USER'S MANUAL

Advanced Graphical Interface
AGI 300/400 series

DEIF A/S · Frisenborgvej 33 · DK-7800 Skive
Tel.: +45 9614 9614 · Fax: +45 9614 9615
info@deif.com · www.deif.com

Document no.: 4189341122B

Subject to change without notice

The information contained in this document is provided for informational purposes only. While efforts were made to verify the accuracy of the information contained in this documentation, it is provided 'as is' without warranty of any kind.

Third-party brands and names are the property of their respective owners.

The example companies, organizations, products, domain names, e-mail addresses, logo, people, places, and events depicted herein are fictitious. No association with any real company, organization, product, domain name, e-mail address, logo, person, place or event is intended or should be inferred.

Contents

1 Getting started

AGI Creator is a software application designed to create graphical HMI pages. AGI Creator 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 AGI Creator.

This document is divided into chapters that describe the key functions of AGI Creator 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.

Assumptions

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

Installing the application

AGI SW Pack installation contains:

- AGI Creator: 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.
- AGI 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 AGI Creator.
- AGI PC Runtime: a standalone application that runs on Win32/Win64 platforms (computers instead of HMI devices).

AGI Creator system requirements

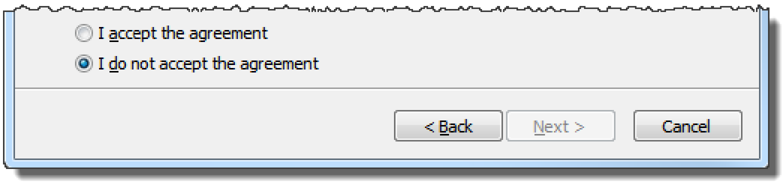
AGI Creator has the following system requirements:

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

Installation procedure

To install AGI SW Pack:

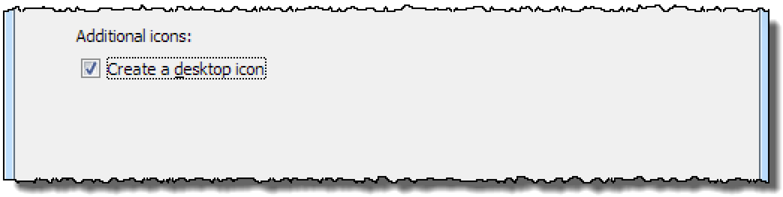
1. Run AGI SW Pack setup and click **Next.**
2. Read the AGI Creator Software License and accept the agreement.



○ I accept the agreement

● I do not accept the agreement

3. Follow the instructions on the screen. The default location for the c software is *C:\Program Files\DEIF\AGI Software Pack*, change path if needed.
4. If the Select Components step is available, select the components you want to install.
5. Select the **Create a desktop icon** option to add a AGI Creator icon on your desktop. A AGI SW Pack group is automatically added to the **Start** menu by the installation procedure.



Additional icons:

☑ Create a desktop icon

6. To run the application click the desktop icon or choose **Start** > **All programs** > **AGI Creator**.

Trial version

AGI Creator is available with a friendly 30 days free trial policy. 30 days after installation a registration form is displayed to enter a license activation key.

Note: Trial version is not supported on virtual machines, only valid licenses can be used.

Licensing

To register the software before the trial period expires, go to **Help** > **Register**.

Note: The registration process requires an Internet connection. Ports TCP 80 and 443 are used for activation.

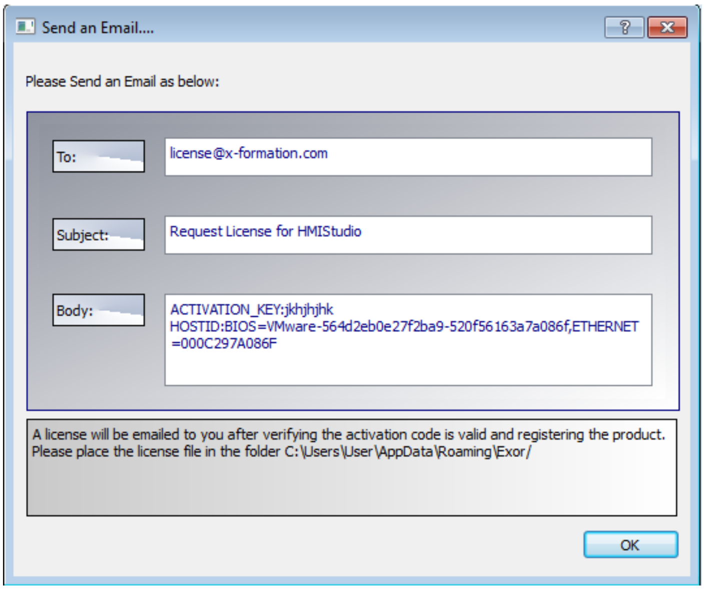
During registration, a license file is downloaded from the licensing server to the computer. License files are saved in following folders depending on OS:

%appdata%\DEIF A/S

Licenses are locked to the **BIOS ID** or to the **Windows product ID** of the computer where the software has been installed.

If AGI Creator is not able to reach the licensing server (for example, no Internet connection is available), a button is displayed to activate the license via email.

Pressing the “Send Mail” button the AGI Creator will display this form:



Send an Email....

Please Send an Email as below:

To: license@x-formation.com

Subject: Request License for HMIStudio

Body: ACTIVATION_KEY:jkhjhjhk
HOSTID:BIOS=VMware-564d2eb0e27f2ba9-520f56163a7a086f,ETHERNET
=000C297A086F

A license will be emailed to you after verifying the activation code is valid and registering the product.
Please place the license file in the folder C:\Users\User\AppData\Roaming\Exor/

This email can be send in a second moment when internet connection will available. You may also activate the licenses and download the licenses file from the web site https://licenses.x-formation.com/licenses. reporting the same data contained in the “Body” of upper form.

Verifying license status

To check the status of your license:

1. Go to: *https://license.x-formation.com/licenses*
2. Enter your activation key and click the **Log In** button.

Installing multiple versions of AGI SW Pack

You may install different instances of AGI SW Pack on the same computer. Each installation has its own settings and can be uninstalled individually.

Three installation scenarios are possible:

Installation scenario	Results
First installation of AGI SW Pack in the system	Software is installed in the specified destination folder
System with only one instance of AGI SW Pack already installed	Current version can be replaced or maintained.
System with multiple instances of AGI SW Pack already installed	Last version installed can be replaced or maintained.

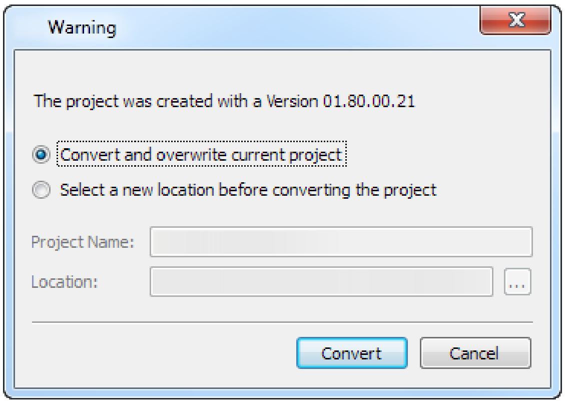
If you try to install a second instance of an already installed version of AGI SW Pack, a warning message is displayed.

Multiple AGI SW Pack installations share a common workspace folder, each sub-folder includes the version number, for example *C:\Program Files\DEIF\AGI Software Pack 2.5*. 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 AGI Creator project (.jpr file) created with an older version of the software AGI Creator asks to convert the project to the current version:


Warning

The project was created with a Version 01.80.00.21

● Convert and overwrite current project

○ Select a new location before converting the project

Project Name:

Location:

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

WARNING: Do not edit projects with a version of AGI Creator older than the version used to create them. This will damage the project and may cause runtime instability.

Multilanguage for AGI Creator

AGI Creator is available in multiple languages. All languages are installed by default as part of AGI SW Pack.

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

Crash reports

A crash report dialog appears whenever AGI Creator freezes or crashes.

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

Note: Crash reports are unavailable in Windows XP.

2 Runtime

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

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 103 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

On the HMI device press and hold on an empty area of the screen for a few seconds to display the context menu.

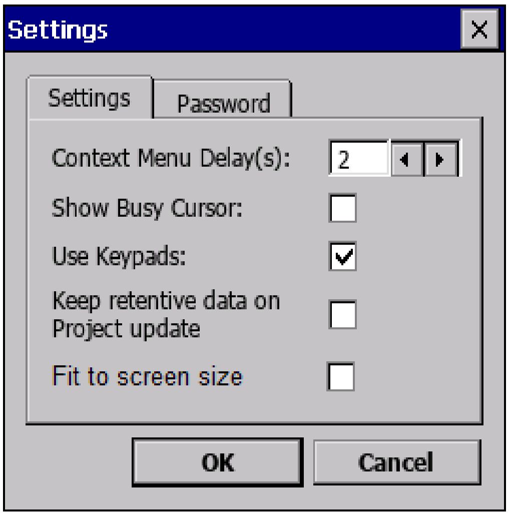
Zoom In/Out

Select view size at runtime

Pan Mode

Enables/disables pan mode after a zoom in

Settings



Settings

Context Menu Delay(s): 2

Show Busy Cursor: □

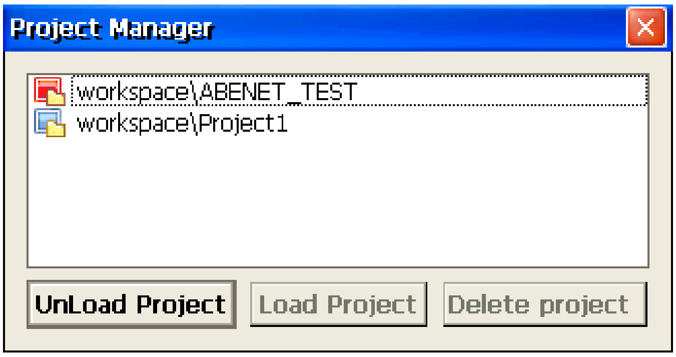
Use Keypads: ☑

Keep retentive data on Project update □

Fit to screen size □

Main parameters	Description
Context Menu Delay (s)	Context menu activation delay. Range: 1–60 seconds.
Show Busy Cursor	Display an hourglass when the system is busy
Use keypads	Display keypads when user touches a data entry field. Set to **disable** when an external USB keyboard is connected to the device.
Keep retentive data on project update	Preserve the content of the retentive data at project download or update.
Fit to screen size	Adapts the view to the screen size
Password	Define password protected operations amongst the following: • Download Project/Runtime • Upload project • Board management (BSP Update) See "Protecting access to HMI devices" on page 529 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

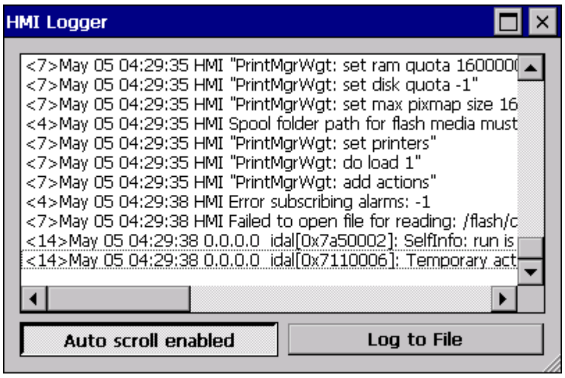
This function loads update packages from an external USB drive. See "Update package" on page 101 for details.

Backup

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

Logging

This function displays a log of system operations.



```
<7>May 05 04:29:35 HMI "PrintMgrWgt: set ram quota 1600000
<7>May 05 04:29:35 HMI "PrintMgrWgt: set disk quota -1"
<7>May 05 04:29:35 HMI "PrintMgrWgt: set max pixmap size 16
<4>May 05 04:29:35 HMI Spool folder path for flash media must
<7>May 05 04:29:35 HMI "PrintMgrWgt: set printers"
<7>May 05 04:29:35 HMI "PrintMgrWgt: do load 1"
<7>May 05 04:29:35 HMI "PrintMgrWgt: add actions"
<4>May 05 04:29:38 HMI Error subscribing alarms: -1
<7>May 05 04:29:38 HMI Failed to open file for reading: /flash/c
<14>May 05 04:29:38 0.0.0.0  idal[0x7a50002]: SelfInfo: run is
<14>May 05 04:29:38 0.0.0.0  idal[0x7110006]: Temporary act
```

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" on page 547 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 82 for details)

About

This function shows information about the Runtime version.

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

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 547 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

AGI 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.

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

AGI PC Runtime system requirements

AGI PC Runtime as the following minimum system requirements:

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

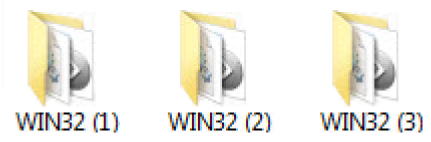
Note: HMI Runtime is not supported on virtual machines.

Installing Runtime

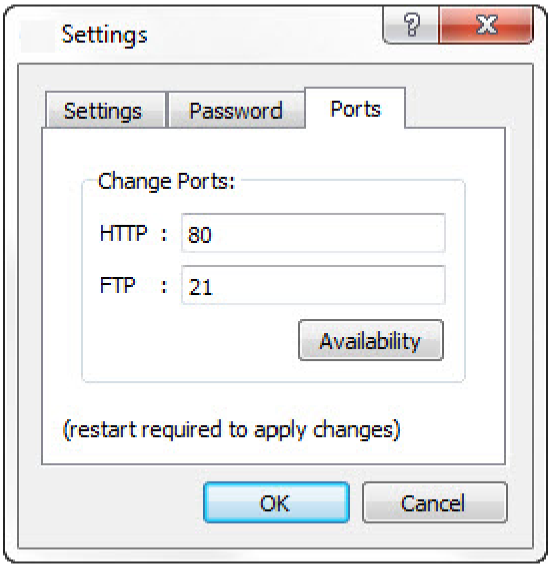
AGI PC Runtime could be distributed as a component of the AGI SW Pack or as a standalone application. When installing the software make sure that you select the **Runtime PC** option in the **Select Components** dialog.

Multiple instances of AGI PC Runtime

AGI 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 AGI PC Runtime can be changed from the **Settings** dialog. Restart the application to apply the port change.


Settings

Change Ports:

HTTP : 80

FTP : 21

(restart required to apply changes)

Trial version

AGI PC Runtime is available with a friendly 30 days free trial policy. 30 days after installation a registration form is displayed to enter license activation keys.

Note: Trial version is not supported on virtual machines, only valid licenses can be used.

Licensing

To register AGI PC Runtime before the trial period expires, from the context menu choose **Register**.

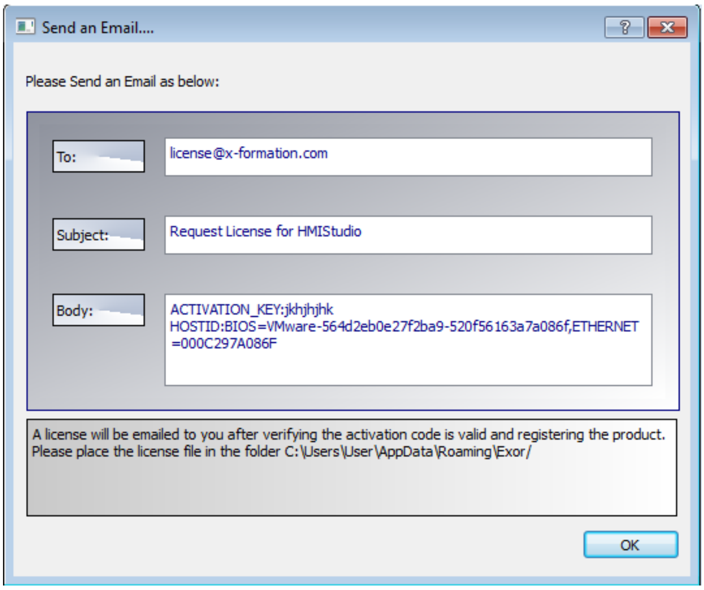
Note: the registration process requires an Internet connection. Ports TCP 80 and 443 are used for activation.

On registration, a license file is downloaded from the License Server to the computer. License files are saved in following folders depending on OS:

%appdata%\DEIF A/S

Licenses are locked to the **BIOS ID** or to the **Windows product ID** of the computer where the software is installed.

If AGI PC Runtime is not able to reach the server (for example, no Internet connection is available), a button is displayed to activate the license via email.



Please Send an Email as below:

To: license@x-formation.com

Subject: Request License for HMIStudio

Body: ACTIVATION_KEY:jkhjhjhk
HOSTID:BIOS=VMware-564d2eb0e27f2ba9-520f56163a7a086f,ETHERNET
=000C297A086F

A license will be emailed to you after verifying the activation code is valid and registering the product.
Please place the license file in the folder C:\Users\User\AppData\Roaming\Exor/

See "Installing the application" on page 2 for instructions on how to verify the activation status.

Limitations

The following features are not supported in AGI 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 541 for more details.

Fullscreen mode

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

To switch to full screen:

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

The workspace folder

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

%appdata%\DEIF A/S\[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.

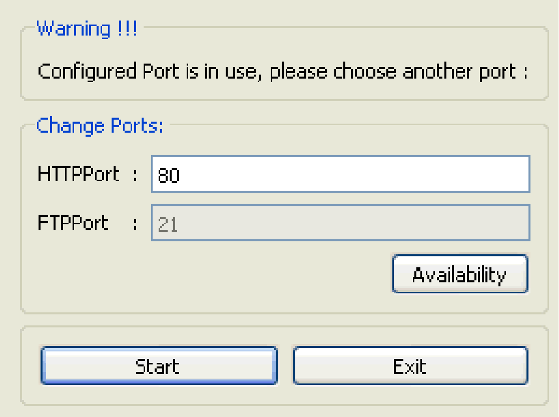
See "Protecting access to HMI devices" on page 529 for details.

In some conditions AGI PC Runtime cannot detect all services running in ports like 80/HTTP and 21/FTP, this forces AGI 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 AGI PC Runtime fails, try one of the following procedures.

Issues with port numbers

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



Warning !!!

Configured Port is in use, please choose another port :

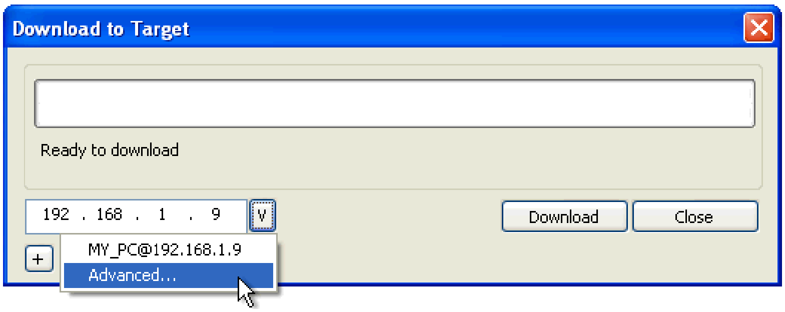
Change Ports:

HTTPPort : 80

FTPPort : 21

Make sure that when you change this port you also change the port used for download to HMI device in AGI Creator.

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

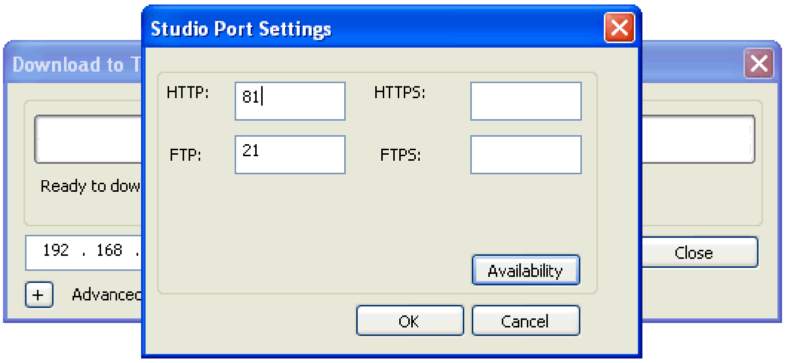


Download to Target

Ready to download

192 . 168 . 1 . 9

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



Studio Port Settings

HTTP: 81

HTTPS:

FTP: 21

FTPS:

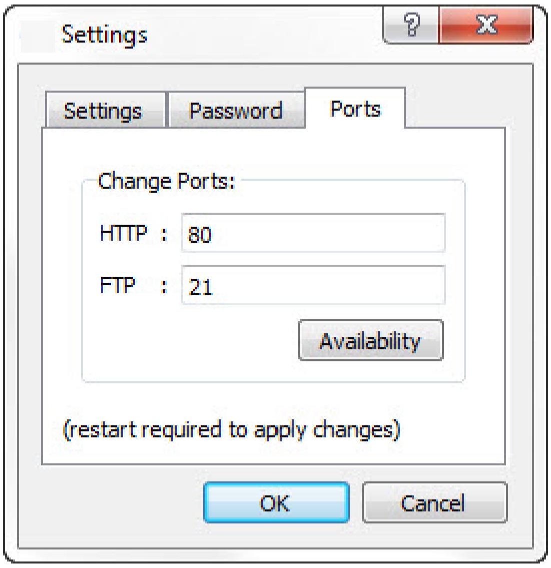
3. Click **OK** to confirm: you can now download you project to the AGI PC Runtime.

Restoring port information

If information about changes made on AGI 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 AGI PC Runtime can be changed from the **Settings** dialog. Restart the application to apply the port change.



Change Ports:

HTTP : 80

FTP : 21

(restart required to apply changes)

Bypassing firewall or antivirus blocks

If AGI Creator is running on the same machine as the AGI PC Runtime, your firewall or antivirus may block the connection from AGI Creator to AGI PC Runtime.

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

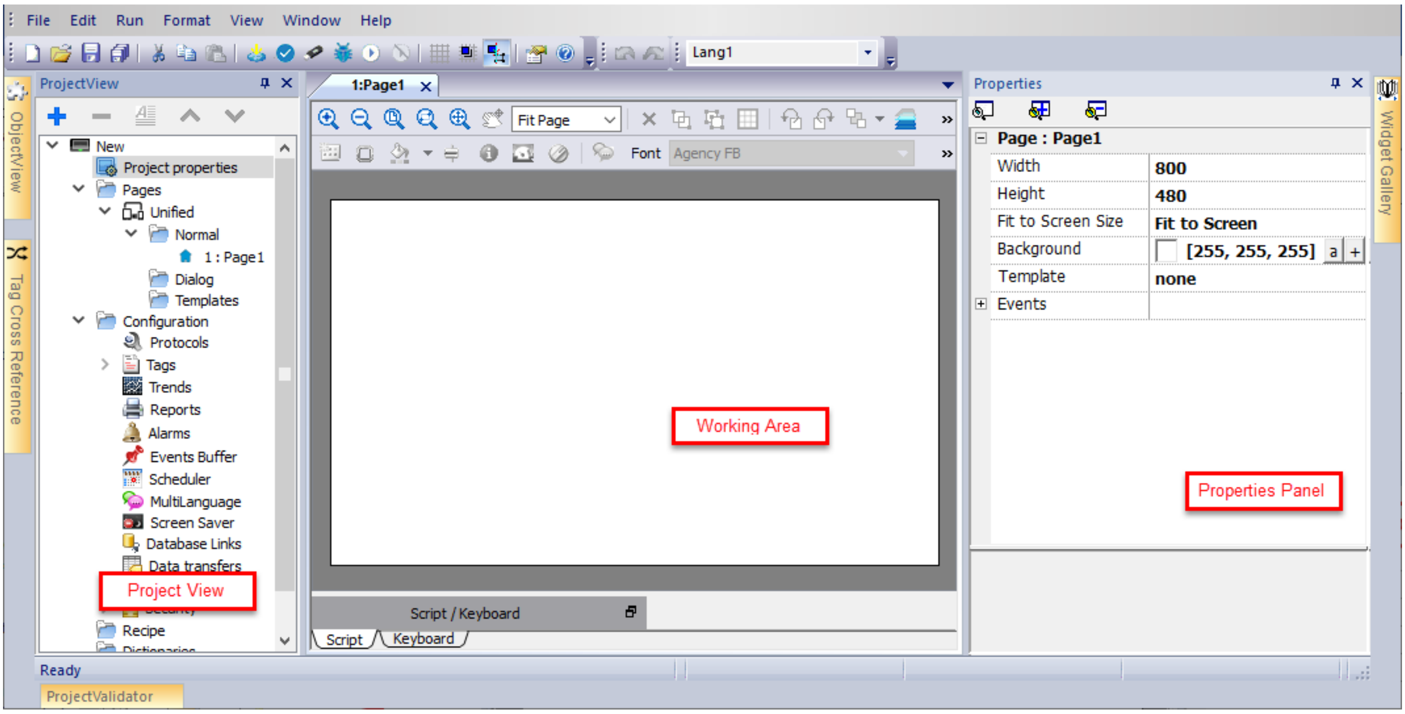
4 My first project

This section describes how to create a simple AGI Creator project.

The workspace

Workspace areas

AGI Creator workspace is divided into the following main areas:



Page : Page1

Width	800
Height	480
Fit to Screen Size	Fit to Screen
Background	[255, 255, 255]
Template	none
Events	

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

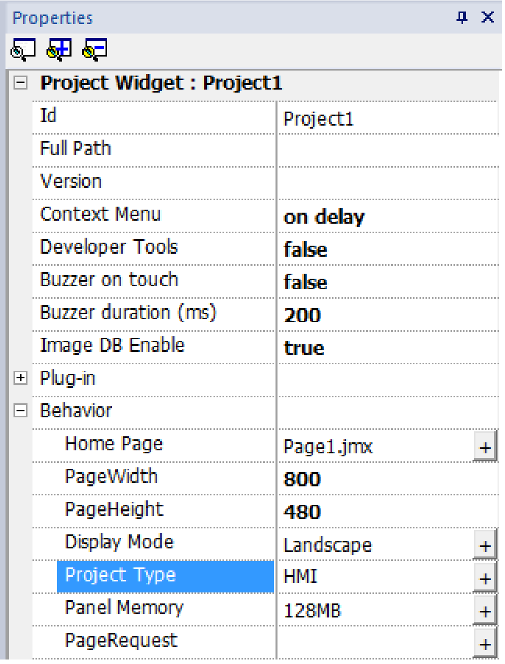
The following elements are not rotated in portrait mode.

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

HMI devices based on Linux platform can be rotated from the BSP (see "*Displays*" tab from the "System Settings" on page 547 and "Linux Devices" on page 548) without these limitations.

Changing the device model

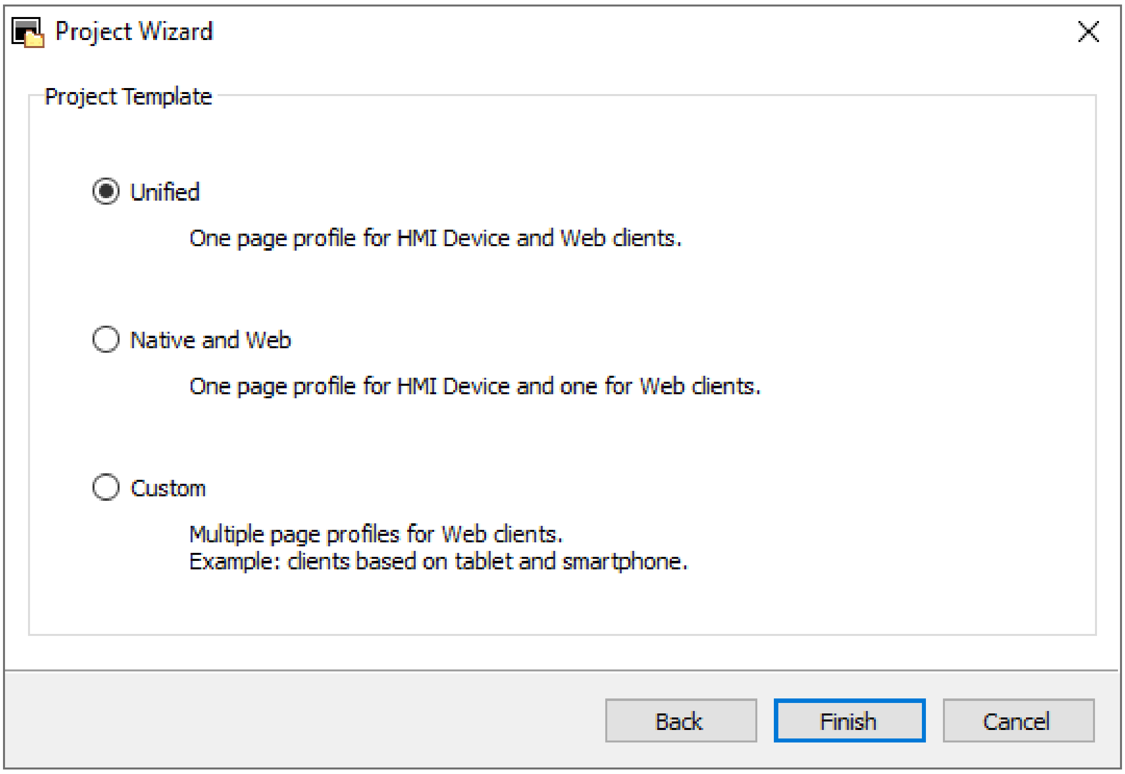
Once you have developed your project you can still change the device model, from the Project Properties pane. This will not resize the widgets, but will relocate them on the screen. A warning will be displayed if some objects cannot be relocated.



Properties

Project Widget : Project1	
Id	Project1
Full Path	
Version	
Context Menu	**on delay**
Developer Tools	**false**
Buzzer on touch	**false**
Buzzer duration (ms)	**200**
Image DB Enable	**true**
Plug-in	
Behavior	
Home Page	Page1.jmx
PageWidth	**800**
PageHeight	**480**
Display Mode	Landscape
Project Type	HMI
Panel Memory	128MB
PageRequest	

Project Template



Project Wizard

Project Template

● Unified

One page profile for HMI Device and Web clients.

○ Native and Web

One page profile for HMI Device and one for Web clients.

○ Custom

Multiple page profiles for Web clients.
Example: clients based on tablet and smartphone.

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
Unified	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.
Native and Web	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.
Custom	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

AGI Creator 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***

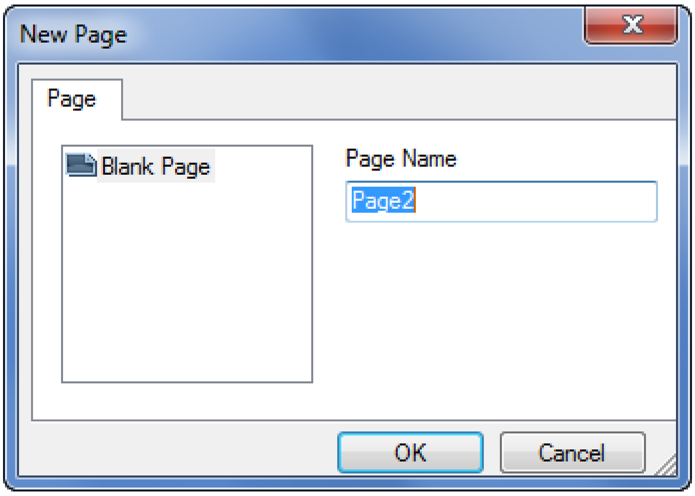
When a project is created, the first page is automatically added and shown in the **Page Editor**.

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 Name: Page2

Importing a page

When importing a page AGI Creator 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

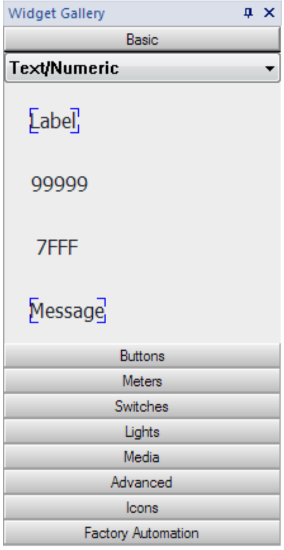
You can group similar pages for easier maintenance. Grouping pages does not affect how pages appears at runtime. To create a group of pages:

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.



Adding a widget to a page

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

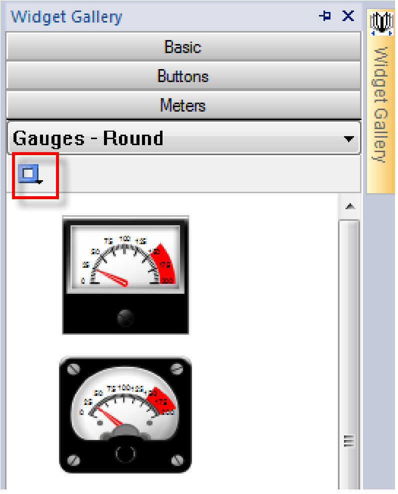
Changing the appearance of a widget

All widgets have properties (**Properties** pane) that can be changed, Some widgets are presented in various styles. You can click the buttons in each category to see available styles.

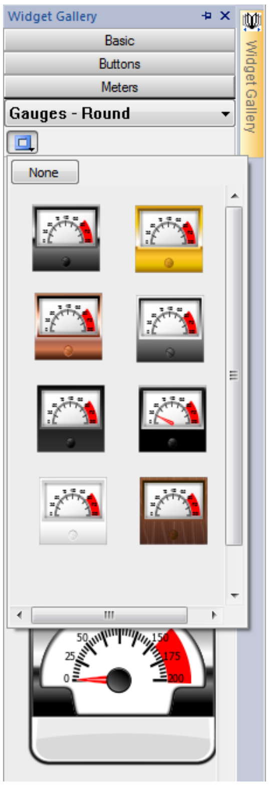
Example

To set the widget style for round gauges:

1. Click the style button to display the available styles for the widget.



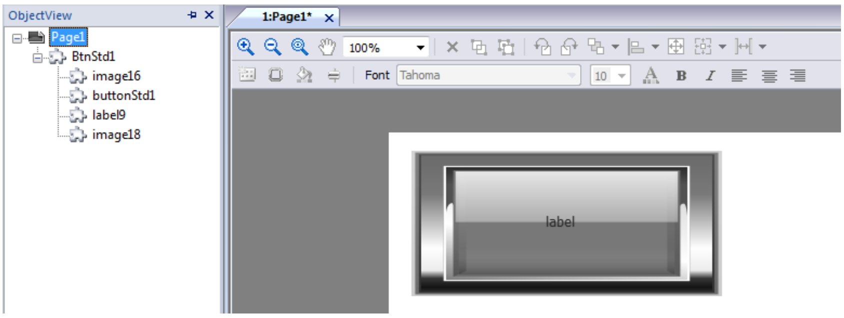
2. Select one of the available styles from the toolbar: depending on the selected widget, different options are available.



Complex widgets

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

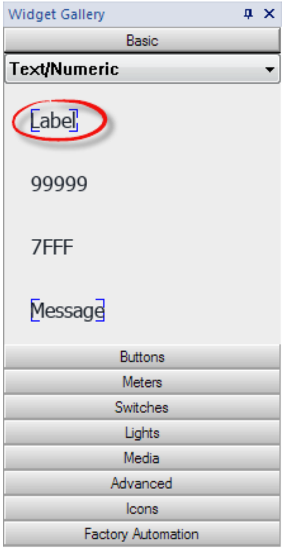
You can select a sub-widget, such as the label in a button, from the **ObjectView** and modify it without ungrouping the whole widget.



Label widget

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

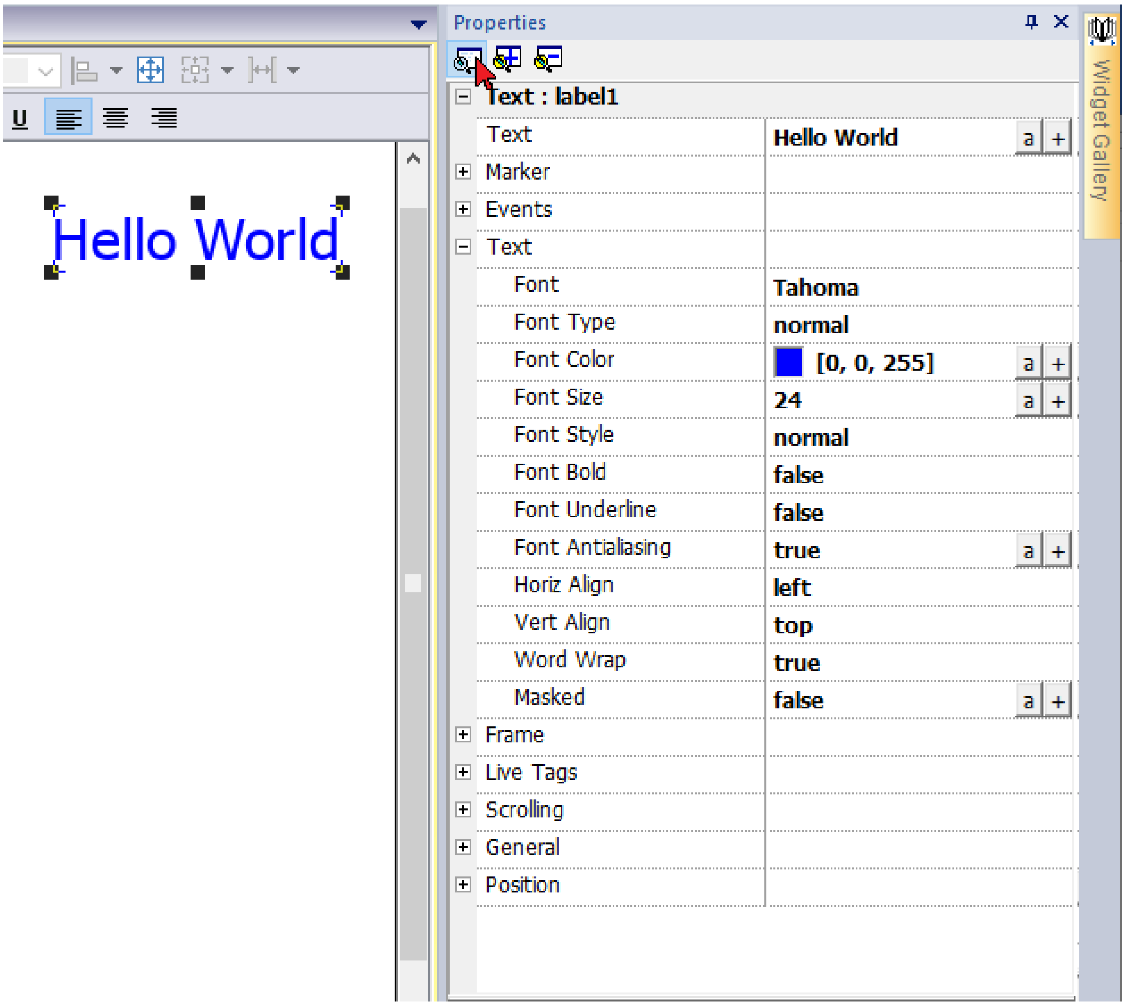
*Path: **View> Toolbars and Docking Windows> Widget Gallery***



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.



Text : label1

Property	Value
Text	Hello World
Marker	
Events	
Text	
Font	Tahoma
Font Type	normal
Font Color	[0, 0, 255]
Font Size	24
Font Style	normal
Font Bold	false
Font Underline	false
Font Antialiasing	true
Horiz Align	left
Vert Align	top
Word Wrap	true
Masked	false
Frame	
Live Tags	
Scrolling	
General	
Position	

Property	Description
Text	The string to display. String can be static or retrieved from a TAG. See "Attaching widget to tags " on page 39
Marker	Enable a Marker around the widget (It is visible only inside AGI Creator)
Events	Action that will be executed if widget contents change. See "Events" on page 55
Text (folder)	Text properties
Frame	Parameters to enable and configure a frame of the widget and/or a color for the background
Live Tags	Enable to use tags values inside the text message. See "Live Tags" on page 30 - Enable Live Tag: Enable live tags placeholder - 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.
Scrolling	Parameters to enable and configure the scrolling of the text message • None • Slow • Normal • Fast • Custom When the custom mode is selected, the below parameters can be defined: - Scroll type: For each timeout, the text is scrolled of a custom amount of characters or pixels. • Characters • Pixels - Scroll delay: The timeout after which label effectively start to scroll (mSec) - Scroll timer: The timeout which defines each scroll step (mSec) - Scroll dots or - Scroll characters: The number of pixels scrolled for each timer timeout or The number of character scrolled for each timer timeout - Scroll behavior: OnlyOnce Text scrolling stops after the first complete. LoopWithDelay Text scrolling restart after each complete cycle, waiting for the delay.

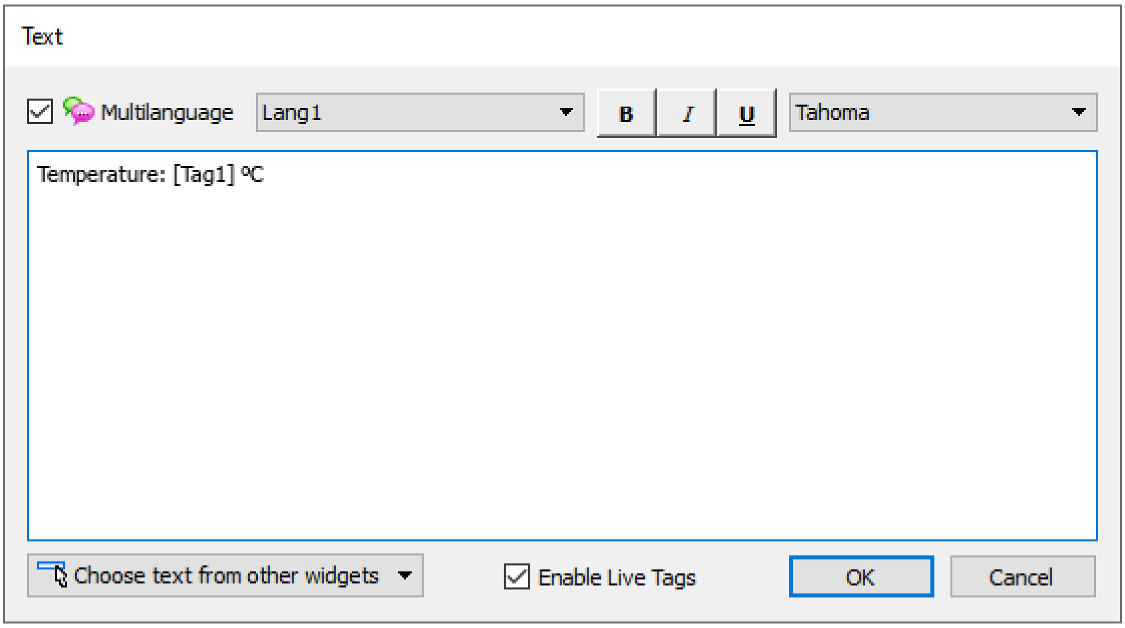
Property	Description	
		Loop Text scrolling never stop
General	General properties	
	- Id	Widget identifier. You can leave the default value or rename it to have a more appropriate name
	- Visible	When false the widget is not visible
	- Opacity	The opacity-level describes the transparency-level, where 1 is not transparent at all, 0.5 is 50% see-through, and 0 is completely transparent.
	- Blink	The text will blink
	- Lock	When True, the widget cannot be selected and moved from the AGI Creator page editor
	- Static Optimization	Normal AGI Creator will decide the best optimization mode to use Static AGI Creator optimize the widget assuming it will never be modified by the runtime Dynamic AGI Creator will not add additional optimizations
Position	The widget position on the display. See "Widget position on the display" on page 31	

Some properties have a couple of buttons:

a 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.



Temperature: [Tag1] °C

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 *"[Tag[1]]"*.

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: [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 34)

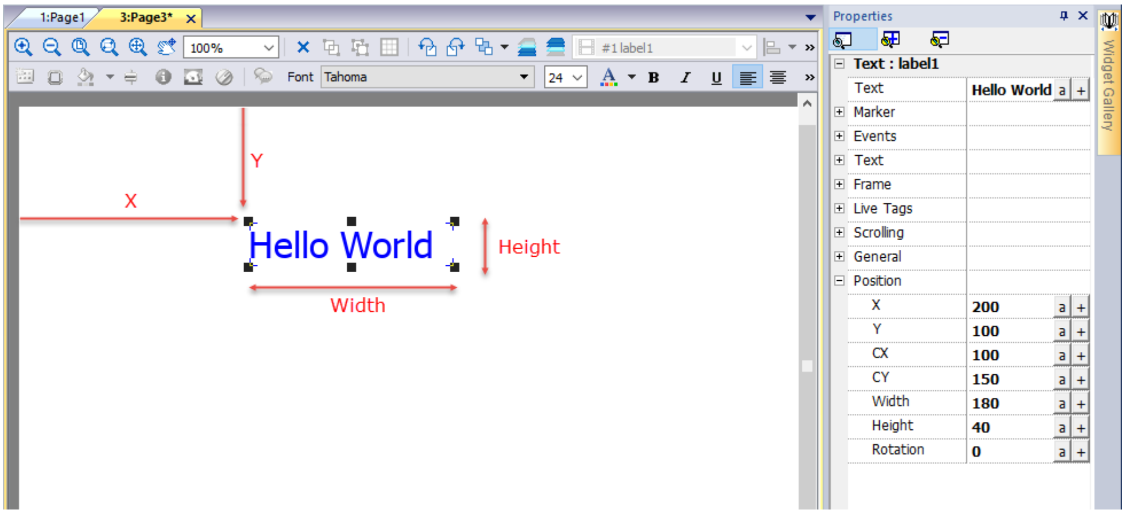
- [TagName|format("###")]

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

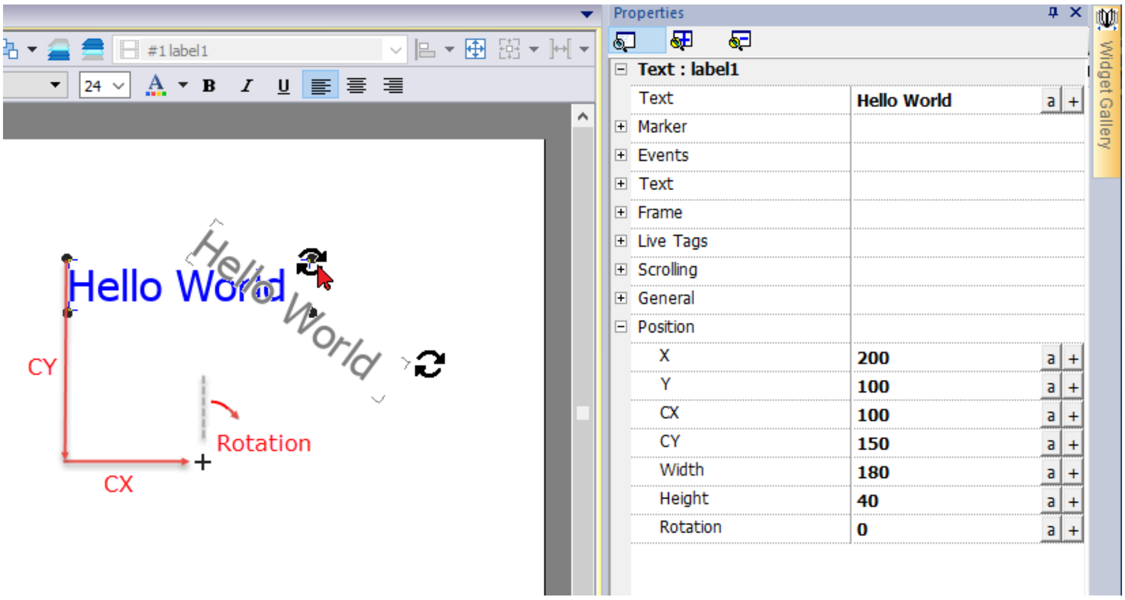


Text : label1

Text	Hello World
Marker	
Events	
Text	
Frame	
Live Tags	
Scrolling	
General	
Position	
X	200
Y	100
CX	100
CY	150
Width	180
Height	40
Rotation	0

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.



Text : label1

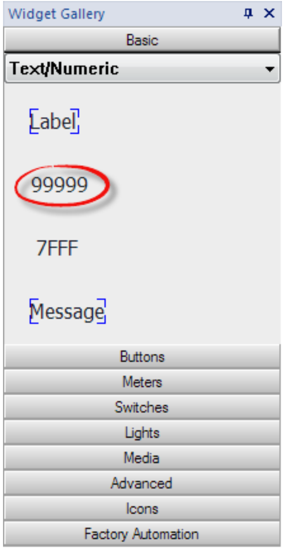
Text	Hello World
Marker	
Events	
Text	
Frame	
Live Tags	
Scrolling	
General	
Position	
X	200
Y	100
CX	100
CY	150
Width	180
Height	40
Rotation	0

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

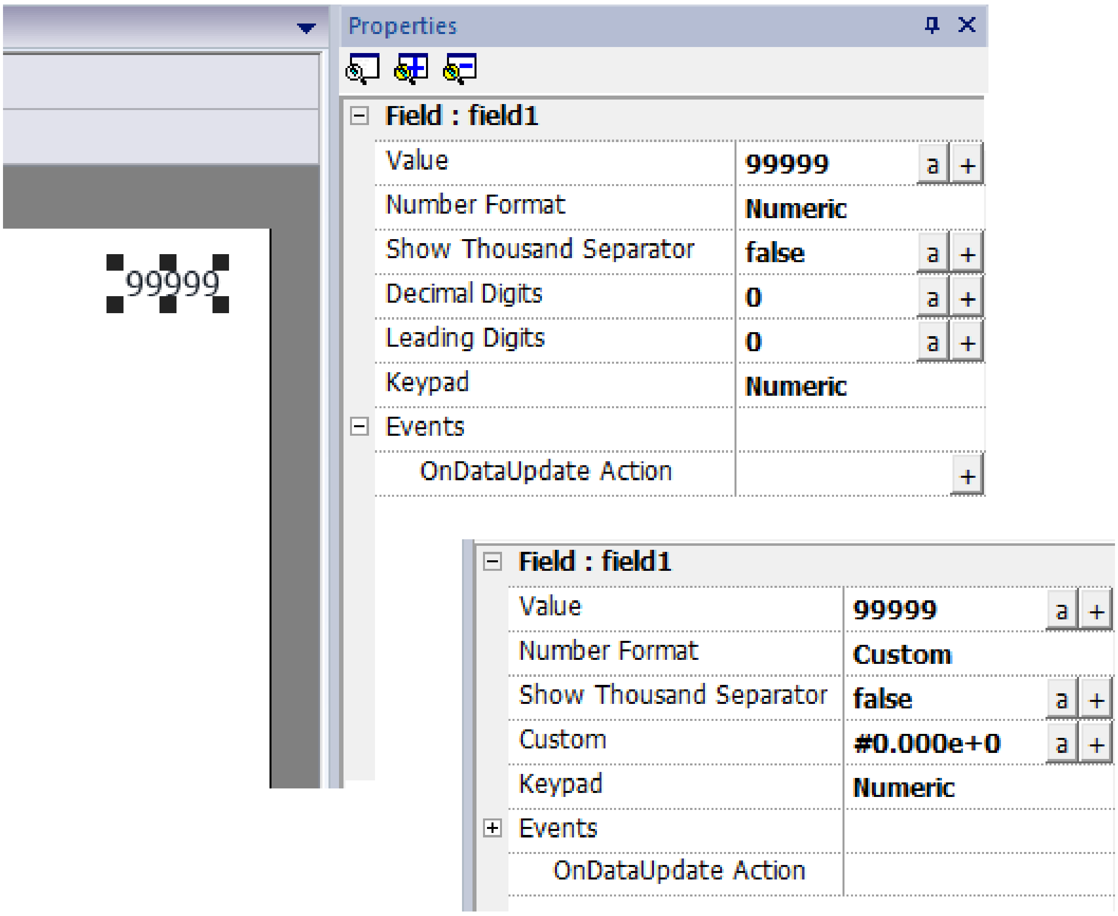
Data field widget

The most common widget is the data field widget that give the possibility to display value of tags. (See "Tag editor" on page 105).

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



Field properties



Field : field1

Value	99999
Number Format	Numeric
Show Thousand Separator	false
Decimal Digits	0
Leading Digits	0
Keypad	Numeric
Events	
OnDataUpdate Action	

Field : field1

Value	99999
Number Format	Custom
Show Thousand Separator	false
Custom	#0.000e+0
Keypad	Numeric
Events	
OnDataUpdate Action	

Property	Description
Value	Tag that contain the information to display
Number Format	Display format • None No restrictions (system decide the format to use) • Numeric Numerical format. Decimal digits and Leading digits can be used to better define the number format • hex Hexadecimal format. Leading digits can be used to better define the number format • HEX The same of "hex" format but using uppercase • scientific Scientific format. Decimal digits can be used to better define the number format • SCIENTIFIC The same of "scientific" format but using uppercase

Property	Description
	• Custom Use the additional "Custom" parameter to better define the format to use (see the below table)
Show Thousand Separator	To show/hidden the thousands separator
Decimal Digits	Number of decimal digits to show (not available on all format types)
Leading Digits	Number of leading digits to show (not available on all format types)
Custom	String that define the number format to use (available only when selected Number Format is CUSTOM)
Keypad	Keypad type to pop up to edit the tab value • None • Alphabetic, Numeric, Etc. Pop up a predefined keypad or a user keypad (see "Keypads" on page 365) • Wheel Keypad will not be displayed . Wheel can be used to increment/decrement the numeric value • Macro Keypad will not be displayed . Keyboard macro can be used to enter keys (see "Keyboard actions" on page 180)
Events	
OnDataUpdate Action	Commands list to execute any time the tag value changes (See "Actions" on page 177 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 91

Some properties have a couple of buttons:

a	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.

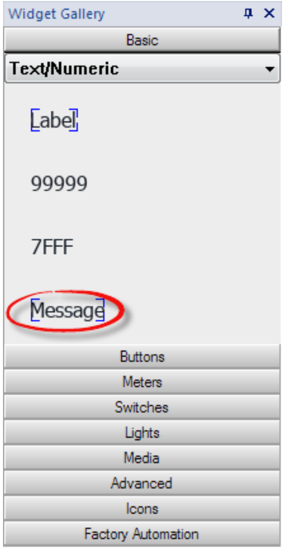
Examples

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.

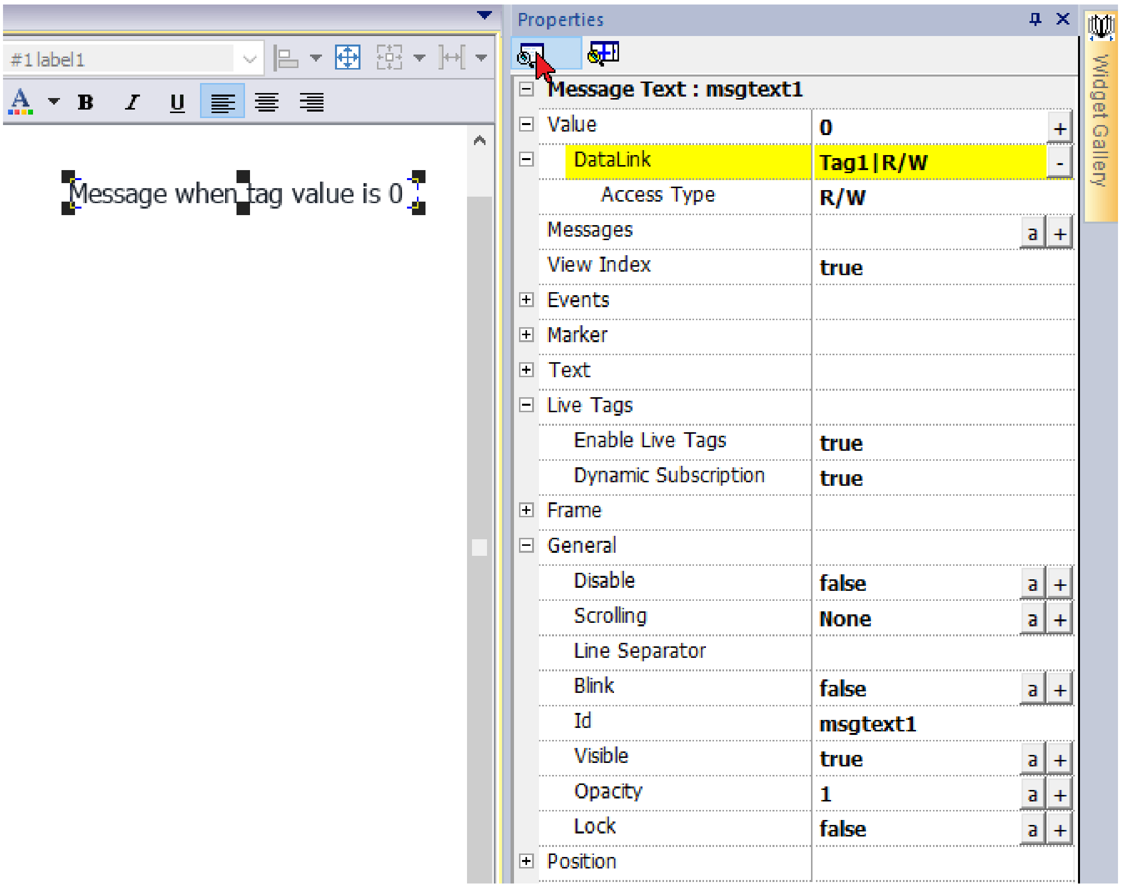
Path: ***View> Toolbars and Docking Windows**> **Widget Gallery***



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.



Message when tag value is 0

Message Text : msgtext1

Value 0
DataLink Tag1|R/W
Access Type R/W
Messages
View Index true
Events
Marker
Text
Live Tags
Enable Live Tags true
Dynamic Subscription true
Frame
General
Disable false
Scrolling None
Line Separator
Blink false
Id msgtext1
Visible true
Opacity 1
Lock false
Position

Property	Description
Value	The tag name to used to dynamically select the message to display. See "Attaching widget to tags " on page 39 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.
Messages	The list of messages to display. Click **+** to open a dialog where you enter messages and the associated index
View Index	If true, when the widget is editable (Access Type = R/W) the selection dialog will also show the index number associated with each message.
Events	Action that will be executed if widget contents change. See "Events" on page 55
Marker	Enable a Marker around the widget (It is visible only inside AGI Creator)
Text	Text properties (font, color, size, etc.)

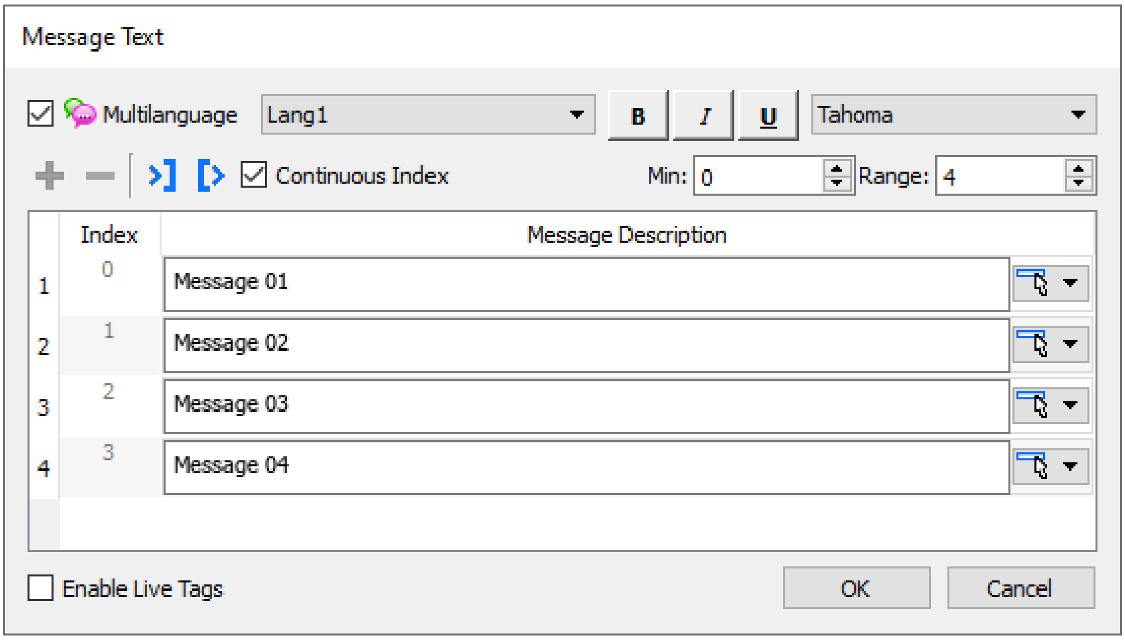
<table>
<tr><th>Property</th><th>Description</th></tr>
<tr><td><b>Live Tags</b></td><td>Enable to use tags values inside the text message. See "Message widget" on page 35
<table>
<tr><td>- Enable Live Tag</td><td>Enable live tags placeholder</td></tr>
<tr><td>- Dynamic Subscription</td><td>When true, 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.</td></tr>
</table></td></tr>
<tr><td><b>Frame</b></td><td>Parameters to enable and configure a frame of the widget and/or a color for the background</td></tr>
<tr><td><b>General</b></td><td>General properties
<table>
<tr><td>- Disable</td><td>Disable user inputs on the widget
Attaching a tag at this property is possible to enable/disable the possibility to modify the value at runtime</td></tr>
<tr><td>- Scrolling</td><td>Parameters to enable and configure the scrolling of the text message
When enabled, all messages are linked together and displayed in scrolling mode.</td></tr>
<tr><td>- Line Separator</td><td>Characters to insert between messages when shown in scrolling mode</td></tr>
<tr><td>- Blink</td><td>The text will blink</td></tr>
<tr><td>- Id</td><td>Widget identifier. You can leave the default value or rename it to have a more appropriate name</td></tr>
<tr><td>- Visible</td><td>When false the widget is not visible</td></tr>
<tr><td>- Opacity</td><td>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.</td></tr>
<tr><td>- Lock</td><td>When True, the widget cannot be selected and moved from the AGI Creator page editor</td></tr>
</table></td></tr>
<tr><td><b>Position</b></td><td>The widget position on the display. See "Widget position on the display" on page 31</td></tr>
</table>

Some properties have a couple of buttons:

a 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.



Message Text

☑ Multilanguage Lang1 B I U Tahoma

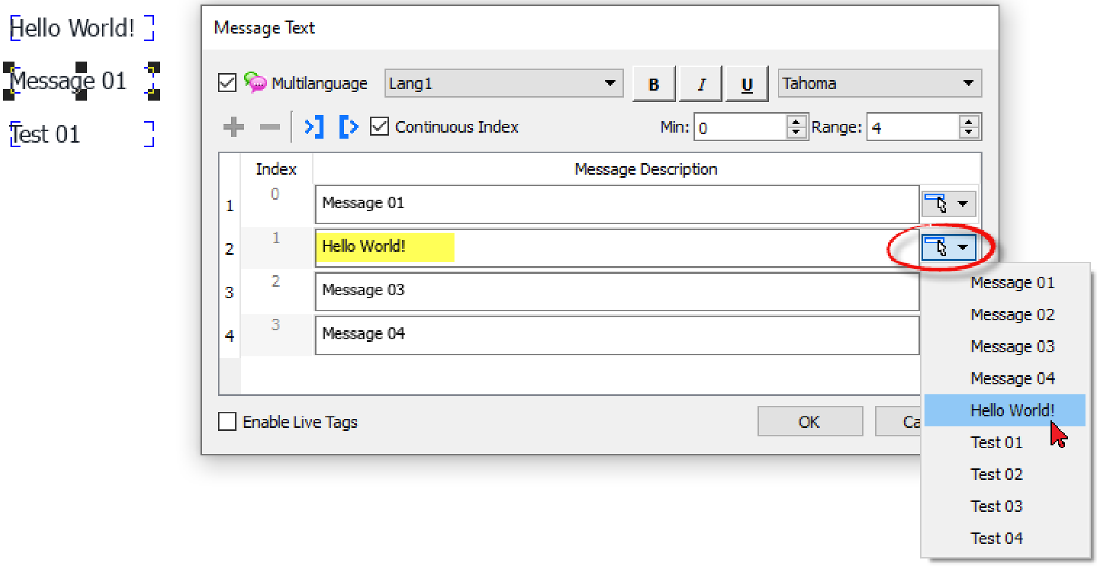
☑ Continuous Index Min: 0 Range: 4

	Index	Message Description
1	0	Message 01
2	1	Message 02
3	2	Message 03
4	3	Message 04

☐ Enable Live Tags OK Cancel

Pick Text Button

The "Pick Text" button gives you the possibility to copy text already used from other widgets



Hello World!

Message 01

Test 01

Message Text

☑ Multilanguage Lang1 B I U Tahoma

☑ Continuous Index Min: 0 Range: 4

	Index	Message Description
1	0	Message 01
2	1	Hello World!
3	2	Message 03
4	3	Message 04

☐ Enable Live Tags OK Ca

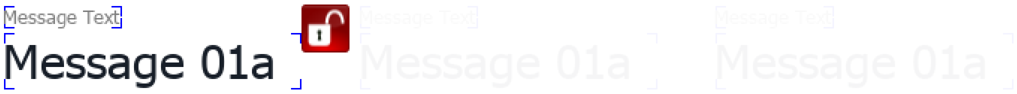
Import/Export

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 452



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

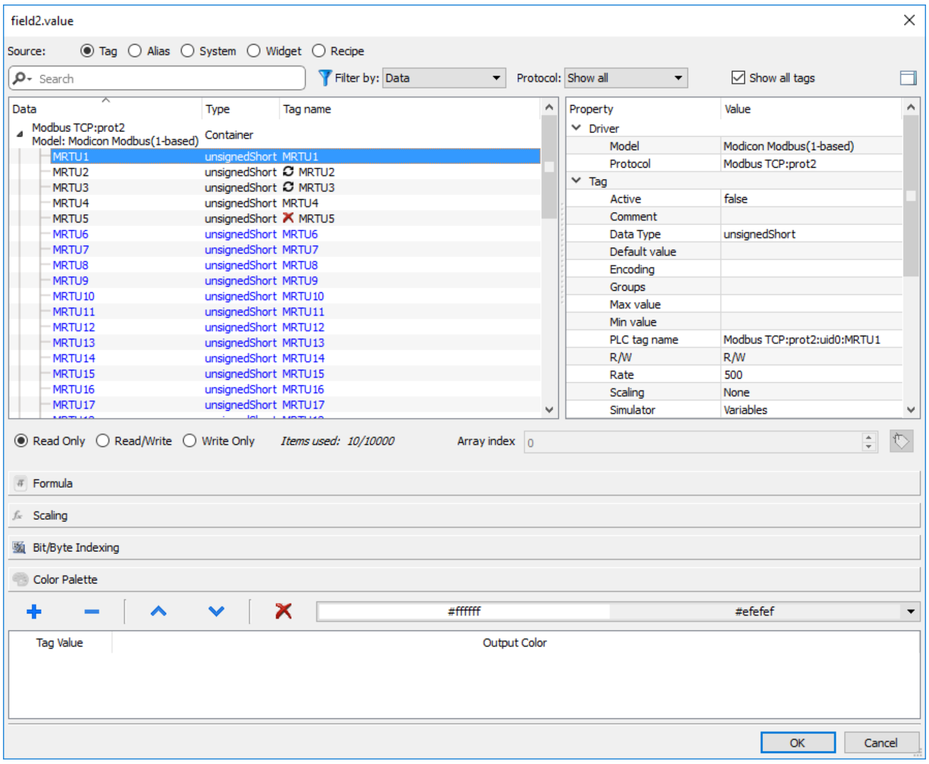
Data sources

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 "System Variables (Attach To)" on page 133)
Widget	Connect to a widget property (for example, value of a slider widget)
Recipe	Data from the Recipe Manager (see "Recipes" on page 259)

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.



field2.value

Source: ● Tag ○ Alias ○ System ○ Widget ○ Recipe

Search | Filter by: Data | Protocol: Show all | ☑ Show all tags

Data	Type	Tag name
Modbus TCP:prot2 Model: Modicon Modbus(1-based)	Container	
MRTU1	unsignedShort	MRTU1
MRTU2	unsignedShort	MRTU2
MRTU3	unsignedShort	MRTU3
MRTU4	unsignedShort	MRTU4
MRTU5	unsignedShort	✗ MRTU5
MRTU6	unsignedShort	MRTU6
MRTU7	unsignedShort	MRTU7
MRTU8	unsignedShort	MRTU8
MRTU9	unsignedShort	MRTU9
MRTU10	unsignedShort	MRTU10
MRTU11	unsignedShort	MRTU11
MRTU12	unsignedShort	MRTU12
MRTU13	unsignedShort	MRTU13
MRTU14	unsignedShort	MRTU14
MRTU15	unsignedShort	MRTU15
MRTU16	unsignedShort	MRTU16
MRTU17	unsignedShort	MRTU17

Property	Value
Driver	
Model	Modicon Modbus(1-based)
Protocol	Modbus TCP:prot2
Tag	
Active	false
Comment	
Data Type	unsignedShort
Default value	
Encoding	
Groups	
Max value	
Min value	
PLC tag name	Modbus TCP:prot2:uid0:MRTU1
R/W	R/W
Rate	500
Scaling	None
Simulator	Variables

● Read Only ○ Read/Write ○ Write Only *Items used: 10/10000* Array index 0

Formula

Scaling

Bit/Byte Indexing

Color Palette

#ffffff #efefef

Tag Value	Output Color

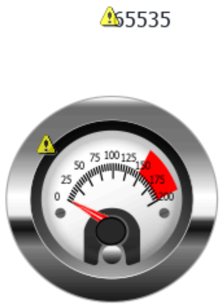
3. Set the access type (for example **Read Only**). The **Array Index** field appears when the selected tag is an array to identify the element of the array to use. The indirect index mode, through an additional tag, is supported.
4. Click **OK** to confirm.

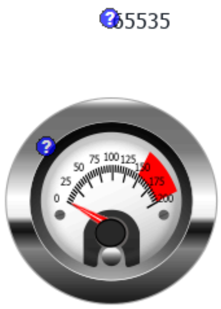
The icons adjacent to the tag name highlight when a definition does not match the tag definition in the dictionary, or when missing. If the **Show all tags** is selected, all the dictionary tags are shown also if not imported within the application. A double-click will import the tags from the dictionary.

See ""Attach to" parameters" on page 45 for details.

Communication Error

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





- ⚠: communication error
- ❓: data not yet available (slow communication protocol)

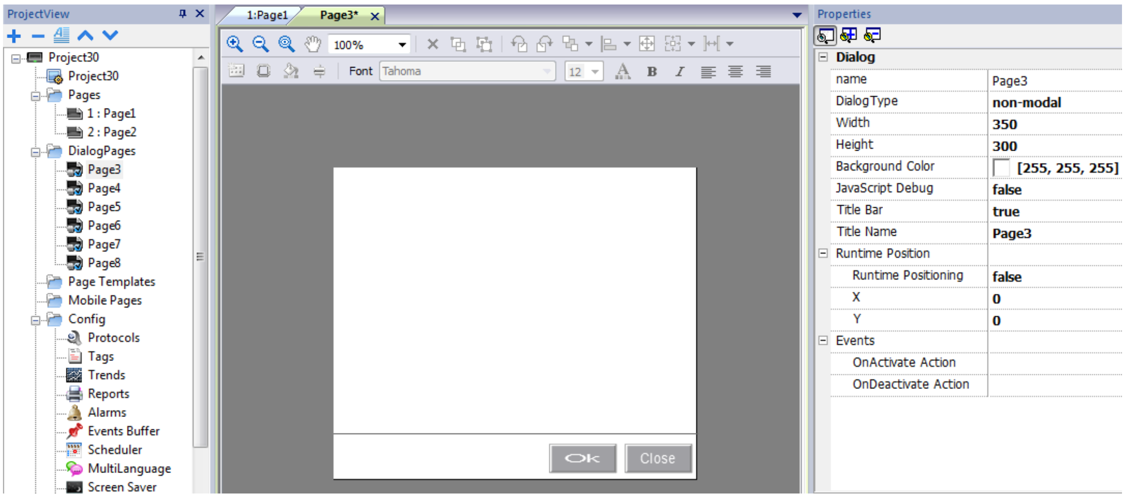
Dialog pages

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



Dialog	
name	Page3
DialogType	non-modal
Width	350
Height	300
Background Color	[255, 255, 255]
JavaScript Debug	false
Title Bar	true
Title Name	Page3
Runtime Position	
Runtime Positioning	false
X	0
Y	0
Events	
OnActivate Action	
OnDeactivate Action	

Maximum number of dialogs

Maximum number of open dialogs is defined in "Functional specifications and compatibility" on page 541.

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

5 Programming concepts

Programming for AGI Creator is based on a few basic concepts and behaviors.

Data types

When creating a tag you have to specify its properties. Data type are specific to AGI Creator, 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 [ ])

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.

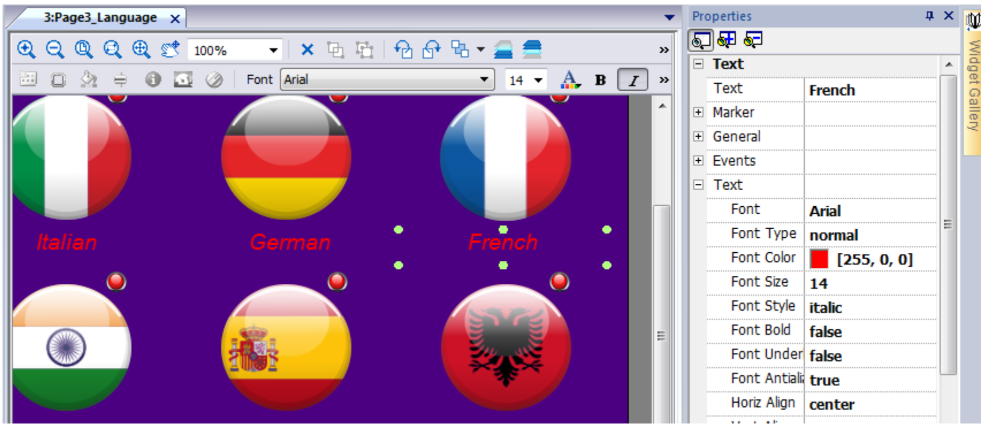
Example:

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 AGI Creator 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.



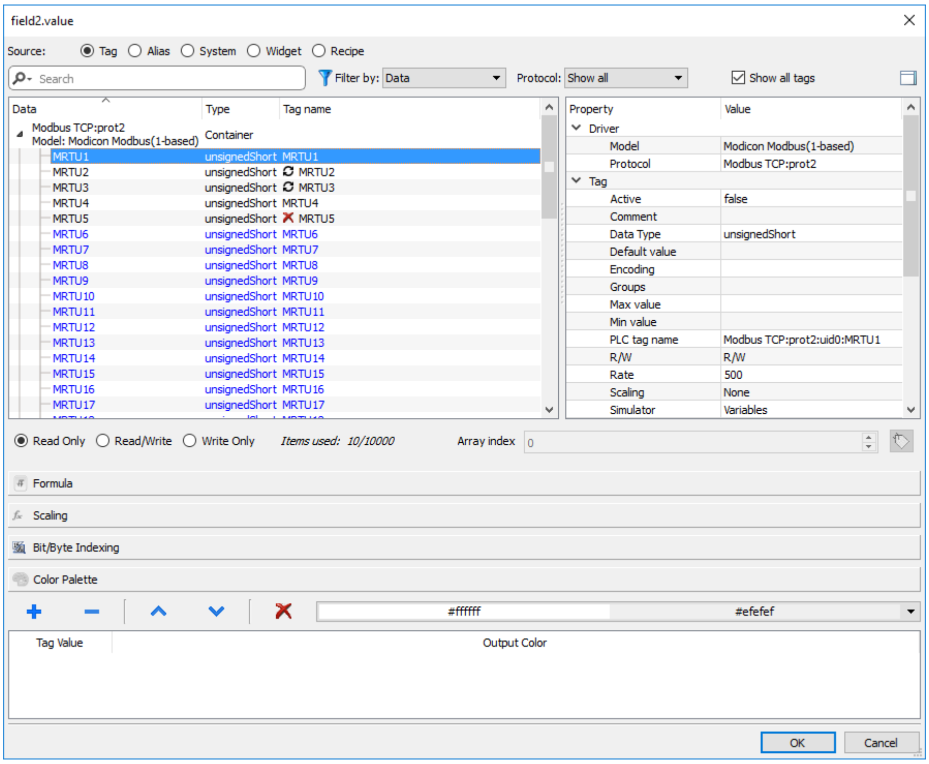
Text	
Text	French
Marker	
General	
Events	
Text	
Font	Arial
Font Type	normal
Font Color	🟥 [255, 0, 0]
Font Size	14
Font Style	italic
Font Bold	false
Font Under	false
Font Antiali	true
Horiz Align	center

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.



field2.value

Source: ● Tag ○ Alias ○ System ○ Widget ○ Recipe

Search | Filter by: Data | Protocol: Show all | ☑ Show all tags

Data	Type	Tag name
Modbus TCP:prot2 Model: Modicon Modbus(1-based)	Container	
MRTU1	unsignedShort	MRTU1
MRTU2	unsignedShort	MRTU2
MRTU3	unsignedShort	MRTU3
MRTU4	unsignedShort	MRTU4
MRTU5	unsignedShort	✗ MRTU5
MRTU6	unsignedShort	MRTU6
MRTU7	unsignedShort	MRTU7
MRTU8	unsignedShort	MRTU8
MRTU9	unsignedShort	MRTU9
MRTU10	unsignedShort	MRTU10
MRTU11	unsignedShort	MRTU11
MRTU12	unsignedShort	MRTU12
MRTU13	unsignedShort	MRTU13
MRTU14	unsignedShort	MRTU14
MRTU15	unsignedShort	MRTU15
MRTU16	unsignedShort	MRTU16
MRTU17	unsignedShort	MRTU17

Property	Value
Driver	
Model	Modicon Modbus(1-based)
Protocol	Modbus TCP:prot2
Tag	
Active	false
Comment	
Data Type	unsignedShort
Default value	
Encoding	
Groups	
Max value	
Min value	
PLC tag name	Modbus TCP:prot2:uid0:MRTU1
R/W	R/W
Rate	500
Scaling	None
Simulator	Variables

● Read Only ○ Read/Write ○ Write Only *Items used: 10/10000* Array index 0

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.

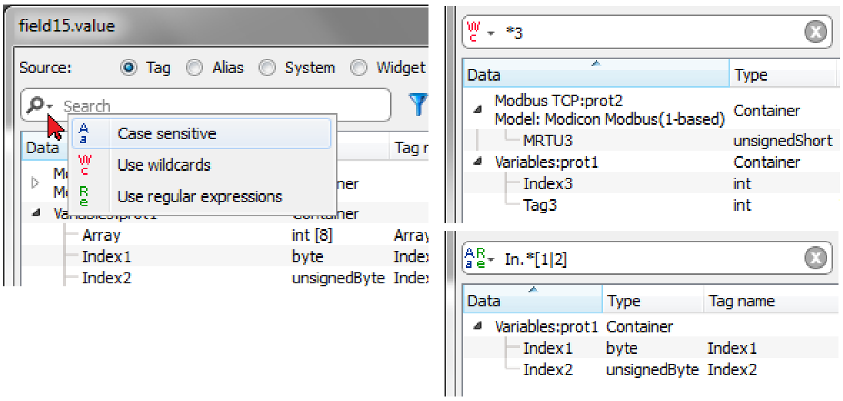
Data sources

Elements to which an object property can be attached:

Data source	Description
Tag	Tag defined in the Tag Editor
Alias	Indexed tag address
System	Predefined system tags (see "System Variables (Attach To)" on page 133)
Widget	Connect to a widget property (for example, value of a slider widget)
Recipe	Data from the Recipe Manager (see "Recipes" on page 259)

Advanced search

Various syntax options can be applied to search box:



Data	Type	Tag r
Array	int [8]	Array
Index1	byte	Inde:
Index2	unsignedByte	Inde:

*3

Data	Type
Modbus TCP:prot2 Model: Modicon Modbus(1-based)	Container
MRTU3	unsignedShort
Variables:prot1	Container
Index3	int
Tag3	int

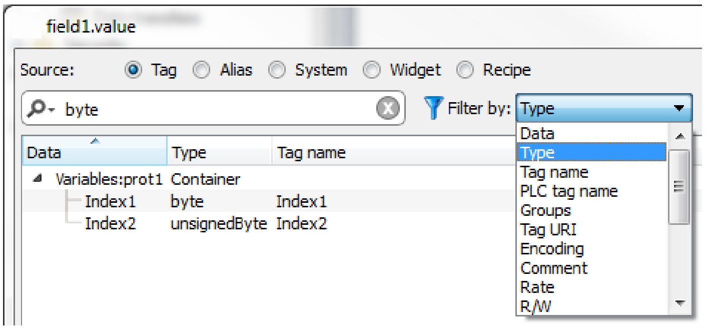
In.*[1|2]

Data	Type	Tag name
Variables:prot1	Container	
Index1	byte	Index1
Index2	unsignedByte	Index2

Main options	Function
Wildcards	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.
Regular Expression	Describes character pattern. See https://en.wikipedia.org/wiki/Regular_expression for additional details regarding regular expressions.

Filtering tags

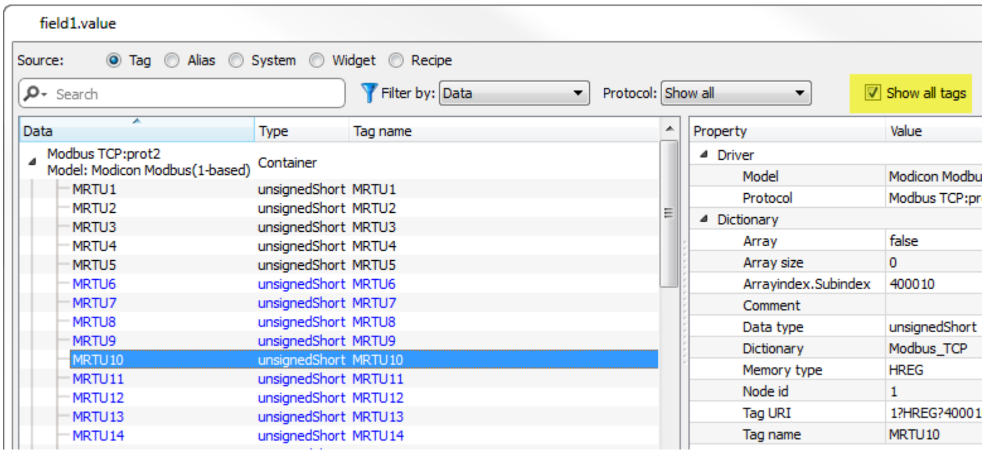
Choose various tag filter criteria:



Data	Type	Tag name
Variables:prot1	Container	
Index1	byte	Index1
Index2	unsignedByte	Index2

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.



field1.value

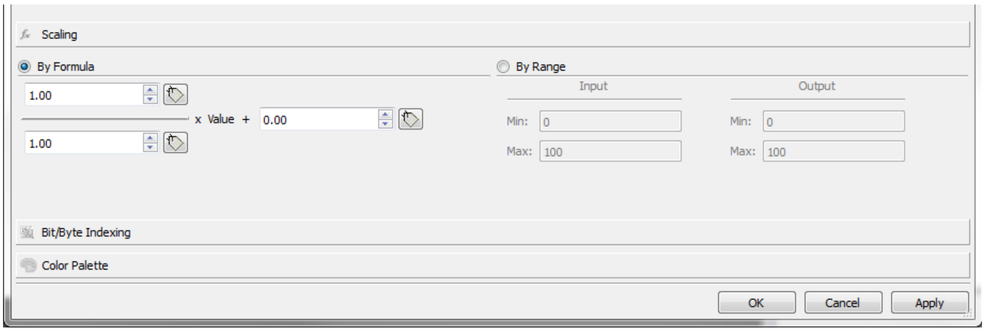
Source: ● Tag ○ Alias ○ System ○ Widget ○ Recipe

Filter by: Data   Protocol: Show all   ☑ Show all tags

Data	Type	Tag name
Modbus TCP:prot2 Model: Modicon Modbus(1-based)	Container	
MRTU1	unsignedShort	MRTU1
MRTU2	unsignedShort	MRTU2
MRTU3	unsignedShort	MRTU3
MRTU4	unsignedShort	MRTU4
MRTU5	unsignedShort	MRTU5
MRTU6	unsignedShort	MRTU6
MRTU7	unsignedShort	MRTU7
MRTU8	unsignedShort	MRTU8
MRTU9	unsignedShort	MRTU9
MRTU10	unsignedShort	MRTU10
MRTU11	unsignedShort	MRTU11
MRTU12	unsignedShort	MRTU12
MRTU13	unsignedShort	MRTU13
MRTU14	unsignedShort	MRTU14

Property	Value
Driver	
Model	Modicon Modbu
Protocol	Modbus TCP:pr
Dictionary	
Array	false
Array size	0
Arrayindex.Subindex	400010
Comment	
Data type	unsignedShort
Dictionary	Modbus_TCP
Memory type	HREG
Node id	1
Tag URI	1?HREG?40001
Tag name	MRTU10

Converting tag value



Scaling

● By Formula   ○ By Range

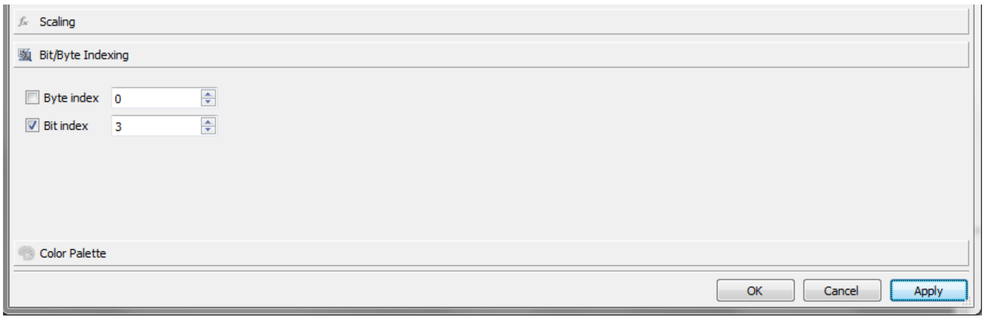
1.00 / 1.00 x Value + 0.00

Input Min: 0 Max: 100   Output Min: 0 Max: 100

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



Bit/Byte Indexing

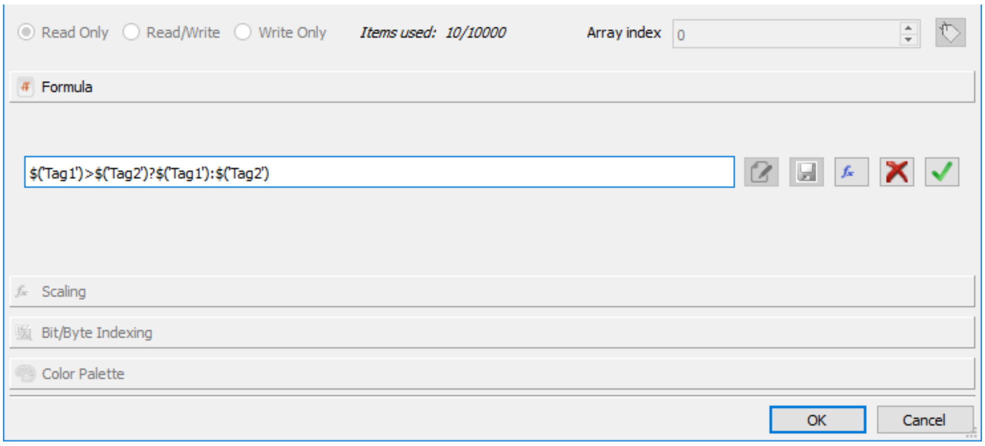
□ Byte index 0

☑ Bit index 3

Use a formula to calculate the value to use

Allow to use a formula to calculate the value to use. See "Formula" on page 50 chapter for additional details.

Note that using a formula the datalink will be ReadOnly



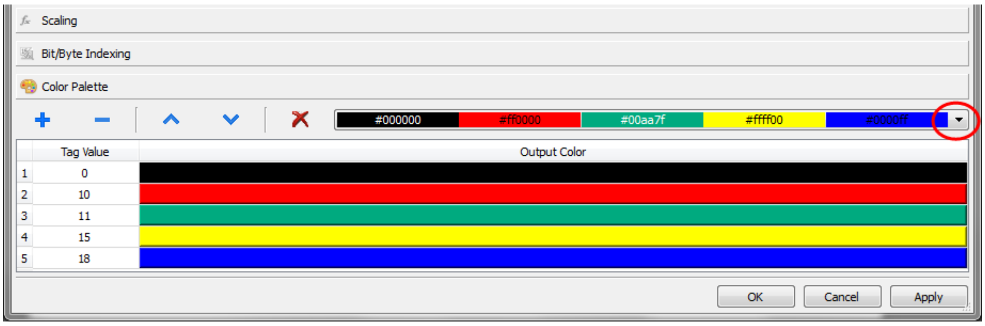
○ Read Only ○ Read/Write ○ Write Only *Items used: 10/10000* Array index 0

Formula

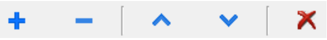
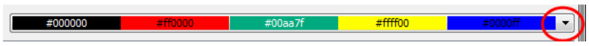
```
$('Tag1')>$('Tag2')?$('Tag1'):$('Tag2')
```

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.



	Tag Value	Output Color
1	0	⬛
2	10	🟥
3	11	🟩
4	15	🟨
5	18	🟦

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

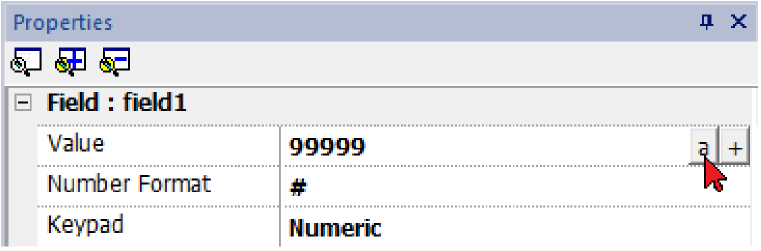
It is responsibility of the application's developer define all items correctly to cover all possible application’s values, we could have unexpected color when the value is not defined inside the defined colors palette.

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.

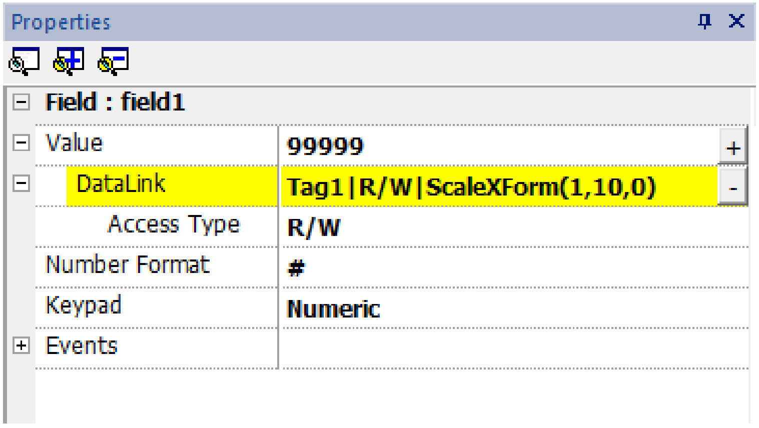
Click **a** button in the **Properties** pane and enter the text that describe the datalink



Properties

Field : field1

Value	99999
Number Format	#
Keypad	Numeric



Properties

Field : field1

Value	99999
DataLink	Tag1|R/W|ScaleXForm(1,10,0)
Access Type	R/W
Number Format	#
Keypad	Numeric
Events	

The data link format is:

```
Tagname [index] | [Atribute] | [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'))

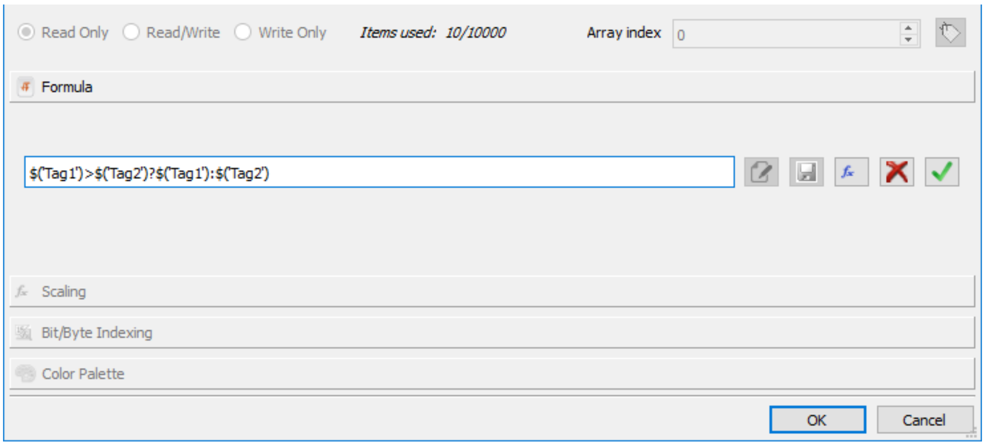
Formula

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.



○ Read Only ○ Read/Write ○ Write Only *Items used: 10/10000* Array index 0

Formula

$('Tag1')>$('Tag2')?$('Tag1'):$('Tag2')

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.
fx	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 $)

Basic Operations	
$FuncName(param1, param2, …)	General function call. (Both default and user ones)
$(‘TagName’)	Tag, or widget property, or recipe, etc. ⚠ **Note that tag name must be string literal**
$(‘TagName’)[index]	Element of a array tag. ⚠ **Note that tag name must be string literal**
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

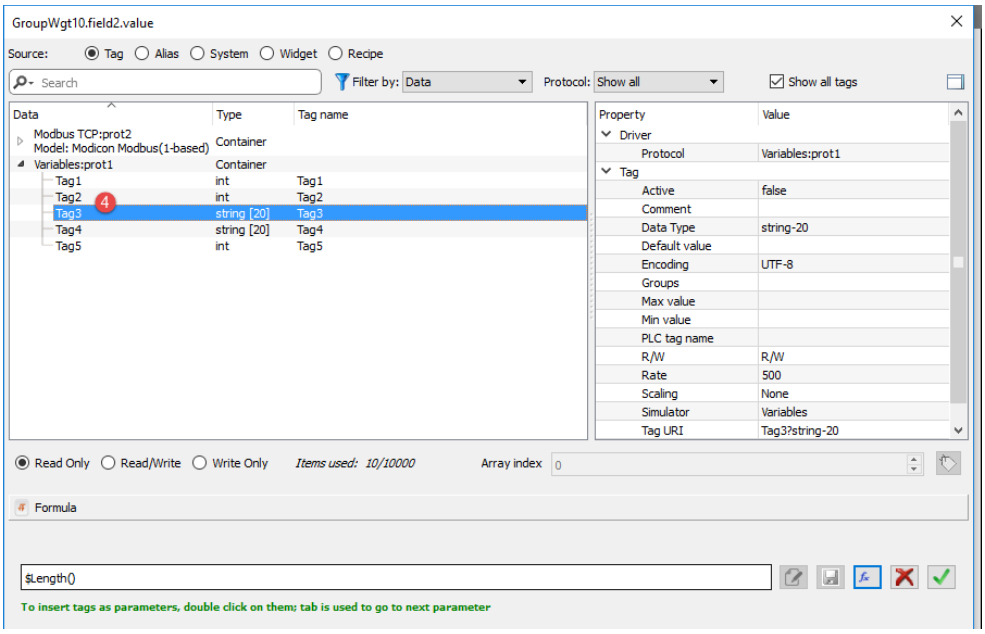
Logical Operators	
&&	AND
||	OR
!	NOT

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



GroupWgt10.field2.value

Source: ● Tag ○ Alias ○ System ○ Widget ○ Recipe

Search Filter by: Data Protocol: Show all ☑ Show all tags

Data	Type	Tag name
Modbus TCP:prot2 Model: Modicon Modbus(1-based)	Container	
Variables:prot1	Container	
Tag1	int	Tag1
Tag2	int	Tag2
Tag3	string [20]	Tag3
Tag4	string [20]	Tag4
Tag5	int	Tag5

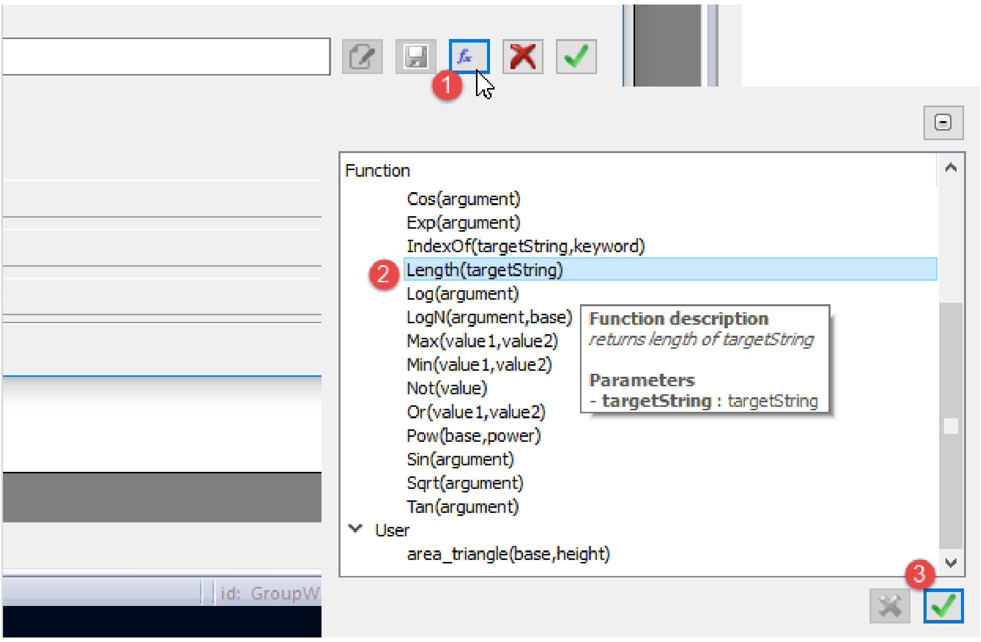
Property	Value
Driver	
Protocol	Variables:prot1
Tag	
Active	false
Comment	
Data Type	string-20
Default value	
Encoding	UTF-8
Groups	
Max value	
Min value	
PLC tag name	
R/W	R/W
Rate	500
Scaling	None
Simulator	Variables
Tag URI	Tag3?string-20

● Read Only ○ Read/Write ○ Write Only *Items used: 10/10000* Array index 0

Formula

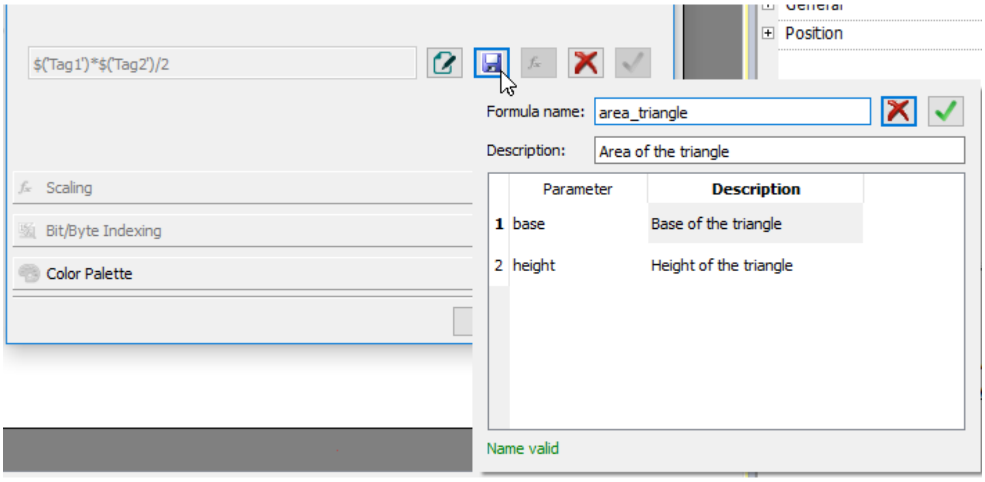
$Length()

To insert tags as parameters, double click on them; tab is used to go to next parameter



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.

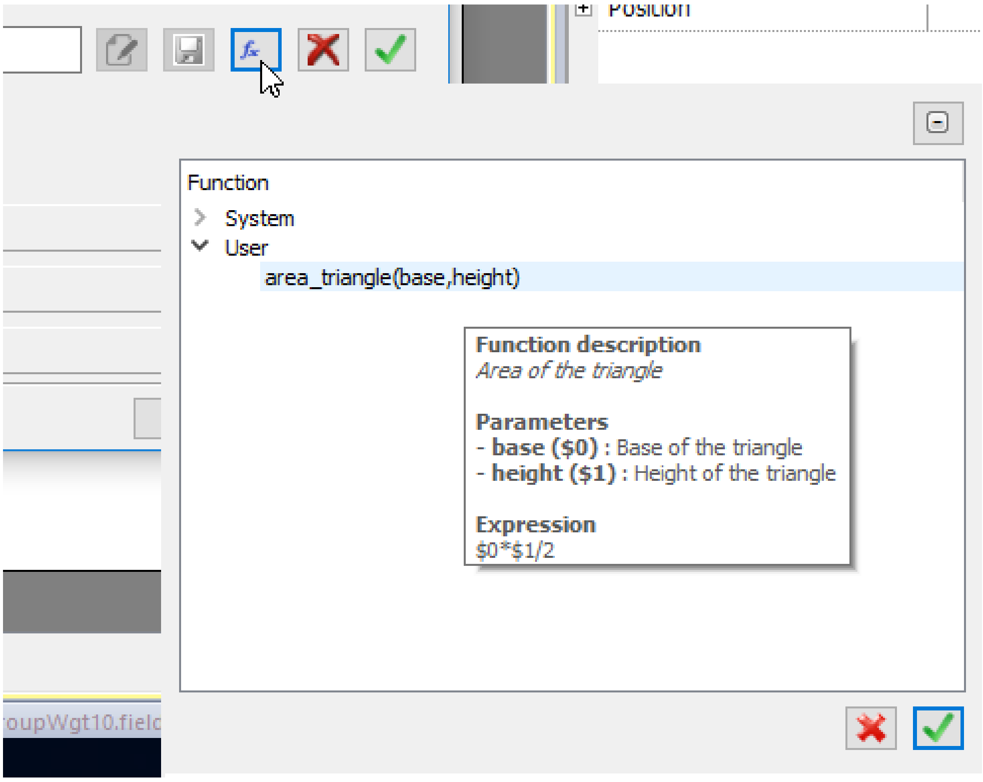


Formula name: area_triangle

Description: Area of the triangle

	Parameter	Description
1	base	Base of the triangle
2	height	Height of the triangle

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



Function description
Area of the triangle

Parameters
- **base ($0)** : Base of the triangle
- **height ($1)** : Height of the triangle

Expression
$0*$1/2

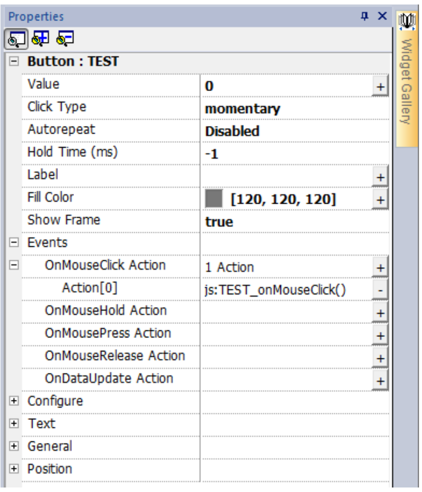
Events

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

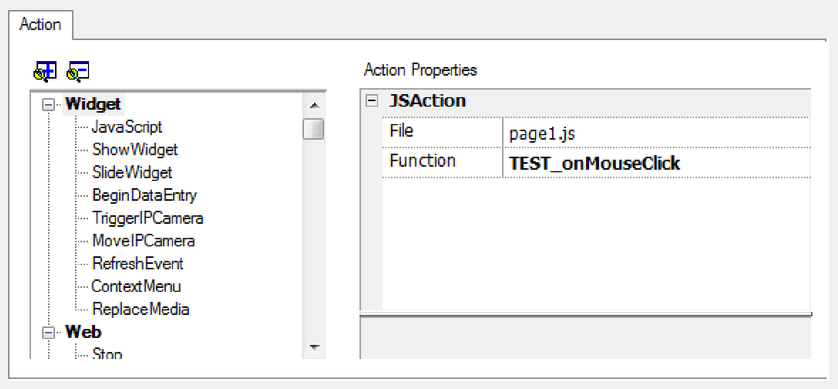
You can attach one or more actions to an event, so that they will be executed whenever the event occurs.

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



Button : TEST

Value	0
Click Type	momentary
Autorepeat	Disabled
Hold Time (ms)	-1
Label	
Fill Color	▇ [120, 120, 120]
Show Frame	true
Events	
OnMouseClick Action	1 Action
Action[0]	js:TEST_onMouseClick()
OnMouseHold Action	
OnMousePress Action	
OnMouseRelease Action	
OnDataUpdate Action	
Configure	
Text	
General	
Position	



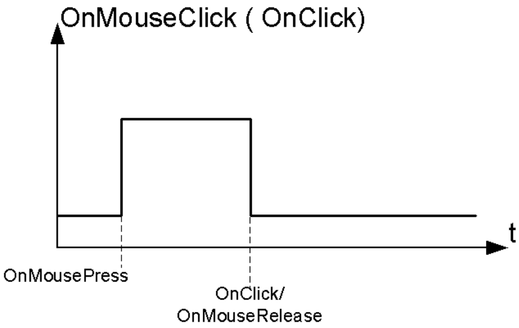
Action Properties

JSAction

File	page1.js
Function	TEST_onMouseClick

OnClick / OnMouseClick

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

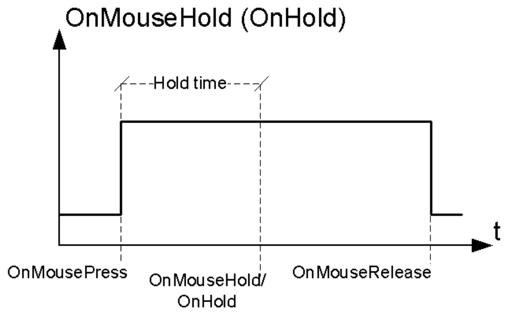


OnHold/OnMouseHold

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.

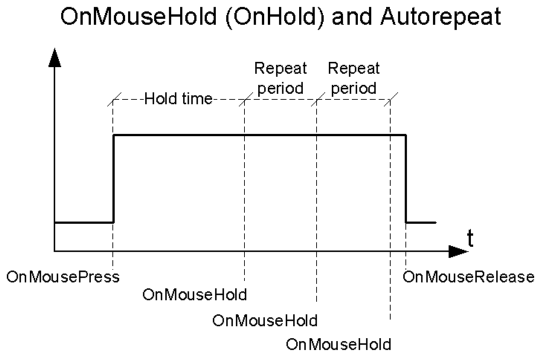
The default **Hold Time** is configured in Project properties but can be redefined for each button/key. See "Project properties" on page 79.

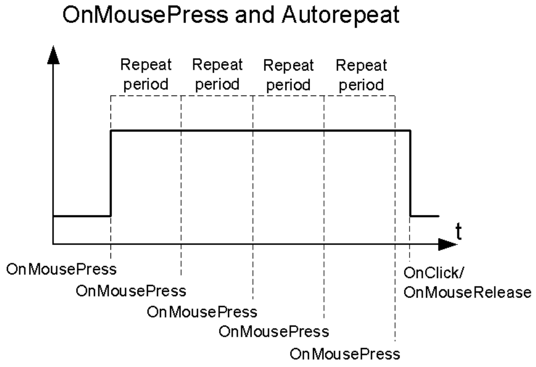
Note: If **Hold Time** is set to -1 for the widget, the project **Hold Time** value will be used.



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





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***

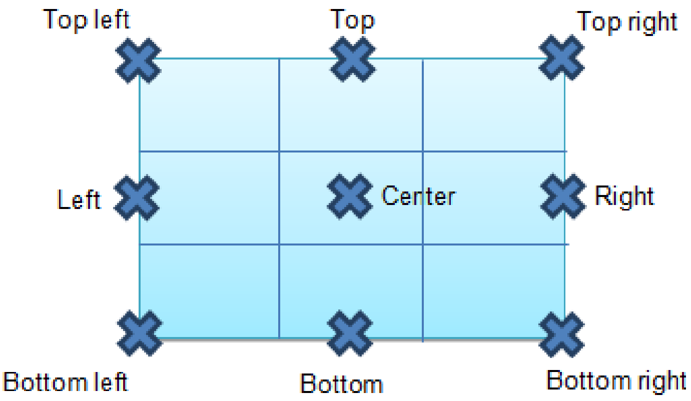
Parameter	Description
Spacing X	Space in pixel between two lines/dots on the X axis
Spacing Y	Space in pixel between two lines/dots on the Y axis
Type	Grid type (dot or line)
Color	Grid color

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:

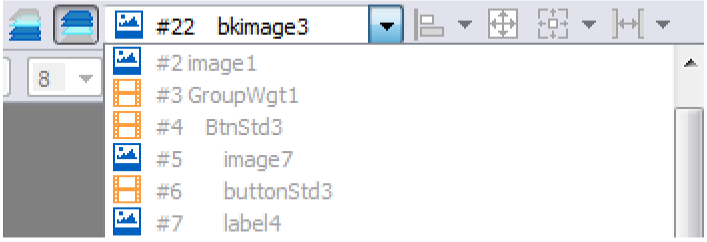


An algorithm finds a matching hot point among the near widgets hot points matching either the x or the y coordinates of the source snap point. For line widgets, the source snap points are the terminal points of the line.

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

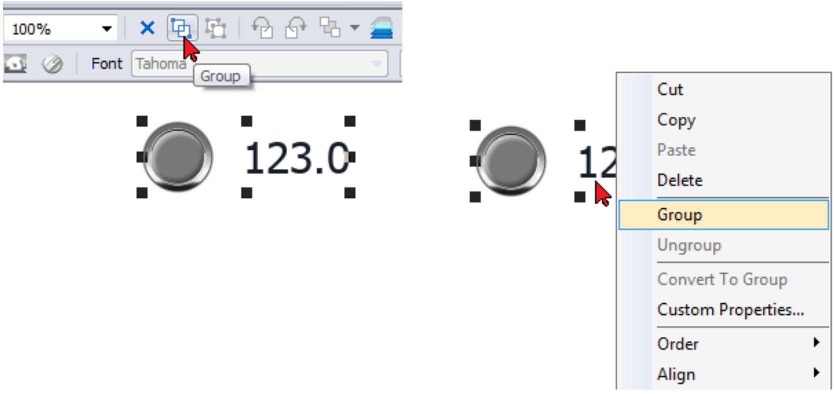
The toolbar allows to:

- 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**.

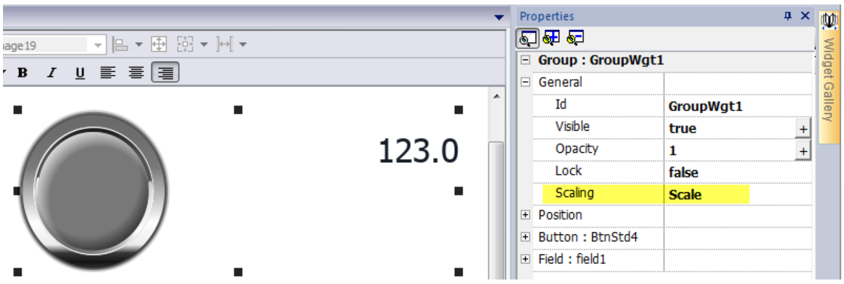


Tip: Double click to enter the group editing mode. In group mode only the group widgets are editable and selectable. All other widgets are partially hidden

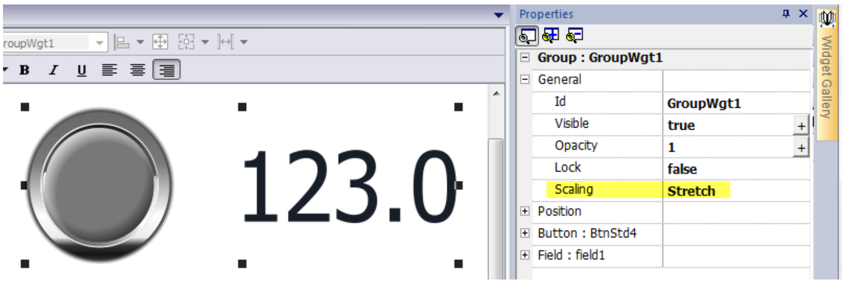
Resizing grouped widgets

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



Group : GroupWgt1	
General	
Id	GroupWgt1
Visible	true
Opacity	1
Lock	false
Scaling	Scale
Position	
Button : BtnStd4	
Field : field1	



Group : GroupWgt1	
General	
Id	GroupWgt1
Visible	true
Opacity	1
Lock	false
Scaling	Stretch
Position	
Button : BtnStd4	
Field : field1	

Grid Layout Group

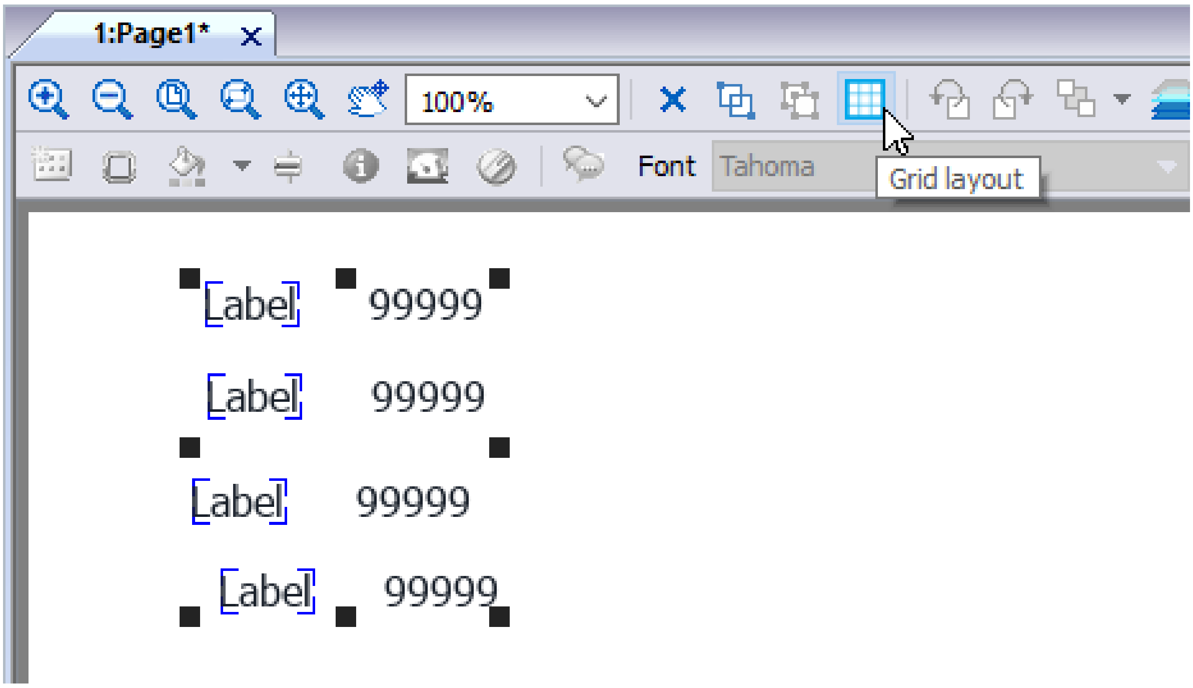
The grid layout add the possibility to configure the spatial relationships among the widgets of the group.

To create a grid layout:

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

or

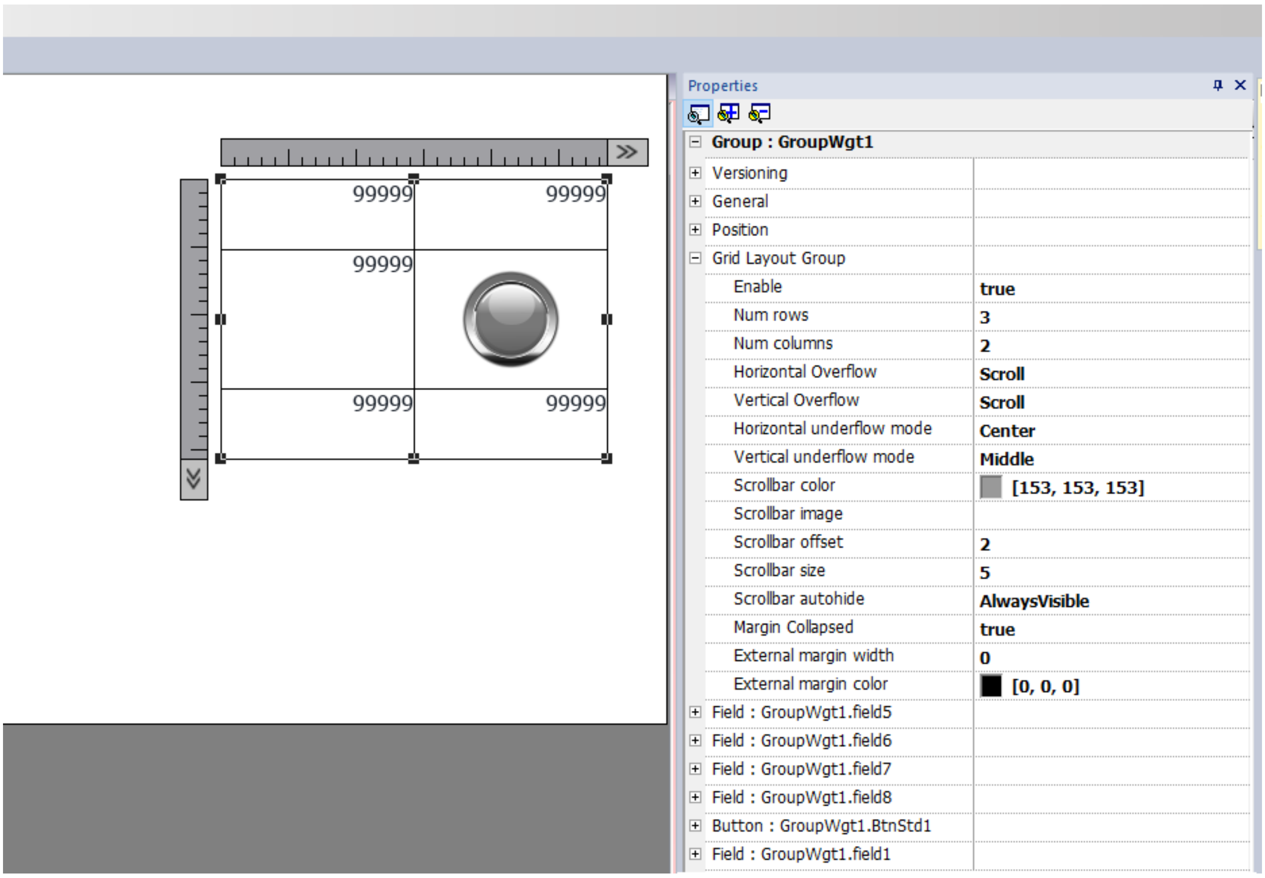
- Select the widgets that will be inside the table and click the “Grid Layout” button on page toolbar. The selected widgets will be aligned and collected inside a group with the grid layout property enabled.



There are several elements associated with the grid layout that can be configured:

- Grid properties
- Rows, Columns Properties
- Cells Properties

Grid Properties



Properties

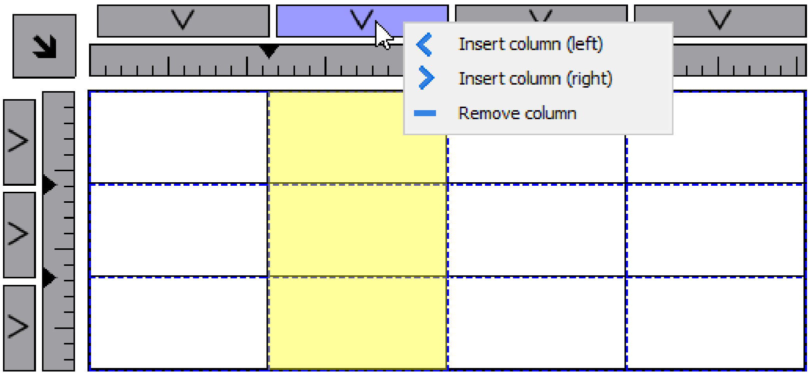
Group : GroupWgt1	
⊞ Versioning	
⊞ General	
⊞ Position	
⊟ Grid Layout Group	
Enable	true
Num rows	3
Num columns	2
Horizontal Overflow	Scroll
Vertical Overflow	Scroll
Horizontal underflow mode	Center
Vertical underflow mode	Middle
Scrollbar color	[153, 153, 153]
Scrollbar image	
Scrollbar offset	2
Scrollbar size	5
Scrollbar autohide	AlwaysVisible
Margin Collapsed	true
External margin width	0
External margin color	[0, 0, 0]
⊞ Field : GroupWgt1.field5	
⊞ Field : GroupWgt1.field6	
⊞ Field : GroupWgt1.field7	
⊞ Field : GroupWgt1.field8	
⊞ Button : GroupWgt1.BtnStd1	
⊞ Field : GroupWgt1.field1	

Parameter	Description
Enable	Enable the grid layout. A grid will be generate around the widgets of the group
Num rows **Num columns**	Number of rows and columns of the grids. Rows and columns can be removed only if their cells are empty .
Horizontal overflow **Vertical overflow**	This parameter define the behavior of the grid when it is too small to contain all rows and columns. • Hidden Rows and columns that do not fit into the grid are not displayed • Visible The grid can not be made smaller than the minimum size required to contain all defined rows and columns • 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.

Parameter	Description
Horizontal underflow **Vertical underflow**	This parameter defines the behavior of the grid when it is larger than the size defined for the rows and columns • Blocked The grid can not be made larger than the maximum size of rows and columns • Left, Center, Right - Top, Middle, Bottom Defines the position of the widgets when cells are bigger than the maximum defined sizes
Scrollbar color **Scrollbar image** **Scrollbar offset** **Scrollbat size** **Scrollbar autohide**	Parameters to define look and position of the scroll bars
Margin collapsed	Collapse all left-right and top-botton margin using the parameters of the stroke with greater width.
External margin width **External margin color**	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.



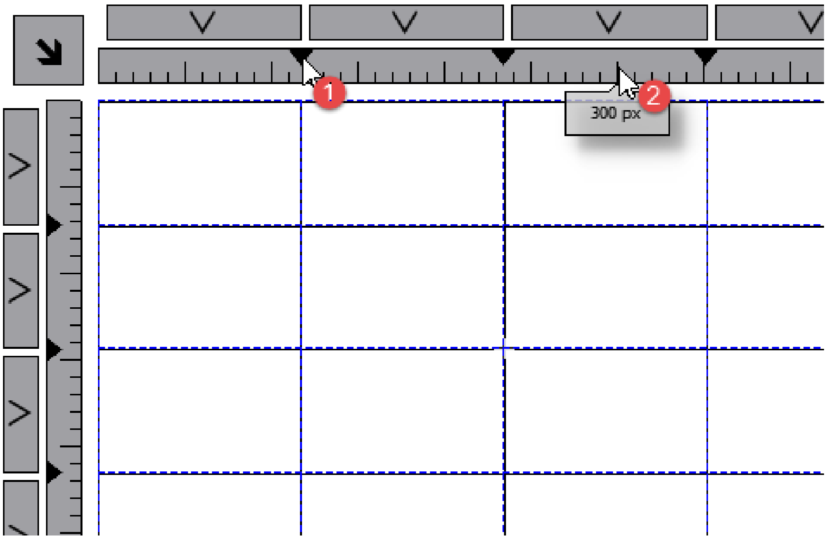
Merge or split rows or columns

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)

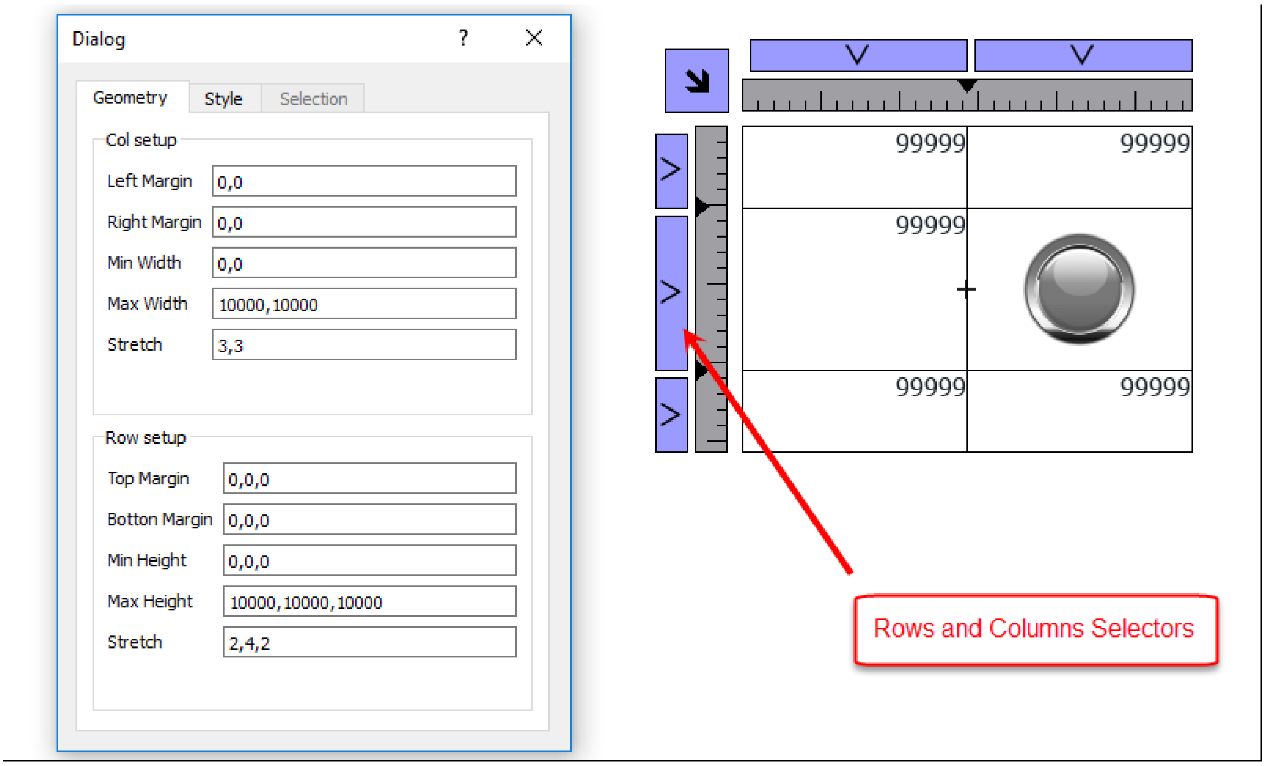
  Note that merge is possible only with an empty row or column.

- 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.



Col setup

Left Margin: 0,0

Right Margin: 0,0

Min Width: 0,0

Max Width: 10000,10000

Stretch: 3,3

Row setup

Top Margin: 0,0,0

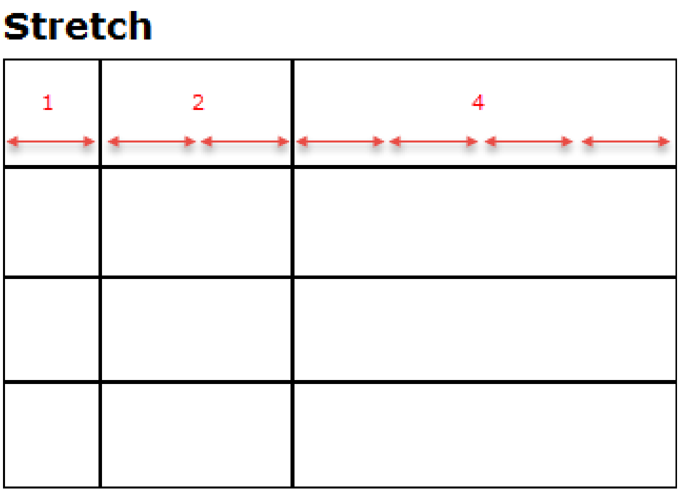
Botton Margin: 0,0,0

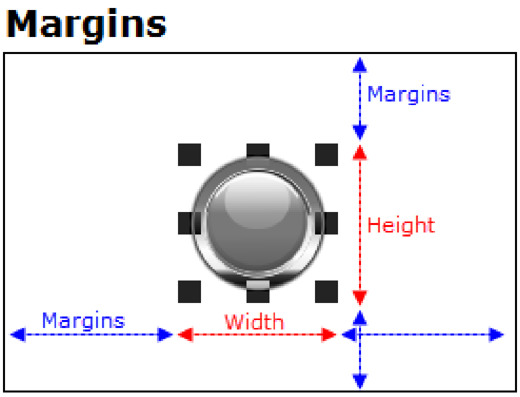
Min Height: 0,0,0

Max Height: 10000,10000,10000

Stretch: 2,4,2

Rows and Columns Selectors





Geometry parameters

Parameter	Description
Left margin **Right margin**	Distance of the widget from the border of the cell
Min width **Max width**	Min/Max width that widget can assume when the cell is stretched
Stretch	Defines the relationship between the widths of the columns that will be maintained if the grid is stretched
Top margin **Bottom margin**	Distance of the widget from the border of the cell
Min height **Max heighty**	Min/Max height that widget can assume when the cell is stretched
Stretch	Defines the relationship between the heights of the rows that will be maintained if the grid is stretched

Style parameters

Parameter	Description
Left stroke width **Right stroke width** **Top stroke width** **Bottom stroke width**	Strokes width
Left stroke color **Right stroke color** **Top stroke color** **Bottom stroke color**	Strokes color
Background color	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.

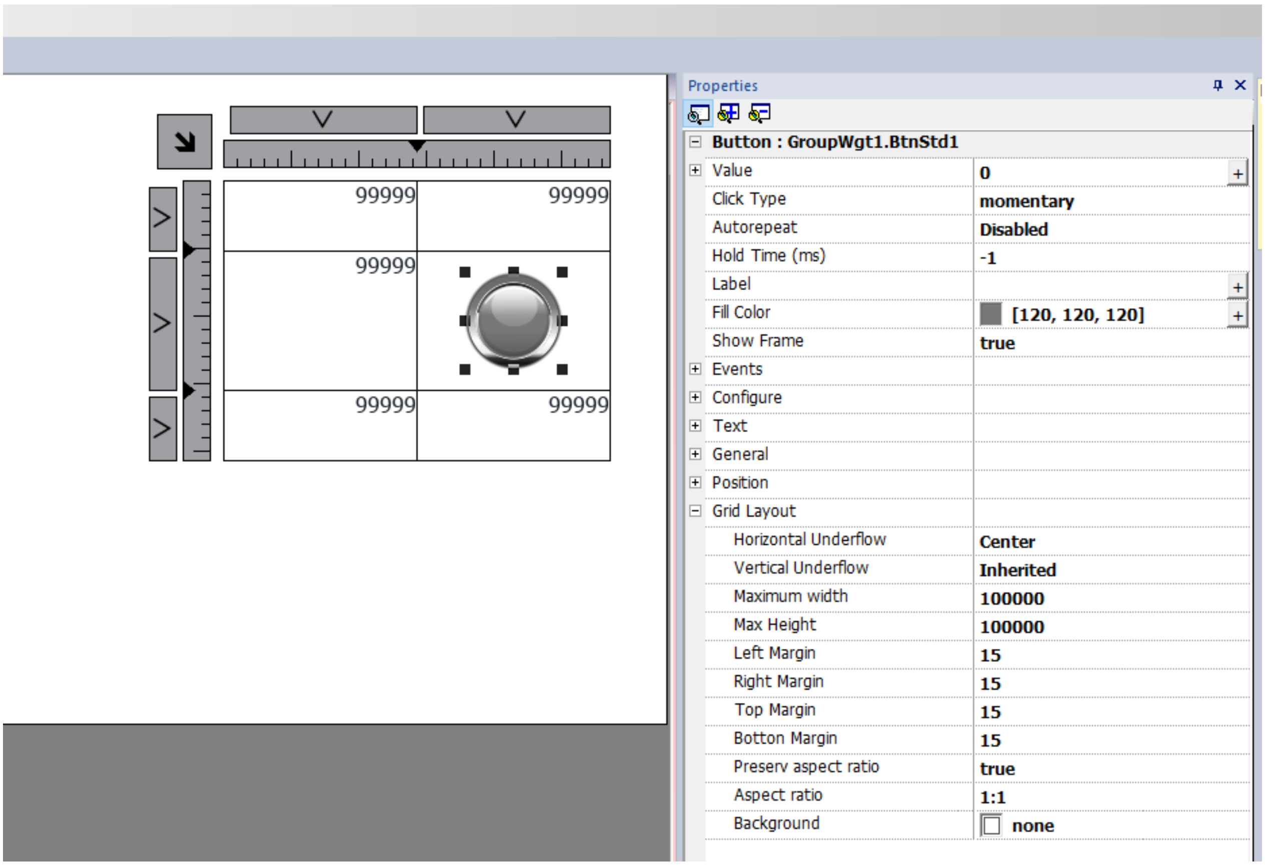
Selection parameters

The selection parameters is available only when the grid is used inside a Table Widget (see "Table widget" on page 431 for details)

Parameter	Description
Forground color **Background color** **Stroke color**	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.



Button : GroupWgt1.BtnStd1

Value	0
Click Type	momentary
Autorepeat	Disabled
Hold Time (ms)	-1
Label	
Fill Color	[120, 120, 120]
Show Frame	true
Events	
Configure	
Text	
General	
Position	
Grid Layout	
Horizontal Underflow	Center
Vertical Underflow	Inherited
Maximum width	100000
Max Height	100000
Left Margin	15
Right Margin	15
Top Margin	15
Botton Margin	15
Preserv aspect ratio	true
Aspect ratio	1:1
Background	□ none

Parameter	Description
Horizontal underflow **Vertical underflow**	This parameter defines the behavior of the widget when the cell is larger than the size defined for widget. • Inherited Inherits the value used for the row or column • Left, Center, Right - Top, Middle, Bottom Defines the position of the widgets when cells are bigger than the maximum defined sizes
Max width **Max height**	Overwrite global grid parameters
Left margin **Right margin** **Top margin** **Bottom margin**	Overwrite global grid parameters Additional pixels that are added to the total margin.
Preserve aspect ratio	Preserve aspect ration of the widget
Aspect ratio	Available only when "*Preserve aspect ratio*" is true
Background	Background color of cell

Printing report

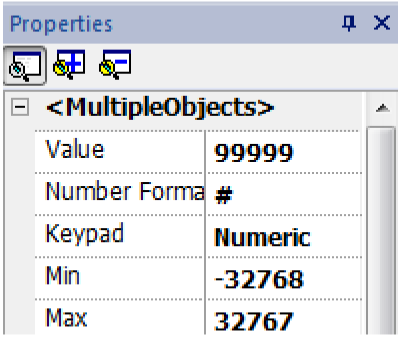
Note the grid layout is available even inside the print report gallery.

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.



<MultipleObjects>

Value	99999
Number Forma	#
Keypad	Numeric
Min	-32768
Max	32767

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

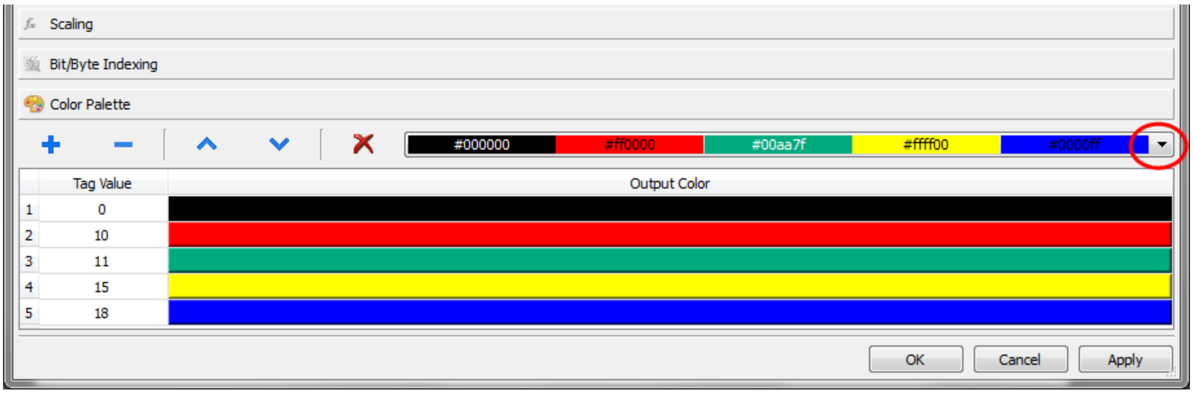
Changing fill color property according to tag values

AGI Creator 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.



	Tag Value	Output Color
1	0	⬛
2	10	🟥
3	11	🟩
4	15	🟨
5	18	🟦

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.

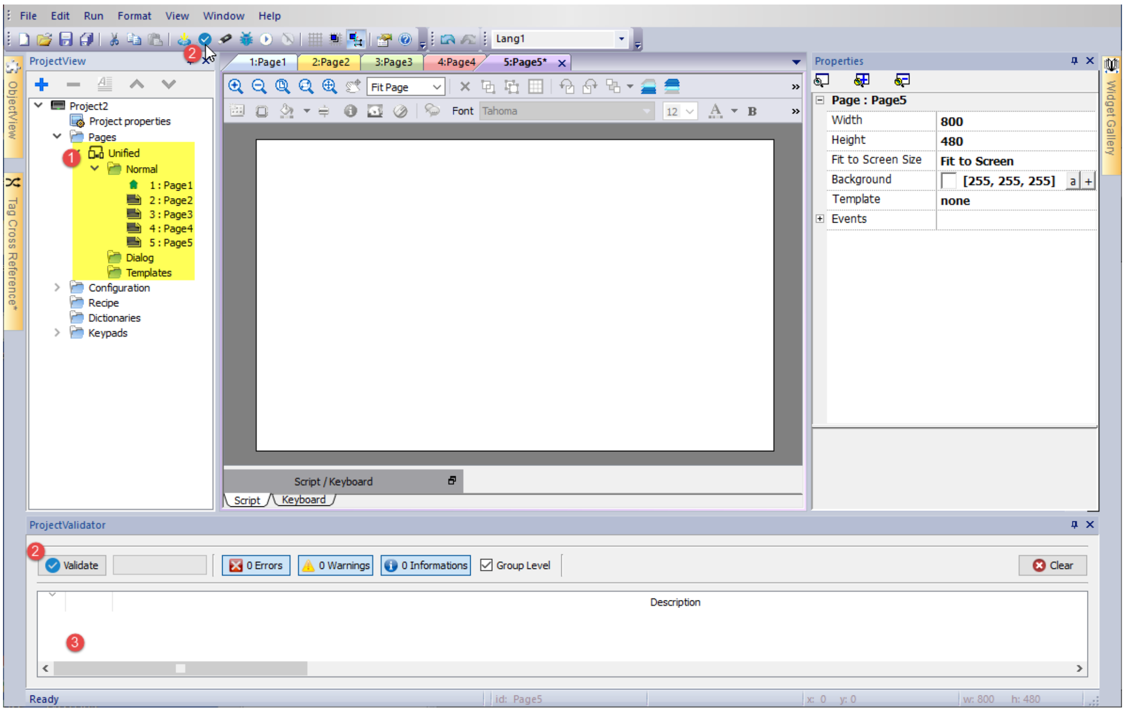
6 Pages

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.

Unified pages

Starting from AGI Creator 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



Page : Page5

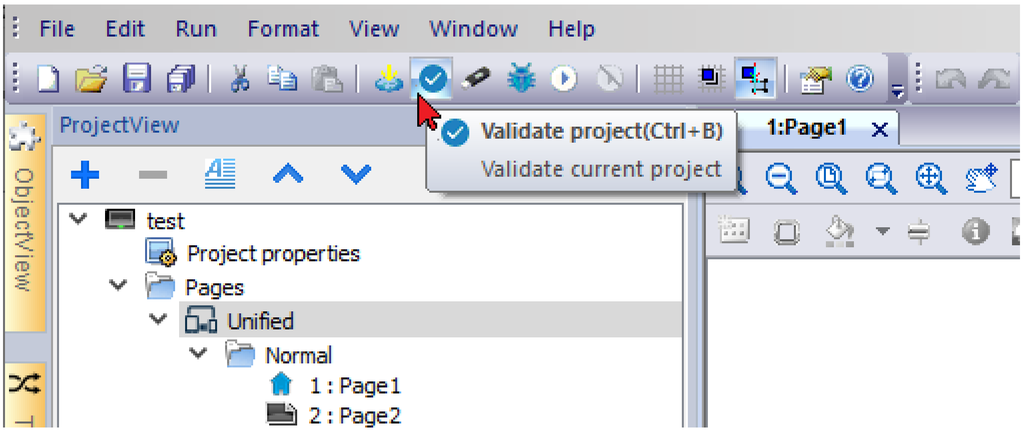
Width	800
Height	480
Fit to Screen Size	Fit to Screen
Background	[255, 255, 255]
Template	none
Events	

Ready | id: Page5 | x: 0 y: 0 | w: 800 h: 480

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.

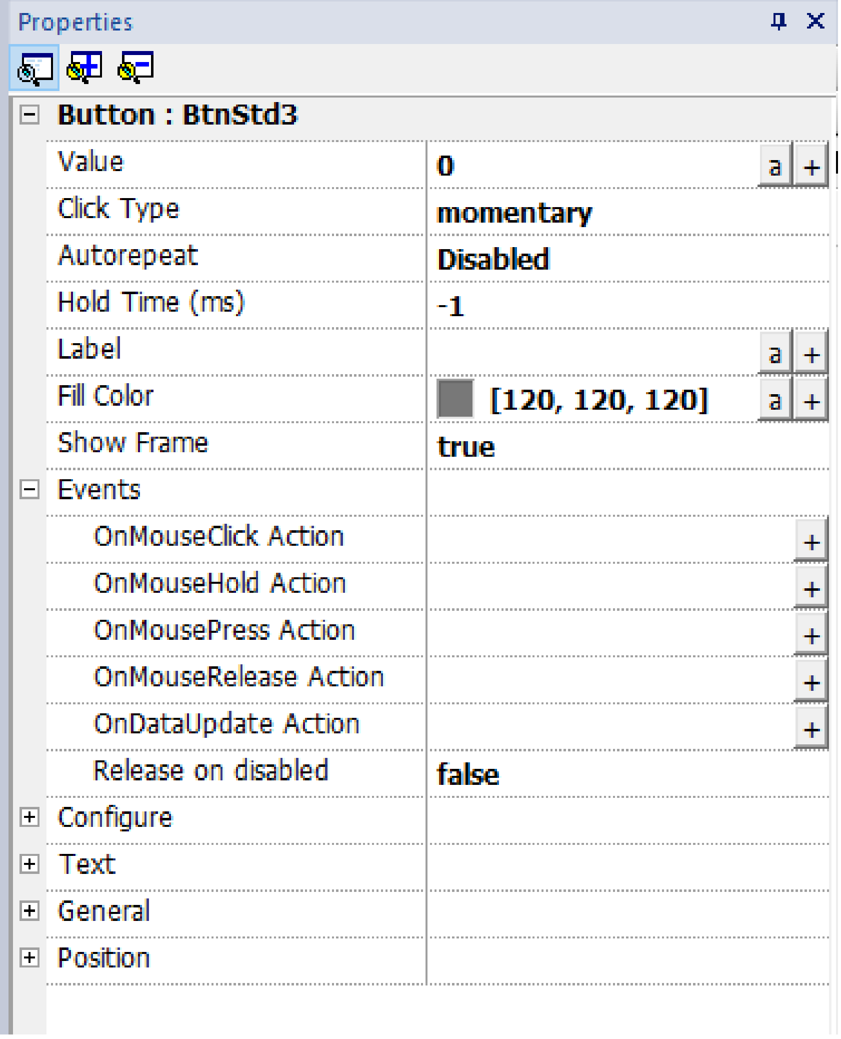
Note that the current version of the Project Validator checks the widget's web compatibility. It is not checking the entire project (e.g. missing tags or Javascript errors)



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.



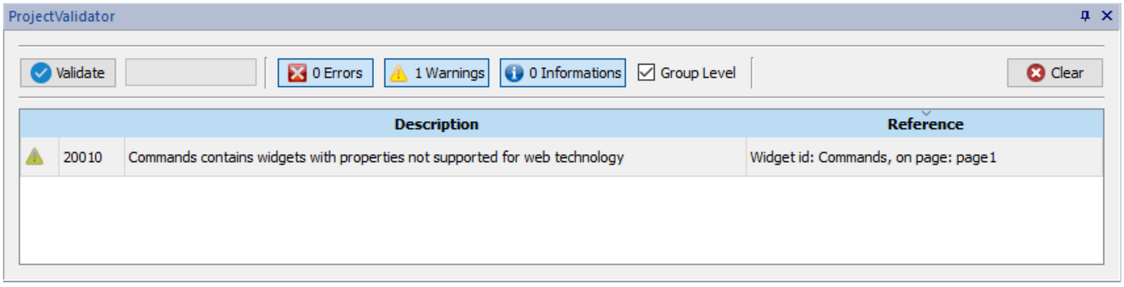
Properties

Button : BtnStd3

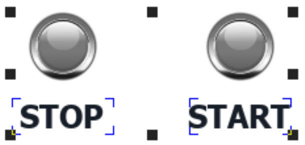
Value	0
Click Type	momentary
Autorepeat	Disabled
Hold Time (ms)	-1
Label	
Fill Color	[120, 120, 120]
Show Frame	true
Events	
OnMouseClick Action	
OnMouseHold Action	
OnMousePress Action	
OnMouseRelease Action	
OnDataUpdate Action	
Release on disabled	false
Configure	
Text	
General	
Position	

Group Level

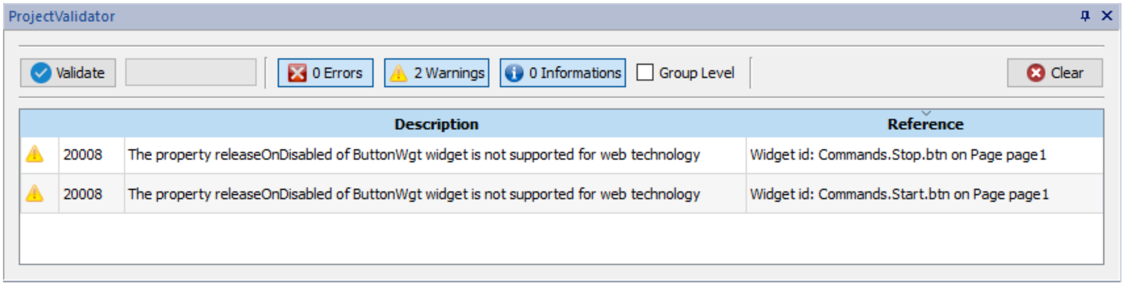
When the "Group Level" is checked, the Project Validator will report the group name that contains one or more widgets with the unsupported properties. A double click will select the grouped widget.



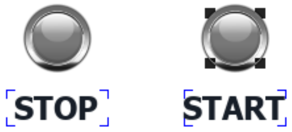
		Description	Reference
⚠	20010	Commands contains widgets with properties not supported for web technology	Widget id: Commands, on page: page1



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.



		Description	Reference
⚠	20008	The property releaseOnDisabled of ButtonWgt widget is not supported for web technology	Widget id: Commands.Stop.btn on Page page1
⚠	20008	The property releaseOnDisabled of ButtonWgt widget is not supported for web technology	Widget id: Commands.Start.btn on Page page1



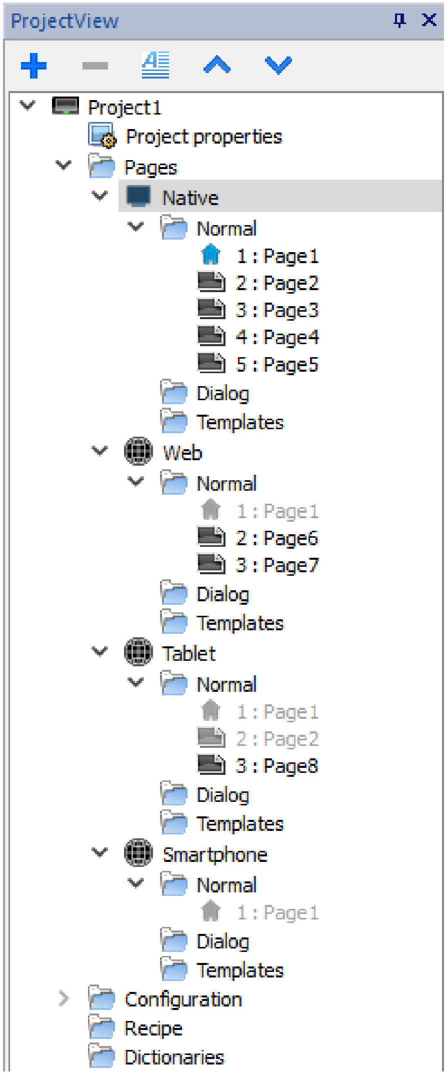
Differentiated pages

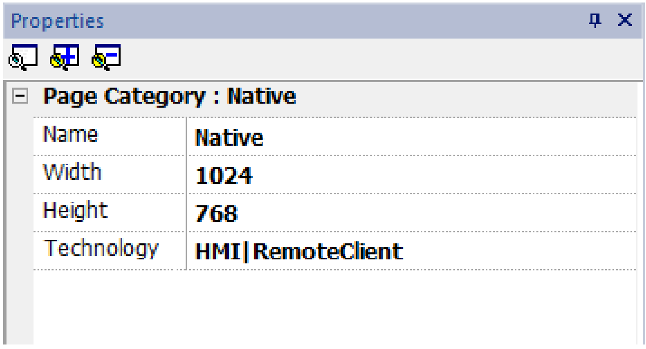
If a project needs to have different pages for the HMI device, web client, tablet client, etc., there is the possibility to add different folder to contain the pages to use on the different clients. Right click on the page folder to add a new category of pages. For each category, you have to define the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

Property	Description
Name	The category name
Width, Height	The default size used when create a new page

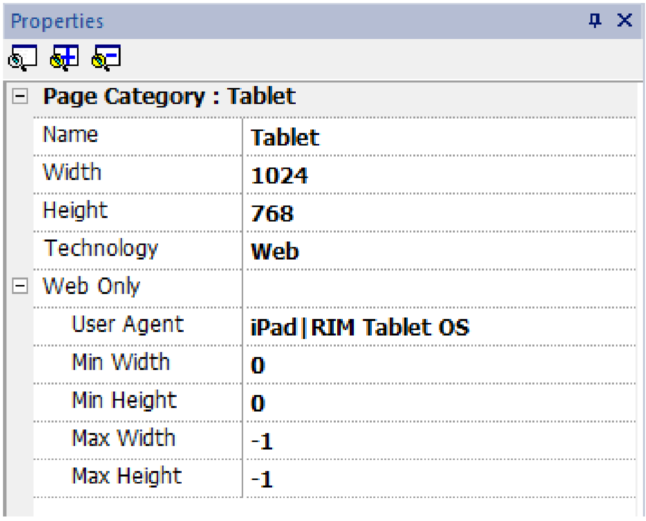
Property	Description
Technology	Identify the clients that can use these pages. It can be a combination of: • Local HMI Device • Remote AGI Client • Web Clients (PC, Tablet, Smart Phones, etc.)
User Agent	It is a regular expression that identifies the web browsers that can display the pages of the category. The user-agent of the web client has to match with this parameter. Example: .* Anything (all web clients) Android Only Android web clients Android|iPhone Only Android or iPhone web clients
Min Width **Min Height** **Max Width** **Max Height**	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.





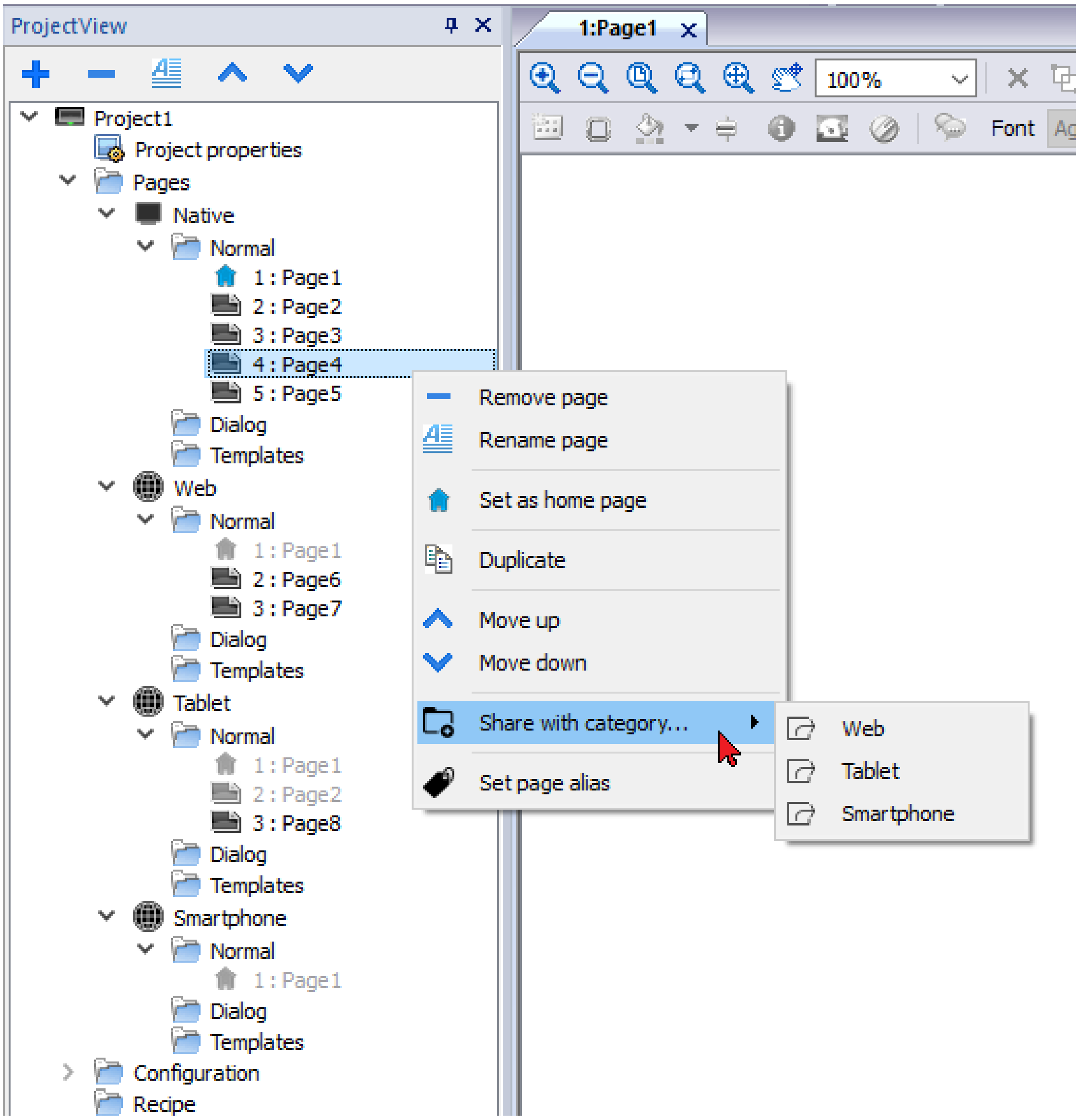
Page Category : Native	
Name	Native
Width	1024
Height	768
Technology	HMI|RemoteClient



Page Category : Tablet	
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.

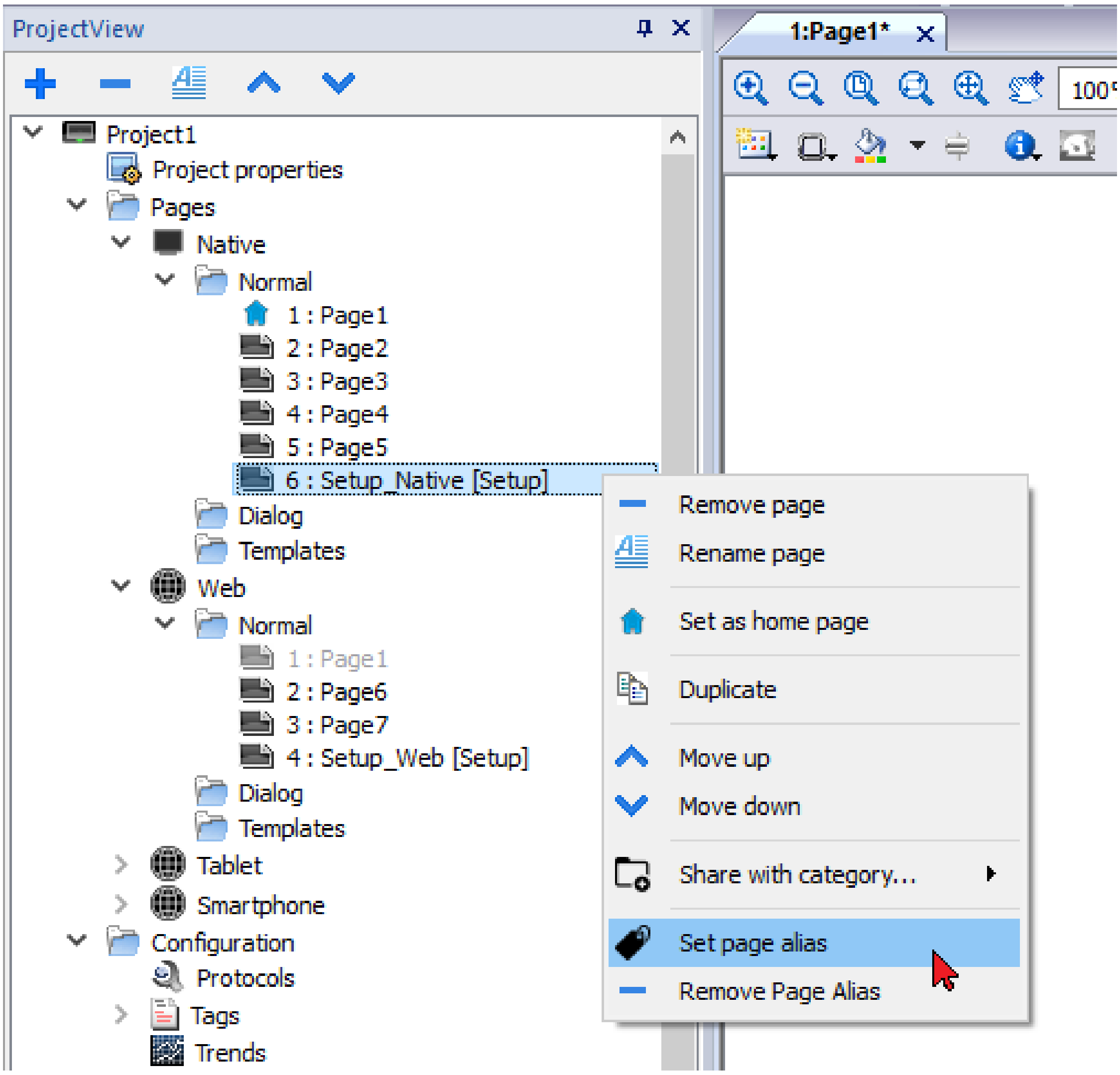
Alias pages

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

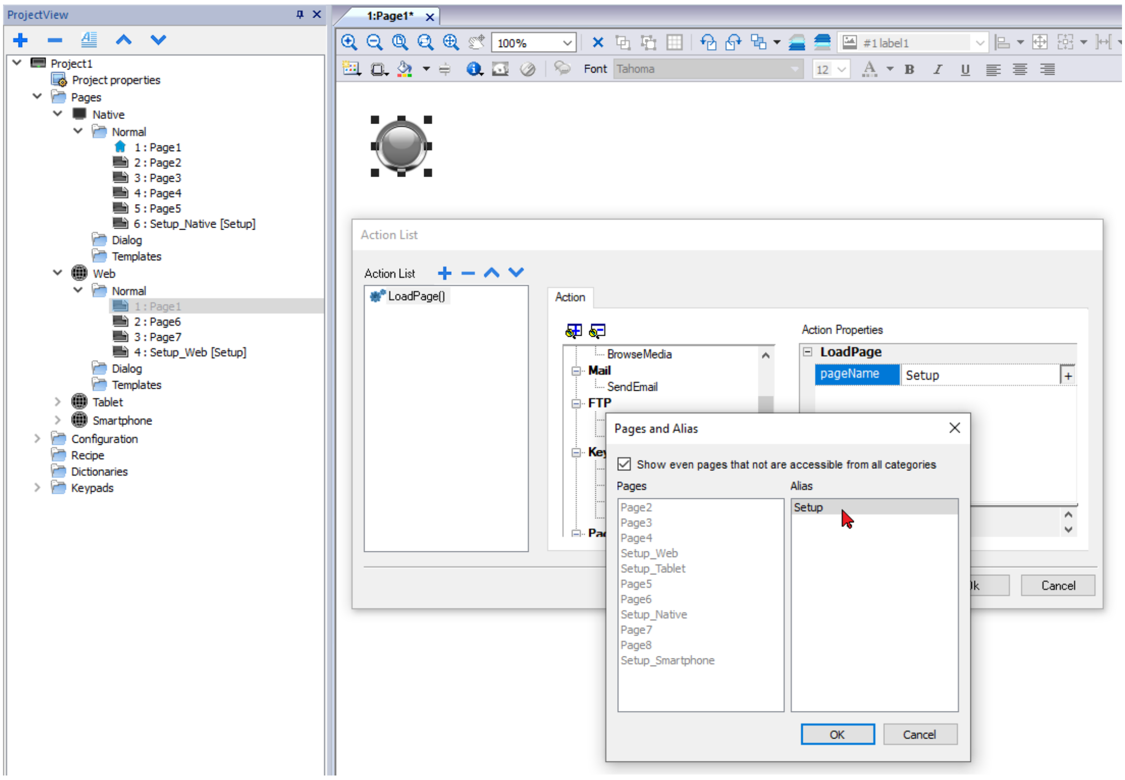
For example, you can have a shared "Page1" common to all categories. Page1 will be shown on both the HMI device and on Web Client, but from this page, you need to add a macro to load a customize setup page. This means a macro that load the page "Setup_Native" on HMI device or a different page "Setup_Web"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



LoadPage

pageName: Setup

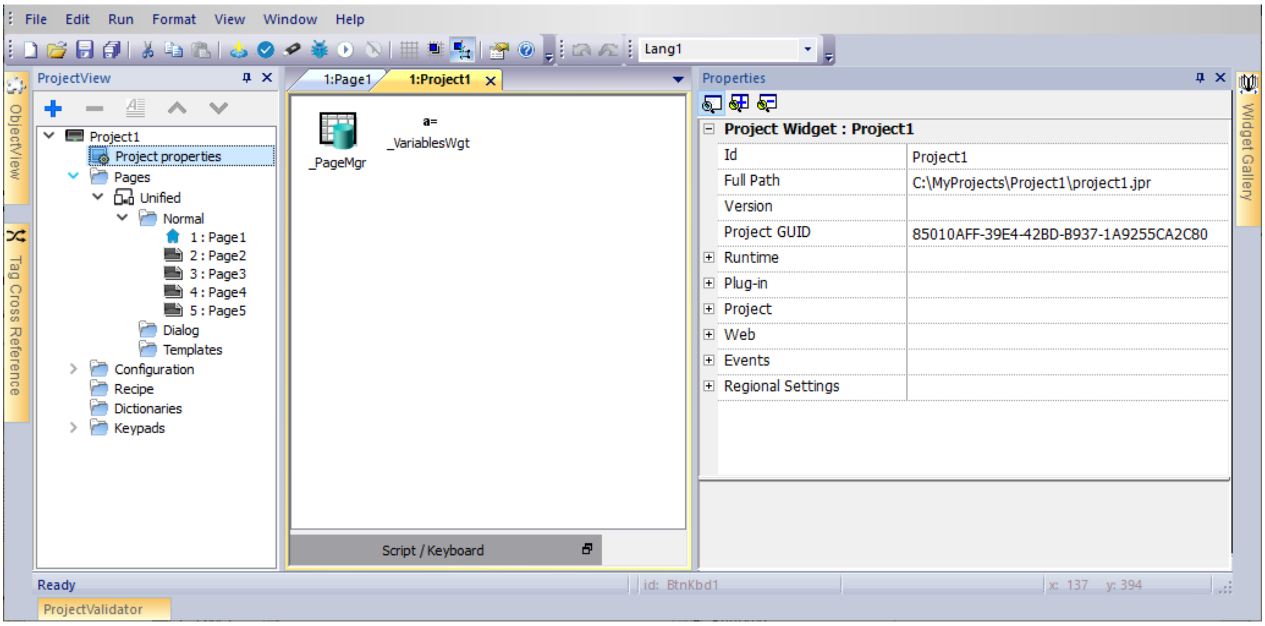
☑ Show even pages that not are accessible from all categories

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.

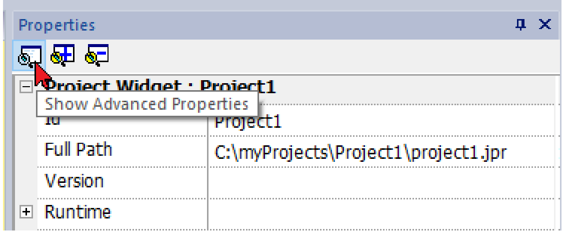


Project Widget : Project1	
Id	Project1
Full Path	C:\MyProjects\Project1\project1.jpr
Version	
Project GUID	85010AFF-39E4-42BD-B937-1A9255CA2C80
Runtime	
Plug-in	
Project	
Web	
Events	
Regional Settings	

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.



Project Widget : Project1	
Id	Project1
Full Path	C:\myProjects\Project1\project1.jpr
Version	
Runtime	

Available properties

Property	Description
Id	Project name (read only)
Full Path	Project path (read only)
Project GUID	Project unique identifier (read only)

Property	Description
Version	The Version field is available for users to report the project version
+ Runtime	Properties related with the application runtime. See "Runtime" belowfor details
+ Plug-In	Optional modules. See "Plug-in" on page 85 for details
+ Project	Properties related with the project. See "Project" on page 86 for details
+ Web	Properties related with the web interface. See "Web" on page 90 for details
+ Events	Global events. See "Events" on page 91 for details
+ Regional Settings	Definition of date format, list separator, thousand and decimal symbol of number. See "Regional Settings" on page 91 for details

Project ID, Project GUI and Project Version are available from system variables. See "Default variables" on page 149 for details.

Runtime

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

Property	Description
Context Menu	Define how context menu should appear in the HMI project. **on delay** = context menu appears touching/pressing and holding for a few seconds an empty area of the runtime screen, or via **Context menu** action **on action** = context menu appears only via **Context menu** action. See "Widget actions" on page 219 for details.
Developer Tool	Enable/disables a collection of runtime debugging utility tools.
Buzzer on Touch	Enables buzzer when touching a widget on HMI device screen. Supported widgets: • buttons • hotspots • needles • fields • external keys • combo boxes • tables items • control list items *On Windows CE panels, available for from v1.76 ARM / 2.79 MIPS.*

Property	Description
Buzzer duration	Default 200 ms
Keyboard	Enables the use of keyboard macros at runtime when using external keyboards.
JavaScript Debug	Enables the JavaScript debugger at runtime for the current project.
Allow JS Remote Debugger	Enables JavaScript remote debugger for current project. Remote debugging not supported on AGI Client.
Image DB enable	Activates an engine used by the Runtime to optimize project performance. **WARNING: This property should only be disabled by technical support for debugging purposes since this might reduce performance at runtime.**
FreeType Font Rendering	Switches to FreeType the font rendering used by AGI Creator 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.
Communication icon delay (ms)	Delay before display the communication error icon (default is 0 mSec) The special value -1 is meaning always disabled
Fast Boot	When fast boot is enabled and the User Interface is started before the background server • Default: User Interface is loaded after the background server is ready to use • Fast UI: User Interface is loaded before loading the background server

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 551)

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

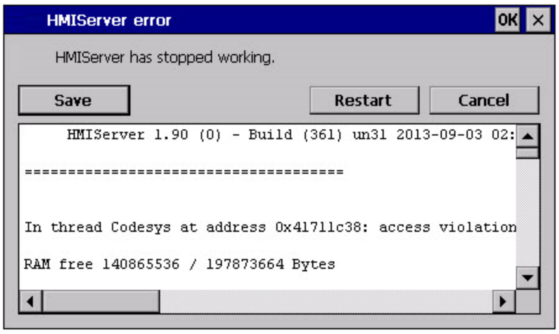
Tool	Description
Show/Hide all	Shows a dialog containing information about device status like CPU load, memory usage, event queues.
CPU statistics	Shows information on CPU load. See "CPU Statistics" on the facing page.
Memory statistics	Shows information about system RAM . A negative value indicates that free memory is decreasing.
Event queues	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.
Timelog summary	Show page loading time.
Embed window	Allows embedding in runtime the scene or leave the developer tool window as a standalone window (dialog).
Reset queue stats	Resets statistical information on event queues.
Disable watchdog	Disable the watchdog function and prevents system restart in case of freeze or crash of services.
Ignore exceptions	Disables crash report function, exceptions are not saved in the crash report window.

Tool	Description
Launch VNC	Launches the VNC server if available in runtime. VNC server is available as a plugin for Windows CE runtime only.
Profiling	Measures the time spent for loading/rendering the active page. See "Profiling" below

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.

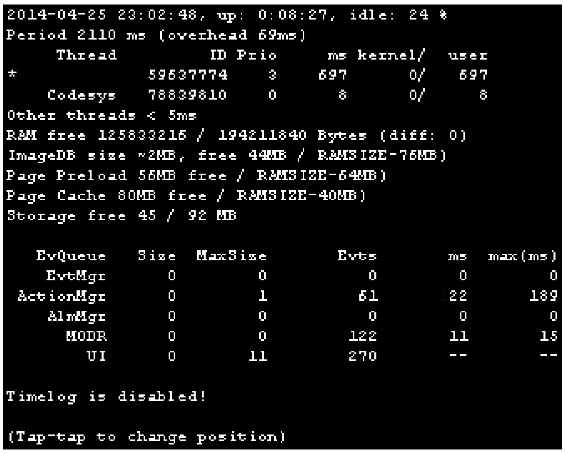


HMIServer has stopped working.

```
      HMIServer 1.90 (0) - Build (361) un31 2013-09-03 02:
====================================

In thread Codesys at address 0x41711c38: access violation

RAM free 140865536 / 197873664 Bytes
```

Important: Save this file for technical support.

CPU Statistics



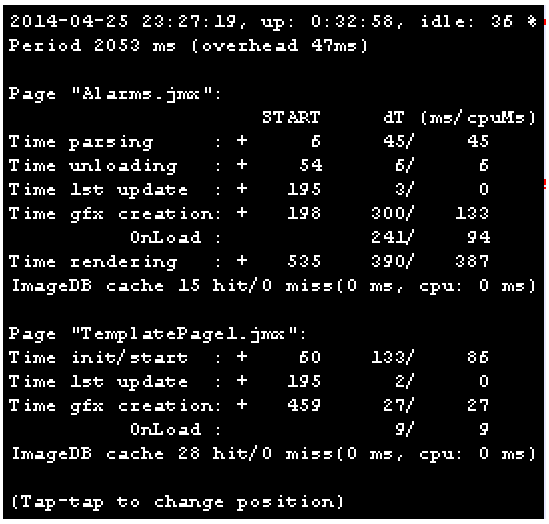
```
2014-04-25 23:02:48, up: 0:08:27, idle: 24 %
Period 2110 ms (overhead 69ms)
     Thread          ID Prio     ms kernel/  user
*              59637774    3    697      0/   697
    Codesys    78839810    0      8      0/     8
Other threads < 5ms
RAM free 125833216 / 194211840 Bytes (diff: 0)
ImageDB 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)
    EvtMgr      0        0           0        0        0
 ActionMgr      0        1          61       22      189
    AlmMgr      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 1st 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/     86
Time 1st 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
Enable timelog	Enable timelog capture. Timing will be visible inside the “Timelog summary” window.
Save timelog to file	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. **Important: The execution of this function may reduce page change performance.**
Overlay OnLoad times **Overlay Rendering times**	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
Select overlay color	Select the overlay color to use

Timelog data

Data	Description
Time parsing	Time spent parsing current page. Depends on page complexity/number of widgets.
Time gfx creation	Time spent for image rendering. Mainly related to the *Onload* method.
Time rendering	Time spent rendering the page.
Time unloading	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 AGI Creator 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
WebKit	Module required by WebBrowser widget
TextEditor	Module required by TextEditor widget
PDF Reader	PDF Reader
VNC Server	VNC Server

Note: Not all software plug-in modules are compatible with all HMI device platform.

Once enabled, software plug-in become part of the runtime. Use AGI Creator 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
Home Page	The first page loaded at runtime (after log-in page if security is enabled in project). When security is enabled, you can specify a different homepage for each groups of users. In this case this setting is ignored. See "User management and passwords" on page 333 for details.
PageWidth **PageHeight**	Defines the default size in pixel of an HMI page. Default is the display resolution of the HMI device model selected when creating the project.
Display Mode	Defines HMI device orientation.
Project Type	Defines HMI device type for the project. According to the model, some project features and properties are automatically adjusted. **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.”**
Panel Memory	Size of the available internal panel memory.
PageRequest **CurrentPage** **SyncOptions**	You can synchronize pages shown on the HMI Runtime and AGI Client from a controller such as a PLC. 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. See the "Web" on page 90 for the Web Browser support
Hold Time	Defines the values for hold time and auto repeat time for buttons and external keyboards.

<table>
<tr><th>Property</th><th>Description</th></tr>
<tr><td><b>Autorepeat Time</b></td><td>ℹ Note: These properties can be redefined for each button or key in their widget property table.</td></tr>
<tr><td><b>Hide Project Loading at boot</b></td><td>When hidden, the splash screen stay on the screen until the application is ready to run.</td></tr>
<tr><td><b>Target Zoom Factor</b></td><td>It is the zoom factor of the HMI device that will be applied when project is loaded at runtime.
<table>
<tr><td><b>Range</b></td><td>0.3–2.9</td></tr>
<tr><td><b>Fit to screen</b></td><td>-1 = Fit to screen size
ℹ 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></td></tr>
<tr><td><b>Background color option</b></td><td>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)
<table>
<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></td></tr>
<tr><td><b>Signature</b></td><td>Algorithm to use to signing<ul><li>sha256</li><li>sha1</li></ul></td></tr>
<tr><td><b>Gesture Passthru Enabled</b></td><td>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.</td></tr>
<tr><td><b>Gesture Passthru Delay (ms)</b></td><td>When enabled, the gesture events are passed to underlying widgets after this delay (see "Gesture events pass thru" on page 415 for details)</td></tr>
<tr><td><b>Gesture Multitouch</b></td><td>Enable multi touch gestures<ul><li>false</li><li>true (default)</li></ul>ℹ 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.</td></tr>
<tr><td><b>On Access Denied</b></td><td>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.<ul><li>None</li></ul></td></tr>
</table>

Property	Description
	• Show Icon
ComboBox View Mode	Select the visualization mode of all the Combo Box widgets of the project (see "Combo Box widget “full screen” mode with images" on page 404 for details) • Context Classic view with drop-down menus • Full screen Enhanced view with configurable texts and images that will pop up in the middle of the screen for easy scroll and selection.

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 AGI Client or to control an HMI project from a controller such as a PLC.

The following properties can be customized:

Property	Description
PageRequest	Page to be shown on the HMI device and on AGI 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.
CurrentPage	Page number displayed on the HMI device or on AGI Client or on both. Attached tag must be available at least as a Write resource and must have integer data type.
SyncOptions	Synchronization of project pages with the value contained into the **CurrentPage** property. Options can be: • **disable**: page number value is ignored, • **local**: page number displayed on HMI, • **remote** : page number displayed on AGI Client. • **local + remote**: page number displayed on HMI and on AGI Client, if different pages are displayed the last page loaded is considered.

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

Set properties as follows:

PageRequest	attached to tag "A"
CurrentPage	empty
SyncOptions	disable

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

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

Set properties as follows:

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

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

Set properties as follows:

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

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

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

Set properties as follows:

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

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

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

Set properties as follows:

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

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

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

Set properties as follows:

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

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

Web

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

<table>
<tr><th>Property</th><th>Description</th></tr>
<tr><td><b>Web Inactivity Timeout</b></td><td>Defines a timeout for AGI Web client. When the timeout expires without any activity the current user is logged out.
<table>
<tr><td><b>Range</b></td><td>1–86400 s (form 1 s to 24 h)</td></tr>
<tr><td><b>Default value</b></td><td>600 s</td></tr>
<tr><td><b>Values</b></td><td>0 = disabled</td></tr>
</table></td></tr>
<tr><td><b>Web Icon</b></td><td>The favorite icon associate at the web pages</td></tr>
<tr><td><b>Refresh Time</b></td><td>Defines the refresh time for the communication between the runtime and AGI Web clients.
<table>
<tr><td><b>Range</b></td><td>500–10000 ms</td></tr>
<tr><td><b>Default value</b></td><td>3000 ms</td></tr>
</table></td></tr>
<tr><td><b>Browser Optimization</b></td><td>
<table>
<tr><td><b>true</b></td><td>Web engine optimization enable (default)</td></tr>
<tr><td><b>false</b></td><td>Web engine optimization disable (useful for old browsers that not support the web engine optimization)</td></tr>
</table></td></tr>
<tr><td><b>Enable Global JavaScript for remote</b></td><td>Define if the JavaScrip 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.
<table>
<tr><td><b>None</b></td><td>Will not be executed on remote clients (run only inside the local HMI device)</td></tr>
<tr><td><b>Client</b></td><td>Will be executed on AGI Client</td></tr>
<tr><td><b>Web</b></td><td>Will be executed on Web client</td></tr>
<tr><td><b>Both</b></td><td>Will be executed on both AGI Client and Web clients</td></tr>
</table></td></tr>
</table>

Property	Description
Max Bandwidth (Kbs)	Limit for maximum data sent by server (useful for old slow browsers). Set to 0 to use all the available bandwidth (default)
Web clients connection mode	**Auto** – The connection mode is selected by the client (default) **SSE** – Force the Server-Sent Events mode **Long Polling** – Force the Long-polling mode
WebPageRequest	You can synchronize pages shown on the AGI Web Clients from a controller such as a PLC. Page to be shown on the AGI Web 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.
Web Communication icon delay (ms)	Delay before display the communication error icon (default is 0 mSec) The special value -1 is meaning always disabled

The project.getClientType() can be used to retrieve the running client type. See "Project object" on page 487 inside JavaScript chapter for additional details.

Events

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

Property	Description
OnWheel	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 **BiStep** to increase/decrease a tag value. **The project's OnWheel Action is executed only when the OnWheel Action will not overwritten from the loaded page.**

Regional Settings

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

Property	Description
Short date format	The date format to use when user select SHORT-DATE in the date format of the widget
Long date format	The date format to use when user select LONG-DATE in the date format of the widget

Property	Description
List separator	List separator character to use inside the dumped files.
Decimal symbol	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)
Thousand symbol	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 413)

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.

The HMI simulator supports:

- online simulation - in communication with real devices (only for protocols with Ethernet or RS-232 communication),
- offline simulation - simulating tag behavior

The data simulation method is set in the **Simulator** column of the Tag Editor.

Data simulation methods

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

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

See "Adding tags" on page 107 for details.

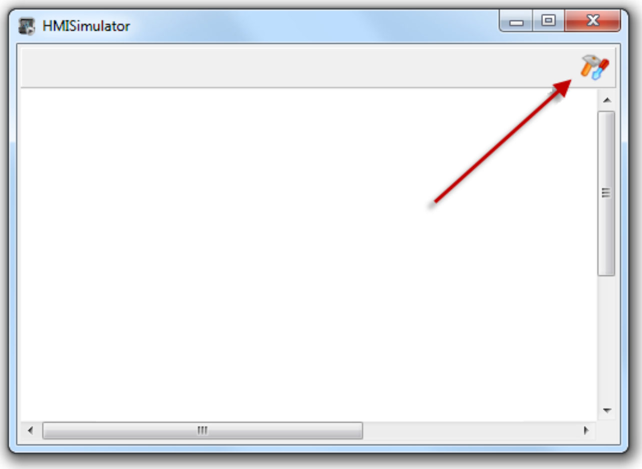
Simulator settings

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

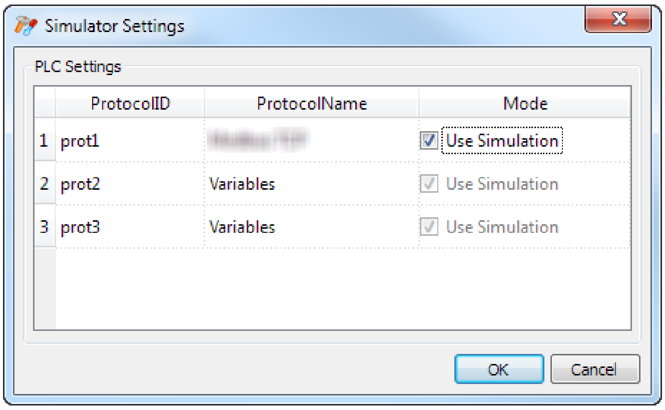
Note: For protocols not supporting communication with external devices, such as the Variables protocol, this option is always disabled.

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.



Simulator Settings

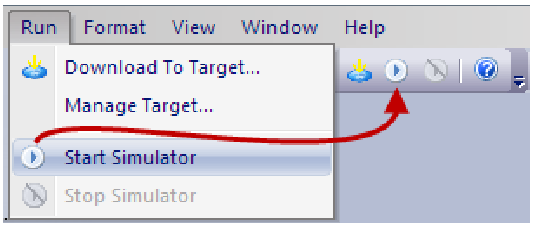
PLC Settings

	ProtocolID	ProtocolName	Mode
1	prot1	[redacted]	☑ Use Simulation
2	prot2	Variables	☑ Use Simulation
3	prot3	Variables	☑ Use Simulation

Launching and stopping the simulator

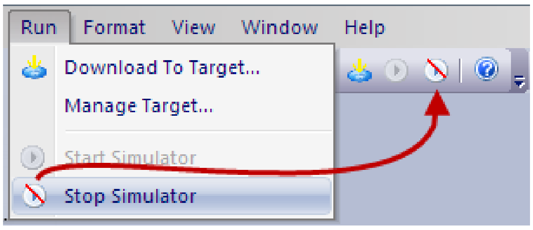
To launch the simulator:

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 AGI Creator 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

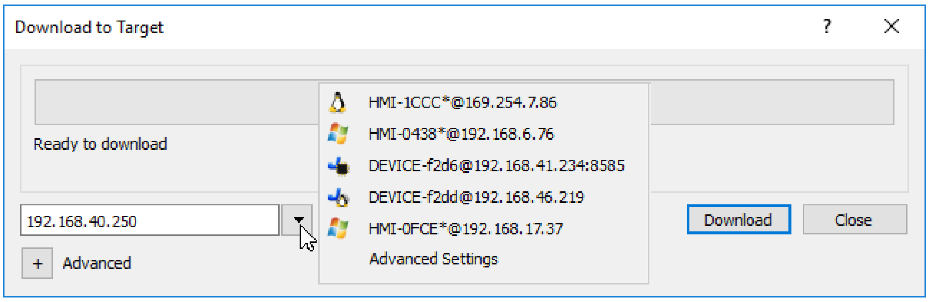
Download to HMI device

*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 8 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.



Download to Target

Ready to download

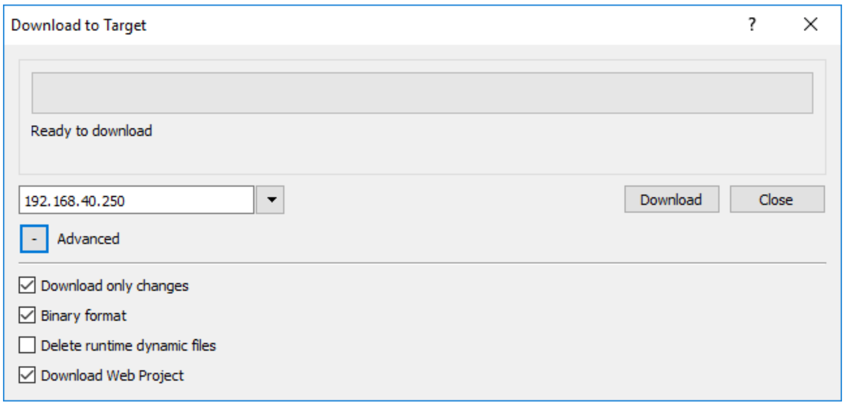
192.168.40.250

   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**: AGI Creator 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



Download to Target

Ready to download

192.168.40.250

- Advanced

☑ Download only changes

☑ Binary format

☐ Delete runtime dynamic files

☑ Download Web Project

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

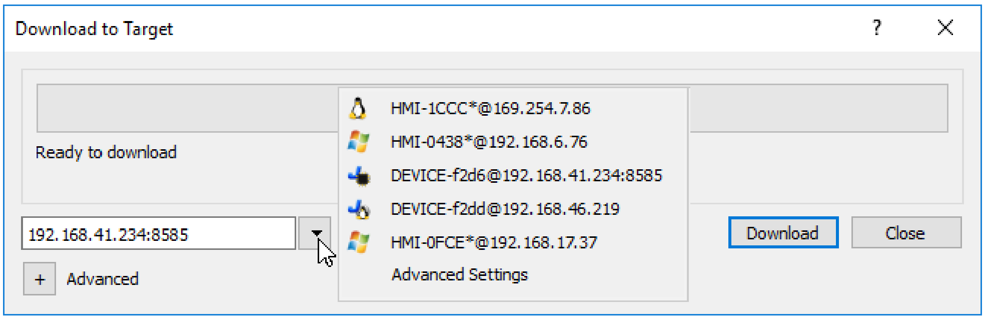
Option	Description
Delete runtime dynamic files	Modified configuration of recipes, users, schedulers, etc. done at runtime will be deleted and overwritten by the configuration defined in the project. ***CAUTION: This operation cannot be undone, deleted dynamic files cannot be restored.*** ***CAUTION: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

When transferring a project, AGI Creator 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 panesl list.


Download to Target

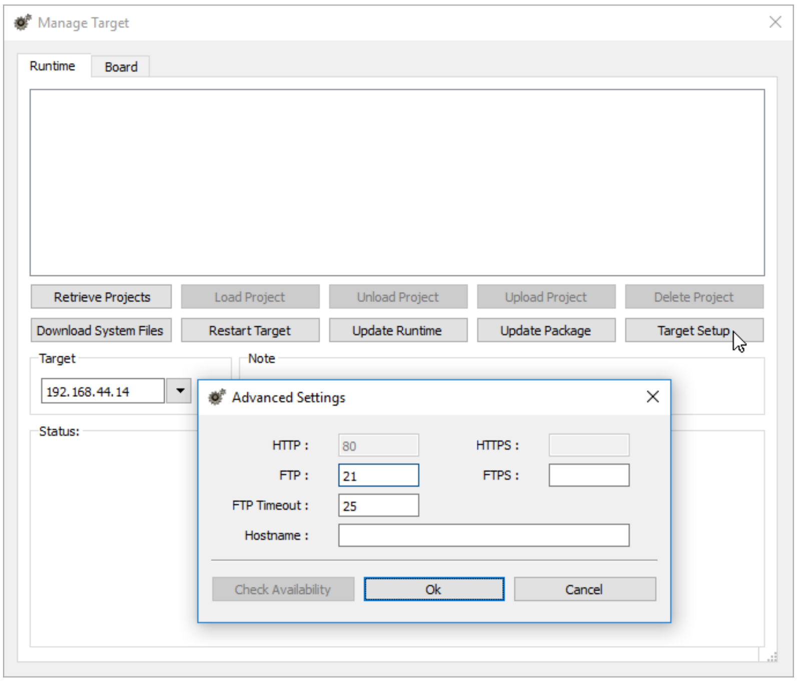
Ready to download

192.168.41.234:8585

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.



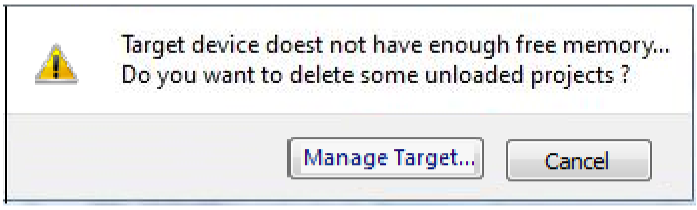
Target: 192.168.44.14

HTTP : 80
HTTPS :
FTP : 21
FTPS :
FTP Timeout : 25
Hostname :

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.


Target device doest not have enough free memory...
Do you want to delete some unloaded projects ?

To free more memory:

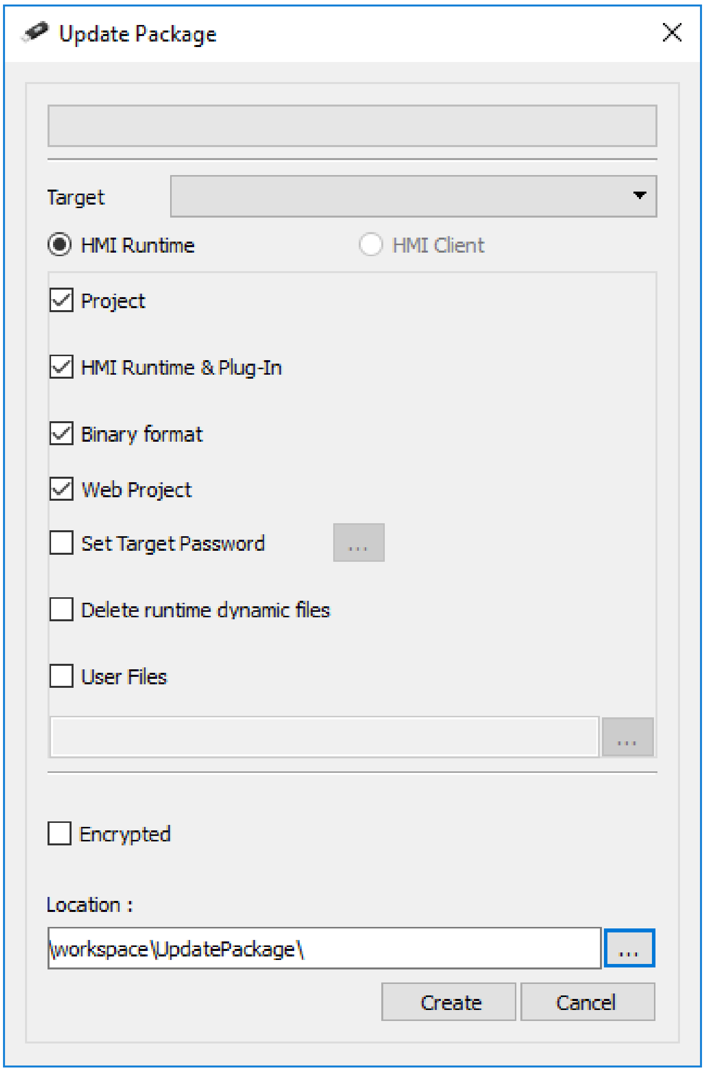
1. Click **Manage Target**.
2. Delete the projects you no longer need t to make more memory available.

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***



Update Package

Target

● HMI Runtime ○ HMI Client

☑ Project

☑ HMI Runtime & Plug-In

☑ Binary format

☑ Web Project

☐ Set Target Password

☐ Delete runtime dynamic files

☐ User Files

☐ Encrypted

Location :

\workspace\UpdatePackage\

Option	Description
Target	HMI device type. Selected automatically if the project is open.
Application Selector	Select the application to insert inside the UpdatePackage.zip • HMI Runtime • HMI Client (Available only on Linux devices)
Project	Adds open project to update package.

Option	Description
HMI Runtime & Plug-In	HMI Runtime is added to the update package. If the project is open the required plugins are also added to update package.
Binary Format	Download files using binary format.
Web Project	Download the AGI Web pages to HMI device.
Set Target Password	Sets password to perform critical tasks (for example, project download/upload , board management) See "Protecting access to HMI devices" on page 529.
Delete runtime dynamic files	When checked, all dynamic files will be deleted and the FRAM will be clean up.
User Files	Selects files to be copied to the QTHM folder of HMI device. Max size 5 MB
Encrypted	Enables encryption of update package so that it can only be unzipped by the HMI Runtime.
Location	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 2 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:

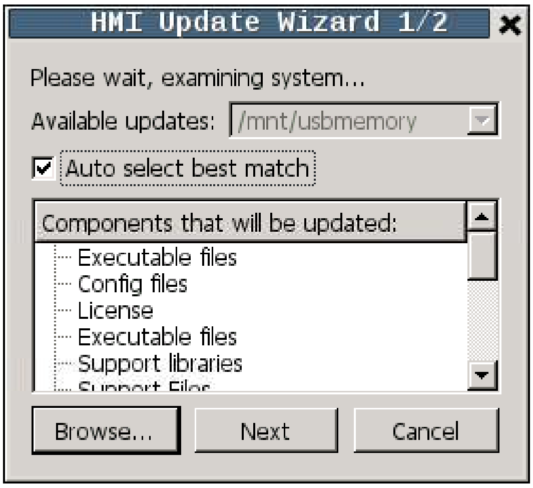
/mnt/data/hmi/qthmi

- *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.



HMI Update Wizard 1/2

Please wait, examining system...

Available updates: /mnt/usbmemory

☑ Auto select best match

Components that will be updated:
- Executable files
- Config files
- License
- Executable files
- Support libraries
- Support Files

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

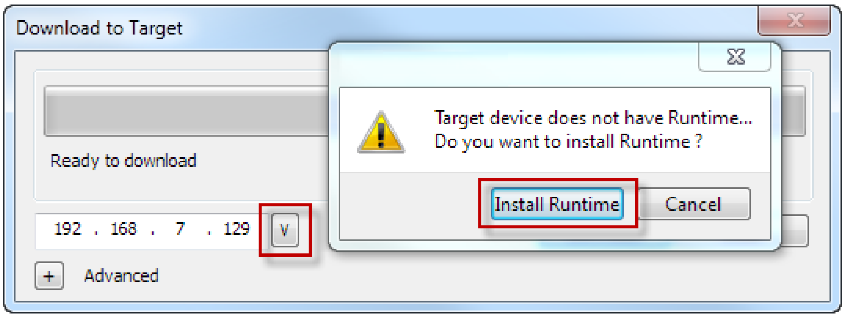
HMI devices are delivered from factory without Runtime.

When you power up the device for the first time, the Runtime Loader window is displayed (see "System Settings" on page 547 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 AGI Creator

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



Download to Target

Ready to download

192 . 168 . 7 . 129

Target device does not have Runtime...
Do you want to install Runtime ?

See "Transferring the project to HMI device" on page 97 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 101
2. Plug the USB drive in the device and follow the instructions for the type of device (see "System Settings" on page 547 for details)

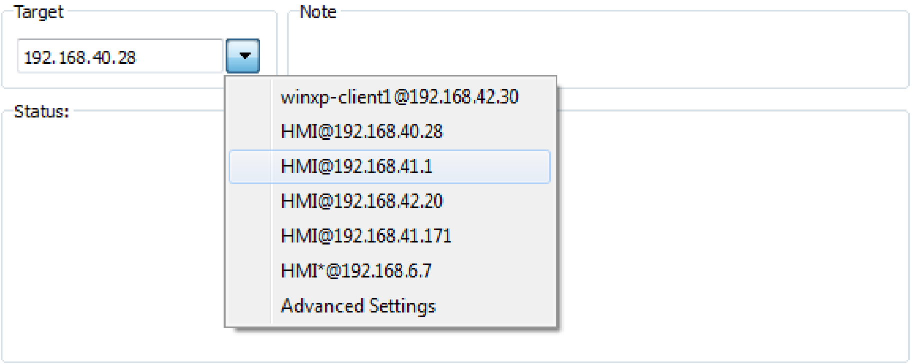
Note: Old versions of HMI devices may not support automatic installation of Runtime. Contact technical support for more information.

Upload projects

*Path: **Run**> **Manage Target***

You can copy a project from the Runtime to the computer where AGI Creator is running.

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



Target: 192.168.40.28

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 529 for details.

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

A copy of the project is saved in:

C:\Users\username\Documents\AGI Creator\workspace\Uploaded\RuntimeIPAddress\workspace\ProjectName

Note: If the upload operation fails, check firewall settings the computer where AGI Creator is running.

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.

Communication protocols

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

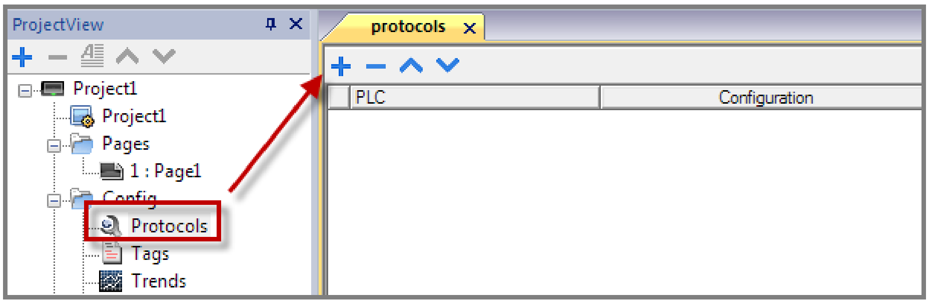
Device communication drivers are configured in the **Protocol Editor**.You can add up to the maximum number of protocols as specified in Table of functions and limits. Variable and System Variables are not counted as protocols.

See "Communication protocols" on page 585 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

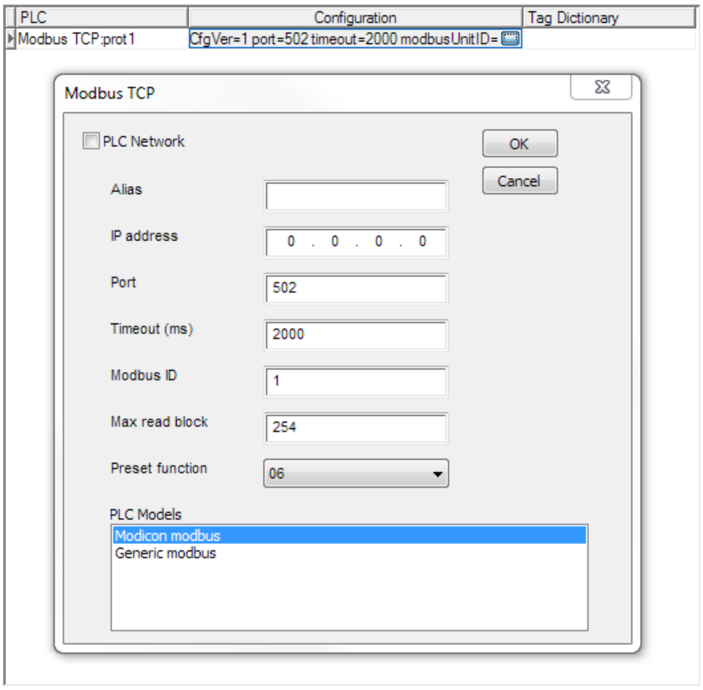
1. Click **+**.



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.



PLC	Configuration	Tag Dictionary
Modbus TCP:prot1	CfgVer=1 port=502 timeout=2000 modbusUnitID=	

Modbus TCP

□ PLC Network

Alias:

IP address: 0 . 0 . 0 . 0

Port: 502

Timeout (ms): 2000

Modbus ID: 1

Max read block: 254

Preset function: 06

PLC Models: Modicon modbus, Generic modbus

Protocol parameters

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

Parameter	Description
Dictionaries	Tags imported for the protocol. See "Importing tags" on page 111 for details.
Enable Offline AlgorithmOffline Retry Timeout	See "Automatic offline node detection" on page 299 for details.
Version	Protocol version available in AGI Creator for selected HMI device.

Adding tags

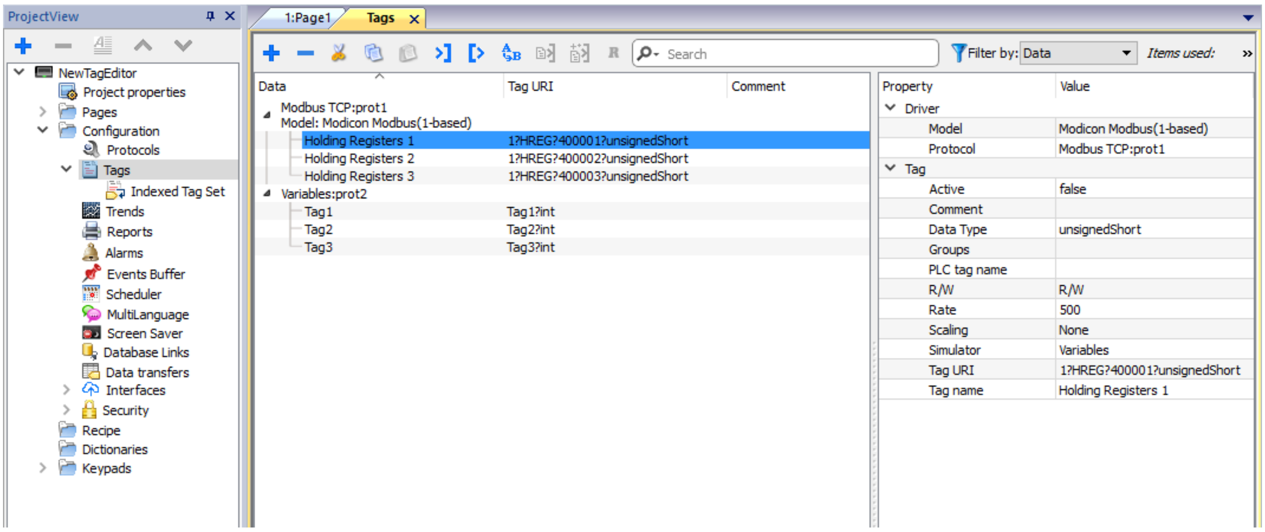
AGI Creator uses tag names to access all device data. All fields and reference locations in the device need to be assigned a tag name to be used in the HMI project.

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 45 for details.

Tag editor

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



Data	Tag URI	Comment
Modbus TCP:prot1 Model: Modicon Modbus(1-based)		
Holding Registers 1	1?HREG?400001?unsignedShort	
Holding Registers 2	1?HREG?400002?unsignedShort	
Holding Registers 3	1?HREG?400003?unsignedShort	
Variables:prot2		
Tag1	Tag1?int	
Tag2	Tag2?int	
Tag3	Tag3?int	

Property	Value
Driver	
Model	Modicon Modbus(1-based)
Protocol	Modbus TCP:prot1
Tag	
Active	false
Comment	
Data Type	unsignedShort
Groups	
PLC tag name	
R/W	R/W
Rate	500
Scaling	None
Simulator	Variables
Tag URI	1?HREG?400001?unsignedShort
Tag name	Holding Registers 1

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
Active	Update mode. • **false** = tags are read from controller only when required by the HMI device. • **true** = tags are continuously read even if not required by the displayed page. **Important: Leave this value set to false for higher communication performance.**
Description	Tag description
Encoding	Encoding type for string data type (UTF-8, Latin1, UTF-2 and UTF-16)
Groups	Group names associated to a tag
PLC tag name	Original PLC tag name, used to match tags used by HMI application (Tag Name) and tags exported from PLC
R/W	R/W tag attribute (R/W, R or W). 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.
Rate	Tag refresh time. Default: 500ms. 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(). **WARNING: Tags refresh rate is the maximum refresh rate. Actual refresh rate depends on: communication type (serial, fieldbus, Ethernet), protocol, amount of data exchanged.**
Scaling	Conversion applied to tag before database storage. • **By Formula** = defined as a linear transformation. • **By Range** = defined as a range conversion. • **Fixed Point** = fixed point scaling
Simulator	Tag behavior during simulation. Several profiles are available.
Tag address	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. **WARNING: Duplicate tag names are not allowed.**

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 585 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.

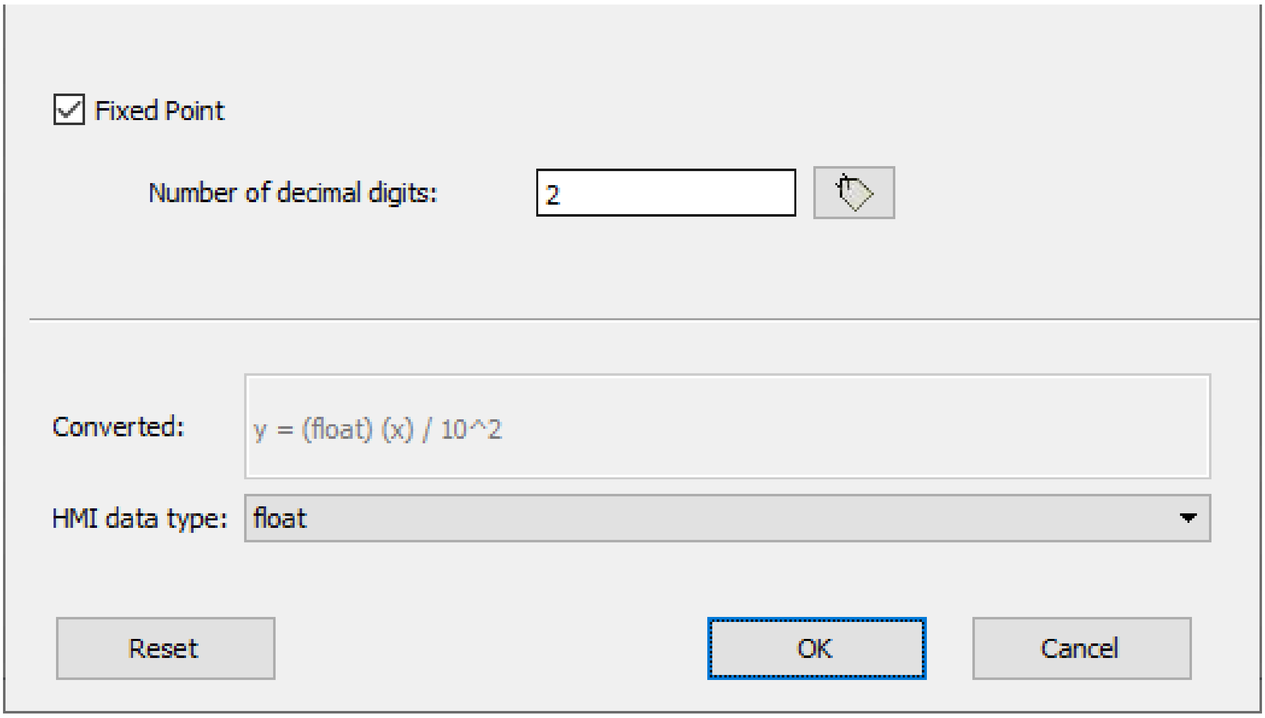
There is the possibility to configure

- 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

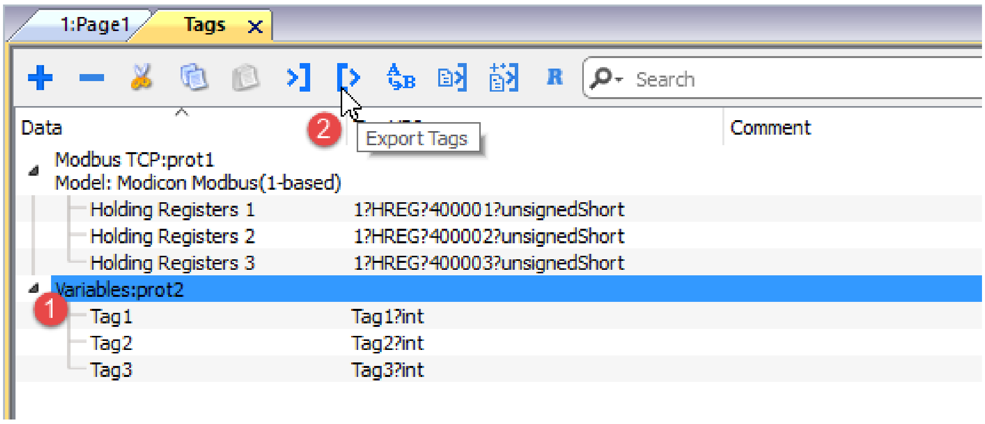
Number of decimal digits: 2

Converted: y = (float) (x) / 10^2

HMI data type: float

Exporting tags

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



Data		Comment
Modbus TCP:prot1 Model: Modicon Modbus(1-based)		
Holding Registers 1	1?HREG?400001?unsignedShort	
Holding Registers 2	1?HREG?400002?unsignedShort	
Holding Registers 3	1?HREG?400003?unsignedShort	
Variables:prot2		
Tag1	Tag1?int	
Tag2	Tag2?int	
Tag3	Tag3?int	

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 585).

Importing is a two step process:

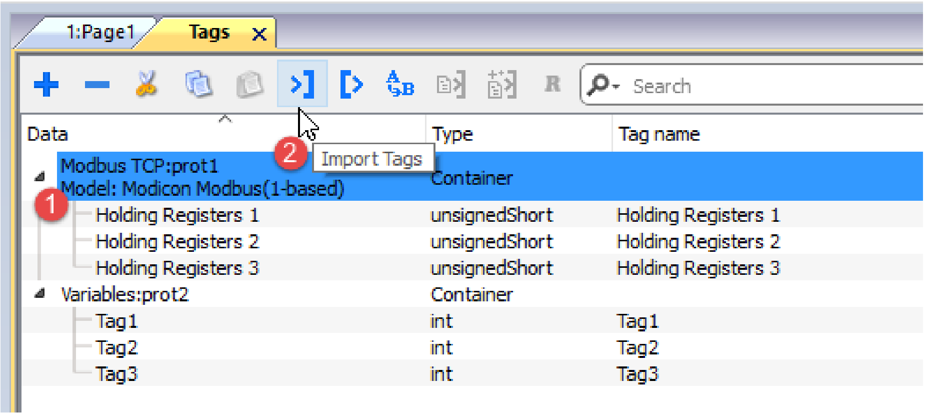
1. Import of the tag definition into a dictionary
2. Import tags from the dictionary to the project

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 307

Importing tags

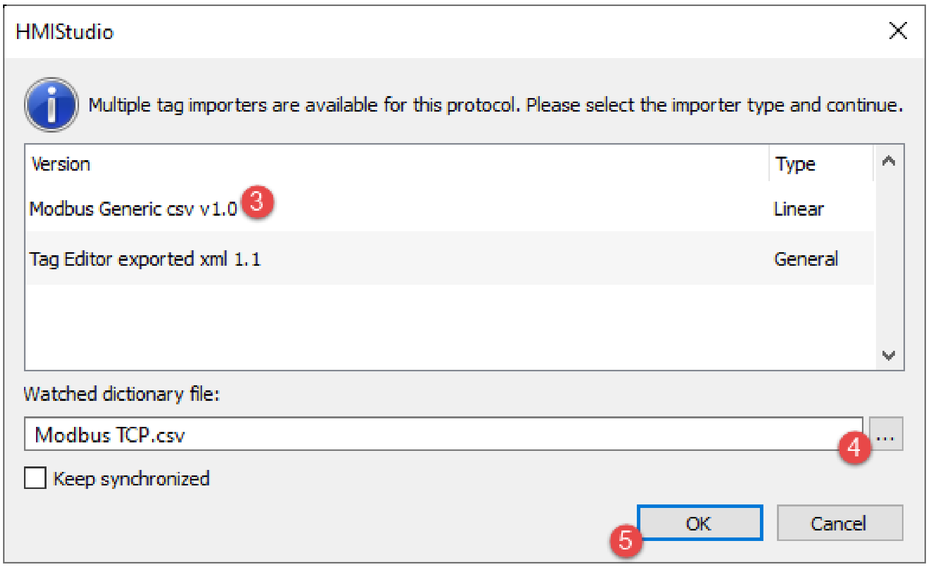
To import tags from an external file:

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



Data	Type	Tag name
Modbus TCP:prot1 Model: Modicon Modbus(1-based)	Container	
Holding Registers 1	unsignedShort	Holding Registers 1
Holding Registers 2	unsignedShort	Holding Registers 2
Holding Registers 3	unsignedShort	Holding Registers 3
Variables:prot2	Container	
Tag1	int	Tag1
Tag2	int	Tag2
Tag3	int	Tag3

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.



HMIStudio

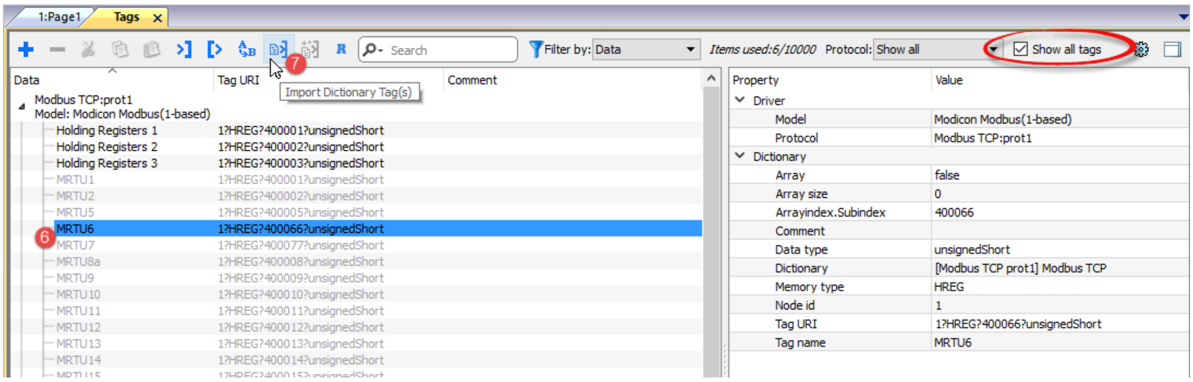
Multiple tag importers are available for this protocol. Please select the importer type and continue.

Version	Type
Modbus Generic csv v1.0	Linear
Tag Editor exported xml 1.1	General

Watched dictionary file: Modbus TCP.csv

□ Keep synchronized

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.



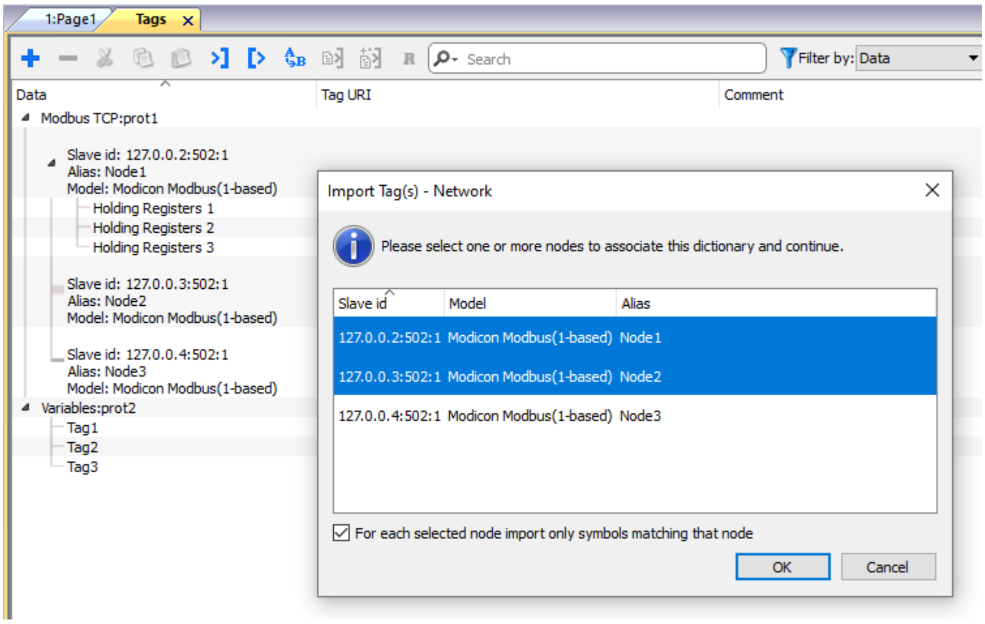
Filter by: Data   *Items used:6/10000*   Protocol: Show all   ☑ Show all tags

Data	Tag URI	Comment
Modbus TCP:prot1 Model: Modicon Modbus(1-based)		
Holding Registers 1	1?HREG?400001?unsignedShort	
Holding Registers 2	1?HREG?400002?unsignedShort	
Holding Registers 3	1?HREG?400003?unsignedShort	
MRTU1	1?HREG?400001?unsignedShort	
MRTU2	1?HREG?400002?unsignedShort	
MRTU5	1?HREG?400005?unsignedShort	
MRTU6	1?HREG?400066?unsignedShort	
MRTU7	1?HREG?400077?unsignedShort	
MRTU8a	1?HREG?400008?unsignedShort	
MRTU9	1?HREG?400009?unsignedShort	
MRTU10	1?HREG?400010?unsignedShort	
MRTU11	1?HREG?400011?unsignedShort	
MRTU12	1?HREG?400012?unsignedShort	
MRTU13	1?HREG?400013?unsignedShort	
MRTU14	1?HREG?400014?unsignedShort	
MRTU15	1?HREG?400015?unsignedShort	

Property	Value
Driver	
Model	Modicon Modbus(1-based)
Protocol	Modbus TCP:prot1
Dictionary	
Array	false
Array size	0
Arrayindex.Subindex	400066
Comment	
Data type	unsignedShort
Dictionary	[Modbus TCP prot1] Modbus TCP
Memory type	HREG
Node id	1
Tag URI	1?HREG?400066?unsignedShort
Tag name	MRTU6

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. 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.



Data	Tag URI	Comment
Modbus TCP:prot1		
Slave id: 127.0.0.2:502:1 Alias: Node1 Model: Modicon Modbus(1-based)		
Holding Registers 1		
Holding Registers 2		
Holding Registers 3		
Slave id: 127.0.0.3:502:1 Alias: Node2 Model: Modicon Modbus(1-based)		
Slave id: 127.0.0.4:502:1 Alias: Node3 Model: Modicon Modbus(1-based)		
Variables:prot2		
Tag1		
Tag2		
Tag3		

Import Tag(s) - Network

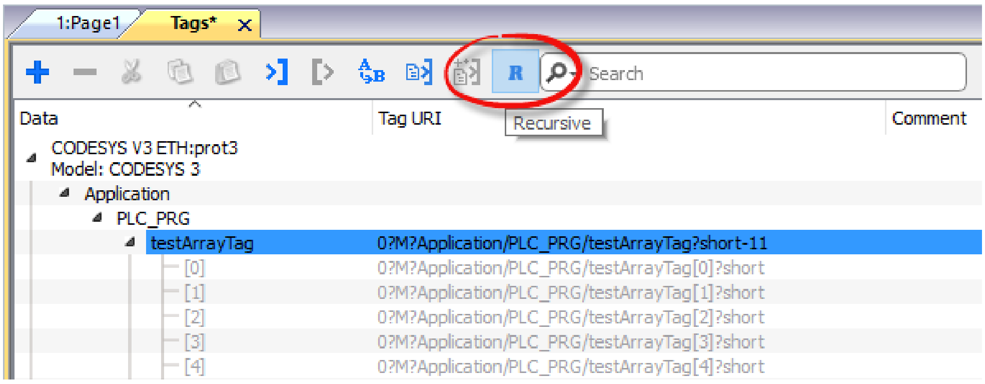
Please select one or more nodes to associate this dictionary and continue.

Slave id	Model	Alias
127.0.0.2:502:1	Modicon Modbus(1-based)	Node1
127.0.0.3:502:1	Modicon Modbus(1-based)	Node2
127.0.0.4:502:1	Modicon Modbus(1-based)	Node3

☑ For each selected node import only symbols matching that node

Recursive

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

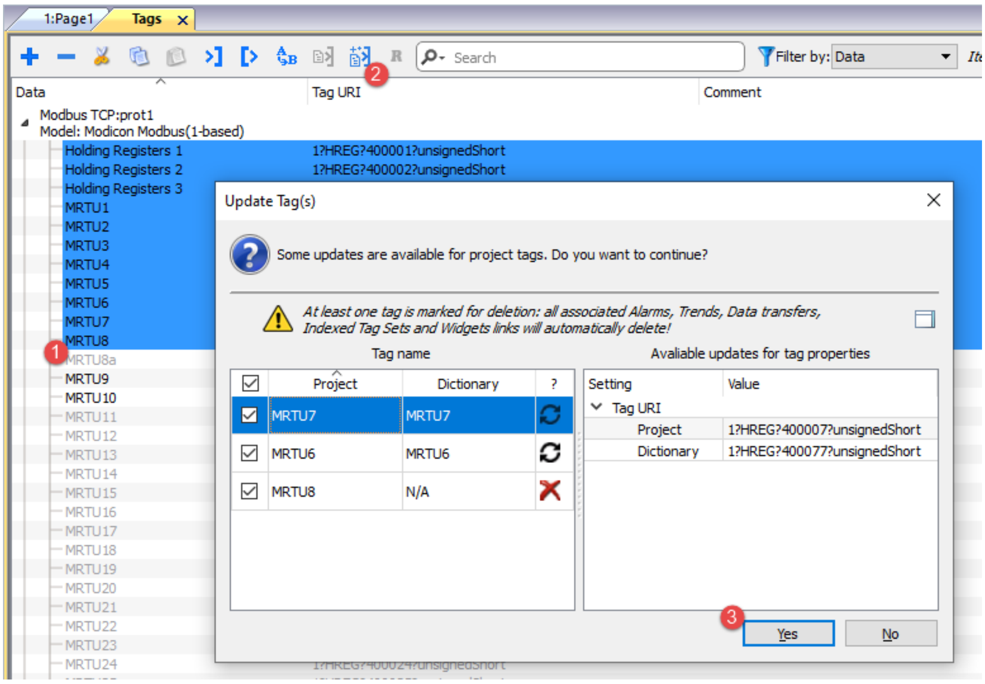


Data	Tag URI	Comment
CODESYS V3 ETH:prot3 Model: CODESYS 3		
Application		
PLC_PRG		
testArrayTag	0?M?Application/PLC_PRG/testArrayTag?short-11	
[0]	0?M?Application/PLC_PRG/testArrayTag[0]?short	
[1]	0?M?Application/PLC_PRG/testArrayTag[1]?short	
[2]	0?M?Application/PLC_PRG/testArrayTag[2]?short	
[3]	0?M?Application/PLC_PRG/testArrayTag[3]?short	
[4]	0?M?Application/PLC_PRG/testArrayTag[4]?short	

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



Data	Tag URI	Comment
Modbus TCP:prot1 Model: Modicon Modbus(1-based)		
Holding Registers 1	1?HREG?400001?unsignedShort	
Holding Registers 2	1?HREG?400002?unsignedShort	
Holding Registers 3		
MRTU1		
MRTU2		
MRTU3		
MRTU4		
MRTU5		
MRTU6		
MRTU7		
MRTU8		
MRTU8a		
MRTU9		
MRTU10		
MRTU11		
MRTU12		
MRTU13		
MRTU14		
MRTU15		
MRTU16		
MRTU17		
MRTU18		
MRTU19		
MRTU20		
MRTU21		
MRTU22		
MRTU23		
MRTU24	1?HREG?400024?unsignedShort	

Update Tag(s)

Some updates are available for project tags. Do you want to continue?

At least one tag is marked for deletion: all associated Alarms, Trends, Data transfers, Indexed Tag Sets and Widgets links will automatically delete!

Tag name

☑	Project	Dictionary	?
☑	MRTU7	MRTU7	⟳
☑	MRTU6	MRTU6	⟳
☑	MRTU8	N/A	✗

Avaliable updates for tag properties

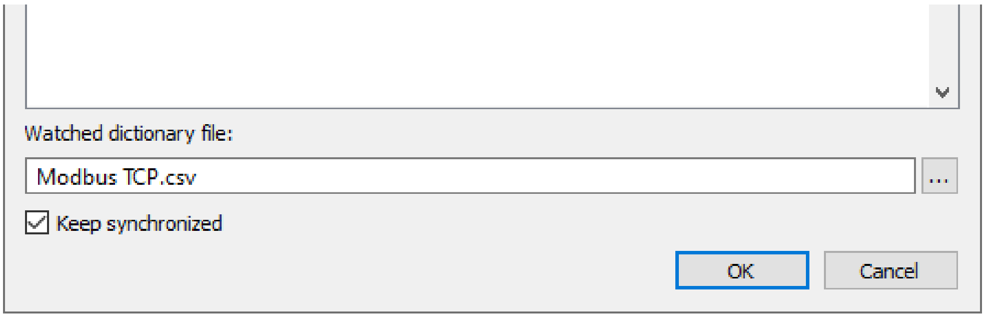
Setting	Value
Tag URI	
Project	1?HREG?400007?unsignedShort
Dictionary	1?HREG?400077?unsignedShort

Yes No

⟳	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 AGI Creator checks and update the tags from file dictionary automatically without user intervention.



Watched dictionary file:

Modbus TCP.csv

☑ Keep synchronized

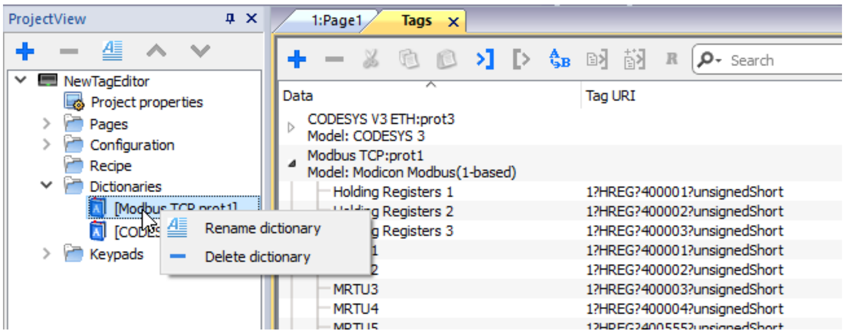
OK Cancel

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.

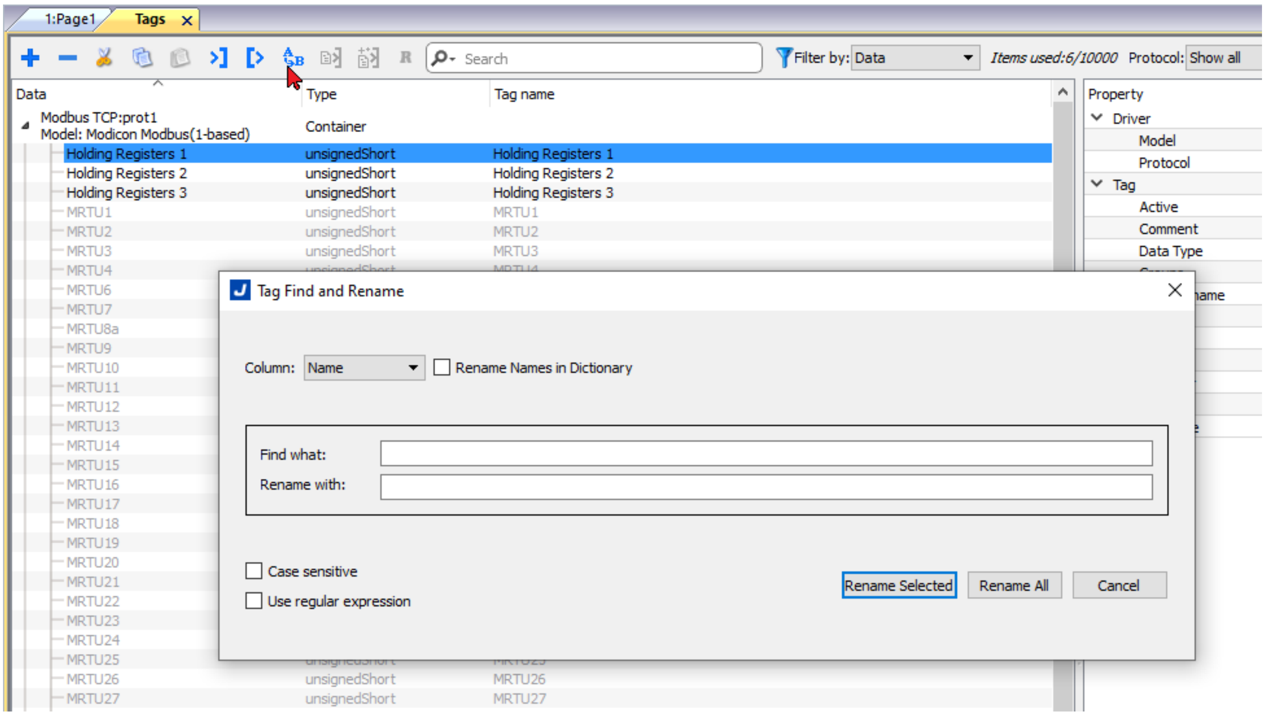


Data	Tag URI
CODESYS V3 ETH:prot3 Model: CODESYS 3	
Modbus TCP:prot1 Model: Modicon Modbus(1-based)	
Holding Registers 1	1?HREG?400001?unsignedShort
Holding Registers 2	1?HREG?400002?unsignedShort
Holding Registers 3	1?HREG?400003?unsignedShort
1	1?HREG?400001?unsignedShort
2	1?HREG?400002?unsignedShort
MRTU3	1?HREG?400003?unsignedShort
MRTU4	1?HREG?400004?unsignedShort

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



Data	Type	Tag name
Modbus TCP:prot1 Model: Modicon Modbus(1-based)	Container	
Holding Registers 1	unsignedShort	Holding Registers 1
Holding Registers 2	unsignedShort	Holding Registers 2
Holding Registers 3	unsignedShort	Holding Registers 3
MRTU1	unsignedShort	MRTU1
MRTU2	unsignedShort	MRTU2
MRTU3	unsignedShort	MRTU3
MRTU26	unsignedShort	MRTU26
MRTU27	unsignedShort	MRTU27

Tag Find and Rename

Column: Name ☐ Rename Names in Dictionary

Find what:

Rename with:

☐ Case sensitive

☐ Use regular expression

Rename Selected | Rename All | Cancel

Parameters	
Column	Select the column to modify with the find and rename operation
Rename Names in Dictionary	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 "Name"

Parameters	
Find what	String to search
Rename with	String to replace
Case sensitive	Takes account of upper and lower case letters
Use regular expression	Enable regular expression in search/replace pattern See https://en.wikipedia.org/wiki/Regular_expression 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

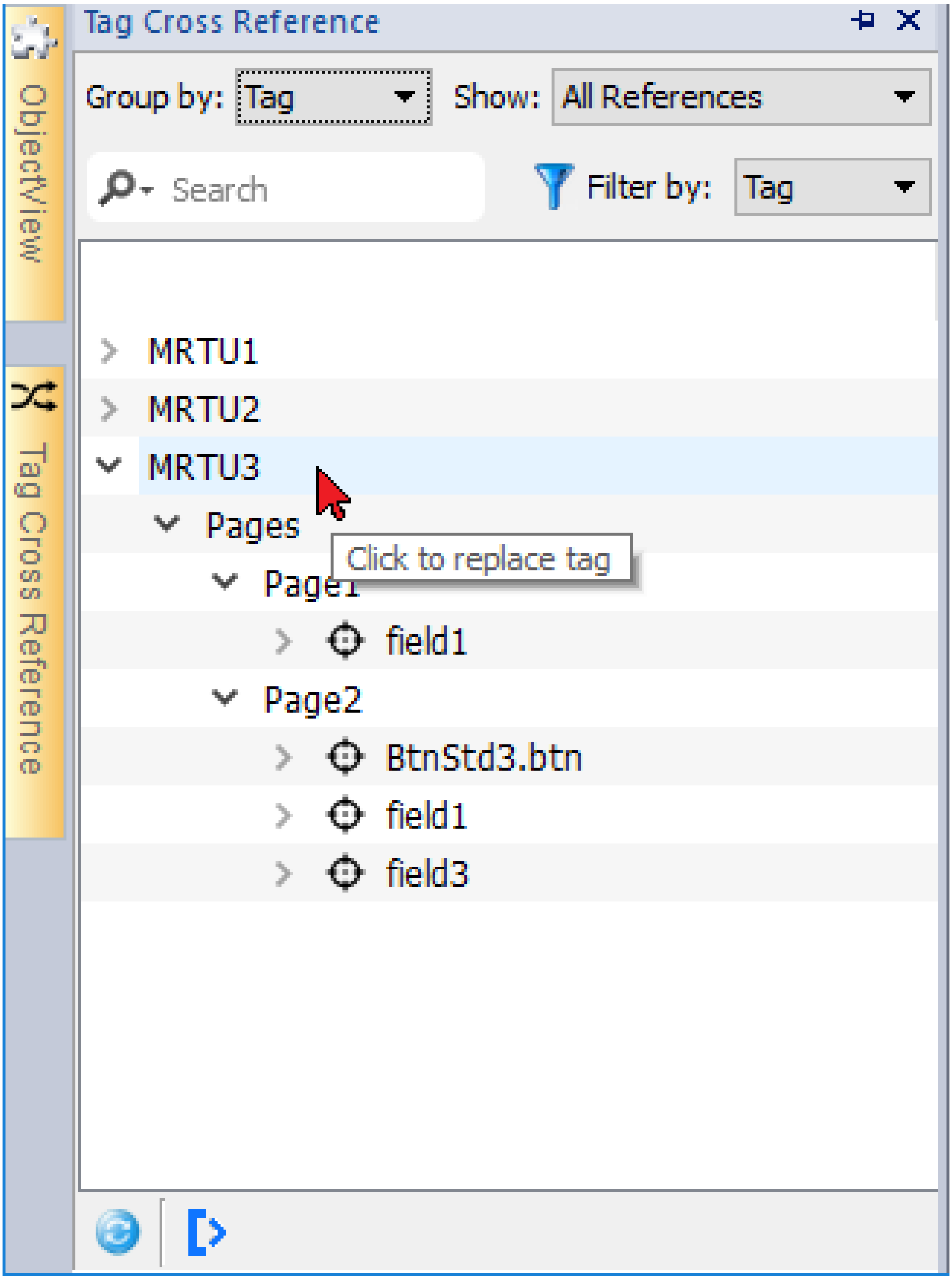
(.*)	is meaning any sequence of characters
\1	is a copy of the first sequence of characters enclosed by (...) found inside the search string

Tag find and replace

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

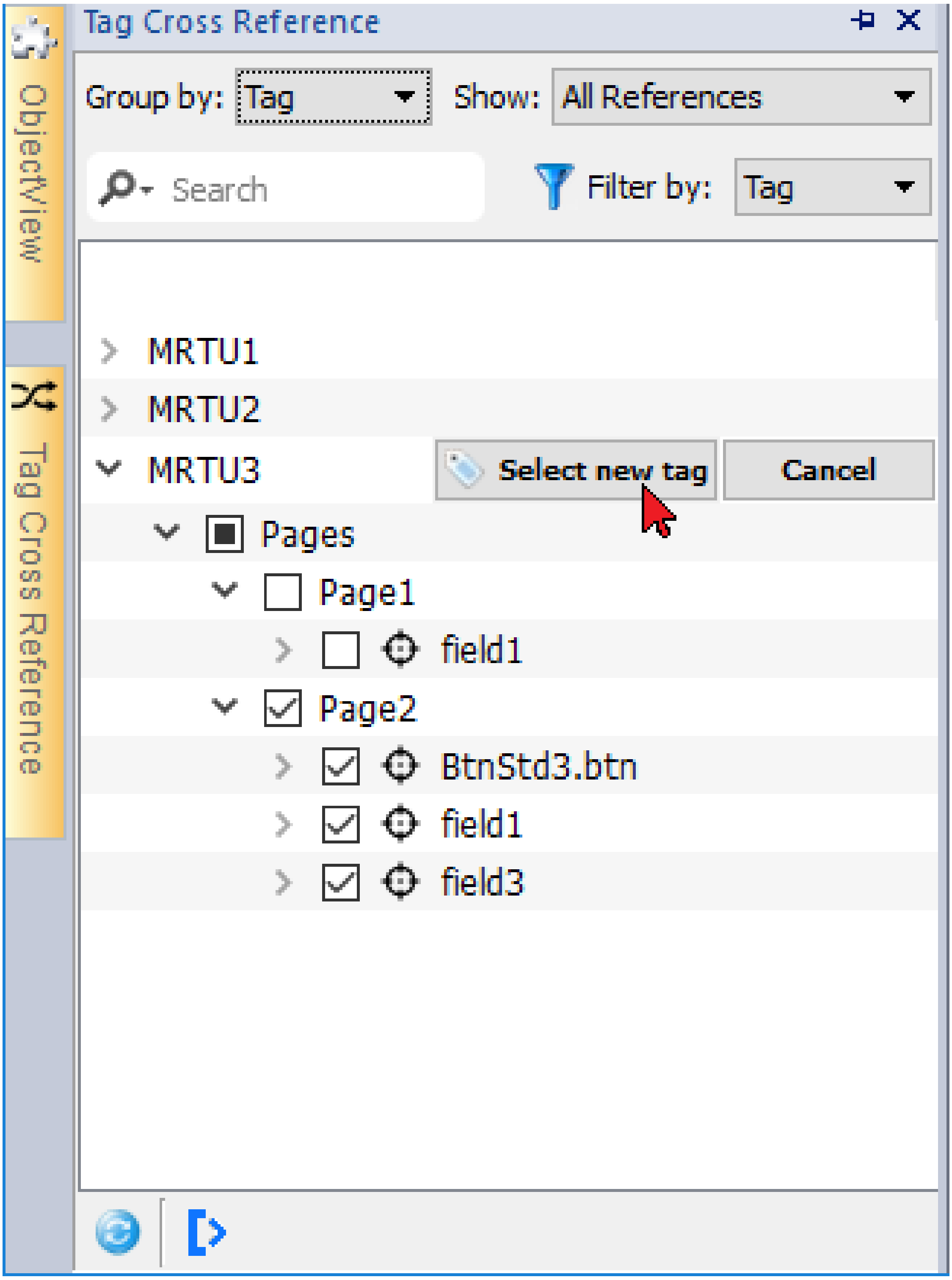
Tag replace is only applicable for Protocol tags which are shown in black color and not for System Variable, Alias and Recipe tags. (See "Opening the Tag Cross Reference pane" on page 128 for the different colors meaning)

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.



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.

