged</b></td> <td>Alarm condition active; awaiting acknowledgment</td> <td>int read only</td> </tr> </tbody> </table>	Variable Name	Description	Data Type	<b>Alarm not acknowledged</b>	True when alarms unacknowledged is pending (Not Triggered Not Acknowledged<>0) OR (Triggered Not Acknowledged<>0)	boolean read only	<b>Alarm triggered</b>	True when at least one alarm is triggered (Triggered Acknowledged<>0) OR (Triggered <>0) OR (Triggered Not Acknowledged<>0)	boolean read only	<b>Number of missed alarm events</b>	Alarms exceeding the event queue. Queue length is defined in the engineconfig.xml file.	int read only	<b>Number of not triggered acknowledged</b>	Alarm condition no longer active; alarms already acknowledged	int read only	<b>Number of not triggered not acknowledged</b>	Alarm condition no longer active; awaiting acknowledgment	int read only	<b>Number of triggered acknowledged</b>	Alarm condition active; alarms already acknowledged	int read only	<b>Number of triggered alarms</b>	Alarm active: acknowledgment not required	int read only	<b>Number of triggered not acknowledged</b>	Alarm condition active; awaiting acknowledgment	int read only
	Variable Name	Description	Data Type																									
	<b>Alarm not acknowledged</b>	True when alarms unacknowledged is pending (Not Triggered Not Acknowledged<>0) OR (Triggered Not Acknowledged<>0)	boolean read only																									
	<b>Alarm triggered</b>	True when at least one alarm is triggered (Triggered Acknowledged<>0) OR (Triggered <>0) OR (Triggered Not Acknowledged<>0)	boolean read only																									
	<b>Number of missed alarm events</b>	Alarms exceeding the event queue. Queue length is defined in the engineconfig.xml file.	int read only																									
	<b>Number of not triggered acknowledged</b>	Alarm condition no longer active; alarms already acknowledged	int read only																									
	<b>Number of not triggered not acknowledged</b>	Alarm condition no longer active; awaiting acknowledgment	int read only																									
	<b>Number of triggered acknowledged</b>	Alarm condition active; alarms already acknowledged	int read only																									
<b>Number of triggered alarms</b>	Alarm active: acknowledgment not required	int read only																										
<b>Number of triggered not acknowledged</b>	Alarm condition active; awaiting acknowledgment	int read only																										




Element	Description		
	<b>Buzzer Variables</b>		
	<b>Variable Name</b>	<b>Description</b>	<b>Data Type</b>
	<b>Buzzer Setup</b>	<p><b>0</b> = disabled</p> <p><b>1</b> = enabled (buzzer sounds as audible on any touchscreen event)</p> <p><b>2</b> = buzzer status controlled by Buzzer Control system variable or by Buzzer on Touch property inside the "Project properties" of main manual</p> <p> <b>Buzzer on touchscreen (Setup=1) is not available on Linux platforms. See "Buzzer on Touch" property in alternative.</b></p>	int
	<b>Buzzer Control</b>	<p><b>0</b> = buzzer off</p> <p><b>1</b> = buzzer on</p> <p><b>2</b> = buzzer blink</p>	int
	<b>Buzzer Off Time</b>	Duration in milliseconds of off time when blink has been selected. Default = 1000. Range: 100–5000	int
	<b>Buzzer On Time</b>	Duration in milliseconds of on time when blink has been selected. Default = 1000. Range: 100–5000	int



Element	Description	
	<b>Communication Variables</b>	
	<b>Variable Name</b>	<b>Description</b>
	<b>Protocol Communication Status</b>	Summarize the status of the communication protocols.  <b>0</b> = No protocol running, protocol drivers might not have been properly downloaded to the HMI device  <b>1</b> = Protocols loaded and started, no communication error  <b>2</b> = At least one communication protocol is reporting an error
	<b>Protocol Error Message</b>	Communication error with error source.  For example: "[xxxx]" where "xxxx" is the protocol abbreviation, the error source.  Multiple acronyms appear in case of multiple error sources. Blank when no errors are reported.
	<b>Protocol Error Count</b>	Number of communication errors occurred since last reset. Reset value with Reset Protocol Error Count action, see "System actions" of main manual
	<b>Database Variables</b>	
	<b>Variable Name</b>	<b>Description</b>
	<b>Database link error message</b>	Last detected error description
	<b>Database link status</b>	<b>0</b> = Undefined (not yet initialized) <b>1</b> = OnLine (ready) <b>2</b> = OffLine (not available) <b>3</b> = Transfer in progress <b>4</b> = Error
	<b>Database link error count</b>	Errors counter. Increased after each error

Element	Description																																										
	<div data-bbox="331 309 400 376" style="float: left; margin-right: 10px;">  </div> <p data-bbox="443 309 1337 405">Each database variable is an array where index select the database link connection (Range 1-10) Variables are updated only when any database connector action is executed</p> <table border="1" data-bbox="320 450 1366 1921" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="3" data-bbox="320 450 1366 506" style="background-color: #e0e0e0;">Daylight Saving Time Variables</th> </tr> <tr> <th data-bbox="320 506 624 562">Variable Name</th> <th data-bbox="624 506 1206 562">Description</th> <th data-bbox="1206 506 1366 562">Data Type</th> </tr> </thead> <tbody> <tr> <td data-bbox="320 562 624 689"><b>Standard Offset</b></td> <td data-bbox="624 562 1206 689">Offset in minutes when standard time is set, with respect to GMT (for example: -8x60 = -480 minutes)</td> <td data-bbox="1206 562 1366 689">int read only</td> </tr> <tr> <td data-bbox="320 689 624 801"><b>Standard Week</b></td> <td data-bbox="624 689 1206 801">Week in which the standard time starts (for example: First = 1)</td> <td data-bbox="1206 689 1366 801">int read only</td> </tr> <tr> <td data-bbox="320 801 624 913"><b>Standard Month</b></td> <td data-bbox="624 801 1206 913">Month in which the standard time starts. Range: 0–11. (for example: November = 10)</td> <td data-bbox="1206 801 1366 913">int read only</td> </tr> <tr> <td data-bbox="320 913 624 1025"><b>Standard Day</b></td> <td data-bbox="624 913 1206 1025">Day of week in which the standard time starts (for example: Sunday = 0)</td> <td data-bbox="1206 913 1366 1025">int read only</td> </tr> <tr> <td data-bbox="320 1025 624 1137"><b>Standard Hour</b></td> <td data-bbox="624 1025 1206 1137">Hour in which the standard time starts (for example: 02 = 2)</td> <td data-bbox="1206 1025 1366 1137">int read only</td> </tr> <tr> <td data-bbox="320 1137 624 1249"><b>Standard Minute</b></td> <td data-bbox="624 1137 1206 1249">Minute in which the standard time starts (for example: 00 = 0)</td> <td data-bbox="1206 1137 1366 1249">int read only</td> </tr> <tr> <td data-bbox="320 1249 624 1361"><b>DST Offset</b></td> <td data-bbox="624 1249 1206 1361">Offset in minutes when DLS time is set, with respect to GMT</td> <td data-bbox="1206 1249 1366 1361">int read only</td> </tr> <tr> <td data-bbox="320 1361 624 1473"><b>DST Week</b></td> <td data-bbox="624 1361 1206 1473">Week in which the DLS time starts</td> <td data-bbox="1206 1361 1366 1473">int read only</td> </tr> <tr> <td data-bbox="320 1473 624 1585"><b>DST Month</b></td> <td data-bbox="624 1473 1206 1585">Month in which the DLS time starts. Range: 0–11</td> <td data-bbox="1206 1473 1366 1585">int read only</td> </tr> <tr> <td data-bbox="320 1585 624 1697"><b>DST Day</b></td> <td data-bbox="624 1585 1206 1697">Day of week in which the DLS time starts</td> <td data-bbox="1206 1585 1366 1697">int read only</td> </tr> <tr> <td data-bbox="320 1697 624 1809"><b>DST Hour</b></td> <td data-bbox="624 1697 1206 1809">Hour in which the DLS time starts</td> <td data-bbox="1206 1697 1366 1809">int read only</td> </tr> <tr> <td data-bbox="320 1809 624 1921"><b>DST Minute</b></td> <td data-bbox="624 1809 1206 1921">Minute in which the DLS time starts</td> <td data-bbox="1206 1809 1366 1921">int read only</td> </tr> </tbody> </table>	Daylight Saving Time Variables			Variable Name	Description	Data Type	<b>Standard Offset</b>	Offset in minutes when standard time is set, with respect to GMT (for example: -8x60 = -480 minutes)	int read only	<b>Standard Week</b>	Week in which the standard time starts (for example: First = 1)	int read only	<b>Standard Month</b>	Month in which the standard time starts. Range: 0–11. (for example: November = 10)	int read only	<b>Standard Day</b>	Day of week in which the standard time starts (for example: Sunday = 0)	int read only	<b>Standard Hour</b>	Hour in which the standard time starts (for example: 02 = 2)	int read only	<b>Standard Minute</b>	Minute in which the standard time starts (for example: 00 = 0)	int read only	<b>DST Offset</b>	Offset in minutes when DLS time is set, with respect to GMT	int read only	<b>DST Week</b>	Week in which the DLS time starts	int read only	<b>DST Month</b>	Month in which the DLS time starts. Range: 0–11	int read only	<b>DST Day</b>	Day of week in which the DLS time starts	int read only	<b>DST Hour</b>	Hour in which the DLS time starts	int read only	<b>DST Minute</b>	Minute in which the DLS time starts	int read only
Daylight Saving Time Variables																																											
Variable Name	Description	Data Type																																									
<b>Standard Offset</b>	Offset in minutes when standard time is set, with respect to GMT (for example: -8x60 = -480 minutes)	int read only																																									
<b>Standard Week</b>	Week in which the standard time starts (for example: First = 1)	int read only																																									
<b>Standard Month</b>	Month in which the standard time starts. Range: 0–11. (for example: November = 10)	int read only																																									
<b>Standard Day</b>	Day of week in which the standard time starts (for example: Sunday = 0)	int read only																																									
<b>Standard Hour</b>	Hour in which the standard time starts (for example: 02 = 2)	int read only																																									
<b>Standard Minute</b>	Minute in which the standard time starts (for example: 00 = 0)	int read only																																									
<b>DST Offset</b>	Offset in minutes when DLS time is set, with respect to GMT	int read only																																									
<b>DST Week</b>	Week in which the DLS time starts	int read only																																									
<b>DST Month</b>	Month in which the DLS time starts. Range: 0–11	int read only																																									
<b>DST Day</b>	Day of week in which the DLS time starts	int read only																																									
<b>DST Hour</b>	Hour in which the DLS time starts	int read only																																									
<b>DST Minute</b>	Minute in which the DLS time starts	int read only																																									

Element	Description	
	 All variables are read only: they cannot be used to update the system clock.	
<b>Device Variables</b>		
Variable Name	Description	Data Type
<b>Available System Memory</b>	Free available RAM memory in bytes	uint64 read only
<b>Backlight Time</b>	Activation time in hours of the display backlight since production of the device	unsignedInt read only
<b>Battery LED</b>	Enables/disables the low battery LED indicator (when available)  <b>0</b> = disabled <b>1</b> = enabled   Not available on Linux platforms (find the platform of your device at <a href="#">"HMI devices capabilities" on page 613</a> )	int
<b>Battery Timeout</b>	Reserved   Not available on Linux platforms (find the platform of your device at <a href="#">"HMI devices capabilities" on page 613</a> )	int
<b>Display Brightness</b>	Returns and adjusts brightness level.  When set to a low light level (0..3), the backlight stays lit to a higher level for 8 seconds to allow the user to make the adjustments and then is switched-off.  Even when set to 0, the backlight is still on and the Backlight Time counter increases. Range: 0–255	int
<b>External Timeout</b>	Non-operational time after which the display backlight is automatically turned off. The backlight is automatically turned on when the user touches the screen	int

Element	Description	
	<b>Device Variables</b>	
	<b>Variable Name</b>	<b>Description</b>
		<p><b>-1 =</b> Switch off backlight and disable touch (switch display off). <b>Backlight Time</b> counter is stopped.</p> <p> On Linux devices requires BSP v1.0.324 or higher.</p> <p><b>-2 =</b> Switch off backlight but not disable touch. If touch is pressed, event is not passed to applications but screen saver exit and backlight return on.</p> <p> Available only on Linux devices. Requires BSP v1.0.324 or higher.</p> <p><b>0 =</b> Switch backlight on (switch display on)</p> <p><b>1..n =</b> Timeout, in seconds, for switch off backlight (screen saver timer)</p> <p> The timeout value is rounded to multiples of one minute (60, 120, 180, etc.) in all Linux devices.</p> <p>Find the platform of your device at "<a href="#">HMI devices capabilities</a>" on page 613</p>
	<b>Flash Free Space</b>	Free space left in internal Flash memory
		uint64 read only
	<b>Manufacturer Code</b>	Code number that identifies the HMI
		short read only
	<b>Server RAM Usage</b>	Current RAM memory used from HMI, expressed in byte
		uint64 read only

Element	Description	
	<b>Device Variables</b>	
	<b>Variable Name</b>	<b>Description</b>
	<b>System Font List</b>	List of system fonts string read only
	<b>System Mode</b>	Runtime operation status 1 = booting 2 = configuration mode 3 = operating mode 4 = restart 5 = shutdown int
	<b>System UpTime</b>	Time the system has been powered since production of the unit (hours) unsignedInt read only
	<b>Dump information Variables</b>	
	<b>Variable Name</b>	<b>Description</b>
	<b>Dump Error Message</b>	Return error message if any error occurs during the dump operation string read only
	<b>Dump Archive Status</b>	0 = initial default state 1 = operation triggered 2 = operation complete successfully 3 = operation completed with errors int read only
	<b>Dump Recipe Status</b>	0 = initial default state 1 = operation triggered 2 = operation complete successfully 3 = operation completed with errors int read only
	<b>Dump Trend Status</b>	0 = initial default state 1 = operation triggered 2 = operation complete successfully 3 = operation completed with errors int read only

Element	Description		
	<b>Dump information Variables</b>		
	<b>Variable Name</b>	<b>Description</b>	<b>Data Type</b>
	<b>Reset Recipe Status</b>	<b>0</b> = initial default state <b>1</b> = operation triggered <b>2</b> = operation complete successfully <b>3</b> = operation completed with errors	int read only
	<b>Restore Recipe Status</b>	<b>0</b> = initial default state <b>1</b> = operation triggered <b>2</b> = operation complete successfully <b>3</b> = operation completed with errors	int read only
	<b>Network Variables</b>		
	<b>Variable Name</b>	<b>Description</b>	<b>Data Type</b>
	<b>Gateway</b>	Gateway address of the main Ethernet interface of HMI	string read only
	<b>IP Address</b>	IP address of the main Ethernet interface of HMI	string read only
	<b>Mac ID</b>	MAC ID of the main Ethernet interface of HMI	string read only
	<b>Network Adapter Parameters</b>	JSON string that can be use to read or update the network adapters parameters	string
	<b>Network Status</b>	Contains the result of the last operation required by writing inside the Adapter Parameters. It is updated after each write operation. <ul style="list-style-type: none"> <li>• Empty string is meaning no errors</li> <li>• Last error descriptions</li> </ul>	string read only
	<b>Subnet Mask</b>	Subnet Mask of the main Ethernet interface of HMI	string read only

Element	Description	
	<b>Screen Variables</b>	
	<b>Variable Name</b>	<b>Description</b>
	<b>X Screen resolution</b>	Display horizontal screen size in pixel int read only
	<b>Y Screen resolution</b>	Display vertical screen size in pixel int read only
	<b>SD Card Variables</b>	
	<b>Variable Name</b>	<b>Description</b>
	<b>SD Card FreeSpace</b>	Available space on card in bytes uint64 read only
	<b>SD Card Name</b>	Name of SD card string read only
	<b>SD Card Size</b>	Size in bytes of the card plugged in the slot uint64 read only
	<b>SD Card Status</b>	0 = SD card unplugged 1 = SD card plugged int read only
	<b>Server Variables</b>	
	<b>Variable Name</b>	<b>Description</b>
	<b>Page name</b>	Name of current page string read only
	<b>Project Name</b>	Name of current project string read only
	<b>Project load time</b>	Date when the project was loaded on the HMI Runtime as in System Date format (milliseconds) uint64 read only
	<b>Last operating mode start time</b>	Seconds elapsed since device started operating mode uint64 read only

Element	Description																																										
	<div style="display: flex; align-items: center; margin-bottom: 10px;"> <p>All variables refer to server, not to HMI Client.</p> </div> <div style="background-color: #cccccc; padding: 5px; margin-bottom: 10px;"><b>Time Variables</b></div> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="background-color: #cccccc;">Variable Name</th> <th style="background-color: #cccccc;">Description</th> <th style="background-color: #cccccc;">Data Type</th> </tr> </thead> <tbody> <tr> <td><b>Day Of Month</b></td> <td>Range: 1–31</td> <td>int</td> </tr> <tr> <td><b>Day Of Week</b></td> <td>Range: 0 = Sunday, .. , 6 = Saturday</td> <td>int</td> </tr> <tr> <td><b>Hour</b></td> <td>Range: 0–23</td> <td>int</td> </tr> <tr> <td><b>Minute</b></td> <td>Range: 0–59</td> <td>int</td> </tr> <tr> <td><b>Month</b></td> <td>Range: 1–12</td> <td>int</td> </tr> <tr> <td><b>Second</b></td> <td>Range: 0–59</td> <td>int</td> </tr> <tr> <td><b>System Time</b></td> <td>The same as UTC time. It can also be set as date/time for this variable</td> <td>unsignedInt</td> </tr> <tr> <td><b>Year</b></td> <td>Current Year</td> <td>int</td> </tr> </tbody> </table> <div style="display: flex; align-items: center; margin-bottom: 10px;"> <p>System time expressed in UTC format</p> </div> <div style="background-color: #cccccc; padding: 5px; margin-bottom: 10px;"><b>USB Drive Variables</b></div> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="background-color: #cccccc;">Variable Name</th> <th style="background-color: #cccccc;">Description</th> <th style="background-color: #cccccc;">Data Type</th> </tr> </thead> <tbody> <tr> <td><b>USB Drive FreeSpace</b></td> <td>Available space in bytes</td> <td>uint64 read only</td> </tr> <tr> <td><b>USB Drive Name</b></td> <td>Name of USB device</td> <td>string read only</td> </tr> <tr> <td><b>USB Drive Size</b></td> <td>Size in bytes of the device plugged in the USB port</td> <td>uint64 read only</td> </tr> <tr> <td><b>USB Drive Status</b></td> <td><b>0</b> = USB Drive unplugged <b>1</b> = USB Drive plugged</td> <td>int read only</td> </tr> </tbody> </table>	Variable Name	Description	Data Type	<b>Day Of Month</b>	Range: 1–31	int	<b>Day Of Week</b>	Range: 0 = Sunday, .. , 6 = Saturday	int	<b>Hour</b>	Range: 0–23	int	<b>Minute</b>	Range: 0–59	int	<b>Month</b>	Range: 1–12	int	<b>Second</b>	Range: 0–59	int	<b>System Time</b>	The same as UTC time. It can also be set as date/time for this variable	unsignedInt	<b>Year</b>	Current Year	int	Variable Name	Description	Data Type	<b>USB Drive FreeSpace</b>	Available space in bytes	uint64 read only	<b>USB Drive Name</b>	Name of USB device	string read only	<b>USB Drive Size</b>	Size in bytes of the device plugged in the USB port	uint64 read only	<b>USB Drive Status</b>	<b>0</b> = USB Drive unplugged <b>1</b> = USB Drive plugged	int read only
Variable Name	Description	Data Type																																									
<b>Day Of Month</b>	Range: 1–31	int																																									
<b>Day Of Week</b>	Range: 0 = Sunday, .. , 6 = Saturday	int																																									
<b>Hour</b>	Range: 0–23	int																																									
<b>Minute</b>	Range: 0–59	int																																									
<b>Month</b>	Range: 1–12	int																																									
<b>Second</b>	Range: 0–59	int																																									
<b>System Time</b>	The same as UTC time. It can also be set as date/time for this variable	unsignedInt																																									
<b>Year</b>	Current Year	int																																									
Variable Name	Description	Data Type																																									
<b>USB Drive FreeSpace</b>	Available space in bytes	uint64 read only																																									
<b>USB Drive Name</b>	Name of USB device	string read only																																									
<b>USB Drive Size</b>	Size in bytes of the device plugged in the USB port	uint64 read only																																									
<b>USB Drive Status</b>	<b>0</b> = USB Drive unplugged <b>1</b> = USB Drive plugged	int read only																																									



Element	Description		
	<b>Version Variables</b>		
	<b>Variable Name</b>	<b>Description</b>	<b>Data Type</b>
	<b>Main OS version</b>	Version of Main OS	string read only
	<b>Runtime version</b>	Version of Runtime	string read only
	<b>Project name</b>	Project name	string read only
	<b>Project version</b>	Project version	string
	<b>Project GUID</b>	Project GUID (unique identifier)	string read only
	<b>Virtual Com Switch Variables</b>		
	<b>Variable Name</b>	<b>Description</b>	<b>Data Type</b>
	<b>VCS status</b>	Provides status of VCS service. <b>0</b> = Service enabled <b>1</b> = Client connected in interleaved mode <b>2</b> = Client connected in exclusive mode <b>3</b> = Service disabled (default)	unsignedByte read only
	<b>VCS disable</b>	Provides manual override of VCS service. <b>0</b> = VCS service enabled <b>1</b> = VCS service disabled (default)	boolean
	<b>VCS port</b>	Provides current listening TCP port on HMI by VCS service	unsignedShort
<b>Data Type</b>	Each system variable has a specific data type, described in above tables. The following table shows the details of any data type used for system variables.		

Element	Description																								
	<table border="1"> <thead> <tr> <th>Data Type</th> <th>Memory Space</th> <th>Limits</th> </tr> </thead> <tbody> <tr> <td><b>short</b></td> <td>16-bit data</td> <td>-32768 ... 32767</td> </tr> <tr> <td><b>int</b></td> <td>32-bit data</td> <td>-2.1e9 ... 2.1e9</td> </tr> <tr> <td><b>unsignedByte</b></td> <td>8-bit data</td> <td>0 ... 255</td> </tr> <tr> <td><b>unsignedShort</b></td> <td>16-bit data</td> <td>0 ... 65535</td> </tr> <tr> <td><b>unsignedInt</b></td> <td>32-bit data</td> <td>0 ... 4.2e9</td> </tr> <tr> <td><b>uint64</b></td> <td>64-bit data</td> <td>0 ... 1.8e19</td> </tr> <tr> <td><b>string</b></td> <td colspan="2">Array of elements containing character code defined by selected encoding</td> </tr> </tbody> </table>	Data Type	Memory Space	Limits	<b>short</b>	16-bit data	-32768 ... 32767	<b>int</b>	32-bit data	-2.1e9 ... 2.1e9	<b>unsignedByte</b>	8-bit data	0 ... 255	<b>unsignedShort</b>	16-bit data	0 ... 65535	<b>unsignedInt</b>	32-bit data	0 ... 4.2e9	<b>uint64</b>	64-bit data	0 ... 1.8e19	<b>string</b>	Array of elements containing character code defined by selected encoding	
Data Type	Memory Space	Limits																							
<b>short</b>	16-bit data	-32768 ... 32767																							
<b>int</b>	32-bit data	-2.1e9 ... 2.1e9																							
<b>unsignedByte</b>	8-bit data	0 ... 255																							
<b>unsignedShort</b>	16-bit data	0 ... 65535																							
<b>unsignedInt</b>	32-bit data	0 ... 4.2e9																							
<b>uint64</b>	64-bit data	0 ... 1.8e19																							
<b>string</b>	Array of elements containing character code defined by selected encoding																								

**Arraysizes**  
 In case of string Tag, this property represents the maximum number of bytes available in the string Tag.  
 Note: number of bytes corresponds to number of string chars if Encoding property is set to UTF-8 or Latin1 in Tag Editor.  
 If Encoding property is set to UCS-2BE, UCS-2LE, UTF-16BE or UTF-16LE one char requires 2 bytes.

**Conversion**  
 Conversion to be applied to the tag.

Depending on data type selected, the list **Allowed** shows one or more conversion types.

Value	Description
<b>Inv bits</b>	<p><b>inv</b>: Invert all the bits of the tag.</p> <p><i>Example:</i>                      1001 → 0110 (in binary format)                      9 → 6 (in decimal format)</p>
<b>Negate</b>	<b>neg</b> : Set the opposite of tag value.

Element	Description														
	<table border="1"> <thead> <tr> <th>Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td></td> <td><i>Example:</i> 25.36 → -25.36</td> </tr> <tr> <td><b>AB -&gt; BA</b></td> <td><b>swapnibbles:</b> Swap nibbles in a byte.  <i>Example:</i> 15D4 → 514D (in hexadecimal format) 5588 → 20813 (in decimal format)</td> </tr> <tr> <td><b>ABCD -&gt; CDAB</b></td> <td><b>swap2:</b> Swap bytes in a word.  <i>Example:</i> 9ACC → CC9A (in hexadecimal format) 39628 → 52378 (in decimal format)</td> </tr> <tr> <td><b>ABCDEFGH -&gt; GHEFCDAB</b></td> <td><b>swap4:</b> Swap bytes in a double word.  <i>Example:</i> 32FCFF54 → 54FFFC32 (in hexadecimal format) 855441236 → 1426062386 (in decimal format)</td> </tr> <tr> <td><b>ABC...NOP -&gt; OPM...DAB</b></td> <td><b>swap8:</b> Swap bytes in a long word.  <i>Example:</i> 142.366 → -893553517.588905 (in decimal format) 0 10000000110 000111001011101101100100010110100001110010101100 0001 → 1 10000011100 101010100001010001011011011011001011011000010011 1101 (in binary format)</td> </tr> <tr> <td><b>BCD</b></td> <td><b>bcd:</b> Separate byte in two nibbles, read them as decimal (from 0 to 9)  <i>Example:</i> 23 → 17 (in decimal format) 0001 0111 = 23 0001 = 1 (first nibble) 0111 = 7 (second nibble)</td> </tr> </tbody> </table>	Value	Description		<i>Example:</i> 25.36 → -25.36	<b>AB -&gt; BA</b>	<b>swapnibbles:</b> Swap nibbles in a byte.  <i>Example:</i> 15D4 → 514D (in hexadecimal format) 5588 → 20813 (in decimal format)	<b>ABCD -&gt; CDAB</b>	<b>swap2:</b> Swap bytes in a word.  <i>Example:</i> 9ACC → CC9A (in hexadecimal format) 39628 → 52378 (in decimal format)	<b>ABCDEFGH -&gt; GHEFCDAB</b>	<b>swap4:</b> Swap bytes in a double word.  <i>Example:</i> 32FCFF54 → 54FFFC32 (in hexadecimal format) 855441236 → 1426062386 (in decimal format)	<b>ABC...NOP -&gt; OPM...DAB</b>	<b>swap8:</b> Swap bytes in a long word.  <i>Example:</i> 142.366 → -893553517.588905 (in decimal format) 0 10000000110 000111001011101101100100010110100001110010101100 0001 → 1 10000011100 101010100001010001011011011011001011011000010011 1101 (in binary format)	<b>BCD</b>	<b>bcd:</b> Separate byte in two nibbles, read them as decimal (from 0 to 9)  <i>Example:</i> 23 → 17 (in decimal format) 0001 0111 = 23 0001 = 1 (first nibble) 0111 = 7 (second nibble)
Value	Description														
	<i>Example:</i> 25.36 → -25.36														
<b>AB -&gt; BA</b>	<b>swapnibbles:</b> Swap nibbles in a byte.  <i>Example:</i> 15D4 → 514D (in hexadecimal format) 5588 → 20813 (in decimal format)														
<b>ABCD -&gt; CDAB</b>	<b>swap2:</b> Swap bytes in a word.  <i>Example:</i> 9ACC → CC9A (in hexadecimal format) 39628 → 52378 (in decimal format)														
<b>ABCDEFGH -&gt; GHEFCDAB</b>	<b>swap4:</b> Swap bytes in a double word.  <i>Example:</i> 32FCFF54 → 54FFFC32 (in hexadecimal format) 855441236 → 1426062386 (in decimal format)														
<b>ABC...NOP -&gt; OPM...DAB</b>	<b>swap8:</b> Swap bytes in a long word.  <i>Example:</i> 142.366 → -893553517.588905 (in decimal format) 0 10000000110 000111001011101101100100010110100001110010101100 0001 → 1 10000011100 101010100001010001011011011011001011011000010011 1101 (in binary format)														
<b>BCD</b>	<b>bcd:</b> Separate byte in two nibbles, read them as decimal (from 0 to 9)  <i>Example:</i> 23 → 17 (in decimal format) 0001 0111 = 23 0001 = 1 (first nibble) 0111 = 7 (second nibble)														
	<p>Select conversion and click +. The selected item will be added to list <b>Configured</b>.</p> <p>If more conversions are configured, they will be applied in order (from top to bottom of list <b>Configured</b>).</p> <p>Use the arrow buttons to order the configured conversions.</p>														

## Retentive Memory variables

System Variables - Retentive Memory protocol allows to create Tags that point to a memory area whose content is maintained when HMI is powered off.

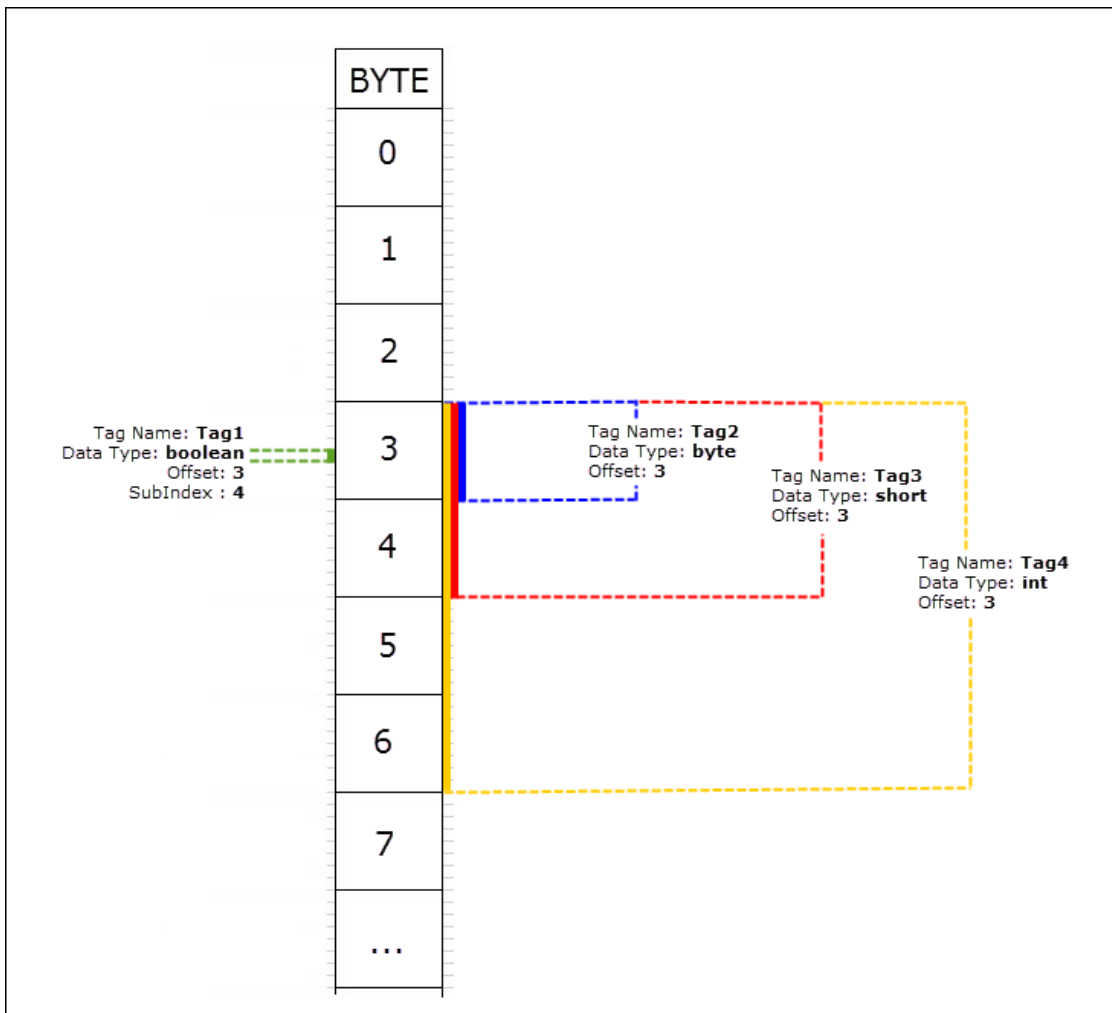
The physical support for retentive memory is based on FRAM technology.



**Important: Not all HMI devices include FRAM memory. If FRAM memory is not available, persistency is supported using user memory storage (Flash or hard disk drive). Flash technology has a limitation in the maximum number of write operations. The use of Flash as storage media for retentive memory with frequent write operations may damage the memory components. Check the HMI device datasheet for the availability of FRAM memory.**



**Important: Retentive memory is 16 KB flat memory area organized in bytes and accessible through an offset. Refer to schema below.**

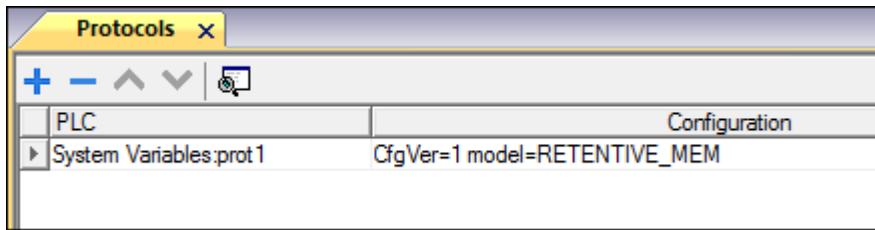


### Retentive memory vs. recipes storage

Recipe data is saved in flash memory (except for PB610 PC Runtime) while retentive data is saved in a FRAM. Flash memory is not suitable for a high number of write operations, while FRAM supports a virtually unlimited number of write operations and should be preferred when frequent write operations are required.

## Protocol Editor Settings

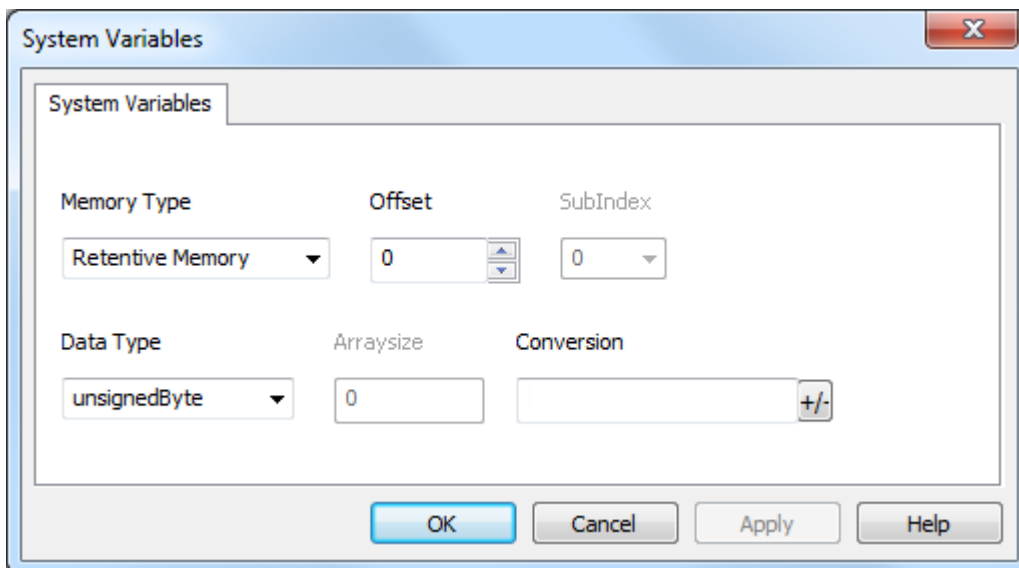
From PLC Model list of Protocol Editor dialog, select Retentive Memory.




## Tag Editor Settings

Path: **ProjectView** > **Config** > double-click **Tags**

1. To add a tag, click **+**: a new line is added.
2. Select **System Variables** from the **Driver** list: tag definition dialog is displayed.



Element	Description												
<b>Memory Type</b>	Fixed to Retentive Memory												
<b>Offset</b>	Offset address where tag is located. Range: 0-16383												
<b>SubIndex</b>	This parameter allows resource offset selection based on selected Data Type												
<b>Data Type</b>	<table border="1"> <thead> <tr> <th>Data Type</th> <th>Memory Space</th> <th>Limits</th> </tr> </thead> <tbody> <tr> <td><b>boolean</b></td> <td>1-bit data</td> <td>0 ... 1</td> </tr> <tr> <td><b>byte</b></td> <td>8-bit data</td> <td>-128 ... 127</td> </tr> <tr> <td><b>short</b></td> <td>16-bit data</td> <td>-32768 ... 32767</td> </tr> </tbody> </table>	Data Type	Memory Space	Limits	<b>boolean</b>	1-bit data	0 ... 1	<b>byte</b>	8-bit data	-128 ... 127	<b>short</b>	16-bit data	-32768 ... 32767
	Data Type	Memory Space	Limits										
	<b>boolean</b>	1-bit data	0 ... 1										
	<b>byte</b>	8-bit data	-128 ... 127										
<b>short</b>	16-bit data	-32768 ... 32767											

Element	Description																																	
	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="background-color: #cccccc;">Data Type</th> <th style="background-color: #cccccc;">Memory Space</th> <th style="background-color: #cccccc;">Limits</th> </tr> </thead> <tbody> <tr> <td><b>int</b></td> <td>32-bit data</td> <td>-2.1e9 ... 2.1e9</td> </tr> <tr> <td><b>int64</b></td> <td>64-bit data</td> <td>-9.2e18 ... 9.2e18</td> </tr> <tr> <td><b>unsignedByte</b></td> <td>8-bit data</td> <td>0 ... 255</td> </tr> <tr> <td><b>unsignedShort</b></td> <td>16-bit data</td> <td>0 ... 65535</td> </tr> <tr> <td><b>unsignedInt</b></td> <td>32-bit data</td> <td>0 ... 4.2e9</td> </tr> <tr> <td><b>uint64</b></td> <td>64-bit data</td> <td>0 ... 1.8e19</td> </tr> <tr> <td><b>float</b></td> <td>IEEE single-precision 32-bit floating point type</td> <td>1.17e-38 ... 3.4e38</td> </tr> <tr> <td><b>double</b></td> <td>IEEE double-precision 64-bit floating point type</td> <td>2.2e-308 ... 1.79e308</td> </tr> <tr> <td><b>string</b></td> <td colspan="2">Array of elements containing character code defined by selected encoding</td> </tr> <tr> <td><b>binary</b></td> <td colspan="2">Arbitrary binary data</td> </tr> </tbody> </table> <p> Note: to define arrays. select one of Data Type format followed by square brackets like "byte[]", "short[]"...</p>	Data Type	Memory Space	Limits	<b>int</b>	32-bit data	-2.1e9 ... 2.1e9	<b>int64</b>	64-bit data	-9.2e18 ... 9.2e18	<b>unsignedByte</b>	8-bit data	0 ... 255	<b>unsignedShort</b>	16-bit data	0 ... 65535	<b>unsignedInt</b>	32-bit data	0 ... 4.2e9	<b>uint64</b>	64-bit data	0 ... 1.8e19	<b>float</b>	IEEE single-precision 32-bit floating point type	1.17e-38 ... 3.4e38	<b>double</b>	IEEE double-precision 64-bit floating point type	2.2e-308 ... 1.79e308	<b>string</b>	Array of elements containing character code defined by selected encoding		<b>binary</b>	Arbitrary binary data	
Data Type	Memory Space	Limits																																
<b>int</b>	32-bit data	-2.1e9 ... 2.1e9																																
<b>int64</b>	64-bit data	-9.2e18 ... 9.2e18																																
<b>unsignedByte</b>	8-bit data	0 ... 255																																
<b>unsignedShort</b>	16-bit data	0 ... 65535																																
<b>unsignedInt</b>	32-bit data	0 ... 4.2e9																																
<b>uint64</b>	64-bit data	0 ... 1.8e19																																
<b>float</b>	IEEE single-precision 32-bit floating point type	1.17e-38 ... 3.4e38																																
<b>double</b>	IEEE double-precision 64-bit floating point type	2.2e-308 ... 1.79e308																																
<b>string</b>	Array of elements containing character code defined by selected encoding																																	
<b>binary</b>	Arbitrary binary data																																	
<b>Arraysize</b>	<ul style="list-style-type: none"> <li>In case of array tag, this property represents the number of array elements.</li> <li>In case of string tag, this property represents the maximum number of bytes available in the string tag.</li> </ul> <p>Note: number of bytes corresponds to number of string characters if Encoding property is set to UTF-8 or Latin1 in Tag Editor. If Encoding property is set to UCS-2BE, UCS-2LE, UTF-16BE or UTF-16LE one character requires 2 bytes.</p>																																	
<b>Conversion</b>	<p>Conversion to be applied to the tag.</p> <div style="border: 1px solid #ccc; padding: 5px;"> <p>Conversion</p> <p>inv,swap2</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="background-color: #cccccc;">Allowed</th> <th style="background-color: #cccccc;">Configured</th> </tr> </thead> <tbody> <tr> <td>BCD</td> <td>Inv bits</td> </tr> <tr> <td>AB-&gt;BA</td> <td>ABCD-&gt;CDAB</td> </tr> <tr> <td>ABCD-&gt;CDAB</td> <td></td> </tr> <tr> <td>ABCDEFGH-&gt;GHEFCBAB</td> <td></td> </tr> <tr> <td>Inv bits</td> <td></td> </tr> </tbody> </table> <p style="text-align: right;"> <input type="button" value="Cancel"/> <input type="button" value="OK"/> </p> </div>	Allowed	Configured	BCD	Inv bits	AB->BA	ABCD->CDAB	ABCD->CDAB		ABCDEFGH->GHEFCBAB		Inv bits																						
Allowed	Configured																																	
BCD	Inv bits																																	
AB->BA	ABCD->CDAB																																	
ABCD->CDAB																																		
ABCDEFGH->GHEFCBAB																																		
Inv bits																																		

Element	Description
	Depending on data type selected, the list <b>Allowed</b> shows one or more conversion types.
Value	Description
<b>Inv bits</b>	<b>inv</b> : Invert all the bits of the tag. <i>Example:</i> 1001 → 0110 (in binary format) 9 → 6 (in decimal format)
<b>Negate</b>	<b>neg</b> : Set the opposite of tag value. <i>Example:</i> 25.36 → -25.36
<b>AB -&gt; BA</b>	<b>swapnibbles</b> : Swap nibbles in a byte. <i>Example:</i> 15D4 → 514D (in hexadecimal format) 5588 → 20813 (in decimal format)
<b>ABCD -&gt; CDAB</b>	<b>swap2</b> : Swap bytes in a word. <i>Example:</i> 9ACC → CC9A (in hexadecimal format) 39628 → 52378 (in decimal format)
<b>ABCDEFGH -&gt; GHEFC DAB</b>	<b>swap4</b> : Swap bytes in a double word. <i>Example:</i> 32FCFF54 → 54FFFC32 (in hexadecimal format) 855441236 → 1426062386 (in decimal format)
<b>ABC...NOP -&gt; OPM...DAB</b>	<b>swap8</b> : Swap bytes in a long word. <i>Example:</i> 142.366 → -893553517.588905 (in decimal format) 0 10000000110 0001110010111011011001000101101000011100101011000001 → 1 10000011100 1010101000010100010110110110110010110110000100111101 (in binary format)
<b>BCD</b>	<b>bcd</b> : Separate byte in two nibbles, read them as decimal (from 0 to 9) <i>Example:</i> 23 → 17 (in decimal format) 0001 0111 = 23 0001 = 1 (first nibble) 0111 = 7 (second nibble)

Element	Description
	<p>Select conversion and click +. The selected item will be added to list <b>Configured</b>.</p> <p>If more conversions are configured, they will be applied in order (from top to bottom of list <b>Configured</b>).</p> <p>Use the arrow buttons to order the configured conversions.</p>

## Cleaning Retentive Memory

Use the **ClearRetentiveMemory** action to clear the content of the retentive memory.



Tip: Use this action to set the memory content to a known status at any time.

See *Actions > Tag Actions* section of main manual for more details.

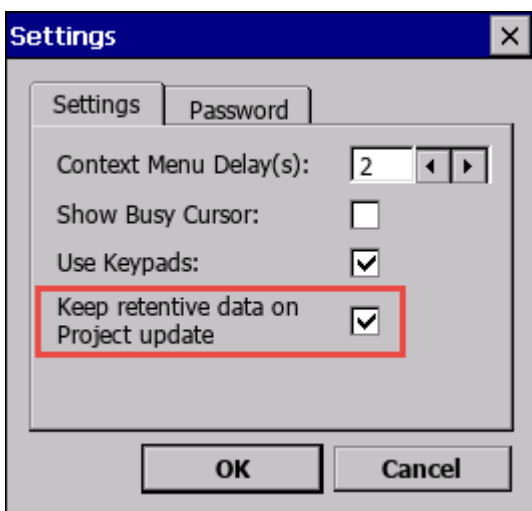


JavaScript interface for this action is:  
`project.clearRetentiveMemory();`

## Preserving Retentive Memory at project download

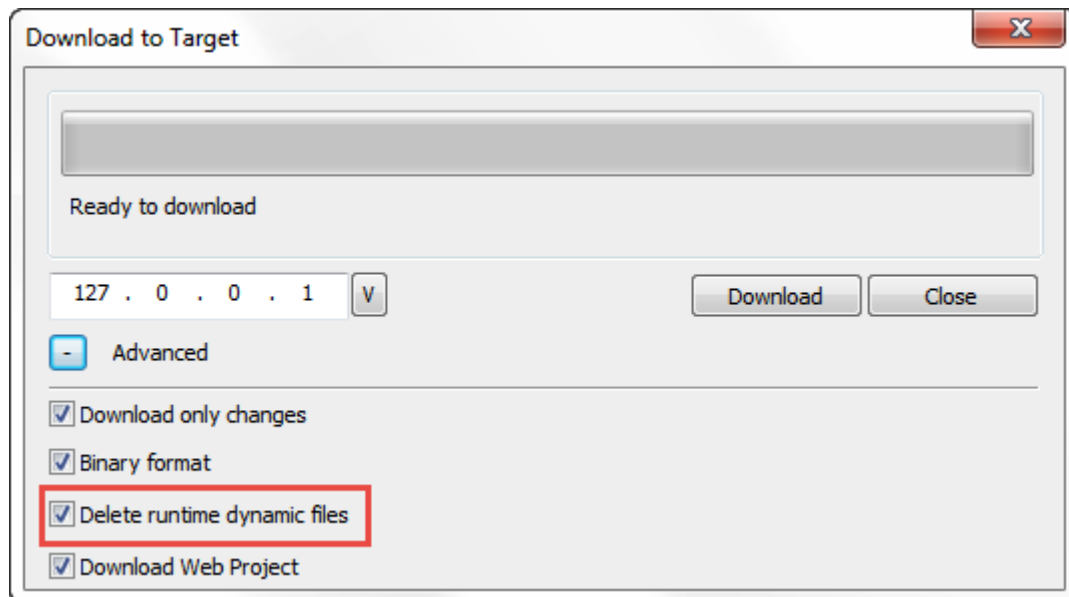
When a project file is downloaded to an HMI, or when the active project is modified, the content of retentive memory is usually deleted.

If is needed to preserve the content of retentive data at project download or update, select the **Keep retentive data on project update** option in the settings tabs of the HMI device.



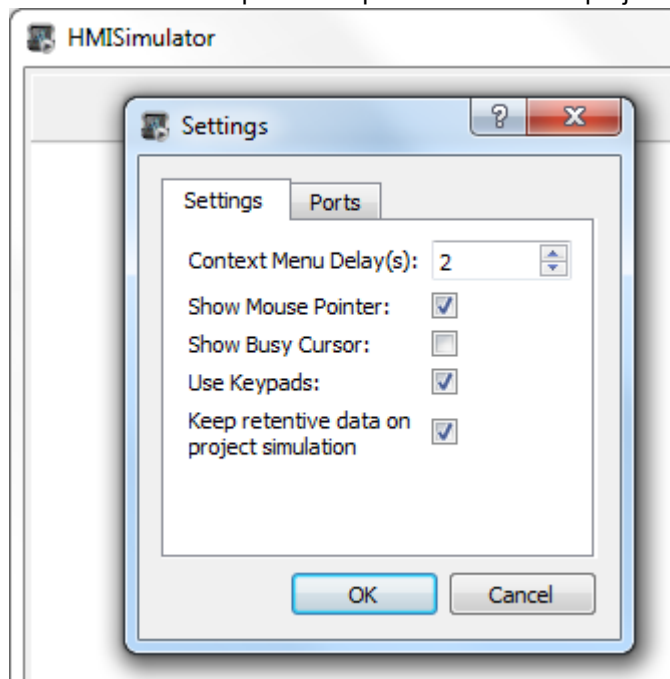
This setting will be ignored if **Delete runtime dynamic files** option is selected from *Download to Target* window.





### Preserving Retentive Memory in Simulator

Simulator of PB610 Panel Builder 600 supports the retentive memory. To enable retentive memory during project simulation use the option "Keep retentive data on project simulation" in context menu.



## Services variables

Services variables give the possibility to read the status and delivering commands to VNC Server.

### Protocol Editor Settings

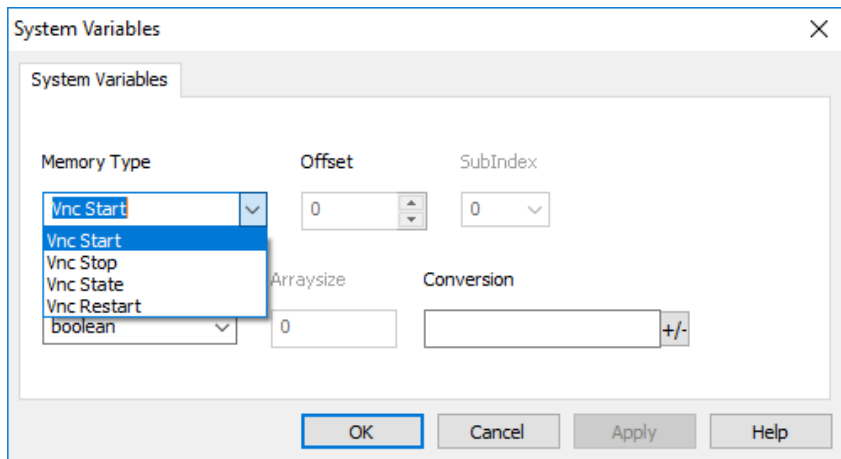
*Path: ProjectView > Protocols*

1. Click **+** and select **System Variables**: the **System Variables** dialog is displayed.
2. Select **Services** from the **PLCModels** list.




### Tag Editor Settings

Path: **ProjectView**> **Config** > double-click **Tags**

1. To add a tag, click **+**: a new line is added.
2. Select **System Variables** from the **Driver** list: tag definition dialog is displayed.



VNC status variables are supported only on Linux devices (See "[HMI devices capabilities](#)" on page 613) - BSP version 1.0.344 or greater is required.

Element	Description	Data type
<b>VNC Start</b>	Write 1 inside this tag to force the VNC server to start.  This is a write only variable, command will executed any time you rewrite it.	boolean Write Only
<b>VNC Stop</b>	Write 1 inside this tag to force the VNC server to stop.  This is a write only variable, command will executed any time you rewrite it.	boolean Write Only
<b>VNC Restart</b>	Write 1 inside this tag to force the VNC server to restart.  This is a write only variable, command will executed any time you rewrite it.	boolean Write Only
<b>VNC State</b>	VNC server state  0 IDLE  10 RUNNING  -1 ERROR	int Read Only



Write Only Variables cannot be read. Be sure to not use the R/W access mode to avoid the read error icon.



R/W



W

Properties	
ButtonsWithLabel : BtnStd2	
Value	0
DataLink	VNC Start W
Access Type	W
Click Type	momentary
Style	2D
Autorepeat	Disabled
Hold Time (ms)	-1
Label	Start
Up Fill Color	[0, 70, 136]
Down Fill Color	[0, 176, 199]
Events	

## NFC variables

NFC Support system variables give the possibility to read/write from to NFC Tag\* using NFC sensor.



Verify that NFC sensor is supported on device in use.



\* In this chapter "NFC Tag" refers to card or token read by NFC sensor.

### Protocol Editor Settings

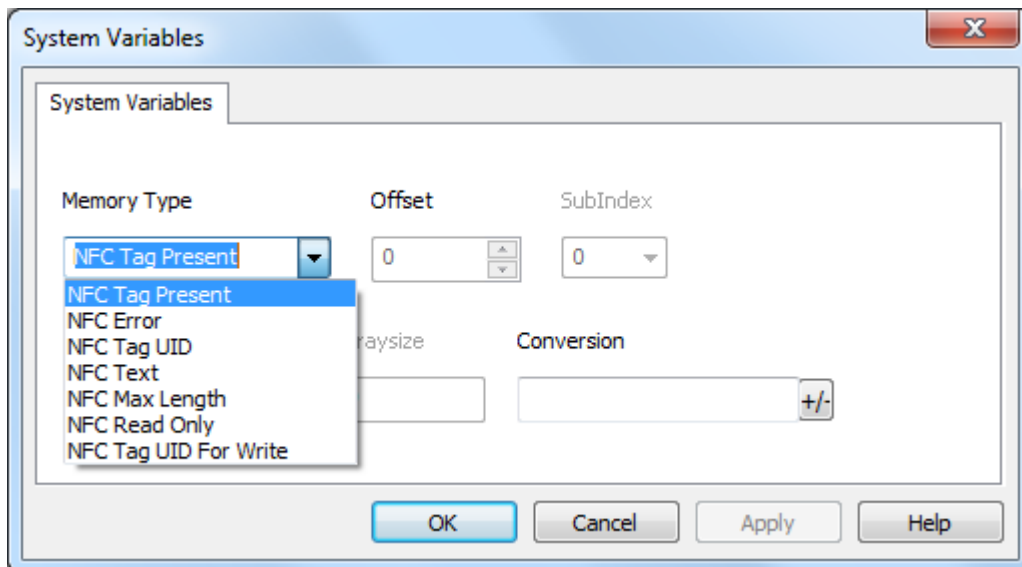
Path: **ProjectView** > **Protocols**

1. Click **+** and select **System Variables**: the **System Variables** dialog is displayed.
2. Select **NFC Support** from the **PLCModels** list.

### Tag Editor Settings

Path: **ProjectView** > **Config** > *double-click Tags*

1. To add a tag, click **+**: a new line is added.
2. Select **System Variables** from the **Driver** list: tag definition dialog is displayed.



Element	Description	Data type
<b>NFC Tag Present</b>	Indicates NFC Tag presence. 0: NFC Tag not detected 1: NFC Tag detected	boolean Read Only
<b>NFC Error</b>	Report last error detected.  -101 Error reading from NFC Tag -50 Tag not detected	int Read Only
<b>NFC Tag UID</b>	Indicates the identifier of NFC Tag, as soon as NFC Tag is detected.	string Read Only
<b>NFC Text</b>	Indicates arbitrary text information in NFC Tag, as soon as NFC Tag is detected. To write on this variable check "Writing procedure" chapter.	string Read/Write
<b>NFC Max Length</b>	Indicates the maximum number in bytes of NFC Text variable, as soon as NFC Tag is detected.	unsignedInt Read Only
<b>NFC Read Only</b>	Indicates if NFC Tag can be written, as soon as NFC Tag is detected. Possible values are:  0 NFC Tag is not "read only": NFC Text can be written.	boolean Read Only
<b>NFC Tag UID For Write</b>	Indicates the identifier of NFC Tag on which NFC Text will be written.	string Read Only

## Writing procedure

- Put NFC Tag near the sensor. The following System Variables will be populated with the values read from NFC Tag:
  - NFC Tag UID
  - NFC Text
  - NFC Max Length
  - NFC Read Only
- Copy the value from "NFC Tag UID" to "NFC Tag UID For Write" in order to prepare the writing.  
Note: Copy can be done by executing "Data Transfer" action from "NFC Tag UID" to "NFC Tag UID For Write", or by JavaScript code.
- Write new content to "NFC Text" variable:
  - if "NFC Tag Present" is 1, write is tried.
  - if "NFC Tag UID For Write" is different from "NFC Tag UID" or is invalid (empty) write is not performed.
  - if "NFC Tag UID For Write" is equal to "NFC Tag UID" write is performed. New inserted value in "NFC Text" will be written into NFT Tag.

## PLCM09 variables

PLCM09 device is a Wireless Modem with LED and digital I/O. The behavior of the related System Variables are depend on how the module has been configured inside the System Settings (see "[PLCM09 Plug-in Wireless Modem](#)" for additional information)

### Protocol Editor Settings

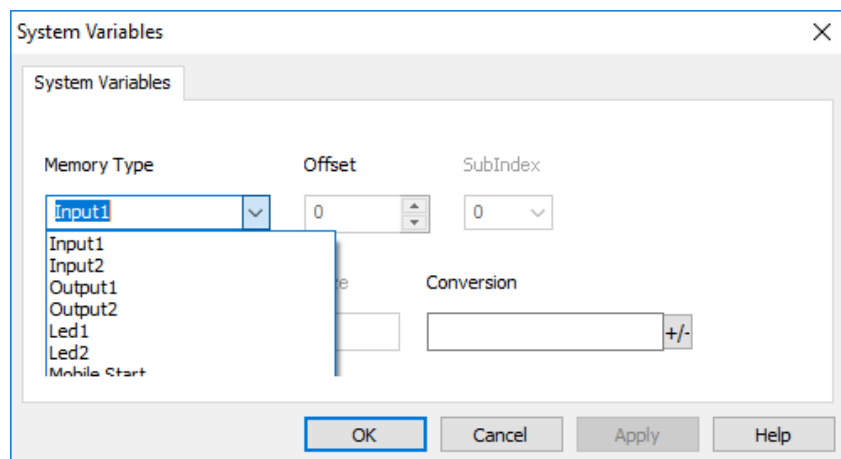
Path: **ProjectView** > **Protocols**




- Click + and select **System Variables**: the **System Variables** dialog is displayed.
- Select PLCM09 from the **PLCModels** list.

### Tag Editor Settings

Path: **ProjectView** > **Config** > double-click **Tags**

- To add a tag, click +: a new line is added.
- Select **System Variables** from the **Driver** list: tag definition dialog is displayed.




Element	Description	Data type
<b>Input1</b> <b>Input2</b>	Value of the Input signals	boolean Read only
<b>Output1</b> <b>Output2</b>	Value of the output signals  Output variables are read/write only when configured as "User controlled" (see " <a href="#">PLCM09 Plug-in Wireless Modem</a> " for additional information). In the other configurations, where output signals are controlled directly from the PLCM09 module, the system variables are read only.	boolean Read/Write
<b>Led1</b> <b>Led2</b>	Value of the LED status  <ul style="list-style-type: none"> <li>• 0 = Off</li> <li>• 1 = On</li> <li>• 2 = Blink</li> </ul> LED variables are read/write only when configured as "User controlled" (see " <a href="#">PLCM09 Plug-in Wireless Modem</a> " for additional information). In the other configurations, where LED status are controlled directly from the PLCM09 module, the system variables are read only.	unsignedByte Read/Write
<b>Mobile Start</b>	Write 1 inside this tag to force the mobile connection to start.   This is a write only variable, command will executed any time you rewrite it.	boolean Write Only
<b>Mobile Stop</b>	Write 1 inside this tag to force the mobile connection to stop.   This is a write only variable, command will executed any time you rewrite it.	boolean Write Only
<b>Mobile Restart</b>	Write 1 inside this tag to force the mobile connection to restart.   This is a write only variable, command will executed any time you rewrite it.	boolean Write Only
<b>Mobile State</b>	Mobile connection state  <ul style="list-style-type: none"> <li>0 IDLE</li> <li>1 STARTING</li> <li>10 RUNNING</li> <li>100 CONNECTING</li> </ul>	int Read Only

Element	Description	Data type
	200 CONNECTED 300 STOPPING -1 GENERICERROR -10 SYSTEMERROR -100 MODEMNOTFOUND -101 MODEMBUSY -110 MODEMCOMM -120 MODEMTIMEOUT -130 MODEMERROR -200 SIMMISSING -300 PINREQUIRED -301 NEWPINREQUIRED -310 PINERROR -320 PUKREQUIRED -330 PUKERROR -400 ROAMINGBLOCKED -500 BADCREDENTIALS	
<b>Mobile Signal</b>	Mobile signal quality (0-100) Value of the signal detected when the device is started	byte Read Only
<b>Mobile Operator</b>	Mobile operator name (e.g. 'Vodafone')	string[8] Read Only
<b>Mobile Access Technology</b>	Mobile access technology -1 N/A 0 GSM (2G) 2 UTRAN (2G) 3 GSM W/EGPRS (2G)	int Read Only


Element	Description	Data type
	4 UTRAN W/HSDPA (3G) 5 UTRAN W/HSUPA (3G) 6 UTRAN W/HSDPA and HSUPA (3G)	
<b>Mobile Registration Status</b>	Mobile registration status  -1 N/A  0 Not registered. Wireless Modem is not currently searching a new operator to register.  1 Registered on home network.  2 Not registered. Wireless Modem is currently searching a new operator to register.  3 Registration denied.  4 Unknown  5 Registered on roaming	int  Read Only
<b>Mobile RX/TX</b>	Number or received/transmitted bytes	unsignedInt[2]  Read Only
<b>Mobile Start Time (Sec)</b>	When mobile connection was started (in seconds since epoch)	unsignedInt  Read Only



Write Only Variables cannot be read. Be sure to not use the R/W access mode to avoid the read error icon.



R/W



W

Properties ⌵ ×

ButtonsWithLabel : BtnStd2

Value	0	+
DataLink	Mobile Start   W	-
Access Type	W	
Click Type	momentary	
Style	2D	
Autorepeat	Disabled	
Hold Time (ms)	-1	
Label	Start	a +
Up Fill Color	[0, 70, 136]	a +
Down Fill Color	[0, 176, 199]	a +
Events		

### JavaScript (Mobile Connection State)

The mobile connection state can be retrieved even from the below JavaScript interface where the "protocolSysVar" is the protocol instance code (e.g. "prot1", "prot2", etc.)



```
Mobile_State = tagMgr.invokeProtocolCommand(protoSysVar,"get_mobile_state", ""); //
get modem status
```

# JSmart variables

JSmart devices have internal sensors that are available through the System Variables protocol.

## Protocol Editor Settings

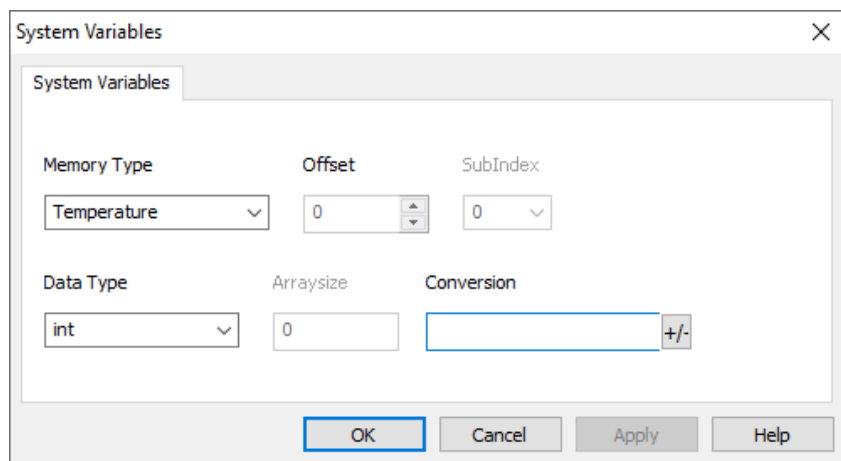
Path: **ProjectView**> **Protocols**

1. Click **+** and select **System Variables**: the **System Variables** dialog is displayed.
2. Select JSmart from the **PLCModels** list.

## Tag Editor Settings


Path: **ProjectView**> **Config** > **double-click Tags**

1. To add a tag, click **+**: a new line is added.
2. Select **System Variables** from the **Driver** list: tag definition dialog is displayed.



Memory Type	Description	Data type
Temperature	Internal temperature in tenths of a degree °C	int Read Only
Humidity	Internal humidity in percentage (0/100%).	int Read Only
Pressures	Internal pressures in mBar	int Read Only
Working Voltage	Working voltage in mV	int Read Only

Memory Type	Description	Data type
<b>Current Consumption</b>	Instant current consumption in mA	int Read Only
<b>Accelerometer Axes</b>	Measure the acceleration (static + dynamic) on the three x, y and z axes. Unit is <b>g</b> , precision is 1/1000 g  x = float with offset 0 or float [0] y = float with offset 1 or float [1] z = float with offset 2 or float [2]	float float[8] Read Only
<b>Accelerometer Angle</b>	Angles in degree between axes. These values are calculated internally from the values of the accelerometer axes.  x^y = float with offset 0 or float [0] x^z = float with offset 1 or float [1] y^z = float with offset 2 or float [2]	float float[8] Read Only
<b>WIFI_Enable</b>	0=Disabled, 1=Enabled	boolean Read Write
<b>WIFI_Mode</b>	0=Station, 1=Access Point	boolean Read Write
<b>WIFI_SSID</b>	Network Name	string[32] Read Write
<b>WIFI_Channel</b>	1/11 Channels  Used when WIFI_Mode=1 (Access Point)	integer Read Write
<b>WIFI_Security</b>	0=NONE, 1=WPA-PSK	integer Read Write
<b>WIFI_Password</b>	Network Password  Used when WIFI_Security is enabled	string[63] Write Only
<b>WIFI_Apply</b>	0=IDLE, 1=APPLY, 2=CANCEL  Values entered inside the WiFi configuration variables (WIFI_Mode, WIFI_SSID, WIFI_Channel, WIFI_Security, WIFI_Password) will be applied when the variable WIFI_Apply will be set to 1, then the variable will reset to 0 after applying the values. When the WIFI_Apply will be set to 2, the WiFi configuration parameters will be restored with the active values.	integer Read Write
<b>WIFI_State</b>	0 = Not Connected, 1 = Connecting, 2 = Connected, -1 = Error	integer Read Only
<b>WIFI_Signal</b>	0..100%	integer Read Only

Memory Type	Description	Data type																					
<b>WIFI_WPS</b>	0=Idle, 1=Active (Return 0 after 2 minute)  When set to 1, the HMI device start the password exchange procedure. When the procedure end (passwords exchanged successfully or after two minutes), the variable will return to 0.	boolean  Read Write																					
<b>WIFI_List</b>	List of available WiFi networks  Used when WIFI_Mode=0 (Station)	string[200]  Read Only																					
<b>WIFI_List_Update</b>	0=Idle, 1=Searching (Return 0 after update)  Command to search the available WiFi networks and fill the WIFI_List. Search start when the value is triggered to 1, when done the value will be restored to 0.	boolean  Read Write																					
<b>WIFI_Error_Code</b>	Return an error code if the parameters provided with the WIFI_Apply command are out of range.  0 = No Error  -11000 = Bad WIFI_Mode  -11011 = Bad WIFI_SSID  -11020 = Bad WIFI_Security  -11030 = Bad WIFI_Password  -11040 = Bad WIFI_Chane	integer  Read Only																					
<b>LED1</b>	Array that controls the RGB LED   <b>BSP v1.0.361 or greater is required</b>  <table border="0" style="width: 100%;"> <tr> <td style="width: 5%;">[0]</td> <td style="width: 45%;">Status</td> <td style="width: 50%;">0=OFF, 1=ON, 2=BLINK</td> </tr> <tr> <td>[1]</td> <td>Color (R - RGB Color)</td> <td>0...255</td> </tr> <tr> <td>[2]</td> <td>Color (G - RGB Color)</td> <td>0...255</td> </tr> <tr> <td>[3]</td> <td>Color (B - RGB Color)</td> <td>0...255</td> </tr> <tr> <td>[4]</td> <td>ON Time (mSec)</td> <td>10 ... 65.535 (65 Sec)</td> </tr> <tr> <td>[5]</td> <td>OFF Time (mSec)</td> <td>10 ... 65.535 (65 Sec)</td> </tr> <tr> <td>[6]</td> <td>Spare</td> <td>N/A</td> </tr> </table>	[0]	Status	0=OFF, 1=ON, 2=BLINK	[1]	Color (R - RGB Color)	0...255	[2]	Color (G - RGB Color)	0...255	[3]	Color (B - RGB Color)	0...255	[4]	ON Time (mSec)	10 ... 65.535 (65 Sec)	[5]	OFF Time (mSec)	10 ... 65.535 (65 Sec)	[6]	Spare	N/A	unsignedInt[8]  Read Write
[0]	Status	0=OFF, 1=ON, 2=BLINK																					
[1]	Color (R - RGB Color)	0...255																					
[2]	Color (G - RGB Color)	0...255																					
[3]	Color (B - RGB Color)	0...255																					
[4]	ON Time (mSec)	10 ... 65.535 (65 Sec)																					
[5]	OFF Time (mSec)	10 ... 65.535 (65 Sec)																					
[6]	Spare	N/A																					

**Notes:**

- WIFI system variables require BSP v1.0.414 or greater
- RGB LED system variable require BSP v1.0.361 or greater

# 15 Actions

---

Actions are functions used to interact with the system and are normally executed when events are triggered.

Events can be triggered by various widgets, for example on press and on release of a button. Not all actions are available for all the events of an object.

Actions are linked to widgets in the **Event** section of the Property pane (Page Editor).

---


<b>Alarm actions</b> .....	<b>194</b>
<b>Database actions</b> .....	<b>195</b>
<b>Event actions</b> .....	<b>199</b>
<b>MultiLanguage actions</b> .....	<b>199</b>
<b>Keyboard actions</b> .....	<b>199</b>
<b>Media Player actions</b> .....	<b>201</b>
<b>FTP actions</b> .....	<b>202</b>
<b>Page actions</b> .....	<b>204</b>
<b>Print actions</b> .....	<b>211</b>
<b>Recipe actions</b> .....	<b>213</b>
<b>Remote Client actions</b> .....	<b>217</b>
<b>ScreenSaver actions</b> .....	<b>218</b>
<b>System actions</b> .....	<b>218</b>
<b>Tag actions</b> .....	<b>230</b>
<b>Trend actions</b> .....	<b>232</b>
<b>Text Editor actions</b> .....	<b>236</b>
<b>User management actions</b> .....	<b>236</b>
<b>Widget actions</b> .....	<b>239</b>

# Alarm actions

Mainly used to acknowledge or reset alarms.


## SelectAllAlarms

Selects all alarms.

Parameter	Description
<b>Mode</b>	<p><b>TOGGLE</b> Reverses the select status.</p> <p> Alarms that are not triggered or have no pending acknowledge or reset requests will never be selected.</p> <p><b>SELECT</b> Selecting all alarms that are triggered or that have acknowledge or reset request pending</p> <p><b>UNSELECT</b> Unselect all alarms</p>

## SelectAlarm

Select a specif alarm.

Parameter	Description
<b>AlarmID</b>	Alarm ID
<b>Selection Flag</b>	<p><b>TRUE</b> Select the alarm.</p> <p> Alarms that are not triggered or have no pending acknowledge or reset requests will not selected.</p> <p><b>FALSE</b> Unselect the alarm.</p>

## AckAlarm

Acknowledge a specific alarm or all selected alarms.

Parameter	Description
<b>AlarmID</b>	Specific Alarm ID
	<b>SELECTED</b> All selected alarms

Acknowledges selected alarms.

## ResetAlarm

Resets a specific alarm or all selected alarms that are not triggered and acknowledged.

Parameter	Description
<b>AlarmID</b>	Specific Alarm ID  <b>SELECTED</b> All selected alarms

## EnableAlarms

Enable or disable a specific alarm or all selected alarms.

Disabled alarms will not generate alarm events.

Parameter	Description
<b>AlarmID</b>	Specific AlarmID  <b>SELECTED</b> All selected alarms  <b>MODIFIED</b> Only alarms with modified enable status
<b>Selection Flag</b>	<b>TRUE</b> Enable the alarm(s).  <b>FALSE</b> Disable the alarm(s).

## Database actions

Using Database actions is possible to exchange data with external SQL databases.

### DBInit



**Important: This action is used only once on an empty database. It is not an initialization command to be called any time the HMI device starts.**

Creates the set of tables required by the project. You do not need to use this action if the database already contains the necessary tables.

## Action Properties

DBInit	
Link Name	myRemoteDB
Custom SQL query	

**Link Name**  
Database link name

Use **Custom SQL query** parameter to define the pages to be created. Leave empty to generate default table names



Tip: Add this command inside a Setup page of your project, used by authorized personal only when installing the application for the first time.

## JavaScript Interface

```
project.dbInit(dbLinkName, sqlCustomQuery);
```

The screenshot shows a software interface with a 'tags' table. The table has three columns: 'tagname', 'tagvalue', and 'Click to Add'. The data rows are:

tagname	tagvalue	Click to Add
Alarm1	0	
SystemTime	1412347792	
Tag01	55	
Tag02	200	
*		

## DBWriteTags, DBReadTags

Transfer the values of the selected tags to/from the remote database.

## Action Properties

DBWriteTags	
Link Name	myRemoteDB
Custom SQL query	
Tag names	Tag01;Tag02

**Link Name**  
Database link name

## Action Properties

DBReadTags	
Link Name	myRemoteDB
Custom SQL query	
Tag names	Tag01;Tag02

**Link Name**  
Database link name

## JavaScript Interface

```
project.dbWriteTags(dbLinkName, sqlCustomQuery, Tags);
```

```
project.dbReadTags(dbLinkName, sqlCustomQuery, Tags);
```

## DBWriteGroups, DBReadGroups

Transfer groups of tags between the HMI device and the database.

Action Properties		Action Properties	
<b>DBWriteGroups</b>		<b>DBReadGroups</b>	
Link Name	myRemoteDB	Link Name	myRemoteDB
Custom SQL query		Custom SQL query	
Group names	Group1	Group names	Group1
<b>Link Name</b> Database link name		<b>Link Name</b> Database link name	

### JavaScript Interface

```
project.dbWriteGroups(dbLinkName, sqlCustomQuery, Groups);
```

```
project.dbReadGroups(dbLinkName, sqlCustomQuery, Groups);
```

## DBWriteTrend

Inserts the values of the last data sampled in the selected range of time inside the Trends table of the remote database.

Action Properties	
<b>DBWriteTrends</b>	
Link Name	myRemoteDB
Custom SQL query	
Trend names	Trend1
Duration	10 min
<b>Link Name</b> Database link name	

### JavaScript Interface

```
project.dbWriteTrends(dbLinkName, sqlCustomQuery, trendName, durationIndex)
```

## DBWriteEvents

Inserts the values of the last events in the selected range of time inside the Events table of the remote database.

Action Properties		Action Properties	
<b>DBWriteEvents</b>		<b>DBWriteEvents</b>	
Link Name	myRemoteDB	Link Name	myRemoteDB
Custom SQL query		Custom SQL query	
Buffer	AlarmBuffer1	Buffer	AuditTrail
Duration	1 hour	Duration	1 hour
<b>Buffer</b> Select Event buffer		<b>Buffer</b> Select Event buffer	



## JavaScript Interface

```
project.dbWriteEvents (dbLinkName, sqlCustomQuery, archiveName, durationIndex)
```

## DBWriteRecipes, DBReadRecipes

Transfer the recipe data to/from the remote database.

Action Properties	
<b>DBWriteRecipes</b>	
Link Name	myRemoteDB
Custom SQL query	
Recipe names	Recipe1 <span style="float: right;">+</span>
<b>Recipe names</b> Recipe names seperated by semicolon(;)	

Action Properties	
<b>DBReadRecipes</b>	
Link Name	myRemoteDB
Custom SQL query	
Recipe names	Recipe1 <span style="float: right;">+</span>
<b>Recipe names</b> Recipe names seperated by semicolon(;)	

## JavaScript Interface

```
project.dbWriteRecipes (dbLinkName, sqlCustomQuery, recipeNames)
```

```
project.dbReadRecipes (dbLinkName, sqlCustomQuery, recipeNames)
```

## DBResetErrors

Reset all the three status variables of the selected database link. (See Database variables in ["Default variables" on page 162](#)).

Action Properties	
<b>DBResetErrors</b>	
Link Name	myRemoteDB
<b>Link Name</b> Database link name	

## JavaScript Interface

```
project.dbResetErrors (dbLinkName)
```

## RefreshDBTable

Executes the SQL query of the selected "DB table data source" widget to update its data.

Action Properties

RefreshDBTable	
Data Source	TableDBSrcWgt

**RefreshDBTable**  
Update the db table.

## Event actions

Used by Alarm History widget to scroll events/alarms backward/forward in table view (event buffer widget).

### ScrollEventsBackward

Scrolls events/alarms backward in table view (event buffer widget).

### ScrollEventsForward

Scrolls events/alarms forward in table view (event buffer widget).

## MultiLanguage actions

Selects the application language.

### SetLanguage

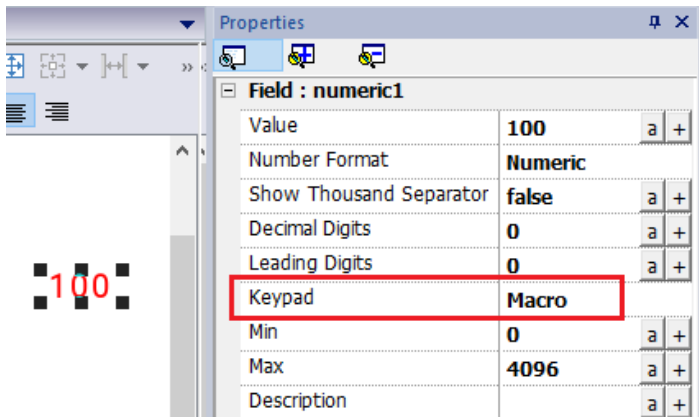
Sets the language used. The selected language will be applied at runtime to all applicable widgets.

## Keyboard actions

Changes the use of keypads.

### SendKey

Sends one character to a numeric widget. The **Keypad** property of the numeric widget must be set as **Macro**.

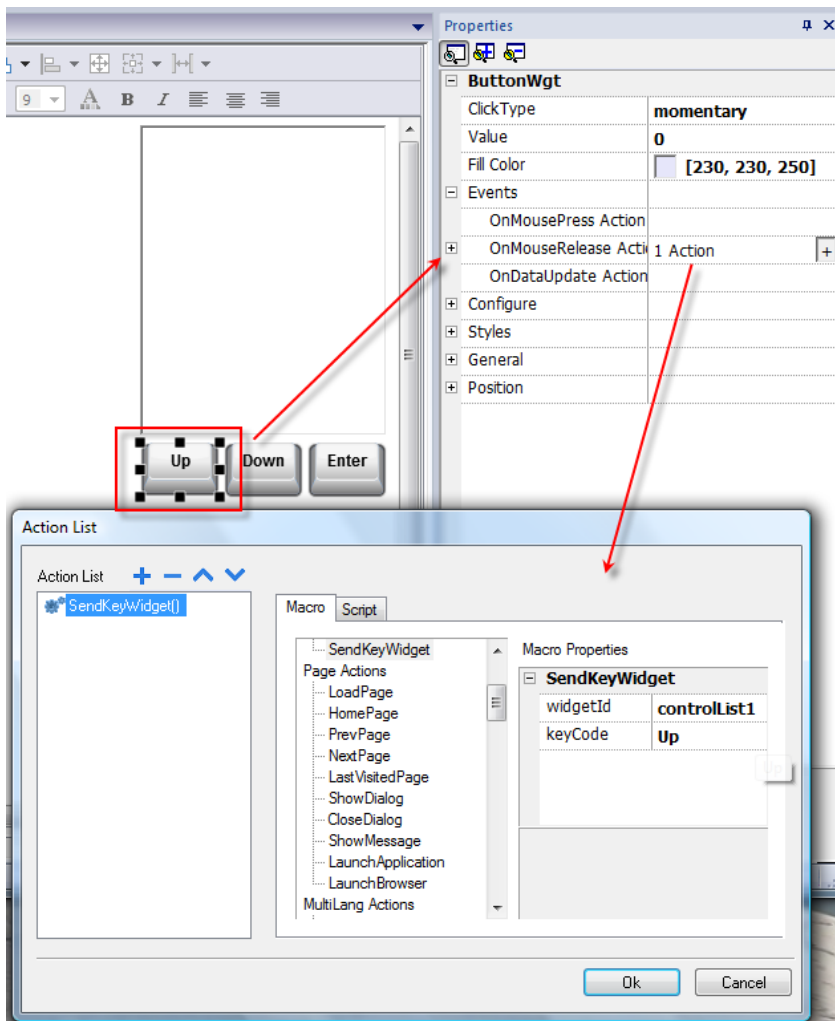


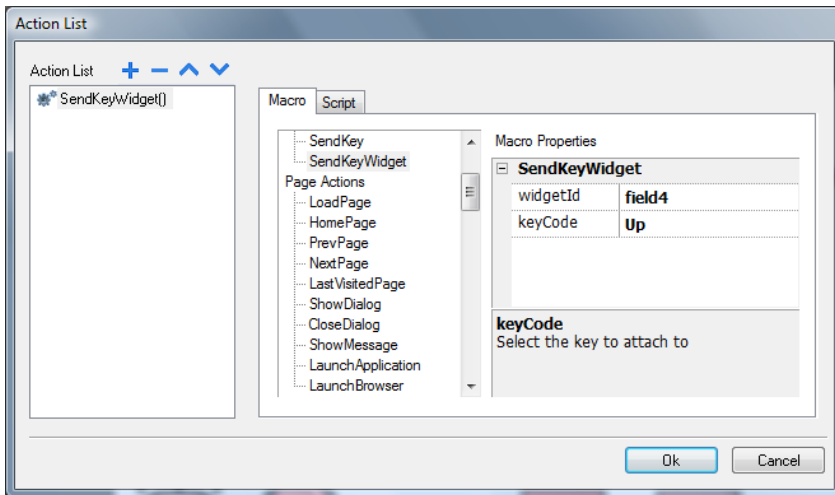
## SendKeyWidget

Sends one character to a specific widget.

### Example

The **Up** and **Down** buttons use the **SendKeyWidget** action in association with the **Control List Widget**.





## ShowKeyPad

Shows the default operating system touch keypad.

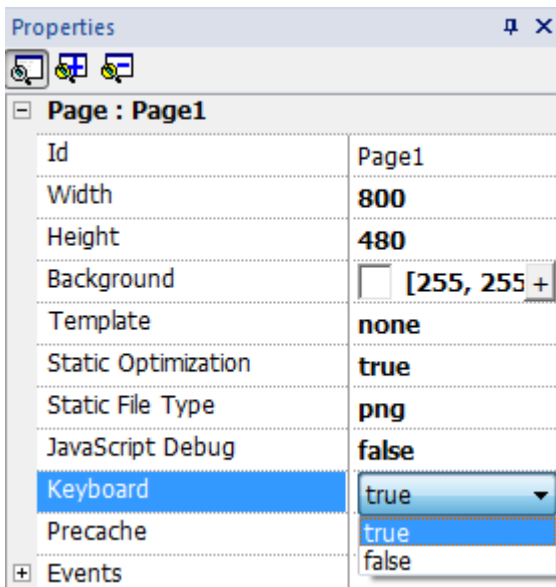


Note: might not be supported by all operating systems.

## KeyboardMacros

Enables/disables the use of actions when using external keyboards. Action execution can be enabled/disabled both at project and at page level.

The effect is equivalent to the use of the property Keyboard for project and page.



## Media Player actions

Interact with the Media Player widget at runtime.

Action	Description
PlayMedia	Starts playing the video.
StopMedia	Stops the video.
ReloadMedia	Restarts video from the beginning.
PauseMedia	Pauses the video.
BrowseMedia	Selects the video to play.

## FTP actions

Used to upload and download files to and from a remote FTP server.

### ftpGET

Download files from a remote FTP server

Parameter	Description
FtpConfig	Configuring the FTP parameters
FtpRemoteFileName	File name on the remote FTP server to download (source)
FtpLocalFileName	File name on local HMI device (destination)

### ftpPUT

Upload files to a remote FTP server

Parameter	Description
FtpConfig	Configuring the FTP parameters
FtpLocalFileName	File name on local HMI device (source)
FtpRemoteFileName	File name on the remote FTP server to download (Destination)



Filenames can contain wildcards.

When transferred, system variables are updated with the status of ongoing operations (see ["FTP client variables" on page 150](#) for details).

## FTP Server Configuration

To configure the FTP parameter, enter the following information for the **FtpConfig** setting:

Parameter	Description
<b>FTP Address</b>	FTP server IP Address
<b>Server Port</b>	Port for FTP connection (default = 21).
<b>Authentication</b>	Select the FTP authentication to use: <ul style="list-style-type: none"> <li>• Normal (Username and password required)</li> <li>• Anonymous</li> </ul>
<b>User Name</b>	Username of the remote FTP account
<b>Password</b>	Password of the remote FTP account

Click + to add more FTP servers configuration.



Tip: Use tags if you want change the server parameters dynamically from the HMI Runtime.

## FTP JavaScript Interface

### ftpConfig

```
ftpCONFIG (IPAddress, Port, Authentication, UserName, Password)
```

Set the FTP parameters to use on next FTP calls

Parameter	Description
<b>IPAddress</b>	FTP server IP Address.
<b>Port</b>	Port for FTP connection (default = 21).
<b>Authentication</b>	Select the FTP authentication to use: <ul style="list-style-type: none"> <li>• Normal (Username and password required)</li> <li>• Anonymous</li> </ul>
<b>UserName</b>	Username of the remote FTP account
<b>Password</b>	Password of the remote FTP account

### ftpGET

```
ftpGET (remoteFileName, localFileName, [callback])
```

Download files from a remote FTP server

Parameter	Description
<b>remoteFileName</b>	File name on the remote FTP server to download (source)
<b>localFileName</b>	File name on local HMI device (destination)
<b>callback</b>	Function that will be call at the end of the FTP transfer

## ftpPUT

```
ftpPUT (localFileName, remoteFileName, [callback])
```

Upload files to a remote FTP server

Parameter	Description
<b>remoteFileName</b>	File name on the remote FTP server to download (source)
<b>localFileName</b>	File name on local HMI device (destination)
<b>callback</b>	Function that will be call at the end of the FTP transfer

Example:

```
project.ftpCONFIG("192.168.0.200", "21", "true", "admin", "admin");

project.ftpGET( "data.txt",
               "\\USBMemory\\data.txt",
               function(ftpStatus) {fnFtpGetFinished(ftpStatus);} );

function fnFtpGetFinished(ftpStatus) {
    alert(ftpStatus);
}
```

## Page actions

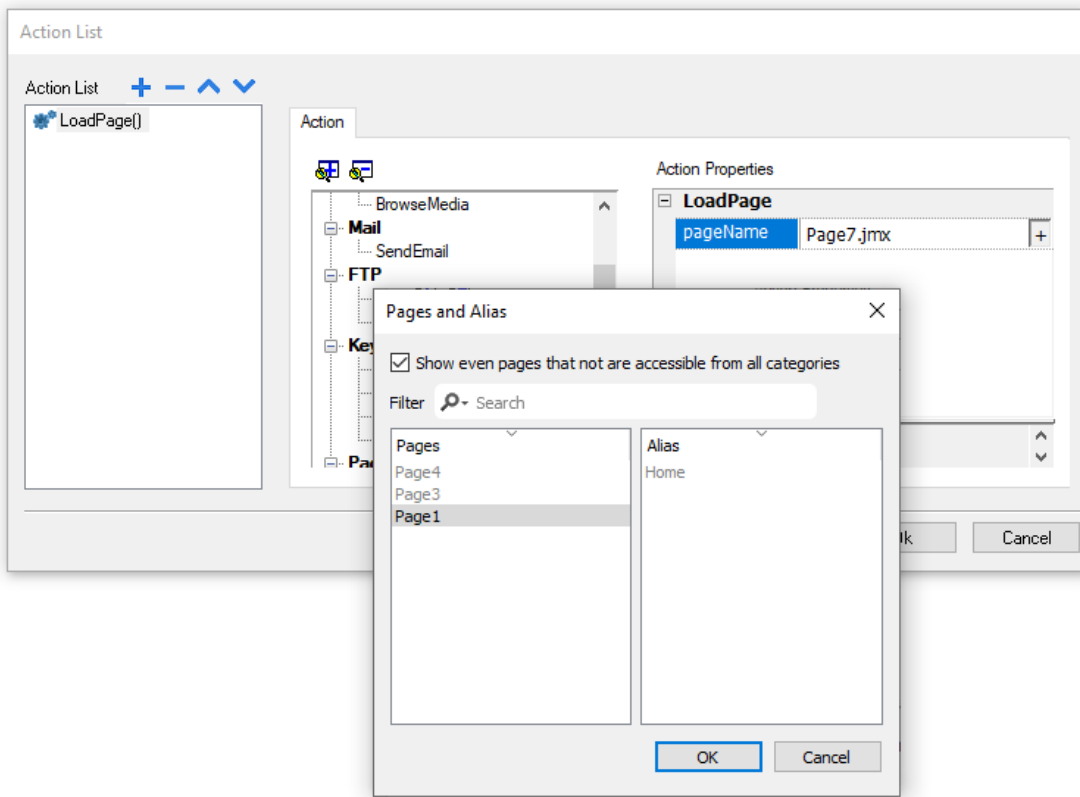
Page navigation. Page actions can be used with the following events:

- OnMouseClicked,
- OnMouseRelease,
- OnMouseHold
- OnActivate
- OnDeactivate
- Alarms
- Schedulers.

## LoadPage

Go to the selected page of the project.

Starting from PB610 Panel Builder 600 v4.0 in addition to the pages you can use the aliases (see ["Alias pages" on page 75](#))



When *"Show even pages that not are accessible from all categories"* is selected, even the pages not reachable by all categories are listed.

## HomePage

Go to the home page.

You can set the home page in the **Behavior** section of the **Project Widget**, see ["Project" on page 87](#)

## PrevPage

Go to the previous page.

## NextPage

Go to the next page.

## LastVisitedPage

Go to the previously displayed page

## ShowDialog

Opens a dialog page defined in the project.

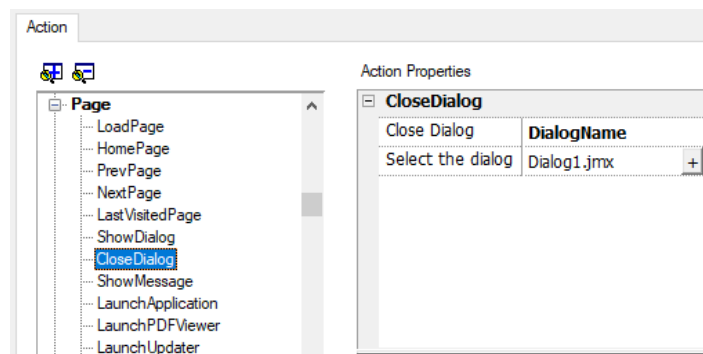


## CloseDialog

Close dialog pages.



Note: This action is applicable only to dialog pages.



### CloseDialog options

Option	Description
<b>All</b>	Closes all open dialogs
<b>Selected</b>	Closes only active dialog
<b>DialogName</b>	Closes dialog specified as <b>fileName</b> property

### JavaScript Interface

```
project.closeDialog(DialogID);
```

Where *DialogID*:

<b>All</b>	Closes all open dialogs
<b>Selected</b>	Closes only active dialog
<b>DialogName.jmx or AliasName</b>	Closes dialog specified as <b>fileName</b> parameter

### Examples

Example	Behavior
<code>project.closeDialog("All");</code>	All open dialogs are closed
<code>project.closeDialog("Selected");</code>	The selected dialog is closed
<code>project.closeDialog("Dialog1.jmx");</code>	All instances of Dialog1 are closed

The function `project.closeDialog()`; without parameter works as `project.closeDialog("Selected");`.

## ShowMessage

Displays a popup message. Enter the text of the message to be displayed.

## LaunchApplication

Launches an external application.

Parameter	Description
<b>App Name</b>	Executable name with extension (for example, "notepad.exe" to run Notepad)
<b>Path</b>	Application path.
<b>Arguments</b>	Application specific arguments (for example, <code>\flash\qthml\Manual.pdf</code> to open the document "Manual.pdf")
<b>Single Instance</b>	Argument to start the application in a single instance or multiple instances.  When single instance is selected, the system first verifies whether the application is already running; if so, then the application is brought to the foreground, if not, then the application is launched.
<b>FlushRuntimeCache</b>	Flush all runtimes cache to free as more ram as possible before running the application.



Note: Arguments with spaces must be quoted (for example, "\Storage Card\Manual.pdf")

Example:

LaunchApplication	
Application Name	\Windows\cmd.exe
Executable path	
arguments	/c "\Flash\New Folder\test.bat" Par1 Par2
Single Instance	true

## LaunchBrowser

Opens the default web browser. You can define URL address as argument.



Note: Only works on platforms having a native web browser (for example, on Windows CE PRO with Internet Explorer enabled).

## LaunchVNC

Starts VNC server and opens the configuration.




Macro available only for HMI devices based on Windows CE platform.  
On HMI devices based on Linux platform the VNC service can be enabled from the "Service" tab of the "System Settings (Linux Devices)" on page 617 BSP v1.0.44 or higher required.

See "Plug-in" on page 86 to include it on Windows CE devices.

## LaunchPDFViewer

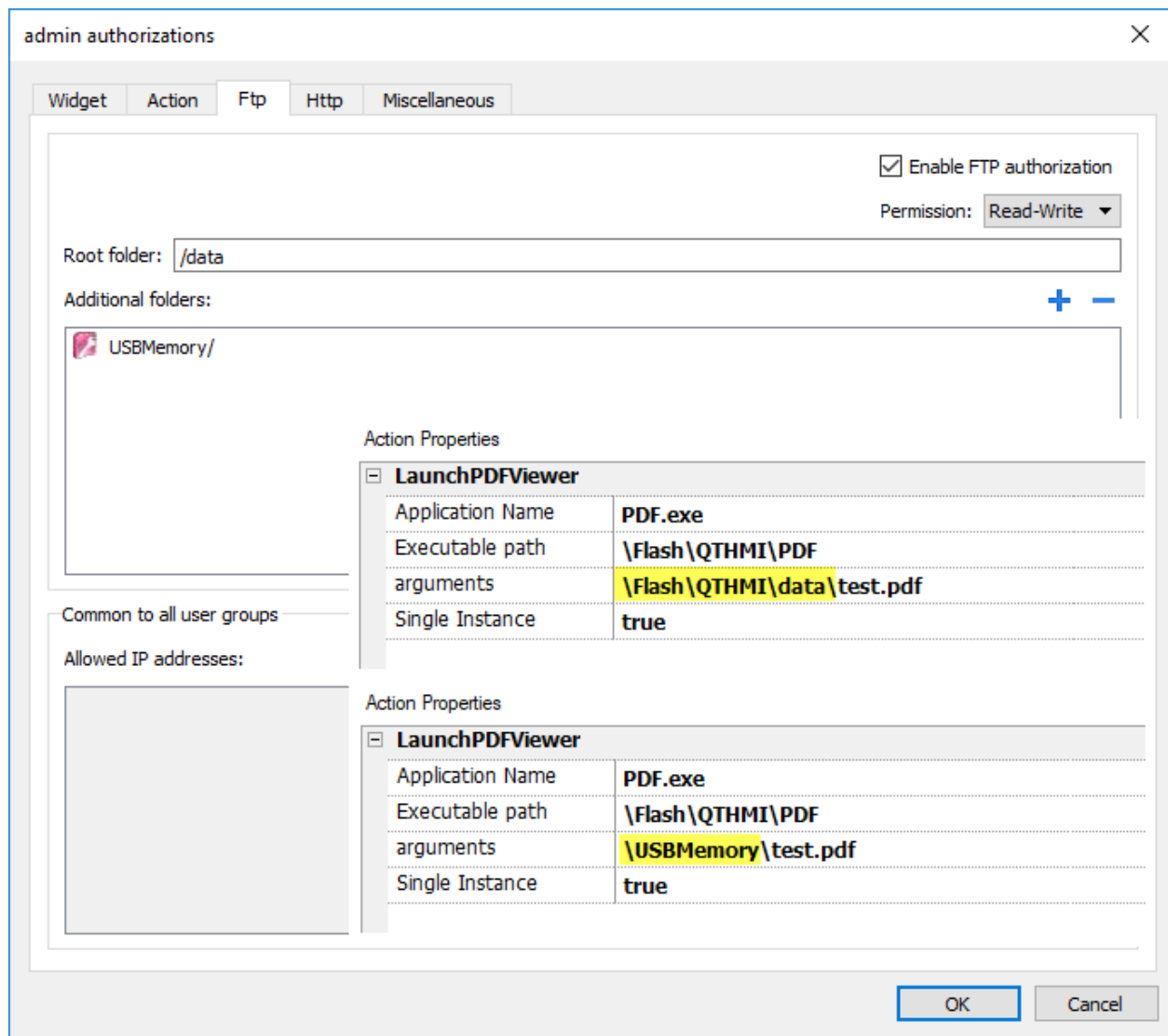
Starts PDF Viewer.

 On WCE devices, only works on devices that include PDF Viewer. See ["Plug-in" on page 86](#) to include it on Windows CE devices.

On Linux devices, BSP v1.00.44 or greater is required.

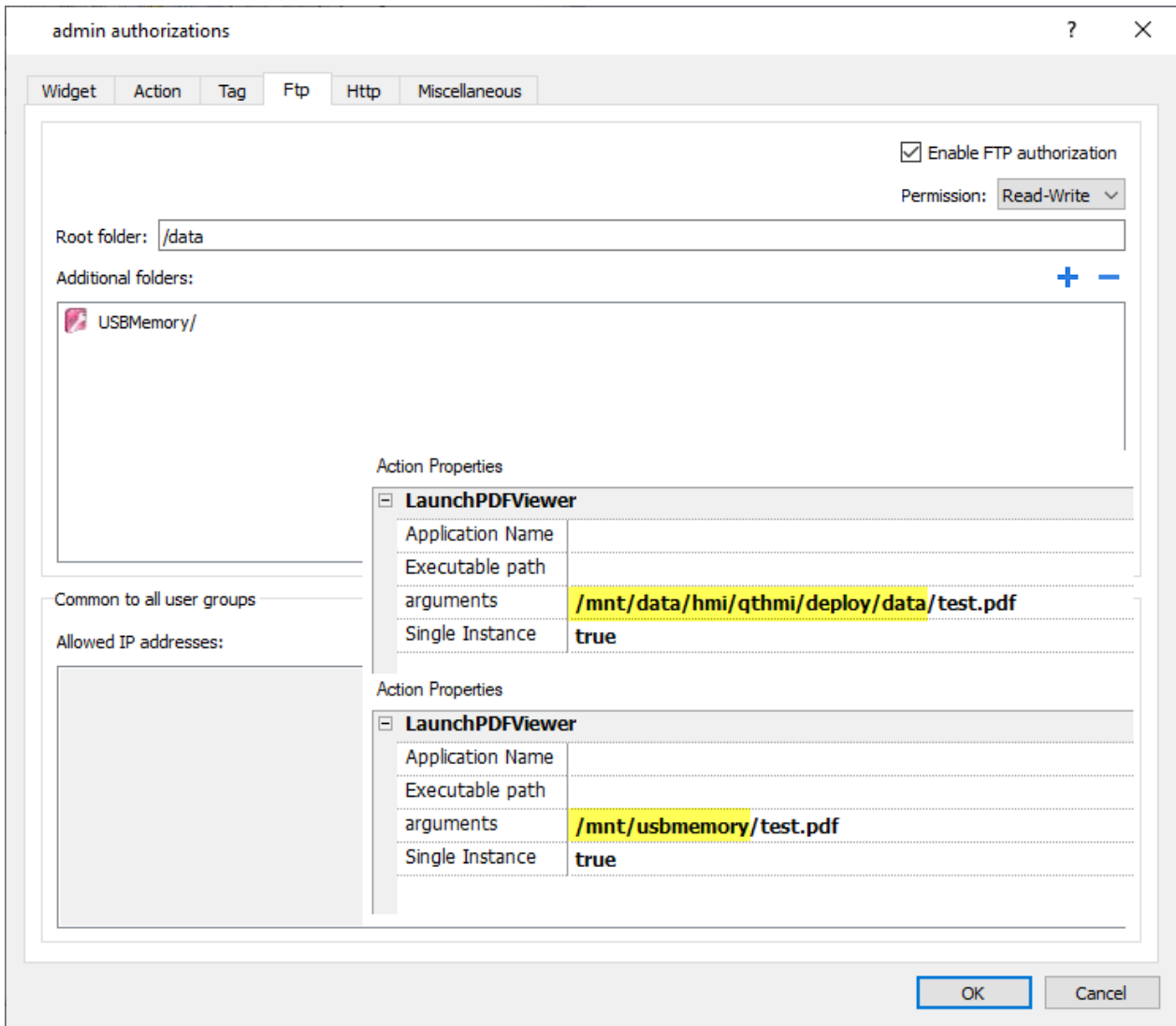
Note that the pathname of the arguments field uses native OS format (see ["HMI devices capabilities" on page 613](#)).

On **WinCE devices**, the HMI application is installed on path \Flash\QTHMI\ and pathname's syntax use the backslash character.



The screenshot shows the 'admin authorizations' dialog box with the 'Ftp' tab selected. The 'Enable FTP authorization' checkbox is checked, and the 'Permission' is set to 'Read-Write'. The 'Root folder' is '/data'. Under 'Additional folders', 'USBMemory/' is listed. Two 'Action Properties' sections are visible, both for the 'LaunchPDFViewer' action. The first section shows the executable path as '\Flash\QTHMI\PDF' and arguments as '\Flash\QTHMI\data\test.pdf'. The second section shows the executable path as '\Flash\QTHMI\PDF' and arguments as '\USBMemory\test.pdf'. The 'Single Instance' property is set to 'true' in both sections. 'OK' and 'Cancel' buttons are at the bottom right.

On **Linux devices**, the HMI application is installed on path /mnt/data/hmi/qthmi/deploy/ and pathname's syntax use the slash character.



**-hide-open-button** (available only on Linux devices)

Using this option, the icon to open a different file will be removed from the PDF toolbar (to restrict navigation to PDF file already opened and passed via command line).

Action Properties

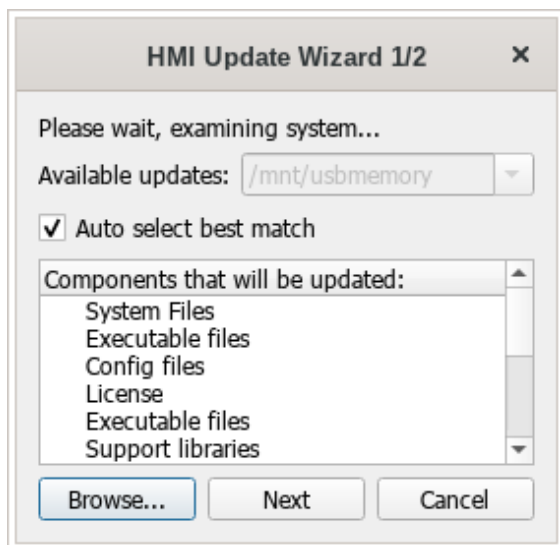
LaunchPDFViewer	
Application Name	
Executable path	
arguments	/mnt/usbmemory/test.pdf <b>-hide-open-button</b>
Single Instance	true

## LaunchUpdater

Updates project and runtime from an external device.

Use **Path** parameter to specify the folder that will contain the update package file. Leave the path parameter empty if you prefer select the file manually on the HMI device when the macro is invoked.

When the LaunchUpdater macro is executed, the below dialog is showed on HMI device



Note: Not supported in devices based on Windows platform.

## JavaScript Interface

`project.launchUpdater(strPath)`

## Examples

```
project.launchUpdater("\\USBMemory")
```

## LockScreen

Temporarily locks the touch screen. Allows cleaning the touch screen.

The system variable **Time remaining to unlock** displays the time remaining to unlock. See ["Screen variables" on page 153](#)

## LoadProject

Unload current project and load the selected project inside the HMI device.

The project name has to be specified using relative path, as for the below example:

Action Properties

LoadProject	
projectName	workspace/project2/project2.jpr

## LastVisitedProject



Unload current project and return to previous project

## Print actions

Manages print tasks.

### PrintGraphicReport

Prints a graphic report.

Parameter	Description
<b>reportName</b>	Assigns a name to the report
<b>silent</b>	<b>false</b> = allows to set printer properties at runtime
<b>fileName</b>	File name (available only for PDF reports) Supported placeholders: <ul style="list-style-type: none"> <li>• %n = Report name</li> <li>• %p = Project name</li> <li>• %y = Year, %M = Month, %d = Day</li> <li>• %h = Hour, %m = Minutes, %s = Seconds.</li> </ul>
<b>folderPath</b>	Folder Path (available only for PDF reports) Note that the pathname of the arguments field uses native OS format (see <a href="#">"HMI devices capabilities" on page 613</a> ). <ul style="list-style-type: none"> <li>• On WinCE devices Path for USB Device is "\USBMemory"</li> <li>• On Linux devices Path for USB Device is "/mnt/usbmemory" "testFolder" will be inside "/mnt/data/hmi/qthmi/deploy/testFolder"</li> </ul>
<b>Signed</b>	When the output is a PDF file, generate a signed file using the x.509 certificate of the panel. <div style="margin-top: 10px;">  <p><b>On Linux devices, the BSP v1.0.507 or greater is required</b> <b>On WinCE devices, the BSP v2.31 or greater is required</b></p> </div> <div style="margin-top: 10px;">  <p>The algorithm to use to signing is defined inside the project properties parameters See <a href="#">"Project" on page 87</a> for the available algorithms</p> </div> <p>See also:</p> <ul style="list-style-type: none"> <li>• <a href="#">"Signed PDF files" on page 359</a></li> </ul>

### PrintText

Prints a string.

Parameter	Description
<b>text</b>	String to be printed
<b>silent</b>	<b>false</b> = allows to set printer properties at runtime

This action works in line printing mode and uses a standard protocol common to all printers that support it. Text is printed immediately line by line or after a timeout custom for each printer model.



Note: printing could a few minutes for models not designed for line printing.



Available on WinCE platforms only (find the platform of your device at ["HMI devices capabilities" on page 613](#))

## PrintBytes

Prints an hexadecimal string representing data to print (for example, "1b30" to print < ESC 0 >).

Parameter	Description
<b>bytes</b>	Exadecimal string to print
<b>silent</b>	<b>false</b> = allows to set printer properties at runtime

This action works in line printing mode and uses a standard protocol common to all printers that support it. Text is printed immediately line by line or after a timeout custom for each printer model.



Note: printing could a few minutes for models not designed for line printing.



Not available on Linux platforms (find the platform of your device at ["HMI devices capabilities" on page 613](#))

## EmptyPrintQueue

Flushes the current printing queue. If executed while executing a job, the queue is cleared at the end of the job.

## PausePrinting

Puts the current printing queue on hold. If executed while executing a job, the queue is paused at the end of the job.

## ResumePrinting

Restarts a queue previously put on hold.

## AbortPrinting

Stop the execution of the current job and removes it from the queue. If the queue has another job, then, after aborting, the next job starts.

## Recipe actions

Used to program recipe management.

### DownloadRecipe

Copy recipe data from HMI device flash memory to the controller (e.g. PLC, local variable, depending on the protocol).

Parameter	Description
<b>RecipeName</b>	Name of recipe to download
<b>RecipeSet</b>	Number of recipe set to copy. <b>curSet</b> = download currently selected recipe set

### UploadRecipe

Saves recipe data from the controller (e.g. PLC, local variable, depending on the protocol) to the device Flash Memory.

Parameter	Description
<b>RecipeName</b>	Name of recipe to upload
<b>RecipeSet</b>	Number of recipe set to copy. <b>curSet</b> = upload currently selected recipe set

### WriteCurrentRecipeSet

Sets the selected recipe as current recipe set.

Parameter	Description
<b>RecipeName</b>	Name of recipe to set as current recipe
<b>RecipeSet</b>	Recipe set to define as current recipe set

### DownloadCurRecipe

Downloads current set of recipe data to the controller.

No parameter is required.

### UploadCurRecipe

Uploads set of controller data to current recipe set.

No parameter is required

### ResetRecipe




Restores factory settings for recipe data. Original recipe data will overwrite uploaded recipes

Select the recipe that you want to reset to factory data.





## DumpRecipeData

Dumps recipe data to internal or external storage. Data is saved in .csv format.

Parameter	Description
<b>RecipeName</b>	Name of recipe to dump
<b>RecipeDataSet</b>	Select the recipe dataset to dump Not available when RecipeName=AllRecipes. In this case, all datasets are dumped.
<b>FilePath</b>	<p>Destination folder</p> <ul style="list-style-type: none"> <li>• Internal = <code>\Flash\QTHMI\workspace\Dump</code></li> <li>• USB drive = <code>\USBMemory</code></li> <li>• SD Card = <code>\Storage Card</code></li> <li>• Public Network = <code>\\&lt;hostname or IP&gt;\sharePath</code></li> <li>• Private Network = <code>\\&lt;username&gt;:&lt;password&gt;@&lt;hostname or IP&gt;\sharePath</code></li> </ul> <p> Note: supported formats for external memory are FAT or FAT32 (NTFS format is not supported).</p> <p> Note: Private networks are supported only from Linux devices with BSP 1.0.25 and above.</p>
<b>FileName</b>	<p>Tag that specifies a filename.</p> <p>The below wildcards are supported</p> <ul style="list-style-type: none"> <li>• %r = Recipe name</li> <li>• %d = Dataset name</li> </ul> <p>Example: %r_%d</p>
<b>DateTimePrefixFileName</b>	<b>true</b> = the dumped file will have date and time as prefix to its name (for example D2012_01_01_T10_10_recipe1.csv)
<b>TimeSpec</b>	<p>Time format:</p> <ul style="list-style-type: none"> <li>• <b>Local</b> = the time values exported are the time of the HMI device.</li> <li>• <b>Global</b> = the time values exported are in UTC format.</li> </ul>
<b>KeepCaseSensitivity</b>	<p>Allow case sensitivity.</p> <p> Only applicable under Linux, other platforms does save in lower case</p>

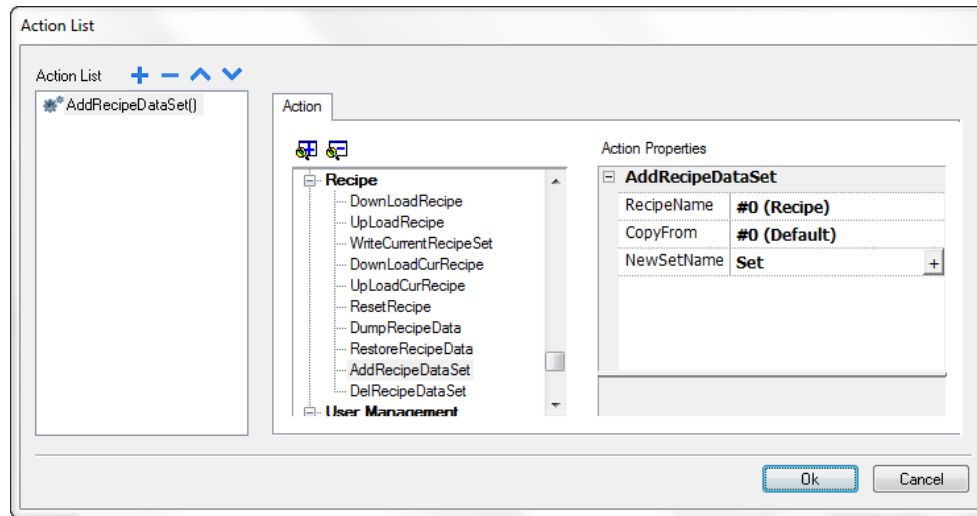
## RestoreRecipeData

Restores previously saved recipe data.

Parameter	Description
<b>RecipeName</b>	<p>Recipes to restore:</p> <ul style="list-style-type: none"> <li>AllRecipes Data of all recipes will be replaced with the data read from the external file</li> <li>CurrentRecipe Only the data of the current selected recipe will be replaced with the data read from the external file</li> </ul>
<b>RecipeDataSet</b>	<p>Available only when RecipeName=CurrentRecipe.</p> <p>Select the data sets to restore:</p> <ul style="list-style-type: none"> <li>AllRecipeDataSet All data set will be restored</li> <li>curSet Only the data set of the current selected data set will be restored</li> </ul>
<b>Restore Type</b>	<p>Available only when RecipeDataSet=AllRecipeDataSet.</p> <p>This parameter defines the behavior when the number of data sets in the file to be restored does not match the number of data sets in the HMI device.</p> <ul style="list-style-type: none"> <li>Replace All data sets in the device are removed and replaced with the data sets from the csv file</li> <li>Match Replace only the data set in the device that has the same data set id</li> <li>MatchAndAdd Replace the data set in the device that has the same data set id and add the additional data set found in the csv file (Note: data sets in the device but not in the csv file are not removed from the device)</li> </ul>
<b>FilePath</b>	<p>Source folder</p> <ul style="list-style-type: none"> <li>Internal = <i>\Flash\QTHMI\workspace\Dump</i></li> <li>USB drive = <i>\USBMemory</i></li> <li>SD Card = <i>\Storage Card</i></li> <li>Public Network = <i>\\&lt;hostname or IP&gt;\sharePath</i></li> <li>Private Network = <i>\\&lt;username&gt;:&lt;password&gt;@&lt;hostname or IP&gt;\sharePath</i></li> </ul> <p> Note: supported formats for external memory are FAT or FAT32 (NTFS format is not supported).</p> <p> Note: Private networks are supported only from Linux devices with BSP 1.0.25 and above.</p>
<b>FileName</b>	Attached tag from which to read the file name at runtime.
<b>BrowseForFile</b>	<p><b>true</b> = shows the Open dialog to browse the file to read.</p> <p><b>false</b> = no dialog is shown,</p>

## AddRecipeDataSet

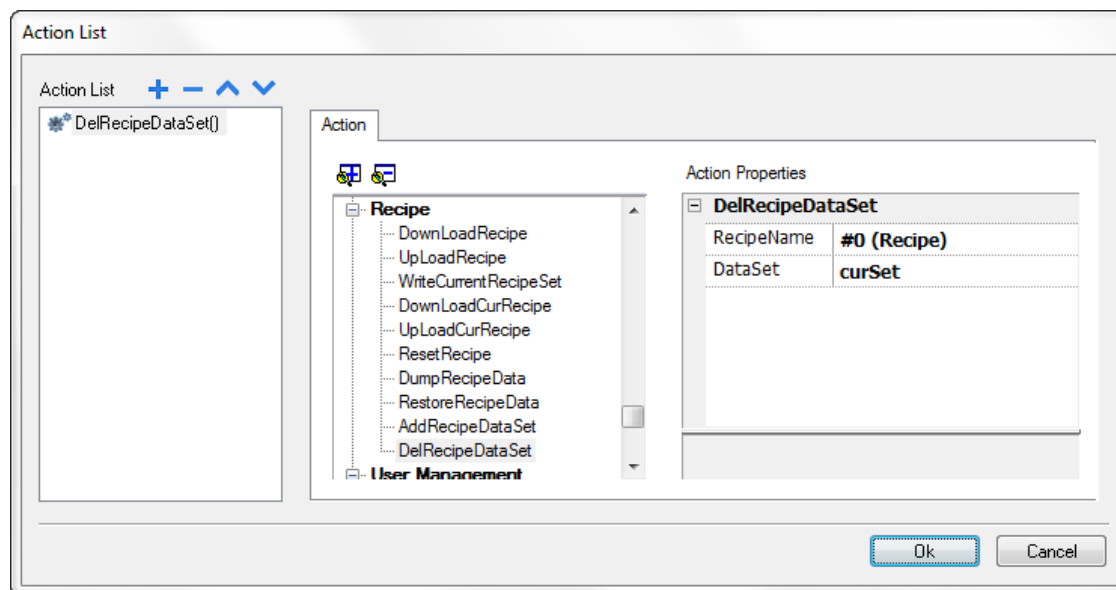
Adds a new dataset to the selected recipe. The new dataset is appended at the end of the already defined datasets.



Parameter	Description
<b>RecipeName</b>	Recipe where the dataset is added.
<b>CopyFrom</b>	Dataset from where parameters values are copied from to initialize the new dataset
<b>NewSetName</b>	Name of new dataset. Here you can use a tag reference.

## DelRecipeDataSet

Deletes a dataset from the selected recipe. Deleting a dataset will rearrange the position number of the datasets that follow.



Parameter	Description
RecipeName	Recipe where the dataset is to be deleted.
DataSet	Dataset to be deleted.

## Remote Client actions

Used to upload or download a bunch of files to and from a remote HMI device. These actions can only be used from a remote HMI Client to access remote files via FTP.



**Important: Enable FTP support and give all necessary user rights to the folders used to transfer files.**

### UploadToHMI

Opens a dialog to select the folder that contains the files to be uploaded to the remote HMI device.

Parameter	Description
Destination	Destination path on HMI device for file upload
Filter	File extensions of the files to be uploaded separated by commas (for example, *.txt, *.csv)

### DownloadFromHMI

Opens a dialog to select the folder that contains the files to be downloaded from the remote HMI device.

Parameter	Description
Source	Source path on the HMI device for file download
Filter	File extensions of the files to be downloaded separated by commas (for example, *.txt, *.csv)

### JavaScript Interface

```
boolean project.uploadToHMI(dirPath, strFilter);
```

```
boolean project.downloadFromHMI(dirPath, strFilter);
```

Parameter	Description
dirPath	Source path on the HMI device for file download/upload
strFilter	File extensions of the files to be displayed separated by commas (for example, *.txt)

Return values:

True	Transfer successful
False	Transfer failed



Note: When transferred, system variables are updated with the status of ongoing operations (see "[Remote Client variables](#)" on page 152)

## ScreenSaver actions

### StartScreenSaver

Enter in the screen saver mode.

#### Java Script Interface

```
project.startScreenSaver();
```

### StopScreenSaver

Exit from the screen saver mode.

#### Java Script Interface

```
project.stopScreenSaver();
```

## System actions



Used to manage system properties.


### Restart

Restarts the runtime.

### DumpTrend

Stores historical trend data to external drives (USB drive or SD card).

Parameter	Description
<b>TrendName</b>	Name of historical trend to store
<b>FolderPath</b>	Destination folder: <ul style="list-style-type: none"> <li>• Internal = <code>\Flash\QTHMI\workspace\Dump</code></li> <li>• USB drive = <code>\USBMemory</code></li> <li>• SD Card = <code>\Storage Card</code></li> <li>• Public Network = <code>\\&lt;hostname or IP&gt;\sharePath</code></li> <li>• Private Network = <code>\\&lt;username&gt;:&lt;password&gt;@&lt;hostname or IP&gt;\sharePath</code></li> </ul> <div style="margin-top: 10px;">  Note: supported formats for external memory are FAT or FAT32 (NTFS format is not supported).         </div> <div style="margin-top: 10px;">  Note: Private networks are supported only from Linux devices with BSP 1.0.25 and         </div>

Parameter	Description
	 above.
<b>FileFormat</b>	<p><b>Binary</b> = the buffer is dumped in binary format (a .dat file and .inf file). Both these files are then required to convert data in .csv format by an external utility.</p> <p><b>Compatibility CSV</b> = the buffer is dumped to the specified location as a .csv file format compatible with versions 1.xx</p> <p><b>Compact CSV</b> = the buffer is dumped to the specified location as a .csv file using a newer format</p> <p>See <a href="#">"Exporting trend buffer data" on page 297</a></p>
<b>DateTimePrefix</b>	<p><b>true</b> = the dumped file will have date and time as prefix to its name (for example D2012_01_01_T10_10_Trend1.csv)</p>
<b>TimeSpec</b>	<p>Time format:</p> <ul style="list-style-type: none"> <li>• <b>Local</b> = the time values exported are the time of the HMI device.</li> <li>• <b>Global</b> = the time values exported are in UTC format.</li> </ul>
<b>FileName</b>	<p>Enabled when the DateTimePrefixFileName=true</p> <p>The below wildcards are supported</p> <ul style="list-style-type: none"> <li>• %n = Trend name</li> <li>• %y = Year</li> <li>• %M = Month</li> <li>• %d = Day</li> <li>• %h = Hour</li> <li>• %m = Minutes</li> <li>• %s = Seconds</li> </ul> <p>Example: \%n\%y%M%d\%h%m%s</p>

Additional parameters available only when the selected FileFormat is **Compact CSV**



When both "Select Fields" and "Select Curves" parameters are empty, the .csv file is dumped in the old "Compact CSV" without columns' selection format. See also ["Exporting trend buffer data" on page 297](#)

Parameter	Description												
<b>Select Fields</b>	<p>Select the columns to export inside the dumped file.</p> <p>Available columns are:</p> <ul style="list-style-type: none"> <li>• DateTime</li> <li>• Date</li> <li>• Time</li> <li>• Value</li> <li>• Quality</li> </ul> <p>Note that "Attach to tag" can be used to define columns to be exported at the runtime from the HMI application. The tag must contain a string with the list of fields to be exported separated by commas.</p> <p>Example:</p> <ul style="list-style-type: none"> <li>• "" (<i>Empty string = all available fields</i>)</li> <li>• "DateTime,Value,Quality"</li> <li>• "Date,Time,Value"</li> </ul>												
<b>Select Curves</b>	<p>Select the curves to export inside the dumped file</p> <p>Note that "Attach to tag" can be used to define curves to be exported at the runtime from the HMI application. The tag must contains a string with the list of curve names to be exported separated by commas.</p> <p>Example:</p> <ul style="list-style-type: none"> <li>• Empty string or "All curves" will export all datasets</li> <li>• "Name1,Name2,Name3"</li> <li>• "Name1,Name3"</li> </ul>												
<b>Date Format</b>	<p>Select the Date and Time format</p> <p>Using "Attach to tag" is possible define the date format at runtime through a string</p> <table border="1" data-bbox="347 1447 1469 1888"> <thead> <tr> <th colspan="2" data-bbox="347 1447 1469 1503">Date Placeholder</th> </tr> </thead> <tbody> <tr> <td data-bbox="347 1503 850 1561"><b>d</b></td> <td data-bbox="850 1503 1469 1561">The day as number without a leading zero (1 to 31)</td> </tr> <tr> <td data-bbox="347 1561 850 1619"><b>dd</b></td> <td data-bbox="850 1561 1469 1619">The day as number with a leading zero (01 to 31)</td> </tr> <tr> <td data-bbox="347 1619 850 1709"><b>ddd</b></td> <td data-bbox="850 1619 1469 1709">The abbreviated localized day name (e.g. 'Mon' to 'Sun')</td> </tr> <tr> <td data-bbox="347 1709 850 1798"><b>dddd</b></td> <td data-bbox="850 1709 1469 1798">The long localized day name (e.g. 'Monday' to 'Sunday').</td> </tr> <tr> <td data-bbox="347 1798 850 1888"><b>M</b></td> <td data-bbox="850 1798 1469 1888">The month as number without a leading zero (1-12)</td> </tr> </tbody> </table>	Date Placeholder		<b>d</b>	The day as number without a leading zero (1 to 31)	<b>dd</b>	The day as number with a leading zero (01 to 31)	<b>ddd</b>	The abbreviated localized day name (e.g. 'Mon' to 'Sun')	<b>dddd</b>	The long localized day name (e.g. 'Monday' to 'Sunday').	<b>M</b>	The month as number without a leading zero (1-12)
Date Placeholder													
<b>d</b>	The day as number without a leading zero (1 to 31)												
<b>dd</b>	The day as number with a leading zero (01 to 31)												
<b>ddd</b>	The abbreviated localized day name (e.g. 'Mon' to 'Sun')												
<b>dddd</b>	The long localized day name (e.g. 'Monday' to 'Sunday').												
<b>M</b>	The month as number without a leading zero (1-12)												

Parameter	Description																																		
	<table border="1"> <thead> <tr> <th colspan="2">Date Placeholder</th> </tr> </thead> <tbody> <tr> <td><b>MM</b></td> <td>The month as number with a leading zero (01-12)</td> </tr> <tr> <td><b>MMM</b></td> <td>The abbreviated localized month name (e.g. 'Jan' to 'Dec').</td> </tr> <tr> <td><b>MMMM</b></td> <td>The long localized month name (e.g. 'January' to 'December').</td> </tr> <tr> <td><b>yy</b></td> <td>The year as two digit number (00-99)</td> </tr> <tr> <td><b>yyyy</b></td> <td>The year as four digit number</td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th colspan="2">Time Placeholder</th> </tr> </thead> <tbody> <tr> <td><b>h</b></td> <td>The hour without a leading zero (0 to 23 or 1 to 12 if AM/PM display)</td> </tr> <tr> <td><b>hh</b></td> <td>The hour with a leading zero (00 to 23 or 01 to 12 if AM/PM display)</td> </tr> <tr> <td><b>m</b></td> <td>The minute without a leading zero (0 to 59)</td> </tr> <tr> <td><b>mm</b></td> <td>The minute with a leading zero (00 to 59)</td> </tr> <tr> <td><b>s</b></td> <td>The second without a leading zero (0 to 59)</td> </tr> <tr> <td><b>ss</b></td> <td>The second with a leading zero (00 to 59)</td> </tr> <tr> <td><b>zzz</b></td> <td>The millisecond with leading zero</td> </tr> <tr> <td><b>z</b></td> <td>The millisecond</td> </tr> <tr> <td><b>AP</b></td> <td>Use AM/PM display. AP will be replaced by either "AM" or "PM".</td> </tr> <tr> <td><b>ap</b></td> <td>Use am/pm display. ap will be replaced by either "am" or "pm".</td> </tr> </tbody> </table>	Date Placeholder		<b>MM</b>	The month as number with a leading zero (01-12)	<b>MMM</b>	The abbreviated localized month name (e.g. 'Jan' to 'Dec').	<b>MMMM</b>	The long localized month name (e.g. 'January' to 'December').	<b>yy</b>	The year as two digit number (00-99)	<b>yyyy</b>	The year as four digit number	Time Placeholder		<b>h</b>	The hour without a leading zero (0 to 23 or 1 to 12 if AM/PM display)	<b>hh</b>	The hour with a leading zero (00 to 23 or 01 to 12 if AM/PM display)	<b>m</b>	The minute without a leading zero (0 to 59)	<b>mm</b>	The minute with a leading zero (00 to 59)	<b>s</b>	The second without a leading zero (0 to 59)	<b>ss</b>	The second with a leading zero (00 to 59)	<b>zzz</b>	The millisecond with leading zero	<b>z</b>	The millisecond	<b>AP</b>	Use AM/PM display. AP will be replaced by either "AM" or "PM".	<b>ap</b>	Use am/pm display. ap will be replaced by either "am" or "pm".
Date Placeholder																																			
<b>MM</b>	The month as number with a leading zero (01-12)																																		
<b>MMM</b>	The abbreviated localized month name (e.g. 'Jan' to 'Dec').																																		
<b>MMMM</b>	The long localized month name (e.g. 'January' to 'December').																																		
<b>yy</b>	The year as two digit number (00-99)																																		
<b>yyyy</b>	The year as four digit number																																		
Time Placeholder																																			
<b>h</b>	The hour without a leading zero (0 to 23 or 1 to 12 if AM/PM display)																																		
<b>hh</b>	The hour with a leading zero (00 to 23 or 01 to 12 if AM/PM display)																																		
<b>m</b>	The minute without a leading zero (0 to 59)																																		
<b>mm</b>	The minute with a leading zero (00 to 59)																																		
<b>s</b>	The second without a leading zero (0 to 59)																																		
<b>ss</b>	The second with a leading zero (00 to 59)																																		
<b>zzz</b>	The millisecond with leading zero																																		
<b>z</b>	The millisecond																																		
<b>AP</b>	Use AM/PM display. AP will be replaced by either "AM" or "PM".																																		
<b>ap</b>	Use am/pm display. ap will be replaced by either "am" or "pm".																																		
<b>Language</b>	Select the language to use.																																		



Note: execution of the DumpTrend action will automatically force a flush to disk of the data temporarily maintained in the RAM memory. See ["History trend widget" on page 300](#) for details on how to save sampled data to disk.



Note: external drives connected to USB port must have format FAT or FAT32. NTFS format is not supported.



**WARNING:** Be aware there are limits in the max number of files that can create inside a folder. Limits are depending of different factors and are not simple to calculate, you can think as 999 the max number of files that can be use inside a folder.



## To convert binary dump files to .csv

The TrendBufferReader.exe tool is stored in the *Utils* folder of the PB610 Panel Builder 600 installation folder.

Use the following syntax:

```
TrendBufferReader -r Trend1 Trend1.csv 1
```

where:

Trend1 = name of the trend buffer without extension resulting from the dump (original file name is trend1.dat)

Trend1.csv = name for the output file.



**WARNING: The TrendBufferReader.exe is an old utility that not work with the new multi tags buffers. Using of this utility is not recommendable. The utility is not more maintenanced because now there is the possibility to dump trend buffer directly in .csv format.**

## .csv file structure

The resulting .csv file has five columns

Column	Description
<b>Data Type</b>	Data type of sampled tag: 0 = empty 1 = boolean 2 = byte 3 = short 4 = int 5 = unsignedByte 6 = unsignedShort 7 = unsignedInt 8 = float 9 = double
<b>Value</b>	Value of the sample
<b>Timestamp (UTC)</b>	Timestamp in UTC format
<b>Sampling Time(ms)</b>	Sampling interval time in milliseconds
<b>Quality</b>	Tag value quality. Information coded according the OPC DA standard and stored in a byte data (8 bits) defined in the form of three bit fields; Quality, Sub status and Limit status.  The eight quality bits are arranged as follows: QQSSSSL. For a complete and detailed description of all the single fields, please refer to the OPC DA official documentation.

## Commonly quality values

The most commonly used quality values returned by the HMI acquisition engine are:

Quality Code	Quality	Description
0	BAD	The value is bad but no specific reason is given
4	BAD	Specific server problem with the configuration. For example, the tag has been deleted from the configuration file (tags.xml).
8	BAD	No value may be available at this time, for example the value has not been provided by the data source.
12	BAD	Device failure detected
16	BAD	Timeout before device response.
24	BAD	Communication failure
28	BAD	No data found for upper or lower bound value Trend interface specific flag.
32	BAD	No data collected (for example, archiving not active). Trend interface specific flag. This value is also used to indicate a temporary offline status (for any condition where sampling was stopped).
64	UNCERTAIN	No specific reason.
65	UNCERTAIN	No specific reason. The value has 'pegged' at some lower limit.
66	UNCERTAIN	No specific reason. The value has 'pegged' at some higher limit.
67	UNCERTAIN	No specific reason. The value is a constant and cannot move.
84	UNCERTAIN	Returned value outside its defined limits defined. In this case the <b>Limits</b> field indicates which limit has been exceeded but the value can move farther out of this range.
85	UNCERTAIN	Returned value outside its defined limits defined. In this case the <b>Limits</b> field indicates which limit has been exceeded but the value can move farther out of this range. The value has 'pegged' at some lower limit.
86	UNCERTAIN	Returned value outside its defined limits defined. In this case the <b>Limits</b> field indicates which limit has been exceeded but the value can move farther out of this range.

Quality Code	Quality	Description
		The value has 'pegged' at some higher limit
87	UNCERTAIN	Returned value outside its defined limits defined. In this case the <b>Limits</b> field indicates which limit has been exceeded but the value can move farther out of this range. The value is a constant and cannot move.
192	GOOD	-



## DeleteTrend




Deletes saved trend data.

Define the name of the trend from which you want to delete logs.

## DumpEventArchive

Stores historical alarm log and audit trail data to external drives, such as USB memory or SD card.

Parameter	Description
<b>EventArchive</b>	Name of buffer to dump data
<b>FolderPath</b>	Destination folder <ul style="list-style-type: none"> <li>• Internal = <i>\Flash\QTHMI\workspace\Dump</i></li> <li>• USB drive = <i>\USBMemory</i></li> <li>• SD Card = <i>\Storage Card</i></li> <li>• Public Network = <i>\\&lt;hostname or IP&gt;\sharePath</i></li> <li>• Private Network = <i>\\&lt;username&gt;:&lt;password&gt;@&lt;hostname or IP&gt;\sharePath</i></li> </ul> <p> Note: supported formats for external memory are FAT or FAT32 (NTFS format is not supported).</p> <p> Note: Private networks are supported only from Linux devices with BSP 1.0.25 and above.</p>
<b>DumpConfigFile</b>	Dump the description files of the archives
<b>DumpAsCSV</b>	<p><b>true</b> = the buffer is dumped to the specified location as a .csv file</p> <p><b>false</b> = the buffer is dumped in binary format (a .dat file and .inf file). Both these files are then required to convert data in .csv format by an external utility.</p>
<b>DateTimePrefix</b>	<b>true</b> = the dumped file will have date and time as prefix to its name (for example D2012_01_01_T10_10_alarmBuffer1.csv)

Parameter	Description
<b>timeSpec</b>	Time format: <ul style="list-style-type: none"> <li>• <b>Local</b> = the time values exported are the time of the HMI device.</li> <li>• <b>Global</b> = the time values exported are in UTC format.</li> </ul>
<b>csv COLUMNS</b>	Select the columns to dump into the .csv file. <p> Available only when the EventArchive is an alarms buffer</p>
<b>FileName</b>	The below wildcards are supported <ul style="list-style-type: none"> <li>• %n = Event archive name</li> <li>• %y = Year</li> <li>• %M = Month</li> <li>• %d = Day</li> <li>• %h = Hour</li> <li>• %m = Minutes</li> <li>• %s = Seconds</li> </ul> <p>Example: \%n\%y%M%d\%h%m%s</p> <p> Available only when the DateTimePrefixFileName=true</p>
<b>Language</b>	Select the language to use. <p> Available only when the EventArchive is an alarms buffer</p>
<b>Separate Date and Time</b>	When enabled the date and time are listed in separate columns.
<b>Date Format</b>	Select a predefined format or use placeholders to define your own format (see <a href="#">"Time and Date placeholders"</a> on page 472)

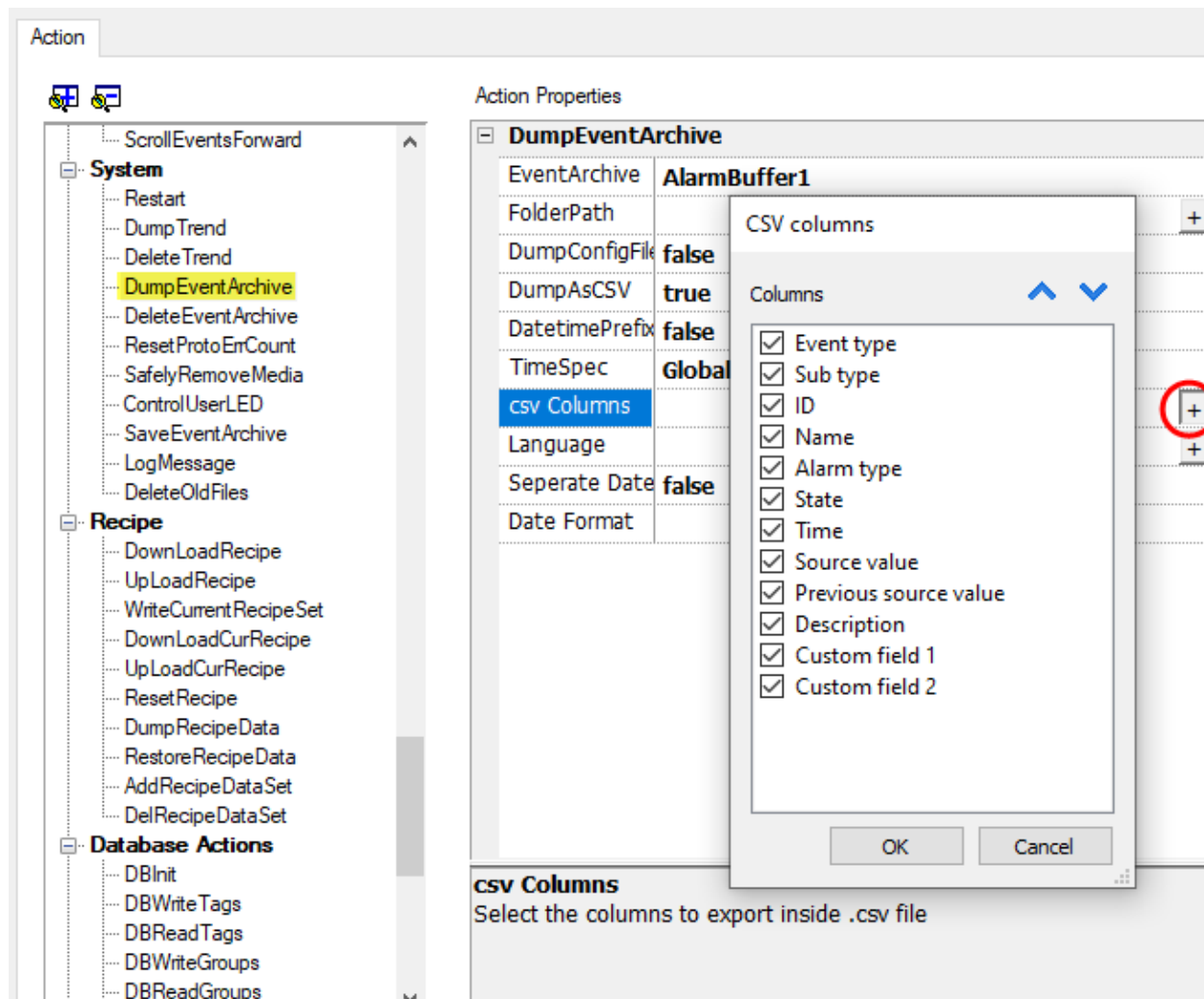
## Dumping in CSV Format

DumpAsCSV = true

For Alarms buffers, the additional "csv COLUMNS" parameter give the possibility to select the columns to export inside the .csv file



Note: available only for Alarms buffers.



## Dumping in BINARY Format

DumpAsCSV = false

When exporting Event buffers in binary format and **DumpConfigFile** is set to true (recommended settings), there are two folders:

- **data**, containing data files,
- **config**, containing configuration files for .csv conversion.

Once the two folders are copied from the USB drive to the computer disk, the folder structure will be:

`\config\`

`alarms.xml`

`eventconfig.xml`

`\data\`

`AlarmBuffer1.dat`

`AlarmBuffer1.inf`

`\`

### *AlarmBufferReader.exe*

#### To convert dump files to .csv

The AlarmBufferReader.exe tool is stored in the *Utils* folder of the PB610 Panel Builder 600 installation folder.

Use the following syntax:

```
AlarmBufferReader AlarmBuffer1 FILE ./AlarmBuffer1.csv
```

where:

- AlarmBuffer1 = name of the dumped .dat without extension
- AlarmBuffer1.csv = name for the output file.

The utility AuditTrailBufferReader.exe is available for Audit Trail buffers.

Use the following syntax:

```
AuditTrailBufferReader AuditTrail FILE ./AuditTrail.csv
```

where:

- AuditTrail = name of the dumped buffer without extension and
- AuditTrail1.csv = name for the output file.



**WARNING: The AlarmBufferReader.exe is an old utility that not work with newer buffer formats. Using of this utility is not recommendable. The utility is not more maintained because now there is the possibility to dump alarm buffer directly in .csv format.**

## DeleteEventArchive

Deletes saved Event buffers log data.

Specify the name of Event buffer to delete from the Event logs.

## ResetProtoErrCount

Resets the Protocol Error Count system variable.

See "[System Variables \(Attach To\)](#)" on page 143 for details.





## SafelyRemoveMedia

Provides for safe removal of SD card or USB drive fromHMI.

## SaveEventArchive

Save the records located within the audit trail to a signed file. The file signature will ensure that the records within the report are not altered.

Parameter	Description
EventArchive	Name of buffer to dump data
FolderPath	Destination folder <ul style="list-style-type: none"> <li>• Internal = \Flash\QTHMI\workspace\Dump</li> </ul>


Parameter	Description
	<ul style="list-style-type: none"> <li>• USB drive = <code>\USBMemory</code></li> <li>• SD Card = <code>\Storage Card</code></li> <li>• Public Network = <code>\\&lt;hostname or IP&gt;\sharePath</code></li> <li>• Private Network = <code>\\&lt;username&gt;:&lt;password&gt;@&lt;hostname or IP&gt;\sharePath</code></li> </ul> <p> Note: supported formats for external memory are FAT or FAT32 (NTFS format is not supported).</p> <p> Note: Private networks are supported only from Linux devices with BSP 1.0.25 and above.</p>
<b>FileName</b>	<p>The below wildcards are supported</p> <ul style="list-style-type: none"> <li>• %n = Event archive name</li> <li>• %y = Year</li> <li>• %M = Month</li> <li>• %d = Day</li> <li>• %h = Hour</li> <li>• %m = Minutes</li> <li>• %s = Seconds</li> </ul> <p>Example: <code>\%n\%y%M%d\%h%m%s</code></p>
<b>Format</b>	<p>Format of the output file</p> <ul style="list-style-type: none"> <li>• CSV</li> </ul>
<b>Signed</b>	<p>Generate the file signature.</p> <p> <b>On Linux devices, the BSP v1.0.239 or greater is required</b>  <b>On WinCE devices, the BSP v2.29 or greater is required</b></p> <p> The algorithm to use to signing is defined inside the project properties parameters  See "<a href="#">Project</a>" on page 87 for the available algorithms</p> <p>See also:</p> <ul style="list-style-type: none"> <li>• "<a href="#">Signed CSV files</a>" on page 358</li> </ul>
<b>TimeSpec</b>	<p>Time format:</p> <ul style="list-style-type: none"> <li>• <b>Local</b> = the time values exported are the time of the HMI device.</li> <li>• <b>Global</b> = the time values exported are in UTC format.</li> </ul>
<b>PeriodMode</b>	<p>Defines the time window to export</p> <ul style="list-style-type: none"> <li>• All events</li> </ul>

Parameter	Description
	<ul style="list-style-type: none"> <li>• Today</li> <li>• Yesterday</li> <li>• Last week</li> <li>• Last month</li> <li>• Current week</li> <li>• Current month</li> <li>• Custom</li> </ul> The additional parameters "periodFrom" and "periodTo" will be shown
<b>Separate Date and Time</b>	Uses two separate columns for Date and Time
<b>Date Format</b>	Select the Date and Time format

### Signed file

When the "Signed file" parameter is true, two files will be added in addition to filename.csv:

- filename.csv.sign  
The file signature will ensure that the records within the file filename.csv file have not been altered
- ssl-HMI.crt  
A copy of the certificate of the HMI device required to verify the authenticity of the report.

Name	Date modified	Type	Size
 AuditTrail-1413.csv	28/03/2018 16:13	Microsoft Excel Comma Separated Values File	1 KB
 AuditTrail-1413.csv.sign	28/03/2018 16:13	SIGN File	1 KB
 ssl-HMI.crt	28/03/2018 16:16	Security Certificate	2 KB

For more information about the certificate and how to verify signed files, see "[x.509 Certificate](#)" on page 354.

For more information about the exported information see "[Exporting audit trail as .csv files](#)" on page 381.

## LogMessage

Add a message into the audit trail buffer.

This macro give the possibility to developer to decide to keep track of some events (e.g. when a button is pressed, when a page is activate, etc.) into the audit trail. The attach to tag to have the possibility to define the message to log at runtime is supported.

Parameter	Description
<b>EventArchive</b>	Name of the audit buffer where add the message
<b>Message</b>	Message to add inside the audit buffer

## DeleteOldFiles

This macros delete files older that a give number of days.



In PC there is no restriction in using path. In panels it is allowed in dynamic media and data partition (/Flash in WinCE and /mnt/data in Linux)



**It will be developer responsibility to configure the application to avoid the possibility to delete system files.**

Parameter	Description
<b>FolderPath</b>	Folder where search the files to delete
<b>FileTypes</b>	List of files to delete separate by comma. Wildcard are supported Example: *.png,*.jpg
<b>OlderDays</b>	Minimum number of days without changes

## Tag actions

Interacts with tags.

### DataTransfer

Exchanges data between:

- two controllers,
- registers within a controller,
- from system variables to controllers,
- from controllers to system variables

The various tag types include a controller tag, a system variable, a recipe tag and widget property.

### ToggleBit

Toggles a bit value of a tag.

**BitIndex** allows you to select the bit to be toggled: toggling requires a read-modify-write operation; the read value is inverted and then written back to the tag.

### SetBit

Sets the selected bit to "1".

**BitIndex** allows you to select the bit position inside the tag.

### ResetBit

Resets the selected bit to "0"

**BitIndex** allows you to select the bit position inside the tag.

### WriteTag

Writes constant values to the controller memory. Specify tag name and value.

## StepTag

Increments or decrements tag value.

Parameter	Description
TagName	Name of tag to increase/decrease
Step	Step value
Do not step over limit	Enables step limit
Step Limit	Value of step limit, if enabled.

## BiStep

This action is similar to the StepTag action but the direction Increment/Decrement is automatically chosen by the rotation of the Wheel. Tag value will be increased when the Wheel is rotated clockwise. Tag value will be decreased in when the Wheel is rotated counterclockwise.

Property	Description
TagName	Name of Tag on which execute BiStep Tag action
Step	Value to be added/subtracted to Tag at every wheel rotation (depends on Event step property)
Event step	<p>This property allows to chose if adding/subtracting step values at every single wheel step, or at every rotation event.</p> <p><b>false</b> = The step value is added/subtracted to the Tag at every rotation event.  <i>Example: rotate the wheel performing 5 wheel steps in a single event, Tag will be increased/decreased by 1.</i></p> <p><b>true</b> = The step value is added/subtracted to the Tag at every single wheel step.  <i>Example: rotate the wheel performing 5 wheel steps in a single rotation, Tag will be increased/decreased by 5.</i></p>
Do not step over limit	If true, enables lower and upper limits, which represents the lower and the higher value that the Tag can assume due a BiStep Tag action
LowerLimit	If "Do not step over limit" is true, this property represents the lower value that the Tag can assume due a BiStep Tag action
UpperLimit	If "Do not step over limit" is true, this property represents the higher value that the Tag can assume due a BiStep Tag action



Available only inside OnWheel Actions

## ActivateGroup

Forces the update of a group of tags.

Tags are updated either when used in the current page or continuously, if defined as active in the Tag Editor. This action forces all the tags of a group to be continuously updated.

## DeactivateGroup

Deactivates a group of tags, that is stops forcing the update of a group of tags.

## EnableNode

Enable/disables action for offline node management. No communication is done with a disabled node.

Parameter	Description
Protocol ID	Unique identifier of selected protocol
NodeID	Node identifier in selected protocol. Can be attached to a tag.
Enable	Node communication status: <b>False</b> = disabled <b>True</b> = enabled When attached to a tag, tag = 0 means <b>False</b>

## BACnetClearPriority

Refer to the BACnet manual inside the “Communication Drivers” folder for a detailed description of BACnet actions.

## BACnetClearAllPriorities

Refer to the BACnet manual inside the “Communication Drivers” folder for a detailed description of BACnet actions.

## BACnetSetPriority

Refer to the BACnet manual inside the “Communication Drivers” folder for a detailed description of BACnet actions.

## ClearRetentiveMemory

When set to 0, clears the content of the Retentive Memory.

## ForceReadTag

Force a refresh of the specified tag from the remote controller.

# Trend actions

Used for Live Data Trends and Historical Trends Widget.

## RefreshTrend

Refreshes the **Trend** window.

It can be used in any Trends/Graphs widgets. Specify the widget as a parameter for the action.

## ScrollLeftTrend

Scrolls the **Trend** window to the left side, by one-tenth (1/10) of the page duration.



Note: with the real-time trends pause the trend using the **PauseTrend** action, or the window will be continuously shifted to the current value.

## ScrollRightTrend

Scrolls the **Trend** window to the right side, by one-tenth (1/10) of the page duration.



Note: with the real-time trends pause the trend using the **PauseTrend** action, or the window will be continuously shifted to the current value.

## PageLeftTrend

Scrolls the **Trend** window by one-page. For example, if the page size is 10 minutes, then use the **PageLeftTrend** action to scroll the trend left for 10 minutes.

## PageRightTrend

Scrolls the **Trend** window by one-page. For example, if the page size is 10 minutes, then use the **PageRightTrend** action to scroll the trend right for 10 minutes.

## ScrollUpTrend

Scroll the trend window up by 1/10 of the period.

## ScrollDownTrend

Scroll the trend window down by 1/10 of the time period.

## PageUpTrend

Scroll up the trend window page by page.

## PageUpTrend

Scroll up the trend window page by page.

## PageDurationTrend

Sets the page duration of the **Trend** window.

Define trend name and page duration.



Note: you can set page duration at runtime using a combo box widget.

## ZoomInTrend

Reduces page duration.

## ZoomOutTrend

Extends page duration.

## ZoomResetTrend

Reset the zoom level back to the original zoom level.

## ZoomInYAxisTrend

Reduces Y Axis.

## ZoomOutYAxisTrend

Extends Y Axis.

## ZoomResetYAxisTrend

Reset the Y Axis zoom level back to the original zoom level.

## PauseTrend

Stops plotting the trend curves in the **Trend** window.

When used with real time trend the plotting stops when the curve reaches the right border of the graph. This action does not stop trend logging.

## ResumeTrend

Resumes trend plotting if paused.

## ShowTrendCursor

Shows value of the curve at a given point on the X axis.

It activates the trend cursor. A cursor (vertical line) will be displayed in the trend widget.

When the graphic cursor is enabled, the scrolling of the trend is stopped.

The **ScrollCursor** action moves the graphic cursor over the curves, or over the entire **Trend** window.

## ScrollTrendCursor

Scrolls the trend cursor backward or forward.

The Y cursor value will display the trend value at the point of the cursor. Scrolling percentage can be set at 1% or 10%. The percentage is calculated on the trend window duration.

Parameter	Description
<b>Trend Name</b>	Select the trend widget ID.
<b>Scroll Direction</b>	Sets the scroll direction.
<b>Scroll Unit</b>	Sets the scrolling as a percentage of the displayed area or as a unit of time.

Parameter	Description
<b>Scroll Percentage</b>	Set the scroll rate as a percentage of the displayed area (used when Scroll Unit = Percentage)
<b>Scroll Timesample</b>	Set the scroll rate as time in seconds (used when Scroll Unit = Timesample)

## SetTrendView

Use this macro to change the axis ranges of the trend view.

When both Min X=0 and Max X=0, the static values defined inside the properties of widget are used. The same for the Y axe.

## ScrollTrendToTime

Scrolls the **Trend** window to a specified point in time.

Use this action when you need to scroll to a specific position in a trend window when a specific event occurred.

Parameter	Description
<b>Trend Name</b>	Select the trend widget ID.
<b>Scroll To Position</b>	Reference position for scrolling the trend window <ul style="list-style-type: none"> <li>• Start</li> <li>• Center</li> <li>• End</li> </ul>
<b>Scroll Trend to Time</b>	Time where move the trend window

### Example

1. Configure an action for an event (for example, an alarm) that executes a data transfer of the system time into a tag.
2. Select that tag as **ScrollTrendtoTime** parameter: the trend windows will be centered at the time when the event was triggered.

## ConsumptionMeterPageScroll

Scrolls the page backward or forward in a Consumption Meter widget.

Parameter	Description
<b>Trend Name</b>	Trend widget ID (for example, TrendWindow3)
<b>Page Scroll Direction</b>	Direction of page scrolling (Forward/backward)

## RefreshTrendTable

Update the trend table.

Parameter	Description
Data Source	The data source of the trend table to refresh

## ScrollTrendTableBackward

Scroll the trend table backward.

Parameter	Description
Data Source	The data source of the trend table to scroll backward

## ScrollTrendTableForward

Scrolls the trend table forward.

Parameter	Description
Data Source	The data source of the trend table to scroll forward

## Text Editor actions

Macros used to interact with the TextEditor widget.

Reference to ["TextEditor widget" on page 514](#) for details

## User management actions

User management and security settings.

### LogOut

Logs off the current user. The default user is then automatically logged in. If no default user has been configured, the logon window is displayed.

### SwitchUser

Switches between two users without logging off the logged user: the user login dialog appears. User can click **Back** to go back to the previously logged user.

User name:

Password:

Show password

Back

Sign In

The server continues running with the previously logged user, until the next user logs on. One user is always logged onto the system.

## ChangePassword

Change current user password: a dialog appears

No parameter is required.

## ResetPassword

Restores the original password together with the settings specified in the project for the current user.

No parameter is required.

## AddUser

*Reserved to users with **Can manage other users** property set.*

Adds a user at runtime: a dialog appears.

---

User name:	<input type="text" value="user3"/>
Password:	<input type="password" value="••••"/> <input type="checkbox"/> Show password
Group:	<input type="text" value="admin"/> ▼
Comments:	<input type="text"/>

---

<input type="checkbox"/> User must change his initial password
Inactivity logoff time (Min) <input type="text" value="0"/>
<input type="button" value="Add"/> <input type="button" value="Cancel"/>

---

## DeleteUser

*Reserved to users with **Can manage other users** property set.*

Deletes a user at runtime: a dialog appears.

No parameter is required.



---

User name:  ▼

Group:  ▼

## EditUsers

*Reserved to users with **Can manage other users** property set.*

Edits user settings.

---

User name:  ▼  Inactive

Password:   Show password

Group:  ▼

Comments:

---

User must change his initial password

Inactivity logoff time (Min)

---

### Inactive

If you set the *Inactive* flag, the user will no longer be able to log in

## DeleteUMDynamicFile

Deletes the dynamic user management file. Changes made to users settings at runtime are erased. The original settings are restored from the project information.

No parameter is required.

## ExportUsers

Exports user settings to an .xml file (*usermgnt\_user.xml*) in encrypted format to be restored when needed.

Set destination folder for the export file.



**Important: The user file is encrypted and cannot be edited.**



Note: supported formats are FAT or FAT32. NTFS format is not supported.

## ImportUsers

Imports user settings from a previously saved export .xml file (*usermgnt\_user.xml*).

Set source folder for the import file.



Note: supported formats are FAT or FAT32. NTFS format is not supported.

## Widget actions

### ShowWidget

Shows or hides page widgets.

Property	Description
Widget	Widget to show/hide

### SlideWidget

Shows the sliding effect of a widget, or of a widget group.



Note: The widget or grouped widgets can actually be outside of visible part of the page in the project and slide in and out of view.

Property	Description
Widget	Widget to slide
Direction	Sliding direction
Speed	Transition speed of sliding widget
X Distance	Travel distance of X coordinate in pixels
Y Distance	Travel distance of Y coordinate in pixels
Slide Limit	Enable/Disable movement limits of the widget with respect to the x, y coordinates
X Limit	Limit position of slide action for x coordinate
Y Limit	Limit position of slide action for y coordinate
Toggle Visibility	Show/hide widget at the end of each slide action
Image Widget	Image displayed during slide action

## BeginDataEntry

Displays a keypad and starts data entry on a data field without touching the widget itself. This action can be used to activate data entry using a barcode scanner.

### Java Script Interface

```
project.beginDataEntry(wgtName [, pageName])
```

Parameter	Description
<b>wgtName</b>	Widget name
<b>pageName</b>	Active page for data entry. Optional parameter. Useful to select a data field inside a non-modal active dialog box.

## TriggerIPCamera

"Force the refresh of IP Camera widget when used in JPEG format. Only works on pages that include an IP Camera widget"

## MoveIPCamera

Sends remote commands to a camera that supports them. See ["IP Camera widgets" on page 487](#) for details. Make sure that the IP Camera supports movement commands.

Parameter	Description
<b>Camera URL</b>	URL of IP Camera
<b>User Name</b>	Name of user allowed to access the camera. Set this parameter when access to the camera is password protected.
<b>Password</b>	Password to access the camera.
<b>Command</b>	Command to send to the PTZ controller (for example, decoder_control.cgi?command=0)

## RefreshEvent

Refreshes the event buffer (alarm or audit) of the widgets on the page

Parameter	Description
<b>Even Name</b>	Set the ID of the event buffer widget associated with Alarm History Widget or Audit Tables to be updated.

## ContextMenu

Displays the context menu.

If **Context Menu** property of Project Widget has been set to **On delay** context menu can appear also touching for a few seconds the background area of the screen. See ["Runtime" on page 80](#) for details.

## ReplaceMedia

Replaces existing media files with new files from USB/SD card. Can be used to replace video files of MediaPlayer widgets, or images of project.



Note: New media files must have same name and format of the files to be replaced.

Parameter	Description
<b>Media Type</b>	Type of file to update
<b>Device</b>	Device where new media files are supplied
<b>sourcePath</b>	Folder where new media files are stored (for example, "\USBMemory")
<b>Image Resize</b>	Resizes new images to the size of images to be replaced. Not applicable to video files.
<b>Silent</b>	Replaces media automatically. As default a dialog is displayed for the user to specify file location.

### Java Script Interface

```
void replaceMedia(var sourcePath, var bSilent, var Device, var nMediaType, var bResize)
```

```
project.replaceMedia("Images", true, "\USBMemory", 1, true);
```

## OpenComboBox

Open the combo box list. Works when Combobox mode is Full Screen.

## CloseComboBox

Close the combo box list. Works when Combobox mode is Full Screen.

## ScrollTable

Scroll rows of the table forward or backward.

Parameter	Description
<b>Table Widget</b>	Table widget name
<b>Direction</b>	The number of rows to jump, forward when positive, backward when negative.

### Java Script Interface

```
page.getWidget (TableWgt) .scrollTo (Direction) ;
```

## SelectAllAlarmsOnSrc

Toggle selection of all Alarms shown in table widget.

Parameter	Description
<b>Table Widget</b>	Table widget name

## ShiftTableDataSrcColumns

Shift left or right the columns of a data table. Note the remapping is applied to the data source widget.

Parameter	Description
<b>Data source widget</b>	Data source widget id
<b>Columns Shift</b>	Data source widget columns are shifted (left or right, depending on sign) by this amount
<b>Fixed left columns</b>	A custom amount of columns (on the left of table) can be kept fixed during shifting
<b>Remap Filter</b>	Table widget filter (if defined) is connected to a data source widget column. This column, by default, is not remapped by shift action, but can be forced to true

### Java Script Interface

```
var ColumnOrder = [0,1,2,3,4,5,6,7,8,9,10];
var json = {_c:ColumnOrder};

page.getWidget("TableDataSrcWgt").remapColumns(json);
```

## ResetTableDataSrcColumns

Restore the original columns order (see "ShiftTableDataSrcColumns" macro)

## SetTableSortingColumn

Select a column and the criteria to use to sort the rows of the table.

Parameter	Description
<b>Table Widget</b>	Table to sort
<b>Data Source Column</b>	Column to use to sort the table
<b>Sorting Mode</b>	Can be: Ascending, Descending or Toggle
<b>Sorting Type</b>	Can be: Alphabetic or Numeric

### Java Script Interface

```
var column = "Column1"; // Colum name (TableDataSource)
var mode = 0; //0=Ascending, 1=Descending
var type = 1; //0=Alphabetic, 1=Numeric

var sorting_rule_1 = { _c : column, _m : mode, _t : type };
var json = [ sorting_rule_1 ]
page.getWidget("TableWgt").setSortingRules(json);
```

## ChartCommand

Commands to control the charts widgets (Ref.: "[Scatter chart widget](#)" on page 304)

Parameter	Description
<b>ActivePanMode</b>	Activate the moving of the graphic through gesture commands
<b>ActiveZoomMode</b>	Activate the zoom mode through gesture commands
<b>ActiveCursorMode</b>	Activate the moving of the cursor through gesture commands
<b>ExecuteZoomIn</b>	Zoom In the graphic
<b>ExecuteZoomOut</b>	Zoom Out the graphic
<b>ExecuteAxesReset</b>	Reset the graphic view
<b>ExecuteNoAction</b>	Disable gesture commands
<b>ToogleAutoScale</b>	Enable or disable the auto-scale feature. Auto-scale ensures that the X axis maximum always take into account the most recent values of the curves.
<b>UpdateStaticCurve</b>	Reads tags values and refresh the curves



# 16 The HMI Client

---

HMI Client is a standalone application which provides remote access to the HMI Runtime, and is included in the PB610 Panel Builder 600. The HMI Client uses the same graphic rendering system as the runtime in the HMI devices, it relies on a specified HMI Runtime as server for live data.

HMI Client acts as a remote client and communicates to the server, sharing the local visualization with the tag values that are maintained or updated by the communication protocol.



HMI projects contain properties indicating which page is currently displayed on the HMI and can force the HMI to switch to a specific page. You can use these properties to synchronize pages showed on the HMI device and HMI Client or to control an HMI device with a PLC. See ["Project" on page 87](#) for details.



**To avoid unexpected behavior:**

- **be sure to use the same version of the HMI Runtime**
- **use "FreeType Font Rendering" to be sure to use the same font rendering engine on both HMI Client and HMI Device (see ["Runtime" on page 80](#))**



**Available only on Linux devices (See: ["HMI devices capabilities" on page 613](#))**

---

<b>Client application on PC</b> .....	<b>246</b>
<b>Client application on HMI</b> .....	<b>247</b>
<b>Settings and time zone options</b> .....	<b>248</b>

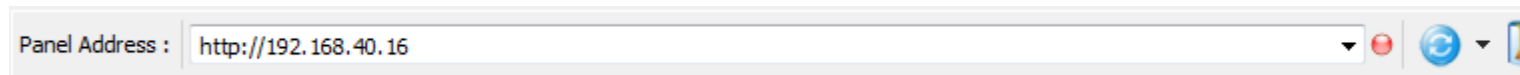


# Client application on PC

To run the HMI Client application on PC:

1. From the **Start** menu > **PB610 Panel Builder 600** > **HMI Client**: the client opens in a browser-like style window.
2. Type the server/device IP address in the address bar (for example: `http://192.168.1.12`): HMI Client will connect to the server and the same graphical application running on the device will be loaded in the client window.

## The Client application toolbar



Element	Description
HMI server address	IP address of the remote HMI device (e.g. 192.168.0.1:80)
Connection status	Network request status. Red during data exchange.
Reload from cache	Reloads project
BookMark	Bookmarks preferred pages and reload them.
Settings	Opens <b>Settings</b> dialog

## Reload options

Option	Description
F5	Reloads project from cache
Shift + F5	Downloads project to client

## Transferring files to a remote HMI device

You can upload and download files to and from a remote HMI device using two dedicated actions. These actions can only be used from a remote HMI Client and access remote files via FTP.

See ["Remote Client actions" on page 217](#) and ["Remote Client variables" on page 152](#).



**Important: Enable FTP support and give all necessary user rights to the folders used to transfer files.**

## Workspace

Project files are uploaded from the device and stored in HMI Client into the following cache folder.

`%appdata%\ABB\[build number]\client\cache`

where:

`[build number]` = folder named as build number, for example 01.90.00.608.

## Client application on HMI

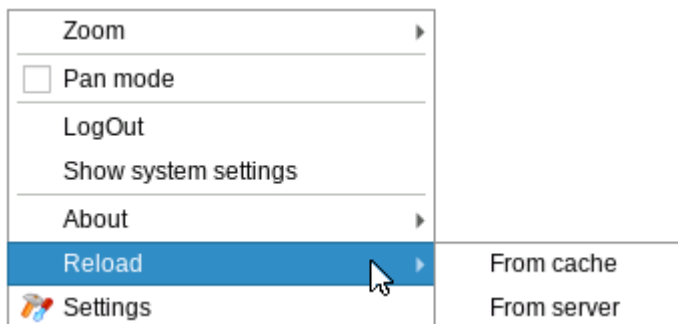
To run the HMI Client application on Linux HMI device:


1. From the **Run > Update Package** menu, create an Update Package and install the HMI Client application in to the HMI device (see ["Update package" on page 104](#) for additional information)
2. Type the server/device IP address in the Setting dialog that will be available when HMI device start (for example: <http://192.168.1.12>): HMI Client will connect to the server and the same graphical application running on the device will be loaded in the client window.



## Context Menu

The Context Menu, available with a right mouse click, will show the below commands:



Option	Description
<b>Zoom</b>	Select view size at runtime <ul style="list-style-type: none"> <li>• Zoom In</li> <li>• Zoom Out</li> <li>• Zoom 100%</li> </ul>
<b>Pane Mode</b>	Enables/disables pan mode after a zoom in
<b>Logout</b>	Logs off the current user.
<b>Show system settings</b>	Allow the HMI settings and the management of system components. See " <a href="#">System Settings</a> " on page 622 for details.
<b>Reload</b>	Reload remote project <ul style="list-style-type: none"> <li>• From cache</li> <li>• From server</li> </ul>
<b>Setting</b>	Open the HMI Client Settings. See " <a href="#">Settings and time zone options</a> " below for details  Could be password protected
<b>About</b>	Shows information about the HMI Client version.

## Settings and time zone options

In the **Settings** dialog you can configure client settings and decide how to display project time stamp information.

## Remote Server



### Connection settings

Parameter	Description
<b>Server Address</b>	IP address of the remote HMI device (e.g. 192.168.0.1:80)
<b>Auto connect at startup</b>	When the panel starts, use the Server Address to try to connect automatically to the remote server.
<b>Fit to screen size</b>	Adapts the view to the screen size

### Time settings

Parameter	Description
<b>Use Widget Defaults</b>	Displays time information according to the widget settings.
<b>Local Time</b>	Translates all timestamps in the project into the computer local time where the client is installed.
<b>Global Time</b>	Translates all timestamps in the project into UTC format.
<b>Server Time</b>	Translates all timestamps in the project into the same used by HMI device/server in order to show the same time.



**Important: Make sure you set the HMI RTC correct time zone and DST options.**

## Settings



### Interface Settings

Parameter	Description
<b>Context Menu Delay(s)</b>	Context menu activation delay. Range: 1–60 seconds.
<b>Show Busy Cursor</b>	Display an hourglass when the system is busy
<b>Use Keypads</b>	Display keypads when user touches a data entry field. Set to <b>disable</b> when an external USB keyboard is connected to the device.

### FTP settings

Parameter	Description
<b>Port</b>	FTP communication port

### HTTP settings

Parameter	Description
<b>Protocols</b>	Communication protocol used by HMI Client to communicate with an HMI device.
<b>Update Rate</b>	Polling frequency to synchronize data from server. Default = 1 s.
<b>Timeout</b>	Maximum wait time before a request is repeated by the HMI Client. Default = 5 s.
<b>Reuse connection</b>	<p>Enables reuse of the same TCP connection for multiple HTTP requests to reduce network traffic.</p> <p> Note: When enabled, this option may cause high latency if the proxy server does not immediately terminate old requests thus saturating connection sockets. This is often the case with 3G connections.</p>
<b>Enable compression</b>	<p>Compresses data to reduce download times. Default = disabled.</p> <p> <b>CAUTION: enabling this option could causes excessive CPU overhead.</b></p>
<b>Time Settings</b>	Used by the client to adapt the widget time stamp information.

## Password



The screenshot shows a 'Settings' dialog box with three tabs: 'Server', 'Settings', and 'Password'. The 'Password' tab is selected. Inside the dialog, there is a section titled 'Change password' containing three text input fields labeled 'Old password:', 'New password:', and 'Confirm password:'. At the bottom of the dialog are two buttons: 'OK' and 'Cancel'.

This dialog give the possibility to change the internal password of the HMI device for the admin user (the default password is "admin").

Password protection is not available on PC version of the HMI client



# 17 Using the integrated FTP server

HMI Runtime system uses an integrated FTP server.

Connect to the HMI device FTP server using any standard FTP client application. By default, the implicit FTP over TLS on port 990 is used.



**Important:** The server supports only one connection at a time; if you are using a multiple connection FTPs client disable this feature on the client program or set the maximum number of connections per session to 1.

## FTP settings

### FTP default credentials

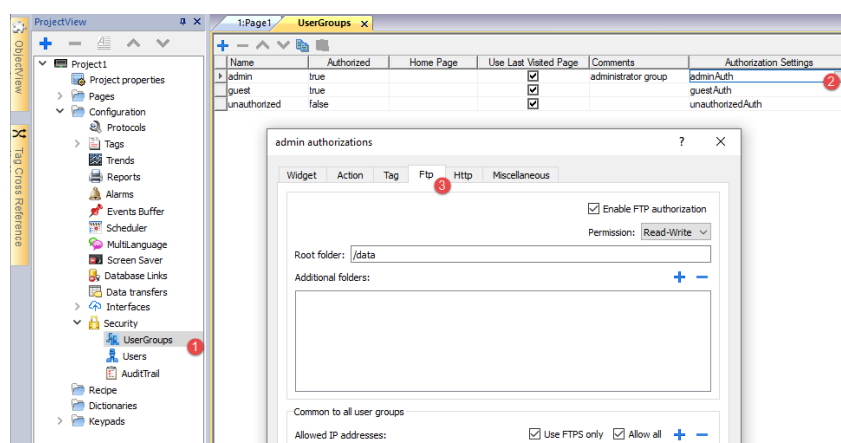
When User Management/Security is disabled use the following credentials for incoming connections:

<b>User name</b>	admin
<b>Password</b>	admin

### Changing FTP settings

**Path:** *ProjectView* > *Security* > *UserGroups* > *Authorization Settings*

You can change FTP permissions and account information in the **Ftp** tab of the **admin authorizations** dialog.



See "[FTP authorizations](#)" on page 367 for details.





# 18 Using VNC for remote access

---

VNC is a remote control software which allows you to see and control the HMI application remotely using your local mouse and keyboard.

Remote access is particularly useful for administration and technical support. In order to use it you need to:

- start a server in the HMI device
- install a viewer on the remote device

---

<b>Starting VNC server on WinCE devices</b> .....	<b>256</b>
<b>Starting VNC server on Linux devices</b> .....	<b>257</b>
<b>Starting VNC viewer</b> .....	<b>257</b>

# Starting VNC server on WinCE devices

VNC server is a plug-in. It can be enabled and downloaded as part of the Runtime. "[Plug-in](#)" on page 86.

## Installing VNC server

*Path: ProjectView > Project properties*

1. In the **Properties** pane set **VNC Server** to **true** to enable the plug-in.
2. Install or update the runtime to add the VNC server.

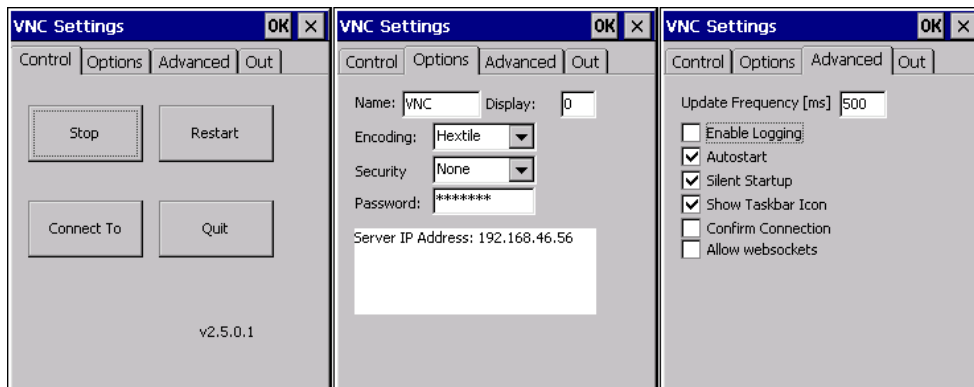
## Starting/stopping the VNC server

The VNC server is located in the folder `\Flash\qthmi\VNC` and is activated using the action `launchVNC`. If enabled from the project properties, it can also be activated from the runtime context menu **Developer tools > Launch VNC**.



To enable the runtime contextual menu see "[Project properties](#)" on page 79

## VNC Options dialog

From the **VNC Options** dialog you can perform several tasks.



**VNC should be disabled after use and autostart is not recommended.**

Tab	Functions
<b>Control</b>	Star/stop the VNC server and connect to viewer
<b>Options</b>	Define security information for server access using a VNC viewer
<b>Advanced</b>	<p>Enable automatic activation of VNC server at HMI device startup.</p> <p> Select <b>Silent Startup</b> to keep the <b>VNC Options</b> dialog in the background when <b>Autostart</b> is enabled.</p> <p> Enable <b>Show Taskbar Icon</b> when debugging out of KIOSK mode.</p>
<b>Out</b>	Contains the configuration settings for an outgoing connection to a listening VNC viewer software.



**Important: Settings in the Advanced tab are reserved to expert users and should be modified when the VNC server is used in conjunction with a VNC repeater to overcome firewall problems or optimize VNC performances according to the network configuration.**

## Connecting to viewer

Many modern VNC viewers offer the possibility to start the software in listening mode. The reason is that mobile devices most of the time do not have a public IP address to refer to. So it is practical to have a public IP address on an Office Computer which runs a listening VNC viewer. A user can then easily call for support by pressing the **Connect to viewer** button on the Control tab.

## VNC default settings

TCP port	5900
Password	null



**Important: The VNC server allows only a single client.**

## Starting VNC server on Linux devices

VNC server is a service embedded inside the BSP that can be activated from the Services tab of the device System Settings. See "[VNC Service](#)" on page 634 for details.



**VNC should be disabled after use and autostart is not recommended.**

## Starting VNC viewer

No VNC viewer is provided as part of PB610 Panel Builder 600.

Many compatible VNC viewers are available for free download (for example, TightVNC).



# 19 Alarms

---

The alarms handling system has been designed to provide alerts through pop-up messages, typically to display warning messages indicating any abnormal condition or malfunction in the system under control.

Whenever a bit changes, or the value of a tag exceeds a threshold set in the alarm configuration, a message is displayed. Specific actions can also be programmed to be executed when an alarm is triggered.



**Important: No default action is associated with any alarm.**

You can define how an alarm is displayed on the HMI device, if it requires user acknowledgment, and if and how it is logged into the event list.

Alarms are configured in the Alarms Configuration Editor and, thus, are available for all the pages of the project. An alarm widget can display more than one alarm at a time, if sized appropriately. You can trigger the opening or closing of the Alarm window with an event.

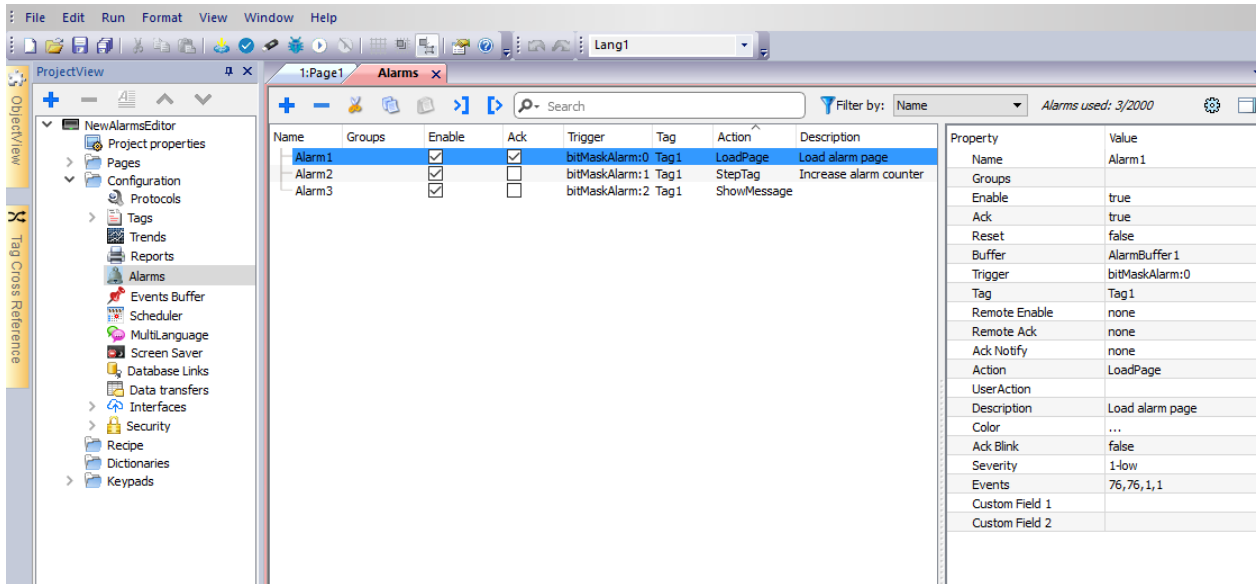
You work with alarms in the same way as you work with any other event. You may not want to display a dialog when an alarm is triggered and you can associate to it any other available action.

---

<b>Alarms Editor</b> .....	<b>260</b>
<b>Remote alarms acknowledge</b> .....	<b>263</b>
<b>Alarm state machine</b> .....	<b>264</b>
<b>Setting events</b> .....	<b>264</b>
<b>Active Alarms widget</b> .....	<b>267</b>
<b>Alarms History widget</b> .....	<b>272</b>
<b>Additional Alarms widgets</b> .....	<b>272</b>
<b>Managing alarms at runtime</b> .....	<b>273</b>
<b>Enable/disable alarms at runtime</b> .....	<b>274</b>
<b>Displaying live alarm data</b> .....	<b>274</b>
<b>Exporting alarm buffers to .csv files</b> .....	<b>276</b>
<b>Exporting alarm configuration</b> .....	<b>276</b>


# Alarms Editor



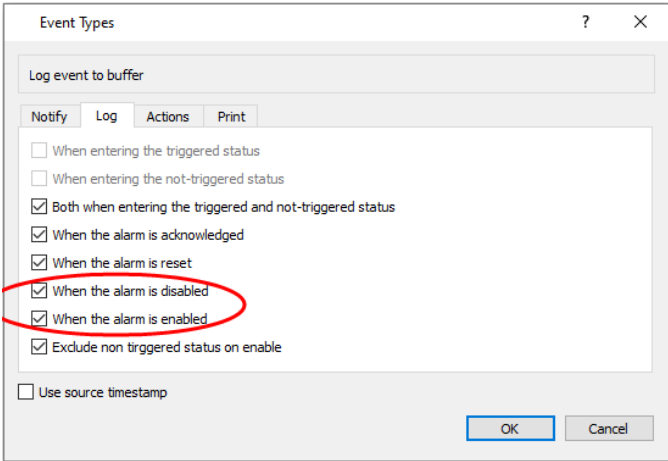
Path: **ProjectView**> **Config** > double-click **Alarms**



## Adding an alarm

Click **+** to add an alarm.

Parameter	Description
<b>Name</b>	Name of alarm
<b>Groups</b>	Groups associated with the alarm. They can be used in widgets display filters.
<b>Enable</b>	Enable/disable triggering of alarm.  Alarms can be enabled or disabled at runtime as well (see <a href="#">"Enable/disable alarms at runtime"</a> on page 274 for details).
<b>Ack</b>	Enable/disable acknowledgment of alarm, if selected the operator must acknowledge the alarm once triggered to remove it from the <b>Active Alarm</b> widget.
<b>Reset</b>	Used with the <b>Ack</b> option, if selected, acknowledged alarms stay in the alarm list, labeled as <b>Not Triggered Acked</b> , until the operator presses the <b>Reset</b> button in the alarm widget.
<b>Buffer</b>	Buffer file where the alarm history will be saved.
<b>Trigger</b>	Triggering condition depending on alarm type: <ul style="list-style-type: none"> <li>• <b>limitAlarm</b>: alarm triggered when tag value exceeds its limits. The alarm is not triggered if the value reaches the limits.</li> <li>• <b>valueAlarm</b> alarm is triggered when tag value is equal to the configured value</li> <li>• <b>bitMaskAlarm</b>: the bitwise AND operator compares each bit of the bitmask with the tag</li> </ul>

Parameter	Description
	<p>value corresponding to that Alarm. If both bits are on, the alarm is set to true. You can specify one or more bit positions (starting from 0) inside the tag. The Bit position must be given in decimal format; if more bits are specified, each position must be separated by a ",".</p> <ul style="list-style-type: none"> <li> <b>deviationAlarm</b>: alarm triggered if the percentage of deviation of the tag value from the set point exceeds a set deviation.                     <math display="block"> Value_{now} - SetPoint  &gt; \left( \frac{deviation}{100} \times SetPoint \right)</math> </li> <li> <b>protAlarm:BACN</b> : alarm is triggered from the remote BACnet device.                        This trigger mode is available only when the protocols configuration contains a BACnet device. Refer to the BACnet manual inside the "Communication Drivers" folder for a detailed description of BACnet alarm events.                 </li> </ul>
<b>Tag</b>	<p>Tag whose value will trigger the alarm when it exceeds the set limits.</p> <p>The alarm can refer to the value of this tag, or to the state of a bit if <b>bitMaskAlarm</b> has been selected as trigger.</p>
<b>Remote Enable</b>	<p>Tag used by the PLC to enable/disable the alarm.</p> <ul style="list-style-type: none"> <li>Changing the enable status from the Alarms Widget will change the tag value</li> <li>When the tag cannot be read (e.g. communication error) the alarm is disabled</li> <li>No tags related to the alarm are refreshed when alarm is disabled.</li> </ul> <p> Tip: It could be useful to enable the logging of the alarm's enable flag</p> 
<b>Remote Ack</b>	<p>Tag used by the PLC to acknowledge the alarm. A transition of this tag from 0 to a non zero value is considered an acknowledgment request.</p> <p>Leave empty if remote acknowledgment is not required.</p> <p>See "<a href="#">Remote alarms acknowledge</a>" on page 263 for details.</p>
<b>Ack Notify</b>	<p>Tag used by the HMI device to notify when the alarm is acknowledged from the device or from</p>



Parameter	Description
	<p>the PLC.</p> <p>0 = set to this value when alarm is triggered</p> <p>1 = set to this value when alarm is acknowledged.</p>
<b>Touch Ack Notify</b>	<p>Tag used by the HMI device to notify when the alarm is acknowledged from the device..</p> <p>0 = set to this value when alarm is triggered</p> <p>1 = set to this value when alarm is acknowledged.</p>
<b>Action</b>	<p>Actions executed when the alarm is triggered. Additional conditions can be specified in the <b>Events</b> column. See "<a href="#">Setting events</a>" on page 264 for details.</p> <p>The macros added in the action field are executed on the server-side with the exception of the below macros that will be executed even on client-side (e.g. PB4Web).</p> <ul style="list-style-type: none"> <li>• loadPage</li> <li>• prevPage</li> <li>• nextPage</li> <li>• showDialog</li> <li>• showMessage</li> <li>• setLanguage</li> <li>• jsAction</li> </ul>
<b>User Action</b>	<p>Actions executed when user press the action button in the active alarm widget.</p> <p>See "<a href="#">Active Alarms widget</a>" on page 267 for details.</p>
<b>Description</b>	<p>Alarm description. This text supports the multiple language features and can be a combination of static and dynamic parts, where the dynamic portion includes one or more tag values.</p> <p>See "<a href="#">Displaying live alarm data</a>" on page 274 for details.</p>
<b>Custom Field #</b>	<p>It is an additional alarm description that can be used to show additional information inside the alarms widgets. For example, could be an index to use to show a picture related with the alarm.</p>
<b>Color</b>	<p>Foreground and background colors of alarm rows based on the status of alarm.</p>
<b>AckBlink</b>	<p>Blinking for triggered alarms. If selected the alarm rows blinks until acknowledged. Only effective if <b>Ack</b> is selected.</p>
<b>Severity</b>	<p>Severity of the alarm. If multiple alarms are triggered simultaneously, actions will be executed based on severity settings.</p> <p>0 = not important</p> <p>1 = low</p> <p>2 = below normal</p> <p>3 = normal</p> <p>4 = above normal</p>

Parameter	Description
	5 = high 6 = critical
Events	Conditions in which the alarms are notified, logged or printed. See " <a href="#">Setting events</a> " on the next page for details.

## Backup alarms events

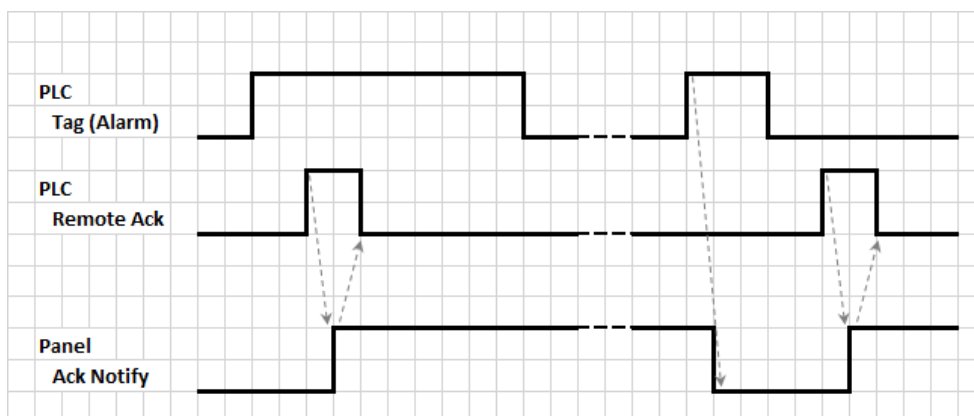
From the "[Events Buffer](#)" on page 281 you can configure the size of the alarms buffer and activate the backup of the alarms events when the buffer is full.

## Remote alarms acknowledge

When the **Remote Ack** parameter is set, an alarm can be acknowledged from a PLC device setting a tag value to a nonzero value. The acknowledged status is notified to the PLC device by the **Ack Notify** flag.

### Alarms acknowledgement process

**Remote Ack** tag is set/reset by the PLC to request the acknowledge, and **Ack Notify** is set/reset by HMI device to notify the execution of the acknowledge.



1. When an alarm condition is detected the HMI device set **Ack Notify** to 0 and all related actions are executed.
2. When the alarm is acknowledged (by HMI device or remotely), **Ack Notify** is set to 1
3. It's up to the controller to set **Remote Ack** to 1 to acknowledge the alarm or reset it to 0 when the HMI device send a notification that the alarm has been acknowledged (**Ack Notify** = 1)



**WARNING:** When an alarm is triggered, some signals need to be update/communicated through the connected devices. We assume the Acknowledge to be a signal pushed from an operator and not released automatically from a controller device. This allows for time required to communicated the original signals.

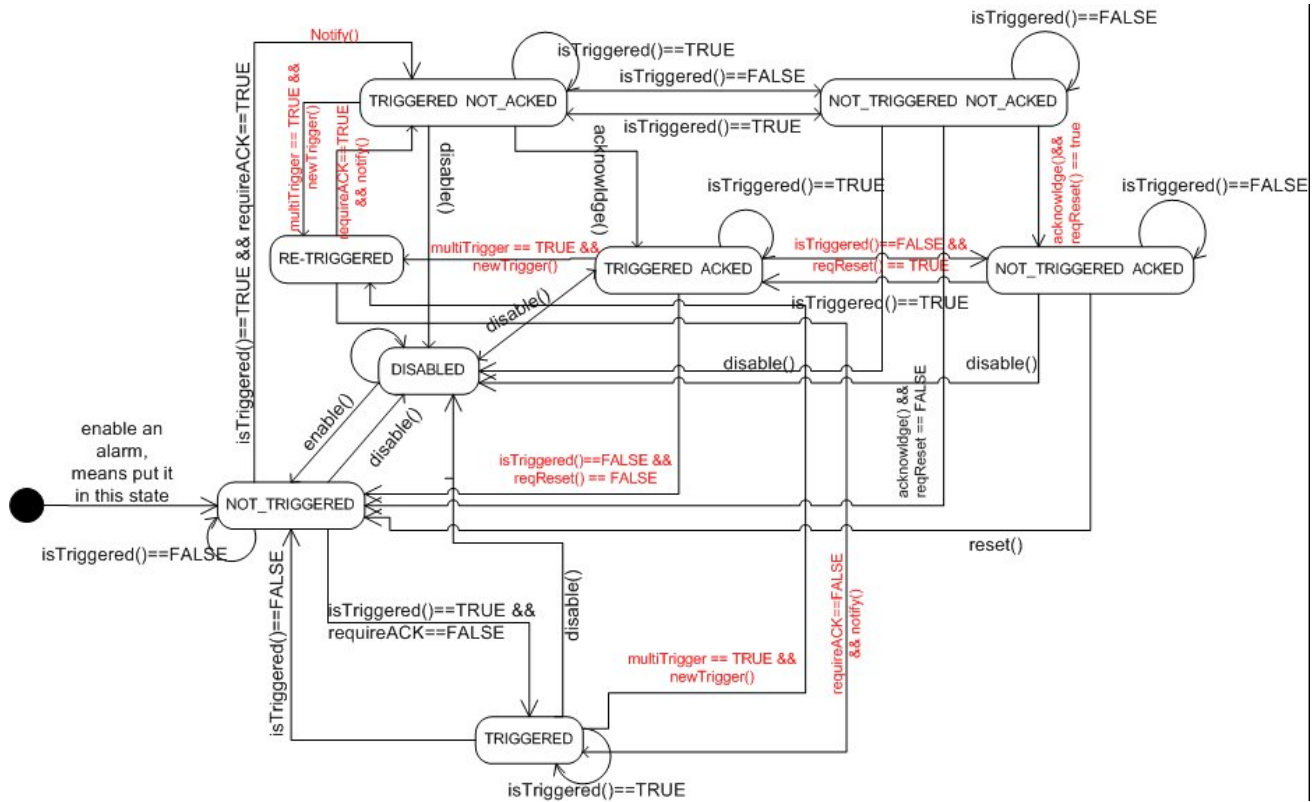


**Tip:** Using the same tag both for **Remote Ack** and **Ack Notify** can connect more devices to the same controller and acknowledge the alarms from any HMI device.

# Alarm state machine

The runtime implements the alarm state machine described in this diagram.

States and transitions between states are described according to the selected options and desired behavior.



## Setting events

Path: **ProjectView** > **Config** > **Alarms** > **Events** property

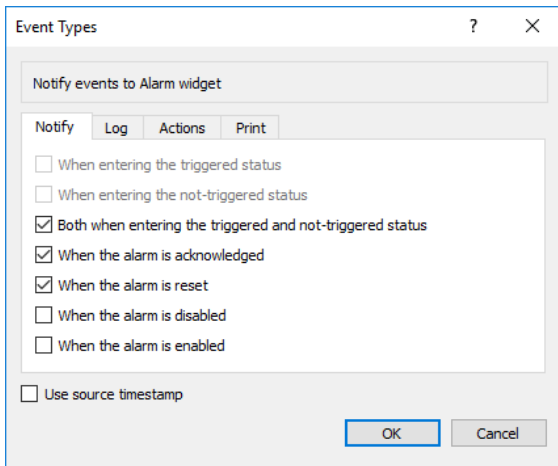
Events are defined using the Alarms Editor.

See "[Alarms Editor](#)" on page 260 for details.

## Notifying events

Path: **ProjectView** > **Config** > **Alarms** > **Events** property > **Notify** tab

Set conditions under which the alarms will be posted in the alarm widget.



Here you define the behavior of the default alarm widget available in the Widget gallery and decide in which cases the widget is updated by a change in an alarm status.

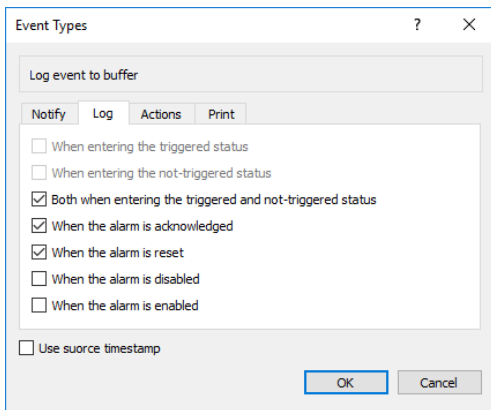


**CAUTION:** Make only the adjustments required by the specific application while leaving all other settings as default.

## Logging events

Path: **ProjectView** > **Config** > **Alarms** > **Events** property > **Log** tab

Set conditions for which you want to store the specific event in an alarm history buffer.

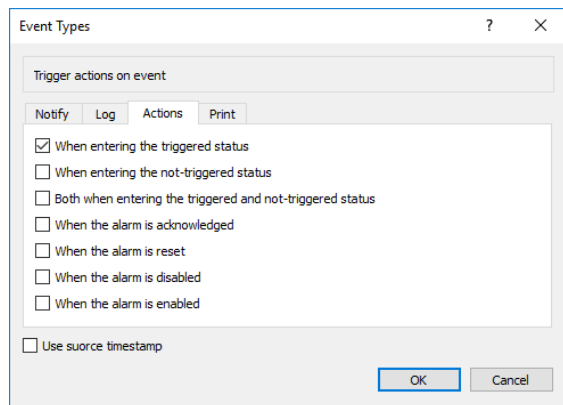


The alarm history is logged in the Event Buffer.

## Executing actions

Path: **ProjectView** > **Config** > **Alarms** > **Events** property > **Actions** tab

Set conditions under which the action(s), configured for the specific alarm, must be executed.

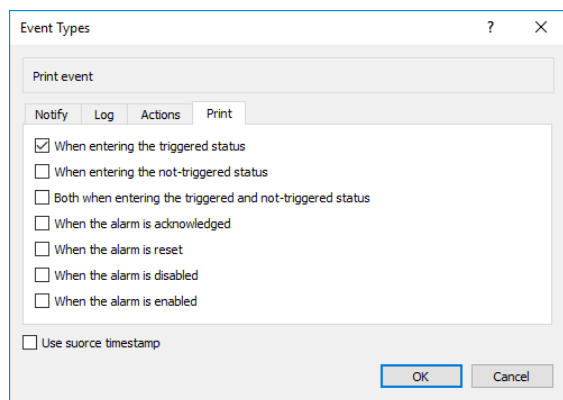


By default, actions are executed only when the alarm is triggered; other alarm states can also be set to execute actions.

## Print events

Path: **ProjectView** > **Config** > **Alarms** > **Events** property > **Print** tab

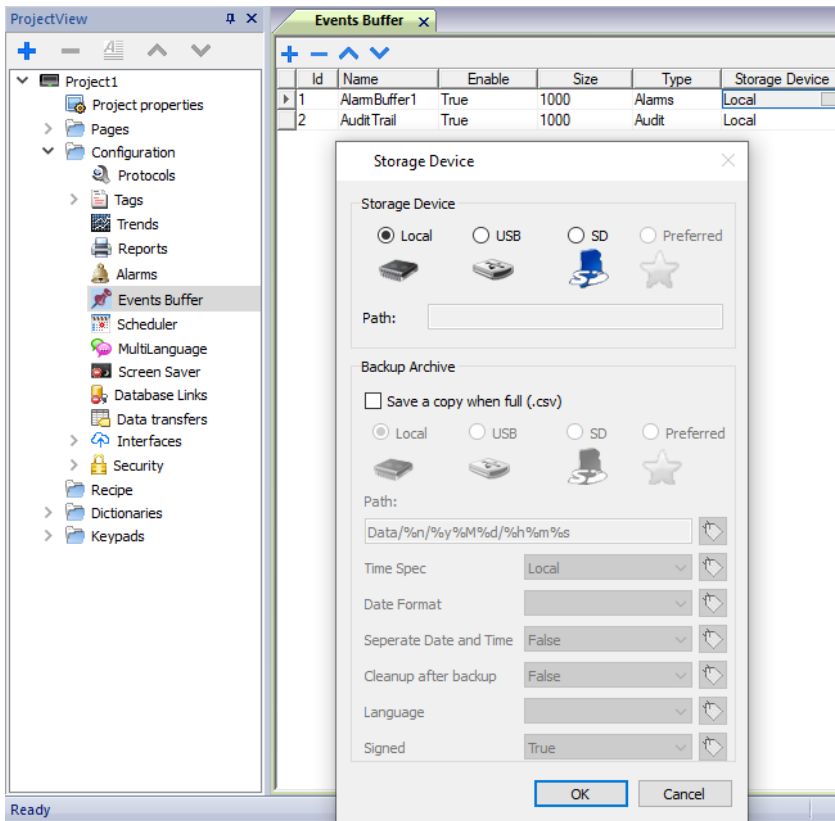
Set conditions for which you want to print the specific event



## Setting storage device

Path: **ProjectView** > **Config** > **Events Buffer** > **Storage Device** tab

1. Open the **Storage Device** dialog.
2. Select a device for event data storage.



Data is automatically saved every five minutes except for alarm data which is saved immediately.

### Use source timestamp

Events are stored with the timestamp of when the HMI device detects the event. When “Use source timestamp” is selected, the events are stored with the timestamp received from the remote device.



Available only for device’s protocols that support this feature (OPC UA Client and BACnet)

## Active Alarms widget

You can insert the **Active Alarms** widget in a page to display the alarms and to acknowledge, reset or enable/disable them.

Active Alarms

Select	Name	State	Value	Time	Description	Severity	Enable

Filter :

## Alarm filters

*Path: ActiveAlarm widget > Properties pane > Filter*

Define filters used to display only some of the configured alarms. Filters are based on alarm fields, which means you can filter alarms according to name, severity, description and so on.

Filter 1 is the default filter. It's managed by the combo box **Filter 1**, and has two options: **Show all alarms** and **Hide Not Triggered** which, when selected, allows to display only active alarms.

Filter 2 is, by default, not configured and available for customization.

Filter's expressions make use of AWK language, the expressions are applied to the data contained in the selected **Filter** column of the Alarm widget.

Alarms List	
Columns	
Sorting	false
Sort Column	Severity
+ Text	
- Filter	
Filter Column	State
- Filter 1	Hide Not Triggered
DataLink	itemData:Combo2
Filter Column	Select
Filter 2	

## Setting filters

*Path: ActiveAlarm widget > Properties pane > Filter*

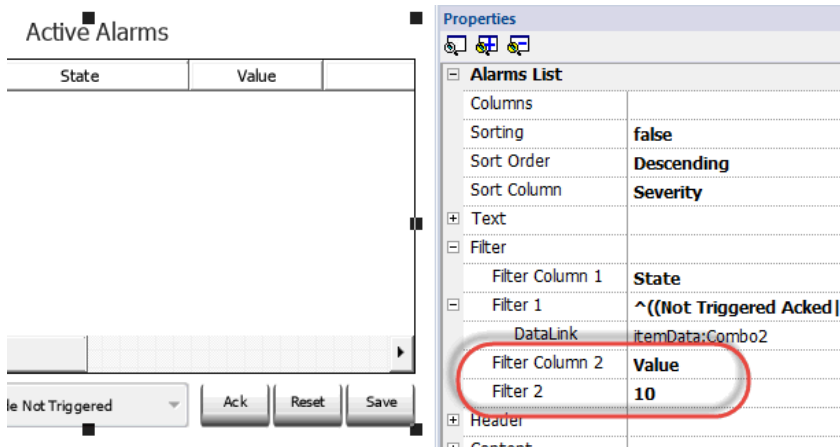
To set one of the two available filters:

1. Select **Filter Column 1** and choose the value to filter for (e.g.: Name, State, Time, Groups)
2. In **DataLink** attach a combo box widget. Use Shift+ left-click to select the combo box.
3. In the **Properties** pane select list property and open dialog to customize combo box values
4. In the combo box configuration dialog, specify **String List** and the regular expression to filter values.

See [https://en.wikipedia.org/wiki/Regular\\_expression](https://en.wikipedia.org/wiki/Regular_expression) for additional details regarding regular expressions.

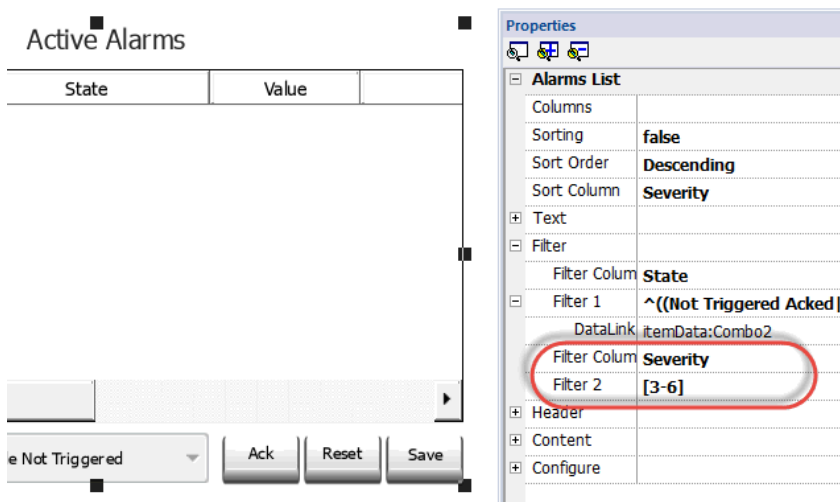
### Filters first example

You want to show all alarms matching Filter 1 with value equal to 10. Then properties settings: **Filter column 2 = Value, Filter 2 = 10**



### Filters second example

You want to show all alarms matching a Severity value from 3 to 6 (Normal to Critical). Then properties settings: **Filter column 2 = Severity, Filter 2 = [3-6]**



### Filters third example

You want to show all alarms matching a value from 11 to 19. Then properties settings: **Filter column 2 = Severity, Filter 2 = ^(1[1-9]\$)**

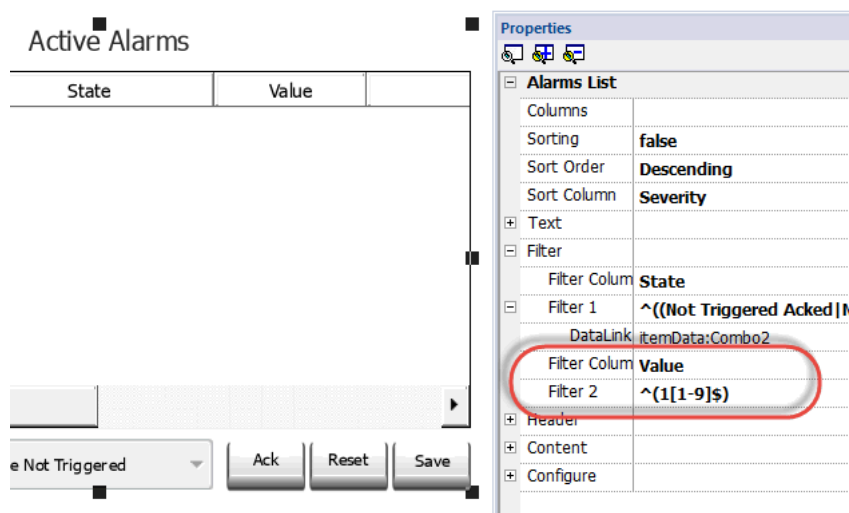
Meaning:

^ = match must starts from the beginning of the string

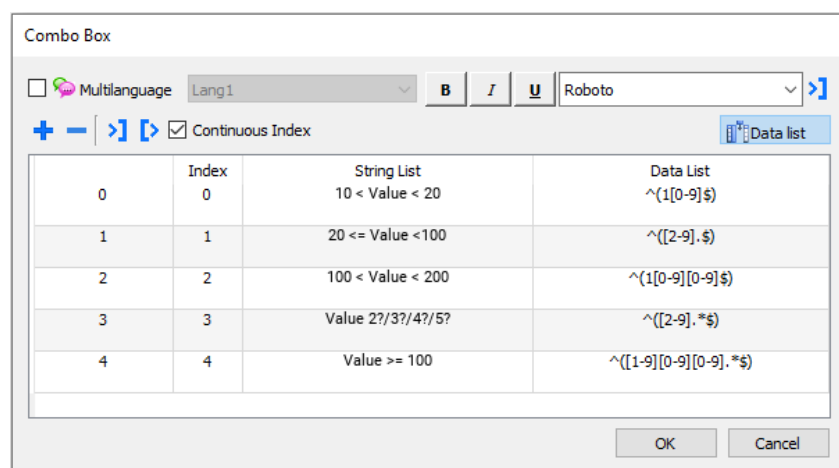
1[1-9] = first char must be 1 and the second char must be between 1 and 9

\$ = end of the comparison.





## Filters expression examples



Filter by	String list	Data list
<b>State</b>	Hide Not Triggered	^((Not Triggered Aced Not Triggered Not Aced Triggered).*\$)
<b>Value</b>	10 < Value < 20	^(1[0-9]\$)
<b>Value</b>	20 <= Value <100	^([2-9].)\$)
<b>Value</b>	100 < Value < 200	^(1[0-9][0-9]\$)
<b>Value</b>	Value 2?/3?/4?/5?	^([2-9].*\$)
<b>Value</b>	Value >= 100	^([1-9][0-9][0-9].*\$)
<b>Value</b>	Value >= 20	^([2-9].*\$)[1-9][0-9][0-9].*\$)

## Sorting alarms

**Path:** *ActiveAlarm* widget > *Properties* pane > **Sorting**

The sorting function allows you to sort alarms at runtime in the alarms widget by clicking on the column header.



Note: The severity value displayed here is set in the Alarm Editor.

## Action

When the "User Action" associate with the alarm (see ["Alarms Editor" on page 260](#) for details) contains valid actions, the Action icon is showed. Pressing the icon, the configured actions will be executed.