Creating an indexed addressing set

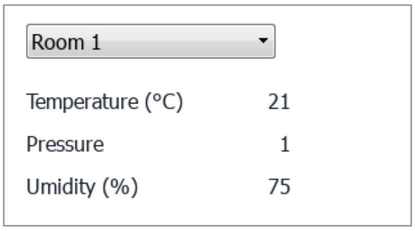
Scenario

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

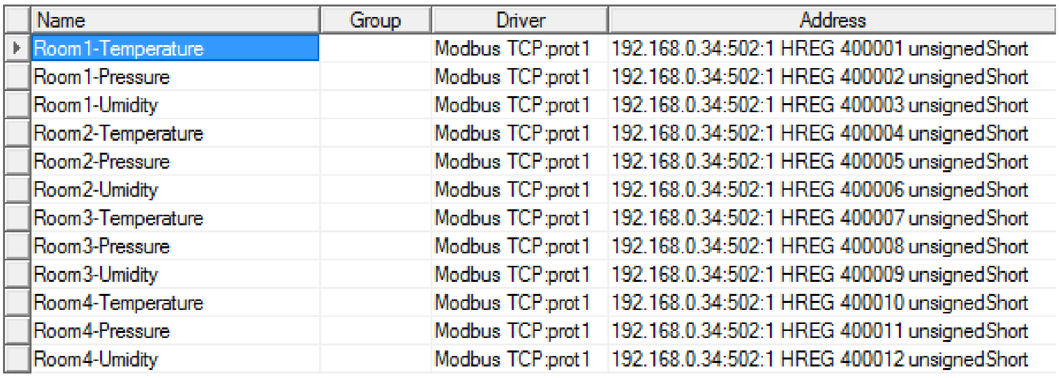
Temperature (°C)	21
Pressure	1
Umidity (%)	75

How to create an indexed tag set

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

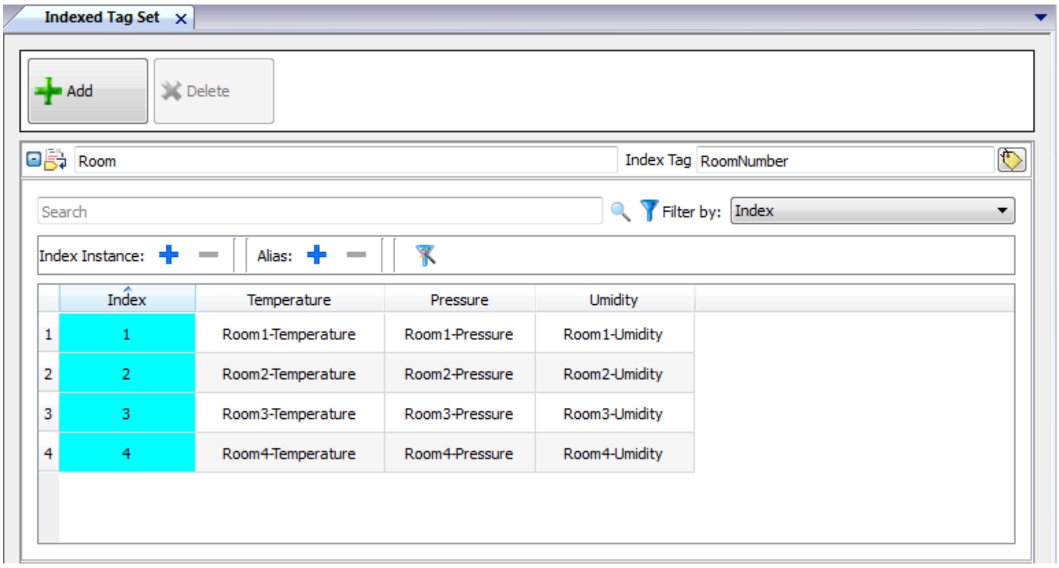
To do this you need to create an indexed tag set.

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.



Name	Group	Driver	Address
Room1-Temperature		Modbus TCP:prot1	192.168.0.34:502:1 HREG 400001 unsignedShort
Room1-Pressure		Modbus TCP:prot1	192.168.0.34:502:1 HREG 400002 unsignedShort
Room1-Umidity		Modbus TCP:prot1	192.168.0.34:502:1 HREG 400003 unsignedShort
Room2-Temperature		Modbus TCP:prot1	192.168.0.34:502:1 HREG 400004 unsignedShort
Room2-Pressure		Modbus TCP:prot1	192.168.0.34:502:1 HREG 400005 unsignedShort
Room2-Umidity		Modbus TCP:prot1	192.168.0.34:502:1 HREG 400006 unsignedShort
Room3-Temperature		Modbus TCP:prot1	192.168.0.34:502:1 HREG 400007 unsignedShort
Room3-Pressure		Modbus TCP:prot1	192.168.0.34:502:1 HREG 400008 unsignedShort
Room3-Umidity		Modbus TCP:prot1	192.168.0.34:502:1 HREG 400009 unsignedShort
Room4-Temperature		Modbus TCP:prot1	192.168.0.34:502:1 HREG 400010 unsignedShort
Room4-Pressure		Modbus TCP:prot1	192.168.0.34:502:1 HREG 400011 unsignedShort
Room4-Umidity		Modbus TCP:prot1	192.168.0.34:502:1 HREG 400012 unsignedShort

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.



Room Index Tag RoomNumber

Filter by: Index

	Index	Temperature	Pressure	Umidity
1	1	Room1-Temperature	Room1-Pressure	Room1-Umidity
2	2	Room2-Temperature	Room2-Pressure	Room2-Umidity
3	3	Room3-Temperature	Room3-Pressure	Room3-Umidity
4	4	Room4-Temperature	Room4-Pressure	Room4-Umidity

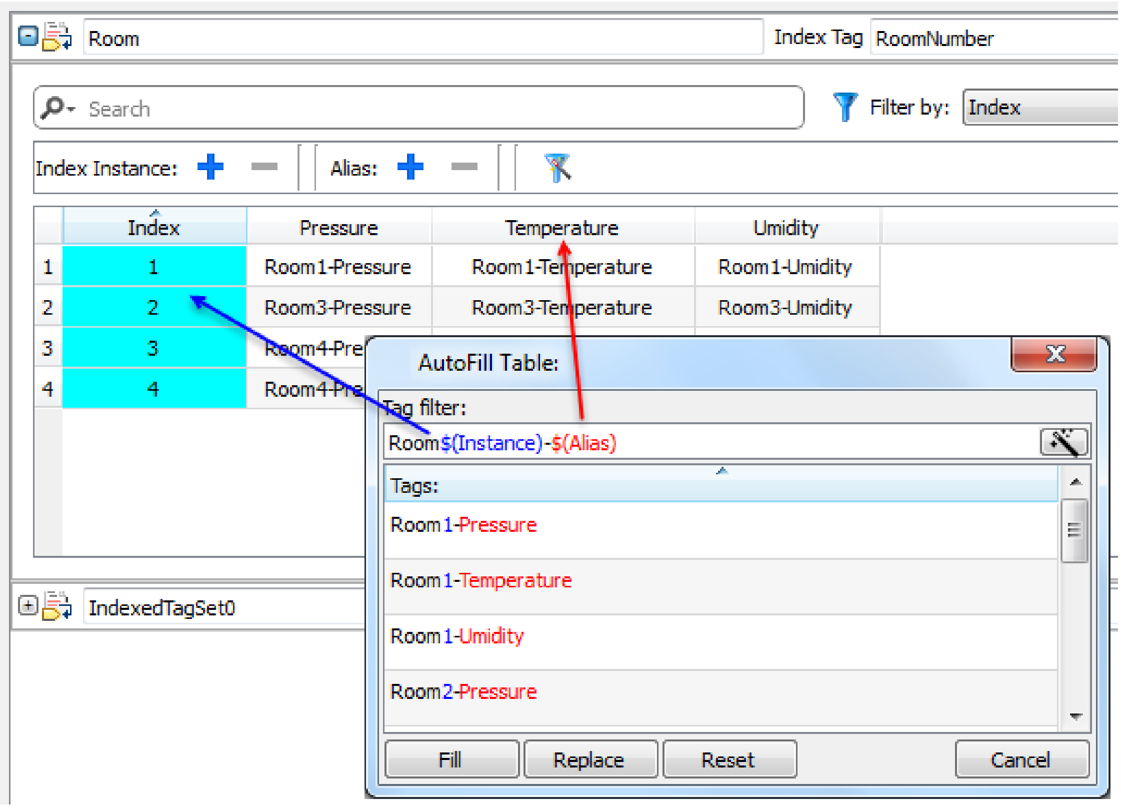
Note: The Index Tag datatype can be a number, a string or any type of simple data types.

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

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.



Room
Index Tag RoomNumber
Filter by: Index

	Index	Pressure	Temperature	Umidity
1	1	Room1-Pressure	Room1-Temperature	Room1-Umidity
2	2	Room3-Pressure	Room3-Temperature	Room3-Umidity
3	3	Room4-Pre		
4	4	Room4-Pre		

IndexedTagSet0

Tag filter:
Room$(Instance)-$(Alias)

Tags:
Room1-Pressure
Room1-Temperature
Room1-Umidity
Room2-Pressure

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 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,

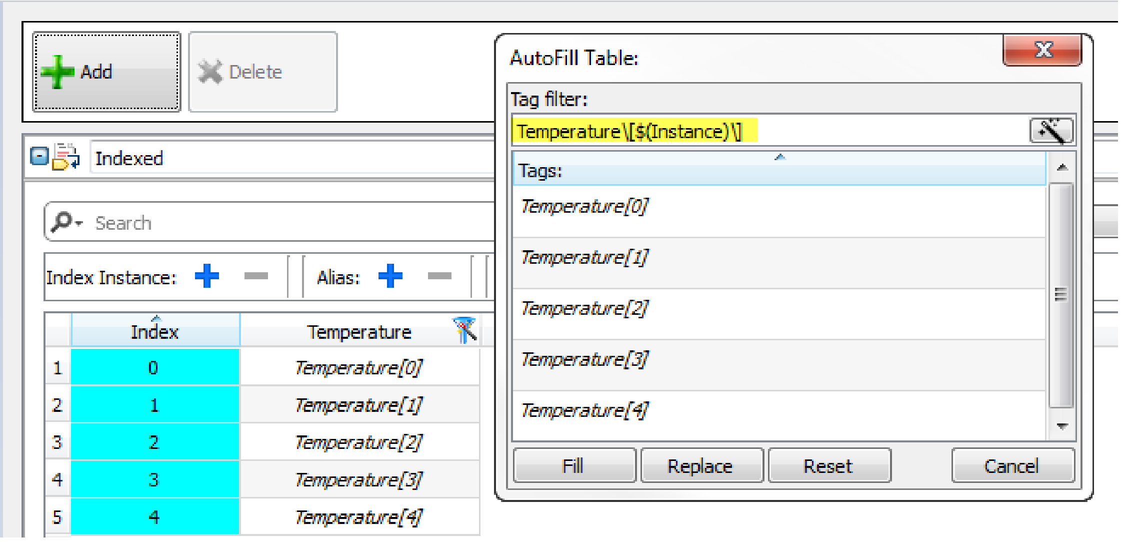
…

Autofill table elements

Element	Description
Fill	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.
Replace	Replace all table entries with those provided by the Autofill table.
Reset	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).



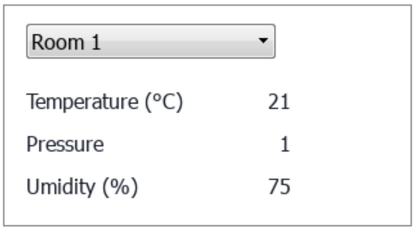
Tag filter: Temperature\[$(Instance)\]

	Index	Temperature
1	0	*Temperature[0]*
2	1	*Temperature[1]*
3	2	*Temperature[2]*
4	3	*Temperature[3]*
5	4	*Temperature[4]*

Tags:
Temperature[0]
Temperature[1]
Temperature[2]
Temperature[3]
Temperature[4]

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.



Room 1

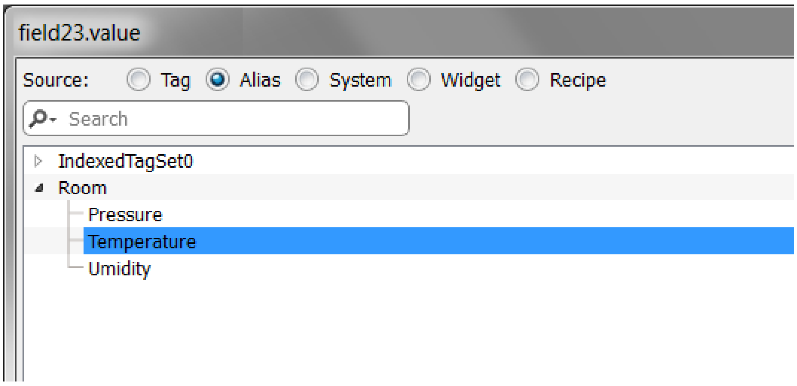
Temperature (°C)	21
Pressure	1
Umidity (%)	75

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**).



field23.value

Source: ○ Tag ● Alias ○ System ○ Widget ○ Recipe

Search

- IndexedTagSet0
- Room
  - Pressure
  - Temperature
  - Umidity

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:

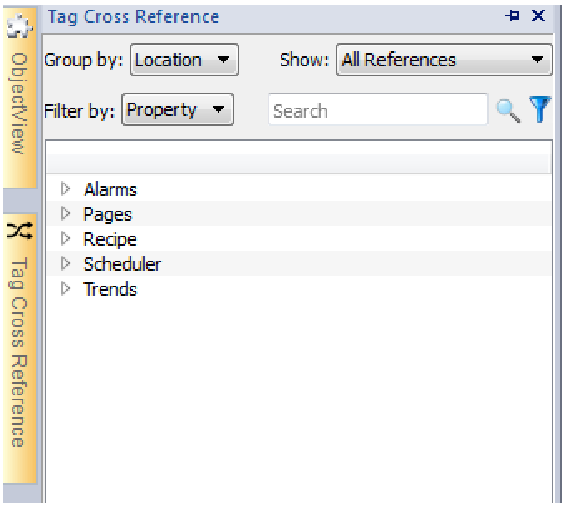
- verify where each tag is used (alarms, pages, recipes, schedulers, trends, and so on)
- identify invalid tag references (references to tags not defined in the tag editor)
- identify tags not used in the project

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

Opening the Tag Cross Reference pane

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

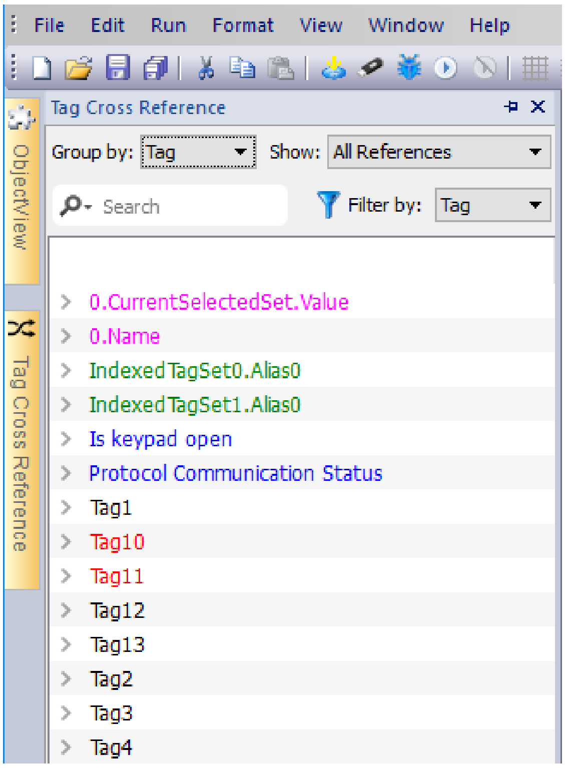
Click the **Tag Cross Reference** tab to open the Tag Cross Reference pane.



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

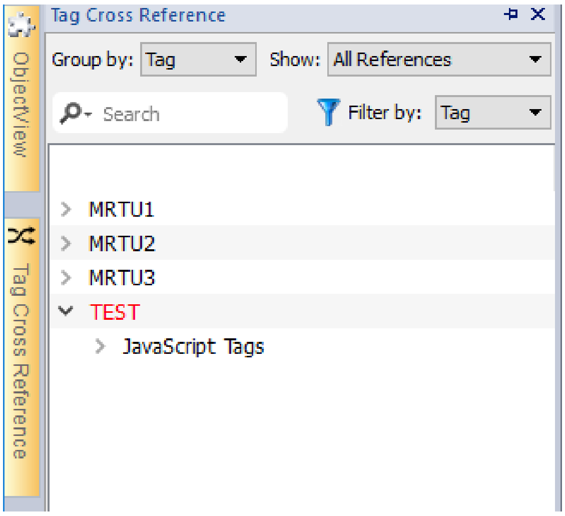
The Tag Cross Reference pane provides a set of standard functions.

Element	Function
Group by	Groups tags by **Location** (alarms, pages, trends and so on) or **Tag** name
Show	Filters tags and displays: • **All Reference**: all tags • **Invalid Tag Reference**: tags not listed in the Tag Editor. • **Unused Tags**: tags listed in the Tag Editor but not used in project.
Search field	Applies a filter to display a limited number of tags
Filter by	Filters tags by **Location**, **Tag** or **Property**.

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

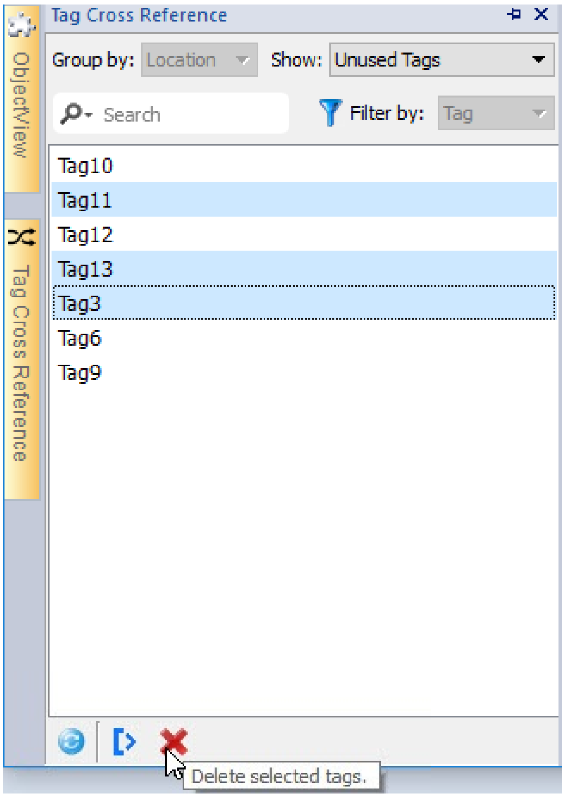
Double-click on a tag to open the editor or page where it is used.

Invalid tag references will be listed in red color:



Delete unused tags

From the unused tags view, is possible 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

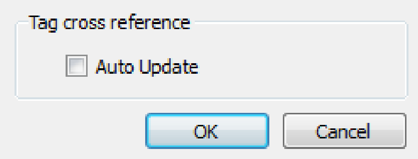
Manual update

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 AGI Creator **Properties** page.



Tag cross reference

□ Auto Update

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

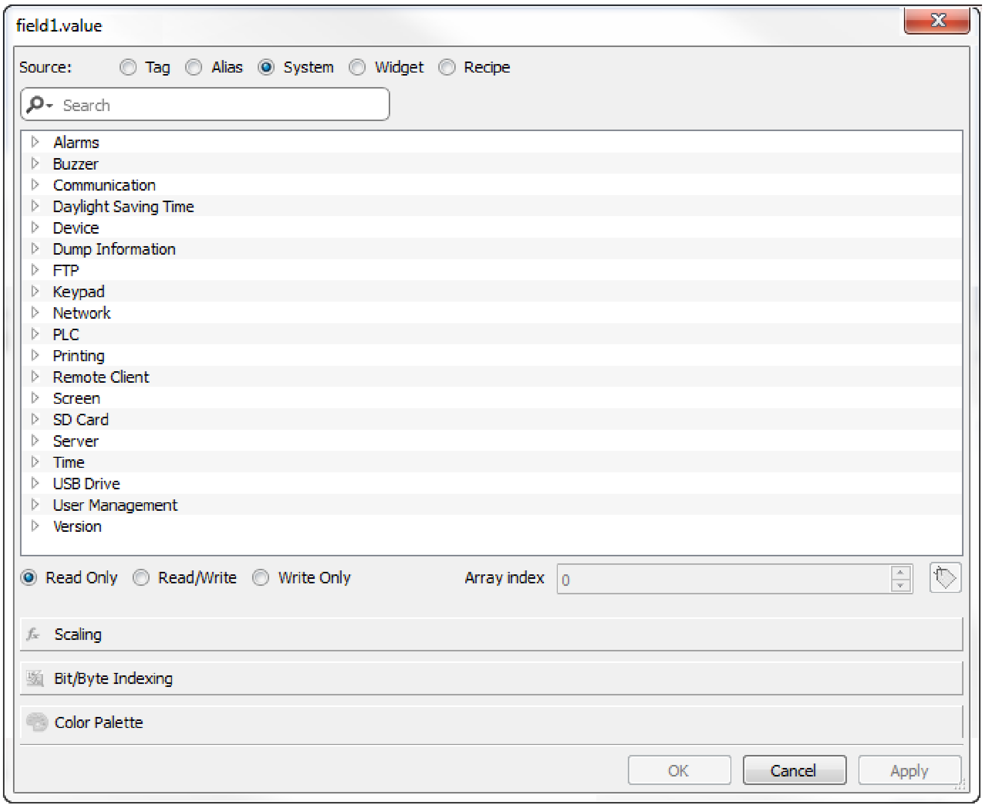
Note: The separators used in export operation depends on regional settings of your computer.

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.



field1.value

Source: ○ Tag ○ Alias ● System ○ Widget ○ Recipe

Search

- Alarms
- Buzzer
- Communication
- Daylight Saving Time
- Device
- Dump Information
- FTP
- Keypad
- Network
- PLC
- Printing
- Remote Client
- Screen
- SD Card
- Server
- Time
- USB Drive
- User Management
- Version

● Read Only ○ Read/Write ○ Write Only Array index 0

Alarms variables

Number of alarms of the requested type.

Variable	Description	Data type
Alarm not acknowledged	True when alarms unacknowledged is pending (Not Triggered Not Acknowledged<>0) OR (Triggered Not Acknowledged<>0)	boolean read only
Alarm triggered	True when at least one alarm is triggered (Triggered Acknowledged<>0) OR (Triggered <>0) OR (Triggered Not Acknowledged<>0)	boolean read only
Number of missed alarm events	Alarms exceeding the event queue. Queue length is defined in the *engineconfig.xml* file.	int read only
Number of not triggered acknowledged	Alarm condition no longer active; alarms already acknowledged	int read only
Number of not triggered not acknowledged	Alarma condition no longer active; awaiting acknowledgment	int read only
Number of triggered acknowledged	Alarm condition active; alarms already acknowledged	int read only
Number of triggered alarms	Alarm active: aknowledgement not required	int read only
Number of triggered not acknowledged	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
Buzzer Setup	**0** = disabled **1** = enabled (buzzer sounds as audible on any touchscreen event) **2** = buzzer status controlled by **Buzzer Control** system variable or by **Buzzer on Touch** property inside the "Project properties" on page 79 ⚠ **Buzzer on touchscreen (Setup=1) is not available on Linux platforms. See "*Buzzer on Touch*" property in alternative.**	int
Buzzer Control	**0** = buzzer off **1** = buzzer on **2** = buzzer blink	int
Buzzer Off Time	Duration in milliseconds of off time when blink has been selected. Default = 1000. Range: 100–5000.	int
Buzzer On Time	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
Protocol Communication Status	Summarize the status of the communication protocols. **0** = No protocol running, protocol drivers might not have been properly downloaded to the HMI device. **1** = Protocols loaded and started, no communication error. **2** = At least one communication protocol is reporting an error.	int Read only
Protocol Error Message	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.	ASCII string Read only
Protocol Error Count	Number of communication errors occurred since last reset. Reset value with Reset Protocol Error Count action, see "System actions" on page 199.	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
Standard Offset	Offset in minutes when standard time is set, with respect to GMT (for example: -8x60 = -480 minutes).
Standard Week	Week in which the standard time starts (for example: First = 1).
Standard Month	Month in which the standard time starts. Range: 0–11. (for example: November = 10).
Standard Day	Day of week in which the standard time starts (for example: Sunday = 0).
Standard Hour	Hour in which the standard time starts (for example: 02 = 2).
Standard Minute	Minute in which the standard time starts (for example: 00 = 0).
DST Offset	Offset in minutes when DLS time is set, with respect to GMT
DST Week	Week in which the DLS time starts
DST Month	Month in which the DLS time starts. Range: 0–11.
DST Day	Day of week in which the DLS time starts
DST Hour	Hour in which the DLS time starts
DST Minute	Minute in which the DLS time starts

Device variables

Device settings and operating status information.

Variable	Description	Data type
Available System Memory	Free available RAM memory in bytes.	uint64 read only
Backlight Time	Activation time in hours of the display backlight since production of the device.	unsignedInt read only
Battery LED	Enables/disables the low battery LED indicator (when available). **0** = disabled **1** = enabled Not available on Linux platforms (find the platform of your device at "HMI devices capabilities" on page 543)	int
Battery	Reserved	int

Variable	Description	Data type
Timeout	ℹ Not available on Linux platforms (find the platform of your device at "HMI devices capabilities" on page 543)	
Display Brightness	Returns and adjusts brightness level. Even when set to 0, the backlight is still on and the **Backlight Time** counter increases. Range: 0–255 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.	int
External Timeout	Non-operational time after which the display backlight is automatically turned off. The backlight is automatically turned on when the user touches the screen. **-1** = Switch off backlight and disable touch (switch display off). **Backlight Time** counter is stopped. ℹ On Linux devices requires BSP v1.0.324 or higher. **-2** = 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. ℹ Available only on Linux devices. Requires BSP v1.0.324 or higher. **0** = Switch backlight on (switch display on) **1..n** = Timeout, in seconds, for switch off backlight (screen saver timer) ℹ The timeout value is rounded to multiples of one minute (60, 120, 180, etc,) in all Linux devices. Find the platform of your device at "HMI devices capabilities" on page 543	int
Flash Free Space	Free space left in internal Flash memory.	uint64 read only
Manufacturer Code	Internal code that identify the HMI type	unsignedInt read only
System Font List	List of system fonts	string read only
System Mode	Runtime operation status.	int

Variable	Description	Data type
	1 = booting **2** = configuration mode **3** = operating mode **4** = restart **5** = shutdown	
System UpTime	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
Dump Error Message	Return error message if any error occurs during the dump operation	string read only
Dump Archive Status	**0** = initial default state **1** = operation triggered **2** = operation complete successfully **3** = operation completed with errors	int read only
Dump Recipe Status	**0** = initial default state **1** = operation triggered **2** = operation complete successfully **3** = operation completed with errors	int read only
Dump Trend Status	**0** = initial default state **1** = operation triggered **2** = operation complete successfully **3** = operation completed with errors	int read only
Reset Recipe Status	**0** = initial default state **1** = operation triggered **2** = operation complete successfully **3** = operation completed with errors	int read only
Restore Recipe Status	**0** = initial default state **1** = operation triggered **2** = operation complete successfully **3** = 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
FTP Current Command	Last FTP command	string read only
FTP Error Message	Last FTP error message	string read only
FTP Progress	Download/upload progress (0/100%)	short read only
FTP Status	Status of last FTP command: • 0 = idle • 1 = active • 2 = done • 3 = error	short read only

Keypad variables

Keypad status.

Variable	Description	Data type
Is keypad open	**0** = no keypad open **1** = keypad open	int read only

Network variables

Device network parameters.

Variable	Description	Data type
Adapters Parameters	This is a JSON string that can be use to read or update the network adapters parameters	string
Gateway	Gateway address of the main Ethernet interface of device	string read only
IP Address	IP address of the main Ethernet interface of device	string read only
Mac ID	MAC ID of the main Ethernet interface of device	string

Variable	Description	Data type
		read only
Status	Contains the result of the last operation required by writing inside the Adapter Parameters. It is updated after each write operation. • Empty string is meaning no errors • Last error descriptions	string read only
Subnet Mask	Subnet Mask of the main Ethernet interface of device	string read only

Printing variables

Information on printing functions.

Variable	Description	Data type
Completion percentage	Percentage of completion of current print job. Range: 0–100	read only
Current disk usage	Folder size in bytes where PDF reports are stored. If *Flash* has been selected as *Spool media type*, this value corresponds to *reportspool*.	read only
Current job	Name of the report the job is processing. Current job is the following: • [report name] for a **Graphic Report** • [first line of text] for a **Text Report**	read only
Current RAM usage	Size in bytes of the RAM used to process the current job	read only
Disk quota	Maximum size in bytes of the folder where PDF reports are stored	read only
Graphic job queue size	Number of available graphic jobs in the printing queue	read only
Last error message	Description of the last returned error	string read only
RAM quota	Maximum size in bytes of the RAM used to generate reports	read only
Status	Printing system status. Values: • **idle**	string read only

Variable	Description	Data type
	- **error** - **paused** - **printing**	
Text job queue size	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
Connection status	**0** = client can not reach the server client. The connection with server is lost. **1** = client can reach the server. The connection with server is active. ℹ This is only a client side variable. On HMI Runtime or AGI PC Runtime it will be always 0.	int (32 bit) read only

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

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

Version variables

Operating System and runtime version.

Variable	Description	Data type
Main OS Version	Version of Main OS.	string
Runtime Version	Version of runtime.	string

Screen variables

Screen status.

Variable	Description
Time remaining to unlock	Time remaining to unlock screen (see **LockScreen** action, "Page actions" on page 185)
X Screen resolution	Display horizontal screen size in pixel
Y Screen resolution	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 AGI 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 **System Date** format (milliseconds).	uint64
Project load time	Date when the project was loaded on the HMI Runtime as in **System Date** format (milliseconds).	uint64

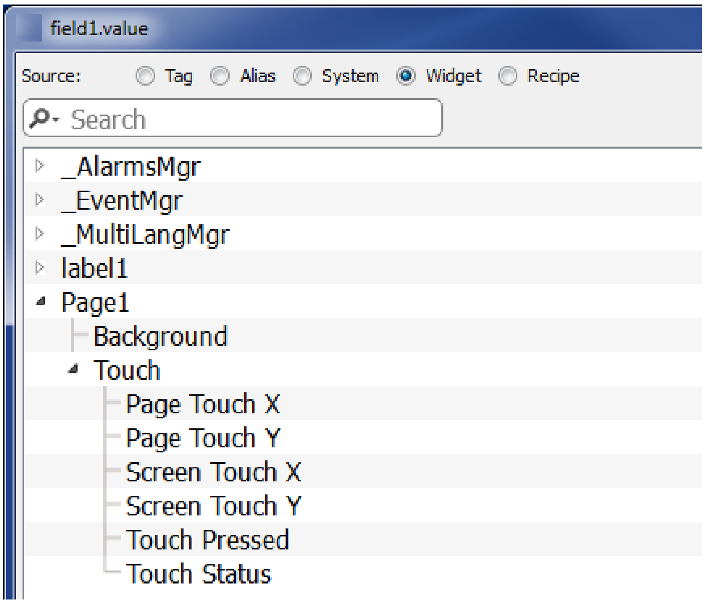
Time variables

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
Second	Range: 0–59	int
System Time	The same as UTC time. It can also be set as date/time for this variable.	unsignedInt
Year	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.


field1.value

Source: ○ Tag ○ Alias ○ System ● Widget ○ Recipe

Search

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

Variable	Description	Java Script
Page Touch X **Page Touch Y**	Cursor position related to page	page.primaryTouch.x page.primaryTouch.y
Screen Touch X **Screen Touch Y**	Cursor position related touchscreen	page.primaryTouch.screenX page.primaryTouch.screenY
Touch Press	**0** = screen not pressed **1** = screen pressed	page.primaryTouch.pressed
Touch Status	Generic touch screen changes. This variable contains the concatenation of **Screen Touch X**, **Screen Touch Y** and **Touch Press** 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
USB Drive free space	Available space in bytes	uint64 read only
USB Drive Name	Name of USB device	string read only
USB Drive Size	Size in bytes of the device plugged in the USB port	uint64 read only
USB Drive Status	0 = USB Drive unplugged 1 = USB Drive plugged	int read only

User management variables

Information on users and groups.

Variable	Description	Data type
This Client User-Name	Name of the user logged to the client where the system variable is displayed.	string read only
This Client Group-Name	Group of currently logged user	string read only
This Client ID	Only for AGI Clients. Local and remote clients connected to the same server (for example, runtime) get a unique ID.	short read only
No Of Remote-Clients Alive	Number of AGI 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.

Protocol Editor Settings

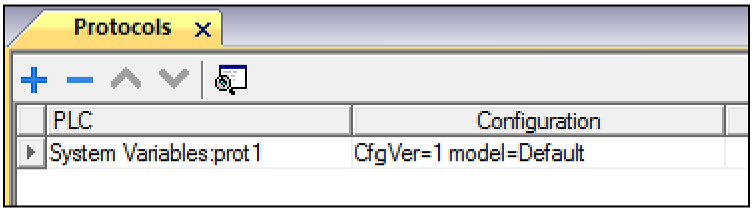
Adding a protocol

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.

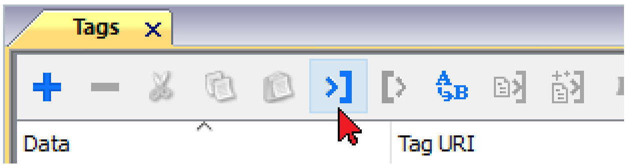
From PLC Model list select the specific System Variables type.



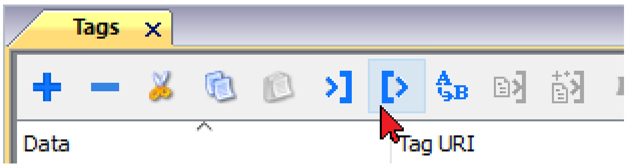
PLC	Configuration
System Variables:prot1	CfgVer=1 model=Default

Tag Import

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

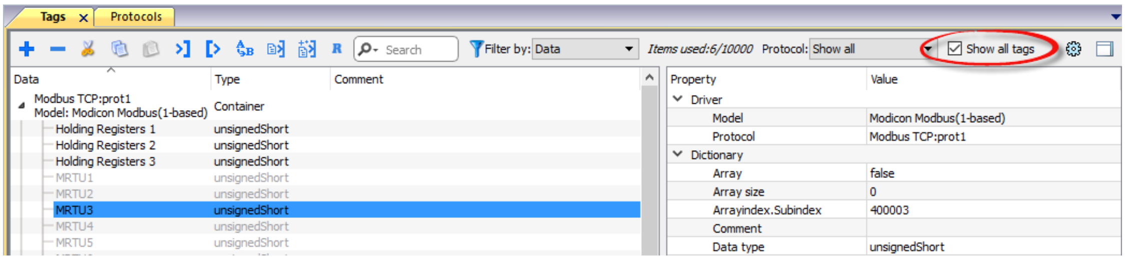


The system will require a generic XML file exported from Tag Editor by appropriate button.



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.

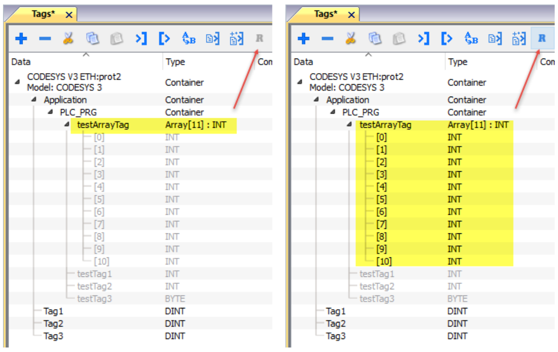


Tags × | Protocols

Search | Filter by: Data | *Items used:6/10000* | Protocol: Show all | ☑ Show all tags

Data	Type	Comment
Modbus TCP:prot1 Model: Modicon Modbus(1-based)	Container	
Holding Registers 1	unsignedShort	
Holding Registers 2	unsignedShort	
Holding Registers 3	unsignedShort	
MRTU1	unsignedShort	
MRTU2	unsignedShort	
MRTU3	unsignedShort	
MRTU4	unsignedShort	
MRTU5	unsignedShort	

Property	Value
Driver	
Model	Modicon Modbus(1-based)
Protocol	Modbus TCP:prot1
Dictionary	
Array	false
Array size	0
Arrayindex.Subindex	400003
Comment	
Data type	unsignedShort

Toolbar item	Description
	Import Tag(s). Select tags to be imported and click on this icon to add tags from tag dictionary to the project
	Update Tag(s). Click on this icon to update the tags in the project, due a new dictionary import.
R	Check this box to import all sub-elements of a tag. Example of both checked and unchecked result:  Tags* × Data / Type / Con CODESYS V3 ETH:prot2 Model: CODESYS 3 — Container Application — Container PLC_PRG — Container testArrayTag — Array[11] : INT [0] — INT [1] — INT [2] — INT [3] — INT [4] — INT [5] — INT [6] — INT [7] — INT [8] — INT [9] — INT [10] — INT testTag1 — INT testTag2 — INT testTag3 — BYTE Tag1 — DINT Tag2 — DINT Tag3 — DINT
Search Filter by: Tag name ▼	Searches tags in the dictionary basing on filter combo-box item selected.

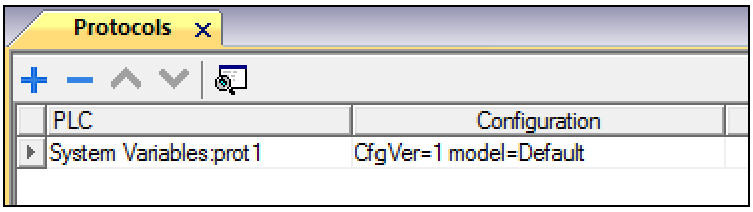
Default variables

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

- Alarms
- Buzzer
- Communication
- 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.

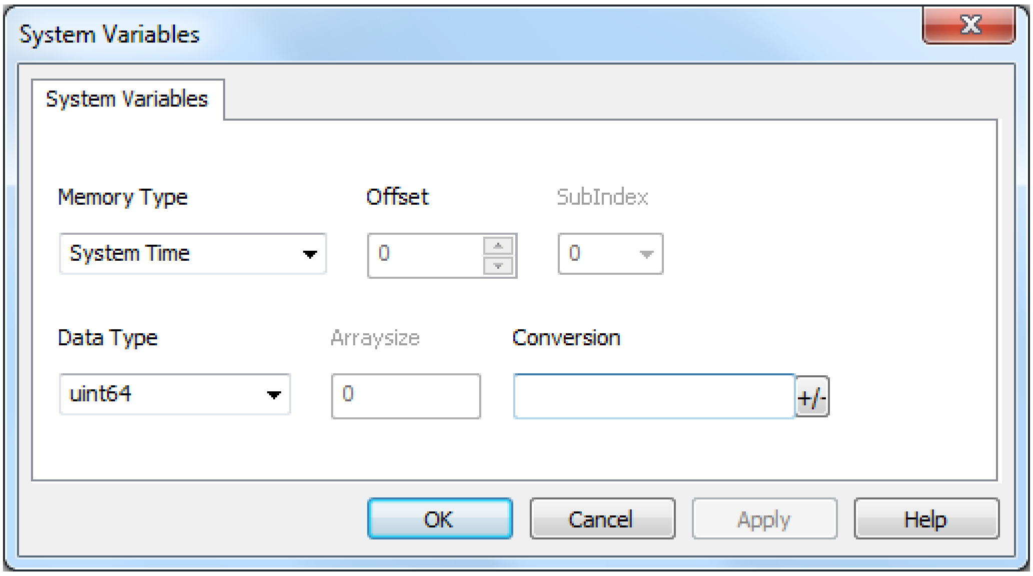


PLC	Configuration
System Variables:prot1	CfgVer=1 model=Default

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.



Memory Type: System Time
Offset: 0
SubIndex: 0
Data Type: uint64
Arraysize: 0
Conversion: +/-

<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td><b>Memory Type</b></td><td>Represents the system variable to which the Tag refers to.
The below section shows the full list of possible system variables, grouped by category.
<table>
<tr><th colspan="3">Alarms Variables</th></tr>
<tr><th>Variable Name</th><th>Description</th><th>Data Type</th></tr>
<tr><td><b>Alarm not acknowledged</b></td><td>True when alarms unacknowledged is pending
(Not Triggered Not Acknowledged<>0) OR (Triggered Not Acknowledged<>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<>0) OR (Triggered <>0) OR (Triggered Not Acknowledged<>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>
</table>
</td></tr>
</table>

<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th colspan="3">Buzzer Variables</th></tr>
<tr><th>Variable Name</th><th>Description</th><th>Data Type</th></tr>
<tr><td><b>Buzzer Setup</b></td><td><b>0</b> = disabled
<b>1</b> = enabled (buzzer sounds as audible on any touchscreen event)
<b>2</b> = buzzer status controlled by Buzzer Control system variable or by Buzzer on Touch property inside the "Project properties" of main manual
⚠ <b>Buzzer on touchscreen (Setup=1) is not available on Linux platforms. See "Buzzer on Touch" property in alternative.</b></td><td>int</td></tr>
<tr><td><b>Buzzer Control</b></td><td><b>0</b> = buzzer off
<b>1</b> = buzzer on
<b>2</b> = buzzer blink</td><td>int</td></tr>
<tr><td><b>Buzzer Off Time</b></td><td>Duration in milliseconds of off time when blink has been selected. Default = 1000. Range: 100–5000</td><td>int</td></tr>
<tr><td><b>Buzzer On Time</b></td><td>Duration in milliseconds of on time when blink has been selected. Default = 1000. Range: 100–5000</td><td>int</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th colspan="3">Communication Variables</th></tr>
<tr><th>Variable Name</th><th>Description</th><th>Data Type</th></tr>
<tr><td><b>Protocol Communication Status</b></td><td>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</td><td>int
read only</td></tr>
<tr><td><b>Protocol Error Message</b></td><td>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.</td><td>string
read only</td></tr>
<tr><td><b>Protocol Error Count</b></td><td>Number of communication errors occurred since last reset. Reset value with Reset Protocol Error Count action, see "System actions" of main manual</td><td>int
read only</td></tr>
</table>
<table>
<tr><th colspan="3">Daylight Saving Time Variables</th></tr>
<tr><th>Variable Name</th><th>Description</th><th>Data Type</th></tr>
<tr><td><b>Standard Offset</b></td><td>Offset in minutes when standard time is set, with respect to GMT (for example: -8x60 = -480 minutes)</td><td>int
read only</td></tr>
<tr><td><b>Standard Week</b></td><td>Week in which the standard time starts (for example: First = 1)</td><td>int
read only</td></tr>
<tr><td><b>Standard Month</b></td><td>Month in which the standard time starts. Range: 0–11. (for example: November = 10)</td><td>int
read only</td></tr>
<tr><td><b>Standard Day</b></td><td>Day of week in which the standard time starts (for example: Sunday = 0)</td><td>int
read only</td></tr>
</table>
</td></tr>
</table>

Element	Description

Daylight Saving Time Variables		
Variable Name	**Description**	**Data Type**
Standard Hour	Hour in which the standard time starts (for example: 02 = 2)	int read only
Standard Minute	Minute in which the standard time starts (for example: 00 = 0)	int read only
DST Offset	Offset in minutes when DLS time is set, with respect to GMT	int read only
DST Week	Week in which the DLS time starts	int read only
DST Month	Month in which the DLS time starts. Range: 0–11	int read only
DST Day	Day of week in which the DLS time starts	int read only
DST Hour	Hour in which the DLS time starts	int read only
DST Minute	Minute in which the DLS time starts	int read only

All variables are read only: they cannot be used to update the system clock.

Device Variables		
Variable Name	**Description**	**Data Type**
Available System Memory	Free available RAM memory in bytes	uint64 read only
Backlight Time	Activation time in hours of the display backlight since production of the device	unsignedInt read only
Battery LED	Enables/disables the low battery LED indicator	int

<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th colspan="3">Device Variables</th></tr>
<tr><th>Variable Name</th><th>Description</th><th>Data Type</th></tr>
<tr><td></td><td>(when available)
<b>0</b> = disabled
<b>1</b> = enabled
ℹ Not available on Linux platforms (find the platform of your device at "HMI devices capabilities" on page 543)</td><td></td></tr>
<tr><td><b>Battery Timeout</b></td><td>Reserved
ℹ Not available on Linux platforms (find the platform of your device at "HMI devices capabilities" on page 543)</td><td>int</td></tr>
<tr><td><b>Display Brightness</b></td><td>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</td><td>int</td></tr>
<tr><td><b>External Timeout</b></td><td>Non-operational time after which the display backlight is automatically turned off. The backlight is automatically turned on when the user touches the screen
<b>-1</b> = Switch off backlight and disable touch (switch display off). <b>Backlight Time</b> counter is stopped.
ℹ On Linux devices requires BSP v1.0.324 or higher.
<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.
ℹ Available only on Linux devices. Requires BSP v1.0.324 or higher.</td><td>int</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th colspan="3">Device Variables</th></tr>
<tr><th>Variable Name</th><th>Description</th><th>Data Type</th></tr>
<tr><td></td><td><b>0</b> = Switch backlight on (switch display on)
<b>1..n</b> = Timeout, in seconds, for switch off backlight (screen saver timer)
ℹ The timeout value is rounded to multiples of one minute (60, 120, 180, etc,) in all Linux devices.
Find the platform of your device at "HMI devices capabilities" on page 543</td><td></td></tr>
<tr><td><b>Flash Free Space</b></td><td>Free space left in internal Flash memory</td><td>uint64
read only</td></tr>
<tr><td><b>Manufacturer Code</b></td><td>Code number that identifies the HMI</td><td>short
read only</td></tr>
<tr><td><b>System RAM Usage</b></td><td>Current RAM memory used from HMI, expressed in byte</td><td>uint64
read only</td></tr>
<tr><td><b>System Font List</b></td><td>List of system fonts</td><td>string
read only</td></tr>
<tr><td><b>System Mode</b></td><td>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</td><td>int</td></tr>
<tr><td><b>System UpTime</b></td><td>Time the system has been powered since production of the unit (hours)</td><td>unsignedInt
read only</td></tr>
</table>
</td></tr>
</table>

Element	Description

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

Network Variables0		
Variable Name	**Description**	**Data Type**
Gateway	Gateway address of the main Ethernet interface of HMI	string read only
IP Address	IP address of the main Ethernet interface of HMI	string read only
Mac ID	MAC ID of the main Ethernet interface of HMI	string read only
Network Adapter	JSON string that can be use to read or update	string

<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th colspan="3">Network Variables0</th></tr>
<tr><th>Variable Name</th><th>Description</th><th>Data Type</th></tr>
<tr><td><b>Parameters</b></td><td>the network adapters parameters</td><td></td></tr>
<tr><td><b>Network Status</b></td><td>Contains the result of the last operation required by writing inside the Adapter Parameters. It is updated after each write operation.
- Empty string is meaning no errors
- Last error descriptions</td><td>string
read only</td></tr>
<tr><td><b>Subnet Mask</b></td><td>Subnet Mask of the main Ethernet interface of HMI</td><td>string
read only</td></tr>
</table>
<table>
<tr><th colspan="3">Screen Variables</th></tr>
<tr><th>Variable Name</th><th>Description</th><th>Data Type</th></tr>
<tr><td><b>X Screen resolution</b></td><td>Display horizontal screen size in pixel</td><td>int
read only</td></tr>
<tr><td><b>Y Screen resolution</b></td><td>Display vertical screen size in pixel</td><td>int
read only</td></tr>
</table>
<table>
<tr><th colspan="3">SD Card Variables</th></tr>
<tr><th>Variable Name</th><th>Description</th><th>Data Type</th></tr>
<tr><td><b>SD Card FreeSpace</b></td><td>Available space on card in bytes</td><td>uint64
read only</td></tr>
<tr><td><b>SD Card Name</b></td><td>Name of SD card</td><td>string
read only</td></tr>
<tr><td><b>SD Card Size</b></td><td>Size in bytes of the card plugged in the slot</td><td>uint64
read only</td></tr>
<tr><td><b>SD Card Status</b></td><td><b>0</b> = SD card unplugged
<b>1</b> = SD card plugged</td><td>int
read only</td></tr>
</table>
</td></tr>
</table>

Element	Description

Server Variables

Variable Name	Description	Data Type
Page name	Name of current page	string read only
Current project	Name of current project	string read only
Project load time	Date when the project was loaded on the HMI Runtime as in System Date format (milliseconds)	uint64 read only
Last operating mode start time	Seconds elapsed since device started operating mode	uint64 read only

All variables refer to server, not to AGI Client.

Time Variables

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

System time expressed in UTC format

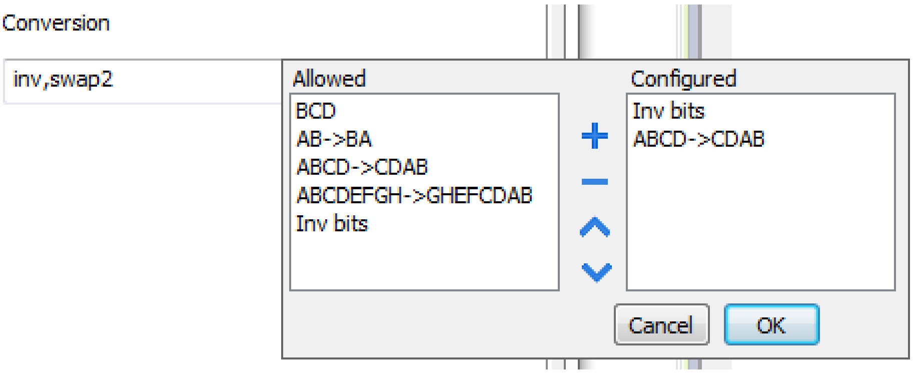
<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th colspan="3">USB Drive Variables</th></tr>
<tr><th>Variable Name</th><th>Description</th><th>Data Type</th></tr>
<tr><td><b>USB Drive FreeSpace</b></td><td>Available space in bytes</td><td>uint64
read only</td></tr>
<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>
</table>
<table>
<tr><th colspan="3">Version Variables</th></tr>
<tr><th>Variable Name</th><th>Description</th><th>Data Type</th></tr>
<tr><td><b>Main OS version</b></td><td>Version of Main OS</td><td>string
read only</td></tr>
<tr><td><b>Runtime version</b></td><td>Version of Runtime</td><td>string
read only</td></tr>
<tr><td><b>Project name</b></td><td>Project name</td><td>string
read only</td></tr>
<tr><td><b>Project version</b></td><td>Project version</td><td>string</td></tr>
<tr><td><b>Project GUID</b></td><td>Project GUID (unique identifier)</td><td>string
read only</td></tr>
</table>
</td></tr>
</table>

Element	Description
	Virtual Com Switch Variables

Virtual Com Switch Variables		
Variable Name	**Description**	**Data Type**
VCS status	Provides status of VCS service. **0** = Service enabled **1** = Client connected in interleaved mode **2** = Client connected in exclusive mode **3** = Service disabled (default)	unsignedByte read only
VCS disable	Provides manual override of VCS service. **0** = VCS service enabled **1** = VCS service disabled (default)	boolean
VCS port	Provides current listening TCP port on HMI by VCS service	unsignedShort

Element	Description
Data Type	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.

Data Type	Memory Space	Limits
short	16-bit data	-32768 ... 32767
int	32-bit data	-2.1e9 ... 2.1e9
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
string	Array of elements containing character code defined by selected encoding	

<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td><b>Arraysize</b></td><td>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.</td></tr>
<tr><td><b>Conversion</b></td><td>Conversion to be applied to the tag.

Conversion
inv,swap2
Allowed
BCD
AB->BA
ABCD->CDAB
ABCDEFGH->GHEFCDAB
Inv bits
Configured
Inv bits
ABCD->CDAB

Depending on data type selected, the list <b>Allowed</b> shows one or more conversion types.
<table>
<tr><th>Value</th><th>Description</th></tr>
<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 -> 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 -> 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 -></b></td><td><b>swap4</b>: Swap bytes in a double word.
<i>Example:</i></td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td><b>GHEFCDAB</b></td><td>32FCFF54 → 54FFFC32 (in hexadecimal format)
855441236 → 1426062386 (in decimal format)</td></tr>
<tr><td><b>ABC...NOP -> OPM...DAB</b></td><td><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)</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>
</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>
</td></tr>
</table>

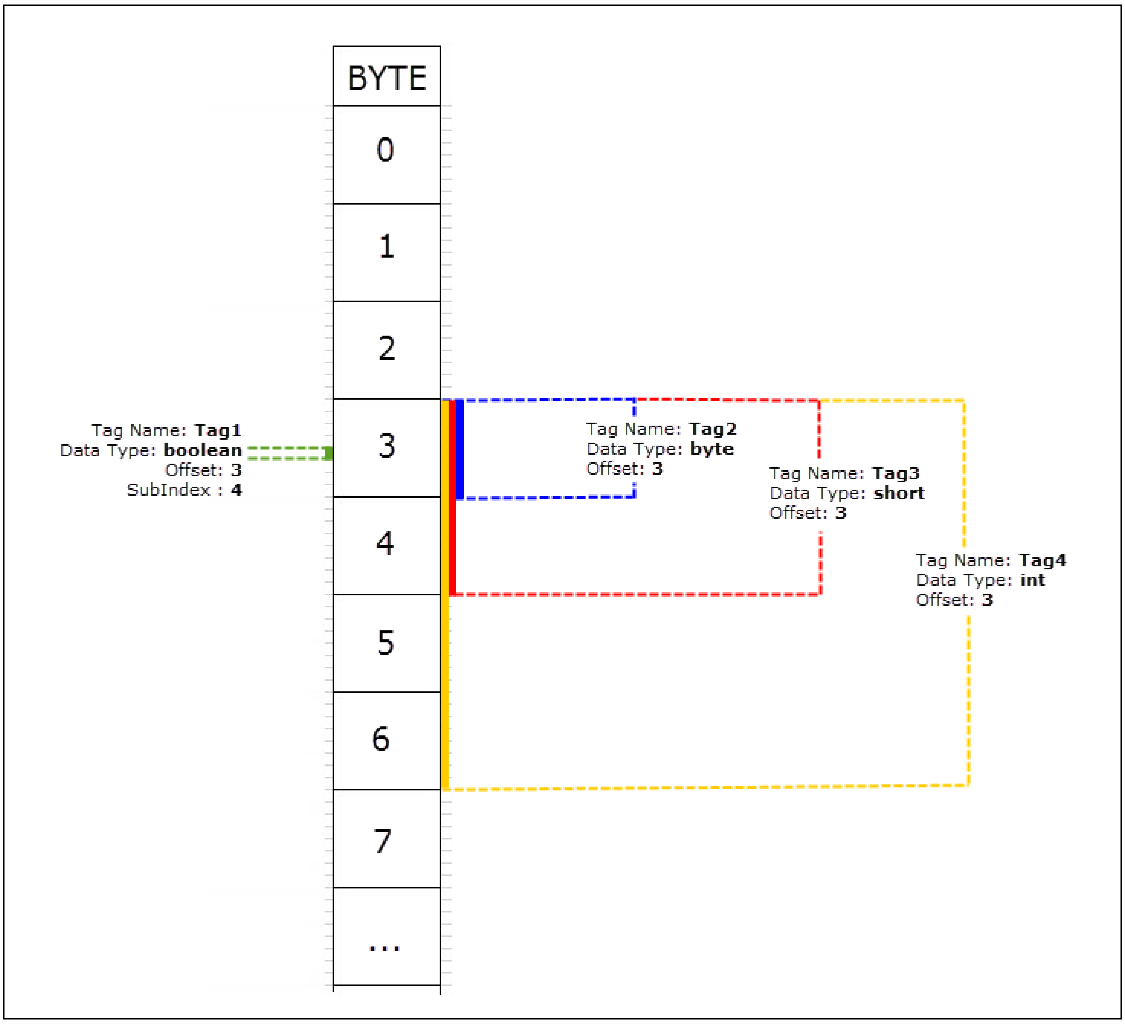
Retentive Memory variables

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 HMI device data for availability of FRAM memory.

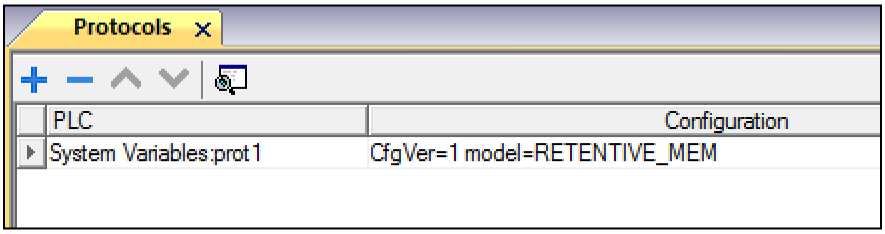
Important: Retentive memory is 16 KB flat memory area organized in bytes and accessible through an offset. Refer to schema below.



Retentive memory vs. recipes storage
Recipe data is saved in flash memory (except forAGI 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.

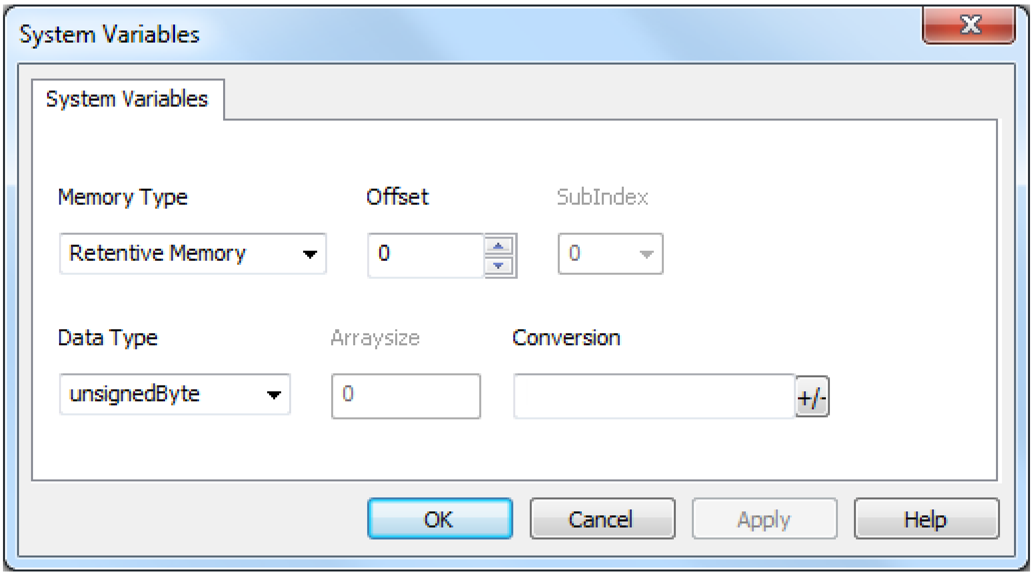


PLC	Configuration
System Variables:prot1	CfgVer=1 model=RETENTIVE_MEM

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.

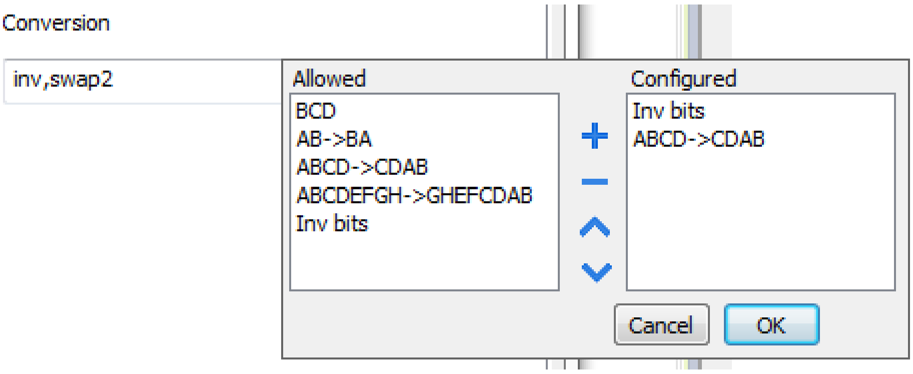


Memory Type: Retentive Memory
Offset: 0
SubIndex: 0
Data Type: unsignedByte
Arraysize: 0
Conversion:

Element	Description
Memory Type	Fixed to Retentive Memory
Offset	Offset address where tag is located. Range: 0-16383
SubIndex	This parameter allows resource offset selection based on selected Data Type
Data Type	

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

<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>Data Type</th><th>Memory Space</th><th>Limits</th></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>
</table>
Note: to define arrays. select one of Data Type format followed by square brackets like “byte []”, “short[]”…
</td></tr>
<tr><td><b>Arraysize</b></td><td>
<ul>
<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>
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.
</td></tr>
<tr><td><b>Conversion</b></td><td>
Conversion to be applied to the tag.



Conversion: inv,swap2

Allowed: BCD, AB->BA, ABCD->CDAB, ABCDEFGH->GHEFCDAB, Inv bits

Configured: Inv bits, ABCD->CDAB

Depending on data type selected, the list <b>Allowed</b> shows one or more conversion types.

<table>
<tr><th>Value</th><th>Description</th></tr>
<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 -> BA</b></td><td><b>swapnibbles</b>: Swap nibbles in a byte.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td></td><td><i>Example:</i>
15D4 → 514D (in hexadecimal format)
5588 → 20813 (in decimal format)</td></tr>
<tr><td><b>ABCD -> 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 -> 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 -> OPM...DAB</b></td><td><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)</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>
</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>
</td></tr>
</table>

.

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.

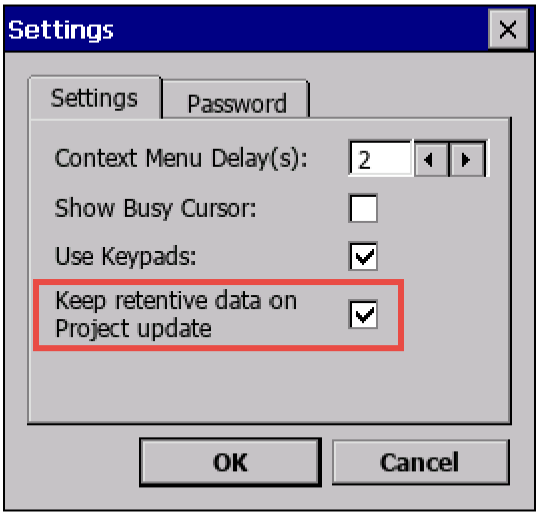
See *Actions* > *Tag Action*s section of main manual for more details.

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.



Settings

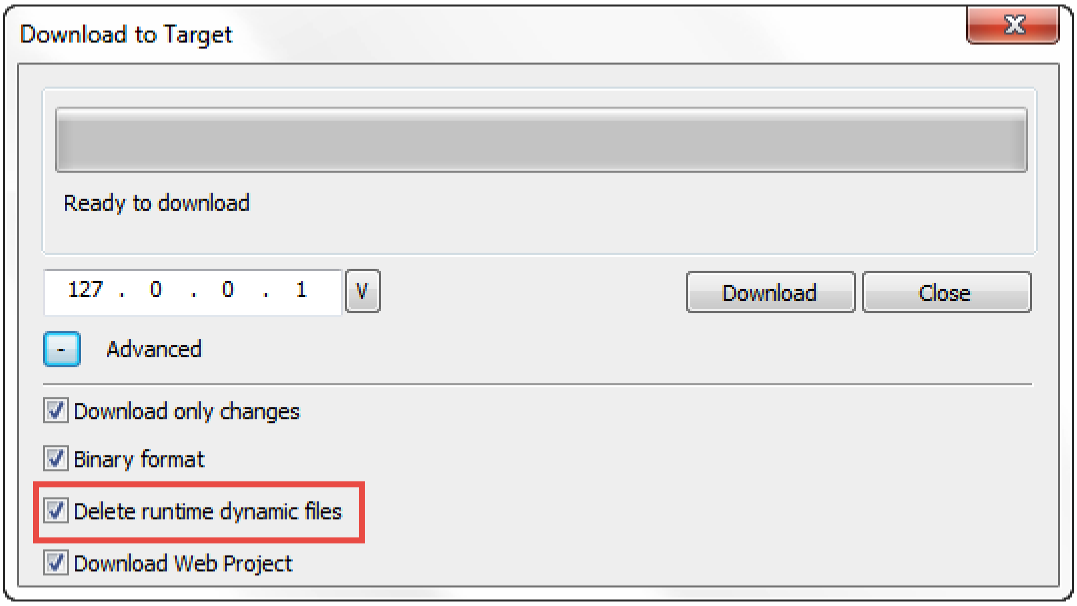
Context Menu Delay(s): 2

Show Busy Cursor: □

Use Keypads: ☑

Keep retentive data on Project update ☑

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



Download to Target

Ready to download

127 . 0 . 0 . 1

☑ Download only changes

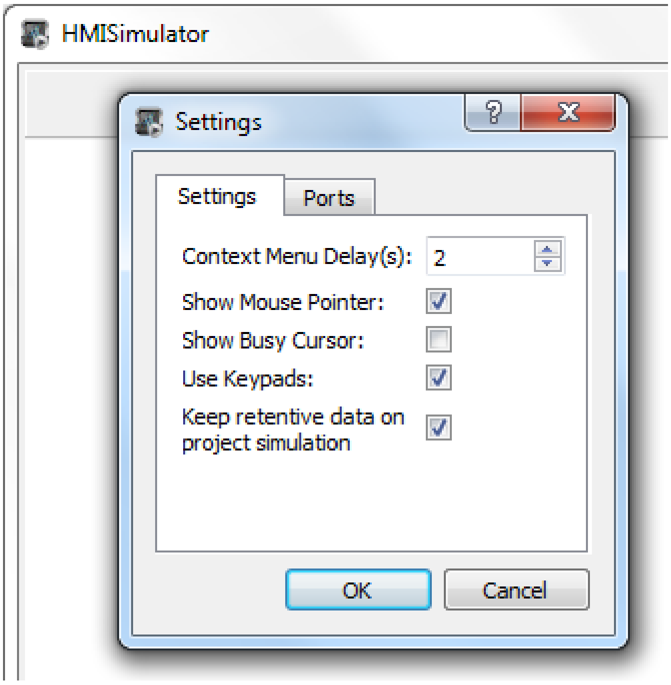
☑ Binary format

☑ Delete runtime dynamic files

☑ Download Web Project

Preserving Retentive Memory in Simulator

Simulator of AGI Creator supports the retentive memory. To enable retentive memory during project simulation use the option "Keep retentive data on project simulation" in context menu**.**



Context Menu Delay(s): 2

Show Mouse Pointer: ☑

Show Busy Cursor: □

Use Keypads: ☑

Keep retentive data on project simulation ☑

Services variables

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

Protocol Editor Settings

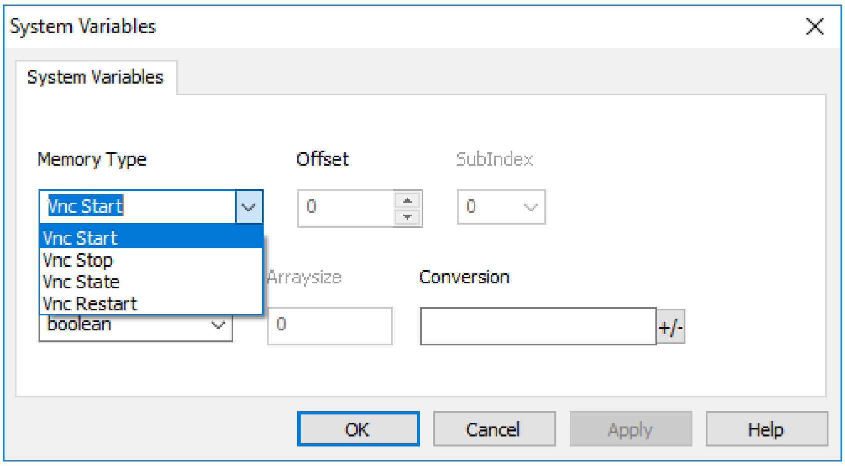
*Path: **ProjectView**> **Protocols***

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.



System Variables

System Variables

Memory Type: Vnc Start

Offset: 0

SubIndex: 0

- Vnc Start
- Vnc Stop
- Vnc State
- Vnc Restart

boolean

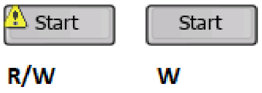
Arraysize: 0

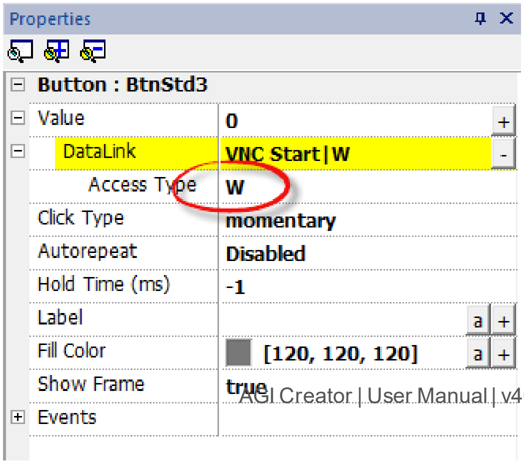
Conversion:

VNC status variables are supported only from Linux devices with BSP version 1.0.344 or greater. (See "HMI devices capabilities" on page 543)

Element	Description	Data type
VNC Start	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
VNC Stop	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
VNC Restart	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
VNC State	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.





Properties

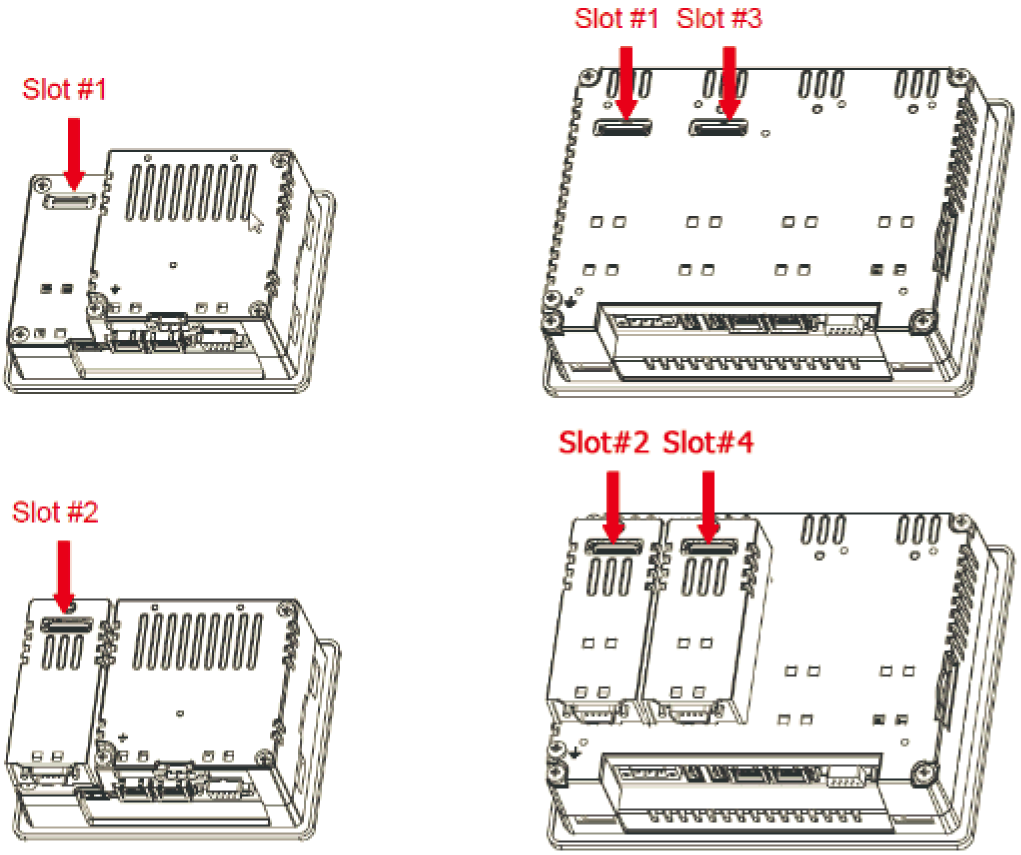
Button : BtnStd3	
Value	0
DataLink	VNC Start|W
Access Type	W
Click Type	momentary
Autorepeat	Disabled
Hold Time (ms)	-1
Label	
Fill Color	[120, 120, 120]
Show Frame	true
Events	

Direct I/O variables

System Variables - Direct I/O protocol allows to create Tags that point to optional local I/O plugin modules.

Install and configure Direct I/O Plug-in modules

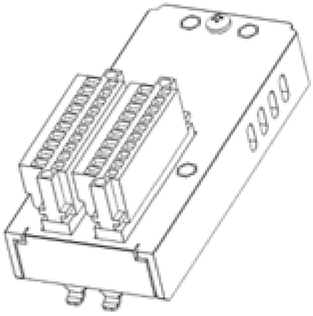
Most HMI offer 1 or 2 slots for connecting optional plug-in modules. Slot numbers are referred in the programming software for configuration of plug-in modules. Numbering of plug-in slots is shown in figure.



Use Plug-in List available in the System Settings menu to check if I/O plug-in modules are correctly recognized in the system and what is their slot number.

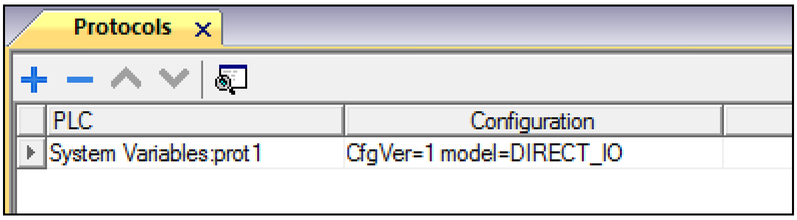
Plug-in I/O Modules details

Plug-in I/O modules have been designed for creating simple applications with a limited number of digital I/O signals.

Module	I/O configration	Image
8 Digital Inputs / 6 Digital Outputs / 1 Relay Output	- 8 optically isolated digital inputs - 6 optically isolated digital outputs - 1 relay output	

Protocol Editor Settings

From PLC Model list of Protocol Editor dialog, select Direct I/O.

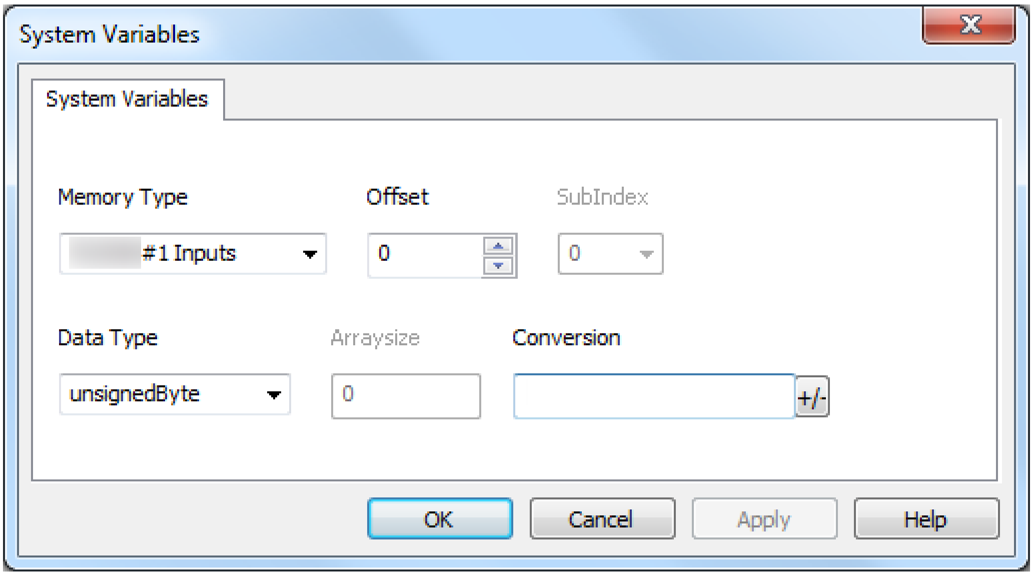


PLC	Configuration
System Variables:prot1	CfgVer=1 model=DIRECT_IO

Tag Editor Settings

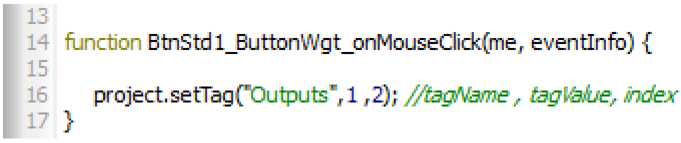
*Path: **ProjectView**> **Config** > double-click **Tags***

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

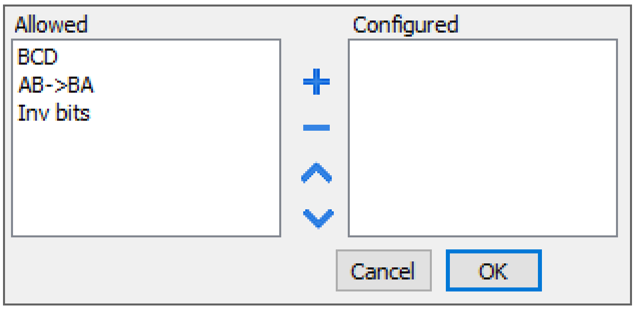


Memory Type: #1 Inputs
Offset: 0
SubIndex: 0
Data Type: unsignedByte
Arraysize: 0
Conversion:

<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td><b>Memory Type</b></td><td>Indicates the resource with this rule:
<module name>#<slot number> <resource type>
<table>
<tr><th>Variable</th><th>Description</th></tr>
<tr><td><b>&lt;<i>module name</i>&gt;#n Diags</b></td><td>Reports I/O diagnostic information. Check <b>Offset</b> section for details</td></tr>
<tr><td><b>&lt;<i>module name</i>&gt;#n Inputs</b></td><td>Reports input status</td></tr>
<tr><td><b>&lt;<i>module name</i>&gt;#n Outputs</b></td><td>Reports ouput status</td></tr>
</table>
</td></tr>
<tr><td><b>Offset</b></td><td>
<table>
<tr><th>Memory Type</th><th>Range</th><th>Description</th></tr>
<tr><td><b>Diags</b></td><td>0 - 2</td><td>
<table>
<tr><th>Bit #</th><th>Description</th></tr>
<tr><td><b>0</b></td><td>DIAG_24VOK parameter.
<b>0</b> = 24V detected
<b>1</b> = 24V missing</td></tr>
<tr><td><b>1</b></td><td>DIGOUT1_DIAG01 parameter.
<b>0</b> = digital ouputs 1-3 not ok
<b>1</b> = digital outputs 1-3 ok</td></tr>
<tr><td><b>2</b></td><td>DIGOUT1_DIAG02 parameter.
<b>0</b> = digital ouputs 4-6 not ok
<b>1</b> = digital outputs 4-6 ok</td></tr>
</table>
</td></tr>
<tr><td><b>Inputs</b></td><td>0 - 7</td><td>
<table>
<tr><th>Bit #</th><th>Description</th></tr>
<tr><td><b>0</b></td><td>Digital Input 1</td></tr>
<tr><td><b>1</b></td><td>Digital Input 2</td></tr>
<tr><td><b>2</b></td><td>Digital Input 3</td></tr>
<tr><td><b>3</b></td><td>Digital Input 4</td></tr>
<tr><td><b>4</b></td><td>Digital Input 5</td></tr>
<tr><td><b>5</b></td><td>Digital Input 6</td></tr>
<tr><td><b>6</b></td><td>Digital Input 7</td></tr>
<tr><td><b>7</b></td><td>Digital Input 8</td></tr>
</table>
</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>Memory Type</th><th>Range</th><th>Description</th></tr>
<tr><td><b>Outputs</b></td><td>0 - 6</td><td>
<table>
<tr><th>Bit #</th><th>Description</th></tr>
<tr><td><b>0</b></td><td>Digital Output 1</td></tr>
<tr><td><b>1</b></td><td>Digital Output 2</td></tr>
<tr><td><b>2</b></td><td>Digital Output 3</td></tr>
<tr><td><b>3</b></td><td>Digital Output 4</td></tr>
<tr><td><b>4</b></td><td>Digital Output 5</td></tr>
<tr><td><b>5</b></td><td>Digital Output 6</td></tr>
<tr><td><b>6</b></td><td>Relay Output</td></tr>
</table>
</td></tr>
</table>
</td></tr>
<tr><td><b>Data Type</b></td><td>
<table>
<tr><th>Data Type</th><th>Memory Space</th><th>Limits</th></tr>
<tr><td><b>unsignedByte</b></td><td>8-bit data</td><td>0 ... 255</td></tr>
<tr><td><b>boolean[]</b></td><td>1-bit data array</td><td>0 ... 1 (for each element)</td></tr>
</table>
Select boolean[] data type to get all resources in a boolean array.
To point or visualize a single resource, a proper Array index property has to be set when using the Tag.
Direct I/O Tags can be accessed with JavaScript programming.
For easier access from JavaScript functions it is preferable to configure Direct I/O tags as boolean[].
Example: JavaScript function to set to TRUE output channel 3

function BtnStd1_ButtonWgt_onMouseClick(me, eventInfo) {

    project.setTag("Outputs",1 ,2); //tagName , tagValue, index
}
</td></tr>
</table>

<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td><b>Arraysize</b></td><td>In case of array Tag, this property represents the number of array elements.
Note: Arraysize depends on the type of plug-in in use. In case of wrong configuration of the tag there will be no error reported, neither in HMI Logger nor in the Protocol Error Message System Variable<convert to text and insert new text here&gt;</td></tr>
<tr><td><b>Conversion</b></td><td>Conversion to be applied to the tag.


Depending on data type selected, the <b>Allowed</b> list shows one or more conversions, listed below.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td><b>BCD</b></td><td>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)</td></tr>
<tr><td><b>AB -> 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>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>
</table>
Select the conversion and click on plus button. The selected item will be added on <b>Configured</b> list.
If more conversions are configured, they will be applied in order (from top to bottom of <b>Configured</b> list).
Use the arrow buttons to order the configured conversions</td></tr>
</table>

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).

Alarm actions

Mainly used to acknowledge or reset alarms.

SelectAllAlarms

Selects all alarms.

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

SelectAlarm

Select a specif alarm.

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

AckAlarm

Acknowlege a specific alarm or all selected alarms.

Parameter	Description
AlarmID	Specific Alarm ID **SELECTED** All selected alarms

Acknowledges selected alarms.

ResetAlarm

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

Parameter	Description
AlarmID	Specific Alarm ID **SELECTED** All selected alarms

EnableAlarms

Enable or disable a specific alarm or all selected alarms.

Disabled alarms will not generate alarm events.

Parameter	Description
AlarmID	Specific Alarm ID **SELECTED** All selected alarms
Selection Flag	**TRUE** Enable the alarm(s). **FALSE** Disable the alarm(s).

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

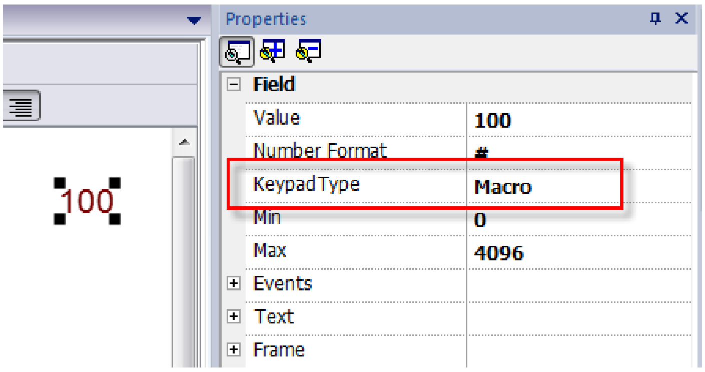
Sets the language used. The selected language will be applied at runtime to all applicable widgets.

Keyboard actions

Changes the use of keypads.

SendKey

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



Properties

Field

Value: 100

Number Format: #

KeypadType: Macro

Min: 0

Max: 4096

Events

Text

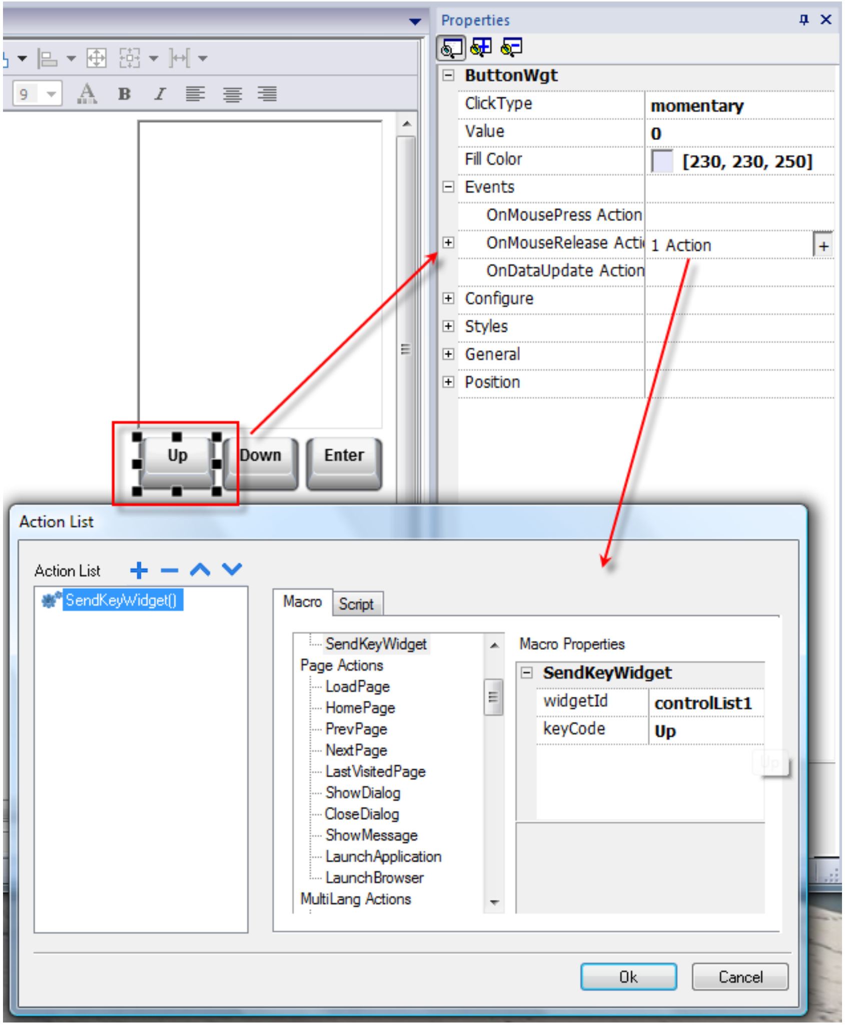
Frame

SendKeyWidget

Sends one character to a specific widget.

Example

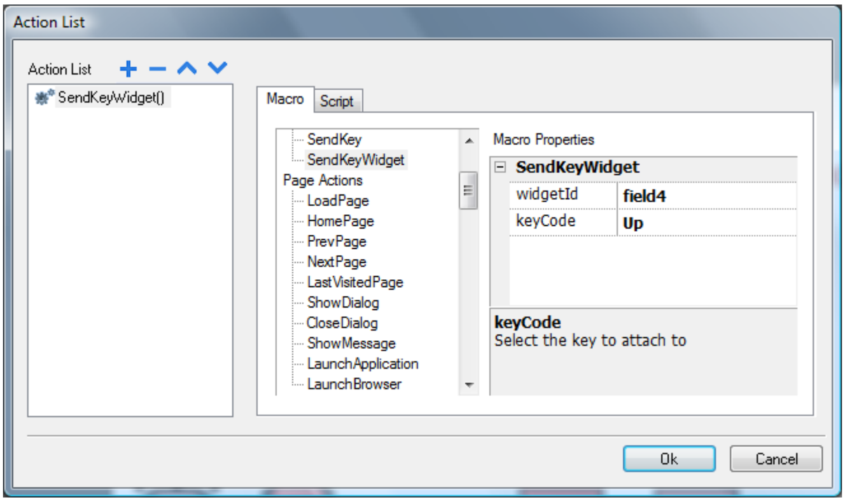
The **Up** and **Down** buttons use the **SendKeyWidget** action in association with the **Control List Widget**.



ButtonWgt
ClickType: momentary
Value: 0
Fill Color: [230, 230, 250]
Events
OnMousePress Action
OnMouseRelease Acti: 1 Action
OnDataUpdate Action
Configure
Styles
General
Position

Action List: SendKeyWidget()

Macro Properties
SendKeyWidget
widgetId: controlList1
keyCode: Up



Action List: SendKeyWidget()

Macro Properties
SendKeyWidget
widgetId: field4
keyCode: Up

keyCode
Select the key to attach to

ShowKeyPad

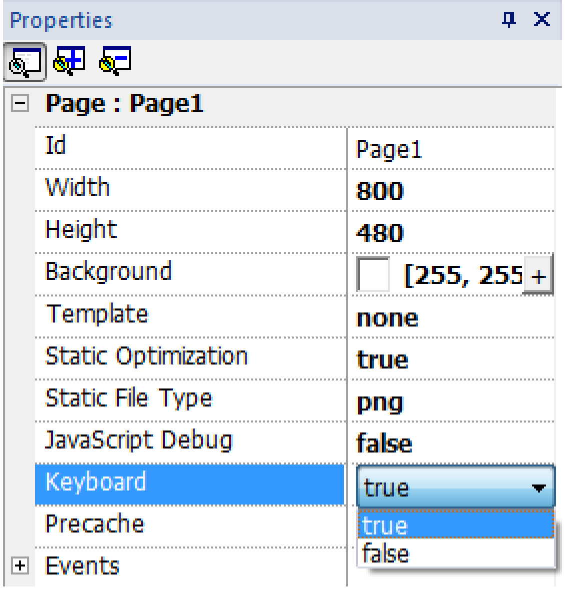
Shows the default operating system touch keypad.

Note: might not be supported by all operating systems.

Keyboard

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.



Page : Page1	
Id	Page1
Width	800
Height	480
Background	[255, 255
Template	none
Static Optimization	true
Static File Type	png
JavaScript Debug	false
Keyboard	true
Precache	
Events	

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 139 for details).

FTP Server Configuration

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

Parameter	Description
FTP Address	FTP server IP Address
Server Port	Port for FTP connection (default = 21).
Authentication	Select the FTP authentication to use: • Normal (Username and password required) • Anonymous
User Name	Username of the remote FTP account
Password	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
IPAddress	FTP server IP Address.
Port	Port for FTP connection (default = 21).
Authentication	Select the FTP authentication to use: • Normal (Username and password required) • Anonymous
UserName	Username of the remote FTP account
Password	Password of the remote FTP account

ftpGET

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

Download files from a remote FTP server

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

ftpPUT

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

Upload files to a remote FTP server

Parameter	Description
remoteFileName	File name on the remote FTP server to download (source)
localFileName	File name on local HMI device (destination)
callback	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

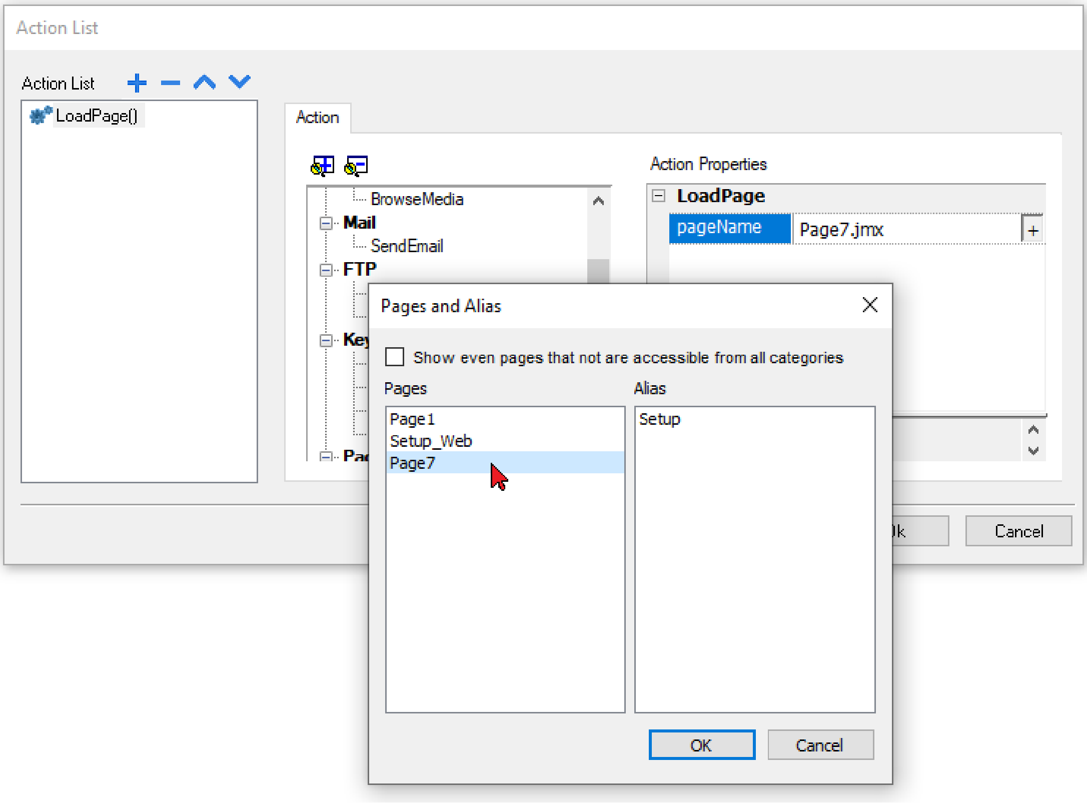
Page navigation. Page actions can be used with the following events:

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

LoadPage

Go to the selected page of the project.

Starting from AGI Creator v4.0 in addition to the pages you can use the aliases (see "Alias pages" on page 75)



HomePage

Go to the home page.

You can set the home page in the **Behavior** section of the **Project Widget**, see "Project" on page 86

PrevPage

Go to the previous page.

NextPage

Go to the next page.

LastVisitedPage

Go to the previously displayed page

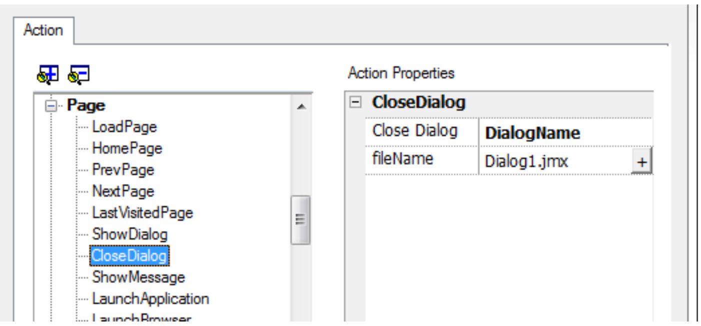
ShowDialog

Opens a dialog page defined in the project.

CloseDialog

Close dialog pages.

Note: This action is applicable only to dialog pages.



CloseDialog	
Close Dialog	DialogName
fileName	Dialog1.jmx

CloseDialog options

Option	Description
All	Closes all open dialogs
Selected	Closes only active dialog
DialogName	Closes dialog specified as **fileName** property

JavaScript Interface

project.closeDialog(DialogID);

Where *DialogID*:

All	Closes all open dialogs
Selected	Closes only active dialog
DialogName.jmx or AliasName	Closes dialog specified as **fileName** parameter

Examples

Example	Behavior
project.closeDialog("All");	All open dialogs are closed
project.closeDialog("Selected");	The selected dialog is closed
project.closeDialog("Dialog1.jmx");	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.

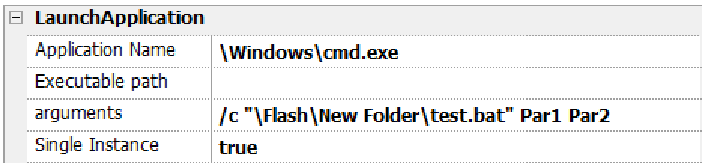
LaunchApplication

Launches an external application.

Parameter	Description
App Name	Executable name with extension (for example, "notepad.exe" to run Notepad)
Path	Application path.
Arguments	Application specific arguments (for example, *\flash\qthmi\Manual.pdf* to open the document "Manual.pdf")
Single Instance	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.
FlushRuntimeCache	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 "Linux Devices" on page 548 BSP v1.0.44 or higher required.

See "Plug-in" on page 85 to include it on Windows CE devices.

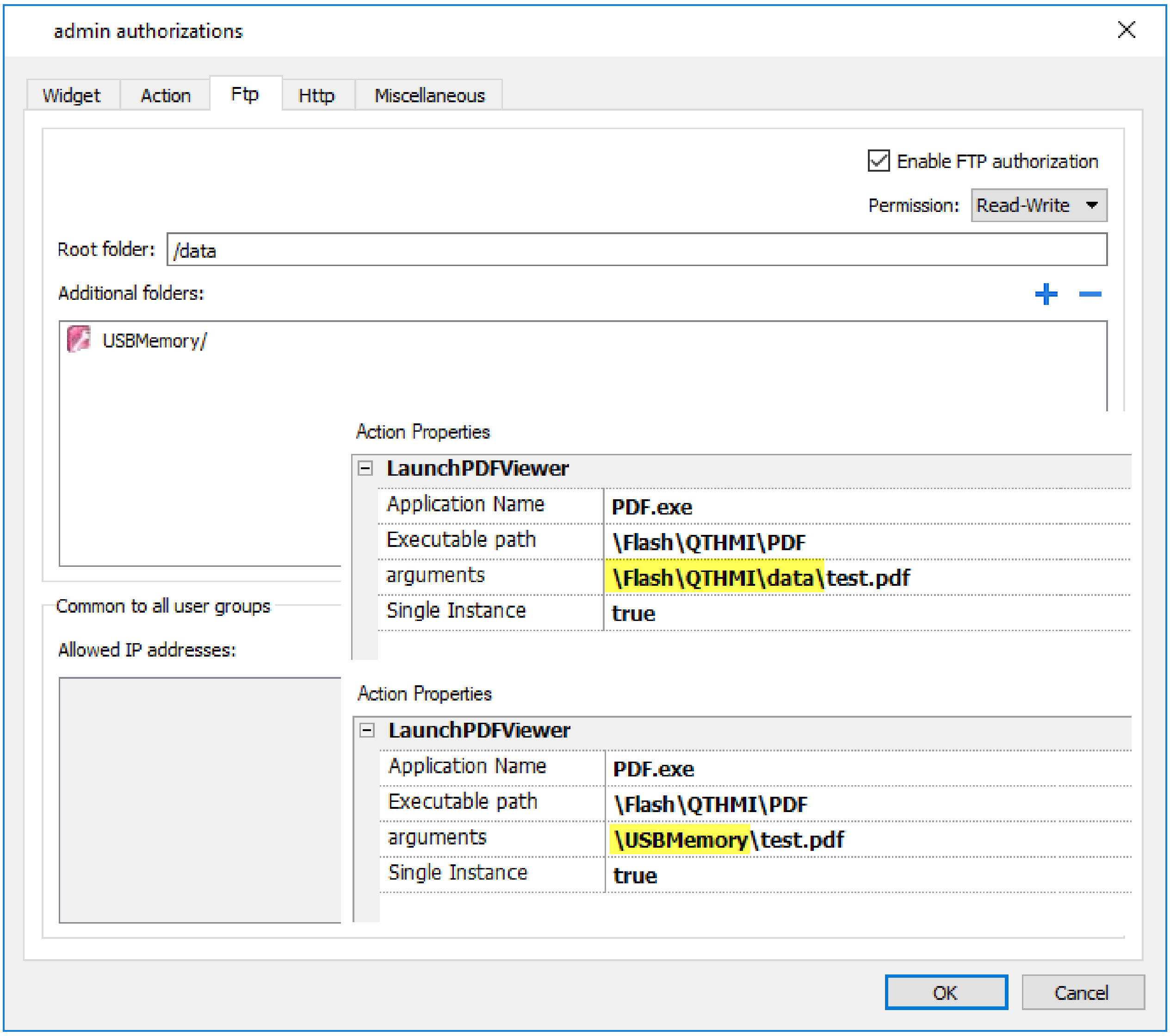
LaunchPDFViewer

Starts PDF Viewer.

On WCE devices, only works on devices that include PDF Viewer. See "Plug-in" on page 85 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 543).

On **WinCE devices**, the HMI application is installed on path \Flash\QTHMI\ and pathname's syntax use the backslash character.



admin authorizations

☑ Enable FTP authorization

Permission: Read-Write

Root folder: /data

Additional folders:

USBMemory/

Allowed IP addresses:

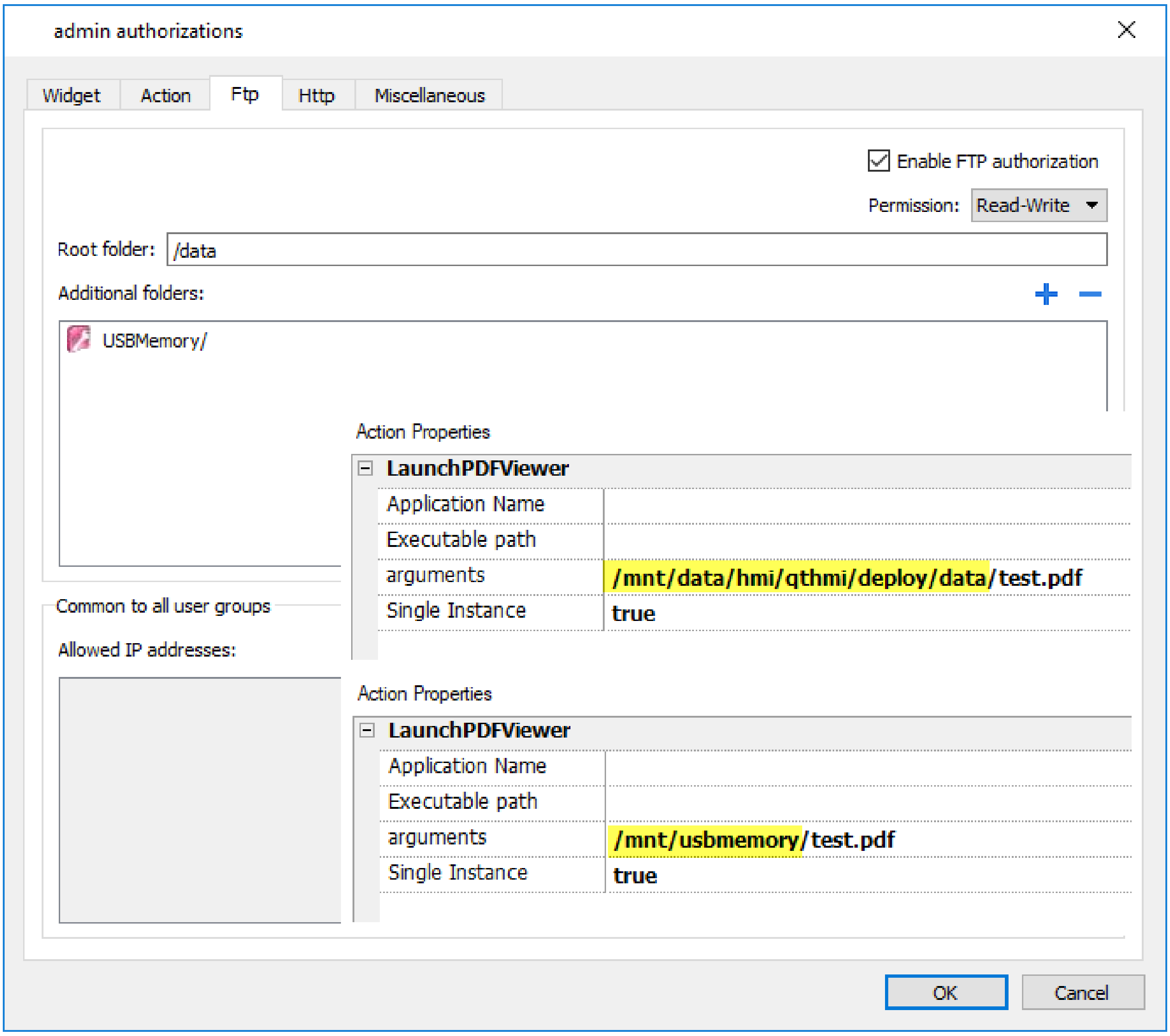
Action Properties

LaunchPDFViewer	
Application Name	PDF.exe
Executable path	\Flash\QTHMI\PDF
arguments	\Flash\QTHMI\data\test.pdf
Single Instance	true

Action Properties

LaunchPDFViewer	
Application Name	PDF.exe
Executable path	\Flash\QTHMI\PDF
arguments	\USBMemory\test.pdf
Single Instance	true

On **Linux devices**, the HMI application is installed on path /mnt/data/hmi/qthmi/deploy/ and pathname's syntax use the slash character.



admin authorizations

☑ Enable FTP authorization

Permission: Read-Write

Root folder: /data

Additional folders:

USBMemory/

Common to all user groups

Allowed IP addresses:

Action Properties

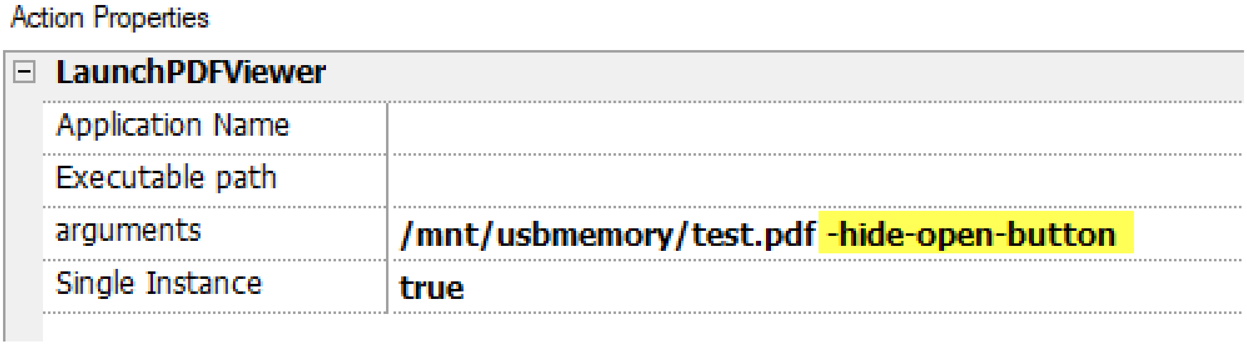
LaunchPDFViewer	
Application Name	
Executable path	
arguments	/mnt/data/hmi/qthmi/deploy/data/test.pdf
Single Instance	true

Action Properties

LaunchPDFViewer	
Application Name	
Executable path	
arguments	/mnt/usbmemory/test.pdf
Single Instance	true

-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

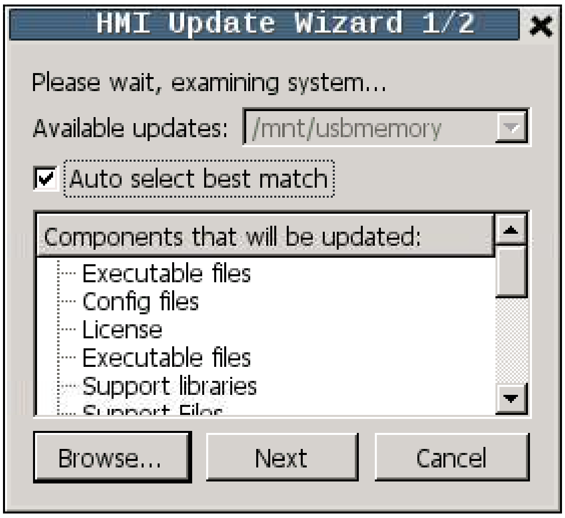
LaunchPDFViewer	
Application Name	
Executable path	
arguments	/mnt/usbmemory/test.pdf -hide-open-button
Single Instance	true

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



HMI Update Wizard 1/2

Please wait, examining system...

Available updates: /mnt/usbmemory

☑ Auto select best match

Components that will be updated:

- Executable files
- Config files
- License
- Executable files
- Support libraries
- Support Files

Note: Not supported in devices based on Windows platform.

JavaScript Interface

project.launchUpdater(strPath)

Examples

```
project.launchUpdater("\\USBMemory")
```

LockScreen

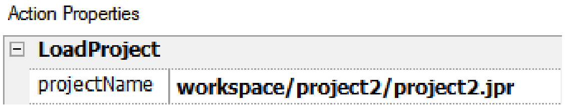
Temporarily locks the touch screen. Allows cleaning the touch screen.

The system variable **Time remaining to unlock** displays the time remaining to unlock.See "Screen variables" on page 143

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
reportName	Assigns a name to the report
silent	**false** = allows to set printer properties at runtime
fileName	File name (available only for PDF reports) Supported placeholders: - %n = Report name - %p = Project name - %y = Year, %M = Month, %d = Day - %h = Hour, %m = Minutes, %s = Seconds.
folderPath	Folder Path (available only for PDF reports) Note that the pathname of the arguments field uses native OS format (see "HMI devices capabilities" on page 543). - On WinCE devices Path for USB Device is "\USBMemory" - On Linux devices Path for USB Device is "/mnt/usbmemory" "testFolder" will be inside "/mnt/data/hmi/qthmi/deploy/testFolder"
Signed	When the output is a PDF file, generate a signed file using the x.509 certificate of the panel. ⚠ **On Linux devices, the BSP v1.0.507 or greater is required** **On WinCE devices, the BSP v2.31 or greater is required** ℹ The algorithm to use to signing is defined inside the project properties parameters See "Project" on page 86 for the available algorithms See also: - "Signed PDF files" on page 328

PrintText

Prints a string.

Parameter	Description
text	String to be printed
silent	**false** = 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 543)

PrintBytes

Prints an hexadecimal string representing data to print (for example, "1b30" to print < ESC 0 >.

Parameter	Description
bytes	Exadecimal string to print
silent	**false** = 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 543)

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
RecipeName	Name of recipe to download
RecipeSet	Number of recipe set to copy. **curSet** = 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
RecipeName	Name of recipe to upload
RecipeSet	Number of recipe set to copy. **curSet** = upload currently selected recipe set

WriteCurrentRecipeSet

Sets the selected recipe as current recipe set.

Parameter	Description
RecipeName	Name of recipe to set as current recipe
RecipeSet	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
RecipeName	Name of recipe to dump
FilePath	Destination folder - Internal = *\Flash\QTHMI\workspace\Dump* - USB drive = *\USBMemory* - SD Card = *\Storage Card* - Public Network = *\\<hostname or IP>\sharePath* - Private Network = *\\<username>:<password>@<hostname or IP>\sharePath* Note: supported formats for external memory are FAT or FAT32 (NTFS format is not supported). Note: Private networks are supported only from Linux devices with BSP 1.0.25 and above.
FileName	Tag that specifies a filename. The below wildcards are supported - %r = Recipe name - %d = Dataset name Example: %r_%d
DateTimePrefixFileName	**true** = the dumped file will have date and time as prefix to its name (for example D2012_01_01_T10_10_recipe1.csv)
TimeSpec	Time format: - **Local** = the time values exported are the time of the HMI device. - **Global** = the time values exported are in UTC format.

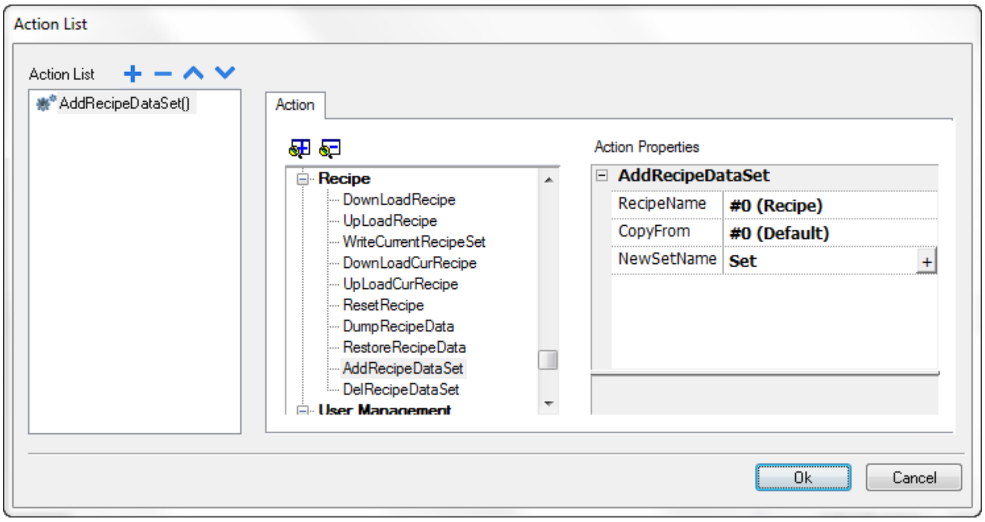
RestoreRecipeData

Restores previously saved recipe data.

Parameter	Description
RecipeName	Recipes to restore: • AllRecipes Data of all recipes will replaced with the data read from the external file • CurrentRecipe Only the data of the current selected recipe will replaced with the data read from the external file
RecipeDataSet	Available only when RecipeName=CurrentRecipe. Select the data sets to restore: • AllRecipeDataSet All data set will restored • curSet Only the data set of the current selected data set will restore
Restore Type	Available only when RecipeDataSet=AllRecipeDataSet. This parameter define the behavior when the numbers of data sets inside the file to restore is not matching with the data set number inside the HMI device • Replace All data sets that are inside the device are removed and replaced with the data sets from the csv file • Match Replace only the data set inside the device that have the same data set id • MatchAndAdd Replace the data set inside the device that have the same data set id and add the additional data set found inside the csv file (Note: data sets that are inside the device but not inside the csv file are not removed from the device)
FilePath	Source folder • Internal = *\Flash\QTHMI\workspace\Dump* • USB drive = *\USBMemory* • SD Card = *\Storage Card* • Public Network = \\\\<*hostname or IP*>*\sharePath* • Private Network = \\\\<*username*>:<*password*>@<*hostname or IP*>*\sharePath* Note: supported formats for external memory are FAT or FAT32 (NTFS format is not supported). Note: Private networks are supported only from Linux devices with BSP 1.0.25 and above.
FileName	Attached tag from which read the file name at runtime.
BrowseForFile	**true** = shows the Open dialog to browse the file to read. **false** = no dialog is shown,

AddRecipeDataSet

Adds a new dataset to the selected recipe. The new dataset is appended at the end of the already defined datasets.



Action Properties

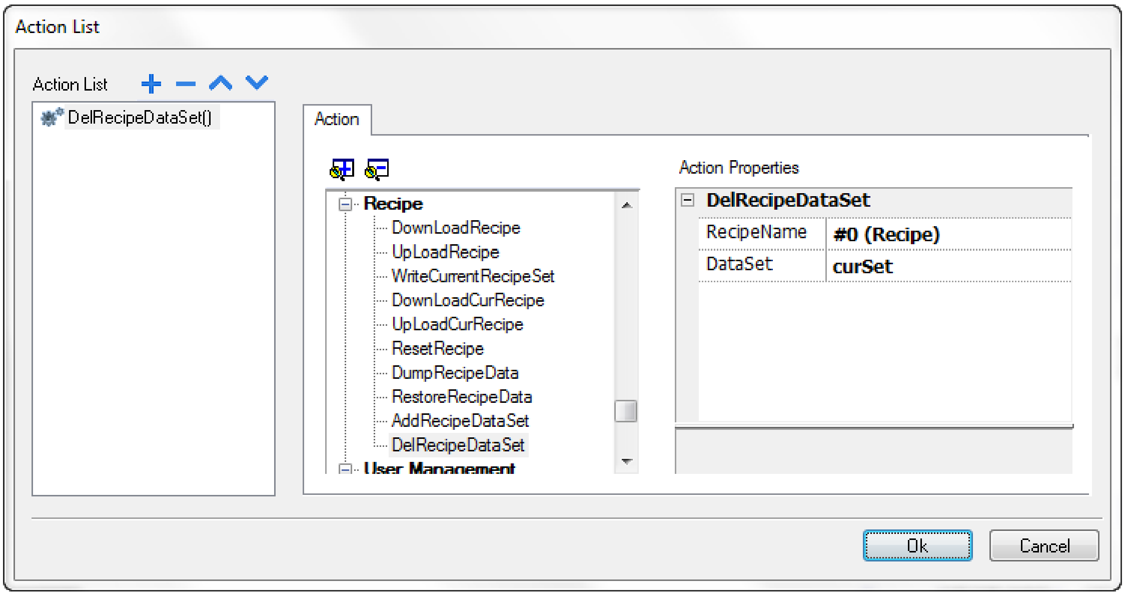
AddRecipeDataSet

RecipeName	#0 (Recipe)
CopyFrom	#0 (Default)
NewSetName	Set

Parameter	Description
RecipeName	Recipe where the dataset is added.
CopyFrom	Dataset from where parameters values are copied from to initialize the new dataset
NewSetName	Name of new dataset. Here you can 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.



Action Properties

DelRecipeDataSet

RecipeName	#0 (Recipe)
DataSet	curSet

Parameter	Description
RecipeName	Recipe where the dataset is to be deleted.
DataSet	Dataset to be deleted.

Remote Client actions

Used to upload and download files to and from a remote HMI device. These actions can only be used from a remote AGI 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 file Open dialog to select a file 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 displayed separated by commas (for example, *.txt)

DownloadFromHMI

Opens a file Open dialog to select a file to be downloaded from the remote HMI device.

Note: Only files matching the set filter are displayed and can be downloaded.

Parameter	Description
Source	Source path on the HMI device for file download
Filter	File extensions of the files to be displayed separated by commas (for example, *.txt)

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.

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
TrendName	Name of historical trend to store
FolderPath	Destination folder: - Internal = *\Flash\QTHMI\workspace\Dump* - USB drive = *\USBMemory* - SD Card = *\Storage Card* - Public Network = *\\<hostname or IP>\sharePath* - Private Network = *\\<username>:<password>@<hostname or IP>\sharePath* Note: supported formats for external memory are FAT or FAT32 (NTFS format is not supported). Note: Private networks are supported only from Linux devices with BSP 1.0.25 and above.
FileFormat	**Binary** = 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. **Compatibility CSV** = the buffer is dumped to the specified location as a .csv file format compatible with versions 1.xx **Compact CSV** = the buffer is dumped to the specified location as a .csv file using a newer format See "Exporting trend buffer data" on page 272
DateTimePrefix	**true** = the dumped file will have date and time as prefix to its name (for example D2012_01_01_T10_10_Trend1.csv)

Parameter	Description
TimeSpec	Time format: - **Local** = the time values exported are the time of the HMI device. - **Global** = the time values exported are in UTC format.
FileName	Enabled when the DateTimePrefixFileName=true The below wildcards are supported - %n = Trend name - %y = Year - %M = Month - %d = Day - %h = Hour - %m = Minutes - %s = Seconds Example: \%n\%y%M%d\%h%m%s

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 272

Parameter	Description
Select Fields	Select the columns to export inside the dumped file. Available columns are: - Date and Time - Date - Time - Value - Quality 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. Example: - "" *(Empty string = all available fields)* - "DateTime,Value,Quality" - "Date,Time,Value"
Select Curves	Select the curves to export inside the dumped file 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.

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td></td><td>Example:
• Empty string or "All curves" will export all datasets
• "Name1,Name2,Name3"
• "Name1,Name3"</td></tr>
<tr><td><b>Date Format</b></td><td>Select the Date and Time format
Using "Attach to tag" is possible define the date format at runtime through a string
<table>
<tr><th colspan="2">Date Placeholder</th></tr>
<tr><td><b>d</b></td><td>The day as number without a leading zero (1 to 31)</td></tr>
<tr><td><b>dd</b></td><td>The day as number with a leading zero (01 to 31)</td></tr>
<tr><td><b>ddd</b></td><td>The abbreviated localized day name (e.g. 'Mon' to 'Sun')</td></tr>
<tr><td><b>dddd</b></td><td>The long localized day name (e.g. 'Monday' to 'Sunday').</td></tr>
<tr><td><b>M</b></td><td>The month as number without a leading zero (1-12)</td></tr>
<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>
</table>
<table>
<tr><th colspan="2">Time Placeholder</th></tr>
<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>
</table>
</td></tr>
</table>

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th colspan="2">Time Placeholder</th></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>
</table>
</td></tr>
<tr><td><b>Language</b></td><td>Select the language to use.</td></tr>
</table>

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 275 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 AGI Creator 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
Data Type	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
Value	Value of the sample
Timestamp (UTC)	Timestamp in UTC format
Sampling Time(ms)	Sampling interval time in milliseconds
Quality	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: QQSSSSLL. 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

Quality Code	Quality	Description
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 **Limits** 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 **Limits** 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 **Limits** field indicates which limit has been exceeded but the value can move farther out of this range. The value has ‘pegged’ at some higher limit
87	UNCERTAIN	Returned value outside its defined limits defined. In this case the **Limits** 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
EventArchive	Name of buffer to dump data
FolderPath	Destination folder - Internal = *\Flash\QTHMI\workspace\Dump* - USB drive = *\USBMemory* - SD Card = *\Storage Card* - Public Network = *\\<hostname or IP>\sharePath* - Private Network = *\\<username>:<password>@<hostname or IP>\sharePath* ℹ Note: supported formats for external memory are FAT or FAT32 (NTFS format is not supported). ℹ Note: Private networks are supported only from Linux devices with BSP 1.0.25 and above.
DumpConfigFile	Dump the description files of the archives
DumpAsCSV	**true** = the buffer is dumped to the specified location as a .csv file **false** = 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.
DateTimePrefix	**true** = the dumped file will have date and time as prefix to its name (for example D2012_01_01_T10_10_alarmBuffer1.csv)
timeSpec	Time format: - **Local** = the time values exported are the time of the HMI device. - **Global** = the time values exported are in UTC format.
csv Colums	Select the columns to dump into the .csv file. ℹ Available only when the EventArchive is an alarms buffer
FileName	The below wildcards are supported - %n = Event archive name - %y = Year - %M = Month - %d = Day - %h = Hour - %m = Minutes - %s = Seconds

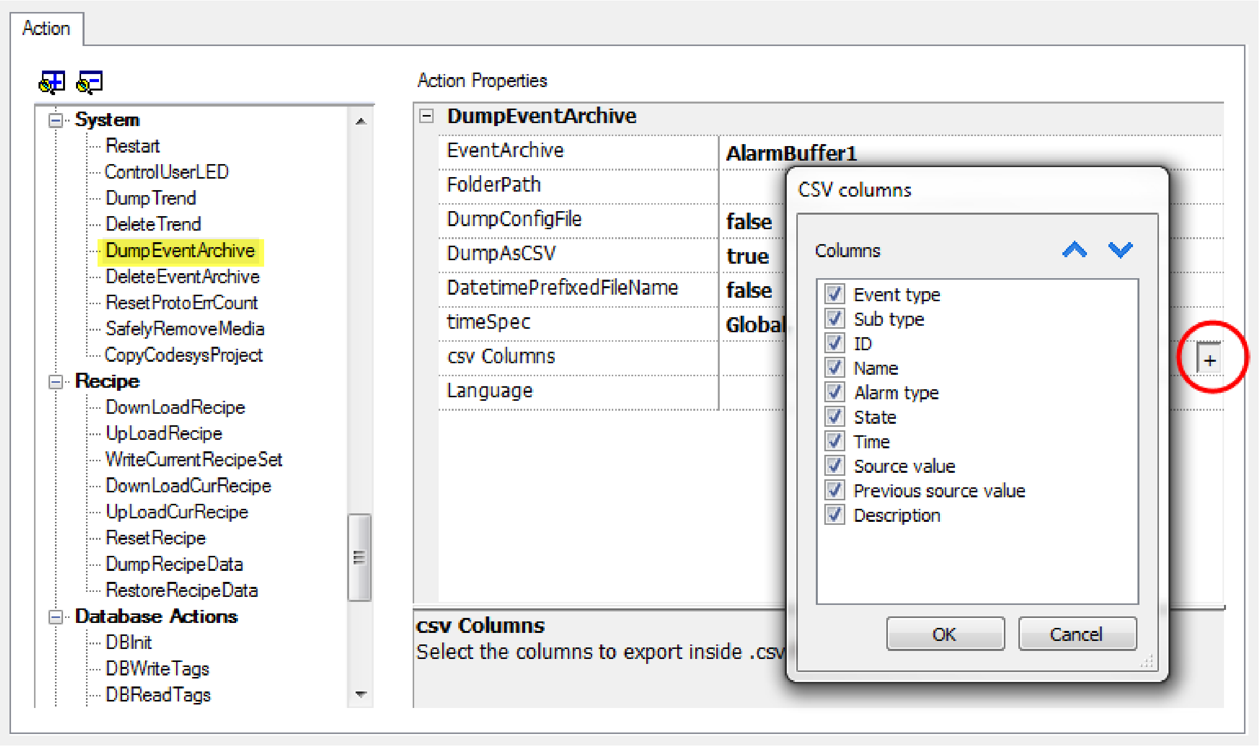
Parameter	Description
	Example: \%n\%y%M%d\%h%m%s Available only when the DateTimePrefixFileName=true
Language	Select the language to use. Available only when the EventArchive is an alarms buffer

Dumping in CSV Format

DumpAsCSV = true

For Alarms buffers, the additional "csv Colums" parameter give the possibility to select the columns to export inside the .csv file

Note: available only for Alarms buffers.



DumpEventArchive	
EventArchive	AlarmBuffer1
FolderPath	
DumpConfigFile	false
DumpAsCSV	true
DatetimePrefixedFileName	false
timeSpec	Global
csv Columns	
Language	

☑ Event type
☑ Sub type
☑ ID
☑ Name
☑ Alarm type
☑ State
☑ Time
☑ Source value
☑ Previous source value
☑ Description

csv Columns
Select the columns to export inside .csv

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 AGI Creator 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 maintenanced 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.

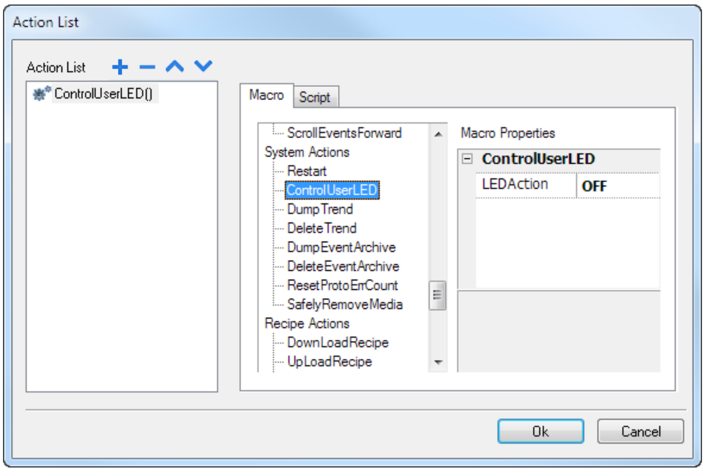
See "System Variables (Attach To)" on page 133 for details.

SafelyRemoveMedia

Provides for safe removal of SD card or USB drive fromHMI.

ControlUserLED

Sets the user LED behavior.



ControlUserLED

LEDAction	OFF

Not available on Linux platforms (find the platform of your device at "HMI devices capabilities" on page 543)

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.

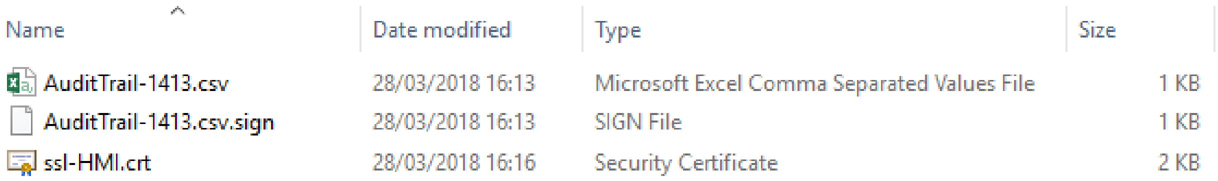
Parameter	Description
EventArchive	Name of buffer to dump data
FolderPath	Destination folder - Internal = *\Flash\QTHMI\workspace\Dump* - USB drive = *\USBMemory* - SD Card = *\Storage Card* - Public Network = *\\<hostname or IP>\sharePath* - Private Network = *\\<username>:<password>@<hostname or IP>\sharePath* Note: supported formats for external memory are FAT or FAT32 (NTFS format is not supported). Note: Private networks are supported only from Linux devices with BSP 1.0.25 and above.
FileName	The below wildcards are supported

Parameter	Description
	- %n = Event archive name - %y = Year - %M = Month - %d = Day - %h = Hour - %m = Minutes - %s = Seconds Example: \%n\%y%M%d\%h%m%s
Format	Format of the output file - CSV
Signed	Generate the file signature. ⚠ **On Linux devices, the BSP v1.0.239 or greater is required** **On WinCE devices, the BSP v2.29 or greater is required** ℹ The algorithm to use to signing is defined inside the project properties parameters See "Project" on page 86 for the available algorithms See also: - "Signed CSV files" on page 327
TimeSpec	Time format: - **Local** = the time values exported are the time of the HMI device. - **Global** = the time values exported are in UTC format.
PeriodMode	Defines the time window to export - All events - Today - Yesterday - Last week - Last month - Current week - Current month - Custom The additional parameters "periodFrom" and "periodTo" will be shown
Separate Date and Time	Uses two separate columns for Date and Time
Date Format	Select the Date and Time format

Signed file

When the "Signed file" parameter is true, two files will be added in addition to fileame.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 324.

For more information about the exported information see "Exporting audit trail as .csv files" on page 352.

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
EventArchive	Name of the audit buffer where add the message
Message	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
FolderPath	Folder where search the files to delete
FileTypes	List of files to delete separate by comma. Wildcard are supported Example: *.png,*.jpg
OlderDays	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	This property allows to chose if adding/subtracting step values at every single wheel step, or at every rotation event. **false** = The step value is added/subtracted to the Tag at every rotation event. *Example: rotate the wheel performing 5 wheel steps in a single event, Tag will be increased/decreased by 1.* **true** = The step value is added/subtracted to the Tag at every single wheel step. *Example: rotate the wheel performing 5 wheel steps in a single rotation, Tag will be increased/decreased by 5.*
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 OnWell 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: **False** = disabled **True** = enabled

Parameter	Description
	When attached to a tag, tag = 0 means **False**

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.

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.

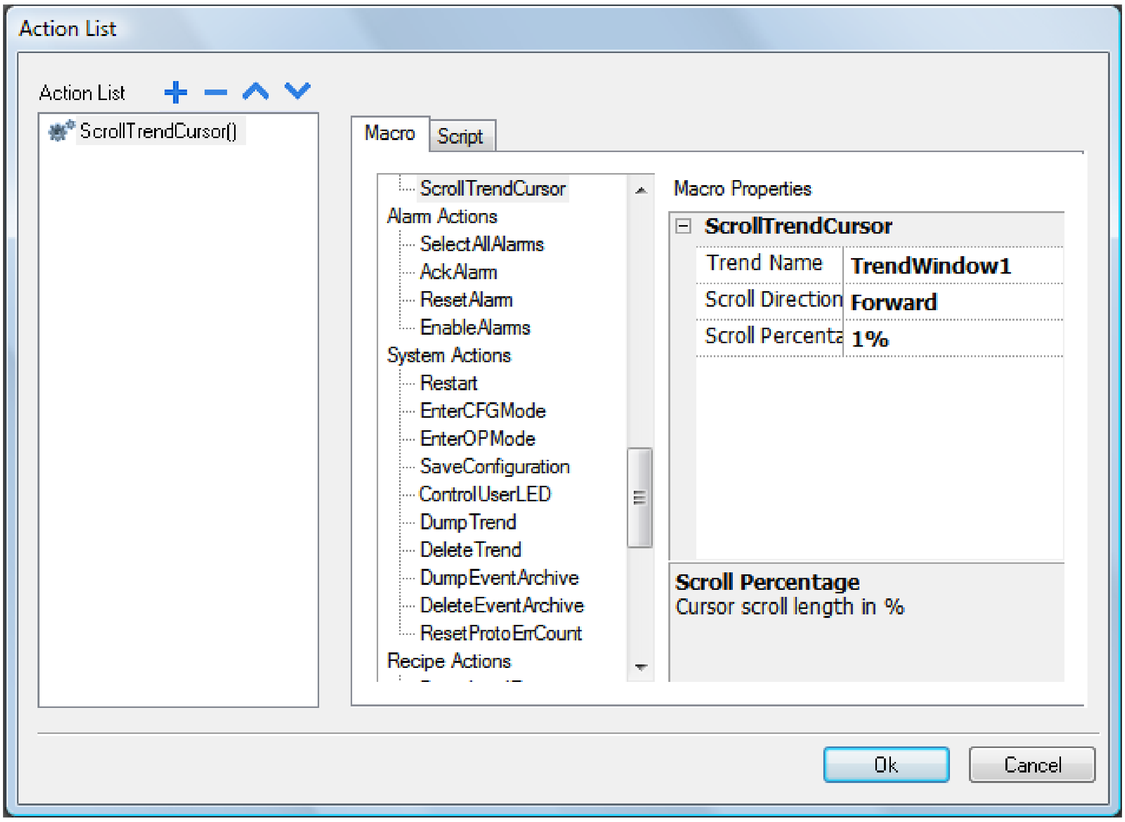
When the graphic cursor is enabled, the scrolling of the trend is stopped.

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.



ScrollTrendCursor

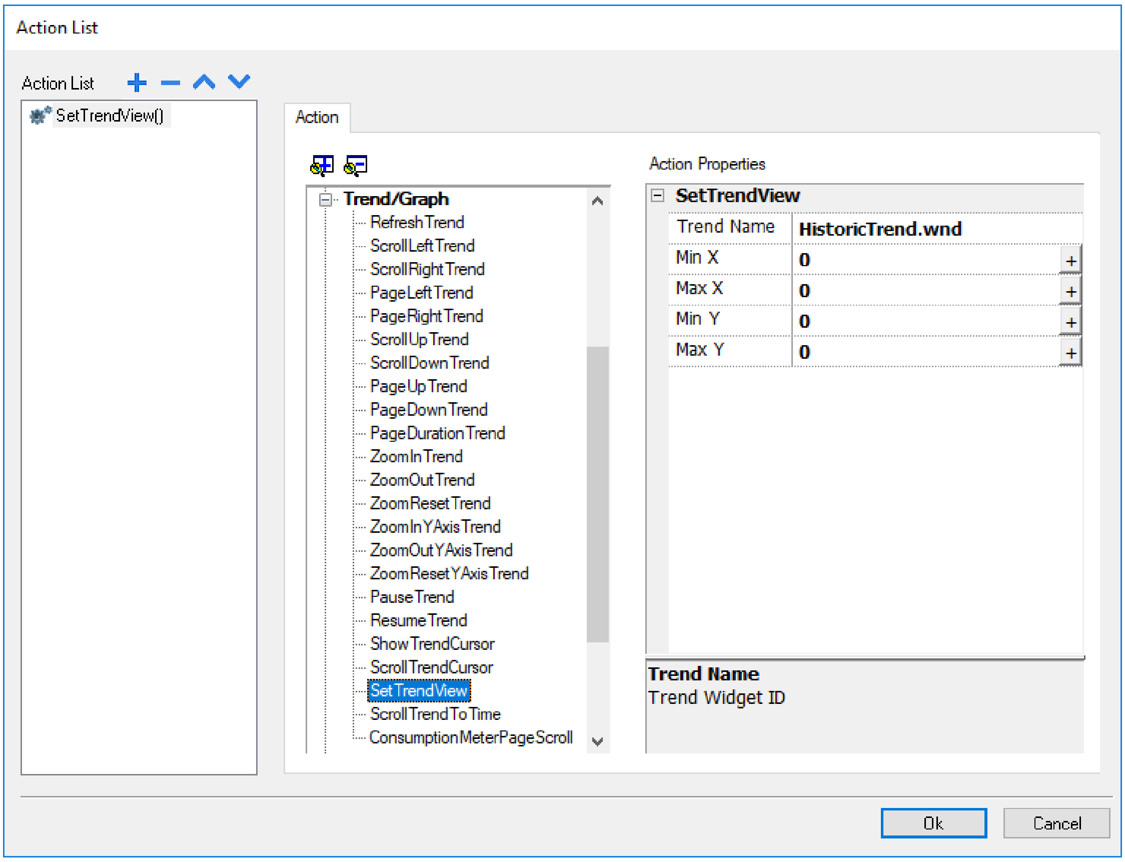
Trend Name	TrendWindow1
Scroll Direction	Forward
Scroll Percenta	1%

Scroll Percentage
Cursor scroll length in %

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.



SetTrendView

Trend Name	HistoricTrend.wnd
Min X	0
Max X	0
Min Y	0
Max Y	0

Trend Name
Trend Widget ID

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.

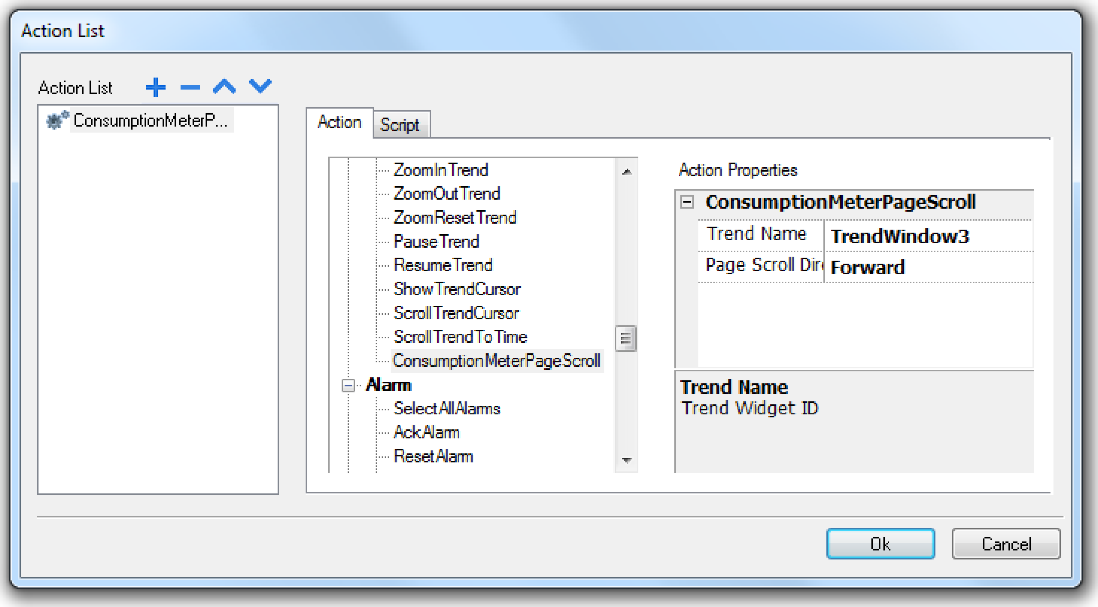
Example

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
Trend Name	Trend widget ID (for example, TrendWindow3)
Page Scroll Direction	Direction of page scrolling (Forward/backward)



ConsumptionMeterPageScroll

Trend Name	TrendWindow3
Page Scroll Dir	Forward

Trend Name
Trend Widget ID

User management actions

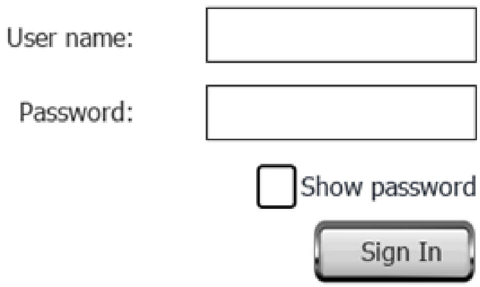
User management and security settings.

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.



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

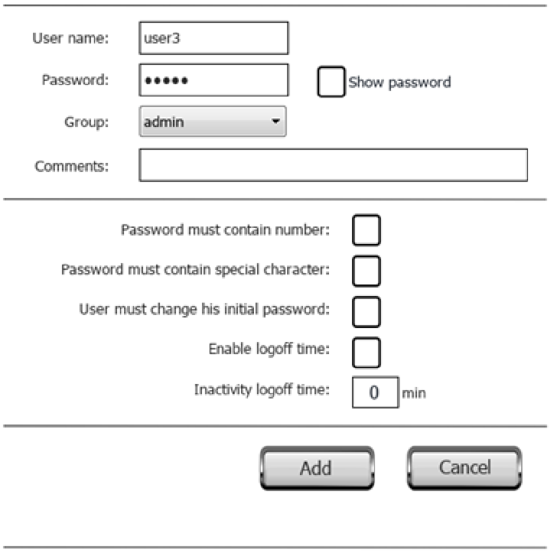
Restores the original password together with the settings specified in the project for the current user.

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: user3

Password: ••••• □ Show password

Group: admin

Comments:

Password must contain number: □

Password must contain special character: □

User must change his initial password: □

Enable logoff time: □

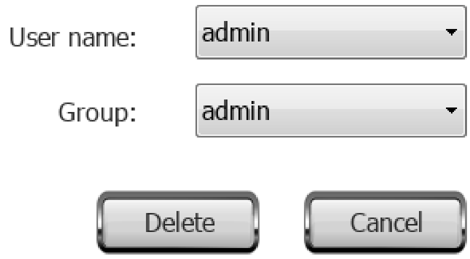
Inactivity logoff time: 0 min

DeleteUser

Reserved to users with ***Can manage other users*** *property set.*

Deletes a user at runtime: a dialog appears.

No parameter is required.



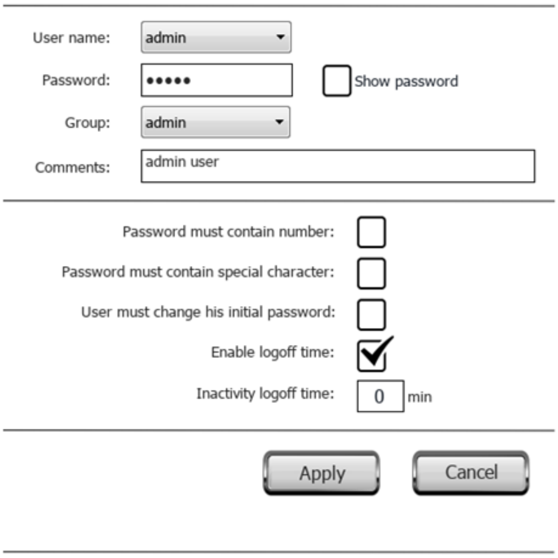
User name: admin

Group: admin

EditUsers

Reserved to users with ***Can manage other users*** *property set.*

Edits user settings.



User name: admin

Password: ●●●●● □ Show password

Group: admin

Comments: admin user

Password must contain number: □

Password must contain special character: □

User must change his initial password: □

Enable logoff time: ☑

Inactivity logoff time: 0 min

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
wgtNameWidget	Widget name
pageName	Active page for data entry. Optional parameter. Useful to select a data field inside a non-modal active dialog box.

TriggerIPCamera

Captures an image from an IP Camera. 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 428 for details. Make sure that the IP Camera supports movement commands.

RefreshEvent

Refreshes the event buffer for **Alarm History** widget. See "Alarms History widget" on page 249 for details.

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 "Project properties" on page 79

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
Media Type	Type of file to update
Device	Device where new media files are supplied
sourcePath	Folder where new media files are stored (for example, "\USBMemory")
Image Resize	Resizes new images to the size of images to be replaced. Not applicable to video files.
Silent	Replaces media automatically. As defau 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);
```

ScrollTable

Scroll rows of the table forward or backward.

Parameter	Description
Table Widget	Table widget name
Direction	The number of rows to jump, forward when positive, backward when negative.

Java Script Interface

```
page.getWidget(TableWgt).scrollTo(Direction);
```

ShiftTableDataSrcColumns

Shift left or right the columns of a data table. Note the remapping is applicated to the data source widget.

Parameter	Description
Data source widget	Data source widget id
Columns Shift	Data source widget columns are shifted (left or right, depeding on sign) by this amount
Fixed left columns	A custom amount of columns (on the left of table) can be kept fixed during shifting
Remap Filter	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
Table Widget	Table to sort
Data Source Column	Column to use to sort the table
Sorting Mode	Can be: Ascending, Descendent or Toggle
Sorting Type	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=Aphabetic, 1=Numeric

var sorting_rule_1 = { _c : column, _m : mode, _t : type };
var json = [ sorting_rule_1 ]
page.getWidget("TableWgt").setSortingRules(json);
```

16 The AGI Client

AGI Client is a standalone application which provides remote access to the HMI Runtime, and is included in the AGI SW Pack. The AGI 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.

AGI 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 AGI Client or to control an HMI device with a PLC. See "Project" on page 86 for details.

To avoid unexpected behavior:

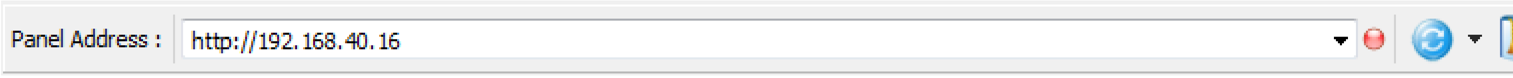
- **be sure to use the same version of the HMI Runtime**
- **use "FreeType Font Rendering" to be sure to use the same font rendering engine on both HMI Client and HMI Device (see "Runtime" on page 80)**

Client application on PC

To run the AGI Client application on PC:

1. From the **Start** menu > **AGI SW Pack** >**AGI 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): AGI 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 **Settings** 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 AGI Client and access remote files via FTP.

See "Remote Client actions" on page 198 and "Remote Client variables" on page 142.

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 AGI Client into the following cache folder.

%appdata%\DEIF A/S\[build number]\client\cache

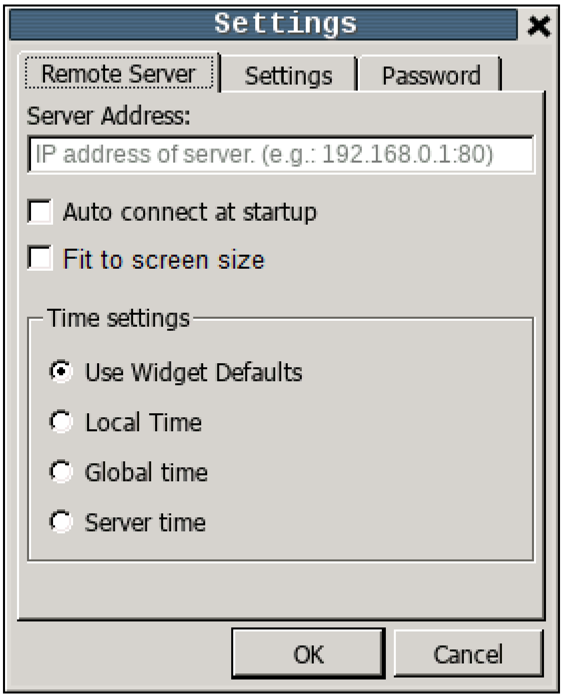
where:

[build number] = folder named as build number, for example 01.90.00.608.

Client application on HMI

To run the AGI 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 101 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.



Settings

Remote Server | Settings | Password

Server Address:

IP address of server. (e.g.: 192.168.0.1:80)

□ Auto connect at startup

□ Fit to screen size

Time settings

● Use Widget Defaults

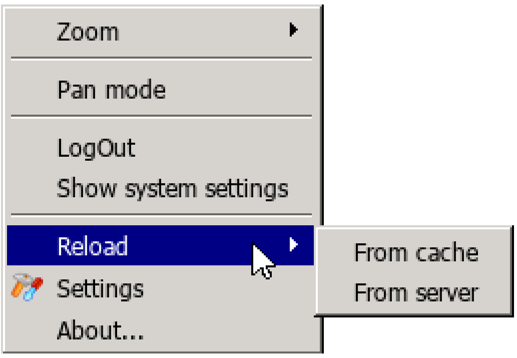
○ Local Time

○ Global time

○ Server time

Context Menu

The Context Menu, available with a right mouse click, will show the below commands:

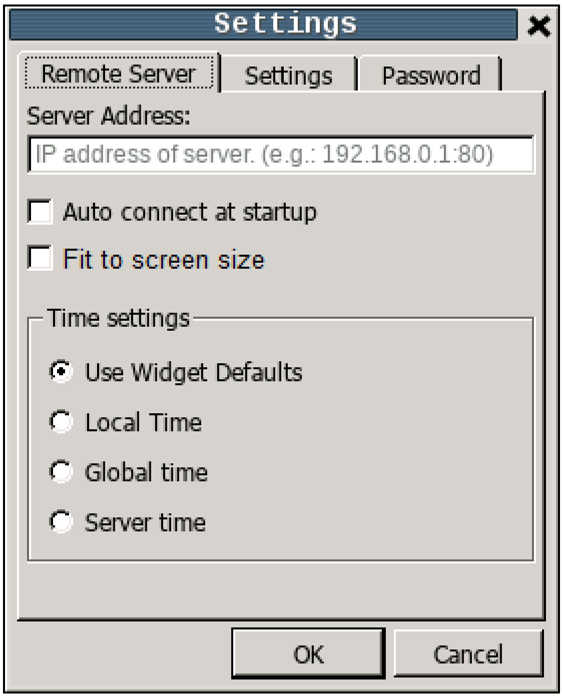


Option	Description
Zoom	Select view size at runtime - Zoom In - Zoom Out - Zoom 100%
Pane Mode	Enables/disables pan mode after a zoom in
Logout	Logs off the current user.
Show system settings	Allow the HMI settings and the management of system components. See "System Settings" on page 547 for details.
Reload	Reload remote project - From cache - From server
Setting	Open the HMI Client Settings. See "Settings and time zone options" below for details ℹ Could be password protected
About	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



Server Address:

IP address of server. (e.g.: 192.168.0.1:80)

☐ Auto connect at startup

☐ Fit to screen size

Time settings

● Use Widget Defaults

○ Local Time

○ Global time

○ Server time

Connection settings

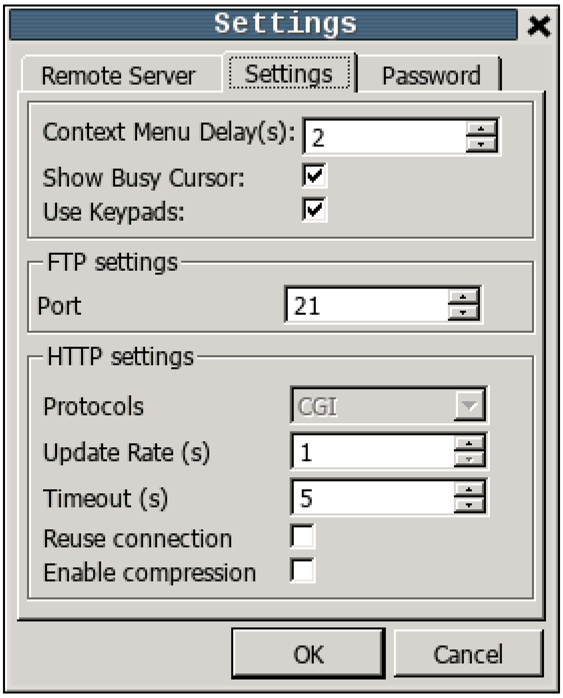
Parameter	Description
Server Address	IP address of the remote HMI device (e.g. 192.168.0.1:80)
Auto connect at startup	When the panel starts, use the Server Address to try to connect automatically to the remote server.
Fit to screen size	Adapts the view to the screen size

Time settings

Parameter	Description
Use Widget Defaults	Displays time information according to the widget settings.
Local Time	Translates all timestamps in the project into the computer local time where the client is installed.
Global Time	Translates all timestamps in the project into UTC format.
Server Time	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



Settings

Remote Server | Settings | Password

Context Menu Delay(s): 2

Show Busy Cursor: ☑

Use Keypads: ☑

FTP settings

Port 21

HTTP settings

Protocols CGI

Update Rate (s) 1

Timeout (s) 5

Reuse connection □

Enable compression □

Interface Settings

Parameter	Description
Context Menu Delay(s)	Context menu activation delay. Range: 1–60 seconds.
Show Busy Cursor	Display an hourglass when the system is busy
Use Keypads	Display keypads when user touches a data entry field. Set to **disable** when an external USB keyboard is connected to the device.

FTP settings

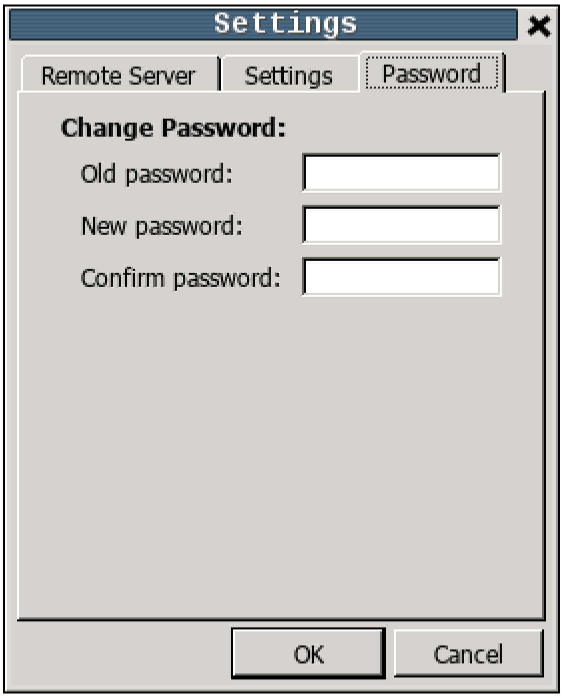
Parameter	Description
Port	FTP communication port

HTTP settings

Parameter	Description
Protocols	Communication protocol used by AGI Client to communicate with an HMI device.
Update Rate	Polling frequency to synchronize data from server. Default = 1 s.
Timeout	Maximum wait time before a request is repeated by the AGI Client. Default = 5 s.
Reuse connection	Enables reuse of the same TCP connection for multiple HTTP requests to reduce network traffic.

Parameter	Description
	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.
Enable compression	Compresses data to reduce download times. Default = disabled. ***CAUTION: enabling this option could causes excessive CPU overhead.***
Time Settings	Used by the client to adapt the widget time stamp information.

Password



Settings

Remote Server | Settings | Password

Change Password:

Old password:

New password:

Confirm password:

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. The FTP server responds on the standard port 21 as default.

Important: The server supports only one connection at a time; if you are using a multiple connection FTP client disable this feature on the client program or set the maximum number of connections per session to 1.

FTP settings

FTP default credentials

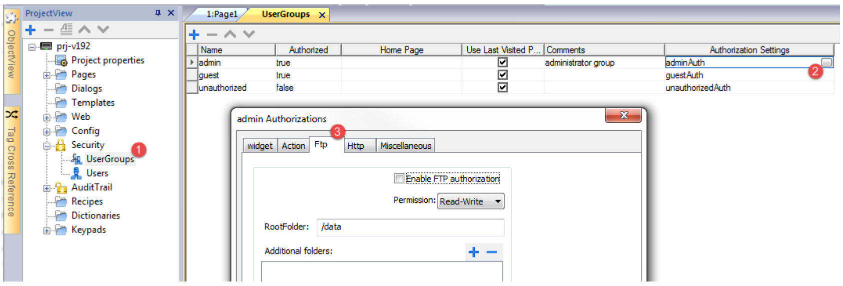
When User Management/Security is disabled use the following credentials for incoming connections:

User name	admin
Password	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.



Name	Authorized	Home Page	Use Last Visited P...	Comments	Authorization Settings
admin	true		☑	administrator group	adminAuth
guest	true		☑		guestAuth
unauthorized	false		☑		unauthorizedAuth

admin Authorizations

☐ Enable FTP authorization

Permission: Read-Write

RootFolder: /data

Additional folders:

See "Configuring groups and authorizations" on page 334 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

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 85.

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.

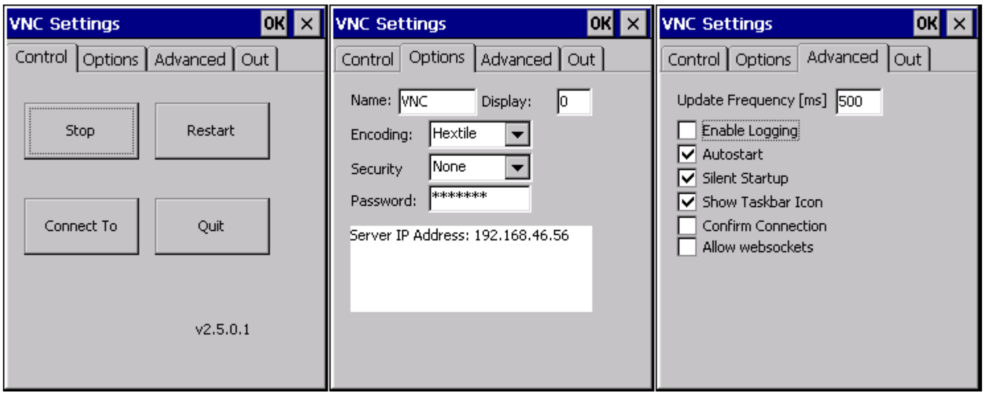
Starting/stopping the VNC server

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.



Name: VNC Display: 0
Encoding: Hextile
Security None
Password: ********
Server IP Address: 192.168.46.56

Update Frequency [ms] 500
☐ Enable Logging
☑ Autostart
☑ Silent Startup
☑ Show Taskbar Icon
☐ Confirm Connection
☐ Allow websockets

VNC should be disabled after use and autostart is not recommended.

Tab	Functions
Control	Star/stop the VNC server and connect to viewer
Options	Define security information for server access using a VNC viewer
Advanced	Enable automatic activation of VNC server at HMI device startup. Select **Silent Startup** to keep the **VNC Options** dialog in the background when **Autostart** is enabled. *Enable **Show Taskbar Icon** when debugging out of KIOSK mode.*
Out	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

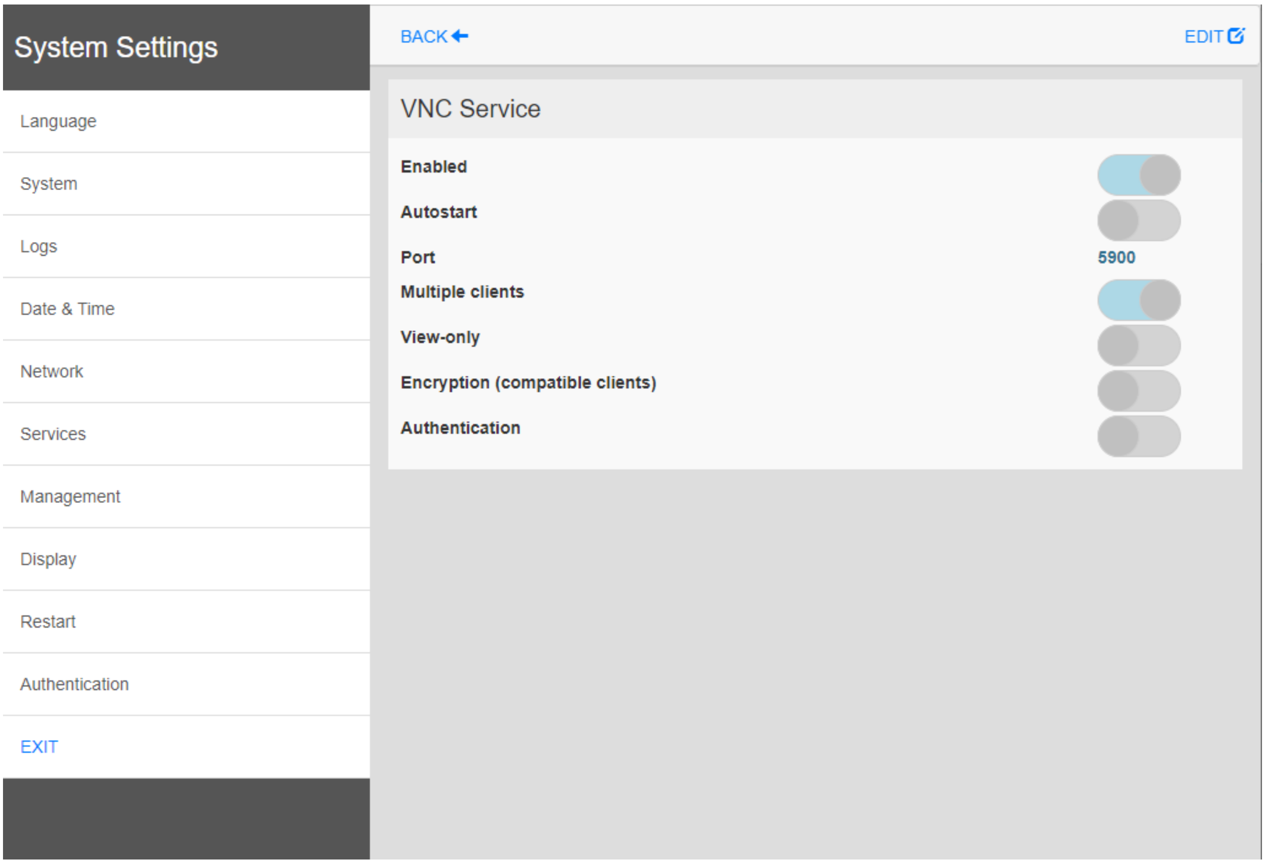
TCP port	5900
Password	null

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 "System Settings" on page 551 for details.

VNC should be disabled after use and autostart is not recommended.



VNC Service

Enabled	☑
Autostart	☐
Port	5900
Multiple clients	☑
View-only	☐
Encryption (compatible clients)	☐
Authentication	☐

Starting VNC viewer

No VNC viewer is provided as part of AGI SW Pack.

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.

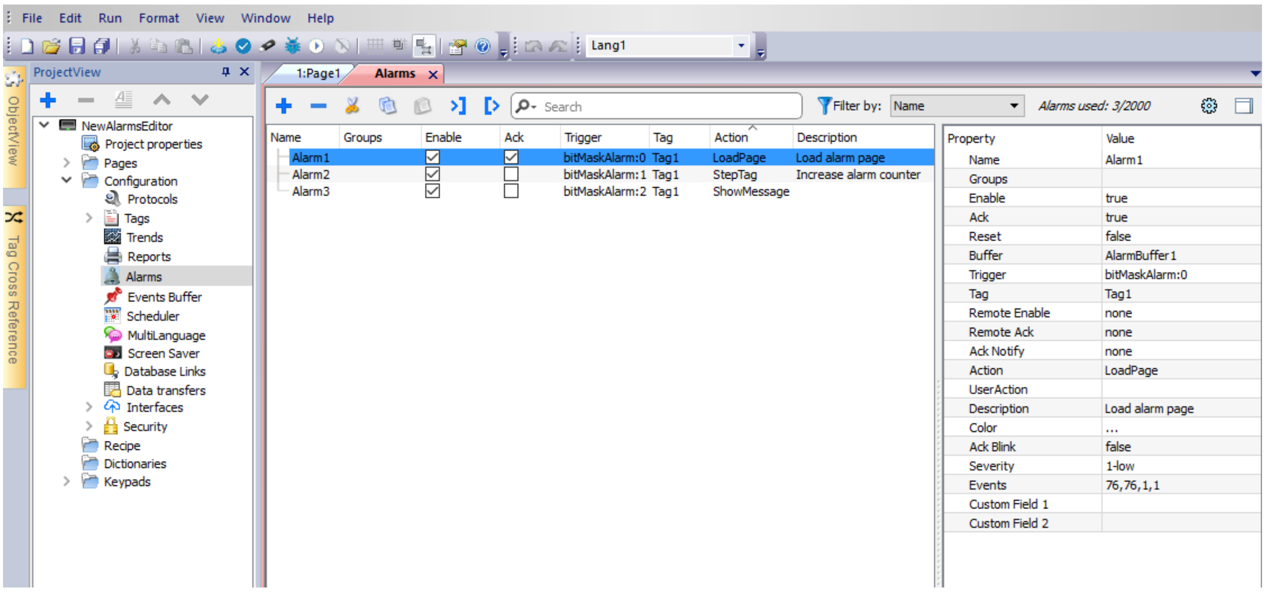
You can define how an alarm is displayed on the HMI device, if it requires user acknowledgment, and if and how it is logged into the event list.

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.

Alarms Editor

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



Alarms used: 3/2000

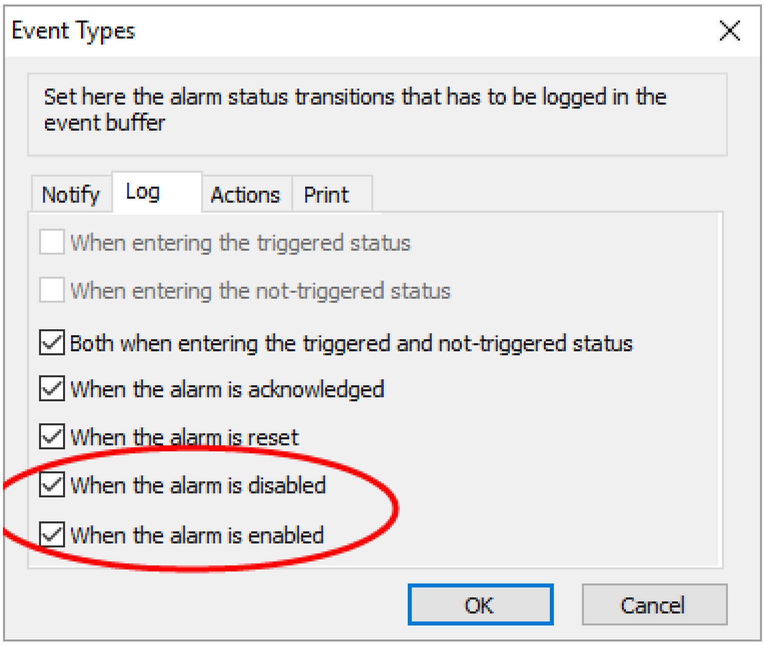
Name	Groups	Enable	Ack	Trigger	Tag	Action	Description
Alarm1		☑	☑	bitMaskAlarm:0	Tag1	LoadPage	Load alarm page
Alarm2		☑	□	bitMaskAlarm:1	Tag1	StepTag	Increase alarm counter
Alarm3		☑	□	bitMaskAlarm:2	Tag1	ShowMessage	

Property	Value
Name	Alarm1
Groups	
Enable	true
Ack	true
Reset	false
Buffer	AlarmBuffer1
Trigger	bitMaskAlarm:0
Tag	Tag1
Remote Enable	none
Remote Ack	none
Ack Notify	none
Action	LoadPage
UserAction	
Description	Load alarm page
Color	...
Ack Blink	false
Severity	1-low
Events	76,76,1,1
Custom Field 1	
Custom Field 2	

Adding an alarm

Click **+** to add an alarm.

Parameter	Description
Name	Name of alarm
Groups	Groups associated with the alarm. They can be used in widgets display filters.
Enable	Enable/disable triggering of alarm. Alarms can be enabled or disabled at runtime as well (see "Enable/disable alarms at runtime" on page 251 for details).
Ack	Enable/disable acknowledgment of alarm, if selected the operator must acknowledge the alarm once triggered to remove it from the **Active Alarm** widget.
Reset	Used with the **Ack** option, if selected, acknowledged alarms stay in the alarm list, labeled as **Not Triggered Acked**, until the operator presses the **Reset** button in the alarm widget.
Buffer	Buffer file where the alarm history will be saved.
Trigger	Triggering condition depending on alarm type: • **limitAlarm**: alarm triggered when tag value exceeds its limits. The alarm is not triggered if the value reaches the limits. • **valueAlarm** alarm is triggered when tag value is equal to the configured value

Parameter	Description
	- **bitMaskAlarm**: the bitwise AND operator compares each bit of the bitmask with the tag 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 ",". - **deviationAlarm**: alarm triggered if the percentage of deviation of the tag value from the set point exceeds a set deviation. $$\lvert Value_{now} - SetPoint \rvert > \left(\frac{deviation}{100} \times SetPoint \right)$$
Tag	Tag whose value will trigger the alarm when it exceeds the set limits. The alarm can refer to the value of this tag, or to the state of a bit if **bitMaskAlarm** has been selected as trigger.
Remote Enable	Tag used by the PLC to enable/disable the alarm. - Changing the enable status from the Alarms Widget will change the tag value - When the tag cannot be read (e.g. communication error) the alarm is disabled - No tags related to the alarm are refreshed when alarm is disabled. Tip: It could be useful to enable the logging of the alarm's enable flag  Event Types Set here the alarm status transitions that has to be logged in the event buffer Notify | Log | Actions | Print ☐ When entering the triggered status ☐ When entering the not-triggered status ☑ Both when entering the triggered and not-triggered status ☑ When the alarm is acknowledged ☑ When the alarm is reset ☑ When the alarm is disabled ☑ When the alarm is enabled OK Cancel
Remote Ack	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. Leave empty if remote acknowledgment is not required. See "Remote alarms acknowledge" on page 241 for details.
Ack Notify	Tag used by the HMI device to notify when the alarm is acknowledged from the device or from the PLC.

Parameter	Description
	0 = set to this value when alarm is triggered 1 = set to this value when alarm is acknowledged.
Action	Actions executed when the alarm is triggered. Additional conditions can be specified in the **Events** column. See "Setting events" on page 242 for details. 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. AGI Web). • loadPage • prevPage • nextPage • showDialog • showMessage • setLanguage • jsAction
User Action	Actions executed when user press the action button in the active alarm widget. See ""Active Alarms widget" on page 244 for details.
Description	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. See "Displaying live alarm data" on page 251 for details.
Custom Field #	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.
Color	Foreground and background colors of alarm rows based on the status of alarm.
AckBlink	Blinking for triggered alarms. If selected the alarm rows blinks until acknowledged. Only effective if **Ack** is selected.
Severity	Severity of the alarm. If multiple alarms are triggered simultaneously, actions will be executed based on severity settings. 0 = not important 1 = low 2 = below normal 3 = normal 4 = above normal 5 = high 6 = critical
Events	Conditions in which the alarms are notified, logged or printed. See "Setting events" on page 242 for details.

Backup alarms events

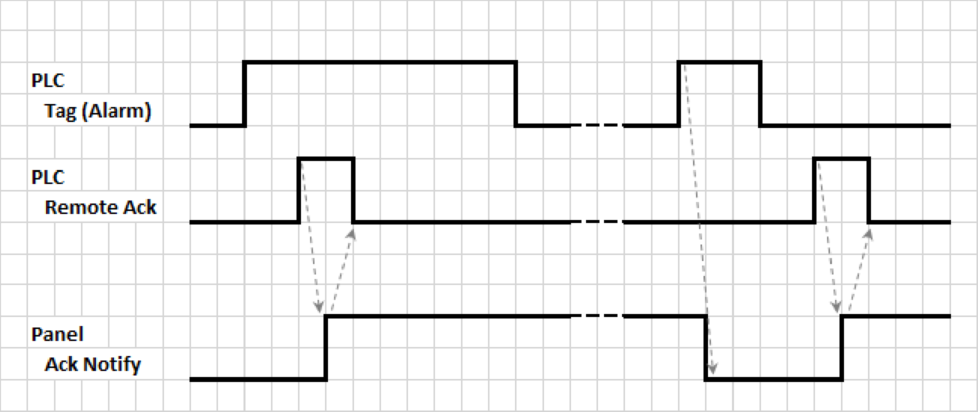
From the "Events Buffer" on page 257 you can configure the size of the alarms buffer and activate the backup of the alarms events when the buffer is full.

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)

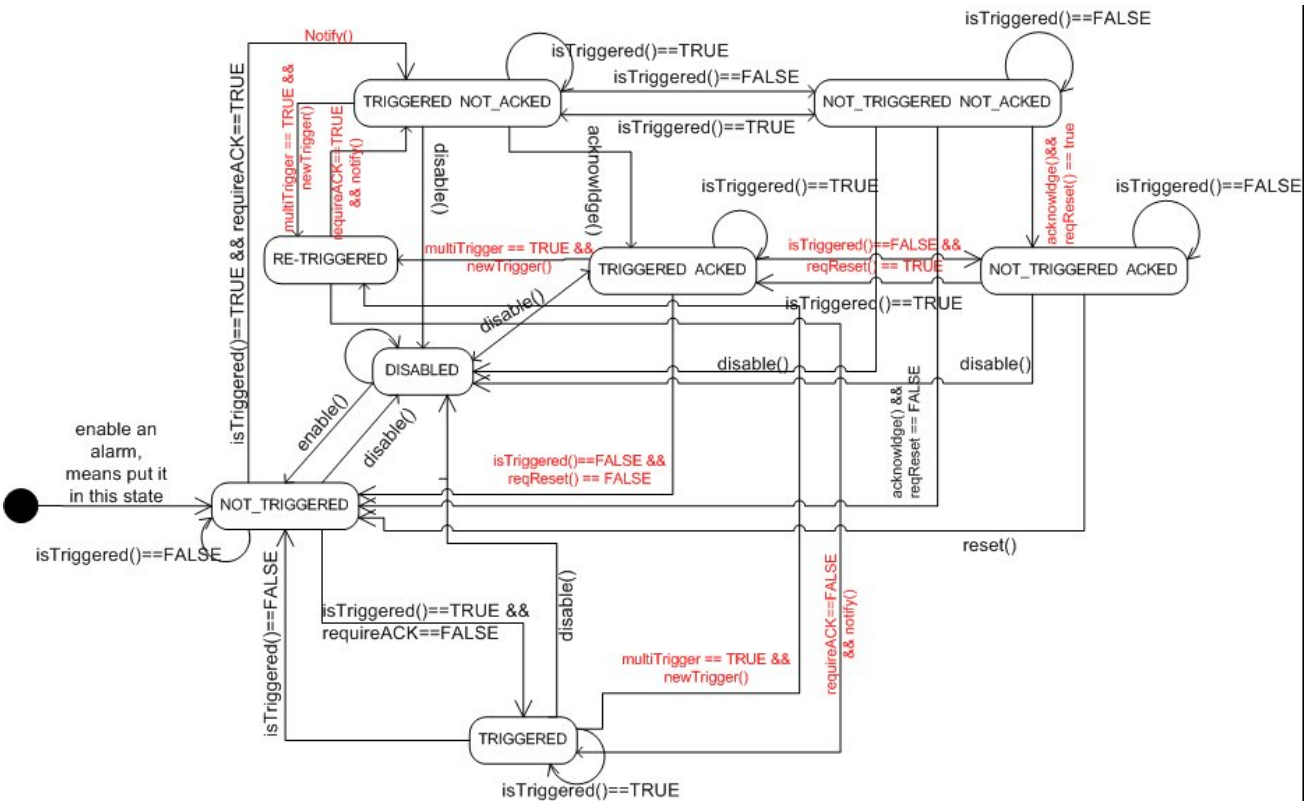
WARNING: When an alarm is triggered, some signals need to be update/communicated through the connected devices. We assume the Acknowledge to be a signal pushed from an operator and not released automatically from a controller device. This allows for time required to communicated the original signals.

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.


isTriggered()==FALSE
Notify()
isTriggered()==TRUE
isTriggered()==FALSE
TRIGGERED_NOT_ACKED
NOT_TRIGGERED_NOT_ACKED
isTriggered()==TRUE
isTriggered() == TRUE && requireACK==TRUE
multiTrigger == TRUE && newTrigger()
requireACK==TRUE && notify()
disable()
acknowldge()
isTriggered()==TRUE
acknowldge()&& reqReset() == true
isTriggered()==FALSE
RE-TRIGGERED
multiTrigger == TRUE && newTrigger()
TRIGGERED_ACKED
isTriggered()==FALSE && reqReset() == TRUE
NOT_TRIGGERED_ACKED
isTriggered()==TRUE
disable()
DISABLED
disable()
disable()
enable an alarm, means put it in this state
enable()
disable()
isTriggered()==FALSE && reqReset() == FALSE
acknowldge() && reqReset == FALSE
NOT_TRIGGERED
reset()
isTriggered()==FALSE
isTriggered()==FALSE
isTriggered()==TRUE && requireACK==FALSE
disable()
requireACK==FALSE && notify()
multiTrigger == TRUE && newTrigger()
TRIGGERED
isTriggered()==TRUE

Setting events

Path: **ProjectView**> **Config** > **Alarms** > **Events** property

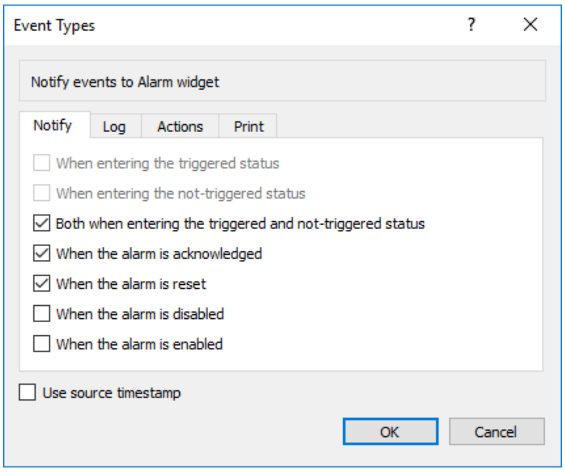
Events are defined using the Alarms Editor.

See "Alarms Editor" on page 238 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.



Event Types

Notify events to Alarm widget

- ☐ When entering the triggered status
- ☐ When entering the not-triggered status
- ☑ Both when entering the triggered and not-triggered status
- ☑ When the alarm is acknowledged
- ☑ When the alarm is reset
- ☐ When the alarm is disabled
- ☐ When the alarm is enabled

☐ Use source timestamp

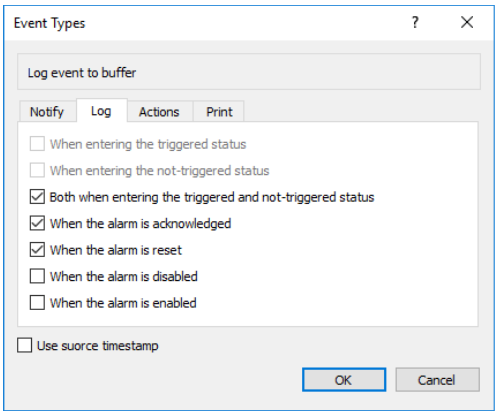
Here you define the behavior of the default alarm widget available in the Widget gallery and decide in which cases the widget is updated by a change in an alarm status.

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.



Event Types

Log event to buffer

Notify | Log | Actions | Print

□ When entering the triggered status
□ When entering the not-triggered status
☑ Both when entering the triggered and not-triggered status
☑ When the alarm is acknowledged
☑ When the alarm is reset
□ When the alarm is disabled
□ When the alarm is enabled

□ Use suorce timestamp

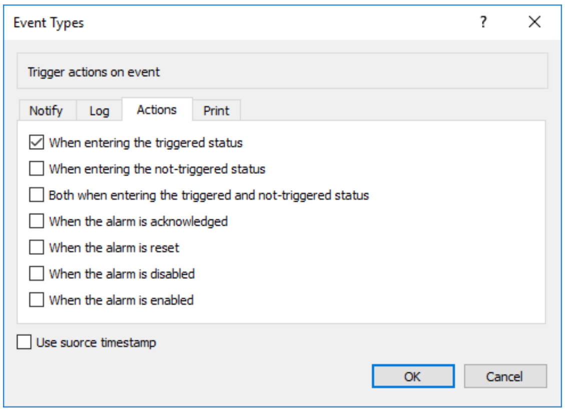
OK | Cancel

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.



Event Types

Trigger actions on event

Notify | Log | Actions | Print

☑ When entering the triggered status
□ When entering the not-triggered status
□ Both when entering the triggered and not-triggered status
□ When the alarm is acknowledged
□ When the alarm is reset
□ When the alarm is disabled
□ When the alarm is enabled

□ Use suorce timestamp

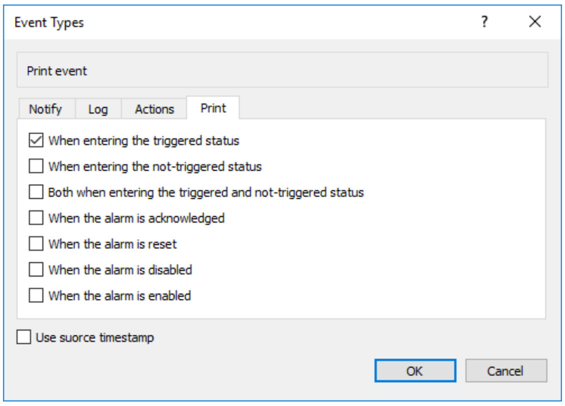
OK | Cancel

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



Event Types ? ×

Print event

☑ When entering the triggered status

□ When entering the not-triggered status

□ Both when entering the triggered and not-triggered status

□ When the alarm is acknowledged

□ When the alarm is reset

□ When the alarm is disabled

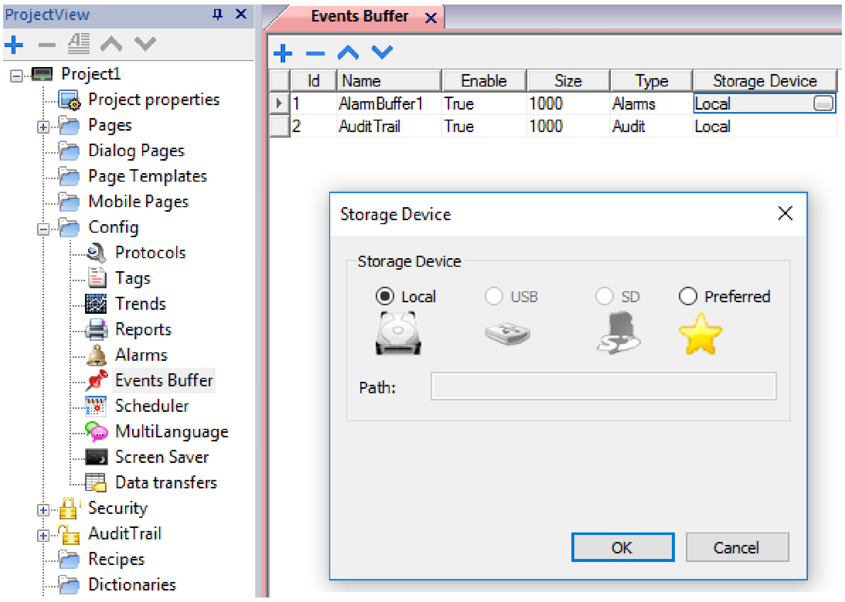
□ When the alarm is enabled

□ Use suorce timestamp

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.



Id	Name	Enable	Size	Type	Storage Device
1	AlarmBuffer1	True	1000	Alarms	Local
2	AuditTrail	True	1000	Audit	Local

Storage Device

Storage Device

● Local ○ USB ○ SD ○ Preferred

Path:

Data is automatically saved every five minutes except for alarm data which is saved immediately.

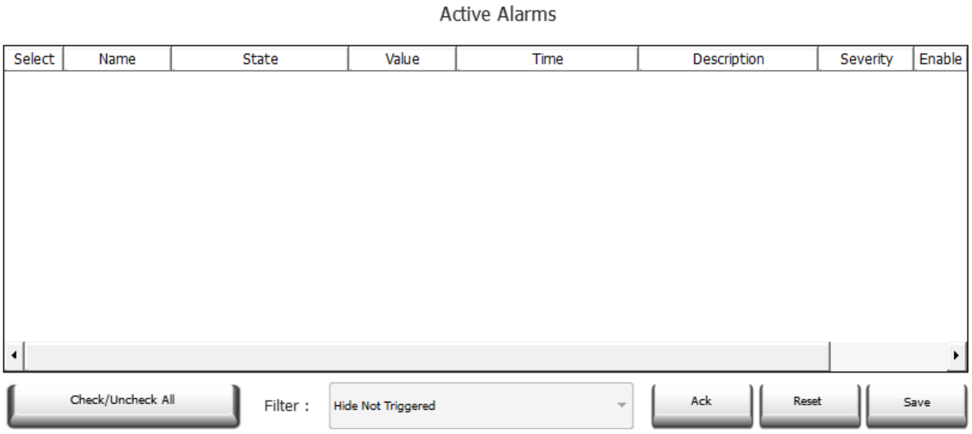
Use source timestamp

Events are stores with the timestamp of when the HMI device detect 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)

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.



Filter : Hide Not Triggered

Alarm filters

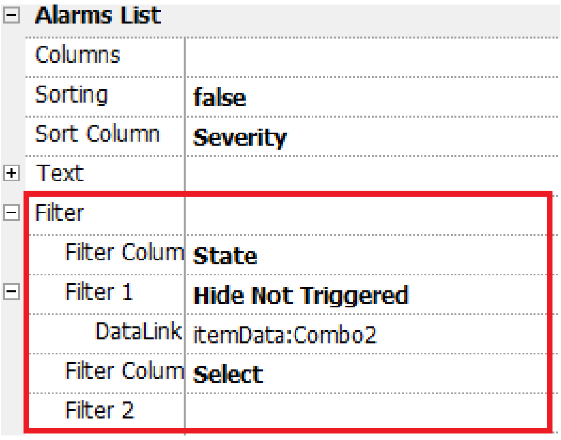
*Path: **ActiveAlarm** widget> **Properties** pane> **Filter***

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 Colum	State
Filter 1	Hide Not Triggered
DataLink	itemData:Combo2
Filter Colum	Select
Filter 2	

Setting filters

*Path: **ActiveAlarm** widget> **Properties** pane> **Filter***

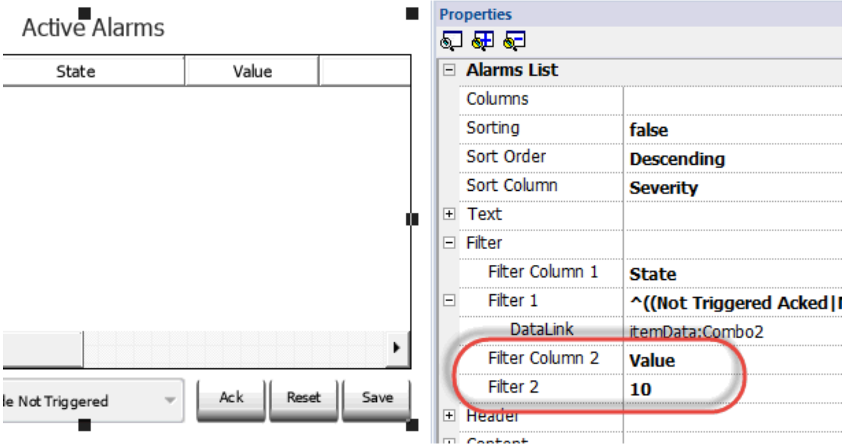
To set one of the two available filters:

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 for additional details regarding regular expressions.

Filters first example

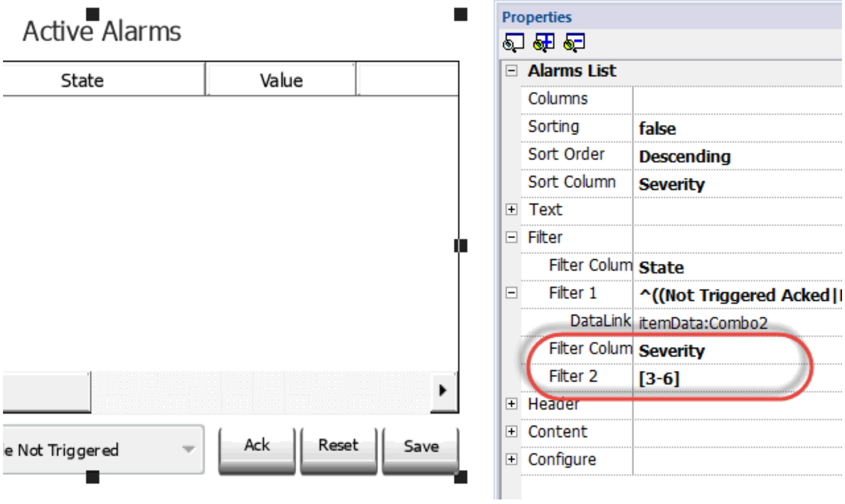
You want to show all alarms matching Filter 1 with value equal to 10. Then properties settings: **Filter column 2** = Value, **Filter 2** = 10



Alarms List	
Columns	
Sorting	false
Sort Order	Descending
Sort Column	Severity
Text	
Filter	
Filter Column 1	State
Filter 1	^((Not Triggered Acked|I
DataLink	itemData:Combo2
Filter Column 2	Value
Filter 2	10
Header	

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]



Alarms List	
Columns	
Sorting	false
Sort Order	Descending
Sort Column	Severity
Text	
Filter	
Filter Colum	State
Filter 1	^((Not Triggered Acked|I
DataLink	itemData:Combo2
Filter Colum	Severity
Filter 2	[3-6]
Header	
Content	
Configure	

Filters third example

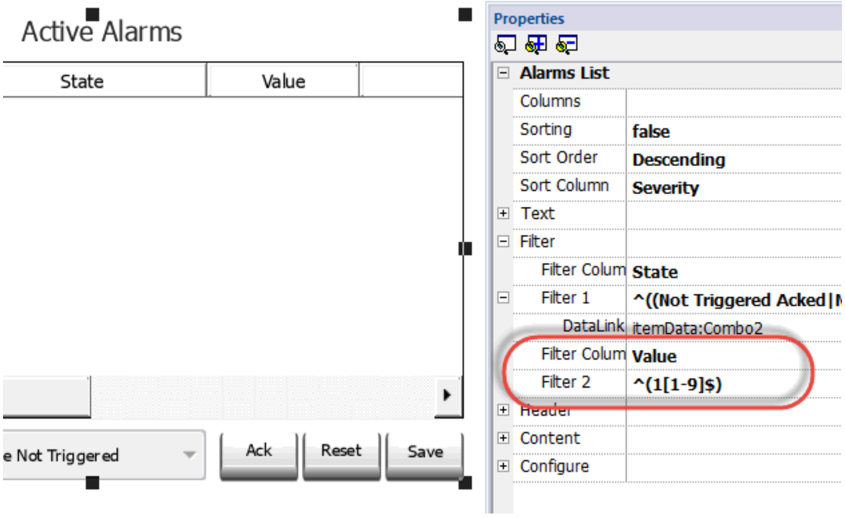
You want to show all alarms matching a value from 11 to 19. Then properties settings: **Filter column 2** = Severity, **Filter 2** = ^(1[1-9]$)

Meaning:

^ = match must starts from the beginning of the string

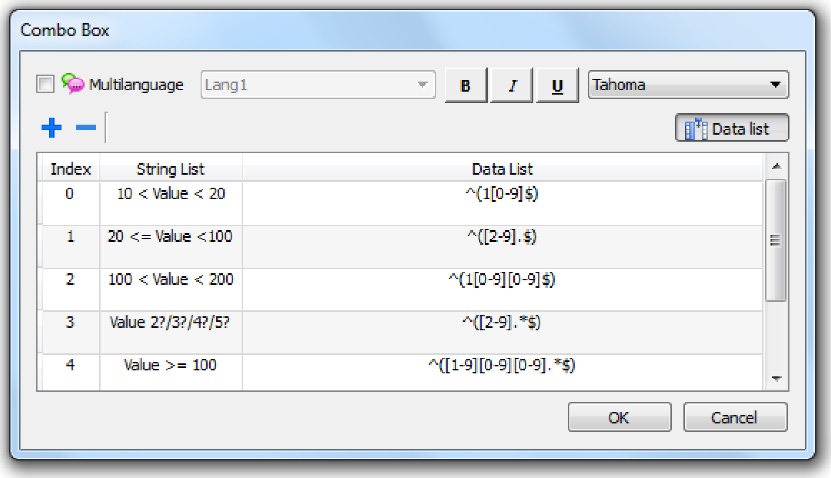
1[1-9] = first char must be 1 and the second char must be between 1 and 9

$ = end of the comparison.



Alarms List	
Columns	
Sorting	false
Sort Order	Descending
Sort Column	Severity
Text	
Filter	
Filter Colum	State
Filter 1	^((Not Triggered Acked|N
DataLink	itemData:Combo2
Filter Colum	Value
Filter 2	^(1[1-9]$)
Header	
Content	
Configure	

Filters expression examples



Index	String List	Data List
0	10 < Value < 20	^(1[0-9]$)
1	20 <= Value <100	^([2-9].$)
2	100 < Value < 200	^(1[0-9][0-9]$)
3	Value 2?/3?/4?/5?	^([2-9].*$)
4	Value >= 100	^([1-9][0-9][0-9].*$)

Filter by	String list	Data list
State	Hide Not Triggered	^((Not Triggered Acked|Not Triggered Not Acked|Triggered).*$)
Value	10 < Value < 20	^(1[0-9]$)
Value	20 <= Value <100	^([2-9].$)
Value	100 < Value < 200	^(1[0-9][0-9]$)
Value	Value 2?/3?/4?/5?	^([2-9].*$)
Value	Value >= 100	^([1-9][0-9][0-9].*$)
Value	Value >= 20	^([2-9].*$\|[1-9][0-9][0-9].*$)

Sorting alarms

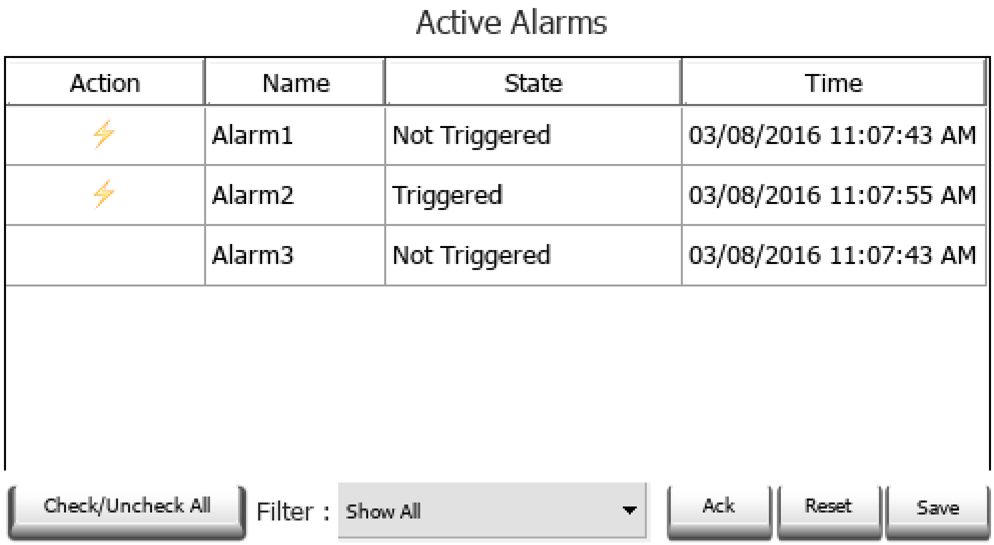
*Path: **ActiveAlarm** widget> **Properties** pane> **Sorting***

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 238 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

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 Colums property of the Active Alarms widget

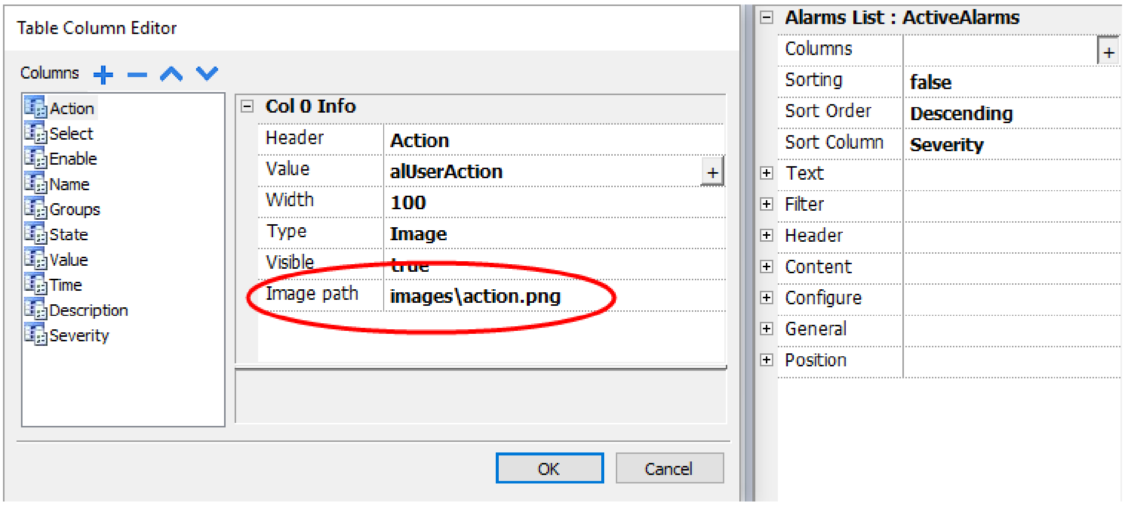


Table Column Editor

Columns

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

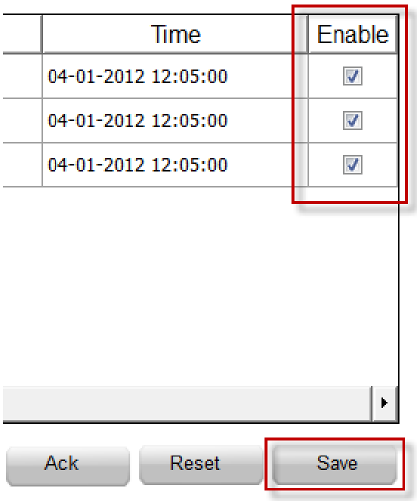
Alarms List : ActiveAlarms

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.

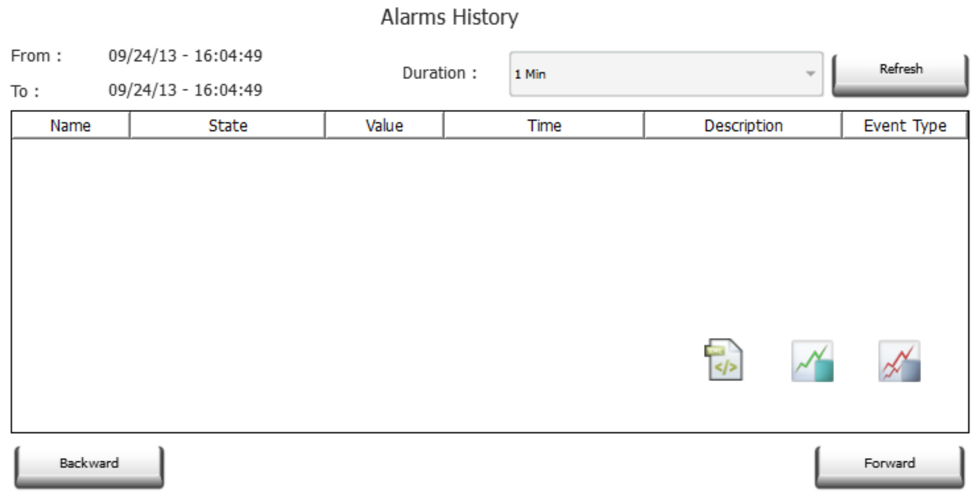
Saves changes made in the **Enable** column in the alarm widget. This action is used with the **Save** button in the alarm widget.



Time	Enable
04-01-2012 12:05:00	☑
04-01-2012 12:05:00	☑
04-01-2012 12:05:00	☑

Alarms History widget

Logs and display an alarm list if **Buffer** property in Alarms Configuration Editor is set.



Alarms History

From : 09/24/13 - 16:04:49

To : 09/24/13 - 16:04:49

Duration : 1 Min

Name	State	Value	Time	Description	Event Type

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

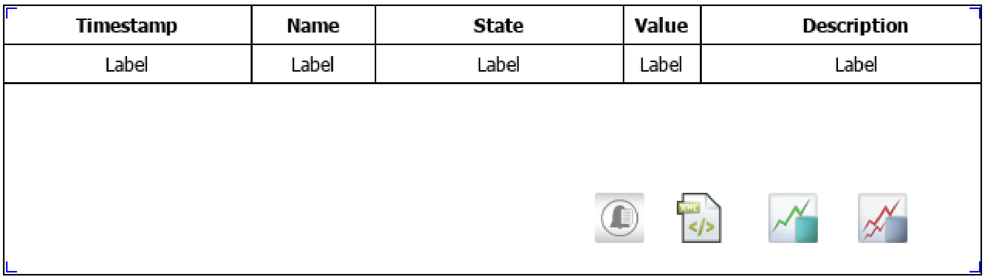
In addition to the two main “Active Alarms” and the “Alarms History” widgets, the Gallery contains some other alarms widgets with a slightly different look but basically similarly at the two main widgets. You are free to choose and use the widget that has the look that better meet your requirements.

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 431 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.

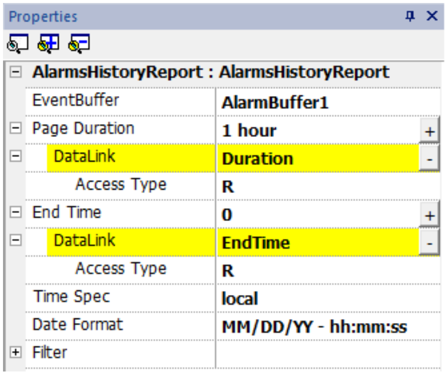


Timestamp	Name	State	Value	Description
Label	Label	Label	Label	Label

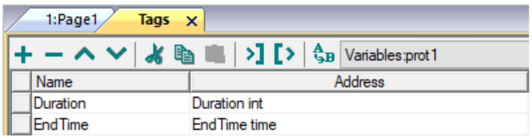
Using the “attach to tag” feature is possible to use tags to define some properties of the historical alarms list to print at runtime:

- Page Duration
- End Time

"Page Duration" with "End Time" define the piece of the alarm buffer to print.



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	



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 244 for details.

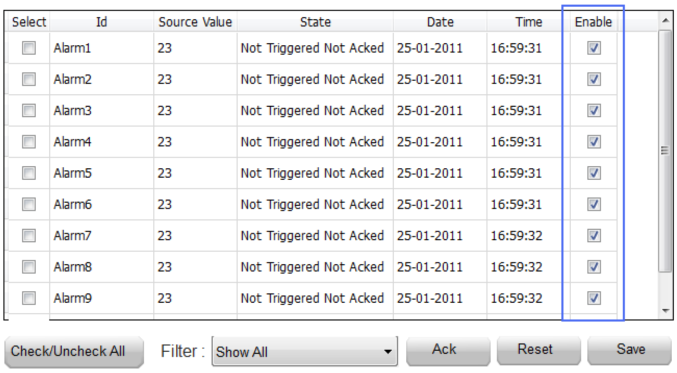
IMPORTANT: The Active Alarms widget is not displayed automatically. You must add a dedicated action that will open the page containing the alarm widget when the alarm is triggered.

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.



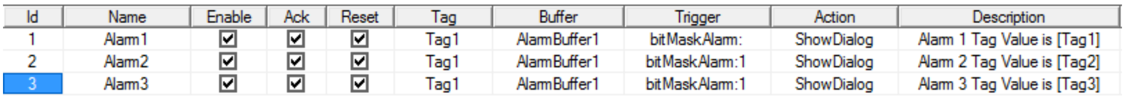
Select	Id	Source Value	State	Date	Time	Enable
□	Alarm1	23	Not Triggered Not Acked	25-01-2011	16:59:31	☑
□	Alarm2	23	Not Triggered Not Acked	25-01-2011	16:59:31	☑
□	Alarm3	23	Not Triggered Not Acked	25-01-2011	16:59:31	☑
□	Alarm4	23	Not Triggered Not Acked	25-01-2011	16:59:31	☑
□	Alarm5	23	Not Triggered Not Acked	25-01-2011	16:59:31	☑
□	Alarm6	23	Not Triggered Not Acked	25-01-2011	16:59:31	☑
□	Alarm7	23	Not Triggered Not Acked	25-01-2011	16:59:32	☑
□	Alarm8	23	Not Triggered Not Acked	25-01-2011	16:59:32	☑
□	Alarm9	23	Not Triggered Not Acked	25-01-2011	16:59:32	☑

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	Alarm1	☑	☑	☑	Tag1	AlarmBuffer1	bitMaskAlarm:	ShowDialog	Alarm 1 Tag Value is [Tag1]
2	Alarm2	☑	☑	☑	Tag1	AlarmBuffer1	bitMaskAlarm:1	ShowDialog	Alarm 2 Tag Value is [Tag2]
3	Alarm3	☑	☑	☑	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 *"[Tag[1]]"*.

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: [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 34)

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

Example:

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

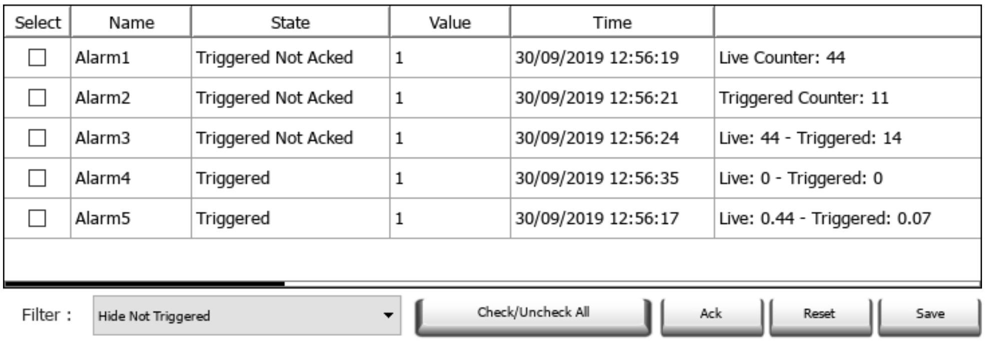
Note that by default, all tags are displayed as an integer. If you want to display a float number, you have to specify how to show the number adding the decimal digits.

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	
□	Alarm1	Triggered Not Acked	1	30/09/2019 12:56:19	Live Counter: 44
□	Alarm2	Triggered Not Acked	1	30/09/2019 12:56:21	Triggered Counter: 11
□	Alarm3	Triggered Not Acked	1	30/09/2019 12:56:24	Live: 44 - Triggered: 14
□	Alarm4	Triggered	1	30/09/2019 12:56:35	Live: 0 - Triggered: 0
□	Alarm5	Triggered	1	30/09/2019 12:56:17	Live: 0.44 - Triggered: 0.07

Filter : Hide Not Triggered

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)

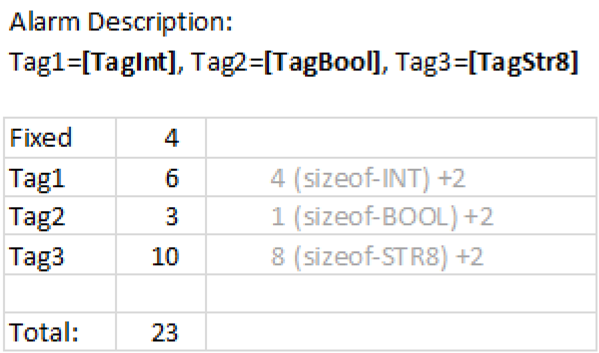
Length limit of the Description field

Number of live tags that can be used inside each alarm's description depends on size of used tags. AGI Creator 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 bytes + (size of tag + 2) + (size of tag + 2) + (size of tag + 2) + . . .
 <= 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
Total:	23	

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(size-INT) x 8(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(size-INT) x 1(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

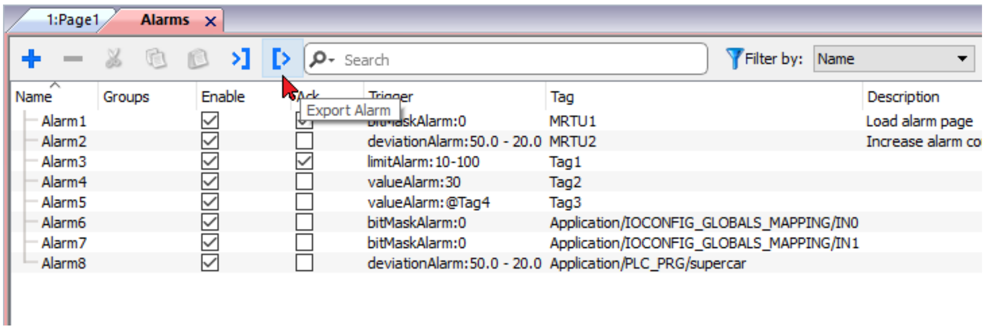
To export an event buffer containing an history alarms list, use the **DumpEventArchive** action.

See "System actions" on page 199 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

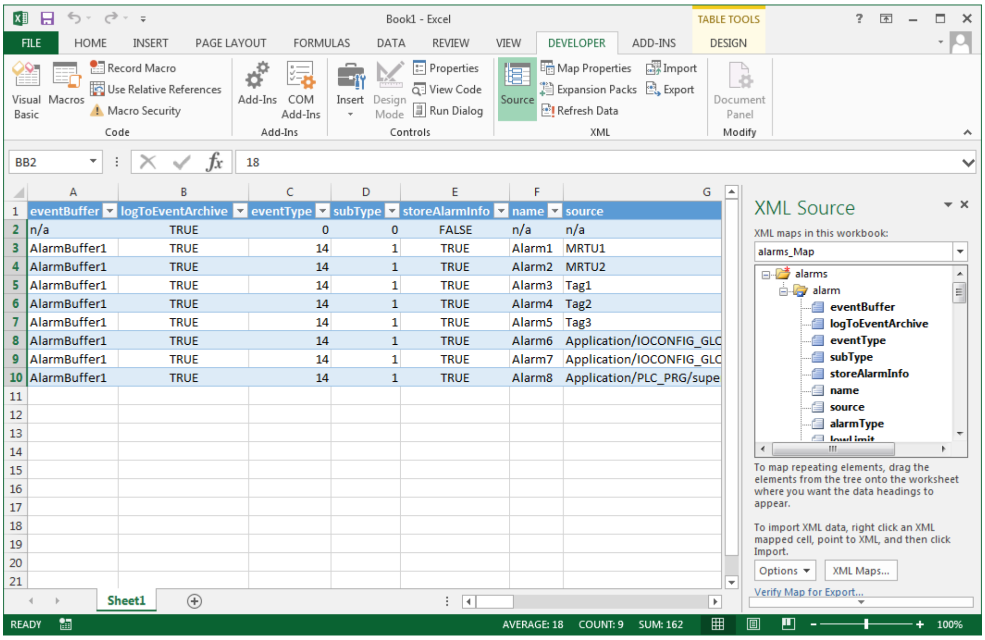
*Path: **ProjectView**> **Config** > double-click **Alarms***



Name	Groups	Enable	Ack	Trigger	Tag	Description
Alarm1		☑	☑	bitMaskAlarm:0	MRTU1	Load alarm page
Alarm2		☑	□	deviationAlarm:50.0 - 20.0	MRTU2	Increase alarm co
Alarm3		☑	☑	limitAlarm:10-100	Tag1	
Alarm4		☑	□	valueAlarm:30	Tag2	
Alarm5		☑	□	valueAlarm:@Tag4	Tag3	
Alarm6		☑	□	bitMaskAlarm:0	Application/IOCONFIG_GLOBALS_MAPPING/IN0	
Alarm7		☑	□	bitMaskAlarm:0	Application/IOCONFIG_GLOBALS_MAPPING/IN1	
Alarm8		☑	□	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) .

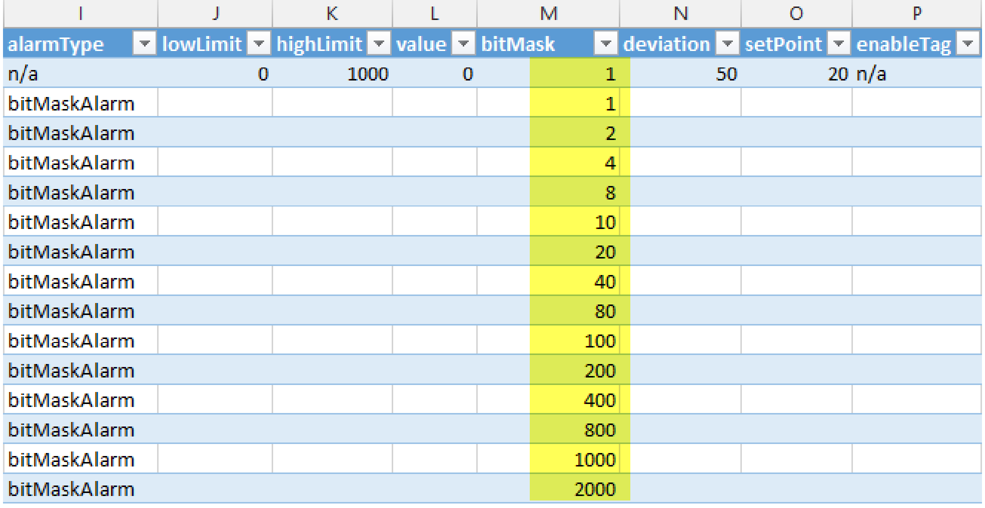


	A	B	C	D	E	F	G
1	eventBuffer	logToEventArchive	eventType	subType	storeAlarmInfo	name	source
2	n/a	TRUE	0	0	FALSE	n/a	n/a
3	AlarmBuffer1	TRUE	14	1	TRUE	Alarm1	MRTU1
4	AlarmBuffer1	TRUE	14	1	TRUE	Alarm2	MRTU2
5	AlarmBuffer1	TRUE	14	1	TRUE	Alarm3	Tag1
6	AlarmBuffer1	TRUE	14	1	TRUE	Alarm4	Tag2
7	AlarmBuffer1	TRUE	14	1	TRUE	Alarm5	Tag3
8	AlarmBuffer1	TRUE	14	1	TRUE	Alarm6	Application/IOCONFIG_GLC
9	AlarmBuffer1	TRUE	14	1	TRUE	Alarm7	Application/IOCONFIG_GLC
10	AlarmBuffer1	TRUE	14	1	TRUE	Alarm8	Application/PLC_PRG/supe

To map repeating elements, drag the elements from the tree onto the worksheet where you want the data headings to appear.

To import XML data, right click an XML mapped cell, point to XML, and then click Import.

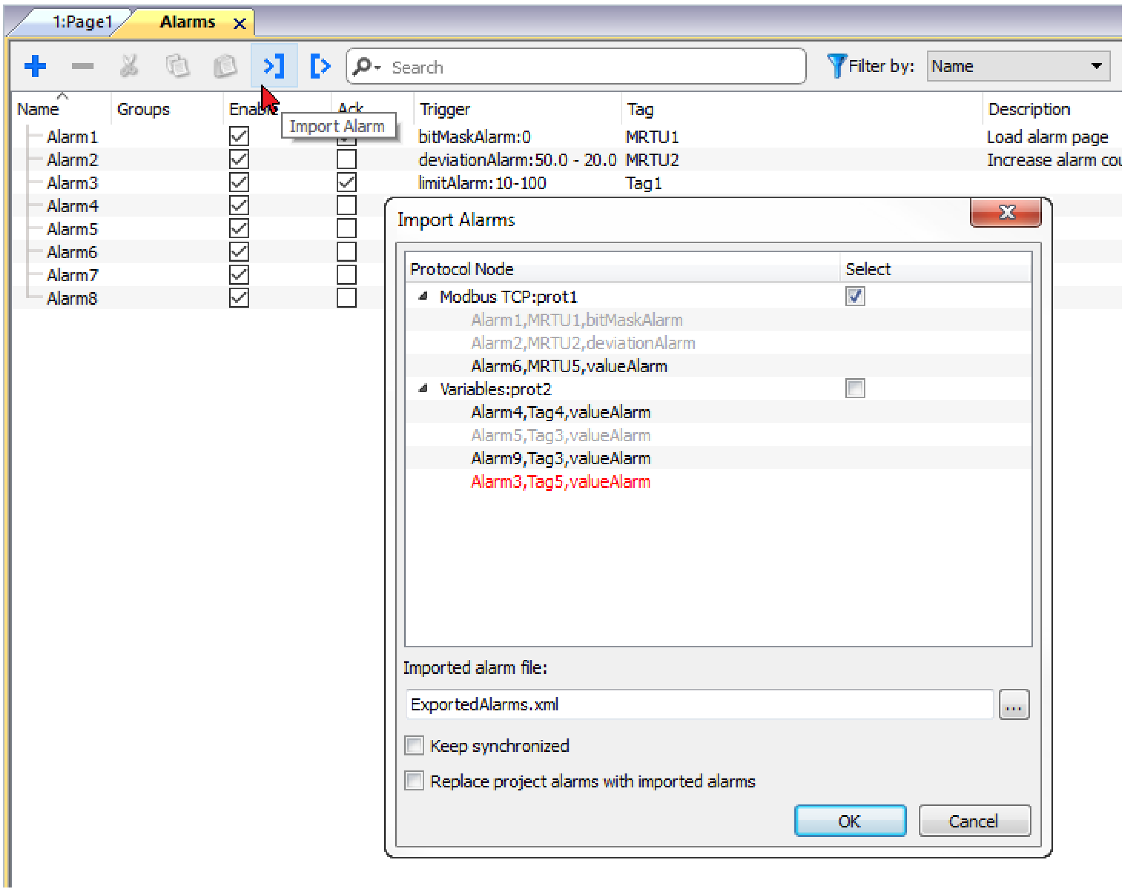
Warning: the bitMask values are reported as 2^BitPosition in Hexadecimal format.



I	J	K	L	M	N	O	P
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***



Name	Groups	Enable	Ack	Trigger	Tag	Description
Alarm1		☑		bitMaskAlarm:0	MRTU1	Load alarm page
Alarm2		☑	□	deviationAlarm:50.0 - 20.0	MRTU2	Increase alarm cou
Alarm3		☑	☑	limitAlarm:10-100	Tag1	
Alarm4		☑	□			
Alarm5		☑	□			
Alarm6		☑	□			
Alarm7		☑	□			
Alarm8		☑	□			

Import Alarms

Protocol Node	Select
Modbus TCP:prot1	☑
Alarm1,MRTU1,bitMaskAlarm	
Alarm2,MRTU2,deviationAlarm	
Alarm6,MRTU5,valueAlarm	
Variables:prot2	□
Alarm4,Tag4,valueAlarm	
Alarm5,Tag3,valueAlarm	
Alarm9,Tag3,valueAlarm	
Alarm3,Tag5,valueAlarm	

Imported alarm file: ExportedAlarms.xml

□ Keep synchronized

□ Replace project alarms with imported alarms

OK Cancel

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

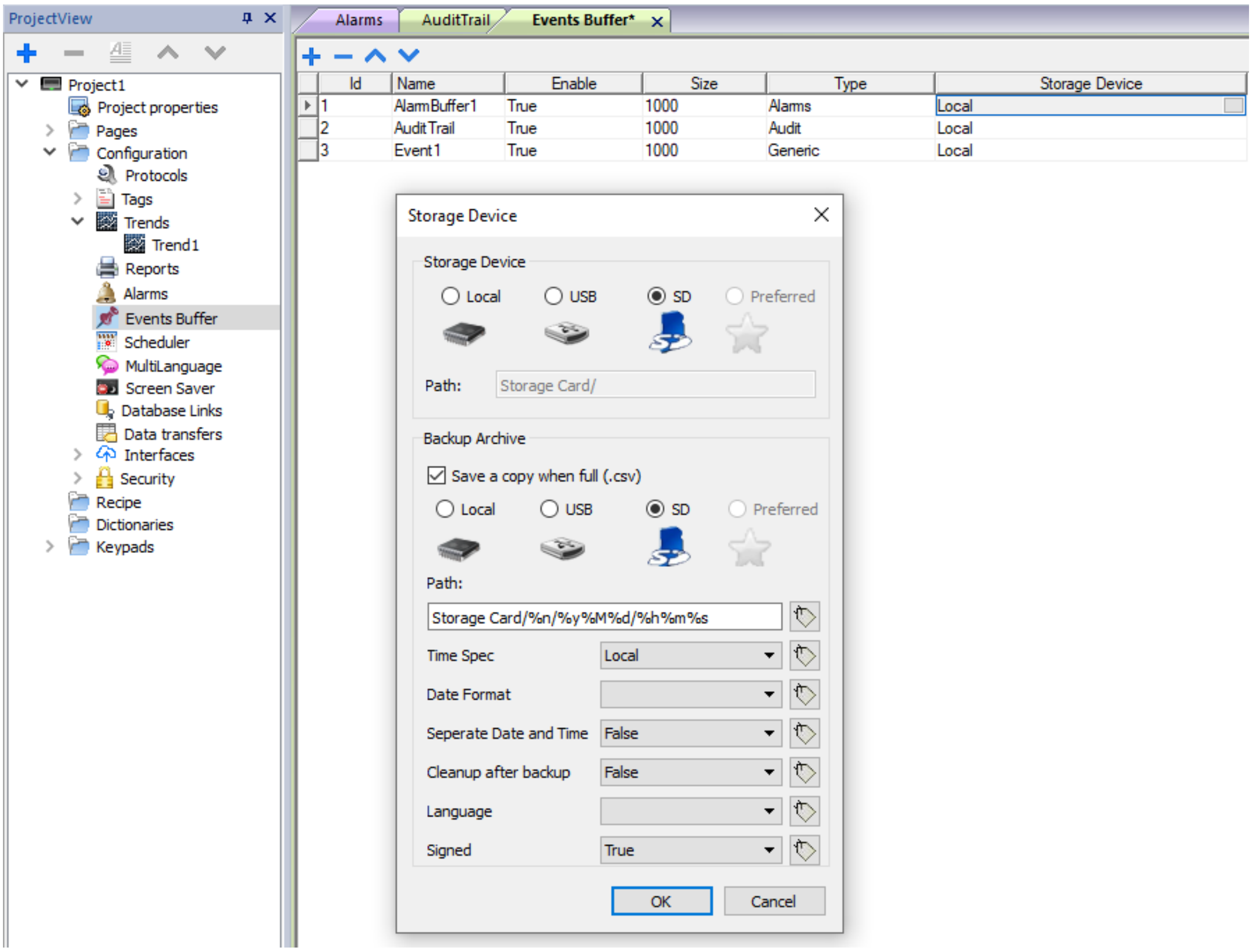
Select the **Keep synchronized** option in the **Import Alarms** dialog to enable the automatic synchronization of the alarm configuration file.

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 AGI Creator.

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.



Id	Name	Enable	Size	Type	Storage Device
1	AlarmBuffer1	True	1000	Alarms	Local
2	Audit Trail	True	1000	Audit	Local
3	Event1	True	1000	Generic	Local

Storage Device

Storage Device

○ Local ○ USB ● SD ○ Preferred

Path: Storage Card/

Backup Archive

☑ Save a copy when full (.csv)

○ Local ○ USB ● SD ○ Preferred

Path: Storage Card/%n/%y%M%d/%h%m%s

Time Spec	Local
Date Format	
Seperate Date and Time	False
Cleanup after backup	False
Language	
Signed	True

Parameter	Description
Id	Buffer identification number
Name	Buffer name
Enable	Enable/disable logging
Size	Size of log file. Data is automatically saved to disk every 5 minutes.
Type	Type of events logged: • **Alarms** • **Audit** • **Generic**
Storage Device	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
Path	Where events buffer data will be copied. The below wild cards are supported • %n = Events buffer name • %y = Year • %M = Month • %d = Day • %h = Hour • %m = Minutes • %s = Seconds
Time Spec	Timestamp of events • Local Use the time of the HMI device where the project is running • Global Use global time (GMT)
Date Format	Time and Date format. Placeholders can be used (see "Time and Date placeholders" on page 413)
Separate Date and Time	When "true", the date and the time are placed into two different fields
Cleanup after backup	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)
Language	Language to use
Signed	When "true", the additional file with the signature is added (see "Signed CSV files" on page 327)

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 AGI Creator 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 194 for details on how to reset recipe data.

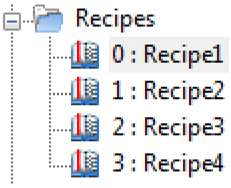
Note: Recipe data can be stored on a Flash memory, on a USB drive or on a SD card.

Managing recipes

Creating a recipe

To create a recipe for your project:

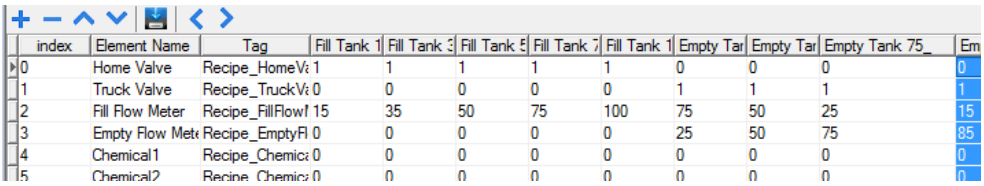
1. In **ProjectView** right-click **Recipes** and select **Insert Recipe**: an empty recipe is added. You create and configure recipes using the Recipe Editor.



Recipes
0 : Recipe1
1 : Recipe2
2 : Recipe3
3 : Recipe4

Recipe editor

*Path: **ProjectView**> **Recipes** > double-click **RecipeName***

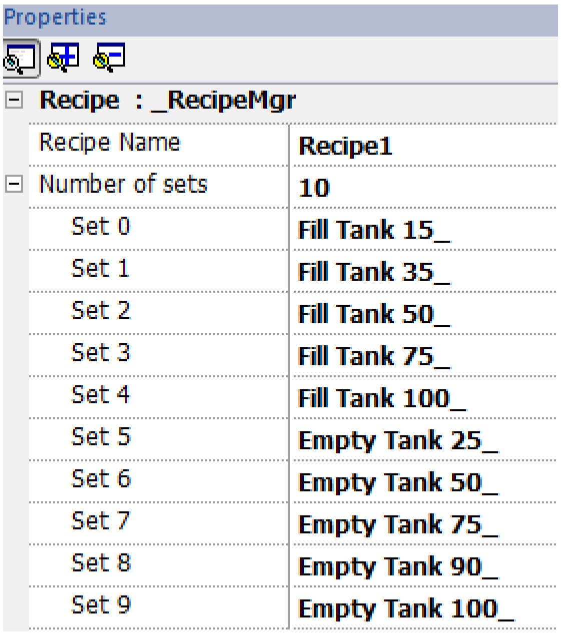


index	Element Name	Tag	Fill Tank 1	Fill Tank 3	Fill Tank 5	Fill Tank 7	Fill Tank 1	Empty Tar	Empty Tar	Empty Tank 75_	Em
0	Home Valve	Recipe_HomeVa	1	1	1	1	1	0	0	0	0
1	Truck Valve	Recipe_TruckVa	0	0	0	0	0	1	1	1	1
2	Fill Flow Meter	Recipe_FillFlowM	15	35	50	75	100	75	50	25	15
3	Empty Flow Mete	Recipe_EmptyFl	0	0	0	0	0	25	50	75	85
4	Chemical1	Recipe_Chemica	0	0	0	0	0	0	0	0	0
5	Chemical2	Recipe_Chemica	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
Recipe Name	Name of the recipe
Number of sets	Number of values sets for each recipe element. Each set has a different configurable name.



Properties

Recipe : _RecipeMgr

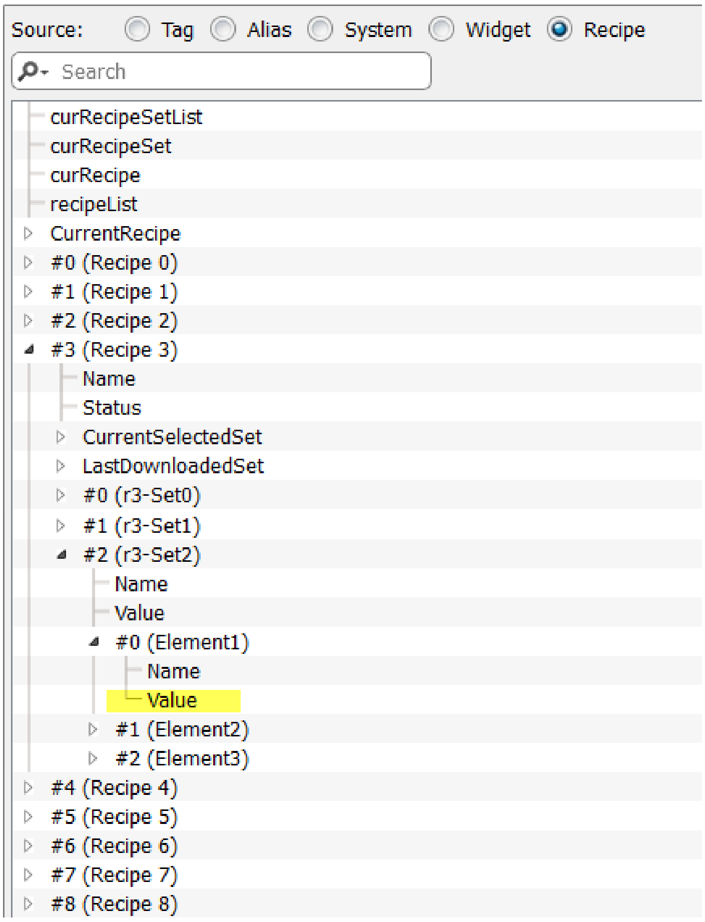
Recipe Name	**Recipe1**
Number of sets	**10**
Set 0	**Fill Tank 15_**
Set 1	**Fill Tank 35_**
Set 2	**Fill Tank 50_**
Set 3	**Fill Tank 75_**
Set 4	**Fill Tank 100_**
Set 5	**Empty Tank 25_**
Set 6	**Empty Tank 50_**
Set 7	**Empty Tank 75_**
Set 8	**Empty Tank 90_**
Set 9	**Empty Tank 100_**

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:

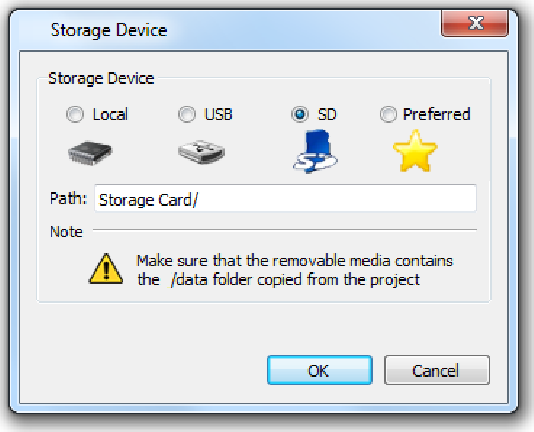
- Current Recipe >Current Selected Recipe Set> Element > Value
- Selected Recipe > Selected Set0 > Element > Value
- recipeList

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.



Storage Device

Storage Device

○ Local ○ USB ● SD ○ Preferred

Path: Storage Card/

Note

Make sure that the removable media contains the /data folder copied from the project

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:

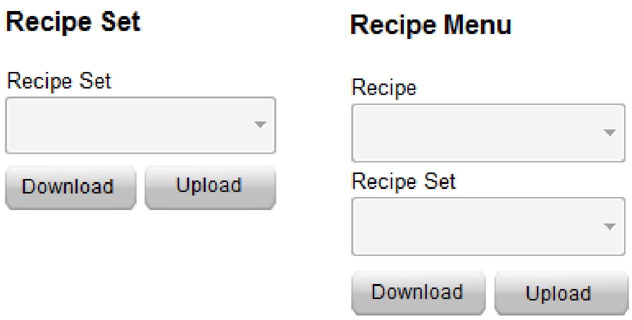
- Comma Separated Values (.csv)
- Unicode Text (.txt)

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 264
- **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

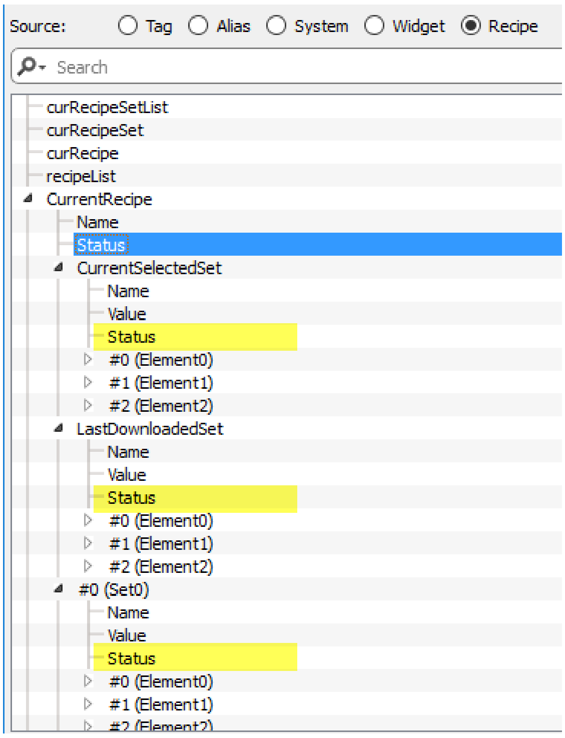
In the **Properties** pane of each **Recipe Set** widget set the following parameter:

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



Source: ○ Tag ○ Alias ○ System ○ Widget ● Recipe

Search

```
curRecipeSetList
curRecipeSet
curRecipe
recipeList
CurrentRecipe
    Name
    Status
    CurrentSelectedSet
        Name
        Value
        Status
        #0 (Element0)
        #1 (Element1)
        #2 (Element2)
    LastDownloadedSet
        Name
        Value
        Status
        #0 (Element0)
        #1 (Element1)
        #2 (Element2)
    #0 (Set0)
        Name
        Value
        Status
        #0 (Element0)
        #1 (Element1)
        #2 (Element2)
```

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	Syncronized	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 45 for details)
- configure the action in an alarm action list (see "Alarm actions" on page 178 for details)
- configure the action in a scheduler action list (see "Scheduling events at runtime" on page 316 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 194

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 194 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.

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 65536 files; FAT max 513 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.

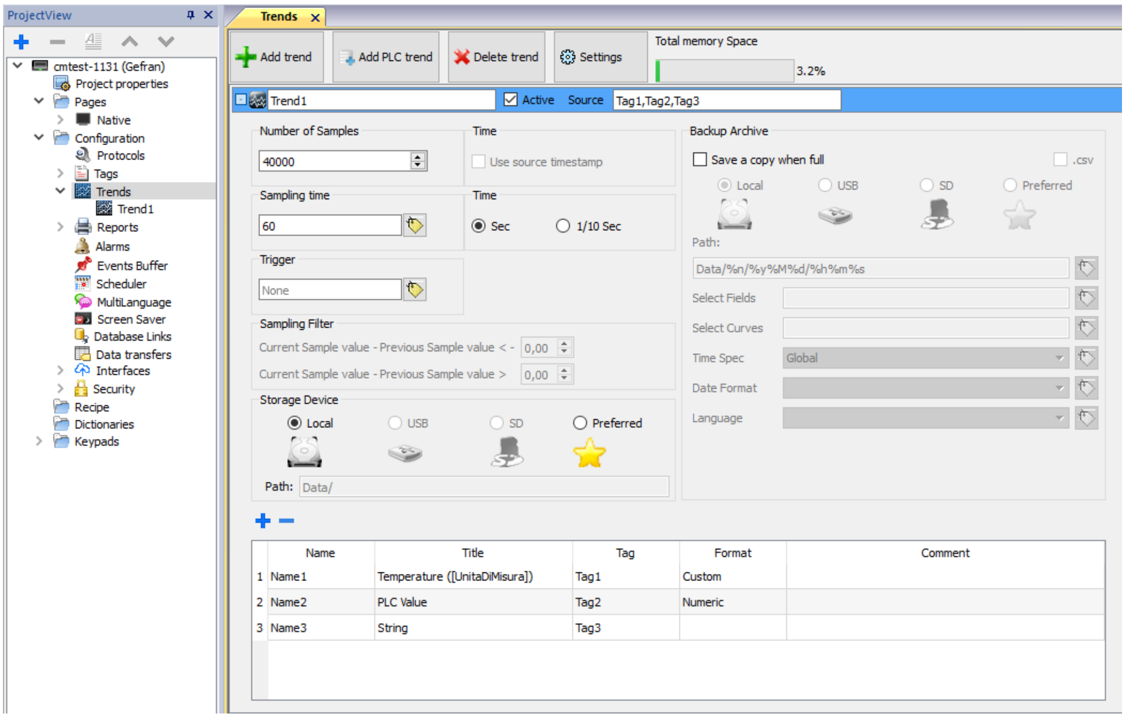
WARNING: If the data/time is moved back, the samples with invalid date/time are removed from the trend buffer. When system detects that data/time is invalid (e.g. battery low), a popup is shown to advise the user and the date/time of the last sample is used to avoid losing data.

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.



Total memory Space 3.2%

Trend1 ☑ Active Source Tag1,Tag2,Tag3

Number of Samples: 40000

Time: □ Use source timestamp

Sampling time: 60

Time: ● Sec ○ 1/10 Sec

Trigger: None

Sampling Filter
Current Sample value - Previous Sample value < - 0,00
Current Sample value - Previous Sample value > 0,00

Storage Device: ● Local ○ USB ○ SD ○ Preferred

Path: Data/

Backup Archive: □ Save a copy when full □ .csv

○ Local ○ USB ○ SD ○ Preferred

Path: Data/%n/%y%M%d/%h%m%s

Select Fields

Select Curves

Time Spec: Global

Date Format

Language

	Name	Title	Tag	Format	Comment
1	Name1	Temperature ([UnitaDiMisura])	Tag1	Custom	
2	Name2	PLC Value	Tag2	Numeric	
3	Name3	String	Tag3		

Toolbar Element	Description
Add trend	Add a trend that will be sampled from the HMI device.
Add PLC trend	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. The parameters depend on the used device, refer to the manual of the selected device.
Delete trend	Remove the select trend.
Settings	Offers the ability to customize the labels that appear in the trend dump header and trend table widgets. • Timestamp • Date • Time • Quality The below placeholders can be used: • **" \n "** (space + \n + space) can be used to split the label into two or more lines • **[TagName]** (tag name enclosed in square brackets) can be used to display a tag value
Total memory Space	Memory used by the defined trend buffers.

Trend Header	Description
Trend Name	Name of the trend collection (set of tags sampled at the same time)
Active	When enabled, the trend runs by default at system startup. ℹ Note: Trends cannot be activated at runtime.
Source	List of the tags sampled by the trend.

Trend Element	Description
Number of Samples	Trend buffer size (see "Number of Samples" on page 272 for additional information)
Sampling Time	Sampling interval. ℹ 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.
Time	Time unit for the sample time. Could be 1 second (default) or 1/10 seconds ℹ Be aware that increasing the sampling rate could impact global HMI device performances.