**Active Alarms**

Action	Name	State	Time
	Alarm1	Not Triggered	03/08/2016 11:07:43 AM
	Alarm2	Triggered	03/08/2016 11:07:55 AM
	Alarm3	Not Triggered	03/08/2016 11:07:43 AM

Filter :



**WARNING:** If you are using an older converted project, you have to substitute the old Active Alarms Widget with the new one from the Widgets gallery



Note: The image can be modified from the Columns property of the Active Alarms widget

Table Column Editor

Columns + - ^ v

- Action
- Select
- Enable
- Name
- Groups
- State
- Value
- Time
- Description
- Severity

Col 0 Info

Header	Action
Value	alUserAction
Width	100
Type	Image
Visible	true
Image path	images\action.png

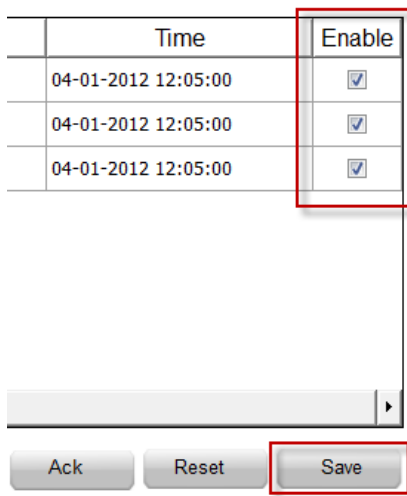
Alarms List : ActiveAlarms

Columns	
Sorting	false
Sort Order	Descending
Sort Column	Severity
Text	
Filter	
Header	
Content	
Configure	
General	
Position	

## Enable/Disable Alarms

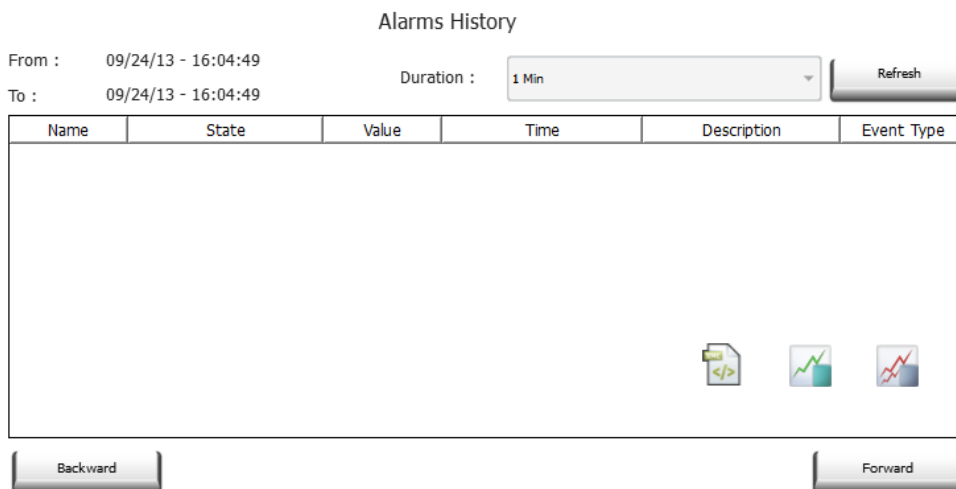
At runtime the Alarms Widget can be used to enable or disable the alarms.

Saves changes made in the **Enable** column in the alarm widget. This action is used with the **Save** button in the alarm widget.



## Alarms History widget

Logs and display an alarm list if **Buffer** property in Alarms Configuration Editor is set.



### Attaching widget to buffer

Path: **AlarmHistory** widget> **Properties** pane> **Buffer** > **EventBuffer**

In **Properties** pane > **Event** select the **Event Buffer** from which the alarm list is retrieved

## Additional Alarms widgets

In addition to the two main “Active Alarms” and the “Alarms History” widgets, the Gallery contains some other alarms widgets with a slightly different look but basically similarly at the two main widgets. You are free to choose and use the widget that has the look that better meet your requirements.



Note that some widgets are available even inside the print report gallery.

Some widgets are based on the new table structure. For these widgets, in addition to the exposed properties, you can select the internal table and use the table capabilities to modify the widget as for your needs and taste (see "[Table widget](#)" on page 490 for additional details).

## Printing the historical alarms list

The print gallery contains historical alarms widgets, based on table structure, that can be used to generate an alarms report. The table can be drawn and enlarged to fill the entire page. If the number of lines to printed is greater of one page, the alarms table will be printed using additional pages.

Timestamp	Name	State	Value	Description
Label	Label	Label	Label	Label

Using the "attach to tag" feature is possible to use tags to define some properties of the historical alarms list to print at runtime:

- Page Duration
- End Time

"Page Duration" with "End Time" define the piece of the alarm buffer to print.

**Properties**

**AlarmsHistoryReport : AlarmsHistoryReport**

- EventBuffer: AlarmBuffer1
- Page Duration: 1 hour
- DataLink: Duration
  - Access Type: R
- End Time: 0
- DataLink: EndTime
  - Access Type: R
  - Time Spec: local
  - Date Format: MM/DD/YY - hh:mm:ss
- Filter

**1:Page1 Tags**

Name	Address
Duration	Duration int
EndTime	EndTime time

## Managing alarms at runtime

When an alarm is triggered it is displayed in the Active Alarms widget where you can acknowledge and reset it. You can filter the alarms displayed using several filters, for example you can hide not triggered alarms or show all alarms.

See "[Active Alarms widget](#)" on page 267 for details.



**IMPORTANT:** The Active Alarms widget is not displayed automatically. You must add a dedicated action that will open the page containing the alarm widget when the alarm is triggered.

## Enable/disable alarms at runtime

You can enable or disable the alarms at runtime.

To enable an alarm select the **Enable** option in the alarm widget.

Disabled alarms are not triggered and therefore not displayed at runtime.

Select	Id	Source Value	State	Date	Time	Enable
<input type="checkbox"/>	Alarm1	23	Not Triggered Not Acked	25-01-2011	16:59:31	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Alarm2	23	Not Triggered Not Acked	25-01-2011	16:59:31	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Alarm3	23	Not Triggered Not Acked	25-01-2011	16:59:31	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Alarm4	23	Not Triggered Not Acked	25-01-2011	16:59:31	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Alarm5	23	Not Triggered Not Acked	25-01-2011	16:59:31	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Alarm6	23	Not Triggered Not Acked	25-01-2011	16:59:31	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Alarm7	23	Not Triggered Not Acked	25-01-2011	16:59:32	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Alarm8	23	Not Triggered Not Acked	25-01-2011	16:59:32	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Alarm9	23	Not Triggered Not Acked	25-01-2011	16:59:32	<input checked="" type="checkbox"/>

Check/Uncheck All   Filter: Show All   Ack   Reset   Save



Note: Alarms can be configured to be enable/disable even from the PLC.  
See [Alarm Configuration Editor](#) for details.

## Displaying live alarm data

Path: **ProjectView > Config > double-click Alarms**

Both in the Active Alarms widget and in the Alarms History widget it is possible to set the description of the alarm, or of the custom fields, to display the data of the live tags.

Id	Name	Enable	Ack	Reset	Tag	Buffer	Trigger	Action	Description
1	Alarm 1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Tag1	AlarmBuffer1	bitMaskAlarm:	ShowDialog	Alarm 1 Tag Value is [Tag1]
2	Alarm2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Tag1	AlarmBuffer1	bitMaskAlarm:1	ShowDialog	Alarm 2 Tag Value is [Tag2]
3	Alarm3	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Tag1	AlarmBuffer1	bitMaskAlarm:1	ShowDialog	Alarm 3 Tag Value is [Tag3]

To show the tag value, set a placeholder in **Description** entering the tag name in square brackets, for example "[Tag1]". At runtime, in **Description** column of Active Alarms widget the current value of the tag will be displayed.

### Live Tags Placeholders

#### Tags

- [TagName]  
The tag value is read and continuously updated



Use '\' before '[' if you want to show the '[' in the description string, for example: `\[Tag\1\]` will display the string "[Tag1]".

Use '\', even when the tag label contains square brackets. For example, to display the live tag value of tag "TAG]3" or "TAG[3]" use:

- TAG\]3 = [TAG]3
- TAG[3\] = [TAG[3]

#### Array Tags

To reference the entire array (all elements will be shown):

- [TagName]  
All array elements will be displayed using a comma separate list.
- [TagName[-1]]  
All array elements will be displayed using a comma separate list.

To reference an element of the array:

- [TagName.Index]  
Example: [MyARRAY.5] will display the sixth element of the MyARRAY
- [TagName[TagIndex]]  
Example: [MyARRAY[TagIndex]] will display the sixth element of the MyARRAY when TagIndex is 5

### Data Formats

Placeholder characters can be used to control how to display the tag value (see ["Custom Formats" on page 38](#))

- [TagName|format("###")]

Example:

Live: [fCounter|format("#.00")] - Triggered: [!fCounter|format("#.00")]



Note that by default, all tags are displayed as an integer. If you want to display a float number, you have to specify how to show the number adding the decimal digits.

### To freeze a live tag value

Live tags are read and continuously updated. If you want to freeze the tag value at the instant the alarm is triggered, use the exclamation point as tag name prefix:

- [TagName]  
When alarm is triggered, tag value is read and continuously updated
- [!TagName]  
When alarm is triggered, tag value is read and frozen

### Example of Alarm widget

Select	Name	State	Value	Time	
<input type="checkbox"/>	Alarm1	Triggered Not Acked	1	30/09/2019 12:56:19	Live Counter: 44
<input type="checkbox"/>	Alarm2	Triggered Not Acked	1	30/09/2019 12:56:21	Triggered Counter: 11
<input type="checkbox"/>	Alarm3	Triggered Not Acked	1	30/09/2019 12:56:24	Live: 44 - Triggered: 14
<input type="checkbox"/>	Alarm4	Triggered	1	30/09/2019 12:56:35	Live: 0 - Triggered: 0
<input type="checkbox"/>	Alarm5	Triggered	1	30/09/2019 12:56:17	Live: 0.44 - Triggered: 0.07

Filter :



**In History Alarms widget or in .csv file, live tag values are the values taken when the alarm's status change (for both types of placeholders)**

## Length limit of the Description field

Number of live tags that can be used inside each alarm's description depends on size of used tags. PB610 Panel Builder 600 will check and show a warning message when too many tags are used.



The sum of the bytes that are calculated using the underlying algorithm must be less than or equal to 50

$$4 \text{ bytes} + (\text{size of tag} + 2) + (\text{size of tag} + 2) + (\text{size of tag} + 2) + \dots \leq 50$$

Example:

Alarm Description:

Tag1=[TagInt], Tag2=[TagBool], Tag3=[TagStr8]

Fixed	4	
Tag1	6	4 (sizeof-INT) +2
Tag2	3	1 (sizeof-BOOL) +2
Tag3	10	8 (sizeof-STR8) +2
<b>Total:</b>	<b>23</b>	

When arrays are used, e.g. Tag1 as an array of 8 integer:

- **[Tag1]** or **[Tag1[-1]]**  
The entire array is shown and the number of the necessary bytes is calculated as  $4(\text{size-INT}) \times 8(\text{array elements}) + 2 = 34$  Byte
- **[Tag1[Index]]**  
An element of the array is shown and the number of the necessary bytes is calculated as  $4(\text{size-INT}) \times 1(\text{array elements}) + 2 = 6$  Byte. In this case, if at runtime the Index assumes the value -1 some values could be lost

## Exporting alarm buffers to .csv files

To export an event buffer containing an history alarms list, use the **DumpEventArchive** action.

See "[System actions](#)" on page 218 for details.



**Note:** Tag values displayed in the alarms description are also included in the buffer. Tags are sampled when the alarm is triggered and that value is logged and included in the description.

## Exporting alarm configuration

Path: **ProjectView > Config > double-click Alarms**

Name	Groups	Enable	Trigger	Tag	Description
Alarm1		<input checked="" type="checkbox"/>	bitMaskAlarm:0	MRTU1	Load alarm page
Alarm2		<input checked="" type="checkbox"/>	deviationAlarm:50.0 - 20.0	MRTU2	Increase alarm co
Alarm3		<input checked="" type="checkbox"/>	limitAlarm:10-100	Tag1	
Alarm4		<input checked="" type="checkbox"/>	valueAlarm:30	Tag2	
Alarm5		<input checked="" type="checkbox"/>	valueAlarm:@Tag4	Tag3	
Alarm6		<input checked="" type="checkbox"/>	bitMaskAlarm:0	Application/IOCONFIG_GLOBALS_MAPPING/IN0	
Alarm7		<input checked="" type="checkbox"/>	bitMaskAlarm:0	Application/IOCONFIG_GLOBALS_MAPPING/IN1	
Alarm8		<input checked="" type="checkbox"/>	deviationAlarm:50.0 - 20.0	Application/PLC_PRG/supercar	

Click the **Export Alarms** button: the alarms configuration table is exported into an .xml file.

You can edit the resulting .xml file using third part tools (for example, Microsoft Excel).

eventBuffer	logToEventArchive	eventType	subType	storeAlarmInfo	name	source
n/a	TRUE	0	0	FALSE	n/a	n/a
AlarmBuffer1	TRUE	14	1	TRUE	Alarm1	MRTU1
AlarmBuffer1	TRUE	14	1	TRUE	Alarm2	MRTU2
AlarmBuffer1	TRUE	14	1	TRUE	Alarm3	Tag1
AlarmBuffer1	TRUE	14	1	TRUE	Alarm4	Tag2
AlarmBuffer1	TRUE	14	1	TRUE	Alarm5	Tag3
AlarmBuffer1	TRUE	14	1	TRUE	Alarm6	Application/IOCONFIG_GLC
AlarmBuffer1	TRUE	14	1	TRUE	Alarm7	Application/IOCONFIG_GLC
AlarmBuffer1	TRUE	14	1	TRUE	Alarm8	Application/PLC_PRG/supe



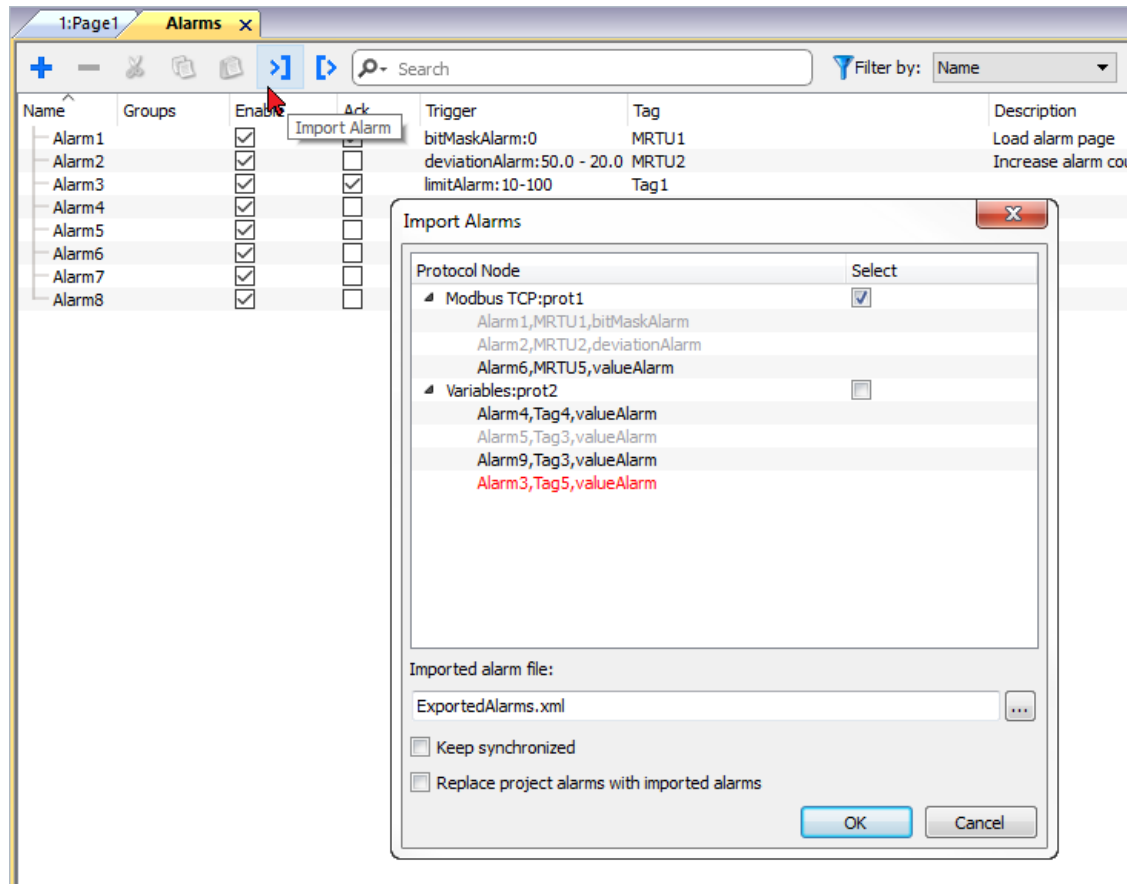
**Warning:** the bitMask values are reported as 2<sup>BitPosition</sup> in Hexadecimal format.

alarmType	lowLimit	highLimit	value	bitMask	deviation	setPoint	enableTag
n/a	0	1000	0	1	50	20	n/a
bitMaskAlarm				1			
bitMaskAlarm				2			
bitMaskAlarm				4			
bitMaskAlarm				8			
bitMaskAlarm				10			
bitMaskAlarm				20			
bitMaskAlarm				40			
bitMaskAlarm				80			
bitMaskAlarm				100			
bitMaskAlarm				200			
bitMaskAlarm				400			
bitMaskAlarm				800			
bitMaskAlarm				1000			
bitMaskAlarm				2000			



## Importing alarm configuration

Path: **ProjectView** > **Config** > double-click **Alarms**



1. Click the **Import Alarms** button and select the .xml file from which to import the alarms configuration: the **Import Alarms** dialog is displayed.
2. Select the group of alarms to import and click **OK** to confirm.

Differences are highlighted in the **Import Alarms** dialog using different colors

Color	Description
Black	This is a new alarm and it will be imported
Red	This alarm has not been found and will be removed (only if check "Replace project alarms with imported alarms" is checked)
Blue	This alarm has been modified and will be updated.
Gray	This alarm is already part of the project and will be skipped.

## Automatic synchronization

Select the **Keep synchronized** option in the **Import Alarms** dialog to enable the automatic synchronization of the alarm configuration file.

Whenever changes occur in the alarms configuration, the file will be automatically updated in silent mode.

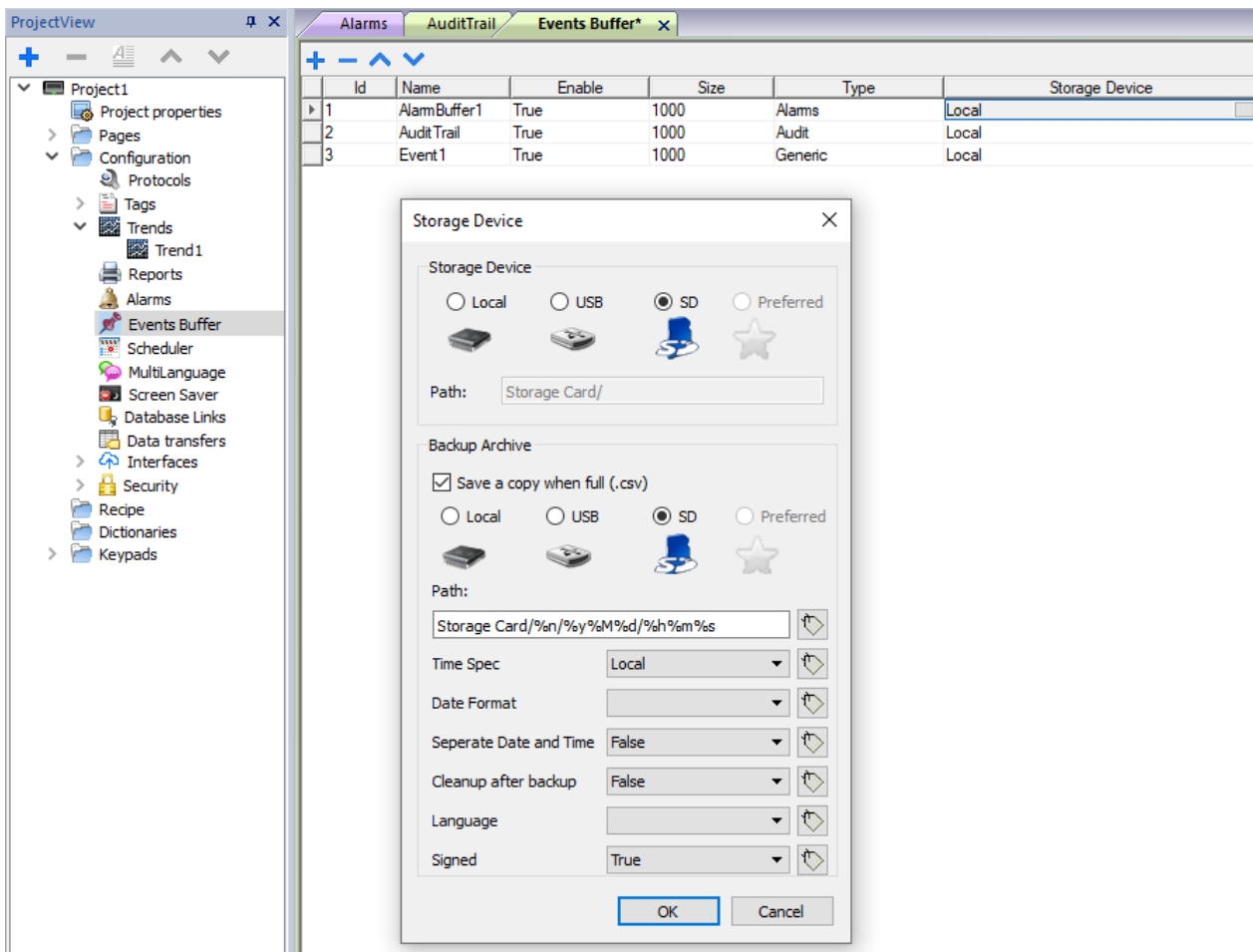


Tip: Enable this function when the alarm file is managed by a different tool (for example, PLC programming software) as well as by PB610 Panel Builder 600.



# 20 Events Buffer

The "Events Buffer" page gives you the possibility to configure the current events buffers (used for store alarms or audit trail information) or add additional events buffers.



Parameter	Description
<b>Id</b>	Buffer identification number
<b>Name</b>	Buffer name
<b>Enable</b>	Enable/disable logging
<b>Size</b>	Size of log file. Data is automatically saved to disk every 5 minutes.

Parameter	Description
<b>Type</b>	Type of events logged: <ul style="list-style-type: none"> <li>• Alarms</li> <li>• Audit</li> <li>• Generic</li> </ul>
<b>Storage Device</b>	Device where the data will be stored

## Backup Archive

If **Save a copy when full** option is enabled, the HMI device will save a copy when the events buffer is full before it is overwritten by newer data.

Parameter	Description
<b>Path</b>	Where events buffer data will be copied. The below wild cards are supported <ul style="list-style-type: none"> <li>• %n = Events buffer name</li> <li>• %y = Year</li> <li>• %M = Month</li> <li>• %d = Day</li> <li>• %h = Hour</li> <li>• %m = Minutes</li> <li>• %s = Seconds</li> </ul>
<b>Time Spec</b>	Timestamp of events <ul style="list-style-type: none"> <li>• Local Use the time of the HMI device where the project is running</li> <li>• Global Use global time (GMT)</li> </ul>
<b>Date Format</b>	Time and Date format. Placeholders can be used (see <a href="#">"Time and Date placeholders" on page 472</a> )
<b>Separate Date and Time</b>	When "true", the date and the time are placed into two different fields
<b>Cleanup after backup</b>	When "true", the event buffer is clean up after completing the backup. When "false", the older events are removed when new events are incoming (circular buffer)
<b>Language</b>	Language to use
<b>Signed</b>	When "true", the additional file with the signature is added (see <a href="#">"Signed CSV files" on page 358</a> )

# 21 Recipes

---

Recipes are collections of tag values organized in sets that satisfy specific application requirements.

For example, if you have to control room variables (temperature and humidity) in the morning, afternoon and evening. You will create three sets (morning, afternoon and evening) in which you will set the proper tag values.

Each element of the recipe is associated to a tag and can be indexed into sets for a more effective use. This feature allows you to extend the capabilities of controllers that have limited memory.

You can add controller data to a page using a recipe widget. Recipe data contains all the controller data items; however data is no longer read directly from the controller but rather from the associated recipe element in the HMI device.

Recipe data is configured in PB610 Panel Builder 600 workspace; the user can specify default values for each element of the data records. In HMI Runtime, data can be edited and saved to a new data file, any change to recipe data is therefore stored to disk. With the use of a separate data file HMI Runtime ensures that modified recipe values are retained throughout different project updates. In other words, a subsequent project update does not influence the recipe data modified by the user in the HMI Runtime.

See "[Recipe actions](#)" on page 213 for details on how to reset recipe data.



Note: Recipe data can be stored on a Flash memory, on a USB drive or on a SD card.

---

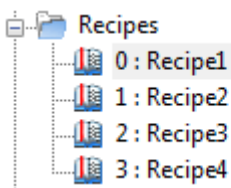
<b>Managing recipes</b> .....	<b>283</b>
<b>Configuring a recipe widget</b> .....	<b>286</b>
<b>Recipe status</b> .....	<b>287</b>
<b>Uploading/downloading a recipe</b> .....	<b>288</b>
<b>Backup and restore recipes data</b> .....	<b>289</b>

## Managing recipes

### Creating a recipe

To create a recipe for your project:

1. In **ProjectView** right-click **Recipes** and select **Insert Recipe**: an empty recipe is added. You create and configure recipes using the Recipe Editor.



## Recipe editor

Path: **ProjectView** > **Recipes** > double-click **RecipeName**

index	Element Name	Tag	Fill Tank 1	Fill Tank 3	Fill Tank 5	Fill Tank 7	Fill Tank 1	Empty Tank	Empty Tank	Empty Tank 75_	Em
0	Home Valve	Recipe_HomeV: 1	1	1	1	1	0	0	0	0	0
1	Truck Valve	Recipe_TruckV: 0	0	0	0	0	1	1	1	1	1
2	Fill Flow Meter	Recipe_FillFlow: 15	35	50	75	100	75	50	25		15
3	Empty Flow Meter	Recipe_EmptyFl: 0	0	0	0	0	25	50	75		85
4	Chemical1	Recipe_Chemic: 0	0	0	0	0	0	0	0		0
5	Chemical2	Recipe_Chemic: 0	0	0	0	0	0	0	0		0

## Configuring recipe properties

In the **Properties** pane of each recipe you set the following parameters:

Parameter	Description
<b>Recipe Name</b>	Name of the recipe
<b>Number of sets</b>	Number of values sets for each recipe element. Each set has a different configurable name.

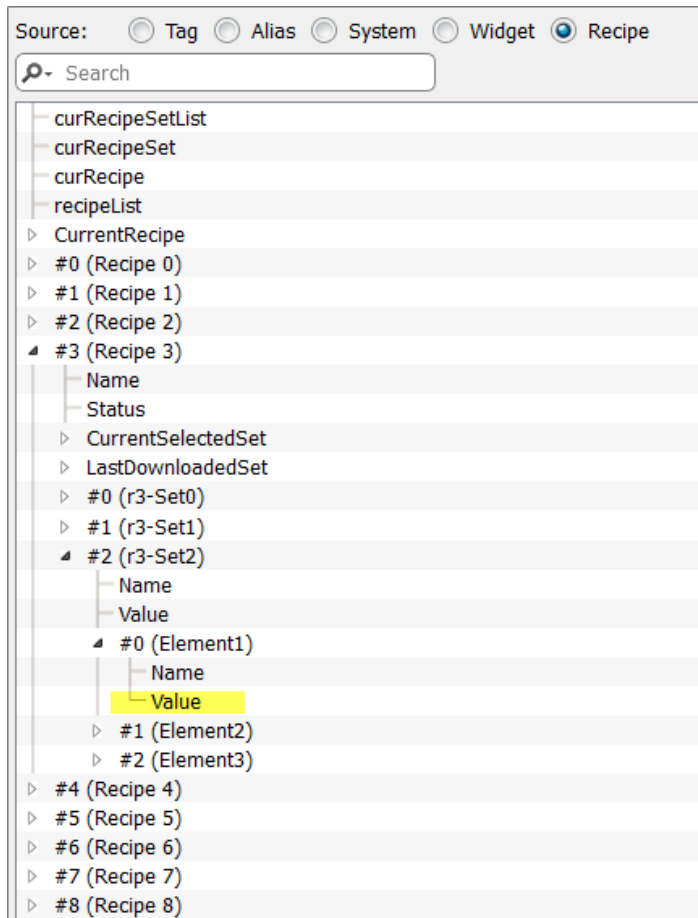
Properties	
Recipe : _RecipeMgr	
Recipe Name	Recipe1
Number of sets	10
Set 0	Fill Tank 15_
Set 1	Fill Tank 35_
Set 2	Fill Tank 50_
Set 3	Fill Tank 75_
Set 4	Fill Tank 100_
Set 5	Empty Tank 25_
Set 6	Empty Tank 50_
Set 7	Empty Tank 75_
Set 8	Empty Tank 90_
Set 9	Empty Tank 100_

## Setting up a recipe

1. Click **+** to add an element of the recipe.
2. Link the tags to each recipe element.

## Defining recipe fields

Create a recipe field in the page using a numeric widget and attaching it to a recipe item after selecting Recipe as the Source.



In the **Attach to** dialog you have the choice of all the different recipe variables, such as:


- Current Recipe >Current Selected Recipe Set> Element > Value
- Selected Recipe > Selected Set0 > Element > Value
- recipeList

When numeric widgets are defined as read/write, the default recipe data can be edited at runtime. These new values are stored in a separate file as modified recipe data.

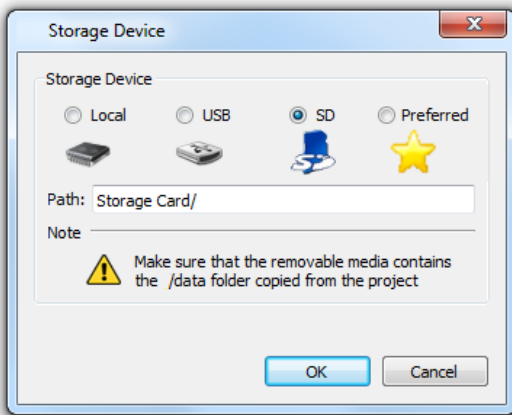


Note: Since JavaScript API functions are used, the recipe elements and sets can be referenced by name or by position. To avoid ambiguity between names and index, the names of the recipe elements and sets must include at least one alphanumeric character.

## Storing recipe data

In the Recipe Editor click the storage type icon  to select where to store recipe data: the **Storage Device** dialog is displayed.





For USB drive and SD card storage you can provide the folder location.



**WARNING:** Recipe configuration files are created automatically when the project is saved and stored in the data subfolder of the project. To use external storage devices, you need to copy this folder into the external device. Note that you have the responsibility to manage the data folder inside external devices. Even dynamic files are not deleted when project is updated using the “Delete dynamic file” option.



**Important:** You can add a subfolder but you must not rename the "data" subfolder.

## Import/Export recipes

To import/export the recipes configuration of your project:

In **ProjectView** right-click **Recipes** and select **Export Recipe** or **Import Recipe**

The following formats are supported for import:

- Comma Separated Values (.csv)
- Unicode Text (.txt)

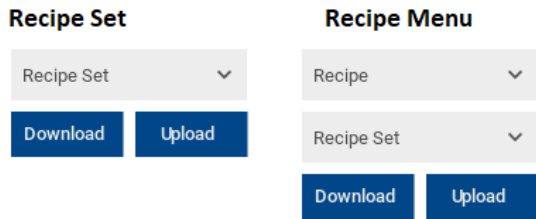


Note: Use the Unicode Text file format when you import a file modified using Microsoft® Excel®.

## Configuring a recipe widget

You can choose one of the two recipe widgets available in the **Widget Gallery**:

- **Recipe set:** allows you to select a recipe set for upload or download. See ["Uploading/downloading a recipe" on page 288](#)
- **Recipe menu:** when more recipes have been created for a project, use this widget to manage all recipes and select the desired sets for each of them.



## Configuring the Recipe Set widget

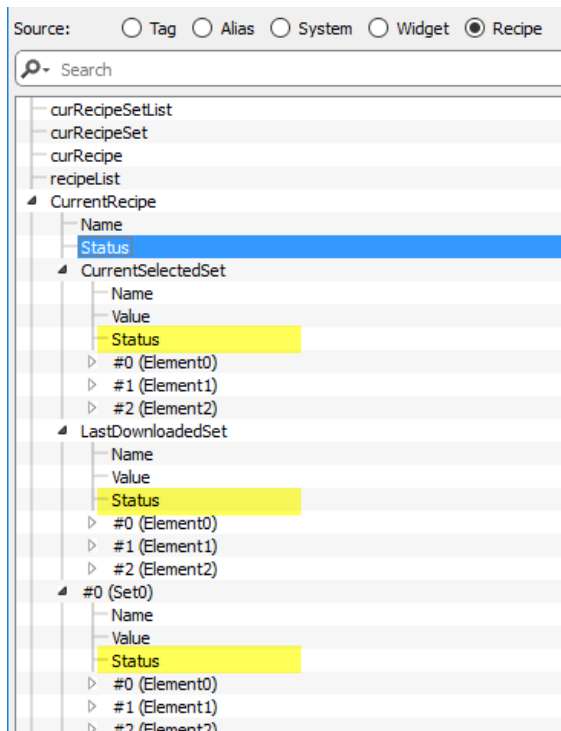
In the **Properties** pane of each **Recipe Set** widget set the following parameter:

Parameter	Description
Recipe Name	Name of the recipe

## Recipe status

Each recipe contains two kinds of status parameters

- Recipe Status (blue in the below picture)  
Give information regarding the last download or upload operation
- DataSet Status (yellow in the below picture)  
Give information of modified datasets



## Recipe Status

After every recipe upload or download, or recipe set modification, the **Recipe Status** parameters contain a value with the result of the operation.

Code	Function	Description
0	Set modified	Selected set changed
1	Download triggered	Download request triggered
2	Download Done	Download action completed
3	Download Error	Error during download (for example, unknown set, unknown recipe, controller not ready, Tags write failed etc.)
4	Upload triggered	Upload request triggered
5	Upload done	Upload action completed
6	Upload Error	Error during upload - same as for download
7	General Error	General error (for example, data not available)

## DataSet Status

The status of each data set indicates that it has been changed. This information may be useful to not forget to download the recipe to synchronize the PLC. Both download or upload operations will reset the **DataSet Status** to 0.

Code	Function	Description
0	Synchronized	User synchronized PLC with the dataset values
1	Modified	User modified some values of the dataset



Note: After a device startup or a recipe reset/restore, all status values will reset to 0.

# Uploading/downloading a recipe

## Uploading a recipe

You upload a recipe to an HMI device using a recipe widget and the **UpLoadRecipe**, **UpLoadCurRecipe** action in one of the following ways:

- attach the action to an event of a button or a switch (see [""Attach to" parameters" on page 49](#) for details)
- configure the action in an alarm action list (see ["Alarm actions" on page 194](#) for details)
- configure the action in a scheduler action list (see ["Scheduling events at runtime" on page 346](#) for details)

## Downloading a recipe

You download a recipe from an HMI device using a recipe widget and the **DownloadRecipe**, **DownLoadCurRecipe** action. See ["Recipe actions" on page 213](#)

## Backup and restore recipes data

The recipe data stored in an HMI device can be exported for backup and later restored. This is done using the **DumpRecipeData** or the **RestoreRecipeData** actions.

See ["Recipe actions" on page 213](#) for details.



# 22 Trends

---

Trends allow you to sample and record the values of specified tags according to specific sampling conditions. The trend function includes trend acquisition and trend display.

Trend acquisition parameters are set in the Trend editor so that data can be stored. Stored data can then be displayed in a graphical format using a trend widget.

---

<b>Data logging</b> .....	<b>292</b>
<b>Exporting trend buffer data</b> .....	<b>297</b>
<b>Realtime trend widget</b> .....	<b>299</b>
<b>History trend widget</b> .....	<b>300</b>
<b>Scatter diagram widget</b> .....	<b>303</b>
<b>Scatter chart widget</b> .....	<b>304</b>
<b>Trend widget tips</b> .....	<b>309</b>
<b>Table trend widget</b> .....	<b>315</b>

# Data logging

Data can be logged and stored to HMI memory. Data logging allows you to store the values of a group of tags all at the same time to a buffer. Data logging can be triggered by a timer or by a dedicated tag. Logged data can be exported to a .csv file or displayed using the historical trend widget. Logged data can be saved locally on a USB device or SD card, or on any available custom network folder.



**WARNING: The operation with removable memory devices (USB Flash drives, SD memory cards) containing a very large number of files may result in a decrease of system performance.**



**WARNING: The max number of files inside a SD memory card depends on the type of formatting (e.g. FAT32 max 65535 files; FAT max 512 files).**



**WARNING: Flash cards support a limited number of write operations. We suggest to use only good quality memory cards; in the case your application use intensively the memory card consider a regular substitution of the memory card.**



**WARNING: If the data/time is moved back, the samples with invalid date/time are removed from the trend buffer. When system detects that data/time is invalid (e.g. battery low), a popup is shown to advise the user and the date/time of the last sample is used to avoid losing data.**









Storage is based on trend buffers. Trend buffers are organized as a FIFO queue: when the buffer is full, the oldest values are discarded unless you configure your trend to create a backup copy of the buffer.


## Adding a trend buffer

Path: **ProjectView** > **Config** > double-click **Trends**




1. Click **Add** to add a new buffer.
2. Click **+** next to each trend buffer to display all configuration parameters.







Name	Title	Tag	Format	Comment
1 Name1	Temperature [UnitOfMeasure]	Tag1	Custom	
2 Name2	PLC Value	Tag2	Numeric	
3 Name3	String	Tag3		

Toolbar Element	Description
	Add a trend that will be sampled from the HMI device.
	Add a trend that will be managed and sampled from the external device, instead of from the HMI device. You need a device that supports this feature to use it (e.g. BACnet devices). The parameters depend on the used device, refer to the manual of the selected device.
	Remove the select trend.
	Offers the ability to customize the labels that appear in the trend dump header and trend table widgets. <ul style="list-style-type: none"> <li>• Timestamp</li> <li>• Date</li> <li>• Time</li> <li>• Quality</li> </ul> <p>The below placeholders can be used:</p> <ul style="list-style-type: none"> <li>• " \n " (space + \n + space) can be used to split the label into two or more lines</li> <li>• <b>[TagName]</b> (tag name enclosed in square brackets) can be used to display a tag value</li> </ul>
	Copy the selected trend
	Paste the selected trend
	Export selected trends to file
	Import trends from file
<b>Total memory Space</b>	Memory used by the defined trend buffers.

Trend Header	Description
<b>Trend Name</b>	Name of the trend collection (set of tags sampled at the same time)
<b>Active</b>	When enabled, the trend runs by default at system startup. <p> Note: Trends cannot be activated at runtime.</p>
<b>Source</b>	List of the tags sampled by the trend.
<b>Import from file</b>	Import and overwrite trend parameters from a file (exported previously).




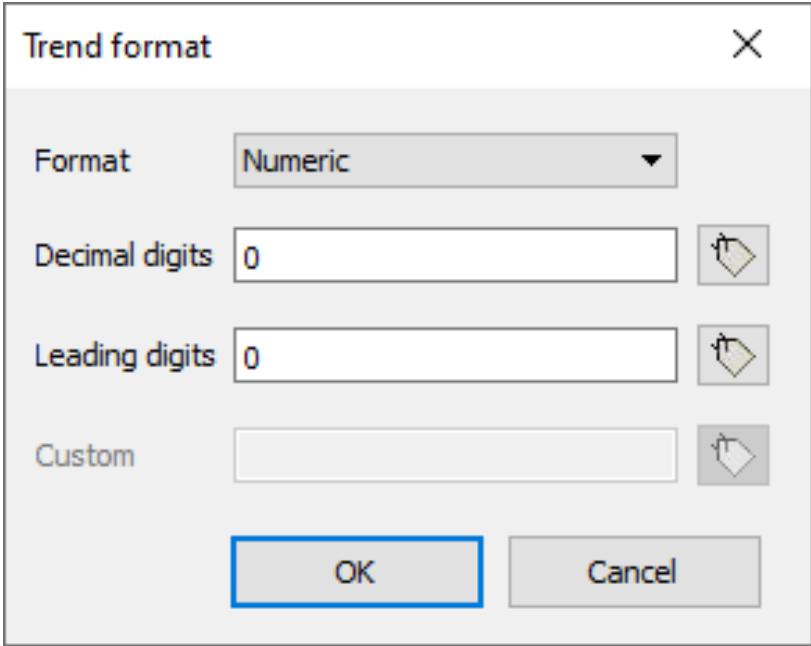
Trend Element	Description						
<b>Number of Samples</b>	Trend buffer size (see <a href="#">"Number of Samples" on page 297</a> for additional information)						
<b>Sampling Time</b>	Sampling interval.  Note that instead of a constant, you can use a Tag to define/change the sample time at runtime. When sample time is 0, or negative, sampling is suspended.						
<b>Time</b>	Time unit for the sample time. Could be 1 second (default) or 1/10 seconds  Be aware that increasing the sampling rate could impact global HMI device performances.						
<b>Timestamp</b>	When checked, samples are stored using the timestamp provided from the remote device. Available only: <ul style="list-style-type: none"> <li>• for device's protocols that support this feature (OPC UA Client and BACnet)</li> <li>• when trend buffer is configured to with a single tag</li> </ul>						
<b>Trigger</b>	Tag triggering the sample. If used, when the value of this tag changes, a sample is collected.  Note: Trigger and Source can refer to the same tag.						
<b>Storage Device</b>	Where trend buffer data will be stored.						
<b>Backup Archive</b>	If <b>Save a copy when full</b> option is enabled, a backup copy of the buffer data is created before it is overwritten by newer data. <table border="1" data-bbox="446 1332 1471 1881"> <tbody> <tr> <td data-bbox="446 1332 622 1388"><b>.csv</b></td> <td data-bbox="622 1332 1471 1388">Backup data using text CSV format.</td> </tr> <tr> <td data-bbox="446 1388 622 1825"><b>Path</b></td> <td data-bbox="622 1388 1471 1825">           Where trend buffer data will be copied.            The below wild cards are supported           <ul style="list-style-type: none"> <li>• %n = Trend name</li> <li>• %y = Year</li> <li>• %M = Month</li> <li>• %d = Day</li> <li>• %h = Hour</li> <li>• %m = Minutes</li> <li>• %s = Seconds</li> </ul> </td> </tr> <tr> <td data-bbox="446 1825 622 1881"><b>Select</b></td> <td data-bbox="622 1825 1471 1881">Fields that will be inside the dump file</td> </tr> </tbody> </table>	<b>.csv</b>	Backup data using text CSV format.	<b>Path</b>	Where trend buffer data will be copied. The below wild cards are supported <ul style="list-style-type: none"> <li>• %n = Trend name</li> <li>• %y = Year</li> <li>• %M = Month</li> <li>• %d = Day</li> <li>• %h = Hour</li> <li>• %m = Minutes</li> <li>• %s = Seconds</li> </ul>	<b>Select</b>	Fields that will be inside the dump file
<b>.csv</b>	Backup data using text CSV format.						
<b>Path</b>	Where trend buffer data will be copied. The below wild cards are supported <ul style="list-style-type: none"> <li>• %n = Trend name</li> <li>• %y = Year</li> <li>• %M = Month</li> <li>• %d = Day</li> <li>• %h = Hour</li> <li>• %m = Minutes</li> <li>• %s = Seconds</li> </ul>						
<b>Select</b>	Fields that will be inside the dump file						

Trend Element	Description										
	<table border="1"> <tr> <td style="background-color: #cccccc;"><b>Fields</b></td> <td>  Note that you can use a string tag to define the fields to dump at runtime.                 </td> </tr> <tr> <td style="background-color: #cccccc;"><b>Select Curves</b></td> <td>                     Curves that will be inside the dump file   Note that you can use a string tag to define the fields to dump at runtime.                 </td> </tr> <tr> <td style="background-color: #cccccc;"><b>Time Spec</b></td> <td>                     Timestamp of samples                     <ul style="list-style-type: none"> <li>• Local Use the time of the HMI device where the project is running</li> <li>• Global Use global time (GMT)</li> </ul> </td> </tr> <tr> <td style="background-color: #cccccc;"><b>Data Format</b></td> <td>Time and Date format. Placeholders can be used (see <a href="#">"Time and Date placeholders" on page 472</a>)</td> </tr> <tr> <td style="background-color: #cccccc;"><b>Language</b></td> <td>Language to use</td> </tr> </table>	<b>Fields</b>	 Note that you can use a string tag to define the fields to dump at runtime.	<b>Select Curves</b>	Curves that will be inside the dump file  Note that you can use a string tag to define the fields to dump at runtime.	<b>Time Spec</b>	Timestamp of samples <ul style="list-style-type: none"> <li>• Local Use the time of the HMI device where the project is running</li> <li>• Global Use global time (GMT)</li> </ul>	<b>Data Format</b>	Time and Date format. Placeholders can be used (see <a href="#">"Time and Date placeholders" on page 472</a> )	<b>Language</b>	Language to use
<b>Fields</b>	 Note that you can use a string tag to define the fields to dump at runtime.										
<b>Select Curves</b>	Curves that will be inside the dump file  Note that you can use a string tag to define the fields to dump at runtime.										
<b>Time Spec</b>	Timestamp of samples <ul style="list-style-type: none"> <li>• Local Use the time of the HMI device where the project is running</li> <li>• Global Use global time (GMT)</li> </ul>										
<b>Data Format</b>	Time and Date format. Placeholders can be used (see <a href="#">"Time and Date placeholders" on page 472</a> )										
<b>Language</b>	Language to use										
<b>Sampling Filter / Trigger Filter</b>	<p>This parameter allows to specify a dynamic filter if required.</p> <p>When sampling is done on time basis the offset is applied to the sampled Tag value. If the new value exceeds the specified limits the new value is considered valid and stored, otherwise the new record will retain the previous saved value.</p> <p>When sampling is done on trigger the offset is applied to the trigger Tag value. If the trigger Tag value change exceeds the specified limits a new sample is taken and stored, otherwise no sampling will be done.</p>										

Use the buttons on the toolbar to add, remove or move tags to be sampled



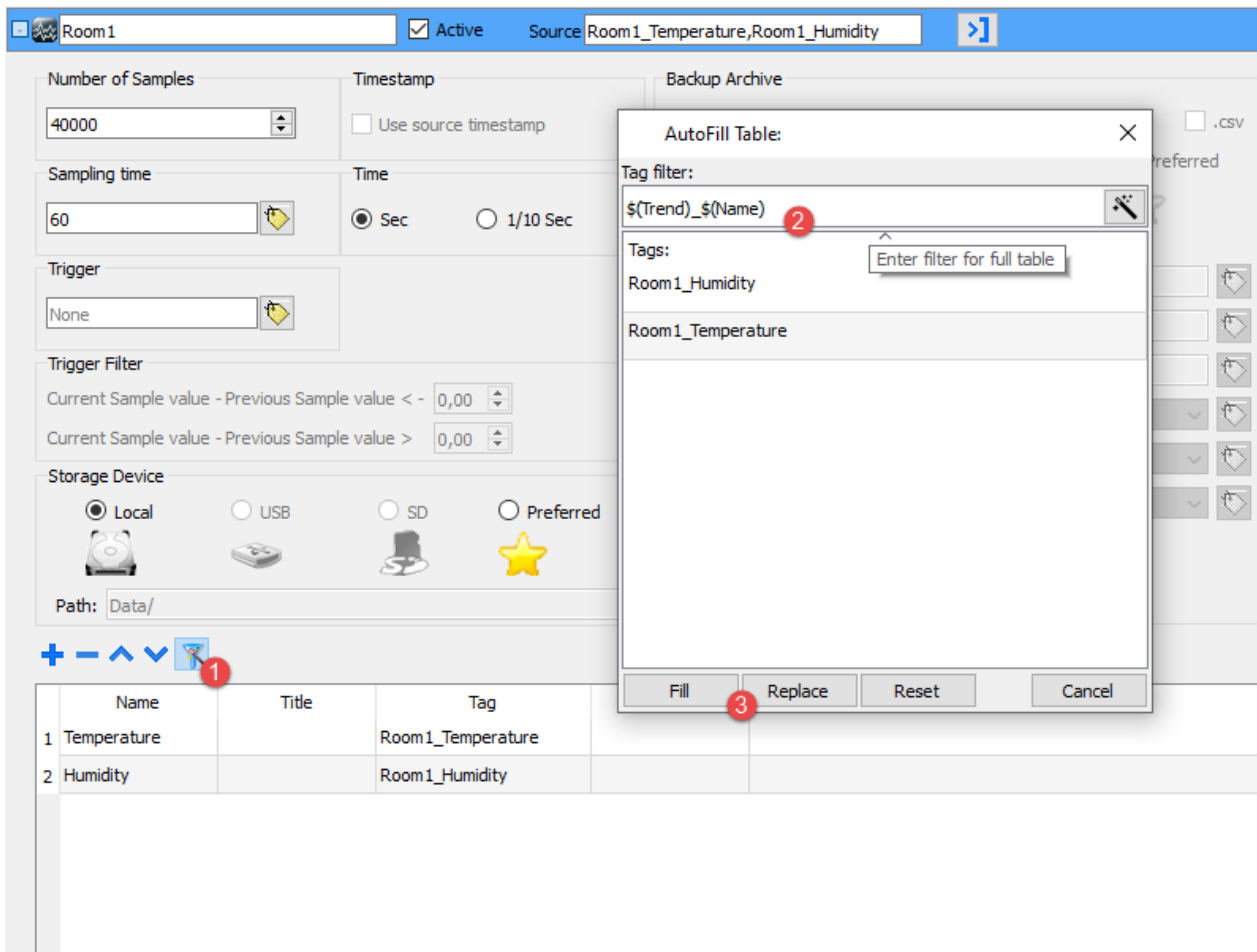
Samples	Description
<b>Name</b>	Trend name
<b>Title</b>	<p>Title that has to appears inside the trend table or the trend dump.</p> <ul style="list-style-type: none"> <li>• The placeholder " \n " (space + \n + space) can be used to split the label into two or more lines</li> <li>• The placeholder <b>[TagName]</b> (tag name enclosed in square brackets) can be used to display a tag value</li> </ul> <p>Example:</p> <p>"Temperature ([UnitDiMisura])" will be shown as "Temperature (°C)" when the tag UnitDiMisura = "°C"</p>

Samples	Description
<b>Tag</b>	Tag that must be sampled  <b>Tags string are supported until 8 bytes. If tag size is greater than 8 bytes, only the first 8 bytes are stored in trend. Unicode chars are not supported.</b>
<b>Format</b>	Display format to use. Note that even the custom format can be used (see " <a href="#">Custom Formats</a> " on page 38). 
<b>Comment</b>	You can write whatever you want here

## Tags Wizard

Tag Wizard Button will allow you to automatically fill in the tag names in the trend table.

You can use the placeholders  $\$(Trend)$  and  $\$(Name)$  in the regular expression to search the available tags. When you press "Fill" or "Replace" buttons the matching tags will be added to the trend table



## Number of Samples

The number of samples that you can have is dependent on the memory size reserved for trend buffers and from the size of each sample.

$$\text{Number of available samples} = \text{Available Memory (bytes)} / \text{Size of sample (bytes)}$$

Where the size of each sample is dependent on how many tags are used and can be calculated using the below formula:

$$\text{Size of sample (bytes)} = \text{TAGS} * 9 + 11$$

You are free to use the entire available memory for a unique trend buffer or split the available memory over several trends.

See also:

- Trend limits on ["Table of functions and limits" on page 612](#)
- Reserved memory for trend buffer on ["HMI devices capabilities" on page 613](#)

## Exporting trend buffer data

Use the **DumpTrend** action to export trend buffer data to a .csv file. See ["DumpTrend" on page 218](#) for the macro parameters details.

The exported .csv file could have different formats defined from the Dump Trend macro parameters. The different formats are maintained mainly for compatibilities reasons.

#### FileFormat: **Compatibility CSV**

	A	B	C	D	E	F	G	H	I	J	K
1	Type	Value	Time Stamp	Refresh Time	Quality	Type	Value	Quality	Type	Value	Quality
2	4	0	2015-09-18T14:42:22.000Z	1000	192	8	0.00E+00	192	3	0	192
3	4	0	2015-09-18T14:42:23.000Z	1000	192	8	0.00E+00	192	3	0	192
4	4	0	2015-09-18T14:42:24.000Z	1000	192	8	0.00E+00	192	3	0	192
5	4	40	2015-09-18T14:42:25.000Z	1000	192	8	0.00E+00	192	3	0	192
6	4	40	2015-09-18T14:42:26.000Z	1000	192	8	0.00E+00	192	3	0	192
7	4	40	2015-09-18T14:42:27.000Z	1000	192	8	0.00E+00	192	3	0	192
8	4	40	2015-09-18T14:42:28.000Z	1000	192	8	5.00E+01	192	3	0	192
9	4	40	2015-09-18T14:42:29.000Z	1000	192	8	5.00E+01	192	3	0	192
10	4	40	2015-09-18T14:42:30.000Z	1000	192	8	5.00E+01	192	3	0	192

#### FileFormat: **Compact CSV**

	A	B	C	D	E	F	G
1	Timestamp	Tag1	4 Tag2		8 Tag3		3
2		Value	Quality	Value	Quality	Value	Quality
3	2015-09-18T14:42:22.000Z	0	192	0.00E+00	192	0	192
4	2015-09-18T14:42:23.000Z	0	192	0.00E+00	192	0	192
5	2015-09-18T14:42:24.000Z	0	192	0.00E+00	192	0	192
6	2015-09-18T14:42:25.000Z	40	192	0.00E+00	192	0	192
7	2015-09-18T14:42:26.000Z	40	192	0.00E+00	192	0	192
8	2015-09-18T14:42:27.000Z	40	192	0.00E+00	192	0	192
9	2015-09-18T14:42:28.000Z	40	192	5.00E+01	192	0	192
10	2015-09-18T14:42:29.000Z	40	192	5.00E+01	192	0	192



Note: The first row of the header contains the tags names and tags data types

#### FileFormat: **Compact CSV** with columns' selection

	A	B	C	D	E	F	G	H
1	Date	Time	Name1(int)	Quality	Name2(int)	Quality	Name3(boolean)	Quality
2	17/04/2018	07:24:29	0	192	10	192	0	192
3	17/04/2018	07:24:30	1	192	11	192	1	192
4	17/04/2018	07:24:31	2	192	12	192	0	192
5	17/04/2018	07:24:32	3	192	13	192	1	192
6	17/04/2018	07:24:33	4	192	14	192	0	192
7	17/04/2018	07:24:34	5	192	15	192	1	192
8	17/04/2018	07:24:35	6	192	16	192	0	192
9	17/04/2018	07:24:36	7	192	17	192	1	192
10	17/04/2018	07:24:37	8	192	18	192	0	192



The time required to dump a trend buffer depends on the number of samples present in the buffer, the memory type, and the HMI device type.

Example in the worst conditions

HMI Type	Buffer Size	Samples	Time
----------	-------------	---------	------

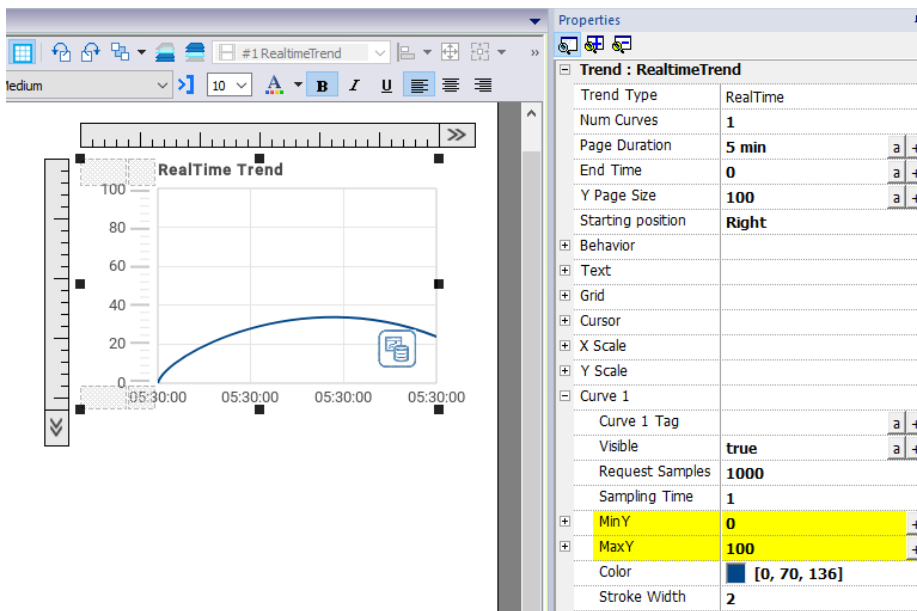
Win32	500 Mb	18.078.800 samples (2 tags)	25 Min
Linux	50 Mb	1.807.800 samples (2 tags)	4 Min
WinCE	25 Mb	903.900 samples (2 tags)	10 Min

## Realtime trend widget

The real-time trend widget can be used to display the changes of value of a tag. Data is not stored in a trend buffer and cannot be retrieved for later analysis.

### To display a real-time trend:

1. Drag and drop the **RealTime Trend** widget from the widget gallery to the page.



2. Attach the tag that you want to sample to the **Curve n Value**. Data is always plotted against time.

### RealTime trend widget properties

Property	Description
<b>Num Curves</b>	Number of trend curves to be displayed
<b>Page Duration</b>	Time window to show
<b>End Time</b>	End time of the time window  This parameter can be used to scroll the time window. When zero, the end time is the current system time.
<b>Starting Position</b>	Specifies where the curve begin to be drawn when the page is opened (can be left, center or right).
<b>Behavior</b>	Definition of:

Property	Description
	<ul style="list-style-type: none"> <li>• Min/Max of Y axis</li> <li>• Number of tickets to draw on the axes</li> <li>• Background image</li> </ul>
<b>Text</b>	Trend title and font properties (font size, label, etc.)
<b>Grid</b>	Properties of grid presentation (colors)
<b>Cursor</b>	Properties of cursor presentation (enable and color)
<b>X Scale</b>	Properties of X Scale presentation
<b>Y Scale</b>	Properties of Y Scale presentation
<b>Curve "n"</b>	<p>Tag that will be plotted in the trend widget.</p> <ul style="list-style-type: none"> <li>• <b>Select Tag</b> select the tag to display</li> <li>• <b>Attach To</b> select a tag (string) that will contain the name of the tag to display (it is an indexed tag selection)</li> </ul>



Tag values can be scaled using the X Forms in the **Attach to** dialog. See [""Attach to" parameters" on page 49](#) for details.

## History trend widget

The data collected and stored from the data logger can be analyzed using the History Trend widget.

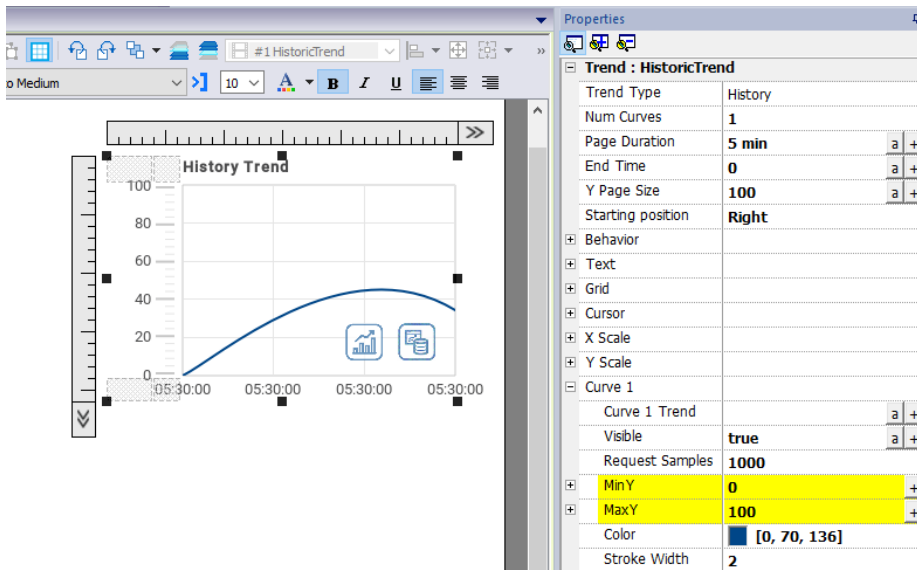
This is a two-step process:

- first you create a trend buffer to collect data for specified tags at specific points in time,
- then you configure a History Trend widget to display the collected data in a graphical format.

See ["Data logging" on page 292](#) for details on how to create a trend buffer

**To display a history trend:**

1. Drag and drop the **History Trend** widget from the widget gallery to the page.



2. Attach the trend buffer that you want to draw to the **Curve *n* Value**. Data is always plotted against time.

## History trend widget properties

Property	Description
<b>Num Curves</b>	Number of trend curves to be displayed
<b>Page Duration</b>	Time window to show
<b>End Time</b>	End time of the time window  This parameter can be used to scroll the time window. When zero, the end time is the current system time.
<b>Starting Position</b>	Specifies where the curve begin to be drawn when the page is opened (can be left, center or right).
<b>Behavior</b>	Definition of: <ul style="list-style-type: none"> <li>• Min/Max of Y axis</li> <li>• Number of tickets to draw on the axes</li> <li>• Background image</li> </ul>
<b>Text</b>	Trend title and font properties (font size, label, etc.)
<b>Grid</b>	Properties of grid presentation (colors)
<b>Cursor</b>	Properties of cursor presentation (enable and color)
<b>X Scale</b>	Properties of X Scale presentation



Property	Description
Y Scale	Properties of Y Scale presentation
Curve "n"	Buffer that contains the tag's values to plotted in the trend widget. <ul style="list-style-type: none"> <li>• Select Trend select the trend buffer to display</li> <li>• Attach To select a tag (string) that will contain the name of the trend buffer to display (it is an indexed trend buffer selection)</li> </ul>



Tag values can be scaled using the X Forms in the **Attach to** dialog. See [""Attach to" parameters" on page 49](#) for details.

## Printing historical trend widget

The historical trend widget can be found and used from the print report gallery.

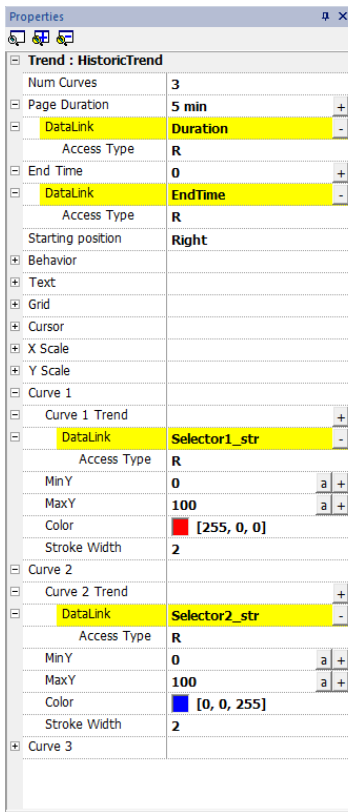
Using the "attach to tag" feature is possible to use tags to define some properties of the historical trend to print at runtime:

- Page Duration
- End Time
- Curve Name

"Page Duration" with "End Time" define the piece of the trend buffer to print. "Curve Name" can be used to select the curve to show. An empty string means no curve to show.



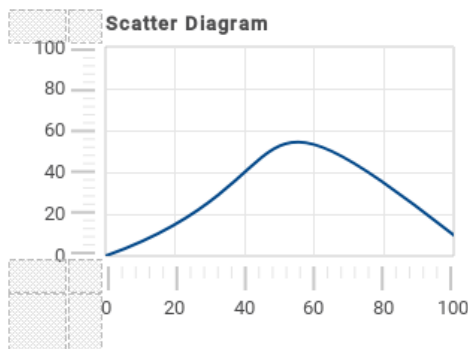
**SetTrendView() and ScrollTrendToTime() are display macros and have no effect on report printing.**



Name	Address
Duration	Duration int
EndTime	EndTime time
Selector1_str	Selector1_str string [20]
Selector2_str	Selector2_str string [20]
Selector3_str	Selector3_str string [20]

## Scatter diagram widget

A scatter diagram is a type of diagram to display values for two variables from a set of data using Cartesian coordinates. The data is displayed as a collection of points, each having the value of one variable determining the position on the horizontal axis and the value of the other variable determining the position on the vertical axis. For this reason it is often called *XY graph*.



Scatter diagram curves are obtained by a linear interpolation of points. To create a new scatter diagram:

1. Add a **Scatter Diagram** widget to the page.
2. Select the number of curves to show: each curve is named as Graph1, Graph2,...
3. Customize the general graph properties such as **X Min**, **X Max**, **Grid** details.
4. Define the max number of samples/values for each curve by setting the **Max Samples** parameter.

Here you set the max number of values to be displayed in the graph starting from first element in the array.

For example: Tag1[20] and Max Samples = 10 will show just first 10 elements of the Tag1 array.

- Define for each curve the two tags of type array to be displayed (**X-Tag** and **Y-Tag**).

When the array tags change, you can force a refresh with the **RefreshTrend** action .



Note: Scatter diagrams support only the **RefreshTrend** action.

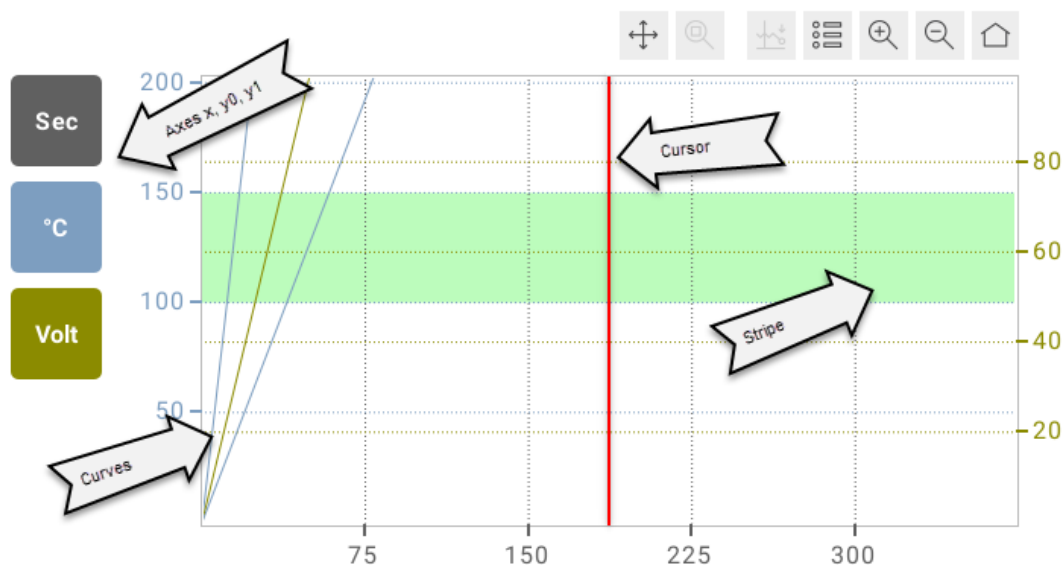
## Printing scatter diagram widget

The scatter diagram widget can be found and used from the print report gallery. Note that using the attach to tag feature is possible to use tags to define some properties of the scatter diagram to print at runtime.

## Scatter chart widget

Path: **Widget Gallery**> **Basic**> **Trends/Chart**> **Scatter Chart**

A scatter chart is a type of diagram to display values for two variables from a set of data using Cartesian coordinates. The data is displayed as a collection of points, each having the value of one variable determining the position on the horizontal axis and the value of the other variable determining the position on the vertical axis. For this reason it is often called *XY graph*.




## Parameters

Main Parameter	Description
<b>Axes Counter</b>	Number of vertical (Y) axes.
<b>Curves Counter</b>	Number of curves to draw

Main Parameter	Description
<b>Stripes Counter</b>	The number of active strips.
<b>Cursor</b>	Properties related to cursor enabling and its graphic style
<b>Style</b>	Properties that allow you to customize the graphic style of the widget
<b>Scatter Update Pause</b>	Curves update frequencies expressed in milliseconds.
<b>Auto Scaling</b>	Properties to configure the automatic scaling feature.
<b>Font</b>	Properties of the characters used in the graphic
<b>X Axis</b> <b>Y Axis 0</b> <b>Y Axis 1</b> ...	Properties to configure the style of the axes
<b>Curve 0</b> <b>Curve 1</b> <b>Curve 2</b> ...	Properties to configure the curves to be drawn
<b>Stripe 0</b> <b>Stripe 1</b> <b>Stripe 2</b> ...	Properties to add and configure the stripes on the charts
Cursor Parameter	Description
<b>Enable</b>	Enable/Disable the visualization of the cursor
<b>Label Enable</b>	Enable the visualization of labels with the values of the curves intersected by the cursor
<b>Position</b>	Position of the cursor in the view. Can be between 0 and 1 (the default value is 0.5)
<b>Color</b>	The colors of the cursor
<b>Labels Color</b>	The colors of the labels with the values of the curves intersected by the cursor.
<b>Labels Width</b> <b>Labels Height</b>	Size of the labels with the value of the curves. The font size will be organized accordingly.
Style Parameter	Description
<b>Horizontal Grid Lines</b>	Number of suggested horizontal grid lines. Chart widget tries to respect these number, but giving priority to the values shown on Y Axis which should be beautiful to read. This means integer values when it is possible or at worst with a single decimal digit.
<b>Vertical Grid Lines</b>	Number of suggested vertical grid lines. Chart widget tries to respect these number, but giving priority to the values shown on X Axis which should be beautiful to read. This means integer values when it is possible or at worst with a single decimal digit.

Style Parameter	Description
<b>Background</b>	Background color of the widget
<b>Foreground</b>	Color of the text where is not explicitly defined (e.g. the unit of measure shown inside axes controls)
<b>Axes Control Background Color</b>	Background color of the built-in axes control
<b>Axes Control Shadow Color</b>	Shadow color of the built-in axes control
<b>Chart Background Color</b>	Background color of the chart area
<b>Toolbar Buttons Size</b>	Width and Height of the built-in toolbar buttons
<b>Grid Values Width</b>	Width of vertical grids values (Y axes)
<b>Grid Values Height</b>	Height of horizontal grid values (X axis)
<b>Width of the axes buttons</b>	Width of the axes buttons
<b>Axes Button Height</b>	Maximum height of the axes buttons
<b>Toolbar Color</b>	Color used for toolbar icons and related items (like zoom rectangle)
<b>Toolbar Enable</b>	Show/Hide the built-in toolbar (on the top)
<b>Axes Control Enable</b>	Show/Hide the built-in axes control (on the left)
Auto Scaling Parameter	Description
<b>Auto Scaling Enable</b>	Enable or disable the auto-scale feature. Auto-scale ensures that the X axis maximum always take into account the most recent values of the curves.
<b>Toolbar Button Visible</b>	Show or hide the auto-scale button on built-in toolbar.
Font Parameter	Description
<b>Font</b>	Font of the chart's strings
<b>Font Maximum Pixel Size</b>	Maximum pixel size. The chart adapts the font size based on available space
<b>Font Style</b>	Font style
<b>Font Bold</b>	Set the font bold

Axis Parameter	Description
Minimum	Axis minimum value
Maximum	Axis maximum value
Color	Axis color
Unit of measure	It is a label associated with the axis/curve
Show Unit Of Measure On Axis	Show unit of measure in the built-in axes control
Show Unit Of Measure On Grid	Show unit of measure in the grid values
Show Unit Of Measure On Cursor	Show unit of measure in the cursor labels (not available for X axis)

Curve Parameter	Description
Scatter Curve Type	<p>Defines how the curve is updated :</p> <ul style="list-style-type: none"> <li>• DYNAMIC curve is updated continuously</li> <li>• STATIC curve is updated only one time or with a specific action (see "<a href="#">ChartCommand</a>" on <a href="#">page 243</a>).</li> </ul>
Scatter Tag X	X array tag to use for the X axis
Scatter Tag Y	<p>Y array tag to use for the Y axis.</p> <p> The Y array size must be smaller than (or equal to) the X array size</p>
Axis	Select the axis associated with the curve
Custom Color Enable	Enable or disable the curve custom color selection. If disabled the color used for the curve is the same as the associated axis.
Style	<p>Curve style, can be:</p> <ul style="list-style-type: none"> <li>• SOLID</li> <li>• DASH</li> <li>• DOT</li> <li>• DASH_DOT</li> <li>• DASH_DOT_DOT</li> </ul>
Line Width	Curve width in pixels. The maximum width is 5 pixels
Visible	Show/Hide the curve

Stripe Parameter	Description
<b>Axis</b>	Associated axis: <ul style="list-style-type: none"> <li>• AXIS_X = Vertical stripe</li> <li>• AXIS_Yn = Horizontal stripe</li> </ul>
<b>Custom Color Enable</b>	Enable or disable the stripe custom color selection. If disabled the color used for the stripe is the same as the associated axis.
<b>Start Value</b>	Stripe start value
<b>End Value</b>	Stripe end value

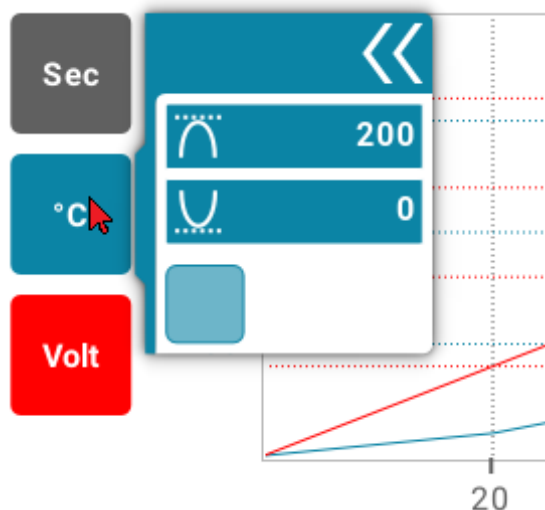
## Built-in axes control

The Built-in axes control can be shown/hide from the "Axes Control Enable" property available inside the "Style Parameter" section.

The built-in axes control has a button for each defined axes. The first one on the top is related to X axis and the others with the defined Y axes.








Each button is managing the OnMouseClicked and the OnMOuseHold events:

- OnMouseClicked  
Is a toggle that show/hide the associate curve (only Y axes)
- OnMOuseHold  
If the button is held down for a couple of seconds, a small panel will be displayed where it will be possible to change the min/Max values of the axis.



## Built-in toolbar

The Built-in toolbar can be shown/hide from the "Toolbar Enable" property available inside the "Style Parameter" section.

Toolbar Element	Description
	Activate the moving of the graphic through gesture commands
	Activate the zoom mode through gesture commands
	Activate the moving of the cursor through gesture commands
	Opens the X axis control pane
	Zoom In the graphic
	Zoom Out the graphic
	Reset the graphic view

## ChartCommand action

The same commands available from the Built-in toolbar are available through the actions (see ["ChartCommand" on page 243](#))

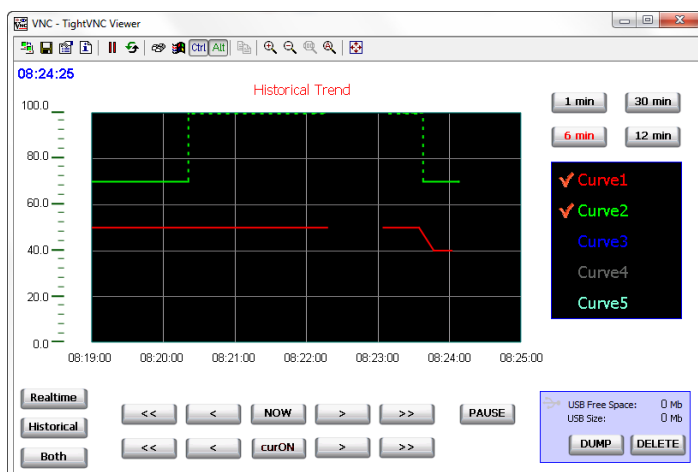
## Trend widget tips



Be aware that some properties are only available when the Properties pane is in Advanced view.

### Values outside range or invalid

When trend value goes beyond the limits set for the trend widget, a dotted line is displayed. When the value of the tag is not available, for example the controller device is offline, no curve is drawn.

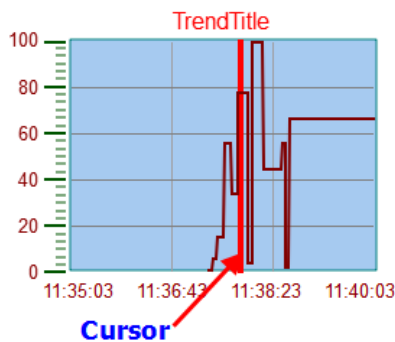




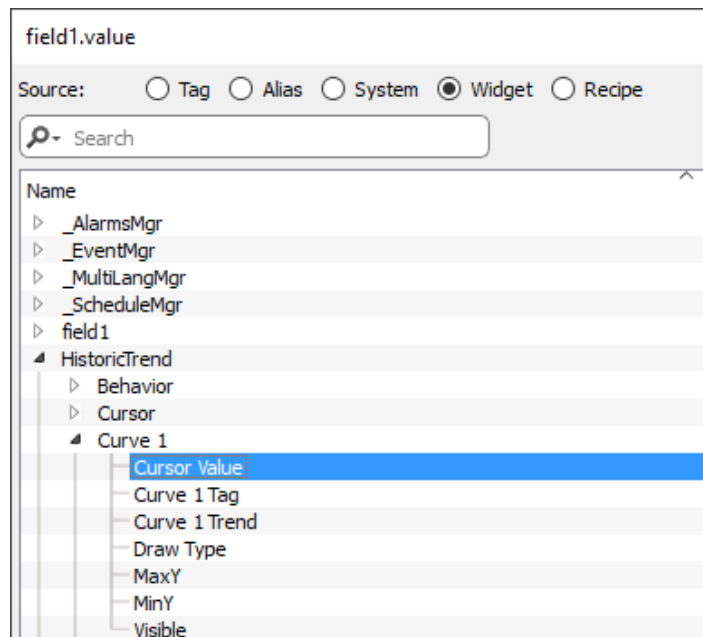
## Showing trend values (cursor)

Trend cursor displays the trend value at a specific point.

Use the actions **ShowTrendCursor** and **ScrollTrendCursor** to enable the trend cursor and move it to the required point to get the value of the curve at that particular point in time.

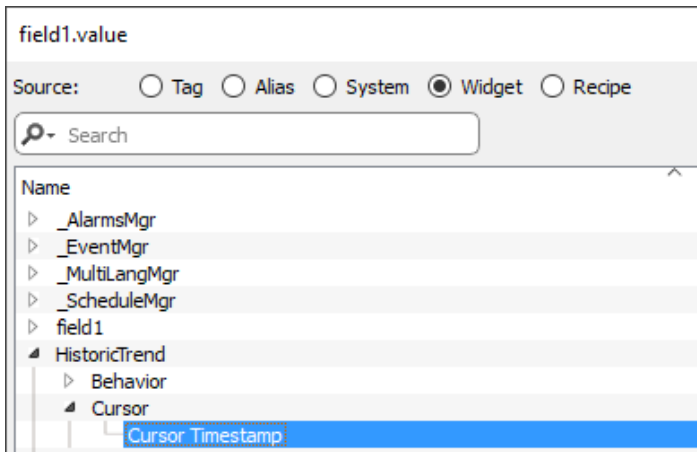


To display the value of the trend cursor on the page, define a numeric field and attach it to the **Cursor Value** widget tag.



In this example the Y axis value of the cursor is displayed.

To display the trend time stamp at the position of the cursor, use a Time/Data widget (available inside Basic->Controls category) and attach the widget's value to the **Cursor Timestamp** property of the trend widget.



## Modify trend widget properties at runtime

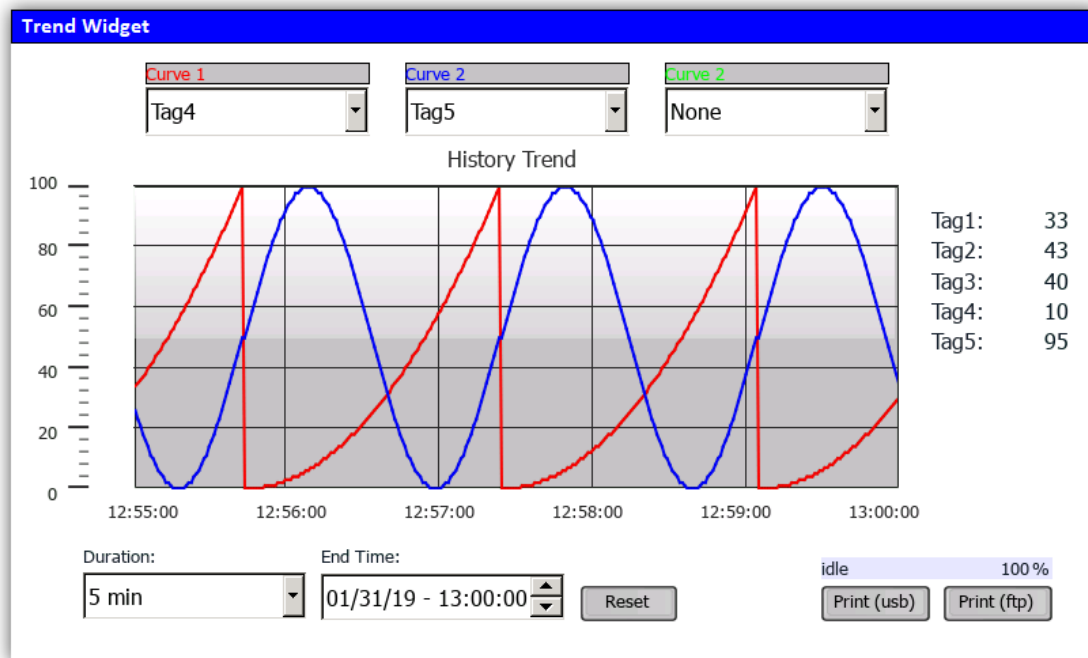
Using the attach to tag feature is possible to use tags to modifies some properties of the trend widgets at runtime.

### Example 1

Using :

- Page Duration
- End Time
- Curve Name

is possible to modify from the runtime application the zoom factor, the window period and the curve to draw.



**Properties**

**Trend : HistoricTrend**

- Num Curves: 3
- Page Duration: 5 min
- DataLink: Duration
- Access Type: R
- End Time: 0
- DataLink: EndTime
- Access Type: R
- Starting position: Right
- Behavior: [ ]
- Text: [ ]
- Grid: [ ]
- Cursor: [ ]
- X Scale: [ ]
- Y Scale: [ ]
- Curve 1
  - Curve 1 Trend: [ ]
  - DataLink: Selector1\_str
  - Access Type: R
  - MinY: 0
  - MaxY: 100
  - Color: [255, 0, 0]
  - Stroke Width: 2
- Curve 2
  - Curve 2 Trend: [ ]
  - DataLink: Selector2\_str
  - Access Type: R
  - MinY: 0
  - MaxY: 100
  - Color: [0, 0, 255]
  - Stroke Width: 2
- Curve 3: [ ]

1:Page1 Tags x

Name	Address
Duration	Duration int
EndTime	EndTime time
Selector1_str	Selector1_str string [20]
Selector2_str	Selector2_str string [20]
Selector3_str	Selector3_str string [20]

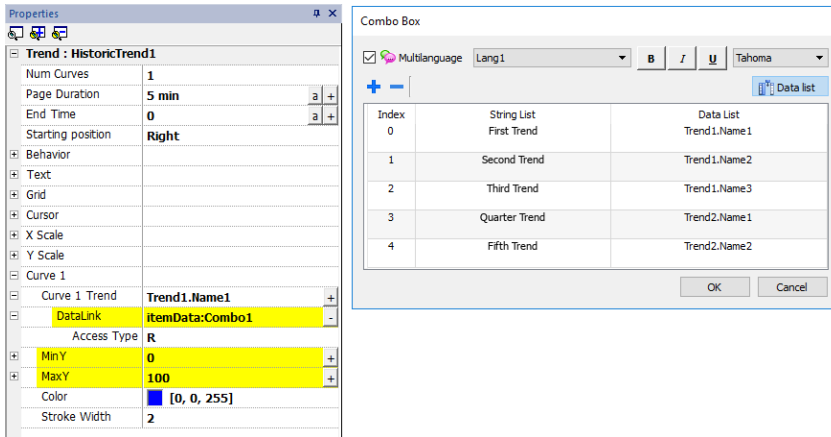
### Example 2

Curve property can be attached to a Combo Box to select the curve to draw

**Properties**

**Trend : HistoricTrend**

- Num Curves: 5
- Page Duration: 5 min
- End Time: 0
- Starting position: Right
- Behavior: [ ]
- Text: [ ]
- Grid: [ ]
- Cursor: [ ]
- X Scale: [ ]
- Y Scale: [ ]
- Curve 1
  - Curve 1 Trend: [ ]
  - MinY: 0
  - MaxY: 100
  - Color: [0, 0, 255]
  - Stroke Width: 2



## Trend widget gestures

Trend widgets support gesture commands:

Gesture	Description
pan	Touch the widget to scroll the curve within the widget area
pinch	Use two fingers to pinch the curve and perform zoom operations



**WARNING: Only multi touch HMI devices can generate pinch events**



Note: In order to support gestures on Y axis, Min/Max properties of the trend widget must be linked to Min/Max values of Behavior parameters (default for new trends).

Trend : RealtimeTrend	
Num Curves	1
Page Duration	5 min +
Y Page Size	100 +
Starting position	Right
Behavior	
Min Y	0 +
Max Y	100 +
X Labels	4 +
Y Labels	6 +
Background Image	true
Text	
Grid	
Cursor	
X Scale	
Y Scale	
Min	0 +
DataLink	y0:RealtimeTrend.wnd -
Access Type	R
Max	100 +
DataLink	y1:RealtimeTrend.wnd -
Access Type	R

## Request Samples

**Request Sample** property can be set for each curve and indicates the maximum numbers of samples read by the widget at one time from the trend buffer.



Tip: You normally do not need to modify the default value. Adjust it to fine tune performances in the trend widget refresh, especially when working with remote clients.

## Color bands

Use the color bands configuration to customize your graphs background, for example to make certain days or hours stand out (weekends, night hours, etc.).

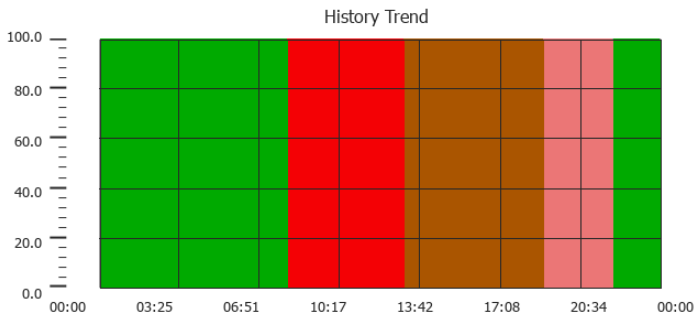
1. In the **Properties** pane, in **Color Bands** property click **+**: the **Configure Bands** window appears.
2. Click **+** to add as many colors you need.

3. Select multiple cells and click on a color band to assign the color to the selected range of cells.



Note: This feature only uses local time in the trend widget, not the global time option.

### Calendar color bands example



## Table trend widget

Path: **Widget Gallery**> **Basic**> **Trends/Graphs**

Display contents of a trend buffer inside a widget

**Trend Table****From:** 02/21/22 - 12:34:21**To:** 02/21/22 - 12:34:21

Refresh

5 Mins



Timestamp	Name1	Name2	Name3	Name4	Name5
02/21/22 - 12:34:09	0	0	0	0	0
02/21/22 - 12:34:10	1	2	3	4	4
02/21/22 - 12:34:11	2	2	6	8	8
02/21/22 - 12:34:12	3	2	9	12	12
02/21/22 - 12:34:13	4	2	12	16	16
02/21/22 - 12:34:14	5	2	15	20	20

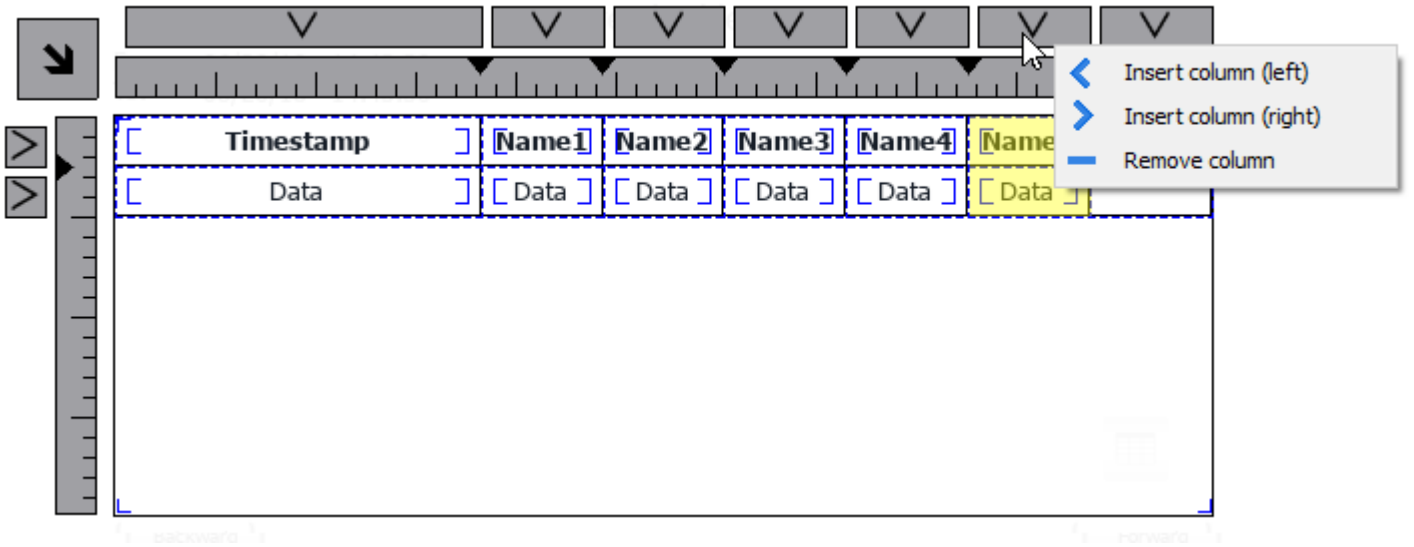
**Buttons:**

- REFRESH  
Retrieve trend data from internal buffer and refresh table view
- BACKWARD/FORWARD  
Move the display window forward or backward as specified in the duration parameter

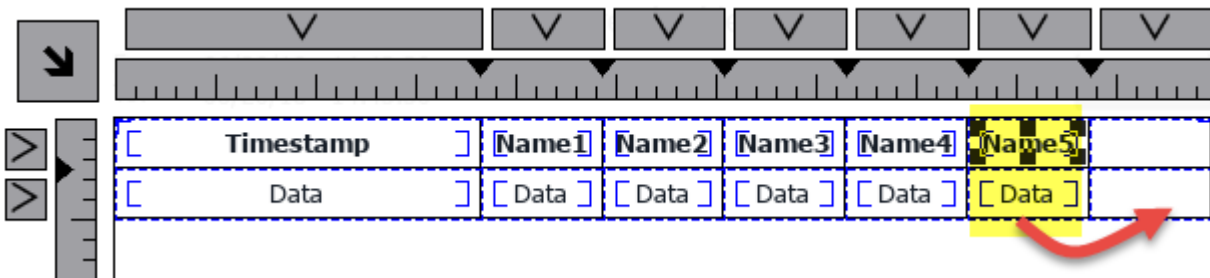
Parameter	Description
<b>TrendName</b>	Trend Buffer from which the samples are retrieved (see <a href="#">"Data logging" on page 292</a> )
<b>Heading</b>	Heading labels  The visible labels inside the PB610 Panel Builder 600editor are only placeholders, the actual labels that will be displayed are defined in the trend configuration (see <a href="#">"Data logging" on page 292</a> )
<b>Page Duration</b>	Time window to show
<b>End Time</b>	End time of the time window  This parameter can be used to scroll the time window. When zero, the end time is the current system time.
<b>Time Spec</b>	Time format: <ul style="list-style-type: none"> <li>• <b>Local</b> = show the time values of the HMI device.</li> <li>• <b>Global</b> = show the time values using UTC format.</li> </ul>
<b>Date Format</b>	Select the Date and Time format
<b>Table Layout</b>	Defines the characteristics of the scroll bar and allows to remove the header of the table

## Adding or removing trend columns

To add or remove a column, double-click on the grid to enter edit mode and right-click on the column selector to open the context menu from where to insert or remove a column.



Copy and past fields from another column

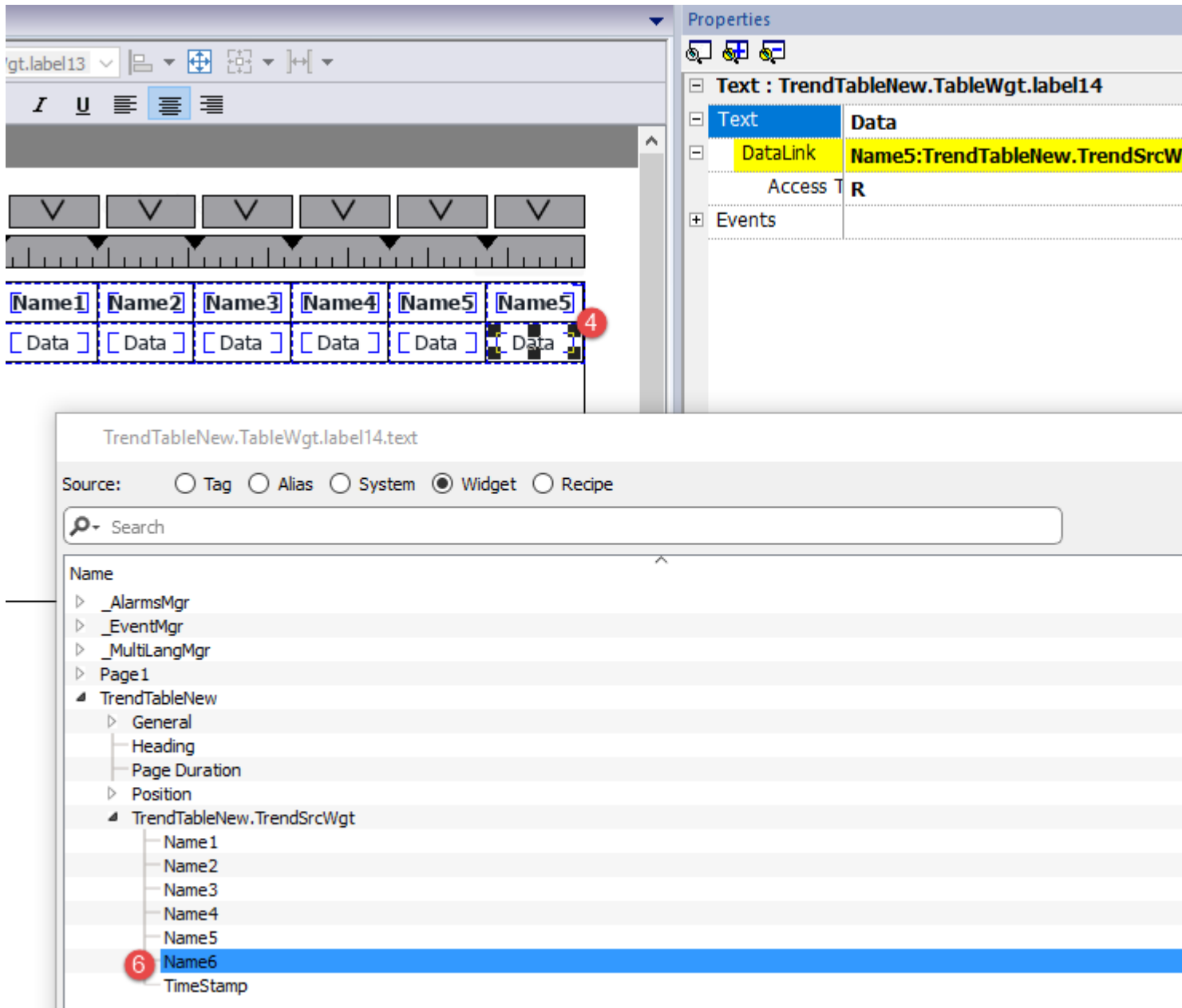


Then use the properties panel to select the trend element to add to the new columns



The screenshot shows a software interface with a table trend widget and its properties. The widget is a grid with columns labeled Name1 to Name5 and rows for Data. A red circle '1' highlights the Name5 column. The Properties panel on the right shows the widget's configuration, including a DataLink to Name5: TrendTableNew.TrendSrcWgt and an Access Type of R.

The screenshot shows the configuration dialog for 'TrendTableNew.TableWgt.label13.text'. It features a 'Source' section with radio buttons for Tag, Alias, System, Widget (selected), and Recipe. Below is a search bar and a tree view of the widget's structure. The tree view shows a hierarchy starting with 'TrendTableNew', followed by 'TrendTableNew.TrendSrcWgt', and then a list of items including Name1, Name2, Name3, Name4, Name5, Name6 (highlighted with a red circle '3'), and TimeStamp.



## Printing trend table

A trend table widget without buttons can be found and used from the print report gallery. The table can be drawn and enlarged to fill the entire page. If the number of lines to printed is greater of one page, the trend table will be printed using additional pages. See the "Table of functions and limits" on page 612 for the max number of printable rows.

Using the "attach to tag" feature is possible to use tags to define some properties of the historical trend to print at runtime:

- Page Duration
- End Time
- Curve Name

"Page Duration" with "End Time" define the piece of the trend buffer to print. "Curve Name" can be used to select the curve to show. An empty string means no curve to show.



**SetTrendView() and ScrollTrendToTime() are display macros and have no effect on report printing.**

Properties

**Trend : HistoricTrend**

- Num Curves: 3
- Page Duration: 5 min
- DataLink: Duration
  - Access Type: R
- End Time: 0
  - DataLink: EndTime
    - Access Type: R
  - Starting position: Right
- Behavior
- Text
- Grid
- Cursor
- X Scale
- Y Scale
- Curve 1
  - Curve 1 Trend
  - DataLink: Selector1\_str
    - Access Type: R
    - MinY: 0
    - MaxY: 100
    - Color: [255, 0, 0]
    - Stroke Width: 2
- Curve 2
  - Curve 2 Trend
  - DataLink: Selector2\_str
    - Access Type: R
    - MinY: 0
    - MaxY: 100
    - Color: [0, 0, 255]
    - Stroke Width: 2
- Curve 3

1:Page1 Tags x

Name	Address
Duration	Duration int
EndTime	EndTime time
Selector1_str	Selector1_str string [20]
Selector2_str	Selector2_str string [20]
Selector3_str	Selector3_str string [20]

# 23 Data transfer

---

Data transfer allows you transferring variable data from one device to another. Using this feature an HMI device can operate as a gateway between two devices, even if they do not use the same communication protocol.

---

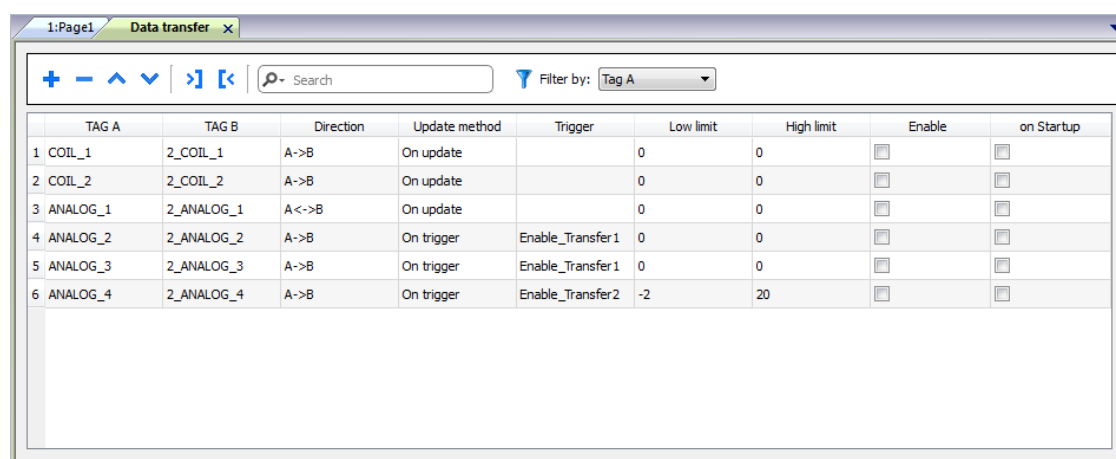
<b>Data transfer editor</b> .....	<b>322</b>
<b>Exporting data to .csv files</b> .....	<b>324</b>
<b>Data transfer limitations and suggestions</b> .....	<b>324</b>

# Data transfer editor

Path: **ProjectView** > **Config** > double-click **Data transfer**

Use the Data transfer editor to map transfer rules.

Each line in the Data transfer editor defines a mapping rule between two tags. Define more mapping rules if you need different direction, update method or trigger.




	TAG A	TAG B	Direction	Update method	Trigger	Low limit	High limit	Enable	on Startup
1	COIL_1	2_COIL_1	A->B	On update		0	0	<input type="checkbox"/>	<input type="checkbox"/>
2	COIL_2	2_COIL_2	A->B	On update		0	0	<input type="checkbox"/>	<input type="checkbox"/>
3	ANALOG_1	2_ANALOG_1	A<->B	On update		0	0	<input type="checkbox"/>	<input type="checkbox"/>
4	ANALOG_2	2_ANALOG_2	A->B	On trigger	Enable_Transfer 1	0	0	<input type="checkbox"/>	<input type="checkbox"/>
5	ANALOG_3	2_ANALOG_3	A->B	On trigger	Enable_Transfer 1	0	0	<input type="checkbox"/>	<input type="checkbox"/>
6	ANALOG_4	2_ANALOG_4	A->B	On trigger	Enable_Transfer 2	-2	20	<input type="checkbox"/>	<input type="checkbox"/>






To add a new rule, click **+**: a new tag line is added.

## Data transfer toolbar

Parameter	Description
<b>Import/ Export</b>	Imports or exports data transfer settings from or to a .csv file.
<b>Search</b>	Displays only rows containing the search keyword.
<b>Filter by</b>	Display only rows matching filter and search field.

## Data transfer parameters

Parameter	Description
<b>TAG A/ TAG B</b>	Pair of tags to be mapped for exchanging through the HMI device.
<b>Direction</b>	Transfer direction.  <b>A-&gt;B</b> and <b>B-&gt;A</b> : Unidirectional transfers, values are always copied from one tag and sent to the other tag in the specified direction.  <b>A&lt;-&gt;B</b> : Bidirectional transfer, values are transferred to and from both tags.
<b>Update Method</b>	<b>On trigger</b> : Data transfer occurs when the value of the tag set as trigger changes above or below the values set as boundaries. Limits are recalculated on the previous tag value, the same that triggered the update.   Note: This method applies only to unidirectional transfers (A->B or B->A).

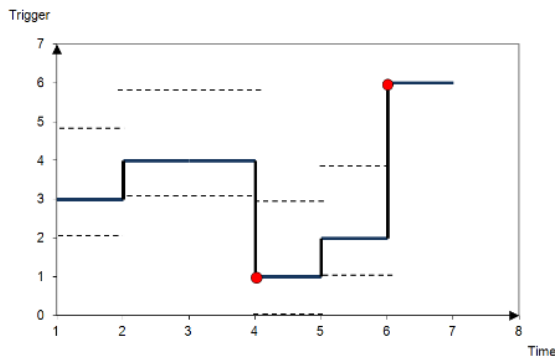
Parameter	Description
	<p><b>On Update:</b> Data transfer occurs whenever the value of the source tag changes.</p> <p> Note: This method applies both to unidirectional and to bidirectional transfers (A-&gt;B, B-&gt;A and A&lt;-&gt;B).</p> <p> Note: The Runtime cyclically monitors source tags changes (trigger tag when using On Trigger or tags to transfer when using On Update) based on Tag editor <b>Rate</b> parameter. If <b>Rate</b> setting for source Tag is 500 ms (default), the system checks for updates every 500 ms.</p> <p> Note: Changes on source tags faster than <b>Rate</b> may be not detected .</p>
<p><b>Trigger, High limit, Low limit</b></p>	<p>Tag that triggers the data transfer process. When this tag changes its value outside the boundaries set as <b>High limit</b> and <b>Low limit</b>, data transfer is started. The range of tolerance is recalculated according to the specified limits on the tag value which triggered the previous update. No action is taken if the change falls within the limits.</p> <p>This mechanism allows triggering data transfers only when significant variations of the reference values occur.</p> <p><b>Low limit</b> is less or equal to zero.</p> <p> Note: If both <b>Low limit</b> and <b>High limit</b> are set to "0", data transfer occurs whenever the value of the trigger tag changes.</p>
<p><b>Enable</b></p>	<p>Enable or disable the data transfer.</p>
<p><b>on Startup</b></p>	<p>When selected, data transfer is forced:</p> <ul style="list-style-type: none"> <li>• on HMI startup if the quality of the source tag is good</li> <li>• after communication errors, when the associate device nodes return active</li> </ul> <p>See "<a href="#">Objects</a>" on page 549 for details on quality.</p> <p> <b>Important: Data transfers executed on startup may have major impact on the HMI device boot time. Enable this option only when necessary.</b></p>

### Example of limit setting

**High limit** = 1,9

**Low limit** = - 0,9

• = points where the data transfer is triggered




## Exporting data to .csv files

Configuration information for data transfers can be exported to a .csv file.

### Example of data transfer settings in .csv file

A	B	C	D	E	F	G	H	I	J
COIL_1	2_COIL_1	A->B	On update		0	0	data1	true	1
COIL_2	2_COIL_2	A->B	On update		0	0	data2	true	1
ANALOG_1	2_ANALOG_1	A<->B	On update		0	0	data3	true	1
ANALOG_2	2_ANALOG_2	A->B	On trigger	Enable_Transfer1	0	0	data4	true	1
ANALOG_3	2_ANALOG_3	B->A	On trigger	Enable_Transfer1	0	0	data5	true	1
ANALOG_4	2_ANALOG_4	A->B	On trigger	Enable_Transfer2	-10	20	data6	true	1

Column	Description
A to G	Same data as in the Data transfer editor
H	Unique identifier automatically associated to each line.   <b>Important: When you edit the .csv file and you add any extra line, make sure you enter a unique identifier in this column.</b>
I and J	Reserved for future use.



Import/export use the separator character defined inside Windows Regional Settings.

## Data transfer limitations and suggestions

Correct definition of data transfer rules is critical for the good performance of the HMI devices. To guarantee reliability of operation and performance, keep in mind the following rules.

### On trigger method

The **On trigger** method allows only unidirectional transfers, (A->B or B->A)

Data transfer based on the **On Trigger** mode should be preferred since it allows you to force the transfer and monitors only the trigger tags and not all the tags involved in the transfer.

## On update method

The **On update** method allows changing the values in accordance with the direction settings only when the source value changes.

Using the **On Update** method you force the system to continuously read all the defined source tags to check if there are changes that need to be transferred. The default value of the update rate of each tag is 500 ms and can be modified with Tag editor.

## Performance observations

Data transfer performance depends on:

- number of data transfers defined,
- number of data transfers eventually occurring at the same time,
- frequency of the changes of the PLC variables that are monitored,



**Important: Always test performance of operation during project development.**



**Important: If inappropriately set, data transfer tasks can lead to conditions where the tags involved create loops. Identify and avoid such conditions.**



Tip: Use the scheduler to calibrate the update rate based on the performance of your entire project.



Tip: Use array type tags to optimize data transfer and reduce workload.



Tip: Reduce the number of data transfers to reduce page change time and boot time.





# 24 Offline node management

---

When one of the controllers communicating with the HMI device goes offline, communication performance of the system may eventually decrease.

The offline node management feature recognizes offline controllers and removes them from communication until they come back online.

Additionally, if you know that any of the controllers included in the installation is going to be offline for a certain time, you can manually disable it to maximize system performance.



Note: This feature is not supported by all communication protocols. Check protocol documentation to know if it is supported or not.

---

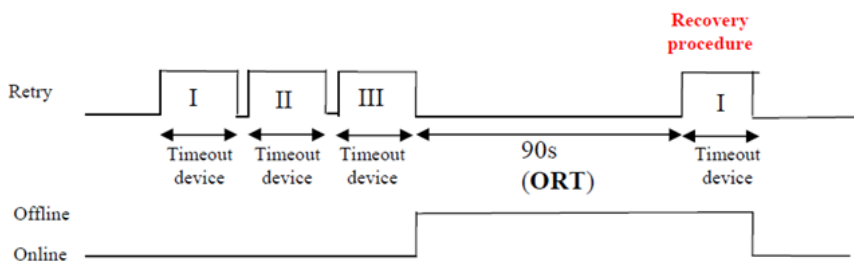
<b>Offline node management process</b> .....	<b>328</b>
<b>Manual offline node management process</b> .....	<b>328</b>
<b>Manual offline configuration</b> .....	<b>328</b>
<b>Automatic offline node detection</b> .....	<b>329</b>

## Offline node management process

Steps of the process are:

- The system communicates normally with a certain device. When the device is not responding to a communication request, the system will repeat the request twice before declaring the device offline.
- When a device is offline, the system sends communication requests to the device with a longer interval, called Offline Retry Timeout. If the device answers to one of these requests, the system declares it online and restarts normal communication.

The diagram shows the three communication attempts and the recovery procedure that starts when the Offline Retry Timeout is elapsed.



## Manual offline node management process

Offline node management can be done manually. When a specific device is online and it is communicating normally you can:

- use an action to declare the device offline: the system stops communication with the device.
- use an action to declare the device online: the system restarts normal communication with the device.

## Manual offline configuration

When you know that some devices in communication with the HMI device are going to remain offline for a certain period of time, you can exclude them from communication using the **EnableNode** action.



**WARNING: All disabled device nodes will remain disabled if the same project is downloaded on the device, on the other hand, if a different project is downloaded, all disabled devices will be re-enabled. The same happens with a package update.**



Tip: To make this feature more dynamic, you may decide not to indicate a specific **NodeID** but attach it to the value of a tag or to an internal variable created to identify different devices that might be installed in your network.

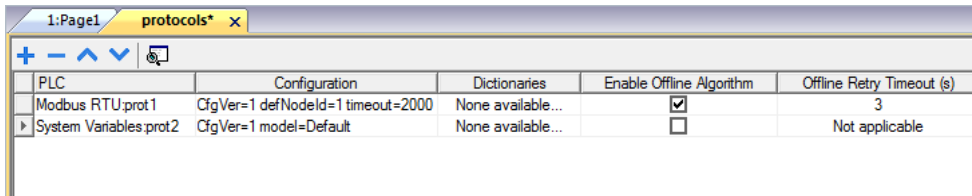


Note: When using the action **EnableNode** to force a device node back online, communication will start immediately.

## Automatic offline node detection

When a device is not answering to communication requests, it is de-activated. The HMI device stops sending requests to this device. After three seconds, the HMI device sends a single command to check if device is available, if so the communication is restarted, otherwise it is disabled for another timeout interval.

Default settings can be modified in Protocol editor.



PLC	Configuration	Dictionaries	Enable Offline Algorithm	Offline Retry Timeout (s)
Modbus RTU:prot1	CfgVer=1 defNodeId=1 timeout=2000	None available...	<input checked="" type="checkbox"/>	3
System Variables:prot2	CfgVer=1 model=Default	None available...	<input type="checkbox"/>	Not applicable



Note: Not all protocols support this feature.

Parameter	Description
<b>Enable Offline Algorithm</b>	Enables offline management for the protocol
<b>Offline Retry Timeout</b>	Interval in seconds for the retry cycle after a device has been deactivated. Range: 1–86.400 seconds (24h).



# 25 Multi-language

---

Multi-language feature has been designed for creating HMI applications that include texts in more than one language at the same time

Multi-language feature uses code pages support to handle the different languages. A code page (or a script file) is a collection of letter shapes used inside each language.

Multi-language feature can be used to define languages and character sets in a project. PB610 Panel Builder 600 also extends the TrueType Fonts provided by Windows systems to provide different font faces associated with different character sets.

PB610 Panel Builder 600 also allows you to provide strings for each of the languages supported.

PB610 Panel Builder 600 also allows you to change the display language so that you can see the page look and feel during the design phase.

Appropriate fonts may need to be installed to manage the different languages. When adding font files, be aware that there may be license rights that need to be acquired in order to use them.



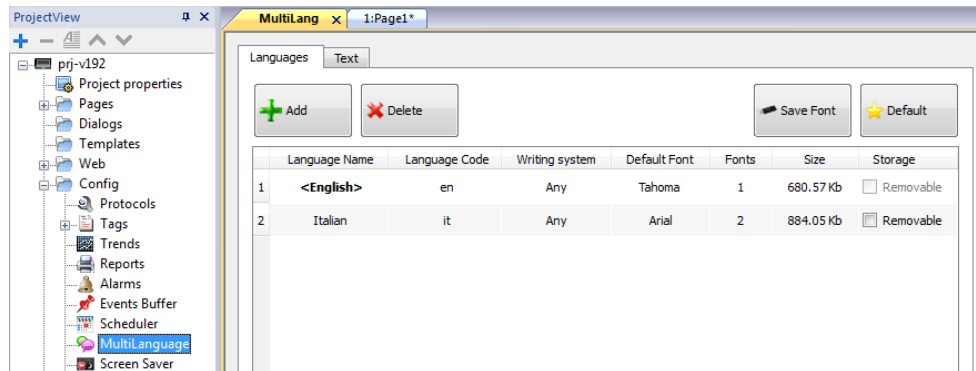
On the Internet, is easy to find several fonts provided with the open-source license as, e.g., the Noto family offered by Google ([www.google.com/get/noto](http://www.google.com/get/noto))

---



<b>The Multi-language editor</b> .....	<b>332</b>
<b>Changing language</b> .....	<b>334</b>
<b>Multi-language widgets</b> .....	<b>334</b>
<b>Exporting/importing multi-language strings</b> .....	<b>336</b>
<b>Changing language at runtime</b> .....	<b>338</b>
<b>Limitations in Unicode support</b> .....	<b>338</b>

# The Multi-language editor

Path: **ProjectView** > **Config** > double-click **MultiLanguage**




## Language settings

Parameter	Description
<b>Language Name</b>	Name identifying the language in the project.
<b>Language Code</b>	ISO 639 language code identifier, used to match language items when importing resources from external xml files.
<b>Writing system</b>	Select the set of fonts to be used with the language
<b>Default Font</b>	Default font for project's widgets.  Note: When you choose a new font you are prompted to replace the font used in the widgets you already created.
<b>Fonts</b>	Number of fonts associated with the selected language.
<b>Size</b>	Memory used to store font files.
<b>Storage</b>	Location of file fonts is a removable external memory.  Tip: Store large font files on removable memory to free memory requirements in the HMI device.

## Adding a language

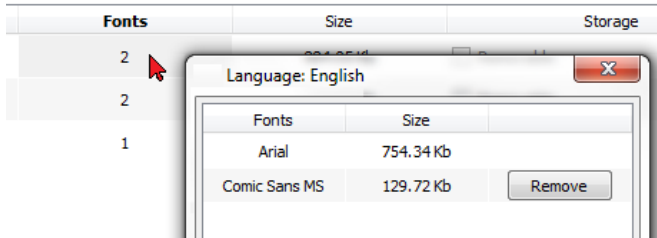
1. In the **Languages** tab, click **+**: a line is added to the table.
2. Enter all language settings.
3. Click **Default** to set the selected language as the default language when the Runtime starts.
4. Click **Save Font** to copy the fonts you marked as **Removable** on an external memory.

 **Important: Font files configured to be stored on removable memory must be provided to the final user to complete font installation on the HMI device.**

## Removing fonts

To remove fonts no longer needed:

1. Click on the font number in the Multi-language editor: a dialog with the list of the used fonts is displayed.



2. Select the fonts to be removed and click **Remove**: removed fonts are replaced with the default font.

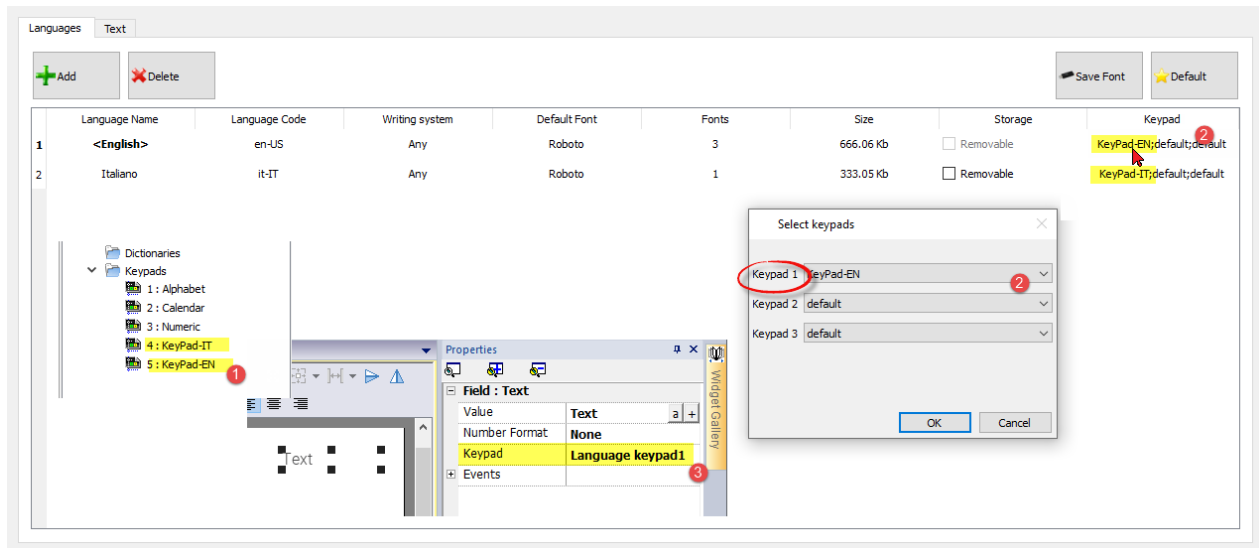
## Language Keyboards

Up to three different custom keyboards can be defined for each language. The runtime will then select the keyboard based on the active language.

How to configure:

1. Create your own keyboards (eg KeyPad-EN and KeyPad-IT)
2. For each language, select the associated custom keyboard
3. In the data fields, select the keyboard of the language to use (e.g. keypad1).

The application on the HMI device will display the custom keyboard according to the selected language





# Changing language

## Changing language during page design

A combo box is available for changing language during page design. If no texts appears, please check **Text** tab in the Multilanguage editor and insert missing string.

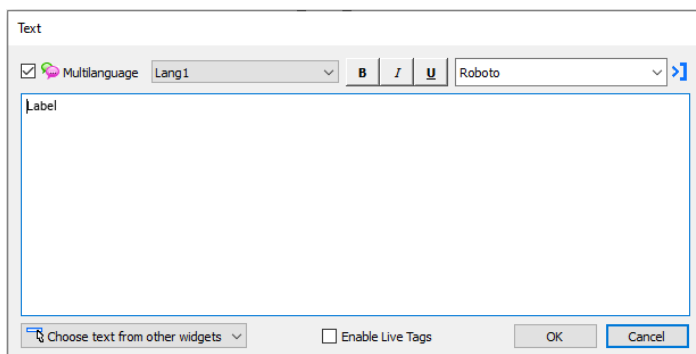


## Multi-language widgets

Multi-language support is available for objects such as buttons, static text, messages, alarm descriptions and pop-up messages.

### Multi-language for label widgets

Double-click on a text widget in a page to open the **Text** dialog.



Enable/disable multi-language function, edit the text for the selected language and choose the font.

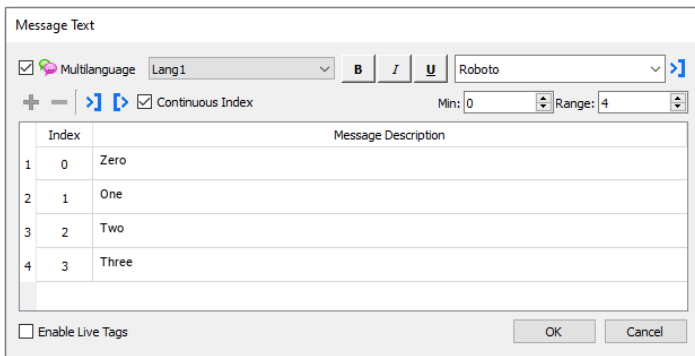


Note: Bold, italic and color properties set here for the widget are applied to all languages .

Parameter	Description
<b>Multilanguage</b>	Enable/disable multi-language function for the widget.
<b>Choose text from other widget</b>	Click on button to browse existing message strings in project to pick text for the widget.

### Multi-language for message widgets

Double-click on a message widget in a page to open the **Message Text** dialog.

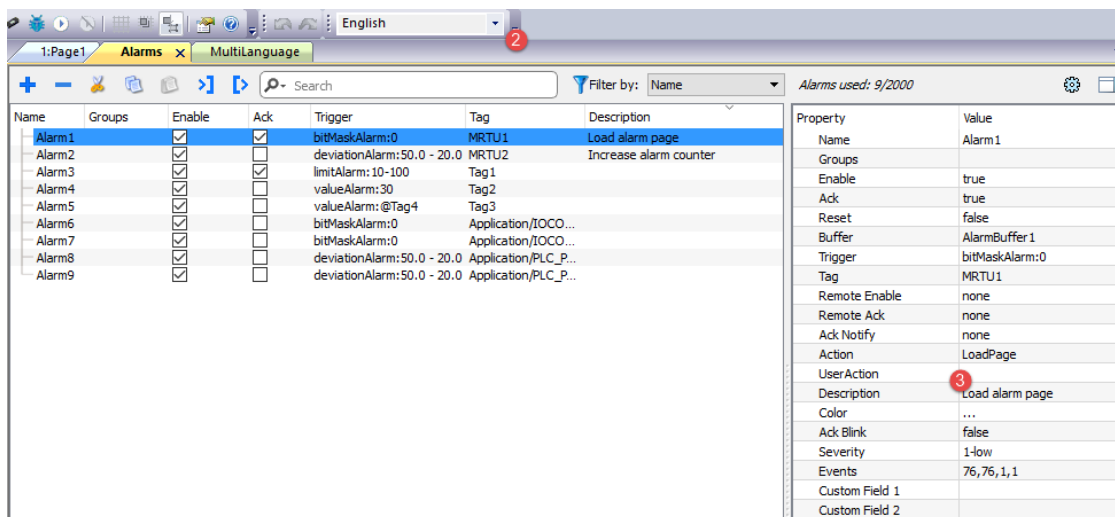


Parameter	Description
<b>Multilanguage</b>	Enable/disable multi-language function for the widget.
<b>Continuous Index</b>	Index for the widget is set of contiguous numbers (example 3, 4,5,6)
<b>Min</b>	Starting number for index
<b>Range</b>	Number of messages
<b>Choose text from other widget</b>	Click on button to browse existing message strings in project to pick text for the widget.

## Multi-language for alarm messages

To add a multi-language strings for alarm messages:

1. Open the Alarm editor.
2. Select a language using the language combo box.
3. Enter the text for the alarm in the **Description** column.

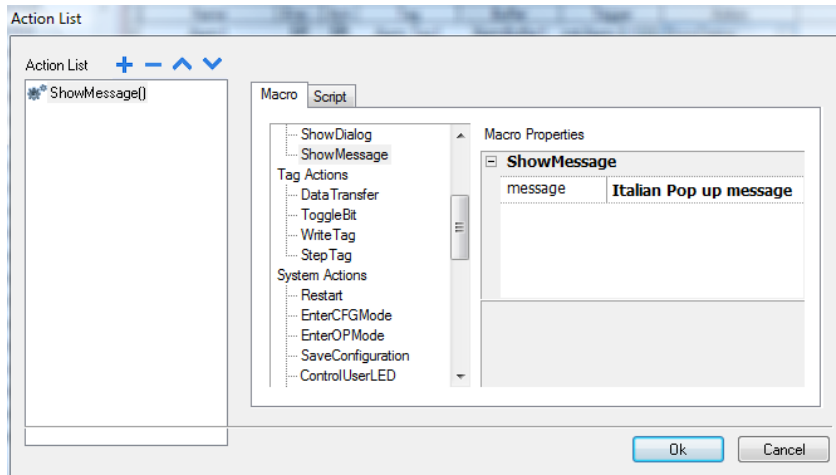


Tip: Text labels with alarm states displayed by alarms widgets can be translated or personalized through the Multilanguage text editor.

## Multi-Language for pop-up messages

To add a multi-language pop-up message:

1. Select a language from the language combo box.
2. Add the Page action **ShowMessage** and enter the text in the selected language.



## Exporting/importing multi-language strings

The easiest way to translate a project into multiple languages is to export all texts to a .csv file, translate the resulting document and then import the translated text back into the project.

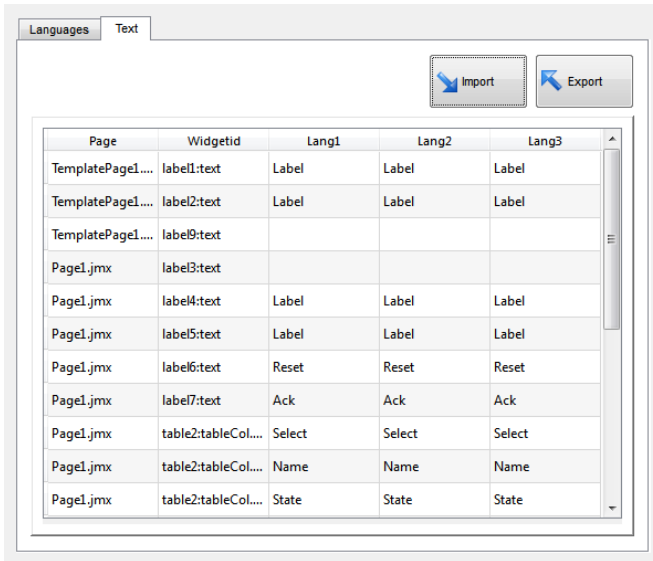
 **Important:** The .csv file exported by PB610 Panel Builder 600 is coded in Unicode, to edit it you need a specific tool supporting Unicode encoded .csv files.

### Exporting and reimporting strings

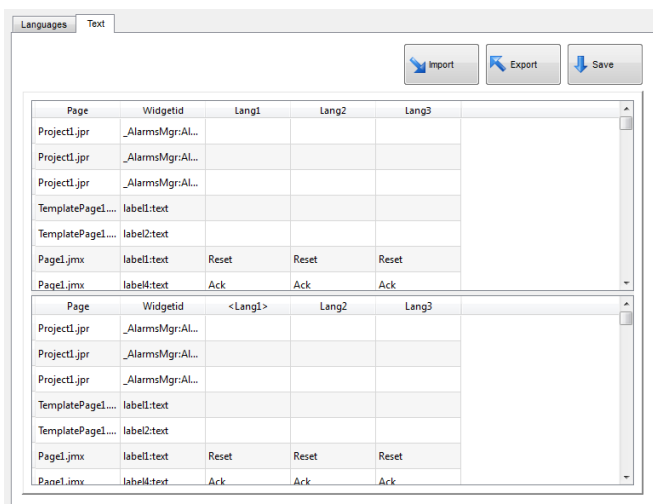
*Path: ProjectView > Config > double-click MultiLanguage*

To export and re-import multi-language strings:

1. In the **Text** tab, click **Export**: all multi-language strings are exported to a .csv file.



**Important:** Set all languages that will be used in the project before exporting the file. This will guarantee that the exported file will contain all columns and language definitions.



2. Once the strings have been translated, click **Import** to re-import them into the project: strings are imported matching the widget ID and the page number of each widget.
3. Click **Save** to save the new widget data.



**Note:** To change the separator used in the exported file, change the regional settings of your computer. When importing, the separator information is retrieved from the file; if not found, the default character "," is used.

## Import constraints

The following formats are supported for import:

- Comma Separated Values (.csv)
- Unicode Text (.txt)



Note: Use the Unicode Text file format when you import a file modified using Microsoft® Excel®.

## Changing language at runtime

### Changing language with an action

After the project download, the HMI Runtime will start using the language set as default. You can change the language using the **SetLanguage** action. See "[MultiLanguage actions](#)" on page 199.



Note: Once the language has been changed, it will be used also in future sessions.

The active language code is available from JavaScript API. See "[curLangCode](#)" on page 567 for additional details.

### Missing fonts

When you change language, if the required fonts are not available in the device memory, a pop-up message prompts you to insert the memory card containing the missing fonts. At the end of the operation you can remove the memory card.



## Limitations in Unicode support

PB610 Panel Builder 600 has been designed for working with Unicode text. However, for compatibility issues with some platforms, Unicode is supported only in a subset of properties.

Area	Property	Charset Accepted	Reserved Chars/Strings
Protocol editor	Alias	ASCII [32..126]	(space) , ; : . < * > '
Tag editor	Name	ASCII [32..126]	\\ / * ? : > <   " & # % ; =
	Group	ASCII [32..126]	<New> \\ / * ? : > <   " & # % ;
	Comment	Unicode	
Trends	Name	ASCII [32..126]	\\ / * ? : > <   " & # % ;
Printing Reports	Name	ASCII [32..126]	\\ / * ? : > <   " & # % ;

Area	Property	Charset Accepted	Reserved Chars/Strings
<b>Alarms</b>	Name	ASCII [36..126]	\ / * ? : > <   " & # % ;
	Description	Unicode	[] - for live tags, \ escape seq for [ and \
<b>Events</b>	Buffer Name	ASCII [32..126]	\ / * ? : > <   " & # % ;
<b>Scheduler</b>	Name	ASCII [32..126]	\ / * ? : > <   " & # % ;
<b>Languages</b>	Language Name	ASCII [32..126]	\ / * ? : > <   " & # % ;
	Texts in widgets	Unicode	-
	Texts from import files	Unicode	-
<b>User Group</b>	Group Name	a-z A-Z _	admin,guest,unauthorized
	Comments	Unicode	-
<b>User</b>	Name	ASCII [32..126]	\ / * ? : > <   " & # % ;
	Password	Unicode	-
	Comment	Unicode	-
<b>Recipes</b>	Name	ASCII [32..126]	\ / * ? : . > <   " & % ; ,
	Set Name	ASCII [32..126]	\ / * ? : . > <   " & % ; ,
	Element name	ASCII [32..126]	\ / * ? : . > <   " & % ; ,
<b>General</b>	Project Name	A-Z,a-z,0-9,-,_,	"PUBLIC", "readme", "index.html"
	Page Name	A-Z,a-z,0-9,-,_,	-
	Dialog Page Name	A-Z,a-z,0-9,-,_,	-
	Template Page Name	A-Z,a-z,0-9,-,_,	-
	Keypad Name	A-Z,a-z,0-9,-,_,	-
	Files (Images/Video/etc..)	A-Z,a-z,0-9,-,_,	-
	Widgets ID	A-Z,a-z,0-9,-,_,	-
<b>Runtime</b>	PLC Communication	UTF-8, Latin1, UCS-2BE, UCS-2LE, UTF-16BE, UTF-16LE	-

Area	Property	Charset Accepted	Reserved Chars/Strings
<b>OPCUA</b>	Node name	Unicode	\ "
	Manufacture name	Unicode	\ "
	Product name	Unicode	\ "
	Server name	Unicode	\ "
	Organization	Unicode	\ "
	Location	Unicode	\ "
	State	Unicode	\ "
	Country	Unicode	\ "
	DNS	Unicode	\ "
	IP Address	Unicode	\ "
<b>MQT</b>	OnChange	Unicode	-
	OnTimer	Unicode	-
	All editable fields	Unicode	-
<b>Data Base Links</b>	Name	Unicode	-
	DSN	Unicode	-
	Username	Unicode	-
	Password	Unicode	-
	Description	Unicode	-
	Database	Unicode	-

# 26 Scheduler

---

PB610 Panel Builder 600 provides a scheduler engine that can execute specific actions at set intervals, or on a time basis.

Creating a schedule is typically a two-step process:

1. You create a schedule with a list of actions to be executed when the scheduled event occurs. You do this in the Scheduler editor
2. You create a runtime user interface that allows the end-user to change settings for each schedule. You do this adding a **Scheduler** widget to a page of your project and configuring it to fit user scheduling needs.

---

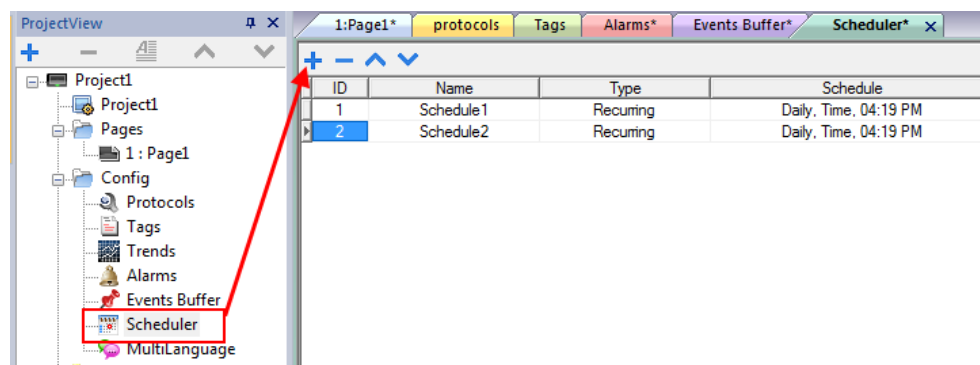
<b>Creating a schedule</b> .....	<b>342</b>
<b>HighResolution schedule</b> .....	<b>343</b>
<b>Recurring schedule</b> .....	<b>343</b>
<b>Configuring location for schedules</b> .....	<b>345</b>
<b>Configuring the Scheduler widget</b> .....	<b>345</b>
<b>Scheduling events at runtime</b> .....	<b>346</b>



# Creating a schedule

Path: **ProjectView**> **Config**> double-click **Scheduler**

- Click **+** to add a schedule.



## Schedule parameters

Parameter	Description
<b>ID</b>	Unique code assigned automatically to the schedule
<b>Name</b>	Name of schedule
<b>Type</b>	Type of schedule: <ul style="list-style-type: none"> <li>• <b>Recurring</b>, see "<a href="#">Recurring schedule</a>" on the facing page for details.</li> <li>• <b>HighResolution</b>, see "<a href="#">HighResolution schedule</a>" on the facing page for details</li> </ul>
<b>Schedule</b>	Scheduler settings and options. See " <a href="#">Recurring schedule</a> " on the facing page for details.
<b>Action</b>	<p>Actions to be executed at the scheduled time</p> <p>The macros added in the action field are executed on the server-side with the exception of the below macros that will be executed even on client-side (e.g. PB4Web).</p> <ul style="list-style-type: none"> <li>• loadPage</li> <li>• prevPage</li> <li>• nextPage</li> <li>• showDialog</li> <li>• showMessage</li> <li>• setLanguage</li> <li>• jsAction</li> </ul>
<b>Priority</b>	Priority level for the event. If two schedules occur at the same time, the event with the higher priority will be executed first.

## HighResolution schedule

The **HighResolution** schedule is used to perform actions that need to be repeated at specified intervals. The interval between executions is set in milliseconds in the **Schedule** column.





Note: You cannot change at runtime the settings of this type of schedule. If you need to change the action time settings at runtime, choose **Recurring** schedule and set **Type** to **Every**. See "[Recurring schedule](#)" below for details.

## Recurring schedule

The Recurring schedule is used to perform actions at specified points in time. Settings can be modified at runtime.

### Recurring scheduler parameters

Parameter	Description
<b>Type</b>	Frequency of the scheduled actions
<b>Mode</b>	Specific settings required by each scheduler type
<b>Condition</b>	<p>Boolean tag (true/false) to activate the specified actions at the moment the timer is triggered. Actions will be executed if tag = true. By default, actions are executed when the timer is triggered.</p> <p> Note: Only tags attached to the Boolean data type are shown.</p>
<b>Actions</b>	<p>Actions to be executed by the schedule.</p> <p> <b>Important: Actions and schedule parameters cannot be modified at runtime</b></p>
<b>Date</b>	Date when the scheduled actions will be executed
<b>Time/Offset</b>	<p>This field display one of the following:</p> <p><b>Time</b> = when the scheduled actions will be executed</p> <p><b>Offset</b>= delay or advance with respect to the selected mode.</p>
<b>Location</b>	Reference location to calculate sunset/sunrise time.
<b>weekdays</b>	Days of the week in which the scheduled actions will be executed.
<b>On startup</b>	Executes schedule at start up
<b>Enable schedule</b>	Enables/disables the schedule
<b>Execute only at startup</b>	Executes the schedule only once at start up

## Schedule type options

Option	Description
<b>By Date</b>	Actions are executed on the specified date and time.
<b>Daily</b>	Actions are executed daily at the specified time.
<b>Every</b>	Actions are executed with the specified interval (Range: 1 s–1 day)
<b>Hourly</b>	Actions are executed every hour at the specified minute.
<b>Monthly</b>	Actions are executed every month at the specified date and time.
<b>Weekly</b>	Actions are executed every week on the specified weekday(s) and time.
<b>Yearly</b>	Actions are executed every year at the specified date and time.

## Schedule mode options

Option	Description
<b>Time</b>	Depends on the schedule type. Allows you to specify date/time/week data.
<b>Random10</b>	Actions are executed in the time interval of 10 minutes before or after the set time. For example, if set time is 10:30, actions are executed any time between 10:20 and 10:40.
<b>Random20</b>	Actions are executed in the time interval of 20 minutes before or after the set time. For example, if set time is 10:30, actions are executed any time between 10:10 and 10:50.
<b>Sunrise+</b>	Actions are executed with a specified delay after sunrise. The delay is set in minutes/hours and sunrise time is location specific.
<b>Sunrise-</b>	Actions are executed with a specified advance before sunrise. The advance is set in minutes/hours and sunrise time is location specific.
<b>Sunset+</b>	Actions are executed with a specified delay after sunset. The delay is set in minutes/hours and sunset time is location specific.
<b>Sunset-</b>	Actions are executed with a specified advance before sunset. The advance is set in minutes/hours and sunset time is location specific.

See "[Configuring location for schedules](#)" on the facing page for details on sunset and sunrise settings.



Note: **Mode** options are not available for all schedule types.

## Configuring location for schedules

Scheduled actions can be configured to be executed at a specific time with respect to sunrise and/or sunset. To do this you need to define the correct location, based on UTC information. The system will automatically calculate the sunrise and sunset time.

Only a few locations are available by default. If your location is not listed, you can add it by entering latitude, longitude and UTC information in the Target\_Location.xml file.



**Important: Each platform has its own Target\_Location.xml file.**

### Location files position

Application	Location file path
<b>PB610 Panel Builder 600</b>	<i>ABB\Panel Builder 600 Suite\languages\shared\studio\config\Target_Location.xml</i>
<b>HMI Devices</b>	<i>ABB\Panel Builder 600 Suite\runtime\&lt;HW Platform&gt;\config\Target_Location.xml</i>
<b>Simulator</b>	<i>ABB\Panel Builder 600 Suite\simulator\config\Target_Location.xml</i>
<b>PB610 PC Runtime</b>	<i>ABB\Panel Builder 600 Suite\server\config\Target_Location.xml</i>

For example, the information for Greenwich (UK) is shown below:

```
<file city="Greenwich,UK" latitude="51.47" longitude="0" utc="0"/>
```

Location information is also displayed in the dialog together with sunset and sunrise times.

**Schedule1 Properties**

Type:  Date:

Mode:  Offset:

Condition:  Location:

Actions:

Mon  
 Tues  
 Wed  
 Thurs  
 Fri  
 Sat  
 Sun

Calculated System Times:(GMT)  
 SunRise Time:04:35:04  
 SunSet Time:19:20:08

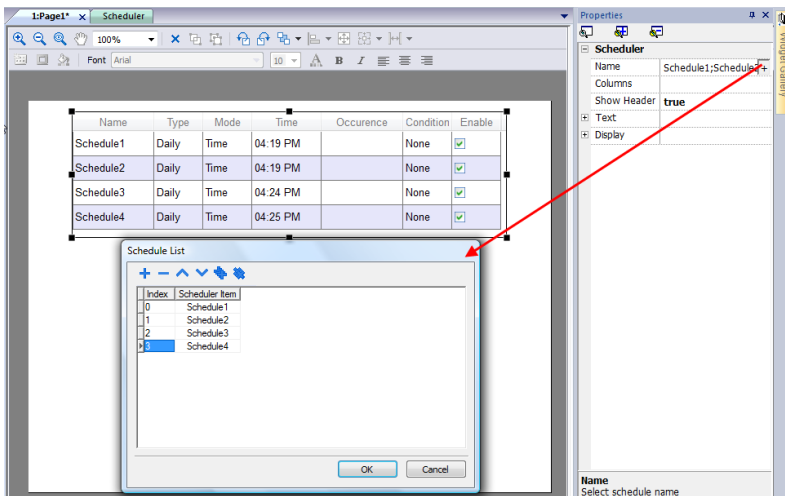
On startup  
 Enable schedule

Ok Cancel

## Configuring the Scheduler widget

To display scheduler data on a page:

1. Drag and drop a **Scheduler** widget from the widget gallery into the page.
2. In the **Properties** pane, click **+** for the **Name** parameter: the **Schedule List** dialog is displayed.
3. Add all the schedules you want to display in the page.



4. In the **Properties** pane, customize all settings.

## Scheduler settings

Parameter	Description
<b>Name</b>	Schedule to be displayed
<b>Columns</b>	Columns to be displayed and their characteristics
<b>Show Header</b>	Shows/hides column headers
<b>Time Spec</b>	Time to be displayed at runtime
<b>Text</b>	Font used for text
<b>Display</b>	Table styles

## Scheduling events at runtime

At runtime you can modify the following scheduling parameters.

Name	Type	Mode	Time	Occurrence	Condition	Enable
Schedule1	By Date	Time	11:01	FEB 24,2022	None	<input checked="" type="checkbox"/>
Schedule2	Monthly	Sunrise+	11:01	Day : 03	None	<input checked="" type="checkbox"/>
Schedule3	Weekly	Rando...	16:19	M T W T F S S	None	<input checked="" type="checkbox"/>
Schedule4	Yearly	Time	01:00			
Schedule5	Daily	Time	01:16			
Schedule6	Every	Time	00:00:05			

Days of the week x

Mon	Tue	Wed	Thu
Fri	Sat	Sun	All

Parameter	Description
<b>Occurrence</b>	Information on the type of schedule and time of execution
<b>Condition</b>	Condition applied to action execution
<b>Enable</b>	Enables/disables the execution of the scheduled actions without deleting the schedule.

See ["Recurring schedule" on page 343](#) for details on schedule parameters.



# 27 21 CFR Part 11 Compliance

PB610 Panel Builder 600 includes a set of functions for responding to the requirements specified in FDA 21 CFR Part 11. The standard is intended to provide a solution for securely handling electronic records and electronic signatures in industrial applications.

The table lists all the requirements specified by the regulation and reports the functions available in PB610 Panel Builder 600 for compliance.



**FDA 21 CFR Part 11 compliance is optional during application development and the application developer is responsible to configure the application in the proper way.**

Chapter	Description	PB610 Panel Builder 600 compliance level
11.10(a)	(a) Validation of systems to ensure accuracy, reliability, consistent intended performance, and the ability to discern invalid or altered records.	<p>Reports generated by PB610 Panel Builder 600 can be signed using x.509 Certificates. A certificate that includes the public key, necessary to verify the signature of reports, will be exported with the report.</p> <p>References:</p> <ul style="list-style-type: none"> <li>• <a href="#">"SaveEventArchive" on page 227</a></li> <li>• <a href="#">"PrintGraphicReport" on page 211</a></li> </ul>
11.10(b)	The ability to generate accurate and complete copies of records in both human readable and electronic form suitable for inspection, review, and copying by the agency. Persons should contact the agency if there are any questions regarding the ability of the agency to perform such review and copying of the electronic records.	<p>Application developer can select the resources (process values, alarms, etc.) whose changes will be tracked to the audit trail. Each change of the selected resources will be recorded with the name of the operator doing the change. The audit trail reports can be exported to .csv or .pdf files.</p> <p>References:</p> <ul style="list-style-type: none"> <li>• <a href="#">"Enable/disable audit trail" on page 376</a></li> <li>• <a href="#">"Exporting audit trail as .csv files" on page 381</a></li> <li>• <a href="#">"SaveEventArchive" on page 227</a></li> <li>• <a href="#">"Printing audit table" on page 380</a></li> <li>• <a href="#">"PrintGraphicReport" on page 211</a></li> </ul>
11.10(c)	Protection of records to enable their accurate and ready retrieval throughout the records retention period.	<p>Applications can be developed to self-generate signed reports to external memory or network folders at predefined interval (e.g. at the end of the day) or when circular buffer is full. User is responsible to keep these reports saved for the retention period.</p> <p>References:</p>



Chapter	Description	PB610 Panel Builder 600 compliance level
		<ul style="list-style-type: none"> <li>• <a href="#">"SaveEventArchive" on page 227</a></li> <li>• <a href="#">"PrintGraphicReport" on page 211</a></li> <li>• <a href="#">"Scheduler" on page 341</a></li> </ul>
11.10(d)	Limiting system access to authorized individuals.	<p>Application developer is responsible for the appropriate security configuration of the application.</p> <p>References:</p> <ul style="list-style-type: none"> <li>• <a href="#">"User management and passwords" on page 363</a></li> </ul>
11.10(e)	Use of secure, computer-generated, time-stamped audit trails to independently record the date and time of operator entries and actions that create, modify, or delete electronic records. Record changes shall not obscure previously recorded information. Such audit trail documentation shall be retained for a period at least as long as that required for the subject electronic records and shall be available for agency review and copying.	<p>Audit trail records are stored using a circular buffer (this is to ensure that the device will not run out of memory). Audit trails cannot be modified by the operator. Each record contains a sequential number to easily check the presence of all records. The application can be developed to save/export a copy of the data at regular intervals (e.g. at the end of each day); operator is responsible for storing copy of reports in a safe place.</p> <p>References:</p> <ul style="list-style-type: none"> <li>• <a href="#">"Exporting audit trail as .csv files" on page 381</a></li> <li>• <a href="#">"SaveEventArchive" on page 227</a></li> <li>• <a href="#">"Printing audit table" on page 380</a></li> <li>• <a href="#">"PrintGraphicReport" on page 211</a></li> <li>• <a href="#">"Scheduler" on page 341</a></li> </ul>
11.10(f)	Use of operational system checks to enforce permitted sequencing of steps and events, as appropriate.	Macros or JavaScript can be used to configure command sequences in the application.
11.10(g)	Use of authority checks to ensure that only authorized individuals can use the system, electronically sign a record, access the operation or computer system input or output device, alter a record, or perform the operation at hand.	<p>The HMI application can be configured</p> <ul style="list-style-type: none"> <li>• to be accessible only after user sign in with its own password</li> <li>• objects can be configured to be available or not available depending on the user who logged in to the system</li> <li>• resources can be configured to require a password confirmation before be modified</li> </ul> <p>References:</p> <ul style="list-style-type: none"> <li>• <a href="#">"User management and passwords" on page 363</a></li> </ul>

Chapter	Description	PB610 Panel Builder 600 compliance level
		<ul style="list-style-type: none"> <li>• <a href="#">"Electronic Signature" on page 377</a></li> </ul>
11.10(h)	Use of device (e.g., terminal) checks to determine, as appropriate, the validity of the source of data input or operational instruction.	<p>Resources can be configured to be accessible only from selected user groups. List of allowed IP address can be configured from the User Management settings.</p> <p>References:</p> <ul style="list-style-type: none"> <li>• <a href="#">"Modifying access permissions" on page 365</a></li> </ul>
11.10(i)	Determination that persons who develop, maintain, or use electronic record/electronic signature systems have the education, training, and experience to perform their assigned tasks.	Application developer is responsible to define and assign the appropriate user rights to each user that have access at the HMI device
11.10(j)	The establishment of, and adherence to, written policies that hold individuals accountable and responsible for actions initiated under their electronic signatures, in order to deter record and signature falsification.	Application developer is responsible for establishing appropriate procedures.
11.10(k)	<p>Use of appropriate controls over systems documentation including:</p> <p>(1) Adequate controls over the distribution of, access to, and use of documentation for system operation and maintenance.</p> <p>(2) Revision and change control procedures to maintain an audit trail that documents time-sequenced development and modification of systems documentation.</p>	Application developer is responsible for establishing appropriate procedures.
11.30	Persons who use open systems to create, modify, maintain, or transmit electronic records shall employ procedures and controls designed to ensure the authenticity, integrity, and, as appropriate, the confidentiality of electronic records from the point of their creation to the point of their receipt. Such procedures and controls shall include those identified in 11.10, as appropriate, and additional measures such as document encryption and use of appropriate digital signature standards to ensure, as necessary under the circumstances, record authenticity, integrity, and confidentiality.	PB610 Panel Builder 600 has been designed for operation in closed systems.
11.50(a)	<p>Signed electronic records shall contain information associated with the signing that clearly indicates all of the following:</p> <p>(1) The printed name of the signer;</p>	<p>All records will be added to the audit trail with time stamp and user id of logged user.</p> <p>References:</p>

Chapter	Description	PB610 Panel Builder 600 compliance level
	<p>(2) The date and time when the signature was executed; and</p> <p>(3) The meaning (such as review, approval, responsibility, or authorship) associated with the signature.</p>	<ul style="list-style-type: none"> <li>• <a href="#">"Exporting audit trail as .csv files" on page 381</a></li> <li>• <a href="#">"Table audit widget" on page 379</a></li> </ul>
11.50(b)	The items identified in paragraphs (a)(1), (a)(2), and (a)(3) of this section shall be subject to the same controls as for electronic records and shall be included as part of any human readable form of the electronic record (such as electronic display or printout).	
11.70	Electronic signatures and handwritten signatures executed to electronic records shall be linked to their respective electronic records to ensure that the signatures cannot be excised, copied, or otherwise transferred to falsify an electronic record by ordinary means.	Application developer is responsible for avoiding using the macros that permit the import/export of user passwords.
11.100(a)	Each electronic signature shall be unique to one individual and shall not be reused by, or reassigned to, anyone else.	System will ensure that two users with the same id cannot be defined. It is user responsibility to avoid removal and reassignment of the same user id to a different user.
11.100(b)	Before an organization establishes, assigns, certifies, or otherwise sanctions an individual's electronic signature, or any element of such electronic signature, the organization shall verify the identity of the individual.	User responsibility.
11.100(c)	<p>Persons using electronic signatures shall, prior to or at the time of such use, certify to the agency that the electronic signatures in their system, used on or after August 20, 1997, are intended to be the legally binding equivalent of traditional handwritten signatures.</p> <p>(1) The certification shall be submitted in paper form and signed with a traditional handwritten signature, to the Office of Regional Operations (HFC-100), 5600 Fishers Lane, Rockville, MD 20857.</p> <p>(2) Persons using electronic signatures shall, upon agency request, provide additional certification or testimony that a specific electronic signature is the legally binding equivalent of the signer's handwritten signature.</p>	User responsibility.
11.200(a)	(a) Electronic signatures that are not based upon	PB610 Panel Builder 600 Security functions are

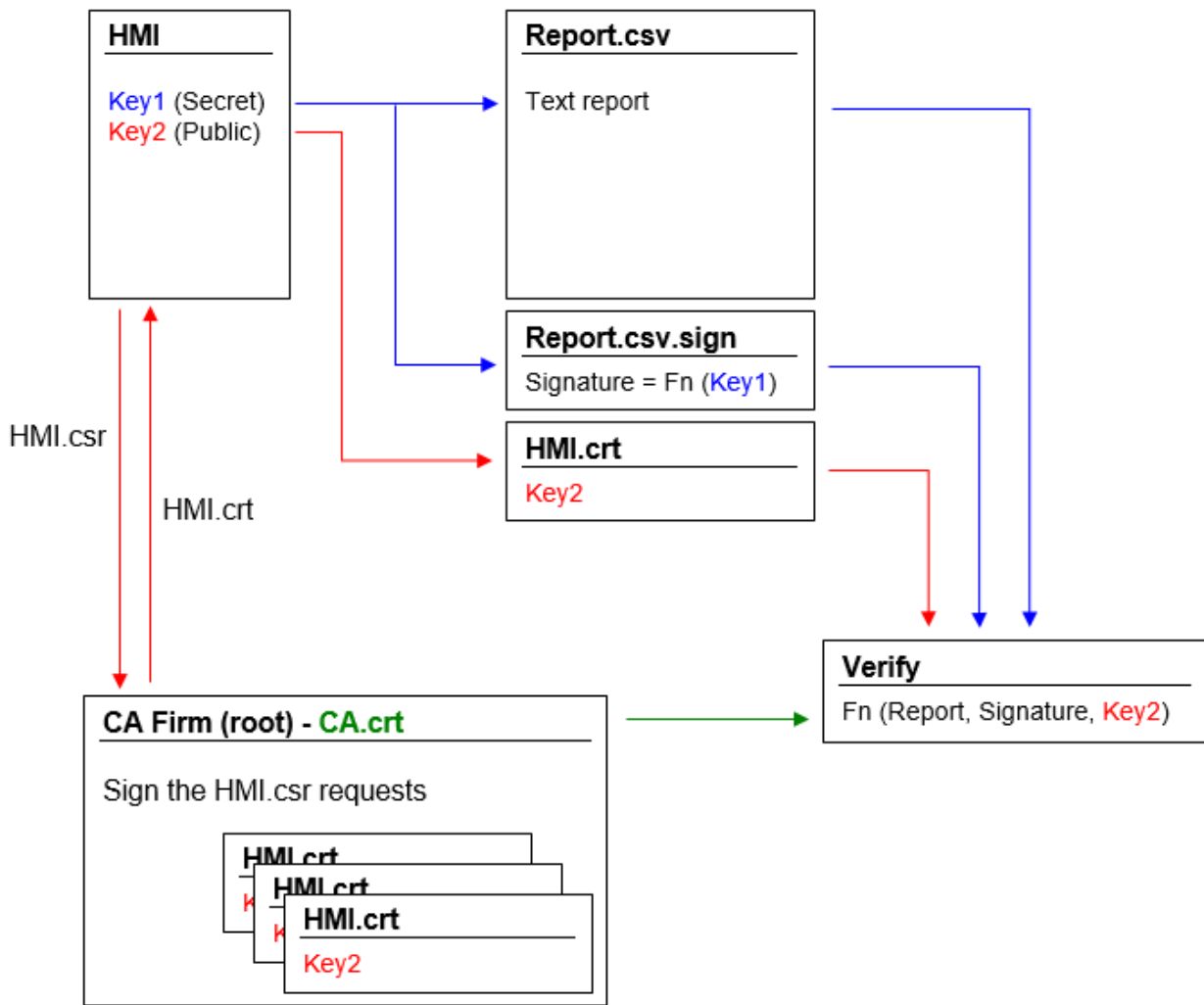
Chapter	Description	PB610 Panel Builder 600 compliance level
	biometrics shall: (1) Employ at least two distinct identification components such as an identification code and password.	based on the combination Username/ Password.
	(i) When an individual executes a series of signings during a single, continuous period of controlled system access, the first signing shall be executed using all electronic signature components; subsequent signings shall be executed using at least one electronic signature component that is only executable by, and designed to be used only by, the individual.  (ii) When an individual executes one or more signings not performed during a single, continuous period of controlled system access, each signing shall be executed using all of the electronic signature components.	Users must enter name and password to access the system. Critical actions can be configured to require entering again the password before execution is started.  References: <ul style="list-style-type: none"> <li>• <a href="#">"User management and passwords" on page 363</a></li> <li>• <a href="#">"Electronic Signature" on page 377</a></li> </ul>
	(2) Be used only by their genuine owners; and  (3) Be administered and executed to ensure that attempted use of an individual's electronic signature by anyone other than its genuine owner requires collaboration of two or more individuals.	Each user is responsible to not divulge own password. Passwords defined by administrator for first access can be forced to be redefined at first use.  References: <ul style="list-style-type: none"> <li>• <a href="#">"Configuring users" on page 371</a></li> </ul>
11.200(b)	Electronic signatures based upon biometrics shall be designed to ensure that they cannot be used by anyone other than their genuine owners.	PB610 Panel Builder 600 does not support biometrics.
11.300(a)	Maintaining the uniqueness of each combined identification code and password, such that no two individuals have the same combination of identification code and password.	It is not possible to define to define two users with the same User ID
11.300(b)	Ensuring that identification code and password issuances are periodically checked, recalled, or revised (e.g., to cover such events as password aging).	System can be configured to force each users to define a new and different password after a configurable number of days  References: <ul style="list-style-type: none"> <li>• <a href="#">"Configuring users" on page 371</a></li> </ul>
11.300(c)	Following loss management procedures to electronically deauthorize lost, stolen, missing, or otherwise potentially compromised tokens, cards, and other devices that bear or generate identification code or password information, and to issue temporary or permanent replacements using	Users can change their password at any time. Administration can redefine each user's password and force them to redefine at the first login.  References: <ul style="list-style-type: none"> <li>• <a href="#">"User management actions" on page 236</a></li> </ul>

Chapter	Description	PB610 Panel Builder 600 compliance level
	suitable, rigorous controls.	<ul style="list-style-type: none"> <li>"Configuring users" on page 371</li> </ul>
11.300(d)	Use of transaction safeguards to prevent unauthorized use of passwords and/or identification codes, and to detect and report in an immediate and urgent manner any attempts at their unauthorized use to the system security unit, and, as appropriate, to organizational management.	Failed logging attempts are logged to audit trail.
11.300(e)	Initial and periodic testing of devices, such as tokens or cards, that bear or generate identification code or password information to ensure that they function properly and have not been altered in an unauthorized manner.	User is responsible for ensuring appropriate measures.

## x.509 Certificate

To ensure authenticity of reports generated by HMI devices, HMI Runtime can generate reports with signed files to verify the authenticity and the integrity of the generated reports.

HMI Runtime uses asymmetric cryptography keys to sign files and x.509 standard to manage public key certificates. The picture shows the architecture.



The public key can be signed by a Certificate Authority (CA) that guarantees its authenticity.

## Workflow

1. Each HMI device contains two keys:
  - Key1 is the secret key, that is used to sign the reports generated by the HMI device. This key is securely stored inside the HMI device.
  - Key2 is the public key that anyone can use to verify the authenticity of the reports signed by the HMI device.
2. The macros "SaveEventArchive" or "PrintGraficReport" can be used to generate signed reports (see "SaveEventArchive" on page 227 or "PrintGraphicReport" on page 211 for additional details)
3. For the .csv file, you can use the public key and the signed file to verify the report is authentic and not tampered. (See "Signed CSV files" on page 358)
4. For the .pdf file, you can use a PDF reader to verify the report is authentic and not tampered. (See "Signed PDF files" on page 359)

## The internal x.509 certificate files

Each HMI devices already have a self-signed certificate. You are free to use it, ask a Certificate Authority to sign it, create a new one using the information that you prefer or to upload and use your own certificate. All operations are available from the device "System Settings" (see the x.509 Certificate section inside the "System Settings" on page 622).



Note that you can never retrieve the private key from the HMI device. You can instead provide a certificate with both private and public keys.

### Use the self-signed certificate

To use the self-signed certificate you have to do nothing. Simply, use the macros that generate signed reports. Even if the certificate will be provided from the macros, you can use the "System settings" to retrieve your copy of the certificate (just to be sure of the originality of the certificate).

### Use a certificate signed from a Certificate Authority

To use your signed HMI certificate from a certificate authority you must download the certificate signing request file from the "System settings" panel. Sending and asking a certificate authority to sign the certificate (generally this is a pay operation) and then upload the signed certificate to the HMI device.



**After retrieving the "certificate signed request" file to send to the certificate authority, be sure to never regenerate a new certificate otherwise the internal private key associated with the certificate send to the authority will be lost.**

### Use your own certificate

If you have your own Certificate and you like to use it, you can upload it inside the HMI device from the "System Settings" panel. Note that you must provide both private and public keys.



**When the certificate contains a private key, the current private key will be substituted with the key found in the certificate and it will not be possible to recover it.**

Example of a certificate with both public and private keys (certificates are encoded base64).



```

ssl-certificate.crt X
1 -----BEGIN CERTIFICATE-----
2 MIIDBDCCAewCCQDeBYW7PYwJsDANBgkqhkiG9w0BAQsFADBEMQswCQYDVQGEwJJ
3 VDEPMA0GA1UEBwwGVmVyb25hMRMwEQYDVQKDApUZXN0T2ZmaWNlMQ8wDQYDVQ
4 DAZITUktMDQwHhcNMTcwNjI2MDgwOTQ1WhcNMTgwNjI2MDgwOTQ1WjBEMQswCQYD
5 VQGEwJJVDEPMA0GA1UEBwwGVmVyb25hMRMwEQYDVQKDApUZXN0T2ZmaWNlMQ8w
6 DQYDVQDDAZITUktMDQwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCd
7 Nlp2kswcbLh4IxS6eeCgQ4EAUHC Rpaz5YPfQ/un9/s0tejaa3S13Pcqv3JqddJM8
8 mJEZaPF/+HhAEhtC+rv57TbgullUQJdoQpfoGChofpULforXZt2BfdWNx67plNoa
9 YM3ElaNtAKIW2o6S9HGEvlkfo9XFLGkFgeMgC59+SejggucNT0m99m6fNa5910I7
10 UDJFinkC3bxtONj+WiL/ieZYkHXacaN9q06fx+2NfmiSsXGfPnmSys5mocqo89tMa
11 TjyeF7jYpDccCpJ9pY4xRjRpcIkDCM7PabVoG/ascSMUUEXPE2R0W4UJ6bPAyGd6
12 QLKCCq0BUie/eUj0pnanAgMBAEwDQYJKoZIhvcNAQELBQADggEBAMLfIEXQOEjS
13 OpwVzNzXmL/A6PLU5BK1hVYHb7ofb2Z37zN69vCn8ESglAFYK7QhkhJu3zAD+jH
14 fYBVKvdxfd3HS8EmcDWxpC6F2lfgqsSqepMRTbKbsSa053a7JsXtwnHVNfG6EBZV
15 8tqS1Gc4RwtJeVZJelUdmWSBD4Fc7asFeBCKqLrHJinz7buj3I4fLcyscTaMTBI9
16 fsE7poEpWvKc7NWtKYZglGG3AG6xONu3sEahcJ5k+UVdh/QQdAiCt3vG+JJ/owYU
17 sd30WI24pNzG/GUH9MbJyvI4ftA8IvEhGxHvi3xt7s1JnvYQDaghOEDhdtGvilOr
18 nJZ2FZOBCEI=
19 -----END CERTIFICATE-----
20 -----BEGIN RSA PRIVATE KEY-----
21 MIIIEpAIBAAKCAQEA3D2adpLMHGy4eCMUunngoEOBAFBwkaWmeWD30P7p/f7NLXo2
22 mt0otz3Kr9yanXSTPjIRGwJxf/h4QBIbQvq7+e024LtZVECXaEKX6BgoaH6VC36K
23 l2bdgX3VjceudTaGmDNxJWjbQCiFtqOkvRxlL5ZH9PVxSxpBYHjIAuffkno4ILg
24 jU9Jvf2unzWuf2dCOlAyRSDZAt28bTjY/loi/4hGWJB12nGjfatOn8ftjX5okrFx
25 j55ksrOZqHKqPPbTgk48nhe42KQ3HAqSfaWOMUY0aXCJAWjOz2mlaBv2rHEjFFO1
26 zxNkdFuFCemzMoA+kCygqgtAVIuv3lI9KZ2pwIDAQABAoIBAGnamsuqrwDu5hGh
27 02H8GHUpvd/3ytIISujHyvgkwTf+FoII32y9uMe0pUy5/3y2v9v9/qm3P3djaFJq
28 gb5Fprxx4dJPKJZaYi2U7U585lesmVqoHneCk/GeGlyH4zWlwo2xgNgBkkhgaIoR
29 zz0m0bachVz+SCD6wUJpbMOW0FBw54oPL0XS/gD+76S9ET7xmQZAS5xV/w8Khnt
30 PtjPFT58GKhqVIC9cMrrBrkuQQPrNrDaJMPsQDxrFp7POQm4+GivrUJ0FA9Vtx46
31 C5QhXqVps/BODo3mjeOcj2b/FqsvG7WCc5PWOAcCqStmDxl+DQZOIVFSTrE4kdpq
32 mNn/80kCgYEA88Xfmqg0ta83lpe9b6U0BaLvvs1gxgXmCmkYvK7Ru+iKyPUMzxB+
33 BjGWeeiZuigmIhXfFu3eBs5xOgDrUxf9j55sJAFamljG4LTyun378RnOdA87fflq
34 rpF4oPKVfTrfXXz2keIq0eX2tD6Lsn3+MJwYqpefovXmyJA3kPgcGv0CgYEA50H0
35 HQififZ22nApGpF/jJpU7hBLC45cSXvE2MX2I3rd3ptGwzKR0/l2ks1bvQutqRln
36 slyEF+c9LCz6g7FYhJoewChLqCVfeZ9GxBzHeJloxZwmxDX18L4vmEDphwlcV8b3
37 ExHqUIMGuINHGelPIR1LKeEsbTQU+OVHuNv443MCgYEA7rMKYh1lC6bYCsjowSMG
38 TqKembX84cqyl+zstp+EVbi99Usm0Lc4f/4cd6EQrp1Twbqi6YPgDdAmRQLTALKp
39 e3FIOFVub4aQr0XgDEcC5bI8W57yxUrZJLjjYs5HHQoB4Dw5m0TOMFnS+enoxs3i
40 kly3Nowjz+fRCYFWN8kLVE0CgYEA43CLLk7ZcW9XKa2cNB00PE1g8A4YMJJfk2nl
41 zKjNj1F9ujyO2NV4RYOsI+RSsFe3ARDJcS6xP200Tc8ixrh57VhCnAxFdGblQpFy
42 oNgJGkf9zjPoMJsqykjSOHTG+CctqaqmPxxkkLScbIW4PPSn/U6KDPNHpVNOuQeO
43 hXHak58CgYBLW1719vgYhUiSWc9Gd3mCSxpAb6y8RcyTgqF76K8v4MallPqFkEtD
44 0BaFt1A+PtMLk2ODTRH4XU18oc9eV+7VDFkPJ8T0A2VwjzjMgNad+vKlM4nOEBTt
45 UhegY0k8yLxS12vuiVnHvKBIoF/G2ckwrxjO9KVE+SA45Ex0Px5qQ==
46 -----END RSA PRIVATE KEY-----

```



You can import inside each HMI device the same certificate file to have a unique public certificate file for all your HMI devices.

### PB610 PC Runtime

When using PB610 PC Runtime the certificate files can be found inside the folder:  
%AppData%\ABB\



## Signed CSV files

Reports generated in CSV format using the **SaveEventArchive** macro can be signed using the x.509 certificate included inside the HMI device. The signature makes sure that nobody tampered with the content of the document since it was signed.

See also:

- The SaveEventArchive parameters ("[SaveEventArchive](#)" on page 227)
- How to provide an x.509 Certificate to Linux devices ("[x.509 Certificate](#)" on page 637)
- How to provide an x.509 Certificate to WinCE devices ("[System Settings](#)" on page 647)

When required, using Signed=True, the SaveEventArchive macro in addition of the [ReportName].csv generate other two files:

- [ReportName].csv.sign
- ssl-[CertificateName].crt

Where the [ReportName].csv.sign is the signature of the report and the ssl-[CertificateName].crt is a copy of the x.509 certificate of the HMI device. Note that you can retrieve the certificate of the HMI device even from the System Setting of the HMI device.

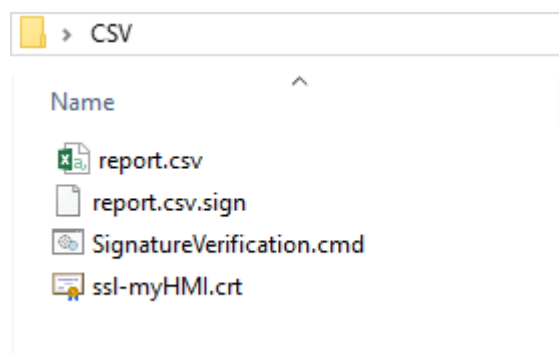
### How to verify the report's signature using the public OpenSSL library

To verify that nobody has tampered the content of the report you need

- be sure the ssl-[CertificateName].crt is coming from the HMI device
- use a tool to verify the signature (e.g. OpenSSL-Win32)

Reference.: <https://www.openssl.org/>

To verify that the .csv report generate from HMI device has not tampered you can install a public OpenSSL library, copy all files generated from the macro inside the same folder and use the below batch file



File: *SignatureVerification.cmd*

```
@echo off
set OpenSSL="C:\Program Files (x86)\OpenSSL-Win32\bin\openssl.exe"
set FileToCheck=Report.csv
set hmiCertificate=ssl-myHMI.crt

rem Extract public key from the certificate
%OpenSSL% x509 -in %hmiCertificate% -pubkey -noout > publicKey.pem
```

```

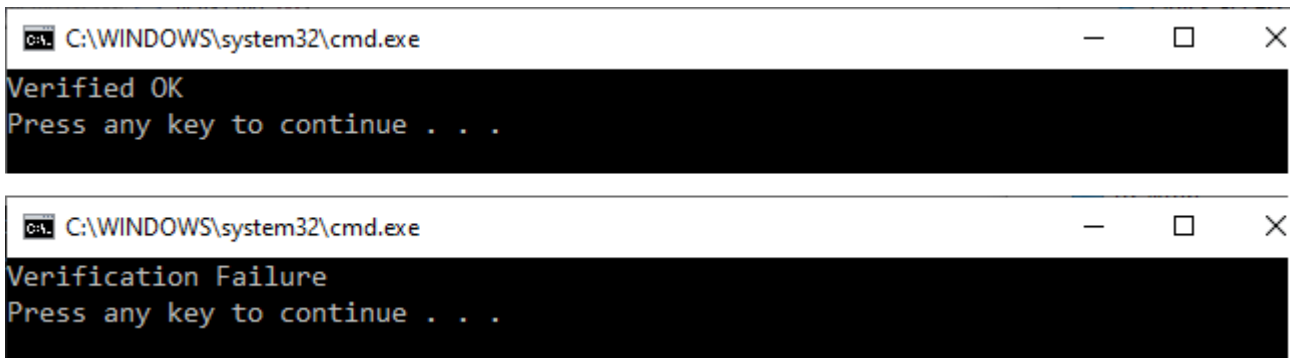
rem Verify Signature
%OpenSSL% dgst -sha256 -verify publicKey.pem -signature %FileToCheck%.sign
%FileToCheck%

rem Remove public key
del publicKey.pem

pause

```

The below pictures are showing the possible outputs of the batch file



**On Linux devices, the BSP v1.0.239 or greater is required**  
**On WinCE devices, the BSP v2.29 or greater is required**

## Signed PDF files

Reports generated in PDF format using the **PrintGraphicReport** macro can be signed using the x.509 certificate included inside the HMI device. The signature makes sure that nobody tampered with the content of the document since it was signed.

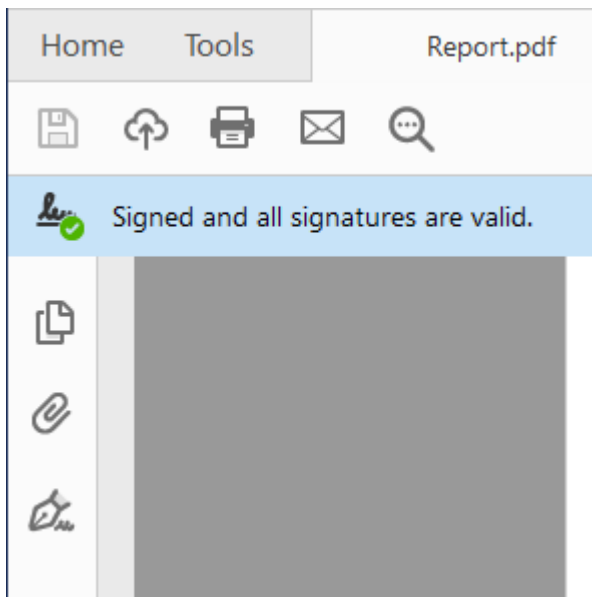
See also:

- The PrintGraphicReport parameters ("[PrintGraphicReport](#)" on page 211)
- How to provide an x.509 Certificate to Linux devices ("[x.509 Certificate](#)" on page 637)
- How to provide an x.509 Certificate to WinCE devices ("[System Settings](#)" on page 647)

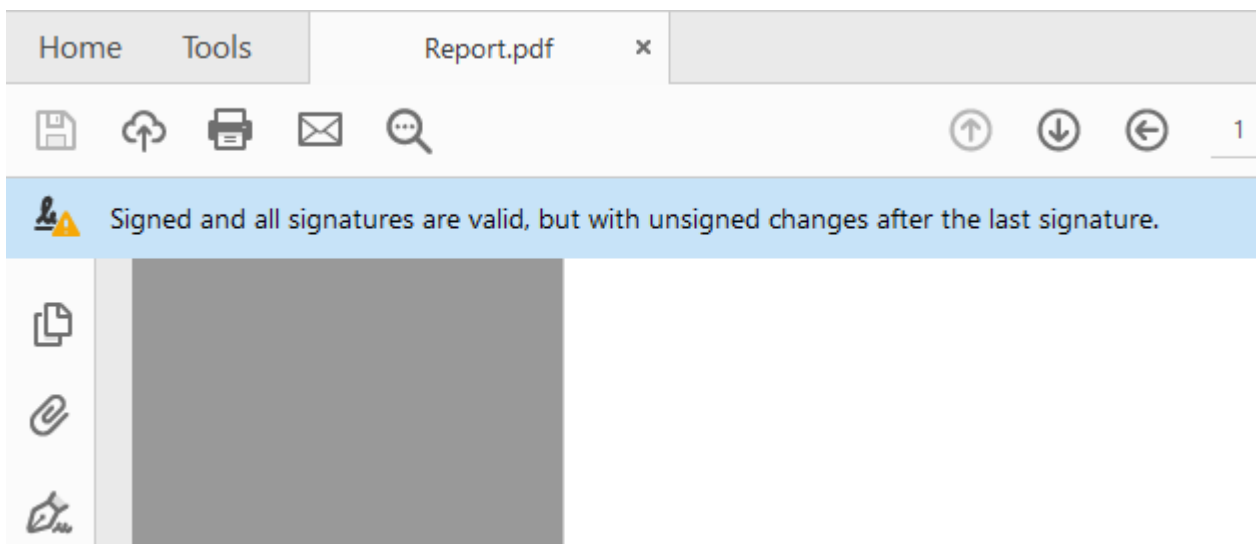
When you open the file, the PDF reader tries to decide if the signature is valid then it looks at the certificate used to sign the document.

### x.509 certificate signed from a Certificate Authority

If you have uploaded to the operator panel a valid x.509 certificate, signed by a Certification Authority, when you open the generated PDF file you will get a message that highlights the document is valid.



If the document has been modified, it will be highlighted with a different message.



### Certificate Trust and Authenticity

Trust of signed certificates depends on the issuer of the certificate. The PDF reader will trust a certificate if you have told it to trust the issuer of that particular certificate. By default the Adobe Reader only trust certificates issued by Adobe or one of their partners. This means that it will show a warning if the certificate wasn't issued by one of these authorities. Microsoft Windows also uses certificates for validating software vendors and content providers. You can configure your Adobe Reader to trust these issuers in addition to the Adobe partners.

Check inside the preferences of the PDF reader if you want to enable the PDF reader to use even the Microsoft Windows certificates

**Windows Integration**  
 Trust ALL root certificates in the Windows Certificate Store for:

- Validating Signatures
- Validating Certified Documents

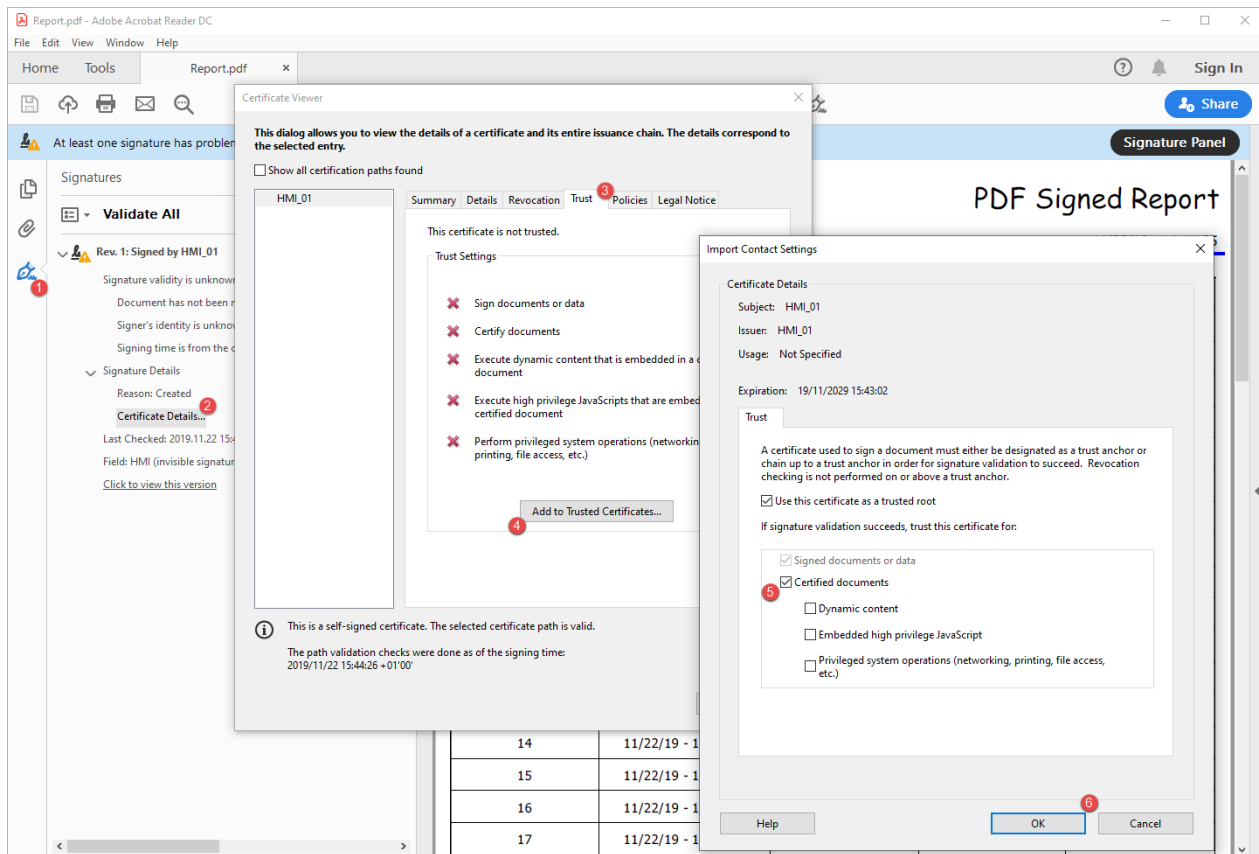
Selecting either of these options may result in arbitrary material being treated as trusted content. Take care before enabling these features.

### x.509 self signed certificate

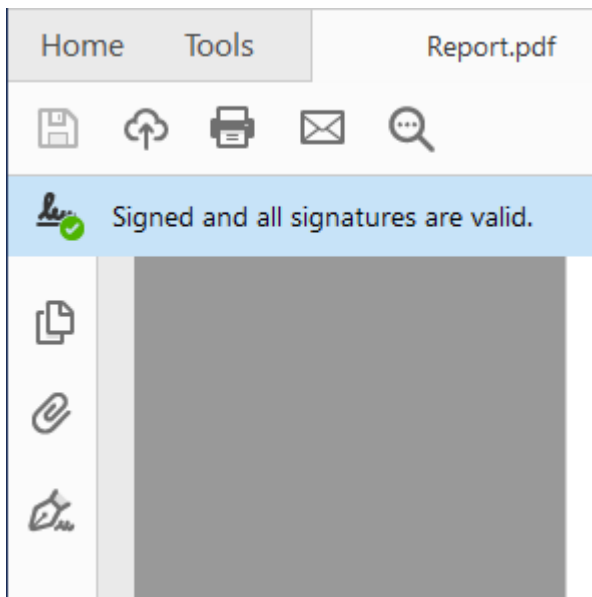
A self-signed certificate is a certificate that is not signed by a certificate authority (CA).

This means that PDF Reader can confirm the file is signed and not tampered, but cannot confirm the signature (alias the certificate) is authentic. Is the user have to take care to verify the certificate is authentic (for example, making sure that the document was actually produced by the panel) and confirm to the PDF reader that the certificate included in the document is valid and that can be considerate valid even for the next reports.

Steps to manual confirm that the certificate is authentic:



Now, if you close and reopen the PDF document you will get the valid signature. Moreover, even all other documents produced from the same HMI device will be shown with the correct signature because the information that the certificate is authentic has been stored inside settings of the PDF Reader.



**On Linux devices, the BSP v1.0.507 or greater is required**  
**On WinCE devices, the BSP v2.31 or greater is required**

## Compliant applications

Suggestions to development a CFR11 compliant applications

### User management macros

User management macros that could be use from any user

- Login
- Logout
- SwitchUser
- ChangePassword

User management macros that could be used from administrator only

- ResetPassword
- AddUser
- EditUsers
- ExportUsers,

Deprecated macros that must not be used inside CFR 21 part 11 compliance applications

- ImportUsers
- DeleteUser
- DeleteUMDynamicFiles,

# 28 User management and passwords

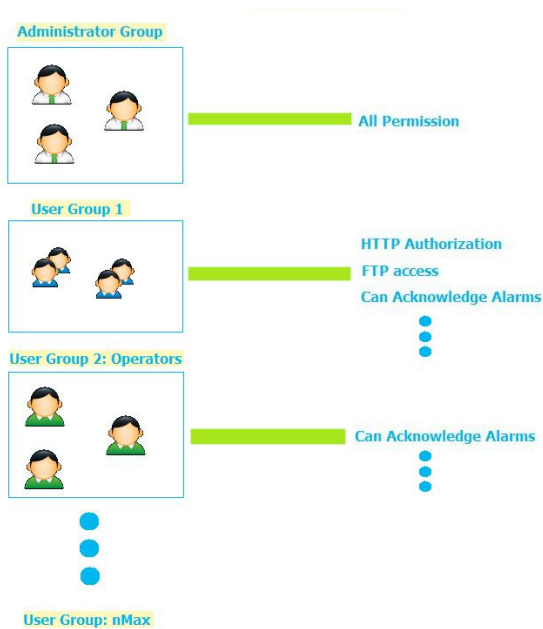
You can restrict access to various widgets and operations by configuring users, users groups and assigning specific authorizations to each group.

Each user must be member of one and only one group. Each group has specific authorizations and permissions.

Authorizations and permissions are divided in two categories:

- Widget permissions: hide, read only, full access
- Action permissions: allowed or not allowed.

By organizing permissions and groups you can define the security options of a project.

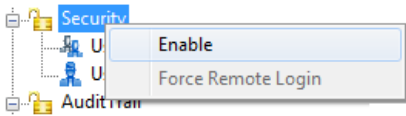


<b>Enable/disable security management</b> .....	<b>364</b>
<b>Configuring groups and authorizations</b> .....	<b>364</b>
<b>Modifying access permissions</b> .....	<b>365</b>
<b>Assigning widget permissions from page view</b> .....	<b>371</b>
<b>Configuring users</b> .....	<b>371</b>
<b>Default user</b> .....	<b>373</b>
<b>Managing users at runtime</b> .....	<b>373</b>
<b>Force remote login</b> .....	<b>374</b>

# Enable/disable security management

Path: **ProjectView**> right-click **Security**> **Enable**

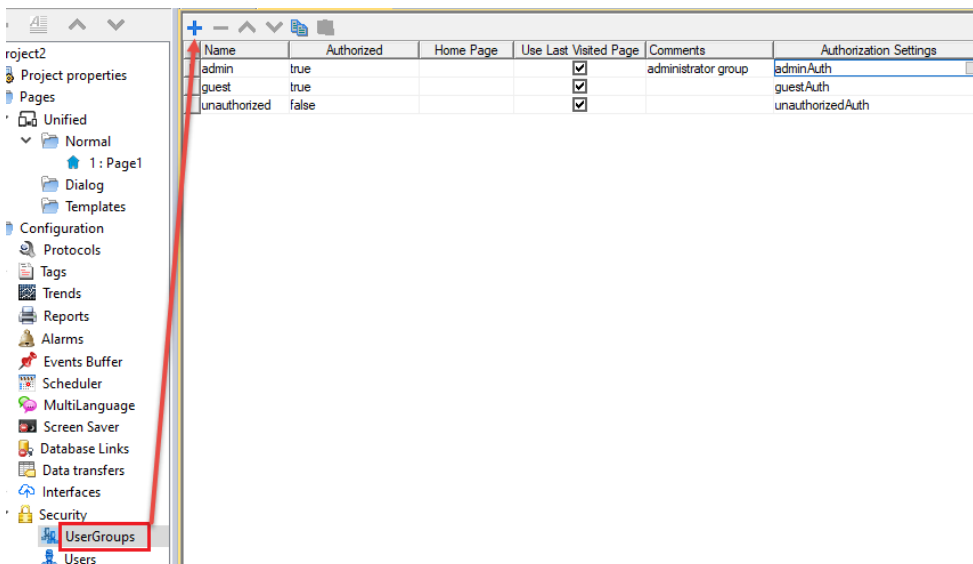
The padlock symbol indicates whether the function is enabled or disabled.



**Important:** Security settings are effective only if the security function is enabled.

## Configuring groups and authorizations

Path: **ProjectView**> **Security**> double-click **UserGroups**



Three predefined groups are available by default (**admin**, **guest** and **unauthorized**): they cannot be deleted nor renamed. You can, however, modify authorizations and other settings.

### Adding a user group

Click **+** to add user group.

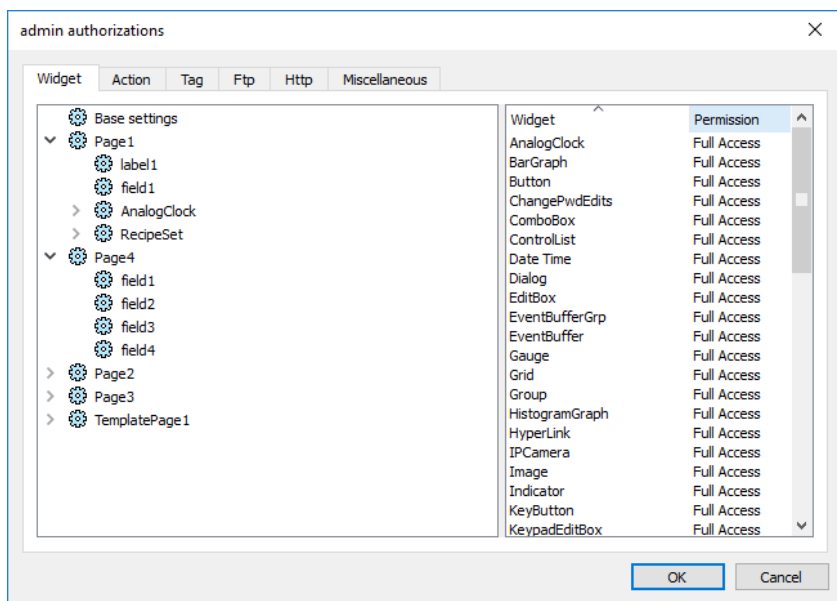
Parameter	Description
<b>Name</b>	Name of users group
<b>Authorized</b>	Authorization granted
<b>Home Page</b>	Page displayed when users belonging to this group log in
<b>Use Last Visited</b>	When selected, the last page displayed by the previous user will be displayed when users

Parameter	Description
Page	belonging to this group log in
Comments	Any comment or description for the group
Authorization Settings	Opens the Admin Authorization dialog to set access permissions. See " <a href="#">Modifying access permissions</a> " below for details.

## Modifying access permissions

Path: **ProjectView**> **Security**> double-click **UserGroups** > **Authorization Settings** column

Click the button: a dialog appears with a list of widgets and actions. You can modify access permissions for each one in the list.



### Widget permissions

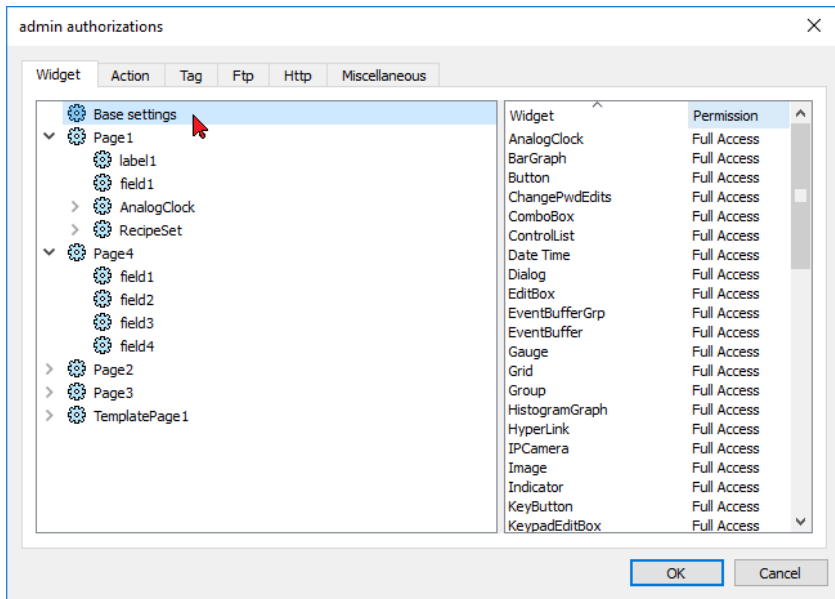
In the **Widget** tab you can define widget access options at project level, at page level or at widget level for all the widgets used in the project. Lower levels permission (for example, widget level) overrides higher levels (that is, page and project levels).

Use **Base settings** to set default permissions at project level.

Possible settings are:

- **Full Access** to enable read/write access to the widget
- **Read Only** to enable readonly access to the widget
- **Hide** to hide widget for selected group





## Changing a widget permission

To change access permission for an individual widget in a page of the project, navigate to that widget within its page on the right pane and customize its access options. Otherwise, all widgets take the permissions set at project or page level.

For example, if page permission for a widget is set at project level to **Read Only**, then all the same widgets will have permission **Read Only**. When you select a widget inside a page from the tree structure, permission is actually set to **Use Base Settings**. You can change this setting and modify access permissions only for this widget in this page.

## Access priority

Widget permissions are considered with the following priority:

Permission level	Priority
Project level - Basic settings	Low
Page level	Medium
Widget level	High

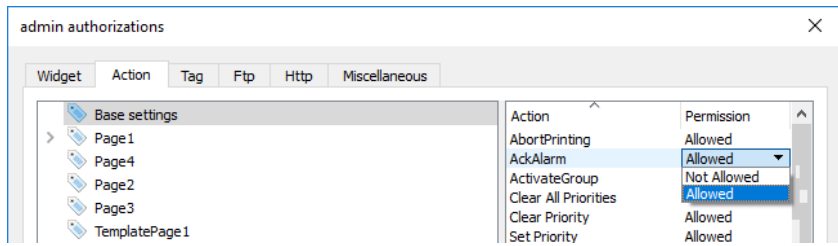
This allows you to specify exceptions for an action or a widget directly from the page view.

For example, if you set permissions for a widget at project level to Read Only and to Full Access at page level then the page level settings will prevail.

Access permissions can be modified directly from the project page. See ["Assigning widget permissions from page view" on page 371](#) for details.

## Action permissions

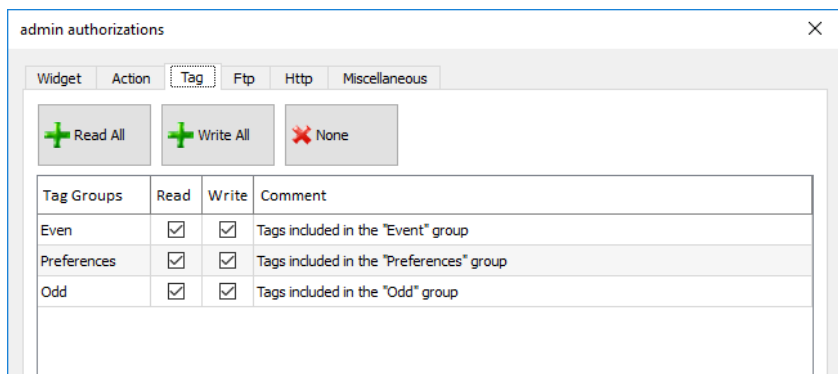
In the **Action** tab you can define action authorizations at project level, at page level or at widget level. Actions can be either **Allowed** or **Not Allowed**.



Action permissions can be modified directly from the project page. See ["Assigning widget permissions from page view" on page 371](#) for details.

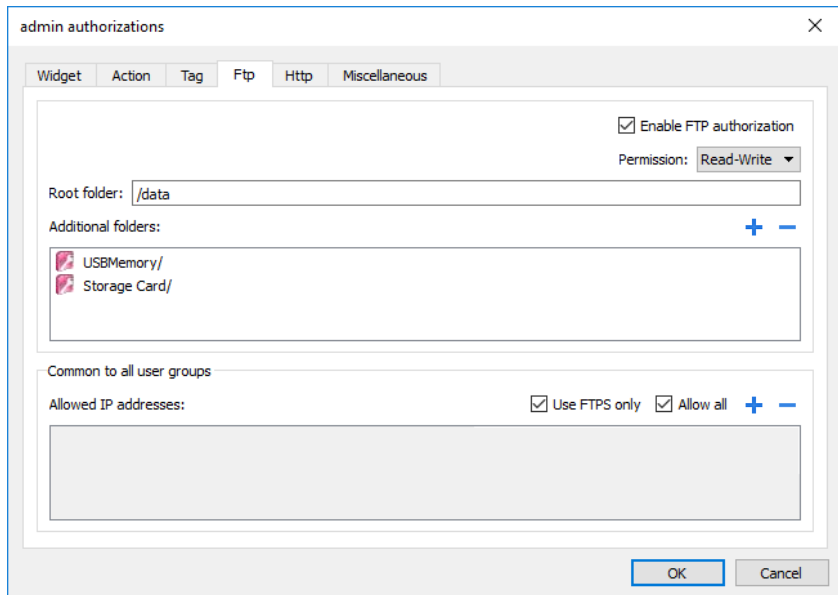
## Tag permissions


For each group of tags, you can define the Read/Write access rights



## FTP authorizations

In the **Ftp** tab you can set specific authorizations for the FTP server.



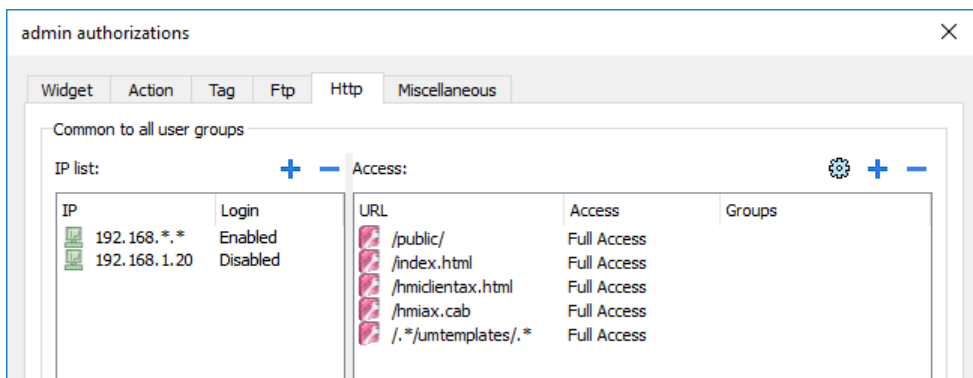
Element	Description
<b>Enable FTP authorization</b>	Enables the FTP function for the specific group
<b>Permission</b>	Type of permission: <ul style="list-style-type: none"> <li>• <b>Read-Only</b></li> <li>• <b>Read-Write</b></li> </ul>
<b>Root Folder</b>	Folder to be used as root for FTP access. This is a relative path.
<b>Additional folder</b>	Extra folders to be used as root for FTP access (for example, on USB drive or SD card)
<b>Allowed IP Addresses</b>	List of IP addresses from which FTP connection can be accepted.  <b>This setting is common to all users groups.</b>
<b>Use FTPS only</b>	You can disable this flag if you need to use an old FTP client that does not support encrypted FTP mode, but please note that this is not a secure connection and all your data (even your password) are sent in the clear over the Internet.


## HTTP authorizations

In the **HTTP** tab you set restrictions to HTTP access to the web server integrated in HMI Runtime.

Wildcards can be used to identify a range of IP addresses.

For example, the two following rules set the HMI device unit can only be accessed by all the IP addresses 192.168.\*.\* on your local network in which only the IP address of 192.168.1.20 can access the device without entering a login name.



Element	Description
<b>IP list</b>	IP addresses authorized to access the HTTP server.  By default the login is required from any IP address (IP=.*, Login=Enabled).
<b>Login</b>	When disabled, the username and password are not required.
<b>Access limits</b>	List of resources for which access is limited

Effect of these settings depends on whether the option **Force Remote Login** has been selected. See "[Force remote login](#)" on page 374 for details.

Force Remote Login	Default Access to workspace	Access limits
-	Full	-
<b>Disable</b>	Full	Can be used to block access to some files/folders or to require authorization
<b>Enable</b>	No Access	Can be used to open access to files/folders



**Important: This setting is common to all users groups.**

## Adding an HTTP configuration

To add and configure a new access click **+**: the **Access limits** dialog is displayed.

To restore the default configuration click the **Set default access limits** icon. Default configuration allows access to the following:

- PUBLIC folder and Index.html


## Miscellaneous settings

In the **Miscellaneous** tab you can define various authorization settings.

The screenshot shows the 'admin authorizations' dialog box with the 'Miscellaneous' tab selected. The settings are as follows:

- Common to all user groups:**
  - Number of users allowed to login: 4
  - Login Lock:**
    - Enable login lock on wrong password
      - Tries without lock: 3
      - Minimum timeout: 10 seconds
      - Maximum timeout: 100 seconds
- Permissions (all checked):**
  - Can enter config mode
  - Can load factory settings
  - Can zoom
  - Can see logs
  - Can create backups
  - Can access from web client
  - Can access from remote client
- Can manage other users:**
  - admin
  - guest
  - unauthorized

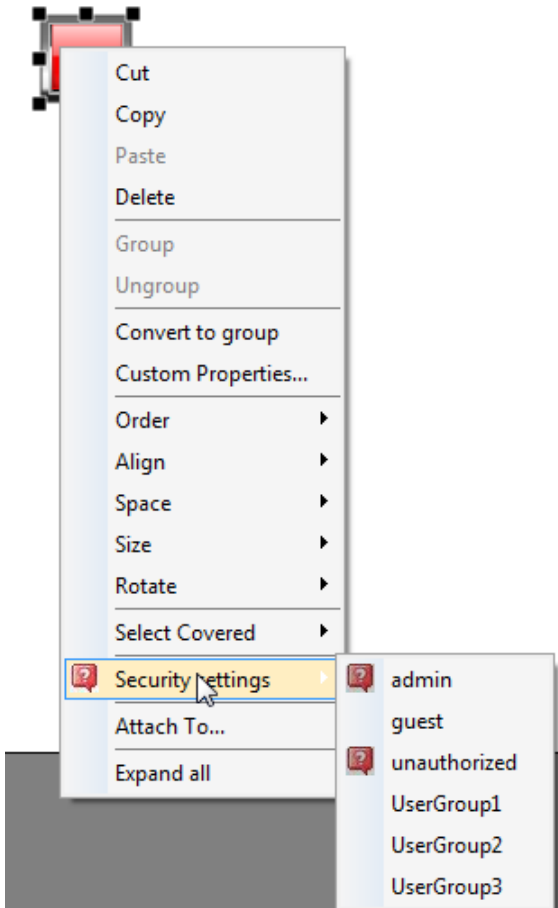
Buttons: OK, Cancel

Option	Description														
<b>Number of users allowed to login</b>	<p>Maximum number of users that can be connected to the HMI Runtime at the same time.</p> <p> <b>This setting is common to all users groups.</b></p>														
<b>Login Lock</b>	<p>Option to prevent “brute force” attacks.</p> <p>When "Enable access lock on bad password" is selected after the number of bad passwords allowed has been exceeded, the system will introduce a delay between one password and another in order to prevent a possible brute force attack. It is possible to define:</p> <ul style="list-style-type: none"> <li>• Tries without lock Number of incorrect passwords accepted before inserting a delay between passwords</li> <li>• Minimum/Maximum timeout The initial delay and the maximum delay will not be further increased.</li> </ul> <p>Timeout uses an exponential growth.</p> <p>Example of usage</p> <ul style="list-style-type: none"> <li>• Tries without lock: 3</li> <li>• Minimum timeout: 2</li> <li>• Maximum timeout: 10</li> </ul> <table data-bbox="619 1182 1129 1420"> <tr> <td>First 3 attempts</td> <td>no timeout</td> </tr> <tr> <td>Attempt 4</td> <td>2 seconds of timeout</td> </tr> <tr> <td>Attempt 5</td> <td>4 seconds of timeout</td> </tr> <tr> <td>Attempt 6</td> <td>8 seconds of timeout</td> </tr> <tr> <td>Attempt 7</td> <td>10 seconds of timeout</td> </tr> <tr> <td>Attempt 100</td> <td>10 seconds of timeout</td> </tr> <tr> <td>Attempt 200</td> <td>10 seconds of timeout</td> </tr> </table>	First 3 attempts	no timeout	Attempt 4	2 seconds of timeout	Attempt 5	4 seconds of timeout	Attempt 6	8 seconds of timeout	Attempt 7	10 seconds of timeout	Attempt 100	10 seconds of timeout	Attempt 200	10 seconds of timeout
First 3 attempts	no timeout														
Attempt 4	2 seconds of timeout														
Attempt 5	4 seconds of timeout														
Attempt 6	8 seconds of timeout														
Attempt 7	10 seconds of timeout														
Attempt 100	10 seconds of timeout														
Attempt 200	10 seconds of timeout														
<b>Can enter config mode</b>	Enables switching from runtime to configuration mode. Normally used for maintenance.														
<b>Can load factory settings</b>	Restores factory settings.														
<b>Can zoom</b>	Enables zoom in/out in context menu at runtime														
<b>Can see log</b>	Allows user to see logs at runtime														
<b>Can create backup</b>	Allows user to backup project.														
<b>Can access from web client</b>	Enables connecting from a web client														
<b>Can access from remote client</b>	Enables connecting from HMI Client														
<b>Can manage other users</b>	Gives super user privileges at runtime to manage the select groups. Allows adding, deleting and modifying users' permissions.														

## Assigning widget permissions from page view

You can assign different levels of security, to different user groups, on a single widget, directly from the project pages.

1. Right-click on the widget and select **Security settings**.
2. Choose the group: the authorization dialog for the group is displayed.
3. Set the security properties to access the widget.



See ["Modifying access permissions" on page 365](#) for details.




## Configuring users

*Path: **ProjectView** > **Security** > double-click **Users***

In the Users editor, click **+** to add a user: one row is added to the table.

The screenshot shows the 'Users' configuration window in ProjectView. The table lists the following users:

Name	Default User	Inactive	Group	Password	Comments	Exception	Change Initial Password	Logoff Time ( In minutes )	Password Minimum Length
admin	<input type="checkbox"/>	<input type="checkbox"/>	admin	*****	admin user	<input type="checkbox"/>	false	0	4
guest	<input checked="" type="checkbox"/>	<input type="checkbox"/>	guest	*****		<input checked="" type="checkbox"/>	false	0	4
user1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	admin	*****		<input type="checkbox"/>	false	0	4
user2	<input type="checkbox"/>	<input type="checkbox"/>	admin	*****		<input type="checkbox"/>	false	0	4
user3	<input type="checkbox"/>	<input type="checkbox"/>	admin	*****		<input type="checkbox"/>	false	0	4

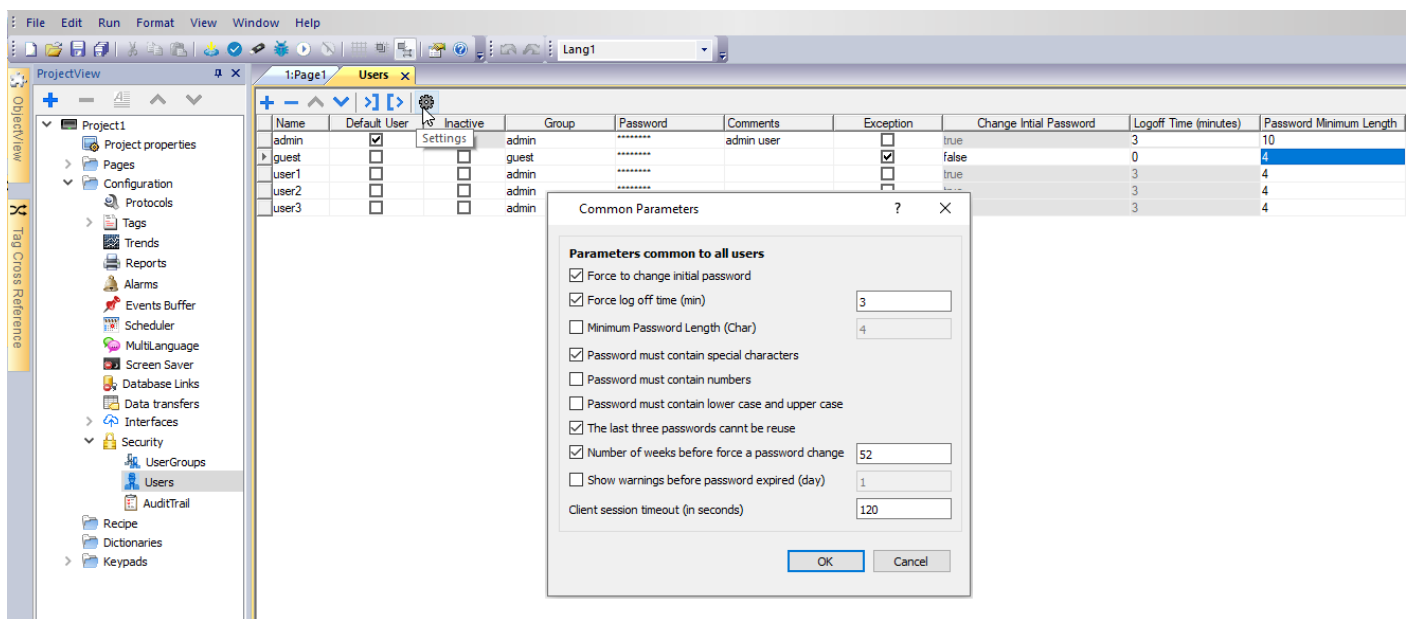
Parameter	Description
<b>Name</b>	User name
<b>Default User</b>	This user is automatically logged in when the system is started or after another user has logged off. Only one Default user can be set
<b>Inactive</b>	Inactive users will no longer be able to log in
<b>Group</b>	User group
<b>Password</b>	User password. Note that for security reasons the password will never be displayed  <b>Passwords are encrypted and cannot be retrieved not even for specialized technicians</b>  New project are create with "admin" default user. Password for this user is "admin". It is recommended to change this password when setting up User Management.
<b>Comments</b>	Further user description
<b>Exception</b>	Allows to change the values forced from the User Settings parameters
<b>Change Initial Password</b>	This user is forced to change his password at first log in.  This option is not supported in simulator.
<b>Logoff time (minutes)</b>	Minutes of inactivity after which the user is logged off. Set to 0 to disable
<b>Password minimum length</b>	Minimum length of password
<b>Must contain special characters</b>	Password must contain at least one special character
<b>Must contain numbers</b>	Password must contain at least one numeric digit
<b>Must contain lower case and upper case</b>	Password must contain lower case and upper case

Parameter	Description
Password cannot be reused	The new password must be different from the last 3 used passwords
Password aging (weeks)	Number of weeks before forcing a password change (1/52 weeks)
Warning (days)	Show a warning message before password expires (1/30 days)

## Users Settings

From the Settings command, there is the possibility to define parameters values that will be common to all users.