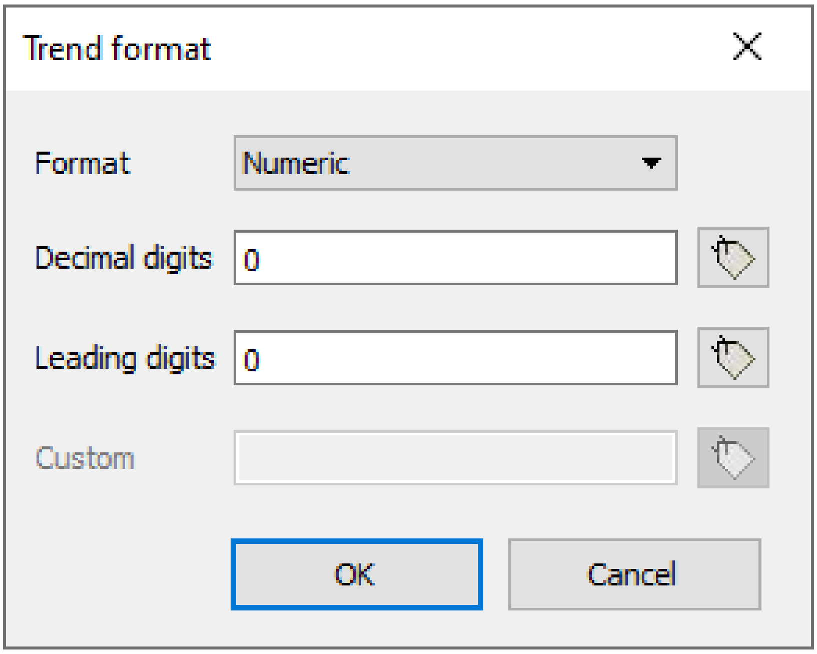
<table>
<tr><th>Trend Element</th><th>Description</th></tr>
<tr><td><b>Timestamp</b></td><td>When checked, samples are stored using the timestamp provided from the remote device.
Available only:
• for device’s protocols that support this feature (OPC UA Client)
• when trend buffer is configured to with a single tag</td></tr>
<tr><td><b>Trigger</b></td><td>Tag triggering the sample.
If used, when the value of this tag changes, a sample is collected.
ℹ Note: Trigger and Source can refer to the same tag.</td></tr>
<tr><td><b>Storage Device</b></td><td>Where trend buffer data will be stored.</td></tr>
<tr><td><b>Backup Archive</b></td><td>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>
<tr><td><b>.csv</b></td><td>Backup data using text CSV format.</td></tr>
<tr><td><b>Path</b></td><td>Where trend buffer data will be copied.
The below wild cards are supported
• %n = Trend name
• %y = Year
• %M = Month
• %d = Day
• %h = Hour
• %m = Minutes
• %s = Seconds</td></tr>
<tr><td><b>Select Fields</b></td><td>Fields that will be inside the dump file
ℹ Note that you can use a string tag to define the fields to dump at runtime.</td></tr>
<tr><td><b>Select Curves</b></td><td>Curves that will be inside the dump file
ℹ Note that you can use a string tag to define the fields to dump at runtime.</td></tr>
<tr><td><b>Time Spec</b></td><td>Timestamp of samples
• Local
Use the time of the HMI device where the project is running</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Trend Element</th><th>Description</th></tr>
<tr><td></td><td><table>
<tr><td></td><td>• Global
Use global time (GMT)</td></tr>
<tr><td><b>Data Format</b></td><td>Time and Date format. Placeholders can be used (see "Time and Date placeholders" on page 413)</td></tr>
<tr><td><b>Language</b></td><td>Language to use</td></tr>
</table></td></tr>
<tr><td><b>Sampling Filter / Trigger Filter</b></td><td>This parameter allows to specify a dynamic filter if required.
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.
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.</td></tr>
</table>

Use the add/remove buttons to add the tags to sample

+ −

Samples	Description
Name	Trend name
Title	Title that has to appears inside the trend table or the trend dump. • The placeholder **" \n "** (space + \n + space) can be used to split the label into two or more lines • The placeholder **[TagName]** (tag name enclosed in square brackets) can be used to display a tag value Example: "Temperature ([UnitaDiMisura])" will be shown as "Temperature (°C)" when the tag UnitaDiMisura = "°C"
Tag	Tag that must be sampled ⚠ **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.**
Format	Display format to use. Note that even the custom format can be used (see "Custom Formats" on page 34).

Samples	Description
	 Trend format Format: Numeric Decimal digits: 0 Leading digits: 0 Custom:
Comment	You can write whatever you want here

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.

```
    Number of available samples = Available Memory / Size of sample
```

Where the size of each sample is dependent on how many tags are used and can be calculated using the below formula:

```
    Size of sample = 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:

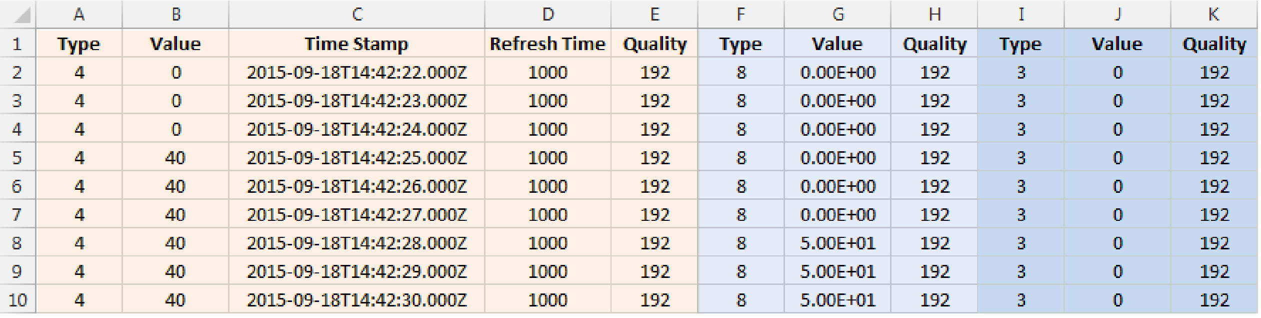
- Trend limits on "Table of functions and limits" on page 542
- Reserved memory for trend buffer on "HMI devices capabilities" on page 543

Exporting trend buffer data

Use the **DumpTrend** action to export trend buffer data to a .csv file. See "DumpTrend" on page 199 for the macro parameters details.

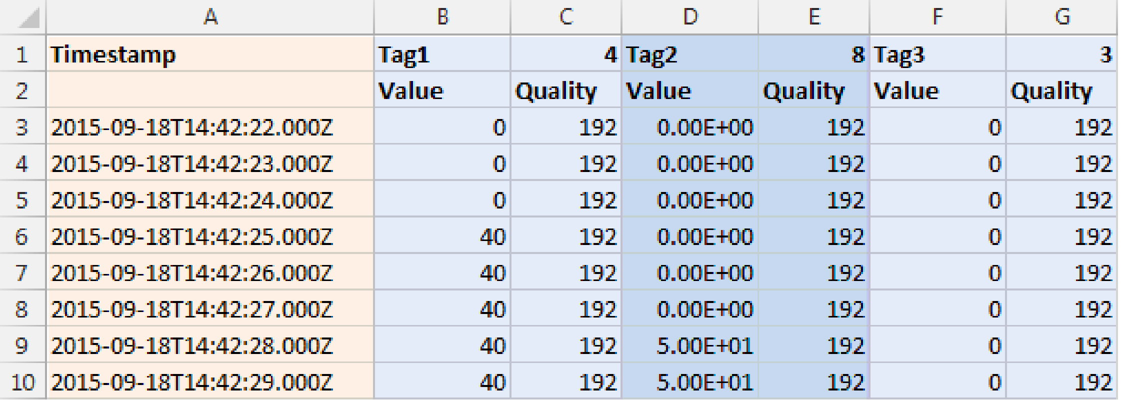
The exported .csv file could have different formats defined from the Dump Trend macro parameters. The different formats are maintained mainly for compatibilities reasons.

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

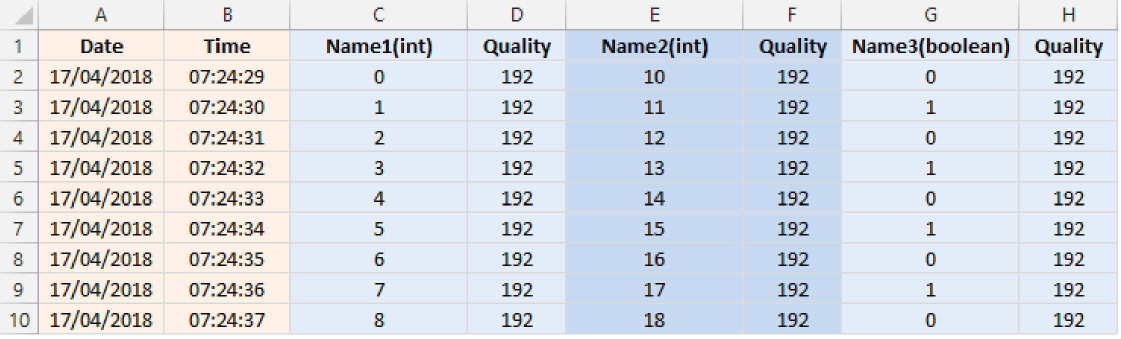
FileFormat: **Compact CSV**



	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

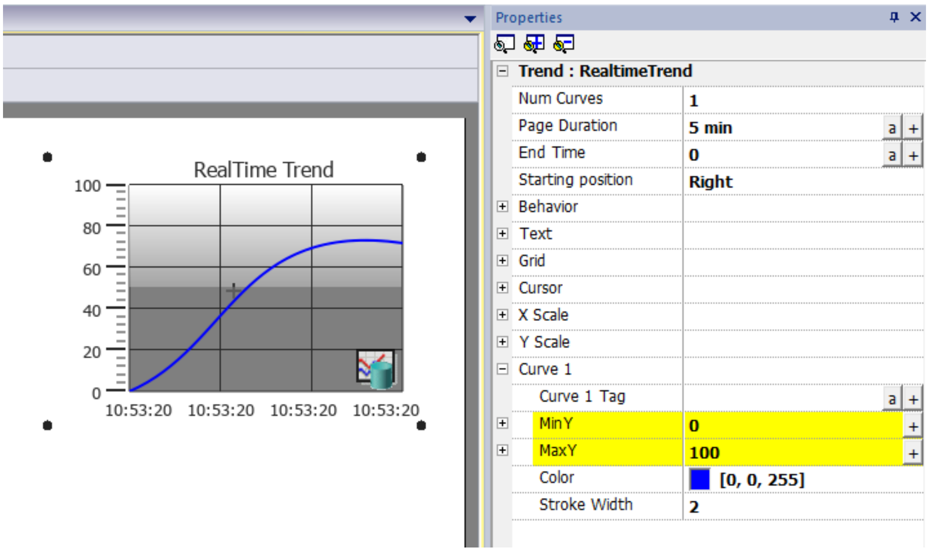
HMI Type	Buffer Size	Samples	Time
Win32	500 Mb	18.078.800 samples (2 tags)	25 Min
Linux	50 Mb	1.807.800 samples (2 tags)	4 Min
WinCE	25 Mb	903.900 samples (2 tags)	10 Min

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.



Trend : RealtimeTrend

Num Curves	1
Page Duration	5 min
End Time	0
Starting position	Right
Behavior	
Text	
Grid	
Cursor	
X Scale	
Y Scale	
Curve 1	
Curve 1 Tag	
MinY	0
MaxY	100
Color	[0, 0, 255]
Stroke Width	2

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
Num Curves	Number of trend curves to be displayed
Page Duration	Time window to show
End Time	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.
Starting Position	Specifies where the curve begin to be drawn when the page is opened (can be left, center or right).
Behavior	Definition of: • Min/Max of Y axis • Number of tickets to draw on the axes • Background image
Text	Trend title and font properties (font size, label, etc.)

Property	Description
Grid	Properties of grid presentation (colors)
Cursor	Properties of cursor presentation (enable and color)
X Scale	Properties of X Scale presentation
Y Scale	Properties of Y Scale presentation
Curve "n"	Tag that will be plotted in the trend widget.

Tag values can be scaled using the X Forms in the **Attach to** dialog. See ""Attach to" parameters" on page 45 for details.

History trend widget

The data collected and stored from the data logger can be analyzed using the History Trend widget.

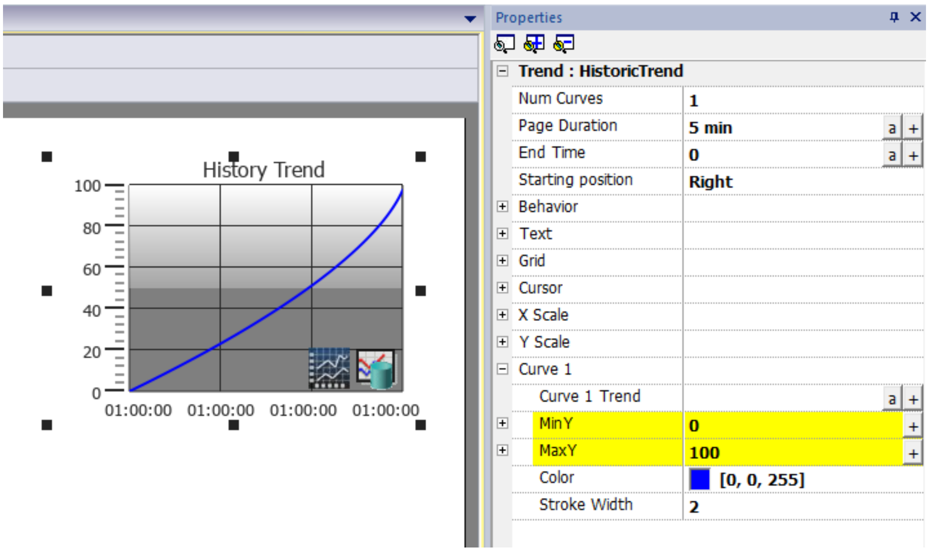
This is a two-step process:

- 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 268 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.



Trend : HistoricTrend	
Num Curves	1
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

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
Num Curves	Number of trend curves to be displayed
Page Duration	Time window to show
End Time	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.
Starting Position	Specifies where the curve begin to be drawn when the page is opened (can be left, center or right).
Behavior	Definition of: • Min/Max of Y axis • Number of tickets to draw on the axes • Background image
Text	Trend title and font properties (font size, label, etc.)
Grid	Properties of grid presentation (colors)
Cursor	Properties of cursor presentation (enable and color)
X Scale	Properties of X Scale presentation
Y Scale	Properties of Y Scale presentation
Curve "n"	Buffer that contains the tag's values to plotted in the trend widget.

Tag values can be scaled using the X Forms in the **Attach to** dialog. See ""Attach to" parameters" on page 45 for details.

Printing historical trend widget

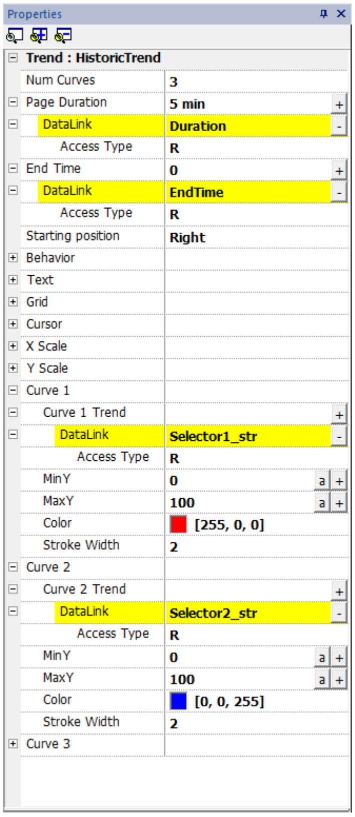
The historical trend widget can be found and used from the print report gallery.

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

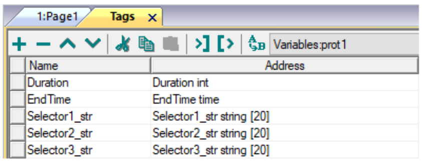
"Page Duration" with "End Time" define the piece of the trend buffer to print. "Curve Name" can be used to select the curve to show. An empty string means no curve to show.

SetTrendView() and ScrollTrendToTime() are display macros and have no effect on report printing.



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	



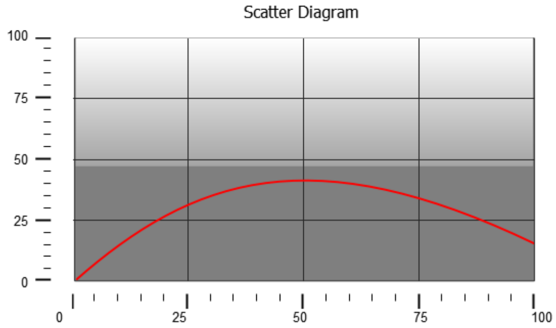
1:Page1 Tags

Variables:prot1

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.

5. 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

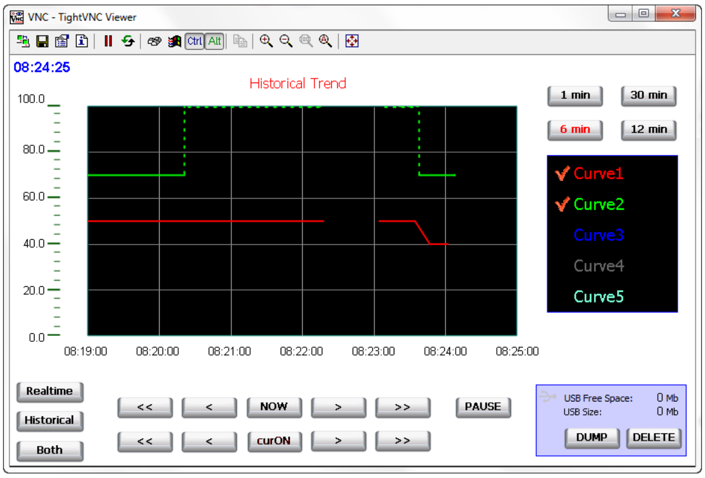
The scatter diagram widget can be found and used from the print report gallery. Note that using the attach to tag feature is possible to use tags to define some properties of the scatter diagram to print at runtime.

Trend widget tips

Be aware that some properties are only available when the Properties pane is in Advanced view.

Values outside range or invalid

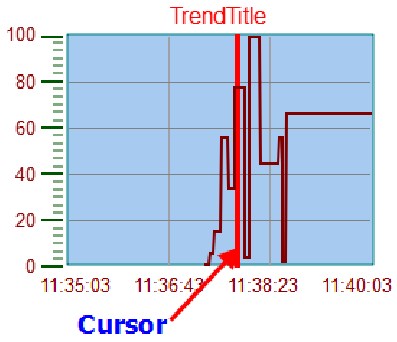
When trend value goes beyond the limits set for the trend widget, a dotted line is displayed. When the value of the tag is not available, for example the controller device is offline, no curve is drawn.



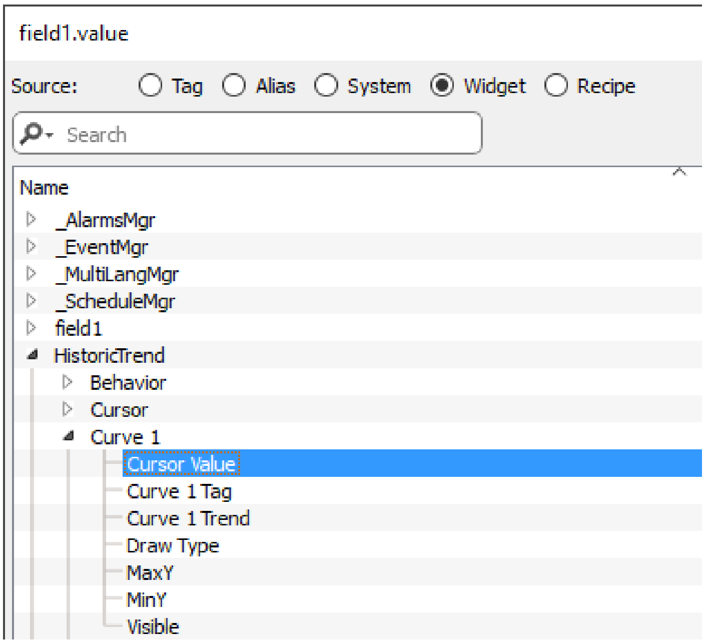
Showing trend values (cursor)

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.



field1.value

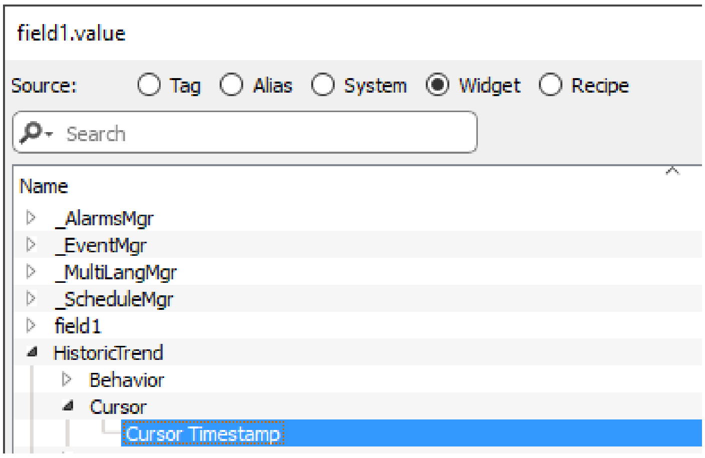
Source: ○ Tag ○ Alias ○ System ● Widget ○ Recipe

Search

Name
- _AlarmsMgr
- _EventMgr
- _MultiLangMgr
- _ScheduleMgr
- field1
- HistoricTrend
  - Behavior
  - Cursor
  - Curve 1
    - Cursor Value
    - Curve 1 Tag
    - Curve 1 Trend
    - Draw Type
    - MaxY
    - MinY
    - Visible

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.



field1.value

Source: ○ Tag ○ Alias ○ System ● Widget ○ Recipe

Search

Name
- _AlarmsMgr
- _EventMgr
- _MultiLangMgr
- _ScheduleMgr
- field1
- HistoricTrend
  - Behavior
  - Cursor
    - Cursor Timestamp

Modify trend widget properties at runtime

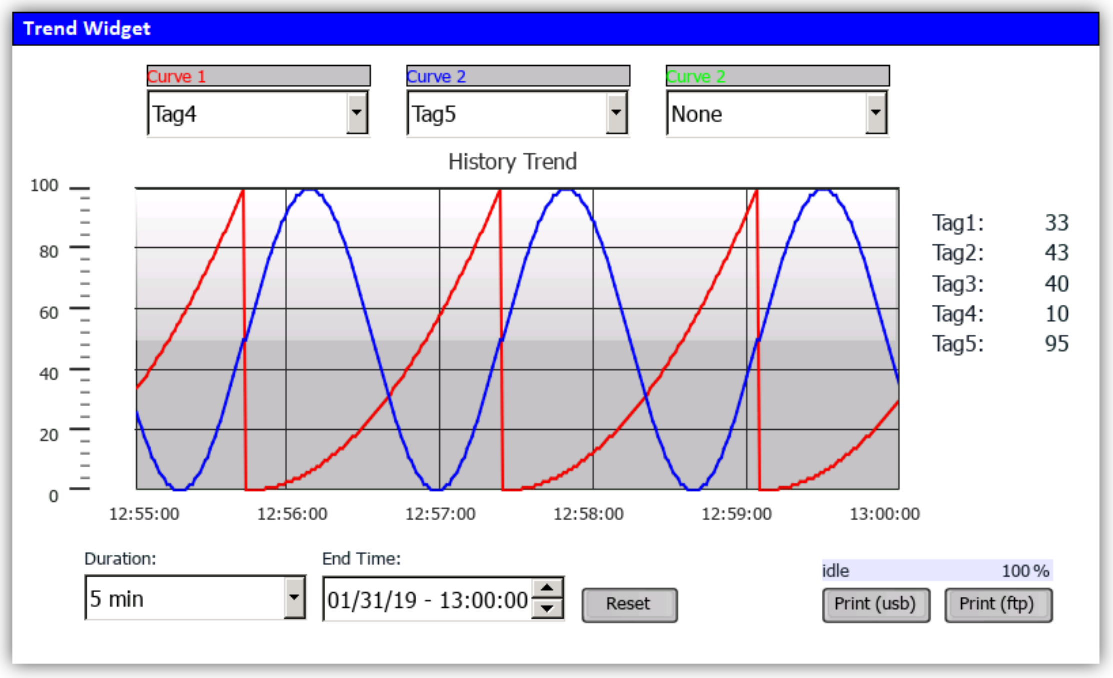
Using the attach to tag feature is possible to use tags to modifies some properties of the trend widgets at runtime.

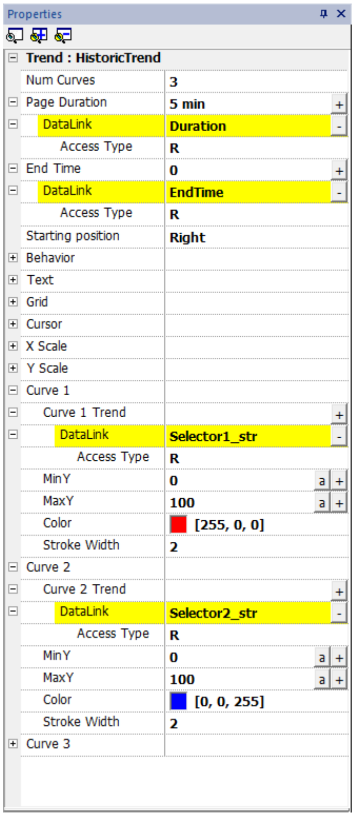
Example 1

Using :

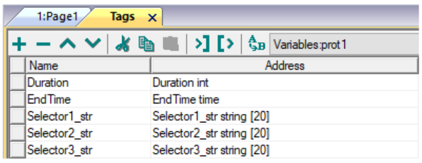
- Page Duration
- End Time
- Curve Name

is possible to modify from the runtime application the zoom factor, the window period and the curve to draw.





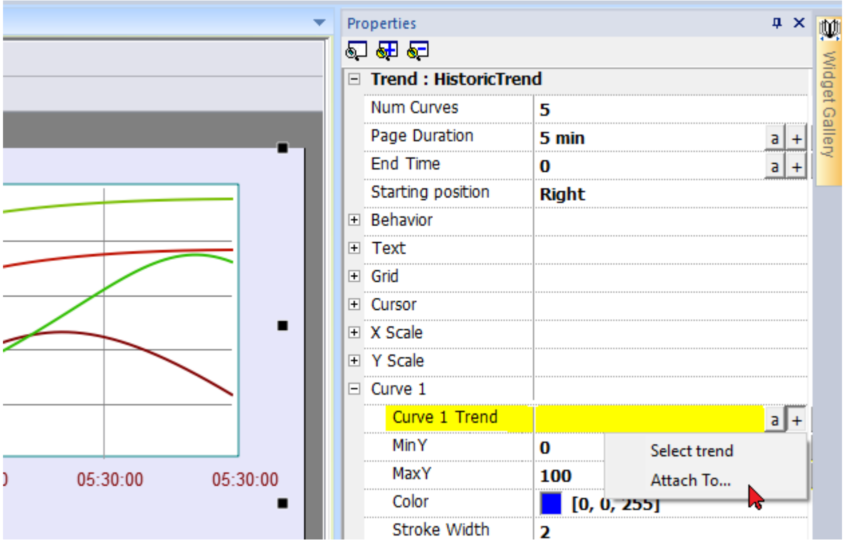
Trend : HistoricTrend	
Num Curves	3
Page Duration	5 min
DataLink	Duration
Access Type	R
End Time	0
DataLink	EndTime
Access Type	R
Starting position	Right
Behavior	
Text	
Grid	
Cursor	
X Scale	
Y Scale	
Curve 1	
Curve 1 Trend	
DataLink	Selector1_str
Access Type	R
MinY	0
MaxY	100
Color	🟥 [255, 0, 0]
Stroke 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	



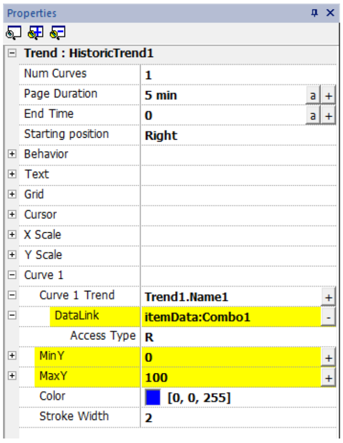
Name	Address
Duration	Duration int
EndTime	EndTime time
Selector1_str	Selector1_str string [20]
Selector2_str	Selector2_str string [20]
Selector3_str	Selector3_str string [20]

Example 2

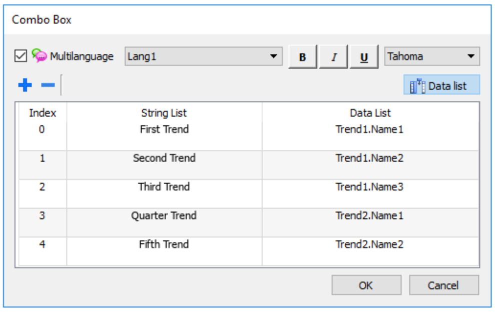
Curve property can be attached to a Combo Box to select the curve to draw



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 : HistoricTrend1	
Num Curves	1
Page Duration	5 min
End Time	0
Starting position	Right
Behavior	
Text	
Grid	
Cursor	
X Scale	
Y Scale	
Curve 1	
Curve 1 Trend	Trend1.Name1
DataLink	itemData:Combo1
Access Type	R
MinY	0
MaxY	100
Color	🟦 [0, 0, 255]
Stroke Width	2



Index	String List	Data List
0	First Trend	Trend1.Name1
1	Second Trend	Trend1.Name2
2	Third Trend	Trend1.Name3
3	Quarter Trend	Trend2.Name1
4	Fifth Trend	Trend2.Name2

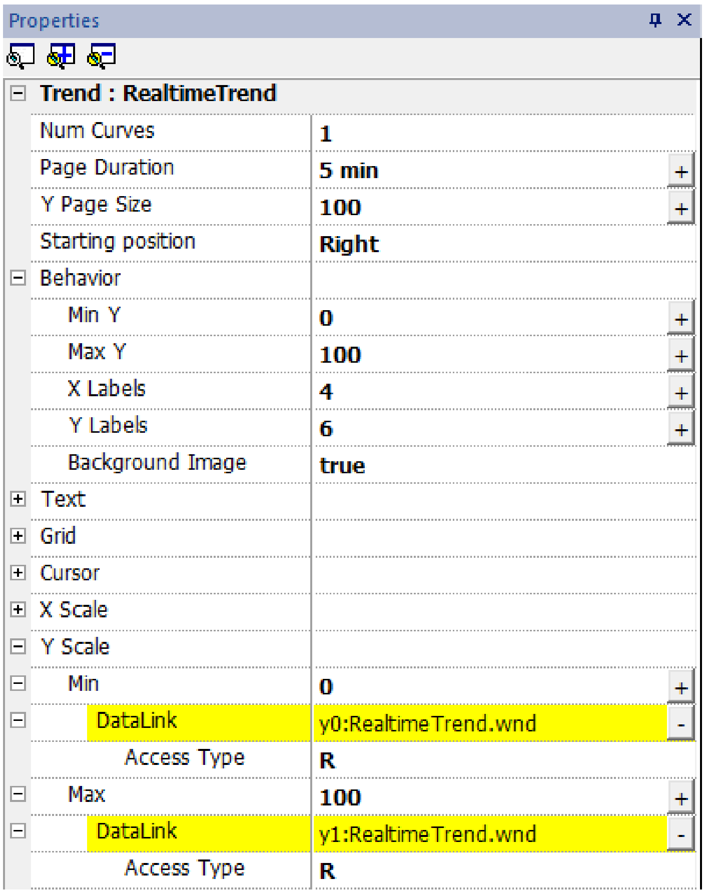
Trend widget gestures

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

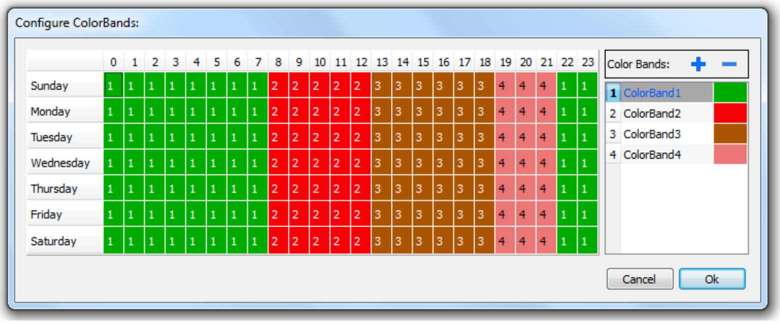
Request Sample property can be set for each curve and indicates the maximum numbers of samples read by the widget at one time from the trend buffer.

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.



Configure ColorBands:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Sunday	1	1	1	1	1	1	1	1	2	2	2	2	2	3	3	3	3	3	3	4	4	4	1	1
Monday	1	1	1	1	1	1	1	1	2	2	2	2	2	3	3	3	3	3	3	4	4	4	1	1
Tuesday	1	1	1	1	1	1	1	1	2	2	2	2	2	3	3	3	3	3	3	4	4	4	1	1
Wednesday	1	1	1	1	1	1	1	1	2	2	2	2	2	3	3	3	3	3	3	4	4	4	1	1
Thursday	1	1	1	1	1	1	1	1	2	2	2	2	2	3	3	3	3	3	3	4	4	4	1	1
Friday	1	1	1	1	1	1	1	1	2	2	2	2	2	3	3	3	3	3	3	4	4	4	1	1
Saturday	1	1	1	1	1	1	1	1	2	2	2	2	2	3	3	3	3	3	3	4	4	4	1	1

Color Bands: + −

1	ColorBand1	🟩
2	ColorBand2	🟥
3	ColorBand3	🟫
4	ColorBand4	🟥

Note: This feature only uses local time in the trend widget, not the global time option.

Calendar color bands example

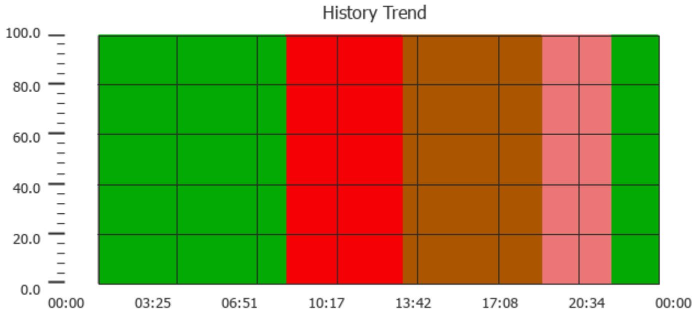
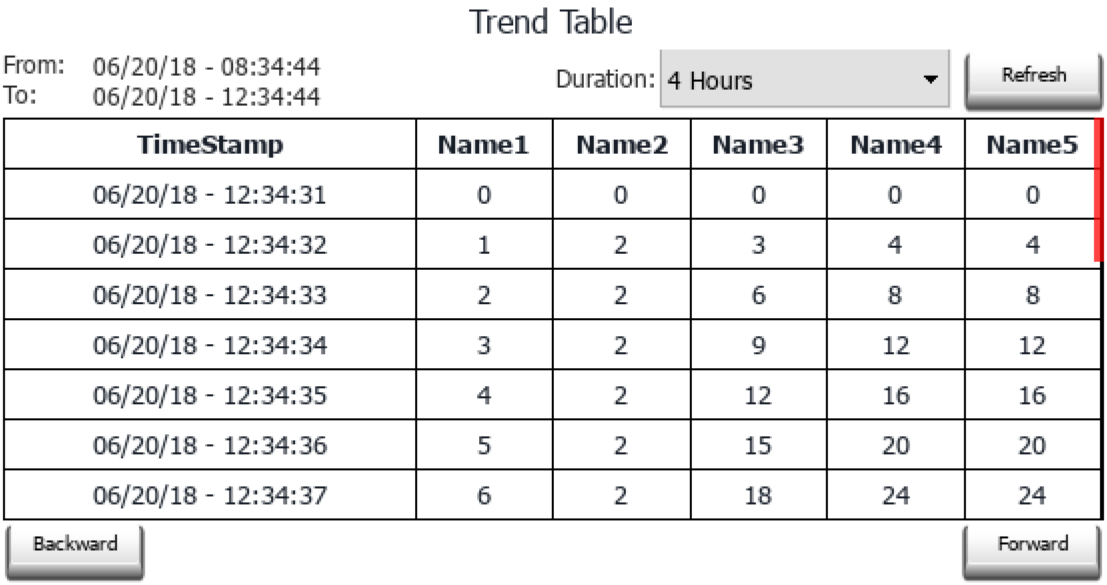


Table trend widget

Path: **Widget Gallery**> **Basic**> **Trends/Graphs**

Display contents of a trend buffer inside a widget



Trend Table

From: 06/20/18 - 08:34:44
To: 06/20/18 - 12:34:44

Duration: 4 Hours

TimeStamp	Name1	Name2	Name3	Name4	Name5
06/20/18 - 12:34:31	0	0	0	0	0
06/20/18 - 12:34:32	1	2	3	4	4
06/20/18 - 12:34:33	2	2	6	8	8
06/20/18 - 12:34:34	3	2	9	12	12
06/20/18 - 12:34:35	4	2	12	16	16
06/20/18 - 12:34:36	5	2	15	20	20
06/20/18 - 12:34:37	6	2	18	24	24

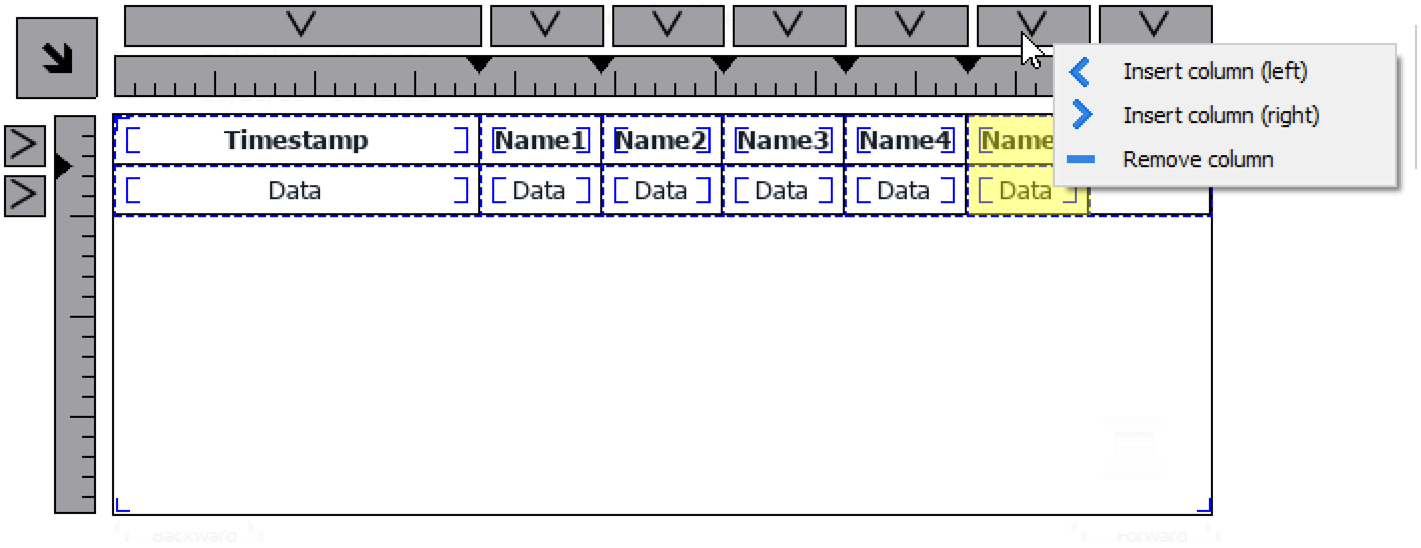
Buttons:

- 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

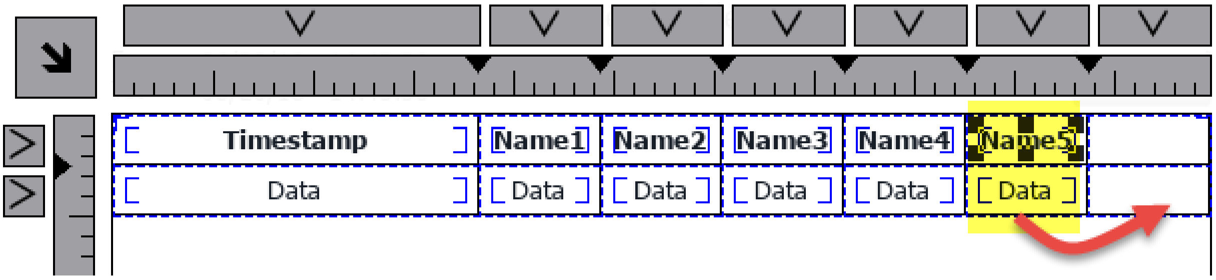
Parameter	Description
TrendName	Trend Buffer from which the samples are retrieved (see "Data logging" on page 268)
Heading	Heading labels The visible labels inside the AGI Creatoreditor are only placeholders, the actual labels that will be displayed are defined in the trend configuration (see "Data logging" on page 268)
Page Duration	Time window to show
End Time	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.
Time Spec	Time format: • **Local** = show the time values of the HMI device. • **Global** = show the time values using UTC format.
Date Format	Select the Date and Time format
Table Layout	Defines the characteristics of the scroll bar and allows to remove the header of the table

Adding or removing trend columns

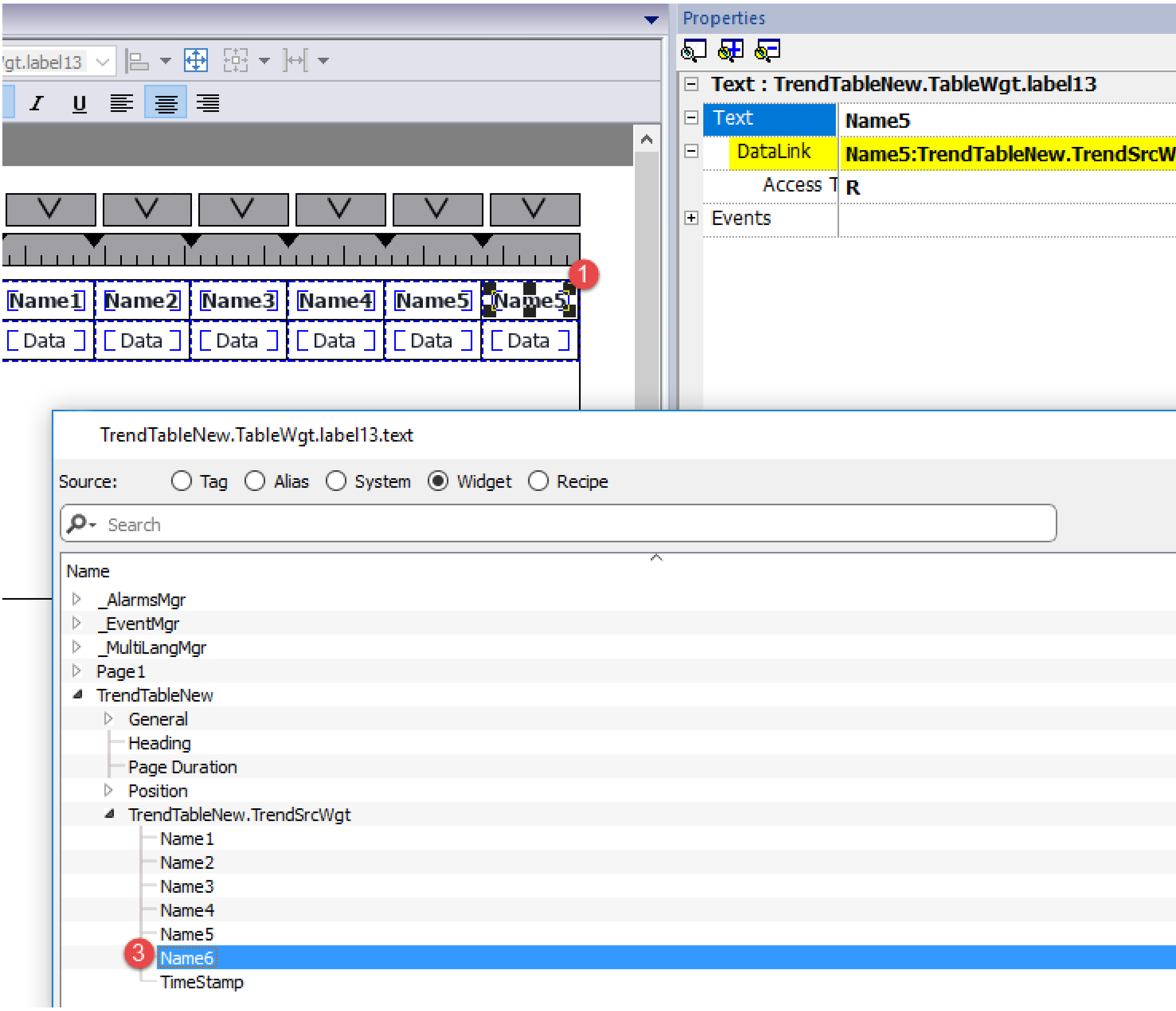
To add or remove a column, double-click on the grid to enter edit mode and right-click on the column selector to open the context menu from where to insert or remove a column.



Copy and past fields from another column



Then use the properties panel to select the trend element to add to the new columns



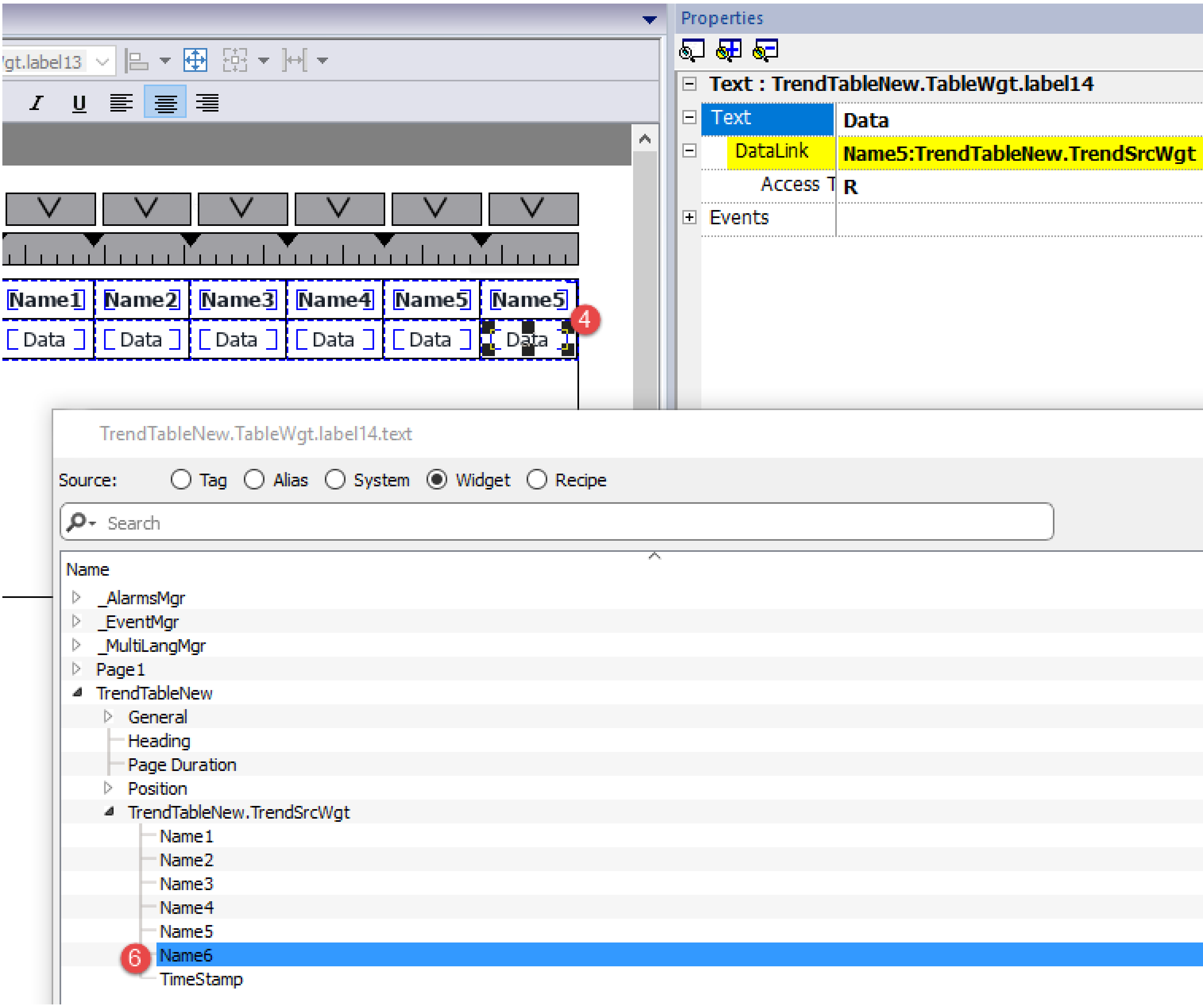
Properties

Text : TrendTableNew.TableWgt.label13

Text	Name5
DataLink	Name5:TrendTableNew.TrendSrcW
Access T	R
Events	

TrendTableNew.TableWgt.label13.text

Source: ○ Tag ○ Alias ○ System ● Widget ○ Recipe



Text : TrendTableNew.TableWgt.label14

Text	Data
DataLink	Name5:TrendTableNew.TrendSrcWgt
Access T	R
Events	

TrendTableNew.TableWgt.label14.text

Source: ○ Tag ○ Alias ○ System ● Widget ○ Recipe

Printing trend table

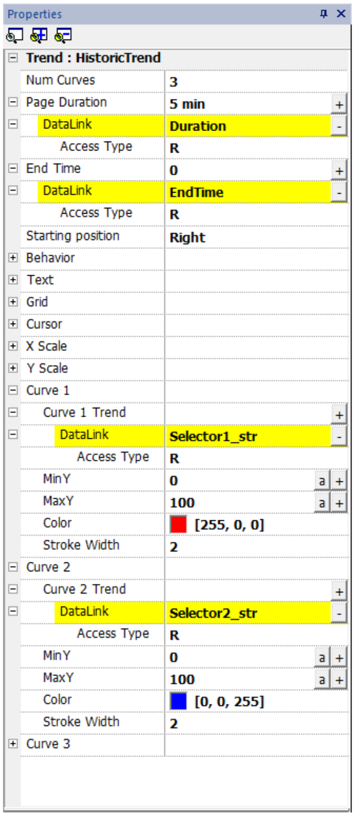
A trend table widget without buttons can be found and used from the print report gallery. The table can be drawn and enlarged to fill the entire page. If the number of lines to printed is greater of one page, the trend table will be printed using additional pages. See the "Table of functions and limits" on page 542 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

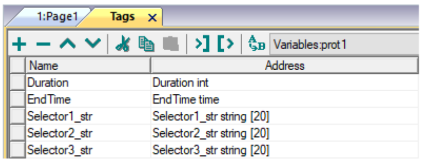
"Page Duration" with "End Time" define the piece of the trend buffer to print. "Curve Name" can be used to select the curve to show. An empty string means no curve to show.

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

Variables:prot1

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

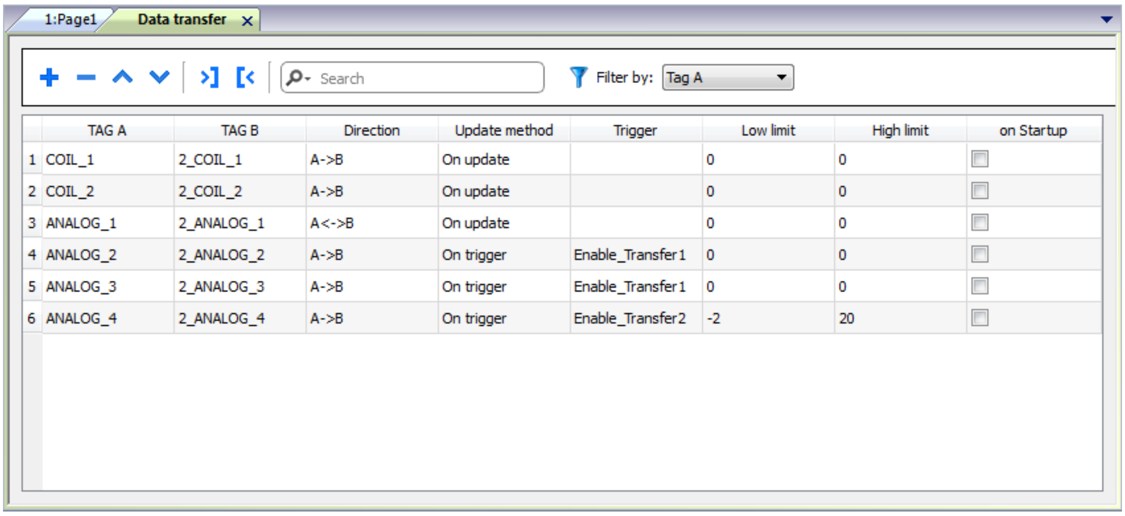
Data transfer allows you transferring variable data from one device to another. Using this feature an HMI device can operate as a gateway between two devices, even if they do not use the same communication protocol.

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	on Startup
1	COIL_1	2_COIL_1	A->B	On update		0	0	□
2	COIL_2	2_COIL_2	A->B	On update		0	0	□
3	ANALOG_1	2_ANALOG_1	A<->B	On update		0	0	□
4	ANALOG_2	2_ANALOG_2	A->B	On trigger	Enable_Transfer1	0	0	□
5	ANALOG_3	2_ANALOG_3	A->B	On trigger	Enable_Transfer1	0	0	□
6	ANALOG_4	2_ANALOG_4	A->B	On trigger	Enable_Transfer2	-2	20	□

To add a new rule, click **+**: a new tag line is added.

Data transfer toolbar

Prameter	Description
Import/ Export	Imports or exports data transfer settings from or to a .csv file.
Search	Displays only rows containing the search keyword.
Filter by	Display only rows matching filter and search field.

Data transfer parameters

Prameter	Description
TAG A/ TAG B	Pair of tags to be mapped for exchanging through the HMI device.
Direction	Transfer direction. **A->B** and **B->A**: Unidirectional transfers, values are always copied from one tag and sent to the other tag in the specified direction. **A<->B**: Bidirectional transfer, values are transferred to and from both tags.
Update Method	**On trigger**: 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.

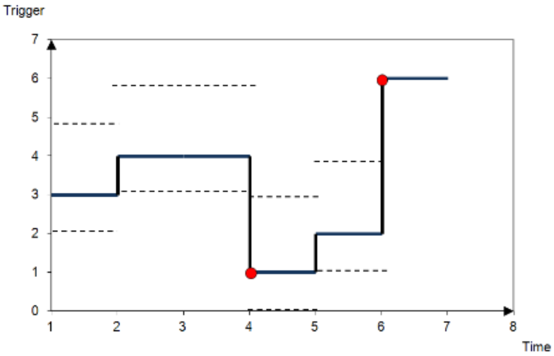
Prameter	Description
	Note: This method applies only to unidirectional transfers (A->B or B->A). **On Update**: Data transfer occurs whenever the value of the source tag changes. Note: This method applies both to unidirectional and to bidirectional transfers (A->B, B->A and A<->B). Note: The Runtime cyclically monitors source tags changes (trigger tag when using On Trigger or tags to transfer when using On Update) based on Tag editor **Rate** parameter. If **Rate** setting for source Tag is 500 ms (default), the system checks for updates every 500 ms. Note: Changes on source tags faster than **Rate** may be not detected .
Trigger, **High limit,** **Low limit**	Tag that triggers the data transfer process. When this tag changes its value outside the boundaries set as **High limit** and **Low limit**, 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. This mechanism allows triggering data transfers only when significant variations of the reference values occur. **Low limit** is less or equal to zero. Note: If both **Low limit** and **High limit** are set to "0", data transfer occurs whenever the value of the trigger tag changes.
on Startup	When selected, data transfer is forced: • on HMI startup if the quality of the source tag is good • after communication errors, when the associate device nodes return active See "Objects" on page 479 for details on quality. **Important: Data transfers executed on startup may have major impact on the HMI device boot time. Enable this option only when necessary.**

Example of limit setting

High limit = 1,9

Low limit = - 0,9

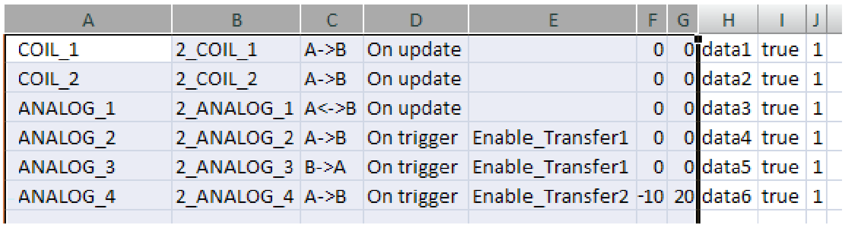
• = points where the data transfer is triggered



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. **Important: When you edit the .csv file and you add any extra line, make sure you enter a unique identifier in this column.**
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.

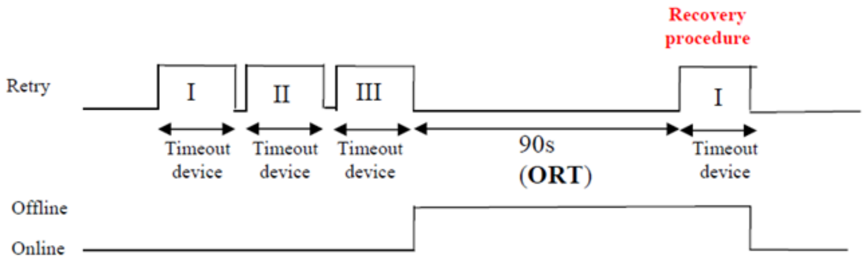
Note: This feature is not supported by all communication protocols. Check protocol documentation to know if it is supported or not.

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.

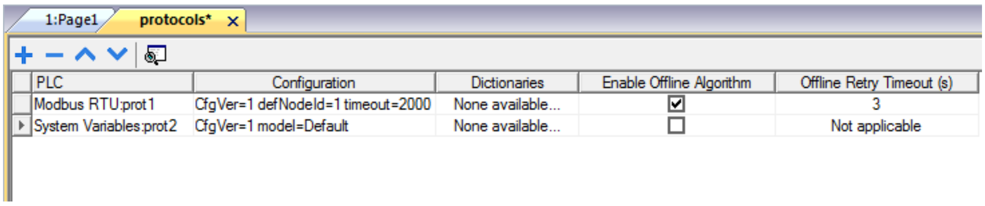
Tip: To make this feature more dynamic, you may decide not to indicate a specific **NodeID** but attach it to the value of a tag or to an internal variable created to identify different devices that might be installed in your network.

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...	☑	3
System Variables:prot2	CfgVer=1 model=Default	None available...	☐	Not applicable

Note: Not all protocols support this feature.

Parameter	Description
Enable Offline Algorithm	Enables offline management for the protocol
Offline Retry Timeout	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. AGI Creator also extends the TrueType Fonts provided by Windows systems to provide different font faces associated with different character sets.

AGI Creator also allows you to provide strings for each of the languages supported.

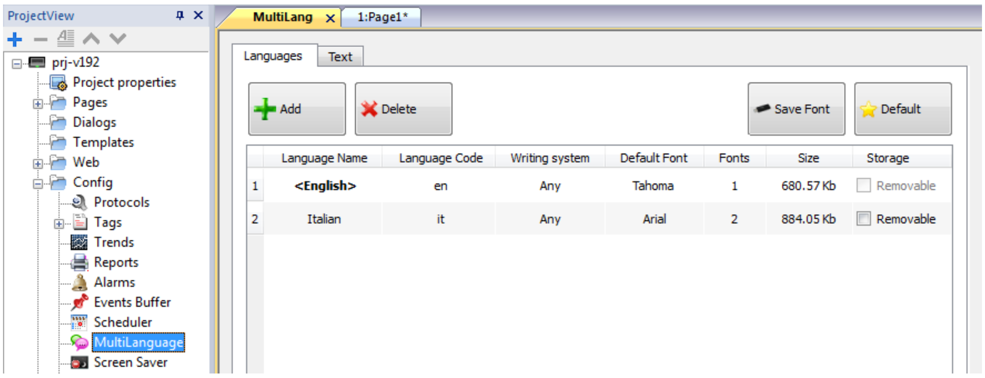
AGI Creator also allows you to change the display language so that you can see the page look and feel during the design phase.

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)

The Multi-language editor

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



	Language Name	Language Code	Writing system	Default Font	Fonts	Size	Storage
1	<English>	en	Any	Tahoma	1	680.57 Kb	□ Removable
2	Italian	it	Any	Arial	2	884.05 Kb	□ Removable

Language settings

Parameter	Description
Language Name	Name identifying the language in the project.
Language Code	ISO 639 language code identifier, used to match language items when importing resources from external xml files.
Writing system	Select the set of fonts to be used with the language
Default Font	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.
Fonts	Number of fonts associated with the selected language.
Size	Memory used to store font files.
Storage	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

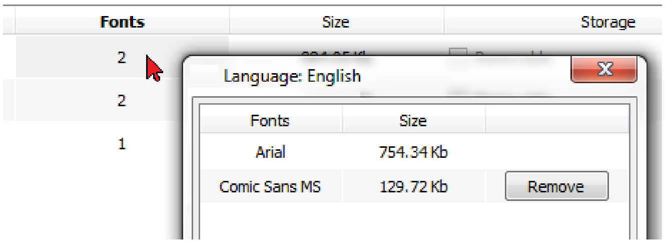
1. In the **Languages** tab, click **+**: a line is added to the table.
2. Enter all language settings.
3. Click **Default** to set the selected language as the default language when the Runtime starts.
4. Click **Save Font** to copy the fonts you marked as **Removable** on an external memory.

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.



Language: English

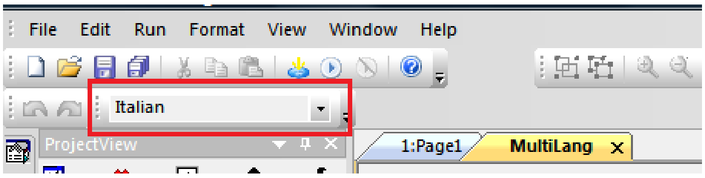
Fonts	Size	
Arial	754.34 Kb	
Comic Sans MS	129.72 Kb	Remove

2. Select the fonts to be removed and click **Remove**: removed fonts are replaced with the default font.

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.

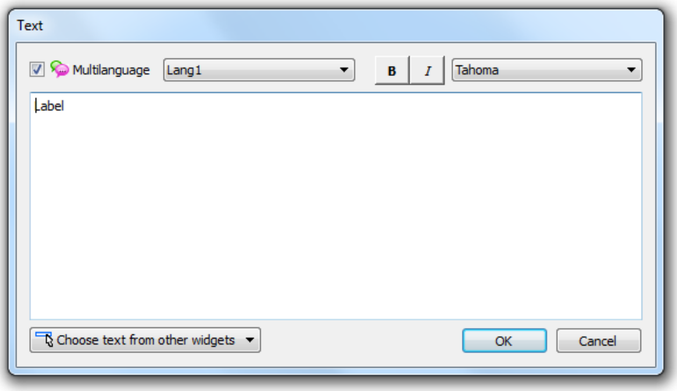


Multi-language widgets

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.



☑ Multilanguage: Lang1
Font: Tahoma

Label

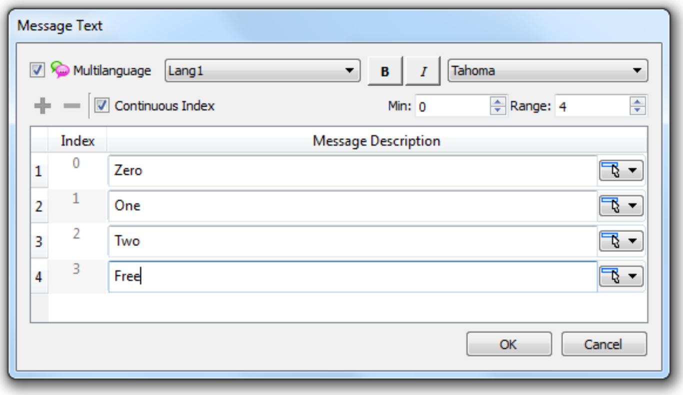
Enable/disable multi-language function, edit the text for the selected language and choose the font.

Note: Bold, italic and color properties set here for the widget are applied to all languages .

Parameter	Description
Multilanguage	Enable/disable multi-language function for the widget.
Choose text from other widget	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.



☑ Multilanguage: Lang1
Font: Tahoma
☑ Continuous Index
Min: 0
Range: 4

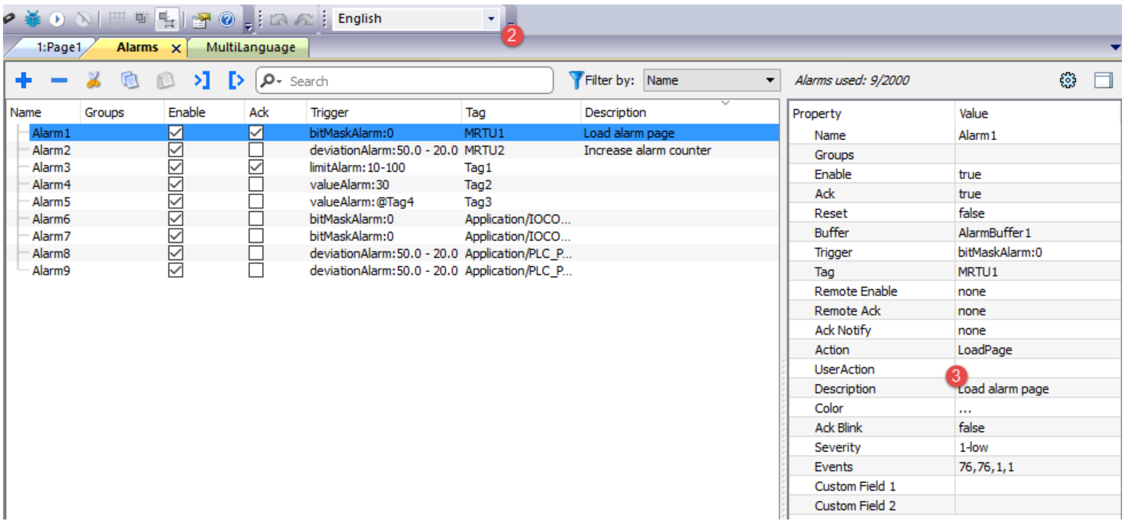
	Index	Message Description
1	0	Zero
2	1	One
3	2	Two
4	3	Free

Parameter	Description
Multilanguage	Enable/disable multi-language function for the widget.
Continuous Index	Index for the widget is set of contiguous numbers (example 3, 4,5,6)
Min	Starting number for index
Range	Number of messages
Choose text from other widget	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.



Name	Groups	Enable	Ack	Trigger	Tag	Description
Alarm1		☑	☑	bitMaskAlarm:0	MRTU1	Load alarm page
Alarm2		☑	□	deviationAlarm:50.0 - 20.0	MRTU2	Increase alarm counter
Alarm3		☑	☑	limitAlarm:10-100	Tag1	
Alarm4		☑	□	valueAlarm:30	Tag2	
Alarm5		☑	□	valueAlarm:@Tag4	Tag3	
Alarm6		☑	□	bitMaskAlarm:0	Application/IOCO...	
Alarm7		☑	□	bitMaskAlarm:0	Application/IOCO...	
Alarm8		☑	□	deviationAlarm:50.0 - 20.0	Application/PLC_P...	
Alarm9		☑	□	deviationAlarm:50.0 - 20.0	Application/PLC_P...	

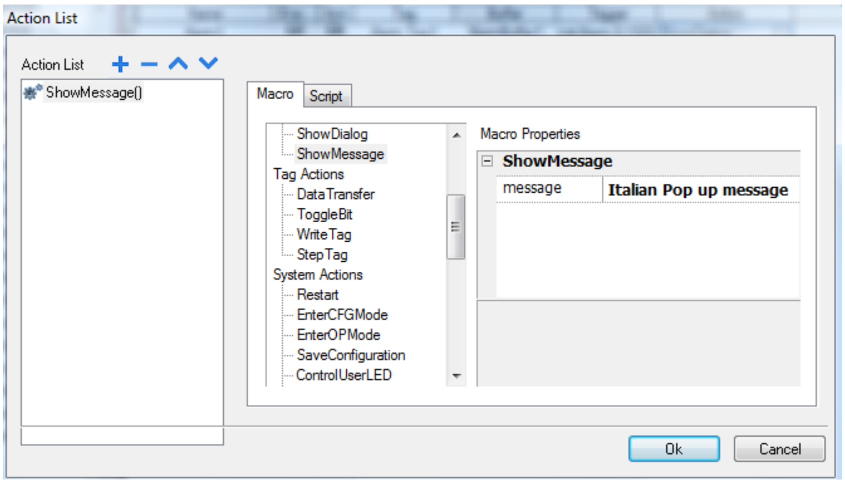
Property	Value
Name	Alarm1
Groups	
Enable	true
Ack	true
Reset	false
Buffer	AlarmBuffer1
Trigger	bitMaskAlarm:0
Tag	MRTU1
Remote Enable	none
Remote Ack	none
Ack Notify	none
Action	LoadPage
UserAction	
Description	Load alarm page
Color	...
Ack Blink	false
Severity	1-low
Events	76,76,1,1
Custom Field 1	
Custom Field 2	

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.



ShowMessage

message	Italian Pop up message

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.

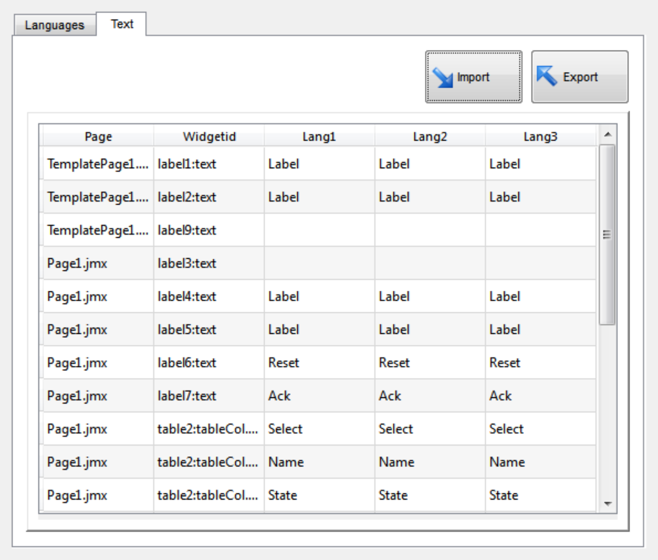
Important: The .csv file exported by AGI Creator is coded in Unicode, to edit it you need a specific tool supporting Unicode encoded .csv files.

Exporting and reimporting strings

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

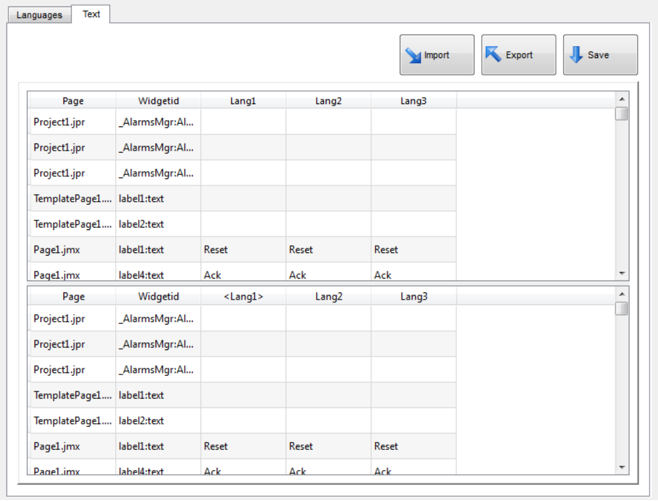
To export and re-import multi-language strings:

1. In the **Text** tab, click **Export**: all multi-language strings are exported to a .csv file.



Page	Widgetid	Lang1	Lang2	Lang3
TemplatePage1....	label1:text	Label	Label	Label
TemplatePage1....	label2:text	Label	Label	Label
TemplatePage1....	label9:text			
Page1.jmx	label3:text			
Page1.jmx	label4:text	Label	Label	Label
Page1.jmx	label5:text	Label	Label	Label
Page1.jmx	label6:text	Reset	Reset	Reset
Page1.jmx	label7:text	Ack	Ack	Ack
Page1.jmx	table2:tableCol....	Select	Select	Select
Page1.jmx	table2:tableCol....	Name	Name	Name
Page1.jmx	table2:tableCol....	State	State	State

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.



Page	Widgetid	Lang1	Lang2	Lang3
Project1.jpr	_AlarmsMgr:Al...			
Project1.jpr	_AlarmsMgr:Al...			
Project1.jpr	_AlarmsMgr:Al...			
TemplatePage1....	label1:text			
TemplatePage1....	label2:text			
Page1.jmx	label1:text	Reset	Reset	Reset
Page1.jmx	label4:text	Ack	Ack	Ack

Page	Widgetid	<Lang1>	Lang2	Lang3
Project1.jpr	_AlarmsMgr:Al...			
Project1.jpr	_AlarmsMgr:Al...			
Project1.jpr	_AlarmsMgr:Al...			
TemplatePage1....	label1:text			
TemplatePage1....	label2:text			
Page1.jmx	label1:text	Reset	Reset	Reset
Page1.jmx	label4:text	Ack	Ack	Ack

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

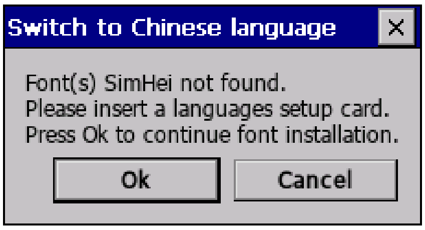
After the project download, the HMI Runtime will start using the language set as default. You can change the language using the **SetLanguage** action. See "MultiLanguage actions" on page 179.

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 497 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.



Switch to Chinese language

Font(s) SimHei not found.
Please insert a languages setup card.
Press Ok to continue font installation.

Limitations in Unicode support

AGI Creator 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]	\ / * ? : > < | " & # % ;
Alarms	Name	ASCII [36..126]	\ / * ? : > < | " & # % ;
	Description	Unicode	[] - for live tags, \ escape seq for [and \
Events	Buffer Name	ASCII [32..126]	\ / * ? : > < | " & # % ;
Scheduler	Name	ASCII [32..126]	\ / * ? : > < | " & # % ;
Languages	Language Name	ASCII [32..126]	\ / * ? : > < | " & # % ;
	Texts in widgets	Unicode	-
	Texts from import files	Unicode	-
User Group	Group Name	a-z A-Z _	admin,guest,unauthorized
	Comments	Unicode	-
User	Name	ASCII [32..126]	\ / * ? : > < | " & # % ;
	Password	Unicode	-
	Comment	Unicode	-
Recipes	Name	ASCII [32..126]	\ / * ? : . > < | " & % ; ,
	Set Name	ASCII [32..126]	\ / * ? : . > < | " & % ; ,
	Element name	ASCII [32..126]	\ / * ? : . > < | " & % ; ,
General	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,-,_	-

Area	Property	Charset Accepted	Reserved Chars/Strings
	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,-,_	-
Runtime	PLC Communication	UTF-8, Latin1, UCS-2BE, UCS-2LE, UTF-16BE, UTF-16LE	-

26 Scheduler

AGI Creator provides a scheduler engine that can execute specific actions at set intervals, or on a time basis.

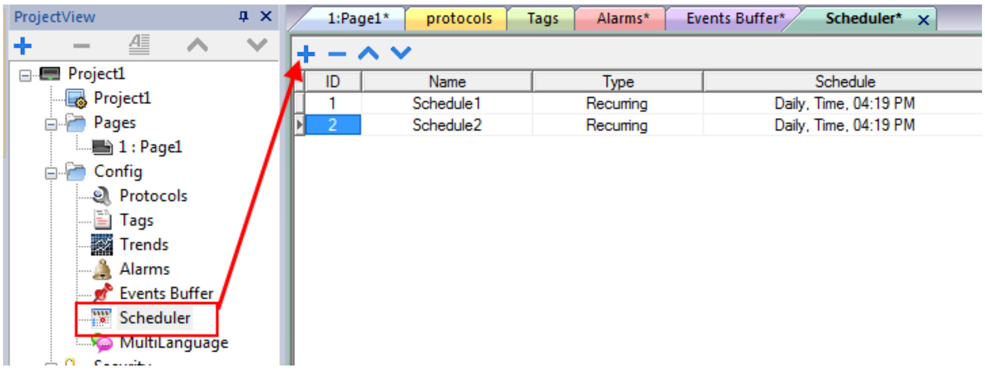
Creating a schedule is typically a two-step process:

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.

Creating a schedule

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

- Click **+** to add a schedule.



ID	Name	Type	Schedule
1	Schedule1	Recurring	Daily, Time, 04:19 PM
2	Schedule2	Recurring	Daily, Time, 04:19 PM

Schedule parameters

Parameter	Description
ID	Unique code assigned automatically to the schedule
Name	Name of schedule
Type	Type of schedule: • **Recurring**, see "Recurring schedule" on the facing page for details. • **HighResolution**, see "HighResolution schedule" on the facing page for details
Schedule	Scheduler settings and options. See "Recurring schedule" on the facing page for details.
Action	Actions to be executed at the scheduled time 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. AGI Web). • loadPage • prevPage • nextPage • showDialog • showMessage • setLanguage • jsAction
Priority	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
Type	Frequency of the scheduled actions
Mode	Specific settings required by each scheduler type
Condition	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. Note: Only tags attached to the Boolean data type are shown.
Actions	Actions to be executed by the schedule. **Important: Actions and schedule parameters cannot be modified at runtime**
Date	Date when the scheduled actions will be executed
Time/Offset	This field display one of the following: **Time** = when the scheduled actions will be executed **Offset**= delay or advance with respect to the selected mode.
Location	Reference location to calculate sunset/sunrise time.
weekdays	Days of the week in which the scheduled actions will be executed.
On startup	Executes schedule at start up
Enable schedule	Enables/disables the schedule
Execute only at startup	Executes the schedule only once at start up

Schedule type options

Option	Description
By Date	Actions are executed on the specified date and time.
Daily	Actions are executed daily at the specified time.
Every	Actions are executed with the specified interval (Range: 1 s–1 day)
Hourly	Actions are executed every hour at the specified minute.
Monthly	Actions are executed every month at the specified date and time.
Weekly	Actions are executed every week on the specified weekday(s) and time.
Yearly	Actions are executed every year at the specified date and time.

Schedule mode options

Option	Description
Time	Depends on the schedule type. Allows you to specify date/time/week data.
Random10	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.
Random20	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.
Sunrise+	Actions are executed with a specified delay after sunrise. The delay is set in minutes/hours and sunrise time is location specific.
Sunrise-	Actions are executed with a specified advance before sunrise. The advance is set in minutes/hours and sunrise time is location specific.
Sunset+	Actions are executed with a specified delay after sunset. The delay is set in minutes/hours and sunset time is location specific.
Sunset-	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" below 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 the 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.

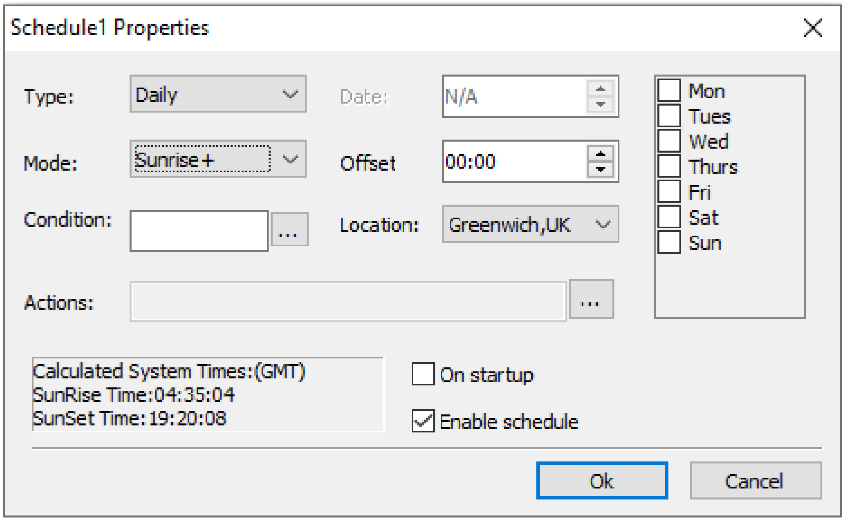
Location files position

Application	Location file path
AGI Creator	*DEIF\AGI Software Pack\languages\shared\studio\config\Target_Location.xml*
HMI Devices	*DEIF\AGI Software Pack\runtime\<HW Platform>\config\Target_Location.xml*
Simulator	*DEIF\AGI Software Pack\simulator\config\Target_Location.xml*
AGI PC Runtime	*DEIF\AGI Software Pack\server\config\Target_Location.xml*

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: Daily

Date: N/A

Mode: Sunrise+

Offset 00:00

Condition:

Location: Greenwich,UK

Actions:

□ Mon
□ Tues
□ Wed
□ Thurs
□ Fri
□ Sat
□ Sun

Calculated System Times:(GMT)
SunRise Time:04:35:04
SunSet Time:19:20:08

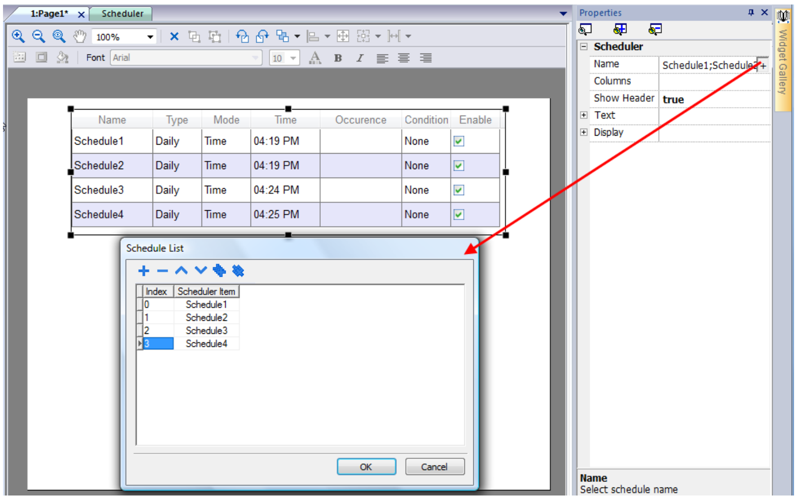
□ On startup

☑ Enable schedule

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.



Name	Type	Mode	Time	Occurence	Condition	Enable
Schedule1	Daily	Time	04:19 PM		None	☑
Schedule2	Daily	Time	04:19 PM		None	☑
Schedule3	Daily	Time	04:24 PM		None	☑
Schedule4	Daily	Time	04:25 PM		None	☑

Schedule List

Index	Scheduler Item
0	Schedule1
1	Schedule2
2	Schedule3
3	Schedule4

Scheduler

Name	Schedule1;Schedule2
Columns	
Show Header	true
Text	
Display	

Name
Select schedule name

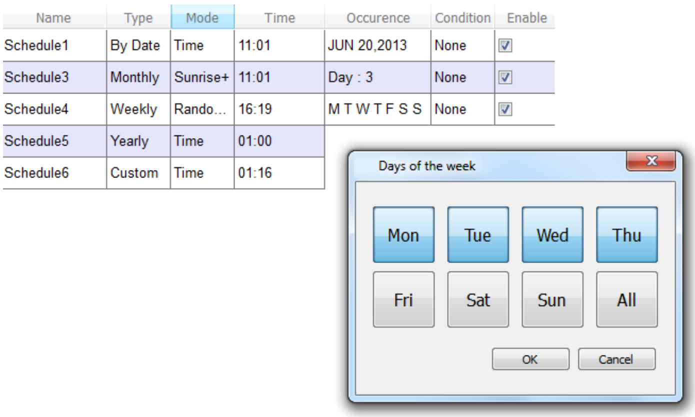
4. In the **Properties** pane, customize all settings.

Scheduler settings

Parameter	Description
Name	Schedule to be displayed
Columns	Columns to be displayed and their characteristics
Show Header	Shows/hides column headers
Time Spec	Time to be displayed at runtime
Text	Font used for text
Display	Table styles

Scheduling events at runtime

At runtime you can modify the following scheduling parameters.



Name	Type	Mode	Time	Occurence	Condition	Enable
Schedule1	By Date	Time	11:01	JUN 20,2013	None	☑
Schedule3	Monthly	Sunrise+	11:01	Day : 3	None	☑
Schedule4	Weekly	Rando...	16:19	M T W T F S S	None	☑
Schedule5	Yearly	Time	01:00			
Schedule6	Custom	Time	01:16			

Parameter	Description
Occurrence	Information on the type of schedule and time of execution
Condition	Condition applied to action execution
Enable	Enabels/disables the execution of the scheduled actions without deleting the schedule.

See "Recurring schedule" on page 313 for details on schedule parameters.

27 21 CFR Part 11 Compliance

AGI Creator 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 AGI Creator 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	AGI Creator compliance level (v2.8)
11.10(a)	(a) Validation of systems to ensure accuracy, reliability, consistent intended performance, and the ability to discern invalid or altered records.	Reports generated by AGI Creator 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. References: • "SaveEventArchive" on page 208 • "PrintGraphicReport" on page 192
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.	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. References: • "Enable/disable audit trail" on page 346 • "Exporting audit trail as .csv files" on page 352 • "SaveEventArchive" on page 208 • "Printing audit table" on page 352 • "PrintGraphicReport" on page 192
11.10(c)	Protection of records to enable their accurate and ready retrieval throughout the records retention period.	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. References: • "SaveEventArchive" on page 208 • "PrintGraphicReport" on page 192

Chapter	Description	AGI Creator compliance level (v2.8)
		- "Scheduler" on page 311
11.10(d)	Limiting system access to authorized individuals.	Application developer is responsible for the appropriate security configuration of the application. References: - "User management and passwords" on page 333
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.	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. References: - "Exporting audit trail as .csv files" on page 352 - "SaveEventArchive" on page 208 - "Printing audit table" on page 352 - "PrintGraphicReport" on page 192 - "Scheduler" on page 311
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.	The HMI application can be configured - to be accessible only after user sign in with its own password - objects can be configured to be available or not available depending on the user who logged in to the system - resources can be configured to require a password confirmation before be modified References: - "User management and passwords" on page 333 - "Electronic Signature" on page 347
11.10(h)	Use of device (e.g., terminal) checks to determine, as appropriate, the validity of the source of data input	Resources can be configured to be accessible only from selected user groups. List of allowed IP address

Chapter	Description	AGI Creator compliance level (v2.8)
	or operational instruction.	can be configured from the User Management settings. References: - "Modifying access permissions" on page 335
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)	Use of appropriate controls over systems documentation including: (1) Adequate controls over the distribution of, access to, and use of documentation for system operation and maintenance. (2) Revision and change control procedures to maintain an audit trail that documents time-sequenced development and modification of systems documentation.	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.	AGI Creator has been designed for operation in closed systems.
11.50(a)	Signed electronic records shall contain information associated with the signing that clearly indicates all of the following: (1) The printed name of the signer; (2) The date and time when the signature was executed; and (3) The meaning (such as review, approval,	All records will be added to the audit trail with time stamp and user id of logged user. References: - "Exporting audit trail as .csv files" on page 352 - "Table audit widget" on page 351

Chapter	Description	AGI Creator compliance level (v2.8)
	responsibility, or authorship) associated with the signature.	
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)	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. (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. (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.	User responsibility.
11.200(a)	(a) Electronic signatures that are not based upon biometrics shall: (1) Employ at least two distinct identification components such as an identification code and password.	AGI Creator Security functions are based on the combination Username/ Password.

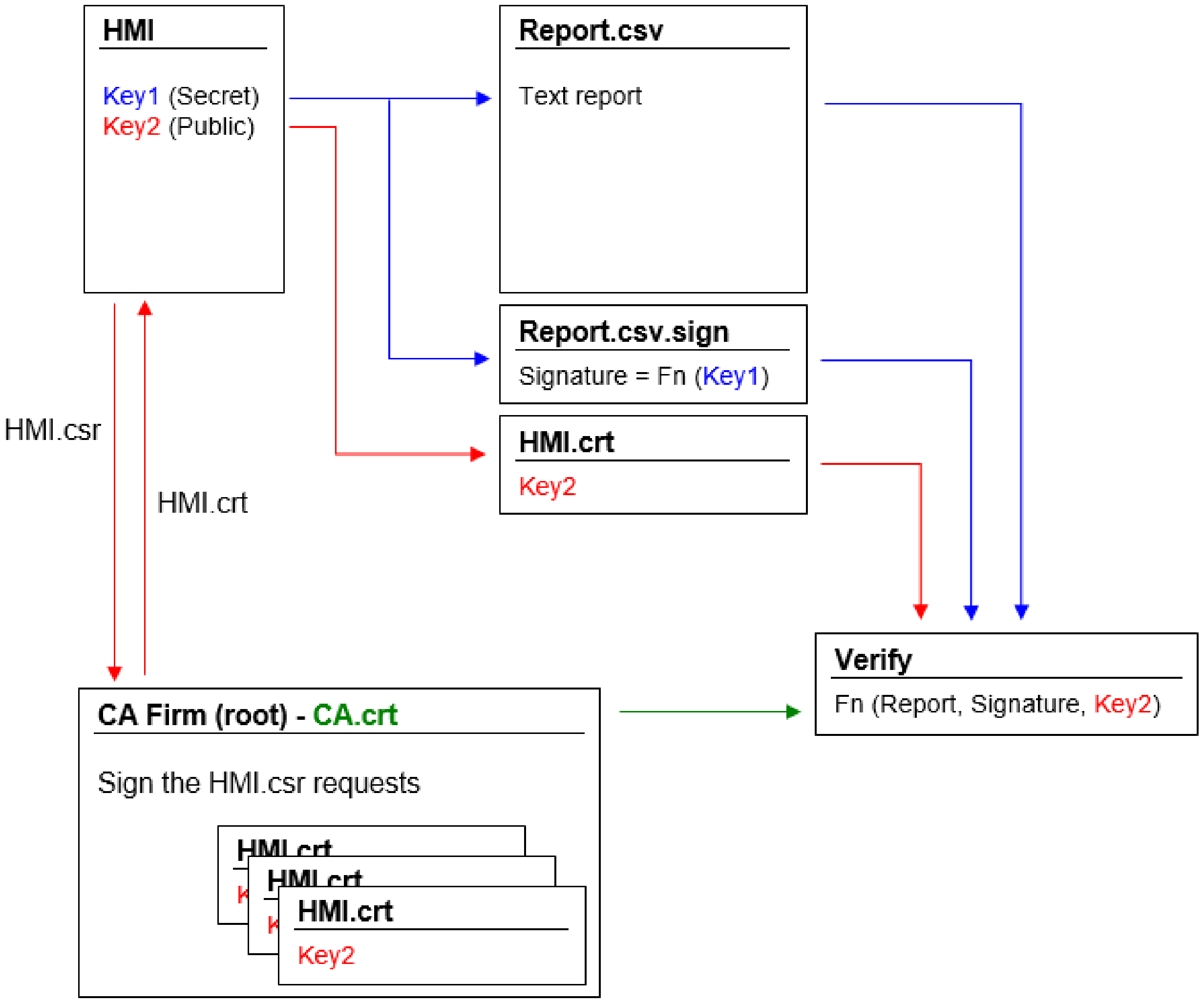
Chapter	Description	AGI Creator compliance level (v2.8)
	(i) When an individual executes a series of signings during a single, continuous period of controlled system access, the first signing shall be executed using all electronic signature components; subsequent signings shall be executed using at least 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: - "User management and passwords" on page 333 - "Electronic Signature" on page 347
	(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: - "Configuring users" on page 341
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.	AGI Creator 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: - "Configuring users" on page 341
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 suitable, rigorous controls.	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: - "User management actions" on page 217 - "Configuring users" on page 341
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	Failed logging attempts are logged to audit trail.

Chapter	Description	AGI Creator compliance level (v2.8)
	use to the system security unit, and, as appropriate, to organizational management.	
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 208 or "PrintGraphicReport" on page 192 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 327)
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 328)

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 547).

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.

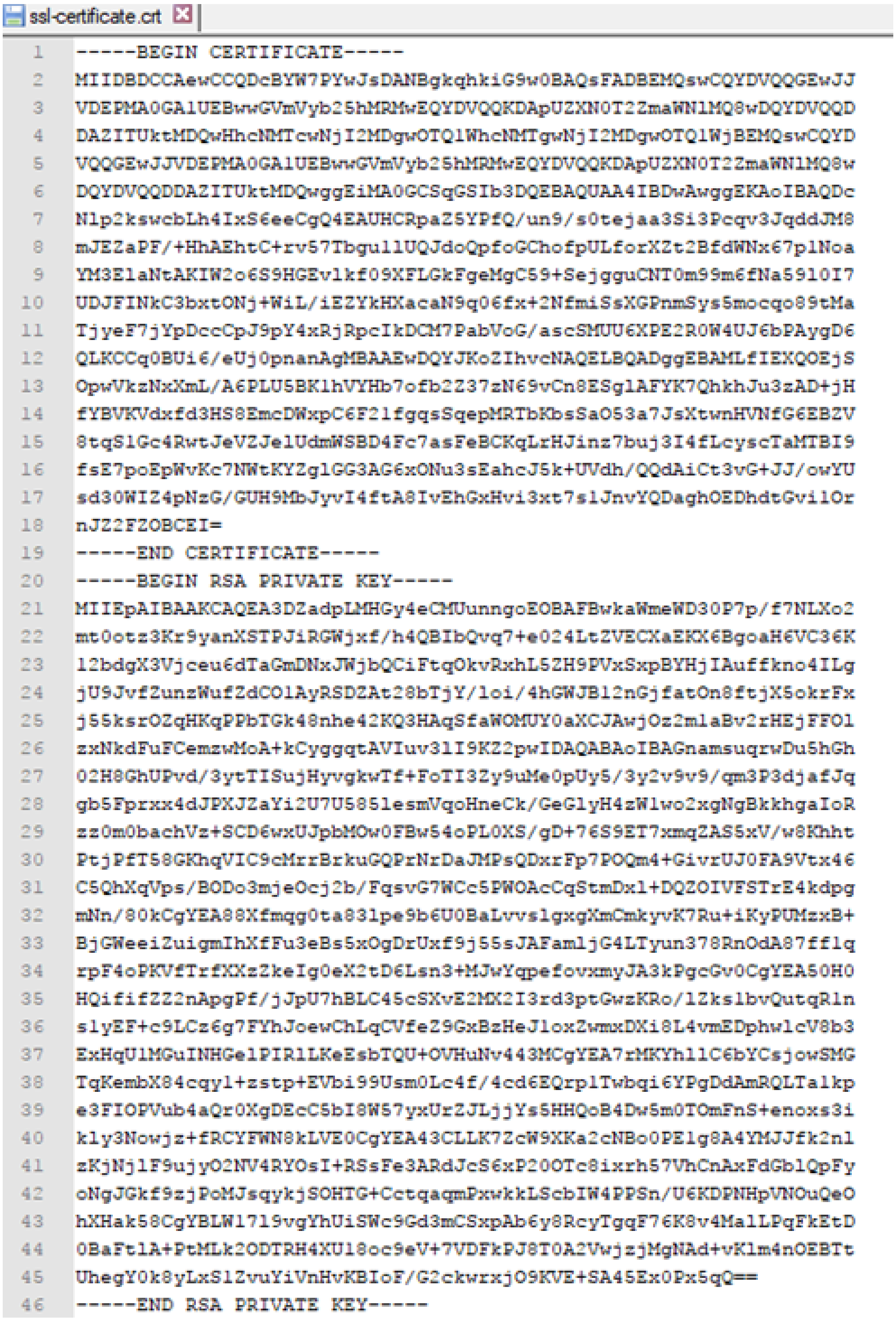
After retrieving the "certificate signed request" file to send to the certificate authority, be sure to never regenerate a new certificate otherwise the internal private key associated with the certificate send to the authority will be lost.

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).



```
-----BEGIN CERTIFICATE-----
MIIDBDCCAewCCQDcBYW7PYwJsDANBgkqhkiG9w0BAQsFADBEMQswCQYDVQQGEwJJ
VDEPMA0GA1UEBwwGVmVyb25hMRMwEQYDVQQKDApUZXN0T2ZmaWN1MQ8wDQYDVQQD
DAZITUktMDQwHhcNMTcwNjI2MDgwOTQ1WhcNMTgwNjI2MDgwOTQ1WjBEMQswCQYD
VQQGEwJJVDEPMA0GA1UEBwwGVmVyb25hMRMwEQYDVQQKDApUZXN0T2ZmaWN1MQ8w
DQYDVQQDDAZITUktMDQwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDc
N1p2kswcbLh4IxS6eeCgQ4EAUHCRpaZ5YPfQ/un9/s0tejaa3Si3Pcqv3JqddJM8
mJEZaPF/+HhAEhtC+rv57TbgullUQJdoQpfoGChofpULforXZt2BfdWNx67p1Noa
YM3ElaNtAKIW2o6S9HGEv1kf09XFLGkFgeMgC59+SejgguCNT0m99m6fNa5910I7
UDJFINkC3bxtONj+WiL/iEZYkHXacaN9q06fx+2NfmiSsXGPnmSys5mocqo89tMa
TjyeF7jYpDccCpJ9pY4xRjRpcIkDCM7PabVoG/ascSMUU6XPE2R0W4UJ6bPAygD6
QLKCCq0BUi6/eUj0pnanAgMBAAEwDQYJKoZIhvcNAQELBQADggEBAMLfIEXQOEjS
OpwVkzNxXmL/A6PLU5BK1hVYHb7ofb2Z37zN69vCn8ESglAFYK7QhkhJu3zAD+jH
fYBVKVdxfd3HS8EmcDWxpC6F21fgqsSqepMRTbKbsSaO53a7JsXtwnHVNfG6EBZV
8tqS1Gc4RwtJeVZJe1UdmWSBD4Fc7asFeBCKqLrHJinz7buj3I4fLcyscTaMTBI9
fsE7poEpWvKc7NWtKYZglGG3AG6xONu3sEahcJ5k+UVdh/QQdAiCt3vG+JJ/owYU
sd30WIZ4pNzG/GUH9MbJyvI4ftA8IvEhGxHvi3xt7slJnvYQDaghOEDhdtGvilOr
nJZ2FZOBCEI=
-----END CERTIFICATE-----
-----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCAQEA3DZadpLMHGy4eCMUunngoEOBAFBwkaWmeWD30P7p/f7NLXo2
mt0otz3Kr9yanXSTPJiRGWjxf/h4QBIbQvq7+e024LtZVECXaEKX6BgoaH6VC36K
l2bdgX3Vjceu6dTaGmDNxJWjbQCiFtqOkvRxhL5ZH9PVxSxpBYHjIAuffkno4ILg
jU9JvfZunzWufZdCOlAyRSDZAt28bTjY/loi/4hGWJBl2nGjfatOn8ftjX5okrFx
j55ksrOZqHKqPPbTGk48nhe42KQ3HAqSfaWOMUY0aXCJAwjOz2mlaBv2rHEjFFOl
zxNkdFuFCemzwMoA+kCyggqtAVIuv31I9KZ2pwIDAQABAoIBAGnamsuqrwDu5hGh
02H8GhUPvd/3ytTISujHyvgkwTf+FoTI3Zy9uMe0pUy5/3y2v9v9/qm3P3djafJq
gb5Fprxx4dJPXJZaYi2U7U585lesmVqoHneCk/GeGlyH4zWlwo2xgNgBkkhgaIoR
zz0m0bachVz+SCD6wxUJpbMOw0FBw54oPL0XS/gD+76S9ET7xmqZAS5xV/w8Khht
PtjPfT58GKhqVIC9cMrrBrkuGQPrNrDaJMPsQDxrFp7POQm4+GivrUJ0FA9Vtx46
C5QhXqVps/BODo3mjeOcj2b/FqsvG7WCc5PWOAcCqStmDxl+DQZOIVFSTrE4kdpg
mNn/80kCgYEA88Xfmqg0ta831pe9b6U0BaLvvslgxgXmCmkyvK7Ru+iKyPUMzxB+
BjGWeeiZuigmIhXfFu3eBs5xOgDrUxf9j55sJAFamljG4LTyun378RnOdA87fflq
rpF4oPKVfTrfXXzZkeIg0eX2tD6Lsn3+MJwYqpefovxmyJA3kPgcGv0CgYEA50H0
HQififZZ2nApgPf/jJpU7hBLC45cSXvE2MX2I3rd3ptGwzKRo/lZkslbvQutqRln
slyEF+c9LCz6g7FYhJoewChLqCVfeZ9GxBzHeJloxZwmxDXi8L4vmEDphwlcV8b3
ExHqUlMGuINHGelPIRlLKeEsbTQU+OVHuNv443MCgYEA7rMKYhllC6bYCsjowSMG
TqKembX84cqyl+zstp+EVbi99Usm0Lc4f/4cd6EQrplTwbqi6YPgDdAmRQLTalkp
e3FIOPVub4aQr0XgDEcC5bI8W57yxUrZJLjjYs5HHQoB4Dw5m0TOmFnS+enoxs3i
kly3Nowjz+fRCYFWN8kLVE0CgYEA43CLLK7ZcW9XKa2cNBo0PElg8A4YMJJfk2nl
zKjNjlF9ujyO2NV4RYOsI+RSsFe3ARdJcS6xP20OTc8ixrh57VhCnAxFdGblQpFy
oNgJGkf9zjPoMJsqykjSOHTG+CctqaqmPxwkkLScbIW4PPSn/U6KDPNHpVNOuQeO
hXHak58CgYBLW1719vgYhUiSWc9Gd3mCSxpAb6y8RcyTgqF76K8v4MalLPqFkEtD
0BaFtlA+PtMLk2ODTRH4XU18oc9eV+7VDFkPJ8T0A2VwjzjMgNAd+vKlm4nOEBTt
UhegY0k8yLxSlZvuYiVnHvKBIoF/G2ckwrxjO9KVE+SA45Ex0Px5qQ==
-----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.

AGI PC Runtime

When using AGI PC Runtime the certificate files can be found inside the folder:
%AppData%\DEIF A/S\<Version>\server\config\ssl-certificate

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 208)
- How to provide an x.509 Certificate to Linux devices ("x.509 Certificate" on page 563)

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.

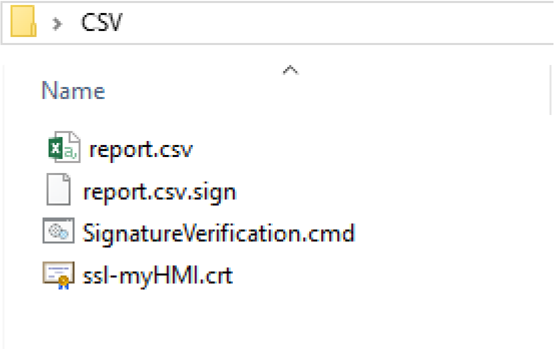
How to verify the report's signature using the public OpenSSL library

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

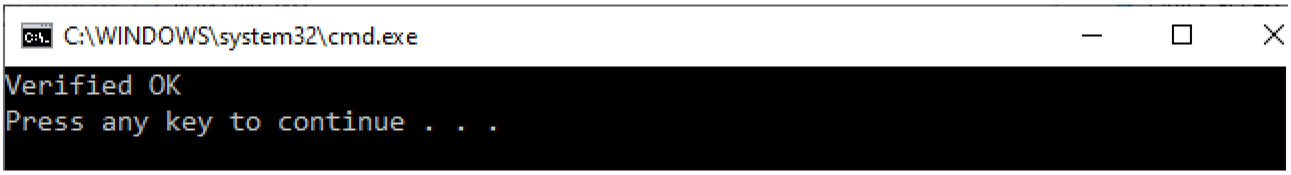


File: *SignatureVerification.cmd*

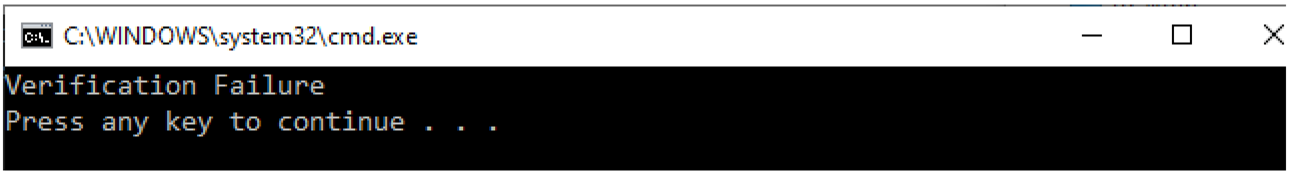
```
@echo off
set OpenSSL="C:\Program Files (x86)\OpenSSL-Win32\bin\openssl.exe"
set FileToCheck=Report.csv
set hmiCertificate=ssl-myHMI.crt

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



```
Verified OK
Press any key to continue . . .
```



```
Verification Failure
Press any key to continue . . .
```

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.

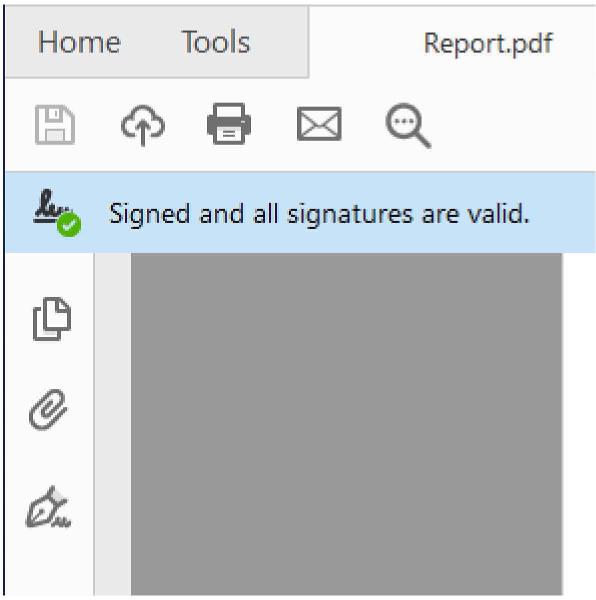
See also:

- The PrintGraficReport parameters ("PrintGraphicReport" on page 192)
- How to provide an x.509 Certificate to Linux devices ("x.509 Certificate" on page 563)

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.

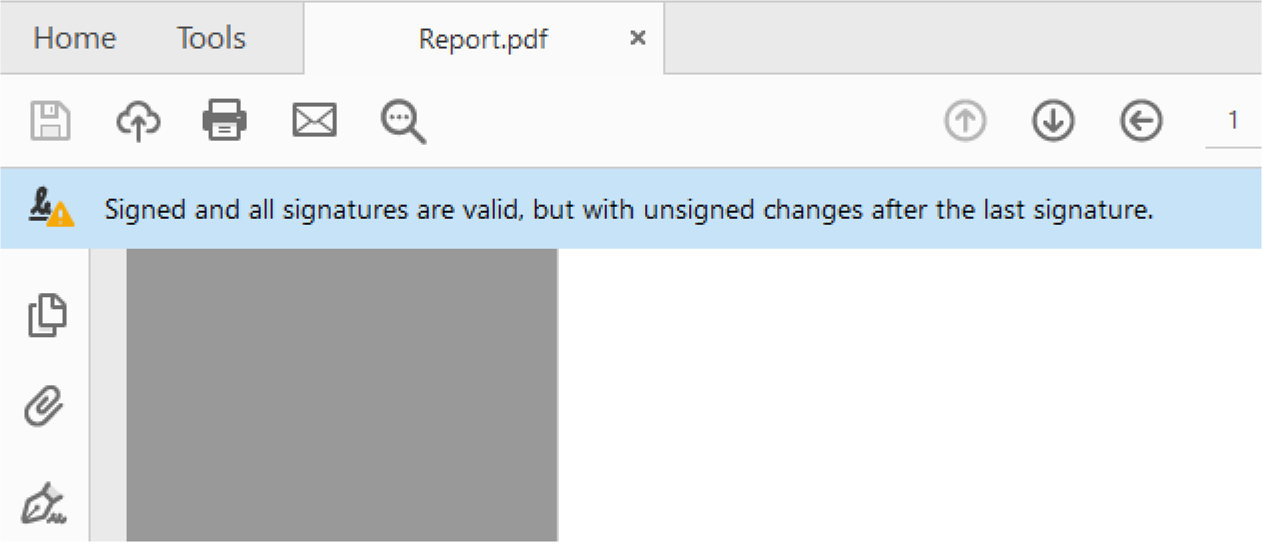
x.509 certificate signed from a Certificate Authority

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.



Signed and all signatures are valid.

If the document has been modified, it will be highlighted with a different message.

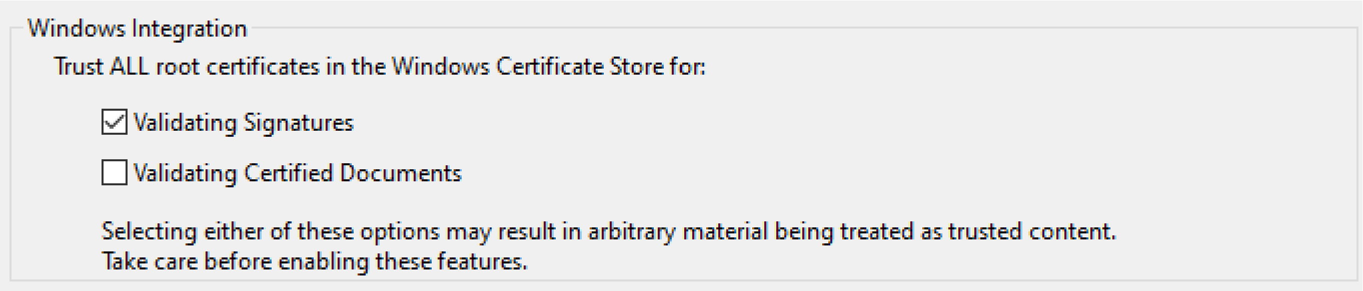


Signed and all signatures are valid, but with unsigned changes after the last signature.

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

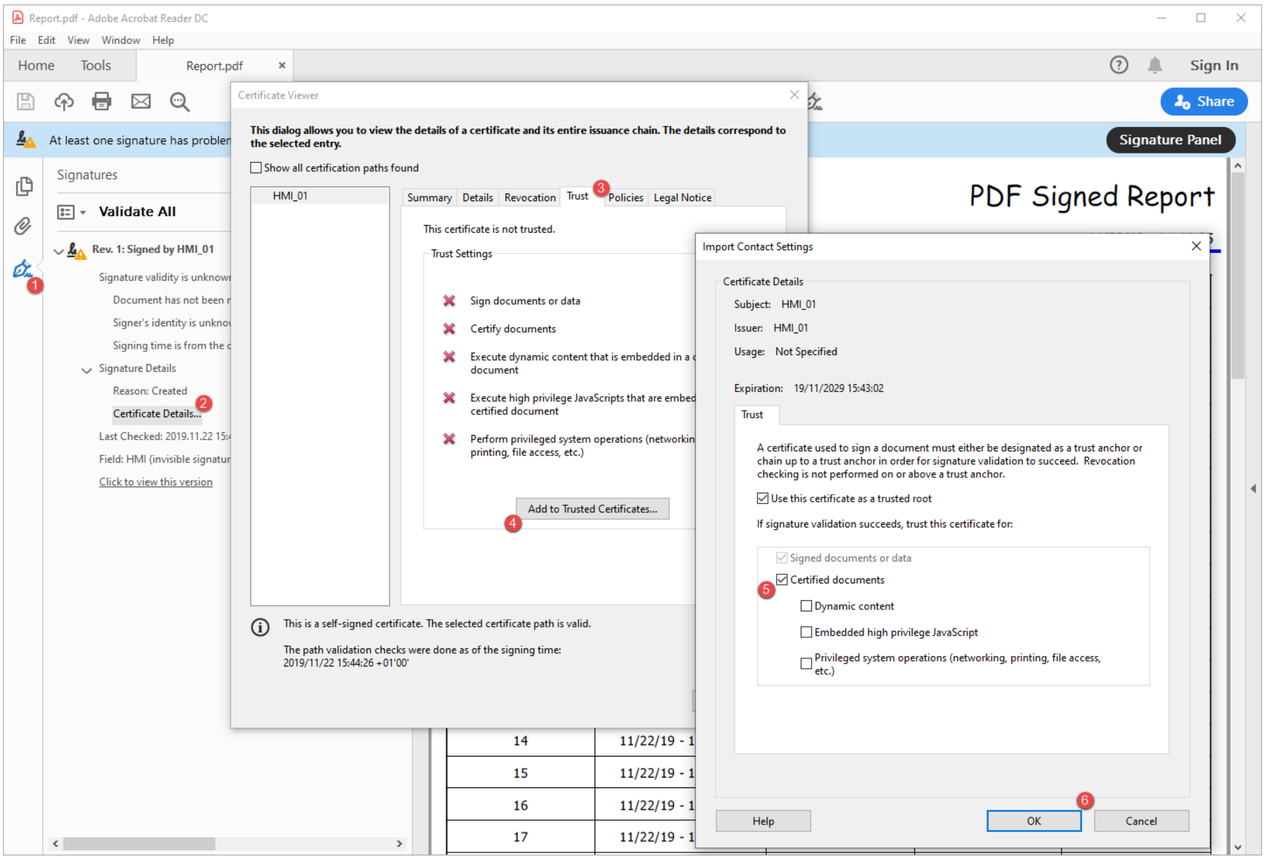
Selecting either of these options may result in arbitrary material being treated as trusted content.
Take care before enabling these features.

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:



At least one signature has problen

Rev. 1: Signed by HMI_01

Signature validity is unknowr

Document has not been r

Signer's identity is unknov

Signing time is from the c

Signature Details

Reason: Created

Certificate Details...

Last Checked: 2019.11.22 15:4

Field: HMI (invisible signatur

Click to view this version

This dialog allows you to view the details of a certificate and its entire issuance chain. The details correspond to the selected entry.

□ Show all certification paths found

This certificate is not trusted.

Trust Settings

☒ Sign documents or data

☒ Certify documents

☒ Execute dynamic content that is embedded in a c document

☒ Execute high privilege JavaScripts that are embed certified document

☒ Perform privileged system operations (networkin printing, file access, etc.)

This is a self-signed certificate. The selected certificate path is valid.

The path validation checks were done as of the signing time:
2019/11/22 15:44:26 +01'00'

Certificate Details

Subject: HMI_01

Issuer: HMI_01

Usage: Not Specified

Expiration: 19/11/2029 15:43:02

Trust

A certificate used to sign a document must either be designated as a trust anchor or chain up to a trust anchor in order for signature validation to succeed. Revocation checking is not performed on or above a trust anchor.

☑ Use this certificate as a trusted root

If signature validation succeeds, trust this certificate for:

☑ Signed documents or data

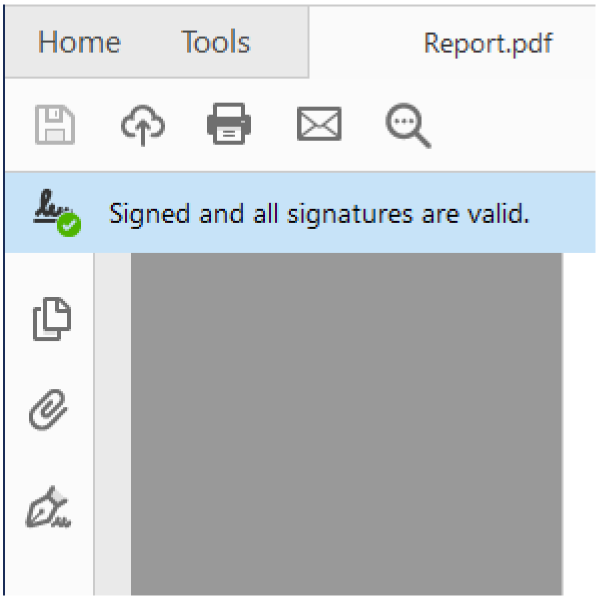
☑ Certified documents

□ Dynamic content

□ Embedded high privilege JavaScript

□ Privileged system operations (networking, printing, file access, etc.)

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.



Signed and all signatures are valid.

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

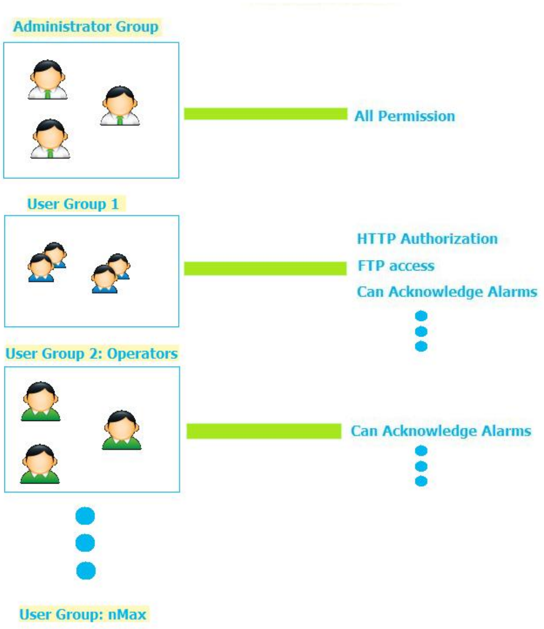
You can restrict access to various widgets and operations by configuring users, users groups and assigning specific authorizations to each group.

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.

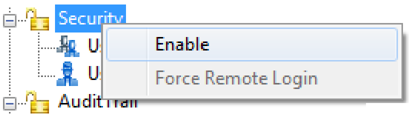
By organizing permissions and groups you can define the security options of a project.



Enable/disable security management

Path: ***ProjectView****> right-click* ***Security****>* ***Enable***

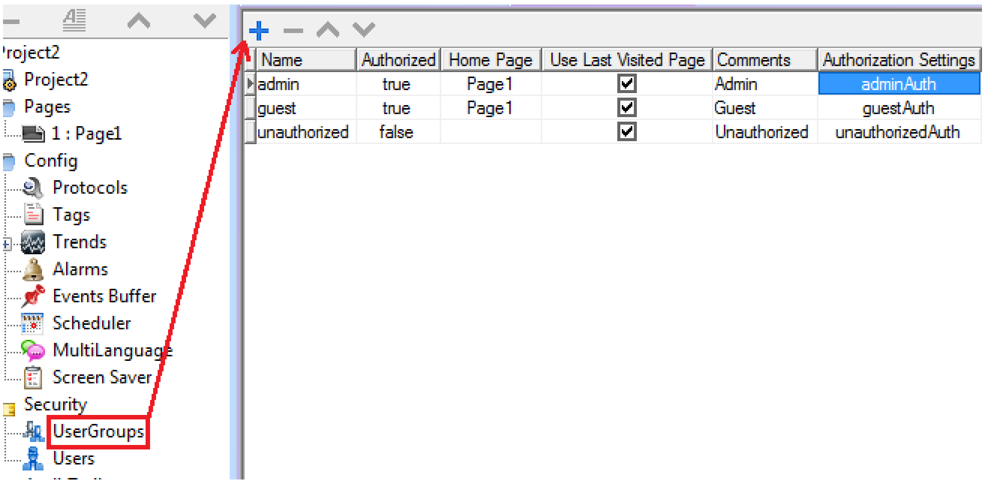
The padlock symbol indicates whether the function is enabled or disabled.



Important: Security settings are effective only if the security function is enabled.

Configuring groups and authorizations

Path: ***ProjectView****>* ***Security****> double-click* ***UserGroups***



Name	Authorized	Home Page	Use Last Visited Page	Comments	Authorization Settings
admin	true	Page1	☑	Admin	adminAuth
guest	true	Page1	☑	Guest	guestAuth
unauthorized	false		☑	Unauthorized	unauthorizedAuth

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.

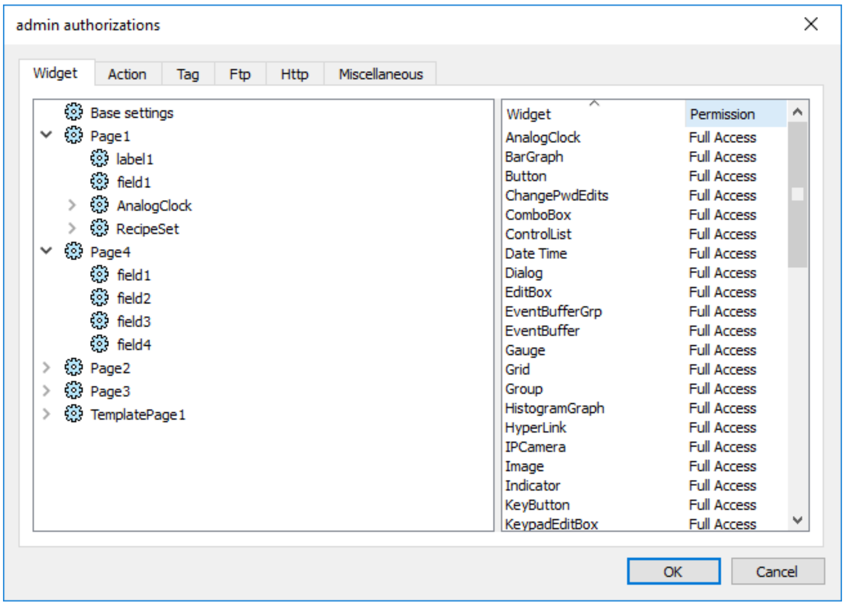
Parameter	Description
Name	Name of users group
Authorized	Authorization granted
Home Page	Page displayed when users belonging to this group log in
Use Last Visited Page	When selected, the last page displayed by the previous user will be displayed when users belonging to this group log in

Parameter	Description
Comments	Any comment or description for the group
Authorization Settings	Opens the Admin Authorization dialog to set access permissions. See "Modifying access permissions" 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.



- Base settings
- Page1
  - label1
  - field1
  - AnalogClock
  - RecipeSet
- Page4
  - field1
  - field2
  - field3
  - field4
- Page2
- Page3
- TemplatePage1

Widget	Permission
AnalogClock	Full Access
BarGraph	Full Access
Button	Full Access
ChangePwdEdits	Full Access
ComboBox	Full Access
ControlList	Full Access
Date Time	Full Access
Dialog	Full Access
EditBox	Full Access
EventBufferGrp	Full Access
EventBuffer	Full Access
Gauge	Full Access
Grid	Full Access
Group	Full Access
HistogramGraph	Full Access
HyperLink	Full Access
IPCamera	Full Access
Image	Full Access
Indicator	Full Access
KeyButton	Full Access
KeypadEditBox	Full Access

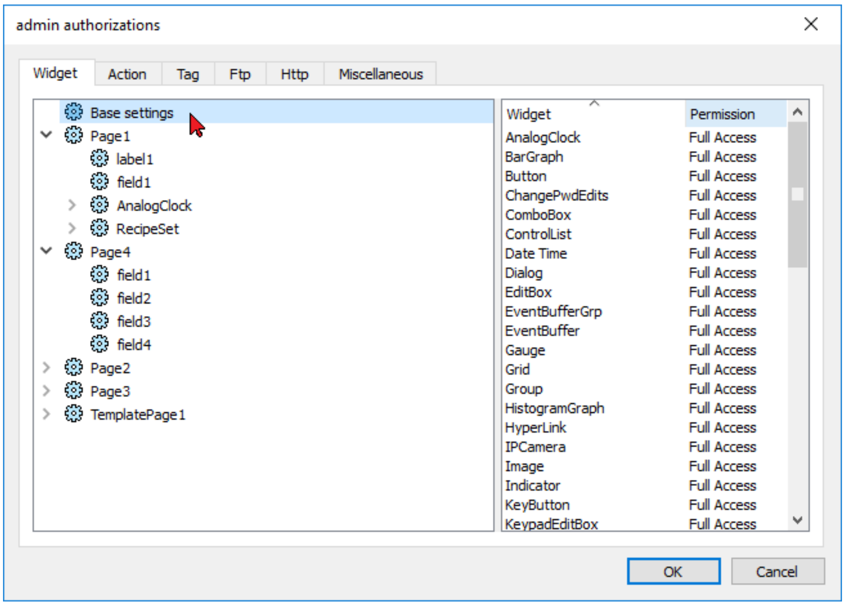
Widget permissions

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



- Base settings
- Page1
  - label1
  - field1
  - AnalogClock
  - RecipeSet
- Page4
  - field1
  - field2
  - field3
  - field4
- Page2
- Page3
- TemplatePage1

Widget	Permission
AnalogClock	Full Access
BarGraph	Full Access
Button	Full Access
ChangePwdEdits	Full Access
ComboBox	Full Access
ControlList	Full Access
Date Time	Full Access
Dialog	Full Access
EditBox	Full Access
EventBufferGrp	Full Access
EventBuffer	Full Access
Gauge	Full Access
Grid	Full Access
Group	Full Access
HistogramGraph	Full Access
HyperLink	Full Access
IPCamera	Full Access
Image	Full Access
Indicator	Full Access
KeyButton	Full Access
KeypadEditBox	Full Access

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

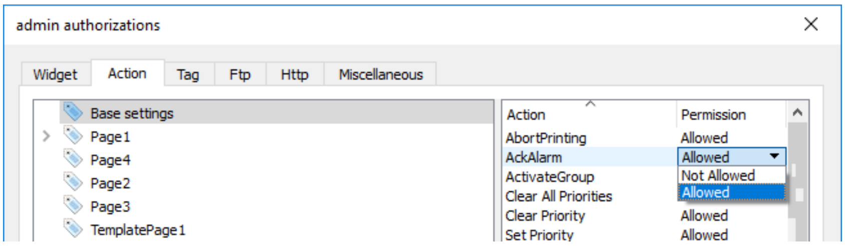
This allows you to specify exceptions for an action or a widget directly from the page view.

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 340 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**.



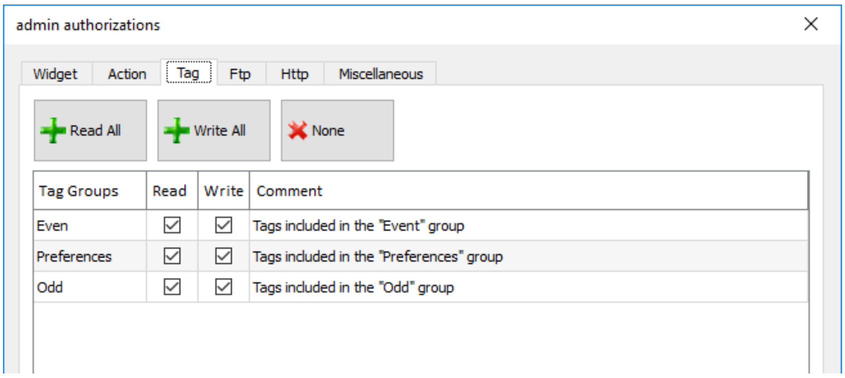
- Base settings
- Page1
- Page4
- Page2
- Page3
- TemplatePage1

Action	Permission
AbortPrinting	Allowed
AckAlarm	Allowed
ActivateGroup	Not Allowed
Clear All Priorities	Allowed
Clear Priority	Allowed
Set Priority	Allowed

Action permissions can be modified directly from the project page. See "Assigning widget permissions from page view" on page 340 for details.

Tag permissions

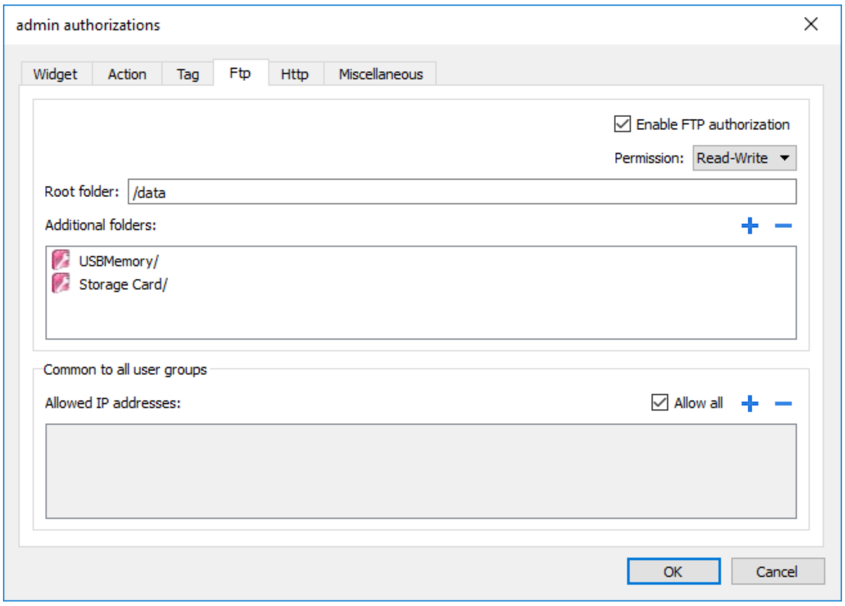
For each group of tags, you can define the Read/Write access rights



Tag Groups	Read	Write	Comment
Even	☑	☑	Tags included in the "Event" group
Preferences	☑	☑	Tags included in the "Preferences" group
Odd	☑	☑	Tags included in the "Odd" group

FTP authorizations

In the **Ftp** tab you can set specific authorizations for the FTP server.



☑ Enable FTP authorization

Permission: Read-Write

Root folder: /data

Additional folders:

- USBMemory/
- Storage Card/

Common to all user groups

Allowed IP addresses: ☑ Allow all

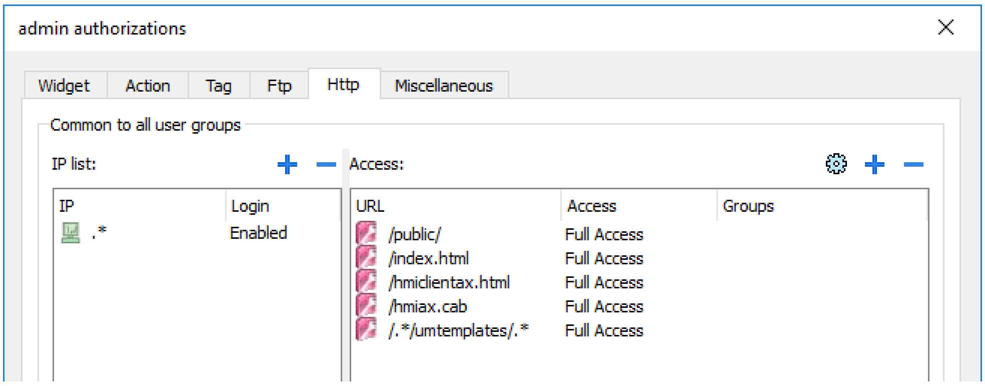
Element	Description
Enable FTP authorization	Enables the FTP function for the specific group
Permission	Type of permission: - **Read-Only** - **Read-Write**
Root Folder	Folder to be used as root for FTP access. This is a relative path.
Additional folder	Extra folders to be used as root for FTP access (for example, on USB drive or SD card)
Allowed IP Addresses	List of IP addresses from which FTP connection can be accepted. **This setting is common to all users groups.**

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.



IP	Login
.*	Enabled

URL	Access	Groups
/public/	Full Access	
/index.html	Full Access	
/hmiclientax.html	Full Access	
/hmiax.cab	Full Access	
/.*/umtemplates/.*	Full Access	

Element	Description
IP list	IP addresses authorized to access the HTTP server. By default the login is required from any IP address (IP=.*, Login=Enabled).
Login	When disabled, the username and password are not required.
Access limits	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 344 for details.

Force Remote Login	Default Access to workspace	Access limits
-	Full	-
Disable	Full	Can be used to block access to some files/folders or to require authorization
Enable	No Access	Can be used to open access to files/folders

Important: This setting is common to all users groups.

Adding an HTTP configuration

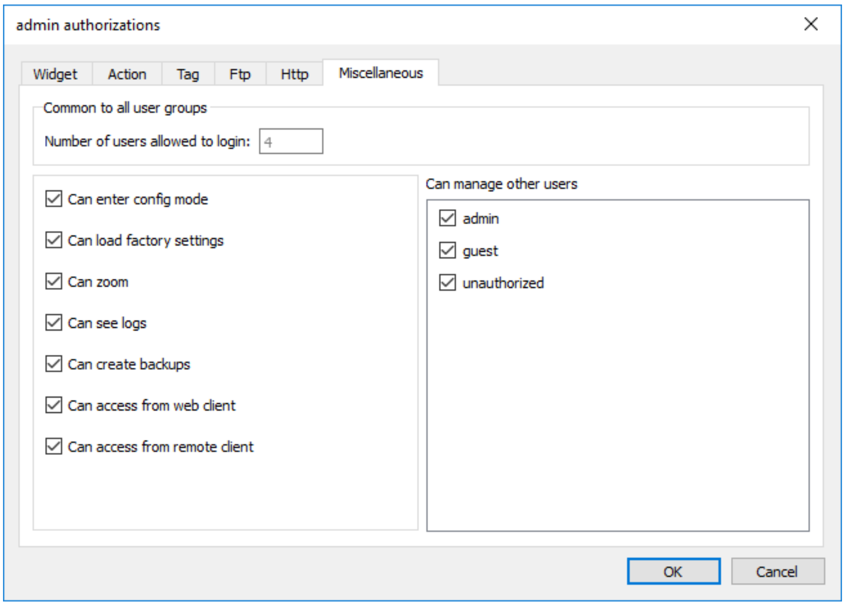
To add and configure a new access click **+**: the **Access limits** dialog is displayed.

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.



Common to all user groups

Number of users allowed to login: 4

☑ 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

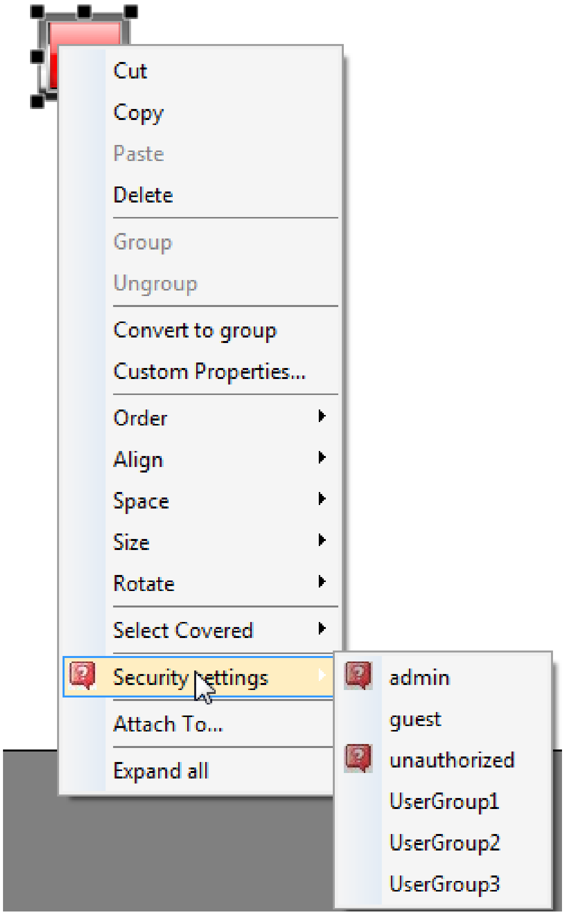
Option	Description
Can enter config mode	Enables switching from runtime to configuration mode. Normally used for maintenance.
Can load factory settings	Restores factory settings.
Can zoom	Enables zoom in/out in context menu at runtime

Option	Description
Can see log	Allows user to see logs at runtime
Can create backup	Allows user to backup project.
Can access from web client	Enables connecting from a web client
Can access from remote client	Enables connecting from AGI Client
Can manage other users	Gives super user privileges at runtime to manage the select groups. Allows adding, deleting and modifying users' permissions.
Number of users allowed to login	Maximum number of users that can be connected to the HMI Runtime at the same time. **This setting is common to all users groups.**

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.

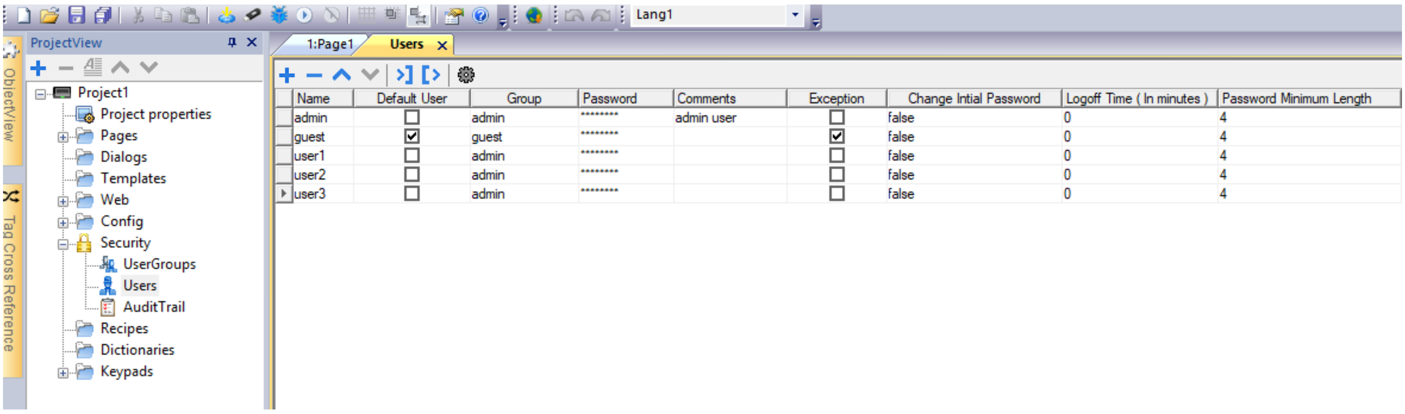


See "Modifying access permissions" on page 335 for details.

Configuring users

*Path: **ProjectView**> **Security**> double-click **Users***

In the Users editor, click **+** to add a user: one row is added to the table.



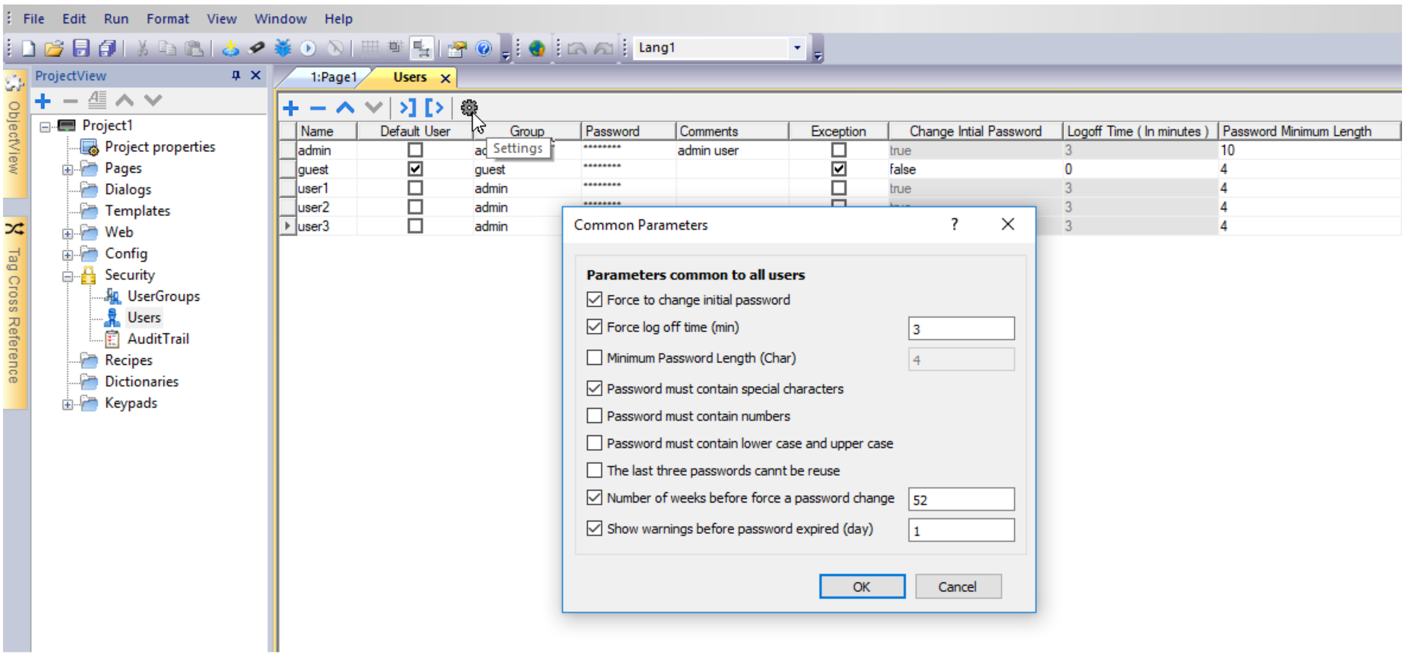
Name	Default User	Group	Password	Comments	Exception	Change Intial Password	Logoff Time (In minutes)	Password Minimum Length
admin	□	admin	*********	admin user	□	false	0	4
guest	☑	guest	*********		☑	false	0	4
user1	□	admin	*********		□	false	0	4
user2	□	admin	*********		□	false	0	4
user3	□	admin	*********		□	false	0	4

Parameter	Description
Name	User name
Default User	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
Group	User group
Password	User password. Note that for security reasons the password will never be displayed **Passwords are encrypted and cannot be retrieved not even for specialized technicians** 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.
Comments	Further user description
Exception	Allows to change the values forced from the User Settings parameters
Change Initial Password	This user is forced to change his password at first log in
Logoff time (minutes)	Minutes of inactivity after which the user is logged off. Set to 0 to disable
Password minimum length	Minimum length of password
Must contain special characters	Password must contain at least one special character
Must contain numbers	Password must contain at least one numeric digit
Must contain lower case and upper case	Password must contain lower case and upper case
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.



Name	Default User	Group	Password	Comments	Exception	Change Intial Password	Logoff Time (In minutes)	Password Minimum Length
admin	□	ad	********	admin user	□	true	3	10
guest	☑	guest	********		☑	false	0	4
user1	□	admin	********		□	true	3	4
user2	□	admin	********				3	4
user3	□	admin					3	4

Common Parameters

Parameters common to all users

- ☑ Force to change initial password
- ☑ Force log off time (min) 3
- □ Minimum Password Length (Char) 4
- ☑ Password must contain special characters
- □ Password must contain numbers
- □ Password must contain lower case and upper case
- □ The last three passwords cannt be reuse
- ☑ Number of weeks before force a password change 52
- ☑ Show warnings before password expired (day) 1

OK Cancel

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 217 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 217 for details on the actions that can be executed on users.

Removing user data

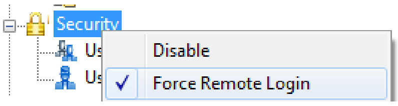
All the user information modified at runtime is stored in dedicated files. To remove these dynamic files and all the changes applied to user configuration at runtime you can:

- on HMI Runtime: execute the action DeleteUMDynamicFile
- with AGI Creator: 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**> **Force Remote Login***



Select this option to force user to log in when using remote access viaAGI 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 334 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 335 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.

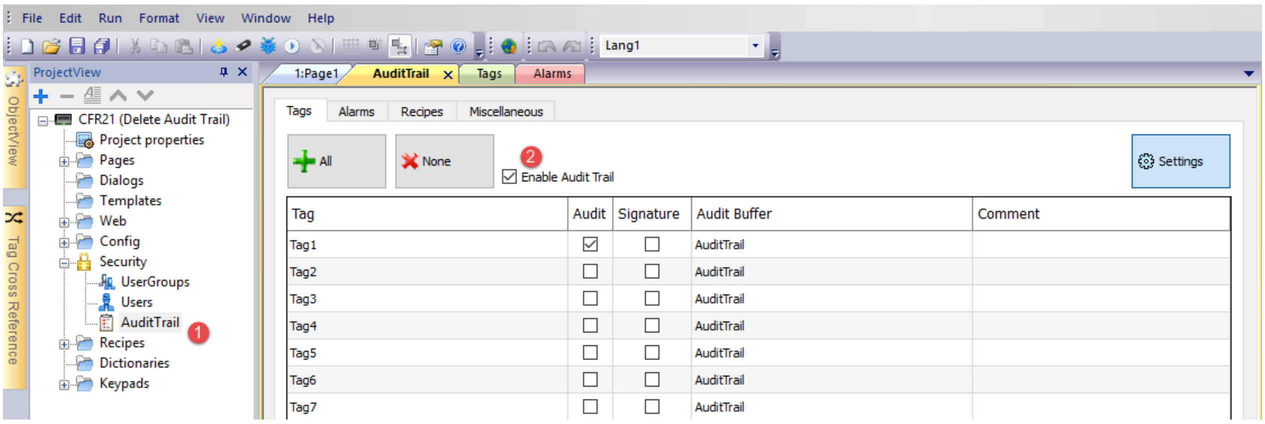
This function provides process tracking and user identification with time stamp for events.

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)



☑ Enable Audit Trail

Tag	Audit	Signature	Audit Buffer	Comment
Tag1	☑	□	AuditTrail	
Tag2	□	□	AuditTrail	
Tag3	□	□	AuditTrail	
Tag4	□	□	AuditTrail	
Tag5	□	□	AuditTrail	
Tag6	□	□	AuditTrail	
Tag7	□	□	AuditTrail	

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
Audit	Enable tracking of the selected resource
Signature	The user password is required before allowing the resource to be modified from the user (see "Electronic Signature" on the facing page to additional information)
Audit Buffer	Internal buffer where store the related audit events (see "Events Buffer" on page 257 to additional information)
Comment	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 210 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.

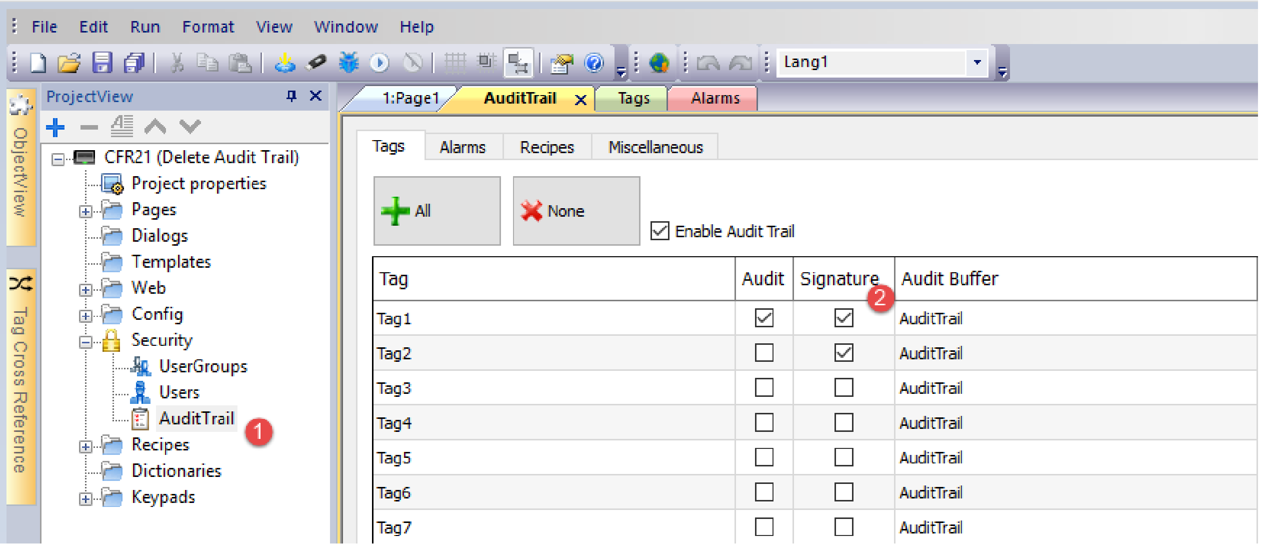
Backup audit events

From the "Events Buffer" on page 257 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.

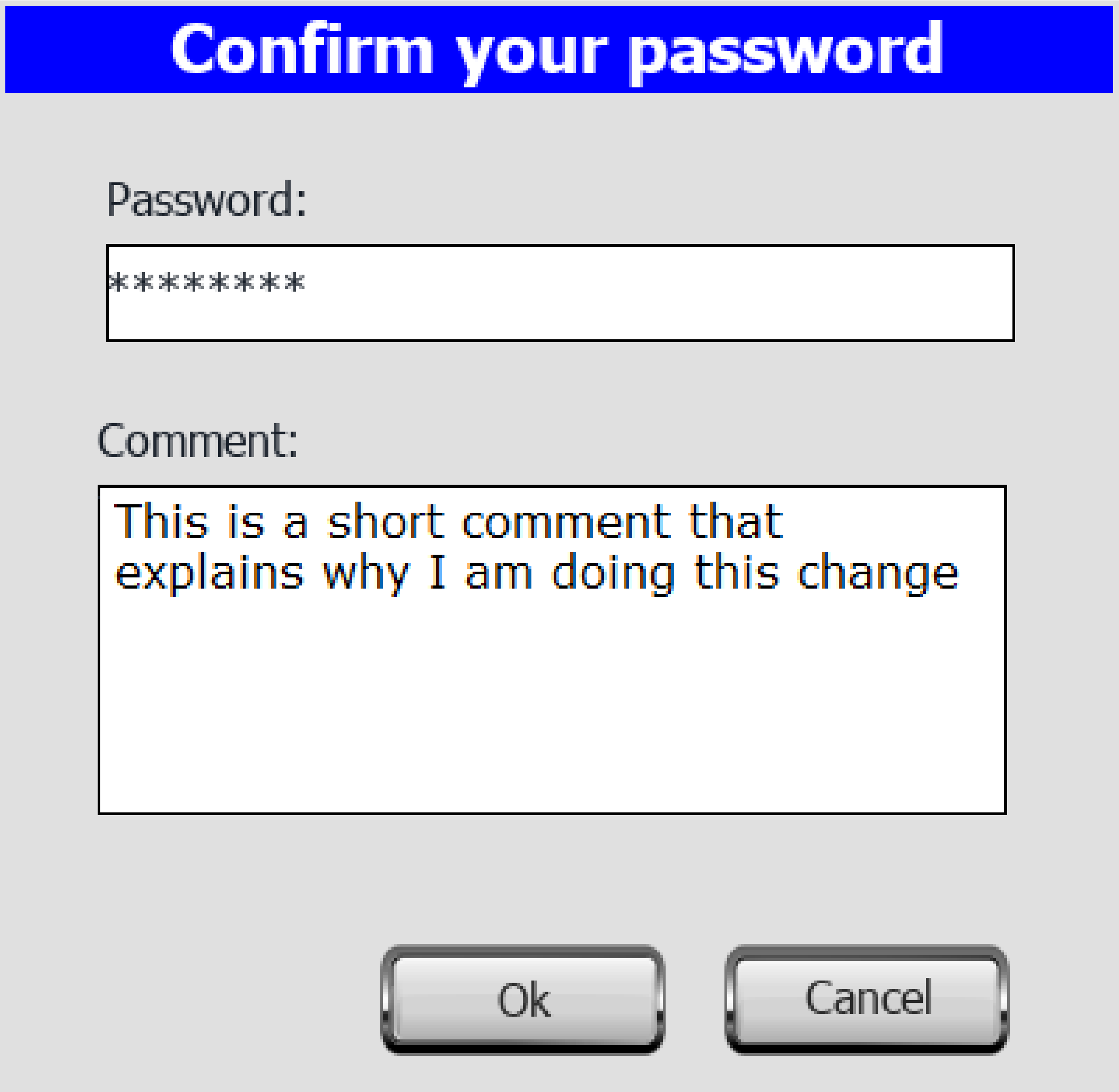
*Path: **ProjectView**> **Security** > double-click **AuditTrail***



☑ Enable Audit Trail

Tag	Audit	Signature	Audit Buffer
Tag1	☑	☑	AuditTrail
Tag2	□	☑	AuditTrail
Tag3	□	□	AuditTrail
Tag4	□	□	AuditTrail
Tag5	□	□	AuditTrail
Tag6	□	□	AuditTrail
Tag7	□	□	AuditTrail

The user password is required before allowing the resource to be modified by the user



Confirm your password

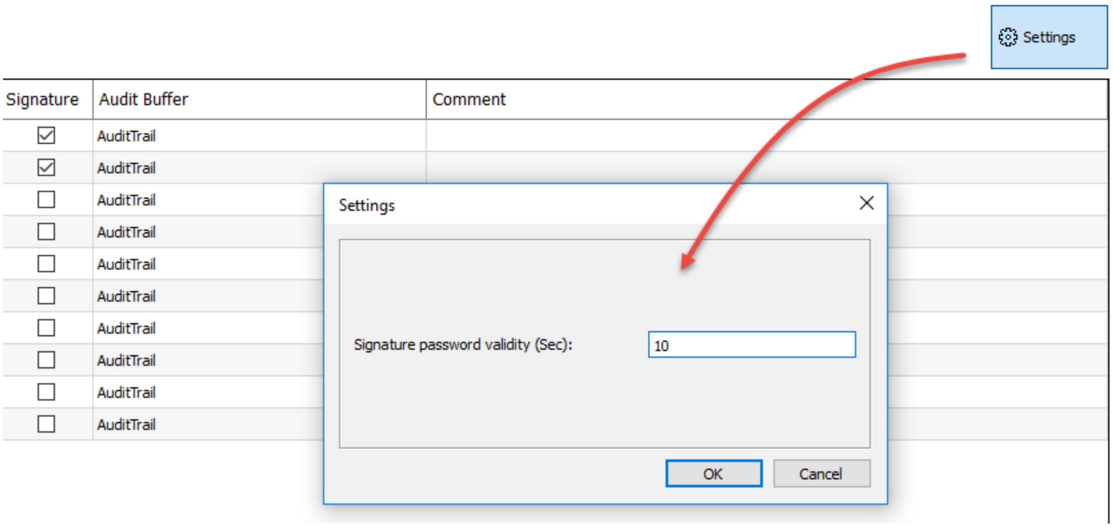
Password:

****

Comment:

This is a short comment that explains why I am doing this change

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.



Signature	Audit Buffer	Comment
☑	AuditTrail	
☑	AuditTrail	
□	AuditTrail	
□	AuditTrail	
□	AuditTrail	
□	AuditTrail	
□	AuditTrail	
□	AuditTrail	
□	AuditTrail	
□	AuditTrail	

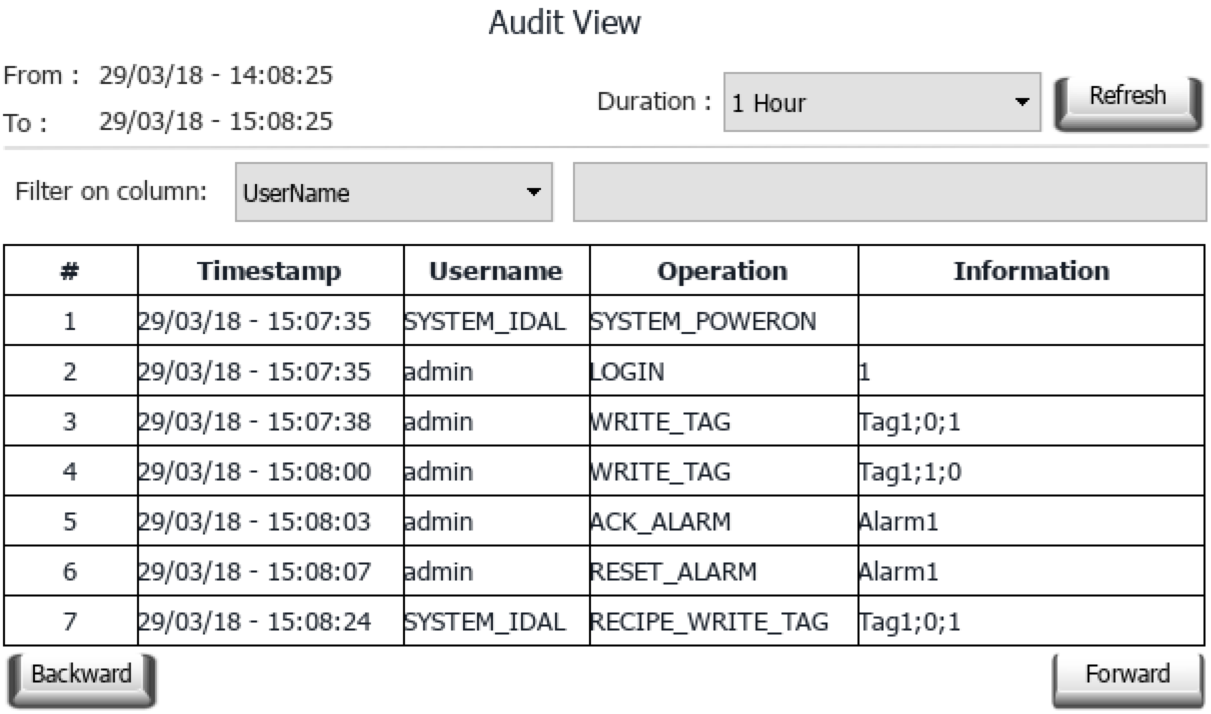
Settings

Signature password validity (Sec): 10

Table audit widget

Path: **Widget Gallery**> **Basic**> **Audit Tables**

Display contents of the audit trail inside a widget



Audit View

From : 29/03/18 - 14:08:25

To : 29/03/18 - 15:08:25

Duration : 1 Hour

Filter on column: UserName

#	Timestamp	Username	Operation	Information
1	29/03/18 - 15:07:35	SYSTEM_IDAL	SYSTEM_POWERON	
2	29/03/18 - 15:07:35	admin	LOGIN	1
3	29/03/18 - 15:07:38	admin	WRITE_TAG	Tag1;0;1
4	29/03/18 - 15:08:00	admin	WRITE_TAG	Tag1;1;0
5	29/03/18 - 15:08:03	admin	ACK_ALARM	Alarm1
6	29/03/18 - 15:08:07	admin	RESET_ALARM	Alarm1
7	29/03/18 - 15:08:24	SYSTEM_IDAL	RECIPE_WRITE_TAG	Tag1;0;1

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
AuditBuffer	Event Buffer from which the event list is retrieved (see "Events Buffer" on page 257)
Heading	Heading label
Default Duration	Initial value of time window to show
Time Spec	Time format: • **Local** = show the time values of the HMI device. • **Global** = show the time values using UTC format.
Date Format	Select the Date and Time format

Parameter	Description
Filter List	Labels to show in filter column selection
Table Layout	Defines the characteristics of the scroll bar and allows to remove the header of the table

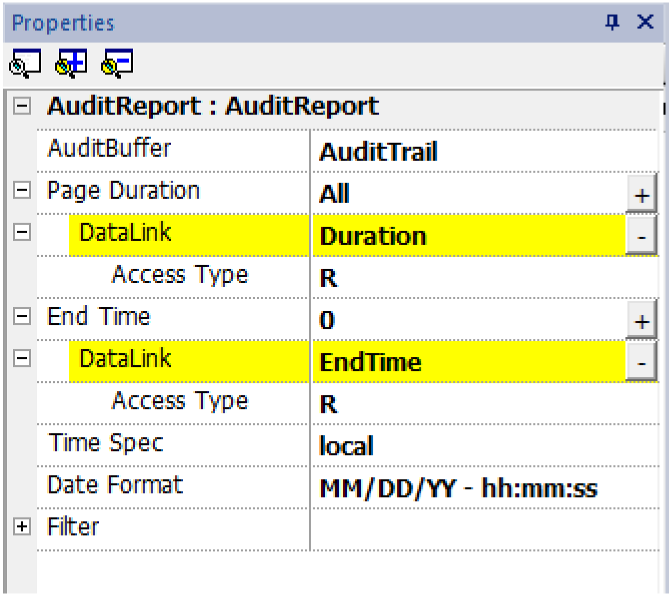
Printing audit table

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

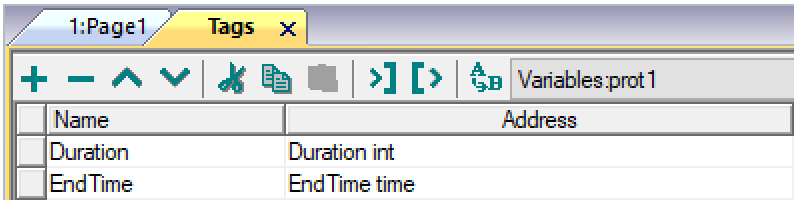
"Page Duration" with "End Time" define the piece of the audit buffer to print.



Properties

AuditReport : AuditReport

AuditBuffer	AuditTrail
Page Duration	All
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	

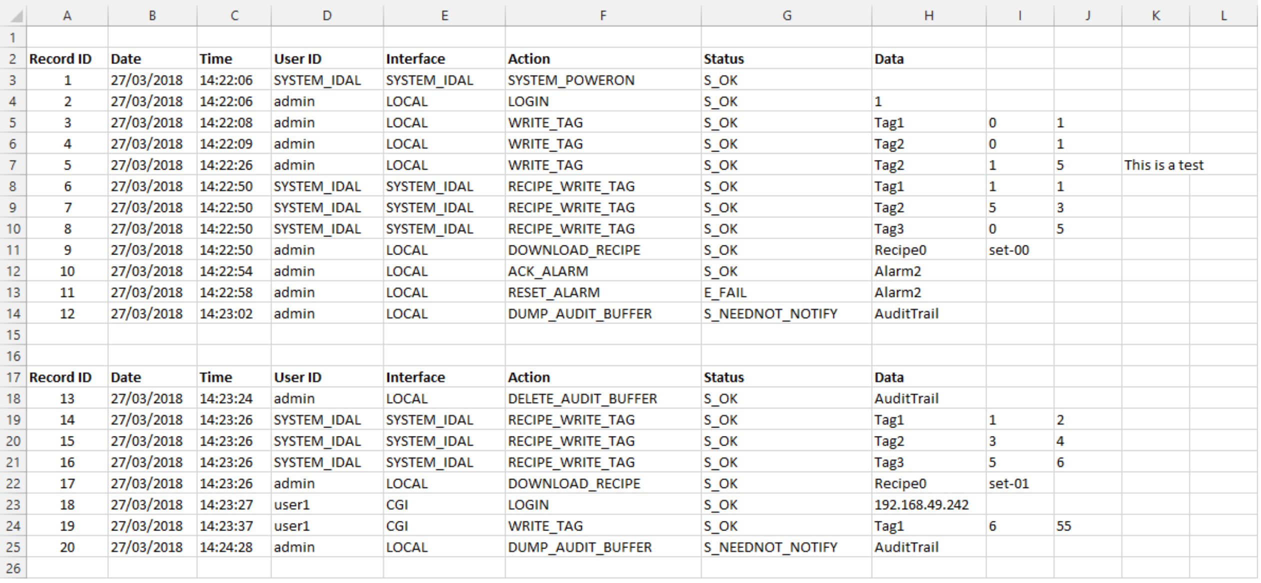


Name	Address
Duration	Duration int
EndTime	EndTime time

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 208 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	
RecordID	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.
Date, Time	Event time stamp. Time can be configured as local or global from the dump action.
User ID	User that perform the operation
Interface	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
Action	Action executed.
Status	Result of the executed action • S_OK Action executed correctly • E_FAIL Action non executed • S_NEEDNOT_NOTIFY Action triggered (will be executed asynchronously)
Information	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, AGI Creator 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 AGI Client.

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
Text Reports	Use for line-by-line printing of alarms. Only used for line printers. Text is sent to the printer without using any special driver. **Important: This printing mode requires using a physical port and only works on Windows CE platforms.**
Graphic Reports	Contain graphical elements and may include complex widgets such as screenshots or alarms. **Important: Each printer requires a specific printer driver. See "Configuring graphic reports" on the facing page for a list of supported printer drivers.**

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.

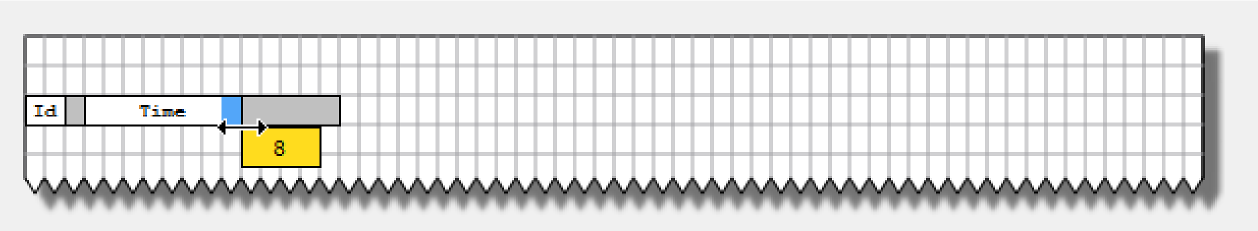
Note: Text reports do not support PDF format.

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.

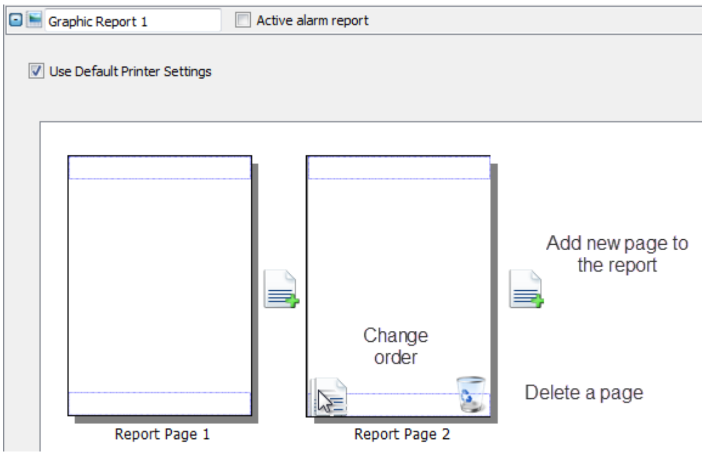


Re-size the field using the mouse, a tool tip shows the dimension in number of characters.

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

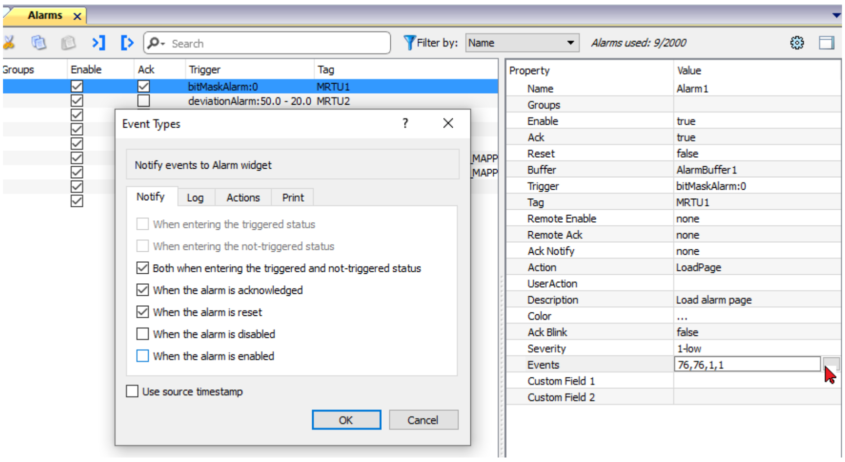
Print triggering events

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.



Groups	Enable	Ack	Trigger	Tag
	☑	☑	bitMaskAlarm:0	MRTU1
	☑	□	deviationAlarm:50.0 - 20.0	MRTU2

Event Types

Notify events to Alarm widget

□ When entering the triggered status
□ When entering the not-triggered status
☑ Both when entering the triggered and not-triggered status
☑ When the alarm is acknowledged
☑ When the alarm is reset
□ When the alarm is disabled
□ When the alarm is enabled

□ Use source timestamp

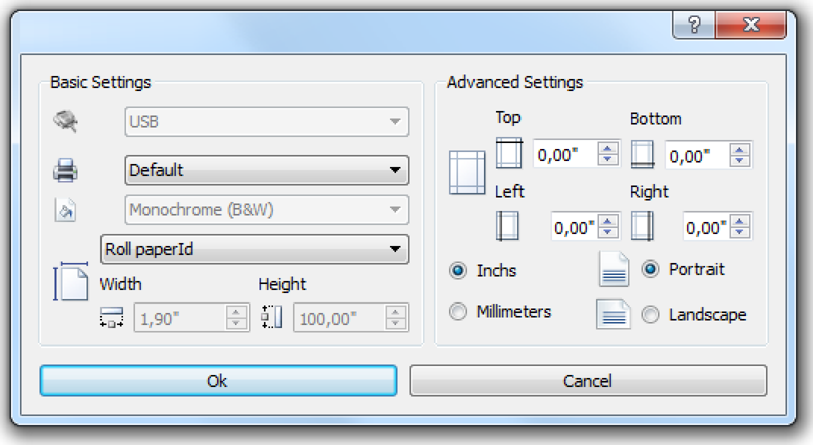
Property	Value
Name	Alarm1
Groups	
Enable	true
Ack	true
Reset	false
Buffer	AlarmBuffer1
Trigger	bitMaskAlarm:0
Tag	MRTU1
Remote Enable	none
Remote Ack	none
Ack Notify	none
Action	LoadPage
UserAction	
Description	Load alarm page
Color	...
Ack Blink	false
Severity	1-low
Events	76,76,1,1
Custom Field 1	
Custom Field 2	

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.



Basic Settings

USB

Default

Monochrome (B&W)

Roll paperId

Width: 1,90"

Height: 100,00"

Advanced Settings

Top: 0,00"

Bottom: 0,00"

Left: 0,00"

Right: 0,00"

● Inchs

○ Millimeters

● Portrait

○ Landscape

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**.

Supported printers

List of printers and printer languages supported by the Windows CE driver printCE.dll. Printers not available in the list but compatible with these languages are supported.

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
Cannon iP100, iP90, BubbleJet	BubbleJet, iP90, iP100
PocketJet II, 200, 3	Pocket Jet
MTE Mobile Pro Spectrum	MTE Mobile Pro Spectrum
Adobe PDF File	Adobe PDF file
SPT-8	SPT-8
M1POS	M1POS
MP300	MP300

Printer	Languages
Zebra	Zebra CPCL language
Intermec PB42, PB50, PB51, PB2, PB3	Intermect PB42/50/51/2/3 with ESC/P language
Datamax Apex	Datamax Apex

Supported ports

The following ports are supported:

- LPT1 (USB printers)
- File (PDF)

Note: On Windows platform, only PDF and default printers are supported. Default printer is the default OS printer and it can be connected with any kind of port (not only USB).

Tested printers

The following printers have been tested with printCE drivers in Windows CE HMI devices.

Driver	Printer Model	Graphic	Line
Custom	Plus 4 Kube II	Yes	Yes
Epson ESC/P 2	Epson AcuLaser M2310	Yes	Simulate
Epson LX (9-pin)	Epson LX-300+II	No	Yes
HP PCL 3	HP LaserJet P2015dm	Yes	Simulate
	HP LaserJet 4700dtn	Yes	Yes
HP PCL 3 GUI	HP Deskjet 1010	Yes	No
	HP Deskjet D5560	Yes	No
	HP LaserJet 4700dtn	No	Yes
HP PCL 5e	HP LaserJet P2015dm	Yes	Simulate
	HP LaserJet 4700dtn		
INTERMEC	Intermec PB50 with ESC/P language with 4 inch roll paper. Note: The HMI device crashes when trying to print on Intermec PB50 printers in standby mode after a first successful print job.	Yes	Yes
PDF	-	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.

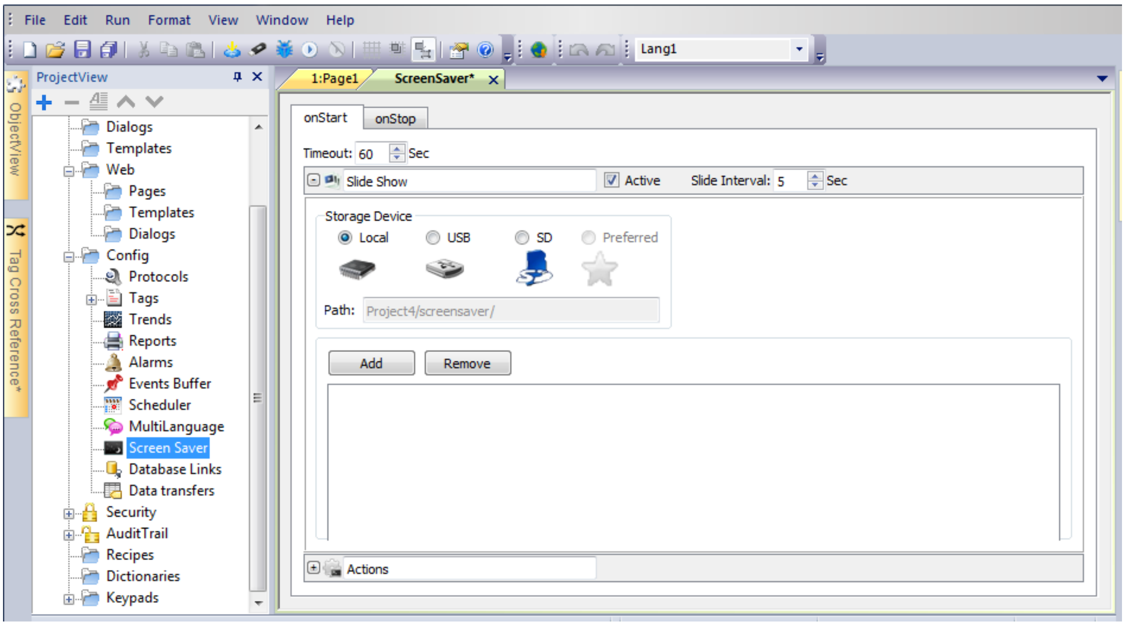
Enabling the screen saver function

*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: 60 Sec

☑ Active Slide Interval: 5 Sec

● Local ○ USB ○ SD ○ Preferred

Path: Project4/screensaver/

Slide show parameters

Parameter	Description
Timeout	Time after which the slide show starts
Slide Interval	Interval between slides
Storage Device	Location of the images used in the slide show. Images stored locally are saved in *workspace\projectname\screensaver* and can be downloaded to the HMI device when the project is downloaded. Images stored on USB or SD devices are saved in a screensaver folder on the device itself. ❗ **Important: Only JPEG and PNG images are supported.**

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
- CODESYS 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 335 for details.

Note: Backup is not supported in Windows Runtime and in AGI Client.

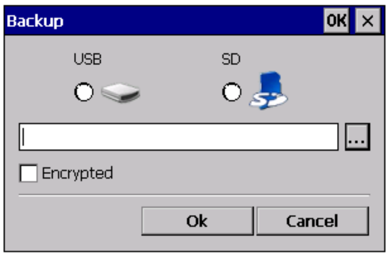
Backup function

The backup function automatically performs the following procedure:

1. Unloads the current project to unlock files in use.
2. Unload CODESYS service
3. 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).
4. 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

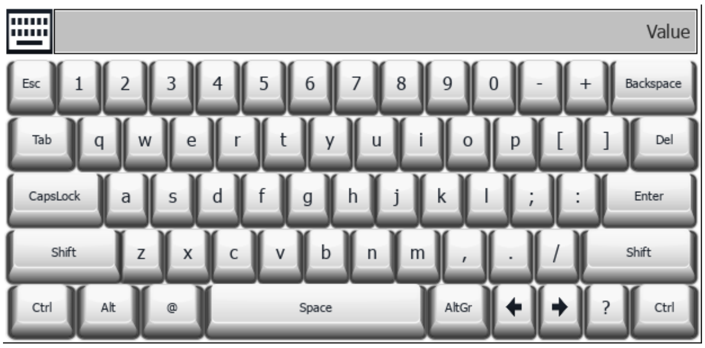
Restore the backup package can be perform on HMI device

- from the Context Menu (see "Update package" on page 101 for details)
- or from the System Settings (see "System Settings" on page 547 for details)

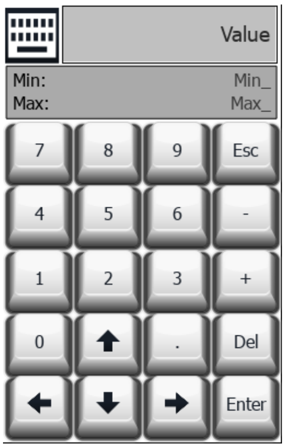
33 Keypads

Several keypads are provided by default in the AGI Creator so that they can be used for data entry.

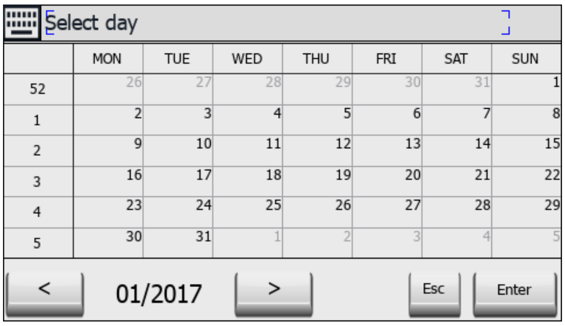
The alphabet keypad can be use associate with a string data type


Value
Esc 1 2 3 4 5 6 7 8 9 0 - + Backspace
Tab q w e r t y u i o p [ ] Del
CapsLock a s d f g h j k l ; : Enter
Shift z x c v b n m , . / Shift
Ctrl Alt @ Space AltGr ← → ? Ctrl

The numeric keypad can be use associate with a numeric data type



The calendar keypad can be use associate with a date data type



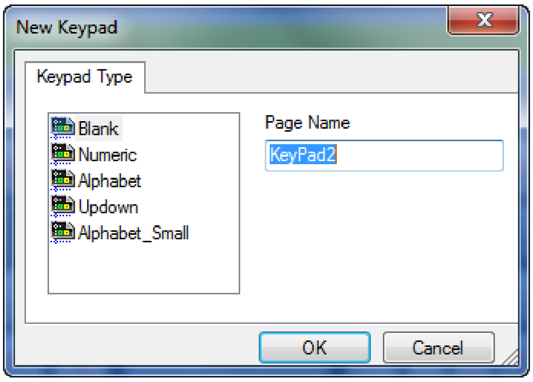
	MON	TUE	WED	THU	FRI	SAT	SUN
52	26	27	28	29	30	31	1
1	2	3	4	5	6	7	8
2	9	10	11	12	13	14	15
3	16	17	18	19	20	21	22
4	23	24	25	26	27	28	29
5	30	31	1	2	3	4	5

Creating and using custom keypads

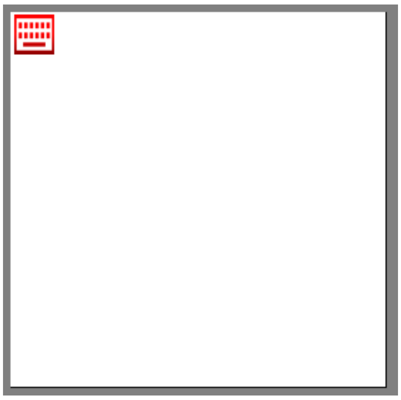
You can either create a new keypad or customize an existing one.

Creating a keypad

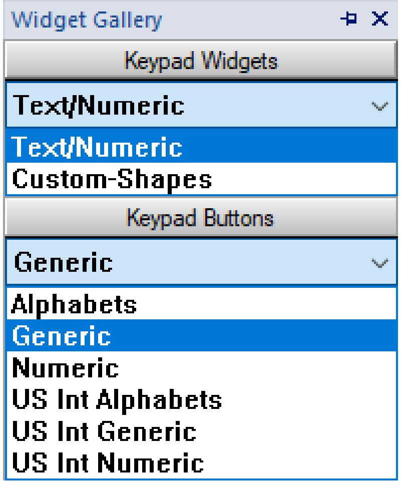
1. In **ProjectView**, right-click **Keypads** and select **Insert Keypad**: the **New Keypad** dialog is displayed.



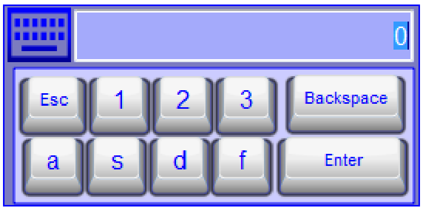
2. Select one of the available keypads, or **Blank** to create a keypad from scratch. In this case a blank keypad is displayed.



3. Use the **Keypad Widgets** and **Keypad Buttons** from the Widget Gallery to create your custom keypad.

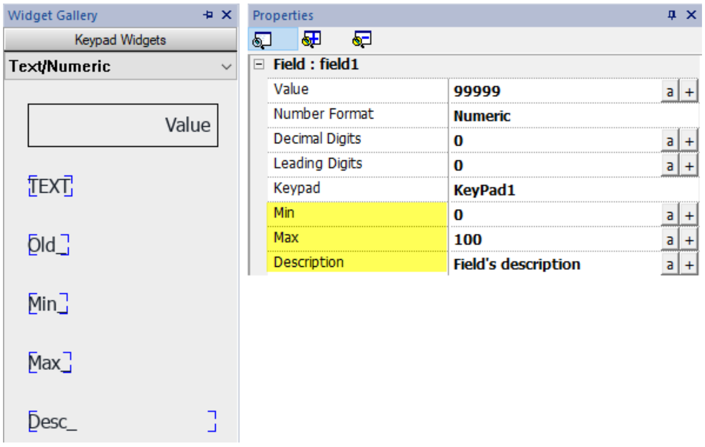


The keypad you create, as in this example, will be saved in the project folder.



Text/Numeric Controls

The Text/Numeric folder contains some specific controls to keypad development.



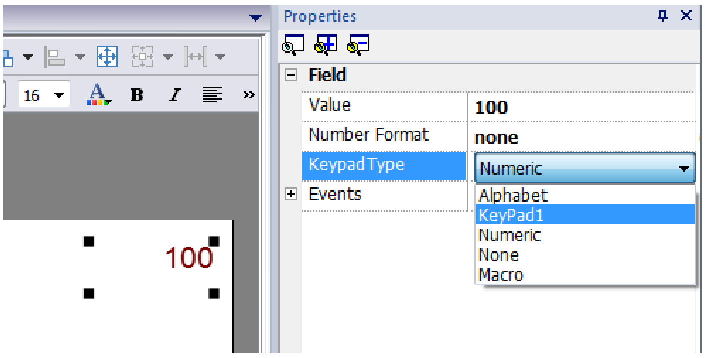
Field : field1

Value	99999
Number Format	Numeric
Decimal Digits	0
Leading Digits	0
Keypad	KeyPad1
Min	0
Max	100
Description	Field's description

Data source	Description
TEXT	Simple text label
Old_	Current value
Value	New value
Min_	Min value defined inside the field's property currently editing.
Max_	Max value defined inside the field's property currently editing.
Desc_	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.



Field

Value	100
Number Format	none
KeypadType	Numeric

Tips and tricks with custom keypads

By default, any numeric widget (read/write numeric field) are assigned the numeric keypad.

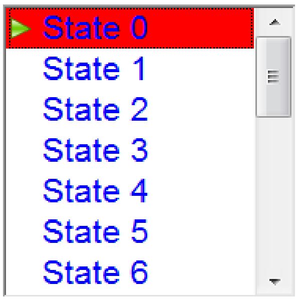
If you want to apply a customized version of the numeric keypad to all the numeric widgets you add to your project proceed as follows:

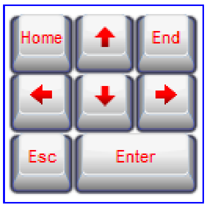
1. Create a new keypad and select **Numeric** as **Keypad** type. This will be a backup of the original settings for the numeric keypad.
2. Customize the default numeric keypad and save it. This customized version of the numeric keypad will now be assigned as default in the project.

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 widgets" on page 408 for details.

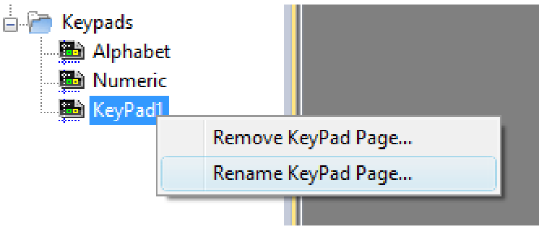




Deleting or renaming custom keypads

In **ProjectView**, right-click on a custom keypad and select one of the options:

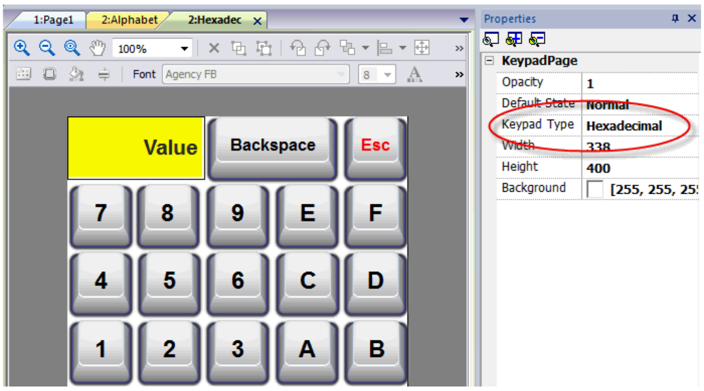
- **Remove KeyPad Page** to remove the keypad from the project
- **Rename Keypad Page** to rename the keypad.



Keypad type

*Path: **ProjectView**> **Keypads** > double-click a keypad > **Properties***

Set **Keypad Type** parameter for a keypad to define the type of data entry.



KeypadPage

Opacity	1
Default State	Normal
Keypad Type	Hexadecimal
Width	338
Height	400
Background	[255, 255, 25

Keypad Type	Description
Auto	Default setting
Decimal	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 filed is hexadecimal.
Hexadecimal	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.
Ascii	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
Automatic	The best position is selected according to here data entry is required.
Absolute	X,Y coordinates are entered to identify the exact position
Left-top **Left-center** **Left-bottom** **Center-top** **Center-center**	Predefined screen positions

Option	Description
Center-bottom	
Right-top	
Right-cente	
Right-bottom	

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.

Keyboards can be used for:

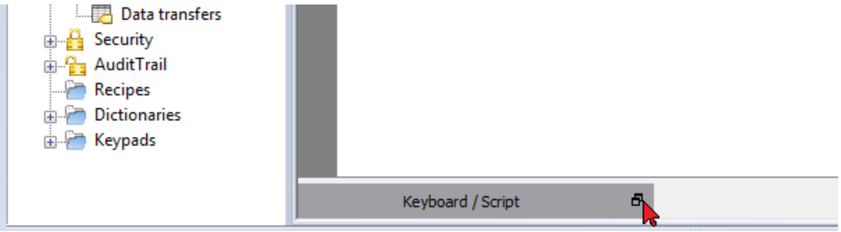
- data entry (default)
- execute actions mapped on specific keys

For example, the right arrow key **OnClick** event can be mapped to the **LoadPage** action.

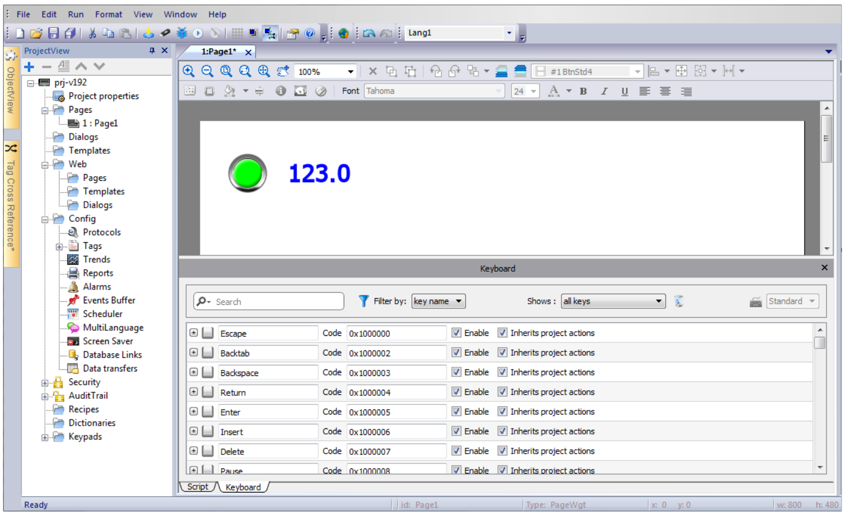
Keyboard can be programmed at project level so that settings will be inherited by all the pages. In each page you can then choose which key setting will be inherited from the project and which one you will customize for the specific page.

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.



Escape	Code	0x1000000	☑ Enable	☑ Inherits project actions
Backtab	Code	0x1000002	☑ Enable	☑ Inherits project actions
Backspace	Code	0x1000003	☑ Enable	☑ Inherits project actions
Return	Code	0x1000004	☑ Enable	☑ Inherits project actions
Enter	Code	0x1000005	☑ Enable	☑ Inherits project actions
Insert	Code	0x1000006	☑ Enable	☑ Inherits project actions
Delete	Code	0x1000007	☑ Enable	☑ Inherits project actions
Pause	Code	0x1000008	☑ Enable	☑ Inherits project actions

For each key, the following information is displayed:

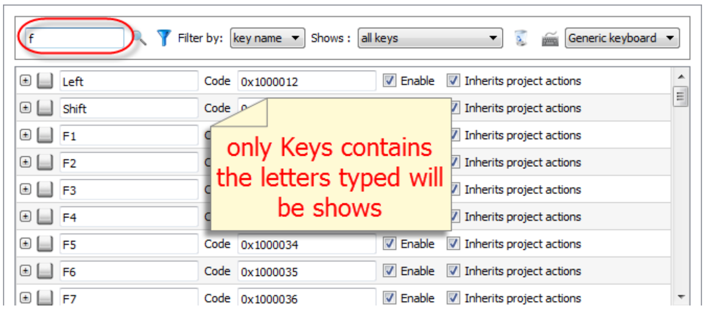
Element	Description
Label	Key name
Code	Key code
Enable	Key enable status
Inherits project actions	Defines whether the key is inheriting the action programmed at the project level

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.

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.

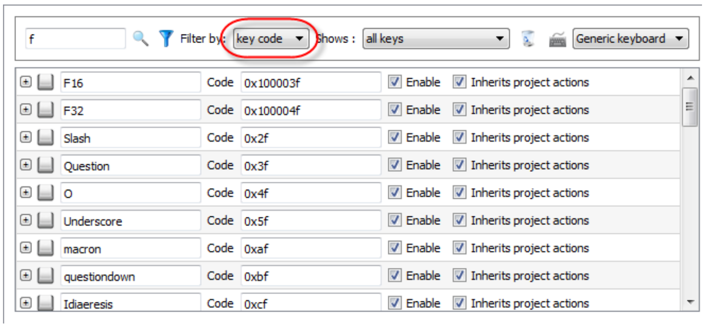


f Filter by: key name Shows : all keys Generic keyboard

Left	Code	0x1000012	☑ Enable	☑ Inherits project actions
Shift	Code	0		☑ Inherits project actions
F1	C			☑ Inherits project actions
F2	C			☑ Inherits project actions
F3	C			☑ Inherits project actions
F4	C			☑ Inherits project actions
F5	Code	0x1000034	☑ Enable	☑ Inherits project actions
F6	Code	0x1000035	☑ Enable	☑ Inherits project actions
F7	Code	0x1000036	☑ Enable	☑ Inherits project actions

only Keys contains the letters typed will be shows

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.



f Filter by: key code Shows : all keys Generic keyboard

F16	Code	0x100003f	☑ Enable	☑ Inherits project actions
F32	Code	0x100004f	☑ Enable	☑ Inherits project actions
Slash	Code	0x2f	☑ Enable	☑ Inherits project actions
Question	Code	0x3f	☑ Enable	☑ Inherits project actions
O	Code	0x4f	☑ Enable	☑ Inherits project actions
Underscore	Code	0x5f	☑ Enable	☑ Inherits project actions
macron	Code	0xaf	☑ Enable	☑ Inherits project actions
questiondown	Code	0xbf	☑ Enable	☑ Inherits project actions
Idiaeresis	Code	0xcf	☑ Enable	☑ Inherits project actions

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**.

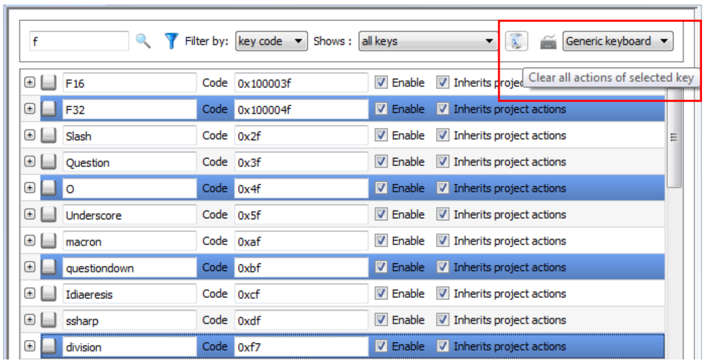
Option	Description
all keys	All keys available in the keyboard layout are listed
modified keys	Only the keys associated with actions at the page level are listed
modified keys in project	Only the keys associated with actions at project level are listed

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.



f | Filter by: key code | Shows : all keys | Generic keyboard

Clear all actions of selected key

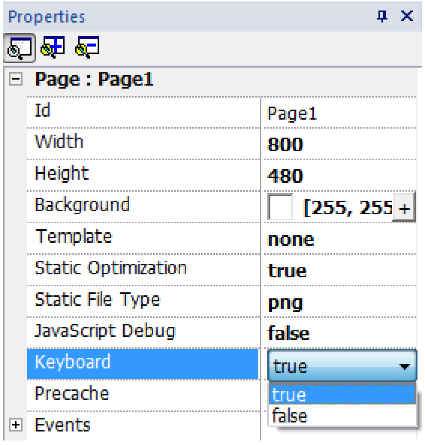
F16	Code	0x100003f	☑ Enable	☑ Inherits projec	
F32	Code	0x100004f	☑ Enable	☑ Inherits project actions	
Slash	Code	0x2f	☑ Enable	☑ Inherits project actions	
Question	Code	0x3f	☑ Enable	☑ Inherits project actions	
O	Code	0x4f	☑ Enable	☑ Inherits project actions	
Underscore	Code	0x5f	☑ Enable	☑ Inherits project actions	
macron	Code	0xaf	☑ Enable	☑ Inherits project actions	
questiondown	Code	0xbf	☑ Enable	☑ Inherits project actions	
Idiaeresis	Code	0xcf	☑ Enable	☑ Inherits project actions	
ssharp	Code	0xdf	☑ Enable	☑ Inherits project actions	
division	Code	0xf7	☑ Enable	☑ Inherits project actions	

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 **Keboard macro** to **true**.



Properties

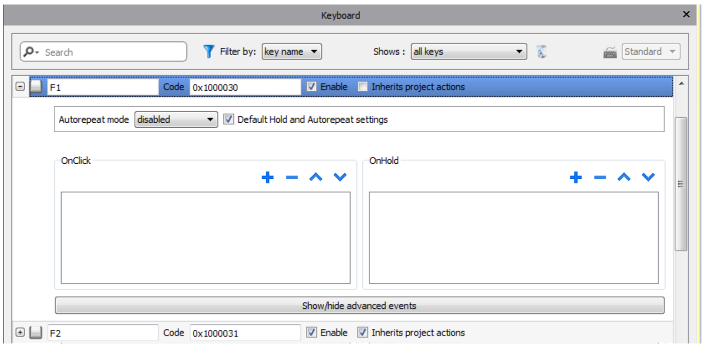
Page : Page1	
Id	Page1
Width	800
Height	480
Background	[255, 255 +
Template	none
Static Optimization	true
Static File Type	png
JavaScript Debug	false
Keyboard	true true false
Precache	
Events	

You can enable/disable keyboard actions also at runtime using the KeyboardMacros action. See "Keyboard actions" on page 180 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.



F1 Code 0x1000030 ☑ Enable □ Inherits project actions

Autorepeat mode disabled ☑ Default Hold and Autorepeat settings

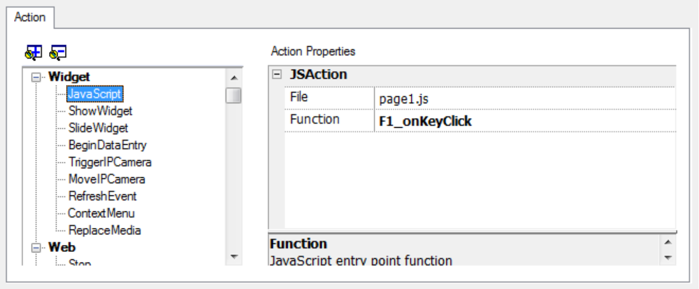
F2 Code 0x1000031 ☑ Enable ☑ Inherits project actions

1. Click **+** to add actions.

You can associate actions both to the **OnClick** event and toe the **OnHold** event.

See "Events" on page 55 for details.

Note: Also JavaScript code can be associated to a key event.



JSAction

File page1.js

Function F1_onKeyClick

Function
JavaScript entry point function

35 OPC UA Server

Path: ***ProjectView****>* ***Config*** *>* ***Interfaces*** *> double-click* ***OPC UA***

Use OPC UA Server to publish data according to the OPC UA standard.

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.
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
Anonymous	Anonymous clients accepted.
User/Password	Authentication with user name is accepted. Any valid user has unrestricted access to OPC UA Server (see "Configuring users" on page 341).

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
None	Connection without certificate is allowed. Not recommended in public networks.
Sign	OPC UA Client must provide its own certificate: communication through signed messages is allowed.
SignAndEncrypt	OPC UA Client must provide its own certificate: communication through signed and encrypted messages is allowed.

Security Policy	Description
Basic128Rsa15	Accepted encryption level (used only when Security Mode is active).
Basic256	Accepted encryption level (used only when Security Mode is active).
Basic256Sha256	Accepted encryption level (used only when Security Mode is active).

Parameter	Description
Automatically trust any new clients	All certificates provided from any OPC UA Clients are accepted.
Trusted Certificates	Only OPC UA Clients that provide one of the listed certificates are accepted. 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.

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
Manufacturer name	Human readable name of the manufacturer of the product. OPC UA Client can retrieve this information from tag: *ServerName|Objects|Server|ServerStatus|BuildInfo|ManufacturerName*
Product name	A human readable name for the product running in the server. The OPC UA Client can retrieve this information from tag: *ServerName|Objects|Server|ServerStatus|BuildInfo|ProductName*

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
Server Name (Common Name)	Name of the certificate (e.g. the device name).
Organization	Organization name
Unit	Organization unit ℹ This field could be useful to differentiate different divisions within an organization.
Location	Locality field denotes the city where organization resides in
State	State or Province field specifies where the organization is physically located. ℹ 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.
Country	The X.509 naming scheme standard requires a 2-character country code. ℹ Country code for the United States is US; country code for Italy is IT.

Parameter	Description
Produc URI	A globally unique identifier for the server. Example: "*urn:NodeName:CompanyName:ServerName*"
DNS Names **IP Addresses**	DNS name or IP Address of the device where this OPC UA Server is installed. Multiple DNS Names and/or IP Addresses can be in a single certificate. ❗ **The certificate will be valid only if the IP address where the OPC UA Server is running is included in this list.**
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/)

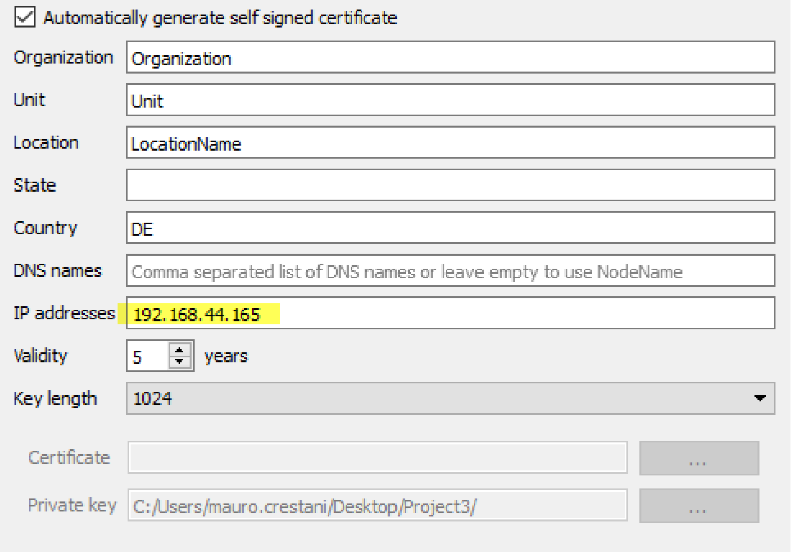
```
@echo off
set OpenSSL="C:\Program Files (x86)\OpenSSL-Win32\bin\openssl.exe"
set NodeName=HMI-Server
set IPAddress=192.168.44.165

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&amp;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



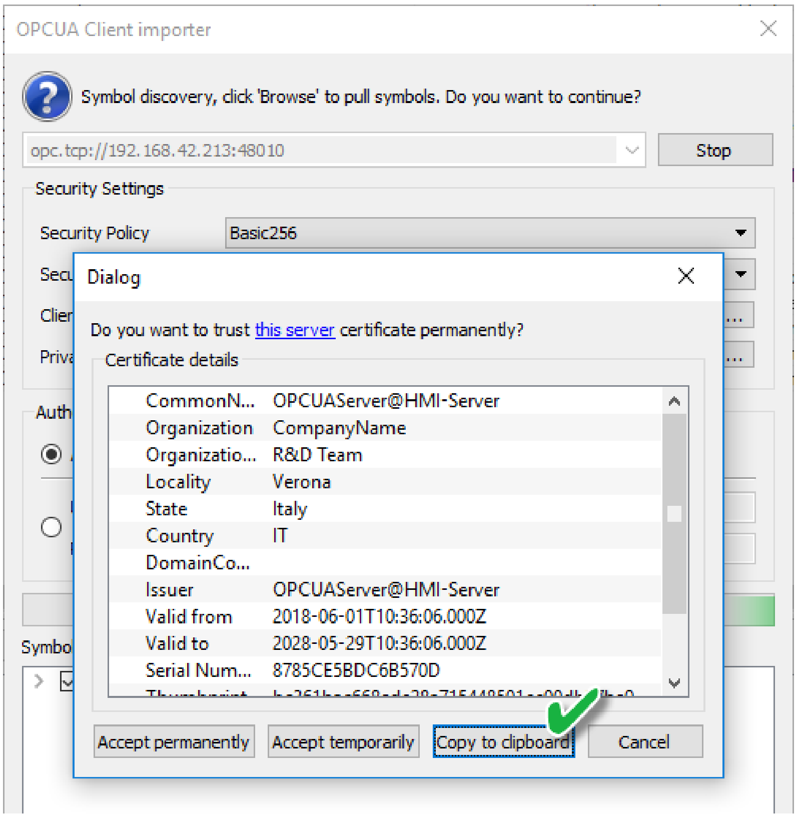
☑ Automatically generate self signed certificate

Organization	Organization
Unit	Unit
Location	LocationName
State	
Country	DE
DNS names	Comma separated list of DNS names or leave empty to use NodeName
IP addresses	192.168.44.165
Validity	5 years
Key length	1024
Certificate	
Private key	C:/Users/mauro.crestani/Desktop/Project3/

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.



Symbol discovery, click 'Browse' to pull symbols. Do you want to continue?

opc.tcp://192.168.42.213:48010

Security Policy: Basic256

Do you want to trust this server certificate permanently?

CommonN...	OPCUAServer@HMI-Server
Organization	CompanyName
Organizatio...	R&D Team
Locality	Verona
State	Italy
Country	IT
DomainCo...	
Issuer	OPCUAServer@HMI-Server
Valid from	2018-06-01T10:36:06.000Z
Valid to	2028-05-29T10:36:06.000Z
Serial Num...	8785CE5BDC6B570D

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\DEIF A/S\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

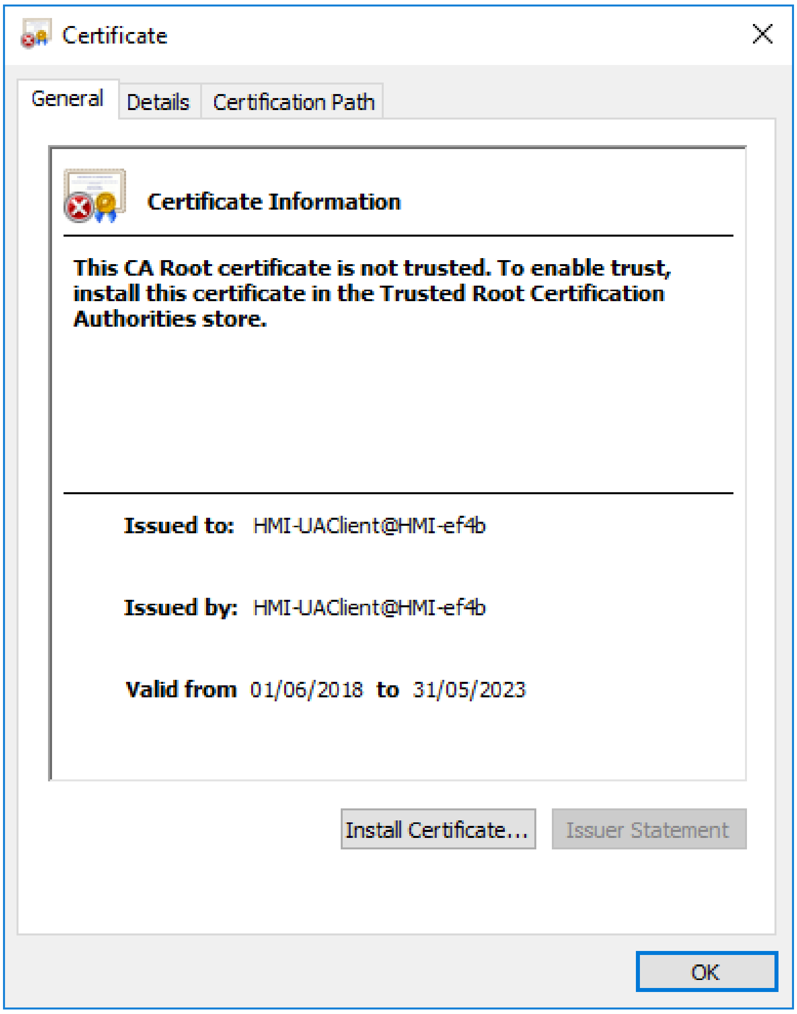
Since in the OPC UA Client protocol parameters we left empty the “Client certificate” field, the OPC UA Client protocol has generated its own certificate and sent it to the OPC UA Server but since the server does not know this certificate it rejects the connection request. Now we have to tell the server 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

   Certificate Information

   This CA Root certificate is not trusted. To enable trust, install this certificate in the Trusted Root Certification Authorities store.

   Issued to: HMI-UAClient@HMI-ef4b

   Issued by: HMI-UAClient@HMI-ef4b

   Valid from 01/06/2018 to 31/05/2023

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.

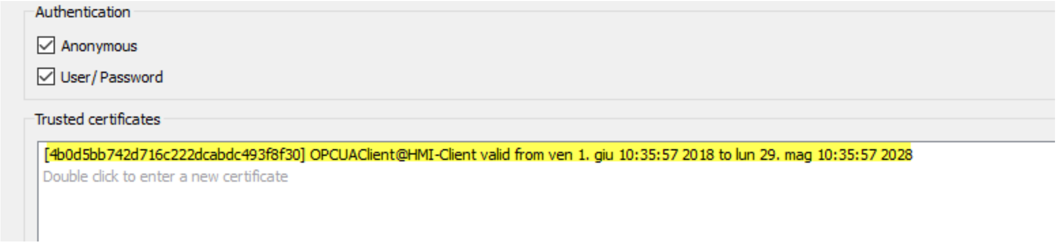
Generate certificates

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 382) 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



Authentication

☑ Anonymous

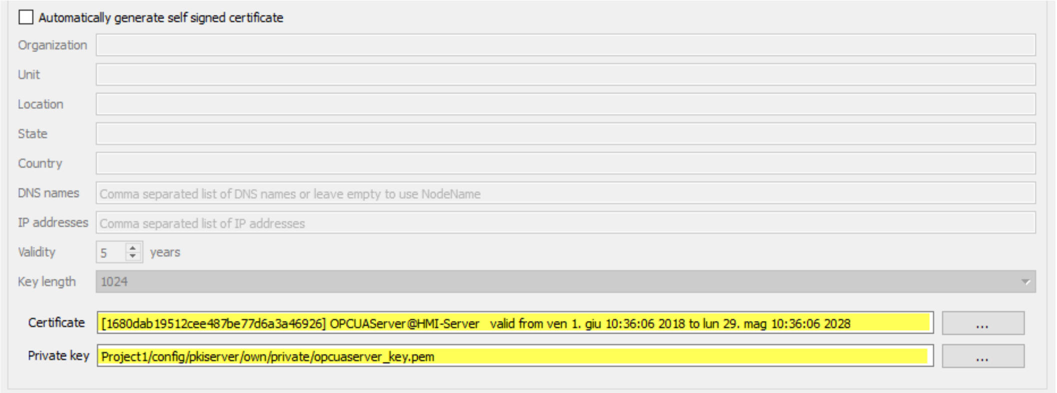
☑ User/ Password

Trusted certificates

[4b0d5bb742d716c222dcabdc493f8f30] OPCUAClient@HMI-Client valid from ven 1. giu 10:35:57 2018 to lun 29. mag 10:35:57 2028

Double click to enter a new certificate

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)



☐ Automatically generate self signed certificate

Organization

Unit

Location

State

Country

DNS names: Comma separated list of DNS names or leave empty to use NodeName

IP addresses: Comma separated list of IP addresses

Validity: 5 years

Key length: 1024

Certificate: [1680dab19512cee487be77d6a3a46926] OPCUAServer@HMI-Server valid from ven 1. giu 10:36:06 2018 to lun 29. mag 10:36:06 2028 ...

Private key: Project1/config/pkiserver/own/private/opcuaserver_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	AGI Creator Alarm state
Opcua.Alarm.Active	True when alarm state is triggered • TRIGGERED • TRIGGERED_NOT_ACKED • TRIGGERED_ACKED
Opcua.Alarm.Acked	True when alarm acknowledgment is not required • TRIGGERED_ACKED • NOT_TRIGGERED_ACKED • NOT_TRIGGERED
Opcua.Alarm.Retain	True when alarm is pending • TRIGGERED | • TRIGGERED_NOT_ACKED | • TRIGGERED_ACKED | • NOT_TRIGGERED_ACKED but a RESET is required
Opcua.Alarm.Confirmed	True when alarm is returned (Not triggered, acknowledged and reset) • NOT_TRIGGERED ℹ This info is available only when alarm is configured to required a RESET

36 MQTT Interface

*Path: **ProjectView**> **Config** > **Interfaces** > double-click **MQTT***

Use MQTT Interface to publish data according to the MQTT standard.

Note that a tag or an alarm, to be transferred through the MQTT protocol, must be defined within a group.

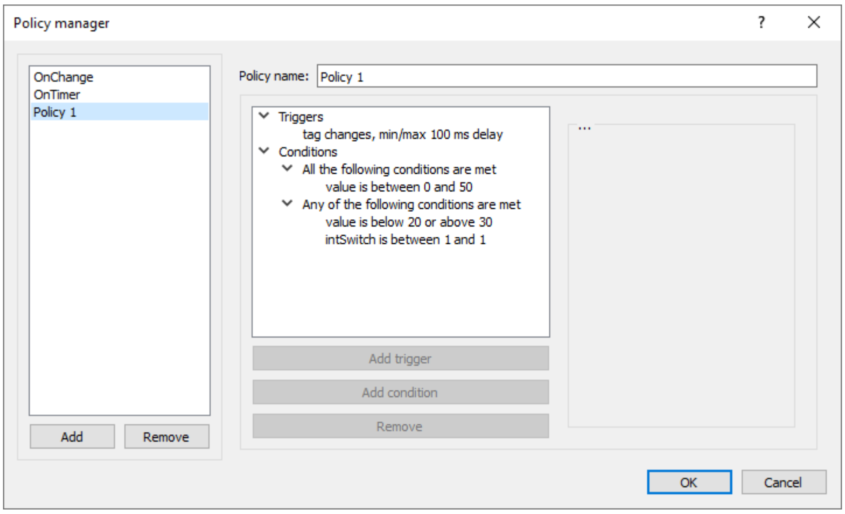
Parameter	Description
Enable MQTT Interface	Main flag to activate MQTT service. The selected groups of tags will be published to the MQTT broker.
Enable Alarms	The selected groups of alarms will be published to the MQTT broker. Alarms are published whenever there is a change in the alarm status.

Tags configuration

Parameter	Description
Enable	Enable the transferring of the tags listed inside the group
Tag Group	List of tags that will be transferred when the assigned police condition will be satisfied.
QoS	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
Retain	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.
Persistence	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.
Policy	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.



OnChange
OnTimer
Policy 1

Policy name: Policy 1

- Triggers
  - tag changes, min/max 100 ms delay
- Conditions
  - All the following conditions are met
    - value is between 0 and 50
  - Any of the following conditions are met
    - value is below 20 or above 30
    - intSwitch is between 1 and 1

Trigger

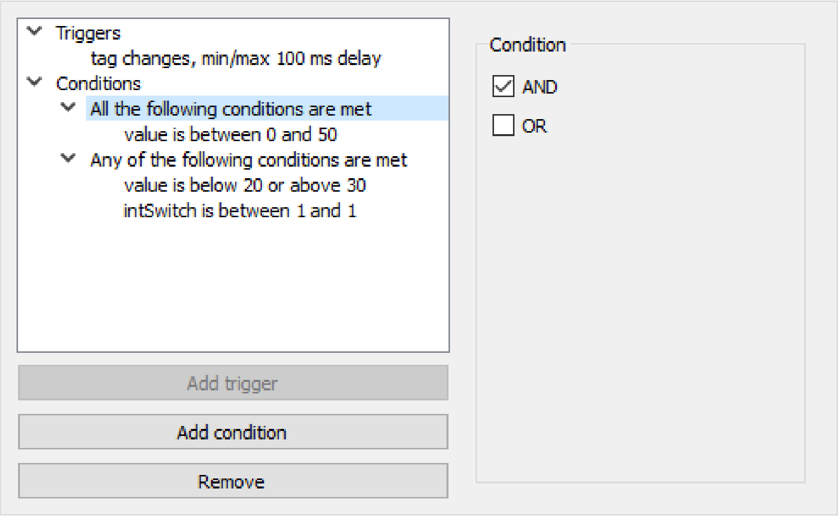
Parameter	Description
Timer	Publish is performed continuously even value is not changing. • Interval (ms) Cyclical publication time
On change	Publish is performed when a tag value changes. • Min interval (ms) Value check interval • Deadband The difference, from previous publish, that must be found to trigger the new publish. • Use percentual Dead band value express in percentage • Tag Name Tags to be checked to activate the publication. If empty, the tag to be published is used.

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.



- Triggers
  - tag changes, min/max 100 ms delay
- Conditions
  - All the following conditions are met
    - value is between 0 and 50
  - Any of the following conditions are met
    - value is below 20 or above 30
    - intSwitch is between 1 and 1

Condition

☑ AND

☐ OR

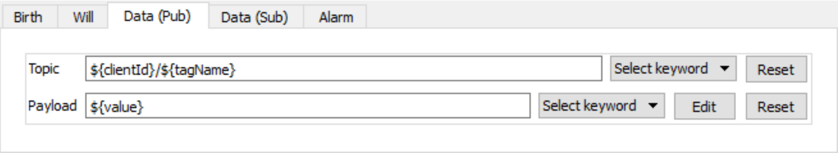
Settings

Topic	Description
Max pending messages	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.
Defaults	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
Birth	This is a special topic that is publish only one time when HMI device start.
Will	This is a special topic that is publish when HMI device start but it is keep hidden and stored from the MQTT Broker. It will be published from the MQTT Broker if it detects that the client has disconnected ungracefully.
Data (Pub)	Topic is used to publish the tags' values following the transmission policies associated with tag groups.
Data (Sub)	Topic is used to subscribe to tags. The payload is the template used to recognize the values of the received tags.
Alarm	Topic used to publish alarms



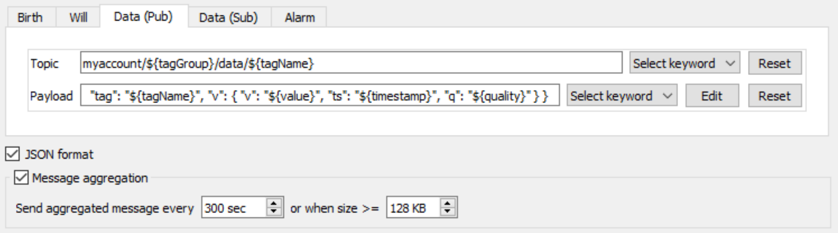
Topic ${clientId}/${tagName}

Payload ${value}

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
${clientId}	MQTT Client ID
${protocolName}	Name of the protocol associated to a tag
${tagGroup}	Name of the group the tag belongs to
${tagName}	Name of the tag
${alarmGroup}	Name of the group the alarm belongs to
${alarmName}	Name of alarm
${timestamp}	Current time
${value}	Last known value of the tag
${activeValue}	Value of the tag when the alarm became active
${quality}	Quality (i.e. reliability) of the tag
${activeTimestamp}	Timestamp of the last event that raised the alarm
${inactiveTimestamp}	Timestamp of the last event that ceased the alarm condition
${ackTimestamp}	Timestamp when the operator acknowledge the alarm
${description}	Alarm description
${customField1}	Alarm Custom Field 1
${customField2}	Alarm Custom Field 2
${state}	Alarm State
${severity}	Alarm Severity
${lowLimit}	Alarm “Low limit”
${highLimit}	Alarm “High limit”
${[0]}	If available in the alarm description, the value of the first live tag, [1] the second, etc.
${[Tag1]}	If available in the alarm description, the value of "Tag1" live tag

JSON Payload



Topic: myaccount/${tagGroup}/data/${tagName}

Payload: "tag": "${tagName}", "v": { "v": "${value}", "ts": "${timestamp}", "q": "${quality}" } }

☑ JSON format

☑ Message aggregation

Send aggregated message every 300 sec or when size >= 128 KB

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.

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
Broker address	Name or IP address of the MQTT server
Port	MQTT server port. Generally, the default TCP/IP port is 1883, or the port 8883 when MQTT over SSL is used.
Client ID	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.
Username **Password**	If the MQTT broker is configured to require client authentication using a valid user name and password
Keep-alive time (s)	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).
Use clean session	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.
Use legacy	The “Use legacy” flag makes client comply with MQTT spec 3.1

Enable TLS

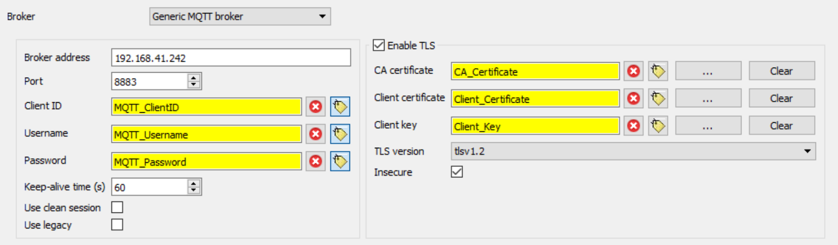
If the MQTT server is configured to works over TLS connection, the HMI device must provide its own certificate to the server. Even it's not mandatory, each client should have its own certificate (however it is possible you can deploy the same certificate to all clients).

It is required that both server and client certificates are signed by the same authority.

Parameter	Description
Enable TLS	Enable the TLS encryption
CA Certificate	Public certificate of the CA that has signed the server certificate on the Mosquitto Broker
Client Certificate	Public certificate of the HMI Device. Must be signed from the CA Certificate
Client Key	Private key associated with the client certificate
TLS Version	TLS Version to use (must be aligned with the encryption level used from the MQTT Broker) • tlsv 1 • tlsv 1.1 • tlsv 1.2
Insecure	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.

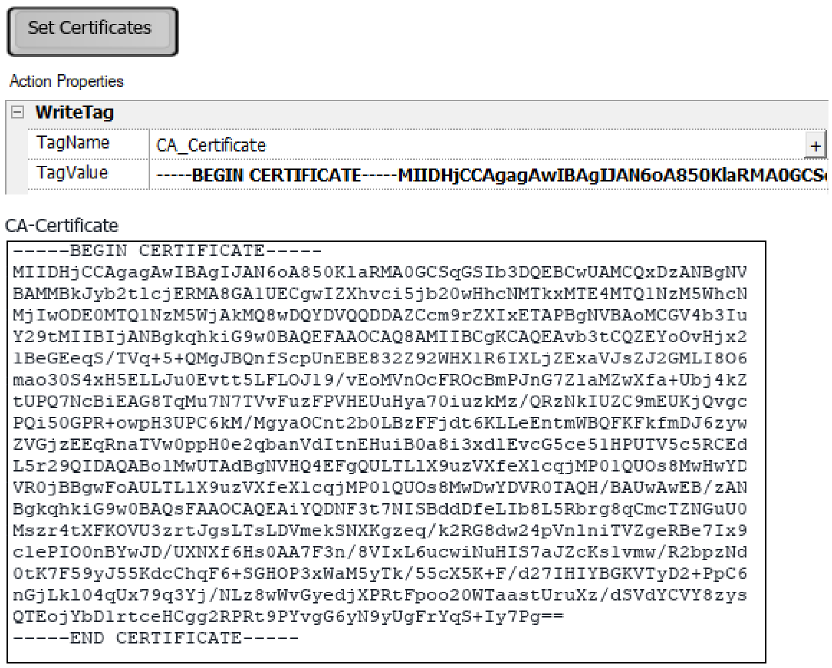


Broker: Generic MQTT broker

Broker address: 192.168.41.242
Port: 8883
Client ID: MQTT_ClientID
Username: MQTT_Username
Password: MQTT_Password
Keep-alive time (s): 60
Use clean session: □
Use legacy: □

☑ Enable TLS
CA certificate: CA_Certificate
Client certificate: Client_Certificate
Client key: Client_Key
TLS version: tlsv1.2
Insecure: ☑

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.



TagName	CA_Certificate
TagValue	-----BEGIN CERTIFICATE-----MIIDHjCCAgagAwIBAgIJAN6oA850KlaRMA0GCS

CA-Certificate

```
-----BEGIN CERTIFICATE-----
MIIDHjCCAgagAwIBAgIJAN6oA850KlaRMA0GCSqGSIb3DQEBCwUAMCQxDzANBgNV
BAMMBkJyb2tlcjERMA8GA1UECgwIZXhvci5jb20wHhcNMTkxMTE4MTQ1NzM5WhcN
MjIwODE0MTQ1NzM5WjAkMQ8wDQYDVQQDDAZCcm9rZXIxETAPBgNVBAoMCGV4b3Iu
Y29tMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAvb3tCQZEYoOvHjx2
lBeGEeqS/TVq+5+QMgJBQnfScpUnEBE832Z92WHX1R6IXLjZExaVJsZJ2GMLI8O6
mao30S4xH5ELLJu0Evtt5LFLOJl9/vEoMVnOcFROcBmPJnG7ZlaMZwXfa+Ubj4kZ
tUPQ7NcBiEAG8TqMu7N7TVvFuzFPVHEUuHya70iuzkMz/QRzNkIUZC9mEUKjQvgc
PQi50GPR+owpH3UPC6kM/MgyaOCnt2b0LBzFFjdt6KLLeEntmWBQFKFkfmDJ6zyw
ZVGjzEEqRnaTVw0ppH0e2qbanVdItnEHuiB0a8i3xdlEvcG5ce5lHPUTV5c5RCEd
L5r29QIDAQABo1MwUTAdBgNVHQ4EFgQULTLlX9uzVXfeXlcqjMP01QUOs8MwHwYD
VR0jBBgwFoAULTLlX9uzVXfeXlcqjMP01QUOs8MwDwYDVR0TAQH/BAUwAwEB/zAN
BgkqhkiG9w0BAQsFAAOCAQEAiYQDNF3t7NISBddDfeLIb8L5Rbrg8qCmcTZNGuU0
Mszr4tXFKOVU3zrtJgsLTsLDVmekSNXKgzeq/k2RG8dw24pVnlniTVZgeRBe7Ix9
clePIO0nBYwJD/UXNXf6Hs0AA7F3n/8VIxL6ucwiNuHIS7aJZcKslvmw/R2bpzNd
0tK7F59yJ55KdcChqF6+SGHOP3xWaM5yTk/55cX5K+F/d27IHIYBGKVTyD2+PpC6
nGjLkl04qUx79q3Yj/NLz8wWvGyedjXPRtFpoo20WTaastUruXz/dSVdYCVY8zys
QTEojYbDlrtceHCgg2RPRt9PYvgG6yN9yUgFrYqS+Iy7Pg==
-----END CERTIFICATE-----
```

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.

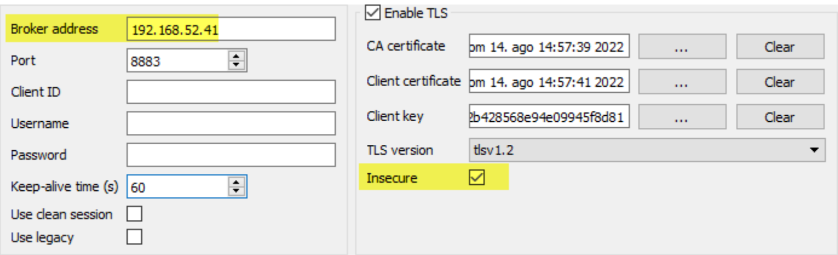
- ca.crt
- server.crt, server.key
- client.crt, client.ket

File: *CreateCertificates.cmd*

```
@echo off
set OpenSSL="C:\Program Files (x86)\OpenSSL-Win32\bin\openssl.exe"

rem Generate self signed CA certificate (Certificate Autority)
    %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
Port: 8883
Client ID:
Username:
Password:
Keep-alive time (s): 60
Use clean session ☐
Use legacy ☐

☑ Enable TLS
CA certificate: pm 14. ago 14:57:39 2022
Client certificate: pm 14. ago 14:57:41 2022
Client key: 2b428568e94e09945f8d81
TLS version: tlsv1.2
Insecure ☑

MQTT Broker configuration

The server certificate (server.crt, server.key) and the authority certificate (ca.crt) must be place 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:

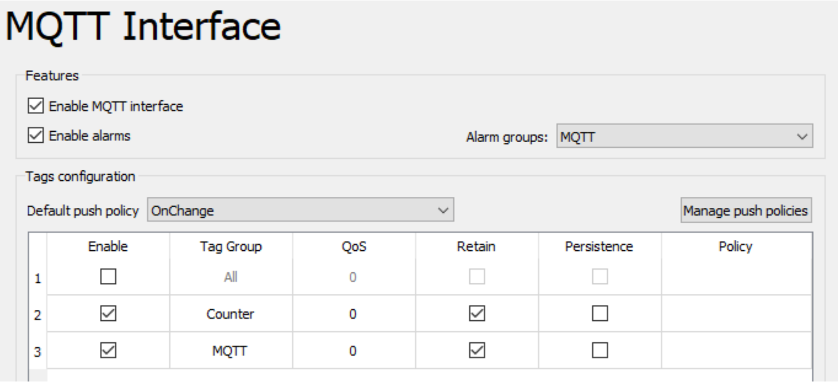
```
mosquitto_sub --cafile certs-client\ca.crt --cert certs-client\client.crt --key certs-
client\client.key -p 8883 -t /#
```

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

	Enable	Tag Group	QoS	Retain	Persistence	Policy
1	☐	All	0	☐	☐	
2	☑	Counter	0	☑	☐	
3	☑	MQTT	0	☑	☐	

37 Special widgets

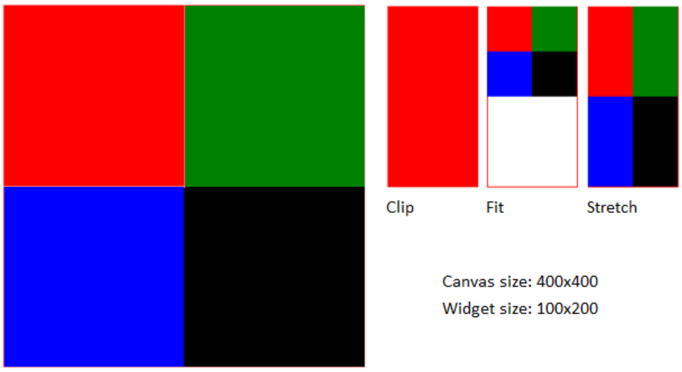
Widgets designed for special purposes are called special widgets and include control lists, date and time widgets, variable widgets and so on.

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
Canvas Width **Canvas Height**	Canvas size. 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.
Draw Hint	Define how fit the canvas inside the widget size - Clip No Transformation is applied, coordinate system is not scaled and drawing is clipped inside the widget bounding rectangle. - Fit to size Fit to the widget size preserving the canvas model aspect ratio. - Stretch Fit to the widget size ignoring the canvas model aspect ratio. Example using a Canvas size larger than the widget size:  Clip Fit Stretch Canvas size: 400x400 Widget size: 100x200
Design Time Preview	Canvas preview inside AGI Creator Note the JavaScript code could use data not available inside AGI Creator but only inside the HMI device
Auto Clear Background	Automatic clear the background before draw canvas. When disabled, the painted items are persisted and is not necessary redraw everything from scratch.

Parameter	Description
OnDraw Action	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.
OnMousePress Action **OnMouseRelease Actions** **OnMouseDrag Actions**	Mouse events

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 CImageWgt 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 );
}
```

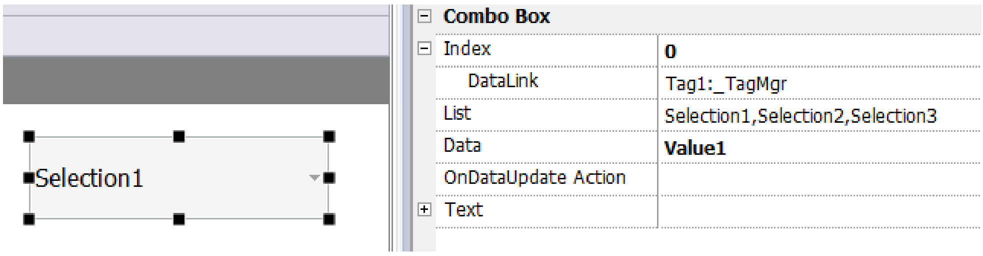
The update method can be used to dynamically redraw a canvas widget

```
function BtnStd1_btn_onMouseClick(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.



Combo Box	
Index	0
DataLink	Tag1:_TagMgr
List	Selection1,Selection2,Selection3
Data	Value1
OnDataUpdate Action	
Text	

Parameter	Description
Index	Index of the selected item.
List / String List	Item strings in the combo box. Note: This field is multi-language.
Data / Data List	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.
Text	Format of displayed text.

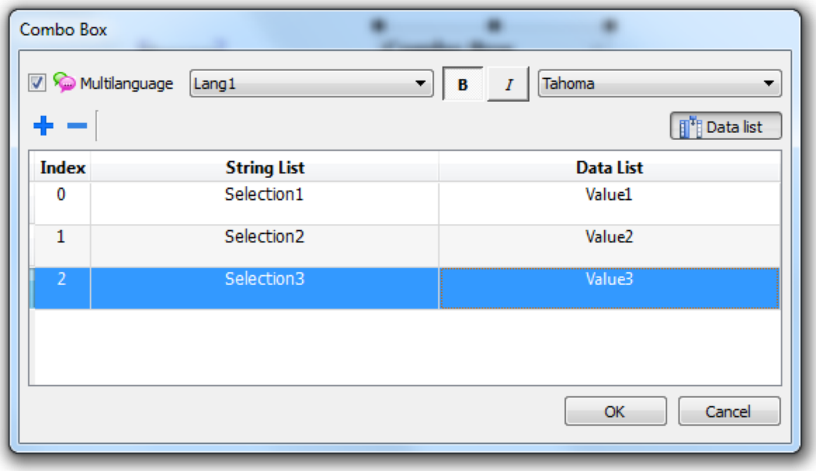
Data List

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



Index	String List	Data List
0	Selection1	Value1
1	Selection2	Value2
2	Selection3	Value3

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.

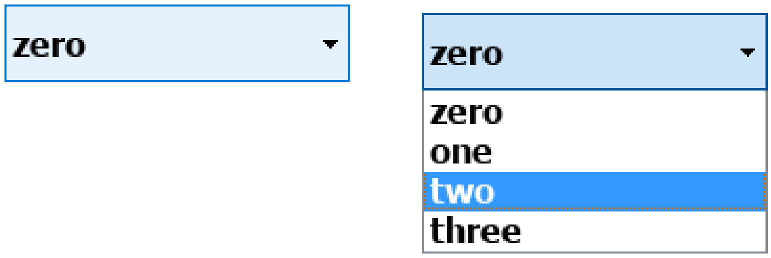
Combo Box widget “full screen” mode with images

From the "Project properties" on page 79 the look and behavior of Combo Boxes can switches from Context mode to Full Screen mode

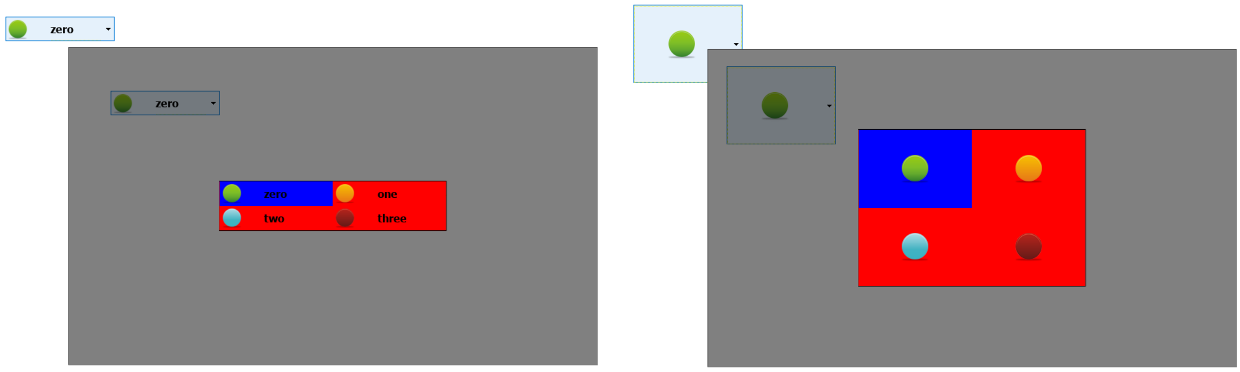
Path: **ProjectView**> double-click **Project properties**> **Properties pane**> **Project**> **ComboBox View Mode**

Parameter	Description
ComboBox View Mode	Select the visualization mode of all the Combo Box widgets of the project **Context** Classic view with drop-down menus **Full screen** 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

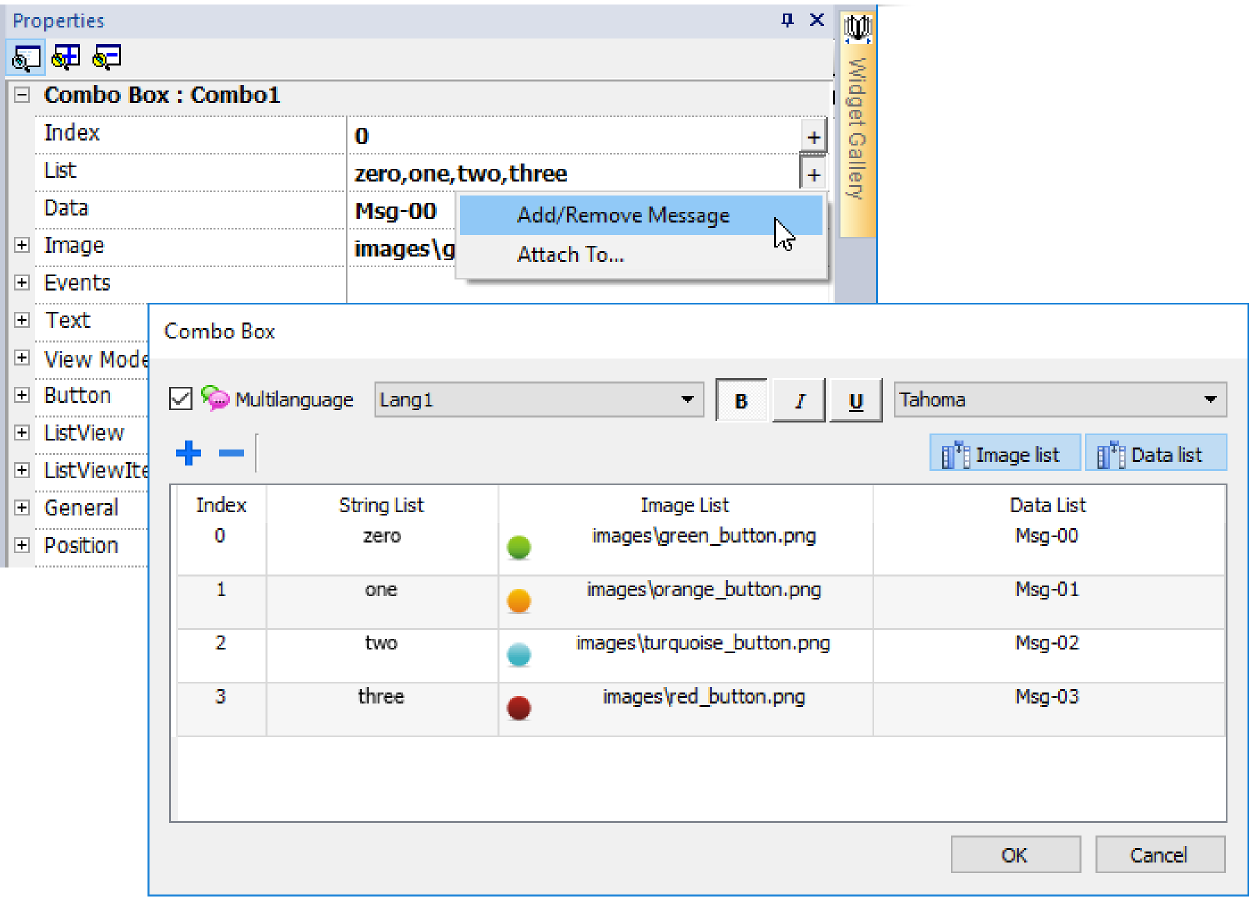


Full screen view example



Additional parameters available in full screen mode

The additional "*Image List*" column will be available inside **Combo Box**> **List** parameter:



Combo Box : Combo1

Index	0
List	zero,one,two,three
Data	Msg-00
Image	images\g
Events	
Text	
View Mode	
Button	
ListView	
ListViewIte	
General	
Position	

☑ Multilanguage Lang1 Tahoma

Index	String List	Image List	Data List
0	zero	images\green_button.png	Msg-00
1	one	images\orange_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
Image	Return, inside the attached tag, the file name of the selected image
Button	Define the look of the Combo Box • Show background = true Combo Box button is showed • Show background = false Only image or text is showed
ListView	Layout parameters of the Combo Box in edit mode
ListViewItems	Define the items type that will be inside the Combo Box Image Mode: • Only Text • Only Images • Text and Images

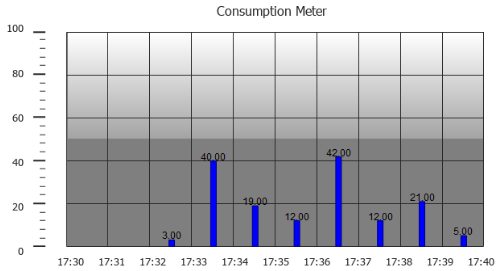
Consumption Meter widget

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.

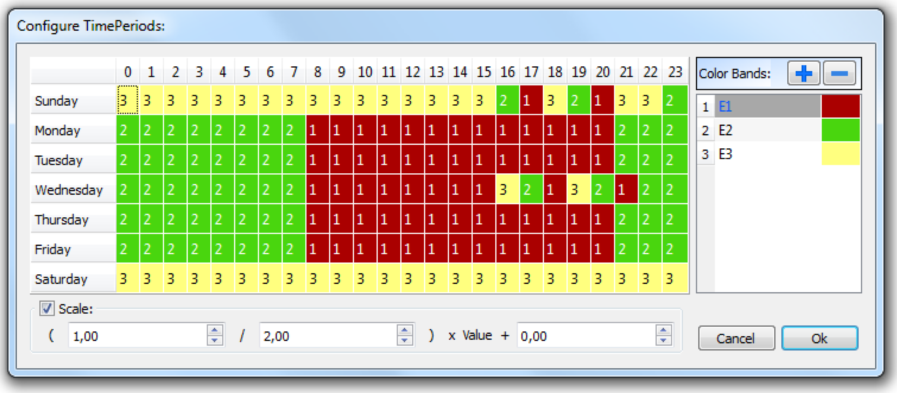


Parameter	Description
Value	Resource monitored
Graph Duration **Graph Duration Units**	Time period displayed in the window
Bar Duration **Bar Duration Units**	Time period represented by each bar in the graph
Time Periods	Assigns a specific color to highlight the increment of the monitored resource in a specified time period (minimum resolution = 1 hour).
Color **Bar Width**	Bar color and width
Bar Value	Show/Hide the value of each bar
Consumption Meter	Number of labels to be displayed on graph.

Example: how to monitor energy consumption

In the following example a widget is design tho 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.



Configure TimePeriods:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Sunday	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	2	1	3	2	1	3	3	2
Monday	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2
Tuesday	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2
Wednesday	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	3	2	1	3	2	1	2	2
Thursday	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2
Friday	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2
Saturday	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3

Color Bands: + −

1	E1	🟥
2	E2	🟩
3	E3	🟨

☑ Scale:

(1,00 / 2,00) x Value + 0,00

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.

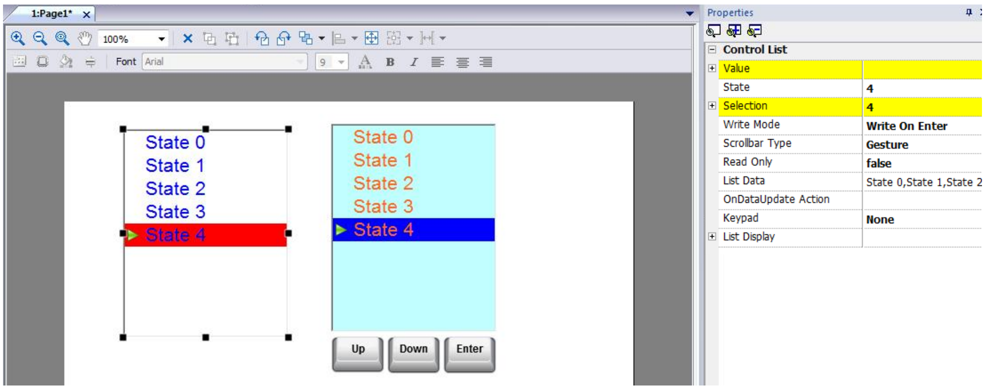
Control list widgets

*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.

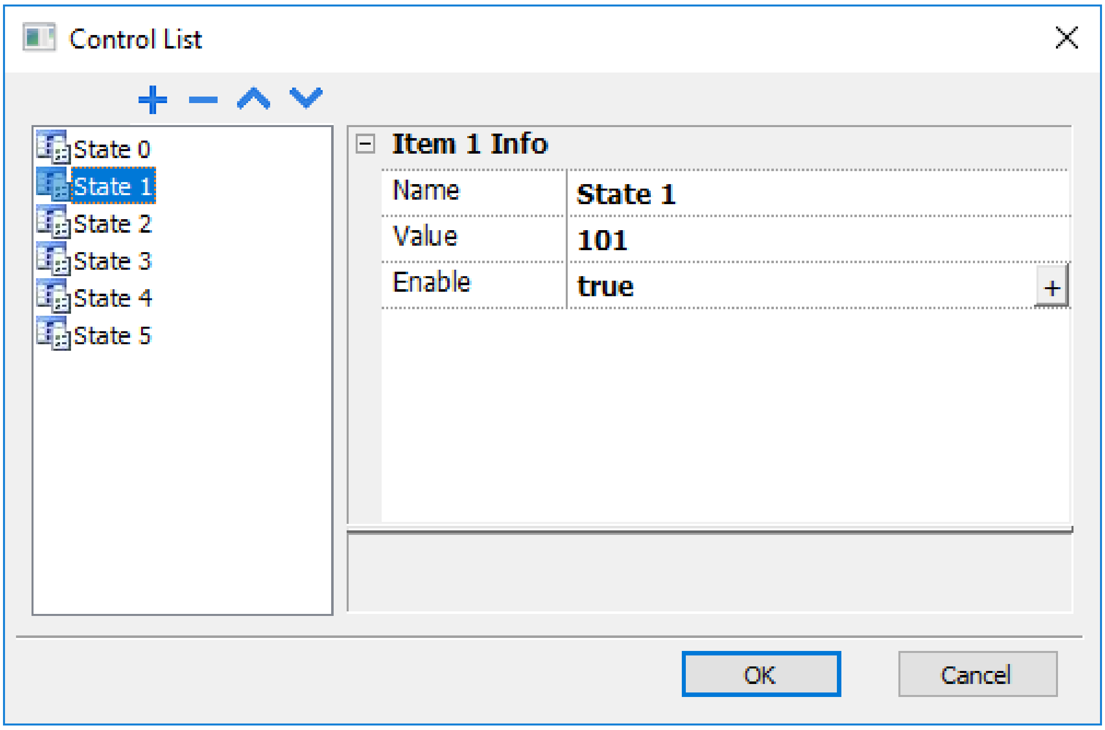
Two types of control lists are available:

- 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.



Control List	
Value	
State	4
Selection	4
Write Mode	Write On Enter
Scrollbar Type	Gesture
Read Only	false
List Data	State 0,State 1,State 2
OnDataUpdate Action	
Keypad	None
List Display	

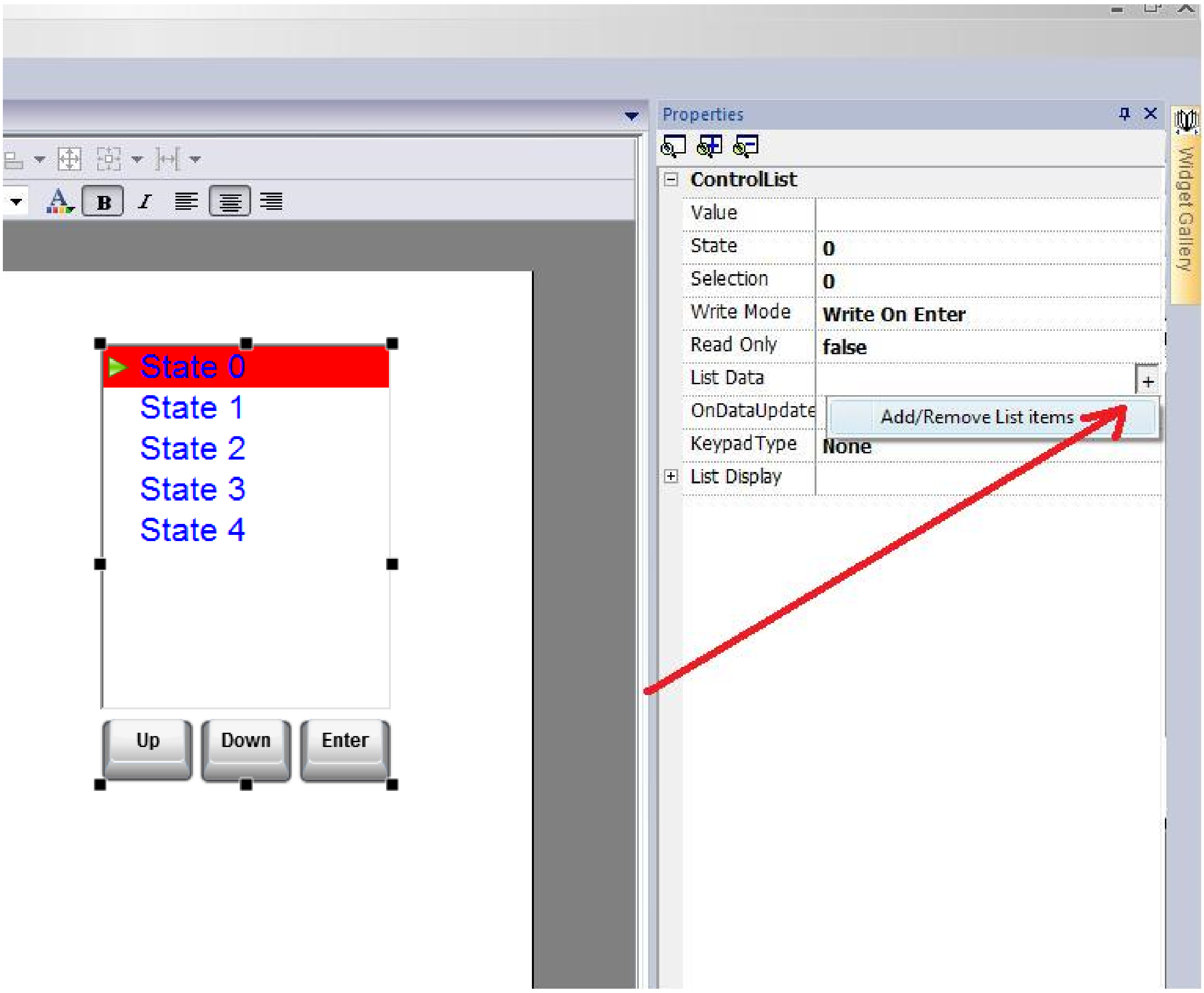
Parameter	Description
Value	The value corresponding to the status of the widget. If there is a tag attached to the value property, when loading the widget, the State will be aligned with the tag value.
State	State of widget. The widget highlights the item related to its State with a different background color (see "state color" in the properties of the widget).
Selection	State selection. The selected item will be displayed with a small triangle on the left side of the list.
Write Mode	Select the State update mode • **Write On Select**: The state is updated automatically to be aligned with the cursor position. • **Write On Enter**: The status is updated with the cursor position only when the user presses enter
Scrollbars Type	Select the scroll mode of the table • **Gesture**: Pan gesture can be used to smoothly scroll the table. • **Scrollbar**: Use the scrollbar to scroll the table

Parameter	Description
Read Only	Defines whether the list is only an indicator.
List Data	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.  Control List State 0 State 1 State 2 State 3 State 4 State 5 OK Cancel Item 1 Info Name: State 1 Value: 101 Enable: true

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** .



ControlList	
Value	
State	0
Selection	0
Write Mode	Write On Enter
Read Only	false
List Data	
OnDataUpdate	
KeypadType	None
List Display	

Manage list data items from JavaScrip 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_onMouseClick(me, eventInfo)
{
    var NewControlList = [["OFF",100,true],["ON",101,true],["MAN",102,true],
["AUTO",103,true]];
    var ControListWgt = page.getWidget("controlListBtn.controlList");
    ControListWgt.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_onMouseClick(me, eventInfo)
{
    var ControListWgt = page.getWidget("controlListBtn.controlList");
    var ListData = ControListWgt.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

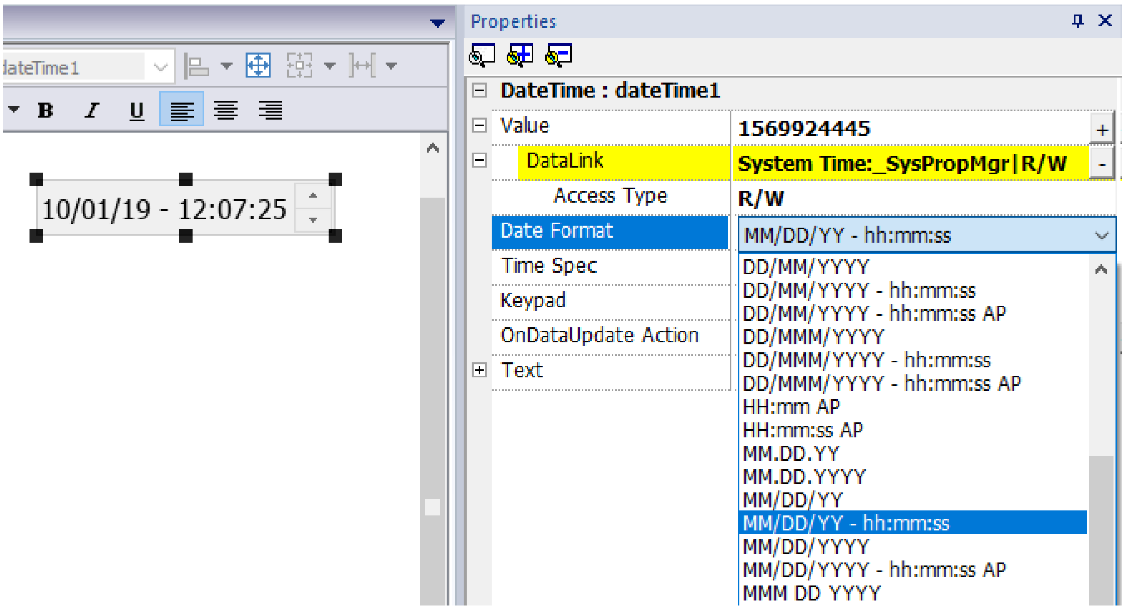
```
function controlListBtn_onDataUpdate(me, eventInfo)
{
    var ControListWgt = page.getWidget("controlListBtn.controlList");
    var State = ControListWgt.getProperty("state");
    project.setTag("State", State);
    return false;
}
```

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.



Properties

DateTime : dateTime1

Value	1569924445
DataLink	System Time:_SysPropMgr|R/W
Access Type	R/W
Date Format	MM/DD/YY - hh:mm:ss
Time Spec	
Keypad	
OnDataUpdate Action	
Text	

Time options

For the **Time Spec** property select which time the widget will show at runtime.

Option	Description
local	shows local time, the time of the HMI device where the project is running
global	shows Global Time (GMT)
server	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
d	the day as number without a leading zero (1 to 31)
dd	the day as number with a leading zero (01 to 31)
ddd	the abbreviated localized day name (e.g. 'Mon' to 'Sun')
dddd	the long localized day name (e.g. 'Monday' to 'Sunday')
M	the month as number without a leading zero (1-12)
MM	the month as number with a leading zero (01-12)
MMM	the abbreviated localized month name (e.g. 'Jan' to 'Dec')
MMMM	the long localized month name (e.g. 'January' to 'December')
yy	the year as two digit number (00-99)
yyyy	the year as four digit number

Time	Description
h	the hour without a leading zero (0 to 23 or 1 to 12 if AM/PM display)
hh	the hour with a leading zero (00 to 23 or 01 to 12 if AM/PM display)
m	the minute without a leading zero (0 to 59)
mm	the minute with a leading zero (00 to 59)
s	the whole second without a leading zero (0 to 59)
ss	the whole second with a leading zero where applicable (00 to 59)
AP or A	use AM/PM display. A/AP will be replaced by either "AM" or "PM"
ap or a	use am/pm display. a/ap will be replaced by either "am" or "pm"

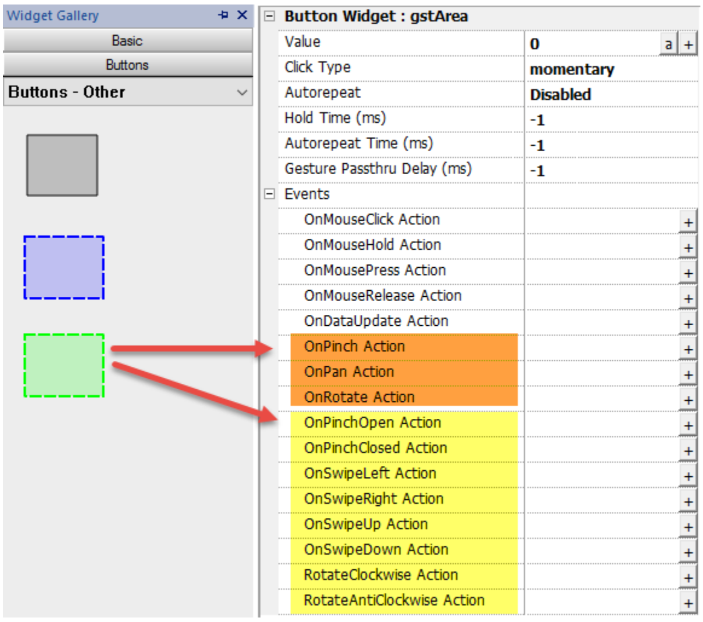
Regional Settings

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 91)

Gesture area widget

*Path: **Widget Gallery**> **Buttons**> **Others***

Gesture Area Widget is a hotspot button that generates gesture events.



Button Widget : gstArea

Value	0
Click Type	momentary
Autorepeat	Disabled
Hold Time (ms)	-1
Autorepeat Time (ms)	-1
Gesture Passthru Delay (ms)	-1
Events	
OnMouseClick Action	
OnMouseHold Action	
OnMousePress Action	
OnMouseRelease Action	
OnDataUpdate Action	
OnPinch Action	
OnPan Action	
OnRotate Action	
OnPinchOpen Action	
OnPinchClosed Action	
OnSwipeLeft Action	
OnSwipeRight Action	
OnSwipeUp Action	
OnSwipeDown Action	
RotateClockwise Action	
RotateAntiClockwise Action	

Gesture Events	Description
OnSwipeLeft **OnSwipeRight**	An event is release when swipe gesture is detected
OnPinchOpen **OnPinchClose**	An event is release when pinch gesture is detected
RotateClockwise **RotateAntiClockwise**	An event is release when rotate gesture is detected
OnPan **OnPinch** **OnRotate**	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. ⚠ **WARNING: Only multi touch HMI devices can generate OnPinch and OnRotate events**

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
me	Object triggering the event.
eventInfo	**id** = Gesture id; it is used to identify different gestures. **running** = True except for last event delivered to notify gesture completion. **dx** = Total X axis movement in screen pixel units from initial touch position . **dy** = 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
me	Object triggering the event
eventInfo	**id** = Gesture id; it is used to identify different gestures. **running** = True except for last event delivered to notify gesture completion. **dx** = 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. **dy** = 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
me	Object triggering the event
eventInfo	**id** = Gesture id; it is used to identify different gestures. **running** = True except for last event delivered to notify gesture completion. **drot** = How many degrees (0/360) have been added since the previous event. **trot** = 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
Gesture Passthru Enabled	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. **default** = Use the value defined in the project properties. See "Project" on page 86 **true** = Gesture passthru enabled **false** = Gesture passthru disabled
Gesture Passthru Delay (ms)	The time that must be waited for to send the command to the underlying object 0/500 mSec -1 Use the delay defined in the project properties. See "Project" on page 86

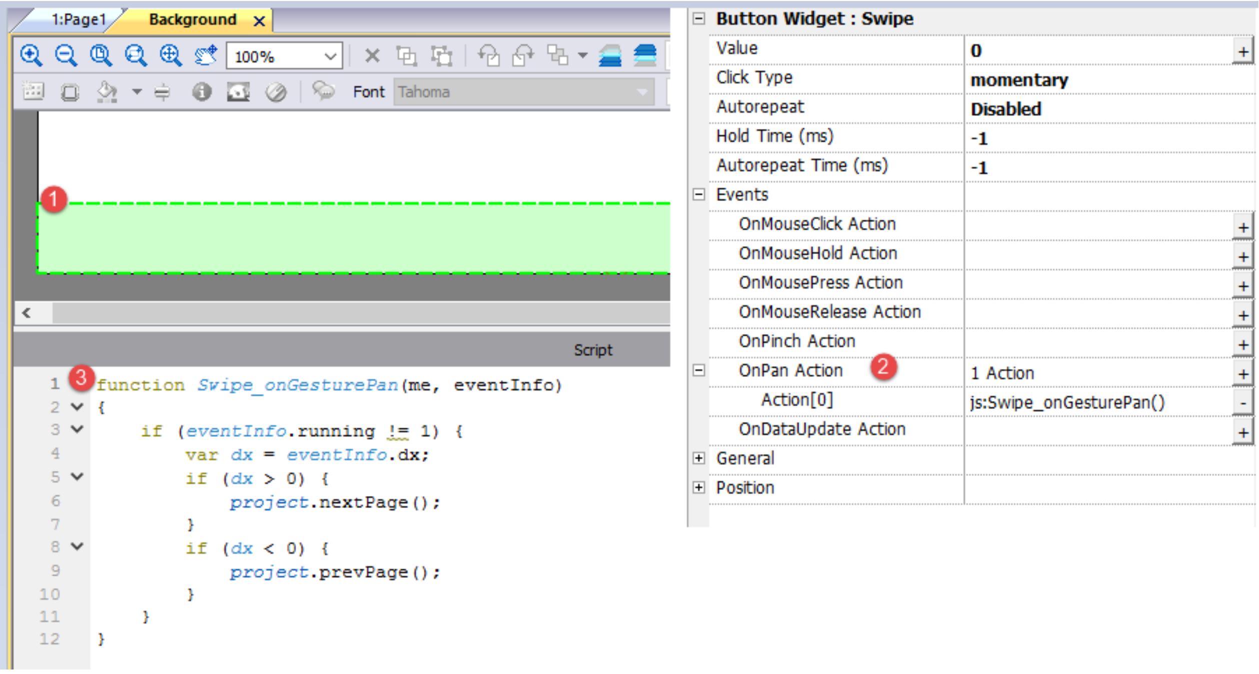
Examples of using gesture events in association with JavaScript

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



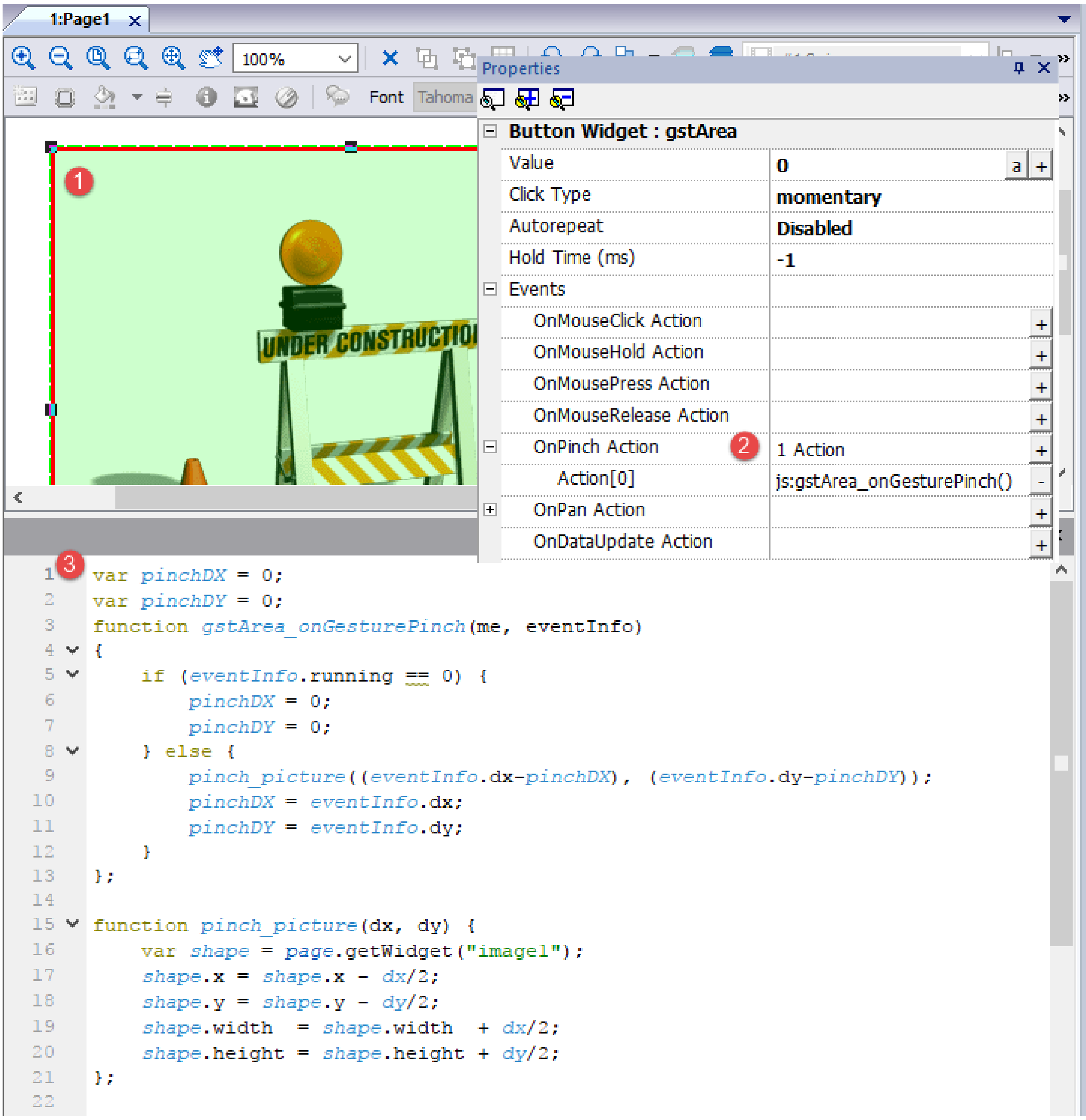
```
function Swipe_onGesturePan(me, eventInfo)
{
    if (eventInfo.running != 1) {
        var dx = eventInfo.dx;
        if (dx > 0) {
            project.nextPage();
        }
        if (dx < 0) {
            project.prevPage();
        }
    }
}
```

Button Widget : Swipe	
Value	0
Click Type	momentary
Autorepeat	Disabled
Hold Time (ms)	-1
Autorepeat Time (ms)	-1
Events	
OnMouseClick Action	
OnMouseHold Action	
OnMousePress Action	
OnMouseRelease Action	
OnPinch Action	
OnPan Action	1 Action
Action[0]	js:Swipe_onGesturePan()
OnDataUpdate Action	
General	
Position	

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



Button Widget : gstArea

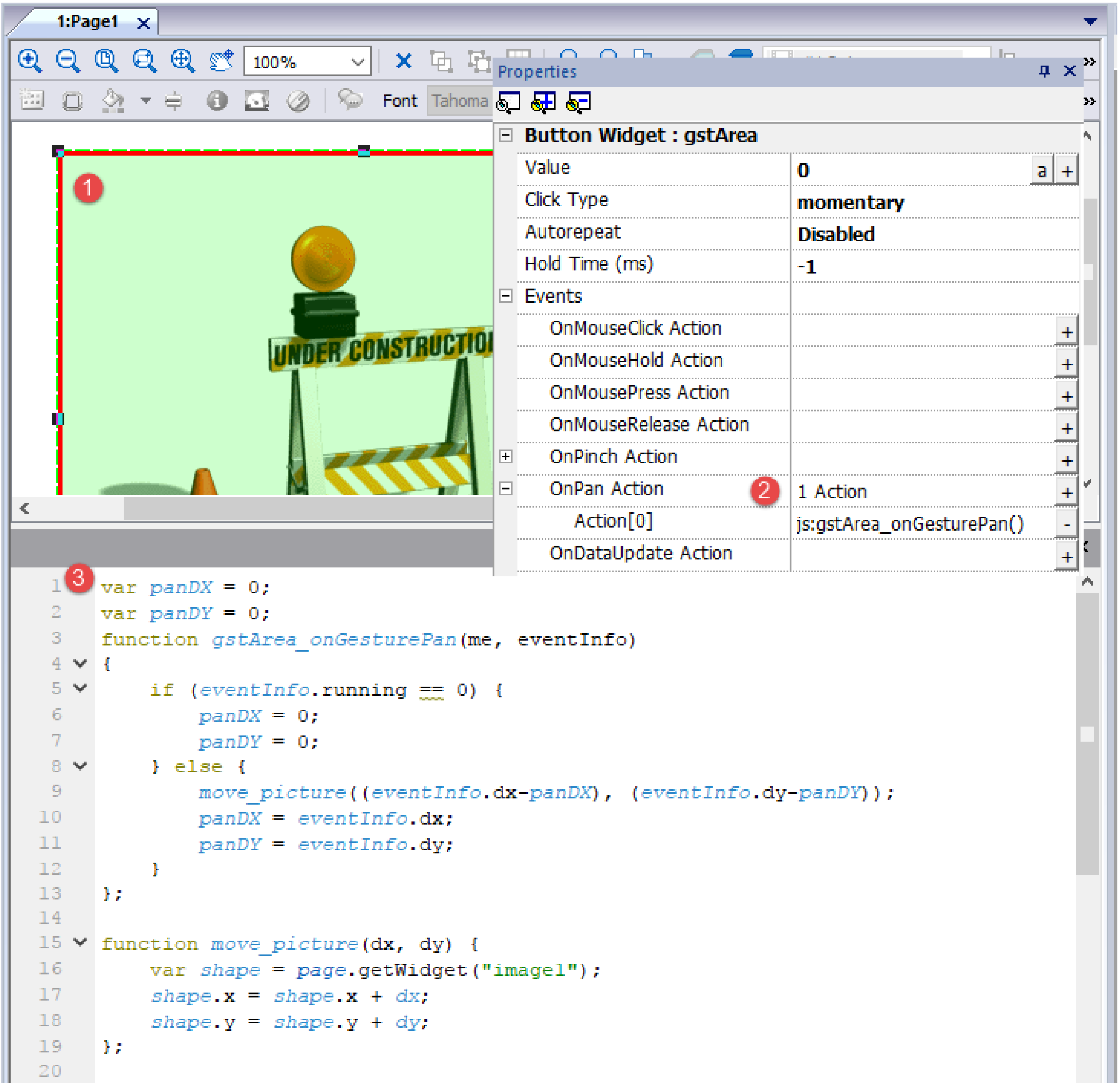
Value	0
Click Type	momentary
Autorepeat	Disabled
Hold Time (ms)	-1
Events	
OnMouseClick Action	
OnMouseHold Action	
OnMousePress Action	
OnMouseRelease Action	
OnPinch Action	1 Action
Action[0]	js:gstArea_onGesturePinch()
OnPan Action	
OnDataUpdate Action	

```
var pinchDX = 0;
var pinchDY = 0;
function gstArea_onGesturePinch(me, eventInfo)
{
    if (eventInfo.running == 0) {
        pinchDX = 0;
        pinchDY = 0;
    } else {
        pinch_picture((eventInfo.dx-pinchDX), (eventInfo.dy-pinchDY));
        pinchDX = eventInfo.dx;
        pinchDY = eventInfo.dy;
    }
};

function pinch_picture(dx, dy) {
    var shape = page.getWidget("image1");
    shape.x = shape.x - dx/2;
    shape.y = shape.y - dy/2;
    shape.width  = shape.width  + dx/2;
    shape.height = shape.height + dy/2;
};
```

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



Button Widget : gstArea	
Value	0
Click Type	momentary
Autorepeat	Disabled
Hold Time (ms)	-1
Events	
OnMouseClick Action	
OnMouseHold Action	
OnMousePress Action	
OnMouseRelease Action	
OnPinch Action	
OnPan Action	1 Action
Action[0]	js:gstArea_onGesturePan()
OnDataUpdate Action	

```
var panDX = 0;
var panDY = 0;
function gstArea_onGesturePan(me, eventInfo)
{
    if (eventInfo.running == 0) {
        panDX = 0;
        panDY = 0;
    } else {
        move_picture((eventInfo.dx-panDX), (eventInfo.dy-panDY));
        panDX = eventInfo.dx;
        panDY = eventInfo.dy;
    }
};

function move_picture(dx, dy) {
    var shape = page.getWidget("image1");
    shape.x = shape.x + dx;
    shape.y = shape.y + dy;
};
```

JavaScript function block widget

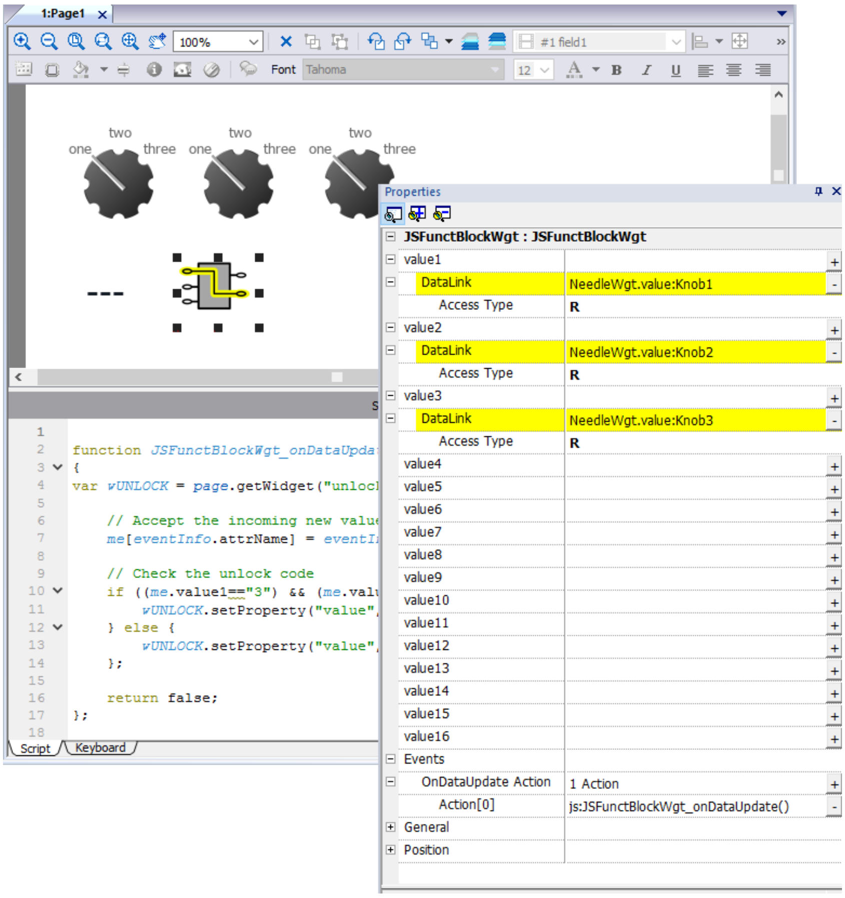
Path: ***Widget Gallery**> **Basic**> **JSFunctionBlock***

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

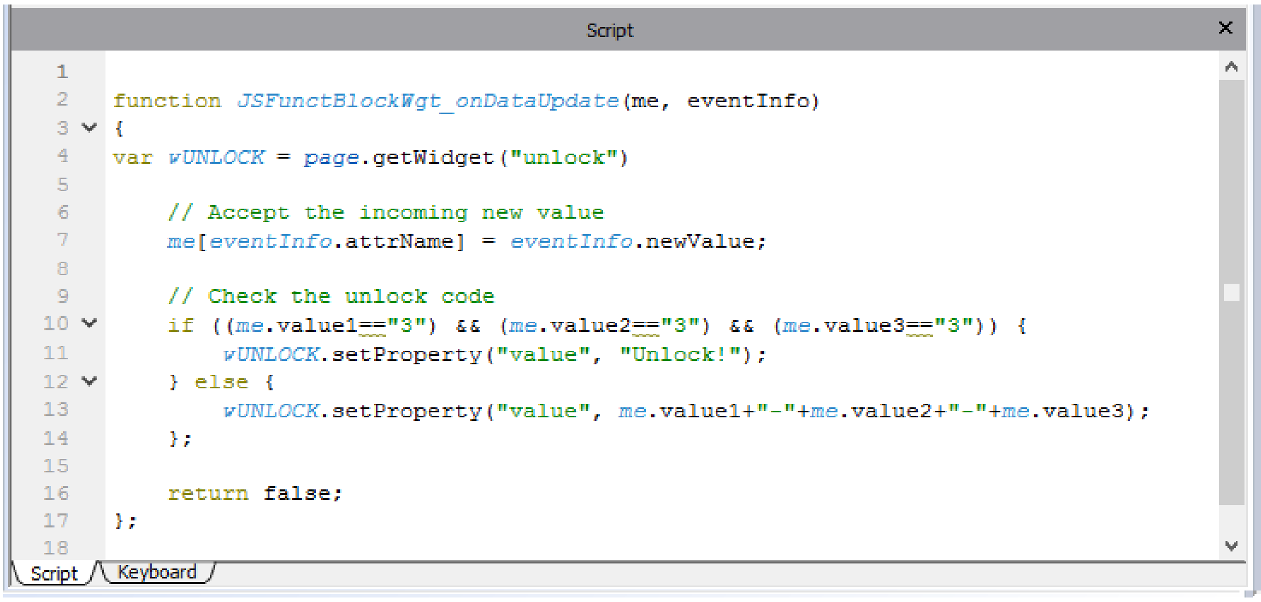
Note: This widget is rendered only in AGI Creator, and it is not rendered in the HMI device.

Example:
A JavaScript code that check the combination lock of three selectors



```
function JSFunctBlockWgt_onDataUpda
{
var wUNLOCK = page.getWidget("unlocl

    // Accept the incoming new value
    me[eventInfo.attrName] = eventI

    // Check the unlock code
    if ((me.value1=="3") && (me.valu
        wUNLOCK.setProperty("value",
    } else {
        wUNLOCK.setProperty("value",
    };

    return false;
};
```

JSFunctBlockWgt : JSFunctBlockWgt	
value1	
DataLink	NeedleWgt.value:Knob1
Access Type	R
value2	
DataLink	NeedleWgt.value:Knob2
Access Type	R
value3	
DataLink	NeedleWgt.value:Knob3
Access Type	R
value4	
value5	
value6	
value7	
value8	
value9	
value10	
value11	
value12	
value13	
value14	
value15	
value16	
Events	
OnDataUpdate Action	1 Action
Action[0]	js:JSFunctBlockWgt_onDataUpdate()
General	
Position	



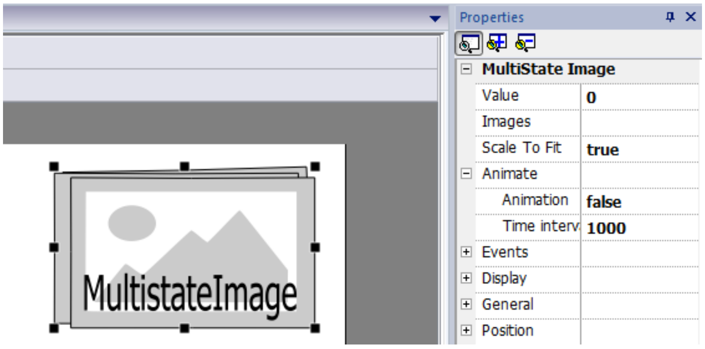
```
function JSFunctBlockWgt_onDataUpdate(me, eventInfo)
{
var wUNLOCK = page.getWidget("unlock")

    // Accept the incoming new value
    me[eventInfo.attrName] = eventInfo.newValue;

    // Check the unlock code
    if ((me.value1=="3") && (me.value2=="3") && (me.value3=="3")) {
        wUNLOCK.setProperty("value", "Unlock!");
    } else {
        wUNLOCK.setProperty("value", me.value1+"-"+me.value2+"-"+me.value3);
    };

    return false;
};
```

See "Widget events" on page 474 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.



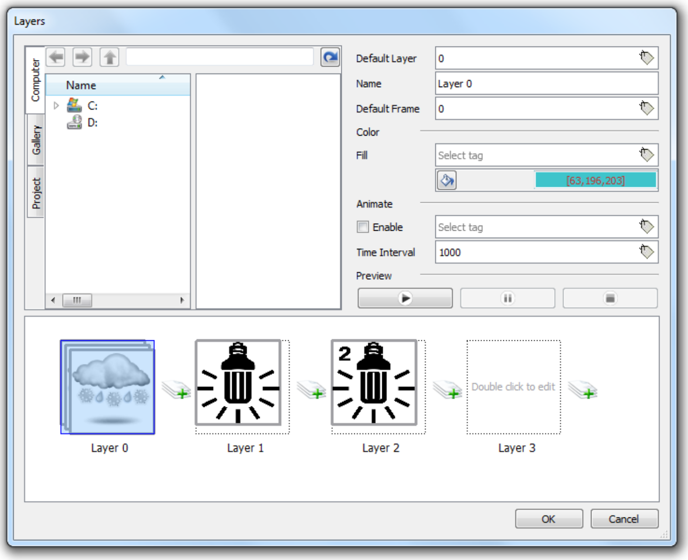
MultiState Image	
Value	0
Images	
Scale To Fit	true
Animate	
Animation	false
Time interv	1000
Events	
Display	
General	
Position	

Parameter	Description
Value	Index of image to display. For example, set Value=0, to display the image with index 0 in the image collection.
Images	Images collection with associated index.
Animate	Set to true, to enable a slide show.
Time interval	Interval between images in the slide show.

Multistate Image Multilayer widget

Path: ***Widget Gallery****>* ***Basic****>* ***Images***

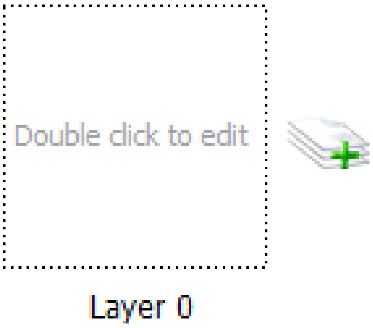
Use this widget to create different animations and select the most suitable at runtime.



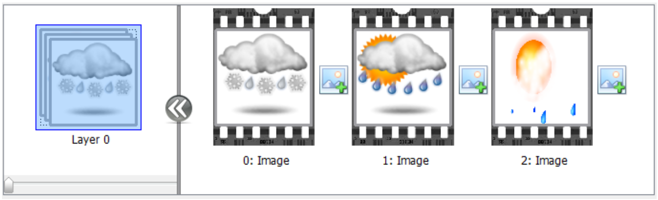
Default Layer: 0
Name: Layer 0
Default Frame: 0
Fill: Select tag [63,196,203]
☐ Enable: Select tag
Time Interval: 1000

Setting up widget layers

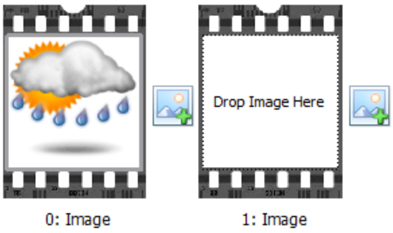
1. Open the **Layers** dialog from the **Properties** pane.
2. Click **+** to add as many layers as you need.



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.

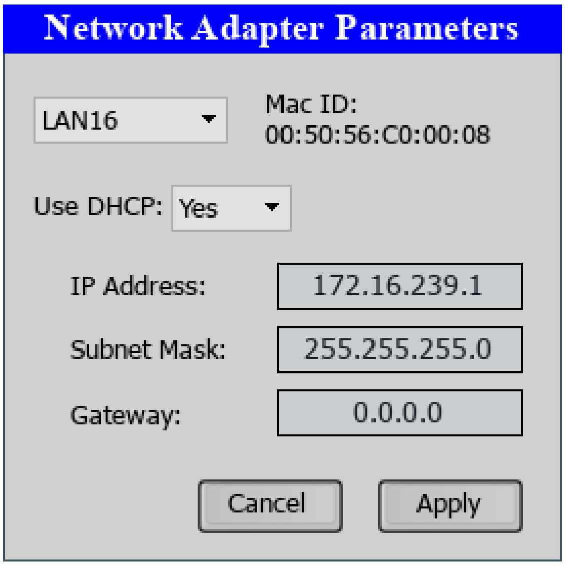
Parameter	Description
Default Layer	Layer shown at runtime.
Name	Name of selected layer.
Default Frame	Frame shown when current layer is displayed.
Color / Fill	Fill color for images of current layer.
Animate	Enables slide show for active layer. Animations can be started/stopped at runtime attaching it to a tag.
Time Interval	Time interval of slide show, if enabled.
Preview	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

LAN16 Mac ID: 00:50:56:C0:00:08

Use DHCP: Yes

IP Address: 172.16.239.1

Subnet Mask: 255.255.255.0

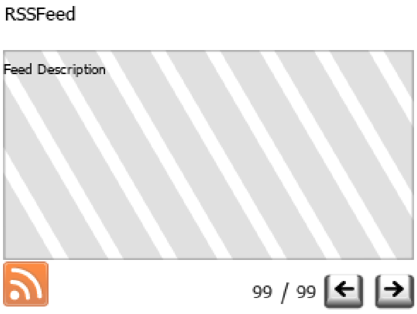
Gateway: 0.0.0.0

The system variable Network->Status contains the result of the last operation performed by the IP Widget (see "Network variables" on page 140 for details)

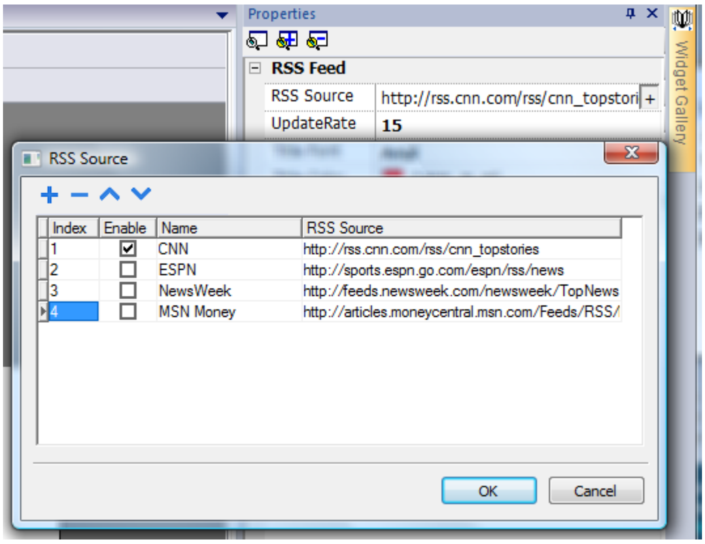
RSS Feed widget

*Path: **Widget Gallery**> **Media**> **RSSFeed Source***

Use this widget to display on the HMI device your favorite RSS feeds directly from the Internet.



Parameter	Description
RSS Source	Feed URL Note: Feed sources cannot be modified at runtime.
UpdateRate	Refresh time



RSS Feed

RSS Source	http://rss.cnn.com/rss/cnn_topstori
UpdateRate	15

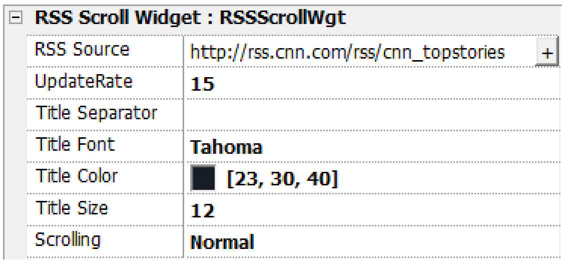
Index	Enable	Name	RSS Source
1	☑	CNN	http://rss.cnn.com/rss/cnn_topstories
2	□	ESPN	http://sports.espn.go.com/espn/rss/news
3	□	NewsWeek	http://feeds.newsweek.com/newsweek/TopNews
4	□	MSN Money	http://articles.moneycentral.msn.com/Feeds/RSS/

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.

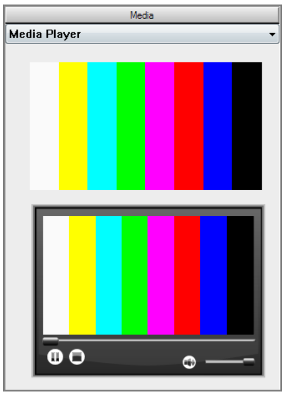
Parameter	Description
Scrolling	Scrolling speed
Title Separator	Separator character between highlights

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
Media Player List	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. 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. ⚠ **Ensure you have the commercial rights of the multimedia files.**
Loop Style	Define how the video is played. • **NoLoop**: plays all the videos in the playlist, then stops. • **LoopOne**: repeats the first video in the playlist. • **LoopAll**: repeats the entire playlist. • **Random**: plays the videos in a random order.

Note: The Media Player widget only works with some HMI devices (see "HMI devices capabilities" on page 543). It doesn’t work the AGI 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 543.

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:

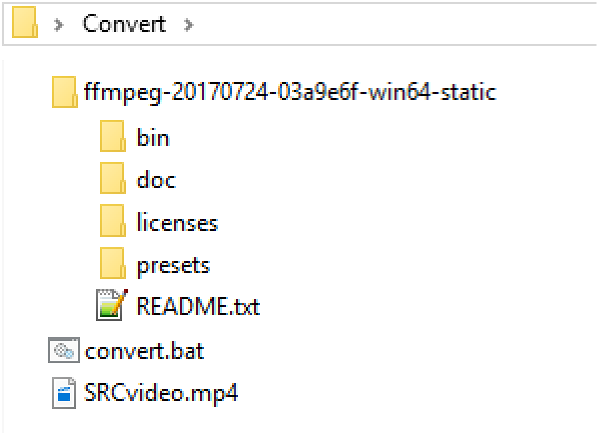
- Microsoft MPEG4 v3 using an AVI container.

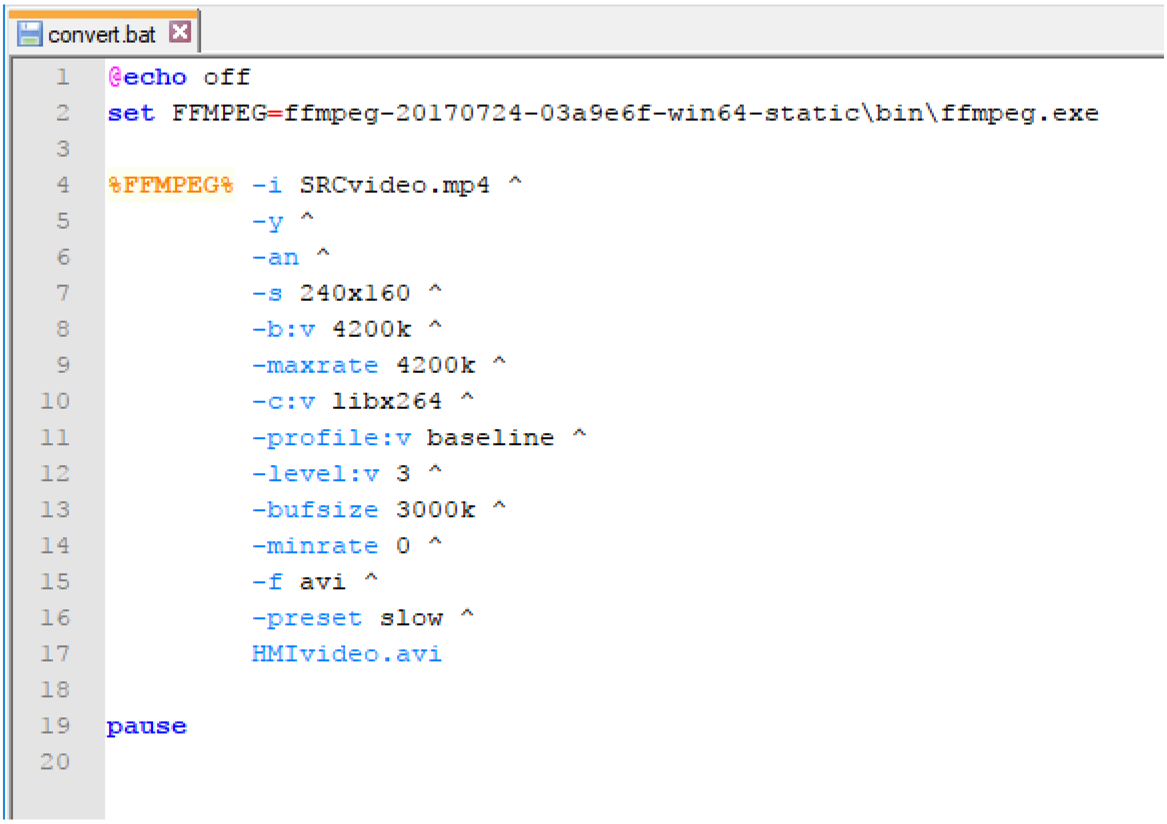
Be aware that video performance are depending from the chosen resolution, bit rate and device capabilities. If video rendering is not smooth, try to reduce the resolution or the bit rate of your video.

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) 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.





```
@echo off
set FFMPEG=ffmpeg-20170724-03a9e6f-win64-static\bin\ffmpeg.exe

%FFMPEG% -i SRCvideo.mp4 ^
         -y ^
         -an ^
         -s 240x160 ^
         -b:v 4200k ^
         -maxrate 4200k ^
         -c:v libx264 ^
         -profile:v baseline ^
         -level:v 3 ^
         -bufsize 3000k ^
         -minrate 0 ^
         -f avi ^
         -preset slow ^
         HMIvideo.avi

pause

```

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 AGI SW Pack.

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 - noloop, 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 469 for details on how to work with JavaScript.

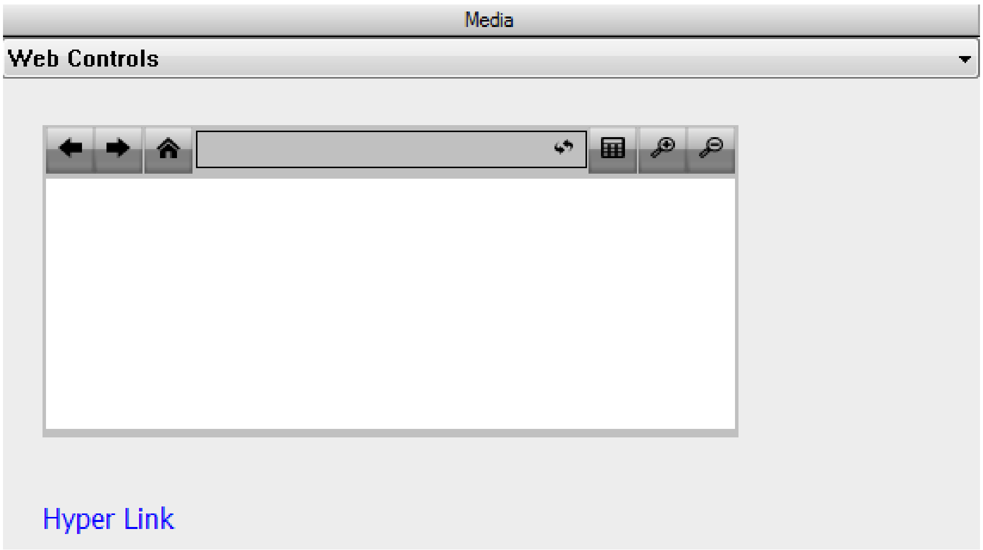
Browser widget

*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 engine.

Note: The WebKit library is available as a plugin (see "Plug-in" on page 85 for details) to download to the HMI Runtime only when required.

Important: This widget is not supported by MIPS based devices.



Parameter	Description
Home Page	Default URL to open when widget is shown on the page.
Zoom to Fit	Automatically scales content to the size of view area.
Time out	Page load timeout in seconds.
Clear History	Automatic history clear on load
Scroll	Shows/hides scrollbars
Show Progress cursor	Shows/hides loading cursor

This allows you to save around 3 MB of space if the widget is not required in your project.

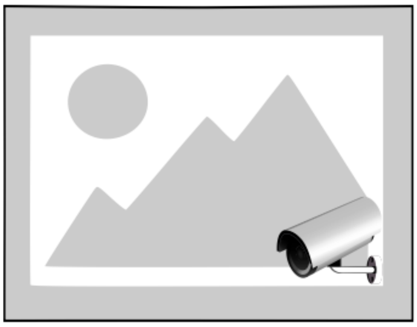
An **Hyper Link** widget is available to create pages hyperlinks. Once clicked these links notify to the browser widget that a particular web page is to be loaded.

Important: HTTPs protocol is not supported.

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
Camera URL	URL of the IP Camera when used in JPEG format.
Refresh Rate	Number of JPEG images for second allowed. Max rate = 1 fps.
User Name	Name of user allowed to access the camera. Set this parameter when access to the camera is password protected.
Password	Password to access the camera.
MJPEG Camera URL	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:

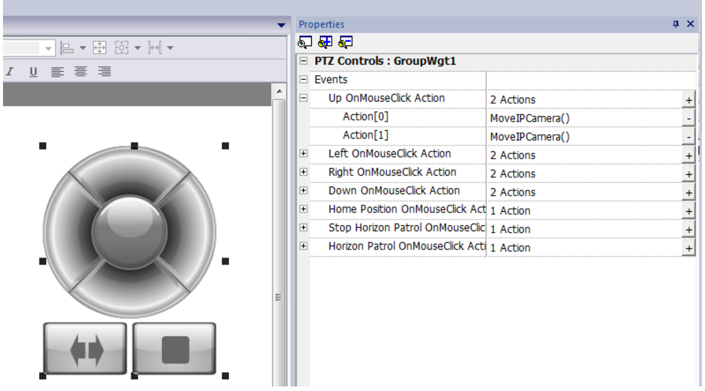
IPCamera	Protocol	URL
Apexis APM-J901-Z-WS PTZ IP Camera	MJPEG HTTP	http://{ip_address}/videostream.cgi http://{ip_address}/snapshot.cgi
AXIS M3027-PVE Network Camera	MJPEG HTTP	http://{ip_address}/axis-cgi/mjpg/video.cgi http://{ip_address}/axis-cgi/jpg/image.cgi
DAHUA DH-IPC-HD2100P-080B 1.3mp Outdoor Vandalproof	HTTP	http://{ip_address}:9988/onvif/media_service/snapshot
D-Link DCS-5605 PTZ	MJPEG	http://{ip_address}/video/mjpg.cgi
D-Link DCS-900W IP Camera	MJPEG	http://{ip_address}/video.cgi
D-Link DCS-932L	MJPEG	http://{ip_address}/video.cgi

IPCamera	Protocol	URL
Edimax IC-7100P PTZ	MJPEG HTTP	http://{ip_address}/mjpg/video.mjpg http://{ip_address}/picture.jpg
Foscam FI8916W	MJPEG HTTP	http://{ip_address}/videostream.cgi http://{ip_address}/snapshot.cgi
Foscam FI9803 EP	MJPEG	http://{ip_address}:88/cgi-bin/CGIStream.cgi?cmd=GetMJStream&usr={user}&pwd={pass} NOTE: • port 88 may be different as per IP Camera settings • {user} = username defined into IP Camera settings • {pass} = password defined into IP Camera settings
Hamlet HNIPCAM IP Camera	MJPEG HTTP	http://{ip_address}/video.cgi http://{ip_address}/image.jpg
MOXA VPort 254 (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
NVS30 network video server	MJPEG HTTP	http://{ip_address}:8070/video.mjpeg http://{ip_address}/jpg/image.jpg
Panasonic WV-Series Network Camera	MJPEG	http://{ip_address}/cgi-bin/mjpeg
Ubiquiti UniFi Video Camera	HTTP	http://{ip_address}:7080/images/snapshot/camera/{camera_guid}?force=true NOTE: • {camera_guiID} can be found into IP Camera Webpage • port 7080 may be different as per IP Camera settings
Zavio F3210 2MP Day & Night Compact IP Came	MJPEG HTTP	http://{ip_address}/stream?uri=video.pro3 http://{ip_address}/cgi-bin/view/image?pro_0 NOTE: • MJPEG video streaming can be configured selecting "video profile 3" with 640x480 resolution into IP Camera settings.

PTZ Controls widget

PTZ (pan–tilt–zoom) cameras are cameras capable of remote directional and zoom control.

The PTZ Controls widget uses the MoveIPCamera action to send HTTP/cgi commands to the PTZ IP Camera.



Properties

PTZ Controls : GroupWgt1

Events	
Up OnMouseClick Action	2 Actions
Action[0]	MoveIPCamera()
Action[1]	MoveIPCamera()
Left OnMouseClick Action	2 Actions
Right OnMouseClick Action	2 Actions
Down OnMouseClick Action	2 Actions
Home Position OnMouseClick Act	1 Action
Stop Horizon Patrol OnMouseClic	1 Action
Horizon Patrol OnMouseClick Acti	1 Action

Parameter	Description
Camera URL	URL of IP Camera
User Name	Name of user allowed to access the camera. Set this parameter when access to the camera is password protected.
Password	Password to access the camera.
Command	Command to send to the PTZ controller (for example, decoder_control.cgi?command=0)

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.

To configure a table:

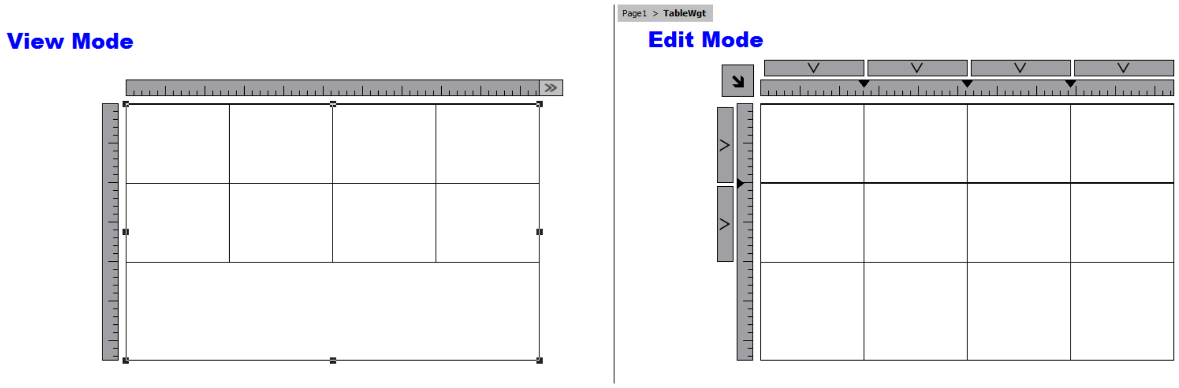
1. Put a table widget on the screen and configure the template of the table.
2. Add widgets into cells to configure one or more rows that will be used as row templates when the table will be filled with data provided from the data source.
3. Select a data source that will be used to fill the rows of the table
4. Define the links from widgets and data source.

Configure the table widget

Table widget has two states:

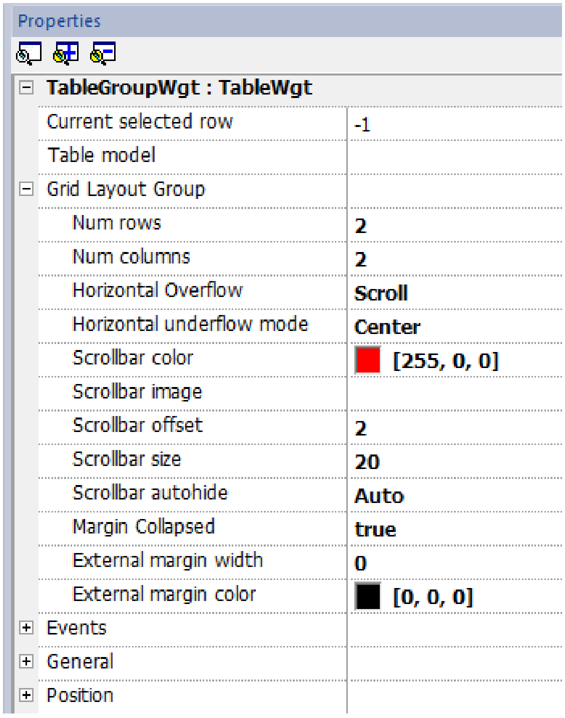
- View mode
- Edit mode.

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

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.



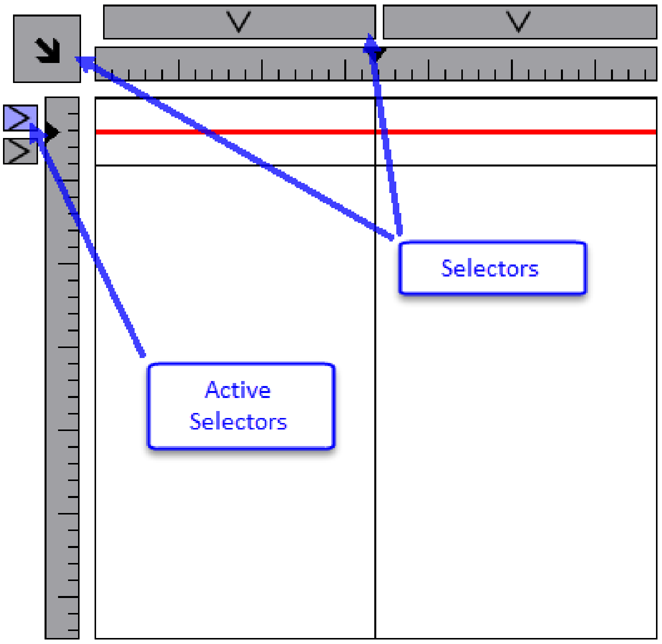
Properties

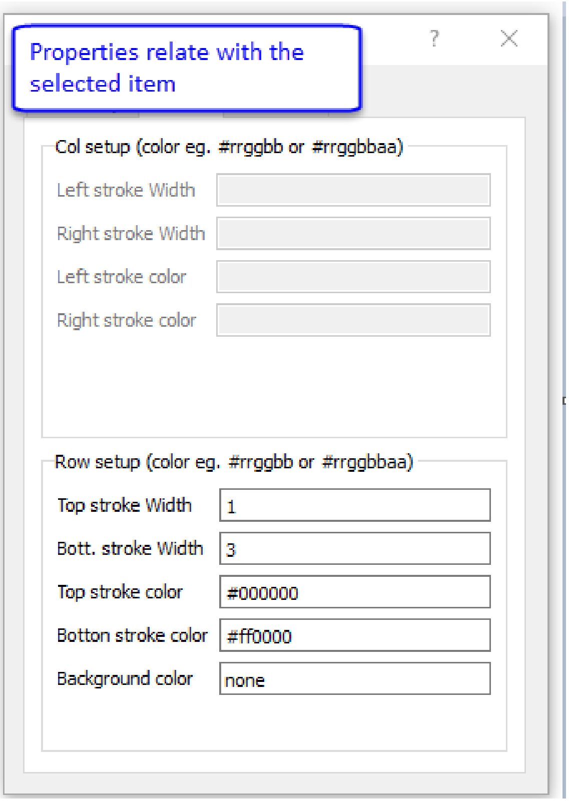
⊟ **TableGroupWgt : TableWgt**	
Current selected row	-1
Table model	
⊟ Grid Layout Group	
Num rows	**2**
Num columns	**2**
Horizontal Overflow	**Scroll**
Horizontal underflow mode	**Center**
Scrollbar color	🟥 **[255, 0, 0]**
Scrollbar image	
Scrollbar offset	**2**
Scrollbar size	**20**
Scrollbar autohide	**Auto**
Margin Collapsed	**true**
External margin width	**0**
External margin color	⬛ **[0, 0, 0]**
⊞ Events	
⊞ General	
⊞ Position	

Edit Mode

In edit mode, it is possible 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.





Col setup (color eg. #rrggbb or #rrggbbaa)

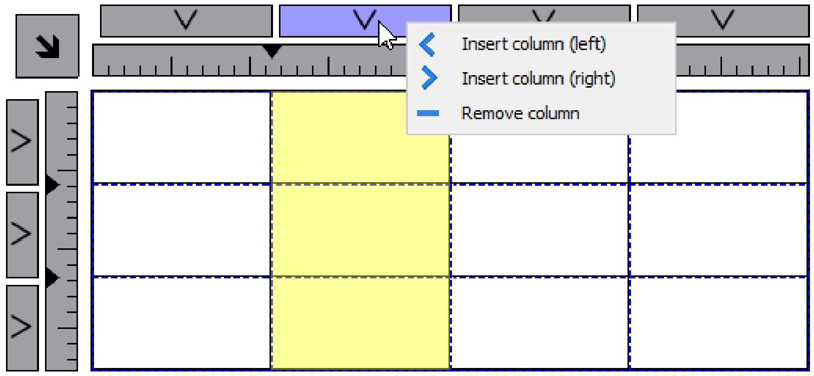
Left stroke Width	
Right stroke Width	
Left stroke color	
Right stroke color	

Row setup (color eg. #rrggbb or #rrggbbaa)

Top stroke Width	1
Bott. stroke Width	3
Top stroke color	#000000
Botton stroke color	#ff0000
Background color	none

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.



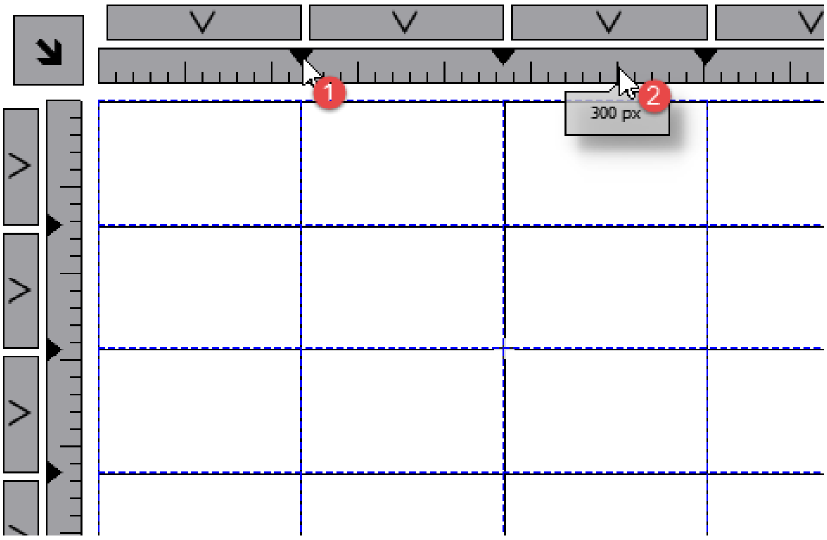
Merge or split rows or columns

To merge or split rows or columns, double click over the grid to enter in edit mode and move the cursor over the ribbons:

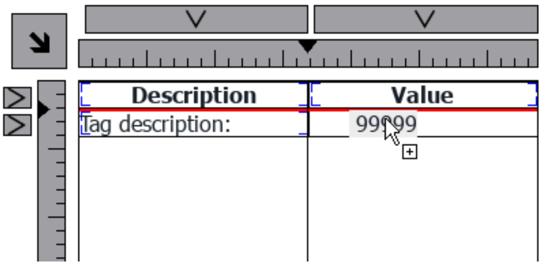
- Double click the black triangle to merge the two adjacent rows or columns (1)

  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.

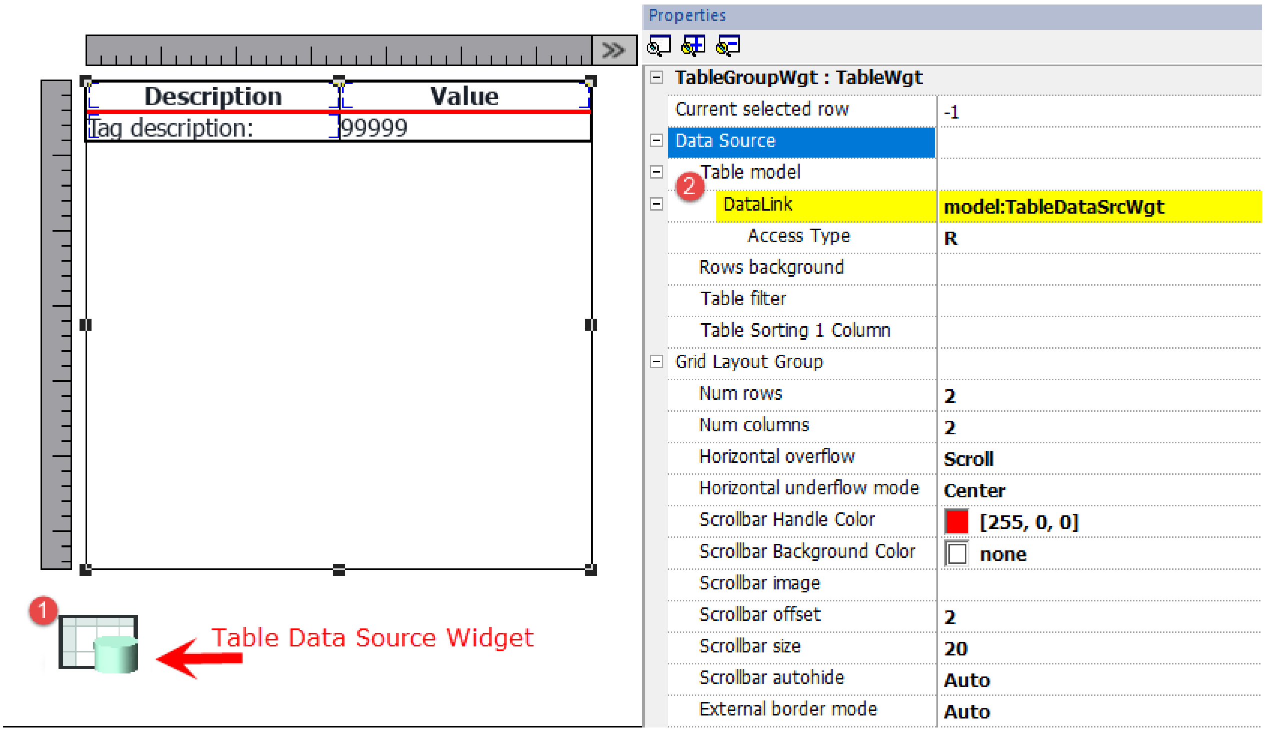
Configuring the data source

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**

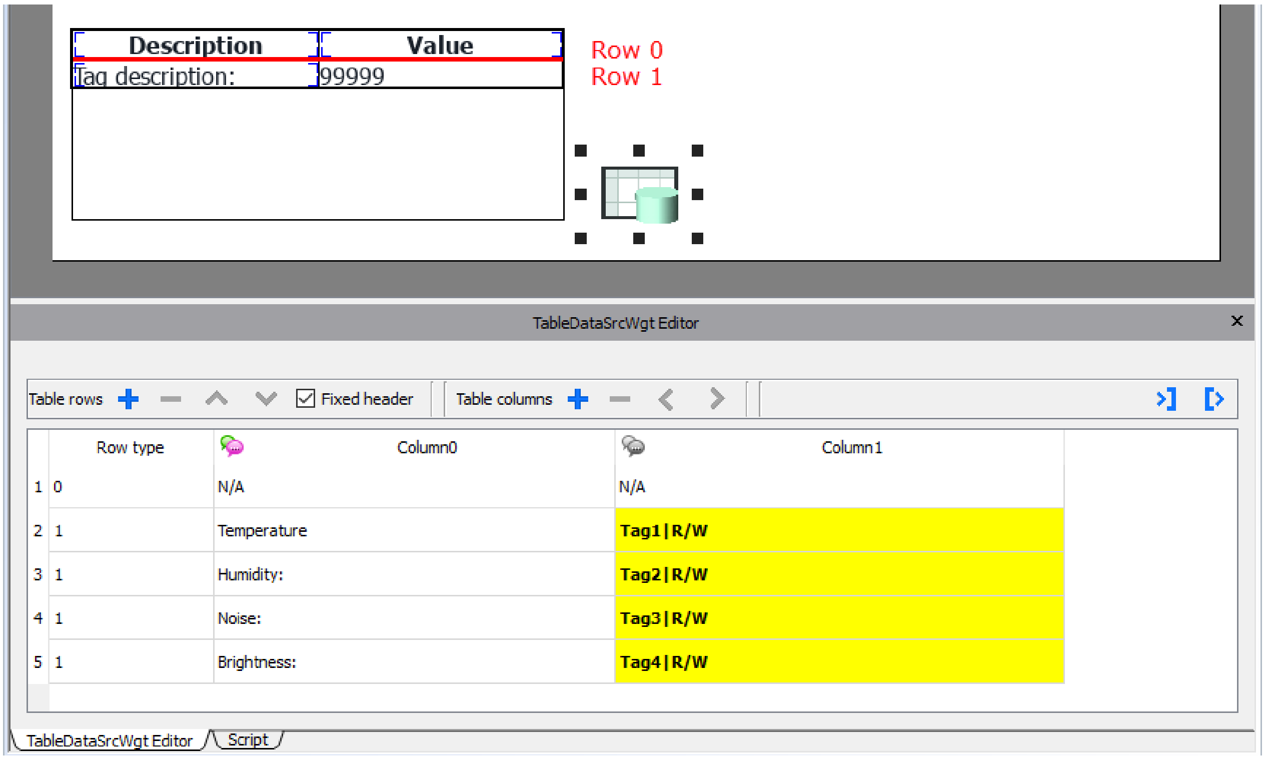
1. Drag and drop a *Table Data Source Widget* onto the page
2. Set the *Table Model* parameter to link at the data source.



TableGroupWgt : TableWgt	
Current selected row	-1
Data Source	
Table model	
DataLink	model:TableDataSrcWgt
Access Type	R
Rows background	
Table filter	
Table Sorting 1 Column	
Grid Layout Group	
Num rows	2
Num columns	2
Horizontal overflow	Scroll
Horizontal underflow mode	Center
Scrollbar Handle Color	🟥 [255, 0, 0]
Scrollbar Background Color	□ none
Scrollbar image	
Scrollbar offset	2
Scrollbar size	20
Scrollbar autohide	Auto
External border mode	Auto

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.

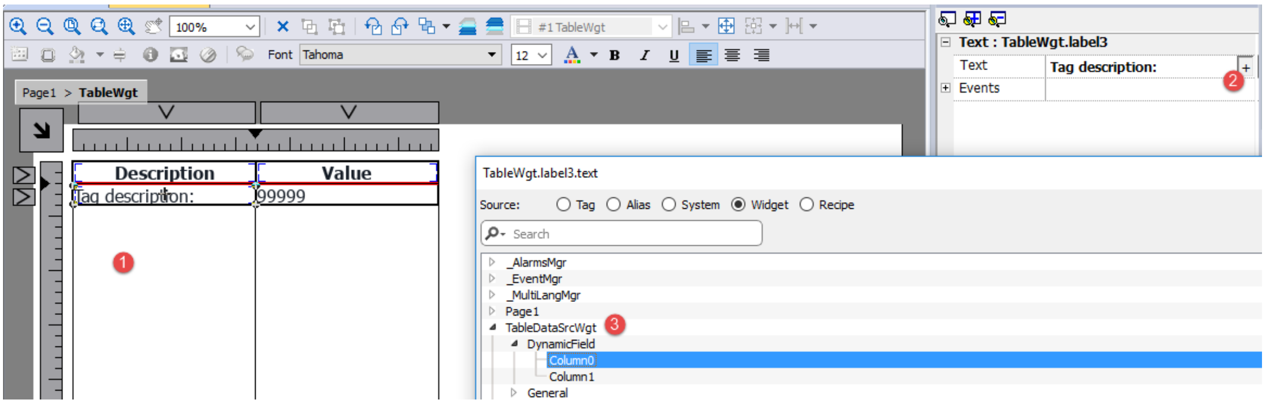


	Row type	Column0	Column1
1	0	N/A	N/A
2	1	Temperature	Tag1|R/W
3	1	Humidity:	Tag2|R/W
4	1	Noise:	Tag3|R/W
5	1	Brightness:	Tag4|R/W

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



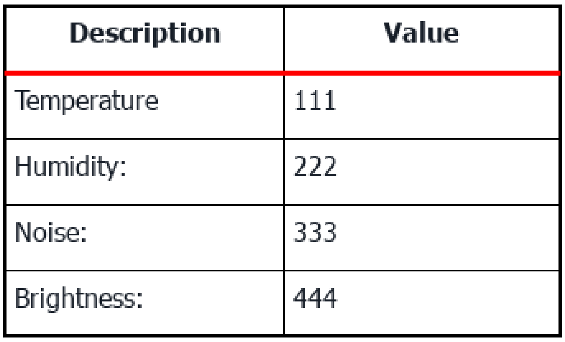
Text : TableWgt.label3

Text: Tag description:

TableWgt.label3.text

Source: ○ Tag ○ Alias ○ System ● Widget ○ Recipe

The below picture is showing how our example will be rendered at runtime



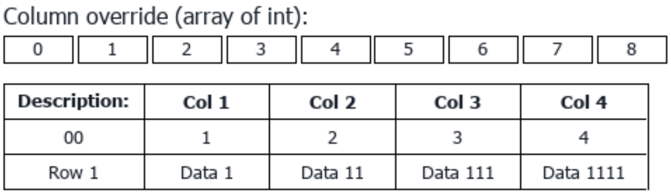
Description	Value
Temperature	111
Humidity:	222
Noise:	333
Brightness:	444

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" propert inside the Data Source properties panel (note the parameter is available only in advanced view).

Column override

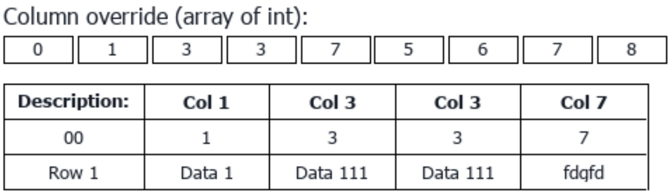
You can use an array of integers to define or modify the columns order at runtime. When you use this property, be sure to attach an array of integer and set the index to -1 (to select the entire array).



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):

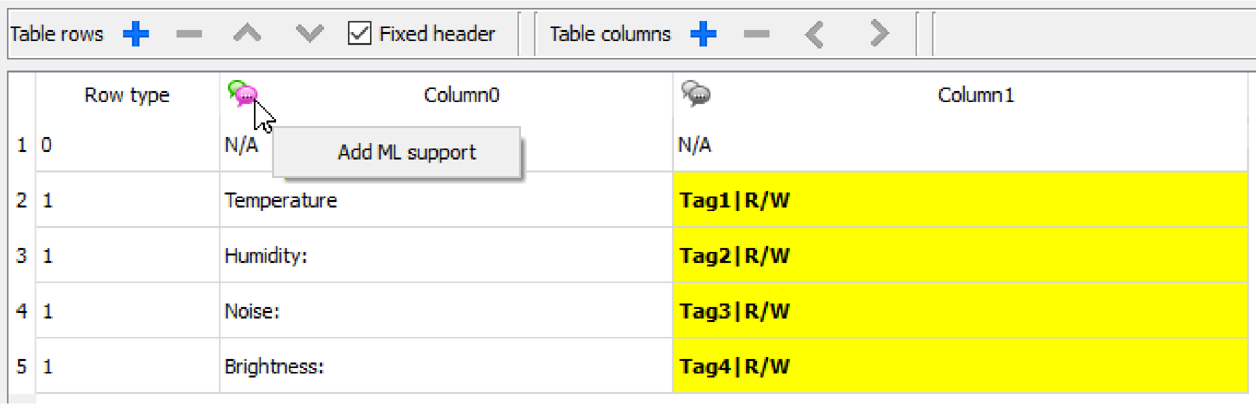
0	1	3	3	7	5	6	7	8

Description:	Col 1	Col 3	Col 3	Col 7
00	1	3	3	7
Row 1	Data 1	Data 111	Data 111	fdqfd

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.

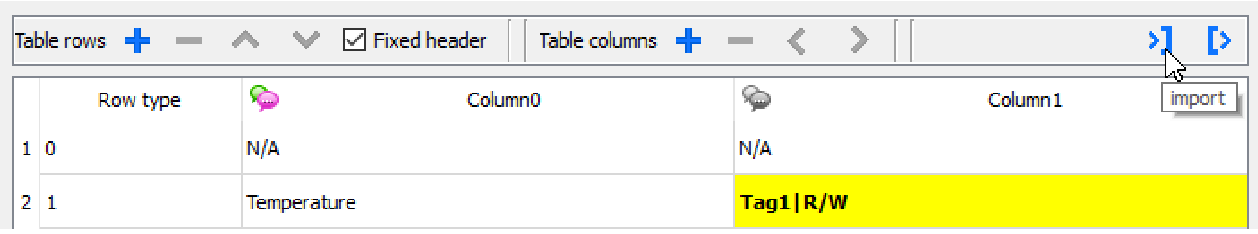
Avoid enabling the Multilanguage support when not necessary to better performance.



	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



	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
_c : "dl"	Identify the element as a Datalink
s : "_TagMgr"	Specify the source of data is the Tag Manager
a : "Tag1", i: 0, m:2	Specify tag name and index (necessary when the tag is an array) and the read/write mode

Parameter	Description
	• m=0 is Read Only • m=1 is Write Only • m=2 is Read/Write

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
_c : "ml"	Identify the element as a Multilanguage text
mltext : { ... }	List of couples: "ID Language":"Text" Example: • "en-US" : "Temperature:" • "it-IT" : "Temperatura:"

Example:

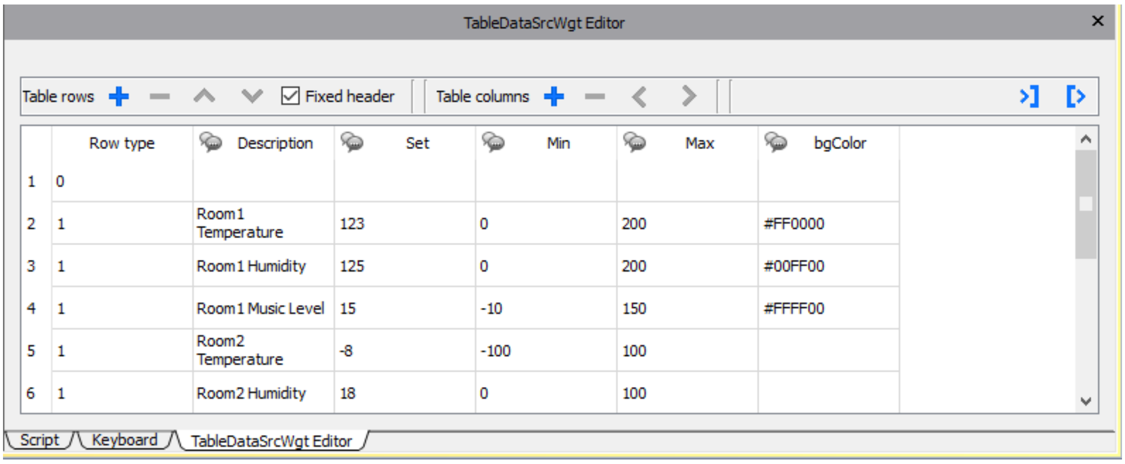
```
var row_data2 = {
    _t : 1,
    _v : [ { _c : "ml" , mltext : { "en-US" : "Temperature:",
                                    "it-IT" : "Temperatura:"}   },
           { _c : "dl" , s : "_TagMgr", a : "Tag1", i: 0, m : 2 }
         ]
}
```

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.

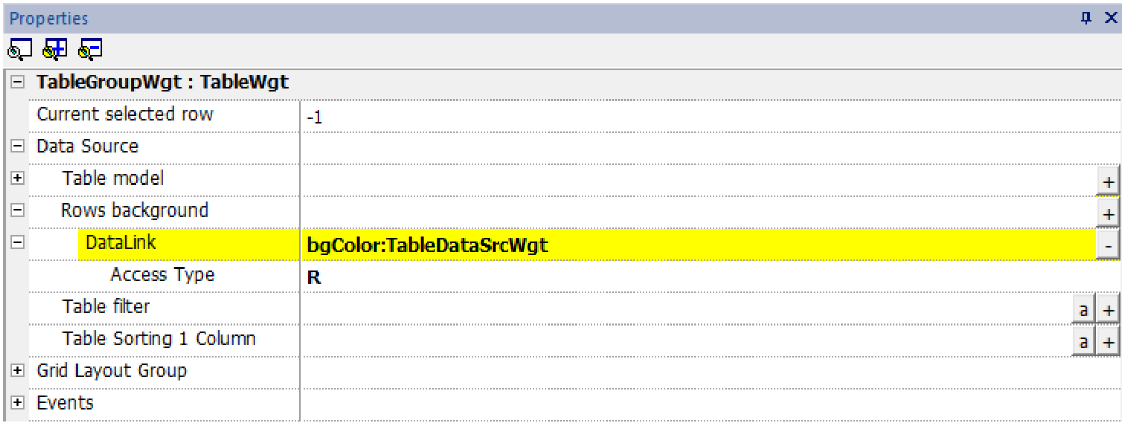
To configure the background color of the rows of the table:

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

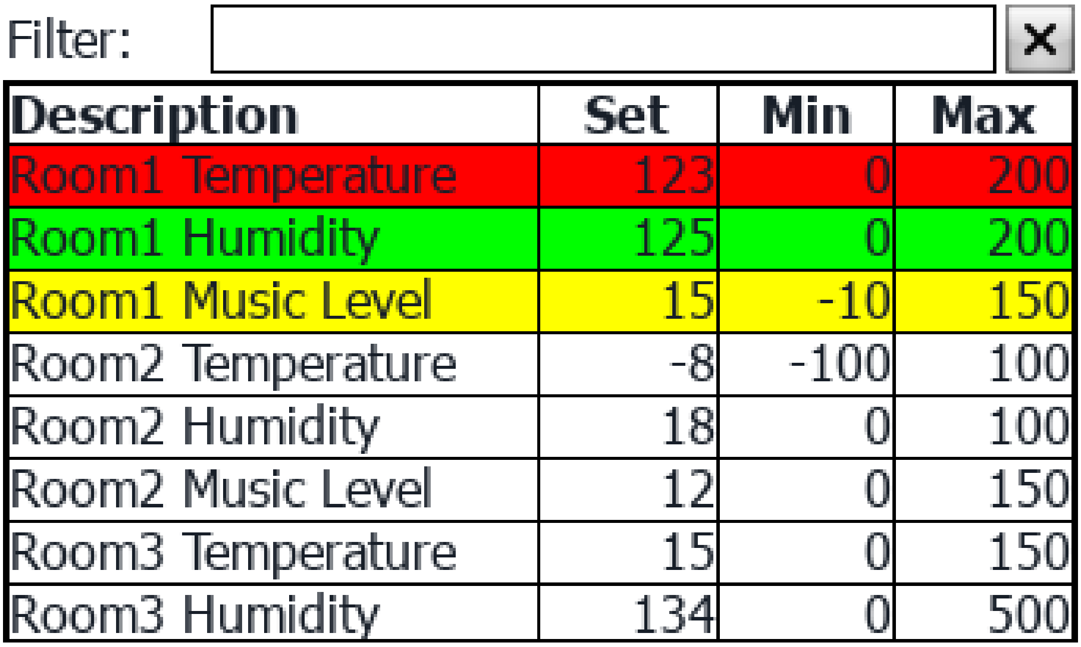


☑ Fixed header

	Row type	Description	Set	Min	Max	bgColor
1	0					
2	1	Room1 Temperature	123	0	200	#FF0000
3	1	Room1 Humidity	125	0	200	#00FF00
4	1	Room1 Music Level	15	-10	150	#FFFF00
5	1	Room2 Temperature	-8	-100	100	
6	1	Room2 Humidity	18	0	100	



Property	Value
TableGroupWgt : TableWgt	
Current selected row	-1
Data Source	
Table model	
Rows background	
DataLink	bgColor:TableDataSrcWgt
Access Type	R
Table filter	
Table Sorting 1 Column	
Grid Layout Group	
Events	



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 50 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.

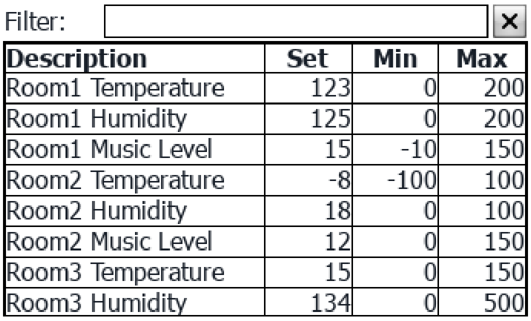
Example 1

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'))
```

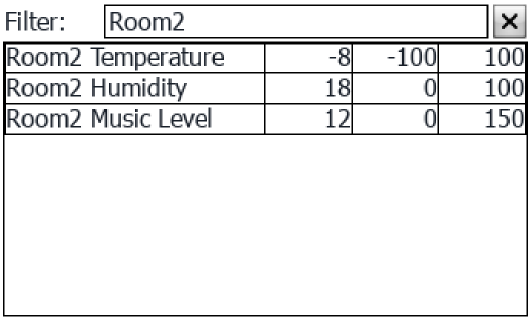
Where

- '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



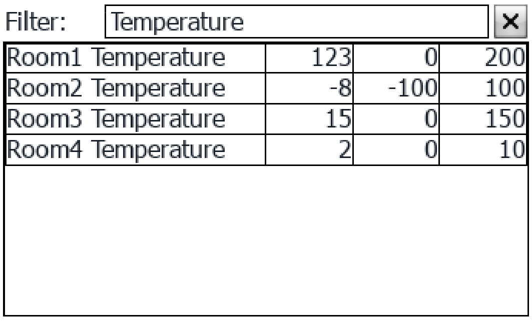
Filter: ✕

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



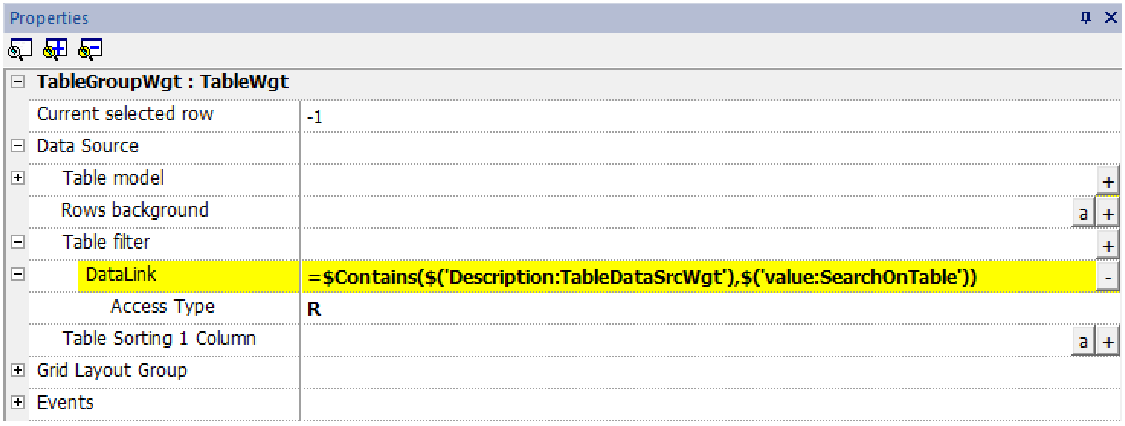
Filter: Room2 ✕

Room2 Temperature	-8	-100	100
Room2 Humidity	18	0	100
Room2 Music Level	12	0	150



Filter: Temperature ✕

Room1 Temperature	123	0	200
Room2 Temperature	-8	-100	100
Room3 Temperature	15	0	150
Room4 Temperature	2	0	10



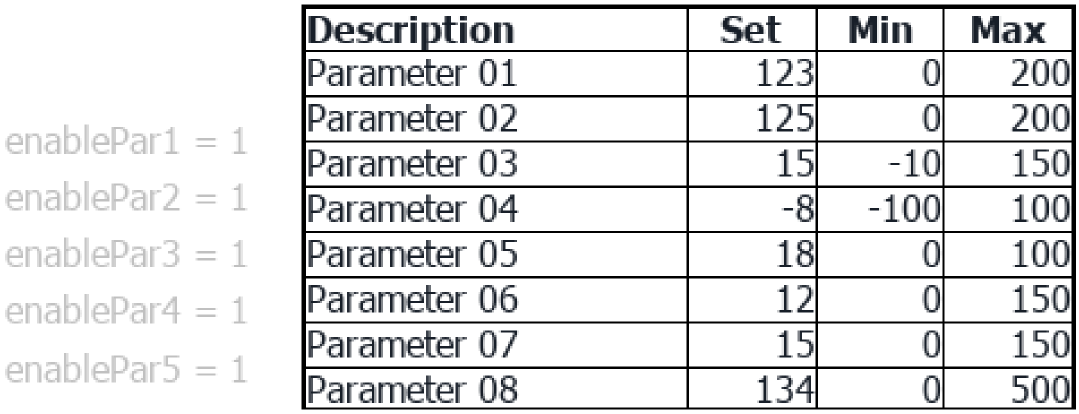
Properties

⊟ 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

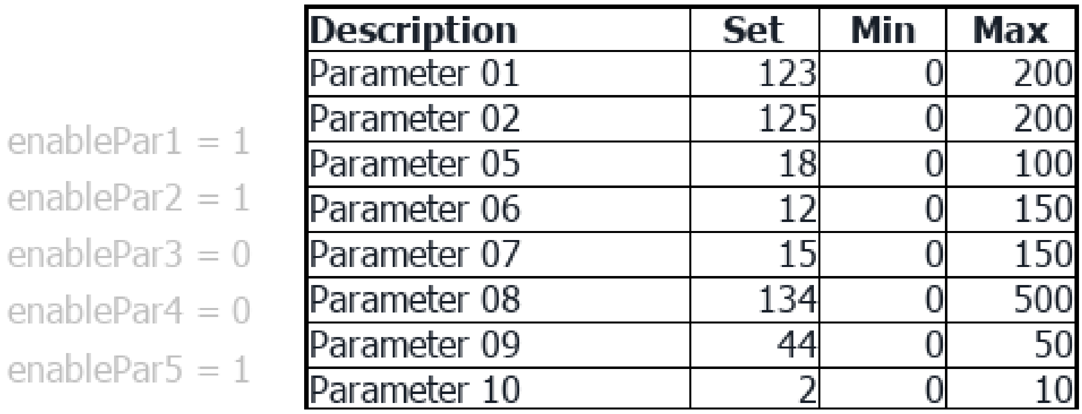
To use flags to define the parameters to expose inside the table:

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



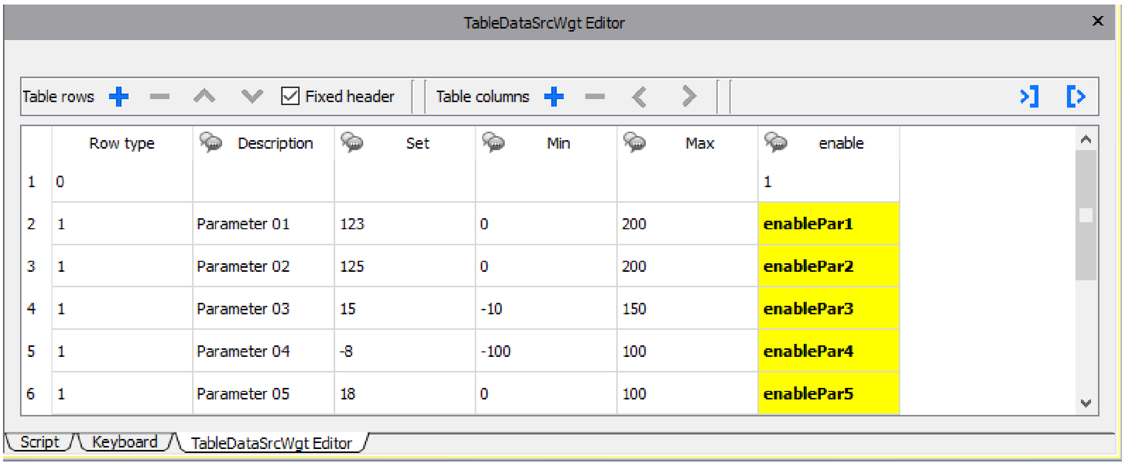
enablePar1 = 1
enablePar2 = 1
enablePar3 = 1
enablePar4 = 1
enablePar5 = 1

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

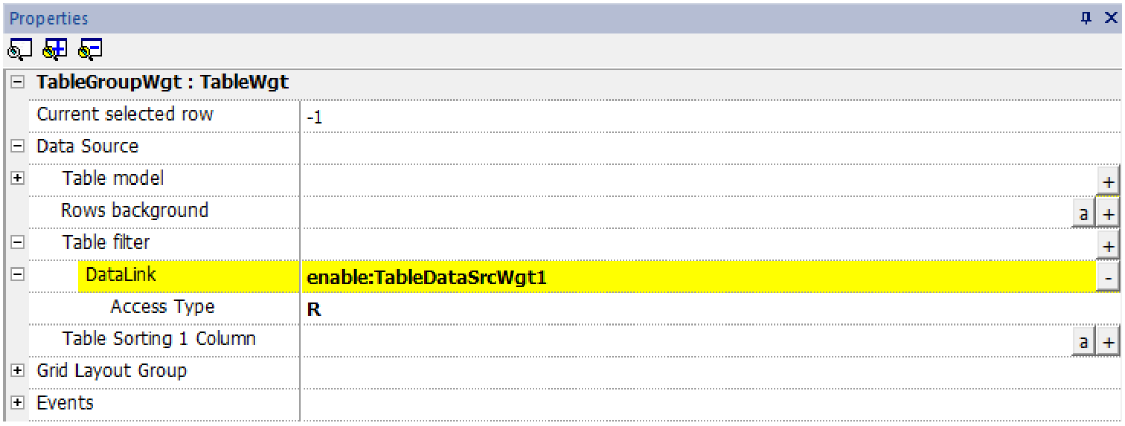


enablePar1 = 1
enablePar2 = 1
enablePar3 = 0
enablePar4 = 0
enablePar5 = 1

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



	Row type	Description	Set	Min	Max	enable
1	0					1
2	1	Parameter 01	123	0	200	enablePar1
3	1	Parameter 02	125	0	200	enablePar2
4	1	Parameter 03	15	-10	150	enablePar3
5	1	Parameter 04	-8	-100	100	enablePar4
6	1	Parameter 05	18	0	100	enablePar5

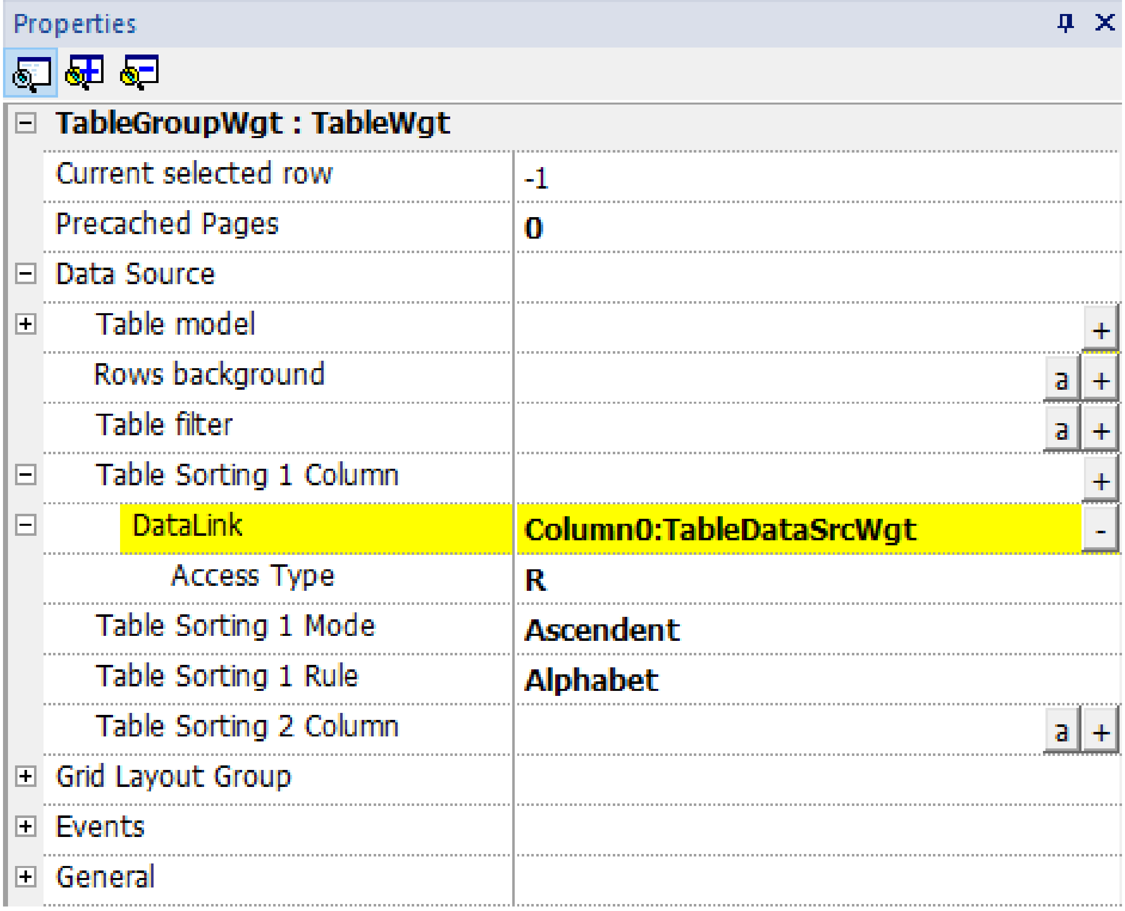


TableGroupWgt : TableWgt	
Current selected row	-1
Data Source	
Table model	
Rows background	
Table filter	
DataLink	enable:TableDataSrcWgt1
Access Type	R
Table Sorting 1 Column	
Grid Layout Group	
Events	

Table Sorting

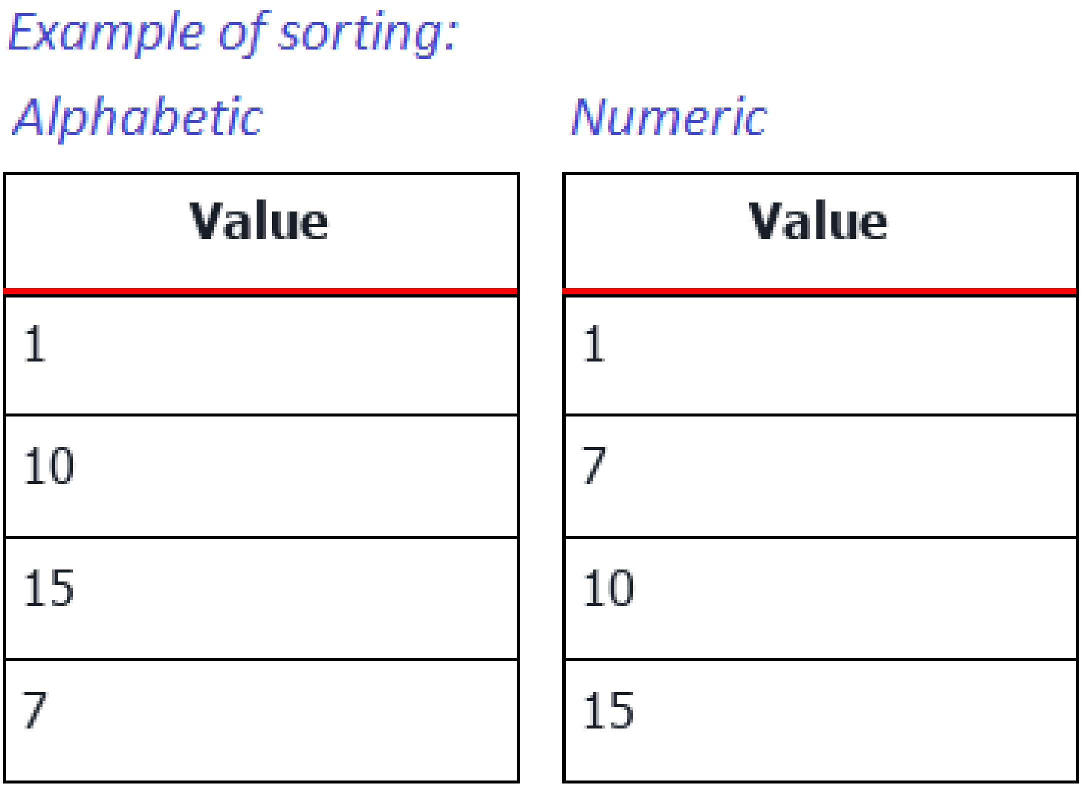
To sort the rows of the table, select the column of the Data Source that you want to use to sort the table

- Sorting mode can be Ascendent or Descendent
- Sorting Rule can be Alphabetic or Numeric



TableGroupWgt : TableWgt	
Current selected row	-1
Precached Pages	0
Data Source	
Table model	
Rows background	
Table filter	
Table Sorting 1 Column	
DataLink	Column0:TableDataSrcWgt
Access Type	R
Table Sorting 1 Mode	Ascendent
Table Sorting 1 Rule	Alphabet
Table Sorting 2 Column	
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

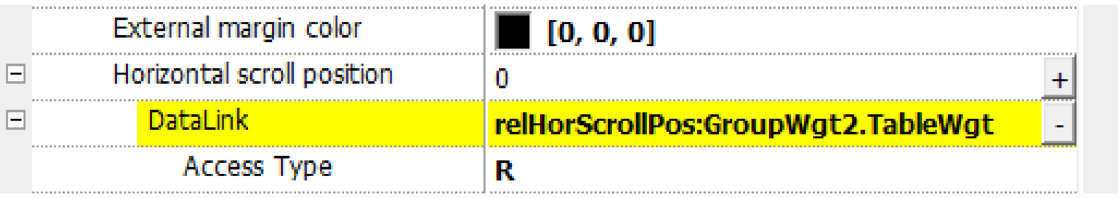
Numeric

Value
1
7
10
15

The table can be ordered even using the SetTableSortingColumn macro (see "SetTableSortingColumn" on page 222 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 +
⊟ DataLink	relHorScrollPos:GroupWgt2.TableWgt -
Access Type	R

Horizontal scroll position parameter is available only in Advanced Proprieties View mode

Precached Pages

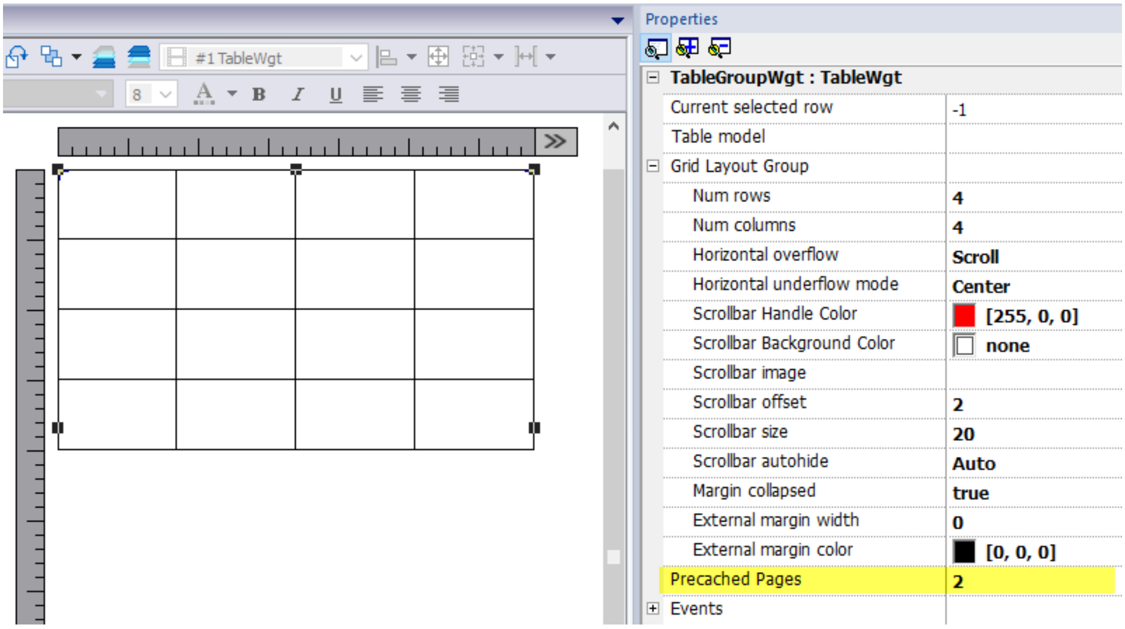
Normally the HMI Runtime retrieve only the data that will be visible into the display. To make table scrolling more pleasant, it may be useful to preload the data of the next and previous rows of the displayed ones. Using the Precached Pages parameter is possible define how many pages will be preloaded

- 0 = no pages preloaded
- N = number of pages to preoload

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)

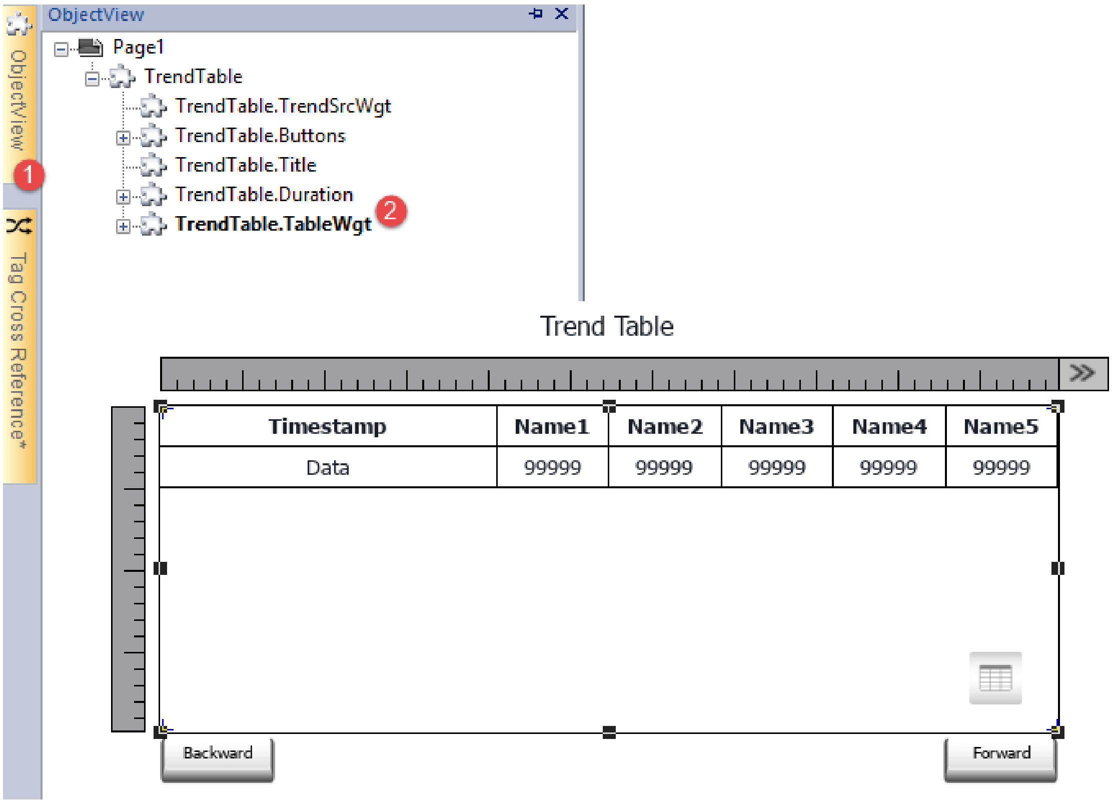


TableGroupWgt : TableWgt	
Current selected row	-1
Table model	
Grid Layout Group	
Num rows	4
Num columns	4
Horizontal overflow	Scroll
Horizontal underflow mode	Center
Scrollbar Handle Color	🟥 [255, 0, 0]
Scrollbar Background Color	□ none
Scrollbar image	
Scrollbar offset	2
Scrollbar size	20
Scrollbar autohide	Auto
Margin collapsed	true
External margin width	0
External margin color	■ [0, 0, 0]
Precached Pages	2
Events	

Precached Pages parameter is available only in Advanced Proprieties 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.



Timestamp	Name1	Name2	Name3	Name4	Name5
Data	99999	99999	99999	99999	99999

Printing table

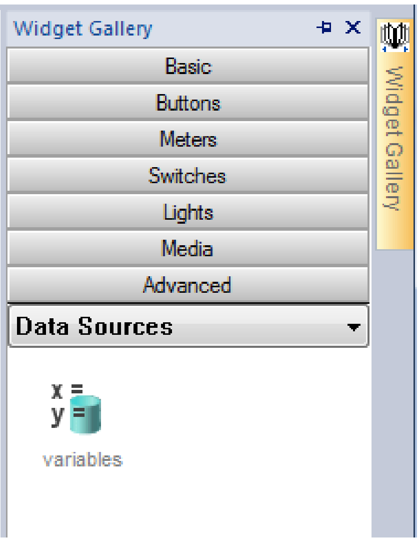
A table widget can be found and used from the print report gallery.

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.

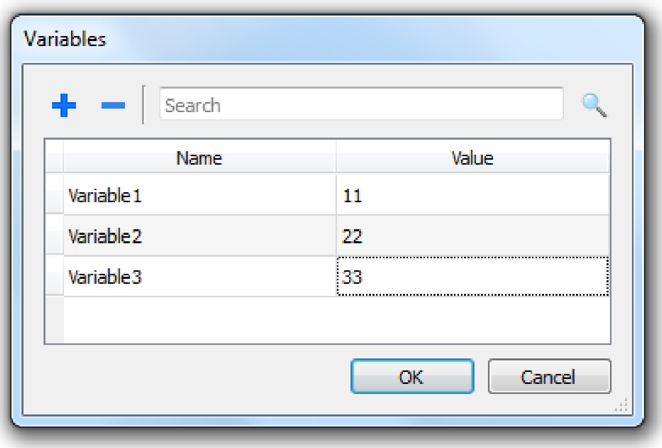
Note: The variables are local to the page where the widget has been inserted.



When you drag and drop this widget into you page, a place holder will be displayed to indicate the widget location, but it will not be visible at runtime.

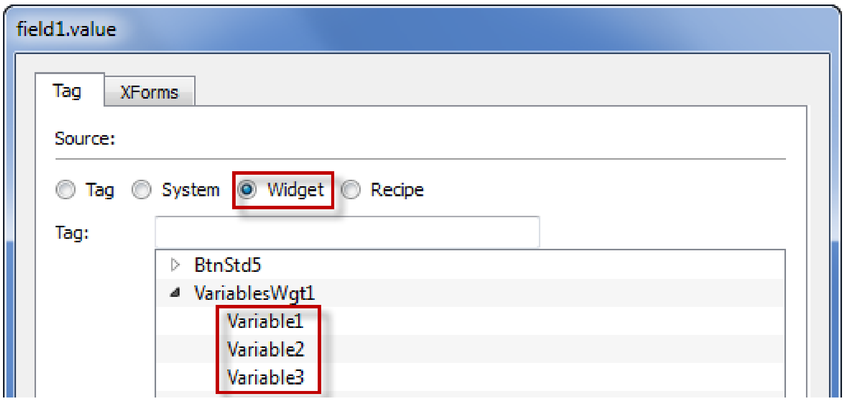
Setting the widget

To create variables and assign values to them, open the **Variables** dialog from the **Variables** property in the **Properties** pane.



Name	Value
Variable1	11
Variable2	22
Variable3	33

These variables can then be referenced from the **Attach tag** dialog, from the Page Editor.



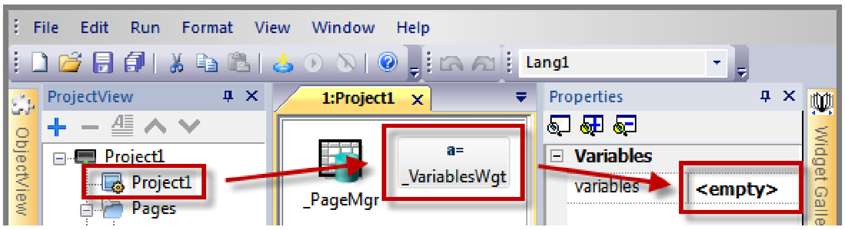
Source:

○ Tag ○ System ● Widget ○ Recipe

Tag:

- ▷ BtnStd5
- ◢ VariablesWgt1
  - Variable1
  - Variable2
  - Variable3

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");
```

38 Custom widgets

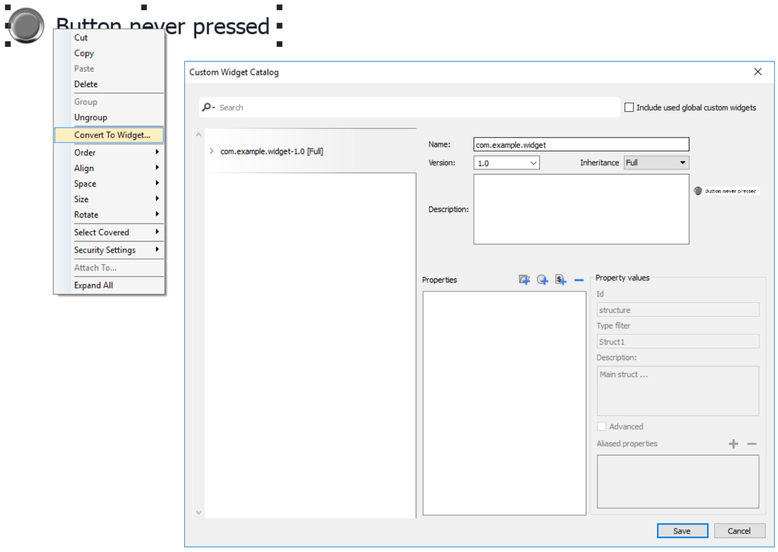
AGI Creator has a large widget library which includes predefined dynamic widgets (buttons, lights, gauges, switches, trends, recipes, and dialog items), as well as static images (shapes, pipes, tanks, motors).

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 loose 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.

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.



Custom Widget Catalog

□ Include used global custom widgets

com.example.widget-1.0 [Full]

Name: com.example.widget

Version: 1.0

Inheritance Full

Description:

Properties

Property values

Id: structure

Type filter: Struct1

Description: Main struct ...

□ Advanced

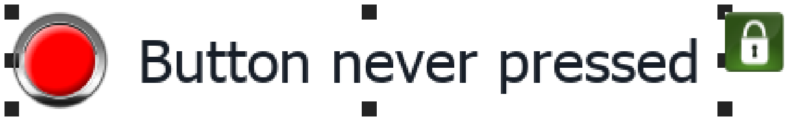
Aliased properties

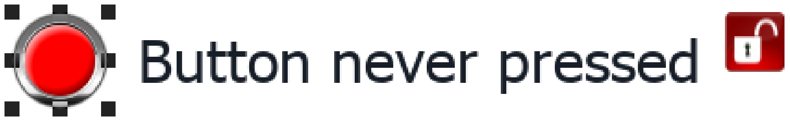
Parameter	Description
Include used custom widgets	When checked, list all the widgets used inside the project. Even system widgets.
Name	You can define everything you prefer, but is common keep a name structure. The folder com.hmi is reserved for the system widgets
Description	Widget description.
Version	Widget version. All widgets that share the same version share the properties defined from the Inheritance parameter.
Inheritance	Properties shared between widgets with the same version • Full (both Graphic and Logic) • Only Graphic • Only Logic • Disable

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.



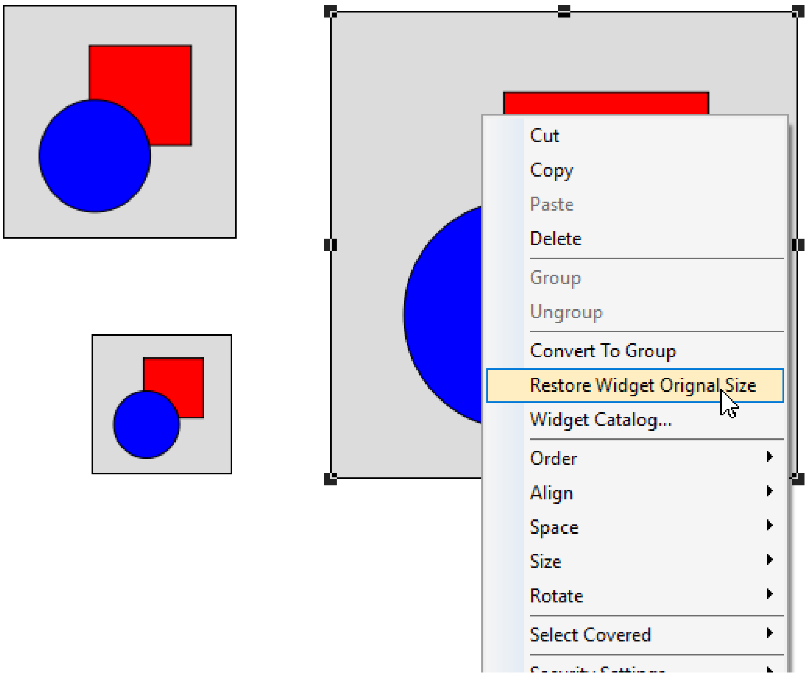


Padlock is showed only when the Inheritance is enabled.

Resize a custom widget

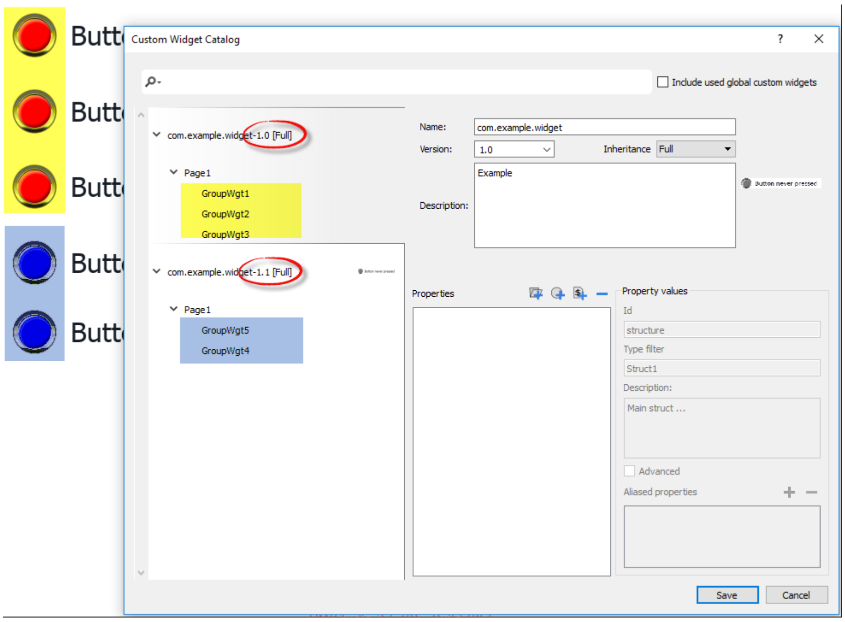
When sizes of custom widget is changed, the new sizes will not be propagated to the other widget instances.

“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.



Name: com.example.widget
Version: 1.0
Inheritance: Full
Description: Example

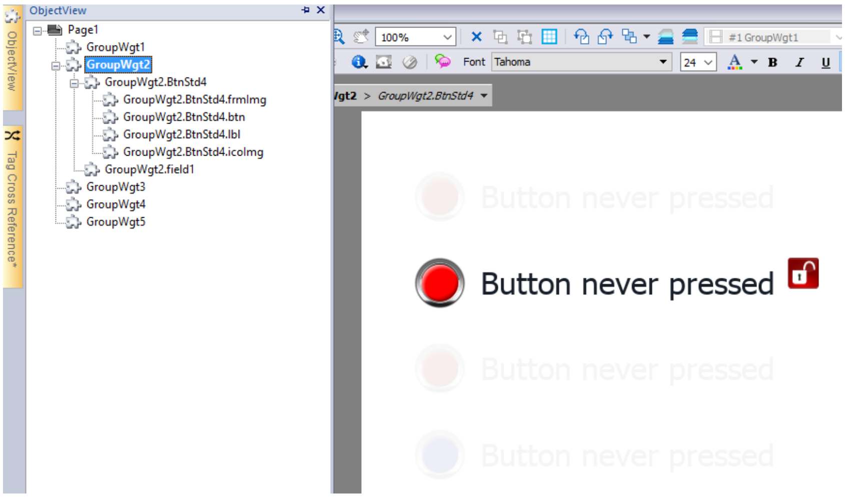
Id: structure
Type filter: Struct1
Description: Main struct ...

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.

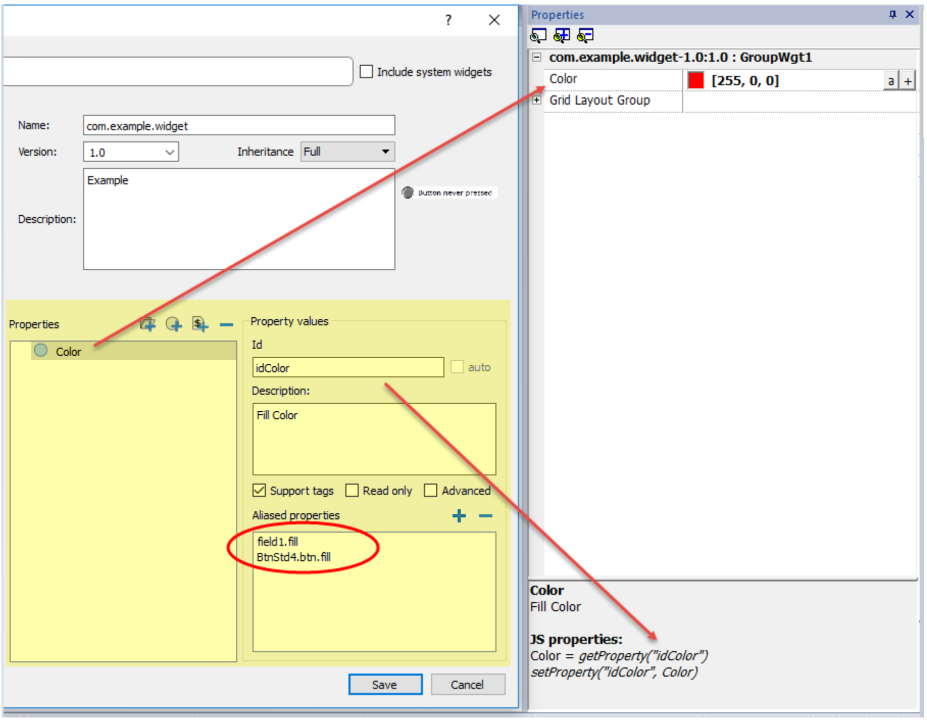
To select a single widget, select it directly form the **ObjectView** pane.



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.



☐ Include system widgets

Name: com.example.widget

Version: 1.0

Inheritance: Full

Description: Example

Properties: Color

Property values

Id: idColor ☐ auto

Description: Fill Color

☑ Support tags ☐ Read only ☐ Advanced

Aliased properties:
field1.fill
BtnStd4.btn.fill

com.example.widget-1.0:1.0 : GroupWgt1

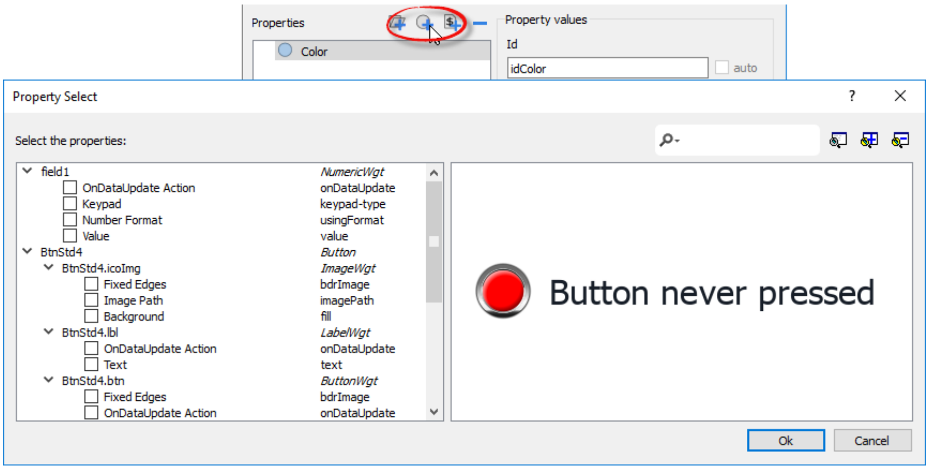
Color: [255, 0, 0]

Grid Layout Group

Color
Fill Color

JS properties:
Color = *getProperty("idColor")*
setProperty("idColor", Color)

2. Click **+** to open the **Property Select** dialog: this lists all the properties of all the grouped widgets.



Property Select

Select the properties:

```
field1                              NumericWgt
    ☐ OnDataUpdate Action           onDataUpdate
    ☐ Keypad                        keypad-type
    ☐ Number Format                 usingFormat
    ☐ Value                         value
BtnStd4                             Button
  BtnStd4.icoImg                    ImageWgt
      ☐ Fixed Edges                 bdrImage
      ☐ Image Path                  imagePath
      ☐ Background                  fill
  BtnStd4.lbl                       LabelWgt
      ☐ OnDataUpdate Action         onDataUpdate
      ☐ Text                        text
  BtnStd4.btn                       ButtonWgt
      ☐ Fixed Edges                 bdrImage
      ☐ OnDataUpdate Action         onDataUpdate
```

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
Properties	Name shown in the **Properties** pane.
Description	Any comment on the property to be displayed in the **Properties** pane.

Parameter	Description
Id	The name exposed by AGI Creator, to JavaScript functions and Attach Tag dialog.
Support Tags	Specifies if the property supports the “Attach to" attribute.
Read only	Property exposed only in read mode
Advanced	Specifies whether each property should appear in the advanced, or in the simple view mode of the **Properties** pane.
Aliased properties	Internal properties linked with the exposed property

Combining properties

To combine two or more properties:

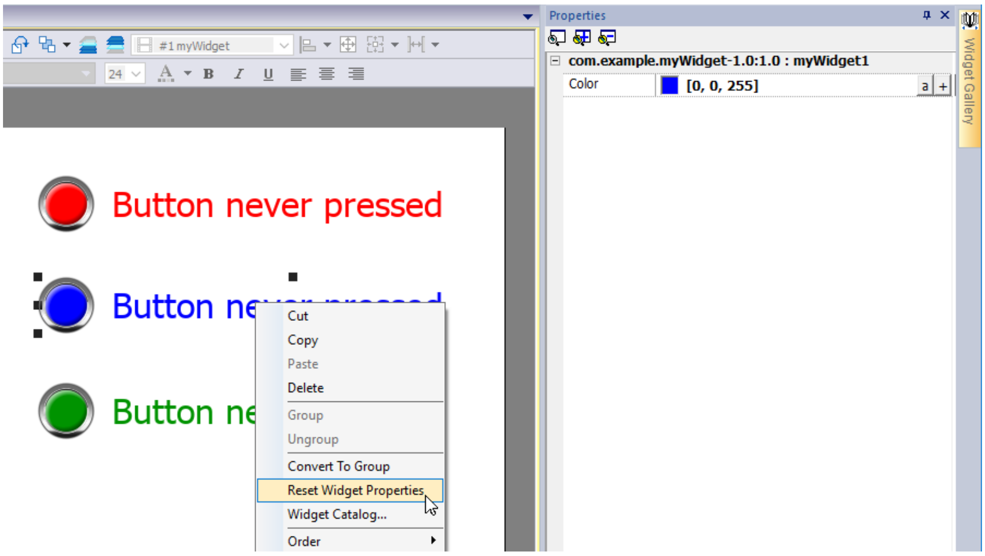
1. Select the primary property in the **Properties** list dialog.
2. Click **+** in the **Aliased properties** toolbar: the **Property Select** dialog is displayed.
3. Select the properties you want to combine.
4. Click **OK:** the combined attributes will be shown in the **Aliased properties** list 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.



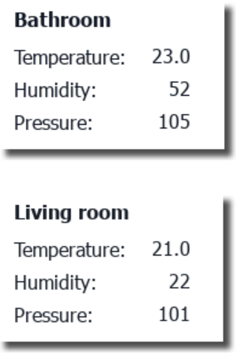
com.example.myWidget-1.0:1.0 : myWidget1

Color [0, 0, 255]

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.

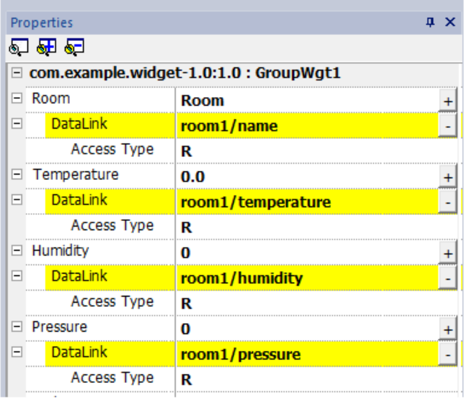


Bathroom

Temperature:	23.0
Humidity:	52
Pressure:	105

Living room

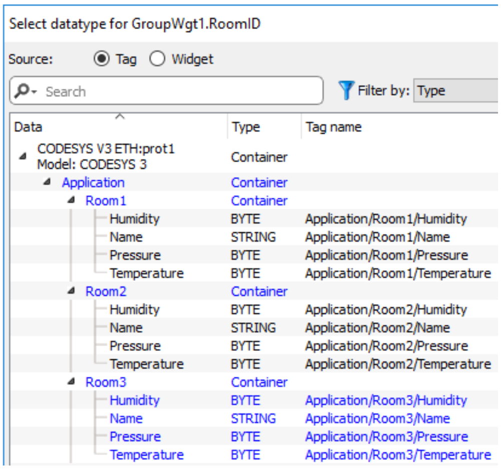
Temperature:	21.0
Humidity:	22
Pressure:	101



Properties

com.example.widget-1.0:1.0 : GroupWgt1

Room	Room
DataLink	room1/name
Access Type	R
Temperature	0.0
DataLink	room1/temperature
Access Type	R
Humidity	0
DataLink	room1/humidity
Access Type	R
Pressure	0
DataLink	room1/pressure
Access Type	R



Select datatype for GroupWgt1.RoomID

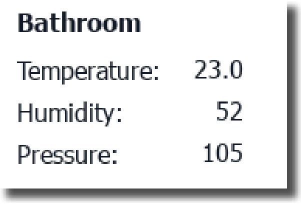
Source: ● Tag ○ Widget

Search

Filter by: Type

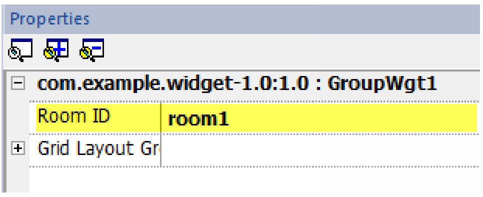
Data	Type	Tag name
CODESYS V3 ETH:prot1 Model: CODESYS 3	Container	
Application	Container	
Room1	Container	
Humidity	BYTE	Application/Room1/Humidity
Name	STRING	Application/Room1/Name
Pressure	BYTE	Application/Room1/Pressure
Temperature	BYTE	Application/Room1/Temperature
Room2	Container	
Humidity	BYTE	Application/Room2/Humidity
Name	STRING	Application/Room2/Name
Pressure	BYTE	Application/Room2/Pressure
Temperature	BYTE	Application/Room2/Temperature
Room3	Container	
Humidity	BYTE	Application/Room3/Humidity
Name	STRING	Application/Room3/Name
Pressure	BYTE	Application/Room3/Pressure
Temperature	BYTE	Application/Room3/Temperature

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

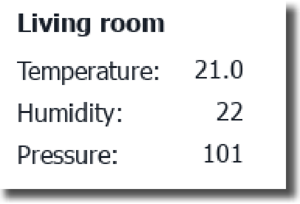
Temperature:	23.0
Humidity:	52
Pressure:	105



Properties

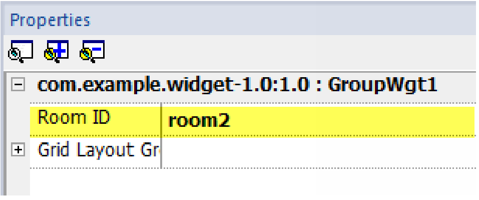
com.example.widget-1.0:1.0 : GroupWgt1

Room ID	room1
Grid Layout Gr	



Living room

Temperature:	21.0
Humidity:	22
Pressure:	101

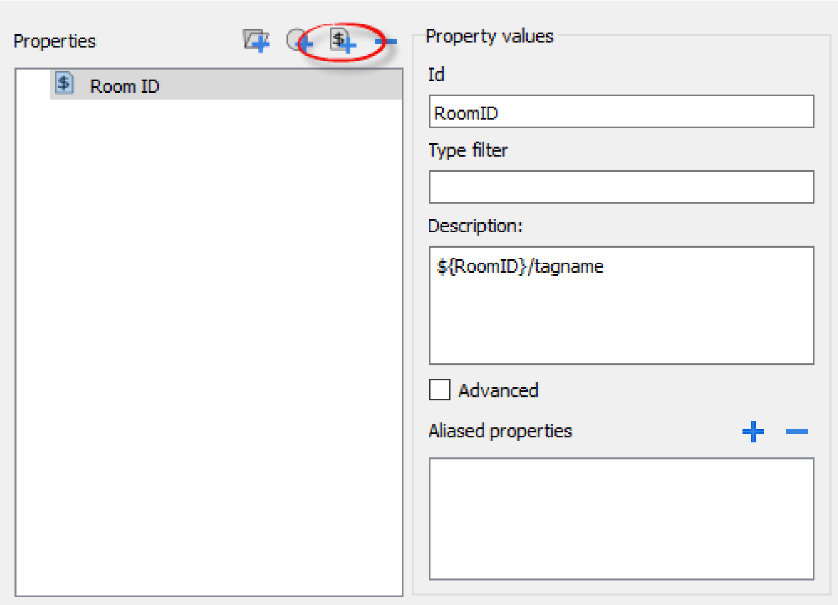


Properties

com.example.widget-1.0:1.0 : GroupWgt1

Room ID	room2
Grid Layout Gr	

A “Parameter” field can be added inside the custom widget using the *"Add Parameter"* icon:



Properties

Room ID

Property values

Id

RoomID

Type filter

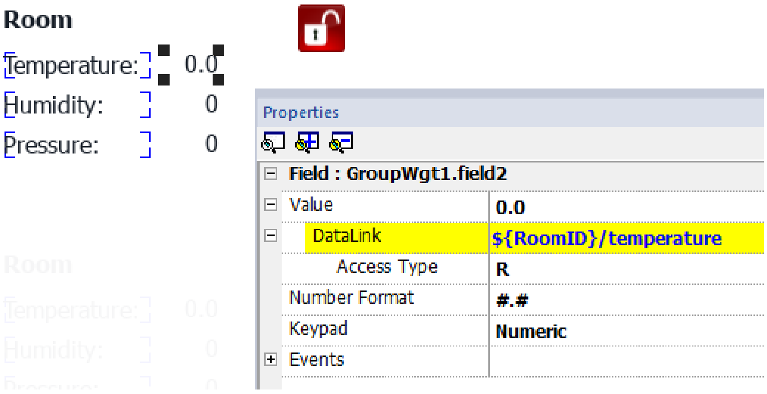
Description:

${RoomID}/tagname

□ Advanced

Aliased properties

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

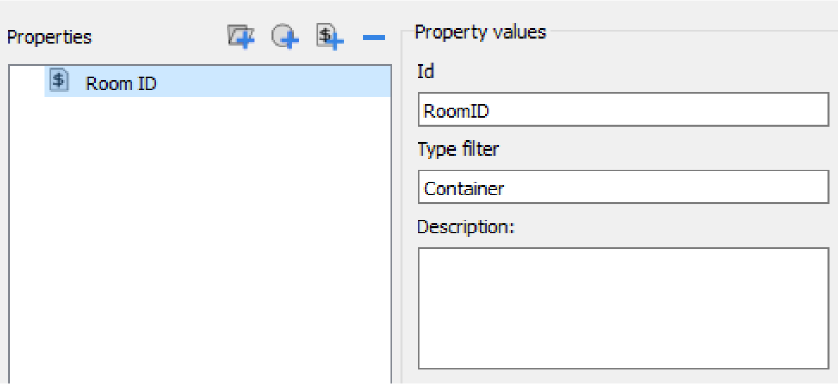
Pressure: 0

Properties

Field : GroupWgt1.field2	
Value	0.0
DataLink	${RoomID}/temperature
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.



Properties

Room ID

Property values

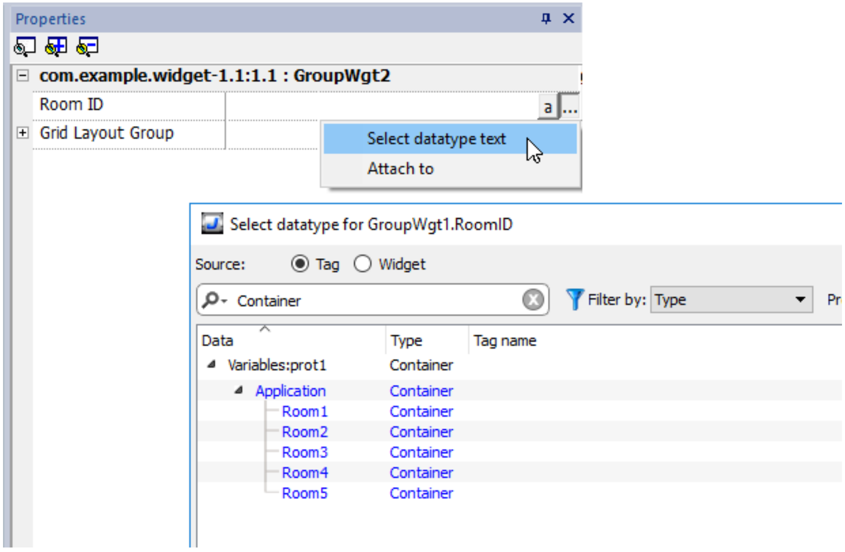
Id

RoomID

Type filter

Container

Description:



Source: ● Tag ○ Widget

Container

Filter by: Type

Data	Type	Tag name
Variables:prot1	Container	
Application	Container	
Room1	Container	
Room2	Container	
Room3	Container	
Room4	Container	
Room5	Container	

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:

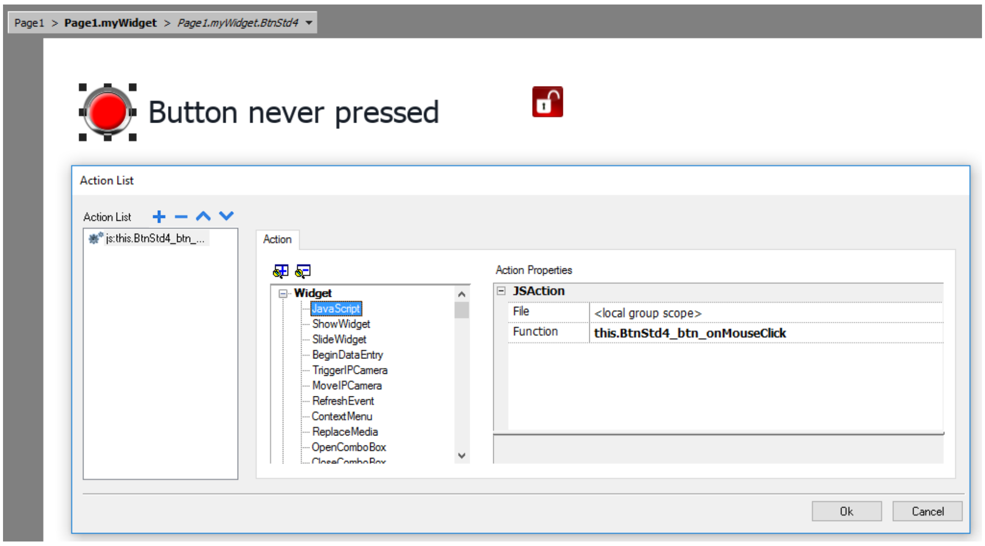
```
var myWidget = page.getWidget("myWidget");
function BtnStd3_btn_onMouseClick(me, eventInfo)
{
     alert("Room is: " + myWidget.getParameter("RoomID"));
}
```

You can also use getProperty(), but getParameter() is more efficient to read custom widget parameters

JavaScript in custom widgets

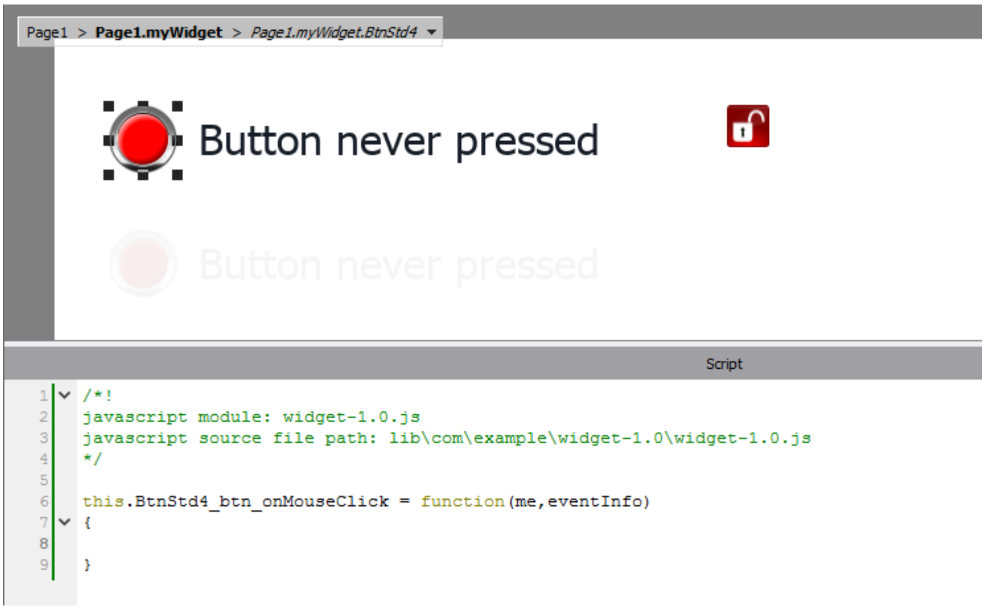
JavaScript functions can be embedded in custom widgets.

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.



JSAction

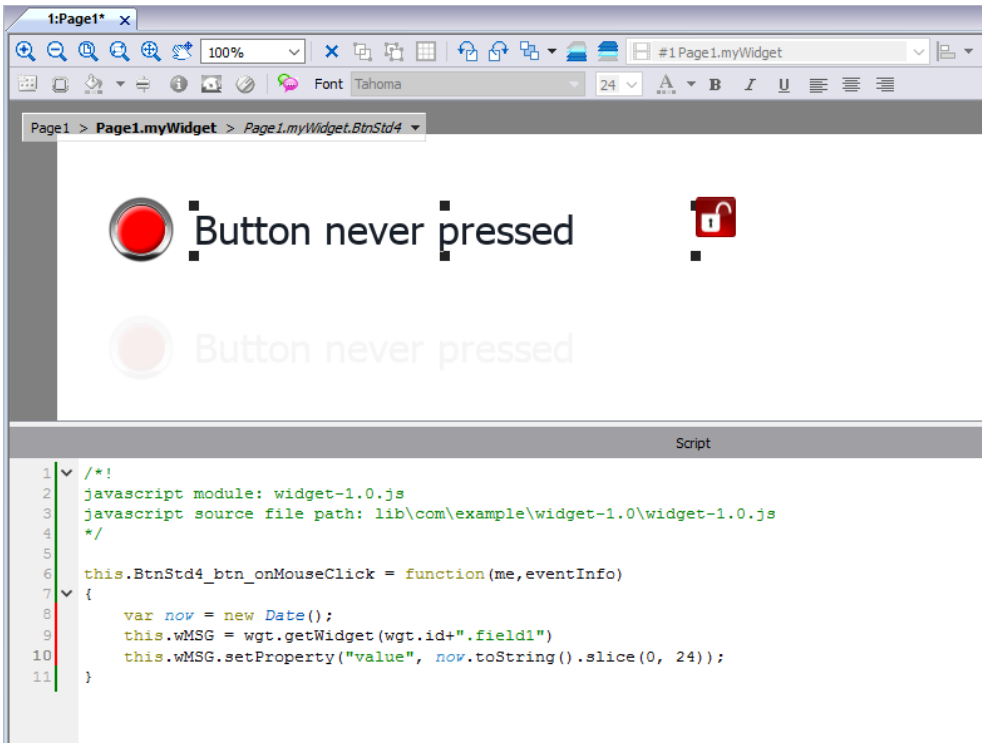
File	<local group scope>
Function	this.BtnStd4_btn_onMouseClick



```
/*!
javascript module: widget-1.0.js
javascript source file path: lib\com\example\widget-1.0\widget-1.0.js
*/

this.BtnStd4_btn_onMouseClick = function(me,eventInfo)
{

}
```

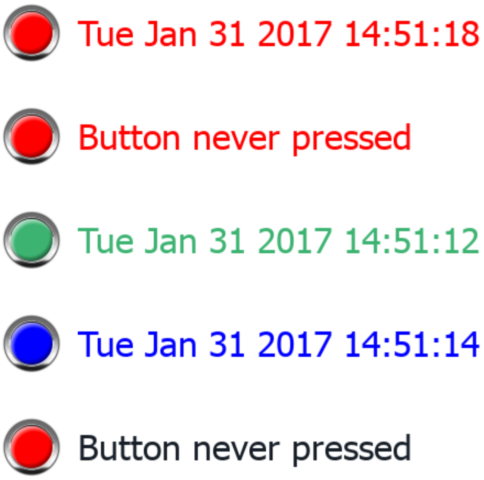
Note the usage of the operator **this.** that is necessary to allow the multiple instance of the custom widget.

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.



```
/*!
javascript module: widget-1.0.js
javascript source file path: lib\com\example\widget-1.0\widget-1.0.js
*/

this.BtnStd4_btn_onMouseClick = function(me,eventInfo)
{
    var now = new Date();
    this.wMSG = wgt.getWidget(wgt.id+".field1")
    this.wMSG.setProperty("value", now.toString().slice(0, 24));
}
```

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

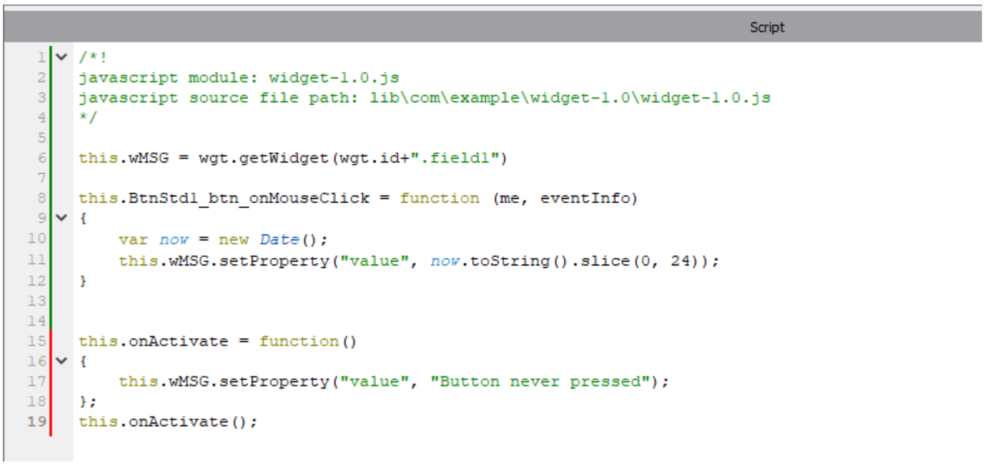
Tue Jan 31 2017 14:51:14

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



```
/*!
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_onMouseClick = 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();
```

Note that the custom widget can also past inside the User’s Gallery for later reuse.

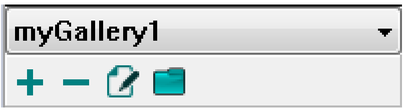
The JavaScript code used inside the examples of this chapter

```
/*!
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_onMouseClick = 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.

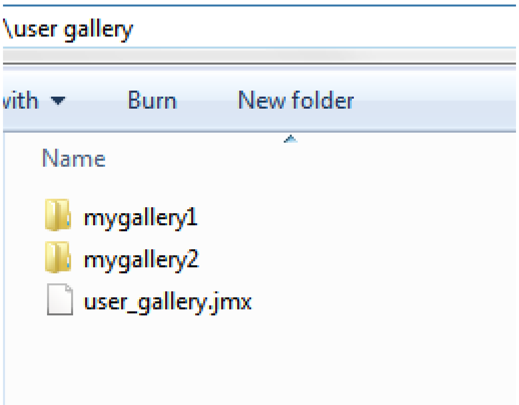
User widgets toolbar



Command	Description
✎	Open the selected widgets folder into the AGI Creator editor
+	Add a new widgets folder
−	Delete current selected folder
📁	Select the user widgets folder

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.

Tip: To import a user gallery sub folder, simply copy the folder to import inside the main user gallery folder.

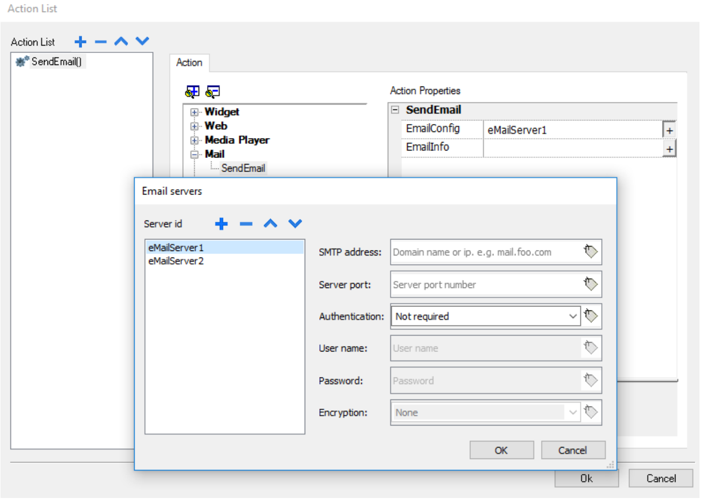


mygallery1
mygallery2
user_gallery.jmx

39 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.



Action List

Action List: SendEmail()

Action Properties

SendEmail

EmailConfig	eMailServer1
EmailInfo	

Email servers

Server id: eMailServer1, eMailServer2

SMTP address:	Domain name or ip. e.g. mail.foo.com
Server port:	Server port number
Authentication:	Not required
User name:	User name
Password:	Password
Encryption:	None

Configuring the email server

To configure the email server, enter the following information for the **EmailConfig** setting:

Parameter	Description
SMTP Address	SMTP server address.
Server Port	Port for SMTP server connection (default = 25).
Require Auth	Select if the SMTP server requires authentication.
User Name	Username for sending mail using SMTP server.
Password	Password for sending mails using SMTP server.
Encryption	Encryption type (none or SSL).

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:

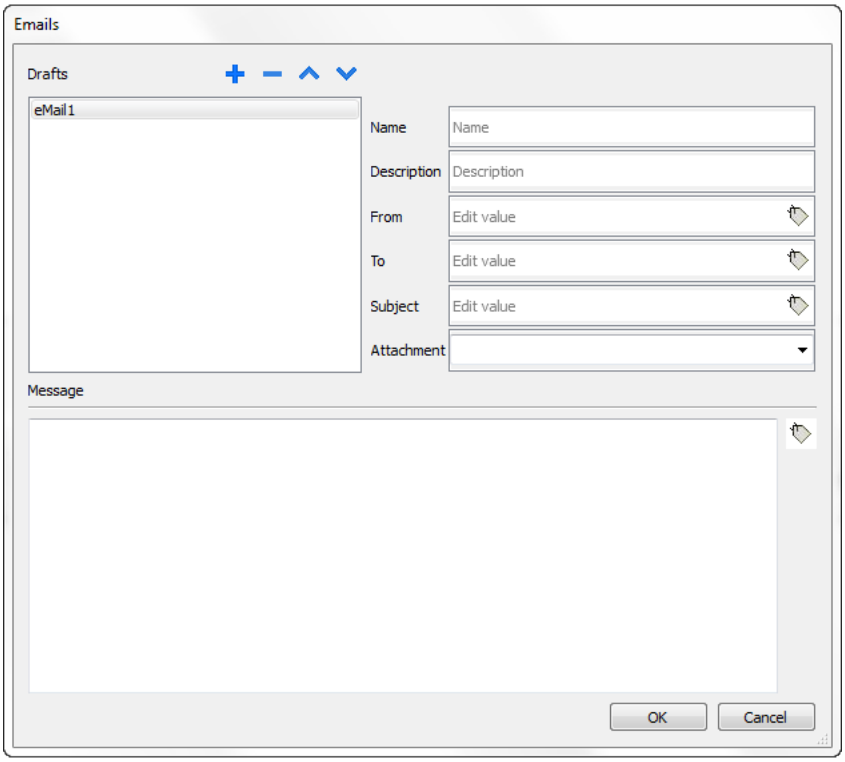
Parameter	Description
Name	Optional, this information is only for the log.
Description	Optional, this information is only for the log.
From	Sender email address (for example, John@domain.com).
To	Recipient e-mail addresses. To enter multiple addresses, separate them with a semi-colon.
Subject	Subject of email.
Attachment	Path of the file to be sent as attachment. Only one attachment at a time can be sent. Note: The maximum size of the attachments is usually set by the SMTP server.
Body	Main content of the email. Here you can insert live tags if you include them in square brackets. 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.

Tip: Attach a string tag to the **From**, **To** and **Subject** fields so that their value can be changed in the HMI Runtime.

WARNING: The maximum size for the message body is 4096 bytes, the exceeding text will be truncated.

Adding email templates

Click **+** to add more templates.



Name: Name
Description: Description
From: Edit value
To: Edit value
Subject: Edit value

40 JavaScript

The purpose of this section is to describe how JavaScript is used in the AGI Creator applications, not to explain the JavaScript language.

AGI Creator JavaScript is based on the ECMAScript programming language http://www.ecmascript.org , as defined in standard ECMA-262.

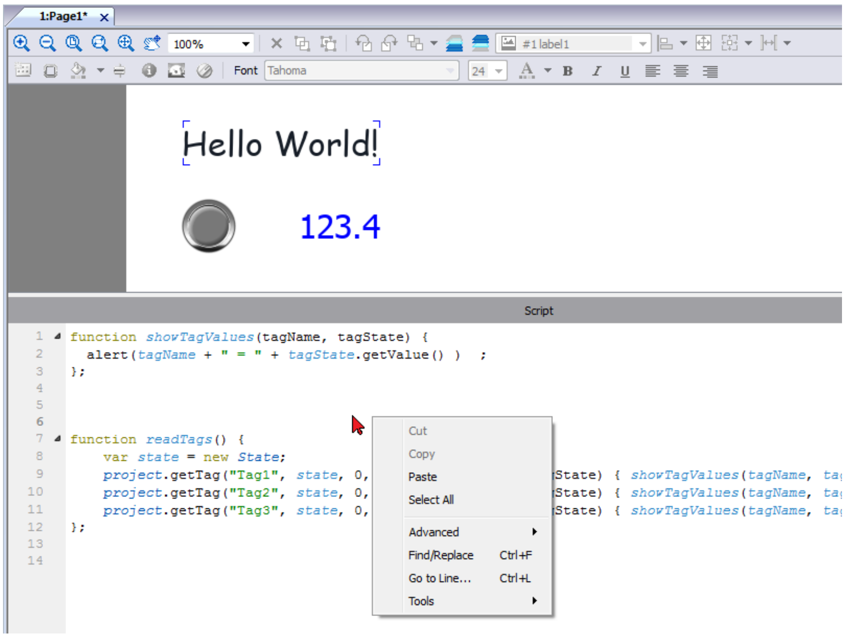
If you are familiar with JavaScript, you can use the same type of commands in AGI Creator as you do in a web browser. If you are not familiar with the ECMAScript language, refer to:

https://developer.mozilla.org/en/JavaScript

JavaScript editor

AGI Creator includes a powerful JavaScript editor.

Right-click in the editor to display available commands.



```
function showTagValues(tagName, tagState) {
  alert(tagName + " = " + tagState.getValue() )  ;
};



function readTags() {
    var state = new State;
    project.getTag("Tag1", state, 0,                     State) { showTagValues(tagName, ta
    project.getTag("Tag2", state, 0,                     State) { showTagValues(tagName, ta
    project.getTag("Tag3", state, 0,                     State) { showTagValues(tagName, ta
};
```

Execution of JavaScript functions

JavaScript functions are executed when events occur. For example, a user can define a script for the OnMouseClick 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.

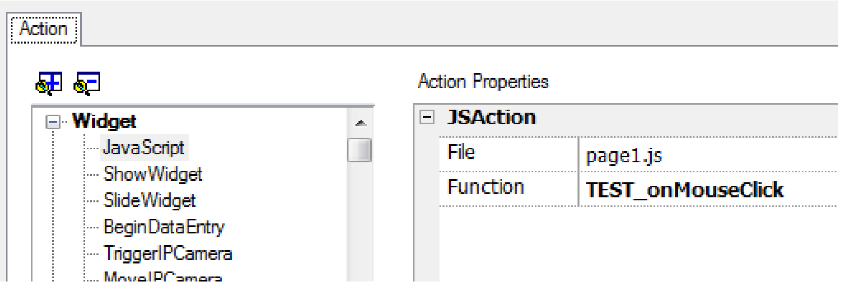
AGI Creator provides a JavaScript engine running on the client side. Each project page can contain scripts having a scope local to the page where they are added; global scripts can be created to be executed by scheduler events or alarm events.

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 AGI 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

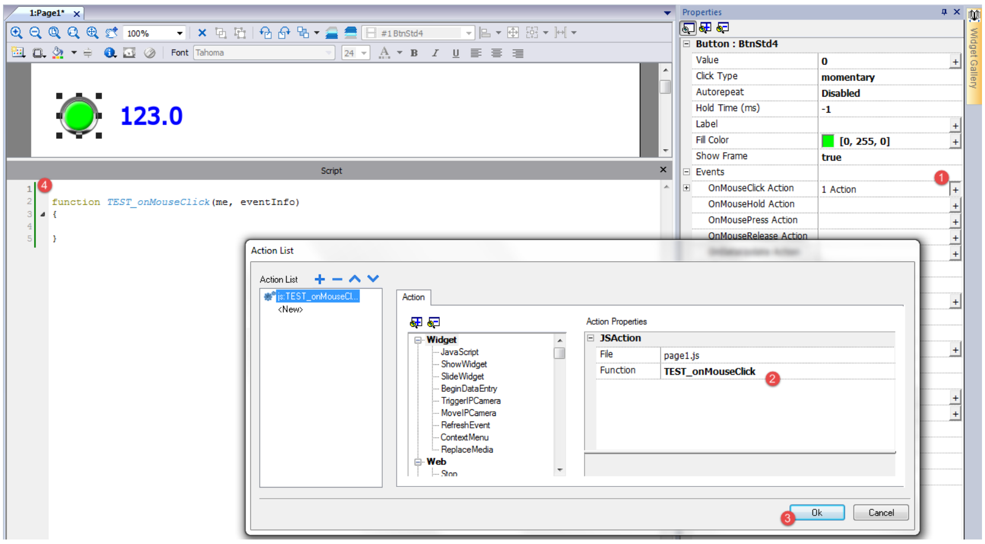
JavaScript editor will open when you add a JavaScript action inside an action list.



Action Properties

JSAction	
File	page1.js
Function	**TEST_onMouseClick**

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.



```
function TEST_onMouseClick(me, eventInfo)
{

}
```

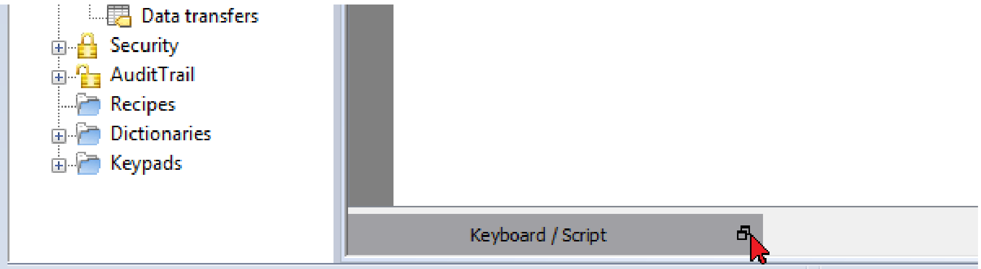
Properties

Button : BtnStd4	
Value	0
Click Type	momentary
Autorepeat	Disabled
Hold Time (ms)	-1
Label	
Fill Color	🟩 [0, 255, 0]
Show Frame	true
Events	
OnMouseClick Action	1 Action
OnMouseHold Action	
OnMousePress Action	
OnMouseRelease Action	

Action Properties

JSAction	
File	page1.js
Function	TEST_onMouseClick

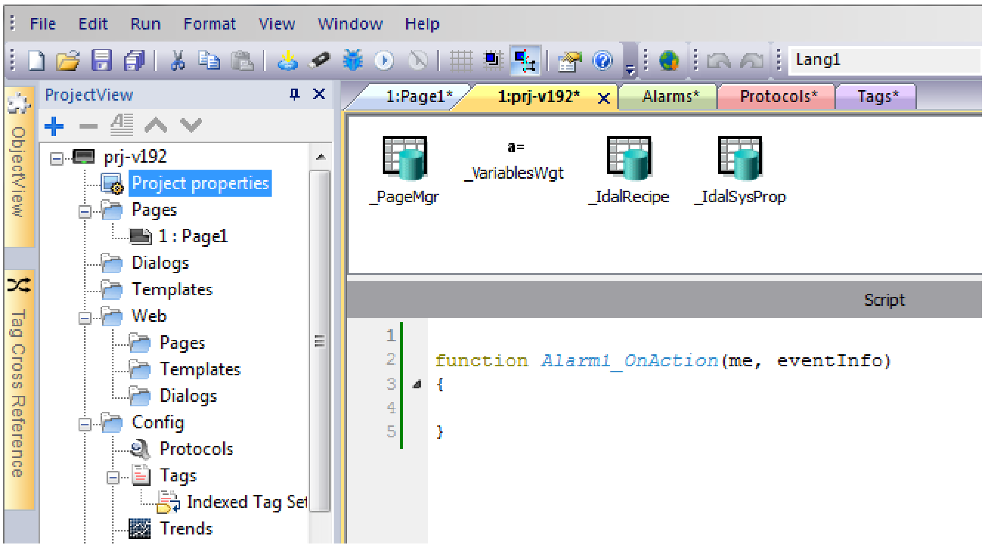
You can also open the JavaScript editor from the **Script** tab at the bottom of the workspace.



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***

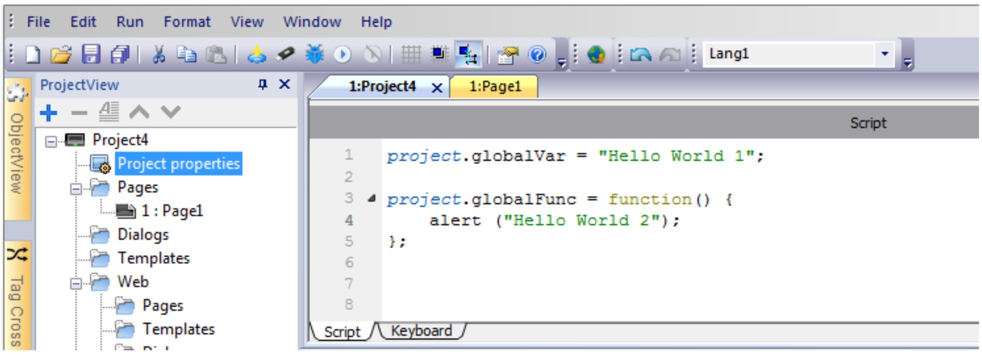


```
function Alarm1_OnAction(me, eventInfo)
{

}
```

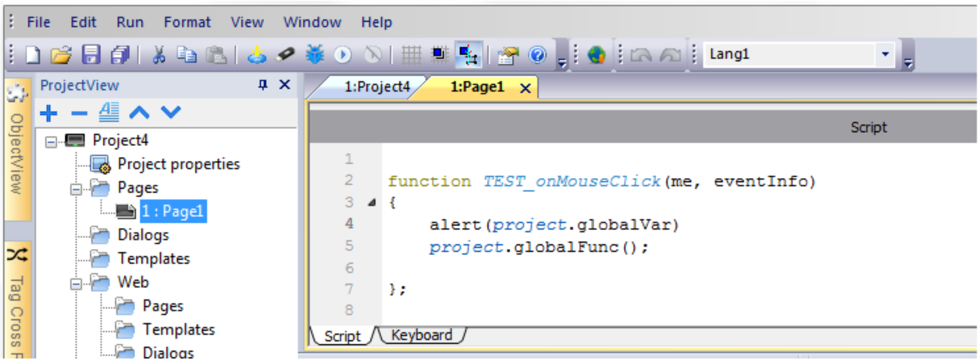
Note: JavaScript actions are client actions so they are executed only when a client is logged in.

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.



```
project.globalVar = "Hello World 1";

project.globalFunc = function() {
    alert ("Hello World 2");
};
```



```
function TEST_onMouseClick(me, eventInfo)
{
    alert(project.globalVar)
    project.globalFunc();

};
```

Events

You can add JavaScript to the following categories of events:

- Widget events
- Page events
- System events

For events of type:

- OnMousePress
- OnMouseRelease
- OnMouseClick
- OnWheel

JavaScript **eventinfo** parameter contains the following additional properties:

Parameter	Description
eventInfo.posX	Local mouse/touch X coordinate with respect to widget coordinates
eventInfo.posY	Local mouse/touch Y coordinate with respect to widget coordinates
eventInfo.pagePosX	Page X mouse/touch coordinate
eventInfo.pagePosY	Page Y mouse/touch coordinate
eventInfo.wheelDelta	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

onMouseClick

```
void onMouseClick (me, eventInfo)
```

This event is available only for buttons and it occurs when the button is pressed and released quickly.

Parameter	Description
me	Object triggering the event
eventInfo	Details of triggered event

```
function buttonStd1_onMouseClick(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
me	Object triggering the event
eventInfo	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
me	Object triggering the event
eventInfo	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
me	Object triggering the event
eventInfo	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
me	Object triggering the event
eventInfo	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 =** W 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
me	Object triggering the event
eventInfo	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:

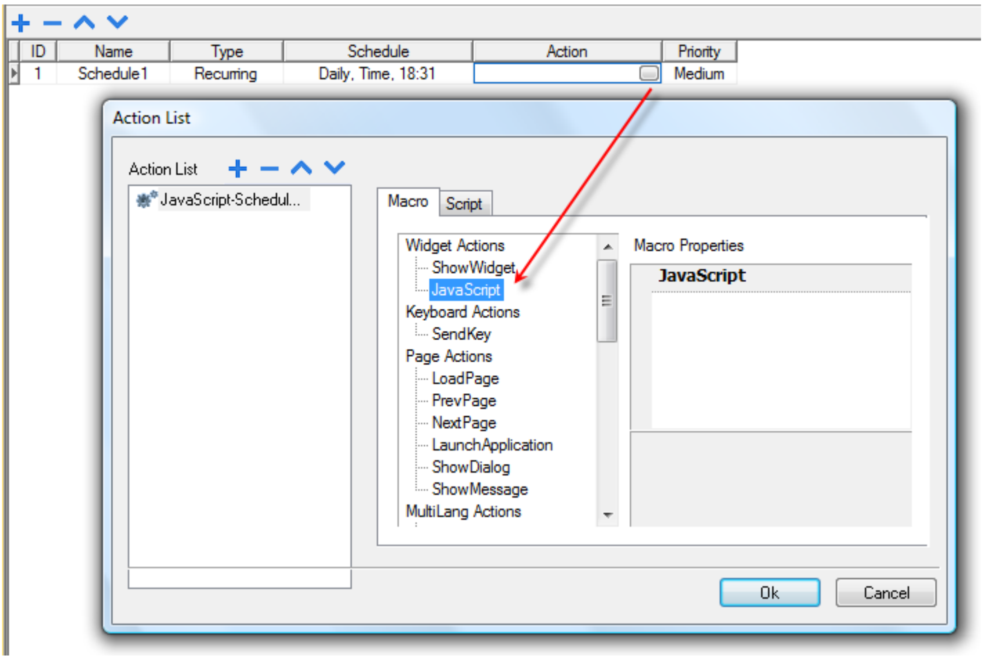
- scheduler
- alarms
- a wheel device

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.

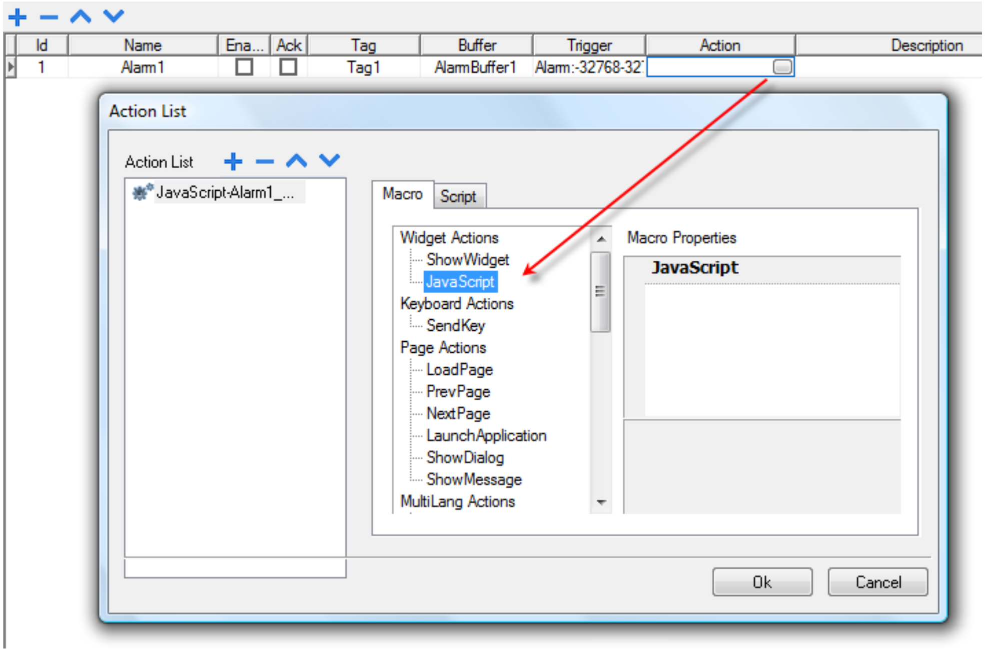


ID	Name	Type	Schedule	Action	Priority
1	Schedule1	Recurring	Daily, Time, 18:31		Medium

You can edit the JavaScript from the **Project Properties** tab.

Alarm events

These events occur when triggered by the associated alarm condition.



Id	Name	Ena...	Ack	Tag	Buffer	Trigger	Action	Description
1	Alarm1	□	□	Tag1	AlarmBuffer1	Alarm:-32768-32		

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
me	Object triggering the event
eventInfo	Details of triggered event

```
function Project1_onMouseWheelClock(me, eventInfo) {
     //do something…
}
```

Objects

AGI Creator 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
propertyName	String containing the name of property to get
index	Index of the element to get from the array (default = 0)

Almost all properties that are shown in the AGI Creator **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
propertyName	String containing the name of property to set
value	String containing the value to set the property.
index	Index of the element to set in the array (default = 0)

Almost all properties that are shown in the AGI Creator **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("multistate1");
     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
wgtName	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
functionName	String containing the name of function to call
delay	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
timerID	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
functionName	String containing the name of function to call
interval	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
groupName	String containing the names of the groups. The and/or expression to retrieve tags list from multiple group is supported. | OR operator & AND operator (...) The brackets can be used to define how evaluate the expression Examples: • project.getGroup("one", group); • project.getGroup("(one|two)", group); • project.getGroup("((one&two)*three)", group);
groupInstance	Group element to be filled
callback	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
tagName	String of tag name
state	State element to be filled
index	Index if the tag is of array type. -1 returns the complete array. Default = 0.
callback	Function name if an asynchronous read is required. Default = “".
forceRefresh	(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
tagName	String of tag name
tagValue	Object containing the value to write
index	Index if the tag is of array type. -1 pass the complete array. Default = 0.
forceWrite	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

Interger 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
recipeName	String representing the recipe name
recipeSet	String representing the recipe set, can be either the recipe set name or 0 based set index.
recipeElement	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
recipeName	String representing the recipe name
recipeSet	String representing the recipe set, can be either the recipe set name or 0 based set index.
recipeElement	String representing the recipe Element, can be either the element name or 0 based element index.
value	An object containing the value to store in the recipe. It can be an array type.

Return value

Interger 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
recipeName	String representing the recipe name
recipeSet	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
recipeName	String representing the recipe name
recipeSet	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
appName	String containing the application name
appPath	String containing the application absolute path
Arguments	String containing the arguments to be sent to application
singleInstance	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 543).

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 AGI 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 AGI 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
reportName	String containing the report name
silentMode	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
text	String to print
silentMode	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
text	Hexadecimal string to print
silentMode	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
message	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
message	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
strPath	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
dirPath	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
dirPath	Folder string

Read file

```
object fs.readFile(strfile [,strFlag]);
```

Opens the strFile file in read mode, reads its contents and returns it.

Parameter	Description
strFile	File name string
strFlag	Read file mode: “b” reads and returns as binary file (otherwise returns a text file)

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
strFile	File name string
fileData	Data to be write on the file in byte array
strFlag	Write file mode: • “a”: appends fileData to the end of the text file • “r”: replaces the contents of the file with fileData • “ab”: appends fileData to the end of the binary file • “rb”: replaces the contents of the binary file with fileData

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
strFile	File name string
fileData	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
strPath	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
strPath	Path string

File status

```
object fs.stat(strPath)
```

Retrieves information on the file/folder present at the specified path.

Parameter	Description
strPath	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:

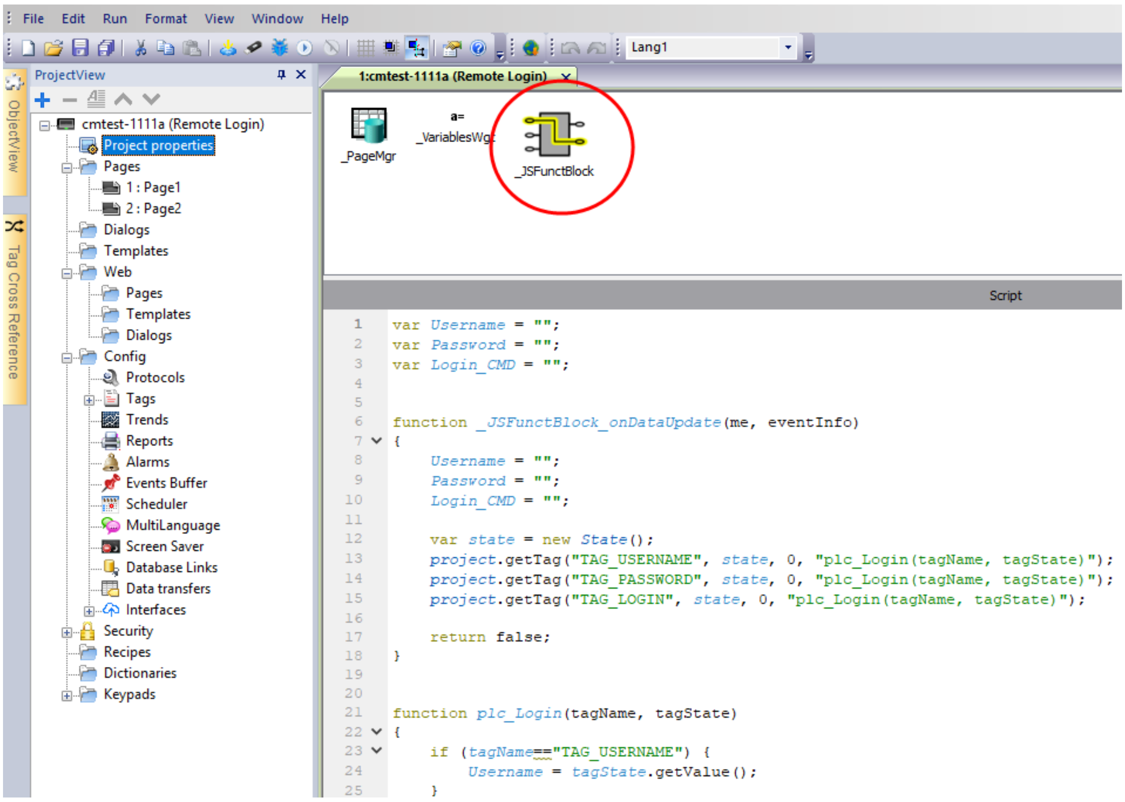
- TAG_USERNAME
- TAG_PASSWORD
- TAG_LOGIN

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 OnDataUpdate Action of the JavaScript function block will execute the required login/logout command.



```
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();
    }
```

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 496

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

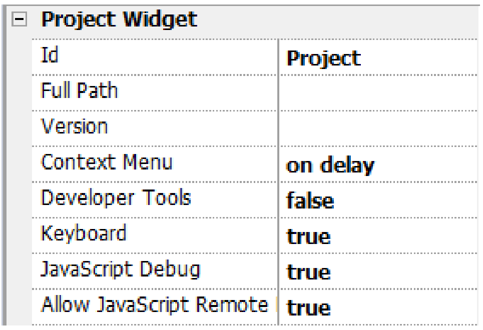
AGI Creator and HMI Runtime include a JavaScript debugger.

Two types of debuggers are available:

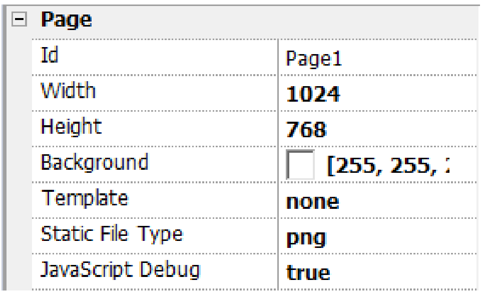
- Runtime debugger: a debugger running directly on the HMI device
- Remote debugger: a debugger running on a remote computer connected to the HMI device via Ethernet (usually computer running AGI Creator)

Enabling debugging

In the **Properties** pane of a page, set **JavaScript Debug** to **true**.



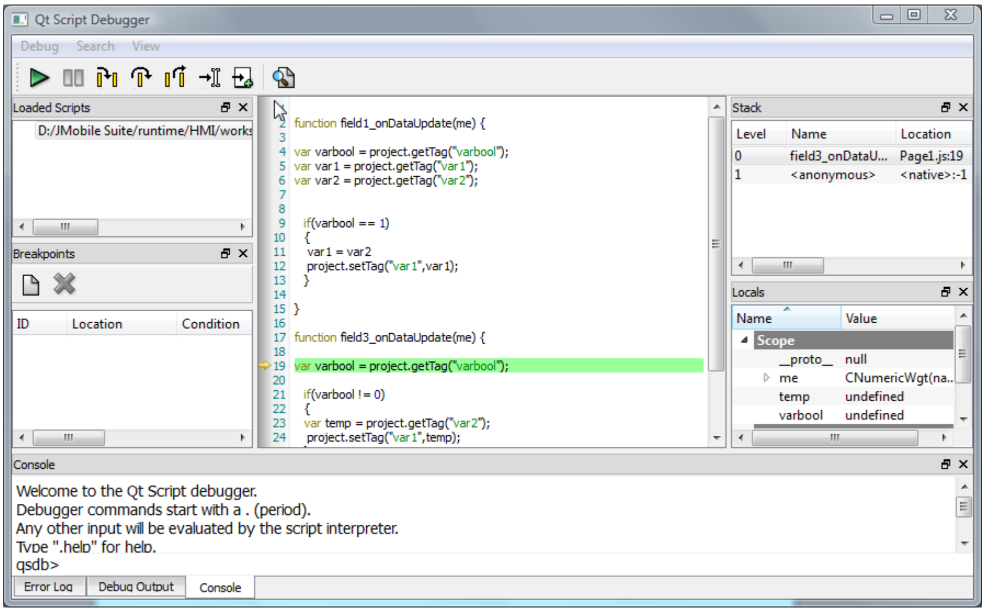
⊟ Project Widget	
Id	Project
Full Path	
Version	
Context Menu	on delay
Developer Tools	false
Keyboard	true
JavaScript Debug	true
Allow JavaScript Remote	true



⊟ Page	
Id	Page1
Width	1024
Height	768
Background	□ [255, 255,
Template	none
Static File Type	png
JavaScript Debug	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.



```
function field1_onDataUpdate(me) {

var varbool = project.getTag("varbool");
var var1 = project.getTag("var1");
var var2 = project.getTag("var2");


  if(varbool == 1)
  {
   var1 = var2
   project.setTag("var1",var1);
  }

}

function field3_onDataUpdate(me) {

var varbool = project.getTag("varbool");

  if(varbool != 0)
  {
  var temp = project.getTag("var2");
   project.setTag("var1",temp);
```

Level	Name	Location
0	field3_onDataU...	Page1.js:19
1	<anonymous>	<native>:-1

Name	Value
Scope	
__proto__	null
me	CNumericWgt(na..
temp	undefined
varbool	undefined

```
Welcome to the Qt Script debugger.
Debugger commands start with a . (period).
Any other input will be evaluated by the script interpreter.
Type ".help" for help.
qsdb>
```

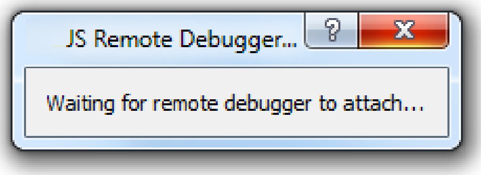
For a complete reference guide about JavaScript Debugger refer to :

http://qt-project.org/doc/qt-4.8/qtscriptdebugger-manual.html

Remote JavaScript Debugger

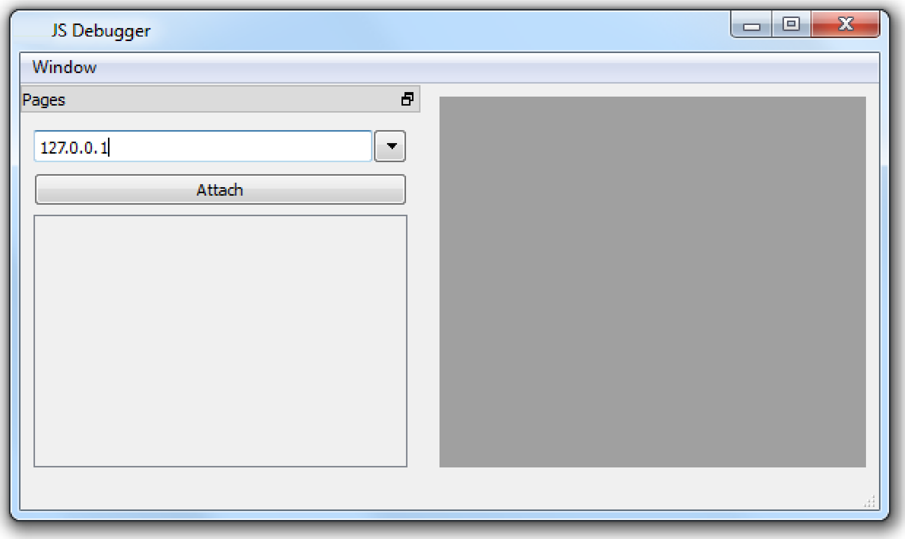
*Path: **Run**> **Start JS Remote Debugger***

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.



Waiting for remote debugger to attach...

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 AGI Client.

JavaScript Memory Usage

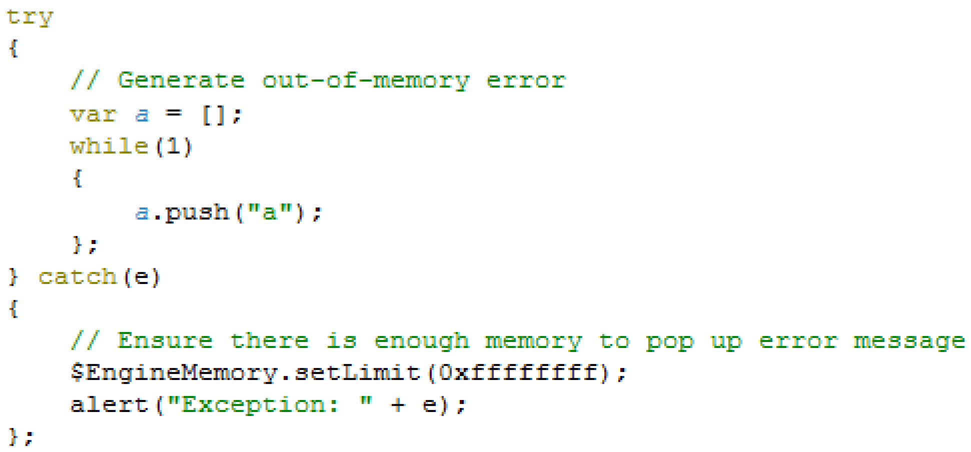
When the memory exceeds the maximum, an out of memory exception is thrown with a custom message. Please note that we do not have a fine control over the actual memory usage so it is mainly a soft limit. Moreover we can't forbid the allocation (this will break the engine implementation), so exception 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.

JavaScrip 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 snipped. 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);
};
```

41 Handling Gestures

Some widgets have the capability to detect and manage pan and pinch gestures.

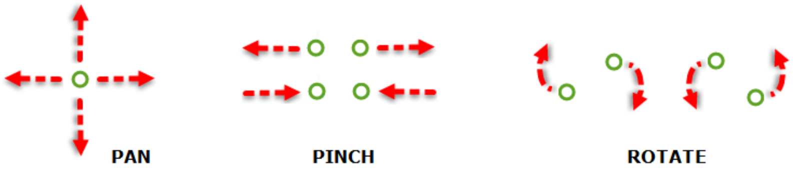
- Trends (see "Trend widget gestures" on page 282 for details)
- Alarms Widget
- Combo box Widget
- Table Widget
- PDF Viewer
- Gesture Area Widget. Special widget designed to customize handling of gesture events (see "Gesture area widget" on page 414 for details)

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. Them are available only with HMI devices supporting multi touch operation (see "HMI devices capabilities" on page 543)

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.



42 Web access

AGI Web 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 AGI Web, users can create a web project to display at a remote location the same graphical display shown on the HMI device. AGI Web 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 AGI Web pages on the HMI device.

Supported platforms and browsers

AGI Web 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

AGI Web works with all modern web browsers. The following browsers have been tested for compatibility with AGI Web:

- Mozilla Firefox 52+
- Microsoft Edge 42+
- Apple Safari 11+
- Google Chrome 57+

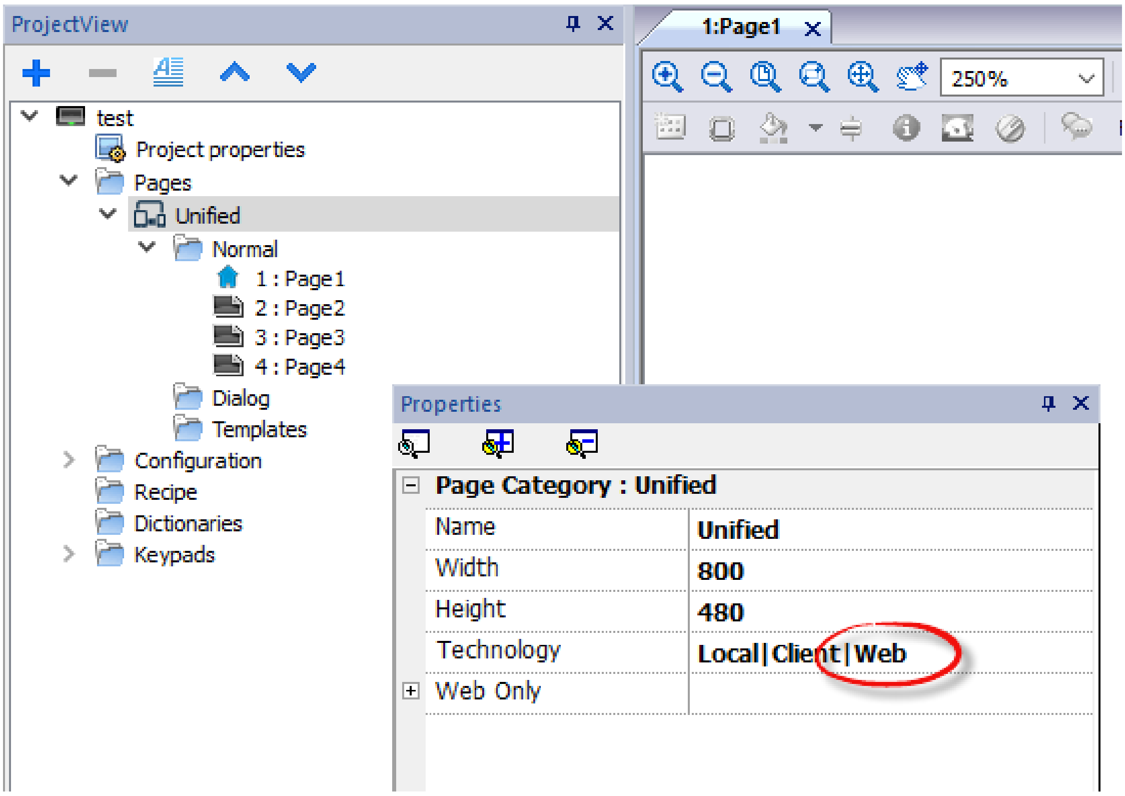
Working with tablets or phones

AGI Web works with most tablet and phone devices. The following tablets have been tested for compatibility with AGI Web:

- iOS 10+ - Mobile Safari
- Android 7+ - Chrome for Android 55+

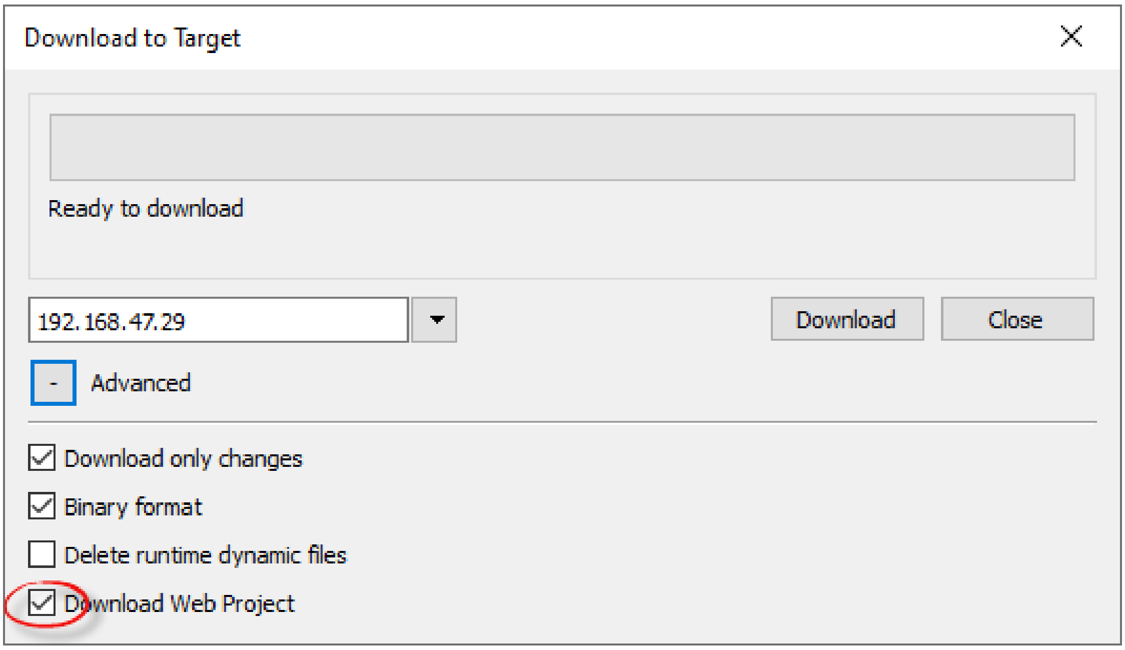
Web pages

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).



Page Category : Unified

Name	Unified
Width	800
Height	480
Technology	Local|Client|Web
Web Only	


Download to Target

Ready to download

192.168.47.29

☑ Download only changes

☑ Binary format

☐ Delete runtime dynamic files

☑ Download Web Project

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 72 chapter.

Web page properties

Any widgets and features can be used in AGI Creator; however, not all features are currently available in AGI Web. If the project includes a feature that is not available, AGI Web will still work correctly but the feature will not be available on the remote client device. See "Web supported features" on page 523 for a list of the features supported in AGI Web 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
Fit to Screen Size	How the page will adapted to the browser's viewport • None • Fit to Screen Simple modify the zoom level to adapted the page to viewport of the browser • Responsive Design Smart modify the zoom level to adapted the page to viewport of the browser respecting the restrictions defined inside the grid layout

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 AGI Web project using the online simulator opening a standalone web page directly from a browser.

Testing with the online simulator

AGI Creator includes an web server in the online simulator. You can start the simulator and access your AGI Web project from a web browser. The pages will be served from the simulator.

1. Create your project (see "Web pages" on page 518).
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 518 for a list of browser compatible with AGI Web).
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 AGI Web.
5. Test your project in the browser.

Important: If you make any changes to the project pages in AGI Creator 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

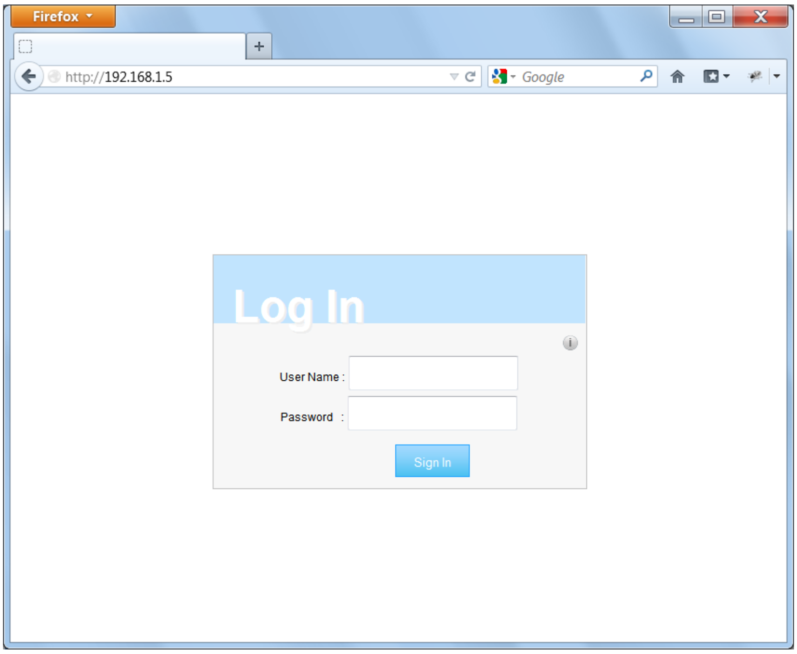
After testing the AGI Web pages, you can download the project to the desired HMI device.

The AGI Web project is downloaded together with the AGI Creator project, see "Download to HMI device" on page 98 for details.

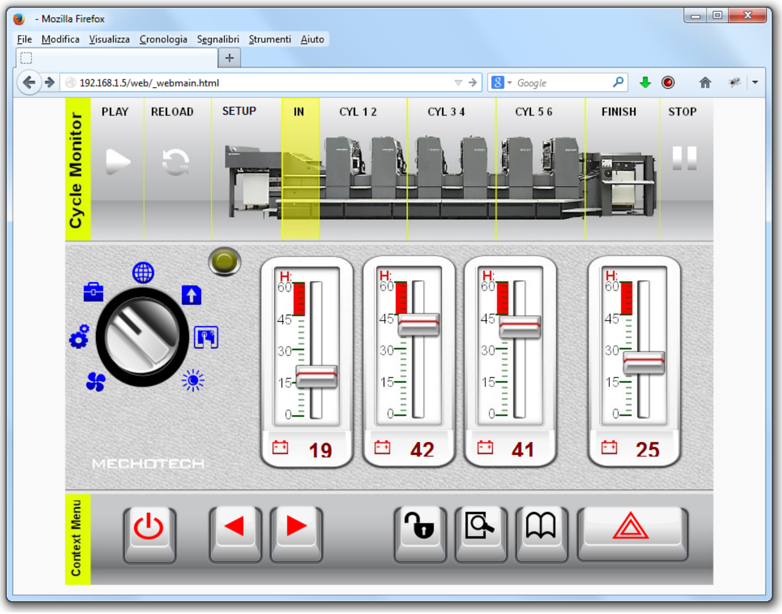
After the download process is completed, the HMI project automatically starts on the HMI device and the AGI Web project is ready to be used.

Running AGI Web from a browser

1. Open a web browser and enter the IP adress 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 333 for details on how to create credentials.

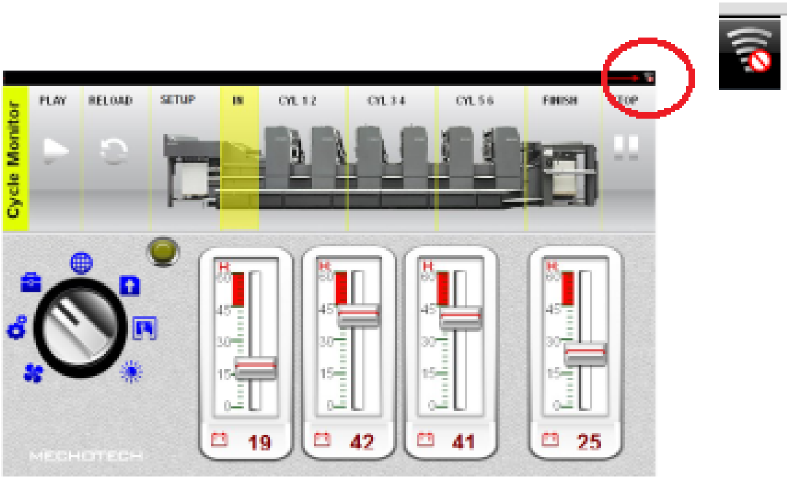
You can interact with the project using the browser in the same way you interact with a device when touching the screen: click buttons to change pages, view indicators and gauges, drag slider handles to change values, and so on. The AGI Web project will manage all communications with the web server while 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 AGI Web 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 AGI Web 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 142 for additional details.

Note: If you make changes in the AGI Web 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

AGI Web 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 AGI Web Project

The AGI Web 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 AGI Web 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 AGI Web cannot perform any server-bound communication.

Web supported features

Some features or widget's properties are not supported by AGI Web. 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
- Button gstArea
- 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 431 for additional details)
- Media Player
- Multistate image multilayer (Multistate image widget can be used)
- Rotation menu widget
- RSS Feed
- RSS Scroll
- Scheduler
- Text Editor
- Web Browser
- Hyper Link

List of actions that are not supported

Widget	SlideWidget, BeginDataEntry, TriggerIPCamera, MoveIPCamera, RefreshEvent, ContextMenu, ReplaceMedia, OpenComboBox, CloseComboBox, SelectAlarmsOnSrc, ShiftTableDataSrcColumns, ResetTableDataSrcColumns, SetTableSortingColumn
Web Browser	All actions are not supported
Text Editor	All actions are not supported
MediaPlayer	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, LaunchHMICloudEnabler, LockScreen, LoadProject, LastVisitedProject
Print	All actions are not supported
Tag	DataTransfer, ActivateGroup, DeactivateGroup, EnableNode, BACnetClearPriority, BACnetClearAllPriorities, BACnetSetPriority, ClearRetentiveMemory, ForceReadTag
Trend/Graph	ConsumptionMeterPageScroll

System	Restart, ResetProtoErrCount, SafelyRemoveMedia, ControlUserLED, SaveEventArchive, LogMessage, CopyCodesysProject
Database Actions	All actions are not supported
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
- External Database (SQL4Automation)
- 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

Font files

Font files without "*Font Embeddability = Installable*" property (to be verified in the font file properties > details) are not loaded from the AGI Web unless the font is already installed in the operating system of the device running the browser

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).

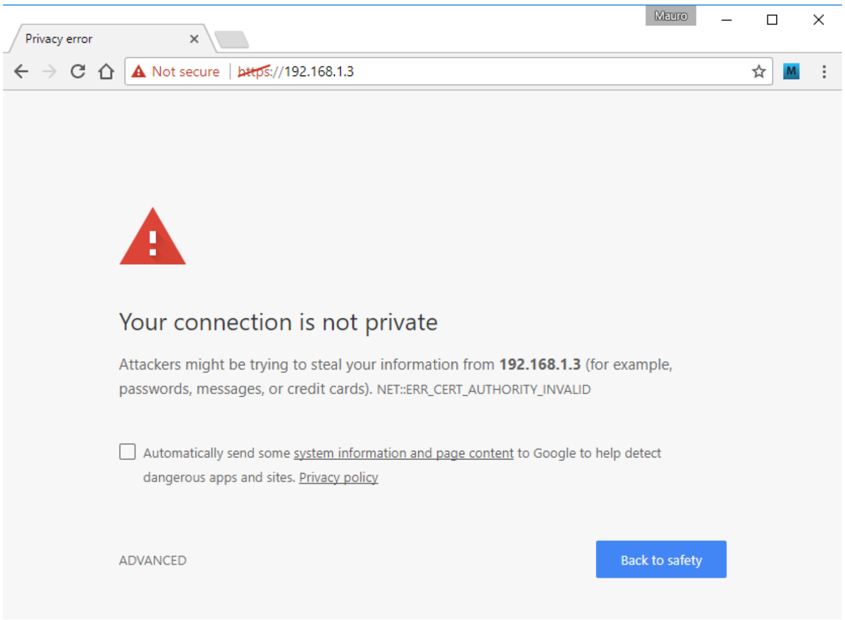
Secure Socket Layer (HTTPS)

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 firmed from a known Authority, you will get a warning message.

Simple click the ADVANCE button to continue.



Your connection is not private

Attackers might be trying to steal your information from **192.168.1.3** (for example, passwords, messages, or credit cards). NET::ERR_CERT_AUTHORITY_INVALID

☐ Automatically send some system information and page content to Google to help detect dangerous apps and sites. Privacy policy

Working with keypads in AGI Web

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

AGI Web requires JavaScript to provide interactivity with the server and the user. AGI Web 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 AGI Web pages.

Browser cache

AGI Web 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 AGI Web 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 AGI Web pages.

Caching is normally enabled by default, for optimal AGI Web performance make sure it has not been disabled.

Note: AGI Web 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 AGI Web 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.

43 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 567 for HMI devices on Linux platform, "Password protection" on page 577 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 334)
- Force remote login (See: "Force remote login" on page 344)

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:

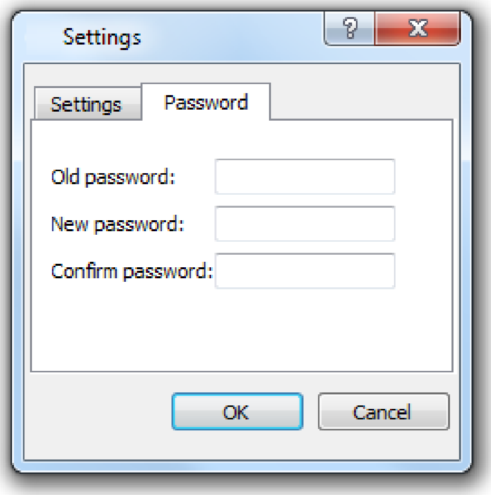
- Firewall
- Intrusion Prevention System (IPS)
- Network segmentation
- Virtual LAN (VLAN)
- Virtual Private Network (VPN)
- Security at physical access level (Port Security).

Further information, guidelines and standards regarding security in information technology: IEC 62443, ISO/IEC 27001.

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 547 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\DEIF A/S\buildNumber\server\config\RemoteUpdateConfig.xm*l.

Ports and firewalls

Here a list of all the ports used by AGI SW Pack components.

Port	Usage	Remote Access	Board Management	Runtime/Project Management
80/tcp	HTTP port	Yes	-	Yes
21/tcp	FTP cmd port	-	-	Yes
2100/tcp	Board port	-	Yes	-
16384-17407/tcp	FTP data port (passive mode)	-	Yes	Yes
990/udp	UDP broadcast (Device discovery)	-	Optional	Optional

Port	Usage	Remote Access	Board Management	Runtime/Project Management
991/udp	UDP broadcast (Device discovery)	-	Optional	Optional
998/udp	UDP broadcast (Device discovery)	-	Optional	Optional
999/udp	UDP broadcast (Device discovery)	-	Optional	Optional
5900/tcp	VNC Server	VNC only	-	-
5100/tcp	JS Remote Debugger	-	-	Optional
11740-11743/tcp 1217, 1740-1743/udp	CODESYS 3	-	-	-
48010/tcp	OPC UA Server	-	-	Optional
25/tcp	SMTP Server	-	-	Optional
See your MQTT Broker	MQTT	-	-	Optional

Remote access

Remote access is required to connect to HMI Runtime using:

- AGI Client
- Internet Browser

Runtime and project management ports

You use these ports to connect to HMI Runtime for operations such as update, installation and project download.

Board management ports

You use these ports to connect to the HMI device for Board operations such as BSP update, splash image download and so on.

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 AGI Creator.

44 Tips and tricks to improve performance

AGI Creator 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.

Static Optimization

Static optimization is a technique used in AGI Creator 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 AGI Creator, 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, AGI Creator 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.

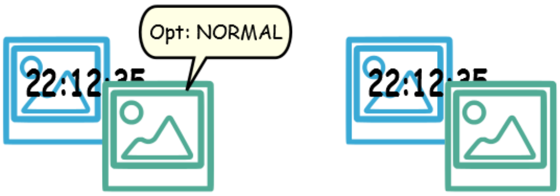
The **Static Optimization** page attribute enables and disables static optimization of the whole page. If it is set to **false** the optimization is totally disabled.

Finer control can be achieved setting the **Static Optimization** attribute of each single widget as follows:

- **Normal**: AGI Creator 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.

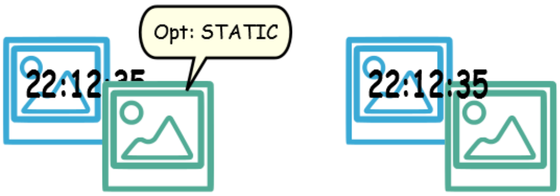
Note: In this case the automatic optimization will fail because it does not make any assumption on invisible areas which might be rendered at runtime.

- **Dynamic**: The widget is not optimized at all. Use this flag when a static widget needs to be changed by JavaScript.



a) Studio

b) Runtime



a) Studio

b) Runtime

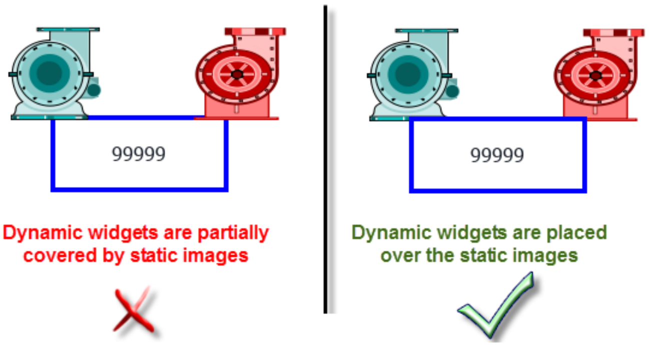
Tips for best performance

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.


99999
Dynamic widgets are partially covered by static images
99999
Dynamic widgets are placed over the static images

Supported image formats

AGI Creator 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	• Fast rendering • Well standardized	• Big file size • Fixed resolution
VECTOR (SVG)	• Small file size • Rescale without quality loss • Can handle dynamic properties	• Complex SVG images with many graphic items and layers can be slow to render. • Creating an optimized SVG is not simple. • Only Tiny 1.2 (http://www.w3.org/TR/SVGTiny12/) supported.

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 AGI Creator(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 untill 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.

45 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.

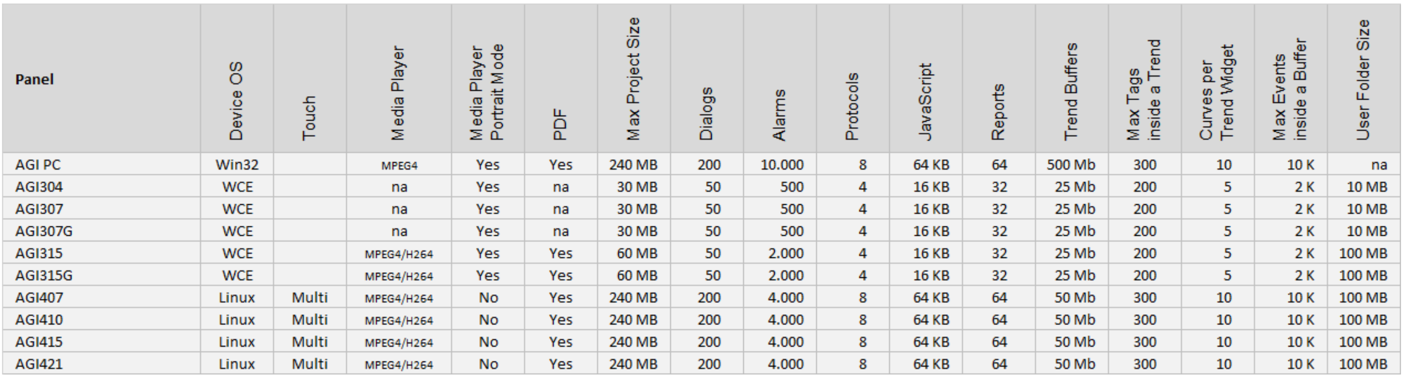
Table of functions and limits

Function	Max limit
Number of pages	1.000
Number of basic widgets	2.000 x page
Number of tags	10.000
Number of dialog pages	See "HMI devices capabilities" on the facing page
Number of dialog pages that can be open at the same time	5
Number of Recipes	32
Number of parameter sets for a recipe	1.000
Number of elements per Recipe	1.000
Number of user groups	50
Number of users	500
Number of concurrent remote clients	4
Number of schedulers	30
Number of alarms	See "HMI devices capabilities" on the facing page
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 "HMI devices capabilities" on the facing page
Memory reserved for trend buffer	See "HMI devices capabilities" on the facing page
Number of curves per trend widget	See "HMI devices capabilities" on the facing page
Number of curves per scatter diagram widget	10
Max number of trend table printable rows	10.000 on HMI Runtime 50.000 on AGI 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 "HMI

Function	Max limit
	devices capabilities" below)
Number of events per buffer	See "HMI devices capabilities" below
Number of event buffers	4
JavaScript file size per page	See "HMI devices capabilities" below
Size of project on disk	See "HMI devices capabilities" below
Number of indexed instances	100
Number of indexed alias	100
Number of indexed tag sets	30
Number of physical protocols	See "HMI devices capabilities" below
Number of reports	See "HMI devices capabilities" below
Number of reports pages	32
Max number of variables in variables widget	255
User folder size (UpdatePackage.zip)	See "HMI devices capabilities" below
Number of concurrent FTP sessions	4
FTP additional folders	5

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
AGI PC	Win32		MPEG4	Yes	Yes	240 MB	200	10.000	8	64 KB	64	500 Mb	300	10	10 K	na
AGI304	WCE		na	Yes	na	30 MB	50	500	4	16 KB	32	25 Mb	200	5	2 K	10 MB
AGI307	WCE		na	Yes	na	30 MB	50	500	4	16 KB	32	25 Mb	200	5	2 K	10 MB
AGI307G	WCE		na	Yes	na	30 MB	50	500	4	16 KB	32	25 Mb	200	5	2 K	10 MB
AGI315	WCE		MPEG4/H264	Yes	Yes	60 MB	50	2.000	4	16 KB	32	25 Mb	200	5	2 K	100 MB
AGI315G	WCE		MPEG4/H264	Yes	Yes	60 MB	50	2.000	4	16 KB	32	25 Mb	200	5	2 K	100 MB
AGI407	Linux	Multi	MPEG4/H264	No	Yes	240 MB	200	4.000	8	64 KB	64	50 Mb	300	10	10 K	100 MB
AGI410	Linux	Multi	MPEG4/H264	No	Yes	240 MB	200	4.000	8	64 KB	64	50 Mb	300	10	10 K	100 MB
AGI415	Linux	Multi	MPEG4/H264	No	Yes	240 MB	200	4.000	8	64 KB	64	50 Mb	300	10	10 K	100 MB
AGI421	Linux	Multi	MPEG4/H264	No	Yes	240 MB	200	4.000	8	64 KB	64	50 Mb	300	10	10 K	100 MB

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 AGI 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:

- AGI Creator 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 AGI Creator update,
- the HMI Runtime update can be done directly from AGI Creator using the Update Target command available in the Run\Manage Target dialog,
- projects created in a AGI Creator version no older than V1.00 (00) can be opened and handled by any newer version,
- projects created with older versions of AGI Creator, 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 AGI Creator is not ensured.

Important: Do not edit projects with a version of AGI Creator 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 the previous page), 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 542 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.

46 System Settings

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.

Note: the system components can be update even from the AGI Creator (see "Updating system components in HMI devices" on page 581 for details)

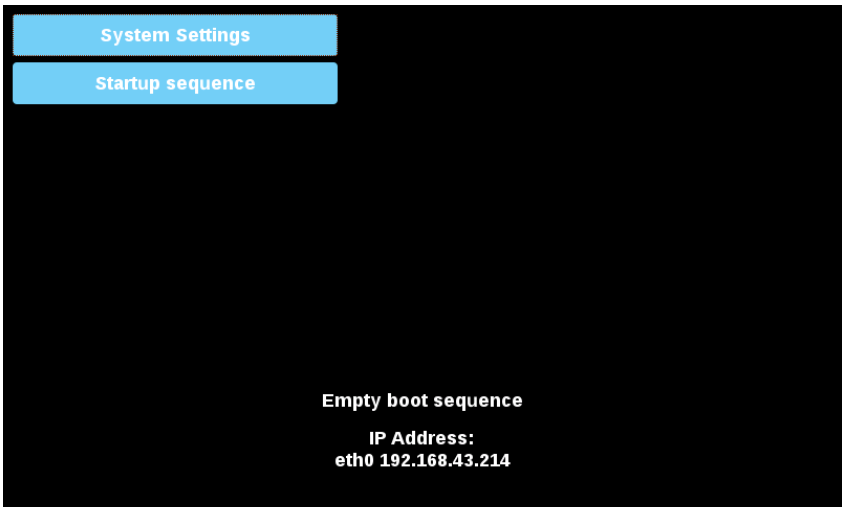
CAUTION: Working with the System Settings tool is a critical operation and, when not performed correctly, may cause product damages requiring service of the product. Contact technical support for assistance.

Linux Devices

AGI 400 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.

Runtime Installation

HMI devices are delivered from factory without Runtime, at first power up HMI shows the “Runtime Loader” screen.



Empty boot sequence

IP Address:
eth0 192.168.43.214

Runtime can be installed:

- Automatically, via Ethernet on first project download with AGI Creator
- Manually via USB Memory, creating an “Update Package”. (See the "Update package" on page 101 to create a runtime package)

Install Runtime via Ethernet

To install Runtime via Ethernet follow the "Download to HMI device" on page 98 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.

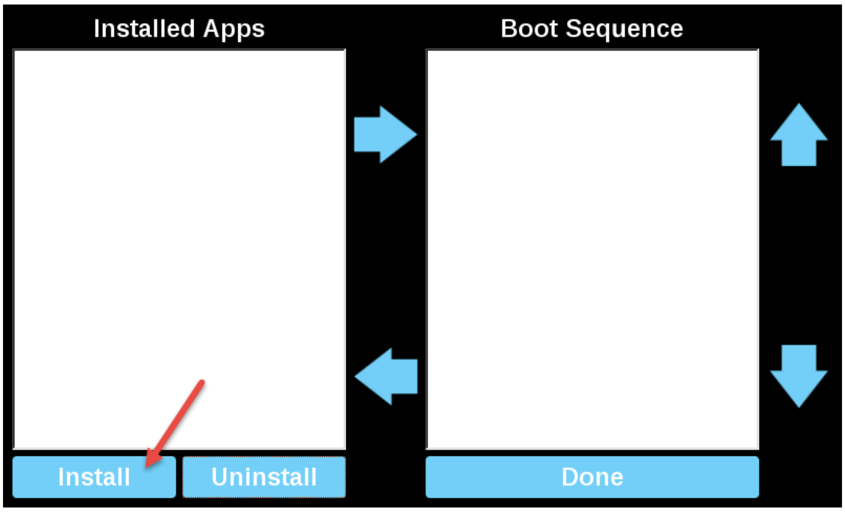
Install Runtime via USB Memory

To install Runtime, UpdatePackage or Backup Package via USB device follow this procedure:

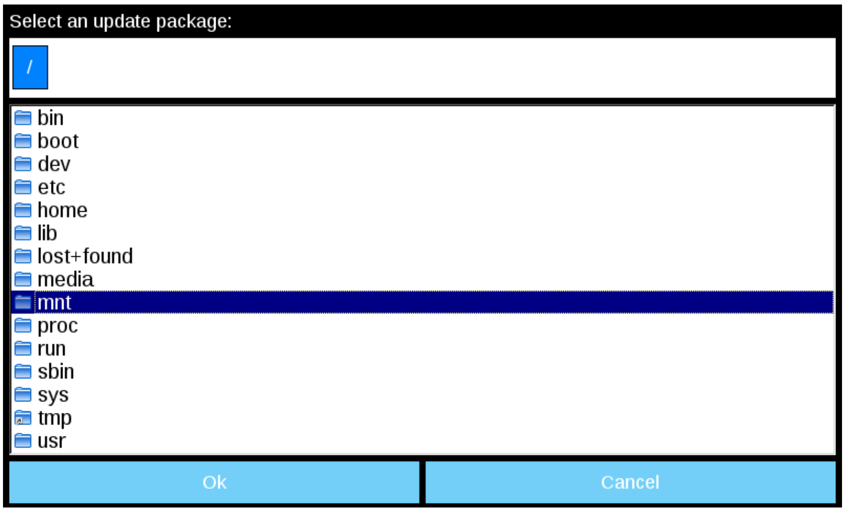
1. Create an Update Package from AGI Creator and copy into an empty USB memory stick

Note: File systems supported are FAT16/32 and Linux Ext2, Ext3 and Ext4.

2. On HMI select [Startup sequence], then [Install]



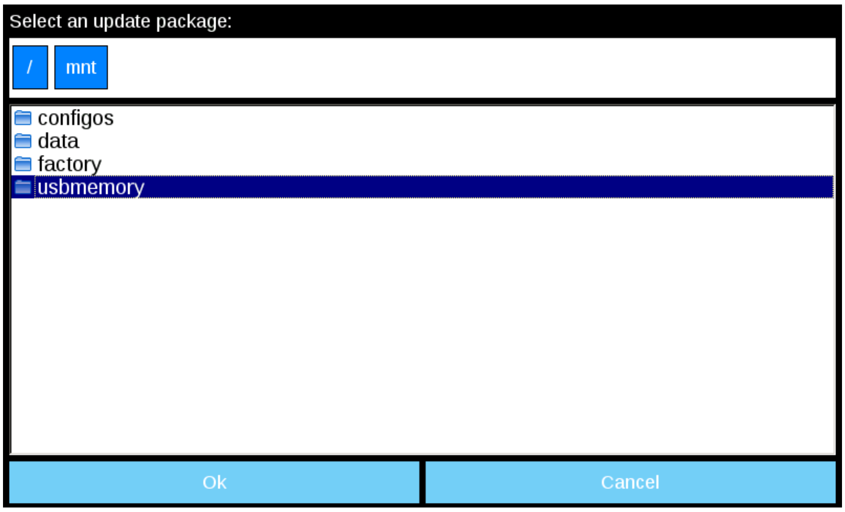
3. Double click on “mnt” to access this folder


Select an update package:

/

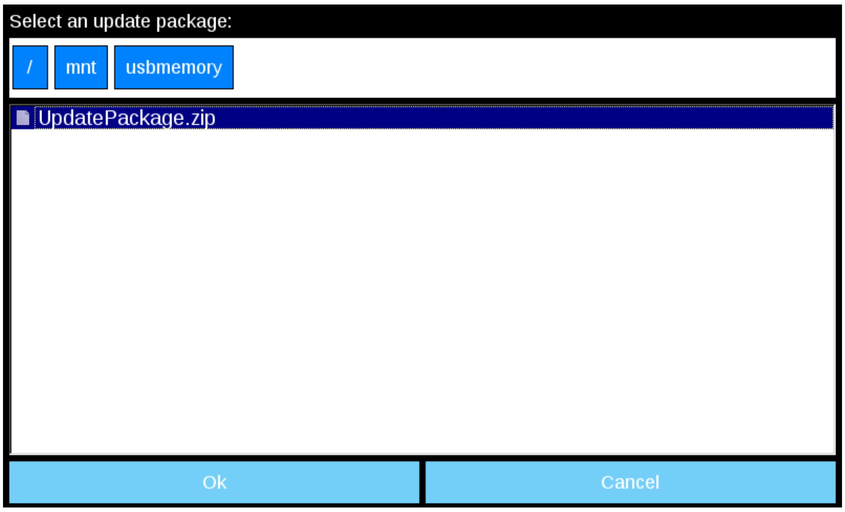
- bin
- boot
- dev
- etc
- home
- lib
- lost+found
- media
- mnt
- proc
- run
- sbin
- sys
- tmp
- usr

4. Then on “usbmemory”



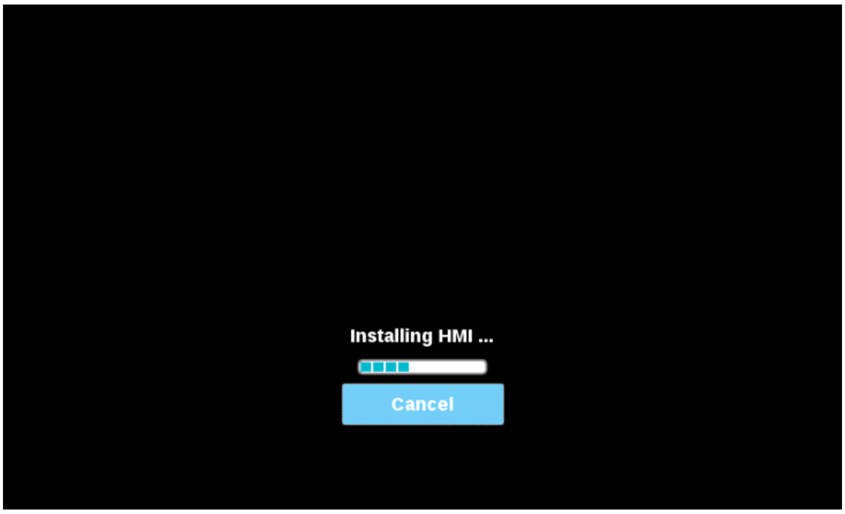
Select an update package:

5. Select “UpdatePackage.zip” and confirm with [Ok]



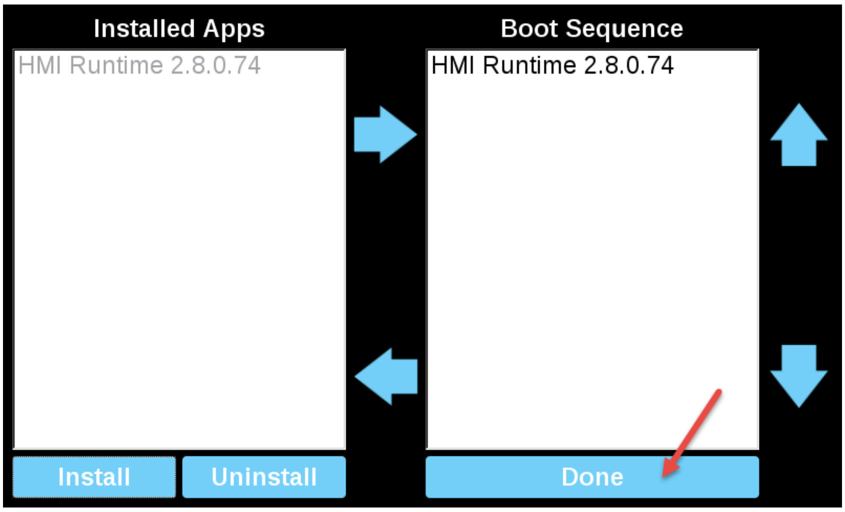
Select an update package:

6. The runtime installation begin

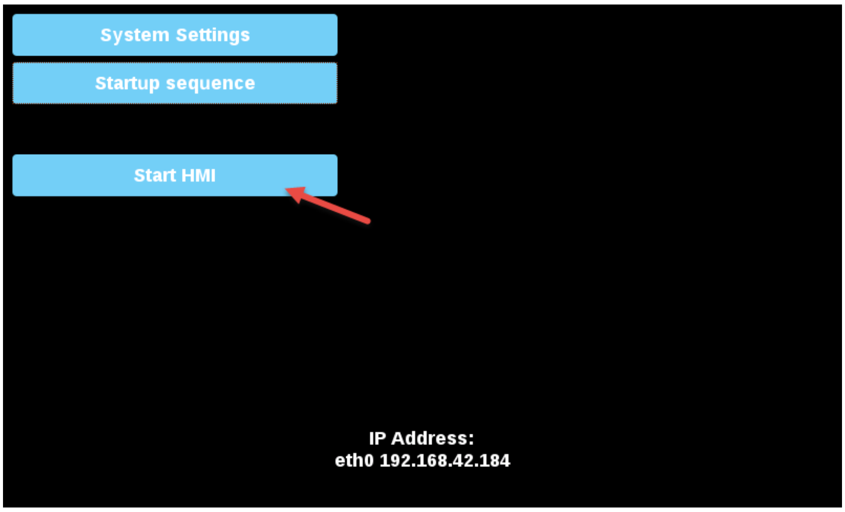


Installing HMI ...

7. At the end press "Done" button



8. Then "Start HMI" 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)

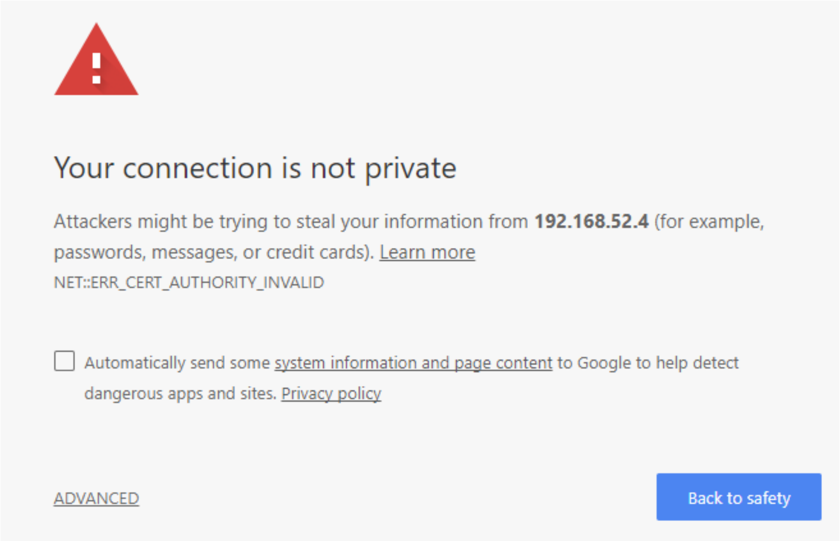
Accessing at the system settings from the HMI device do not require to enter a password until the default "admin" password is not changed.

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.



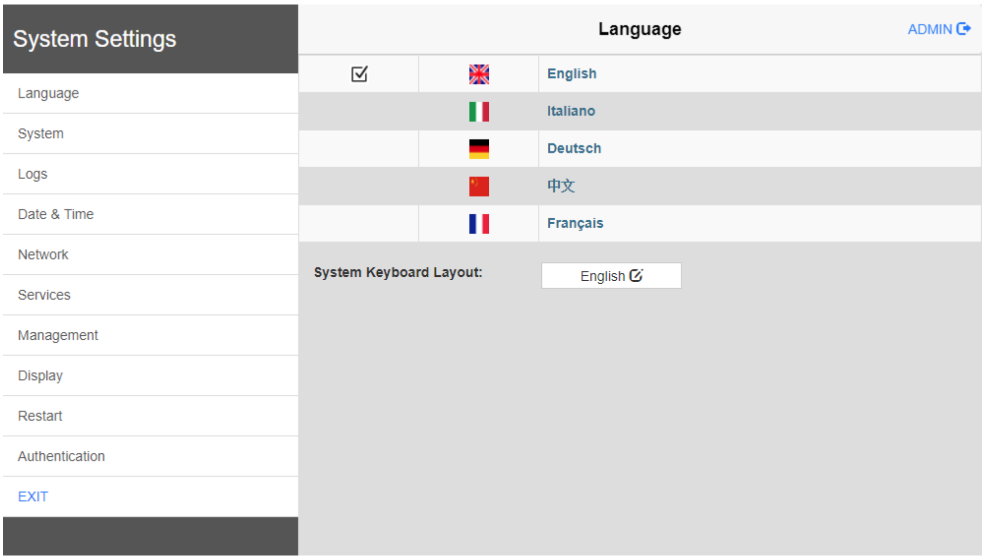
Your connection is not private

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

Browse through the options available in the menu on the left: the active item is highlighted and related information is displayed on the right.



☑		English
		Italiano
		Deutsch
		中文
		Français

System Keyboard Layout: English

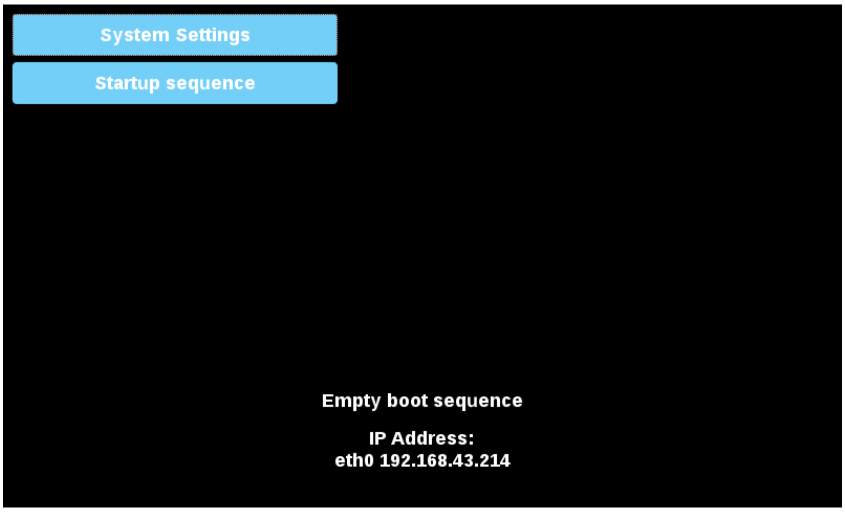
Default security protocols proposed by the HTTPS server in the AGI 400 HMI device are:

- 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

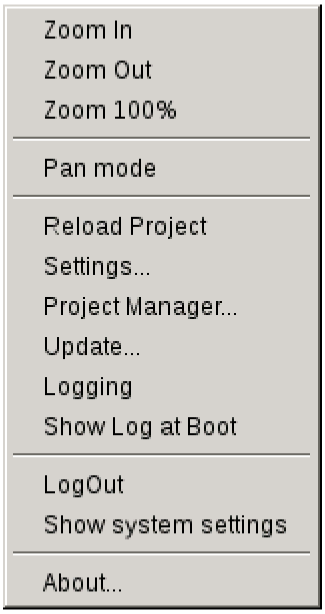
When Runtime is not installed, the System Settings is accessible from the Runtime Loader screen,



Empty boot sequence

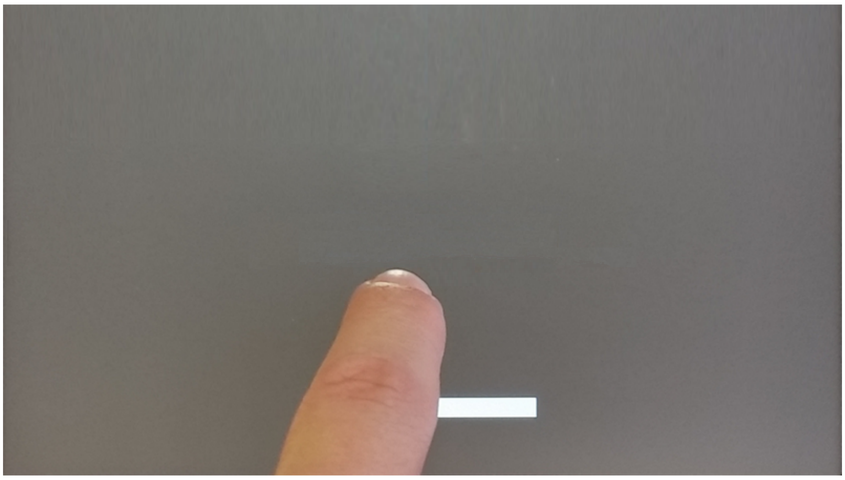
IP Address:
eth0 192.168.43.214

When Runtime is installed the System Settings is accessible selecting “Show System Settings” option of Context Menu,

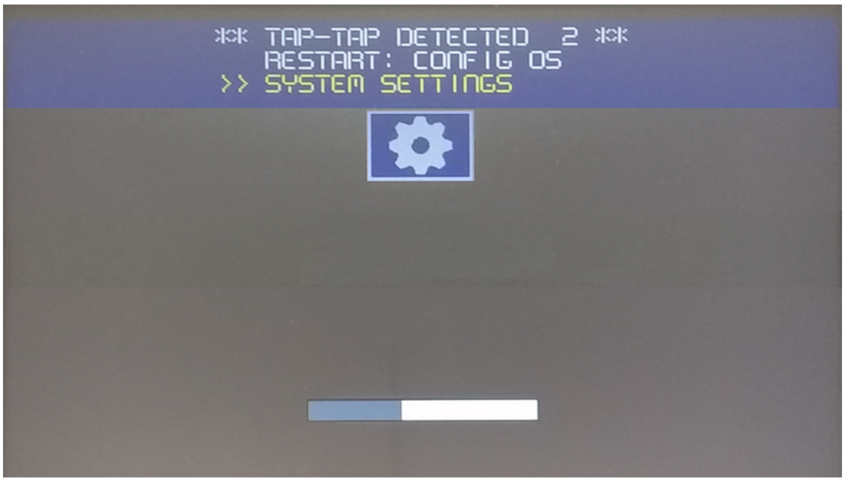


Enter System Settings via tap-tap procedure

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.