Users with the Exception flag checked are not force to use the common parameters.



## Default user

You can define only one default user in a project. This is the user automatically logged in at system start up and when the currently logged user logs out or is logged out after time-out.

To log into HMI Runtime with a different user, use one of the actions:

- **SwitchUser**
- **LogOut**

See "User management actions" on page 236 for details.

## Managing users at runtime

The default user, if any, is automatically logged in when the HMI Runtime is started. If no default user is configured, the system requires a user name and password. See "User management actions" on page 236 for details on the actions that



can be executed on users.

## Removing user data

All the user information modified at runtime is stored in dedicated files. To remove these dynamic files and all the changes applied to user configuration at runtime you can:

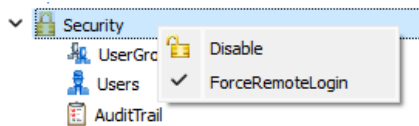
- on HMI Runtime: execute the action DeleteUMDynamicFile
- with PB610 Panel Builder 600: select the **Delete Dynamic Files** in the download dialog.



Note: When any modification is performed on user management in Studio, it is needed to delete User Management dynamic files to apply new User Management settings.

## Force remote login

Path: **ProjectView**> right-click **Security**> **ForceRemoteLogin**



Select this option to force user to log in when using remote access via HMI Client. If not selected, remote access will use the same level of protection of local access.



**Important: This function only works when user management is enabled.**



**WARNING: Use this option when you have a default user but at the same time you want to protect remote access.**

See "[Enable/disable security management](#)" on page 364 for details.

The only files/folders still accessible when this flag is enabled are:

- PUBLIC folder and Index.html.

See "[Modifying access permissions](#)" on page 365 for details on HTTP access limits.

# 29 Audit trails

---

The Audit trail is a chronological sequence of audit records. Each record contains information on the actions executed and the user that performed them.

This function provides process tracking and user identification with time stamp for events.

---

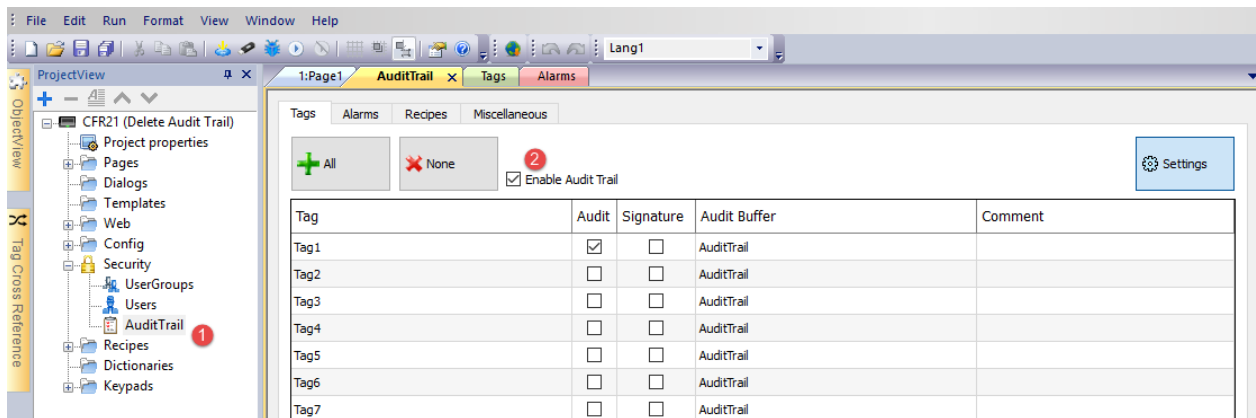
<b>Enable/disable audit trail</b> .....	<b>376</b>
<b>Electronic Signature</b> .....	<b>377</b>
<b>Table audit widget</b> .....	<b>379</b>
<b>Exporting audit trail as .csv files</b> .....	<b>381</b>

# Enable/disable audit trail

Path: **ProjectView** > **Security** > double-click **AuditTrail**

Audit trail logging can be enabled from the “Enable Audit Trail” check box

When enabled, all changes to the selected resources will be logged to the audit buffer with the time stamp, user name that performed the operation and some additional information concerning the modified resource (e.g. new value and previous value for tags)



From the main tabs (Tags, Alarms, Recipes and Miscellaneous) of the Audit trail Editor you can switch between the list views of the available resources.

Parameter	Description
<b>Audit</b>	Enable tracking of the selected resource
<b>Signature</b>	The user password is required before allowing the resource to be modified from the user (see <a href="#">"Electronic Signature" on the facing page</a> to additional information)
<b>Audit Buffer</b>	Internal buffer where store the related audit events (see <a href="#">"Events Buffer" on page 281</a> to additional information)
<b>Comment</b>	Comment space available for the developers

## Tags

- Keep track of when tag value changes.

## Alarms

- Keep track of when user acknowledges or resets an alarm event

## Recipes

- Keep track of when user downloads or uploads recipes

## Miscellaneous Resources

- User login details  
Keep track of when user login, logout or change password
- User management actions  
Keep track of when a user is added, removed or when the user properties are modified

- System actions  
Keep track of system actions (HMI Device Restart, Power On, Backup, Update, Download, enter in System Setting, open Project Manager)
- FTP actions  
Keep track of ftpGET, ftpPUT, OpenTextEditor, SaveTextEditor
- Buffer actions  
Keep track of dump and delete actions on alarms, audit or trends buffers

### LogMessage Macro

In addition of that, the LogMessage macro gives the possibility to define additional events to log to the audit trail buffer. See "[LogMessage](#)" on page 229 for additional details.

### Cache Memory



Data is temporarily saved in cache memory and flushed to file system when at least one of the following conditions is true:

- temporary cache buffer is full
- an explicit dump procedure has been called
- 5 minutes cycle time has expired

Warning: data in cache memory will be lost if there is a power failure before data has been flushed to the file system.

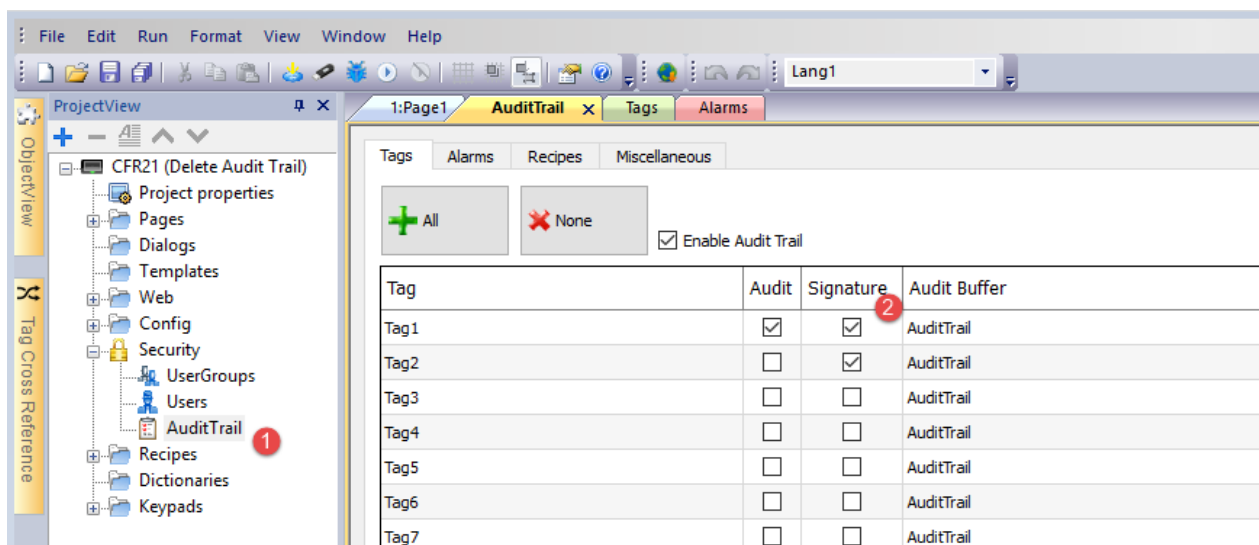
### Backup audit events

From the "[Events Buffer](#)" on page 281 you can configure the size of the audit buffer and activate the backup of the audit events when the buffer is full.

## Electronic Signature

For each resources listed within the Audit Trail editor, it is possible configure the HMI Runtime to require the password confirmation before changing it. If the audit trail log is enabled, the user has the option of adding a comment that will be recorded within the Track Log.

Path: **ProjectView**> **Security** > double-click **AuditTrail**



The user password is required before allowing the resource to be modified by the user

## Confirm your password

Password:

\*\*\*\*\*

Comment:

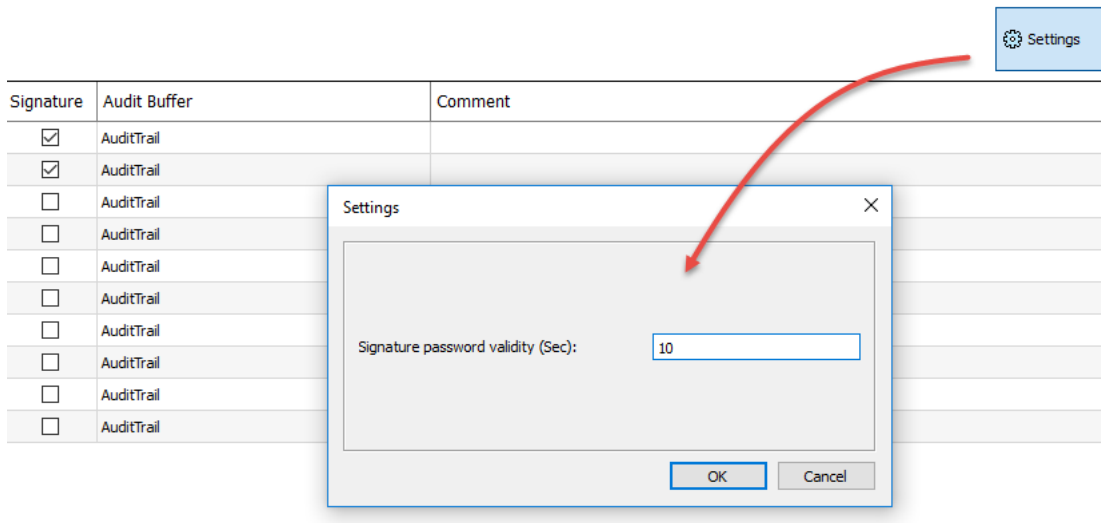
This is a short comment that explains why I am doing this change

OK

Cancel



The introduced password will be not required again for the commands released in the next 10 Sec. The validity time can be modified from the Settings dialog.



## Table audit widget

Path: **Widget Gallery**> **Basic**> **Audit Tables**

Display contents of the audit trail inside a widget

### Audit View

**From:** 04/20/22 - 15:03:21

**To:** 04/20/22 - 16:03:21

**Refresh**

1 Hour ▼

Filter: UserName ▼ 🔍 X

Record ID	Timestamp	UserName	Operation	Status	Information
1	04/20/22 - 16:02:56	admin	LOGOUT	S_OK	1
2	04/20/22 - 16:03:02	system admin	DOWNLOAD_PROJECT	S_OK	project82.jpr
3	04/20/22 - 16:03:04	admin	LOGIN	S_OK	1
4	04/20/22 - 16:03:15	admin	WRITE_TAG	S_OK	Tag1;0;0
5	04/20/22 - 16:03:16	admin	WRITE_TAG	S_OK	Tag1;0;1
6	04/20/22 - 16:03:17	admin	WRITE_TAG	S_OK	Tag1;1;0

▼ ▲

### Buttons:

- **REFRESH**  
Retrieve trend data from internal buffer and refresh table view
- **BACKWARD/FORWARD**  
Move the display window forward or backward as specified in the duration parameter

### Filter:

Use the combo box to select the column where search for and the text filed on the right to enter the string to search to.

Parameter	Description
<b>AuditBuffer</b>	Event Buffer from which the event list is retrieved (see <a href="#">"Events Buffer" on page 281</a> )
<b>Heading</b>	Heading label
<b>Default Duration</b>	Initial value of time window to show
<b>End Time</b>	Upper limit of the time displayed in the table in units of 1 second
<b>Time Spec</b>	Time format: <ul style="list-style-type: none"> <li>• <b>Local</b> = show the time values of the HMI device.</li> <li>• <b>Global</b> = show the time values using UTC format.</li> </ul>
<b>Date Format</b>	Select the Date and Time format
<b>Filter List</b>	Labels to show in filter column selection
<b>Timestamp Sorting</b>	Set how to sort the time stamp data <ul style="list-style-type: none"> <li>• Ascending</li> <li>• Descending</li> </ul>
<b>Table Layout</b>	Defines the characteristics of the scroll bar and allows to remove the header of the table

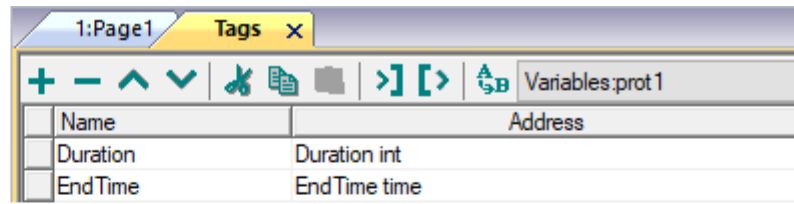
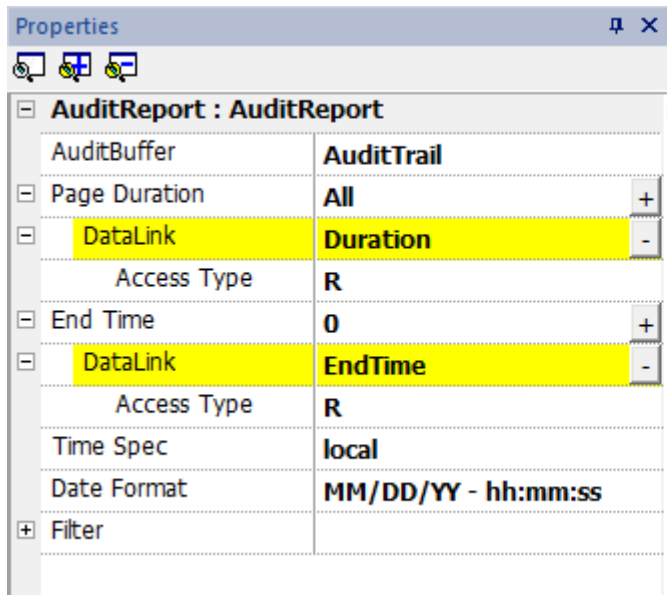
## Printing audit table

An audit table widget without buttons can be found and used from the print report gallery. The table can be drawn and enlarged to fill the entire page. If the number of lines to printed is greater of one page, the audit table will be printed using additional pages.

Using the "attach to tag" feature is possible to use tags to define some properties of the historical trend to print at runtime:

- Page Duration
- End Time

"Page Duration" with "End Time" define the piece of the audit buffer to print.



## Exporting audit trail as .csv files

Data recorded inside the audit trail can be exported inside a csv file using the **SaveEventArchive** action. See ["SaveEventArchive" on page 227](#) for details.

### File structure

	A	B	C	D	E	F	G	H	I	J	K	L
1												
2	Record ID	Date	Time	User ID	Interface	Action	Status	Data				
3	1	27/03/2018	14:22:06	SYSTEM_IDAL	SYSTEM_IDAL	SYSTEM_POWERON	S_OK					
4	2	27/03/2018	14:22:06	admin	LOCAL	LOGIN	S_OK	1				
5	3	27/03/2018	14:22:08	admin	LOCAL	WRITE_TAG	S_OK	Tag1	0	1		
6	4	27/03/2018	14:22:09	admin	LOCAL	WRITE_TAG	S_OK	Tag2	0	1		
7	5	27/03/2018	14:22:26	admin	LOCAL	WRITE_TAG	S_OK	Tag2	1	5	This is a test	
8	6	27/03/2018	14:22:50	SYSTEM_IDAL	SYSTEM_IDAL	RECIPE_WRITE_TAG	S_OK	Tag1	1	1		
9	7	27/03/2018	14:22:50	SYSTEM_IDAL	SYSTEM_IDAL	RECIPE_WRITE_TAG	S_OK	Tag2	5	3		
10	8	27/03/2018	14:22:50	SYSTEM_IDAL	SYSTEM_IDAL	RECIPE_WRITE_TAG	S_OK	Tag3	0	5		
11	9	27/03/2018	14:22:50	admin	LOCAL	DOWNLOAD_RECIPE	S_OK	Recipe0	set-00			
12	10	27/03/2018	14:22:54	admin	LOCAL	ACK_ALARM	S_OK	Alarm2				
13	11	27/03/2018	14:22:58	admin	LOCAL	RESET_ALARM	E_FAIL	Alarm2				
14	12	27/03/2018	14:23:02	admin	LOCAL	DUMP_AUDIT_BUFFER	S_NEEDNOT_NOTIFY	AuditTrail				
15												
16												
17	Record ID	Date	Time	User ID	Interface	Action	Status	Data				
18	13	27/03/2018	14:23:24	admin	LOCAL	DELETE_AUDIT_BUFFER	S_OK	AuditTrail				
19	14	27/03/2018	14:23:26	SYSTEM_IDAL	SYSTEM_IDAL	RECIPE_WRITE_TAG	S_OK	Tag1	1	2		
20	15	27/03/2018	14:23:26	SYSTEM_IDAL	SYSTEM_IDAL	RECIPE_WRITE_TAG	S_OK	Tag2	3	4		
21	16	27/03/2018	14:23:26	SYSTEM_IDAL	SYSTEM_IDAL	RECIPE_WRITE_TAG	S_OK	Tag3	5	6		
22	17	27/03/2018	14:23:26	admin	LOCAL	DOWNLOAD_RECIPE	S_OK	Recipe0	set-01			
23	18	27/03/2018	14:23:27	user1	CGI	LOGIN	S_OK	192.168.49.242				
24	19	27/03/2018	14:23:37	user1	CGI	WRITE_TAG	S_OK	Tag1	6	55		
25	20	27/03/2018	14:24:28	admin	LOCAL	DUMP_AUDIT_BUFFER	S_NEEDNOT_NOTIFY	AuditTrail				
26												



Exported data file has the following content	
<b>RecordID</b>	Each record is stored with a progressive number which will give the possibility to easily identify missing records or confirm that they are not lost. Note that the progressive number is not reset to zero when the buffer is deleted.
<b>Date, Time</b>	Event time stamp. Time can be configured as local or global from the dump action.
<b>User ID</b>	User that perform the operation
<b>Interface</b>	LOCAL: when the action is performed in the HMI device CGI: when the action is performed by a remote client. SYSTEM_IDAL: when the action is performed from the HMI Runtime application
<b>Action</b>	Action executed.
<b>Status</b>	Result of the executed action <ul style="list-style-type: none"> <li>• S_OK Action executed correctly</li> <li>• E_FAIL Action non executed</li> <li>• S_NEEDNOT_NOTIFY Action triggered (will be executed asynchronously)</li> </ul>
<b>Information</b>	Additional info related with the executed action.

# 30 Reports

---

A report is a collection of information that will be printed when triggered by an event. When the programmed event is triggered, the printing starts in background.

You can configure reports, their contents, trigger conditions and output printer in the Reports editor.

Not all widgets can be used in reports. When configuring reports, PB610 Panel Builder 600 provides access to a dedicated widget gallery featuring only widgets available for reports.

Reports format can be customized using predefined templates for page layout.



Note: Report printing is not supported by HMI Client.

---



<b>Adding a report</b> .....	<b>384</b>
<b>Configuring text reports</b> .....	<b>384</b>
<b>Configuring graphic reports</b> .....	<b>385</b>
<b>Print triggering events</b> .....	<b>386</b>
<b>Default printer</b> .....	<b>387</b>

# Adding a report

Path: **ProjectView** > **Config** > double-click **Reports**

In **Reports** editor, click **Graphic Report** or **Text Report**: one new row is added to the table.

## Report types

Report type	Description
<b>Text Reports</b>	<p>Use for line-by-line printing of alarms.</p> <p>Only used for line printers.</p> <p>Text is sent to the printer without using any special driver.</p> <p> <b>Important: This printing mode requires using a physical port and only works on Windows CE platforms.</b></p>
<b>Graphic Reports</b>	<p>Contain graphical elements and may include complex widgets such as screenshots or alarms.</p> <p> <b>Important: Each printer requires a specific printer driver. See "<a href="#">Configuring graphic reports</a>" on the facing page for a list of supported printer drivers.</b></p>

## Configuring text reports

Use the **Reports** editor . **Paper Size** in number of characters.

### Setting printer options

Use printer options to control flush of pages on printer.

Printing starts either immediately or after a timeout. In printer options you can force flush as soon as a specific condition occurs, after a specified number of events, lines or seconds.



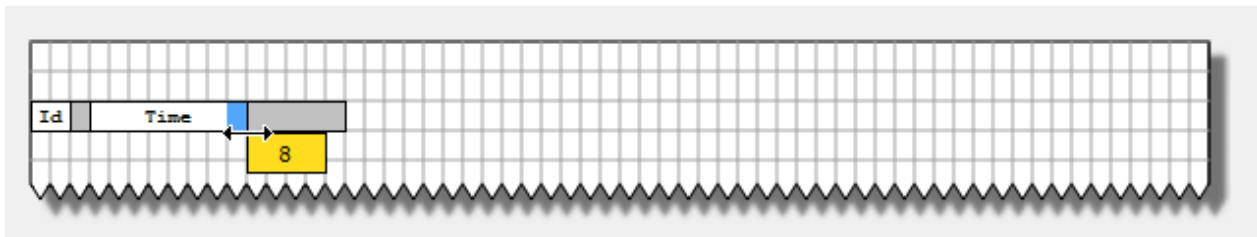
Note: Text reports do not support PDF format.

### Setting alarms layout

**Paper Size** is the width of paper in number of characters.

### Adding fields to the report

To add an item to the report, drag and drop it on the template page from the **Available fields** list.



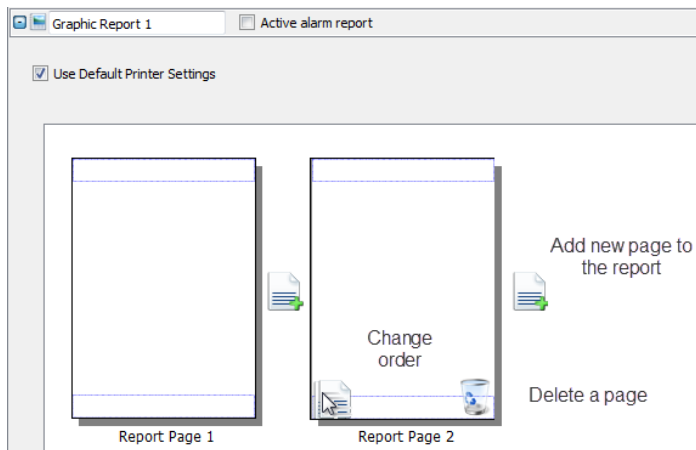
Re-size the field using the mouse, a tool tip shows the dimension in number of characters.



Note: If the text does not fit in the dedicated space, the auto wrap is applied.

## Configuring graphic reports

Use the **Report** editor to configure graphic reports.



### Adding a report page

Click **+** to add a new page to the report layout.

When the mouse goes over a page, two icons are displayed and allow you to reorder or delete the pages.


### Modifying report page content

1. Double click on a page to edit its content: the **Graphic Report** editor appears.  
Each page is divided in: header, footer and page body.
2. Double click on the area you want to edit: the edit area is shown in white, others are grayed out.

The Widget Gallery is context-sensitive and displays only the widgets available for the area you are editing.

### Widgets available for reports

Widgets that can be used for a graphic report:

Widget	Function
Page Number	Automatic page numbering
Screenshot	Screen capture of the page currently displayed by the HMI device. The report page is automatically resized to fit the HMI device page.   Note: The full screen is printed, including all open dialogs.
Alarm	Entire contents of the event buffer (default buffer is Alarm Buffer1).
Text	Widgets such as labels and numeric fields

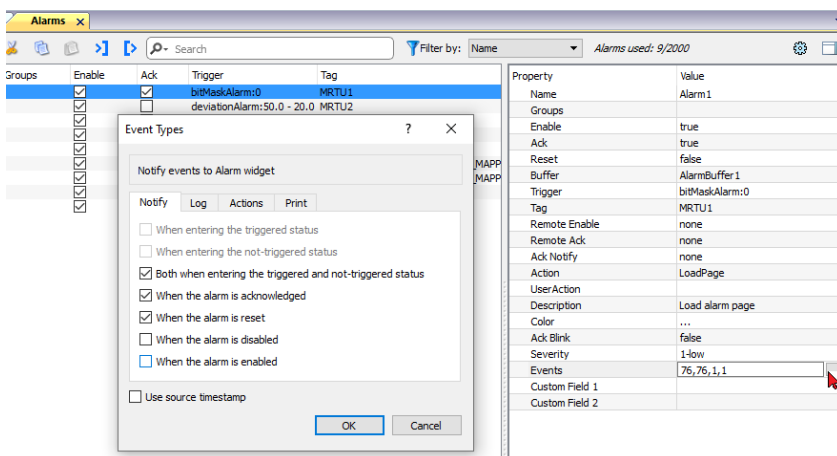
## Print triggering events

Report printing can be triggered by events.

### Configuring alarm printing

Path: **ProjectView** > **Config** > double-click **Alarms**

1. In the Alarms editor, open the **Event Types** dialog from the **Events** property.
2. In **Print** tab select all the conditions for which you want to trigger printing.

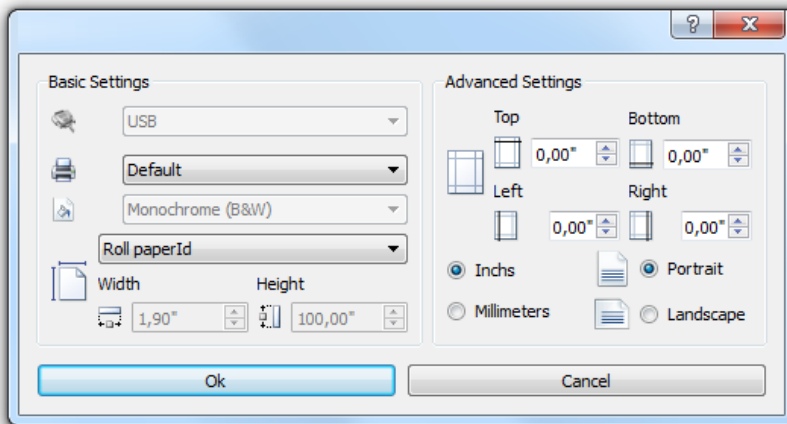


 **Important:** Only one report can be set as Active alarm report in a project and it can be either a text report or a graphic report.

### Adjusting printer settings at runtime

A graphic report printing can be started also using the action **PrintGraphicReport**.

Set the action property **silent** to **false** to have a pop-up dialog.



## Default printer

### Printer setting

You can set a default printer for all graphic reports. Each report can then be configured to use the default printer or any other printer available. Click **Printer Setting** button to set printer parameters.

For PDF printers you also define the folder where files are saved by using **Printed Files Location**.



On WIN32, only pdf and default printers (from any port) are supported.



On Linux platforms USB printers are not supported

### Printers that should work with the Windows CE driver preinstalled on the system

The below table list the printers for which the printer manufacturer states that they work with the Windows CE printer driver.



**If a mentioned printer really works must be tested before by the user.**

Printer	Languages
HP PCL 3, HP PCL 5e, HP PCL3GUI	HP PCL3/PCL5e/PCL3GUI, including DeskJet, LaserJet, DesignJet
Epson ESC/P2	ESC/P2, LQ
Epson Stylus Color	Epson Stylus Color
Epson LX (9-pin)	9-pin printers, Epson LX, FX, PocketJet

Printer	Languages
<b>Cannon iP100, iP90, BubbleJet</b>	BubbleJet, iP90, iP100
<b>PocketJet II, 200, 3</b>	Pocket Jet
<b>MTE Mobile Pro Spectrum</b>	MTE Mobile Pro Spectrum
<b>Adobe PDF File</b>	Adobe PDF file
<b>SPT-8</b>	SPT-8
<b>M1POS</b>	M1POS
<b>MP300</b>	MP300
<b>Zebra</b>	Zebra CPCL language
<b>Intermec PB42, PB50, PB51, PB2, PB3</b>	Intermec PB42/50/51/2/3 with ESC/P language
<b>Datamax Apex</b>	Datamax Apex

## Supported ports


The following ports are supported:

- LPT1 (USB printers) only at panel operating system Windows CE
- File (PDF)

## Tested printers

The following printers have been tested with printCE drivers in Windows CE HMI devices.

Driver	Printer Model	Graphic	Line
<b>Custom</b>	Plus 4 Kube II	Yes	Yes
<b>Epson ESC/P 2</b>	Epson AcuLaser M2310	Yes	Simulate
<b>Epson LX (9-pin)</b>	Epson LX-300+II	No	Yes
<b>HP PCL 3</b>	HP LaserJet P2015dm	Yes	Simulate
	HP LaserJet 4700dtn	Yes	Yes
<b>HP PCL 3 GUI</b>	HP Deskjet 1010	Yes	No
	HP Deskjet D5560	Yes	No
	HP LaserJet 4700dtn	No	Yes
<b>HP PCL 5e</b>	HP LaserJet P2015dm	Yes	Simulate
	HP LaserJet 4700dtn		

Driver	Printer Model	Graphic	Line
<b>INTERMEC</b>	Intermec PB50 with ESC/P language with 4 inch roll paper.  Note: During the tests of this printer with UN31 devices, after a successfully printing, the device reported unexpected behavior. This does not affects the other Windows CE platforms, and it is reproducible only with printers in standby mode.	Yes	Yes
<b>PDF</b>	-	Yes	No





# 31 Screen saver

Screen saver can be used to execute actions and/or display a slide show when the HMI device is not in use. The screen saver starts after a timeout if none of the following events occur:

- touch of display
- mouse movement
- external keyboard key pressed
- active dialogs

When the display is touched or a mouse movement is detected or a key from an external keyboard is pressed or a dialog is launched, if the screen saver is active it is deactivated.

## Enabling the screen saver function

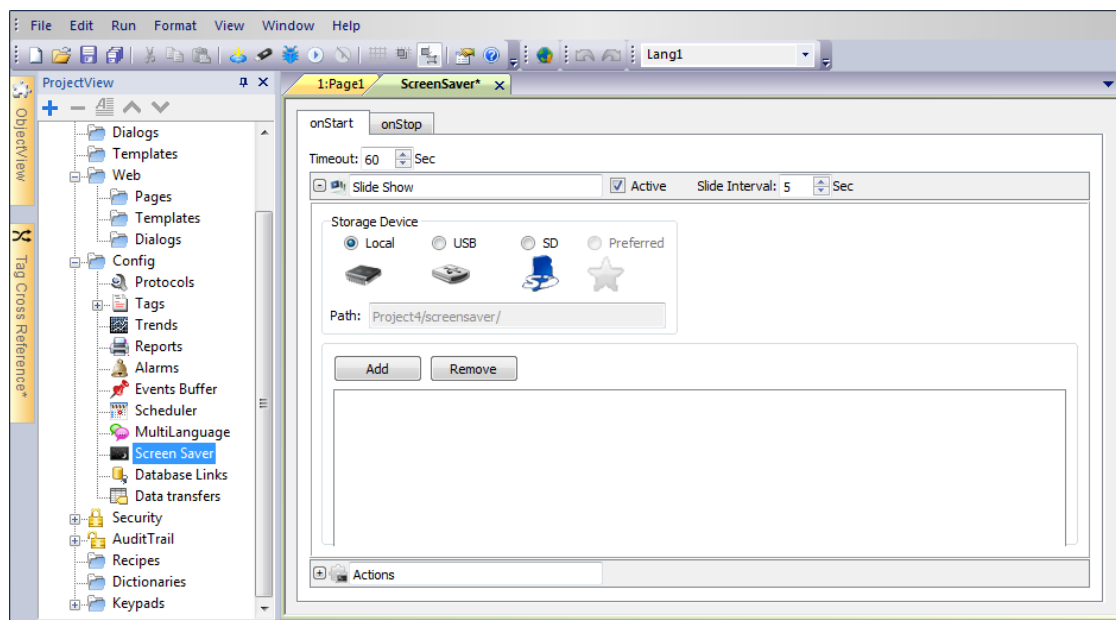
Path: **ProjectView**> **Config** > right-click **Screen Saver**> **Enable**



**Important:** You must enable the screen saver before you can configure it.

## Configuring a screen saver

Path: **ProjectView**> **Config** > double-click **Screen Saver**




### Timeout

Time after which the slide show starts

---

## Slide show parameters

Parameter	Description
Slide Interval	Interval between slides.
Storage Device	<p>Location of the images used in the slide show.</p> <p>Images stored locally are saved in <i>workspace\projectname\screensaver</i> and can be downloaded to the HMI device when the project is downloaded.</p> <p>Images stored on USB or SD devices are saved in a screensaver folder on the device itself.</p> <p> <b>Important: Only JPEG and PNG images are supported.</b></p>
Add/Remove	Add / Remove images to show.

## Associating actions to the screen saver

Actions can be triggered by the screen saver start and/or stop.

- Click + next to **Actions** in the **onStart** tab to configure actions to be executed when the screen saver starts.
- Click + next to **Actions** in the **onStop** tab to configure actions to be executed when the screen saver stops.

# 32 Backup/restore of Runtime and project

---

You can backup all the content of the HMI device, including

- HMI Runtime
- HMI Application Project

to an external memory. This backup copy can be used to restore the content of the HMI device at a later time or copy it to a new HMI device.

The backup function is available only if enabled for the logged user. See ["Modifying access permissions" on page 365](#) for details.



Note: Backup is not supported in Windows Runtime and in HMI Client.

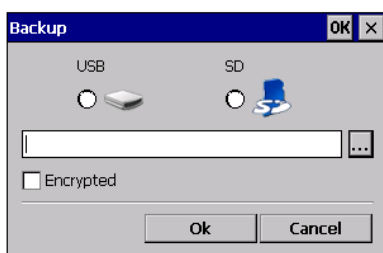
## Backup function

The backup function automatically performs the following procedure:

1. Unloads the current project to unlock files in use.
2. Archives the content of the \QTHMI folder (containing HMI Runtime, projects, dynamic files such as recipes, alarms, trends and so on) to a .zip file (standard or encrypted).
3. Reset the HMI device (reloads the project).

To start the backup procedure:

1. In HMI Runtime right click to open the context menu.
2. Select **Backup**: the **Backup** dialog is displayed.



3. Select the path for storing the backup file.



Note: The backup process does not include files stored in USB and SD cards. Dynamic data such as recipes, trends, events stored in these devices will not be included in the backup.

## Restore function

Restore the backup package can be perform on HMI device

- 
- from the Context Menu (see ["Update package"](#) on page 104 for details)
  - or from the System Settings (see ["System Settings"](#) on page 622 for details)

# 33 Keypads

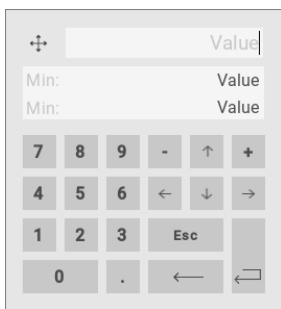
---

Several keypads are provided by default in the PB610 Panel Builder 600 so that they can be used for data entry.

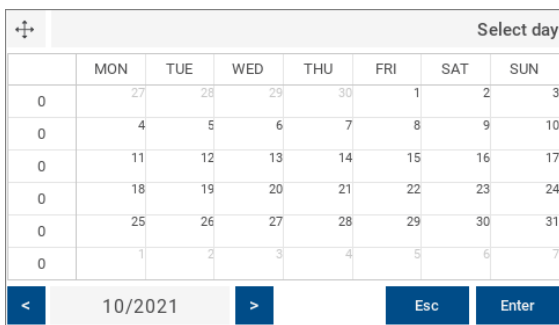
The alphabet keypad can be use associate with a string data type



The numeric keypad can be use associate with a numeric data type



The calendar keypad can be use associate with a date data type




---

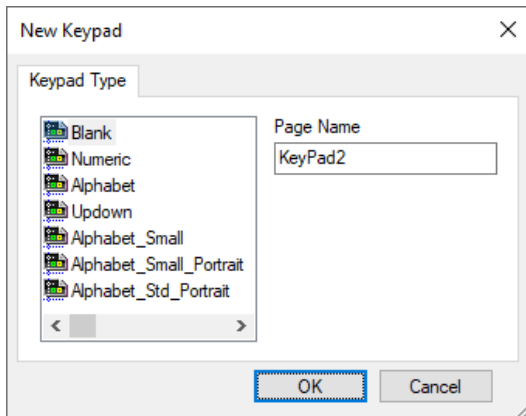
<b>Creating and using custom keypads</b> .....	<b>396</b>
<b>Deleting or renaming custom keypads</b> .....	<b>399</b>
<b>Keypad type</b> .....	<b>400</b>
<b>Keypad position</b> .....	<b>400</b>

# Creating and using custom keypads

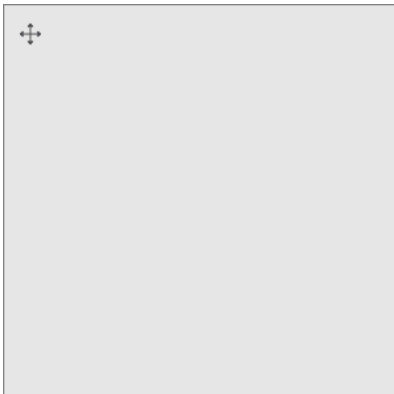
You can either create a new keypad or customize an existing one.

## Creating a keypad

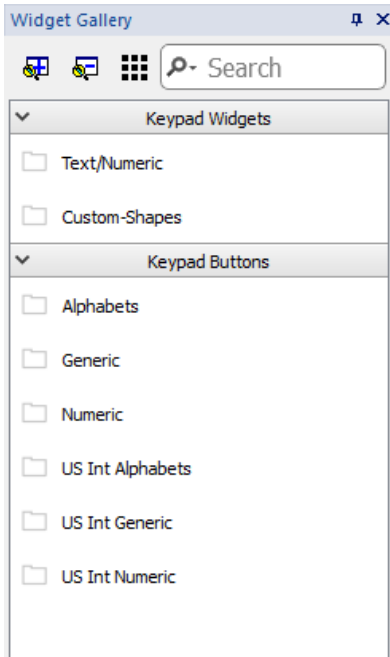
1. In **ProjectView**, right-click **Keypads** and select **Insert Keypad**: the **New Keypad** dialog is displayed.



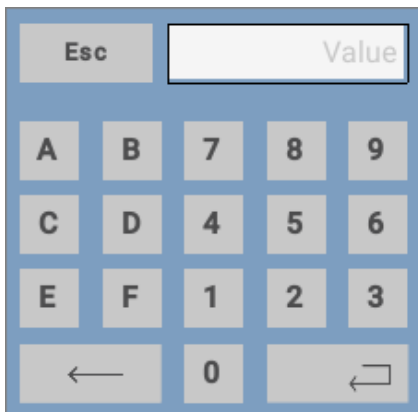
2. Select one of the available keypads, or **Blank** to create a keypad from scratch. In this case a blank keypad is displayed.



- Use the **Keypad Widgets** and **Keypad Buttons** from the Widget Gallery to create your custom keypad.



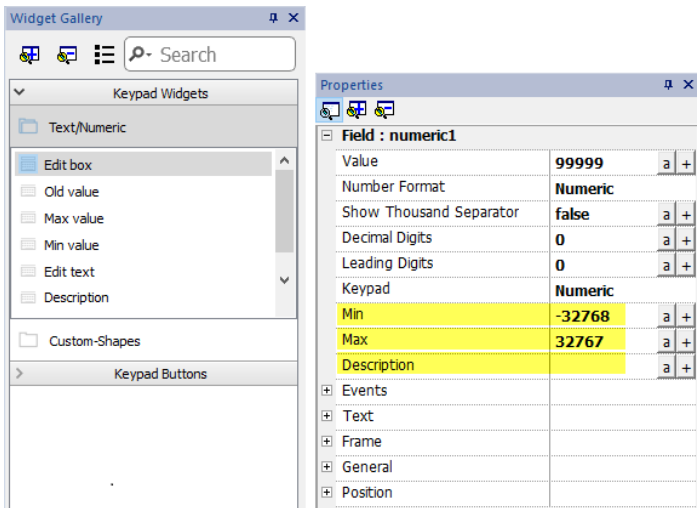
The keypad you create, as in this example, will be saved in the project folder.



### Text/Numeric Controls

The Text/Numeric folder contains some specific controls to keypad development.

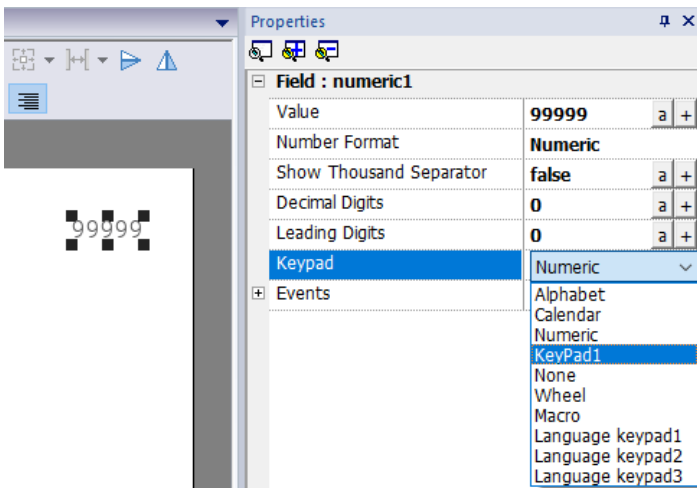




Data source	Description
<b>Edit box</b>	New value
<b>Old value</b>	Current value
<b>Min value</b>	Min value defined inside the field's property currently editing.
<b>Max value</b>	Max value defined inside the field's property currently editing.
<b>Edit text</b>	Simple text label
<b>Description</b>	Description defined inside the field's property currently editing.

## Attaching custom keypads to fields

Custom keypads can then be reused for any field where the **Keypad** property points to it as in this example.



## Tips and tricks with custom keypads

By default, any numeric widget (read/write numeric field) are assigned the numeric keypad.

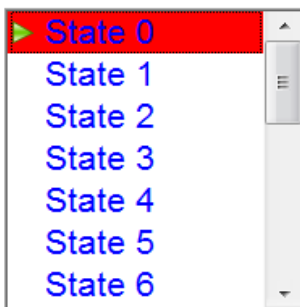
If you want to apply a customized version of the numeric keypad to all the numeric widgets you add to your project proceed as follows:

1. Create a new keypad and select **Numeric** as **Keypad** type. This will be a backup of the original settings for the numeric keypad.
2. Customize the default numeric keypad and save it. This customized version of the numeric keypad will now be assigned as default in the project.

See "[Deleting or renaming custom keypads](#)" below for details on how to rename a custom keypad.

## Up-down arrows keypad

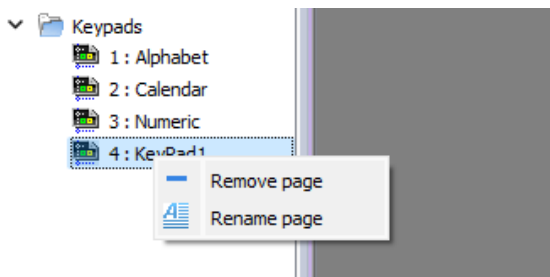
This type of keypad is particularly useful to move the cursor up and down within widget requiring this functionality. Here an example using a **Control List** widget. See "[Control list widget](#)" on page 463 for details.



## Deleting or renaming custom keypads

In **ProjectView**, right-click on a custom keypad and select one of the options:

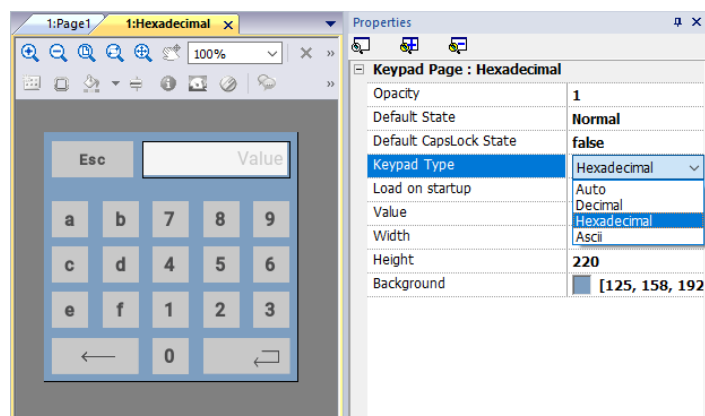
- **Remove Keypad Page** to remove the keypad from the project
- **Rename Keypad Page** to rename the keypad.



# Keypad type

Path: **ProjectView** > **Keypads** > double-click a keypad > **Properties**

Set **Keypad Type** parameter for a keypad to define the type of data entry.



Keypad Type	Description
<b>Auto</b>	Default setting
<b>Decimal</b>	Only numeric keys are accepted. Entering 10, the keypad returns 10 that will be displayed as "10" if the attached field is numeric or ASCII, as 'A' if the attached field is hexadecimal.
<b>Hexadecimal</b>	Only hexadecimal keys are accepted. Entering 10, the keypad returns 16 that will be displayed as "16" if the attached field is numeric or ASCII, as "10" if the attached field is hexadecimal.
<b>Ascii</b>	All keys are enabled. Entering 1A, the keypad returns 1A that will be displayed as '1' if the attached field is numeric, as "1A" if the attached field is ASCII or if the attached field is hexadecimal.

# Keypad position

**Runtime Positioning** property of keypads can be used to define where keypads will appear in the screen.

Option	Description
<b>Automatic</b>	The best position is selected according to here data entry is required.
<b>Absolute</b>	X,Y coordinates are entered to identify the exact position
<b>Left-top</b>	Predefined screen positions
<b>Left-center</b>	
<b>Left-bottom</b>	
<b>Center-top</b>	

Option	Description
<b>Center-center</b>	
<b>Center-bottom</b>	
<b>Right-top</b>	
<b>Brightener</b>	
<b>Right-bottom</b>	

Select the **Lock Keypad position** option if you do not want the keypad to be moved by dragging.



# 34 External keyboards

HMI Runtime has been designed to work with external keyboards connected via USB.



USB Barcode readers that support keyboard emulation mode can be used.

Keyboards can be used for:

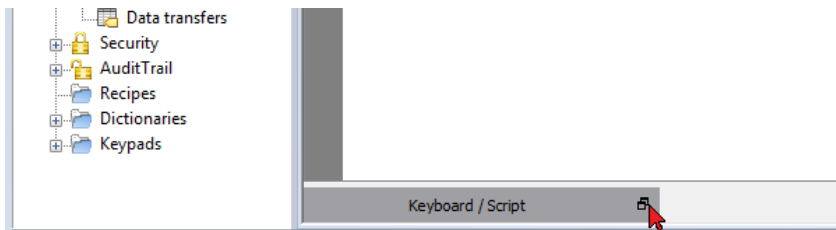
- data entry (default)
- execute actions mapped on specific keys

For example, the right arrow key **OnClick** event can be mapped to the **LoadPage** action.

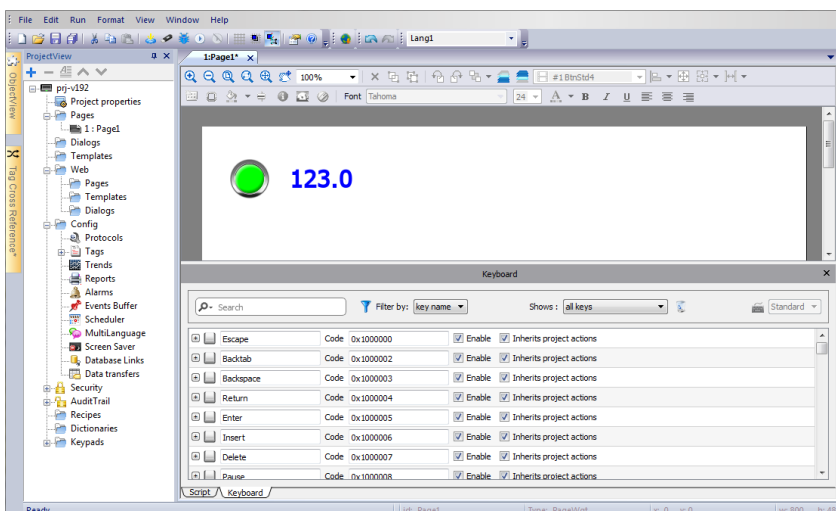
Keyboard can be programmed at project level so that settings will be inherited by all the pages. In each page you can then choose which key setting will be inherited from the project and which one you will customize for the specific page.

## Opening external keyboards

1. In the Page Editor, click on the icon on the right of **Keyboard/Script** at the bottom of the workspace: the Keyboard/Script Editor is displayed.
2. Select the **Keyboard** tab.



Each row in the Keyboard Editor corresponds to a key.



For each key, the following information is displayed:

Element	Description
Label	Key name
Code	Key code
Enable	Key enable status
Inherits project actions	Defines whether the key is inheriting the action programmed at the project level

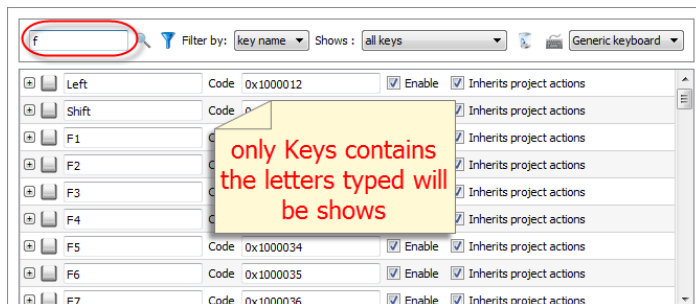
Here the possible configurations:

Enable	Inherits project actions	Editor appearance	HMI Runtime behavior
Checked	Unchecked	Action lists show the page actions (or nothing if the list is empty)	Only the page actions (if any) will be executed.
Checked	Checked	Action lists show the project actions only and cannot be edited	Only the configured project actions (if any) will be executed.
Unchecked	Checked	Inherits project actions check box and all action lists are disabled. Action lists show the project actions only.	No page or project action will be executed.
Unchecked	Unchecked	Inherits project actions check box and all action lists are disabled. Action lists show the project actions only.	No page or project action will be executed.

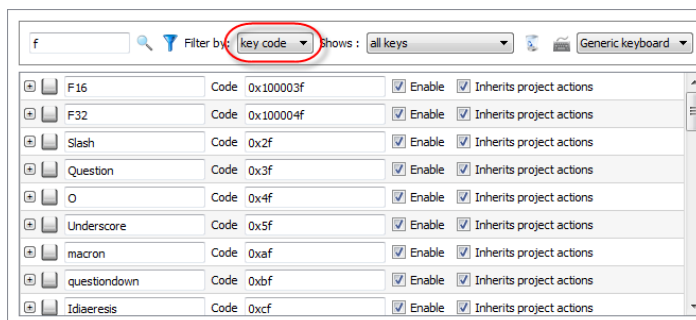
<b>Search and filter</b> .....	<b>405</b>
<b>Displayed keys</b> .....	<b>405</b>
<b>Removing action associations</b> .....	<b>405</b>
<b>Keyboard layout</b> .....	<b>406</b>
<b>Enable/disable keyboard</b> .....	<b>406</b>
<b>Associating actions to keys</b> .....	<b>406</b>

## Search and filter

To display a filtered set of keys, in **Filter by** select **key name** and type a letter in the search field: only the keys containing that letter in their name will be displayed in the Keyboard editor.



Alternatively, in **Filter by** select **key code** and type a letter in the search field: only the key containing that letter in their code will be displayed in the Keyboard editor.



## Displayed keys

You can easily select what keys will be listed in the Keyboard editor window. To display a limited set of keys, select an option in **Shows**.

Option	Description
<b>all keys</b>	All keys available in the keyboard layout are listed
<b>modified keys</b>	Only the keys associated with actions at the page level are listed
<b>modified keys in project</b>	Only the keys associated with actions at project level are listed

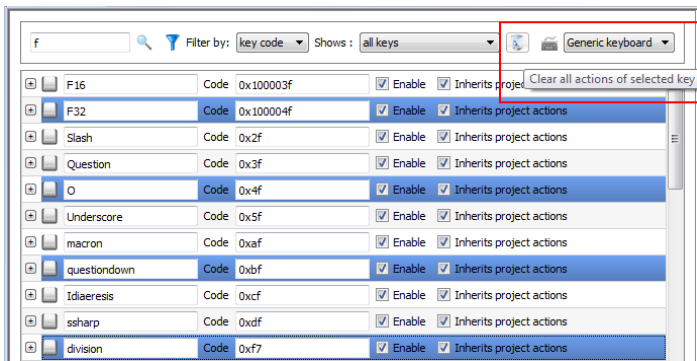
## Removing action associations

To remove all the associations you created between keys and actions:

1. Select the keys for which you want to remove the association.
2. Click the **Clear all actions of selected keys** button.

If you are working at page level, page actions will be removed, if you are working a project level, project actions will be removed.



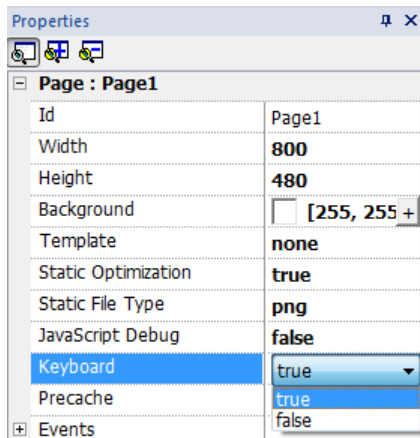


## Keyboard layout

Select the layout of the keyboard from the **Keyboard Layout** combo box. **Generic Keyboard** refers to a generic international keyboard layout.

## Enable/disable keyboard

You can enable/disable keyboard actions both at project and at page level. To enable keyboard actions, in the **Properties** pane set **Keyboard macro** to **true**.

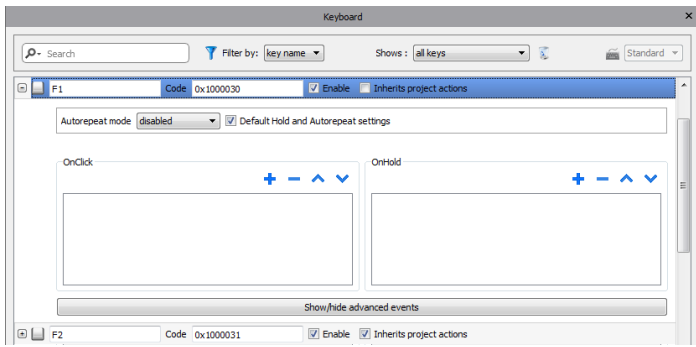


You can enable/disable keyboard actions also at runtime using the KeyboardMacros action. See ["Keyboard actions" on page 199](#) for details.

## Associating actions to keys

You associate actions to a keys from the Keyboard editor.

1. Click **+** next to the key you want to program: the fields for key configuration are displayed.



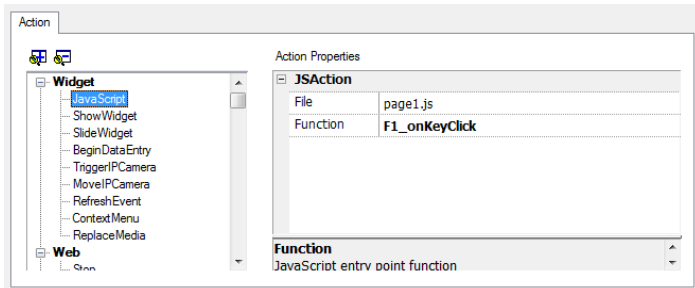
1. Click **+** to add actions.

You can associate actions both to the **OnClick** event and to the **OnHold** event.

See ["Events" on page 59](#) for details.



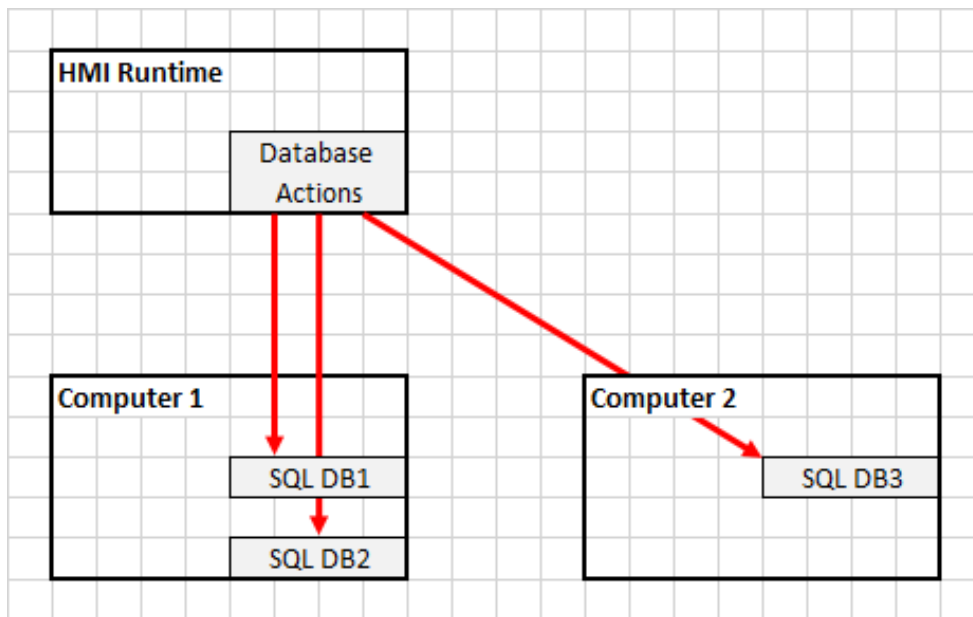
Note: Also JavaScript code can be associated to a key event.





# 35 Storing data to external databases

PB610 Panel Builder 600 allows connecting to external databases through macros or SQL commands (structured query language).



The supported databases are:

- MYSQL
- MariaDB
- PostgreSQL
- ODBC
- SQL4Automation



Available only on Linux devices (See: ["HMI devices capabilities" on page 613](#))

---

<b>Configuring the HMI project</b> .....	<b>411</b>
<b>Transfer data with JavaScript</b> .....	<b>412</b>
<b>Database tables</b> .....	<b>416</b>
<b>Custom tables</b> .....	<b>417</b>
<b>DB table data source</b> .....	<b>417</b>
<b>Database Configuration</b> .....	<b>419</b>
<b>SQL4Automation (legacy)</b> .....	<b>427</b>
<b>Installing SQL4Automation</b> .....	<b>427</b>

---

---

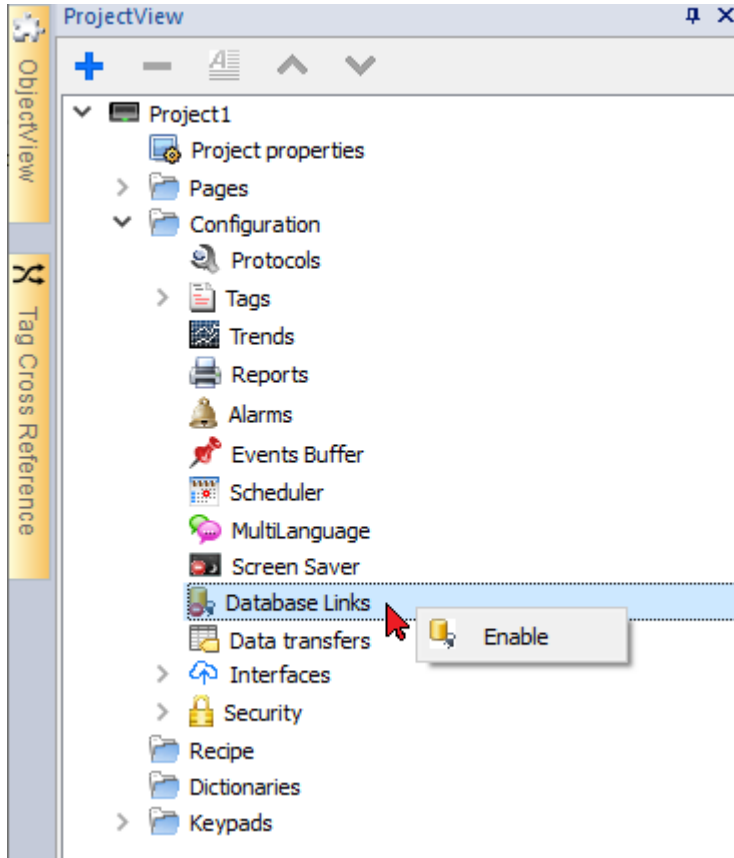
<b>Configuring SQL4Automation .....</b>	<b>428</b>
<b>Connection Limits .....</b>	<b>430</b>

# Configuring the HMI project

## Enable/disable DB Connectors

Path: **ProjectView** > right-click **Database Links** > **Enable**

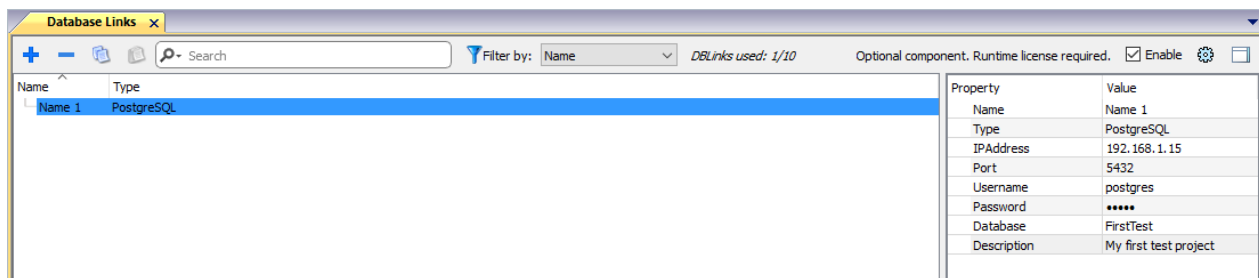
Note the icon change to indicates whether the function is enabled or disabled.



## Configure a Database

Path: **ProjectView** > **Config** > double-click **Database Links**

To use an external database it is necessary to create a connection with the specific database by specifying the database access parameters. Later you can use the macros and the JavaScript APIs by referencing the database by the name given to the link.



Parameter	Description
<b>Name</b>	The name to use to reference the database defined in this row.
<b>Type</b>	The available supported databases are:: <ul style="list-style-type: none"> <li>• MYSQL</li> <li>• MariaDB</li> <li>• PostgreSQL</li> <li>• ODBC</li> <li>• SQL4Automation</li> </ul>
<b>Description</b>	A free text that will help you to recognize the database.
...	The other parameters identify how the database is accessed and depend on the type of the selected database (Ref. " <a href="#">Database Configuration</a> " on page 419).

Status of database connection is available through system variable tags. See "[Default variables](#)" on page 162

Error status can be reset with actions. See "[DBResetErrors](#)" on page 198

### Using the database

You can exchange data with the database through macros or through JavaScript API:

- "[Database actions](#)" on page 195
- "[Transfer data with JavaScript](#)" below

## Transfer data with JavaScript

Note that you can use the database actions defined in "[Database actions](#)" on page 195 even from the JavaScript code as for the below examples:

```
function myButton1_onMouseClicked(me, eventInfo) {
    var CustomSQL = ' ' ;
    var DatabaseLink ='Link1';
    project.dbInit(DatabaseLink, CustomSQL);
};

function myButton2_onMouseClicked(me, eventInfo) {
    var CustomSQL = ' ' ;
    var DatabaseLink ='Link1';
    var Tags ='Alarm1;SystemTime;Tag01;Tag02;';
    project.dbReadTags(DatabaseLink, CustomSQL, Tags);
};

function myButton3_onMouseClicked(me, eventInfo) {
    var CustomSQL = ' ' ;
```

```

var DatabaseLink ='Link1';
var Tags ='Alarm1;SystemTime;Tag01;Tag02;';
project.dbWriteTags(DatabaseLink, CustomSQL, Tags);
};

```

## dbQuery

```
project.dbQuery(databaseLink, customSQL, dbCallback);
```

Using this query you can execute SQL Queries.

Parameter	Description
<b>databaseLink</b>	Link to the database to use
<b>customSQL</b>	String with the SQL query
<b>dbCallback()</b>	Function that will be call when query data are ready

## dbCallBack

```
project.dbCallBack(dbStatus, dbResponse);
```

Parameter	Description
<b>dbStatus</b>	See the below errors list.
<b>dbResponse</b>	Query response. Table column names followed by its rows:  In the example:  Tagname - Tagvalue Tag09 - 103 Tag10 - 302

```

Script
1
2 function JS1_onMouseClicked(me, eventInfo) {
3
4     var customSQL = "SELECT Tagname, Tagvalue FROM Tags WHERE Tagname='Tag09' OR Tagname='Tag10' ORDER BY Tagname"
5     var databaseLink ="Link1";
6     project.dbQuery(databaseLink, customSQL, dbCallback)
7 };
8
9
10 function dbCallback(dbStatus, dbResponse){
11
12     alert("SQL Answer = " + dbResponse + "\ndbStatus = " + dbStatus);
13 };
14
15

```

## Error codes



dbStatus	Description
0	SQL_NO_ERROR
-1	SQL_ERROR
-2	SOCKET_COMM_ERROR_CMD_READ
-3	SOCKET_COMM_ERROR_CMD_VALUE_READ
-4	SOCKET_COMM_ERROR_CMD_STR_DIFF
-5	SOCKET_COMM_ERROR_CMD_READ_RS
-6	SOCKET_COMM_ERROR_CMD_VALUE_READ_RS
-7	SOCKET_COMM_ERROR_CMD_STR_DIFF_RS
-9	SOCKET_COMM_ERROR_CMD_READ_RR
-10	SOCKET_COMM_ERROR_CMD_VALUE_READ_RR
-11	SOCKET_COMM_ERROR_CMD_STR_DIFF_RR
-12	SOCKET_COMM_ERROR_CMD_READ_RC
-13	SOCKET_COMM_ERROR_CMD_VALUE_READ_RC
-14	SOCKET_COMM_ERROR_CMD_STR_DIFF_RC
-15	SOCKET_COMM_ERROR_CMD_READ_RN
-16	SOCKET_COMM_ERROR_CMD_VALUE_READ_RN
-17	SOCKET_COMM_ERROR_CMD_STR_DIFF_RN
-18	SOCKET_COMM_ERROR_READ_STR_RN
-19	SOCKET_COMM_ERROR_CMD_READ_RD
-20	SOCKET_COMM_ERROR_CMD_VALUE_READ_RD
-21	SOCKET_COMM_ERROR_CMD_STR_DIFF_RD
-22	SOCKET_COMM_ERROR_READ_STR_RD
-23	SOCKET_COMM_ERROR_STRING_SIZE
-24	SQL_WRITE_ERROR
-25	SOCKET_COMM_ERROR_CREATE_SOCKET
-26	SQL_ERROR_READ_TAG_FROM_ENGINE
-27	SQL_DB_WRITE_TAG_INSERT_NOT_TRIED
-28	SQL_DB_WRITE_TAG_UPDATE_NOT_SUCCESS

dbStatus	Description
-29	SQL_ERROR_READ_TAG_NAME_COL_NOT_FOUND
-30	SQL_ERROR_READ_TAG_VALUE_COL_NOT_FOUND
-31	SQL_ERROR_READ_TAG_NOT_FOUND
-32	SQL_ERROR_GROUP_READ_FROM_ENGINE
-33	SQL_ERROR_GROUP_NOT_FOUND
-34	SQL_ERROR_GROUP_READ_CHUNK_MISSED
-35	SQL_ERROR_TREND_READ_FROM_ENGINE
-36	SQL_ERROR_EVENT_READ_FROM_ENGINE
-37	SQL_ERROR_EVENT_INIT_PARAMS
-38	SQL_ERROR_RECIPE_READ_FROM_ENGINE
-39	SQL_DB_WRITE_RECIPE_UPDATE_NOT_SUCCESS
-40	SQL_DB_WRITE_RECIPE_INSERT_NOT_TRIED
-41	SQL_ERROR_RECIPE_NOT_FOUND
-42	SQL_ERROR_READ_RECIPE_NAME_COL_NOT_FOUND
-43	SQL_ERROR_READ_RECIPE_SET_COL_NOT_FOUND
-44	SQL_ERROR_READ_RECIPE_ELEMENT_COL_NOT_FOUND
-45	SQL_ERROR_READ_RECIPE_VALUE_COL_NOT_FOUND
-46	SQL_ERROR_READ_RECIPE_ELEMENT_NOT_FOUND
-47	SQL_ERROR_RECIPE_READ_CHUNK_MISSED
-48	SQL_EVENT_TREND_ERROR_NO_DATA
-49	SQL_ERROR_TRENDEVENT_SAMPLE_SQL_ERROR
-50	SOCKET_COMM_ERROR_CMD_VALUE_READ_AR
-51	SQL_DB_WRITE_TAG_EMPTY_GOOD_QUALITY_TAGS
-52	SQL_ERROR_READ_TAGS_WRITE_TO_ENGINE_FAILED
-53	SQL_DB_WRITE_TAG_FAILED
-54	SQL_DB_GROUP_TAGS_NOT_FOUND
-55	SQL_DB_WRITE_RECIPE_INSERT_FAILED
-56	SQL_ERROR_READ_RECIPE_WRITE_TO_ENGINE_FAILED
-200	QT_SQL_ERROR

dbStatus	Description
	Check "Database link error message" system variable for more details. (See Database variables in " <a href="#">Default variables</a> " on page 162).
-300	DB_CONNECTOR_LICENSE_EXPIRED

## Database tables

Here the structure of the database tables used by the database actions.



Note: These tables can be generated on an empty database from the **DBInit** action.

### Table: Tags

FieldName	Text(255)	PRIMARY KEY
TagValue	Text(255)	

### Table: Trends

Id	Long Integer	PRIMARY KEY
TrendName	Text(255)	
SampleTime	Text(255)	
TrendValue	Text(255)	
Quality	Text(255)	
RefreshTime	Text(255)	

### Table: Recipes

Recipe	Text(255)	PRIMARY KEY
SetName	Text(255)	PRIMARY KEY
ElementName	Text(255)	PRIMARY KEY
SetValue	Text(255)	

### Table: Event

Id	Long Integer	PRIMARY KEY
EventName	Text(255)	

<b>SampledTime</b>	Text(255)	
<b>EventType</b>	Text(255)	
<b>EventSubTime</b>	Text(255)	
<b>EventValue</b>	Text(255)	

## Custom tables

SQL queries released from the DB actions are listed inside the project file config\dbconnector.xml.

Modify the commands defined inside this file to customize the SQL strings released from the DB actions and then get access to a different structured database.

### Example

```
CREATE TABLE myTagsTable (tagname VARCHAR(255) PRIMARY KEY, tagvalue VARCHAR(255))
UPDATE myTagsTable SET Tagvalue= '%_JMV' WHERE Tagname= '%_JMT'
INSERT INTO myTagsTable (Tagname, Tagvalue) Values ('%_JMT', '%_JMV')
```

Where "%\_JMV" will be replaced with the tag value and "%\_JMT" with the tag name.

## DB table data source

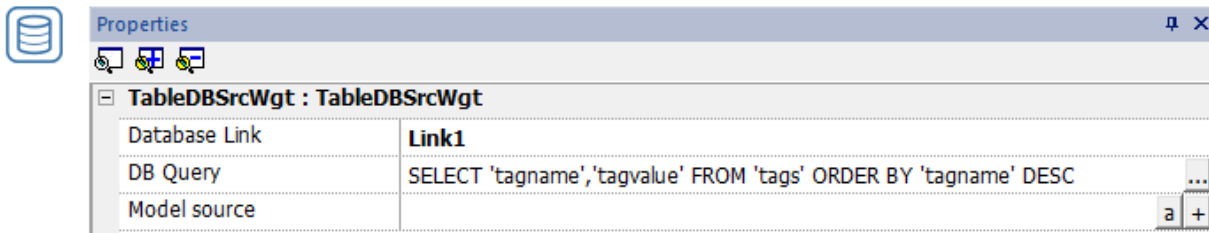
Path: **Widget Gallery**> **Basic**> **Table**> **DB table data source**

The "DB table data source" is a widget that collects data from an SQL database through a SQL query to fill a table.

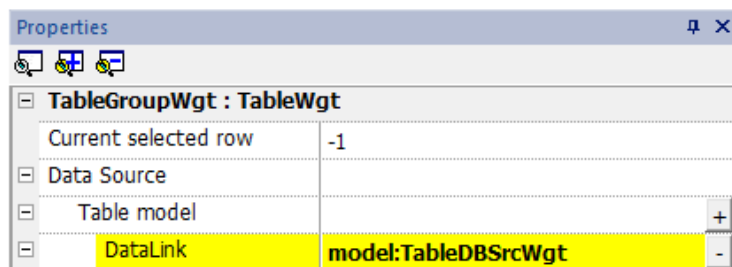
To configure a table with a "DB table data source":

1. Put a "DB table data source" on the page and configure it with the appropriate SQL query
2. Put a "Table group" on the page and select the defined "DB table data source"

## DB table data source



## Table group



## DB Query

Note that each parameter must be enclosed in quotation marks. If quotation marks are required in the SQL query, enclose the parameter in double quotation marks. See the example below:

```
SELECT 'tagname','tagvalue' FROM 'tags' ORDER BY 'tagname' DESC
SELECT 'tagname','tagvalue' FROM 'tags' WHERE 'tagname'="'Tag09'" OR
'tagname'="'Tag10'" ORDER BY 'tagname' DESC
```

## Placeholder

The DB Query accepts placeholders in the form \$(tag name). Those placeholders will be substituted with the value contained inside the tags

For example, if we have a string tag called WHERE (long enough to contain the text we will write), we can configure a filter that can be activated by writing inside the WHERE tag.

The QB Query:

```
SELECT 'tagname','tagvalue' FROM 'tags' $(WHERE) ORDER BY 'tagname' DESC
```

Writing an empty string:

```
WHERE = ""
```

the query will be without filter

```
SELECT 'tagname','tagvalue' FROM 'tags' ORDER BY 'tagname' DESC
```

Writing the filter string:

```
WHERE 'tagname'="'Tag09'" OR 'tagname'="'Tag10'"
```

the query will be with the filter

```
SELECT 'tagname','tagvalue' FROM 'tags' WHERE 'tagname'="'Tag09'" OR
'tagname'="'Tag10'" ORDER BY 'tagname' DESC
```



For the other data source parameters see the ["Configuring the data source"](#) on page 493

## RefreshDBTable

There is an Action, available inside the "Database actions" folder, that can be used to executes the SQL query of the "DB table data source" widget to update its data.

# Database Configuration

This chapter contains some examples of how to configure the parameters needed to access external databases

## Configuring MYSQL, MariaDB and PostgreSQL databases

Property	Value
Name	Link1
Type	PostgreSQL
IPAddress	10.1.35.117
Port	5432
Username	Smith
Password	*****
Database	testDB
Description	

Parameter	Description
<b>Name</b>	Define the name to use to reference the database defined in this item.
<b>Type</b>	Select: ODBC
<b>IPAddress</b>	Server IP Address
<b>Port</b>	Port used from MSSQL Server (generally 1433)
<b>Username</b>	Username for accessing the MSSQL database
<b>Password</b>	Password for accessing the MSSQL database
<b>Database</b>	The name of the database you want to access
<b>Description</b>	This is just a reminder (write here what you prefer)

## Configuring ODBC on PB610 PC Runtime

Open Database Connectivity offers different types of connections

## Accessing to Microsoft databases using MSSQL Server

Property	Value
Name	Link1
Type	ODBC
IPAddress	10.1.35.117
Port	1433
Connection Type	Driver
ODBC Database	MSSQL Server
Driver Name	ODBC Driver 17 for SQL Server
Username	Smith
Password	*****
Database	testDB
Description	

Parameter	Description
<b>Name</b>	Define the name to use to reference the database defined in this item.
<b>Type</b>	Select: ODBC
<b>IPAddress</b>	Server IP Address
<b>Port</b>	Port used from MSSQL Server (generally 1433)
<b>Connection Type</b>	Select: Driver
<b>ODBC Database</b>	Select: MSSQL Server
<b>Driver Name</b>	Type a driver name installed on Windows PC: " <i>ODBC Driver 17 for SQL Server</i> "
<b>Username</b>	Username for accessing the MSSQL database
<b>Password</b>	Password for accessing the MSSQL database
<b>Database</b>	The name of the database you want to access
<b>Description</b>	This is just a reminder (write here what you prefer)

## Accessing to Microsoft databases using MSSQL Server and DSN

Property	Value
Name	Link1
Type	ODBC
Connection Type	DSN
DSN	myODBC
ODBC Database	MSSQL Server
Username	Smith
Password	*****
Description	

Parameter	Description
<b>Name</b>	Define the name to use to reference the database defined in this item.
<b>Type</b>	Select: ODBC
<b>Connection Type</b>	Select: DSN
<b>DSN</b>	Select the DSN to use
<b>ODBC Database</b>	Select: MSSQL Server
<b>Username</b>	Username for accessing the MSSQL database
<b>Password</b>	Password for accessing the MSSQL database
<b>Description</b>	This is just a reminder (write here what you prefer)

### Accessing to an internal "MS Access" file

Property	Value
Name	Link1
Type	ODBC
Connection Type	Driver
ODBC Database	MS Access
Driver Name	Microsoft Access Driver (*.mdb)
MS AccessFile	C:\000\myDB\Database2.mdb
Description	

Parameter	Description
<b>Name</b>	Define the name to use to reference the database defined in this item.
<b>Type</b>	Select: ODBC
<b>Connection Type</b>	Select: Driver
<b>ODBC Database</b>	Select: MS Access
<b>Driver Name</b>	Type a driver name installed on your PC, generally: " <i>Microsoft Access Driver (*.mdb)</i> "
<b>MS AccessFile</b>	Type the MS Access file to use
<b>Description</b>	This is just a reminder (write here what you prefer)



## Accessing to an internal "MS Access" file using DSN

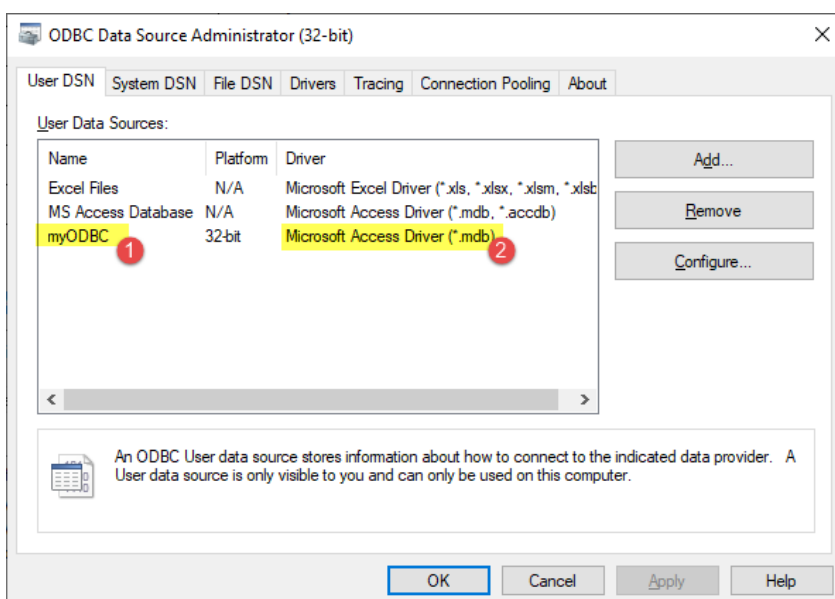
Property	Value
Name	Link1
Type	ODBC
Connection Type	DSN
DSN	myODBC
ODBC Database	MS Access
Description	

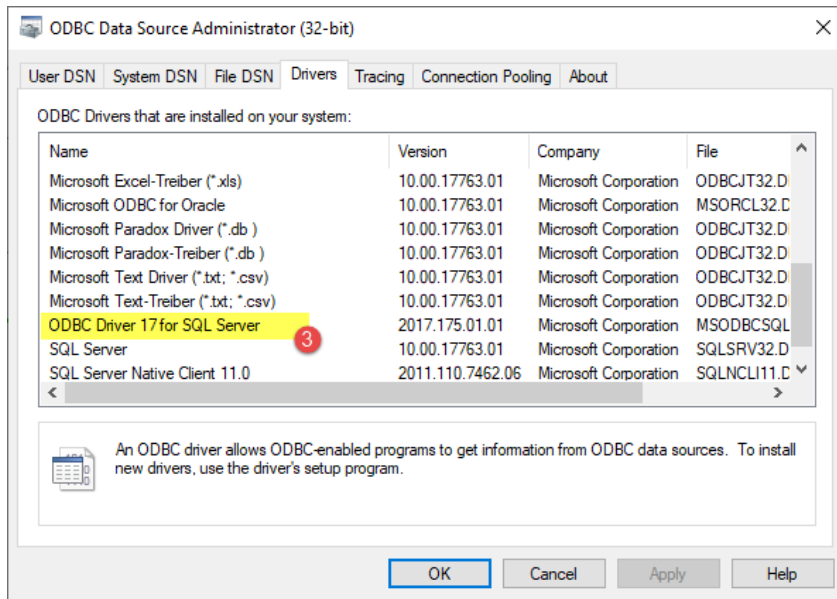
Parameter	Description
<b>Name</b>	Define the name to use to reference the database defined in this item.
<b>Type</b>	Select: ODBC
<b>Connection Type</b>	Select: DSN
<b>DSN</b>	Select the DSN to use
<b>ODBC Database</b>	MS Access
<b>Description</b>	This is just a reminder (write here what you prefer)

## Configuring DSN on Windows PC

The "*ODBC Data Source Administrator*" system tool can be used to configure DSN elements or to retrieve the installed driver names.

1. Name of the DSN element
2. Driver name to type using "MS Access"
3. Driver name to type using "MSSQL Server"





## Configuring ODBC on HMI Runtime

Open Database Connectivity offers different types of connections

### Accessing to Microsoft databases using MSSQL Server

Property	Value
Name	Link1
Type	ODBC
IPAddress	10.1.35.117
Port	1433
Connection Type	Driver
ODBC Database	MSSQL Server
Driver Name	libtdsodbc.so.0
Username	Smith
Password	*****
Database	testDB
Description	

Parameter	Description
<b>Name</b>	Define the name to use to reference the database defined in this item.
<b>Type</b>	Select: ODBC
<b>IPAddress</b>	Server IP Address
<b>Port</b>	Port used from MSSQL Server (generally 1433)
<b>Connection Type</b>	Select: Driver
<b>ODBC Database</b>	Select: MSSQL Server
<b>Driver Name</b>	Type a driver name installed on the HMI device: " <i>libtdsodbc.so.0</i> "
<b>Username</b>	Username for accessing the MSSQL database

Parameter	Description
Password	Password for accessing the MSSQL database
Database	The name of the database you want to access
Description	This is just a reminder (write here what you prefer)

### Accessing to Microsoft databases using MSSQL Server and DSN



This is a special configuration, for advanced users only, which requires the modification of internal files of the HMI device

Property	Value
Name	Link1
Type	ODBC
Connection Type	DSN
DSN	myODBC
ODBC Database	MSSQL Server
Username	Smith
Password	••••
Description	

Parameter	Description
Name	Define the name to use to reference the database defined in this item.
Type	Select: ODBC
Connection Type	Select: DSN
DSN	Select the DSN to use
ODBC Database	Select: MSSQL Server
Username	Username for accessing the MSSQL database
Password	Password for accessing the MSSQL database
Description	This is just a reminder (write here what you prefer)

### Accessing to an internal "MS Access" file

This configuration is not fully supported on HMI devices. MS Access databases can only be read.

Property	Value
Name	Link1
Type	ODBC
Connection Type	Driver
ODBC Database	MS Access
Driver Name	libmdbodbc.so
MS AccessFile	/mnt/data/hmi/Database1.mdb
Description	

Parameter	Description
<b>Name</b>	Define the name to use to reference the database defined in this item.
<b>Type</b>	Select: ODBC
<b>Connection Type</b>	Select: Driver
<b>ODBC Database</b>	Select: MS Access
<b>Driver Name</b>	Type a driver name installed on the HMI device: " <i>libmdbodbc.so</i> "
<b>MS AccessFile</b>	Type the MS Access file to use
<b>Description</b>	This is just a reminder (write here what you prefer)

### Accessing to an internal "MS Access" file using DSN

This configuration is not fully supported on HMI devices. MS Access databases can only be read.



**This is a special configuration, for advanced users only, which requires the modification of internal files of the HMI device**

Property	Value
Name	Link1
Type	ODBC
Connection Type	DSN
DSN	myODBC
ODBC Database	MS Access
Description	

Parameter	Description
<b>Name</b>	Define the name to use to reference the database defined in this item.
<b>Type</b>	Select: ODBC
<b>Connection Type</b>	Select: DSN
<b>DSN</b>	Select the DSN to use
<b>ODBC Database</b>	MS Access
<b>Description</b>	This is just a reminder (write here what you prefer)

### Configuring DSN on HMI device

There are two files of the UnixODBC Driver manager to write and copy to `/mnt/data/hmi/qthmi/deploy/data`

- `odbc.ini`
- `odbcinst.ini`

### DNS configuration to support MS SQL Server

#### `odbcinst.ini`

```
[FreeTDS]
Description = v0.91 with protocol v7.2
Driver = /mnt/data/hmi/qthmi/deploy/libtdsodbc.so.0
```

**odbc.ini**

```
[Datasourcename]
Driver = FreeTDS
Description =
Trace =
Server =
Port =
TDS version =
Database =
```

**Example:**

The screenshot shows two side-by-side text editors. The left editor, titled 'odbcinst.ini', contains the following content:

```
1 [FreeTDS]
2 Description = v0.91 with protocol v7.2
3 Driver = /mnt/data/hmi/qthmi/deploy/libtdsodbc.so.0
4
```

The right editor, titled 'odbc.ini', contains the following content:

```
1 [myODBC]
2 Driver = FreeTDS
3 Description = Any description
4 Trace = No
5 Server = 192.168.20.189
6 Port = 1433
7 TDS version = 7.2
8 Database = Database1
9
```

**DNS configuration to support MS Access****odbcinst.ini**

```
[MDBTools]
Description=MDBTools Driver
Driver=/mnt/data/hmi/qthmi/deploy/libmdbodbc.so
```

**odbc.ini**

```
[MSAccess]
Driver = MDBTools
Description =
Database =
Server =
```

**Example:**

The screenshot shows two side-by-side text editors. The left editor, titled 'odbcinst.ini', contains the following content:

```
1 [MDBTools]
2 Description=MDBTools Driver
3 Driver=/mnt/data/hmi/qthmi/deploy/libmdbodbc.so
4
```

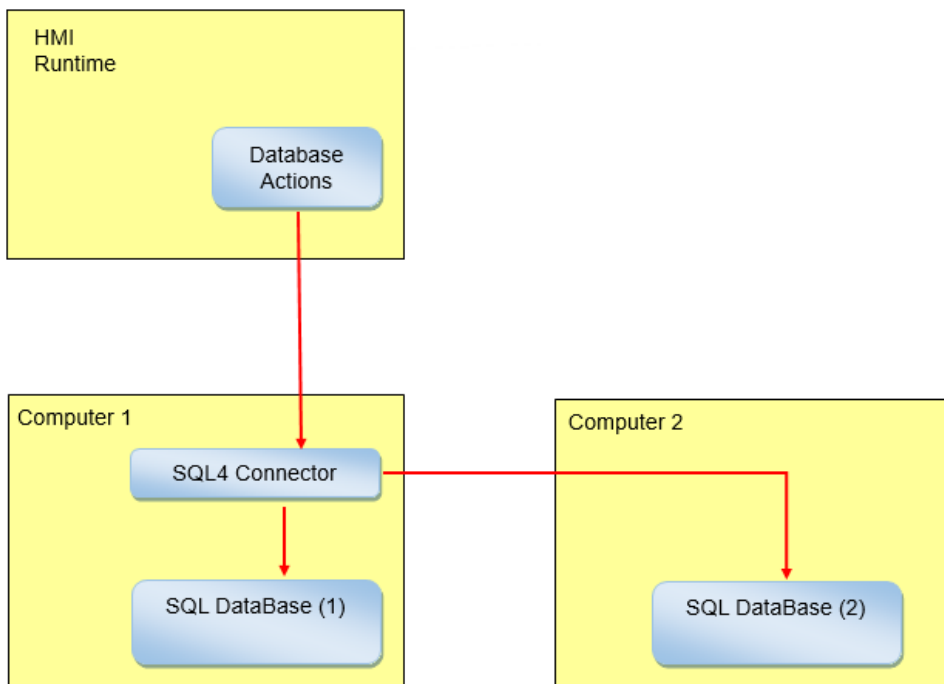
The right editor, titled 'odbc.ini', contains the following content:

```
1 [myODBC]
2 Driver = MDBTools
3 Description = Sample Database
4 Database = /home/admin/Database1.mdb
5 Server = 127.0.0.1
6
```

## SQL4Automation (legacy)

PB610 Panel Builder 600 allow to connect to the SQL4Automation Connector, a software solution for the industrial usage. It connects HMI, PLC and robotic controls directly with SQL databases. HMI directly access SQL databases via the connector and can query data from tables, insert, change and delete data in tables by using SQL commands [structured query language].

The database site communicates by ODBC. Therefore all SQL databases can be integrated, which support an ODBC interface. The SQL syntax needs to be adapted to the given database, e.g. MS SQL Server, MySQL, MS Office Access, SQLite, Oracle, PostgreSQL...



To store data into an external database:

1. Install the SQL4Automation tool on the computer hosting the database or in a computer between the HMI device and the database.
2. Configure the SQL4Automation tool.
3. Create a project that use the dedicated DB actions to access at the external database.

## Installing SQL4Automation

Download the latest version of SQL4automation and install it on the computer. Refer to [www.sql4automation.com](http://www.sql4automation.com) for details and download.

Procedures described in this document refer to SQL4Automation Connector Version 3.3.2.0

# Configuring SQL4Automation



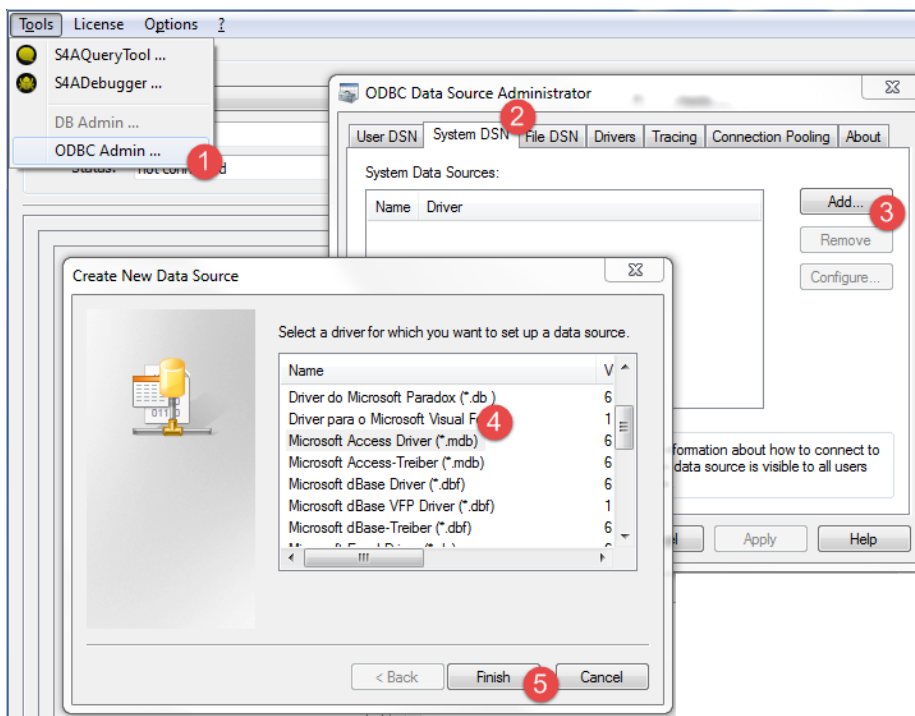
**Important: Refer to the SQL4Automation user manual for detailed configuration instruction.**

Here is a quick description of how to access to a MS Office Database (MS Access).

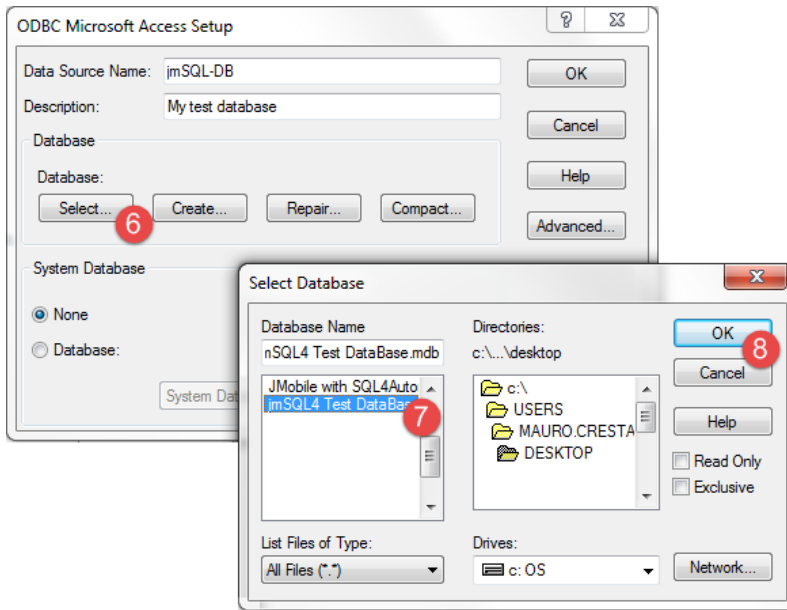
You must have the MS Office Suite installed on a computer and create an empty database using Microsoft Access.

Start SQL4Automation and follow the procedure to configure your SQL4Automation Connector:

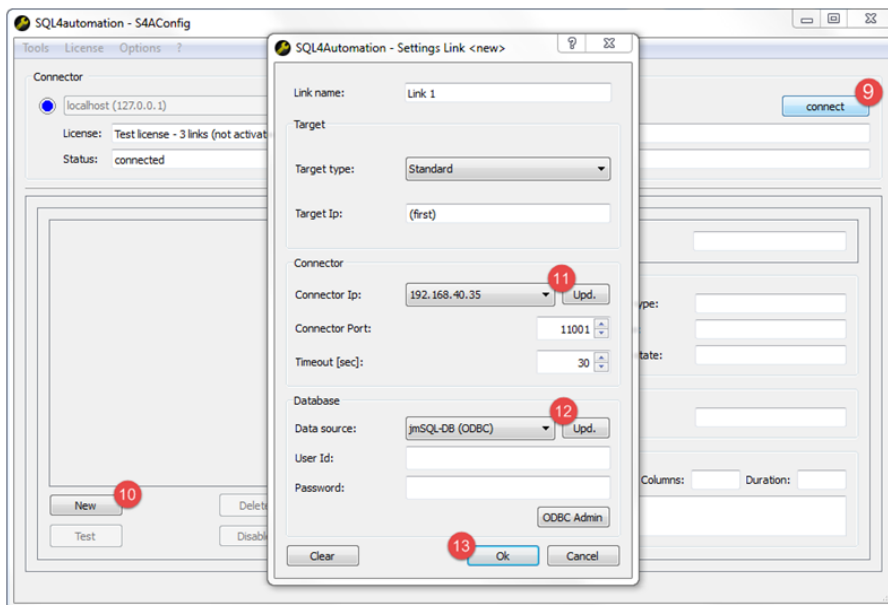
1. Select **ODBC Admin**: the **ODBC Data Source Administrator** dialog is displayed.
2. Select the **System DSN** tab.
3. Click **Add**: the **Create New Data Source** dialog is displayed.
4. Select the Microsoft Access Drive
5. Click **Finish** to confirm.



6. Enter **Data Source Name** and **Description** then click **Select**: the **Select Database** dialog is displayed.
7. Select your Access database.
8. Click **OK** to confirm.

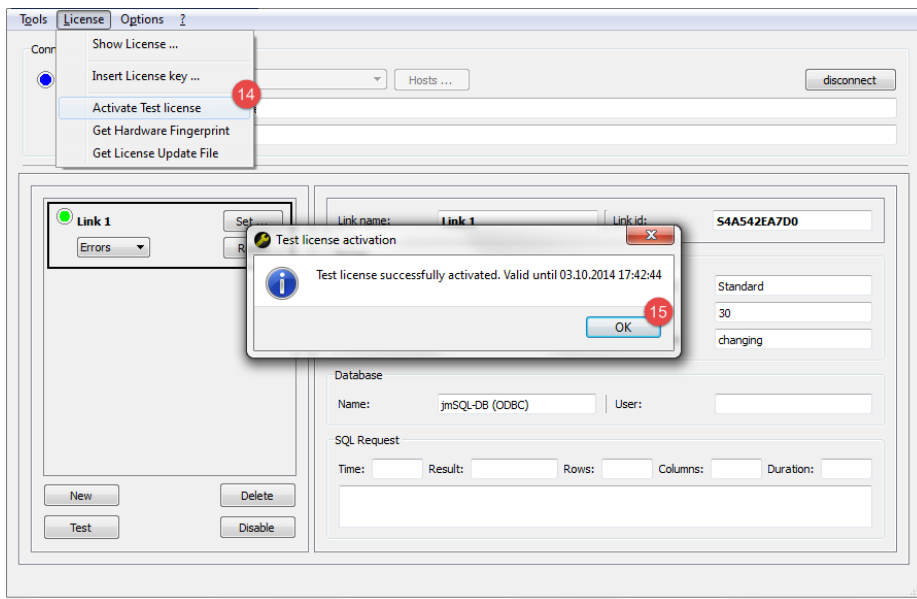


9. At the first connection, click **New** to select your Data Source
10. Select the IP address of your computer. This will be the connection IP Address used from your HMI device.
11. Select the Data Source.
12. Click **OK** to confirm.
13. Click **Connect**.



14. Select **License> Activate Test License**: when the **Link 1** led turns green the procedure has been completed correctly.
15. Click **OK** to confirm.

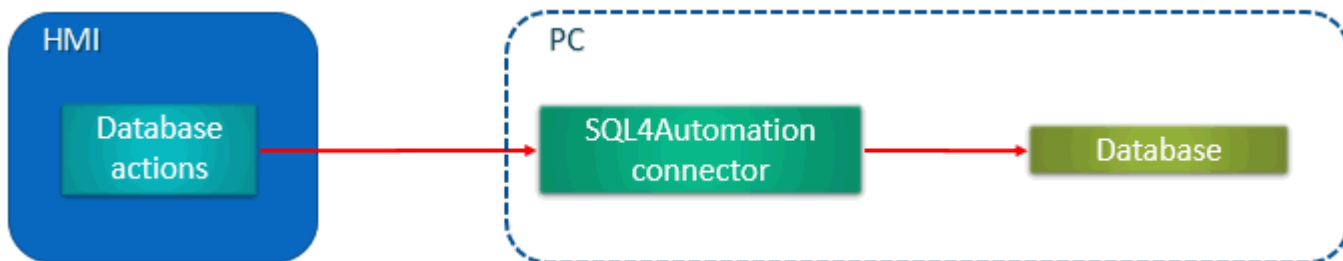




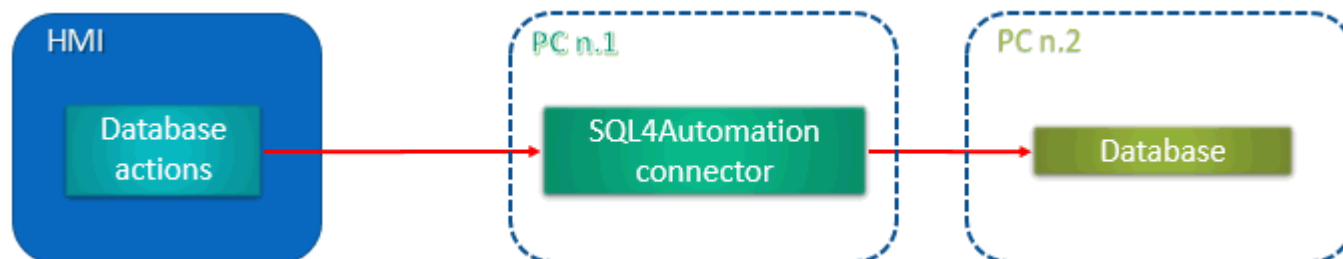
## Connection Limits

SQL4Automation is delivered as a USB dongle with a license for a predefined number of connections. Connections are called "Links" inside SQL4Automation Connector. The number of connections depends on the license you have purchased.

SQL4Automation Connector can be installed on the same Computer/Server running the database:



Or onto a separated Computer/Server:



# 36 OPC UA Server

Path: **ProjectView** > **Config** > **Interfaces** > double-click **OPC UA**


Use OPC UA Server to publish data according to the OPC UA standard.




Available only on Linux devices (See: ["HMI devices capabilities" on page 613](#))

Parameter	Description
Enable OPC UA Server	Main flag to activate OPC UA Server. Data values defined in the HMI device are published by the OPC UA Server.

## Features


Parameter	Description
Enable alarms	Activates publication of real-time alarms data (active alarms).
Enable historical alarms	Activates publication of historical alarms data.  Only default "Alarmbuffer1" is exposed as historical alarm data. (See <a href="#">"Events Buffer" on page 281</a> )
Enable trends	Activates publication of trends data.
Tag groups	Only tags belonging to selected groups will be available to the OPC UA Server.
Alarm groups	Only alarms belonging to selected groups will be available to the OPC UA Server.

## Network


Parameter	Description
Node Name	Enter node name or leave empty to use host name.
Port	Port number of OPC UA Server.  Port number proposed as default may be different from port used by OPC UA Client.

## Authentication

Select authentication options for OPC UA Server.


 OPC UA Clients will be responsible for choosing, from available options, the most appropriate option to use according to their capabilities.

### User authentication


Parameter	Description
<b>Anonymous</b>	Anonymous clients accepted.
<b>User/Password</b>	Authentication with user name is accepted.   Any valid user has unrestricted access to OPC UA Server (see " <a href="#">Configuring users</a> " on page 371).


## Using x.509 Certificates

OPC UA provides a secure communication channel using digital certificates. Configurable levels of end-to-end security ensuring encryption, confidentiality and integrity of each message are available. When enabled, the server validates the client certificate and vice versa.

 OPC UA Clients will be responsible for choosing, from available options, the most appropriate option to use according to their capabilities.

Security Mode	Description
<b>Policy</b>	Select the acceptable Security Policies (see the next table)
<b>Sign</b>	OPC UA Client must provide its own certificate: communication through signed messages is allowed.
<b>SignAndEncrypt</b>	OPC UA Client must provide its own certificate: communication through signed and encrypted messages is allowed.

Security Policy	Description
<b>None</b>	 Not recommended in public networks.
<b>Basic128Rsa15</b>	Accepted encryption level.
<b>Basic256</b>	Accepted encryption level.
<b>Basic256Sha256</b>	Accepted encryption level.

Parameter	Description
<b>Automatically trust any new clients</b>	All certificates provided from any OPC UA Clients are accepted.
<b>Trusted Certificates</b>	<p>Only OPC UA Clients that provide one of the listed certificates are accepted.</p> <p> To add a new certificate to the list of trusted certificates, you must have the certificate file supplied by the owner of the OPC UA Client device. Both binary and ASCII certificate file formats are accepted.</p>

## Global Discovery Server

OPC UA Server is compatible with the GDS Push Model. This means that you can use a remote GDS tool for central certificate management.



**To be able to successfully connect to OPC UA Server, you must retrieve the certificate of the GDS tool and add it to the trusted certificate list of OPC UA Server.**

Example

When an OPC UA Client attempts a connection with the OPC UA Server, the server checks if the client certificate is available inside its own trusted certificate list. If it is not found, the communication will be rejected and the certificate will be stored in a list of unreliable certificates.


Using a GDS tool, you can connect to the OPC UA Server, inspect available certificates and define trust or not trust state of each certificate.


## Certificate Files

HMI device will store certificates inside the subfolders of folder `"/workspace/<ApplicationName>/config/pkiserver"`

- own  
Own certificate and private key
- trusted  
Trusted self signed certificates and CA certificates
- rejected  
Rejected certificates
- issuers  
Trusted intermediate (not directly trusted) CA required to validate the trust chain

## Server Identity

Parameter	Description
<b>Manufacturer name</b>	<p>Human readable name of the manufacturer of the product.</p> <p> OPC UA Client can retrieve this information from tag: <code>ServerName Objects Server ServerStatus BuildInfo ManufacturerName</code></p>
<b>Product name</b>	A human readable name for the product running in the server.

Parameter	Description
	 The OPC UA Client can retrieve this information from tag: <code>ServerName Objects Server ServerStatus BuildInfo ProductName</code>

## Certificate Parameters




Server certificate can be either generated automatically or by adding an existing certificate file.

### Automatically generate self-signed certificate


If auto generated certificate is enabled the certificate is regenerated after every change made by user to certificate parameters. The certificate is also replaced by any explicitly set certificate.

### Certificate parameters

Each certificate must contain information that should identify the certificate and its restrictions. If you have chosen to use a self-generated certificate, enter the information you want to be inside the self-generated certificate. Otherwise parameters are read from the certificate you have supplied.

Parameter	Description
<b>Server Name (Common Name)</b>	Name of the certificate (e.g. the device name).
<b>Organization</b>	Organization name
<b>Unit</b>	Organization unit  This field could be useful to differentiate different divisions within an organization.
<b>Location</b>	Locality field denotes the city where organization resides in
<b>State</b>	State or Province field specifies where the organization is physically located.  Content of State or Province field should not be abbreviated. For example, "CA" is not a valid state name. "California" is the proper state name.
<b>Country</b>	The X.509 naming scheme standard requires a 2-character country code.  Country code for the United States is US; country code for Italy is IT.

Parameter	Description
<b>Product URI</b>	A globally unique identifier for the server. Example: <code>"urn:NodeName:CompanyName:ServerName"</code>
<b>DNS Names</b>	DNS name or IP Address of the device where this OPC UA Server is installed.

Parameter	Description
IP Addresses	Multiple DNS Names and/or IP Addresses can be in a single certificate.  <b>The certificate will be valid only if the IP address where the OPC UA Server is running is included in this list.</b>
Validity	Period of validity of the certificate starting from creation date
Key Length	Length of the key used by RSA encrypting algorithm

### Script to generate a Certificate

If you want provide your own certificate, note that the certificate must include the “Subject Alternative Name (SAN)” parameters as required by the OPC UA standard.

Here is an example of how to generate a certificate using a public OpenSSL-Win32 library (Reference: <https://www.openssl.org/>)

```
@echo off
set OpenSSL="C:\Program Files (x86)\OpenSSL-Win32\bin\openssl.exe"
set NodeName=HMI-Server
set IPAddress=192.168.44.165

rem Generate an RSA key
%OpenSSL% genrsa -out server-key.pem 2048

rem Creating Certificate Signing Requests
%OpenSSL% req -new -key server-key.pem -out server.csr -subj "/ST=NY/C=US/L=New
York/O=CompanyName/OU=R&D Team/CN=OPCUAServer@%NodeName%"

rem Creating Certificate (.pem)
echo subjectAltName=URI:urn:%NodeName%:CompanyName:OPCUAServer,IP:%IPAddress% >
san.txt
echo
keyUsage=digitalSignature,nonRepudiation,keyEncipherment,dataEncipherment,keyCertSign
>> san.txt
echo extendedKeyUsage=critical,serverAuth,clientAuth >> san.txt
echo authorityKeyIdentifier=keyid,issuer >> san.txt
echo basicConstraints=CA:TRUE >> san.txt
%OpenSSL% x509 -req -days 3650 -in server.csr -signkey server-key.pem -out
server.crt -extfile san.txt

rem Convert Certificate (.der)
%OpenSSL% x509 -in server.crt -outform der -out server.der

rem Not necessary files
del san.txt

pause
```

# Using self-signed certificates

This chapter is a step by step example that explains how to configure two HMI devices to communicate using self-signed certificates

## OPC UA Server

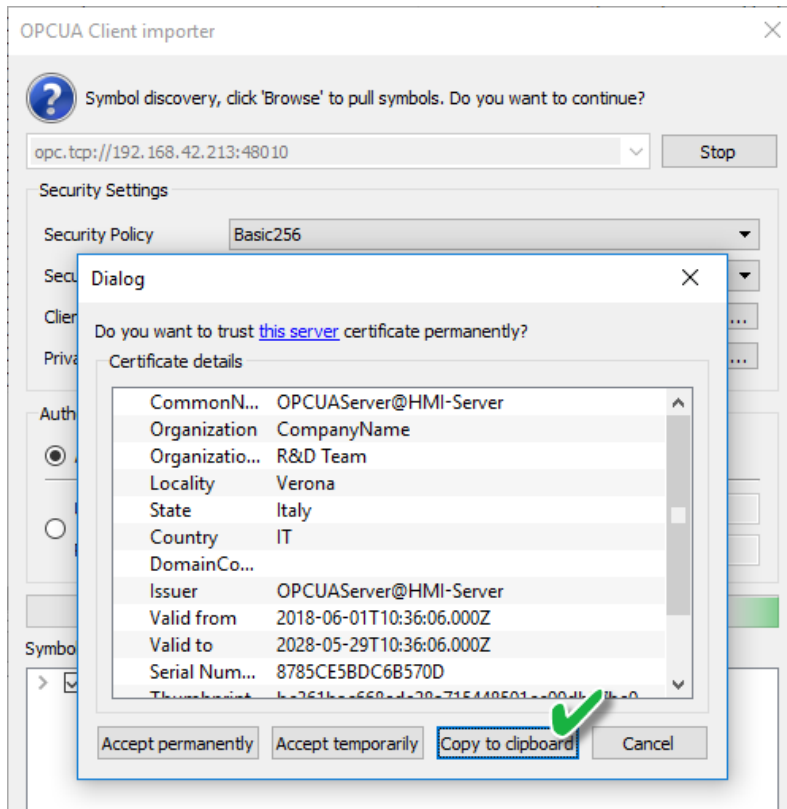
1. Create a simple project including a few tags
2. Open the OPC UA dialog and enable OPC UA Server. Be sure to enable tag groups (e.g. select "All")
3. Enter in "IP addresses field" the IP address of the HMI device where OPC UA Server will run

The screenshot shows the OPC UA Server configuration dialog. The 'Automatically generate self signed certificate' checkbox is checked. The 'IP addresses' field is highlighted in yellow and contains the value '192.168.44.165'. Other fields include Organization, Unit, LocationName, State, Country (DE), DNS names, Validity (5 years), and Key length (1024). There are also fields for Certificate and Private key with browse buttons.

4. Download the project to the HMI device

## OPC UA Client

5. Create a simple project
6. Add the OPC UA Client protocol. Enter the IP address of the remote OPC UA server and its port number (48010). Leave certificate parameters empty.
7. Open tag editor and import tags. Select "OPC UA Discovery" mode
8. Choose to copy the certificate to the clipboard as shown in the figure. Then, close this dialog and return to protocol configuration dialog to paste the certificate inside the "Server Certificate" field.



9. Repeat step 7, accept the Server OPC UA certificate and import some tags. Note that you can accept the certificate permanently or temporarily. If you accept the certificate permanently, a copy of the certificate will be saved inside your computer for later use without popup again the dialog to asking for confirmation.



The certificate file will be copied inside the folder:  
 %AppData%\Roaming\ABB\studio\OPCUA\pki\trusted\certs

10. Open again the protocol dialog box. Select the Security Policy = Basic256 and Security Mode = SignAndEncrypt
11. Download the project to the HMI device

Since in the OPC UA Client protocol parameters we left empty the “Client certificate” field, the OPC UA Client protocol has generated its own certificate and sent it to the OPC UA Server but since the server does not know this certificate it rejects the connection request. Now we have to tell the server to trust these certificates. There are different ways to do it.

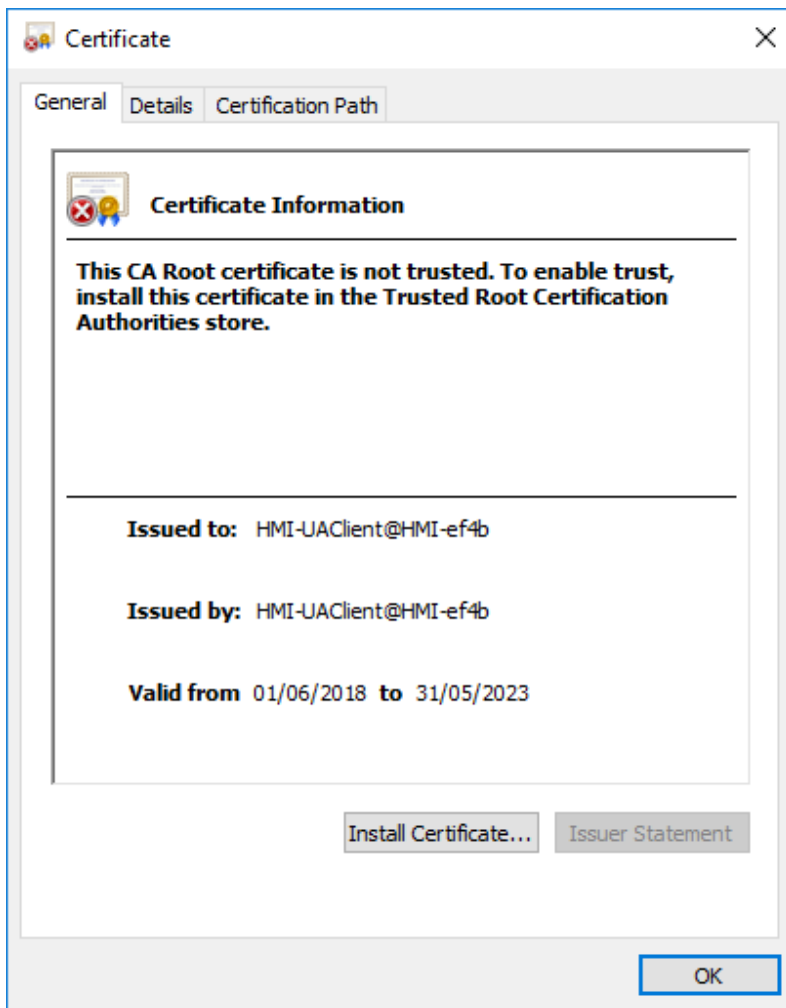
#### Make rejected certificate trusted using FTP client:

1. Connect to OPC UA device using an FTP client
2. Look inside the certificate folders and move the rejected certificate from the rejected folder to the trusted folder.

```
/workspace/<YourProjectName>/config/pkiserver/rejected
/workspace/<YourProjectName>/config/pkiserver/trusted/certs
```

You can double click the certificate file to open it and look to certificate parameters to be sure about the certificate you are validating





Now the communication will start

#### Make rejected certificate trusted using GDS tool:

1. Open the GDS tool and export its certificate
2. Open the project and add the certificate of the GDS tool to the Trusted Certificate list
3. Download the updates project to the HMI device

Now you can manage certificates using the tools in the HMI device.

## Using external certificates

This chapter is a step by step example explaining how to configure two HMI devices to communicate using external certificates.

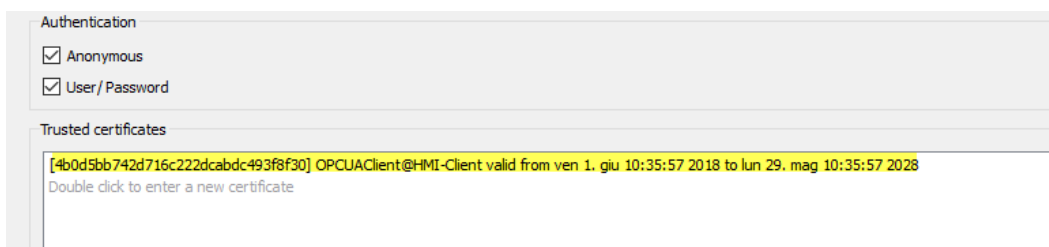
#### Generate certificates

You can use the script given in this manual to generate a copy of your own certificates, one for OPC UA Server and another one for OPC UA Client.

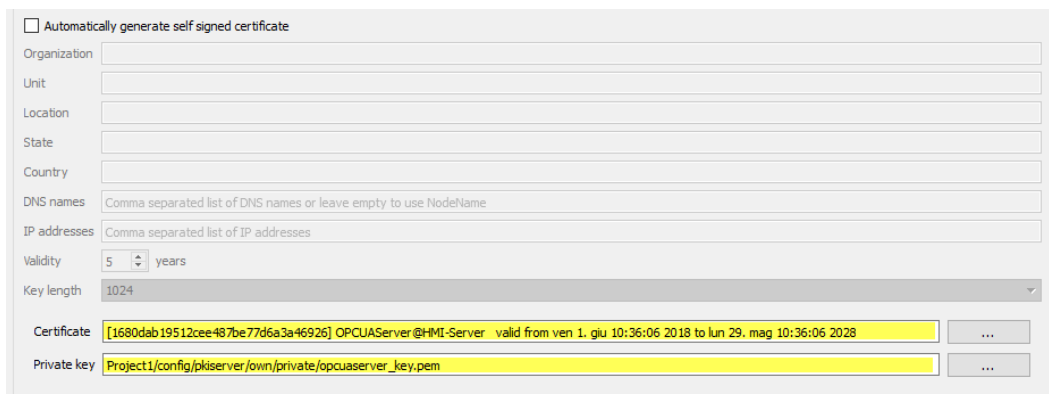
1. Install a OpenSSL-Win32 library (Reference.: <https://www.openssl.org/>)
2. Use the script ("[Script to generate a Certificate](#)" on page 435) to generate OPC UA Server certificate. Be sure to set the IPAddress variable with the IP Address of the HMI device where OPC UA Server will run before running the script.
3. Find in the OPC UA Client protocol manual a sample script to generate a certificate for the OPC UA Client protocol

### OPC UA Server

4. Create a simple project that using a few tags
5. Open the OPC UA dialog and enable OPC UA Server. Be sure to enable tag groups (e.g. select "All")
6. Add the client.der certificate to the Trusted Certificate area to enable the OPC UA Client to communicate with OPC UA Server



7. Remove the check on "*Automatically generate self-signed certificate*" and add the server certificate (server.der) and the server private key (server-key.pem)



8. Download the project to the HMI device

### OPC UA Client

9. Create a simple project
10. Add OPC UA Client protocol.
11. Enter the IP address of the remote OPC UA server and its port number (48010).
12. Open the ASCII version of the server certificate (server.crt), remove all Newline characters and then copy and paste the ASCII characters of your certificate inside the Server Certificate field.
13. Repeat the same with Client Certificate (client.crt) and Client private key (client-key.pem)
14. Select the Security Policy Basic256 and the Security Mode = SignAndEncrypt
15. Open tag editor and import tags. Select "OPC UA Discovery" mode
16. Accept OPC UA Server certificate, import some. Note that you can accept the certificate permanently or temporarily. If you accept the certificate permanently, a copy of the certificate is saved inside your computer to


later usage without asking you for confirmation

17. Download the project to the HMI device

You will note that OPC UA Client is retrieving data from OPC UA Server using the given certificates.

## Alarm map

The alarm states are mapped to OPC UA states according to the following rules:

OPC UA Alarm state	PB610 Panel Builder 600 Alarm state
<b>Opcua.Alarm.Active</b>	True when alarm state is triggered <ul style="list-style-type: none"> <li>• TRIGGERED</li> <li>• TRIGGERED_NOT_ACKED</li> <li>• TRIGGERED_ACKED</li> </ul>
<b>Opcua.Alarm.Acked</b>	True when alarm acknowledgment is not required <ul style="list-style-type: none"> <li>• TRIGGERED_ACKED</li> <li>• NOT_TRIGGERED_ACKED</li> <li>• NOT_TRIGGERED</li> </ul>
<b>Opcua.Alarm.Retain</b>	True when alarm is pending <ul style="list-style-type: none"> <li>• TRIGGERED  </li> <li>• TRIGGERED_NOT_ACKED  </li> <li>• TRIGGERED_ACKED  </li> <li>• NOT_TRIGGERED_ACKED but a RESET is required</li> </ul>
<b>Opcua.Alarm.Confirmed</b>	True when alarm is returned (Not triggered, acknowledged and reset) <ul style="list-style-type: none"> <li>• NOT_TRIGGERED</li> </ul> <p> This info is available only when alarm is configured to required a RESET</p>

# 37 MQTT Interface

Path: **ProjectView**> **Config** > **Interfaces** > double-click **MQTT**

Use MQTT Interface to publish data according to the MQTT standard.



**Available only on Linux devices (See: "HMI devices capabilities" on page 613)**

Note that a tag or an alarm, to be transferred through the MQTT protocol, must be defined within a group.

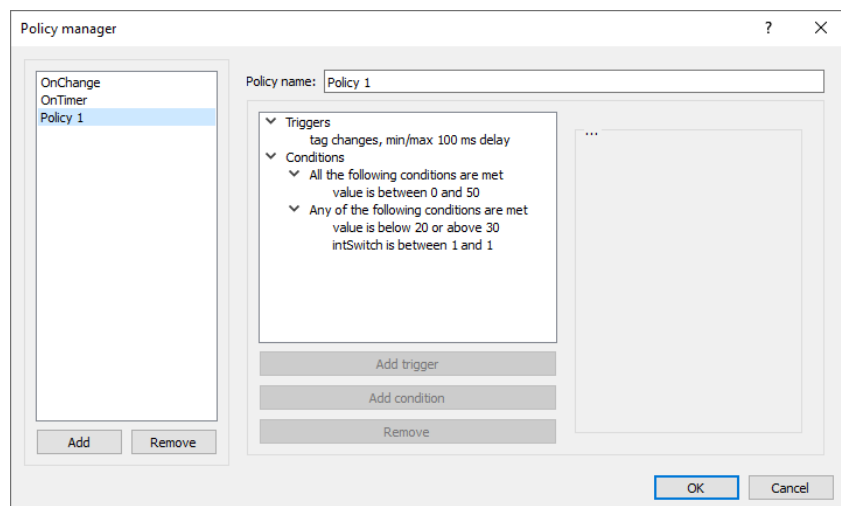
Parameter	Description
<b>Enable MQTT Interface</b>	Main flag to activate MQTT service. The selected groups of tags will be published to the MQTT broker.
<b>Enable Alarms</b>	The selected groups of alarms will be published to the MQTT broker. Alarms are published whenever there is a change in the alarm status.

## Tags configuration

Parameter	Description
<b>Enable</b>	Enable tags included within the group to trigger the publication "Data (Pub)". The tags included in the group can be updated by the subscription "Data (Sub)".
<b>Tag Group</b>	List of tags that will be transferred when the assigned policy conditions will be satisfied.
<b>QoS</b>	QoS to use  0 = Delivered at most once (Fire and forget) which means no confirmation  1 = Delivered at least once, which means confirmation required  2 = Delivered exactly once, which means a 4 step handshake is done
<b>Retain</b>	This flag defines whether the message is saved by the broker as the last known good value for a specified topic. When a new client subscribes to a topic, they receive the last message that is retained on that topic.
<b>Persistence</b>	When true, the messages with QoS greater than 0 are queued into the file system file to be available even after a panel reset or when a broken communication with the MQTT server is reestablished.
<b>Policy</b>	Defines the criteria for deciding when to publish the value of a tag. When it is empty, the "Default push policy", defined on top of the table, is used.

## Manage push policy

A policy consists of a trigger criterion and several (optional) conditions that must be verified in order for the tag value to be transmitted.



### Trigger

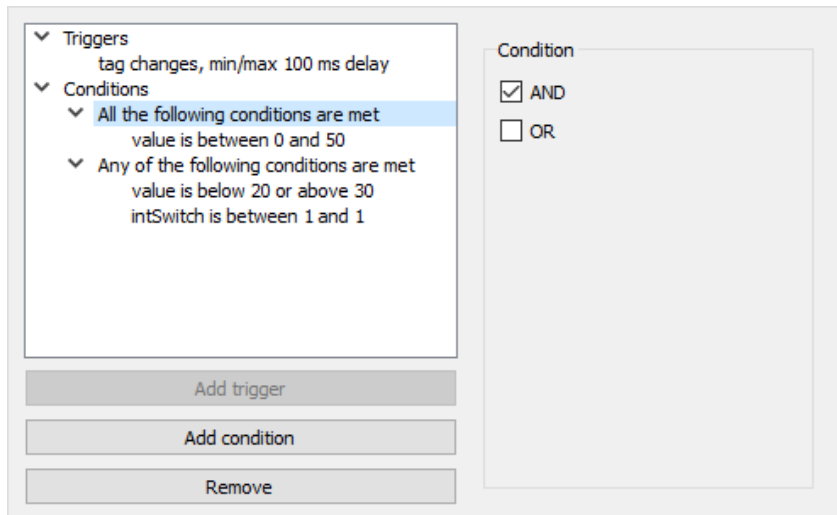
Parameter	Description
<b>Timer</b>	<p>Publish is performed continuously even value is not changing.</p> <ul style="list-style-type: none"> <li>Interval (ms) Cyclical publication time</li> </ul>
<b>On change</b>	<p>Publish is performed when a tag value changes.</p> <ul style="list-style-type: none"> <li>Min interval (ms) Value check interval</li> <li>Deadband The difference, from previous publish, that must be found to trigger the new publish.</li> <li>Use percentual Dead band value express in percentage</li> <li>Tag Name Tags to be checked to activate the publication. If empty, the tag to be published is used.</li> </ul>

### Conditions

Conditions contain folders of conditions. Each folder can be of two types:

- All the following conditions are met (AND)
- Any of the following conditions are met (OR)

All folders must be validated to have the transmission requested by the trigger. A folder of type "All the following condition are met" is validated when all the contained conditions are true while a folder of type "Any of the following conditions are met" is validated when at least one contained condition is true.



## Tags interface

The "Tags Interface" allows you to define tags that will be used to exchange information about the status of the MQTT interface.

Parameter	Description	Data Type
<b>Enable</b>	Control the status of the MQTT interface.  If this parameter is empty (not attached to a tag), the MQTT interface will be always enabled. <ul style="list-style-type: none"> <li>• 0 = Disabled</li> <li>• 1 = Enabled</li> </ul>	boolean  Read Write
<b>Status</b>	Report the status of the MQTT interface. <ul style="list-style-type: none"> <li>• 0 = Disconnected</li> <li>• 1 = Connected</li> <li>• 2 = Connecting</li> <li>• 3 = Disconnecting</li> </ul>	unsignedInt  Read Only
<b>Last error</b>	In case of errors, it will report the error message.  You can write an empty string to clear the error message.	string  Read Write
<b>Messages sent</b>	Amount of messages sent from the HMI Device to the remote MQTT Server.	unsignedInt  Read Only
<b>Messages received</b>	Amount of messages received by the HMI Device from the remote MQTT Server.	unsignedInt  Read Only
<b>Queue length</b>	The number of pending messages to be sent that	unsignedInt

Parameter	Description	Data Type
	are temporarily cached in the local buffer.	Read Only
<b>Queue size</b>	The maximum number of messages that can be temporarily cached in the local buffer.	unsignedInt Read Only

## Settings

Topic	Description
<b>Max pending messages</b>	The number of messages that can be queued in RAM when there are communication errors. Queue messages will be released as soon as the MQTT Server returns reachable.
<b>Defaults</b>	Values of "QoS", "Retain" and "Persistence" parameter to use for the topics that are not defined inside the "Tags configuration" table.

## Topic and Payload

There are five types of supported topics:

Topic	Description
<b>Birth</b>	This is a special topic that is publish only one time when HMI device start.
<b>Will</b>	Special topic that is published when device starts but stored and kept hidden by the MQTT Broker. It will be published by the MQTT Broker if it detects that the client has disconnected ungracefully.
<b>Data (Pub)</b>	Topic is used to publish the tags' values following the transmission policies associated with tag groups.
<b>Data (Sub)</b>	Topic is used to subscribe to tags. The payload is the template used to recognize the values of the received tags.
<b>Alarm</b>	Topic used to publish alarms

Birth
Will
Data (Pub)
Data (Sub)
Alarm

For each topic, the payload defines the structure of the associated value. Note that in topic and payload definitions can be used placeholders.

Placeholder	Description
<code>#{clientId}</code>	MQTT Client ID
<code>#{timestamp}</code>	Return the time when the tag got updated in UTC format
<code>#{localTimestamp}</code>	Return the time when the tag got updated in local format
<code>#{currentTimestamp}</code>	Return the time when the MQTT message publish
<code>#{protocolName}</code>	Name of the protocol associated to a tag
<code>#{tagGroup}</code>	Name of the group the tag belongs to
<code>#{tagName}</code>	Name of the tag
<code>#{alarmGroup}</code>	Name of the group the alarm belongs to
<code>#{alarmName}</code>	Name of alarm
<code>#{value}</code>	Last known value of the tag
<code>#{activeValue}</code>	Value of the tag when the alarm became active
<code>#{quality}</code>	Quality (i.e. reliability) of the tag
<code>#{activeTimestamp}</code>	Timestamp of the last event that raised the alarm
<code>#{inactiveTimestamp}</code>	Timestamp of the last event that ceased the alarm condition
<code>#{ackTimestamp}</code>	Timestamp when the operator acknowledge the alarm
<code>#{description}</code>	Alarm description
<code>#{customField1}</code>	Alarm Custom Field 1
<code>#{customField2}</code>	Alarm Custom Field 2
<code>#{state}</code>	Alarm State
<code>#{severity}</code>	Alarm Severity
<code>#{lowLimit}</code>	Alarm "Low limit"
<code>#{highLimit}</code>	Alarm "High limit"
<code>#{[0]}</code>	If available in the alarm description, the value of the first live tag, [1] the second, etc.
<code>#{[Tag1]}</code>	If available in the alarm description, the value of "Tag1" live tag
<code>#{any}</code>	A generic label. Useful in the subscribe payloads

### JSON Payload



When the **JSON format** is selected, the quotation marks are added around string values to conform to the JSON syntax.

With the use of the JSON format is possible to optimize the communication to include multiple messages inside a single message. When the **"Message aggregation"** is selected, the messages are sent to MQTT Server after the selected timeout expired or when the message to send reaches the select size.

## Multiple Tags

Using JSON format, in the subscribe/publish topics is possible to manage multiple tags as for the following examples:

To match an incoming message like:

```
{ "x": { "tagName": "Tag1", "value": 1 }, "y": { "tagName": "Tag2", "value": 2 } }
```

you can use pattern:

```
{ "${any}": { "tagName": "${tagName}", "value": "${value}" } }
```

To match an incoming message like:

```
[ { "tagName": "Tag1", "value": 1 }, { "tagName": "Tag2", "value": 2 } ]
```

you can use pattern:

```
[ { "tagName": "${tagName}", "value": "${value}" } ]
```

To match an incoming message like:

```
{ "Tag1": { "value": 1 }, "Tag2": { "value": 2 } }
```

you can use pattern:

```
{ "${tagName}": { "value": "${value}" } }
```

To match an element, or all elements, of an array incoming message like:

```
{ "Tag1": { "value": "[1, 2, 3, 4, 5, 6, 7, 8]" } }
```

you can use pattern:

```
{ "${tagName}": { "value": "${value.1}" } }
```

```
{ "${tagName}": { "value": "${value}" } }
```

For published topics, a tag/group of tags is required to establish the trigger condition but the message can now contain other tag values using **live tag** syntax. The trigger tag can even be omitted in the payload pattern.

Examples:

```
{ "tag1" : "${[Tag1]}", "tag2" : "${[Tag2]}" }
```

For subscribed topics, the same **live tag** syntax can be used to write any tag value (note that only tags listed from the trigger condition can be written).

Examples:

```
{ "tag1" : "${[Tag1]}", "tag2" : "${[Tag2]}" }
```

```
{ "t" : "${tagName}", "v" : "${value}", "tag2" : "${[Tag2]}" }
```


## MQTT Broker Settings

Current supported MQTT Broker are:

- Generic MQTT broker
- Azure
- Amazon AWS
- IBM BlueMix
- Murano

Note that some parameters depend on the broker has chosen.

### Generic MQTT Broker

Parameter	Description
<b>Broker address</b>	Name or IP address of the MQTT server with the port number (e.g. "127.0.0.1:1883")  Generally, the default TCP/IP port is 1883, or the port 8883 when MQTT over SSL is used.
<b>Client ID</b>	The client identifier is an identifier of each MQTT client connecting to an MQTT broker. You can write what you prefer, but it has to be unique per broker. The broker uses it for identifying the client and the current state of the client.
<b>Username Password</b>	If the MQTT broker is configured to require client authentication using a valid user name and password
<b>Keep-alive time (s)</b>	Time interval before sending a PING request to the server when there are no data flows (useful to know if both client and server are still alive and reachable).
<b>Use clean session</b>	When the clean session flag is set to false, the broker creates a persistent session for the client. All information and messages are preserved until the next time that the client requests a clean session. If the clean session flag is set to false and the broker already has a session available for the client, it uses the existing session and delivers previously queued messages to the client.
<b>Use legacy</b>	The "Use legacy" flag makes client comply with MQTT spec 3.1

### Enable TLS

If the MQTT server is configured to works over TLS connection, the HMI device must provide its own certificate to the server. Even it's not mandatory, each client should have its own certificate (however it is possible you can deploy the same certificate to all clients).

It is required that both server and client certificates are signed by the same authority.

Parameter	Description
<b>Enable TLS</b>	Enable the TLS encryption
<b>CA Certificate</b>	Public certificate of the CA that has signed the server certificate on the Mosquitto Broker
<b>Client Certificate</b>	Public certificate of the HMI Device. Must be signed from the CA Certificate
<b>Client Key</b>	Private key associated with the client certificate
<b>TLS Version</b>	TLS Version to use (must be aligned with the encryption level used from the MQTT Broker) <ul style="list-style-type: none"> <li>• tlsv 1</li> <li>• tlsv 1.1</li> <li>• tlsv 1.2</li> </ul>
<b>Insecure</b>	This option disables verification of the server host name in the server certificate. This can be useful when testing initial server configurations but makes it possible for a malicious third party to impersonate your server through DNS spoofing, for example. Use this option in testing only.

Note that you can use the "attach to tag" to entered the MQTT parameters at runtime using, e.g., macros, JavaScript or a configuration page. This could be useful to have different values (e.g. for the ClientID) even downloading the same project to different HMI devices.



**If you use tags to define MQTT settings (e.g. Client ID), be sure to not include these tags into the tags list exchanged with the MQTT server to avoid to receive back wrong settings.**

The string Tags used for the certificate must be great enough (e.g. 2.048 bytes) to contain the entire certificate. The format of the certificates must be ASCII with the first and the last text line included as for the below example.



## MQTT Broker Example

Here is a little example that explains how to configure an application to communicate with an MQTT server. In this example, we configure the HMI device to communicate with an open-source MQTT broker (<https://mosquitto.org>) using certificates. The certificates will be created using a public OpenSSL-Win32 library (<https://www.openssl.org>).

### Generate the certificates

The following script file will create a couple of server and client certificates and a public Certificate Authority that will be used to sign server and client certificates and to verify the authenticity of these certificates.

- ca.crt
- server.crt, server.key
- client.crt, client.ket

File: *CreateCertificates.cmd*

```
@echo off
set OpenSSL="C:\Program Files (x86)\OpenSSL-Win32\bin\openssl.exe"

rem Generate self signed CA certificate (Certificate Authority)
%OpenSSL% req -nodes -batch -new -x509 -days 1000 -keyout ca.key -subj
"/CN=Broker/O=company.com" -out ca.crt

rem Generate MQTT Server private key
%OpenSSL% genrsa -out server.key 2048

rem Generate MQTT Server certificate signed request
%OpenSSL% req -batch -new -key server.key -subj "/CN=localhost/O=company.com" -out
server.csr
```

```

rem Sign the MQTT Server certificate
  %OpenSSL% x509 -req -days 1000 -in server.csr -CA ca.crt -CAkey ca.key -
CAcreateserial -out server.crt

rem Generate HMI Client private key
  %OpenSSL% genrsa -out client.key 2048

rem Generate HMI Client Server certificate signed request
  %OpenSSL% req -batch -new -key client.key -subj "/CN=client/O=company.com" -out
client.csr

rem Sign the HMI Client certificate
  %OpenSSL% x509 -req -days 1000 -in client.csr -CA ca.crt -CAkey ca.key -
CAcreateserial -out client.crt

rem Remove unnecessary files
  del *.rnd *.srl *.csr

pause

```

Note the server hostname is localhost (/CN=localhost), this means that you cannot use the secure connection if in the Broker address parameter you cannot write the "localhost" domain. You can use the "localhost" domain only if both the MQTT Server and the HMI device are running on the same device otherwise, to be able to reach the MQTT server, you must use the IP Address and the "Insecure" flag.

Broker address	192.168.52.41	<input checked="" type="checkbox"/> Enable TLS
Port	8883	CA certificate: pm 14. ago 14:57:39 2022 ... Clear
Client ID		Client certificate: pm 14. ago 14:57:41 2022 ... Clear
Username		Client key: b428568e94e09945f8d81 ... Clear
Password		TLS version: tlsv1.2
Keep-alive time (s)	60	Insecure <input checked="" type="checkbox"/>
Use clean session	<input type="checkbox"/>	
Use legacy	<input type="checkbox"/>	

## MQTT Broker configuration

The server certificate (server.crt, server.key) and the authority certificate (ca.crt) must be placed inside a subfolder of the MQTT folder, e.g. inside the "certs" subfolder.

The "*mosquitto.conf*" file has to be configured to use the TLS support

```

# =====
# Default listener
# =====
... (omiss) ...
# Port to use for the default listener.
#port 1883
port 8883

# -----
# Certificate based SSL/TLS support
# -----

```

```
... (omiss) ...
#cafile
#capath
cafile certs/ca.crt
certfile certs/server.crt
keyfile certs/server.key
tls_version tlsv1.2
```

MQTT Broker can be started using the below command from a dos command window:

```
mosquitto -v -c mosquitto.conf
```

### MQTT Client

For testing purposes, it could be useful to start an MQTT client with the subscription of all the topics so that you can see the messages that will be exchanged with HMI Device. Since we are using TLS communication, we must provide the client certificate. We can copy client.crt, client.key and the authority certificate ca.crt inside the certs-client subfolder.

So the command to activate an MQTT client is:

```
mosquitto_sub --cafile certs-client\ca.crt --cert certs-client\client.crt --key certs-client\client.key -p 8883 -t /#
```

### HMI Device

To configure the HMI device we must provide:

- set the broker address parameter with the IP address where the MQTT server is running
- set the port address to 8883
- load the authority certificates, the client certificate and the client key files
- set the TLS Version to version 1.2 to be aligned with the MQTT server settings
- since it is probably that you are referencing the MQTT server using the IP address, which is different from the domain declared by the server certificate you must set the "Insecure" flag

To perform the first tests, you can leave the default values on topics and payloads and configure the alarms groups and tags groups that you want to transfer to the MQTT broker.

## MQTT Interface

**Features**

Enable MQTT interface

Enable alarms Alarm groups: MQTT

---

**Tags configuration**

Default push policy: OnChange Manage push policies

	Enable	Tag Group	QoS	Retain	Persistence	Policy
1	<input type="checkbox"/>	All	0	<input type="checkbox"/>	<input type="checkbox"/>	
2	<input checked="" type="checkbox"/>	Counter	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
3	<input checked="" type="checkbox"/>	MQTT	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	



# 38 Special widgets

---

Widgets designed for special purposes are called special widgets and include control lists, date and time widgets, variable widgets and so on.

---

<b>BACnet widget</b> .....	<b>454</b>
<b>Canvas Widget</b> .....	<b>454</b>
<b>Combo Box widget</b> .....	<b>457</b>
<b>Consumption Meter widget</b> .....	<b>461</b>
<b>Control list widget</b> .....	<b>463</b>
<b>TabBar widget</b> .....	<b>466</b>
<b>ToolBar widget</b> .....	<b>468</b>
<b>Stack widget</b> .....	<b>468</b>
<b>DateTime widget</b> .....	<b>471</b>
<b>Gesture area widget</b> .....	<b>473</b>
<b>JavaScript function block widget</b> .....	<b>477</b>
<b>Multistate Image widget</b> .....	<b>479</b>
<b>Multistate Image Multilayer widget</b> .....	<b>479</b>
<b>Network Adapters widget</b> .....	<b>481</b>
<b>RSS Feed widget</b> .....	<b>481</b>
<b>Scrolling RSS Feed widget</b> .....	<b>482</b>
<b>Media Player widgets</b> .....	<b>483</b>
<b>Browser widget</b> .....	<b>486</b>
<b>IP Camera widgets</b> .....	<b>487</b>
<b>Table widget</b> .....	<b>490</b>
<b>Grid Layout widget</b> .....	<b>507</b>
<b>TextEditor widget</b> .....	<b>514</b>
<b>Variables widget</b> .....	<b>516</b>
<b>QR Code widget</b> .....	<b>518</b>



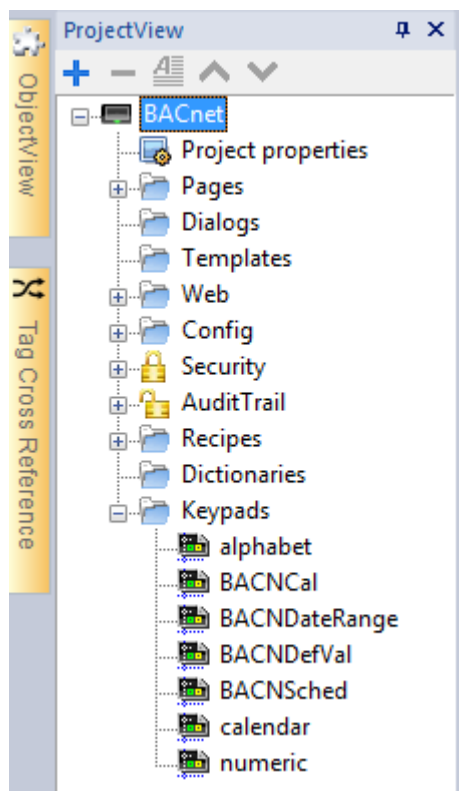
# BACnet widget

Path: **Widget Gallery**> **BACnet**

BACnet widgets are special widgets that let you interact with native BACnet objects.

- BACnet Calendar
- BACnet Scheduler
- BACnet Effective Period

These widgets are using special keypads that are added into the keypads folder when widgets are used. Generally, you do not need to take care of these keypads unless you want customized them.



Refer to the BACnet manual inside the “Communication Drivers” folder for a detailed description of BACnet special widgets.



**BACnet widgets are not supported on Web pages.**

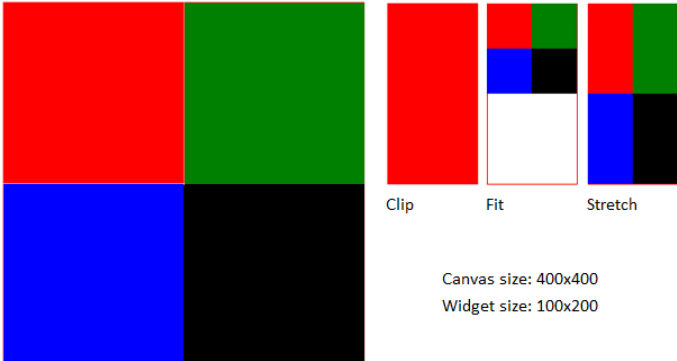

# Canvas Widget

Path: **Widget Gallery**> **Basic**> **Generic Canvas**

Canvas widget can be used to draw graphic via JavaScript scripting.



Note: the JavaScript methods are the same that are available for the HTML5 <canvas> tag

Parameter	Description
<b>Canvas Width</b> <b>Canvas Height</b>	<p>Canvas size.</p> <p>Note this is not the widget size. For example, the canvas size could be 500x500 pixels where the widget size could be 100x100 pixels. Draw Hint parameter will define how to stretch the canvas size to fit the widget size.</p>
<b>Draw Hint</b>	<p>Define how fit the canvas inside the widget size</p> <ul style="list-style-type: none"> <li>• <b>Clip</b> No Transformation is applied, coordinate system is not scaled and drawing is clipped inside the widget bounding rectangle.</li> <li>• <b>Fit to size</b> Fit to the widget size preserving the canvas model aspect ratio.</li> <li>• <b>Stretch</b> Fit to the widget size ignoring the canvas model aspect ratio.</li> </ul> <p>Example using a Canvas size larger than the widget size:</p> 
<b>Design Time Preview</b>	<p>Canvas preview inside PB610 Panel Builder 600</p> <p> Note the JavaScript code could use data not available inside PB610 Panel Builder 600 but only inside the HMI device</p>
<b>Auto Clear Background</b>	<p>Automatic clear the background before draw canvas. When disabled, the painted items are persisted and is not necessary redraw everything from scratch.</p>
<b>OnDraw Action</b>	<p>The OnDraw event is executed when the page is painted. This event has to be linked with the JavaScript code that draws the canvas graphic.</p>
<b>OnMousePress Action</b> <b>OnMouseRelease Actions</b> <b>OnMouseDrag Actions</b>	<p>Mouse events</p>

## Available Canvas Methods

// Painter Save/Restore

- void save(); // calls painter save
- void restore(); // calls painter restore

---

#### // Scale/Transform

- void scale(qreal x, qreal y);
- void rotate(qreal angle);
- void translate(qreal x, qreal y);
- void transform(qreal m11, qreal m12, qreal m21, qreal m22, qreal dx, qreal dy);
- void setTransform(qreal m11, qreal m12, qreal m21, qreal m22, qreal dx, qreal dy);

#### // Gradient

- CanvasGradient createLinearGradient(qreal x0, qreal y0, qreal x1, qreal y1);
- CanvasGradient createRadialGradient(qreal x0, qreal y0, qreal r0, qreal x1, qreal y1, qreal r1);

#### // Rectangle Functions

- void clearRect(qreal x, qreal y, qreal w, qreal h);
- void fillRect(qreal x, qreal y, qreal w, qreal h);
- void strokeRect(qreal x, qreal y, qreal w, qreal h);
- void rect(qreal x, qreal y, qreal w, qreal h);

#### // Path

- void beginPath();
- void closePath();
- void moveTo(qreal x, qreal y);
- void lineTo(qreal x, qreal y);
- void quadraticCurveTo(qreal cpx, qreal cpy, qreal x, qreal y);
- void bezierCurveTo(qreal cp1x, qreal cp1y, qreal cp2x, qreal cp2y, qreal x, qreal y);

#### // Drawing Text

- void fillText(const QString &text, qreal x, qreal y);

#### // Arc

- void arcTo(qreal x1, qreal y1, qreal x2, qreal y2, qreal radius);
- void arc(qreal x, qreal y, qreal radius, qreal startAngle, qreal endAngle, bool anticlockwise);

#### // Fill/Stroke

- void fill();
- void stroke();
- void clip();
- bool isPointInPath(qreal x, qreal y) const;

#### // Image manipulation (Draw QImageWgt using target and source rect)

- void drawImage(QObject \*pObjImage, qreal sx, qreal sy, qreal sw, qreal sh, qreal dx, qreal dy, qreal dw, qreal dh);
- void drawImage(QObject \*pObjImage, qreal dx, qreal dy);
- void drawImage(QObject \*pObjImage, qreal dx, qreal dy, qreal dw, qreal dh);
- void drawImage(const QVariant& image, int width, int height, const QString& format, qreal sx, qreal sy, qreal sw, qreal sh, qreal dx, qreal dy, qreal dw, qreal dh);

#### // Pixel manipulation

- `ImageData createImageData(double sw, double sh);`//Empty Image
- `ImageData createImageData(ImageData fromImage);`//from another Image
- `ImageData createImageData(ArrayBuffer value);` //From arraybuffer
- `void putImageData(ImageData imgData,double dx, double dy);`
- `void putImageData(ImageData imagedata, double dx, double dy, double dirtyX, double dirtyY, double dirtyWidth, double dirtyHeight);`
- `ImageData getImageData(qreal sx, qreal sy, qreal sw, qreal sh);`

## Canvas JavaScript Example

The canvas is initially blank. To display something, a script first needs to access the rendering context and draw on it:

```
var ctx = me.context2d;
```

then you can use the canvas methods, as in the below example

```
function GenericCanvasWgt1_onDraw(me, eventInfo)
{
    var ctx = me.context2d;
    ctx.fillStyle = 'red';
    ctx.fillRect(0,0,250,250);
    ctx.fillStyle = 'green';
    ctx.fillRect(250,0,250,250);
    ctx.fillStyle = 'blue';
    ctx.fillRect(0,250,250,250);
    ctx.fillStyle = 'black';
    ctx.fillRect(250,250,250,250);
}

function GenericCanvasWgt1_onMouseDown(me, eventInfo)
{
    alert("X = " + eventInfo.posX + "\nY = " + eventInfo.posY );
}
```

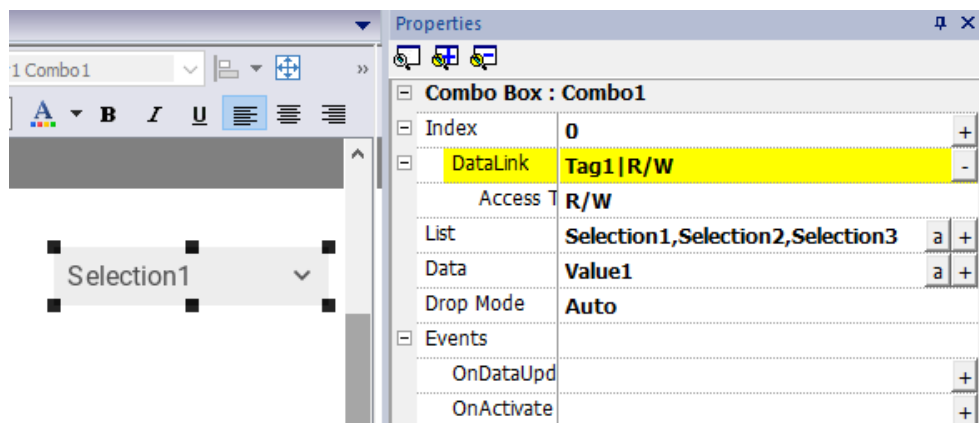
The update method can be used to dynamically redraw a canvas widget



```
function BtnStd1_btn_onMouseClicked(me, eventInfo)
{
    var myCanvasWidget = page.getWidget("GenericCanvasWgt1");
    myCanvasWidget.update();
}
```

## Combo Box widget

**Path:** *Widget Gallery*> *Basic*> *Controls*

Use this widget as a selector widget or to filter rows in a table to display only the values selected in the combo box.



Parameter	Description
<b>Index</b>	Index of the selected item.
<b>List / String List</b>	Item strings in the combo box.  Note: This field is multi-language.
<b>Data / Data List</b>	Returns the value in the Data List column (as string) in the Data field of the widget.  Tip: Use this parameter to return a custom value based on an item selected in the combo box.
<b>Text</b>	Format of displayed text.

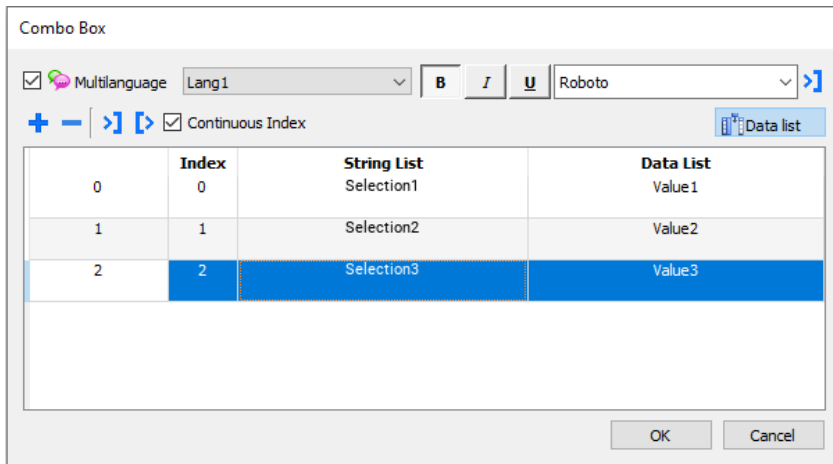
### Data List

The Data List is associated with the "listData" property and can be modified dynamically using the JavaScript code.

```
// To read the Data List
var comboWgt = page.getWidget("Combo1");
var listData = comboWgt.getProperty("listData")

// To write the Data List
var comboWgt = page.getWidget("Combo1");
comboWgt.setProperty("listData", "NewData1,NewData2,NewData3");
```

## Attaching data vs. attaching indexes



In many projects you may need to attach fields such as **Index** or **Data** to tags to know the values of the selected item in the combo box. Use:

- **Index**: to display the index (integer) of the selected item (0...n).
- **Data**: to display the data value (string) specified in the Data List column.

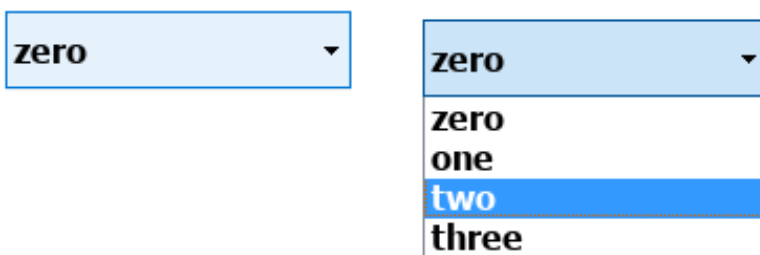
## Combo Box widget “full screen” mode with images

From the "Project properties" on page 79 the look and behavior of Combo Boxes can switches from Context mode to Full Screen mode

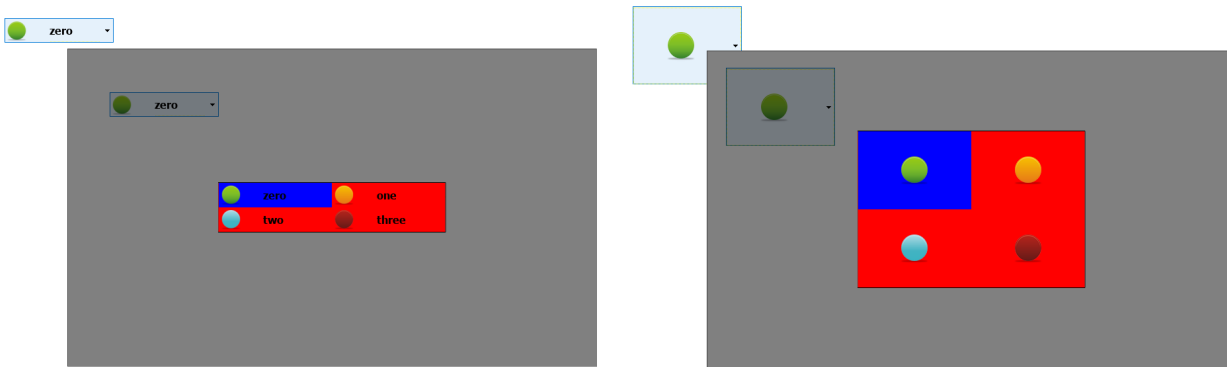
Path: **ProjectView**> double-click **Project properties**> **Properties pane**> **Project**> **ComboBox View Mode**

Parameter	Description
<b>ComboBox View Mode</b>	Select the visualization mode of all the Combo Box widgets of the project  <b>Context</b> Classic view with drop-down menus  <b>Full screen</b> Enhanced view with configurable texts and images that will pop up in the middle of the screen for easy scroll and selection.

**Context** view example



### Full screen view example



### Additional parameters available in full screen mode

The additional "Image List" column will be available inside **Combo Box > List** parameter:

Index	String List	Image List	Data List
0	zero	images\green_button.png	Msg-00
1	one	images\prange_button.png	Msg-01
2	two	images\turquoise_button.png	Msg-02
3	three	images\red_button.png	Msg-03



Note: Some properties are displayed only in advanced mode.

Parameter	Description
<b>Image</b>	Return, inside the attached tag, the file name of the selected image
<b>Button</b>	Define the look of the Combo Box <ul style="list-style-type: none"> <li>Show background = true Combo Box button is showed</li> <li>Show background = false Only image or text is showed</li> </ul>
<b>ListView</b>	Layout parameters of the Combo Box in edit mode
<b>ListViewItems</b>	Define the items type that will be inside the Combo Box <p>Image Mode:</p> <ul style="list-style-type: none"> <li>Only Text</li> <li>Only Images</li> <li>Text and Images</li> </ul>

### Additional macros available in full screen mode

- OpenComboBox
- CloseComboBox

## Consumption Meter widget

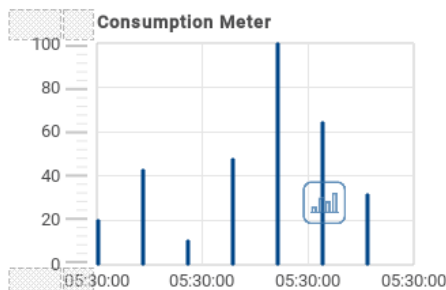
Path: **Widget Gallery**> **Basic**> **Trends/Graphs**

Use this widget to monitor a resource which is continuously increasing. The system reads the value of the resource and calculates the increment in a set range of time, the increment is then displayed in a bar-graph in a trend-like window.

Different colors can be used to used in the graph based on the time frame.



Tip: Use this widget to calculate the power consumption of a system.



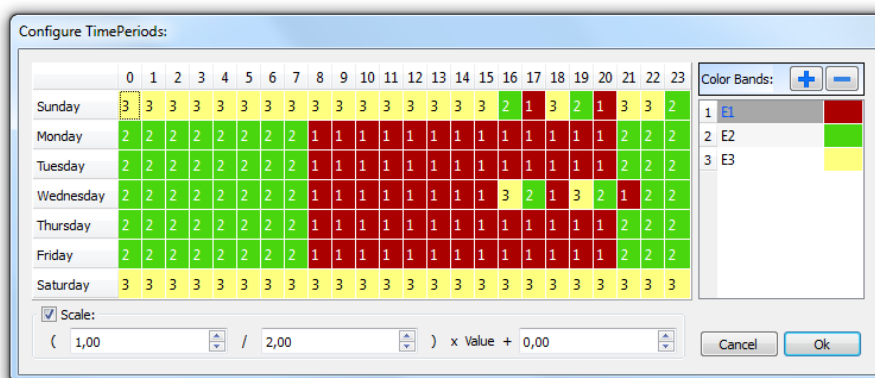


Parameter	Description
<b>Value</b>	Resource monitored
<b>Graph Duration</b> <b>Graph Duration</b> <b>Units</b>	Time period displayed in the window
<b>Bar Duration</b> <b>Bar Duration</b> <b>Units</b>	Time period represented by each bar in the graph
<b>Time Periods</b>	Assigns a specific color to highlight the increment of the monitored resource in a specified time period (minimum resolution = 1 hour).
<b>Color</b> <b>Bar Width</b>	Bar color and width
<b>Bar Value</b>	Show/Hide the value of each bar
<b>Consumption</b> <b>Meter</b>	Number of labels to be displayed on graph.

## Example: how to monitor energy consumption

In the following example a widget is design to monitor energy consumption with a weekly scale and a daily unit.

1. Attach a tag to the physical variable to monitor. In this example, to the total energy consumed (Tag KWh). This tag contains an incremental number that indicates how many KW/h have been consumed from when energy consumption started.
2. Add a Trend and link it to the tag to be monitored, Tag KWh.
3. Add a **Consumption Meter** widget to a page.
4. Attach the **Value** property of the Consumption Meter to the Trend you created in step 2.
5. Set **Graph Duration/Units** to 1 week: this will give you a weekly graph of consumed energy.
6. Set **Bar Duration/Units** to 1 day, this is the time range when energy consumption is calculated.
7. In **Consumption Meter** set the number of labels to show in the bar graph, in this case 7 to display a weekly graph.
8. From the **Time Periods** property open the **Configure Time Periods** dialog: set the different colors for different values of Tag KWh in each bar.





Tip: To assign the color to the cells of the table, select the cells and click on the desired color, or enter the index value of the band (1, 2, 3) into the cell.

9. Add as many color bands as you need, in this example 3 color bands.
10. Assign a band to each hour in the weekly table, in this example a red band (E1) is used to indicate the range of time in the day/week where the cost of energy is the highest.



Note: You can apply a scale factor to each color band, if needed.

The result is a bar graph consumption meter showing daily consumption of energy in KW/h, with colors indicating the different energy costs. The height of each bar represents the amount of energy in the time range considered, 1 day in this example.

Use the action ConsumptionMeterPageScroll to scroll the bar graph back and forth and the action RefreshTrend to refresh the bar graph since data is not refreshed automatically.



**Important: No other Trend action is currently supported by the Consumption Meter widget.**

## Control list widget

Path: **Widget Gallery > Advanced > Control List**

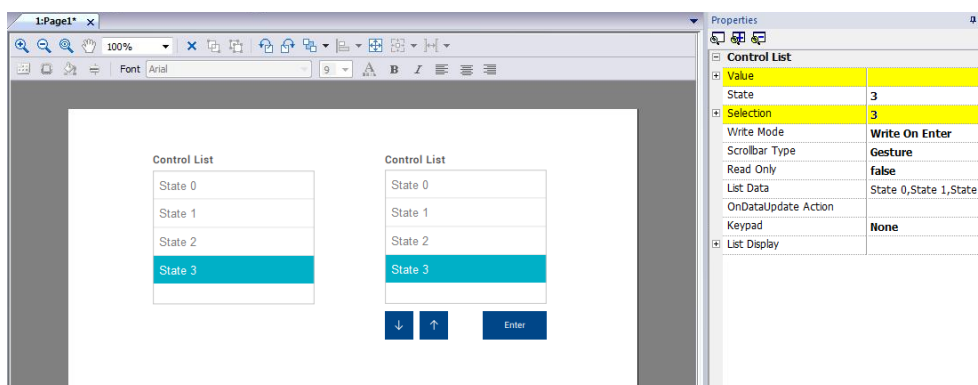
Use these widgets to represent the status associated with a particular process and to control that process from the same widget.



**Not yet available in the new gallery (use the old gallery to get this widget)**

Two types of control lists are available:

- a group control list, with a limited set of navigation button already included, and
- a basic control list with no pre-configured button to be navigated using the touch screen feature.

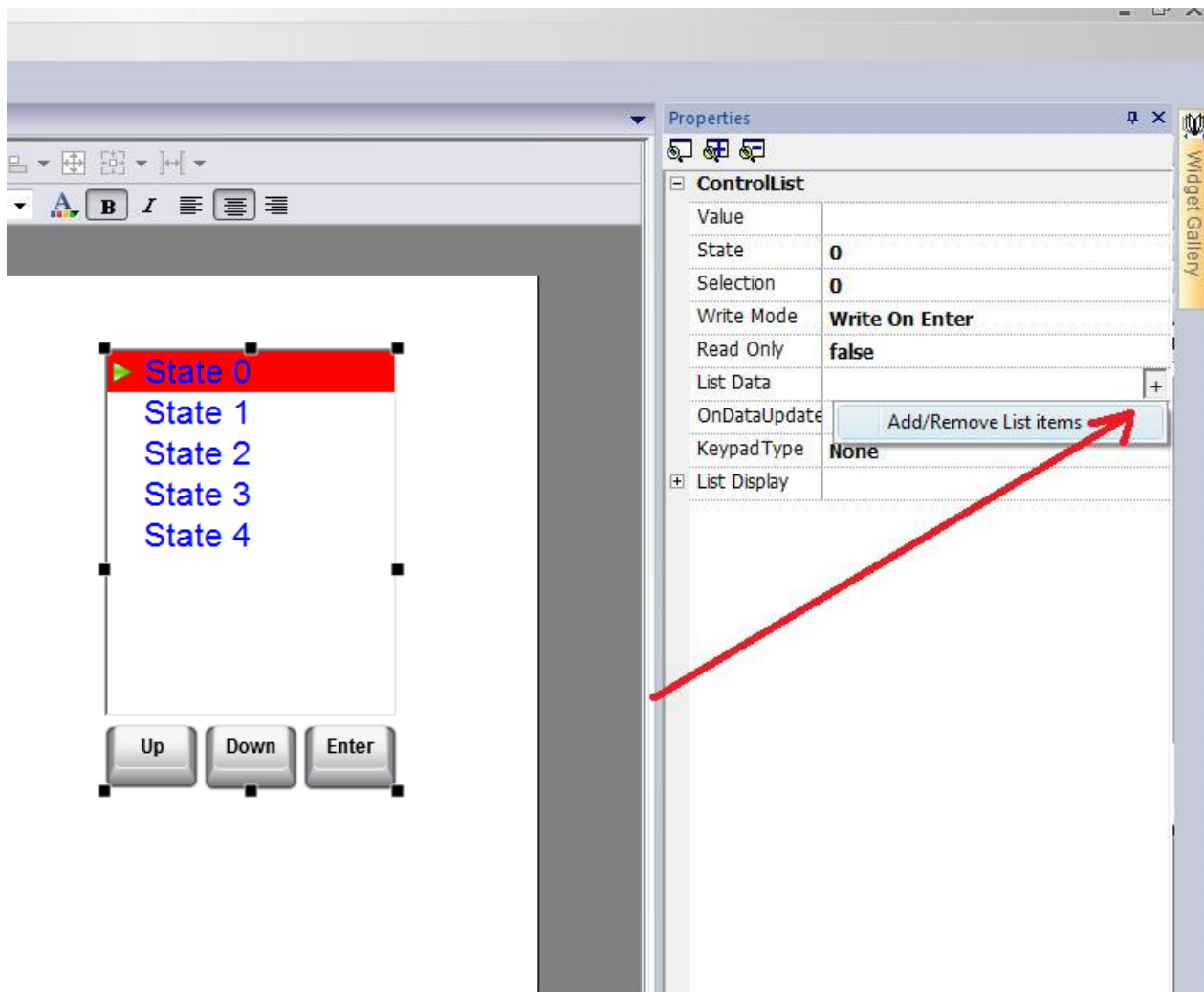


Parameter	Description
<b>Value</b>	The value corresponding to the status of the widget. If there is a tag attached to the value property, when loading the widget, the State will be aligned with the tag value.
<b>State</b>	State of widget. The widget highlights the item related to its State with a different background color (see "state color" in the properties of the widget).
<b>Selection</b>	State selection. The selected item will be displayed with a small triangle on the left side of the list.
<b>Write Mode</b>	Select the State update mode <ul style="list-style-type: none"> <li>• <b>Write On Select:</b> The state is updated automatically to be aligned with the cursor position.</li> <li>• <b>Write On Enter:</b> The status is updated with the cursor position only when the user presses enter</li> </ul>
<b>Scrollbars Type</b>	Select the scroll mode of the table <ul style="list-style-type: none"> <li>• <b>Gesture:</b> Pan gesture can be used to smoothly scroll the table.</li> <li>• <b>Scrollbar:</b> Use the scrollbar to scroll the table</li> </ul>
<b>Read Only</b>	Defines whether the list is only an indicator.
<b>List Data</b>	List of status items. Each item has a status name, a corresponding value and a flag that enables to display the item inside the widget. <div data-bbox="384 1102 1356 1736" data-label="Image"> </div>

## Defining states

Add/remove states, that is items in the list, from the **List Data** property.

Any value can be assigned to a state. When you activate the state, by selecting the related item if in **WriteOnSelect** mode or selecting it and confirming with enter if **Write On Enter**, this will write the value assigned to state to the tag linked to the Control List widget **Value** .



## Manage list data items from JavaScript code

The list of data items can be modified, at runtime, from JavaScript code using the **`setProperty("listData", <NewControlList>`**). The below example shows how to modify the list of items

```
function SetItemsList_btn_onMouseClicked(me, eventInfo)
{
    var NewControlList = [{"OFF",100,true}, {"ON",101,true}, {"MAN",102,true}, {"AUTO",103,true}];
    var ControlListWgt = page.getWidget("controlListBtn.controlList");
    ControlListWgt.setProperty("listData", NewControlList);
}
```

### Where

- `NewControlList` is an array with the items description
- `controlListBtn.controlList` is the ID of the Control List Widget to modify

The **`getProperty("listData")`**, instead, will just return a comma separated string of just the names.

```
function Read_btn_onMouseClicked(me, eventInfo)
{
    var ControlListWgt = page.getWidget("controlListBtn.controlList");
    var ListData = ControlListWgt.getProperty("listData");
}
```

Where the result of ListData will be: "OFF,ON,MAN,AUTO"

### State

The `getProperty("state")` can be used to retrieve the State value. Here is an example of the JavaScript code

```
function controlListBtn_onDataUpdate(me, eventInfo)
{
    var ControlListWgt = page.getWidget("controlListBtn.controlList");
    var State = ControlListWgt.getProperty("state");
    project.setTag("State", State);
    return false;
}
```

## TabBar widget

*Path: Widget Gallery > Basic > Control > Horizontal tab*

Path: **Widget Gallery > Basic > Control > Vertical tab**

The widget should contain a list of checkable buttons (one or more) where one and only one button can be checked at a time (in the example image only the Tab 1 button is checked). When the size of the widget is not sufficient to show all checkable buttons, two other buttons (the two arrows in the image) should appear in order to control the viewport position. In the example image, only the right button is enabled; we will refer to these two buttons as viewport buttons. The embedded gestures to scroll left/right the tabs bar are supported.



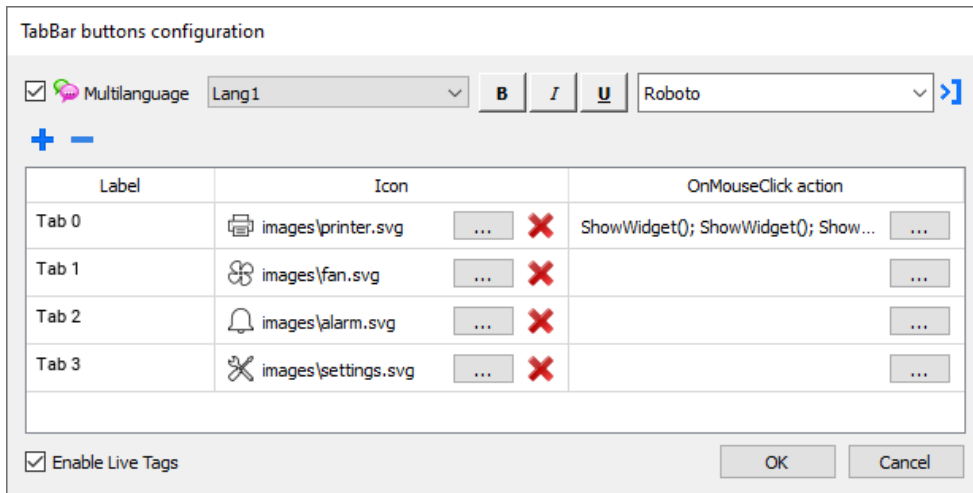
### TabBar properties

Property	Description
<b>Current Index</b>	The currently checked tab button (default 0)
<b>Button Expanding</b>	When true, the buttons will adjust to take up all available space
<b>Minimum Button Size</b>	Minimum Button Size "Text Elide Mode" and "Button Expanding" can increase the size of the button
<b>Background Color</b>	Color used for unchecked tab buttons and for disabled viewport control buttons
<b>Foreground Color</b>	Color used for text and viewport buttons icons
<b>Accent Color</b>	Color used for the checked tab button and for enabled viewport control buttons
<b>Text Elide Mode</b>	The display mode when the text is too large. Can be

Property	Description
	<ul style="list-style-type: none"> <li>• Elide Right</li> <li>• Expanding</li> </ul>
<b>Icons</b>	Icons parameters
<b>- Expanding</b>	When true, the icons are expanding to use all available space
<b>- Position</b>	Icon position
<b>- Min Width</b>	Minimum width for icons
<b>- Min Height</b>	Minimum height for icons
<b>Viewport</b>	Viewport buttons parameters
<b>- Mode</b>	The visualization mode of the viewport control buttons. Can be: <ul style="list-style-type: none"> <li>• whenNeeded</li> <li>• alwaysOn</li> <li>• alwaysOFF</li> </ul>
<b>- Button Size</b>	Size of the viewport buttons
<b>- Position</b>	The viewport position (should be between 0.0 to 1.0)  This property is useful for connecting the viewport of two TabBars to each other, in order to synchronize the visible buttons (for example identical TabBars on different pages)
<b>Tabs</b>	Tabs configuration. Each tab can have: <ul style="list-style-type: none"> <li>• label</li> <li>• an icon</li> <li>• associated actions</li> </ul>
<b>Style</b>	TabBar Style  There are some predefined styles ready to use and a custom style. By selecting the "Custom" style, the new "Custom Style" sub-folder will appear with a list of all the style properties that can be used to precisely define any display details.
<b>Custom Style</b>	Available only when Style = Custom  All the properties that are used to draw the TabBar widget.

## TabBar buttons configuration

Double click over the TabBar widget or press the Tab property to open the "TabBar buttons configuration". From this dialog, you can define the label, the icon, and the action associated with each button.



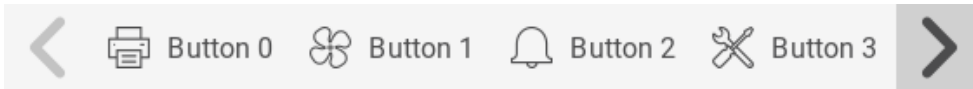
The text in the label supports the live tag placeholder (See ["Live Tags" on page 34](#))

## ToolBar widget

*Path: Widget Gallery > Basic > Control > Horizontal toolbar*

*Path: Widget Gallery > Basic > Control > Vertical toolbar*

The ToolBar widget is a list of clickable buttons that does not keep the status of the "selected button". All properties are the same as the TabBar widget with the exception of the "Current Index" property that is missing. (See ["TabBar widget" on page 466](#))



## Stack widget

*Path: Widget Gallery > Layout > Stack*

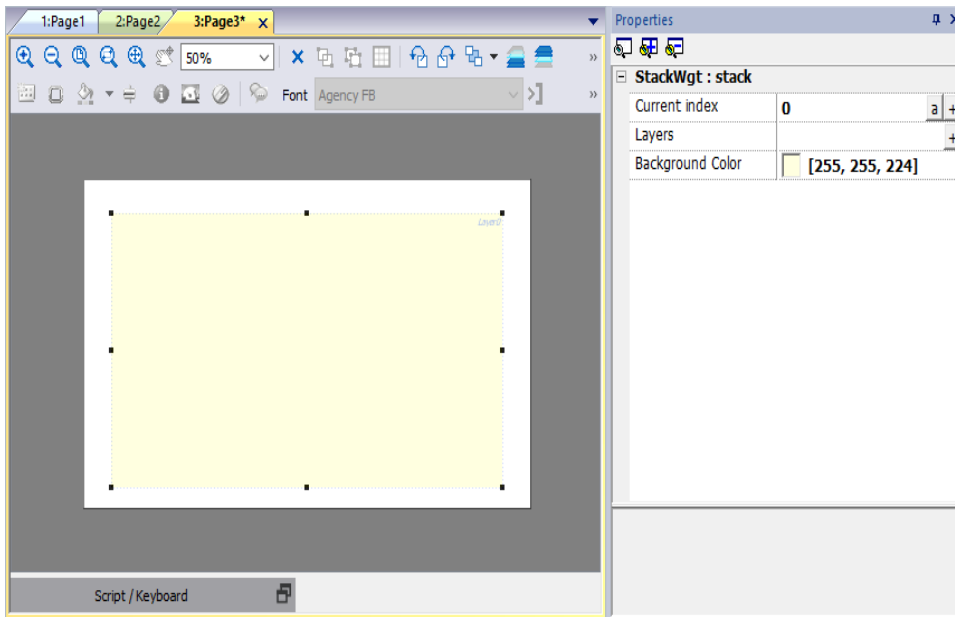
The Stack widget is a widget to easily manage objects on multilayers.



**Available only on Linux devices (See: ["HMI devices capabilities" on page 613](#))**

To edit a layer:

- Use "Current Index", in the property panel, to select the layer to edit
- Double click over the Stack widget
- Drag and drop widgets from the Gallery to the Stack widget



At runtime, the value of the "Current Index" property selects the layer to show.



**The Stack widget is not an alternative to the pages. The resources of the HMI Device are better optimized for the pages.**

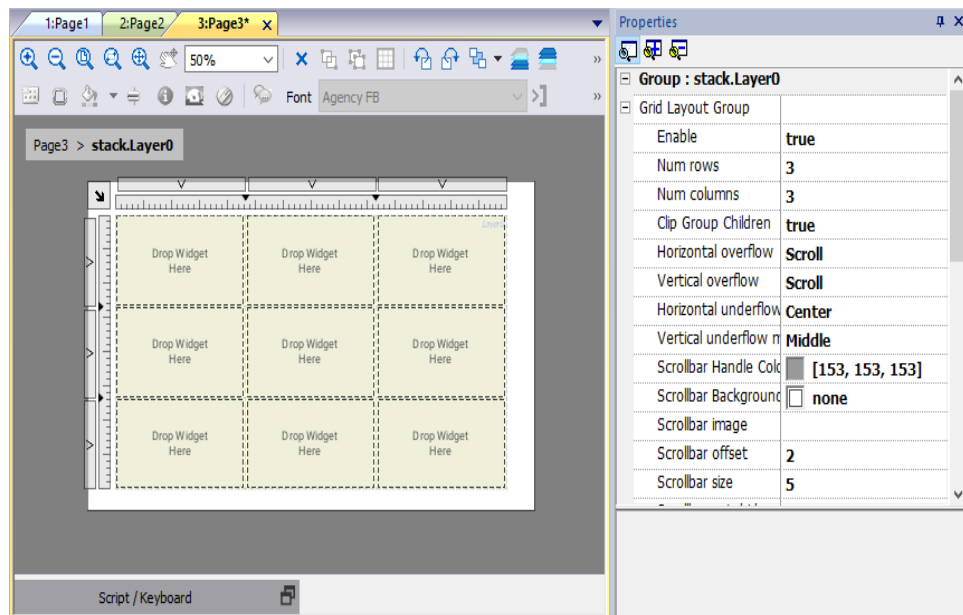
## Stack properties

Property	Description
<b>Current Index</b>	The currently visible layer
<b>Layers</b>	Number of levels with their name (you can use up to 10 levels)
<b>Background Color</b>	Color used for unchecked tab buttons and for disabled viewport control buttons

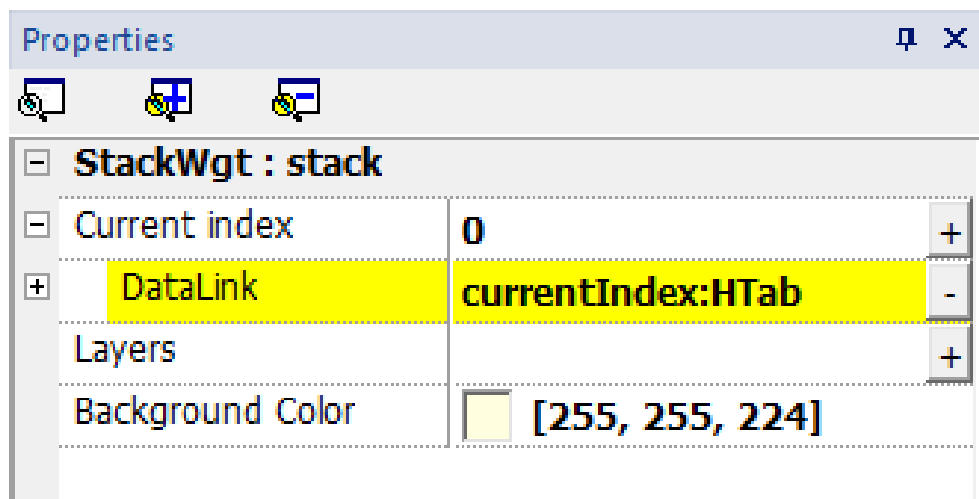
## Example of use

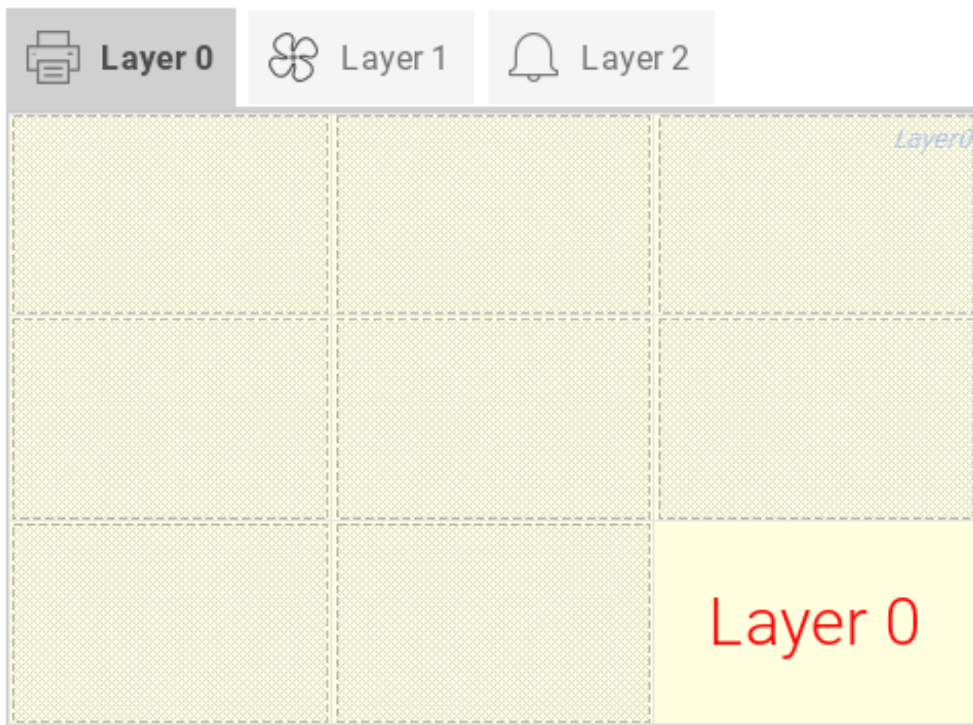
When the Stack widget is enabled for editing, in the properties panel it is possible to enable the grid layout and define some rows and columns to create the cells where to place the objects (Ref. "[Grid Layout widget](#)" on page 507).





If you add a TabBar widget and synchronize the "Current Index" between the TabBar and the Stack widget you can easily get a layer view managed by tabs.



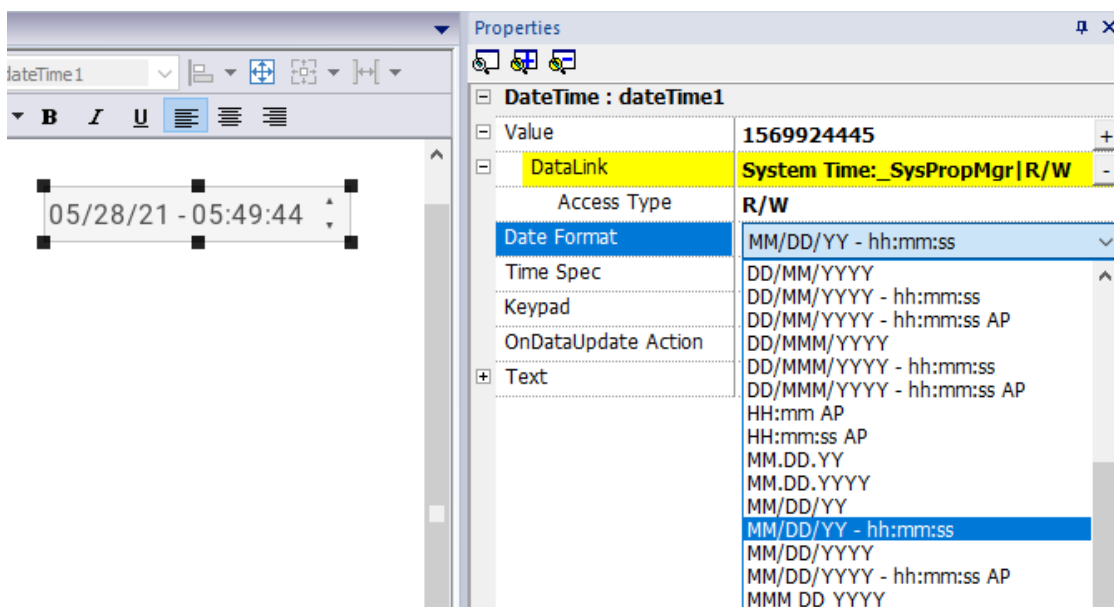


## DateTime widget

**Path:** *Widget Gallery*> *Basic*> *Controls*

Use this widget to display and edit current date and time .

In the **Properties** pane different formats are available for representing date and time.



### Time options

For the **Time Spec** property select which time the widget will show at runtime.

Option	Description
<b>local</b>	shows local time, the time of the HMI device where the project is running
<b>global</b>	shows Global Time (GMT)
<b>server</b>	shows time information as handled by the server side of the HMI device

## Time and Date placeholders

You can use placeholders to freely define the Time and Date format

Date	Description
<b>d</b>	the day as number without a leading zero (1 to 31)
<b>dd</b>	the day as number with a leading zero (01 to 31)
<b>ddd</b>	the abbreviated localized day name (e.g. 'Mon' to 'Sun')
<b>dddd</b>	the long localized day name (e.g. 'Monday' to 'Sunday')
<b>M</b>	the month as number without a leading zero (1-12)
<b>MM</b>	the month as number with a leading zero (01-12)
<b>MMM</b>	the abbreviated localized month name (e.g. 'Jan' to 'Dec')
<b>MMMM</b>	the long localized month name (e.g. 'January' to 'December')
<b>yy</b>	the year as two digit number (00-99)
<b>yyyy</b>	the year as four digit number

Time	Description
<b>h</b>	the hour without a leading zero (0 to 23 or 1 to 12 if AM/PM display)
<b>hh</b>	the hour with a leading zero (00 to 23 or 01 to 12 if AM/PM display)
<b>m</b>	the minute without a leading zero (0 to 59)
<b>mm</b>	the minute with a leading zero (00 to 59)
<b>s</b>	the whole second without a leading zero (0 to 59)
<b>ss</b>	the whole second with a leading zero where applicable (00 to 59)
<b>AP or A</b>	use AM/PM display. A/AP will be replaced by either "AM" or "PM"
<b>ap or a</b>	use am/pm display. a/ap will be replaced by either "am" or "pm"

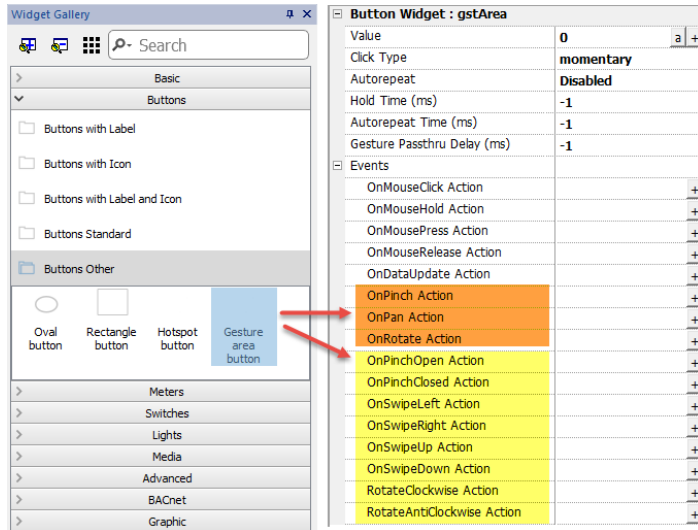
## Regional Settings


You can use even the SHORT-DATE or the LONG-DATE placeholders to use the format defined inside the Regional Setting (see "[Regional Settings](#)" on page 93)

# Gesture area widget

Path: **Widget Gallery**> **Buttons**> **Others**

Gesture Area Widget is a hotspot button that generates gesture events.



Gesture Events	Description
<b>OnSwipeLeft</b> <b>OnSwipeRight</b>	An event is release when swipe gesture is detected
<b>OnPinchOpen</b> <b>OnPinchClose</b>	An event is release when pinch gesture is detected
<b>RotateClockwise</b> <b>RotateAntiClockwise</b>	An event is release when rotate gesture is detected
<b>OnPan</b> <b>OnPinch</b> <b>OnRotate</b>	A series of events released during the gesture. Only JavaScript can be used to service these events, through the JavaScript code the developer can manage the gestures events as he prefer.  <b>WARNING: Only multi touch HMI devices can generate OnPinch and OnRotate events</b>

## OnPan

`boolean onGesturePan(me, eventInfo)`

This event occurs when one point inside the area has pressed and a linear movement has been detected.

Parameter	Description
<b>me</b>	Object triggering the event.
<b>eventInfo</b>	<b>id</b> = Gesture id; it is used to identify different gestures. <b>running</b> = True except for last event delivered to notify gesture completion. <b>dx</b> = Total X axis movement in screen pixel units from initial touch position . <b>dy</b> = Total Y axis movement in screen pixel units from initial touch position.

## OnPinch

`boolean onGesturePinch(me, eventInfo)`

This event occurs when two points inside the area have been pressed and a linear movement has been detected.

Parameter	Description
<b>me</b>	Object triggering the event
<b>eventInfo</b>	<b>id</b> = Gesture id; it is used to identify different gestures. <b>running</b> = True except for last event delivered to notify gesture completion. <b>dx</b> = Total X axis movement in screen pixel units from initial touch position. It represents the distance change between fingers. Positive value means that the distance is increasing; negative value means that the distance is decreasing. This amount may be used to control a zoom value. <b>dy</b> = Total Y axis movement in screen pixel units (see dx).

## OnRotate

`boolean onGestureRotate(me, eventInfo)`

This event occurs when two points inside the area have been pressed and a rotate movement has been detected.

Parameter	Description
<b>me</b>	Object triggering the event
<b>eventInfo</b>	<b>id</b> = Gesture id; it is used to identify different gestures. <b>running</b> = True except for last event delivered to notify gesture completion. <b>drot</b> = How many degrees (0/360) have been added since the previous event. <b>trot</b> = Total degrees (0/360) of the entire movement. Positive numbers meaning clockwise rotation, negative anticlockwise rotation.

## Gesture events pass thru

To use a widget (e.g. a button or a slider) covered from a gesture object, you have to keep pressed the widget 200 mSec to move the control to the underlying object. The time that must be waited for to send the command to the underlying

object can be modified from the “Gesture Passthru Delay” parameter that is available in the advanced properties view.

Parameter	Description
<b>Gesture Passthru Enabled</b>	<p>Enable the possibility to pass gesture events to underlying widgets after a configurable delay. User has to keep pressed the finger and then execute the gesture.</p> <p><b>default</b> = Use the value defined in the project properties. See <a href="#">"Project" on page 87</a></p> <p><b>true</b> = Gesture passthru enabled</p> <p><b>false</b> = Gesture passthru disabled</p>
<b>Gesture Passthru Delay (ms)</b>	<p>The time that must be waited for to send the command to the underlying object</p> <p>0/500            mSec</p> <p>-1                Use the delay defined in the project properties. See <a href="#">"Project" on page 87</a></p>

## Examples of using gesture events in association with JavaScript

Here some example of using gesture events in association with JavaScript code to identify gestures and program the requested actions

### Swipe Gesture

How to recognize a "swipe" gesture to change page in the application.

1. Put a Gesture area widget into the page
2. Configure the OnPan Action to trigger a JavaScript function
3. Write the JavaScript code that recognize and manage the swipe gesture

The screenshot shows a development environment with a widget configuration panel on the right and a script editor on the left. The widget is a 'Swipe' gesture area. The configuration panel shows the following settings:

- Value: 0
- Click Type: momentary
- Autorepeat: Disabled
- Hold Time (ms): -1
- Autorepeat Time (ms): -1
- Events:
  - OnClick Action
  - OnMouseHold Action
  - OnMousePress Action
  - OnMouseRelease Action
  - OnPinch Action
  - OnPan Action (highlighted with a red circle 2): 1 Action
    - Action[0]: js:Swipe\_onGesturePan()
  - OnDataUpdate Action
- General
- Position

The script editor shows the following JavaScript code:

```

1 function Swipe_onGesturePan(me, eventInfo)
2 {
3     if (eventInfo.running !== 1) {
4         var dx = eventInfo.dx;
5         if (dx > 0) {
6             project.nextPage();
7         }
8         if (dx < 0) {
9             project.prevPage();
10        }
11    }
12 }
    
```

Red circles 1 and 2 in the image highlight the widget area and the OnPan Action configuration, respectively.

## Pinch Gesture

How to recognize a "pinch" gesture to resize an image.

1. Put a Gesture area widget into the page over the image
2. Configure the OnPinch Action to trigger a JavaScript function
3. Write the JavaScript code that recognize and manage the pinch gesture

The screenshot shows a software development environment with a canvas on the left and a Properties panel on the right. The canvas displays a construction sign with a yellow light and a 'UNDER CONSTRUCTION' sign. A red dashed box with a red circle '1' is drawn over the top-left corner of the sign, indicating the gesture area. The Properties panel shows the configuration for a 'Button Widget : gstArea'. The 'OnPinch Action' is set to '1 Action', which is 'js:gstArea\_onGesturePinch()'. A red circle '2' is next to the 'OnPinch Action' entry. Below the Properties panel, the JavaScript code for the 'gstArea\_onGesturePinch' function is shown, with a red circle '3' next to the function definition. The code includes variables for pinchDX and pinchDY, and a function to resize the image based on the pinch gesture.

Button Widget : gstArea	
Value	0
Click Type	momentary
Autorepeat	Disabled
Hold Time (ms)	-1
Events	
OnMouseClicked Action	+
OnMouseHold Action	+
OnMousePress Action	+
OnMouseRelease Action	+
OnPinch Action	1 Action
Action[0]	js:gstArea_onGesturePinch()
OnPan Action	+
OnDataUpdate Action	+

```

1 3 var pinchDX = 0;
2  var pinchDY = 0;
3  function gstArea_onGesturePinch(me, eventInfo)
4  {
5      if (eventInfo.running == 0) {
6          pinchDX = 0;
7          pinchDY = 0;
8      } else {
9          pinch_picture((eventInfo.dx-pinchDX), (eventInfo.dy-pinchDY));
10         pinchDX = eventInfo.dx;
11         pinchDY = eventInfo.dy;
12     }
13 };
14
15 function pinch_picture(dx, dy) {
16     var shape = page.getWidget("image1");
17     shape.x = shape.x - dx/2;
18     shape.y = shape.y - dy/2;
19     shape.width = shape.width + dx/2;
20     shape.height = shape.height + dy/2;
21 };
22

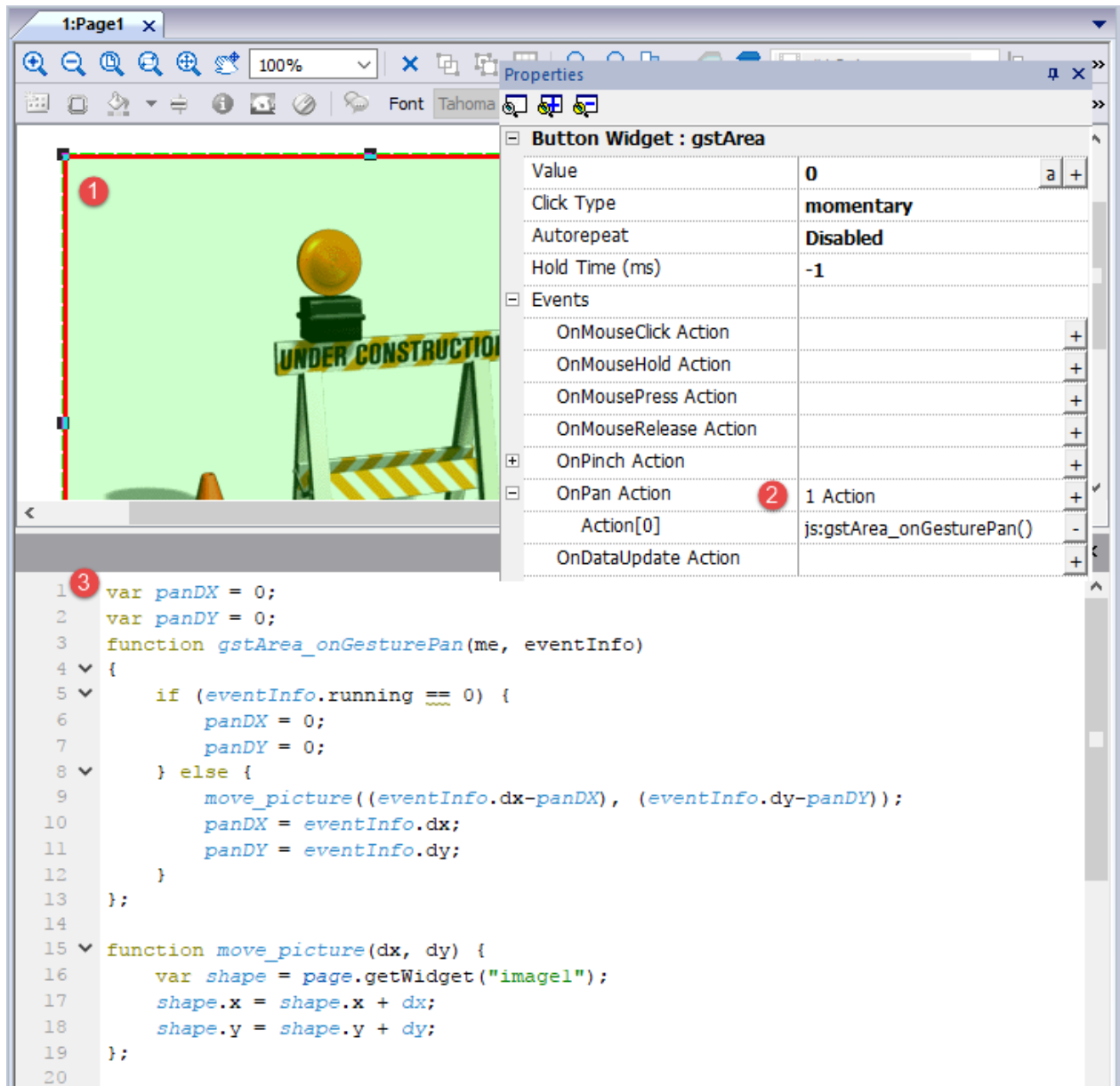
```

## Pan Gesture

How to recognize a "pan" gesture to move an image.

1. Put a Gesture area widget into the page over the image
2. Configure the OnPan Action to trigger a JavaScript function

3. Write the JavaScript code that recognize and manage the pan gesture



## JavaScript function block widget

Path: *Widget Gallery > Basic > JSFunctionBlock*

JavaScript Function Block is a widget that contains JavaScript logic that is executed when tags values change.

Parameter	Description
value1 ... value16	Objects that will trigger the OnDataUpdate action.
OnDataUpdate	Action that will be executed when a change of an associated value is detected





Note: This widget is rendered only in PB610 Panel Builder 600, and it is not rendered in the HMI device.

### Example:

A JavaScript code that check the combination lock of three selectors

The screenshot shows the Panel Builder 600 interface with three selector widgets (ONE, TWO, THREE) and a JavaScript function block widget. The Properties window is open, showing the configuration for the JSFuncBlockWgt widget. The widget has three DataLink properties (value1, value2, value3) connected to NeedleWgt.value:Knob1, NeedleWgt.value:Knob2, and NeedleWgt.value:Knob3 respectively. The Access type is set to R. The Events section shows an OnDataUpdate event with an Action of js:JSFuncBlock\_onDataUpdate() and an initial delay of 0.

```

1 function JSFuncBlock_onDataUpdate(me, e
2 {
3   var wUNLOCK = page.getWidget("unlock")
4
5   //Accept the incoming new value
6   me[eventInfo.attrName] = eventInfo.n
7
8   //Check the unlock code
9   if ((me.value1=="3") && (me.value2=="
10    wUNLOCK.setProperty("value1", me
11  } else {
12    wUNLOCK.setProperty("value", me.
13  });
14
15  return false;
16 }
17 }

```

The screenshot shows the Script editor with the following JavaScript code:

```

1
2 function JSFuncBlockWgt_onDataUpdate(me, eventInfo)
3 {
4   var wUNLOCK = page.getWidget("unlock")
5
6   // Accept the incoming new value
7   me[eventInfo.attrName] = eventInfo.newValue;
8
9   // Check the unlock code
10  if ((me.value1=="3") && (me.value2=="3") && (me.value3=="3")) {
11    wUNLOCK.setProperty("value", "Unlock!");
12  } else {
13    wUNLOCK.setProperty("value", me.value1+"-"+me.value2+"-"+me.value3);
14  };
15
16  return false;
17 };
18

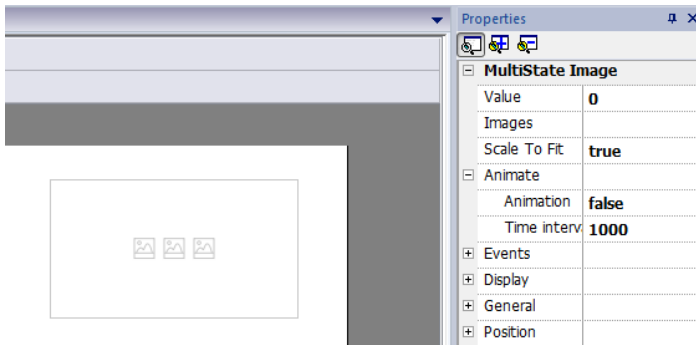
```

See ["Widget events" on page 544](#) for the description of the onDataUpdate parameters

## Multistate Image widget

*Path: **Widget Gallery**> **Basic**> **Images***

Use this widget to display an image from a collection based on the value of a tag used as Index. You can use this widget also for simple animations.

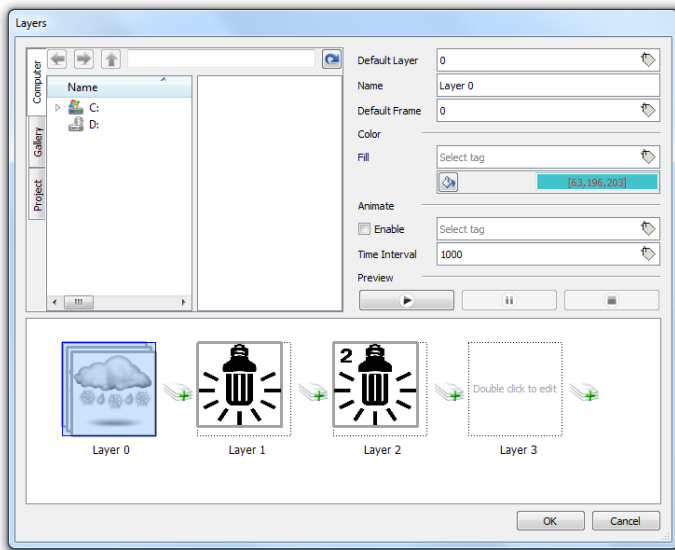


Parameter	Description
<b>Value</b>	Index of image to display. For example, set Value=0, to display the image with index 0 in the image collection.
<b>Images</b>	Images collection with associated index.
<b>Animate</b>	Set to true, to enable a slide show.
<b>Time interval</b>	Interval between images in the slide show.

## Multistate Image Multilayer widget

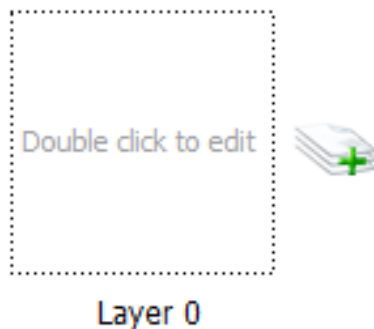
*Path: **Widget Gallery**> **Basic**> **Images***

Use this widget to create different animations and select the most suitable at runtime.

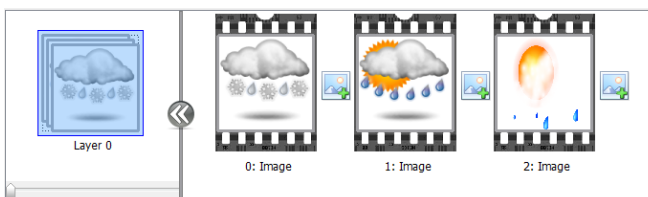


## Setting up widget layers

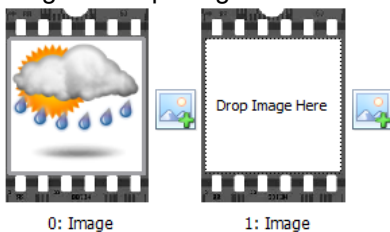
1. Open the **Layers** dialog from the **Properties** pane.
2. Click **+** to add as many layers as you need.



3. Double click on each layer to add as many images as you want to include in the layer.



4. Drag and drop images into the frame to add it to current layer.



5. Define widget properties.

Parameter	Description
<b>Default Layer</b>	Layer shown at runtime.
<b>Name</b>	Name of selected layer.
<b>Default Frame</b>	Frame shown when current layer is displayed.
<b>Color / Fill</b>	Fill color for images of current layer.
<b>Animate</b>	Enables slide show for active layer. Animations can be started/stopped at runtime attaching it to a tag.
<b>Time Interval</b>	Time interval of slide show, if enabled.
<b>Preview</b>	Slide show simulation.