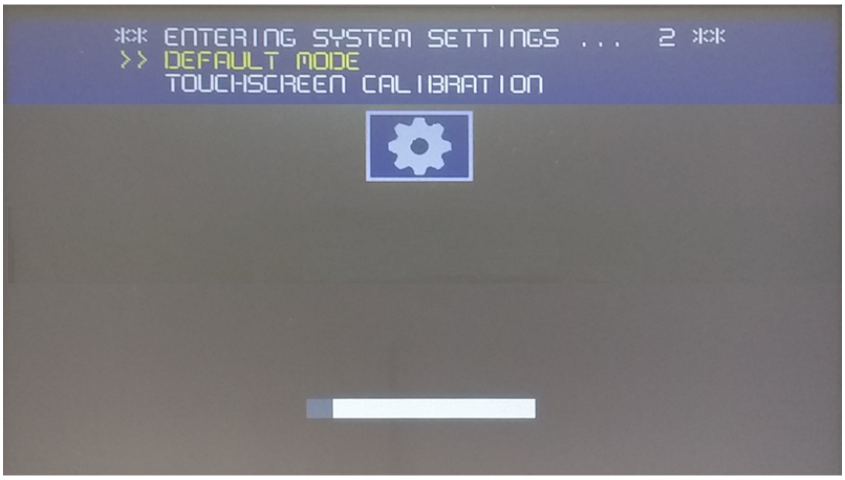


When “tap-tap detected” message appears on the top of the screen. Wait for 5 seconds (without touching the screen) to enter System Settings sub menu



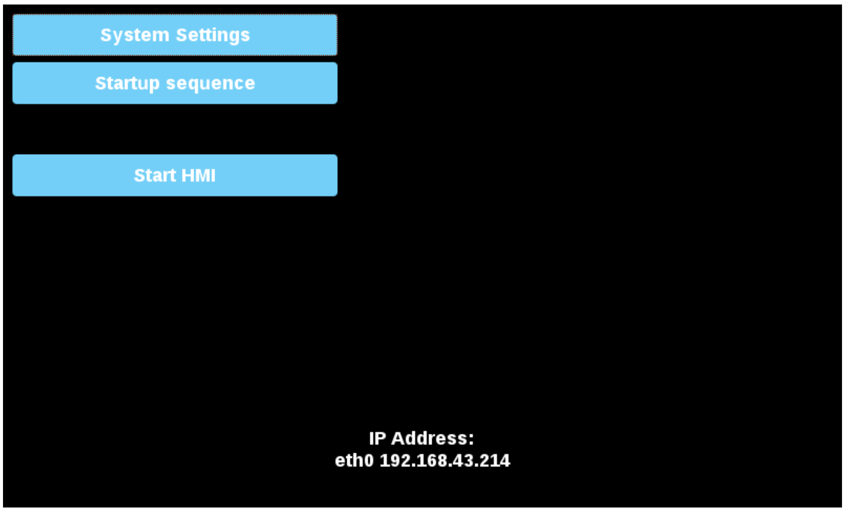
```
** TAP-TAP DETECTED  2 **
   RESTART: CONFIG OS
>> SYSTEM SETTINGS
```

Wait for 5 more seconds (without touching the screen) to enter Default Mode



```
** ENTERING SYSTEM SETTINGS ...   2 **
>> DEFAULT MODE
   TOUCHSCREEN CALIBRATION
```

Select "System Setting" from the HMI Default Mode screen



IP Address:
eth0 192.168.43.214

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.

Languages

Select the language for the system settings interface

- System keyboard layout: select the layout of the virtual keyboard

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
Current Timezone	Timezone region
Current Date **Local Time**	Date and Time can set manually only when the Automatic Update is disabled.
Automatic Update (NTP)	Enable to keep date and time synchronized from a remote server • NTP Server Specify the Internet NTP Server address 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. 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.
Accept NTP requests	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
General Settings	Device hostname Avahi Hostname (see "Avahi Daemon" on the facing page)
Network Interface	Network parameters of the available interfaces • DHCP • IP Address • Net Mask • Gateway
DNS	DNS Servers Generally provided from the DHCP servers, but can be modified in edit mode Search Domains Optional domains that will be used in concatenation with the provided urls

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.

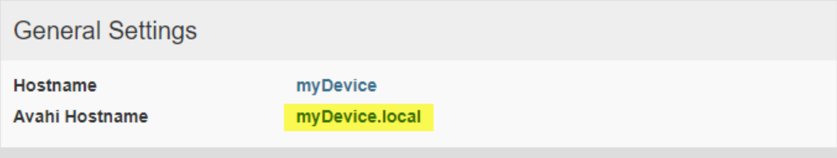
Autorun scripts from external storage

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

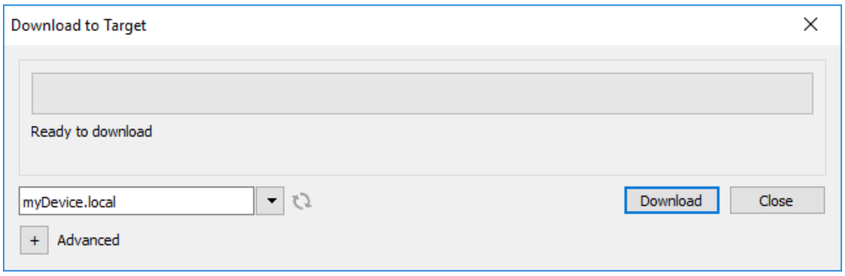
Avahi Daemon

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).



General Settings

Hostname	myDevice
Avahi Hostname	myDevice.local



Download to Target

Ready to download

myDevice.local

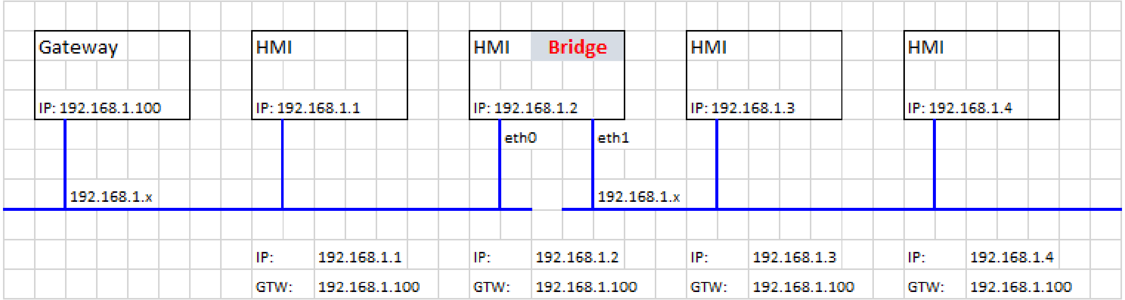
Avahi Daemon runs on UDP port 5353

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
Enabled	Enable the DHCP Server on the selected interface
Start IP **Stop IP**	IP addresses distributed from the DHCP Server
Gateway	The gateway address
Netmask	The provided netmask
DNS Server	The DNS server address
Lease Time (seconds)	Lease time, default is 86400s (1 day) Acceptable values are from 60s to 864000s (10 days)

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).

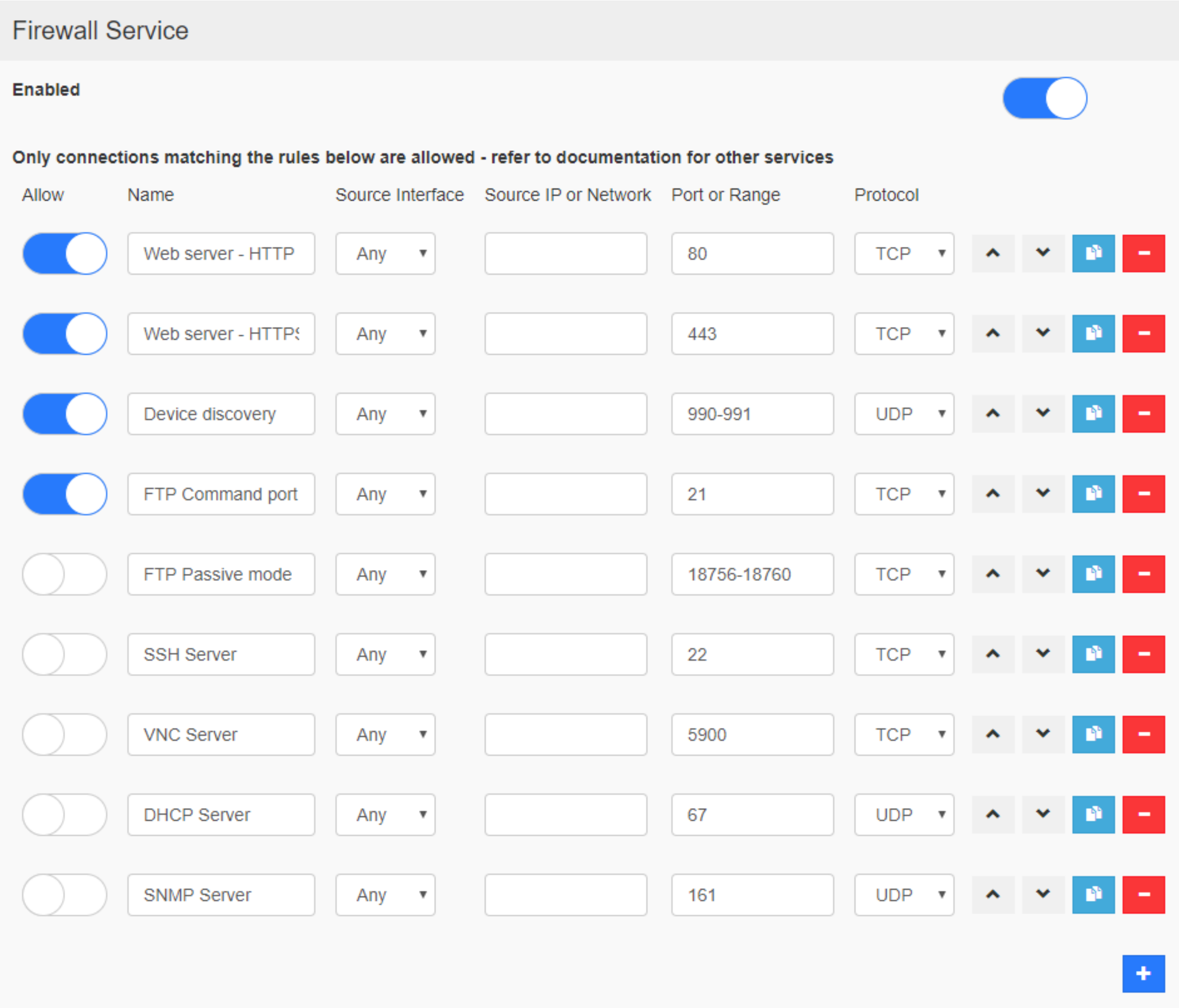
To obtain best performance, in addition of enabling the fast boot mode, it is recommended to:

- 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
☑	Web server - HTTP	Any		80	TCP
☑	Web server - HTTPS	Any		443	TCP
☑	Device discovery	Any		990-991	UDP
☑	FTP Command port	Any		21	TCP
□	FTP Passive mode	Any		18756-18760	TCP
□	SSH Server	Any		22	TCP
□	VNC Server	Any		5900	TCP
□	DHCP Server	Any		67	UDP
□	SNMP Server	Any		161	UDP

Notes:

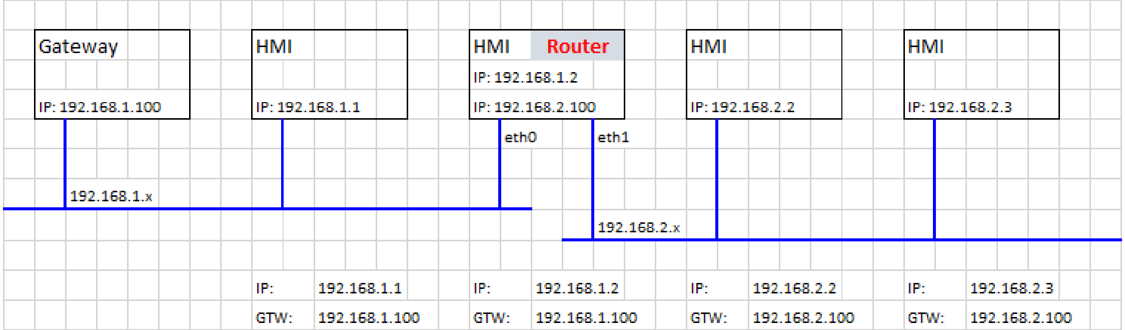
- The firewall is based on IP tables which operates only at layer 3 (layer 2 packets won't be filtered, e.g. ARP)
- Only INPUT and FORWARD packets are filtered, not OUTPUT
- PING/ICMP echo reply packets are always allowed
- Internet sharing scenarios (e.g. 3g or wifi connection to endpoints) are not supported
- Packets filtered by the firewall are dropped

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 564).

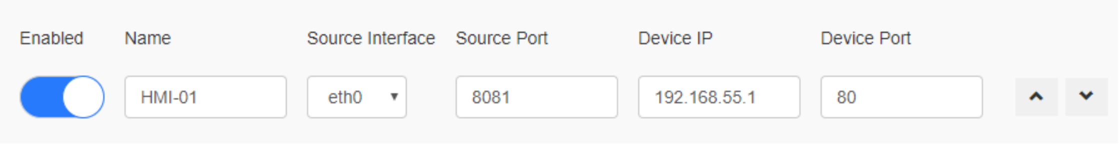
Router Service

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.



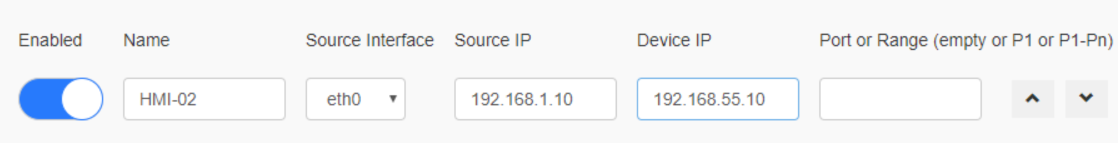
Enabled	Name	Source Interface	Source Port	Device IP	Device Port
●	HMI-01	eth0	8081	192.168.55.1	80

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)
●	HMI-02	eth0	192.168.1.10	192.168.55.10	

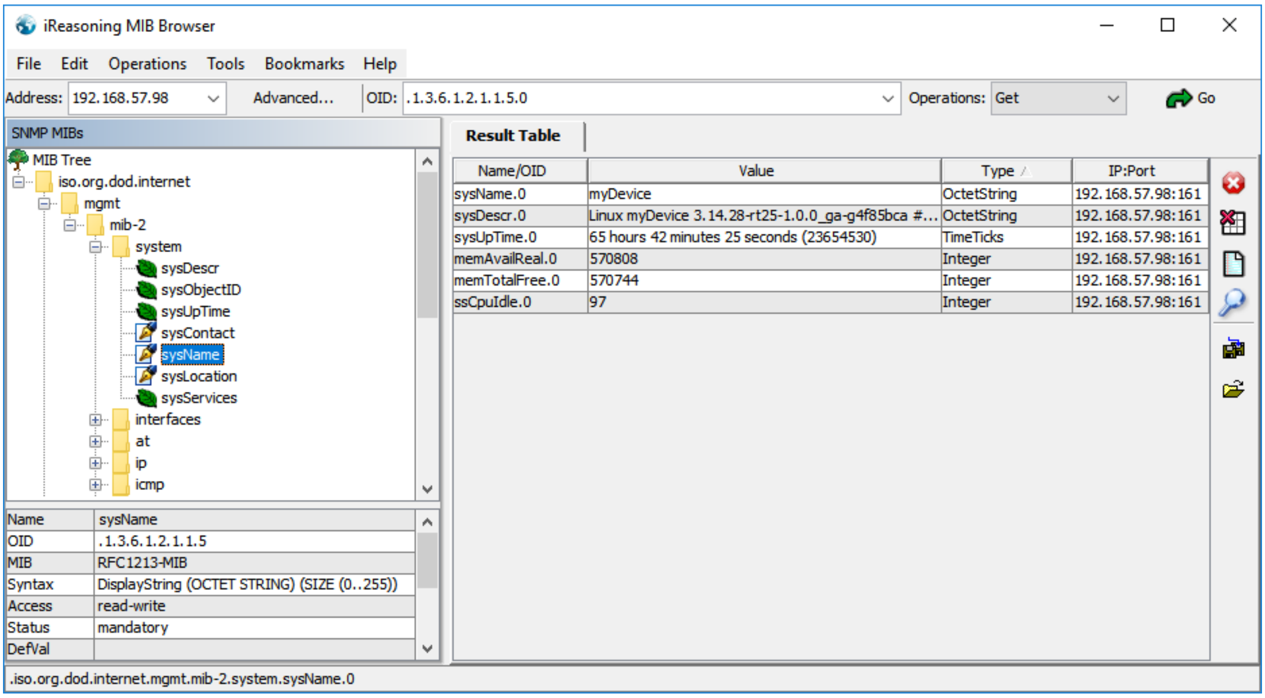
Show loading bar during boot

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.



Address: 192.168.57.98

OID: .1.3.6.1.2.1.1.5.0

Operations: Get

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

Name	sysName
OID	.1.3.6.1.2.1.1.5
MIB	RFC1213-MIB
Syntax	DisplayString (OCTET STRING) (SIZE (0..255))
Access	read-write
Status	mandatory
DefVal	

.iso.org.dod.internet.mgmt.mib-2.system.sysName.0

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

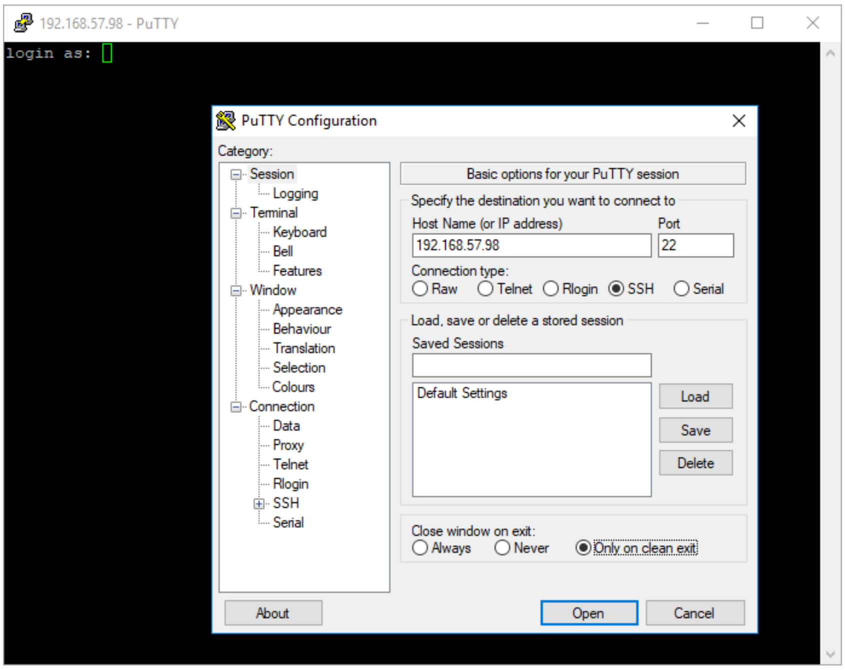
SNMP Server runs on UDP port 161

This service is designed to be used during the development phase. For security reasons, remember to disable the service before switch to production.

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 563 chapter to additional information.



192.168.57.98 - PuTTY

```
login as:
```

PuTTY Configuration

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): 192.168.57.98

Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Load, save or delete a stored session

Saved Sessions

Default Settings

Close window on exit: ○ Always ○ Never ● Only on clean exit

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.

- Port: VNC Server listens for connections on TCP port 5900 (default)
- Multiple clients: allow multiple sessions on the same port (if disabled, previously logged clients are disconnected upon a new incoming connection)
- View only: do not allow active user interactions (clients can only watch)
- Encryption: activate SSL encryption of connections (not widely supported - check client compatibility)
- Authentication: 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)

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 the next page 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

Restart

HMI device restart command

Authentication

Enter in edit mode to change the authentication passwords.

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

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
Device Name	The name of your device
Organization	The legal name of your organization
Unit	The division of your organization handling the certificate

Parameter	Description
State	The state/region where your organization is located
Location	The city where your organization is located
Country	The two-letter ISO code for the country where your organization is location
Valid (days)	Validity of the certificate
Key Length	Number of bits of the key used from the cryptographic algorithm

Managed certificates are base64 encoding

Required BSP v1.0.239 or greater

EXIT

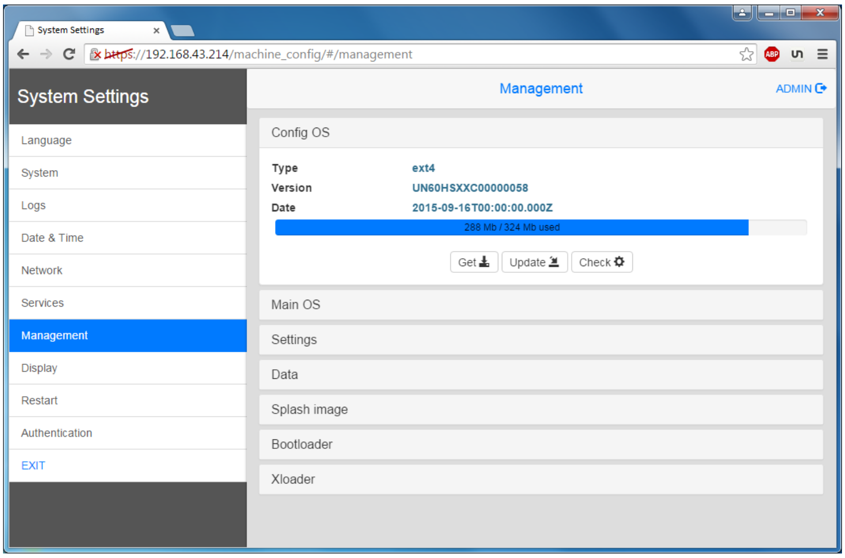
Exit from the System Setting tool.

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 provided from tech support).

The system components of the AGI 400 device can update locally using an 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.

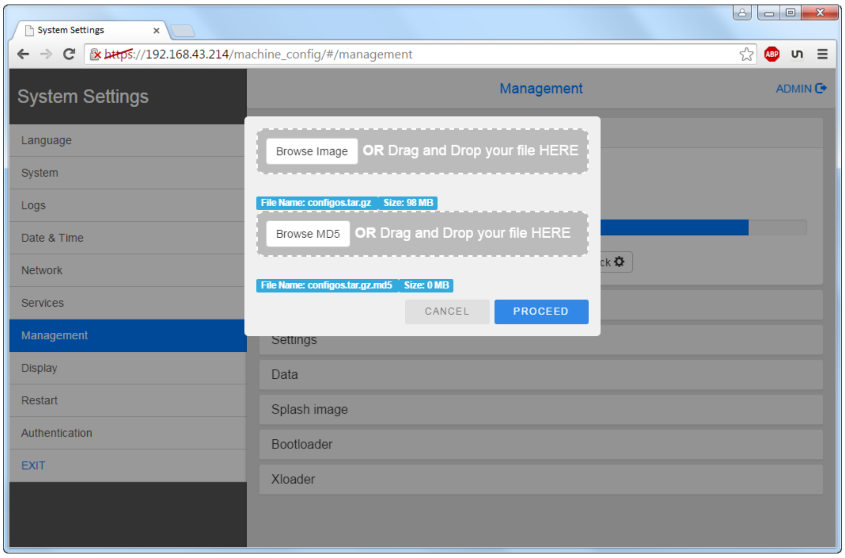


~~https~~://192.168.43.214/machine_config/#/management

Type ext4
Version UN60HSXXC00000058
Date 2015-09-16T00:00:00.000Z
288 Mb / 324 Mb used

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.



File Name: configos.tar.gz Size: 98 MB

File Name: configos.tar.gz.md5 Size: 0 MB

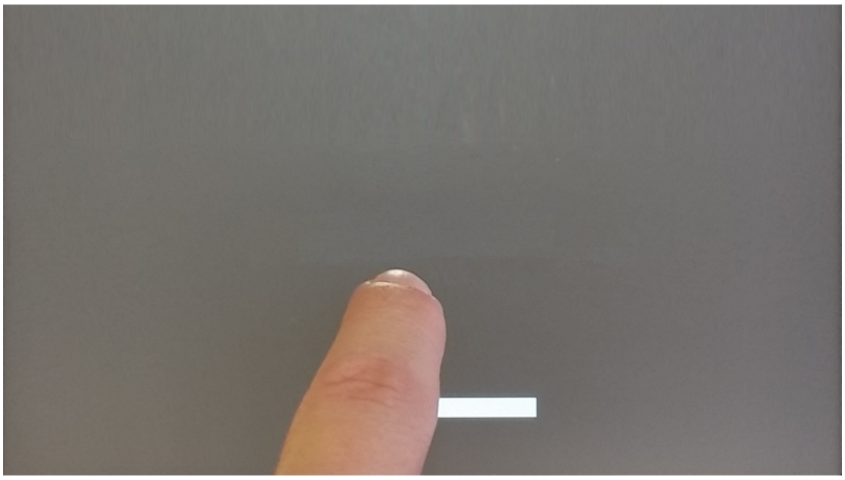
Important: Do not turn off the device while a system component is being upgraded.

At the end of the component update, restart HMI and leave it starting normally.

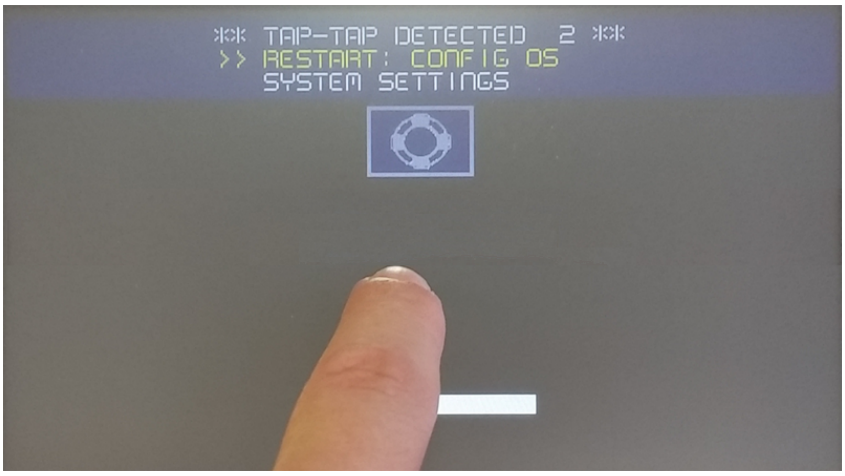
Enter System Settings in Config OS mode via tap-tap procedure

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.

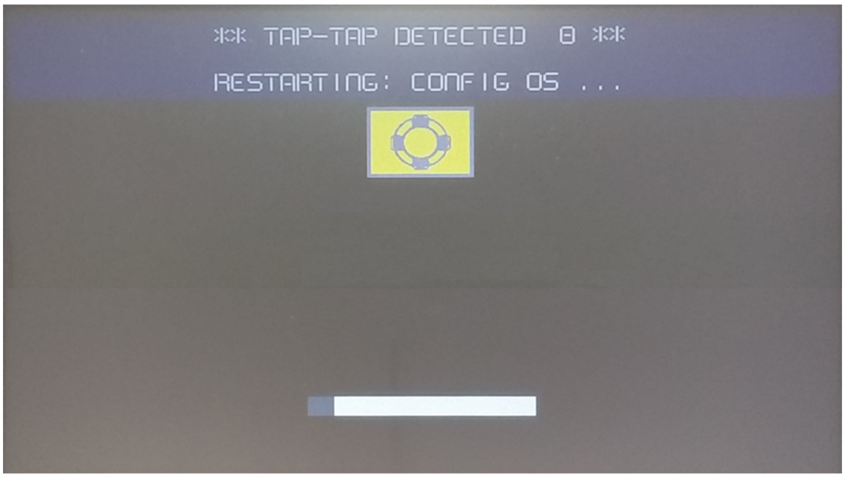
Tap-tap consist in a sequence of several touch activations by simple means of the finger tapping the touch screen performed during the power-up phase and started immediately after the HMI is powered on.



When “tap-tap detected” message appears on the top of the screen, press and hold the finger on touchscreen, to select “Restart: Config OS”



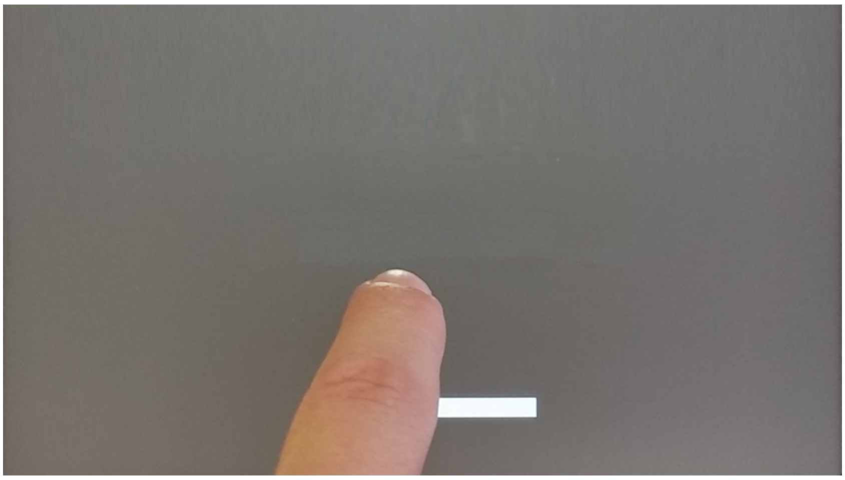
HMI will restart into System Settings in Config OS mode:



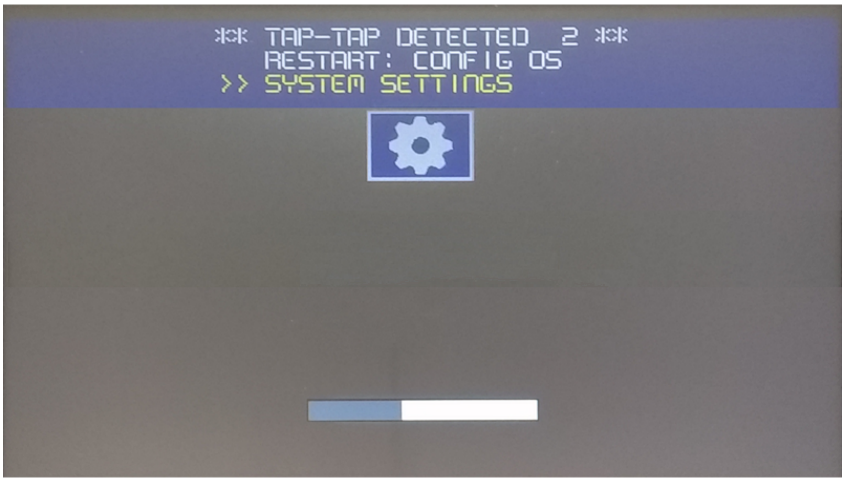
Touchscreen calibration

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.

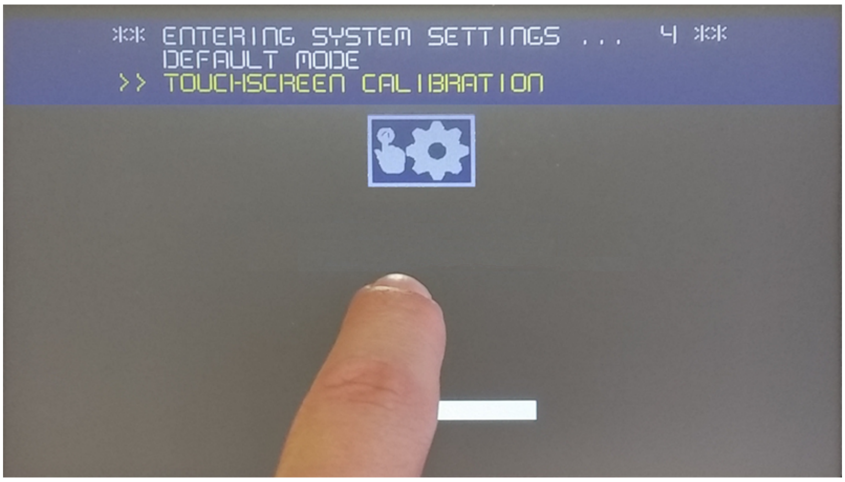


When “tap-tap detected” message appears on the top of the screen, wait for 5 seconds (without touching the screen) to enter System Settings sub menu



```
** TAP-TAP DETECTED  2 **
   RESTART: CONFIG OS
>> SYSTEM SETTINGS
```

Press on touch screen, “Touchscreen calibration” voice will be highlighted in yellow, hold pressed for few seconds until touchscreen calibration procedure starts



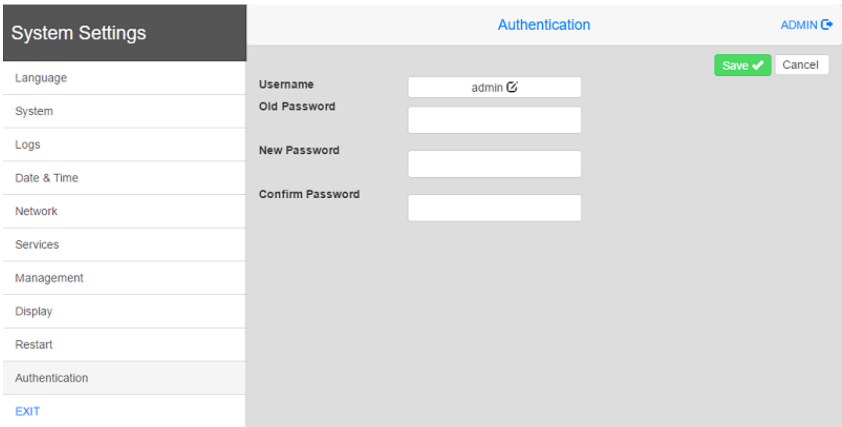
```
** ENTERING SYSTEM SETTINGS ...   4 **
   DEFAULT MODE
>> TOUCHSCREEN CALIBRATION
```

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 551, activate the edit mode and select the username to change the associated password.



Username: admin

Old Password

New Password

Confirm Password

Password for admin user can modified even from the context menu of theHMI Runtime (see "Context menu options" on page 8 for details) and from the update package (see "Update package" on page 101 for details).

Accessing at the system settings from the HMI device do not require to enter a password until the default "admin" password is not changed.

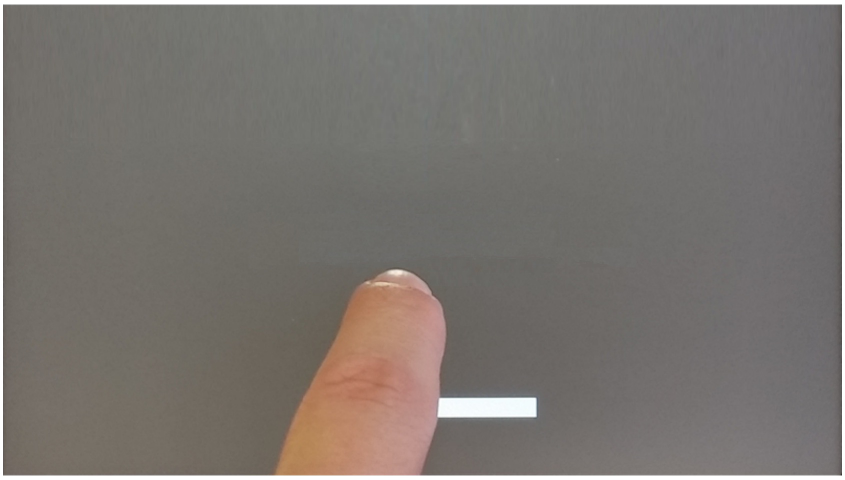
There are two usernames:

- Administrator username with full access right is "**admin**" with default password "admin".
- Generic username is “**user**” with default password “user”

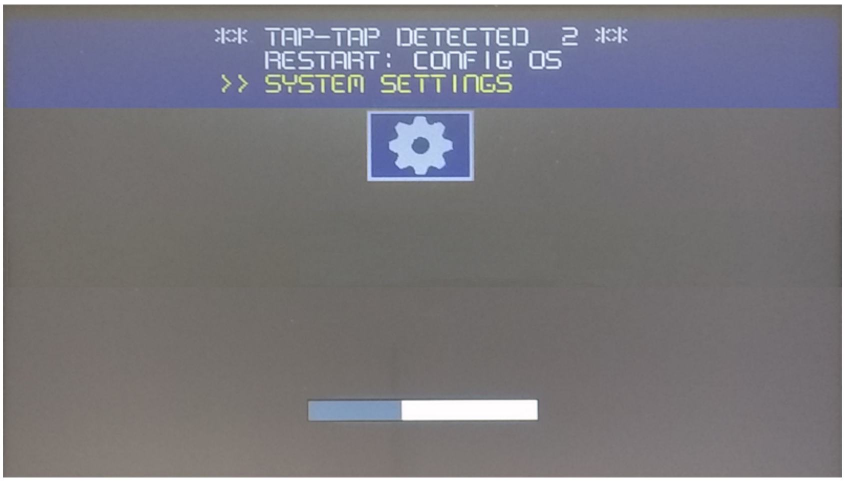
Factory restore

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.

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.

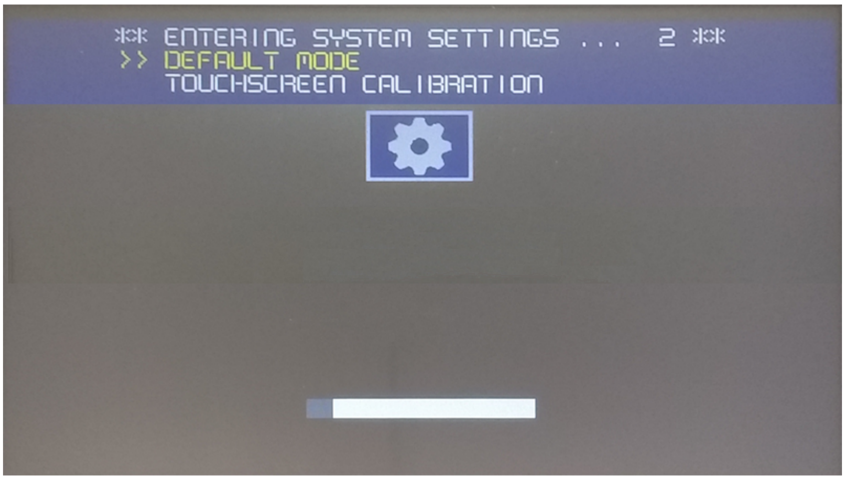


When “tap-tap detected” message appears on the top of the screen. Wait for 5 seconds (without touching the screen) to enter System Settings sub menu



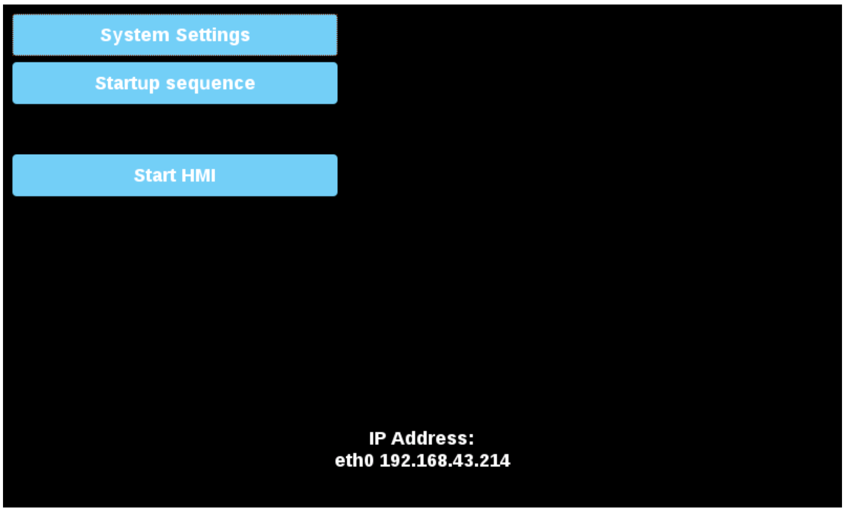
```
** TAP-TAP DETECTED  2 **
   RESTART: CONFIG OS
>> SYSTEM SETTINGS
```

Wait for 5 more seconds (without touching the screen) to enter Default Mode



```
** ENTERING SYSTEM SETTINGS ...   2 **
>> DEFAULT MODE
   TOUCHSCREEN CALIBRATION
```

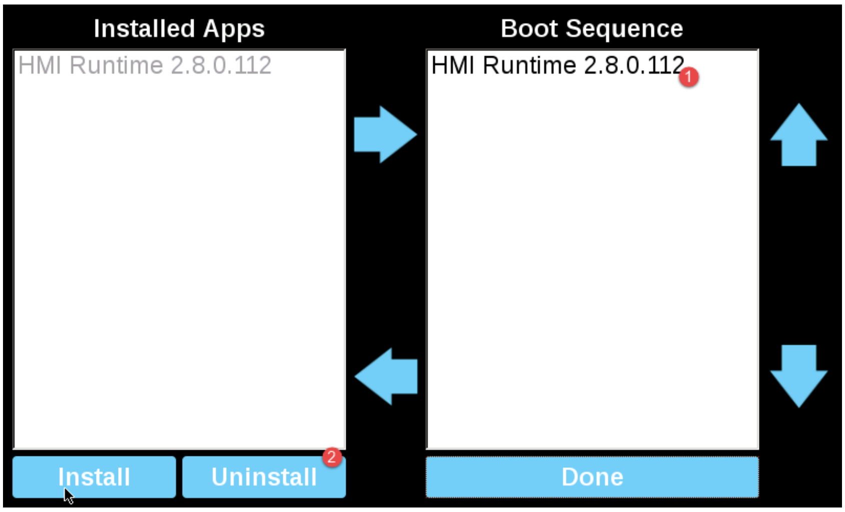
To uninstall the Runtime from HMI in Default Mode screen select [Startup Sequence]:



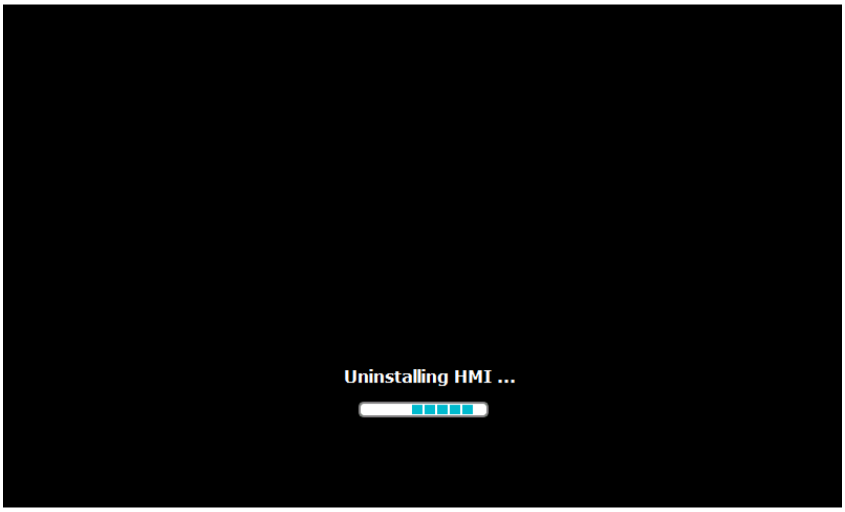
IP Address:
eth0 192.168.43.214

From the installed applications view:

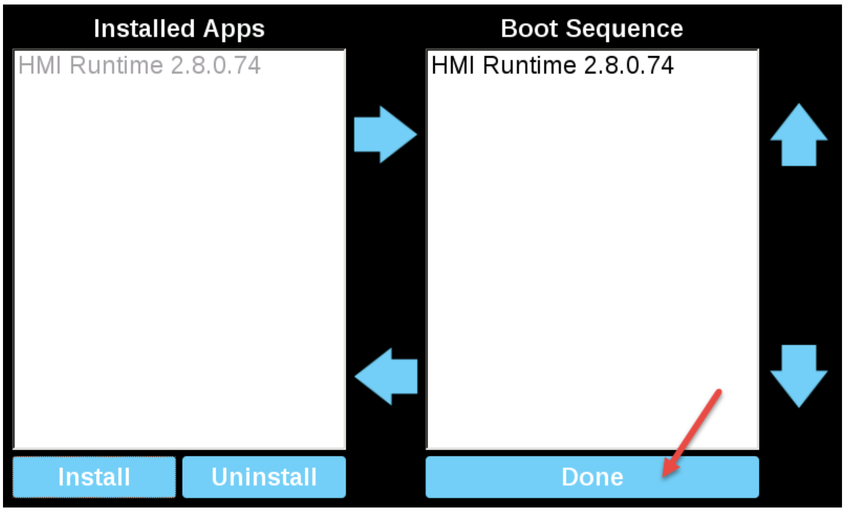
1. Select the Runtime you want to remove
2. Uninstall the selected Runtime



Runtime uninstall process will be performed:


Uninstalling HMI ...

At the end press "Done" button



Configuration mode

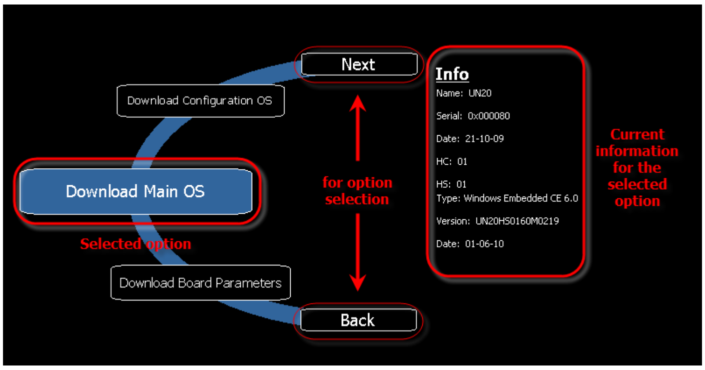
In the case that it is not even possible to boot the device, there is a special procedure to recovery the device by booting it in a special mode called configuration mode. From this mode you can open the device management dialog from where you can delete user data, restore system setting or update the 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 heart 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.

WinCE Devices

The System Settings tool includes a rotating menu, and navigation buttons to scroll between the available options.


Next
Download Configuration OS
Download Main OS
Selected option
Download Board Parameters
Back
for option selection
Info
Current information for the selected option

Name: UN20

Serial: 0x000080

Date: 21-10-09

HC: 01

HS: 01

Type: Windows Embedded CE 6.0

Version: UN20HS0160M0219

Date: 01-06-10

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.

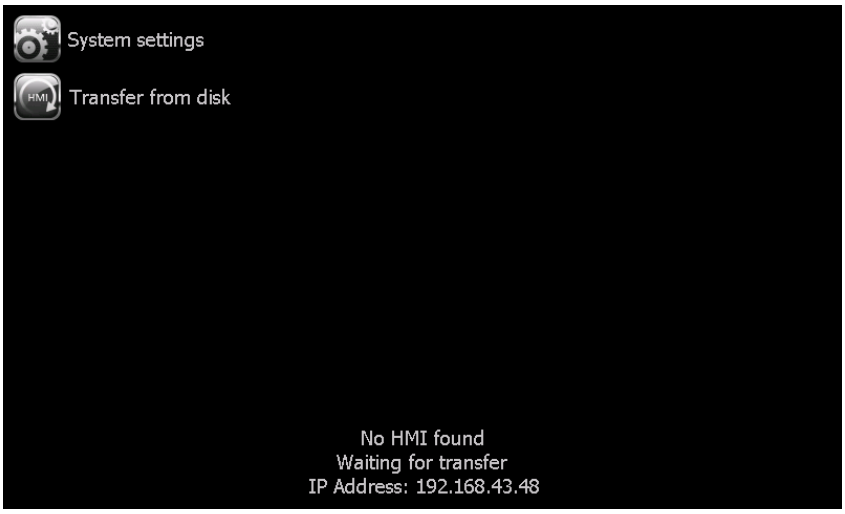
The System Settings tool can be used in two operating modes:

- User mode
- System mode.

For each mode different options are available.

Runtime Installation

HMI devices are delivered from factory without Runtime, at first power up HMI shows the “Runtime Loader” screen.



No HMI found
Waiting for transfer
IP Address: 192.168.43.48

Runtime can be installed:

- Automatically, via Ethernet on first project download with AGI Creator
- 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 98 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:

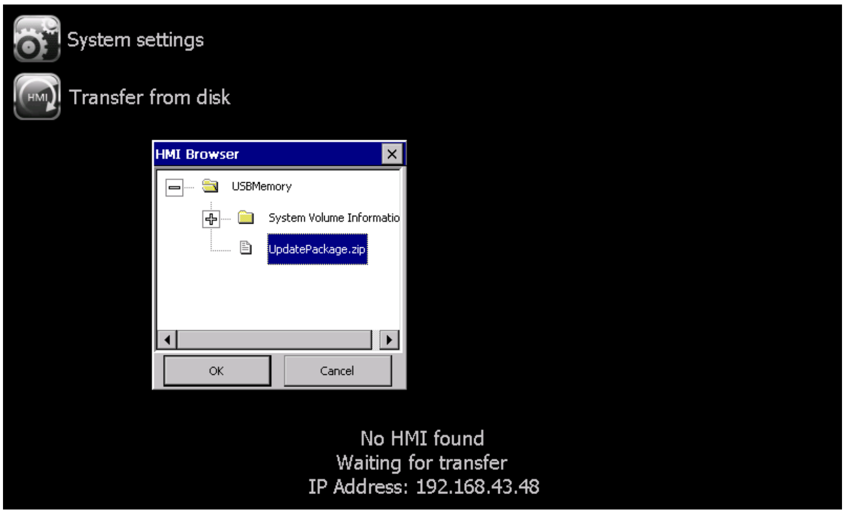
- *Automatically via DHCP server.* This option is enabled by default. If a DHCP server is available on the network IP address will be assigned automatically by the server.
- *Automatically via Auto-IP feature.* If DHCP assignment is enabled but no DHCP server is available on the network the HMI assigns 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 AGI Creator and copy into an empty USB memory stick

2. On HMI select [Transfer from disk] and select the UpdatePackage.zip to load.



No HMI found
Waiting for transfer
IP Address: 192.168.43.48

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
Calibrate Touch	Calibrate the touch screen
Display settings	Control backlight inactivity timeout and brightness
Time	Set HMI device date and time manually or configure NTP servers
Regional Settings	Select or customize the regional setting parameters
BSP Settings	Display operating system version and unit operating timers to control buzzer and battery led.
Network	Sets IP address and other network settings
Plug-in List	List the plug-in modules installed and recognized by the system.

Element	Description
	Note: this option may not be supported by all platforms and all versions.
Close	Closes the system setting page
Restart	Restart the HMI device - Main OS Restart the HMI device in the operating mode - Configuration OS Restart the HIM device with System Setting tool active in System Mode

Elements available in System Mode

In addition to those available in User Mode, the following features are also available:

Element	Description
Format Flash	Formats the internal device flash disk. All projects and the HMI Runtime will be erased, returning the device to its factory settings.
Restore Factory Settings	Restores factory settings as an alternative to Format Flash, in a more flexible way. The following options are available: **Uninstall HMI**: 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. **Clear System Settings**: resets system parameters (registry settings) and deletes the following files: *\\Flash\\Documents and Settings\\system.hv* *\\Flash\\Documents and Settings\\default\\user.hv* *\\Flash\\Documents and Settings\\default.mky* *\\Flash\\Documents and Settings\\default.vol* System Mode password is also reset. **Clear Controller Application**: clears current folders used by CODESYS V3 internal controllers for applications: - *\Flash\QtHmi\RTS\APP\*.** - *\Flash\QtHmi\RTS\VISU\*.** - *\Flash\QtHmi\codesys\** - *\Flash\$SysData$\codesys** **Clear sysdata settings**: clears \Flash\$SysData$ folder *Service call: To be used only by technical support to fix display problems.*

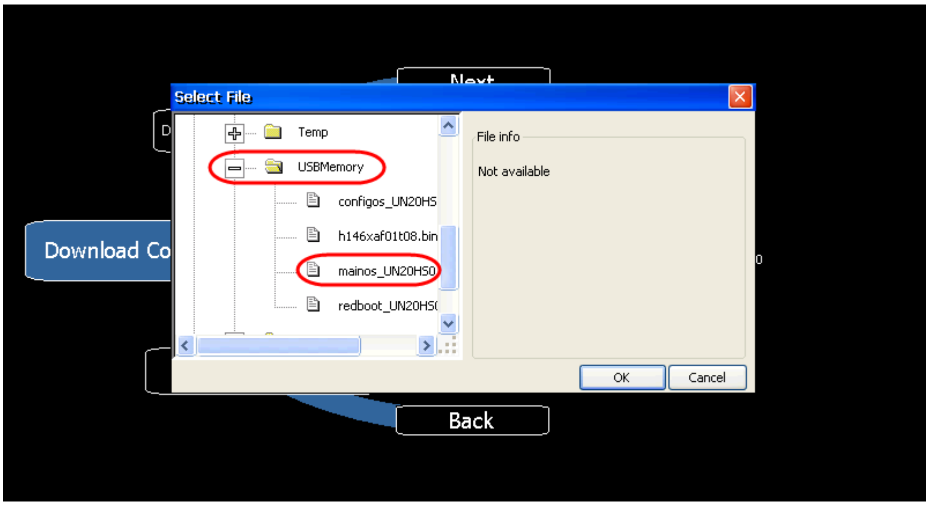
Element	Description
	Note: Not all these options are available for all HMI devices and BSPs.
Resize Image Area	Resizes the flash memory reserved to store the splash screen image displayed at power up. Default settings are normally suitable for all units.
Download Configuration OS	Checks and upgrades the current version of the operating system used in System Mode
Download Main OS	Checks and upgrades the current version of the main operating system
Download Splash Image	Loads a new file for the splash screen image displayed by the unit at power up. Tip: Update the splash screen image directly from the AGI Creator programming software. See "Update of system components from the application" on page 582 for details.
Download Bootloader	Checks and upgrades the current version of the system boot loader.
Download Main FPGA	Checks and upgrades the current version of the main FPGA file. This function may not be available for all platforms and versions.
Download Safe FPGA	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.
Download System Supervisor	Checks and upgrades the current version of the system supervisor firmware (used for the RTC and power supply handling).
Upload Configuration OS **Upload Main OS** **Upload Splash Image** **Upload Bootloader** **Upload Main FPGA** **Upload Safe FPGA** **Upload System Supervisor**	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 573 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 in 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 **Run**> **Update Package** 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 **Important: Use the same file for updating Main and Safe FPGA components.**
System Supervisor	Firmware of the system supervisor controller (for example: packaged_GekkoZigBee_v4.13.bin). *The System Supervisor component can be upgraded from v4.13 or above.* **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.**

Touchscreen calibration

System Setting Calibration allows to calibrate Touchscreen device, can be accessed from System Settings

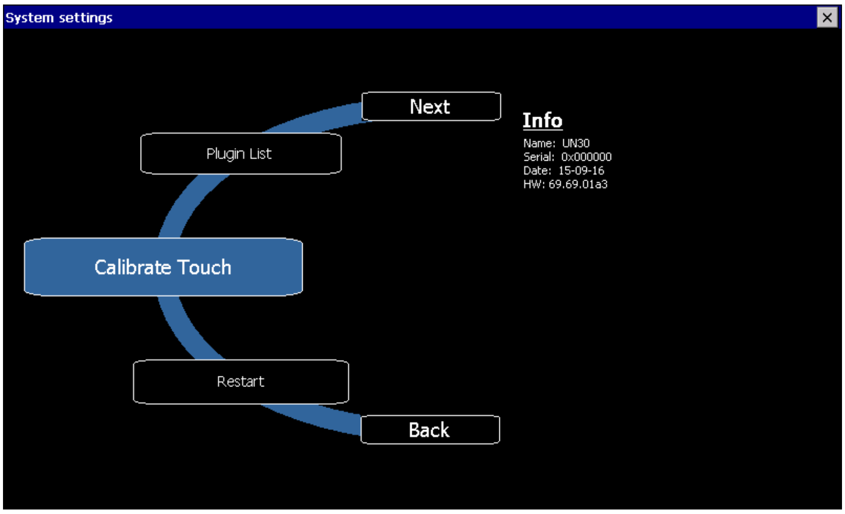
To access System Settings:

- 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.



Info

Name: UN30
Serial: 0x000000
Date: 15-09-16
HW: 69.69.01a3

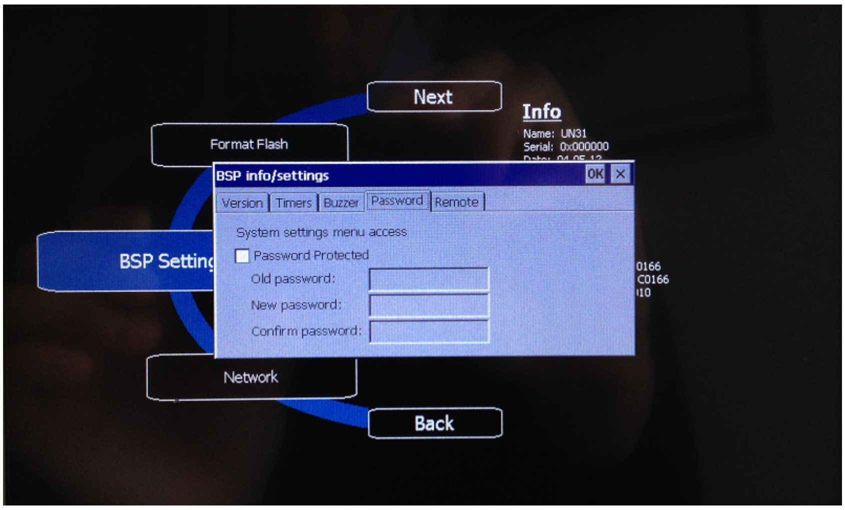
Password protection

Internal password of the HMI device can be defined from the System Settings in System Mode (see "System Settings" on page 573 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.



BSP info/settings

Version | Timers | Buzzer | Password | Remote

System settings menu access

□ Password Protected

Old password:

New password:

Confirm password:

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 modified even from the context menu of theHMI Runtime (see "Context menu options" on page 8 for details) and from the update package (see "Update package" on page 101for 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 573 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

47 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 547 for details)
- From AGI Creator 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.

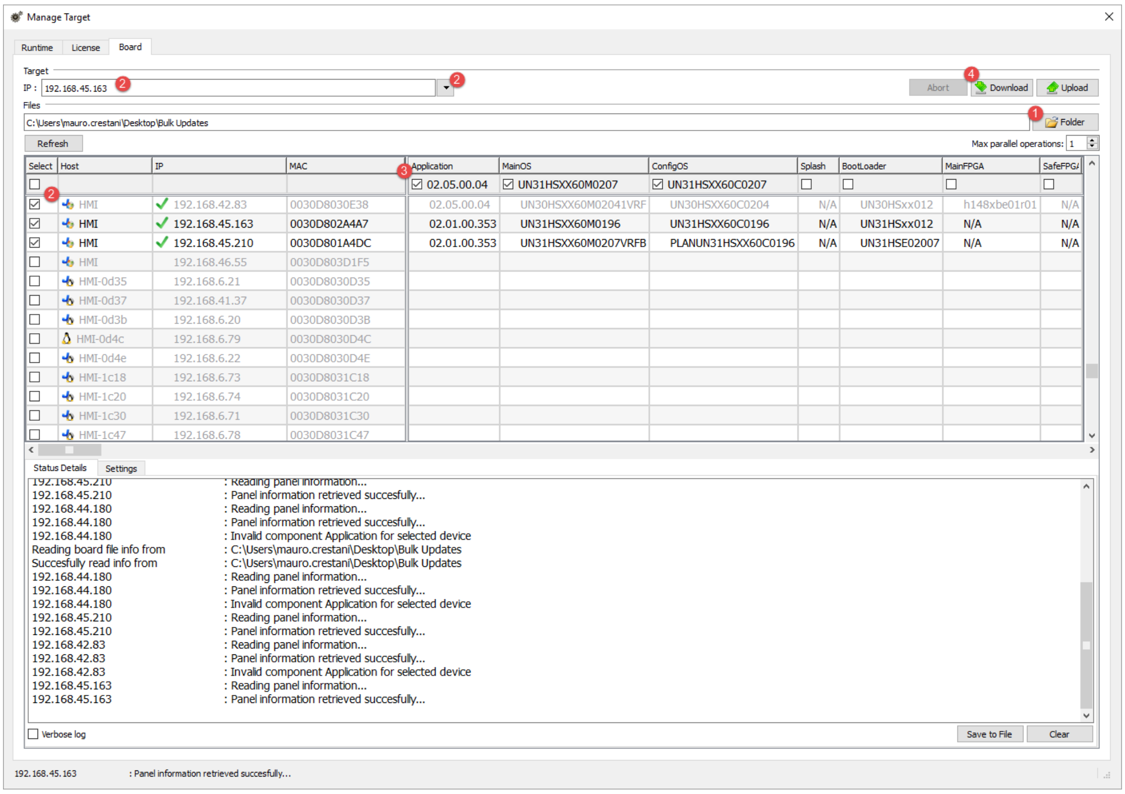
Note: Upgrade files are distributed upon request as a part of technical support activity.

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.

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***



IP : 192.168.45.163

C:\Users\mauro.crestani\Desktop\Bulk Updates

Max parallel operations: 1

Select	Host	IP	MAC	Application	MainOS	ConfigOS	Splash	BootLoader	MainFPGA	SafeFPG
☐				☑ 02.05.00.04	☑ UN31HSXX60M0207	☑ UN31HSXX60C0207	☐	☐	☐	☐
☑	HMI	✓ 192.168.42.83	0030D8030E38	02.05.00.04	UN30HSXX60M02041VRF	UN30HSXX60C0204	N/A	UN30HSxx012	h148xbe01r01	N/A
☑	HMI	✓ 192.168.45.163	0030D802A4A7	02.01.00.353	UN31HSXX60M0196	UN31HSXX60C0196	N/A	UN31HSxx012	N/A	N/A
☑	HMI	✓ 192.168.45.210	0030D801A4DC	02.01.00.353	UN31HSXX60M0207VRFB	PLANUN31HSXX60C0196	N/A	UN31HSE02007	N/A	N/A
☐	HMI	192.168.46.55	0030D803D1F5							
☐	HMI-0d35	192.168.6.21	0030D8030D35							
☐	HMI-0d37	192.168.41.37	0030D8030D37							
☐	HMI-0d3b	192.168.6.20	0030D8030D3B							
☐	HMI-0d4c	192.168.6.79	0030D8030D4C							
☐	HMI-0d4e	192.168.6.22	0030D8030D4E							
☐	HMI-1c18	192.168.6.73	0030D8031C18							
☐	HMI-1c20	192.168.6.74	0030D8031C20							
☐	HMI-1c30	192.168.6.71	0030D8031C30							
☐	HMI-1c47	192.168.6.78	0030D8031C47							

```
192.168.45.210                 : Reading panel information...
192.168.45.210                 : Panel information retrieved succesfully...
192.168.44.180                 : Reading panel information...
192.168.44.180                 : Panel information retrieved succesfully...
192.168.44.180                 : Invalid component Application for selected device
Reading board file info from   : C:\Users\mauro.crestani\Desktop\Bulk Updates
Succesfully read info from     : C:\Users\mauro.crestani\Desktop\Bulk Updates
192.168.44.180                 : Reading panel information...
192.168.44.180                 : Panel information retrieved succesfully...
192.168.44.180                 : Invalid component Application for selected device
192.168.45.210                 : Reading panel information...
192.168.45.210                 : Panel information retrieved succesfully...
192.168.42.83                  : Reading panel information...
192.168.42.83                  : Panel information retrieved succesfully...
192.168.42.83                  : Invalid component Application for selected device
192.168.45.163                 : Reading panel information...
192.168.45.163                 : Panel information retrieved succesfully...
```

☐ Verbose log

```
192.168.45.163          : Panel information retrieved succesfully...
```

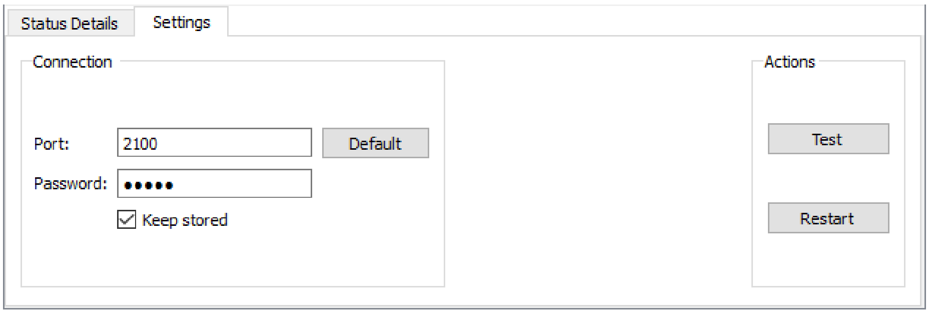
1. Select the folder that contains the files to download to the HMI device or where to upload files from the HMI device
2. Select one or more HMI device.
3. Select the components that you will download (or upload) to/from the devices
4. Start the Download to HMI or the Upload from the HMI operation

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.



Port: 2100

Password: ●●●●●

☑ Keep stored

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.

48 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.

A-B DF1

The A-B DF1 communication driver has been designed to connect HMI devices to a Allen-Bradley controllers through serial communication.

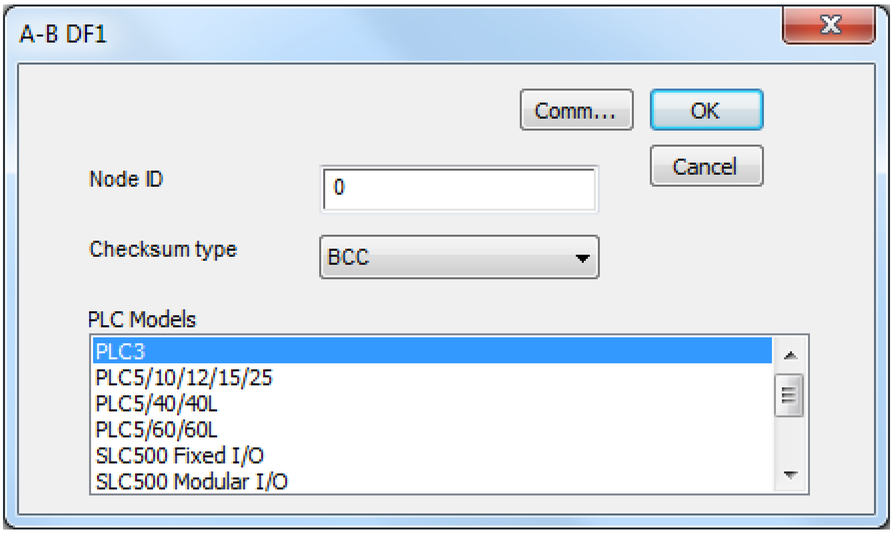
Protocol Editor Settings

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.

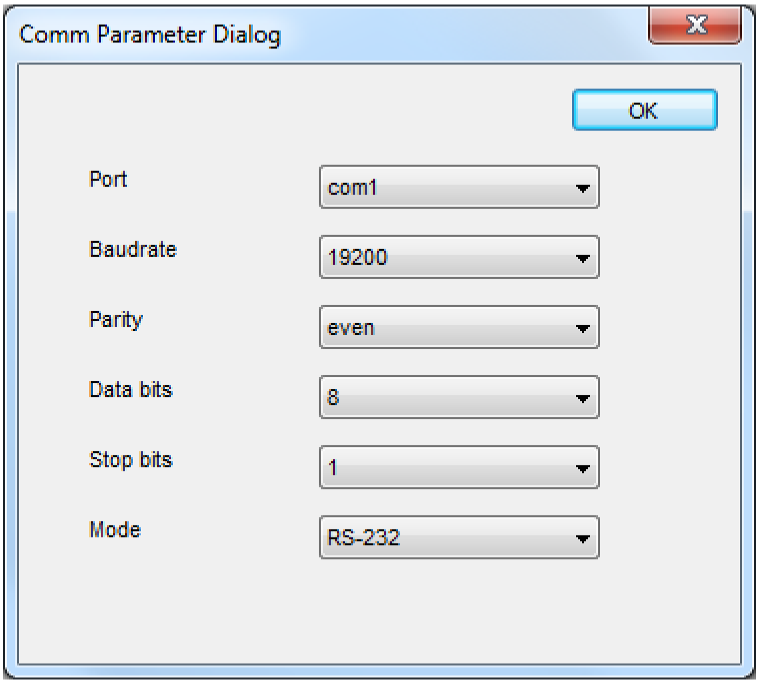

A-B DF1

Node ID: 0

Checksum type: BCC

PLC Models
- PLC3
- PLC5/10/12/15/25
- PLC5/40/40L
- PLC5/60/60L
- SLC500 Fixed I/O
- SLC500 Modular I/O

Element	Description
Node ID	Serial node associated to the PLC.
Checksum type	It can be **BCC** or **CRC**, depending on PLC settings.
PLC Models	PLC models available: • PLC3 • PLC5/10/12/15/25 • PLC5/40/40L • PLC5/60/60L • SLC500 Fixed I/O

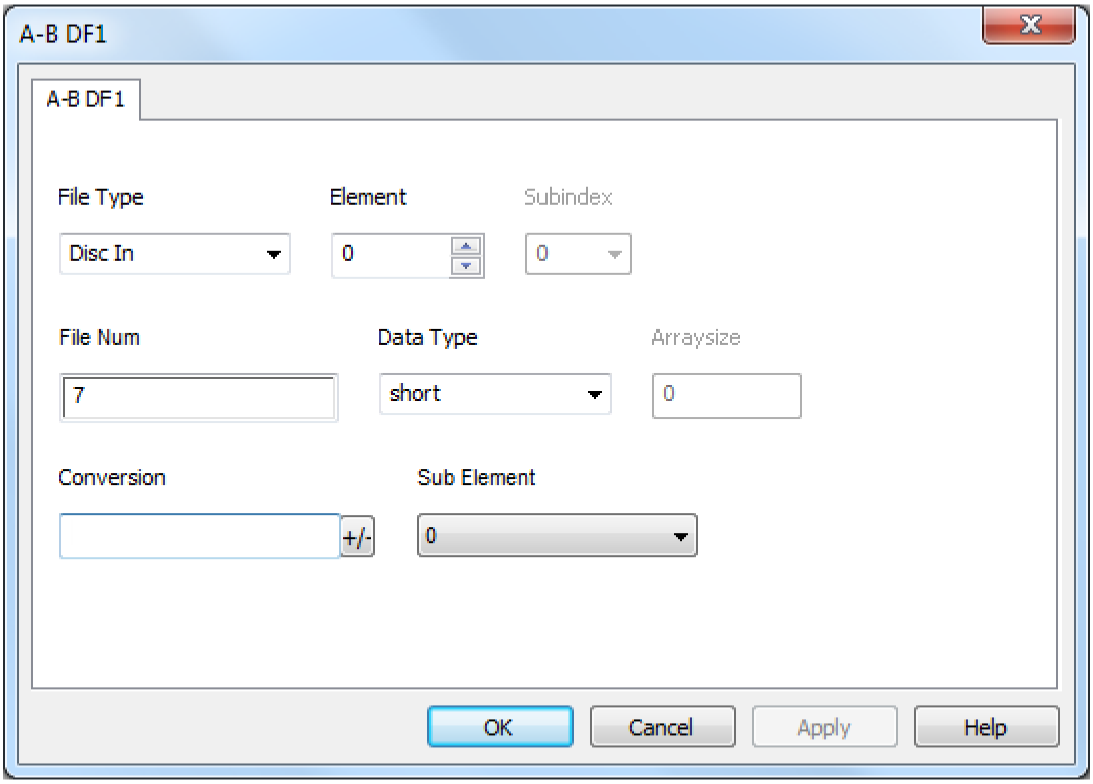
<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td></td><td>
<ul>
<li>SLC500 Modular I/O</li>
<li>Micrologix 1000</li>
<li>Micrologix 1500</li>
<li>Ultra5000</li>
</ul>
</td></tr>
<tr><td><b>Comm...</b></td><td>If clicked displays the communication parameters setup dialog.


Port: com1
Baudrate: 19200
Parity: even
Data bits: 8
Stop bits: 1
Mode: RS-232
<table>
<tr><th>Element</th><th>Parameter</th></tr>
<tr><td><b>Port</b></td><td>Serial port selection.
<ul>
<li><b>COM1</b>: device PLC port.</li>
<li><b>COM2</b>: computer/printer port on panels with 2 serial ports or optional Plug-In module plugged on Slot 1/2 for panels with 1 serial port on-board.</li>
<li><b>COM3:</b> optional Plug-In module plugged on Slot 3/4 for panels with 1 serial port on-board.</li>
</ul>
</td></tr>
<tr><td><b>Baudrate, Parity, Data Bits, Stop bits</b></td><td>Serial line parameters.</td></tr>
<tr><td><b>Mode</b></td><td>Serial port mode. Available modes:
<ul>
<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>
</table>
</td></tr>
</table>

Tag Editor Settings

In Tag Editor select the protocol **A-B DF1**.

Add a tag using [+] button. Tag setting can be defined using the following dialog:



A-B DF1

File Type: Disc In

Element: 0

Subindex: 0

File Num: 7

Data Type: short

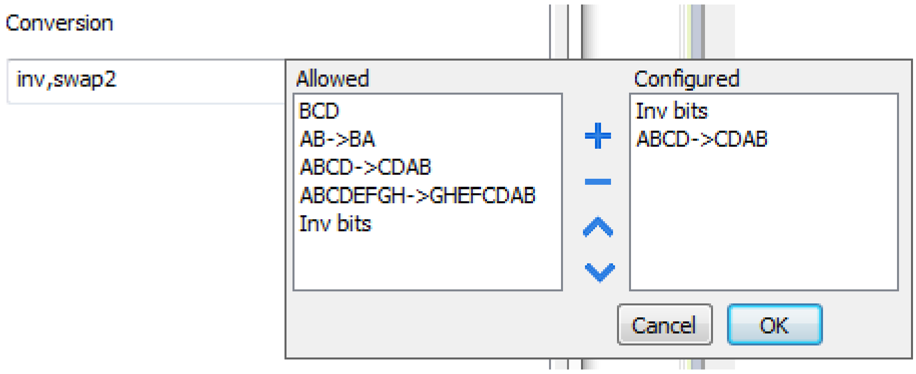
Arraysize: 0

Conversion:

Sub Element: 0

<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td><b>Memory Type</b></td><td>
<table>
<tr><th>Memory Type</th><th>Description</th></tr>
<tr><td><b>Disc Out</b></td><td>Discrete output value. <b>O</b> resource on PLC.</td></tr>
<tr><td><b>Disc In</b></td><td>Discrete input value. <b>I</b> resource on PLC.</td></tr>
<tr><td><b>Status</b></td><td>Status value. <b>S</b> resource on PLC.</td></tr>
<tr><td><b>Bit</b></td><td>Bit value. <b>B</b> resource on PLC.</td></tr>
<tr><td><b>Timer</b></td><td>Timer value. <b>T</b> resource on PLC.</td></tr>
<tr><td><b>Counter</b></td><td>Counter value. <b>C</b> resource on PLC.</td></tr>
<tr><td><b>Control</b></td><td>Control value. <b>R</b> resource on PLC.</td></tr>
<tr><td><b>Integer</b></td><td>Integer value. <b>N</b> resource on PLC.</td></tr>
<tr><td><b>Float</b></td><td>Float value. <b>F</b> resource on PLC.</td></tr>
</table>
</td></tr>
<tr><td><b>Element</b></td><td>Represents the line of the resource while monitoring PLC values.</td></tr>
<tr><td><b>Subindex</b></td><td>Represents the column of the resource while monitoring PLC values.</td></tr>
<tr><td><b>File Num</b></td><td>Instance of resource of the PLC.</td></tr>
<tr><td><b>Data Type</b></td><td>Available data types:
<ul>
<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>string</b></li>
<li><b>binary</b></li>
</ul>
See "Programming concepts" section in the main manual.
Note: To define arrays, select one of Data Type format followed by square brackets (byte[], short[]…).</td></tr>
<tr><td><b>Arraysize</b></td><td>
<ul>
<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>
</td></tr>
</table>

<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td></td><td>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.</td></tr>
<tr><td><b>Sub Element</b></td><td>Allows to point to specific part of a resource:
<ul>
<li>0 (entire resource)</li>
<li>PRE</li>
<li>ACC</li>
<li>LEN</li>
<li>POS</li>
</ul></td></tr>
<tr><td><b>Conversion</b></td><td>Conversion to be applied to the tag.



Conversion: inv,swap2

Allowed: BCD, AB->BA, ABCD->CDAB, ABCDEFGH->GHEFCDAB, Inv bits

Configured: Inv bits, ABCD->CDAB

Depending on data type selected, the list <b>Allowed</b> shows one or more conversion types.

<table>
<tr><th>Value</th><th>Description</th></tr>
<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 -> 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 -> CDAB</b></td><td><b>swap2</b>: Swap bytes in a word.</td></tr>
</table>
</td></tr>
</table>

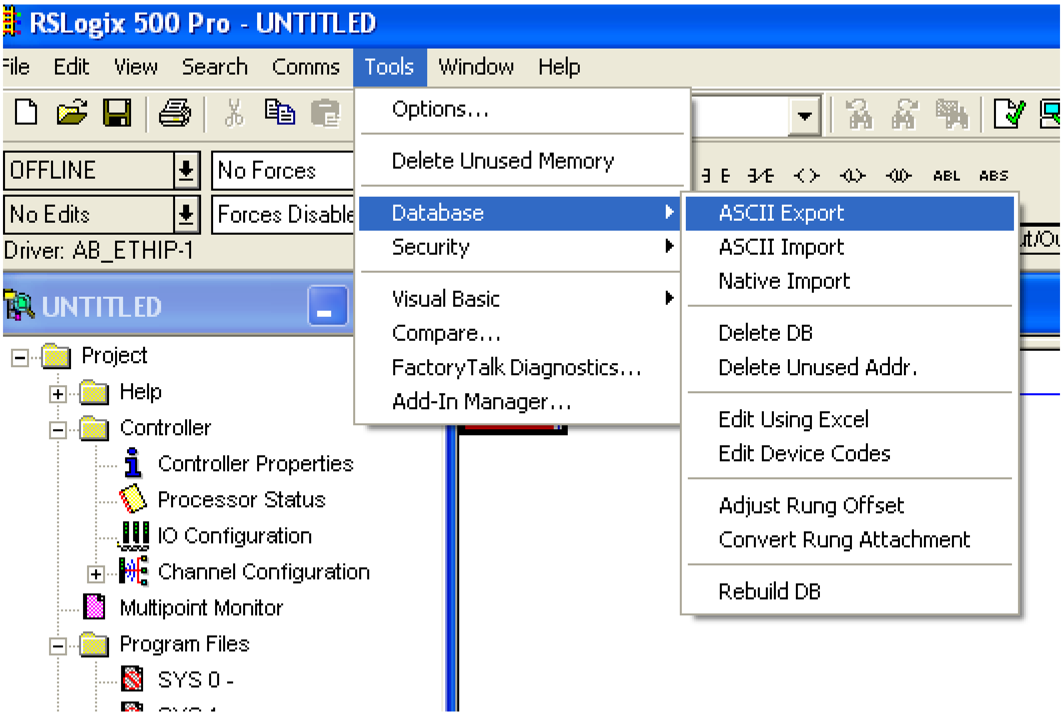
<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td></td><td><i>Example:</i>
9ACC → CC9A (in hexadecimal format)
39628 → 52378 (in decimal format)</td></tr>
<tr><td><b>ABCDEFGH -> GHEFCDAB</b></td><td><b>swap4</b>: Swap bytes in a double word.
<i>Example:</i>
32FCFF54 → 54FFFC32 (in hexadecimal format)
855441236 → 1426062386 (in decimal format)</td></tr>
<tr><td><b>ABC...NOP -> OPM...DAB</b></td><td><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)</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>
</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>
</td></tr>
</table>

Tag Import

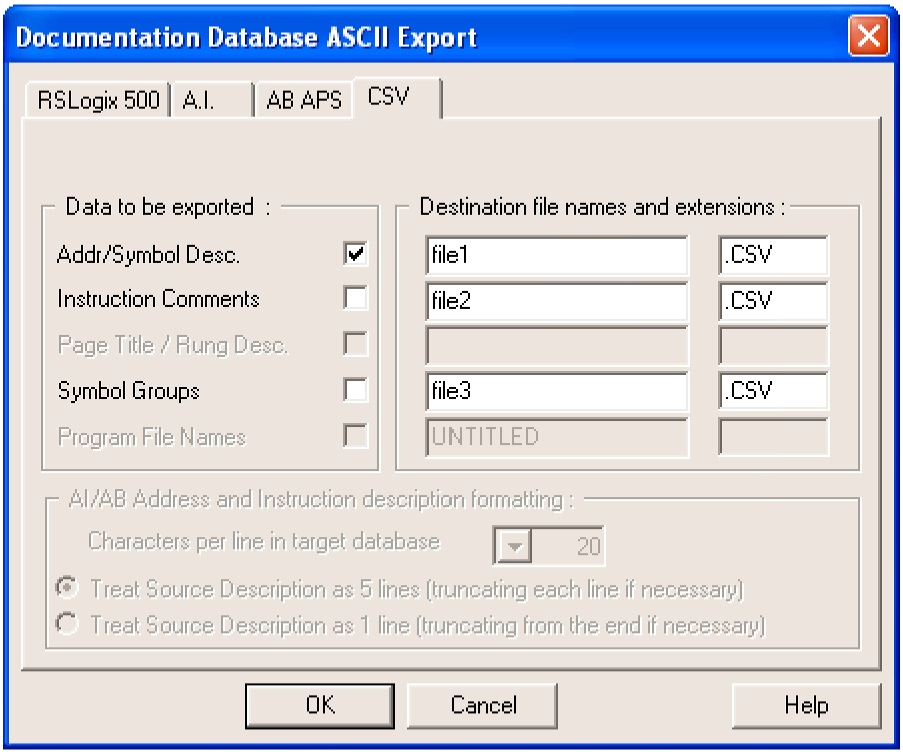
Exporting Tags from PLC

The A-B DF1 tag import filter accepts symbol files with extension “.csv” created by the Rockwell RSLogix 500.

To create the file select **Tool > Database > ASCII Export**



From **CSV** tab select the data to be exported and give a name to the output csv file.


Documentation Database ASCII Export

RSLogix 500 | A.I. | AB APS | CSV

Data to be exported :

Data to be exported		Destination file names and extensions	
Addr/Symbol Desc.	☑	file1	.CSV
Instruction Comments	☐	file2	.CSV
Page Title / Rung Desc.	☐		
Symbol Groups	☐	file3	.CSV
Program File Names	☐	UNTITLED	

AI/AB Address and Instruction description formatting :

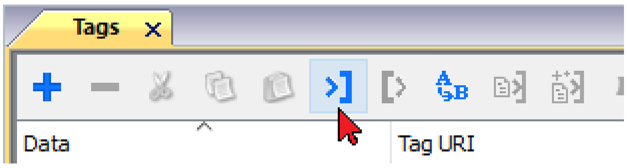
Characters per line in target database 20

● Treat Source Description as 5 lines (truncating each line if necessary)

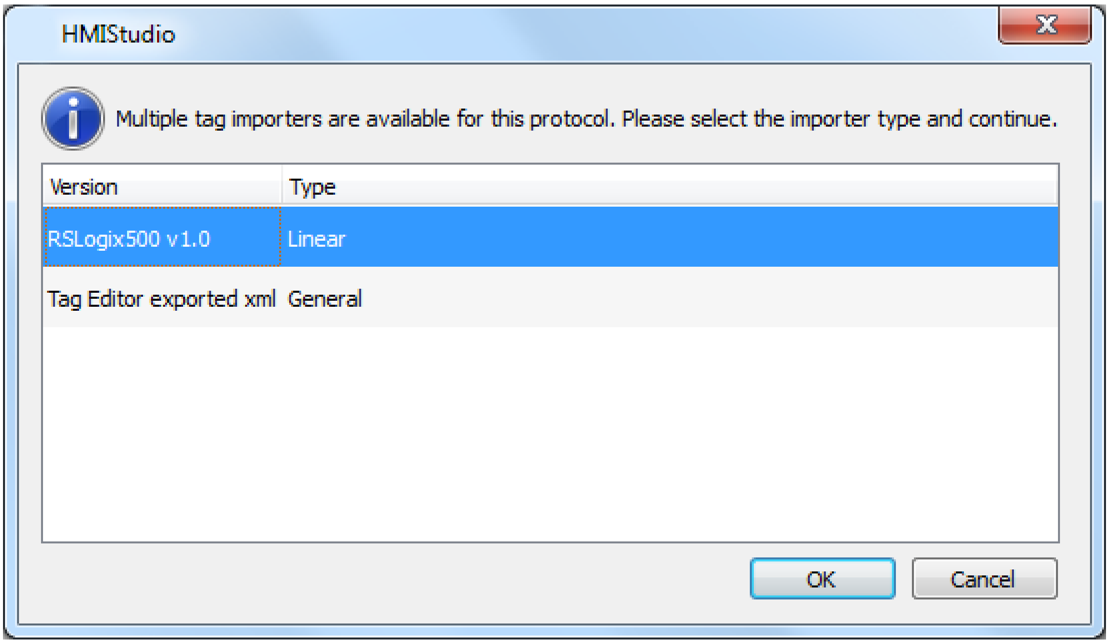
○ Treat Source Description as 1 line (truncating from the end if necessary)

Importing Tags in Tag Editor

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

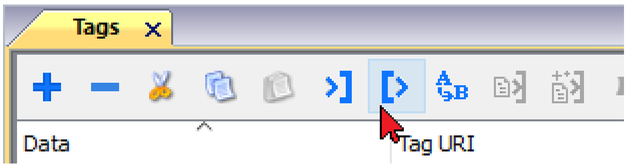


The following dialog shows which importer type can be selected.


HMIStudio

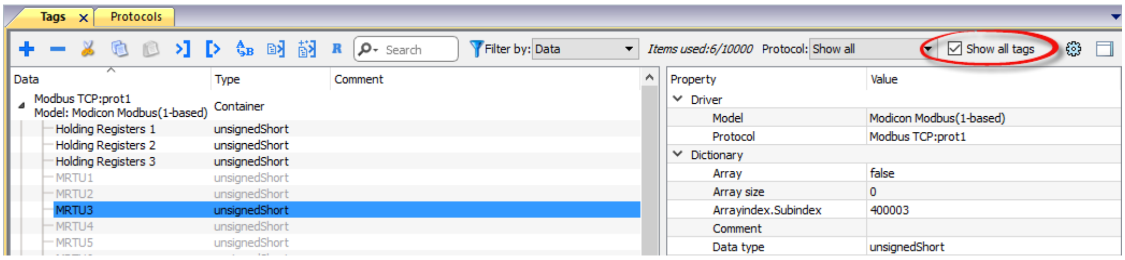
Multiple tag importers are available for this protocol. Please select the importer type and continue.

Version	Type
RSLogix500 v1.0	Linear
Tag Editor exported xml	General

Importer	Description
RSLogix500 v1.0 Linear	Requires an **.csv** file. 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. 

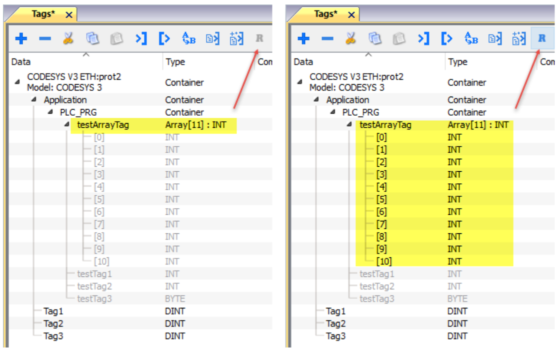
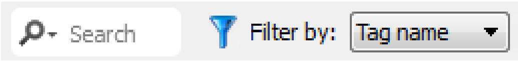
Once the importer has been selected, locate the symbol file and click **Open**.

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.



Data	Type	Comment
Modbus TCP:prot1 Model: Modicon Modbus(1-based)	Container	
Holding Registers 1	unsignedShort	
Holding Registers 2	unsignedShort	
Holding Registers 3	unsignedShort	
MRTU1	unsignedShort	
MRTU2	unsignedShort	
MRTU3	unsignedShort	
MRTU4	unsignedShort	
MRTU5	unsignedShort	

Property	Value
Driver	
Model	Modicon Modbus(1-based)
Protocol	Modbus TCP:prot1
Dictionary	
Array	false
Array size	0
Arrayindex.Subindex	400003
Comment	
Data type	unsignedShort

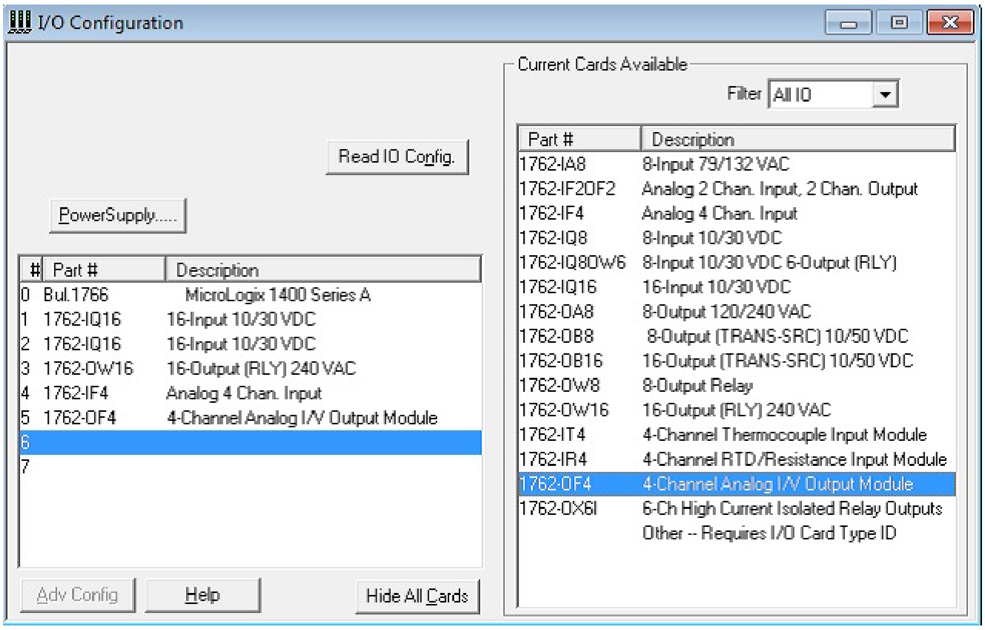
Toolbar item	Description
	Import Tag(s). Select tags to be imported and click on this icon to add tags from tag dictionary to the project
	Update Tag(s). Click on this icon to update the tags in the project, due a new dictionary import.
R	Check this box to import all sub-elements of a tag. Example of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

Logical I/O addressing

When addressing Allen Bradley I/O data, the panel uses logical addressing rather than physical addressing. While physical addressing refers to the element number as the slot number, logical addressing refers to the first element for the first I/O card of a specific file type.

AGI Creator addressing depends on the mapping of the PLC CPU memory and not on the slot number, therefore you should be careful when changing the configuration in order to avoid remapping.

Use the RSLogix 500 I/O Configuration tool layout of the PLC I/O to configure I/O as in the example.



I/O Configuration

Read IO Config.

PowerSupply.....

#	Part #	Description
0	Bul.1766	MicroLogix 1400 Series A
1	1762-IQ16	16-Input 10/30 VDC
2	1762-IQ16	16-Input 10/30 VDC
3	1762-OW16	16-Output (RLY) 240 VAC
4	1762-IF4	Analog 4 Chan. Input
5	1762-OF4	4-Channel Analog I/V Output Module
6		
7		

Adv Config | Help | Hide All Cards

Current Cards Available

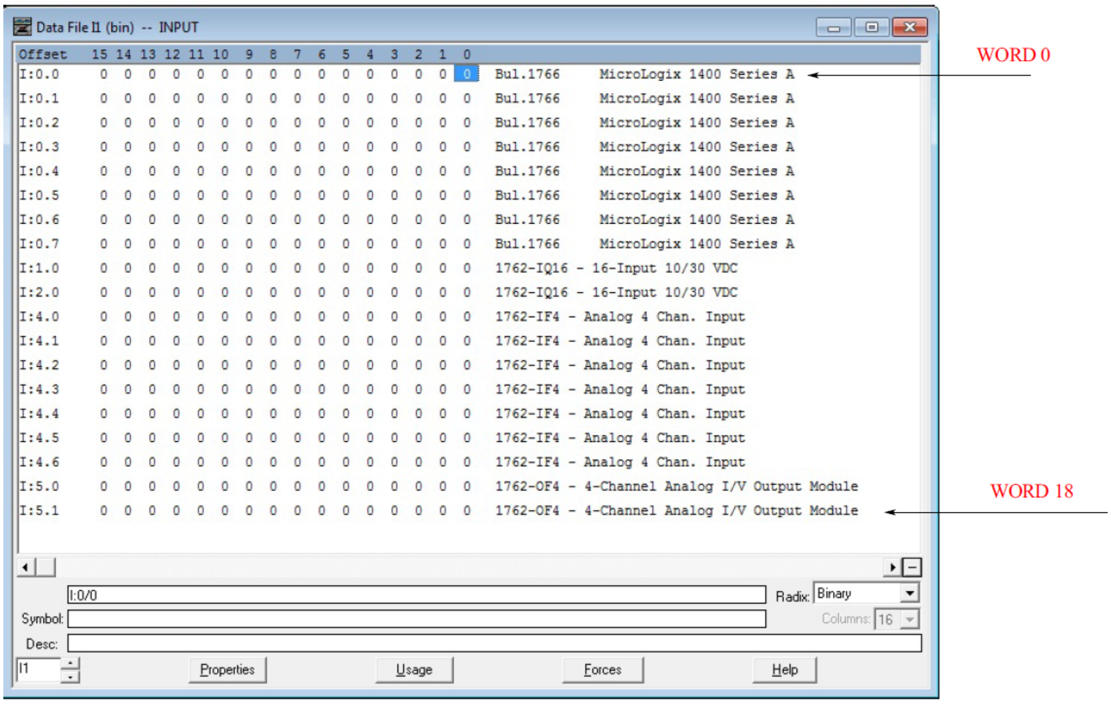
Filter: All IO

Part #	Description
1762-IA8	8-Input 79/132 VAC
1762-IF2OF2	Analog 2 Chan. Input, 2 Chan. Output
1762-IF4	Analog 4 Chan. Input
1762-IQ8	8-Input 10/30 VDC
1762-IQ8OW6	8-Input 10/30 VDC 6-Output (RLY)
1762-IQ16	16-Input 10/30 VDC
1762-OA8	8-Output 120/240 VAC
1762-OB8	8-Output (TRANS-SRC) 10/50 VDC
1762-OB16	16-Output (TRANS-SRC) 10/50 VDC
1762-OW8	8-Output Relay
1762-OW16	16-Output (RLY) 240 VAC
1762-IT4	4-Channel Thermocouple Input Module
1762-IR4	4-Channel RTD/Resistance Input Module
1762-OF4	4-Channel Analog I/V Output Module
1762-OX6I	6-Ch High Current Isolated Relay Outputs
	Other -- Requires I/O Card Type ID

Note: When using a module with a configurable I/O size (for example, Devicenet Scanner) make sure you configure it to the largest possible size or you will have to remap it if you need to allocate more space.

Use the Data File Browser to see how the PLC allocates memory.

This example shows how to configure the AGI Creator Tag for pointing to PLC resource O:1/19 (O1:1.1/3 in word terms).



Data File I1 (bin) -- INPUT

```
Offset  15 14 13 12 11 10  9  8  7  6  5  4  3  2  1  0
I:0.0    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   Bul.1766    MicroLogix 1400 Series A
I:0.1    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   Bul.1766    MicroLogix 1400 Series A
I:0.2    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   Bul.1766    MicroLogix 1400 Series A
I:0.3    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   Bul.1766    MicroLogix 1400 Series A
I:0.4    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   Bul.1766    MicroLogix 1400 Series A
I:0.5    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   Bul.1766    MicroLogix 1400 Series A
I:0.6    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   Bul.1766    MicroLogix 1400 Series A
I:0.7    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   Bul.1766    MicroLogix 1400 Series A
I:1.0    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   1762-IQ16 - 16-Input 10/30 VDC
I:2.0    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   1762-IQ16 - 16-Input 10/30 VDC
I:4.0    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   1762-IF4 - Analog 4 Chan. Input
I:4.1    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   1762-IF4 - Analog 4 Chan. Input
I:4.2    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   1762-IF4 - Analog 4 Chan. Input
I:4.3    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   1762-IF4 - Analog 4 Chan. Input
I:4.4    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   1762-IF4 - Analog 4 Chan. Input
I:4.5    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   1762-IF4 - Analog 4 Chan. Input
I:4.6    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   1762-IF4 - Analog 4 Chan. Input
I:5.0    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   1762-OF4 - 4-Channel Analog I/V Output Module
I:5.1    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   1762-OF4 - 4-Channel Analog I/V Output Module
```

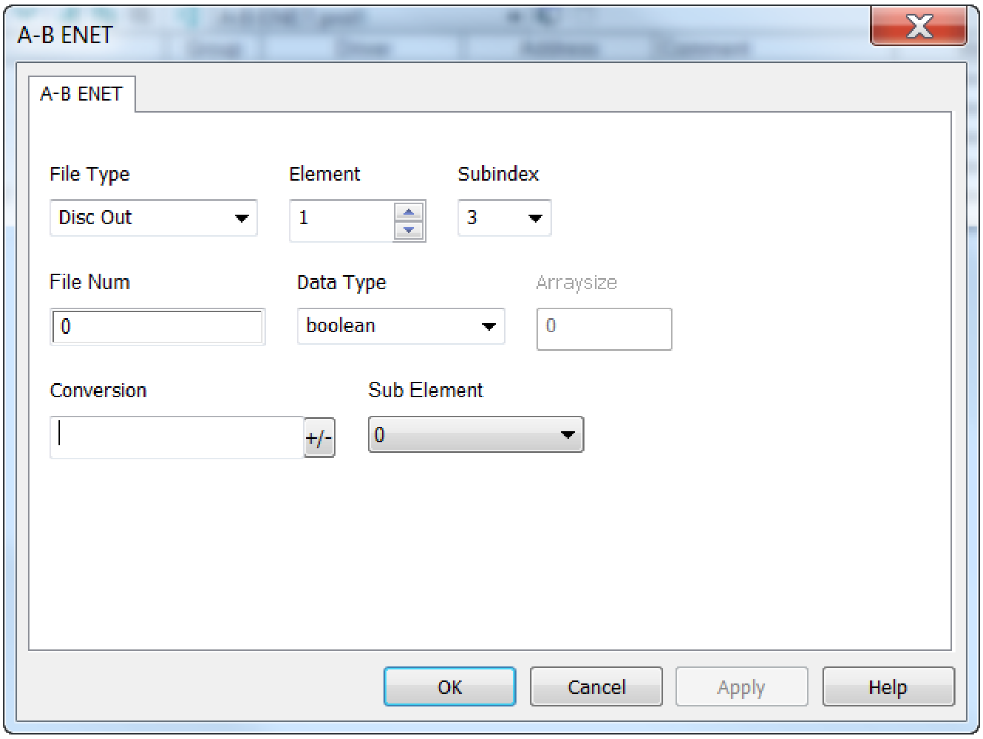
I:0/0   Radix: Binary

Symbol: Columns: 16

Desc:

I1   Properties   Usage   Forces   Help

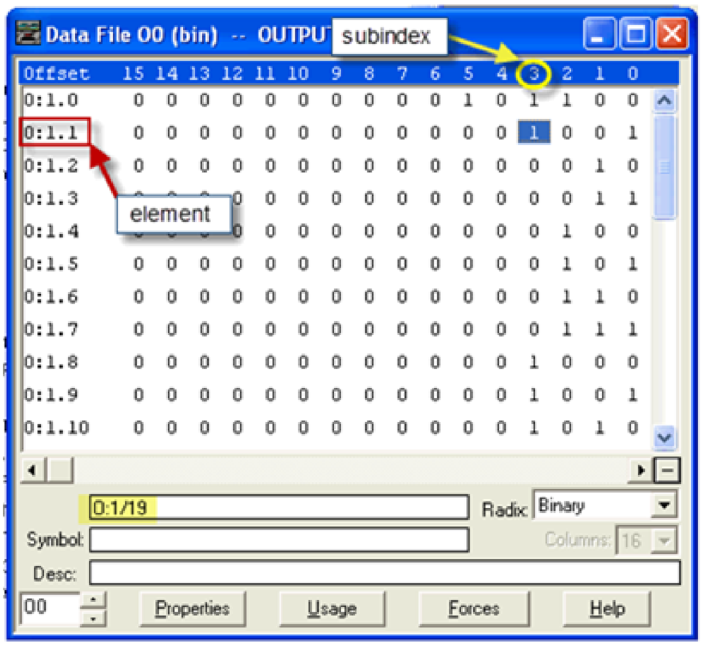
The following figure shows the AGI Creator Tag configuration.



A-B ENET

File Type: Disc Out
Element: 1
Subindex: 3
File Num: 0
Data Type: boolean
Arraysize: 0
Conversion:
Sub Element: 0

The AGI Creator Tag configured in the example above points on the element shown in the following figure.



Data File O0 (bin) -- OUTPU

Offset	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0:1.0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0
0:1.1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1
0:1.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0:1.3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
0:1.4	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0:1.5	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
0:1.6	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
0:1.7	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
0:1.8	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
0:1.9	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1
0:1.10	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0

0:1/19
Radix: Binary
Symbol:
Columns: 16
Desc:
00

Examples

I:0/19 (I1:0.1/3 in word terms) – 20th Input on CPU

Parameter	Setting
File Type	Disc In
File Num	1
Data Type	Boolean

In the Data File Browser, word 0.1 is Word 1:

Element	1
Sub Index	3

I:1/15 (I1:1.0/15 in word terms) - Last Input on Slot 1 Input Card

Parameter	Setting
File Type	Disc In
File Num	1
Data Type	Boolean

In the Data File Browser, word 1.0 is Word 8:

Element	8
Sub Index	15

I:4.0 (I1:4.0 in word terms) - First Analog Input

Parameter	Setting
File Type	Disc In
File Num	1
Data Type	Short

In the Data File Browser, word 4.0 is Word 10:

Element	10
Sub Index	-

A-B DH-485

The A-B DH-485 communication driver has been designed to connect HMI devices to a Allen-Bradley controllers through serial communication.

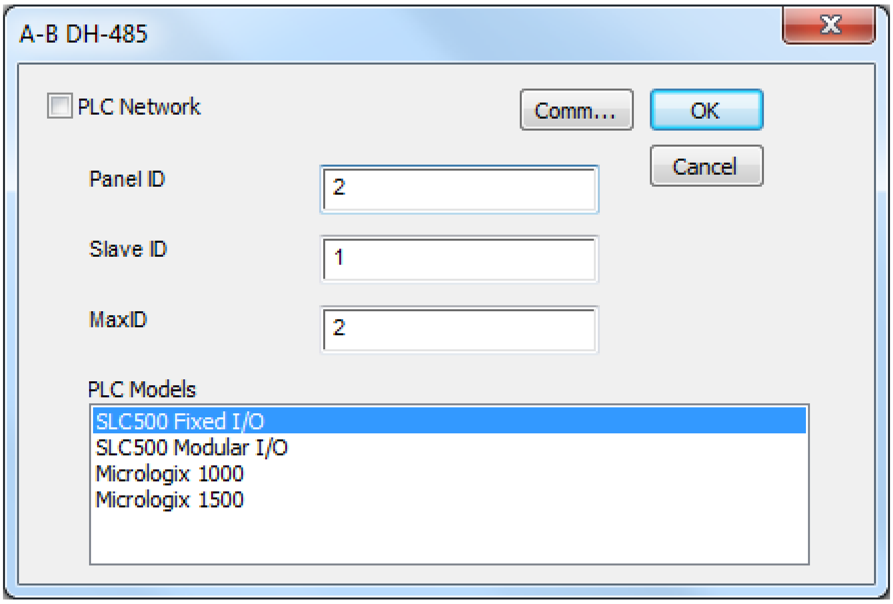
Protocol Editor Settings

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.


A-B DH-485

□ PLC Network

Panel ID: 2

Slave ID: 1

MaxID: 2

PLC Models

- SLC500 Fixed I/O
- SLC500 Modular I/O
- Micrologix 1000
- Micrologix 1500

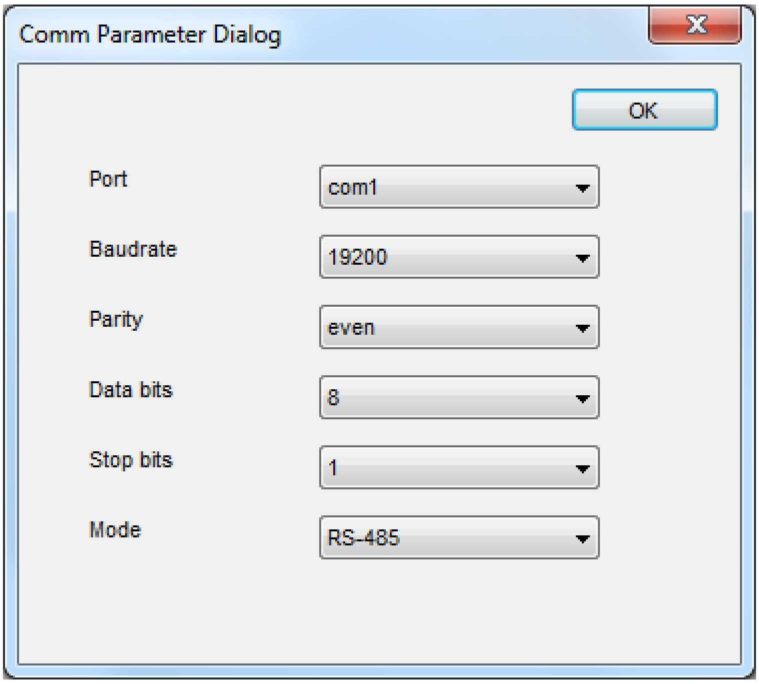
Element	Description
Panel ID	Serial node associated to the HMI.
Slave ID	Serial node associated to the PLC.
MaxID	Represent the maximum ID available in the serial network.
PLC Models	PLC models available: • SLC500 Fixed I/O • SLC500 Modular I/O

<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td></td><td>

- Micrologix 1000
- Micrologix 1500

</td></tr>
<tr><td><b>Comm...</b></td><td>

If clicked displays the communication parameters setup dialog.



Port: com1
Baudrate: 19200
Parity: even
Data bits: 8
Stop bits: 1
Mode: RS-485

<table>
<tr><th>Element</th><th>Parameter</th></tr>
<tr><td><b>Port</b></td><td>

Serial port selection.

- **COM1**: device PLC port.
- **COM2**: computer/printer port on panels with 2 serial ports or optional Plug-In module plugged on Slot 1/2 for panels with 1 serial port on-board.
- **COM3:** optional Plug-In module plugged on Slot 3/4 for panels with 1 serial port on-board.

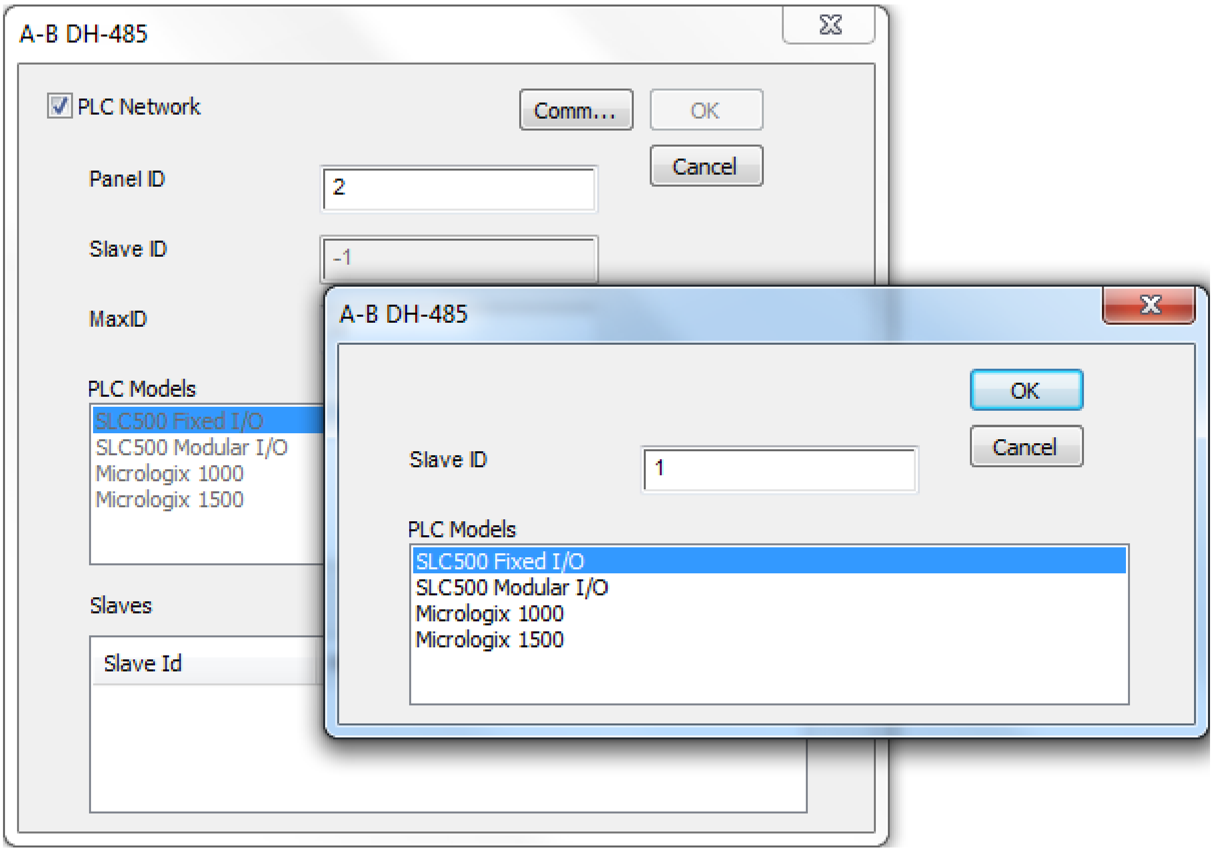
</td></tr>
<tr><td><b>Baudrate, Parity, Data Bits, Stop bits</b></td><td>Serial line parameters.</td></tr>
<tr><td><b>Mode</b></td><td>

Serial port mode. Available modes:

- **RS-232**.
- **RS-485** (2 wires).
- **RS-422** (4 wires).

</td></tr>
</table>

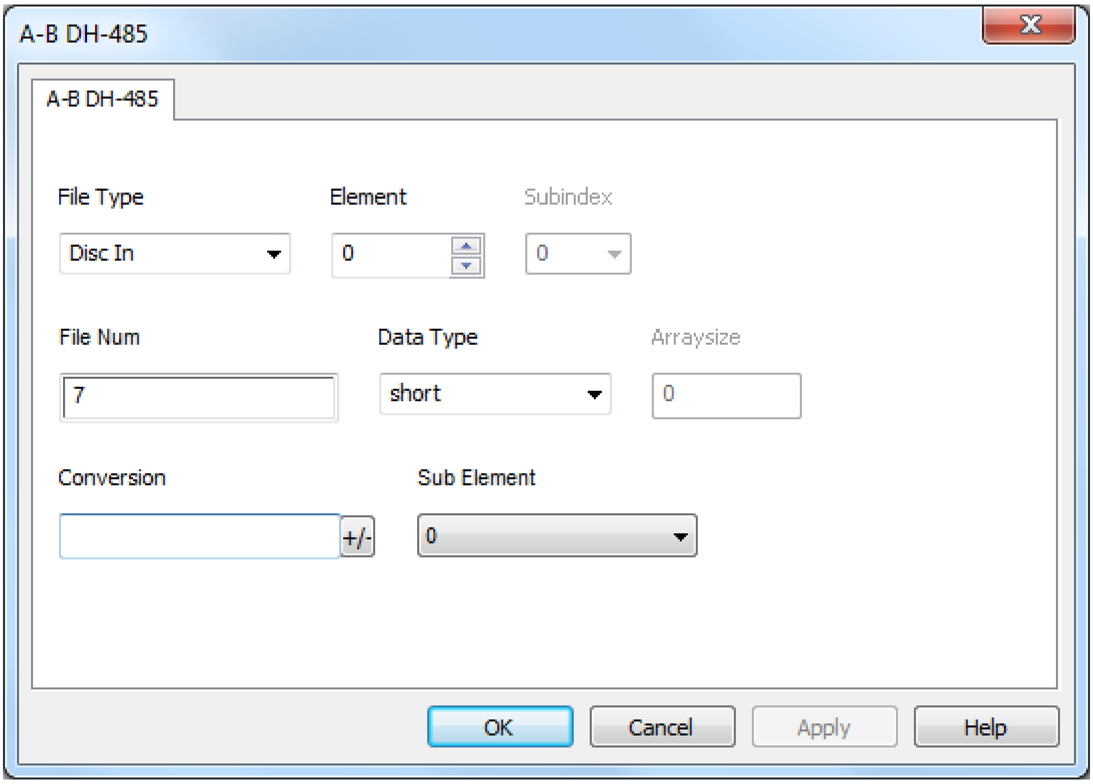
</td></tr>
<tr><td><b>PLC Network</b></td><td>IP address for all controllers in multiple connections. <b>PLC Network</b> must be selected to enable multiple connections.</td></tr>
</table>

Element	Description
	 A-B DH-485 ☑ PLC Network Panel ID: 2 Slave ID: -1 MaxID PLC Models: SLC500 Fixed I/O (selected), SLC500 Modular I/O, Micrologix 1000, Micrologix 1500 Slaves: Slave Id A-B DH-485 Slave ID: 1 PLC Models: SLC500 Fixed I/O (selected), SLC500 Modular I/O, Micrologix 1000, Micrologix 1500 Comm... OK Cancel

Tag Editor Settings

In Tag Editor select the protocol **A-B DH-485**.

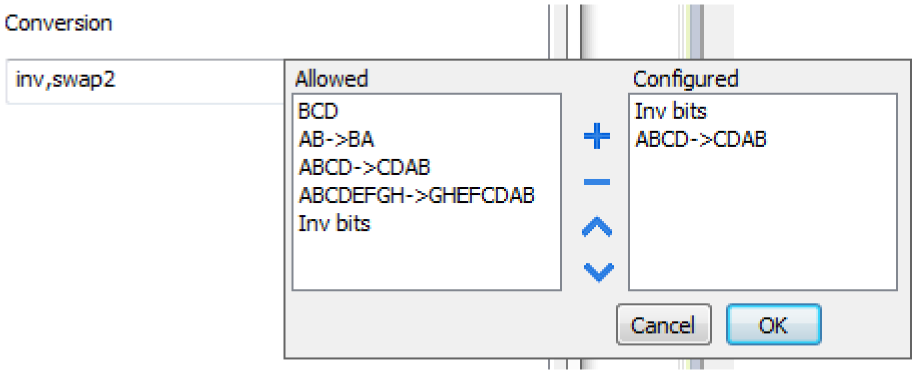
Add a tag using [+] button. Tag setting can be defined using the following dialog:



<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td><b>Memory Type</b></td><td>
<table>
<tr><th>Memory Type</th><th>Description</th></tr>
<tr><td><b>Disc Out</b></td><td>Discrete output value. <b>O</b> resource on PLC.</td></tr>
<tr><td><b>Disc In</b></td><td>Discrete input value. <b>I</b> resource on PLC.</td></tr>
<tr><td><b>Status</b></td><td>Status value. <b>S</b> resource on PLC.</td></tr>
<tr><td><b>Bit</b></td><td>Bit value. <b>B</b> resource on PLC.</td></tr>
<tr><td><b>Timer</b></td><td>Timer value. <b>T</b> resource on PLC.</td></tr>
<tr><td><b>Counter</b></td><td>Counter value. <b>C</b> resource on PLC.</td></tr>
<tr><td><b>Control</b></td><td>Control value. <b>R</b> resource on PLC.</td></tr>
<tr><td><b>Integer</b></td><td>Integer value. <b>N</b> resource on PLC.</td></tr>
<tr><td><b>Float</b></td><td>Float value. <b>F</b> resource on PLC.</td></tr>
<tr><td><b>String</b></td><td>String value. <b>STR</b> resource on PLC.</td></tr>
</table>
</td></tr>
<tr><td><b>Element</b></td><td>Represents the line of the resource while monitoring PLC values.</td></tr>
<tr><td><b>Subindex</b></td><td>Represents the column of the resource while monitoring PLC values.</td></tr>
<tr><td><b>File Num</b></td><td>Instance of resource of the PLC.</td></tr>
</table>

Element	Description
Data Type	Available data types: - **boolean** - **byte** - **short** - **int** - **unsignedByte** - **unsignedShort** - **unsignedInt** - **float** - **double** - **string** - **binary** See "Programming concepts" section in the main manual. Note: To define arrays, select one of Data Type format followed by square brackets (byte[], short[]…).
Arraysize	- 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.

<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td><b>Sub Element</b></td><td>Allows to point to specific part of a resource:<ul><li>0 (entire resource)</li><li>PRE</li><li>ACC</li><li>LEN</li><li>POS</li></ul></td></tr>
<tr><td><b>Conversion</b></td><td>Conversion to be applied to the tag.



Conversion: inv,swap2

Allowed
BCD
AB->BA
ABCD->CDAB
ABCDEFGH->GHEFCDAB
Inv bits

Configured
Inv bits
ABCD->CDAB

Depending on data type selected, the list <b>Allowed</b> shows one or more conversion types.

<table>
<tr><th>Value</th><th>Description</th></tr>
<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 -> 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 -> 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 -</b></td><td><b>swap4</b>: Swap bytes in a double word.</td></tr>
</table>
</td></tr>
</table>

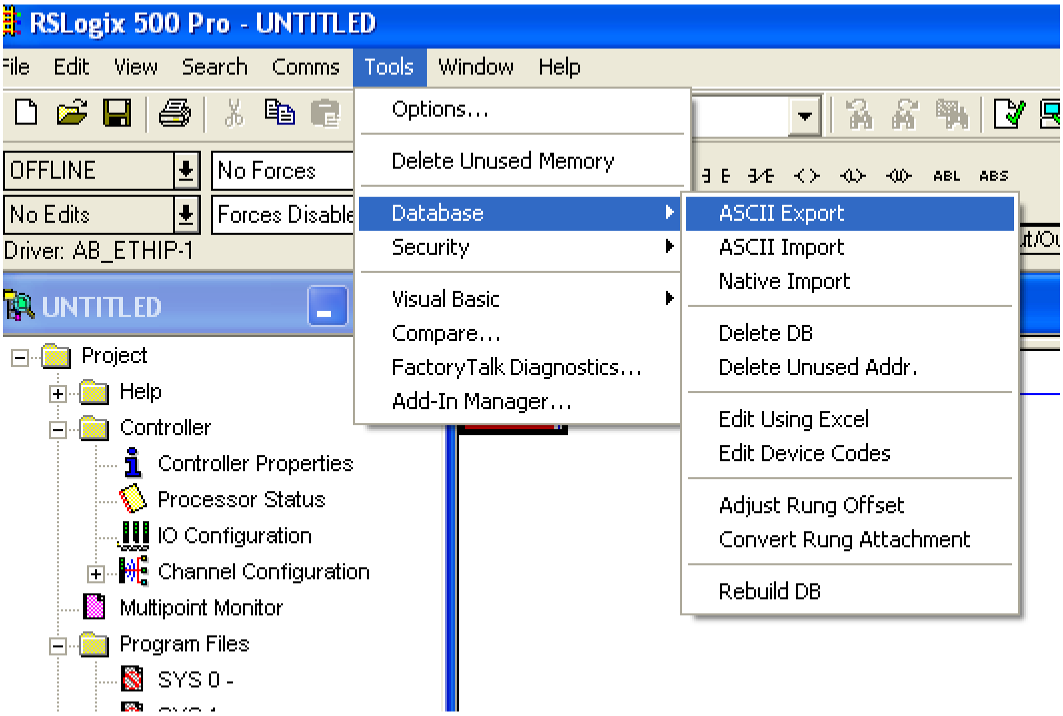
<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td><b>&gt; GHEFCDAB</b></td><td><i>Example:</i>
32FCFF54 → 54FFFC32 (in hexadecimal format)
855441236 → 1426062386 (in decimal format)</td></tr>
<tr><td><b>ABC...NOP -> OPM...DAB</b></td><td><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)</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>
</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>
</td></tr>
</table>

Tag Import

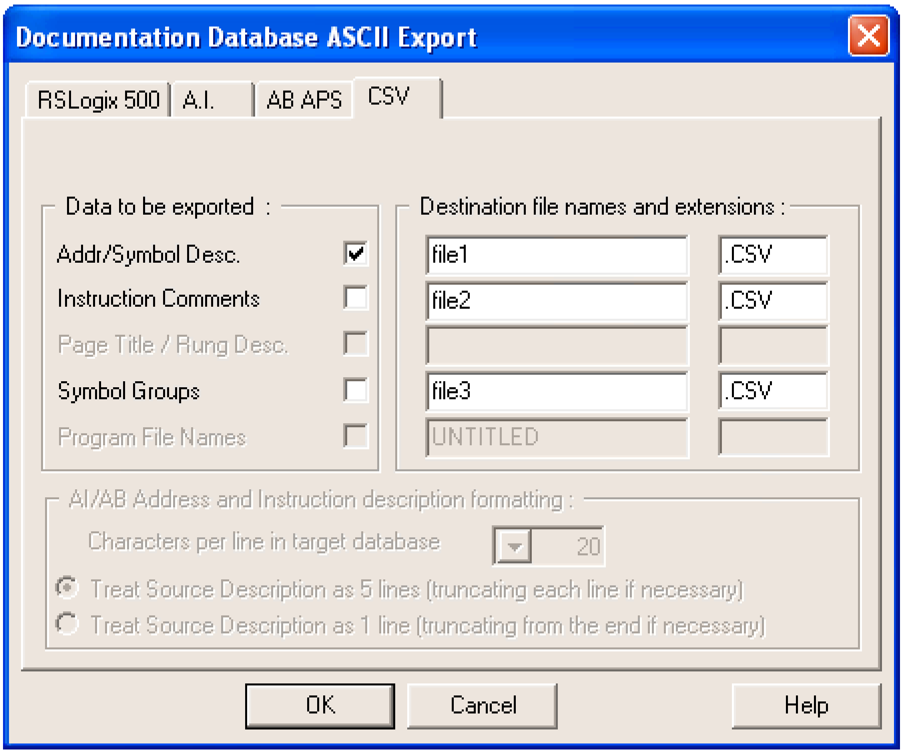
Exporting Tags from PLC

The A-B DF1 tag import filter accepts symbol files with extension “.csv” created by the Rockwell RSLogix 500.

To create the file select **Tool > Database > ASCII Export**



From **CSV** tab select the data to be exported and give a name to the output csv file.



Data to be exported :

- Addr/Symbol Desc. ☑
- Instruction Comments ☐
- Page Title / Rung Desc. ☐
- Symbol Groups ☐
- Program File Names ☐

Destination file names and extensions :

file1	.CSV
file2	.CSV
file3	.CSV
UNTITLED	

AI/AB Address and Instruction description formatting :

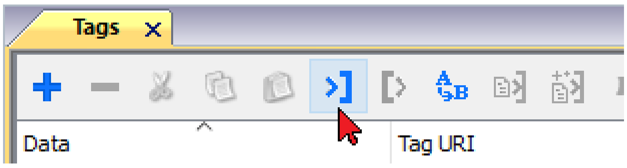
Characters per line in target database 20

● Treat Source Description as 5 lines (truncating each line if necessary)

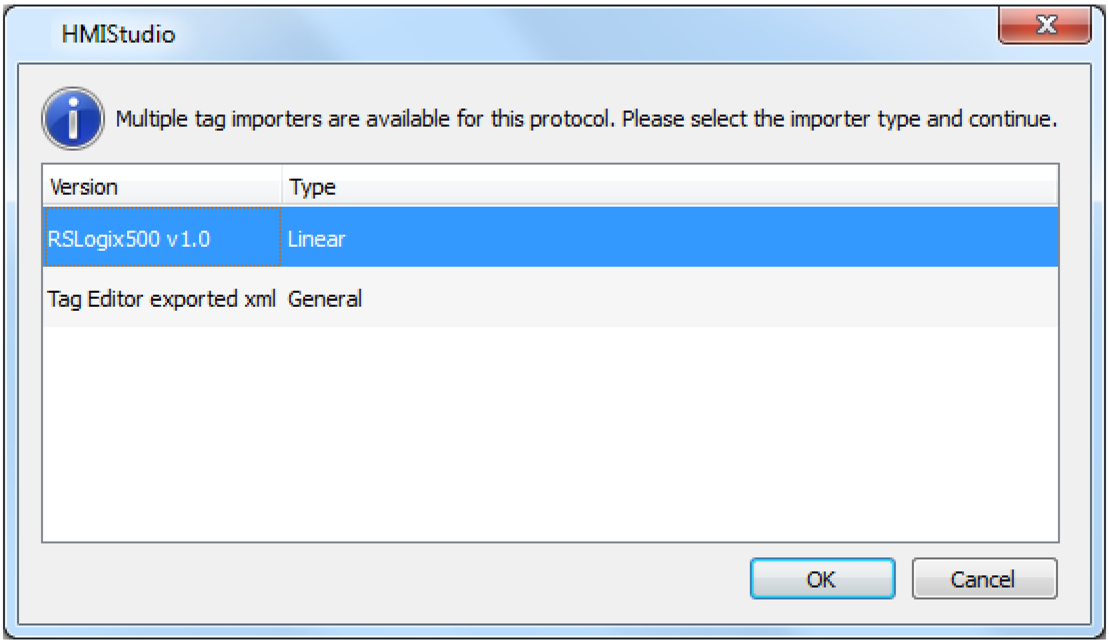
○ Treat Source Description as 1 line (truncating from the end if necessary)

Importing Tags in Tag Editor

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

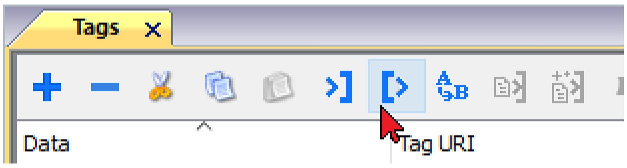


The following dialog shows which importer type can be selected.



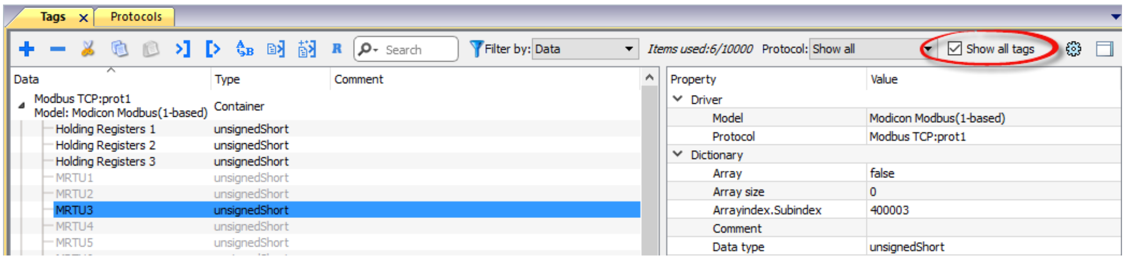
Multiple tag importers are available for this protocol. Please select the importer type and continue.

Version	Type
RSLogix500 v1.0	Linear
Tag Editor exported xml	General

Importer	Description
RSLogix500 v1.0 Linear	Requires an **.csv** file. 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. 

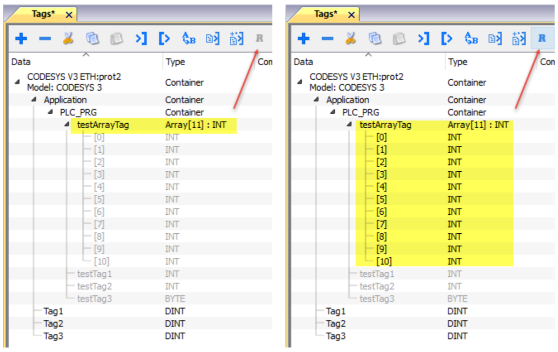
Once the importer has been selected, locate the symbol file and click **Open**.

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.



Data	Type	Comment
Modbus TCP:prot1 Model: Modicon Modbus(1-based)	Container	
Holding Registers 1	unsignedShort	
Holding Registers 2	unsignedShort	
Holding Registers 3	unsignedShort	
MRTU1	unsignedShort	
MRTU2	unsignedShort	
MRTU3	unsignedShort	
MRTU4	unsignedShort	
MRTU5	unsignedShort	

Property	Value
Driver	
Model	Modicon Modbus(1-based)
Protocol	Modbus TCP:prot1
Dictionary	
Array	false
Array size	0
Arrayindex.Subindex	400003
Comment	
Data type	unsignedShort

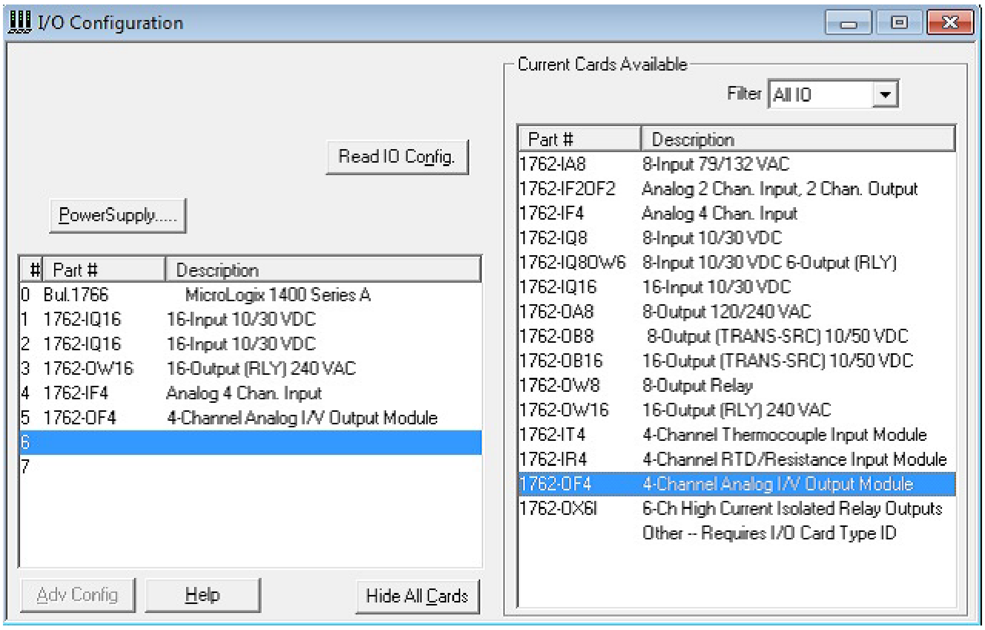
Toolbar item	Description
	Import Tag(s). Select tags to be imported and click on this icon to add tags from tag dictionary to the project
	Update Tag(s). Click on this icon to update the tags in the project, due a new dictionary import.
R	Check this box to import all sub-elements of a tag. Example of both checked and unchecked result:  Tags* Data / Type / Con CODESYS V3 ETH:prot2 Model: CODESYS 3 — Container Application — Container PLC_PRG — Container testArrayTag — Array[11] : INT [0] — INT [1] — INT [2] — INT [3] — INT [4] — INT [5] — INT [6] — INT [7] — INT [8] — INT [9] — INT [10] — INT testTag1 — INT testTag2 — INT testTag3 — BYTE Tag1 — DINT Tag2 — DINT Tag3 — DINT
 Search Filter by: Tag name	Searches tags in the dictionary basing on filter combo-box item selected.

Logical I/O addressing

When addressing Allen Bradley I/O data, the panel uses logical addressing rather than physical addressing. While physical addressing refers to the element number as the slot number, logical addressing refers to the first element for the first I/O card of a specific file type.

AGI Creator addressing depends on the mapping of the PLC CPU memory and not on the slot number, therefore you should be careful when changing the configuration in order to avoid remapping.

Use the RSLogix 500 I/O Configuration tool layout of the PLC I/O to configure I/O as in the example.



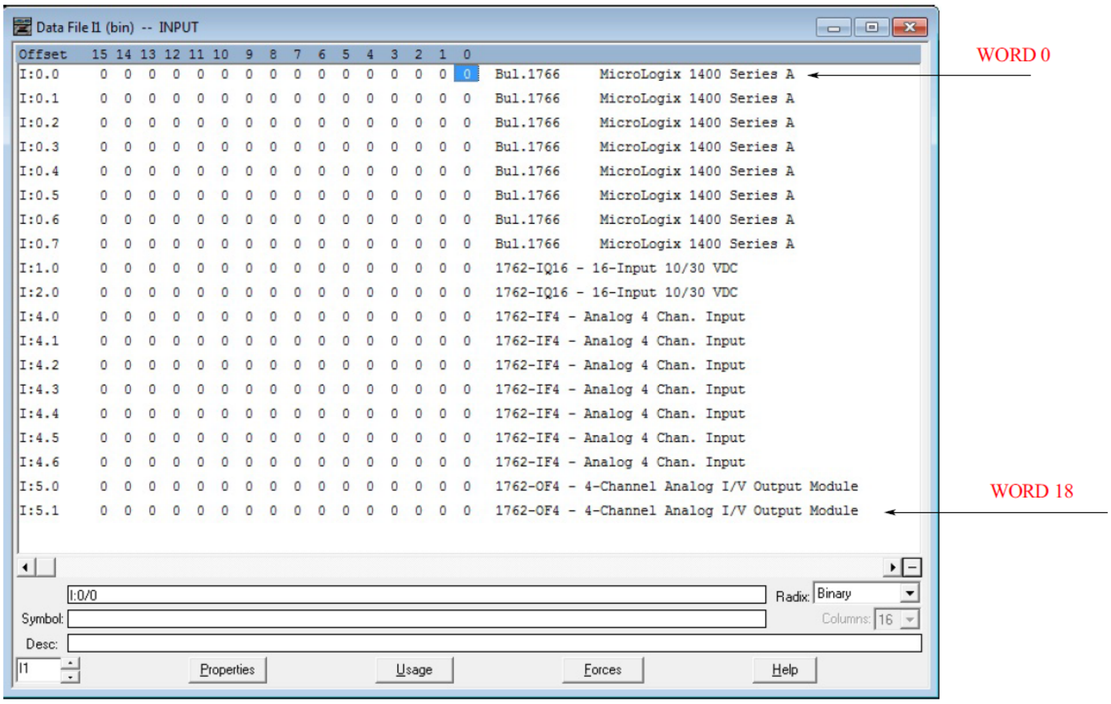
#	Part #	Description
0	Bul.1766	MicroLogix 1400 Series A
1	1762-IQ16	16-Input 10/30 VDC
2	1762-IQ16	16-Input 10/30 VDC
3	1762-OW16	16-Output (RLY) 240 VAC
4	1762-IF4	Analog 4 Chan. Input
5	1762-OF4	4-Channel Analog I/V Output Module
6		
7		

Part #	Description
1762-IA8	8-Input 79/132 VAC
1762-IF2OF2	Analog 2 Chan. Input, 2 Chan. Output
1762-IF4	Analog 4 Chan. Input
1762-IQ8	8-Input 10/30 VDC
1762-IQ8OW6	8-Input 10/30 VDC 6-Output (RLY)
1762-IQ16	16-Input 10/30 VDC
1762-OA8	8-Output 120/240 VAC
1762-OB8	8-Output (TRANS-SRC) 10/50 VDC
1762-OB16	16-Output (TRANS-SRC) 10/50 VDC
1762-OW8	8-Output Relay
1762-OW16	16-Output (RLY) 240 VAC
1762-IT4	4-Channel Thermocouple Input Module
1762-IR4	4-Channel RTD/Resistance Input Module
1762-OF4	4-Channel Analog I/V Output Module
1762-OX6I	6-Ch High Current Isolated Relay Outputs
	Other -- Requires I/O Card Type ID

Note: When using a module with a configurable I/O size (for example, Devicenet Scanner) make sure you configure it to the largest possible size or you will have to remap it if you need to allocate more space.

Use the Data File Browser to see how the PLC allocates memory.

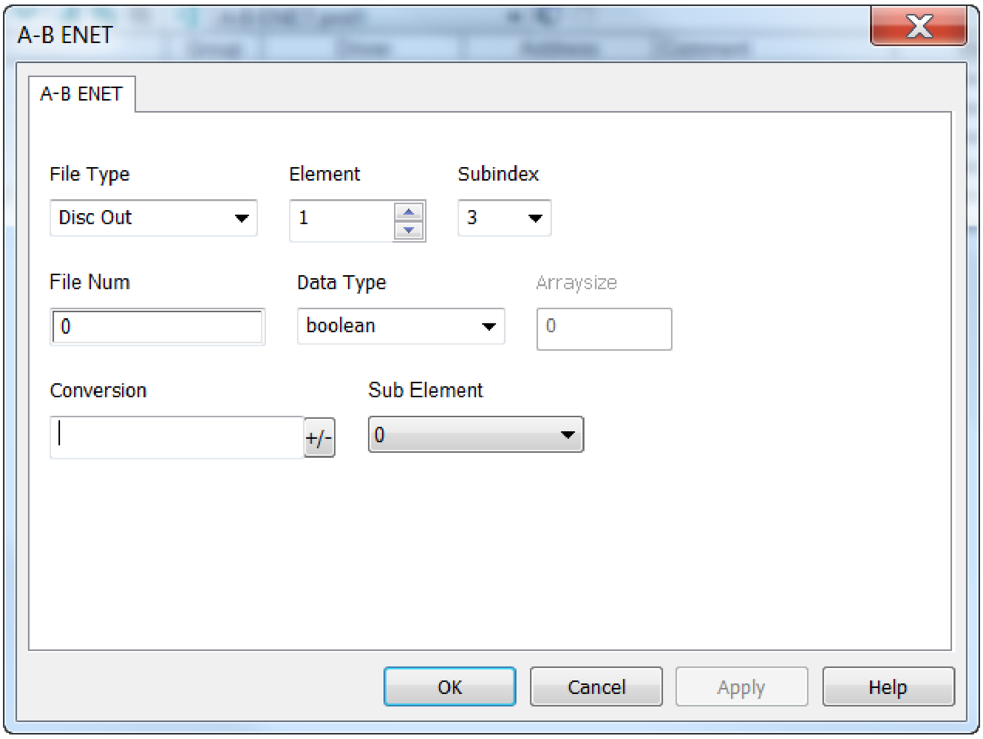
This example shows how to configure the AGI Creator Tag for pointing to PLC resource O:1/19 (O1:1.1/3 in word terms).



```
Offset   15 14 13 12 11 10  9  8  7  6  5  4  3  2  1  0
I:0.0     0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   Bul.1766    MicroLogix 1400 Series A
I:0.1     0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   Bul.1766    MicroLogix 1400 Series A
I:0.2     0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   Bul.1766    MicroLogix 1400 Series A
I:0.3     0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   Bul.1766    MicroLogix 1400 Series A
I:0.4     0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   Bul.1766    MicroLogix 1400 Series A
I:0.5     0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   Bul.1766    MicroLogix 1400 Series A
I:0.6     0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   Bul.1766    MicroLogix 1400 Series A
I:0.7     0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   Bul.1766    MicroLogix 1400 Series A
I:1.0     0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   1762-IQ16 - 16-Input 10/30 VDC
I:2.0     0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   1762-IQ16 - 16-Input 10/30 VDC
I:4.0     0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   1762-IF4 - Analog 4 Chan. Input
I:4.1     0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   1762-IF4 - Analog 4 Chan. Input
I:4.2     0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   1762-IF4 - Analog 4 Chan. Input
I:4.3     0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   1762-IF4 - Analog 4 Chan. Input
I:4.4     0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   1762-IF4 - Analog 4 Chan. Input
I:4.5     0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   1762-IF4 - Analog 4 Chan. Input
I:4.6     0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   1762-IF4 - Analog 4 Chan. Input
I:5.0     0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   1762-OF4 - 4-Channel Analog I/V Output Module
I:5.1     0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   1762-OF4 - 4-Channel Analog I/V Output Module
```

I:0/0

The following figure shows the AGI Creator Tag configuration.



A-B ENET

A-B ENET

File Type: Disc Out

Element: 1

Subindex: 3

File Num: 0

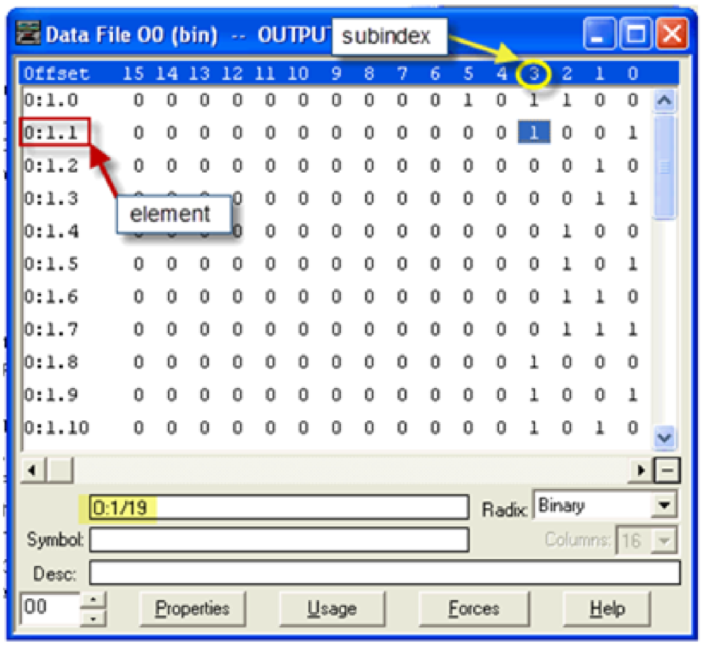
Data Type: boolean

Arraysize: 0

Conversion:

Sub Element: 0

The AGI Creator Tag configured in the example above points on the element shown in the following figure.



Data File 00 (bin) -- OUTPU

Offset	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0:1.0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0
0:1.1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1
0:1.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0:1.3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
0:1.4	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0:1.5	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
0:1.6	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
0:1.7	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
0:1.8	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
0:1.9	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1
0:1.10	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0

0:1/19

Radix: Binary

Symbol:

Columns: 16

Desc:

00

Examples

I:0/19 (I1:0.1/3 in word terms) – 20th Input on CPU

Parameter	Setting
File Type	Disc In
File Num	1
Data Type	Boolean

In the Data File Browser, word 0.1 is Word 1:

Element	1
Sub Index	3

I:1/15 (I1:1.0/15 in word terms) - Last Input on Slot 1 Input Card

Parameter	Setting
File Type	Disc In
File Num	1
Data Type	Boolean

In the Data File Browser, word 1.0 is Word 8:

Element	8
Sub Index	15

I:4.0 (I1:4.0 in word terms) - First Analog Input

Parameter	Setting
File Type	Disc In
File Num	1
Data Type	Short

In the Data File Browser, word 4.0 is Word 10:

Element	10
Sub Index	-

A-B ENET

The A-B ENET communication protocol is normally used on the Allen-Bradley controllers via Ethernet communication.

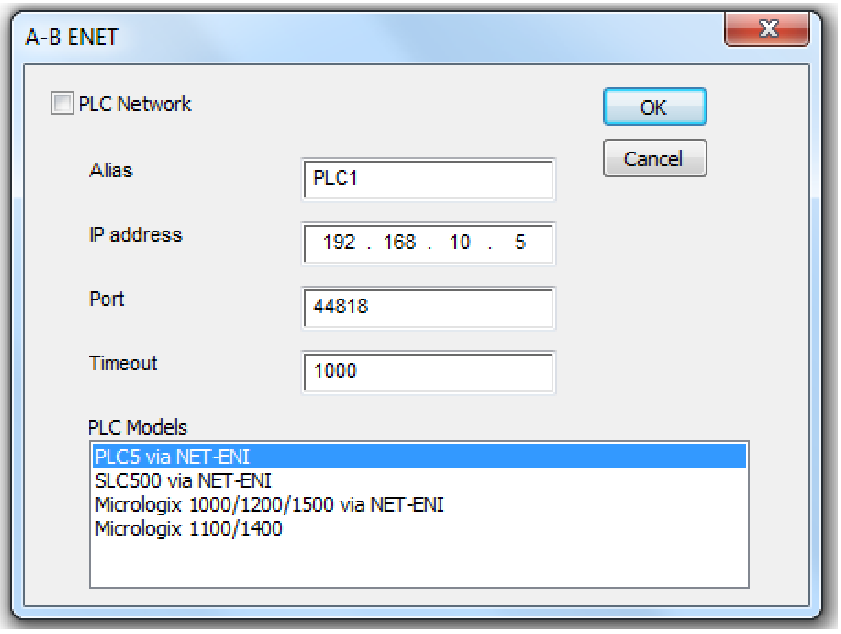
Protocol Editor Settings

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.



A-B ENET

□ PLC Network

Alias: PLC1

IP address: 192 . 168 . 10 . 5

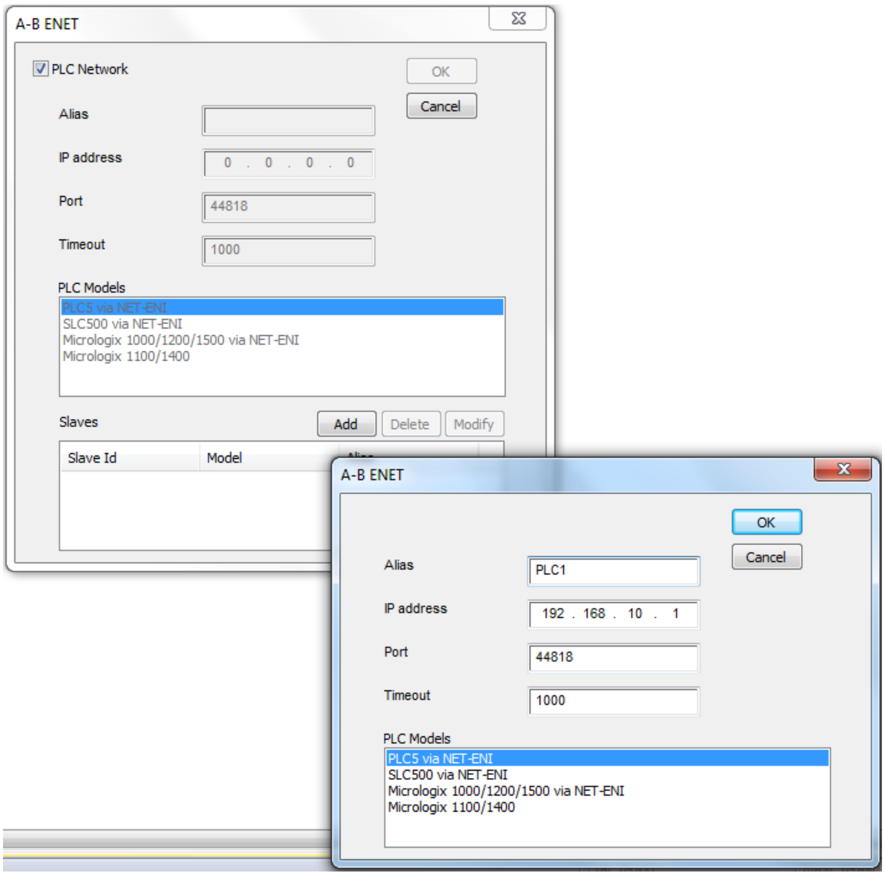
Port: 44818

Timeout: 1000

PLC Models

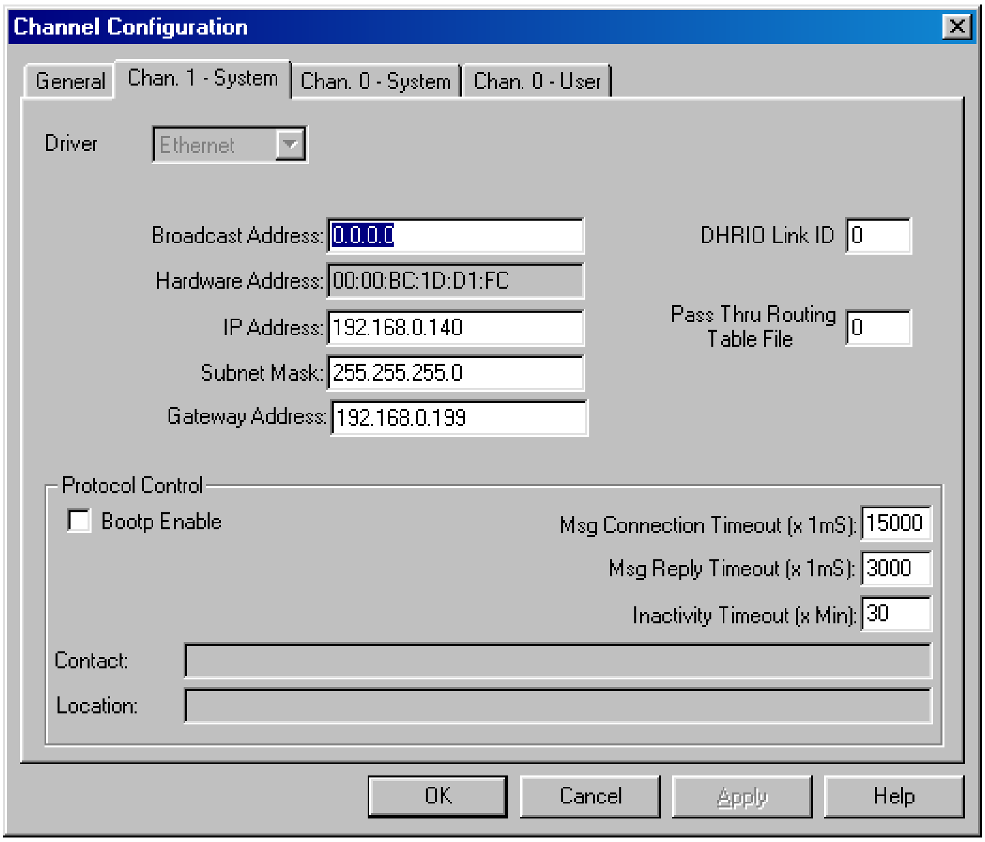
- PLC5 via NET-ENI
- SLC500 via NET-ENI
- Micrologix 1000/1200/1500 via NET-ENI
- Micrologix 1100/1400

Element	Description
Alias	Name identifying nodes in network configurations. The name will be added as a prefix to each tag name imported for each network node.
IP Address	Ethernet IP address of the controller.
Port	Port number used by the Ethernet interface.

Element	Description
Timeout	Time delay in milliseconds between two retries in case of missing response from the controller.
PLC Network	Enable access to multiple networked controllers. For every controller (slave) set the proper option.  A-B ENET ☑ PLC Network Alias: IP address: 0 . 0 . 0 . 0 Port: 44818 Timeout: 1000 PLC Models: PLC5 via NET-ENI (selected), SLC500 via NET-ENI, Micrologix 1000/1200/1500 via NET-ENI, Micrologix 1100/1400 Slaves Slave Id | Model | Alias A-B ENET Alias: PLC1 IP address: 192 . 168 . 10 . 1 Port: 44818 Timeout: 1000 PLC Models: PLC5 via NET-ENI (selected), SLC500 via NET-ENI, Micrologix 1000/1200/1500 via NET-ENI, Micrologix 1100/1400

Controller configuration

The PLC has to be correctly configured to match the IP address configured in the Protocol Editor. Normally the PLC configuration can be left as default.



Channel Configuration

General | Chan. 1 - System | Chan. 0 - System | Chan. 0 - User

Driver: Ethernet

Broadcast Address: 0.0.0.0

Hardware Address: 00:00:BC:1D:D1:FC

IP Address: 192.168.0.140

Subnet Mask: 255.255.255.0

Gateway Address: 192.168.0.199

DHRIO Link ID: 0

Pass Thru Routing Table File: 0

Protocol Control

☐ Bootp Enable

Msg Connection Timeout (x 1mS): 15000

Msg Reply Timeout (x 1mS): 3000

Inactivity Timeout (x Min): 30

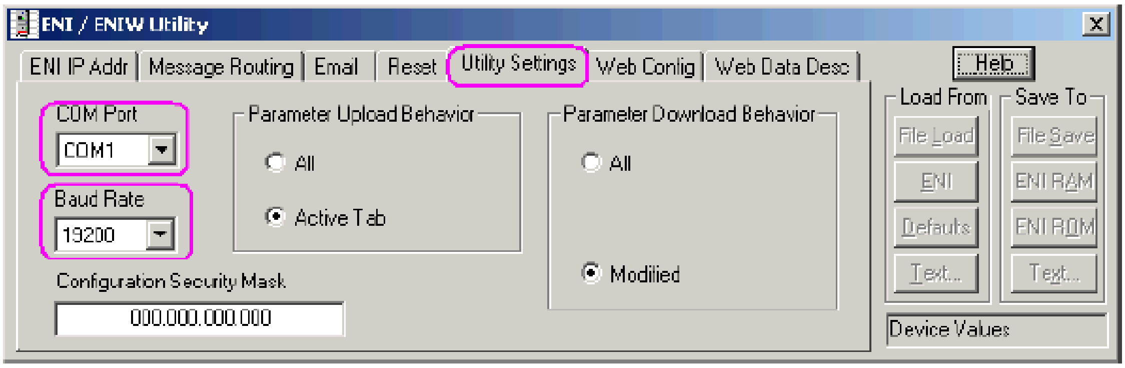
Contact:

Location:

Configuring 1761-NET-ENI

Here is the procedure to configure the 1761-NET-ENI module using the Allen Bradley's ENI/ENIW Utility. The procedure requires a 1761-CBL-PM02 communication cable.

1. Connect the 8 pin din to the port 2 on the NET-ENI device and the 9 pin female D-shell to the computer COM port.
2. Connect the SLC 5/0x controller and go online.
3. In the **Utility Settings** tab, set **COM Port** and **Baud Rate**.



ENI / ENIW Utility

ENI IP Addr | Message Routing | Email | Reset | Utility Settings | Web Config | Web Data Desc

COM Port: COM1

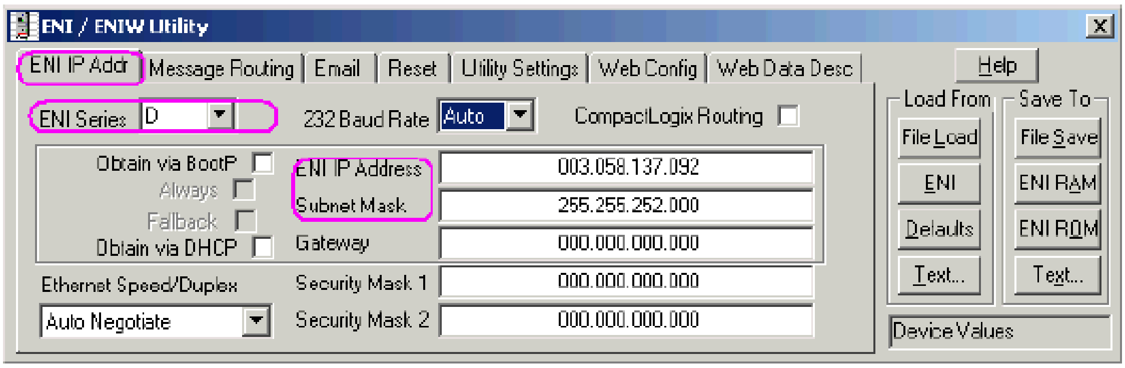
Baud Rate: 19200

Configuration Security Mask: 000.000.000.000

Parameter Upload Behavior: ○ All ● Active Tab

Parameter Download Behavior: ○ All ● Modified

4. In the **ENI IP Addr** tab, select the correct **ENI Series** from the list and set **ENI IP Address, Subnet Mask** and **Baud Rate**, if needed.



ENI Series: D

232 Baud Rate: Auto

CompactLogix Routing ☐

Obtain via BootP ☐

Always ☐

Fallback ☐

Obtain via DHCP ☐

ENI IP Address: 003.058.137.092

Subnet Mask: 255.255.252.000

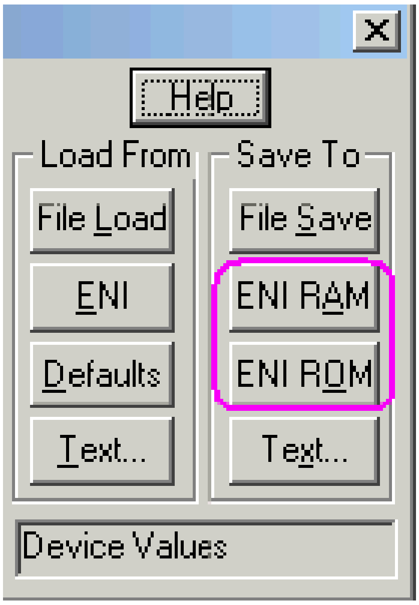
Gateway: 000.000.000.000

Security Mask 1: 000.000.000.000

Security Mask 2: 000.000.000.000

Ethernet Speed/Duplex: Auto Negotiate

5. Save the configuration to the NET-ENI device.



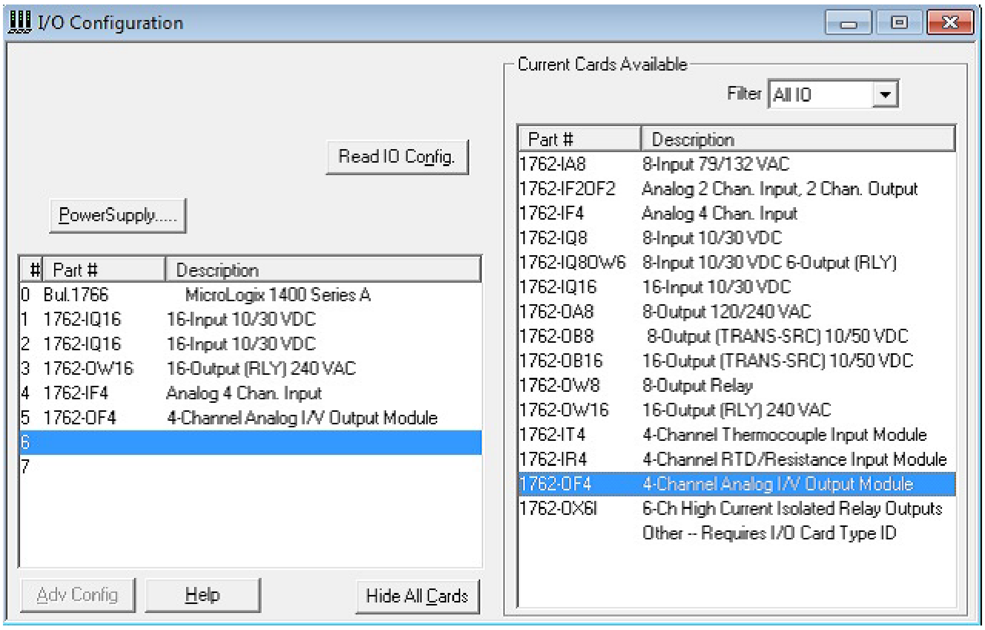
Two separate memory areas are reserved for saving the configuration : **ENI/RAM** (for temporary configurations) and **ENI/ROM** (for permanent configurations).

Logical I/O addressing

When addressing Allen Bradley I/O data, the panel uses logical addressing rather than physical addressing. While physical addressing refers to the element number as the slot number, logical addressing refers to the first element for the first I/O card of a specific file type.

AGI Creator addressing depends on the mapping of the PLC CPU memory and not on the slot number, therefore you should be careful when changing the configuration in order to avoid remapping.

Use the RSLogix 500 I/O Configuration tool layout of the PLC I/O to configure I/O as in the example.



I/O Configuration

Read IO Config.

PowerSupply.....

#	Part #	Description
0	Bul.1766	MicroLogix 1400 Series A
1	1762-IQ16	16-Input 10/30 VDC
2	1762-IQ16	16-Input 10/30 VDC
3	1762-OW16	16-Output (RLY) 240 VAC
4	1762-IF4	Analog 4 Chan. Input
5	1762-OF4	4-Channel Analog I/V Output Module
6		
7		

Adv Config | Help | Hide All Cards

Current Cards Available

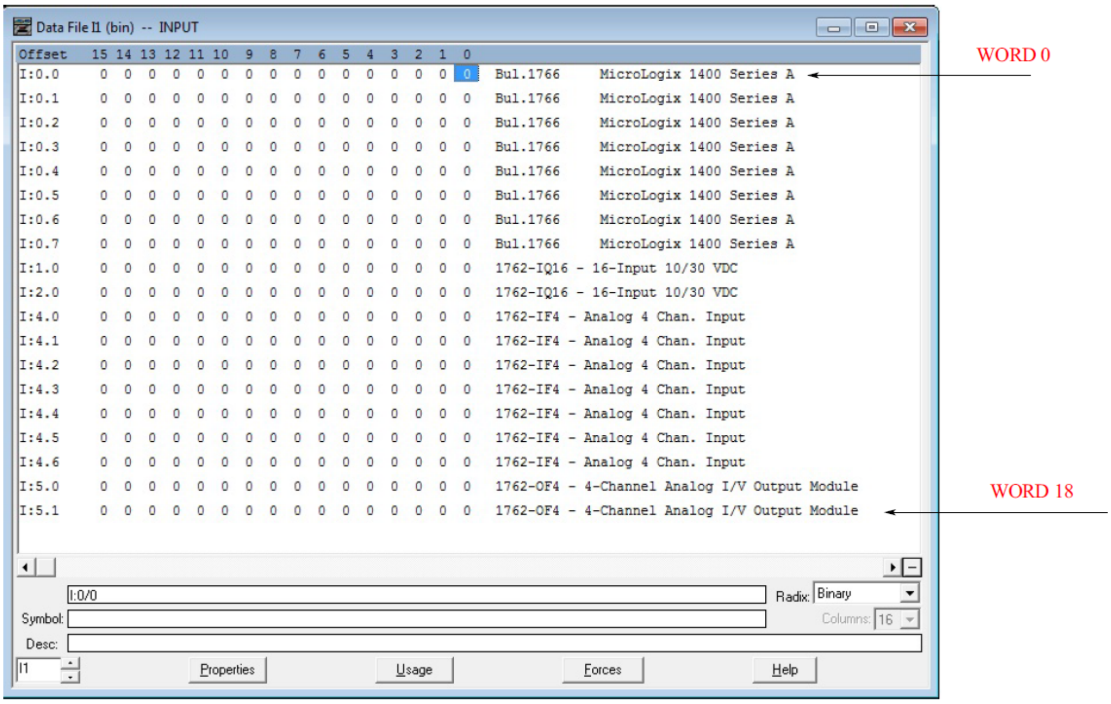
Filter: All IO

Part #	Description
1762-IA8	8-Input 79/132 VAC
1762-IF2OF2	Analog 2 Chan. Input, 2 Chan. Output
1762-IF4	Analog 4 Chan. Input
1762-IQ8	8-Input 10/30 VDC
1762-IQ8OW6	8-Input 10/30 VDC 6-Output (RLY)
1762-IQ16	16-Input 10/30 VDC
1762-OA8	8-Output 120/240 VAC
1762-OB8	8-Output (TRANS-SRC) 10/50 VDC
1762-OB16	16-Output (TRANS-SRC) 10/50 VDC
1762-OW8	8-Output Relay
1762-OW16	16-Output (RLY) 240 VAC
1762-IT4	4-Channel Thermocouple Input Module
1762-IR4	4-Channel RTD/Resistance Input Module
1762-OF4	4-Channel Analog I/V Output Module
1762-OX6I	6-Ch High Current Isolated Relay Outputs
	Other -- Requires I/O Card Type ID

Note: When using a module with a configurable I/O size (for example, Devicenet Scanner) make sure you configure it to the largest possible size or you will have to remap it if you need to allocate more space.

Use the Data File Browser to see how the PLC allocates memory.

This example shows how to configure the AGI Creator Tag for pointing to PLC resource O:1/19 (O1:1.1/3 in word terms).



Data File I1 (bin) -- INPUT

```
Offset  15 14 13 12 11 10  9  8  7  6  5  4  3  2  1  0
I:0.0    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   Bul.1766     MicroLogix 1400 Series A
I:0.1    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   Bul.1766     MicroLogix 1400 Series A
I:0.2    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   Bul.1766     MicroLogix 1400 Series A
I:0.3    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   Bul.1766     MicroLogix 1400 Series A
I:0.4    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   Bul.1766     MicroLogix 1400 Series A
I:0.5    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   Bul.1766     MicroLogix 1400 Series A
I:0.6    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   Bul.1766     MicroLogix 1400 Series A
I:0.7    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   Bul.1766     MicroLogix 1400 Series A
I:1.0    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   1762-IQ16 - 16-Input 10/30 VDC
I:2.0    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   1762-IQ16 - 16-Input 10/30 VDC
I:4.0    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   1762-IF4 - Analog 4 Chan. Input
I:4.1    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   1762-IF4 - Analog 4 Chan. Input
I:4.2    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   1762-IF4 - Analog 4 Chan. Input
I:4.3    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   1762-IF4 - Analog 4 Chan. Input
I:4.4    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   1762-IF4 - Analog 4 Chan. Input
I:4.5    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   1762-IF4 - Analog 4 Chan. Input
I:4.6    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   1762-IF4 - Analog 4 Chan. Input
I:5.0    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   1762-OF4 - 4-Channel Analog I/V Output Module
I:5.1    0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0   1762-OF4 - 4-Channel Analog I/V Output Module
```

I:0/0 | Radix: Binary

Symbol: | Columns: 16

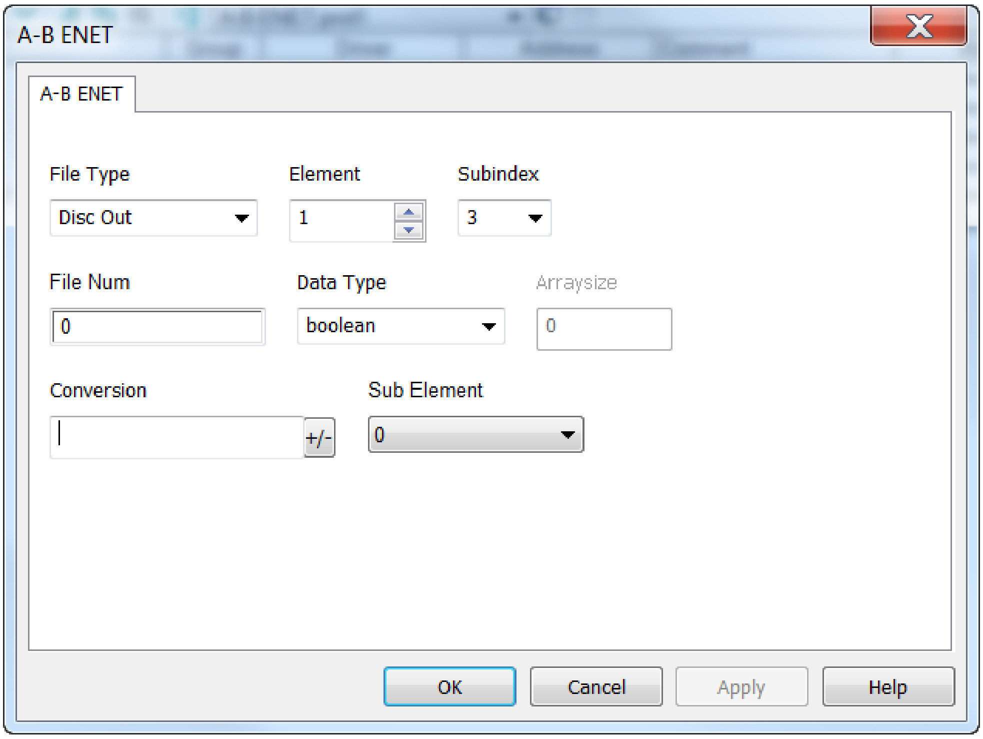
Desc:

I1 | Properties | Usage | Forces | Help

WORD 0

WORD 18

The following figure shows the AGI Creator Tag configuration.



A-B ENET

A-B ENET

File Type: Disc Out

Element: 1

Subindex: 3

File Num: 0

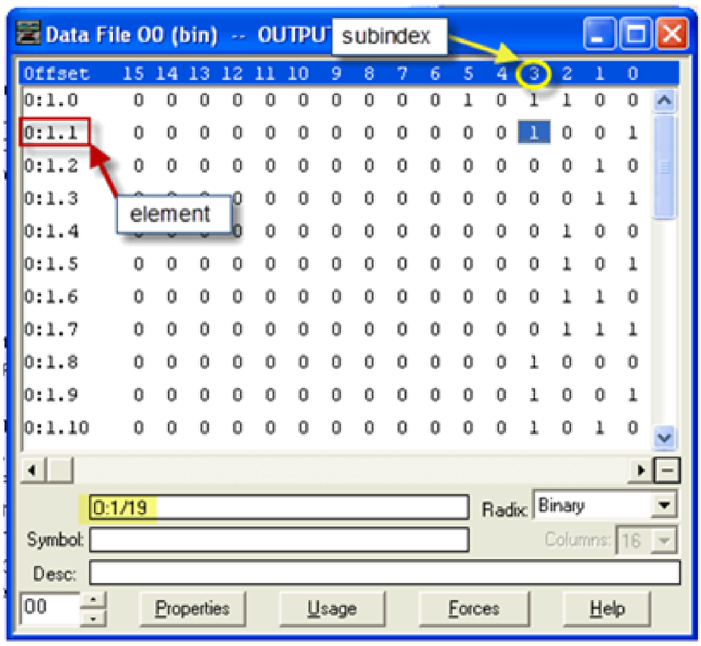
Data Type: boolean

Arraysize: 0

Conversion:

Sub Element: 0

The AGI Creator Tag configured in the example above points on the element shown in the following figure.



Data File 00 (bin) -- OUTPU

Offset	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0:1.0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0
0:1.1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1
0:1.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0:1.3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
0:1.4	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0:1.5	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
0:1.6	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
0:1.7	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
0:1.8	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
0:1.9	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1
0:1.10	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0

0:1/19

Radix: Binary

Symbol:

Columns: 16

Desc:

00

Examples

I:0/19 (I1:0.1/3 in word terms) – 20th Input on CPU

Parameter	Setting
File Type	Disc In
File Num	1
Data Type	Boolean

In the Data File Browser, word 0.1 is Word 1:

Element	1
Sub Index	3

I:1/15 (I1:1.0/15 in word terms) - Last Input on Slot 1 Input Card

Parameter	Setting
File Type	Disc In
File Num	1
Data Type	Boolean

In the Data File Browser, word 1.0 is Word 8:

Element	8
Sub Index	15

I:4.0 (I1:4.0 in word terms) - First Analog Input

Parameter	Setting
File Type	Disc In
File Num	1
Data Type	Short

In the Data File Browser, word 4.0 is Word 10:

Element	10
Sub Index	-

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.

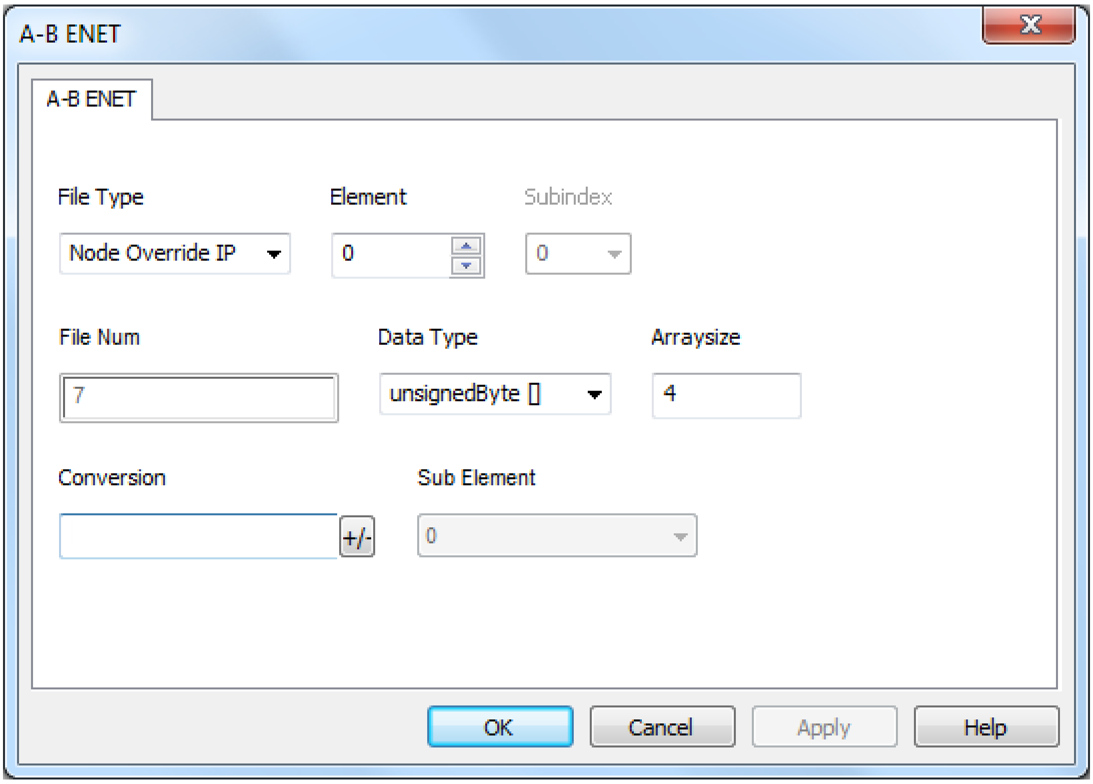
Node Override IP	PLC operation
0.0.0.0	Communication with the controller is stopped, no request frames are generated anymore.
Different from 0.0.0.0	It is interpreted as node IP override and the target IP address is replaced runtime with the new value.

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.


A-B ENET

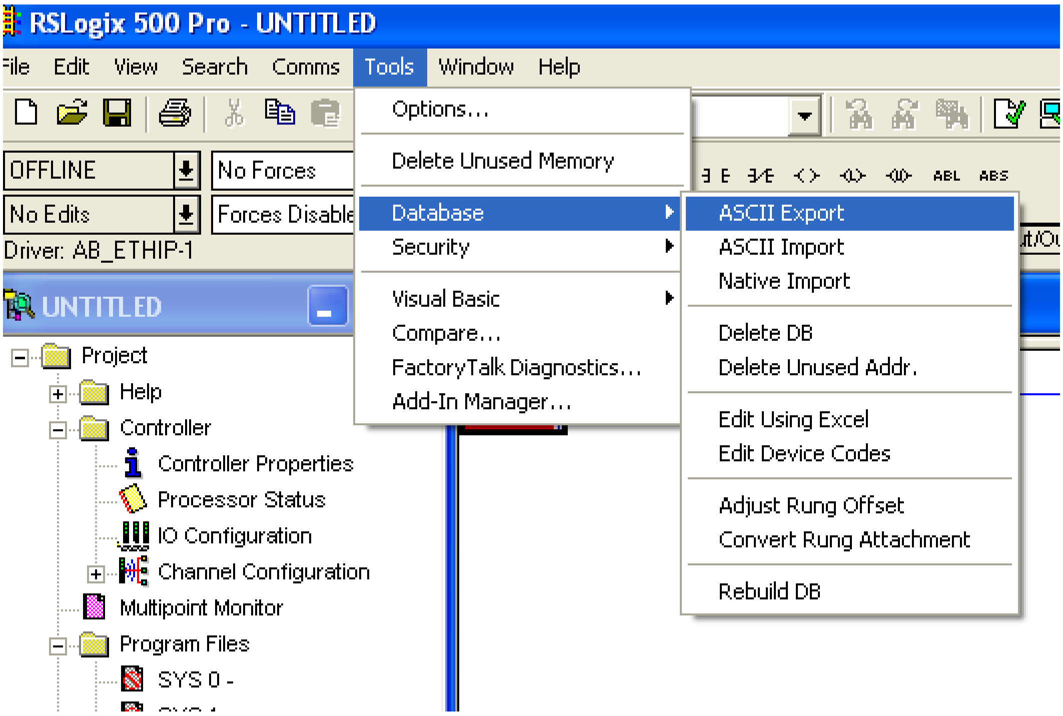
File Type: Node Override IP
Element: 0
Subindex: 0
File Num: 7
Data Type: unsignedByte []
Arraysize: 4
Conversion:
Sub Element: 0

Tag Import

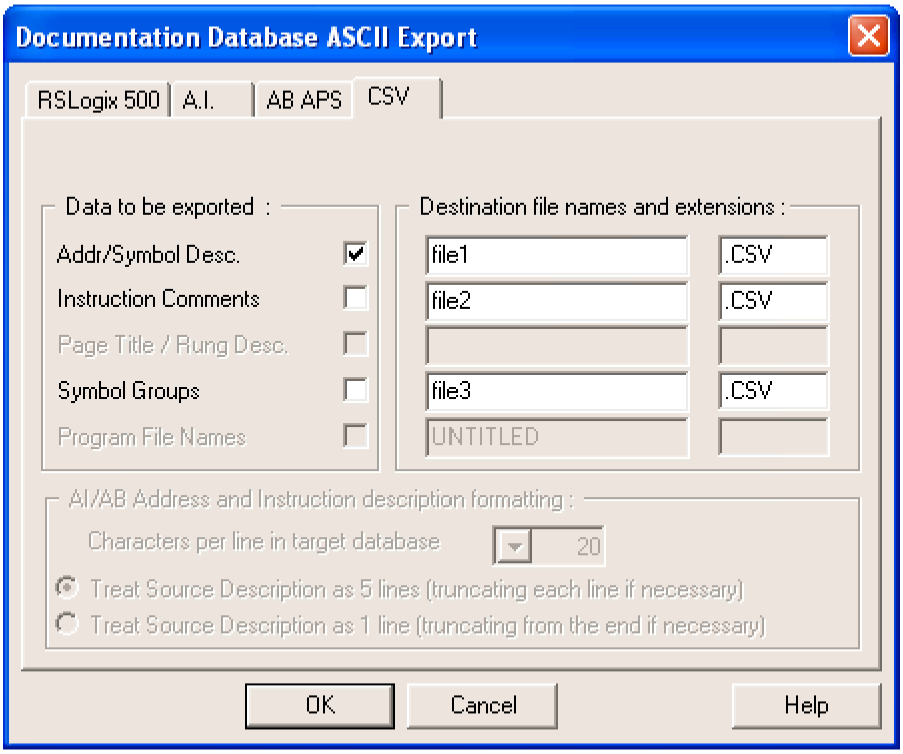
Exporting Tags from PLC

The A-B Ethernet tag import filter accepts symbol files with extension “.csv” created by the Rockwell RSLogix 500.

To create the file select **Tool > Database > ASCII Export**



From **CSV** tab select the data to be exported and give a name to the output csv file.


Documentation Database ASCII Export

RSLogix 500 | A.I. | AB APS | CSV

Data to be exported :

Addr/Symbol Desc.	☑	file1	.CSV
Instruction Comments	□	file2	.CSV
Page Title / Rung Desc.	□		
Symbol Groups	□	file3	.CSV
Program File Names	□	UNTITLED	

Destination file names and extensions :

AI/AB Address and Instruction description formatting :

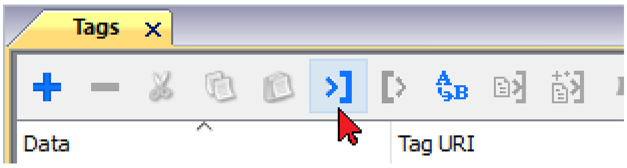
Characters per line in target database 20

● Treat Source Description as 5 lines (truncating each line if necessary)

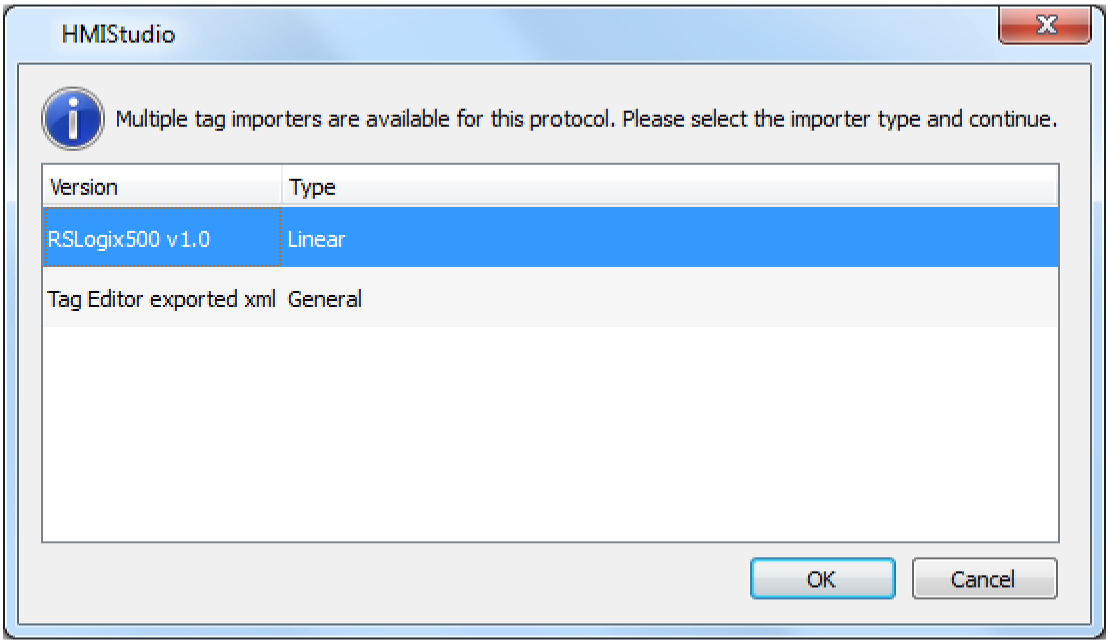
○ Treat Source Description as 1 line (truncating from the end if necessary)

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.

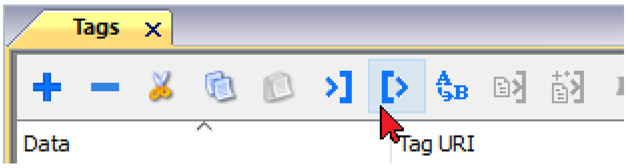


HMIStudio

Multiple tag importers are available for this protocol. Please select the importer type and continue.

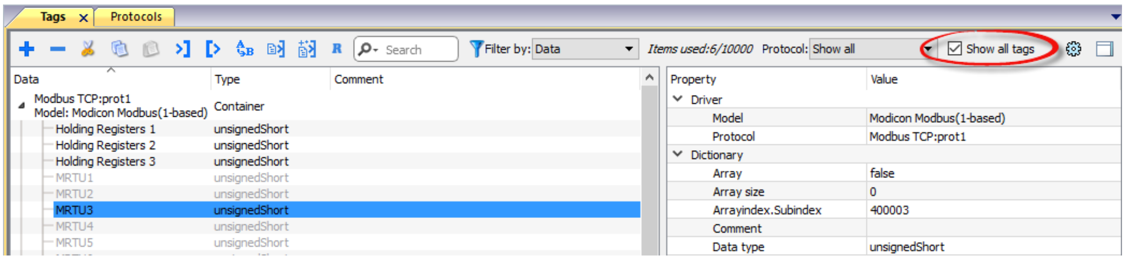
Version	Type
RSLogix500 v1.0	Linear
Tag Editor exported xml	General

OK Cancel

Importer	Description
RSLogix500 v1.0 Linear	Requires an **.csv** file. 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.

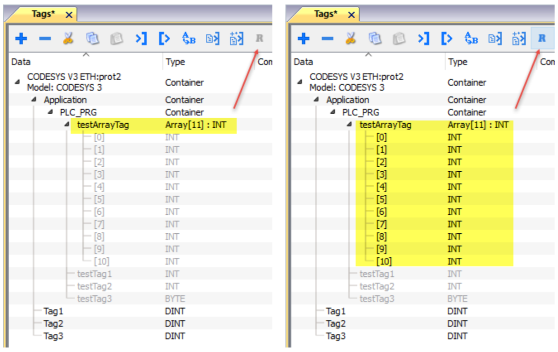


Tags | Protocols

Search Filter by: Data Items used:6/10000 Protocol: Show all ☑ Show all tags

Data	Type	Comment
Modbus TCP:prot1 Model: Modicon Modbus(1-based)	Container	
Holding Registers 1	unsignedShort	
Holding Registers 2	unsignedShort	
Holding Registers 3	unsignedShort	
MRTU1	unsignedShort	
MRTU2	unsignedShort	
MRTU3	unsignedShort	
MRTU4	unsignedShort	
MRTU5	unsignedShort	

Property	Value
Driver	
Model	Modicon Modbus(1-based)
Protocol	Modbus TCP:prot1
Dictionary	
Array	false
Array size	0
Arrayindex.Subindex	400003
Comment	
Data type	unsignedShort

Toolbar item	Description
	Import Tag(s). Select tags to be imported and click on this icon to add tags from tag dictionary to the project
	Update Tag(s). 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: 
	Searches tags in the dictionary basing on filter combo-box item selected.

Tags*

Data	Type	Con
CODESYS V3 ETH:prot2 Model: CODESYS 3	Container	
Application	Container	
PLC_PRG	Container	
testArrayTag	Array[11] : INT	
[0]	INT	
[1]	INT	
[2]	INT	
[3]	INT	
[4]	INT	
[5]	INT	
[6]	INT	
[7]	INT	
[8]	INT	
[9]	INT	
[10]	INT	
testTag1	INT	
testTag2	INT	
testTag3	BYTE	
Tag1	DINT	
Tag2	DINT	
Tag3	DINT	

Tags*

Data	Type	Con
CODESYS V3 ETH:prot2 Model: CODESYS 3	Container	
Application	Container	
PLC_PRG	Container	
testArrayTag	Array[11] : INT	
[0]	INT	
[1]	INT	
[2]	INT	
[3]	INT	
[4]	INT	
[5]	INT	
[6]	INT	
[7]	INT	
[8]	INT	
[9]	INT	
[10]	INT	
testTag1	INT	
testTag2	INT	
testTag3	BYTE	
Tag1	DINT	
Tag2	DINT	
Tag3	DINT	

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
NAK	The controller replies with a not acknowledge.	-
Timeout	A request is not replied within the specified timeout period.	Check if the controller is connected and properly configured to get network access.
Invalid response	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.
General Error	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).

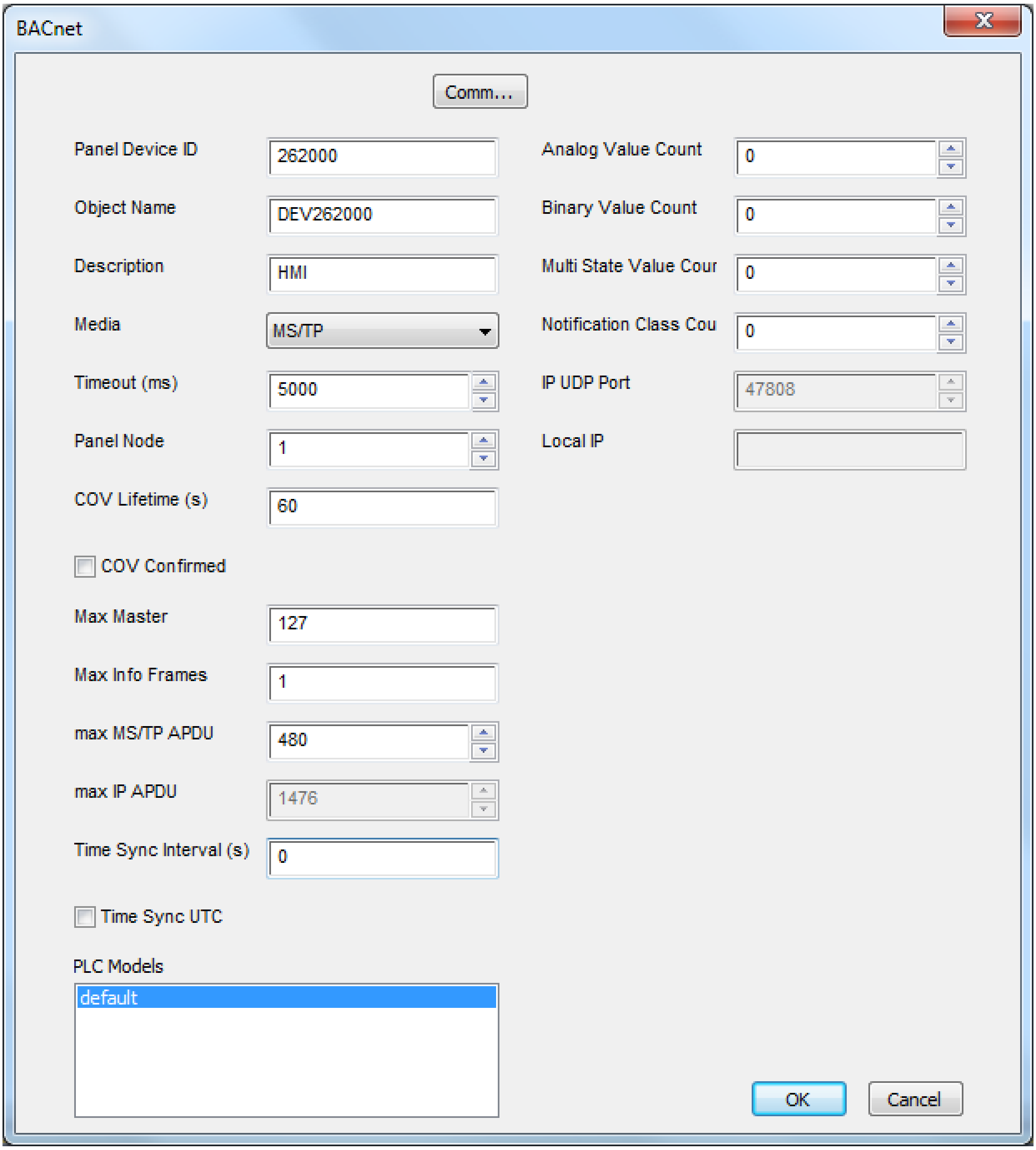
Protocol Editor Settings

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	262000	Analog Value Count	0
Object Name	DEV262000	Binary Value Count	0
Description	HMI	Multi State Value Cour	0
Media	MS/TP	Notification Class Cou	0
Timeout (ms)	5000	IP UDP Port	47808
Panel Node	1	Local IP	
COV Lifetime (s)	60		

□ COV Confirmed

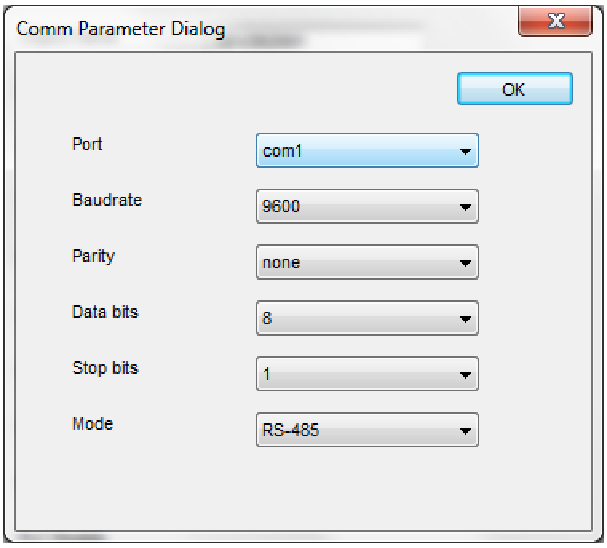
Max Master	127
Max Info Frames	1
max MS/TP APDU	480
max IP APDU	1476
Time Sync Interval (s)	0

□ Time Sync UTC

PLC Models

default

Element	Description
Panel Device ID	Identifies the HMI device in the network.
Object Name	BACnet Object Name for the HMI device.
Description	HMI device description, for documentation purposes.
Media	Type of communication of the protocol. • **MS/TP**: Master-Slave/Token-Passing communication (RS-485).

Element	Description
	• **IP**: based on standard UDP/IP communication.
Timeout (ms)	Time delay in milliseconds between two retries in case of missing response from the BACnet device.
Panel Node *	MS/TP address. Physical device address on the link; it is not passed through routers.
COV Lifetime (s)	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.
Max Master *	Highest allowable address for master nodes. Must be less than or equal to 127.
Max Info Frames *	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.
Max MS/TP APDU *	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).
Max IP APDU **	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).
Time Sync Interval (s)	Represent the interval between every time synchronization, in seconds. If left to 0, time synchronization is disabled.
Time Sync UTC	Option to synchronize time in UTC format. If disabled, local time format used.
PLC Models	Reserved for future use.
Comm... *	If clicked displays the communication parameters setup dialog.  Comm Parameter Dialog Port: com1 Baudrate: 9600 Parity: none Data bits: 8 Stop bits: 1 Mode: RS-485

<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>Element</th><th>Description</th></tr>
<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:
• <b>RS-232</b>
• <b>RS-485</b>
• <b>RS-422</b></td></tr>
</table>
</td></tr>
<tr><td><b>Analog Value Count ***</b></td><td>Number of Analog Value objects to be instanced in BACnet Server.
Min: 0
Max: 200</td></tr>
<tr><td><b>Binary Value Count ***</b></td><td>Number of Binary Value objects to be instanced in BACnet Server.
Min: 0
Max: 200</td></tr>
<tr><td><b>Multi State Value Count ***</b></td><td>Number of Multi State Value objects to be instanced in BACnet Server.
Min: 0
Max: 200</td></tr>
<tr><td><b>Notification Class Count ***</b></td><td>Number of Notifications Class objects to be instanced in BACnet Server.
Min: 0
Max: 200</td></tr>
<tr><td><b>IP UDP Port **</b></td><td>Port number for IP communication.</td></tr>
<tr><td><b>Local IP **</b></td><td>IP Address of the network adapter to use for protocol. Not required if the device has only one Ethernet adapter.</td></tr>
</table>

Note *: Available only if media is set to **MS/TP**.

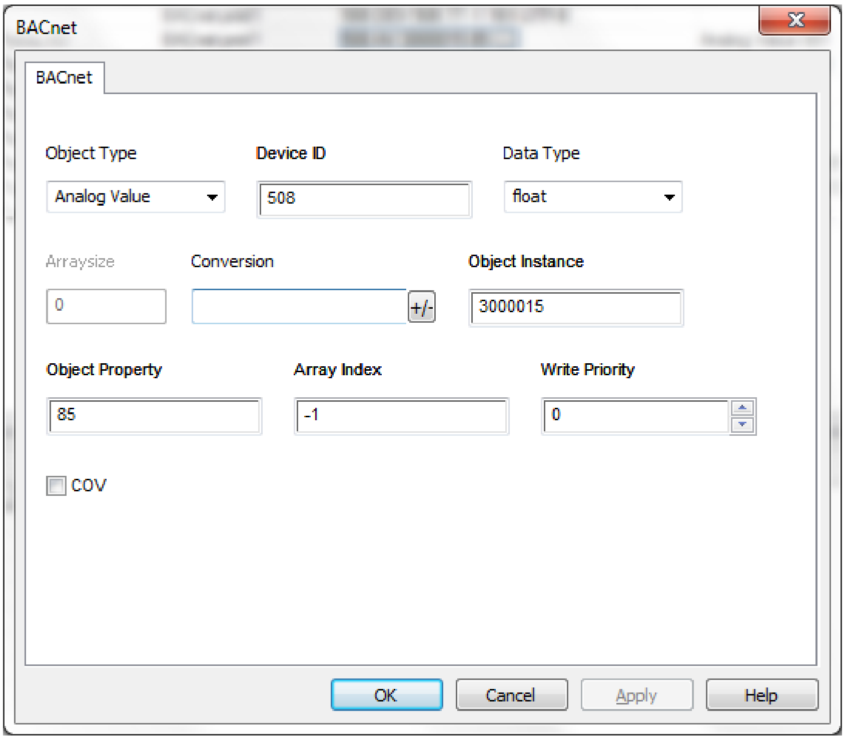
Note **: Available only if media is set to **IP**.

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.



Object Type: Analog Value

Device ID: 508

Data Type: float

Arraysize: 0

Conversion: +/-

Object Instance: 3000015

Object Property: 85

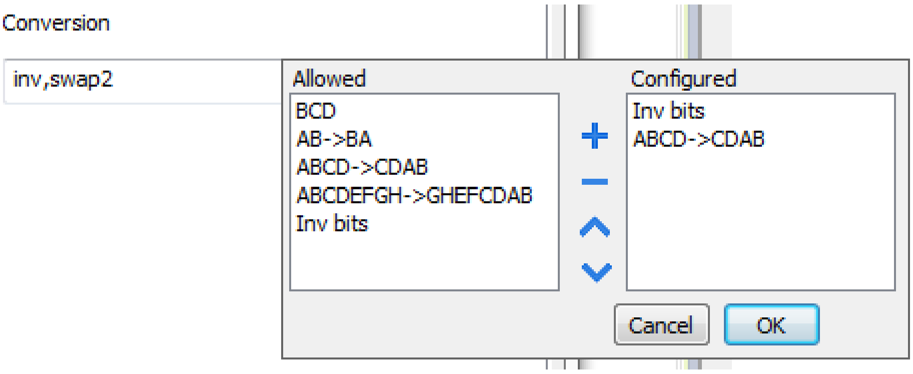
Array Index: -1

Write Priority: 0

□ COV

Element	Description
Object Type	Type of BACnet object to be referenced. Available object types: - **Device** - **Analog Input** - **Analog Output** - **Analog Value** - **Binary Input** - **Binary Output** - **Binary Value** - **Multi-state Input** - **Multi-state Output** - **Multi-state Value** - **Integer Value** - **Positive Integer Value** - **Large Analog Value**
Device ID	ID of the device containing the object.
Data Type	Data type for display presentation. Available data types: - **boolean**

<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td></td><td>
<ul>
<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>
<tr><th>BACnet data type</th><th>Software data type</th><th>Notes</th></tr>
<tr><td><b>BOOLEAN</b></td><td>Boolean</td><td><b>-</b></td></tr>
<tr><td><b>INTEGER</b></td><td>Int</td><td><b>-</b></td></tr>
<tr><td><b>UNSIGNED_INTEGER</b></td><td>unsignedInt</td><td><b>-</b></td></tr>
<tr><td><b>REAL</b></td><td>Float</td><td><b>-</b></td></tr>
<tr><td><b>BIT_STRING</b></td><td>boolean-x</td><td><b>x = size</b></td></tr>
<tr><td><b>CHARACTER_ STRING</b></td><td>string-x</td><td><b>x = size</b></td></tr>
<tr><td><b>OCTET_STRING</b></td><td>binary-x</td><td><b>x = size</b></td></tr>
<tr><td><b>DATE</b></td><td>int or unsignedInt</td><td><b>-</b></td></tr>
<tr><td><b>TIME</b></td><td>int or unsignedInt</td><td><b>-</b></td></tr>
<tr><td><b>BACnetObjectIdentifier</b></td><td>int or unsignedInt</td><td><b>Use conversions instance and objType for proper display</b></td></tr>
</table>
</td></tr>
<tr><td><b>Arraysize</b></td><td>
<ul>
<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>
</td></tr>
<tr><td><b>Conversion</b></td><td>Conversion to be applied to the tag.</td></tr>
</table>

<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td></td><td>


Conversion
inv,swap2
Allowed
BCD
AB->BA
ABCD->CDAB
ABCDEFGH->GHEFCDAB
Inv bits
Configured
Inv bits
ABCD->CDAB
Cancel OK

Depending on data type selected, the list <b>Allowed</b> shows one or more conversion types.
<table>
<tr><th>Value</th><th>Description</th></tr>
<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 -> 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 -> 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 -> 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 -> OPM...DAB</b></td><td><b>swap8</b>: Swap bytes in a long word.
Example:
142.366 → -893553517.588905 (in decimal format)
0 10000000110</td></tr>
</table>
</td></tr>
</table>

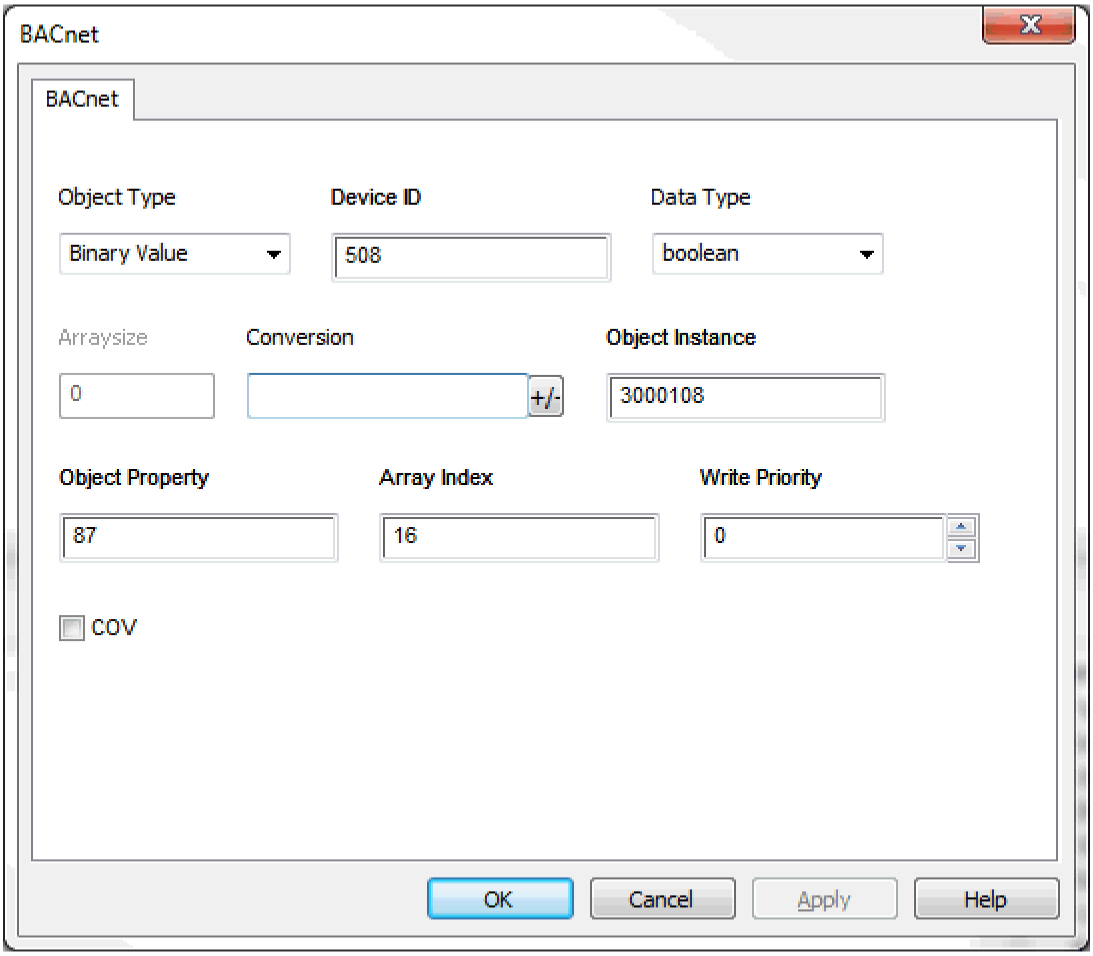
<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td></td><td>0001110010111011011001000101101000011100101011000001
→
1 10000011100
1010101000010100010110110110110010110110000100111101
(in binary format)</td></tr>
<tr><td><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>
</table>
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.</td></tr>
<tr><td><b>Object Instance</b></td><td>BACnet ID of the object to be referenced.</td></tr>
<tr><td><b>Object Property</b></td><td>Numeric value of the property to be referenced (example: the value 85 means <i>present-value</i> for most standard objects).
The table below specifies all the BACnet Object Properties.
<table>
<tr><th>Property</th><th>Value</th><th>Property</th><th>Value</th><th>Property</th><th>Value</th><th>Property</th><th>Value</th></tr>
<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-enable</td><td>35</td><td>max-segment</td><td>167</td><td>record-count</td><td>141</td></tr>
</table></td></tr>
</table>

Element	Description							
	Property	**Value**	**Property**	**Value**	**Property**	**Value**	**Property**	**Value**
					s-accepted			
	action-text	3	event-state	36	member-of	159	reliability	103
	active-text	4	event-time-stamps	130	minimum-off-time	66	relinquish-default	104
	active-vt-sessions	5	event-type	37	minimum-on-time	67	required	105
	active-cov-subscriptions	152	event-parameters	83	minimum-output	68	resolution	106
	adjust-value	176	exception-schedule	38	minimum-value	136	scale	187
	alarm-value	6	fault-values	39	minimum-value-timestamp	150	scale-factor	188
	alarm-values	7	feedback-value	40	min-pres-value	69	schedule-default	174
	all	8	file-access-method	41	mode	160	segmentation-supported	107
	all-writes-successful	9	file-size	42	model-name	70	setpoint	108
	apdu-segment-timeout	10	file-type	43	modification-date	71	setpoint-reference	109
	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

Element	Description							
	Property	**Value**	**Property**	**Value**	**Property**	**Value**	**Property**	**Value**
	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
	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

Element	Description							
	Property	Value	Property	Value	Property	Value	Property	Value
	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-resubscription-interval	128	log-enable	133	process-identifier	89	value-set	191
	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-	158	proportional-	93	vt-classes-supported	122

<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>Property</th><th>Value</th><th>Property</th><th>Value</th><th>Property</th><th>Value</th><th>Property</th><th>Value</th></tr>
<tr><td></td><td></td><td>required</td><td></td><td>constant</td><td></td><td></td><td></td></tr>
<tr><td>derivative-constant-units</td><td>27</td><td>manipulated-variable-reference</td><td>60</td><td>proportional-constant-units</td><td>94</td><td>weekly-schedule</td><td>123</td></tr>
<tr><td>description</td><td>28</td><td>manual-slave-address-binding</td><td>170</td><td>protocol-object-types-supported</td><td>96</td><td>window-interval</td><td>147</td></tr>
<tr><td>description-of-halt</td><td>29</td><td>maximum-output</td><td>61</td><td>protocol-revision</td><td>139</td><td>window-samples</td><td>148</td></tr>
<tr><td>device-address-binding</td><td>30</td><td>maximum-value</td><td>135</td><td>protocol-services-supported</td><td>97</td><td>zone-members</td><td>165</td></tr>
<tr><td>device-type</td><td>31</td><td>maximum-value-timestamp</td><td>149</td><td>protocol-version</td><td>98</td><td></td><td></td></tr>
<tr><td>direct-reading</td><td>156</td><td>max-apdu-length-accepted</td><td>62</td><td>read-only</td><td>99</td><td></td><td></td></tr>
</table>
</td></tr>
<tr><td><b>Array Index</b></td><td>Index for subscribing elements in BACnet arrays.
<ul><li>-1 means read all elements</li><li>0 to n means read the specified element</li></ul>
<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.</td></tr>
</table>

Element	Description
	 BACnet Object Type: Binary Value Device ID: 508 Data Type: boolean Arraysize: 0 Conversion: Object Instance: 3000108 Object Property: 87 Array Index: 16 Write Priority: 0 □ COV
Write Priority	Write requests priority level. The value is in the range 1-16. 0 is interpreted as 16.
COV	Enable the Change Of Value notification.

Clear/Set Priority

The system offers actions for a more flexible handling of Write Priority.

Action	Description
BACnetClearPriority	Clears the priority array at the position associated to the BACnet tag passed as parameter. This action has immediate effect on the BACnet device.
BACnetClearAllPriorities	Clears all positions in the priority array. This action has immediate effect on the BACnet device.
BACnetSetPriority	Overrides the Write Priority value configured in the BACnet tag definition. This action has two parameters: • **TagName**: name of the BACnet tag. • **TagPriority**: new value of Write Priority for the BACnet tag passed as parameter. 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. The priority value will be valid until: • A new call to the BACnetSetPriority action changes it. • The HMI device is restarted. The value of WritePriority defined in the project is valid in this case.

Tag Import

BACnet object information can be imported from BACnet EDE (Engineering Data Exchange) files. The EDE file must have the .csv extension.

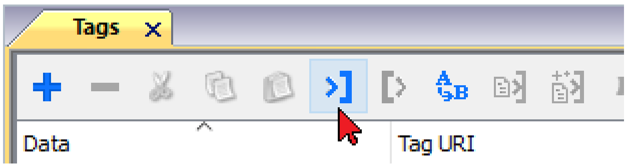
The importer uses the characters “,” and “;” as delimiters. They are considered as reserved characters and you cannot use them in file name.

Use the hierarchical importer to have a ordered list of BACnet objects and properties.

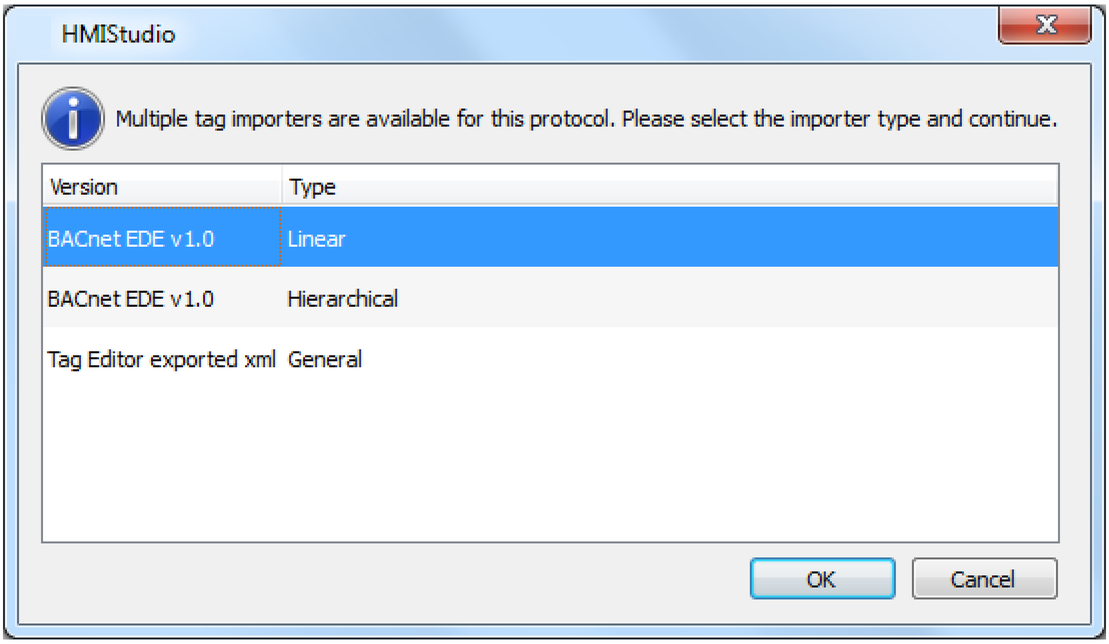
Tags will be created using the string specified in the column object-name of the EDE file. The importer will add the device ID as a prefix to avoid duplication of tag names.

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.



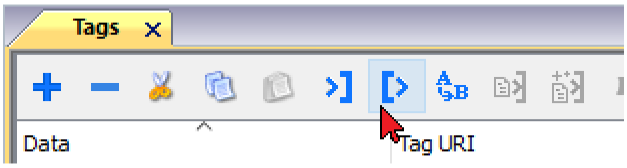
The following dialog shows which importer type can be selected.



HMIStudio

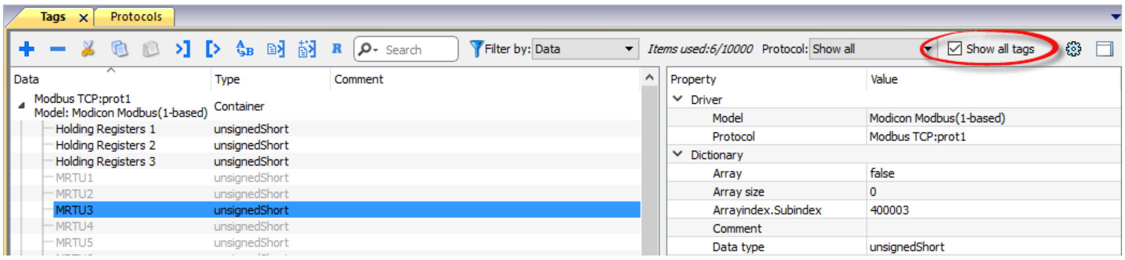
Multiple tag importers are available for this protocol. Please select the importer type and continue.

Version	Type
BACnet EDE v1.0	Linear
BACnet EDE v1.0	Hierarchical
Tag Editor exported xml	General

Importer	Description
BACnet EDE v1.0 Linear	Requires a **.csv** file. All variables will be displayed at the same level.
BACnet EDE v1.0 Hierarchical	Requires a **.csv** file. All variables will be displayed according to BACnet EDE Hierarchical view.
Tag Editor exported xml	Select this importer to read a generic XML file exported from Tag Editor by appropriate button. 

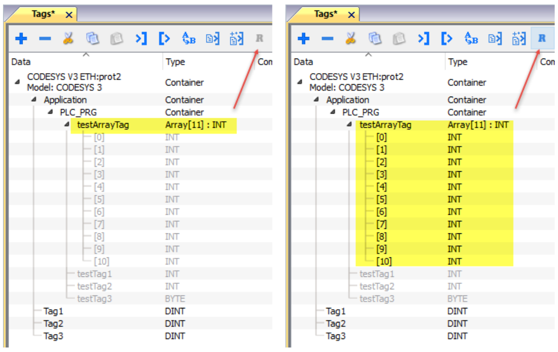
Once the importer has been selected, locate the symbol file and click **Open**.

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.



Data	Type	Comment
Modbus TCP:prot1 Model: Modicon Modbus(1-based)	Container	
Holding Registers 1	unsignedShort	
Holding Registers 2	unsignedShort	
Holding Registers 3	unsignedShort	
MRTU1	unsignedShort	
MRTU2	unsignedShort	
MRTU3	unsignedShort	
MRTU4	unsignedShort	
MRTU5	unsignedShort	

Property	Value
Driver	
Model	Modicon Modbus(1-based)
Protocol	Modbus TCP:prot1
Dictionary	
Array	false
Array size	0
Arrayindex.Subindex	400003
Comment	
Data type	unsignedShort

Toolbar item	Description
	Import Tag(s). Select tags to be imported and click on this icon to add tags from tag dictionary to the project
	Update Tag(s). Click on this icon to update the tags in the project, due a new dictionary import.
R	Check this box to import all sub-elements of a tag. Example of both checked and unchecked result:  Tags* Data / Type / Con CODESYS V3 ETH:prot2 Model: CODESYS 3 / Container Application / Container PLC_PRG / Container testArrayTag / Array[11] : INT [0] / INT [1] / INT [2] / INT [3] / INT [4] / INT [5] / INT [6] / INT [7] / INT [8] / INT [9] / INT [10] / INT testTag1 / INT testTag2 / INT testTag3 / BYTE Tag1 / DINT Tag2 / DINT Tag3 / DINT
Search Filter by: Tag name ▾	Searches tags in the dictionary basing on filter combo-box item selected.

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

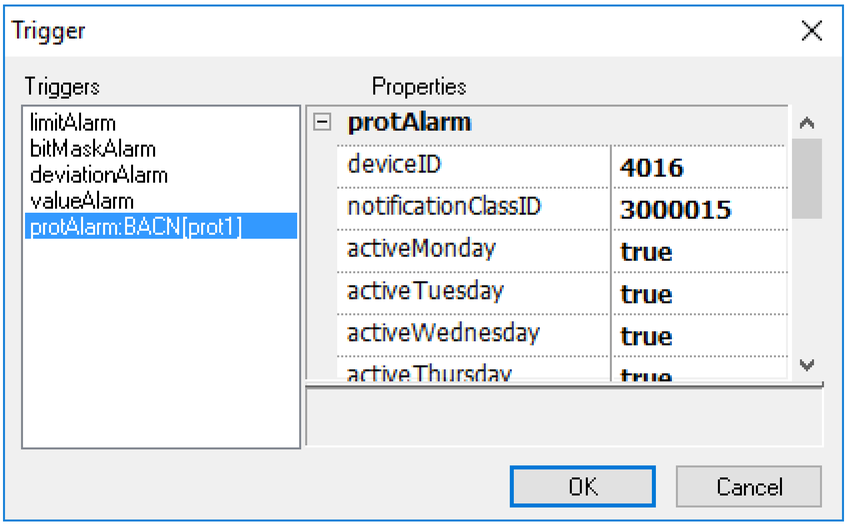
A BACnet network scanner can detect properties when exploring the network and obtaining data from HMI device.

This 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, give the possibility to receive alarm events from the BACnet native alarms module.



Trigger

Triggers

- limitAlarm
- bitMaskAlarm
- deviationAlarm
- valueAlarm
- protAlarm:BACN[prot1]

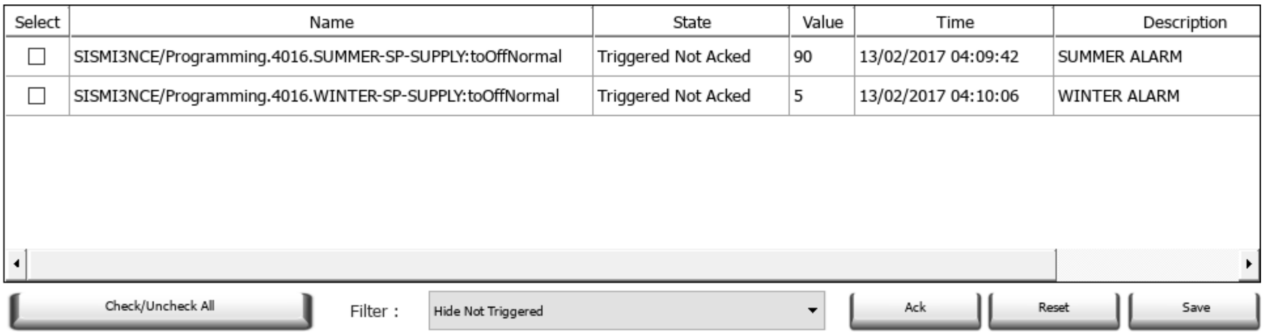
Properties

protAlarm	
deviceID	4016
notificationClassID	3000015
activeMonday	true
activeTuesday	true
activeWednesday	true
activeThursday	true

Property	Description
deviceID	Identifies the BACnet device in the network.
notificationClassID	Notification Class ID to subscribe for the alarm events retrieving
processID	Not used
activeMonday **activeTuesday** **activeWednesday** **activeThrusday** **activeFriday** **activeSaturday** **activeSunday**	Define in which days keep active the alarm events subscription • False Subscription not active • True Subscription active

startHour
startMinute
startSecond
endHour
endMinute
endSecond
	Define the time window where the alarm events subscription will be active

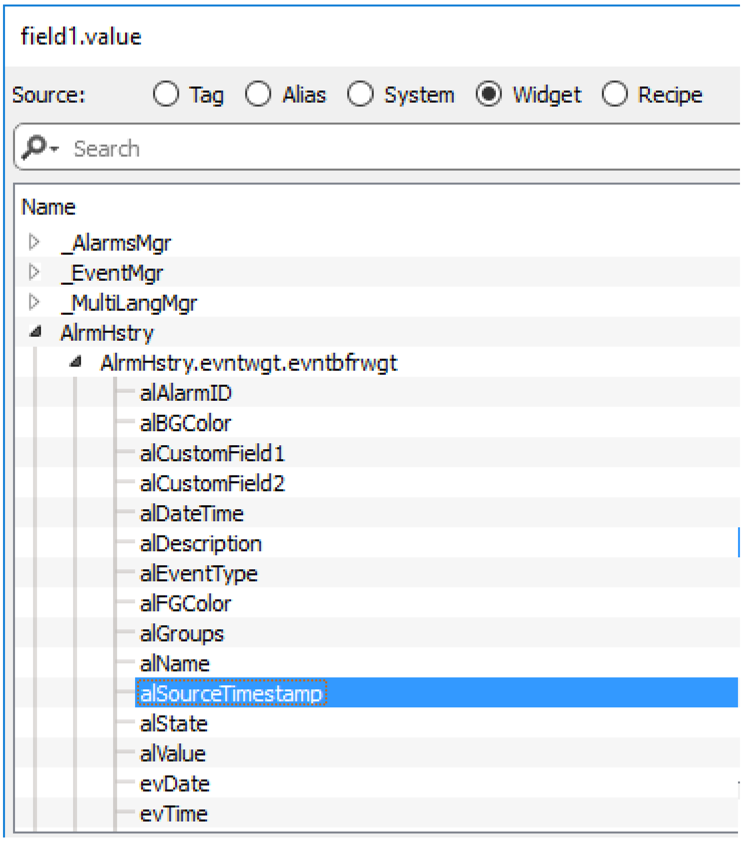
The alarm widgets will report the alarm information that are provided from the BACnet device.



Select	Name	State	Value	Time	Description
□	SISMI3NCE/Programming.4016.SUMMER-SP-SUPPLY:toOffNormal	Triggered Not Acked	90	13/02/2017 04:09:42	SUMMER ALARM
□	SISMI3NCE/Programming.4016.WINTER-SP-SUPPLY:toOffNormal	Triggered Not Acked	5	13/02/2017 04:10:06	WINTER ALARM

Filter : Hide Not Triggered

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.



field1.value

Source: ○ Tag ○ Alias ○ System ● Widget ○ Recipe

Search

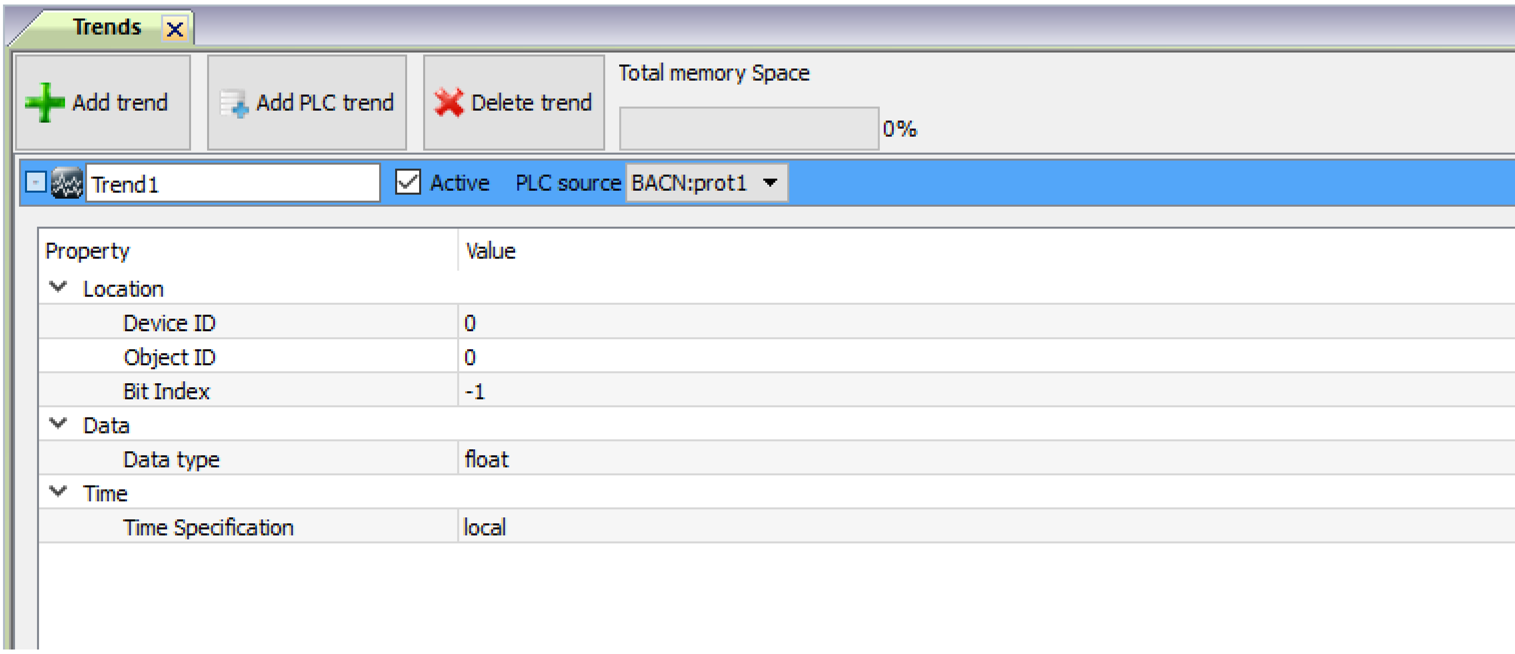
Name
- _AlarmsMgr
- _EventMgr
- _MultiLangMgr
- AlrmHstry
  - AlrmHstry.evntwgt.evntbfrwgt
    - alAlarmID
    - alBGColor
    - alCustomField1
    - alCustomField2
    - alDateTime
    - alDescription
    - alEventType
    - alFGColor
    - alGroups
    - alName
    - alSourceTimestamp
    - alState
    - alValue
    - evDate
    - evTime

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.



Total memory Space 0%

Trend1 ☑ Active PLC source BACN:prot1

Property	Value
Location	
Device ID	0
Object ID	0
Bit Index	-1
Data	
Data type	float
Time	
Time Specification	local

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: • boolean • int • unsignedInt • float
Time Specification	Time format used inside the selected BACnet trend object • local • global (UTC)

The trend buffer thus configured can then be used inside any trend widgets.

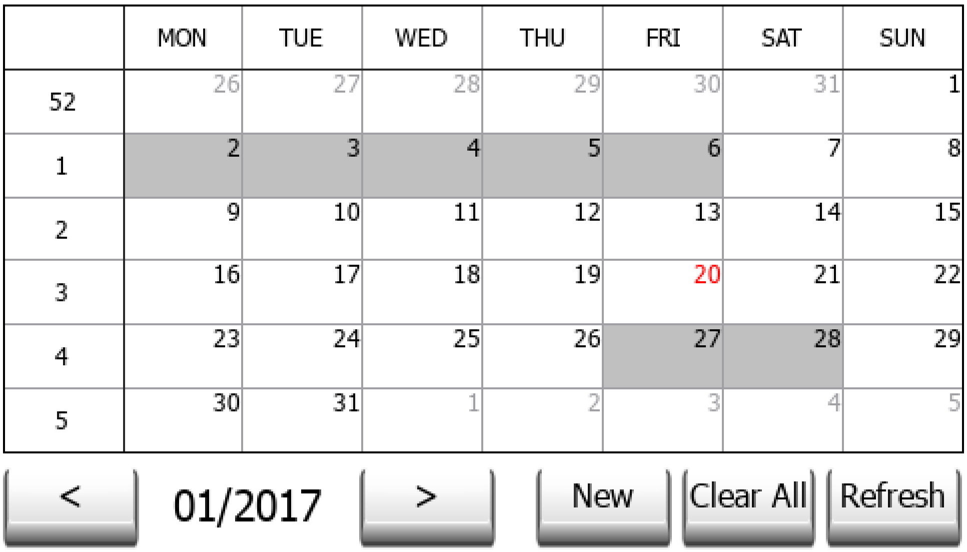
BACnet Calendar Widget

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. 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.

Operation of Calendar Widget

The widget shows data for one month.



	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

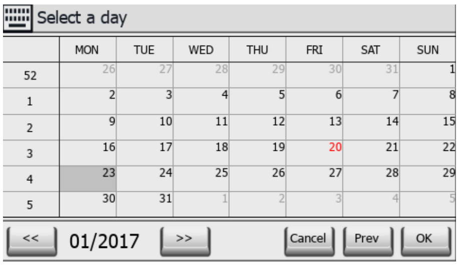
Use the < and > buttons to select the month to be displayed. The date of first day of the month is shown.

Swing gesture can be used on the widget to select the date.

New

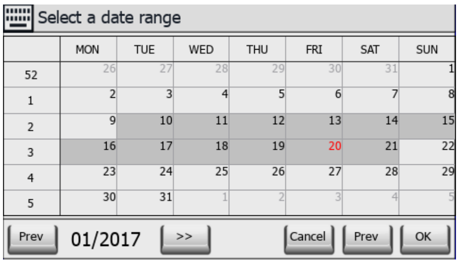
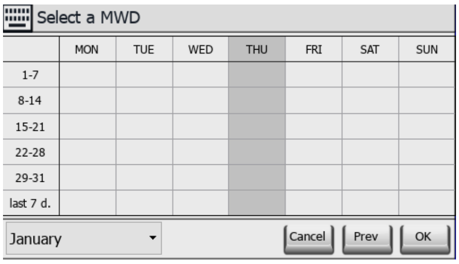
Press the button “New” to enter a new calendar item. The button is active only if the tag associated to the calendar has been configured as Read/Write.

<table>
<tr><th>Calendar item</th><th>Description</th></tr>
<tr><td><b>Single</b></td><td>Click on a day to select a single day into the calendar



<table>
<tr><th></th><th>MON</th><th>TUE</th><th>WED</th><th>THU</th><th>FRI</th><th>SAT</th><th>SUN</th></tr>
<tr><td>52</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td><td>1</td></tr>
<tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td></tr>
<tr><td>2</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td></tr>
<tr><td>3</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td></tr>
<tr><td>4</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td></tr>
<tr><td>5</td><td>30</td><td>31</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr>
</table>

</td></tr>
<tr><td><b>Range</b></td><td>Click on the first day and on the last day to select a range of days into the</td></tr>
</table>

Calendar item	Description
	calendar. - 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.  Select a date range 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 Prev 01/2017 >> Cancel Prev OK
MWD	Select a Day or a Week for each year or each month.  Select a MWD MON TUE WED THU FRI SAT SUN 1-7 8-14 15-21 22-28 29-31 last 7 d. January Cancel Prev OK

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.

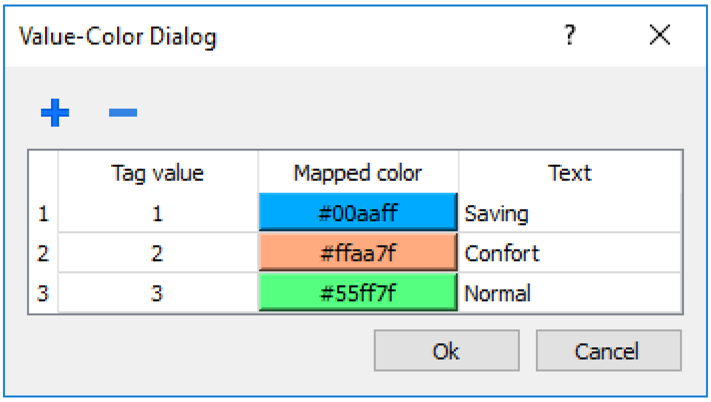
Refresh

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
Type	Select the type of BACnet object controlled by the schedule. Options are: - Binary - Real - Multistate
Weekly_Schedule	Attach to the Weekly_Schedule tag of the schedule object. The tag can be Read Only or Read/Write.
Exception_Schedule	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.
Default_Value	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.
Cal. 0 (Date_List)	Optionally attach to the Date_List tag of the schedule widget in Read Only mode. Use this options to show the “calendar reference” exceptions. 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 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. Note: If it is not attached to a calendar, it is not possible to insert calendar exception. See BACNSchedKeypad for details.
Cal. 0 (Object_Name)	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.
Cal. 1 (Date_List)	Option for a second calendar.

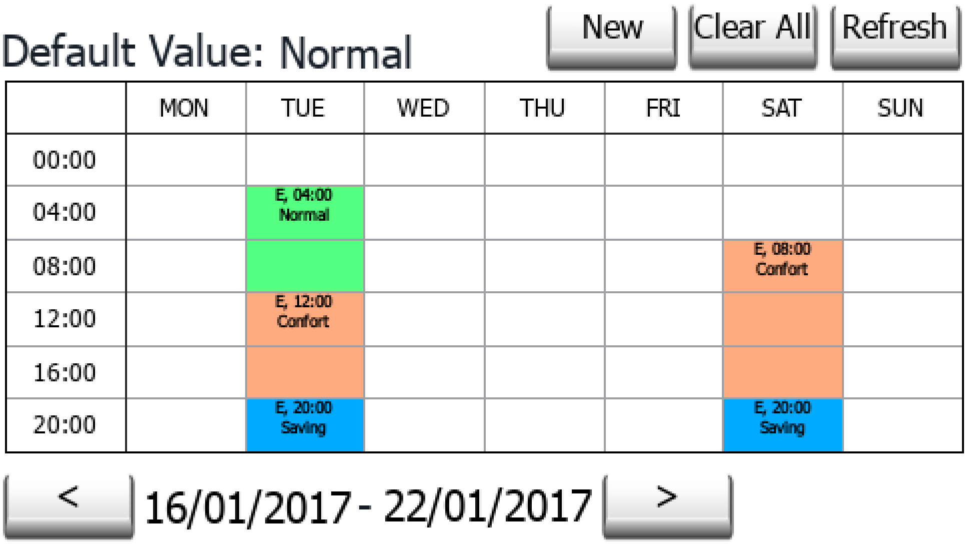
Property	Description
Cal. 1 (Object_Name)	Option for a second calendar.
Value-color-text Map	Defines the association value – Color/Text shown in the schedule. Use this option to define all possible values available in the BACNSched keypad. 

Value-Color Dialog

	Tag value	Mapped color	Text
1	1	#00aaff	Saving
2	2	#ffaa7f	Confort
3	3	#55ff7f	Normal

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	

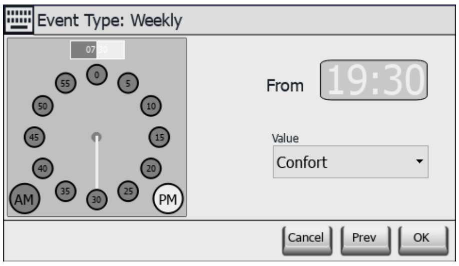
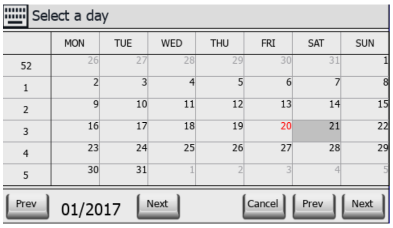
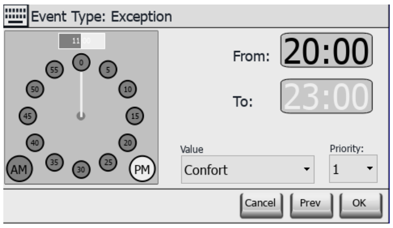
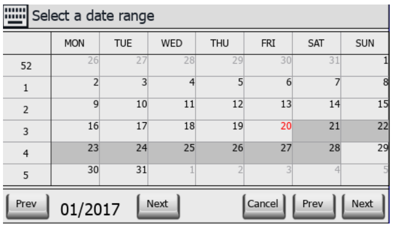
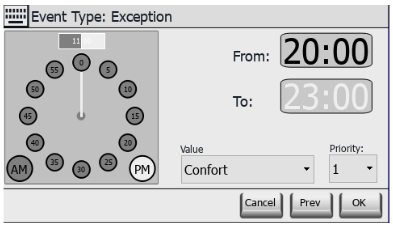
16/01/2017 - 22/01/2017

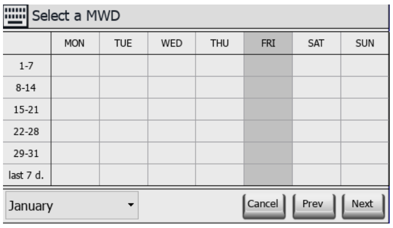
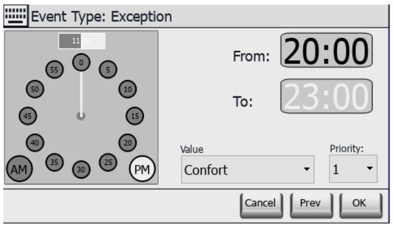
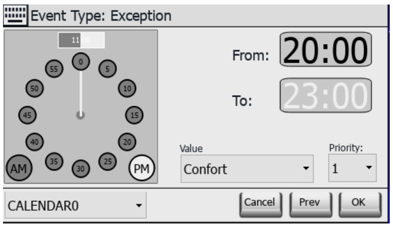
Use the < and > buttons to select the week to be displayed. The date of first day and last day of the week is shown.

Swing gesture can be used on the widget to select the date.

New

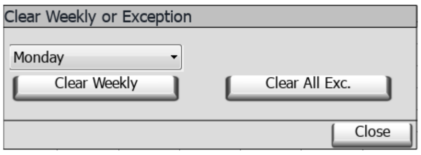
Press the button “New” to enter a new schedule item. The button is active only if the tag associated to Weekly Schedule or Exception Schedule has been configured as Read/Write.

Schedule item	Description
Weekly	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.  Event Type: Weekly From 19:30 Value: Confort
Exception Single	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.  Select a day 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  Event Type: Exception From: 20:00 To: 23:00 Value: Confort Priority: 1
Exception Range	Click on the first day and on the last day to select a range of days into the calendar. • 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.  Select a date range 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  Event Type: Exception From: 20:00 To: 23:00 Value: Confort Priority: 1

Schedule item	Description
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.  Select a MWD MON TUE WED THU FRI SAT SUN 1-7 8-14 15-21 22-28 29-31 last 7 d. January Cancel Prev Next  Event Type: Exception From: 20:00 To: 23:00 AM PM Value Confort Priority: 1 Cancel Prev OK
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.  Event Type: Exception From: 20:00 To: 23:00 AM PM Value Confort Priority: 1 CALENDAR0 Cancel Prev OK

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.



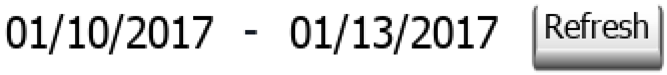
Refresh

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.

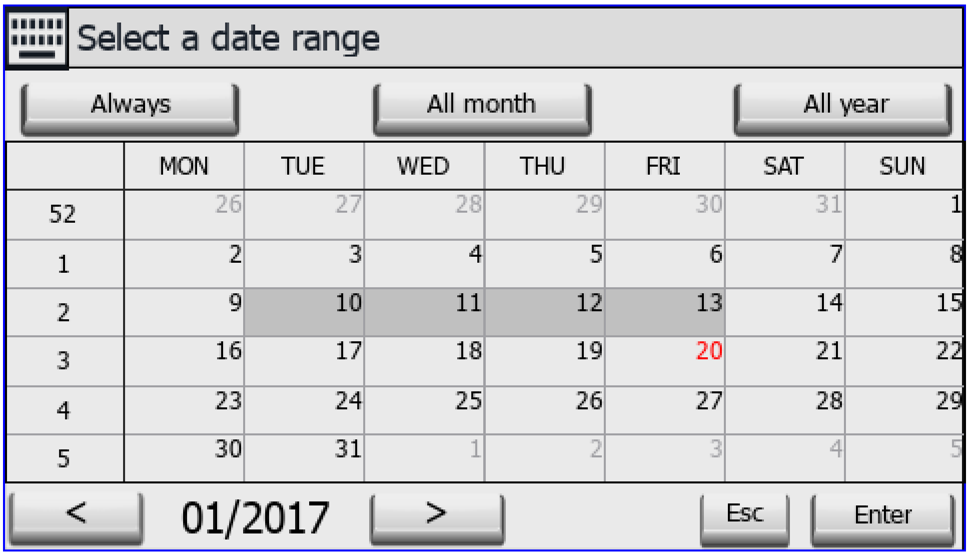
Property	Description
BACnet Effective_ Period	Attach to the Effective_Period tag of the Schedule object


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.



	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

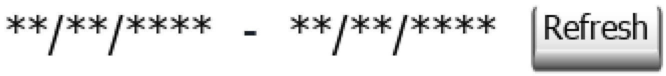
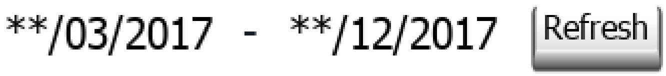
The keypad shows data for one month.

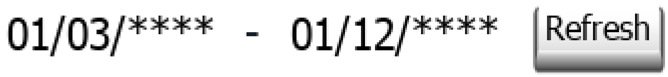
Use the < and > buttons to select the month to be displayed. The date of first day of the month is shown.

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
Always	The schedule will be always active.  **/**/**** - **/**/****
All Month	The selected period will be extended to all months.  **/03/2017 - **/12/2017
All Year	The selected period will be extended to all years.

Option	Description
	 01/03/**** - 01/12/****

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
BACNCal	Keypad for BACnet Calendar.
BACNDateRange	Keypad for BACnet Effective_Period.
BACNDefVal	Keypad for default value (embedded in the BACnet Schedule).
BACNSched	Keypad for BACnet Schedule. This keypad is context sensitive. It will show different options depending on the type of schedule.

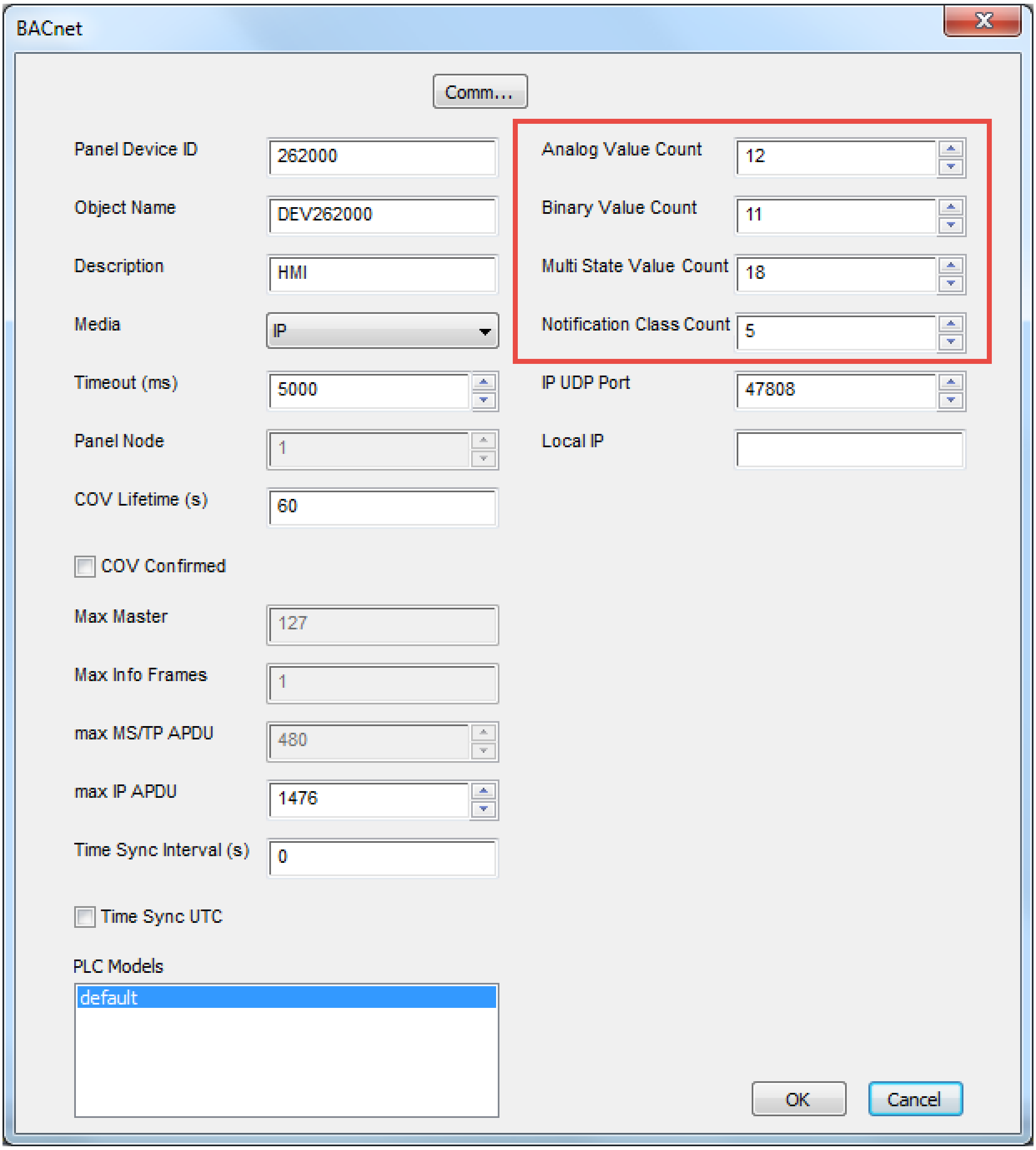
The system is configured to recall the appropriate keypad for each BACnet widget.

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

Object Name: DEV262000

Description: HMI

Media: IP

Timeout (ms): 5000

Panel Node: 1

COV Lifetime (s): 60

□ COV Confirmed

Max Master: 127

Max Info Frames: 1

max MS/TP APDU: 480

max IP APDU: 1476

Time Sync Interval (s): 0

□ Time Sync UTC

PLC Models: default

Analog Value Count: 12

Binary Value Count: 11

Multi State Value Count: 18

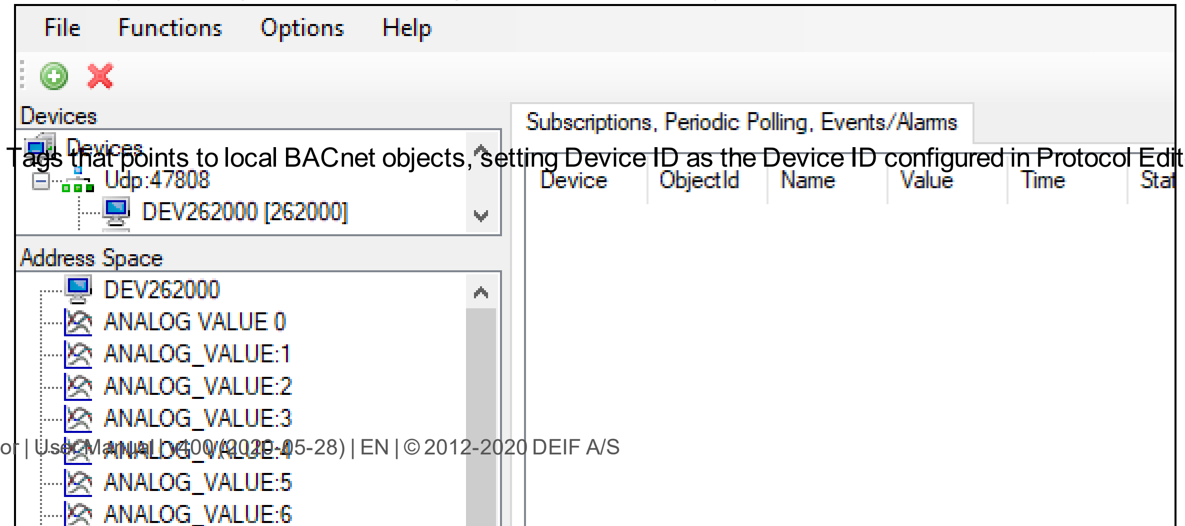
Notification Class Count: 5

IP UDP Port: 47808

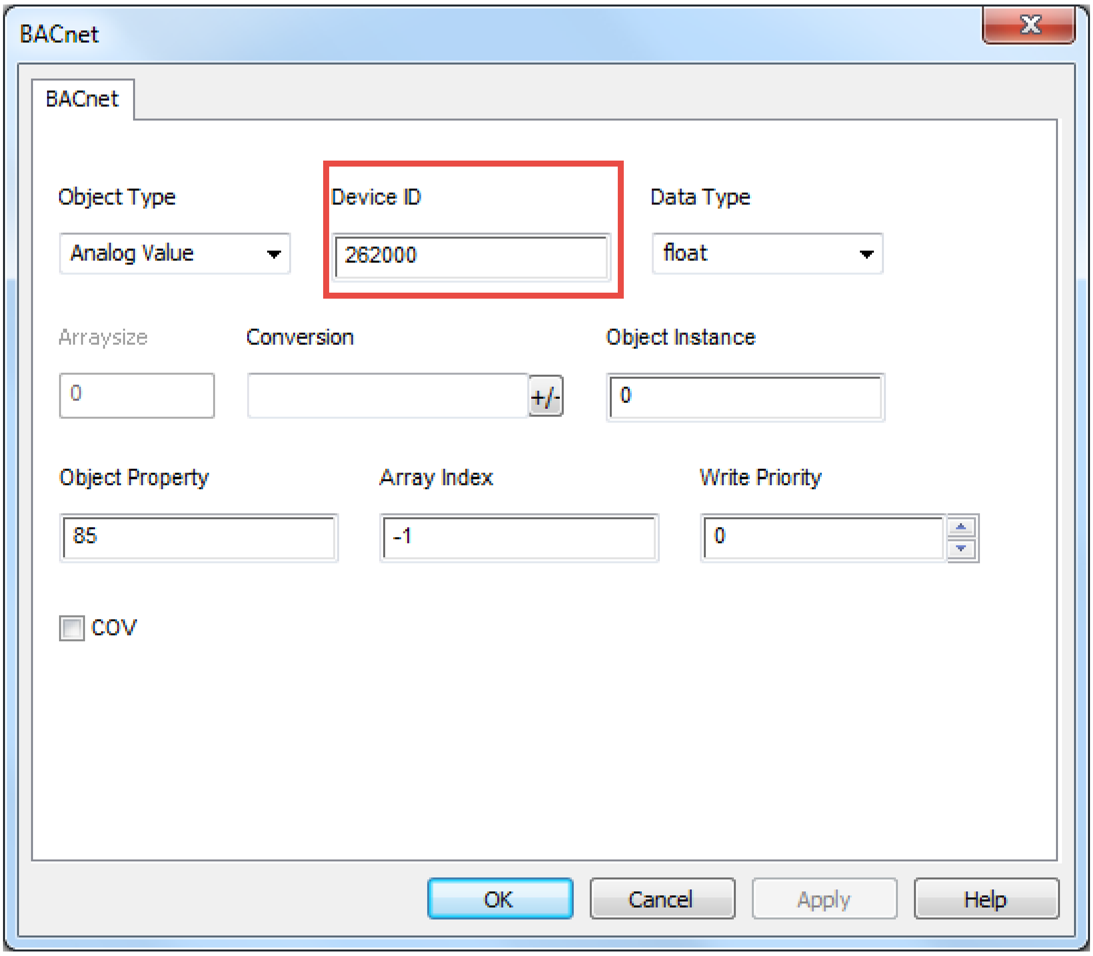
Local IP:

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:



Object Type: Analog Value

Device ID: 262000

Data Type: float

Arraysize: 0

Conversion:

Object Instance: 0

Object Property: 85

Array Index: -1

Write Priority: 0

□ COV

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
Location	58	Parameter	Yes		String

Property Name	Code	Default value	Permanent	Note	Data Type
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		Boolen(3)
COV increment	22	0	Yes		Float
Deadband	25	0	Yes		Float
Description	28	"ANALOG 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)
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
Event enable	35	0	Yes		Boolean(3)
Event state	36	0	Read-only		UnsignedInt

Property Name	Code	Default value	Permanent	Note	Data Type
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)
Event state	36	0	Read-only		UnsignedInt
Event time stamps	130		Yes		UnsignedInt(3)
Fault values	39		Yes	Defines number of array elements	UnsignedInt

Property Name	Code	Default value	Permanent	Note	Data Type
				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

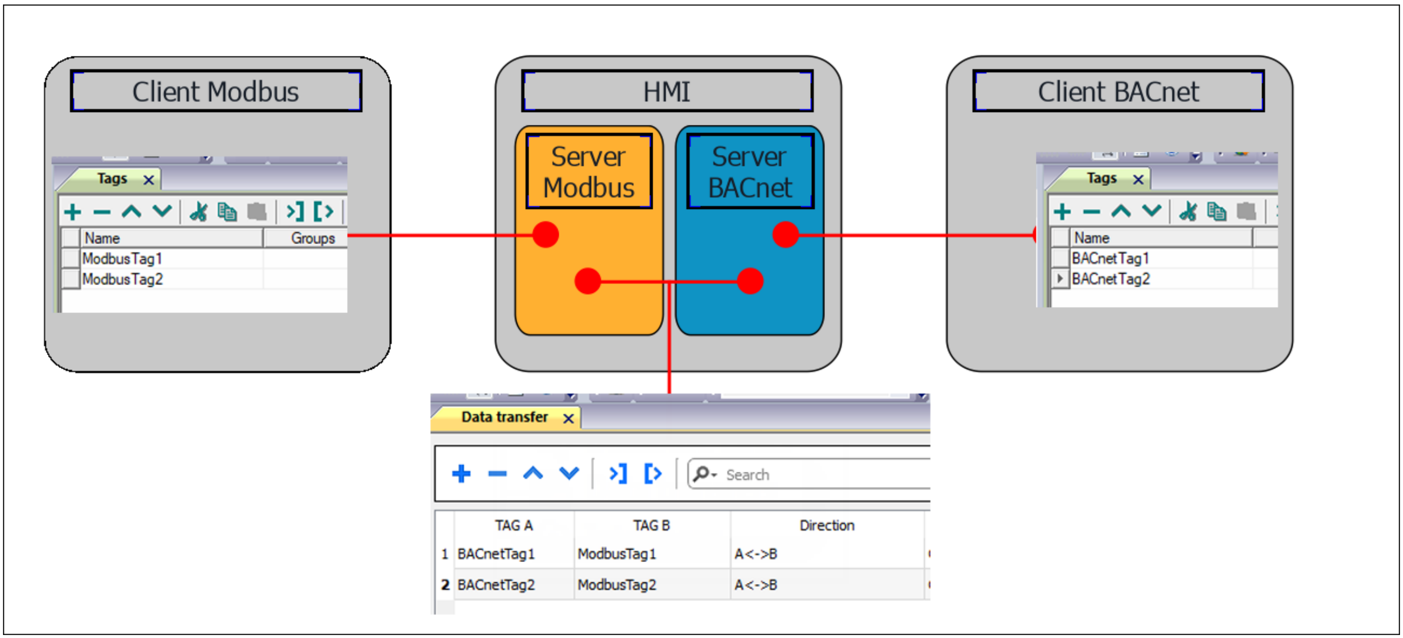
Notification Class objects description

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 + Conversion
Object name	77	"NOTIFICATION CLASS n"	Yes		String
Object type	79	15	Read-only		UnsignedInt
Priority	86	255,255,255	Yes		UnsignedInt(3)
Recipient list	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.



	TAG A	TAG B	Direction
1	BACnetTag1	ModbusTag1	A<->B
2	BACnetTag2	ModbusTag2	A<->B

JavaScript Interface

Beside Tag interface the user can access the protocol via JavaScript.

Although defined Tags can be accesses 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
scan (minID, maxID, <timeout>)	Executes a scan for devices in the given range.
scan_status	Get the scanning result.

Command	Description
devices	Get the list of devices.
objectCount (deviceID, objectType)	Get the object count of given object types in given device.
objectNames (start, count)	Get the part of object names asked by previous objectCount.
properties (deviceID, objectType, objectInstance)	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 ad 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"],"modelNames":["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 containts 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
Cannot bind to the device_id	Cannot establish communication with the Device ID provided for this tag.
Cannot read the property data type	The type of the property to write cannot be determined.
write conversion error	A conversion associated to this tag has failed.
Cannot write ICOM type …. BACnet type ….	A datatype selected for this tag is not compatible with the BACnet property to set.
Timeout on COV subscription	A request for COV subscription for this tag has timed out.
Timeout on waiting COV update	A COV notification has not been received for this tag whithin timeout.

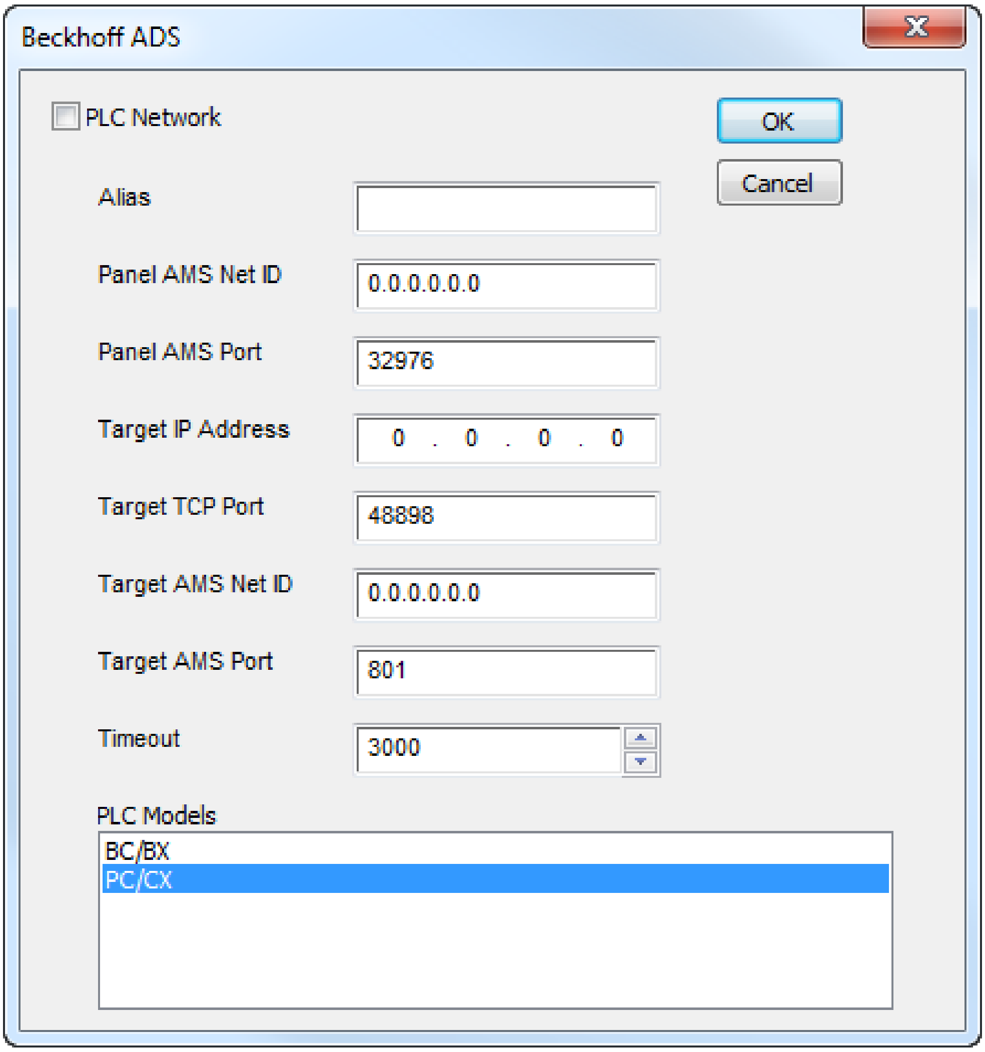
Error	Cause
Can't get COV for this property	The selected property for COV notification is unsupported.
datagramItem conversion error	A conversion associated to a tag that is part of a datagram has failed.
Timeout waiting on response	No response for a request of read or write property within timeout.
datagram element …., no data available	No data available for a tag that is part of datagram.
datagram element …., Unsupported BACnet data type	Read datagram element is of unsupported BACnet type.
datagram element …., can't convert BACnet type to ….	A Data Type selected for a tag which is part of a datagram is not compatible with the BACnet property to read.
No data in response	No data available for a tag.
Datagram element 'element_URI' error: 'error_class': error_code	The reading of indicated datagram element 'element_URI' was reported as error. The error descriptions **error_class** and **error_code** are included in the message.
datagram object does not match	The object of the received datagram item does not match the asked object.
datagram property does not match	The property of the received datagram item does not match the asked property.
BACnet abort: reason_of abort	BACnet abort message was received. The reason of abort is given.
BACnet reject: reason_of_rejection	BACnet reject message was received. The reason of rejection is given.
BACnet error: error_class: error_code	BACnet error message was received. The error description is given as combination of **error_class** and **error_code**.
parameter 'parameter_name' out of range	The protocol parameter **parameter_name** value is out of range.

Beckhoff ADS

Beckhoff ADS protocol driver is used for communication with Beckhoff controllers through Ethernet connection. This implementation of Beckhoff ADS protocol driver is based on the information published by Beckhoff.

Protocol Editor Settings

Add (+) a driver in the Protocol editor and select the protocol “Beckhoff ADS” from the list of available protocols.



□ PLC Network

Alias:

Panel AMS Net ID: 0.0.0.0.0.0

Panel AMS Port: 32976

Target IP Address: 0 . 0 . 0 . 0

Target TCP Port: 48898

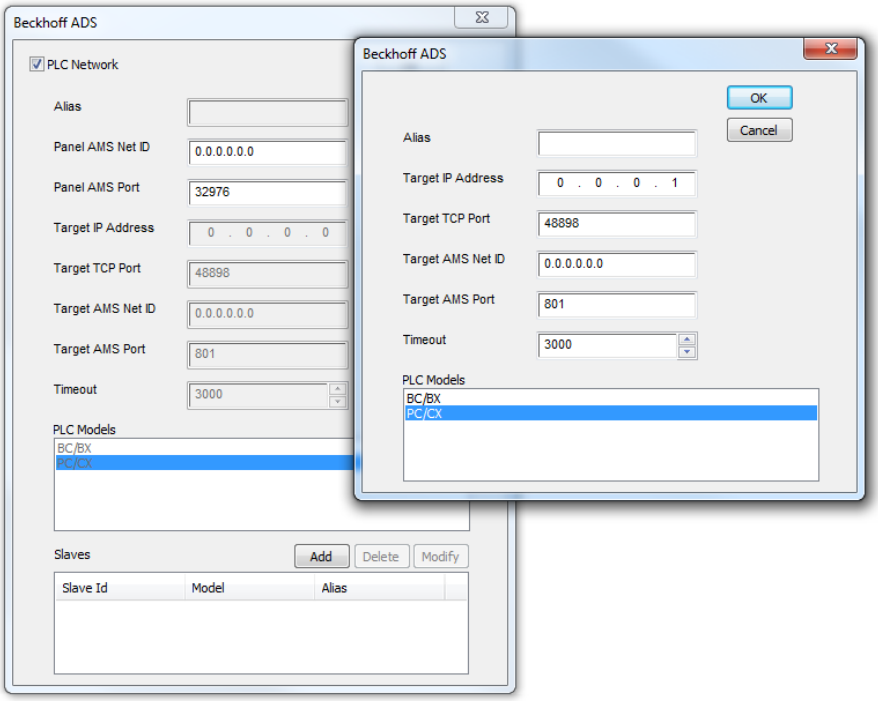
Target AMS Net ID: 0.0.0.0.0.0

Target AMS Port: 801

Timeout: 3000

PLC Models: BC/BX, PC/CX (selected)

Element	Description
Alias	Name to be used to identify nodes in the plc network configuration. The name will be added as a prefix to each tag name imported for each network node.
Panel AMS Net ID	Specifies the AMS net ID of the panel; the first 4 bytes must match the panel IP address assigned to the HMI device. If panel has IP address 192.168.10.100 then AMS Net ID could be 192.168.10.100.1.1
Panel	Specifies the panel AMS port number to be used on panel.

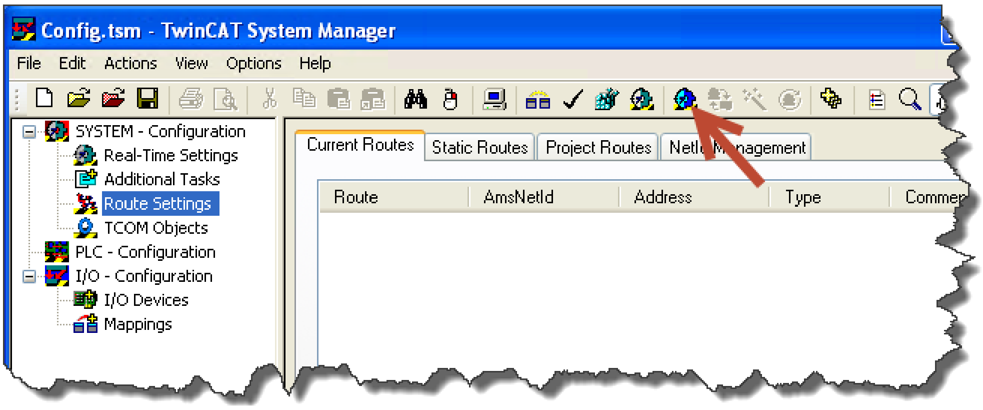
Element	Description
AMS Port	Using TwinCAT2, default Panel AMS Port is 32976. Using TwinCAT3, default Panel AMS Port is 32844.
Target IP Address	Specifies the IP address of the target controller.
Target AMS Net ID	Specifies the Target AMS net ID of the target controller.
Target AMS Port	Specifies the port number dedicated to the communication on target device. Using TwinCAT2, default Target AMS Port is 801. Using TwinCAT3, default Target AMS Port is 851.
Timeout	The number of milliseconds between retries when communication fails.
PLC models	Select the model which corresponds to the device to be connected. Model selection is very important to be set properly.
PLC Network	The protocol allows the connection of multiple controllers to one operator panel. To set-up multiple connections, check “PLC network” checkbox and enter the Target Controller settings for  Beckhoff ADS ☑ PLC Network Alias: Panel AMS Net ID: 0.0.0.0.0.0 Panel AMS Port: 32976 Target IP Address: 0 . 0 . 0 . 0 Target TCP Port: 48898 Target AMS Net ID: 0.0.0.0.0.0 Target AMS Port: 801 Timeout: 3000 PLC Models: BC/BX, PC/CX Slaves: Add, Delete, Modify Slave Id, Model, Alias Beckhoff ADS OK, Cancel Alias: Target IP Address: 0 . 0 . 0 . 1 Target TCP Port: 48898 Target AMS Net ID: 0.0.0.0.0.0 Target AMS Port: 801 Timeout: 3000 PLC Models: BC/BX, PC/CX every node.

TwinCAT2 Route Settings

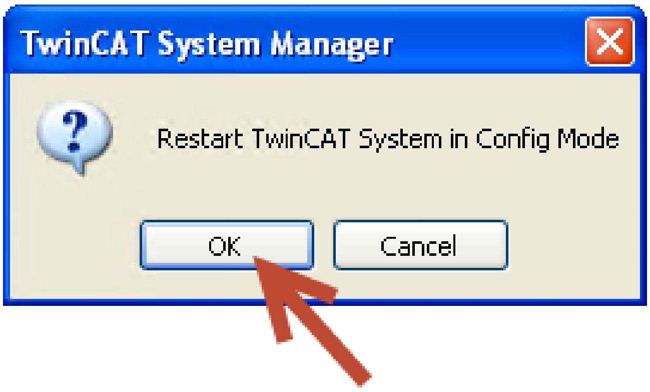
Beckhoff controllers require some specific settings to allow connection from HMI devices.

In TwinCAT2 System Manager you need to configure Static Route.

First of all the system must be reset in Configuration Mode using the toolbar button as showed in the following figure.



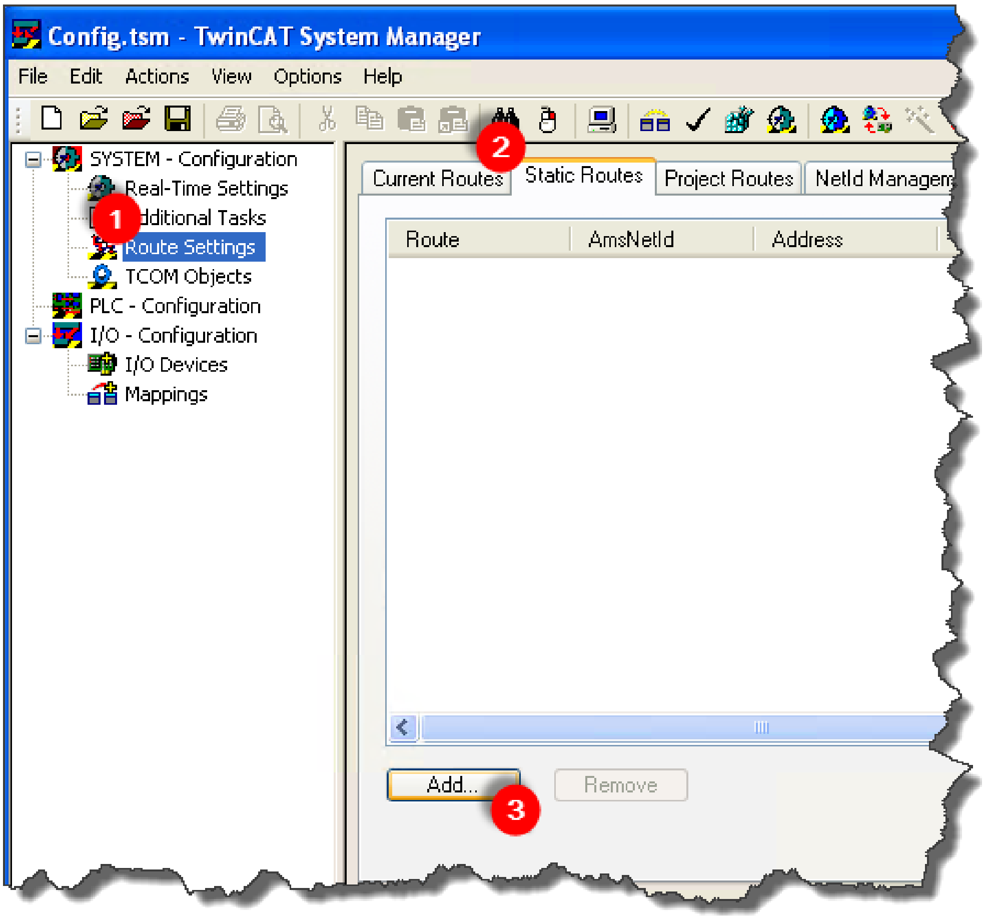
Then confirm to Restart TwinCAT2 System in Config Mode as in the figure below.


TwinCAT System Manager

Restart TwinCAT System in Config Mode

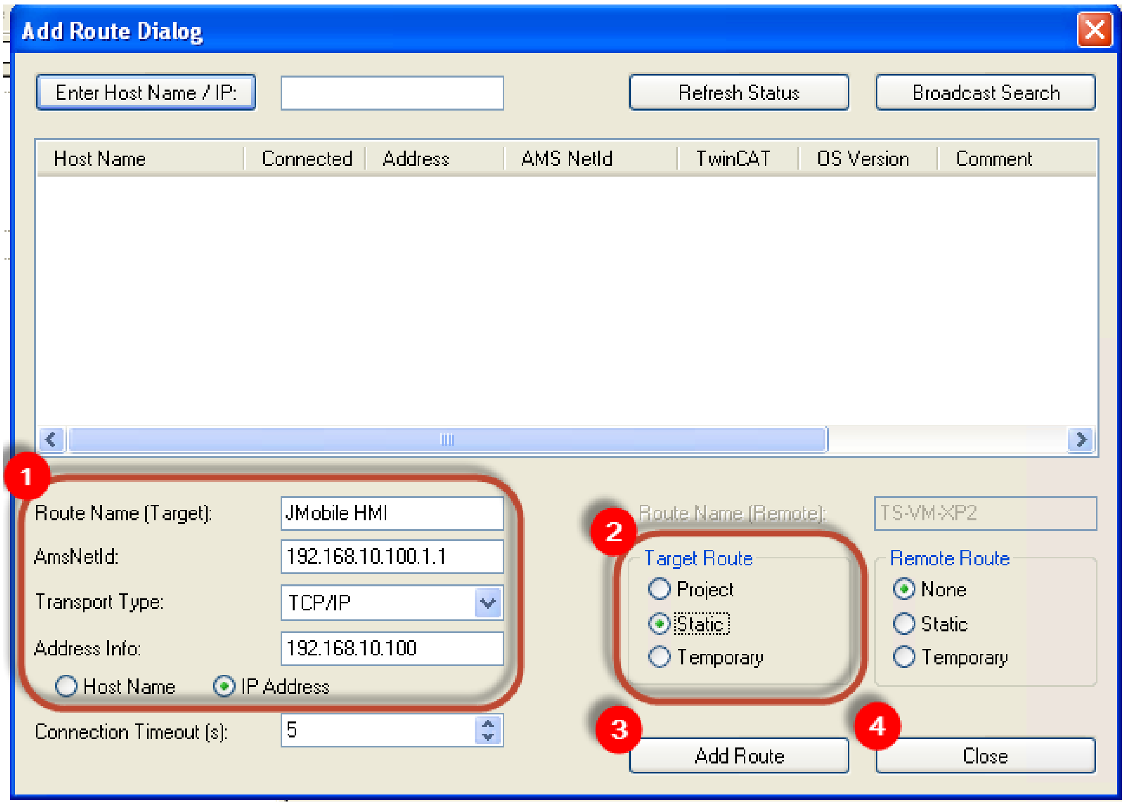
Once restarted, as in the next figure, follow these steps to add a new Route:

1. Open Route Settings.
2. Select Static Routes tab.
3. Click on [Add] button.



Into Add Route Dialog user must set:

1. Route Name: a name useful to indentify the Route i.e. “HMI”, AmsNetId: The Panel AMS Net ID as configured into Beckhoff ADS protocol, Transport Type: TCP/IP.
   Address Info: Type in the Panel IP Address with “IP Address” option selected.
2. Target Route: Static.
3. Click on [Add Route] button. Note: no warning or message will be shown.
4. Click on [Close] button.



Route Name (Target): JMobile HMI

AmsNetId: 192.168.10.100.1.1

Transport Type: TCP/IP

Address Info: 192.168.10.100

○ Host Name ● IP Address

Connection Timeout (s): 5

Route Name (Remote): TS-VM-XP2

Target Route: ○ Project ● Static ○ Temporary

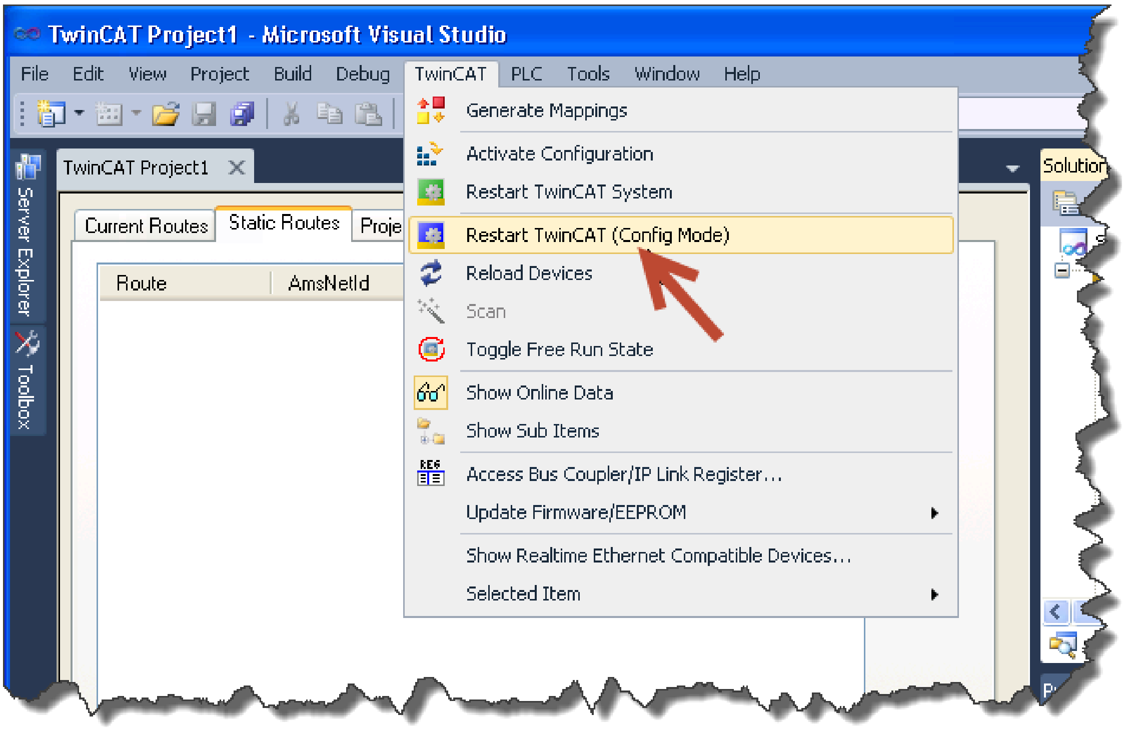
Remote Route: ● None ○ Static ○ Temporary

Then the route will appear under Static Routes list.

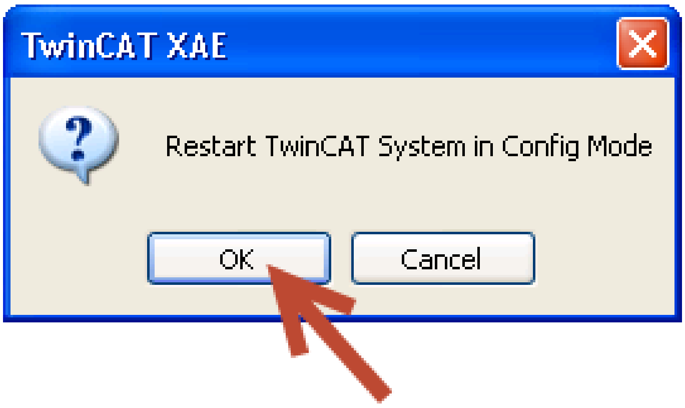
TwinCAT3 Route Settings

Beckhoff controllers require some specific settings to allow connection from HMI devices. In TwinCAT3 XAE you need to configure a Static Route.

First of all TwinCAT3 system must be reset in Configuration Mode using the toolbar button as showed in the following figure.

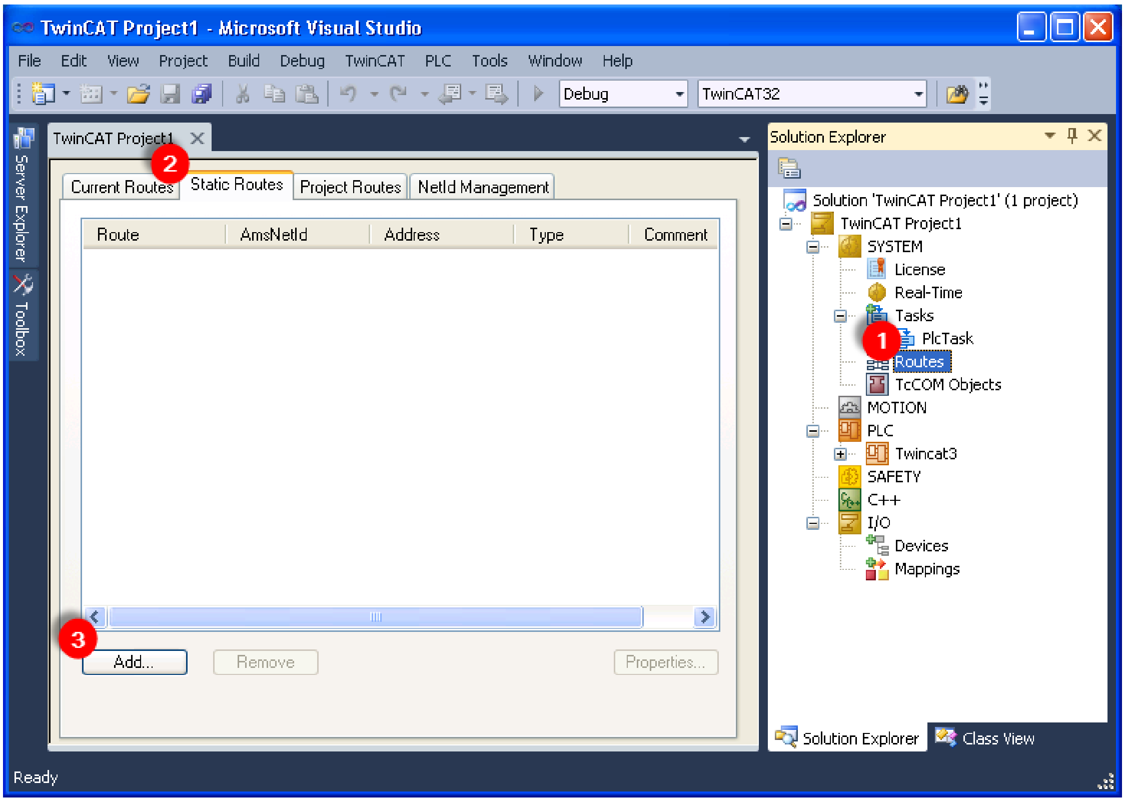


Then confirm to Restart TwinCAT3 System in Config Mode.



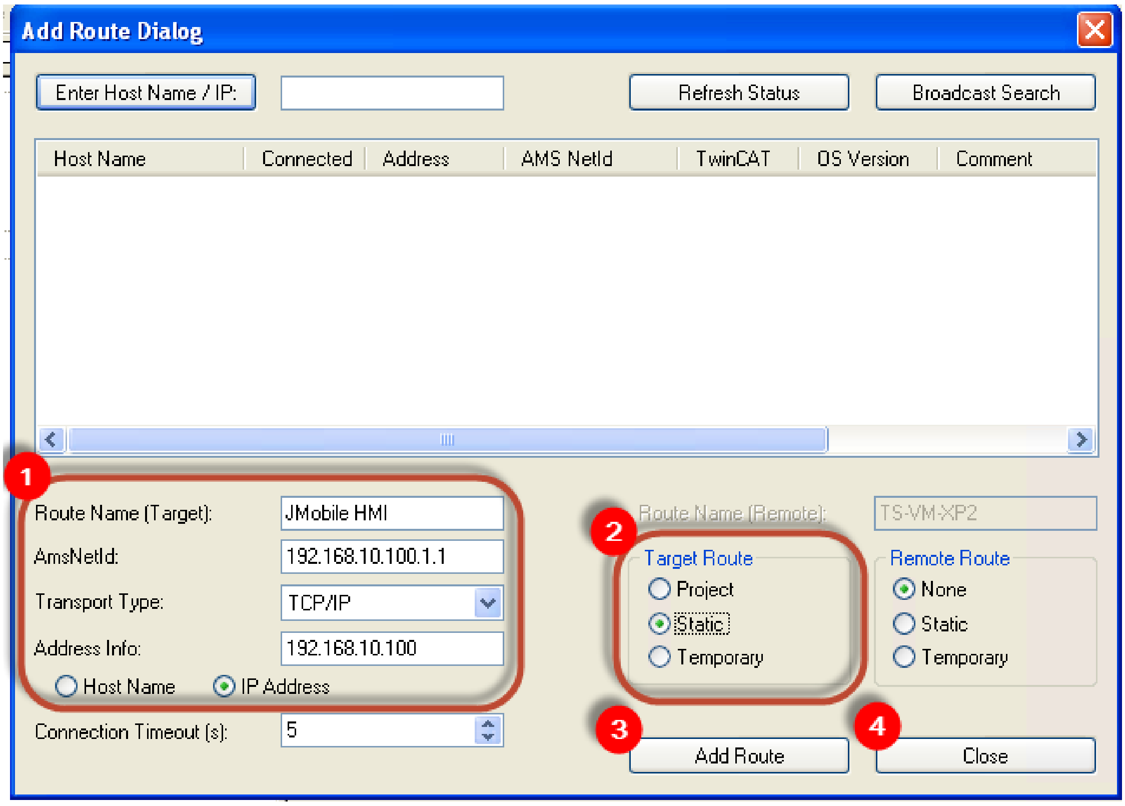
Once restarted, as in the next figure, follow these steps to add a new Route:

1. Open Routes.
2. Select Static Routes tab.
3. Click on [Add] button.



Into Add Route Dialog user must set:

1. Route Name: a name useful to indentify the Route i.e. “HMI”, AmsNetId: The Panel AMS Net ID as configured into Beckhoff ADS protocol, Transport Type: TCP/IP.
   Address Info: Type in the Panel IP Address with “IP Address” option selected.
2. Target Route: Static.
3. Click on [Add Route] button. Note: no warning or message will be shown.
4. Click on [Close] button.



Route Name (Target): JMobile HMI

AmsNetId: 192.168.10.100.1.1

Transport Type: TCP/IP

Address Info: 192.168.10.100

○ Host Name ● IP Address

Connection Timeout (s): 5

Route Name (Remote): TS-VM-XP2

Target Route: ○ Project ● Static ○ Temporary

Remote Route: ● None ○ Static ○ Temporary

Then the route will appear under Static Routes list.

Tag Import

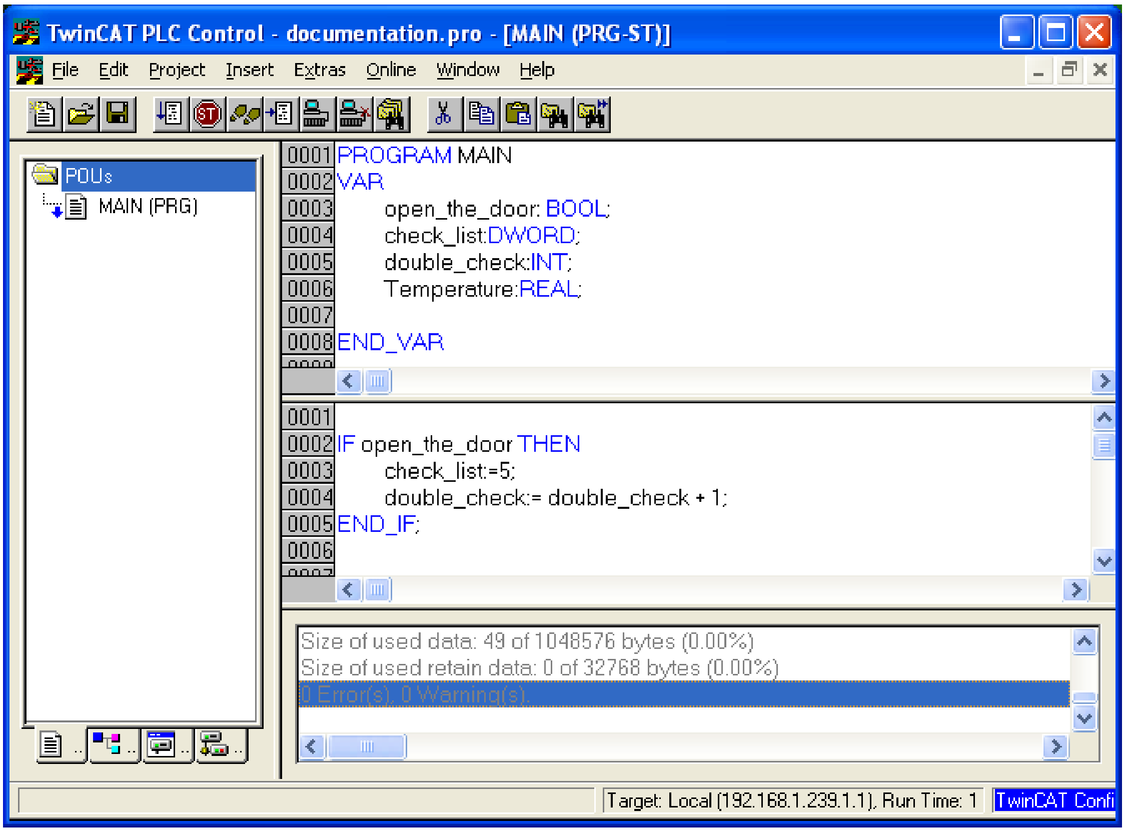
Exporting Tags from PLC

The data in the Beckhoff system is based on tags.

The organization of the internal memory of the controller is not fixed but it is configured by the user at development time. Each data item can be identified by a string called “tag”.

The TwinCAT development environment generates the list of tags created for each controller in the configuration of the application.

The project in the panel must refer to the tag names assigned in the TwinCAT PLC Control programming software at development time. The Designer Tag Editor supports direct import of the tag file generated by the Beckhoff software.



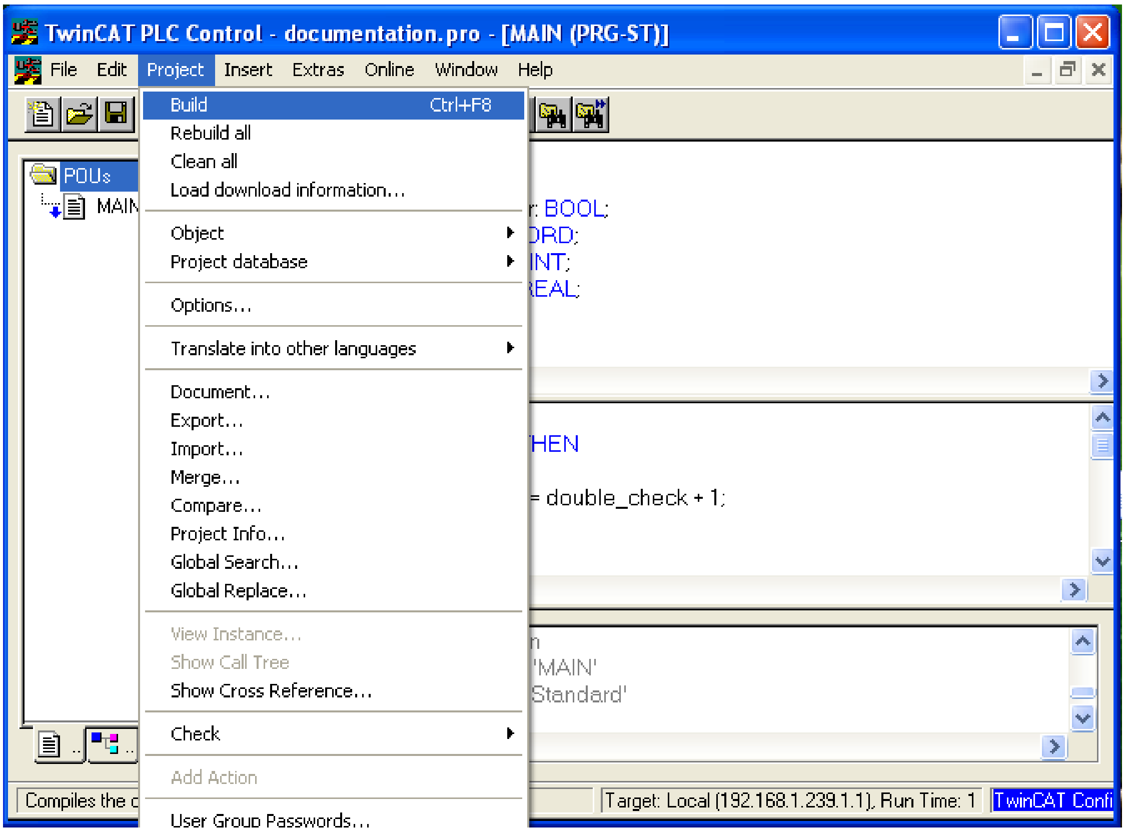
```
PROGRAM MAIN
VAR
    open_the_door: BOOL;
    check_list:DWORD;
    double_check:INT;
    Temperature:REAL;

END_VAR
```

```

IF open_the_door THEN
    check_list:=5;
    double_check:= double_check + 1;
END_IF;

```

```
Size of used data: 49 of 1048576 bytes (0.00%)
Size of used retain data: 0 of 32768 bytes (0.00%)
0 Error(s), 0 Warning(s).
```

Target: Local (192.168.1.239.1.1), Run Time: 1 TwinCAT Confi

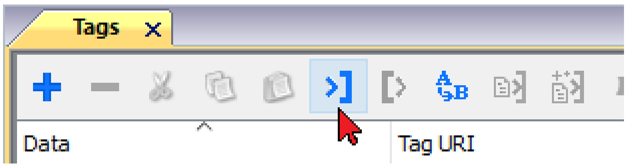
To export tags defined for the selected controller, click on Project > Build as shown.



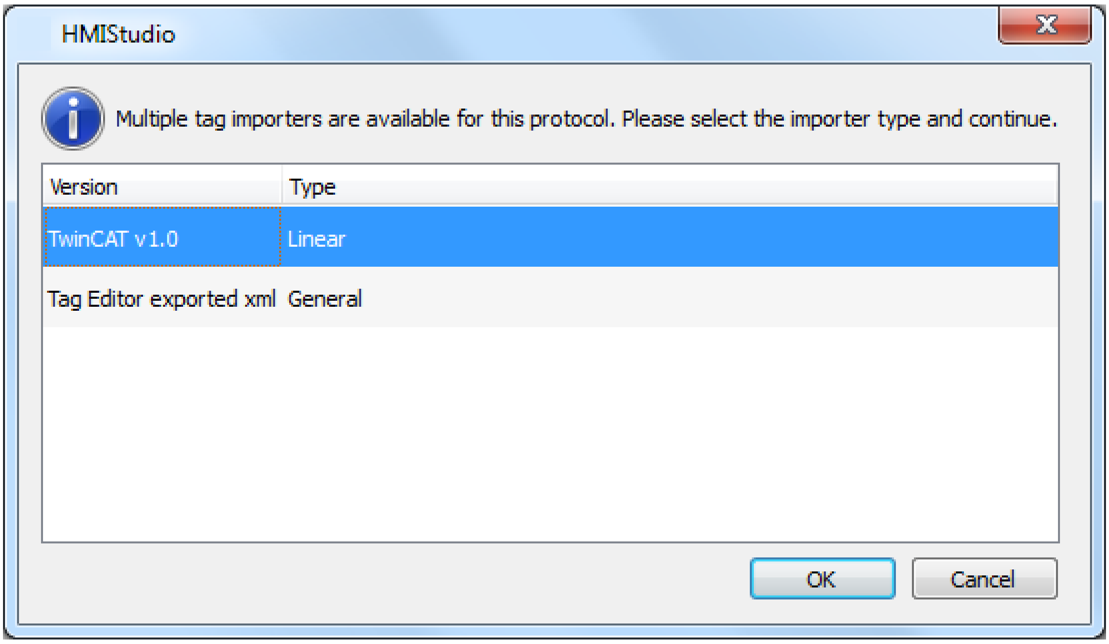
The TwinCAT PLC Control software will create a file with extension TPY.

Importing Tags in Tag Editor

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



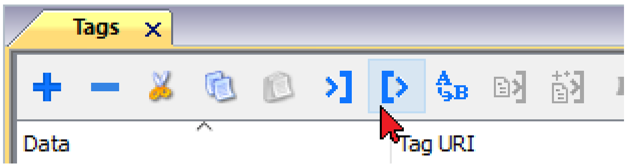
The following dialog shows which importer type can be selected.



HMIStudio

Multiple tag importers are available for this protocol. Please select the importer type and continue.

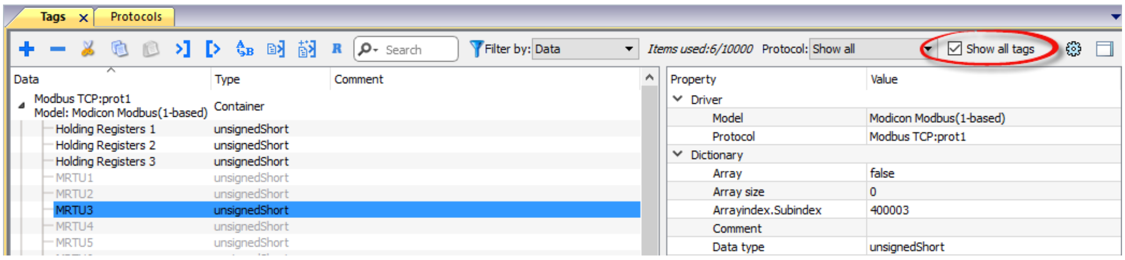
Version	Type
TwinCAT v1.0	Linear
Tag Editor exported xml	General

Importer	Description
TwinCAT v1.0 Linear	Requires a **.tpy** file. 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. 

Note: the Beckhoff driver supports direct access to the PLC tags using the handles; this means that if no tags are added to the PLC and the PLC program is just re-compiled, you do not need to re-import tags as the access to them does not depend from the offset, but only from name.

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.

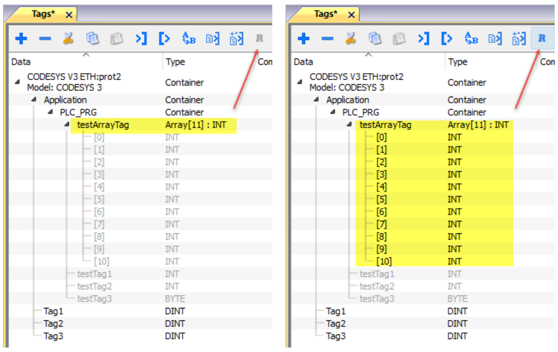


Tags × | Protocols

Search | Filter by: Data | *Items used:6/10000* | Protocol: Show all | ☑ Show all tags

Data	Type	Comment
Modbus TCP:prot1 Model: Modicon Modbus(1-based)	Container	
Holding Registers 1	unsignedShort	
Holding Registers 2	unsignedShort	
Holding Registers 3	unsignedShort	
MRTU1	unsignedShort	
MRTU2	unsignedShort	
MRTU3	unsignedShort	
MRTU4	unsignedShort	
MRTU5	unsignedShort	

Property	Value
Driver	
Model	Modicon Modbus(1-based)
Protocol	Modbus TCP:prot1
Dictionary	
Array	false
Array size	0
Arrayindex.Subindex	400003
Comment	
Data type	unsignedShort

Toolbar item	Description
	Import Tag(s). Select tags to be imported and click on this icon to add tags from tag dictionary to the project
	Update Tag(s). Click on this icon to update the tags in the project, due a new dictionary import.
R	Check this box to import all sub-elements of a tag. Example of both checked and unchecked result:  Tags* × Data / Type / Con CODESYS V3 ETH:prot2 Model: CODESYS 3 / Container Application / Container PLC_PRG / Container testArrayTag / Array[11] : INT [0] / INT [1] / INT [2] / INT [3] / INT [4] / INT [5] / INT [6] / INT [7] / INT [8] / INT [9] / INT [10] / INT testTag1 / INT testTag2 / INT testTag3 / BYTE Tag1 / DINT Tag2 / DINT Tag3 / DINT Tags* × Data / Type / Con CODESYS V3 ETH:prot2 Model: CODESYS 3 / Container Application / Container PLC_PRG / Container testArrayTag / Array[11] : INT [0] / INT [1] / INT [2] / INT [3] / INT [4] / INT [5] / INT [6] / INT [7] / INT [8] / INT [9] / INT [10] / INT testTag1 / INT testTag2 / INT testTag3 / BYTE Tag1 / DINT Tag2 / DINT Tag3 / DINT
Search Filter by: Tag name	Searches tags in the dictionary basing on filter combo-box item selected.

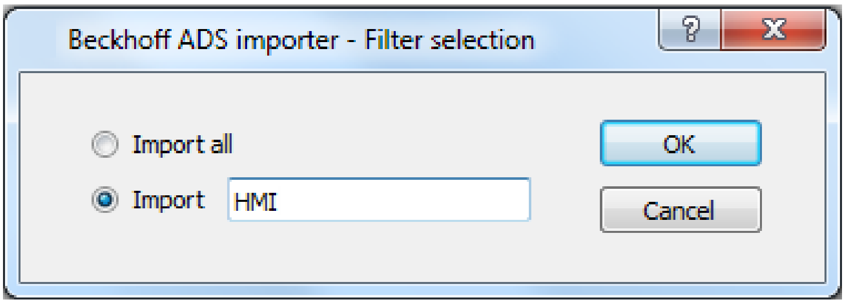
Using TwinCAT v1.0 Import Filter

When importing tags, the user can decide to import all the tags from the **.tpy** file or apply a filter importing only a subset of them.

The figure below shows how to specify the filter. The filter consist in a string (no wildcards are supported). The import filter will import only the tags having the specified string in the description.

If the description is applied to an “instance declaration” of a Function Block, all the tags within the block will be imported.

If the string is contained only as comment of some variables inside the Function Block, only that variables will be imported.



○ Import all
● Import HMI

As an example for the use of the import filter, please see the following case.

```
FUNCTION_BLOCK FB_Motor
VAR_INPUT
     bStartMotor: BOOL;
     bReset: BOOL;
END_VAR
VAR_OUTPUT
     bMotorOn: BOOL;
     bAlarm: BOOL;(* HMI Thermal alarm *)
END_VAR
VAR
     sData: STRING;
     bResetStatistics: BOOL; (* HMI Reset statistics *)
END_VAR
VAR PERSISTENT
     stStat: ST_MotorStats; (* HMI Motor statistics *)
END_VAR

Function block instances declaration:
VAR
     fbMotor1: FB_Motor;
     fbMotor2: FB_Motor; (* HMI only show Motor 2!! *)
END_VAR
```

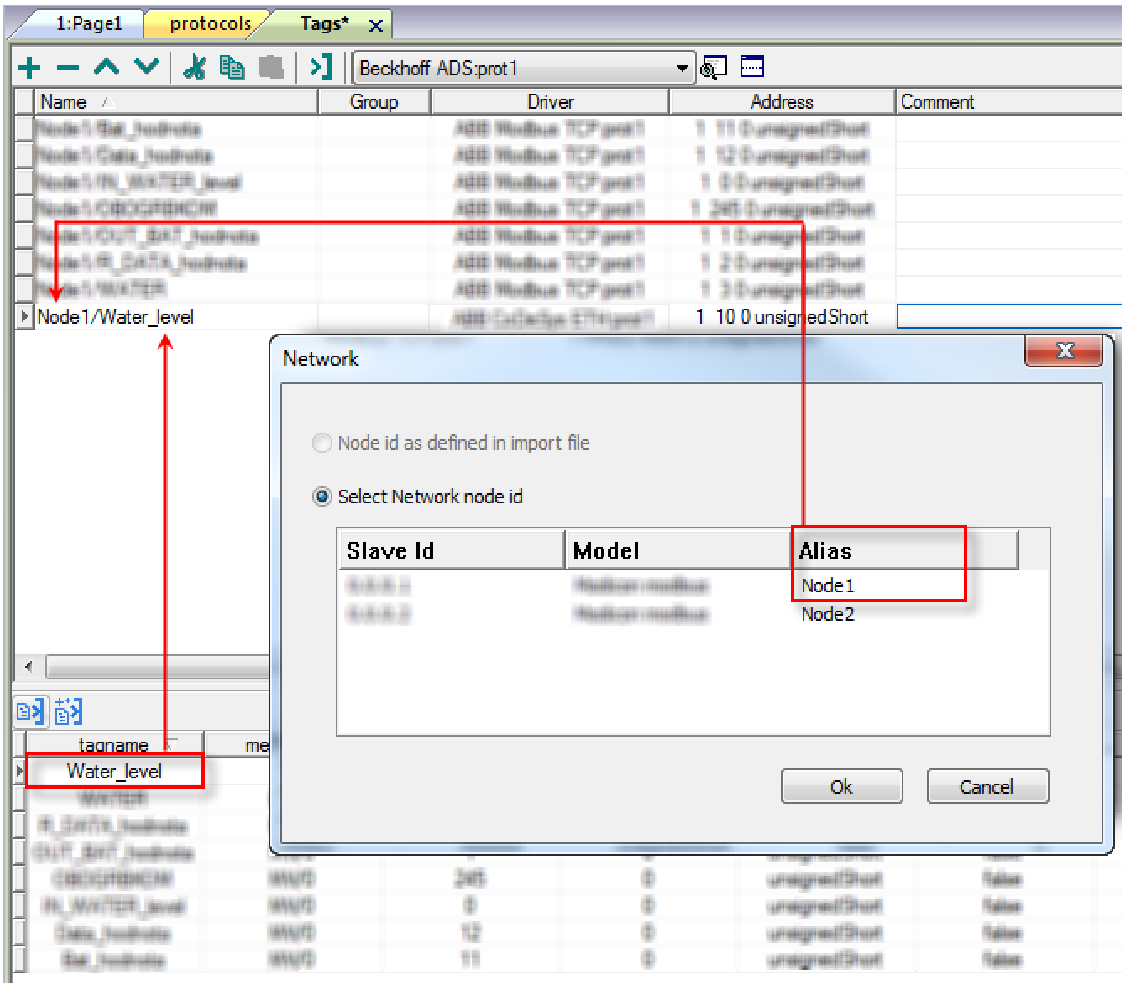
The following tags will be imported:

- MAIN/fbMotor2/bAlarm
- MAIN/fbMotor2/bResetStatistics
- MAIN/fbMotor2/ST_MotorStats

Aliasing Tag Names in Network Configurations

Tag names must be unique at project level; it often happens that the same tag names are to be used for different controller nodes (for example when the HMI is connected to two devices that are running the same application). Since tags include also the identification of the node and Tag Editor does not support duplicate tag names, the import facility in Tag Editor has an aliasing feature that can automatically add a prefix to imported tags. With this feature tag names can be done unique at project level.

The feature works when importing tags for a specific protocol. Each tag name will be prefixed with the string specified by the “Alias”. As shown in the figure below, 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.



Network

○ Node id as defined in import file

● Select Network node id

Slave Id	Model	Alias
[illegible]	[illegible]	Node1
[illegible]	[illegible]	Node2

Note: Aliasing tag names is only available when tags can be imported. 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 to the tag name only at the moment the tags are imported using Tag Editor. 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 imported again, all tags will be imported again with the new prefix string.

Communication Status

The current communication status can be displayed using the dedicated system variables. Please refer to the User Manual for further information about available system variables and their use.

The codes supported for this communication driver are:

Error	Notes
NAK	Returned in case the controller replies with a not acknowledge
Timeout	Returned when a request is not replied within the specified timeout period; ensure the controller is connected and properly configured to get network access
Invalid response	The panel did receive from the controller a response, but its format or its contents is not as expected; ensure the data programmed in the project are consistent with the controller resources
General Error	Error cannot be identified; should never be reported; contact technical support

CANopen HMI

The CANopen HMI communication driver has been designed to connect HMI products to a CANopen network. A new device communication profile has been developed for the HMI. This profile takes advantage from the advanced user interface features of the products, while retaining the simple networking concept supported by the CANopen network.

The basic idea is create a client/server communication structure where the HMI is the client and the CANopen controller is the server.

Connection to CANopen network requires the optional CANopen communication module. Verify the suitable version for your HMI model.

Please note that changes in the controller protocol or hardware, which may interfere with the functionality of this driver, may have occurred since this documentation was created. Therefore, always test and verify the functionality of the application. To accommodate developments in the controller protocol and hardware, drivers are continuously updated. Please ensure that the latest driver is used in the application.

CANopen HMI Profile

In this communication model the HMI initiates the communication sessions, acting as a source of messages.

The basic messages are PDO messages with the standard size of 8 bytes.

The COB-ID of the messages is defined in a way that makes clear, from the well-known CANopen rules, what is the target of the PDO message.

The format of the PDO message has been defined according to a custom application layer protocol. This application layer protocol defines a device-independent communication profile optimized for HMI applications.

When the CANopen master controller receives the PDO message, it will interpret its contents and produce a PDO message with the response addressed to the HMI device.

The definition of this client/server relationship is independent of the CANopen Master in the sense that it can easily be supported in any particular CANopen master system. The resulting solution is easily portable to any CANopen master.

The software IDE offers a user interface that adapts itself to show the typical addressing model of CANopen master controller where the panel is going to be connected.

Adapting to different masters is possible using a profile customization file that may contain data definitions for different controller types.

Profile Details

This chapter provides the specification of the HMI profile and describes the subset of the request/response formats used by this implementation of the protocol.

The communication driver in the HMI generates PDO messages initiating communication request sessions as soon as the HMI runtime requires data from the protocol.

The panel is using the first transmit PDO identified by the COB-ID 0x180 combined with the Node Number assigned to the panel.

The communication profile uses only one transmit PDO and one receive PDO; the limited number of bytes available in standard PDO message maybe limiting, in some cases, the driver capabilities especially in terms of performance.

Request Format: HMI to Controller (Transmit PDO)

The PDO message transmitted by the HMI is formatted according to the following table.

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Offset Low	Offset High	Data 0	Data 1	Data 2	Data 3	Data Length and Job Number	Operation Type and Controller ID

The request frame includes the following elements:

Offset Low	Low byte of the offset (16 bits address) for the requested block of data
Offset High	High byte of the offset (16 bits address) for the requested block of data
Data 0 … Data 3	Data for Write Operations; not used in Read Operations
Data Length and Job Number	Contains: • number of requested bytes • job Number indicator;
Operation Type and Controller ID	Contains: • type of operation requested • the Controller ID that identifies the target of the message;

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Data Length [1]	Data Length [0]	Job Number [5]	Job Number [4]	Job Number [3]	Job Number [2]	Job Number [1]	Job Number [0]

The “Data Length” parameter is coded in 2 bits and takes values between 1 and 4 according to the following rules:

00	1 bytes
01	2 bytes
10	3 bytes
11	4 bytes

Note that the elementary size of each data item depends on the Controller memory organization.

The “Job Number” occupies 6 bits and can have values between 0 and 63; the “Job number” parameter is placed as last element in the PDO to ensure data consistency; the PLC program running the controller should constantly monitor the value of the “Job Number” parameter and consider the received message as valid only when detecting a change in the value of the “Job Number” field. “Job Number” is automatically increased at each new communication session (new request frame).

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Operation Type	Controller ID [6]	Controller ID [5]	Controller ID [4]	Controller ID [3]	Controller ID [2]	Controller ID [1]	Controller ID [0]

The “Operation Type” uses one bit with the following definition:

0	Read	data is transferred from controller
1	Write	data is transferred to controller

The “Controller ID” uses 6 bits; it represents the Node Number in the CANopen network of the master controller addressed by the current request.

This parameter is required in case the CAN network has more than one master controller; the CANopen standard defines in fact the COB-ID of the messages in a way that all the partners of the bus known the originator. In case more than one master device is present in the same network, the “Controller ID” field will specify the target of each individual request message. Only the master controller that recognizes in this field its own Node ID will consider the message and process the PDO contents.

Response Format: Controller to Panel (Receive PDO)

The PDO message returned by the controller must be formatted as defined in the following table.

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Status Flag / Error Code	Dummy – Always 0	Data 0	Data 1	Data 2	Data 3	Data Length and Job Number	Operation Type and Controller ID

The request frame consists of the following elements:

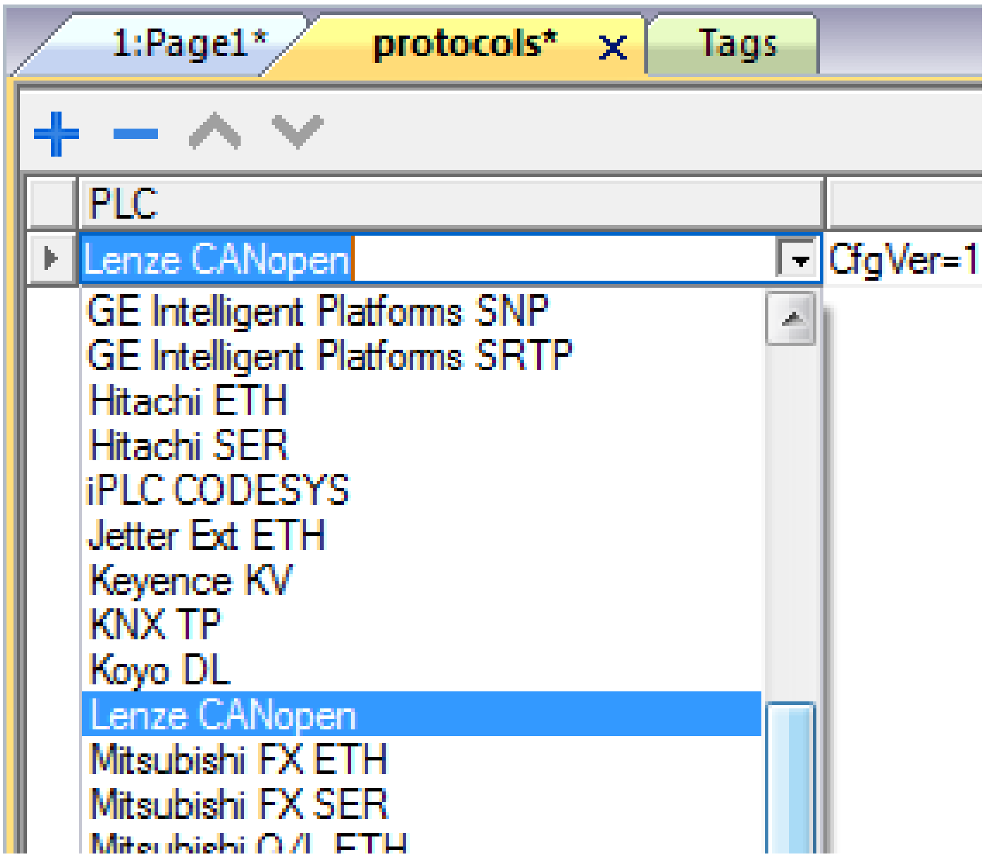
Status Flag / Error Code	Contains the information related to the execution of the operation type of the request; the next table shows the coding information
Data 0 … Data 3	Contain the data information returned to the panel in response to a Read request
Data Length and Job Number	It is the copy of the corresponding field of the request frame
Operation Type and Controller ID	It is the copy of the corresponding field of the request frame

	Status Flag / Error Code	
Operation Type in the Request Frame	**No Errors**	**Error**
Read	0x01	0x81
Write	0x02	0x82

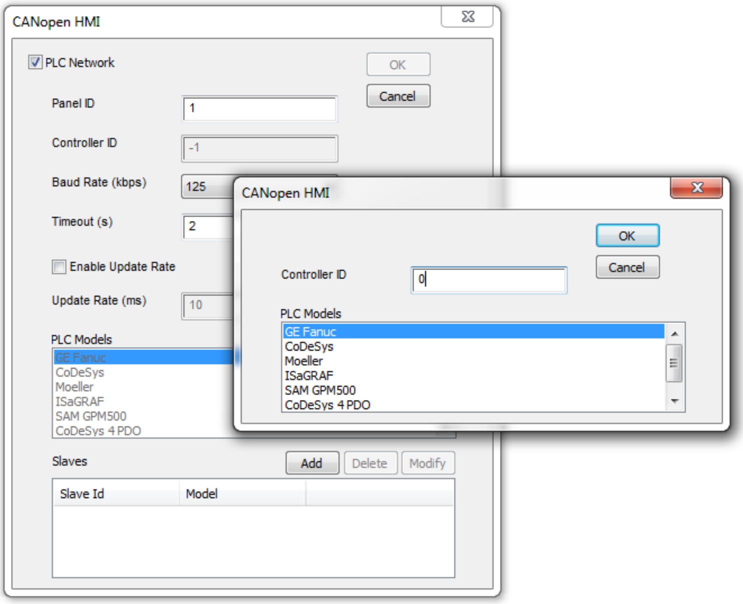
Protocol Editor Settings

Add (+) a driver in the Protocol editor and select the protocol called “CANopen HMI” from the list of available protocols.

The driver configuration dialog is shown in figure.



Element	Description
Panel ID	CANopen node ID assigned to the HMI
Controller ID	CANopen Node ID assigned to the CAN controller device
Baud Rate (kbps)	Speed of the CANopen network
Timeout (s)	Maximum allowed time the driver will wait for a response from the PLC before reporting a communication error
Enable Update Rate	Use this option to enable a wait time between two communication requests
Update Rate (ms)	Minimum interval time between two requests; it can be useful when the bus load needs to be properly controller and limited

Element	Description
PLC Models	The list allows selecting the controller model you are going to connect to. The selection will influence the data range offset per each data type according to the specific controller memory resources
PLC Network	The protocol allows the connection of multiple controllers to one operator panel. To set-up multiple connections, check “PLC network” checkbox and enter the node ID per each slave you need to access.  CANopen HMI ☑ PLC Network Panel ID: 1 Controller ID: -1 Baud Rate (kbps): 125 Timeout (s): 2 ☐ Enable Update Rate Update Rate (ms): 10 PLC Models: GE Fanuc, CoDeSys, Moeller, ISaGRAF, SAM GPM500, CoDeSys 4 PDO Slaves: Slave Id, Model CANopen HMI Controller ID: 0 PLC Models: GE Fanuc, CoDeSys, Moeller, ISaGRAF, SAM GPM500, CoDeSys 4 PDO

Connecting the HMI to CODESYS V2 Controllers

This chapter describes all the steps you have to follow in order to establish a successful connection between the HMI and CODESYS CANopen master controller.

The PLC support program has been developed with CODESYS programming software version 2.

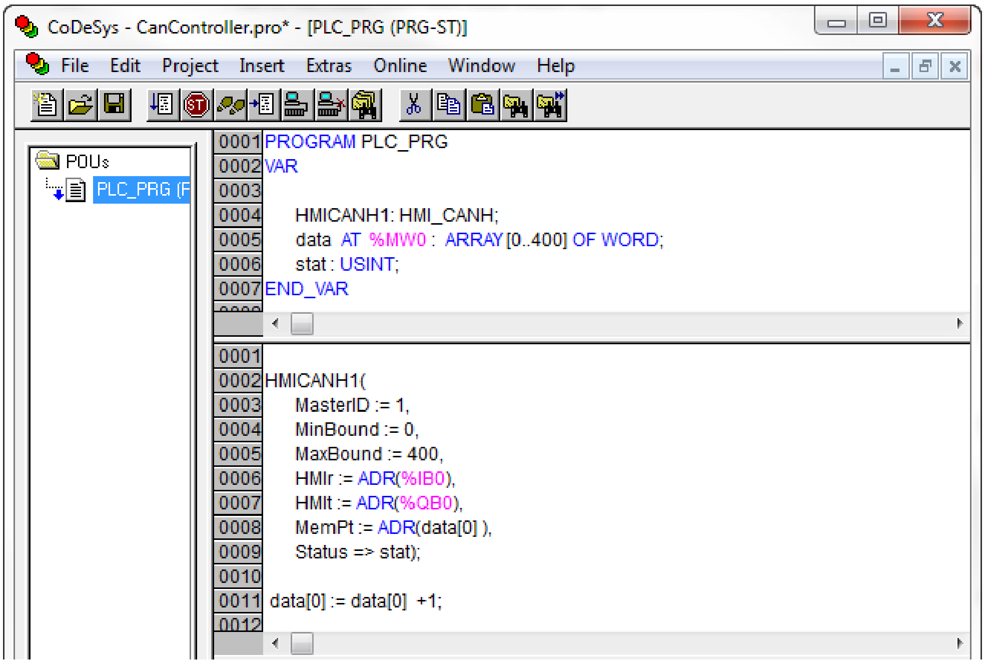
PLC Library Call

The server function running in the PLC program has been designed in the form of Library called “HMI_Canh”, written using the “ST” programming language. Proper working example is available on demand.

The Function Block parameters are the following:

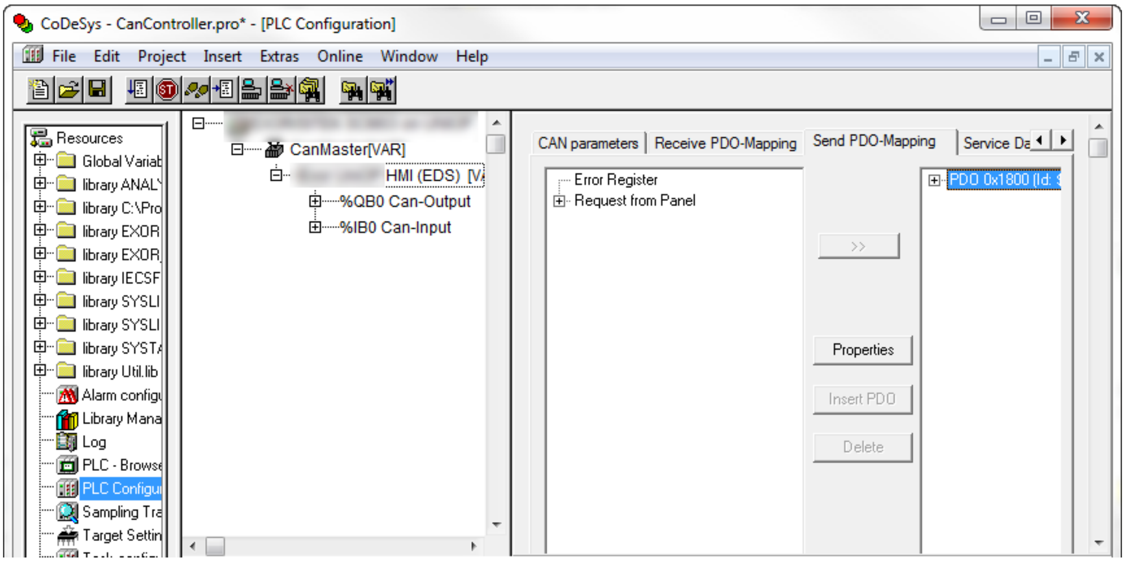
MasterID	CANopen Master Node number;
MinBound	Lower limit of the PLC memory addressable (visible) by the HMI
MaxBound	Upper limit of the PLC memory addressable (visible) by the HMI
HHIr	Offset in the PLC memory where the PDO message received from the panel is mapped
HMIt	Offset in the PLC memory where the PDO message to be sent to the panel is mapped
MemPt	Offset in the PLC memory where the data is received
Status	Status

The PLC Function block support the use of more than one panel simply repeating the call of the same function for all the additional units specifying before each call the proper calling parameters.


CoDeSys - CanController.pro* - [PLC_PRG (PRG-ST)]
File Edit Project Insert Extras Online Window Help
POUs
PLC_PRG (F

```
PROGRAM PLC_PRG
VAR

    HMICANH1: HMI_CANH;
    data  AT  %MW0 :  ARRAY [0..400] OF WORD;
    stat : USINT;
END_VAR
```

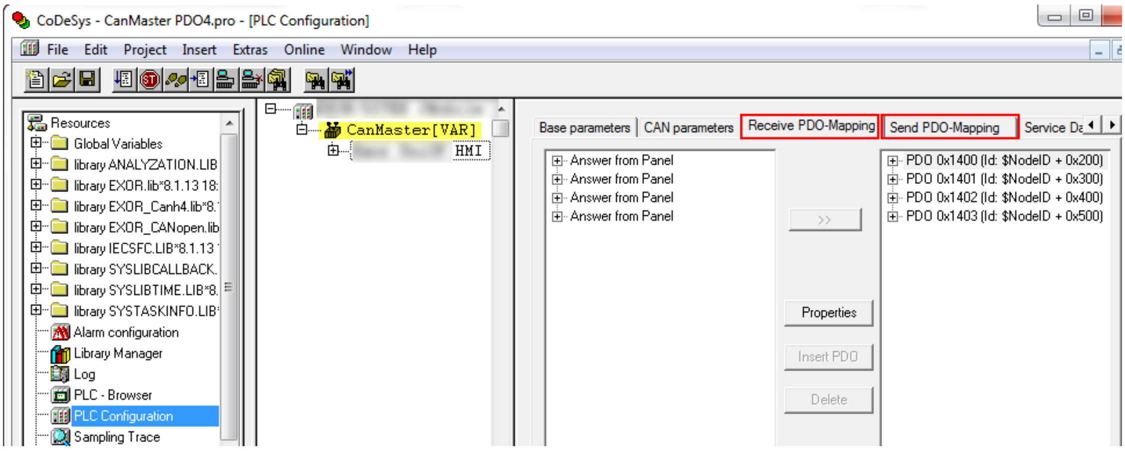
```

HMICANH1(
    MasterID := 1,
    MinBound := 0,
    MaxBound := 400,
    HMIr := ADR(%IB0),
    HMIt := ADR(%QB0),
    MemPt := ADR(data[0] ),
    Status => stat);

 data[0] := data[0]  +1;
```



CODESYS V2 4PDO

In some cases it is useful to choose the model “CODESYS 4 PDO” where 4 PDO objects are used for transmission and 4 for reception. This solution may provide higher communication speed between the two devices.

To operate with 4 PDO the correct model should be set in HMI project and the PDOs for receive and transmit slots.



Note: CANopen Master PLC Configuration must be configured properly. In case of “CODESYS 4 PDO”.

Communication Status

The current communication status can be displayed using the dedicated system variables. Please refer to the User Manual for further information about available system variables and their use.

The codes supported for this communication driver are:

Error	Notes
NAK	Controller replies with a not acknowledge.
Timeout	Request is not replied within the specified timeout period; ensure the controller is connected and properly configured for network access
Line Error	Returned when an error on the communication parameter setup is detected (baud rate); ensure the communication parameter settings of the controller is compatible with panel communication setup
Invalid response	The panel did receive from the controller a response, but its format or its contents or its length is not as expected; ensure the data programmed in the project are consistent with the controller resources.
CAN port not found	Make sure option module is correctly plugged
CAN port in use	Make sure option module is not already in use
General error	Error cannot be identified; should never be reported; contact technical support

CODESYS V2 ETH

CODESYS V2 ETH communication driver for supports communication through Ethernet connection with controllers based on the CODESYS V2.3 version.

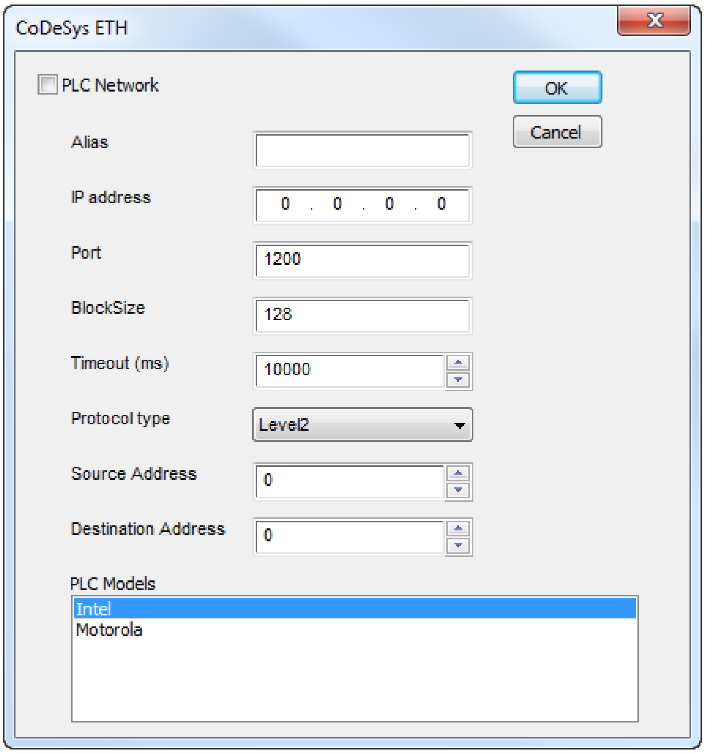
Protocol Editor settings

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.



□ PLC Network

Alias:

IP address: 0 . 0 . 0 . 0

Port: 1200

BlockSize: 128

Timeout (ms): 10000

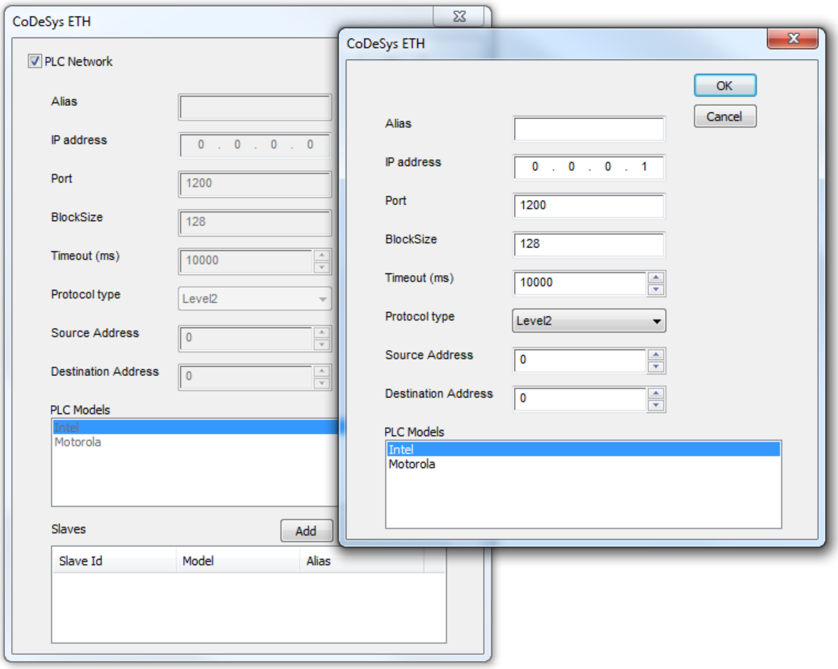
Protocol type: Level2

Source Address: 0

Destination Address: 0

PLC Models:
Intel
Motorola

Element	Description
Alias	Name identifying nodes in network configurations. The name will be added as a prefix to each tag name imported for each network node.
IP address	Ethernet IP address of the controller.
Port	Port number used by the CODESYS V2 Ethernet driver. The default value is set to **1200**, which is also the default setting of CODESYS-based controllers.
Block Size	Maximum block size supported by your controller (limit is 1024 KB).

Element	Description
Timeout (ms)	Time delay in milliseconds between two retries of the same message when communication fails.
Protocol type	Protocol variant to be used. Please make sure you check which protocol variant is supported by the CODESYS runtime you want to connect.
Source Address, Destination Address	Available only when **TCP/IP Level 2 Route** is selected in **Protocol Type**. 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.
PLC Models	Two PLC models are available. - **Intel** - **Motorola**
PLC Network	IP address for all controllers in multiple connections. **PLC network** check box must be selected to enable multiple connections.  CoDeSys ETH ☑ PLC Network Alias: IP address: 0 . 0 . 0 . 0 Port: 1200 BlockSize: 128 Timeout (ms): 10000 Protocol type: Level2 Source Address: 0 Destination Address: 0 PLC Models: Intel, Motorola Slaves Slave Id | Model | Alias CoDeSys ETH Alias: IP address: 0 . 0 . 0 . 1 Port: 1200 BlockSize: 128 Timeout (ms): 10000 Protocol type: Level2 Source Address: 0 Destination Address: 0 PLC Models: Intel, Motorola

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).

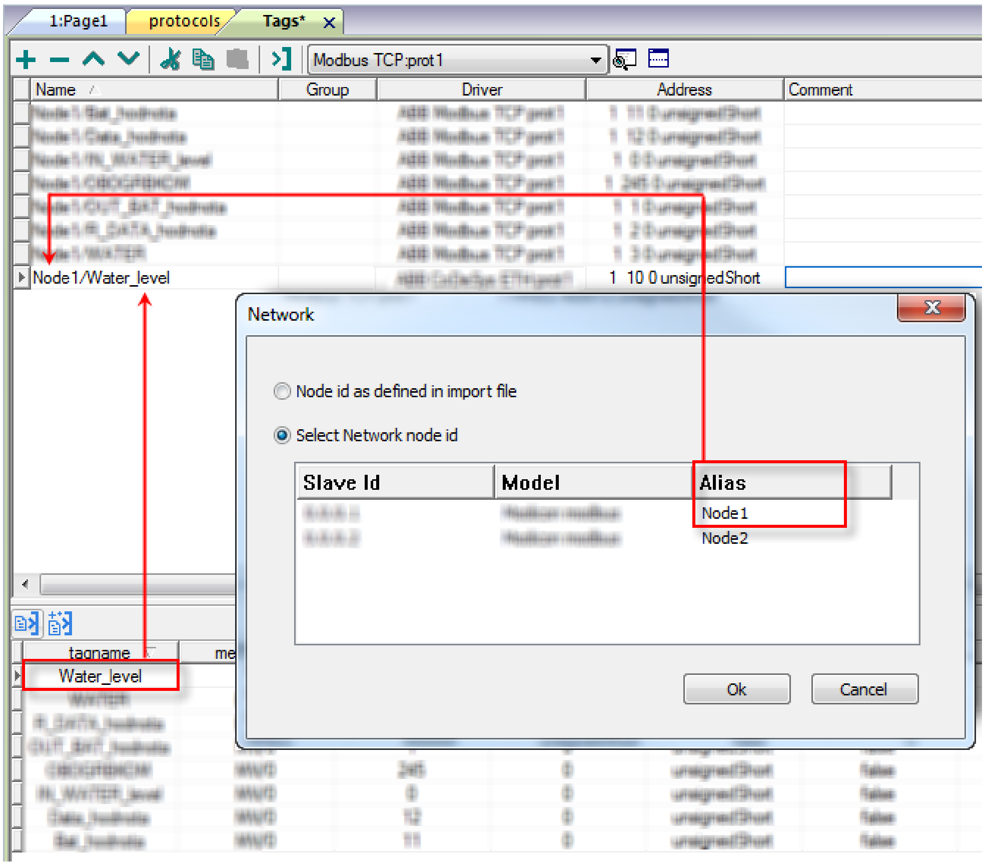
iPLC CODESYS supports communication with CODESYS V2 Ethernet driver with symbol based support starting from V1.55 and above.

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.



Name	Group	Driver	Address	Comment
Node1/Water_level			1 10 0 unsignedShort	

Network

○ Node id as defined in import file

● Select Network node id

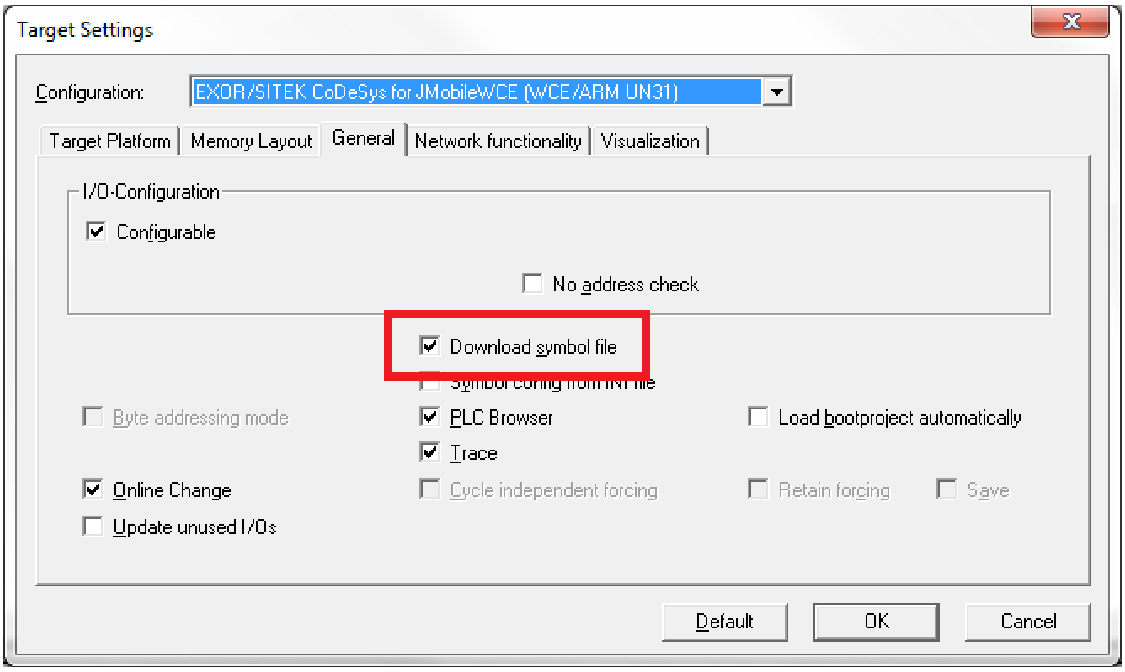
Slave Id	Model	Alias
		Node1
		Node2

tagname
Water_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 creating the project in CODESYS, select **Download symbol file**.



Target Settings

Configuration: EXOR/SITEK CoDeSys for JMobileWCE (WCE/ARM UN31)

I/O-Configuration

☑ Configurable

☐ No address check

☑ Download symbol file

☐ Symbol config from INI file

☐ Byte addressing mode

☑ PLC Browser

☐ Load bootproject automatically

☑ Trace

☑ Online Change

☐ Cycle independent forcing

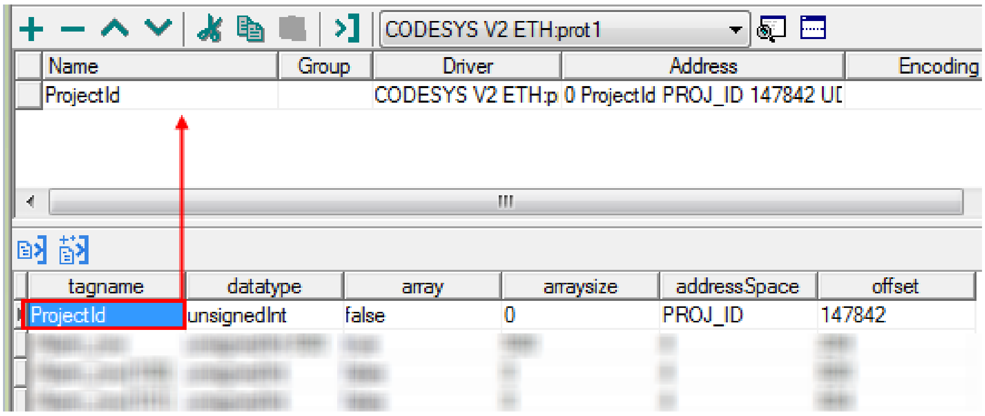
☐ Retain forcing

☐ Save

☐ Update unused I/Os

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.



Name	Group	Driver	Address	Encoding
ProjectId		CODESYS V2 ETH:p	0 ProjectId PROJ_ID 147842 UI	

tagname	datatype	array	arraysize	addressSpace	offset
ProjectId	unsignedInt	false	0	PROJ_ID	147842

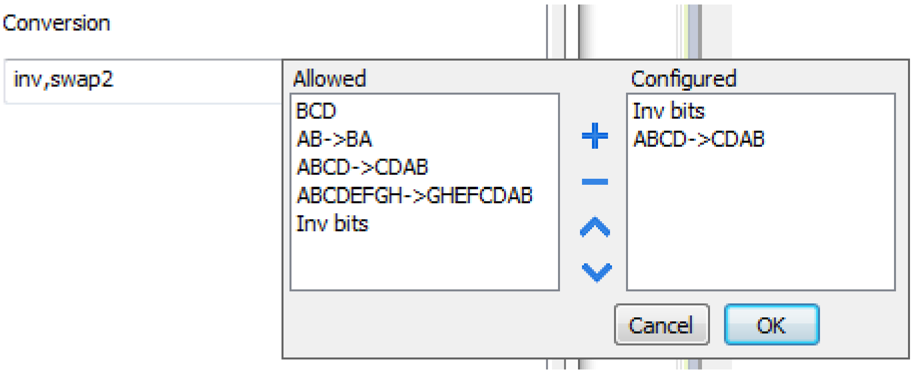
Data types

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: inv,swap2

Allowed:
- BCD
- AB->BA
- ABCD->CDAB
- ABCDEFGH->GHEFCDAB
- Inv bits

Configured:
- Inv bits
- ABCD->CDAB

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

Value	Description
Inv bits	**inv**: Invert all the bits of the tag. *Example:* 1001 → 0110 (in binary format) 9 → 6 (in decimal format)
Negate	**neg**: Set the opposite of tag value. *Example:* 25.36 → -25.36
AB -> BA	**swapnibbles**: Swap nibbles in a byte. *Example:* 15D4 → 514D (in hexadecimal format) 5588 → 20813 (in decimal format)
ABCD -> CDAB	**swap2**: Swap bytes in a word. *Example:* 9ACC → CC9A (in hexadecimal format) 39628 → 52378 (in decimal format)
ABCDEFGH -> GHEFCDAB	**swap4**: Swap bytes in a double word. *Example:* 32FCFF54 → 54FFFC32 (in hexadecimal format) 855441236 → 1426062386 (in decimal format)
ABC...NOP -> OPM...DAB	**swap8**: Swap bytes in a long word. Example: 142.366 → -893553517.588905 (in decimal format) 0 10000000110 0001110010111011011001000101101000011100101011000001 → 1 10000011100 1010101000010100010110110110110010110110000100111101 (in binary format)
BCD	**bcd**: 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.

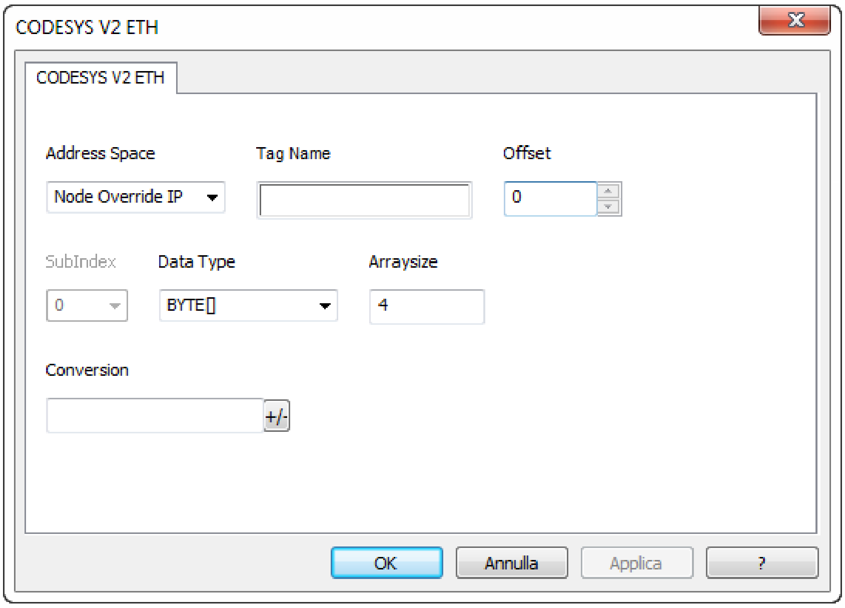
Node Override IP	PLC operation
0.0.0.0	Communication with the controller is stopped, no request frames are generated anymore.
Different from 0.0.0.0	It is interpreted as node IP override and the target IP address is replaced runtime with the new value.

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 V2 ETH

CODESYS V2 ETH

Address Space: Node Override IP

Tag Name:

Offset: 0

SubIndex: 0

Data Type: BYTE[]

Arraysize: 4

Conversion:

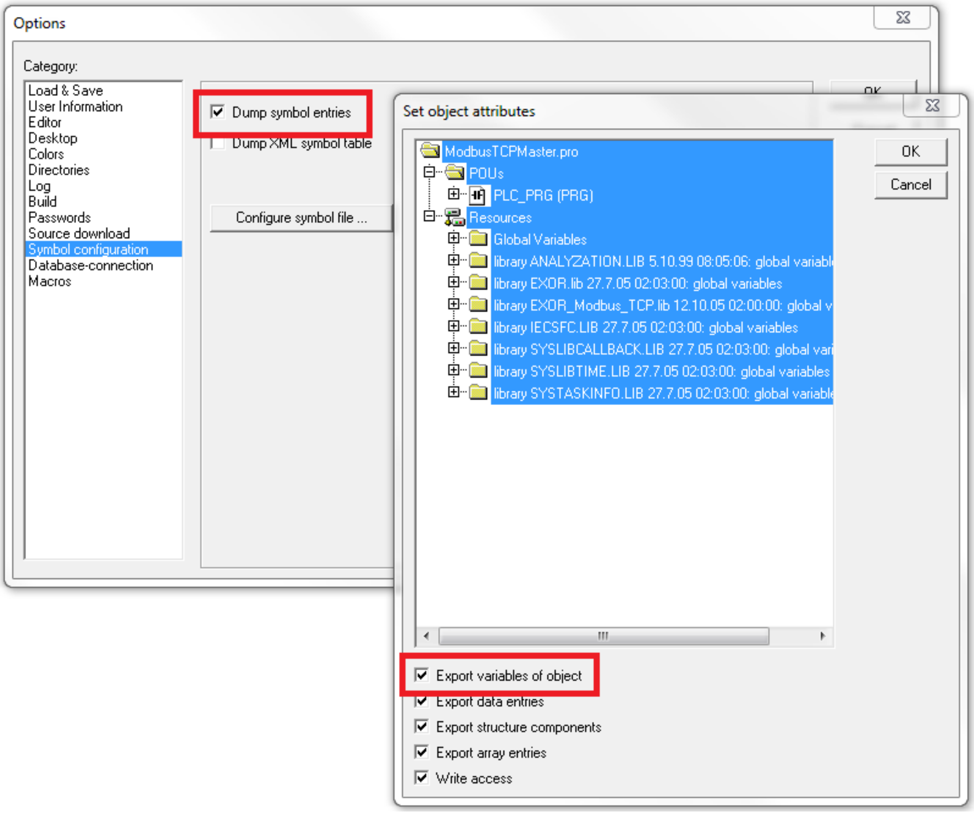
Tag Import

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.



Options

Category:

Load & Save
User Information
Editor
Desktop
Colors
Directories
Log
Build
Passwords
Source download
Symbol configuration
Database-connection
Macros

☑ Dump symbol entries
□ Dump XML symbol table

Configure symbol file ...

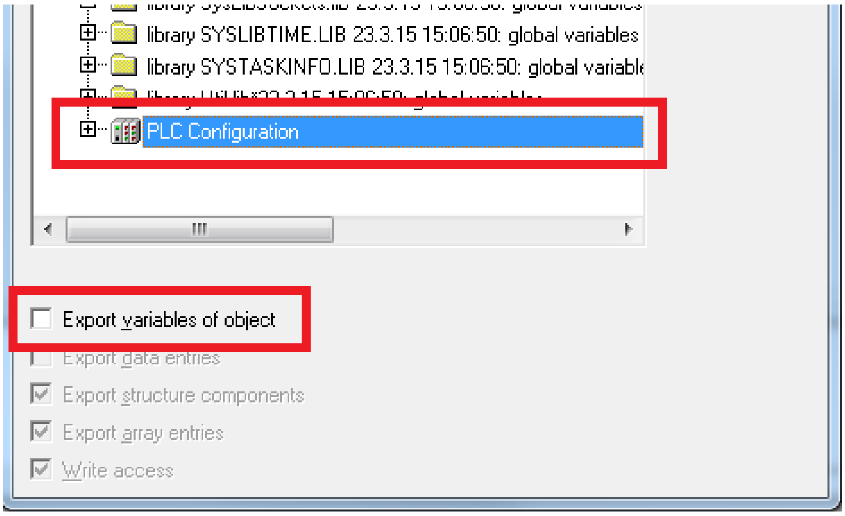
Set object attributes

ModbusTCPMaster.pro
POUs
PLC_PRG (PRG)
Resources
Global Variables
library ANALYZATION.LIB 5.10.99 08:05:06: global variabl
library EXOR.lib 27.7.05 02:03:00: global variables
library EXOR_Modbus_TCP.lib 12.10.05 02:00:00: global v
library IECSFC.LIB 27.7.05 02:03:00: global variables
library SYSLIBCALLBACK.LIB 27.7.05 02:03:00: global vari
library SYSLIBTIME.LIB 27.7.05 02:03:00: global variables
library SYSTASKINFO.LIB 27.7.05 02:03:00: global variabl

OK
Cancel

☑ Export variables of object
☑ Export data entries
☑ Export structure components
☑ Export array entries
☑ Write access

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".

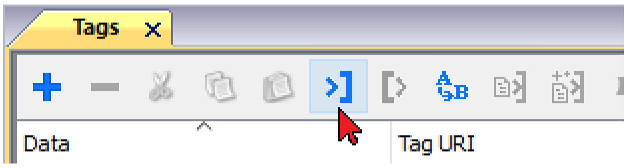


library SYSLIBTIME.LIB 23.3.15 15:06:50: global variables
library SYSTASKINFO.LIB 23.3.15 15:06:50: global variable
PLC Configuration

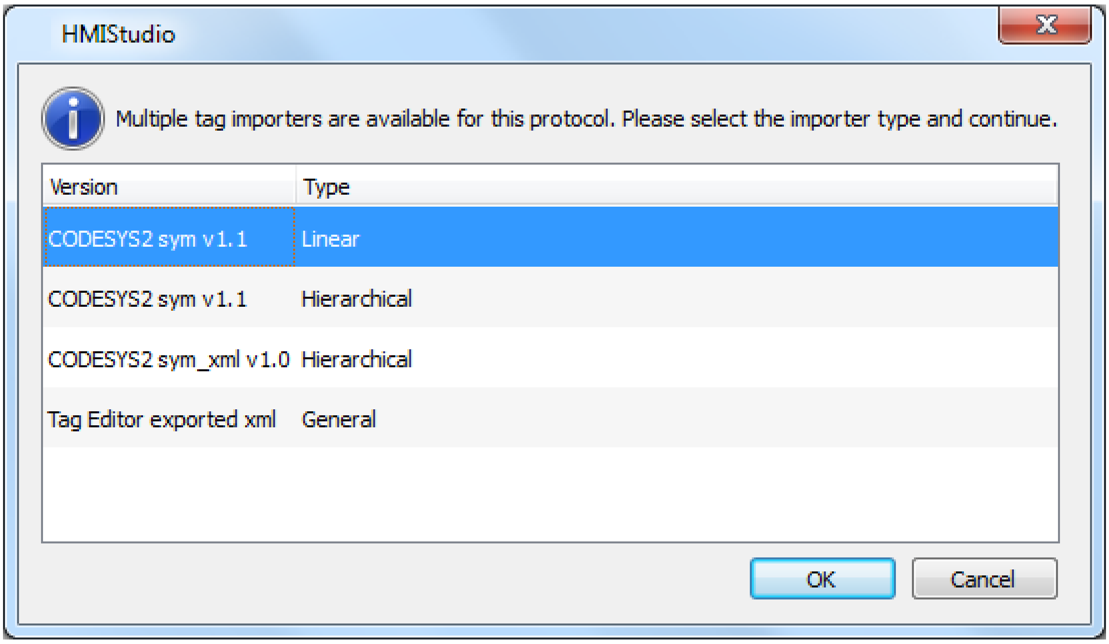
□ Export variables of object
□ Export data entries
☑ Export structure components
☑ Export array entries
☑ Write access

Importing Tags in Tag Editor

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

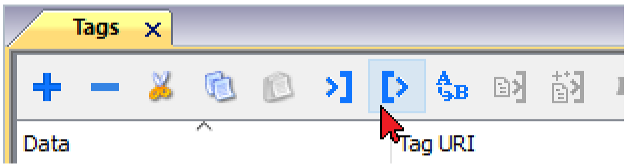


The following dialog shows which importer type can be selected.


HMIStudio

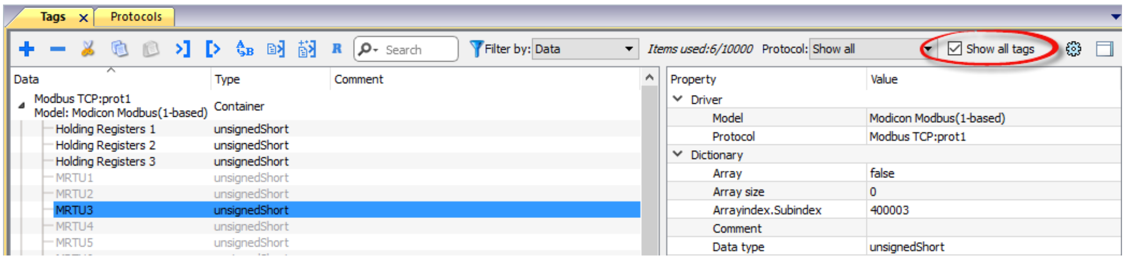
Multiple tag importers are available for this protocol. Please select the importer type and continue.

Version	Type
CODESYS2 sym v1.1	Linear
CODESYS2 sym v1.1	Hierarchical
CODESYS2 sym_xml v1.0	Hierarchical
Tag Editor exported xml	General

Importer	Description
CODESYS2 sym v1.1 Linear	Requires a **.sym** file. All variables will be displayed at the same level.
CODESYS2 sym v1.1 Hierarchical	Requires a **.sym** file. All variables will be displayed according to CODESYS V2 Hierarchical view.
CODESYS2 sym_xml v1.0 Hierarchical	Requires a **.sym_xml** file. All variables will be displayed according to CODESYS V2 Hierarchical view.
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**.

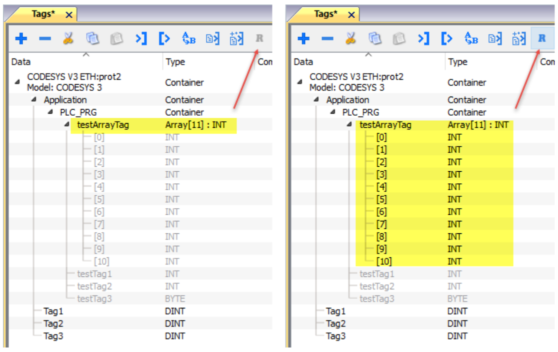
The tags available within the Dictionary but not imported into the project are gray and are visible only when the "Show all tags" check box is selected.



Search Filter by: Data *Items used:6/10000* Protocol: Show all ☑ Show all tags

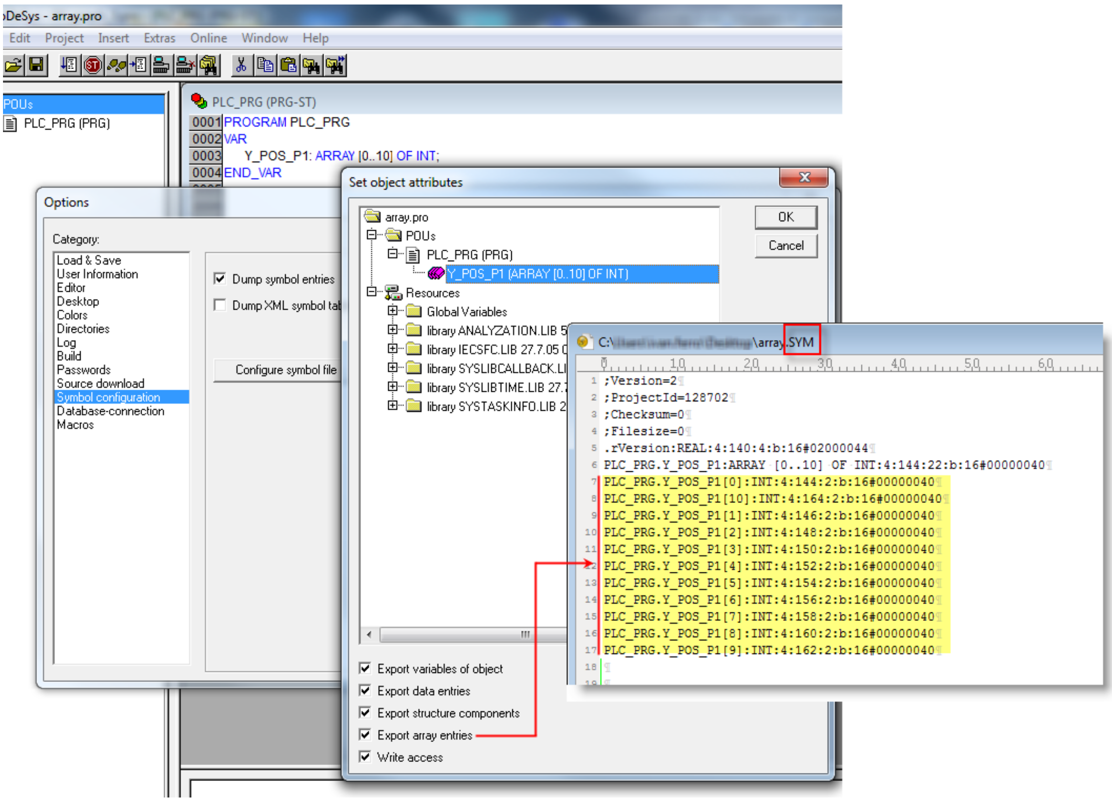
Data	Type	Comment
Modbus TCP:prot1 Model: Modicon Modbus(1-based)	Container	
Holding Registers 1	unsignedShort	
Holding Registers 2	unsignedShort	
Holding Registers 3	unsignedShort	
MRTU1	unsignedShort	
MRTU2	unsignedShort	
MRTU3	unsignedShort	
MRTU4	unsignedShort	
MRTU5	unsignedShort	

Property	Value
Driver	
Model	Modicon Modbus(1-based)
Protocol	Modbus TCP:prot1
Dictionary	
Array	false
Array size	0
Arrayindex.Subindex	400003
Comment	
Data type	unsignedShort

Toolbar item	Description
	Import Tag(s). Select tags to be imported and click on this icon to add tags from tag dictionary to the project
	Update Tag(s). Click on this icon to update the tags in the project, due a new dictionary import.
R	Check this box to import all sub-elements of a tag. Example of both checked and unchecked result:  Tags* Data | Type CODESYS V3 ETH:prot2 Model: CODESYS 3 | Container Application | Container PLC_PRG | Container testArrayTag | Array[11] : INT [0] | INT [1] | INT [2] | INT [3] | INT [4] | INT [5] | INT [6] | INT [7] | INT [8] | INT [9] | INT [10] | INT testTag1 | INT testTag2 | INT testTag3 | BYTE Tag1 | DINT Tag2 | DINT Tag3 | DINT Tags* Data | Type CODESYS V3 ETH:prot2 Model: CODESYS 3 | Container Application | Container PLC_PRG | Container testArrayTag | Array[11] : INT [0] | INT [1] | INT [2] | INT [3] | INT [4] | INT [5] | INT [6] | INT [7] | INT [8] | INT [9] | INT [10] | INT testTag1 | INT testTag2 | INT testTag3 | BYTE Tag1 | DINT Tag2 | DINT Tag3 | DINT
Search Filter by: Tag name	Searches tags in the dictionary basing on filter combo-box item selected.

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.



```
PROGRAM PLC_PRG
VAR
    Y_POS_P1: ARRAY [0..10] OF INT;
END_VAR
```

☑ Dump symbol entries

□ Dump XML symbol tab

☑ Export variables of object

☑ Export data entries

☑ Export structure components

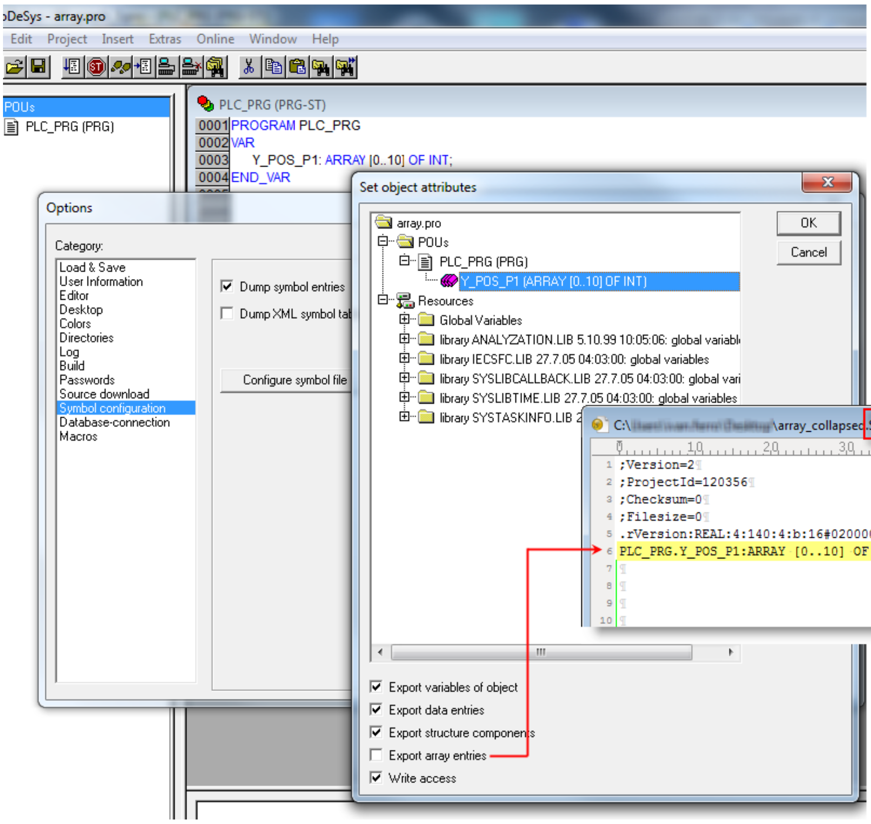
☑ Export array entries

☑ Write access

```
;Version=2
;ProjectId=128702
;Checksum=0
;Filesize=0
.rVersion:REAL:4:140:4:b:16#02000044
PLC_PRG.Y_POS_P1:ARRAY [0..10] OF INT:4:144:22:b:16#00000040
PLC_PRG.Y_POS_P1[0]:INT:4:144:2:b:16#00000040
PLC_PRG.Y_POS_P1[10]:INT:4:164:2:b:16#00000040
PLC_PRG.Y_POS_P1[1]:INT:4:146:2:b:16#00000040
PLC_PRG.Y_POS_P1[2]:INT:4:148:2:b:16#00000040
PLC_PRG.Y_POS_P1[3]:INT:4:150:2:b:16#00000040
PLC_PRG.Y_POS_P1[4]:INT:4:152:2:b:16#00000040
PLC_PRG.Y_POS_P1[5]:INT:4:154:2:b:16#00000040
PLC_PRG.Y_POS_P1[6]:INT:4:156:2:b:16#00000040
PLC_PRG.Y_POS_P1[7]:INT:4:158:2:b:16#00000040
PLC_PRG.Y_POS_P1[8]:INT:4:160:2:b:16#00000040
PLC_PRG.Y_POS_P1[9]:INT:4:162:2:b:16#00000040
```

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.


CoDeSys - array.pro
Edit Project Insert Extras Online Window Help
POUs
PLC_PRG (PRG)
PLC_PRG (PRG-ST)

```
0001 PROGRAM PLC_PRG
0002 VAR
0003     Y_POS_P1: ARRAY [0..10] OF INT;
0004 END_VAR
```

Options
Category:
Load & Save
User Information
Editor
Desktop
Colors
Directories
Log
Build
Passwords
Source download
Symbol configuration
Database-connection
Macros
Configure symbol file

☑ Dump symbol entries
☐ Dump XML symbol tal

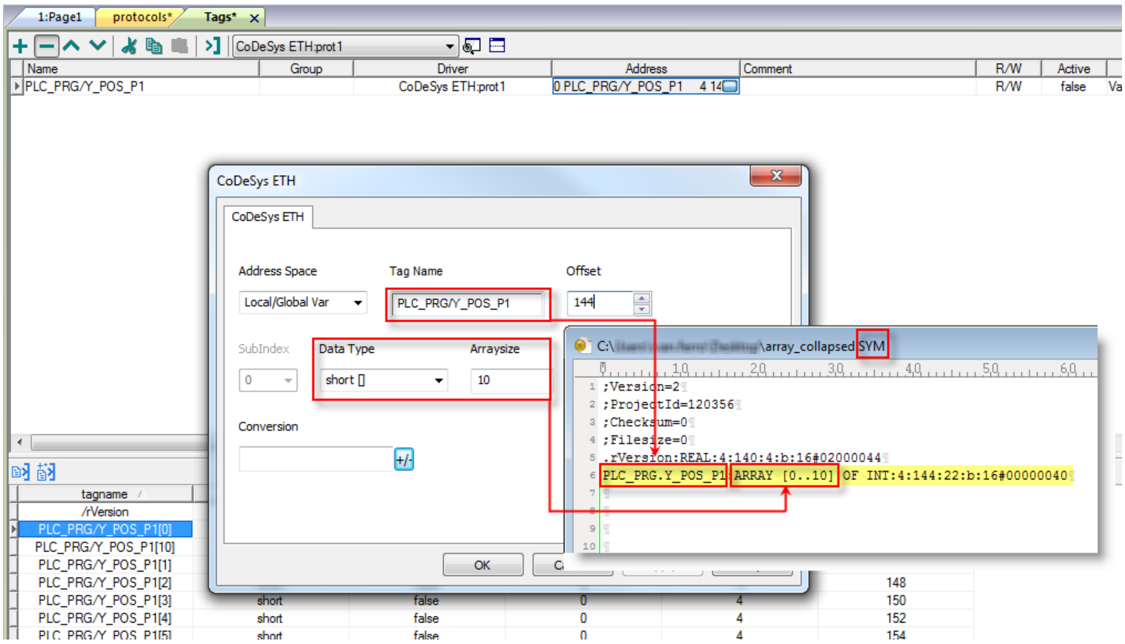
Set object attributes
OK
Cancel
array.pro
POUs
PLC_PRG (PRG)
Y_POS_P1 (ARRAY [0..10] OF INT)
Resources
Global Variables
library ANALYZATION.LIB 5.10.99 10:05:06: global variabl
library IECSFC.LIB 27.7.05 04:03:00: global variables
library SYSLIBCALLBACK.LIB 27.7.05 04:03:00: global vari
library SYSLIBTIME.LIB 27.7.05 04:03:00: global variables
library SYSTASKINFO.LIB 2

☑ Export variables of object
☑ Export data entries
☑ Export structure components
☐ Export array entries
☑ Write access

C:\[redacted]\array_collapsed.SYM

```
;Version=2
;ProjectId=120356
;Checksum=0
;Filesize=0
.rVersion:REAL:4:140:4:b:16#02000044
PLC_PRG.Y_POS_P1:ARRAY [0..10] OF INT:4:144:22:b:16#00000040
```

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.



Name	Group	Driver	Address	Comment	R/W	Active
PLC_PRG/Y_POS_P1		CoDeSys ETH:prot1	0 PLC_PRG/Y_POS_P1 4 14		R/W	false

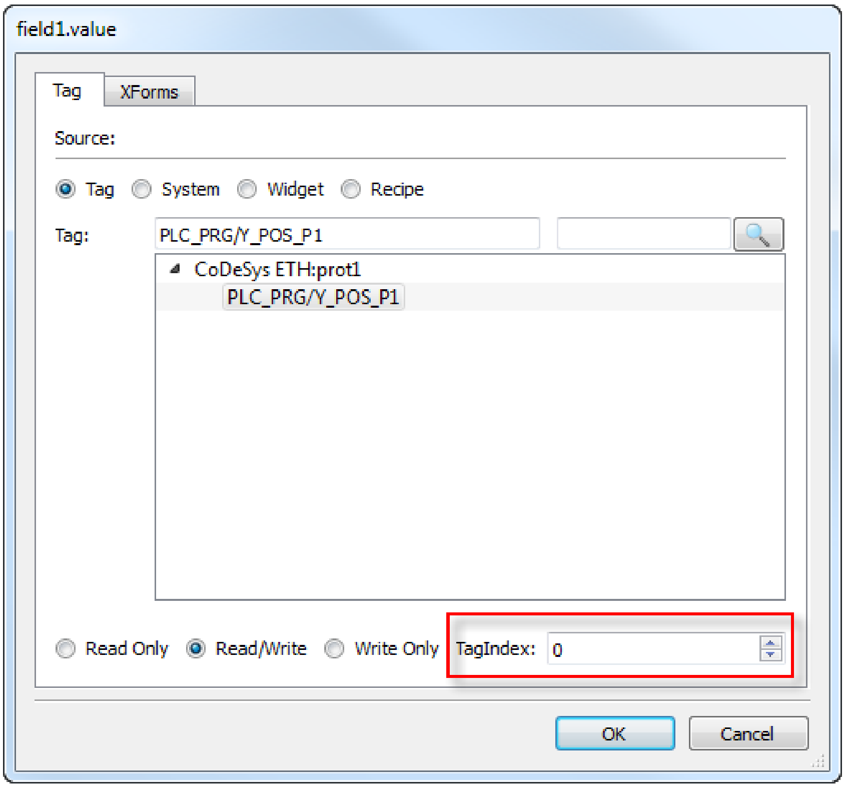
Address Space: Local/Global Var
Tag Name: PLC_PRG/Y_POS_P1
Offset: 144
SubIndex: 0
Data Type: short []
Arraysize: 10
Conversion:

C:\[redacted]\array_collapsed.SYM

```
;Version=2
;ProjectId=120356
;Checksum=0
;Filesize=0
.rVersion:REAL:4:140:4:b:16#02000044
PLC_PRG.Y_POS_P1:ARRAY [0..10] OF INT:4:144:22:b:16#00000040
```

tagname					
/rVersion					
PLC_PRG/Y_POS_P1[0]					
PLC_PRG/Y_POS_P1[10]					
PLC_PRG/Y_POS_P1[1]					
PLC_PRG/Y_POS_P1[2]					148
PLC_PRG/Y_POS_P1[3]	short	false	0	4	150
PLC_PRG/Y_POS_P1[4]	short	false	0	4	152
PLC_PRG/Y_POS_P1[5]	short	false	0	4	154

All tag elements can be referenced in the editor using **TagIndex** in the **Attach to Tag** dialog.



Source:

● Tag ○ System ○ Widget ○ Recipe

Tag: PLC_PRG/Y_POS_P1

CoDeSys ETH:prot1
PLC_PRG/Y_POS_P1

○ Read Only ● Read/Write ○ Write Only TagIndex: 0

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
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 Acknoledge	Controller didn’t send acknowledge.
Time out on last Acknoledge	Controller didn’t sent last ack.
Time out on data reciving	Controller does not reply with data.
Connection timeout	Device not connected.

CODESYS V3 ETH

The CODESYS V3 ETH communication driver supports communication thought 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.

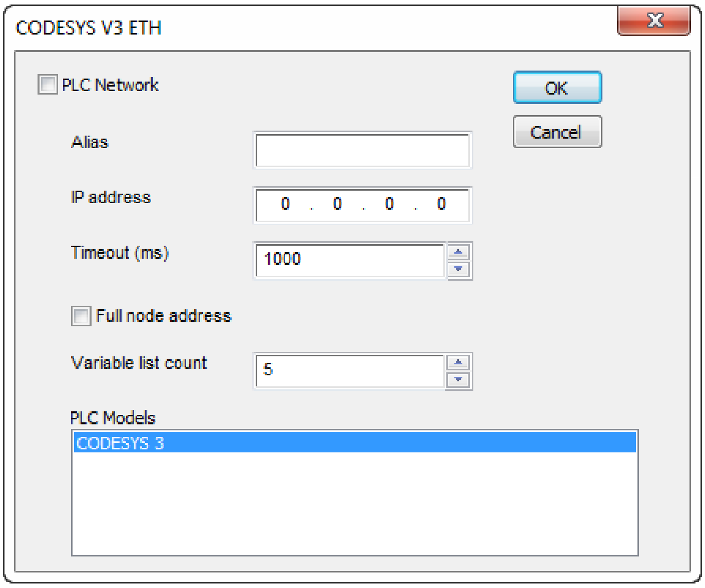
Protocol Editor Settings

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.


CODESYS V3 ETH

☐ PLC Network

Alias:

IP address: 0 . 0 . 0 . 0

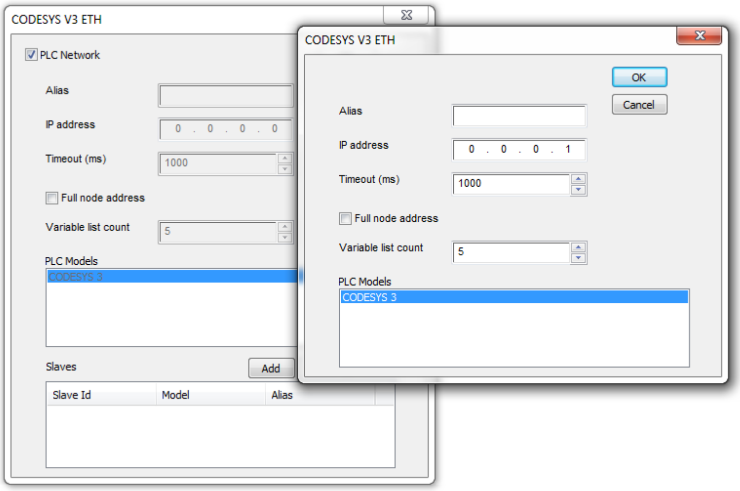
Timeout (ms): 1000

☐ Full node address

Variable list count: 5

PLC Models: CODESYS 3

Element	Description
Alias	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.
IP address	Ethernet IP address of the controller
Full	Since some implementations of CODESYS V3 at runtime require all four values of the IP address

Element	Description
node address	to be specified in the protocol frames, this flag forces the protocol to create IP addresses using all four address fields of the IP.
Variable list count	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. 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.
PLC Model	Byte order that will be used by the communication driver when sending communication frames to the PLC.
Timeout	Number of milliseconds between retries when communication fails.
PLC Network	Enable access to multiple networked controllers. For every controller (slave) set the proper option.  CODESYS V3 ETH ☑ PLC Network Alias: IP address: 0 . 0 . 0 . 0 Timeout (ms): 1000 ☐ Full node address Variable list count: 5 PLC Models: CODESYS 3 Slaves: Slave Id, Model, Alias CODESYS V3 ETH Alias: IP address: 0 . 0 . 0 . 1 Timeout (ms): 1000 ☐ Full node address Variable list count: 5 PLC Models: CODESYS 3

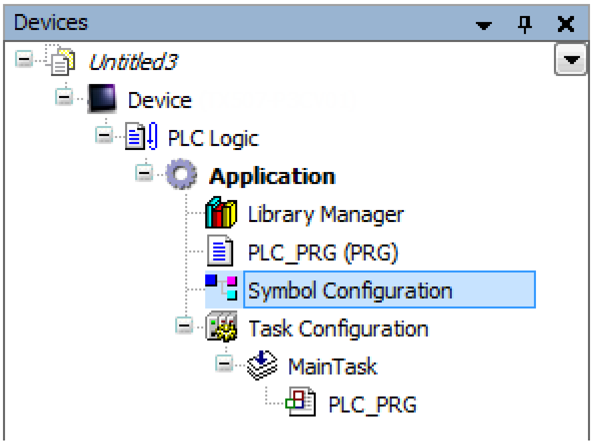
Note: Refer to the controller documentation to verify required values for the parameters **Full node address** or **Variable list count**.

Tag Import

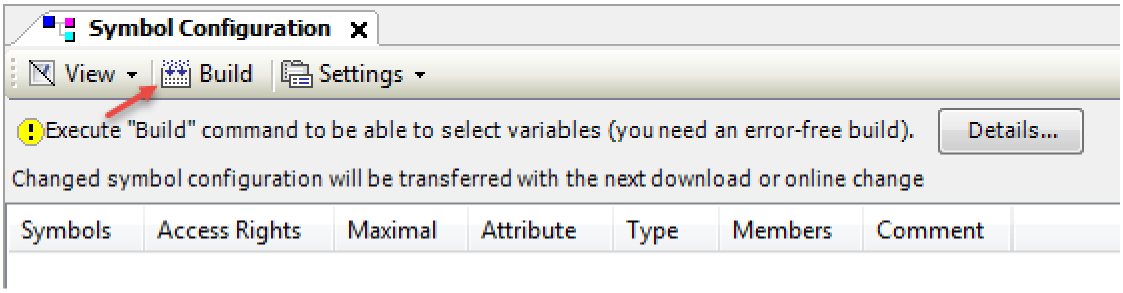
Exporting Tags from PLC

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.

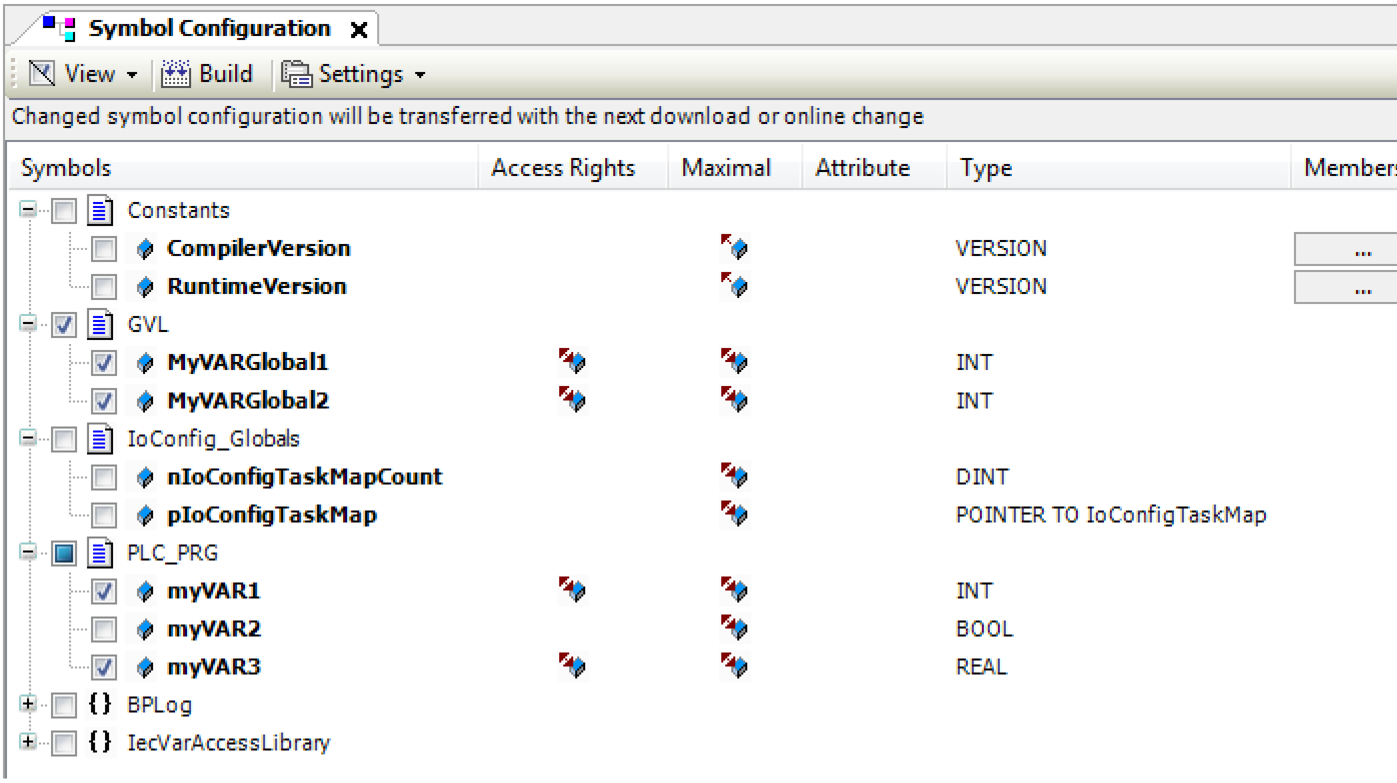


Execute "Build" command to be able to select variables (you need an error-free build).

Changed symbol configuration will be transferred with the next download or online change

Symbols	Access Rights	Maximal	Attribute	Type	Members	Comment

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.

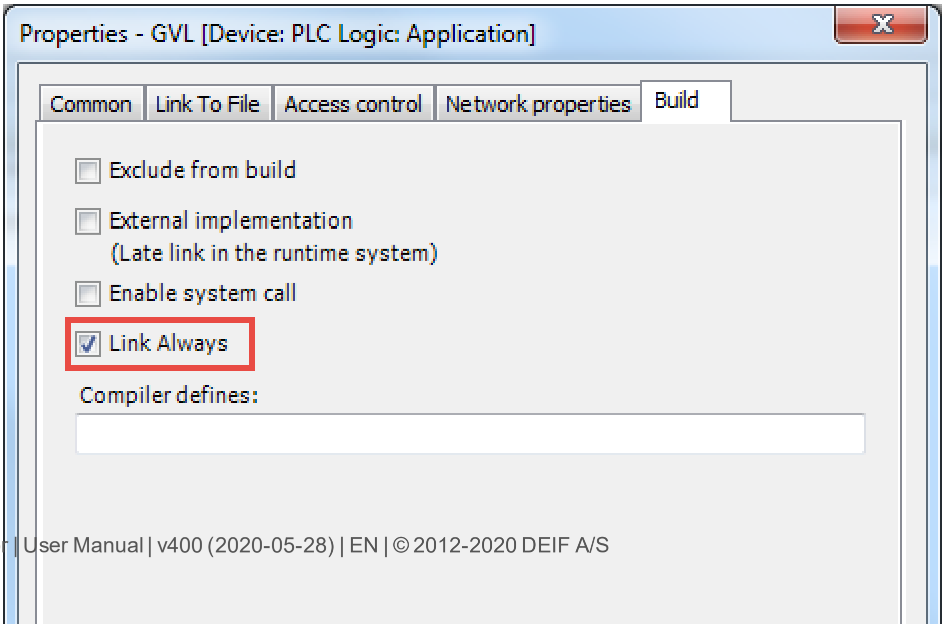


Changed symbol configuration will be transferred with the next download or online change

Symbols	Access Rights	Maximal	Attribute	Type	Members
☐ Constants					
☐ CompilerVersion		◆		VERSION	...
☐ RuntimeVersion		◆		VERSION	...
☑ GVL					
☑ MyVARGlobal1	◆	◆		INT	
☑ MyVARGlobal2	◆	◆		INT	
☐ IoConfig_Globals					
☐ nIoConfigTaskMapCount		◆		DINT	
☐ pIoConfigTaskMap		◆		POINTER TO IoConfigTaskMap	
■ PLC_PRG					
☑ myVAR1	◆	◆		INT	
☐ myVAR2		◆		BOOL	
☑ myVAR3	◆	◆		REAL	
☐ {} BPLog					
☐ {} IecVarAccessLibrary					

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.



☐ Exclude from build

☐ External implementation
(Late link in the runtime system)

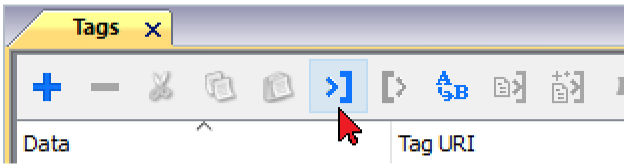
☐ Enable system call

☑ Link Always

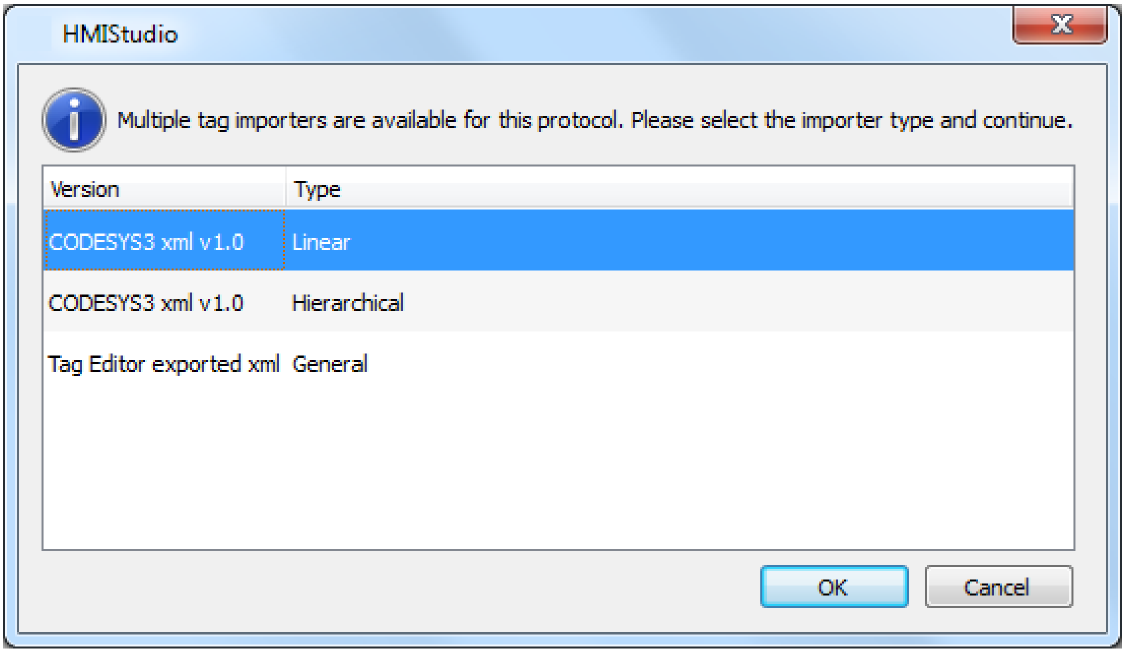
Compiler defines:

Importing Tags in Tag Editor

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

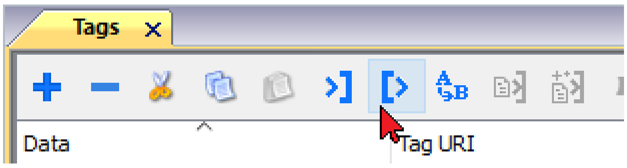


The following dialog shows which importer type can be selected.



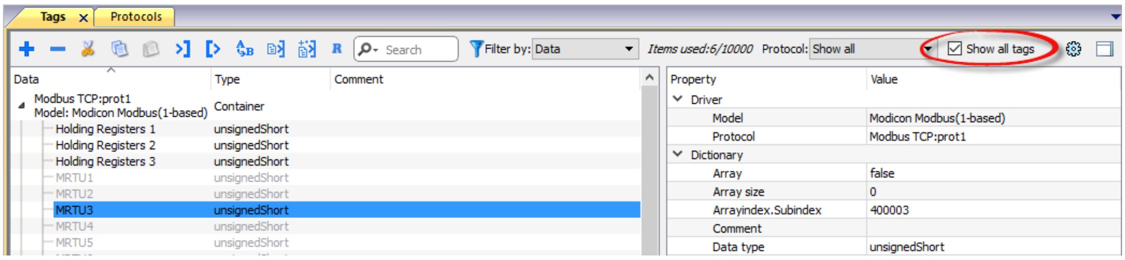
Multiple tag importers are available for this protocol. Please select the importer type and continue.

Version	Type
CODESYS3 xml v1.0	Linear
CODESYS3 xml v1.0	Hierarchical
Tag Editor exported xml	General

Importer	Description
CODESYS3 xml v1.0 Linear	Requires an **.xml** file. All variables will be displayed at the same level.
CODESYS3 xml v1.0 Hierarchical	Requires an **.xml** file. All variables will be displayed according to CODESYS V3 Hierarchical view.
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.

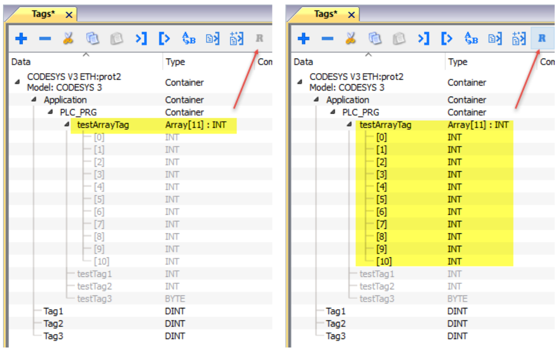


Tags | Protocols

Search | Filter by: Data | *Items used:6/10000* | Protocol: Show all | ☑ Show all tags

Data	Type	Comment
Modbus TCP:prot1 Model: Modicon Modbus(1-based)	Container	
Holding Registers 1	unsignedShort	
Holding Registers 2	unsignedShort	
Holding Registers 3	unsignedShort	
MRTU1	unsignedShort	
MRTU2	unsignedShort	
MRTU3	unsignedShort	
MRTU4	unsignedShort	
MRTU5	unsignedShort	

Property	Value
Driver	
Model	Modicon Modbus(1-based)
Protocol	Modbus TCP:prot1
Dictionary	
Array	false
Array size	0
Arrayindex.Subindex	400003
Comment	
Data type	unsignedShort

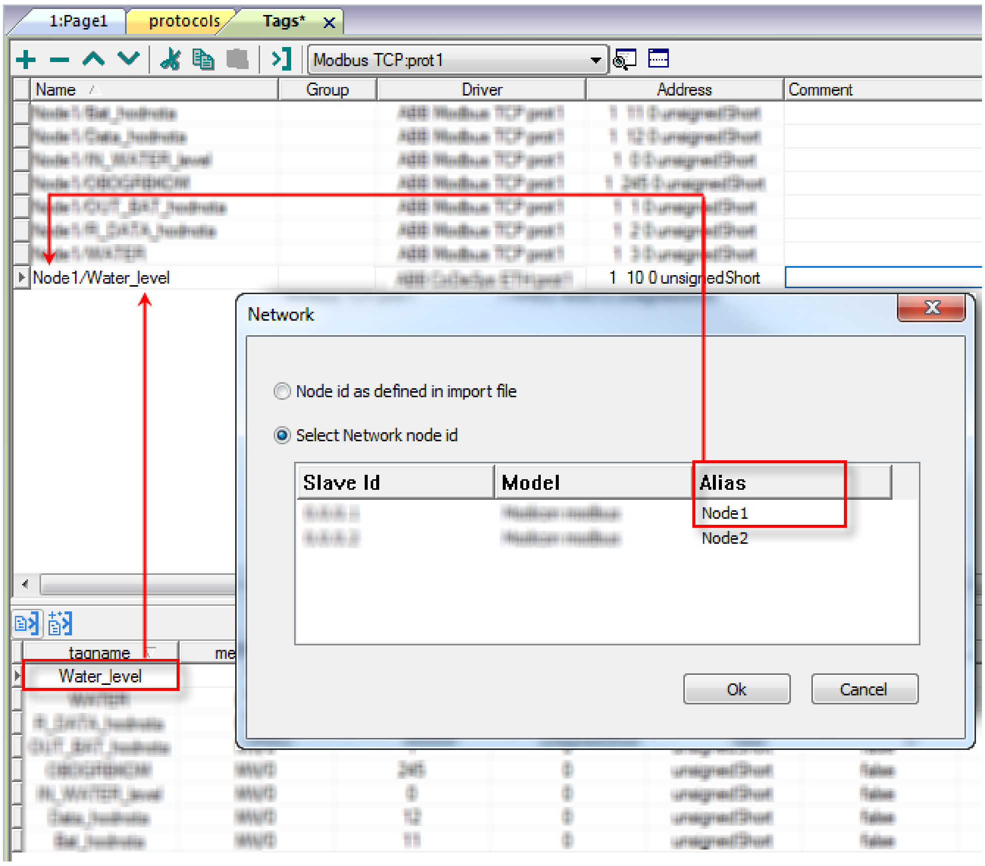
Toolbar item	Description
	Import Tag(s). Select tags to be imported and click on this icon to add tags from tag dictionary to the project
	Update Tag(s). Click on this icon to update the tags in the project, due a new dictionary import.
R	Check this box to import all sub-elements of a tag. Example of both checked and unchecked result:  Tags* Data / Type / Con CODESYS V3 ETH:prot2 Model: CODESYS 3 / Container Application / Container PLC_PRG / Container testArrayTag / Array[11] : INT [0] / INT [1] / INT [2] / INT [3] / INT [4] / INT [5] / INT [6] / INT [7] / INT [8] / INT [9] / INT [10] / INT testTag1 / INT testTag2 / INT testTag3 / BYTE Tag1 / DINT Tag2 / DINT Tag3 / DINT
Search Filter by: Tag name ▾	Searches tags in the dictionary basing on filter combo-box 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.



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.

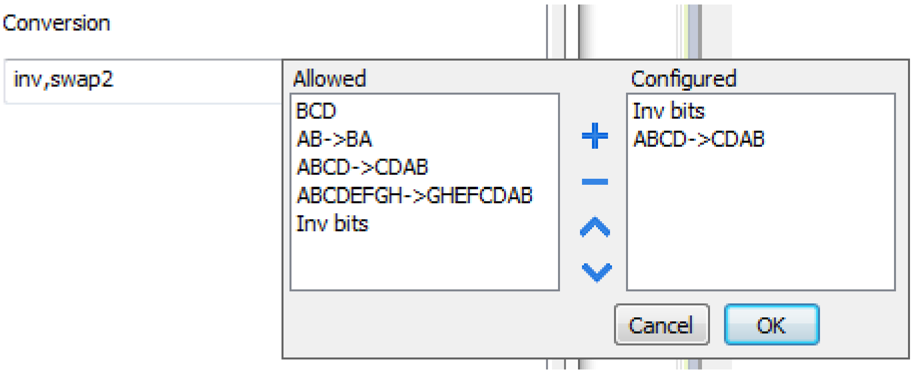
Data Types

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

inv,swap2

Allowed

BCD
AB->BA
ABCD->CDAB
ABCDEFGH->GHEFCDAB
Inv bits

Configured

Inv bits
ABCD->CDAB

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

Value	Description
Inv bits	**inv**: Invert all the bits of the tag. *Example:* 1001 → 0110 (in binary format) 9 → 6 (in decimal format)
Negate	**neg**: Set the opposite of tag value. *Example:* 25.36 → -25.36
AB -> BA	**swapnibbles**: Swap nibbles in a byte. *Example:* 15D4 → 514D (in hexadecimal format) 5588 → 20813 (in decimal format)
ABCD -> CDAB	**swap2**: Swap bytes in a word. *Example:* 9ACC → CC9A (in hexadecimal format) 39628 → 52378 (in decimal format)
ABCDEFGH -> GHEFCDAB	**swap4**: Swap bytes in a double word. *Example:* 32FCFF54 → 54FFFC32 (in hexadecimal format) 855441236 → 1426062386 (in decimal format)
ABC...NOP -> OPM...DAB	**swap8**: Swap bytes in a long word. Example: 142.366 → -893553517.588905 (in decimal format) 0 10000000110 0001110010111011011001000101101000011100101011000001 → 1 10000011100 1010101000010100010110110110110010110110000100111101 (in binary format)
BCD	**bcd**: 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.

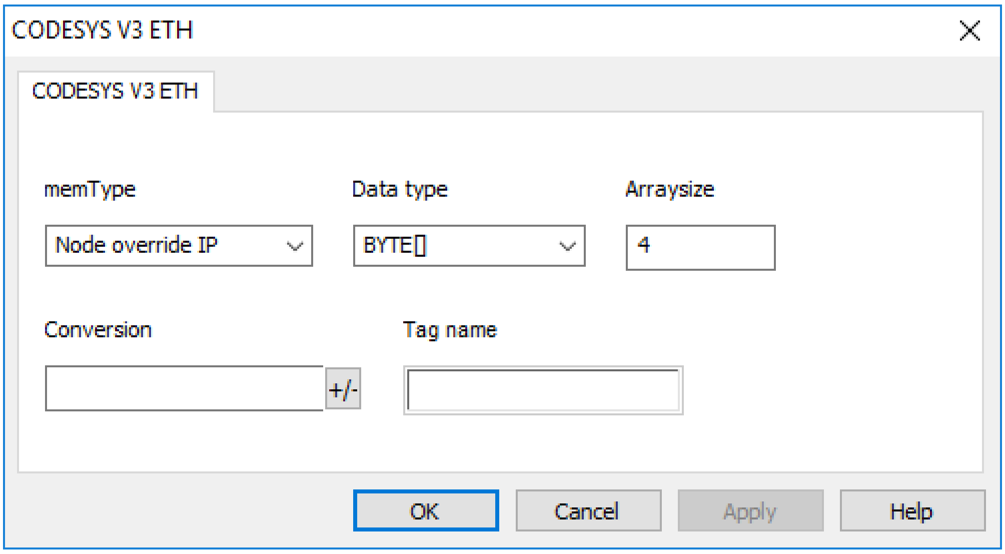
Node Override IP	PLC operation
0.0.0.0	Communication with the controller is stopped, no request frames are generated anymore.
Different from 0.0.0.0	It is interpreted as node IP override and the target IP address is replaced runtime with the new value.

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

CODESYS V3 ETH

memType: Node override IP

Data type: BYTE[]

Arraysize: 4

Conversion:

Tag name:

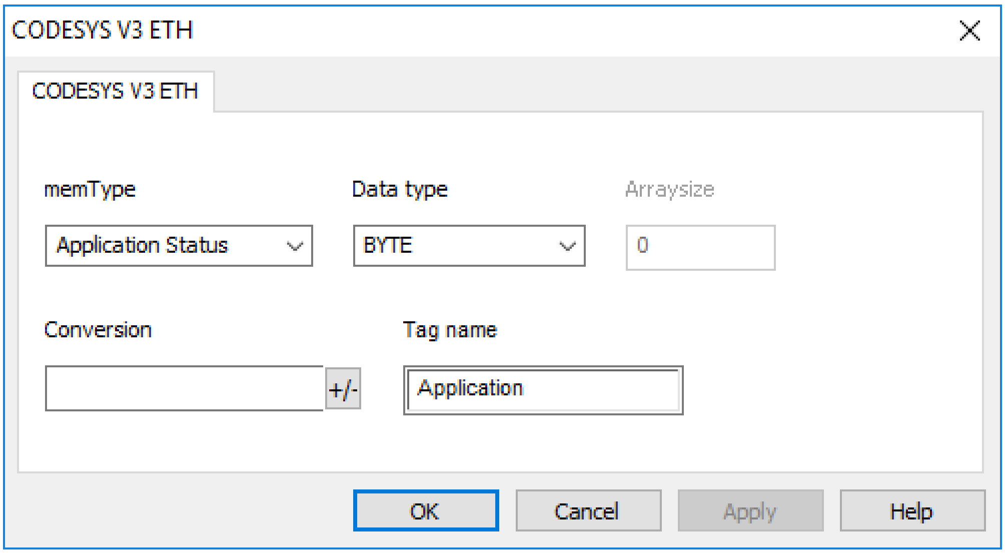
Application Status

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.



CODESYS V3 ETH

memType: Application Status
Data type: BYTE
Arraysize: 0
Conversion:
Tag name: Application

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.

Direct Serial

Direct Serial communication driver is a generic protocol that allows low level access to serial functions.

Using this protocol the application itself can realize some serial based protocol (RS-232/485/422) without requirement for a development of a dedicated protocol.

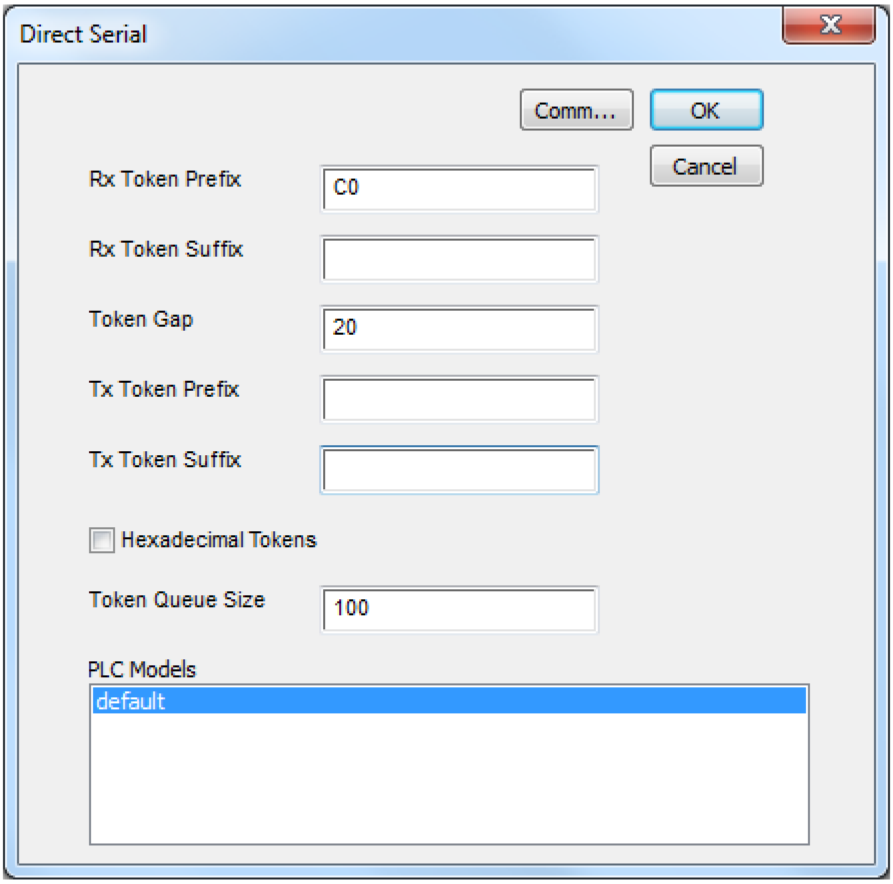
Protocol Editor Settings

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.



Rx Token Prefix: C0

Rx Token Suffix:

Token Gap: 20

Tx Token Prefix:

Tx Token Suffix:

☐ Hexadecimal Tokens

Token Queue Size: 100

PLC Models: default

Element	Description
Rx Token Prefix	Indicates the prefix for read token, as string specified by hexadecimal characters.
Rx Token Suffix	Indicates the suffix for read token, as string specified by hexadecimal characters.
Token Gap	Indicates the period between tokens, in milliseconds.
Tx Token Prefix	Indicates the prefix for sent token, as string specified by hexadecimal characters.
Tx Token Suffix	Indicates the suffix for sent token, as string specified by hexadecimal characters.
Hexadecimal Tokens	checked = tokens are in hexadecimal not checked = tokens are not in hexadecimal
Token Queue Size	Indicates the number of tokens in the queue, as an integer value from 1 to 10000 (default: 100)

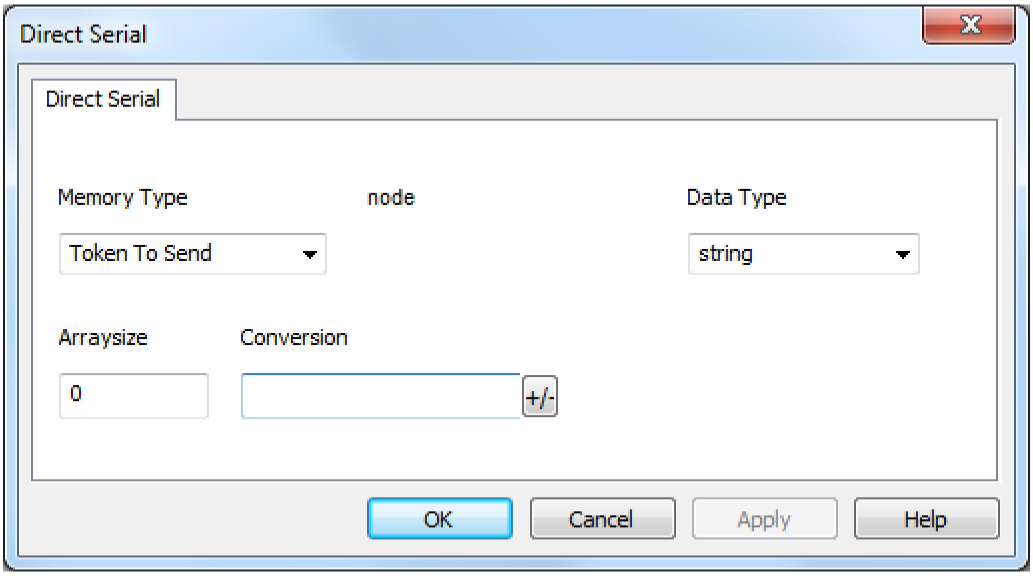
These parameters are determining the behavior of the driver during RX and TX operations, as defined in next paragraphs. In addition the standard communication parameters are available.

All protocols parameters can be overwritten at runtime using the appropriate memory types, so the complete setup can be achieved during runtime using Tags. Settings using memory types are saved to permanent storage using standard procedures. The “Serial Done” memory type is used in order that all set parameters are transferred to usage at once. If any of the serial parameter is changed the serial driver is re-programmed.

Tag Editor Settings

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

1. To add a tag, click **+**: a new line is added.
2. Select **Direct Serial** from the protocol list: tag definition dialog is displayed.



Direct Serial

Memory Type: Token To Send

node

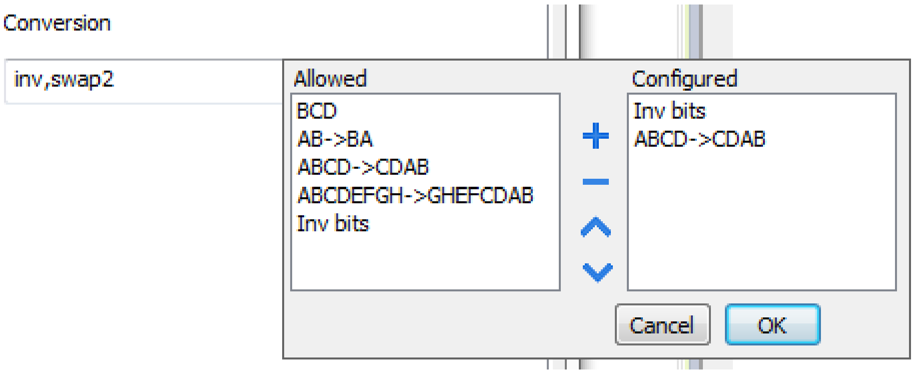
Data Type: string

Arraysize: 0

Conversion:

<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td><b>Memory Type</b></td><td>
<table>
<tr><th>Name</th><th>Datatype</th><th>Description</th></tr>
<tr><td><b>Token To Send</b></td><td>string</td><td>Write only. Writing on this memory type sends the given string to communication.</td></tr>
<tr><td><b>Token Received</b></td><td>string</td><td>Read only. Reading from this memory type gets the front token from the receiving queue.</td></tr>
<tr><td><b>Length of Token Received</b></td><td>unsignedInt</td><td>Read only. Returns the length in bytes of the front token from the receiving queue.</td></tr>
<tr><td><b>Tokens Available</b></td><td>unsignedInt</td><td>Read only. Gives the number of tokens in the receiving queue.</td></tr>
<tr><td><b>Token Acknowledge</b></td><td>boolean</td><td>Write only. Writing to this memory type removes the front token from the receiving queue.</td></tr>
<tr><td><b>Serial Baudrate</b></td><td>unsignedInt</td><td>Overrides serial baudrate parameter.</td></tr>
<tr><td><b>Serial Bits</b></td><td>unsignedByte</td><td>Overrides serial bits parameter.</td></tr>
<tr><td><b>Serial Stop Bits</b></td><td>unsignedByte</td><td>Overrides serial stop bit parameter.</td></tr>
<tr><td><b>Serial Parity</b></td><td>unsignedByte</td><td>Overrides serial parity parameter.</td></tr>
<tr><td><b>Serial Mode</b></td><td>unsignedByte</td><td>Overrides serial mode parameter.</td></tr>
<tr><td><b>Rx Token Prefix</b></td><td>string</td><td rowspan="7">Overrides protocol parameters. Check "<i>Protocol Editor Settings</i>" from details.</td></tr>
<tr><td><b>Rx Token Suffix</b></td><td>string</td></tr>
<tr><td><b>Token Gap</b></td><td>unsignedInt</td></tr>
<tr><td><b>Tx Token Prefix</b></td><td>string</td></tr>
<tr><td><b>Tx Token Suffix</b></td><td>string</td></tr>
<tr><td><b>Hexadecimal Tokens</b></td><td>boolean</td></tr>
<tr><td><b>Token Queue Size</b></td><td>unsignedInt</td></tr>
<tr><td><b>Serial Done</b></td><td>boolean</td><td>Writing to this memory type transfers all new values written in the other tags to protocol parameters, and to permanent storage.</td></tr>
</table>
</td></tr>
<tr><td><b>Data Type</b></td><td>
<table>
<tr><th>Data Type</th><th>Memory Space</th><th>Limits</th></tr>
<tr><td><b>boolean</b></td><td>1-bit data</td><td>0 ... 1</td></tr>
<tr><td><b>unsignedByte</b></td><td>8-bit data</td><td>0 ... 255</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>Data Type</th><th>Memory Space</th><th>Limits</th></tr>
<tr><td><b>unsignedInt</b></td><td>32-bit data</td><td>0 ... 4.2e9</td></tr>
<tr><td><b>string</b></td><td colspan="2">Array of elements containing character code defined by selected encoding</td></tr>
</table>
Note: to define arrays. select one of Data Type format followed by square brackets like “byte[]”, “short[]”…
</td></tr>
<tr><td><b>Arraysize</b></td><td>
<ul>
<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>
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.
</td></tr>
<tr><td><b>Conversion</b></td><td>
Conversion to be applied to the tag.



Conversion

inv,swap2

Allowed: BCD, AB->BA, ABCD->CDAB, ABCDEFGH->GHEFCDAB, Inv bits

Configured: Inv bits, ABCD->CDAB

Depending on data type selected, the list <b>Allowed</b> shows one or more conversion types.

<table>
<tr><th>Value</th><th>Description</th></tr>
<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 -> BA</b></td><td><b>swapnibbles</b>: Swap nibbles in a byte.</td></tr>
</table>
</td></tr>
</table>

Element	Description

Value	Description
	Example: 15D4 → 514D (in hexadecimal format) 5588 → 20813 (in decimal format)
ABCD -> CDAB	**swap2**: Swap bytes in a word. *Example:* 9ACC → CC9A (in hexadecimal format) 39628 → 52378 (in decimal format)
ABCDEFGH -> GHEFCDAB	**swap4**: Swap bytes in a double word. *Example:* 32FCFF54 → 54FFFC32 (in hexadecimal format) 855441236 → 1426062386 (in decimal format)
ABC...NOP -> OPM...DAB	**swap8**: Swap bytes in a long word. Example: 142.366 → -893553517.588905 (in decimal format) 0 10000000110 0001110010111011011001000101101000011100101011000001 → 1 10000011100 1010101000010100010110110110110010110110000100111101 (in binary format)
BCD	**bcd**: 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.

Implementation Details

Receiving algorithm

The protocol applies a separate thread that receives the characters from specified serial port.

When tokens (substrings) are identified they are put into the receiving queue (as strings).

Both ASCII and binary mode are available. When binary data can be present into receiving stream the **Hexadecimal Tokens** parameter can be set. In this case tokens are stored in queue using hex string coding (each byte is stored using two chars representing the hex value 0 to F). When defining the tags used to read tokens the appropriate string length should be computed considering the binary mode.

The **Token Queue Size** parameter specifies the maximum number of tokens saved into the queue. When the queue becomes full the oldest token is discarded.

The token identification is as follows:

- if the parameters specify a rx-prefix all characters before detecting the prefix are ignored
- if protocol specifies a rx-suffix it is used to detect the token end
- if rx-suffix is specified the parameter 'gap' specifies the timeout after which the token receiving is restarted
- if rx-suffix is not specified the parameter gap specifies the timeout that terminates the token (anything received up to this interval). If within this time the rx-prefix is detected again the token is ended and stored and reception of a new token is started

In summary we can have four combinations:

a. No rx-prefix and rx-suffix: the incoming stream is divided in tokens according to gap detection
b. Rx-prefix specified but no suffix: all the received chars before prefix are ignored. All the chars after prefix are stored in a token till the gap detection
c. Rx-prefix and Rx-suffix specified: all the chars between prefix and suffix are stored in a token. All the chars received before prefix or after suffix till the gap detection or till a new prefix are ignored
d. Rx-suffix specified but not RX-prefix: all the chars received till suffix are stored in a token. All the chars received after suffix till the gap detection are ignored

The rx-prefix and rx-suffix parameters are specified as hex strings, so any characters can be specified (like DLE STX CR LF etc...). i.e. to define the string “STR” as prefix the string “535452” must be used.

Before putting string to the receiving queue the prefix and suffix are removed (only 'payload' saved).

Transmission algorithm

The strings to be transmitted are prepared adding the “Tx-prefix” in front and the “Tx-suffix” in the end, if defined. Then the whole string is transmitted immediately.

Interface to user project

Reading a tag defined as **Token Received** gets the front string from the queue. If there are no new tokens an empty string is returned.

Reading a tag defined as **Length of Token Received** gets the length in bytes of the token.

Reading a tag defined as **Tokens Available** gets the number of tokens currently stored in the queue.

Writing to a tag defined as **Token Acknowledge** removes the token from queue and makes available the next token if present.

Writing to a tag defined as **Token To Send** means immediate sending, without any queue used.

JavaScript Interface

Beside Tag interface the user can access the protocol via JavaScript.

Although defined Tags can be accesses 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.

This interface provides the access to token queue and sending function. The following commands are supported:

Command	Description
put	Put the token to send contained in string parameter.
get	Get the received token.
get_token_length	Get the length of received token.
tokens_available	Get number of tokens received.
token_ack	Acknowledge reading token.

Using the command interface the following JS code should receive data:

```
var tagMgr = project.getWidget("_TagMgr");
var protID = "prot2"; // to be set according to protocol numbering

var avail = tagMgr.invokeProtocolCommand(protID, "tokens_available", "");
while (pasteInt(avail) > 0)
{
      var str = tagMgr.invokeProtocolCommand(protID, "get", ""); // get the next
token
      var status = tagMgr.invokeProtocolCommand(protID, "token_ack",""); //
acknowledge current token
      avail = tagMgr.invokeProtocolCommand(protID, "tokens_available",""); // get
number of available tokens in queue
}
```

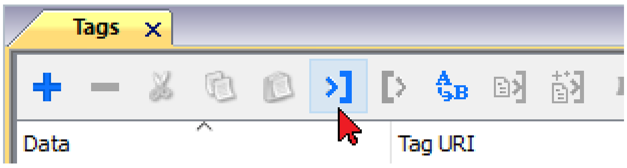
VCS access

The protocol supports the remote (virtual com port) access in exclusive mode.

When VCS is enabled the serial line usage is suspended and serial line becomes available for remote user. At the end the protocol is restarted. The content of the token queue is lost.

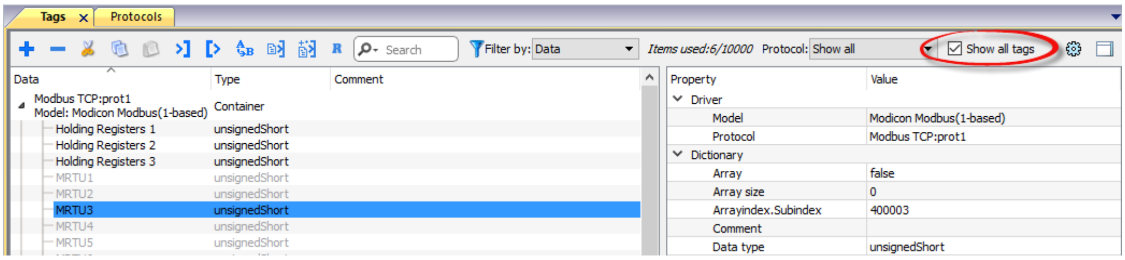
Tag Import

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



Locate the Tag Editor Exported symbol file and click **Open**.

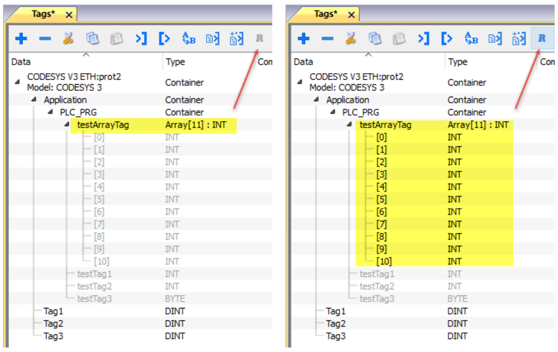
Tags included in the symbol file are listed in the tag dictionary. The tag dictionary is displayed at the bottom of the screen.



Filter by: Data ▼ *Items used:6/10000* Protocol: Show all ▼ ☑ Show all tags

Data	Type	Comment
Modbus TCP:prot1 Model: Modicon Modbus(1-based)	Container	
Holding Registers 1	unsignedShort	
Holding Registers 2	unsignedShort	
Holding Registers 3	unsignedShort	
MRTU1	unsignedShort	
MRTU2	unsignedShort	
MRTU3	unsignedShort	
MRTU4	unsignedShort	
MRTU5	unsignedShort	

Property	Value
Driver	
Model	Modicon Modbus(1-based)
Protocol	Modbus TCP:prot1
Dictionary	
Array	false
Array size	0
Arrayindex.Subindex	400003
Comment	
Data type	unsignedShort

Toolbar item	Description
	Import Tag(s). Select tags to be imported and click on this icon to add tags from tag dictionary to the project
	Update Tag(s). Click on this icon to update the tags in the project, due a new dictionary import.
R	Check this box to import all sub-elements of a tag. Example of both checked and unchecked result: 
Search Filter by: Tag name ▼	Searches tags in the dictionary basing on filter combo-box item selected.

Direct Socket

Direct Socket protocol is a generic protocol that allows low level access to socket functions.

Using this protocol the application itself can realize some IP based protocol without requirement for a development of a dedicated protocol.

Direct Socket protocol can be used as a standard (tag interface) protocol but also there is the appropriate implementation of DoCommand interface to enable using protocol from JavaScript.

The protocol can be used only with client socket type.

The protocol supports just one client socket. In case that application requires many sockets there could be many protocols installed, as the protocol supports multi-instance.

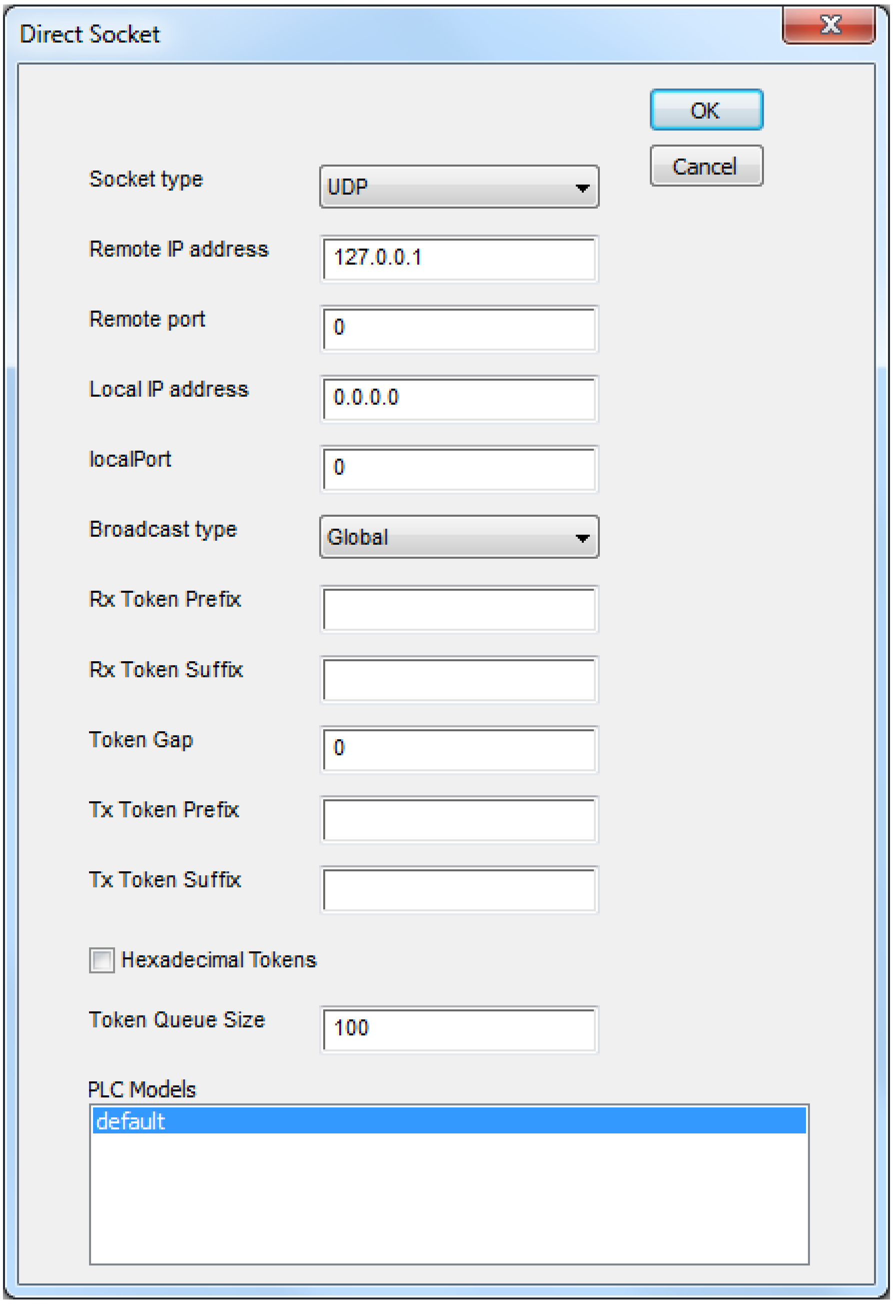
Protocol Editor Settings

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.



Socket type: UDP
Remote IP address: 127.0.0.1
Remote port: 0
Local IP address: 0.0.0.0
localPort: 0
Broadcast type: Global
Rx Token Prefix:
Rx Token Suffix:
Token Gap: 0
Tx Token Prefix:
Tx Token Suffix:
□ Hexadecimal Tokens
Token Queue Size: 100
PLC Models: default

Protocol parameters define a way how the connection is set and how the tokens are exchanged. The parameters are generally defined by the project. Many parameters can be accessed also as variables, allowing the runtime changes.

Element	Description
Socket type	Type of socket used for communication. Possible choices are UDP or TCP.
Remote IP Address	String. Indicates the IP address of remote device.
Remote Port	Integer. Indicates the port used by remote device.
Local IP Address	String. Indicates the IP address of local device. Mandatory for UDP usage.

Element	Description
Local Port	Integer. Indicates the port used by local device. Mandatory for UDP usage.
Broadcast Type	Type of broadcast used. Possible choices are Global or Local.

The following parameters are determining the behavior of the driver during RX and TX operations, as defined *Implementation Details* chapter.

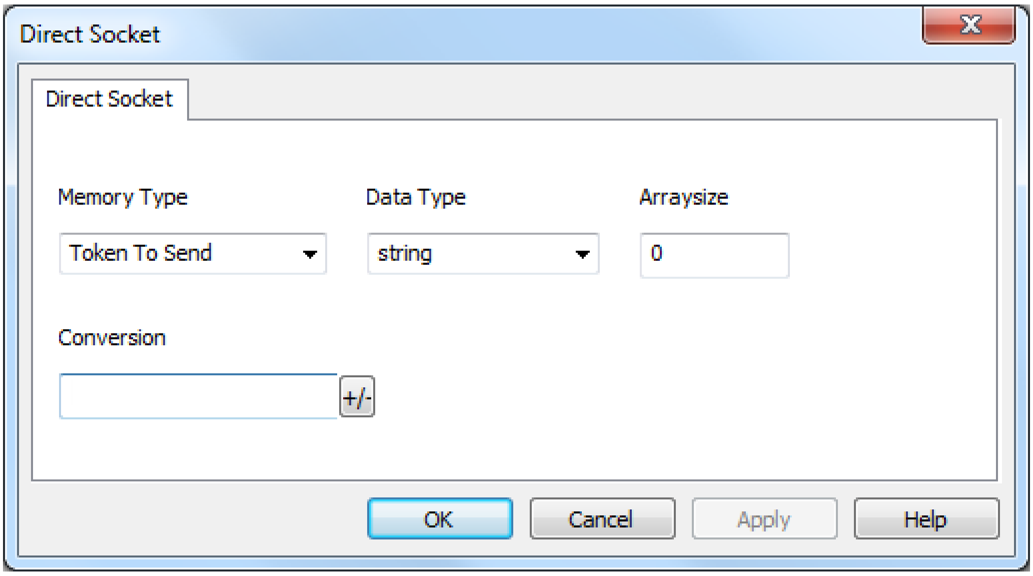
Element	Description
Rx Token Prefix	Indicates the prefix for read token, as string specified by hexadecimal characters.
Rx Token Suffix	Indicates the suffix for read token, as string specified by hexadecimal characters.
Token Gap	Indicates the period between tokens, in milliseconds.
Tx Token Prefix	Indicates the prefix for sent token, as string specified by hexadecimal characters.
Tx Token Suffix	Indicates the suffix for sent token, as string specified by hexadecimal characters.
Hexadecimal Tokens	checked = tokens are in hexadecimal not checked = tokens are not in hexadecimal
Token Queue Size	Indicates the number of tokens in the queue, as an integer value from 1 to 10000 (default: 100)

All protocols parameters can be overwritten at runtime using the appropriate memory types, so the complete setup can be achieved during runtime using Tags. Settings using memory types are saved to permanent storage using standard procedures. The “Done” memory type is used in order that all set parameters are transferred to usage at once. If any parameter is changed the driver is re-programmed.

Tag Editor Settings

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

1. To add a tag, click **+**: a new line is added.
2. Select **Direct Socket** from the protocol list: tag definition dialog is displayed.



Direct Socket

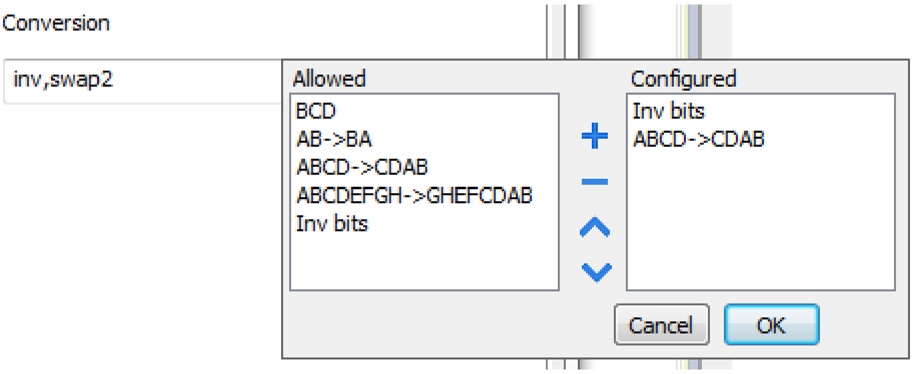
Memory Type: Token To Send

Data Type: string

Arraysize: 0

Conversion:

<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td><b>Memory Type</b></td><td>
<table>
<tr><th>Name</th><th>Datatype</th><th>Description</th></tr>
<tr><td><b>Token To Send</b></td><td>string</td><td>Write only. Writing on this memory type sends the given string to communication.</td></tr>
<tr><td><b>Token Received</b></td><td>string</td><td>Read only. Reading from this memory type gets the front token from the receiving queue.</td></tr>
<tr><td><b>Length of Token Received</b></td><td>unsignedInt</td><td>Read only. Returns the length in bytes of the front token from the receiving queue.</td></tr>
<tr><td><b>Tokens Available</b></td><td>unsignedInt</td><td>Read only. Gives the number of tokens in the receiving queue.</td></tr>
<tr><td><b>Token Acknowledge</b></td><td>boolean</td><td>Write only. Writing to this memory type removes the front token from the receiving queue.</td></tr>
<tr><td><b>Connect</b></td><td>boolean</td><td>Write only. Writing 1 to this variable enables the connection.</td></tr>
<tr><td><b>Connection Status</b></td><td>boolean</td><td>Read only. Gives the status of the connection In TCP mode it reflects effective connection with the peer. In UDP mode it is TRUE as soon as Connect is TRUE</td></tr>
<tr><td><b>Socket type</b></td><td>string</td><td>Overrides protocol parameters. Check "<i>Protocol Editor Settings</i>" from details.</td></tr>
<tr><td><b>Remote IP Address</b></td><td>string</td><td></td></tr>
<tr><td><b>Remote Port</b></td><td>unsignedShort</td><td></td></tr>
<tr><td><b>Local IP Address</b></td><td>strgin</td><td></td></tr>
<tr><td><b>Local Port</b></td><td>unsignedShort</td><td></td></tr>
<tr><td><b>Broadcast Type</b></td><td>string</td><td></td></tr>
<tr><td><b>Rx Token Prefix</b></td><td>string</td><td></td></tr>
<tr><td><b>Rx Token Suffix</b></td><td>string</td><td></td></tr>
<tr><td><b>Token Gap</b></td><td>unsignedInt</td><td></td></tr>
<tr><td><b>Tx Token Prefix</b></td><td>string</td><td></td></tr>
<tr><td><b>Tx Token Suffix</b></td><td>string</td><td></td></tr>
<tr><td><b>Hexadecimal Tokens</b></td><td>boolean</td><td></td></tr>
<tr><td><b>Token Queue Size</b></td><td>unsignedInt</td><td></td></tr>
<tr><td><b>Done</b></td><td>boolean</td><td>Writing to a tag of this memory type transfers all new values written in the other tags to protocol parameters, and to permanent storage.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td><b>Data Type</b></td><td>
<table>
<tr><th>Data Type</th><th>Memory Space</th><th>Limits</th></tr>
<tr><td><b>boolean</b></td><td>1-bit data</td><td>0 ... 1</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>string</b></td><td colspan="2">Array of elements containing character code defined by selected encoding</td></tr>
</table>
Note: to define arrays. select one of Data Type format followed by square brackets like “byte[]”, “short[]”…
</td></tr>
<tr><td><b>Arraysize</b></td><td>
<ul>
<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>
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.
</td></tr>
<tr><td><b>Conversion</b></td><td>
Conversion to be applied to the tag.

Conversion: inv,swap2
Allowed: BCD, AB->BA, ABCD->CDAB, ABCDEFGH->GHEFCDAB, Inv bits
Configured: Inv bits, ABCD->CDAB

Depending on data type selected, the list <b>Allowed</b> shows one or more conversion types.
</td></tr>
</table>

<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>Value</th><th>Description</th></tr>
<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 -> 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 -> 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 -> 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 -> OPM...DAB</b></td><td><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)</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>
</table>
Select conversion and click +. The selected item will be added to list <b>Configured</b>.</td></tr>
</table>

Element	Description
	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.

Implementation Details

Principle of operation

Protocol is parameterized by number of protocols parameters. The parameters define which socket type is used and the host address.

The data access is based on ‘tokens’. Token is data string that can be surrounded by prefix and suffix.

The protocol receiving process reads data from the specified IP/port and identifies tokens. Identified tokens are put to the queue from where they can be read by application. In the sending direction the application writes the token to protocol.

Protocol adds the defined tx_prefix/tx_suffix and sends data to the defined host.

Token extraction

The token extraction is slightly different for UDP and TCP sockets.

UDP protocols starts searching for tokens at the start of the received datagram. The search ends at the datagram end. If no rx_prefix is specified the token starts at datagram start. If no rx_suffix is specified the token ends on the datagram end. By specifying neither prefix nor suffix the whole datagram is delivered as a token. When both prefix and suffix are specified there can be many tokens extracted from a single datagram.

TCP protocol starts searching for tokens immediately after the previous rx_prefix. The search ends either when suffix is found or if the time gap without data is detected. If neither prefix nor suffix is specified the tokens will be all received data separated by time gaps.

The tokens can be plain ASCII strings, or hexadecimal strings. This is defined by the parameter ‘hex_tokens’.

The prefix/suffix strings must always be in hexadecimal format.

Common behavior

Both ASCII and binary mode are available. When binary data can be present into receiving stream the **Hexadecimal Tokens** parameter can be set. In this case tokens are stored in queue using hex string coding (each byte is stored using two chars representing the hex value 0 to F). When defining the tags used to read tokens the appropriate string length should be computed considering the binary mode.

The **Token Queue Size** parameter specifies the maximum number of tokens saved into the queue. When the queue becomes full the oldest token is discarded.

The token identification is as follows:

- if the parameters specify a rx-prefix all characters before detecting the prefix are ignored
- if protocol specifies a rx-suffix it is used to detect the token end
- if rx-suffix is specified the parameter 'gap' specifies the timeout after which the token receiving is restarted
- if rx-suffix is not specified the parameter gap specifies the timeout that terminates the token (anything received up to this interval). If within this time the rx-prefix is detected again the token is ended and stored and reception of a new token is started

In summary we can have four combinations:

a. No rx-prefix and rx-suffix: the incoming stream is divided in tokens according to gap detection
b. Rx-prefix specified but no suffix: all the received chars before prefix are ignored. All the chars after prefix are stored in a token till the gap detection
c. Rx-prefix and Rx-suffix specified: all the chars between prefix and suffix are stored in a token. All the chars received before prefix or after suffix till the gap detection or till a new prefix are ignored
d. Rx-suffix specified but not RX-prefix: all the chars received till suffix are stored in a token. All the chars received after suffix till the gap detection are ignored

The rx-prefix and rx-suffix parameters are specified as hex strings, so any characters can be specified (like DLE STX CR LF etc...). i.e. to define the string “STR” as prefix the string “535452” must be used

Before putting string to the receiving queue the prefix and suffix are removed (only 'payload' saved).

Interface to user project

Reading a tag defined as **Token Received** gets the front string from the queue. If there are no new tokens an empty string is returned.

Reading a tag defined as **Length of Token Received** gets the length in bytes of the token.

Reading a tag defined as **Tokens Available** gets the number of tokens currently stored in the queue.

Writing to a tag defined as **Token Acknowledge** removes the token from queue and makes available the next token if present.

Writing to a tag defined as **Token To Send** means immediate sending, without any queue used.

Data traffic control

The TCP sockets can be controlled by variables “Connect” and “Connection Status”. If the bool variable “Connect” is set the protocol will permanently try to make the connection to the specified host. If the TCP connection breaks it will be re-established automatically. If the variable “Connect” is false the protocol will wait. The state of connection can be read by variable Connection Status”.

For UDP there is no connection control. The socket is always connected and sends/receives data.

JavaScript Interface

Beside Tag interface the user can access the protocol via JavaScript.

Although defined Tags can be accesses 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.

This interface provides the access to token queue and sending function. The following commands are supported:

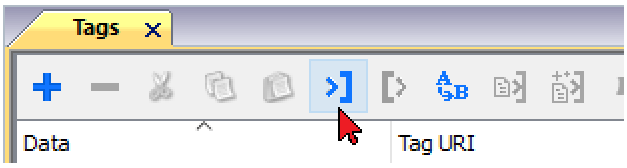
Command	Description
set_ip_address <ip> <port>	Specify the remote IP/port couple to use for connection. If protocol is already connected it is disconnected from current peer and re-connected to new one.
connect <ON|OFF>	Enables/disables the connection.
get_stat	Status of connection <connected|disconnected>.
put <string>	Put the token to send contained in string parameter.
get	Get the received token.
get_token_length	Get the length of received token.
tokens_available	Get number of tokens received.
token_ack	Acknowledge reading token.

Using the command interface the following JS code should receive data:

```
var tagMgr = project.getWidget("_TagMgr");
var protID = "prot2"; // to be set according to protocol numbering

var avail = tagMgr.invokeProtocolCommand(protID, "tokens_available", "");
while (pasteInt(avail) > 0)
{
      var str = tagMgr.invokeProtocolCommand(protID, "get", ""); // get the next
token
      var status = tagMgr.invokeProtocolCommand(protID, "token_ack",""); //
acknowledge current token
      avail = tagMgr.invokeProtocolCommand(protID, "tokens_available",""); // get
number of available tokens in queue
}
```

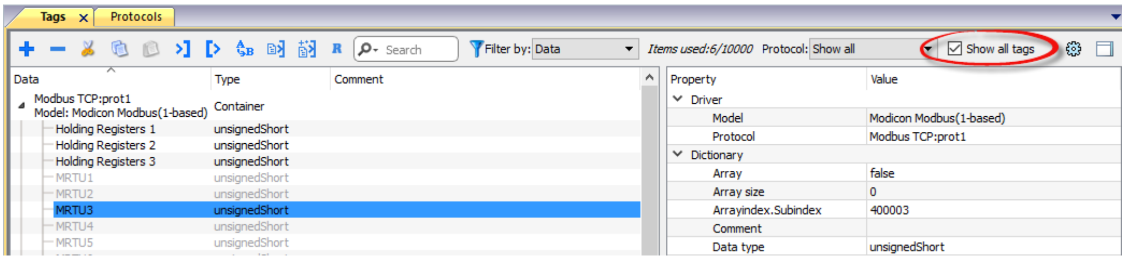
Tag Import

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



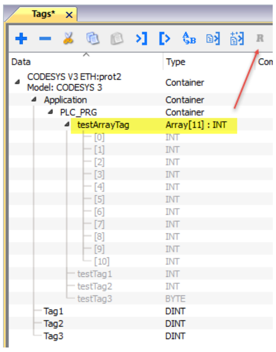
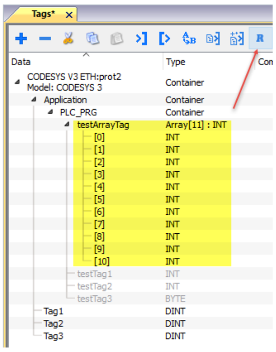
Locate the Tag Editor Exported symbol file and click **Open**.

Tags included in the symbol file are listed in the tag dictionary. The tag dictionary is displayed at the bottom of the screen.



Data	Type	Comment
Modbus TCP:prot1 Model: Modicon Modbus(1-based)	Container	
Holding Registers 1	unsignedShort	
Holding Registers 2	unsignedShort	
Holding Registers 3	unsignedShort	
MRTU1	unsignedShort	
MRTU2	unsignedShort	
MRTU3	unsignedShort	
MRTU4	unsignedShort	
MRTU5	unsignedShort	

Property	Value
Driver	
Model	Modicon Modbus(1-based)
Protocol	Modbus TCP:prot1
Dictionary	
Array	false
Array size	0
Arrayindex.Subindex	400003
Comment	
Data type	unsignedShort

Toolbar item	Description
	Import Tag(s). Select tags to be imported and click on this icon to add tags from tag dictionary to the project
	Update Tag(s). Click on this icon to update the tags in the project, due a new dictionary import.
R	Check this box to import all sub-elements of a tag. Example of both checked and unchecked result:  Tags* Data / Type / Com CODESYS V3 ETH:prot2 Model: CODESYS 3 — Container Application — Container PLC_PRG — Container testArrayTag — Array[11] : INT [0] INT [1] INT [2] INT [3] INT [4] INT [5] INT [6] INT [7] INT [8] INT [9] INT [10] INT testTag1 INT testTag2 INT testTag3 BYTE Tag1 DINT Tag2 DINT Tag3 DINT  Tags* Data / Type / Con CODESYS V3 ETH:prot2 Model: CODESYS 3 — Container Application — Container PLC_PRG — Container testArrayTag — Array[11] : INT [0] INT [1] INT [2] INT [3] INT [4] INT [5] INT [6] INT [7] INT [8] INT [9] INT [10] INT testTag1 INT testTag2 INT testTag3 BYTE Tag1 DINT Tag2 DINT Tag3 DINT
Search Filter by: Tag name	Searches tags in the dictionary basing on filter combo-box item selected.

Ethernet/IP CIP

The protocol has been implemented according to the published Ethernet/IP specifications (available from 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.

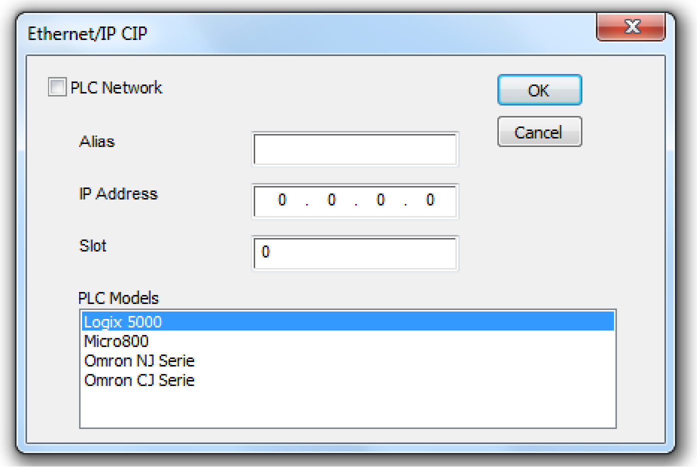
Protocol Editor Settings

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.


Ethernet/IP CIP

□ PLC Network

Alias

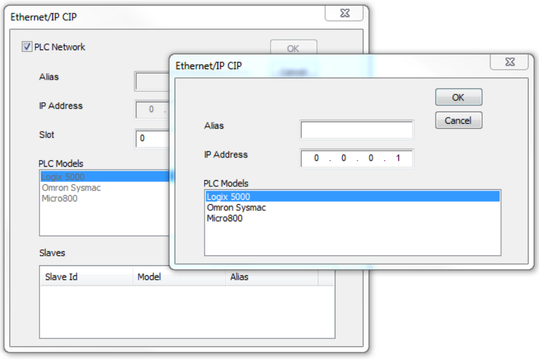
IP Address 0 . 0 . 0 . 0

Slot 0

PLC Models

Logix 5000
Micro800
Omron NJ Serie
Omron CJ Serie

Field	Description
Alias	Name identifying nodes in network configurations. The name will be added as a prefix to each tag name imported for each network node.
IP Address	Ethernet IP address of the controller.
Slot	CPU slot number for Logix 5000 models (typically 0). Refer to the controller documentation for further details.

Field	Description
PLC Models	PLC model used to import tags file.
PLC Network	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.

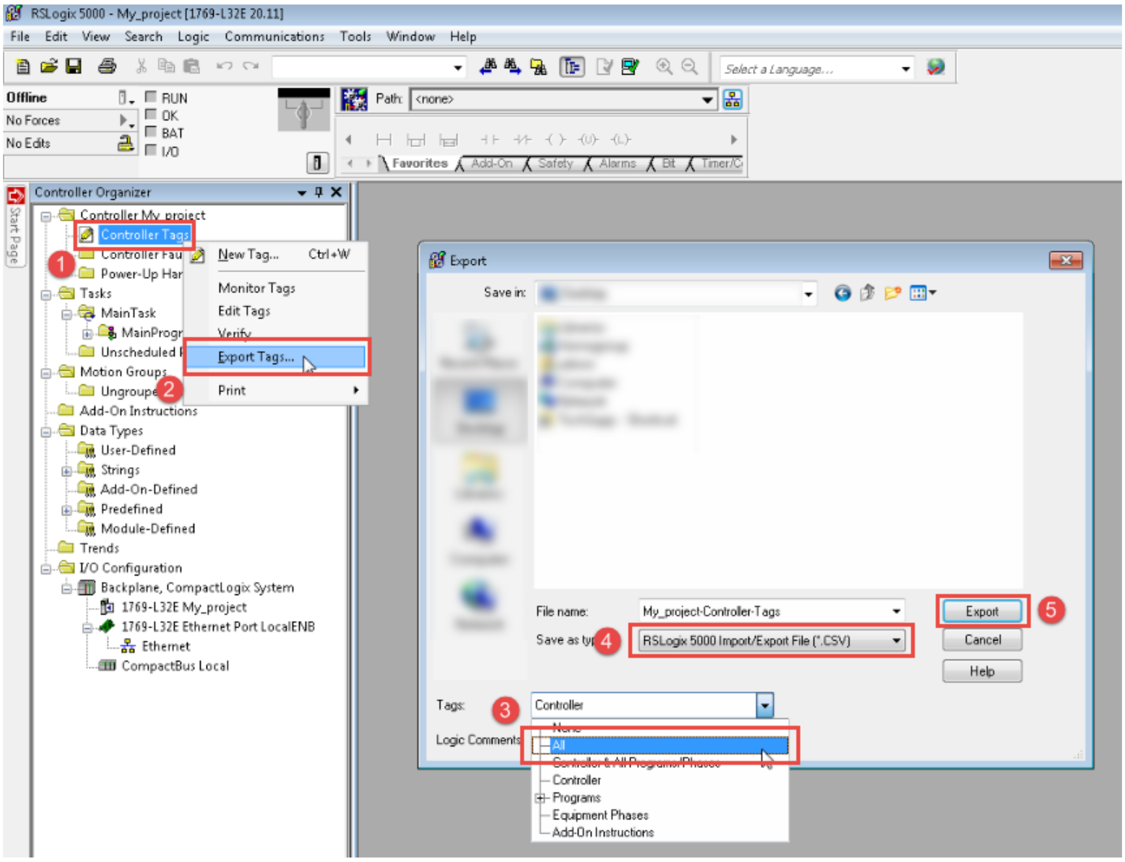
The implementation of the Ethernet/IP driver also supports access to structured data types which can be imported from .L5X files.

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.



File name: My_project-Controller-Tags

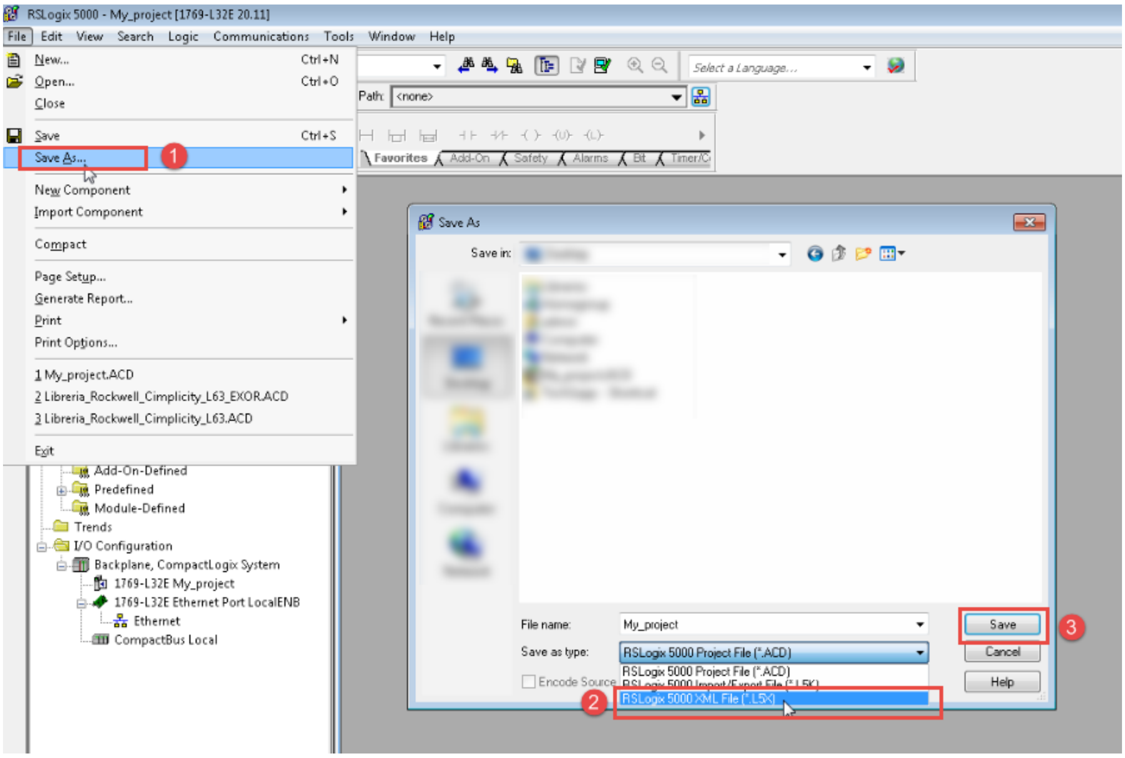
Save as type: RSLogix 5000 Import/Export File (*.CSV)

Tags: Controller

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.



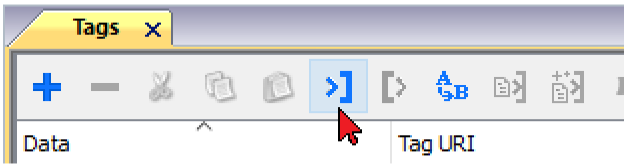
File name: My_project

Save as type: RSLogix 5000 Project File (*.ACD)

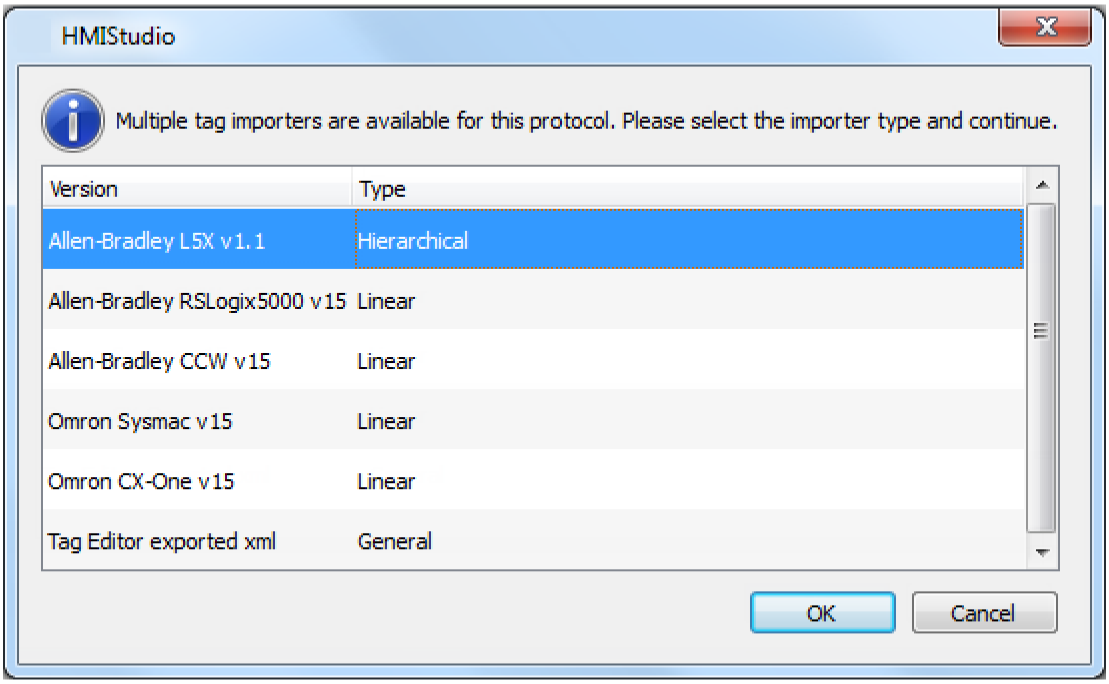
RSLogix 5000 Project File (*.ACD)
RSLogix 5000 Import/Export File (*.L5K)
RSLogix 5000 XML File (*.L5X)

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.



HMIStudio

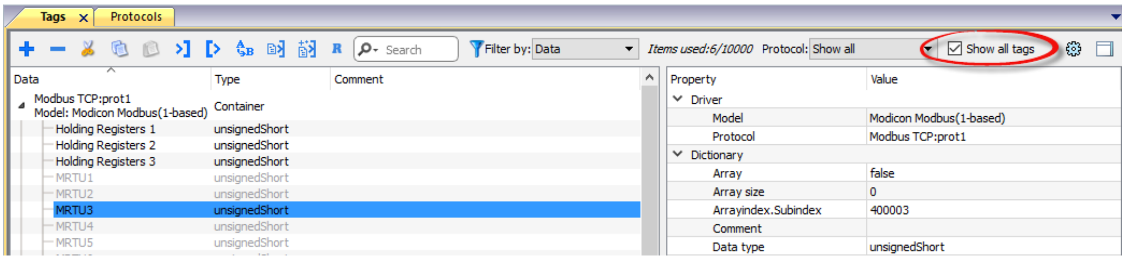
Multiple tag importers are available for this protocol. Please select the importer type and continue.

Version	Type
Allen-Bradley L5X v1.1	Hierarchical
Allen-Bradley RSLogix5000 v15	Linear
Allen-Bradley CCW v15	Linear
Omron Sysmac v15	Linear
Omron CX-One v15	Linear
Tag Editor exported xml	General

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.

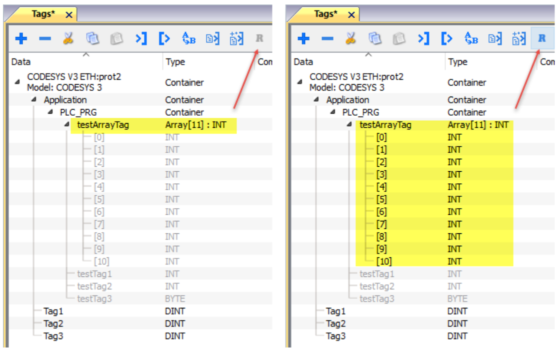
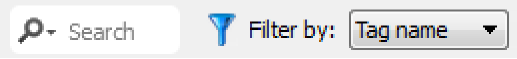


Filter by: Data   *Items used:6/10000*   Protocol: Show all   ☑ Show all tags

Data	Type	Comment
Modbus TCP:prot1 Model: Modicon Modbus(1-based)	Container	
Holding Registers 1	unsignedShort	
Holding Registers 2	unsignedShort	
Holding Registers 3	unsignedShort	
MRTU1	unsignedShort	
MRTU2	unsignedShort	
MRTU3	unsignedShort	
MRTU4	unsignedShort	
MRTU5	unsignedShort	

Property	Value
Driver	
Model	Modicon Modbus(1-based)
Protocol	Modbus TCP:prot1
Dictionary	
Array	false
Array size	0
Arrayindex.Subindex	400003
Comment	
Data type	unsignedShort

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

Toolbar item	Description
	 Tags* Data / Type / Con CODESYS V3 ETH:prot2 Model: CODESYS 3 — Container Application — Container PLC_PRG — Container testArrayTag — Array[11] : INT [0] — INT [1] — INT [2] — INT [3] — INT [4] — INT [5] — INT [6] — INT [7] — INT [8] — INT [9] — INT [10] — INT testTag1 — INT testTag2 — INT testTag3 — BYTE Tag1 — DINT Tag2 — DINT Tag3 — DINT
 	Searches tags in the dictionary basing on filter combo-box 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

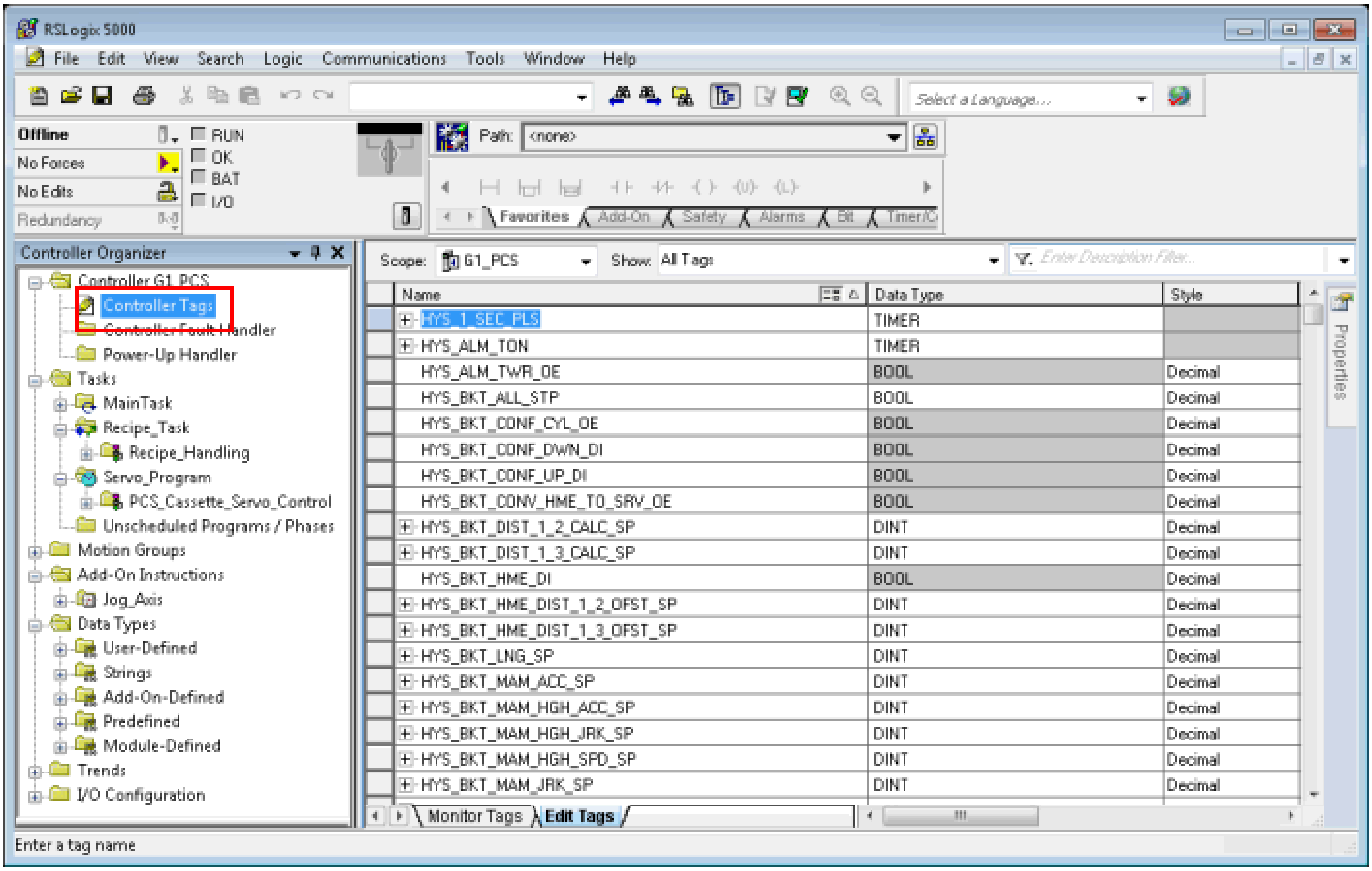
RSLogix 5000 allows you to define Tags with several data types.

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:

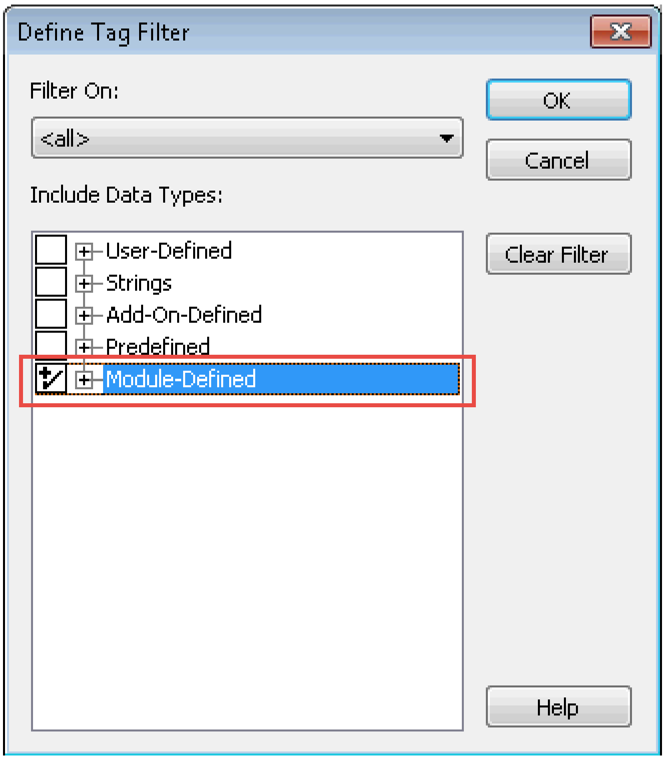
1. From the **Controller Organizer** pane, select **Controller Tags**.



Scope: G1_PCS Show: All Tags

Name	Data Type	Style
HYS_1_SEC_PLS	TIMER	
HYS_ALM_TON	TIMER	
HYS_ALM_TWR_OE	BOOL	Decimal
HYS_BKT_ALL_STP	BOOL	Decimal
HYS_BKT_CONF_CYL_OE	BOOL	Decimal
HYS_BKT_CONF_DWN_DI	BOOL	Decimal
HYS_BKT_CONF_UP_DI	BOOL	Decimal
HYS_BKT_CONV_HME_TO_SRV_OE	BOOL	Decimal
HYS_BKT_DIST_1_2_CALC_SP	DINT	Decimal
HYS_BKT_DIST_1_3_CALC_SP	DINT	Decimal
HYS_BKT_HME_DI	BOOL	Decimal
HYS_BKT_HME_DIST_1_2_OFST_SP	DINT	Decimal
HYS_BKT_HME_DIST_1_3_OFST_SP	DINT	Decimal
HYS_BKT_LNG_SP	DINT	Decimal
HYS_BKT_MAM_ACC_SP	DINT	Decimal
HYS_BKT_MAM_HGH_ACC_SP	DINT	Decimal
HYS_BKT_MAM_HGH_JRK_SP	DINT	Decimal
HYS_BKT_MAM_HGH_SPD_SP	DINT	Decimal
HYS_BKT_MAM_JRK_SP	DINT	Decimal

2. Filter tags to display only **Module-Defined** Tags.

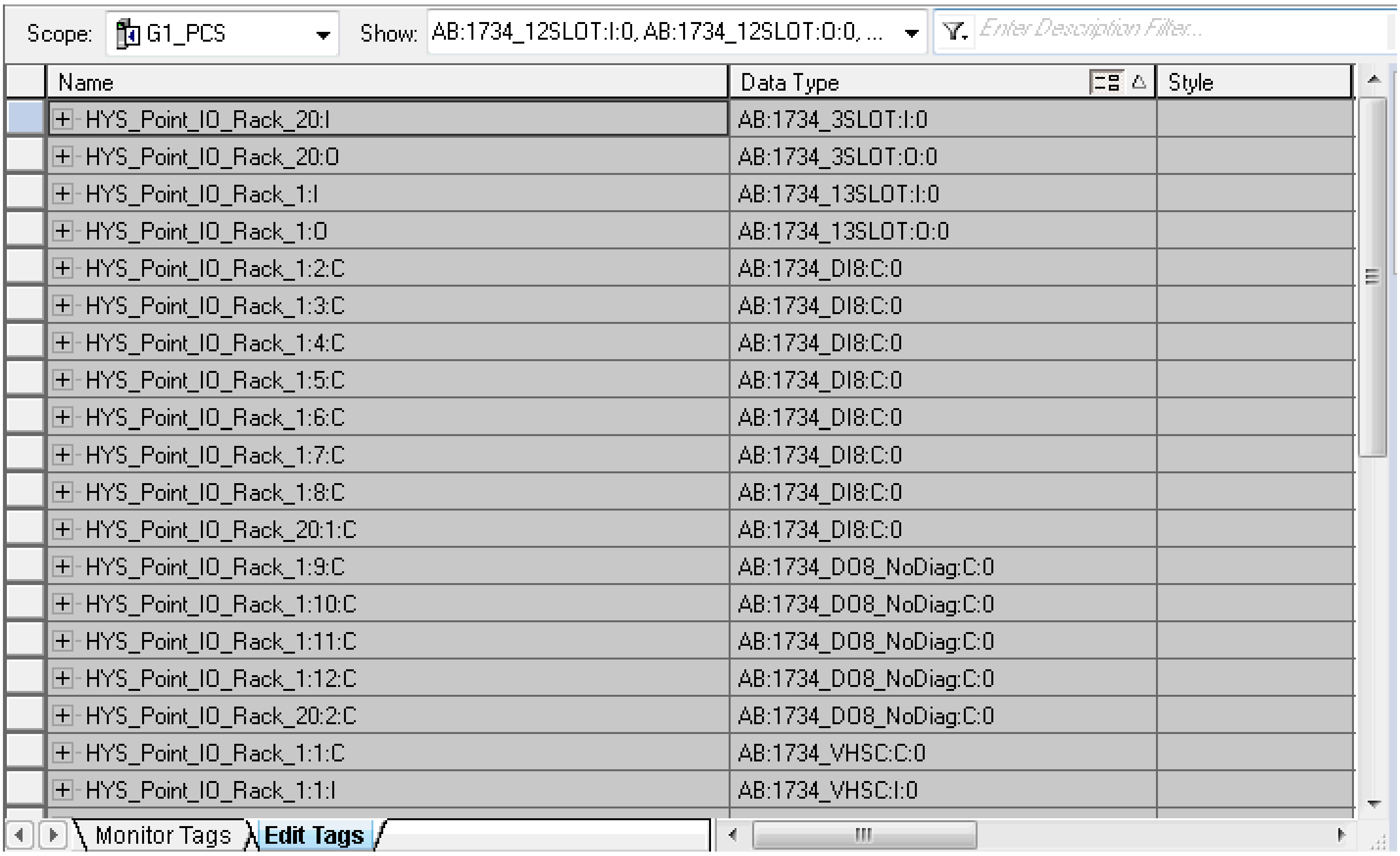


Filter On: <all>

Include Data Types:

- □ User-Defined
- □ Strings
- □ Add-On-Defined
- □ Predefined
- ☑ Module-Defined

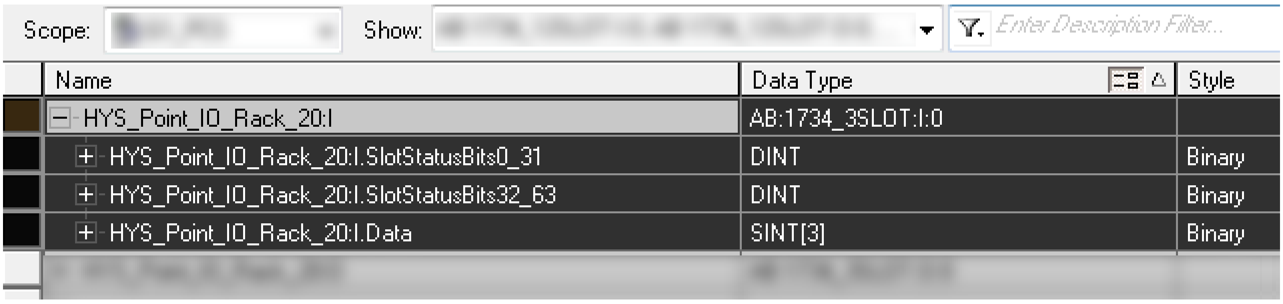
Only tags (alias) with data type belonging to optional I/O Modules will be displayed.



Scope: G1_PCS   Show: AB:1734_12SLOT:I:0, AB:1734_12SLOT:0:0, ...   Enter Description Filter...

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:

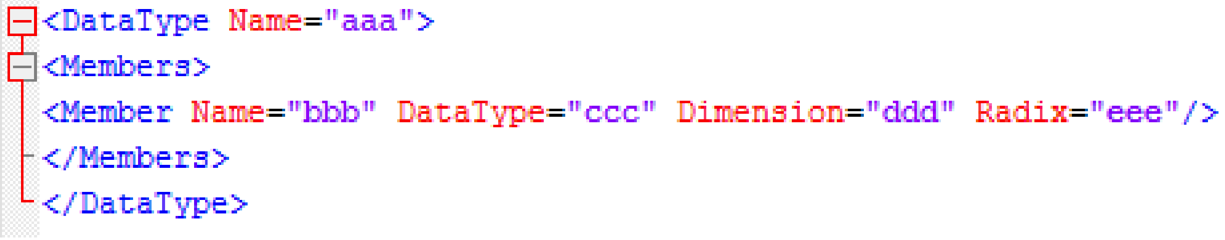


Scope: Show: Enter Description Filter...

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:

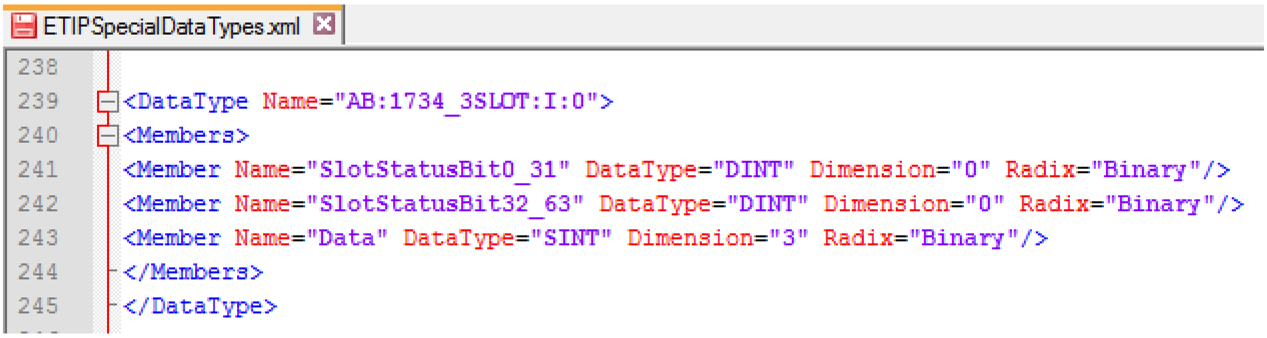


```
<DataType Name="aaa">
<Members>
<Member Name="bbb" DataType="ccc" Dimension="ddd" Radix="eee"/>
</Members>
</DataType>
```

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:



ETIPSpecialDataTypes.xml

```
<DataType Name="AB:1734_3SLOT:I:0">
<Members>
<Member Name="SlotStatusBit0_31" DataType="DINT" Dimension="0" Radix="Binary"/>
<Member Name="SlotStatusBit32_63" DataType="DINT" Dimension="0" Radix="Binary"/>
<Member Name="Data" DataType="SINT" Dimension="3" Radix="Binary"/>
</Members>
</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.

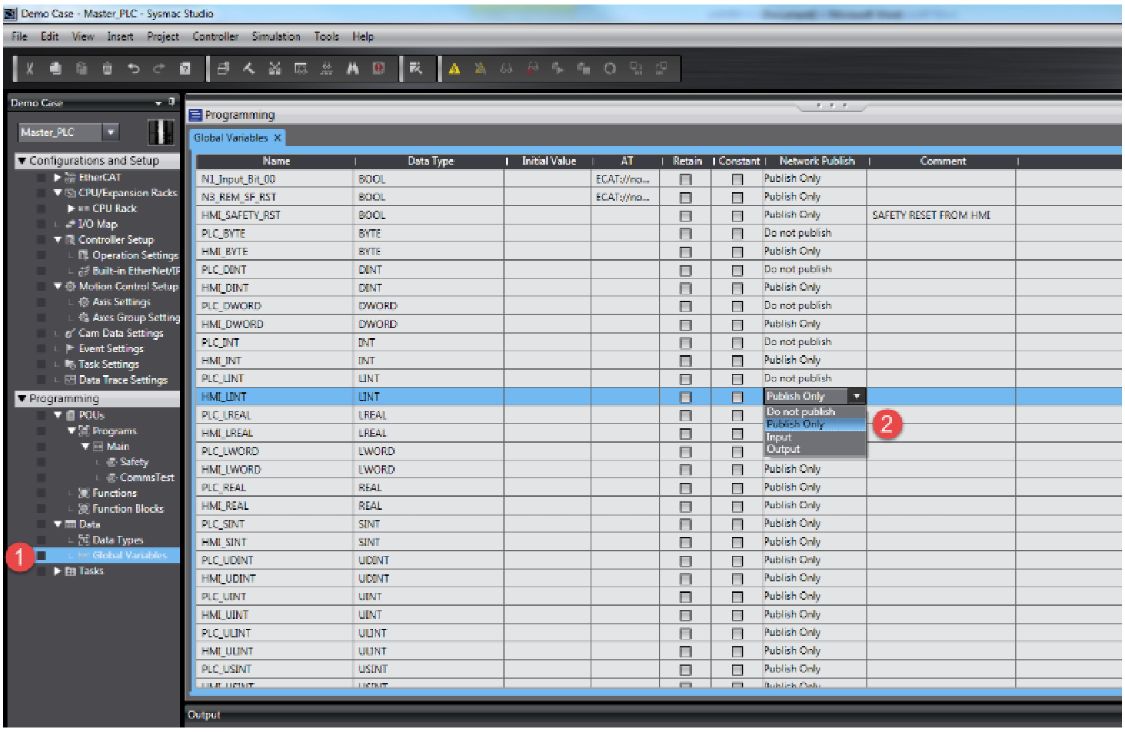
The project loaded on the HMI device must refer to the Tag names assigned in the programming software at development time. The Tag Editor supports direct import of the Tag file generated by Sysmac Studio software in .NJF format or generated by CX-One in the .CJF format.

All Tags to be accessed by the HMI device must be declared as Global Variables.

Export NJF files using Sysmac Studio

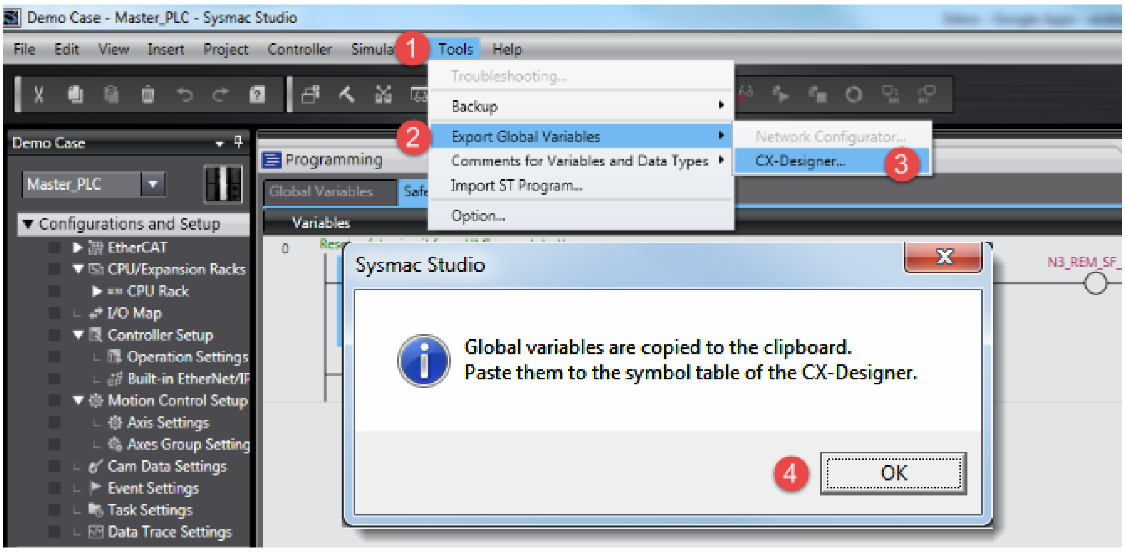
To export the **.NJF** Tag file:

1. In Sysmac Studio declare Tags as **Global Variables**.
2. Set the **Network Publish** attribute to **Publish Only**.



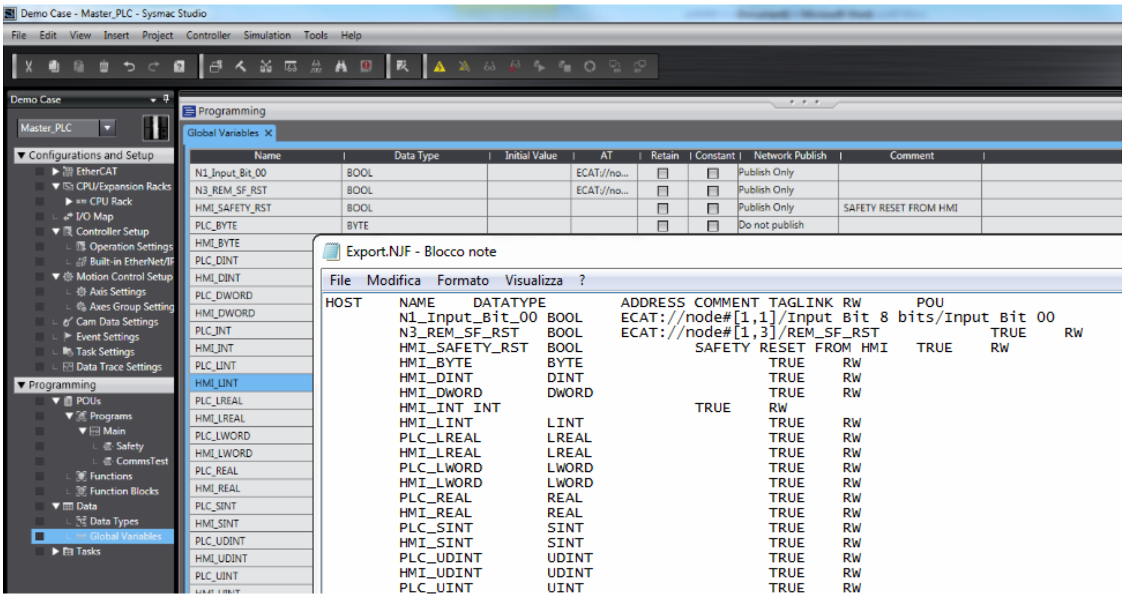
Name	Data Type	Initial Value	AT	Retain	Constant	Network Publish	Comment
N1_Input_Bit_00	BOOL		ECAT://no...	□	□	Publish Only	
N3_REM_SF_RST	BOOL		ECAT://no...	□	□	Publish Only	
HMI_SAFETY_RST	BOOL			□	□	Publish Only	SAFETY RESET FROM HMI
PLC_BYTE	BYTE			□	□	Do not publish	
HMI_BYTE	BYTE			□	□	Publish Only	
PLC_DINT	DINT			□	□	Do not publish	
HMI_DINT	DINT			□	□	Publish Only	
PLC_DWORD	DWORD			□	□	Do not publish	
HMI_DWORD	DWORD			□	□	Publish Only	
PLC_INT	INT			□	□	Do not publish	
HMI_INT	INT			□	□	Publish Only	
PLC_LINT	LINT			□	□	Do not publish	
HMI_LINT	LINT			□	□	Publish Only	
PLC_LREAL	LREAL			□	□		
HMI_LREAL	LREAL			□	□		
PLC_LWORD	LWORD			□	□		
HMI_LWORD	LWORD			□	□	Publish Only	
PLC_REAL	REAL			□	□	Publish Only	
HMI_REAL	REAL			□	□	Publish Only	
PLC_SINT	SINT			□	□	Publish Only	
HMI_SINT	SINT			□	□	Publish Only	
PLC_UDINT	UDINT			□	□	Publish Only	
HMI_UDINT	UDINT			□	□	Publish Only	
PLC_UINT	UINT			□	□	Publish Only	
HMI_UINT	UINT			□	□	Publish Only	
PLC_ULINT	ULINT			□	□	Publish Only	
HMI_ULINT	ULINT			□	□	Publish Only	
PLC_USINT	USINT			□	□	Publish Only	
HMI_USINT	USINT			□	□	Publish Only	

2. From the **Tools** menu, choose **Export Global Variables > CX-Designer**.



Global variables are copied to the clipboard.
Paste them to the symbol table of the CX-Designer.

3. Click **OK** to confirm.
4. Cut and paste the content of the clipboard in any text editor.



Name	Data Type	Initial Value	AT	Retain	Constant	Network Publish	Comment
N1_Input_Bit_00	BOOL		ECAT://no...	□	□	Publish Only	
N3_REM_SF_RST	BOOL		ECAT://no...	□	□	Publish Only	
HMI_SAFETY_RST	BOOL			□	□	Publish Only	SAFETY RESET FROM HMI
PLC_BYTE	BYTE			□	□	Do not publish	
HMI_BYTE							
PLC_DINT							
HMI_DINT							
PLC_DWORD							
HMI_DWORD							
PLC_INT							
HMI_INT							
PLC_LINT							
HMI_LINT							
PLC_LREAL							
HMI_LREAL							
PLC_LWORD							
HMI_LWORD							
PLC_REAL							
HMI_REAL							
PLC_SINT							
HMI_SINT							
PLC_UDINT							
HMI_UDINT							
PLC_UINT							

Export.NJF - Blocco note

```
HOST     NAME     DATATYPE        ADDRESS COMMENT TAGLINK RW      POU
         N1_Input_Bit_00 BOOL     ECAT://node#[1,1]/Input Bit 8 bits/Input Bit 00
         N3_REM_SF_RST   BOOL     ECAT://node#[1,3]/REM_SF_RST             TRUE     RW
         HMI_SAFETY_RST  BOOL             SAFETY RESET FROM HMI   TRUE     RW
         HMI_BYTE        BYTE                     TRUE     RW
         HMI_DINT        DINT                     TRUE     RW
         HMI_DWORD       DWORD                    TRUE     RW
         HMI_INT INT                      TRUE     RW
         HMI_LINT        LINT                     TRUE     RW
         PLC_LREAL       LREAL                    TRUE     RW
         HMI_LREAL       LREAL                    TRUE     RW
         PLC_LWORD       LWORD                    TRUE     RW
         HMI_LWORD       LWORD                    TRUE     RW
         PLC_REAL        REAL                     TRUE     RW
         HMI_REAL        REAL                     TRUE     RW
         PLC_SINT        SINT                     TRUE     RW
         HMI_SINT        SINT                     TRUE     RW
         PLC_UDINT       UDINT                    TRUE     RW
         HMI_UDINT       UDINT                    TRUE     RW
         PLC_UINT        UINT                     TRUE     RW
```

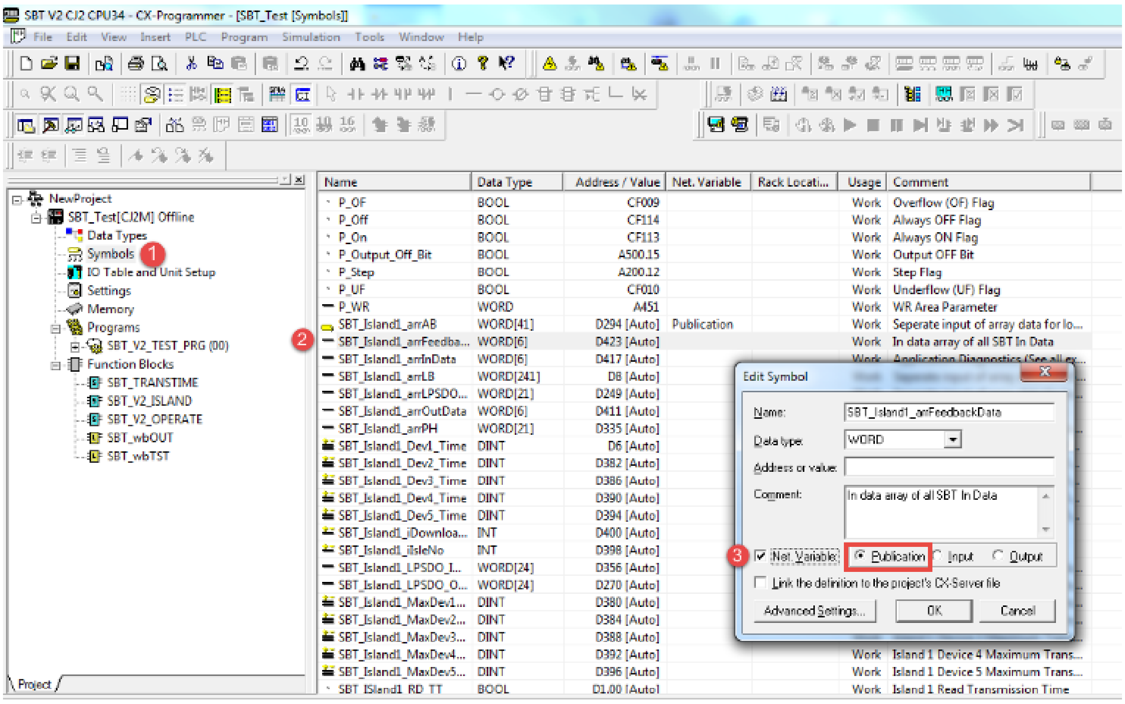
4. Save the file as **.NJF**.

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**.



Name	Data Type	Address / Value	Net. Variable	Rack Locati...	Usage	Comment
P_OF	BOOL	CF009			Work	Overflow (OF) Flag
P_Off	BOOL	CF114			Work	Always OFF Flag
P_On	BOOL	CF113			Work	Always ON Flag
P_Output_Off_Bit	BOOL	A500.15			Work	Output OFF Bit
P_Step	BOOL	A200.12			Work	Step Flag
P_UF	BOOL	CF010			Work	Underflow (UF) Flag
P_WR	WORD	A451			Work	WR Area Parameter
SBT_Island1_arrAB	WORD[41]	D294 [Auto]	Publication		Work	Seperate input of array data for lo...
SBT_Island1_arrFeedba...	WORD[6]	D423 [Auto]			Work	In data array of all SBT In Data
SBT_Island1_arrInData	WORD[6]	D417 [Auto]			Work	Application Diagnostics (See all ex...
SBT_Island1_arrLB	WORD[241]	D8 [Auto]				
SBT_Island1_arrLPSDO...	WORD[21]	D249 [Auto]				
SBT_Island1_arrOutData	WORD[6]	D411 [Auto]				
SBT_Island1_arrPH	WORD[21]	D335 [Auto]				
SBT_Island1_Dev1_Time	DINT	D6 [Auto]				
SBT_Island1_Dev2_Time	DINT	D382 [Auto]				
SBT_Island1_Dev3_Time	DINT	D386 [Auto]				
SBT_Island1_Dev4_Time	DINT	D390 [Auto]				
SBT_Island1_Dev5_Time	DINT	D394 [Auto]				
SBT_Island1_iDownloa...	INT	D400 [Auto]				
SBT_Island1_iIsleNo	INT	D398 [Auto]				
SBT_Island1_LPSDO_I...	WORD[24]	D356 [Auto]				
SBT_Island1_LPSDO_O...	WORD[24]	D270 [Auto]				
SBT_Island1_MaxDev1...	DINT	D380 [Auto]				
SBT_Island1_MaxDev2...	DINT	D384 [Auto]				
SBT_Island1_MaxDev3...	DINT	D388 [Auto]				
SBT_Island1_MaxDev4...	DINT	D392 [Auto]			Work	Island 1 Device 4 Maximum Trans...
SBT_Island1_MaxDev5...	DINT	D396 [Auto]			Work	Island 1 Device 5 Maximum Trans...
SBT_ISland1_RD_TT	BOOL	D1.00 [Auto]			Work	Island 1 Read Transmission Time

Edit Symbol

Name: SBT_Island1_arrFeedbackData

Data type: WORD

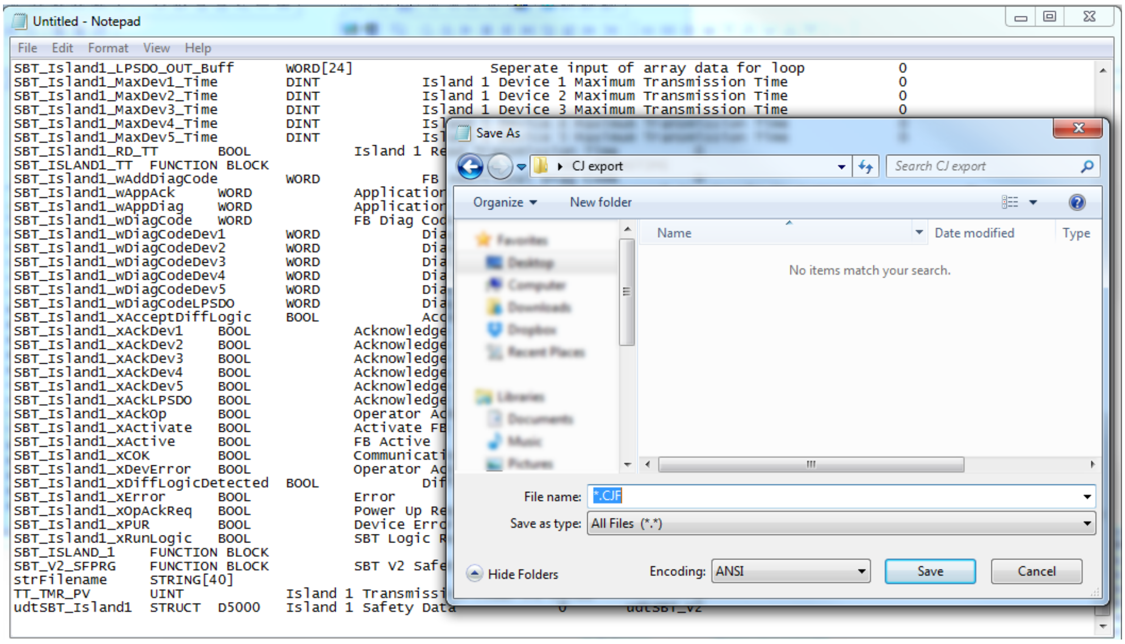
Address or value:

Comment: In data array of all SBT In Data

☑ Net. Variable: ● Publication ○ Input ○ Output

□ Link the definition to the project's CX-Server file

3. Copy and paste all the Tags in any text editor.