Note: **Default Layer**, **Default Frame**, **Color** and **Fill** can be changed at runtime, attaching the to a tag.

## Network Adapters widget

Path: *Widget Gallery* > *Basic* > *Control*

Use the IP Widget to set the network adapters parameters.

### Network Adapter Parameters

<b>Mac ID:</b> 00:15:5D:59:A1:C6	LAN12	▼
<b>Use DHCP:</b>	No	▼
<b>IP Address:</b>	172.26.144.1	
<b>Subnet Mask:</b>	255.255.240.0	
<b>Gateway:</b>	0.0.0.0	
	<b>CANCEL</b>	<b>APPLY</b>

The system variable Network->Status contains the result of the last operation performed by the IP Widget (see "[Network variables](#)" on page 151 for details)


## RSS Feed widget

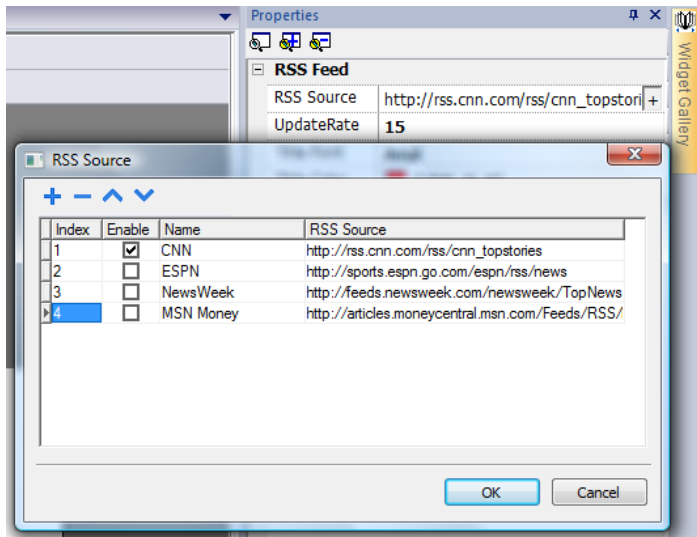
Path: *Widget Gallery* > *Media* > *RSSFeed Source*

Use this widget to display on the HMI device your favorite RSS feeds directly from the Internet.

RSSFeed



Parameter	Description
<b>RSS Source</b>	Feed URL   Note: Feed sources cannot be modified at runtime.
<b>UpdateRate</b>	Refresh time

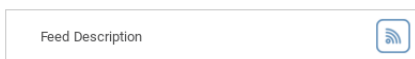


The RSS Feed widget has been specifically designed to work with Pocket Internet Explorer.

## Scrolling RSS Feed widget

**Path:** *Widget Gallery > Media > RSSFeed Scroll*

Use this version of the main RSS Feed widget to display highlights inside a text line using a smoothing scrolling text.



RSS Scroll Widget : RSSScrollWgt	
RSS Source	http://rss.cnn.com/rss/cnn_topstories +
UpdateRate	15
Title Separator	
Title Font	Tahoma
Title Color	■ [23, 30, 40]
Title Size	12
Scrolling	Normal

This widget has additional properties.

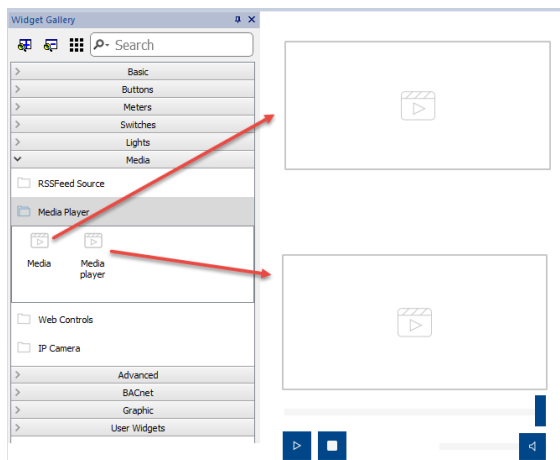
Parameter	Description
Scrolling	Scrolling speed
Title Separator	Separator character between highlights


## Media Player widgets

**Path:** *Widget Gallery > Media > Media Player*

Use these widgets to play videos from a playlist. The video files can be stored on a USB drive, on the Flash card or an SD Card.

Two widgets are available: one includes a multimedia frame with buttons to play and stop the video, the other is a plain frame where the video is played without user control.



Parameter	Description
<b>Media Player List</b>	<p>Open Windows file browser for selecting video files to collect in the play list. Selected files will be downloaded to the HMI device together with the project.</p> <p>When a USB device or an SD Card has been selected, files must be placed in a subfolder "mediafiles" of the external memory media. Video files will be played according to filename alphabetical order.</p> <p> <b>Ensure you have the commercial rights of the multimedia files.</b></p>
<b>Loop Style</b>	<p>Define how the video is played.</p> <ul style="list-style-type: none"> <li>• <b>NoLoop</b>: plays all the videos in the playlist, then stops.</li> <li>• <b>LoopOne</b>: repeats the first video in the playlist.</li> <li>• <b>LoopAll</b>: repeats the entire playlist.</li> <li>• <b>Random</b>: plays the videos in a random order.</li> </ul>



Note: The Media Player widget only works with some HMI devices (see ["HMI devices capabilities" on page 613](#)). It doesn't work the HMI Client.



Note: You can have only one Media Player widget in a page.

## Supported video encoding

Two groups of codecs are supported:

- DSP based video codecs
- Software video codecs



List of HMI devices that support the DSP (video hardware acceleration) is available on ["HMI devices capabilities" on page 613](#).

## DSP video codecs

These include:

- H264 using AVI/MP4 container, CABAC off and Level 3 (suggested)
- MPEG4 using MP4 container



*On WinCE devices BSP v1.55 or greater is required  
On Linux devices BSP v1.0.269 or greater is required*

## Software video codecs

This is only:

- Microsoft MPEG4 v3 using an AVI container.

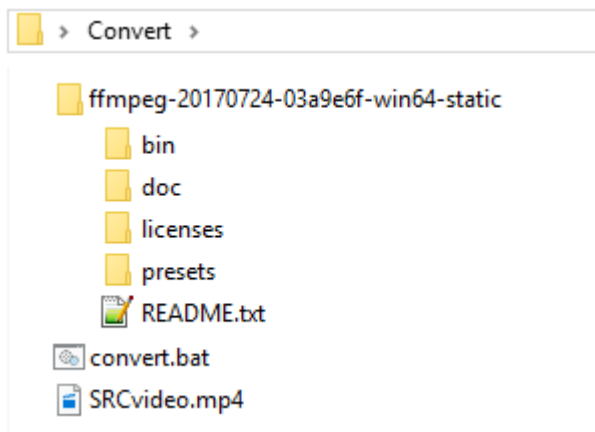


Be aware that video performance are depending from the chosen resolution, bit rate and device capabilities. If video rendering is not smooth, try to reduce the resolution or the bit rate of your video.

The videos encoded with Microsoft MPEG4 v3 are not using the hardware acceleration and have more limitations. To prevent the videos from running jerky, a maximum resolution of 640x512 pixels and a bit rate of 1300 kb/s are suggested. In addition, the size of the Media Player widget used on the page should have the same size as the videos in the play list, in order to avoid up scaling and down scaling. Audio is not supported.

## Converting a video

The FFMPEG ([www.ffmpeg.org](http://www.ffmpeg.org)) can be used to convert a video into the correct codec supported from the HMI device. Using the folder structure of the below picture, the following batch file could be used to convert any video file.



```

convert.bat x
1 @echo off
2 set FFMPEG=ffmpeg-20170724-03a9e6f-win64-static\bin\ffmpeg.exe
3
4 %FFMPEG% -i SRCvideo.mp4 ^
5         -y ^
6         -an ^
7         -s 240x160 ^
8         -b:v 4200k ^
9         -maxrate 4200k ^
10        -c:v libx264 ^
11        -profile:v baseline ^
12        -level:v 3 ^
13        -bufsize 3000k ^
14        -minrate 0 ^
15        -f avi ^
16        -preset slow ^
17        HMIvideo.avi
18
19 pause
20

```



Now you can open the converted video with a standard video player, such as Windows Media Player and check the quality. You can add the resulting video to the play list of the Media Player widget.



Note : The FFMPEG tool is not distributed with the PB610 Panel Builder 600.

## Using Media Player in JavaScript

The Media Player widget can be also referenced in JavaScript programs with the following syntax:

```
//get the mediaplayer widget.  
var mediaWgt = page.getWidget('MediaPlayerWgt2');  
//load the play list  
mediaWgt.setProperty('medialist', '/Storage Card/demo_3.avi,/Storage Card/video1_  
3.avi');  
// set the loopstyle 0 - no loop, 1 - loop one, 2- loop all, 3 - random  
mediaWgt.setProperty('loopstyle', 2);  
//start playing the first file.  
mediaWgt.mediapath = '/Storage Card/demo_3.avi';
```

See "[JavaScript](#)" on [page 539](#) for details on how to work with JavaScript.

## Browser widget

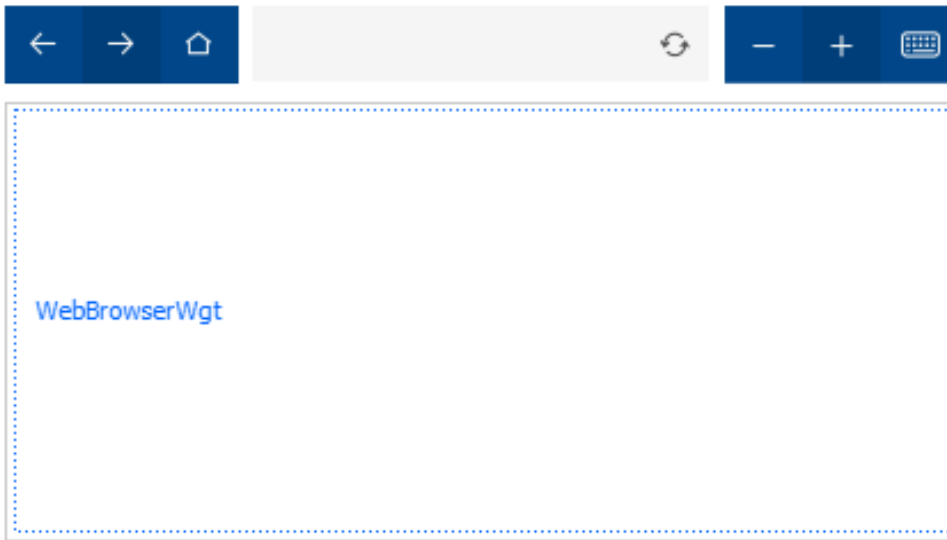
**Path:** *Widget Gallery > Media > Web Controls*

Use this widget to embed web pages into your HMI device pages. This is an HTML5 compatible browser widget based on the WebKit/QT4 (HMI devices with BSP v1.0) or WebEngine/QT5 (HMI devices with BSP v1.3).

This is an HTML5 compatible browser widget, based on WebEngine. Can be used to embed web pages into the HMI device pages. On WinCE HMI devices it is based on the WebKit.



Note: The WebKit library is available as a plugin (see "[Plug-in](#)" on [page 86](#) for details) to download to the HMI Runtime only when required. This allows you to save around 3 MB of space if the widget is not required in your project.



Parameter	Description
<b>Home Page</b>	Default URL to open when widget is shown on the page.
<b>Zoom to Fit</b>	Automatically scales content to the size of view area.
<b>Time out</b>	Page load timeout in seconds.
<b>Show Progress cursor</b>	Shows/hides loading cursor
<b>Clear History</b>	Automatic history clear on load
<b>Accept Language</b>	HTTP header tag that specifies the preferred languages e.g. "IT; en-US, EN"
<b>Save Cookie</b>	Select the location where cookie can be saved. Leave the field empty if you don't want to save the cookies

Gestures are supported to scroll the page, move the page, zoom in/out.

## Hyper Link

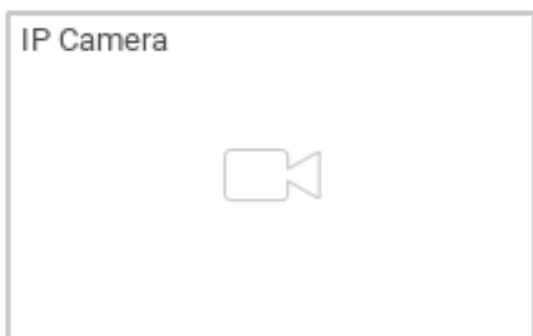
Path: **Widget Gallery**> **Media**> **Hyper Link**

n **Hyper Link** widget is available to create pages hyperlinks. Once clicked these links notify to the browser widget that a particular web page is to be loaded.

## IP Camera widgets

Path: **Widget Gallery**> **Media**> **IP Camera**

Use these these widgets to show images captured from an IP Camera or a video stream.



Parameter	Description
<b>Camera URL</b>	URL of the IP Camera when used in JPEG format.
<b>Refresh Rate</b>	Number of JPEG images for second allowed. Max rate = 1 fps.
<b>User Name</b>	Name of user allowed to access the camera. Set this parameter when access to the camera is password protected.
<b>Password</b>	Password to access the camera.
<b>MJPEG Camera URL</b>	URL of MJPEG streaming (for example, http://192.168.0.1/video.cgi)

When this widget is used to stream HTTP MJPEG, **Camera URL** and **Refresh Rate** are ignored.

Performance of streaming is not fixed and depends on many factors such as: frame size, frame compression level, CPU of HMI device, quality of IPCamera. Based on these factors the widget can reach up to 25 fps.

You can add multiple IP Camera widgets, but this will reduce the frame rate for each widget.

## Supported IPCameras

The following IP Cameras have been tested so far:

IPCamera	Protocol	URL
<b>Apexis APM-J901-Z-WS PTZ IP Camera</b>	MJPEG	http://{ip_address}/videostream.cgi
	HTTP	http://{ip_address}/snapshot.cgi
<b>AXIS M3027-PVE Network Camera</b>	MJPEG	http://{ip_address}/axis-cgi/mjpg/video.cgi
	HTTP	http://{ip_address}/axis-cgi/jpg/image.cgi
<b>DAHUA DH-IPC-HD2100P-080B 1.3mp Outdoor Vandalproof</b>	HTTP	http://{ip_address}:9988/onvif/media_service/snapshot
<b>D-Link DCS-5605 PTZ</b>	MJPEG	http://{ip_address}/video/mjpg.cgi
<b>D-Link DCS-900W IP Camera</b>	MJPEG	http://{ip_address}/video.cgi

IPCamera	Protocol	URL
<b>D-Link DCS-932L</b>	MJPEG	http://{ip_address}/video.cgi
<b>Edimax IC-7100P PTZ</b>	MJPEG HTTP	http://{ip_address}/mjpg/video.mjpg http://{ip_address}/picture.jpg
<b>Foscam FI8916W</b>	MJPEG HTTP	http://{ip_address}/videostream.cgi http://{ip_address}/snapshot.cgi
<b>Foscam FI9803 EP</b>	MJPEG	http://{ip_address}:88/cgi-bin/CGIStream.cgi?cmd=GetMJStream&usr={user}&pwd={pass}  NOTE: <ul style="list-style-type: none"> <li>• port 88 may be different as per IP Camera settings</li> <li>• {user} = username defined into IP Camera settings</li> <li>• {pass} = password defined into IP Camera settings</li> </ul>
<b>Hamlet HNIPCAM IP Camera</b>	MJPEG HTTP	http://{ip_address}/video.cgi http://{ip_address}/image.jpg
<b>MOXA VPort 254</b> (Rugged 4-channel MJPEG/MPEG4 industrial video encoder)	MJPEG HTTP	http://{ip_address}/moxa-cgi/mjpeg.cgi http://{ip_address}/moxa-cgi/getSnapShot.cgi?chindex=1
<b>NVS30 network video server</b>	MJPEG HTTP	http://{ip_address}:8070/video.mjpeg http://{ip_address}/jpg/image.jpg
<b>Panasonic WV-Series Network Camera</b>	MJPEG	http://{ip_address}/cgi-bin/mjpeg
<b>Ubiquiti UniFi Video Camera</b>	HTTP	http://{ip_address}:7080/images/snapshot/camera/{camera_guid}?force=true  NOTE: <ul style="list-style-type: none"> <li>• {camera_guid} can be found into IP Camera Webpage</li> <li>• port 7080 may be different as per IP Camera settings</li> </ul>
<b>Zavio F3210 2MP Day &amp; Night Compact IP Came</b>	MJPEG HTTP	http://{ip_address}/stream?uri=video.pro3 http://{ip_address}/cgi-bin/view/image?pro_0  NOTE: <ul style="list-style-type: none"> <li>• MJPEG video streaming can be configured selecting "video profile 3" with 640x480 resolution into IP Camera settings.</li> </ul>

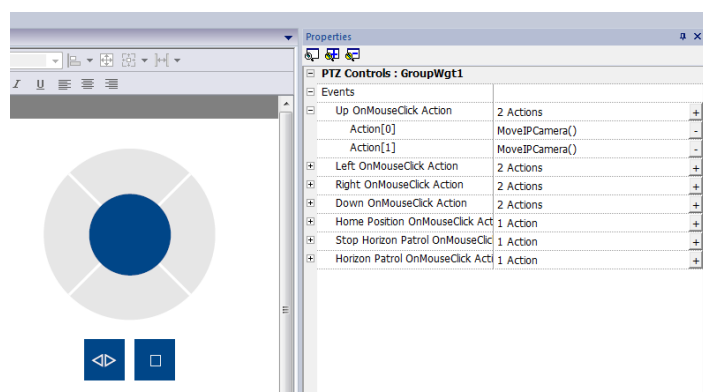
## PTZ Controls widget

PTZ (pan–tilt–zoom) cameras are cameras capable of remote directional and zoom control.



**Not yet available in the new gallery (use the old gallery to get this widget)**

The PTZ Controls widget uses the MoveIPCamera action to send HTTP/cgi commands to the PTZ IP Camera.



## Authentication methods

The authentication method is automatically set by the camera web server to which the widget connects. Authentication methods supported are:

- Basic
- NTLM version 1
- Digest-MD5

## Web Browser

On the Web Browser, only the "Basic Authentication" mode is supported. When used, the IP Camera with authentication shows a pop dialog to enter login and password.

Widget is supported by Chrome and Firefox, we found issues using the current version of the Edge browser.

## Table widget

Path: **Widget Gallery**> **Basic**> **Table**

Use this widget to create a table with data provided from a data source.



**Available only on Linux devices (See: "[HMI devices capabilities](#)" on page 613)**

To configure a table:

1. Put a table widget on the screen and configure the template of the table.
2. Add widgets into cells to configure one or more rows that will be used as row templates when the table will be filled with data provided from the data source.

3. Select a data source that will be used to fill the rows of the table
4. Define the links from widgets and data source.

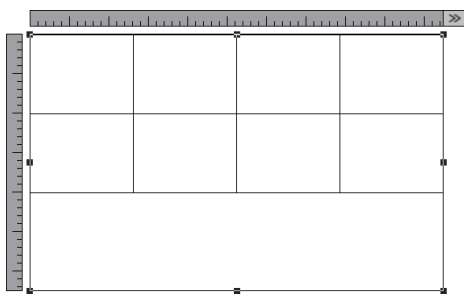
## Configure the table widget

Table widget has two states:

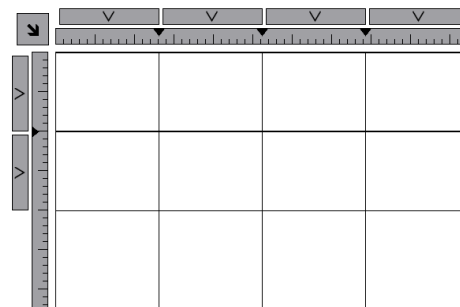
- View mode
- Edit mode.

Click on the table to manage the widget in view mode, double click to enter in the edit mode. To exit and return to view mode click outside the table.

### View Mode



### Edit Mode



### View Mode

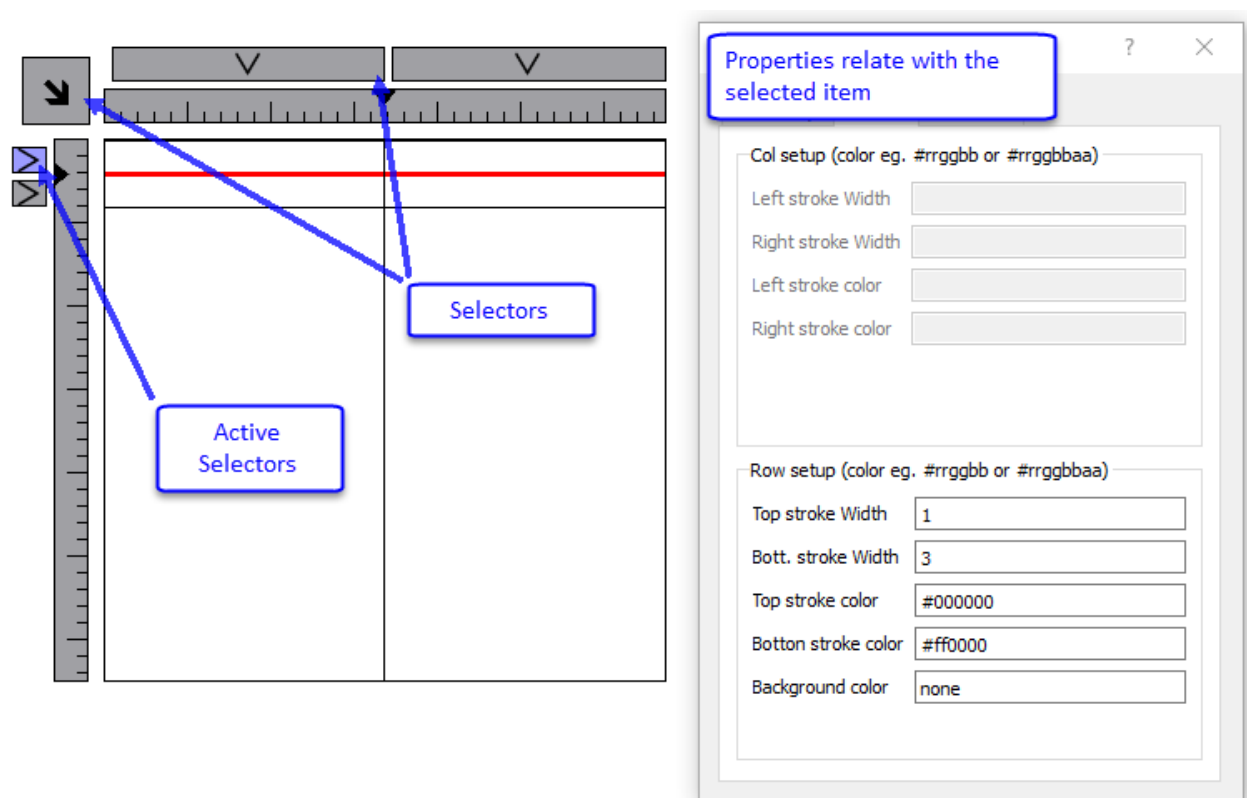
In view mode, you can configure the table layout. Drag and drop the table onto the page, resize the table, define number of template rows, number of columns and the main table properties.

Properties	
TableGroupWgt : TableWgt	
Current selected row	-1
Precached Pages	0
+ Data Source	
- Grid Layout Group	
Num rows	2
Num columns	4
Horizontal overflow	Scroll
Horizontal underflow mode	Center
Scrollbar Handle Color	[0, 70, 136]
Scrollbar Background Color	[237, 237, 237]
Scrollbar image	
Scrollbar offset	0
Scrollbar size	10
Scrollbar autohide	Auto
External border mode	Fixed
Margin collapsed	true
External margin width	1
External margin color	[200, 200, 200]
Horizontal scroll position	0
Vertical scroll position	0
+ Events	
+ General	
+ Position	

## Edit Mode

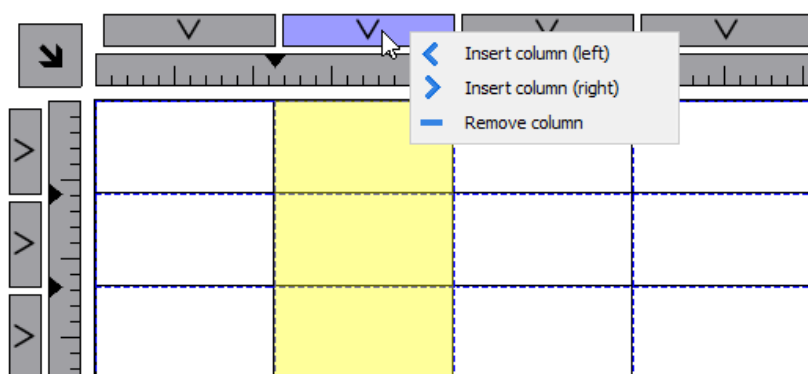
In edit mode, it is possible to configure the format and the content of each cell of the table. Each row of the table will act as a row template.

To configure the look of the table, click on table's selectors to select the item to configure.



## Add or remove rows or columns

To add or remove rows or columns, double click over the grid to enter in edit mode and right click over column or row selector to open the context menu.



## Merge or split rows or columns

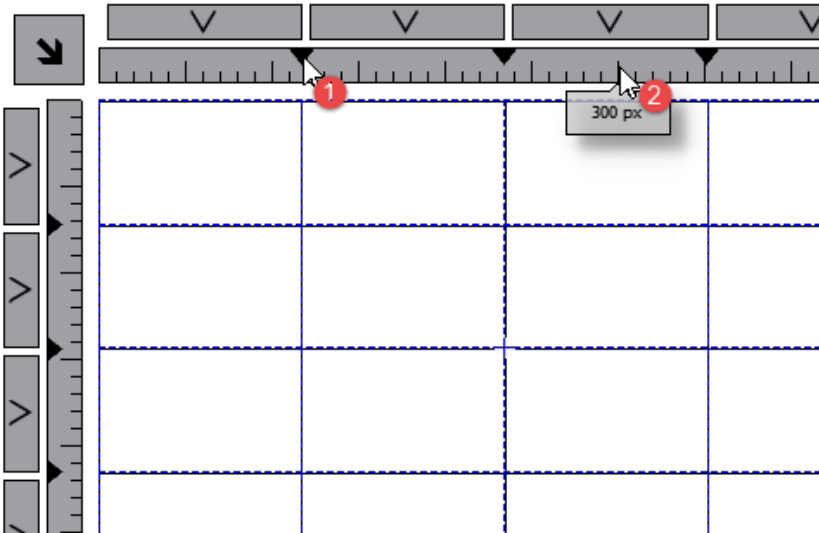
To merge or split rows or columns, double click over the grid to enter in edit mode and move the cursor over the ribbons:

- Double click the black triangle to merge the two adjacent rows or columns (1)

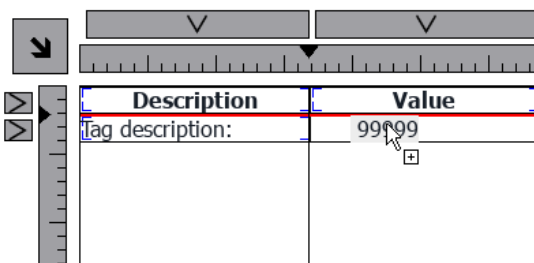


Note that merge is possible only with an empty row or column.

- Double click on ribbon to split the selected row or column (2)



To configure the contents of cells, drag and drop the widgets inside the cells.



If you need more widgets inside a single cell, create a group of widgets and copy the group from the page to the cell.

## Configuring the data source

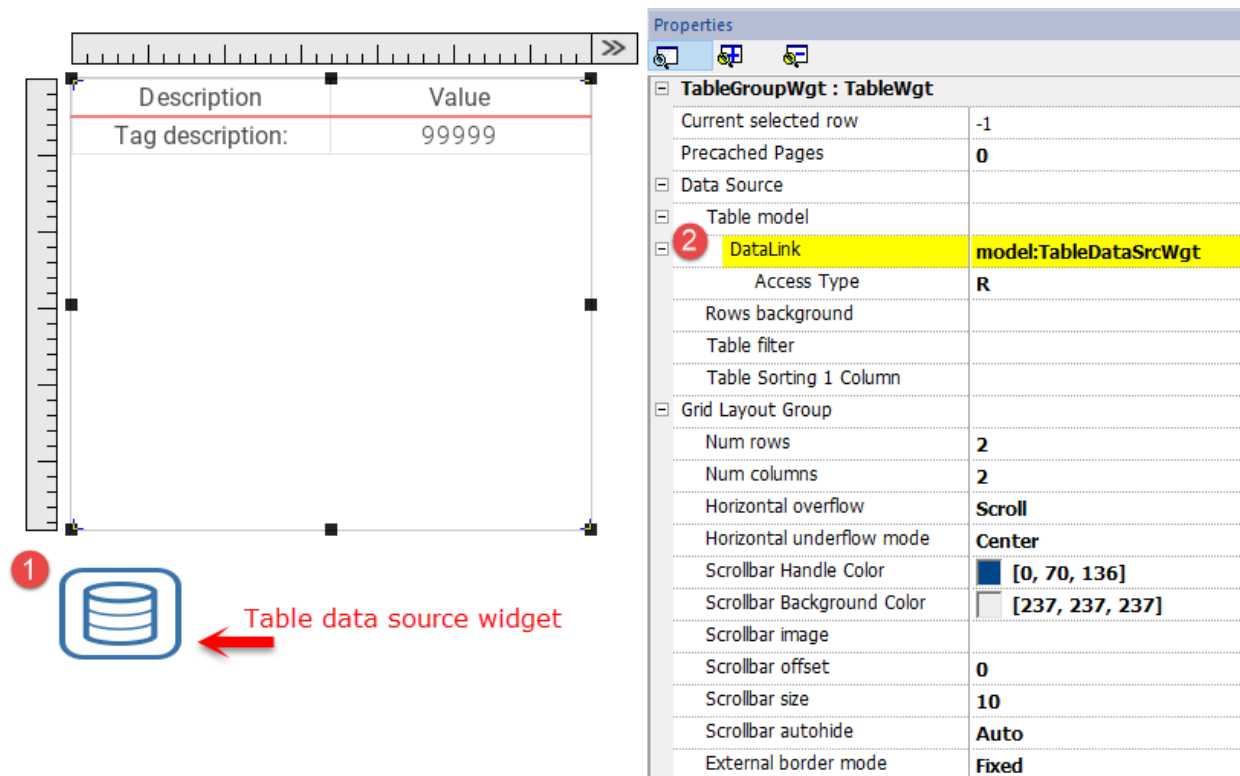
The data source, that provide the data to fill the table, could be a Table Data Source Widget or a JavaScript Object.

### Table Data Source Widget

Path: **Widget Gallery**> **Basic**> **Table**

1. Drag and drop a *Table Data Source Widget* onto the page
2. Set the *Table Model* parameter to link at the data source.





The screenshot shows a table widget with the following content:

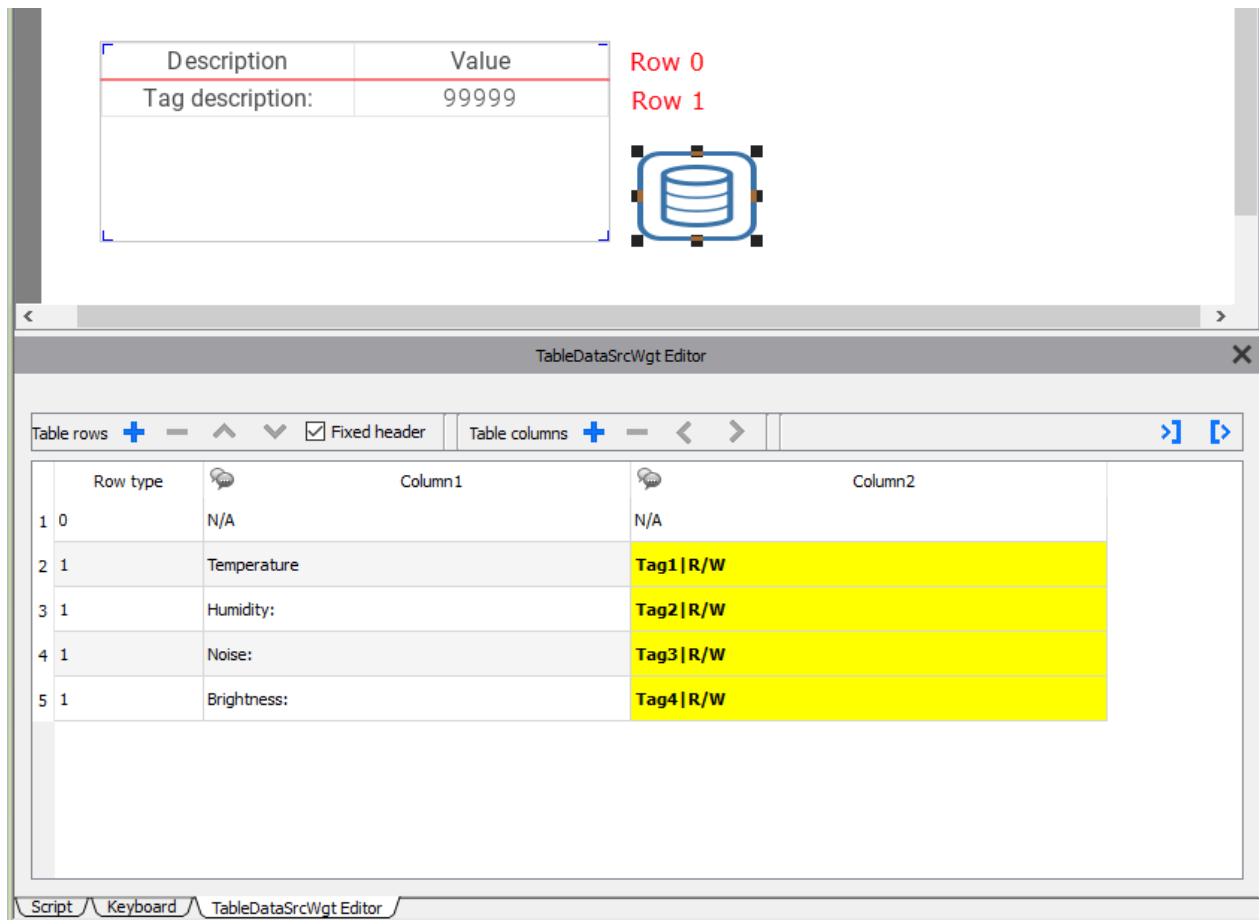
Description	Value
Tag description:	99999

The Properties panel for **TableGroupWgt : TableWgt** includes the following settings:

- Current selected row: -1
- Precached Pages: 0
- Data Source: (empty)
- Table model: (empty)
- DataLink**: model:TableDataSrcWgt (highlighted with a red circle '2')
- Access Type: R
- Rows background: (empty)
- Table filter: (empty)
- Table Sorting 1 Column: (empty)
- Grid Layout Group: (empty)
- Num rows: 2
- Num columns: 2
- Horizontal overflow: Scroll
- Horizontal underflow mode: Center
- Scrollbar Handle Color: [0, 70, 136]
- Scrollbar Background Color: [237, 237, 237]
- Scrollbar image: (empty)
- Scrollbar offset: 0
- Scrollbar size: 10
- Scrollbar autohide: Auto
- External border mode: Fixed

Select the Data Source and inside the TableDataSrcWgt Editor add the rows and columns that are needed. In the following example, we have defined two row templates:

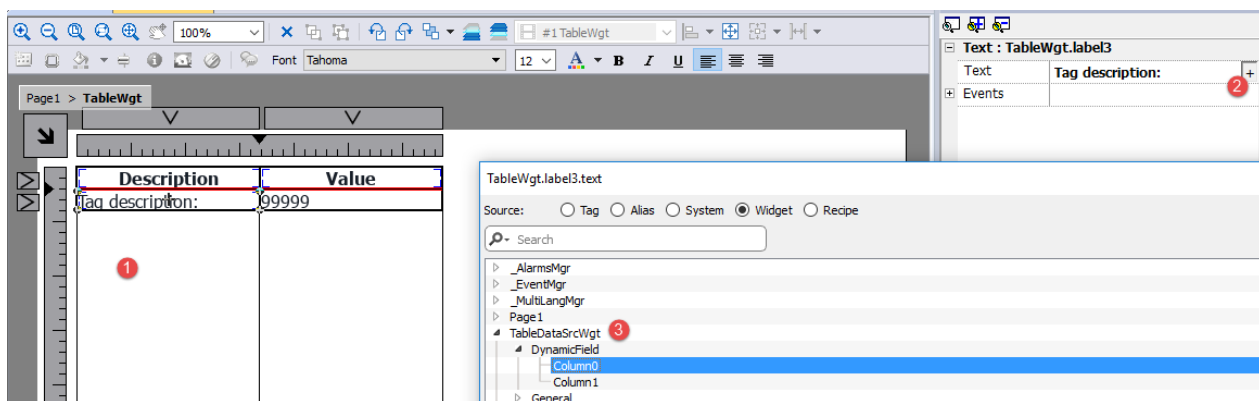
- Row 0  
Header of the table. Contains only static text.
- Row 1  
Template of rows with data. On the first column we added a label that will contain the description and on the second column a field that will contain the value.



Each row must be assigned a row type. The row will take on the format of the corresponding row template. Widgets that were placed in each cell of the row template will appear in rows of that type.

**Define links with data source**

1. Double click over the Table widget to enter in edit mode and select a widget
2. Select the property that is to be read from the data source
3. Select the column of the data source that will provide the data



The below picture is showing how our example will be rendered at runtime

Description	Value
Temperature	111
Humidity:	222
Noise:	333
Brightness:	444

### Fixed Header

If you want the first row will be not scrollable, check the "Fix Header" check box on Data Source toolbox or set true the "Show Header" propret inside the Data Source properties panel (note the parameter is available only in advanced view).

### Column override

You can use an array of integers to define or modify the columns order at runtime. When you use this property, be sure to attach an array of integer and set the index to -1 (to select the entire array).

Column override (array of int):

0	1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---	---

Description:	Col 1	Col 2	Col 3	Col 4
00	1	2	3	4
Row 1	Data 1	Data 11	Data 111	Data 1111

Column override (array of int):

0	1	3	3	7	5	6	7	8
---	---	---	---	---	---	---	---	---

Description:	Col 1	Col 3	Col 3	Col 7
00	1	3	3	7
Row 1	Data 1	Data 111	Data 111	fdqfd

### Fetch Mode

When enabled, the table will load a minimal number of rows to fill the view. Allows very fast page loading for large tables. When scrolling, when the table reaches the end, new rows are loaded and you can scroll through the database.

The "Rows loaded" parameter gives you the possibility to choose how many rows need to be loaded per cycle. The higher the value, the longer it will take to load the page, but the more rows will be loaded ready to scroll (less load during scrolling).

### Multilanguage

To enable the Multilanguage support right click on the Multilanguage icon of the column. The icon will change color to indicate that the support is enabled.



Avoid enabling the Multilanguage support when not necessary to better performance.

Table rows		Table columns	
Row type	Column0	Column1	
1 0	N/A	N/A	
2 1	Temperature	Tag1   R/W	
3 1	Humidity:	Tag2   R/W	
4 1	Noise:	Tag3   R/W	
5 1	Brightness:	Tag4   R/W	

### Import/Export Data Source

The configuration of the Data Source can be imported/exported using xml files

Table rows		Table columns	
Row type	Column0	Column1	
1 0	N/A	N/A	
2 1	Temperature	Tag1   R/W	

### JavaScript Object

In alternative to the Data Source Widget, for data to fill the table could be provided from a JavaScript Object. In this case, we have to fill an array of elements with the data to use and assign the array to the table widget.

```
var myTable = page.getWidget("TableWgt1");
myTable.model = model;
```

**model** is an array of elements with the table definition and data. The first element of the array will contain the template of the rows while the other elements will contain the data to fill in the rows of the table

```
model[0] = row_templates; // row templates
model[1] = row_data1; // data of the row1
model[2] = row_data2; // data of the row2
model[3] = row_data3; // data of the row3
model[4] = row_data4; // data of the row4
model[5] = row_data4; // data of the row5
```

The **row templates** is a multi dimensional array where each array defines the datalink of one template row.

On the below example, we have a template for two rows.

```
var row_templates = {
  _h : [
    [ [], [], [] ], //rowType = 0
    [ ["text"], ["value"] ] //rowType = 1
  ]
}
```

```
}

```

The first row has two columns that do not contain data links. We use this template for the header on the first row of our table.

The second row defines the template of one row with the “text” property of the widget into the first column and the “value” property of the widget into the second column. They will be dynamically filled using the data provided inside the model variable.

On the below example we define a **row of data**

```
var row_data = {
  _t : 1,
  _v : ["Temperature:", { _c : "dl" , s : "_TagMgr", a : "Tag1", i: 0, m : 2 }]
}
```

The first element is the row template to use while the second element is the array with the data to use. In our example “Temperature:” is the text to use inside the widget on the first column, while the other element is a datalink that will provide the value to fill the value property of the widget into the second column.

The datalink element:

Parameter	Description
<b>_c : "dl"</b>	Identify the element as a Datalink
<b>s : "_TagMgr"</b>	Specify the source of data is the Tag Manager
<b>a : "Tag1", i: 0, m:2</b>	Specify tag name and index (necessary when the tag is an array) and the read/write mode <ul style="list-style-type: none"> <li>• m=0 is Read Only</li> <li>• m=1 is Write Only</li> <li>• m=2 is Read/Write</li> </ul>

The below JavaScript code will generate the same table of the previous example using the Table Data Source Widget

```
var model = [];

var row_templates = {
  _h : [
    [ [] , [] ], //rowType = 0
    [ ["text"] , ["value"] ] //rowType = 1
  ]
}

var row_data1 = {
  _t : 0,
  _v : [],
  _h : true
}
```

```

var row_data2 = {
  _t : 1,
  _v : ["Temperature:", { _c : "dl" , s : "_TagMgr", a : "Tag1", i: 0, m : 2 }]
}

var row_data3 = {
  _t : 1,
  _v : ["Humidity:",      { _c : "dl" , s : "_TagMgr", a : "Tag2", i: 0, m : 2 }]
}

var row_data4 = {
  _t : 1,
  _v : ["Noise:",        { _c : "dl" , s : "_TagMgr", a : "Tag3", i: 0, m : 2 }]
}

var row_data5 = {
  _t : 1,
  _v : ["Brightness:",   { _c : "dl" , s : "_TagMgr", a : "Tag4", i: 0, m : 2 }]
}

model[0] = row_templates;
model[1] = row_data1;
model[2] = row_data2;
model[3] = row_data3;
model[4] = row_data4;
model[5] = row_data5;

var myTable = page.getWidget("TableWgt1");
myTable.model = model;

```

Note the first row (row\_data1) contains the directive `_h: true` to avoid the first line will be scrollable.

```

var row_data1 = {
  _t : 0,
  _v : [],
  _h : true
}

```

## Multilanguage

A multi languages text can be entered using the below element:

```
{ _c : "ml" , mltext : {"en-US" : "Temperature:" , "it-IT" : "Temperatura:"} }
```

Parameter	Description
<code>_c : "ml"</code>	Identify the element as a Multilanguage text
<code>mltext : { ... }</code>	List of couples: "ID Language":"Text"

Parameter	Description
	Example: <ul style="list-style-type: none"> <li>• "en-US" : "Temperature:"</li> <li>• "it-IT" : "Temperatura:"</li> </ul>

Example:

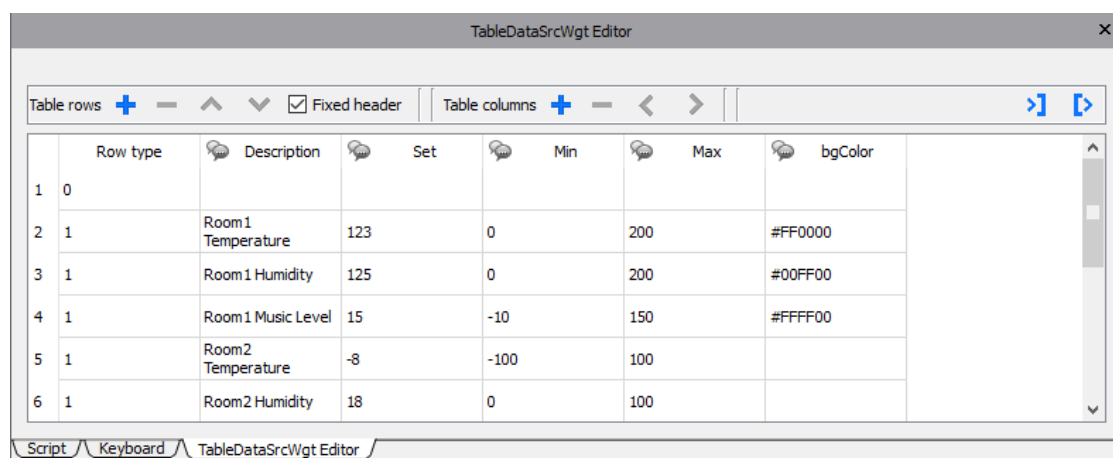
```
var row_data2 = {
  _t : 1,
  _v : [ { _c : "ml" , mltext : { "en-US" : "Temperature:",
                                "it-IT" : "Temperatura:"} },
        { _c : "dl" , s : "_TagMgr", a : "Tag1", i: 0, m : 2 }
      ]
}
```

## Row background color

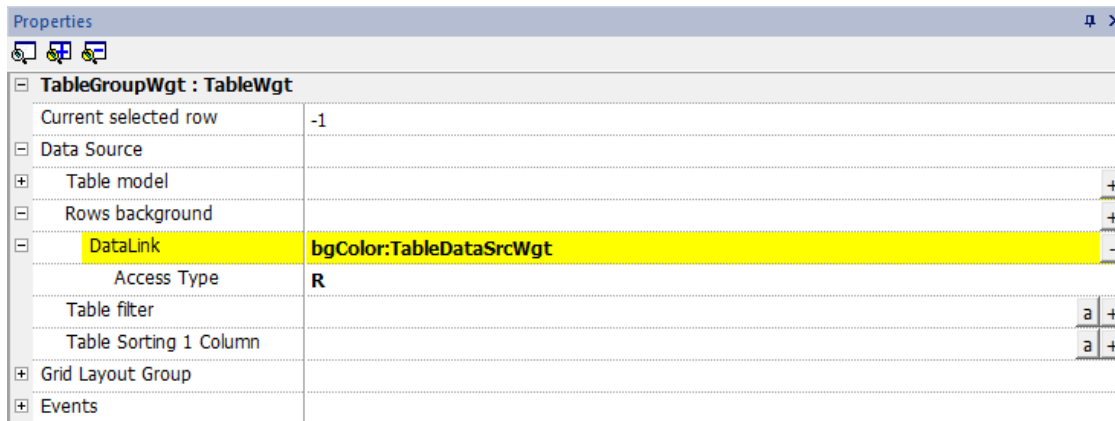
Using the Rows background parameter is possible define the column of the Data Source Widget that will contains the background color of the associate row.

To configure the background color of the rows of the table:

1. Add a new column inside the Data Source Widget to contain the background color of each row
2. Configure the "Row background" color parameter of the Table to point to the color column of the Data Source Widget



Row type	Description	Set	Min	Max	bgColor
0					
1	Room1 Temperature	123	0	200	#FF0000
1	Room1 Humidity	125	0	200	#00FF00
1	Room1 Music Level	15	-10	150	#FFFF00
1	Room2 Temperature	-8	-100	100	
1	Room2 Humidity	18	0	100	



Filter:

Description	Set	Min	Max
Room1 Temperature	123	0	200
Room1 Humidity	125	0	200
Room1 Music Level	15	-10	150
Room2 Temperature	-8	-100	100
Room2 Humidity	18	0	100
Room2 Music Level	12	0	150
Room3 Temperature	15	0	150
Room3 Humidity	134	0	500

### Table Filter

Content visible inside the table can be filtered using the “Table Filter” property. On datalink you can use a formula (see ["Formula" on page 55](#) chapter for additional details) to define the criteria to use to filter the data.

Each row of the table will be visible only when the Datalink of the Table Filter return true value.

#### Example 1

If you want choice to see only the rows that contain “something” inside the Description column, you can use the below formula:

```
= $Contains ($ ('Description:TableDataSrcWgt'), $ ('value:SearchOnTable'))
```

Where

- 'Description:TableDataSrcWgt' is a Dynamic Field of the Data Source Widget used from the table to identify the column to check
- 'value:SearchOnTable' is the value of a text field that will contains the string to search



Filter:  ✕

Description	Set	Min	Max
Room1 Temperature	123	0	200
Room1 Humidity	125	0	200
Room1 Music Level	15	-10	150
Room2 Temperature	-8	-100	100
Room2 Humidity	18	0	100
Room2 Music Level	12	0	150
Room3 Temperature	15	0	150
Room3 Humidity	134	0	500

Filter:  ✕

Room2 Temperature	-8	-100	100
Room2 Humidity	18	0	100
Room2 Music Level	12	0	150

Filter:  ✕

Room1 Temperature	123	0	200
Room2 Temperature	-8	-100	100
Room3 Temperature	15	0	150
Room4 Temperature	2	0	10

Properties ✕

**TableGroupWgt : TableWgt**

- Current selected row: -1
- Data Source
  - Table model: +
  - Rows background: a +
  - Table filter: +
  - DataLink: =\$Contains(\$('Description:TableDataSrcWgt'),\$('value:SearchOnTable')) -**
  - Access Type: R
  - Table Sorting 1 Column: a +
- Grid Layout Group: +
- Events: +

## Example 2

To use flags to define the parameters to expose inside the table:

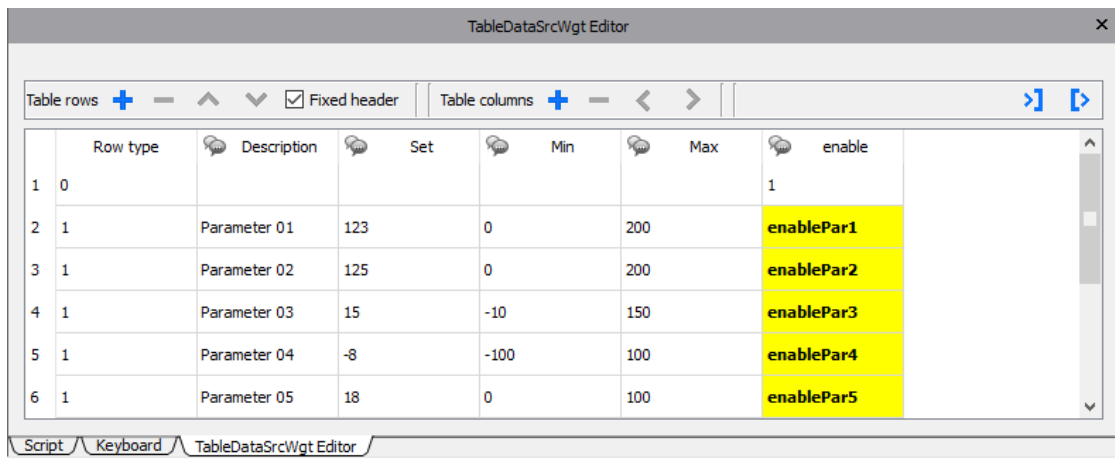
First, add a new column inside the Data Source that will contains the flags that will enable the associate row. Then, link the datalink of the table filter to the new column that contains the flags

enablePar1 = 1  
 enablePar2 = 1  
 enablePar3 = 1  
 enablePar4 = 1  
 enablePar5 = 1

Description	Set	Min	Max
Parameter 01	123	0	200
Parameter 02	125	0	200
Parameter 03	15	-10	150
Parameter 04	-8	-100	100
Parameter 05	18	0	100
Parameter 06	12	0	150
Parameter 07	15	0	150
Parameter 08	134	0	500

enablePar1 = 1  
 enablePar2 = 1  
 enablePar3 = 0  
 enablePar4 = 0  
 enablePar5 = 1

Description	Set	Min	Max
Parameter 01	123	0	200
Parameter 02	125	0	200
Parameter 05	18	0	100
Parameter 06	12	0	150
Parameter 07	15	0	150
Parameter 08	134	0	500
Parameter 09	44	0	50
Parameter 10	2	0	10



Properties	
TableGroupWgt : TableWgt	
Current selected row	-1
Data Source	
Table model	
Rows background	a +
Table filter	
DataLink	enable:TableDataSrcWgt1
Access Type	R
Table Sorting 1 Column	a +
Grid Layout Group	
Events	

## Table Sorting

To sort the rows of the table, select the column of the Data Source that you want to use to sort the table

- Sorting mode can be Ascendent or Descendent
- Sorting Rule can be Alphabetic or Numeric

Properties	
TableGroupWgt : TableWgt	
Current selected row	-1
Precached Pages	0
Data Source	
Table model	
Rows background	a +
Table filter	a +
Table Sorting 1 Column	
DataLink	Column0:TableDataSrcWgt
Access Type	R
Table Sorting 1 Mode	Ascendent
Table Sorting 1 Rule	Alphabet
Table Sorting 2 Column	a +
Grid Layout Group	
Events	
General	

Multiple sorting (STABLE sorting) is useful when a column has repetitions. You can use up to three sort columns.

### Example of sorting:

#### Alphabetic

Value
1
10
15
7

#### Numeric

Value
1
7
10
15



The table can be ordered even using the SetTableSortingColumn macro (see "SetTableSortingColumn" on page 242 for details).

### Horizontal scroll position

The "Horizontal scroll position" give the possibility to keep synchronized the horizontal scroll movements of two tables.

External margin color	[0, 0, 0]
Horizontal scroll position	0 <span style="float: right;">+</span>
DataLink	relHorScrollPos:GroupWgt2.TableWgt <span style="float: right;">-</span>
Access Type	R



**Horizontal scroll position parameter is available only in Advanced Properties View mode**

### Precached Pages

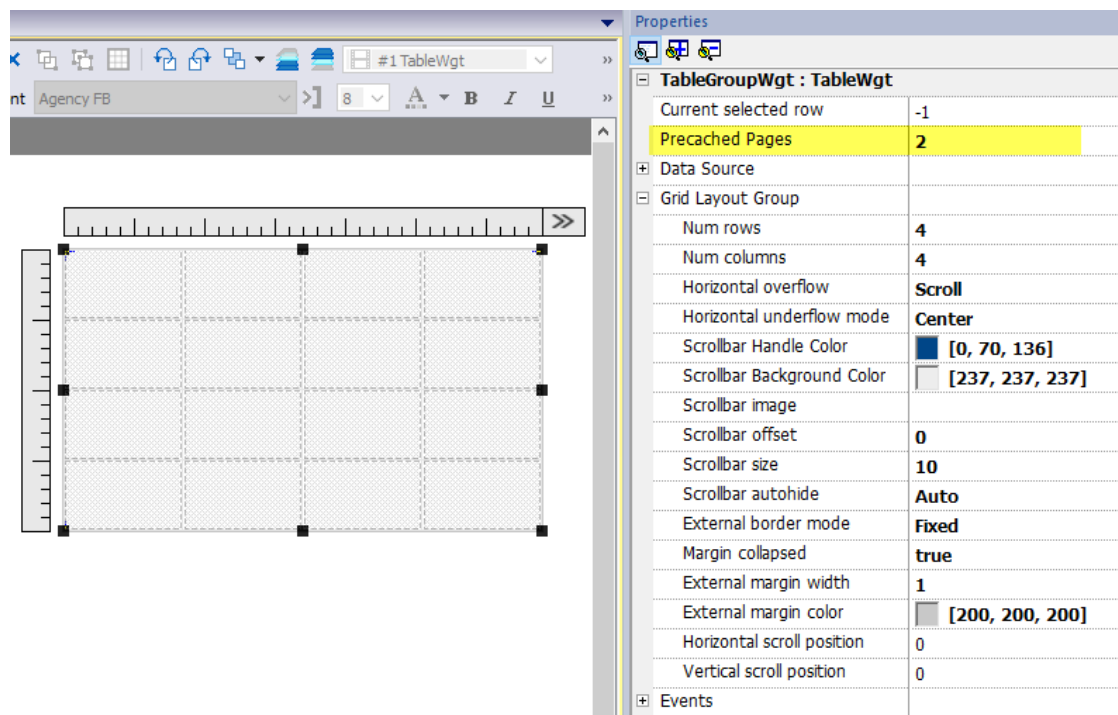
Normally the HMI Runtime retrieve only the data that will be visible into the display. To make table scrolling more pleasant, it may be useful to preload the data of the next and previous rows of the displayed ones. Using the Precached Pages parameter is possible define how many pages will be preloaded

- 0 = no pages preloaded
- N = number of pages to preload

Example:

Using a table with 4 rows and Precache Pages = 2

- Number of row to preload are 8 (2 pages x 4 rows)
- 4 rows before (to be ready to manage scroll table up)
- 4 row above (to be ready to manage scroll table down)



The screenshot shows a software interface with a table widget and its properties. The table has 4 rows and 4 columns. The 'Precached Pages' property is highlighted in yellow in the Properties panel, set to 2. The Properties panel also shows other settings like Num rows (4), Num columns (4), and Scrollbar Handle Color.

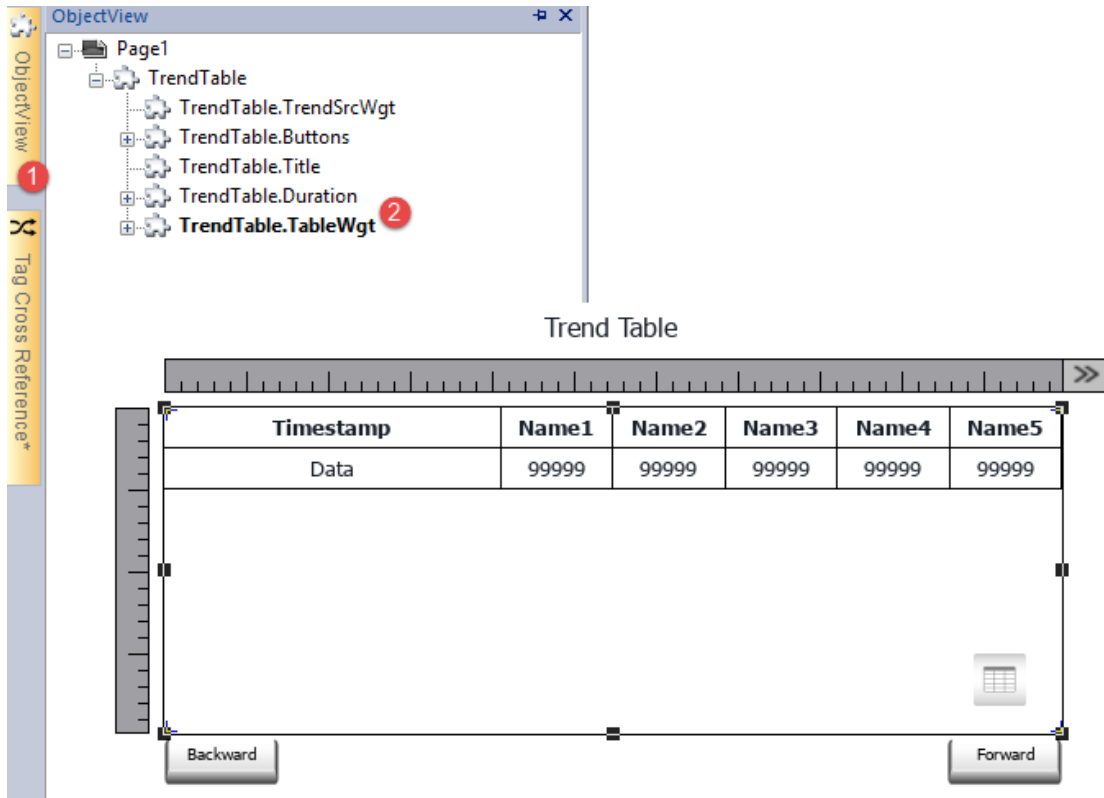
TableGroupWgt : TableWgt	
Current selected row	-1
<b>Precached Pages</b>	<b>2</b>
Data Source	
Grid Layout Group	
Num rows	4
Num columns	4
Horizontal overflow	Scroll
Horizontal underflow mode	Center
Scrollbar Handle Color	[0, 70, 136]
Scrollbar Background Color	[237, 237, 237]
Scrollbar image	
Scrollbar offset	0
Scrollbar size	10
Scrollbar autohide	Auto
External border mode	Fixed
Margin collapsed	true
External margin width	1
External margin color	[200, 200, 200]
Horizontal scroll position	0
Vertical scroll position	0
Events	



**Precached Pages parameter is available only in Advanced Properties View mode**

## Widgets that contain tables

Inside the gallery, there are widgets that contain tables, e.g. trend table, audit table, etc. To open the table's properties or the data source's properties you can use the Object View tab and select the component that you need to configure.



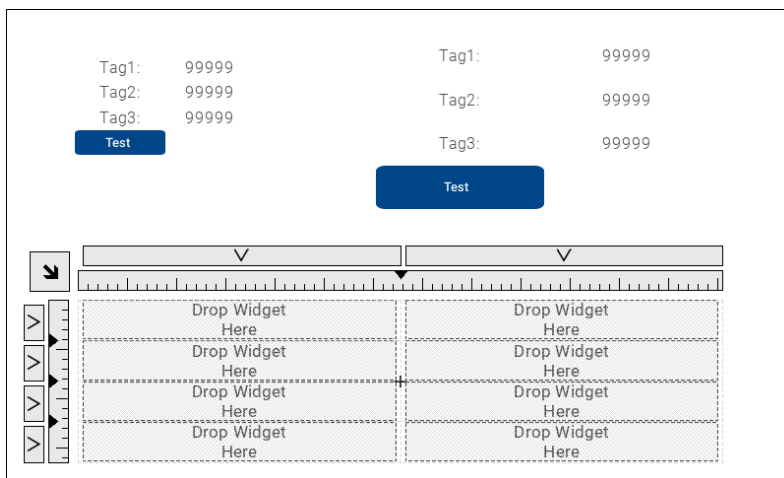
## Printing table

A table widget can be found and used from the print report gallery.

# Grid Layout widget

Path: **Widget Gallery**> **Basic**> **Layout**

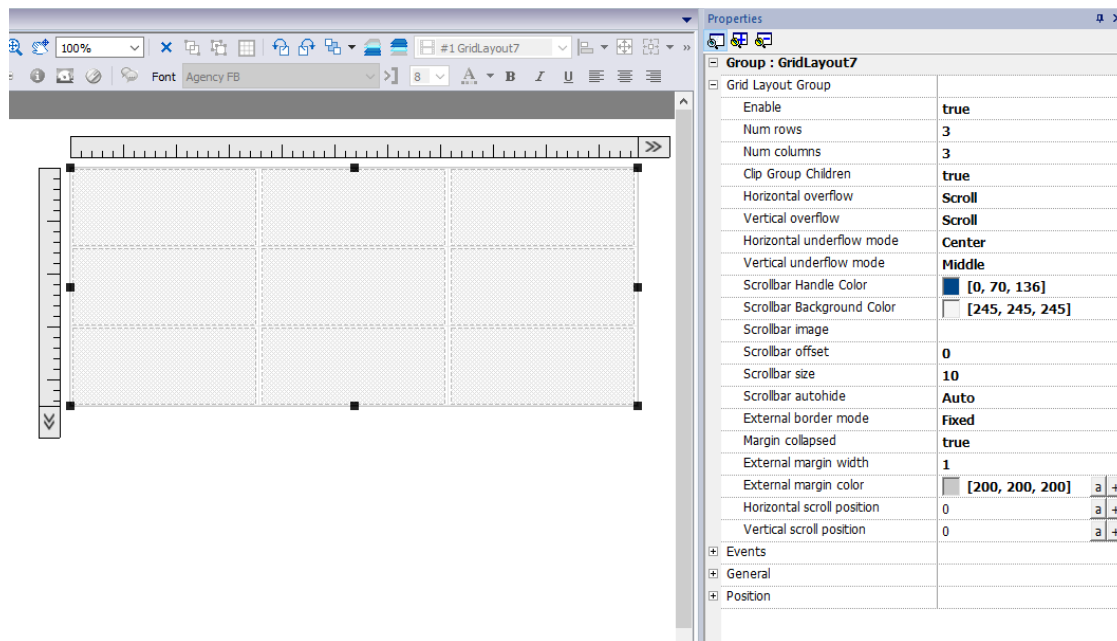
The grid layout is a widget that adds the possibility to configure the spatial relationships among the widgets.




There are several elements associated with the grid layout that can be configured:

- Grid properties
- Rows, Columns Properties
- Cells Properties

## Grid Properties

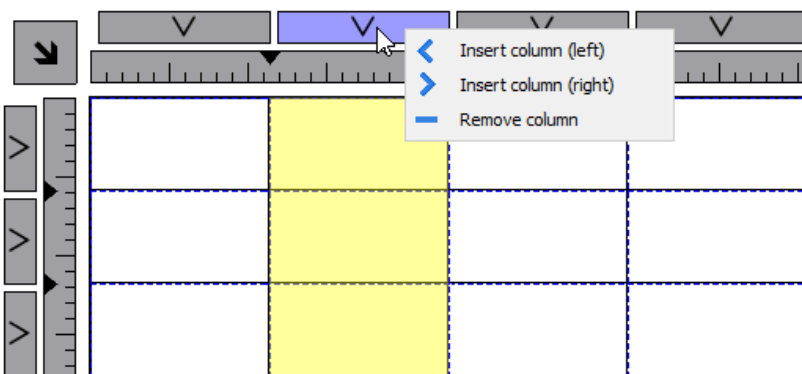


Parameter	Description
<b>Enable</b>	<p>Enable the grid layout.</p> <ul style="list-style-type: none"> <li>• true A grid will be generate around the widgets of the group</li> <li>• false Remove the grid layout</li> </ul>
<b>Num rows</b> <b>Num columns</b>	<p>Number of rows and columns of the grids.</p> <p> Rows and columns can be removed only if their cells are empty .</p>
<b>Clip Group Children</b>	<ul style="list-style-type: none"> <li>• true group's children are always clipped inside the group bounds (necessary for scrollbar).</li> <li>• false children are not clipped</li> </ul>
<b>Horizontal overflow</b> <b>Vertical overflow</b>	<p>This parameter define the behavior of the grid when it is too small to contain all rows and columns.</p> <ul style="list-style-type: none"> <li>• Hidden Rows and columns that do not fit into the grid are not displayed</li> </ul>

Parameter	Description
	<ul style="list-style-type: none"> <li>• Scroll When the grid is too small to hold all the defined rows and columns, the scroll bars can be used to shift the content of the grid.</li> </ul>
<b>Horizontal underflow mode</b> <b>Vertical underflow mode</b>	<p>This parameter defines the behavior of the grid when it is larger than the size defined for the rows and columns</p> <ul style="list-style-type: none"> <li>• Blocked The grid can not be made larger than the maximum size of rows and columns</li> <li>• Left, Center, Right - Top, Middle, Bottom Defines the position of the widgets when cells are bigger than the maximum defined sizes</li> </ul>
<b>Scrollbar Handle Color</b> <b>Scrollbar Background Color</b> <b>Scrollbar image</b> <b>Scrollbar offset</b> <b>Scrollbat size</b> <b>Scrollbar autohide</b>	Parameters to define look and position of the scroll bars
<b>External border mode</b>	<p>Set the display method of the outer frame.</p> <ul style="list-style-type: none"> <li>• Fixed The outer frame is displayed according to the widget size.</li> <li>• Auto Display the outer frame according to the table.</li> </ul>
<b>Margin collapsed</b>	Collapse all left-right and top-bottom margin using the parameters of the stroke with greater width.
<b>External margin width</b> <b>External margin color</b>	External margin parameters

**Add or remove rows or columns**

To add or remove rows or columns, double click over the grid to enter in edit mode and right click over column or row selector to open the context menu.





## Merge or split rows or columns

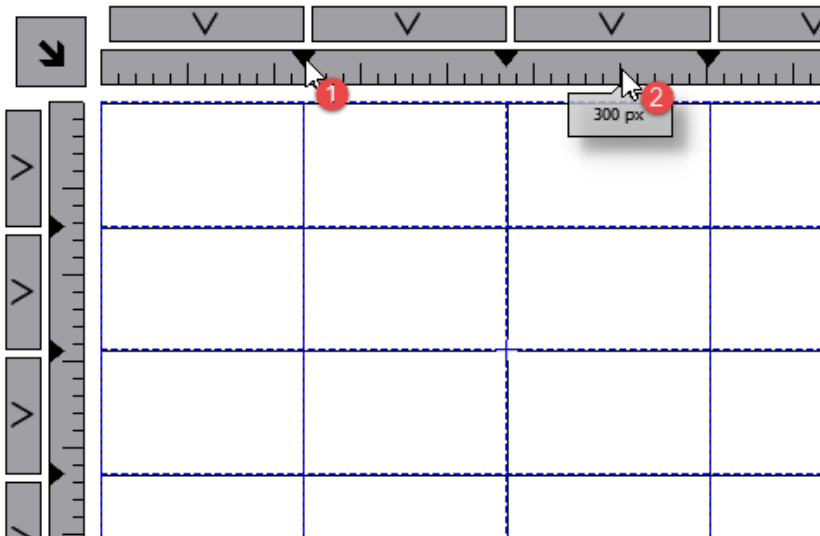
To merge or split rows or columns, double click over the grid to enter in edit mode and move the cursor over the ribbons:

- Double click the black triangle to merge the two adjacent rows or columns (1)



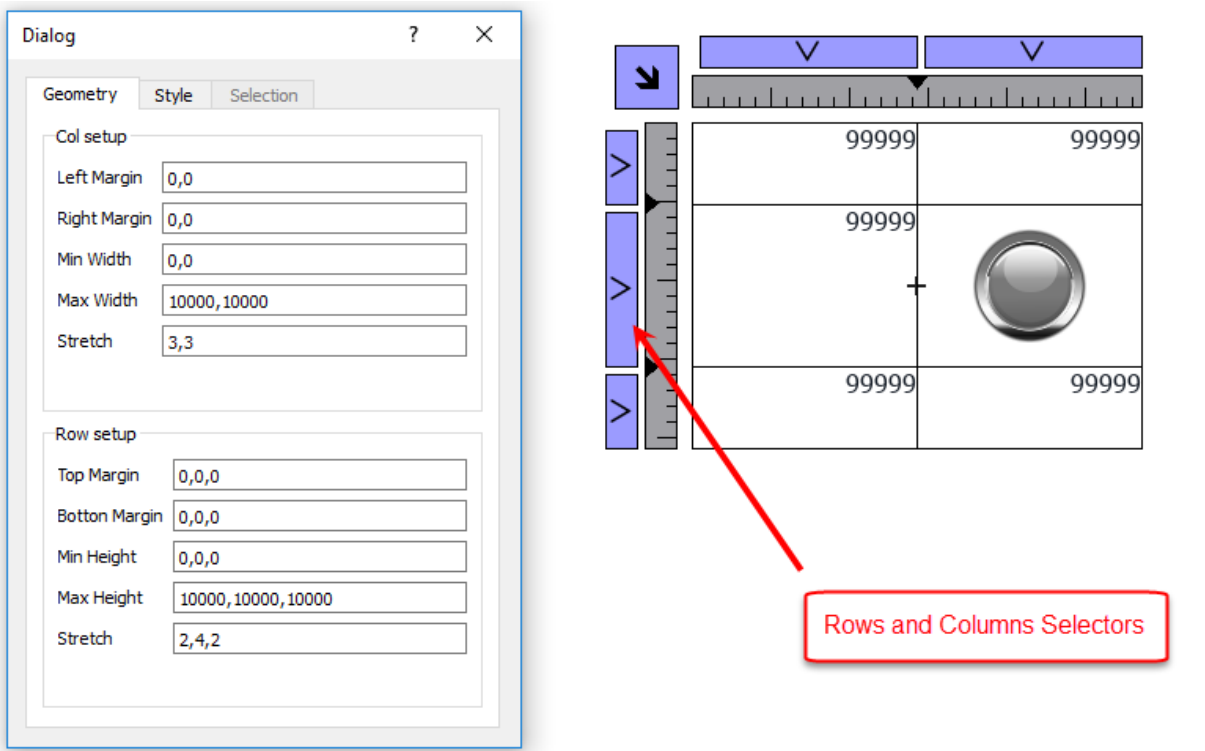
Note that merge is possible only with an empty row or column.

- Double click on ribbon to split the selected row or column (2)

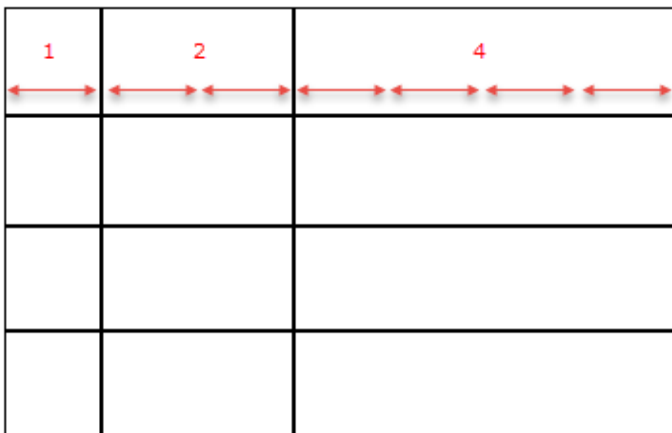


## Rows, Columns Properties

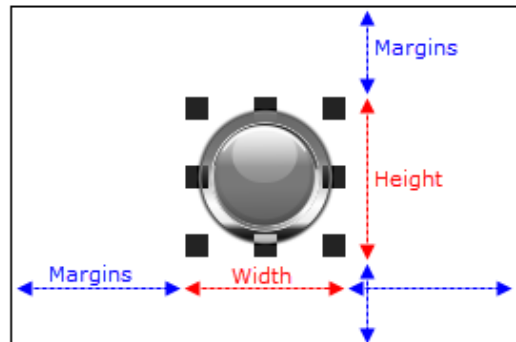
Row and columns properties are available inside a pop up dialog after clicking on the row and column selectors, that are visible after double clicking the group of widgets.



### Stretch



### Margins




### Geometry parameters


Parameter	Description
<b>Left margin</b> <b>Right margin</b>	Distance of the widget from the border of the cell
<b>Min width</b> <b>Max width</b>	Min/Max width that widget can assume when the cell is stretched
<b>Stretch</b>	Defines the relationship between the widths of the columns that will be maintained if the grid is stretched
<b>Top margin</b>	Distance of the widget from the border of the cell

Parameter	Description
<b>Bottom margin</b>	
<b>Min height</b> <b>Max height</b>	Min/Max height that widget can assume when the cell is stretched
<b>Stretch</b>	Defines the relationship between the heights of the rows that will be maintained if the grid is stretched

### Style parameters

Parameter	Description
<b>Left stroke width</b> <b>Right stroke width</b> <b>Top stroke width</b> <b>Bottom stroke width</b>	Strokes width
<b>Left stroke color</b> <b>Right stroke color</b> <b>Top stroke color</b> <b>Bottom stroke color</b>	Strokes color
<b>Background color</b>	Row background color

 The list of values that are separated by a comma, are related to rows and columns. Example, the first value is for row 0, the second value for row 1, and so on.

 Color format could be #rrggbb or #rrggbbaa, where "aa" is the alpha value which defines the opacity of the color.

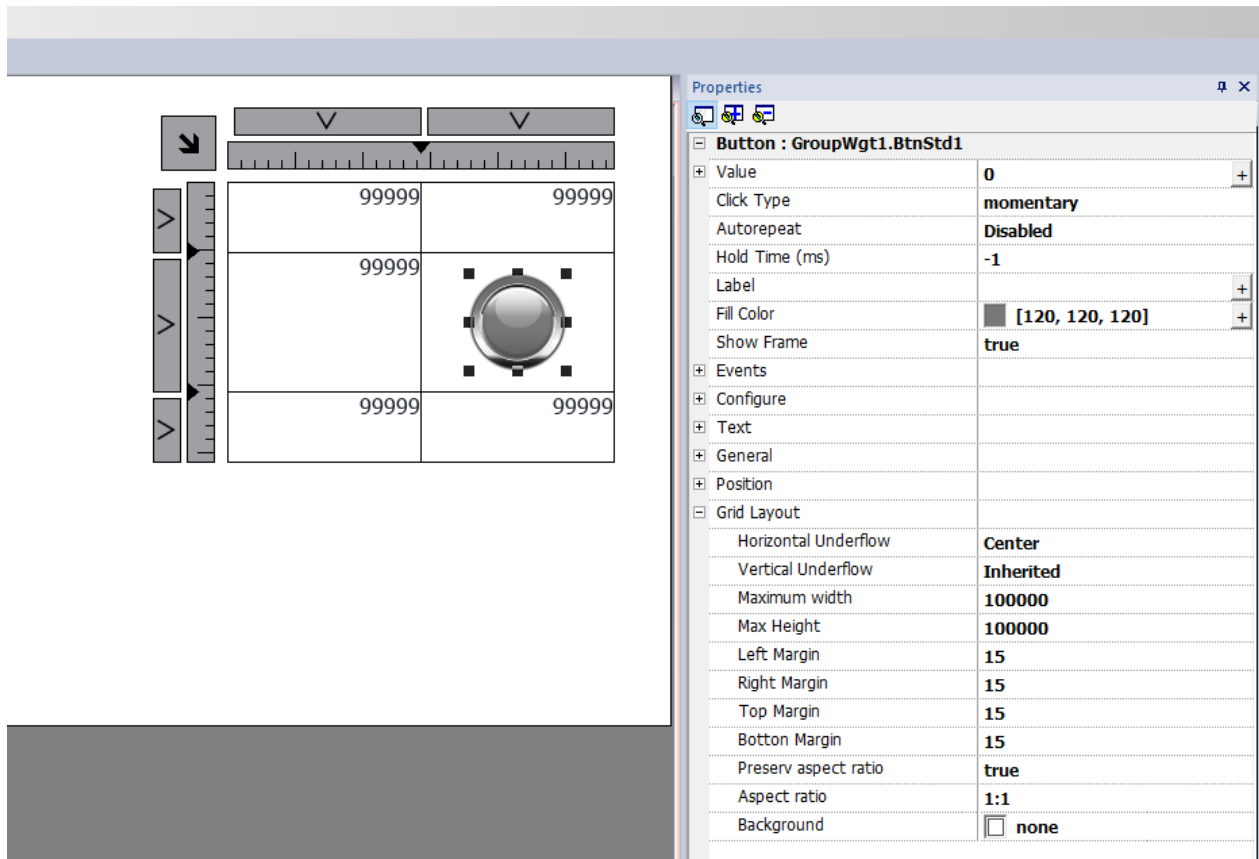
### Selection parameters

The selection parameters is available only when the grid is used inside a Table Widget (see "[Table widget](#)" on page 490 for details)

Parameter	Description
<b>Foreground color</b> <b>Background color</b> <b>Stroke color</b>	Colors that the row assume when it is selected  The list of colors is related with row templates. First color is for row template 0, second color is for row template 1, and so on.

### Cells Properties

Properties of a single cell are available inside the properties panel when a cell is selected. To select a cell: first double click the widget group, then click the cell to select.



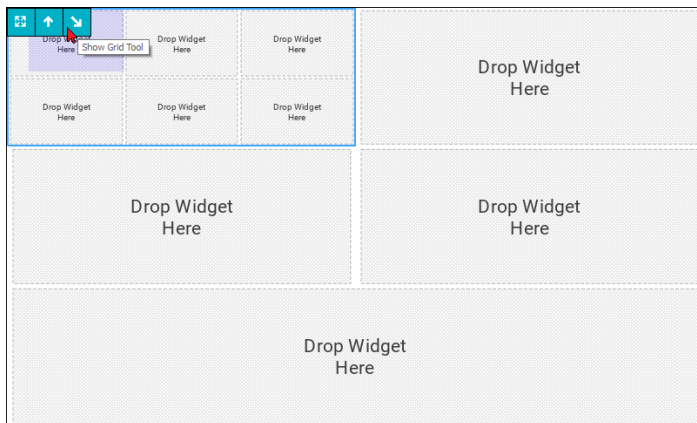
Parameter	Description
<b>Horizontal underflow</b> <b>Vertical underflow</b>	This parameter defines the behavior of the widget when the cell is larger than the size defined for widget. <ul style="list-style-type: none"> <li>• Inherited Inherits the value used for the row or column</li> <li>• Left, Center, Right - Top, Middle, Bottom Defines the position of the widgets when cells are bigger than the maximum defined sizes</li> </ul>
<b>Max width</b> <b>Max height</b>	Overwrite global grid parameters
<b>Left margin</b> <b>Right margin</b> <b>Top margin</b> <b>Bottom margin</b>	Overwrite global grid parameters Additional pixels that are added to the total margin.
<b>Preserve aspect ratio</b>	Preserve aspect ration of the widget
<b>Aspect ratio</b>	Available only when " <i>Preserve aspect ratio</i> " is true
<b>Background</b>	Background color of cell

## Add widgets inside the Grid Layout

To add a widget inside a cell of the Grid layout, double click the Grid Layout to enter in edit mode and drag and drop the widgets inside the cells.

### Dashboard

The dashboards that are associated with the pages are special Grid Layouts where you can drag and drop widgets inside the cells without the need for a double click to enable the edit mode. Use the toolbars to show the grid properties or to move between the nested grids.



## Printing report

Note the grid layout is available even inside the print report gallery.

# TextEditor widget

*Path: **Widget Gallery**> **Advanced**> **Editor***

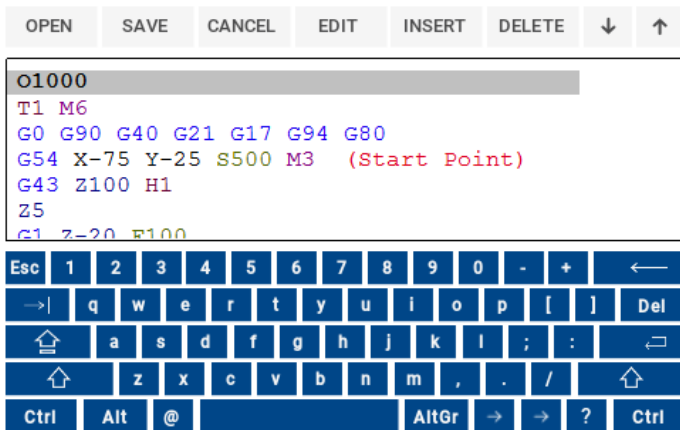
Use this widget to edit text files. Widget can load the text file from the local HMI device or download the file from a remote device using an ftp connection.



**Available only on Linux devices (See: ["HMI devices capabilities" on page 613](#))**



Note: TextEditor widget is available as a plugin (see ["Plug-in" on page 86](#) for details) to download to the HMI Runtime only when required.



## Widget Buttons

Button	Description
Open	Load text file inside the TextEditor
Save	Save text file
Cancel	Remove all changes from last OPEN or SAVE command
Edit	Enter in edit mode
Insert	Insert a new line
Delete	Delete current line
Up/Down	Move cursor up/down

## Widget Properties

Option	Description						
Keyboard	<p>TextEditor widget has an embedded keyboard. When widget is used without the embedded keyboard, the alphabetic keyboard will be displayed when enter edit mode.</p> <ul style="list-style-type: none"> <li>• Hidden</li> <li>• Visible</li> </ul>						
FTP Config	<p>FTP parameters to download the text file from a remote FTP server. Leave this field empty to load the text file from the local HMI device.</p> <table border="1"> <thead> <tr> <th>Parameter</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>FTP Address</td> <td>FTP server IP Address</td> </tr> <tr> <td>Server Port</td> <td>Port for FTP connection (default = 21).</td> </tr> </tbody> </table>	Parameter	Description	FTP Address	FTP server IP Address	Server Port	Port for FTP connection (default = 21).
Parameter	Description						
FTP Address	FTP server IP Address						
Server Port	Port for FTP connection (default = 21).						

Option	Description	
	<b>Parameter</b>	<b>Description</b>
	<b>Authentication</b>	Select the FTP authentication to use: <ul style="list-style-type: none"> <li>• Normal (Username and password required)</li> <li>• Anonymous</li> </ul>
	<b>User Name</b>	Username of the remote FTP account
	<b>Password</b>	Password of the remote FTP account
<b>File Name</b>	File name to edit. When empty a file browser to load a local file will be opened	
<b>Syntax Highlight Type</b>	Displays text in different colors according to the selected language <ul style="list-style-type: none"> <li>• None</li> <li>• GCode</li> </ul>	

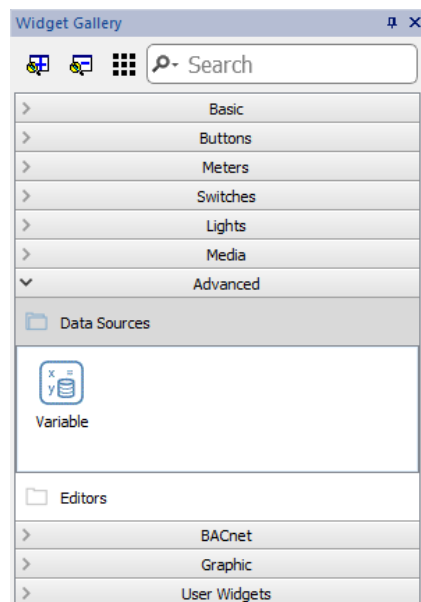
## Variables widget

*Path: Widget Gallery > Advanced > Data Sources*

Use this widget to add internal variables for operations such as data transfer or to be used in JavaScript programs.



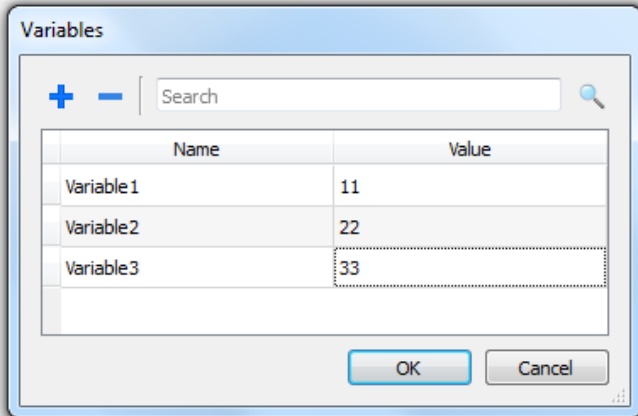
Note: The variables are local to the page where the widget has been inserted.



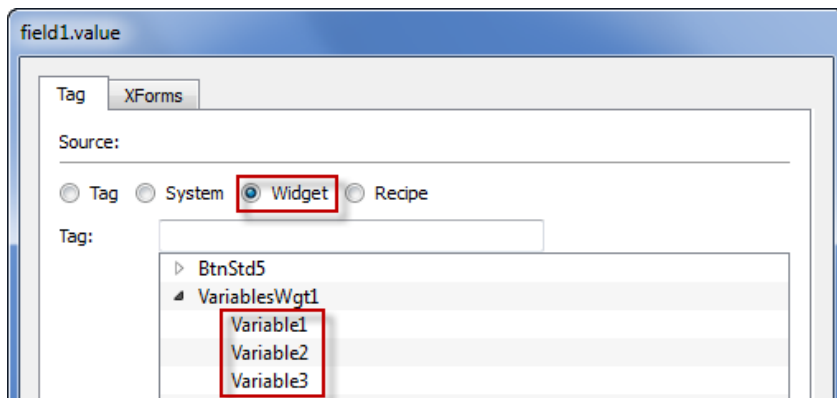
When you drag and drop this widget into your page, a placeholder will be displayed to indicate the widget location, but it will not be visible at runtime.

## Setting the widget

To create variables and assign values to them, open the **Variables** dialog from the **Variables** property in the **Properties** pane.

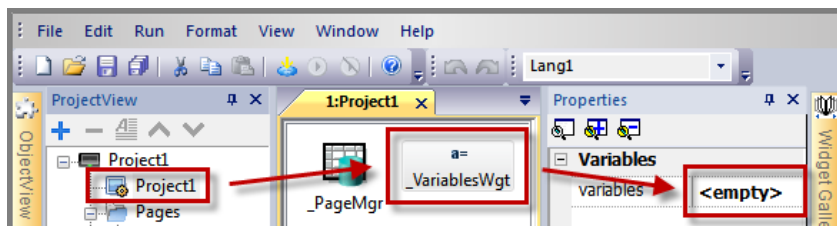


These variables can then be referenced from the **Attach tag** dialog, from the Page Editor.



## Global Variable Widget

If you need global variables, configure them at project level, adding the desired variables to the global variable widget.



## Using variables in JavaScript

Variables can be also referenced in JavaScript programs with the following syntax:

For local variables:

```
var varWgt = page.getWidget("_VariablesWgt");
```



```
var compVar = varWgt.getProperty("VariableName");
```

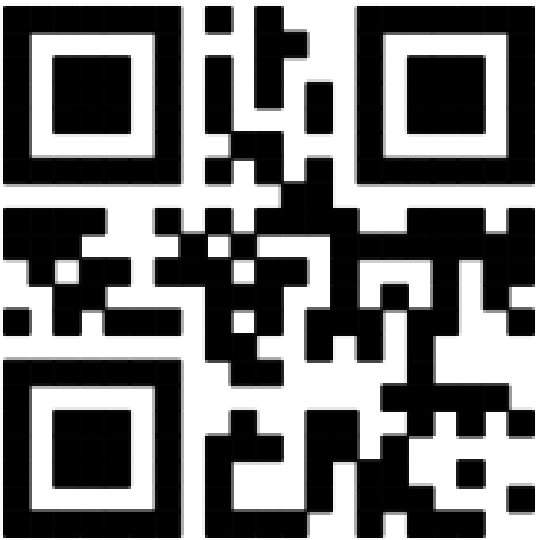
For global variables:

```
var varWgt = project.getWidget("_VariablesWgt");
var compVar = varWgt.getProperty("VariableName");
```

## QR Code widget

**Path:** *Widget Gallery > Media > Web Controls*

This widget will generate a QR code image that will be useful to easily read the contained information from a mobile device.



Available only on Linux devices (See: ["HMI devices capabilities" on page 613](#))

Parameter	Description
<b>Page Name</b>	<p>The name of the page of the HMI device to open. For example, if the value is "Phone" the generated URL will be "<a href="http://192.168.1.100/index.html?loadPage=Phone">http://192.168.1.100/index.html?loadPage=Phone</a>" to open the "Phone" page on the HMI device.</p> <p>This field must be left blank if you want to define a generic URL.</p>
<b>IP Address</b>	The IP Address of the HMI device or a generic URL (without the http/https prefix)
<b>Ethernet adapter</b>	<p>Ethernet adapter can be used as an alternative to the IP Address parameter. When it is used, the IP Address parameter will be ignored and the HMI runtime will fill the URL with the IP Address of the selected adapter.</p> <p>The adapter names are: "eth0", "eth1", etc.</p>

---

Parameter	Description
<b>Prefix</b>	Can be http or https
<b>Ethernet Port</b>	The port number to use (default is 80)
<b>Cell Color</b>	QR Code color
<b>Background Color</b>	QR Code background color



Cell Color and Background Color are available in the advanced view.



# 39 Custom widgets

---

PB610 Panel Builder 600 has a large widget library which includes predefined dynamic widgets (buttons, lights, gauges, switches, trends, recipes, and dialog items), as well as static images (shapes, pipes, tanks, motors).

You can drag and drop an object from the gallery to the page, and then size, move, rotate or transform it. All widgets in the gallery are vector based, so they do not lose definition when resized.

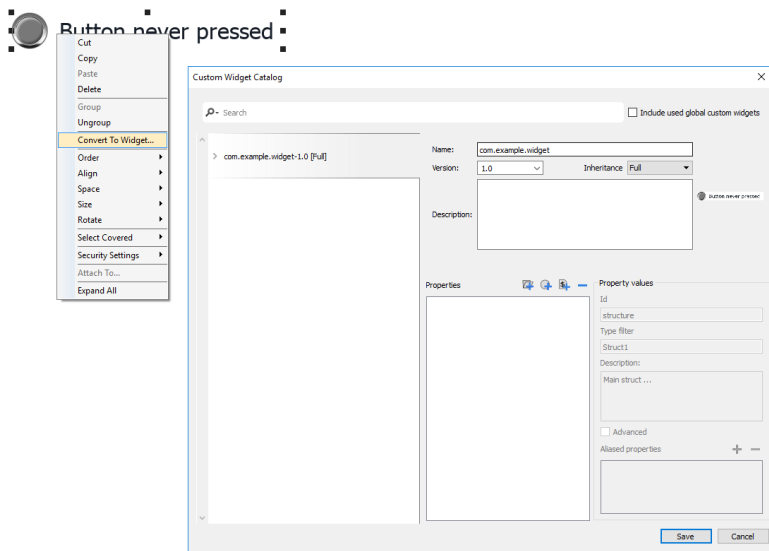
You can, however, modify any of the pre-defined widgets to create your own custom widget. Custom widgets can be made up of several elements only including the properties needed to their purpose.

---

<b>Creating a custom widget</b> .....	<b>522</b>
<b>Adding properties to a custom widget</b> .....	<b>524</b>
<b>Using structured tags</b> .....	<b>527</b>
<b>JavaScript in custom widgets</b> .....	<b>529</b>
<b>User's Gallery</b> .....	<b>532</b>

# Creating a custom widget

1. Drag and drop on a page all the widget you want to use to compose your custom widget.
2. Select and group them.
3. Right-click on the grouped object and select **Convert To Widget**: the **Custom Widgets Catalog** dialog is displayed.



Parameter	Description
<b>Include used custom widgets</b>	When checked, list all the widgets used inside the project. Even system widgets.
<b>Name</b>	You can define everything you prefer, but is common keep a name structure. The folder com.hmi is reserved for the system widgets
<b>Description</b>	Widget description.
<b>Version</b>	Widget version. All widgets that share the same version share the properties defined from the Inheritance parameter.
<b>Inheritance</b>	Properties shared between widgets with the same version <ul style="list-style-type: none"> <li>• Full (both Graphic and Logic)</li> <li>• Only Graphic</li> <li>• Only Logic</li> <li>• Disable</li> </ul>

## Modify a custom widget

To modify a custom widget, simple double clicks the custom widget to enter in edit mode.

If the Inheritance flag is enabled, a lock icon will appear to warn you that you are add changes that will be propagated to all the other custom widgets that share the same version. Click the padlock icon to confirm to enter in edit mode, padlock will be open. Click again when modifies are done.

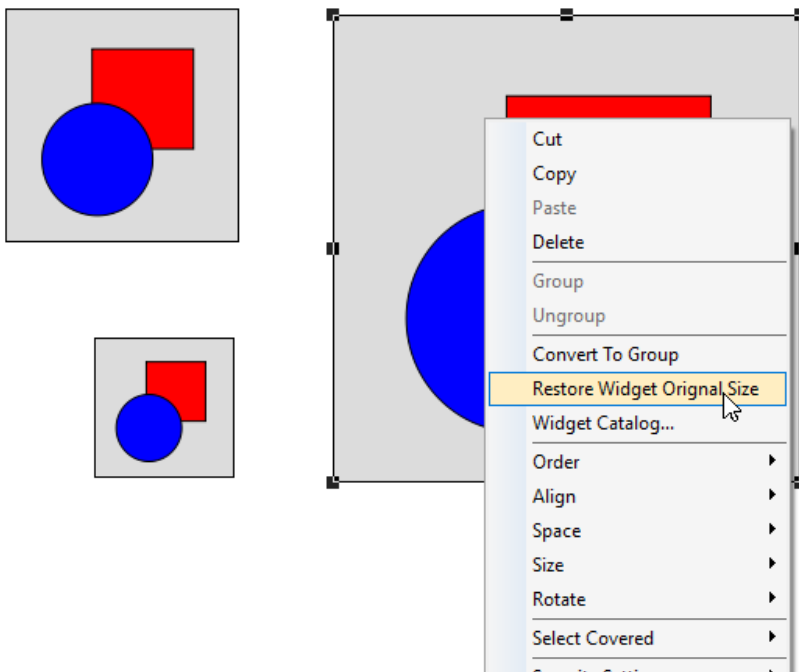


Padlock is showed only when the Inheritance is enabled.

## Resize a custom widget

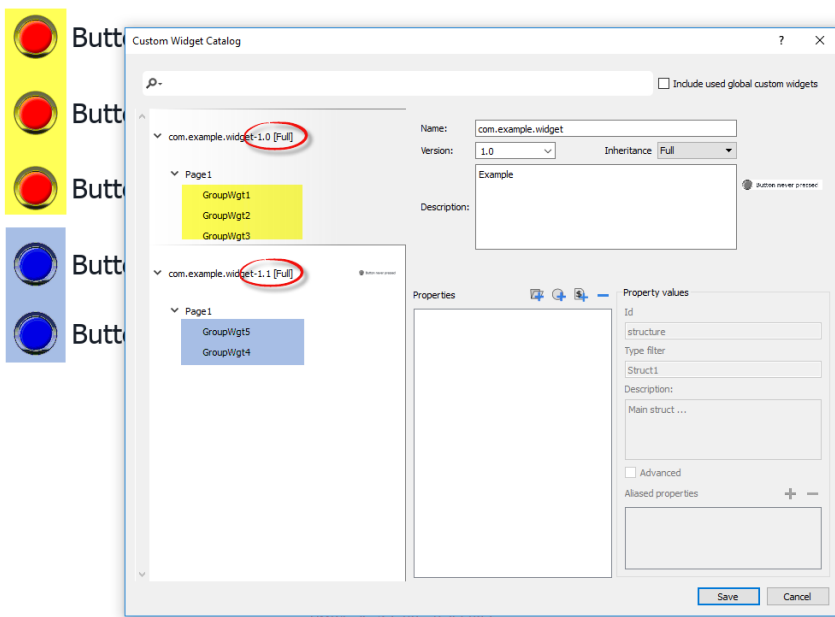
When sizes of custom widget is changed, the new sizes will not be propagated to the other widget instances.

“Restore Widget Original Size” command can be used.



## Share properties

When a custom widget is modified, all the modifies will be propagated to all the other custom widgets that share the same version and that are configured to inherit the widget properties.

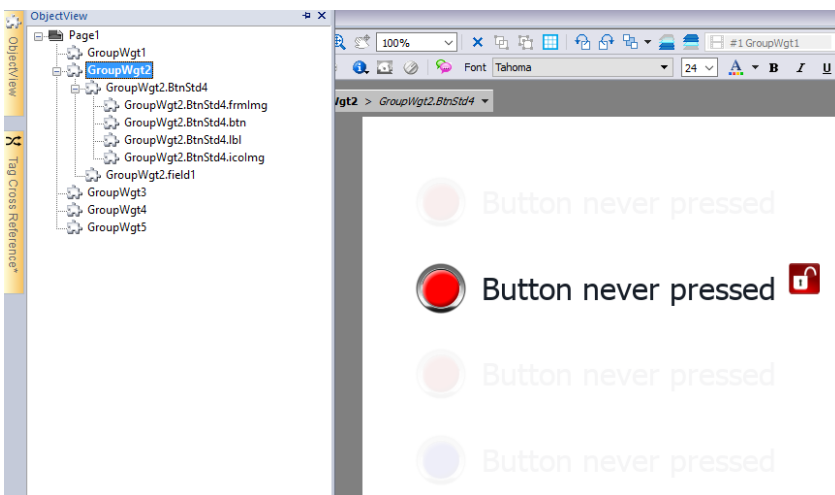


## Using widgets components

Widgets are usually made up of many parts, for example a button is a complex widget including two image widgets, a button widget and label.

To display a list of all the elements that are part of a widget, select the widget, open the padlock and open the **ObjectView** pane: all the element making up a complex widget are listed in hierarchical order.

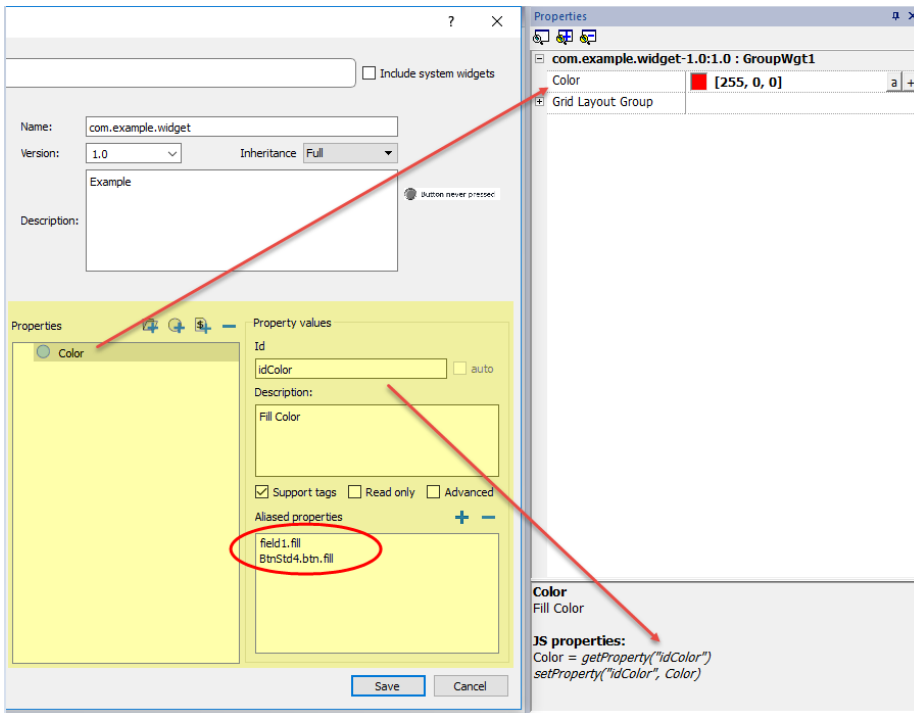
To select a single widget, select it directly form the **ObjectView** pane.



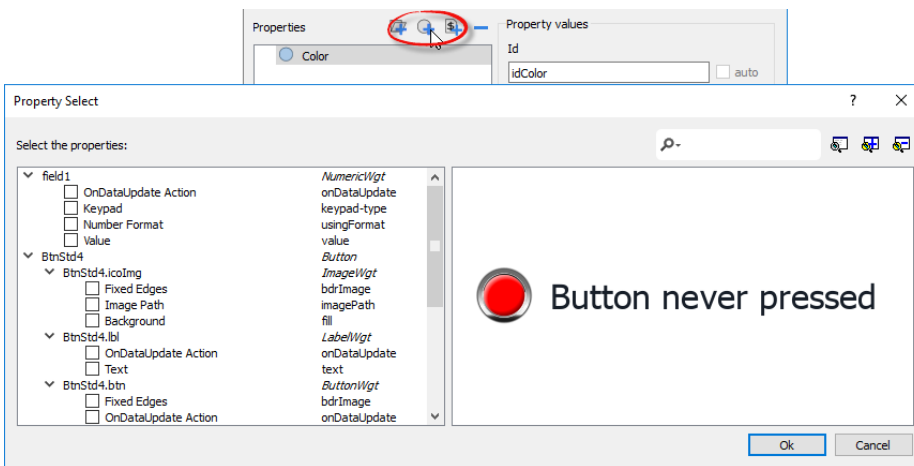
## Adding properties to a custom widget

When you create a custom widget, you need to define the properties that will be displayed for it in the **Properties** pane.

1. Right-click on the grouped object and select **Widget catalog**: the properties dialog is displayed.



2. Click **+** to open the **Property Select** dialog: this lists all the properties of all the grouped widgets.



3. Select the properties you want to define for your custom widget.
4. Define each property's details.



Note that you can create folders and use drag & drop to move or reorganize the **Properties** list

Parameter	Description
<b>Properties</b>	Name shown in the <b>Properties</b> pane.
<b>Description</b>	Any comment on the property to be displayed in the <b>Properties</b> pane.



Parameter	Description
<b>Id</b>	The name exposed by PB610 Panel Builder 600, to JavaScript functions and Attach Tag dialog.
<b>Support Tags</b>	Specifies if the property supports the "Attach to" attribute.
<b>Read only</b>	Property exposed only in read mode
<b>Advanced</b>	Specifies whether each property should appear in the advanced, or in the simple view mode of the <b>Properties</b> pane.
<b>Aliased properties</b>	Internal properties linked with the exposed property

## Combining properties

To combine two or more properties:

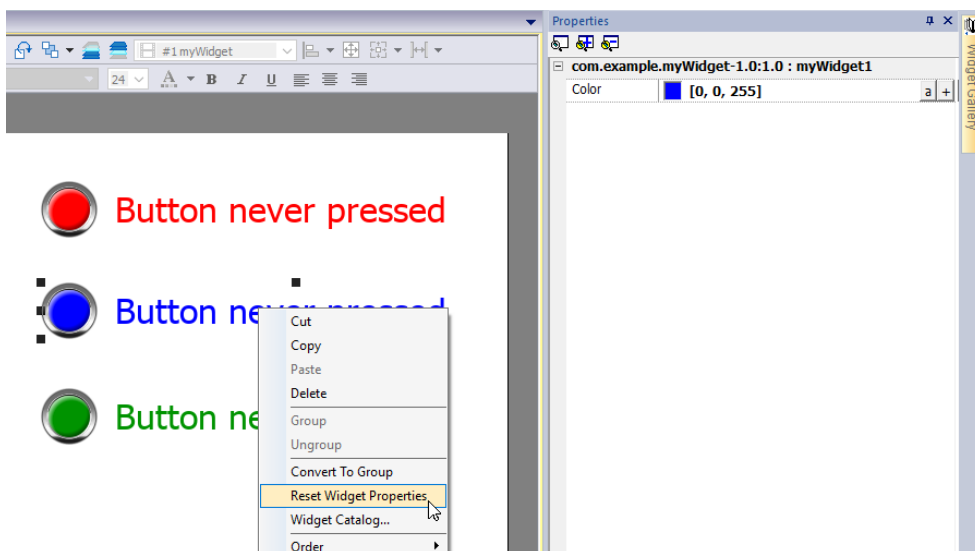
1. Select the primary property in the **Properties** list dialog.
2. Click **+** in the **Aliased properties** toolbar: the **Property Select** dialog is displayed.
3. Select the properties you want to combine.
4. Click **OK**: the combined attributes will be shown in the **Aliased properties** list box.

### Example

If you insert into a "Color" property the fill color of all widgets (e.g. filed1.fill and BtnStd4.btn.fill) when you set the exposed Color property of the custom widget all colors of the included widgets will changes.

## Reset Widget Properties

The "Reset Widget Properties" reset the modified properties values to original values.



# Using structured tags

A common problem using a widget that use many tags is the need to create instances of the widget by giving only the tag name of the structure that contains all the tags instead to configure each single tag.

For example, think about the below widget. It use four tags, the room name, temperature, humidity and pressure. If we want use two instances of this widget for two different rooms we have to configure eight tags, four tags for each room.

**Bathroom**

Temperature: 23.0  
Humidity: 52  
Pressure: 105

**Living room**

Temperature: 21.0  
Humidity: 22  
Pressure: 101

Properties window for `com.example.widget-1.0:1.0 : GroupWgt1`. It shows a table with columns for property name and value. The 'DataLink' property is highlighted in yellow for each of the four tags: `room1/name`, `room1/temperature`, `room1/humidity`, and `room1/pressure`.

Select datatype for GroupWgt1.RoomID dialog. Source: Tag (selected) / Widget. It shows a tree view of data sources. Under 'CODESYS V3 ETH:prot1 Model: CODESYS 3', there is an 'Application' container, which contains three 'Room' containers (Room1, Room2, Room3). Each room container lists its properties: Humidity (BYTE), Name (STRING), and Pressure (BYTE). The tag names are listed as `Application/Room1/Humidity`, `Application/Room1/Name`, `Application/Room1/Pressure`, etc.

By using a **Parameter** property, is possible to set all the data links of the widget by giving only the name of the structure.

**Bathroom**

Temperature: 23.0  
Humidity: 52  
Pressure: 105

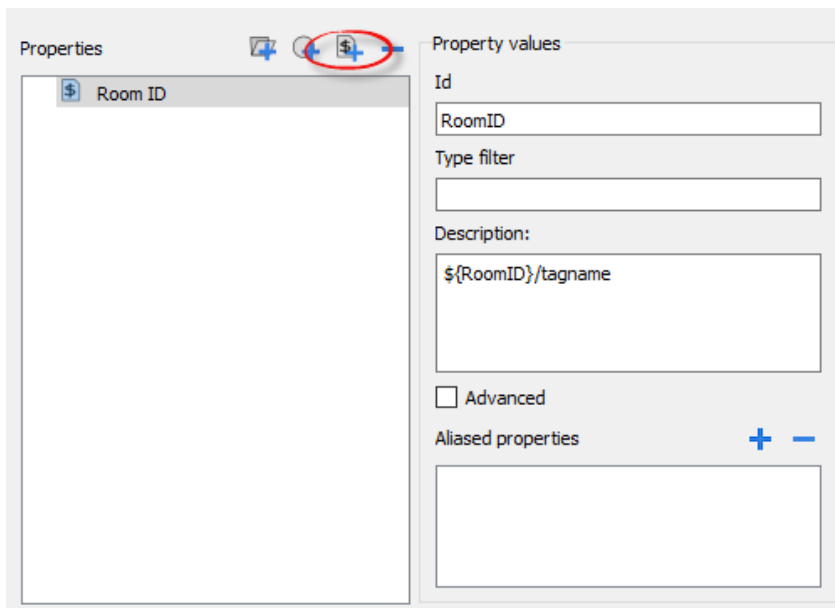
Properties window for `com.example.widget-1.0:1.0 : GroupWgt1`. It shows a 'Room ID' property highlighted in yellow with the value `room1`.

**Living room**


Temperature: 21.0  
Humidity: 22  
Pressure: 101

Properties window for `com.example.widget-1.0:1.0 : GroupWgt1`. It shows a 'Room ID' property highlighted in yellow with the value `room2`.

A "Parameter" field can be added inside the custom widget using the "Add Parameter" icon:



To configure the data links of the custom widget the keyword `${RoomID}` can be used to reference at the structure instance

**Room** 

Temperature: ] 0.0

Humidity: ] 0

Pressure: ] 0

Room

Temperature: ] 0.0

Humidity: ] 0

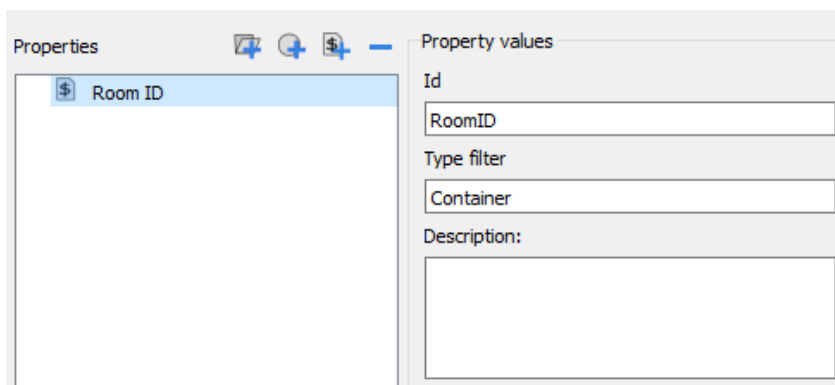
Pressure: ] 0

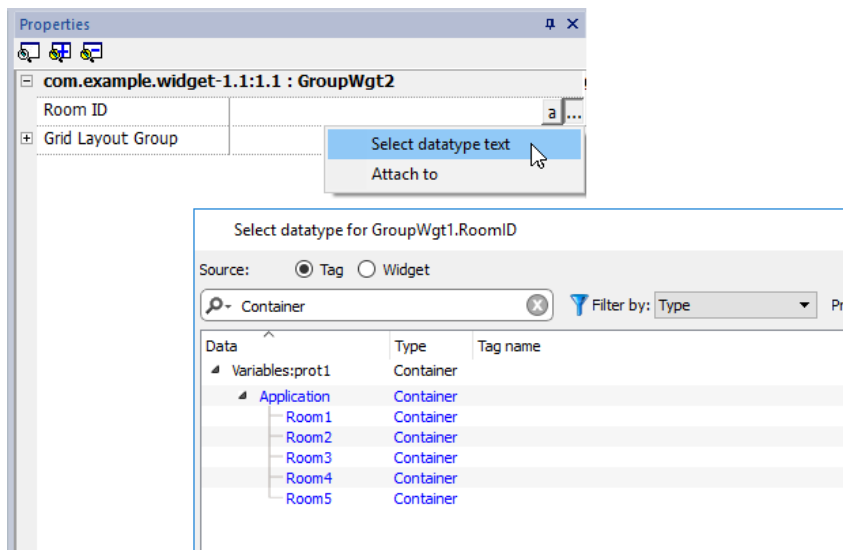
**Properties**

Field : GroupWgt1.field2	
Value	0.0
DataLink	<code>\${RoomID}/temperature</code>
Access Type	R
Number Format	##
Keypad	Numeric
Events	

### Type filter

Typically, value of the parameter will be an element of a structured tag. Using the “*Type filter*” parameter, the “*Select datatype text*” will list filtered tags.





The “*Select datatype text*” will return a string while the “*Attach to*” will return a datalink to a tag that will contains the string to use.

## getParameter

From JavaScript you can read the parameters' value using the `getParameter()`

```
object getParameter(paramID)
```

Example:

```
var myWidget = page.getWidget("myWidget");
function BtnStd3_btn_onMouseClicked(me, eventInfo)
{
    alert("Room is: " + myWidget.getParameter("RoomID"));
}
```

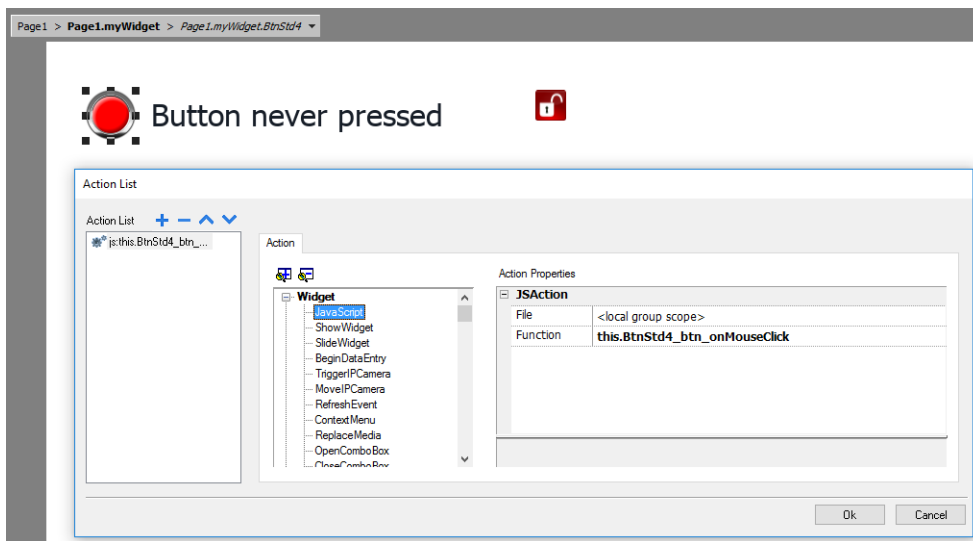


You can also use `getProperty()`, but `getParameter()` is more efficient to read custom widget parameters

## JavaScript in custom widgets

JavaScript functions can be embedded in custom widgets.


After doing a double click on the custom widget and clicked on the padlock, the edit mode is active and it is possible to associate the JavaScript code to the available events.



If you need to reference to an element of the widget, you can use the keyword **wgt..** For example, use wgt.id to reference at the id of the active widget instance.





If you cut and paste some instances of the custom widget of the above example and execute it, e.g. inside the simulator, you will obtain the below result.

 Tue Jan 31 2017 14:51:18

 Button never pressed

 Tue Jan 31 2017 14:51:12

 Tue Jan 31 2017 14:51:14

 Button never pressed

## onActivate property

To initialize the custom widget is possible to define the onActive property with an initializing function as for the below example.

The onActivate() function will be execute when the page is loading

```

1  v  /*!
2      javascript module: widget-1.0.js
3      javascript source file path: lib\com\example\widget-1.0\widget-1.0.js
4      */
5
6      this.wMSG = wgt.getWidget(wgt.id+".field1")
7
8      this.BtnStd1_btn_onMouseClicked = function (me, eventInfo)
9      {
10         var now = new Date();
11         this.wMSG.setProperty("value", now.toString().slice(0, 24));
12     }
13
14     this.onActivate = function()
15     {
16         this.wMSG.setProperty("value", "Button never pressed");
17     };
18     this.onActivate();
19

```



Note that the custom widget can also past inside the User's Gallery for later reuse.

### The JavaScript code used inside the examples of this chapter

```

/*!
javascript module: widget-1.0.js
javascript source file path: lib\com\example\widget-1.0\widget-1.0.js
*/

this.wMSG = wgt.getWidget(wgt.id+".field1")

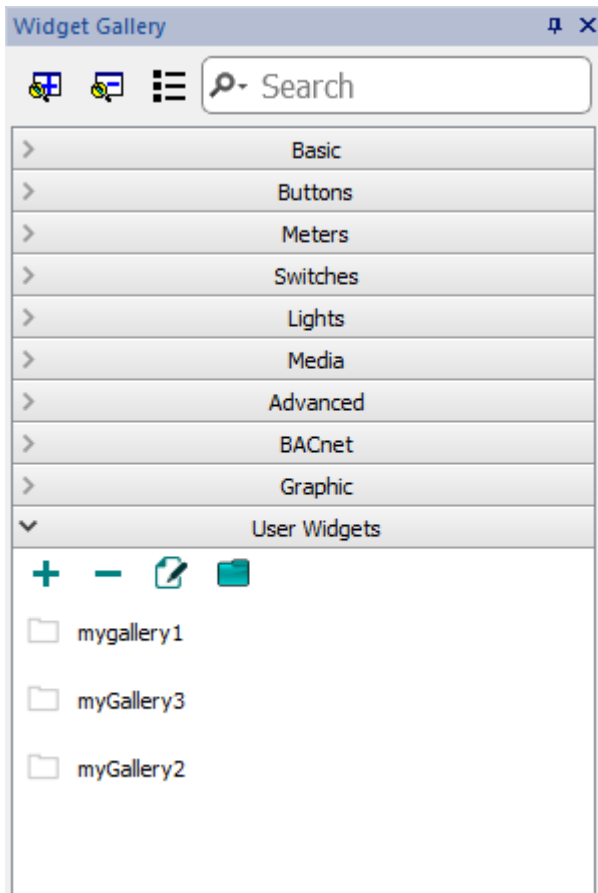
this.BtnStd1_btn_onMouseClicked = function (me, eventInfo)
{
    var now = new Date();
    this.wMSG.setProperty("value", now.toString().slice(0, 24));
}

this.onActivate = function()
{
    this.wMSG.setProperty("value", "Button never pressed");
};
this.onActivate();





```

## User's Gallery

Widgets created from the developers can be saved inside the Widgets Gallery to be available during development of new projects.



## User widgets toolbar

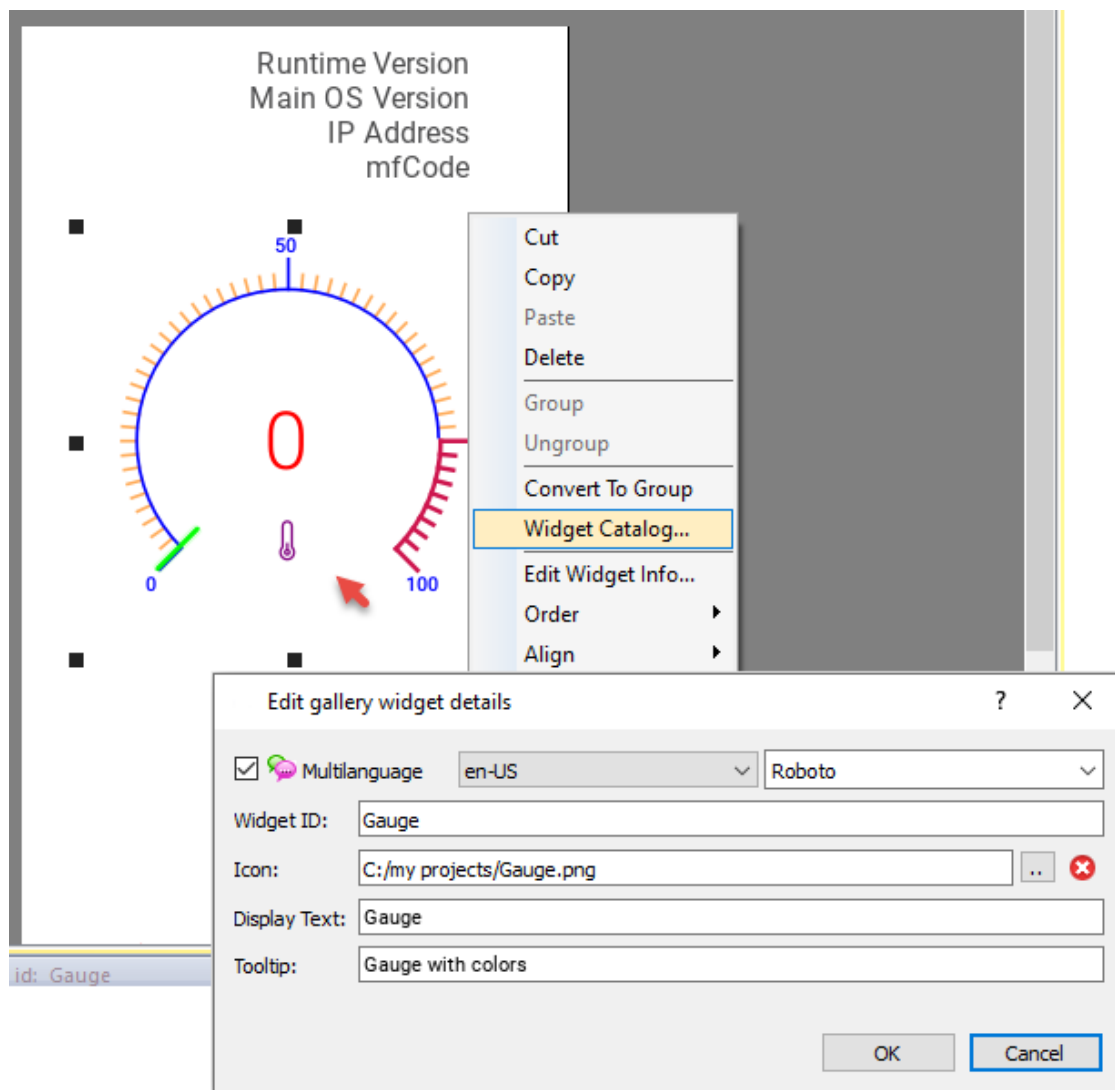
Command	Description
	Add a new widgets folder
	Delete current selected folder
	Open the selected widgets folder into the PB610 Panel Builder 600 editor
	Select the user widgets folder

## Adding a new widget

To add a new widget into the user gallery, open the widget folder and then edit the gallery page creating or adding the new widget.

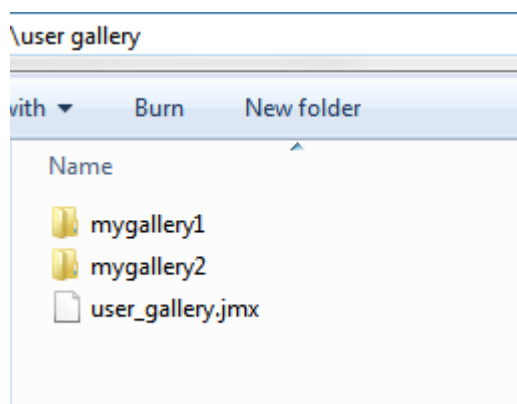
To define the icon, the tooltip, and the widget's description, select the widget, right-click on the widget and select "Edit Widget Info..." in the context menu. The "Edit gallery widget details" dialog will be open.





## Import a user gallery sub folder

To import a user gallery sub folder, simply copy the folder to import inside the main user gallery folder.

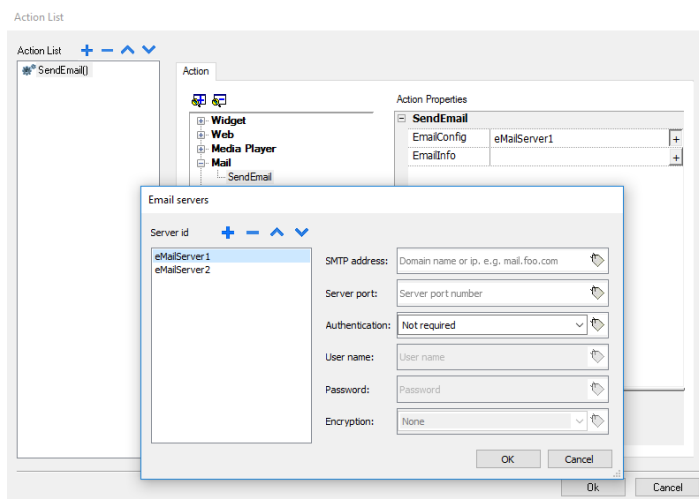


# 40 Sending an email message

---

Send emails using the SendMail action, including tags in the email body and attachments.

The SendMail action has been created for working with alarms and schedulers but can be triggered and executed by many other events.





---

<b>Configuring the email server</b> .....	<b>536</b>
<b>Configure emails</b> .....	<b>536</b>

# Configuring the email server

To configure the email server, enter the following information for the **EmailConfig** setting:

Parameter	Description
<b>SMTP Address</b>	SMTP server address.
<b>Server Port</b>	Port for SMTP server connection.
<b>Require Auth</b>	Select if the SMTP server requires authentication.
<b>User Name</b>	Username for sending mail using SMTP server.
<b>Password</b>	Password for sending mails using SMTP server.
<b>Encryption</b>	<p>Encryption type (none, SSL or TLS).</p> <p> TLS 1.2 and TLS 1.3 are supported. ("none" and "SSL" are obsolete encryption modes, kept for compatibility only)</p> <p> <b>Available only on Linux devices (See: "HMI devices capabilities" on page 613)</b></p>

Click + to add more email servers.




Tip: Use tags if you want change the server parameters dynamically from the HMI Runtime.

## Configure emails

Enter the following information for the **EmailInfo** setting:

Parameter	Description
<b>Name</b>	Optional, this information is only for the log.
<b>Description</b>	Optional, this information is only for the log.
<b>From</b>	Sender email address (for example, John@domain.com).
<b>To</b>	Recipient e-mail addresses. To enter multiple addresses, separate them with a semi-colon.
<b>Subject</b>	Subject of email.

Parameter	Description
<b>Attachment</b>	<p>Path of the file to be sent as attachment. Only one attachment at a time can be sent.</p> <p> Note: The maximum size of the attachments is usually set by the SMTP server.</p>
<b>Body</b>	<p>Main content of the email. Here you can insert live tags if you include them in square brackets.</p> <p>For example, a message body as “Tag1 value is [Tag1]”, will be sent as “Tag1 value is 45”, if the current value of Tag1 is 45.</p>



Tip: Attach a string tag to the **From**, **To** and **Subject** fields so that their value can be changed in the HMI Runtime.

**WARNING**

- HMI limit in the email *From* and *To* fields is 32767 chars.
- The maximum size for the message *Body* is 4096 bytes, the exceeding text will be truncated.
- Check the email server (e.g. Gmail, Outlook) about its limits regarding no of chars, no of recipients, total mail size, attachment size, no of emails that can be sent in a day, etc.

**Adding email templates**

Click **+** to add more templates.

The screenshot shows a software interface for configuring email templates. At the top, there's a title bar 'Emails'. Below it, a 'Drafts' section contains a list with one item 'eMail1'. To the right of the list are several input fields: 'Name' (with 'Name' as a placeholder), 'Description' (with 'Description' as a placeholder), 'From' (with 'Edit value' and a tag icon), 'To' (with 'Edit value' and a tag icon), 'Subject' (with 'Edit value' and a tag icon), and 'Attachment' (a dropdown menu). Below these fields is a large 'Message' text area with a tag icon on its right side. At the bottom of the dialog are 'OK' and 'Cancel' buttons.



# 41 JavaScript

---

The purpose of this section is to describe how JavaScript is used in the PB610 Panel Builder 600 applications, not to explain the JavaScript language.

PB610 Panel Builder 600 JavaScript is based on the ECMAScript programming language <http://www.ecmascript.org>, as defined in standard ECMA-262.

If you are familiar with JavaScript, you can use the same type of commands in PB610 Panel Builder 600 as you do in a web browser. If you are not familiar with the ECMAScript language, refer to:

<https://developer.mozilla.org/en/JavaScript>

---

<b>JavaScript editor</b> .....	<b>541</b>
<b>Execution of JavaScript functions</b> .....	<b>541</b>
<b>Events</b> .....	<b>543</b>
<b>Widget events</b> .....	<b>544</b>
<b>Page events</b> .....	<b>546</b>
<b>System events</b> .....	<b>547</b>
<b>Objects</b> .....	<b>549</b>
<b>Widget class objects</b> .....	<b>549</b>
<b>Widget properties</b> .....	<b>550</b>
<b>Widget methods</b> .....	<b>553</b>
<b>Page object</b> .....	<b>554</b>
<b>Page object properties</b> .....	<b>554</b>
<b>Page object methods</b> .....	<b>555</b>
<b>Project object</b> .....	<b>557</b>
<b>Project object properties</b> .....	<b>557</b>
<b>Project object methods</b> .....	<b>558</b>
<b>Project object widgets</b> .....	<b>567</b>
<b>Print reports object</b> .....	<b>568</b>
<b>Group object</b> .....	<b>571</b>
<b>Group object methods</b> .....	<b>571</b>
<b>State object</b> .....	<b>572</b>
<b>State object methods</b> .....	<b>572</b>
<b>Keywords</b> .....	<b>573</b>

---

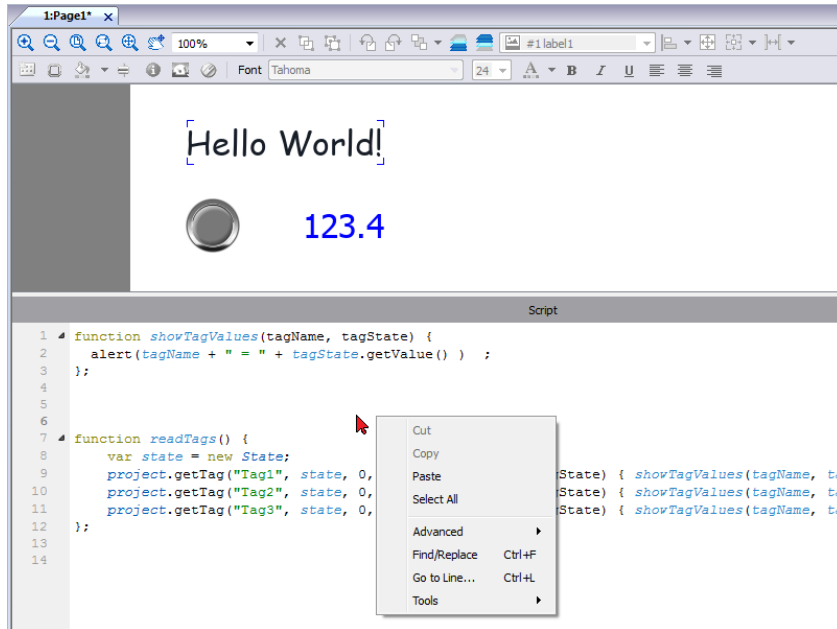
---

<b>Global functions</b> .....	<b>574</b>
<b>Handling read/write files</b> .....	<b>575</b>
<b>Sign in from JavaScript</b> .....	<b>578</b>
<b>Limitations in working with widgets in JavaScript</b> .....	<b>580</b>
<b>Debugging of JavaScript</b> .....	<b>580</b>

## JavaScript editor

PB610 Panel Builder 600 includes a powerful JavaScript editor.

Right-click in the editor to display available commands.



## Execution of JavaScript functions

JavaScript functions are executed when events occur. For example, a user can define a script for the OnMouseClicked event and the JavaScript script will be executed when the button is pressed on the HMI device.

JavaScript functions are executed only when the programmed event occurs and not cyclically. This approach minimizes the overhead required to execute logic in the HMI device.

PB610 Panel Builder 600 provides a JavaScript engine running on the client side. Each project page can contain scripts having a scope local to the page where they are added; global scripts can be created to be executed by scheduler events or alarm events.

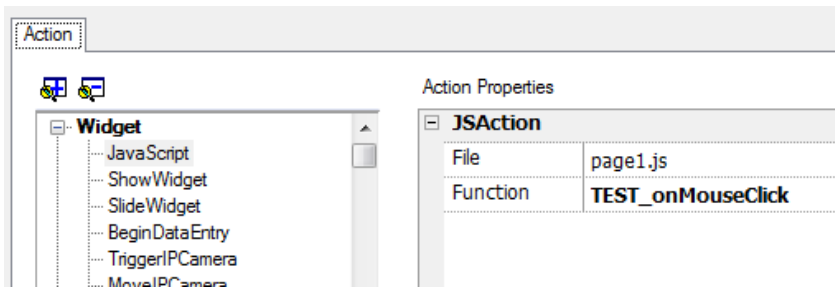
In both cases scripts are executed on the client. This means that if more than one client is connected to the HMI device (for external computer running the HMI Client), each client will run the same script, providing different output results depending on the input, since inputs provided to different clients may be different.

For example, if a script acts according to the position of a slider and this position is different on the different clients, the result of the script will be different on each client.

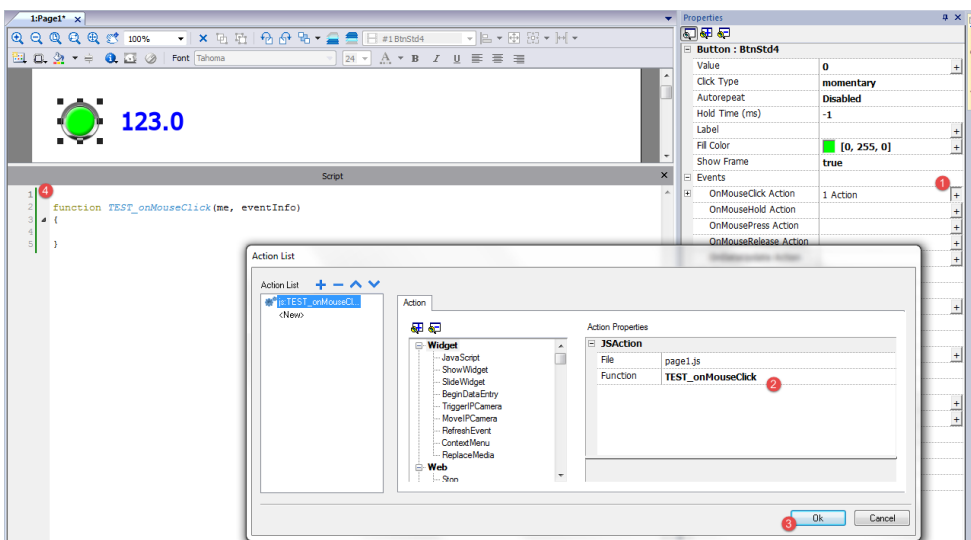
### JavaScript functions for page events

JavaScript editor will open when you add a JavaScript action inside an action list.

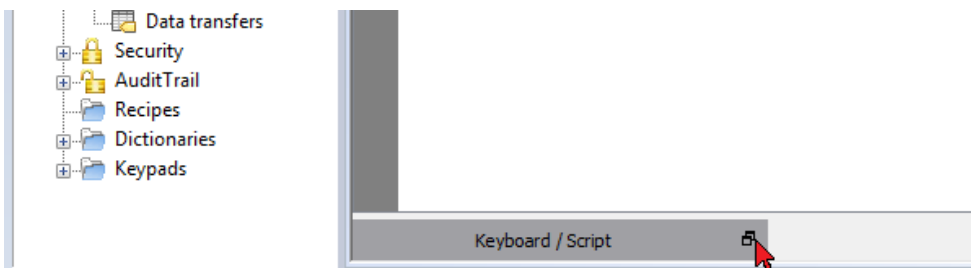




1. Select the even that will execute the action.
2. Add a **JavaScript** action from the **Widget** category.
3. Either leave the default function name, or type a new one.
4. Click **OK** to confirm: the JavaScript editor displays your function structure.



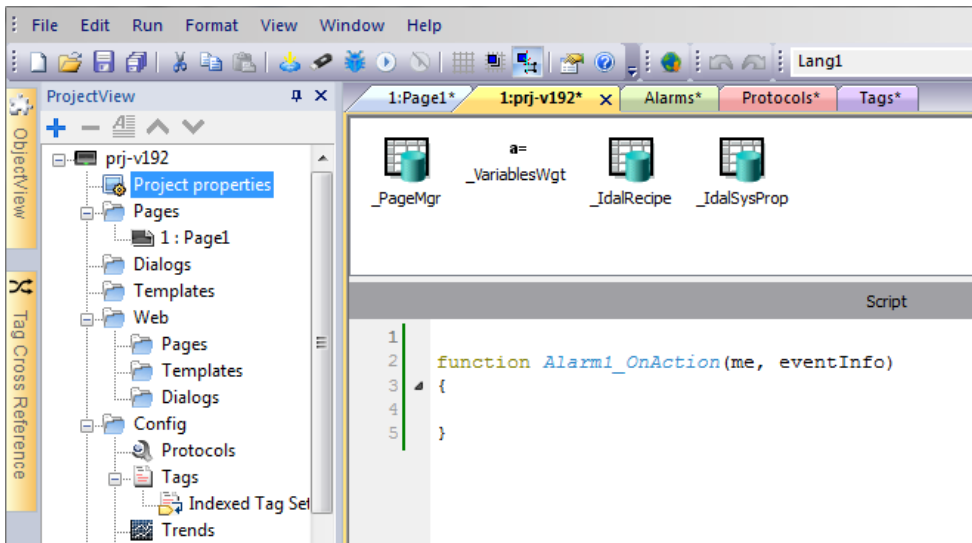
You can also open the JavaScript editor from the **Script** tab at the bottom of the workspace.



## JavaScript functions for alarms and scheduled events

JavaScript code associated with alarms and scheduled events and not associated with a specific page, can be edited from the main **Project properties** page.

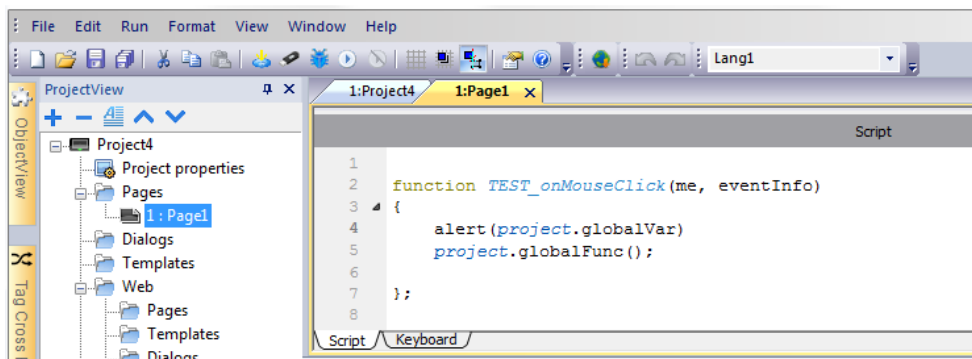
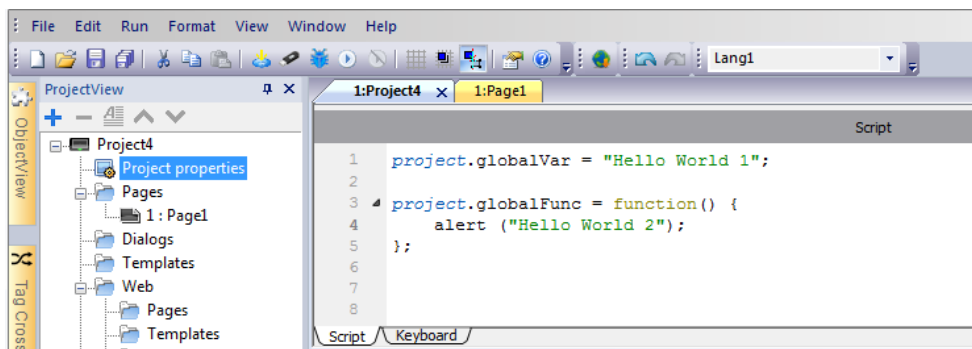
*Path: ProjectView > double-click Project properties*



Note: JavaScript actions are client actions so they are executed only when a client is logged in.

## Shared JavaScript code

The **project** global variable can be used to share JavaScript code between the pages. Variables are created/initialized from the main JavaScript code from the main **Project properties** page and can then be used from the project pages.



## Events

You can add JavaScript to the following categories of events:

- Widget events
- Page events
- System events

For events of type:

- OnMousePress
- OnMouseRelease
- OnMouseClicked
- OnWheel

JavaScript **eventInfo** parameter contains the following additional properties:

Parameter	Description
<b>eventInfo.posX</b>	Local mouse/touch X coordinate with respect to widget coordinates
<b>eventInfo.posY</b>	Local mouse/touch Y coordinate with respect to widget coordinates
<b>eventInfo.pagePosX</b>	Page X mouse/touch coordinate
<b>eventInfo.pagePosY</b>	Page Y mouse/touch coordinate
<b>eventInfo.wheelDelta</b>	Mouse wheel delta. Integer value with sign representing the rotation direction.  The actual value is the rotation amount in eighths of a degree. The smallest value depends on the mouse resolution. Typically this is 120, corresponding to 15 degrees.

## Widget events

### onMouseClicked

```
void onMouseClick (me, eventInfo)
```

This event is available only for buttons and it occurs when the button is pressed and released quickly.

Parameter	Description
<b>me</b>	Object triggering the event
<b>eventInfo</b>	Details of triggered event

```
function buttonStd1_onMouseClicked(me, eventInfo) {
    //do something...
}
```

### onMouseHold

```
void onMouseHold (me, eventInfo)
```

This event is available only for buttons and it occurs when the button is pressed and released after the number of seconds set as **Hold Time** in the widget properties.

Parameter	Description
<b>me</b>	Object triggering the event
<b>eventInfo</b>	Details of triggered event

```
function buttonStd1_onMouseHold(me, eventInfo) {
    //do something...
}
```

## onMousePress

```
void onMousePress (me, eventInfo)
```

This event is available only for buttons and it occurs when the button is pressed.

Parameter	Description
<b>me</b>	Object triggering the event
<b>eventInfo</b>	Details of triggered event

```
function buttonStd1_onMousePress(me, eventInfo) {
    //do something...
}
```

## onMouseRelease

```
void onMouseRelease (me, eventInfo)
```

This event is available only for buttons and it occurs when the button is released.

Parameter	Description
<b>me</b>	Object triggering the event
<b>eventInfo</b>	Details of triggered event

```
function buttonStd1_onMouseRelease(me, eventInfo) {
    //do something...
}
```

## onDataUpdate

```
boolean onDataUpdate (me, eventInfo)
```

This event occurs when data attached to the widget changes.

Parameter	Description
<b>me</b>	Object triggering the event
<b>eventInfo</b>	An object with the fields listed below (you can refer fields using "." - dot notation) oldValue = Widget value before the change newValue = Value which will be updated to the widget attrName = Attribute on which the event is generated index = Integer attribute index if any, default = 0 mode = When the user is writing to the widget. R in all others status.

The event is triggered before the value is passed to the widget. A JavaScript code can intercept the event and decide to avoid to update the widget by return true value.



Note: if there are additional macros associate at the event, all macros will be execute regardless of the return value used inside the JavaScript code.

```
function buttonStd1_onDataUpdate(me, eventInfo) {
if ( eventInfo.newValue > 100) {
//do something...
return true; // To avoid to continue and update
// the widget (e.g. not update the linked tag)
}
return false; // To continue and update the widget
// (e.g. update the linked tag)
}
```

## Page events

### onActivate

```
void onActivate( me, eventInfo )
```

This event occurs each time the page is displayed.

Parameter	Description
<b>me</b>	Object triggering the event
<b>eventInfo</b>	Reserved for future use

JavaScript will be executed when the page is active, that is when the page is loaded.

```
function Page1_onActivate(me, eventInfo) {
//do something...
}
```

## onDeactivate

```
void onDeactivate( me, eventInfo )
```

This event occurs when leaving the page.

Parameter	Description
me	Object triggering the event
eventInfo	Reserved for future use

```
function Page1_onDeactivate(me, eventInfo) {
    //do something...
}
```

## onWheel

```
void onMouseWheelClock( me, eventInfo )
```

This event occurs when a wheel device is moving (for example, a mouse wheel).

Parameter	Description
me	Object triggering the event
eventInfo	Details of triggered event

```
function Page1_onMouseWheelClock(me, eventInfo) {
    //do something...
}
```

## System events

System events can be related to:

- scheduler
- alarms
- a wheel device

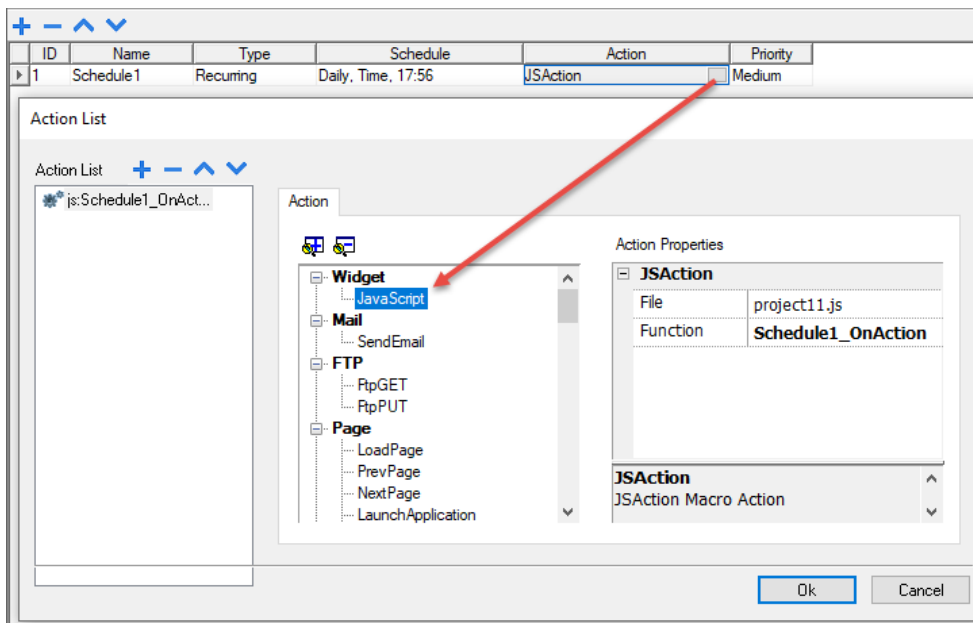


**Important: Make sure you do not duplicate JavaScript function names at page and project level. When a conflict happens, that is two functions with the same name in current page and at project level, the system execute the JavaScript callback at page level.**

When a JavaScript callback is not found in the current page, the system automatically searches for it at project level.

### Scheduler events

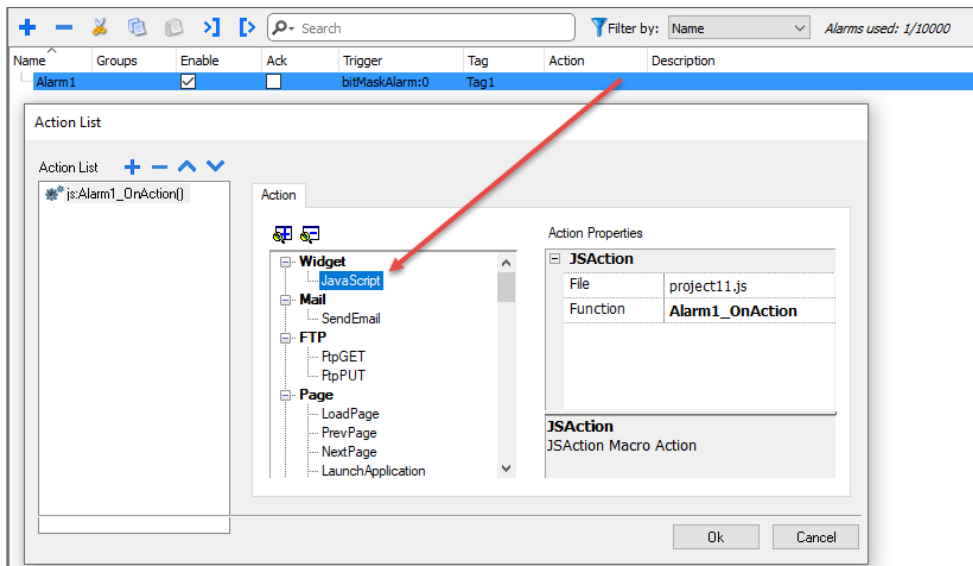
These events occur when triggered by the associated action in the scheduler.



You can edit the JavaScript from the **Project Properties** tab.

## Alarm events

These events occur when triggered by the associated alarm condition.



You can edit the JavaScript from the **Project Properties** tab.

## onWheel

```
void onMouseWheelClock( me, eventInfo )
```

This event occurs when a wheel device is moving (for example, a mouse wheel).

Parameter	Description
<code>me</code>	Object triggering the event
<code>eventInfo</code>	Details of triggered event

```
function Project1_onMouseWheelClock(me, eventInfo) {
    //do something...
}
```

## Objects

PB610 Panel Builder 600 uses JavaScript objects to access the elements of the page. Each object is composed of properties and methods that are used to define the operation and appearance of the page element. The following objects are used to interact with elements of the HMI device page:

Object	Description
Widget	This is the base class for all elements on the page including the page element
Page	This object references the current HMI device page. The page is the top-level object of the screen.
Group	This object associates a set of tags to allow uniform operation on a set of logically connected tags
Project	This object defines the project widget. The project widget is used to retrieve data about the project such as tags, alarms, recipes, schedules, tags and so on. There is only one widget for the project and it can be referenced through the project variable.
State	This object is the class holding the state of a variable acquired from the controlled environment. Beside the value itself, it contains the timestamp indicating when the value was collected and flags marking the quality of the value.

## Widget class objects

The Widget class is the base class for all the elements on a page including the page element.

Widget, in this case, is not used to indicate a specific screen object but a JavaScript class.

### Changing widget properties with JavaScript

If you want to change the properties of widgets with JavaScript set the widget property **Static Optimization** to **Dynamic**.



**Important: If the widget property Static Optimization is not set to Dynamic, changes to properties will be ignored.**

Whenever a call to `getWidget` fails, the remote debugger reports the following error:

*“Trying to access static optimized widget "label1". Disable widget static optimization to access widget from script.”*



This error is visible also using following code fragment:

```
var wgt;
try {
wgt = page.getWidget('label1');
} catch(err) {
alert("" + err);
}
```

## Widget properties

Some properties are common to all widgets.

### objectName

string objectName

Gets the name of the widget, a unique id.

```
function btnStd04_onMouseRelease(me) {
    var wgt = page.getWidget("rect1");
    var name = wgt.objectName;
}
```

(Available on web pages)

### x

number x

Gets or sets the widget 'x' position in pixels.

```
function btnStd1_onMouseRelease(me) {
    var wgt = page.getWidget("rect1");
    wgt.x = 10;
}
```

(Available on web pages)

### y

number y

Gets or sets the widget 'y' position in pixels.

```
function btnStd1_onMouseRelease(me) {
    var wgt = page.getWidget("rect1");
    wgt.y = 10;
}
```

```
}
```

(Available on web pages)

## width

number width

Gets or sets the widget width in pixels.

```
function btnStd1_onMouseRelease(me) {  
    var wgt = page.getWidget("rect1");  
    wgt.width = 10;  
}
```

(Available on web pages)

## height

number height

Gets or sets the widget height in pixels.

```
function btnStd1_onMouseRelease(me) {  
    var wgt = page.getWidget("rect1");  
    wgt.height = 10;  
}
```

(Available on web pages)

## visible

boolean visible

Gets or sets the widget visible state.

```
function btnStd4_onMouseRelease(me) {  
    var wgt = page.getWidget("rect1");  
    wgt.visible = false;  
}  
  
function btnStd5_onMouseRelease(me) {  
    var wgt = page.getWidget("rect1");  
    wgt.visible = true;  
}
```

## value

number value

Gets or sets the widget value.

```
function btnStd6_onMouseRelease(me) {  
    var wgt = page.getWidget("field1");  
    wgt.value = 100;  
}
```

## opacity

number opacity (range from 0 to 1)

Gets or sets the widget opacity. Values are decimals from 0 to 1, where 1 is 100% opaque.

```
function btnStd8_onMouseRelease(me) {  
    var wgt = page.getWidget("rect1");  
    wgt.opacity = 0.5;  
}
```

(Available on web pages)

## rotation

number rotation (in degrees)

Gets or sets the rotation angle for the widget. The rotation is done clockwise and by degrees, starting at the East position.

```
function btnStd9_onMouseRelease(me) {  
    var wgt = page.getWidget("rect1");  
    wgt.rotation = 45;  
}
```

(Available on web pages)

## userValue

string userValue

Gets or sets a user-defined value for the widget. This field can be used by JavaScript functions to store additional data with the widget.

```
function btnStd9_onMouseRelease(me) {  
    var wgt = page.getWidget("rect1");  
    wgt.userValue = "Here I can store custom data";  
}
```

Every widget has some specific properties that you can access using dot notation. For an up-to-date and detailed list of properties you can use the JavaScript Debugger inspecting the widget methods and properties.

## Widget methods

Some methods are common to all widgets.

### getProperty

```
object getProperty( propertyName, [index] )
```

Returns a property.

Parameter	Description
<b>propertyName</b>	String containing the name of property to get
<b>index</b>	Index of the element to get from the array (default = 0)

Almost all properties that are shown in the PB610 Panel Builder 600 **Properties** pane can be retrieved using the `getProperty` method. The index value is optional and only used for widgets that support arrays.

```
function buttonStd1_onMouseRelease(me, eventInfo) {
    var shape = page.getWidget("rect2");
    var y_position = shape.getProperty("y");
}
```

```
function buttonStd2_onMouseRelease(me, eventInfo) {
    var image = page.getWidget("multistate1");
    var image3 = image.getProperty("imageList", 2);
    //...
}
```

(Available on web pages)

### setProperty

```
boolean setProperty( propertyName, value, [index] )
```

Sets a property for the widget.

#### Parameters

Parameter	Description
<b>propertyName</b>	String containing the name of property to set
<b>value</b>	String containing the value to set the property.
<b>index</b>	Index of the element to set in the array (default = 0)

Almost all properties that are shown in the PB610 Panel Builder 600 **Properties** pane can be set by this method. The index value is optional and only used for Widgets that support arrays (for example, a MultiState Image widget). The `setProperty` method returns a boolean value (true or false) to indicate if the property was set or not.

```
function buttonStd1_onMouseRelease(me, eventInfo) {
    var setting_result = shape.setProperty("y", 128);
    if (setting_result)
        alert("Shape returned to start position");
}
```

```
function buttonStd2_onMouseRelease(me, eventInfo) {
    var image = page.getWidget("multistat1");
    var result = image.setProperty("imageList", "Fract004.png", 2);
    //...
}
```

(Available on web pages)

## Page object

This object references the current HMI device page. The page is the top-level object of the screen.

## Page object properties

Properties available at page level.

### backgroundColor

string backgroundColor (in format rgb(xxx, xxx, xxx) where xxx range from 0 to 255)

Page background color.

```
function btnStd11_onMouseRelease(me) {
    page.backgroundColor = "rgb(128,0,0)";
}
```

(Available on web pages)

### width

number width

Page width in pixels.

```
function btnStd05_onMouseRelease(me) {
    var middle_x = page.width / 2;
}
```

(Available on web pages, get only)

## height

number height

Page height in pixels.

```
function btnStd05_onMouseRelease(me) {
    var middle_y = page.height / 2;
}
```

(Available on web pages, get only)

## userValue

string userValue

Gets or sets a user-defined value for the widget. This field can be used by JavaScript functions to store additional data with the page.

```
function btnStd9_onMouseRelease(me) {
    page.userValue = "Here I can store custom data";
}
```

(Available on web pages)

# Page object methods

Methods that can be used at page level.

## getWidget

object getWidget( wgtName )

Returns the widget with the given name.

Parameter	Description
<b>wgtName</b>	String containing the widget name

### Return value

An object representing the widget. If the widget does not exist, null is returned.

```
function btnStd1_onMouseRelease(me) {
    var my_button = page.getWidget("btnStd1");
}
```

(Available on web pages)

## setTimeout

```
number setTimeout( functionName, delay )
```

Starts a timer to call a given function after a given delay.

Parameter	Description
<b>functionName</b>	String containing the name of function to call
<b>delay</b>	Delay in milliseconds

### Return value

A number corresponding to the timerID.

```
var duration = 3000;
var myTimer = page.setTimeout("innerChangeWidth()", duration);
```

(Available on web pages)

## clearTimeout

```
void clearTimeout( timerID )
```

Stops and clears the timeout timer with the given timer.

Parameter	Description
<b>timerID</b>	Timer to be cleared and stopped

```
var duration = 3000;
var myTimer = page.setTimeout("innerChangeWidth()", duration);
// do something
page.clearTimeout(myTimer);
```

(Available on web pages)

## setInterval

```
number setInterval( functionName, interval )
```

Starts a timer that executes the given function with the given interval.

Parameter	Description
<b>functionName</b>	String containing the name of function to call
<b>interval</b>	Interval in milliseconds

### Return value

A number corresponding to the timerID.

```
var interval = 3000;
var myTimer = page.setInterval("innerChangeWidth()", interval);
```

(Available on web pages)

## clearInterval

```
void clearInterval( timerID )
```

Stops and clears the interval timer with the given timer.

Parameter	Description
timerID	Timer to be cleared and stopped

```
var interval = 3000;
var myTimer = page.setInterval("innerChangeWidth()", interval);
// do something
page.clearInterval(myTimer);
```

(Available on web pages)

## clearAllTimeouts

```
void clearAllTimeouts()
```

Clears all the timers started.

```
page.clearAllTimeouts();
```

(Available on web pages)

# Project object

This object defines the project widget. The project widget is used to retrieve data about the project such as tags, alarms, recipes, schedules, tags and so on. There is only one widget for the project and it can be referenced through the project variable.

## Project object properties

Properties to be set at project level.

### startPage

```
string startPage
```

Page shown when the project is started.

```
var startPage = project.startPage;
```



```
project.startPage = "Page2.jmx";
```

## Project object methods

Methods to be used at project level.

### nextPage

```
void nextPage()
```

The script executes the Next page action.

```
project.nextPage();
```

(Available on web pages)

### prevPage

```
void prevPage()
```

The script executes the previous page action.

```
project.prevPage();
```

(Available on web pages)

### lastVisitedPage

```
void lastVisitedPage()
```

The script executes the last visited page action.

```
project.lastVisitedPage();
```

(Available on web pages)

### homepage

```
void homePage()
```

The script executes the Home page action.

```
project.homePage();
```

(Available on web pages)

### loadPage

```
void loadPage(pageName)
```

The script executes to load the set page defined in the script.

```
project.loadPage("Page5.jmx");
```

(Available on web pages)



**WARNING: When page change, all active time events are forced to removed and the JavaScript procedure will run until the end before switch to the new page.**

## showDialog

```
void showDialog(pageName)
```

The script executes to show the dialog page.

```
project.showDialog("Dialog.jmx");
```

(Available on web pages)

## closeDialog

```
void closeDialog()
```

The script executes to close the currently-opened dialog page.

```
project.closeDialog();
```

(Available on web pages)

## showMessage

```
void showMessage( message )
```

The script executes to display the message popup.

```
project.showMessage("Hi This is test message");
```

(Available on web pages)

## getGroup

```
number getGroup( groupName, groupInstance, [callback] )
```

Fast read method; this gets the values of all tags in a group.

Parameter	Description						
<b>groupName</b>	<p>String containing the names of the groups.</p> <p>The and/or expression to retrieve tags list from multiple group is supported.</p> <table border="0"> <tr> <td> </td> <td>OR operator</td> </tr> <tr> <td>&amp;</td> <td>AND operator</td> </tr> <tr> <td>(...)</td> <td>The brackets can be used to define how evaluate the expression</td> </tr> </table> <p>Examples:</p> <ul style="list-style-type: none"> <li>• <code>project.getGroup("one", group);</code></li> <li>• <code>project.getGroup("(one two)", group);</code></li> <li>• <code>project.getGroup("((one&amp;two)*three)", group);</code></li> </ul>		OR operator	&	AND operator	(...)	The brackets can be used to define how evaluate the expression
	OR operator						
&	AND operator						
(...)	The brackets can be used to define how evaluate the expression						
<b>groupInstance</b>	Group element to be filled						
<b>callback</b>	String containing the name of the function to be called when the group is ready						

### Return value

A number value that is the status: 1 for success, 0 for fail.

```
var group = new Group();
var status = project.getGroup ("enginesettings", group);
if (status == 1) {
    var value = group.getTag("Tag1");
    if (value!=undefined) {
        // do something with the value
    }
}
```

```
var g = new Group();
var status = project.getGroup ("enginesettings", g,
    function (groupName, group) { fnGroupReady(groupName, group);} );

function fnGroupReady(groupName, group) {
    var val = group.getTag("Tag1");
    if (val!=undefined) {
        // do something with the value
    }
}
```

(Available on web pages)

## getTag

object getTag( tagName, state, index, forceRefresh)

```
void getTag( tagName, state, index, callback, forceRefresh)
```

It returns the tag value or the complete array if index value is -1 of the given tagName.

Parameter	Description
<b>tagName</b>	String of tag name
<b>state</b>	State element to be filled
<b>index</b>	Index if the tag is of array type. -1 returns the complete array. Default = 0.
<b>callback</b>	Function name if an asynchronous read is required. Default = "".
<b>forceRefresh</b>	(Optional parameter) True = the Runtime will read an updated value of the tag directly from the device. Default is false.

### Return value

Tags value is returned. If tag is array type and index = -1 then the complete array is returned. For non-array tags provide index as 0.

```
var state = new State();
var value = project.getTag("Tag1", state, 0);
//
//for non array type
//tags index is not considered, so can be left as 0
//
if (value!=undefined) {
//...do something with s
}
```

```
var state = new State();
project.getTag("Tag1", state, -1,
    function(tagName, tagState) { fnTagReady(tagName, tagState); });
function fnTagReady(tagName, tagState) {
    if (tagName=="Tag1") {
        var myValue = tagState.getValue();
    }
}
```

(Available on web pages)

## setTag

number setTag( tagName, tagValue, [index], [forceWrite] )

Sets the given tag in the project. Name and value are in strings.

Parameter	Description
<b>tagName</b>	String of tag name
<b>tagValue</b>	Object containing the value to write
<b>index</b>	Index if the tag is of array type. -1 pass the complete array. Default = 0.
<b>forceWrite</b>	Boolean value for enabling force write of tags, the function will wait for the value to be written before it returns back. Default = false.

### Return value

Integer value for denoting success and failure of action when forceWrite is true. 0 means success and -1 means failure. If forceWrite is false, returned value will be undefined.

```
var val = [1,2,3,4,5];
var status = project.setTag("Tag1", val, -1, true);
if (status == 0) {
    // Success
} else {
    // Failure
}
```

```
var val = "value";
project.setTag("Tag1", val);
```

(Available on web pages)

## updateSystemVariables

```
void project.updateSystemVariables()
```

Force system variables to refresh.

```
project.updateSystemVariables()
```

## selectAllAlarms

```
void project.selectAllAlarms(bool selected)
```

Select/unselect all alarms

```
project.selectAllAlarms(true)
```

(Available on web pages)

## ackAlarms

```
void project.ackAlarms()
```

Acknowledge all selected alarms

```
project.selectAllAlarms(true);
project.ackAlarms();
project.selectAllAlarms(false);
```

(Available on web pages)

## resetAlarms

```
void project.resetAlarms()
```

Reset all selected alarms

```
project.selectAllAlarms(true);
project.resetAlarms();
project.selectAllAlarms(false);
```

(Available on web pages)

## enableAlarms

```
void project.enableAlarms()
```

Enable all selected alarms

```
project.selectAllAlarms(true);
project.enableAlarms();
project.selectAllAlarms(false);
```

(Available on web pages)

## getRecipeItem

```
object getRecipeItem (recipeName, recipeSet, recipeElement)
```

Gets the value of the given recipe set element.

Parameter	Description
<b>recipeName</b>	String representing the recipe name
<b>recipeSet</b>	String representing the recipe set, can be either the recipe set name or 0 based set index.
<b>recipeElement</b>	String representing the recipe Element, can be either the element name or 0 based element index.

### Return value

An object with the value of the recipe. undefined is returned if invalid. If of type array, an array object type is returned.

```
var value = project.getRecipeItem("recipeName", "Set", "Element");
```

## setRecipeItem

number setRecipeItem (recipeName, recipeSet, recipeElement, value )

Gets the value of the given recipe set element.

Parameter	Description
<b>recipeName</b>	String representing the recipe name
<b>recipeSet</b>	String representing the recipe set, can be either the recipe set name or 0 based set index.
<b>recipeElement</b>	String representing the recipe Element, can be either the element name or 0 based element index.
<b>value</b>	An object containing the value to store in the recipe. It can be an array type.

### Return value

Integer value for denoting success and failure of action. A '0' means success and '-1' means failure.

```
var val = [2,3,4];
project.setRecipeItem("recipeName", "Set", "Element", val);
if (status == 0) {
    // Success
} else {
    // Failure
}
```

## downloadRecipe

void downloadRecipe (recipeName, recipeSet )

Downloads the recipe set to the corresponding tag.

Parameter	Description
<b>recipeName</b>	String representing the recipe name
<b>recipeSet</b>	String representing the recipe set, can be either the recipe set name or 0 based set index.

```
project.downloadRecipe("recipeName", "Set");
```

## uploadRecipe

void uploadRecipe (recipeName, recipeSet )

Uploads the value of tags into the provided recipe set.

Parameter	Description
<b>recipeName</b>	String representing the recipe name
<b>recipeSet</b>	String representing the recipe set, can be either the recipe set name or 0 based set index.

```
project.uploadRecipe("recipeName", "Set");
```

## launchApp

```
void launchApp( appName, appPath, arguments, singleInstance)
```

Executes an external application.

Parameter	Description
<b>appName</b>	String containing the application name
<b>appPath</b>	String containing the application absolute path
<b>Arguments</b>	String containing the arguments to be sent to application
<b>singleInstance</b>	true = only single instance allowed, false = multiple instances allowed

Note the pathname's syntax depend from the native OS format (see ["HMI devices capabilities" on page 613](#)).

On **WinCE devices**, the pathname's syntax need double backslash character.

```
project.launchApp("PDF.exe", "\\Flash\\QTHMI\\PDF", "\\USBMemory\\file.pdf", "true");
```

On **Linux devices**, the pathname's syntax need slash character (even double slash character is permitted).

```
project.launchApp
("pdfViewer", "/mnt/data/hmi/qthmi/deploy", "/mnt/usbmemory/test.pdf", "true");
```

## getClientType

```
string getClientType()
```

Return the client type

Client Type	Description
local	Running on HMI device
remote	Running on HMI Client client
web	Running on Web client

```
var clientType = project.getClientType();
if (clientType=="web") {
    // Currently running on web client
} else if (clientType=="remote") {
    // Currently running on HMI Client
} else if (clientType=="local") {
    // Currently running on HMI Device
}
```

(Available on web pages)



## login

```
int project.login("username", "password")
```

Access to the system with the given credentials

```
var ReplyCode;
ReplyCode = project.login("admin", "admin");
if (ReplyCode != 0) {
    alert("Access denied");
}
```

### Return value

0	No Error
1	Error: You are not authorized.
2	Error: Connection lost with the Runtime.
3	Error: The username or password you entered is incorrect
4	Error: The password entered is incorrect
5	Error: Action cannot be executed
6	Error: Passwords do not match
7	Error: Password length too short
8	Error: Password must contain numbers
9	Error: Password must contain special characters
10	Error: Password must be different than previous passwords
11	Error : User already exist
12	Error: Password cannot be empty
13	Error: Your password has expired
14	Warning: Your password will expire soon

## logout

```
project.logout(AllowDefaultUser)
```

Exiting the system

```
project.logout(); // Logout even from default user
project.logout(true); // Logout even from default user
project.logout(false); // Logout only if not logged as default user
```

# Project object widgets

## getCurrentPageName

```
string getCurrentPageName()
```

Return the name of current active page

```
// Get PageMgr widget
var pageMgr = project.getWidget( "_PageMgr" );

// Show Current Page
var currentPageName = pageMgr.getCurrentPageName();
project.showMessage( "Current active page is: " + currentPageName );
```

(Available on web pages)

## hasPage

```
boolean hasPage(string pageName)
```

Return true if the page exist, false otherwise

```
// Get PageMgr widget
var pageMgr = project.getWidget( "_PageMgr" );

//Page exists
var pageExists = pageMgr.hasPage( "Page10" );
if (pageExists) {
    project.showMessage( "Page10 exists" );
} else {
    project.showMessage( "Hei Page10 not exists!" );
}
```

(Available on web pages)

## curLangCode

```
string curLangCode
```

Property of MultiLangMgr widget. Contains the code of the active language.

```
// Get MultiLangMgr widget
var MultiLangMgr = project.getWidget( "_MultiLangMgr" );

// Show curLangCode
var curLangCode = MultiLangMgr.curLangCode;
project.showMessage( "Current active language is: " + curLangCode );
```

# Print reports object

## printGfxReport

```
void printGfxReport( reportName, silentMode)
```

Prints the graphic report specified by reportName.

Parameter	Description
<b>reportName</b>	String containing the report name
<b>silentMode</b>	True = silent mode enabled. No printer settings dialog is displayed.

```
project.printGfxReport("Report Graphics 1", true);
```

## printText

```
void printText( text, silentMode)
```

Prints a fixed text.

Parameter	Description
<b>text</b>	String to print
<b>silentMode</b>	True = silent mode enabled. No printer settings dialog is displayed.

```
project.printText("Hello I Am Text Printing",true);
```

## printBytes

```
void printBytes( text, silentMode)
```

Prints a hexadecimal string representing data to print. For example, "1b30" to print < ESC 0 >

Parameter	Description
<b>text</b>	Hexadecimal string to print
<b>silentMode</b>	True = silent mode enabled. No printer settings dialog is displayed.

```
project.printText("1B30",true); // Print: ESC 0
```

## emptyPrintQueue

```
void emptyPrintQueue()
```

Empties the print queue. Current job will not be aborted.

```
project.emptyPrintQueue();
```

## pausePrinting

```
void pausePrinting();
```

Suspends printing operations. Will not suspend the print of a page already sent to the printer.

```
project.pausePrinting();
```

## resumePrinting

```
void resumePrinting();
```

Resumes previously suspended printing.

```
project.resumePrinting();
```

## abortPrinting

```
void abortPrinting();
```

Aborts current print operation and proceed with the next one in queue. This command will not abort the print of a page already sent to the printer.

```
project.abortPrinting();
```

## printStatus

```
project.printStatus;
```

Returns a string representing current printing status.

Status string	Description
error	An error occurred during printing
printing	Ongoing printing
idle	System is ready to accept new jobs
paused	Printing has be suspended

```
var status = project.printStatus;
project.setTag("PrintStatus", status);
```

## printGfxJobQueueSize

```
project.printGfxJobQueueSize;
```

Returns the number of graphic reports in queue for printing.

```
var gfxqueuesize = project.printGfxJobQueueSize;
project.setTag("printGfxJobQueueSize",gfxqueuesize);
```

## printTextJobQueueSize

```
project.printTextJobQueueSize;
```

Returns the number of text reports in queue for printing.

```
var textjobqueuesize = project.printTextJobQueueSize;
project.setTag("printTextJobQueueSize",textjobqueuesize);
```

## printCurrentJob

```
project.printCurrentJob;
```

Returns a string representing current job being printed

```
var currentjob = project.printCurrentJob;
project.setTag("printCurrentJob",currentjob);
```

## printActualRAMUsage

```
project.printActualRAMUsage;
```

Returns an estimate of RAM usage for printing queues

```
var myVar = project.printActualRAMUsage;
alert(" actual ram usage is "+ myVar);
```

## printRAMQuota

```
project.printRAMQuota;
```

Returns the maximum allowed RAM usage for printing queues

```
var ramquota = project.printRAMQuota;
project.setTag("printRAMQuota",ramquota);
```

## printActualDiskUsage

```
project.printActualDiskUsage;
```

Returns the spool folder disk usage (for PDF printouts)

```
var myVar1 = project.printActualDiskUsage;
alert(" actual disk usage is "+ myVar1);
```

## printDiskQuota

```
project.printDiskQuota;
```

Returns the maximum allowed size of spool folder (for PDF printouts).

```
var ramquota = project.printRAMQuota;  
var diskquota = project.printDiskQuota;
```

## printSpoolFolder

```
project.printSpoolFolder;
```

Returns current spool folder path (for PDF printouts).

```
var spoolfolder = project.printSpoolFolder;  
project.setTag("printSpoolFolder", spoolfolder);
```

## printPercentage

```
project.printPercentage;
```

Returns current job completion percentage (meaningful only for multipage graphic reports)

```
var percentage = project.printPercentage;  
project.setTag("printPercentage", percentage);
```

# Group object

A group is a basic logical element that associates a set of logical tags.

## Group object methods

Methods that can be used with group objects.

### getTag

```
object getTag( TagName )
```

Gets the tag specified by TagName from the group object.

Parameter	Description
TagName	String representing the tag name

### Return value

An object that is the value of the tag or, if tag value is an array, the complete array. If you need to retrieve an element of the array, check the method `getTag` available in the project object. Undefined is returned if tag is invalid.

```
var group = new Group();
project.getGroup("GroupName", group);
var value = group.getTag("Tag1");
```

(Available on web pages)

## getCount

number getCount()

Returns total number of tags in this group.

```
var group = new Group();
project.getGroup("GroupName", group);
var value = group.getCount();
```

(Available on web pages)

## getTags

object getTags()

Returns the list of all tags in group.

```
function {
var group = new Group();
project.getGroup("enginesettings", group);
var tagList = group.getTags();
for(var i = 0; i < tagList.length; i++){
    var tagName = tagList[i];
    //do something...
};
```

(Available on web pages)

# State object

This is the class holding the state of a tag acquired from the controlled environment.

## State object methods

Methods to be used with state objects.

### getQualityBits

number getQualityBits()

Returns an integer - a combination of bits indicating tag value quality.

```
var state = new State();
var value = project.getTag("Tag1", state, 0);
var qbits = state.getQualityBits();
```

(Available on web pages)

## getTimestamp

number getTimestamp()

Returns time the value was sampled.

### Return value

A number containing the timestamp (for example 1315570524492).



Note: Date is a native JavaScript data type.

```
var state = new State();
var value = project.getTag("Tag1", state, 0);
var ts = state.getTimestamp();
```

## isQualityGood

boolean isQualityGood()

Returns whether the value contained in this state object is reliable.

### Return value

A Boolean true if quality is good, false otherwise.

```
var state = new State();
var value = project.getTag("Tag1", state, 0);
if (state.isQualityGood()) {
    // do something...
}
```

(Available on web pages)

## Keywords

Global objects are predefined and can be referenced by the following names.

### page

object page

References the page object for the current page.



```
function btnStd04_onMouseRelease(me) {
    var wgt = page.getWidget("rect1");
    var name = wgt.objectName;
}
```

## project

object project

References the project widget.

```
var group = new Group();
project.getGroup("GroupName", group);
var value = group.getCount("Tag1");
```

# Global functions

## print

void print( message )

Prints a message to the HMI Logger window.

Parameter	Description
<b>message</b>	Message string

```
print("Test message");
```

## alert

void alert( message )

Displays a pop-up dialog with the given message. The user must press the **OK** button in the dialog to continue with the execution of the script.

Parameter	Description
<b>message</b>	Message string



Note: The alert function may be used for debugging JavaScript functions.

```
alert("Test message");
```

(Available on web pages)

# Handling read/write files

## Create folder

```
boolean fs.mkdir(strPath);
```

Creates a folder, if not already existing, in the specified path. Returns true on success and false if it fails.

Parameter	Description
<b>strPath</b>	Path string

## Remove folder

```
boolean fs.rmdir(dirPath);
```

Remove directory at strPath if exists and empty. Returns true on success and false if it fails.

Parameter	Description
<b>dirPath</b>	Folder string

## Read folder content

```
object fs.readdir(dirPath);
```

Reads the contents of a folder. Returns an array of the names of the files in the folder excluding '.' and '..'. Returns empty list if it fails.

Parameter	Description
<b>dirPath</b>	Folder string

## Read file

```
object fs.readFile(strfile [,strFlag]);
```

Opens the strFile file in read mode, reads its contents and returns it.

Parameter	Description
<b>strFile</b>	File name string
<b>strFlag</b>	Read file mode: <ul style="list-style-type: none"> <li>• "b": reads and returns as binary file (otherwise returns a text file)</li> </ul>

## Write file

```
fs.writeFile(strFile, fileData, [strFlag]);
```

Creates the strFile file if not present. Opens the strFile file in write mode and writes the data fileData to the file.

Parameter	Description
<b>strFile</b>	File name string
<b>fileData</b>	Data to be write on the file in byte array
<b>strFlag</b>	Write file mode: <ul style="list-style-type: none"> <li>• “a”: appends fileData to the end of the text file</li> <li>• “r”: replaces the contents of the file with fileData</li> <li>• “ab”: appends fileData to the end of the binary file</li> <li>• “rb”: replaces the contents of the binary file with fileData</li> </ul>

Default flag is for writing text file in append and write mode. File path will be created if not present.

Returns -1 if write error occurs.

## Append file

```
int fs.appendFile(strFile, fileData);
```

If the files does not exist creates it, otherwise append to existing file. Returns the number of character written or -1 on error.

Parameter	Description
<b>strFile</b>	File name string
<b>fileData</b>	Data to be write on the file in byte array

## File exists

```
boolean fs.exists(strPath)
```

Returns true if the file or folder exists at strPath.

Parameter	Description
<b>strPath</b>	Path string

## Remove file

```
boolean fs.unlink(strPath)
```

Removes the given file at strPath from filesystem if exists. Returns true on success and false if it fails.

Parameter	Description
<b>strPath</b>	Path string

## File status

```
object fs.stat(strPath)
```

Retrieves information on the file/folder present at the specified path.

Parameter	Description
<b>strPath</b>	File/folder path string

```
var fileStats = var fs.stat(strPath)
```

fileStats.isFile	True if path is a file
fileStats.isDir	True if path is a folder
fileStats.size	Size in bytes of that file
fileStats.atime	Date object representing the last read access time
fileStats.mtime	Date object representing the last write access time
fileStats.ctime	Date object representing the creation time
fileStats.perm	File permissions

If path is invalid both isFile and isDir fields return false.

### File permission table

0x4000	File is readable by the owner of the file
0x2000	File is writable by the owner of the file
0x1000	File is executable by the owner of the file
0x0400	File is readable by the user
0x0200	File is writable by the user
0x0100	File is executable by the user
0x0040	File is readable by the group
0x0020	File is writable by the group
0x0010	File is executable by the group
0x0004	File is readable by anyone
0x0002	File is writable by anyone

## Important notes on file handling

Path for files and folders are expected to be UNIX style. This means the backslash character (\) is not recognized. Use slash character (/) instead.

File system object is a client side object. So operations are performed on local file system, not on server file system.

Current JavaScript API to get access at the device file system has been designed to manipulate small files. When a file is read, the entire file contents is temporarily stored inside the RAM available for JavaScript environment (16MB) and an

exception is raised when there is not enough available memory. Good programming practice is to include the `fs.readFile()` call inside a try/catch block.

## Sign in from JavaScript

Using the `project.login()` and `project.logout()` function is possible automatize the user sign in from a remote device. This could be useful, e.g., to perform the sign in by reading a user badge with a badge reader device.

This chapter show an example of how configure the application to manage the sign in by a remote device.

### The application must have a default user

Since the project's functions are working only when the application is active, the application must start with a default user, maybe with read only privilege. Reading the badge, the application can be switched to a user with additional privilege. Later, the logout command will reactivate the default user without any particular privileges

In the below example we are using three tags to communicate with the remote device:

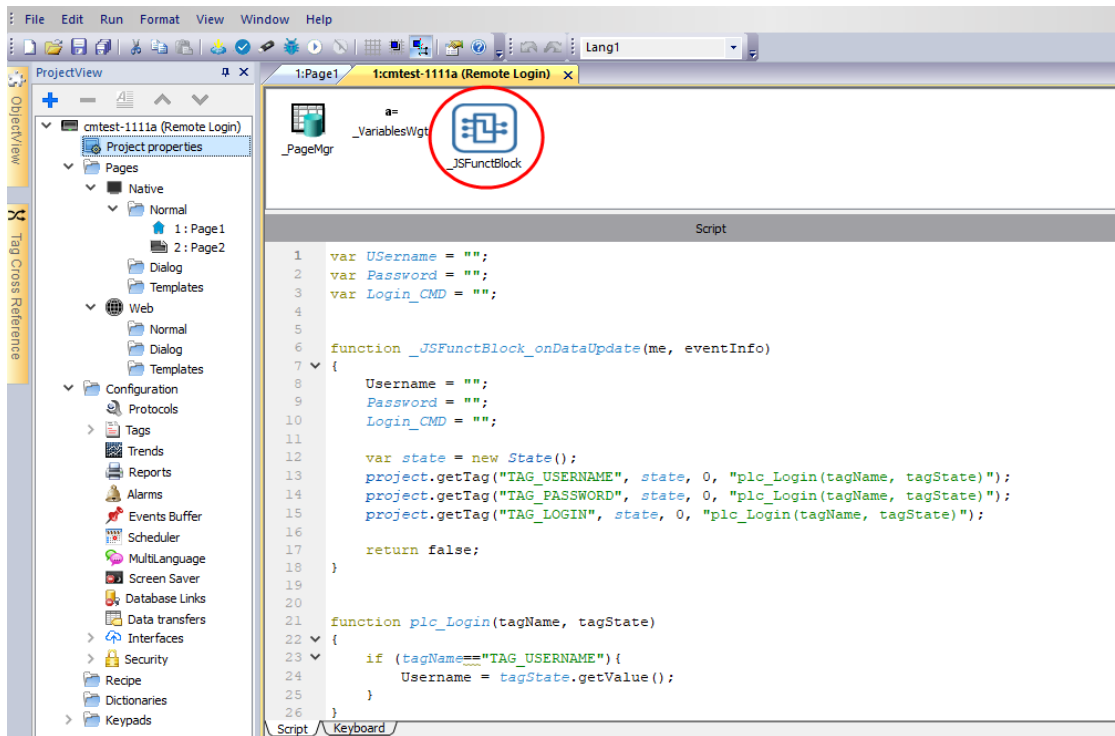
- TAG\_USERNAME
- TAG\_PASSWORD
- TAG\_LOGIN

The TAG\_LOGIN will be the command code to execute.

The remote device has to fill the required TAG\_USERNAME and TAG\_PASSWORD parameters, then fill the TAG\_LOGIN parameter with the required login or logout command. Engine on HMI-RUNTIME will detect the TAG\_LOGIN changes and perform the required command, then reset the TAG\_LOGIN to its idle status.

TAG_LOGIN Commands	
0	Idle
1	Login request
2	Logout request

At the project level, we have to add a JavaScript function block to detect when TAG\_LOGIN will changes. The JavaScript code attached at the onDataChange Action of the JavaScript function block will execute the required login/logout command.



### The JavaScript code attached at the OnDataUpdate Action

```

var Username;
var Password;
var Login_CMD;

function _JSFunctBlock_onDataUpdate(me, eventInfo)
{
    Username = "";
    Password = "";
    Login_CMD = "";

    var state = new State();
    project.getTag("TAG_USERNAME", state, 0, "plc_Login(tagName, tagState)");
    project.getTag("TAG_PASSWORD", state, 0, "plc_Login(tagName, tagState)");
    project.getTag("TAG_LOGIN", state, 0, "plc_Login(tagName, tagState)");
    return false;
}

function plc_Login(tagName, tagState)
{
    if (tagName=="TAG_USERNAME") {
        Username = tagState.getValue();
    }
    if (tagName=="TAG_PASSWORD") {
        Password = tagState.getValue();
    }
    if (tagName=="TAG_LOGIN") {
        Login_CMD = tagState.getValue();
    }
}

```

```

if (Username!="" && Password!="" && Login_CMD!=""){
  if (Login_CMD==1) {
    Reply = project.login(Username, Password);
  };
  if (Login_CMD==2) {
    Reply = project.logout(false); // Logout only if not logged as default
user
  };
  project.setTag("TAG_LOGIN", 0);
  project.setTag("TAG_REPLY", parseInt(Reply));
}
}

```

See also:

- "login" on page 566

## Limitations in working with widgets in JavaScript

Widgets cannot be instantiated by JavaScript, they can only be accessed and changed. If you need additional widgets on the page, you can add hidden widgets on the page, and then display or position them using JavaScript.

## Debugging of JavaScript

PB610 Panel Builder 600 and HMI Runtime include a JavaScript debugger.

Two types of debuggers are available:

- Runtime debugger: a debugger running directly on the HMI device
- Remote debugger: a debugger running on a remote computer connected to the HMI device via Ethernet (usually computer running PB610 Panel Builder 600)

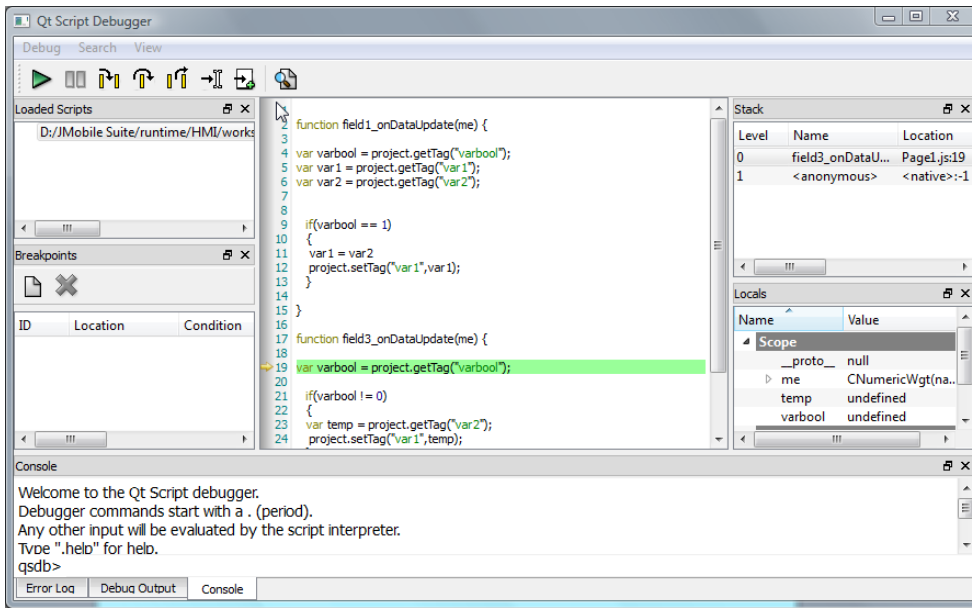
### Enabling debugging

In the **Properties** pane of a page, set **JavaScript Debug** to **true**.

Project Widget		Page	
Id	Project	Id	Page1
Full Path		Width	1024
Version		Height	768
Context Menu	on delay	Background	<input type="checkbox"/> [255, 255, ;
Developer Tools	false	Template	none
Keyboard	true	Static File Type	png
JavaScript Debug	true	JavaScript Debug	true
Allow JavaScript Remote	true		

For schedulers and alarms debugging, enable JavaScript Debug in Project properties.

In the HMI Runtime, when the events are called, the debugger will show the debug information. In the **Locals** pane you can inspect all variables and elements.



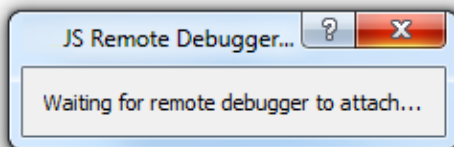
For a complete reference guide about JavaScript Debugger refer to :

<http://qt-project.org/doc/qt-4.8/qtscriptdebugger-manual.html>

## Remote JavaScript Debugger

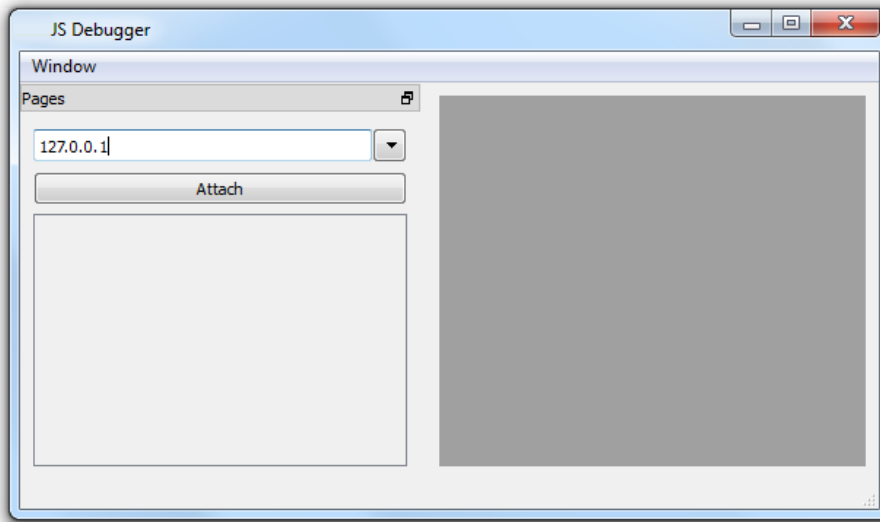
**Path: Run> Start JS Remote Debugger**

1. Set the **Allow JavaScript Remote** and the **JavaScript Debug** parameters in the project Properties to true in all the pages where debugging is required.
2. Download the project: the following message is displayed on the runtime.



3. In the **JS Debugger** window, select the IP of the HMI device and click **Attach** to connect the debugger to the HMI device.





Remote JavaScript debugger connects to HMI Runtime using port 5100/TCP.



Note: The Remote JavaScript debugger tool is not supported in HMI Client.

## JavaScript Memory Usage

When the memory exceeds the maximum, an out of memory exception is thrown with a custom message. Please note that we do not have a fine control over the actual memory usage so it is mainly a soft limit. Moreover we can't forbid the allocation (this will break the engine implementation), so exception is thrown only when the memory is already over the limit. Before raising the exception, a garbage collection is forced to see if some memory can be freed.

JavaScript memory limit can be accessed from the global object **\$EngineMemory**. The default is 16MB, which should be enough for the typical JavaScript usage (mainly control, without many allocations).

- `$EngineMemory.setLimit()`  
set maximum memory allowed for JavaScript (the default limit is 0x00FFFFFF)
- `$EngineMemory.getLimit()`  
get maximum memory allowed for JavaScript
- `$EngineMemory.getSize()`  
get currently used memory from JS (fastMallocStat)

### Test memory exception

To generate and test memory exception you can use the following snippet. Please note that we need to reset the memory limit to 0xffffffff to be able to run the alert, otherwise the memory allocations required to pop up the alert would fail.

```
try
{
    // Generate out-of-memory error
    var a = [];
    while(1)
    {
        a.push("a");
    };
} catch(e)
{
    // Ensure there is enough memory to pop up error message
    $EngineMemory.setLimit(0xffffffff);
    alert("Exception: " + e);
};
```



# 42 Handling Gestures

---

Some widgets have the capability to detect and manage pan and pinch gestures.

- Trends (see ["Trend widget gestures" on page 313](#) for details)
- Alarms Widget
- Combo box Widget
- Table Widget
- PDF Viewer
- Gesture Area Widget. Special widget designed to customize handling of gesture events (see ["Gesture area widget" on page 473](#) for details)

For widgets based on table presentation, when the **Scrollbars Type** parameter has been set to "Gesture", the pan gesture is used to smoothly scroll the table.

- Alarms
- Control List



**WARNING: Pinch and Rotate gesture requires two fingers. They are available only with HMI devices supporting multi touch operation (see ["HMI devices capabilities" on page 613](#))**



Tip: Using multi touch HMI device you can implement safe commands by programming a command to be executed only when two buttons are pressed at the same time.



PAN



PINCH



ROTATE



# 43 Web access

---

PB4Web allows users to access HMI projects from a remote web browser running on a computer or on a mobile device such as a tablet or a phone. With PB4Web, users can create a web project to display at a remote location the same graphical display shown on the HMI device. PB4Web projects are based on HTML5 technology which means that no plug-ins or external software is needed for displaying the information.

This document assumes that you have a basic understanding of how to operate the web browser on your mobile devices as well as how to set up a connection to the HMI device where the server is running. For example, you must know how to set-up Wi-Fi access if you are working with tablet or phone devices to access the PB4Web pages on the HMI device.

---

<b>Supported platforms and browsers</b> .....	<b>588</b>
<b>Web pages</b> .....	<b>588</b>
<b>Testing the Web project</b> .....	<b>590</b>
<b>Downloading the Web project</b> .....	<b>591</b>
<b>Web connectivity issues</b> .....	<b>591</b>
<b>Web supported features</b> .....	<b>592</b>
<b>Troubleshooting and FAQ</b> .....	<b>595</b>
<b>Privacy</b> .....	<b>596</b>

# Supported platforms and browsers

PB4Web supports 3 platforms:

- web, for desktop browsers,
- phone, for smart phone devices
- tablet, for tablet devices

You can therefore create pages of different content and size for the different platforms. For example, you may want to create a set of smaller pages in your project for phones whereas you will use full size pages for desktop web browsers and tablets.

## Working with a computer

PB4Web works with all modern web browsers. The following browsers have been tested for compatibility with PB4Web:

- Mozilla Firefox 52+
- Microsoft Edge 42+
- Apple Safari 11+
- Google Chrome 57+

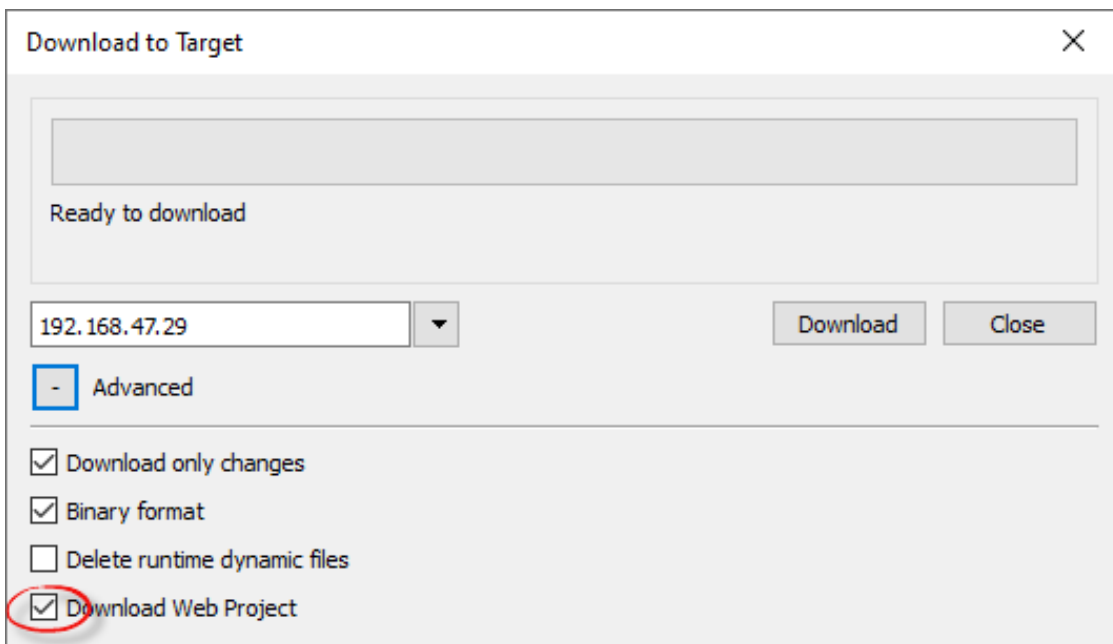
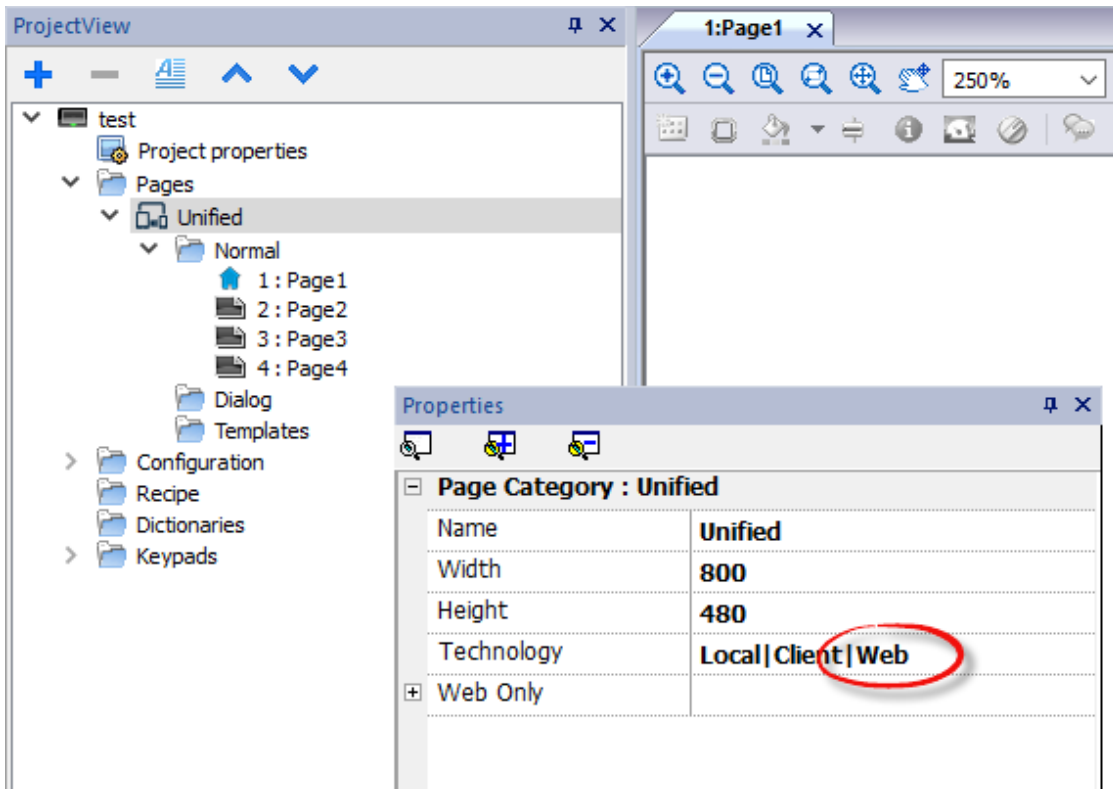
## Working with tablets or phones

PB4Web works with most tablet and phone devices. The following tablets have been tested for compatibility with PB4Web:

- iOS 10+ - Mobile Safari
- Android 7+ - Chrome for Android 55+

## Web pages

To enable web clients to access at the pages is necessary to include the "Web" reference in the Technology parameter of the page's category and make sure that when you download the project to the HMI device the **Download Web Project** option is selected (normally checked by default).



If the application needs to send different pages to different web clients (e.g. Smart phone instead of PC browser, etc.), have a look at the "Differentiated pages" on page 73 chapter.

## Web page properties

Any widgets and features can be used in PB610 Panel Builder 600; however, not all features are currently available in PB4Web. If the project includes a feature that is not available, PB4Web will still work correctly but the feature will not be



available on the remote client device. See "[Web supported features](#)" on page 592 for a list of the features supported in PB4Web and of the existing limitations.

You can use the **Project Validator** tool to check if your project contains widgets configured with properties that are currently not supported in Web technology (see "[Project Validator](#)" on page 70)

In addition to the standard page properties, there is an additional property to configure how the page will be adapted to the browser's viewport.

Property	Description
<b>Fit to Screen Size</b>	<p>How the page will adapted to the browser's viewport</p> <ul style="list-style-type: none"> <li>• None</li> <li>• Fit to Screen Simple modify the zoom level to adapted the page to viewport of the browser</li> <li>• Responsive Design Smart modify the zoom level to adapted the page to viewport of the browser respecting the restrictions defined inside the grid layout</li> </ul>

## Redirect to specific page using url request

You can access a specific web page by entering an URL with this syntax:

```
http://address/index.html?loadPage=pageName
```

# Testing the Web project


You can test your PB4Web project using the online simulator opening a standalone web page directly from a browser.

## Testing with the online simulator

PB610 Panel Builder 600 includes an web server in the online simulator. You can start the simulator and access your PB4Web project from a web browser. The pages will be served from the simulator.

1. Create your project (see "[Web pages](#)" on page 588).
2. On the **Run** file, choose **Start Simulator**: the project will start running in a separate window.
3. Open a web browser (see "[Supported platforms and browsers](#)" on page 588 for a list of browser compatible with PB4Web).
4. Enter the following address: `http://localhost:81`: this tells the web browser to read the web pages from the local computer and use port 81, used by default by the online simulator in PB4Web.
5. Test your project in the browser.

 **Important: If you make any changes to the project pages in PB610 Panel Builder 600 you must stop and restart the simulator.**

 **Note:** If you are using a device (for example, a smartphone) that is not the localhost where the simulator is running, you will be required to enter username and password.

## Downloading the Web project

After testing the PB4Web pages, you can download the project to the desired HMI device.

The PB4Web project is downloaded together with the PB610 Panel Builder 600 project, see ["Download to HMI device" on page 102](#) for details.

After the download process is completed, the HMI project automatically starts on the HMI device and the PB4Web project is ready to be used.

### Running PB4Web from a browser

1. Open a web browser and enter the IP address of your HMI device: the login page is displayed.
2. Enter **User Name** and **Password** and click **Sign In**: the Home page will be displayed.

See ["User management and passwords" on page 363](#) for details on how to create credentials.

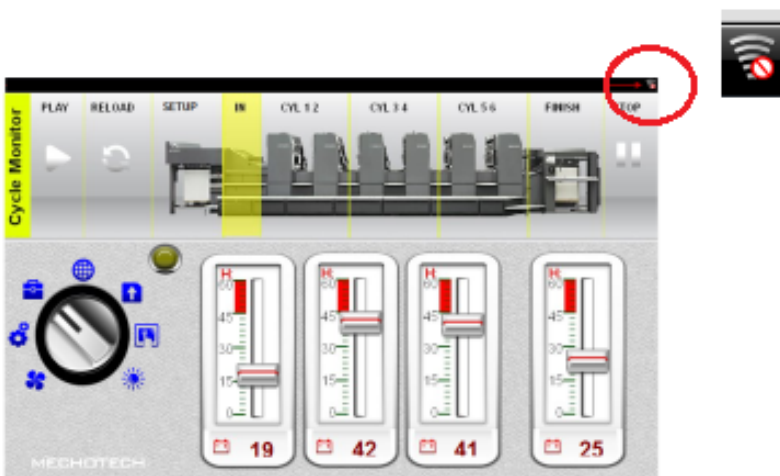
You can interact with the project using the browser in the same way you interact with a device when touching the screen: click buttons to change pages, view indicators and gauges, drag slider handles to change values, and so on. The PB4Web project will manage all communications with the web server while you are interacting with the HMI device remotely.

## Web connectivity issues

Here are described the most common issues you might encounter when connecting remotely to your HMI device.

### Server disconnection

Since PB4Web runs remotely from the HMI device, the server might disconnect from the browser (for example if the server is stopped or the network cable is unplugged). If this happens, a 'disconnect' icon will appear in a toolbar on top of the PB4Web as in this example.



Once the server is back online, the red circle-bar icon will disappear indicating normal communications with the device.



The "Connection status" system variable can be used to know the status of the connection. See ["Remote Client variables" on page 152](#) for additional details.



Note: If you make changes in the PB4Web pages while the server is disconnected, these changes will be visible on the client but will not be transferred to the server until the connection is restored.

## Inactivity timeout

PB4Web will require you to re-enter your login credentials if the browser has been inactive for several minutes. If no activity is detected for 10 minutes, the login screen will reappear and you need to enter your login credentials to continue operation. A timeout feature guarantees that no unauthorized access is possible. The web inactivity timeout can be modified from the **Project Properties** table.

## User session termination

A user session can be terminated either from the server or from the user.

In specific conditions the server might send a request to the client (browser) to perform the login process. In this case the user is redirected to the login page and then back to the page where he was working. This will happen for example if the user clears the browser cache or browser cookies.



Note: If the user is working in a dialog when redirected to the login page, he will be then redirected to the page from which the dialog was opened.

## Non-Active PB4Web Project

The PB4Web page displayed in your browser might come from a project that is no longer active in the device. In this case a confirmation box is displayed and you can return to the active project.



Note: This redirection assumes that the current active project has PB4Web pages in it.

If you choose to stay in the non-active project all the actions you perform in the browser may not be executed properly as the PB4Web cannot perform any server-bound communication.

# Web supported features

Some features or widget's properties are not supported by PB4Web. When not supported widgets are used, you will get the widgets only on the pages in the HMI panel, while on the web pages the unsupported widgets will be not visible. Note that you can run the **Project Validator** to check if pages contains unsupported widgets (see "[Project Validator](#)" on [page 70](#))

## List of widgets that are not supported

- Analog Clocks
- Analog Video
- BACnet
- Consumption meter
- Control List
- DateTime Combo (Date or Time can be set using other widgets)
- IP Widget (The IP can be configured via system settings with a browser)
- IPCamera (Supported for Chrome and Firefox. See "[Web Browser](#)" on [page 490](#) for additional details)

- Media Player
- Multistate image multilayer (Multistate image widget can be used)
- Rotation menu widget
- RSS Feed
- RSS Scroll
- Scheduler: only viewing is supported
- Text Editor
- Web Browser
- Hyper Link
- TabBar and ToolBar
- Scatter Chart

## List of actions that are not supported

Widget	SlideWidget, BeginDataEntry, TriggerIPCamera, MoveIPCamera, ContextMenu, ReplaceMedia, OpenComboBox, CloseComboBox, ShiftTableDataSrcColumns, ResetTableDataSrcColumns, remapColumns, clearRemapping
Web Browser	All actions are not supported
Text Editor	All actions are not supported
MediaPlayer (Unsupported)	All actions are not supported
Mail	All actions are not supported
FTP	All actions are not supported
Keypad	All actions are not supported
Page	LaunchApplication, LaunchBrowser, LaunchVNC, LaunchPDFViewer, LaunchUpdater, LockScreen, LoadProject, LastVisitedProject
Print	All actions are not supported
Tag	ActivateGroup, DeactivateGroup, EnableNode, BACnetClearPriority, BACnetClearAllPriorities, BACnetSetPriority, ClearRetentiveMemory, ForceReadTag
Trend/Graph	ConsumptionMeterPageScroll, ShiftTableDataSrcColumns, ResetTableDataSrcColumns, SetTableSortingColumn, ChartCommand
System	Restart, ResetProtoErrCount, SafelyRemoveMedia, SaveEventArchive, LogMessage, DumpeventArchive
UserManagement	SwitchUser, ResetPassword, AddUser, DeleteUser, EditUsers, DeleteDynamicFiles, ExportUsers, ImportUsers
RemoteClient	All actions are not supported

## List of features that are not supported

- Context menu
- Buzzer on touch
- Javascript debugger
- Wheel actions (Browser use wheel events to manage scroll bars)
- Combo box full-screen mode (Standard "context" mode is supported)
- Keypads
- ScreenSaver
- Display Rotation
- Electronic Signature

### System Variables

Using the "Attach To", only the system variables listed below are supported, while all system variables are supported using the protocol "System Variables"

- System Time
- X Screen resolution
- Y Screen resolution
- This Client Group-Name
- This Client User-Name
- Connection status
- This Client ID
- Available System Memory
- Current Language Id
- Current Language Name
- Current Language Code

### Allarms

- Alarm color based on trigger condition is not supported in Web
- Can not edit the Alarm widgets in runtime
- On Smartphone/Tablet (in general embedded devices) based on HW a user could expect performance problems with > 500 alarms.
- Page actions are not supported in alarm trigger condition

### Others

- The dialog pages support only modal dialogs.
- Some specific widgets properties are not yet supported, in this case, the default value is used. You can use the Project Validator to check if the used widgets contain properties that are not supported (see "[Project Validator](#)" on [page 70](#)).

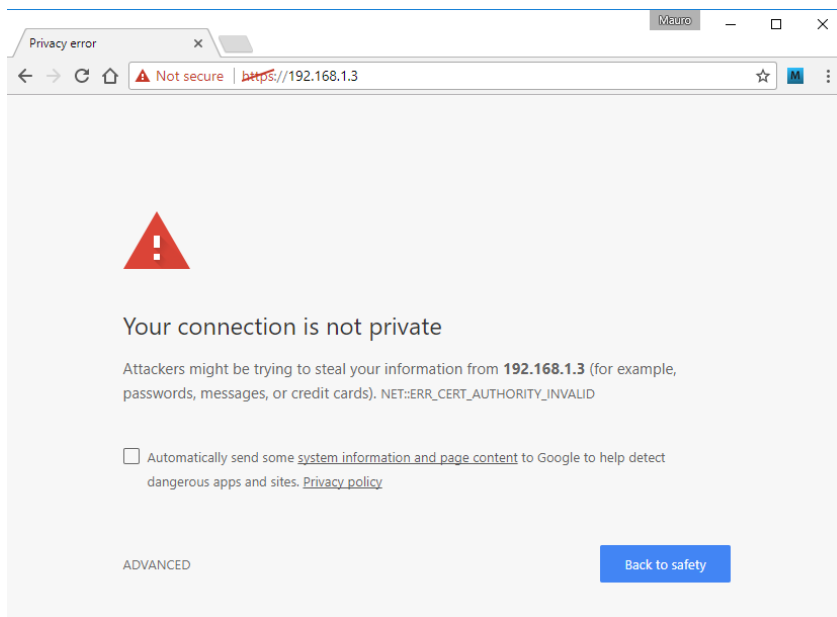
## Secure Socket Layer (HTTPS)

Linux devices support the Transfer Protocol over Secure Socket Layer (HTTPS). To use this protocol access at the web page using the below syntax:

```
https://<device_ip_address>
```

Note that since the self-certificate provided from the HMI device is not signed from a known Authority, you will get a warning message.

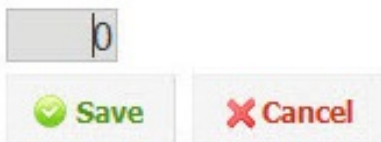
Simply click the ADVANCE button to continue.



## Working with keypads in PB4Web

The user can click on the Numeric widget and a text box will be displayed in which the new value can be inserted.

After inserting the value the user can either press **Enter**, or equivalent in touch devices, or click **Save** to make the newly inserted value permanent. Only meaningful numbers will be accepted during the save process. Anything else will be ignored and will not result in a value change.



## Troubleshooting and FAQ

### Enable JavaScript

PB4Web requires JavaScript to provide interactivity with the server and the user. PB4Web will not work if JavaScript is disabled in your browser.


By default most browsers come with JavaScript enabled. But if you have disabled JavaScript in the past, please re-enable JavaScript before accessing PB4Web pages.

### Browser cache

PB4Web includes resources that change infrequently such as CSS files, image files and JavaScript files. These resources take time to download over the network which increases the time required to load the PB4Web page in your

browser. Browser caching allows these resources to be saved by a browser and used without requesting them each time from the server. This results in faster loading of PB4Web pages.

Caching is normally enabled by default, for optimal PB4Web performance make sure it has not been disabled.

 Note: PB4Web pages will still work properly with disabled browser caching, however resource loading time will be slower compared with normal cached operations.

## Using a proxy

Some users may be accessing the PB4Web project through a proxy. The proxies may control the number of parallel connection for the browser.

Make sure that the maximum parallel connections allowed (max connections) is not more than 16 and not less than 12.

## Why I'm not able to see changes in the web pages?

Every time a new web page is added edited into the project, you need to download the project to the device. However, when you connect the device IP address, the web browser might display cached pages instead of the latest downloaded pages. To avoid this behavior you can:

- disable cache of your web browser
- force web page refresh
- by-pass browser cache

## Privacy

We do not use cookies to collect private information from any user.

A cookie is a piece of data stored on the user's hard drive containing information about the user. Usage of a cookie is in no way linked to any personally identifiable information while on our device. Once the user closes their browser, the cookie simply terminates.

# 44 Protecting access to HMI devices

---

The following operations are password protected on the HMI device:

- HMI Runtime management: install HMI Runtime and update HMI Runtime
- Board management: replace main BSP components such as Main OS, Configuration OS, Bootloader, and so on
- Download and upload of project files
- Optional services on Linux devices (e.g. SSH Protocol, VNC Server)



## **WARNING: For security reasons**

- Change the default passwords (See: ["Password protection" on page 641](#) for HMI devices on Linux platform, ["Password protection" on page 653](#) for HMI devices on WinCE platform, ["Changing password on HMI device" on the next page](#) for HMI devices on Win32 platform)
- Enable security management (See: ["Enable/disable security management" on page 364](#))
- Force remote login (See: ["Force remote login" on page 374](#))
- Configure the Context Menu to be accessible only using a "macro". In this way, it is possible to configure the use of the macro to authorized users only. (See ["Runtime" on page 80](#))



## **WARNING: Unauthorized access to the device can cause damage or malfunctions. When connecting the device to a network protect the network against unauthorized access.**

Measures for protecting the network include:

- Firewall
- Intrusion Prevention System (IPS)
- Network segmentation
- Virtual LAN (VLAN)
- Virtual Private Network (VPN)
- Security at physical access level (Port Security).

Further information, guidelines and standards regarding security in information technology: IEC 62443, ISO/IEC 27001.

---

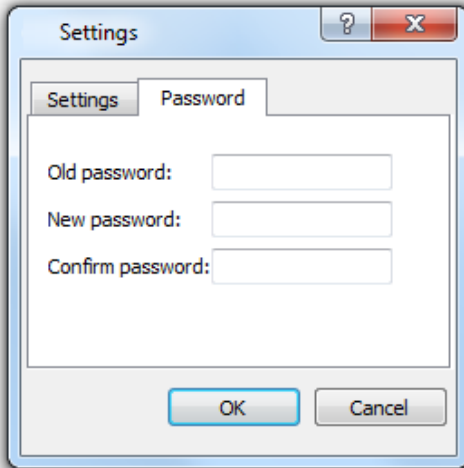
<b>Changing password on HMI device</b> .....	<b>598</b>
<b>Ports and firewalls</b> .....	<b>598</b>
<b>Project Files Encryption</b> .....	<b>599</b>
<b>Project Signature</b> .....	<b>602</b>



## Changing password on HMI device

To change the password on the HMI device, use one of the following methods:

- From the HMI Runtime context menu: **Settings**> **Password** tab.



- Use the **Set Target Password** function in update package: the password is updated by HMI Runtime just after the update process is completed.
- Using HMI device "[System Settings](#)" on [page 622](#) Tool



Leave "Old password" empty as default if target password is not set.



For Windows HMI Runtime, password is saved into:  
*Users\[username]\AppData\Roaming\ABB\buildNumber\server\config\RemoteUpdateConfig.xml.*

## Ports and firewalls

### Ports used for the main operations

Port		Where is used
443/tcp	HTTPs	Project management, System Settings Remote access (Remote Client, Web Browser)
80/tcp	HTML	Old port deprecated in favor that HTTPs
990/tcp	FTP	Project management, System Settings Remote access (Remote Client, Web Browser)
21/tcp	FTP	Old port deprecated in favor that FTPs

Port		Where is used
8000/tcp	HTML	Used to redirect HTTP requests to HTTPs protocol
18756-18759/tcp	FTP	FTP data port (passive mode)
990-991/udp		UDP broadcast (Device discovery)
998-999/udp		
2100/tcp		Manage Target
5100/tcp		JS Remote Debugger

#### Ports used for the optional services

Port		Where is used
5900/tcp		VNC Server (Can be configured to use ssl/tls)
48010/tcp		OPC UA Server
25/tcp		SMTP Server
N/A		MQTT (See your MQTT Broker)
22/tcp	SSH	Terminal Server
123/udp		NTP service
N/A		Ethernet communication drivers, the ports depend on Protocol configuration.



Note: When broadcast service is not available, for example in VPN networks, type in the exact IP address to connect to the HMI device from PB610 Panel Builder 600.

## Project Files Encryption

A project can be encrypted to secure intellectual property and not be readable or editable by unauthorized users.



**When you use a password to encrypt a project you must be aware that if you lose the password there will be no way to recover the project (you can only delete it).**



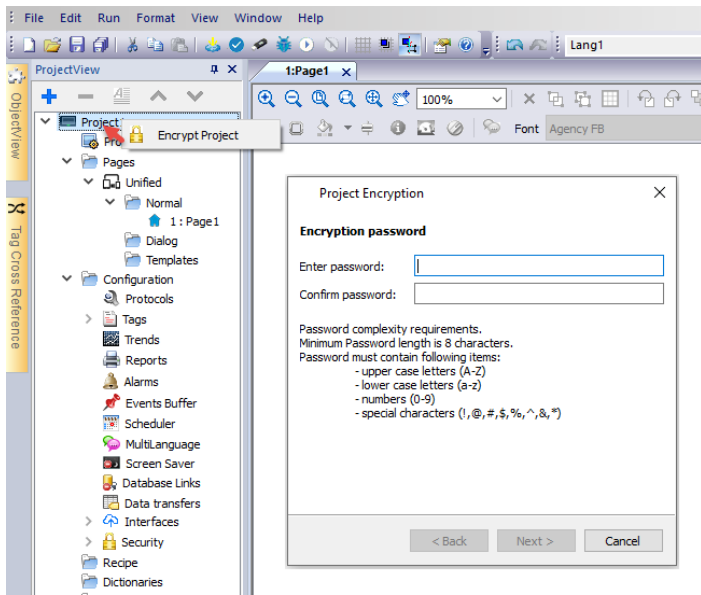
**Available only on Linux devices (See: ["HMI devices capabilities" on page 613](#))**

### Encrypt the project

Path: **ProjectView**> right-click **"Project Name"**> **Encrypt Project**

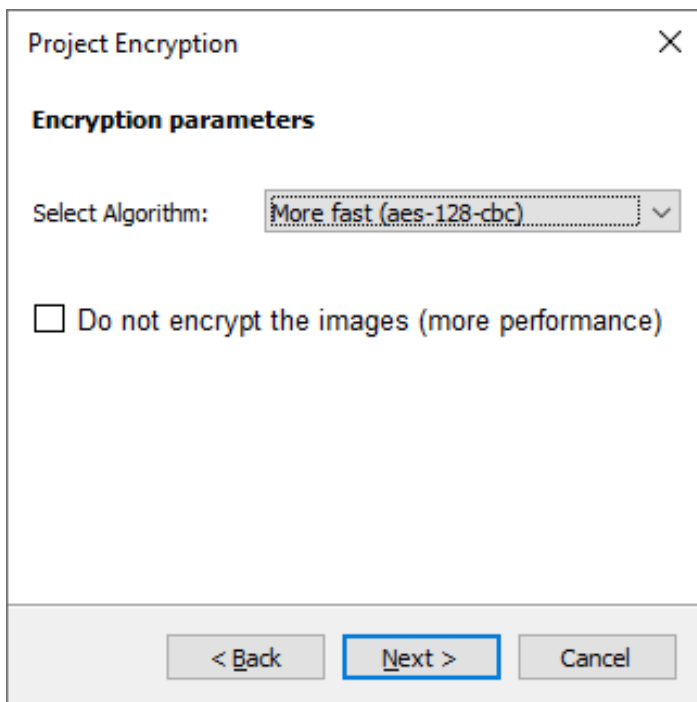


If the project is already encrypted, the same command will ask to enter the password to decrypt the project.



In the next dialog, you can select the encryption level which can be the Asymmetric Encryption "AES-128 bit CBC" or "AES-256 bit CBC".

If it is not necessary, you can also choose not to encrypt the images thus allowing the HMI device to be more performing.



When the project is encrypted, every time you open the project on PB610 Panel Builder 600 you will be asked to enter the password.

## HMI Runtime

When the HMI Runtime detects that a project is encrypted and does not know the password to decrypt the project, will show a dialog where to enter the password. The password will be requested only once and then stored in a secure area of the HMI device.



## Protected area containing passwords

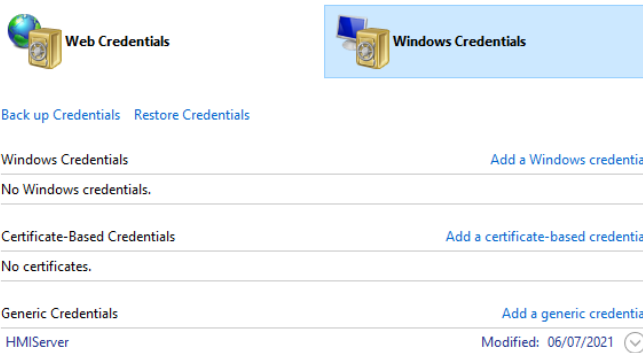
The password is stored in a secure area of the HMI device. It is possible to access this area to enter the password in advance to avoid it being requested when the application runs for the first time

### PB610 PC Runtime

The password used from the PB610 PC Runtime is accessible using Windows Credential Manager application under the "HMI Server" item inside the "Windows Credential" tab .

#### Manage your credentials

View and delete your saved logon information for websites, connected applications and networks.



### HMI Runtime

The password used from the HMI Runtime is accessible from the System Settings of the HMI device in the Security area (Ref. "Security" on page 628).

Within the Security area, it is stored using the following parameters:

- Domain = HMI Runtime
- Secret ID = Project Encryption
- Type = Password
- Secret Info = Type here the password necessary to read the encrypted project.

Credentials				
Domain	Secret ID	Type	Secret Info	Description
HMI Runtime	Project Encryption	Password	*****	Password to encrypt the project

## Project Signature

The HMI device can be configured to accept only signed projects. The signature makes sure that only authorized users can update the HMI Runtime application.



**Available only on Linux devices (See: ["HMI devices capabilities" on page 613](#))**

To configure the HMI device to accept only signed projects, an x.509 certificate is required to sign the projects.

The x.509 certificate consists of two parts:

1. **certificate.pfx**

A file with the primary key necessary to sign the project that must be installed on the PC and used from PB610 Panel Builder 600 to sign the project to download on the HMI device (the primary key is a reserved file because whoever owns it has the possibility to modify the project on the panel)

2. **certificate.crt**

A file with the public key that must be loaded on the HMI device to give the device the possibility to check if the project is correctly signed (this file will be saved in a protected area of the HMI device because if replaced, the protection would be lost)

We use Secure Hash Algorithm (SHA256)

### How to install the certificate on the PC

To install the certificate on the PC, double-click on the *certificate.pfx* file to activate the Windows installation wizard. You will be prompted for the password associated with the certificate and where to install it (for example it could be installed on the "Personal" folder)

To remove the certificate from the PC, open the Windows Credential Manager and remove the "HMIServer/prjsign" item.

### How to install the certificate on the HMI device

On the HMI device the *certificate.crt* can be installed from the System Settings of the HMI device in the Security area (Ref. ["Security" on page 628](#)).

In the Security area, select:

- Domain = HMI Runtime
- Secret ID = Project Signature
- Type = Certificate
- Use the "Update" button to load the certificate

Credentials				
Domain	Secret ID	Type	Secret Info	Description
HMI Runtime	Project Signature	Certificate		Certificate to verify the signature of the project

## How to install the certificate on the PB610 PC Runtime

You can select the certificate to use from the context menu (Ref.: ["Context menu options" on page 10](#))



Note that the certificate to use must be installed on your PC (you can double-click the certificate file to install it)

## How to configure PB610 Panel Builder 600 to sign the project before downloading it

After installing the two files relating to the certificate, it is possible to sign the application that will be downloaded to the panel by setting the "Sign Project" property, available in the "Project Properties", to true (see ["Project" on page 87](#)). When you will download a project on the HMI device, you will be prompted for the certificate to use which must correspond to the certificate installed on the HMI device.

## Script to generate a Certificate

Here is an example of how to generate a certificate using a public OpenSSL-Win32 library (Reference: <https://www.openssl.org/>)

File: *CreateCertificates.cmd*

```
@echo off
set OpenSSL="C:\Program Files (x86)\OpenSSL-Win32\bin\openssl.exe"
set CertificateName=MyCertificate

rem Generate an RSA key
%OpenSSL% genrsa -out certificate.key 4096

rem Creating Certificate Signing Requests
%OpenSSL% req -new -sha256 -key certificate.key -out certificate.csr -subj
"/ST=NY/C=US/L=New York/O=CompanyName/OU=R&D Team/CN=%CertificateName%"

rem Self Sign the Certificate Signing Requests
%OpenSSL% x509 -req -days 365 -in certificate.csr -signkey certificate.key -out
certificate.crt

rem Convert to .pfx file
%OpenSSL% pkcs12 -export -out certificate.pfx -inkey certificate.key -in
certificate.crt -CSP "Microsoft Enhanced RSA and AES Cryptographic Provider"

pause
```



The procedure will require the creation of a password which will then be required to access the primary key of the certificate.



# 45 Tips and tricks to improve performance

---

PB610 Panel Builder 600 allows great flexibility for a project designers.

Follow these guidelines to create projects that perform better in terms of boot time, page change and animations.

---

<b>Static Optimization</b> .....	<b>606</b>
<b>FAQ on Static Optimization</b> .....	<b>609</b>
<b>Page caching</b> .....	<b>609</b>
<b>Image DB</b> .....	<b>609</b>
<b>Precaching</b> .....	<b>610</b>
<b>FAQ on precaching</b> .....	<b>610</b>




# Static Optimization

Static optimization is a technique used in PB610 Panel Builder 600 to improve runtime performance.

Using a lot of images and pictures in a project might degrade performances, static optimization merges several images into a single background image thus reducing rendering and loading times. Using this method only one raster image needs to be loaded and rendered instead of many single raster and/or vector images.

When you create a project in PB610 Panel Builder 600, the pages might contain widgets such as texts, images, background images, background colors and so on which can be classified as:

- **Static:** values or properties do not change at runtime.
- **Dynamic:** values or properties change at runtime.

 Note: Based on security settings, static parts of widgets could be not merged to background. This happens when a widget is configured as “hide” in security settings.

 **Important: When you change the properties of widgets with JavaScript set the widget Static Optimization to Dynamic, otherwise changes to properties will be ignored.**

When downloading or validating a project, PB610 Panel Builder 600 identifies static components and renders them as background images to .png files. These background images are saved as a part of the project under the folder `/opt`.


Background images can be created as follows:

- full page background images, containing all widgets merged to page background
- group background images, containing a group of static widgets merged together to form a group background. For example, the Gauge group is normally composed by a background, a scale, a label and a needle, where background scale and label can all be merged to a single background image.

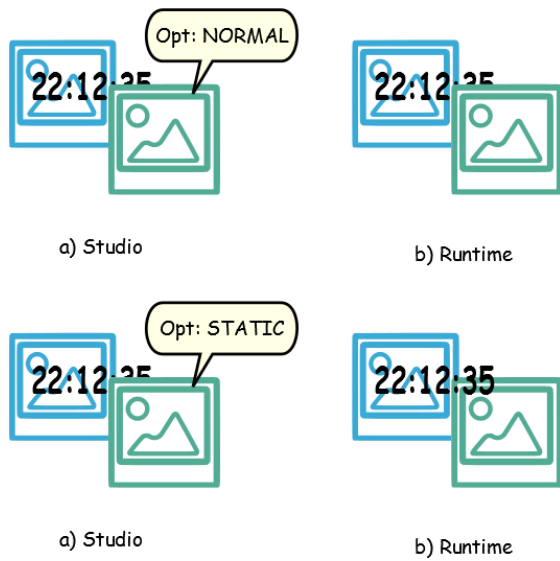
The **Static Optimization** page attribute enables and disables static optimization of the whole page. If it is set to **false** the optimization is totally disabled.

Finer control can be achieved setting the **Static Optimization** attribute of each single widget as follows:

- **Normal:** PB610 Panel Builder 600 automatically detects if the widget can be merged with the background. This can be used if the widget is not a dynamic widget and does not overlap, that is it is not stacked above, a dynamic widget.
- **Static:** The image is forced to be merged with the background. This can be used when the static widget overlaps a dynamic transparent widget.

 Note: In this case the automatic optimization will fail because it does not make any assumption on invisible areas which might be rendered at runtime.

- **Dynamic:** The widget is not optimized at all. Use this flag when a static widget needs to be changed by JavaScript.



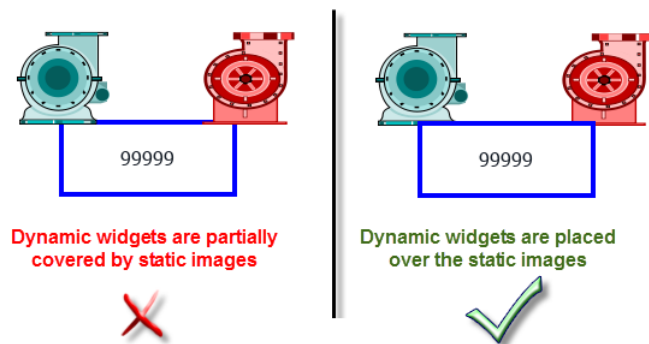
## Tips for best performance

1. First of all: avoid placing static widgets over a dynamic widget. The overlapping area is computed considering the bounding rectangles of the widgets, that is the rectangles delimited by editing handles.
2. Do not use static optimization if your pages contain almost only dynamic objects. Static optimization would save many almost identical full size images for each page using up a lot of memory space that could be more effectively used to improve project performance with other techniques (such as, for example, page caching).
3. Bounding rectangles can include transparent areas, minimize transparent areas (for example splitting the image in multiple images) since they can be a waste of resources even when optimized.
4. Optimize image size. The image will be rendered at the size of the image widget containing the image. For best performances the widget needs to be the same size of the image.
5. Avoid using **Scale to fit** for image widgets, since this forces a rescaling at runtime for dynamic images and “hides” the actual image size during editing.
6. Use **Size to fit** to make the widget to the real size of his contents.
7. If overlapping cannot be avoided make sure to place the static widgets in the back, that is behind the dynamic widget.
8. Choose the image file format based on the HMI device you are connecting to.
9. Avoid using too many widgets in a single page. Often widgets are placed outside the visible area or their transparency is controlled by a tag. Since widgets are loaded even if they are not visible, having too many widgets in a page can significantly slow down the page change time.
10. Split a page with many widgets into multiple pages with less widgets.
11. For popping up new graphic elements in a page, prefer dialog pages with controlled positioning to transparent widgets.
12. Check the *opt* folder to see if static optimization is working as expected, the widgets z-order might need to be adjusted.
13. Numeric fields are often used to run JavaScript code on *OnDataUpdate* event even if the widget doesn't need to be visible on the page. In this case place the widget outside the page visible area instead of making it invisible, altering font color or visibility property. In the latter case you might end up with many left over wedges.
14. Use a HotSpot button if you need a touch area to react to user inputs.

15. If you reuse a widget from the gallery or you create your own, remember to set the correct optimization properties. For example button widgets are dynamic widgets, if you use a button widget just for its frame it won't be optimized since the button widget is dynamic. If you just need the frame you should use the Up image.
16. With many pages having many dynamic widgets and using a common template:
  1. set template static optimization to **true**,
  2. set page static optimization to **false**, since the background is already provided by the template.

In this scenario the background image can be reused by many different pages thus saving memory space.
17. Do not use dynamic widgets, such as buttons, only for graphic purposes, when the button function is not needed, use image widgets instead to obtain the same graphical effect.

Here is an example of a correct and an incorrect use of static optimization.



## Supported image formats

PB610 Panel Builder 600 supports several raster formats like BMP, PNG, JPEG, TIFF and the vector format SVG. Here a list of pros and cons:

Image format	Pros	Cons
RASTER	<ul style="list-style-type: none"> <li>• Fast rendering</li> <li>• Well standardized</li> </ul>	<ul style="list-style-type: none"> <li>• Big file size</li> <li>• Fixed resolution</li> </ul>
VECTOR (SVG)	<ul style="list-style-type: none"> <li>• Small file size</li> <li>• Rescale without quality loss</li> <li>• Can handle dynamic properties</li> </ul>	<ul style="list-style-type: none"> <li>• Complex SVG images with many graphic items and layers can be slow to render.</li> <li>• Creating an optimized SVG is not simple.</li> <li>• Only Tiny 1.2 (<a href="http://www.w3.org/TR/SVGTiny12/">http://www.w3.org/TR/SVGTiny12/</a>) supported.</li> </ul>



Note: Scour software is free tool that can be used to remove foreign code from file (<http://www.codedread.com/scour/>).

## Static optimization of templates

Template pages can have large amounts of static content. However, static optimization cannot be applied to a template page, since where the template is used is based on the page design.

If a huge background image should be repeated in every page that uses the same template, this would increase the footprint of the device as the same static image would be created for each of the pages using the template page.

## FAQ on Static Optimization

**Q: In a page where there are a few identical widgets, in the *opt* folder I see a PNG for each one of them. If they are really identical, why should the software duplicate them instead of having just one PNG?**

A: The software does not know if static images are actually the same since each widget could have different settings/properties altering the actual rendering at runtime.

**Q: Why are the static images stored in a separate folder called *opt* instead of storing them directly in the project folder?**

A: This avoids name collisions and allows skipping the upload of optimization images

**Q: Why are the static images stored as a PNG files instead of common JPEG files?**

A: PNG format uses a lossless compression for images and supports transparencies. JPEG files would render fuzzier compared to the PNG files with a different result in PB610 Panel Builder 600(not using optimization) and HMI Runtime.

**Q: What will happen when no optimization is done in the software?**

A: Every single widget is rendered at runtime. In particular SVG images may require a lot of time to render in an embedded platform.

## Page caching

Once accessed all pages are kept in a RAM cache up to the maximum allowed cache size depending on the actual platform's available RAM. This allows a much faster access since cached pages, once reloaded, only need to re-paint their content without reloading all page resources.

## Image DB

Image DB is a technique used to track the usage of image files and reduce the cost of image loading by caching most frequently used images (example, Push Button images, Gauge needles, Slider thumbs and so on). The same image used in many different places is therefore loaded just once.

The image DB function will preload the top most used images at startup until memory limit is reached. This would further improve the individual page loading times.

The file `imagecachelist.xml` is created in `project/opt` folder, containing relevant information:

- Fill color (in case of SVG images)
- Size of SVG image

- Number of times an image is used in the project
- Number of different sizes for the same image

## Tips for using the Image DB function

1. Use uniform size of buttons, gauges and other widgets wherever possible.
2. Use same color themes among widgets of the same kind.

# Precaching

The Precache attribute of pages can be used to notify HMI Runtime to preload some pages in RAM at boot time for quicker access. Precaching is useful for complex pages having many dynamic widgets.

When this function is enabled on a page, access to the page is faster, however it also slows down boot-time since the system is not ready until all pages to be precached are not saved into the RAM.

## Tips to precaching

1. Enable the precache function just for few pages having many dynamic widgets or for pages frequently used by users.
2. Do not enable the precache function for all the pages in the project since you would run out of memory and have no benefit at all.
3. Disable static optimization for pages where the precache function is enabled to reduce memory used.

# FAQ on precaching

## Page limit for precaching

Based on the size and complexity of a page, the space required for precaching can be from 1,5Mb to 3Mb.

When a project is loaded, HMI Runtime proceeds as follows:

1. Page images are preloaded until 76 MB of memory space is still available (imageDBLowMem)
2. Pages where precache is set to **true** are preloaded until 64 MB of memory space is still available (pageCacheLowMemMax). The images of these pages are loaded in the RAM (into the Image DB).

When the project is ready:

1. Any new page visited is saved in the cache (RAM) with all related images until 40 MB of memory space is still available (pageCacheLowMemMin)
2. When a page change happens and space in RAM is critical (<40MB), the HMI Runtime starts emptying the cache (RAM) removing pages and related images until 64 MB of memory space is made available. HMI Runtime removes data stored in the cache in the following order:
  1. last visited pages and bigger and unused images (>320x240),
  2. if more memory is needed also the pages in precache and all images loaded in Image DB can be removed.

# 46 Functional specifications and compatibility

---

Here is an overview of the supported functions and related limitations. Limitations indicated here represent a safe limitation, beyond that proper operation and state-of-the-art performance of the system is not guaranteed.

---

<b>Table of functions and limits</b> .....	<b>612</b>
<b>HMI devices capabilities</b> .....	<b>613</b>
<b>Compatibility</b> .....	<b>614</b>
<b>Converting projects between different HMI devices</b> .....	<b>615</b>

## Table of functions and limits

Function	Max limit
Number of pages	1000
Number of basic widgets	2000 x page
Number of tags	10000
Number of dialog pages	See <a href="#">"HMI devices capabilities" on the facing page</a>
Number of dialog pages that can be open at the same time	5
Number of Recipes	32
Number of parameter sets for a recipe	1000
Number of elements per Recipe	1000
Number of user groups	50
Number of users	500
Number of concurrent remote clients	4
Number of schedulers	30
Number of alarms	See <a href="#">"HMI devices capabilities" on the facing page</a>
Number of data transfers	1000
Number of templates pages	50
Number of actions programmable per button state	32
Number of trend buffers	30
Number of tags per trend buffer	See <a href="#">"HMI devices capabilities" on the facing page</a>
Memory reserved for trend buffer	See <a href="#">"HMI devices capabilities" on the facing page</a>
Number of curves per trend widget	See <a href="#">"HMI devices capabilities" on the facing page</a>
Number of curves per scatter diagram widget	10
Max number of trend table printable rows	10000 on HMI Runtime 50000 on PB610 PC Runtime
Number of messages in a message field	1024
Number of languages	24 HMI Devices based on WCE platform support until 12 languages (See <a href="#">"HMI</a>

Function	Max limit
	<a href="#">devices capabilities" on the facing page)</a>
Number of events per buffer	See <a href="#">"HMI devices capabilities" below</a>
Number of event buffers	4
JavaScript file size per page	See <a href="#">"HMI devices capabilities" below</a>
Size of project on disk	See <a href="#">"HMI devices capabilities" below</a>
Number of indexed instances	100
Number of indexed alias	100
Number of indexed tag sets	30
Number of physical protocols	See <a href="#">"HMI devices capabilities" below</a>
Number of reports	See <a href="#">"HMI devices capabilities" below</a>
Number of reports pages	32
Max number of variables in variables widget	255
User folder size (UpdatePackage.zip)	See <a href="#">"HMI devices capabilities" below</a>
Number of concurrent FTP sessions	4
FTP additional folders	5
MQTT max number of bytes in a payload packet	There are no limits for the MQTT payload, this is based on the broker's limit. We can use up to the tag string size limit.
MQTT max number of topic	Topics are created dynamically, there are no limits
MQTT max number of Kb that can be stored in the persistence	50000
MQTT max pending messages	10000

## HMI devices capabilities

See ["Table of functions and limits" on the previous page](#) for the standard capabilities.



Panel	Device OS	Touch	Media Player	Media Player Portrait Mode	PDF	Max Project Size	Dialogs	Alarms	Protocols	JavaScript	Reports	Trend Buffers	Max Tags inside a Trend	Curves per Trend Widget	Max Events inside a Buffer	User Folder Size
Runtime PC	Win32		MPEG4	Yes	Yes	240 MB	200	10,000	8	64 KB	64	500 MB	300	10	10 K	na
CP620	WCE		na	Yes	na	30 MB	50	500	4	16 KB	32	25 MB	200	5	2 K	10 MB
CP630	WCE		na	Yes	na	30 MB	50	500	4	16 KB	32	25 MB	200	5	2 K	10 MB
CP635	WCE		na	Yes	na	30 MB	50	500	4	16 KB	32	25 MB	200	5	2 K	10 MB
CP651	WCE		MPEG4/H264	Yes	Yes	60 MB	50	2,000	4	16 KB	32	25 MB	200	5	2 K	100 MB
CP661	WCE		MPEG4/H264	Yes	Yes	60 MB	50	2,000	4	16 KB	32	25 MB	200	5	2 K	100 MB
CP665	WCE		MPEG4/H264	Yes	Yes	60 MB	50	2,000	4	16 KB	32	25 MB	200	5	2 K	100 MB
CP676	WCE		MPEG4/H264	Yes	Yes	60 MB	50	2,000	4	16 KB	32	25 MB	200	5	2 K	100 MB
CP635-Fx	WCE		MPEG4/H264	Yes	Yes	60 MB	50	2,000	4	16 KB	32	25 MB	200	5	2 K	100 MB
CP604	Linux		na	Yes	Yes	60 MB	50	500	4	64 KB	32	25 MB	200	5	2 K	100 MB
CP607	Linux		na	Yes	Yes	60 MB	50	500	4	64 KB	32	25 MB	200	5	2 K	100 MB
CP610	Linux		MPEG4	Yes	Yes	240 MB	50	3,000	4	64 KB	32	25 MB	200	5	10 K	512 MB
CP6607	Linux	Multi	MPEG4/H264 (*)	No	Yes	240 MB	200	4,000	8	64 KB	64	200 MB	300	10	10 K	512 MB
CP6610	Linux	Multi	MPEG4/H264 (*)	No	Yes	240 MB	200	4,000	8	64 KB	64	200 MB	300	10	10 K	512 MB
CP6615	Linux	Multi	MPEG4/H264 (*)	No	Yes	240 MB	200	4,000	8	64 KB	64	200 MB	300	10	10 K	512 MB
CP6621	Linux	Multi	MPEG4/H264 (*)	No	Yes	240 MB	200	4,000	8	64 KB	64	200 MB	300	10	10 K	512 MB
CP6605	Linux	Multi	MPEG4	Yes	Yes	240 MB	50	3,000	4	64 KB	32	25 MB	200	5	10 K	512 MB
CP6407	Linux		MPEG4	Yes	Yes	240 MB	100	3,000	6	64 KB	48	35 MB	200	10	10 K	512 MB
CP6410	Linux		MPEG4	Yes	Yes	240 MB	100	3,000	6	64 KB	48	35 MB	200	10	10 K	512 MB
CP6415	Linux		MPEG4	Yes	Yes	240 MB	100	3,000	6	64 KB	48	35 MB	200	10	10 K	512 MB

(\*) Auto resize is supported



The acronym “WCE” indicates Microsoft Windows Embedded CE 6.0 R3

Features not available in Linux devices:

- LaunchBrowser macro
- Printer devices are not supported. Reports can be printed only on PDF files. Print of text reports and alarm events are not supported.

Features not available in PB610 PC Runtime:

- VNC and PDF Readers plug-in
- Manage Target
- System Settings Tool
- Backup/Restore
- Serial protocols that requires special hardware
- Multi touch features will be available if supported from the hardware and the operative system of the PC

## Compatibility

The following compatibility policy has been adopted:

- PB610 Panel Builder 600 version must always be aligned with HMI Runtime on the device,
- the user is responsible for updating HMI Runtime components on the HMI device at any PB610 Panel Builder 600 update,
- the HMI Runtime update can be done directly from PB610 Panel Builder 600 using the Update Target command available in the Run\Manage Target dialog,
- projects created in a PB610 Panel Builder 600 version no older than V1.00 (00) can be opened and handled by any newer version,

- projects created with older versions of PB610 Panel Builder 600, opened with later versions and deployed to compatible HMI Runtime, are ensured to maintain the performance and functionality,
- compatibility between newer versions of HMI Runtime and projects created and deployed with older versions of PB610 Panel Builder 600 is not ensured.



**Important: Do not edit projects with a version of PB610 Panel Builder 600 older than the one used to create them. It can result in a damage of the project and to HMI Runtime instability.**

## Converting projects between different HMI devices

Project conversion from different HMI device models is supported, however, some manual operations may be required if the project uses features not supported in the destination device.

### Guideline

Before converting a project have a look if some unsupported features are present (see ["HMI devices capabilities" on page 613](#)), and adjust your project by removing the unsupported features before converting the project.

In particular:

- Verify limitations and features not supported by the new HMI device (see ["Table of functions and limits" on page 612](#) for details).
- Remove unsupported widgets, actions, system variables, protocols, project properties.
- If the project uses external storage, verify if the same storage path is still available.
- Adjust OS-specific external applications or paths.
- If necessary, reduce project size according to the new HMI device type limitations (see "Limitations" for details).
- Since HMI devices are based on different hardware platforms with different CPU speed, RAM memory size, cache size, make sure to check project boot time and page loading time for each page in the project.
- Verify JavaScript code for OS-specific operations.

### OS-specific features

Linux is case sensitive while Windows CE is not. Consequently, projects on Linux HMI devices might have different files named based on upper and lower case, e.g. 'dump1.csv' and 'Dump1.csv' are not possible on Windows CE HMI devices.



# 47 System Settings (Linux Devices)

CP600-eCo products offer a powerful integrated tool called System Settings that allows management and upgrade of system components. Operations can be done directly on HMI or remotely using web browser.

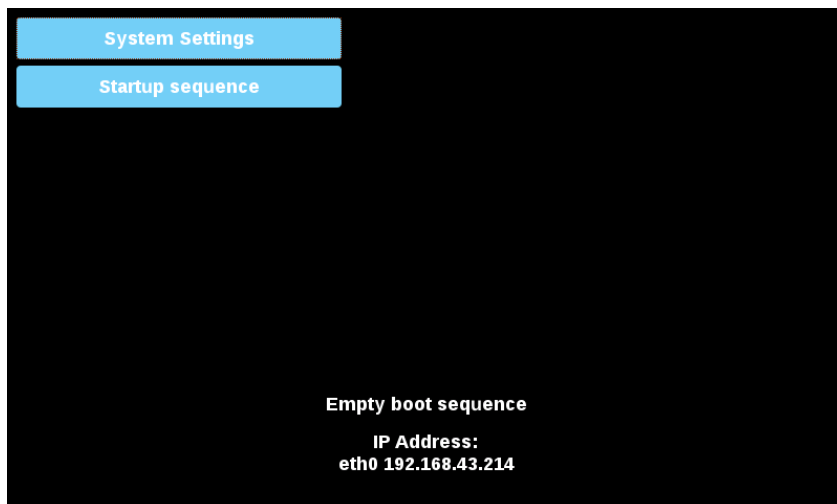


**CAUTION: Working with the System Settings tool is a critical operation and, when not performed correctly, may cause product damages requiring service of the product. Contact technical support for assistance.**

Runtime Installation .....	617
System Settings .....	622
Update System Components .....	638
Touchscreen calibration .....	640
Password protection .....	641
Backup and Restore .....	642
Recovery Mode .....	643

## Runtime Installation

If the HMI device is delivered from the factory without Runtime, at first power up HMI shows the “Runtime Loader” screen.



Runtime can be installed:

- Automatically, via Ethernet on first project download with PB610 Panel Builder 600
- Manually via USB Memory, creating an “Update Package”. (See the ["Update package" on page 104](#) to create a runtime package)

## Install Runtime via Ethernet

To install Runtime via Ethernet follow the "[Download to HMI device](#)" on page 102 procedure.



**WARNING: Runtime installation via Ethernet download requires the HMI to have a valid IP address.**

The IP address can be assigned in three ways:

- *Automatically via DHCP server.* If a DHCP server is available on the network IP address will be assigned automatically by the server.
- *Automatically via Auto-IP feature.* If DHCP assignment is enabled but no DHCP server is available on the network the HMI assigns itself an IP Address into range 169.254.x.x with subnet mask 255.255.0.0
- *Manually via System Settings.* From System Settings menu, in Network section the IP address can be manually assigned, disabling the DHCP server assignment feature.

## Install Runtime via USB Memory

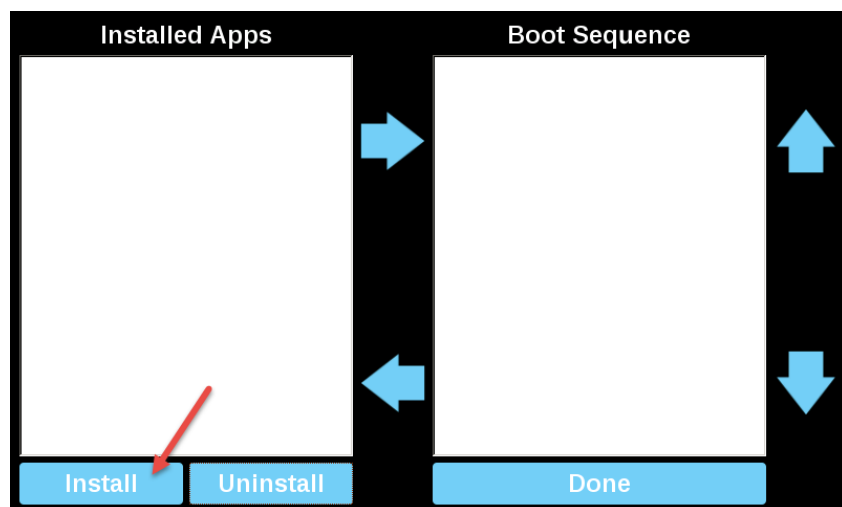
To install Runtime, UpdatePackage or Backup Package via USB device follow this procedure:

1. Create an Update Package from PB610 Panel Builder 600 and copy into an empty USB memory stick

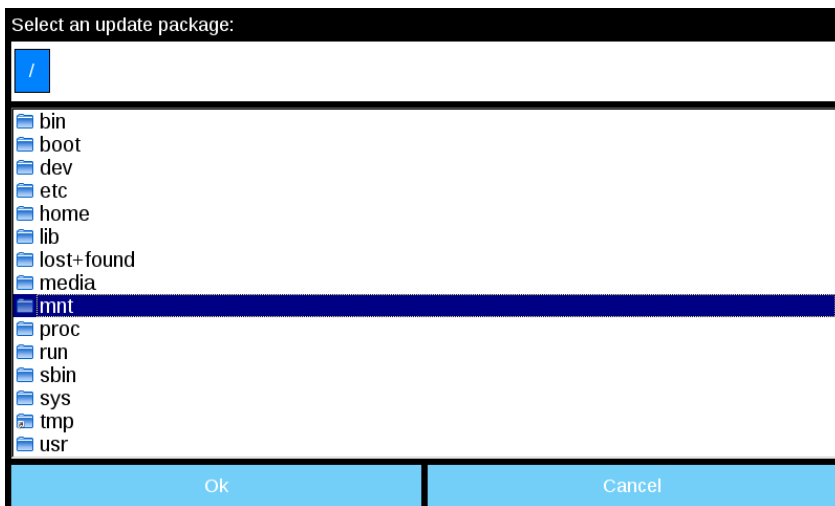


Note: File systems supported are FAT16/32 and Linux Ext2, Ext3 and Ext4.

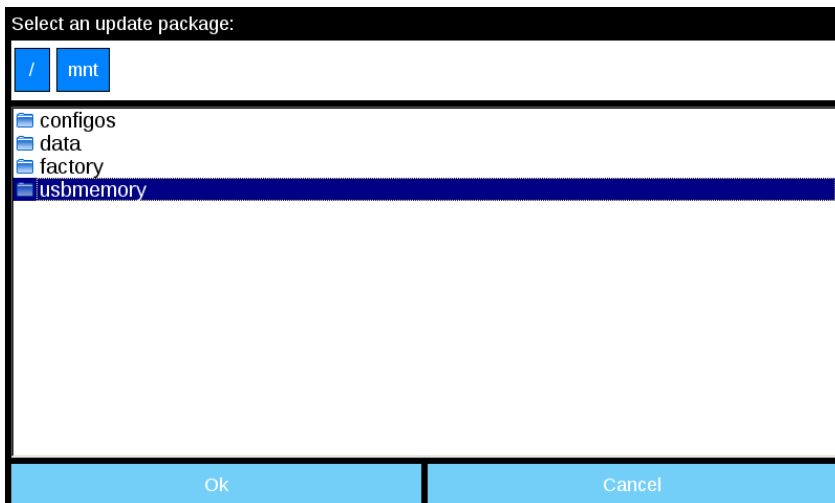
2. On HMI select [Startup sequence], then [Install]



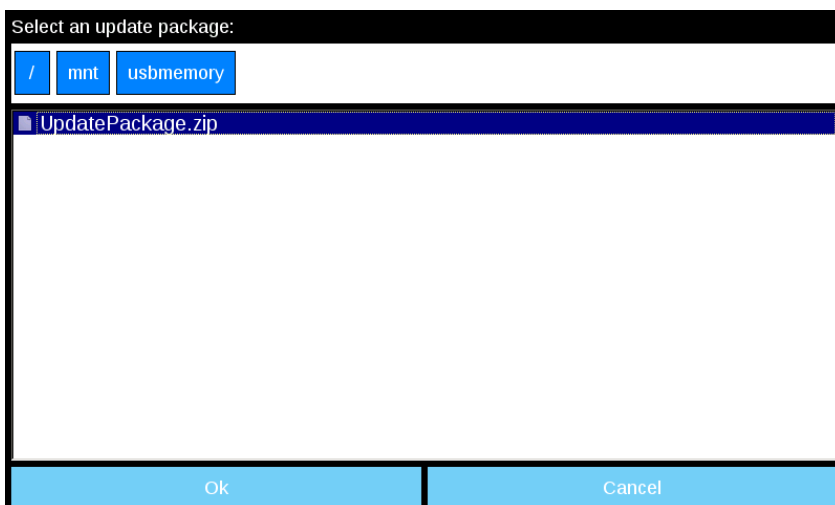
3. Double click on "mnt" to access this folder



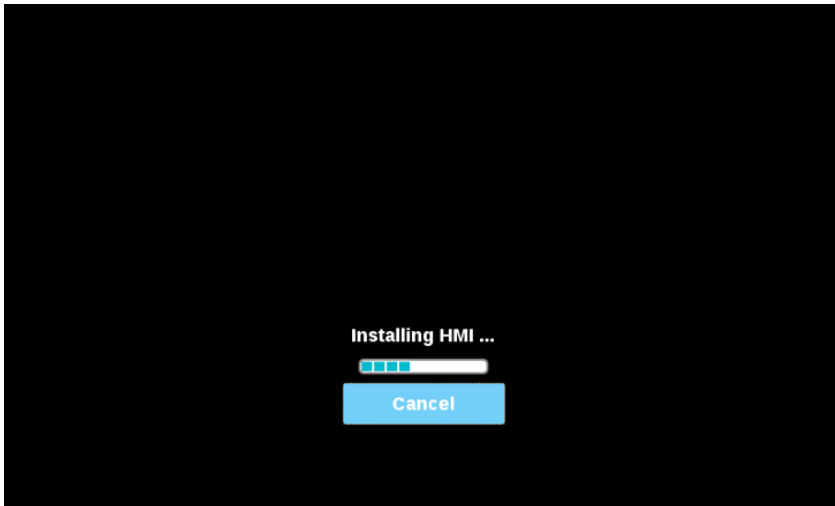
4. Then on “usbmemory”



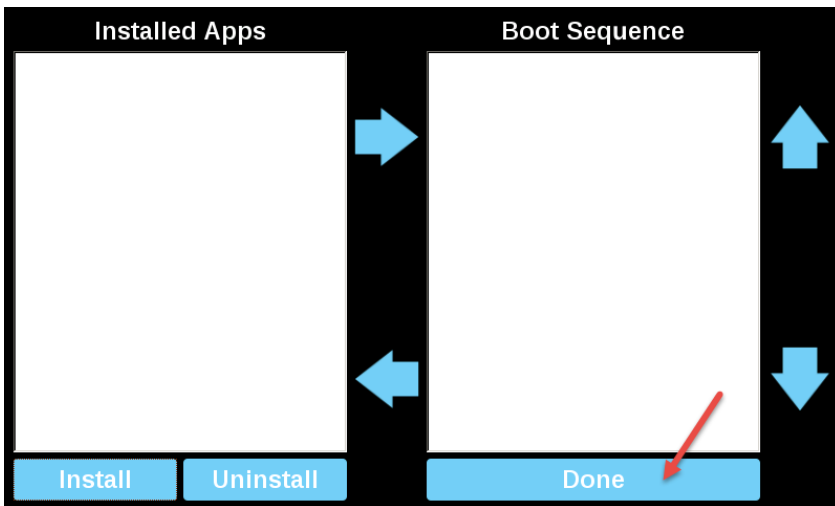
5. Select “UpdatePackage.zip” and confirm with [Ok]



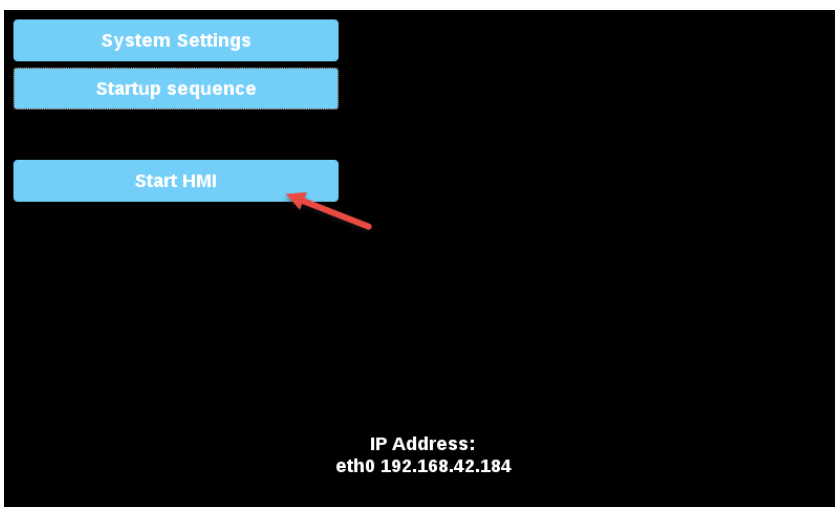
6. The runtime installation begin



7. At the end press "Done" button



8. Then "Start HMI" button

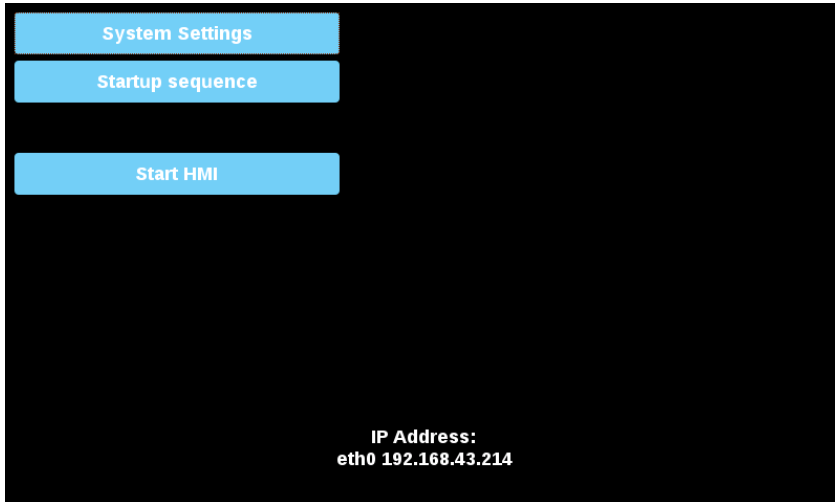


## Runtime Uninstall

System Settings in Default mode allows to uninstall HMI Runtime or change Startup sequence, this mode is available via tap-tap sequence and can be accessed also when HMI is facing a software failure.

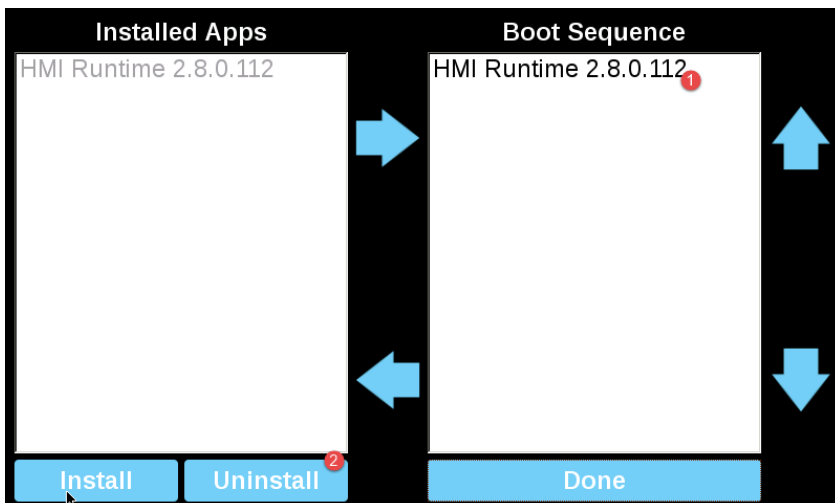
See ["Enter System Settings via tap-tap procedure "](#) on page 624

To uninstall the Runtime from HMI in Default Mode screen select [Startup Sequence]:



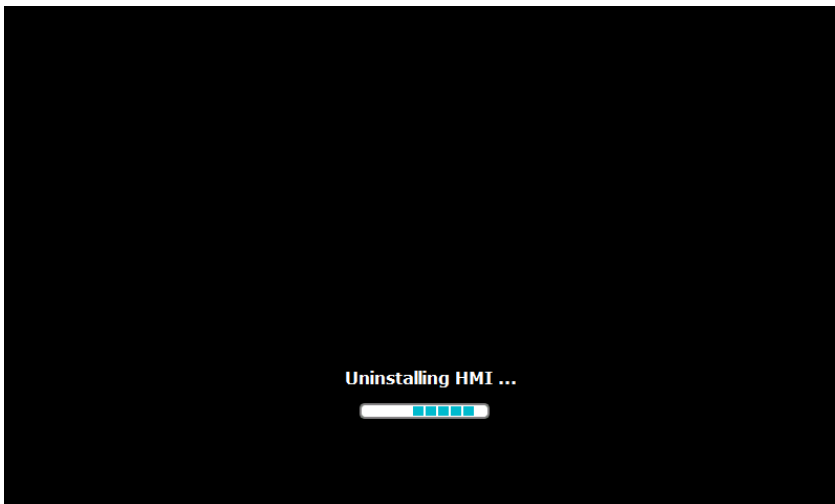
From the installed applications view:

1. Select the Runtime you want to remove
2. Uninstall the selected Runtime

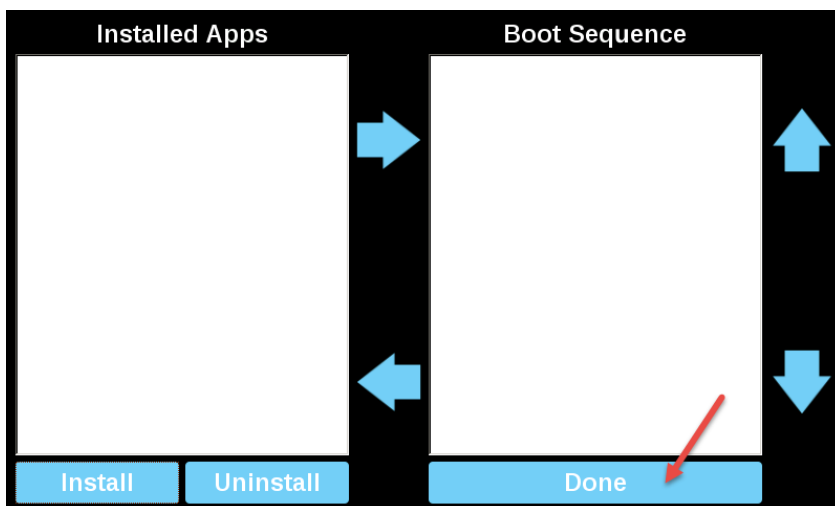


Runtime uninstall process will be performed:





At the end press "Done" button



## System Settings

The user interface of System Settings is based on HTML pages and can be accessed both locally on the HMI device screen and remotely using a Web browser.

Administrator username with full access right is "admin" with default password "admin". Generic username is "user" with default password "user"



**WARNING: For security reasons, change the default passwords for both usernames (passwords can be modified from the "System Settings -> Authentication" command)**



Accessing at the system settings from the HMI device do not require to enter a password until the default "admin" password is not changed.

### System Setting access from Web browser

To access System Settings using a Web browser, enter the IP address of the device, in the following format:

*https://IP/machine\_config*



Note the remote access use encrypted https protocol on port 443. When the connection is established, the HMI device send a certificate to use for the encryption. Since the certificate is not signed from a Certificate Authority you will get a warning message. Please, click on advanced options and choice to proceeding.

### Your connection is not private

Attackers might be trying to steal your information from **192.168.52.4** (for example, passwords, messages, or credit cards). [Learn more](#)

NET::ERR\_CERT\_AUTHORITY\_INVALID

Automatically send some [system information and page content](#) to Google to help detect dangerous apps and sites. [Privacy policy](#)

ADVANCED
Back to safety

Browse through the options available in the menu on the left: the active item is highlighted and related information is displayed on the right.

System Settings	Localisation <span style="float: right; font-size: small; color: #4285f4;">ADMIN <a href="#">↗</a></span>
Localisation	Language:
System	<input checked="" type="checkbox"/> English
Logs	<input type="checkbox"/> Italiano
Date & Time	<input type="checkbox"/> Deutsch
Network	<input type="checkbox"/> Français
Applications	<input type="checkbox"/> 简体中文
Services	<input type="checkbox"/> 繁體中文
Plugins	<input type="checkbox"/> 한국어
Management	<input type="checkbox"/> 日本語
Display	<input type="checkbox"/> Español
Fonts	<input type="checkbox"/> Português - Brasileiro
Authentication	<input type="checkbox"/> Русский
Restart	Country Code: <small>(REQUIRED for WLAN Regulatory Domain)</small>
EXIT	<input type="text" value="00 Unspecified"/> <a href="#">↗</a>
	System keyboard layout:
	<input type="text" value="English (United States)"/> <a href="#">↗</a>

Default security protocols proposed by the HTTPS server in the CP600-eCo HMI device are:

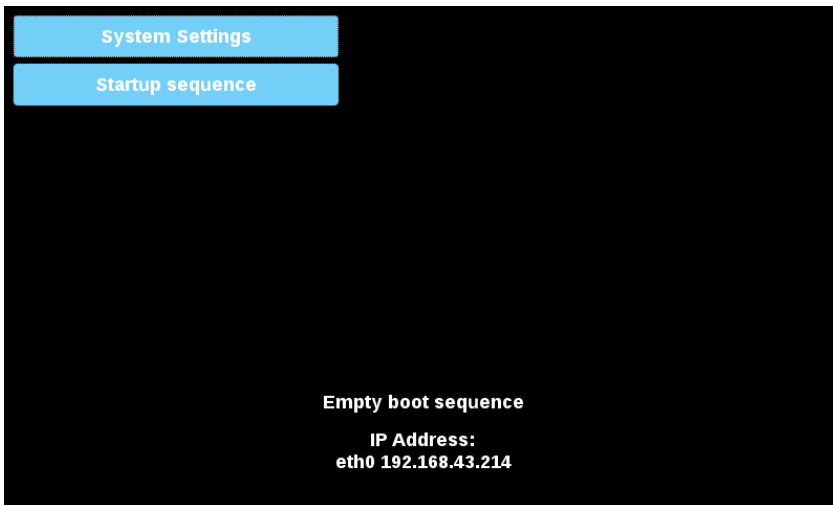
- SSLv3 256 bits ECDHE-RSA-AES256-SHA
- TLSv1 256 bits ECDHE-RSA-AES256-SHA



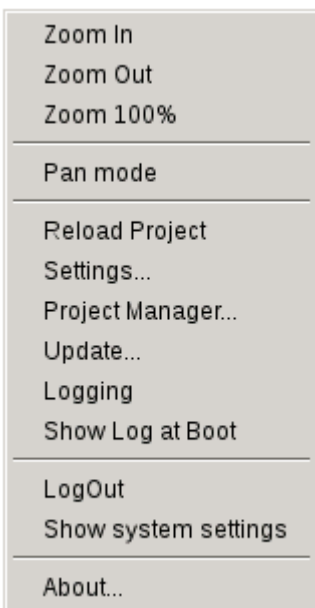
**WARNING: We discourage usage of CBC cyber suites in the context of SSL3 or TLSv1.0 connections since potentially affected by some vulnerabilities.**

## System Setting access from HMI device

When Runtime is not installed, the System Settings is accessible from the Runtime Loader screen,

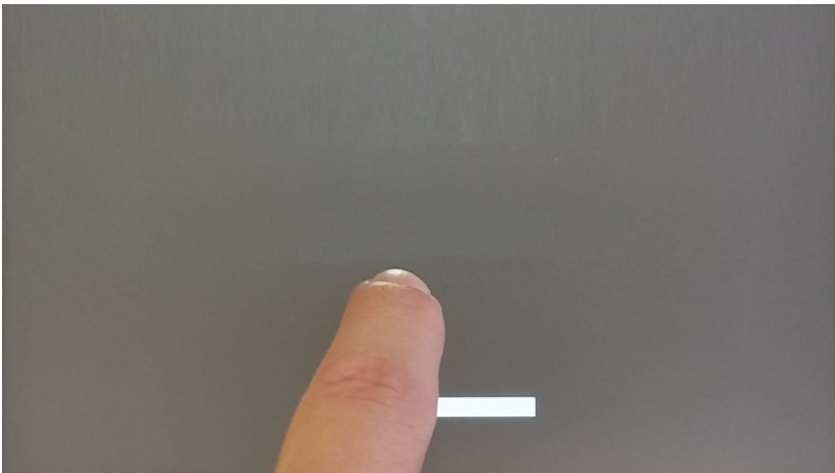


When Runtime is installed the System Settings is accessible selecting “Show System Settings” option of Context Menu,

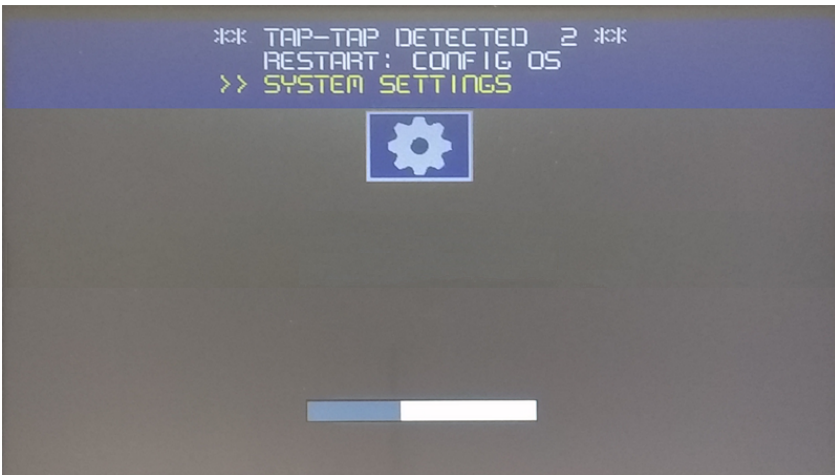


## Enter System Settings via tap-tap procedure

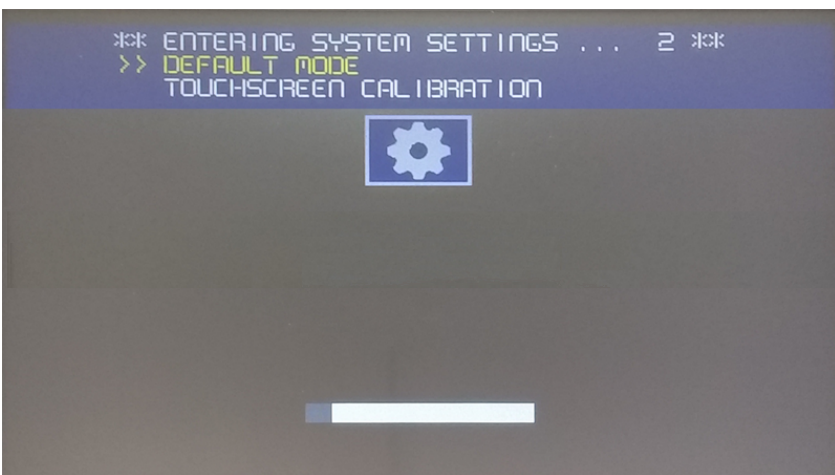
Tap-tap consists in a sequence of several touch activations by simple means of the finger tapping the touch screen performed during the power-up phase and started immediately after the HMI is powered on.



When "tap-tap detected" message appears on the top of the screen. Wait for 5 seconds (without touching the screen) to enter System Settings sub menu



Wait for 5 more seconds (without touching the screen) to enter Default Mode



Select "System Setting" from the HMI Default Mode screen



## System Settings Sections

To change system settings values, enter in edit mode by click the edit button on the right top.



The edit button is available only inside the dialogs that contains modifiable parameters.

## Localization

Set the below parameters to adapt the device to your country.

- Country Code (only needed on 5G devices)
- Language for the system settings interface
- Layout of the virtual keyboard



**Country Code is required for the WLAN Regulatory Domain and the device will not use the WiFi until this parameter will not have been set.**

The country settings are required for operation complying with the approvals. Selecting a country that does not match the country in which the device is operated may be punishable by law. After selecting the Country Code, the corresponding channels allocation and setting and for power level will be automatic.

## System

Parameter	Description
Info	Device information
Status	Device status (Free RAM, Up time, CPU Load)
Timers	Device timers (System on, Back light on)
PlugIn	Hardware plugins information

## Logs

Set the persistent log option if you want maintain the log files saved after a power reset.

Use save button to export a copy of the log files.



The log files manager cyclically fill 3 files of 4Mb

## Date & Time

Device date and time.

Parameter	Description
<b>Current Timezone</b>	Timezone region
<b>Current Date Local Time</b>	Date and Time can set manually only when the Automatic Update is disabled.
<b>Automatic Update (NTP)</b>	<p>Enable to keep date and time synchronized from a remote server</p> <ul style="list-style-type: none"> <li>NTP Server Specify the Internet NTP Server address</li> </ul> <p> The NTP Client of the HMI Device is a complete implementation of the Network Time Protocol (NTP) version 4, but also retains compatibility with version 3, as defined by RFC-1305, and version 1 and 2, as defined by RFC-1059 and RFC-1119, respectively.</p> <p>The poll process sends NTP packets at intervals determined by the clock discipline algorithm. The process is designed to provide a sufficient update rate to maximize accuracy while minimizing network overhead. The process is designed to operate in a changeable mode between 8 sec and 36 hr.</p>
<b>Accept NTP requests</b>	When enabled the device will accepts NTP requests from outside. When automatic update is not enabled the device will share the local RTC clock time.

## Networks

Network parameters. Available parameter in edit mode:

Parameter	Description
<b>General Settings</b>	<p>Device hostname</p> <p>Avahi Hostname (see "<a href="#">Avahi Daemon</a>" on page 629)</p>
<b>Network Interface</b>	<p>Network parameters of the available interfaces</p> <ul style="list-style-type: none"> <li>DHCP</li> <li>IP Address</li> <li>Net Mask</li> <li>Gateway</li> </ul>
<b>DNS</b>	<p>DNS Servers</p> <p>Generally provided from the DHCP servers, but can be modified in edit mode</p>

Parameter	Description
	Search Domains Optional domains that will be used in concatenation with the provided urls

## Security



Services are available only when logged as admin.

The security area contains passwords and certificates, required by applications.

Parameter	Description
<b>Domain</b>	Identifies a set of secret information that can be used by installed applications that have the rights to use it. The preconfigured domains are: <ul style="list-style-type: none"> <li>• General This space is available for third party applications</li> <li>• System This space is used from the services embedded in the device (e.g. the VNC Server)</li> <li>• HMI Runtime This space is used from the HMI Runtime application</li> </ul>
<b>Secret ID</b>	Name used to identify each secret information included in the selected domain.
<b>Type</b>	Type of information to be stored. <ul style="list-style-type: none"> <li>• Text</li> <li>• Password</li> <li>• Certificate</li> <li>• File</li> </ul>
<b>Secret Info</b>	The secret information to keep stored..  In case of text or password, type the text or the password to store. In case of certificate or file use the "Update" button to upload the file to store.
<b>Description</b>	A free text that you can insert at will.

### Import/Export

Using the Import/Export commands, it is possible to export the stored information and import it, e.g., into other devices. Note that the export command will prompt you to define a password which will then be required in order to import the exported file.

## Applications

The applications page is listing the applications loaded on the HMI devices. From this page is possible to manage the applications.

Parameter	Description
<b>Name</b>	Application name
<b>Autostart</b>	If selected, the application will start when the operator panel is turned on

### App Management

Press the "App Manager" button to enter the application management mode from where you can:

- upload new applications
- update existing applications
- remove application
- define the startup sequence.

### Services



Services are available only when logged as admin.

Mouse click on the enable button to enable/disable the service. Click the service name to list the associate parameters.

### Autorun scripts from external storage

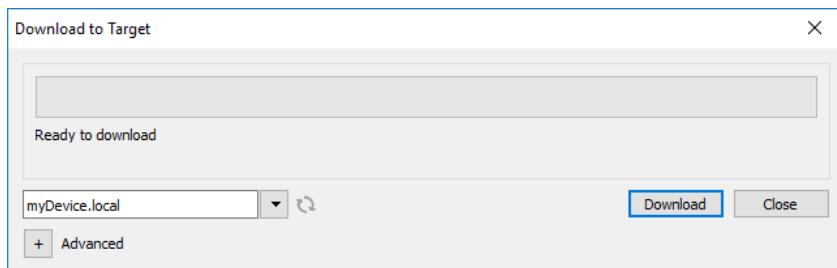
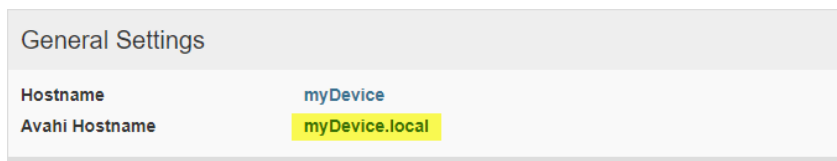
Enable/Disable the possibility to run the "autoexec.sh" script file when a USB key is plugged into the device. Disable this service if you want to prevent unauthorized access through the USB interface.



**Required BSP v1.0.212 or greater**


### Avahi Daemon

Avahi is a system which enables programs to publish and discover services and hosts running on a local network. When it is enabled, the HMI device can be reached even using the device's host name (in alternative to the IP Address).



Avahi Daemon runs on UDP port 5353

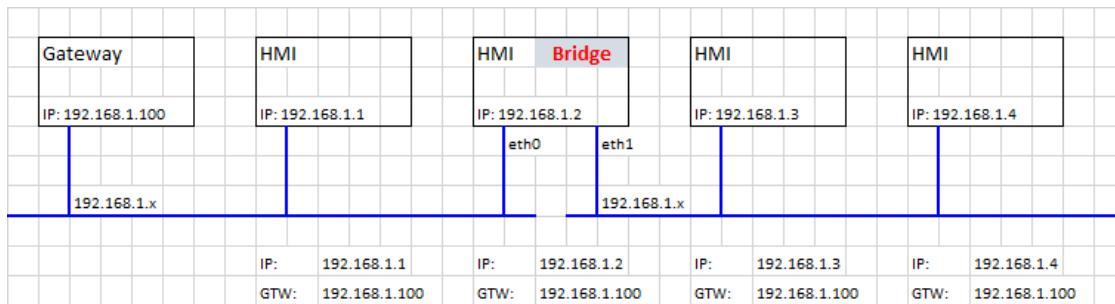


 On Linux and Apple PCs, the Avahi service comes for free with the OS. On Windows PCs instead, you need to install an Avahi service to be able to reach the panel by his Avahi host name (e.g. you need to install the Apple Bonjour application - Bonjour is a trademark of Apple inc.).

## Bridge/Switch Service

Using the bridge service is possible connect together the WAN (eth0) network adapter with the other network interfaces. When used, the two Ethernet interfaces are bridged and both Ethernet interfaces are sharing the same IP address.

Bridge Service creates a Linux-based layer-2 Network Bridge between two or more network interfaces. If both WAN and endpoint devices are attached to such bridge, the two networks will be physically joined and endpoints will be available as if they were directly connected to the WAN



## DHCP Server

Provide the DHCP Server on the selected interfaces.

Parameter	Description
<b>Enabled</b>	Enable the DHCP Server on the selected interface
<b>Start IP Stop IP</b>	IP addresses distributed from the DHCP Server
<b>Gateway</b>	The gateway address
<b>Netmask</b>	The provided netmask
<b>DNS Server</b>	The DNS server address
<b>Lease Time (seconds)</b>	Lease time, default is 86400s (1 day) Acceptable values are from 60s to 864000s (10 days)

## Enable device restore via TAP TAP option

When enabled, it gives the possibility to reset the operator panel in case the administrator password is forgotten. (See.: ["Forgot password" on page 644](#))



**This option is enabled by default. You can disable it to increase the security of the device (this will remove the possibility of recovering a forgotten password)..**

## Fast Boot

When fast boot is enabled, at the power up the HMI device will start the HMI application as fast as possible. In this mode, there are not showed diagnostic information (e.g. the loading bar) but only the minimum necessary features are loaded

before loading the User Interface (e.g. System Settings, VNC, SSH, etc. will be load after loading the HMI application).

To obtain best performance, in addition of enabling the fast boot mode, it is recommended to:

- disable any service that is not necessary
- avoid keeping enabled the persistent log
- use static IP address instead of DHCP service



**Required BSP v1.0.242 or greater**

## Firewall Service

When the firewall is enabled, only connections matching the defined rules are allowed. Note that some rules must be enabled for the HMI can to work properly.

### Firewall Service

**Enabled**

**Only connections matching the rules below are allowed - refer to documentation for other services**

Allow	Name	Source Interface	Source IP or Network	Port or Range	Protocol	
<input checked="" type="checkbox"/>	Web server - HTTP	Any		80	TCP	⬆️ ⬆️ 📄 -
<input checked="" type="checkbox"/>	Web server - HTTPS	Any		443	TCP	⬆️ ⬆️ 📄 -
<input checked="" type="checkbox"/>	Device discovery	Any		990-991	UDP	⬆️ ⬆️ 📄 -
<input checked="" type="checkbox"/>	FTP Command port	Any		21	TCP	⬆️ ⬆️ 📄 -
<input type="checkbox"/>	FTP Passive mode	Any		18756-18760	TCP	⬆️ ⬆️ 📄 -
<input type="checkbox"/>	SSH Server	Any		22	TCP	⬆️ ⬆️ 📄 -
<input type="checkbox"/>	VNC Server	Any		5900	TCP	⬆️ ⬆️ 📄 -
<input type="checkbox"/>	DHCP Server	Any		67	UDP	⬆️ ⬆️ 📄 -
<input type="checkbox"/>	SNMP Server	Any		161	UDP	⬆️ ⬆️ 📄 -

+

**Notes:**

- The firewall is based on IP tables which operates only at layer 3 (layer 2 packets won't be filtered, e.g. ARP)
- Only INPUT and FORWARD packets are filtered, not OUTPUT
- PING/ICMP echo reply packets are always allowed

- Internet sharing scenarios (e.g. 3g or wifi connection to endpoints) are not supported
- Packets filtered by the firewall are dropped

### Source IP or Network

If this field is unspecified, access will be allowed from any source host. Otherwise, access can be restricted to a single IP address (e.g. 192.168.100.123) or a range of IP addresses in CIDR format (e.g. 192.168.100.0/24). For details on valid subnet specifications following such format, please refer to: [https://en.wikipedia.org/wiki/Classless\\_Inter-Domain\\_Routing](https://en.wikipedia.org/wiki/Classless_Inter-Domain_Routing)



**If you enable the Firewall and you need to use the FTP passive mode with HMI Runtime older than version 2.10.0.280 then you need to open the ports 1024-2048/tcp and 16384-17407/tcp. From version 2.10.0.280 instead, HMI Runtime uses the ports 18756-18760/tcp that are proposed into Firewall settings by default.**

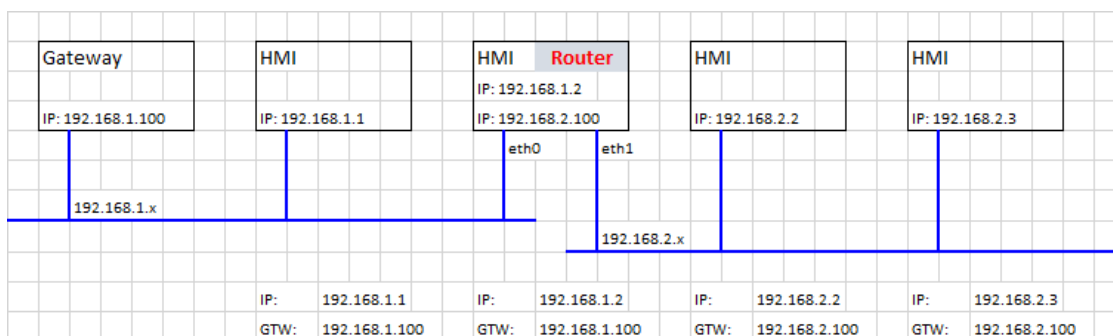


**Firewall is available from BSP v1.0.532**

**If you are updating from an old BSP version and you don't see the default rules, you have to reset the system settings (see "[Update System Components](#)" on page 638).**

### Router Service

This service uses IP Forwarding and Network Address Translation to share the connection from WAN (eth0) towards LAN (eth1 or eth2): connected endpoints may reach the same networks reachable by the gateway (including Internet if available).



### Port Forwarding

Port forwarding redirects incoming TCP packets requests from WLAN interface from one address and port number combination to another combination of address and port number.

Enabled	Name	Source Interface	Source Port	Device IP	Device Port
<input checked="" type="checkbox"/>	HMI-01	eth0	8081	192.168.55.1	80



**Available from BSP v1.0.507**

### 1:1 NAT

1:1 NAT, create alias IP on WLAN and forward all packets (or given port range) with that destination IP to another device attached to a LAN



**Available from BSP v1.0.507**

Enabled	Name	Source Interface	Source IP	Device IP	Port or Range (empty or P1 or P1-Pn)
<input checked="" type="checkbox"/>	HMI-02	eth0	192.168.1.10	192.168.55.10	



**Warning: make sure the value entered for “Source IP” is not the same as real IP address assigned to the physical Ethernet port specified as “Source Interface”.**

### Show loading bar during boot

Enable/Disable the display of the loading bar during the boot phase.

### SNMP Server

SNMP is a network protocol that allow to manage network infrastructures. It is commonly used to monitor network devices as switches, routers, etc. connected to a LAN network.

When the SNMP service is enabled, an SNMP Manager can retrieve information from the HMI device using the SNMP protocol. Currently, there are not proprietary MIBs available. Only the standard public community MIBs are available in read only mode.

Name/OID	Value	Type /	IP:Port
sysName.0	myDevice	OctetString	192.168.57.98:161
sysDescr.0	Linux myDevice 3.14.28-rt25-1.0.0_ga-g4f85bca #...	OctetString	192.168.57.98:161
sysUpTime.0	65 hours 42 minutes 25 seconds (23654530)	TimeTicks	192.168.57.98:161
memAvailReal.0	570808	Integer	192.168.57.98:161
memTotalFree.0	570744	Integer	192.168.57.98:161
ssCpuIdle.0	97	Integer	192.168.57.98:161

Example:

- System Name: .1.3.6.1.2.1.1.5.0
- System Description: .1.3.6.1.2.1.1.1.0
- System UpTime: .1.3.6.1.2.1.1.3.0
- Total RAM used: .1.3.6.1.4.1.2021.4.6.0
- Total RAM Free: .1.3.6.1.4.1.2021.4.11.0
- Idle CPU time (%): .1.3.6.1.4.1.2021.11.11.0

SNMP Server runs on UDP port 161



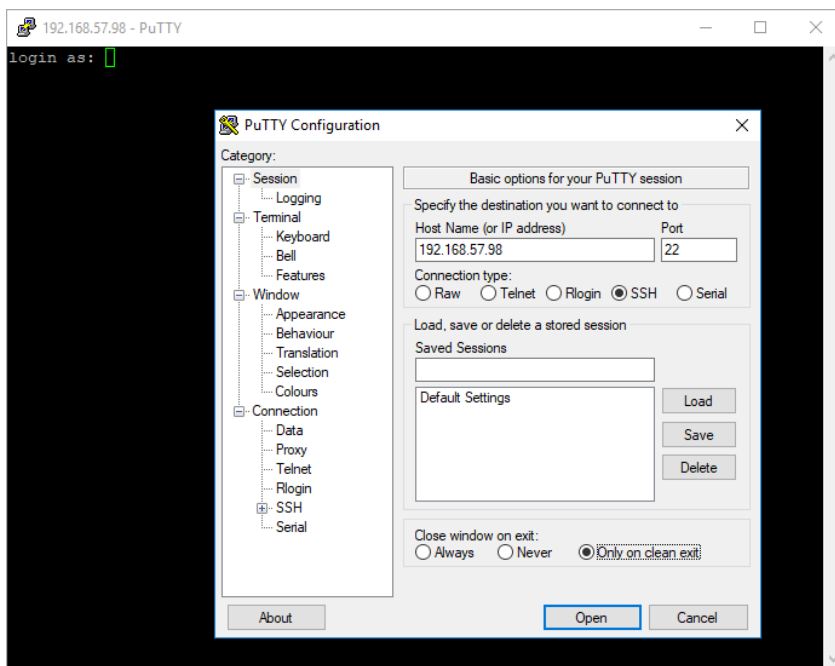
**For security reasons, do not enable the service if you do not need it.**

## SSH Server

SSH service has been designed only for advanced users. It provides remote login to HMI device using the secure shell protocol. On PC you can run a SSH Client as, for example, PuTTY that is an open source software distributed under the MIT license.



The default password for the username the admin is "admin". See the ["Authentication" on page 636](#) chapter to additional information.



SSH Server runs on TCP port 22



**This service is designed to be used during the development phase. For security reasons, remember to disable the service before switch to production.**


## VNC Service

VNC is a service that allows remote access to the display of the HMI device. VNC clients can be used to get the remote control of the HMI device.



**VNC should be disabled after use and autostart is not recommended.**

Parameter	Description
<b>Enable</b>	Enable the VNC server
<b>Autostart</b>	Keep the VNC server enabled when HMI device starts

Parameter	Description
<b>Port</b>	VNC Server listens for connections on TCP port 5900 (default)
<b>Inactivity timeout (seconds)</b>	“Inactivity timeout” occurs if no user interaction is detected (via keyboard, mouse, transfers or other RFB protocol interactions). The special value 0 indicates that idle timeout is disabled. Default value is 600 (10 minutes).
<b>Multiple clients</b>	Allow multiple sessions on the same port (if disabled, previously logged clients are disconnected upon a new incoming connection)
<b>View only</b>	Do not allow active user interactions (clients can only watch)
<b>Encryption</b>	<p>Activate SSL encryption of connections</p> <p><b>Custom certificate (Security/VNC KeyPair)</b></p> <p>The HMI device certificate that is necessary to permit the remote VNC client to verify the authenticity of the HMI device. The certificate must contain both the private and the public keys and can be .pem format.</p> <p> <b>The encryption features are not widely supported, check your VNC client compatibility</b></p>
<b>Authentication</b>	Whether users are authenticated upon session creation. A custom VNC specific password can be set or system passwords can be used (this option is only available if also Encryption is enabled)

Example of how to generate a certificate using OpenSSL library:

```
@echo off
set OpenSSL="C:\Program Files\OpenSSL-Win64\bin\openssl.exe"
set CertificateName=HMI-Certificate
set DeviceIP=192.168.1.56

rem Create the certificate keys
%OpenSSL% req -x509 -newkey rsa -days 365 -nodes -keyout private.pem -out public.pem -
subj "/ST=NY/C=US/L=New York/O=CompanyName/OU=Department/CN=%CertificateName%" -addext
"subjectAltName=IP:%DeviceIP%"

rem Create .pem file
copy private.pem + public.pem hmi-certificate.pem

echo.
echo.
pause
```

## Web Server

This page will show the parameters available to configure the Web Server. Note that it is not possible to disable the Web Server because it is necessary to allow access to the System Settings of the device.

- Allow only Secure HTTPS connections

Disabled by default to maintain backward compatibility, but it is recommended to enable it to improve the HMI device security.

- CORS domains enabled

When disabled (default), access to external domains is not allowed. When enabled, access to external domains listed in the "CORS domains filter" is allowed.

- CORS domains filter

You can enter the domain to which access is allowed or use a regular expression to define multiple domains. The regular expression must have the prefix "re:".

Leave the filter blank (default) if you want to maintain compatibility with older versions and allow access to all domains (this is not recommended).

Examples of "CORS domains filter":

- www.test.com
- re:(www.test1.com|www.test2.com)
- re:(www.test.(com|org))
- re:(www.test[1-9]+.com)

## Plugins

This page will show the parameters available to configure the optional plugins modules attached to the HMI device. See the description of the each plug-in module to additional information.

## Management



Management is available only when logged as admin.

From the management area is possible ["Update System Components" on page 638](#) of the HMI device.



**CAUTION: Working in the Management area is a critical operation and, when not performed correctly, may cause product damages requiring service of the product. Contact technical support for assistance.**

Use the "Clear" command inside the "Data" section to remove HMI Runtime from the device (Factory Restore)

## Display

Parameter	Description
Brightness	Brightness level of the display
Back light timeout	Backlight inactivity timeout
Orientation	Display orientation

## Authentication

Enter in edit mode to change the authentication passwords or to personalize the x.509 certificate of the HMI device.

## Users

There are two usernames:

- Administrator username with full access rights is **"admin"**
- Generic username with basic access rights is **"user"**

## x.509 Certificate

HMI Device use a self-certificate to encrypt the Internet communication trough the HTTPS protocol. You can personalize the certificate with the data of your Company and ask to a Certificate Authority to firm it.

The procedure to personalize and firm your certificate is:

1. Enter in edit mode and fill the necessary parameters, then push GENERATE button to generate a self-signed certificate with your data.
2. Export the "Certificate Signed Request"
3. Sent the "Certificate Signed Request" to a Certificate Authority to firm it (general this is a paid service)
4. Import the signed certificate into the HMI device

### Certificate's parameters

Parameter	Description
<b>Device Name</b>	The name of your device
<b>Organization</b>	The legal name of your organization
<b>Unit</b>	The division of your organization handling the certificate
<b>State</b>	The state/region where your organization is located
<b>Location</b>	The city where your organization is located
<b>Country</b>	The two-letter ISO code for the country where your organization is location
<b>Valid (days)</b>	Validity of the certificate
<b>Key Length</b>	Number of bits of the key used from the cryptographic algorithm

Managed certificates are base64 encoding



**Required BSP v1.0.239 or greater**

## Restart

HMI device restart command

## EXIT

Exit from the System Setting tool.



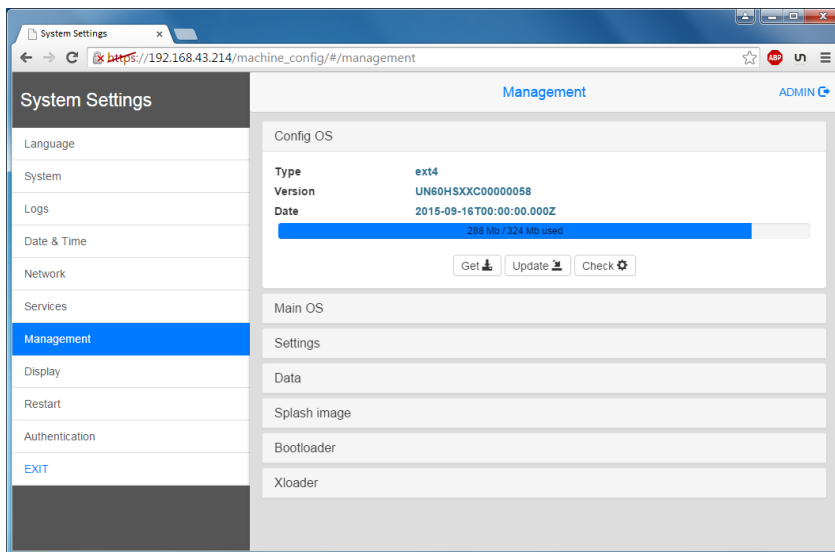
# Update System Components



**CAUTION:** Working in the Management area is a critical operation and, when not performed correctly, may cause product damages requiring service of the product. Contact technical support for assistance (the latest BSP files will be provided from tech support).

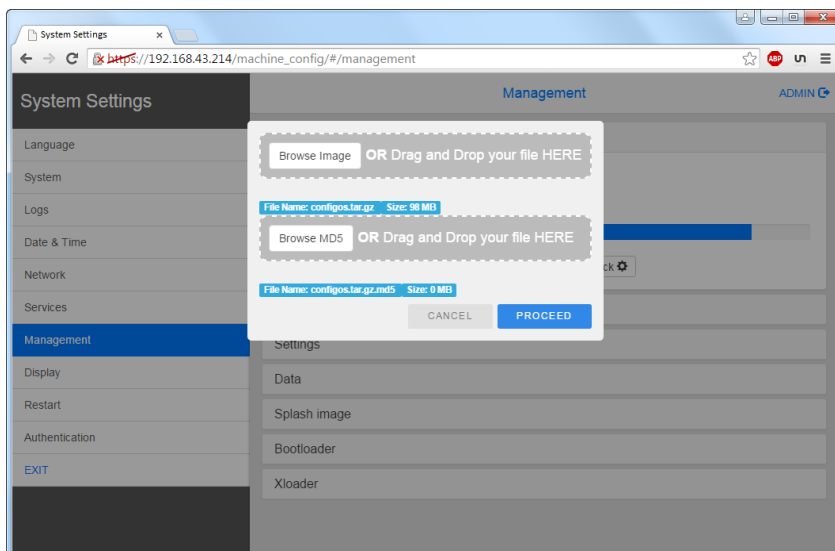
The system components of the CP600-eCo device can update locally using a USB memory key or remotely via web browser.

To update system components enter System Settings in Config OS mode via tap-tap procedure on HMI or open web browser to `https://<HMI-IP-address>/machine_config` and select the “Management” section.



Expand the component to update and select [Update]

On the opened dialog, click [Browse Image], then select the “xxx-mainos-xxx.tar.gz” file. Click then on [Browse MD5] and select the “xxx-mainos-xxx.tar.gz.md5” file.



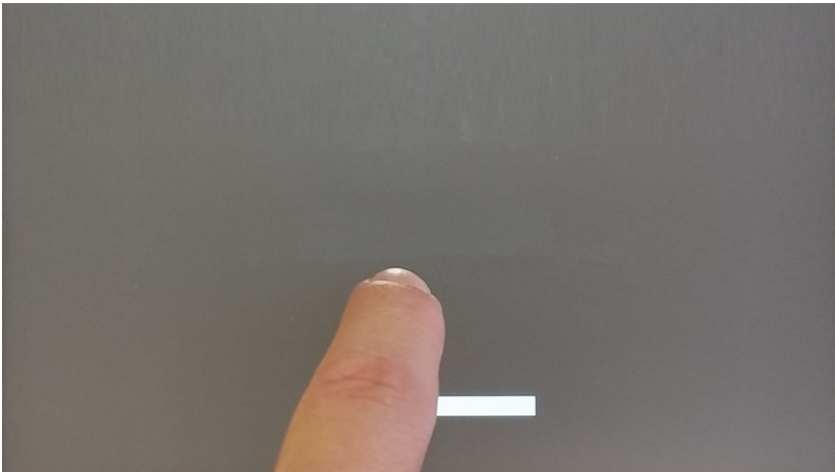
**Important:** Do not turn off the device while a system component is being upgraded.

At the end of the component update, restart HMI and leave it starting normally.

### Enter System Settings in Config OS mode via tap-tap procedure

System Setting in Config OS mode is available via tap-tap sequence, this mode can be accessed also when HMI is facing a software failure.

Tap-tap consist in a sequence of several touch activations by simple means of the finger tapping the touch screen performed during the power-up phase and started immediately after the HMI is powered on.



When “tap-tap detected” message appears on the top of the screen, press and hold the finger on touchscreen, to select “Restart: Config OS”



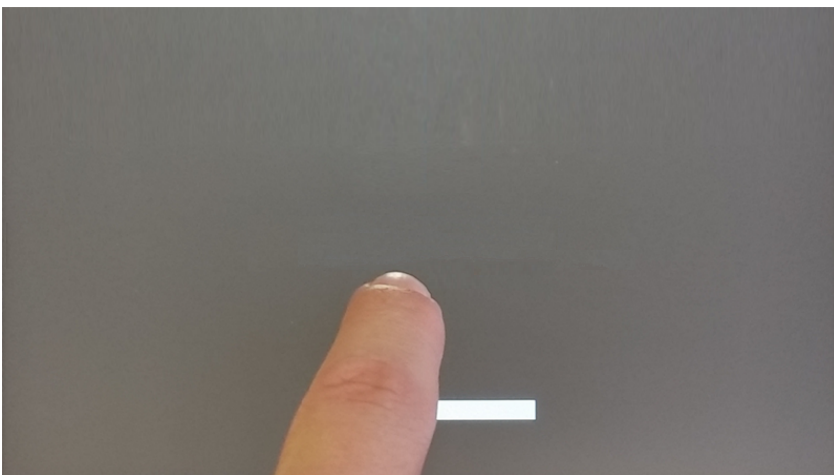
HMI will restart into System Settings in Config OS mode:



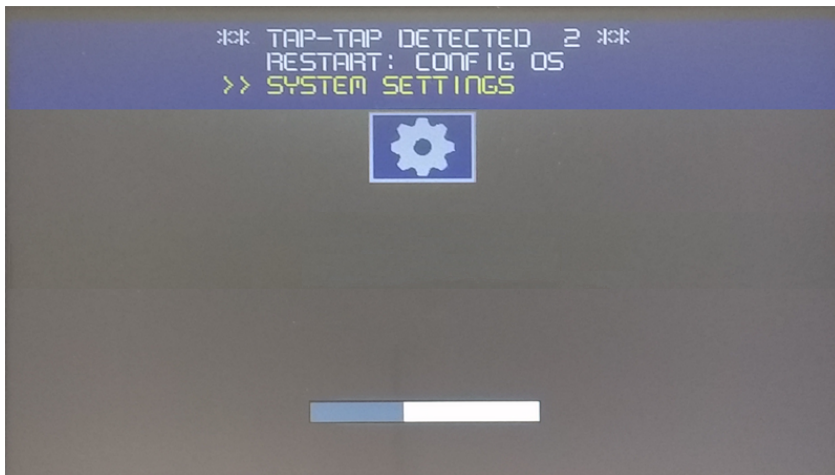
## Touchscreen calibration

System Setting Calibration allows to calibrate Touchscreen device, can be accessed by tap-tap procedure.

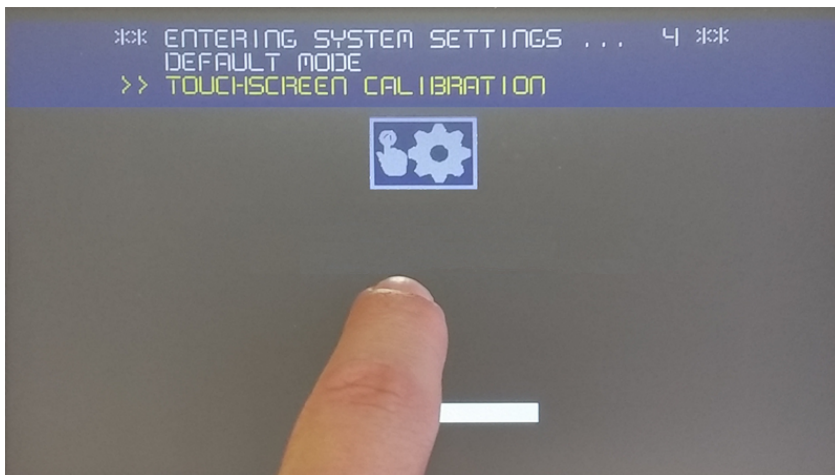
Tap-tap consists in a sequence of several touch activations by simple means of the finger tapping the touch screen performed during the power-up phase and started immediately after the HMI is powered on.



When "tap-tap detected" message appears on the top of the screen, wait for 5 seconds (without touching the screen) to enter System Settings sub menu



Press on touch screen, “Touchscreen calibration” voice will be highlighted in yellow, hold pressed for few seconds until touchscreen calibration procedure starts



Follow the instructions on screen to complete the calibration procedure, system will prompt to touch specific points to calibrate the touchscreen device.

## Password protection

Internal password of the HMI device.

From the Authentication tab, inside the ["System Settings" on page 622](#), activate the edit mode and select the username to change the associated password.

There are two usernames:

- Administrator username with full access rights is **"admin"**
- Generic username with basic access rights is **"user"**

**System Settings** CANCEL ✕

Localisation

System

Logs

Date & Time

Network

Security

Applications

Services

Management

Display

Fonts

Authentication

Restart

EXIT

---

**Users**

Username:

Old Password:

New Password:

Confirm Password:

Passwords are required to include:

- At least 8 characters in total
- At least one lower case and one upper case letter
- At least one numeric character
- At least one special character (eg. # ! @ ?)

x.509 Certificate

Password for admin user can be modified even from the context menu of the HMI Runtime (see ["Context menu options"](#) on page 10 for details) and from the update package (see ["Update package"](#) on page 104 for details).

If you forgot the password, check the ["Forgot password"](#) on page 644

**The first time the HMI device is turned on it is necessary to enter with the user "admin" and password "admin" to proceed with the definition of the passwords for both users (admin and user)**

Note that passwords must include:

- At least 8 characters in total
- At least one lower case and one upper case letter
- At least one numeric character
- At least one special character (eg. # ! @ ?)

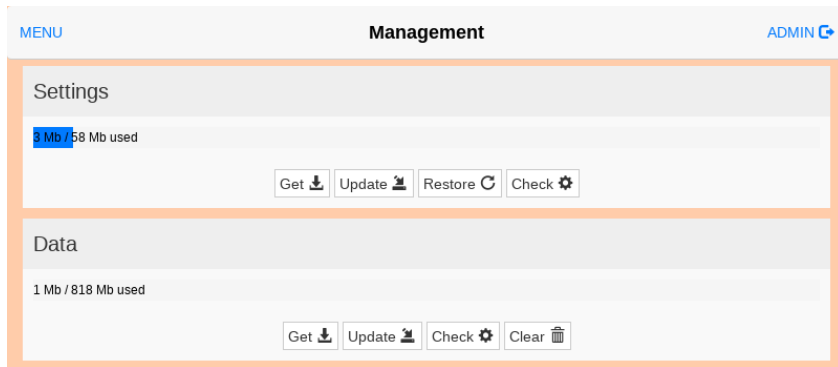
## Backup and Restore

To backup or restore all the installed applications with their settings, you must open the System Settings interface in Config OS mode using the tap-tap procedure.

See ["Enter System Settings in Config OS mode via tap-tap procedure"](#) on page 639

Then log as admin and select the "Management" option. From this page, you can use the "Get" button to backup inside an external memory (e.g. USB key) the contents of the **Data** and the **Settings** partitions. Use instead the "Update" button to restore the contents from a previous backup.

Management command is available only when logged as admin.



### Data Partition

The data partition contains the applications and they settings

### Settings Partition

The settings partition contains the settings of your device (this means the configuration parameters entered using the System Settings interface)



**When you update the System Settings from a backup you must be sure that the backup was executed from a device with the same BSP version (Main OS).**

### The MD5 file

The "Get" command will provide only a file with the contents of the partition (e.g. data.tar.gz), but if you want to restore the same file, using the "Update" command, you must provide even an MD5 checksum file.

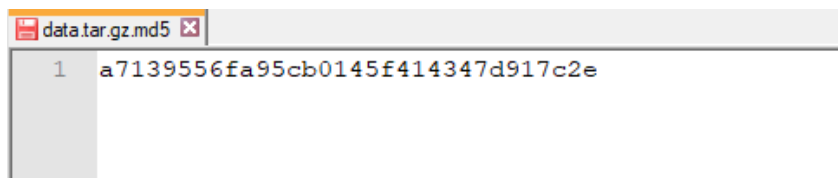
The MD5 checksum file must have the same name as the files that you want to load with the .md5 suffix as e.g.:

- data.tar.gz
- data.tar.gz.md5

On the Internet, it is easy to find various tools that calculate the MD5 checksum of a file. On Windows 10 it is also possible to use the "CertUtil" utility on the command line, e.g.

```
CertUtil -hashfile data.tar.gz MD5 > data.tar.gz.md5
```

The MD5 checksum file must have only one line. If the utility that calculates the checksum generates a file with multiple lines, the additional lines must be deleted.



## Recovery Mode

In the case that it is not even possible to boot the device, there is a special procedure to recovery the device by booting it in a special mode called configuration mode. From this mode you can open the device management dialog from where you can delete user data, restore system setting or update the firmware of the device.

To boot the device in configuration mode choice one of the below procedures

- Power on the device and immediately power off when splash screen appear on the screen (if you cannot see the splash screen, power off the device when you hear the beep-beep). Repeat this procedure for three time then power on again the device and wait the configuration mode appears.
- Create a special file named "\$0030D8\$.bin" and put it inside an empty SD card. Insert the SD card into the device and power on the device. Device will start in configuration mode.

## Forgot password

If you have forgotten the admin password, you have the possibility to reset it to the "admin" value. Note this procedure will erase the entire memory of the HMI device and any previously downloaded project will be removed.

### TAP TAP option

The procedure is available only if it has not been explicitly disabled through the "Enable device restore via TAP TAP option" available in the device system settings (Ref.: ["Enable device restore via TAP TAP option" on page 630](#))

Steps to reset the admin password:

1. Power off the HMI device.
2. Power on the HMI device and when the logo appears start to "tap tap" the touch panel (Ref.: ["Recovery Mode" on the previous page](#)).
3. When "TAP TAP" is detected select "System Settings" on the first menu, "Default mode" on the second menu, and finally "**Device restore**" on the third menu.

### USB option

The procedure is available only if it has not been explicitly disabled through the "Enable device restore via USB option" available in the device system settings (Ref.: ["Enable device restore via TAP TAP option" on page 630](#))

Steps to reset the admin password:

1. Placing a file named "*device-factory-restore*" into a USB stick and plugging it into the device.
2. The device restore process starts automatically. The buzzer is played once at the beginning and 3 times at the end if the operation is successful.
3. The "*device-factory-restore*" is deleted from the USB stick and the device rebooted.



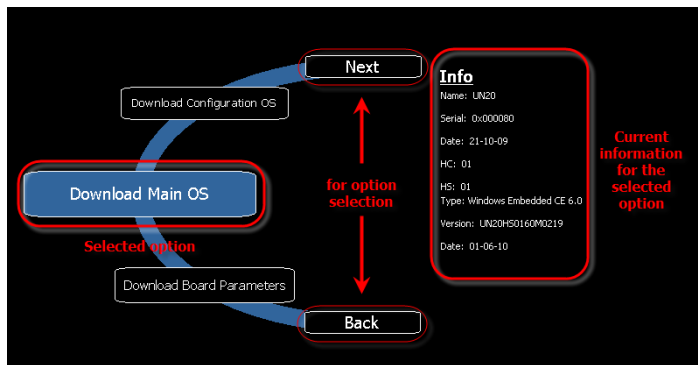
# 47 System Settings (WinCE Devices)

System Settings is an internal tool of the HMI device that can be used for the basic device settings or for the system components update.



**CAUTION: Working with the System Settings tool is a critical operation and, when not performed correctly, may cause product damages requiring service of the product. Contact technical support for assistance.**

The System Settings tool includes a rotating menu, and navigation buttons to scroll between the available options.



For each function and component on the left, the **Info** pane on the right displays all available information. In the example the version of the Main OS component is shown.

The System Settings tool can be used in two operating modes:

- User mode
- System mode.

For each mode different options are available.

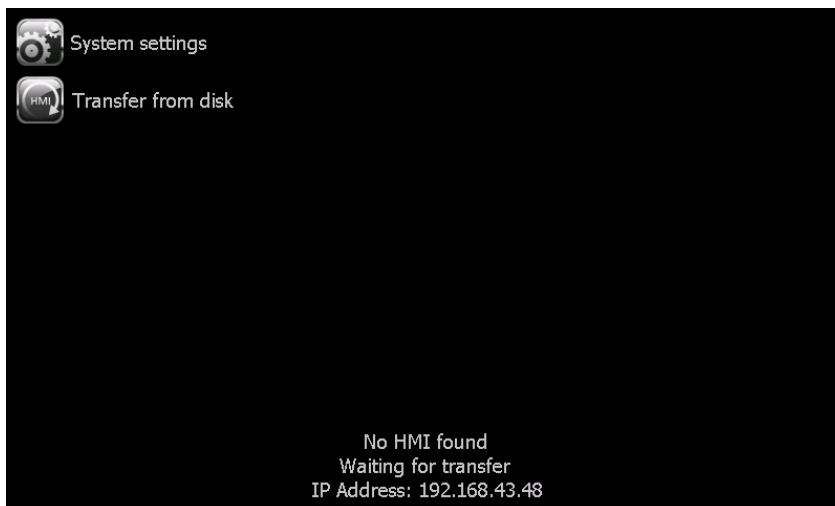
---

<b>Runtime Installation</b> .....	<b>645</b>
<b>System Settings</b> .....	<b>647</b>
<b>Update System Components</b> .....	<b>650</b>
<b>Touchscreen calibration</b> .....	<b>652</b>
<b>Password protection</b> .....	<b>653</b>
<b>Factory restore</b> .....	<b>653</b>

## Runtime Installation

HMI devices are delivered from factory without Runtime, at first power up HMI shows the “Runtime Loader” screen.





Runtime can be installed:

- Automatically, via Ethernet on first project download with PB610 Panel Builder 600
- Manually via USB Memory, creating an “Update Package”

## Install Runtime via Ethernet

To install Runtime via Ethernet follow the ["Download to HMI device" on page 102](#) procedure.



**WARNING: Runtime installation via Ethernet download requires the HMI to have a valid IP address.**

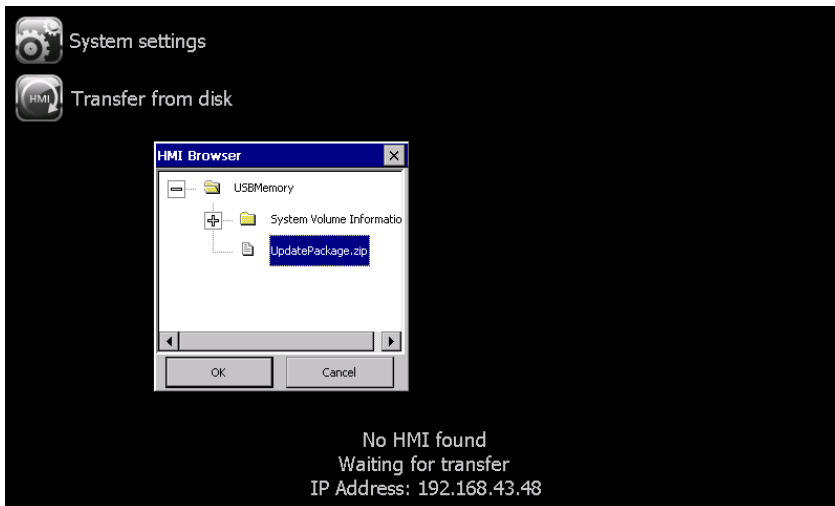
The IP address can be assigned in three ways:

- *Automatically via DHCP server.* This option is enabled by default. If a DHCP server is available on the network IP address will be assigned automatically by the server.
- *Automatically via Auto-IP feature.* If DHCP assignment is enabled but no DHCP server is available on the network the HMI assigns itself an IP Address into range 169.254.x.x with subnet mask 255.255.0.0
- *Manually via System Settings.* From System Settings menu, in Network section the IP address can be manually assigned, disabling the DHCP server assignment feature.

## Install Runtime via USB Memory

To install Runtime, UpdatePackage or Backup Package via USB device follow this procedure:

1. Create an Update Package from PB610 Panel Builder 600 and copy into an empty USB memory stick
2. On HMI select [Transfer from disk] and select the UpdatePackage.zip to load.



## System Settings

System Settings has two operating modes:

- **User Mode**  
a simplified interface that gives users access to the basic settings of the HMI device.
- **System Mode**  
a full interface that gives users access to all the tool's options.

When you access the tool at runtime selecting "Show system settings" from the context menu, the tool is started by default in User Mode.






Note: Press and hold on a screen area without buttons or other touch sensitive elements to display the context menu.


To access System Mode:

- Execute a tap sequence on the touch screen during the power-up phase. A tap sequence is a high frequency sequence of touch activations executed immediately after the device has been powered.
- From the System Setting page in User Mode, restart the panel in Configuration OS mode

### Elements available in User Mode



Element	Description
<b>Calibrate Touch</b>	Calibrate the touch screen
<b>Display settings</b>	Control backlight inactivity timeout and brightness
<b>Time</b>	Set HMI device date and time manually or configure NTP servers  <i>For the ARM devices, the synchronization frequency parameter of the NTP server is available from the BSP v2.21. On the MIPS devices and on the ARM devices with older BSP versions, the system searches the NTP server when turned on, or once a week if the HMI</i>


Element	Description																		
	 <i>device is not turned off.</i>																		
<b>Regional Settings</b>	Select or customize the regional setting parameters																		
<b>BSP Settings</b>	Display operating system version and unit operating timers to control buzzer and battery led.																		
<b>Network</b>	Sets IP address and other network settings																		
<b>Plug-in List</b>	<p>List the plug-in modules installed and recognized by the system.</p>  Note: this option may not be supported by all platforms and all versions.																		
<b>Close</b>	Closes the system setting page																		
<b>x.509 Certificate</b>	<p>You can personalize the certificate with the data of your Company and ask to a Certificate Authority to firm it.</p> <p>The procedure to personalize and firm your certificate is:</p> <ol style="list-style-type: none"> <li>1. Fill the necessary parameters, then push GENERATE button to generate a self-signed certificate with your data.</li> <li>2. Export the "Certificate Signed Request"</li> <li>3. Sent the "Certificate Signed Request" to a Certificate Authority to firm it (general this is a paid service)</li> <li>4. Import the signed certificate into the HMI device</li> </ol> <p><b>Certificate's parameters</b></p> <table border="1"> <thead> <tr> <th>Parameter</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td><b>Device Name</b></td> <td>The name of your device</td> </tr> <tr> <td><b>Organization</b></td> <td>The legal name of your organization</td> </tr> <tr> <td><b>Unit</b></td> <td>The division of your organization handling the certificate</td> </tr> <tr> <td><b>State</b></td> <td>The state/region where your organization is located</td> </tr> <tr> <td><b>Location</b></td> <td>The city where your organization is located</td> </tr> <tr> <td><b>Country</b></td> <td>The two-letter ISO code for the country where your organization is location</td> </tr> <tr> <td><b>Valid (days)</b></td> <td>Validity of the certificate</td> </tr> <tr> <td><b>Key Length</b></td> <td>Number of bits of the key used from the cryptographic algorithm</td> </tr> </tbody> </table>	Parameter	Description	<b>Device Name</b>	The name of your device	<b>Organization</b>	The legal name of your organization	<b>Unit</b>	The division of your organization handling the certificate	<b>State</b>	The state/region where your organization is located	<b>Location</b>	The city where your organization is located	<b>Country</b>	The two-letter ISO code for the country where your organization is location	<b>Valid (days)</b>	Validity of the certificate	<b>Key Length</b>	Number of bits of the key used from the cryptographic algorithm
Parameter	Description																		
<b>Device Name</b>	The name of your device																		
<b>Organization</b>	The legal name of your organization																		
<b>Unit</b>	The division of your organization handling the certificate																		
<b>State</b>	The state/region where your organization is located																		
<b>Location</b>	The city where your organization is located																		
<b>Country</b>	The two-letter ISO code for the country where your organization is location																		
<b>Valid (days)</b>	Validity of the certificate																		
<b>Key Length</b>	Number of bits of the key used from the cryptographic algorithm																		

Element	Description
	<p>Managed certificates are base64 encoding</p> <p> <b>Required BSP v2.29 or greater</b></p>
<b>Restart</b>	<p>Restart the HMI device</p> <ul style="list-style-type: none"> <li>• Main OS Restart the HMI device in the operating mode</li> <li>• Configuration OS Restart the HIM device with System Setting tool active in System Mode</li> </ul>

### Elements available in System Mode


In addition to those available in User Mode, the following features are also available:

Element	Description
<b>Format Flash</b>	<p>Formats the internal device flash disk. All projects and the HMI Runtime will be erased, returning the device to its factory settings.</p>
<b>Restore Factory Settings</b>	<p>Restores factory settings as an alternative to Format Flash, in a more flexible way. The following options are available:</p> <p><b>Uninstall HMI:</b> removes the HMI Runtime (entire qthmi folder) at the next start the device will behave as a brand new unit. This command does not reset settings such as IP address, brightness or RTC.</p> <p><b>Clear System Settings:</b> resets system parameters (registry settings) and deletes the following files:</p> <p><i>\\Flash\\Documents and Settings\\system.hv</i></p> <p><i>\\Flash\\Documents and Settings\\default\\user.hv</i></p> <p><i>\\Flash\\Documents and Settings\\default.mkv</i></p> <p><i>\\Flash\\Documents and Settings\\default.vol</i></p> <p>System Mode password is also reset.</p> <p><b>Clear sysdata settings:</b> clears \\Flash\\\$SysData\$ folder</p> <p> <i>Service call: To be used only by technical support to fix display problems.</i></p> <p> Note: Not all these options are available for all HMI devices and BSPs.</p>
<b>Resize Image Area</b>	<p>Resizes the flash memory reserved to store the splash screen image displayed at power up. Default settings are normally suitable for all units.</p>

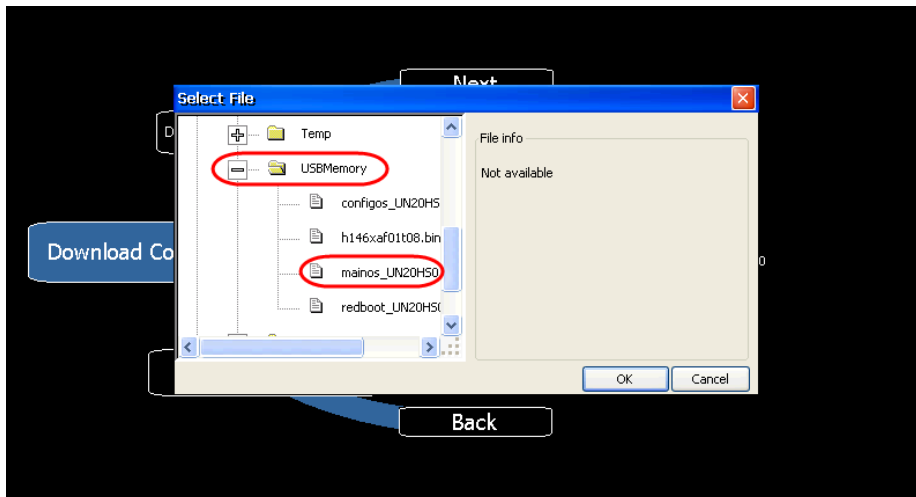
Element	Description
<b>Download Configuration OS</b>	Checks and upgrades the current version of the operating system used in System Mode
<b>Download Main OS</b>	Checks and upgrades the current version of the main operating system
<b>Download Splash Image</b>	<p>Loads a new file for the splash screen image displayed by the unit at power up.</p>  <p>Tip: Update the splash screen image directly from the PB610 Panel Builder 600 programming software.</p> <p>See "<a href="#">Update of system components from the application</a>" on page 656 for details.</p>
<b>Download Bootloader</b>	Checks and upgrades the current version of the system boot loader.
<b>Download Main FPGA</b>	Checks and upgrades the current version of the main FPGA file. This function may not be available for all platforms and versions.
<b>Download Safe FPGA</b>	Checks and upgrades the current version of the backup copy of the FPGA file. This function may not be available for all platforms and versions.
<b>Download System Supervisor</b>	Checks and upgrades the current version of the system supervisor firmware (used for the RTC and power supply handling).
<b>Upload Configuration OS</b> <b>Upload Main OS</b> <b>Upload Splash Image</b> <b>Upload Bootloader</b> <b>Upload Main FPGA</b> <b>Upload Safe FPGA</b> <b>Upload System Supervisor</b>	Copy the system files from the operator panel on the external device (usually an USB stick).

## Update System Components

System components can be updated using a USB flash drives. For each component, a couple of specific update files are provided.

 Note: Upgrading procedures depend on hardware and operating system versions. Contact technical support for assistance.

1. Copy all the upgrade files you need to a USB drive and plug it into the USB port of the HMI device.
2. Start the System Settings tool in System Mode (see "[System Settings](#)" on page 647 for details).
3. Click on the desired download function.
4. Browse the content of the USB drive to the files to download. The example shows Main OS components.



5. Click **Download** to transfer files to the HMI device.



Note: From this dialog click **Upload** to transfer files to the USB device.

6. Follow the instructions displayed to complete the update: the progress of the operation is displayed in a progress bar.

This operation may require a few minutes.





**Important: Do not turn off the device while a system component is being upgraded.**

## List of upgradable components

The HMI devices support the upgrade of the following components:

Component	Description
Application	The HMI Application and the HMI Runtime generated from the <b>Run&gt; Update Package</b> command
Main OS	Main Operating System
Configuration OS	Backup operating system that ensures units recovery in case of main operating system corruption
Splash	The initial screen shown during the startup of the HMI device
Bootloader	Loader to handle device startup
Main FPGA	FPGA firmware

Component	Description
Safe FPGA	Backup copy of the Main FPGA that ensures unit booting in case of main FPGA corruption   <b>Important: Use the same file for updating Main and Safe FPGA components.</b>
System Supervisor	Firmware of the system supervisor controller (for example: packaged_GekkoZigBee_v4.13.bin).  <div style="background-color: #e0f2f1; padding: 5px; border-radius: 10px; text-align: center;"> <i>The System Supervisor component can be upgraded from v4.13 or above.</i> </div>  <b>Important: Do not try to update versions V4.08, V4.09, V4.10 and V4.11 since they do not support automatic update from System Settings.</b>

## Touchscreen calibration

System Setting Calibration allows to calibrate Touchscreen device, can be accessed from System Settings

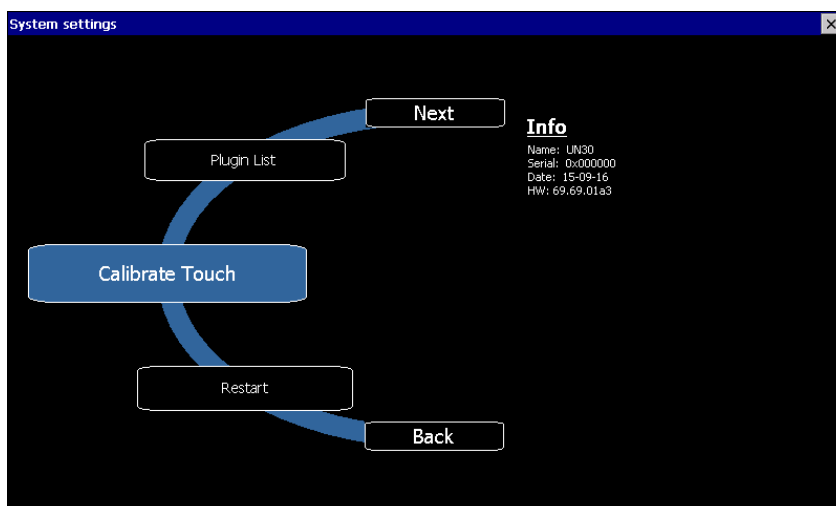
To access System Settings:

- Execute a tap sequence on the touch screen during the power-up phase. A tap sequence is a high frequency sequence of touch activations executed immediately after the device has been powered.

or

- Press and hold on an empty area of the screen for a few seconds to display the context menu.

From the rotating menu, select “*Calibrate Touch*” and follow the instructions on screen to complete the calibration procedure, system will prompt to touch specific points to calibrate the touchscreen device.



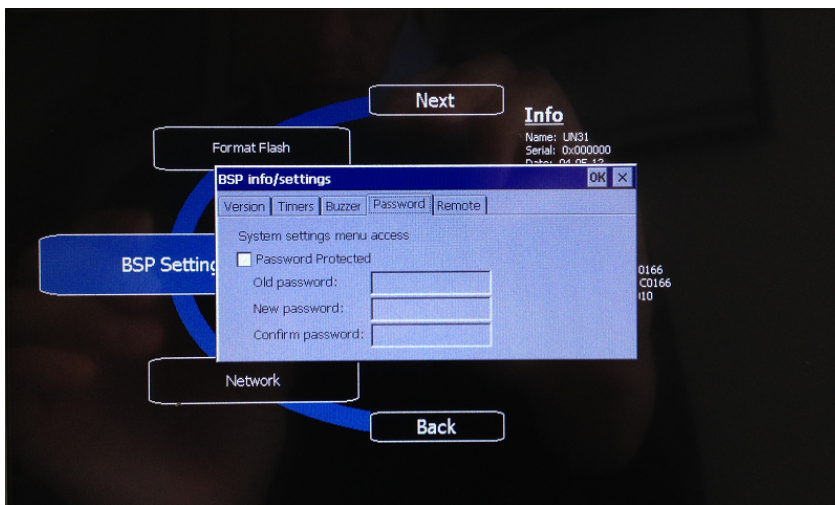
## Password protection

Internal password of the HMI device can be defined from the System Settings in System Mode (see "[System Settings](#)" on [page 647](#) for entering in system settings mode)

There are two password tabs:

- Password  
Protect the system settings from local accessing
- Remote  
Protect the system settings from remote accessing

From the rotating menu, select "[BSP Settings](#)" and then the Password tab or the Remote tab to open the set password dialogs.



The password must be at least 5 characters long.



Leave "Old password" empty as default if target password is not set.



The "Remote" password can be modified even from the context menu of the HMI Runtime (see "[Context menu options](#)" on [page 10](#) for details) and from the update package (see "[Update package](#)" on [page 104](#) for details)



*This feature is available from BSP versions V1.64 ARM UN30/31 and V2.73 MIPS UN20 based on WCE OS.*

## Factory restore

If you're having problems with the HMI device, try and restore factory default settings from System Mode.

1. Enter **System Mode**.
2. Use one of the following operations available in rotating menu:
  - **Format Flash**, to clean the flash drive and registry configuration.
  - **Restore Factory Settings**, to clean only the select components.





Note: Both operations do not involve firmware factory restore (MainOS, ConfigOS, Bootloader, FPGA images, etc).

See "[System Settings](#)" on page 647 for details.

### System Mode

To access System Settings tool in System Mode you may use one of the below procedures:

- TAP-TAP sequence: this procedure consists in a tap-tap sequence over the touch interface during power-up phase. Tap-tap consist in a high frequency sequence of touch activations by simple means of the finger tapping the touch screen performed during the power-up phase and started immediately after the device is powered on. When the procedure succeeded the system returns a visual feedback, the text: "Tap Tap detected, Going to Config Mode" is displayed in page.
- BIN file procedure: this procedure consists in the use of an SD Card, containing a specific file, called "\$0030D8\$.bin". Copy the BIN file into the SD Card, then insert the card into the proper slot on the device. Power on the device, when the file is detected the system returns a visual feedback, the text: "Tap Tap detected, Going to Config Mode" is displayed in page.



**Warning: if you are updating a Series Glass Device it is necessary to use the BIN file procedure**

# 48 Updating system components in HMI devices

---

Most of the system software components can be easily upgraded ensuring a high degree of flexibility in providing updates and fixes to existing and running systems.

New software modules can be updated

- Directly on HMI device using an USB flash drives (see "[System Settings](#)" on page 622 for details)
- From PB610 Panel Builder 600 application (see "[Update of system components from the application](#)" on the next page for details)

Each HMI device is labeled with a product code including all factory settings (hardware, software and firmware components). Refer to this label for information on your HMI device. The HMI device update tool also provides detail on the components actually running on the device.



**CAUTION:** Make sure you use the correct upgrade files, since loading upgrade files unsuitable for your device will cause serious system malfunction. Always check your device product code.



Note: Upgrade files are distributed upon request as a part of technical support activity.



*Service call: Downgrade operations are complex tasks which might cause serious damage to your equipment if not performed correctly. These operations are reserved to technical support.*

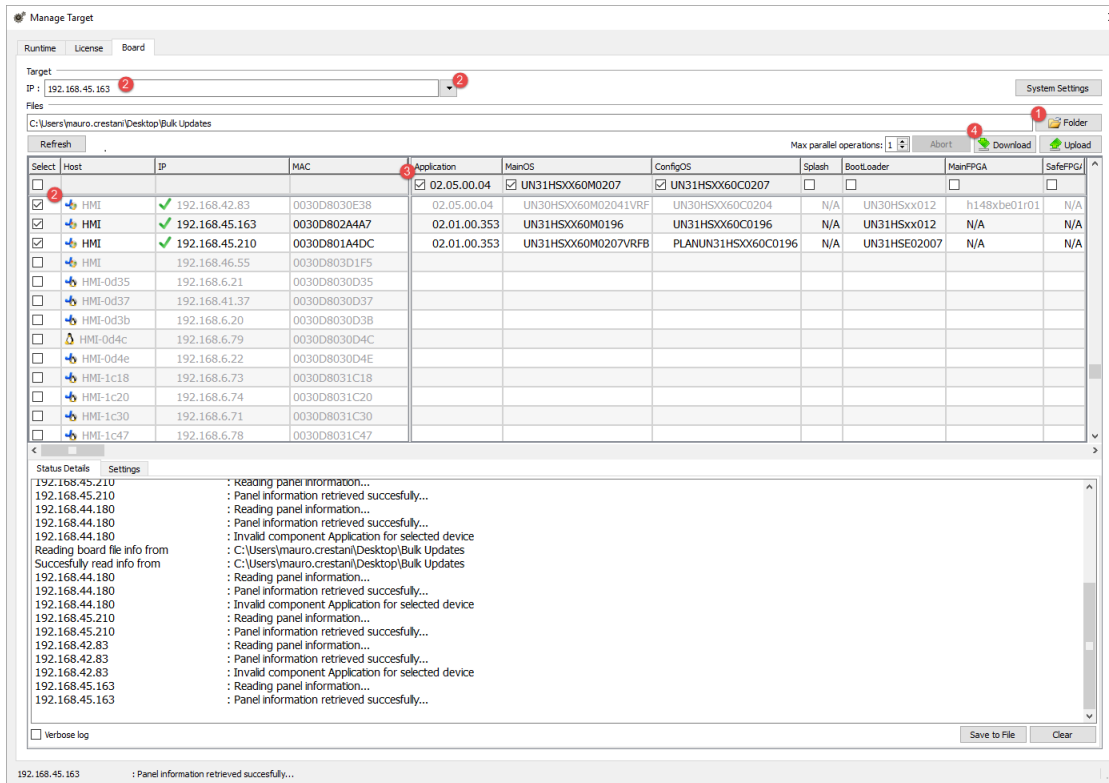
---

<b>Update of system components from the application</b> .....	<b>656</b>
---	------------

# Update of system components from the application

You can download system components to a single HMI device or to a bulk of HMI devices of the same type using the Ethernet communication interface.

**Path: Run > Manage Target > Board**



1. Select the folder that contains the files to download to the HMI device or where to upload files from the HMI device
2. Select one or more HMI device.
3. Select the components that you will download (or upload) to/from the devices
4. Start the Download to HMI or the Upload from the HMI operation

## Note:

- The tool is designed to update multiple HMI devices of the same type. Please avoid putting files for different device type into the same folder
- If the desired target IP is not listed, type it directly into the box. The discovery service is a broadcast service. When a remote connection is done via VPN or from external networks, it will not work and you will have to enter the address manually.
- Download of the selected components will be performed only to the compatible devices
- Based on your network and hardware capabilities you can increase the number of devices to update in parallel
- You need to restart the HMI device to finalize the update.

## Settings

From the **Settings** tab you can specify the Port and the Password parameters to use to communicate with the HMI devices. Leave Password empty if no password is set on the HMI device side.



**WARNING: Bulk mode is working only with the HMI devices that have the same connection parameters**

## Uploading a splash screen picture

You can replace the default splash screen image shown by the devices during the power up phase.

The image used as splash screen must comply with the following requirements:

Filename	splash.bmp
Format	Bitmap, RGB 565 format
Size	< 500 KB
Bitmap width	Even number (for example 430x239)

To upload the splash screen image:

1. Rename the new image splash.bmp and copy it in the source folder.
2. Select HMI devices
3. Click **Download**.



To ensure the best visual results, splash screen images must have a black background.



# 49 Communication protocols

---

This section describes the available protocols.



Note: Changes in controller hardware or protocols may have occurred since this documentation was created. Always test and verify the functionality of the application. To accommodate developments in the controller hardware and protocols, drivers are continuously updated. Accordingly, always ensure that the latest driver is used in the application.

Different physical media, gateways, routers and hubs can be used in the communication network. Also, other devices can independently make simultaneous use of the network. However, it is important to ensure that the traffic generated by these devices does not degrade the communication speed (round-trip time) to an unacceptable level.

---



# ABB CODESYS Ethernet

The ABB CODESYS Ethernet communication driver for Ethernet has been specifically designed to support communication with ABB controllers series AC500 designed for standardized IEC 61131-3 programming, based on the CODESYS V2.3 system.



Note: CODESYS Ethernet Driver is supported with AC500 FW 2.1 or newer and not with AC500 FW 1.xx



Note: To accommodate developments in the controller protocol and hardware, drivers are continuously updated. Make sure the latest driver is used in the application.



Note: Changes in the controller protocol or hardware may have occurred since this documentation was created. This may interfere with the functionality of this driver. Therefore, always test and verify the functionality of the application.

## Limitations

CODESYS Level 4 is not supported. Maximum block size is 1024.

## Protocol Editor Settings

### Adding a protocol

To configure the protocol:

1. In **Config** node double-click **Protocols**.
2. To add a driver, click **+**: a new line is added.
3. Select the protocol from the **PLC** list.

The protocol configuration dialog is displayed.

Add a driver in the Protocol editor and select **ABB CODESYS ETH** from the list of available protocols.

The following protocols type are supported:

- Tcp/Ip Level 2 Route
- ABB Tcp/Ip Level 2 Route AC
- Tcp/Ip

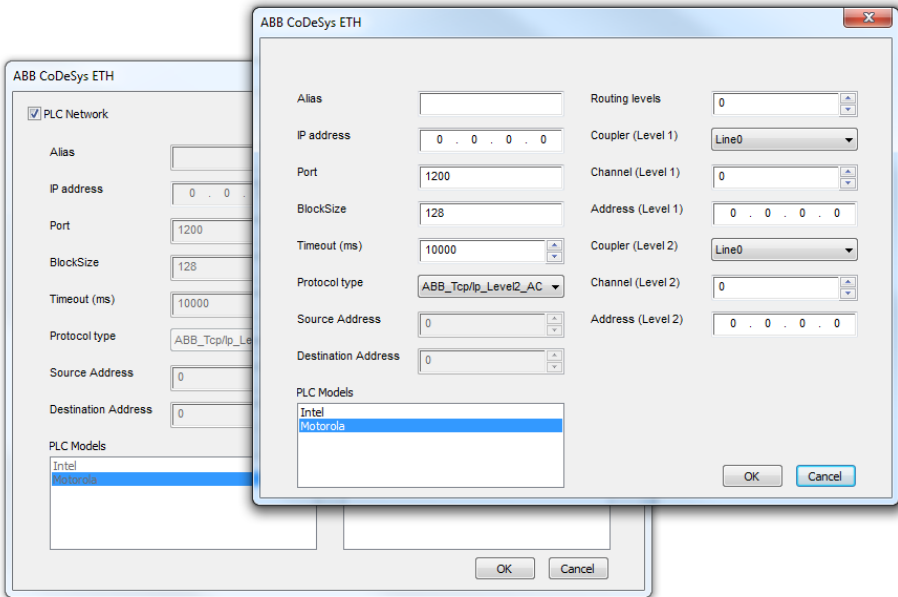
Select protocol type from the **Protocol type** combo box in the **ABB CODESYS ETH** dialog.



Some of the parameters of the dialog are common to the different protocol types, others are specific.

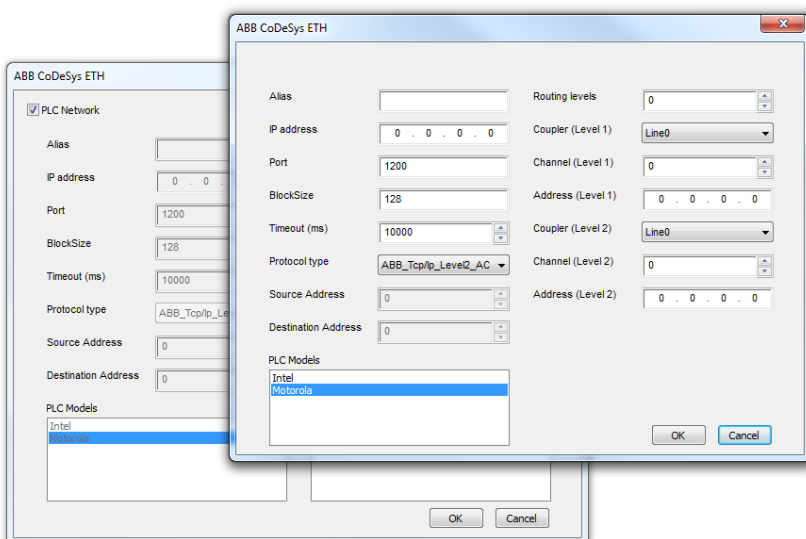
The parameters common to the different protocol types are:

Element	Description
<b>Alias</b>	Name identifying nodes in network configurations. The name will be added as a prefix to each tag name imported for each network node.
<b>IP address</b>	Ethernet IP address of the controller
<b>Port</b>	Port number used for the communication. Default value is 1200 for ABB drivers. For AC500 and 3S drivers select port 1201.
<b>BlockSize</b>	The max block size supported by your controller
<b>Timeout</b>	Time delay in milliseconds between two retries in case of missing response from the server device.
<b>PLC Model</b>	Byte order that will be used by the communication driver when sending communication frames to the PLC; Intel is also commonly referred as “little-endian”, Motorola as “big-endian” Select “Motorola” for AC500.

Element	Description
<b>Protocol type</b>	<p>Three different protocol types are available:</p> <ul style="list-style-type: none"> <li>• <b>Tcp/Ip</b></li> <li>• <b>Tcp/IP Level2 Route</b></li> <li>• <b>ABB Tcp/Ip Level2 AC</b></li> </ul>
<b>PLC Network</b>	<p>The protocol allows the connection to multiple controllers. To set-up multiple connections, check "PLC network" checkbox and enter the IP Address per each slave you need to access.</p> 

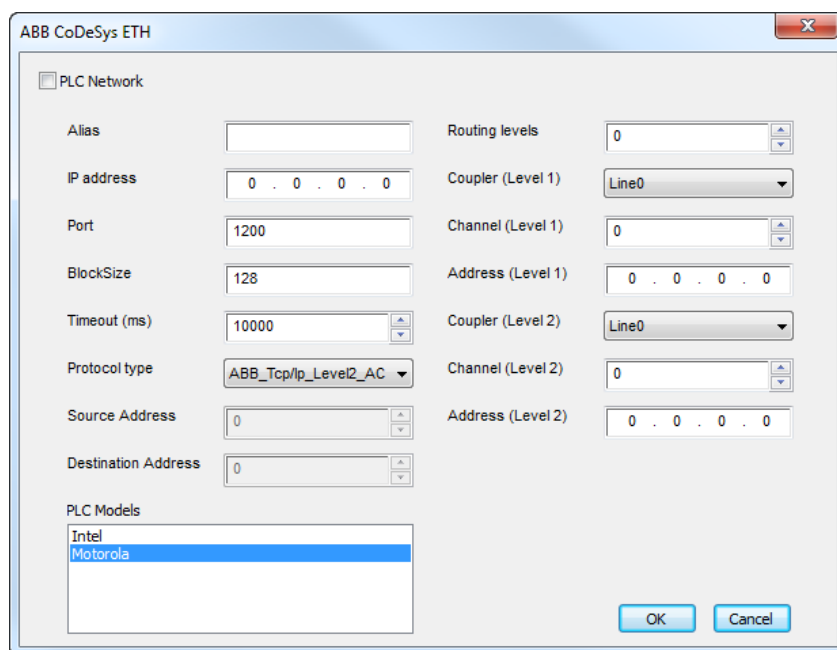
## Protocol types

The **Tcp/Ip** protocol type corresponds to the 3S Level 4 driver and does not require additional setup besides the common parameters.



The **Tcp/IP Level2 Route** protocol type corresponds to the standard 3S Level 2 Route driver and requires two additional parameters:

Parameter	Description
<b>Source Address (SrcAdr), Destination Address</b>	Destination is the node of the PLC and allows the protocol to read variables in a sub-network. The address is used to read variables when multiple PLCs are connected in a sub-network (serial network) but only one of it has the Ethernet interface.  <i>This is currently not applicable for AC500 PLCs.</i>



The **ABB Tcp/Ip Level2 AC** protocol type implements a specific variation of the standard Level 2 protocol with the additional use of a routing driver. This protocol type is normally used to connect to PLCs via other PLCs acting as gateways.

This protocol type requires the following additional parameters:

- Routing Levels
- Coupler (Level 1)
- Channel (Level 1)
- Address (Level 1)
- Coupler (Level 2)
- Channel (Level 2)
- Address (Level 2)

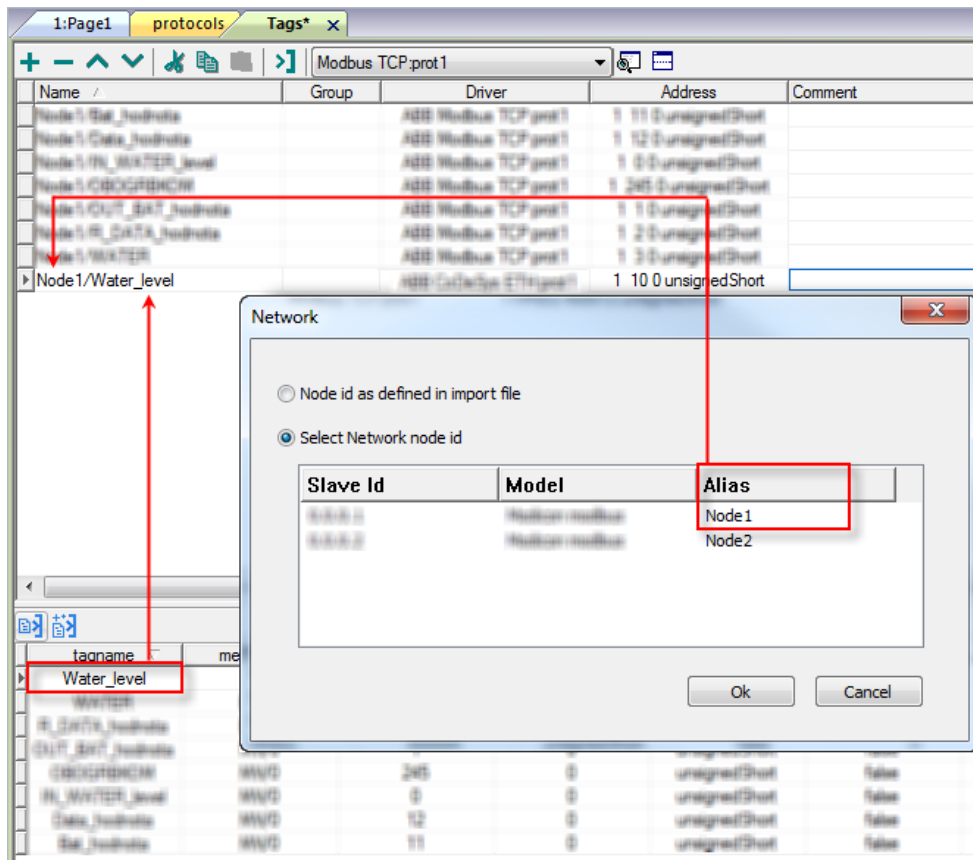
For detailed information see *AC500 and Control Builder* documentation, chapter *Programming interfaces to the AC500 used by the Control Builder*.

## Adding an alias name to a protocol

Tag names must be unique at project level, however, the same tag names might need to be used for different controller nodes (for example when the HMI device is connected to two devices running the same application).

When creating a protocol you can add an alias name that will be added to tag names imported for this protocol.

In the example, the connection to a certain controller is assigned the name **Node1**. When tags are imported for this node, all tag names will have the prefix **Node1** making each of them unique at the network/project level.

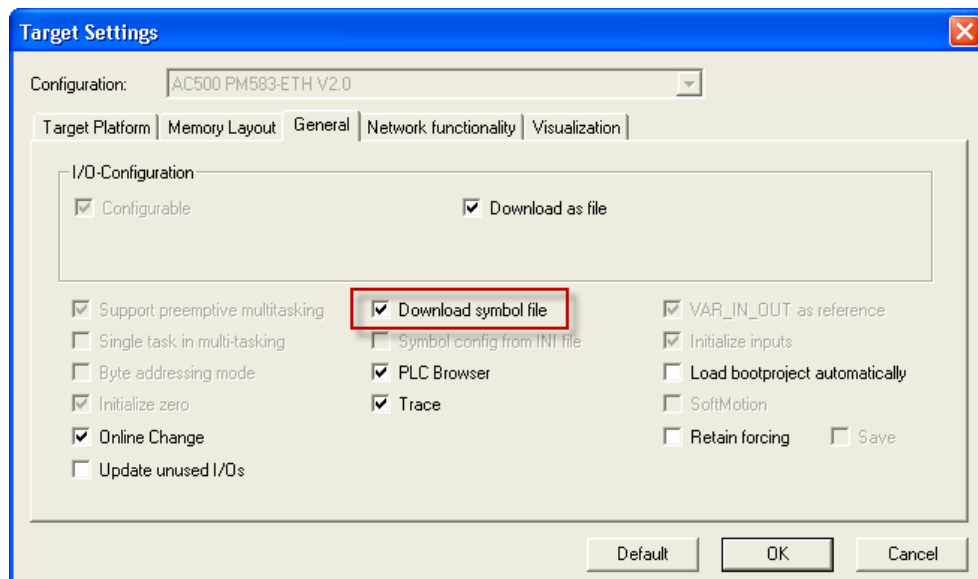



Note: Aliasing tag names is only available for imported tags. Tags added manually in the Tag Editor cannot have the Alias prefix in the tag name.

The Alias string is attached at the time of tag import. If you modify the Alias string after the tag import has been completed, there will be no effect on names already present in the dictionary. When the Alias string is changed and tags are re-imported, all tags will be re-imported with the new prefix string.

## CODESYS Software Settings

When you create the project in CODESYS V2, select **Download symbol file** (*Target Settings > General*).



 Note: ABB CODESYS Ethernet driver supports the automatic symbol file (SDB) upload from the controller; any change in the tag offset due to new compilation on PLC software side does not require a symbol file re-import. The Tag file has to be re-imported only in case of tag renaming or addition of new tags.

## Standard Data Types

The import module supports variables of standard data types and user defined data types.

The following are considered as standard data types:

### Supported data types

- BOOL
- WORD
- DWORD
- INT
- UINT
- UDINT
- DINT
- STRING\*
- REAL
- TIME
- DATE & TIME

and 1-dimensional ARRAY of the types above. See "Programming concepts" section in the main manual.

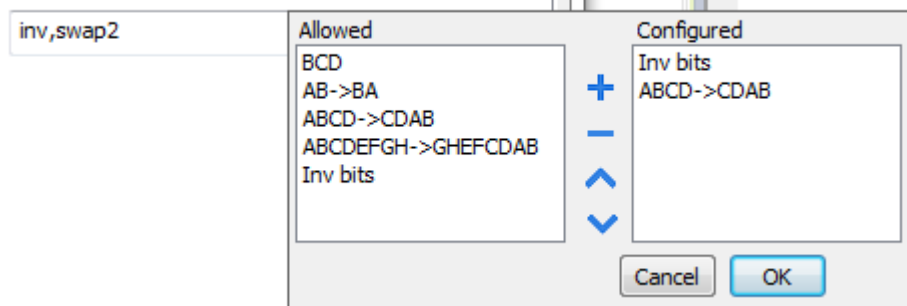


Note \*: String length for a STRING variable in PLC should be max 80 characters. Declare a STRING variable either with a specified size (str: STRING(35) or default size (str: STRING) which is 80 characters.

## Tag Conversion

Conversion to be applied to the tag.

Conversion



Depending on data type selected, the list **Allowed** shows one or more conversion types.

Value	Description
<b>Inv bits</b>	<p><b>inv</b>: Invert all the bits of the tag.</p> <p><i>Example:</i>                      1001 → 0110 (in binary format)                      9 → 6 (in decimal format)</p>
<b>Negate</b>	<p><b>neg</b>: Set the opposite of tag value.</p> <p><i>Example:</i>                      25.36 → -25.36</p>
<b>AB -&gt; BA</b>	<p><b>swapnibbles</b>: Swap nibbles in a byte.</p> <p><i>Example:</i>                      15D4 → 514D (in hexadecimal format)                      5588 → 20813 (in decimal format)</p>
<b>ABCD -&gt; CDAB</b>	<p><b>swap2</b>: Swap bytes in a word.</p> <p><i>Example:</i>                      9ACC → CC9A (in hexadecimal format)                      39628 → 52378 (in decimal format)</p>
<b>ABCDEFGH -&gt; GHEFCDAB</b>	<p><b>swap4</b>: Swap bytes in a double word.</p> <p><i>Example:</i>                      32FCFF54 → 54FFFC32 (in hexadecimal format)                      855441236 → 1426062386 (in decimal format)</p>

Value	Description
<b>ABC...NOP -&gt; OPM...DAB</b>	<p><b>swap8</b>: Swap bytes in a long word.</p> <p>Example:            142.366 → -893553517.588905 (in decimal format)            0 10000000110            0001110010111011011001000101101000011100101011000001            →            1 10000011100            1010101000010100010110110110110010110110000100111101            (in binary format)</p>
<b>BCD</b>	<p><b>bcd</b>: Separate byte in two nibbles, read them as decimal (from 0 to 9)</p> <p>Example:            23 → 17 (in decimal format)            0001 0111 = 23            0001 = 1 (first nibble)            0111 = 7 (second nibble)</p>

Select conversion and click +. The selected item will be added to list **Configured**.

If more conversions are configured, they will be applied in order (from top to bottom of list **Configured**).

Use the arrow buttons to order the configured conversions.

## Node Override IP

The protocol provides the special data type Node Override IP which allows you to change the IP address of the target controller at runtime.

This memory type is an array of 4 unsigned bytes, one per each byte of the IP address.

The Node Override IP is initialized with the value of the controller IP specified in the project at programming time.

Node Override IP	PLC operation
<b>0.0.0.0</b>	Communication with the controller is stopped, no request frames are generated anymore.
<b>Different from 0.0.0.0</b>	It is interpreted as node IP override and the target IP address is replaced runtime with the new value.

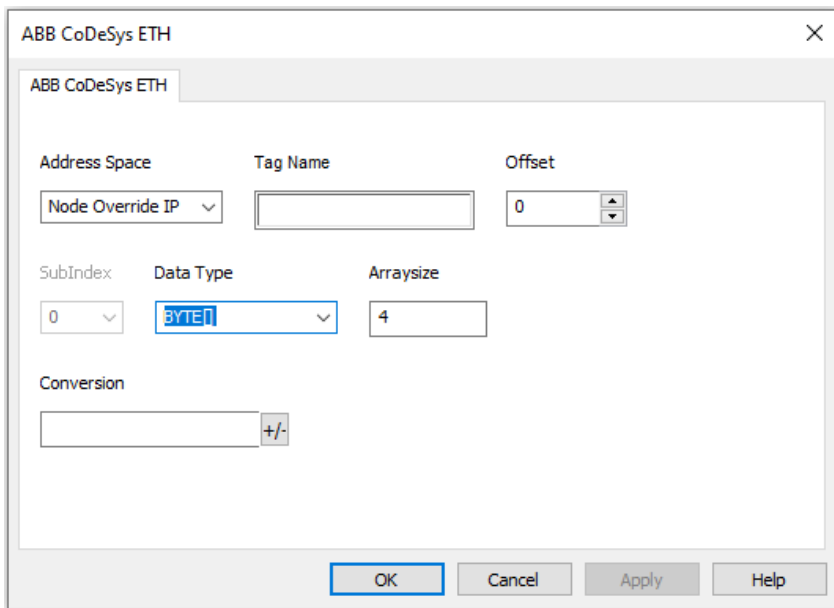
If the HMI device is connected to a network with more than one controller node, each node has its own Node Override IP variable.



Note: Node Override IP values assigned at runtime are retained through power cycles.

## Hostname DNS or mDNS

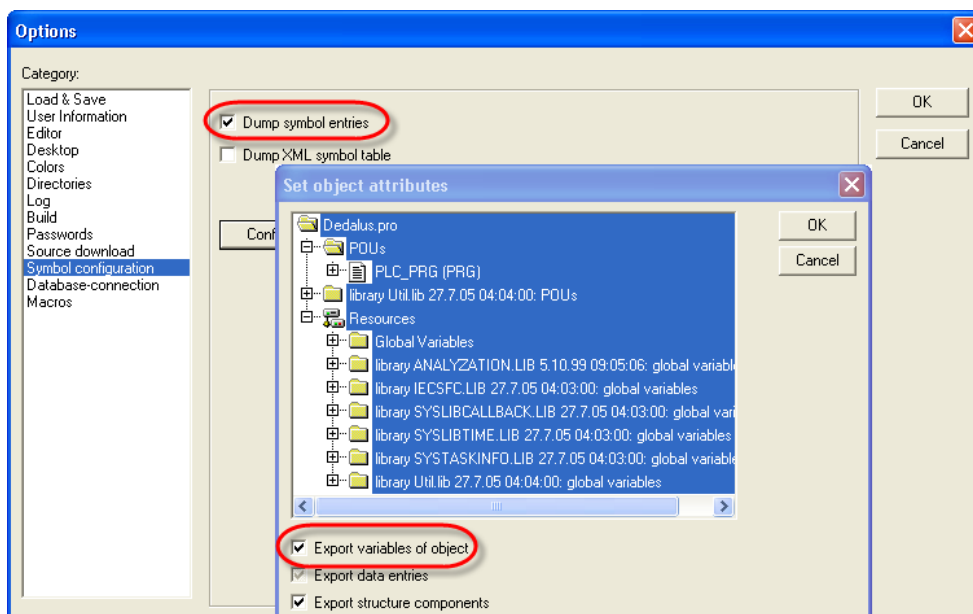
In addition to the array of bytes, string memory type can be selected to be able use the DNS or mDNS hostname as an alternative to the IP Address.



## Exporting tags from the controller

When configuring PLC using the manufacturer's configuration software, enable Symbol file (file with .sym extension) creation under the CODESYS programming software:

1. In the **Project** menu, click **Options**.
2. Select **Symbol configuration**.
3. Select **Dump symbol entries**.
4. Click **Configure symbol file**: the **Set object attributes** dialog is displayed.
5. Select **Export variables of object**.
6. Click **OK**.





## Importing tags

You may import tags from a .sym file exported from a controller. See "My first project" section in the main manual.

## Communication status

The current communication status can be displayed using system variables.

See "System Variables" section in the main manual.

Codes supported for this communication driver:

Error	Cause and action
<b>Symbols file not present</b>	Check Symbol file and download again the PLC program
<b>"tag" not present in Symbols files</b>	Check if the Tag is present into the PLC project
<b>Time out on Acknowledge</b>	Controller didn't send acknowledge
<b>Time out on last Acknowledge</b>	Controller didn't send last acknowledge
<b>Time out on data receiving</b>	Controller didn't reply with data
<b>Connection timeout</b>	Device not connected

# ABB CODESYS Serial

The ABB CODESYS Serial communication driver has been specifically designed to support communication with Series 500 ABB controllers designed for IEC 61131-3 programming, based on the CODESYS V2.3 system.

## Limitations

This protocol does not support AC500 firmware version earlier than V2.0.

## Protocol Editor Settings

### Adding a protocol

To configure the protocol:

1. In **Config** node double-click **Protocols**.
2. To add a driver, click **+**: a new line is added.
3. Select the protocol from the **PLC** list.

The protocol configuration dialog is displayed.

ABB CoDeSys Serial

PLC Network Comm...

Alias:

BlockSize:

Timeout (ms):

Num of repeats:

Protocol type:

Routing levels:

PLC Models:

Coupler (Level 1):

Channel (Level 1):

Address (Level 1):


Coupler (Level 2):

Channel (Level 2):

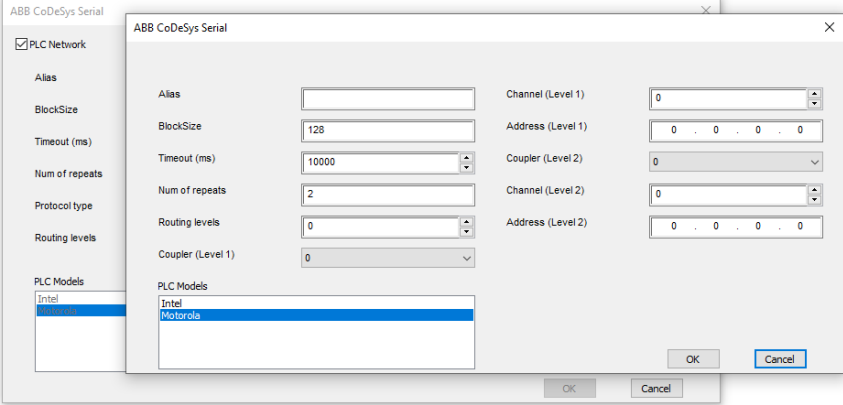
Address (Level 2):

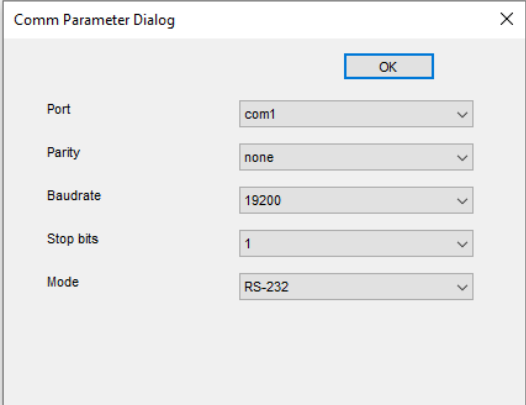
OK Cancel


Element	Description
<b>Alias</b>	Name identifying nodes in network configurations. The name will be added as a prefix to each tag name imported for each network node.
<b>BlockSize</b>	The max block size supported by your controller (limit is 1024 kB).
<b>Timeout (ms)</b>	Time delay in milliseconds between two retries in case of missing response from the server device.
<b>Num of repeats</b>	Number of times a certain message will be sent to the controller before reporting the communication error status.
<b>Protocol type</b>	Two different protocol types available: <ul style="list-style-type: none"><li>• <b>Serial_RS232</b>: corresponds to the standard 3S driver.</li></ul>

Element	Description
	<ul style="list-style-type: none"> <li>• <b>ABB_RS232_AC</b>: implements a specific variation of the standard Level 2 protocol with the additional use of a routing driver. Normally used to connect to PLCs via other PLCs acting as gateways.</li> </ul> <p> The ABB_RS232_AC protocol type requires the proper settings of the following additional parameters:</p> <ul style="list-style-type: none"> <li>• Routing Levels</li> <li>• Coupler (Level 1)</li> <li>• Channel (Level 1)</li> <li>• Address (Level 1)</li> <li>• Coupler (Level 2)</li> <li>• Channel (Level 2)</li> <li>• Address (Level 2)</li> </ul> <p>For detailed information see <i>AC500 and Control Builder</i> documentation, chapter <i>Programming interfaces to the AC500 used by the Control Builder</i>.</p>
<b>PLC Models</b>	The list allows selecting the PLC model you are going to connect to. The selection will influence the data range offset per each data type according to the specific PLC memory resources.

Element	Description
---------	-------------

<p><b>PLC Network</b></p>	<p>The protocol allows the connection to multiple controllers. To set-up multiple connections, check "PLC network" checkbox and enter the node ID per each slave you need to access.</p> 
---------------------------	---

<p><b>Comm...</b></p>	<p>If clicked displays the communication parameters setup dialog.</p> 
-----------------------	--

Element	Description
<p><b>Port</b></p>	<p>Serial port selection.</p> <ul style="list-style-type: none"> <li>• <b>COM1</b>: device PLC port.</li> <li>• <b>COM2</b>: computer/printer port.</li> </ul>
<p><b>Parity, Baudrate, Stop bits</b></p>	<p>Serial line parameters.</p> <p> <b>Parity must be set to none for AC500.</b></p>
<p><b>Mode</b></p>	<p>Serial port mode. Available modes:</p> <ul style="list-style-type: none"> <li>• <b>RS-232</b></li> <li>• <b>RS-485</b> (2 wires)</li> <li>• <b>RS-422</b> (4 wires)</li> </ul>

## Adding an alias name to a protocol

Tag names must be unique at project level, however, the same tag names might need to be used for different controller nodes (for example when the HMI device is connected to two devices running the same application).

When creating a protocol you can add an alias name that will be added to tag names imported for this protocol.

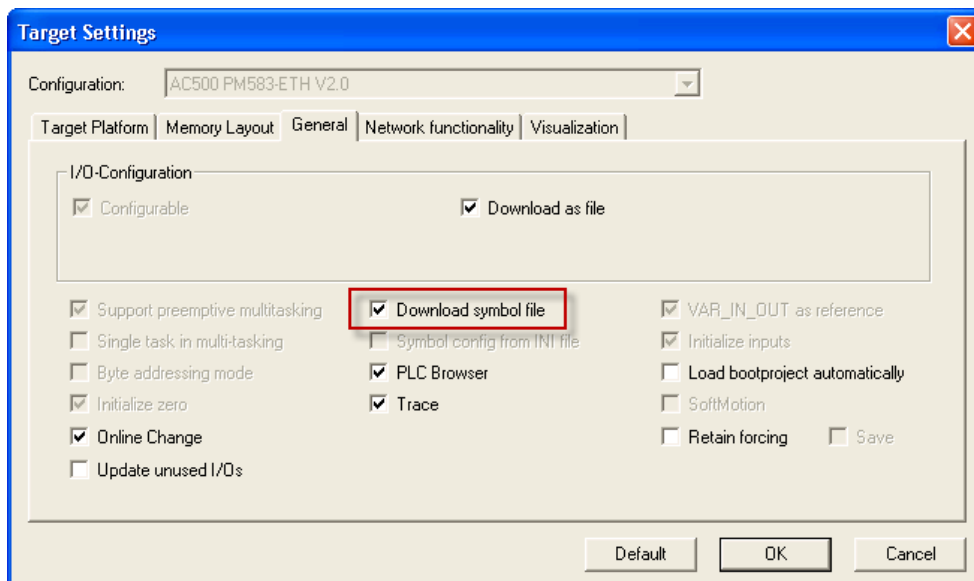
In the example, the connection to a certain controller is assigned the name **Node1**. When tags are imported for this node, all tag names will have the prefix **Node1** making each of them unique at the network/project level.

The screenshot shows the 'Tags' window in CODESYS with a list of tags. A red box highlights the tag 'Node1/Water\_Level'. A 'Network' dialog box is open, showing the option 'Select Network node id' selected. Below this is a table with columns 'Slave Id', 'Model', and 'Alias'. The 'Alias' column contains 'Node1' and 'Node2'. A red box highlights the 'Node1' entry in the 'Alias' column. The 'Water\_Level' tag name is also highlighted in a red box in the tag list below the dialog.

**i** Note: Aliasing tag names is only available for imported tags. Tags added manually in the Tag Editor cannot have the Alias prefix in the tag name. The Alias string is attached at the time of tag import. If you modify the Alias string after the tag import has been completed, there will be no effect on names already present in the dictionary. When the Alias string is changed and tags are re-imported, all tags will be re-imported with the new prefix string.

## CODESYS software settings

When you create the project in CODESYS V2, select **Download symbol file** (*Target Settings > General*).

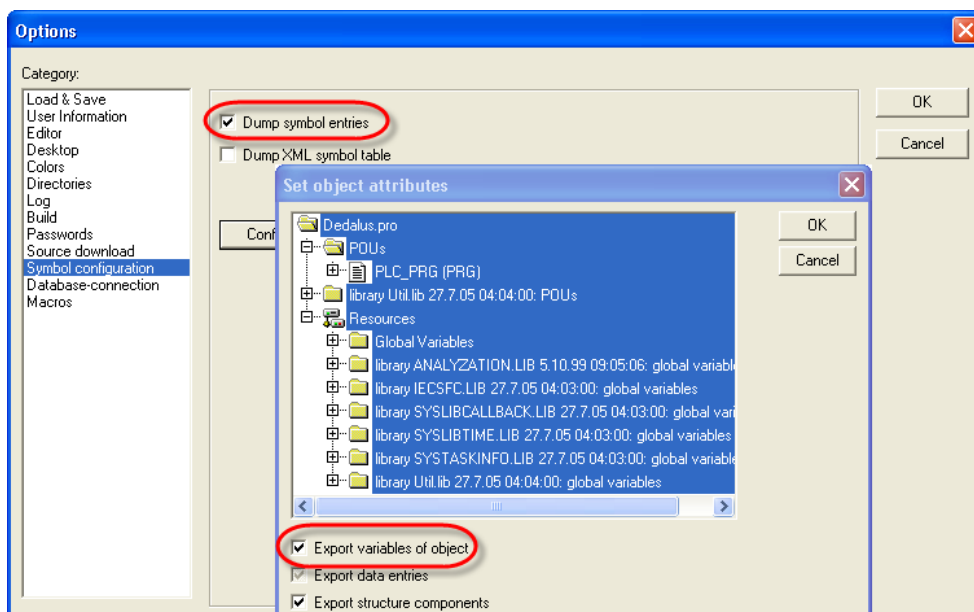


Note: ABB CODESYS Serial driver supports the automatic symbol file (SDB) upload from the controller; any change in the tag offset due to new compilation on PLC software side does not require a symbol file re-import. The Tag file has to be re-imported only in case of tag renaming or addition of new tags.

## Exporting tags from the controller

When configuring PLC using the manufacturer's configuration software, enable Symbol file (file with .sym extension) creation under the CODESYS programming software:

1. In the **Project** menu, click **Options**.
2. Select **Symbol configuration**.
3. Select **Dump symbol entries**.
4. Click **Configure symbol file**: the **Set object attributes** dialog is displayed.
5. Select **Export variables of object**.
6. Click **OK**.




## Importing tags

You may import tags from a .sym file exported from a controller. See "My first project" section in the main manual.

## Data types

The import module supports variables of standard data types and user defined data types.

<b>Supported data types</b>	<ul style="list-style-type: none"> <li>• BOOL</li> <li>• WORD</li> <li>• DWORD</li> <li>• INT</li> <li>• UINT</li> <li>• UDINT</li> <li>• DINT</li> <li>• STRING*</li> <li>• REAL</li> <li>• TIME</li> <li>• DATE &amp; TIME</li> </ul> <p>and 1-dimensional ARRAY of the types above. See "Programming concepts" section in the main manual.</p> <p> Note *: String length for a STRING variable in PLC should be max 80 characters. Declare a STRING variable either with a specified size (str: STRING(35) or default size (str: STRING) which is 80 characters.</p>
<b>Unsupported data types</b>	<ul style="list-style-type: none"> <li>• LWORD</li> <li>• LINT</li> <li>• LREAL</li> </ul>

## Communication status

Current communication status can be displayed using system variables. See "System Variables" section in the main manual.

Codes supported for this communication driver:

Error	Cause and action
Symbols file not present	Check Symbol file and download again the PLC program
"tag" not present in Symbols files	Check if the Tag is present into the PLC project
Time out on Acknowledge	Controller didn't send acknowledge
Time out on last Acknowledge	Controller didn't send last acknowledge
Time out on data receiving	Controller didn't reply with data
Connection timeout	Device not connected

---

# ABB IRC5

The ABB IRC5 robot controller communication driver has been designed for communication to the ABB Robotics robot controller family IRC5. This version supports communication towards the IRC5 robot controller I/O system (input and output signals).

The communication driver cannot meet any hard real-time demands for a number of reasons:

- It executes on a non-real-time operating system.
- Communication is performed with TCP/IP over a network.
- The actual controller might have more high-priority tasks to perform.

A minimal response time in the order of 10-100ms can be expected.

## Implementation details

An IRC5 robot controller system uses input and output signals to control the process. Signals can be digital, analog, or group signal. Such signals are accessible using the PanelBuilder tool.

Signal changes in the robot system are often significant. There are many scenarios where end-users need notification of these changes.

In the IRC5 controller manual mode, a signal value can be modified only if the signal Access Level is ALL and the FlexPendant isn't connected. If not, the controller has to be in auto mode. Use the IRC5 robot controller tools RobotStudio or FlexPendant to change the access level of a signal.

A PanelBuilder application acts as remote client in comparison with FlexPendant which is a local client.

Remote clients do not have all the privileges of a local client, especially in manual mode. For example, in manual mode the FlexPendant can start and stop IRC5 robot controller programs by using system inputs. This is only available for a remote client in automatic mode.

A remote client has the possibility to monitor and access several robot controllers from another location.

## Safety of personnel



**WARNING: A robot is heavy and extremely powerful regardless of its speed. A pause or long stop in movement can be followed by a fast hazardous movement. Even if a pattern of movement is predicted, a change in operation can be triggered by an external signal resulting in an unexpected movement.**

**All safety regulations must be followed when entering safeguarded space. Make sure you are familiar with the safety regulations as described in the IRC5 operating manuals.**

## ABB IRC5 driver

*The IRC5 driver is supported starting from version PB610 V1.90.0.778.*

HMI devices can be connected to ABB robot networks as clients using this communication driver.

The ABB IRC5 driver provides easy handling of the connections to the ABB robot controllers providing specific support for tag import facilities.

The minimum requirement is the operating system version (BSP version) on the HMI device. Refer to the software user manual to know the BSP version of your HMI device.



Products	Change index	Date of production	BSP version
CP650, CP660, CP675	B2 or newer	WK10 2013 (1103) or later	V2.80 or newer
CP620, CP630, CP635	B2 or newer	WK10 2013 (1103) or later	V1.76 or newer
CP651, CP661, CP665, CP676	A0		V1.76 or newer

### Requirements for IRC5

- Minimum RobotWare Version: RW 5.07
- PC Interface Option must be set
- IP mask of the CP600 has to be the same as in IRC5: e.g. 255.255.255.0
- IP Address must not be in the range of 192.168.125.xxx ~ 192.168.130.xxx

## Protocol Editor Settings

### Adding a protocol

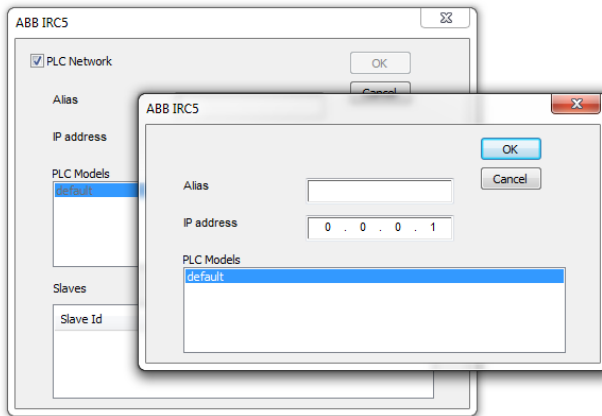
To configure the protocol:

1. In **Config** node double-click **Protocols**.
2. To add a driver, click **+**: a new line is added.
3. Select the protocol from the **PLC** list.

The protocol configuration dialog is displayed.

Element	Description
<b>Alias</b>	Name to be used to identify nodes in network configurations. The name will be added as a prefix to each tag name imported for each network node
<b>IP address</b>	Ethernet IP address of the controller.

Element	Description
<b>PLC Models</b>	Only one model type is currently available.
<b>PLC Network</b>	IP address for all controllers in multiple connections. <b>PLC Network</b> must be selected to enable multiple connections.

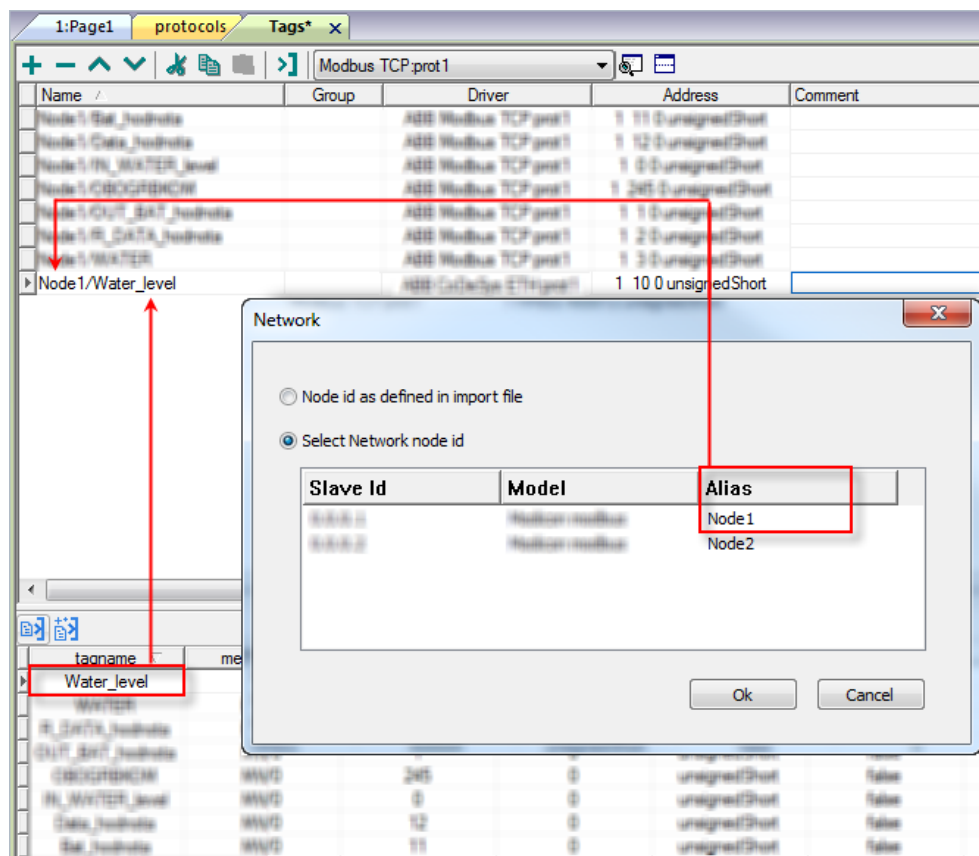


## Adding an alias name to a protocol

Tag names must be unique at project level, however, the same tag names might need to be used for different controller nodes (for example when the HMI device is connected to two devices running the same application).

When creating a protocol you can add an alias name that will be added to tag names imported for this protocol.

In the example, the connection to a certain controller is assigned the name **Node1**. When tags are imported for this node, all tag names will have the prefix **Node1** making each of them unique at the network/project level.



**i** Note: Aliasing tag names is only available for imported tags. Tags added manually in the Tag Editor cannot have the Alias prefix in the tag name. The Alias string is attached at the time of tag import. If you modify the Alias string after the tag import has been completed, there will be no effect on names already present in the dictionary. When the Alias string is changed and tags are re-imported, all tags will be re-imported with the new prefix string.

## Exporting tags from the controller

ABB Robotics controllers programming tool can generate a .cfg symbol file.

The import module supports variables of the following standard data types.

- boolean
- unsignedInt
- float

**i** Note: Only I/O signals can be used and only output signals of the robot controller can be modified by the CP600 device. They can only be modified as long as the IRC5 controller is in automatic mode or the FlexPendant is not connected and the signal access level is ALL. Input signals can only be monitored.

## Importing tags

You may import tags from a .cfg file exported from a controller. See "My first project" section in the main manual.

---

## Communication status

Current communication status can be displayed using system variables. See "System Variables" section in the main manual.

Codes supported by this communication driver:

Error	Cause	Action
<b>Can't find the node x.x.x.x "</b>	A request is not replied within the specified timeout period.	Check if the controller is connected and properly configured to get network access.
<b>I/O signal reading error</b>	The device did received a response with invalid format or contents from the controller.	Check if the data programmed in the project are consistent with the controller resources.
<b>I/O signal quality not good</b>	The device did receive a response from the controller with poor signal quality.	-
<b>Error requesting mastership</b>	The device has no permissions to write data into the controller.	-

# ABB IRC5 Linux

The ABB IRC5 Linux robot controller communication driver has been designed for communication to the ABB Robotics robot controller family IRC5 from a CP600-Pro or CP600-eCo Panel (with Linux OS).

This version supports communication towards the IRC5 robot controller I/O system (input and output signals) and to variables in the RAPID user system module.

The communication driver cannot meet any hard real-time demands for a number of reasons:

- It executes on a non-real-time operating system.
- Communication is performed with TCP/IP over a network.
- The actual controller might have more high-priority tasks to perform.

A minimal response time in the order of 10-100ms can be expected.

## Implementation details

An IRC5 robot controller system uses input and output signals to control the process. Signals can be digital, analog, or group signal. Such signals are accessible using the PanelBuilder tool.

With the ABB IRC5 Linux driver also variables which are defined in the system module "User" can be accessed.

Signal changes in the robot system are often significant. There are many scenarios where end-users need notification of these changes.

In the IRC5 controller manual mode, a signal value can be modified only if the signal Access Level is given. Use the IRC5 robot controller tools RobotStudio or FlexPendant to change the access level of a signal.

A PanelBuilder application acts as remote client in comparison with FlexPendant which is a local client.

A remote client has the possibility to monitor and access several robot controllers from another location.

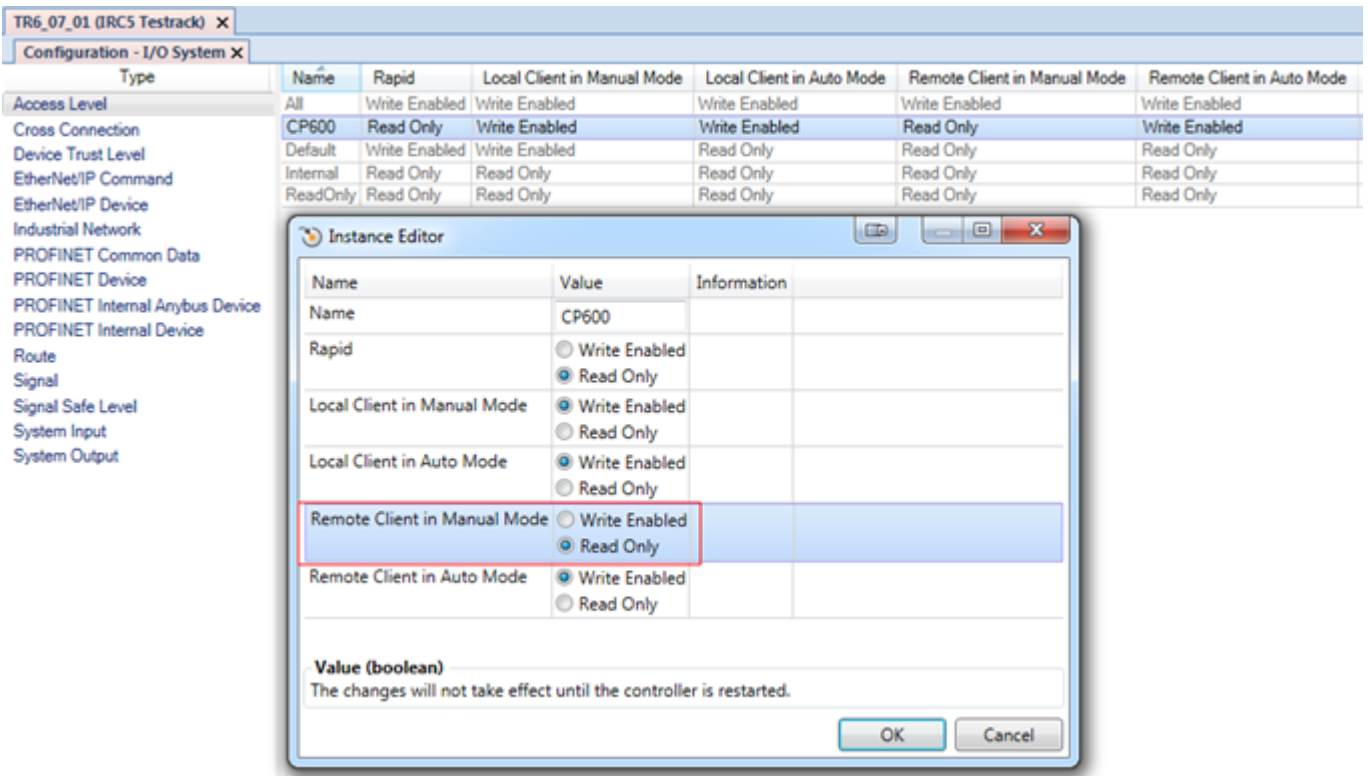
Remote clients do not have all the privileges of a local client, especially in manual mode, depending on their IO configuration. For example, in manual mode the FlexPendant can start and stop IRC5 robot controller programs by using system inputs. This is only available for a remote client in automatic mode.

Therefore it is strongly recommended to set their access level property "Remote client in manual mode" to "Read Only" for all signals used in the CP600.

Otherwise it would be possible to control the robot from CP600, even if the IRC5 is in manual mode and a person is acting next to the robot using the FlexPendant for control.

To achieve this, configure a new "Access Level Rule" (e.g. "CP600") with following settings:

- Local Client in Manual Mode: Write Enabled
- Local Client in Auto Mode: Write Enabled
- Remote Client in Manual Mode: Read Only
- Remote Client in Auto Mode: Write Enabled



Then assign this Access Level (e.g. CP600) to all signals that will be used in the CP600 panel

Type	Name	Type of Signal	Assigned to	Signal Ident	Device Mappi	Categc	Access Level	Default V
Access Level	AS1	Digital Input	PANEL	Automatic St	13	safety	ReadOnly	0
Cross Connection	AS2	Digital Input	PANEL	Automatic St	14	safety	ReadOnly	0
Device Trust Level	AUTO1	Digital Input	PANEL	Automatic M	5	safety	ReadOnly	0
EtherNet/IP Command	AUTO2	Digital Input	PANEL	Automatic M	6	safety	ReadOnly	0
EtherNet/IP Device	CH1	Digital Input	PANEL	Run Chain 1	22	safety	ReadOnly	0
Industrial Network	CH2	Digital Input	PANEL	Run Chain 2	23	safety	ReadOnly	0
PROFINET Common Data	CP600_AI0	Analog Input			N/A		CP600	0
PROFINET Device	CP600_AI1	Analog Input			N/A		CP600	0
PROFINET Internal Anybus Device	CP600_AI2	Analog Input			N/A		CP600	0
PROFINET Internal Device	CP600_AI3	Analog Input			N/A		CP600	0
Route	CP600_AI4	Analog Input			N/A		CP600	0
Signal	CP600_AI5	Analog Input			N/A		CP600	0
Signal Safe Level	CP600_AI6	Analog Input			N/A		CP600	0
System Input	CP600_AI7	Analog Input			N/A		CP600	0
System Output	CP600_AO0	Analog Output			N/A		CP600	0
	CP600_AO1	Analog Output			N/A		CP600	0
	CP600_AO2	Analog Output			N/A		CP600	0
	CP600_AO3	Analog Output			N/A		CP600	0
	CP600_AO4	Analog Output			N/A		CP600	0
	CP600_AO5	Analog Output			N/A		CP600	0
	CP600_AO6	Analog Output			N/A		CP600	0


Then the IO signals can only be modified by the CP600 if the IRC5 controller is in automatic mode.


The RAPID system variables also can only be modified by the CP600 if the IRC5 controller is in automatic mode.

Note: If you change the panel type from a CP600-Pro or CP600-eCo panel to another one or vice versa you need to change the protocol type from “ABB IRC5” to “ABB IRC5 Linux” or vice versa. After this change a new import of the EIO.cfg file has to be performed.

Be aware that only the “ABB IRC5 Linux” driver has the possibility to access RAPID system user variables.

## Safety of personnel

 **WARNING: A robot is heavy and extremely powerful regardless of its speed. A pause or long stop in movement can be followed by a fast hazardous movement. Even if a pattern of movement is predicted, a change in operation can be triggered by an external signal resulting in an unexpected movement.**

 **All safety regulations must be followed when entering safeguarded space. Make sure you are familiar with the safety regulations as described in the IRC5 operating manuals.**

## ABB IRC5 Linux driver

*The IRC5 Linux driver is supported starting from version PB610 V2.8.0.364.*

HMI devices can be connected to ABB robot networks as clients using this communication driver.

The ABB IRC5 Linux driver provides easy handling of the connections to the ABB robot controllers providing specific support for tag import facilities.

The minimum requirement is the operating system version (BSP version) on the HMI device. Refer to the software user manual to know the BSP version of your HMI device.

Products	BSP version
CP600-eCo: CP604, CP607, CP610	V 307 or newer
CP600-Pro: CP6605, CP6607, CP6610, CP6615, CP6621, CP6407, CP6410, CP6415	V 304 or newer
CP600-2nd Gen: CP6407, CP6410, CP6415	V 538 or newer

### Requirements for IRC5 Linux

- Minimum RobotWare Version: RW 6.x – (tested with RW6.07 and RW8.0)
- PC Interface Option must be set
- IP mask of the CP600 has to be the same as in IRC5: e.g. 255.255.255.0
- IP Address must not be in the range of 192.168.125.xxx ~ 192.168.130.xxx

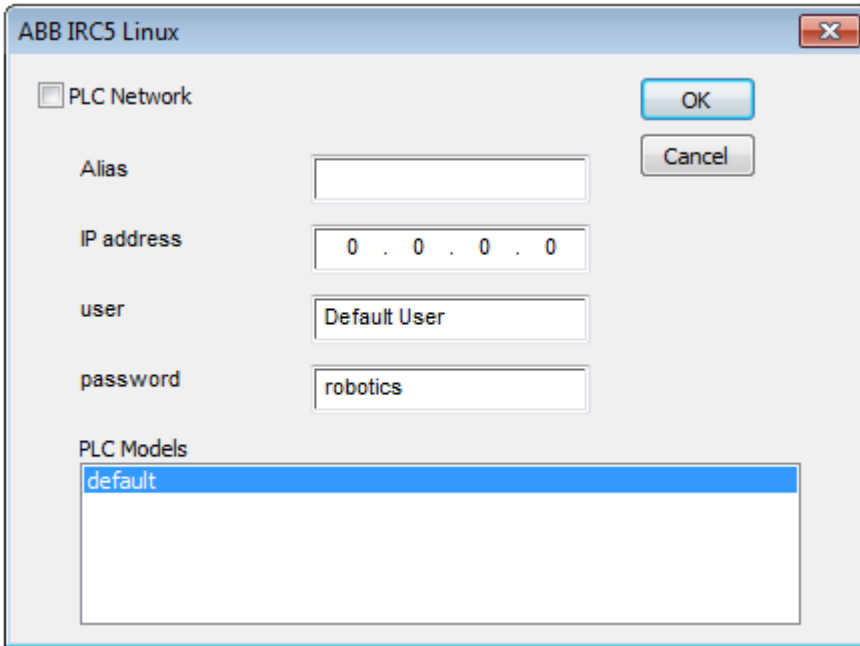
## Protocol Editor Settings

### Adding a protocol

To configure the protocol:

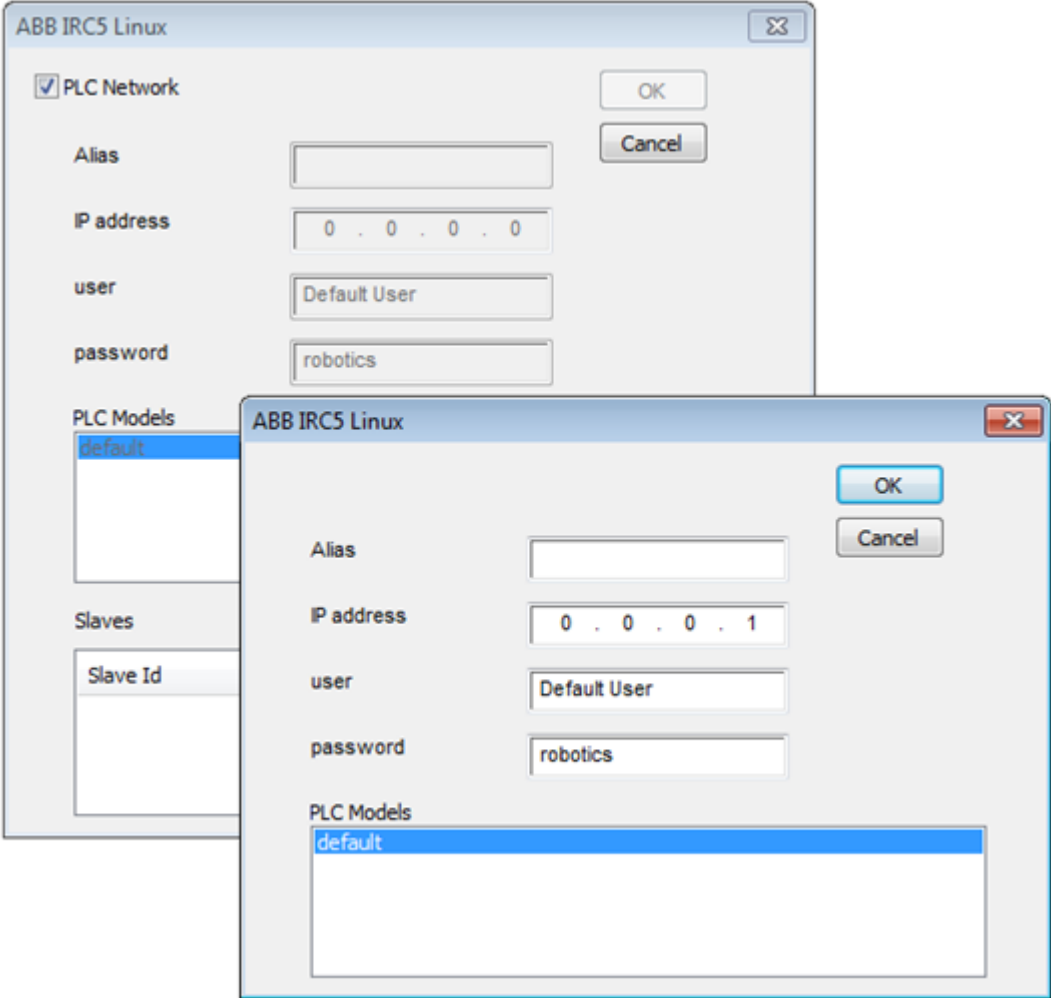
1. In **Config** node double-click **Protocols**.
2. To add a driver, click **+**: a new line is added.
3. Select the protocol from the **PLC** list.

The protocol configuration dialog is displayed.



Element	Description
<b>Alias</b>	Name to be used to identify nodes in network configurations. The name will be added as a prefix to each tag name imported for each network node
<b>IP address</b>	Ethernet IP address of the controller.
<b>user</b>	Keep the “Default User”
<b>password</b>	Keep the password “robotics”



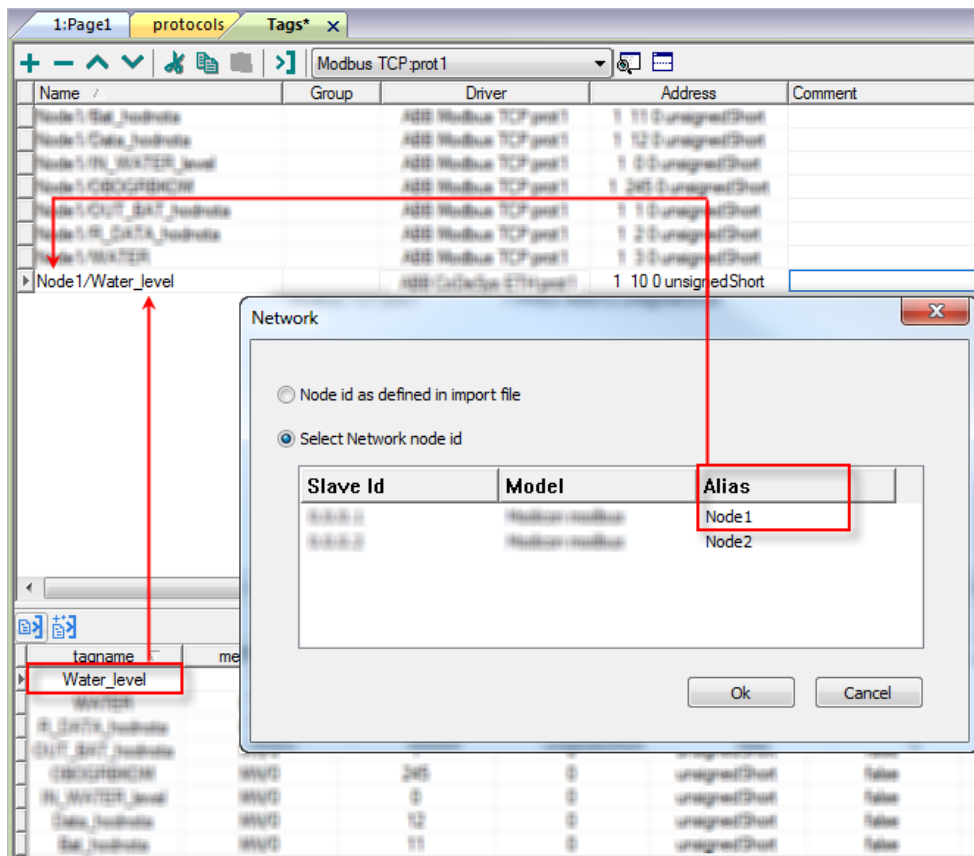
Element	Description
<b>PLC Models</b>	Only one model type is currently available.
<b>PLC Network</b>	<p>IP address for all controllers in multiple connections. <b>PLC Network</b> must be selected to enable multiple connections.</p> 

## Adding an alias name to a protocol

Tag names must be unique at project level, however, the same tag names might need to be used for different controller nodes (for example when the HMI device is connected to two devices running the same application).

When creating a protocol you can add an alias name that will be added to tag names imported for this protocol.

In the example, the connection to a certain controller is assigned the name **Node1**. When tags are imported for this node, all tag names will have the prefix **Node1** making each of them unique at the network/project level.



Note: Aliasing tag names is only available for imported tags. Tags added manually in the Tag Editor cannot have the Alias prefix in the tag name. The Alias string is attached at the time of tag import. If you modify the Alias string after the tag import has been completed, there will be no effect on names already present in the dictionary. When the Alias string is changed and tags are re-imported, all tags will be re-imported with the new prefix string.

## Exporting tags from the controller

ABB Robotics controllers programming tool can generate a .cfg symbol file. (Mostly named EIO.cfg)

The import module supports variables of the following standard data types.

- boolean
- int
- unsignedInt
- float
- double



Note: In the EIO.cfg import only I/O signals can be used. They can only be modified if the signal access level is "Write Enable" for Automatic Mode. See "Implementation details" chapter.

## Importing tags

You may import tags from a .cfg file exported from a controller. See "My first project" section in the main manual.

This will not include the RAPID system user tags. See "Accessing RAPID tags" chapter.

The variables can be read periodically asking their value to robot controller. To reduce the network traffic it is possible to "subscribe" the variable that means to ask to the controller to update the value 'on change'. The type of reading is defined for each variable via the **Subscribe** checkbox. Additionally it is possible to define the "priority" of subscribed variables. **Priority** can assume values of "high" "medium" or "low". Robot controller can accept a limited number of high priority subscriptions but it assures a fast update of value (< 1sec). Other values are update by some background process and updated value can be delayed.



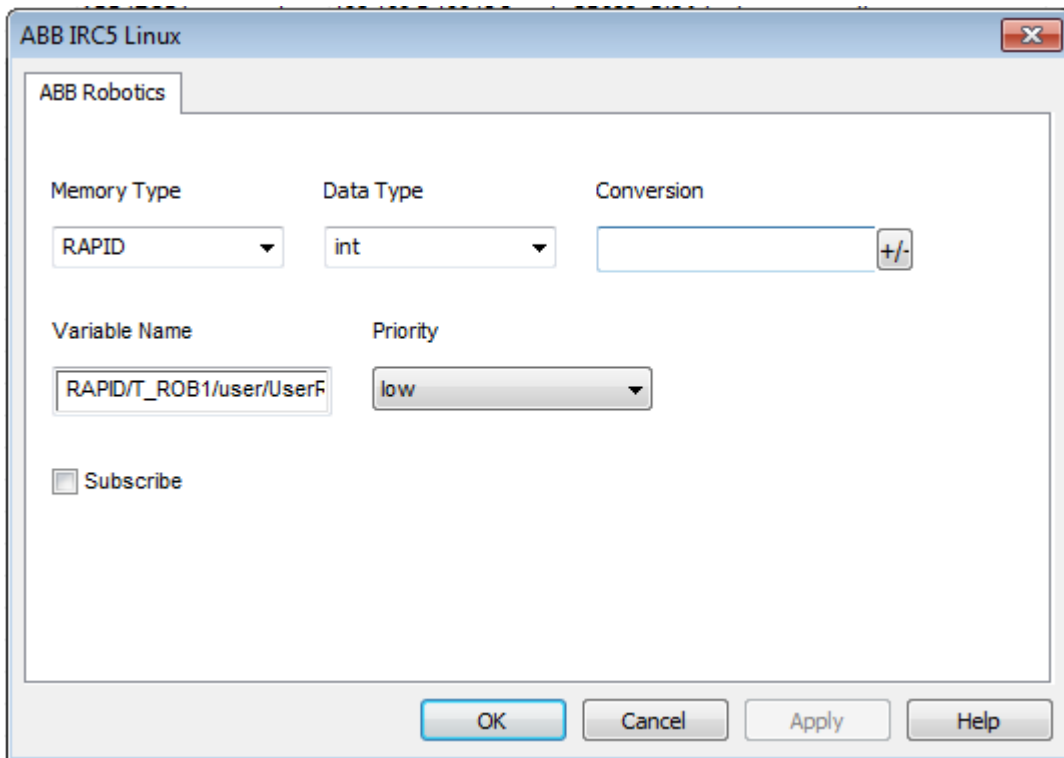
**WARNING:** Subscription is a powerful tool but it must be used carefully: in case of many variables changing value frequently, it can cause saturation of the communication channel.

## Accessing RAPID Tags

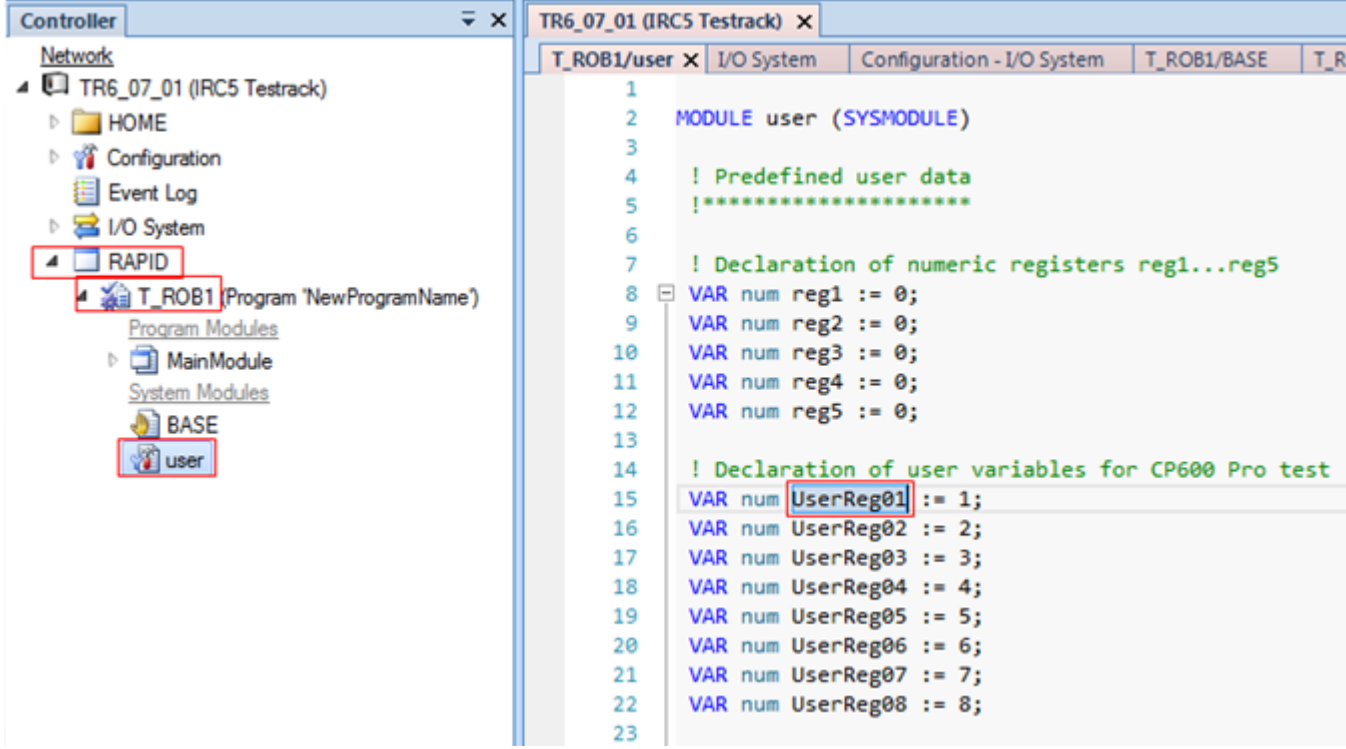

It is possible to access Tags that were created in the system module "user".

Those tags must be created manually in the Tag editor.

Add a new Tag under the ABB IRC5 Linux protocol



Element	Description
<b>Memory Type</b>	Select "RAPID"
<b>Data Type</b>	Select the type – for 32 bit tags, "double" is recommended
<b>Conversion</b>	Not needed
<b>Variable Name</b>	Use the path to the variable in screenshot in following format: <b>RAPID/T_ROB1/user/UserReg01</b>

Element	Description
	
<b>Priority</b>	<p>The variables can be read periodically asking their value to robot controller. To reduce the network traffic it is possible to "subscribe" the variable that means to ask to the controller to update the value 'on change'. The type of reading is defined for each variable via the <b>Subscribe</b> checkbox. Additionally it is possible to define the "priority" of subscribed variables. <b>Priority</b> can assume values of "high" "medium" or "low". Robot controller can accept a limited number of high priority subscriptions but it assures a fast update of value (&lt; 1sec). Other values are update by some background process and updated value can be delayed.</p>
<b>Subscribe</b>	<p> <b>WARNING: Subscription is a powerful tool but it must be used carefully: in case of many variables changing value frequently, it can cause saturation of the communication channel.</b></p>

It is possible to access simple scalar variables as well as fields of structured variables. To access a field of a structure an additional notation must be added in the end of the variable name: {index1, index2, index3, ...}. Each element corresponds to one level of nesting in the structure. Indexes starts from 1

i.e.

RAPID/T\_ROB1/testmode/gripper in a structured variable. Its full value is:

```
[FALSE, [[0,0,0], [0.94,0,0.383,0]], [5, [23,0,75], 1,0,0,0], 0,0,0]]
```

To get the values of fields the following notation must be used:

```
RAPID/T_ROB1/testmode/gripper{1} = FALSE
```

```
RAPID/T_ROB1/testmode/gripper{2,1,1} = 0
```

```
RAPID/T_ROB1/testmode/gripper{2,1,2} = 0
```

```

RAPID/T_ROB1/testmode/gripper{2,1,3} = 0
RAPID/T_ROB1/testmode/gripper{2,2,1} = 0.94
RAPID/T_ROB1/testmode/gripper{2,2,2} = 0
RAPID/T_ROB1/testmode/gripper{2,2,3} = 0.383
RAPID/T_ROB1/testmode/gripper{2,2,4} = 0
RAPID/T_ROB1/testmode/gripper{3,1} = 5
RAPID/T_ROB1/testmode/gripper{3,2,1} = 23
RAPID/T_ROB1/testmode/gripper{3,2,2} = 0
RAPID/T_ROB1/testmode/gripper{3,2,3} = 75
RAPID/T_ROB1/testmode/gripper{3,3} = 1
RAPID/T_ROB1/testmode/gripper{3,4} = 0
RAPID/T_ROB1/testmode/gripper{3,5} = 0
RAPID/T_ROB1/testmode/gripper{3,6} = 0
RAPID/T_ROB1/testmode/gripper{4} = 0
RAPID/T_ROB1/testmode/gripper{5} = 0
RAPID/T_ROB1/testmode/gripper{6} = 0

```

Each tag must correspond to a scalar field of the structure.

This kind of tag does not allow subscription

## Communication status

Current communication status can be displayed using system variables. See "System Variables" section in the main manual.

Codes supported by this communication driver:

Error	Cause	Action
<b>Can't find the node x.x.x.x "</b>	A request is not replied within the specified timeout period.	Check if the controller is connected and properly configured to get network access.
<b>I/O signal reading error</b>	The device did received a response with invalid format or contents from the controller.	Check if the data programmed in the project are consistent with the controller resources.
<b>I/O signal quality not good</b>	The device did receive a response from the controller with poor signal quality.	-
<b>Error requesting mastership</b>	The device has no permissions to write data into the controller.	-

# ABB Modbus RTU

The HMI devices can be connected to a Modbus network as the network master using this driver.

This specific implementation of the Modbus RTU driver provides easy handling of the connections to ABB controllers providing specific support for PLC models and tag import facilities.

## Implementation details

The ABB Modbus RTU implementation supports only a subset of the Modbus standard RTU function codes.

Code	Function	Description
01	Read Coil Status	Reads multiple bits in the device Coil area
02	Read Input Status	Read the ON/OFF status of the discrete inputs (1x reference) in the slave
03	Read Holding Registers	Read multiple Registers
04	Read Input Registers	Reads the binary contents of input registers (3x reference) in the slave
05	Force Single Coil	Forces a single Coil to either ON or OFF
06	Preset Single Register	Presets a value in a Register
16	Preset Multiple Registers	Presets value in multiple Registers



Note: Communication speed with controllers is supported up to 115200 baud.



Note: Floating point data format is IEEE standard compliant.

## Protocol Editor Settings

### Adding a protocol

To configure the protocol:

1. In **Config** node double-click **Protocols**.
2. To add a driver, click **+**: a new line is added.
3. Select the protocol from the **PLC** list.

The protocol configuration dialog is displayed.

ABB Modbus RTU
✕

PLC Network
 Comm...
OK
Cancel

Alias

Node ID

Timeout(ms)

delay

Num of repeats

Transmission Mode

PLC Models

ABB AC31 Series 90

ABB PM554, PM564

ABB PM556, PM566

ABB PM571, PM572

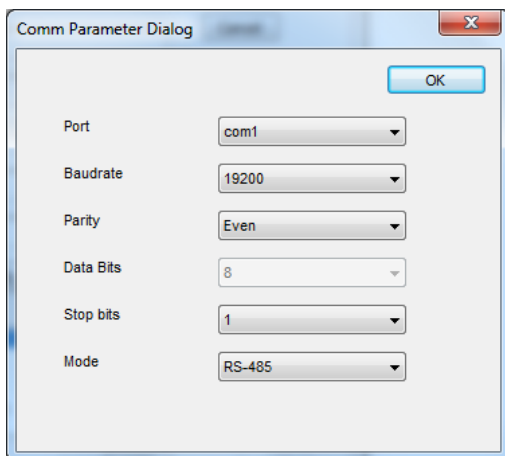
ABB PM59x, PM58x, PM573

ABB PM56xx

Element	Description
<b>Alias</b>	Name identifying nodes in network configurations. The name will be added as a prefix to each tag name imported for each network node.
<b>Node ID</b>	Modbus node of the slave device.
<b>Timeout (ms)</b>	Time delay in milliseconds between two retries in case of missing response from the server device.
<b>delay</b>	Time delay in milliseconds between the end of the last received frame and the starting of a new request. If set to 0, the new request will be issued as soon as the internal system is able to reschedule it.
<b>Num of repeats</b>	Number of times a certain message will be sent to the controller before reporting the communication error status.
<b>Transmission Mode</b>	<ul style="list-style-type: none"> <li>• <b>RTU</b>: use RTU mode</li> <li>• <b>ASCII</b>: use ASCII mode</li> </ul> <div style="margin-top: 10px;">  Note: When PLC network is active, all nodes will be configured with the same Transmission Mode.         </div>
<b>PLC Models</b>	PLC model you are going to connect to. The selection influences the data range offset per each data type according to the specific PLC memory resources.
<b>Comm...</b>	If clicked displays the communication parameters setup dialog.