```
SBT_Island1_LPSDO_OUT_Buff      WORD[24]                Seperate input of array data for loop       0
SBT_Island1_MaxDev1_Time        DINT            Island 1 Device 1 Maximum Transmission Time         0
SBT_Island1_MaxDev2_Time        DINT            Island 1 Device 2 Maximum Transmission Time         0
SBT_Island1_MaxDev3_Time        DINT            Island 1 Device 3 Maximum Transmission Time         0
SBT_Island1_MaxDev4_Time        DINT            Isl
SBT_Island1_MaxDev5_Time        DINT            Isl
SBT_Island1_RD_TT       BOOL            Island 1 Re
SBT_ISLAND1_TT  FUNCTION BLOCK
SBT_Island1_wAddDiagCode        WORD            FB
SBT_Island1_wAppAck     WORD            Application
SBT_Island1_wAppDiag    WORD            Application
SBT_Island1_wDiagCode   WORD            FB Diag Cod
SBT_Island1_wDiagCodeDev1       WORD            Dia
SBT_Island1_wDiagCodeDev2       WORD            Dia
SBT_Island1_wDiagCodeDev3       WORD            Dia
SBT_Island1_wDiagCodeDev4       WORD            Dia
SBT_Island1_wDiagCodeDev5       WORD            Dia
SBT_Island1_wDiagCodeLPSDO      WORD            Dia
SBT_Island1_xAcceptDiffLogic    BOOL            Acc
SBT_Island1_xAckDev1    BOOL            Acknowledge
SBT_Island1_xAckDev2    BOOL            Acknowledge
SBT_Island1_xAckDev3    BOOL            Acknowledge
SBT_Island1_xAckDev4    BOOL            Acknowledge
SBT_Island1_xAckDev5    BOOL            Acknowledge
SBT_Island1_xAckLPSDO   BOOL            Acknowledge
SBT_Island1_xAckOp      BOOL            Operator Ac
SBT_Island1_xActivate   BOOL            Activate FB
SBT_Island1_xActive     BOOL            FB Active
SBT_Island1_xCOK        BOOL            Communicati
SBT_Island1_xDevError   BOOL            Operator Ac
SBT_Island1_xDiffLogicDetected  BOOL            Dif
SBT_Island1_xError      BOOL            Error
SBT_Island1_xOpAckReq   BOOL            Power Up Re
SBT_Island1_xPUR        BOOL            Device Erro
SBT_Island1_xRunLogic   BOOL            SBT Logic R
SBT_ISLAND_1    FUNCTION BLOCK          SBT V2 Safe
SBT_V2_SFPRG    FUNCTION BLOCK          SBT V2 Safe
strFilename     STRING[40]
TT_TMR_PV       UINT            Island 1 Transmissi
udtSBT_Island1  STRUCT  D5000   Island 1 Safety Data
```

File name: *.CJF

Save as type: All Files (*.*)

Encoding: ANSI

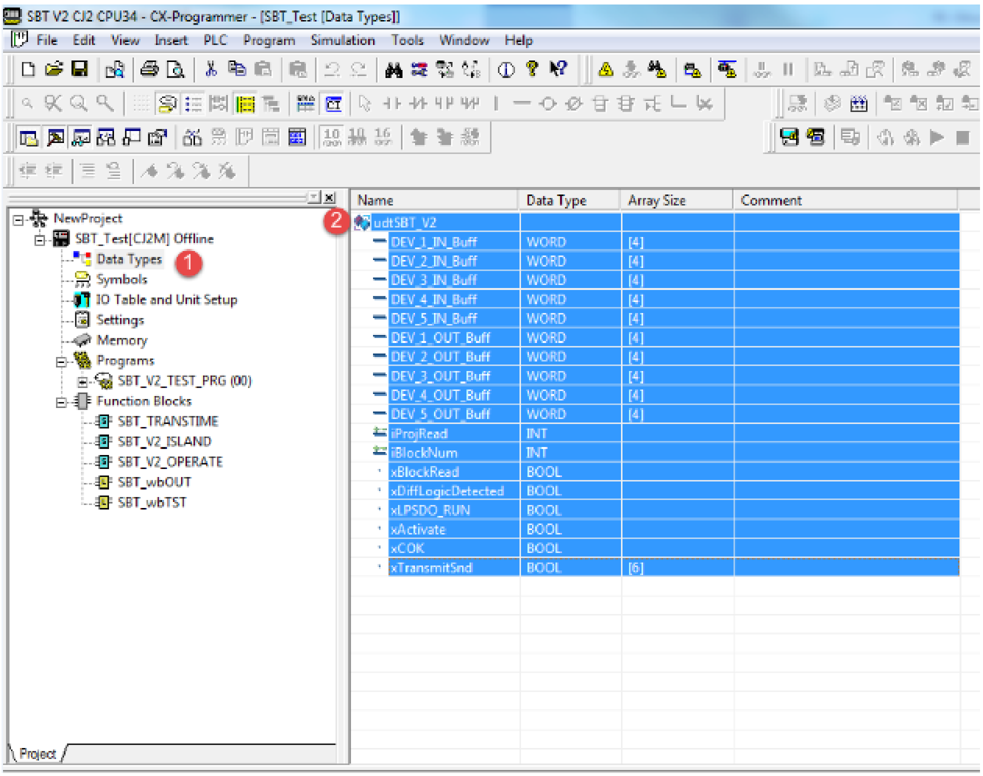
4. Save the file as **.CJF**.

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

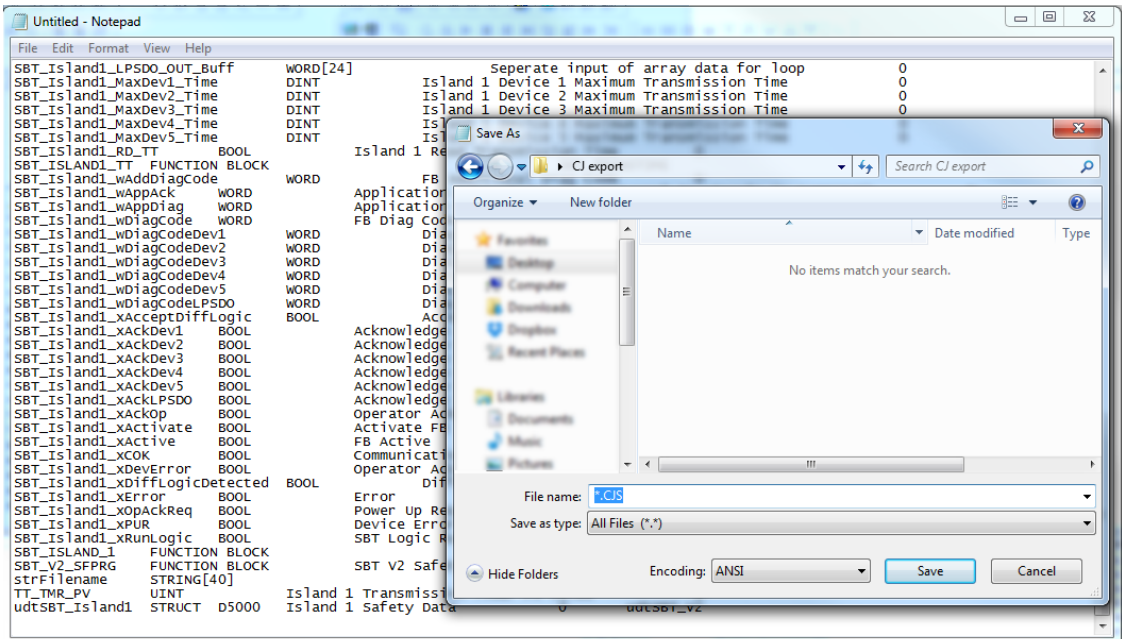
To export the **.CJS** Tag file:

1. In CX-One open the Data Types file in the project.



Name	Data Type	Array Size	Comment
udtSBT_V2			
DEV_1_IN_Buff	WORD	[4]	
DEV_2_IN_Buff	WORD	[4]	
DEV_3_IN_Buff	WORD	[4]	
DEV_4_IN_Buff	WORD	[4]	
DEV_5_IN_Buff	WORD	[4]	
DEV_1_OUT_Buff	WORD	[4]	
DEV_2_OUT_Buff	WORD	[4]	
DEV_3_OUT_Buff	WORD	[4]	
DEV_4_OUT_Buff	WORD	[4]	
DEV_5_OUT_Buff	WORD	[4]	
iProjRead	INT		
iBlockNum	INT		
xBlockRead	BOOL		
xDiffLogicDetected	BOOL		
xLPSDO_RUN	BOOL		
xActivate	BOOL		
xCOK	BOOL		
xTransmitSnd	BOOL	[6]	

2. Copy and paste all the Tags in any text editor.



```
SBT_Island1_LPSDO_OUT_Buff      WORD[24]              Seperate input of array data for loop         0
SBT_Island1_MaxDev1_Time        DINT            Island 1 Device 1 Maximum Transmission Time         0
SBT_Island1_MaxDev2_Time        DINT            Island 1 Device 2 Maximum Transmission Time         0
SBT_Island1_MaxDev3_Time        DINT            Island 1 Device 3 Maximum Transmission Time         0
SBT_Island1_MaxDev4_Time        DINT            Isl
SBT_Island1_MaxDev5_Time        DINT            Isl
SBT_Island1_RD_TT       BOOL            Island 1 Re
SBT_ISLAND1_TT  FUNCTION BLOCK
SBT_Island1_wAddDiagCode        WORD            FB
SBT_Island1_wAppAck     WORD            Applicatio
SBT_Island1_wAppDiag    WORD            Applicatio
SBT_Island1_wDiagCode   WORD            FB Diag Cod
SBT_Island1_wDiagCodeDev1       WORD            Dia
SBT_Island1_wDiagCodeDev2       WORD            Dia
SBT_Island1_wDiagCodeDev3       WORD            Dia
SBT_Island1_wDiagCodeDev4       WORD            Dia
SBT_Island1_wDiagCodeDev5       WORD            Dia
SBT_Island1_wDiagCodeLPSDO      WORD            Dia
SBT_Island1_xAcceptDiffLogic    BOOL            Acc
SBT_Island1_xAckDev1    BOOL            Acknowledge
SBT_Island1_xAckDev2    BOOL            Acknowledge
SBT_Island1_xAckDev3    BOOL            Acknowledge
SBT_Island1_xAckDev4    BOOL            Acknowledge
SBT_Island1_xAckDev5    BOOL            Acknowledge
SBT_Island1_xAckLPSDO   BOOL            Acknowledge
SBT_Island1_xAckOp      BOOL            Operator Ac
SBT_Island1_xActivate   BOOL            Activate FB
SBT_Island1_xActive     BOOL            FB Active
SBT_Island1_xCOK        BOOL            Communicati
SBT_Island1_xDevError   BOOL            Operator Ac
SBT_Island1_xDiffLogicDetected  BOOL            Dif
SBT_Island1_xError      BOOL            Error
SBT_Island1_xOpAckReq   BOOL            Power Up Re
SBT_Island1_xPUR        BOOL            Device Erro
SBT_Island1_xRunLogic   BOOL            SBT Logic R
SBT_ISLAND_1    FUNCTION BLOCK          
SBT_V2_SFPRG    FUNCTION BLOCK          SBT V2 Safe
strFilename     STRING[40]
TT_TMR_PV       UINT            Island 1 Transmissi
udtSBT_Island1  STRUCT  D5000   Island 1 Safety Data            0       udtSBT_V2
```

No items match your search.

File name: *.CJS

Save as type: All Files (*.*)

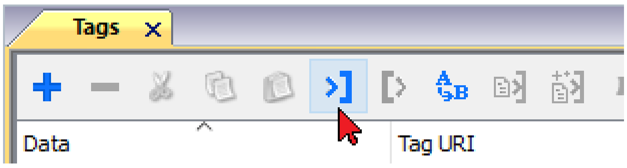
Encoding: ANSI

3. Save the file as **.CJS**.

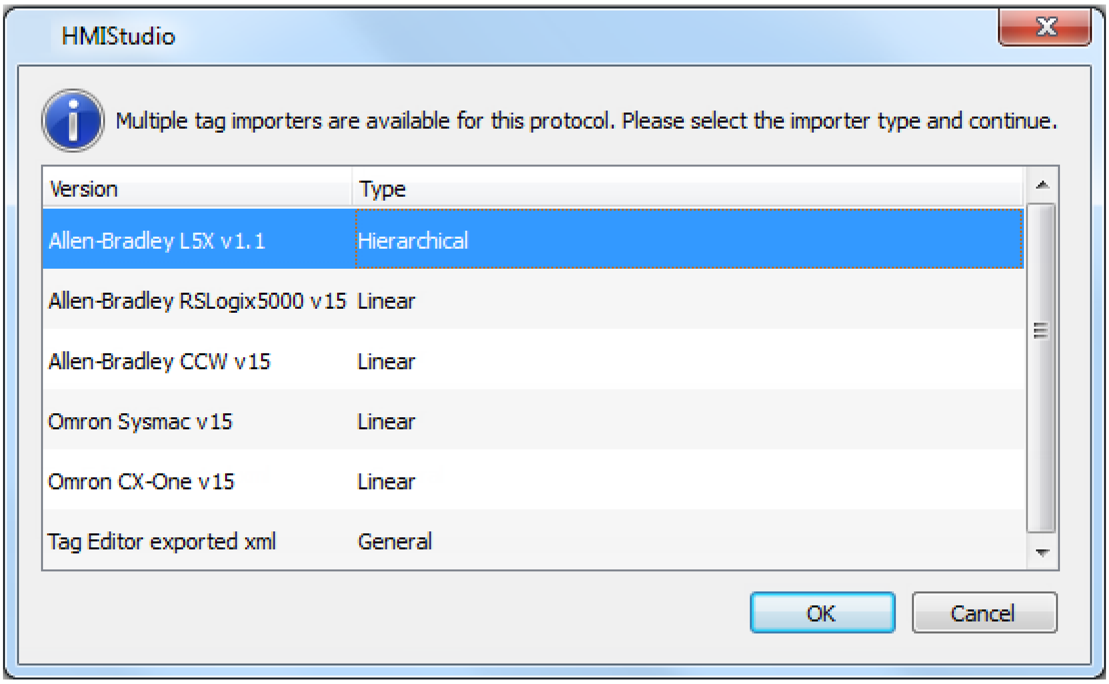
Note: Using Notepad as text editor, make sure to save the text file with **.CJS** extension by selecting "Save as type" as "All Files" although the file will be named *.cjs.txt and it will not be visible from importer.

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.



HMIStudio

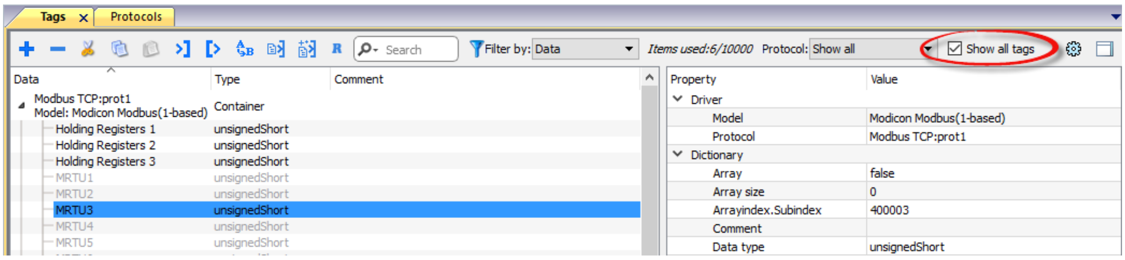
Multiple tag importers are available for this protocol. Please select the importer type and continue.

Version	Type
Allen-Bradley L5X v1.1	Hierarchical
Allen-Bradley RSLogix5000 v15	Linear
Allen-Bradley CCW v15	Linear
Omron Sysmac v15	Linear
Omron CX-One v15	Linear
Tag Editor exported xml	General

Select **Omron Sysmac** to import a **.NJF** Tags file or **Omron CX-One** to import a **.CJF** Tags file.

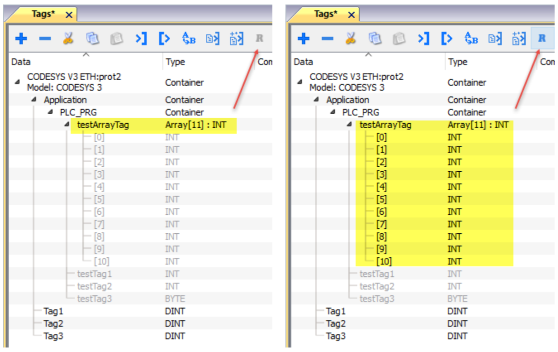
Once the importer has been selected, locate the Tags file and click **Open**. The system will ask for User Defined structures **.CJS** file. If not required, skip the dialog by clicking on Cancel button.

Tags included in the symbol file are listed in the tag dictionary. The tag dictionary is displayed at the bottom of the screen.



Data	Type	Comment
Modbus TCP:prot1 Model: Modicon Modbus(1-based)	Container	
Holding Registers 1	unsignedShort	
Holding Registers 2	unsignedShort	
Holding Registers 3	unsignedShort	
MRTU1	unsignedShort	
MRTU2	unsignedShort	
MRTU3	unsignedShort	
MRTU4	unsignedShort	
MRTU5	unsignedShort	

Property	Value
Driver	
Model	Modicon Modbus(1-based)
Protocol	Modbus TCP:prot1
Dictionary	
Array	false
Array size	0
Arrayindex.Subindex	400003
Comment	
Data type	unsignedShort

Toolbar item	Description
	Import Tag(s). Select tags to be imported and click on this icon to add tags from tag dictionary to the project
	Update Tag(s). Click on this icon to update the tags in the project, due a new dictionary import.
R	Check this box to import all sub-elements of a tag. Example of both checked and unchecked result:  Data / Type / Con CODESYS V3 ETH:prot2 Model: CODESYS 3 / Container Application / Container PLC_PRG / Container testArrayTag / Array[11] : INT [0] INT, [1] INT, [2] INT, [3] INT, [4] INT, [5] INT, [6] INT, [7] INT, [8] INT, [9] INT, [10] INT testTag1 / INT testTag2 / INT testTag3 / BYTE Tag1 / DINT Tag2 / DINT Tag3 / DINT
 Search Filter by: Tag name ▼	Searches tags in the dictionary basing on filter combo-box 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.

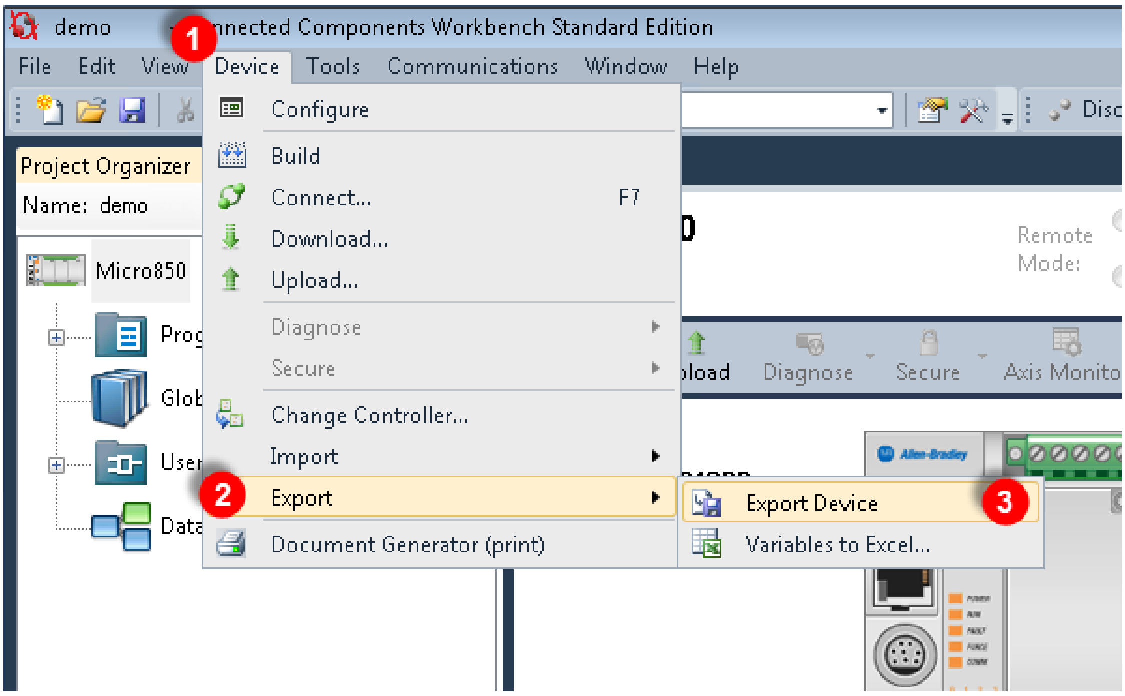
The scope of variables into a Micro800 controller can be local to a program or global:

Scope	Description
Local Variables	Program-scoped Tags. Tags are assigned to a specific program in the project and available only to that program. These Tags are **not supported** within this driver.
Global Variables	Controller-scoped Tags. Tags belong to the controller in the project and are available to any program in the project. These Tags are **supported** within this driver.

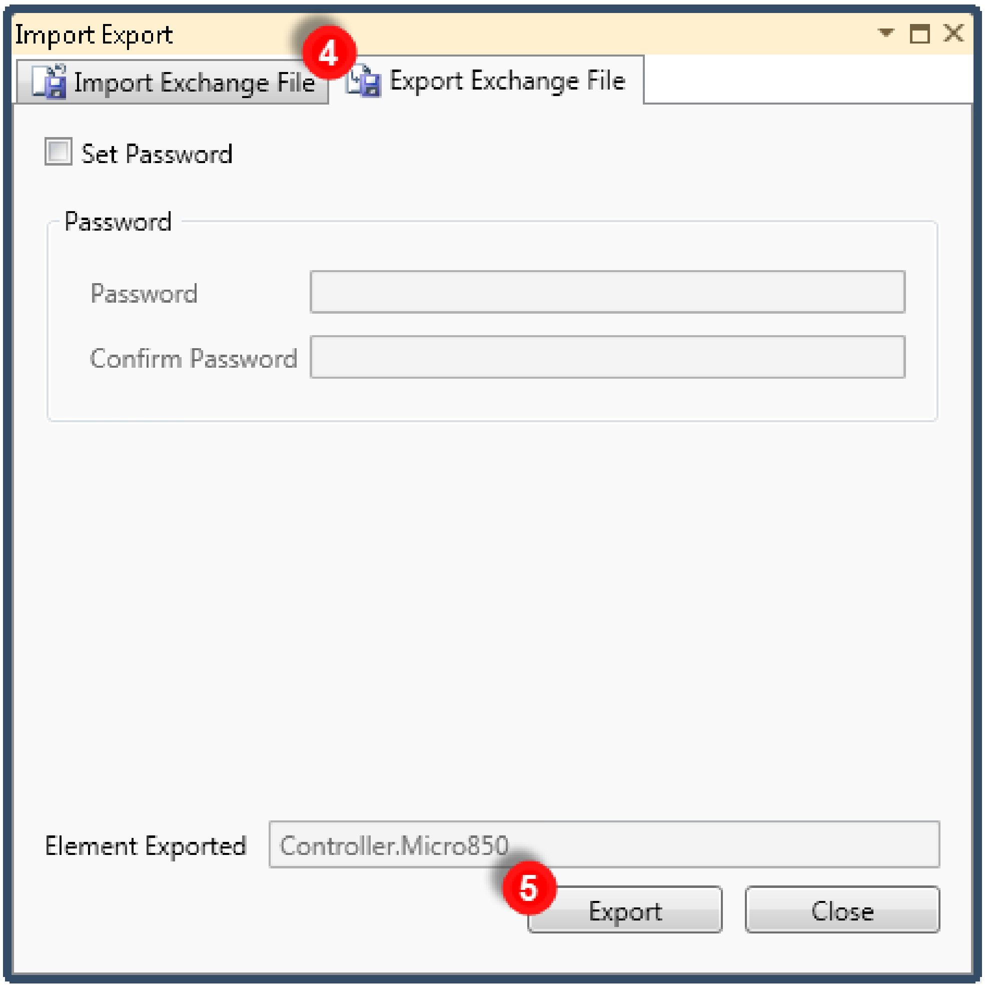
Export ISAXML file using Connected Component Workbench

To export **.ISAXML** global variables including I/O tags:

1. Select **Device** tab.
2. Expand **Export** item.
3. Select **Export Device**.



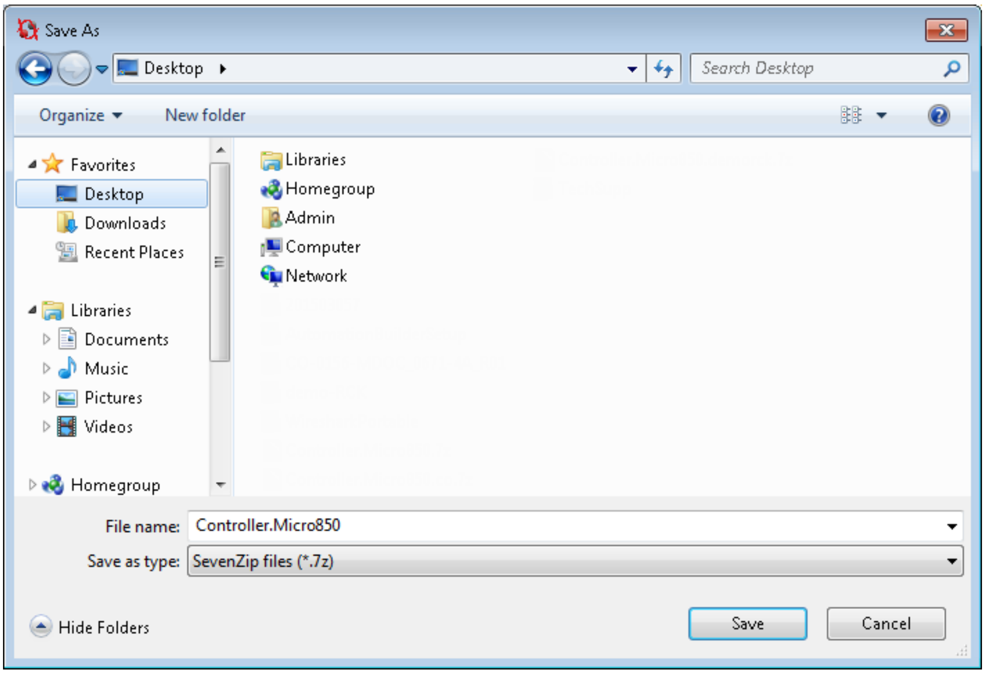
4. Click onExport **Exchange File** tab.
5. Click **Export** button.



☐ Set Password

Element Exported: Controller.Micro850

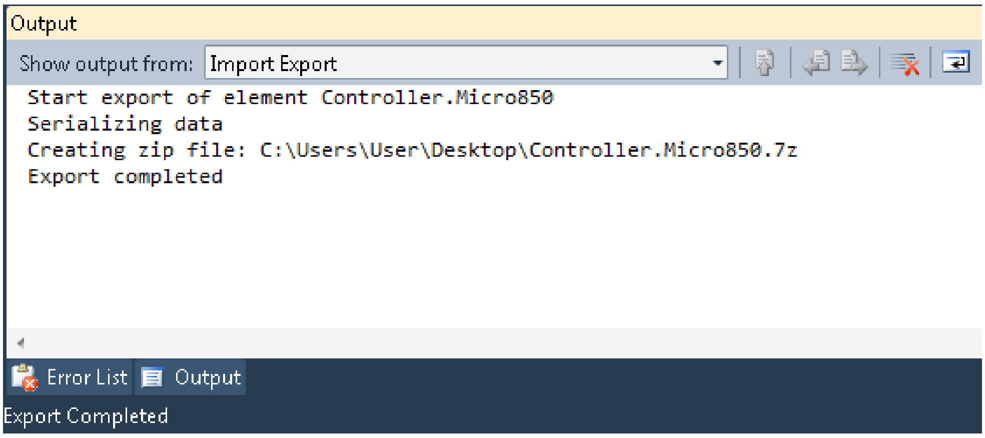
6. Choose a location where to save the export file and click **Save**.



File name: Controller.Micro850

Save as type: SevenZip files (*.7z)

7. When the export is completed successfully the output information is displayed:



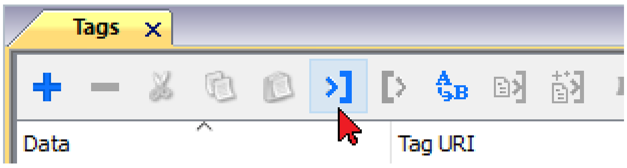
```
Start export of element Controller.Micro850
Serializing data
Creating zip file: C:\Users\User\Desktop\Controller.Micro850.7z
Export completed
```

Export Completed

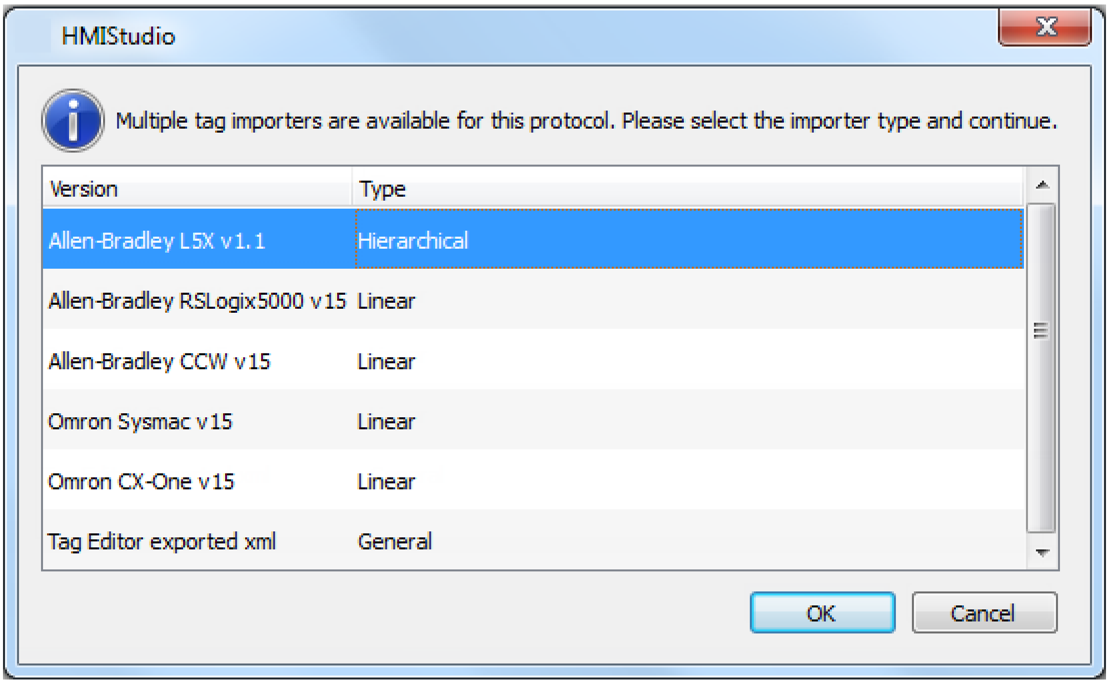
Note: CCW export file is a 7-zip compressed archive. Use a suitable zip utility to extract archive content into a local folder.

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.



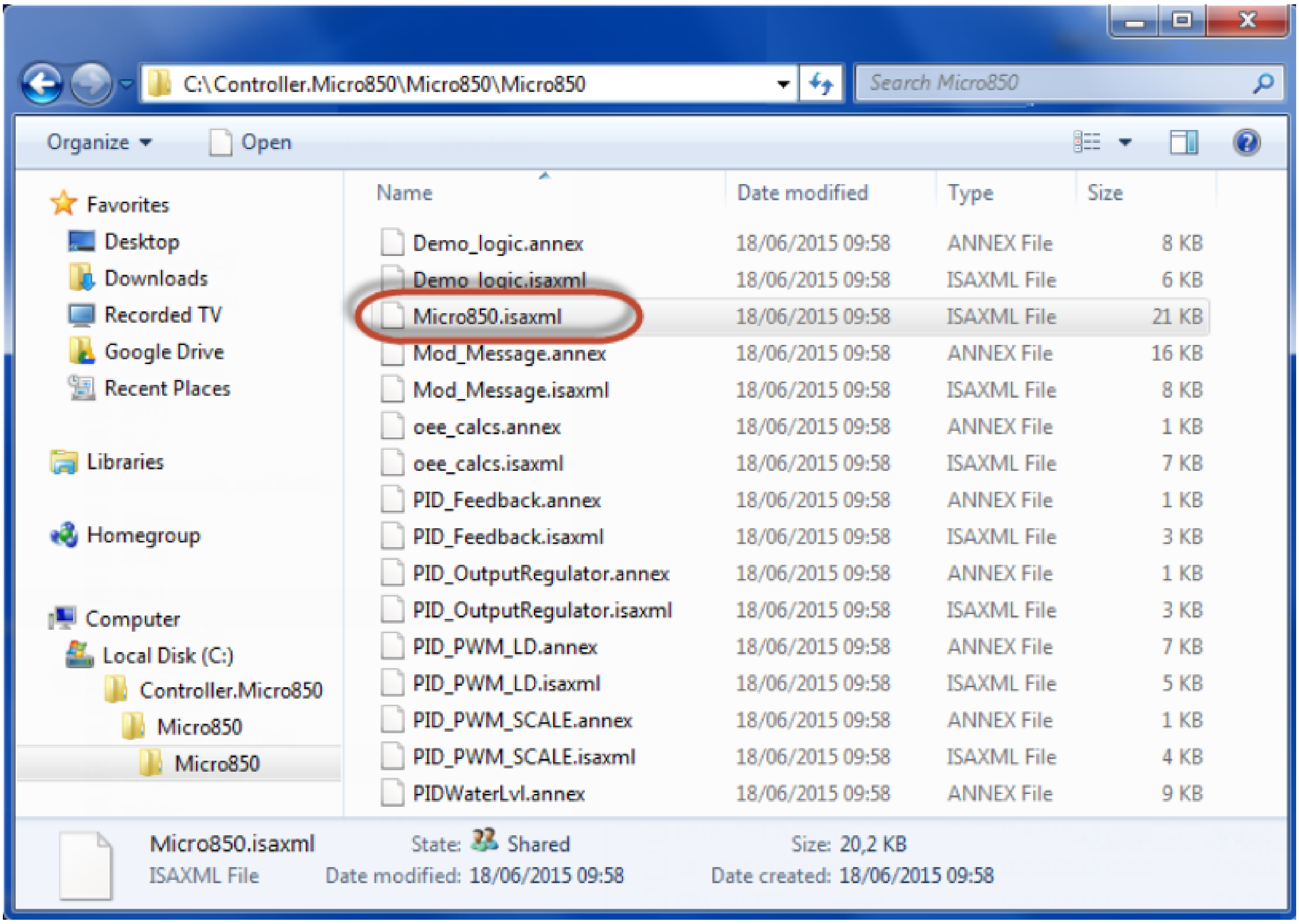
Multiple tag importers are available for this protocol. Please select the importer type and continue.

Version	Type
Allen-Bradley L5X v1.1	Hierarchical
Allen-Bradley RSLogix5000 v15	Linear
Allen-Bradley CCW v15	Linear
Omron Sysmac v15	Linear
Omron CX-One v15	Linear
Tag Editor exported xml	General

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:



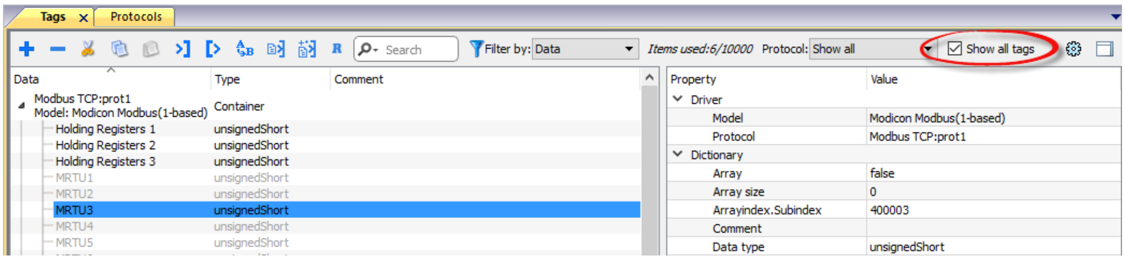
C:\Controller.Micro850\Micro850\Micro850

Name	Date modified	Type	Size
Demo_logic.annex	18/06/2015 09:58	ANNEX File	8 KB
Demo_logic.isaxml	18/06/2015 09:58	ISAXML File	6 KB
Micro850.isaxml	18/06/2015 09:58	ISAXML File	21 KB
Mod_Message.annex	18/06/2015 09:58	ANNEX File	16 KB
Mod_Message.isaxml	18/06/2015 09:58	ISAXML File	8 KB
oee_calcs.annex	18/06/2015 09:58	ANNEX File	1 KB
oee_calcs.isaxml	18/06/2015 09:58	ISAXML File	7 KB
PID_Feedback.annex	18/06/2015 09:58	ANNEX File	1 KB
PID_Feedback.isaxml	18/06/2015 09:58	ISAXML File	3 KB
PID_OutputRegulator.annex	18/06/2015 09:58	ANNEX File	1 KB
PID_OutputRegulator.isaxml	18/06/2015 09:58	ISAXML File	3 KB
PID_PWM_LD.annex	18/06/2015 09:58	ANNEX File	7 KB
PID_PWM_LD.isaxml	18/06/2015 09:58	ISAXML File	5 KB
PID_PWM_SCALE.annex	18/06/2015 09:58	ANNEX File	1 KB
PID_PWM_SCALE.isaxml	18/06/2015 09:58	ISAXML File	4 KB
PIDWaterLvl.annex	18/06/2015 09:58	ANNEX File	9 KB

Micro850.isaxml State: Shared Size: 20,2 KB
ISAXML File Date modified: 18/06/2015 09:58 Date created: 18/06/2015 09:58

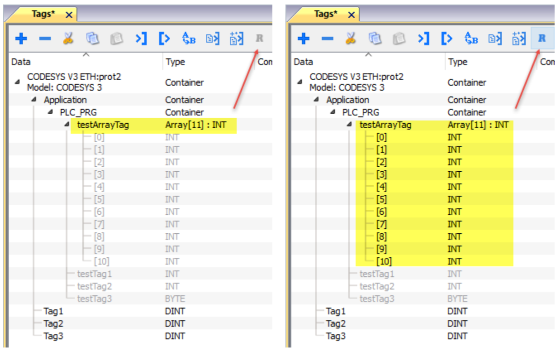
Once the importer has been selected, locate the symbol file and click **Open**.

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.



Data	Type	Comment
Modbus TCP:prot1 Model: Modicon Modbus(1-based)	Container	
Holding Registers 1	unsignedShort	
Holding Registers 2	unsignedShort	
Holding Registers 3	unsignedShort	
MRTU1	unsignedShort	
MRTU2	unsignedShort	
MRTU3	unsignedShort	
MRTU4	unsignedShort	
MRTU5	unsignedShort	

Property	Value
Driver	
Model	Modicon Modbus(1-based)
Protocol	Modbus TCP:prot1
Dictionary	
Array	false
Array size	0
Arrayindex.Subindex	400003
Comment	
Data type	unsignedShort

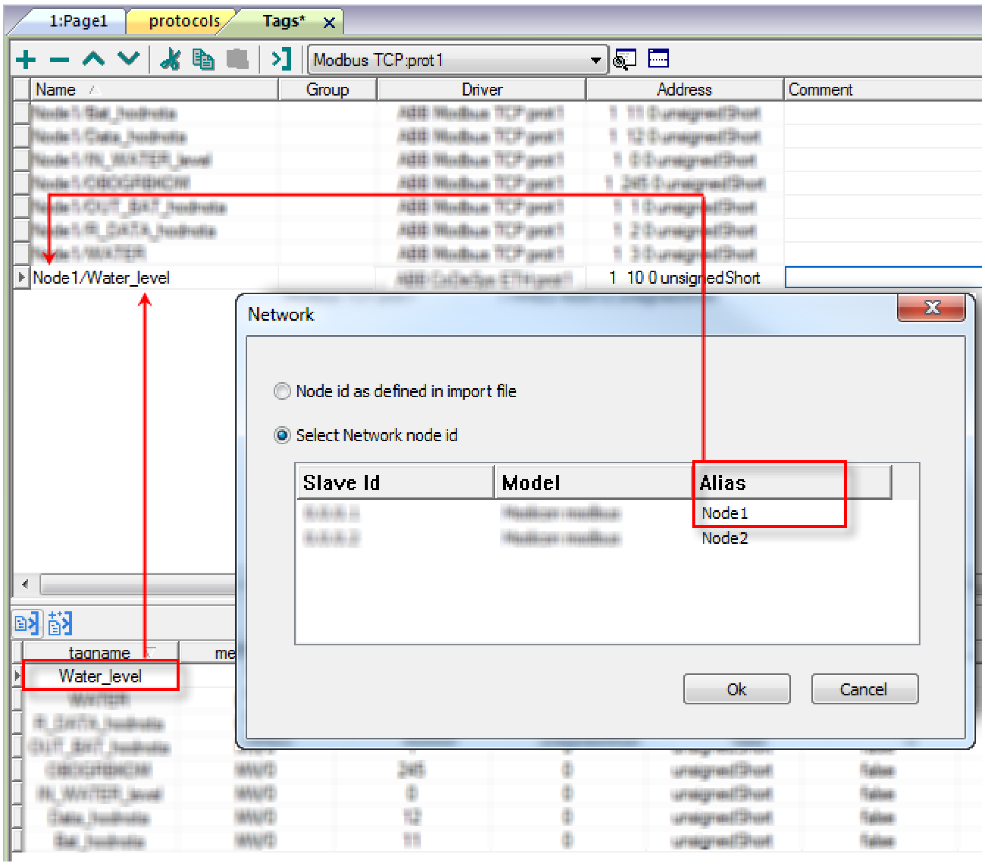
Toolbar item	Description
	Import Tag(s). Select tags to be imported and click on this icon to add tags from tag dictionary to the project
	Update Tag(s). Click on this icon to update the tags in the project, due a new dictionary import.
R	Check this box to import all sub-elements of a tag. Example of both checked and unchecked result:  Tags* Data · Type · Con CODESYS V3 ETH:prot2 Model: CODESYS 3 · Container Application · Container PLC_PRG · Container testArrayTag · Array[11] : INT [0] · INT [1] · INT [2] · INT [3] · INT [4] · INT [5] · INT [6] · INT [7] · INT [8] · INT [9] · INT [10] · INT testTag1 · INT testTag2 · INT testTag3 · BYTE Tag1 · DINT Tag2 · DINT Tag3 · DINT
 	Searches tags in the dictionary basing on filter combo-box 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.



Node1/Water_level 1 10 0 unsignedShort

Network

○ Node id as defined in import file

● Select Network node id

Slave Id	Model	Alias
[illegible]	[illegible]	Node1
[illegible]	[illegible]	Node2

Ok Cancel

Water_level

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.

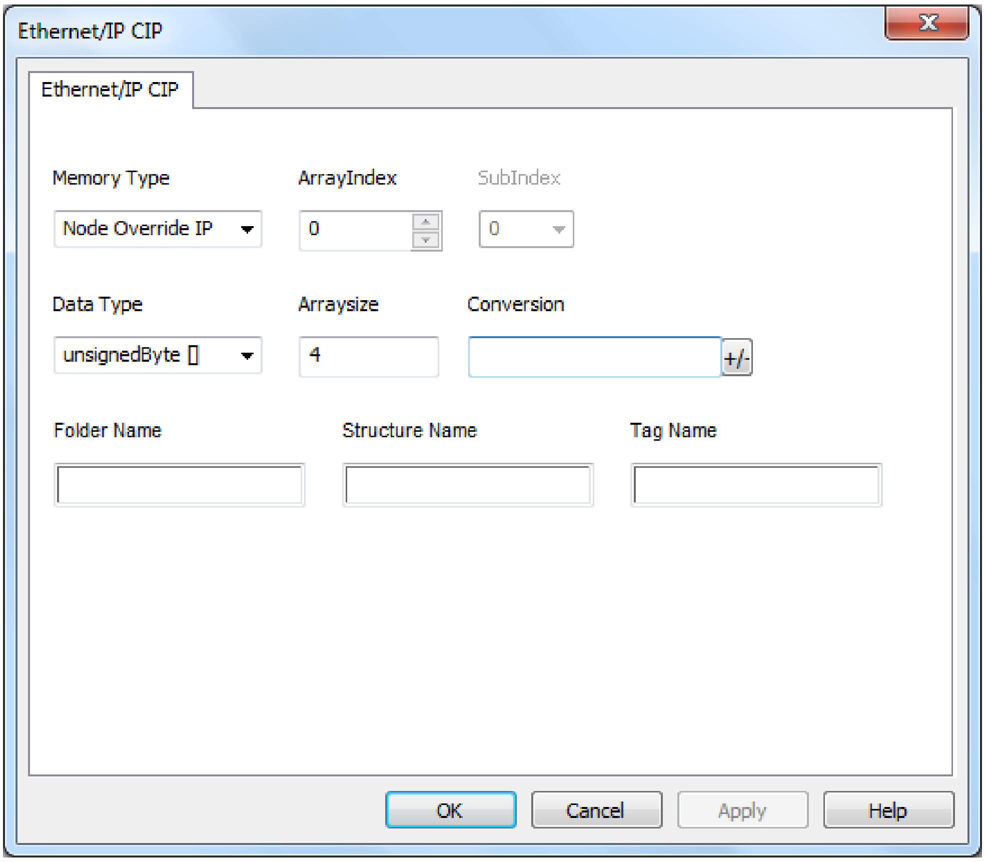
Node Override IP	PLC operation
0.0.0.0	Communication with the controller is stopped, no request frames are generated anymore.
Different from 0.0.0.0	It is interpreted as node IP override and the target IP address is replaced runtime with the new value.

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.



Memory Type: Node Override IP

ArrayIndex: 0

SubIndex: 0

Data Type: unsignedByte []

Arraysize: 4

Conversion:

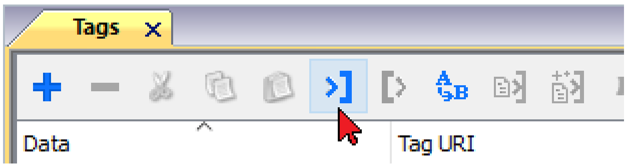
Folder Name:

Structure Name:

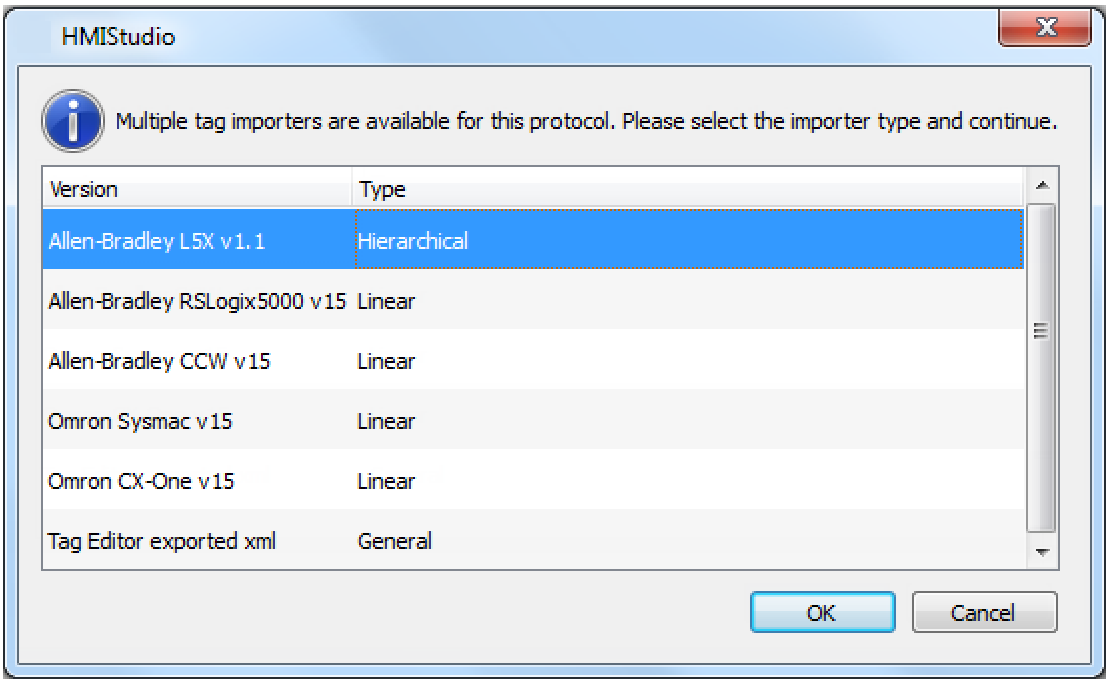
Tag Name:

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.

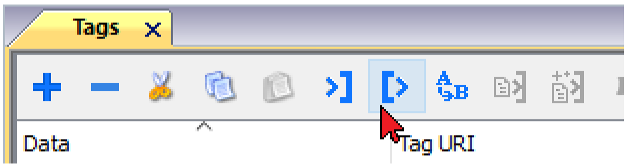


HMIStudio

Multiple tag importers are available for this protocol. Please select the importer type and continue.

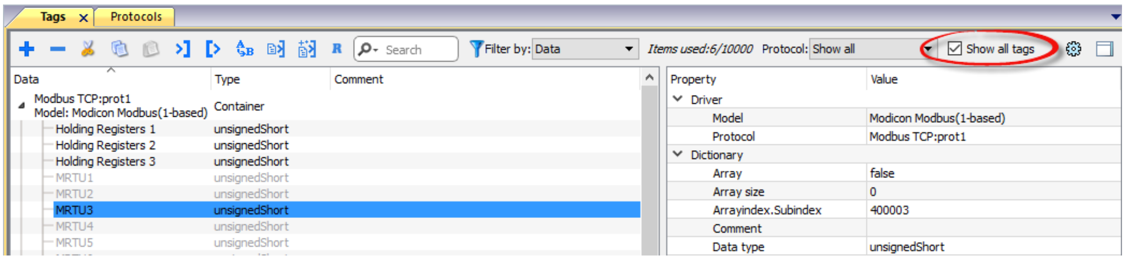
Version	Type
Allen-Bradley L5X v1.1	Hierarchical
Allen-Bradley RSLogix5000 v15	Linear
Allen-Bradley CCW v15	Linear
Omron Sysmac v15	Linear
Omron CX-One v15	Linear
Tag Editor exported xml	General

Importer	Description
Allen-Bradley L5X v1.1 Hierarchical	Requires a .L5X file. Check **Controller Model Logix 5000** for more details. All variables will be displayed according to RSLogix5000 Hierarchical view.
Allen-Bradley RSLogix5000 v15 Linear	Requires a **.CSV**and **.L5X** (optional) files. Check **Controller Model Logix 5000** for more details. All variables will be displayed at the same level.
Allen-Bradley CCW v15 Linear	Requires a **.ISAXML** file. Check **Controller Model Micro800** for more details. All variables will be displayed at the same level.
Omron Sysmac v15 Linear	Requires a **.NJF** file. Check **Controller Model Omron Sysmac** for more details. All variables will be displayed at the same level.

Importer	Description
Omron CX-One v15 Linear	Requires a **.CJF**and **.CJS** (optional) files. Check **Controller Model Omron Sysmac** for more details. 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.



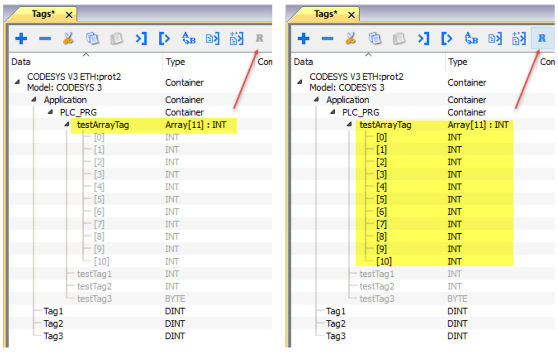
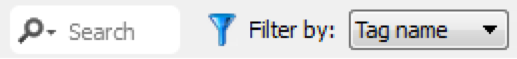
Tags | Protocols

Search | Filter by: Data | *Items used:6/10000* | Protocol: Show all | ☑ Show all tags

Data	Type	Comment
Modbus TCP:prot1 Model: Modicon Modbus(1-based)	Container	
Holding Registers 1	unsignedShort	
Holding Registers 2	unsignedShort	
Holding Registers 3	unsignedShort	
MRTU1	unsignedShort	
MRTU2	unsignedShort	
MRTU3	unsignedShort	
MRTU4	unsignedShort	
MRTU5	unsignedShort	

Property	Value
Driver	
Model	Modicon Modbus(1-based)
Protocol	Modbus TCP:prot1
Dictionary	
Array	false
Array size	0
Arrayindex.Subindex	400003
Comment	
Data type	unsignedShort

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

Toolbar item	Description
	 Tags* × Data | Type | Con CODESYS V3 ETH:prot2 Model: CODESYS 3 | Container Application | Container PLC_PRG | Container testArrayTag | Array[11] : INT [0] | INT [1] | INT [2] | INT [3] | INT [4] | INT [5] | INT [6] | INT [7] | INT [8] | INT [9] | INT [10] | INT testTag1 | INT testTag2 | INT testTag3 | BYTE Tag1 | DINT Tag2 | DINT Tag3 | DINT
 Search Filter by: Tag name	Searches tags in the dictionary basing on filter combo-box item selected.

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
NAK	The controller replies with a not acknowledge.	-
Timeout	A request is not replied within the specified timeout period.	Check if the controller is connected and properly configured to get network access.
Invalid response	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.
General Error	Unidentifiable error. Should never be reported.	Contact technical support.

J1939

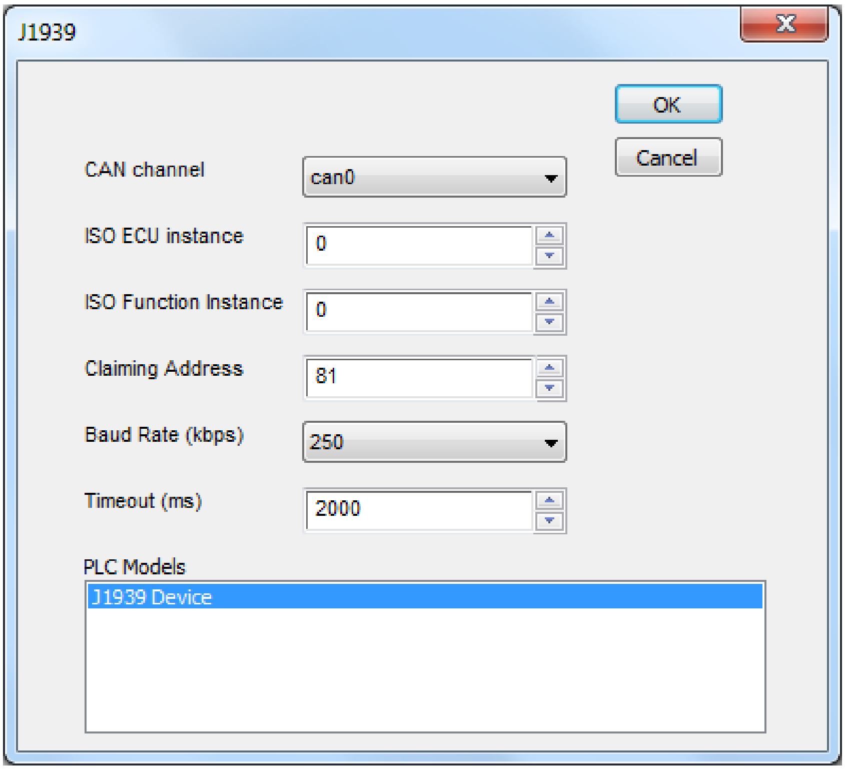
Use this communication driver to connect HMI devices to CAN networks including devices communicating with SAE J1939.

Please note that changes in the communication protocol specifications or J1939 hardware may have occurred since this documentation was created. Some changes may eventually affect the functionality of this communication driver. Always test and verify the functionality of your application. To fully support changes in J1939 hardware and communication protocols, communication drivers are continuously updated. Always ensure that the latest version of communication driver is used in your application.

Protocol Editor Settings

Select Add [+] in Protocol Editor and select J1939.

The driver configuration dialog is shown in figure.



J1939

CAN channel: can0

ISO ECU instance: 0

ISO Function Instance: 0

Claiming Address: 81

Baud Rate (kbps): 250

Timeout (ms): 2000

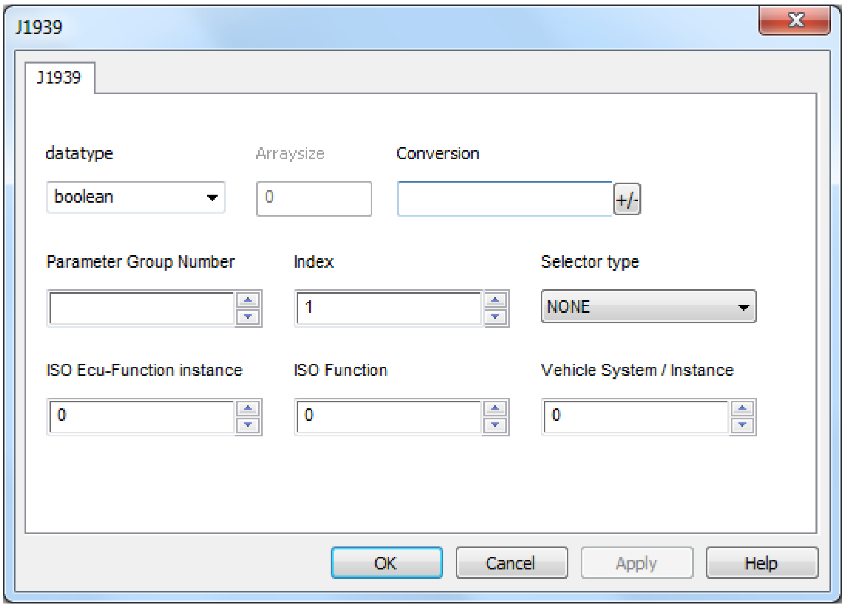
PLC Models: J1939 Device

Element	Description
CAN Channel	Configure the CAN Channel. CAN interface is available only with a proper option module. UN31 platforms allow only one module, select Can0. UN30 platforms allow up to two modules, select Can0 or Can1.
ISO ECU	Identifier of the equipment in the J1939 network (in case several HMI are coexisting in the

Element	Description
Instance	network)
ISO Function Instance	Identifier of the function in the network (in case more than one device is providing the same functionality)
Claiming Address	Default value of the address of the equipment used as starting value for the Address Claim algorithm
Baud Rate (kbps)	Baud rate of the CAN bus (typical is 250)
Timeout (ms)	Timeout for the validity of received values. After the time indicated since last reception any value is declared “old” and its quality changed to “bad”. The value 0 disables the timeout check

Tag Editor Settings

In Tag Editor select the protocol “J1939” from the list of defined protocols and add a tag using [+] button. Tag settings can be defined using the following dialog:



J1939

datatype: boolean

Arraysize: 0

Conversion:

Parameter Group Number:

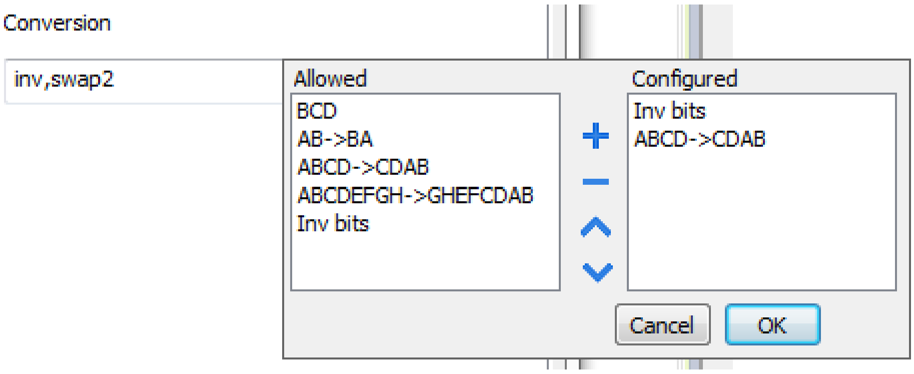
Index: 1

Selector type: NONE

ISO Ecu-Function instance: 0

ISO Function: 0

Vehicle System / Instance: 0

<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td><b>Data Type</b></td><td>
<table>
<tr><th>Data Type</th><th>Memory Space</th><th>Limits</th></tr>
<tr><td>boolean</td><td>1 bit data</td><td>0 ... 1</td></tr>
<tr><td>byte</td><td>8-bit data</td><td>-128 ... 127</td></tr>
<tr><td>short</td><td>16-bit data</td><td>-32768 ... 32767</td></tr>
<tr><td>int</td><td>32-bit data</td><td>-2.1e9 ... 2.1e9</td></tr>
<tr><td>unsignedByte</td><td>8-bit data</td><td>0 ... 255</td></tr>
<tr><td>unsignedShort</td><td>16-bit data</td><td>0 ... 65535</td></tr>
<tr><td>unsignedInt</td><td>32-bit data</td><td>0 ... 4.2e9</td></tr>
<tr><td>float</td><td>IEEE single-precision
32-bit floating point type</td><td>1.17e-38 ... 3.40e38</td></tr>
<tr><td>string</td><td colspan="2">Array of elements containing character code defined by selected encoding</td></tr>
</table>
Note: to define arrays, select one of Data Type format followed by square brackets like “byte[]”, “short[]”…
</td></tr>
<tr><td><b>Arraysize</b></td><td>
<ul>
<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>
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.
</td></tr>
<tr><td><b>Conversion</b></td><td>
Conversion to be applied to the tag.

Conversion: inv,swap2

Depending on data type selected, the list <b>Allowed</b> shows one or more conversion types.
</td></tr>
</table>

<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>Value</th><th>Description</th></tr>
<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 -> 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 -> 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 -> 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 -> OPM...DAB</b></td><td><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)</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>
</table>
</td></tr>
</table>

Element	Description
	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.
Parameter Group Number	Parameter Group Number value
Index	Index value
Selector Type	When adding tags it can be necessary to duplicate them to read data coming from several devices generating same physical quantity. In this case the Address of the tag must be edited. The Tag Editor dialog is shown in figure: In case of duplication of the tag, the selection of incoming data can be done using one of following methods: **NONE** Selector Type not selected **INSTANCE** uses a defined bitfield value in data of PGN to distinguish between the possible sources. The value of received bitfield is compared with parameter “Vehicle System / Instance” for matching **DEVICE** uses the source address to find out the device sending the PGN based on Address Claim algorithm. The devices are selected based on parameter “ISO function” **ADDRESS** uses directly the source address as it is to select the source. The received source address is compared with parameter “ISO Ecu – Function Instance”
ISO Ecu-Function Instance	Instance of ISO Ecu-Function checked with Selector Type “DEVICE”
ISO Function	ISO Function parameter
Vehicle System/Instance	Vehicle System / Instance parameter used with Selector Type “INSTANCE”

J1939 PGN Definition File

J1939 can connect hundreds of different devices offering access to thousands of different physical values. The standard defines several hundred PGNs for various applications. However, many devices use manufacturer-specific PGN definitions.

In order to manage this complex application scenario, the J1939 driver loads the PGN definition table at startup from a configuration file. The file with the PGN definition table is “J1939_pgnTable.csv” located in the folder “*target\protocols\*”; it is loaded automatically from disk when downloading the project.

The file containing the PGN defined by the standard protocol specification is placed in the proper folder when the driver is installed. It can be edited adding or removing PGN definitions. The user must respect the following rules:

- the file contains most of the PGN defined by the standard. Custom PGN and SPN can be added assigning free indexes.
- description of a PGN is composed by a PGN declaration line followed by a list of Field description lines

PGN declaration line

```
PGN: Name, PGN number, DefaultPriority, DefaultRate, InstanceIndex, Direction [, PGN request rate]
```

Name	Name of the PGN
PGN number	Number code of PGN
DefaultPriority	Transmission priority (output PGN)
DefaultRate	Transmission rate (output PGN)
Instance Index	Index of instance (output PGN)
Direction	INPUT/OUTPUT
PGN request rate	Optional parameter. Time in milliseconds. If PGN not received in the meanwhile, it is requested

Example of PGN declaration:

PGN: Torque/Speed Control 1, 0, 3, 100, 0, INPUT

// Torque/Speed Control 1 id PGN nr.0, its default priority is 3 and default transmission rate is 100 ms. Instance Index is 0 and direction is INPUT

Field declaration line

```
FieldIndex, FieldName, FieldPosition, FieldBitSize, SPN Conversion, AccessType, FieldDataType
```

FieldIndex	Index of field in the PGN
FieldName	Name of the field
FieldPosition	N (1 to 8) byte position N.M (1.1 to 8.8) bit position N-M (N from 1 to 7, M from 2 to 8) byte range
FieldBitSize	1-64 number of bits of the field
SPN Conversion	SPN conversion is indicated by "SPN"index es. SPN79 SPN0 indicates a raw copy of data
AccessType	Defines usage of field in combination with PGN direction.

If PGN direction is declared as OUTPUT, the fields can be only used for write operations.

If PGN direction is declared as INPUT the fields can always be read. In case they are written the behavior is described below.

PGN Direction	Access Type	Behavior
OUTPUT	WRITE	the PGN is sent immediately with current value of the fields
	READ_ ONLY REPLY	the PGN is sent as soon as all the fields are written with a fresh value
INPUT	READ_ ONLY	Error
	REPLY	the PGN is sent only if it was received almost once, with update value of the written field
	WRITE	the PGN is sent immediately with current value of the fields

FieldDataType Boolean

boolean-nn

byte

unsignedByte

short

unsignedShort

int

unsignedInt

float

double

string-nn

Example of Field declaration:

1, Engine Override Control Mode, 1.1, 2, SPN0, READ_ONLY, unsignedByte

SPN declaration line

```
SPN: index, constK, constL, type [,bigEndian]
```

index index of SPN

constK SPN conversion parameters

constL the conversion applied when reading is:

var(type) = raw value * constK + constL

the conversion applied when writing is:

raw value = (var(type) - constL) / constK

type bits

char

uchar

short

ushort

int

uint

float

double

longlong

ulonglong

float80

bigEndian Optional parameter. Defines if endianity conversion is needed on raw data before applying the SPN conversion.

0 default endianity, do not change

1 apply endianity transformation

Example of SPN declaration:

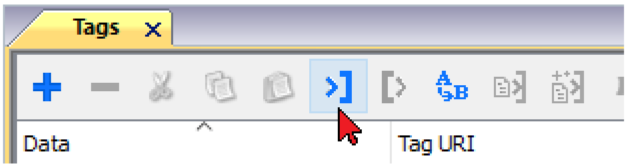
SPN:, 79, 0.03125, -273, short, 1

Tag Import

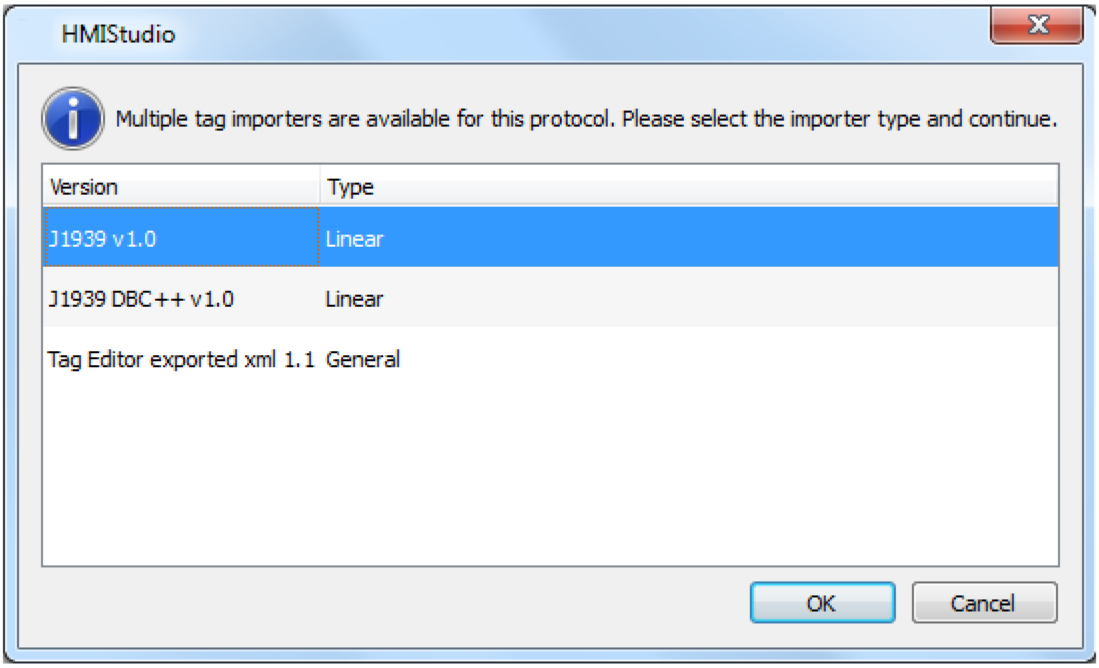
The J1939 driver can import tag information from any CSV file, following same rules of PGN definition file and maintain several dictionaries for different scenarios.

The user can also import the whole “J1939_pgnTable.csv” and use only one large dictionary.

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



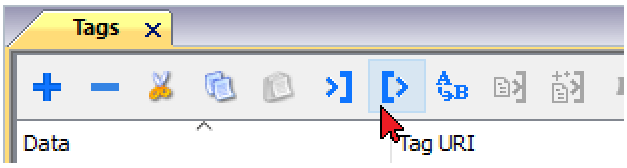
The following dialog shows which importer type can be selected



HMIStudio

Multiple tag importers are available for this protocol. Please select the importer type and continue.

Version	Type
J1939 v1.0	Linear
J1939 DBC++ v1.0	Linear
Tag Editor exported xml 1.1	General

Type	Description
J1939 v1.0 Linear	Requires a **.csv** file. All variables will be displayed at the same level.
J1939 DBC+ v1.0 Linear	Requires a .dbc file generated by Vector CANdb++ Editor All the frames will be generated with type = Rx, so frames created for transmission must be reedit after importation
Tag Editor exported xml	Select this importer to read a generic XML file exported from Tag Editor by appropriate button. 

The tags resulting from the import process may be used as they are if there is only one source for such value in the network. When several sources are supplying the same value the associated tags must be duplicated and named using one of the addressing methods shown in the Tag Editor chapter.

Communication Diagnostic

The error types supported for this communication driver are:

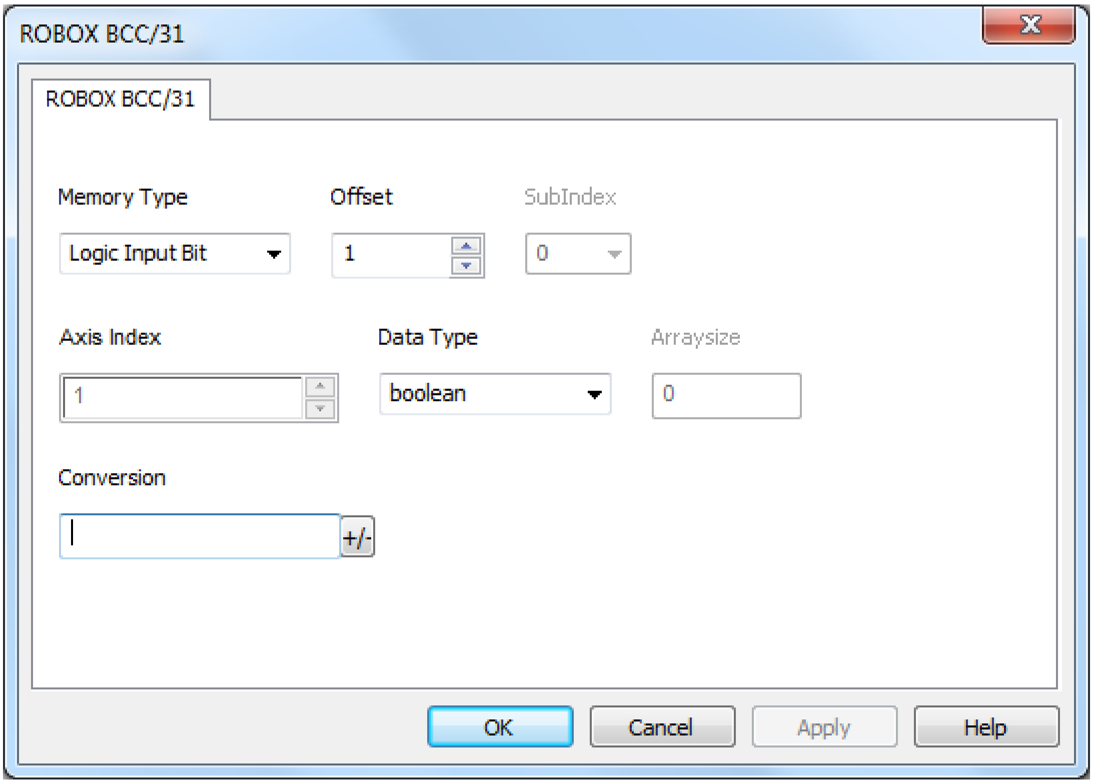
Error Class	Error	Notes
Configuration Errors	invalid CAN channel	
	cannot read MACID	
	Unable to access the PGN Table	
	Unable to get the PGN file path	
	SPN conversion not supported	
	Sending PGN with dynamic field length not supported	
	Preparing PGN field for sending failed	
	Writing a read-only tag	
	The output PGN can't be read	
	invalid offset in PGN	
	Not byte boundary on dynamic field	
	Something wrong with the PGN data block size	
	Too many bits to use	
	Not byte boundary on dynamic field	
	SPN conversion not supported	
Runtime Errors	Communication Failure > Can't send the APL PGN message	
	Not Connected > The PGN for the command reply has not been received yet	
	Not Connected > PGN block not registered	
	Not Connected > the value never received	
	Timeout Error > timeout on the value refresh	

Error Class	Error	Notes
Tag Definition Errors	there must be 7 tag specification fields	
	PGN field missing	
	SPN definition not found in the table	
	index field missing	
	ecuFunctionInstance field missing	
	function field missing	
	classOrInstance field missing	
	icomType field missing	
	Can't access protocol common parameters	
	Can't access protocol node parameters	
	Can't access model	
	Can't access memory type	
	strError.c_str()	
	not allowed icom type	
	invalid natural data type for this memory type	
	invalid field 'selector type'	
	PGN definition not found in the table	
	The field not found in this PGN	

Tag Editor Settings

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

1. To add a tag, click **+**: a new line is added.
2. Select **ROBOX BCC/31** from the **Driver** list: tag definition dialog is displayed.



ROBOX BCC/31

Memory Type: Logic Input Bit

Offset: 1

SubIndex: 0

Axis Index: 1

Data Type: boolean

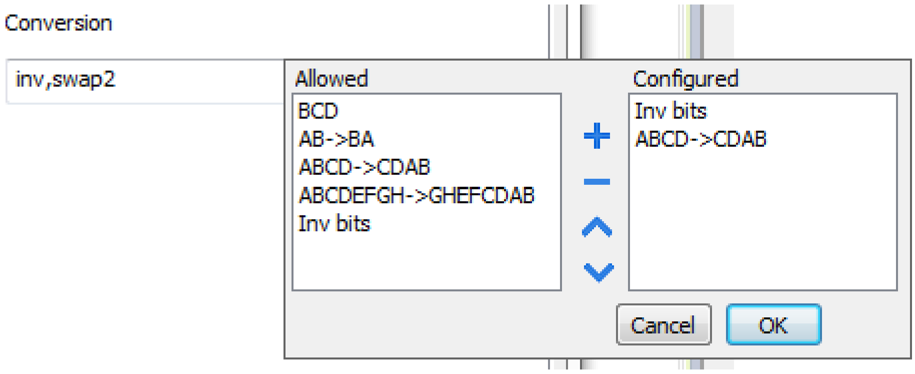
Arraysize: 0

Conversion:

<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td><b>Memory Type</b></td><td>Resource where tag is located on PLC.
Available resources are:
• Logic Input Bit
• Logic Input Word
• Logic Output Bit
• Logic Output Word
• Phis Input Bit
• Phys Input Word
• Phys Output Bit
• Phys Output Word
• Non Volatile I32
• Non Volatile Double
• Non Volatile string
• Volatile I32
• Volatile Double
• Volatile string
• Parameter I32
• Parameter Double
• Axis Parameter I32
• Axis Parameter Double
• Alarm Mask
• Alarm Code
• Alarm string</td></tr>
<tr><td><b>Offset</b></td><td>Offset address where tag is located.
Offset addresses are six digits composed by one digit data type prefix + five digits resource address.</td></tr>
<tr><td><b>SubIndex</b></td><td>This allows resource offset selection within the selected memory type.</td></tr>
<tr><td><b>Axis Index</b></td><td>Allows to select Axis index. Available only for Axis memory types.</td></tr>
<tr><td><b>Data Type</b></td><td><table>
<tr><th>Data Type</th><th>Memory Space</th><th>Limits</th></tr>
<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>
</table></td></tr>
</table>

<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>Data Type</th><th>Memory Space</th><th>Limits</th></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>
</table>

Note: to define arrays. select one of Data Type format followed by square brackets like “byte[]”, “short[]”…
</td></tr>
<tr><td><b>Arraysize</b></td><td>
<ul>
<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>
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.
</td></tr>
<tr><td><b>Conversion</b></td><td>
Conversion to be applied to the tag.



Conversion: inv,swap2

Allowed: BCD, AB-&gt;BA, ABCD-&gt;CDAB, ABCDEFGH-&gt;GHEFCDAB, Inv bits

Configured: Inv bits, ABCD-&gt;CDAB

Depending on data type selected, the list <b>Allowed</b> shows one or more conversion types.
</td></tr>
</table>

<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>Value</th><th>Description</th></tr>
<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 -> 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 -> 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 -> 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 -> OPM...DAB</b></td><td><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)</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>
</table>
Select conversion and click +. The selected item will be added to list <b>Configured</b>.
</td></tr>
</table>

Element	Description
	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.

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.

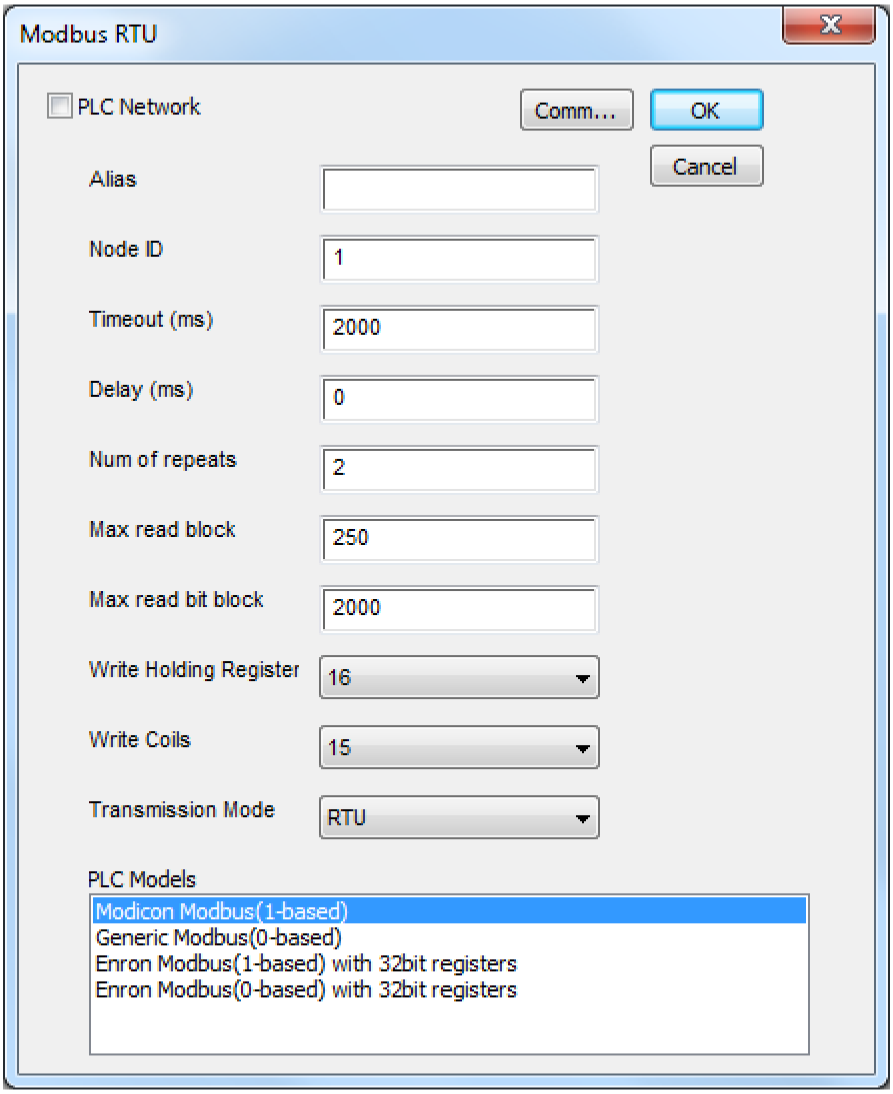
Protocol Editor Settings

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

Alias:

Node ID: 1

Timeout (ms): 2000

Delay (ms): 0

Num of repeats: 2

Max read block: 250

Max read bit block: 2000

Write Holding Register: 16

Write Coils: 15

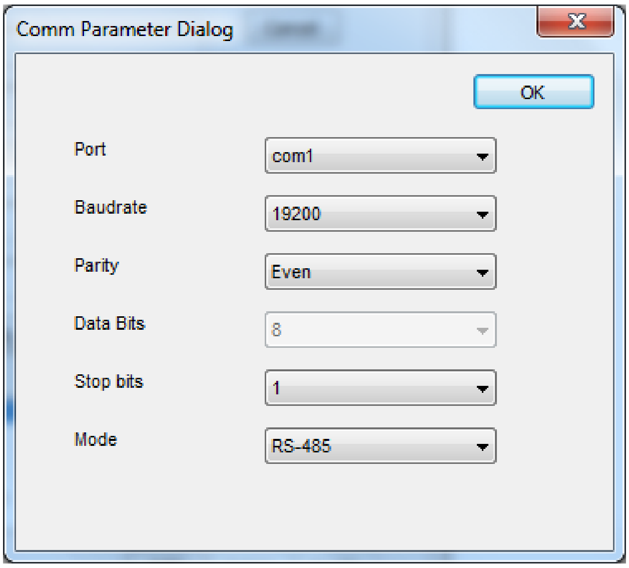
Transmission Mode: RTU

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
Alias	Name identifying nodes in network configurations. The name will be added as a prefix to each tag name imported for each network node.
Node ID	Modbus node of the slave device.
Timeout (ms)	Time delay in milliseconds between two retries in case of missing response from the server device.
Delay (ms)	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.
Num of repeats	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.

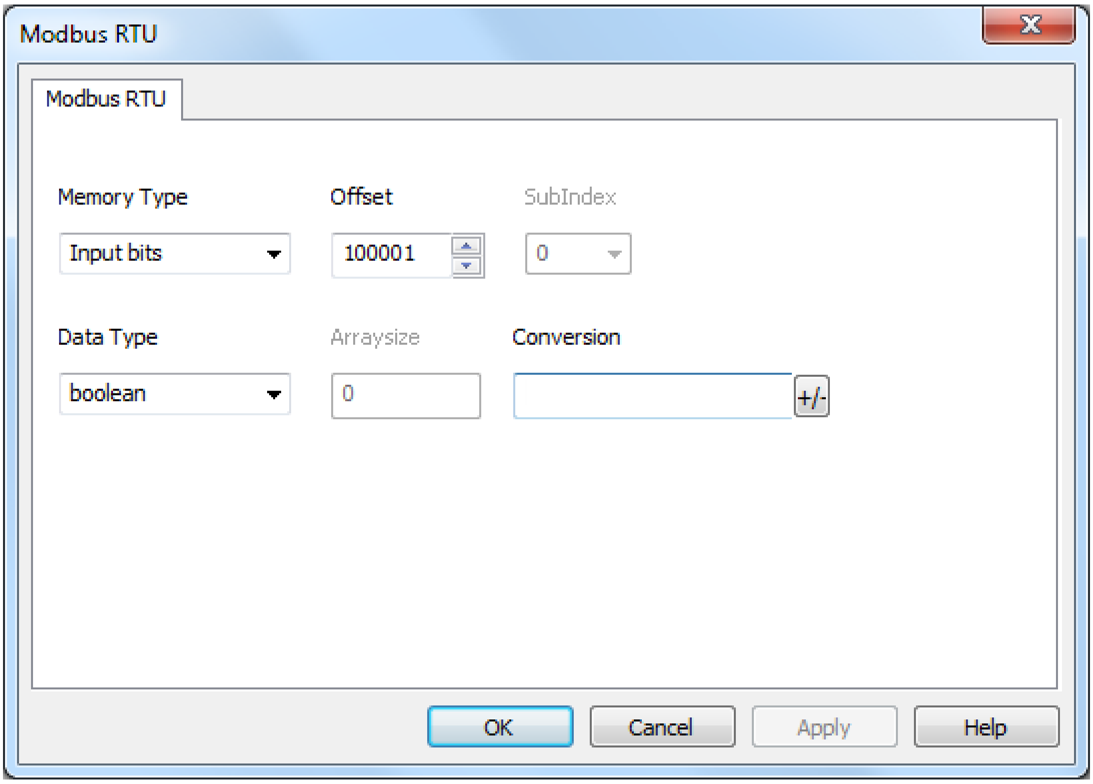
Element	Description
Max read block	Maximum length in bytes of a data block request. It applies only to read access of Holding Registers.
Max read bit block	Maximum length in bits of a block request. It applies only to read access of Input Bits and Output Coils.
Write Holding Register	Modbus function for write operations to Holding Registers. Select between the function **06** (preset single register) and function **16** (preset multiple registers). If function **06** is selected, the protocol will always use function **06** for writing to the controller, even when writing to multiple consecutive registers. If function **16** is selected, the protocol will always use function **16** to write to the controller, even for a single register write request and the **Max read block size** parameter of the query is set to **2**. The use of function **16** may result in higher communication performance.
Write Coils	Modbus function for write operations to Output Coils. Select between the function **05** (write single coil) and function **15** (write multiple coils). If Modbus function **05** is selected, the protocol will always use function **05** for writing to the controller, even when writing to multiple consecutive coils. If Modbus function **15** is selected, the protocol will always use function **15** to write to the controller, even for a single coil write request. The use of function **15** may result in higher communication performance.
Transmission Mode	• **RTU**: use RTU mode • **ASCII**: use ASCII mode Note: When PLC network is active, all nodes will be configured with the same Transmission Mode.
PLC Models	Allows to select between different PLC models: • **Modicon Modbus (1-based)**: Modbus implementation where all resources starts with offset 1. • **Generic Modbus (0-based)**: Modbus implementation where all resources starts with offset 0. • **Enron Modbus (1-based)**: Extends Modicon Mobdus implementation with 32 bit registers memory area. • **Enron Modbus (0-base)**: Extends Generic Modbus implementation with 32 bit registers memory area. 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.

<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td></td><td>

Comm Parameter Dialog
OK
Port: com1
Baudrate: 19200
Parity: Even
Data Bits: 8
Stop bits: 1
Mode: RS-485
<table>
<tr><th>Element</th><th>Parameter</th></tr>
<tr><td><b>Port</b></td><td>Serial port selection.<ul><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><b>Baudrate, Parity, Data Bits, Stop bits</b></td><td>Serial line parameters.</td></tr>
<tr><td><b>Mode</b></td><td>Serial port mode. Available modes:<ul><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>
</table>
</td></tr>
<tr><td><b>PLC Network</b></td><td>Multiple controllers can be connected to one HMI device. To set-up multiple connections, select <b>PLC networkand</b> click <b>Add</b> to configure each slave</td></tr>
</table>

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

Memory Type: Input bits

Offset: 100001

SubIndex: 0

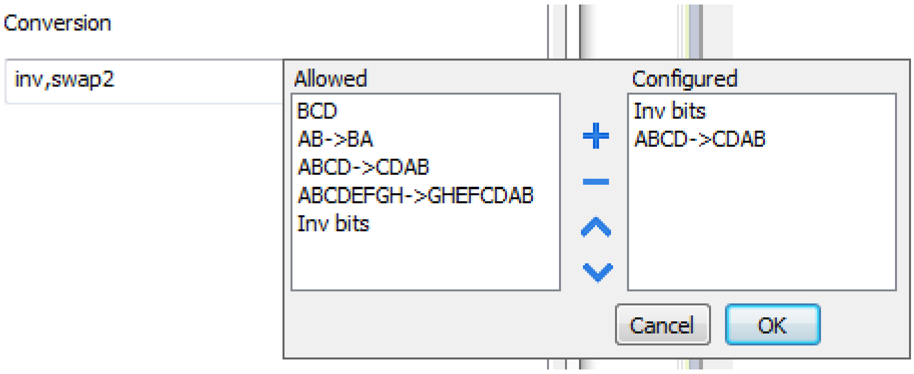
Data Type: boolean

Arraysize: 0

Conversion:

<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td><b>Memory Type</b></td><td>Modbus resource where tag is located.
<table>
<tr><th>Memory Type</th><th>Description</th></tr>
<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>
</table>
</td></tr>
<tr><td><b>Offset</b></td><td>Offset address where tag is located.
Offset addresses are six digits composed by one digit data type prefix + five digits resource address.
<table>
<tr><th>Memory Type</th><th>Studio Offset range</th><th>Modicon Offset range</th><th>Generic Modbus Offset range</th></tr>
<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>
</table>
</td></tr>
<tr><td><b>SubIndex</b></td><td>This allows resource offset selection within the register.</td></tr>
</table>

<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td><b>Data Type</b></td><td>
<table>
<tr><th>Data Type</th><th>Memory Space</th><th>Limits</th></tr>
<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>
<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>
</table>
Note: to define arrays. select one of Data Type format followed by square brackets like “byte[]”, “short[]”…
</td></tr>
<tr><td><b>Arraysize</b></td><td>
<ul>
<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>
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.
</td></tr>
<tr><td><b>Conversion</b></td><td>Conversion to be applied to the tag.</td></tr>
</table>

<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td></td><td>


Conversion

inv,swap2

Allowed:
BCD
AB->BA
ABCD->CDAB
ABCDEFGH->GHEFCDAB
Inv bits

Configured:
Inv bits
ABCD->CDAB

Depending on data type selected, the list <b>Allowed</b> shows one or more conversion types.

<table>
<tr><th>Value</th><th>Description</th></tr>
<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 -> 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 -> 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 -> 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 -> OPM...DAB</b></td><td><b>swap8</b>: Swap bytes in a long word.
Example:
142.366 → -893553517.588905 (in decimal format)
0 10000000110
0001110010111011011001000101101000011100101011000001</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td></td><td>→
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>
</table>
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.
</td></tr>
</table>

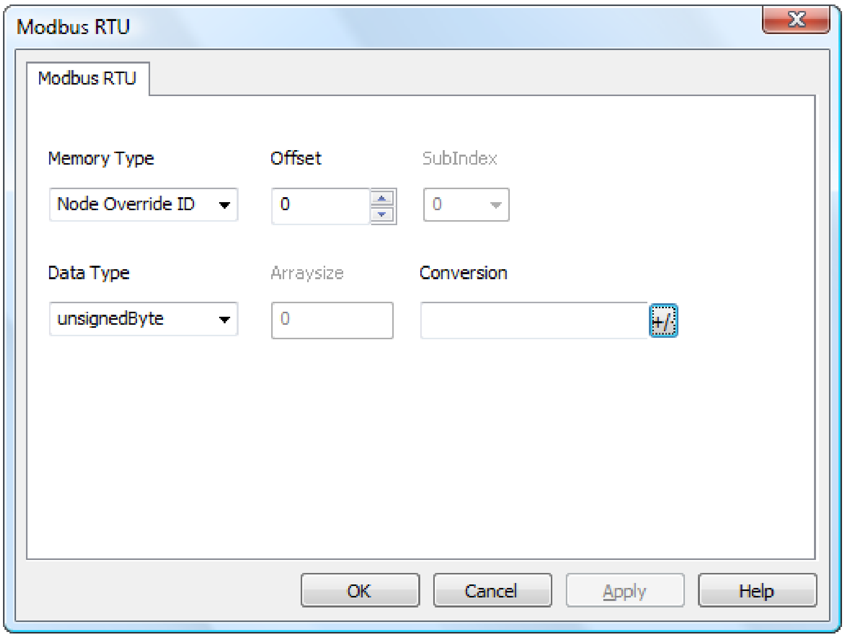
Node Override ID

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.



Modbus RTU

Memory Type: Node Override ID

Offset: 0

SubIndex: 0

Data Type: unsignedByte

Arraysize: 0

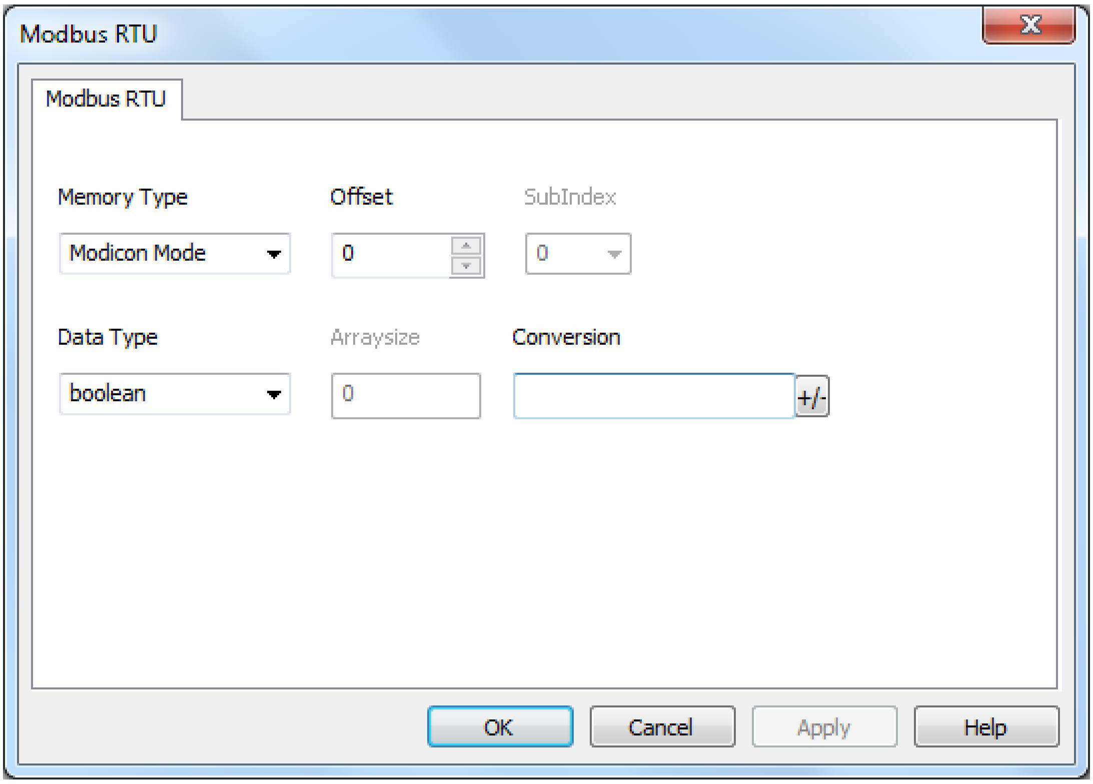
Conversion:

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.



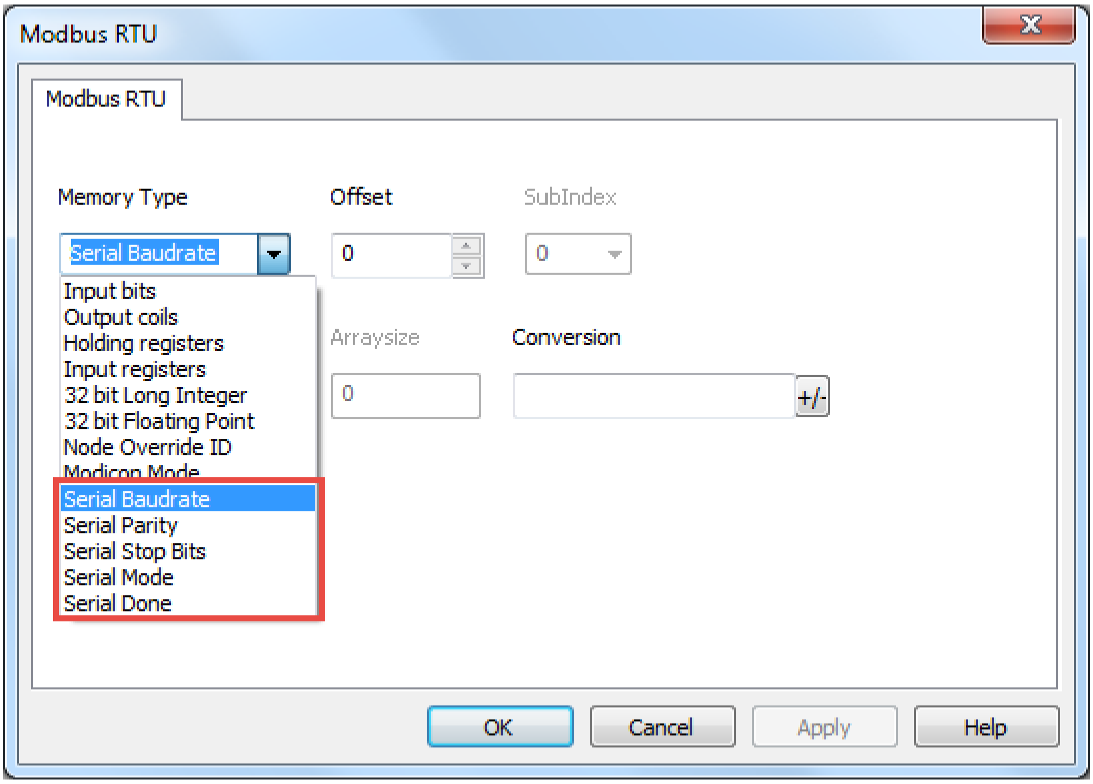
Memory Type: Modicon Mode
Offset: 0
SubIndex: 0
Data Type: boolean
Arraysize: 0
Conversion:

Serial Parameters Override

The protocol provide special data types that can be used to override the serial parameters at runtime.

Parameter	Description
Serial Baudrate	unsigned 32 bit value for baudrate overriding. Possible values are 150, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200.
Serial Parity	unsigned 8 bit value for parity overriding. Possible values are described in the following list. **Value** – **Description** **0** – none parity **1** – even parity **2** – odd parity
Serial Stop Bits	unsigned 8 bit value for stop bits overriding. Possible values are 1, 2.
Serial Mode	unsigned 8 bit value for serial mode overriding. Possible values are described in the following list.

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td></td><td><table><tr><th>Value</th><th>Description</th></tr><tr><td><b>0</b></td><td>RS-232 mode</td></tr><tr><td><b>1</b></td><td>RS-485 mode</td></tr><tr><td><b>2</b></td><td>RS-422 mode</td></tr></table></td></tr>
<tr><td><b>Serial Done</b></td><td>Set to 1 to overwrite the communication line parameters. The parameters are processed all together only when this variable is set to value 1</td></tr>
</table>



Modbus RTU

Memory Type: Serial Baudrate

Offset: 0

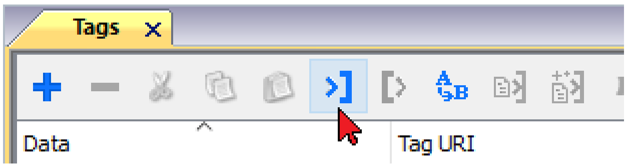
SubIndex: 0

Arraysize: 0

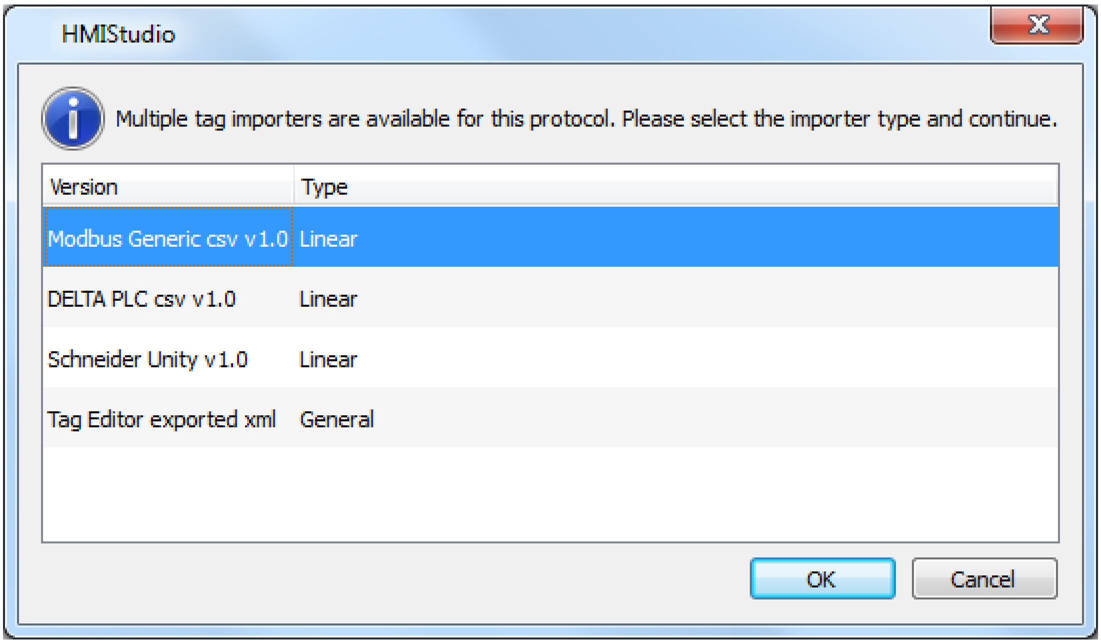
Conversion:

Tag Import

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



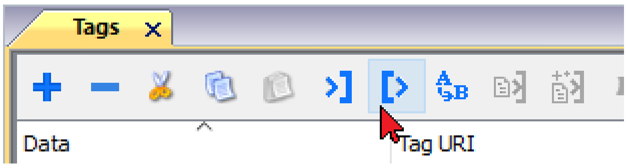
The following dialog shows which importer type can be selected.



HMIStudio

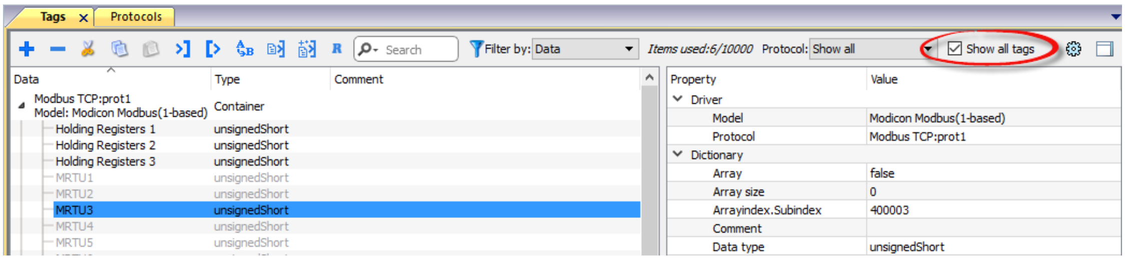
Multiple tag importers are available for this protocol. Please select the importer type and continue.

Version	Type
Modbus Generic csv v1.0	Linear
DELTA PLC csv v1.0	Linear
Schneider Unity v1.0	Linear
Tag Editor exported xml	General

Type	Description
Modbus Generic csv v1.0 Linear	Requires a **.csv** file. All variables will be displayed at the same level.
DELTA PLC csv v1.0	Requires a **.csv** file. All variables will be displayed at the same level.
Schneider Unity v1.0 Linear	Requires a **.uny** file. The file containing symbols must be exported in **.txt** format and later renamed as **.uny**. The importer considers only variables located at fixed address and disregards arrays of strings. All other arrays, except for boolean type, are expanded.
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.

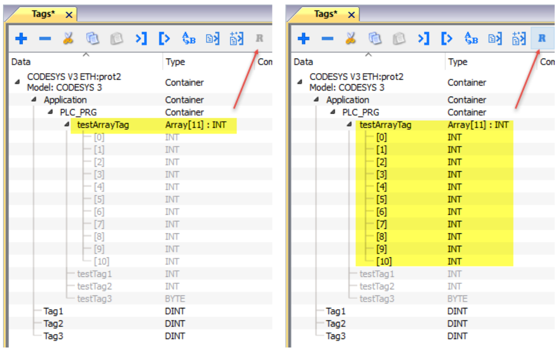


Tags × | Protocols

Search | Filter by: Data | Items used:6/10000 | Protocol: Show all | ☑ Show all tags

Data	Type	Comment
Modbus TCP:prot1 Model: Modicon Modbus(1-based)	Container	
Holding Registers 1	unsignedShort	
Holding Registers 2	unsignedShort	
Holding Registers 3	unsignedShort	
MRTU1	unsignedShort	
MRTU2	unsignedShort	
MRTU3	unsignedShort	
MRTU4	unsignedShort	
MRTU5	unsignedShort	

Property	Value
Driver	
Model	Modicon Modbus(1-based)
Protocol	Modbus TCP:prot1
Dictionary	
Array	false
Array size	0
Arrayindex.Subindex	400003
Comment	
Data type	unsignedShort

Toolbar item	Description
	Import Tag(s). Select tags to be imported and click on this icon to add tags from tag dictionary to the project
	Update Tag(s). Click on this icon to update the tags in the project, due a new dictionary import.
R	Check this box to import all sub-elements of a tag. Example of both checked and unchecked result:  Tags* × Data | Type | Con CODESYS V3 ETH:prot2 Model: CODESYS 3 | Container Application | Container PLC_PRG | Container testArrayTag | Array[11] : INT [0] | INT [1] | INT [2] | INT [3] | INT [4] | INT [5] | INT [6] | INT [7] | INT [8] | INT [9] | INT [10] | INT testTag1 | INT testTag2 | INT testTag3 | BYTE Tag1 | DINT Tag2 | DINT Tag3 | DINT
 Search Filter by: Tag name	Searches tags in the dictionary basing on filter combo-box item selected.

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
NodeID	Node the tag belongs to
TagName	Tag description
MemoryType	• OUTP • INP • IREG • HREG
Address	Offset compatible with Modbus notation
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.

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 .	-
The received message too short	The device received a response with an invalid format from the controller .	-
Incorrect writing data acknowledge	The controller did not accept a write request.	Check if project data is consistent with the controller resources.

Modbus RTU Server

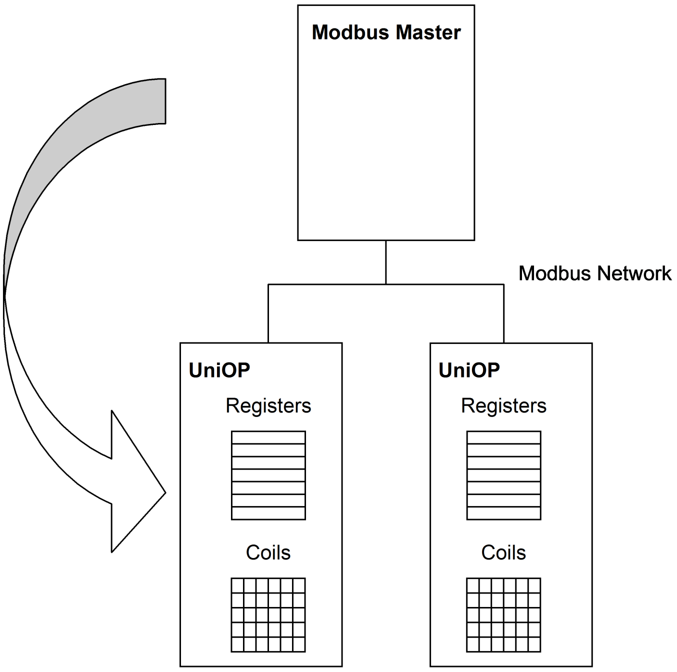
Modbus RTU Server communication driver allows connecting the HMI device as a slave in a Modbus RTU network. Standard Modbus messages are used for information exchange.

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	**Illegal Function**. the function code received in the query is not supported
02	**Illegal Data Address**. Data Address received in the query exceeds the predefined data range (see **Tag Definition** for detailed ranges of all types).
03	**Illegal Data Value**. A sub function other than 00 is specified in Loopback Diagnostic Test (Code 08).

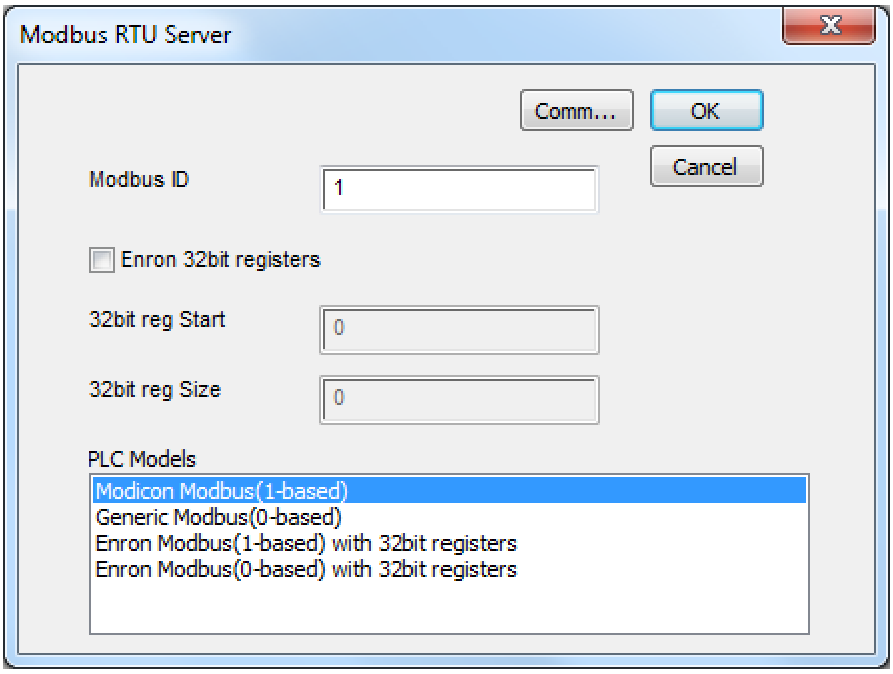
Protocol Editor Settings

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 ID: 1

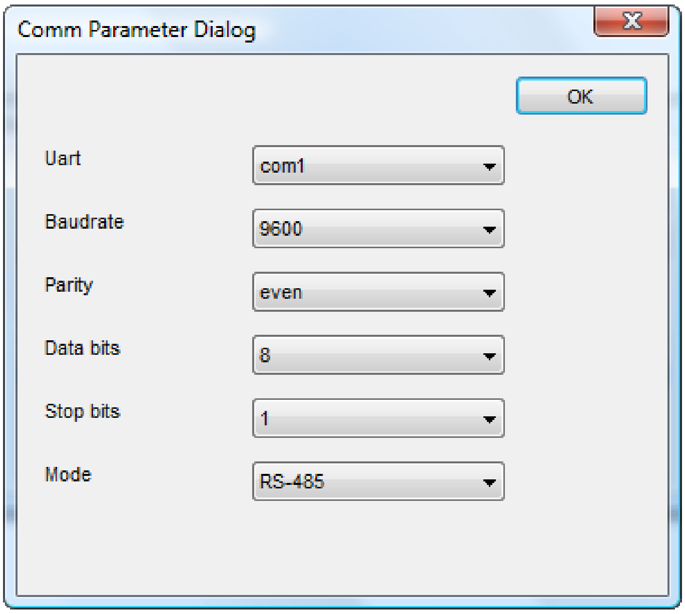
□ Enron 32bit registers

32bit reg Start: 0

32bit reg Size: 0

PLC Models:
- Modicon Modbus(1-based)
- Generic Modbus(0-based)
- Enron Modbus(1-based) with 32bit registers
- Enron Modbus(0-based) with 32bit registers

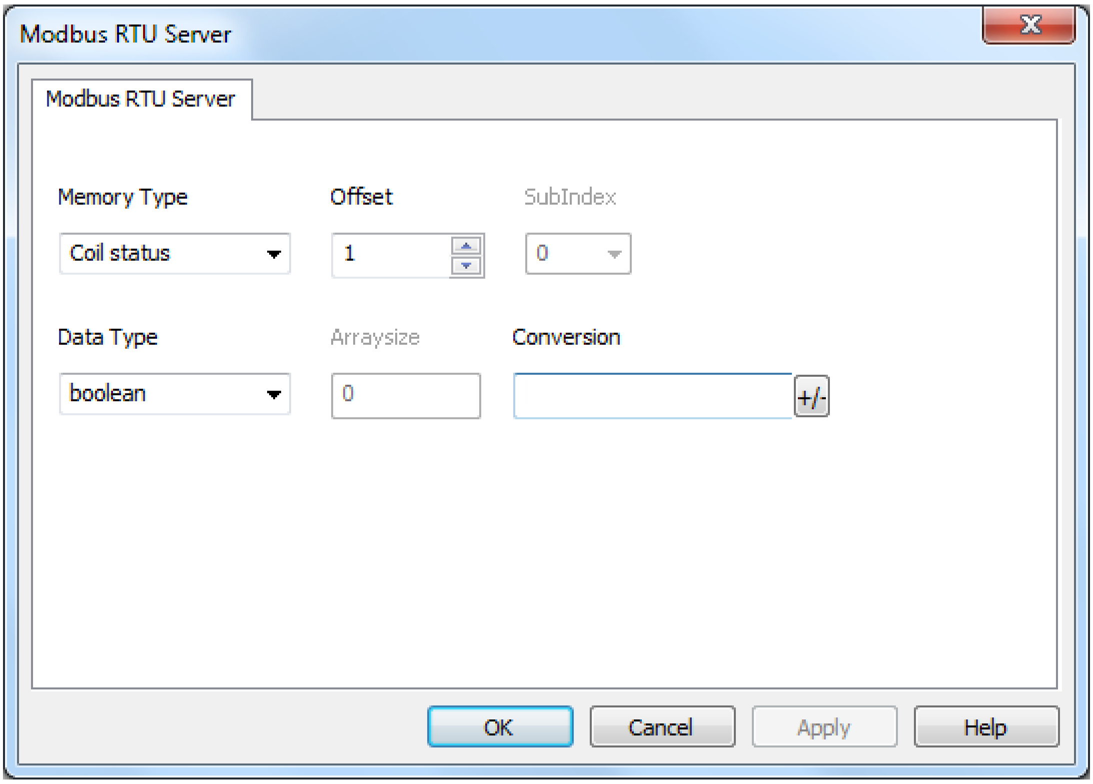
Element	Description
Modbus ID	Modbus node ID. Every Modbus server device in the network must have its own Modbus ID.
Enron 32bit registers	If selected, allows to define the first register address and the number of registers for 32 bit registers memory area. Note: 32 bit registers are available only for **Enron Modbus** PLC Models.
32bit reg Start **32bit reg Size**	32 bit registries memory area definition. **Start** value represents the first register address. **Size** value represents the number of registries. Note: A request to one of the registries inside this area gives a 4 byte answer.
PLC Models	Allows to select between different PLC models: • **Modicon Modbus (1-based)**: Modbus implementation where all resources starts with offset 1. • **Generic Modbus (0-based)**: Modbus implementation where all resources starts with offset 0. • **Enron Modbus (1-based)**: Extends Modicon Mobdus implementation with 32 bit registers memory area. • **Enron Modbus (0-base)**: Extends Generic Modbus implementation with 32 bit registers memory area.

<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td></td><td>ℹ 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.</td></tr>
<tr><td><b>Comm...</b></td><td>If clicked, displays the communication parameters setup dialog.
You have to set parameters according to the values programmed in Modbus Master.

Comm Parameter Dialog
Uart: com1
Baudrate: 9600
Parity: even
Data bits: 8
Stop bits: 1
Mode: RS-485
<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td><b>Uart</b></td><td>Serial port selection.<ul><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><b>Baudrate, Parity, Data bits, Stop bits</b></td><td>Serial line parameters.</td></tr>
<tr><td><b>Mode</b></td><td>Serial port mode. Available options:<ul><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>
</table></td></tr>
</table>

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.



Modbus RTU Server

Modbus RTU Server

Memory Type: Coil status

Offset: 1

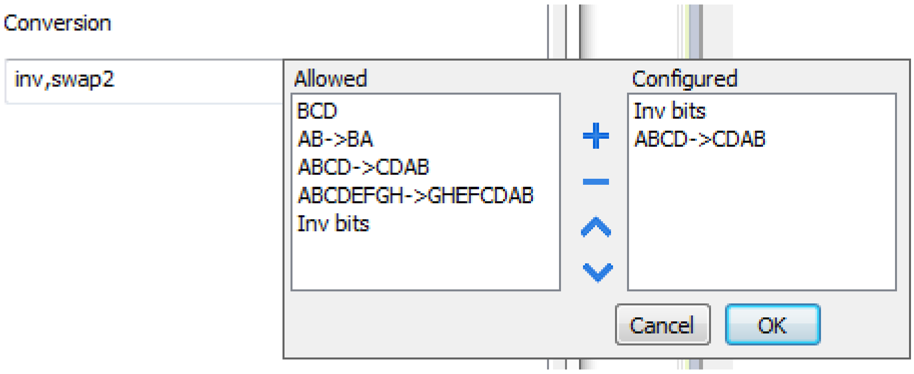
SubIndex: 0

Data Type: boolean

Arraysize: 0

Conversion:

<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td><b>Memory Type</b></td><td>Modbus resource where tag is located.
<table>
<tr><th>Memory Type</th><th>Modbus Resource</th></tr>
<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>
</table></td></tr>
<tr><td><b>Offset</b></td><td>Offset address where tag is located.
Offset addresses are six digits composed by one digit data type prefix + five digits resource address.
<table>
<tr><th>Memory Type</th><th>Studio Offset range</th><th>Modicon Offset range</th><th>Generic Modbus Offset range</th></tr>
<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>
</table></td></tr>
<tr><td><b>SubIndex</b></td><td>This allows resource offset selection within the register.</td></tr>
</table>

<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td><b>Data type</b></td><td>
<table>
<tr><th>Data Type</th><th>Memory Space</th><th>Limits</th></tr>
<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>
<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>
</table>
Note: to define arrays. select one of Data Type format followed by square brackets like “byte[]”, “short[]”…
</td></tr>
<tr><td><b>Arraysize</b></td><td>When configuring array or string tags, this option define the amount of array elements or characters of the string.</td></tr>
<tr><td><b>Conversion</b></td><td>Conversion to be applied to the tag.

Conversion: inv,swap2
Allowed: BCD, AB->BA, ABCD->CDAB, ABCDEFGH->GHEFCDAB, Inv bits
Configured: Inv bits, ABCD->CDAB

Depending on data type selected, the list <b>Allowed</b> shows one or more conversion types.</td></tr>
</table>

<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>Value</th><th>Description</th></tr>
<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 -> 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 -> 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 -> 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 -> OPM...DAB</b></td><td><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)</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>
</table>
Select conversion and click +. The selected item will be added to list <b>Configured</b>.</td></tr>
</table>

Element	Description
	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.

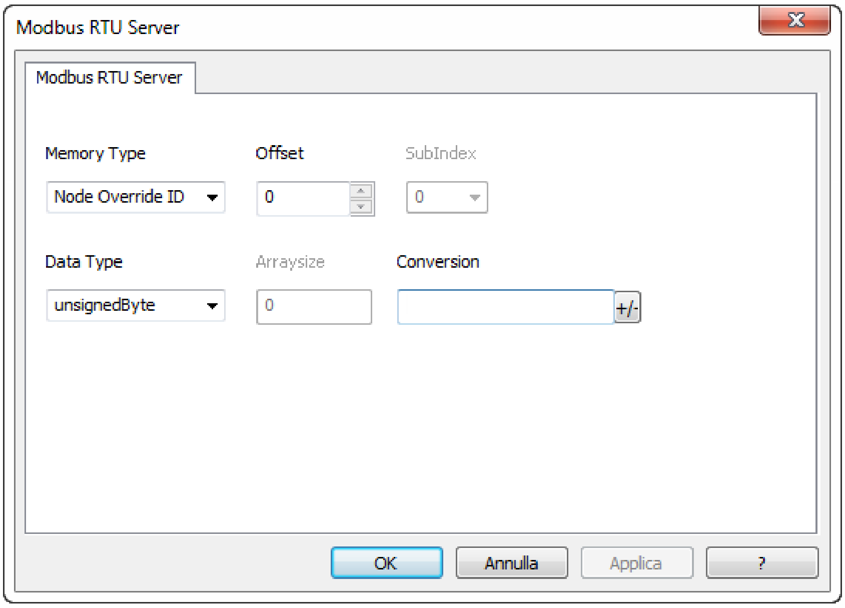
Node Override ID

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.


Modbus RTU Server

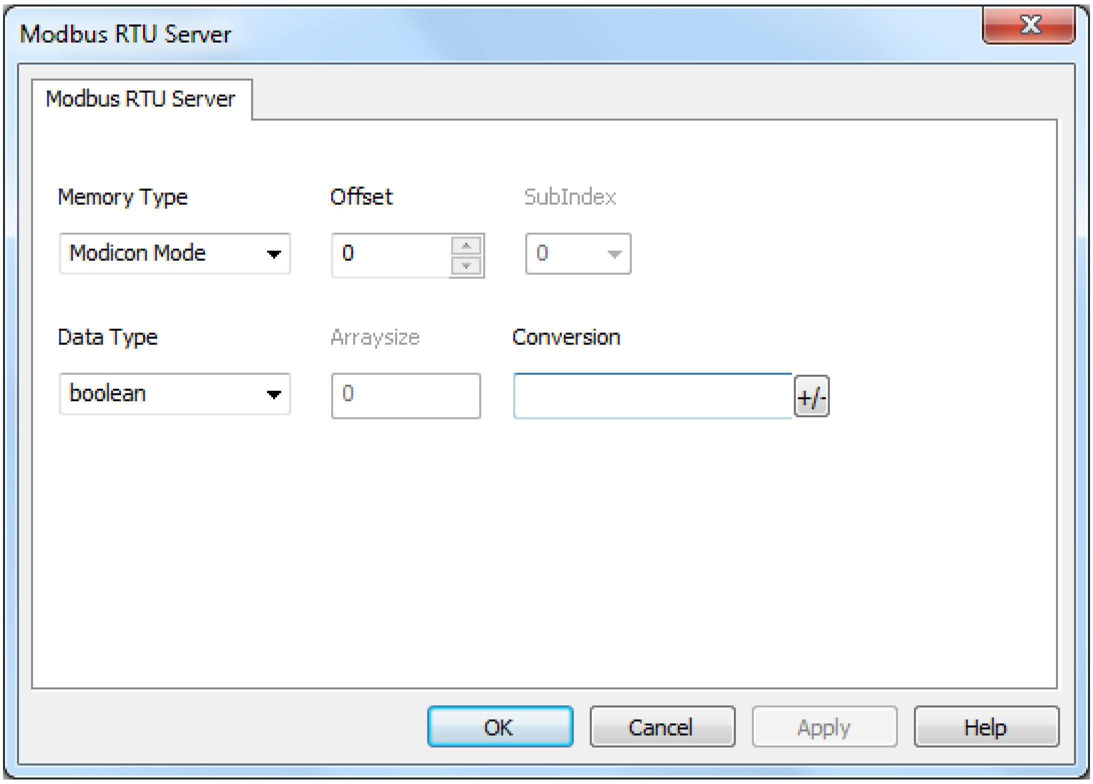
Memory Type: Node Override ID
Offset: 0
SubIndex: 0
Data Type: unsignedByte
Arraysize: 0
Conversion:

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.



Modbus RTU Server

Memory Type: Modicon Mode

Offset: 0

SubIndex: 0

Data Type: boolean

Arraysize: 0

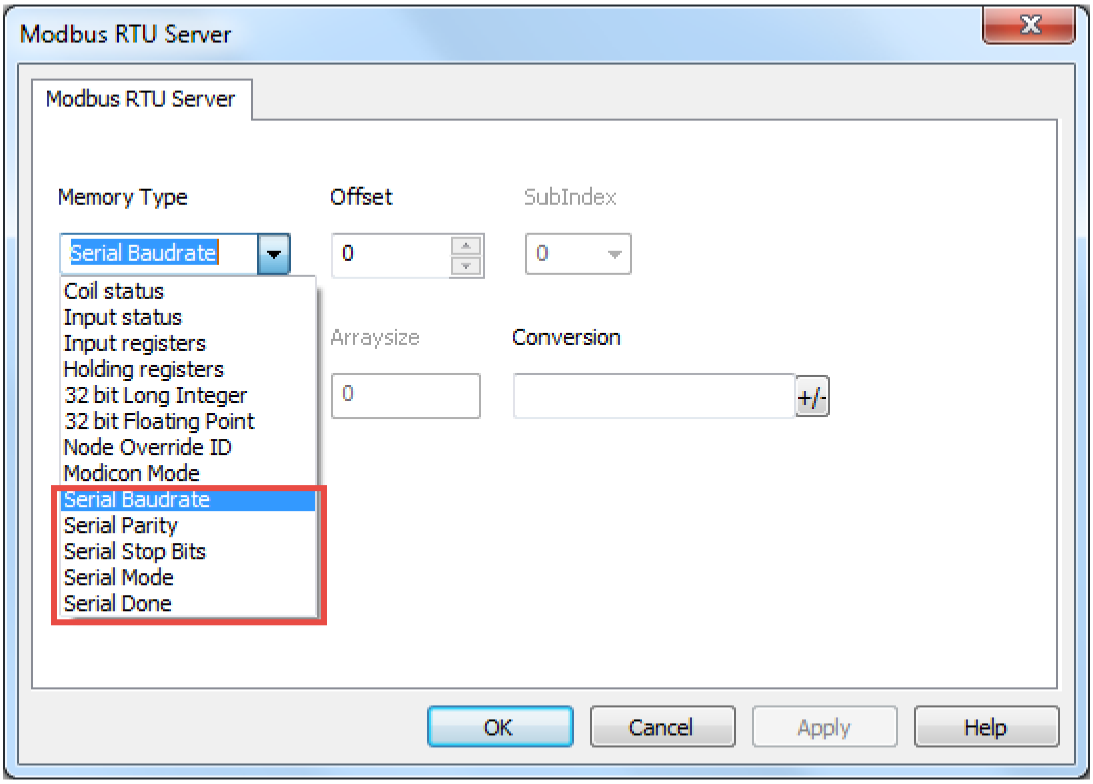
Conversion:

Serial Parameters Override

The protocol provide special data types that can be used to override the serial parameters at runtime.

Parameter	Description
Serial Baudrate	unsigned 32 bit value for baudrate overriding. Possible values are 150, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200.
Serial Parity	unsigned 8 bit value for parity overriding. Possible values are described in the following list.

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td></td><td><table><tr><th>Value</th><th>Description</th></tr><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></table></td></tr>
<tr><td><b>Serial Stop Bits</b></td><td>unsigned 8 bit value for stop bits overriding. Possible values are 1, 2.</td></tr>
<tr><td><b>Serial Mode</b></td><td>unsigned 8 bit value for serial mode overriding. Possible values are described in the following list.<table><tr><th>Value</th><th>Description</th></tr><tr><td>0</td><td>RS-232 mode</td></tr><tr><td>1</td><td>RS-485 mode</td></tr><tr><td>2</td><td>RS-422 mode</td></tr></table></td></tr>
<tr><td><b>Serial Done</b></td><td>Set to 1 to overwrite the communication line parameters. The parameters are processed all together only when this variable is set to value 1</td></tr>
</table>



Modbus RTU Server

Memory Type: Serial Baudrate

Offset: 0

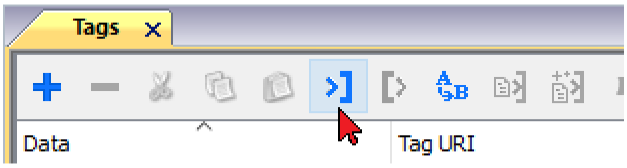
SubIndex: 0

Arraysize: 0

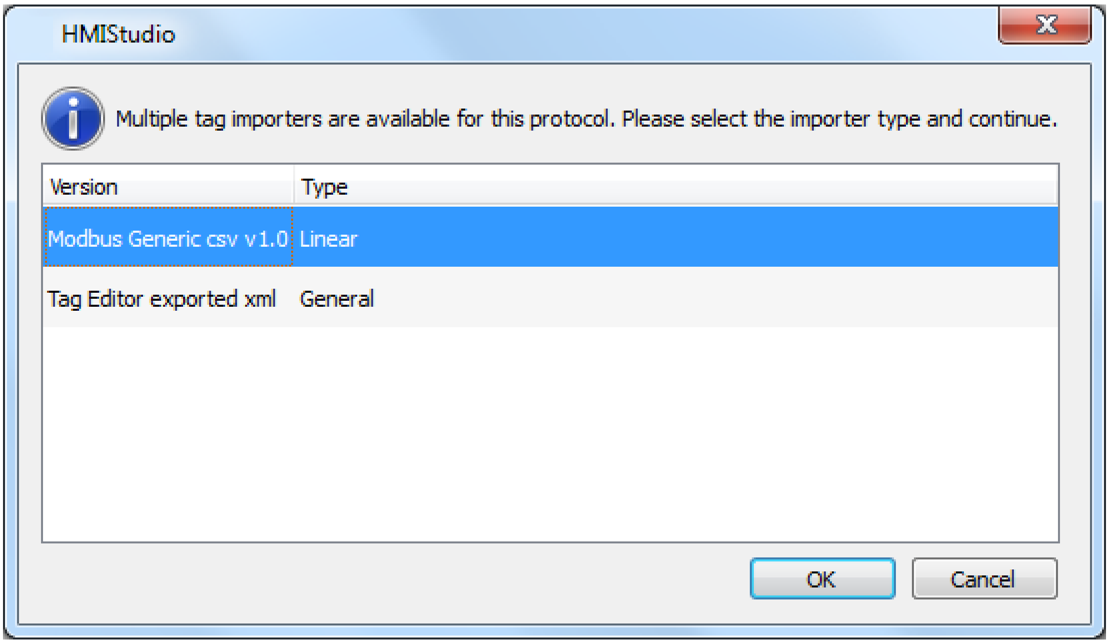
Conversion:

Tag Import

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

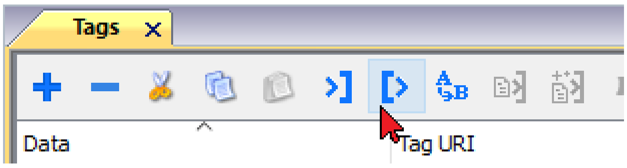


The following dialog shows which importer type can be selected.


HMIStudio

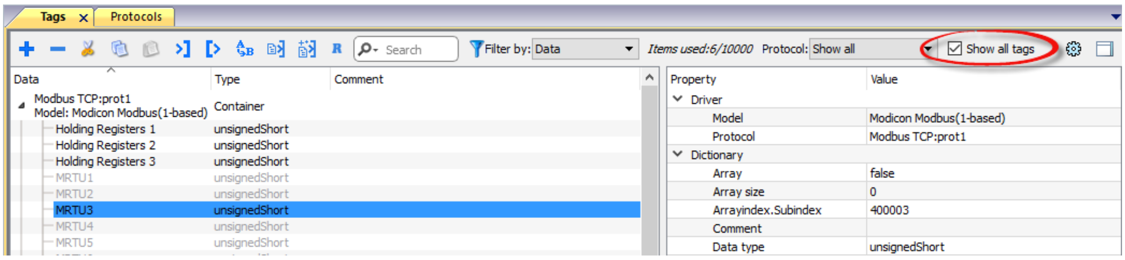
Multiple tag importers are available for this protocol. Please select the importer type and continue.

Version	Type
Modbus Generic csv v1.0	Linear
Tag Editor exported xml	General

Type	Description
Modbus Generic csv v1.0 **Linear**	Requires a **.csv** file. 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. 

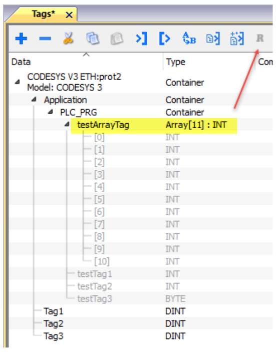
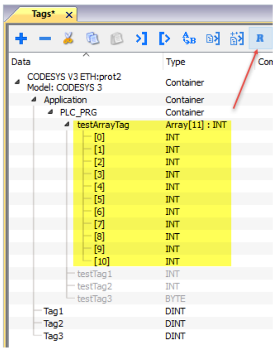
Once the importer has been selected, locate the symbol file and click **Open**.

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.



Data	Type	Comment
Modbus TCP:prot1 Model: Modicon Modbus(1-based)	Container	
Holding Registers 1	unsignedShort	
Holding Registers 2	unsignedShort	
Holding Registers 3	unsignedShort	
MRTU1	unsignedShort	
MRTU2	unsignedShort	
MRTU3	unsignedShort	
MRTU4	unsignedShort	
MRTU5	unsignedShort	

Property	Value
Driver	
Model	Modicon Modbus(1-based)
Protocol	Modbus TCP:prot1
Dictionary	
Array	false
Array size	0
Arrayindex.Subindex	400003
Comment	
Data type	unsignedShort

Toolbar item	Description
	Import Tag(s). Select tags to be imported and click on this icon to add tags from tag dictionary to the project
	Update Tag(s). Click on this icon to update the tags in the project, due a new dictionary import.
R	Check this box to import all sub-elements of a tag. Example of both checked and unchecked result: 
Search Filter by: Tag name ▾	Searches tags in the dictionary basing on filter combo-box item selected.

Data	Type
CODESYS V3 ETH:prot2 Model: CODESYS 3	Container
Application	Container
PLC_PRG	Container
testArrayTag	Array[11] : INT
[0]	INT
[1]	INT
[2]	INT
[3]	INT
[4]	INT
[5]	INT
[6]	INT
[7]	INT
[8]	INT
[9]	INT
[10]	INT
testTag1	INT
testTag2	INT
testTag3	BYTE
Tag1	DINT
Tag2	DINT
Tag3	DINT

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
NodeID	Node the tag belongs to
TagName	Tag description
MemoryType	• OUTP • INP • IREG • HREG
Address	Offset compatible with Modbus notation
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.

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.

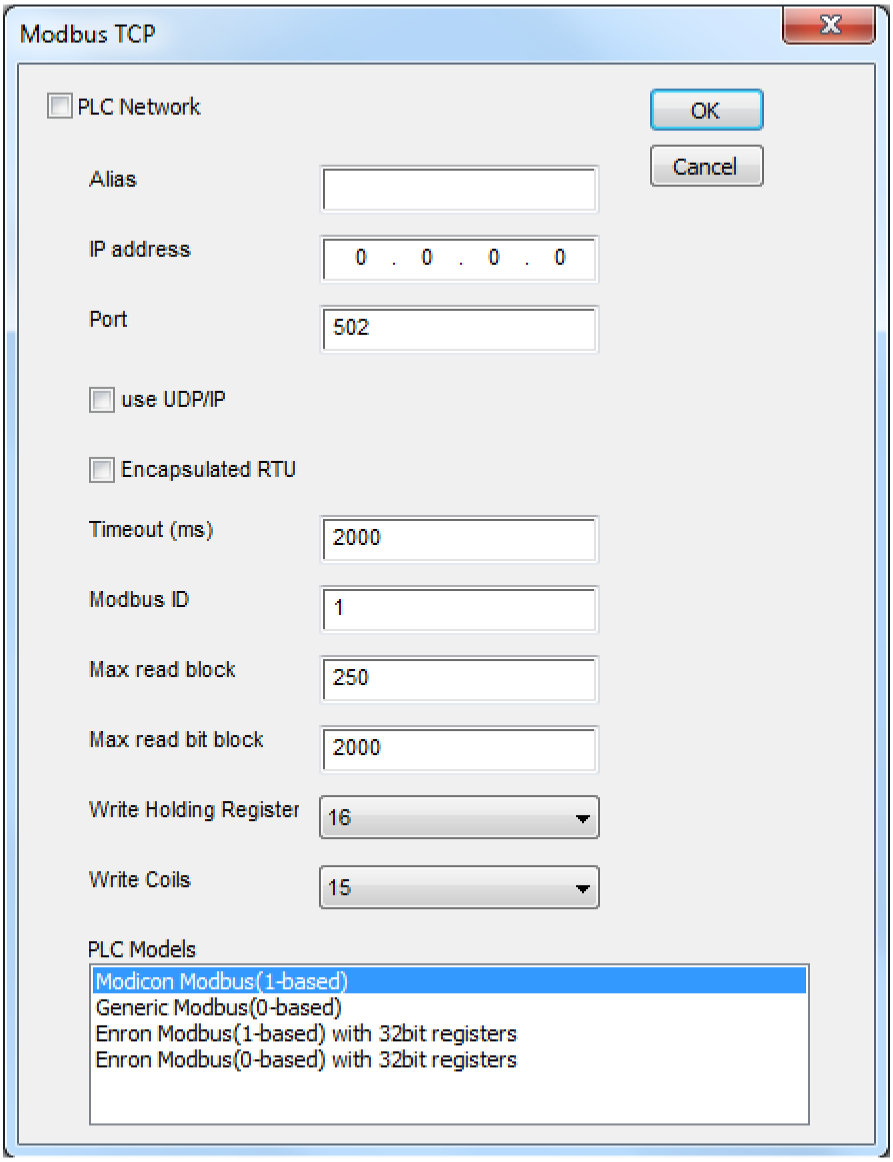
Protocol Editor Settings

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: 0 . 0 . 0 . 0

Port: 502

□ use UDP/IP

□ Encapsulated RTU

Timeout (ms): 2000

Modbus ID: 1

Max read block: 250

Max read bit block: 2000

Write Holding Register: 16

Write Coils: 15

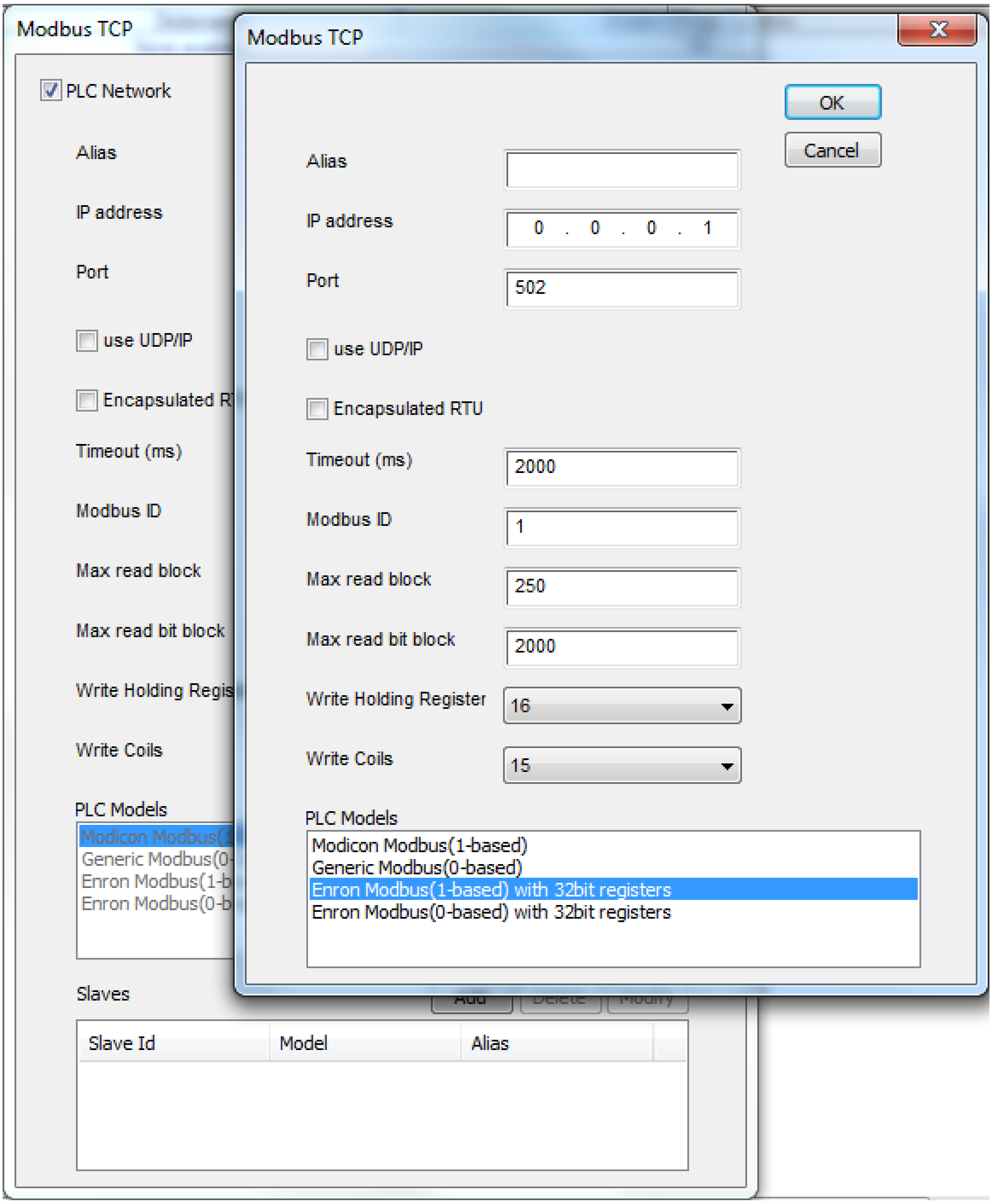
PLC Models

- Modicon Modbus(1-based)
- Generic Modbus(0-based)
- Enron Modbus(1-based) with 32bit registers
- Enron Modbus(0-based) with 32bit registers

Element	Description
Alias	Name identifying nodes in network configurations. The name will be added as a prefix to each tag name imported for each network node.
IP address	Address of the controller.
Port	Port number used by the Modbus TCP driver. The default value is **502** and can be changed when the communication goes through routers or Internet gateways where the default port number is already in use.
use UDP/IP	If selected, the protocol will use connectionless UDP datagrams.
Encapsulated RTU	If selected, the protocol will use serial RTU protocol over Ethernet instead of Modbus TCP protocol, independently from TCP or UDP usage.
Timeout (ms)	Time delay in milliseconds between two retries in case of missing response from the server device.

Element	Description
Modbus ID	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.
Max read block	Maximum length in bytes of a data block request. It applies only to read access of Holding Registers.
Max read bit block	Maximum length in bits of a block request. It applies only to read access of Input Bits and Output Coils.
Write Holding Register	Modbus function for write operations to Holding Registers. Select between the function **06** (preset single register) and function **16** (preset multiple registers). If **06** is selected, the protocol will always use function **06** for writing to the controller, even when writing to multiple consecutive registers. If **16** is selected, the protocol will always use function **16** to write to the controller, even for a single register write request and the **Max read block size** parameter of the query is set to **2**. The use of function **16** may result in higher communication performance. If **Auto** is selected, the protocol will use both function **06** or function **16** depending on number of registries to be written.
Write Coils	Modbus function for write operations to Output Coils. Select between the function **05** (write single coil) and function **15** (write multiple coils). If Modbus function **05** is selected, the protocol will always use function **05** for writing to the controller, even when writing to multiple consecutive coils. If Modbus function **15** is selected, the protocol will always use function **15** to write to the controller, even for a single coil write request. The use of function **15** may result in higher communication performance.

Element	Description
PLC Models	Allows to select between different PLC models: • **Modicon Modbus (1-based)**: Modbus implementation where all resources starts with offset 1. • **Generic Modbus (0-based)**: Modbus implementation where all resources starts with offset 0. • **Enron Modbus (1-based)**: Extends Modicon Mobdus implementation with 32 bit registers memory area. • **Enron Modbus (0-base)**: Extends Generic Modbus implementation with 32 bit registers memory area. 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.
PLC Network	IP address for all controllers in multiple connections. **PLC Network** must be selected to enable multiple connections.

Element	Description
	

Modbus TCP

☑ PLC Network

Modbus TCP

Alias:

IP address: 0 . 0 . 0 . 1

Port: 502

☐ use UDP/IP

☐ Encapsulated RTU

Timeout (ms): 2000

Modbus ID: 1

Max read block: 250

Max read bit block: 2000

Write Holding Register: 16

Write Coils: 15

PLC Models

- Modicon Modbus(1-based)
- Generic Modbus(0-based)
- Enron Modbus(1-based) with 32bit registers
- Enron Modbus(0-based) with 32bit registers

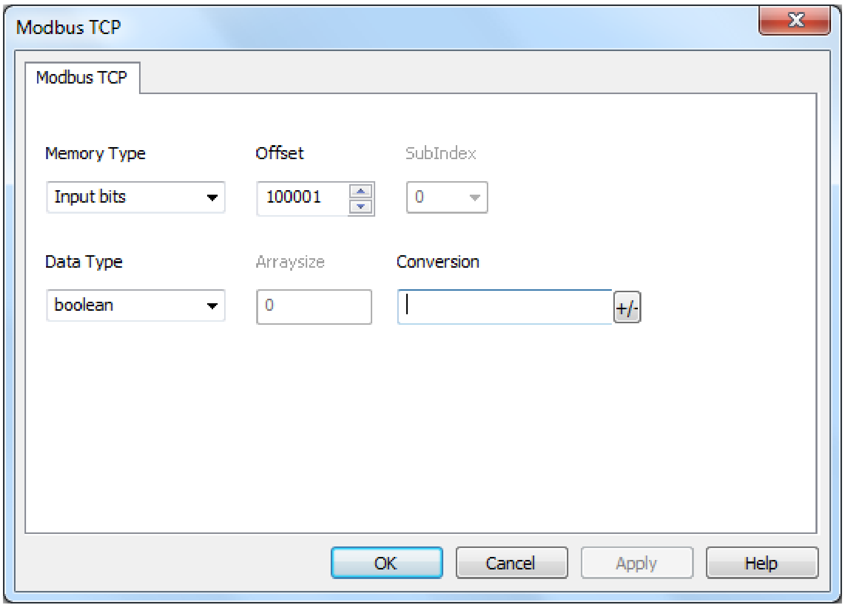
Slaves

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.



Memory Type: Input bits

Offset: 100001

SubIndex: 0

Data Type: boolean

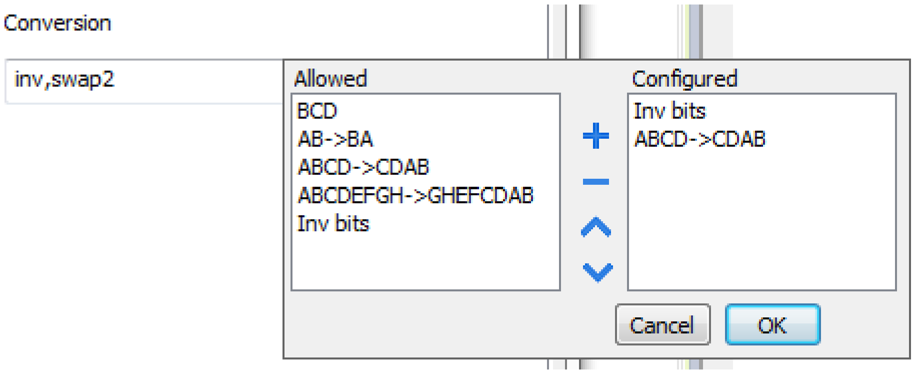
Arraysize: 0

Conversion:

<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td><b>Memory Type</b></td><td>Modbus resource where tag is located.
<table>
<tr><th>Memory Type</th><th>Modbus Resource</th></tr>
<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 IP</b></td><td rowspan="4">protocol parameter (see <b>Special Data Types</b> for mode details)</td></tr>
<tr><td><b>Node Override Port</b></td></tr>
<tr><td><b>Node Override ID</b></td></tr>
<tr><td><b>Modicon Mode</b></td></tr>
</table></td></tr>
<tr><td><b>Offset</b></td><td>Offset address where tag is located.
Offset addresses are six digits composed by one digit data type prefix + five digits resource address.</td></tr>
</table>

<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>Memory Type</th><th>Studio
Offset range</th><th>Modicon
Offset range</th><th>Generic Modbus
Offset range</th></tr>
<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>
</table>
</td></tr>
<tr><td><b>SubIndex</b></td><td>This allows resource offset selection within the register.</td></tr>
<tr><td><b>Data Type</b></td><td>
<table>
<tr><th>Data Type</th><th>Memory Space</th><th>Limits</th></tr>
<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>
<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>
</table>
Note: to define arrays. select one of Data Type format followed by square brackets like “byte[]”, “short[]”…
</td></tr>
<tr><td><b>Arraysize</b></td><td>
<ul>
<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>
</td></tr>
</table>

<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td></td><td>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.</td></tr>
<tr><td><b>Conversion</b></td><td>
Conversion to be applied to the tag.



Conversion

inv,swap2

Depending on data type selected, the list <b>Allowed</b> shows one or more conversion types.

<table>
<tr><th>Value</th><th>Description</th></tr>
<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 -> 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 -> 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 -> 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>
</table>
</td></tr>
</table>

<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td><b>ABC...NOP -> OPM...DAB</b></td><td><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)</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>
</table>
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.
</td></tr>
</table>

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.

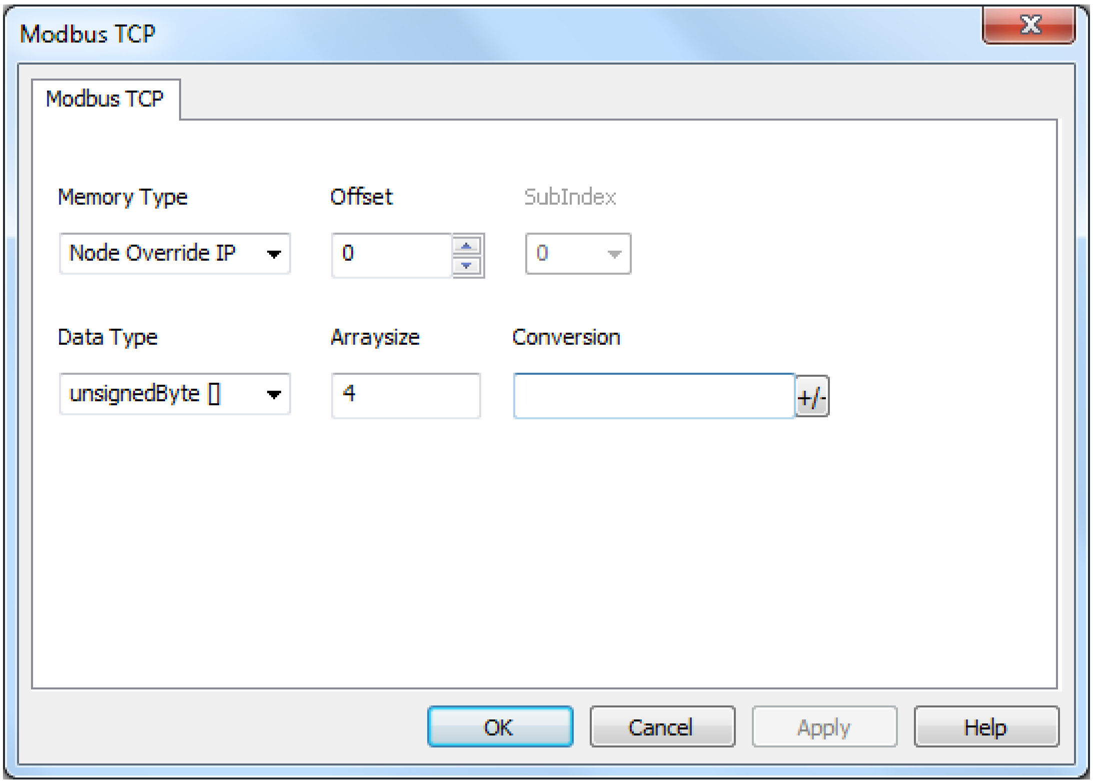
Node Override IP	PLC operation
0.0.0.0	Communication with the controller is stopped, no request frames are generated anymore.
Different from 0.0.0.0	It is interpreted as node IP override and the target IP address is replaced runtime with the new value.

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.



Memory Type: Node Override IP

Offset: 0

SubIndex: 0

Data Type: unsignedByte []

Arraysize: 4

Conversion: +/-

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.

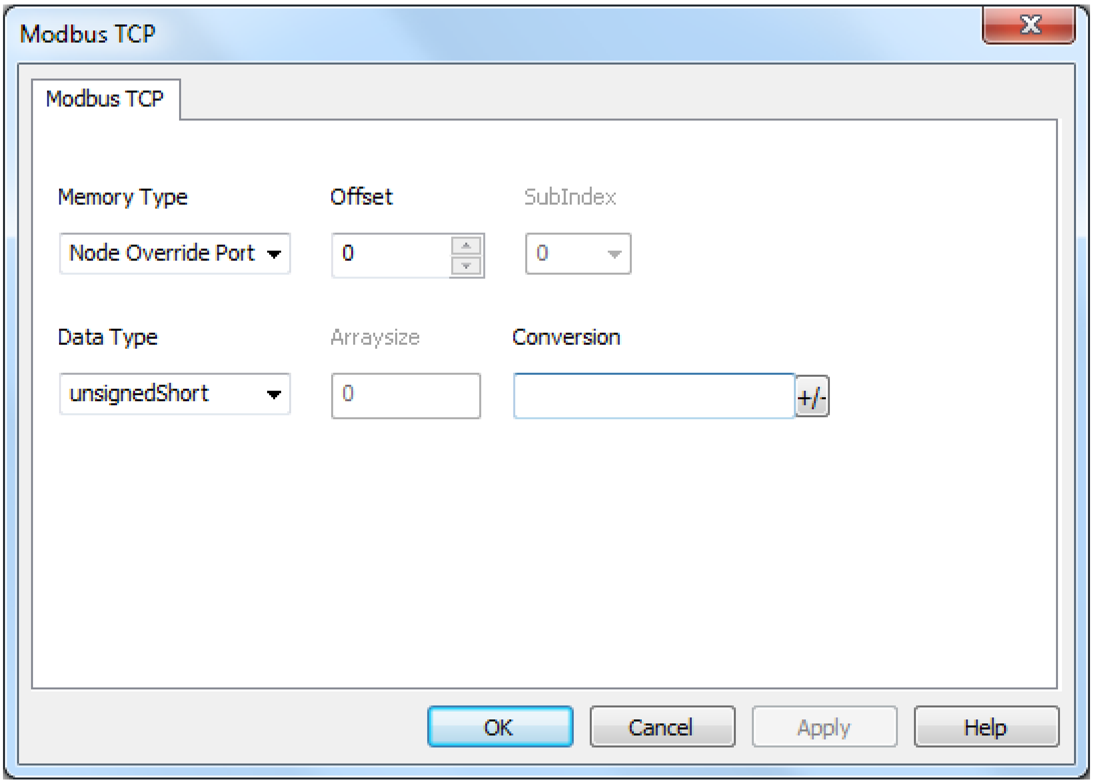
This memory type is unsigned short.

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.



Memory Type: Node Override Port

Offset: 0

SubIndex: 0

Data Type: unsignedShort

Arraysize: 0

Conversion:

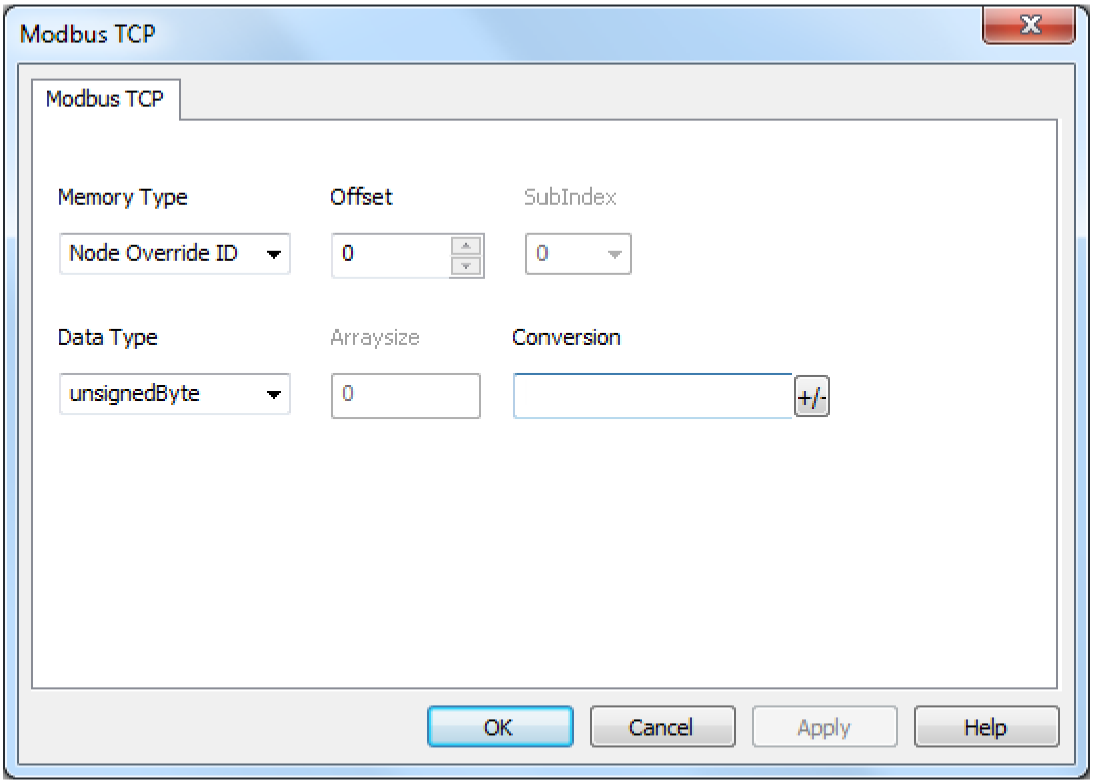
Node Override ID

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.



Memory Type: Node Override ID

Offset: 0

SubIndex: 0

Data Type: unsignedByte

Arraysize: 0

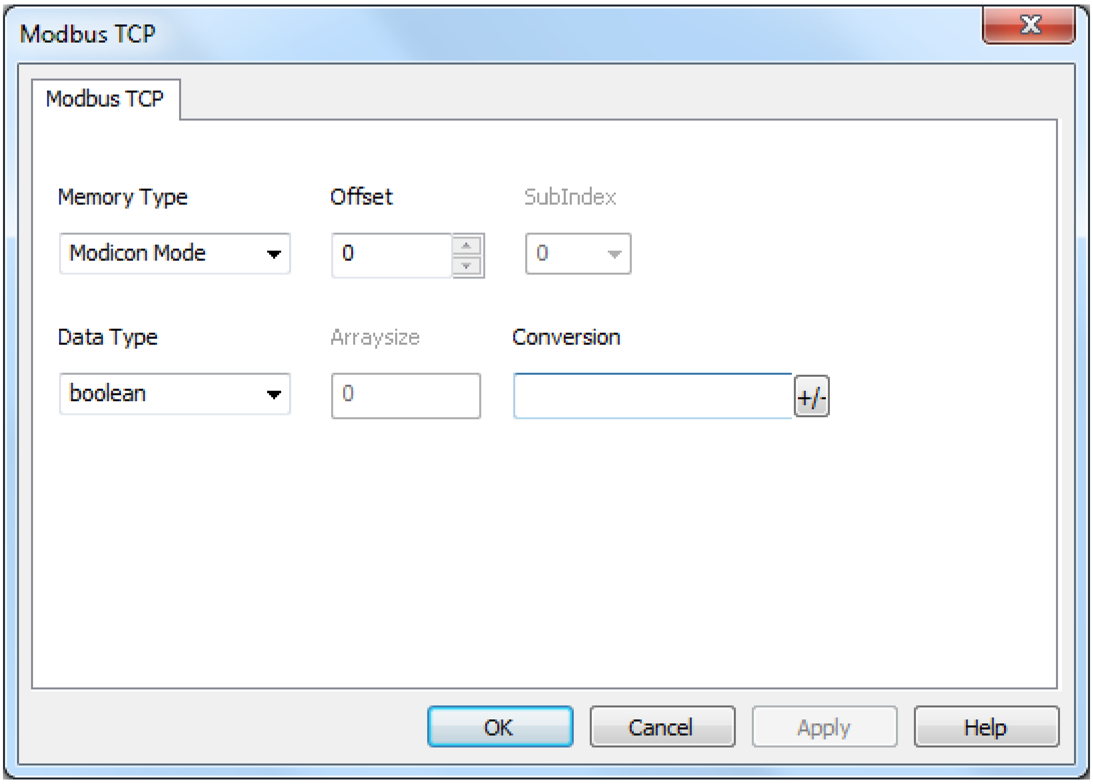
Conversion: +/-

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.



Modbus TCP

Memory Type: Modicon Mode

Offset: 0

SubIndex: 0

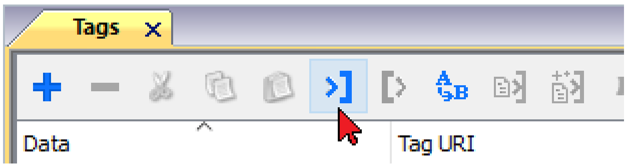
Data Type: boolean

Arraysize: 0

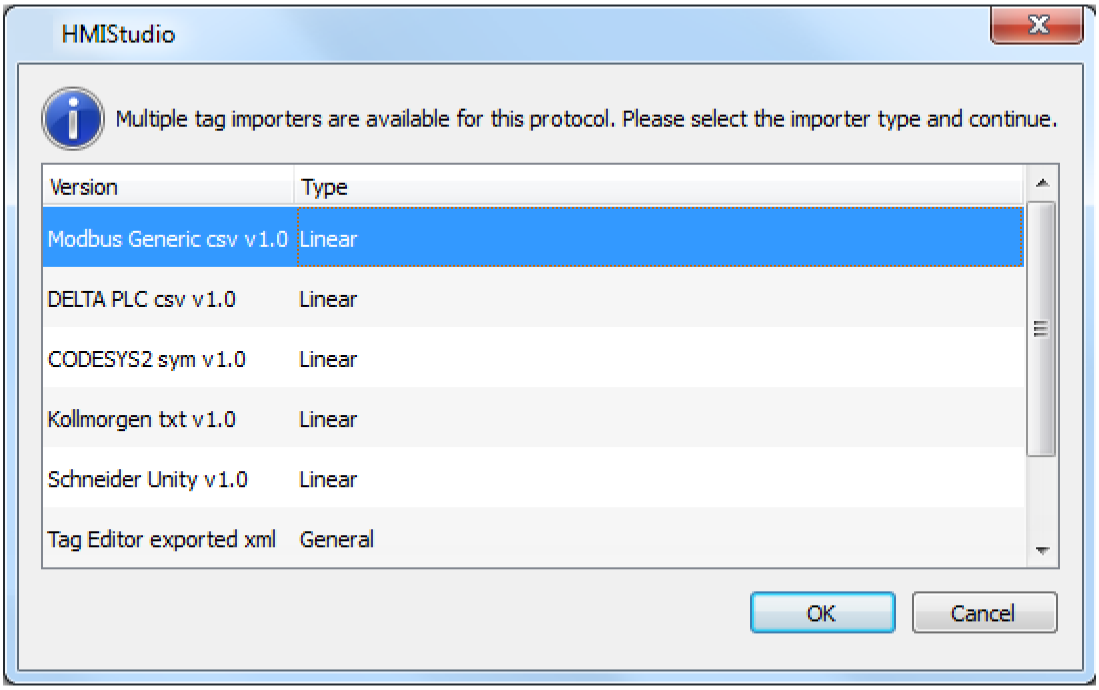
Conversion:

Tag Import

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



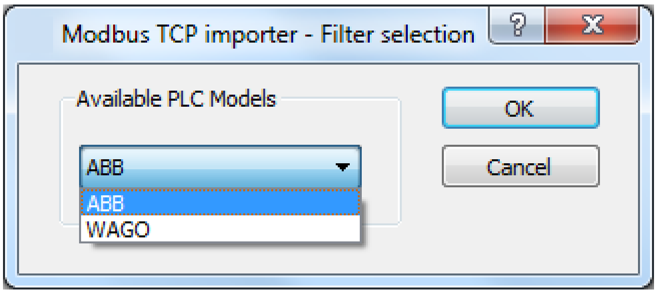
The following dialog shows which importer type can be selected.

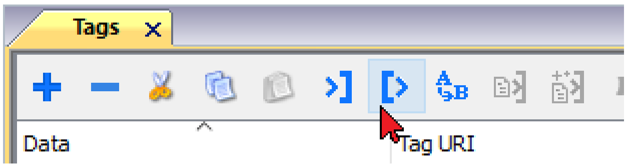


HMIStudio

Multiple tag importers are available for this protocol. Please select the importer type and continue.

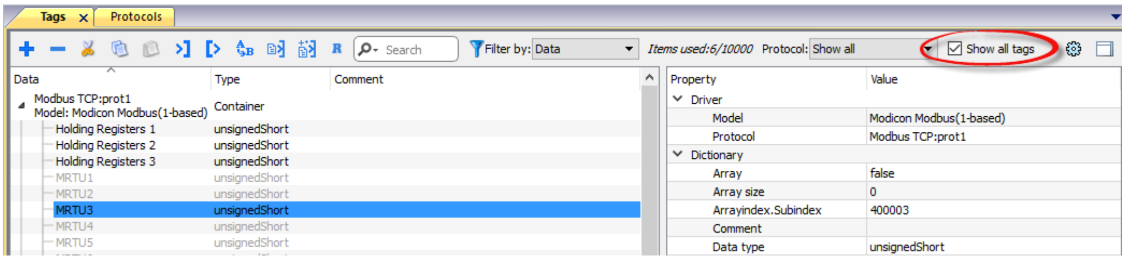
Version	Type
Modbus Generic csv v1.0	Linear
DELTA PLC csv v1.0	Linear
CODESYS2 sym v1.0	Linear
Kollmorgen txt v1.0	Linear
Schneider Unity v1.0	Linear
Tag Editor exported xml	General

Type	Description
Modbus Generic csv v1.0 **Linear**	Requires a **.csv** file. All variables will be displayed at the same level.
DELTA PLC csv v1.0	Requires a **.csv** file. All variables will be displayed at the same level.
CODESYS2 sym v1.0 **Linear**	Requires a **.sym** file. All variables will be displayed at the same level. After selecting the **.sym** file, the following dialog will appear for PLC model selection.  Modbus TCP importer - Filter selection Available PLC Models: ABB ABB WAGO OK Cancel
Kollmorgen txt v1.0 **Linear**	Requires a **.txt** file. All variables will be displayed at the same level.
Schneider Unity v1.0 **Linear**	Requires a **.uny** file.

Type	Description
	The file containing symbols must be exported in **.txt** format and later renamed as **.uny**. The importer considers only variables located at fixed address and disregards arrays of strings. All other arrays, except for boolean type, are expanded.
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**.

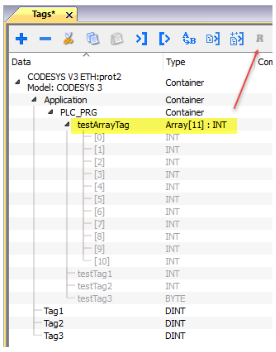
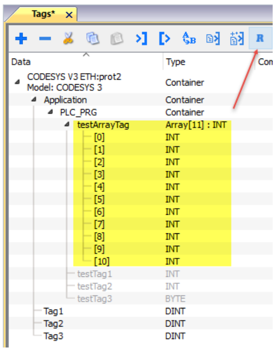
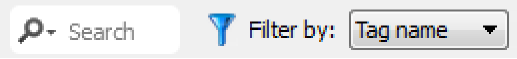
The tags available within the Dictionary but not imported into the project are gray and are visible only when the "Show all tags" check box is selected.



Data	Type	Comment
Modbus TCP:prot1 Model: Modicon Modbus(1-based)	Container	
Holding Registers 1	unsignedShort	
Holding Registers 2	unsignedShort	
Holding Registers 3	unsignedShort	
MRTU1	unsignedShort	
MRTU2	unsignedShort	
MRTU3	unsignedShort	
MRTU4	unsignedShort	
MRTU5	unsignedShort	

Property	Value
Driver	
Model	Modicon Modbus(1-based)
Protocol	Modbus TCP:prot1
Dictionary	
Array	false
Array size	0
Arrayindex.Subindex	400003
Comment	
Data type	unsignedShort

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

Toolbar item	Description
	 Tags* Data Type Con CODESYS V3 ETH:prot2 Model: CODESYS 3 Container Application Container PLC_PRG Container testArrayTag Array[11] : INT [0] INT [1] INT [2] INT [3] INT [4] INT [5] INT [6] INT [7] INT [8] INT [9] INT [10] INT testTag1 INT testTag2 INT testTag3 BYTE Tag1 DINT Tag2 DINT Tag3 DINT  Tags* Data Type Con CODESYS V3 ETH:prot2 Model: CODESYS 3 Container Application Container PLC_PRG Container testArrayTag Array[11] : INT [0] INT [1] INT [2] INT [3] INT [4] INT [5] INT [6] INT [7] INT [8] INT [9] INT [10] INT testTag1 INT testTag2 INT testTag3 BYTE Tag1 DINT Tag2 DINT Tag3 DINT
	Searches tags in the dictionary basing on filter combo-box item selected.

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
NodeID	Node the tag belongs to
TagName	Tag description
MemoryType	• OUTP • INP • IREG • HREG
Address	Offset compatible with Modbus notation
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,
```

ℹ Note: This line has no comment. When the Comment is missing, the comma as a terminator character is mandatory.

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 .	-
The received message too short	The device received a response with an invalid format from the controller .	-
Incorrect writing data acknowledge	The controller did not accept a write request.	Check if project data is consistent with the controller resources.

Modbus TCP Server

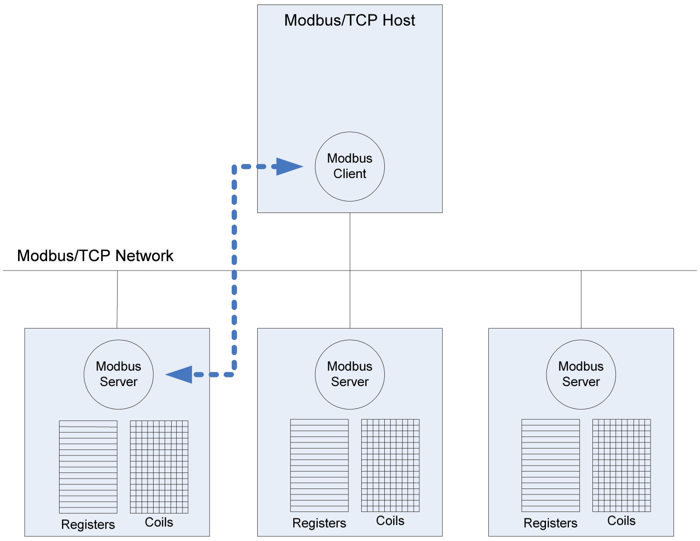
Modbus TCP Server communication driver allows connecting the HMI device as a server in a Modbus TCP network. It is possible for Modbus TCP clients to connect then to multiple HMI panels acting as servers. Standard Modbus TCP messages are used for information exchange.

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	**Illegal Function**. the function code received in the query is not supported
02	**Illegal Data Address**. Data Address received in the query exceeds the predefined data range (see **Tag Editor Settings** for detailed ranges of all types).
03	**Illegal Data Value**. A sub function other than 00 is specified in Loopback Diagnostic Test (Code 08).

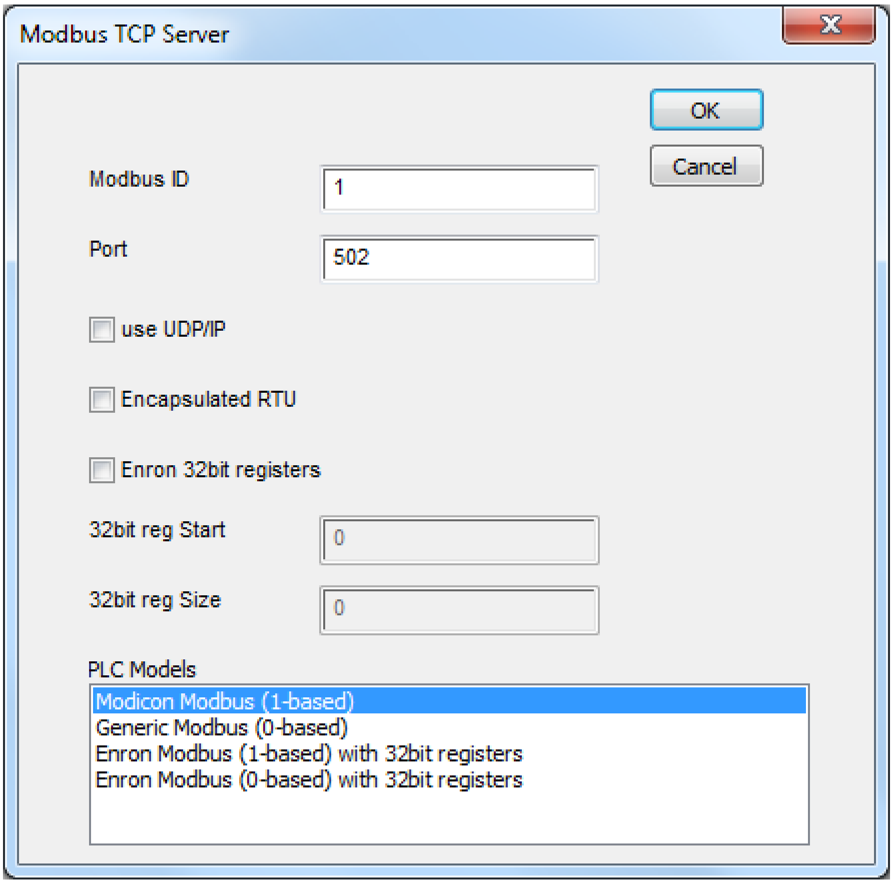
Protocol Editor Settings

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

Modbus ID: 1

Port: 502

□ use UDP/IP

□ Encapsulated RTU

□ Enron 32bit registers

32bit reg Start: 0

32bit reg Size: 0

PLC Models
- Modicon Modbus (1-based)
- Generic Modbus (0-based)
- Enron Modbus (1-based) with 32bit registers
- Enron Modbus (0-based) with 32bit registers

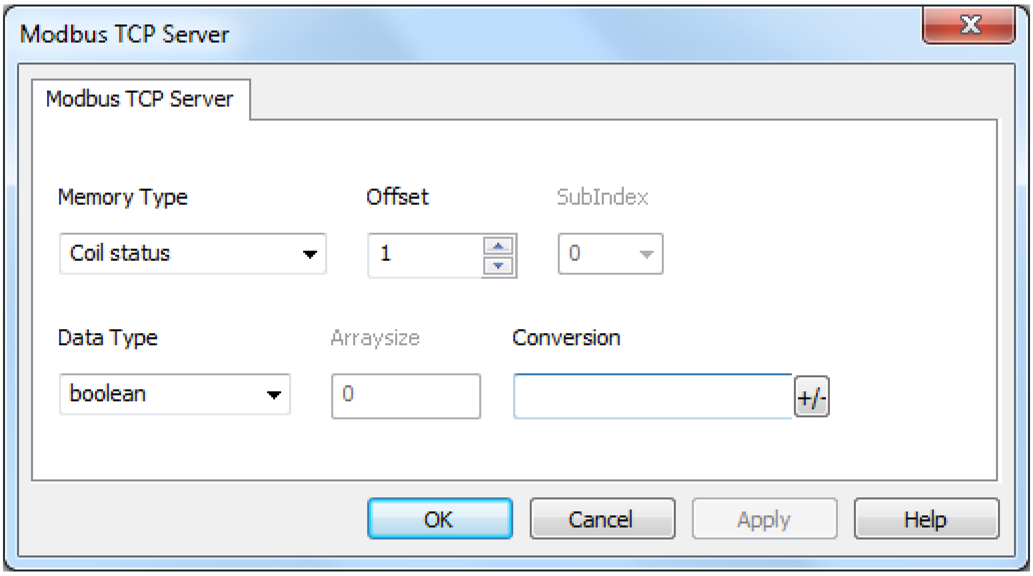
Element	Description
Modbus ID	Modbus node ID of the HMI device. Every Modbus server device in the network must have its own Modbus ID.
Port	Port number used by the Modbus TCP protocol. Default value is **502**. Set the value accordingly to the port number used by your Modbus TCP Network.
use UDP/IP	If selected, the protocol will use connectionless UDP datagrams.
Encapsulated RTU	If selected, the protocol will use serial RTU protocol over Ethernet instead of Modbus TCP protocol, independently from TCP or UDP usage.
Enron 32bit registers	If selected, allows to define the first register address and the number of registers for 32 bit registers memory area. Note: 32 bit registers are available only for **Enron Modbus** PLC Models.

Element	Description
32bit reg Start **32bit reg Size**	32 bit registries memory area definition. **Start** value represents the first register address. **Size** value represents the number of registries. Note: A request to one of the registries inside this area gives a 4 byte answer.
PLC Models	Allows to select between different PLC models: • **Modicon Modbus (1-based)**: Modbus implementation where all resources starts with offset 1. • **Generic Modbus (0-based)**: Modbus implementation where all resources starts with offset 0. • **Enron Modbus (1-based)**: Extends Modicon Mobdus implementation with 32 bit registers memory area. • **Enron Modbus (0-base)**: Extends Generic Modbus implementation with 32 bit registers memory area. 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.

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

Memory Type: Coil status

Offset: 1

SubIndex: 0

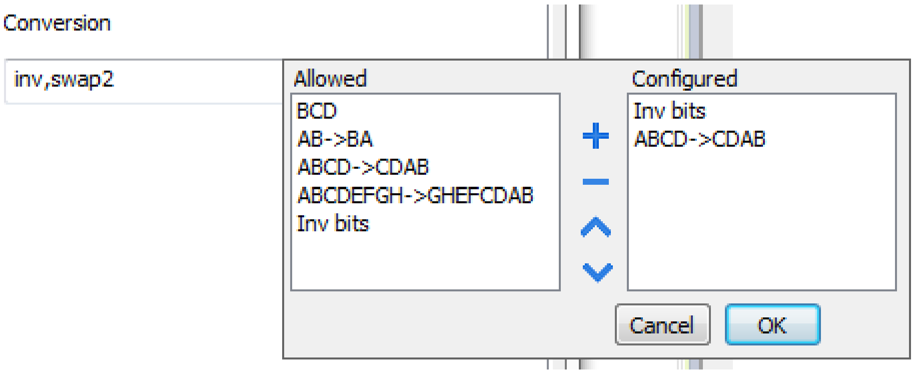
Data Type: boolean

Arraysize: 0

Conversion:

<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td><b>Memory Type</b></td><td>Modbus resource where tag is located.
<table>
<tr><th>Memory Type</th><th>Modbus Resource</th></tr>
<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>Modicon Mode</b></td><td>protocol parameter (see <b>Special Data Types</b> for mode details)</td></tr>
</table></td></tr>
<tr><td><b>Offset</b></td><td>Offset address where tag is located.
Offset addresses are six digits composed by one digit data type prefix + five digits resource address.
<table>
<tr><th>Memory Type</th><th>Studio
Offset range</th><th>Modicon
Offset range</th><th>Generic Modbus Offset range</th></tr>
<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>
</table></td></tr>
<tr><td><b>SubIndex</b></td><td>This allows resource offset selection within the register.</td></tr>
<tr><td><b>Data type</b></td><td>
<table>
<tr><th>Data Type</th><th>Memory Space</th><th>Limits</th></tr>
<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>
</table></td></tr>
</table>

<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>Data Type</th><th>Memory Space</th><th>Limits</th></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>
</table>

Note: to define arrays. select one of Data Type format followed by square brackets like “byte[]”, “short[]”…
</td></tr>
<tr><td><b>Arraysize</b></td><td>
<ul><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>
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.
</td></tr>
<tr><td><b>Conversi on</b></td><td>
Conversion to be applied to the tag.

Conversion: inv,swap2
Allowed: BCD, AB->BA, ABCD->CDAB, ABCDEFGH->GHEFCDAB, Inv bits
Configured: Inv bits, ABCD->CDAB

Depending on data type selected, the list <b>Allowed</b> shows one or more conversion types.
</td></tr>
</table>

<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>Value</th><th>Description</th></tr>
<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 -> 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 -> 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 -> 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 -> OPM...DAB</b></td><td><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)</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>
</table>

Select conversion and click +. The selected item will be added to list <b>Configured</b>.</td></tr>
</table>

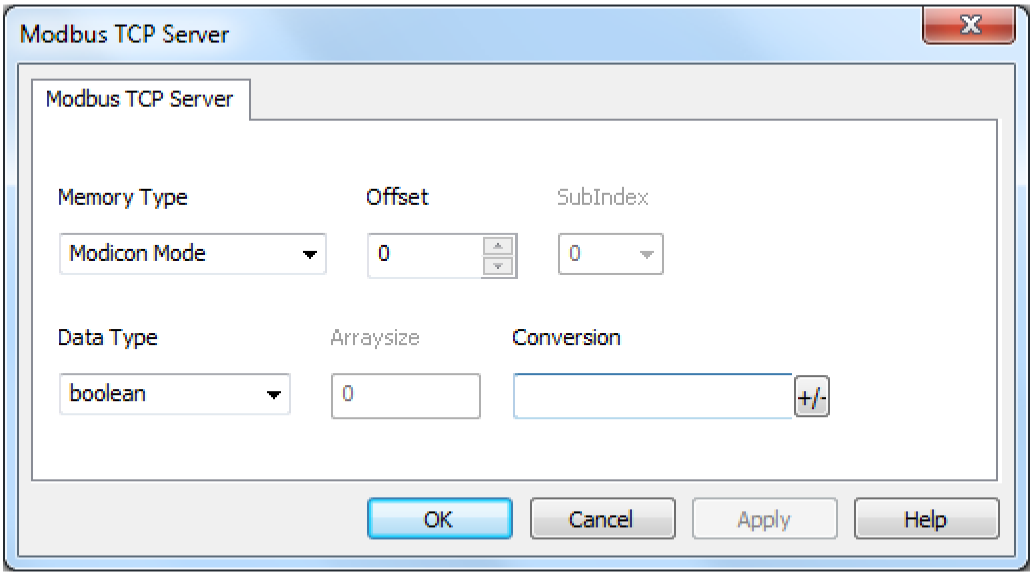
Element	Description
	If more conversions are configured, they will be applied in order (from top to bottom of list **Configured**). Use the arrow buttons to order the configured conversions.

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.

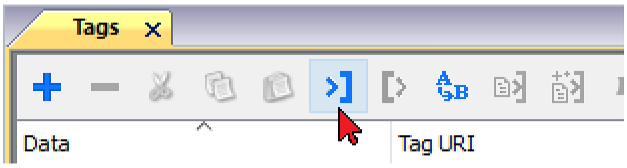
Note: Modicon Mode parameter value assigned at runtime is retained through power cycles.


Modbus TCP Server

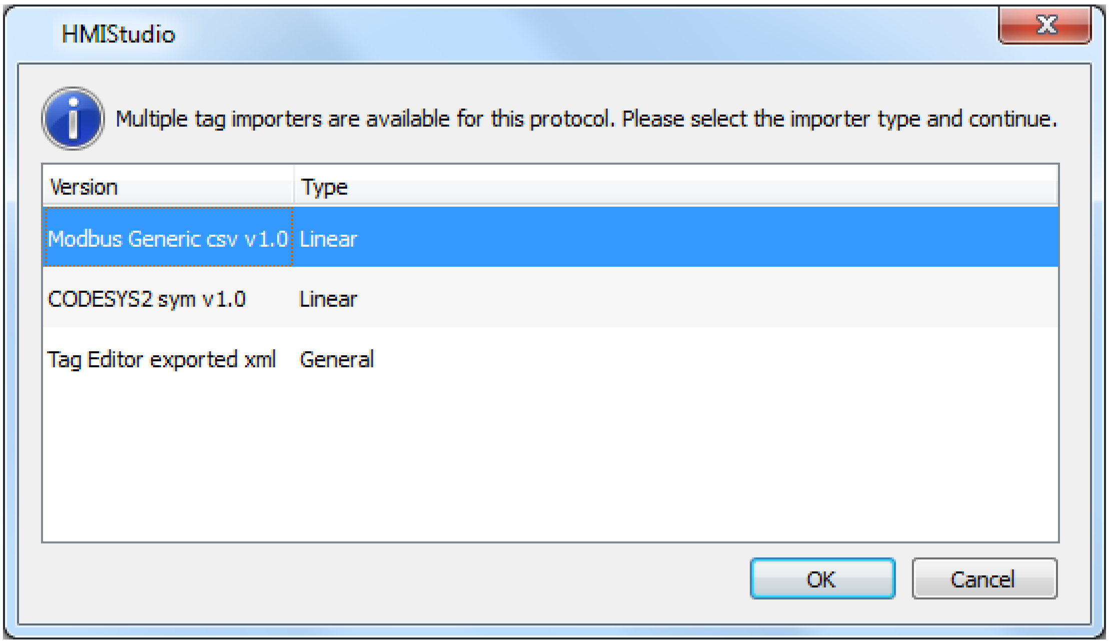
Memory Type: Modicon Mode
Offset: 0
SubIndex: 0
Data Type: boolean
Arraysize: 0
Conversion:

Tag Import

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



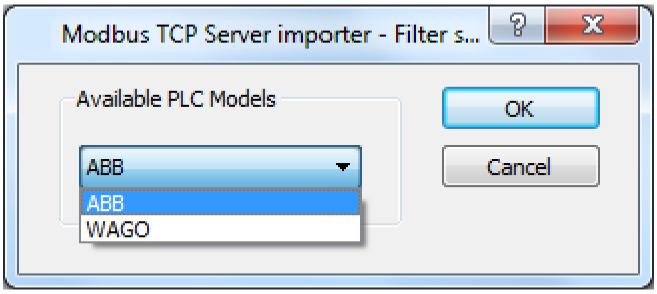
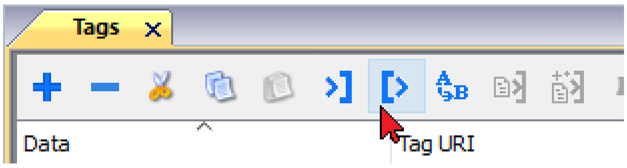
The following dialog shows which importer type can be selected.



HMIStudio

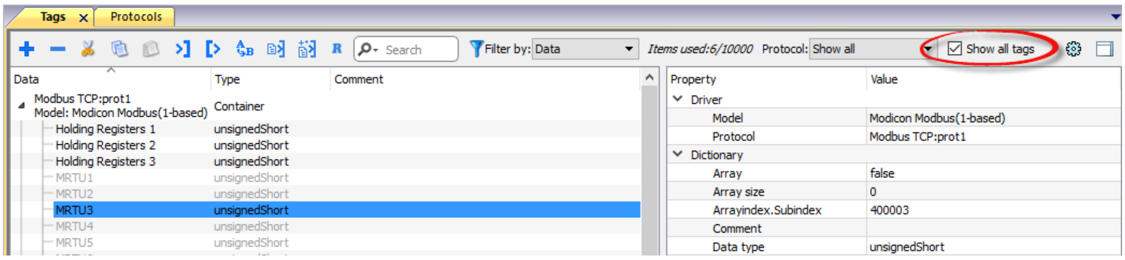
Multiple tag importers are available for this protocol. Please select the importer type and continue.

Version	Type
Modbus Generic csv v1.0	Linear
CODESYS2 sym v1.0	Linear
Tag Editor exported xml	General

Importer	Description
Modbus Generic csv v1.0 **Linear**	Requires a **.csv** file. All variables will be displayed at the same level.
CODESYS2 sym v1.0 **Linear**	Requires a **.sym** file. All variables will be displayed at the same level. After selecting the **.sym** file, the following dialog will appear for PLC model selection.  Modbus TCP Server importer - Filter s... Available PLC Models: ABB ABB WAGO OK Cancel
Tag Editor exported xml	Select this importer to read a generic XML file exported from Tag Editor by appropriate button. 

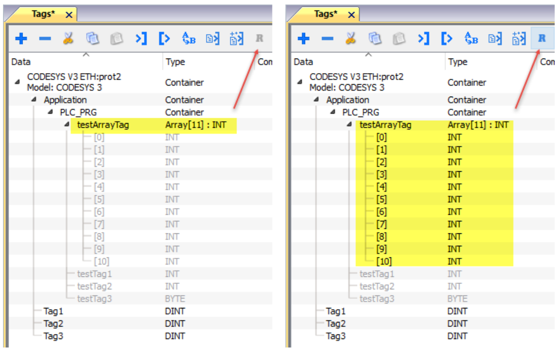
Once the importer has been selected, locate the symbol file and click **Open**.

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.



Data	Type	Comment
Modbus TCP:prot1 Model: Modicon Modbus(1-based)	Container	
Holding Registers 1	unsignedShort	
Holding Registers 2	unsignedShort	
Holding Registers 3	unsignedShort	
MRTU1	unsignedShort	
MRTU2	unsignedShort	
MRTU3	unsignedShort	
MRTU4	unsignedShort	
MRTU5	unsignedShort	

Property	Value
Driver	
Model	Modicon Modbus(1-based)
Protocol	Modbus TCP:prot1
Dictionary	
Array	false
Array size	0
Arrayindex.Subindex	400003
Comment	
Data type	unsignedShort

Toolbar item	Description
	Import Tag(s). Select tags to be imported and click on this icon to add tags from tag dictionary to the project
	Update Tag(s). Click on this icon to update the tags in the project, due a new dictionary import.
R	Check this box to import all sub-elements of a tag. Example of both checked and unchecked result:  Tags* Data / Type / Com CODESYS V3 ETH:prot2 Model: CODESYS 3 / Container Application / Container PLC_PRG / Container testArrayTag / Array[11] : INT [0] / INT [1] / INT [2] / INT [3] / INT [4] / INT [5] / INT [6] / INT [7] / INT [8] / INT [9] / INT [10] / INT testTag1 / INT testTag2 / INT testTag3 / BYTE Tag1 / DINT Tag2 / DINT Tag3 / DINT
Search Filter by: Tag name	Searches tags in the dictionary basing on filter combo-box item selected.

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
NodeID	Node the tag belongs to
TagName	Tag description
MemoryType	• OUTP • INP • IREG • HREG
Address	Offset compatible with Modbus notation
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.

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.

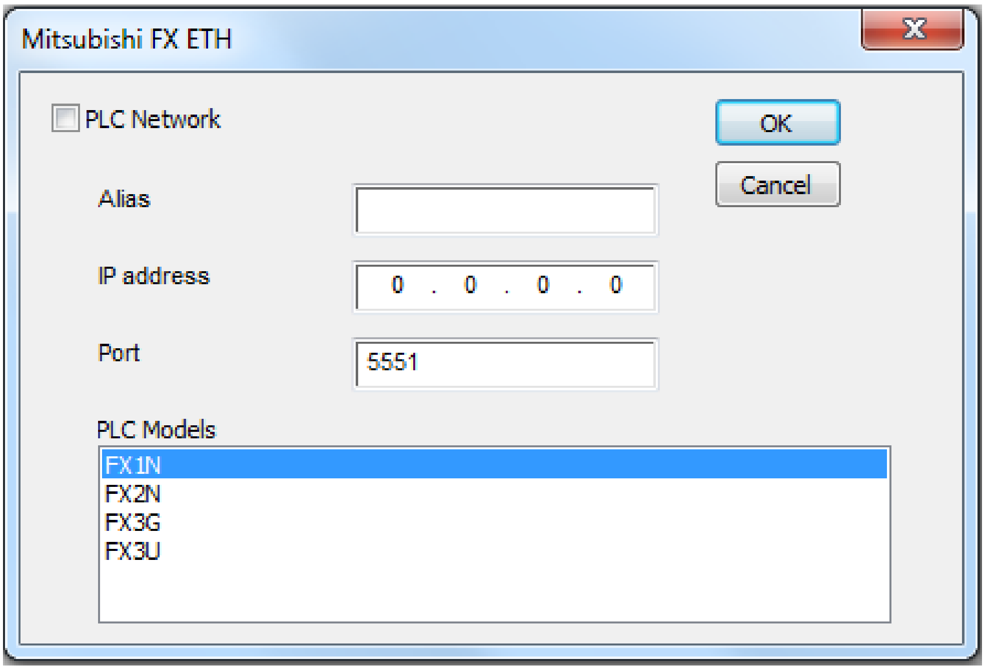
Mitsubishi FX ETH

Mitsubishi FX ETH implements the MELSEC-F (or MC) communication protocol that can be used with FX CPUs as described in the Mitsubishi document “FX3U-ENET USER’S MANUAL”, chapter 8 “Communication using MC protocol”.

Note: Mitsubishi FX3U controller must be equipped with the appropriate Ethernet module: FX3U-ENET

Protocol Editor Settings

Add [+] a driver in the Protocol editor and select the protocol called “Mitsubishi FX ETH” from the list of available protocols.


Mitsubishi FX ETH

□ PLC Network

Alias

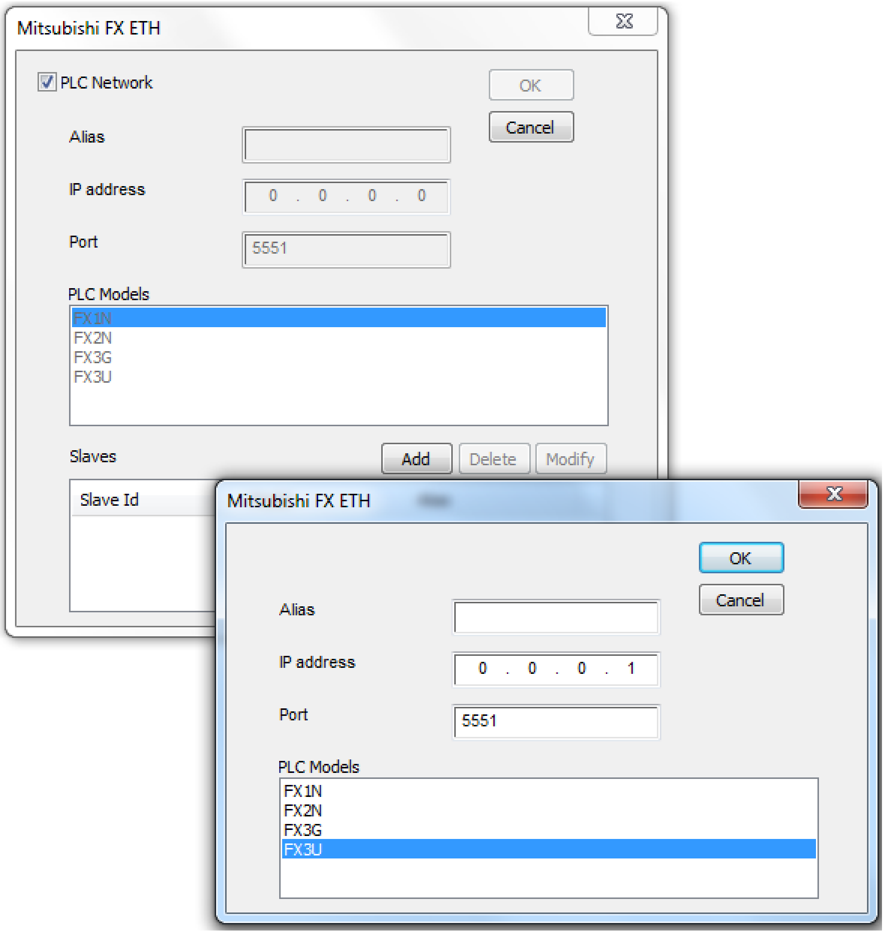
IP address 0 . 0 . 0 . 0

Port 5551

PLC Models

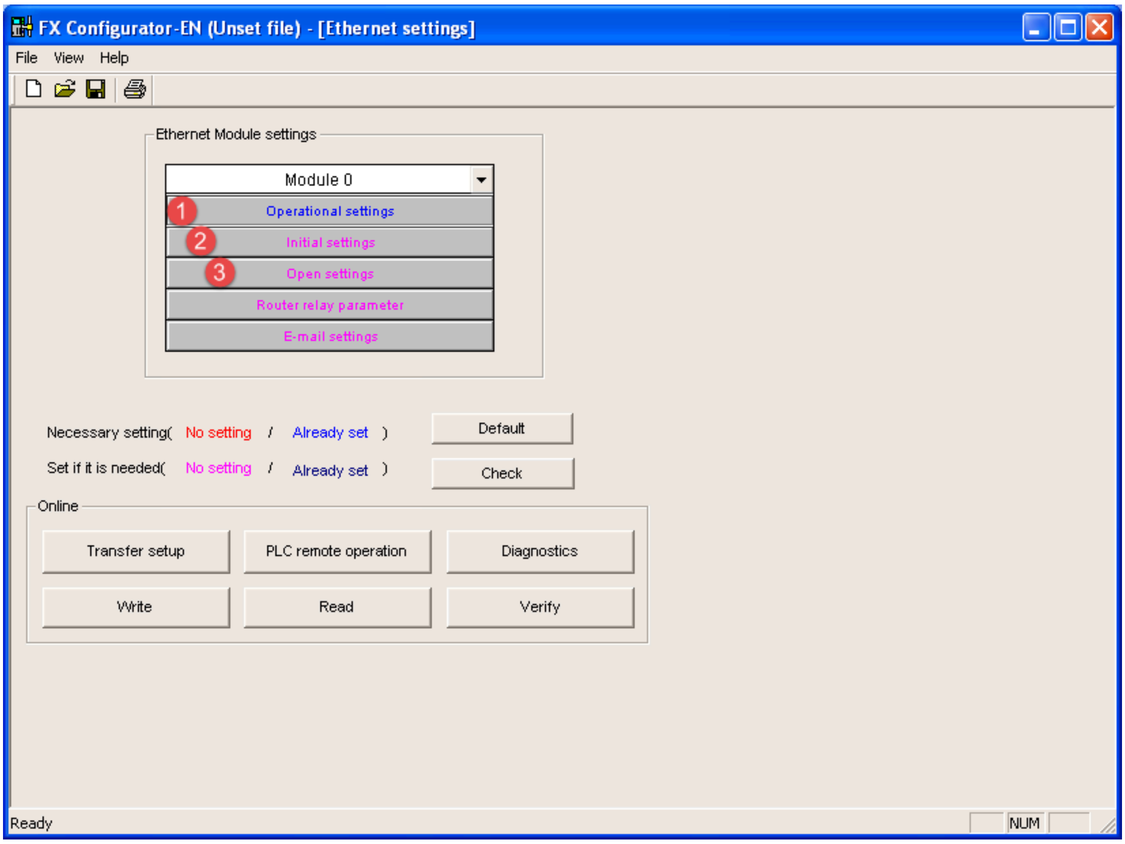
FX1N
FX2N
FX3G
FX3U

Element	Description
IP address	Ethernet IP address of the controller
Port	Specifies the port number (decimal) used in the communication with the PLC.

Element	Description
PLC Model	Defines the PLC model connected
PLC Network	The protocol allows the connection of multiple controllers to one operator panel. To set-up multiple connections, check “PLC network” checkbox and enter IP Address for all controllers.  Mitsubishi FX ETH ☑ PLC Network Alias IP address 0 . 0 . 0 . 0 Port 5551 PLC Models: FX1N (selected), FX2N, FX3G, FX3U Slaves Slave Id Mitsubishi FX ETH Alias IP address 0 . 0 . 0 . 1 Port 5551 PLC Models: FX1N, FX2N, FX3G, FX3U (selected)

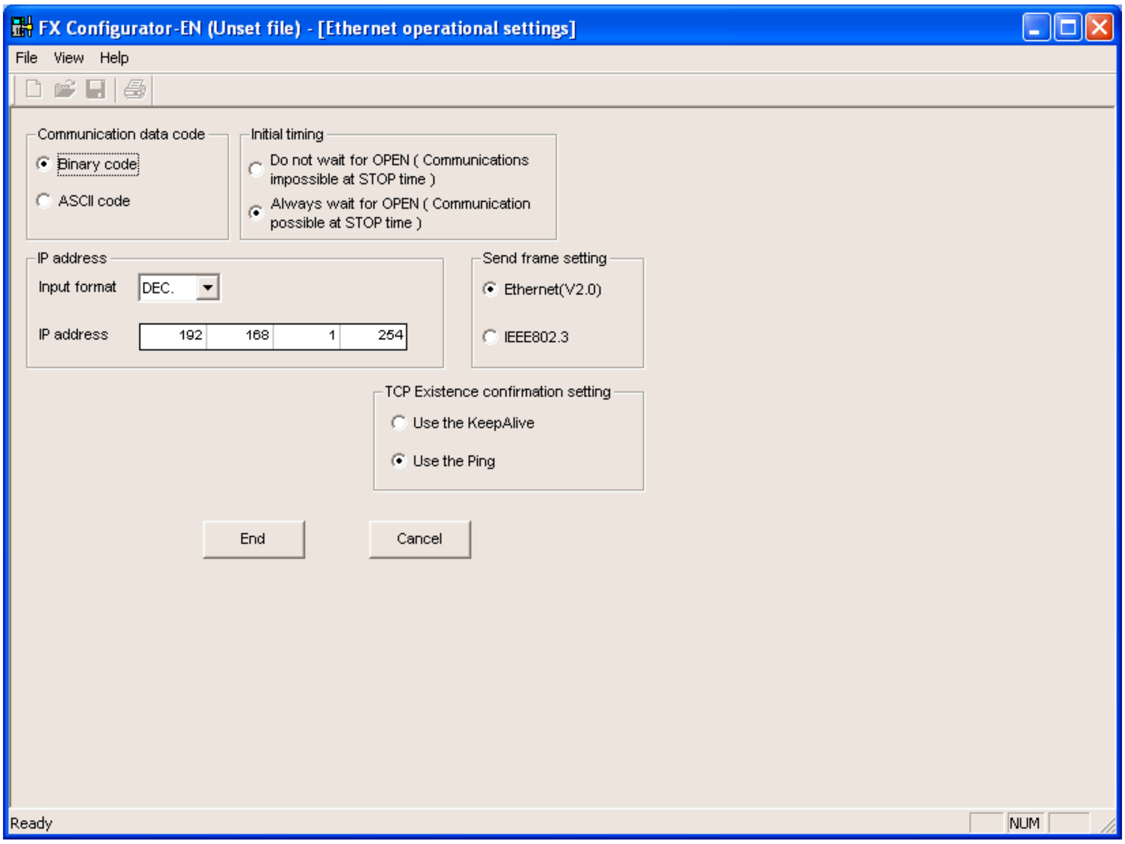
Controller Settings with GX Developer

The Mitsubishi FX system must be properly configured for Ethernet communication using the Mitsubishi FX Configurator. Click on “Operational settings” as shown at point (1) in the following figure:



Into Operational Settings dialog, verify the “Communication data code” is set to “Binary code”,

Then type-in the Controller IP Address and confirm with [End] button.


FX Configurator-EN (Unset file) - [Ethernet operational settings]
File View Help
End
Cancel
Ready
NUM

Communication data code
● Binary code
○ ASCII code

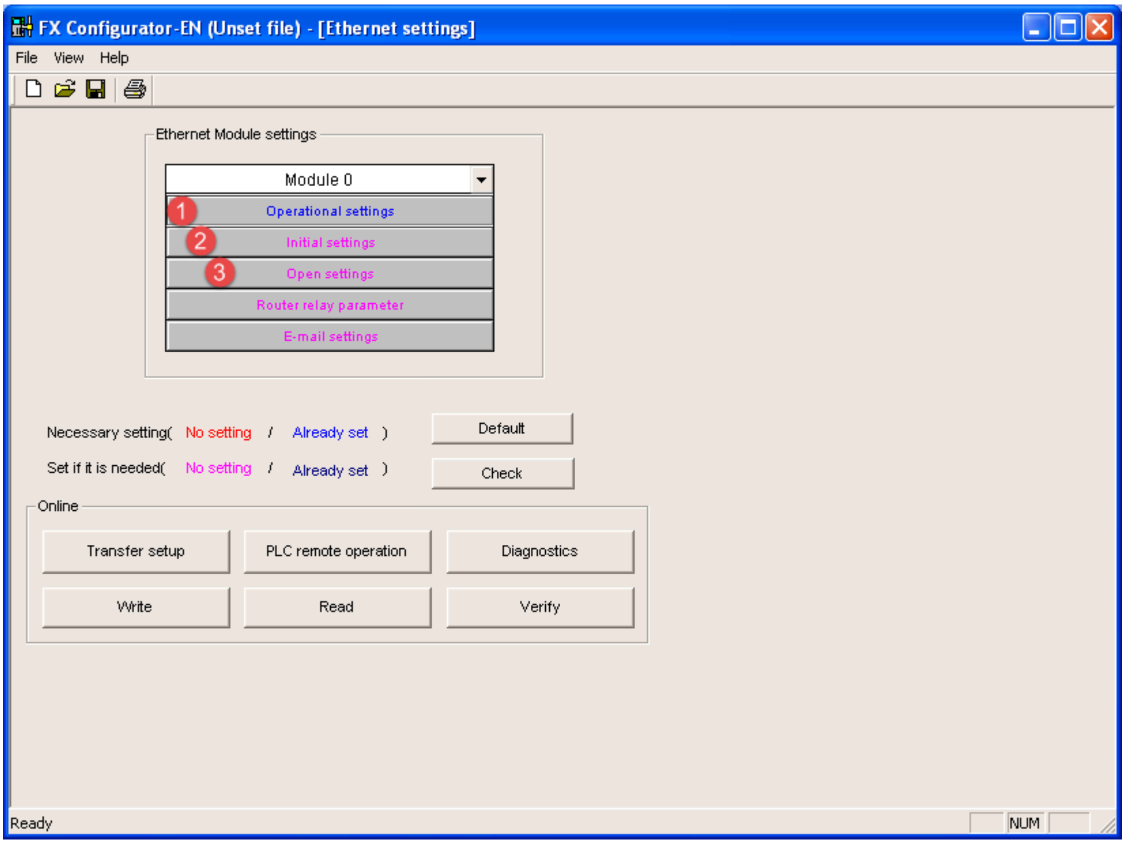
Initial timing
○ Do not wait for OPEN (Communications impossible at STOP time)
● Always wait for OPEN (Communication possible at STOP time)

IP address
Input format DEC.
IP address 192 168 1 254

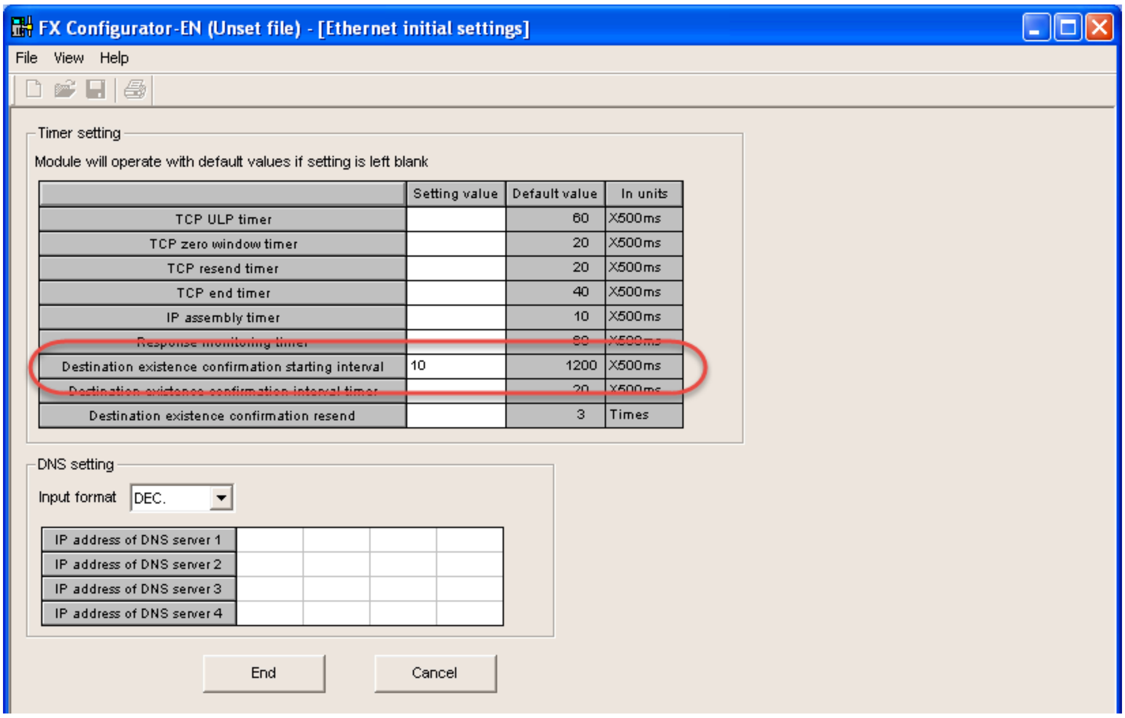
Send frame setting
● Ethernet(V2.0)
○ IEEE802.3

TCP Existence confirmation setting
○ Use the KeepAlive
● Use the Ping

Click now on “Initial settings” as shown at point (2) of Figure below:



For proper communication between HMI and controller it is required to change “Destination existence confirmation starting interval” from the default value of 1200 to 10ms.


FX Configurator-EN (Unset file) - [Ethernet initial settings]
File View Help

Timer setting

Module will operate with default values if setting is left blank

	Setting value	Default value	In units
TCP ULP timer		60	X500ms
TCP zero window timer		20	X500ms
TCP resend timer		20	X500ms
TCP end timer		40	X500ms
IP assembly timer		10	X500ms
Response monitoring timer		60	X500ms
Destination existence confirmation starting interval	10	1200	X500ms
Destination existence confirmation interval timer		20	X500ms
Destination existence confirmation resend		3	Times

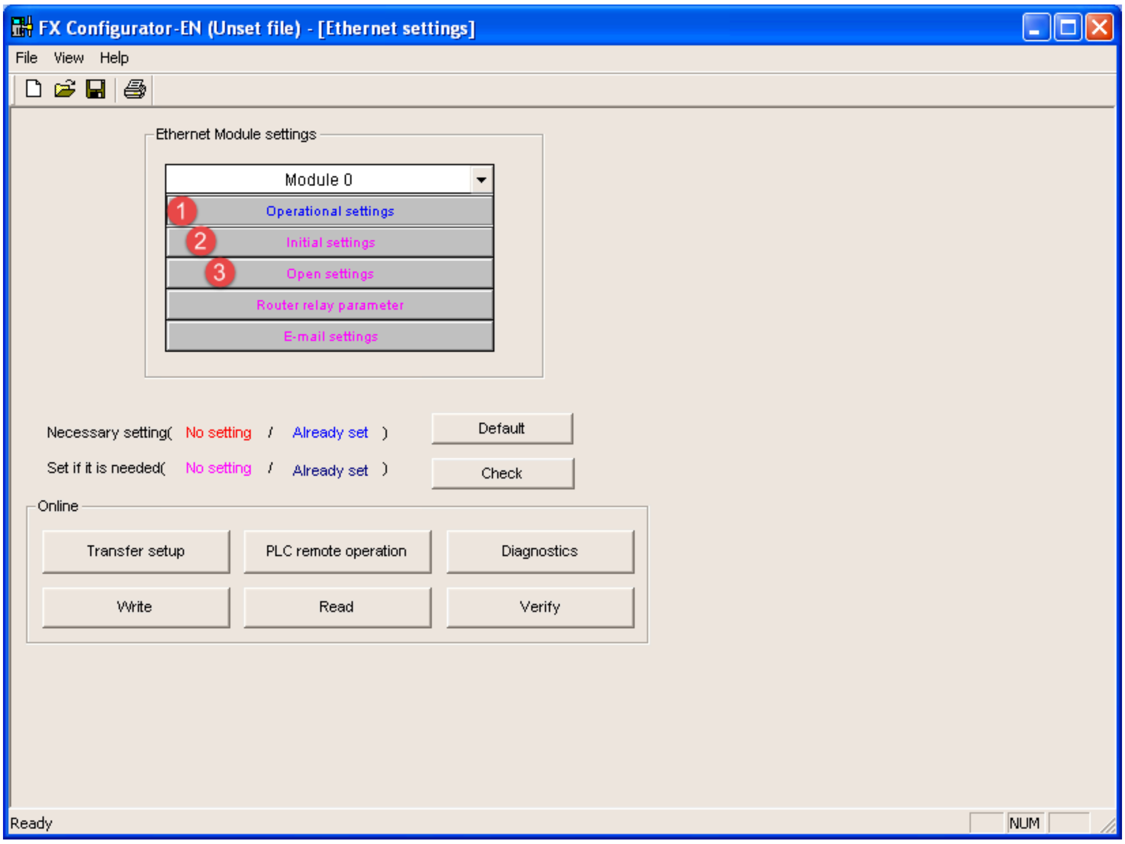
DNS setting

Input format DEC.

IP address of DNS server 1				
IP address of DNS server 2				
IP address of DNS server 3				
IP address of DNS server 4				

In case of communication error, this avoid controller keeps alive the connection for a too long time before to allow a new connection from the HMI.

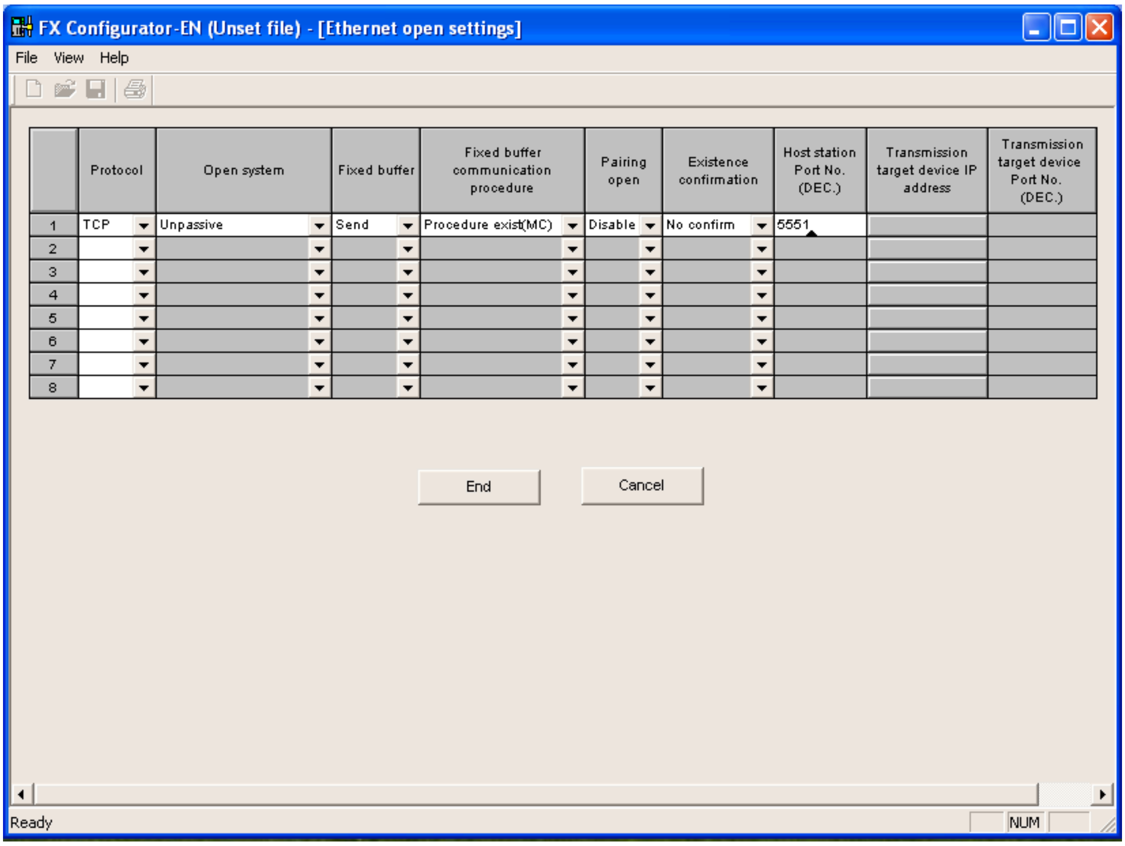
Click now on “Open settings” as shown at point (3) of Figure below



The next figure shows the “Ethernet open settings” configuration.

The detailed explanation of the meaning of each setting is available in Chapter 5.5 of the Mitsubishi “FX3U-ENET USER’S MANUAL”.

“Host station Port No.” defined here is the same must be used into Protocol Editor Settings chapter.

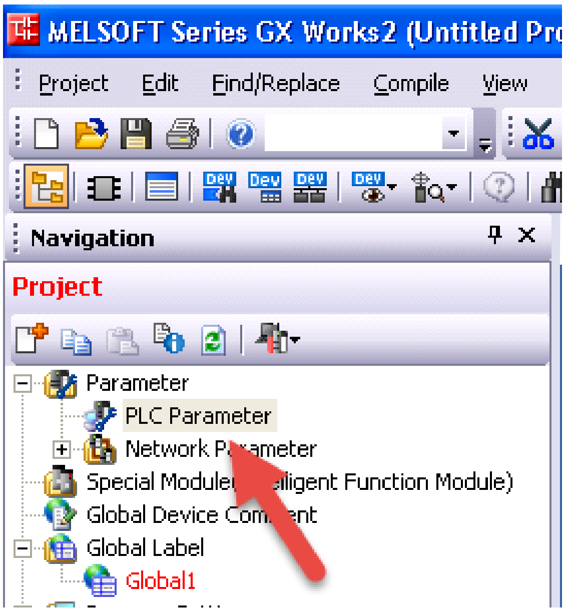


	Protocol	Open system	Fixed buffer	Fixed buffer communication procedure	Pairing open	Existence confirmation	Host station Port No. (DEC.)	Transmission target device IP address	Transmission target device Port No. (DEC.)
1	TCP	Unpassive	Send	Procedure exist(MC)	Disable	No confirm	5551		
2									
3									
4									
5									
6									
7									
8									

Note: the usage of more than one panel communicating with the same controller requires to define proper settings in the “Open settings” configuration dialog: one connection per each panel must be configured with proper properties

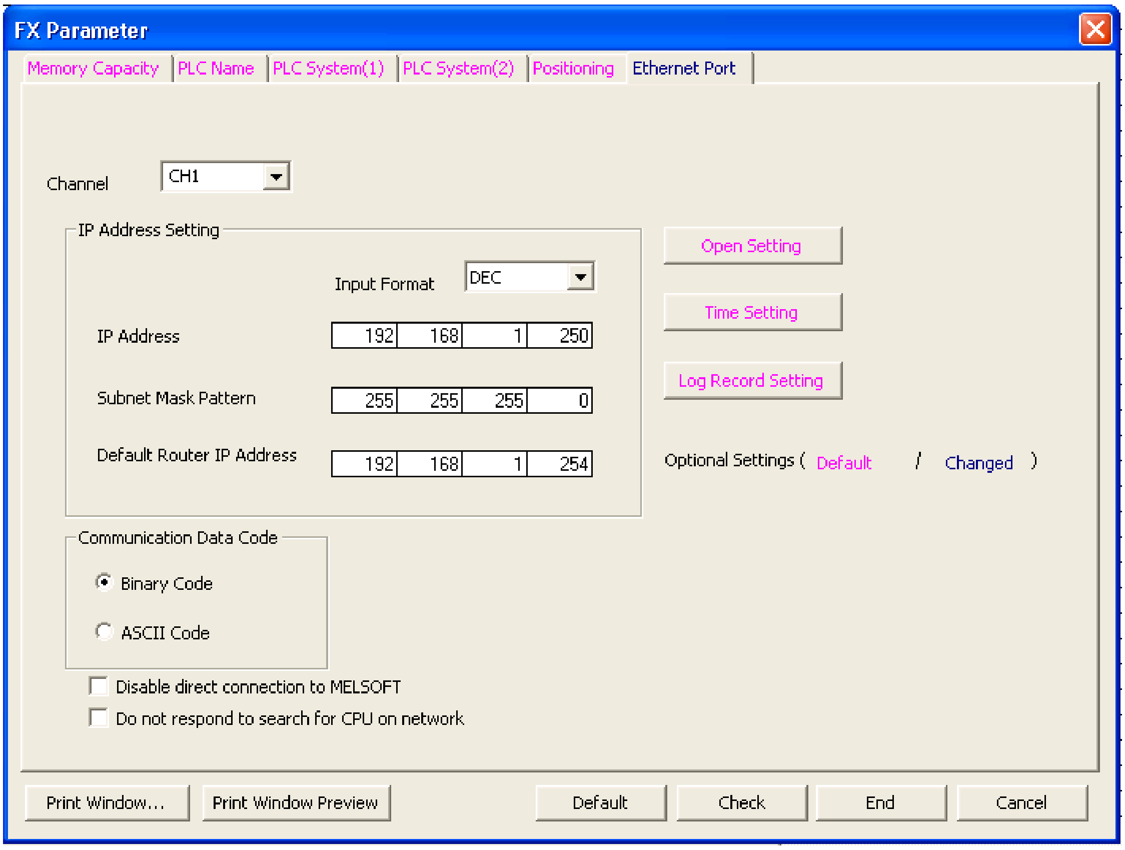
Controller Settings with GX Works2

The Mitsubishi FX system must be properly configured for Ethernet communication inside GX Works2 programming suite. FX Parameter dialog can be recalled with double-click on PLC Parameter:



Then select “Ethernet Port” tab where is possible to configure IP Address.

Verify the “Communication data code” is set to “Binary code” as shown below:



Channel CH1

IP Address Setting

Input Format DEC

IP Address 192 168 1 250

Subnet Mask Pattern 255 255 255 0

Default Router IP Address 192 168 1 254

Communication Data Code

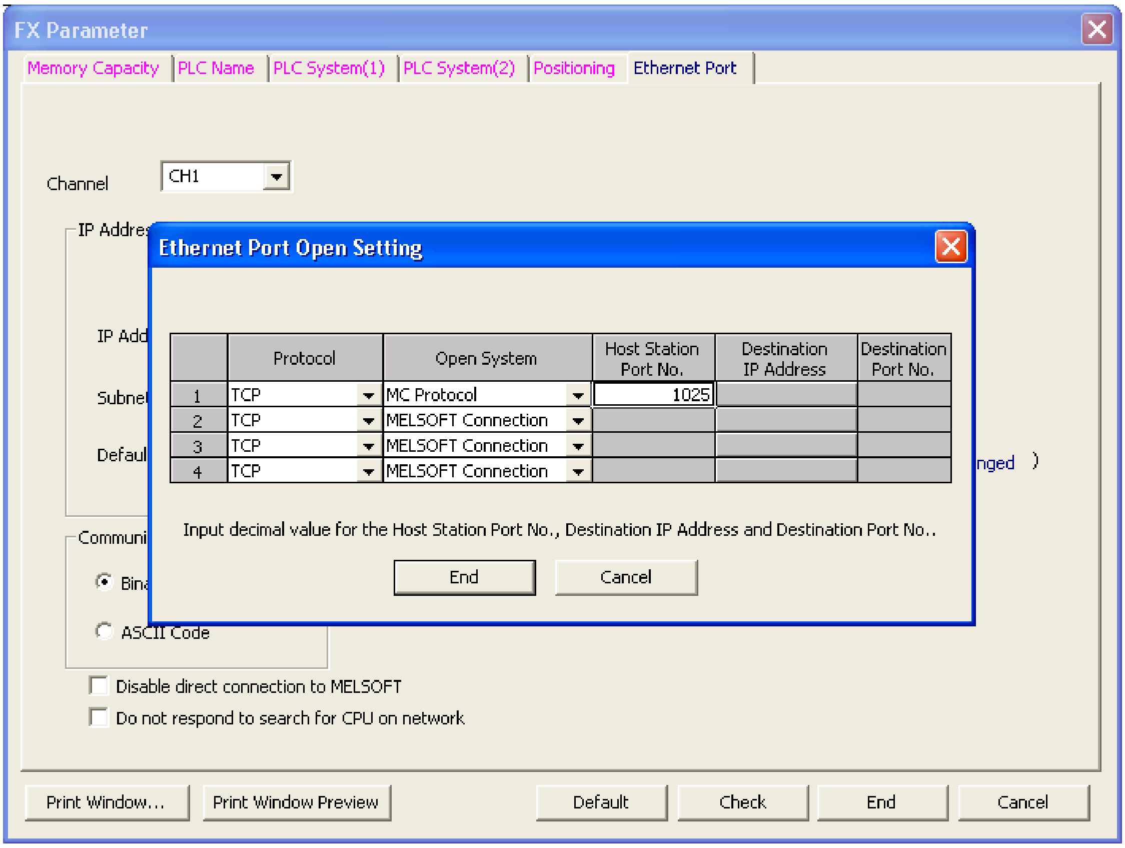
● Binary Code

○ ASCII Code

□ Disable direct connection to MELSOFT

□ Do not respond to search for CPU on network

Then click on “Open Settings” button to recall the “Ethernet Port Open Setting” dialog.



Channel CH1

Ethernet Port Open Setting

	Protocol	Open System	Host Station Port No.	Destination IP Address	Destination Port No.
1	TCP	MC Protocol	1025		
2	TCP	MELSOFT Connection			
3	TCP	MELSOFT Connection			
4	TCP	MELSOFT Connection			

Input decimal value for the Host Station Port No., Destination IP Address and Destination Port No..

● Bina

○ ASCII Code

□ Disable direct connection to MELSOFT

□ Do not respond to search for CPU on network

“Host station Port No.” defined here is the same must be used into Protocol Editor Settings chapter.

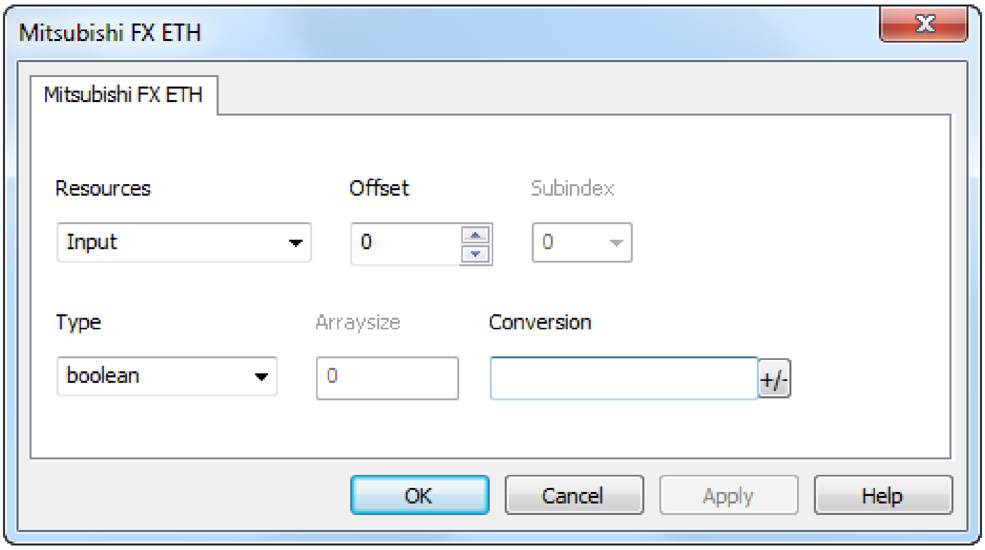
Note: For FX3GE Controller, the Open System must be set as “Data Monitor” and Port set to 1025.

Note: the usage of more than one panel communicating with the same controller requires to define proper settings in the “Open settings” configuration dialog: one connection per each panel must be configured with proper properties.

Tag Editor Settings

Into Tag editor select the protocol “Mitsubishi FX ETH” from the list of defined protocols and add a tag using [+] button.

Tag settings can be defined using the following dialog:



Resources: Input

Offset: 0

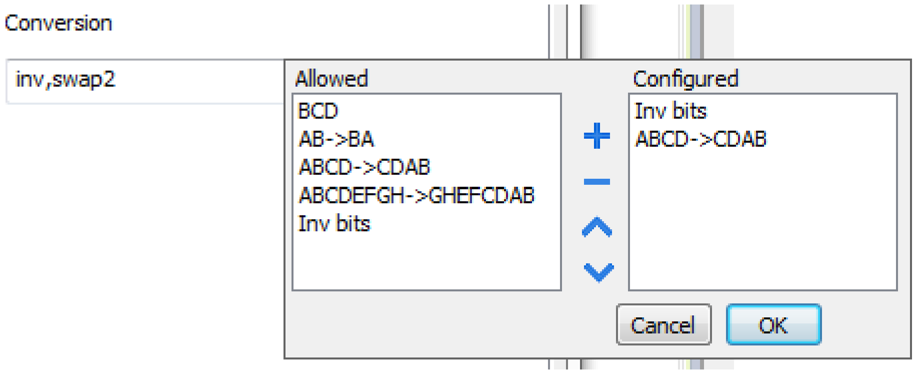
Subindex: 0

Type: boolean

Arraysize: 0

Conversion:

<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td><b>Resources</b></td><td>Area of PLC where tag is located</td></tr>
<tr><td><b>Offset</b></td><td>Offset address where tag is located.</td></tr>
<tr><td><b>SubIndex</b></td><td>This allows resource offset selection within the register.</td></tr>
<tr><td><b>Type</b></td><td>
<table>
<tr><th>Data Type</th><th>Memory Space</th><th>Limits</th></tr>
<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>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>float</b></td><td>IEEE single-precision
32-bit floating point type</td><td>1.17e-38 ... 3.40e38</td></tr>
<tr><td><b>string</b></td><td colspan="2">Refer to “String data type chapter”</td></tr>
</table>
Note: to define arrays, select one of Data Type format followed by square brackets like “byte[]”, “short[]”…
</td></tr>
<tr><td><b>Arraysize</b></td><td>
<ul>
<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>
</td></tr>
</table>

<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td></td><td>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.</td></tr>
<tr><td><b>Conversion</b></td><td>Conversion to be applied to the tag.