Element	Description
---------	-------------



Element	Parameter
<b>Port</b>	Serial port selection. <ul style="list-style-type: none"> <li>• <b>COM1</b>: device PLC port.</li> <li>• <b>COM2</b>: computer/printer port.</li> </ul>
<b>Baudrate, Parity, Data Bits, Stop bits</b>	Serial line parameters.
<b>Mode</b>	Serial port mode. Available modes: <ul style="list-style-type: none"> <li>• <b>RS-232</b>.</li> <li>• <b>RS-485</b> (2 wires).</li> <li>• <b>RS-422</b> (4 wires).</li> </ul>

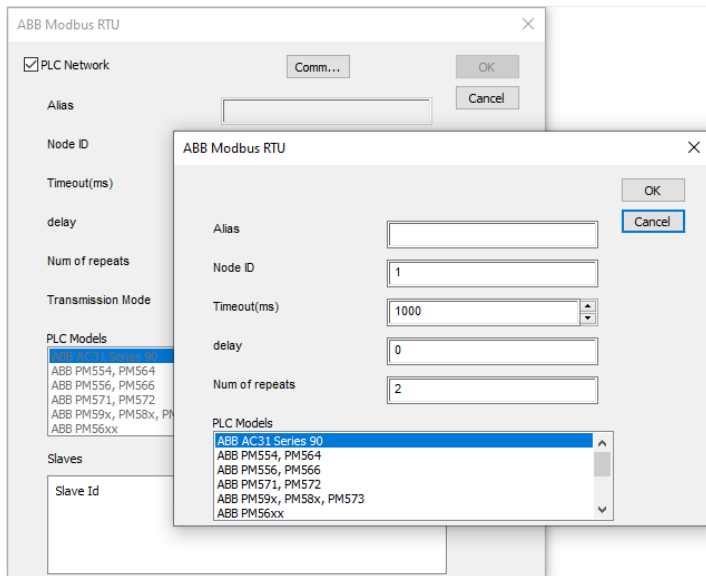
When using the controllers:

- ABB NextMove ES / ESB-2
- ABB e100 Motion Product
- ABB e150 Motion Product

make sure that Parity has been set to "None"

<b>PLC Network</b>	The protocol allows the connection of multiple controllers to one operator panel. <b>PLC Network</b> must be selected to enable multiple connections.
--------------------	---

Element	Description
---------	-------------



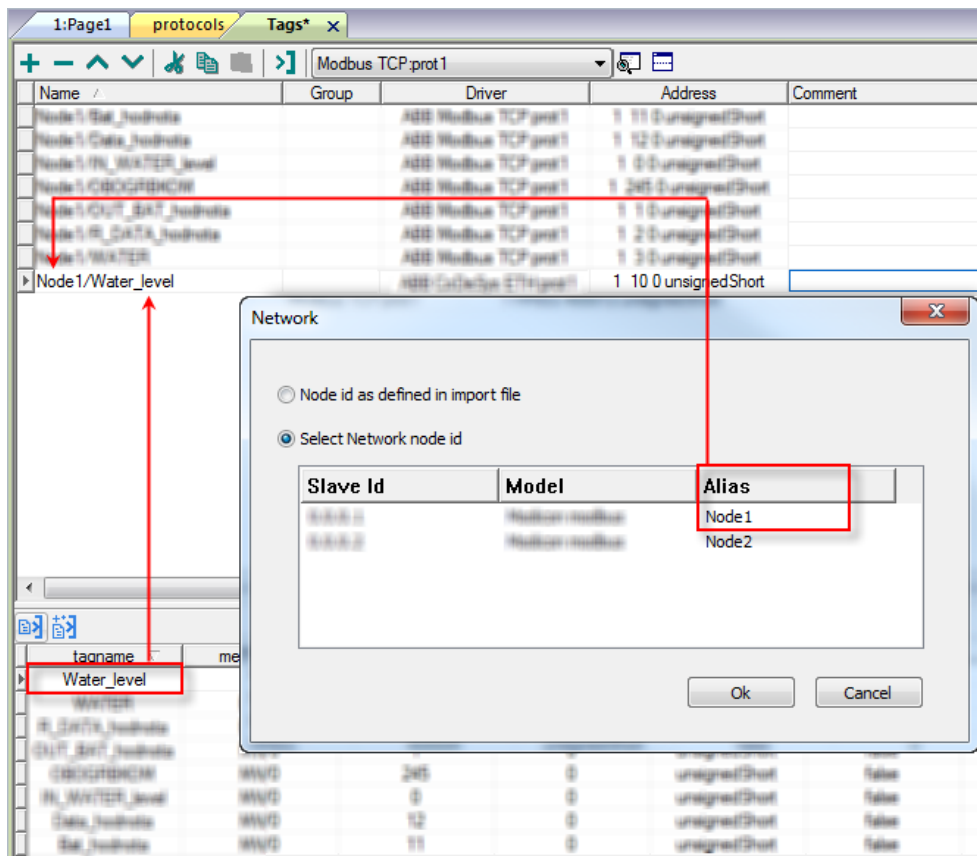
Note: PLC model **Pluto Safety PLC** is available for compatibility reasons. If you need to connect to this PLC model, please select **ABB Pluto** protocol.

## Adding an alias name to a protocol

Tag names must be unique at project level, however, the same tag names might need to be used for different controller nodes (for example when the HMI device is connected to two devices running the same application).

When creating a protocol you can add an alias name that will be added to tag names imported for this protocol.

In the example, the connection to a certain controller is assigned the name **Node1**. When tags are imported for this node, all tag names will have the prefix **Node1** making each of them unique at the network/project level.



**i** Note: Aliasing tag names is only available for imported tags. Tags added manually in the Tag Editor cannot have the Alias prefix in the tag name. The Alias string is attached at the time of tag import. If you modify the Alias string after the tag import has been completed, there will be no effect on names already present in the dictionary. When the Alias string is changed and tags are re-imported, all tags will be re-imported with the new prefix string.

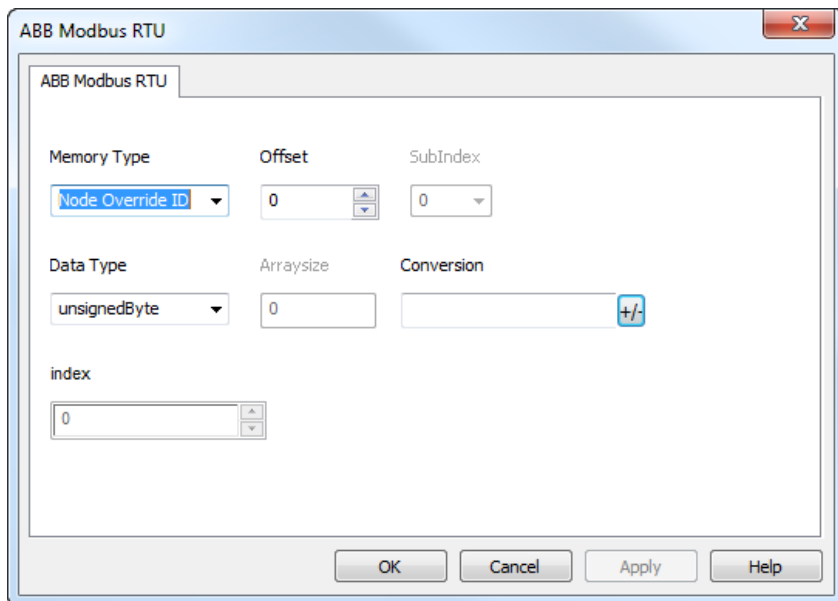
### Node Override ID (master devices)

The protocol provides the special data type Node Override ID which allows you to change the node ID of the target controller at runtime. This memory type is an unsigned byte.

The node Override ID is initialized with the value of the node ID specified in the project at programming time.

Node Override ID	Modbus operation
0	Communication with the controller is stopped. In case of write operation, the request will be transmitted without waiting for a reply.
1 to 254	It is interpreted as the value of the new node ID and is replaced for runtime operation.
255	Communication with the controller is stopped; no request messages are generated.

**i** Note: Node Override ID value assigned at runtime is retained through power cycles.

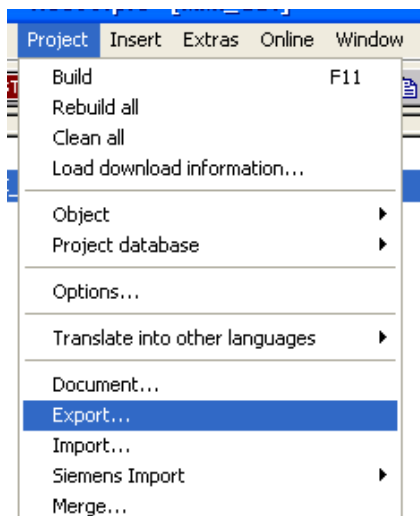


## Exporting tags from a controller

The ABB controllers programming supports tag export in .exp format.

To export tags:

Select **Project> Export...**: an .exp file will be created.



## Importing tags

You may import tags from an .exp file exported from a controller. See "My first project" section in the main manual.

## Communication status

Current communication status can be displayed using System Variables. See "System Variables" section in the main manual.

Codes supported for this communication driver:

<b>Error</b>	<b>Cause</b>	<b>Action</b>
<b>NAK</b>	The controller replies with a not acknowledge.	-
<b>Timeout</b>	A request is not replied within the specified timeout period.	Check if the controller is connected and properly configured to get network access.
<b>Line Error</b>	An error on the communication parameter setup is detected (parity, baud rate, data bits, stop bits).	Check if the communication parameter settings of the controller is compatible with the device communication setup.
<b>Invalid response</b>	The device did received a response with invalid format or contents from the controller .	Ensure the data programmed in the project are consistent with the controller resources.
<b>General Error</b>	Unidentifiable error. Should never be reported.	Contact technical support.

---

# ABB Modbus TCP

ABB Modbus TCP driver provides easy handling of the connection to the ABB controllers providing specific supports for PLC models and tag import facilities.

Various Modbus TCP-capable devices can be connected to the HMI device. To set-up your Modbus TCP device, please refer to the documentation you have received with the device.

The implementation of the protocol operates as a Modbus TCP client.

## Implementation details

The ABB Modbus TCP supports only a subset of the standard Modbus TCP function codes.

Code	Function	Description
01	Read Coil Status	Reads multiple bits in the device Coil area
02	Read Input Status	Read the ON/OFF status of the discrete inputs (1x reference) in the slave
03	Read Holding Registers	Read multiple Registers
04	Read Input Registers	Reads the binary contents of input registers (3x reference) in the slave
05	Force Single Coil	Forces a single Coil to either ON or OFF
06	Preset Single Register*	Presets a value in a Register
16	Preset Multiple Registers*	Presets value in multiple Registers

## Protocol Editor Settings

### Adding a protocol

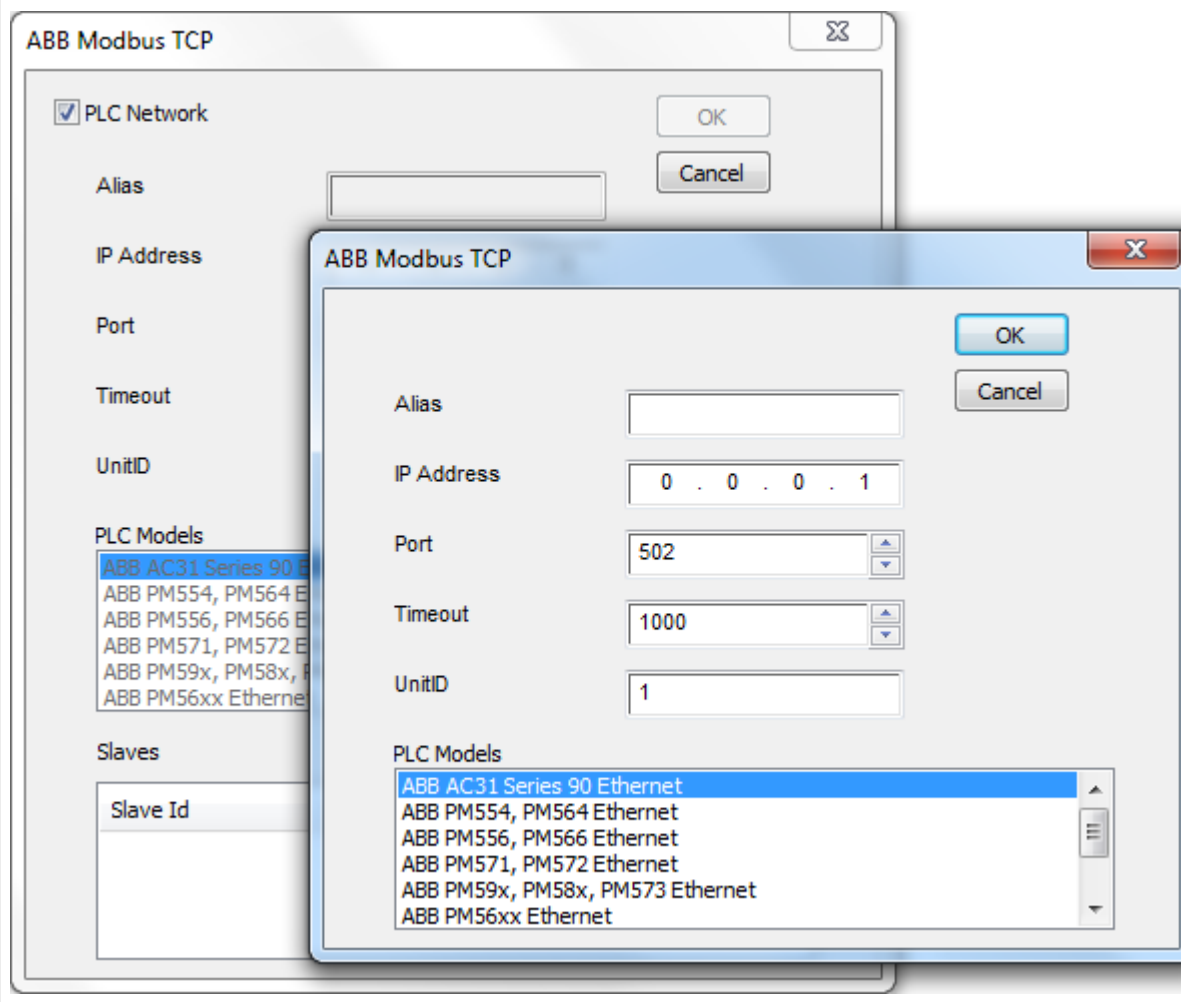
To configure the protocol:

1. In **Config** node double-click **Protocols**.
2. To add a driver, click **+**: a new line is added.
3. Select the protocol from the **PLC** list.

The protocol configuration dialog is displayed.

Element	Description
<b>Alias</b>	Name identifying nodes in network configurations. The name will be added as a prefix to each tag name imported for each network node.
<b>IP Address</b>	Ethernet IP address of the controller.
<b>Port</b>	Port number used by the Modbus TCP driver. The default value can be changed when the communication goes through routers or Internet gateways where the default port number is already in use.
<b>Timeout</b>	Time delay in milliseconds between two retries in case of missing response from the server device.
<b>UnitID</b>	Usually used when communicating over Ethernet-to-serial gateways and then interpreted as the Slave ID. This value is simply copied into the Unit Identifier field of the Modbus TCP communication frame. This is rarely used and in most cases can be left zero.

Element	Description
<b>PLC Models</b>	PLC model you are going to connect to. The selection influences the data range offset per each data type according to the specific PLC memory resources.
<b>PLC Network</b>	IP address for all controllers in multiple connections. <b>PLC Network</b> check box must be selected to enable multiple connections.



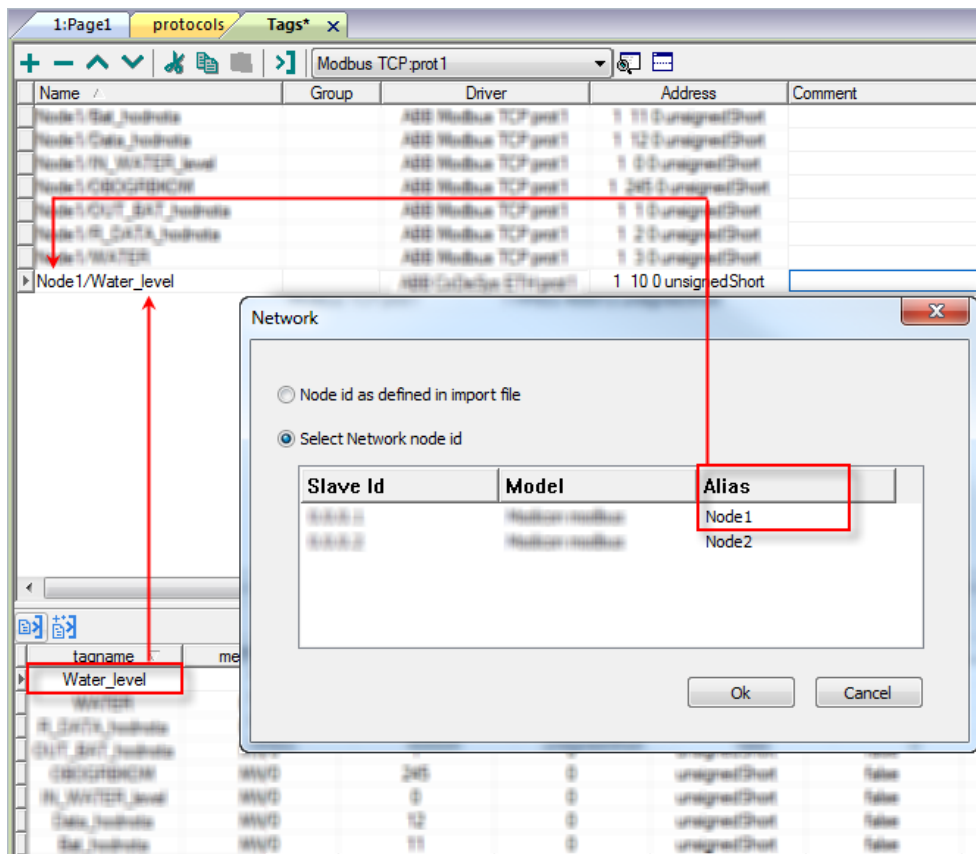
## Adding an alias name to a protocol

Tag names must be unique at project level, however, the same tag names might need to be used for different controller nodes (for example when the HMI device is connected to two devices running the same application).

When creating a protocol you can add an alias name that will be added to tag names imported for this protocol.

In the example, the connection to a certain controller is assigned the name **Node1**. When tags are imported for this node, all tag names will have the prefix **Node1** making each of them unique at the network/project level.





**i** Note: Aliasing tag names is only available for imported tags. Tags added manually in the Tag Editor cannot have the Alias prefix in the tag name. The Alias string is attached at the time of tag import. If you modify the Alias string after the tag import has been completed, there will be no effect on names already present in the dictionary. When the Alias string is changed and tags are re-imported, all tags will be re-imported with the new prefix string.

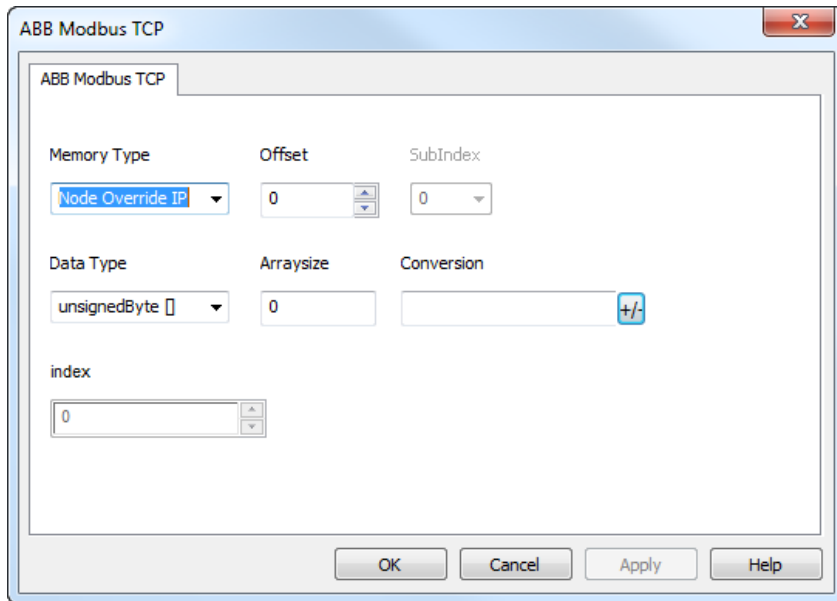
### Node Override ID (master devices)

The protocol provides the special data type Node Override ID which allows you to change the node ID of the target controller at runtime. This memory type is an unsigned byte.

The node Override ID is initialized with the value of the node ID specified in the project at programming time.

Node Override ID	Modbus operation
0	Communication with the controller is stopped. In case of write operation, the request will be transmitted without waiting for a reply.
1 to 254	It is interpreted as the value of the new node ID and is replaced for runtime operation.
255	Communication with the controller is stopped; no request messages are generated.

**i** Note: Node Override ID value assigned at runtime is retained through power cycles.

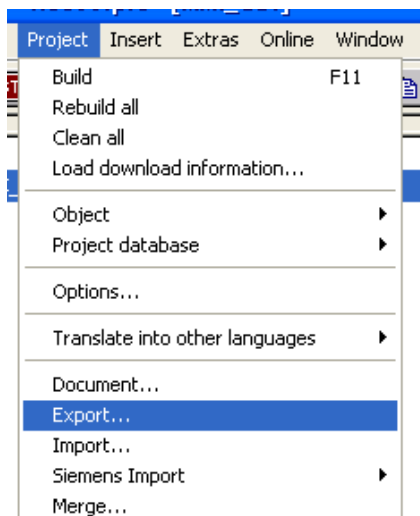


## Exporting tags from the controller

The ABB controllers programming supports tag export in .exp format.

To export tags:

Select **Project> Export...**: an .exp file will be created.



## Importing tags

You may import tags from an .exp file exported from a controller. See "My first project" section in the main manual.

## Communication status

Current communication status can be displayed using system variables. See "System Variables" section in the main manual.

Codes supported for this communication driver:

---

<b>Error</b>	<b>Cause</b>	<b>Action</b>
<b>NAK</b>	The controller replies with a not acknowledge.	-
<b>Timeout</b>	A request is not replied within the specified timeout period.	Check if the controller is connected and properly configured to get network access.
<b>Invalid response</b>	The device did received a response with invalid format or contents from the controller.	Check if the data programmed in the project are consistent with the controller resources.
<b>General Error</b>	Unidentifiable error.	Contact technical support.

---

# ABB Pluto

The HMI devices can be connected to a Modbus network as the network master using this generic driver.

This specific implementation of the Modbus RTU driver provides easy handling of the connections to the ABB controllers providing specific support for ABB Pluto Safety PLC and tag import facilities.

## Implementation details

This Modbus RTU implementation supports only a subset of the standard Modbus function codes.

Code	Function	Description
01	Read Coil Status	Reads multiple bits in the device Coil area
02	Read Input Status	Read the ON/OFF status of the discrete inputs (1x reference) in the slave
03	Read Holding Registers	Read multiple Registers
04	Read Input Registers	Reads the binary contents of input registers (3x reference) in the slave
05	Force Single Coil	Forces a single Coil to either ON or OFF
06	Preset Single Register	Presets a value in a Register
16	Preset Multiple Registers	Presets value in multiple Registers



Note: Communication speed with controllers is supported up to 115200 baud.



Note: Floating point data format is IEEE standard compliant.

## Protocol Editor Settings

### Adding a protocol

To configure the protocol:

1. In **Config** node double-click **Protocols**.
2. To add a driver, click **+**: a new line is added.
3. Select the protocol from the **PLC** list.

The protocol configuration dialog is displayed.

The driver configuration dialog is shown in the following figure.

ABB Pluto

Comm... OK Cancel

Alias

Node ID

Timeout (ms)

Delay (ms)

Num of repeats

Transmission Mode

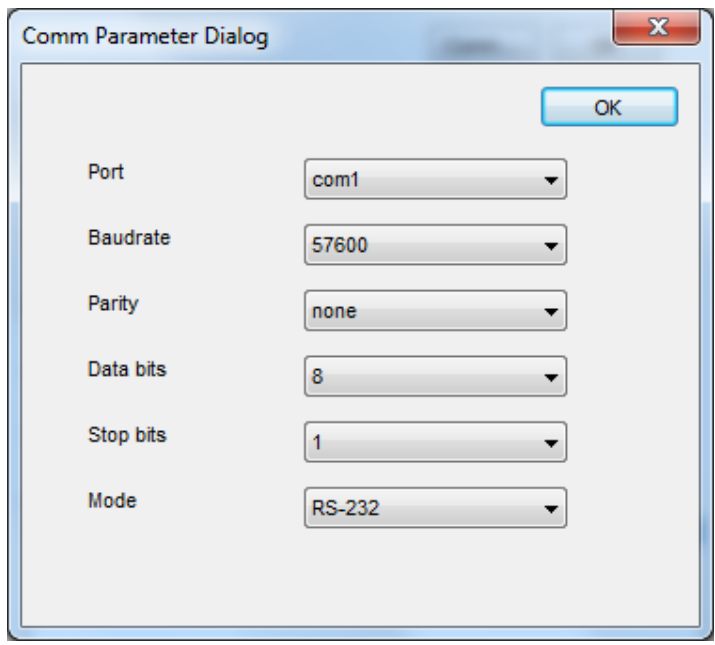
PLC Models

Pluto Safety PLC

Element	Description
<b>Alias</b>	Name identifying nodes in network configurations. The name will be added as a prefix to each tag name imported for each network node.
<b>Node ID</b>	Modbus node of the slave device.
<b>Timeout (ms)</b>	Time delay in milliseconds between two retries in case of missing response from the slave device.
<b>Delay (ms)</b>	Time delay in milliseconds between the end of the last received frame and the starting of a new request. If set to 0, the new request will be issued as soon as the internal system is able to reschedule it.
<b>Num of repeats</b>	Number of times a certain message will be sent to the controller before reporting the communication error status.  When set to 1 the panel will report the communication error if the response to the first request packet is not correct.
<b>Transmission Mode</b>	<ul style="list-style-type: none"> <li>• <b>RTU</b>: use RTU mode</li> <li>• <b>ASCII</b>: use ASCII mode</li> </ul>

Element	Description
<b>PLC Models</b>	PLC model you are going to connect to. The selection influences the data range offset per each data type according to the specific PLC memory resources.

**Comm...** If clicked displays the communication parameters setup dialog.



Element	Description
<b>Port</b>	Serial port selection. <ul style="list-style-type: none"> <li>• <b>COM1</b>: device PLC port.</li> <li>• <b>COM2</b>: computer/printer port (if available)</li> </ul>
<b>Baudrate, Parity, Data bits, Stop bits</b>	Serial line parameters.
<b>Mode</b>	Serial port mode. Available modes: <ul style="list-style-type: none"> <li>• <b>RS-232</b>.</li> <li>• <b>RS-485</b> (2 wires).</li> <li>• <b>RS-422</b> (4 wires).</li> </ul>

## Node Override ID (master devices)

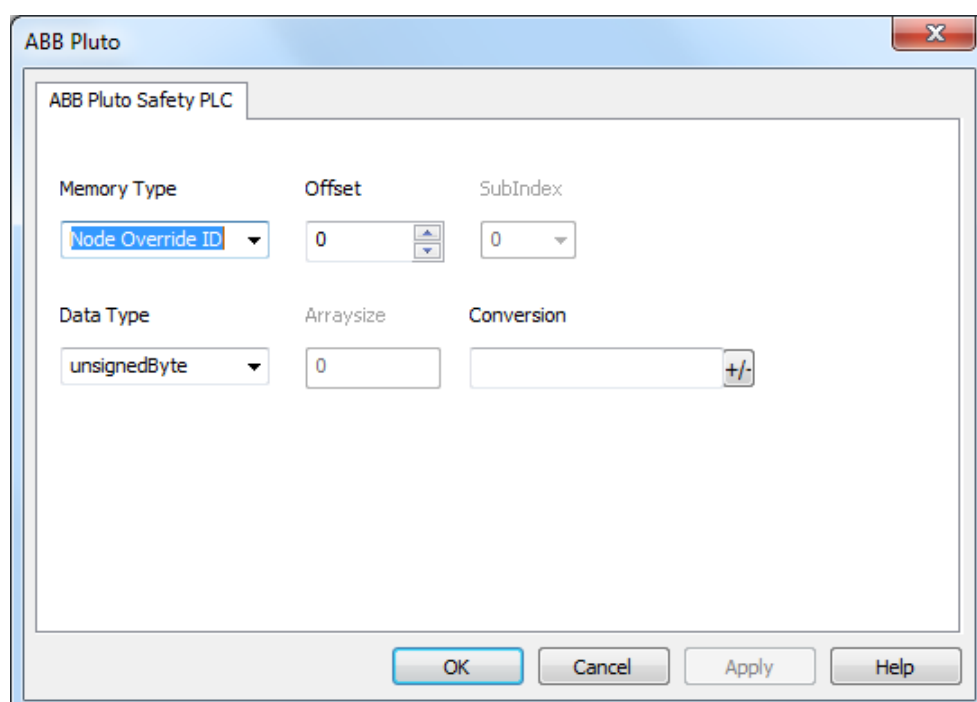
The protocol provides the special data type Node Override ID which allows you to change the node ID of the target controller at runtime. This memory type is an unsigned byte.

The node Override ID is initialized with the value of the node ID specified in the project at programming time.

Node Override ID	Modbus operation
0	Communication with the controller is stopped. In case of write operation, the request will be transmitted without waiting for a reply.
1 to 254	It is interpreted as the value of the new node ID and is replaced for runtime operation.
255	Communication with the controller is stopped; no request messages are generated.




Note: Node Override ID value assigned at runtime is retained through power cycles.

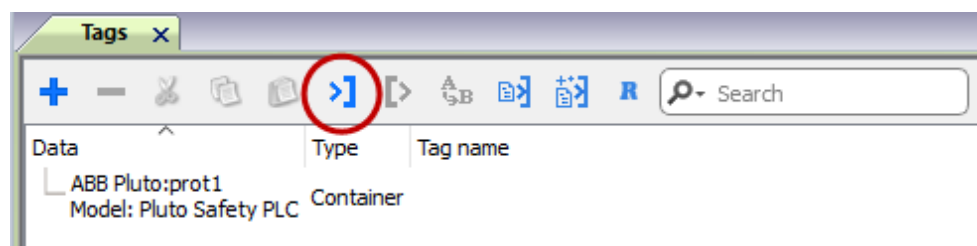


## Tag import

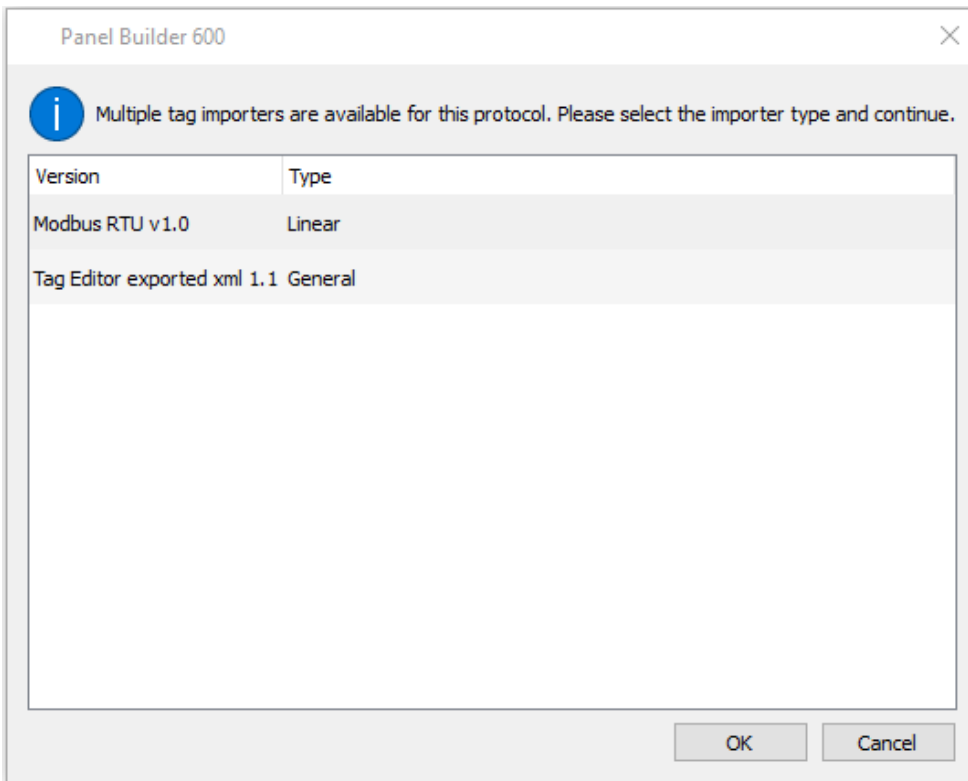
ABB Pluto driver supports tag import.


The ABB Pluto Safety PLC programming suite allows to export tags in .csv format.

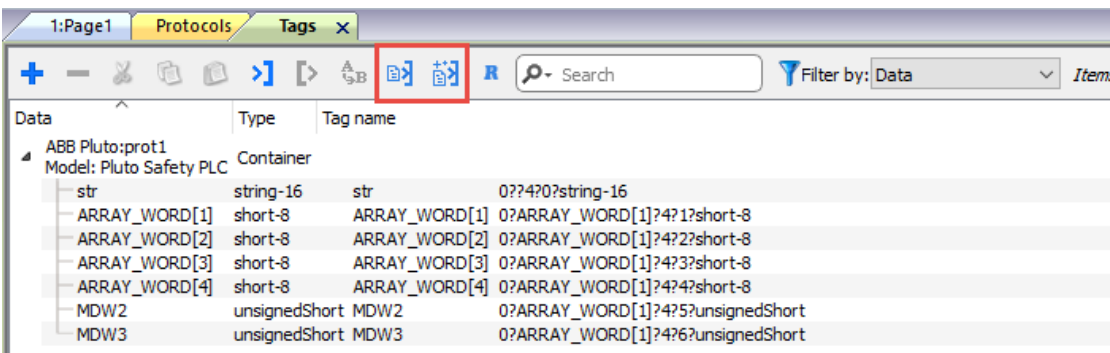
1. In the Tag Editor select the driver.
2. Click the **Import Tags**  button to start the import.



3. Locate the .csv file and confirm.



4. To import tags, select one or more tags in the .csv file and click the  **Import tag** button: tags are copied to the project.



See "My first project" section in the main manual.

## Communication status

Current communication status can be displayed using System Variables. See "System Variables" section in the main manual.

Codes supported for this communication driver:

Error	Cause	Action
<b>NAK</b>	The controller replies with a not acknowledge.	-
<b>Timeout</b>	A request is not replied within the specified	Check if the controller is connected and properly



---

<b>Error</b>	<b>Cause</b>	<b>Action</b>
	timeout period.	configured to get network access.
<b>Line Error</b>	An error on the communication parameter setup is detected (parity, baud rate, data bits, stop bits).	Check if the communication parameter settings of the controller is compatible with the device communication setup.
<b>Invalid response</b>	The device did received a response with invalid format or contents from the controller .	Ensure the data programmed in the project are consistent with the controller resources.
<b>General Error</b>	Unidentifiable error. Should never be reported.	Contact technical support.

# ABB Mint Controller HCP

This communication protocol allows the HMI devices to connect to the ABB motion and servo drive devices using the HCP and HCP2 communication protocols.

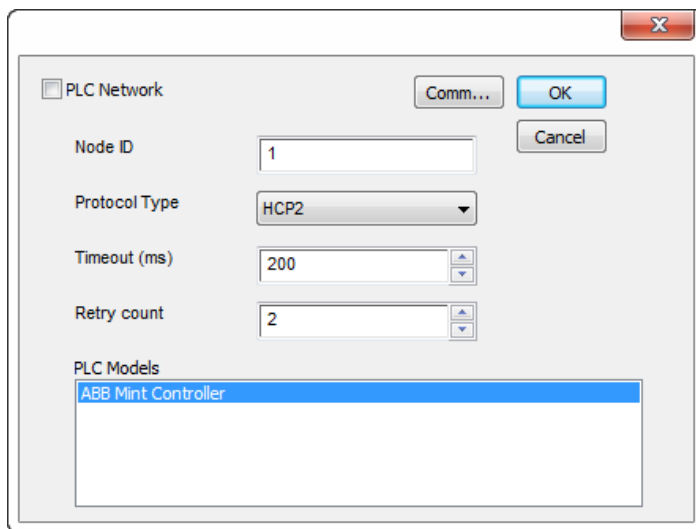
## Protocol Editor Settings

### Adding a protocol

To configure the protocol:

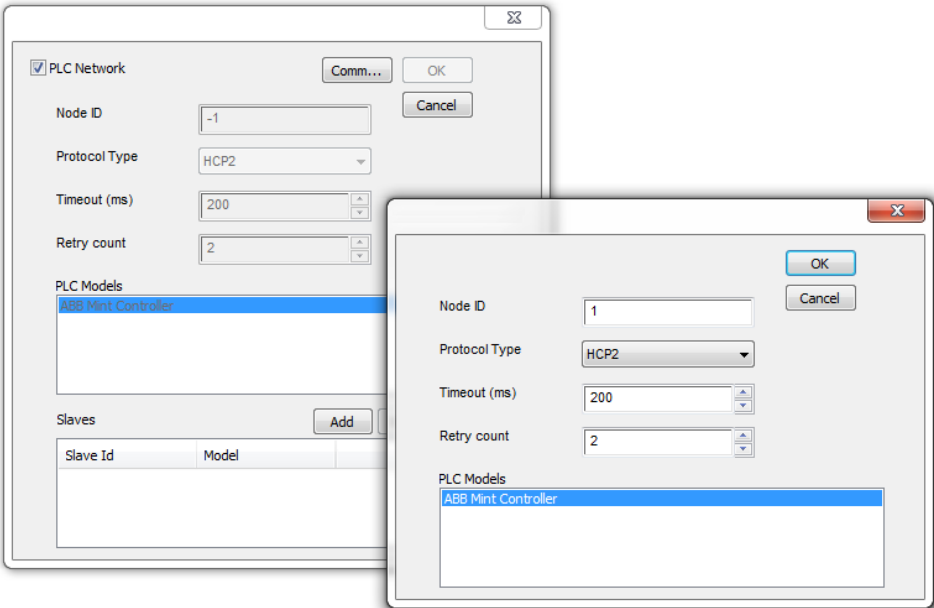
1. In **Config** node double-click **Protocols**.
2. To add a driver, click **+**: a new line is added.
3. Select the protocol from the **PLC** list.

The protocol configuration dialog is displayed.

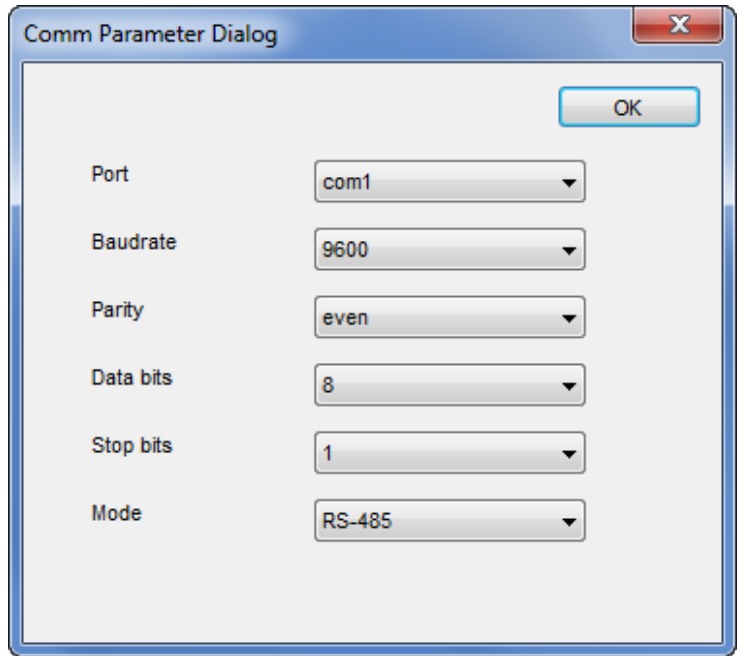


Element	Description
<b>Node ID</b>	Node ID assigned to the controller device.
<b>Protocol Type</b>	Two protocols are available: <ul style="list-style-type: none"><li>• <b>HCP</b></li><li>• <b>HCP2</b></li></ul>
<b>Timeout (ms)</b>	Time delay in milliseconds between two retries in case of missing response from the server device.
<b>Retry count</b>	Number of times a certain message will be sent to the controller before reporting the communication error status.
<b>PLC Models</b>	PLC model you are going to connect to.

Element	Description
---------	-------------

<b>PLC Network</b>	<p>The protocol allows the connection of multiple controllers to one HMI device. To set-up multiple connections, check “PLC network” checkbox and enter the node ID per each slave you need to access.</p> 
--------------------	--

<b>Comm...</b>	If clicked displays the communication parameters setup dialog.
----------------	--



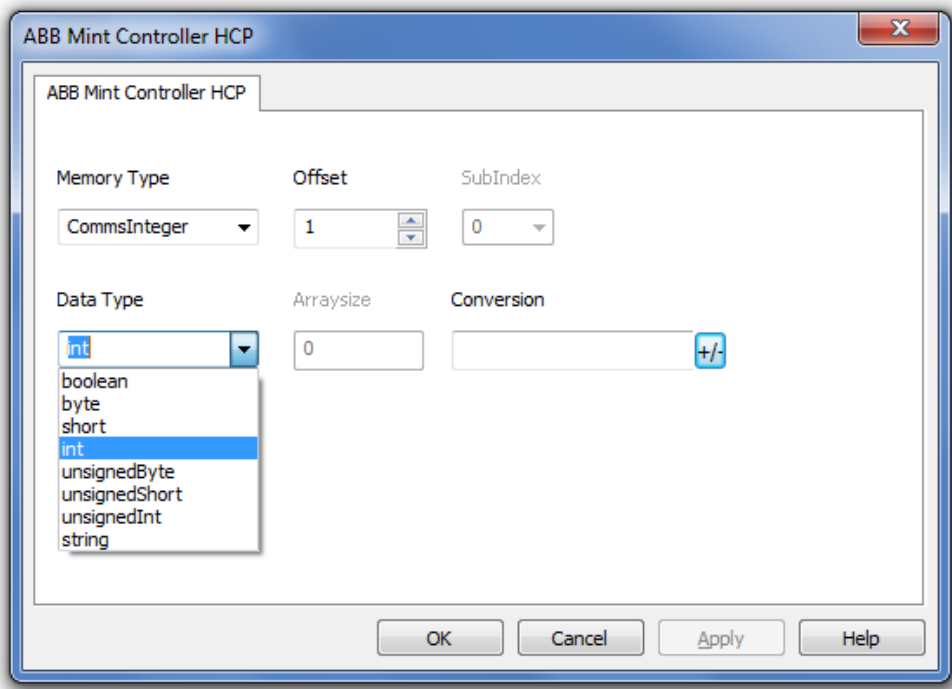
Element	Description	
	<b>Element</b>	<b>Description</b>
	<b>Port</b>	Serial port selection. <ul style="list-style-type: none"> <li>• <b>COM1</b>= device PLC port.</li> <li>• <b>COM2</b>= computer/printer port.</li> </ul>
	<b>Baudrate, Parity, Data Bits, Stop bits</b>	Serial line parameters.
	<b>Mode</b>	Serial port mode. Available modes: <ul style="list-style-type: none"> <li>• <b>RS-232</b>.</li> <li>• <b>RS-485</b> (2 wires).</li> <li>• <b>RS-422</b> (4 wires).</li> </ul>

## Data types

The ABB Mint Controller HCP driver provides the support for two Memory Types which are referring to the same physical memory area in the Mint controller:

- **Comms**: should only be used with floating point values. The Mint program on the ABB controller should use COMMS to access this data.
- **CommsInteger**: allows a variety of integer-based data types to be selected.

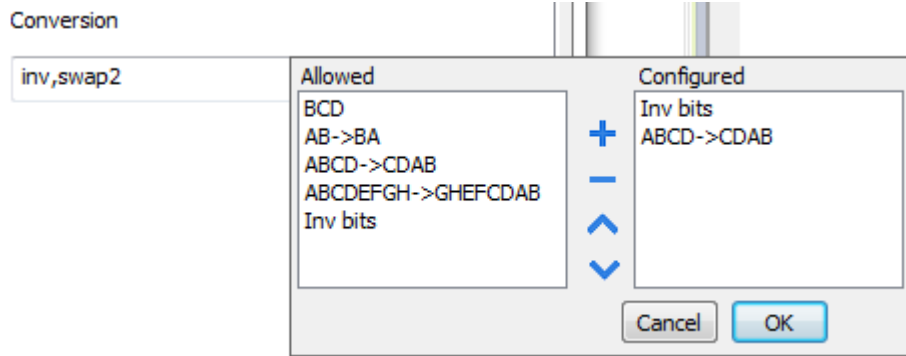
If the Mind controller program uses...	then...
COMMS keyword for a tag setup to use the Commsinteger memory type	only the bottom 23 bits will be accurate (due to floating point precision of the COMMS keyword).
COMMSINTEGER keyword for a tag setup to use the Commsinteger memory type	the value is precise for the full 32 bits.



See "Programming concepts" section in the main manual.

## Tag Conversion

Conversion to be applied to the tag.



Depending on data type selected, the list **Allowed** shows one or more conversion types.

Value	Description
<b>Inv bits</b>	<p><b>inv</b>: Invert all the bits of the tag.</p> <p><i>Example:</i>            1001 → 0110 (in binary format)            9 → 6 (in decimal format)</p>
<b>Negate</b>	<p><b>neg</b>: Set the opposite of tag value.</p> <p><i>Example:</i></p>

Value	Description
	25.36 → -25.36
<b>AB -&gt; BA</b>	<b>swapnibbles:</b> Swap nibbles in a byte. <i>Example:</i> 15D4 → 514D (in hexadecimal format) 5588 → 20813 (in decimal format)
<b>ABCD -&gt; CDAB</b>	<b>swap2:</b> Swap bytes in a word. <i>Example:</i> 9ACC → CC9A (in hexadecimal format) 39628 → 52378 (in decimal format)
<b>ABCDEFGH -&gt; GHEFCDAB</b>	<b>swap4:</b> Swap bytes in a double word. <i>Example:</i> 32FCFF54 → 54FFFC32 (in hexadecimal format) 855441236 → 1426062386 (in decimal format)
<b>ABC...NOP -&gt; OPM...DAB</b>	<b>swap8:</b> Swap bytes in a long word. <i>Example:</i> 142.366 → -893553517.588905 (in decimal format) 0 1000000110 000111001011101101100100010110100001110010101100001 → 1 10000011100 101010100001010001011011011001011011000100111101 (in binary format)
<b>BCD</b>	<b>bcd:</b> Separate byte in two nibbles, read them as decimal (from 0 to 9) <i>Example:</i> 23 → 17 (in decimal format) 0001 0111 = 23 0001 = 1 (first nibble) 0111 = 7 (second nibble)

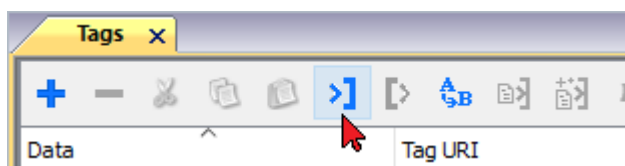
Select conversion and click +. The selected item will be added to list **Configured**.

If more conversions are configured, they will be applied in order (from top to bottom of list **Configured**).

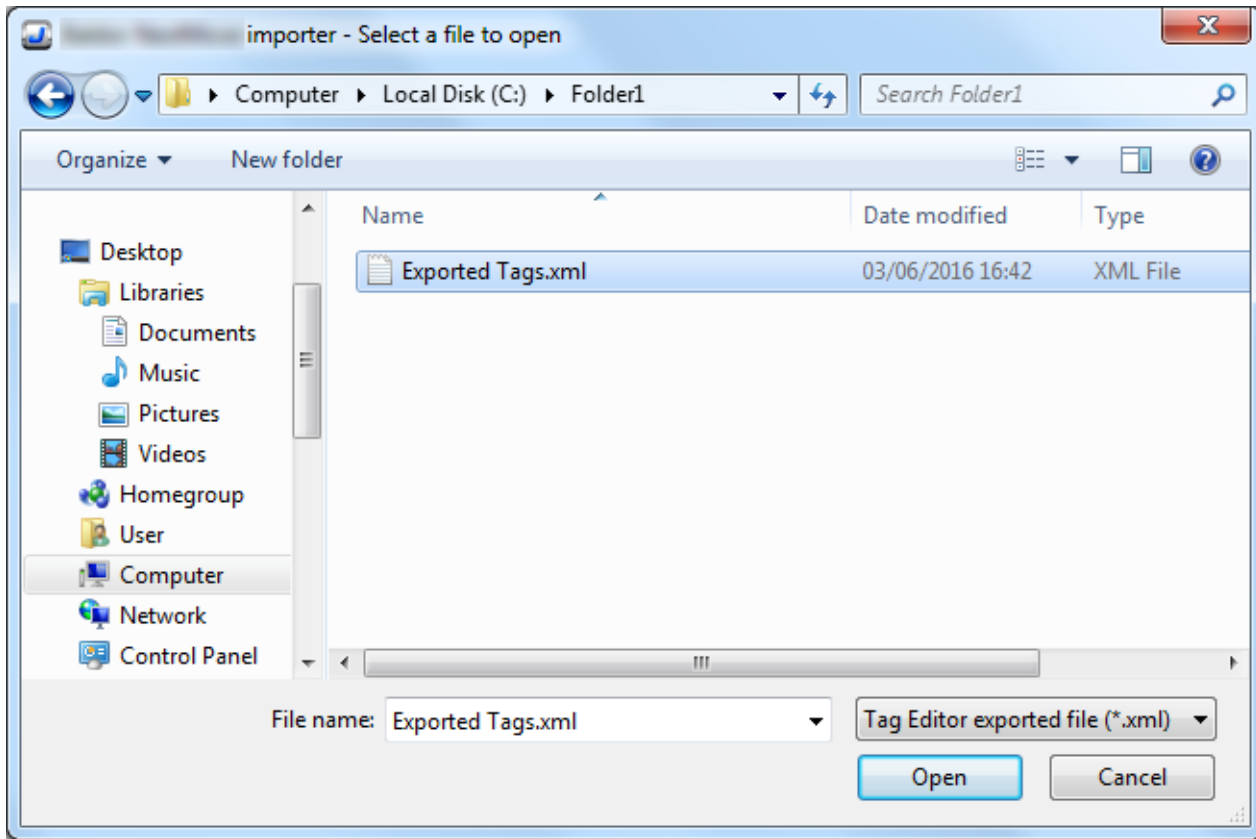
Use the arrow buttons to order the configured conversions.

## Tag Import

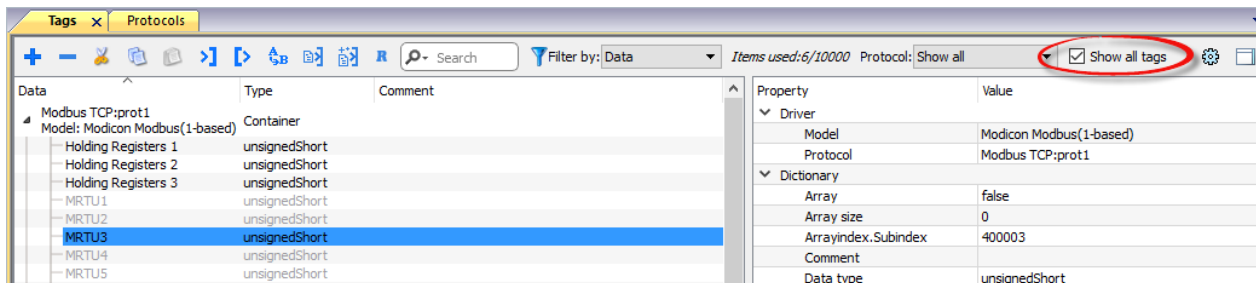
Select the driver in Tag Editor and click on the **Import Tags** button to start the importer.






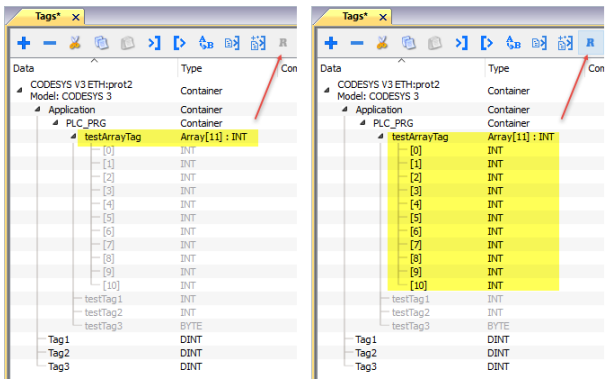
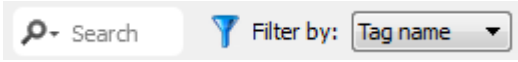
Locate the **.xml** file exported from Tag Editor and click **Open**.



Tags included in the symbol file are listed in the tag dictionary. The tag dictionary is displayed at the bottom of the screen.



Toolbar item	Description
	<p><b>Import Tag(s).</b></p> <p>Select tags to be imported and click on this icon to add tags from tag dictionary to the project</p>
	<p><b>Update Tag(s).</b></p> <p>Click on this icon to update the tags in the project, due a new dictionary import.</p>
	<p>Check this box to import all sub-elements of a tag.</p>

Toolbar item	Description
	<p>Example of both checked and unchecked result:</p> 
	<p>Searches tags in the dictionary basing on filter combo-box item selected.</p>

## Communication status

Current communication status can be displayed using system variables. See "System Variables" section in the main manual.

Codes supported by this communication driver:

Error	Cause	Action
<b>NAK</b>	The controller replies with a not acknowledge.	-
<b>Timeout</b>	A request is not replied within the specified timeout period.	Check if the controller is connected and properly configured to get network access.
<b>Line Error</b>	An error on the communication parameter setup is detected (parity, baud rate, data bits, stop bits).	Check if the communication parameter settings of the controller is compatible with the device communication setup.
<b>Invalid response</b>	The device did received a response with invalid format or contents from the controller.	Ensure the data programmed in the project are consistent with the controller resources.
<b>General Error</b>	Unidentifiable error. Should never be reported.	Contact technical support.



# BACnet

The BACnet communication driver has been designed to connect HMI devices to BACnet networks and supports IP and MS/TP communication.

The HMI device operates as a BACnet device.

## Implementation details

This implementation of the BACnet communication protocol allows integrating HMIs in a BACnet network and exchange data between HMI and other devices connected to the BACnet network. HMIs provide client capability for displaying properties of BACnet objects in real time using BACnet/IP or MS/TP network types.

BACnet communication protocol can be:

- Configured as BACnet IP: communication with BACnet devices is established over Ethernet using HMI Ethernet port;
- Configured as BACnet MS/SP: communication with BACnet devices is established over serial line, using HMI serial port;

Communication protocol configuration allows defining HMI BACnet ID and object name used to identify HMI in BACnet network.

BACnet object properties are reachable from HMI using explicit Tag configuration. A single Tag represents a single property for a BACnet object.

Using the property Present\_Value (85) in Tag configuration, the Tag will be connected to the current value of a specific object (for example in the case of analog values, it will be the measured value).

## Protocol Editor Settings

### Adding a protocol

To configure the protocol:

1. In **Config** node double-click **Protocols**.
2. To add a driver, click **+**: a new line is added.
3. Select the protocol from the **PLC** list.

The protocol configuration dialog is displayed.

BACnet

Comm...

Panel Device ID	<input type="text" value="262000"/>	Analog Value Count	<input type="text" value="0"/>
Object Name	<input type="text" value="DEV262000"/>	Binary Value Count	<input type="text" value="0"/>
Description	<input type="text" value="HMI"/>	Multi State Value Count	<input type="text" value="0"/>
Media	<input type="text" value="MS/TP"/>	Notification Class Count	<input type="text" value="0"/>
Timeout (ms)	<input type="text" value="5000"/>	IP UDP Port	<input type="text" value="47808"/>
Panel Node	<input type="text" value="1"/>	Local IP	<input type="text"/>
COV Lifetime (s)	<input type="text" value="60"/>		
<input type="checkbox"/> COV Confirmed			
Max Master	<input type="text" value="127"/>		
Max Info Frames	<input type="text" value="1"/>		
max MS/TP APDU	<input type="text" value="480"/>		
max IP APDU	<input type="text" value="1476"/>		
Time Sync Interval (s)	<input type="text" value="0"/>		
<input type="checkbox"/> Time Sync UTC			
PLC Models	<input type="text" value="default"/>		

Element	Description
<b>Panel Device ID</b>	Identifies the HMI device in the network.
<b>Object Name</b>	BACnet Object Name for the HMI device.
<b>Description</b>	HMI device description, for documentation purposes.
<b>Media</b>	Type of communication of the protocol. <ul style="list-style-type: none"> <li>• <b>MS/TP</b>: Master-Slave/Token-Passing communication (RS-485).</li> </ul>

Element	Description
	<ul style="list-style-type: none"> <li><b>IP:</b> based on standard UDP/IP communication.</li> </ul>
<b>Timeout (ms)</b>	Time delay in milliseconds between two retries in case of missing response from the BACnet device.
<b>Panel Node *</b>	MS/TP address. Physical device address on the link; it is not passed through routers.
<b>COV Lifetime (s)</b>	Desired lifetime of the subscription in seconds before the it shall be automatically cancelled.. A value of zero indicates an indefinite lifetime, without automatic cancellation.
<b>Max Master *</b>	Highest allowable address for master nodes. Must be less than or equal to 127.
<b>Max Info Frames *</b>	Maximum number of information frames the node may send before it must pass the token. Max Info Frames may have different values on different nodes and may be used to allocate more or less of the available link bandwidth to particular nodes.
<b>Max MS/TP APDU *</b>	Maximum length of APDU (Application Layer Protocol Data Unit), which means the actual packet length on BACnet network. This value cannot exceed 480 (default value).
<b>Max IP APDU **</b>	Maximum length of APDU (Application Layer Protocol Data Unit), which means the actual packet length on BACnet network. This value cannot exceed 1476 (default value).
<b>Time Sync Interval (s)</b>	Represent the interval between every time synchronization, in seconds. If left to 0, time synchronization is disabled.
<b>Time Sync UTC</b>	Option to synchronize time in UTC format. If disabled, local time format used.
<b>PLC Models</b>	Reserved for future use.
<b>Comm... *</b>	If clicked displays the communication parameters setup dialog. <div data-bbox="316 1328 922 1868" data-label="Image"> </div>

Element	Description								
	<table border="1"> <thead> <tr> <th>Element</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td><b>Port</b></td> <td>Communication port.</td> </tr> <tr> <td><b>Baudrate, Parity, Data bits, Stop bits</b></td> <td>Communication parameters.</td> </tr> <tr> <td><b>Mode</b></td> <td>Communication mode. Available modes: <ul style="list-style-type: none"> <li>• <b>RS-232</b></li> <li>• <b>RS-485</b></li> <li>• <b>RS-422</b></li> </ul> </td> </tr> </tbody> </table>	Element	Description	<b>Port</b>	Communication port.	<b>Baudrate, Parity, Data bits, Stop bits</b>	Communication parameters.	<b>Mode</b>	Communication mode. Available modes: <ul style="list-style-type: none"> <li>• <b>RS-232</b></li> <li>• <b>RS-485</b></li> <li>• <b>RS-422</b></li> </ul>
Element	Description								
<b>Port</b>	Communication port.								
<b>Baudrate, Parity, Data bits, Stop bits</b>	Communication parameters.								
<b>Mode</b>	Communication mode. Available modes: <ul style="list-style-type: none"> <li>• <b>RS-232</b></li> <li>• <b>RS-485</b></li> <li>• <b>RS-422</b></li> </ul>								
<b>Analog Value Count ***</b>	Number of Analog Value objects to be instanced in BACnet Server. Min: 0 Max: 200								
<b>Binary Value Count ***</b>	Number of Binary Value objects to be instanced in BACnet Server. Min: 0 Max: 200								
<b>Multi State Value Count ***</b>	Number of Multi State Value objects to be instanced in BACnet Server. Min: 0 Max: 200								
<b>Notification Class Count ***</b>	Number of Notifications Class objects to be instanced in BACnet Server. Min: 0 Max: 200								
<b>IP UDP Port **</b>	Port number for IP communication.								
<b>Local IP **</b>	IP Address of the network adapter to use for protocol. Not required if the device has only one Ethernet adapter.								



Note \*: Available only if media is set to **MS/TP**.



Note \*\*: Available only if media is set to **IP**.



Note \*\*\*: Check **Using BACnet Server** chapter.

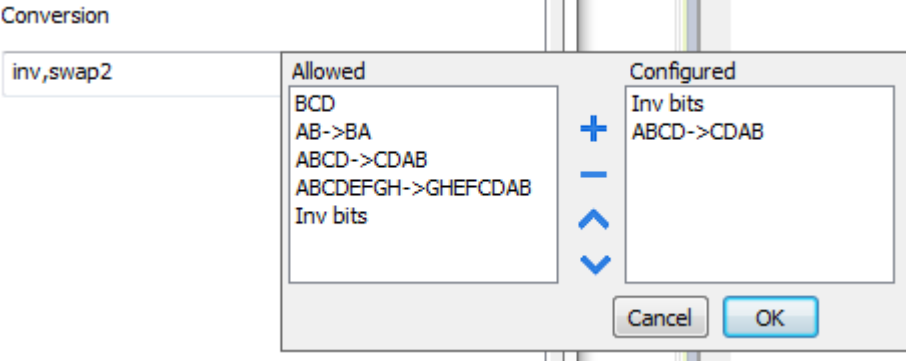
## Tag Editor Settings

Path: **ProjectView** > **Config** > double-click **Tags**

1. To add a tag, click **+**: a new line is added.
2. Select **BACnet** from the **Driver** list: the tag definition dialog is displayed.

Element	Description
<b>Object Type</b>	Type of BACnet object to be referenced. Available object types: <ul style="list-style-type: none"> <li>• <b>Device</b></li> <li>• <b>Analog Input</b></li> <li>• <b>Analog Output</b></li> <li>• <b>Analog Value</b></li> <li>• <b>Binary Input</b></li> <li>• <b>Binary Output</b></li> <li>• <b>Binary Value</b></li> <li>• <b>Multi-state Input</b></li> <li>• <b>Multi-state Output</b></li> <li>• <b>Multi-state Value</b></li> <li>• <b>Integer Value</b></li> <li>• <b>Positive Integer Value</b></li> <li>• <b>Large Analog Value</b></li> </ul>
<b>Device ID</b>	ID of the device containing the object.
<b>Data Type</b>	Data type for display presentation. Available data types: <ul style="list-style-type: none"> <li>• <b>boolean</b></li> </ul>

Element	Description																																	
	<ul style="list-style-type: none"> <li>• <b>int</b></li> <li>• <b>unsignedInt</b></li> <li>• <b>float</b></li> <li>• <b>double</b></li> <li>• <b>string</b></li> <li>• <b>binary</b></li> <li>• <b>boolean[]</b></li> </ul> <p>These data types are data types as defined in the software.</p> <p>The equivalence with BACnet data types is shown in the table:</p> <table border="1" data-bbox="252 763 1254 1541"> <thead> <tr> <th data-bbox="252 763 598 853">BACnet data type</th> <th data-bbox="598 763 786 853">Software data type</th> <th data-bbox="786 763 1254 853">Notes</th> </tr> </thead> <tbody> <tr> <td data-bbox="252 853 598 913"><b>BOOLEAN</b></td> <td data-bbox="598 853 786 913">Boolean</td> <td data-bbox="786 853 1254 913">-</td> </tr> <tr> <td data-bbox="252 913 598 974"><b>INTEGER</b></td> <td data-bbox="598 913 786 974">Int</td> <td data-bbox="786 913 1254 974">-</td> </tr> <tr> <td data-bbox="252 974 598 1034"><b>UNSIGNED_INTEGER</b></td> <td data-bbox="598 974 786 1034">unsignedInt</td> <td data-bbox="786 974 1254 1034">-</td> </tr> <tr> <td data-bbox="252 1034 598 1095"><b>REAL</b></td> <td data-bbox="598 1034 786 1095">Float</td> <td data-bbox="786 1034 1254 1095">-</td> </tr> <tr> <td data-bbox="252 1095 598 1155"><b>BIT_STRING</b></td> <td data-bbox="598 1095 786 1155">boolean-x</td> <td data-bbox="786 1095 1254 1155"><b>x = size</b></td> </tr> <tr> <td data-bbox="252 1155 598 1216"><b>CHARACTER_STRING</b></td> <td data-bbox="598 1155 786 1216">string-x</td> <td data-bbox="786 1155 1254 1216"><b>x = size</b></td> </tr> <tr> <td data-bbox="252 1216 598 1276"><b>OCTET_STRING</b></td> <td data-bbox="598 1216 786 1276">binary-x</td> <td data-bbox="786 1216 1254 1276"><b>x = size</b></td> </tr> <tr> <td data-bbox="252 1276 598 1359"><b>DATE</b></td> <td data-bbox="598 1276 786 1359">int or unsignedInt</td> <td data-bbox="786 1276 1254 1359">-</td> </tr> <tr> <td data-bbox="252 1359 598 1442"><b>TIME</b></td> <td data-bbox="598 1359 786 1442">int or unsignedInt</td> <td data-bbox="786 1359 1254 1442">-</td> </tr> <tr> <td data-bbox="252 1442 598 1541"><b>BACnetObjectIdentifier</b></td> <td data-bbox="598 1442 786 1541">int or unsignedInt</td> <td data-bbox="786 1442 1254 1541"><b>Use conversions instance and objType for proper display</b></td> </tr> </tbody> </table>	BACnet data type	Software data type	Notes	<b>BOOLEAN</b>	Boolean	-	<b>INTEGER</b>	Int	-	<b>UNSIGNED_INTEGER</b>	unsignedInt	-	<b>REAL</b>	Float	-	<b>BIT_STRING</b>	boolean-x	<b>x = size</b>	<b>CHARACTER_STRING</b>	string-x	<b>x = size</b>	<b>OCTET_STRING</b>	binary-x	<b>x = size</b>	<b>DATE</b>	int or unsignedInt	-	<b>TIME</b>	int or unsignedInt	-	<b>BACnetObjectIdentifier</b>	int or unsignedInt	<b>Use conversions instance and objType for proper display</b>
BACnet data type	Software data type	Notes																																
<b>BOOLEAN</b>	Boolean	-																																
<b>INTEGER</b>	Int	-																																
<b>UNSIGNED_INTEGER</b>	unsignedInt	-																																
<b>REAL</b>	Float	-																																
<b>BIT_STRING</b>	boolean-x	<b>x = size</b>																																
<b>CHARACTER_STRING</b>	string-x	<b>x = size</b>																																
<b>OCTET_STRING</b>	binary-x	<b>x = size</b>																																
<b>DATE</b>	int or unsignedInt	-																																
<b>TIME</b>	int or unsignedInt	-																																
<b>BACnetObjectIdentifier</b>	int or unsignedInt	<b>Use conversions instance and objType for proper display</b>																																
<b>Arraysize</b>	<ul style="list-style-type: none"> <li>• In case of array tag, this property represents the number of array elements.</li> <li>• In case of string tag, this property represents the maximum number of bytes available in the string tag.</li> </ul> <p>Note: number of bytes corresponds to number of string characters if Encoding property is set to UTF-8 or Latin1 in Tag Editor. If Encoding property is set to UCS-2BE, UCS-2LE, UTF-16BE or UTF-16LE one character requires 2 bytes.</p>																																	
<b>Conversion</b>	Conversion to be applied to the tag.																																	

Element	Description														
	<p>Conversion</p>  <p>Depending on data type selected, the list <b>Allowed</b> shows one or more conversion types.</p> <table border="1" data-bbox="220 786 1217 1899"> <thead> <tr> <th>Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td><b>Inv bits</b></td> <td><b>inv:</b> Invert all the bits of the tag. <i>Example:</i> 1001 → 0110 (in binary format) 9 → 6 (in decimal format)</td> </tr> <tr> <td><b>Negate</b></td> <td><b>neg:</b> Set the opposite of tag value. <i>Example:</i> 25.36 → -25.36</td> </tr> <tr> <td><b>AB -&gt; BA</b></td> <td><b>swapnibbles:</b> Swap nibbles in a byte. <i>Example:</i> 15D4 → 514D (in hexadecimal format) 5588 → 20813 (in decimal format)</td> </tr> <tr> <td><b>ABCD -&gt; CDAB</b></td> <td><b>swap2:</b> Swap bytes in a word. <i>Example:</i> 9ACC → CC9A (in hexadecimal format) 39628 → 52378 (in decimal format)</td> </tr> <tr> <td><b>ABCDEFGH -&gt; GHEFCDAB</b></td> <td><b>swap4:</b> Swap bytes in a double word. <i>Example:</i> 32FCFF54 → 54FFFC32 (in hexadecimal format) 855441236 → 1426062386 (in decimal format)</td> </tr> <tr> <td><b>ABC...NOP -&gt; OPM...DAB</b></td> <td><b>swap8:</b> Swap bytes in a long word. <i>Example:</i> 142.366 → -893553517.588905 (in decimal format) 0 1000000110</td> </tr> </tbody> </table>	Value	Description	<b>Inv bits</b>	<b>inv:</b> Invert all the bits of the tag. <i>Example:</i> 1001 → 0110 (in binary format) 9 → 6 (in decimal format)	<b>Negate</b>	<b>neg:</b> Set the opposite of tag value. <i>Example:</i> 25.36 → -25.36	<b>AB -&gt; BA</b>	<b>swapnibbles:</b> Swap nibbles in a byte. <i>Example:</i> 15D4 → 514D (in hexadecimal format) 5588 → 20813 (in decimal format)	<b>ABCD -&gt; CDAB</b>	<b>swap2:</b> Swap bytes in a word. <i>Example:</i> 9ACC → CC9A (in hexadecimal format) 39628 → 52378 (in decimal format)	<b>ABCDEFGH -&gt; GHEFCDAB</b>	<b>swap4:</b> Swap bytes in a double word. <i>Example:</i> 32FCFF54 → 54FFFC32 (in hexadecimal format) 855441236 → 1426062386 (in decimal format)	<b>ABC...NOP -&gt; OPM...DAB</b>	<b>swap8:</b> Swap bytes in a long word. <i>Example:</i> 142.366 → -893553517.588905 (in decimal format) 0 1000000110
Value	Description														
<b>Inv bits</b>	<b>inv:</b> Invert all the bits of the tag. <i>Example:</i> 1001 → 0110 (in binary format) 9 → 6 (in decimal format)														
<b>Negate</b>	<b>neg:</b> Set the opposite of tag value. <i>Example:</i> 25.36 → -25.36														
<b>AB -&gt; BA</b>	<b>swapnibbles:</b> Swap nibbles in a byte. <i>Example:</i> 15D4 → 514D (in hexadecimal format) 5588 → 20813 (in decimal format)														
<b>ABCD -&gt; CDAB</b>	<b>swap2:</b> Swap bytes in a word. <i>Example:</i> 9ACC → CC9A (in hexadecimal format) 39628 → 52378 (in decimal format)														
<b>ABCDEFGH -&gt; GHEFCDAB</b>	<b>swap4:</b> Swap bytes in a double word. <i>Example:</i> 32FCFF54 → 54FFFC32 (in hexadecimal format) 855441236 → 1426062386 (in decimal format)														
<b>ABC...NOP -&gt; OPM...DAB</b>	<b>swap8:</b> Swap bytes in a long word. <i>Example:</i> 142.366 → -893553517.588905 (in decimal format) 0 1000000110														

Element	Description																																								
	<table border="1" style="width: 100%;"> <thead> <tr> <th style="background-color: #cccccc;">Value</th> <th style="background-color: #cccccc;">Description</th> </tr> </thead> <tbody> <tr> <td></td> <td>0001110010111011011001000101101000011100101011000001 → 1 10000011100 1010101000010100010110110110110010110110000100111101 (in binary format)</td> </tr> <tr> <td><b>BCD</b></td> <td> <b>bcd</b>: Separate byte in two nibbles, read them as decimal (from 0 to 9)   <i>Example:</i>            23 → 17 (in decimal format)            0001 0111 = 23            0001 = 1 (first nibble)            0111 = 7 (second nibble)         </td> </tr> </tbody> </table> <p>Select conversion and click +. The selected item will be added to list <b>Configured</b>.</p> <p>If more conversions are configured, they will be applied in order (from top to bottom of list <b>Configured</b>).</p> <p>Use the arrow buttons to order the configured conversions.</p>	Value	Description		0001110010111011011001000101101000011100101011000001 → 1 10000011100 1010101000010100010110110110110010110110000100111101 (in binary format)	<b>BCD</b>	<b>bcd</b> : Separate byte in two nibbles, read them as decimal (from 0 to 9)  <i>Example:</i> 23 → 17 (in decimal format) 0001 0111 = 23 0001 = 1 (first nibble) 0111 = 7 (second nibble)																																		
Value	Description																																								
	0001110010111011011001000101101000011100101011000001 → 1 10000011100 1010101000010100010110110110110010110110000100111101 (in binary format)																																								
<b>BCD</b>	<b>bcd</b> : Separate byte in two nibbles, read them as decimal (from 0 to 9)  <i>Example:</i> 23 → 17 (in decimal format) 0001 0111 = 23 0001 = 1 (first nibble) 0111 = 7 (second nibble)																																								
<b>Object Instance</b>	BACnet ID of the object to be referenced.																																								
<b>Object Property</b>	<p>Numeric value of the property to be referenced (example: the value 85 means <i>present-value</i> for most standard objects).</p> <p>The table below specifies all the BACnet Object Properties.</p> <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th style="background-color: #cccccc;">Property</th> <th style="background-color: #cccccc;">Value</th> <th style="background-color: #cccccc;">Property</th> <th style="background-color: #cccccc;">Value</th> <th style="background-color: #cccccc;">Property</th> <th style="background-color: #cccccc;">Value</th> <th style="background-color: #cccccc;">Property</th> <th style="background-color: #cccccc;">Value</th> </tr> </thead> <tbody> <tr> <td>accepted-modes</td> <td>175</td> <td>effective-period</td> <td>32</td> <td>max-info-frames</td> <td>63</td> <td>reason-for-halt</td> <td>100</td> </tr> <tr> <td>acked-transitions</td> <td>0</td> <td>elapsed-active-time</td> <td>33</td> <td>max-master</td> <td>64</td> <td>recipient-list</td> <td>102</td> </tr> <tr> <td>ack-required</td> <td>1</td> <td>error-limit</td> <td>34</td> <td>max-pres-value</td> <td>65</td> <td>records-since-notification</td> <td>140</td> </tr> <tr> <td>action</td> <td>2</td> <td>event-</td> <td>35</td> <td>max-</td> <td>167</td> <td>record-</td> <td>141</td> </tr> </tbody> </table>	Property	Value	Property	Value	Property	Value	Property	Value	accepted-modes	175	effective-period	32	max-info-frames	63	reason-for-halt	100	acked-transitions	0	elapsed-active-time	33	max-master	64	recipient-list	102	ack-required	1	error-limit	34	max-pres-value	65	records-since-notification	140	action	2	event-	35	max-	167	record-	141
Property	Value	Property	Value	Property	Value	Property	Value																																		
accepted-modes	175	effective-period	32	max-info-frames	63	reason-for-halt	100																																		
acked-transitions	0	elapsed-active-time	33	max-master	64	recipient-list	102																																		
ack-required	1	error-limit	34	max-pres-value	65	records-since-notification	140																																		
action	2	event-	35	max-	167	record-	141																																		



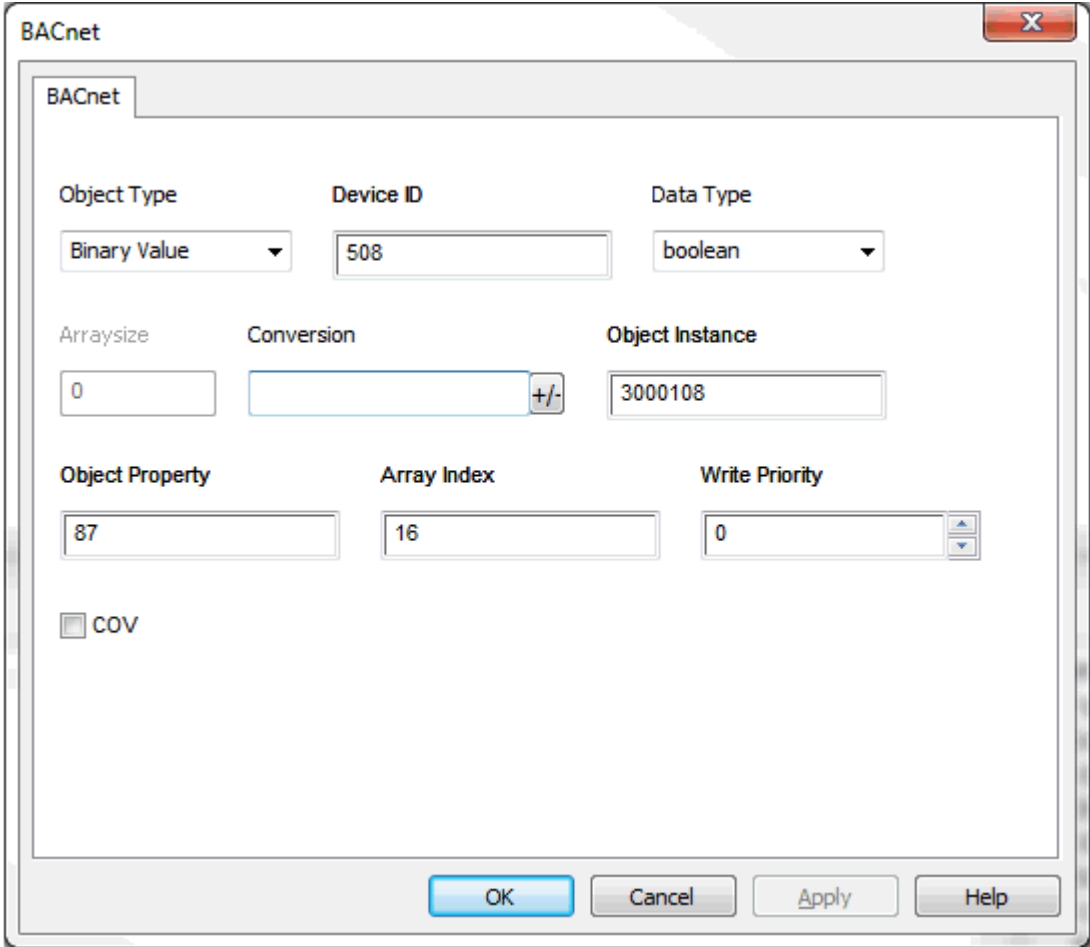
Element	Description															
	Property		Value		Property		Value		Property		Value		Property		Value	
				enable		segment		count								
	action-text	3		event-state	36	s-accepted		reliability	103							
	active-text	4		event-time-stamps	130	member-of	159	relinquish-default	104							
	active-vt-sessions	5		event-type	37	minimum-off-time	66	required	105							
	active-cov-subscriptions	152		event-parameters	83	minimum-on-time	67	resolution	106							
	adjust-value	176		exception-schedule	38	minimum-output	68	scale	187							
	alarm-value	6		fault-values	39	minimum-value	136	scale-factor	188							
	alarm-values	7		feedback-value	40	minimum-value-timestamp	150	schedule-default	174							
	all	8		file-access-method	41	min-pres-value	69	segmentation-supported	107							
	all-writes-successful	9		file-size	42	mode	160	setpoint	108							
	apdu-segment-timeout	10		file-type	43	model-name	70	setpoint-reference	109							
						modification-date	71									

Element	Description							
	Property	Value	Property	Value	Property	Value	Property	Value
	apdu-timeout	11	firmware-revision	44	notification-class	17	slave-address-binding	171
	application-software-version	12	high-limit	45	notification-threshold	137	setting	162
	archive	13	inactive-text	46	notify-type	72	silenced	163
	attempted-samples	124	in-process	47	number-of-APDU-retries	73	start-time	142
	auto-slave-discovery	169	input-reference	181	number-of-states	74	state-text	110
	average-value	125	instance-of	48	object-identifier	75	status-flags	111
	backup-failure-timeout	153	integral-constant	49	object-list	76	stop-time	143
	bias	14	integral-constant-units	50	object-name	77	stop-when-full	144
	buffer-size	126	last-notify-record	173	object-property-reference	78	system-status	112
	change-of-state-count	15	last-restore-time	157	object-type	79	time-delay	113
	change-of-state-time	16	life-safety-alarm-values	166	operation-expected	161	time-of-active-time-reset	114

Element	Description							
	Property	Value	Property	Value	Property	Value	Property	Value
	client-cov-increment	127	limit-enable	52	optional	80	time-of-state-count-reset	115
	configuration-files	154	limit-monitoring-interval	182	out-of-service	81	time-synchronization-recipients	116
	controlled-variable-reference	19	list-of-group-members	53	output-units	82	total-record-count	145
	controlled-variable-units	20	list-of-object-property-references	54	polarity	84	tracking-value	164
	controlled-variable-value	21	list-of-session-keys	55	prescale	185	units	117
	count	177	local-date	56	present-value	85	update-interval	118
	count-before-change	178	local-time	57	priority	86	update-time	189
	count-change-time	179	location	58	pulse-rate	186	utc-offset	119
	cov-increment	22	log-buffer	131	priority-array	87	valid-samples	146
	cov-period	180	log-device-object-property	132	priority-for-writing	88	value-before-change	190
	cov-resubscri	128	log-enable	133	process-identifier	89	value-set	191

Element	Description							
	Property	Value	Property	Value	Property	Value	Property	Value
	ption-interval							
	database-revision	155	log-interval	134	profile-name	168	value-change-time	192
	date-list	23	logging-object	183	program-change	90	variance-value	151
	daylight-savings-status	24	logging-record	184	program-location	91	vendor-identifier	120
	deadband	25	low-limit	59	program-state	92	vendor-name	121
	derivative-constant	26	maintenance-required	158	proportional-constant	93	vt-classes-supported	122
	derivative-constant-units	27	manipulated-variable-reference	60	proportional-constant-units	94	weekly-schedule	123
	description	28	manual-slave-address-binding	170	protocol-object-types-supported	96	window-interval	147
	description-of-halt	29	maximum-output	61	protocol-revision	139	window-samples	148
	device-address-binding	30	maximum-value	135	protocol-services-supported	97	zone-members	165

Element	Description							
	Property	Value	Property	Value	Property	Value	Property	Value
	device-type	31	maximum-value-timestamp	149	protocol-version	98		
	direct-reading	156	max-apdu-length-accepted	62	read-only	99		
<b>Array Index</b>	<p>Index for subscribing elements in BACnet arrays.</p> <ul style="list-style-type: none"> <li>-1 means read all elements</li> <li>0 to n means read the specified element</li> </ul> <p><b>Priority Array example</b>            To read a priority array object it is necessary to set <b>Object Property = 87</b> and <b>Array Index</b> has to refer to the priority item to be read.            The following figure shows how to read the 16th item of a priority array.</p>							

Element	Description
	
<b>Write Priority</b>	Write requests priority level. The value is in the range 1-16. 0 is interpreted as 16.
<b>COV</b>	Enable the Change Of Value notification.

## Clear/Set Priority

The system offers actions for a more flexible handling of Write Priority.

Action	Description
<b>BACnetClearPriority</b>	<p>Clears the priority array at the position associated to the BACnet tag passed as parameter.</p> <p>This action has immediate effect on the BACnet device.</p>
<b>BACnetClearAllPriorities</b>	<p>Clears all positions in the priority array.</p> <p>This action has immediate effect on the BACnet device.</p>
<b>BACnetSetPriority</b>	<p>Overrides the Write Priority value configured in the BACnet tag definition.</p> <p>This action has two parameters:</p> <ul style="list-style-type: none"> <li>• <b>TagName</b>: name of the BACnet tag.</li> <li>• <b>TagPriority</b>: new value of Write Priority for the BACnet tag passed as parameter.</li> </ul> <p>This action only overrides the value of Write Priority in the BACnet tag definition and does not perform any communication with the BACnet device. Any write command that will be performed to the Present Value property of the BACnet device identified by the tag, will be performed using the new Write Priority value.</p> <p>The priority value will be valid until:</p> <ul style="list-style-type: none"> <li>• A new call to the BACnetSetPriority action changes it.</li> <li>• The HMI device is restarted. The value of WritePriority defined in the project is valid in this case.</li> </ul>


## Tag Import

BACnet object information can be imported from BACnet EDE (Engineering Data Exchange) files. The EDE file must have the .csv extension.

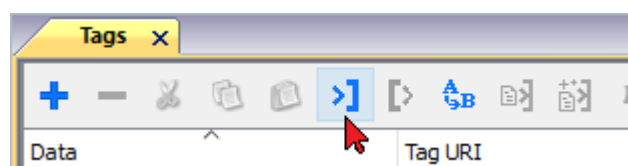
The importer uses the characters “,” and “;” as delimiters. They are considered as reserved characters and you cannot use them in file name.

Use the hierarchical importer to have a ordered list of BACnet objects and properties.

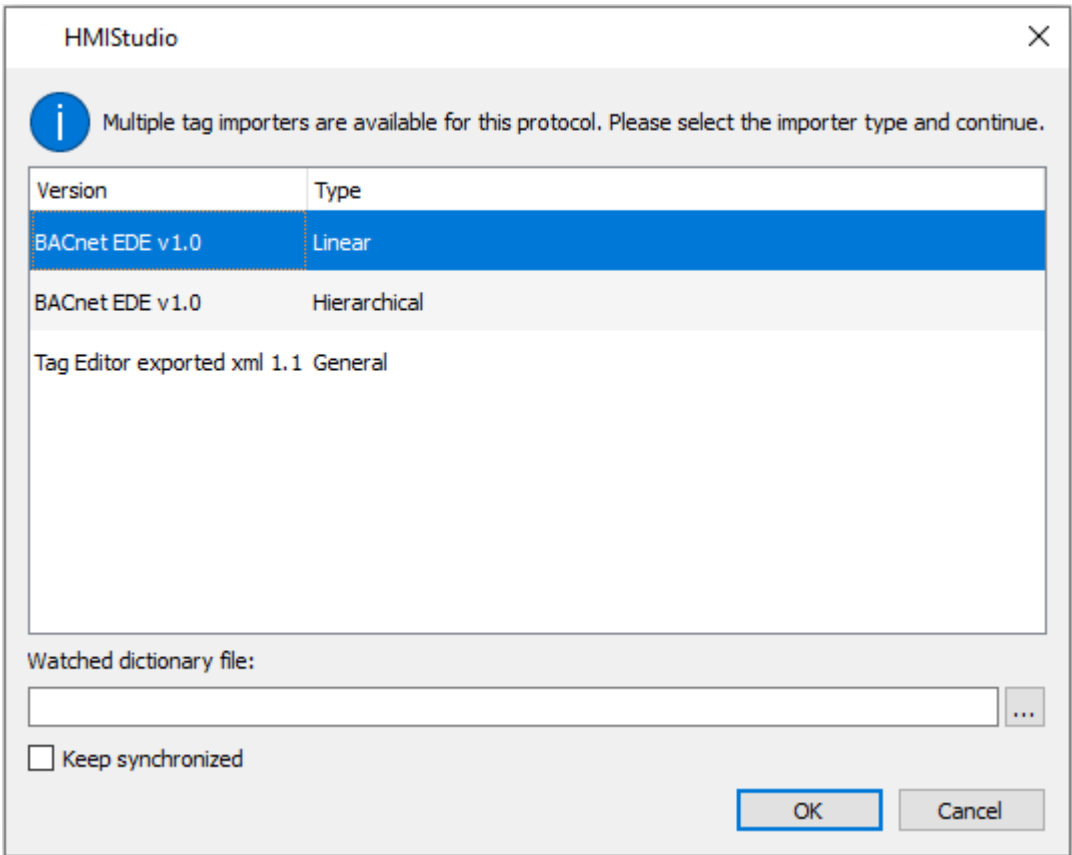
Tags will be created using the string specified in the column object-name of the EDE file. The importer will add the device ID as a prefix to avoid duplication of tag names.

 Note: The importer will ask to locate the State-Texts, Unit-Texts and Object-Types files. Click Cancel to ignore.

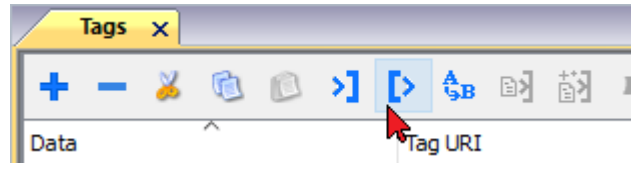
Select the driver in Tag Editor and click on the **Import Tags** button to start the importer.



The following dialog shows which importer type can be selected.



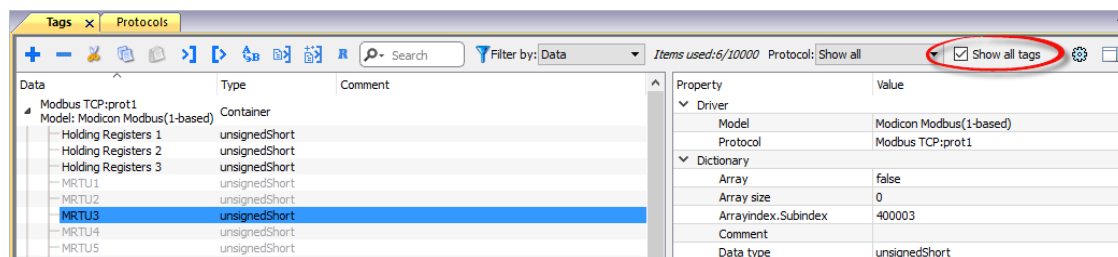
Importer	Description
<b>BACnet EDE v1.0 Linear</b>	Requires a <b>.csv</b> file. All variables will be displayed at the same level.
<b>BACnet EDE v1.0 Hierarchical</b>	Requires a <b>.csv</b> file. All variables will be displayed according to BACnet EDE Hierarchical view.
<b>Tag Editor exported xml</b>	Select this importer to read a generic XML file exported from Tag Editor by appropriate button.




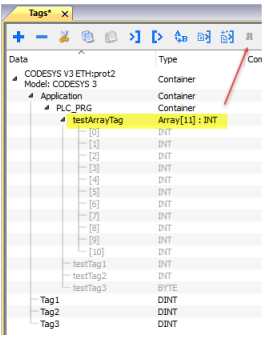
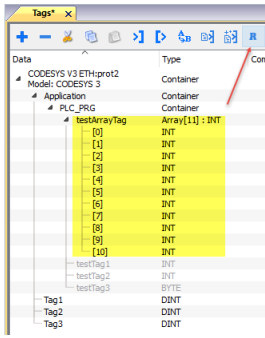



Once the importer has been selected, locate the symbol file and click **Open**.

The tags available within the Dictionary but not imported into the project are gray and are visible only when the "Show all tags" check box is selected.





Toolbar item	Description
	<p><b>Import Tag(s).</b></p> <p>Select tags to be imported and click on this icon to add tags from tag dictionary to the project</p>
	<p><b>Update Tag(s).</b></p> <p>Click on this icon to update the tags in the project, due a new dictionary import.</p>
	<p>Check this box to import all sub-elements of a tag.</p> <p>Example of both checked and unchecked result:</p> <div style="display: flex; justify-content: space-around;">   </div>
	<p>Searches tags in the dictionary basing on filter combobox item selected.</p>

For tags referring to BACnet objects of type Calendar or Schedule the tag refresh rate is set to “Manual”.

The following BACnet object properties are required for operation of the widgets.

Object	Tags to import
Calendar	Date_List
Schedule	Weekly_Schedule Exception_Schedule Default_Value Effective_Period

---

## DEVICE Object Properties

A BACnet network scanner can detect properties when exploring the network and obtaining data from HMI device.

These are the supported DEVICE object properties:

Property	Description
Object_Identifier	BACnetObjectIdentifier
Object_Name	CharacterString
Object_Type	BACnetObjectType
System_Status	BACnetDeviceStatus
Vendor_Name	CharacterString
Vendor_Identifier	Unsigned16
Model_Name	CharacterString
Firmware_Revision	CharacterString
Application_Software_Version	CharacterString
Protocol_Version	Unsigned
Protocol_Revision	Unsigned
Protocol_Services_Supported	BACnetServicesSupported
Protocol_Object_Types_Supported	BACnetObjectTypesSupported
Object_List	BACnetARRAY[N]of BACnetObjectIdentifier
Max_APDU_Length_Accepted	Unsigned
Segmentation_Supported	BACnetSegmentation
APDU_Timeout	Unsigned
Number_Of_APDU_Retries	Unsigned
Device_Address_Binding	List of BACnetAddressBinding
Database_Revision	Unsigned

## BACnet Alarm Events

The special “protAlarm:BACN” trigger mode, available from the Alarms Editor, gives the possibility to receive alarm events from the BACnet native alarms module.

Property	Description
<b>deviceID</b>	Identifies the BACnet device in the network.
<b>notificationClassID</b>	Notification Class ID to subscribe for the alarm events retrieving
<b>processID</b>	Not used
<b>activeMonday</b> <b>activeTuesday</b> <b>activeWednesday</b> <b>activeThursday</b> <b>activeFriday</b> <b>activeSaturday</b> <b>activeSunday</b>	Define in which days keep active the alarm events subscription <ul style="list-style-type: none"> <li>• False Subscription not active</li> <li>• True Subscription active</li> </ul>
<b>startHour</b> <b>startMinute</b> <b>startSecond</b> <b>endHour</b> <b>endMinute</b> <b>endSecond</b>	Define the time window where the alarm events subscription will be active

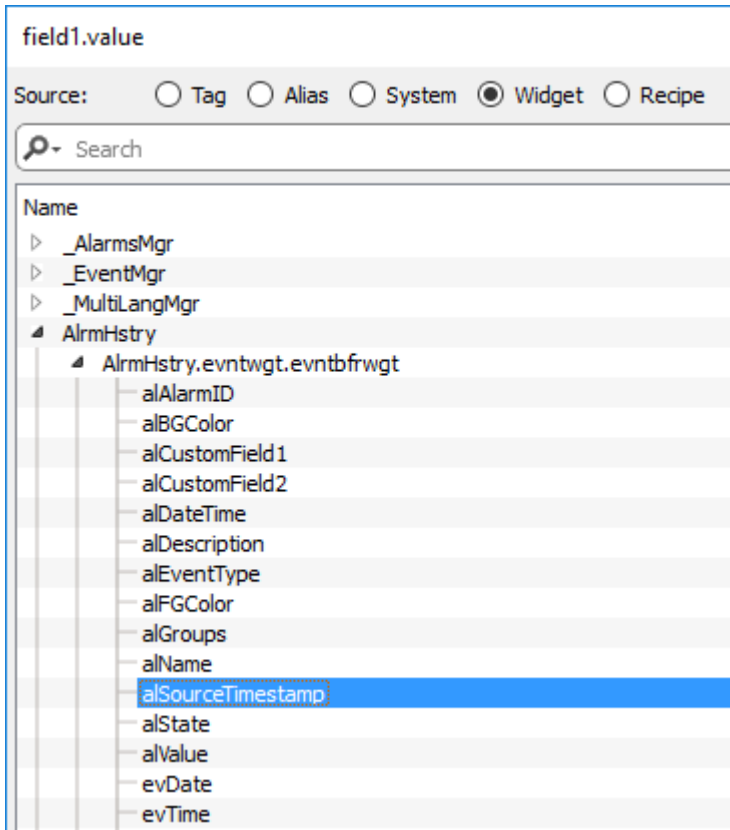
The alarm widgets will report the alarm information that are provided from the BACnet device.

Select	Name	State	Value	Time	Description
<input type="checkbox"/>	SISMI3NCE/Programming.4016.SUMMER-SP-SUPPLY:toOffNormal	Triggered Not Acked	90	13/02/2017 04:09:42	SUMMER ALARM
<input type="checkbox"/>	SISMI3NCE/Programming.4016.WINTER-SP-SUPPLY:toOffNormal	Triggered Not Acked	5	13/02/2017 04:10:06	WINTER ALARM

Filter :



When the special “protAlarm:BACN” trigger mode is used, the widget of the active alarms show the timestamp provided from the BACnet device while the widget of the historical alarms show the timestamp of when the alarm events are received from the HMI device. Generally, both timestamps are the same but if you need to show the timestamp from the BACnet device even inside the widget of the historical alarms you can add a new column configured to use the “allSourceTimestamp” value from the alarm history widget.

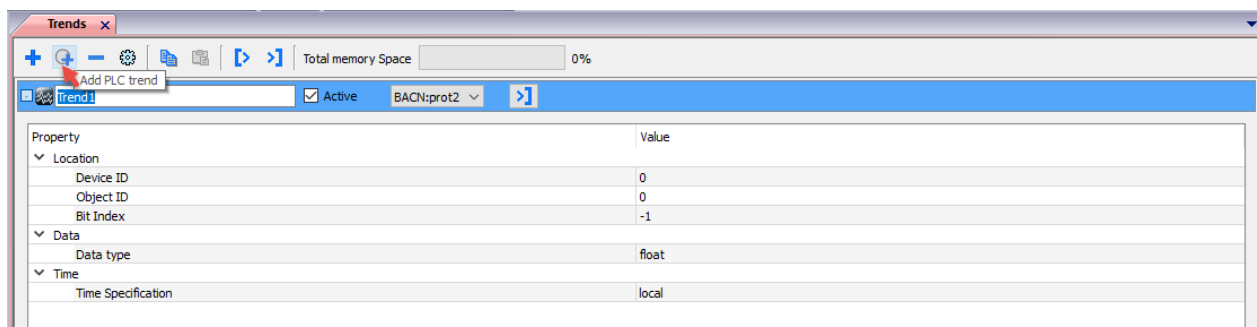


**BACnet alarm is a special alarm that require a double space to be stored inside the events buffer. This means, for example, if the events buffer is configured to contain 1.000 events only the last 500 BACnet events will be stored.**

## BACnet Trend Buffer

To use a BACnet trend object as a trend buffer:

1. Open the Trends Editor
2. Click the "Add PLC Trend" button (This button is enabled only when at least one BACnet protocol is configured)
3. Configure the below parameters to identify the BACnet trend object to use.




Property	Description
Device ID	Identifies the BACnet device in the network.
Object ID	BACnet ID of the trend object to be referenced.
Bit Index	When the data type is boolean, it is the index to select the bit to use inside the BACnet bit_string. It is not used with the other data types.
Data type	Specify the type of data of the BACnet trend object. The supported data types are: <ul style="list-style-type: none"> <li>boolean</li> <li>int</li> <li>unsignedInt</li> <li>float</li> </ul>
Time Specification	Time format used inside the selected BACnet trend object <ul style="list-style-type: none"> <li>local</li> <li>global (UTC)</li> </ul>

The trend buffer thus configured can then be used inside any trend widgets.

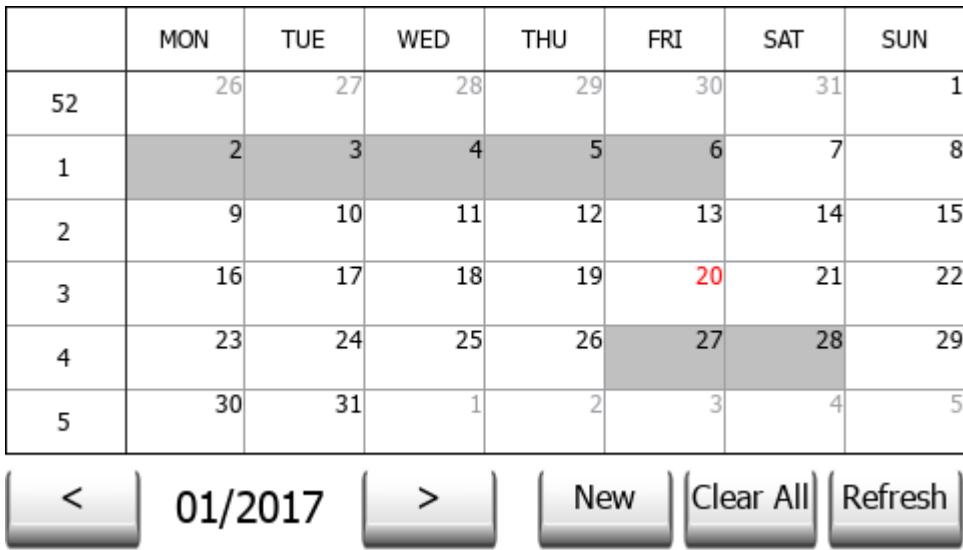
## BACnet Calendar Widget

Use Calendar widget to display content of a BACnet Calendar object.

Property	Description
Date_List	Connect to the "Date_List" tag of a BACnet calendar object in ReadOnly or Read/Write. <div data-bbox="427 1317 496 1384" style="display: inline-block; vertical-align: middle;">  </div> <p>Note: it can be connected to an alias which indexes a list of BACnet calendar Date_List(s), in order to use one calendar widget for more than one calendar object.</p>

### Operation of Calendar Widget

The widget shows data for one month.

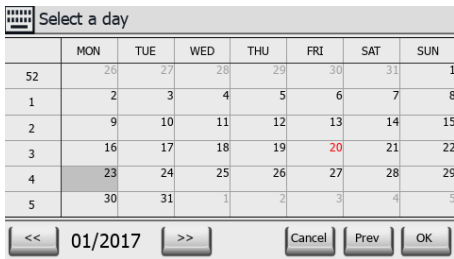
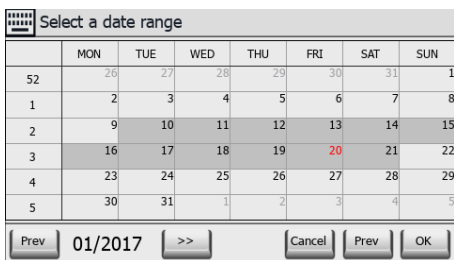


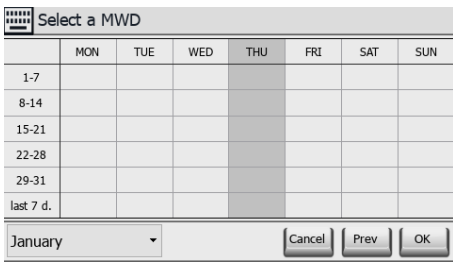
Use the < and > buttons to select the month to be displayed. The date of first day of the month is shown.

Swing gesture can be used on the widget to select the date.

### New

Press the button “New” to enter a new calendar item. The button is active only if the tag associated to the calendar has been configured as Read/Write.

Calendar item	Description
Single	<p>Click on a day to select a single day into the calendar</p> 
Range	<p>Click on the first day and on the last day to select a range of days into the calendar.</p> <ul style="list-style-type: none"> <li>• Single click on a day to change previous selected last day of the range.</li> <li>• Double click on a day to change previous selected first selected day of the range.</li> </ul> 

Calendar item	Description
<b>MWD</b>	<p>Select a Day or a Week for each year or each month.</p> 

### Clear All


Press the button “Clear All” to clear the content of the calendar object. The button is active only if the tag associated to the calendar has been configured as Read/Write. The button is configured to react to an onMouseHold event, to reduce risk of data loss.



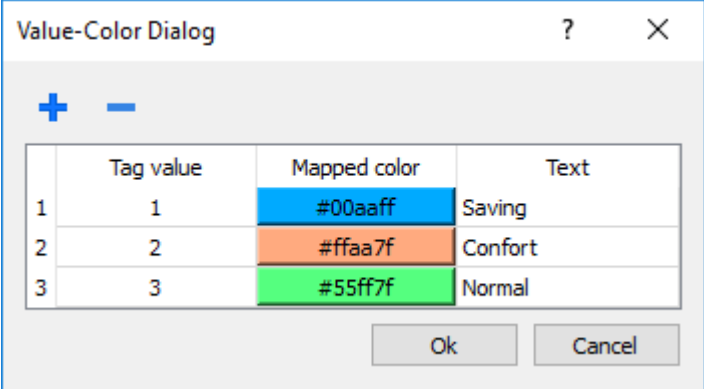
### Refresh

Press the “Refresh” button to start a manual refresh of the data of the widget. Always press the Refresh button after entering data in the calendar.

## BACnet Schedule Widget

Use Schedule widget to display content of BACnet Schedule object.

Property	Description
<b>Type</b>	<p>Select the type of BACnet object controlled by the schedule.</p> <p>Options are:</p> <ul style="list-style-type: none"> <li>• Binary</li> <li>• Real</li> <li>• Multistate</li> </ul>
<b>Weekly_Schedule</b>	<p>Attach to the Weekly_Schedule tag of the schedule object. The tag can be Read Only or Read/Write.</p>
<b>Exception_Schedule</b>	<p>Optionally attach to the Exception_Schedule tag of the schedule object. The tag can be Read Only or Read/Write. Only attach this property if exceptions are used.</p>
<b>Default_Value</b>	<p>Optionally attach to the Default_Value tag of the schedule object. The tag can be Read Only or Read/Write. Only attach this property if default values are used.</p>
<b>Cal. 0 (Date_List)</b>	<p>Optionally attach to the Date_List tag of the schedule widget in Read Only mode. Use this options to show the “calendar reference” exceptions.</p> <p> Note: An exception can be a single date, a date range, a mwd or a calendar reference. In this last case, exception_list does not contain</p>

Property	Description
	<p> the date information, but only time-value-priority and a reference to the calendar. The date_list needed to show the scheduling into the widget is stored into the relative BACNCalendar, and this is why we need this datalink. If there is no need to show calendar exceptions in the schedule, this property can be left void.</p> <p> Note: If it is not attached to a calendar, it is not possible to insert calendar exception. See BACNSchedKeypad for details.</p>
<b>Cal. 0 (Object_Name)</b>	Optionally attach to the property of the calendar. This name is used to identify the calendar in the BACNSchedKeypad used to insert calendar exceptions. If Object_Name is not attached, the calendar is identified with its instance number. This property is used only if a Cal. 0 (Date_List) is attached to a calendar.
<b>Cal. 1 (Date_List)</b>	Option for a second calendar.
<b>Cal. 1 (Object_Name)</b>	Option for a second calendar.
<b>Value-color-text Map</b>	<p>Defines the association value – Color/Text shown in the schedule. Use this option to define all possible values available in the BACNSched keypad.</p> 

### Operation of Schedule Widget

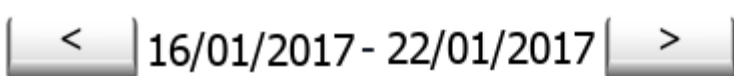
The widget shows data for one week.



Default Value: Normal



	MON	TUE	WED	THU	FRI	SAT	SUN
00:00							
04:00		E, 04:00 Normal					
08:00						E, 08:00 Confort	
12:00		E, 12:00 Confort					
16:00							
20:00		E, 20:00 Saving				E, 20:00 Saving	

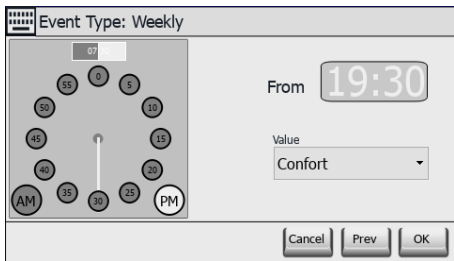
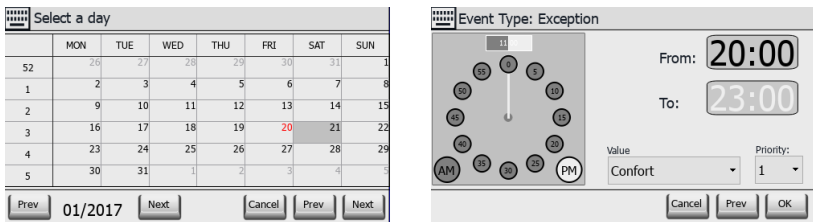


Use the < and > buttons to select the week to be displayed. The date of first day and last day of the week is shown.

Swing gesture can be used on the widget to select the date.

### New

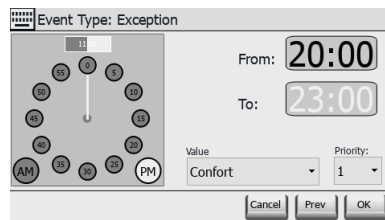
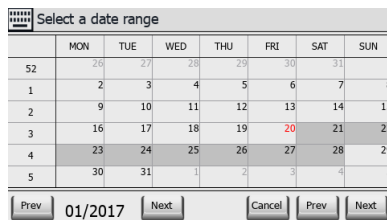
Press the button “New” to enter a new schedule item. The button is active only if the tag associated to Weekly Schedule or Exception Schedule has been configured as Read/Write.

Schedule item	Description
<b>Weekly</b>	<p>Select the day and click Weekly button, the following dialog box appears. Then select the desired value and the time when it should be set. Press OK to confirm the new item.</p> 
<b>Exception Single</b>	<p>Click on a day to select a single day into the calendar. On the next dialog select the time window, the desired value and its priority.</p> 
<b>Exception Range</b>	<p>Click on the first day and on the last day to select a range of days into the calendar.</p>

Schedule item	Description
---------------	-------------

- Single click on a day to change previous selected last day of the range.
- Double click on a day to change previous selected first selected day of the range.

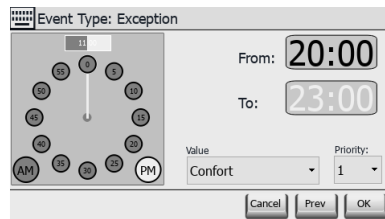
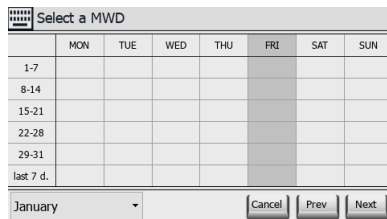
On the next dialog select the time window, the desired value and its priority.



**Exception MWD**

Select a Day or a Week for each year or each month.

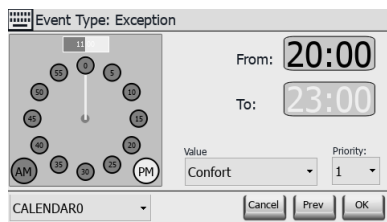
On the next dialog select the time window, the desired value and its priority.



**Exception Cal Ref**

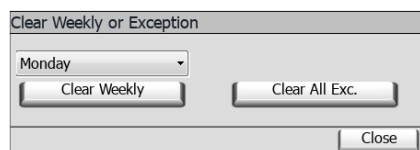
This option is available only if scheduler is linked to a calendar (configured as Read/Write)

Select the time window, the desired value and its priority. Value will set on all days defined from the calendar. If there are more calendars associated with Scheduler widget, select the calendar to use.



**Clear All**

Press the button “Clear All” to clear the content of the schedule object. The button is active only if the tag associated to the calendar has been configured as Read/Write. The button is configured to react to onMouseClick and onMouseHold events. The onMouseHold event will clear all data in the schedule. The onMouseClick event will recall a dialog box for selection of data to clear. It is needed to choice to clear weekly data or exception data.



**Refresh**

Press the “Refresh” button to start a manual refresh of the data of the widget. Always press the Refresh button after entering data in the schedule.

## BACnet Effective Period Widget

Use the Effective Period widget to feed information to the Effective\_Period tag of a Schedule object, if this is requested.

Property	Description
BACnet Effective_Period	Attach to the Effective_Period tag of the Schedule object

01/10/2017 - 01/13/2017

### Operation of Effective Period Widget

The widget shows starting date and end date for the period.

Click on the area showing the dates to activate the data entry procedure showing the keypad BACNDateRange.

⋮ **Select a date range**

Always
All month
All year

	MON	TUE	WED	THU	FRI	SAT	SUN
52	26	27	28	29	30	31	1
1	2	3	4	5	6	7	8
2	9	10	11	12	13	14	15
3	16	17	18	19	20	21	22
4	23	24	25	26	27	28	29
5	30	31	1	2	3	4	5

<
01/2017
>
Esc
Enter

The keypad shows data for one month.

Use the < and > buttons to select the month to be displayed. The date of first day of the month is shown.

You may use the swing gesture on the widget to select the date.

Select the period clicking of first day and last day of the period. The Effective\_Period is show with a different color.

The keypad offers three predefined options:

Option	Description
<b>Always</b>	The schedule will be always active.  <div style="text-align: right;"> <input type="text" value="**/**/*****"/> - <input type="text" value="**/**/*****"/> <input type="button" value="Refresh"/> </div>
<b>All Month</b>	The selected period will be extended to all months.  <div style="text-align: right;"> <input type="text" value="**/03/2017"/> - <input type="text" value="**/12/2017"/> <input type="button" value="Refresh"/> </div>
<b>All Year</b>	The selected period will be extended to all years.  <div style="text-align: right;"> <input type="text" value="01/03/*****"/> - <input type="text" value="01/12/*****"/> <input type="button" value="Refresh"/> </div>

### Refresh

Press the “Refresh” button to start a manual refresh of the data of the widget. Always press the Refresh button after entering data in the widget.

## BACnet Keypads

BACnet widgets require dedicated keypads for data entry.

Keypad	Description
<b>BACNCal</b>	Keypad for BACnet Calendar.
<b>BACNDateRange</b>	Keypad for BACnet Effective_Period.
<b>BACNDefVal</b>	Keypad for default value (embedded in the BACnet Schedule).
<b>BACNSched</b>	Keypad for BACnet Schedule.  This keypad is context sensitive. It will show different options depending on the type of schedule.

The system is configured to recall the appropriate keypad for each BACnet widget.

## Using BACnet Server

BACnet protocol is capable to act as BACnet Server, by exposing BACnet objects.

To properly setup BACnet Server, it is needed to execute the following steps:

1. Configure objects to expose from **Protocol Editor Settings**.

**BACnet**

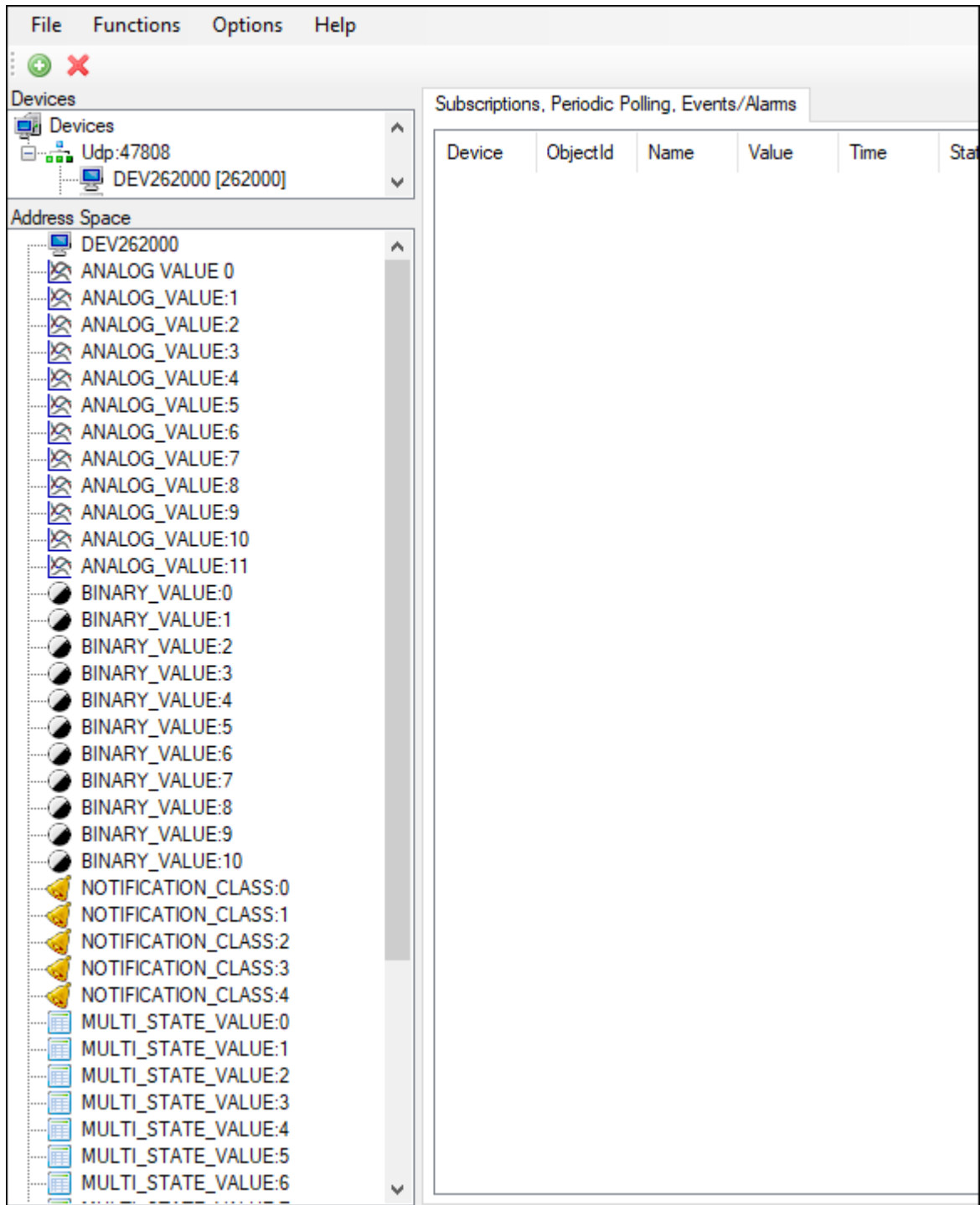
Comm...

Panel Device ID	262000	Analog Value Count	12
Object Name	DEV262000	Binary Value Count	11
Description	HMI	Multi State Value Count	18
Media	IP	Notification Class Count	5
Timeout (ms)	5000	IP UDP Port	47808
Panel Node	1	Local IP	
COV Lifetime (s)	60		
<input type="checkbox"/> COV Confirmed			
Max Master	127		
Max Info Frames	1		
max MS/TP APDU	480		
max IP APDU	1476		
Time Sync Interval (s)	0		
<input type="checkbox"/> Time Sync UTC			
PLC Models	default		

OK Cancel



Note: Objects configured in above image can be discovered by BACnet clients:



2. Create Tags that points to local BACnet objects, setting Device ID as the Device ID configured in Protocol Editor Settings:

The screenshot shows a 'BACnet' configuration window. The 'Device ID' field is highlighted with a red box and contains the value '262000'. Other fields include 'Object Type' (Analog Value), 'Data Type' (float), 'Arraysize' (0), 'Conversion' (+/-), 'Object Instance' (0), 'Object Property' (85), 'Array Index' (-1), and 'Write Priority' (0). A 'COV' checkbox is also present.

### Device objects description

Property Name	Code	Default value	Permanent	Note	Data Type
APDU timeout	11	Parameter	Yes		UnsignedInt
Application software version	12		Read-only		String
Database version	155		Read-only		UnsignedInt
Daylight saving status	24		Read-only		Boolean
Read-only	28	Parameter	Yes		String
Device address binding	30		Read-only		String
Firmware revision	44		Read-only		String
Local date	56		Read-only		UnsignedInt
Local time	57		Read-only		UnsignedInt

Property Name	Code	Default value	Permanent	Note	Data Type
Location	58	Parameter	Yes		String
Max APDU length accepted	62		Read-only		UnsignedInt
Max info frames	63	Parameter	Yes	Only if MSTP	String
Max master	64	Parameter	Yes	Only if MSTP	String
Model name	70		Read-only		String
Number of APDU retries	73	Parameter	Yes		UnsignedInt
Object identifier	75	Parameter	Yes		UnsignedInt + Conversion
Object list	76		Read-only		UnsignedInt + Conversion
Object name	77	Parameter	Yes		String
Object type	79		Read-only		UnsignedInt
Protocol object types supported	96		Read-only		Boolean(51)
Protocol revision	139		Read-only		UnsignedInt
Protocol services supported	97		Read-only		Boolean(40)
Protocol version	98		Read-only		UnsignedInt
Segmentation supported	107		Read-only		UnsignedInt
System status	112		Read-only		UnsignedInt
UTC offset	119		Read-only		Int
Vendor identifier	120		Read-only		UnsignedInt
Vendor name	121		Read-only		String

### Analog Value objects description

Property Name	Code	Default value	Permanent	Note	Data Type
Acked transitions	0		Read-only		Boolean(3)
COV increment	22	0	Yes		Float
Deadband	25	0	Yes		Float
Description	28	"ANALOG	Yes		String



Property Name	Code	Default value	Permanent	Note	Data Type
		VALUE n"			
Event enable	35	0	Yes		Boolean(3)
Event state	36	0	Read-only		UnsignedInt
Event time stamps	130		Yes		UnsignedInt(3)
High limit	45	0	Yes		Float
Limit enable	52	0	Yes		Boolean(2)
Low limit	59	0	Yes		Float
Notification class	17	4194303	Yes		UnsignedInt
Notify type	72	0	Yes		UnsignedInt
Object identifier	75	2:n	Read-only		UnsignedInt + Conversion
Object name	77	"ANALOG VALUE n"	Yes		String
Object type	79	2	Read-only		UnsignedInt
Out of service	81	0	Yes		Boolean
Present value	85	0			Float
Priority array	87		Read-only		16 Single tag String
Reliability	103	0	Yes		UnsignedInt
Relinquish default	104	0	Yes		Float
Status flags	111		Read-only		Boolean(4)
Time delay	113	0	Yes		UnsignedInt
Units	117	98	Yes		Units

### Binary Value objects description

Property Name	Code	Default value	Permanent	Note	Data Type
Acked transitions	0		Read-only		Boolean(3)
Active text	4		Yes		String
Alarm value	6	0	Yes		Boolean
Description	28	"BINARY VALUE n"	Yes		String

Property Name	Code	Default value	Permanent	Note	Data Type
Event enable	35	0	Yes		Boolean(3)
Event state	36	0	Read-only		UnsignedInt
Event time stamps	130		Yes		UnsignedInt(3)
Inactive text	46		Yes		String
Notification class	17	4194303	Yes		UnsignedInt
Notify type	72	0	Yes		UnsignedInt
Object identifier	75	5:n	Read-only		UnsignedInt + Conversion
Object name	77	"BINARY VALUE n"	Yes		String
Object type	79	5	Read-only		UnsignedInt
Out of service	81	0	Yes		Boolean
Polarity	84	0	Yes		UnsignedInt
Present value	85	0			Boolean
Priority array	87		Read-only		16 Single tag String
Reliability	103	0	Yes		UnsignedInt
Relinquish default	104	0	Yes		Boolean
Status flags	111		Read-only		Boolean(4)
Time delay	113	0	Yes		UnsignedInt

### Multi State Value objects description

Property Name	Code	Default value	Permanent	Note	Data Type
Acked transitions	0		Read-only		Boolean(3)
Alarm values	7		Yes	Defines number of array elements	UnsignedInt
				Array of alarm values (0:n)	UnsignedInt(n)
Description	28	"MULTI STATE VALUE n"	Yes		String
Event enable	35	0	Yes		Boolean(3)

Property Name	Code	Default value	Permanent	Note	Data Type
Event state	36	0	Read-only		UnsignedInt
Event time stamps	130		Yes		UnsignedInt(3)
Fault values	39		Yes	Defines number of array elements	UnsignedInt
				Array of fault values (0:n)	UnsignedInt(n)
Number of states	74	1	Yes		UnsignedInt
Notification class	17	4194303	Yes		UnsignedInt
Notify type	72	0	Yes		UnsignedInt
Object identifier	75	19:n	Read-only		UnsignedInt + Conversion
Object name	77	"MULTI STATE VALUE n"	Yes		String
Object type	79	19	Read-only		UnsignedInt
Out of service	81	0	Yes		Boolean
Present value	85	0			UnsignedInt
Priority array	87		Read-only		16 Single tag String
Reliability	103	0	Yes		UnsignedInt
Relinquish default	104	0	Yes		UnsignedInt
State text	110		Yes		UnsignedInt
Status flags	111		Read-only		Boolean(4)
Time delay	113	0	Yes		UnsignedInt

### Notification Class objects description

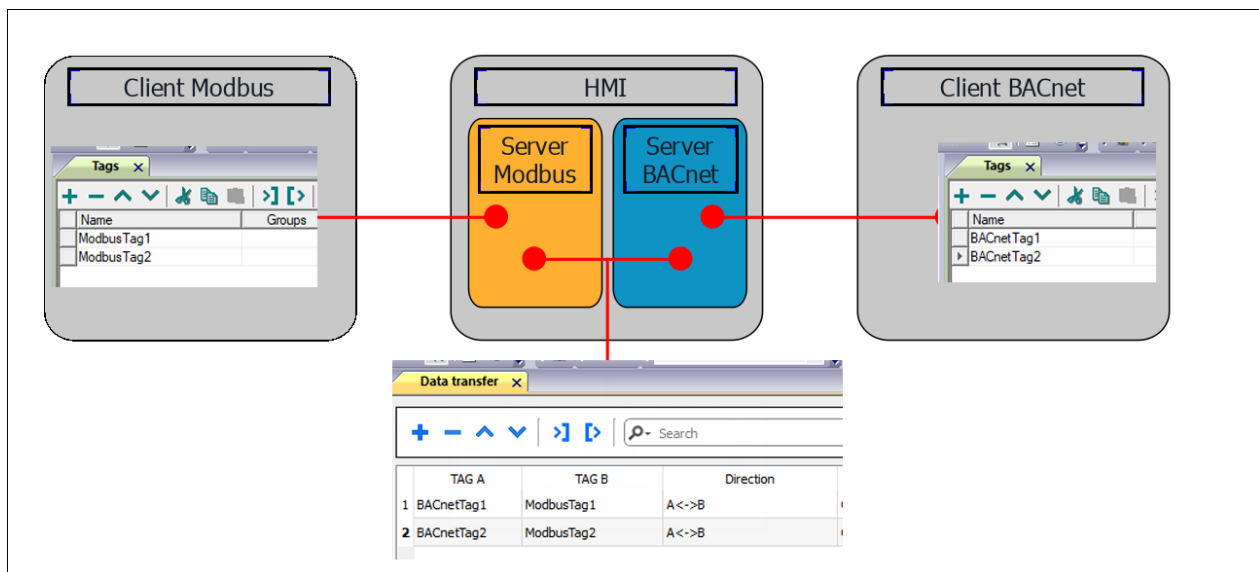
Property Name	Code	Default value	Permanent	Note	Data Type
Ack required	1	0	Yes		Boolean(3)
Description	38	"NOTIFICATION CLASS n"	Yes		String
Notification class	17	4194303	Yes		UnsignedInt
Object identifier	75	15:n	Read-only		UnsignedInt +

Property Name	Code	Default value	Permanent	Note	Data Type
					Conversion
<b>Object name</b>	77	"NOTIFICATION CLASS n"	Yes		String
<b>Object type</b>	79	15	Read-only		UnsignedInt
<b>Priority</b>	86	255,255,255	Yes		UnsignedInt(3)
<b>Recipient list</b>	102		Yes		UnsignedInt(n)

### Example of usage

Once BACnet Server Tags are configured, they can be used in combination with Data Transfer feature.

Example: Modbus TCP/RTU Tags can be transferred to BACnet Tags (with same data type). In this way, all BACnet clients can reach BACnet Server and see actual value of Modbus Tags, using BACnet Tags as interface.



### JavaScript Interface

Beside Tag interface the user can access the protocol via JavaScript.

Although defined Tags can be accessed by JavaScript too, JavaScript can access directly to a Command interface implemented in protocol. This interface does not require the definition of Tags and is direct to protocol resulting in more efficiency.

The following commands are supported:

Command	Description
<b>scan (minID, maxID, &lt;timeout&gt;)</b>	Executes a scan for devices in the given range.
<b>scan_status</b>	Get the scanning result.
<b>devices</b>	Get the list of devices.
<b>objectCount (deviceID, objectType)</b>	Get the object count of given object types in given device.
<b>objectNames (start, count)</b>	Get the part of object names asked by previous objectCount.
<b>properties (deviceID, objectType, objectInstance)</b>	Get the properties of given device/object.

- **scan**

Scan the bus to find all present devices having ID in the range minID – maxID.

To scan the whole network use 0 and 999999 as minID and maxID.

The optional timeout can be indicated in milliseconds. Default value is 2000 ms.

The function starts the scan operation; the function scan\_status can be used to know the status of the operation.

The result of the operation is “**scanning**”.

- **scan\_status**

Get the status of last started scan operation. It returns “**scanning**” or “**finished**”.

Scan operation finishes when the timeout time is expired

- **devices**

Get the list of devices found by latest scan operation. The result is a JSON string containing of each device:

- device name
- model name
- vendor name
- vendor ID

Example:

```
{ "minID":0, "maxID":999999, "devices": [262000, 1101], "deviceNames":
["DEV262000", "S01101"], "modelName": ["HMI model", "EY-AS525F001"], "vendorNames":
["Company Name", "SAUTER"], "vendorIDs": [262, 80] }
```

- **objects**

Get the list of all objects from the devices having the given ID.

The list is returned as a JSON string containing for each object

- type
- instance number

type can be:

OBJECT\_ANALOG\_INPUT = 0,

OBJECT\_ANALOG\_OUTPUT = 1,

OBJECT\_ANALOG\_VALUE = 2,

---

OBJECT\_BINARY\_INPUT = 3,  
OBJECT\_BINARY\_OUTPUT = 4,  
OBJECT\_BINARY\_VALUE = 5,  
OBJECT\_CALENDAR = 6,  
OBJECT\_COMMAND = 7,  
OBJECT\_DEVICE = 8,  
OBJECT\_EVENT\_ENROLLMENT = 9,  
OBJECT\_FILE = 10,  
OBJECT\_GROUP = 11,  
OBJECT\_LOOP = 12,  
OBJECT\_MULTI\_STATE\_INPUT = 13,  
OBJECT\_MULTI\_STATE\_OUTPUT = 14,  
OBJECT\_NOTIFICATION\_CLASS = 15,  
OBJECT\_PROGRAM = 16,  
OBJECT\_SCHEDULE = 17,  
OBJECT\_AVERAGING = 18,  
OBJECT\_MULTI\_STATE\_VALUE = 19,  
OBJECT\_TRENDLOG = 20,  
OBJECT\_LIFE\_SAFETY\_POINT = 21,  
OBJECT\_LIFE\_SAFETY\_ZONE = 22,  
OBJECT\_ACCUMULATOR = 23,  
OBJECT\_PULSE\_CONVERTER = 24,  
OBJECT\_EVENT\_LOG = 25,  
OBJECT\_GLOBAL\_GROUP = 26,  
OBJECT\_TREND\_LOG\_MULTIPLE = 27,  
OBJECT\_LOAD\_CONTROL = 28,  
OBJECT\_STRUCTURED\_VIEW = 29,  
OBJECT\_ACCESS\_DOOR = 30,  
OBJECT\_TIMER = 31,  
OBJECT\_ACCESS\_CREDENTIAL = 32,  
OBJECT\_ACCESS\_POINT = 33,  
OBJECT\_ACCESS\_RIGHTS = 34,

OBJECT\_ACCESS\_USER = 35,  
OBJECT\_ACCESS\_ZONE = 36,  
OBJECT\_CREDENTIAL\_DATA\_INPUT = 37,  
OBJECT\_NETWORK\_SECURITY = 38,  
OBJECT\_BITSTRING\_VALUE = 39,  
OBJECT\_CHARACTERSTRING\_VALUE = 40,  
OBJECT\_DATE\_PATTERN\_VALUE = 41,  
OBJECT\_DATE\_VALUE = 42,  
OBJECT\_DATETIME\_PATTERN\_VALUE = 43,  
OBJECT\_DATETIME\_VALUE = 44,  
OBJECT\_INTEGER\_VALUE = 45,  
OBJECT\_LARGE\_ANALOG\_VALUE = 46,  
OBJECT\_OCTETSTRING\_VALUE = 47,  
OBJECT\_POSITIVE\_INTEGER\_VALUE = 48,  
OBJECT\_TIME\_PATTERN\_VALUE = 49,  
OBJECT\_TIME\_VALUE = 50,  
OBJECT\_NOTIFICATION\_FORWARDER = 51,  
OBJECT\_ALERT\_ENROLLMENT = 52,  
OBJECT\_CHANNEL = 53,  
OBJECT\_LIGHTING\_OUTPUT = 54,  
OBJECT\_BINARY\_LIGHTING\_OUTPUT = 55,  
OBJECT\_NETWORK\_PORT = 56,

Other types are manufacturer specific.

- **objectCount**

Returns the number of objects of a defined type in the device having the indicated ID.  
If specified type is -1 the command will return the number of all objects.

Example:

```
objectCount 1101 -1  
77
```

```
objectCount 1101 0  
1
```

```
objectCount 1101 1  
1
```

```
objectCount 1101 3  
2
```

---

objectCount 1101 29  
16

- **objectNames**

Returns a part of the objects listed by a previous **objectCount** command, from start index. The list contains only counted objects according to filter previously used

The list is returned as a JSON string containing for each object

- type
- instance number
- name

Example:

```
{"deviceID":1101,"objects":[{"type":29,"instance":0,"name":"0x7400000"},  
{"type":29,"instance":16,"name":"0x7400010"},  
{"type":29,"instance":18,"name":"0x7400012"},  
{"type":29,"instance":19,"name":"0x7400013"},  
{"type":29,"instance":20,"name":"0x7400014"},  
{"type":29,"instance":21,"name":"0x7400015"},  
{"type":29,"instance":22,"name":"0x7400016"},  
{"type":29,"instance":23,"name":"0x7400017"},  
{"type":29,"instance":24,"name":"0x7400018"},  
{"type":29,"instance":25,"name":"0x7400019"},  
{"type":29,"instance":26,"name":"0x740001a"},  
{"type":29,"instance":27,"name":"0x740001b"},  
{"type":29,"instance":28,"name":"0x740001c"},  
{"type":29,"instance":29,"name":"0x740001d"},  
{"type":29,"instance":30,"name":"0x740001e"},  
{"type":29,"instance":31,"name":"0x740001f"}]}
```

- **properties**

Returns the list of properties available for object with given type and instance number in device having the given ID.

The list is returned as a JSON string containing for each object

- deviceID
- object type
- object instance
- list of available properties

Example:

```
{"deviceID":1101,"objectType":2,"objectInstance":1,  
"properties":  
[22,28,36,65,69,75,77,79,81,85,87,103,104,111,117,168,8309,8314,8332,8333]}
```

Example of usage:



```

var tagMgr = project.getWidget("_TagMgr");
var protID = "prot2"; // to be set according to protocol numbering
var params = String(fromId) + " " + String(toId) + " " + String
(timeout); // fromID and toID are min and max IDs

var json_str = tagMgr.invokeProtocolCommand(protID , "scan", params, state); //json_
str contains JSON string with scanned devices.

```

## Communication status

Current communication status can be displayed using system variables. See "System Variables" section in the main manual.

Codes supported by this communication driver:

Error	Cause
<b>Cannot bind to the device_id</b>	Cannot establish communication with the Device ID provided for this tag.
<b>Cannot read the property data type</b>	The type of the property to write cannot be determined.
<b>write conversion error</b>	A conversion associated to this tag has failed.
<b>Cannot write ICOM type .... BACnet type ....</b>	A datatype selected for this tag is not compatible with the BACnet property to set.
<b>Timeout on COV subscription</b>	A request for COV subscription for this tag has timed out.
<b>Timeout on waiting COV update</b>	A COV notification has not been received for this tag within timeout.
<b>Can't get COV for this property</b>	The selected property for COV notification is unsupported.
<b>datagramItem conversion error</b>	A conversion associated to a tag that is part of a datagram has failed.
<b>Timeout waiting on response</b>	No response for a request of read or write property within timeout.
<b>datagram element ....., no data available</b>	No data available for a tag that is part of datagram.
<b>datagram element ....., Unsupported BACnet data type</b>	Read datagram element is of unsupported BACnet type.
<b>datagram element ....., can't convert BACnet type to ....</b>	A Data Type selected for a tag which is part of a datagram is not compatible with the BACnet property to read.

<b>Error</b>	<b>Cause</b>
<b>No data in response</b>	No data available for a tag.
<b>Datagram element 'element_URI' error: 'error_class': error_code</b>	The reading of indicated datagram element 'element_URI' was reported as error. The error descriptions <b>error_class</b> and <b>error_code</b> are included in the message.
<b>datagram object does not match</b>	The object of the received datagram item does not match the asked object.
<b>datagram property does not match</b>	The property of the received datagram item does not match the asked property.
<b>BACnet abort: reason_of_abort</b>	BACnet abort message was received. The reason of abort is given.
<b>BACnet reject: reason_of_rejection</b>	BACnet reject message was received. The reason of rejection is given.
<b>BACnet error: error_class: error_code</b>	BACnet error message was received. The error description is given as combination of <b>error_class</b> and <b>error_code</b> .
<b>parameter 'parameter_name' out of range</b>	The protocol parameter <b>parameter_name</b> value is out of range.

# Client System Variables

Client System Variables communication driver allows to create Tags that point to system information.

Refer to [Client System Variables > Protocol](#) chapter of User's Manual.

---

## Protocol Editor Settings

Client System Variables communication driver allows to create Tags that point to system information.

Refer to [Client System Variables > Protocol](#) chapter of User's Manual.

# CODESYS V2 ETH

CODESYS V2 ETH communication driver for supports communication through Ethernet connection with controllers based on the CODESYS V2.3 version.

## Protocol Editor settings

### Adding a protocol

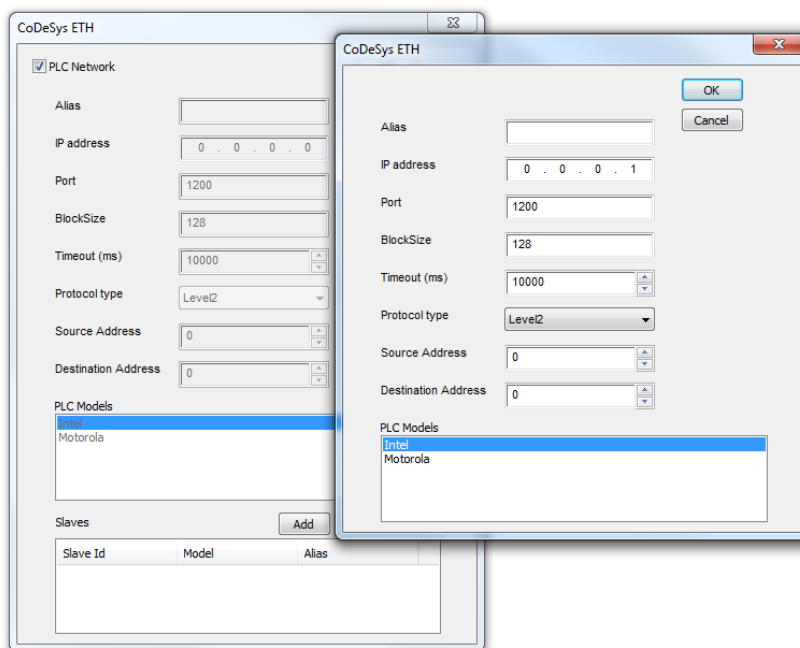
To configure the protocol:

1. In **Config** node double-click **Protocols**.
2. To add a driver, click **+**: a new line is added.
3. Select the protocol from the **PLC** list.

The protocol configuration dialog is displayed.

Element	Description
<b>Alias</b>	Name identifying nodes in network configurations. The name will be added as a prefix to each tag name imported for each network node.
<b>IP address</b>	Ethernet IP address of the controller.
<b>Port</b>	Port number used by the CODESYS V2 Ethernet driver. The default value is set to <b>1200</b> , which is also the default setting of CODESYS-based controllers.
<b>Block Size</b>	Maximum block size supported by your controller (limit is 1024 KB ).
<b>Timeout (ms)</b>	Time delay in milliseconds between two retries of the same message when communication fails.

Element	Description
<b>Protocol type</b>	Protocol variant to be used. Please make sure you check which protocol variant is supported by the CODESYS runtime you want to connect.
<b>Source Address, Destination Address</b>	Available only when <b>TCP/IP Level 2 Route</b> is selected in <b>Protocol Type</b> . The Destination is the node of the PLC and allows the protocol to read variables in a sub-network. The address is used to read variables when multiple PLCs are connected in a sub-network (serial network) but only one have the Ethernet interface.
<b>PLC Models</b>	Two PLC models are available. <ul style="list-style-type: none"> <li>• Intel</li> <li>• Motorola</li> </ul>
<b>PLC Network</b>	IP address for all controllers in multiple connections. <b>PLC network</b> check box must be selected to enable multiple connections.



*CODESYS V2 Ethernet driver supports connection to multiple controllers starting from version V1.60.*



Note: CODESYS V2 Ethernet driver is recommended when creating projects for the internal controller iPLC CODESYS. To use the CODESYS V2 Ethernet driver with iPLC, configure the IP address of the PLC as localhost (127.0.0.1).

*iPLC CODESYS supports communication with CODESYS V2 Ethernet driver with symbol based support starting from V1.55 and above.*

## Adding an alias name to a protocol

Tag names must be unique at project level, however, the same tag names might need to be used for different controller nodes (for example when the HMI device is connected to two devices running the same application).

When creating a protocol you can add an alias name that will be added to tag names imported for this protocol.

In the example, the connection to a certain controller is assigned the name **Node1**. When tags are imported for this node, all tag names will have the prefix **Node1** making each of them unique at the network/project level.

The screenshot shows the 'Tags' editor for a 'Modbus TCP:prot1' connection. The main table lists tags with columns: Name, Group, Driver, Address, and Comment. The 'Node1/Water\_Level' tag is highlighted. A 'Network' dialog box is open, showing a table with columns: Slave Id, Model, and Alias. The 'Alias' column contains 'Node1' and 'Node2'. A red box highlights the 'Node1' alias, and a red arrow points from it to the 'Node1/Water\_Level' tag in the main list.

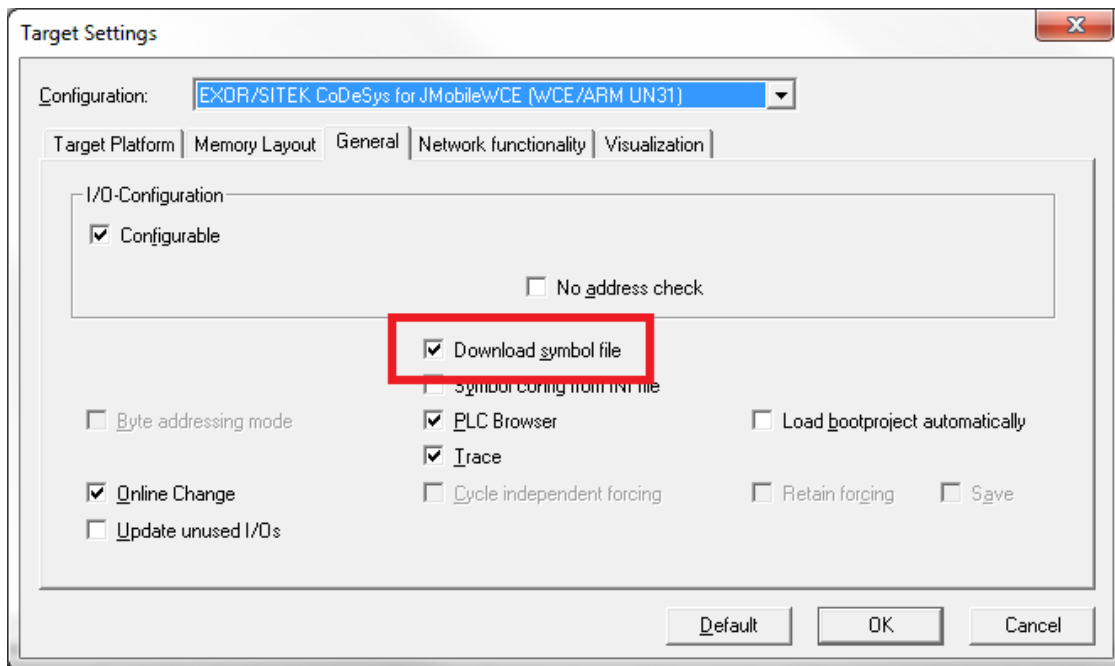
Name	Group	Driver	Address	Comment
Node1/Bat_Level		ABB Modbus TCP prot1	1 11 0 unsignedShort	
Node1/Data_Level		ABB Modbus TCP prot1	1 12 0 unsignedShort	
Node1/IN_WATER_Level		ABB Modbus TCP prot1	1 1 0 unsignedShort	
Node1/OBIOGRBCHN		ABB Modbus TCP prot1	1 245 0 unsignedShort	
Node1/OUT_BAT_Level		ABB Modbus TCP prot1	1 1 0 unsignedShort	
Node1/R_DATA_Level		ABB Modbus TCP prot1	1 2 0 unsignedShort	
Node1/WATER		ABB Modbus TCP prot1	1 3 0 unsignedShort	
Node1/Water_Level		ABB Modbus TCP prot1	1 10 0 unsignedShort	

Slave Id	Model	Alias
1-1-1	Modbus modbus	Node1
1-1-2	Modbus modbus	Node2

**i** Note: Aliasing tag names is only available for imported tags. Tags added manually in the Tag Editor cannot have the Alias prefix in the tag name. The Alias string is attached at the time of tag import. If you modify the Alias string after the tag import has been completed, there will be no effect on names already present in the dictionary. When the Alias string is changed and tags are re-imported, all tags will be re-imported with the new prefix string.

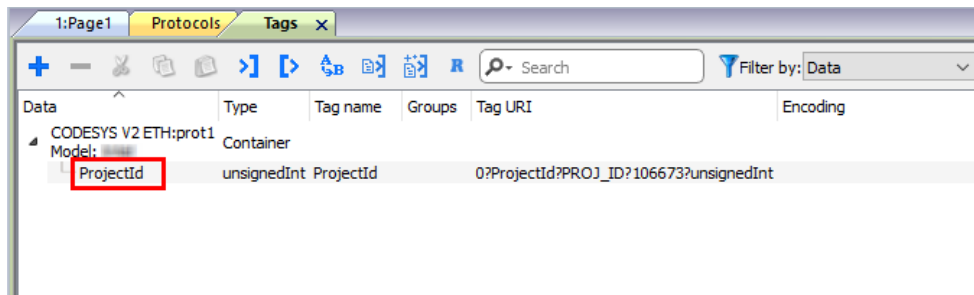
## CODESYS software settings

When creating the project in CODESYS, select **Download symbol file**.



Note: CODESYS V2 Ethernet communication driver supports the automatic symbol file (SDB) upload from the PLC; any change in the tag offset due to new compilation of the PLC program does not require a symbol file re-import. Tag file has to be re-imported only in case of tag rename or definition of new tags.

When the option **Download symbol file** is not available or cleared, the protocol can work only if the **ProjectId** tag is imported. If the tag offset changes because of a new compilation of the PLC program, the symbol file must be re-imported.



## Data types

The import module supports variables of standard data types and user defined data types.



**Supported data types**

- BOOL
- WORD
- DWORD
- INT
- UINT
- UDINT
- DINT
- STRING \*
- REAL
- TIME
- DATE & TIME

and 1-dimensional ARRAY of the types above. See "Programming concepts" section in the main manual.



Note \*: String length for a STRING variable in PLC should be max 80 characters. Declare a STRING variable either with a specified size (str: STRING(35) or default size (str: STRING) which is 80 characters.

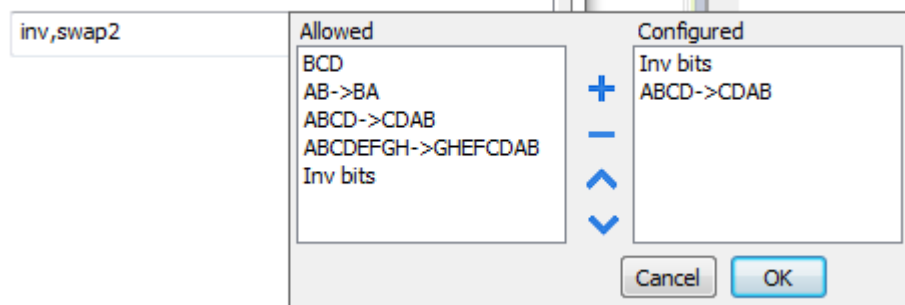
**Unsupported data types**

- LWORD
- LINT
- LREAL

**Tag conversion**

Conversion to be applied to the tag.

Conversion



Depending on data type selected, the list **Allowed** shows one or more conversion types.

Value	Description
<b>Inv bits</b>	<p><b>inv:</b> Invert all the bits of the tag.</p> <p><i>Example:</i>            1001 → 0110 (in binary format)            9 → 6 (in decimal format)</p>
<b>Negate</b>	<p><b>neg:</b> Set the opposite of tag value.</p> <p><i>Example:</i>            25.36 → -25.36</p>
<b>AB -&gt; BA</b>	<p><b>swapnibbles:</b> Swap nibbles in a byte.</p> <p><i>Example:</i>            15D4 → 514D (in hexadecimal format)            5588 → 20813 (in decimal format)</p>
<b>ABCD -&gt; CDAB</b>	<p><b>swap2:</b> Swap bytes in a word.</p> <p><i>Example:</i>            9ACC → CC9A (in hexadecimal format)            39628 → 52378 (in decimal format)</p>
<b>ABCDEFGH -&gt; GHEFCDAB</b>	<p><b>swap4:</b> Swap bytes in a double word.</p> <p><i>Example:</i>            32FCFF54 → 54FFFC32 (in hexadecimal format)            855441236 → 1426062386 (in decimal format)</p>
<b>ABC...NOP -&gt; OPM...DAB</b>	<p><b>swap8:</b> Swap bytes in a long word.</p> <p><i>Example:</i>            142.366 → -893553517.588905 (in decimal format)            0 10000000110            0001110010111011011001000101101000011100101011000001            →            1 10000011100            1010101000010100010110110110110010110110000100111101            (in binary format)</p>
<b>BCD</b>	<p><b>bcd:</b> Separate byte in two nibbles, read them as decimal (from 0 to 9)</p> <p><i>Example:</i>            23 → 17 (in decimal format)            0001 0111 = 23            0001 = 1 (first nibble)            0111 = 7 (second nibble)</p>

Select conversion and click +. The selected item will be added to list **Configured**.

If more conversions are configured, they will be applied in order (from top to bottom of list **Configured**).

Use the arrow buttons to order the configured conversions.

## Node Override IP

The protocol provides the special data type Node Override IP which allows you to change the IP address of the target controller at runtime.

This memory type is an array of 4 unsigned bytes, one per each byte of the IP address.

The Node Override IP is initialized with the value of the controller IP specified in the project at programming time.

Node Override IP	PLC operation
0.0.0.0	Communication with the controller is stopped, no request frames are generated anymore.
<b>Different from 0.0.0.0</b>	It is interpreted as node IP override and the target IP address is replaced runtime with the new value.

If the HMI device is connected to a network with more than one controller node, each node has its own Node Override IP variable.



Note: Node Override IP values assigned at runtime are retained through power cycles.

## Hostname DNS or mDNS

In addition to the array of bytes, string memory type can be selected to be able use the DNS or mDNS hostname as an alternative to the IP Address.

## Tag Import

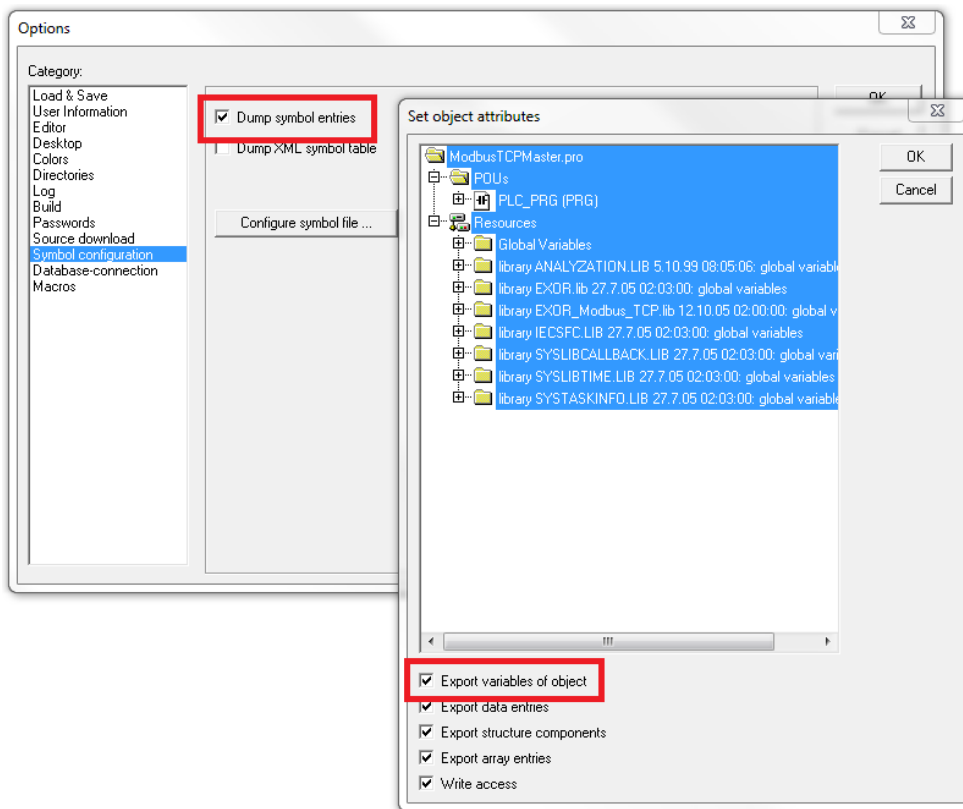
### Exporting Tags from PLC

When configuring PLC using the manufacturer's configuration software, enable Symbol file (.sym extension) creation under the CODESYS programming software:

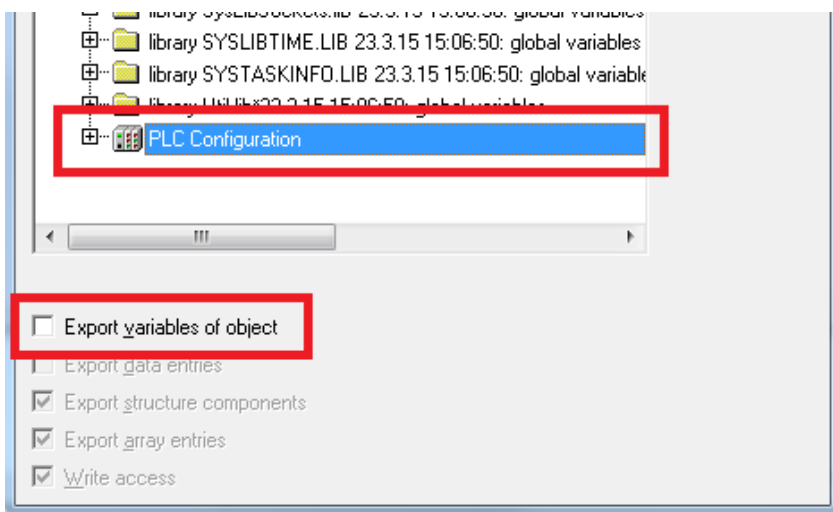
1. In the **Project** menu, click **Options**.
2. Click **Symbol configuration**.
3. Select **Dump symbol entries**.
4. Click **OK**.



Note: Click then **Configure symbol file...** and select **Export variables of object**. We recommend to clear the check box and re-select to be sure about the proper settings.

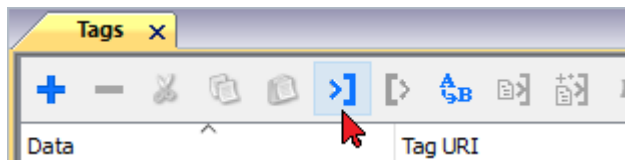


In some cases, duplication of symbols for variables associated to integrated I/O modules in the “.sym” file may be experienced. To remove the duplication selected the “PLC Configuration” voice from the objects list and uncheck the option “Export variables of object”.

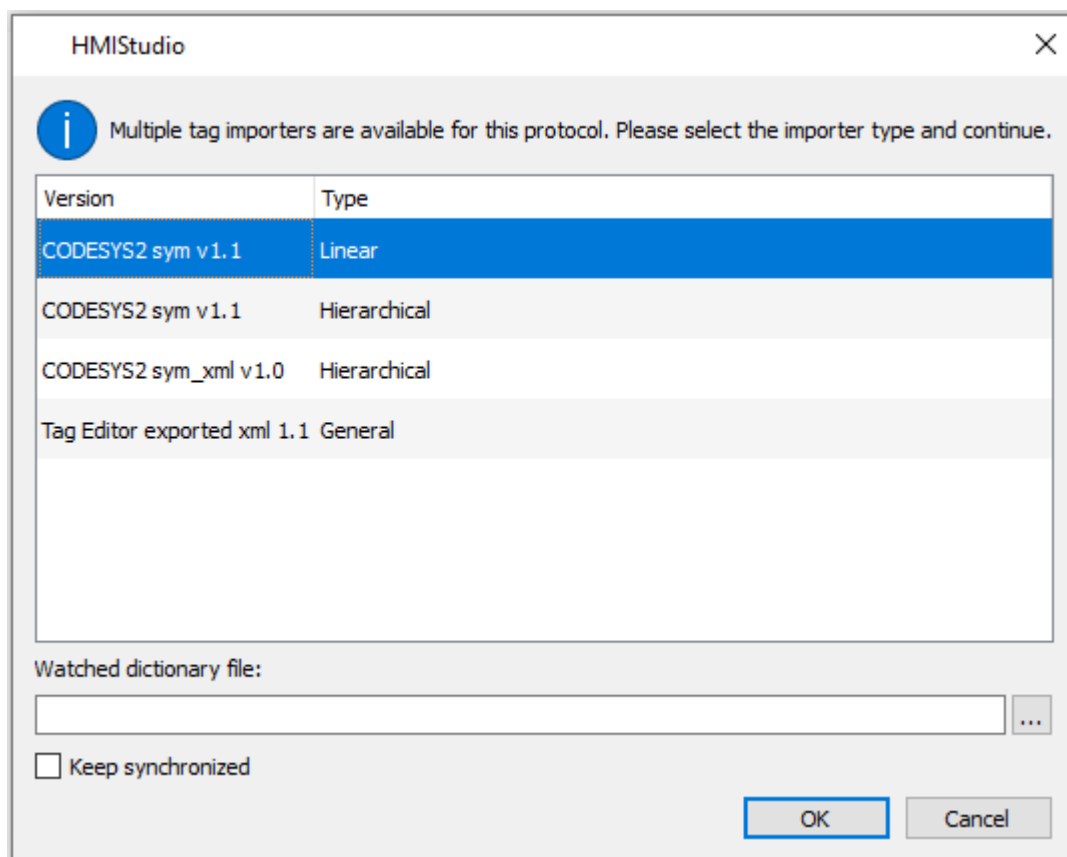


## Importing Tags in Tag Editor


Select the driver in Tag Editor and click on the **Import Tags** button to start the importer.



The following dialog shows which importer type can be selected.

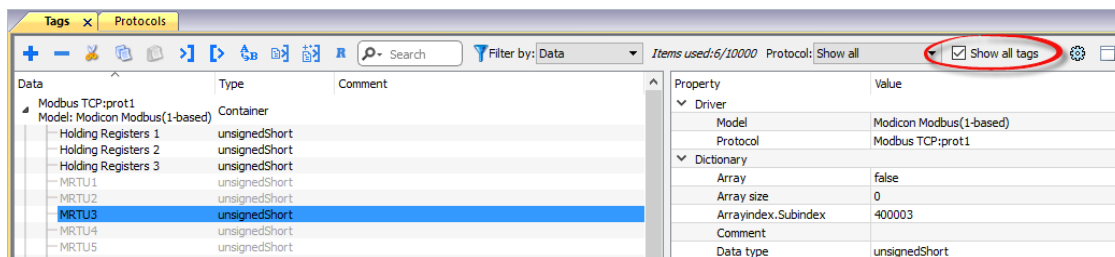





Importer	Description
<b>CODESYS2 sym v1.1 Linear</b>	Requires a <b>.sym</b> file. All variables will be displayed at the same level.
<b>CODESYS2 sym v1.1 Hierarchical</b>	Requires a <b>.sym</b> file. All variables will be displayed according to CODESYS V2 Hierarchical view.


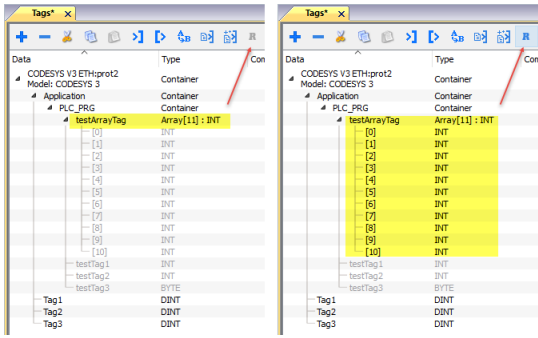

Importer	Description
<b>CODESYS2 sym_xml v1.0 Hierarchical</b>	Requires a <b>.sym_xml</b> file. All variables will be displayed according to CODESYS V2 Hierarchical view.
<b>Tag Editor exported xml</b>	Select this importer to read a generic XML file exported from Tag Editor by appropriate button. 

Once the importer has been selected, locate the symbol file and click **Open**.

The tags available within the Dictionary but not imported into the project are gray and are visible only when the "Show all tags" check box is selected.

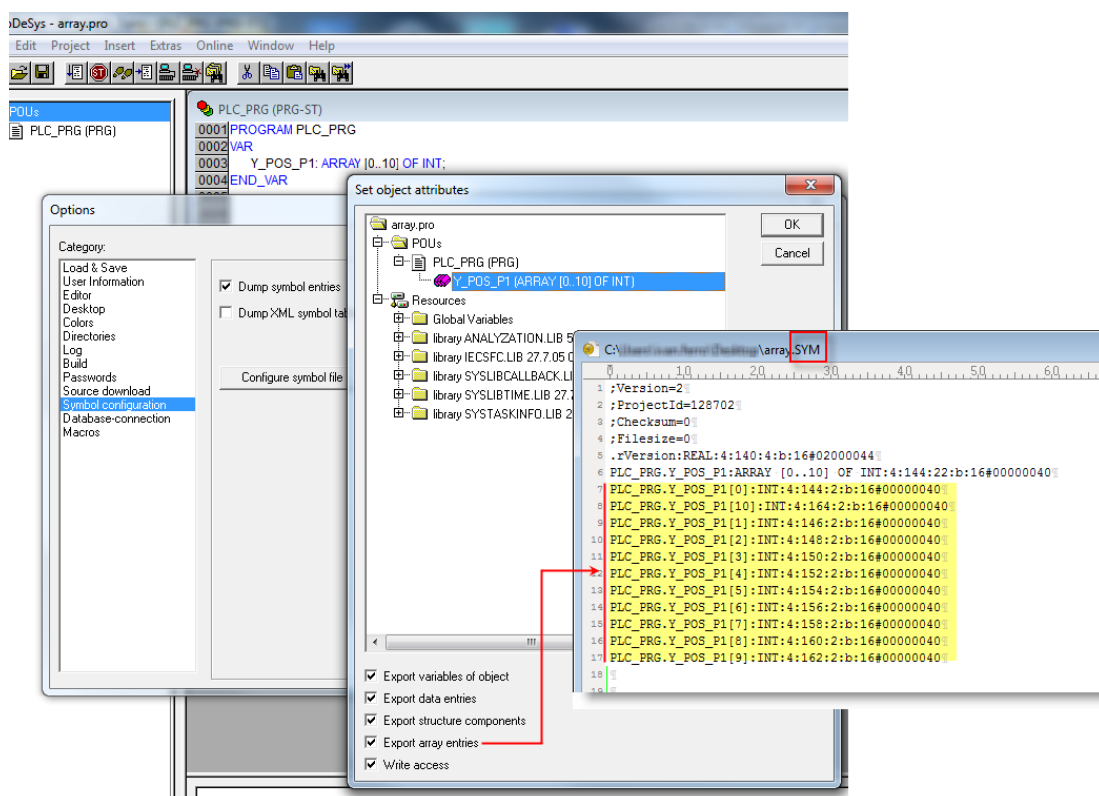


Toolbar item	Description
	<b>Import Tag(s).</b> Select tags to be imported and click on this icon to add tags from tag dictionary to the project
	<b>Update Tag(s).</b> Click on this icon to update the tags in the project, due a new dictionary import.
	Check this box to import all sub-elements of a tag. Example of both checked and unchecked result:

Toolbar item	Description
	
	<p>Searches tags in the dictionary basing on filter combobox item selected.</p>

## Exporting tag arrays

In CODESYS V2 program tag arrays are split into individual elements and one tag for each element is created. In the following example one array with 10 elements.



The screenshot shows the 'Set object attributes' dialog for the object 'Y\_POS\_P1 (ARRAY [0..10] OF INT)'. The 'Export array entries' checkbox is checked, and the 'SYM' file is highlighted in the file list. The resulting symbol table shows individual tags for each array element:

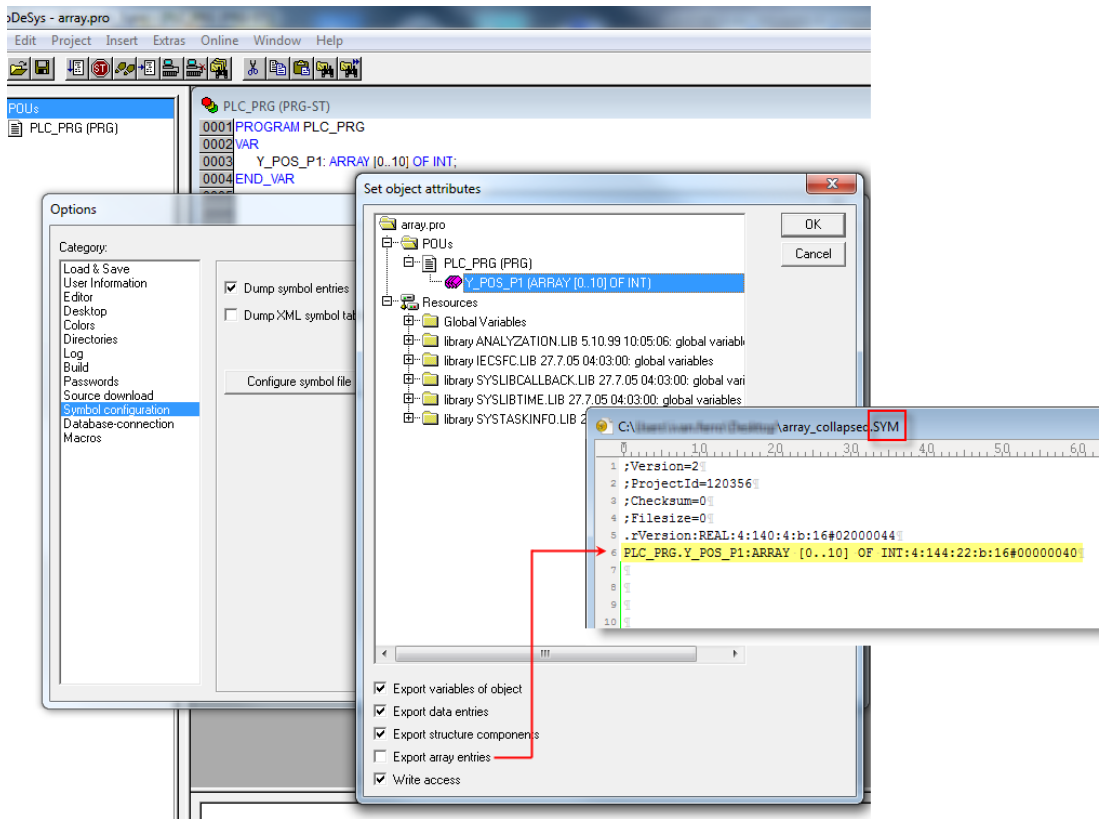
```

1 ;Version=2
2 ;ProjectId=128702
3 ;Checksum=0
4 ;Filesize=0
5 ;rVersion:REAL:4:140:4:b:16#02000044
6 PLC_PRG_Y_POS_P1:ARRAY [0..10] OF INT:4:144:22:b:16#00000040
7 PLC_PRG_Y_POS_P1[0]:INT:4:144:2:b:16#00000040
8 PLC_PRG_Y_POS_P1[10]:INT:4:164:2:b:16#00000040
9 PLC_PRG_Y_POS_P1[1]:INT:4:146:2:b:16#00000040
10 PLC_PRG_Y_POS_P1[2]:INT:4:148:2:b:16#00000040
11 PLC_PRG_Y_POS_P1[3]:INT:4:150:2:b:16#00000040
12 PLC_PRG_Y_POS_P1[4]:INT:4:152:2:b:16#00000040
13 PLC_PRG_Y_POS_P1[5]:INT:4:154:2:b:16#00000040
14 PLC_PRG_Y_POS_P1[6]:INT:4:156:2:b:16#00000040
15 PLC_PRG_Y_POS_P1[7]:INT:4:158:2:b:16#00000040
16 PLC_PRG_Y_POS_P1[8]:INT:4:160:2:b:16#00000040
17 PLC_PRG_Y_POS_P1[9]:INT:4:162:2:b:16#00000040
18
19

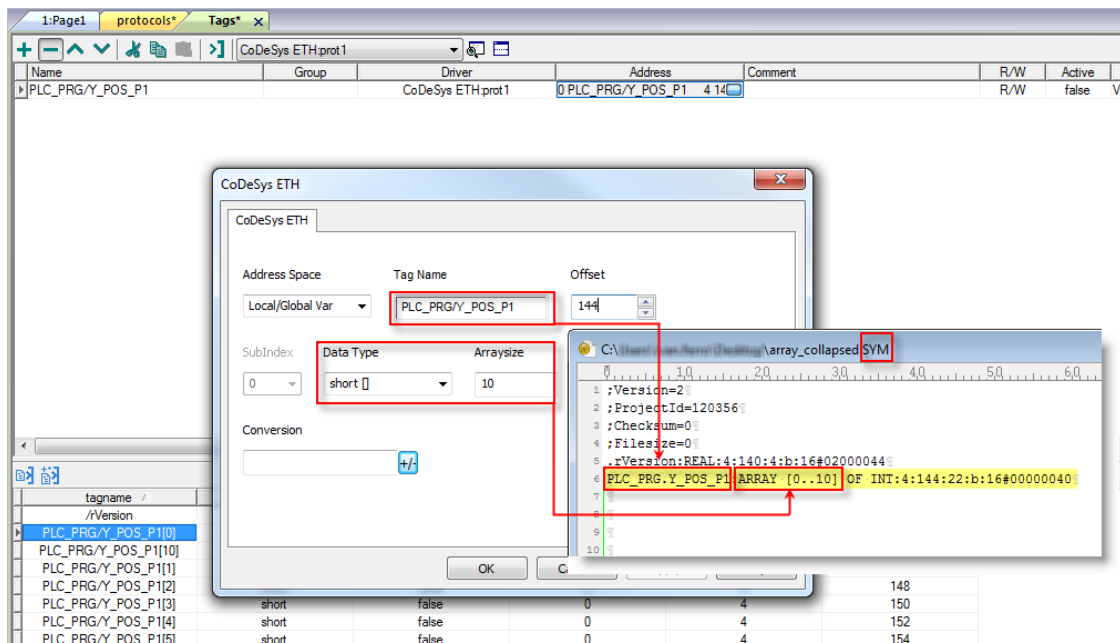
```

**Note:** If **Export array entries** is selected, a tag for each element will be created and exported into the .sym file. The entire tag list will be automatically imported into the Tag editor.

By clearing **Export array entries** only one tag for each one array can be created.

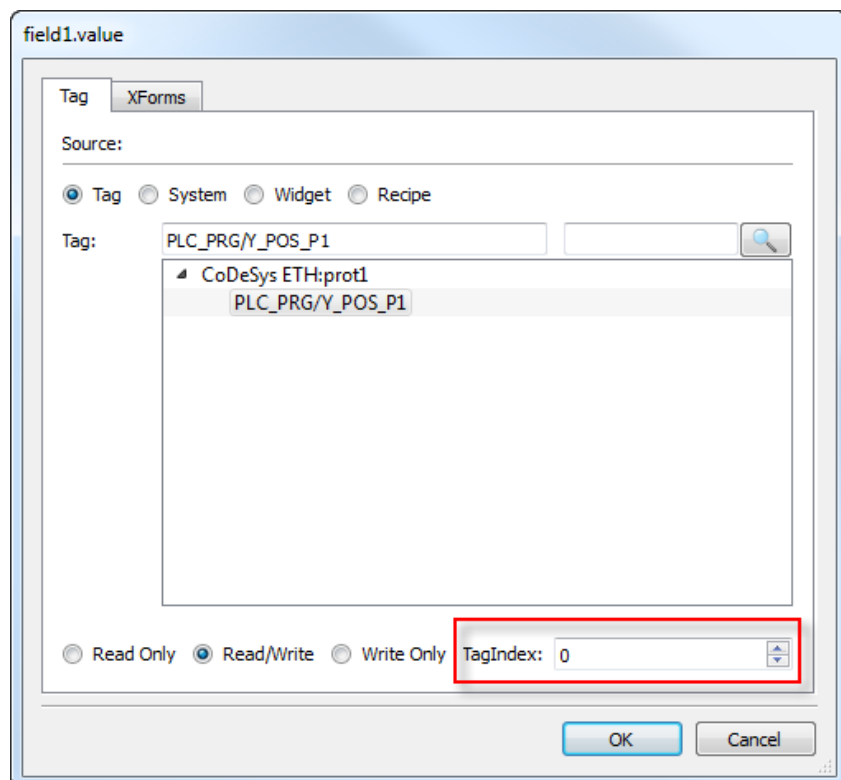


Note: When **Export array entries** has been cleared, only one tag is created and exported into the .sym file. The array is not automatically imported in the Tag editor and tags need to be manually configured in Tag editor.



All tag elements can be referenced in the editor using **TagIndex** in the **Attach to Tag** dialog.





## Communication status

Current communication status can be displayed using system variables. See "System Variables" section in the main manual.

Codes supported by this communication driver:

Error	Cause and action
<b>Symbols file not present</b>	Check Symbol file and download again the PLC program.
<b>"tag" not present in Symbols files</b>	Check if the Tag is present into the PLC project.
<b>Time out on Acknowledge</b>	Controller didn't send acknowledge.
<b>Time out on last Acknowledge</b>	Controller didn't sent last ack.
<b>Time out on data reciving</b>	Controller does not reply with data.
<b>Connection timeout</b>	Device not connected.

# CODESYS V3 ETH

The CODESYS V3 ETH communication driver supports communication through Ethernet connection with controllers based on the CODESYS V3 PLC software by the company 3S.



Note: To accommodate developments in the controller protocol and hardware, drivers are continuously updated. Make sure the latest driver is used in the application.



Note: Changes in the controller protocol or hardware may have occurred since this documentation was created. This may interfere with the functionality of this driver. Therefore, always test and verify the functionality of the application.

## Protocol Editor Settings

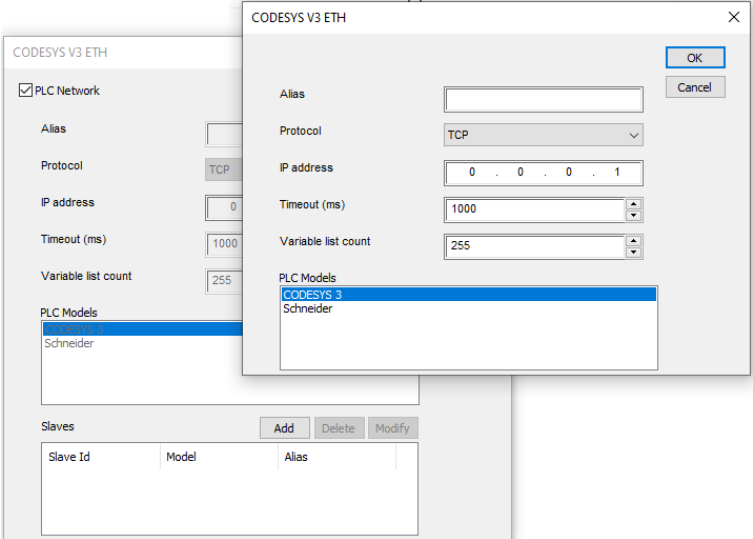
### Adding a protocol

To configure the protocol:

1. In **Config** node double-click **Protocols**.
2. To add a driver, click **+**: a new line is added.
3. Select the protocol from the **PLC** list.

The protocol configuration dialog is displayed.

Element	Description
<b>Alias</b>	Name to be used to identify nodes in network configurations. The name will be added as a prefix to each tag name imported for each network node.
<b>Protocol</b>	Select between TCP and UDP protocol type.
<b>IP address</b>	Ethernet IP address of the controller
<b>Variable</b>	Variable List is the best method to achieve higher performance in the CODESYS V3

Element	Description
<b>list count</b>	<p>communication protocol, as it allows requesting multiple data items in a single protocol session.</p> <p>Since some implementations of CODESYS V3 at runtime have a limited number of Variable Lists that can be allocated, this parameter allows you to set the maximum number of Variable Lists the communication driver tries to create in the PLC.</p>
<b>PLC Model</b>	Byte order that will be used by the communication driver when sending communication frames to the PLC.
<b>Timeout</b>	Number of milliseconds between retries when communication fails.
<b>PLC Network</b>	<p>Enable access to multiple networked controllers. For every controller (slave) set the proper option.</p> 



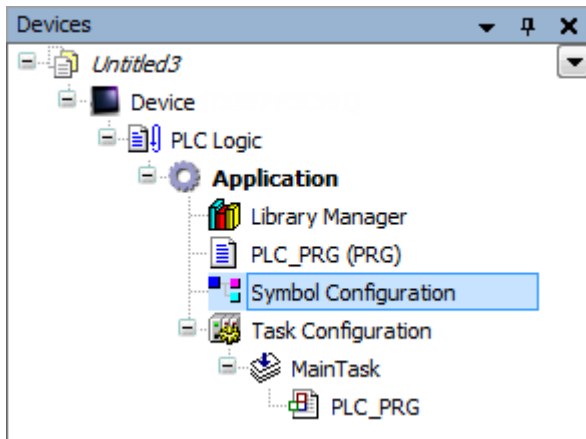
Note: Refer to the controller documentation to verify required values for the parameters **Full node address** or **Variable list count**.

## Tag Import

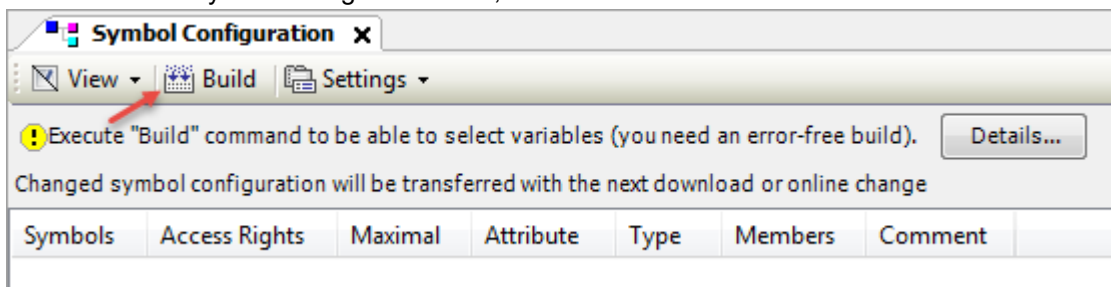
### Exporting Tags from PLC

When creating the project using CODESYS V3, properly configure the symbol file to contain the required variables.

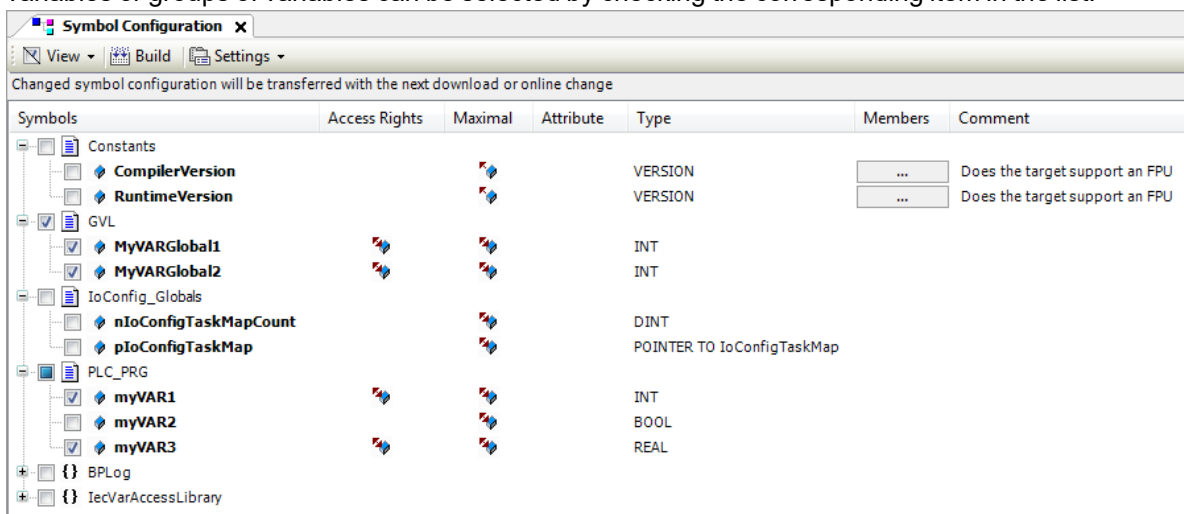
- To add the Symbol configuration in CODESYS V3 project, right click on the Application item from the project tree, then into the context menu select Add Object > Symbol configuration. The symbol configuration item will be added to the project tree.



2. Double click on Symbol configuration item, then click on "Build" button.



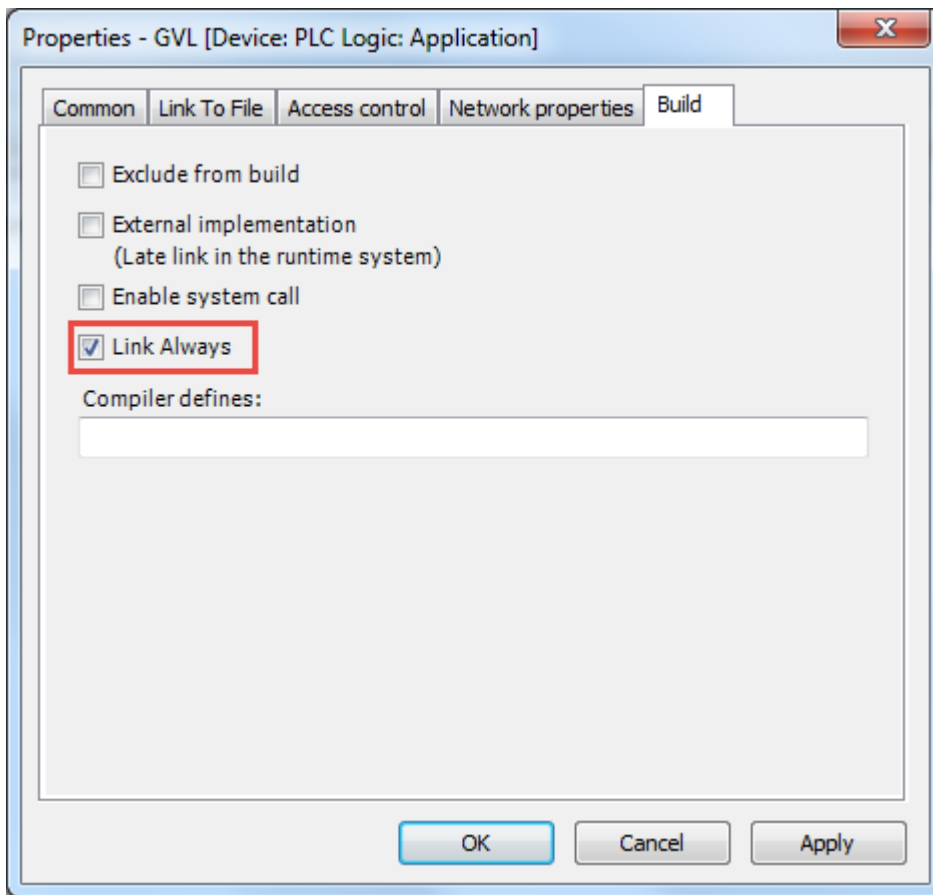
3. Symbol configuration item contains a list of all the variables available into the CODESYS V3 project, single variables or groups of variables can be selected by checking the corresponding item in the list.



4. After the symbols have been configured, download the project or use the **Generate code** function (Build > Generate code) to create an .xml file containing all the variables read to be imported in the Tag Editor.

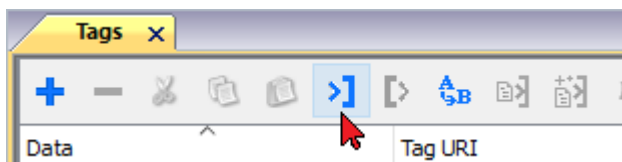


Note: GVL global variables are listed in Symbols Configuration only if they are used in PLC program. To always list global variables right click on GVL and select "Properties". From "Build" tab check "Link Always" option.

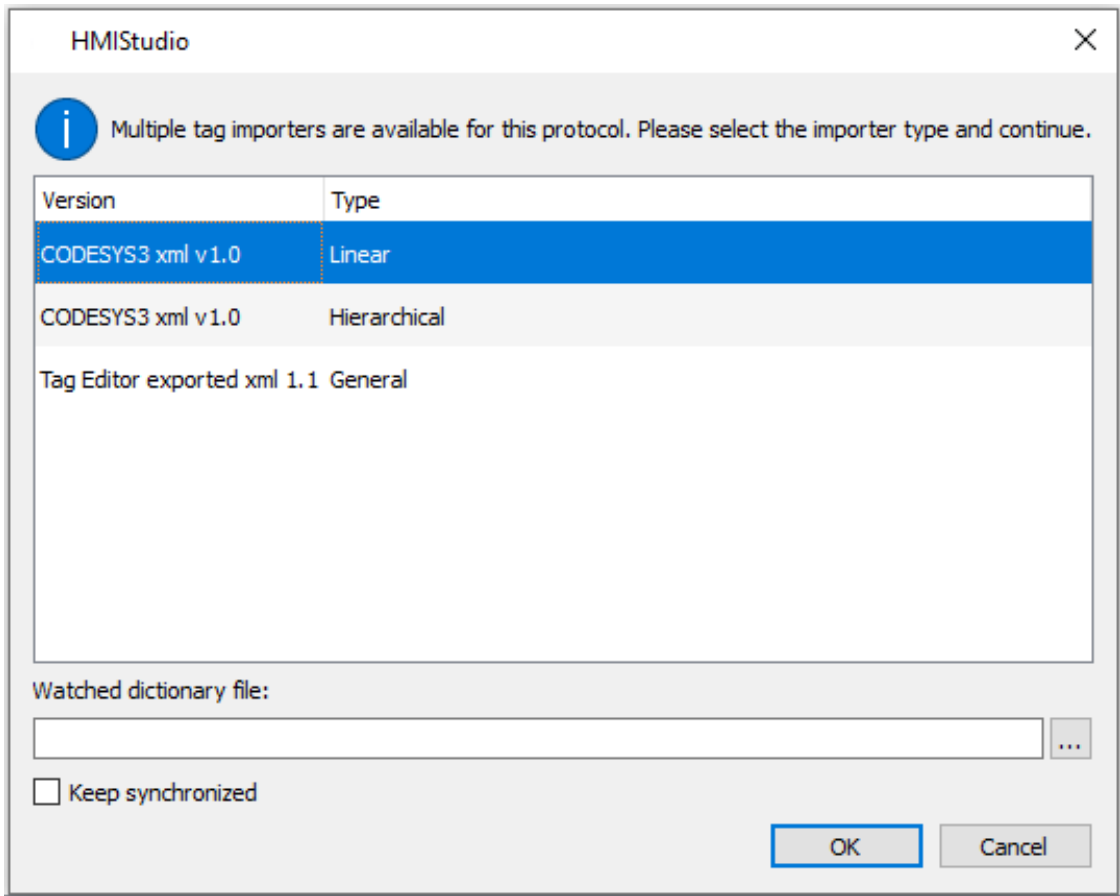


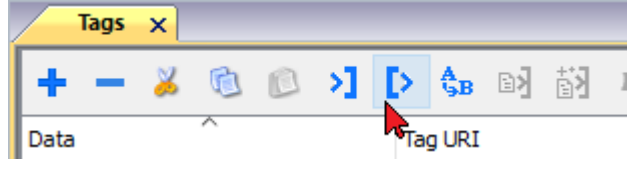
## Importing Tags in Tag Editor

Select the driver in Tag Editor and click on the **Import Tags** button to start the importer.



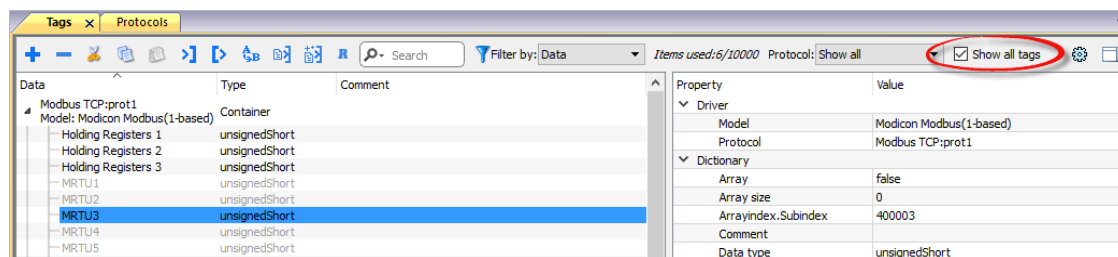
The following dialog shows which importer type can be selected.




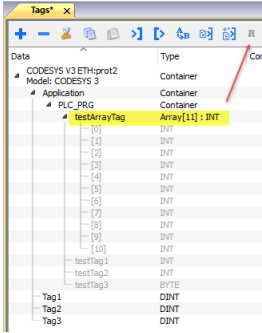
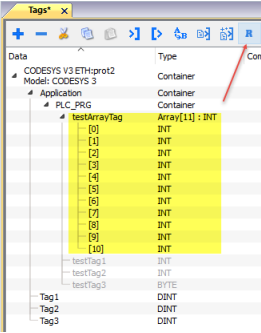
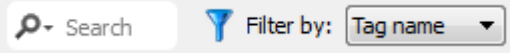


Importer	Description
<b>CODESYS3 xml v1.0 Linear</b>	Requires an <b>.xml</b> file. All variables will be displayed at the same level.
<b>CODESYS3 xml v1.0 Hierarchical</b>	Requires an <b>.xml</b> file. All variables will be displayed according to CODESYS V3 Hierarchical view.
<b>Tag Editor exported xml</b>	Select this importer to read a generic XML file exported from Tag Editor by appropriate button. 

Once the importer has been selected, locate the symbol file and click **Open**.

The tags available within the Dictionary but not imported into the project are gray and are visible only when the "Show all tags" check box is selected.



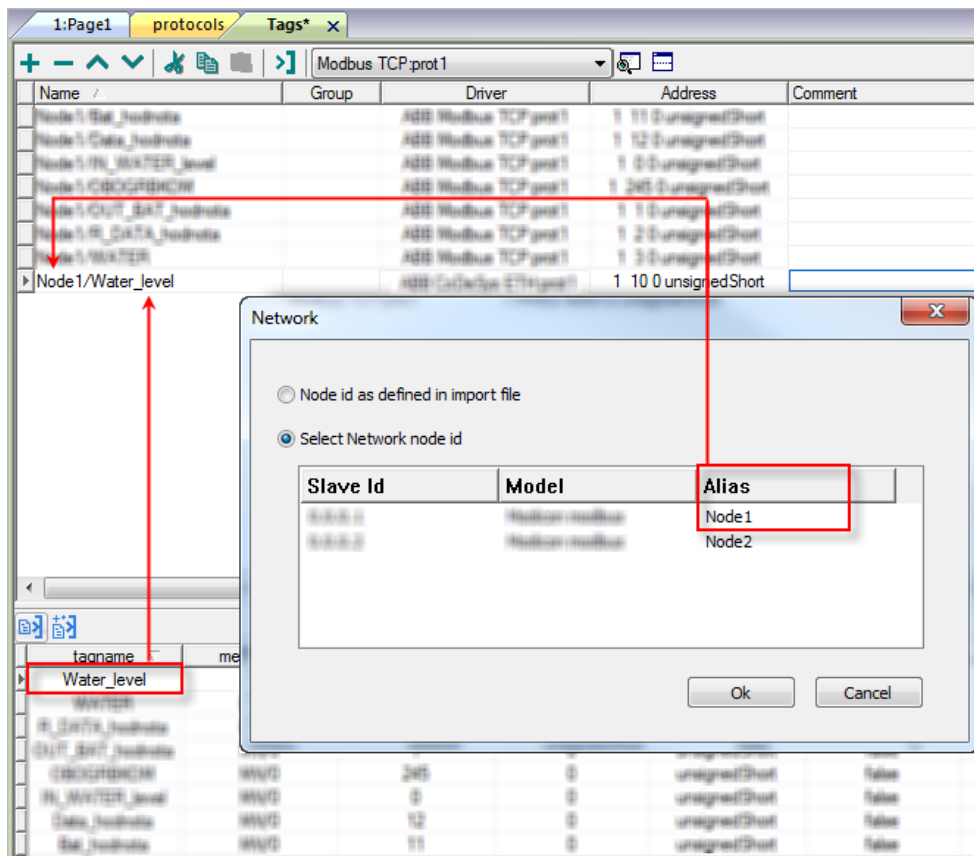
Toolbar item	Description
	<p><b>Import Tag(s).</b></p> <p>Select tags to be imported and click on this icon to add tags from tag dictionary to the project</p>
	<p><b>Update Tag(s).</b></p> <p>Click on this icon to update the tags in the project, due a new dictionary import.</p>
	<p>Check this box to import all sub-elements of a tag.</p> <p>Example of both checked and unchecked result:</p> <div style="display: flex; justify-content: space-around;">   </div>
	<p>Searches tags in the dictionary basing on filter combobox item selected.</p>

## Adding an alias name to a protocol

Tag names must be unique at project level, however, the same tag names might need to be used for different controller nodes (for example when the HMI device is connected to two devices running the same application).

When creating a protocol you can add an alias name that will be added to tag names imported for this protocol.

In the example, the connection to a certain controller is assigned the name **Node1**. When tags are imported for this node, all tag names will have the prefix **Node1** making each of them unique at the network/project level.



Note: Aliasing tag names is only available for imported tags. Tags added manually in the Tag Editor cannot have the Alias prefix in the tag name. The Alias string is attached at the time of tag import. If you modify the Alias string after the tag import has been completed, there will be no effect on names already present in the dictionary. When the Alias string is changed and tags are re-imported, all tags will be re-imported with the new prefix string.

## Data Types

The import module supports variables of standard data types and user defined data types.



**Supported data types**

- BOOL
- INT
- SINT
- UINT
- UDINT
- DINT
- STRING\*
- REAL
- LREAL
- BYTE
- ULINT
- LINT

and 1-dimensional ARRAY of the types above. See "Programming concepts" section in the main manual.



Note \*: String length for a STRING variable in PLC should be max 80 characters. Declare a STRING variable either with a specified size (str: STRING(35) or default size (str: STRING) which is 80 characters.

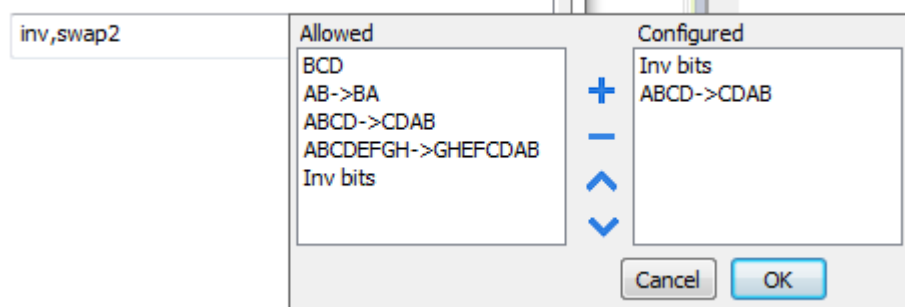
**Unsupported data types**

- LWORD
- LINT

**Tag conversion**

Conversion to be applied to the tag.

Conversion



Depending on data type selected, the list **Allowed** shows one or more conversion types.

Value	Description
<b>Inv bits</b>	<p><b>inv</b>: Invert all the bits of the tag.</p> <p><i>Example:</i>            1001 → 0110 (in binary format)            9 → 6 (in decimal format)</p>
<b>Negate</b>	<p><b>neg</b>: Set the opposite of tag value.</p> <p><i>Example:</i>            25.36 → -25.36</p>
<b>AB -&gt; BA</b>	<p><b>swapnibbles</b>: Swap nibbles in a byte.</p> <p><i>Example:</i>            15D4 → 514D (in hexadecimal format)            5588 → 20813 (in decimal format)</p>
<b>ABCD -&gt; CDAB</b>	<p><b>swap2</b>: Swap bytes in a word.</p> <p><i>Example:</i>            9ACC → CC9A (in hexadecimal format)            39628 → 52378 (in decimal format)</p>
<b>ABCDEFGH -&gt; GHEFCDAB</b>	<p><b>swap4</b>: Swap bytes in a double word.</p> <p><i>Example:</i>            32FCFF54 → 54FFFC32 (in hexadecimal format)            855441236 → 1426062386 (in decimal format)</p>
<b>ABC...NOP -&gt; OPM...DAB</b>	<p><b>swap8</b>: Swap bytes in a long word.</p> <p><i>Example:</i>            142.366 → -893553517.588905 (in decimal format)            0 10000000110            0001110010111011011001000101101000011100101011000001            →            1 10000011100            1010101000010100010110110110110010110110000100111101            (in binary format)</p>
<b>BCD</b>	<p><b>bcd</b>: Separate byte in two nibbles, read them as decimal (from 0 to 9)</p> <p><i>Example:</i>            23 → 17 (in decimal format)            0001 0111 = 23            0001 = 1 (first nibble)            0111 = 7 (second nibble)</p>

Select conversion and click +. The selected item will be added to list **Configured**.

If more conversions are configured, they will be applied in order (from top to bottom of list **Configured**).

Use the arrow buttons to order the configured conversions.

## Node Override IP

The protocol provides the special data type Node Override IP which allows you to change the IP address of the target controller at runtime.

This memory type is an array of 4 unsigned bytes, one per each byte of the IP address.

The Node Override IP is initialized with the value of the controller IP specified in the project at programming time.

Node Override IP	PLC operation
0.0.0.0	Communication with the controller is stopped, no request frames are generated anymore.
Different from 0.0.0.0	It is interpreted as node IP override and the target IP address is replaced runtime with the new value.

If the HMI device is connected to a network with more than one controller node, each node has its own Node Override IP variable.



Note: Node Override IP values assigned at runtime are retained through power cycles.

## Hostname DNS or mDNS

In addition to the array of bytes, string memory type can be selected to be able use the DNS or mDNS hostname as an alternative to the IP Address.

## Application Status

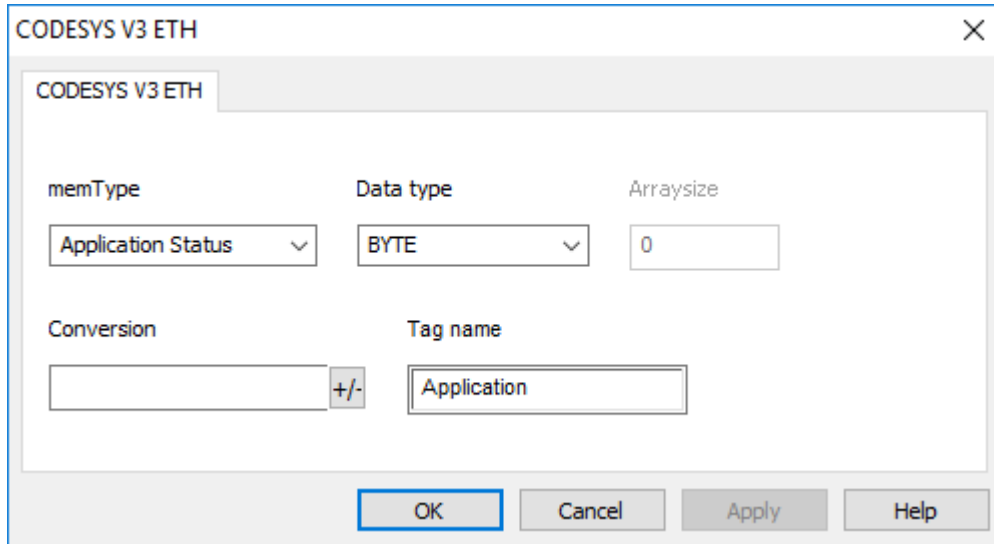
The protocol provides the special data type Application Status which allows you to check or change the applications status.




Functionality available only if supported by the CODESYS device

The tags pointing to Application Status must contains into field "**Tag name**" the name of the PLC application (frequently the default name is "Application")

If the HMI device is connected to a network with more than one controller node, each node has its own Application Status variable.



Application Status	Description
0	RUNNING
1	STOPPED
2	HALTED ON BreakPoint  It is not possible to write 2 as new status
251	Reboot CODESYS device
252	Shutdown CODESYS
253	Reset ORIGIN
254	Reset COLD
255	Reset WARM

## Communication Status

Current communication status can be displayed using System Variables. See "System Variables" section in the main manual.

# CODESYS V3 Handler

The CODESYS V3 Handler communication driver supports communication through Ethernet connection with controllers based on the CODESYS V3 PLC software by the company 3S, through CODESYS PLC Handler.



Note: To accommodate developments in the controller protocol and hardware, drivers are continuously updated. Make sure the latest driver is used in the application.



Note: Changes in the controller protocol or hardware may have occurred since this documentation was created. This may interfere with the functionality of this driver. Therefore, always test and verify the functionality of the application.

## Protocol Editor Settings

### Adding a protocol

To configure the protocol:

1. In **Config** node double-click **Protocols**.
2. To add a driver, click **+**: a new line is added.
3. Select the protocol from the **PLC** list.

The protocol configuration dialog is displayed.

CODESYS V3 Handler

PLC Network

Alias

IP address

Timeout (ms)

Variable list count

User name

Password

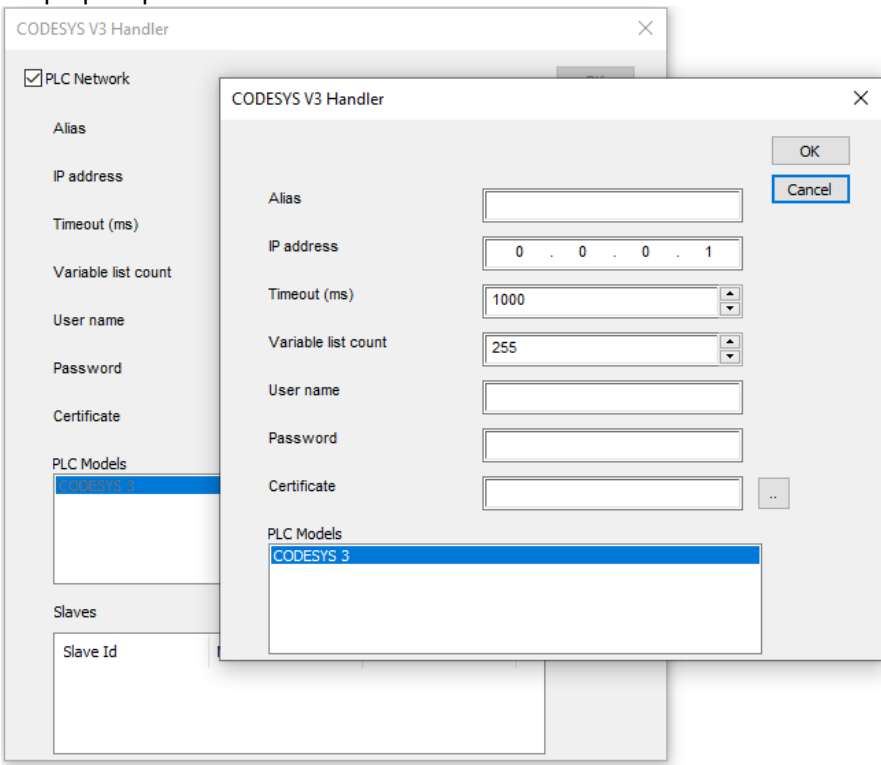
Certificate


PLC Models

CODESYS 3

OK

Cancel

Element	Description
<b>Alias</b>	Name to be used to identify nodes in network configurations. The name will be added as a prefix to each tag name imported for each network node.
<b>IP address</b>	Ethernet IP address of the controller
<b>Timeout (ms)</b>	Number of milliseconds between retries when communication fails.
<b>Variable list count</b>	<p>Variable List is the best method to achieve higher performance in the CODESYS V3 communication protocol, as it allows requesting multiple data items in a single protocol session.</p> <p>Since some implementations of CODESYS V3 at runtime have a limited number of Variable Lists that can be allocated, this parameter allows you to set the maximum number of Variable Lists the communication driver tries to create in the PLC.</p>
<b>User name</b>	User name set in CODESYS project. Needed when CODESYS project has security enabled.
<b>Password</b>	Password in CODESYS project. Needed when CODESYS project has security enabled.
<b>Certificate</b>	Certificate generated by CODESYS runtime. Needed when CODESYS project has security enabled with encryption.
<b>PLC Model</b>	Model of driver. Placeholder for eventual future specializations.
<b>PLC Network</b>	<p>Enable access to multiple networked controllers. For every controller (slave) set the proper option.</p> 

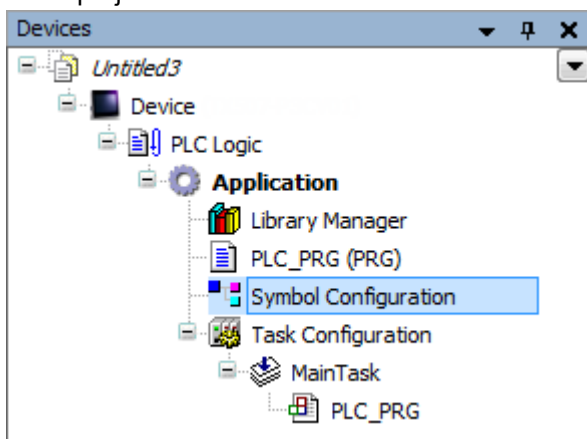
 Note: Refer to the controller documentation to verify required values for the parameters **Full node address** or **Variable list count**.

## Tag Import

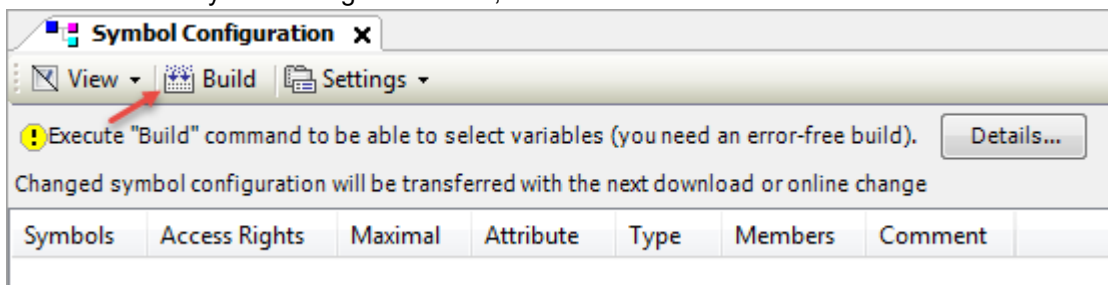
### Exporting Tags from PLC

When creating the project using CODESYS V3, properly configure the symbol file to contain the required variables.

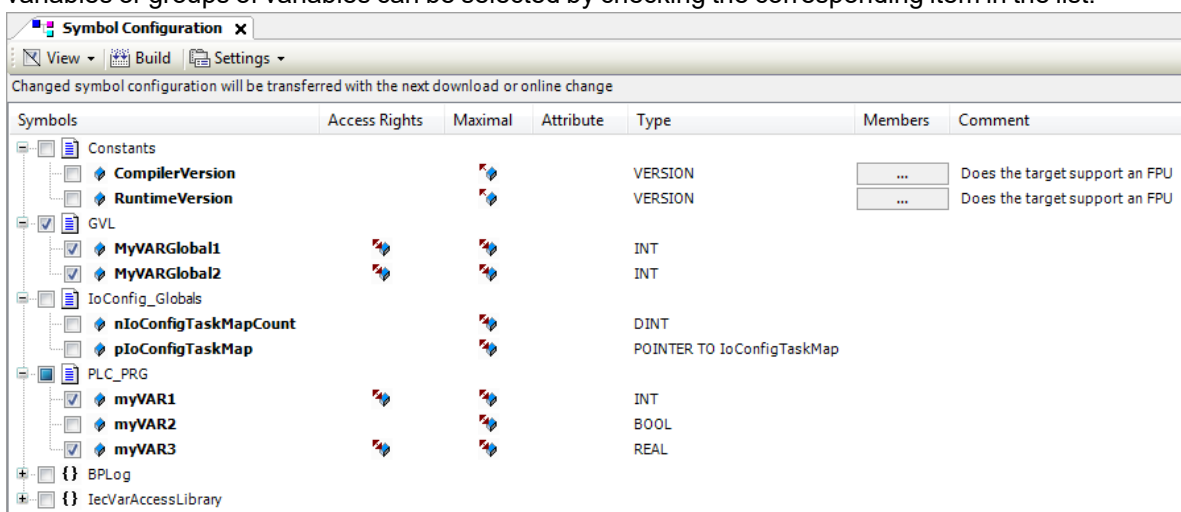
1. To add the Symbol configuration in CODESYS V3 project, right click on the Application item from the project tree, then into the context menu select Add Object > Symbol configuration. The symbol configuration item will be added to the project tree.



2. Double click on Symbol configuration item, then click on "Build" button.



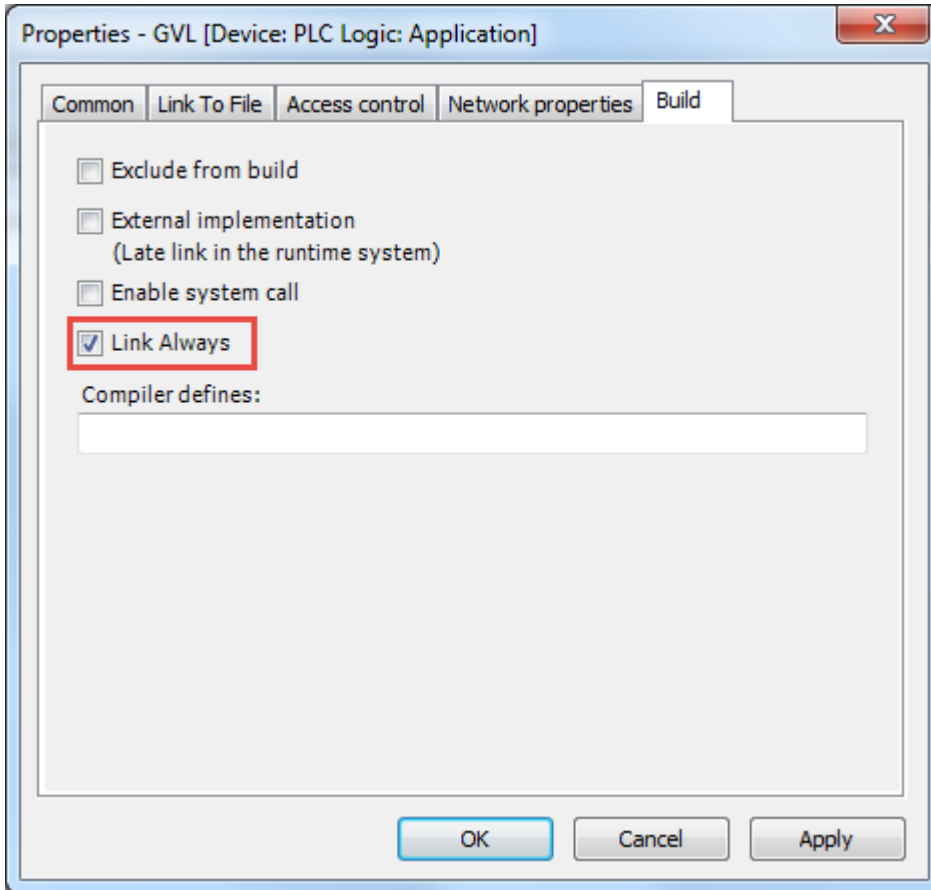
3. Symbol configuration item contains a list of all the variables available into the CODESYS V3 project, single variables or groups of variables can be selected by checking the corresponding item in the list.



- After the symbols have been configured, download the project or use the **Generate code** function (Build > Generate code) to create an .xml file containing all the variables read to be imported in the Tag Editor.

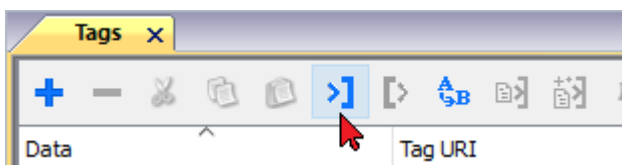


Note: GVL global variables are listed in Symbols Configuration only if they are used in PLC program. To always list global variables right click on GVL and select "Properties". From "Build" tab check "Link Always" option.



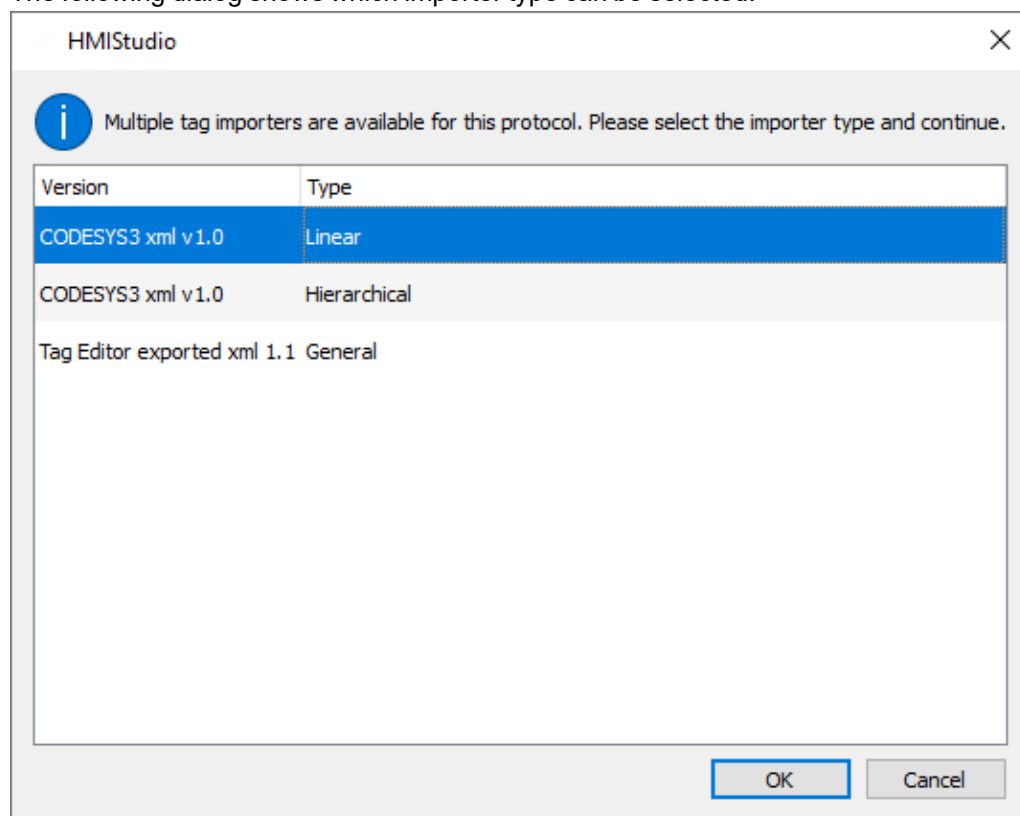
## Importing Tags in Tag Editor


Select the driver in Tag Editor and click on the **Import Tags** button to start the importer.





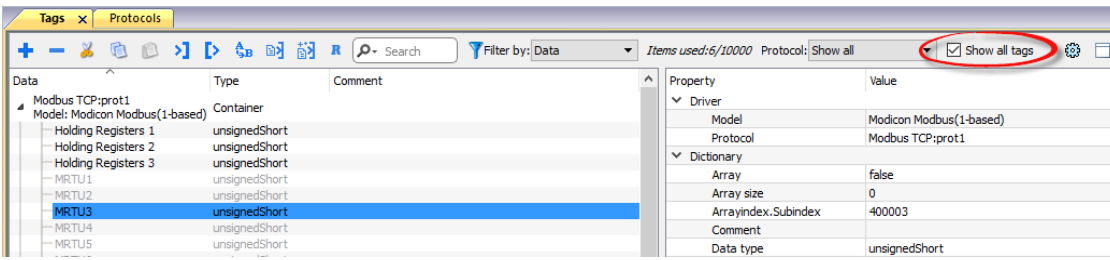
The following dialog shows which importer type can be selected.

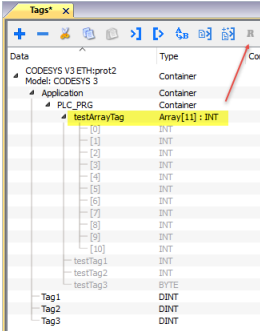
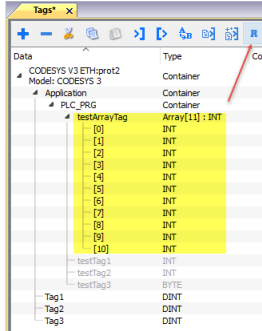


Importer	Description
<b>CODESYS3 xml v1.0 Linear</b>	Requires an <b>.xml</b> file. All variables will be displayed at the same level.
<b>CODESYS3 xml v1.0 Hierarchical</b>	Requires an <b>.xml</b> file. All variables will be displayed according to CODESYS V3 Hierarchical view.
<b>Tag Editor exported xml</b>	Select this importer to read a generic XML file exported from Tag Editor by appropriate button. 

Once the importer has been selected, locate the symbol file and click **Open**.

The tags available within the Dictionary but not imported into the project are gray and are visible only when the "Show all tags" check box is selected.



Toolbar item	Description
	<p><b>Import Tag(s).</b></p> <p>Select tags to be imported and click on this icon to add tags from tag dictionary to the project</p>
	<p><b>Update Tag(s).</b></p> <p>Click on this icon to update the tags in the project, due a new dictionary import.</p>
	<p>Check this box to import all sub-elements of a tag.</p> <p>Example of both checked and unchecked result:</p> <div style="display: flex; justify-content: space-around;">   </div>
	<p>Searches tags in the dictionary basing on filter combo-box item selected.</p>

## Adding an alias name to a protocol

Tag names must be unique at project level, however, the same tag names might need to be used for different controller nodes (for example when the HMI device is connected to two devices running the same application).

When creating a protocol you can add an alias name that will be added to tag names imported for this protocol.

In the example, the connection to a certain controller is assigned the name **Node1**. When tags are imported for this node, all tag names will have the prefix **Node1** making each of them unique at the network/project level.

The screenshot shows the 'Tags\*' window in CODESYS. A table lists tags with columns: Name, Group, Driver, Address, and Comment. The tag 'Node1/Water\_Level' is selected. A 'Network' dialog box is open, showing options for node ID assignment. The 'Select Network node id' option is chosen. Below this is a table with columns 'Slave Id', 'Model', and 'Alias'. The 'Alias' column contains 'Node1' and 'Node2'. Red boxes highlight the 'Water\_Level' tag in the main window and the 'Alias' column in the dialog. Red arrows point from the 'Water\_Level' tag to the 'Node1' alias in the dialog.

Name	Group	Driver	Address	Comment
Node1/Bat_InstData		ADD Modbus TCP prot1	1 11 0 unsignedShort	
Node1/Data_InstData		ADD Modbus TCP prot1	1 12 0 unsignedShort	
Node1/IN_WATER_Level		ADD Modbus TCP prot1	1 0 0 unsignedShort	
Node1/OBSCUREDCON		ADD Modbus TCP prot1	1 245 0 unsignedShort	
Node1/OUT_BAT_InstData		ADD Modbus TCP prot1	1 1 0 unsignedShort	
Node1/R_DATA_InstData		ADD Modbus TCP prot1	1 2 0 unsignedShort	
Node1/WATER_Level		ADD Modbus TCP prot1	1 3 0 unsignedShort	

Slave Id	Model	Alias
Node1	Modbus modbus	Node1
Node2	Modbus modbus	Node2

**Note:** Aliasing tag names is only available for imported tags. Tags added manually in the Tag Editor cannot have the Alias prefix in the tag name. The Alias string is attached at the time of tag import. If you modify the Alias string after the tag import has been completed, there will be no effect on names already present in the dictionary. When the Alias string is changed and tags are re-imported, all tags will be re-imported with the new prefix string.

## Data Types

The import module supports variables of standard data types and user defined data types.

### Supported data types

- BOOL
- INT
- SINT
- UINT
- UDINT
- DINT
- STRING\*
- REAL
- LREAL
- BYTE
- ULINT
- LINT

and 1-dimensional ARRAY of the types above. See "Programming concepts" section in the main manual.



Note \*: String length for a STRING variable in PLC should be max 80 characters. Declare a STRING variable either with a specified size (str: STRING(35)) or default size (str: STRING) which is 80 characters.

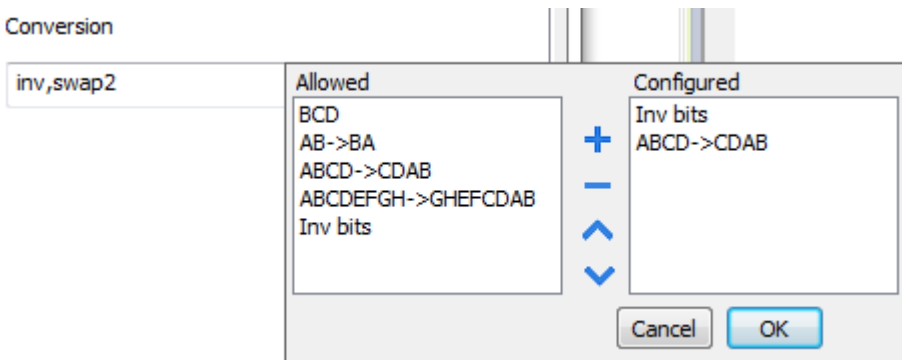
### Unsupported data types

- LWORD
- LINT

## Tag conversion

Conversion to be applied to the tag.

Conversion



Depending on data type selected, the list **Allowed** shows one or more conversion types.

Value	Description
<b>Inv bits</b>	<b>inv</b> : Invert all the bits of the tag. <i>Example:</i> 1001 → 0110 (in binary format) 9 → 6 (in decimal format)
<b>Negate</b>	<b>neg</b> : Set the opposite of tag value. <i>Example:</i> 25.36 → -25.36
<b>AB -&gt; BA</b>	<b>swapnibbles</b> : Swap nibbles in a byte. <i>Example:</i> 15D4 → 514D (in hexadecimal format) 5588 → 20813 (in decimal format)
<b>ABCD -&gt; CDAB</b>	<b>swap2</b> : Swap bytes in a word. <i>Example:</i> 9ACC → CC9A (in hexadecimal format) 39628 → 52378 (in decimal format)
<b>ABCDEFGH -&gt; GHEFC DAB</b>	<b>swap4</b> : Swap bytes in a double word. <i>Example:</i> 32FCFF54 → 54FFFC32 (in hexadecimal format) 855441236 → 1426062386 (in decimal format)
<b>ABC...NOP -&gt; OPM...DAB</b>	<b>swap8</b> : Swap bytes in a long word. <i>Example:</i> 142.366 → -893553517.588905 (in decimal format) 0 10000000110 0001110010111011011001000101101000011100101011000001 → 1 10000011100 1010101000010100010110110110110010110110000100111101 (in binary format)
<b>BCD</b>	<b>bcd</b> : Separate byte in two nibbles, read them as decimal (from 0 to 9) <i>Example:</i> 23 → 17 (in decimal format) 0001 0111 = 23 0001 = 1 (first nibble) 0111 = 7 (second nibble)

Select conversion and click +. The selected item will be added to list **Configured**.

If more conversions are configured, they will be applied in order (from top to bottom of list **Configured**).

Use the arrow buttons to order the configured conversions.

## Node Override IP

The protocol provides the special data type Node Override IP which allows you to change the IP address of the target controller at runtime.

This memory type is an array of 4 unsigned bytes, one per each byte of the IP address.

The Node Override IP is initialized with the value of the controller IP specified in the project at programming time.

Node Override IP	PLC operation
0.0.0.0	Communication with the controller is stopped, no request frames are generated anymore.
Different from 0.0.0.0	It is interpreted as node IP override and the target IP address is replaced runtime with the new value.

If the HMI device is connected to a network with more than one controller node, each node has its own Node Override IP variable.



Note: Node Override IP values assigned at runtime are retained through power cycles.

## Hostname DNS or mDNS

In addition to the array of bytes, string memory type can be selected to be able use the DNS or mDNS hostname as an alternative to the IP Address.

CODESYS V3 ETH

memType Data type Arraysizes

Node override IP BYTE[] 4

Conversion +/- Tag name

OK Cancel Apply Help

## Application Status


The protocol provides the special data type Application Status which allows you to check or change the applications status.



Functionality available only if supported by the CODESYS device

The tags pointing to Application Status must contains into field "**Tag name**" the name of the PLC application (frequently the default name is "Application")

If the HMI device is connected to a network with more than one controller node, each node has its own Application Status variable.

Application Status	Description
0	RUNNING
1	STOPPED
2	HALTED ON BreakPoint  It is not possible to write 2 as new status
251	Reboot CODESYS device
252	Shutdown CODESYS
253	Reset ORIGIN
254	Reset COLD
255	Reset WARM

## Communication Status

Current communication status can be displayed using System Variables. See "System Variables" section in the main manual.

# Environment Variables

This protocol gives the possibility to copy the environment variables of the hosting Operative System inside tags. All variables will be read only, namely, is not possible to modify them.



Environment Variables communication driver is not counted as physical protocol.  
Refer to **Table of functions and limits** from main manual in "Number of physical protocols" line.

## Protocol Editor Settings

### Adding a protocol

To configure the protocol:

1. In the **Config** node double-click **Protocols**.
2. To add a driver, click **+**: a new line is added.
3. Select the **Environment Variables** protocol from the **PLC** list.

## Tag Editor Settings

*Path: ProjectView > Config > double-click Tags*

1. To add a tag, click **+**: a new line is added.
2. Select **Environment Variables** from the protocol list: tag definition dialog is displayed.

Element	Description
<b>Name</b>	Name of the environment variable that you want to read.
<b>Data Type</b>	System variables are of type string, but if a different type is chosen, e.g. int, casting to the chosen type will be made.
<b>Arraysize</b>	This property represents the maximum number of bytes available in the string or in the array Tag.



# Ethernet/IP CIP

The protocol has been implemented according to the published Ethernet/IP specifications (available from [www.odva.org](http://www.odva.org)).

The Ethernet/IP CIP driver has been designed to provide the best performance with the least amount of impact on the system's overall performance. Although the Ethernet/IP CIP driver is fast, we suggest to use short Tag names. Tags are read from and written to the device by specifying their symbolic name in the communications request, therefore the longer the tag name is, the larger the request will be.

## Protocol Editor Settings

### Adding a protocol

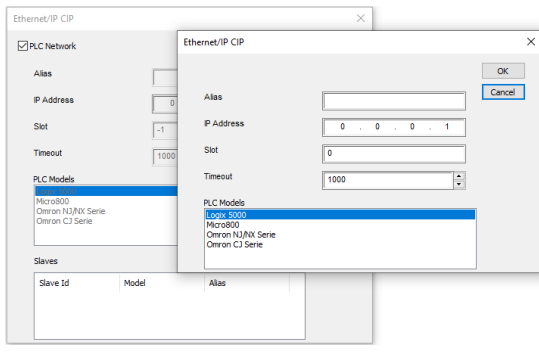
To configure the protocol:

1. In **Config** node double-click **Protocols**.
2. To add a driver, click **+**: a new line is added.
3. Select the protocol from the **PLC** list.

The protocol configuration dialog is displayed.

Field	Description
<b>Alias</b>	Name identifying nodes in network configurations. The name will be added as a prefix to each tag name imported for each network node.
<b>IP Address</b>	Ethernet IP address of the controller.
<b>Slot</b>	CPU slot number for Logix 5000 models (typically 0). Refer to the controller documentation for further details.

Field	Description
<b>PLC Models</b>	PLC model used to import tags file.
<b>PLC Network</b>	Enable access to multiple networked controllers. For every controller (slave) set the proper option.



## Controller Model Logix 5000

The Ethernet/IP CIP driver allows to connect Allen-Bradley ControlLogix and CompactLogix Ethernet controllers.

Communication with ControlLogix® 5500 controllers can be accomplished through an Ethernet/IP communication module for Ethernet such as the 1756-EN2T or 1756-ENET.

Ethernet communication with CompactLogix™ 5300 controllers requires a processor with a built-in Ethernet/IP port such as the 1769-L32E.

All trademarks are the property of their respective owners.

The internal memory organization of the Logix CPUs is not fixed but configured by the user at development time. Each data item can be identified by a string called “Tag”. The RSLogix 5000 software can then export to the application the list of Tags created for each controller.

The project loaded on the HMI device must refer to Tag names assigned in RSLogix 5000 software at development time. The Tag Editor supports direct import of the Tag file generated by RSLogix 5000 software in .CSV format.

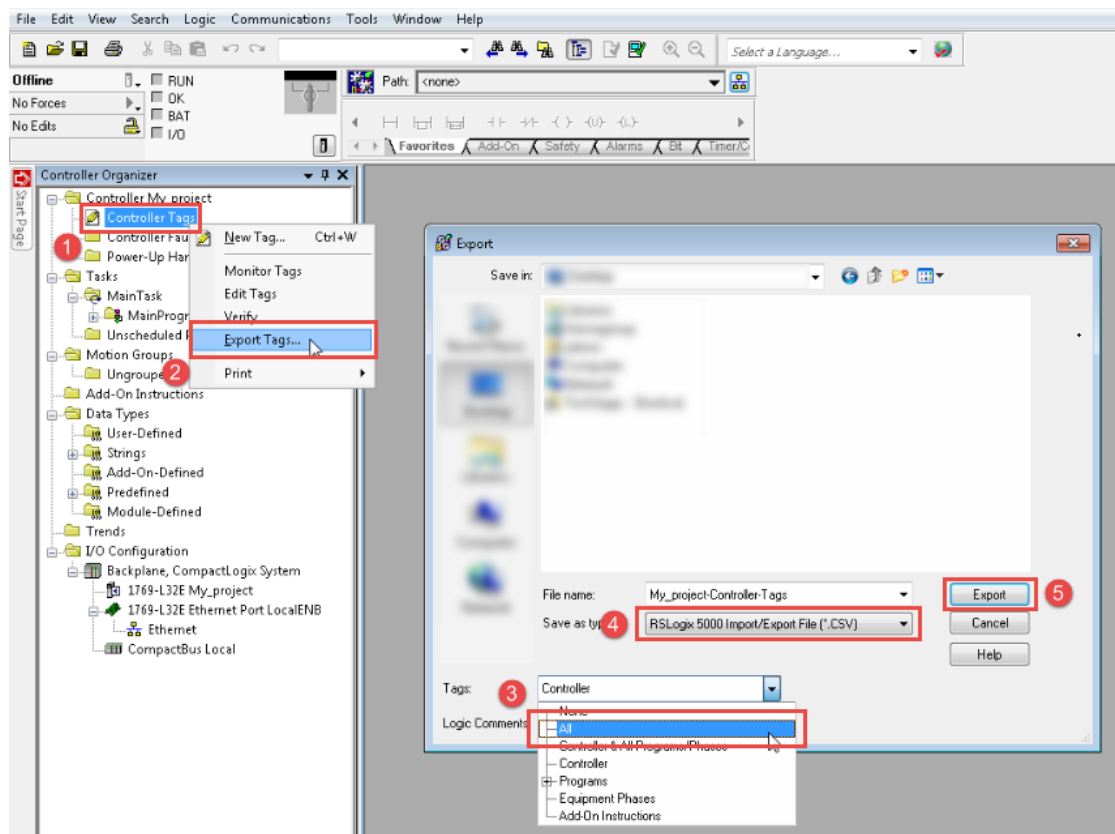
The implementation of the Ethernet/IP driver also supports access to structured data types which can be imported from .L5X files.

The driver supports access to both Controller and Program Tags.

## Export CSV and L5X files using RSLogix5000

To export the .CSV Tag file:

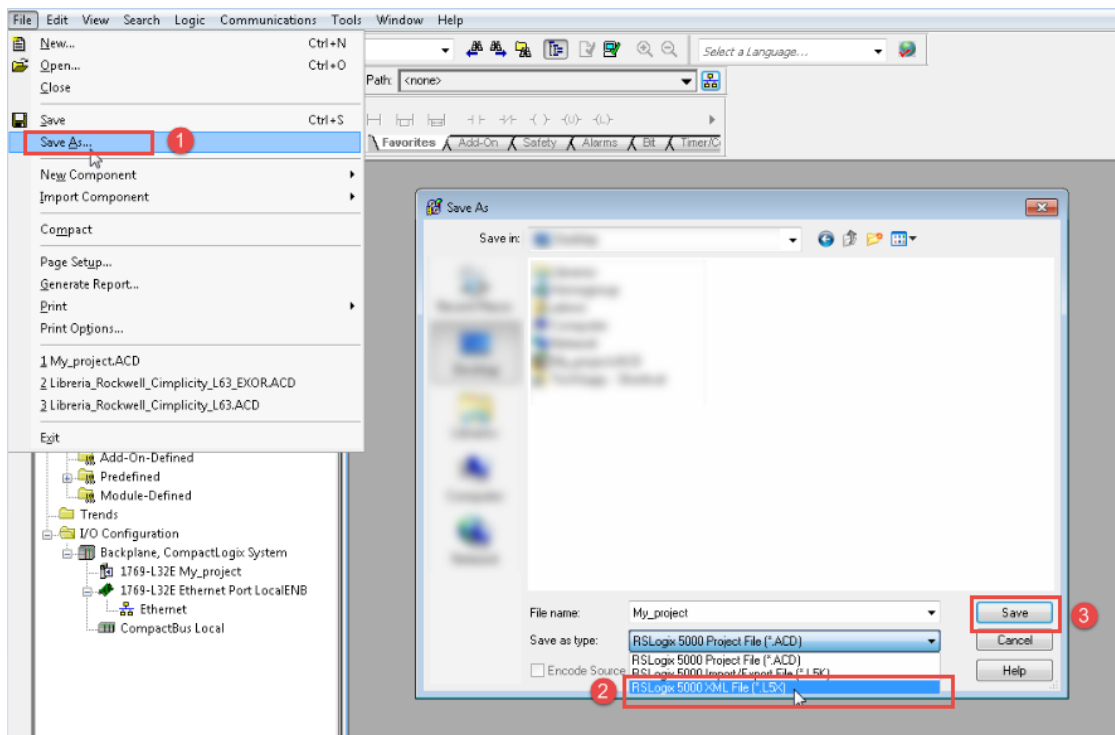
1. From the **Controller Organizer** pane, right-click on **Controller Tags**.
2. Select **Export Tags**: the **Export** dialog is displayed.



3. Choose **All** from the **Tags** list to export all Tags.
4. Select the **Save as type** option to **.CSV**.
5. Click **Export**: all the Tags are exported to an **.CSV** file.

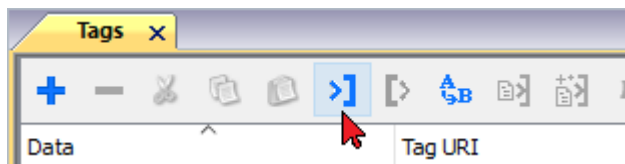
To export the **.L5X** data type file:

1. Choose **File > Save As**.
2. Select the **Save as type** option to **.L5X**.
3. Click **Save**: all the Tags are exported to an **.L5X** file.

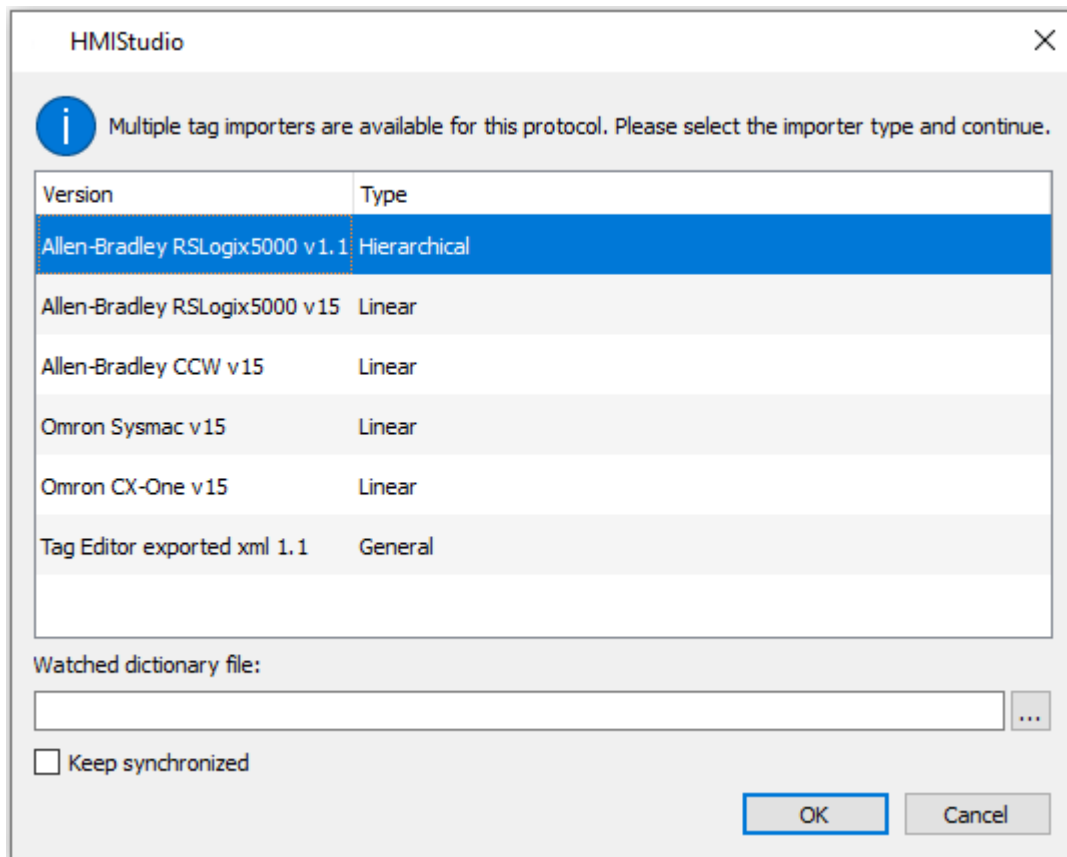


## Import Files in Tag Editor

Select the driver in Tag Editor and click on the **Import Tags** button to start the importer.



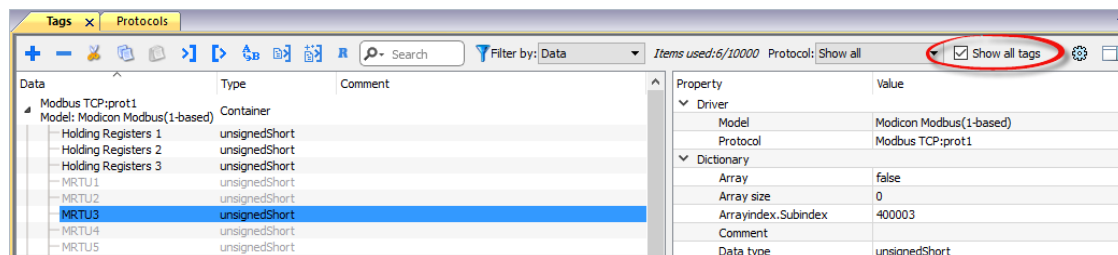
The following dialog shows which importer type can be selected.





Select **Allen-Bradley RSLogix5000 v15** option.

Once the importer has been selected, locate the symbol file and click **Open**.

The tags available within the Dictionary but not imported into the project are gray and are visible only when the "Show all tags" check box is selected.

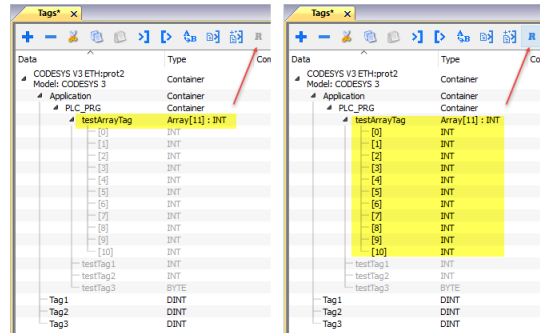


Toolbar item	Description
	<p><b>Import Tag(s).</b></p> <p>Select tags to be imported and click on this icon to add tags from tag dictionary to the project</p>
	<p><b>Update Tag(s).</b></p> <p>Click on this icon to update the tags in the project, due a new dictionary import.</p>

Toolbar item	Description
--------------	-------------



Check this box to import all sub-elements of a tag.  
Example of both checked and unchecked result:



Searches tags in the dictionary basing on filter combobox item selected.



Note: When importing the array data types, the importer is expanding them creating individual Tags per each array element; this is valid for all the data types, except for arrays of boolean. In this case they are imported as “boolean-32” and the single array element can be addressed using “Tag Index” parameter from “Attach to...” dialog.

## Module-Defined and User-Defined data types

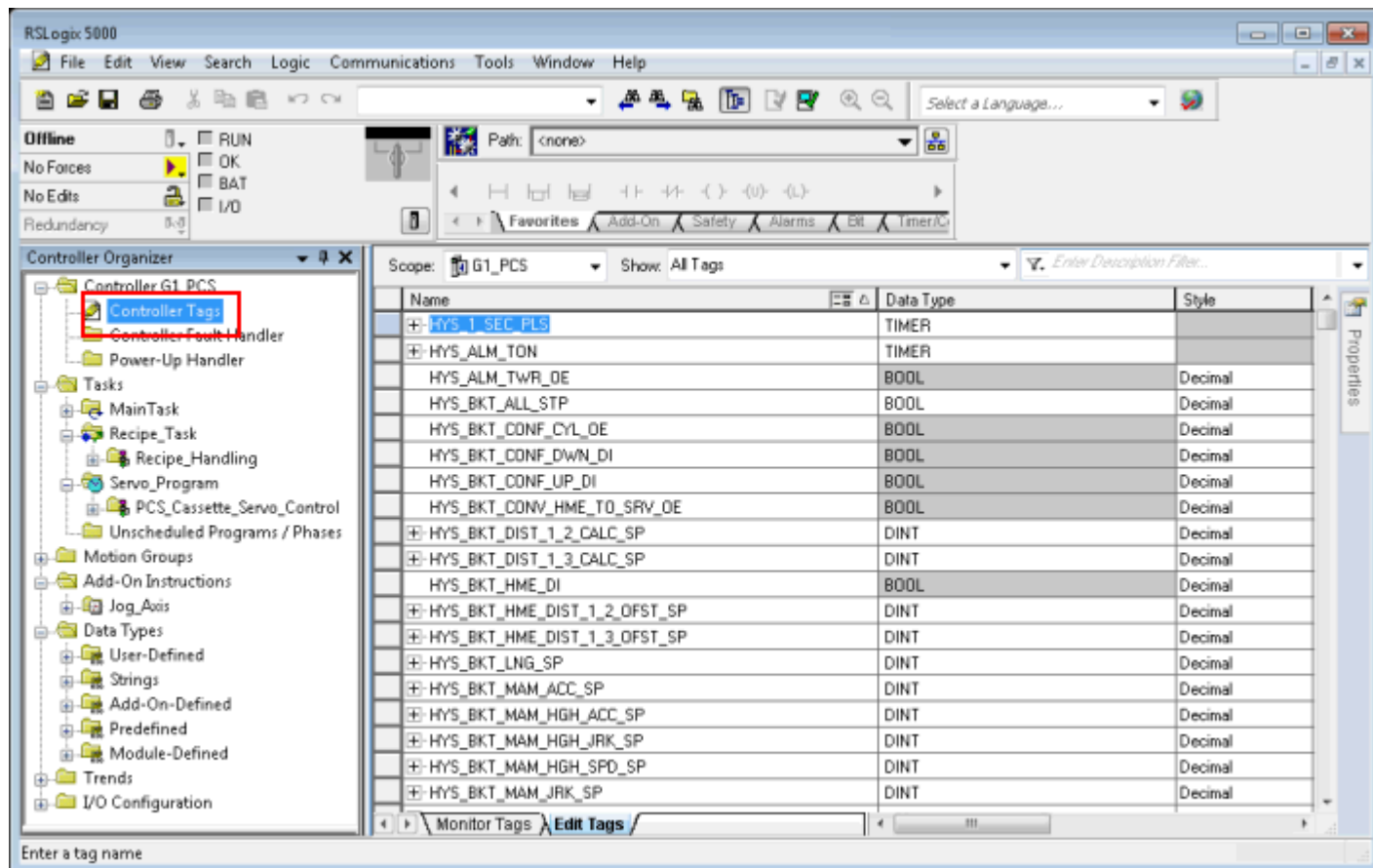
RSLogix 5000 allows you to define Tags with several data types.

Data type group	Description
Predefined	Standard data types such as BOOL, DINT, SINT, INT and other less common data types such as PID, COUNTER, TIMER.
Module-Defined	Data type associated with I/O optional modules usually referenced by aliases.
User-Defined	Custom data type defined by user

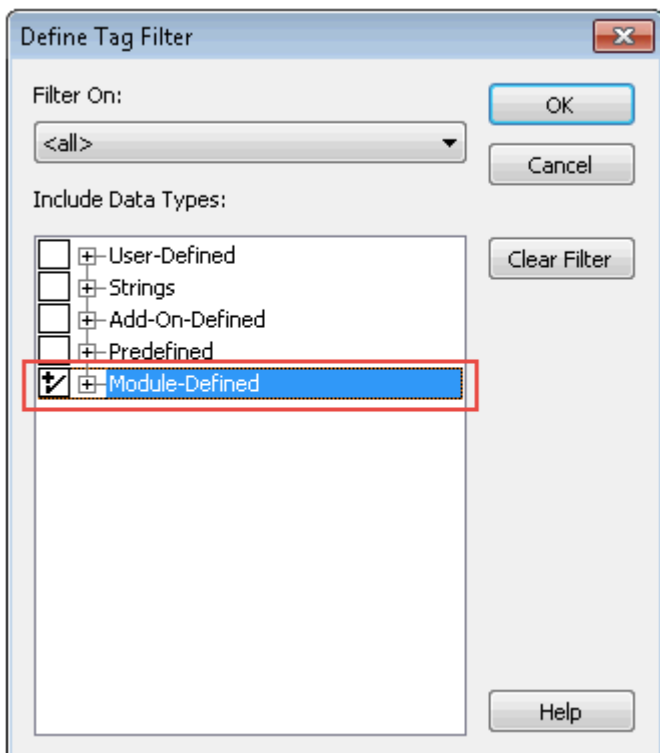
In order to import Predefined (with the exception of standard data types which are always imported) and Module-Defined data type you need to edit the ETIPSpecialDataTypes.xml file located under *languages\shared\studio\tagimport* or *studio\tagimport* depending on installed version.

In RSLogix5000 software:

1. From the **Controller Organizer** pane, select **Controller Tags**.



2. Filter tags to display only **Module-Defined** Tags.



Only tags (alias) with data type belonging to optional I/O Modules will be displayed.

Name	Data Type	Style
+ HYS_Point_IO_Rack_20:I	AB:1734_3SLOT:I:0	
+ HYS_Point_IO_Rack_20:O	AB:1734_3SLOT:O:0	
+ HYS_Point_IO_Rack_1:I	AB:1734_13SLOT:I:0	
+ HYS_Point_IO_Rack_1:O	AB:1734_13SLOT:O:0	
+ HYS_Point_IO_Rack_1:2:C	AB:1734_DI8:C:0	
+ HYS_Point_IO_Rack_1:3:C	AB:1734_DI8:C:0	
+ HYS_Point_IO_Rack_1:4:C	AB:1734_DI8:C:0	
+ HYS_Point_IO_Rack_1:5:C	AB:1734_DI8:C:0	
+ HYS_Point_IO_Rack_1:6:C	AB:1734_DI8:C:0	
+ HYS_Point_IO_Rack_1:7:C	AB:1734_DI8:C:0	
+ HYS_Point_IO_Rack_1:8:C	AB:1734_DI8:C:0	
+ HYS_Point_IO_Rack_20:1:C	AB:1734_DI8:C:0	
+ HYS_Point_IO_Rack_1:9:C	AB:1734_DO8_NoDiag:C:0	
+ HYS_Point_IO_Rack_1:10:C	AB:1734_DO8_NoDiag:C:0	
+ HYS_Point_IO_Rack_1:11:C	AB:1734_DO8_NoDiag:C:0	
+ HYS_Point_IO_Rack_1:12:C	AB:1734_DO8_NoDiag:C:0	
+ HYS_Point_IO_Rack_20:2:C	AB:1734_DO8_NoDiag:C:0	
+ HYS_Point_IO_Rack_1:1:C	AB:1734_VHSC:C:0	
+ HYS_Point_IO_Rack_1:1:I	AB:1734_VHSC:I:0	

In this example alias HYS\_Point\_IO\_Rack\_20:I refers to data type AB:1734\_3SLOT:I:0. Expand this tag to see how this data type is structured:

Name	Data Type	Style
- HYS_Point_IO_Rack_20:I	AB:1734_3SLOT:I:0	
+ HYS_Point_IO_Rack_20:I.SlotStatusBits0_31	DINT	Binary
+ HYS_Point_IO_Rack_20:I.SlotStatusBits32_63	DINT	Binary
+ HYS_Point_IO_Rack_20:I.Data	SINT[3]	Binary

To make sure that HYS\_Point\_IO\_Rack\_20:I, and all his sub-tags, will be imported into the project, open the ETIPSpecialDataTypes.xml file in any text editor and check if the AB:1734\_3SLOT:I:0 data type is included. If so you can proceed with the following data type. If not, you need to add it manually.

The structure is as in this example:

```

<DataType Name="aaa">
  <Members>
    <Member Name="bbb" DataType="ccc" Dimension="ddd" Radix="eee"/>
  </Members>
</DataType>

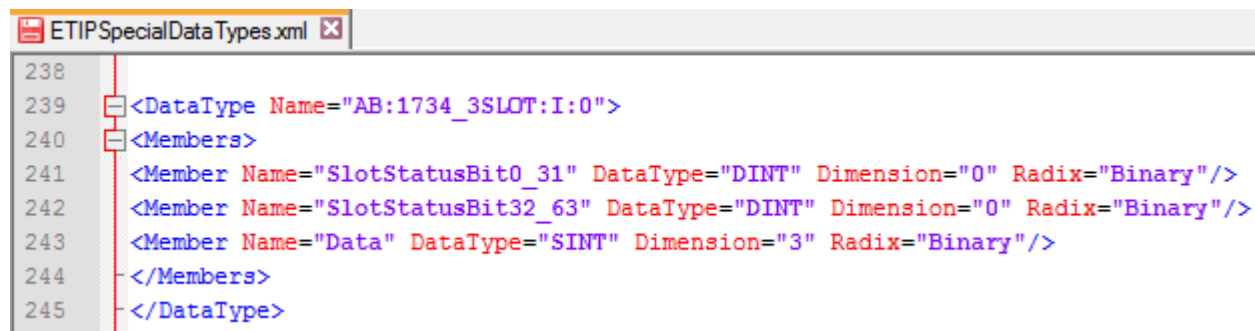
```



where:

- aaa = Alias/Tag data type
- bbb = Sub-tag Name (it's sub-tag name part after dot)
- ccc = Sub-tag data type
- ddd = Array dimension (0 if it is not an array)
- eee = Style

In the example above:



```

238
239 <DataType Name="AB:1734_3SLOT:I:0">
240 <Members>
241 <Member Name="SlotStatusBit0_31" DataType="DINT" Dimension="0" Radix="Binary"/>
242 <Member Name="SlotStatusBit32_63" DataType="DINT" Dimension="0" Radix="Binary"/>
243 <Member Name="Data" DataType="SINT" Dimension="3" Radix="Binary"/>
244 </Members>
245 </DataType>

```

3. Repeat step 2 for all Module-Defined data types.
4. Repeat the procedure from step 2, filtering Tags to display only **Predefined** Tags.

## Controller Model Omron Sysmac

Data in NJ and CJ controllers can be accessed via CIP protocol.

Each data item can be identified by a string called "Tag". Use appropriate programming tools for controller to export the list of Tags.

NJ series controller are programmed using Sysmac Studio:

- NJ301-xxxx
- NJ501-xxxx

CJ series controller are programmed using CX-One:

- CJ2M CPU-3x
- CJ2H CPU 6x-EIP
- Any CPU with a CJ1W-EIP21 attached.

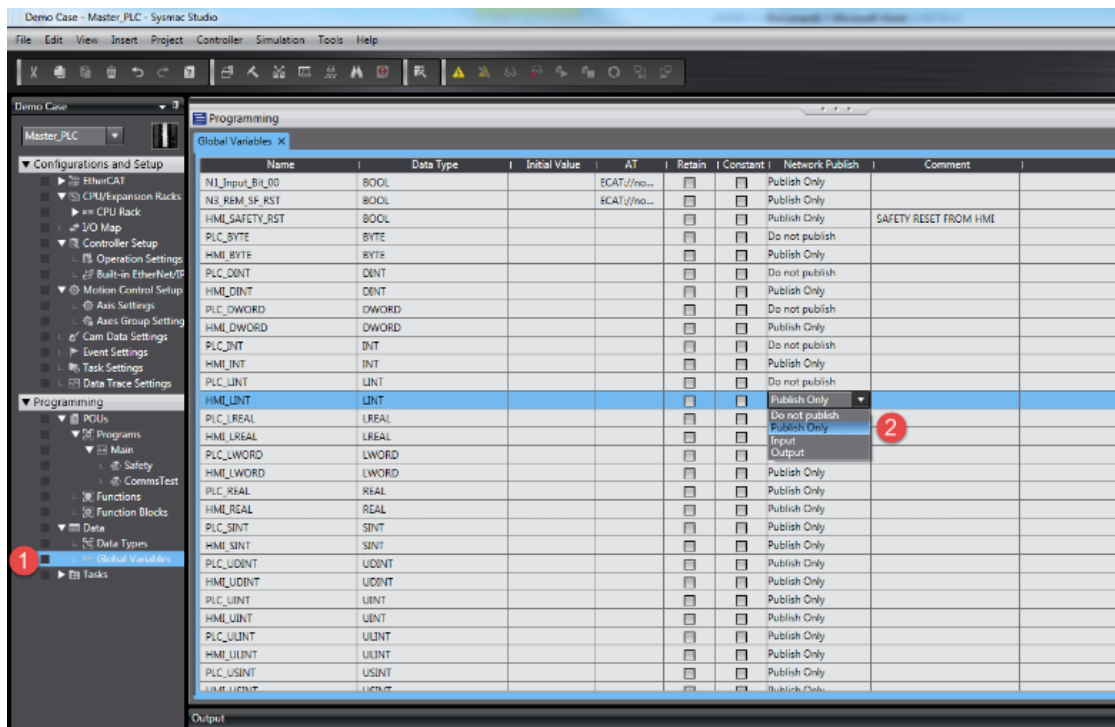
The project loaded on the HMI device must refer to the Tag names assigned in the programming software at development time. The Tag Editor supports direct import of the Tag file generated by Sysmac Studio software in .NJF format or generated by CX-One in the .CJF format.

All Tags to be accessed by the HMI device must be declared as Global Variables.

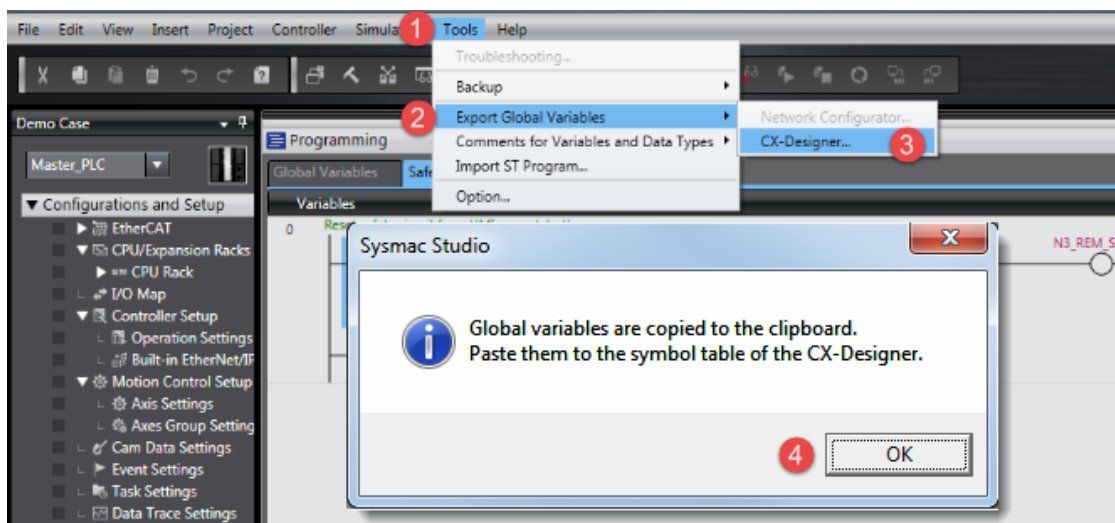
### Export NJF files using Sysmac Studio

To export the .NJF Tag file:

1. In Sysmac Studio declare Tags as **Global Variables**.
2. Set the **Network Publish** attribute to **Publish Only**.

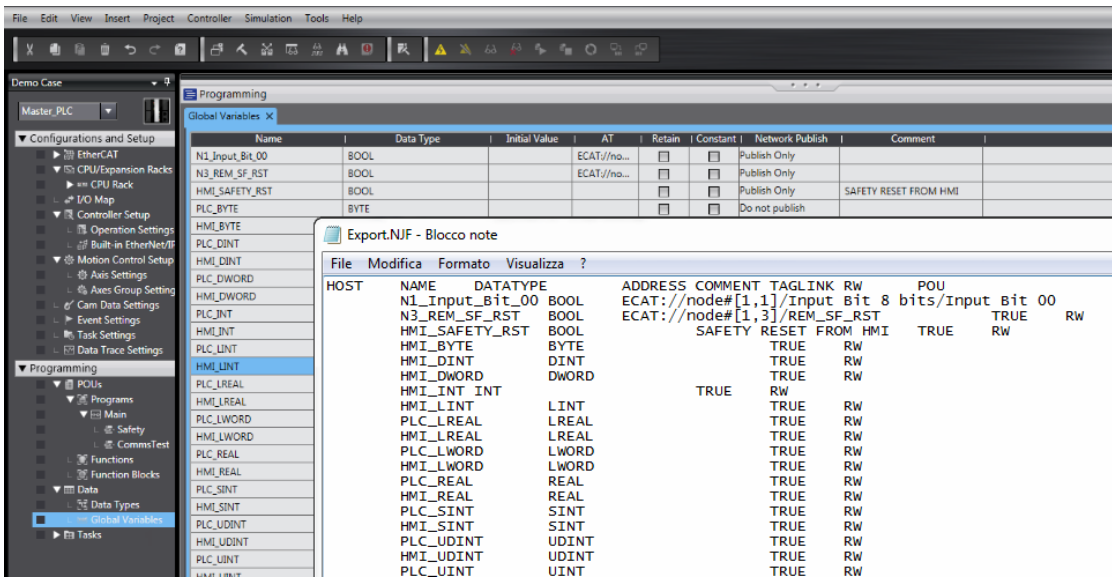


2. From the **Tools** menu, choose **Export Global Variables > CX-Designer**.




3. Click **OK** to confirm.

4. Cut and paste the content of the clipboard in any text editor.



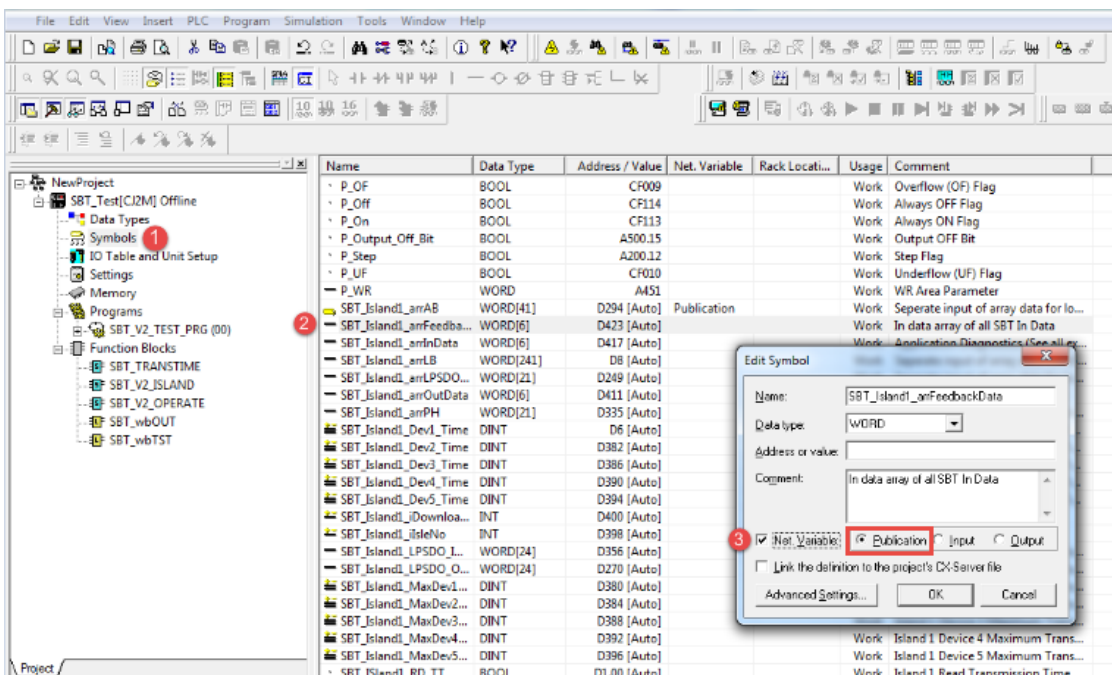
4. Save the file as .NJF.

 Note: Using Notepad as text editor, make sure to save the text file with .NJF extension by selecting "Save as type" as "All Files" although the file will be named \*.njf.txt and it will not be visible from importer.

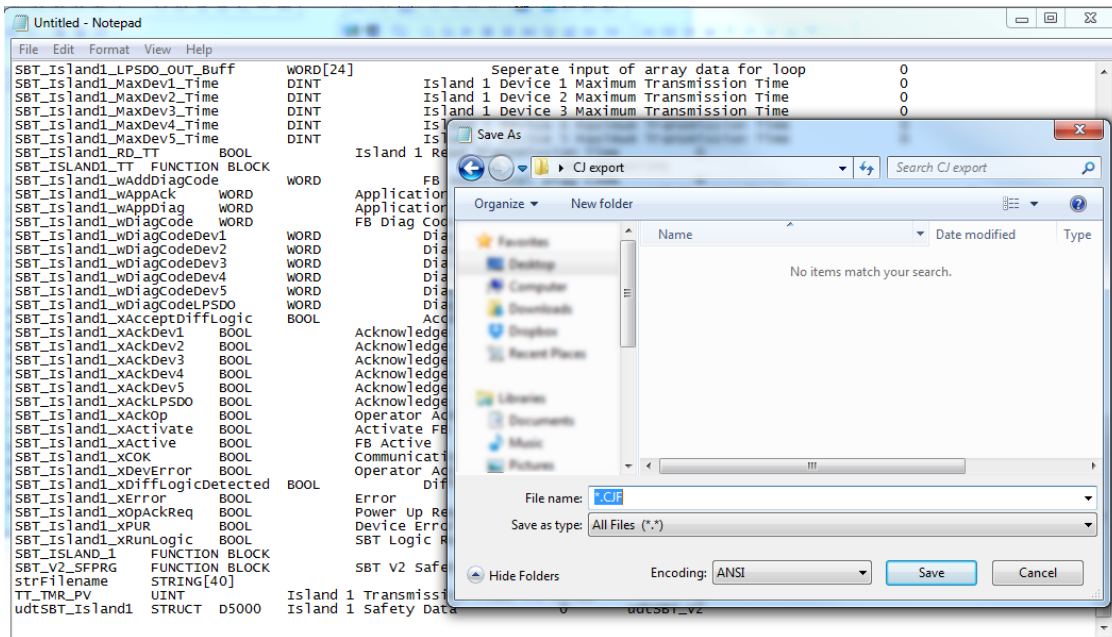
### Export CJF file using CX-One

To export the .CJF Tag file:

1. In CX-One open the Symbols file in the project.
2. In the **Edit Symbol** dialog set the **Net. Variables** attribute to **Publication**.



3. Copy and paste all the Tags in any text editor.



4. Save the file as **.CJF**.

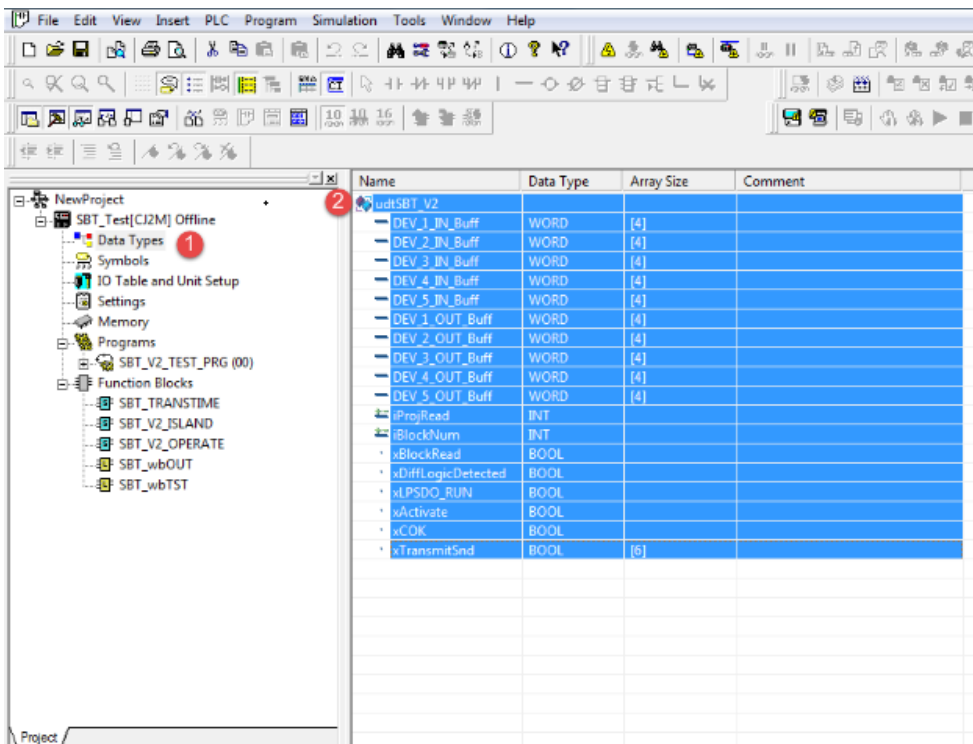


Note: Using Notepad as text editor, make sure to save the text file with **.CJF** extension by selecting "Save as type" as "All Files" although the file will be named \*.cjf.txt and it will not be visible from importer.

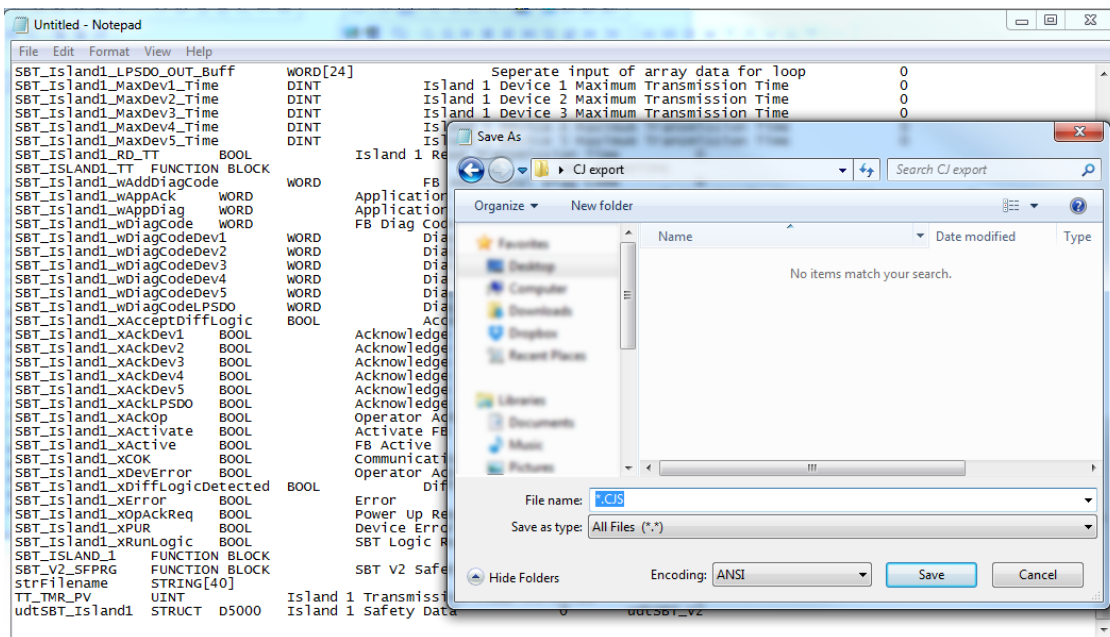
## Export User Defined structures

To export the **.CJS** Tag file:


1. In CX-One open the Data Types file in the project.



2. Copy and paste all the Tags in any text editor.

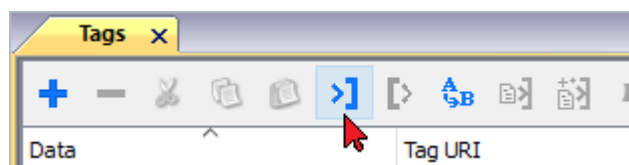


3. Save the file as .CJS.

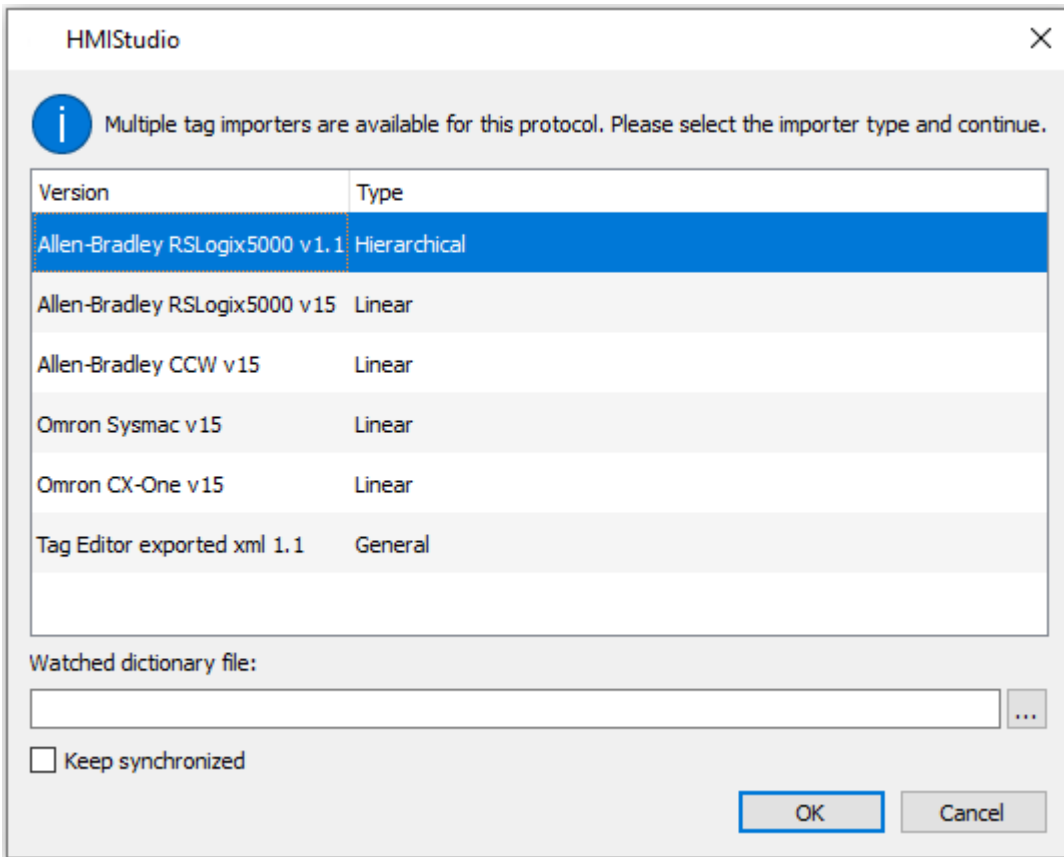
 Note: Using Notepad as text editor, make sure to save the text file with **.CJS** extension by selecting "Save as type" as "All Files" although the file will be named \*.cjs.txt and it will not be visible from importer.

### Import Files in Tag Editor

Select the driver in Tag Editor and click on the **Import Tags** button to start the importer.



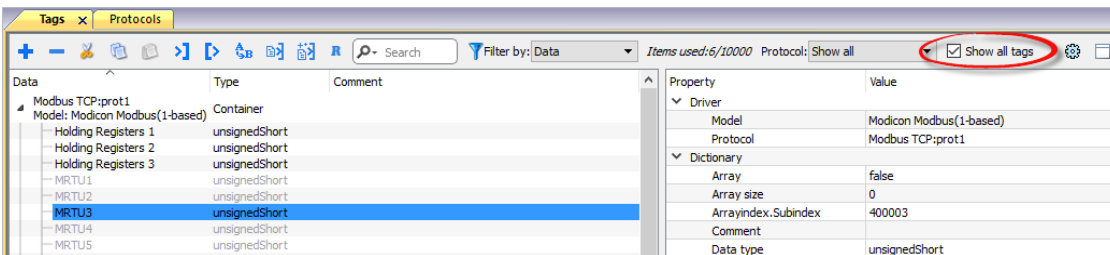
The following dialog shows which importer type can be selected.




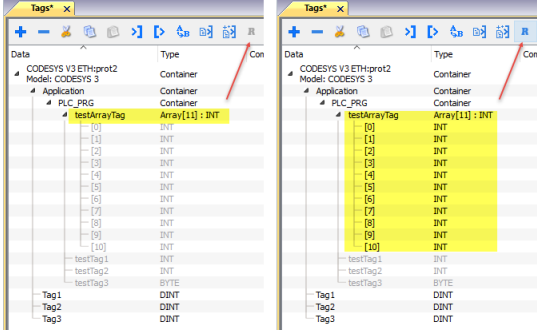
Select **Omron Sysmac** to import a **.NJF** Tags file or **Omron CX-One** to import a **.CJF** Tags file.

Once the importer has been selected, locate the Tags file and click **Open**. The system will ask for User Defined structures **.CJS** file. If not required, skip the dialog by clicking on Cancel button.

Tags included in the symbol file are listed in the tag dictionary. The tag dictionary is displayed at the bottom of the screen.




Toolbar item	Description
	<b>Import Tag(s).</b> Select tags to be imported and click on this icon to add tags from tag dictionary to the project
	<b>Update Tag(s).</b> Click on this icon to update the tags in the project, due a

Toolbar item	Description
	<p>new dictionary import.</p> <p>Check this box to import all sub-elements of a tag. Example of both checked and unchecked result:</p> 



Searches tags in the dictionary basing on filter combobox item selected.

 Note: When importing the array data types, the importer is expanding them creating individual Tags per each array element; this is valid for all the data types, except for arrays of boolean. In this case they are imported as “boolean-32” and the single array element can be addressed using “Tag Index” parameter from “Attach to...” dialog.

## Controller Model Micro800

The Ethernet/IP CIP driver provides an easy and reliable way to connect to Allen-Bradley Micro800 controllers.

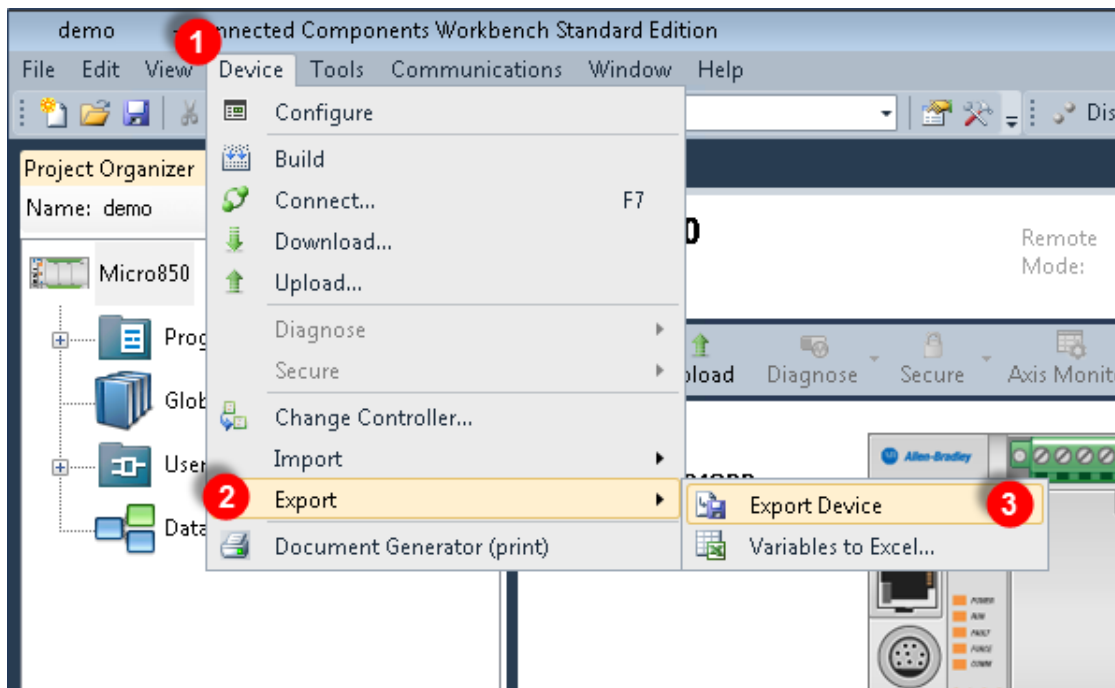
The scope of variables into a Micro800 controller can be local to a program or global:

Scope	Description
<p><b>Local Variables</b></p>	<p>Program-scoped Tags. Tags are assigned to a specific program in the project and available only to that program.</p> <p>These Tags are <b>not supported</b> within this driver.</p>
<p><b>Global Variables</b></p>	<p>Controller-scoped Tags. Tags belong to the controller in the project and are available to any program in the project.</p> <p>These Tags are <b>supported</b> within this driver.</p>

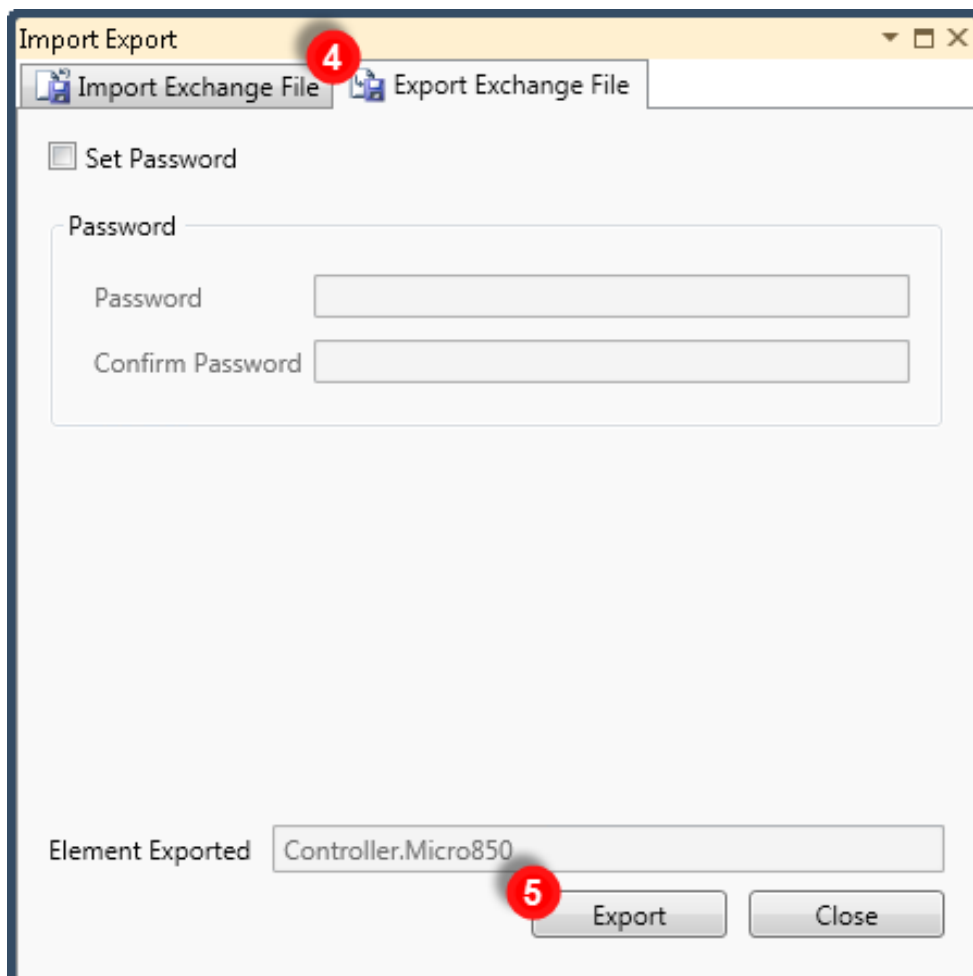
## Export ISAXML file using Connected Component Workbench

To export .ISAXML global variables including I/O tags:

1. Select **Device** tab.
2. Expand **Export** item.
3. Select **Export Device**.

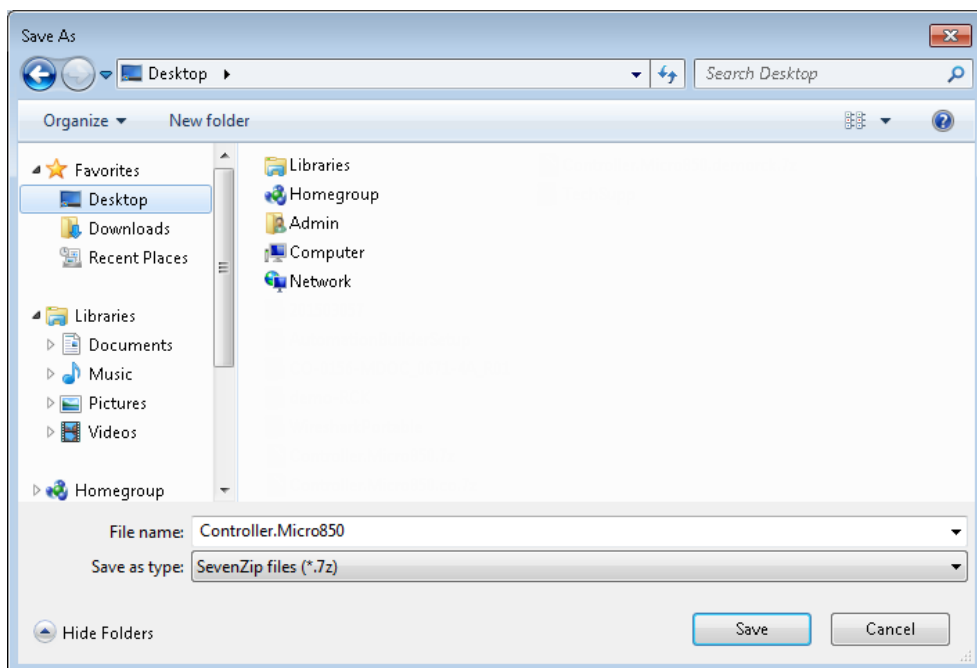


4. Click on **Export Exchange File** tab.
5. Click **Export** button.

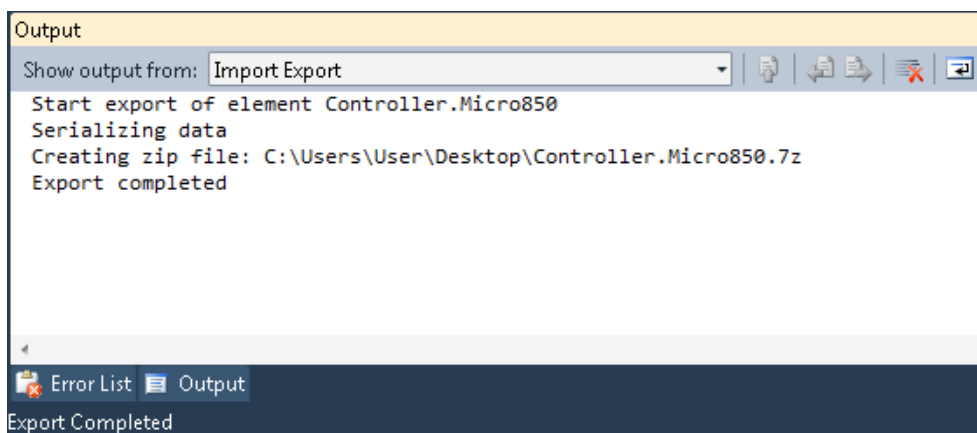





6. Choose a location where to save the export file and click **Save**.



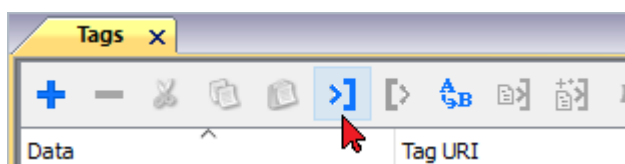
7. When the export is completed successfully the output information is displayed:



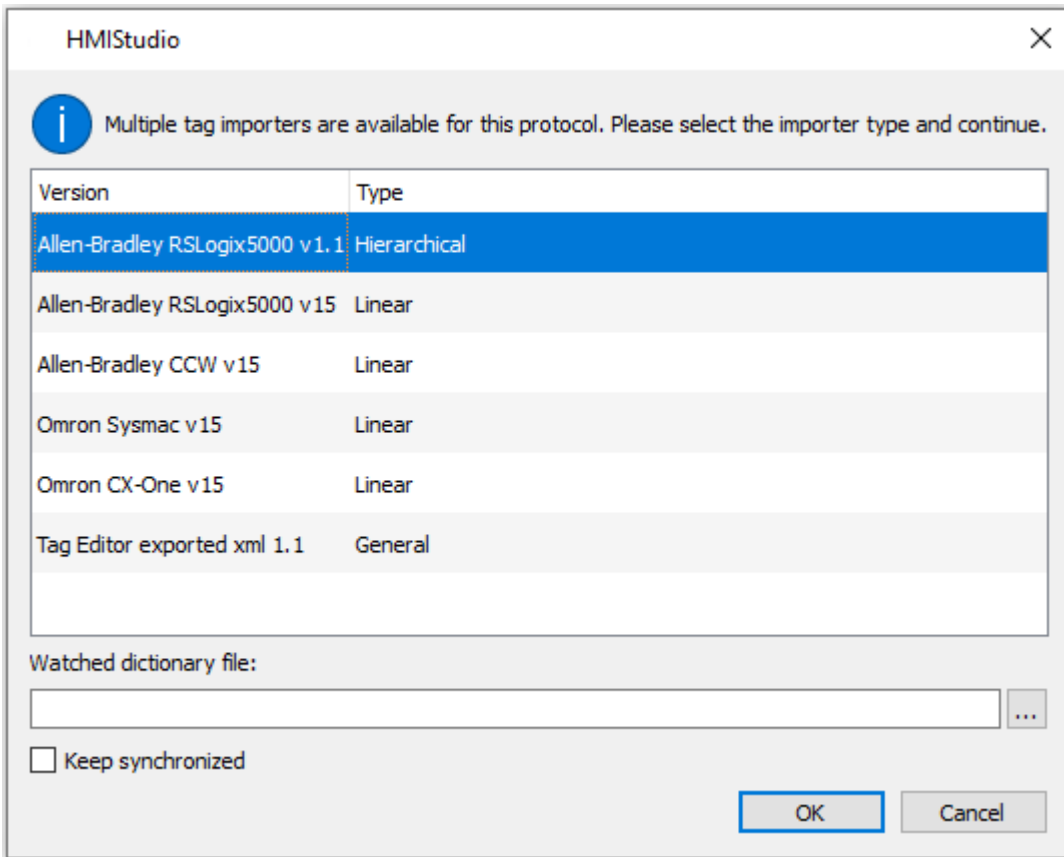
 Note: CCW export file is a 7-zip compressed archive. Use a suitable zip utility to extract archive content into a local folder.

## Import Files in Tag Editor

Select the driver in Tag Editor and click on the **Import Tags** button to start the importer.



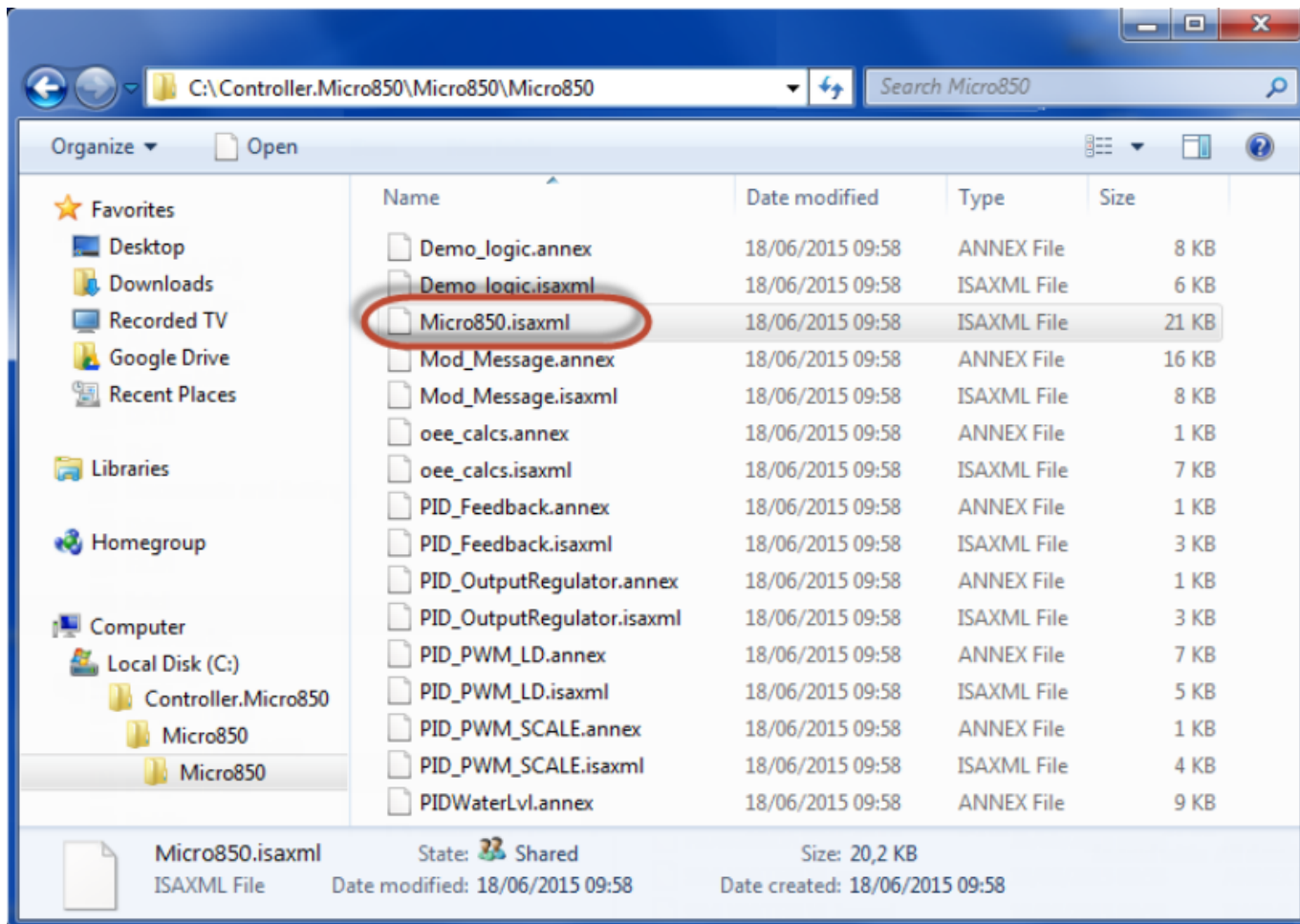
The following dialog shows which importer type can be selected.



Select **Allen-Bradely CCW v15** option.

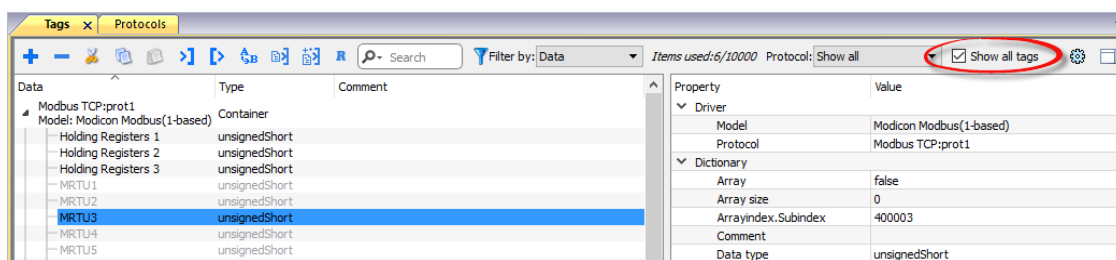
Directory structure extracted from 7z file is something like: “..\<folder\_name>\Micro8xx\Micro8xx\”



Inside this last folder, select the Micro8xx.isaxml file as shown below:


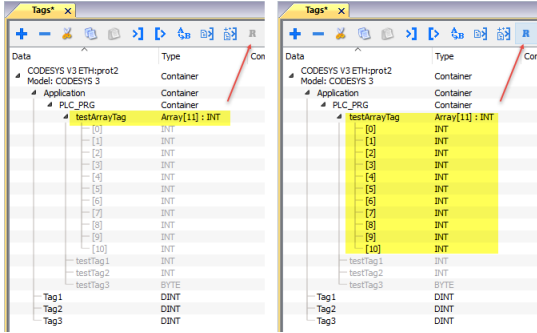
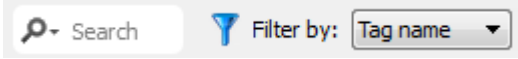


Once the importer has been selected, locate the symbol file and click **Open**.

The tags available within the Dictionary but not imported into the project are gray and are visible only when the "Show all tags" check box is selected.



Toolbar item	Description
	<p><b>Import Tag(s).</b></p> <p>Select tags to be imported and click on this icon to add tags from tag dictionary to the project</p>
	<p><b>Update Tag(s).</b></p>

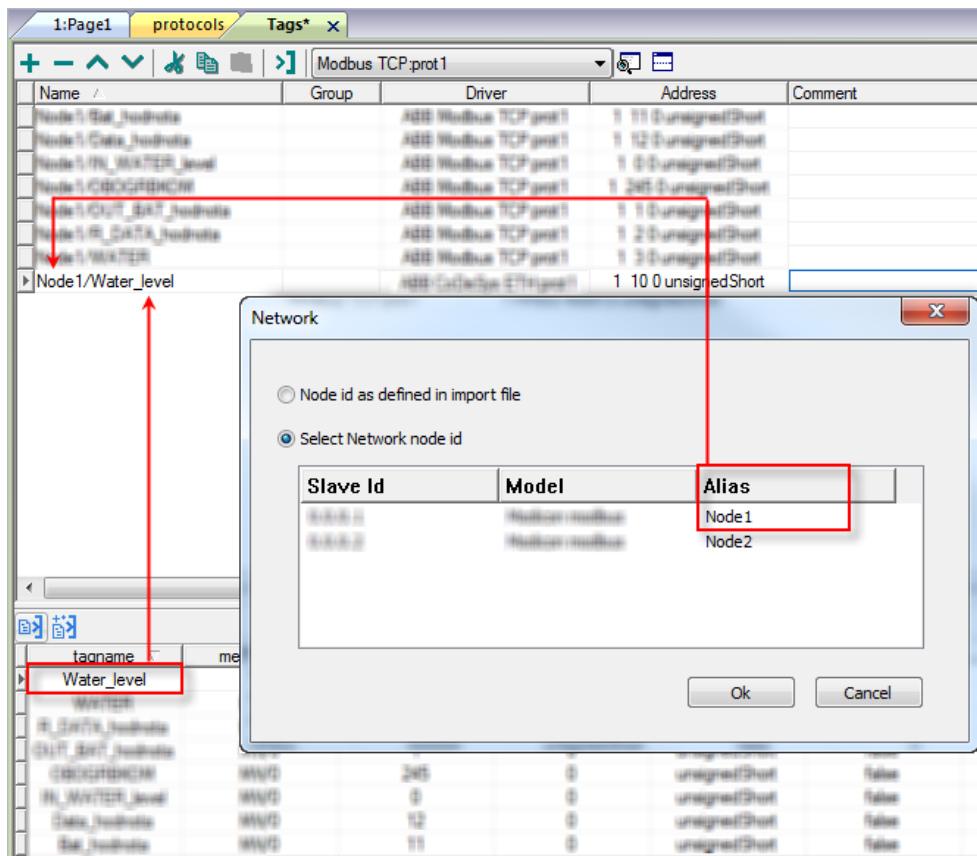
Toolbar item	Description
	Click on this icon to update the tags in the project, due a new dictionary import.
	<p>Check this box to import all sub-elements of a tag.</p> <p>Example of both checked and unchecked result:</p> 
	Searches tags in the dictionary basing on filter combobox item selected.

## Adding an alias name to a protocol

Tag names must be unique at project level, however, the same tag names might need to be used for different controller nodes (for example when the HMI device is connected to two devices running the same application).

When creating a protocol you can add an alias name that will be added to tag names imported for this protocol.

In the example, the connection to a certain controller is assigned the name **Node1**. When tags are imported for this node, all tag names will have the prefix **Node1** making each of them unique at the network/project level.



**i** Note: Aliasing tag names is only available for imported tags. Tags which are added manually in the Tag Editor do not need to have the Alias prefix in the tag name. The Alias string is attached on the import. If you modify the Alias string after the tag import has been completed, there will be no effect on the names already present in the dictionary. When the Alias string is changed and tags are re-imported, all tags will be re-imported with the new prefix string.

## Node Override IP

The protocol provides the special data type Node Override IP which allows you to change the IP address of the target controller at runtime.

This memory type is an array of 4 unsigned bytes, one per each byte of the IP address.

The Node Override IP is initialized with the value of the controller IP specified in the project at programming time.

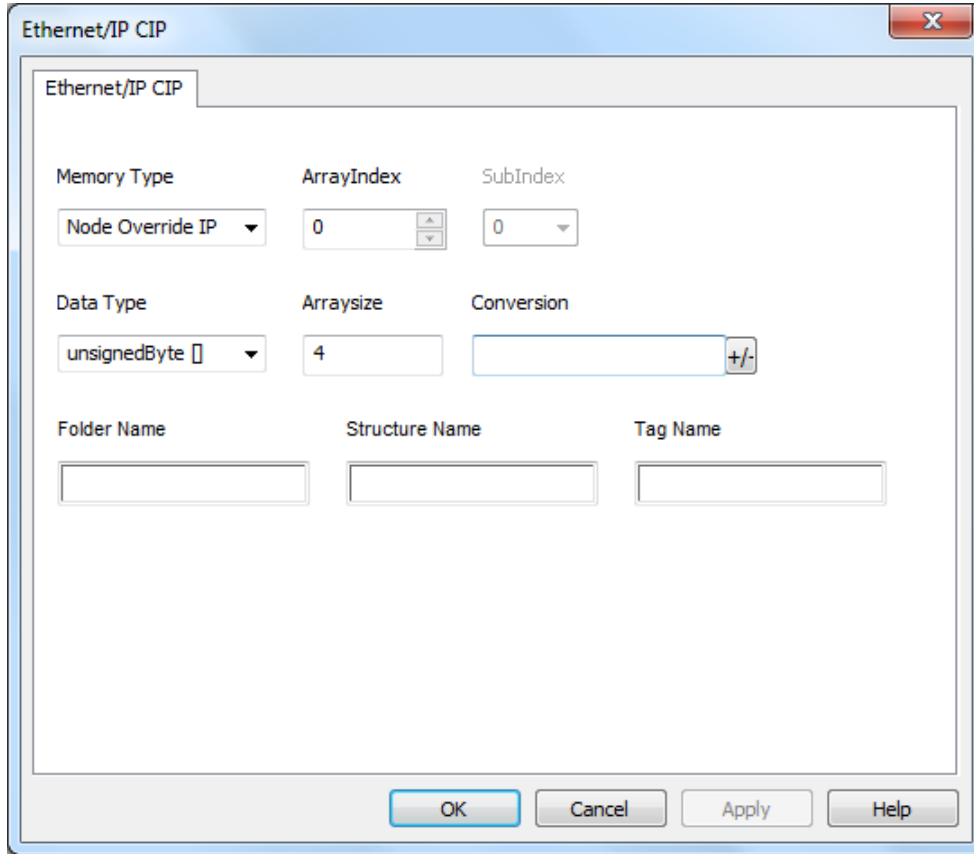
Node Override IP	PLC operation
0.0.0.0	Communication with the controller is stopped, no request frames are generated anymore.
<b>Different from 0.0.0.0</b>	It is interpreted as node IP override and the target IP address is replaced runtime with the new value.

If the HMI device is connected to a network with more than one controller node, each node has its own Node Override IP variable.

**i** Note: Node Override IP values assigned at runtime are retained through power cycles.

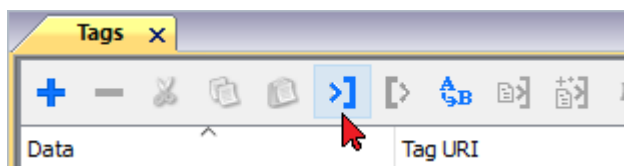
## Hostname DNS or mDNS

In addition to the array of bytes, string memory type can be selected to be able use the DNS or mDNS hostname as an alternative to the IP Address.

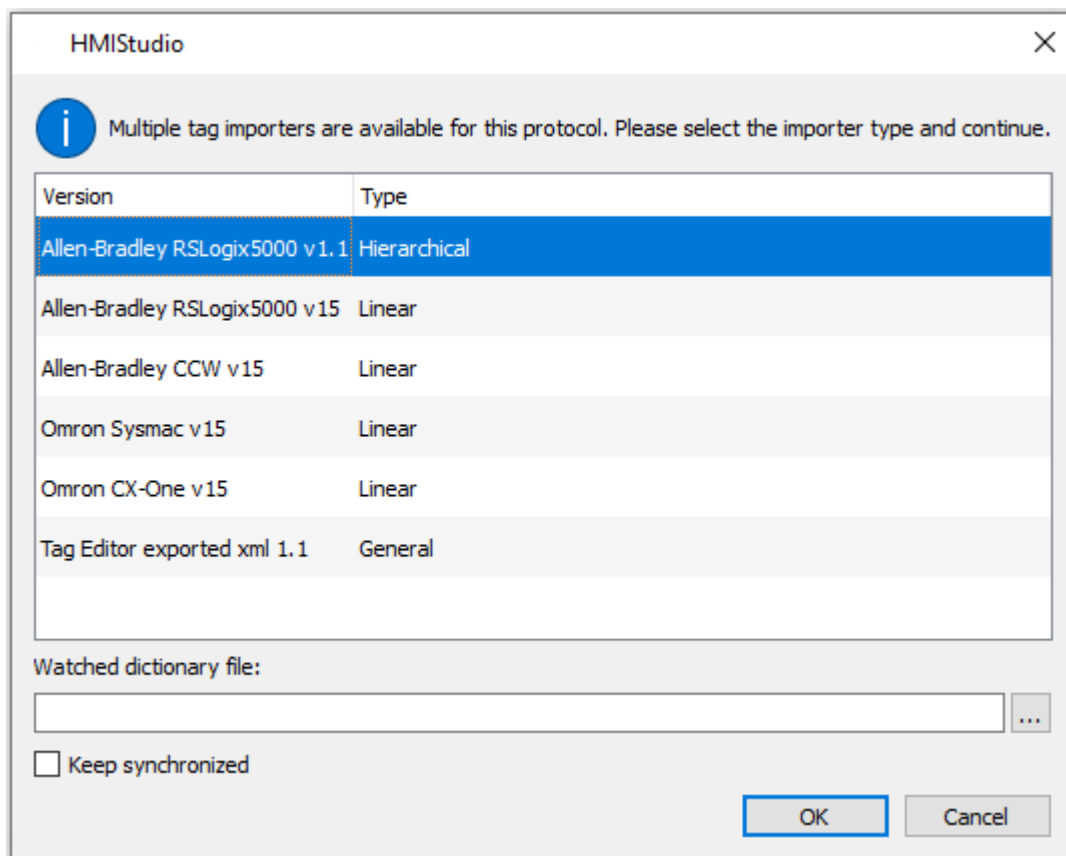


## Tag Import

Select the driver in Tag Editor and click on the **Import Tags** button to start the importer.



The following dialog shows which importer type can be selected.

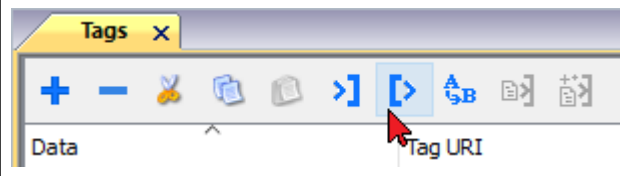


Importer	Description
<b>Allen-Bradley L5X v1.1 Hierarchical</b>	Requires a <b>.L5X</b> file. Check <b>Controller Model Logix 5000</b> for more details. All variables will be displayed according to RSLogix5000 Hierarchical view.
<b>Allen-Bradley RSLogix5000 v15 Linear</b>	Requires a <b>.CSV</b> and <b>.L5X</b> (optional) files. Check <b>Controller Model Logix 5000</b> for more details. All variables will be displayed at the same level.
<b>Allen-Bradley CCW v15 Linear</b>	Requires a <b>.ISAXML</b> file. Check <b>Controller Model Micro800</b> for more details. All variables will be displayed at the same level.
<b>Omron Sysmac v15 Linear</b>	Requires a <b>.NJF</b> file. Check <b>Controller Model Omron Sysmac</b> for more details. All variables will be displayed at the same level.
<b>Omron CX-One v15 Linear</b>	Requires a <b>.CJF</b> and <b>.CJS</b> (optional) files. Check <b>Controller Model Omron Sysmac</b> for more details.

Importer	Description
----------	-------------

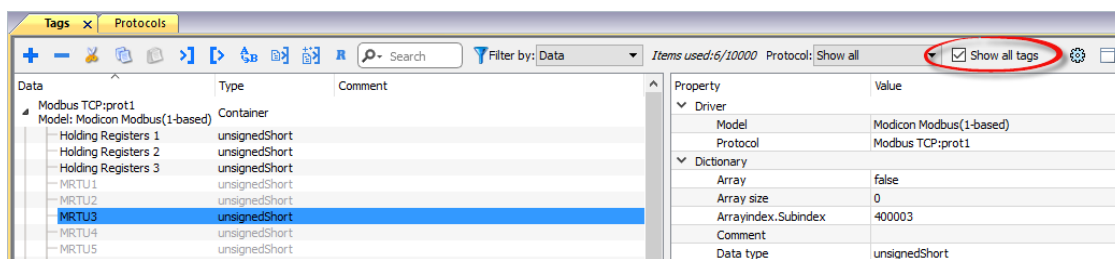
All variables will be displayed at the same level.

**Tag Editor exported xml** Select this importer to read a generic XML file exported from Tag Editor by appropriate button.




Once the importer has been selected, locate the symbol file and click **Open**.

The tags available within the Dictionary but not imported into the project are gray and are visible only when the "Show all tags" check box is selected.


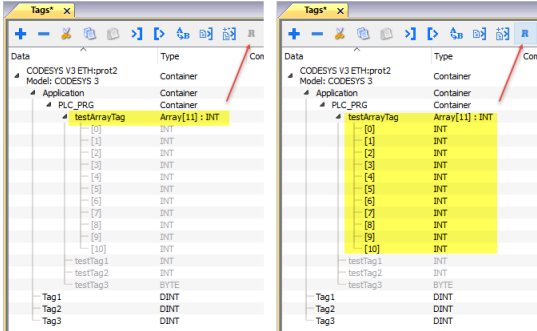
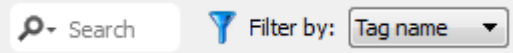


Toolbar item	Description
--------------	-------------

	<p><b>Import Tag(s).</b></p> <p>Select tags to be imported and click on this icon to add tags from tag dictionary to the project</p>
---	--

	<p><b>Update Tag(s).</b></p> <p>Click on this icon to update the tags in the project, due a new dictionary import.</p>
---	--



Toolbar item	Description
	<p>Check this box to import all sub-elements of a tag.</p> <p>Example of both checked and unchecked result:</p> 
	<p>Searches tags in the dictionary basing on filter combo-box item selected.</p>

## Communication status

Current communication status can be displayed using System Variables. See "System Variables" section in the main manual.

Codes supported for this communication driver:

Error	Cause	Action
<b>NAK</b>	The controller replies with a not acknowledge.	-
<b>Timeout</b>	A request is not replied within the specified timeout period.	Check if the controller is connected and properly configured to get network access.
<b>Invalid response</b>	The device did received a response with invalid format or contents from the controller .	Ensure the data programmed in the project are consistent with the controller resources.
<b>General Error</b>	Unidentifiable error. Should never be reported.	Contact technical support.

---

# Modbus RTU

The operator panels can be connected to a Modbus network as the network master using this communication driver.

## Implementation details

The Modbus RTU implementation supports only a subset of the Modbus standard RTU function codes.

Code	Function	Description
01	Read Coil Status	Reads multiple bits in the device Coil area
02	Read Input Status	Read the ON/OFF status of the discrete inputs (1x reference) in the slave
03	Read Holding Registers	Read multiple Registers
04	Read Input Registers	Reads the binary contents of input registers (3x reference) in the slave
05	Force Single Coil	Forces a single Coil to either ON or OFF
06	Preset Single Register	Presets a value in a Register
16	Preset Multiple Registers	Presets value in multiple Registers



Note: Communication speed with controllers is supported up to 115200 baud.



Note: Floating point data format is IEEE standard compliant.

## Protocol Editor Settings

### Adding a protocol

To configure the protocol:

1. In **Config** node double-click **Protocols**.
2. To add a driver, click **+**: a new line is added.
3. Select the protocol from the **PLC** list.

The protocol configuration dialog is displayed.

Modbus RTU

PLC Network

Comm... OK Cancel

Alias

Node ID

Timeout (ms)

Delay (ms)

Num of repeats

Max read block

Max read bit block

Write Holding Register



Write Coils

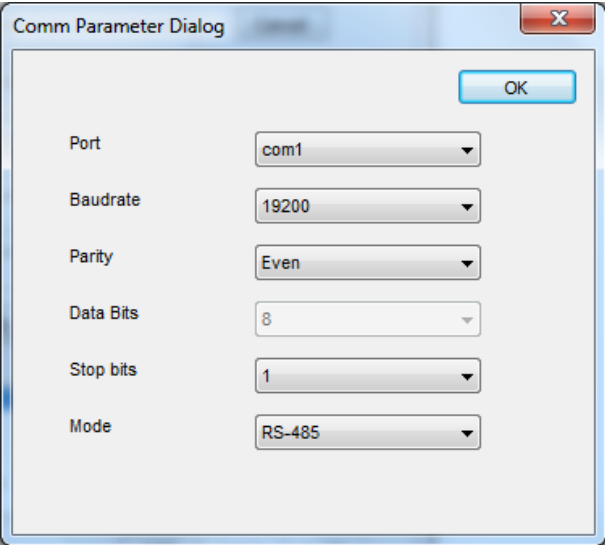
Transmission Mode

PLC Models

- Modicon Modbus(1-based)
- Generic Modbus(0-based)
- Enron Modbus(1-based) with 32bit registers
- Enron Modbus(0-based) with 32bit registers

Element	Description
<b>Alias</b>	Name identifying nodes in network configurations. The name will be added as a prefix to each tag name imported for each network node.
<b>Node ID</b>	Modbus node of the slave device.
<b>Timeout (ms)</b>	Time delay in milliseconds between two retries in case of missing response from the server device.
<b>Delay (ms)</b>	Time delay in milliseconds between the end of the last received frame and the starting of a new request. If set to 0, the new request will be issued as soon as the internal system is able to reschedule it.
<b>Num of repeats</b>	Number of times a certain message will be sent to the controller before reporting the communication error status.  When set to 1 the panel will report the communication error if the response to the first request packet is not correct.

Element	Description
<b>Max read block</b>	Maximum length in bytes of a data block request. It applies only to read access of Holding Registers.
<b>Max read bit block</b>	Maximum length in bits of a block request. It applies only to read access of Input Bits and Output Coils.
<b>Write Holding Register</b>	<p>Modbus function for write operations to Holding Registers. Select between the function <b>06</b> (preset single register) and function <b>16</b> (preset multiple registers).</p> <p>If function <b>06</b> is selected, the protocol will always use function <b>06</b> for writing to the controller, even when writing to multiple consecutive registers.</p> <p>If function <b>16</b> is selected, the protocol will always use function <b>16</b> to write to the controller, even for a single register write request and the <b>Max read block size</b> parameter of the query is set to <b>2</b>. The use of function <b>16</b> may result in higher communication performance.</p>
<b>Write Coils</b>	<p>Modbus function for write operations to Output Coils. Select between the function <b>05</b> (write single coil) and function <b>15</b> (write multiple coils).</p> <p>If Modbus function <b>05</b> is selected, the protocol will always use function <b>05</b> for writing to the controller, even when writing to multiple consecutive coils.</p> <p>If Modbus function <b>15</b> is selected, the protocol will always use function <b>15</b> to write to the controller, even for a single coil write request. The use of function <b>15</b> may result in higher communication performance.</p>
<b>Transmission Mode</b>	<ul style="list-style-type: none"> <li>• <b>RTU</b>: use RTU mode</li> <li>• <b>ASCII</b>: use ASCII mode</li> </ul> <p> Note: When PLC network is active, all nodes will be configured with the same Transmission Mode.</p>
<b>PLC Models</b>	<p>Allows to select between different PLC models:</p> <ul style="list-style-type: none"> <li>• <b>Modicon Modbus (1-based)</b>: Modbus implementation where all resources starts with offset 1.</li> <li>• <b>Generic Modbus (0-based)</b>: Modbus implementation where all resources starts with offset 0.</li> <li>• <b>Enron Modbus (1-based)</b>: Extends Modicon Modbus implementation with 32 bit registers memory area.</li> <li>• <b>Enron Modbus (0-base)</b>: Extends Generic Modbus implementation with 32 bit registers memory area.</li> </ul> <p> Note: The address range used in the Modbus frames is always between 0 and 65535 for the Holding Registers and between 0 and 65535 for Coils.</p>
<b>Comm...</b>	If clicked displays the communication parameters setup dialog.

Element	Description								
	 <table border="1" data-bbox="312 880 1327 1574"> <thead> <tr> <th data-bbox="312 880 954 936">Element</th> <th data-bbox="954 880 1327 936">Parameter</th> </tr> </thead> <tbody> <tr> <td data-bbox="312 936 954 1272"><b>Port</b></td> <td data-bbox="954 936 1327 1272">Serial port selection. <ul style="list-style-type: none"> <li>• <b>COM1</b>: On-board port</li> <li>• <b>COM2</b>: Optional Plug-in module plugged on slot#1 or slot#2</li> <li>• <b>COM3</b>: Optional Plug-in module plugged on slot#3 or slot#4</li> </ul> </td> </tr> <tr> <td data-bbox="312 1272 954 1328"><b>Baudrate, Parity, Data Bits, Stop bits</b></td> <td data-bbox="954 1272 1327 1328">Serial line parameters.</td> </tr> <tr> <td data-bbox="312 1328 954 1574"><b>Mode</b></td> <td data-bbox="954 1328 1327 1574">Serial port mode. Available modes: <ul style="list-style-type: none"> <li>• <b>RS-232</b>.</li> <li>• <b>RS-485</b> (2 wires).</li> <li>• <b>RS-422</b> (4 wires).</li> </ul> </td> </tr> </tbody> </table>	Element	Parameter	<b>Port</b>	Serial port selection. <ul style="list-style-type: none"> <li>• <b>COM1</b>: On-board port</li> <li>• <b>COM2</b>: Optional Plug-in module plugged on slot#1 or slot#2</li> <li>• <b>COM3</b>: Optional Plug-in module plugged on slot#3 or slot#4</li> </ul>	<b>Baudrate, Parity, Data Bits, Stop bits</b>	Serial line parameters.	<b>Mode</b>	Serial port mode. Available modes: <ul style="list-style-type: none"> <li>• <b>RS-232</b>.</li> <li>• <b>RS-485</b> (2 wires).</li> <li>• <b>RS-422</b> (4 wires).</li> </ul>
Element	Parameter								
<b>Port</b>	Serial port selection. <ul style="list-style-type: none"> <li>• <b>COM1</b>: On-board port</li> <li>• <b>COM2</b>: Optional Plug-in module plugged on slot#1 or slot#2</li> <li>• <b>COM3</b>: Optional Plug-in module plugged on slot#3 or slot#4</li> </ul>								
<b>Baudrate, Parity, Data Bits, Stop bits</b>	Serial line parameters.								
<b>Mode</b>	Serial port mode. Available modes: <ul style="list-style-type: none"> <li>• <b>RS-232</b>.</li> <li>• <b>RS-485</b> (2 wires).</li> <li>• <b>RS-422</b> (4 wires).</li> </ul>								
<b>PLC Network</b>	Multiple controllers can be connected to one HMI device. To set-up multiple connections, select <b>PLC network</b> and click <b>Add</b> to configure each slave								

## Tag Editor Settings

Path: **ProjectView** > **Config** > double-click **Tags**

1. To add a tag, click **+**: a new line is added.
2. Select **Modbus RTU** from the protocol list: tag definition dialog is displayed.


Modbus RTU

Modbus RTU

Memory Type	Offset	SubIndex
Input bits	100001	0
Data Type	Arraysize	Conversion
boolean	0	

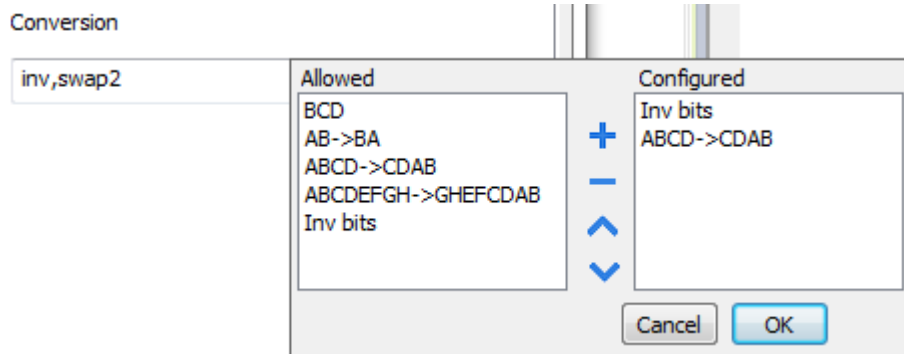
OK Cancel Apply Help

Element	Description																				
<b>Memory Type</b>	Modbus resource where tag is located.																				
	<table border="1"> <thead> <tr> <th>Memory Type</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td><b>Coil Status</b></td> <td>Coils</td> </tr> <tr> <td><b>Input Status</b></td> <td>Discrete Input</td> </tr> <tr> <td><b>Input Registers</b></td> <td>Input Registers</td> </tr> <tr> <td><b>Holding Registers</b></td> <td>Holding Registers</td> </tr> <tr> <td><b>32 bit Registers</b></td> <td>32 bit registers memory area. Available only for <b>Enron Modbus</b> PLC Models</td> </tr> <tr> <td><b>Node Override ID</b></td> <td rowspan="7">protocol parameter (see <b>Special Data Types</b> for mode details)</td> </tr> <tr> <td><b>Modicon Mode</b></td> </tr> <tr> <td><b>Serial Baudrate</b></td> </tr> <tr> <td><b>Serial Parity</b></td> </tr> <tr> <td><b>Serial Stop Bits</b></td> </tr> <tr> <td><b>Serial Mode</b></td> </tr> <tr> <td><b>Serial Done</b></td> </tr> </tbody> </table>	Memory Type	Description	<b>Coil Status</b>	Coils	<b>Input Status</b>	Discrete Input	<b>Input Registers</b>	Input Registers	<b>Holding Registers</b>	Holding Registers	<b>32 bit Registers</b>	32 bit registers memory area. Available only for <b>Enron Modbus</b> PLC Models	<b>Node Override ID</b>	protocol parameter (see <b>Special Data Types</b> for mode details)	<b>Modicon Mode</b>	<b>Serial Baudrate</b>	<b>Serial Parity</b>	<b>Serial Stop Bits</b>	<b>Serial Mode</b>	<b>Serial Done</b>
	Memory Type	Description																			
	<b>Coil Status</b>	Coils																			
	<b>Input Status</b>	Discrete Input																			
	<b>Input Registers</b>	Input Registers																			
	<b>Holding Registers</b>	Holding Registers																			
	<b>32 bit Registers</b>	32 bit registers memory area. Available only for <b>Enron Modbus</b> PLC Models																			
	<b>Node Override ID</b>	protocol parameter (see <b>Special Data Types</b> for mode details)																			
	<b>Modicon Mode</b>																				
	<b>Serial Baudrate</b>																				
	<b>Serial Parity</b>																				
<b>Serial Stop Bits</b>																					
<b>Serial Mode</b>																					
<b>Serial Done</b>																					
<b>Offset</b>	Offset address where tag is located.																				
	Offset addresses are six digits composed by one digit data type prefix + five digits resource address.																				
	<table border="1"> <thead> <tr> <th>Memory Type</th> <th>Studio Offset range</th> <th>Modicon Offset range</th> <th>Generic Modbus Offset range</th> </tr> </thead> <tbody> <tr> <td><b>Coil Status</b></td> <td>0 – 65535</td> <td rowspan="5">1 – 65536</td> <td rowspan="5">0 – 65535</td> </tr> <tr> <td><b>Input Status</b></td> <td>100000 – 165535</td> </tr> <tr> <td><b>Input Registers</b></td> <td>300000 – 365535</td> </tr> <tr> <td><b>Holding Registers</b></td> <td>400000 – 465535</td> </tr> <tr> <td><b>32 bit Registers</b></td> <td>0 – 65535</td> </tr> </tbody> </table>	Memory Type	Studio Offset range	Modicon Offset range	Generic Modbus Offset range	<b>Coil Status</b>	0 – 65535	1 – 65536	0 – 65535	<b>Input Status</b>	100000 – 165535	<b>Input Registers</b>	300000 – 365535	<b>Holding Registers</b>	400000 – 465535	<b>32 bit Registers</b>	0 – 65535				
	Memory Type	Studio Offset range	Modicon Offset range	Generic Modbus Offset range																	
	<b>Coil Status</b>	0 – 65535	1 – 65536	0 – 65535																	
	<b>Input Status</b>	100000 – 165535																			
	<b>Input Registers</b>	300000 – 365535																			
<b>Holding Registers</b>	400000 – 465535																				
<b>32 bit Registers</b>	0 – 65535																				
<b>SubIndex</b>	This allows resource offset selection within the register.																				

Element	Description		
<b>Data Type</b>	<b>Data Type</b>	<b>Memory Space</b>	<b>Limits</b>
	<b>boolean</b>	1-bit data	0 ... 1
	<b>byte</b>	8-bit data	-128 ... 127
	<b>short</b>	16-bit data	-32768 ... 32767
	<b>int</b>	32-bit data	-2.1e9 ... 2.1e9
	<b>int64</b>	64-bit data	-9.2e18 ... 9.2e18
	<b>unsignedByte</b>	8-bit data	0 ... 255
	<b>unsignedShort</b>	16-bit data	0 ... 65535
	<b>unsignedInt</b>	32-bit data	0 ... 4.2e9
	<b>uint64</b>	64-bit data	0 ... 1.8e19
	<b>float</b>	IEEE single-precision 32-bit floating point type	1.17e-38 ... 3.4e38
	<b>double</b>	IEEE double-precision 64-bit floating point type	2.2e-308 ... 1.79e308
	<b>string</b>	Array of elements containing character code defined by selected encoding	
	<b>binary</b>	Arbitrary binary data	
	 Note: to define arrays, select one of Data Type format followed by square brackets like "byte[]", "short[]"...		
<b>Arraysizes</b>	<ul style="list-style-type: none"> <li>• In case of array tag, this property represents the number of array elements.</li> <li>• In case of string tag, this property represents the maximum number of bytes available in the string tag.</li> </ul> <p>Note: number of bytes corresponds to number of string characters if Encoding property is set to UTF-8 or Latin1 in Tag Editor. If Encoding property is set to UCS-2BE, UCS-2LE, UTF-16BE or UTF-16LE one character requires 2 bytes.</p>		
<b>Conversion</b>	Conversion to be applied to the tag.		



Element	Description
---------	-------------



Depending on data type selected, the list **Allowed** shows one or more conversion types.

Value	Description
<b>Inv bits</b>	<p><b>inv</b>: Invert all the bits of the tag.</p> <p><i>Example:</i>            1001 → 0110 (in binary format)            9 → 6 (in decimal format)</p>
<b>Negate</b>	<p><b>neg</b>: Set the opposite of tag value.</p> <p><i>Example:</i>            25.36 → -25.36</p>
<b>AB -&gt; BA</b>	<p><b>swapnibbles</b>: Swap nibbles in a byte.</p> <p><i>Example:</i>            15D4 → 514D (in hexadecimal format)            5588 → 20813 (in decimal format)</p>
<b>ABCD -&gt; CDAB</b>	<p><b>swap2</b>: Swap bytes in a word.</p> <p><i>Example:</i>            9ACC → CC9A (in hexadecimal format)            39628 → 52378 (in decimal format)</p>
<b>ABCDEFGH -&gt; GHEFCADB</b>	<p><b>swap4</b>: Swap bytes in a double word.</p> <p><i>Example:</i>            32FCFF54 → 54FFFC32 (in hexadecimal format)            855441236 → 1426062386 (in decimal format)</p>
<b>ABC...NOP -&gt; OPM...DAB</b>	<p><b>swap8</b>: Swap bytes in a long word.</p> <p><i>Example:</i>            142.366 → -893553517.588905 (in decimal format)            0 1000000110            0001110010111011011001000101101000011100101011000</p>

Element	Description	
	<b>Value</b>	<b>Description</b>
		001 → 1 10000011100 1010101000010100010110110110010110110000100111 101 (in binary format)
	<b>BCD</b>	<b>bcd:</b> Separate byte in two nibbles, read them as decimal (from 0 to 9)  <i>Example:</i> 23 → 17 (in decimal format) 0001 0111 = 23 0001 = 1 (first nibble) 0111 = 7 (second nibble)
	Select conversion and click +. The selected item will be added to list <b>Configured</b> .  If more conversions are configured, they will be applied in order (from top to bottom of list <b>Configured</b> ).  Use the arrow buttons to order the configured conversions.	

## Node Override ID

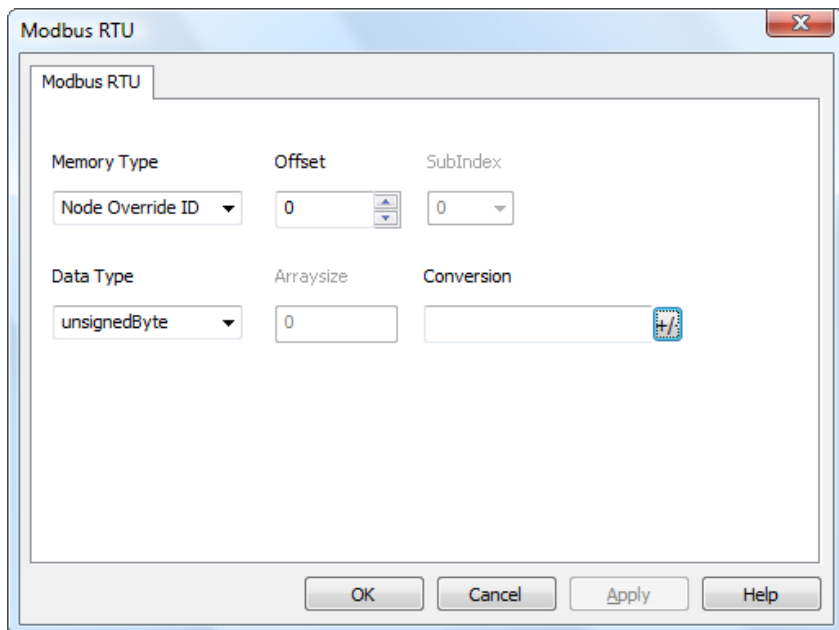
The protocol provides the special data type Node Override ID which allows you to change the node ID of the slave at runtime. This memory type is an unsigned byte.

The node Override ID is initialized with the value of the node ID specified in the project at programming time.

Node Override ID	Modbus operation
0	Communication with the controller is stopped. In case of write operation, the request will be transmitted without waiting for a reply.
1 to 254	It is interpreted as the value of the new node ID and is replaced for runtime operation.
255	Communication with the controller is stopped; no request messages are generated.



Note: Node Override ID value assigned at runtime is retained through power cycles.



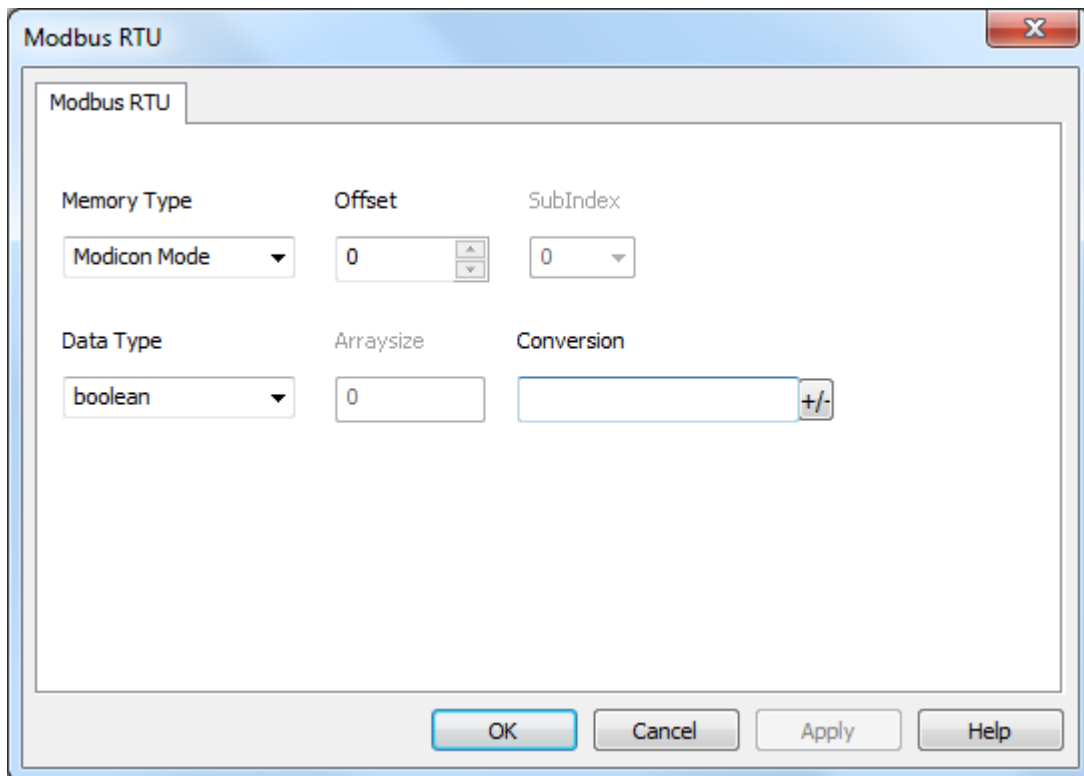
## Modicon Mode

The protocol provide a special data type that can be used to override the Modicon Mode parameter at runtime.

Modicon Mode	Description
0	Generic Modbus (0-based). Register indexes start from 0.
1	Modicon Modbus (1-based). Register indexes start from 1.



Note: Modicon Mode parameter value assigned at runtime is retained through power cycles.

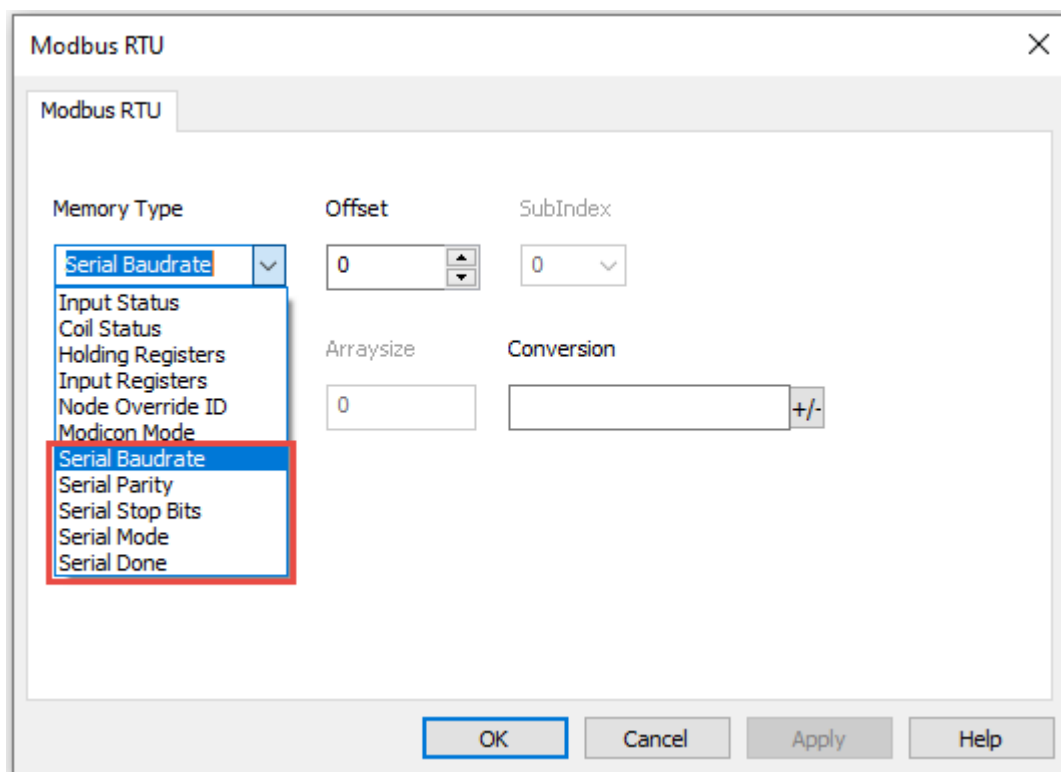


## Serial Parameters Override

The protocol provide special data types that can be used to override the serial parameters at runtime.

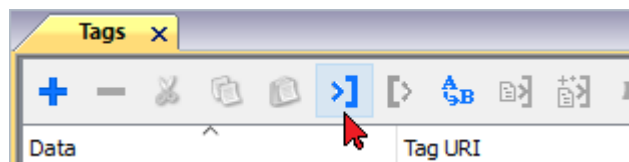
Parameter	Description								
<b>Serial Baudrate</b>	unsigned 32 bit value for baudrate overriding. Possible values are 150, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200.								
<b>Serial Parity</b>	unsigned 8 bit value for parity overriding. Possible values are described in the following list. <table border="1" data-bbox="368 1435 1506 1675"> <thead> <tr> <th>Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>none parity</td> </tr> <tr> <td>1</td> <td>even parity</td> </tr> <tr> <td>2</td> <td>odd parity</td> </tr> </tbody> </table>	Value	Description	0	none parity	1	even parity	2	odd parity
Value	Description								
0	none parity								
1	even parity								
2	odd parity								
<b>Serial Stop Bits</b>	unsigned 8 bit value for stop bits overriding. Possible values are 1, 2.								
<b>Serial Mode</b>	unsigned 8 bit value for serial mode overriding. Possible values are described in the following list.								

Parameter	Description	
	<b>Value</b>	<b>Description</b>
	0	RS-232 mode
	1	RS-485 mode
	2	RS-422 mode
<b>Serial Done</b>	Set to 1 to overwrite the communication line parameters. The parameters are processed all together only when this variable is set to value 1	

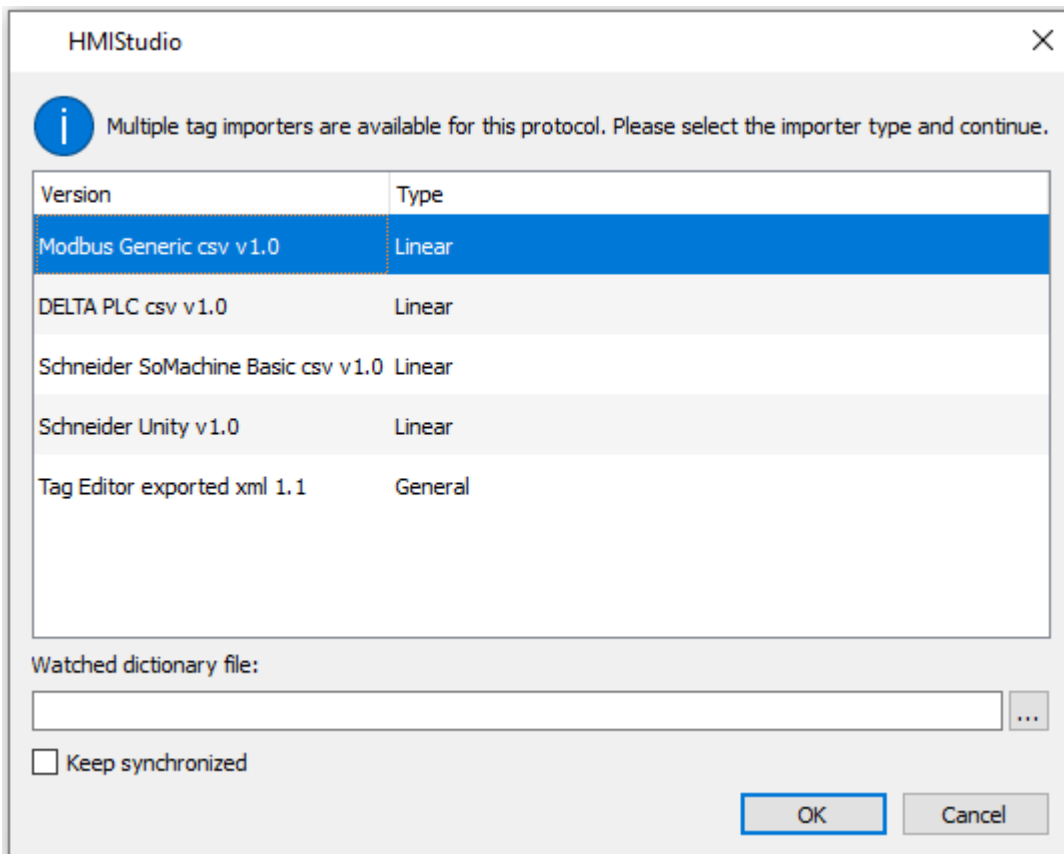



## Tag Import

Select the driver in Tag Editor and click on the **Import Tags** button to start the importer.



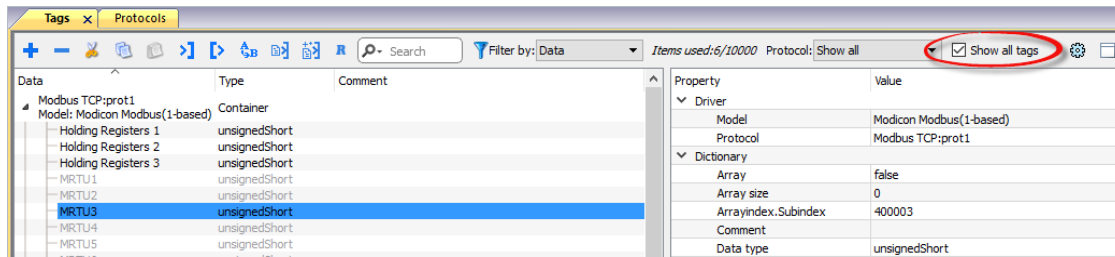
The following dialog shows which importer type can be selected.




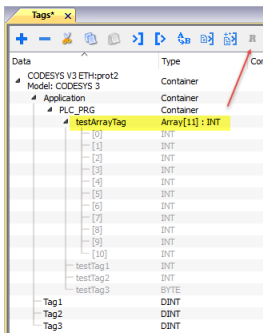
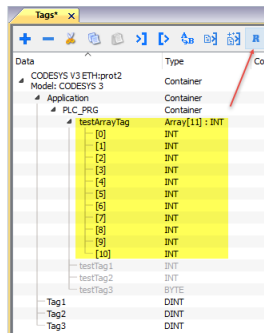


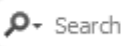

Type	Description
<b>Modbus Generic csv v1.0</b> Linear	Requires a <b>.csv</b> file. All variables will be displayed at the same level.
<b>DELTA PLC csv v1.0</b>	Requires a <b>.csv</b> file. All variables will be displayed at the same level.
<b>Schneider Unity v1.0</b> Linear	Requires a <b>.uny</b> file. The file containing symbols must be exported in <b>.txt</b> format and later renamed as <b>.uny</b> . The importer considers only variables located at fixed address and disregards arrays of strings. All other arrays, except for boolean type, are expanded.
<b>Tag Editor exported xml</b>	Select this importer to read a generic XML file exported from Tag Editor by appropriate button. 

Once the importer has been selected, locate the symbol file and click **Open**.

The tags available within the Dictionary but not imported into the project are gray and are visible only when the "Show all tags" check box is selected.



Toolbar item	Description
	<p><b>Import Tag(s).</b></p> <p>Select tags to be imported and click on this icon to add tags from tag dictionary to the project</p>
	<p><b>Update Tag(s).</b></p> <p>Click on this icon to update the tags in the project, due a new dictionary import.</p>
	<p>Check this box to import all sub-elements of a tag.</p> <p>Example of both checked and unchecked result:</p> <div style="display: flex; justify-content: space-around;">   </div>

 Search  Filter by: <input type="text" value="Tag name"/>	<p>Searches tags in the dictionary basing on filter combobox item selected.</p>
---	---

## Modbus Generic csv file structure

This protocol supports the import of tag information when provided in **.csv** format according to the following format:

NodeID, TagName, MemoryType, Address, DataFormat, ..., [Comment]



Note: Fields in brackets are optional as well as fields between Data Format and Comment.

Field	Description
<b>NodeID</b>	Node the tag belongs to
<b>TagName</b>	Tag description
<b>MemoryType</b>	<ul style="list-style-type: none"> <li>• OUP</li> <li>• INP</li> <li>• IREG</li> <li>• HREG</li> </ul>
<b>Address</b>	Offset compatible with Modbus notation
<b>DataFormat</b>	Data type in internal notation. See "Programming concepts" section in the main manual.
<b>Comment</b>	Optional additional description.

### Tag file example

Example of .csv line:

```
2,Holding Register 1, HREG, 400001, unsignedShort,
```



Note: This line has no comment. When the Comment is missing, the comma as a terminator character is mandatory.

Example of .csv file:

```
1, Holding Register 2 TagName, HREG, 400002, unsignedShort, comment for this register
```

```
1, HRegTag 400011_15 TagName, HREG, 400011.15, boolean, comment for this register
```

```
2, InpRegTag 300501 TagName, IREG, 300501, unsignedShort, comment for this register
```

```
1, InpRegTag 301999_8 TagName, IREG, 301999.8, boolean, comment for this register
```

```
27, OutputTag 999 TagName, OUP, 999, boolean, comment for this register
```

```
11, InputTag 100101 TagName, INP, 100101, boolean, comment for this register
```

```
2, HRegTag 409999 TagName, HREG, 409999, unsignedShort, comment for this register
```

### Communication status

Current communication status can be displayed using System Variables. See "System Variables" section in the main manual.

Codes supported for this communication driver:



---

<b>Error</b>	<b>Cause</b>	<b>Action</b>
<b>No response</b>	No reply within the specified timeout.	Check if the controller is connected and properly configured to get network access.
<b>Incorrect node address in response</b>	The device received a response with an invalid node address from the controller .	-
<b>The received message too short</b>	The device received a response with an invalid format from the controller .	-
<b>Incorrect writing data acknowledge</b>	The controller did not accept a write request.	Check if project data is consistent with the controller resources.

# Modbus RTU Server

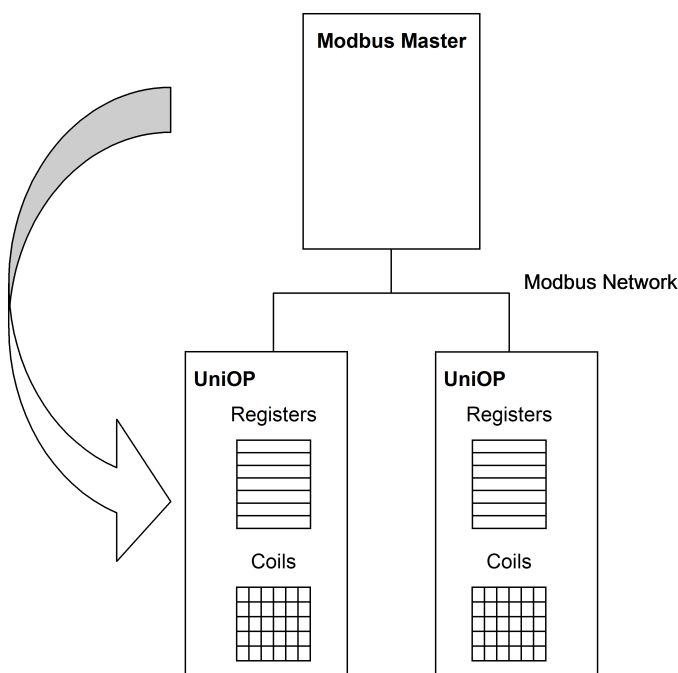
Modbus RTU Server communication driver allows connecting the HMI device as a slave in a Modbus RTU network. Standard Modbus messages are used for information exchange.

This approach allows connecting HMI devices to SCADA systems through the universally supported Modbus RTU communication protocol.

## Principle of operation

This communication driver implements a Modbus RTU slave unit in the HMI device. A subset of the complete range of Modbus function codes is supported. The available function codes allow data transfer between the master and the slave.

The following diagram shows the system architecture.



The HMI device is actually simulating the communication interface of a PLC: Coils and Registers are respectively boolean and 16 bit integers.

The device always access data in its internal memory. Data can be transferred to and from the Modbus Master only on initiative of the Master itself.

## Implementation details

This Modbus RTU slave implementation supports only a subset of the standard Modbus function codes.

Code	Function	Description
01	Read Coil Status	Reads multiple bits in the device Coil area.
03	Read Holding Registers	Read multiple device Registers.

Code	Function	Description
05	Force Single Coil	Forces a single device Coil to either ON or OFF.
06	Preset Single Register	Presets a value in a device Register.
08	Loopback Diagnostic Test	Only sub function 00 (Return Query Data) is supported.
15	Force Multiple Coils	Forces multiple device Coils to either ON or OFF.
16	Preset Multiple Registers	Presets value in multiple device Registers.
17	Report Slave ID	Returns diagnostic information of the controller present at the slave address.
23	Read Write Multiple Registers	Read & presets values in multiple device Registers

### Exception Codes

Code	Description
01	<b>Illegal Function.</b> the function code received in the query is not supported
02	<b>Illegal Data Address.</b> Data Address received in the query exceeds the predefined data range (see <b>Tag Definition</b> for detailed ranges of all types).
03	<b>Illegal Data Value.</b> A sub function other than 00 is specified in Loopback Diagnostic Test (Code 08).

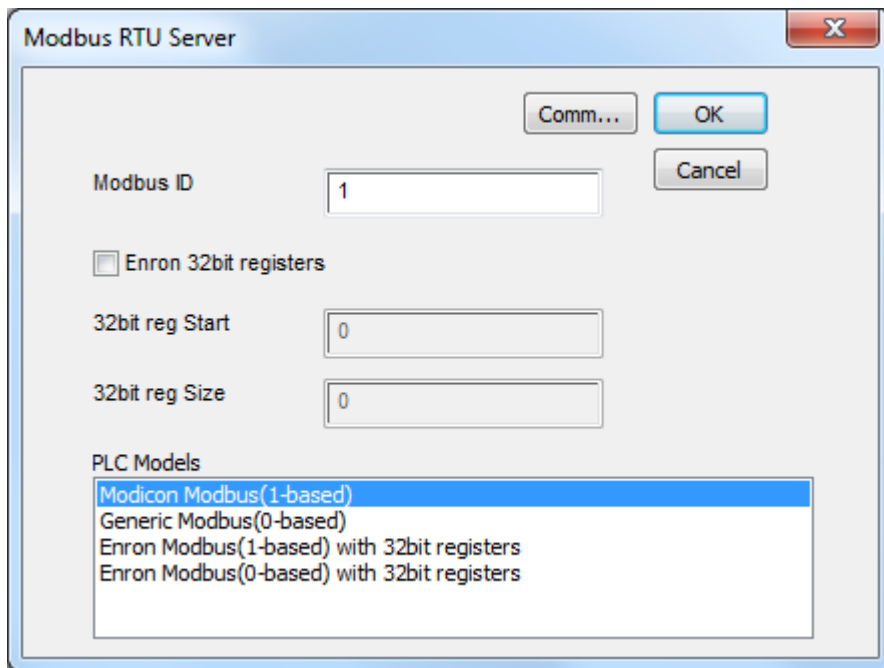
## Protocol Editor Settings



### Adding a protocol

To configure the protocol:

1. In **Config** node double-click **Protocols**.
2. To add a driver, click **+**: a new line is added.
3. Select the protocol from the **PLC** list.

The protocol configuration dialog is displayed.



Element	Description
<b>Modbus ID</b>	Modbus node ID. Every Modbus server device in the network must have its own Modbus ID.
<b>Enron 32bit registers</b>	<p>If selected, allows to define the first register address and the number of registers for 32 bit registers memory area.</p> <p> Note: 32 bit registers are available only for <b>Enron Modbus</b> PLC Models.</p>
<b>32bit reg Start</b>	32 bit registries memory area definition.
<b>32bit reg Size</b>	<p><b>Start</b> value represents the first register address.</p> <p><b>Size</b> value represents the number of registries.</p> <p> Note: A request to one of the registries inside this area gives a 4 byte answer.</p>
<b>PLC Models</b>	<p>Allows to select between different PLC models:</p> <ul style="list-style-type: none"> <li>• <b>Modicon Modbus (1-based)</b>: Modbus implementation where all resources starts with offset 1.</li> <li>• <b>Generic Modbus (0-based)</b>: Modbus implementation where all resources starts with offset 0.</li> <li>• <b>Enron Modbus (1-based)</b>: Extends Modicon Modbus implementation with 32 bit registers memory area.</li> <li>• <b>Enron Modbus (0-base)</b>: Extends Generic Modbus implementation with 32 bit registers memory area.</li> </ul>

Element	Description
---------	-------------

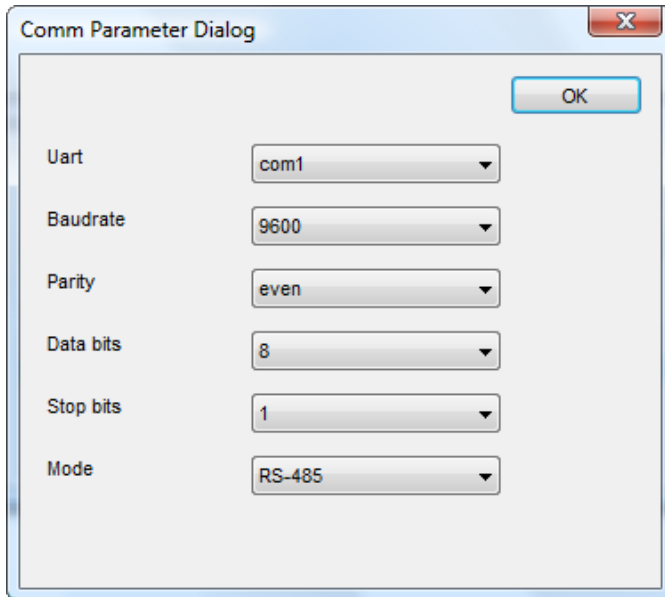


Note: The address range used in the Modbus frames is always between 0 and 65535 for the Holding Registers and between 0 and 65535 for Coils.

Comm...

If clicked, displays the communication parameters setup dialog.

You have to set parameters according to the values programmed in Modbus Master.

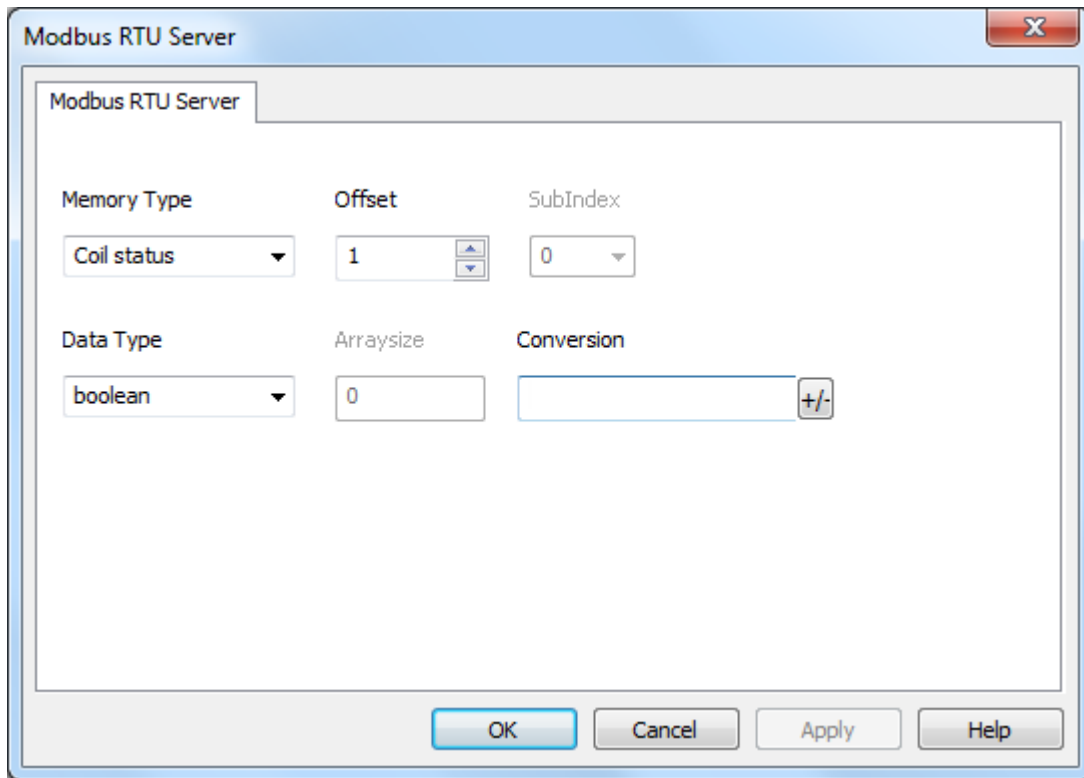


Element	Description
<b>Uart</b>	Serial port selection. <ul style="list-style-type: none"> <li>• <b>COM1</b>: On-board port</li> <li>• <b>COM2</b>: Optional Plug-in module plugged on slot#1 or slot#2</li> <li>• <b>COM3</b>: Optional Plug-in module plugged on slot#3 or slot#4</li> </ul>
<b>Baudrate, Parity, Data bits, Stop bits</b>	Serial line parameters.
<b>Mode</b>	Serial port mode. Available options: <ul style="list-style-type: none"> <li>• <b>RS-232</b></li> <li>• <b>RS-485</b> (2 wires)</li> <li>• <b>RS-422</b> (4 wires)</li> </ul>

## Tag Editor Settings

Path: *ProjectView* > *Config* > double-click *Tags*

1. To add a tag, click **+**: a new line is added.
2. Select **Modbus RTU Server** from the protocol list: tag definition dialog is displayed.




The image shows a dialog box titled "Modbus RTU Server" with a close button (X) in the top right corner. The dialog contains the following fields:

Memory Type	Offset	SubIndex
Coil status	1	0
Data Type	Arraysize	Conversion
boolean	0	

At the bottom of the dialog, there are four buttons: "OK", "Cancel", "Apply", and "Help".

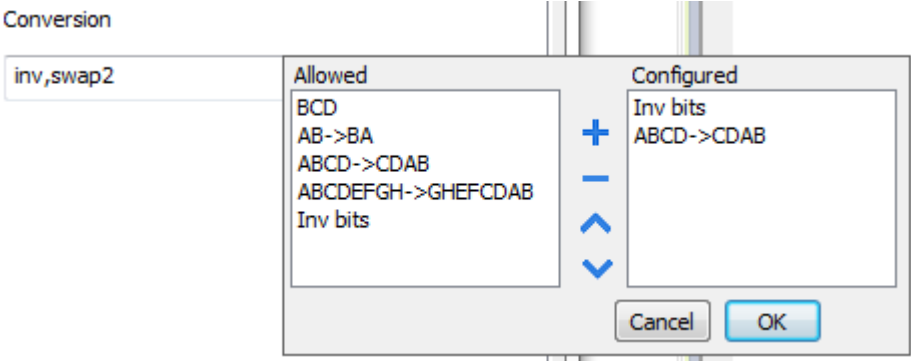
Element	Description																				
<b>Memory Type</b>	Modbus resource where tag is located.																				
	<table border="1"> <thead> <tr> <th>Memory Type</th> <th>Modbus Resource</th> </tr> </thead> <tbody> <tr> <td><b>Coil Status</b></td> <td>Coils</td> </tr> <tr> <td><b>Input Status</b></td> <td>Discrete Input</td> </tr> <tr> <td><b>Input Registers</b></td> <td>Input Registers</td> </tr> <tr> <td><b>Holding Registers</b></td> <td>Holding Registers</td> </tr> <tr> <td><b>32 bit Registers</b></td> <td>32 bit registers memory area. Available only for <b>Enron Modbus</b> PLC Models</td> </tr> <tr> <td><b>Node Override ID</b></td> <td rowspan="7">protocol parameter (see <b>Special Data Types</b> for mode details)</td> </tr> <tr> <td><b>Modicon Mode</b></td> </tr> <tr> <td><b>Serial Baudrate</b></td> </tr> <tr> <td><b>Serial Parity</b></td> </tr> <tr> <td><b>Serial Stop Bits</b></td> </tr> <tr> <td><b>Serial Mode</b></td> </tr> <tr> <td><b>Serial Done</b></td> </tr> </tbody> </table>	Memory Type	Modbus Resource	<b>Coil Status</b>	Coils	<b>Input Status</b>	Discrete Input	<b>Input Registers</b>	Input Registers	<b>Holding Registers</b>	Holding Registers	<b>32 bit Registers</b>	32 bit registers memory area. Available only for <b>Enron Modbus</b> PLC Models	<b>Node Override ID</b>	protocol parameter (see <b>Special Data Types</b> for mode details)	<b>Modicon Mode</b>	<b>Serial Baudrate</b>	<b>Serial Parity</b>	<b>Serial Stop Bits</b>	<b>Serial Mode</b>	<b>Serial Done</b>
	Memory Type	Modbus Resource																			
	<b>Coil Status</b>	Coils																			
	<b>Input Status</b>	Discrete Input																			
	<b>Input Registers</b>	Input Registers																			
	<b>Holding Registers</b>	Holding Registers																			
	<b>32 bit Registers</b>	32 bit registers memory area. Available only for <b>Enron Modbus</b> PLC Models																			
	<b>Node Override ID</b>	protocol parameter (see <b>Special Data Types</b> for mode details)																			
	<b>Modicon Mode</b>																				
	<b>Serial Baudrate</b>																				
	<b>Serial Parity</b>																				
	<b>Serial Stop Bits</b>																				
<b>Serial Mode</b>																					
<b>Serial Done</b>																					
<b>Offset</b>	Offset address where tag is located.																				
	Offset addresses are six digits composed by one digit data type prefix + five digits resource address.																				
	<table border="1"> <thead> <tr> <th>Memory Type</th> <th>Studio Offset range</th> <th>Modicon Offset range</th> <th>Generic Modbus Offset range</th> </tr> </thead> <tbody> <tr> <td><b>Coil Status</b></td> <td>0 – 65535</td> <td rowspan="5">1 – 65536</td> <td rowspan="5">0 – 65535</td> </tr> <tr> <td><b>Input Status</b></td> <td>100000 – 165535</td> </tr> <tr> <td><b>Input Registers</b></td> <td>300000 – 365535</td> </tr> <tr> <td><b>Holding Registers</b></td> <td>400000 – 465535</td> </tr> <tr> <td><b>32 bit Registers</b></td> <td>0 – 65535</td> </tr> </tbody> </table>	Memory Type	Studio Offset range	Modicon Offset range	Generic Modbus Offset range	<b>Coil Status</b>	0 – 65535	1 – 65536	0 – 65535	<b>Input Status</b>	100000 – 165535	<b>Input Registers</b>	300000 – 365535	<b>Holding Registers</b>	400000 – 465535	<b>32 bit Registers</b>	0 – 65535				
	Memory Type	Studio Offset range	Modicon Offset range	Generic Modbus Offset range																	
	<b>Coil Status</b>	0 – 65535	1 – 65536	0 – 65535																	
	<b>Input Status</b>	100000 – 165535																			
	<b>Input Registers</b>	300000 – 365535																			
<b>Holding Registers</b>	400000 – 465535																				
<b>32 bit Registers</b>	0 – 65535																				
<b>SubIndex</b>	This allows resource offset selection within the register.																				

Element	Description		
Data type	<b>Data Type</b>	<b>Memory Space</b>	<b>Limits</b>
	<b>boolean</b>	1-bit data	0 ... 1
	<b>byte</b>	8-bit data	-128 ... 127
	<b>short</b>	16-bit data	-32768 ... 32767
	<b>int</b>	32-bit data	-2.1e9 ... 2.1e9
	<b>int64</b>	64-bit data	-9.2e18 ... 9.2e18
	<b>unsignedByte</b>	8-bit data	0 ... 255
	<b>unsignedShort</b>	16-bit data	0 ... 65535
	<b>unsignedInt</b>	32-bit data	0 ... 4.2e9
	<b>uint64</b>	64-bit data	0 ... 1.8e19
	<b>float</b>	IEEE single-precision 32-bit floating point type	1.17e-38 ... 3.4e38
	<b>double</b>	IEEE double-precision 64-bit floating point type	2.2e-308 ... 1.79e308
	<b>string</b>	Array of elements containing character code defined by selected encoding	
	<b>binary</b>	Arbitrary binary data	
 Note: to define arrays, select one of Data Type format followed by square brackets like "byte[]", "short[]"...			

**Arrays size** When configuring array or string tags, this option define the amount of array elements or characters of the string.

**Conversion** Conversion to be applied to the tag.

Conversion





Element	Description												
	<p>Depending on data type selected, the list <b>Allowed</b> shows one or more conversion types.</p> <table border="1" data-bbox="268 360 1222 1283"> <thead> <tr> <th data-bbox="268 360 456 416">Value</th> <th data-bbox="456 360 1222 416">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="268 416 456 595"><b>Inv bits</b></td> <td data-bbox="456 416 1222 595"> <b>inv</b>: Invert all the bits of the tag.   <i>Example:</i>            1001 → 0110 (in binary format)            9 → 6 (in decimal format)         </td> </tr> <tr> <td data-bbox="268 595 456 741"><b>Negate</b></td> <td data-bbox="456 595 1222 741"> <b>neg</b>: Set the opposite of tag value.   <i>Example:</i>            25.36 → -25.36         </td> </tr> <tr> <td data-bbox="268 741 456 920"><b>AB -&gt; BA</b></td> <td data-bbox="456 741 1222 920"> <b>swapnibbles</b>: Swap nibbles in a byte.   <i>Example:</i>            15D4 → 514D (in hexadecimal format)            5588 → 20813 (in decimal format)         </td> </tr> <tr> <td data-bbox="268 920 456 1099"><b>ABCD -&gt; CDAB</b></td> <td data-bbox="456 920 1222 1099"> <b>swap2</b>: Swap bytes in a word.   <i>Example:</i>            9ACC → CC9A (in hexadecimal format)            39628 → 52378 (in decimal format)         </td> </tr> <tr> <td data-bbox="268 1099 456 1283"><b>ABCDEFGH -&gt; GHEFCBAB</b></td> <td data-bbox="456 1099 1222 1283"> <b>swap4</b>: Swap bytes in a double word.   <i>Example:</i>            32FCFF54 → 54FFFC32 (in hexadecimal format)            855441236 → 1426062386 (in decimal format)         </td> </tr> </tbody> </table>	Value	Description	<b>Inv bits</b>	<b>inv</b> : Invert all the bits of the tag.  <i>Example:</i> 1001 → 0110 (in binary format) 9 → 6 (in decimal format)	<b>Negate</b>	<b>neg</b> : Set the opposite of tag value.  <i>Example:</i> 25.36 → -25.36	<b>AB -&gt; BA</b>	<b>swapnibbles</b> : Swap nibbles in a byte.  <i>Example:</i> 15D4 → 514D (in hexadecimal format) 5588 → 20813 (in decimal format)	<b>ABCD -&gt; CDAB</b>	<b>swap2</b> : Swap bytes in a word.  <i>Example:</i> 9ACC → CC9A (in hexadecimal format) 39628 → 52378 (in decimal format)	<b>ABCDEFGH -&gt; GHEFCBAB</b>	<b>swap4</b> : Swap bytes in a double word.  <i>Example:</i> 32FCFF54 → 54FFFC32 (in hexadecimal format) 855441236 → 1426062386 (in decimal format)
Value	Description												
<b>Inv bits</b>	<b>inv</b> : Invert all the bits of the tag.  <i>Example:</i> 1001 → 0110 (in binary format) 9 → 6 (in decimal format)												
<b>Negate</b>	<b>neg</b> : Set the opposite of tag value.  <i>Example:</i> 25.36 → -25.36												
<b>AB -&gt; BA</b>	<b>swapnibbles</b> : Swap nibbles in a byte.  <i>Example:</i> 15D4 → 514D (in hexadecimal format) 5588 → 20813 (in decimal format)												
<b>ABCD -&gt; CDAB</b>	<b>swap2</b> : Swap bytes in a word.  <i>Example:</i> 9ACC → CC9A (in hexadecimal format) 39628 → 52378 (in decimal format)												
<b>ABCDEFGH -&gt; GHEFCBAB</b>	<b>swap4</b> : Swap bytes in a double word.  <i>Example:</i> 32FCFF54 → 54FFFC32 (in hexadecimal format) 855441236 → 1426062386 (in decimal format)												

Element	Description	
	<b>Value</b>	<b>Description</b>
	<b>ABC...NOP - &gt; OPM...DAB</b>	<p><b>swap8:</b> Swap bytes in a long word.</p> <p>Example:            142.366 → -893553517.588905 (in decimal format)            0 10000000110            0001110010111011011001000101101000011100101011000            001            →            1 10000011100            1010101000010100010110110110110010110110000100111            101            (in binary format)</p>
	<b>BCD</b>	<p><b>bcd:</b> Separate byte in two nibbles, read them as decimal (from 0 to 9)</p> <p><i>Example:</i>            23 → 17 (in decimal format)            0001 0111 = 23            0001 = 1 (first nibble)            0111 = 7 (second nibble)</p>
<p>Select conversion and click +. The selected item will be added to list <b>Configured</b>.</p> <p>If more conversions are configured, they will be applied in order (from top to bottom of list <b>Configured</b>).</p> <p>Use the arrow buttons to order the configured conversions.</p>		

## Node Override ID

The protocol provides the special data type Node Override ID which allows you to change the node ID of the slave at runtime. This memory type is an unsigned byte.

The node Override ID is initialized with the value of the node ID specified in the project at programming time.

Node Override ID	Modbus operation
0	Communication with the slave is stopped. In case of write operation, the device will not respond to request frames.
1 to 255	It is interpreted as the value of the new node ID and is replaced for runtime operation.



Note: Node Override ID value assigned at runtime is retained through power cycles.

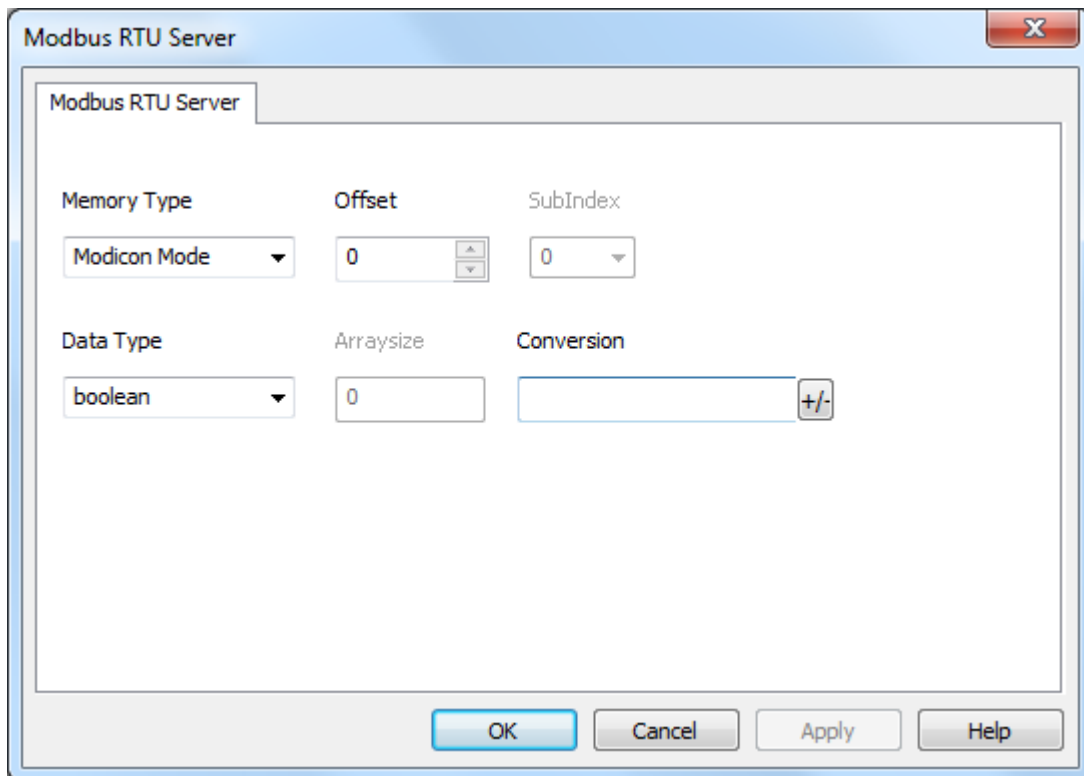
## Modicon Mode

The protocol provide a special data type that can be used to override the Modicon Mode parameter at runtime.

Modicon Mode	Description
0	Generic Modbus (0-based). Register indexes start from 0.
1	Modicon Modbus (1-based). Register indexes start from 1.



Note: Modicon Mode parameter value assigned at runtime is retained through power cycles.

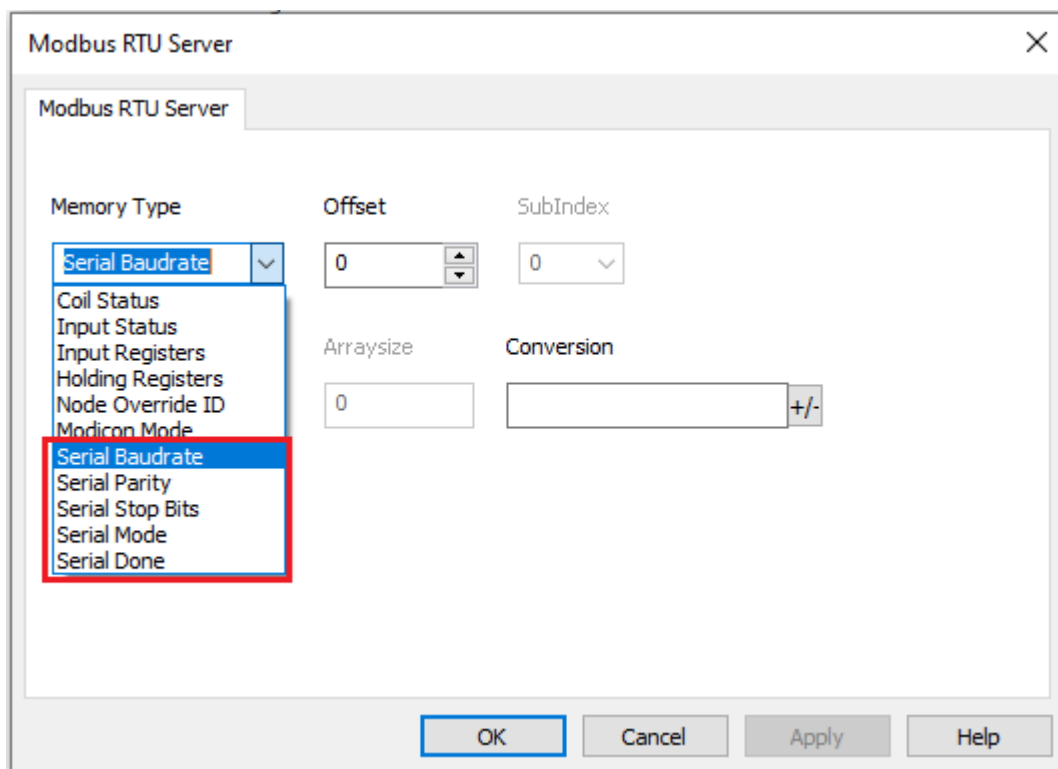


## Serial Parameters Override

The protocol provide special data types that can be used to override the serial parameters at runtime.

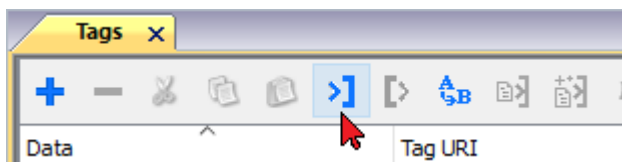
Parameter	Description								
<b>Serial Baudrate</b>	unsigned 32 bit value for baudrate overriding. Possible values are 150, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200.								
<b>Serial Parity</b>	unsigned 8 bit value for parity overriding. Possible values are described in the following list. <table border="1" data-bbox="368 1435 1506 1675"> <thead> <tr> <th>Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>none parity</td> </tr> <tr> <td>1</td> <td>even parity</td> </tr> <tr> <td>2</td> <td>odd parity</td> </tr> </tbody> </table>	Value	Description	0	none parity	1	even parity	2	odd parity
Value	Description								
0	none parity								
1	even parity								
2	odd parity								
<b>Serial Stop Bits</b>	unsigned 8 bit value for stop bits overriding. Possible values are 1, 2.								
<b>Serial Mode</b>	unsigned 8 bit value for serial mode overriding. Possible values are described in the following list.								

Parameter	Description	
	<b>Value</b>	<b>Description</b>
	0	RS-232 mode
	1	RS-485 mode
	2	RS-422 mode
<b>Serial Done</b>	Set to 1 to overwrite the communication line parameters. The parameters are processed all together only when this variable is set to value 1	

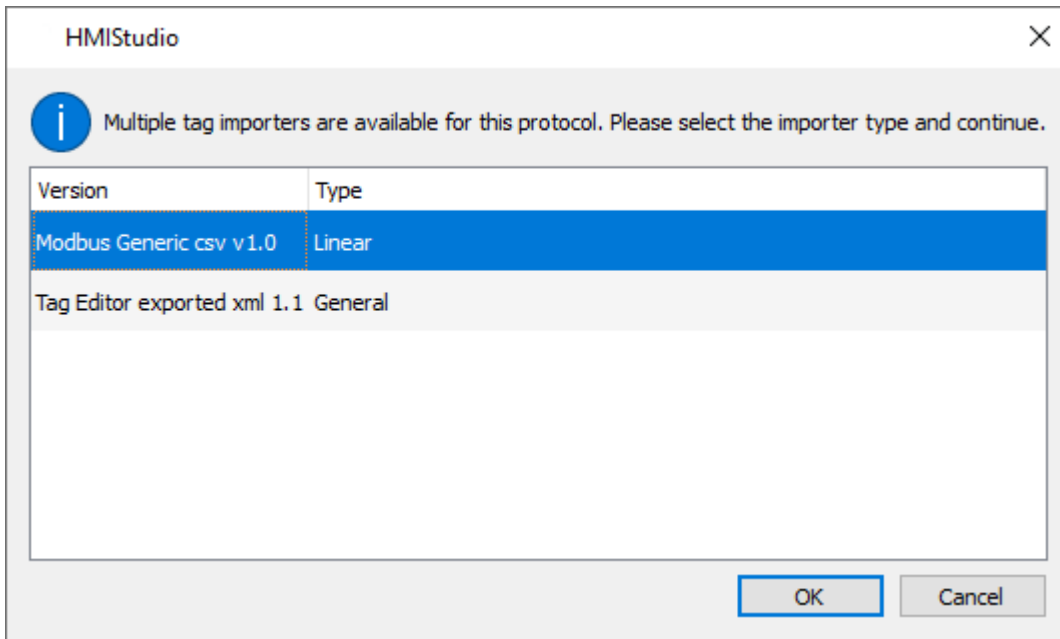


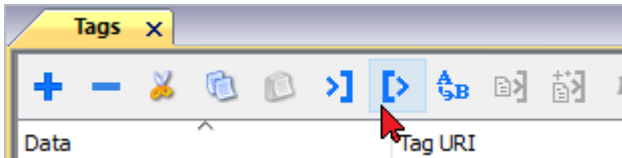
## Tag Import

Select the driver in Tag Editor and click on the **Import Tags** button to start the importer.



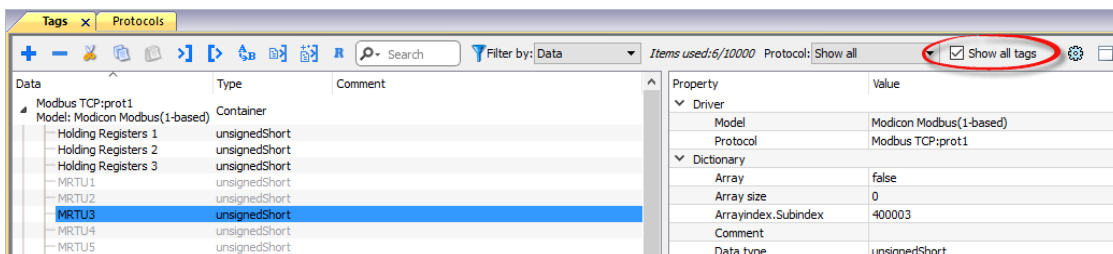
The following dialog shows which importer type can be selected.




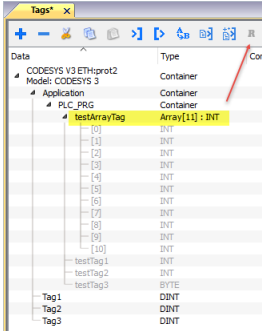
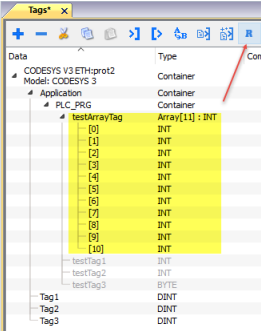
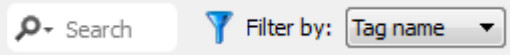


Type	Description
<b>Modbus Generic csv v1.0 Linear</b>	Requires a <b>.csv</b> file. All variables will be displayed at the same level.
<b>Tag Editor exported xml</b>	Select this importer to read a generic XML file exported from Tag Editor by appropriate button. 

Once the importer has been selected, locate the symbol file and click **Open**.

The tags available within the Dictionary but not imported into the project are gray and are visible only when the "Show all tags" check box is selected.



Toolbar item	Description
	<p><b>Import Tag(s).</b></p> <p>Select tags to be imported and click on this icon to add tags from tag dictionary to the project</p>
	<p><b>Update Tag(s).</b></p> <p>Click on this icon to update the tags in the project, due a new dictionary import.</p>
	<p>Check this box to import all sub-elements of a tag.</p> <p>Example of both checked and unchecked result:</p> <div style="display: flex; justify-content: space-around;">   </div>
	<p>Searches tags in the dictionary basing on filter combo-box item selected.</p>

## Modbus Generic csv file structure

This protocol supports the import of tag information when provided in **.csv** format according to the following format:

`NodeID, TagName, MemoryType, Address, DataFormat, ..., [Comment]`



Note: Fields in brackets are optional as well as fields between Data Format and Comment.

Field	Description
<b>NodeID</b>	Node the tag belongs to
<b>TagName</b>	Tag description
<b>MemoryType</b>	<ul style="list-style-type: none"> <li>• OUTP</li> <li>• INP</li> <li>• IREG</li> <li>• HREG</li> </ul>
<b>Address</b>	Offset compatible with Modbus notation

Field	Description
DataFormat	Data type in internal notation. See "Programming concepts" section in the main manual.
Comment	Optional additional description.

## Tag file example

Example of .csv line:

```
2, Holding Register 1, HREG, 400001, unsignedShort,
```



Note: This line has no comment. When the Comment is missing, the comma as a terminator character is mandatory.

Example of .csv file:

```
1, Holding Register 2 TagName, HREG, 400002, unsignedShort, comment for this register
```

```
1, HRegTag 400011_15 TagName, HREG, 400011.15, boolean, comment for this register
```

```
2, InpRegTag 300501 TagName, IREG, 300501, unsignedShort, comment for this register
```

```
1, InpRegTag 301999_8 TagName, IREG, 301999.8, boolean, comment for this register
```

```
27, OutputTag 999 TagName, OOTP, 999, boolean, comment for this register
```

```
11, InputTag 100101 TagName, INP, 100101, boolean, comment for this register
```

```
2, HRegTag 409999 TagName, HREG, 409999, unsignedShort, comment for this register
```

## Communication status

Current communication status can be displayed using system variables. This communication protocol acts as server and doesn't return any specific Protocol Error Message.

See "System Variables" section in the main manual.



# Modbus TCP

Various Modbus TCP-capable devices can be connected to HMI devices. To set-up your Modbus TCP device, please refer to the documentation you have received with the device.

The implementation of the protocol operates as a Modbus TCP client only.

## Implementation details

This Modbus TCP implementation supports only a subset of the Modbus TCP standard function codes.

Code	Function	Description
01	Read Coil Status	Reads multiple bits in the HMI device Coil area.
02	Read Input Status	Reads the ON/OFF status of the discrete inputs (1x reference) in the slave.
03	Read Holding Registers	Reads multiple registers.
04	Read Input Registers	Reads the binary contents of input registers (3x reference) in the slave.
05	Force Single Coil	Forces a single coil to either ON or OFF.
06	Preset Single Register	Writes a value to one register.
15	Write Multiple Coils	Writes each coil in a sequence of coils to either ON or OFF.
16	Preset Multiple Registers	Writes values to a block of registers in sequence.

## Protocol Editor Settings

### Adding a protocol

To configure the protocol:

1. In **Config** node double-click **Protocols**.
2. To add a driver, click **+**: a new line is added.
3. Select the protocol from the **PLC** list.

The protocol configuration dialog is displayed.

Modbus TCP
✕

PLC Network

Alias

IP address

Port

use UDP/IP

Encapsulated RTU

Timeout (ms)

Server Busy Timeout (ms)

Busy Retry Time (ms)

Modbus ID

Max read block

Max read bit block

Write Holding Register


Write Coils

PLC Models

Modicon Modbus(1-based)  
Generic Modbus(0-based)  
Enron Modbus(1-based) with 32bit registers  
Enron Modbus(0-based) with 32bit registers  
EPSON Robot

Element	Description
<b>Alias</b>	Name identifying nodes in network configurations. The name will be added as a prefix to each tag name imported for each network node.
<b>IP address</b>	Address of the controller.
<b>Port</b>	Port number used by the Modbus TCP driver. The default value is <b>502</b> and can be changed when the communication goes through routers or Internet gateways where the default port number is already in use.
<b>use UDP/IP</b>	If selected, the protocol will use connectionless UDP datagrams.
<b>Encapsulate</b>	If selected, the protocol will use serial RTU protocol over Ethernet instead of Modbus TCP

Element	Description
<b>d RTU</b>	protocol, independently from TCP or UDP usage.
<b>Timeout (ms)</b>	Time delay in milliseconds between two retries in case of missing response from the server device.
<b>Modbus ID</b>	Usually used when communicating over Ethernet-to-serial gateways and then interpreted as the Slave ID. This value is simply copied into the Unit Identifier field of the Modbus TCP communication frame. This must correspond to server configuration. In most cases, server answers to Modbus ID 1, so this parameter can be left 1.
<b>Max read block</b>	Maximum length in bytes of a data block request. It applies only to read access of Holding Registers.
<b>Max read bit block</b>	Maximum length in bits of a block request. It applies only to read access of Input Bits and Output Coils.
<b>Write Holding Register</b>	<p>Modbus function for write operations to Holding Registers. Select between the function <b>06</b> (preset single register) and function <b>16</b> (preset multiple registers).</p> <p>If <b>06</b> is selected, the protocol will always use function <b>06</b> for writing to the controller, even when writing to multiple consecutive registers.</p> <p>If <b>16</b> is selected, the protocol will always use function <b>16</b> to write to the controller, even for a single register write request and the <b>Max read block size</b> parameter of the query is set to <b>2</b>. The use of function <b>16</b> may result in higher communication performance.</p> <p>If <b>Auto</b> is selected, the protocol will use both function <b>06</b> or function <b>16</b> depending on number of registries to be written.</p>
<b>Write Coils</b>	<p>Modbus function for write operations to Output Coils. Select between the function <b>05</b> (write single coil) and function <b>15</b> (write multiple coils).</p> <p>If Modbus function <b>05</b> is selected, the protocol will always use function <b>05</b> for writing to the controller, even when writing to multiple consecutive coils.</p> <p>If Modbus function <b>15</b> is selected, the protocol will always use function <b>15</b> to write to the controller, even for a single coil write request. The use of function <b>15</b> may result in higher communication performance.</p>

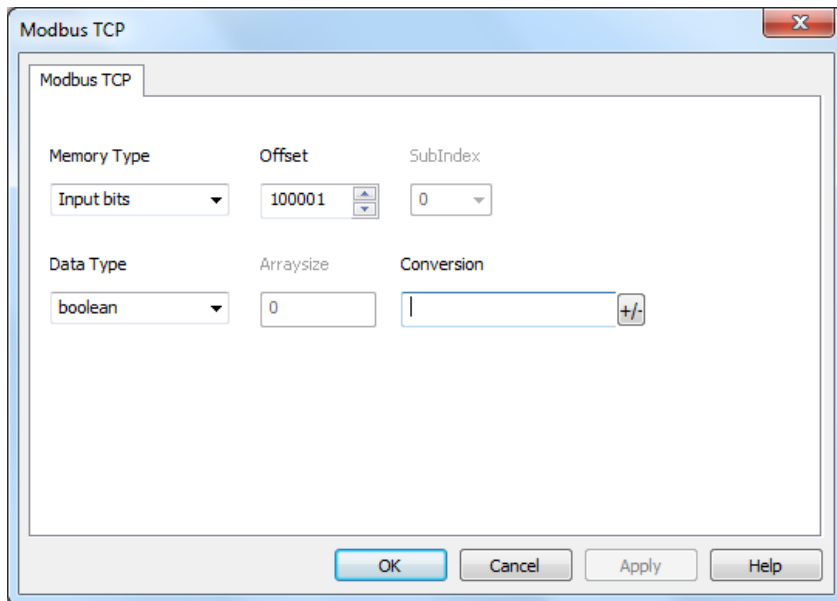
Element	Description
<b>PLC Models</b>	<p>Allows to select between different PLC models:</p> <ul style="list-style-type: none"> <li>• <b>Modicon Modbus (1-based)</b>: Modbus implementation where all resources starts with offset 1.</li> <li>• <b>Generic Modbus (0-based)</b>: Modbus implementation where all resources starts with offset 0.</li> <li>• <b>Enron Modbus (1-based)</b>: Extends Modicon Modbus implementation with 32 bit registers memory area.</li> <li>• <b>Enron Modbus (0-base)</b>: Extends Generic Modbus implementation with 32 bit registers memory area.</li> </ul> <p> Note: The address range used in the Modbus frames is always between 0 and 65535 for the Holding Registers and between 0 and 65535 for Coils.</p>
<b>PLC Network</b>	<p>IP address for all controllers in multiple connections. <b>PLC Network</b> must be selected to enable multiple connections.</p>

Element	Description						
	<div style="border: 1px solid gray; padding: 5px;"> <div style="display: flex; justify-content: space-between; align-items: center;"> <div style="border-bottom: 1px solid gray; padding-bottom: 5px;">Modbus TCP</div> <div style="border-bottom: 1px solid gray; padding-bottom: 5px;">Modbus TCP <span style="float: right;">✕</span></div> </div> <div style="padding: 5px;"> <div style="display: flex; justify-content: space-between; align-items: center;"> <input checked="" type="checkbox"/> PLC Network                     <div style="text-align: right;">OK</div> </div> <div style="display: flex; justify-content: space-between; align-items: center;"> <div style="width: 60%;"> <p>Alias <input style="width: 100%;" type="text"/></p> <p>IP address <input style="width: 100%;" type="text" value="0.0.0.1"/></p> <p>Port <input style="width: 100%;" type="text" value="502"/></p> <p><input type="checkbox"/> use UDP/IP</p> <p><input type="checkbox"/> Encapsulated RTU</p> <p>Timeout (ms) <input style="width: 100%;" type="text" value="2000"/></p> <p>Server Busy Timeout <input style="width: 100%;" type="text"/></p> <p>Busy Retry Time (ms) <input style="width: 100%;" type="text"/></p> <p>Modbus ID <input style="width: 100%;" type="text" value="1"/></p> <p>Max read block <input style="width: 100%;" type="text" value="250"/></p> <p>Max read bit block <input style="width: 100%;" type="text" value="2000"/></p> <p>Write Holding Register <input style="width: 100%;" type="text" value="16"/></p> <p>Write Coils <input style="width: 100%;" type="text" value="15"/></p> <p>PLC Models</p> <div style="border: 1px solid gray; padding: 2px;"> <ul style="list-style-type: none"> <li style="background-color: #0070C0; color: white; padding: 2px;">Modicon Modbus(1-based)</li> <li style="padding: 2px;">Generic Modbus(0-based)</li> <li style="padding: 2px;">Enron Modbus(1-based) with 32bit registers</li> <li style="padding: 2px;">Enron Modbus(0-based) with 32bit registers</li> <li style="padding: 2px;">EPSON Robot</li> </ul> </div> </div> <div style="width: 35%; text-align: right;"> <div style="border: 1px solid gray; padding: 2px; margin-bottom: 5px;"> <ul style="list-style-type: none"> <li style="background-color: #0070C0; color: white; padding: 2px;">Modicon Modbus(1-based)</li> <li style="padding: 2px;">Generic Modbus(0-based)</li> <li style="padding: 2px;">Enron Modbus(1-based) with 32bit registers</li> <li style="padding: 2px;">Enron Modbus(0-based) with 32bit registers</li> <li style="padding: 2px;">EPSON Robot</li> </ul> </div> <div style="display: flex; justify-content: space-between; align-items: center;"> <span>Slaves</span> <span>Add</span> <span>Delete</span> <span>Modify</span> </div> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 30%;">Slave Id</th> <th style="width: 30%;">Model</th> <th style="width: 30%;">Alias</th> </tr> </thead> <tbody> <tr> <td> </td> <td> </td> <td> </td> </tr> </tbody> </table> </div> </div> </div> </div>	Slave Id	Model	Alias			
Slave Id	Model	Alias					


## Tag Editor Settings

Path: **ProjectView** > **Config** > double-click **Tags**

1. To add a tag, click **+**: a new line is added.
2. Select **Modbus TCP** from the **Driver** list: tag definition dialog is displayed.

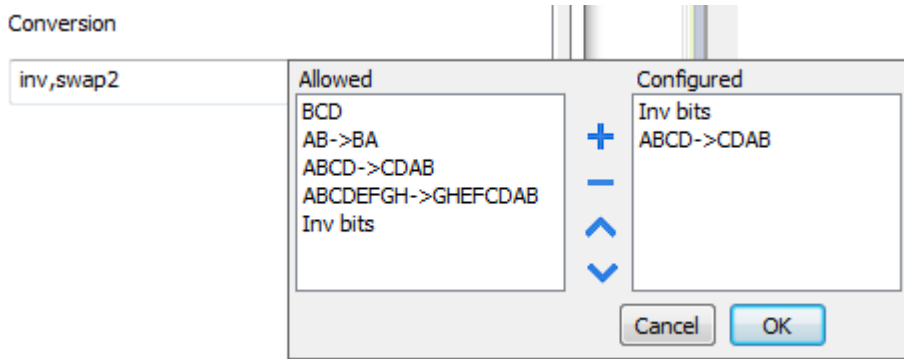


Element	Description	
<b>Memory Type</b>	Modbus resource where tag is located.	
	<b>Memory Type</b>	<b>Modbus Resource</b>
	<b>Coil Status</b>	Coils
	<b>Input Status</b>	Discrete Input
	<b>Input Registers</b>	Input Registers
	<b>Holding registers</b>	Holding Registers
	<b>32 bit Registers</b>	32 bit registers memory area. Available only for <b>Enron Modbus</b> PLC Models
	<b>Node Override IP</b>	protocol parameter (see <b>Special Data Types</b> for mode details)
	<b>Node Override Port</b>	
	<b>Node Override ID</b>	
<b>Modicon Mode</b>		
<b>Offset</b>	Offset address where tag is located. Offset addresses are six digits composed by one digit data type prefix + five digits resource address.	

Element	Description			
	<b>Memory Type</b>	<b>Studio Offset range</b>	<b>Modicon Offset range</b>	<b>Generic Modbus Offset range</b>
	<b>Coil Status</b>	0 – 65535	1 – 65536	0 – 65535
	<b>Input Status</b>	100000 – 165535		
	<b>Input Registers</b>	300000 – 365535		
	<b>Holding Registers</b>	400000 – 465535		
	<b>32 bit Registers</b>	0 – 65535		
<b>SubIndex</b>	This allows resource offset selection within the register.			
<b>Data Type</b>	<b>Data Type</b>	<b>Memory Space</b>	<b>Limits</b>	
	<b>boolean</b>	1-bit data	0 ... 1	
	<b>byte</b>	8-bit data	-128 ... 127	
	<b>short</b>	16-bit data	-32768 ... 32767	
	<b>int</b>	32-bit data	-2.1e9 ... 2.1e9	
	<b>int64</b>	64-bit data	-9.2e18 ... 9.2e18	
	<b>unsignedByte</b>	8-bit data	0 ... 255	
	<b>unsignedShort</b>	16-bit data	0 ... 65535	
	<b>unsignedInt</b>	32-bit data	0 ... 4.2e9	
	<b>uint64</b>	64-bit data	0 ... 1.8e19	
	<b>float</b>	IEEE single-precision 32-bit floating point type	1.17e-38 ... 3.4e38	
	<b>double</b>	IEEE double-precision 64-bit floating point type	2.2e-308 ... 1.79e308	
	<b>string</b>	Array of elements containing character code defined by selected encoding		
	<b>binary</b>	Arbitrary binary data		
	 Note: to define arrays, select one of Data Type format followed by square brackets like “byte []”, “short[]”...			
<b>Arraysize</b>	<ul style="list-style-type: none"> <li>• In case of array tag, this property represents the number of array elements.</li> <li>• In case of string tag, this property represents the maximum number of bytes available in the</li> </ul>			

Element	Description
	<p>string tag.</p> <p>Note: number of bytes corresponds to number of string characters if Encoding property is set to UTF-8 or Latin1 in Tag Editor.</p> <p>If Encoding property is set to UCS-2BE, UCS-2LE, UTF-16BE or UTF-16LE one character requires 2 bytes.</p>

**Conversion** Conversion to be applied to the tag.



Depending on data type selected, the list **Allowed** shows one or more conversion types.

Value	Description
<b>Inv bits</b>	<p><b>inv</b>: Invert all the bits of the tag.</p> <p><i>Example:</i>            1001 → 0110 (in binary format)            9 → 6 (in decimal format)</p>
<b>Negate</b>	<p><b>neg</b>: Set the opposite of tag value.</p> <p><i>Example:</i>            25.36 → -25.36</p>
<b>AB -&gt; BA</b>	<p><b>swapnibbles</b>: Swap nibbles in a byte.</p> <p><i>Example:</i>            15D4 → 514D (in hexadecimal format)            5588 → 20813 (in decimal format)</p>
<b>ABCD -&gt; CDAB</b>	<p><b>swap2</b>: Swap bytes in a word.</p> <p><i>Example:</i>            9ACC → CC9A (in hexadecimal format)            39628 → 52378 (in decimal format)</p>
<b>ABCDEFGH -&gt; GHEFCDAB</b>	<p><b>swap4</b>: Swap bytes in a double word.</p> <p><i>Example:</i>            32FCFF54 → 54FFFC32 (in hexadecimal format)</p>



Element	Description	
	<b>Value</b>	<b>Description</b>
		855441236 → 1426062386 (in decimal format)
	<b>ABC...NOP -&gt; OPM...DAB</b>	<b>swap8</b> : Swap bytes in a long word. Example: 142.366 → -893553517.588905 (in decimal format) 0 10000000110 0001110010111011011001000101101000011100101011000001 → 1 10000011100 1010101000010100010110110110110010110110000100111101 (in binary format)
	<b>BCD</b>	<b>bcd</b> : Separate byte in two nibbles, read them as decimal (from 0 to 9) Example: 23 → 17 (in decimal format) 0001 0111 = 23 0001 = 1 (first nibble) 0111 = 7 (second nibble)

Select conversion and click +. The selected item will be added to list **Configured**.

If more conversions are configured, they will be applied in order (from top to bottom of list **Configured**).

Use the arrow buttons to order the configured conversions.

## Node Override IP

The protocol provides the special data type Node Override IP which allows you to change the IP address of the target controller at runtime.

This memory type is an array of 4 unsigned bytes, one per each byte of the IP address.

The Node Override IP is initialized with the value of the controller IP specified in the project at programming time.

Node Override IP	PLC operation
<b>0.0.0.0</b>	Communication with the controller is stopped, no request frames are generated anymore.
<b>Different from 0.0.0.0</b>	It is interpreted as node IP override and the target IP address is replaced runtime with the new value.

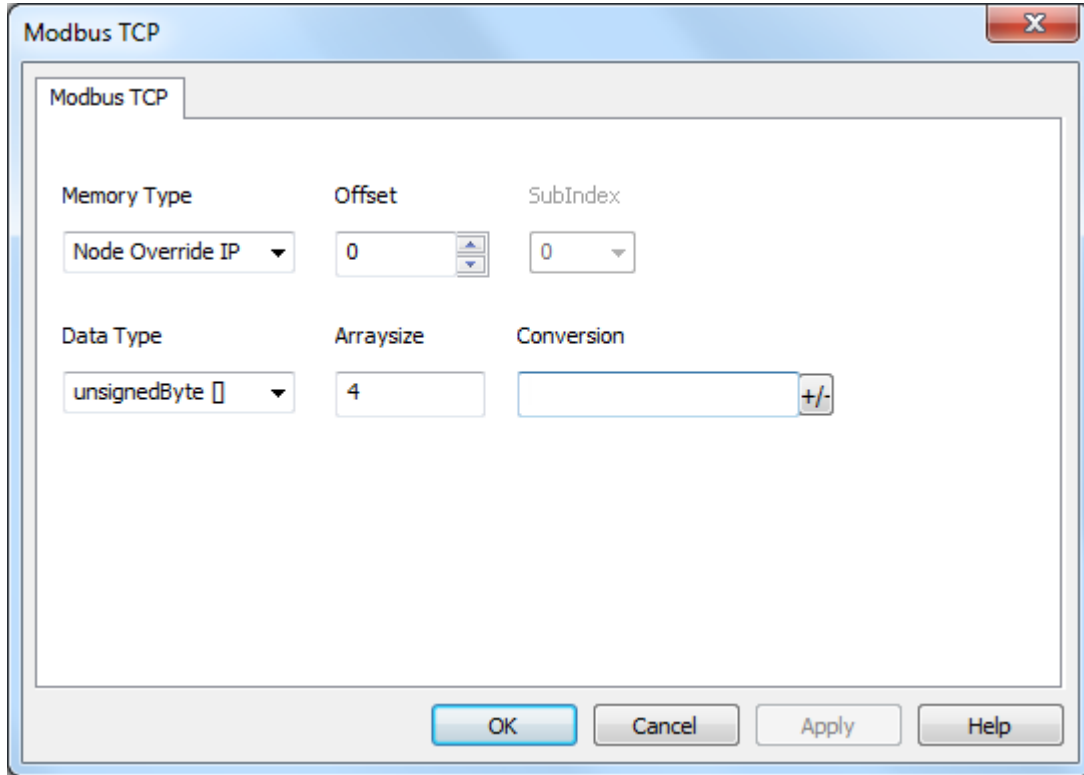
If the HMI device is connected to a network with more than one controller node, each node has its own Node Override IP variable.



Note: Node Override IP values assigned at runtime are retained through power cycles.

## Hostname DNS or mDNS

In addition to the array of bytes, string memory type can be selected to be able use the DNS or mDNS hostname as an alternative to the IP Address.



## Node Override Port

The protocol provides the special data type Node Override Port which allows you to change the network Port of the target controller at runtime.

This memory type is unsigned short.

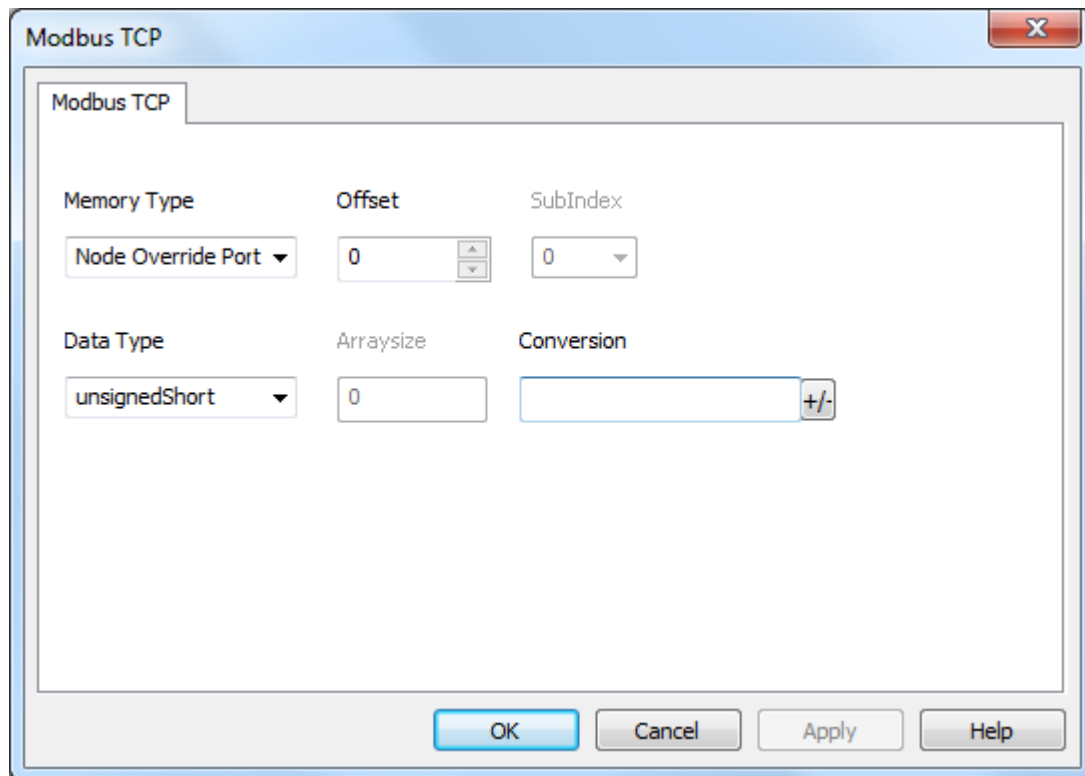
Node Override Port is initialized with the value of the controller Port specified in the project at programming time.

Node Override Port	Modbus operation
0	Communication with the controller is stopped, no request frames are generated anymore.
Different from 0	It is interpreted as the value of the new port and is replaced for runtime operation.

If the HMI device is connected to a network with more than one controller node, each node has its own Node Override Port variable.



Note: Node Override Port values assigned at runtime are retained through power cycles.



## Node Override ID

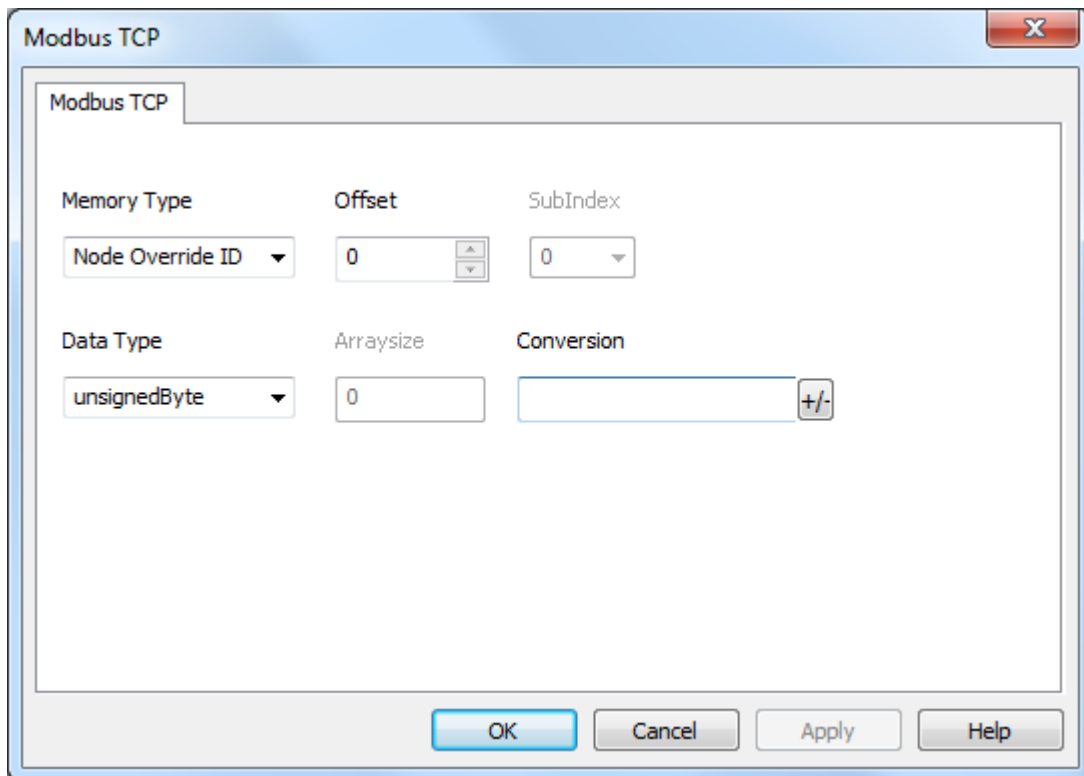
The protocol provides the special data type Node Override ID which allows you to change the node ID of the slave at runtime. This memory type is an unsigned byte.

The node Override ID is initialized with the value of the node ID specified in the project at programming time.

Node Override ID	Modbus operation
0	Communication with the controller is stopped. In case of write operation, the request will be transmitted without waiting for a reply.
1 to 254	It is interpreted as the value of the new node ID and is replaced for runtime operation.
255	Communication with the controller is stopped; no request messages are generated.



Note: Node Override ID value assigned at runtime is retained through power cycles.



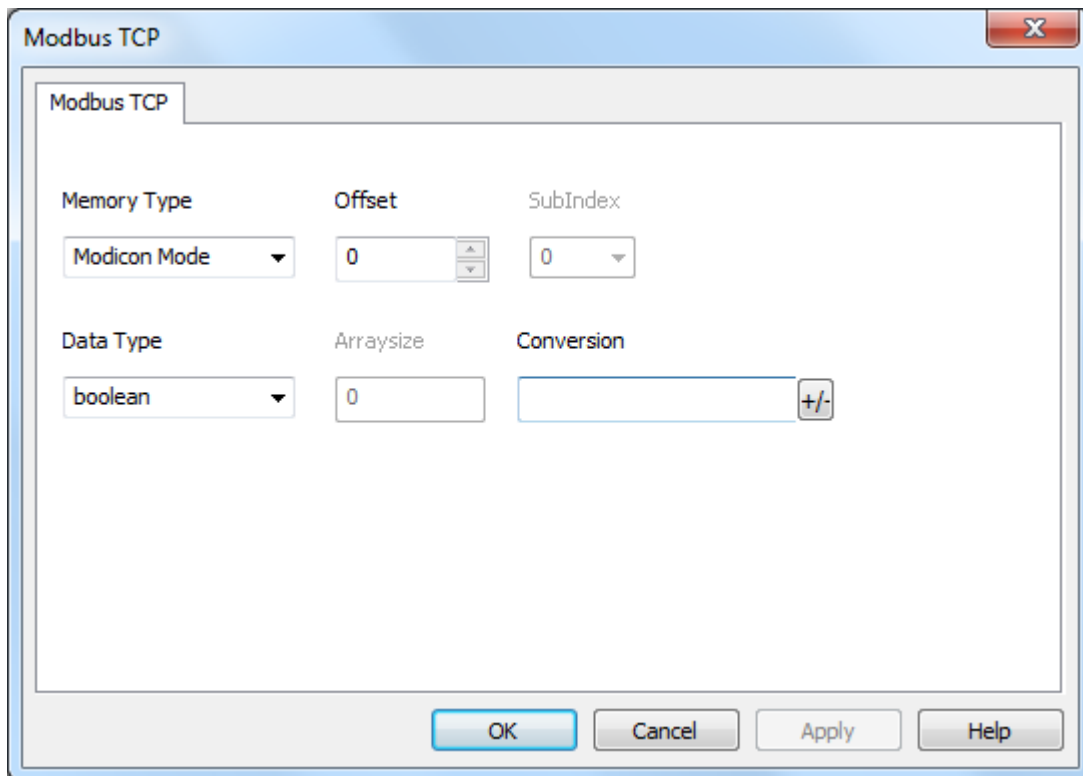
## Modicon Mode

The protocol provide a special data type that can be used to override the Modicon Mode parameter at runtime.

Modicon Mode	Description
0	Generic Modbus (0-based). Register indexes start from 0.
1	Modicon Modbus (1-based). Register indexes start from 1.

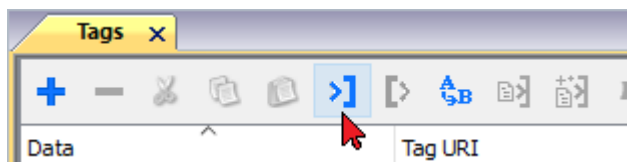


Note: Modicon Mode parameter value assigned at runtime is retained through power cycles.

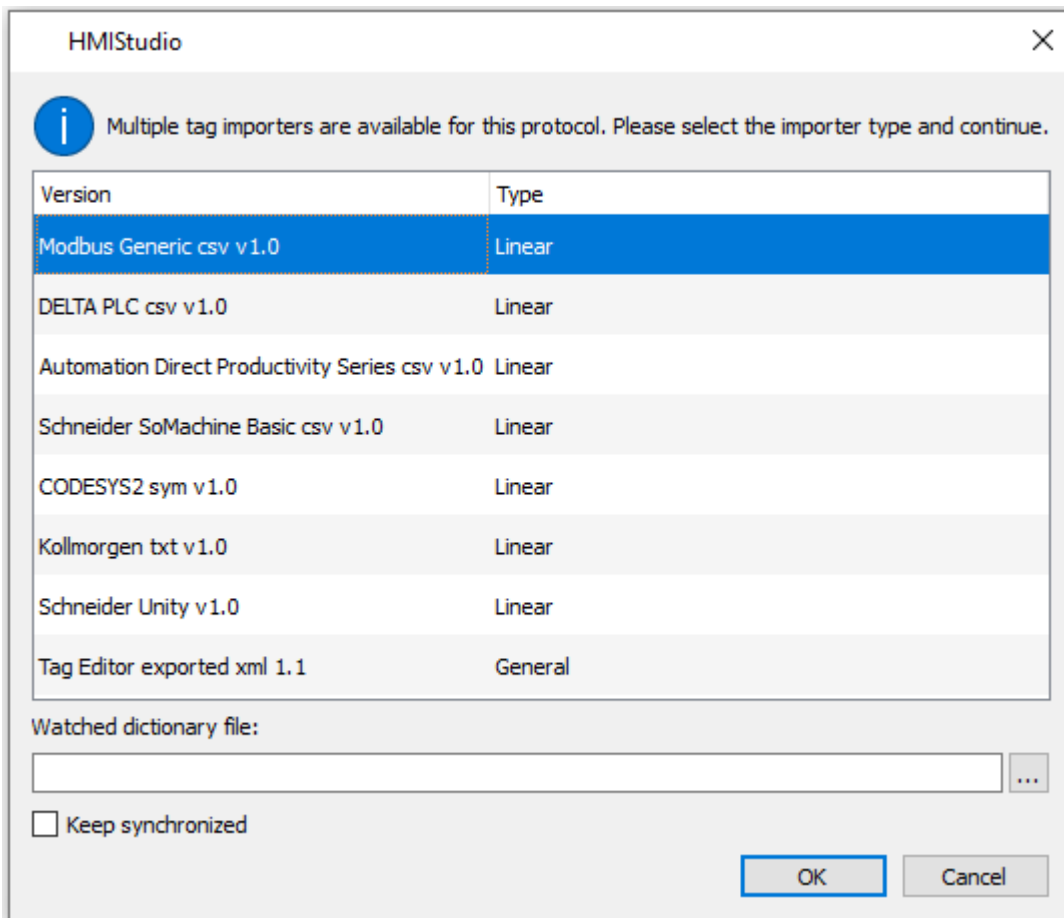


## Tag Import

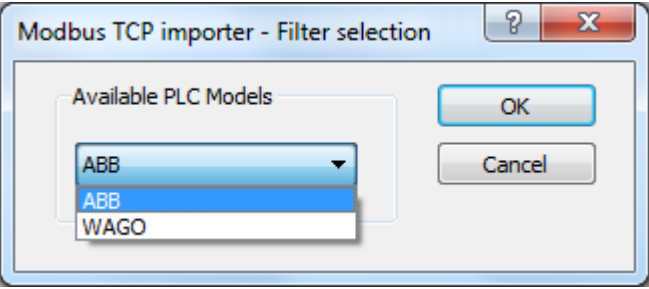

Select the driver in Tag Editor and click on the **Import Tags** button to start the importer.



The following dialog shows which importer type can be selected.

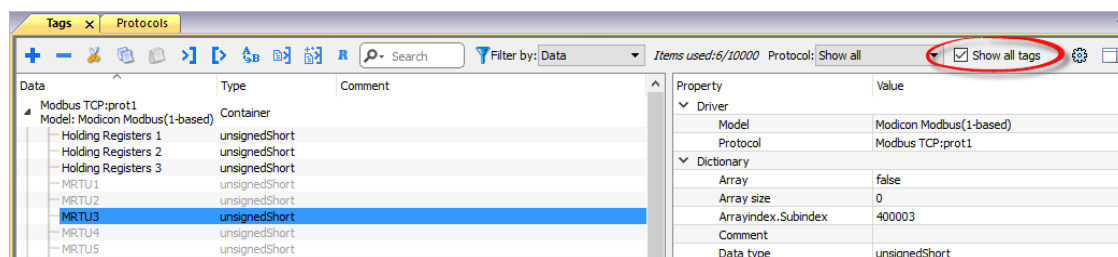





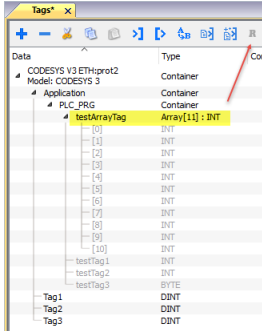
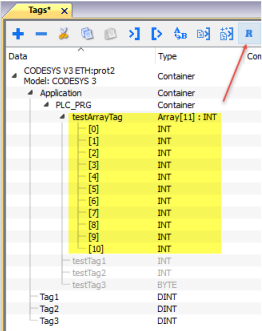
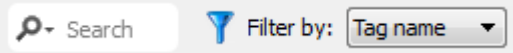
Type	Description
<b>Modbus Generic csv v1.0</b> Linear	Requires a <b>.csv</b> file. All variables will be displayed at the same level.
<b>DELTA PLC csv v1.0</b>	Requires a <b>.csv</b> file. All variables will be displayed at the same level.
<b>CODESYS2 sym v1.0</b> Linear	Requires a <b>.sym</b> file. All variables will be displayed at the same level. After selecting the <b>.sym</b> file, the following dialog will appear for PLC model selection.

Type	Description
	
<p><b>Kollmorgen txt v1.0</b> <b>Linear</b></p>	<p>Requires a <b>.txt</b> file.</p> <p>All variables will be displayed at the same level.</p>
<p><b>Schneider Unity v1.0</b> <b>Linear</b></p>	<p>Requires a <b>.uny</b> file.</p> <p>The file containing symbols must be exported in <b>.txt</b> format and later renamed as <b>.uny</b>. The importer considers only variables located at fixed address and disregards arrays of strings. All other arrays, except for boolean type, are expanded.</p>
<p><b>Tag Editor exported xml</b></p>	<p>Select this importer to read a generic XML file exported from Tag Editor by appropriate button.</p> 

Once the importer has been selected, locate the symbol file and click **Open**.

The tags available within the Dictionary but not imported into the project are gray and are visible only when the "Show all tags" check box is selected.



Toolbar item	Description
	<p><b>Import Tag(s).</b></p> <p>Select tags to be imported and click on this icon to add tags from tag dictionary to the project</p>
	<p><b>Update Tag(s).</b></p> <p>Click on this icon to update the tags in the project, due a new dictionary import.</p>
	<p>Check this box to import all sub-elements of a tag.</p> <p>Example of both checked and unchecked result:</p> <div style="display: flex; justify-content: space-around;">   </div>
	<p>Searches tags in the dictionary basing on filter combo-box item selected.</p>

## Modbus Generic csv file structure

This protocol supports the import of tag information when provided in **.csv** format according to the following format:

`NodeID, TagName, MemoryType, Address, DataFormat, ..., [Comment]`



Note: Fields in brackets are optional as well as fields between Data Format and Comment.

Field	Description
<b>NodeID</b>	Node the tag belongs to
<b>TagName</b>	Tag description
<b>MemoryType</b>	<ul style="list-style-type: none"> <li>• OUTP</li> <li>• INP</li> <li>• IREG</li> <li>• HREG</li> </ul>
<b>Address</b>	Offset compatible with Modbus notation



Field	Description
DataFormat	Data type in internal notation. See "Programming concepts" section in the main manual.
Comment	Optional additional description.

### Tag file example

Example of .csv line:

```
2, Holding Register 1, HREG, 400001, unsignedShort,
```



Note: This line has no comment. When the Comment is missing, the comma as a terminator character is mandatory.

Example of .csv file:

```
1, Holding Register 2 TagName, HREG, 400002, unsignedShort, comment for this register
```

```
1, HRegTag 400011_15 TagName, HREG, 400011.15, boolean, comment for this register
```

```
2, InpRegTag 300501 TagName, IREG, 300501, unsignedShort, comment for this register
```

```
1, InpRegTag 301999_8 TagName, IREG, 301999.8, boolean, comment for this register
```

```
27, OutputTag 999 TagName, OUP, 999, boolean, comment for this register
```

```
11, InputTag 100101 TagName, INP, 100101, boolean, comment for this register
```

```
2, HRegTag 409999 TagName, HREG, 409999, unsignedShort, comment for this register
```

## Communication status

Current communication status can be displayed using system variables. See "System Variables" section in the main manual.

Codes supported for this communication driver:

Error	Cause	Action
No response	No reply within the specified timeout.	Check if the controller is connected and properly configured to get network access.
Incorrect node address in response	The device received a response with an invalid node address from the controller .	-

---

<b>Error</b>	<b>Cause</b>	<b>Action</b>
<b>The received message too short</b>	The device received a response with an invalid format from the controller .	-
<b>Incorrect writing data acknowledge</b>	The controller did not accept a write request.	Check if project data is consistent with the controller resources.

# Modbus TCP Server

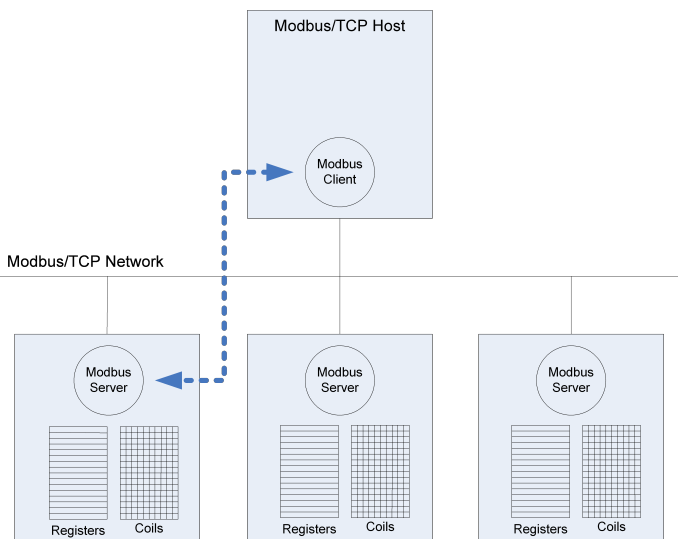
Modbus TCP Server communication driver allows connecting the HMI device as a server in a Modbus TCP network. It is possible for Modbus TCP clients to connect then to multiple HMI panels acting as servers. Standard Modbus TCP messages are used for information exchange.

This approach allows connecting HMI devices to SCADA systems through the universally supported Modbus TCP communication protocol.

## Principle of operation

This communication driver implements a Modbus TCP Server unit in HMI device. A subset of the complete range of Modbus function codes is supported. The available function codes allow data transfer between clients on the TCP network and the server. The HMI device acts as a server in the network. It can exchange data with up to 32 clients. This means that up to 32 clients can be connected to the HMI device at the same time. If all the 32 available connections are in use, any further attempt to connect by a client will be refused by the system.

The following diagram shows the system architecture.



The device simulates the communication interface of a PLC: Coils and Registers data types are respectively boolean and 16 bit integers.

The device always access data in its internal memory. Data can be transferred to and from the Modbus Client only on the initiative of the client itself.

## Implementation details

This Modbus TCP Server implementation supports only a subset of the Modbus standard function codes.

Code	Function	Description
01	Read Coil Status	Reads multiple bits in the device Coil area.
02	Read Input Status	Reads multiple bits in the device Coil area.
03	Read Holding Registers	Read multiple device Registers.

Code	Function	Description
04	Read Input Registers	Read multiple device Registers.
05	Force Single Coil	Forces a single device Coil to either ON or OFF.
06	Preset Single Register	Presets a value in a device Register.
15	Force Multiple Coils	Forces multiple device Coils to either ON or OFF.
16	Preset Multiple Registers	Presets value in multiple device Registers.
23	Read Write Multiple Registers	Read & presets values in multiple device Registers



Note: For both PLC models the Read Coil Status and Read Input Status function codes both access the same Coil memory area in the HMI device memory. The Read Holding Registers and Read Input Registers function codes both access the same Register area in the HMI device memory.

## Exception Codes

Code	Description
01	<b>Illegal Function.</b> the function code received in the query is not supported
02	<b>Illegal Data Address.</b> Data Address received in the query exceeds the predefined data range (see <b>Tag Editor Settings</b> for detailed ranges of all types).
03	<b>Illegal Data Value.</b> A sub function other than 00 is specified in Loopback Diagnostic Test (Code 08).

## Protocol Editor Settings

### Adding a protocol

To configure the protocol:

1. In **Config** node double-click **Protocols**.
2. To add a driver, click **+**: a new line is added.
3. Select the protocol from the **PLC** list.

The protocol configuration dialog is displayed.

Modbus TCP Server

PLC Network

Modbus ID:

Port:

Connection Lifetime:

use UDP/IP

Encapsulated RTU

Enron 32bit registers

32bit reg Start:



32bit reg Size:



PLC Models

- Modicon Modbus (1-based)
- Generic Modbus (0-based)
- Enron Modbus (1-based) with 32bit registers
- Enron Modbus (0-based) with 32bit registers

OK

Cancel

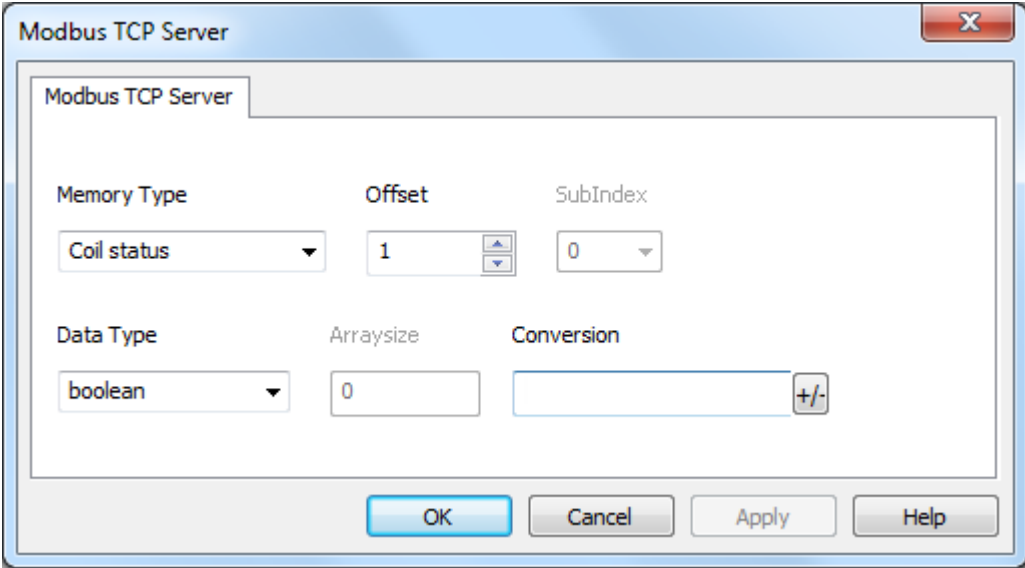
Element	Description
<b>Modbus ID</b>	<p>Modbus node ID of the HMI device. Every Modbus server device in the network must have its own Modbus ID.</p> <p> Note: The valid range admitted is between 1 and 255. Writing the value 0 means node disabled.</p>
<b>Port</b>	<p>Port number used by the Modbus TCP protocol. Default value is <b>502</b>. Set the value accordingly to the port number used by your Modbus TCP Network.</p>
<b>use UDP/IP</b>	<p>If selected, the protocol will use connectionless UDP datagrams.</p>
<b>Encapsulated RTU</b>	<p>If selected, the protocol will use serial RTU protocol over Ethernet instead of Modbus TCP protocol, independently from TCP or UDP usage.</p>
<b>Enron 32bit registers</b>	<p>If selected, allows to define the first register address and the number of registers for 32 bit registers memory area.</p> <p> Note: 32 bit registers are available only for <b>Enron Modbus</b> PLC Models.</p>
	<p>32 bit registries memory area definition.</p>

Element	Description
<b>32bit reg Start</b>	<b>Start</b> value represents the first register address.
<b>32bit reg Size</b>	<p><b>Size</b> value represents the number of registries.</p> <p> Note: A request to one of the registries inside this area gives a 4 byte answer.</p>
<b>PLC Models</b>	<p>Allows to select between different PLC models:</p> <ul style="list-style-type: none"> <li>• <b>Modicon Modbus (1-based)</b>: Modbus implementation where all resources starts with offset 1.</li> <li>• <b>Generic Modbus (0-based)</b>: Modbus implementation where all resources starts with offset 0.</li> <li>• <b>Enron Modbus (1-based)</b>: Extends Modicon Modbus implementation with 32 bit registers memory area.</li> <li>• <b>Enron Modbus (0-base)</b>: Extends Generic Modbus implementation with 32 bit registers memory area.</li> </ul> <p> Note: The address range used in the Modbus frames is always between 0 and 65535 for the Holding Registers and between 0 and 65535 for Coils.</p>

## Tag Editor Settings

Path: **ProjectView** > **Config** > double-click **Tags**

1. To add a tag, click **+**: a new line is added.
2. Select **Modbus TCP Server** from the protocol list: tag definition dialog is displayed.



**Modbus TCP Server**


Modbus TCP Server

Memory Type: Coil status    Offset: 1    SubIndex: 0

Data Type: boolean    Arraysize: 0    Conversion: +/-

OK    Cancel    Apply    Help

Element	Description																		
<b>Memory Type</b>	Modbus resource where tag is located.																		
	<table border="1"> <thead> <tr> <th>Memory Type</th> <th>Modbus Resource</th> </tr> </thead> <tbody> <tr> <td><b>Coil Status</b></td> <td>Coils</td> </tr> <tr> <td><b>Input Status</b></td> <td>Discrete Input</td> </tr> <tr> <td><b>Input Registers</b></td> <td>Input Registers</td> </tr> <tr> <td><b>Holding Registers</b></td> <td>Holding Registers</td> </tr> <tr> <td><b>32 bit Registers</b></td> <td>32 bit registers memory area. Available only for <b>Enron Modbus PLC Models</b>.</td> </tr> <tr> <td><b>Modicon Mode</b></td> <td>protocol parameter (see <b>Special Data Types</b> for mode details)</td> </tr> </tbody> </table>	Memory Type	Modbus Resource	<b>Coil Status</b>	Coils	<b>Input Status</b>	Discrete Input	<b>Input Registers</b>	Input Registers	<b>Holding Registers</b>	Holding Registers	<b>32 bit Registers</b>	32 bit registers memory area. Available only for <b>Enron Modbus PLC Models</b> .	<b>Modicon Mode</b>	protocol parameter (see <b>Special Data Types</b> for mode details)				
	Memory Type	Modbus Resource																	
	<b>Coil Status</b>	Coils																	
	<b>Input Status</b>	Discrete Input																	
	<b>Input Registers</b>	Input Registers																	
	<b>Holding Registers</b>	Holding Registers																	
	<b>32 bit Registers</b>	32 bit registers memory area. Available only for <b>Enron Modbus PLC Models</b> .																	
<b>Modicon Mode</b>	protocol parameter (see <b>Special Data Types</b> for mode details)																		
<b>Coil Status</b>	Coils																		
<b>Input Status</b>	Discrete Input																		
<b>Input Registers</b>	Input Registers																		
<b>Holding Registers</b>	Holding Registers																		
<b>32 bit Registers</b>	32 bit registers memory area. Available only for <b>Enron Modbus PLC Models</b> .																		
<b>Modicon Mode</b>	protocol parameter (see <b>Special Data Types</b> for mode details)																		
<b>Offset</b>	Offset address where tag is located.																		
	Offset addresses are six digits composed by one digit data type prefix + five digits resource address.																		
	<table border="1"> <thead> <tr> <th>Memory Type</th> <th>Studio Offset range</th> <th>Modicon Offset range</th> <th>Generic Modbus Offset range</th> </tr> </thead> <tbody> <tr> <td><b>Coil Status</b></td> <td>0 – 65535</td> <td rowspan="5">1 – 65536</td> <td rowspan="5">0 – 65535</td> </tr> <tr> <td><b>Input Status</b></td> <td>100000 – 165535</td> </tr> <tr> <td><b>Input Registers</b></td> <td>300000 – 365535</td> </tr> <tr> <td><b>Holding Registers</b></td> <td>400000 – 465535</td> </tr> <tr> <td><b>32 bit Registers</b></td> <td>0 – 65535</td> </tr> </tbody> </table>	Memory Type	Studio Offset range	Modicon Offset range	Generic Modbus Offset range	<b>Coil Status</b>	0 – 65535	1 – 65536	0 – 65535	<b>Input Status</b>	100000 – 165535	<b>Input Registers</b>	300000 – 365535	<b>Holding Registers</b>	400000 – 465535	<b>32 bit Registers</b>	0 – 65535		
	Memory Type	Studio Offset range	Modicon Offset range	Generic Modbus Offset range															
	<b>Coil Status</b>	0 – 65535	1 – 65536	0 – 65535															
	<b>Input Status</b>	100000 – 165535																	
	<b>Input Registers</b>	300000 – 365535																	
<b>Holding Registers</b>	400000 – 465535																		
<b>32 bit Registers</b>	0 – 65535																		
<b>Coil Status</b>	0 – 65535	1 – 65536	0 – 65535																
<b>Input Status</b>	100000 – 165535																		
<b>Input Registers</b>	300000 – 365535																		
<b>Holding Registers</b>	400000 – 465535																		
<b>32 bit Registers</b>	0 – 65535																		
<b>SubIndex</b>	This allows resource offset selection within the register.																		
<b>Data type</b>	<table border="1"> <thead> <tr> <th>Data Type</th> <th>Memory Space</th> <th>Limits</th> </tr> </thead> <tbody> <tr> <td><b>boolean</b></td> <td>1-bit data</td> <td>0 ... 1</td> </tr> <tr> <td><b>byte</b></td> <td>8-bit data</td> <td>-128 ... 127</td> </tr> <tr> <td><b>short</b></td> <td>16-bit data</td> <td>-32768 ... 32767</td> </tr> <tr> <td><b>int</b></td> <td>32-bit data</td> <td>-2.1e9 ... 2.1e9</td> </tr> <tr> <td><b>int64</b></td> <td>64-bit data</td> <td>-9.2e18 ... 9.2e18</td> </tr> </tbody> </table>	Data Type	Memory Space	Limits	<b>boolean</b>	1-bit data	0 ... 1	<b>byte</b>	8-bit data	-128 ... 127	<b>short</b>	16-bit data	-32768 ... 32767	<b>int</b>	32-bit data	-2.1e9 ... 2.1e9	<b>int64</b>	64-bit data	-9.2e18 ... 9.2e18
	Data Type	Memory Space	Limits																
	<b>boolean</b>	1-bit data	0 ... 1																
	<b>byte</b>	8-bit data	-128 ... 127																
	<b>short</b>	16-bit data	-32768 ... 32767																
	<b>int</b>	32-bit data	-2.1e9 ... 2.1e9																
<b>int64</b>	64-bit data	-9.2e18 ... 9.2e18																	
<b>Data Type</b>	<b>Memory Space</b>	<b>Limits</b>																	
<b>boolean</b>	1-bit data	0 ... 1																	
<b>byte</b>	8-bit data	-128 ... 127																	
<b>short</b>	16-bit data	-32768 ... 32767																	
<b>int</b>	32-bit data	-2.1e9 ... 2.1e9																	
<b>int64</b>	64-bit data	-9.2e18 ... 9.2e18																	

Element	Description		
	<b>Data Type</b>	<b>Memory Space</b>	<b>Limits</b>
	<b>unsignedByte</b>	8-bit data	0 ... 255
	<b>unsignedShort</b>	16-bit data	0 ... 65535
	<b>unsignedInt</b>	32-bit data	0 ... 4.2e9
	<b>uint64</b>	64-bit data	0 ... 1.8e19
	<b>float</b>	IEEE single-precision 32-bit floating point type	1.17e-38 ... 3.4e38
	<b>double</b>	IEEE double-precision 64-bit floating point type	2.2e-308 ... 1.79e308
	<b>string</b>	Array of elements containing character code defined by selected encoding	
	<b>binary</b>	Arbitrary binary data	
		Note: to define arrays, select one of Data Type format followed by square brackets like "byte[]", "short[]"...	

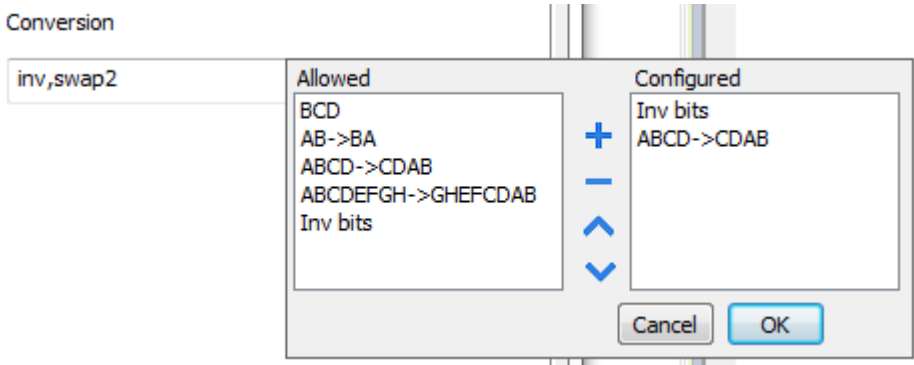
**Arraysiz**

- In case of array tag, this property represents the number of array elements.
- In case of string tag, this property represents the maximum number of bytes available in the string tag.

Note: number of bytes corresponds to number of string characters if Encoding property is set to UTF-8 or Latin1 in Tag Editor.  
If Encoding property is set to UCS-2BE, UCS-2LE, UTF-16BE or UTF-16LE one character requires 2 bytes.

**Conversion**

Conversion to be applied to the tag.



Depending on data type selected, the list **Allowed** shows one or more conversion types.



Element	Description	
	<b>Value</b>	<b>Description</b>
	<b>Inv bits</b>	<p><b>inv:</b> Invert all the bits of the tag.</p> <p><i>Example:</i>            1001 → 0110 (in binary format)            9 → 6 (in decimal format)</p>
	<b>Negate</b>	<p><b>neg:</b> Set the opposite of tag value.</p> <p><i>Example:</i>            25.36 → -25.36</p>
	<b>AB -&gt; BA</b>	<p><b>swapnibbles:</b> Swap nibbles in a byte.</p> <p><i>Example:</i>            15D4 → 514D (in hexadecimal format)            5588 → 20813 (in decimal format)</p>
	<b>ABCD -&gt; CDAB</b>	<p><b>swap2:</b> Swap bytes in a word.</p> <p><i>Example:</i>            9ACC → CC9A (in hexadecimal format)            39628 → 52378 (in decimal format)</p>
	<b>ABCDEFGH -&gt; GHEFCDAB</b>	<p><b>swap4:</b> Swap bytes in a double word.</p> <p><i>Example:</i>            32FCFF54 → 54FFFC32 (in hexadecimal format)            855441236 → 1426062386 (in decimal format)</p>
	<b>ABC...NOP -&gt; OPM...DAB</b>	<p><b>swap8:</b> Swap bytes in a long word.</p> <p><i>Example:</i>            142.366 → -893553517.588905 (in decimal format)            0 1000000110            00011100101110110110010001011010000111001010110000            01            →            1 10000011100            10101010000101000101101101101100101101100001001111            01            (in binary format)</p>
	<b>BCD</b>	<p><b>bcd:</b> Separate byte in two nibbles, read them as decimal (from 0 to 9)</p> <p><i>Example:</i>            23 → 17 (in decimal format)            0001 0111 = 23            0001 = 1 (first nibble)            0111 = 7 (second nibble)</p>

Element	Description
	Select conversion and click +. The selected item will be added to list <b>Configured</b> .
	If more conversions are configured, they will be applied in order (from top to bottom of list <b>Configured</b> ).
	Use the arrow buttons to order the configured conversions.

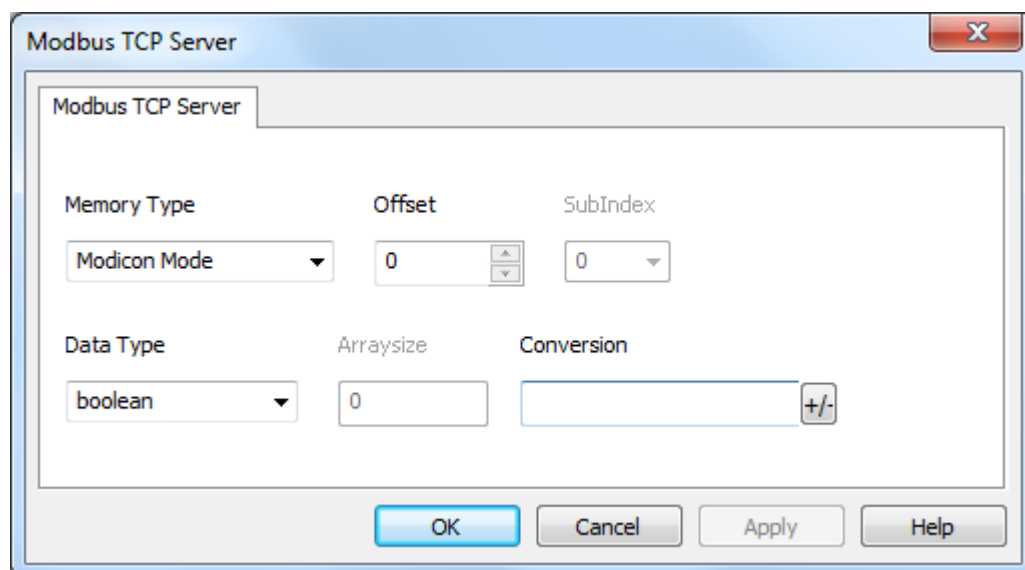
## Modicon Mode

The protocol provide a special data type that can be used to override the Modicon Mode parameter at runtime.

Modicon Mode	Description
0	Generic Modbus (0-based). Register indexes start from 0.
1	Modicon Modbus (1-based). Register indexes start from 1.

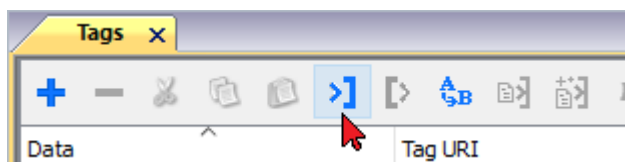


Note: Modicon Mode parameter value assigned at runtime is retained through power cycles.

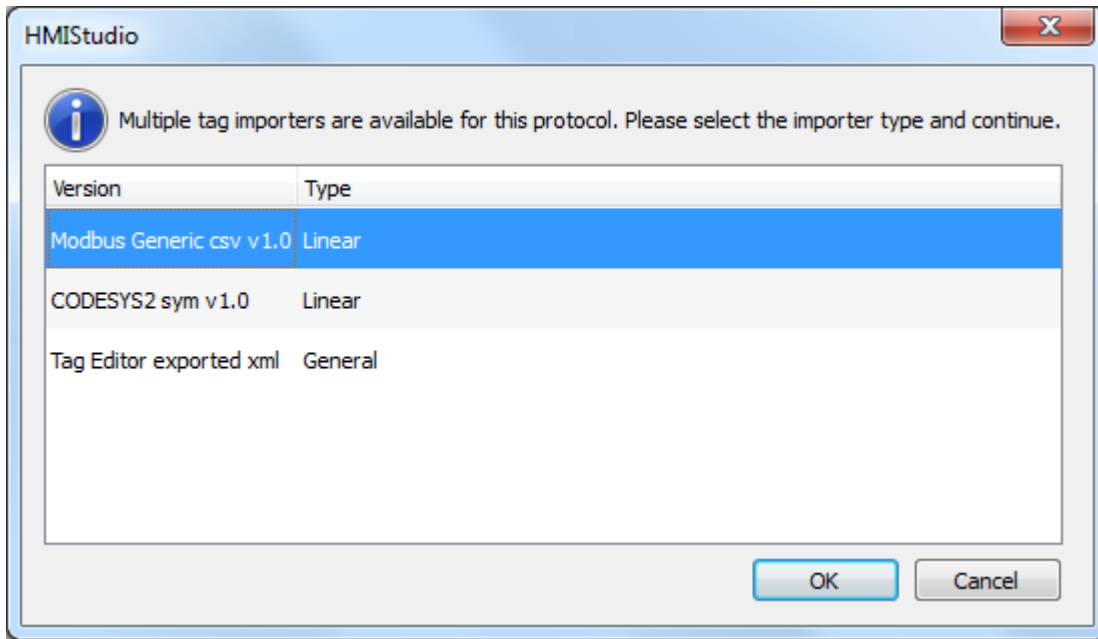


## Tag Import

Select the driver in Tag Editor and click on the **Import Tags** button to start the importer.



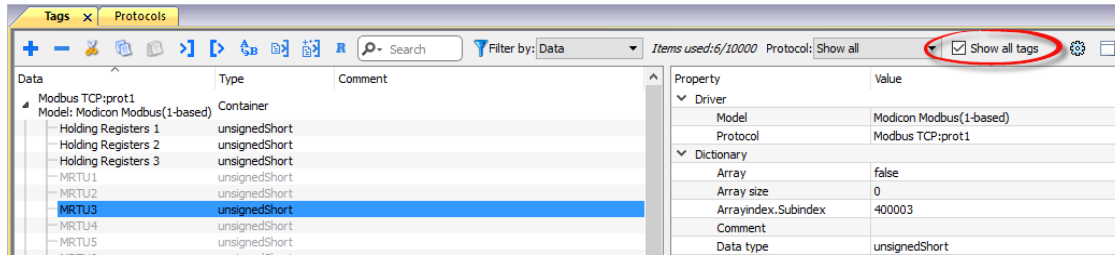
The following dialog shows which importer type can be selected.




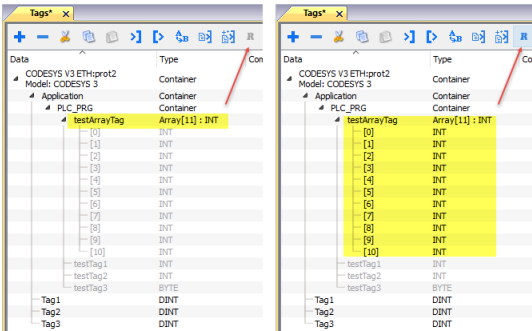
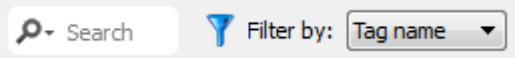


Importer	Description
<b>Modbus Generic csv v1.0</b> Linear	Requires a <b>.csv</b> file. All variables will be displayed at the same level.
<b>CODESYS2 sym v1.0</b> Linear	Requires a <b>.sym</b> file. All variables will be displayed at the same level. After selecting the <b>.sym</b> file, the following dialog will appear for PLC model selection. <div data-bbox="387 1317 1042 1608" style="border: 1px solid gray; padding: 5px; margin: 10px 0;"> </div>
<b>Tag Editor exported xml</b>	Select this importer to read a generic XML file exported from Tag Editor by appropriate button. <div data-bbox="387 1724 1013 1881" style="border: 1px solid gray; padding: 5px; margin: 10px 0;"> </div>

Once the importer has been selected, locate the symbol file and click **Open**.

The tags available within the Dictionary but not imported into the project are gray and are visible only when the "Show all tags" check box is selected.



Toolbar item	Description
	<p><b>Import Tag(s).</b></p> <p>Select tags to be imported and click on this icon to add tags from tag dictionary to the project</p>
	<p><b>Update Tag(s).</b></p> <p>Click on this icon to update the tags in the project, due a new dictionary import.</p>
	<p>Check this box to import all sub-elements of a tag.</p> <p>Example of both checked and unchecked result:</p> 
	<p>Searches tags in the dictionary basing on filter combo-box item selected.</p>

## Modbus Generic csv file structure

This protocol supports the import of tag information when provided in .csv format according to the following format:

NodeID, TagName, MemoryType, Address, DataFormat, ..., [Comment]



Note: Fields in brackets are optional as well as fields between Data Format and Comment.

Field	Description
<b>NodeID</b>	Node the tag belongs to
<b>TagName</b>	Tag description
<b>MemoryType</b>	<ul style="list-style-type: none"> <li>• OUP</li> <li>• INP</li> <li>• IREG</li> <li>• HREG</li> </ul>
<b>Address</b>	Offset compatible with Modbus notation
<b>DataFormat</b>	Data type in internal notation. See "Programming concepts" section in the main manual.
<b>Comment</b>	Optional additional description.

### Tag file example

Example of .csv line:

```
2, Holding Register 1, HREG, 400001, unsignedShort,
```



Note: This line has no comment. When the Comment is missing, the comma as a terminator character is mandatory.

Example of .csv file:

```
1, Holding Register 2 TagName, HREG, 400002, unsignedShort, comment for this register
```

```
1, HRegTag 400011_15 TagName, HREG, 400011.15, boolean, comment for this register
```

```
2, InpRegTag 300501 TagName, IREG, 300501, unsignedShort, comment for this register
```

```
1, InpRegTag 301999_8 TagName, IREG, 301999.8, boolean, comment for this register
```

```
27, OutputTag 999 TagName, OUP, 999, boolean, comment for this register
```

```
11, InputTag 100101 TagName, INP, 100101, boolean, comment for this register
```

```
2, HRegTag 409999 TagName, HREG, 409999, unsignedShort, comment for this register
```

### Communication status

The HMI device is a server station in the Modbus TCP network. The current implementation of the protocol doesn't report any communication error code apart from standard communication error codes related to the proper driver loading.

See "System Variables" section in the main manual.

# OPC UA Client

The OPC UA Client communication driver has been designed to connect HMI devices to OPC UA servers.

This implementation of the protocol operates as a client only.


OPC UA client can be used for encrypted communication between AC500 V3 and CP600. Communication has to be encrypted in order to protect passwords and other data.

## Protocol Editor Settings

Add (+) a driver in the Protocol editor and select the protocol called “OPC UA Client” from the list of available protocols.

The driver configuration dialog is shown as in the following figure:

The screenshot shows the 'OPC UA Client' configuration dialog box. It features a title bar with a close button (X) and a 'PLC Network' checkbox. The main area contains several configuration fields: 'Alias' (text input), 'Host' (text input with '0.0.0.0'), 'Port' (spin box with '4840'), 'Timeout (ms)' (spin box with '1000'), 'Security Policy' (dropdown menu with 'None'), 'Security Mode' (dropdown menu with 'None'), 'Username' (text input), 'Password' (text input), 'Server Certificate' (text input with a browse button), 'Client Certificate' (text input with a browse button), and 'Client Private Key' (text input with a browse button). Below these are three checked checkboxes: 'Hostname validation', 'App URI validation', and 'Time validation'. At the bottom, there is a 'PLC Models' section with a list box containing 'Default'.

Element	Description
<b>PLC Network</b>	Enable access to multiple networked controllers. For every controller set proper options.
<b>Alias</b>	Name identifying nodes in network configurations. The name will be added as a prefix to each tag name imported for each network node.
<b>IP Address</b>	IP address of the server.
<b>Port</b>	Port number where the server is listening.
<b>Timeout (ms)</b>	Time delay in milliseconds between two retries in case of no response from the server device.
<b>Security Mode</b>	Type of authentication: <ul style="list-style-type: none"> <li>• <b>None</b>: Certificates are not used</li> <li>• <b>Sign</b>: Certificates only used for authentication with server.</li> <li>• <b>SignAndEncrypt</b>: Certificates used for authentication with server and data encryption.</li> </ul>
<b>Security Policy</b>	Encryption level to use (used only when Security Mode is active). <ul style="list-style-type: none"> <li>• Basic256</li> <li>• Basic256Sha256</li> </ul>
<b>Username Password</b>	Authentication with user name and password
<b>Server Certificate</b>	Certificate for OPC UA Server. <div data-bbox="357 1227 427 1301" style="display: inline-block; vertical-align: middle;"></div> Server certificate can be downloaded using tag importer. See " <a href="#">Remote OPC UA Server certificate</a> " on page 894
<b>Client Certificate</b>	Certificate used by OPC UA client. If blank, a certificate is automatically generated.
<b>Client Private Key</b>	Key used by OPC UA client. If blank, a key is automatically generated.
<b>PLC Models</b>	No options available.

## Notes:

- Before choosing security options, be aware that not all security modes might be supported by the OPC UA server. Make sure to use security mode that is supported.
- When working within a private network you do not need to provide devices' certificates because you trust used devices. On a public network, instead, the certificate will give you a guarantee of the identity of devices.

**External Certificate**

ASCII version of the certificate (usually a file with .pem extension) is required.

Edit the certificate files and then copy and paste the full text of your certificate to the certificate fields.

**Step 1:** Remove header and footer lines

-----BEGIN CERTIFICATE-----

```
MIIDNjCCAh4CCQCJtJgjqDDUqjANBgbkqhkiG9w0BAQsFAADBdMQswCQYDVQGEwJJ
VDEPMA0GA1UEBwwGVmVyb25hMRQwEgYDVQQKDatDb21wYW55TmFtZTERMA8GA1UE
CwwIUjZEIHRlYW0xZDASBgNVBAMMC0hNSURldmljZU1EMB4XDTE4MDMyNjA5MTAz
OFoXDTI4MDMyMzA5MTAzOFowXTELMakGA1UEBhmMCSVQxDzANBgNVBACMB1Zlcm9u
YTEUMBIGAlUECgwLQ29tcGFueU5hbWUxETAPBgNVBAsMCFImRCBUZWFtMRQwEgYD
VQQDDatITU1EZXXZpY2VJRDCASiWdQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEB
ALONTzGwlrGv6cXH8i7sNWbwmX9Xo4tp20khnt/VJnDLoYHv7ZvV1vQYHom3/HiC
IaWV/uUvYnXaNBlxHnPsQPv0bEEg26Np01ne8jXEHY6bcMVK3XBV3eno3adOwHA5
vio0MmF6fPQVWtfyVb4/MrcfqUkelgWk3sFlFxEtXlRLOwNK1+G7Wbnb3Oj4oPL
Ev60VN3DwisDzvivpW7Nv4RPjNK9XJ2DVI+/+KDCNNLlP8GpD0xBliIpj1S8BwqZ
oml+SUsl0IMlcfv/AfArZj9QaIo3c2uPwkLncqQxfDvmlC1fCfsRVxm5N3bmimwC
2F6hbkZksLp7ovCx/haKhfkCAwEAATANBgbkqhkiG9w0BAQsFAAOCAQEALVjkNEa/
40JnMZIVkSZZWGylHHGZ8rphcUPH4olbq7MkaHk7mKacYKqI/qorriPhmKf7Y2x5
UcTN4Uff6NT0xjrMUg2Q6Lp+a/fBqOUvEebtm8NYbhjTs4iVYg3R/NBlgrfx9N
6Ipp06OJoOhYXjwDZU0HADnSXVABeBxzaESvLVK7mxgXypdB1D+kgcC6hL9Xv4u5
melNI24LNkRiBT35Exlo2YTu4I9YHFelc5iILvC6DpUYHeSlIEKiNmccL2DDGEBZ
TscrZykvWRilXpm2WMzjbf9HE0XNRM8DTCkOscxcrYZrcTVpm0a0WH5OD2531LnF
XsH5sLPyOxtKFw==
```

-----END CERTIFICATE-----

**Step 2:** Remove all Newline characters

```
MIIDNjCCAh4CCQCJtJgjqDDUqjANBgbkqhkiG9w0BAQsFAADBdMQswCQYDVQGEwJJ VDEPMA0GA1 . . . .
```

**Step 3:** Copy and paste the single text line of the certificate to the protocol dialog

### Script to generate a Certificate

If you want to use your own certificate, note that the certificate must include the “Subject Alternative Name (SAN)” parameters as required by the OPC UA standard.

Here is an example of how to generate certificate files using a public OpenSSL-Win32 library (Reference: <https://www.openssl.org/>)

```
@echo off
set OpenSSL="C:\Program Files (x86)\OpenSSL-Win32\bin\openssl.exe"
set NodeName=HMI-Client

rem Generate an RSA key
%OpenSSL% genrsa -out client-key.pem 2048

rem Creating Certificate Signing Requests
%OpenSSL% req -new -key client-key.pem -out client.csr -subj "/ST=NY/C=US/L=New
York/O=CompanyName/OU=R&D Team/CN=OPCUAClient@%NodeName%"

rem Creating Certificate (.pem)
echo subjectAltName=URI:urn:%NodeName%:CompanyName:OPCUAClient > san.txt
echo
keyUsage=digitalSignature,nonRepudiation,keyEncipherment,dataEncipherment,keyCertSign
>> san.txt
echo extendedKeyUsage=critical,serverAuth,clientAuth >> san.txt
echo authorityKeyIdentifier=keyid,issuer >> san.txt
echo basicConstraints=CA:TRUE >> san.txt
```



```
%OpenSSL% x509 -req -days 3650 -in client.csr -signkey client-key.pem -out
client.crt -extfile san.txt

rem Convert Certificate (.der)
%OpenSSL% x509 -in client.crt -outform der -out client.der

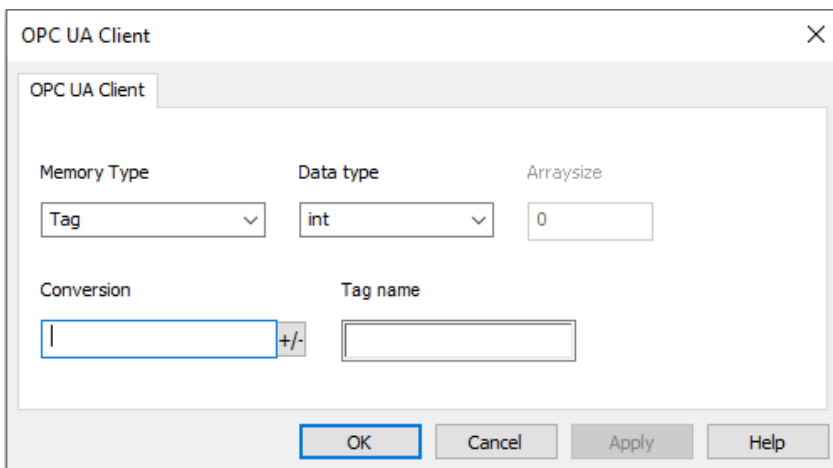
rem Not necessary files
del san.txt

pause
```

## Tag Editor Settings

Path: **ProjectView** > **Config** > double-click **Tags**


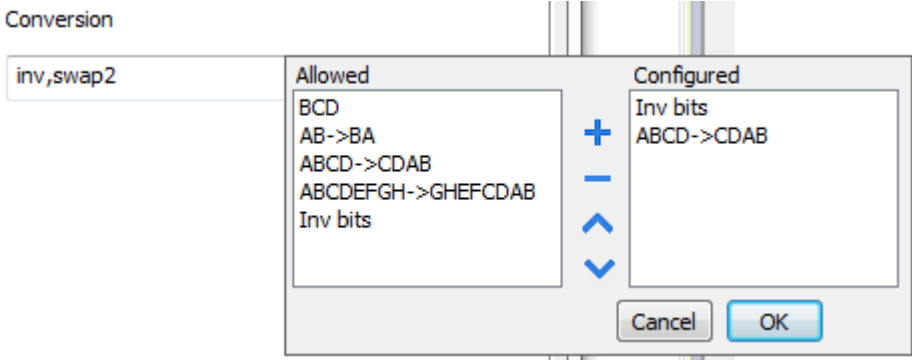
1. Select **OPC UA Client** from the protocol list.
2. To add a tag, click **+**: tag definition dialog is displayed.



The screenshot shows the 'OPC UA Client' dialog box for defining a tag. The dialog has a title bar with a close button (X). Inside, there is a tab labeled 'OPC UA Client'. The main area contains several fields:

- Memory Type:** A dropdown menu with 'Tag' selected.
- Data type:** A dropdown menu with 'int' selected.
- Arraysize:** A text input field containing '0'.
- Conversion:** A text input field containing a vertical bar '|', followed by a '+/-' button.
- Tag name:** An empty text input field.

At the bottom of the dialog, there are four buttons: 'OK', 'Cancel', 'Apply', and 'Help'. The 'OK' button is highlighted with a blue border.

Element	Description
<b>Data Type</b>	<p>Available data types:</p> <ul style="list-style-type: none"> <li>• <b>boolean</b></li> <li>• <b>byte</b></li> <li>• <b>short</b></li> <li>• <b>int</b></li> <li>• <b>unsignedByte</b></li> <li>• <b>unsignedShort</b></li> <li>• <b>unsignedInt</b></li> <li>• <b>float</b></li> <li>• <b>double</b></li> <li>• <b>time</b></li> <li>• <b>uint64</b></li> <li>• <b>int64</b></li> <li>• <b>string</b></li> <li>• <b>binary</b></li> </ul> <p>See "Programming concepts" section in the main manual.</p> <p> Note: To define arrays, select one of Data Type format followed by square brackets.</p>
<b>Arrays size</b>	<ul style="list-style-type: none"> <li>• In case of array tag, this property represents the number of array elements.</li> <li>• In case of string tag, this property represents the maximum number of bytes available in the string tag.</li> </ul> <p>Note: number of bytes corresponds to number of string characters if Encoding property is set to UTF-8 or Latin1 in Tag Editor. If Encoding property is set to UCS-2BE, UCS-2LE, UTF-16BE or UTF-16LE one character requires 2 bytes.</p>
<b>Conversion</b>	<p>Conversion to be applied to the tag.</p>  <p>Depending on data type selected, the list <b>Allowed</b> shows one or more conversion types.</p>

Element	Description	
	Value	Description
	<b>Inv bits</b>	<p><b>inv:</b> Invert all the bits of the tag.</p> <p><i>Example:</i>                      1001 → 0110 (in binary format)                      9 → 6 (in decimal format)</p>
	<b>Negate</b>	<p><b>neg:</b> Set the opposite of tag value.</p> <p><i>Example:</i>                      25.36 → -25.36</p>
	<b>AB -&gt; BA</b>	<p><b>swapnibbles:</b> Swap nibbles in a byte.</p> <p><i>Example:</i>                      15D4 → 514D (in hexadecimal format)                      5588 → 20813 (in decimal format)</p>
	<b>ABCD -&gt; CDAB</b>	<p><b>swap2:</b> Swap bytes in a word.</p> <p><i>Example:</i>                      9ACC → CC9A (in hexadecimal format)                      39628 → 52378 (in decimal format)</p>
	<b>ABCDEFGH -&gt; GHEFCDAB</b>	<p><b>swap4:</b> Swap bytes in a double word.</p> <p><i>Example:</i>                      32FCFF54 → 54FFFC32 (in hexadecimal format)                      855441236 → 1426062386 (in decimal format)</p>
	<b>ABC...NOP -&gt; OPM...DAB</b>	<p><b>swap8:</b> Swap bytes in a long word.</p> <p><i>Example:</i>                      142.366 → -893553517.588905 (in decimal format)                      0 1000000110                      000111001011101101100100010110100001110010101100                      0001                      →                      1 10000011100                      101010100001010001011011011011001011011000010011                      1101                      (in binary format)</p>
	<b>BCD</b>	<p><b>bcd:</b> Separate byte in two nibbles, read them as decimal (from 0 to 9)</p> <p><i>Example:</i>                      23 → 17 (in decimal format)                      0001 0111 = 23                      0001 = 1 (first nibble)                      0111 = 7 (second nibble)</p>

Element	Description
	<p>Select conversion and click +. The selected item will be added to list <b>Configured</b>.</p> <p>If more conversions are configured, they will be applied in order (from top to bottom of list <b>Configured</b>).</p> <p>Use the arrow buttons to order the configured conversions.</p>
<b>Tag name</b>	Name of tag to be used in communication.



Note: Tag properties result from import process. In most cases manual creation of new tags is not necessary.

## Node Override IP

The protocol provides the special data type Node Override IP which allows you to change the IP address of the target controller at runtime.

This memory type is an array of 4 unsigned bytes, one per each byte of the IP address.

The Node Override IP is initialized with the value of the controller IP specified in the project at programming time.

Node Override IP	PLC operation
<b>0.0.0.0</b>	Communication with the controller is stopped, no request frames are generated anymore.
<b>Different from 0.0.0.0</b>	It is interpreted as node IP override and the target IP address is replaced runtime with the new value.

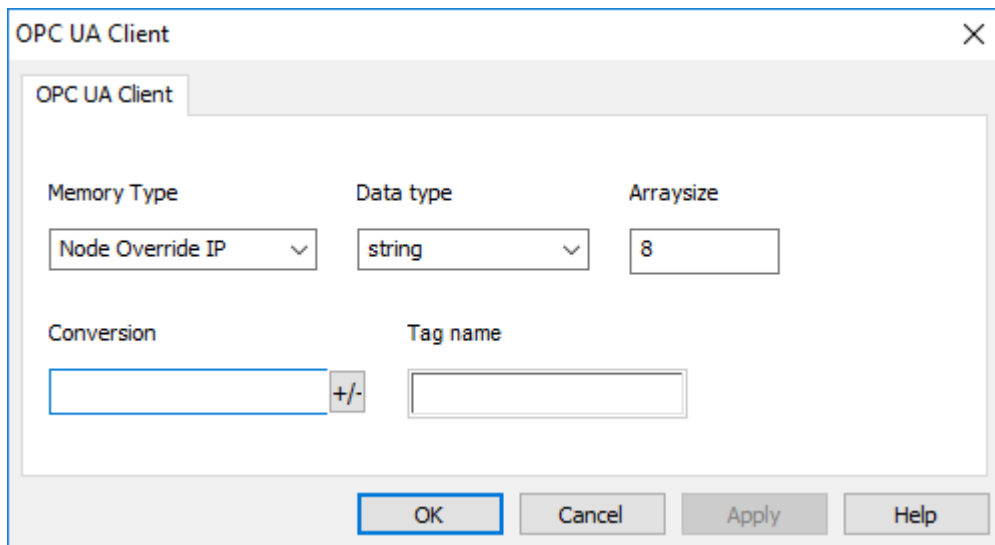
If the HMI device is connected to a network with more than one controller node, each node has its own Node Override IP variable.



Note: Node Override IP values assigned at runtime are retained through power cycles.

## Hostname DNS or mDNS

In addition to the array of bytes, string memory type can be selected to be able use the DNS or mDNS hostname as an alternative to the IP Address.



## Node Override Port

The protocol provides the special data type Node Override Port which allows you to change the network Port of the target controller at runtime.

This memory type is unsigned short.

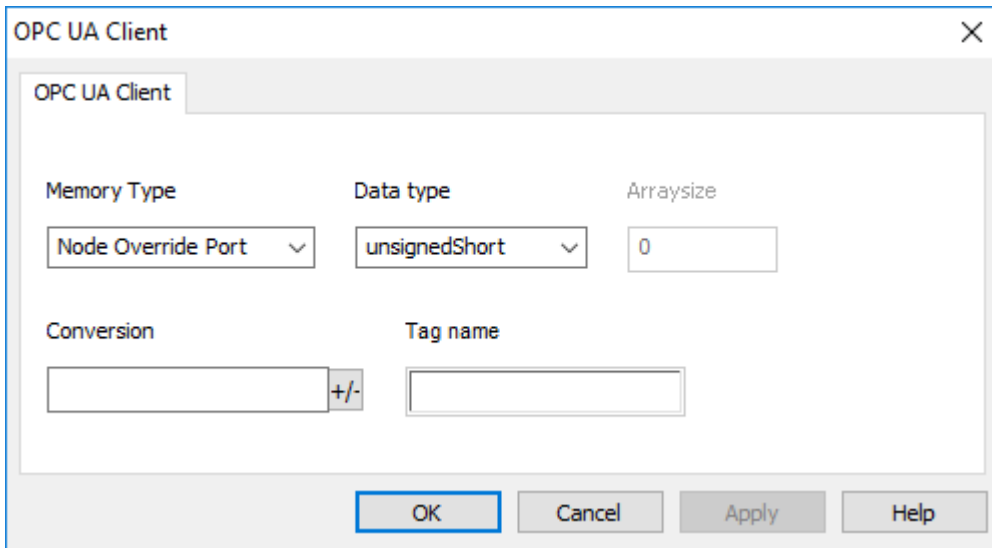
Node Override Port is initialized with the value of the controller Port specified in the project at programming time.

Node Override Port	Modbus operation
0	Communication with the controller is stopped, no request frames are generated anymore.
<b>Different from 0</b>	It is interpreted as the value of the new port and is replaced for runtime operation.

If the HMI device is connected to a network with more than one controller node, each node has its own Node Override Port variable.



Note: Node Override Port values assigned at runtime are retained through power cycles.

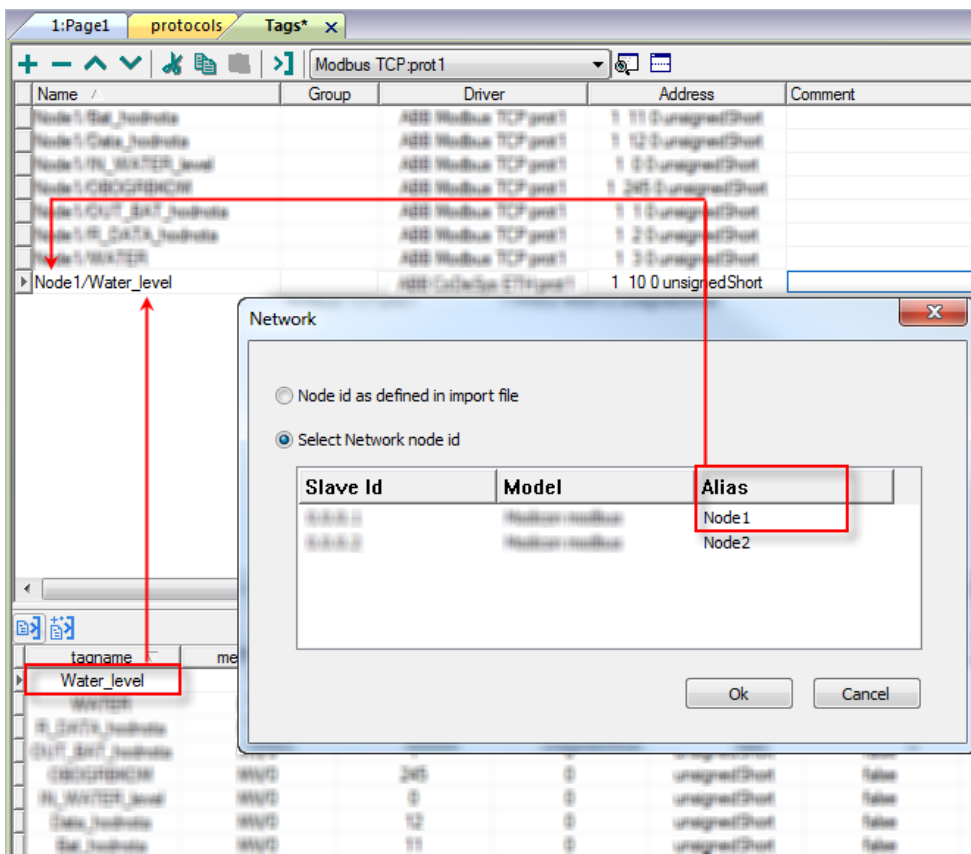


## Adding an alias name to a protocol

Tag names must be unique at project level, however, the same tag names might need to be used for different controller nodes (for example when the HMI device is connected to two devices running the same application).

When creating a protocol you can add an alias name that will be added to tag names imported for this protocol.

In the example, the connection to a certain controller is assigned the name **Node1**. When tags are imported for this node, all tag names will have the prefix **Node1** making each of them unique at the network/project level.





Note: Aliasing tag names is only available for imported tags. Tags added manually in the Tag Editor cannot have the Alias prefix in the tag name.

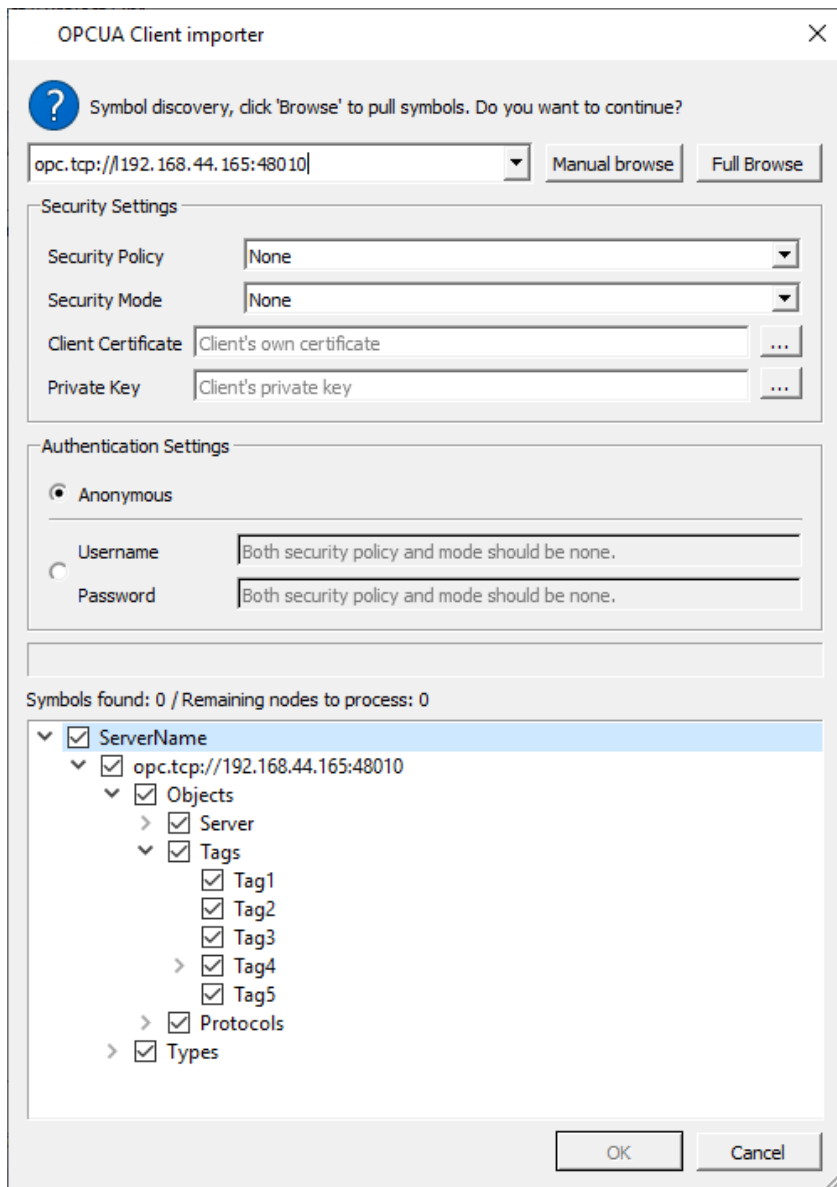
The Alias string is attached at the time of tag import. If you modify the Alias string after the tag import has been completed, there will be no effect on names already present in the dictionary. When the Alias string is changed and tags are re-imported, all tags will be re-imported with the new prefix string.

## Importing tags

Tags for OPC UA Client protocol must be imported from OPC UA servers.


*Path: **ProjectView** > **Config** > double-click **Tags***


1. Select **OPC UA Client** in the list of available protocols.
2. Click **Import Tags**.
3. Select **Hierarchical importer**.
4. Enter address of the server.
5. Choose Security and Authentication mode.
6. Click **Browse** to connect and retrieve tag dictionary from the OPC UA server.
7. The OPC UA Server will provide its own certificate. You have to accept the certificate to continue and retrieve data.
8. When the discovery process has been completed, click **OK** to create the dictionary with the tags.



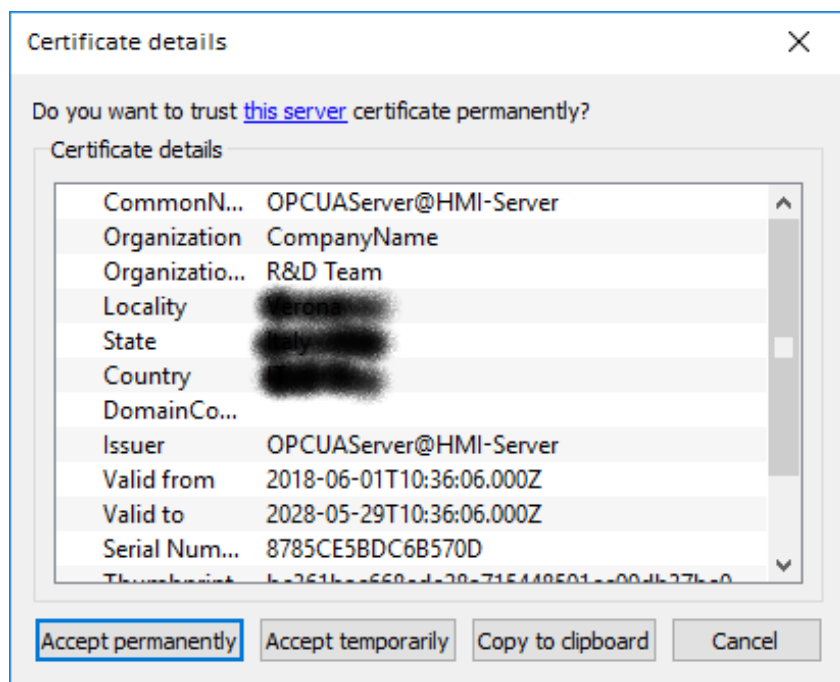
Element	Description
Remote URI	Address of OPC UA Server in the form: <i>opc.tcp:&lt;IPAddress&gt;:&lt;Port&gt;</i> Example: <ul style="list-style-type: none"> <li>opc.tcp://192.168.44.165:4840</li> </ul>
Security Mode	Type of authentication: <ul style="list-style-type: none"> <li><b>None:</b> No authentication with server and no data encryption.</li> <li><b>Sign:</b> Certificates only used for authentication with server.</li> <li><b>SignAndEncrypt:</b> Certificates used for authentication with server and data encryption.</li> </ul>
Security Policy	Encryption level to use (used only when Security Mode is active).



Element	Description
	<ul style="list-style-type: none"> <li>• Basic128Rsa15</li> <li>• Basic256</li> <li>• Basic256Sha256</li> </ul>
<b>Username Password</b>	Authentication with user name and password
<b>Client Certificate</b>	<p>Certificate used by OPC UA client. If blank, a certificate is automatically generated.</p> <p> The certificate is used by the importer only if requested by the server</p>
<b>Client Private Key</b>	Key used by OPC UA client. If blank, a key is automatically generated.


 To be allowed to retrieve data from the OPC UA Server you must provide the required security parameters. Dialog will be filled automatically with the parameters provided by protocol editor settings (you can simply accept the proposed values)

**Remote OPC UA Server certificate**



When OPC UA Server provides its own certificate, you have the option to:

- **Accept temporarily**  
Certificate is accepted for current working session only.
- **Accept permanently**  
Certificate is accepted and copied to computer. Any future import request for the same OPC UA Server will be accepted automatically without asking confirmation.

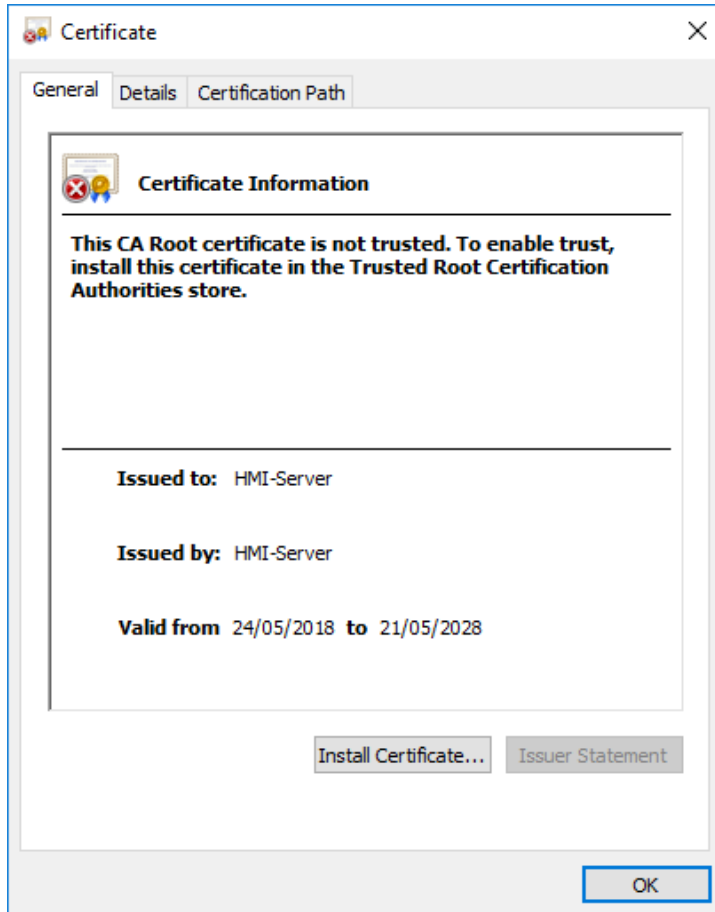
 The certificate file will be copied inside the folder:  
%AppData%\Roaming\...\studio\OPCUA\pki\trusted\certs

- **Copy to clipboard**

ASCII format of the certificate is copied to the clipboard to allow you to verify its authenticity, save and insert it into protocol configuration (if required).



To verify a certificate, use a text editor to paste it from the clipboard to a text file with the extension .crt. You can then double-click the .crt file to allow Windows to view the properties of certificate.



- **Cancel**

Cancel the import operation

## Communication status

Current communication status can be displayed using System Variables. See "System Variables" section in the main manual.

Codes supported for this communication driver:

Error	Description
<b>Connecting</b> <Error description>	Error during connection
<b>Connection while reading:</b> <Error description>	Error encountered when connecting for read operation
<b>Bad status while reading:</b> <Error description>	Error in read operation

Error	Description
<b>Connection while writing: &lt;Error description&gt;</b>	Error encountered when connecting for write operation
<b>Bad status writing: &lt;Error description&gt;</b>	Error in write operation
<b>OPC UA client for given node ID not found</b>	Wrong node ID information

<Error description> can be one of the following:

Error	Notes
<b>BadTimeout</b>	Timeout error. No answer from server.
<b>BadSecurityChecksFailed</b>	Error during exchange of certificates. Typically occurs when the server does not accept the client certificate as trusted.
<b>BadCertificateInvalid</b>	Error in client or server certificate.
<b>BadNodeUnknown</b>	The tag (node) does not exist.
<b>BadAttributeNotFound</b>	Attempt to access an invalid attribute.
<b>BadNotWritable</b>	Attempt to write to a read-only attribute.

# Simatic S7 ETH

Simatic S7 ETH communication driver has been designed to communicate with Simatic controllers through Ethernet connection.

The Simatic controller must either have an on-board Ethernet port or be equipped with an appropriate Ethernet interface (either built-in or with a module).

Communication is based on the PG/OP (ISO on TCP) communication functions.

This documents describes the driver settings to be applied in programming IDE software and in S7 PLC programming software.

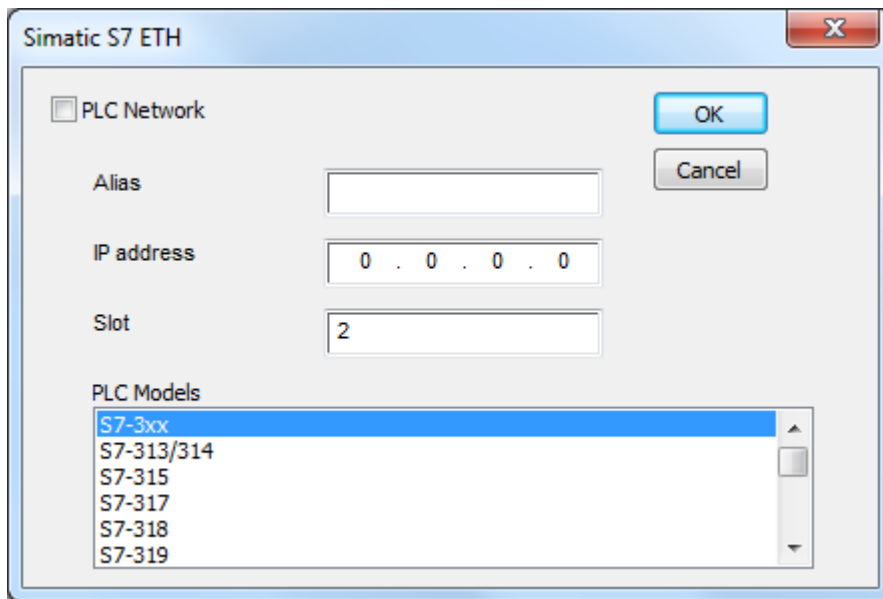
## Protocol Editor Settings

### Adding a protocol

To configure the protocol:

1. In **Config** node double-click **Protocols**.
2. To add a driver, click **+**: a new line is added.
3. Select the protocol from the **PLC** list.

The protocol configuration dialog is displayed.



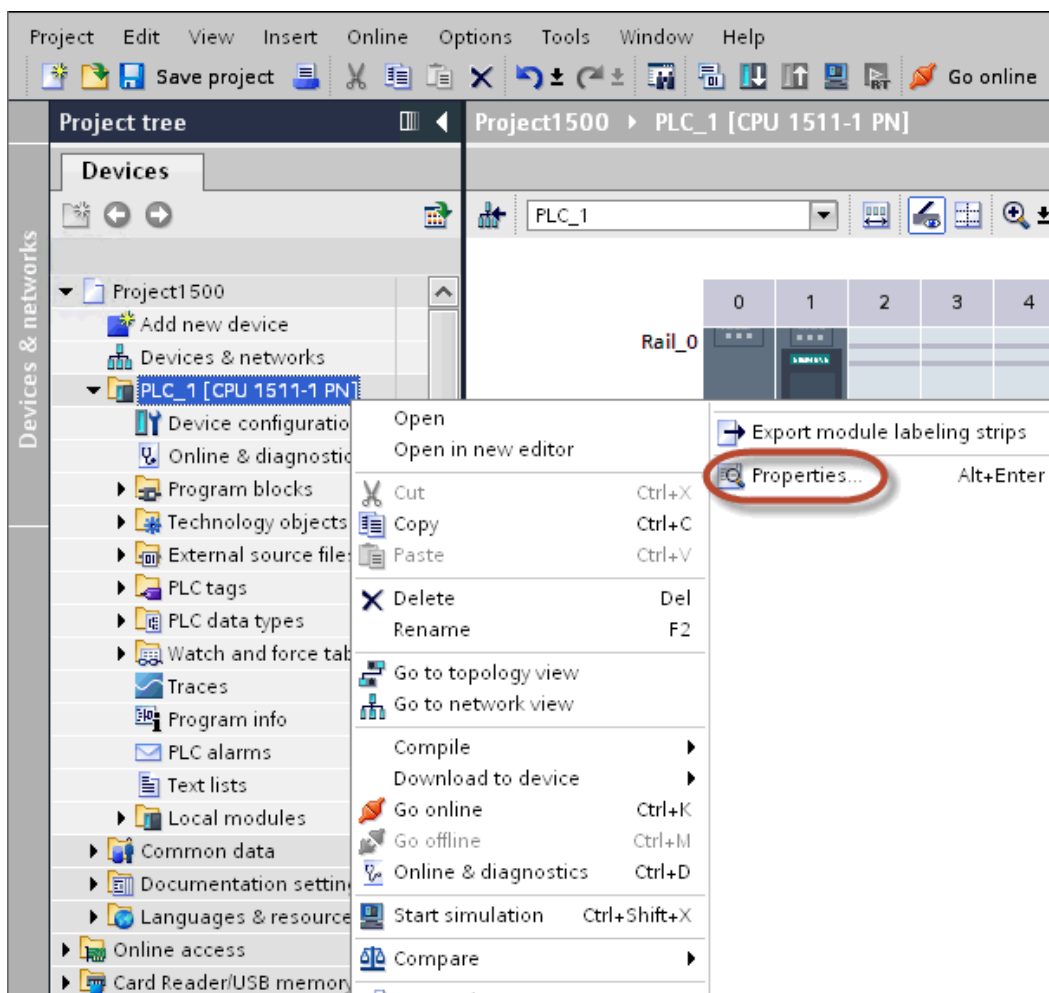
Element	Description
<b>Alias</b>	Name identifying nodes in network configurations. The name will be added as a prefix to each tag name imported for each network node.
<b>IP address</b>	Ethernet IP address of the controller.
<b>Slot</b>	Number of the slot where the CPU is mounted. 2 for S7-300, may take a higher value for S7-

Element	Description
	400 systems.
<b>PLC Models</b>	List of compatible controller models. Make sure to select the correct PLC model in this list when configuring the protocol.
<b>PLC Network</b>	Enable access to multiple networked controllers. For every controller (slave) set the proper option.

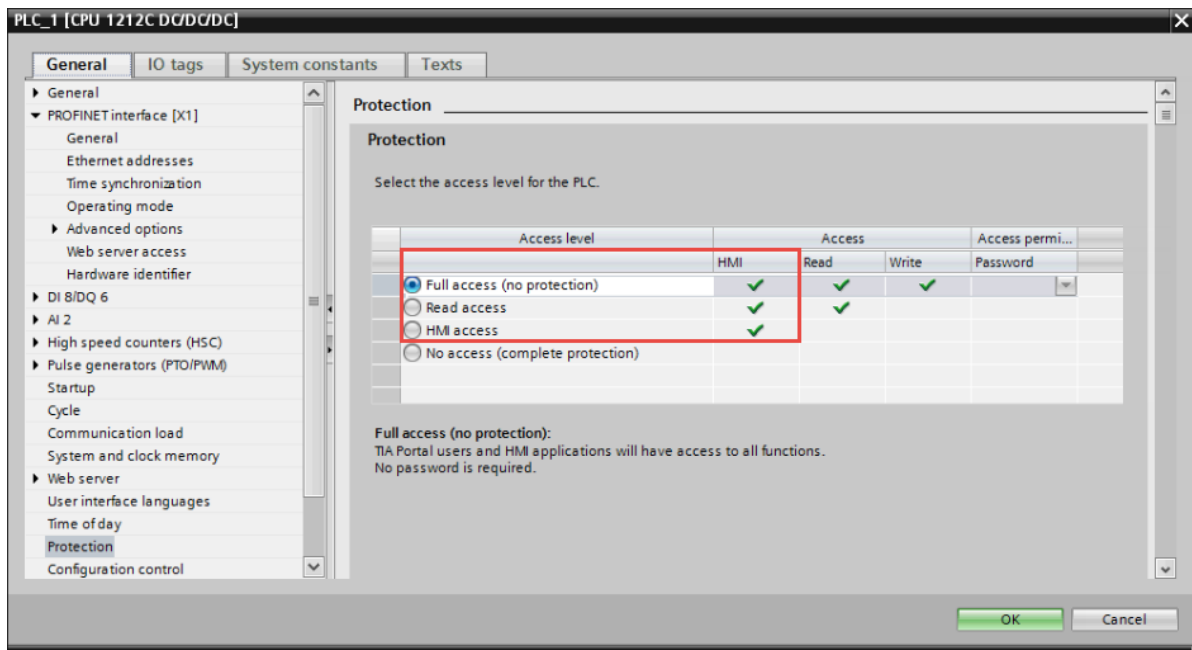
## S7-1200 and S7-1500 PLC configuration

S7-1200 (starting from firmware version 4.0) and S7-1500 PLC Series from Siemens have a built-in firewall; by default the maximum protection level is enabled. To establish communication with these PLC models it is necessary to enable S7 communication with 3<sup>rd</sup> party devices; this setting is available in TIA Portal programming software.

1. Open the PLC project in TIA Portal.
2. Select the PLC from the project tree and open PLC Properties.

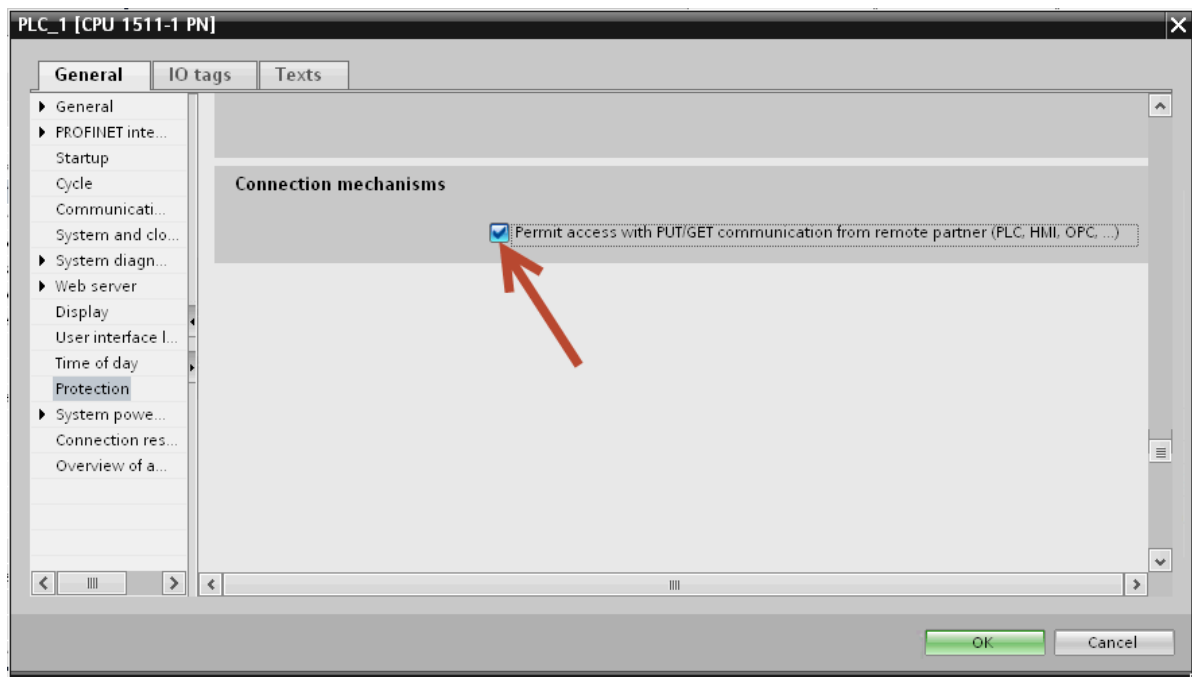


3. In General > Protection choose a permission between the top three (make sure that the tick is present on HMI column).



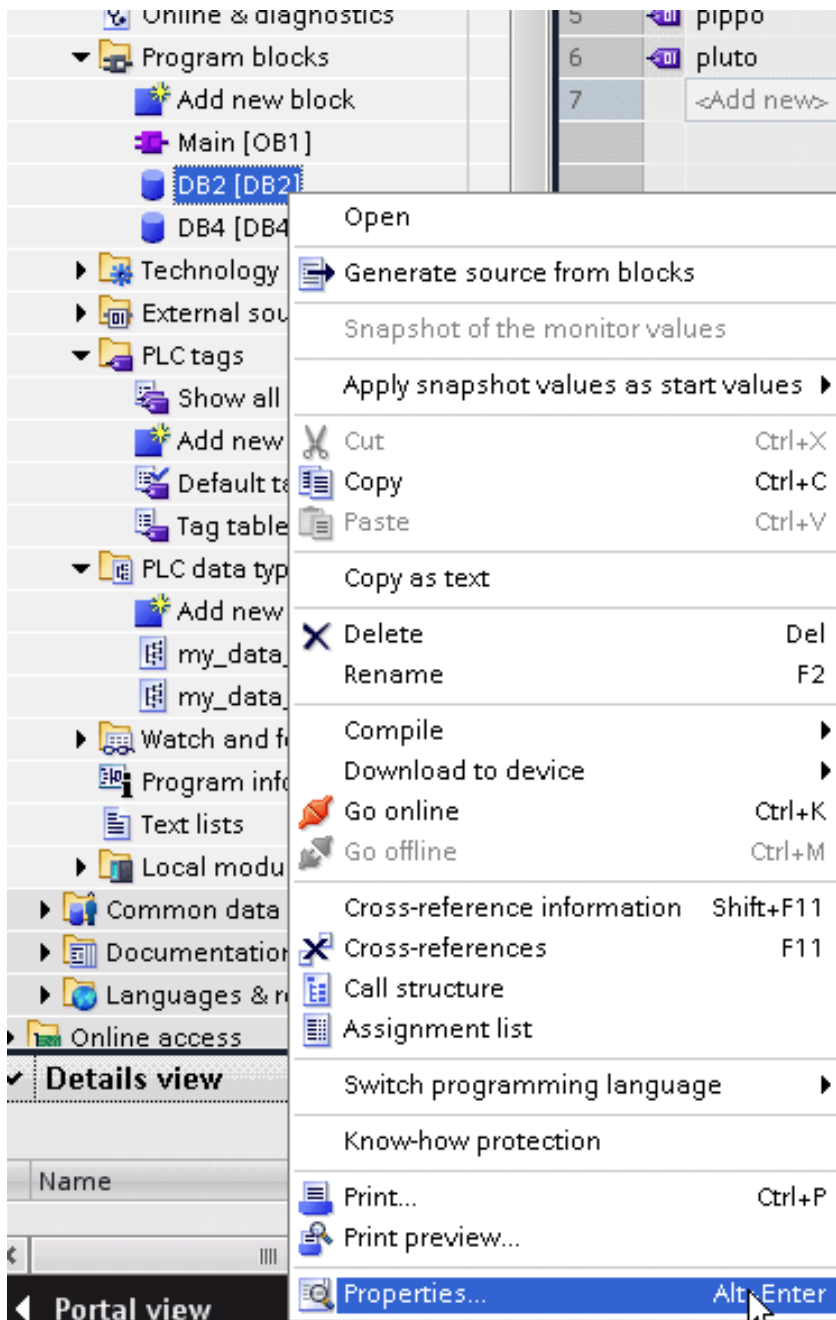
Note: If "No access" is selected, the communication with the panel will not be established.

4. Scroll down the page and check "Permit access with PUT/GET communication from remote partner".

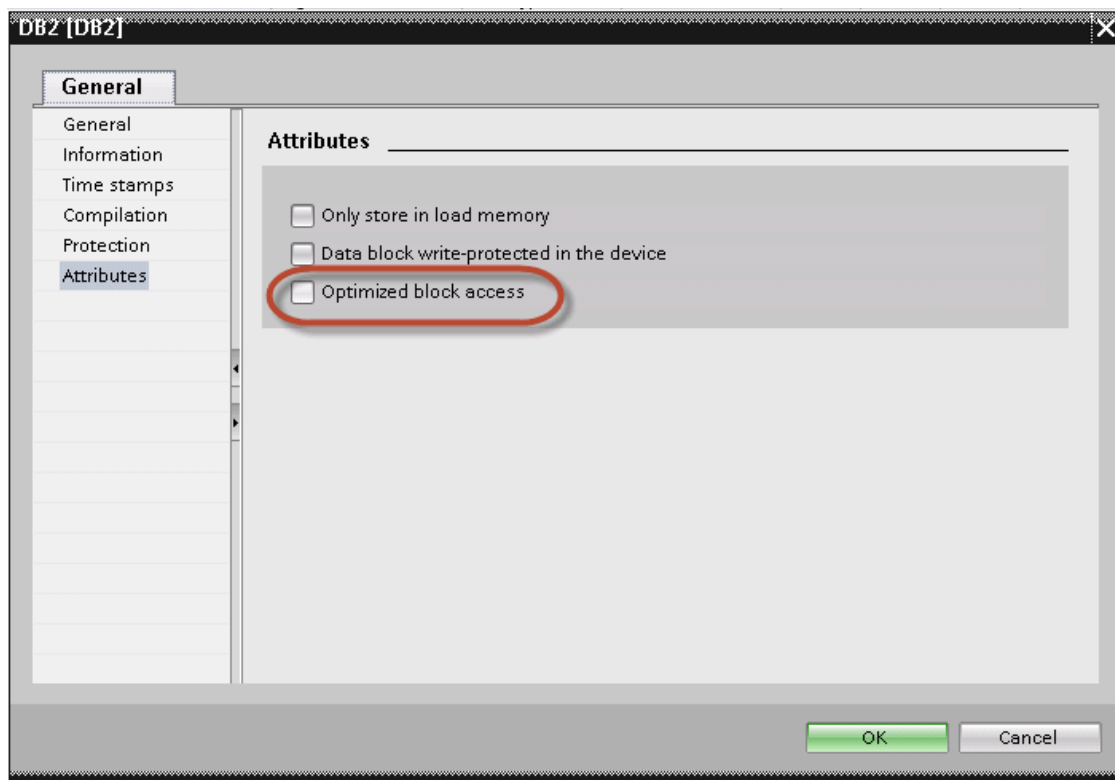


Note: If variables are defined in "Program blocks", DB must be configured as "Not optimized".

To check or change DB optimization, open DB Properties:



In General > Attributes uncheck "Optimized block access":



If check box "Optimized block access" is not available (grayed-out) it could be because DB is an "instance DB" linked to an "optimized access FB".

After compiling the project, tag offsets will be shown close to variable name.

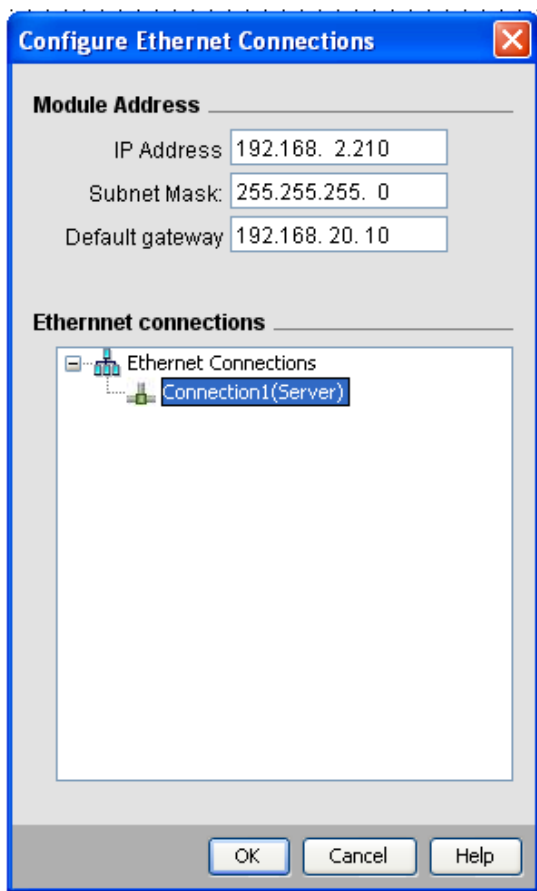
These settings can be applied to TIA Portal programming software, S7-1200 PLC family starting from PLC firmware version 4.0 and S7-1500 PLC family.

## Logo! PLC configuration

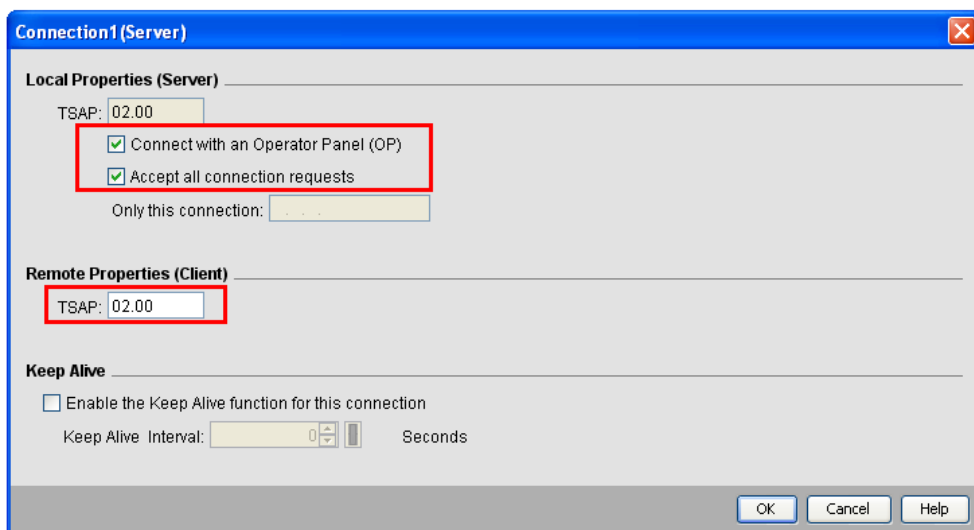
To configure communication with Logo! PLC:

1. Open the Logo!Soft Comfort project.
2. Select **Tools > Ethernet Connections**: the Configure Ethernet Connections dialog is displayed.





3. Right-click on **Ethernet Connections** and add a server connection.
4. Double-click on the newly created connection: the connection properties dialog is displayed.



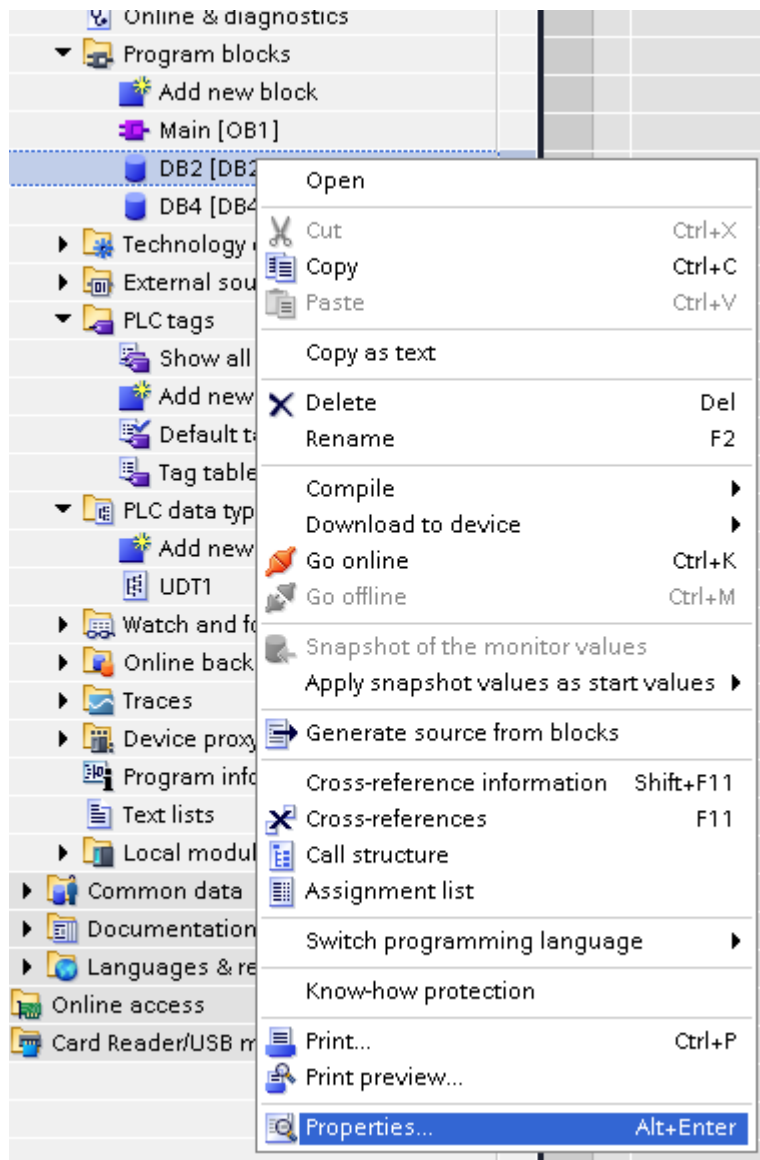
5. Select the **Connect with an operator panel (OP)** (0BA7 model only, do noth check for Logo! 0BA8 model)
6. Select Accept all connection requests options.
7. In the **Remote Properties (Client)** section, set **TSAP** to 02.00.

## Direct Import of TIA Portal project

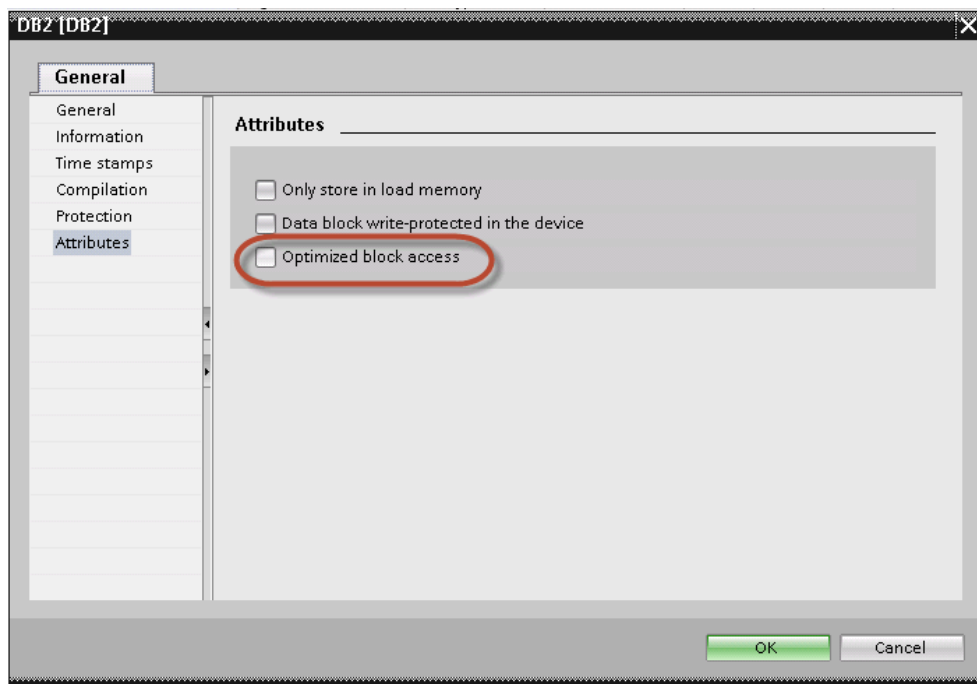
It is possible to import TIA Portal variables directly from TIA Portal project, by selecting "TIA Portal Project v12 or newer" from import selection (refer to "Tag Import" chapter).

Data Blocks must be set as Not optimized:

1. Configure the Data Block as **Not optimized**.
2. Right-click on the Data Block and choose **Properties**:



3. In the **General** tab select **Attributes** and unselect **Optimized block access**.



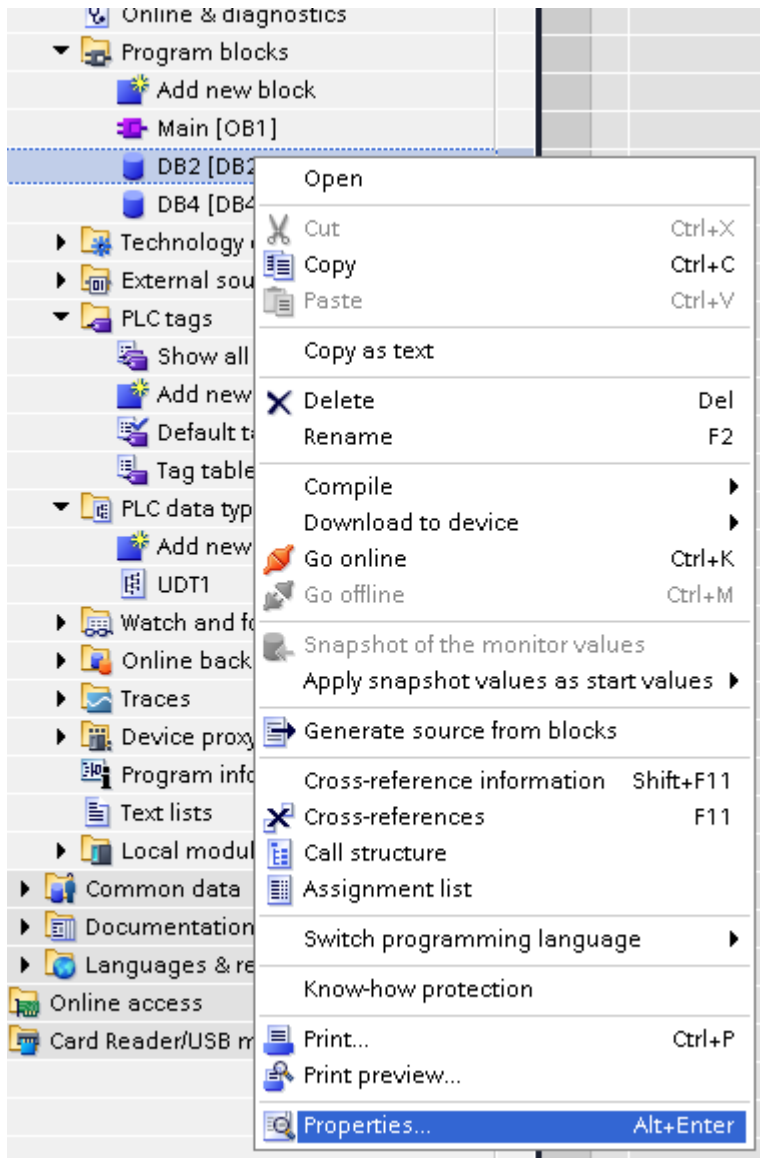
Note: If the options **Optimized block access** is not enabled (checkbox grayed out) this might mean that the Data Block is an "instance DB" linked to an "optimized access FB".

## Export using TIA Portal v13, v14 or newer

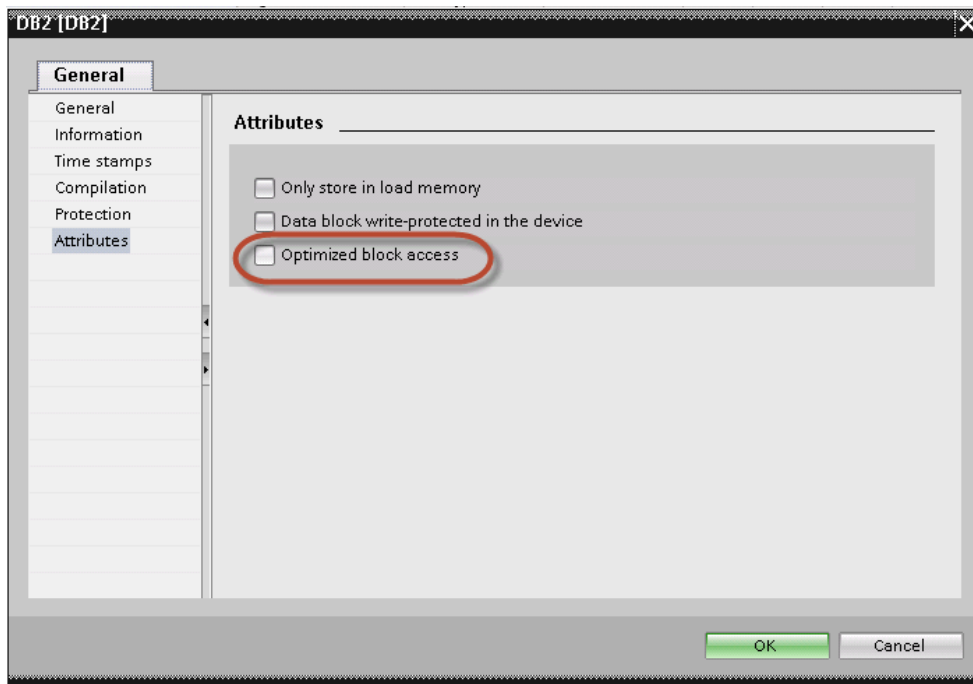
### Exporting Program blocks


These files refer to DB tags defined in **Program blocks**.

1. Configure the Data Block as **Not optimized**.
2. Right-click on the Data Block and choose **Properties**:

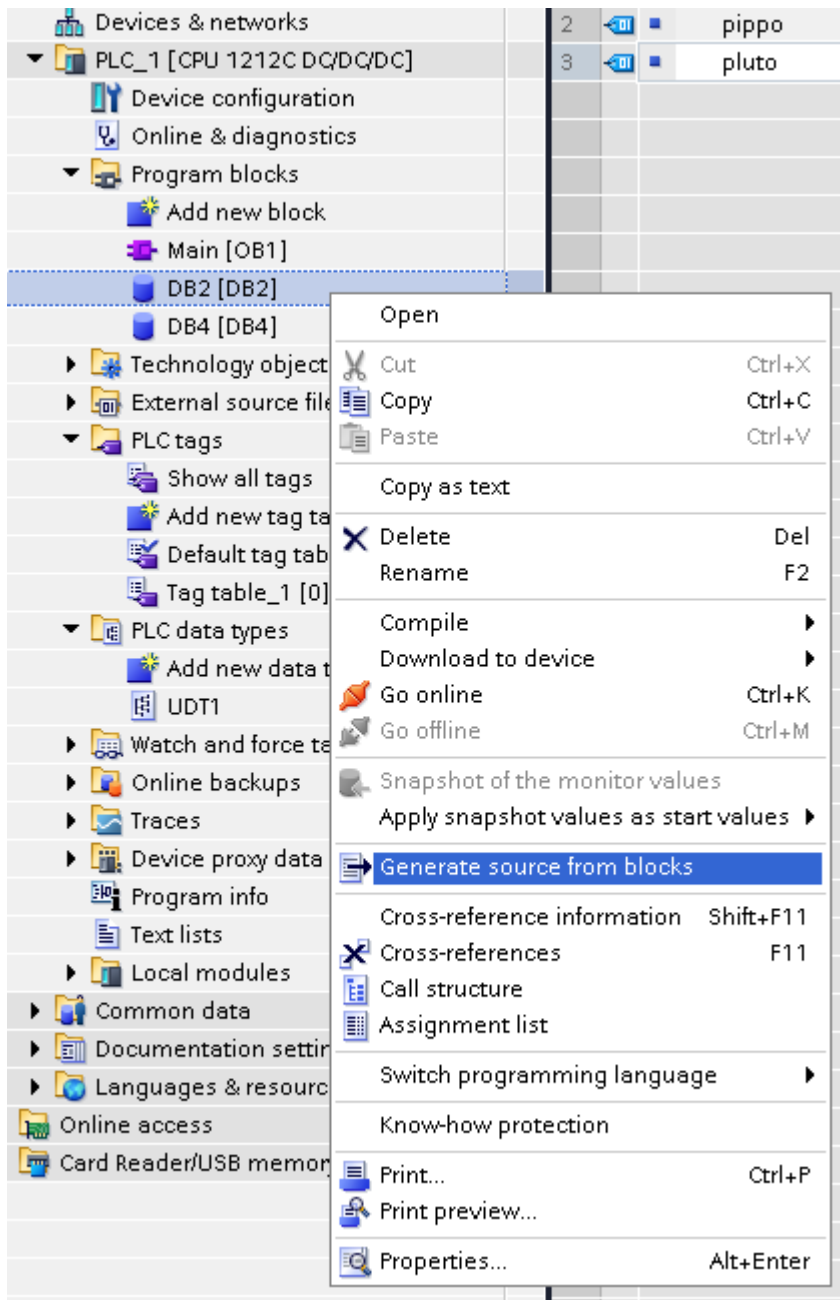


3. In the **General** tab select **Attributes** and unselect **Optimized block access**.

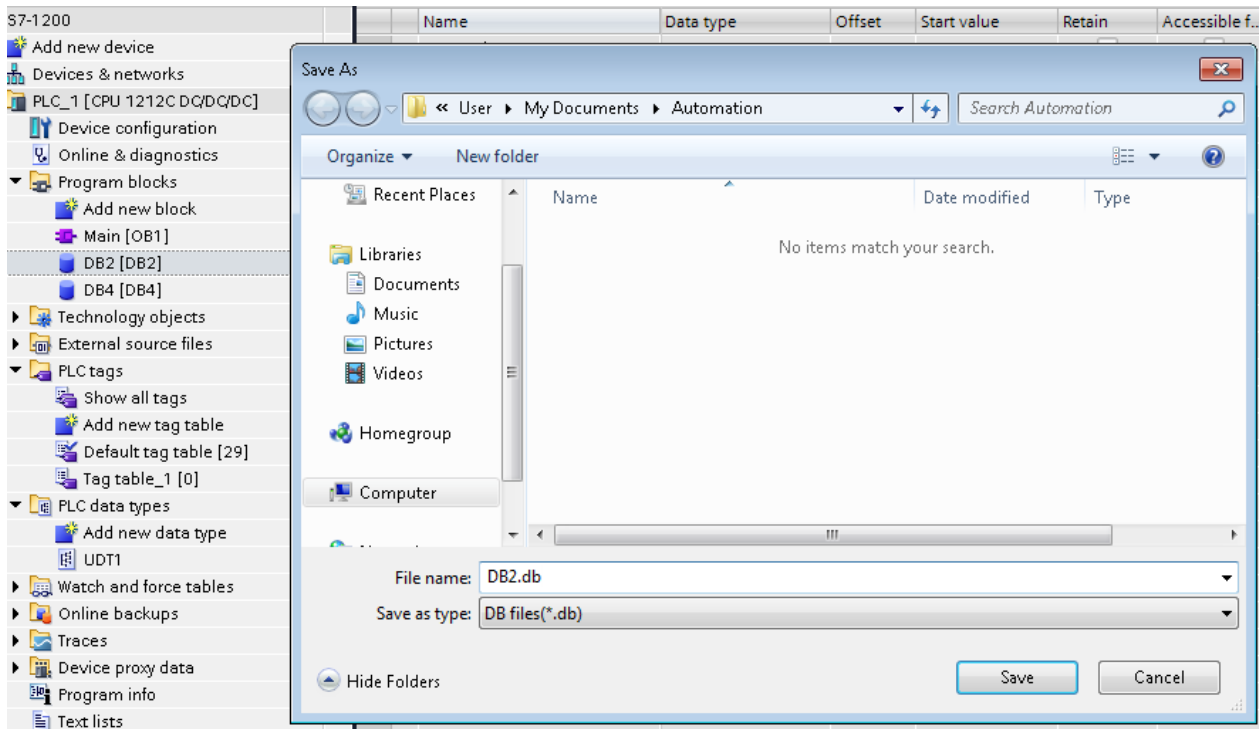


 Note: If the options **Optimized block access** is not enabled (checkbox grayed out) this might mean that the Data Block is an "instance DB" linked to an "optimized access FB".

4. Right-click on the Data Block and choose **Generate source from blocks**:



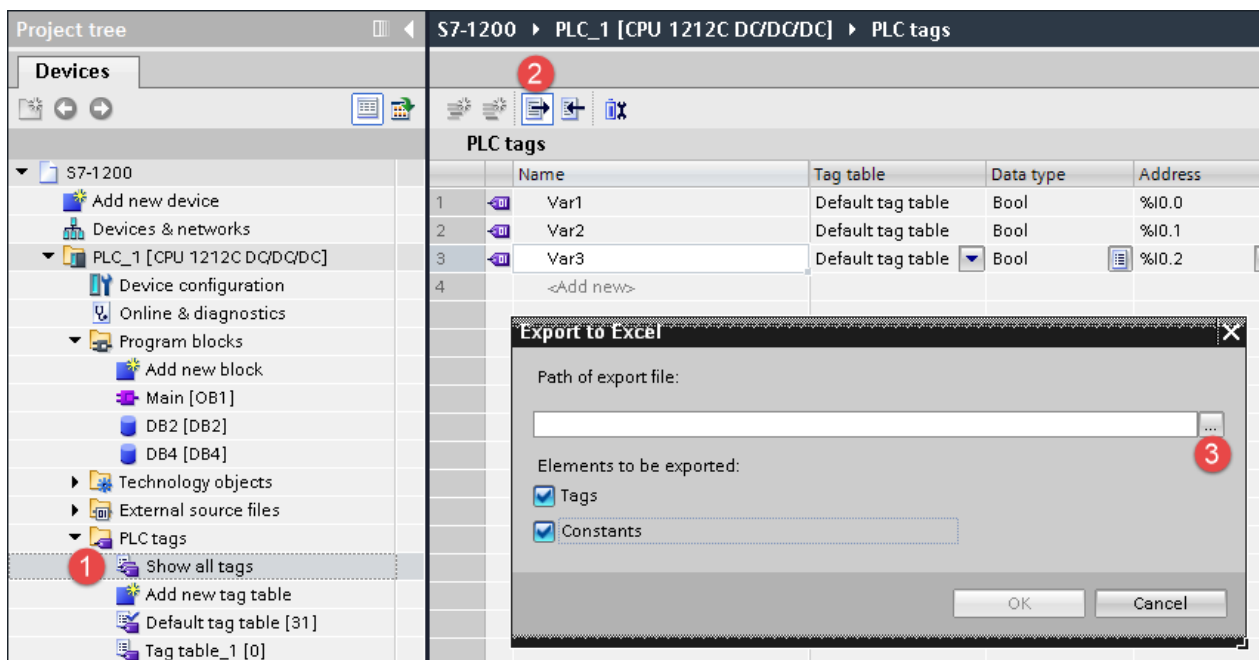
5. Save the file as DBxxx.db, where xxx=number of DB.



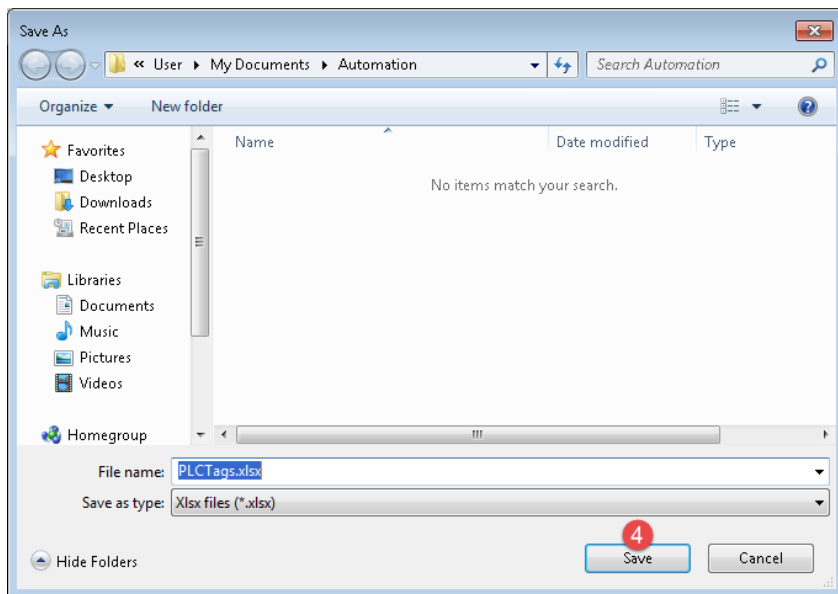
## Exporting PLC tags

An Excel file refers to PLC tags.

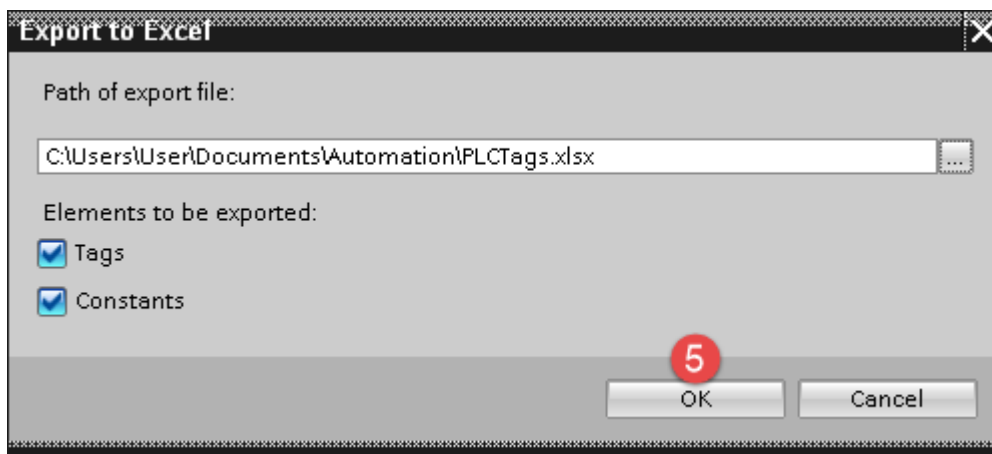
1. Double-click **Show all tags**: the tag table is displayed.
2. Click the **Export** button and browse for path file.
3. Define file name.



4. Click **Save** to confirm.



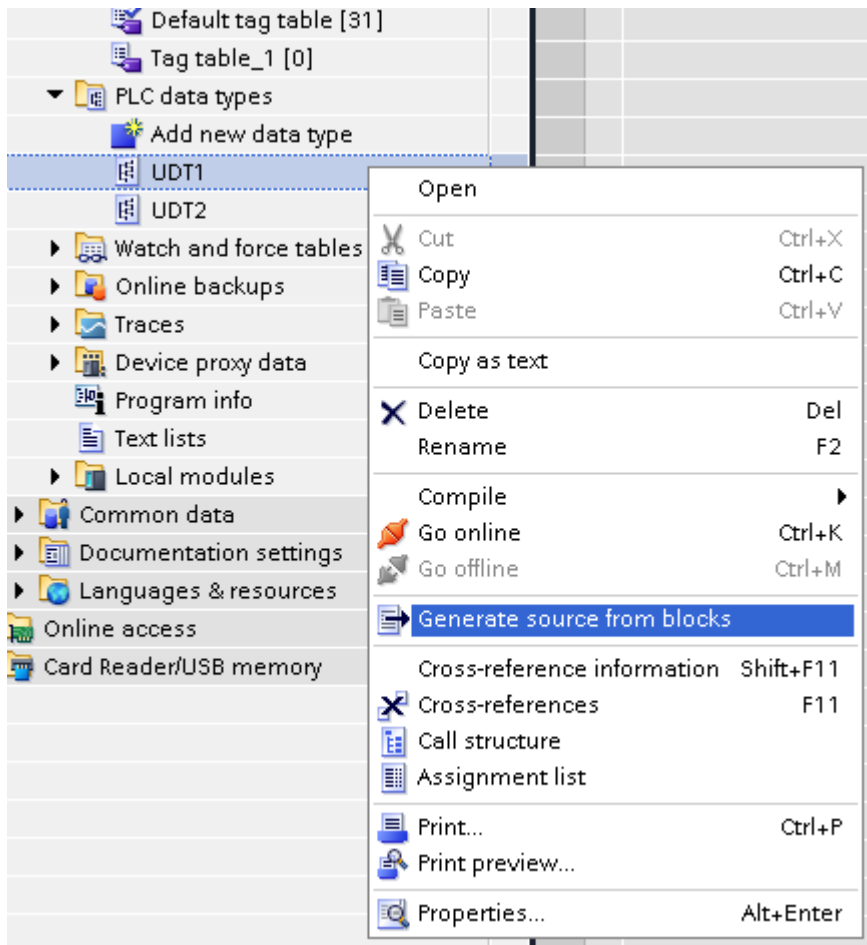
5. Click **OK** to export.



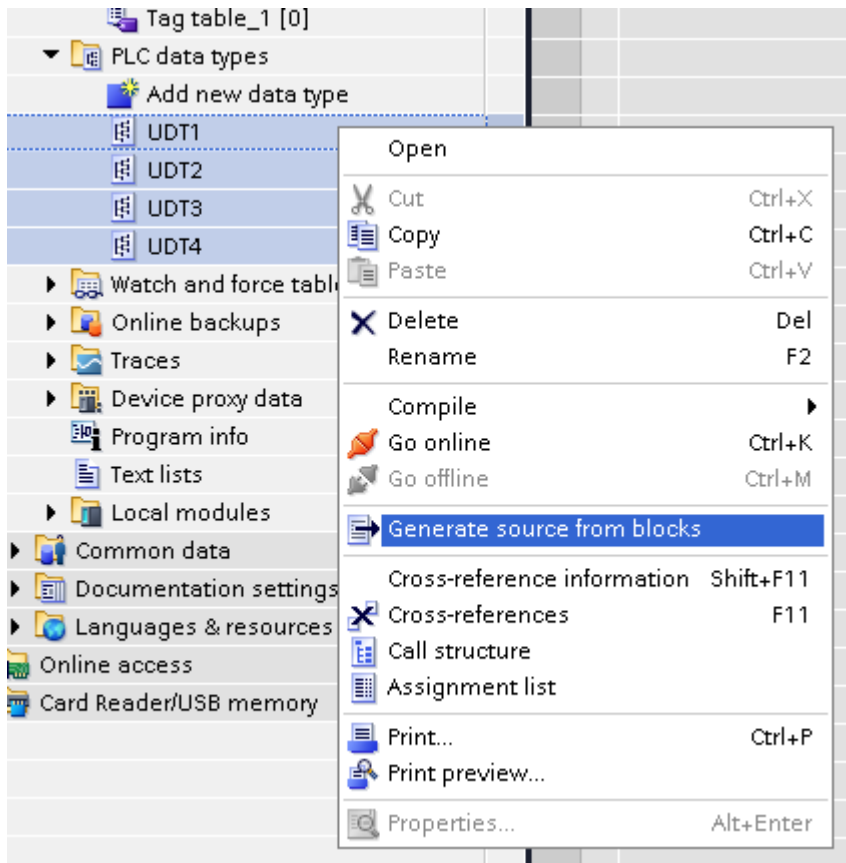
## Exporting PLC data types

To create the file, expand **PLC data types** item from TIA Portal project tree and right click on the user defined structure. Then click on **Generate source from blocks**.

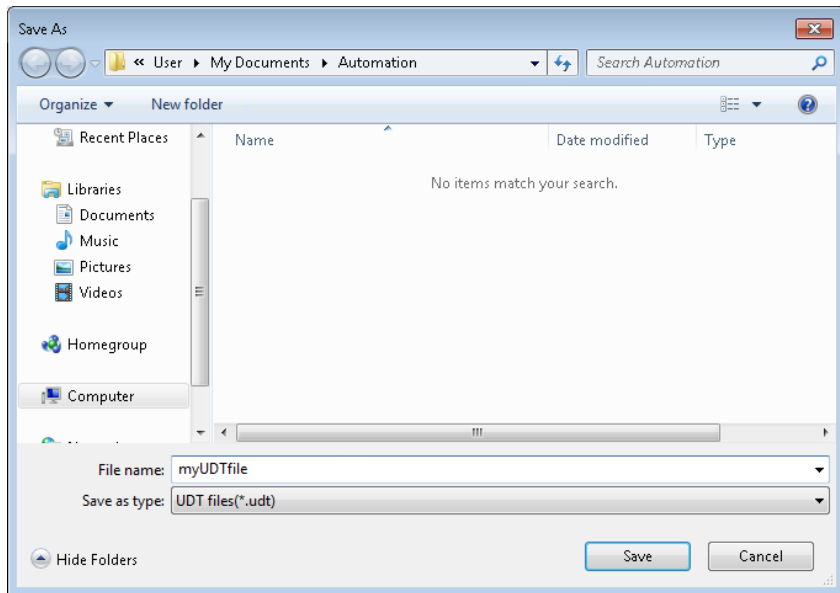




In case of multiple PLC data types in PLC project, it is necessary to select them all from **PLC data types** list, right click and select **Generate source from blocks** to create the .UDT file that contains all the PLC data types defined.



In the next step, give a name to the .UDT file and choose the path to where to save the file.



This file will content all the PLC data types and it can be used for importing tags in Tag Editor.

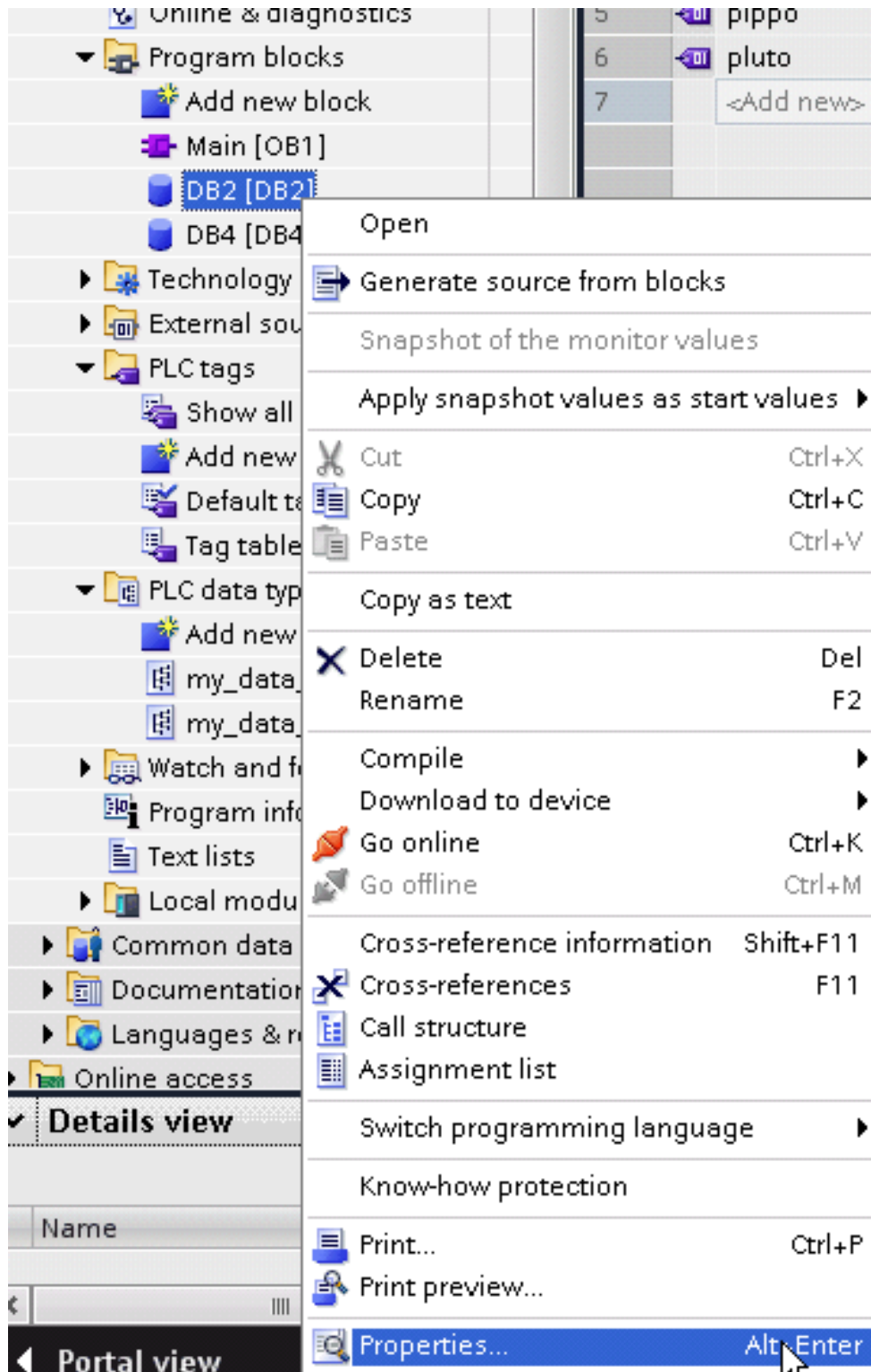
Check **Tag Import** chapter for more details.

## Export using TIA Portal v10, v11, v12

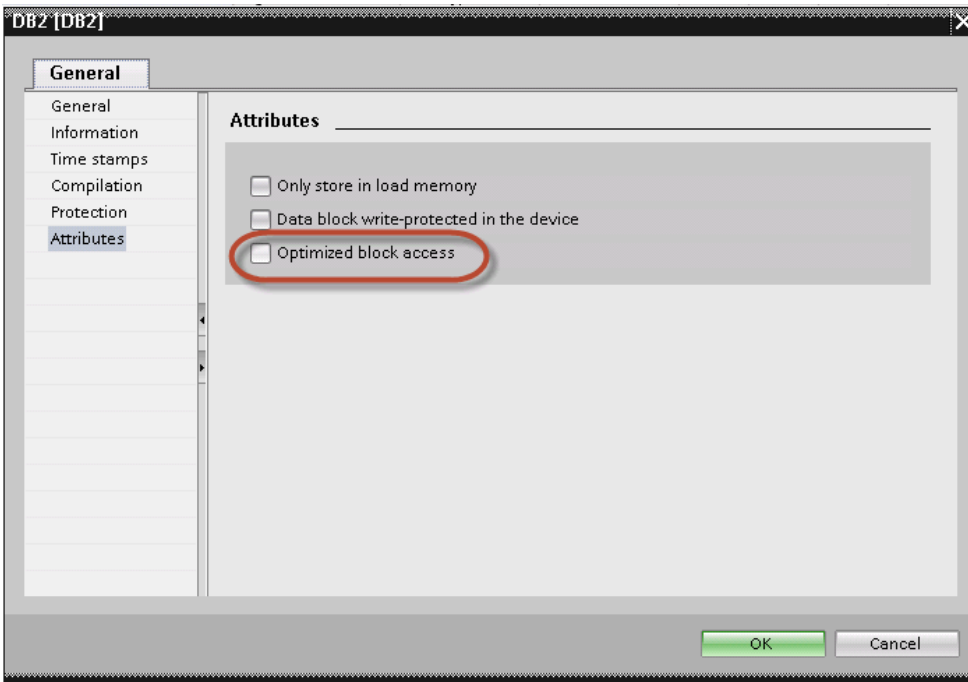
## Exporting Program blocks

These files refer to DB tags defined in **Program blocks**.

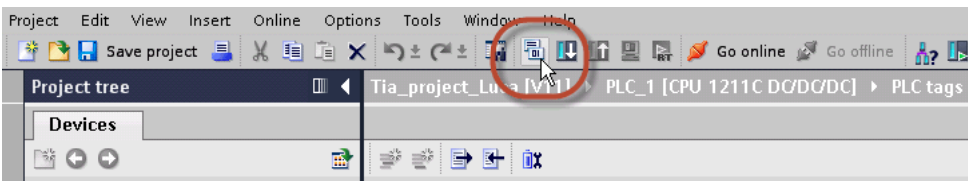
1. Configure the Data Block as **Not optimized**.
2. Right-click on the Data Block and choose **Properties**:



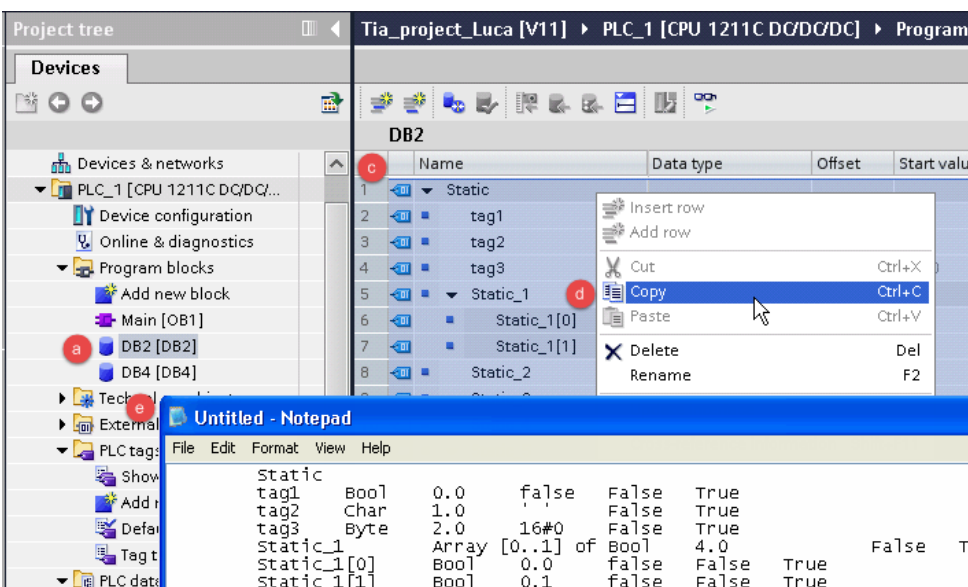
3. In the **General** tab select **Attributes** and unselect **Optimized block access**.




Note: If the options **Optimized block access** is not enabled (checkbox grayed out) this might mean that the Data Block is an "instance DB" linked to an "optimized access FB".




4. Build the project to make sure TIA Portal calculates the tags offset.

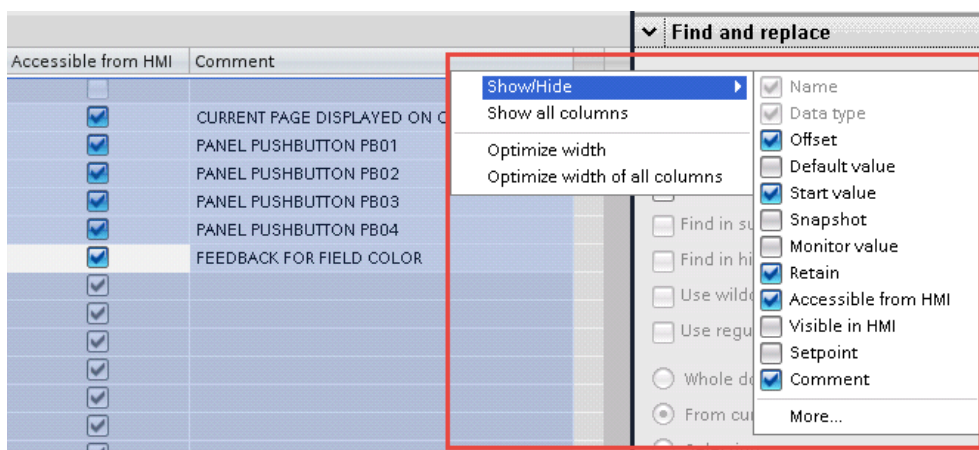


5. Double-click on a DB name.
6. Expand the view of program block selected.
7. Select all rows.
8. Copy and paste into any text editor.
9. Save the file as DBxxx.tia, where xxx=number of DB.

 Note: Make sure you use the **Save As** function or the file will be named DB2.tia.txt and will not be visible from the importer.

10. Repeat from step 5 for all program blocks.

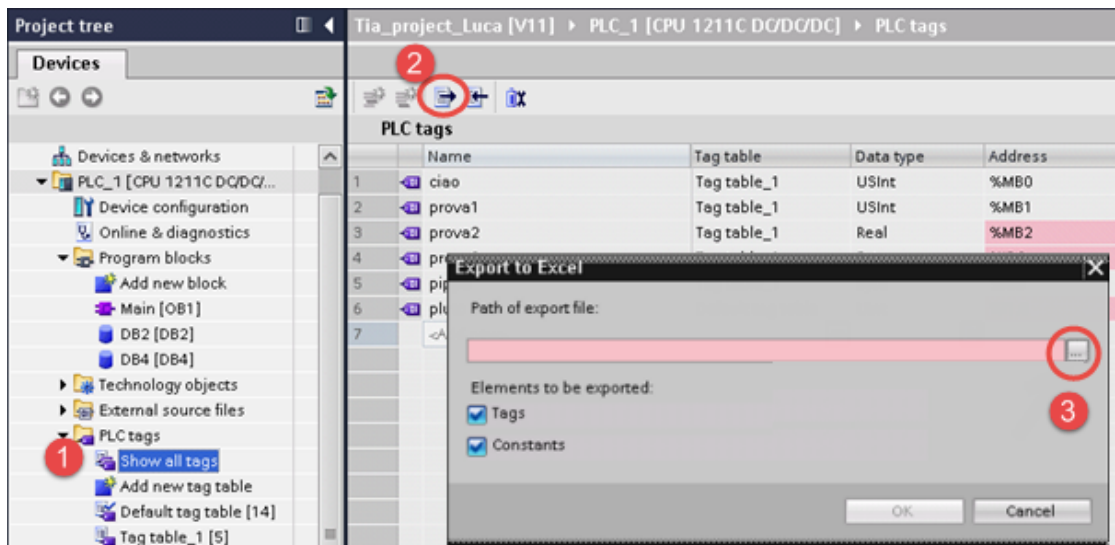
 Note: Make sure that only the following columns are shown in DB editor before copying all data in the txt file



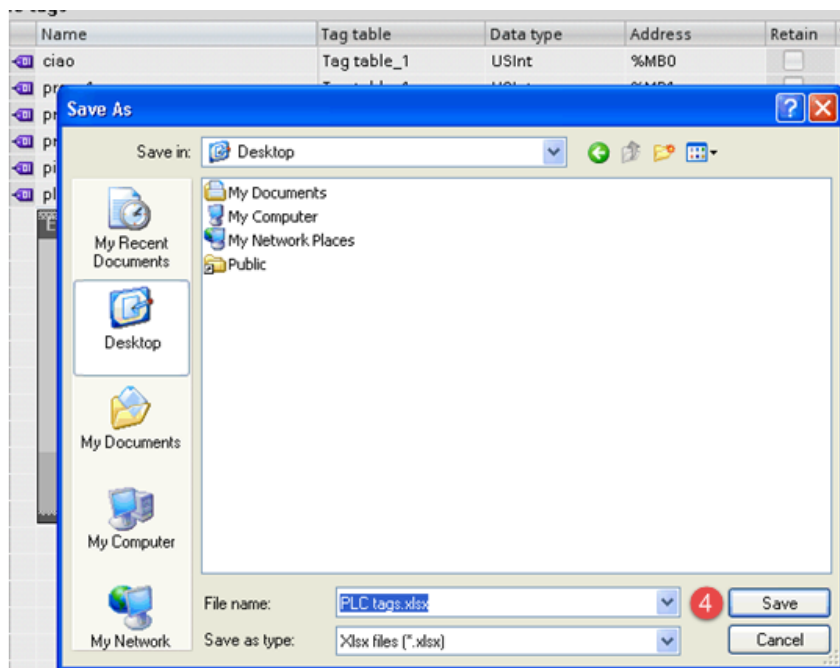
## Exporting PLC tags

An Excel file refers to PLC tags.

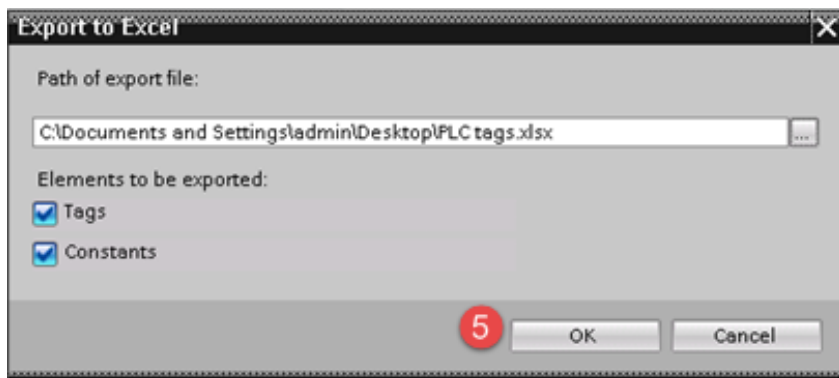
1. Double-click **Show all tags**: the tag table is displayed.



2. Click the **Export** button and browse for path file.
3. Define file name.
4. Click **Save** to confirm.

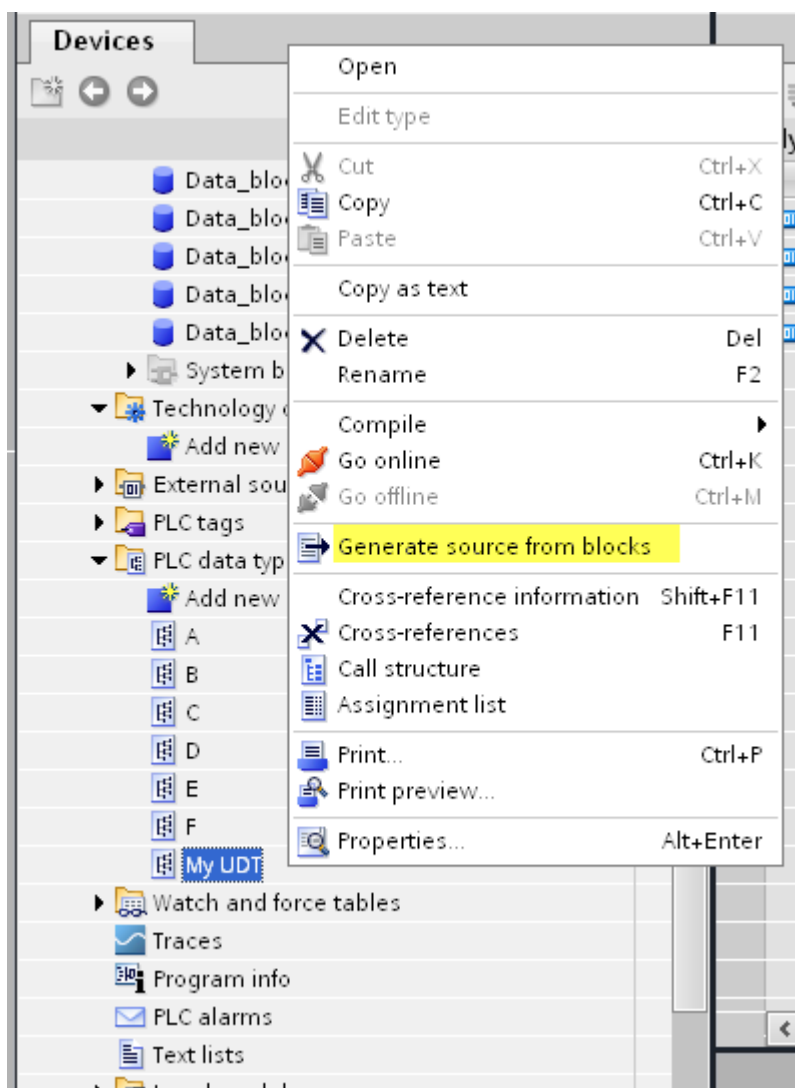


5. Click **OK** to export.

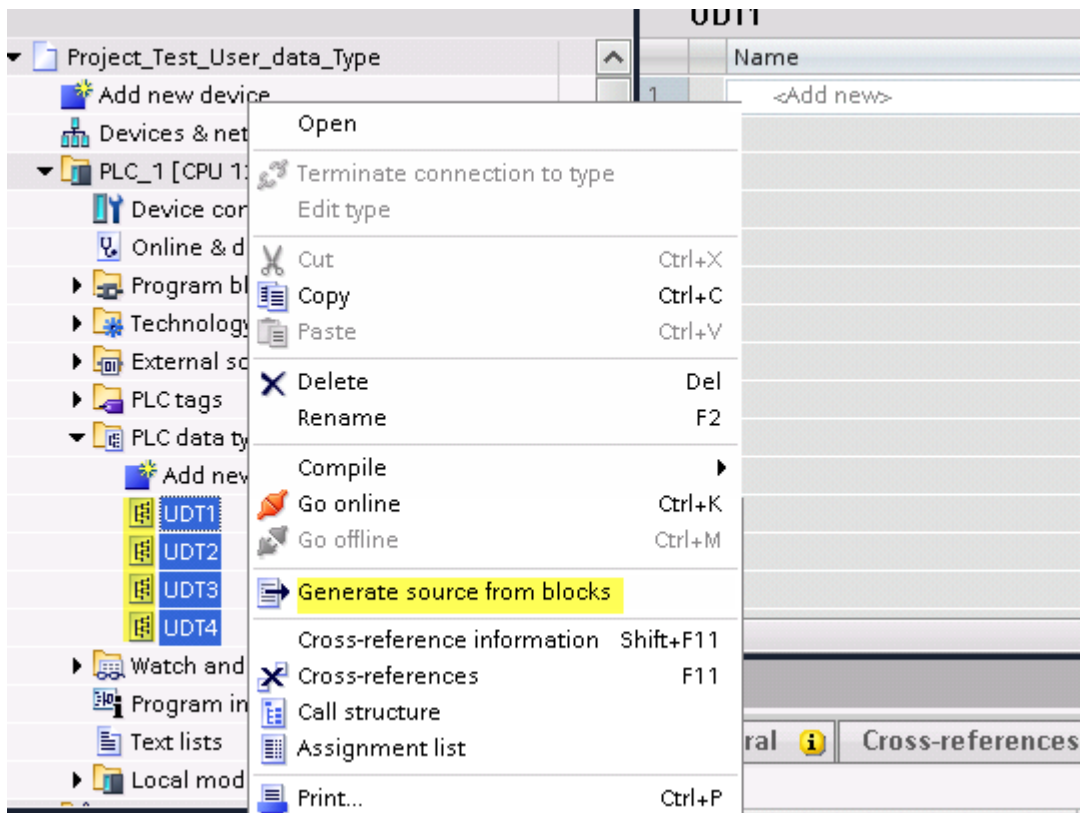


### Exporting PLC data types

To create the file, expand **PLC data types** item from TIA Portal project tree and right click on the user defined structure. Then click on **Generate source from blocks**.

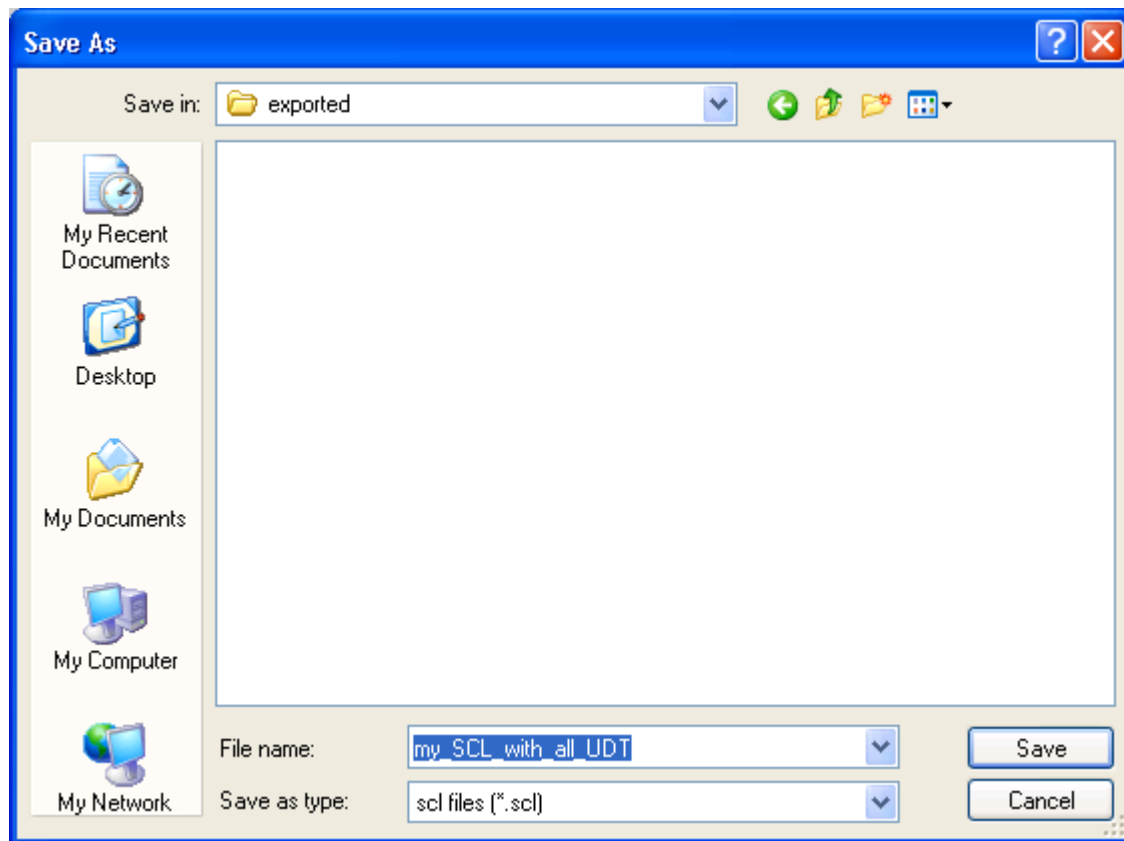


In case of multiple PLC data types in PLC project, it is necessary to select them all from **PLC data types** list, right click and select **Generate source from blocks** to create the .SCL file that contains all the PLC data types defined.



In the next step, give a name to the .SCL file and choose the path to where to save the file.





This file will content all the PLC data types and it can be used for importing tags in Tag Editor.

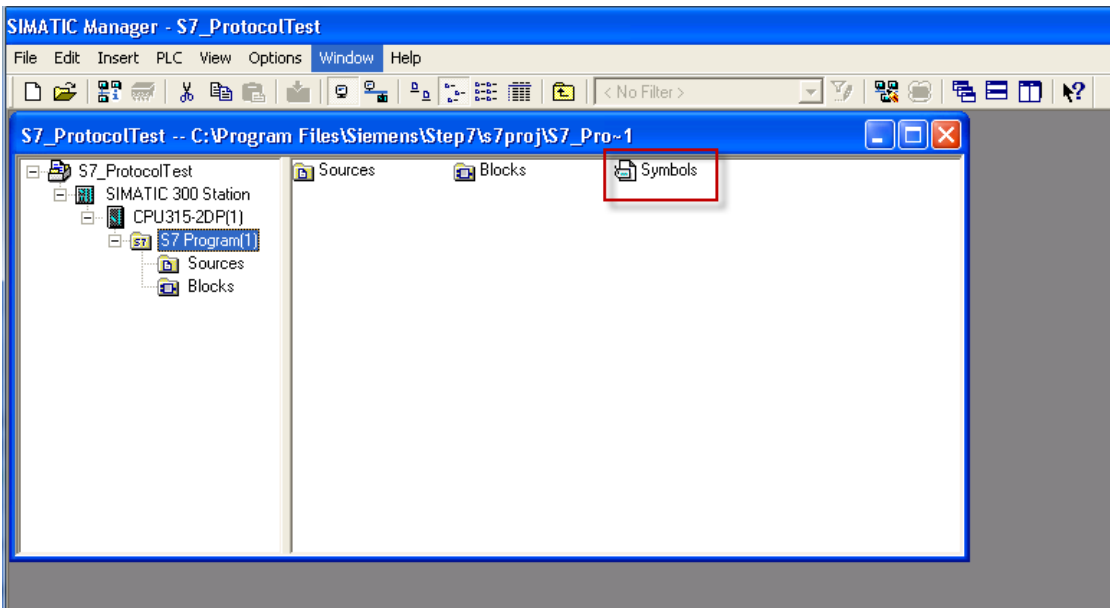
Check **Tag Import** chapter for more details.

## Export using STEP7

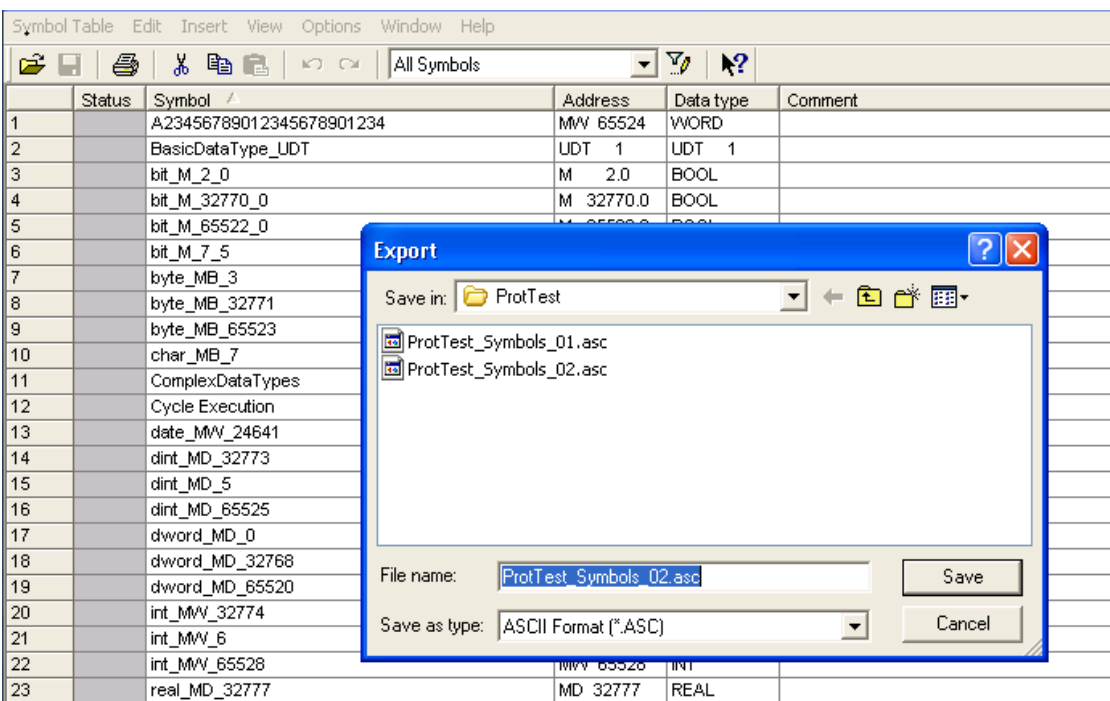
The Simatic S7 ETH Tag importer accepts symbol files (ASCII format .asc) and source files (.awl extension) created by the Simatic Step7. The symbol file can be previously exported using the Step7 symbol table utility.

### Exporting Symbols table

Symbol files (.asc) can be exported from the symbol table utility.



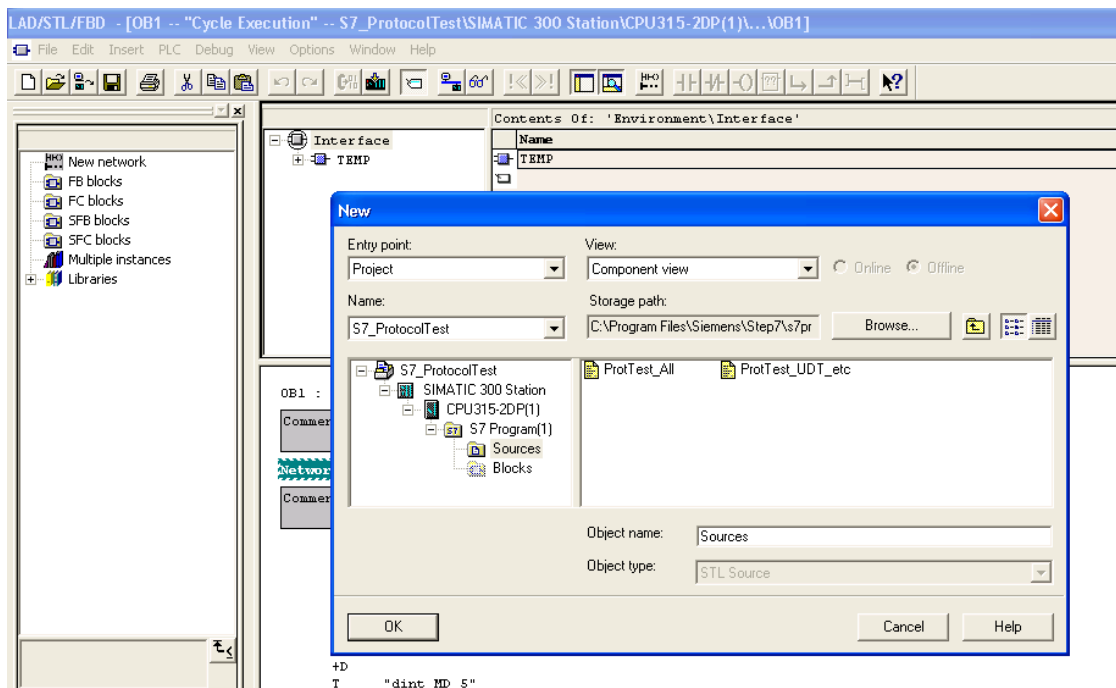
1. From the **Symbol Table** menu in the Symbol Editor choose **Export**.
2. Assign a name and save the symbol table as ASCII file.



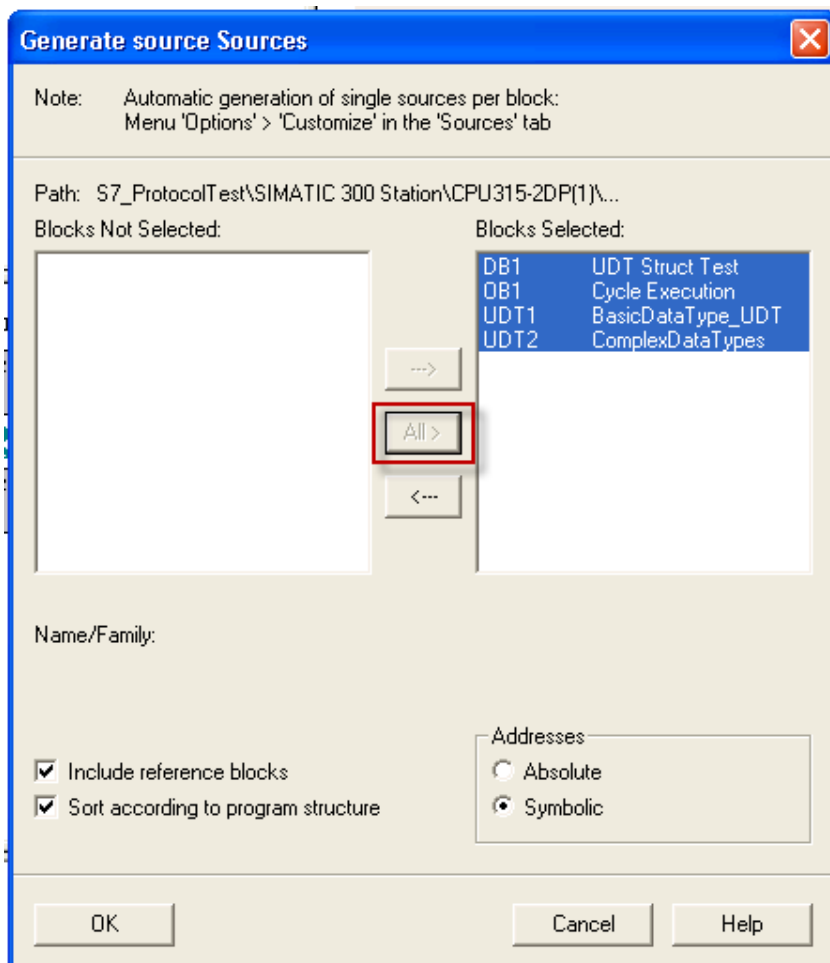
## Exporting Sources

These files are created exporting source code.

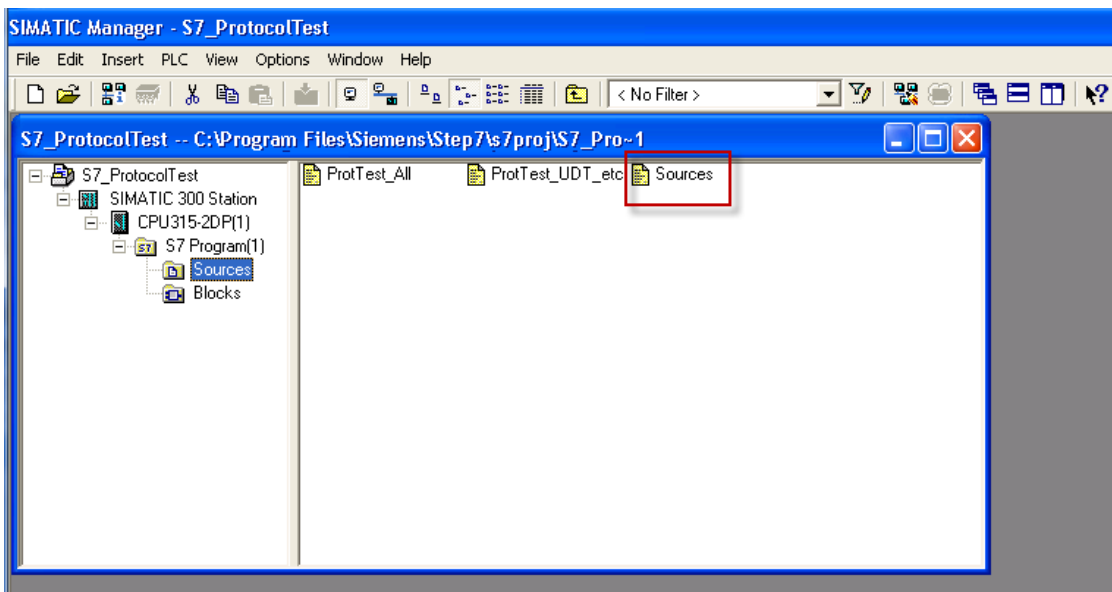
1. Open any program block in the editor, "OB1" in this example.
2. From the **File** menu choose **Generate Source**: the following dialog is displayed:



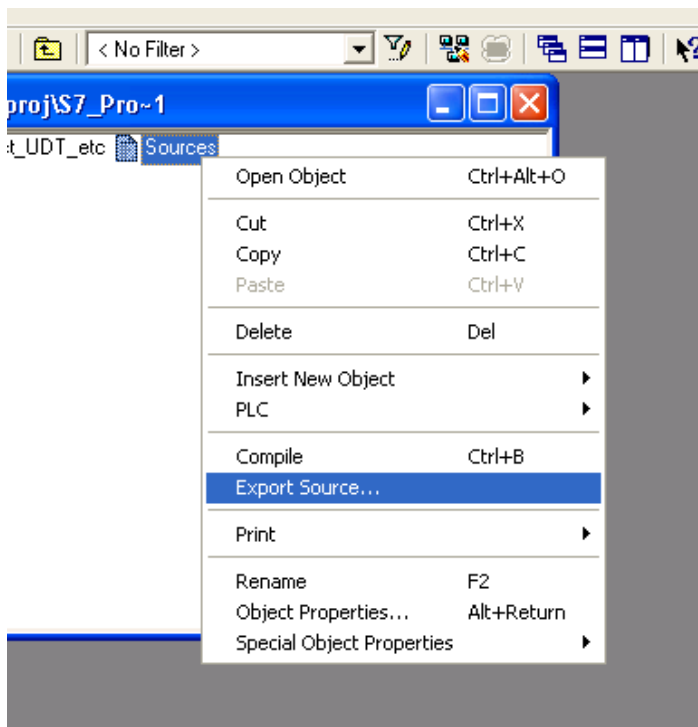
1. Assign a name, "Sources" in the example, and click **OK**: the **Generate source Sources** dialog is displayed.



2. Click **All >** to generate source for all blocks.
3. Select the following options:
  - **Include reference blocks**
  - **Sort according to program structure**
  - **Symbolic address**
4. Click **OK** to confirm: the "Sources" object is generated in the Step7 project as in the example.



5. Right click on the object and select **Export Sources**.



The generated .awl file can be imported in the Tag Editor.



Note: The .awl file contains additional information not included in the .asc file exported from the symbol table.

Make sure that reference to all data blocks is inserted in the symbol table. The tags from a data block are imported only if the symbol table contains a line with the data block name and related comment.

Status	Symbol	Address	Data type	Comment
1	CPU_FLT	OB 84	OB 84	CPU Fault
2	I/O_FLT2	OB 83	OB 83	I/O Point Fault 2
3	OBNL_FLT	OB 85	OB 85	OB Not Loaded Fault
4	Prova Data Block	DB 123	DB 123	
5	Prova MBO	MB 0	BYTE	
6	VAT_1	VAT 1		
7				

Each entry enables the import filter to import the tags related to the specified data block.

## Tag Editor Settings

In the Tag Editor select “Simatic S7 ETH” from the list of defined protocols and click + to add a tag.

Simatic S7 ETH

Simatic S7 ETH

Memory Type: Internal Memory

Offset: 0

SubIndex: 0


Data Block: 1

Data Type: unsignedByte

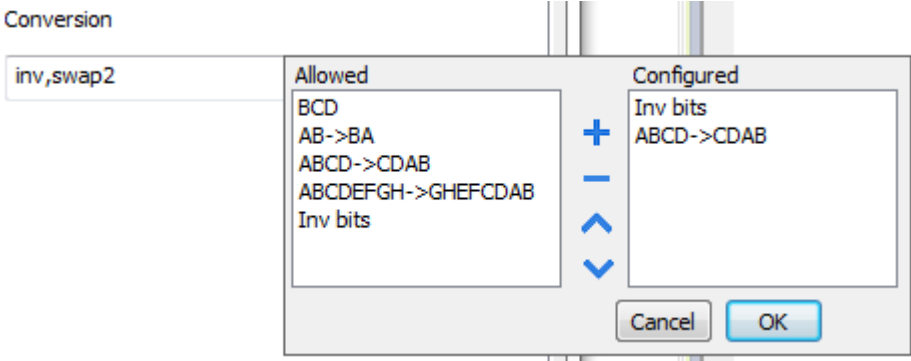
Arraysize: 0

Conversion: +/-

Buttons: OK, Cancel, Apply, Help

Element	Description														
<b>Memory Type</b>	Area of PLC where tag is located.														
	<table border="1"> <thead> <tr> <th>Data Type</th> <th>Simatic Type</th> </tr> </thead> <tbody> <tr> <td>Internal Memory</td> <td>M</td> </tr> <tr> <td>Data Block</td> <td>DB</td> </tr> <tr> <td>Input</td> <td>I (E)</td> </tr> <tr> <td>Output</td> <td>O (A)</td> </tr> <tr> <td>Timer value</td> <td>T</td> </tr> <tr> <td>Counter value</td> <td>C</td> </tr> </tbody> </table>	Data Type	Simatic Type	Internal Memory	M	Data Block	DB	Input	I (E)	Output	O (A)	Timer value	T	Counter value	C
	Data Type	Simatic Type													
	Internal Memory	M													
	Data Block	DB													
	Input	I (E)													
	Output	O (A)													
	Timer value	T													
Counter value	C														
<b>Offset</b>	Offset address where tag is located.														
<b>SubIndex</b>	Resource offset within the register.														
<b>Data Block</b>	Data block number for Data Block Memory Type.														
<b>Data Type</b>	<p>Available data types:</p> <ul style="list-style-type: none"> <li>• <b>boolean</b></li> <li>• <b>byte</b></li> <li>• <b>short</b></li> <li>• <b>int</b></li> <li>• <b>unsignedByte</b></li> <li>• <b>unsignedShort</b></li> <li>• <b>unsignedInt</b></li> <li>• <b>float</b></li> <li>• <b>string</b></li> </ul> <p>See "Programming concepts" section in the main manual.</p> <p> Note: To define arrays, select one of Data Type format followed by square brackets.</p>														

Element	Description
<b>Arraysize</b>	<ul style="list-style-type: none"> <li>In case of array tag, this property represents the number of array elements.</li> <li>In case of string tag, this property represents the maximum number of bytes available in the string tag.</li> </ul> <p>Note: number of bytes corresponds to number of string characters if Encoding property is set to UTF-8 or Latin1 in Tag Editor. If Encoding property is set to UCS-2BE, UCS-2LE, UTF-16BE or UTF-16LE one character requires 2 bytes.</p>

<b>Conversion</b>	<p>Conversion to be applied to the tag.</p>  <p>Depending on data type selected, the <b>Allowed</b> list shows one or more conversions, listed below.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td><b>Inv bits</b></td> <td>Invert all the bits of the tag. <i>Example:</i> 1001 → 0110 (in binary format) 9 → 6 (in decimal format)</td> </tr> <tr> <td><b>Negate</b></td> <td>Set the opposite of the tag value. <i>Example:</i> 25.36 → -25.36</td> </tr> <tr> <td><b>AB -&gt; BA</b></td> <td>Swap nibbles of a byte. <i>Example:</i> 15D4 → 514D (in hexadecimal format) 5588 → 20813 (in decimal format)</td> </tr> <tr> <td><b>ABCD -&gt; CDAB</b></td> <td>Swap bytes of a word. <i>Example:</i> 9ACC → CC9A (in hexadecimal format) 39628 → 52378 (in decimal format)</td> </tr> </tbody> </table>	Value	Description	<b>Inv bits</b>	Invert all the bits of the tag. <i>Example:</i> 1001 → 0110 (in binary format) 9 → 6 (in decimal format)	<b>Negate</b>	Set the opposite of the tag value. <i>Example:</i> 25.36 → -25.36	<b>AB -&gt; BA</b>	Swap nibbles of a byte. <i>Example:</i> 15D4 → 514D (in hexadecimal format) 5588 → 20813 (in decimal format)	<b>ABCD -&gt; CDAB</b>	Swap bytes of a word. <i>Example:</i> 9ACC → CC9A (in hexadecimal format) 39628 → 52378 (in decimal format)
Value	Description										
<b>Inv bits</b>	Invert all the bits of the tag. <i>Example:</i> 1001 → 0110 (in binary format) 9 → 6 (in decimal format)										
<b>Negate</b>	Set the opposite of the tag value. <i>Example:</i> 25.36 → -25.36										
<b>AB -&gt; BA</b>	Swap nibbles of a byte. <i>Example:</i> 15D4 → 514D (in hexadecimal format) 5588 → 20813 (in decimal format)										
<b>ABCD -&gt; CDAB</b>	Swap bytes of a word. <i>Example:</i> 9ACC → CC9A (in hexadecimal format) 39628 → 52378 (in decimal format)										

Element	Description	
	<b>Value</b>	<b>Description</b>
	<b>ABCDEFGH -&gt; GHEFCDAB</b>	Swap bytes of a double word.  <i>Example:</i> 32FCFF54 → 54FFFC32 (in hexadecimal format) 855441236 → 1426062386 (in decimal format)
	<b>ABC...NOP -&gt; OPM...DAB</b>	Swap bytes of a long word.  <i>Example:</i> 142.366 → -893553517.588905 (in decimal format) 0 1000000110 0001110010111011011001000101101000011100101011000001 → 1 10000011100 1010101000010100010110110110010110110000100111101 (in binary format)
	<b>BCD</b>	Separate the byte in two nibbles, and reads them as decimal (from 0 to 9)  <i>Example:</i> 23 → 17 (in decimal format) 0001 0111 = 23 0001 = 1 (first nibble) 0111 = 7 (second nibble)
	<b>S5timer(BCD)</b>	Used to support S5timer. Check <b>Simatic S5timer special data type</b> for more details.
	<b>S5timer(BIN)</b>	Legacy transformation for S5timer in binary format.
<p>Select the conversion and click on plus button. The selected item will be added on <b>Configured</b> list.</p> <p>If more conversions are configured, they will be applied in order (from top to bottom of <b>Configured</b> list).</p> <p>Use the arrow buttons to order the configured conversions.</p>		

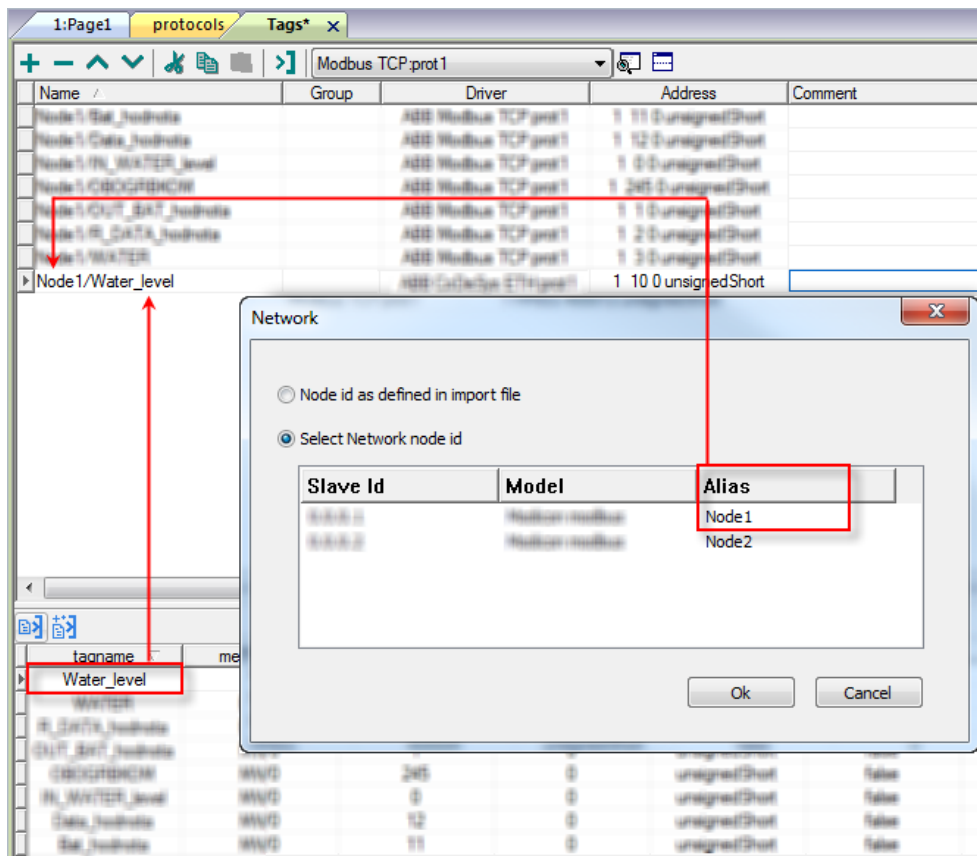
## Adding an alias name to a protocol

Tag names must be unique at project level, however, the same tag names might need to be used for different controller nodes (for example when the HMI device is connected to two devices running the same application).

When creating a protocol you can add an alias name that will be added to tag names imported for this protocol.

In the example, the connection to a certain controller is assigned the name **Node1**. When tags are imported for this node, all tag names will have the prefix **Node1** making each of them unique at the network/project level.





**i** Note: Aliasing tag names is only available for imported tags. Tags added manually in the Tag Editor cannot have the Alias prefix in the tag name. The Alias string is attached at the time of tag import. If you modify the Alias string after the tag import has been completed, there will be no effect on names already present in the dictionary. When the Alias string is changed and tags are re-imported, all tags will be re-imported with the new prefix string.

### String data type

In Simatic S7 PLC two different types of tags manage string variables:

- as Array [1..xx] of characters,
- as String[xx].

Step7 string declaration is shown in this example:

Address	Name	Type	Initial value	Comment
0.0		STRUCT		
+0.0	String1	STRING[254]	'sample'	
+256.0	String2	ARRAY[1..10]		
*1.0		CHAR		
=266.0		END_STRUCT		

S7 String

String as array of char

TIA Portal string declaration is shown in this example:

	Name	Data type	Offset	Start value	Retain	Accessible ...	Visible in ...
1	Static	Static			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2	String1	String	...	'sample'	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3	String2	Array [1 .. 10] of Char	...		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>



Note: When using String[xx] data type specific a conversion must be applied to the tag. If the tag dictionary is imported from TIA Portal or Step7 using the import tool, however, conversion of the string tags is performed automatically and no further action is required.

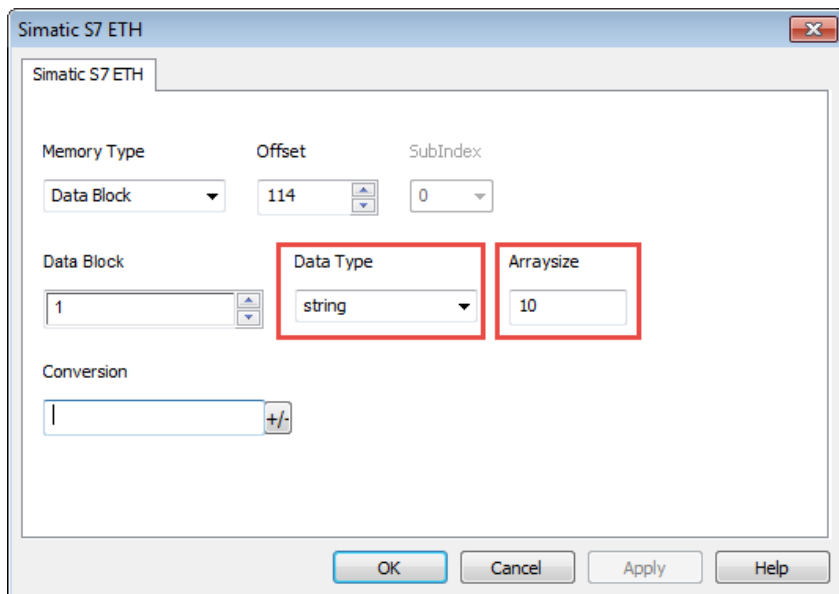
To add a string as an array of characters:

1. Press the **+** in the Tag Editor.

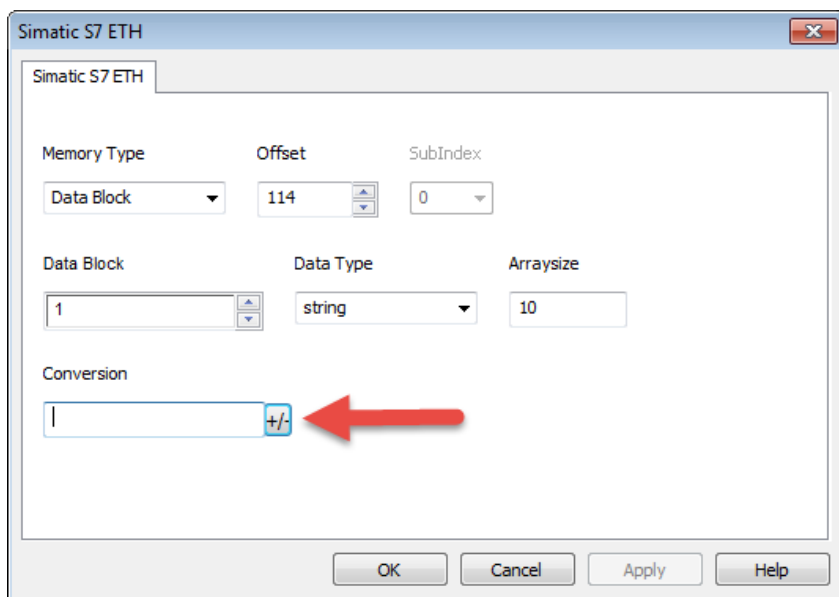
2. Select **string** as **Data Type**.
3. Enter string length in **Arraysizes**.
4. Click **OK** to confirm.

To add a string data type:

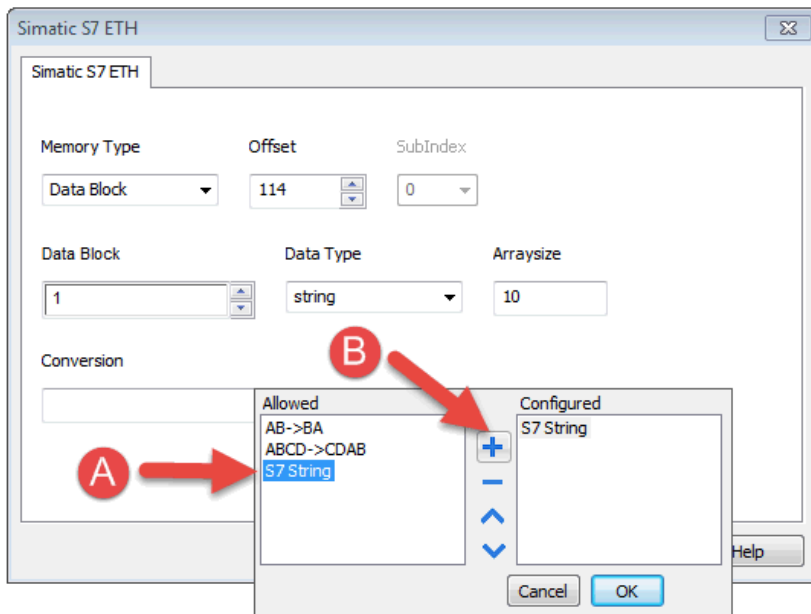
1. Press the **+** in the Tag Editor.



2. Select **string** as **Data Type**.
3. Enter string length in **Arraysize**.
4. Click **+/-** to open the Conversion dialog.



5. In the conversion dialog select the **S7 String** conversion type.



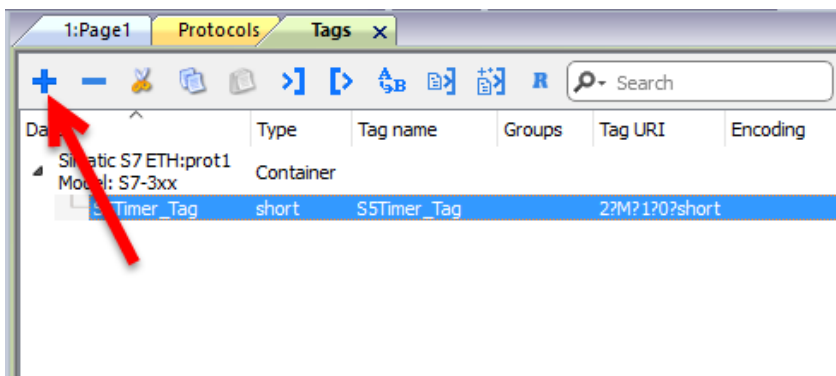
6. Click **+** to add the conversion: the conversion will be listed into the **Configured** list on the right.
7. Click **OK** to confirm.

## Simatic S5Timer data type

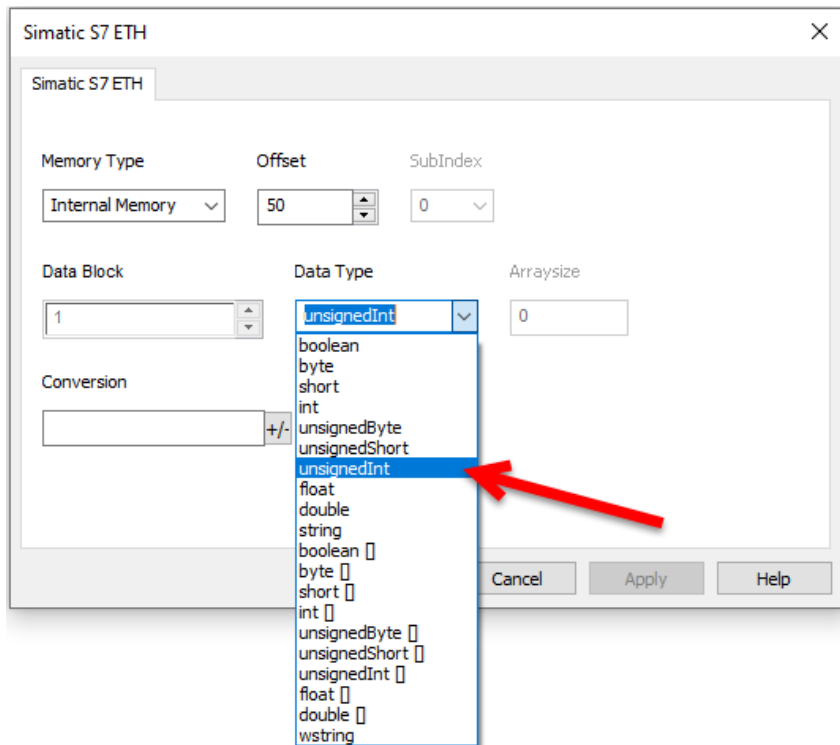
Simatic drivers support a special data type, the S5Timer data type.

The tag must be configured with a specific data type and a conversion must be applied to the tag to correctly read/write a Simatic S5Timer Variable.

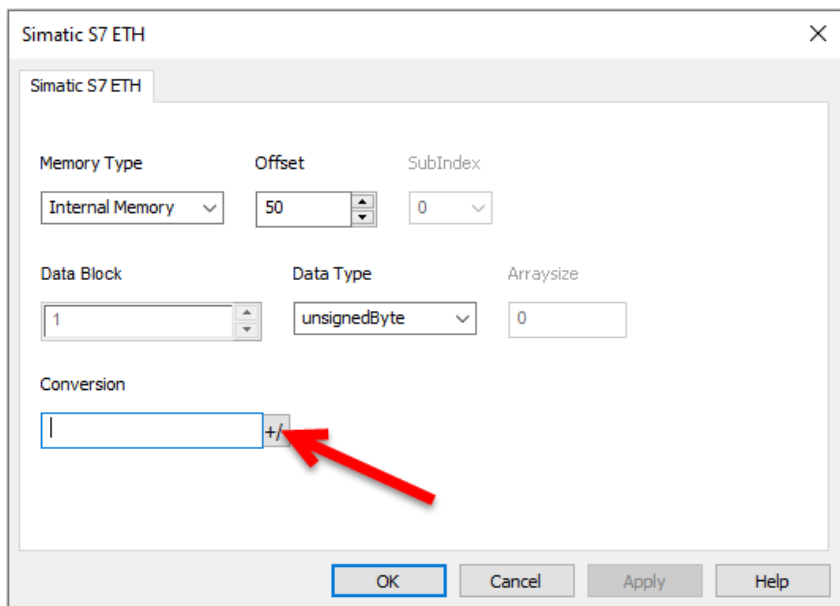
1. In the Tag Editor click **+** to add a tag.



2. Select **unsignedInt** as **Data Type**.

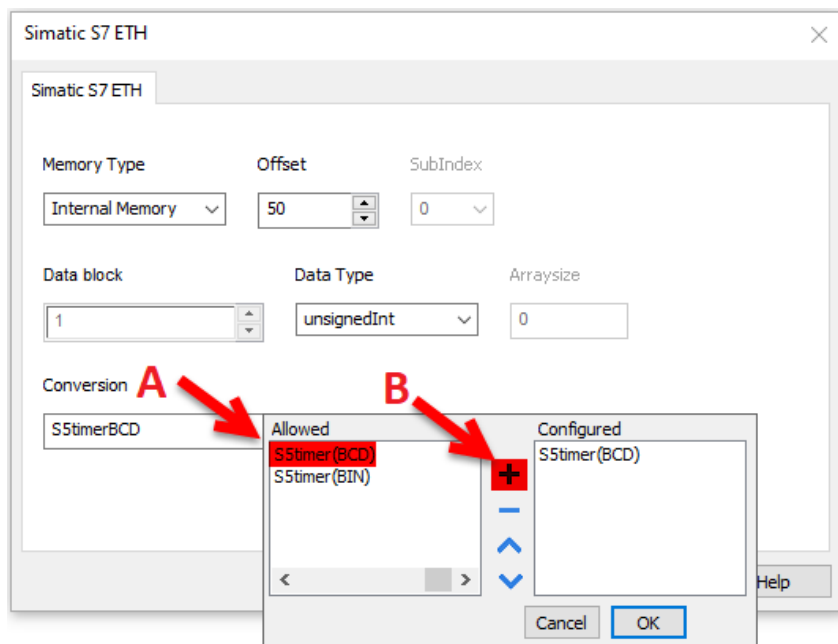


3. Click +/- to open the Conversion dialog.



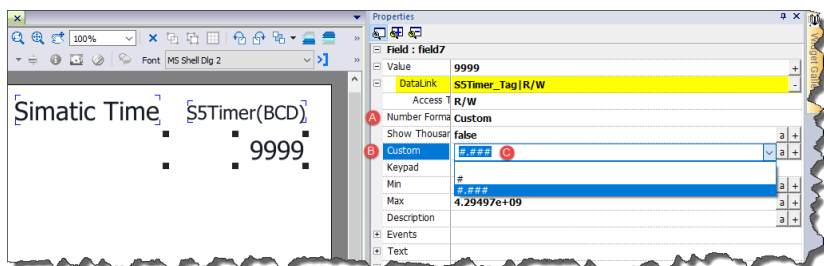
4. In the conversion dialog select the **S5timer(BCD)** conversion type.

5. Click + to add the conversion: the conversion will be listed into the **Configured** list on the right.



6. Click **OK** to confirm.

7. Define **Custom** voice in the **Number Format** property on numeric field and use **"#"** characters on **Custom** parameter to display properly the decimal part of the request value.



## Node Override IP

The protocol provides the special data type Node Override IP which allows you to change the IP address of the target controller at runtime.

This memory type is an array of 4 unsigned bytes, one per each byte of the IP address.

The Node Override IP is initialized with the value of the controller IP specified in the project at programming time.

Node Override IP	PLC operation
<b>0.0.0.0</b>	Communication with the controller is stopped, no request frames are generated anymore.
<b>Different from 0.0.0.0</b>	It is interpreted as node IP override and the target IP address is replaced runtime with the new value.

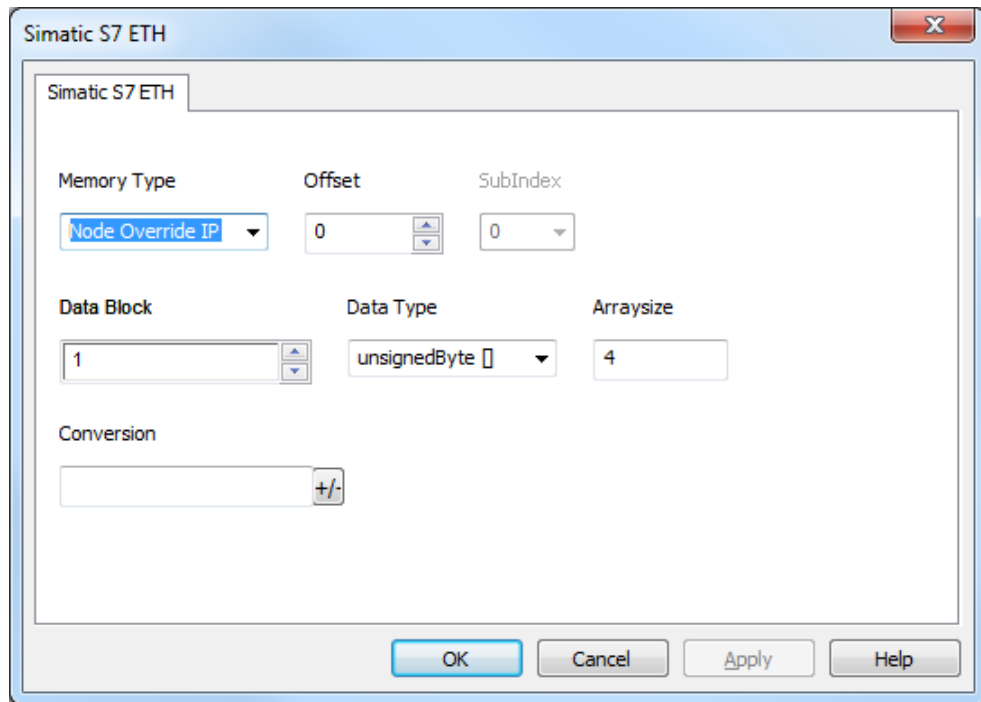
If the HMI device is connected to a network with more than one controller node, each node has its own Node Override IP variable.



Note: Node Override IP values assigned at runtime are retained through power cycles.

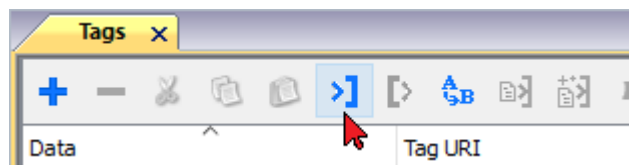
### Hostname DNS or mDNS

In addition to the array of bytes, string memory type can be selected to be able use the DNS or mDNS hostname as an alternative to the IP Address.

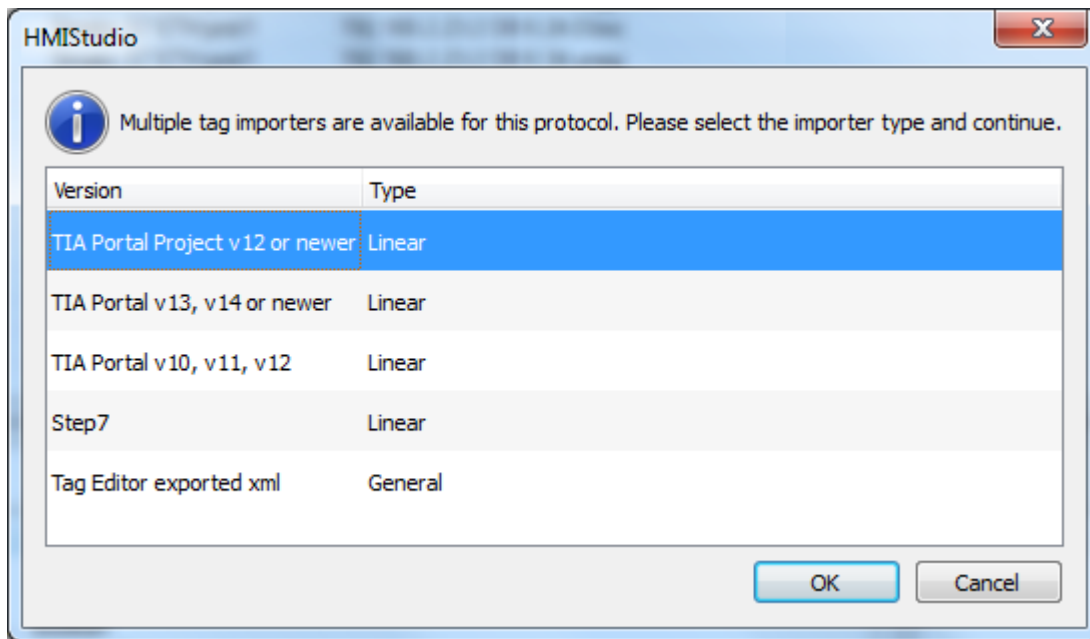


### Tag Import

Select the driver in Tag Editor and click on the **Import Tags** button to start the importer.



The following dialog shows which importer type can be selected.



Importer	Description
<b>TIA Portal Project v12 or newer</b> Linear	Allows to import the whole TIA Portal project file using <b>.apxx</b> file (where "xx" is the TIA Portal version, example: for TIA Portal 13 , file name is "project.ap13").  All variables will be displayed at the same level.
<b>TIA Portal v13, v14 or newer</b> Linear	Allows to import: <ul style="list-style-type: none"> <li>• Program blocks using <b>.db</b> file</li> <li>• PLC tags using <b>.xlsx</b> file</li> <li>• PLC data types using <b>.udt</b> file</li> </ul> Check <b>Export using TIA Portal v13, v14 or newer</b> for more details.  All variables will be displayed at the same level.
<b>TIA Portal v10, v11, v12</b> Linear	Allows to import: <ul style="list-style-type: none"> <li>• Program blocks using <b>.tia</b> file</li> <li>• PLC tags using <b>.xlsx</b> file</li> <li>• PLC data types using <b>.scl</b> file</li> </ul> Check <b>Export using TIA Portal v10, v11, v12</b> for more details.  All variables will be displayed at the same level.
<b>Step7</b> Linear	Allows to import: <ul style="list-style-type: none"> <li>• Symbols table <b>.asc</b> file</li> <li>• Sources using <b>.awl</b> file</li> </ul> Check <b>Export using STEP7</b> for more details.

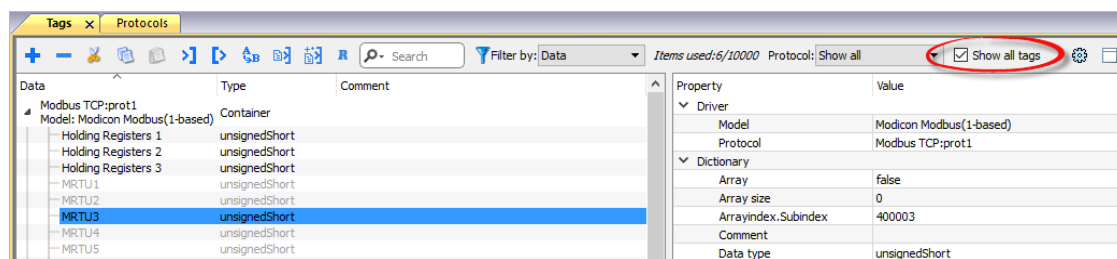




Importer	Description
	All variables will be displayed at the same level.
<b>Tag Editor exported xml</b>	Select this importer to read a generic XML file exported from Tag Editor by appropriate button.


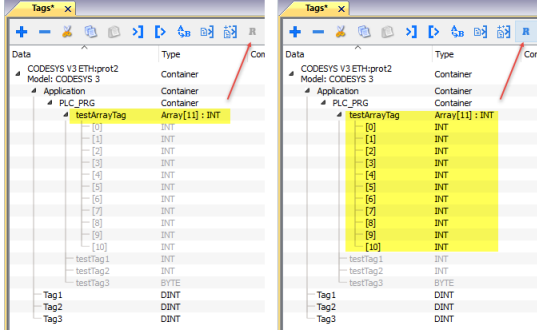
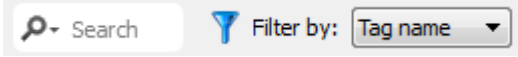


Once the importer has been selected, locate the symbol file and click **Open**.

The tags available within the Dictionary but not imported into the project are gray and are visible only when the "Show all tags" check box is selected.



Toolbar item	Description
	<p><b>Import Tag(s).</b></p> <p>Select tags to be imported and click on this icon to add tags from tag dictionary to the project</p>
	<p><b>Update Tag(s).</b></p> <p>Click on this icon to update the tags in the project, due a new dictionary import.</p>

Toolbar item	Description
	<p>Check this box to import all sub-elements of a tag.</p> <p>Example of both checked and unchecked result:</p> 
	<p>Searches tags in the dictionary basing on filter combo box item selected.</p>

## Communication status

Current communication status can be displayed using system variables. See "System Variables" section in the main manual.

Codes supported by this communication driver:

Error	Cause	Action
<b>NAK</b>	The controller replies with a not acknowledge.	-
<b>Timeout</b>	A request is not replied within the specified timeout period.	Check if the controller is connected and properly configured to get network access.
<b>Invalid response</b>	The device did received a response with invalid format or contents from the controller .	Ensure the data programmed in the project are consistent with the controller resources.
<b>General Error</b>	Unidentifiable error. Should never be reported.	Contact technical support.

# System Variables

System Variables communication driver allows to create Tags that point to system information.

Refer to [System Variables > Protocol](#) chapter of User's Manual.

---

## Protocol Editor Settings

System Variables communication driver allows to create Tags that point to system information.

Refer to [System Variables > Protocol](#) chapter of User's Manual.

# Variables

Variables communication driver allows to define Tags which points to HMI internal memory.

Variables Tags are not retentive: when the project starts, the starting value of any Variables Tag is 0 (or "" in case of string Tag).



Variables communication driver is not counted as physical protocol.  
Refer to **Table of functions and limits** from main manual in "Number of physical protocols" line.

## Protocol Editor Settings

### Adding a protocol

To configure the protocol:


1. In the **Config** node double-click **Protocols**.
2. To add a driver, click **+**: a new line is added.
3. Select the **Variables** protocol from the **PLC** list.

## Tag Editor Settings

Path: **ProjectView** > **Config** > double-click **Tags**

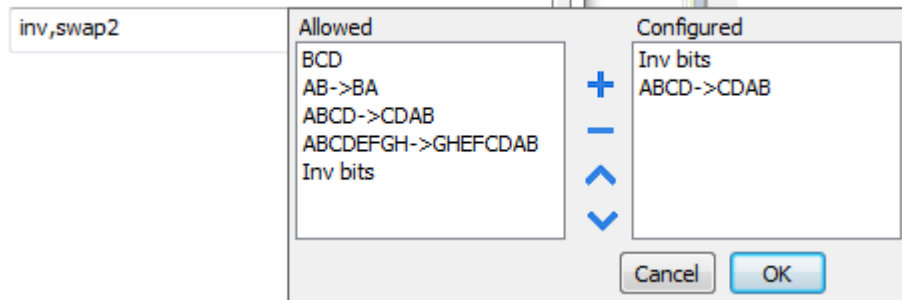
1. To add a tag, click **+**: a new line is added.
2. Select **Variables** from the protocol list: tag definition dialog is displayed.

The screenshot shows a dialog box titled "Variables". It contains a tab labeled "Variables". Below the tab, there are three input fields: "Data type" (a dropdown menu showing "boolean"), "Arraysize" (a text box containing "0"), and "Conversion" (a text box with a "+/-" button). At the bottom of the dialog, there are four buttons: "OK", "Cancel", "Apply", and "Help".

Element	Description		
<b>Data Type</b>	<b>Data Type</b>	<b>Memory Space</b>	<b>Limits</b>
	<b>boolean</b>	1-bit data	0 ... 1
	<b>byte</b>	8-bit data	-128 ... 127
	<b>short</b>	16-bit data	-32768 ... 32767
	<b>int</b>	32-bit data	-2.1e9 ... 2.1e9
	<b>int64</b>	64-bit data	-9.2e18 ... 9.2e18
	<b>unsignedByte</b>	8-bit data	0 ... 255
	<b>unsignedShort</b>	16-bit data	0 ... 65535
	<b>unsignedInt</b>	32-bit data	0 ... 4.2e9
	<b>uint64</b>	64-bit data	0 ... 1.8e19
	<b>float</b>	IEEE single-precision 32-bit floating point type	1.17e-38 ... 3.4e38
	<b>double</b>	IEEE double-precision 64-bit floating point type	2.2e-308 ... 1.79e308
	<b>string</b>	Array of elements containing character code defined by selected encoding	
<b>binary</b>	Arbitrary binary data		
	 Note: to define arrays, select one of Data Type format followed by square brackets like "byte[]", "short[]"...		
<b>Arraysizes</b>	<ul style="list-style-type: none"> <li>• In case of array tag, this property represents the number of array elements.</li> <li>• In case of string tag, this property represents the maximum number of bytes available in the string tag.</li> </ul> <p>Note: number of bytes corresponds to number of string characters if Encoding property is set to UTF-8 or Latin1 in Tag Editor. If Encoding property is set to UCS-2BE, UCS-2LE, UTF-16BE or UTF-16LE one character requires 2 bytes.</p>		
<b>Conversion</b>	Conversion to be applied to the tag.		

Element	Description
---------	-------------

Conversion



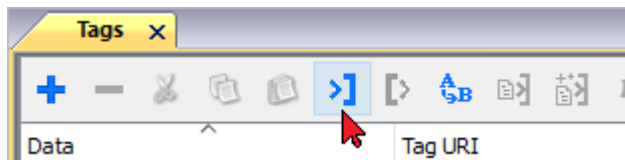
Depending on data type selected, the list **Allowed** shows one or more conversion types.

Value	Description
<b>Inv bits</b>	<p><b>inv:</b> Invert all the bits of the tag.</p> <p><i>Example:</i>            1001 → 0110 (in binary format)            9 → 6 (in decimal format)</p>
<b>Negate</b>	<p><b>neg:</b> Set the opposite of tag value.</p> <p><i>Example:</i>            25.36 → -25.36</p>
<b>AB -&gt; BA</b>	<p><b>swapnibbles:</b> Swap nibbles in a byte.</p> <p><i>Example:</i>            15D4 → 514D (in hexadecimal format)            5588 → 20813 (in decimal format)</p>
<b>ABCD -&gt; CDAB</b>	<p><b>swap2:</b> Swap bytes in a word.</p> <p><i>Example:</i>            9ACC → CC9A (in hexadecimal format)            39628 → 52378 (in decimal format)</p>
<b>ABCDEFGH -&gt; GHEFCDAB</b>	<p><b>swap4:</b> Swap bytes in a double word.</p> <p><i>Example:</i>            32FCFF54 → 54FFFC32 (in hexadecimal format)            855441236 → 1426062386 (in decimal format)</p>
<b>ABC...NOP -&gt; OPM...DAB</b>	<p><b>swap8:</b> Swap bytes in a long word.</p> <p><i>Example:</i>            142.366 → -893553517.588905 (in decimal format)            0 1000000110            000111001011101101100100010110100001110010101100</p>

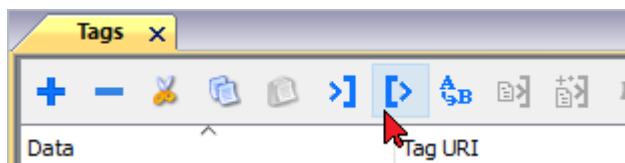
Element	Description	
	<b>Value</b>	<b>Description</b>
		0001 → 1 10000011100 101010100001010001011011011001011011000010011 1101 (in binary format)
	<b>BCD</b>	<b>bcd</b> : Separate byte in two nibbles, read them as decimal (from 0 to 9)  <i>Example:</i> 23 → 17 (in decimal format) 0001 0111 = 23 0001 = 1 (first nibble) 0111 = 7 (second nibble)
	Select conversion and click +. The selected item will be added to list <b>Configured</b> .  If more conversions are configured, they will be applied in order (from top to bottom of list <b>Configured</b> ).  Use the arrow buttons to order the configured conversions.	

## Tag Import

Select the driver in Tag Editor and click on the **Import Tags** button to start the importer.



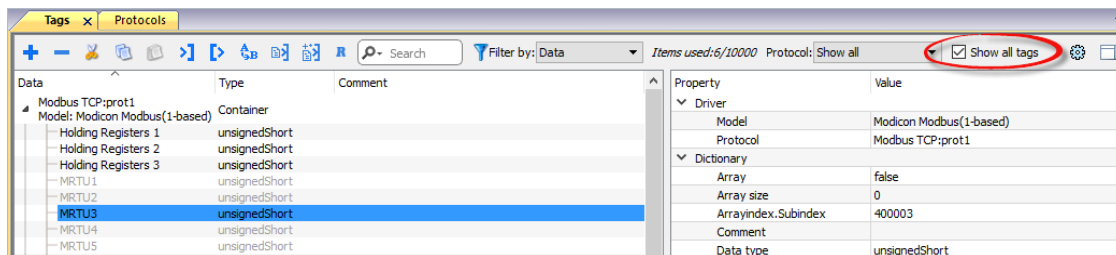
The system will require a generic XML file exported from Tag Editor by appropriate button.




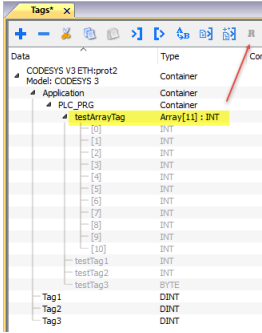
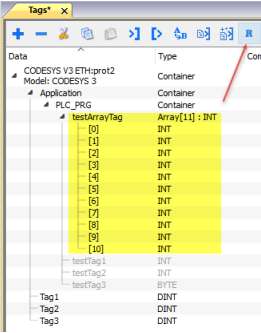
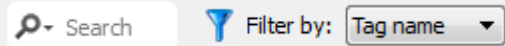


Once the importer has been selected, locate the symbol file and click **Open**.

The tags available within the Dictionary but not imported into the project are gray and are visible only when the "Show all tags" check box is selected.





Toolbar item	Description
	<p><b>Import Tag(s).</b></p> <p>Select tags to be imported and click on this icon to add tags from tag dictionary to the project</p>
	<p><b>Update Tag(s).</b></p> <p>Click on this icon to update the tags in the project, due a new dictionary import.</p>
	<p>Check this box to import all sub-elements of a tag.</p> <p>Example of both checked and unchecked result:</p> <div style="display: flex; justify-content: space-around;">   </div>
	<p>Searches tags in the dictionary basing on filter combobox item selected.</p>



—  
**ABB AG**

**[new.abb.com/plc](https://new.abb.com/plc)**  
**[new.abb.com/plc/automationbuilder](https://new.abb.com/plc/automationbuilder)**  
**[new.abb.com/contact-centers](https://new.abb.com/contact-centers)**

**Note:**

We reserve the right to make technical changes or modify the contents of this document without prior notice.

With regard to purchase orders, the agreed particulars shall prevail.

ABB AG does not accept any responsibility whatsoever for potential errors or possible lack of information in this document.

We reserve all rights in this document and in the subject matter and illustrations contained therein.

Any reproduction, disclosure to third parties or utilization of its contents – in whole or in parts – is forbidden without prior written consent of ABB AG.

© Copyright 2011 - 2023 ABB.  
All rights reserved.