Conversion: inv,swap2
Allowed: BCD, AB->BA, ABCD->CDAB, ABCDEFGH->GHEFCDAB, Inv bits
Configured: Inv bits, ABCD->CDAB

Depending on data type selected, the list <b>Allowed</b> shows one or more conversion types.
<table>
<tr><th>Value</th><th>Description</th></tr>
<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 -> 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 -> 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 -> 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>
</table>
</td></tr>
</table>

<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td><b>ABC...NOP -> OPM...DAB</b></td><td><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)</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>
</table>
Select conversion and click <b>+</b>. 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.
</td></tr>
</table>

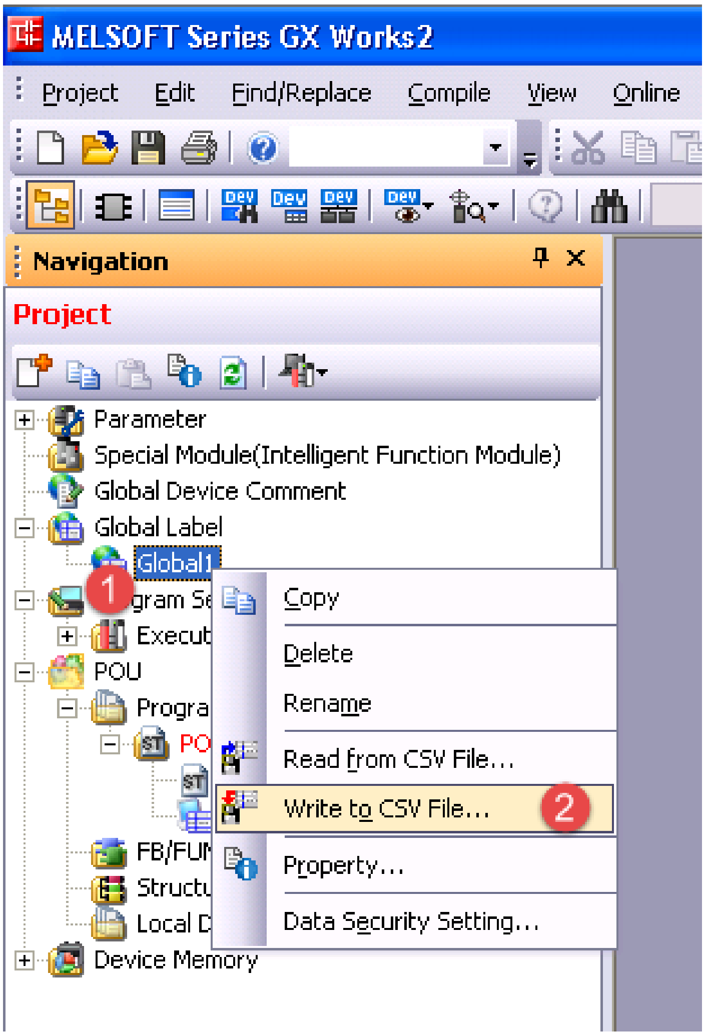
Tag Import

Exporting Tags from PLC

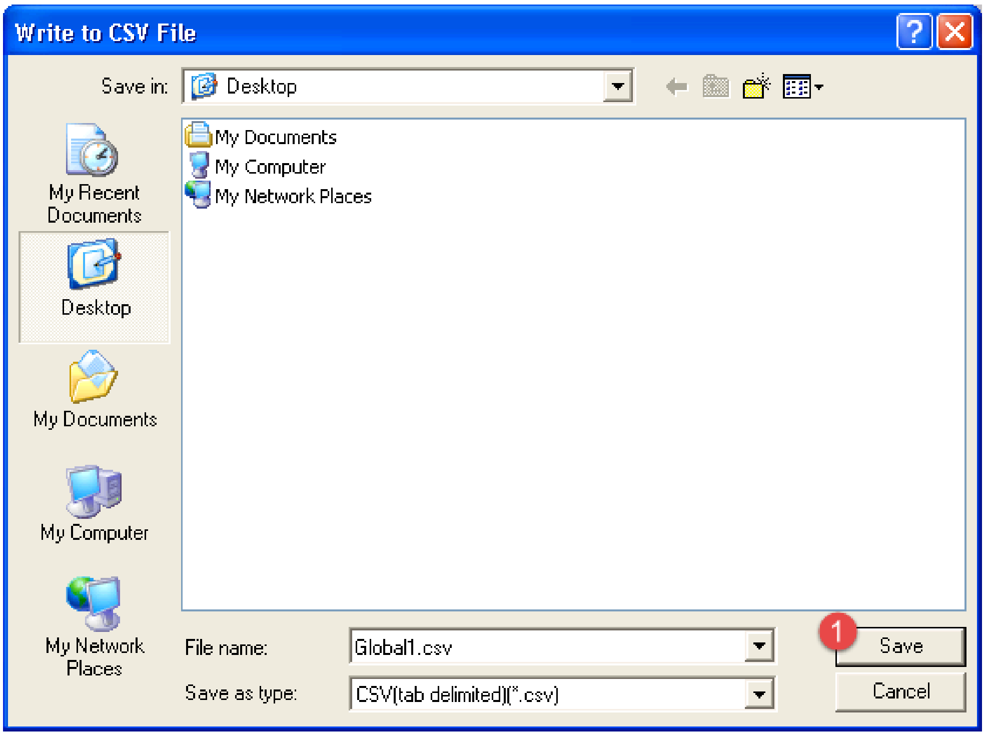
The Mitsubishi FX Ethernet tag import accepts symbol files with extension “csv” created by the Mitsubishi GX Works2 (Not from GX Developer).

The “.csv” file can be exported from the Project tree, as shown in the following figure.

1. Right-click on the Global variable list that need to be exported,
2. Select “Write to CSV File…”

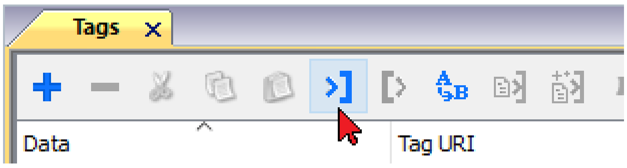


Into following dialog select the file name and location:

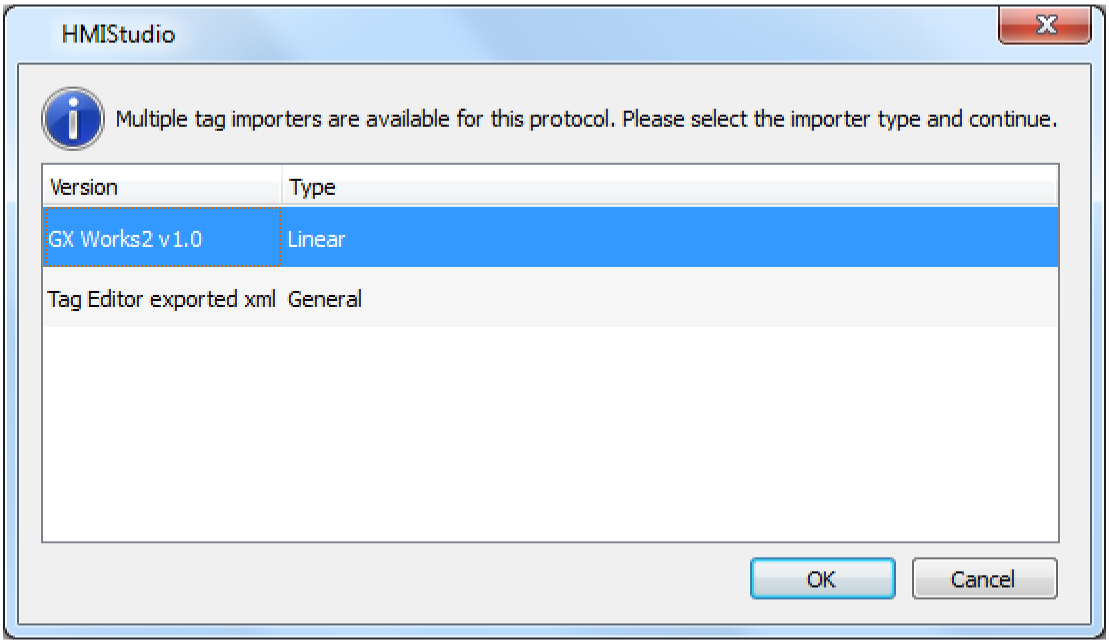


Importing Tags in Tag Editor

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

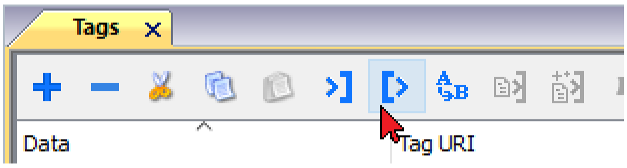


The following dialog shows which importer type can be selected.



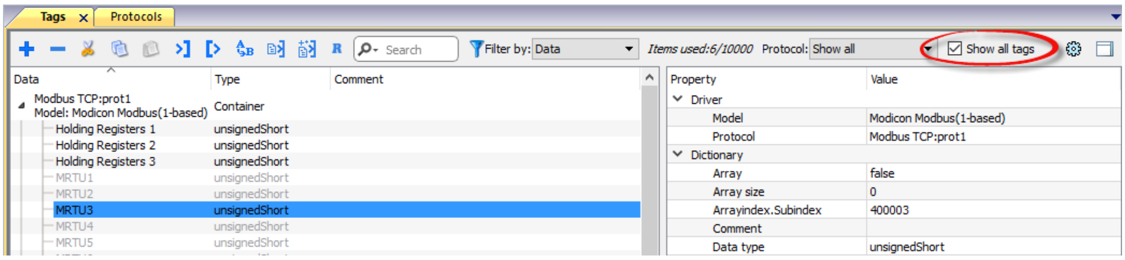
Multiple tag importers are available for this protocol. Please select the importer type and continue.

Version	Type
GX Works2 v1.0	Linear
Tag Editor exported xml	General

Importer	Description
GX Works2 v1.0 Linear	Requires a **.csv**file. 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**.

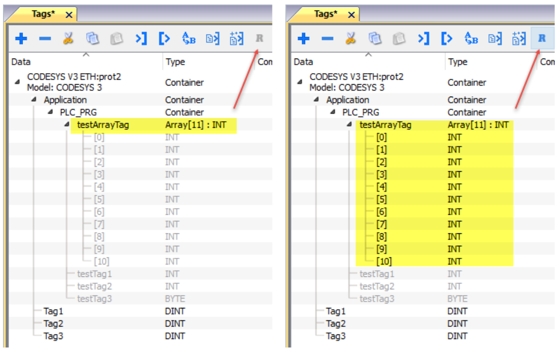
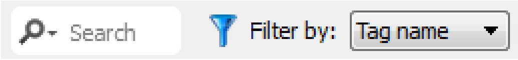
The tags available within the Dictionary but not imported into the project are gray and are visible only when the "Show all tags" check box is selected.



Data	Type	Comment
Modbus TCP:prot1 Model: Modicon Modbus(1-based)	Container	
Holding Registers 1	unsignedShort	
Holding Registers 2	unsignedShort	
Holding Registers 3	unsignedShort	
MRTU1	unsignedShort	
MRTU2	unsignedShort	
MRTU3	unsignedShort	
MRTU4	unsignedShort	
MRTU5	unsignedShort	

Property	Value
Driver	
Model	Modicon Modbus(1-based)
Protocol	Modbus TCP:prot1
Dictionary	
Array	false
Array size	0
Arrayindex.Subindex	400003
Comment	
Data type	unsignedShort

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

Toolbar item	Description
	 Tags* Data | Type | Con CODESYS V3 ETH:prot2 Model: CODESYS 3 | Container Application | Container PLC_PRG | Container testArrayTag | Array[11] : INT [0] | INT [1] | INT [2] | INT [3] | INT [4] | INT [5] | INT [6] | INT [7] | INT [8] | INT [9] | INT [10] | INT testTag1 | INT testTag2 | INT testTag3 | BYTE Tag1 | DINT Tag2 | DINT Tag3 | DINT
 Search Filter by: Tag name	Searches tags in the dictionary basing on filter combo-box item selected.

Communication Status

The current communication status can be displayed using the dedicated system variables. Please refer to the User Manual for further information about available system variables and their use.

The codes supported for this communication driver are:

Error	Notes
NAK	Returned in case the controller replies with a not acknowledge
Timeout	Returned when a request is not replied within the specified timeout period; ensure the controller is connected and properly configured to get network access
Invalid response	The panel did receive from the controller a response, but its format or its contents is not as expected; ensure the data programmed in the project are consistent with the controller resources
General Error	Error cannot be identified; should never be reported; contact technical support

Mitsubishi FX SER

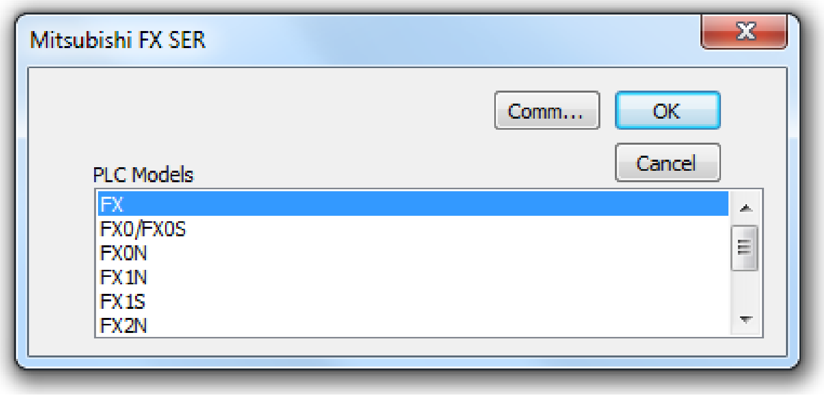
The HMI operator panels can be connected to Mitsubishi FX PLC as the network master using this communication driver.

The protocol has been designed to connect to the programming port of the PLC.

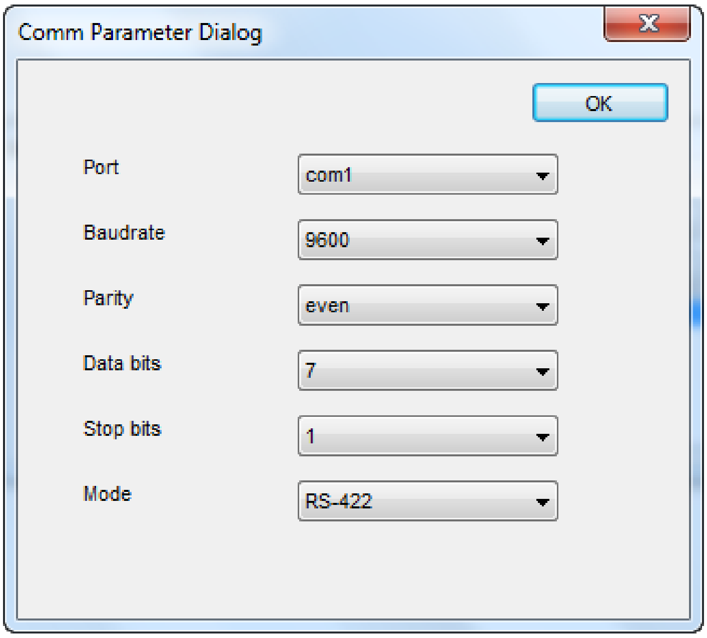
Please note that changes in the communication protocol specifications or PLC hardware may have occurred since this documentation was created. Some changes may eventually affect the functionality of this communication driver. Always test and verify the functionality of your application. To fully support changes in PLC hardware and communication protocols, communication drivers are continuously updated. Always ensure that the latest version of communication driver is used in your application.

Protocol Editor Settings

Add [+] a driver in the Protocol editor and select the protocol called “Mitsubishi FX SER” from the list of available protocols.



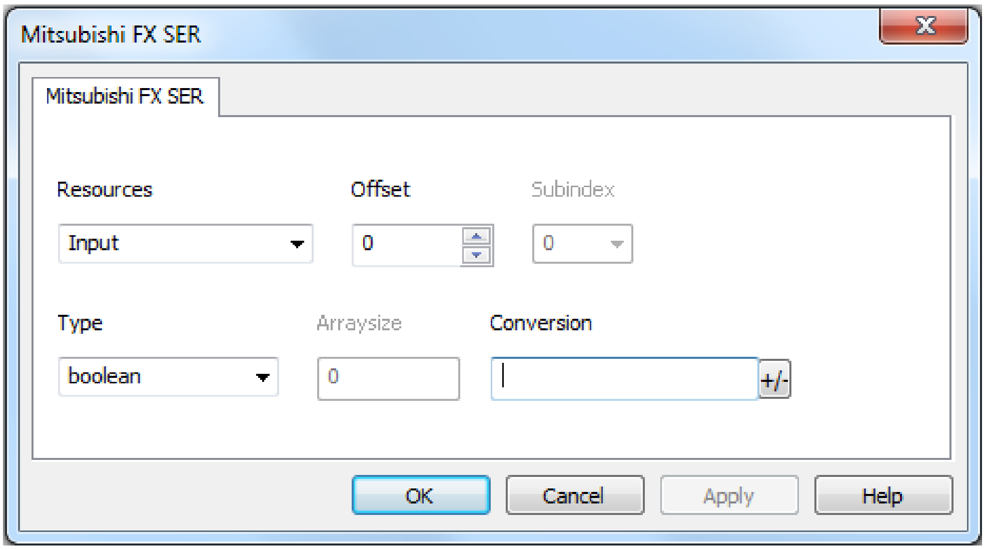
Element	Description
PLC Models	The list allows selecting the PLC model you are going to connect to. The selection will influence the data range offset per each data type according to the specific PLC memory resources.
Comm...	Gives access to the serial port configuration parameters as shown in the figure below.

<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td></td><td>
Comm Parameter Dialog
Port: com1
Baudrate: 9600
Parity: even
Data bits: 7
Stop bits: 1
Mode: RS-422
</td></tr>
<tr><td><b>Port</b></td><td>Serial port selection:
<table>
<tr><th>Port</th><th>Series 400</th><th>Series 500/600</th></tr>
<tr><td><b>com1</b></td><td>PLC Port</td><td>Onboard Serial Port</td></tr>
<tr><td><b>com2</b></td><td>PC/Printer Port</td><td>Optional Module on slot #1 or #2</td></tr>
<tr><td><b>com3</b></td><td>Not available</td><td>Optional Module on slot #3 or #4</td></tr>
</table></td></tr>
<tr><td><b>Baud rate, Parity, Data bits, Stop bits</b></td><td>Communication parameters for serial communication</td></tr>
<tr><td><b>Mode</b></td><td>Serial port mode; available options:
RS-232,
RS-485 (2 wires)
RS-422 (4 wires)</td></tr>
</table>

Tag Editor Settings

Into Tag editor select the protocol “Mitsubishi FX ETH” from the list of defined protocols and add a tag using [+] button.

Tag settings can be defined using the following dialog:



Mitsubishi FX SER

Resources: Input

Offset: 0

Subindex: 0

Type: boolean

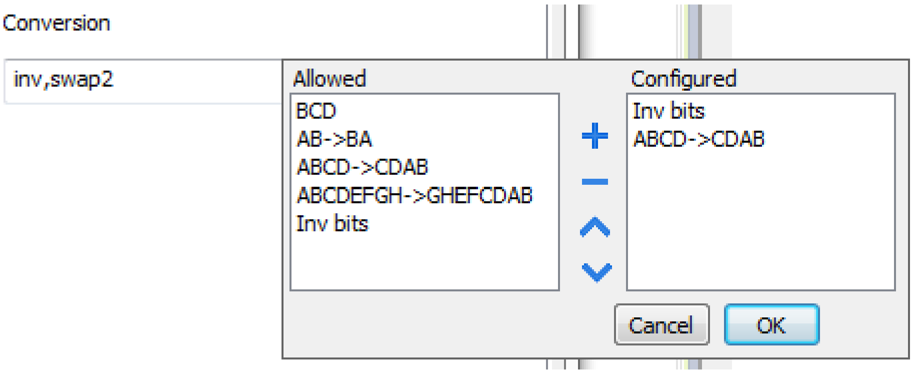
Arraysize: 0

Conversion:

Element	Description
Resources	Area of PLC where tag is located
Offset	Offset address where tag is located.
SubIndex	This allows resource offset selection within the register.
Type	see table below
Arraysize	• 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.

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
unsignedByte	8-bit data	0 ... 255
unsignedShort	16-bit data	0 ... 65535
unsignedInt	32-bit data	0 ... 4.2e9
float	IEEE single-precision 32-bit floating point type	1.17e-38 ... 3.40e38
string	Refer to “String data type chapter”	

Note: to define arrays, select one of Data Type format followed by square brackets like “byte[]”, “short[]”…

Element	Description
	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.  Conversion: inv,swap2 Allowed BCD AB->BA ABCD->CDAB ABCDEFGH->GHEFCDAB Inv bits Configured Inv bits ABCD->CDAB Cancel OK Depending on data type selected, the list **Allowed** shows one or more conversion types.

Value	Description
Inv bits	**inv**: Invert all the bits of the tag. *Example:* 1001 → 0110 (in binary format) 9 → 6 (in decimal format)
Negate	**neg**: Set the opposite of tag value. *Example:* 25.36 → -25.36
AB -> BA	**swapnibbles**: Swap nibbles in a byte. *Example:* 15D4 → 514D (in hexadecimal format) 5588 → 20813 (in decimal format)
ABCD -> CDAB	**swap2**: Swap bytes in a word. *Example:* 9ACC → CC9A (in hexadecimal format) 39628 → 52378 (in decimal format)
ABCDEFGH -> GHEFCDAB	**swap4**: Swap bytes in a double word. *Example:* 32FCFF54 → 54FFFC32 (in hexadecimal format) 855441236 → 1426062386 (in decimal format)

<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td><b>ABC...NOP -> OPM...DAB</b></td><td><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)</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>
</table>
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.
</td></tr>
</table>

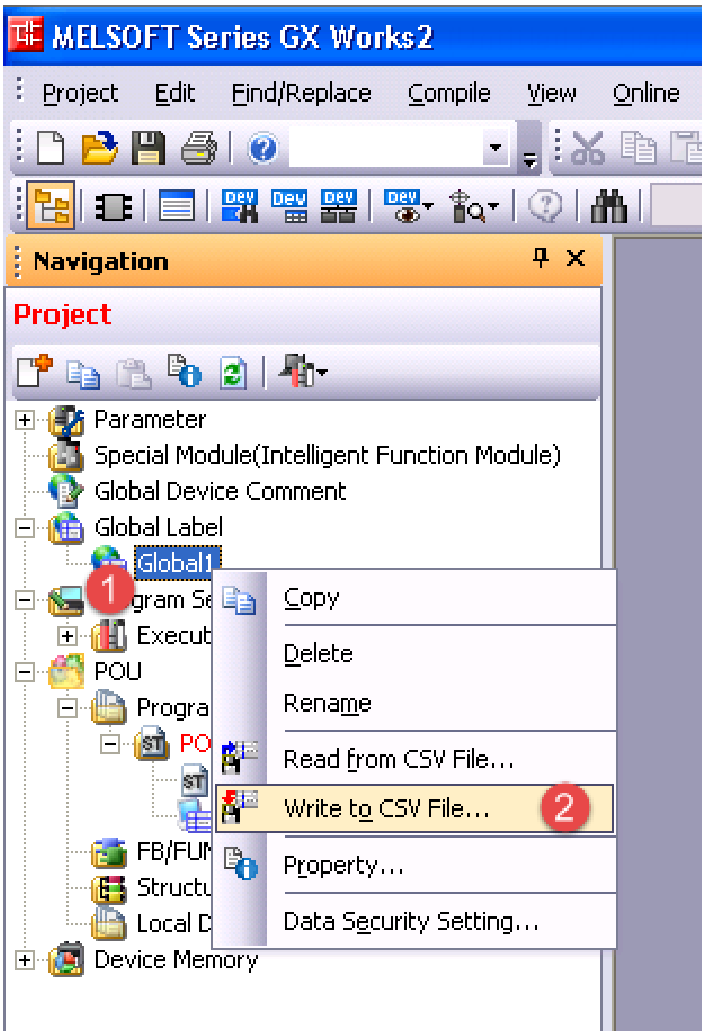
Tag Import

Exporting Tags from PLC

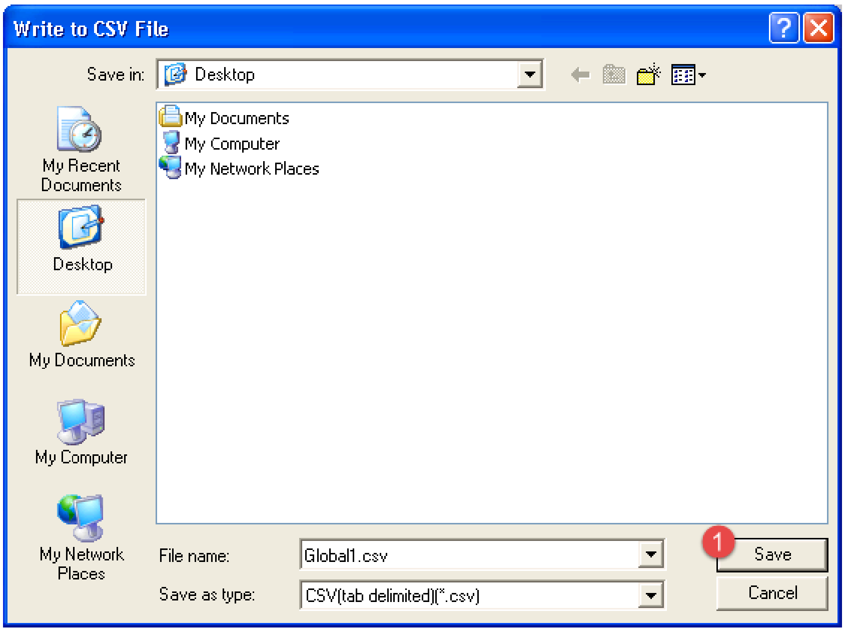
The Mitsubishi FX Serial tag import accepts symbol files with extension “csv” created by the Mitsubishi GX Works2 (Not from GX Developer).

The “.csv” file can be exported from the Project tree, as shown in the following figure.

1. Right-click on the Global variable list that need to be exported,
2. Select “Write to CSV File…”

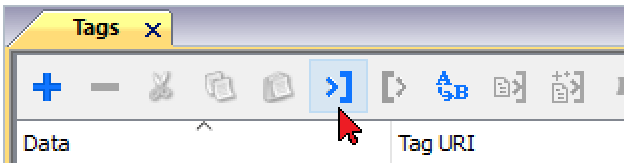


Into following dialog select the file name and location:

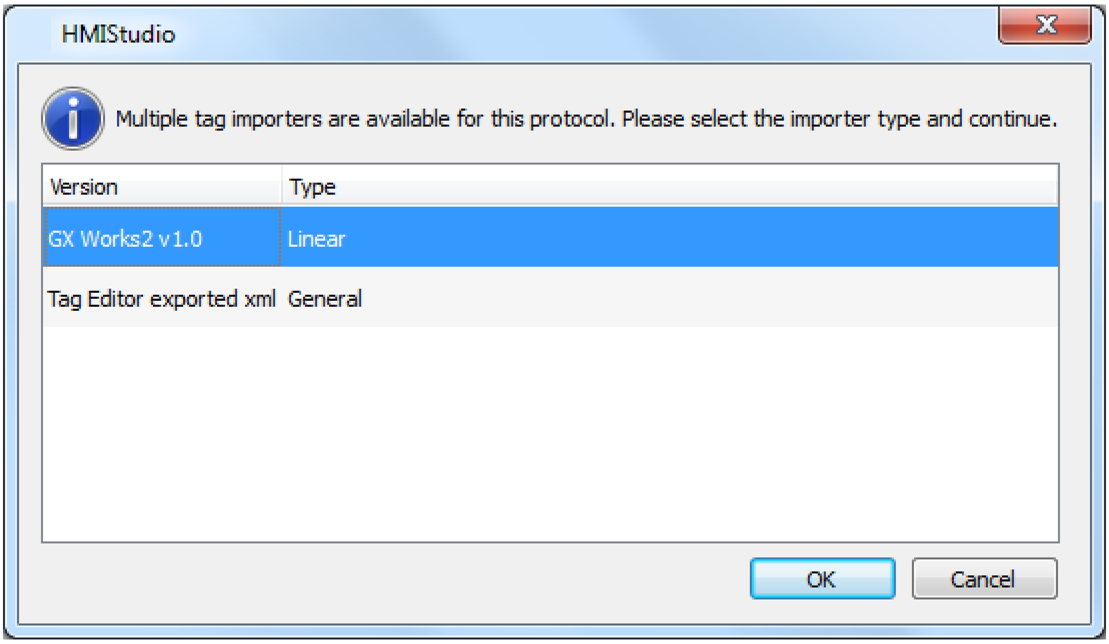


Importing Tags in Tag Editor

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

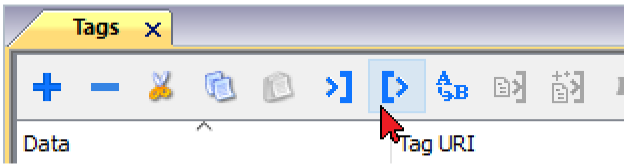


The following dialog shows which importer type can be selected.



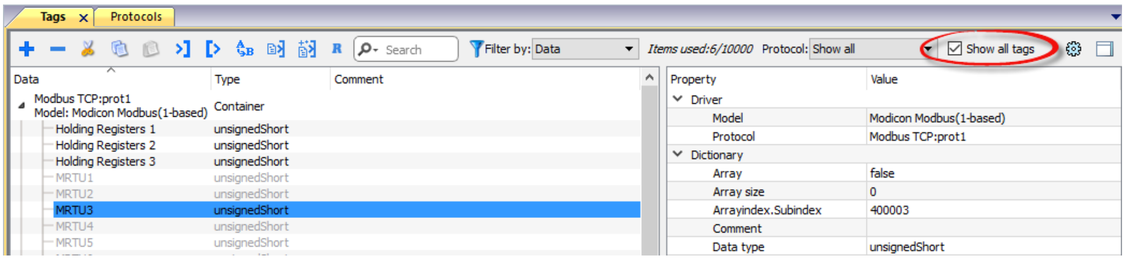
Multiple tag importers are available for this protocol. Please select the importer type and continue.

Version	Type
GX Works2 v1.0	Linear
Tag Editor exported xml	General

Importer	Description
GX Works2 v1.0 Linear	Requires a **.csv** file. 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. 

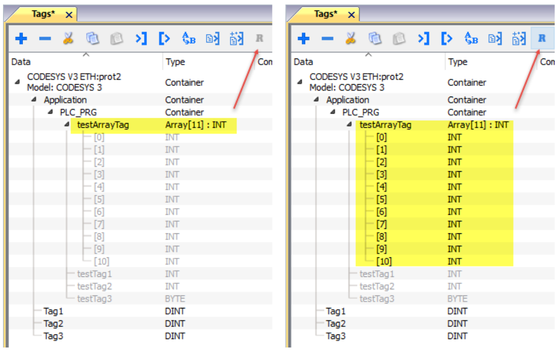
Once the importer has been selected, locate the symbol file and click **Open**.

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.



Data	Type	Comment
Modbus TCP:prot1 Model: Modicon Modbus(1-based)	Container	
Holding Registers 1	unsignedShort	
Holding Registers 2	unsignedShort	
Holding Registers 3	unsignedShort	
MRTU1	unsignedShort	
MRTU2	unsignedShort	
MRTU3	unsignedShort	
MRTU4	unsignedShort	
MRTU5	unsignedShort	

Property	Value
Driver	
Model	Modicon Modbus(1-based)
Protocol	Modbus TCP:prot1
Dictionary	
Array	false
Array size	0
Arrayindex.Subindex	400003
Comment	
Data type	unsignedShort

Toolbar item	Description
	Import Tag(s). Select tags to be imported and click on this icon to add tags from tag dictionary to the project
	Update Tag(s). Click on this icon to update the tags in the project, due a new dictionary import.
R	Check this box to import all sub-elements of a tag. Example of both checked and unchecked result:  Tags* × Data / Type / Con CODESYS V3 ETH:prot2 Model: CODESYS 3 — Container Application — Container PLC_PRG — Container testArrayTag — Array[11] : INT [0] … [10] — INT testTag1 — INT testTag2 — INT testTag3 — BYTE Tag1 — DINT Tag2 — DINT Tag3 — DINT
Search Filter by: Tag name ▾	Searches tags in the dictionary basing on filter combo-box item selected.

Communication Status

The current communication status can be displayed using the dedicated system variables. Please refer to the User Manual for further information about available system variables and their use.

The codes supported for this communication driver are:

Error	Notes
NAK	Returned in case the controller replies with a not acknowledge
Timeout	Returned when a request is not replied within the specified timeout period; ensure the controller is connected and properly configured to get network access

Error	Notes
Line Error	Returned when an error on the communication parameter setup is detected (parity, baud rate, data bits, stop bits); ensure the communication parameter settings of the controller is compatible with panel communication setup
Invalid response	The panel did receive from the controller a response, but its format or its contents is not as expected; ensure the data programmed in the project are consistent with the controller resources
General Error	Error cannot be identified; should never be reported; contact technical support

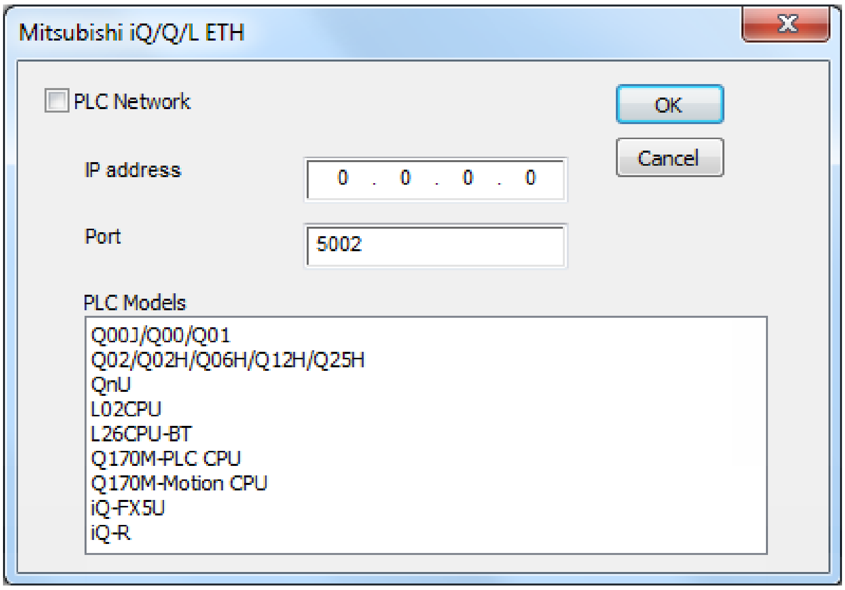
Mitsubishi iQ/Q/L ETH

The Mitsubishi iQ/Q/L ETH driver supports communication with Mitsubishi controllers with integrated Ethernet port and with external Ethernet card (QJ71E71-100).

Protocol Editor Settings

Add (+) a driver in the Protocol editor and select the protocol called “Mitsubishi iQ/Q/L ETH” from the list of available protocols.

The driver configuration dialog is shown as in the following figure:



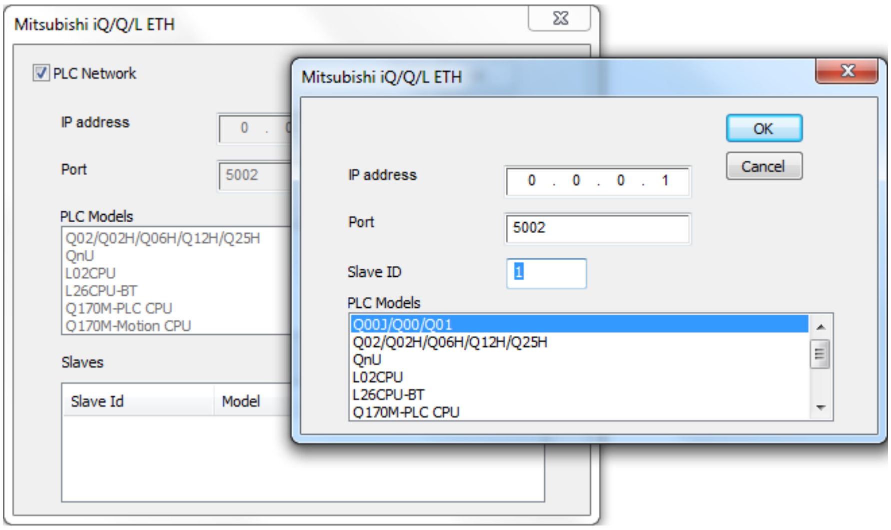
□ PLC Network

IP address 0 . 0 . 0 . 0

Port 5002

PLC Models
Q00J/Q00/Q01
Q02/Q02H/Q06H/Q12H/Q25H
QnU
L02CPU
L26CPU-BT
Q170M-PLC CPU
Q170M-Motion CPU
iQ-FX5U
iQ-R

Element	Description
IP address	Ethernet IP address of the controller
Port	Specifies the port number (decimal) used in the communication with the PLC.

Element	Description
PLC Model	The driver supports communication with different Mitsubishi iQ, Q and L controllers. Note: PLC Model selection has only effect on range values of variables. If a particular model is not present in the list, try selecting a similar one. If range values of variables are the same, the communication will be correctly established.
PLC Network	The protocol allows the connection of multiple controllers to one HMI device. To set-up multiple connections, check “PLC network” checkbox and create your network using the command “Add” per each slave device you need to include in the network. 

Mitsubishi iQ/Q/L ETH

☑ PLC Network

IP address: 0 . 0 . 0 . 1

Port: 5002

PLC Models: Q02/Q02H/Q06H/Q12H/Q25H, QnU, L02CPU, L26CPU-BT, Q170M-PLC CPU, Q170M-Motion CPU

Slaves: Slave Id, Model

Mitsubishi iQ/Q/L ETH

IP address: 0 . 0 . 0 . 1

Port: 5002

Slave ID: 1

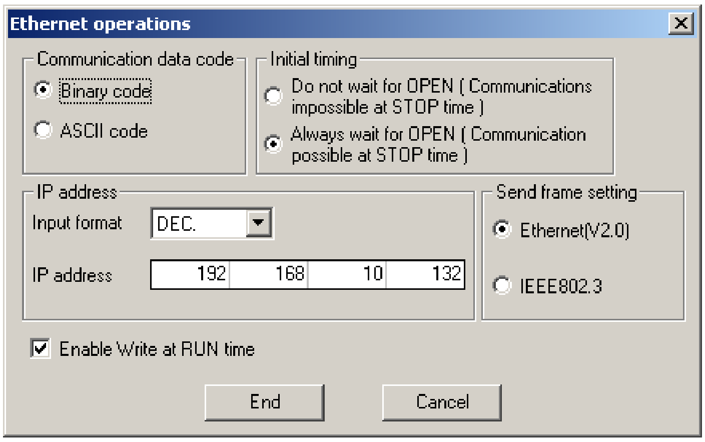
PLC Models: Q00J/Q00/Q01, Q02/Q02H/Q06H/Q12H/Q25H, QnU, L02CPU, L26CPU-BT, Q170M-PLC CPU

Controller Settings

GX Works2

The Mitsubishi Q system must be properly configured for Ethernet communication using the Mitsubishi GX Developer software version 7 or higher, from GX Works2 software.

The Figure below shows an example of network configuration for Ethernet communication.



Ethernet operations

Communication data code: ● Binary code ○ ASCII code

Initial timing: ○ Do not wait for OPEN (Communications impossible at STOP time) ● Always wait for OPEN (Communication possible at STOP time)

IP address: Input format DEC.; IP address 192 168 10 132

Send frame setting: ● Ethernet(V2.0) ○ IEEE802.3

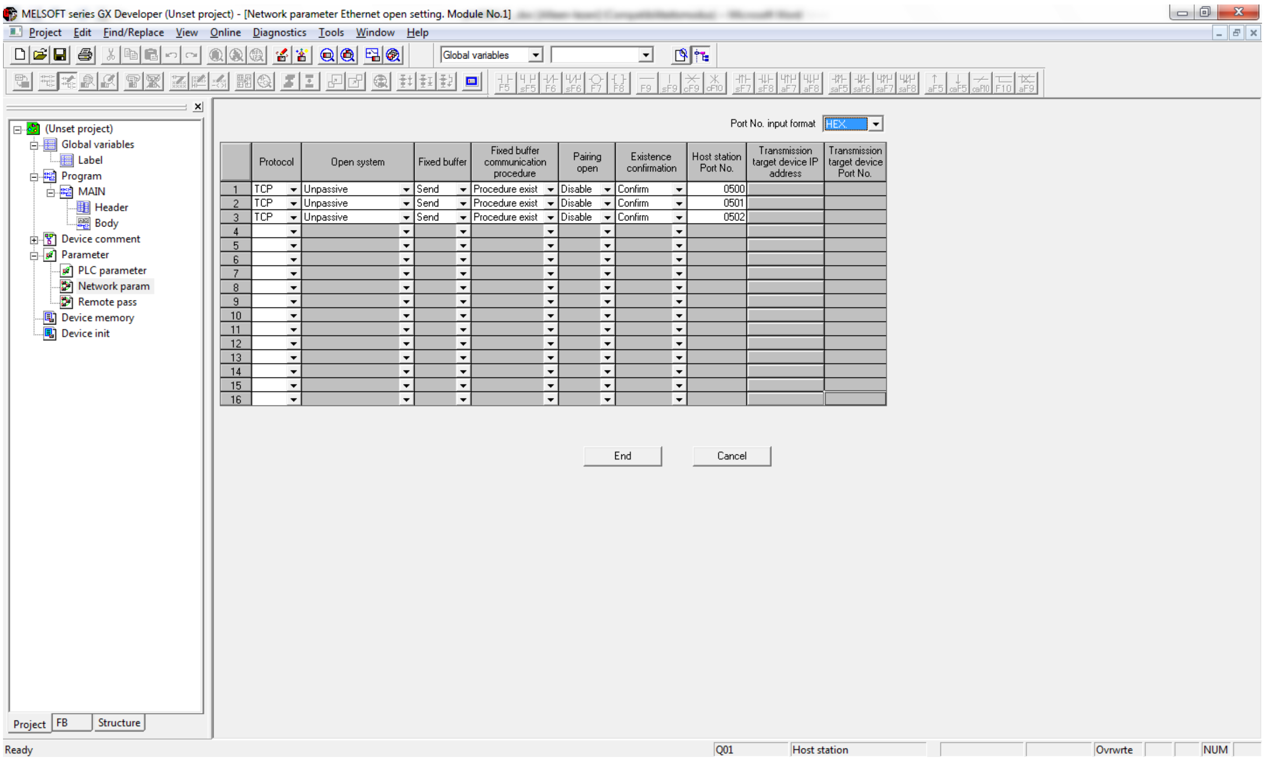
☑ Enable Write at RUN time

Please note that the communication protocol supports only Binary code communication.

The PLC system must be configured to accept incoming data from the external device.

In the GX Developer Software open “Parameters”, “Network Param” and select Ethernet/ CC IE/ MELSECNET”. Add the number of connections of the operator panels you want to configure in the network.

When using the Mitsubishi CPU with external Ethernet card (QJ71E71-100) the connections have to be configured according to the following figure as "Unpassive":



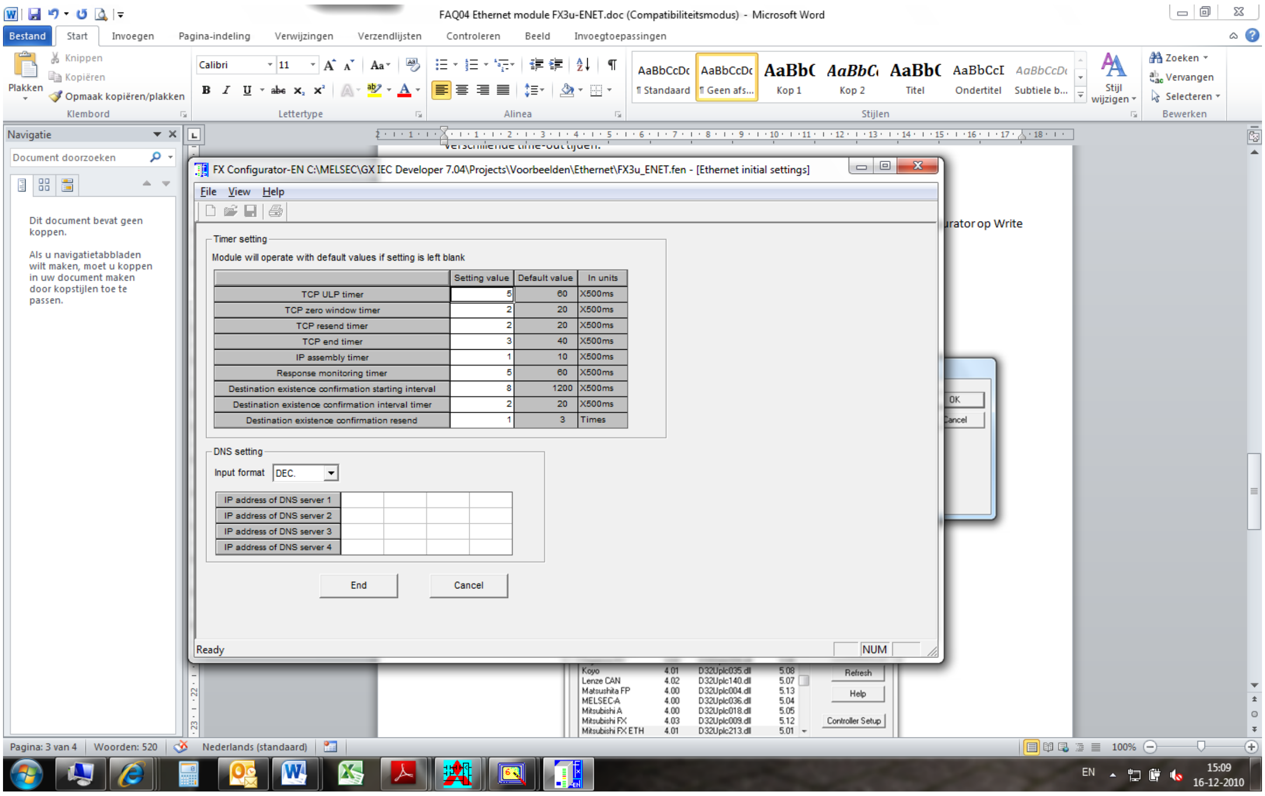
Port No. input format: HEX.

	Protocol	Open system	Fixed buffer	Fixed buffer communication procedure	Pairing open	Existence confirmation	Host station Port No.	Transmission target device IP address	Transmission target device Port No.
1	TCP	Unpassive	Send	Procedure exist	Disable	Confirm	0500		
2	TCP	Unpassive	Send	Procedure exist	Disable	Confirm	0501		
3	TCP	Unpassive	Send	Procedure exist	Disable	Confirm	0502		
4									
5									
6									
7									
8									
9									
10									
11									
12									
13									
14									
15									
16									

When the “Existence confirmation” setting has been set to Confirm, the TCP connection will be closed when it is not used (connection lost); by default the TCP port remains open and it is not possible to reconnect.

Note: The GX Developer software allows entering the conventional representation settings (decimal or hexadecimal) for the port number; in the above figure it is in hexadecimal.

In the next figures there are 2 examples about how to set “Initial settings” for 5 and 15 seconds timeout.



FX Configurator-EN C:\MELSEC\GX IEC Developer 7.04\Projects\Voorbeelden\Ethernet\FX3u_ENET.fen - [Ethernet initial settings]

Timer setting

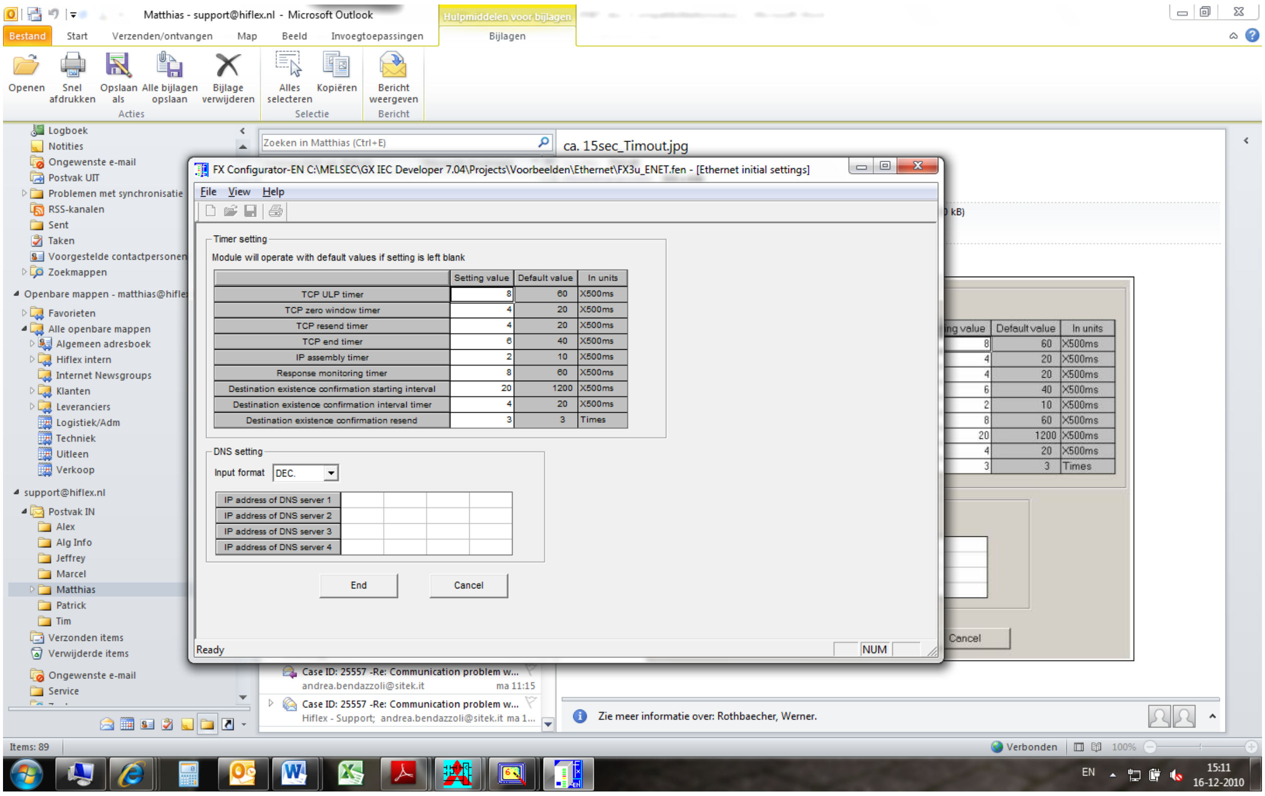
Module will operate with default values if setting is left blank

	Setting value	Default value	In units
TCP ULP timer	5	60	X500ms
TCP zero window timer	2	20	X500ms
TCP resend timer	2	20	X500ms
TCP end timer	3	40	X500ms
IP assembly timer	1	10	X500ms
Response monitoring timer	5	60	X500ms
Destination existence confirmation starting interval	8	1200	X500ms
Destination existence confirmation interval timer	2	20	X500ms
Destination existence confirmation resend	1	3	Times

DNS setting

Input format: DEC.

IP address of DNS server 1				
IP address of DNS server 2				
IP address of DNS server 3				
IP address of DNS server 4				



FX Configurator-EN C:\MELSEC\GX IEC Developer 7.04\Projects\Voorbeelden\Ethernet\FX3u_ENET.fen - [Ethernet initial settings]

Timer setting

Module will operate with default values if setting is left blank

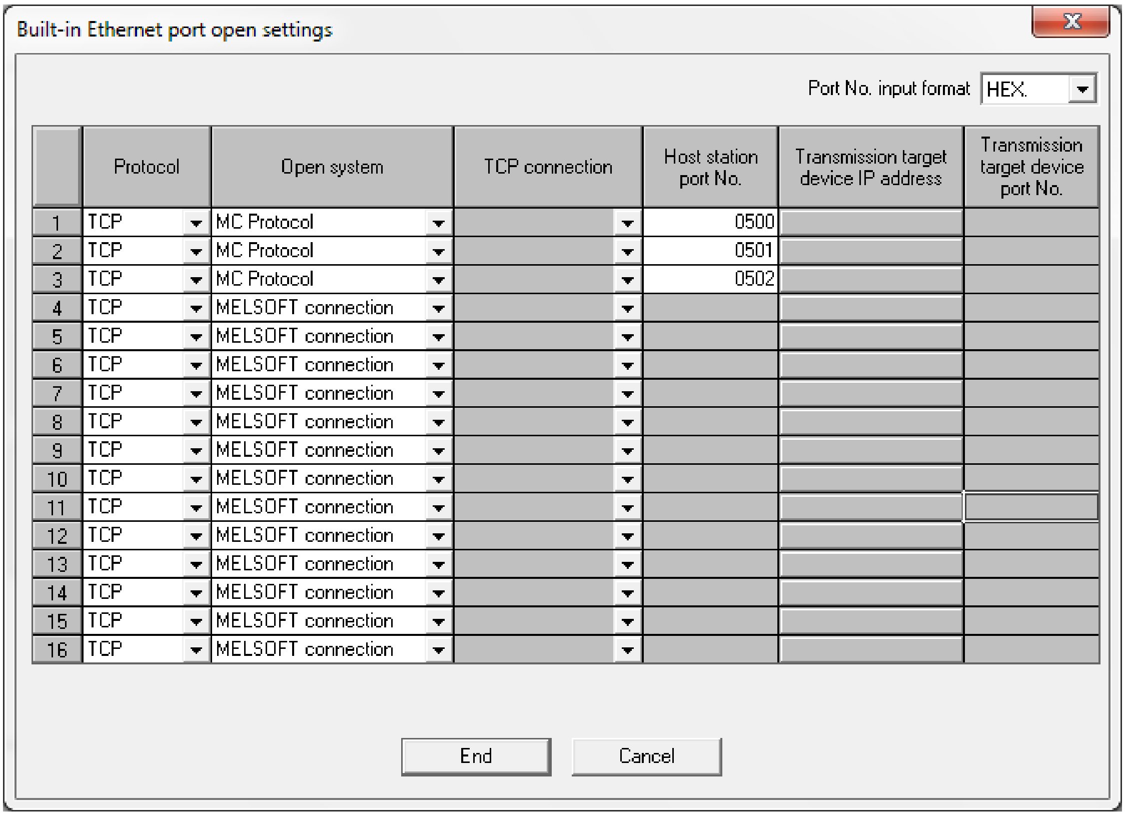
	Setting value	Default value	In units
TCP ULP timer	8	60	X500ms
TCP zero window timer	4	20	X500ms
TCP resend timer	4	20	X500ms
TCP end timer	6	40	X500ms
IP assembly timer	2	10	X500ms
Response monitoring timer	8	60	X500ms
Destination existence confirmation starting interval	20	1200	X500ms
Destination existence confirmation interval timer	4	20	X500ms
Destination existence confirmation resend	3	3	Times

DNS setting

Input format: DEC.

IP address of DNS server 1				
IP address of DNS server 2				
IP address of DNS server 3				
IP address of DNS server 4				

When using Mitsubishi CPU with integrated Ethernet port the "Open System" settings should be changed to "MC connection"


Built-in Ethernet port open settings

Port No. input format: HEX.

	Protocol	Open system	TCP connection	Host station port No.	Transmission target device IP address	Transmission target device port No.
1	TCP	MC Protocol		0500		
2	TCP	MC Protocol		0501		
3	TCP	MC Protocol		0502		
4	TCP	MELSOFT connection				
5	TCP	MELSOFT connection				
6	TCP	MELSOFT connection				
7	TCP	MELSOFT connection				
8	TCP	MELSOFT connection				
9	TCP	MELSOFT connection				
10	TCP	MELSOFT connection				
11	TCP	MELSOFT connection				
12	TCP	MELSOFT connection				
13	TCP	MELSOFT connection				
14	TCP	MELSOFT connection				
15	TCP	MELSOFT connection				
16	TCP	MELSOFT connection				

Note: The number format for Host Station Port No. is hexadecimal, not decimal.

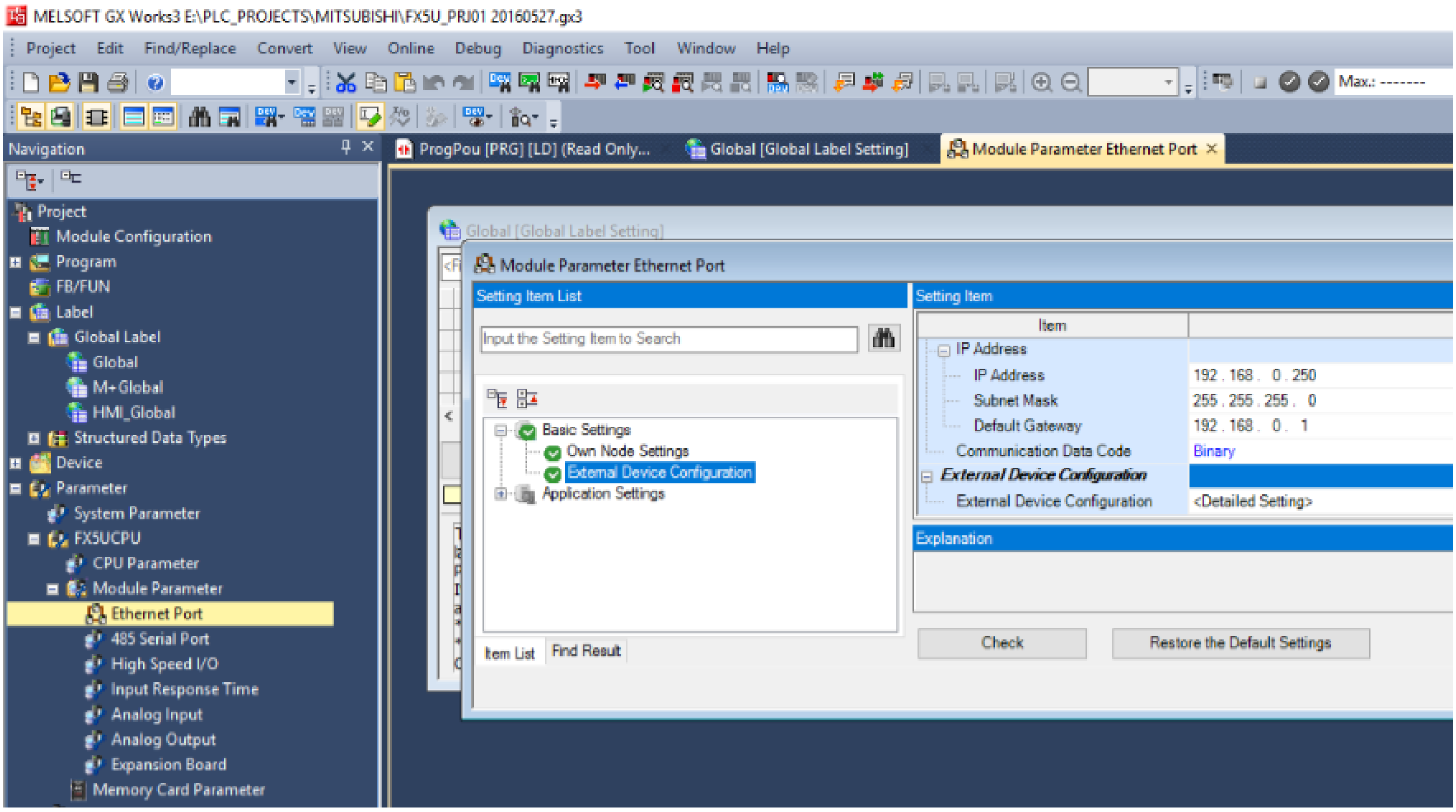
GX Works3

The Mitsubishi Q system must be properly configured for Ethernet communication using GX Works3 software.

The communication driver is based on SLMP function.

SLMP (Seamless Message Protocol) is a protocol for accessing SLMP-compatible devices from an external device (such as HMI) using TCP or UDP through Ethernet.

From GX Works3 software, Ethernet port parameters must be set from **Module parameter > Ethernet Port > Basic Settings > Own Node Settings**.

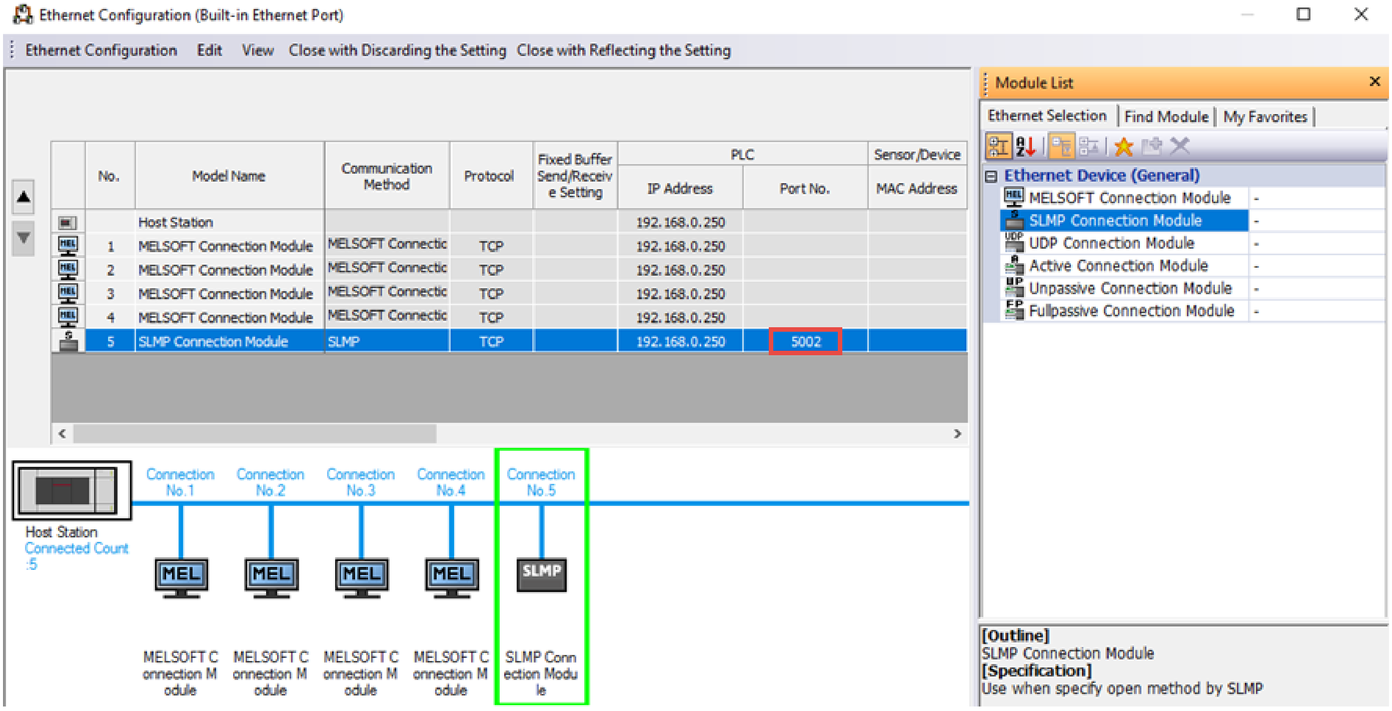


MELSOFT GX Works3 E:\PLC_PROJECTS\MITSUBISHI\FX5U_PRJ01 20160527.gx3

Module Parameter Ethernet Port

Item	
IP Address	
IP Address	192 . 168 . 0 . 250
Subnet Mask	255 . 255 . 255 . 0
Default Gateway	192 . 168 . 0 . 1
Communication Data Code	Binary
External Device Configuration	
External Device Configuration	<Detailed Setting>

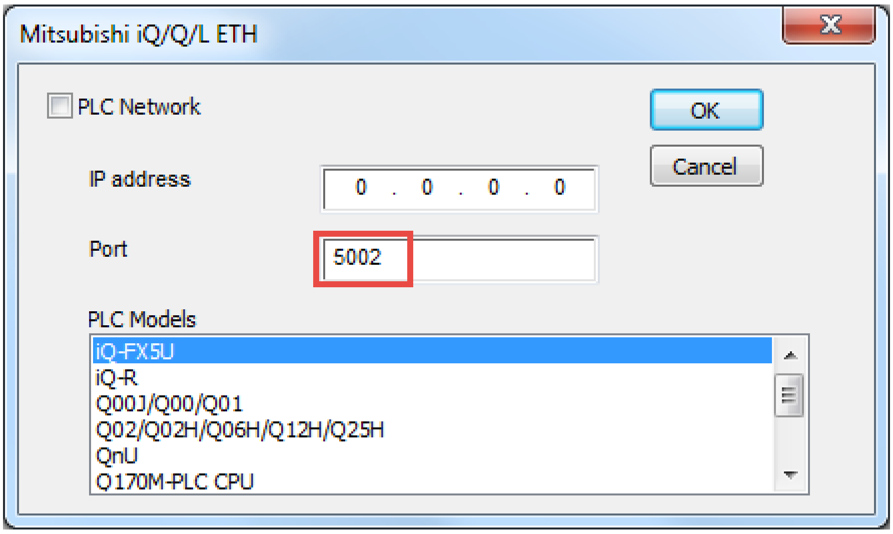
SLMP Connection Module must be added in **Module parameter > Ethernet Port > Basic Settings > External Device Configuration > Detailed Settings > Ethernet Configuration (Built-in Ethernet Port)**. **Port No.** parameter must be the same as per **Port** parameter from Protocol Editor Settings (see images below).


Ethernet Configuration (Built-in Ethernet Port)
Ethernet Configuration
Edit
View
Close with Discarding the Setting
Close with Reflecting the Setting
Module List
Ethernet Selection
Find Module
My Favorites

No.	Model Name	Communication Method	Protocol	Fixed Buffer Send/Receive Setting	PLC IP Address	PLC Port No.	Sensor/Device MAC Address
	Host Station				192.168.0.250		
1	MELSOFT Connection Module	MELSOFT Connectic	TCP		192.168.0.250		
2	MELSOFT Connection Module	MELSOFT Connectic	TCP		192.168.0.250		
3	MELSOFT Connection Module	MELSOFT Connectic	TCP		192.168.0.250		
4	MELSOFT Connection Module	MELSOFT Connectic	TCP		192.168.0.250		
5	SLMP Connection Module	SLMP	TCP		192.168.0.250	5002	

Ethernet Device (General)	
MELSOFT Connection Module	-
SLMP Connection Module	-
UDP Connection Module	-
Active Connection Module	-
Unpassive Connection Module	-
Fullpassive Connection Module	-

[Outline]
SLMP Connection Module
[Specification]
Use when specify open method by SLMP



□ PLC Network

IP address: 0 . 0 . 0 . 0

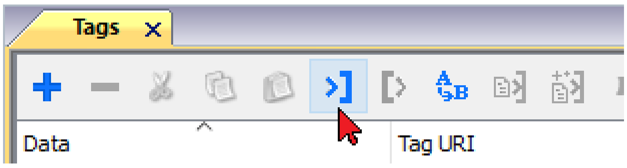
Port: 5002

PLC Models
- iQ-FX5U
- iQ-R
- Q00J/Q00/Q01
- Q02/Q02H/Q06H/Q12H/Q25H
- QnU
- Q170M-PLC CPU

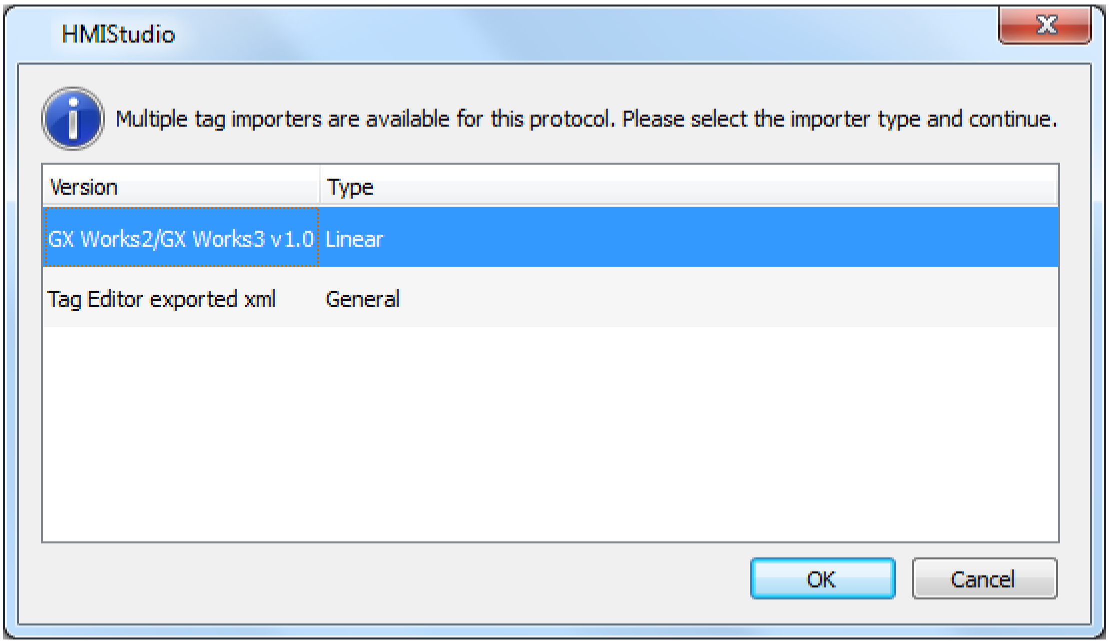
Note: To actually get communication with HMI it is necessary to initialize the PLC after the above settings have been applied.
To initialize the PLC it possibile to use the Run/Stop/Reset switch or by simply rebooting the PLC.

Tag Import

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

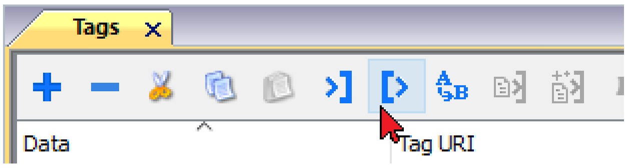


The following dialog shows which importer type can be selected.



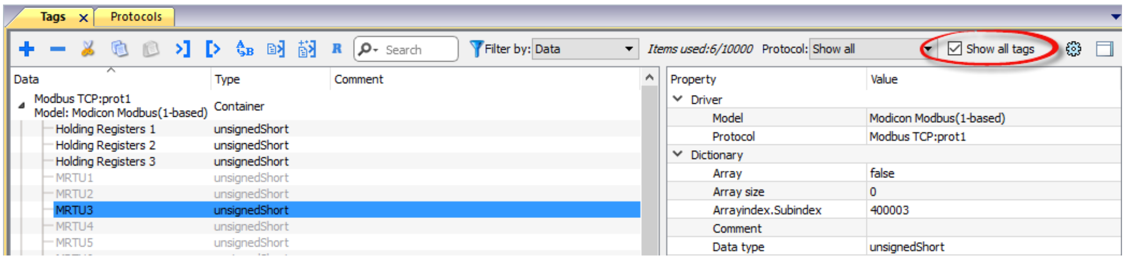
Multiple tag importers are available for this protocol. Please select the importer type and continue.

Version	Type
GX Works2/GX Works3 v1.0	Linear
Tag Editor exported xml	General

Importer	Description
GX Works2/GX Works3 v1.0 **Linear**	Requires a **.csv**file. 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. 

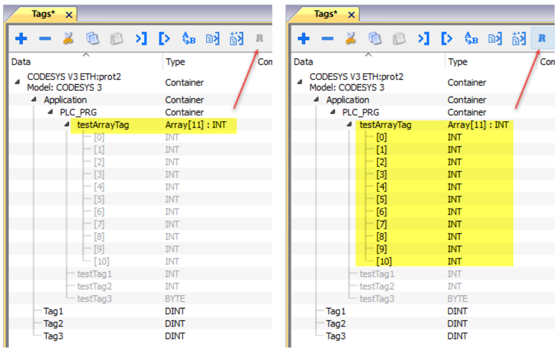
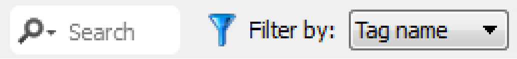
Once the importer has been selected, locate the symbol file and click **Open**.

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.



Data	Type	Comment
Modbus TCP:prot1 Model: Modicon Modbus(1-based)	Container	
Holding Registers 1	unsignedShort	
Holding Registers 2	unsignedShort	
Holding Registers 3	unsignedShort	
MRTU1	unsignedShort	
MRTU2	unsignedShort	
MRTU3	unsignedShort	
MRTU4	unsignedShort	
MRTU5	unsignedShort	

Property	Value
Driver	
Model	Modicon Modbus(1-based)
Protocol	Modbus TCP:prot1
Dictionary	
Array	false
Array size	0
Arrayindex.Subindex	400003
Comment	
Data type	unsignedShort

Toolbar item	Description
	Import Tag(s). Select tags to be imported and click on this icon to add tags from tag dictionary to the project
	Update Tag(s). Click on this icon to update the tags in the project, due a new dictionary import.
R	Check this box to import all sub-elements of a tag. Example of both checked and unchecked result:  Tags* × Data Type Con CODESYS V3 ETH:prot2 Model: CODESYS 3 Container Application Container PLC_PRG Container testArrayTag Array[11] : INT [0] INT [1] INT [2] INT [3] INT [4] INT [5] INT [6] INT [7] INT [8] INT [9] INT [10] INT testTag1 INT testTag2 INT testTag3 BYTE Tag1 DINT Tag2 DINT Tag3 DINT Tags* × Data Type Con CODESYS V3 ETH:prot2 Model: CODESYS 3 Container Application Container PLC_PRG Container testArrayTag Array[11] : INT [0] INT [1] INT [2] INT [3] INT [4] INT [5] INT [6] INT [7] INT [8] INT [9] INT [10] INT testTag1 INT testTag2 INT testTag3 BYTE Tag1 DINT Tag2 DINT Tag3 DINT
 Search Filter by: Tag name	Searches tags in the dictionary basing on filter combo-box item selected.

Communication Status

The current communication status can be displayed using the dedicated system variables. Please refer to the User Manual for further information about available system variables and their use.

The codes supported for this communication driver are:

Error	Notes
NAK	Returned in case the controller replies with a not acknowledge
Timeout	Returned when a request is not replied within the specified timeout period; ensure the controller is connected and properly configured to get network access
Invalid response	The panel did receive from the controller a response, but its format or its contents is not as expected; ensure the data programmed in the project are consistent with the controller resources
General Error	Error cannot be identified; should never be reported; contact technical support

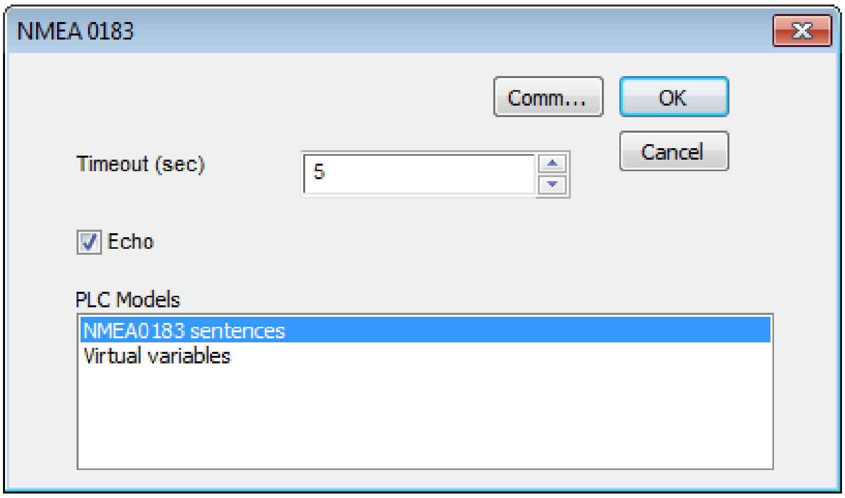
NMEA 0183

The NMEA 0183 driver has been developed to communicate with NMEA 0183 compatible devices trough the operator panel serial ports.

Protocol Editor Settings

Add (+) a new driver in the Protocol editor and select the protocol called “NMEA 0183” from the list of available protocols.

The driver configuration dialog is shown in the following figure.



NMEA 0183

Timeout (sec) 5

☑ Echo

PLC Models

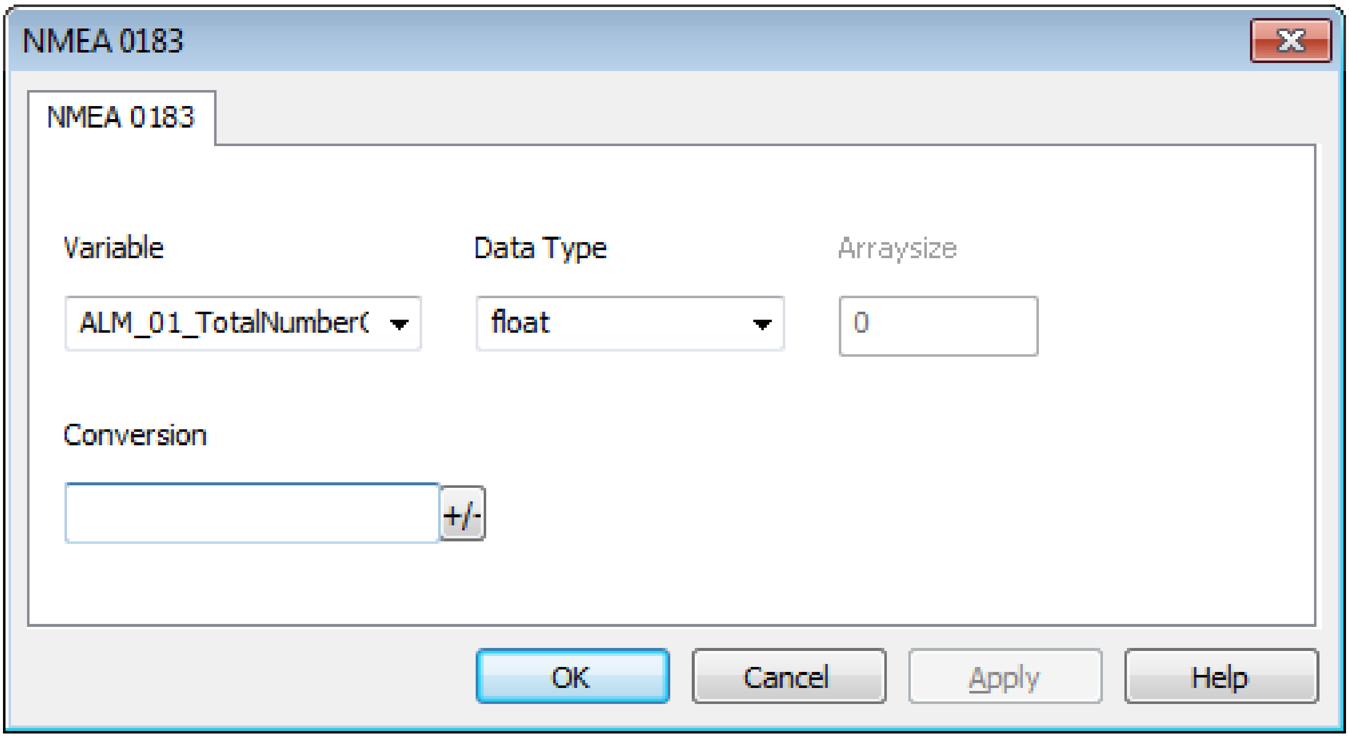
NMEA0183 sentences
Virtual variables

Element	Description
Timeout (sec)	Defines the time inserted by the protocol between two retries of the same message in case of missing response from the server device. It is expressed in seconds.
Echo	If selected the NMEA messages received on the RX channel of serial port are sent out from the TX channel. This allows to continue the NMEA network downstream of the operator panel whether required.
PLC Models	Two PLC models are available: NMEA 0183 Sentences: when selected the Tags will point univocally to the specified NMEA sentence. Virtual variables: when selected the Tag will show the value coming from any NMEA sentence of the specified type, for example any NMEA sentence of Latitude type.

Tag Editor Settings

Into Tag editor select the protocol “NMEA 0183” from the list of defined protocols and add a tag using [+] button.

Tag settings can be defined using the following dialog:


NMEA 0183

Variable: ALM_01_TotalNumber(

Data Type: float

Arraysize: 0

Conversion:

Element	Description
Variable	The NMEA Sentence or Virtual variable
Data Type	

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
unsignedByte	8-bit data	0 ... 255
unsignedShort	16-bit data	0 ... 65535
unsignedInt	32-bit data	0 ... 4.2e9
float	IEEE single-precision 32-bit floating point type	1.17e-38 ... 3.40e38
string	String data	

Element	Description
Arraysize	• 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.

<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td></td><td>If Encoding property is set to UCS-2BE, UCS-2LE, UTF-16BE or UTF-16LE one character requires 2 bytes.</td></tr>
<tr><td><b>Conversion</b></td><td>Conversion to be applied to the tag.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>Degrees</td><td>Shows Degrees data only from coordinates sentence</td></tr>
<tr><td>Minutes</td><td>Shows Minutes data only from coordinates sentence</td></tr>
<tr><td>Seconds</td><td>Shows Seconds data only from coordinates sentence</td></tr>
</table>
</td></tr>
</table>

List of supported NMEA 0183 commands

The NMEA 0183 commands supported from the communication protocol are the following:

AAM_01_StatusArrivalCircle

AAM_02_StatusPerpendicular

AAM_03_ArrivalCircleRadius

AAM_04_UnitsOfRadius

AAM_05_WaypointID

ACK_01_LocalAlarmNumber

ALM_01_TotalNumberOfMessages

ALM_02_MessageNumber

ALM_03_SatelliteNumber

ALM_04_WeekNumber

ALM_05_SVhealth

ALM_06_Eccentricity

ALM_07_AlmanacReferenceTime

ALM_08_InclinacionAngle

ALM_09_RateOfRightAscension

ALM_10_RootOfSemimajorAxis

ALM_11_ArgumentOfPerigee

ALM_12_LongitudeOfAscesionNode

ALM_13_MeanAnomaly

ALM_14_ClockParameter0

ALM_15_ClockParameter1

ALR_01_TimeOfAlarmConditionChange

ALR_02_LocalAlarmNumber

ALR_03_AlarmCondition

ALR_04_AlarmAcknowledgeState

ALR_05_AlarmDescriptionState

APB_01_StatusSNR

APB_02_StatusLock

APB_03_MagnitudeOfXTE

APB_04_DirectionToStear

APB_05_UnitsXTE

APB_06_StatusArrivalCircle

APB_07_StatusPerpendicular

APB_08_BearingOriginToDestination

APB_09_MagneticOrTrue

APB_10_DestinatonWaypointID

APB_11_Bearing

APB_12_BearingMagneticOrTrue

APB_13_HeadingToSteer

APB_14_HeadingMagneticOrTrue

APB_15_ModeIndicator

BEC_01_ObservationUTC

BEC_02_WaypointLatitude

BEC_03_WaypointLatitudeInd

BEC_04_WaypointLongitude

BEC_05_WaypointLongitudeInd

BEC_06_BearingTrue

BEC_07_BearingTrueInd

BEC_08_BearingMagnetic

BEC_09_BearingMagneticInd)

BEC_10_Distance

BEC_11_DistanceUnits

BEC_12_WaypointID

BOD_01_BearingTrue

BOD_02_BearingTrueInd

BOD_03_BearingMagnetic

BOD_04_BearingMagneticInd

BOD_05_DestinationWaypointID

BOD_06_OriginWaypointID

BWC_01_ObservationUTC

BWC_02_WaypointLatitude

BWC_03_WaypointLatitudeInd

BWC_04_WaypointLongitude

BWC_05_WaypointLongitudeInd)

BWC_06_BearingTrue

BWC_07_BearingTrueInd

BWC_08_BearingMagnetic

BWC_09_BearingMagneticInd

BWC_10_Distance

BWC_11_DistanceUnits

BWC_12_WaypointID

BWC_13_ModeIndicator

BWR_01_ObservationUTC

BWR_02_WaypointLatitude

BWR_03_WaypointLatitudeInd

BWR_04_WaypointLongitude

BWR_05_WaypointLongitudeInd

BWR_06_BearingTrue

BWR_07_BearingTrueInd

BWR_08_BearingMagnetic

BWR_09_BearingMagneticInd

BWR_10_Distance

BWR_11_DistanceInd

BWR_12_WaypointID

BWR_13_ModeIndicator

BWW_01_BearingTrue

BWW_02_BearingTrueInd

BWW_03_BearingMagnetic

BWW_04_BearingMagneticInd

BWW_05_ToWaypointID

BWW_06_FromWaypointID

DBT_01_WaterDepthFeet

DBT_02_WaterDepthFeetInd

DBT_03_WaterDepthMeters

DBT_04_WaterDepthMetersInd

DBT_05_WaterDepthFathoms

DBT_06_WaterDepthFathomsInd

DCN_01_DeccaChainIdentifier

DCN_02_RedZoneIdentifier

DCN_03_RedLineOfPosition

DCN_04_StatusRedMasterLine

DCN_05_GreenZoneIdentifier

DCN_06_GreenLineOfPosition

DCN_07_StatusGreenMasterLine

DCN_08_PurpleZoneIdentifier

DCN_09_PurpleLineOfPosition

DCN_10_StatusPurpleMasterLine

DCN_11_RedLineNavigationUse, A=Valid

DCN_12_GreenLineNavigationUse, A=Valid

DCN_13_PurpleLineNavigationUse, A=Valid

DCN_14_PositionUncertainty

DCN_15_PositionUncertaintyInd

DCN_16_FixDataBasis

DPT_01_WaterDepth

DPT_02_OffsetFromTransducer

DPT_03_MaximumRangeScale

DSC_01_FormatSpecifier

DSC_02_Address

DSC_03_Cattegory

DSC_04_NatureOfDistress

DSC_05_TypeOfCommunication

DSC_06_PositionOrChannel

DSC_07_TimeOrTelNo

DSC_08_ShipMMSI

DSC_09_NatureOfDistress

DSC_10_Acknowledgment

DSC_11_ExpansionIndicator

DSE_01_TotalNumberOfMessages

DSE_02_MessageNumber

DSE_03_Query_ReplyFlag

DSE_04_Vessel_MMSI

DSE_05_DataSet1Code

DSE_06_Dataset1Data

DSE_07_Dataset2Code

DSE_08_Dataset2Data

DSE_09_Dataset3Code

DSE_10_Dataset3Data

DSE_11_Dataset4Code

DSE_12_Dataset4Data

DSE_13_Dataset5Code

DSE_14_Dataset5Data

DSE_15_Dataset6Code

DSE_16_Dataset6Data

DSE_17_Dataset7Code

DSE_18_Dataset7Data

DSE_19_Dataset8Code

DSE_20_Dataset8Data

DSE_21_Dataset9Code

DSE_22_Dataset9Data

DSE_23_Dataset10Code

DSE_24_Dataset10Data

DSI_01_TotalNumberOfMessages

DSI_02_MessageNumber

DSI_03_Vessel_MMSI

DSI_04_VesselCourse

DSI_05_VesselType

DSI_06_GeographicArea

DSI_07_Commandset1Code

DSI_08_Commandset1Data

DSI_09_Commandset2Code

DSI_10_Commandset2Data

DSI_11_Commandset3Code

DSI_12_Commandset3Data

DSI_13_ExpansionIndicator

DSR_01_TotalNumberOfMessages

DSR_02_MessageNumber

DSR_03_Vessel_MMSI

DSR_04_Dataset1Code

DSR_05_Dataset1Data

DSR_06_Dataset2Code

DSR_07_Dataset2Data

DSR_08_Dataset3Code

DSR_09_Dataset3Data

DSR_10_ExpansionIndicator

DTM_01_LocalDatumCode

DTM_02_LocalDatumSubdivisioncode

DTM_03_LatOffset

DTM_04_LatOffsetInd

DTM_05_LonOffset

DTM_06_LonOffsetInd

DTM_07_AltitudeOffset

DTM_08_ReferenceDatumCode

FSI_01_TransmitingFrequency

FSI_02_ReceivingFrequency

FSI_03_ModeOfOperation

FSI_04_PowerLevel

GBS_01_UTC

GBS_02_ExpectedLatitudeError

GBS_03_ExpectedLongitudeError

GBS_04_ExpectedAltitudeError

GBS_05_FailedSatelliteID

GBS_06_ProbabilityOfMissedDetection

GBS_07_EstimateOfBiasMeters

GBS_08_StandardDeviationOfBiasEstimate

GGA_01_UTC

GGA_02_Latitude

GGA_03_LatitudeInd

GGA_04_Longitude

GGA_05_LongitudeInd

GGA_06_QualityIndicator

GGA_07_NumberOfSatellitesInUse

GGA_08_HorizontalDilutionOfPrecision

GGA_09_Altitude

GGA_10_AltitudeInd

GGA_11_GeoidalSeparation

GGA_12_GeoidalSeparationInd

GGA_13_AgeOfDifferentialData

GGA_14_DifferentialReferenceID

GLC_01_GRI

GLC_02_MasterTOA

GLC_03_SignalStatus1

GLC_04_TD1

GLC_05_SignalStatus2

GLC_06_TD2

GLC_07_SignalStatus3

GLC_08_TD3

GLC_09_SignalStatus4

GLC_10_TD4

GLC_11_SignalStatus5

GLC_12_TD5

GLC_13_SignalStatus6

GLL_01_Latitude

GLL_02_LatitudeInd

GLL_03_Longitude

GLL_04_LongitudeInd

GLL_05_UTC

GLL_06_Status

GLL_07_ModeIndicator

GNS_01_UTC

GNS_02_Latitude

GNS_03_LatitudeInd

GNS_04_Longitude

GNS_05_LongitudeInd

GNS_06_ModeIndicator

GNS_07_NumberOfSatellitesInUse

GNS_08_HDOP

GNS_09_AntennaAltitude

GNS_10_GeoidalSeparation

GNS_11_AgeOfDifferentialData

GNS_12_DifferentialStationID

GRS_01_UTC

GRS_02_Mode

GRS_03_RangeResidual

GRS_04_RangeResidual

GRS_05_RangeResidual

GRS_06_RangeResidual

GRS_07_RangeResidual

GRS_08_RangeResidual

GRS_09_RangeResidual

GRS_10_RangeResidual

GRS_11_RangeResidual

GRS_12_RangeResidual

GRS_13_RangeResidual

GRS_14_RangeResidual

GSA_01_Mode

GSA_02_Mode

GSA_03_ID

GSA_04_ID

GSA_05_ID

GSA_06_ID

GSA_07_ID

GSA_08_ID

GSA_09_ID

GSA_10_ID

GSA_11_ID

GSA_12_ID

GSA_13_ID

GSA_14_ID

GSA_15_PDOP

GSA_16_HDOP

GSA_17_VDOP

GST_01_UTC

GST_02_RMSvalueOfStandardDeviation

GST_03_StandardDeviationOfSemiMajorAxis

GST_04_StandardDeviationOfSemiMinorAxis

GST_05_OrientationOfSemiMajorAxis

GST_06_StandardDeviationOfLatitude

GST_07_StandardDeviationOfLongitude

GST_08_StandardDeviationOfAltitude

GSV_01_NumberOfMessages

GSV_02_MessageNumber

GSV_03_NumberOfSatellitesInView

GSV_04_SET1_SatelliteID

GSV_05_SET1_Elevation

GSV_06_SET1_Azimuth

GSV_07_SET1_SNR

GSV_08_SET2_SatelliteID

GSV_09_SET2_Elevation

GSV_10_SET2_Azimuth

GSV_11_SET2_SNR

GSV_12_SET3_SatelliteID

GSV_13_SET3_Elevation

GSV_14_SET3_Azimuth

GSV_15_SET3_SNR

GSV_16_SET4_SatelliteID

GSV_17_SET4_Elevation

GSV_18_SET4_Azimuth

GSV_19_SET4_SNR

HDG_01_MagneticHeading

HDG_02_MagneticDeviation

HDG_03_MagneticDeviationInd

HDG_04_MagneticVariation

HDG_05_MagneticVariation

HDM_01_MagneticHeading

HDM_02_MagneticHeadingInd

HDT_01_Heading

HDT_02_HeadingInd

HMR_01_HeadingSensor1ID

HMR_02_HeadingSensor2ID

HMR_03_DifferenceLimit

HMR_04_HeadingSensorDifference

HMR_05_WarningFlag

HMR_06_HeadingReadingSensor1

HMR_07_StatusSensor1

HMR_08_TypeSensor1

HMR_09_DeviationSensor1

HMR_10_DeviationSensor1Ind)

HMR_11_HeadingReadingSensor

HMR_12_StatusSensor2

HMR_13_TypeSensor2

HMR_14_DeviationSensor2

HMR_15_DeviationSensor2Ind)

HMR_16_Variation

HMR_17_VariationInd)

HMS_01_HeadingSensor1ID

HMS_02_HeadingSensor2ID

HMS_03_MaximumDifference

HSC_01_CommandedHeading

HSC_02_CommandedHeadingInd

HSC_03_CommandedHeadingMagnetic

HSC_04_CommandedHeadingMagneticInd

HTC_01_Override

HTC_02_CommandedRudderAngle

HTC_03_CommandedRudderDirection

HTC_04_SelectedSteeringMmode

HTC_05_TurnMode

HTC_06_CommandedRudderLimit

HTC_07_CommandedOffHeadingLimit

HTC_08_CommandedRadiusOfTurn

HTC_09_CommandedRateOfTurn

HTC_10_CommandedHeadingToSteer

HTC_11_CommandedOffTrackLimit

HTC_12_CommandedTrack

HTC_13_HeadingReferenceInUse

HTD_01_Override

HTD_02_CommandedRudderAngle

HTD_03_CommandedRudderDirection

HTD_04_SelectedSteeringMode

HTD_05_TurnMode

HTD_06_CommandedRudderLimit

HTD_07_CommandedOffHeadingLimit

HTD_08_CommandedRadiusOfTurn

HTD_09_CommandedRateOfTurn

HTD_10_CommandedHeadingToSteer

HTD_11_CommandedOffTrackLimit

HTD_12_CommandedTrack

HTD_13_HeadingReferenceInUse

HTD_14_RudderStatus

HTD_15_OffHeadingStatus

HTD_16_OffTrackstatus

HTD_17_VesselHeading

LCD_01_GRI

LCD_02_MasterSNR

LCD_03_MasterECD

LCD_04_Secondary1_SNR

LCD_05_Secondary1_ECD

LCD_06_Secondary2_SNR

LCD_07_Secondary2_ECD

LCD_08_Secondary3_SNR

LCD_09_Secondary3_ECD

LCD_10_Secondary4_SNR

LCD_11_Secondary4_ECD

LCD_12_Secondary5_SNR

LCD_13_Secondary5_ECD

MDA_01_BarometricPressureInchesOfMercury

MDA_02_BarometricPressureInchesOfMercuryInd

MDA_03_Barometric pressureBars

MDA_04_Barometric pressureBarsInd

MDA_05_AirTemperature

MDA_06_AirTemperatureInd

MDA_07_WaterTemperature

MDA_08_WaterTemperatureInd

MDA_09_RelativeHumidity

MDA_10_AbsoluteHumidity

MDA_11_DewPoint

MDA_12_DewPointInd

MDA_13_WindDirectionTrue

MDA_14_WindDirectionTrueInd

MDA_15_WindDirectionMagnetic

MDA_16_WindDirectionMagneticInd

MDA_17_WindSpeedKnots

MDA_18_WindSpeedKnotsInd

MDA_19_WindSpeedMs

MDA_20_WindSpeedMsInd

MLA_01_TotalNumberOfMessages

MLA_02_MessageNumber

MLA_03_SatelliteID

MLA_04_CalendarDay

MLA_05_GeneralizedHealth

MLA_06_Eccentricity

MLA_07_DOT

MLA_08_ArgumentOfPerigee

MLA_09_SystemTimeScaleCorrectionMSB

MLA_10_CorrectionOfAverageValueDraconitic

MLA_11_TimeOfAscensionNode

MLA_12_GreenwichLongitude

MLA_13_CorrectionToAverageValueInclination

MLA_14_SystemTimeScaleCorrectionLSB

MLA_15_CourseValueOfTimeScaleShift

MSK_01_BeaconFrequency

MSK_02_Auto_Manual_Frequency

MSK_03_BeaconBitRate

MSK_04_Auto_Manual_BitRate

MSK_05_IntervalForSending

MSK_06_ChannelNumber

MSS_01_SignalStrength

MSS_02_SNR

MSS_03_BeaconFrequency

MSS_04_BeaconBitRate

MSS_05_ChannelNumber

MTW_01_Temperature

MTW_02_TemperatureInd

MWD_01_WindDirection

MWD_02_WindDirectionInd

MWD_03_WindDirectionMagnetic

MWD_04_WindDirectionMagneticInd

MWD_05_WindSpeedKnots

MWD_06_WindSpeedKnotsInd

MWD_07_WindSpeedMs

MWD_08_WindSpeedMsInd

MWV_01_WindAngle

MWV_02_Reference

MWV_03_WindSpeed

MWV_04_WindSpeedInd

MWV_05_Status

NMEA_Altitude

NMEA_Course

NMEA_Latitude

NMEA_LatitudeInd

NMEA_Longitude

NMEA_LongitudeInd

NMEA_SpeedKnots

NMEA_UTC

OSD_01_Heading

OSD_02_HeadingStatus

OSD_03_VesselCourse

OSD_04_CourseReference

OSD_05_VesselSpeed

OSD_06_SpeedReference

OSD_07_VesselSet

OSD_08_VesselDrift

OSD_09_SpeedUnits

RMA_01_Status

RMA_02_Latitude

RMA_03_LatitudeInd

RMA_04_Longitude

RMA_05_LongitudeInd

RMA_06_TimeDifferenceA

RMA_07_TimeDifferenceB

RMA_08_SpeedOverGroundKnots

RMA_09_CourseOverGround

RMA_10_MagneticVariation

RMA_11_MagneticVariationInd

RMA_12_ModeIndicator

RMB_01_DataStatus

RMB_02_CrossTrackError

RMB_03_DirectionToSteer

RMB_04_OriginWaypointID

RMB_05_DestinationwaypointID

RMB_06_DestinationwaypointLat

RMB_07_DestinationwaypointLatInd

RMB_08_DestinationWaypointLongitude

RMB_09_DestinationWaypointLongitudeInd

RMB_10_RangeToDestination

RMB_11_BearingToDestination

RMB_12_DestinationClosingVelocity

RMB_13_ArrivalStatus

RMB_14_ModeIndicator

RMC_01_UTC

RMC_02_Status

RMC_03_Latitude

RMC_04_LatitudeInd

RMC_05_Longitude

RMC_06_LongitudeInd

RMC_07_SpeedOverGround

RMC_08_CourseOverGround

RMC_09_Date

RMC_10_MagneticVariation

RMC_11_MagneticVariationInd

RMC_12_ModeIndicator

ROT_01_RateOfTurn

ROT_02_Status

RPM_01_SourceShaftEngine

RPM_02_EngineOfShaftNumber

RPM_03_Speed

RPM_04_PropellerPitch

RPM_05_Status

RSA_01_StarboardRudderSensor

RSA_02_StatusRudderSensor)

RSA_03_PortRudderSensor

RSA_04_StatusPortRudderSensor)

RSD_01_Origin1Range

RSD_02_Origin1Bearing

RSD_03_VariableRangeMarker1

RSD_04_BearingLine1

RSD_05_Origin2Range

RSD_06_Origin2Bearing

RSD_07_VRM2

RSD_08_EBL2

RSD_09_CursorRange

RSD_10_CursorBearing

RSD_11_RangeScale

RSD_12_RangeScaleUnits

RSD_13_DisplayRotation

RTE_01_TotalNumberOfMessages

RTE_02_MessageNumber

RTE_03_MessageMode

RTE_04_RouteIdentifier

RTE_05_WaypointIdentifier1

RTE_06_WaypointIdentifier2

RTE_07_WaypointIdentifier3

RTE_08_WaypointIdentifier4

RTE_09_WaypointIdentifier5

RTE_10_WaypointIdentifier6

RTE_11_WaypointIdentifier7

RTE_12_WaypointIdentifier8

RTE_13_WaypointIdentifier9

RTE_14_WaypointIdentifier10

SFI_01_TotalNumberOfMessages

SFI_02_MessageNumber

SFI_03_1stFrequency

SFI_04_1stMode

SFI_05_2ndFrequency

SFI_06_2ndMode

SFI_07_3rdFrequency

SFI_08_3rdMode

SFI_09_4thFrequency

SFI_10_4thMode

SFI_11_5thFrequency

SFI_12_5thMode

SFI_13_6thFrequency

SFI_14_6thMode

STN_01_TalkerID

TLB_01_TargetNumber

TLB_02_LabelAssigned

TLB_03_TargetNumber1

TLB_04_LabelAssigned1

TLB_05_TargetNumber2

TLB_06_LabelAssigned2

TLB_07_TargetNumber3

TLB_08_LabelAssigned3

TLB_09_TargetNumber4

TLB_10_LabelAssigned4

TLB_11_TargetNumber5

TLB_12_Labelassigned5

TLB_13_TargetNumber6

TLB_14_LabelAssigned6

TLB_15_TargetNumber7

TLB_16_LabelAassigned7

TLB_17_TargetNumber8

TLB_18_LabelAssigned8

TLB_19_TargetNumberReported

TLB_20_TargetLabelAssigned

TLL_01_TargetNumber

TLL_02_TargetLatitude

TLL_03_TargetLatitudeInd

TLL_04_TargetLongitude

TLL_05_TargetLongitudeInd

TLL_06_TargetName

TLL_07_UTC

TLL_08_TargetStatus

TLL_09_ReferenceTarget

TTM_01_TargetNumber

TTM_02_TargetDistance

TTM_03_Bearing

TTM_04_BearingInd

TTM_05_TargetSpeed

TTM_06_TargetCourse

TTM_07_TargetCourseInd

TTM_08_DistanceOfClosestPoint

TTM_09_TimeToCPA

TTM_10_SpeedAndDistanceUnits

TTM_11_TargetName

TTM_12_TargetStatus

TTM_13_ReferenceTarget

TTM_14_UTC

TTM_15_TypeOfAcquisition

TXT_01_TotalNumberOfMessages

TXT_02_MessageNumber

TXT_03_TextIdentifier

TXT_04_TextMessage

VBW_01_LongitudinalWaterSpeed

VBW_02_TransverseWaterSpeed

VBW_03_StatusWaterSpeed

VBW_04_LongitudinalGroundSpeed

VBW_05_TransverseGroundSpeed

VBW_06_StatusGroundSpeed

VBW_07_SternTransverseWaterSpeed

VBW_08_StatusSternWaterSpeed

VBW_09_SternTransverseGroundSpeed

VBW_10_StatusSternGroundSpeed

VDR_01_Direction

VDR_02_DirectionInd

VDR_03_DirectionMagnetic

VDR_04_DirectionMagneticInd

VDR_05_CurrentSpeed

VDR_06_CurrentspeedInd

VHW_01_Heading

VHW_02_HeadingInd

VHW_03_HeadingMagnetic

VHW_04_HeadingMagneticInd

VHW_05_SpeedKnots

VHW_06_SpeedKnotsInd

VHW_07_SpeedKmh

VHW_08_SpeedKmhInd

VLW_01_TotalCumulativeDistance

VLW_02_TotalCumulativeDistanceInd

VLW_03_DistanceSinceReset

VLW_04_DistanceSinceResetInd

VPW_01_SpeedKnots

VPW_02_SpeedKnotsInd)

VPW_03_SpeedMs

VPW_04_SpeedMsInd

VTG_01_CourseOverGround

VTG_02_CourseOverGroundInd

VTG_03_CourseOverGroundMagnetic

VTG_04_CourseOverGroundMagneticInd

VTG_05_SpeedOverGroundKnots

VTG_06_SpeedOverGroundKnotsInd

VTG_07_SpeedOverGroundKmh

VTG_08_SpeedOverGroundKmhInd

VTG_09_ModeIndicator

VWR_01_MeasuredWindAngle

VWR_02_VesselHeading

VWR_03_MeasuredWindSpeed

VWR_04_MeasuredWindSpeedInd

VWR_05_WindSpeedMeters

VWR_06_WindSpeedMetersInd

VWR_07_WindSpeedKmh

VWR_08_WindSpeedKmhInd

VWT_01_CalculatedWindAngle

VWT_02_VesselHeading

VWT_03_CalculatedWindSpeed

VWT_04_CalculatedWindSpeedInd

VWT_05_WindSpeedMeters

VWT_06_WindSpeedMetersInd

VWT_07_WindSpeedKmh

VWT_08_WindSpeedKmhInd

WCV_01_VelocityComponent

WCV_02_VelocityComponentInd

WCV_03_WaypointIdentifier

WCV_04_ModeIndicator

WNC_01_DistanceMiles

WNC_02_DistanceMilesInd

WNC_03_DistanceKm

WNC_04_DisttanceKmInd

WNC_05_WaypointIdentifierFrom

WNC_06_WaypointIdentifierTo

WPL_01_WaypointLatitude

WPL_02_WaypointLatitudeInd

WPL_03_WaypointLongitude

WPL_04_WaypointLongitudeInd

WPL_05_WaypointIdentifier

XDR_01_Transducer1Type

XDR_02_Measurmnt1Data

XDR_03_UnitsOfMeasure1

XDR_04_Transducer1

XDR_05_Transducer2Type

XDR_06_Measurment2Data

XDR_07_UnitsOfMeasure2

XDR_08_Transducer2

XDR_09_Transducer3Type

XDR_10_Measurment3Data

XDR_11_UnitsOfMeasure3

XDR_12_Transducer3

XDR_13_Transducer4Type

XDR_14_Measurment4Data

XDR_15_UnitsOfMeasure4

XDR_16_Transducer4

XDR_17_Transducer5Type

XDR_18_Measurment5Data

XDR_19_UnitsOfMeasure5

XDR_20_Transducer5

XDR_21_Transducer6Type

XDR_22_Measurment6Data

XDR_23_UnitsOfMeasure6

XDR_24_Transducer6

XDR_25_Transducer7Type

XDR_26_Measurment7Data

XDR_27_UnitsOfMeasure7

XDR_28_Transducer7

XDR_29_Transducer8Type

XDR_30_Measurment8Data

XDR_31_UnitsOfMeasure8

XDR_32_Transducer8

XTE_01_Status1

XTE_02_Status2

XTE_03_MagnitudeOfCrossTrackError

XTE_04_DirectionToSteer

XTE_05_Units

XTE_06_ModeIndicator

XTR_01_MagnitudeOfCrossTrackError

XTR_02_DirectionToSteer

XTR_03_Units

ZDA_01_UTC

ZDA_02_Day

ZDA_03_Month

ZDA_04_Year

ZDA_05_LocalZoneHours

ZDA_06_LocalZoneMinutes

ZDL_01_TimeToPoint

ZDL_02_DistanceToPoint

ZDL_03_TypeOfPoint

ZFO_01_UTC

ZFO_02_ElapsedTime

ZFO_03_OriginWaypointID

ZTG_01_UTC

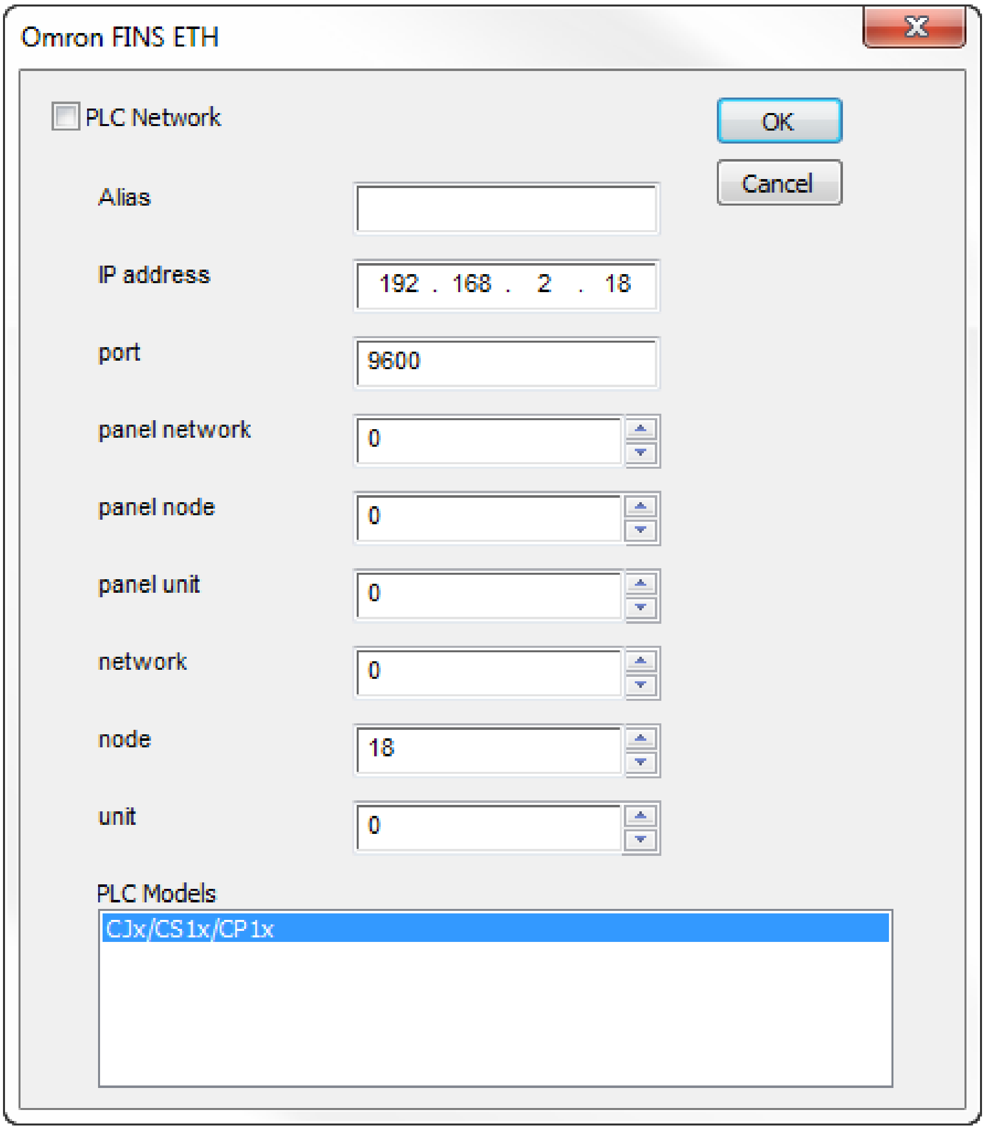
ZTG_02_TimeToGo

ZTG_03_DestinationWaypointID

Omron FINS ETH

This driver supports the FINS protocol via Ethernet connection. For a list of models that support the FINS Communications Service, refer to the manufacturer's website.

Protocol Editor Settings



□ PLC Network

Alias:

IP address: 192 . 168 . 2 . 18

port: 9600

panel network: 0

panel node: 0

panel unit: 0

network: 0

node: 18

unit: 0

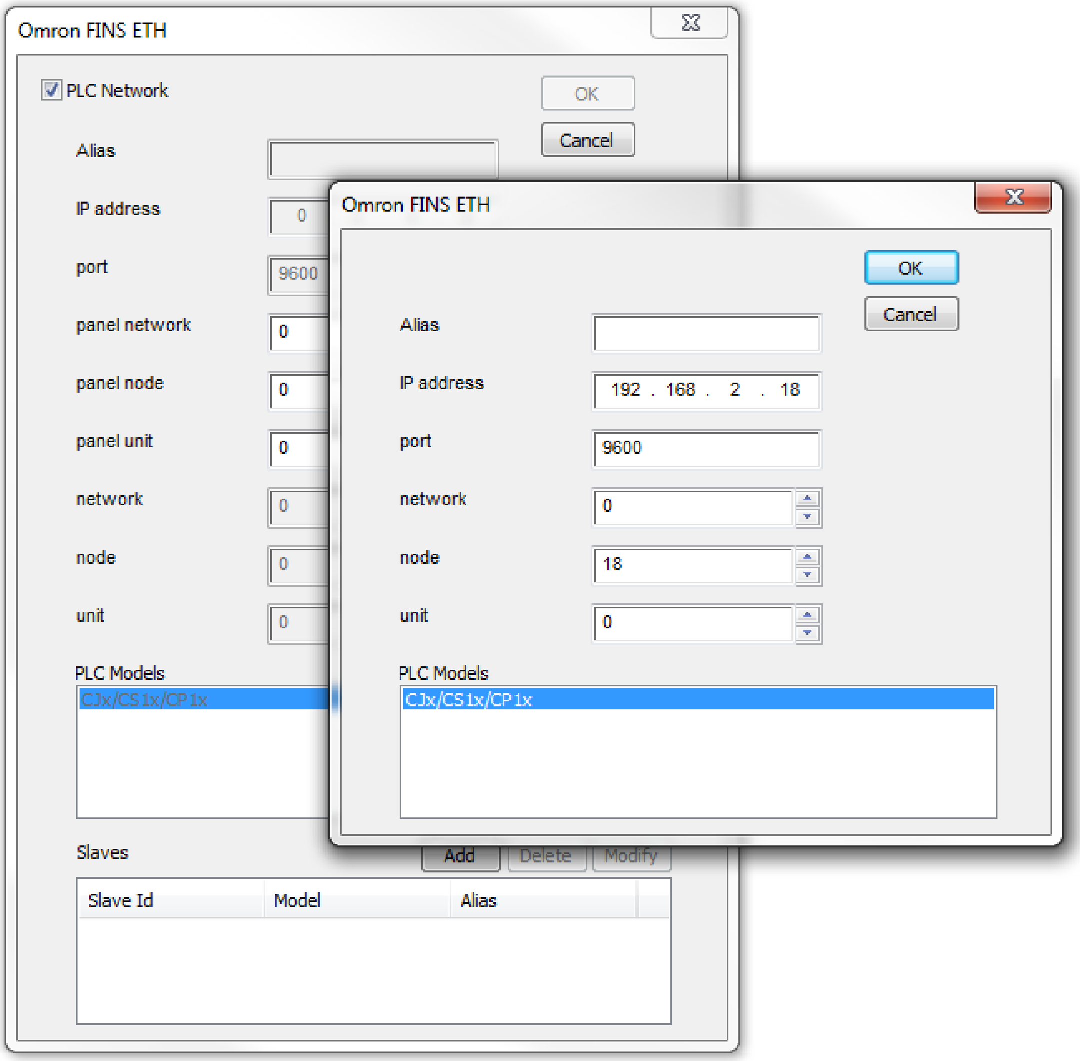
PLC Models: CJx/CS1x/CP1x

Element	Description
Alias	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
IP address	The Ethernet IP address of the controller connected to the operator panel
Port	Defines the port number used in the communication with the PLC. The UDP Port number must

Element	Description
	match the value specified in the PLC configuration; the default value is 9600. Most applications will use the default value.
Network Node Unit	Parameters that define the FINS address of the device. There is a conversion rule to determine the IP address of a device starting from the FINS address in the Omron network. When using the FINS communication service, it is necessary to specify the node addressing according to the FINS addressing scheme. Even in this case, data must be sent and received on the Ethernet network using IP addresses. Therefore, IP addresses are converted from FINS addresses. There are three ways to convert the FINS addresses into the corresponding IP address; they are: • Automatic generation (default) • IP address table • Combined method (uses Automatic and IP address table) The Omron documentation contains all the details related to determine the IP address of the controller depending on the FINS address assigned to it. The next chapter shows an example of controller configuration based on IP address table.
Panel Network **Panel Node** **Panel Unit**	The Panel Network/Node/Unit parameters assigned to HMI should be compatible with the ones assigned in the Omron network to the PLC: • Network Number must match the one specified for the PLC • Node Number should match the last number of the IP address of the HMI; in the figure above the panel has been configured with IP address 192.168.2.15. • Unit represent the possible different network cards over the same node; for the HMI should be always set to zero since there is always only one communication unit.

The protocol supports the connections to multiple controllers.

To enable this, check the "PLC Network" check box and provide the configuration per each node.



☑ PLC Network

Alias:
IP address: 0
port: 9600
panel network: 0
panel node: 0
panel unit: 0
network: 0
node: 0
unit: 0
PLC Models: CJx/CS1x/CP1x
Slaves

Omron FINS ETH

Alias:
IP address: 192 . 168 . 2 . 18
port: 9600
network: 0
node: 18
unit: 0
PLC Models: CJx/CS1x/CP1x

Controller Settings

PLC must be properly configured to handle the communication with HMI.

Below an example of configuration based on a real scenario.

Configuration windows in this chapter are depending on PLC model.
Following lines must be used as guidelines for any specific configuration.

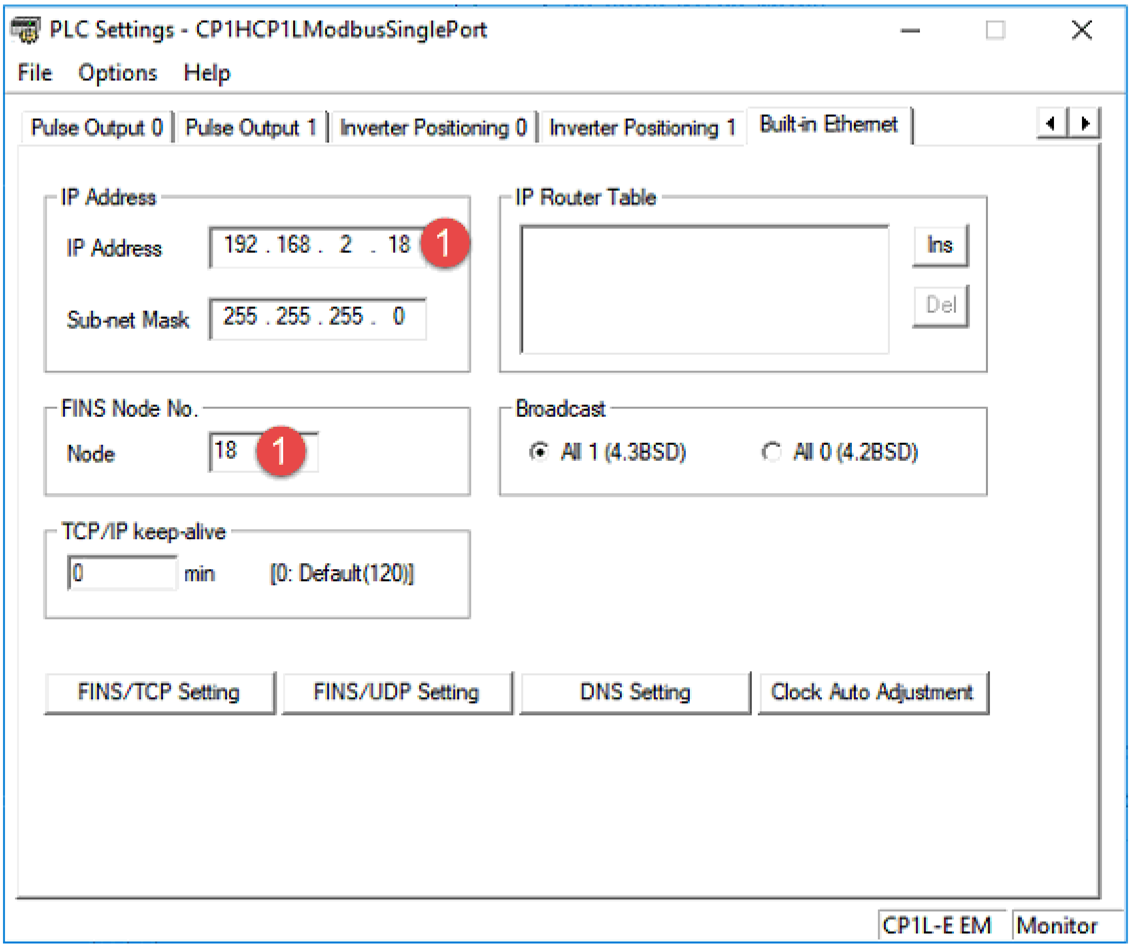
Example Setup

HMI IP address = 192.168.2.16

PLC IP address = 192.168.2.18

In Ethernet configuration Tab:

1. Make sure that last number of IP address is the same of FINS Node No.



PLC Settings - CP1HCP1LModbusSinglePort

File Options Help

Pulse Output 0 | Pulse Output 1 | Inverter Positioning 0 | Inverter Positioning 1 | Built-in Ethernet

IP Address

IP Address 192 . 168 . 2 . 18 1

Sub-net Mask 255 . 255 . 255 . 0

IP Router Table

FINS Node No.

Node 18 1

Broadcast

● All 1 (4.3BSD) ○ All 0 (4.2BSD)

TCP/IP keep-alive

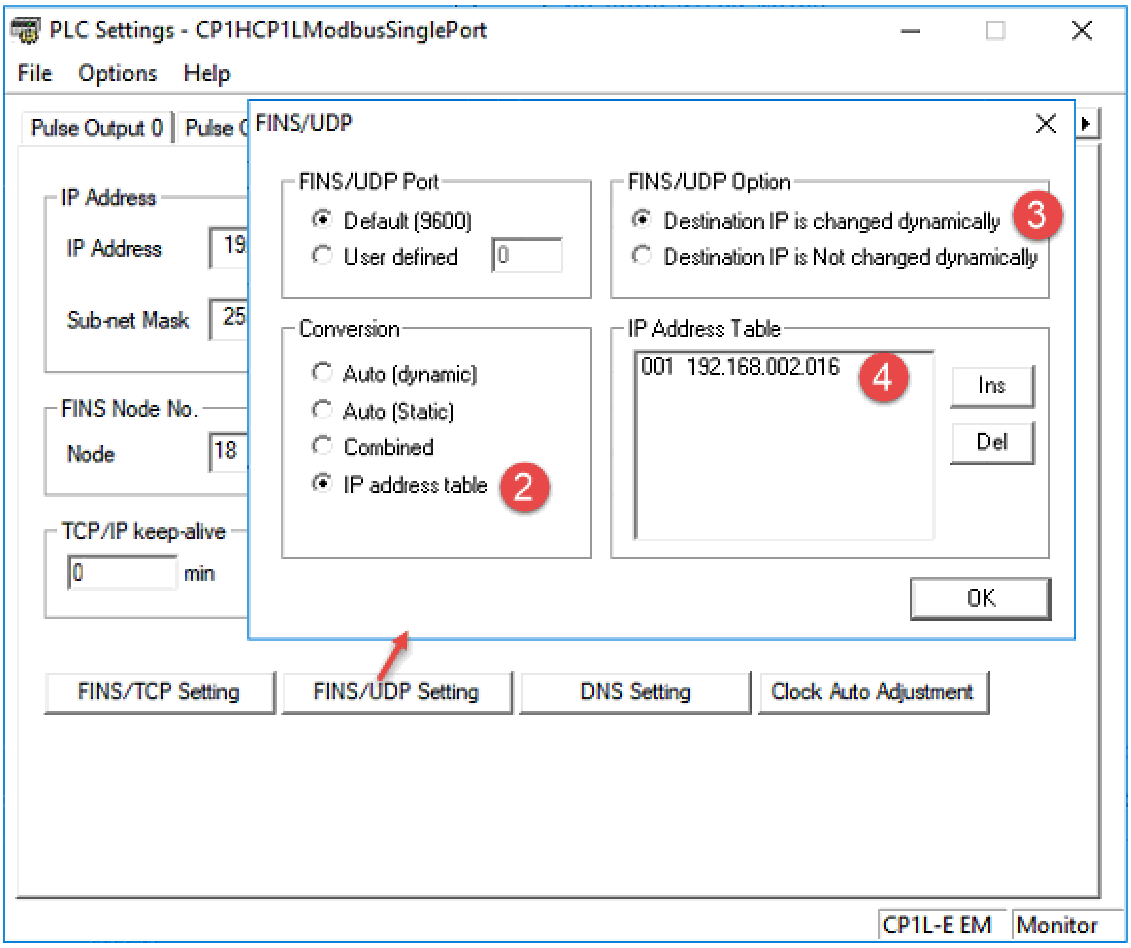
0 min [0: Default(120)]

In FINS/UDP Setting

2. Set Conversion to "IP address table"

3. Set FINS/UDP Options to "Destination IP is changed dynamically"

4. Insert HMI IP address



IP Address
IP Address 19
Sub-net Mask 25

FINS Node No.
Node 18

TCP/IP keep-alive
0 min

FINS/UDP

FINS/UDP Port
● Default (9600)
○ User defined 0

FINS/UDP Option
● Destination IP is changed dynamically
○ Destination IP is Not changed dynamically

Conversion
○ Auto (dynamic)
○ Auto (Static)
○ Combined
● IP address table

IP Address Table
001 192.168.002.016

IP Address Table can contain more than one address.
In these cases make sure that index of IP addresses is consecutive:
001 192.168.002.016
002 192.168.002.017
003 192.168.002.033

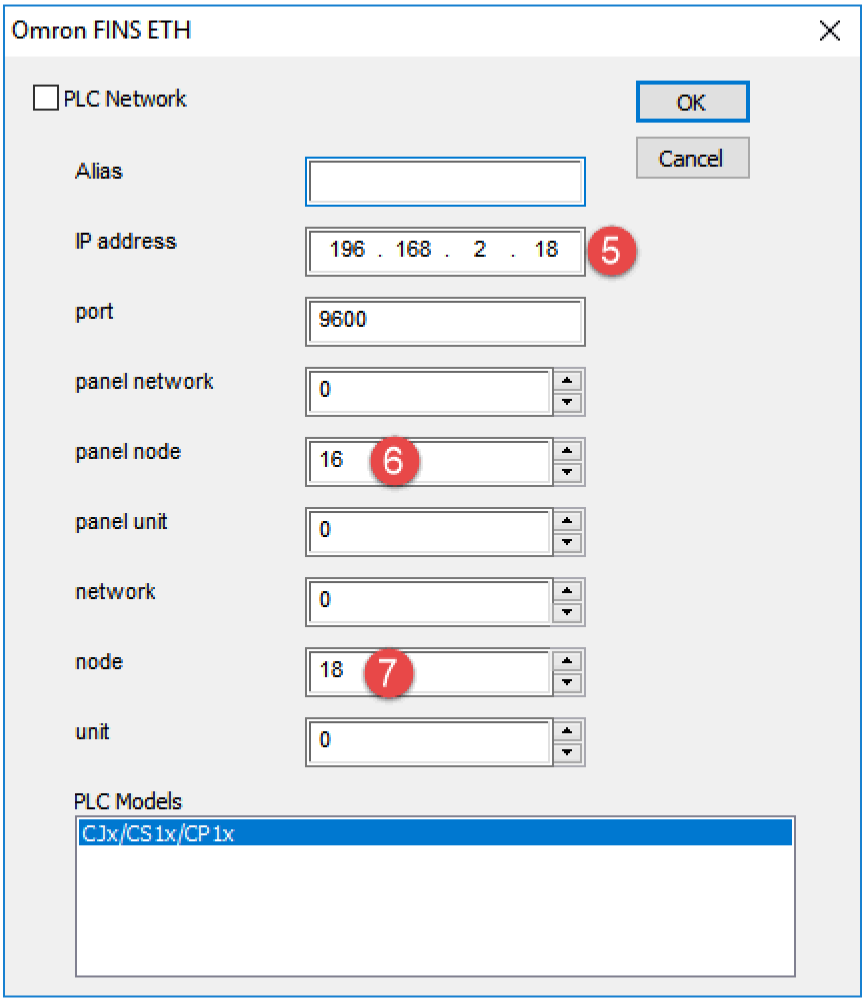
Add PC IP address in IP Address Table described above to allow communication between PLC and online Simulation.

In protocol editor

5. Set the IP address of PLC

6. Insert last number of HMI IP address in panel node parameter

7. Insert last number of PLC IP address in node parameter



Omron FINS ETH

☐ PLC Network

Alias:

IP address: 196 . 168 . 2 . 18

port: 9600

panel network: 0

panel node: 16

panel unit: 0

network: 0

node: 18

unit: 0

PLC Models: CJx/CS1x/CP1x

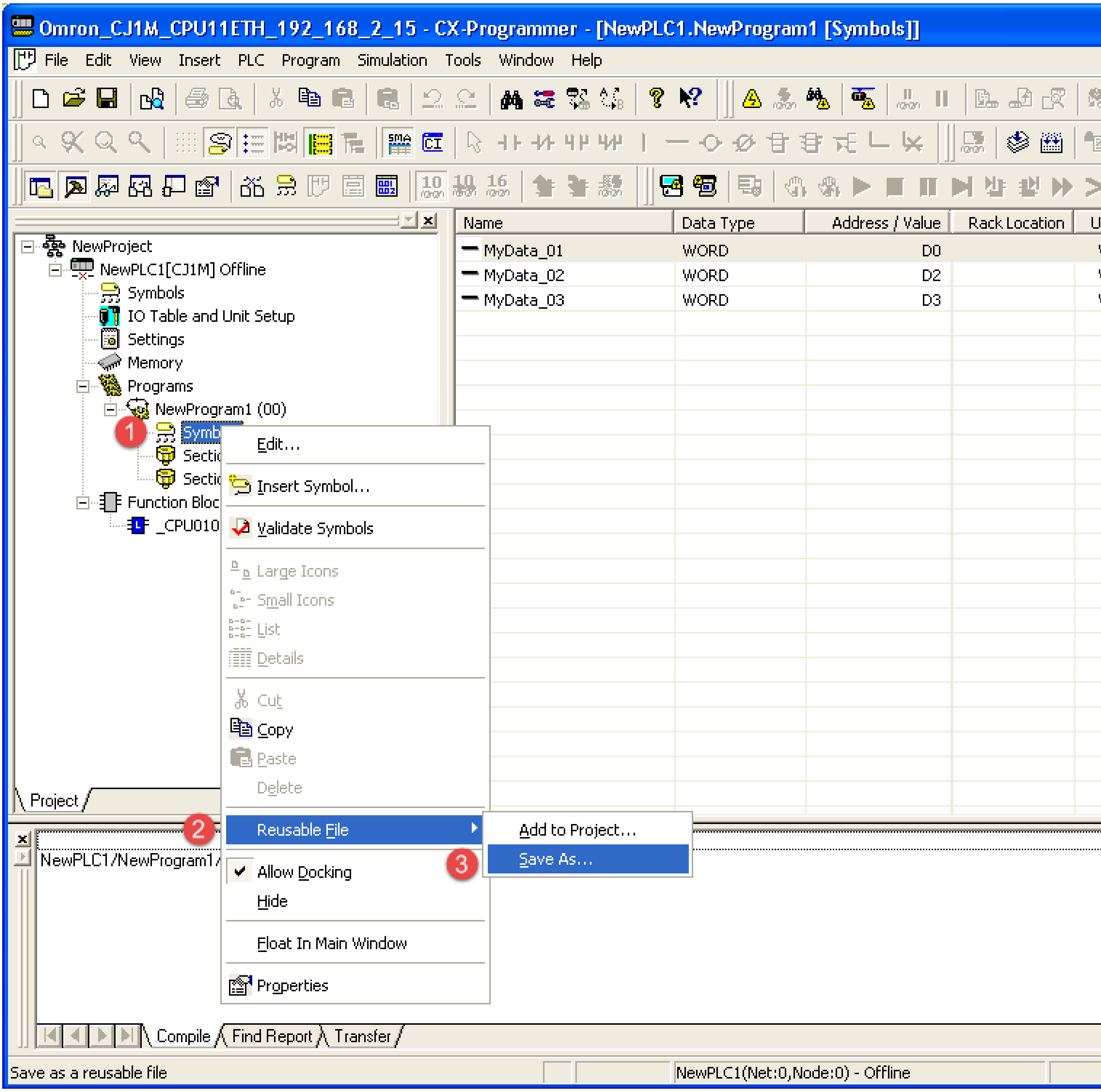
Tag Import

Exporting Tags from PLC

The Omron FINS Ethernet driver can import tag information from CX-Programmer PLC programming software. The tag import filter accepts symbol files with extension ".cxr" created by the Omron programming tool.

The ".cxr" files can be exported from the symbol table utility.

See in figure how to access the Symbol Table (if configured) from the Omron programming software.



Omron_CJ1M_CPU11ETH_192_168_2_15 - CX-Programmer - [NewPLC1.NewProgram1 [Symbols]]

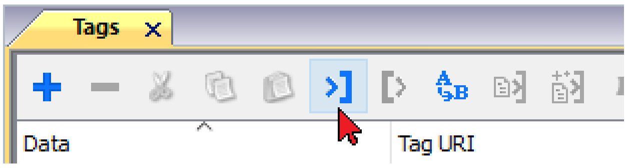
Name	Data Type	Address / Value	Rack Location
MyData_01	WORD	D0	
MyData_02	WORD	D2	
MyData_03	WORD	D3	

Save as a reusable file

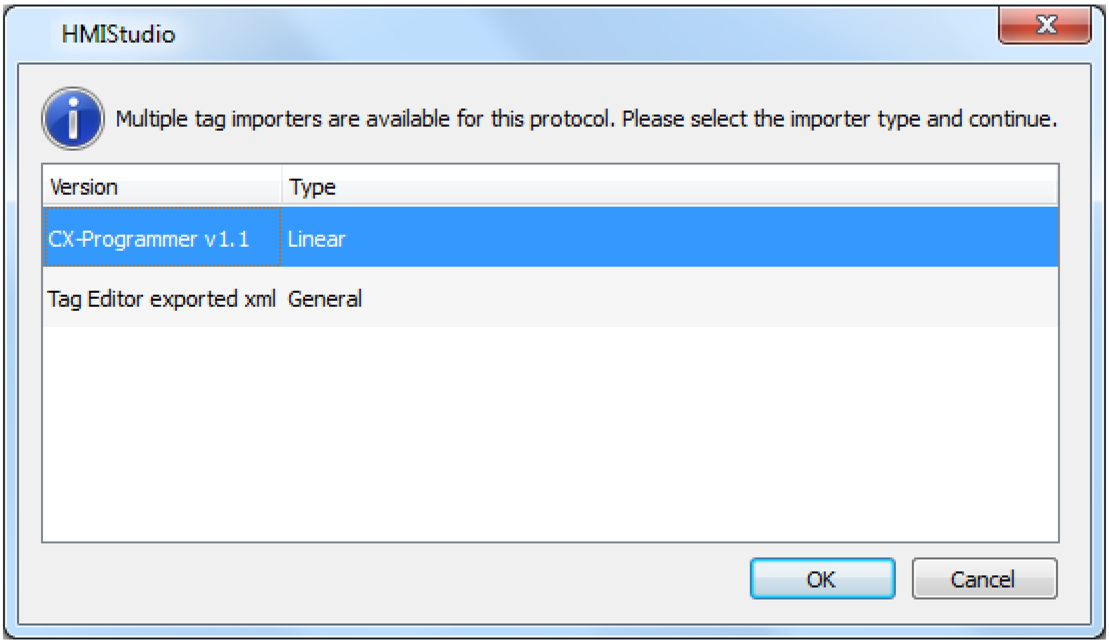
NewPLC1(Net:0,Node:0) - Offline

Importing Tags in Tag Editor

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

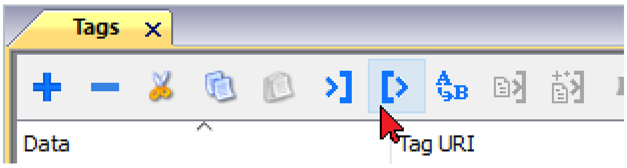


The following dialog shows which importer type can be selected.



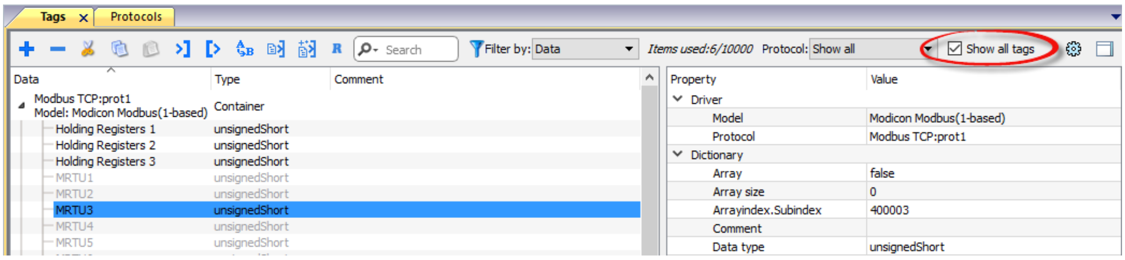
Multiple tag importers are available for this protocol. Please select the importer type and continue.

Version	Type
CX-Programmer v1.1	Linear
Tag Editor exported xml	General

Importer	Description
CX-Programmer v1.1 Linear	Requires a **.cxr**file. 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. 

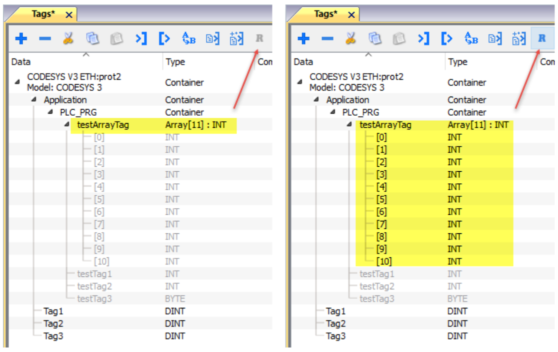
Once the importer has been selected, locate the symbol file and click **Open**.

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.



Data	Type	Comment
Modbus TCP:prot1 Model: Modicon Modbus(1-based)	Container	
Holding Registers 1	unsignedShort	
Holding Registers 2	unsignedShort	
Holding Registers 3	unsignedShort	
MRTU1	unsignedShort	
MRTU2	unsignedShort	
MRTU3	unsignedShort	
MRTU4	unsignedShort	
MRTU5	unsignedShort	

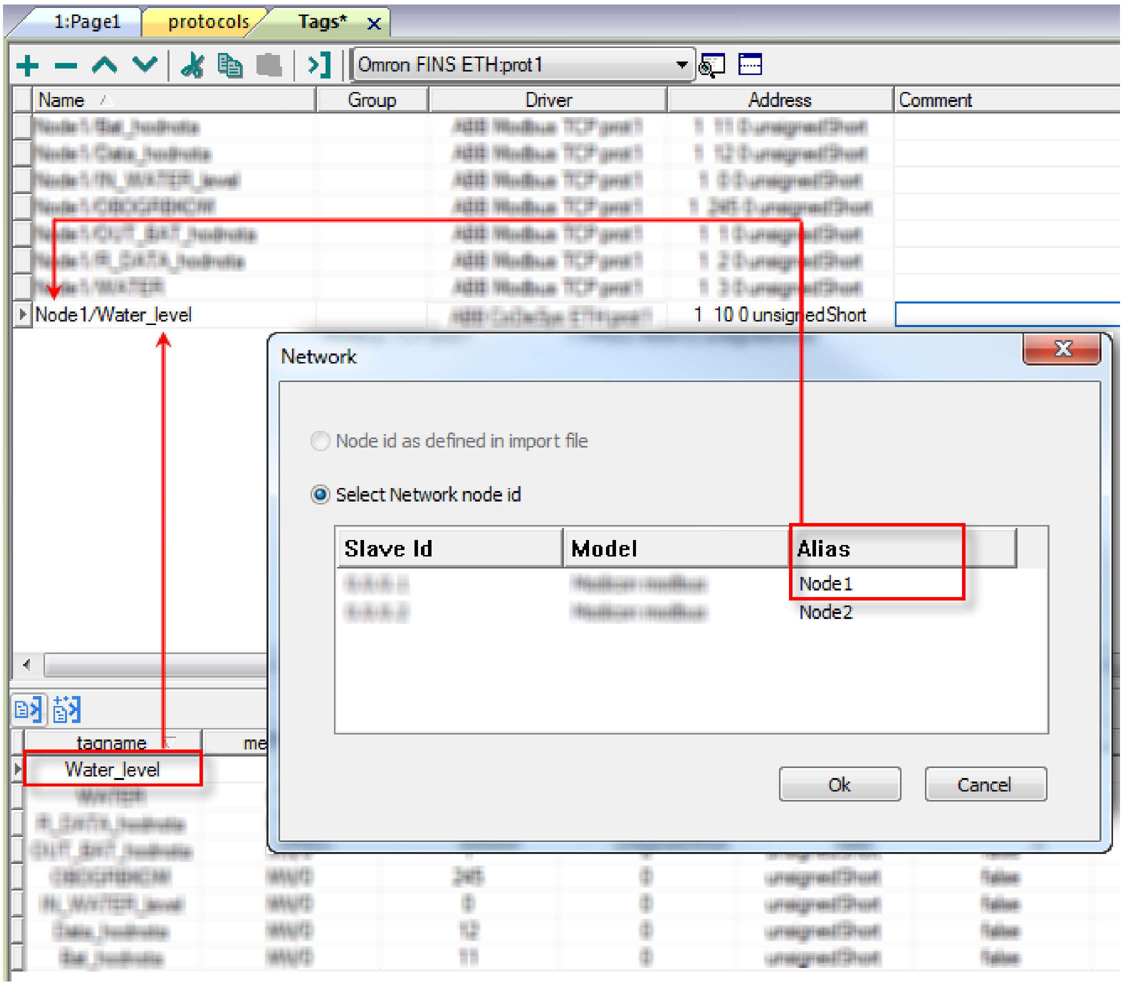
Property	Value
Driver	
Model	Modicon Modbus(1-based)
Protocol	Modbus TCP:prot1
Dictionary	
Array	false
Array size	0
Arrayindex.Subindex	400003
Comment	
Data type	unsignedShort

Toolbar item	Description
	Import Tag(s). Select tags to be imported and click on this icon to add tags from tag dictionary to the project
	Update Tag(s). Click on this icon to update the tags in the project, due a new dictionary import.
R	Check this box to import all sub-elements of a tag. Example of both checked and unchecked result:  Tags* R Data Type Con CODESYS V3 ETH:prot2 Model: CODESYS 3 Container Application Container PLC_PRG Container testArrayTag Array[11] : INT [0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] INT testTag1 INT testTag2 INT testTag3 BYTE Tag1 DINT Tag2 DINT Tag3 DINT
	Searches tags in the dictionary basing on filter combo-box item selected.

Aliasing Tag Names in Network Configurations

Tag names must be unique at project level; it often happens that the same tag names are to be used for different controller nodes (for example when the HMI is connected to two devices that are running the same application). Since tags include also the identification of the node and Tag Editor does not support duplicate tag names, the import facility in Tag Editor has an aliasing feature that can automatically add a prefix to imported tags. With this feature tag names can be done unique at project level.

The feature works when importing tags for a specific protocol. Each tag name will be prefixed with the string specified by the “Alias”. As shown in the figure below, 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.



Name	Group	Driver	Address	Comment
Node1/Water_level			1 10 0 unsignedShort	

Network

○ Node id as defined in import file

● Select Network node id

Slave Id	Model	Alias
		Node1
		Node2

Note: aliasing tag names is only available when tags can be imported. 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 to the tag name only at the moment the tags are imported using Tag Editor. 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 imported again, all tags will be imported again with the new prefix string.

Communication Status

The current communication status can be displayed using the dedicated system variables. Please refer to the User Manual for further information about available system variables and their use.

The codes supported for this communication driver are:

Error	Notes
NAK	Returned in case the controller replies with a not acknowledge; can be returned also in case the network/node/unit parameters contained in the PLC response are not matching with panel configuration
Timeout	Returned when a request is not replied within the specified timeout period; ensure the controller is connected and properly configured to get network access

Error	Notes
Invalid response	The panel did receive from the controller a response, but its format or its contents is not as expected; ensure the data programmed in the project are consistent with the controller resources. The same error can be returned also in case the PLC could not complete the processing of the panel request and sent back to the panel and invalid/not completed response.
Cnt error	Returned when a specific control character in the protocol frame received does not match with the corresponding one in the request; verify the proper settings of the controller network configuration
General Error	Error cannot be identified; should never be reported; contact technical support

Omron FINS SER

This driver supports the FINS protocol via serial connetcion. For a list of models that support the FINS Communications Service, refer to the manufacturer's website.

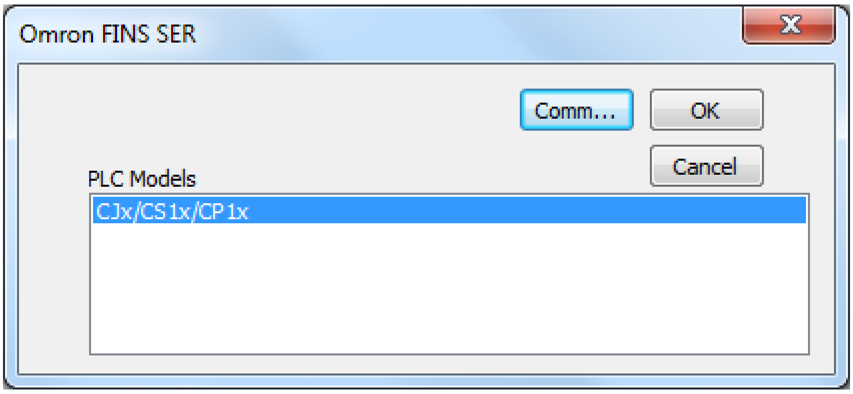
Protocol Editor Settings

Adding a protocol

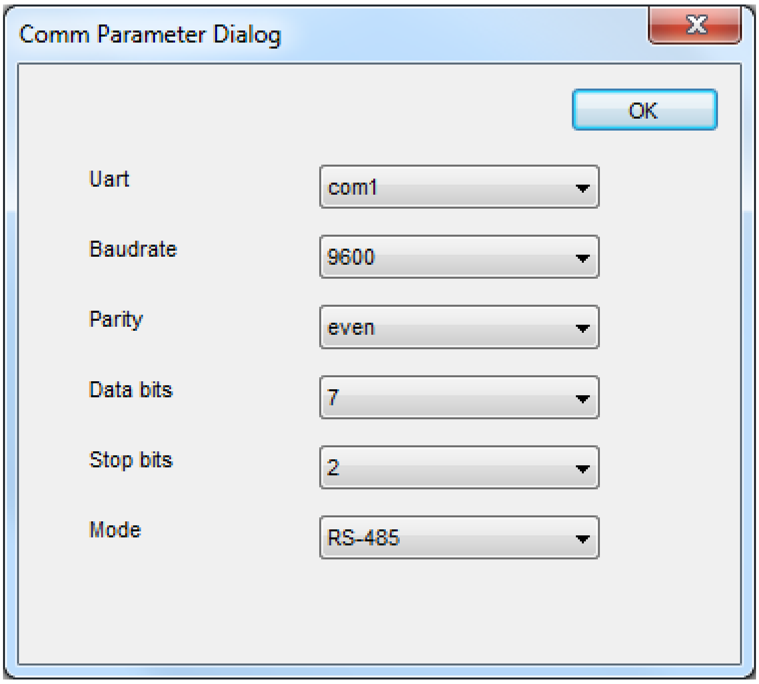
To configure the protocol:

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.



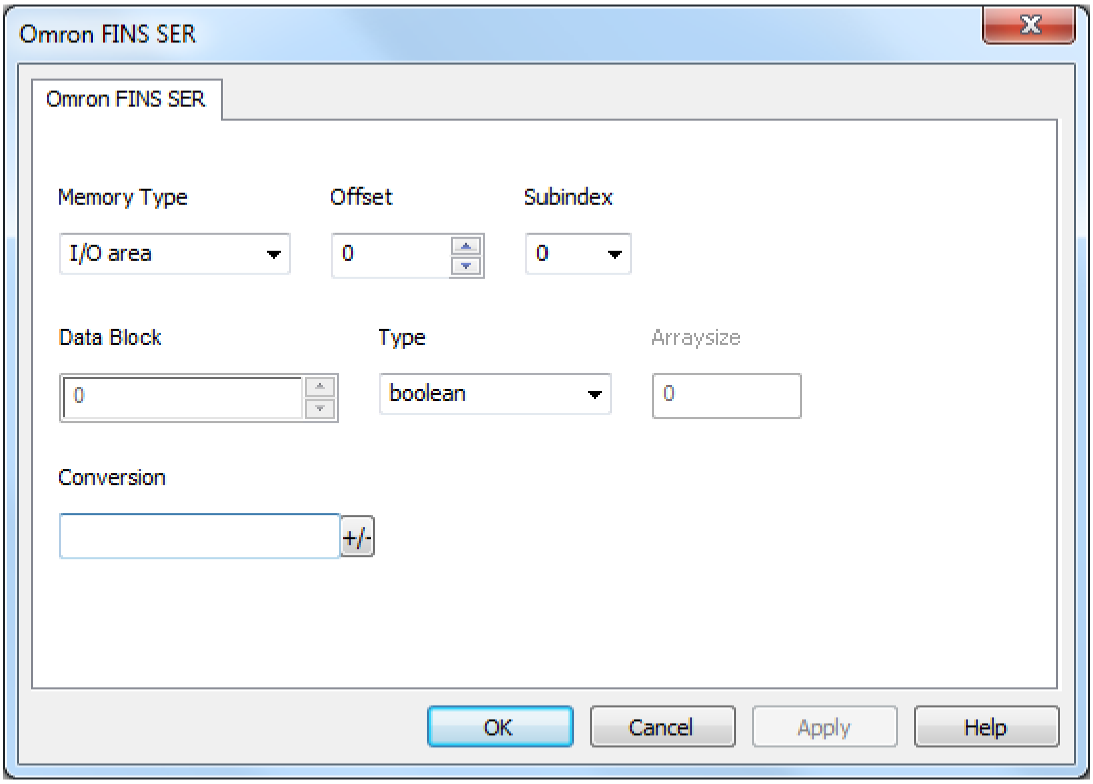
Element	Description
PLC Models	PLC models available: • CJx/CSx/CP1x
Comm...	If clicked displays the communication parameters setup dialog.

<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td></td><td>

Comm Parameter Dialog
OK
Uart: com1
Baudrate: 9600
Parity: even
Data bits: 7
Stop bits: 2
Mode: RS-485
<table>
<tr><th>Element</th><th>Parameter</th></tr>
<tr><td><b>Port</b></td><td>Serial port selection.<ul><li><b>COM1</b>: device PLC port.</li><li><b>COM2</b>: computer/printer port on panels with 2 serial ports or optional Plug-In module plugged on Slot 1/2 for panels with 1 serial port on-board.</li><li><b>COM3:</b> optional Plug-In module plugged on Slot 3/4 for panels with 1 serial port on-board.</li></ul></td></tr>
<tr><td><b>Baudrate, Parity, Data Bits, Stop bits</b></td><td>Serial line parameters.</td></tr>
<tr><td><b>Mode</b></td><td>Serial port mode. Available modes:<ul><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>
</table>
</td></tr>
</table>

Tag Editor Settings

In Tag Editor select the protocol **Omron FINS SER**.

Add a tag using [+] button. Tag setting can be defined using the following dialog:



Omron FINS SER

Memory Type: I/O area

Offset: 0

Subindex: 0

Data Block: 0

Type: boolean

Arraysize: 0

Conversion:

Element	Description
Memory Type	<table><tr><th>Memory Type</th><th>Description</th></tr><tr><td>**I/O area**</td><td>Corresponds to **CIO** resource on PLC</td></tr><tr><td>**Auxiliary area**</td><td>Corresponds to **A** resource on PLC</td></tr><tr><td>**Holding area**</td><td>Corresponds to **H** resource on PLC</td></tr><tr><td>**Timer completion flags**</td><td>Corresponds to **T** resource on PLC</td></tr><tr><td>**Timer PVs**</td><td>Corresponds to **TPV** resource on PLC</td></tr><tr><td>**DM area**</td><td>Corresponds to **D** resource on PLC</td></tr><tr><td>**Counter completion area**</td><td>Corresponds to **C** resource on PLC</td></tr><tr><td>**Counter CVs**</td><td>Corresponds to **CVS** resource on PLC</td></tr><tr><td>**EM area**</td><td>Corresponds to **E** resource on PLC</td></tr><tr><td>**Work area**</td><td>Corresponds to **W** resource on PLC</td></tr><tr><td>**Index registers**</td><td>Corresponds to **IR** resource on PLC</td></tr><tr><td>**Data registers**</td><td>Corresponds to **DR** resource on PLC</td></tr></table>
Offset	Starting address for the Tag. The possible range depend on memory type selected.

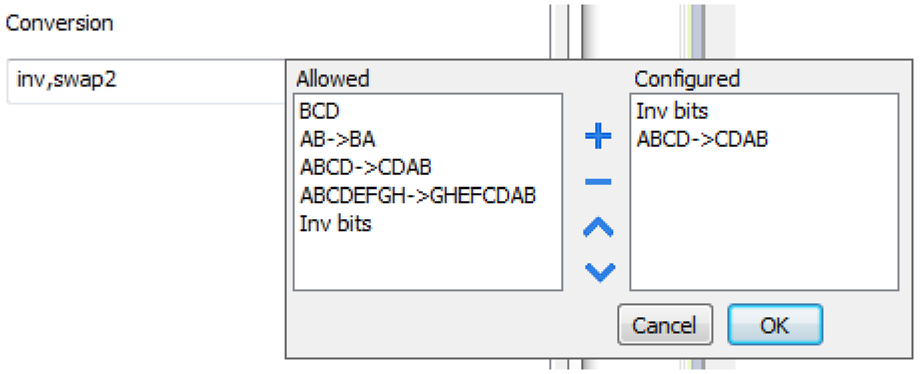
Element	Description
Subindex	This parameter allow to select a single part of the resource if the selected data type is shorter than the resource data type
Data block	Instance of resource of the PLC.
Data Type	Available data types: • **boolean** • **byte** • **short** • **int** • **unsignedByte** • **unsignedShort** • **unsignedInt** • **float** • **double** • **string** • **binary** See "Programming concepts" section in the main manual. Note: To define arrays, select one of Data Type format followed by square brackets (byte[], short[]…).

<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td><b>Arraysize</b></td><td>

- 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.
</td></tr>
<tr><td><b>Conversion</b></td><td>

Conversion to be applied to the tag.



Conversion: inv,swap2

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

<table>
<tr><th>Value</th><th>Description</th></tr>
<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 -> 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 -> 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 -</b></td><td><b>swap4</b>: Swap bytes in a double word.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td><b>&gt; GHEFCDAB</b></td><td><i>Example:</i>
32FCFF54 → 54FFFC32 (in hexadecimal format)
855441236 → 1426062386 (in decimal format)</td></tr>
<tr><td><b>ABC...NOP -> OPM...DAB</b></td><td><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)</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>
</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>
</td></tr>
</table>

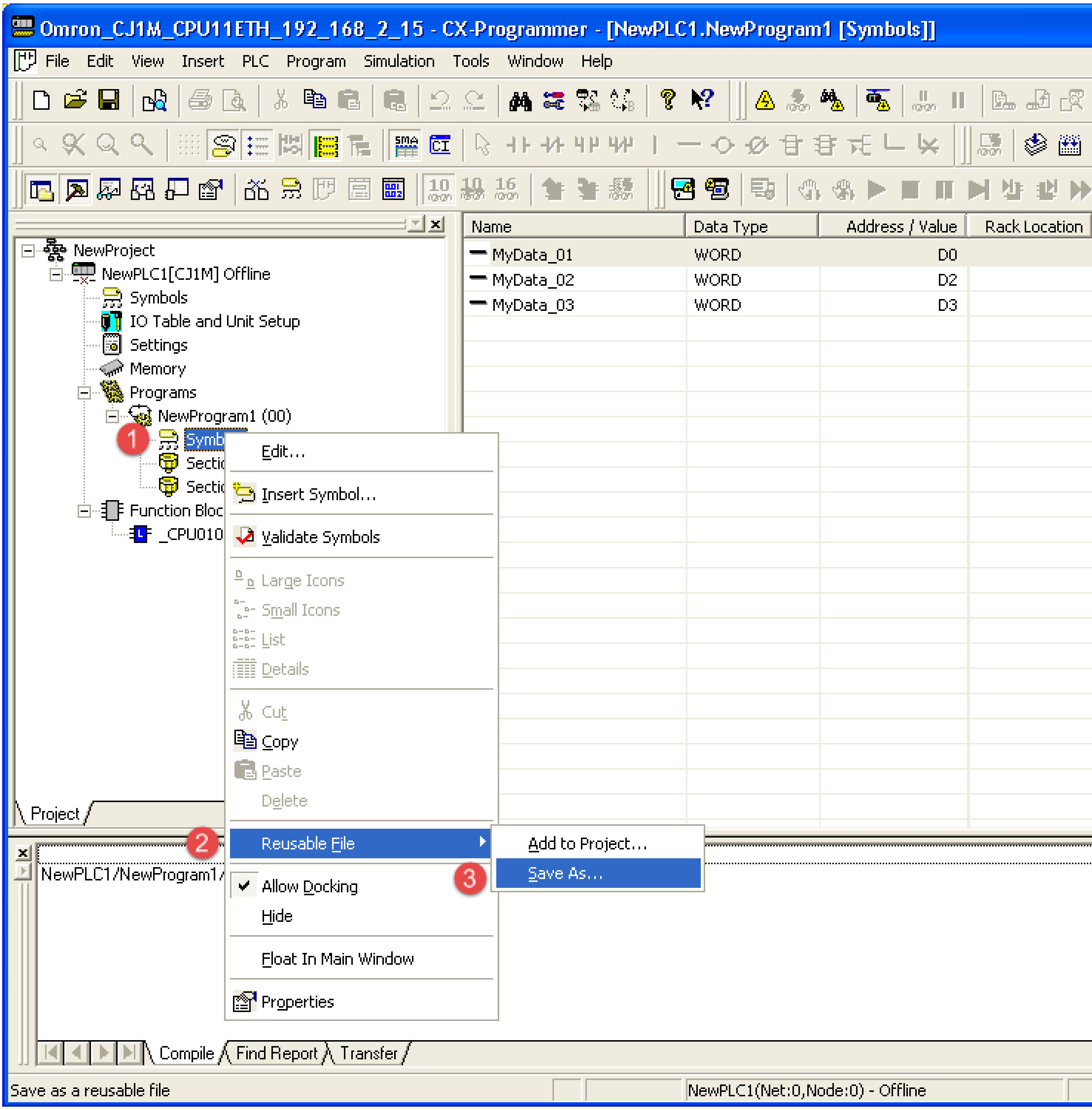
Tag Import

Exporting Tags from PLC

The Omron FINS SER driver can import tag information from CX-Programmer PLC programming software. The tag import filter accepts symbol files with extension ".cxr" created by the Omron programming tool.

The ".cxr" files can be exported from the symbol table utility.

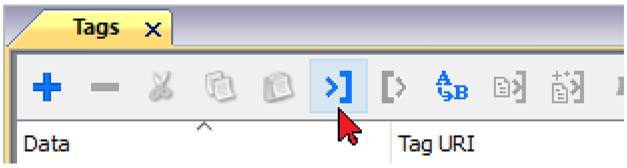
See in figure how to access the Symbol Table (if configured) from the Omron programming software.


Omron_CJ1M_CPU11ETH_192_168_2_15 - CX-Programmer - [NewPLC1.NewProgram1 [Symbols]]

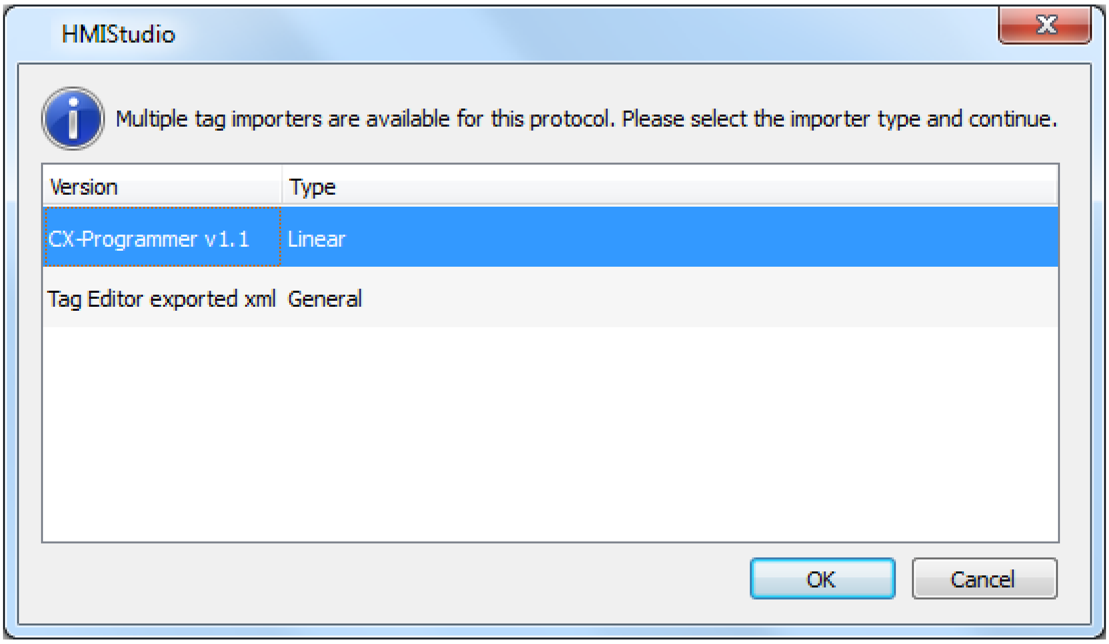
Name	Data Type	Address / Value	Rack Location
MyData_01	WORD	D0	
MyData_02	WORD	D2	
MyData_03	WORD	D3	

Importing Tags in Tag Editor

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



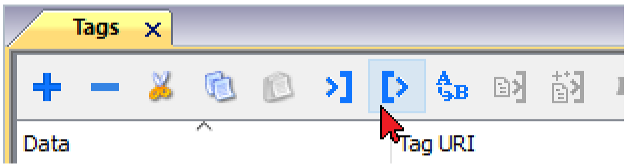
The following dialog shows which importer type can be selected.



HMIStudio

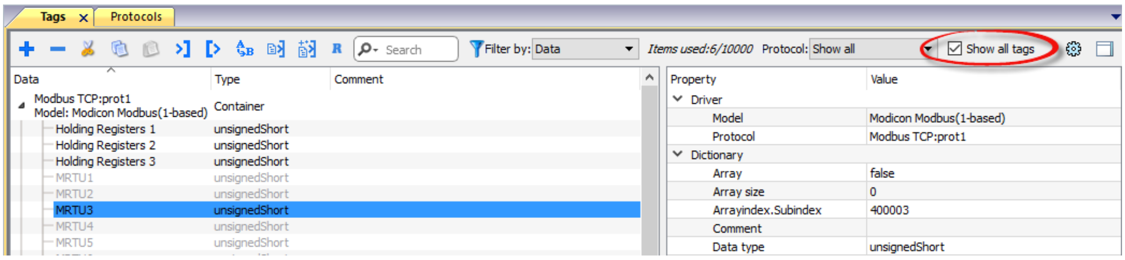
Multiple tag importers are available for this protocol. Please select the importer type and continue.

Version	Type
CX-Programmer v1.1	Linear
Tag Editor exported xml	General

Importer	Description
CX-Programmer v1.1 Linear	Requires a **.cxr** file. 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**.

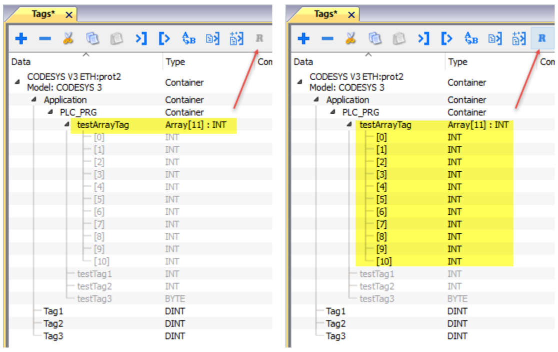
Tags included in the symbol file are listed in the tag dictionary. The tag dictionary is displayed at the bottom of the screen.



Filter by: Data
Items used:6/10000
Protocol: Show all
☑ Show all tags

Data	Type	Comment
Modbus TCP:prot1 Model: Modicon Modbus(1-based)	Container	
Holding Registers 1	unsignedShort	
Holding Registers 2	unsignedShort	
Holding Registers 3	unsignedShort	
MRTU1	unsignedShort	
MRTU2	unsignedShort	
MRTU3	unsignedShort	
MRTU4	unsignedShort	
MRTU5	unsignedShort	

Property	Value
Driver	
Model	Modicon Modbus(1-based)
Protocol	Modbus TCP:prot1
Dictionary	
Array	false
Array size	0
Arrayindex.Subindex	400003
Comment	
Data type	unsignedShort

Toolbar item	Description
	Import Tag(s). Select tags to be imported and click on this icon to add tags from tag dictionary to the project
	Update Tag(s). Click on this icon to update the tags in the project, due a new dictionary import.
R	Check this box to import all sub-elements of a tag. Example of both checked and unchecked result: 
Search Filter by: Tag name	Searches tags in the dictionary basing on filter combo-box item selected.

Tags* ×

Data	Type	Con
CODESYS V3 ETH:prot2 Model: CODESYS 3	Container	
Application	Container	
PLC_PRG	Container	
testArrayTag	Array[11] : INT	
[0]	INT	
[1]	INT	
[2]	INT	
[3]	INT	
[4]	INT	
[5]	INT	
[6]	INT	
[7]	INT	
[8]	INT	
[9]	INT	
[10]	INT	
testTag1	INT	
testTag2	INT	
testTag3	BYTE	
Tag1	DINT	
Tag2	DINT	
Tag3	DINT	

Tags* ×

Data	Type	Con
CODESYS V3 ETH:prot2 Model: CODESYS 3	Container	
Application	Container	
PLC_PRG	Container	
testArrayTag	Array[11] : INT	
[0]	INT	
[1]	INT	
[2]	INT	
[3]	INT	
[4]	INT	
[5]	INT	
[6]	INT	
[7]	INT	
[8]	INT	
[9]	INT	
[10]	INT	
testTag1	INT	
testTag2	INT	
testTag3	BYTE	
Tag1	DINT	
Tag2	DINT	
Tag3	DINT	

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.

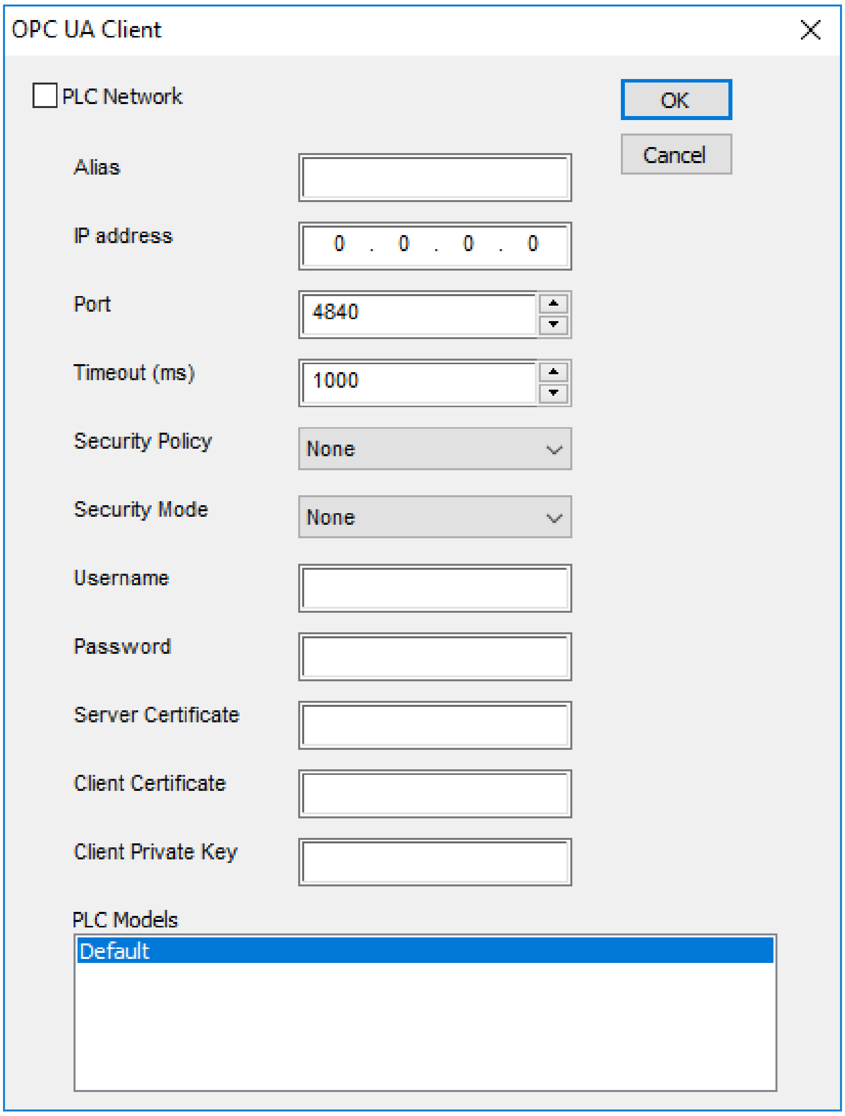
Protocol Editor Settings

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.



OPC UA Client

□ PLC Network

Alias	
IP address	0 . 0 . 0 . 0
Port	4840
Timeout (ms)	1000
Security Policy	None
Security Mode	None
Username	
Password	
Server Certificate	
Client Certificate	
Client Private Key	

PLC Models

Default

Element	Description
PLC Network	Enable access to multiple networked controllers. For every controller set proper options.
Alias	Name identifying nodes in network configurations. The name will be added as a prefix to each tag name imported for each network node.
IP Address	IP address of the server.
Port	Port number where the server is listening.
Timeout (ms)	Time delay in milliseconds between two retries in case of no response from the server device.
Security Mode	Type of authentication: • **None**: Certificates are not used • **Sign**: Certificates only used for authentication with server. • **SignAndEncrypt**: Certificates used for authentication with server and data encryption.
Security Policy	Encryption level to use (used only when Security Mode is active). • Basic256 • Basic256Sha256
Username **Password**	Authentication with user name and password
Server Certificate	Certificate for OPC UA Server. ℹ Server certificate can be downloaded using tag importer. See "Remote OPC UA Server certificate" on page 931
Client Certificate	Certificate used by OPC UA client. If blank, a certificate is automatically generated.
Client Private Key	Key used by OPC UA client. If blank, a key is automatically generated.
PLC Models	No options available.

Notes:

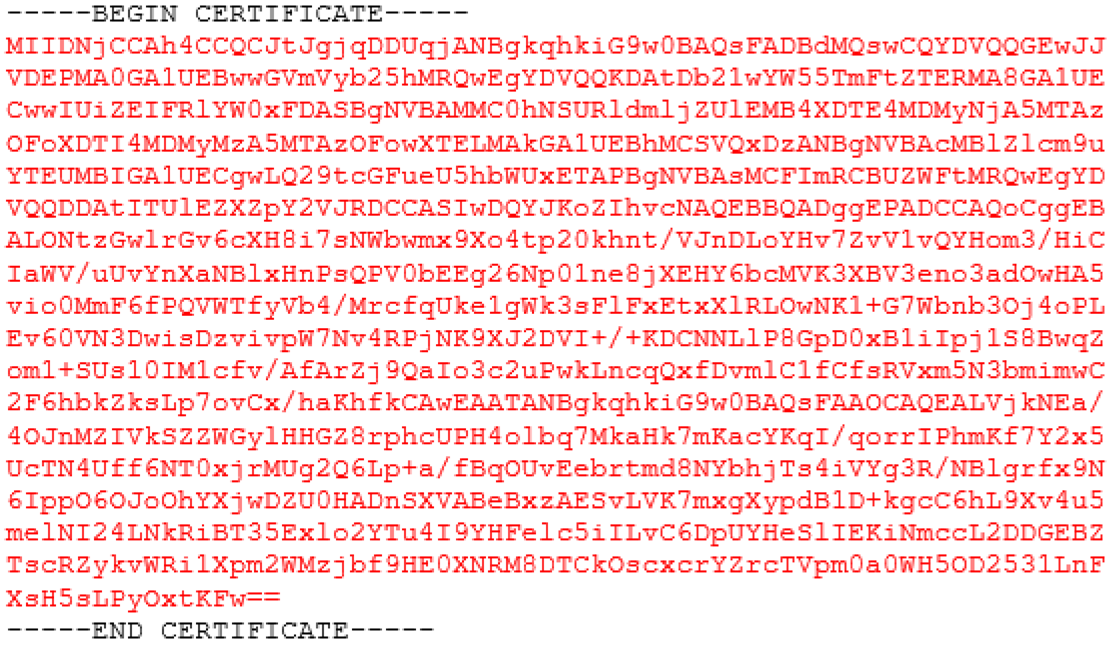
- Before choosing security options, be aware that not all security modes might be supported by the OPC UA server. Make sure to use security mode that is supported.
- When working within a private network you do not need to provide devices' certificates because you trust used devices. On a public network, 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.

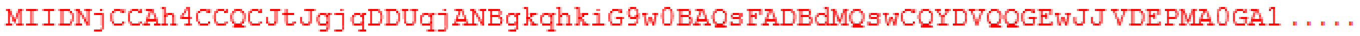
Edit the certificate files and then copy and paste the full text of your certificate to the certificate fields.

Step 1: Remove header and footer lines



```
-----BEGIN CERTIFICATE-----
MIIDNjCCAh4CCQCJtJgjqDDUqjANBgkqhkiG9w0BAQsFADBdMQswCQYDVQQGEwJJ
VDEPMA0GA1UEBwwGVmVyb25hMRQwEgYDVQQKDAtDb21wYW55TmFtZTERMA8GA1UE
CwwIUiZEIFRlYW0xFDASBgNVBAMMC0hNSURldmljZUlEMB4XDTE4MDMyNjA5MTAz
OFoXDTI4MDMyMzA5MTAzOFowXTELMAkGA1UEBhMCSVQxDzANBgNVBAcMBlZlcm9u
YTEUMBIGA1UECgwLQ29tcGFueU5hbWUxETAPBgNVBAsMCFImRCBUZWFtMRQwEgYD
VQQDDAtITUlEZXZpY2VJRDCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEB
ALONtzGwlrGv6cXH8i7sNWbwmx9Xo4tp20khnt/VJnDLoYHv7ZvV1vQYHom3/HiC
IaWV/uUvYnXaNBlxHnPsQPV0bEEg26Np01ne8jXEHY6bcMVK3XBV3eno3adOwHA5
vio0MmF6fPQVWTfyVb4/MrcfqUke1gWk3sFlFxEtxXlRLOwNK1+G7Wbnb3Oj4oPL
Ev60VN3DwisDzvivpW7Nv4RPjNK9XJ2DVI+/+KDCNNLlP8GpD0xB1iIpj1S8BwqZ
om1+SUs10IM1cfv/AfArZj9QaIo3c2uPwkLncqQxfDvmlC1fCfsRVxm5N3bmimwC
2F6hbkZksLp7ovCx/haKhfkCAwEAATANBgkqhkiG9w0BAQsFAAOCAQEALVjkNEa/
4OJnMZIVkSZZWGylHHGZ8rphcUPH4olbq7MkaHk7mKacYKqI/qorrIPhmKf7Y2x5
UcTN4Uff6NT0xjrMUg2Q6Lp+a/fBqOUvEebrtmd8NYbhjTs4iVYg3R/NBlgrfx9N
6IppO6OJoOhYXjwDZU0HADnSXVABeBxzAESvLVK7mxgXypdB1D+kgcC6hL9Xv4u5
melNI24LNkRiBT35Exlo2YTu4I9YHFelc5iILvC6DpUYHeSlIEKiNmccL2DDGEBZ
TscRZykvWRi1Xpm2WMzjbf9HE0XNRM8DTCkOscxcrYZrcTVpm0a0WH5OD2531LnF
XsH5sLPyOxtKFw==
-----END CERTIFICATE-----
```

Step 2: Remove all Newline characters



```
MIIDNjCCAh4CCQCJtJgjqDDUqjANBgkqhkiG9w0BAQsFADBdMQswCQYDVQQGEwJJVDEPMA0GA1.....
```

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.

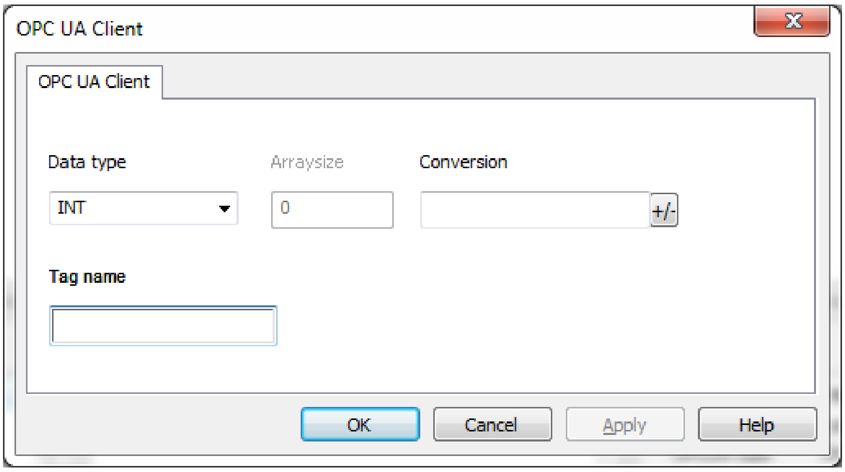
Here is an example of how to generate certificate files using a public OpenSSL-Win32 library (Reference: https://www.openssl.org/)

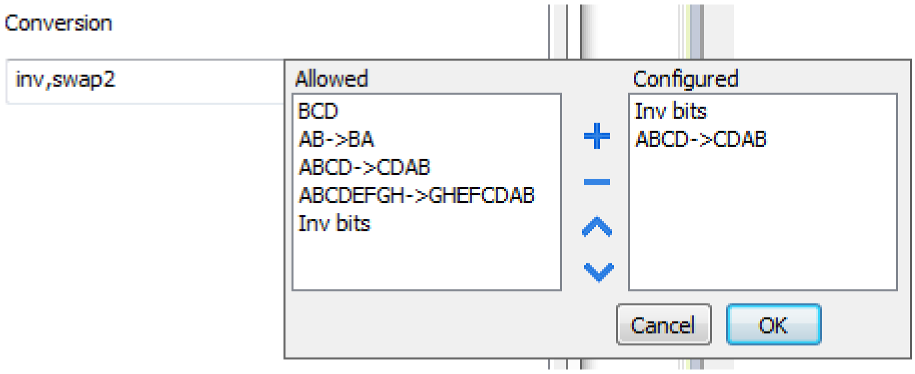
```
@echo off
set OpenSSL="C:\Program Files (x86)\OpenSSL-Win32\bin\openssl.exe"
set NodeName=HMI-Client

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.



Element	Description
Data Type	Available data types: • **boolean** • **byte** • **short** • **int** • **unsignedByte** • **unsignedShort** • **unsignedInt** • **float** • **double** • **time** • **uint64** • **int64** • **string** • **binary** See "Programming concepts" section in the main manual. Note: To define arrays, select one of Data Type format followed by square brackets.
Arraysize	• 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.  Conversion: inv,swap2 Allowed: BCD, AB->BA, ABCD->CDAB, ABCDEFGH->GHEFCDAB, Inv bits Configured: Inv bits, ABCD->CDAB Depending on data type selected, the list **Allowed** shows one or more conversion types.

<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>Value</th><th>Description</th></tr>
<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 -> 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 -> 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 -> 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 -> OPM...DAB</b></td><td><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)</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>
</table>
Select conversion and click +. The selected item will be added to list <b>Configured</b>.</td></tr>
</table>

Element	Description
	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 name	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.

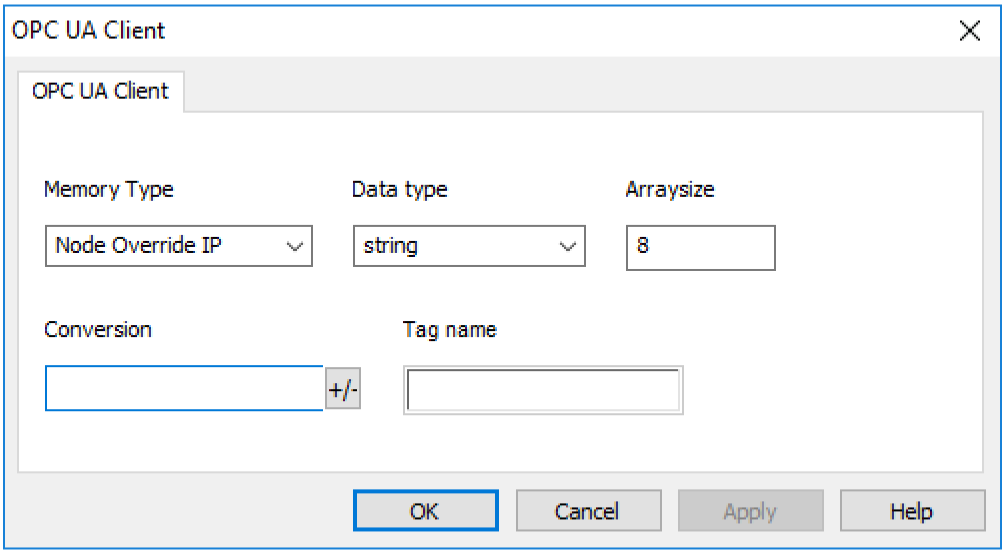
Node Override IP	PLC operation
0.0.0.0	Communication with the controller is stopped, no request frames are generated anymore.
Different from 0.0.0.0	It is interpreted as node IP override and the target IP address is replaced runtime with the new value.

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.



Memory Type: Node Override IP

Data type: string

Arraysize: 8

Conversion:

Tag name:

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.

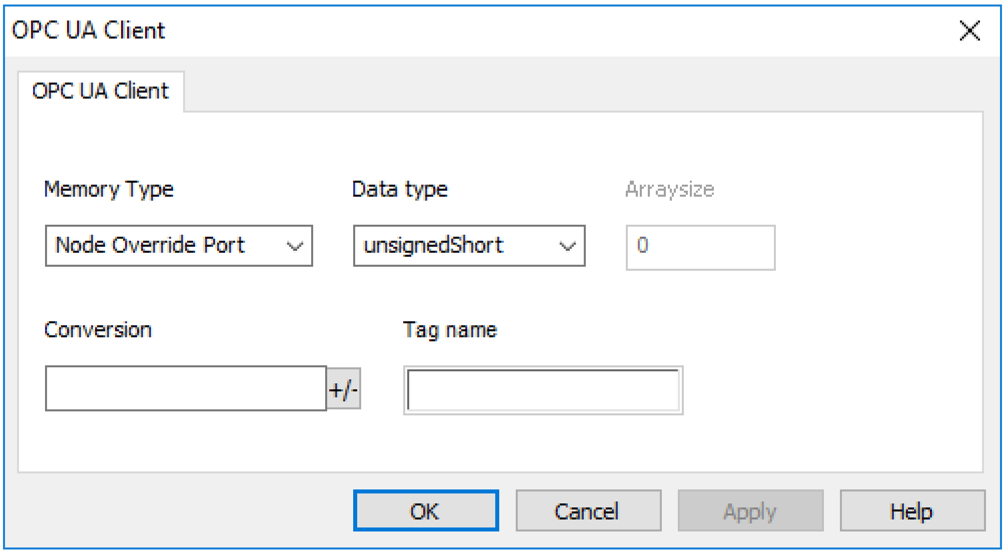
This memory type is unsigned short.

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.



OPC UA Client

Memory Type: Node Override Port

Data type: unsignedShort

Arraysize: 0

Conversion: +/-

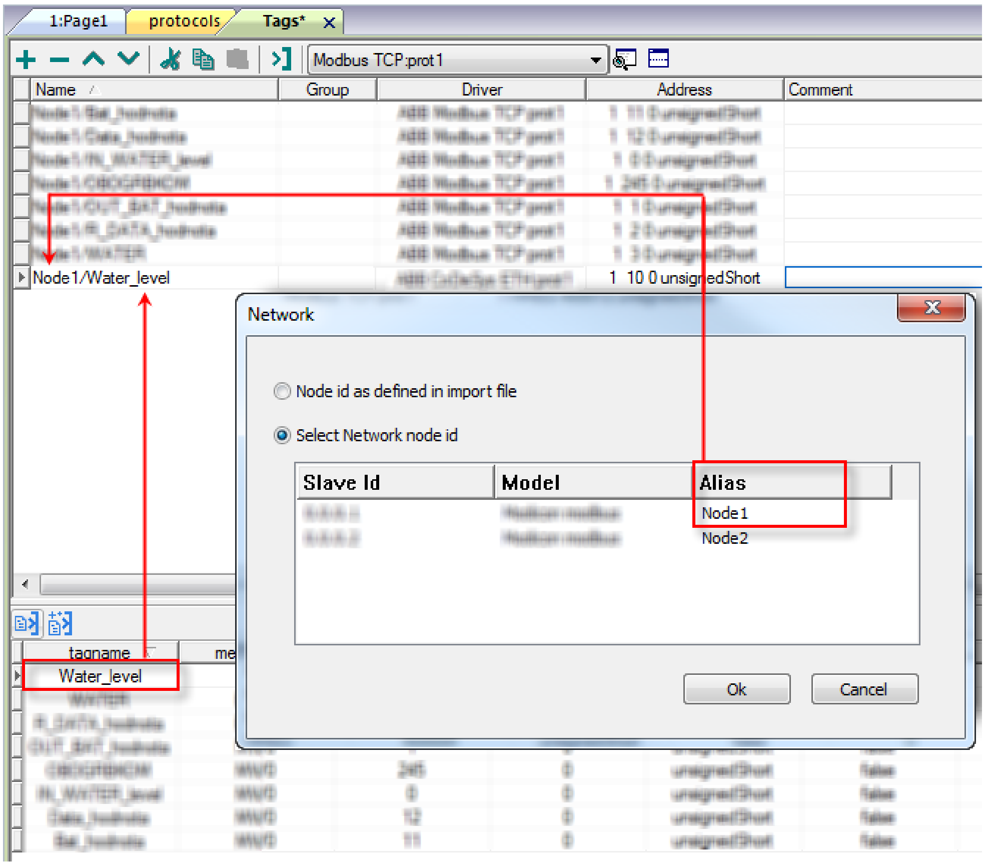
Tag name:

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.



Modbus TCP:prot1

Name	Group	Driver	Address	Comment
Node1/Water_level			1 10 0 unsignedShort	

Network

○ Node id as defined in import file

● Select Network node id

Slave Id	Model	Alias
		Node1
		Node2

tagname
Water_level

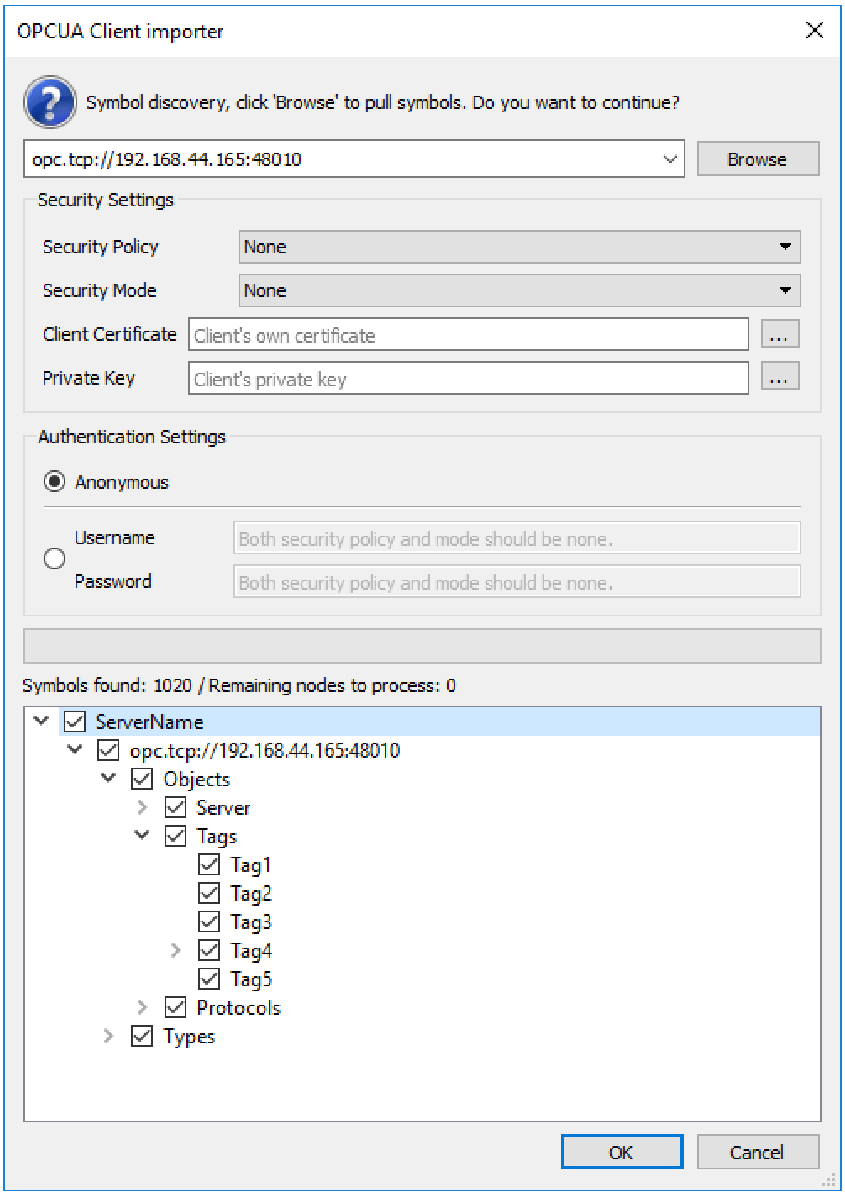
Note: Aliasing tag names is only available for imported tags. Tags added manually in the Tag Editor cannot have the Alias prefix in the tag name.
The Alias string is attached at the time of tag import. If you modify the Alias string after the tag import has been completed, there will be no effect on 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.



OPCUA Client importer

Symbol discovery, click 'Browse' to pull symbols. Do you want to continue?

opc.tcp://192.168.44.165:48010

Security Settings

Security Policy: None

Security Mode: None

Client Certificate: Client's own certificate

Private Key: Client's private key

Authentication Settings

● Anonymous

○ Username: Both security policy and mode should be none.
Password: Both security policy and mode should be none.

Symbols found: 1020 / Remaining nodes to process: 0

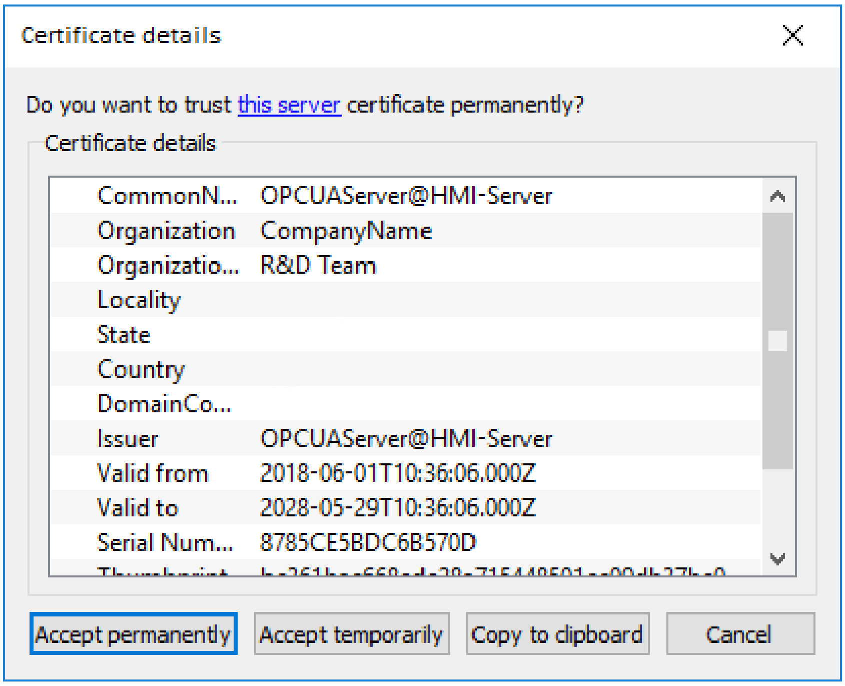
- ☑ ServerName
  - ☑ opc.tcp://192.168.44.165:48010
    - ☑ Objects
      - ☑ Server
      - ☑ Tags
        - ☑ Tag1
        - ☑ Tag2
        - ☑ Tag3
        - ☑ Tag4
        - ☑ Tag5
      - ☑ Protocols
    - ☑ Types

Element	Description
Remote URI	Address of OPC UA Server in the form: *opc:tcp:<IPAddress>:<Port>* Example: • opc.tcp://192.168.44.165:4840
Security Mode	Type of authentication: • **None**: No authentication with server and no data encryption. • **Sign**: Certificates only used for authentication with server. • **SignAndEncrypt**: Certificates used for authentication with server and data encryption.
Security Policy	Encryption level to use (used only when Security Mode is active).

Element	Description
	• Basic128Rsa15 • Basic256 • Basic256Sha256
Username **Password**	Authentication with user name and password
Client Certificate	Certificate used by OPC UA client. If blank, a certificate is automatically generated. The certificate is used by the importer only if requested by the server
Client Private Key	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



Certificate details

Do you want to trust this server certificate permanently?

Certificate details

CommonN...	OPCUAServer@HMI-Server
Organization	CompanyName
Organizatio...	R&D Team
Locality	
State	
Country	
DomainCo...	
Issuer	OPCUAServer@HMI-Server
Valid from	2018-06-01T10:36:06.000Z
Valid to	2028-05-29T10:36:06.000Z
Serial Num...	8785CE5BDC6B570D

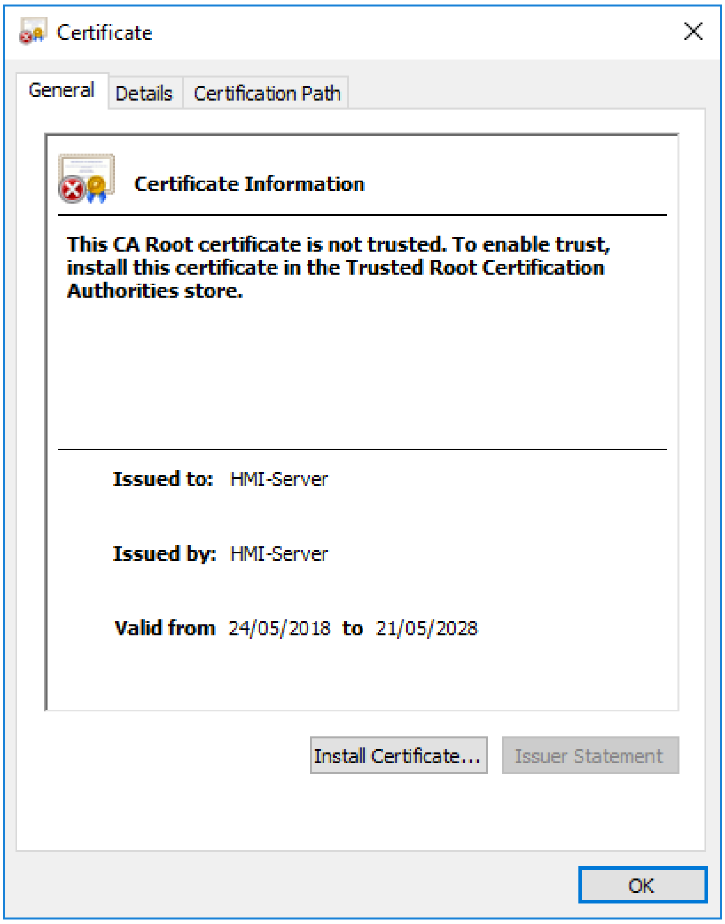
When OPC UA Server provides its own certificate, you have the option to:

- **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.



This CA Root certificate is not trusted. To enable trust, install this certificate in the Trusted Root Certification Authorities store.

Issued to: HMI-Server

Issued by: HMI-Server

Valid from 24/05/2018 to 21/05/2028

- **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
Connecting <Error description>	Error during connection
Connection while reading: <Error description>	Error encountered when connecting for read operation
Bad status while reading: <Error description>	Error in read operation
Connection while writing: <Error description>	Error encountered when connecting for write operation
Bad status writing: <Error description>	Error in write operation
OPC UA client for given node ID not found	Wrong node ID information

<Error description> can be one of the following:

Error	Notes
BadTimeout	Timeout error. No answer from server.
BadSecurityChecksFailed	Error during exchange of certificates. Typically occurs when the server does not accept the client certificate as trusted.
BadCertificatexxxInvalid	Error in client or server certificate.
BadNodeUnknown	The tag (node) does not exist.
BadAttributeNotFound	Attempt to access an invalid attribute.
BadNotWritable	Attempt to write to a read-only attribute.

Simatic S7 PPI

HMI devices can be connected to the Siemens Simatic S7-200 family of PLCs. The communication is performed via the PLC programming ports using the PPI and the PPI+ protocols.

This document describes the PPI+ protocol and includes the information needed for a successful connection.

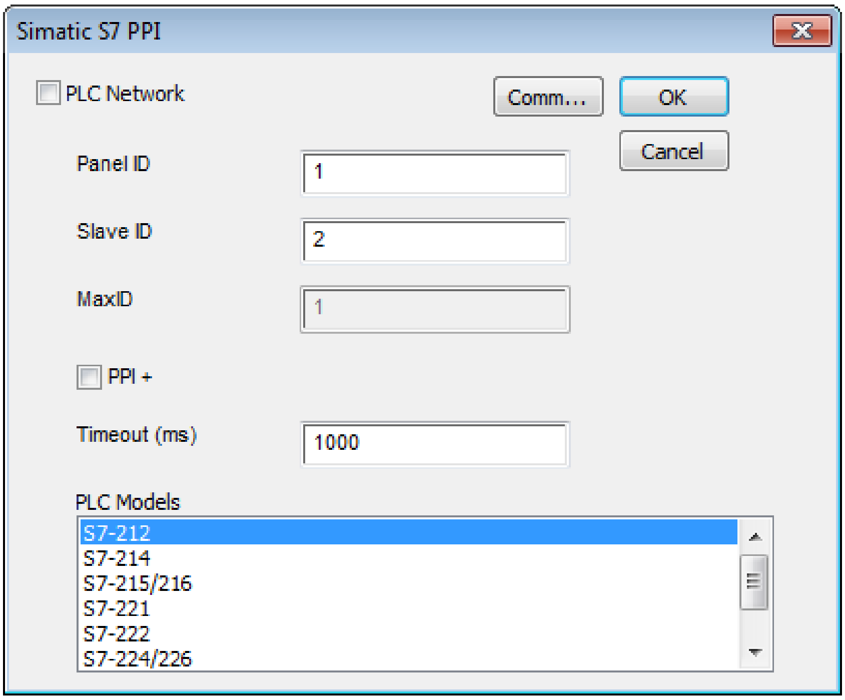
Protocol Editor Settings

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.



Simatic S7 PPI

☐ PLC Network

Panel ID: 1

Slave ID: 2

MaxID: 1

☐ PPI +

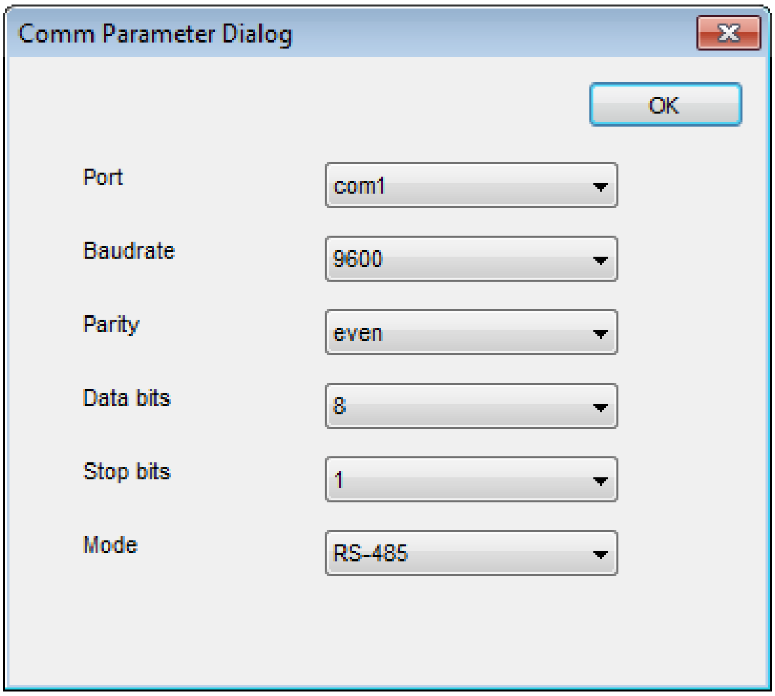
Timeout (ms): 1000

PLC Models: S7-212, S7-214, S7-215/216, S7-221, S7-222, S7-224/226

Element	Description
PLC Network	Enable access to multiple networked controllers. For every controller (slave) set the proper option.
Panel ID	Node number of the operator panel.
Slave ID	Node number of the connected PLC.
Max ID	Available only if PPI+ protocol is in use. Contains the highest node number in PPI+ network.

Element	Description
PPI+	Checked to use PPI+ protocol instead of PPI protocol.
Timeout (ms)	Time delay in milliseconds between two retries of the same message when no answer is received from the controller.

Element	Description
PLC Models	Several Siemens controllers are supported. Please check directly in the programming IDE software for a complete list of supported controllers.
Comm...	If clicked displays the communication parameters setup dialog.



Comm Parameter Dialog

Port: com1

Baudrate: 9600

Parity: even

Data bits: 8

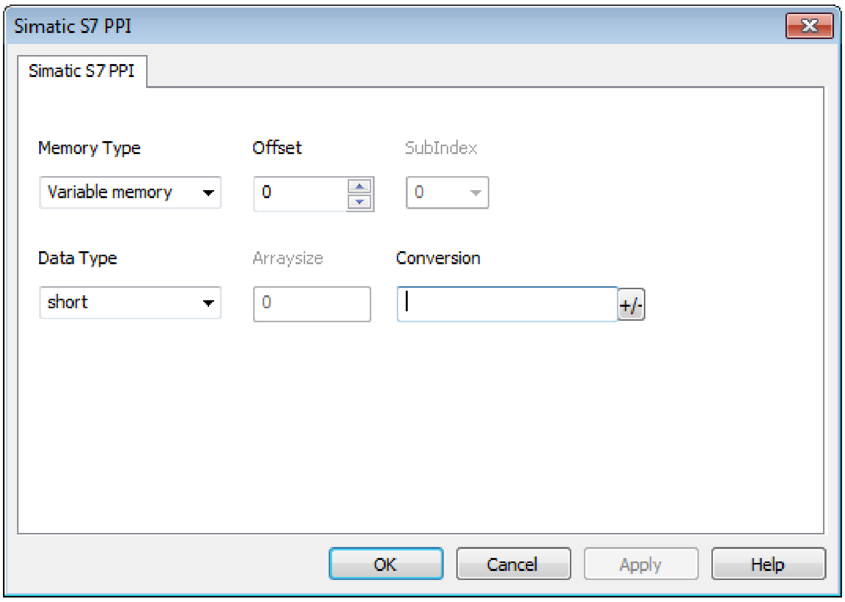
Stop bits: 1

Mode: RS-485

Element	Parameter
Port	Serial port selection. On UN20: • **COM1**: device PLC port. • **COM2**: PC/printer port On UN31 or UN30: • **COM1**: integrated serial port • **COM2**: optional module plugged on Slot 1/2 • **COM3**: optional module plugged on Slot 3/4
Baudrate, Parity, Data Bits, Stop bits	Serial line parameters.
Mode	Serial port mode. Available modes: • **RS-232**. • **RS-485** (2 wires). • **RS-422** (4 wires).

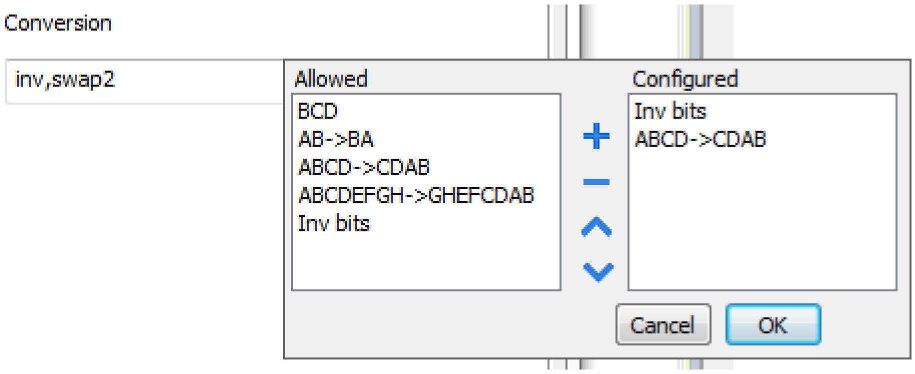
Tag Editor Settings

In the Tag Editor select Simatic S7 PPI from the list of defined protocols and click **+** to add a tag.



Element	Description
Memory Type	Area of PLC where tag is located.
Offset	Offset address where tag is located.
SubIndex	In case of Boolean data type, this is the offset of single bit.
Data Type	Available data types: • **boolean** • **byte** • **short** • **int** • **unsignedByte** • **unsignedShort** • **unsignedInt** • **float** • **string** See "Programming concepts" section in the main manual.
Arraysize	• 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

Element	Description
	requires 2 bytes.
Conversion	Conversion to be applied to the tag.



Conversion

inv,swap2

Allowed

BCD
AB->BA
ABCD->CDAB
ABCDEFGH->GHEFCDAB
Inv bits

Configured

Inv bits
ABCD->CDAB

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

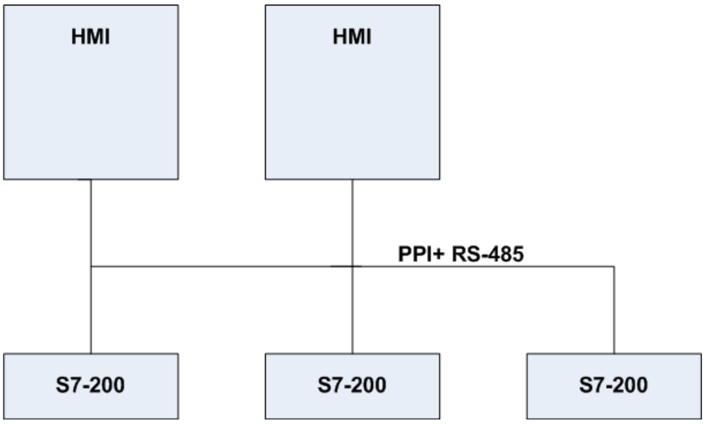
Value	Description
Inv bits	**inv**: Invert all the bits of the tag. *Example:* 1001 → 0110 (in binary format) 9 → 6 (in decimal format)
Negate	**neg**: Set the opposite of tag value. *Example:* 25.36 → -25.36
AB -> BA	**swapnibbles**: Swap nibbles in a byte. *Example:* 15D4 → 514D (in hexadecimal format) 5588 → 20813 (in decimal format)
ABCD -> CDAB	**swap2**: Swap bytes in a word. *Example:* 9ACC → CC9A (in hexadecimal format) 39628 → 52378 (in decimal format)
ABCDEFGH -> GHEFCDAB	**swap4**: Swap bytes in a double word. *Example:* 32FCFF54 → 54FFFC32 (in hexadecimal format) 855441236 → 1426062386 (in decimal format)
ABC...NOP -> OPM...DAB	**swap8**: Swap bytes in a long word. Example:

<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td></td><td>142.366 → -893553517.588905 (in decimal format)
0 10000000110
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>
</table>
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.
</td></tr>
</table>

PPI+ Connectivity

HMI devices can be connected to more than one CPU S7-200, more than one operator panel can also be connected to the same PLC.

Operator panels will not interfere with PPI+ communication between the PLC’s.



PPI+ protocol allows you to use more complex configurations than the standard PPI protocol.

Each PLC can execute read and write operations to and from other PLCs. At the same time more than one panel can be connected on the PPI network and can access all the variables from all the PLCs.

PLC programming software can be used and online programming can be performed without interfering with the panel-PLC communication .

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
NAK	The controller replies with a not acknowledge.	-
Timeout	A request is not replied within the specified timeout period.	Check if the controller is connected and properly configured to get network access.
Invalid response	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.
General Error	Unidentifiable error. Should never be reported.	Contact technical support.

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.

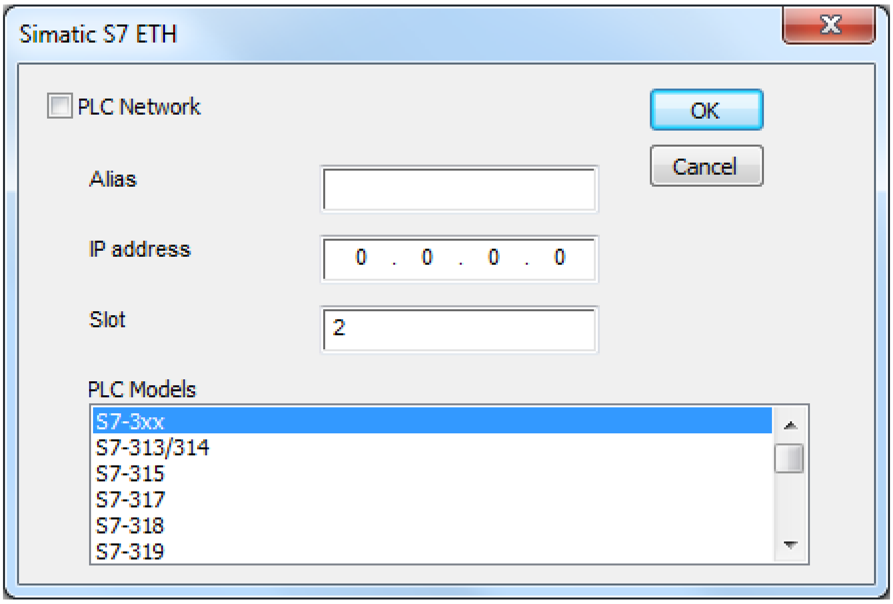
Protocol Editor Settings

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.



Simatic S7 ETH

□ PLC Network

Alias:

IP address: 0 . 0 . 0 . 0

Slot: 2

PLC Models: S7-3xx (selected), S7-313/314, S7-315, S7-317, S7-318, S7-319

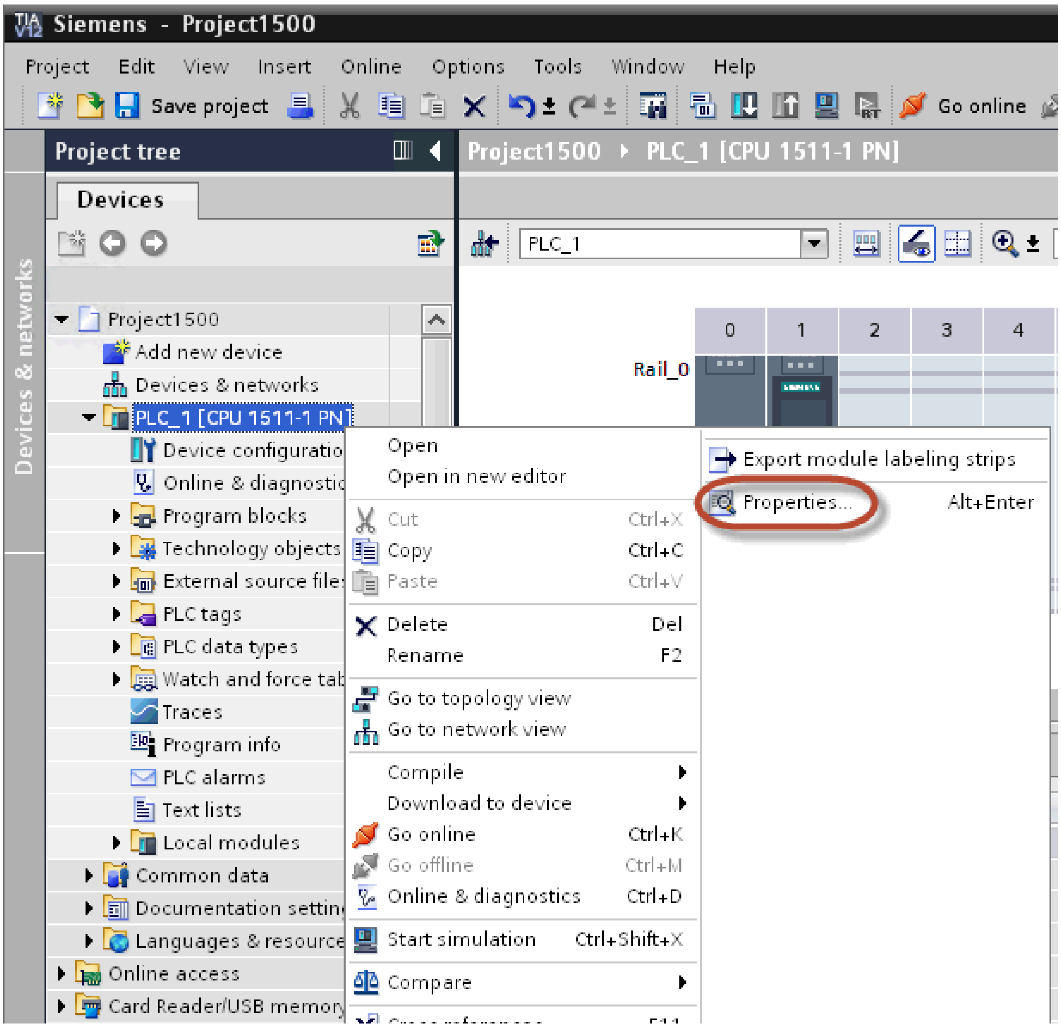
Element	Description
Alias	Name identifying nodes in network configurations. The name will be added as a prefix to each tag name imported for each network node.
IP address	Ethernet IP address of the controller.
Slot	Number of the slot where the CPU is mounted. 2 for S7-300, may take a higher value for S7-400

Element	Description
	systems.
PLC Models	List of compatible controller models. Make sure to select the correct PLC model in this list when configuring the protocol.
PLC Network	Enable access to multiple networked controllers. For every controller (slave) set the proper option.

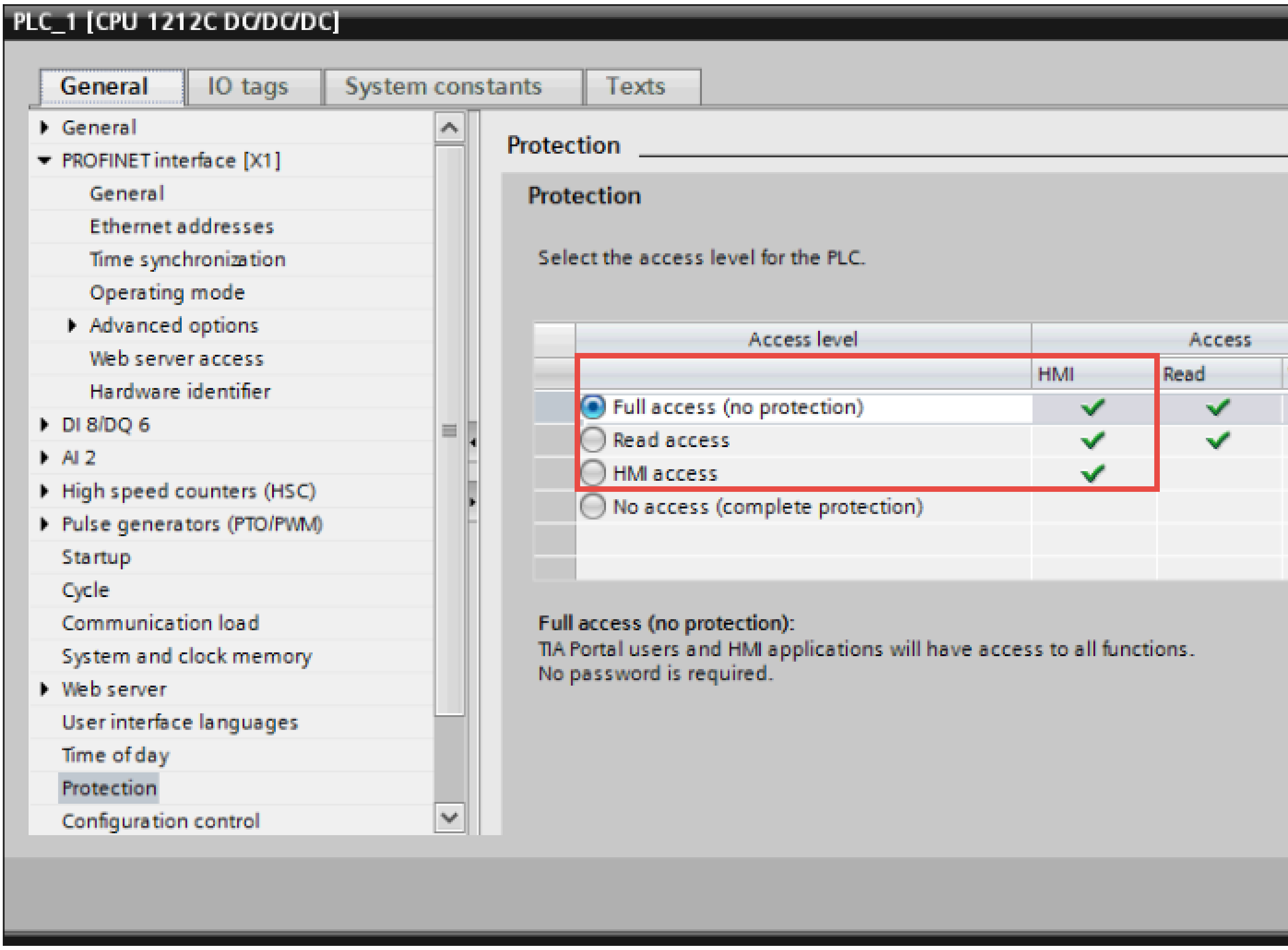
S7-1200 and S7-1500 PLC configuration

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 3rd 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).



Protection

Protection

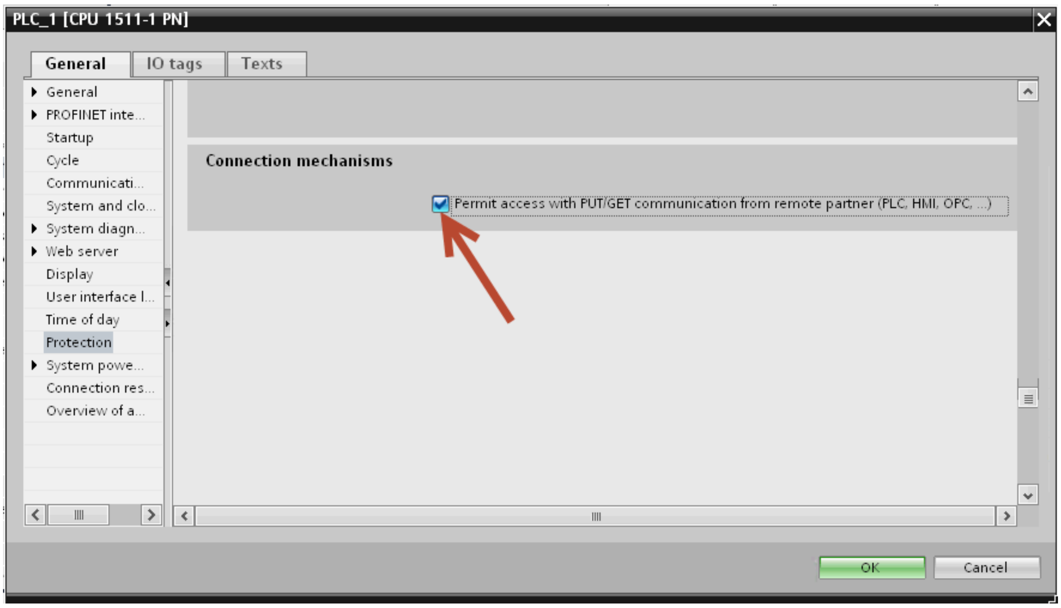
Select the access level for the PLC.

Access level	Access	
	HMI	Read
● Full access (no protection)	✓	✓
○ Read access	✓	✓
○ HMI access	✓	
○ No access (complete protection)		

Full access (no protection):
TIA Portal users and HMI applications will have access to all functions.
No password is required.

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".

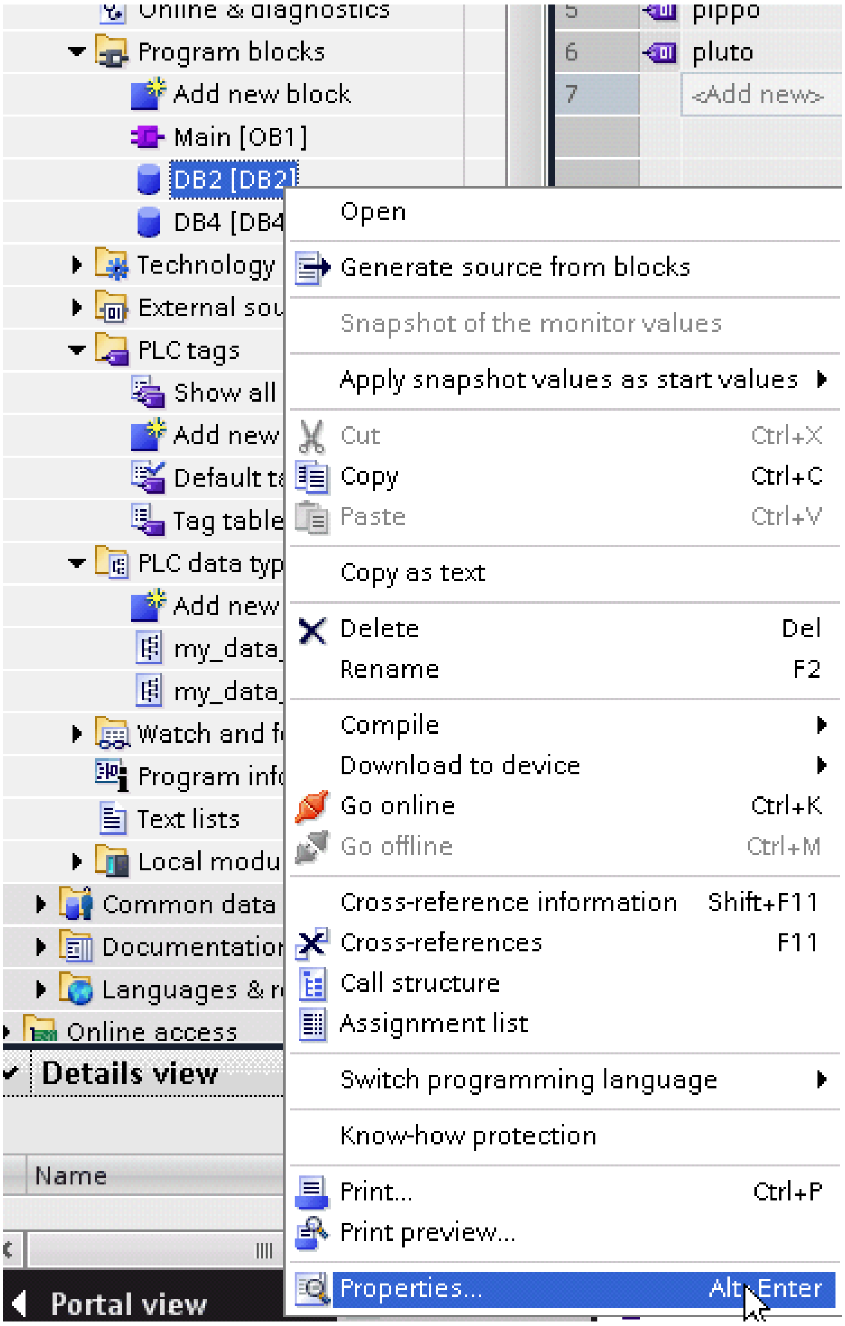


Connection mechanisms

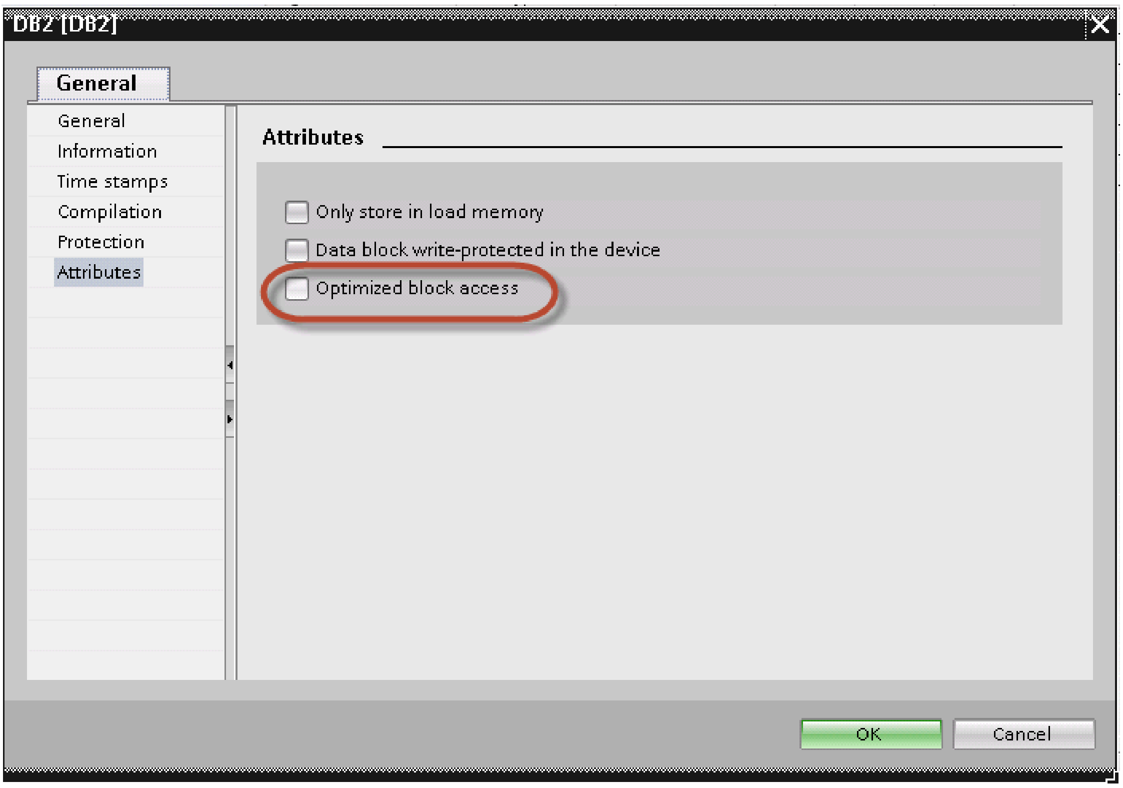
☑ Permit access with PUT/GET communication from remote partner (PLC, HMI, OPC, ...)

Note: If variables are defined in "Program blocks", DB must configured as "Not optimized".

To check or change DB optimization, open DB Properties:



In General > Attributes uncheck "Optimized block access":



Attributes

□ Only store in load memory

□ Data block write-protected in the device

□ 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".

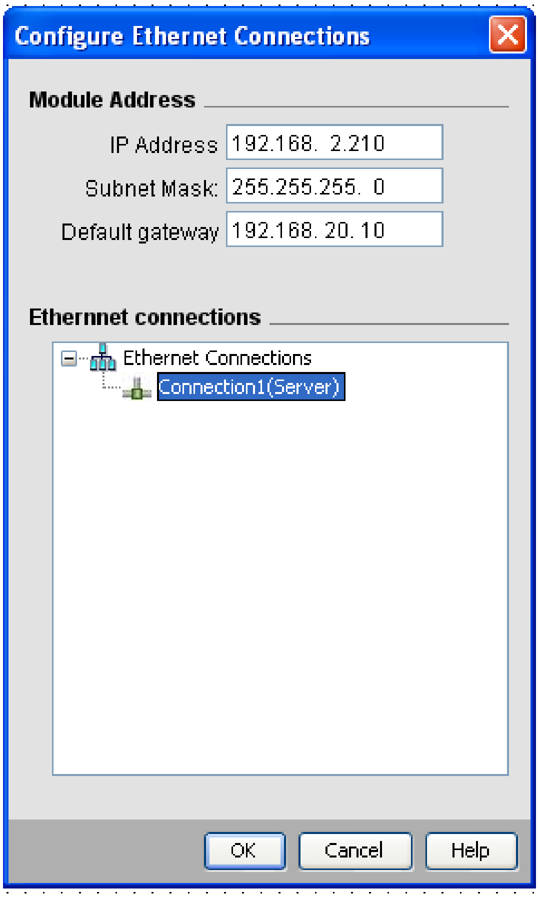
After compiling the project, tag offsets will be shown close to variable name.

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.



Configure Ethernet Connections

Module Address

IP Address 192.168. 2.210

Subnet Mask: 255.255.255. 0

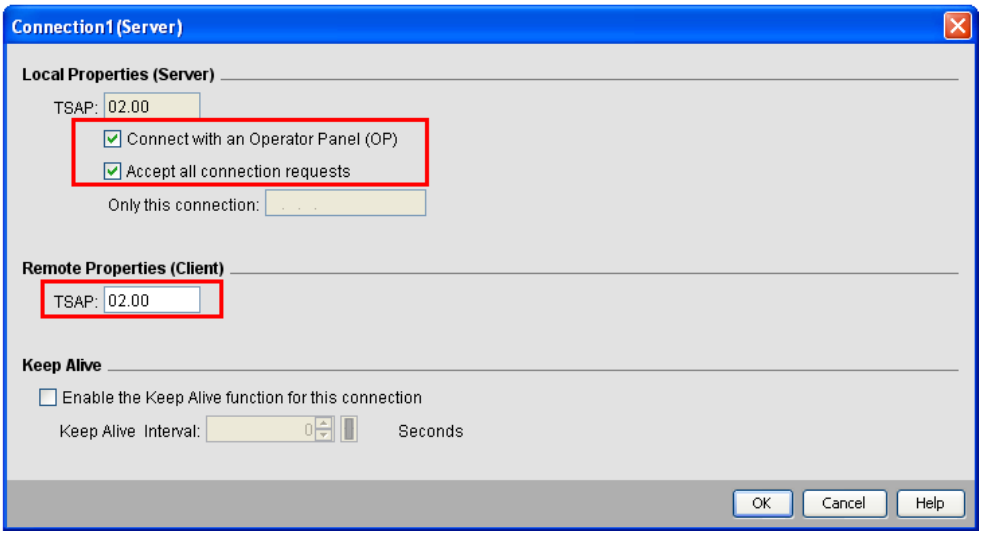
Default gateway 192.168. 20. 10

Ethernnet connections

- Ethernet Connections
  - Connection1(Server)

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.



Connection1(Server)

Local Properties (Server)

TSAP: 02.00

☑ Connect with an Operator Panel (OP)

☑ Accept all connection requests

Only this connection: . . .

Remote Properties (Client)

TSAP: 02.00

Keep Alive

□ Enable the Keep Alive function for this connection

Keep Alive Interval: 0 Seconds

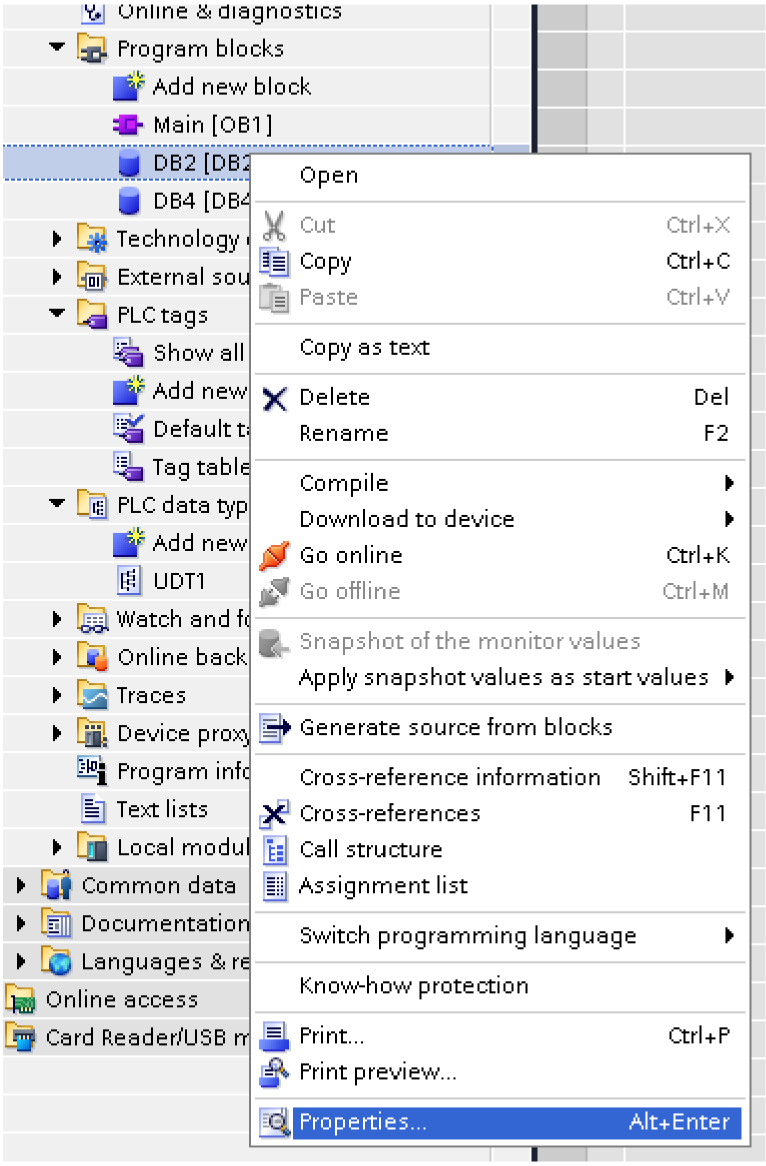
5. Select the **Connect with an operator panel (OP)** (0BA7 model only, do noth check for Logo! 0BA8 model)
6. Select Accept all connection requests options.
7. In the **Remote Properties (Client)** section, set **TSAP** to 02.00.

Direct Import of TIA Portal project

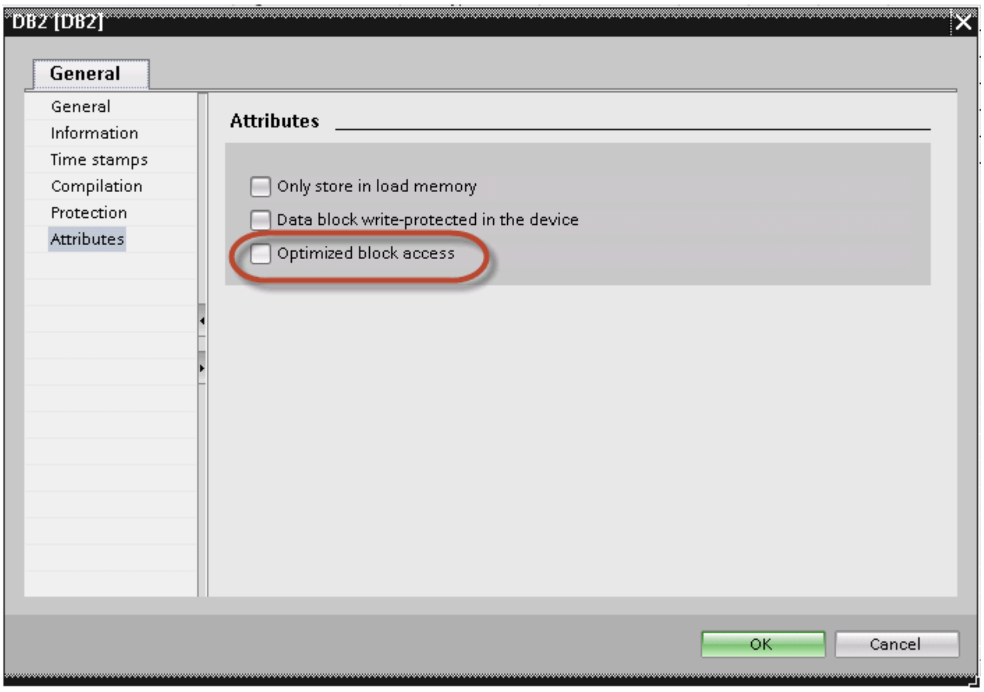
It is possible to import TIA Portal variables directly from TIA Portal project, by selecting "TIA Portal Project v12 or newer" from import selection (refer to "Tag Import" chapter).

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**.



Attributes

□ Only store in load memory

□ Data block write-protected in the device

□ Optimized block access

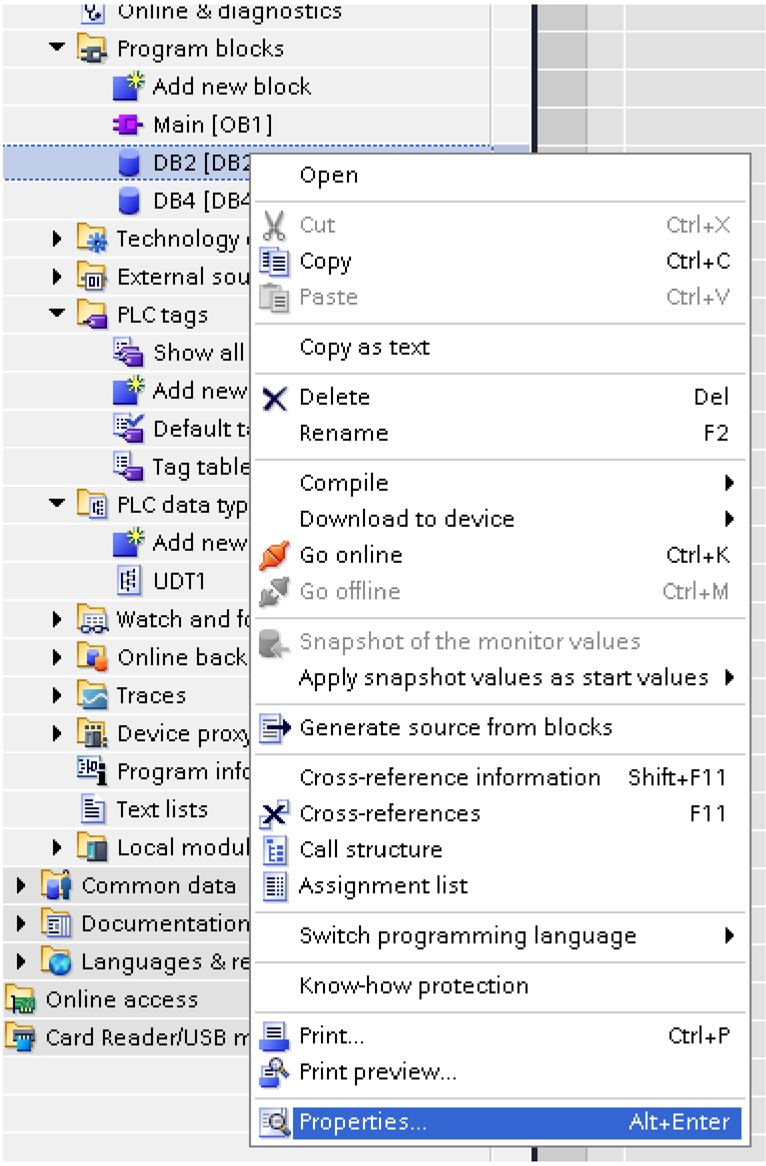
Note: If the options **Optimized block access** is not enabled (checkbox grayed out) this might mean that the Data Block is an "instance DB" linked to an "optimized access FB".

Export using TIA Portal v13, v14 or newer

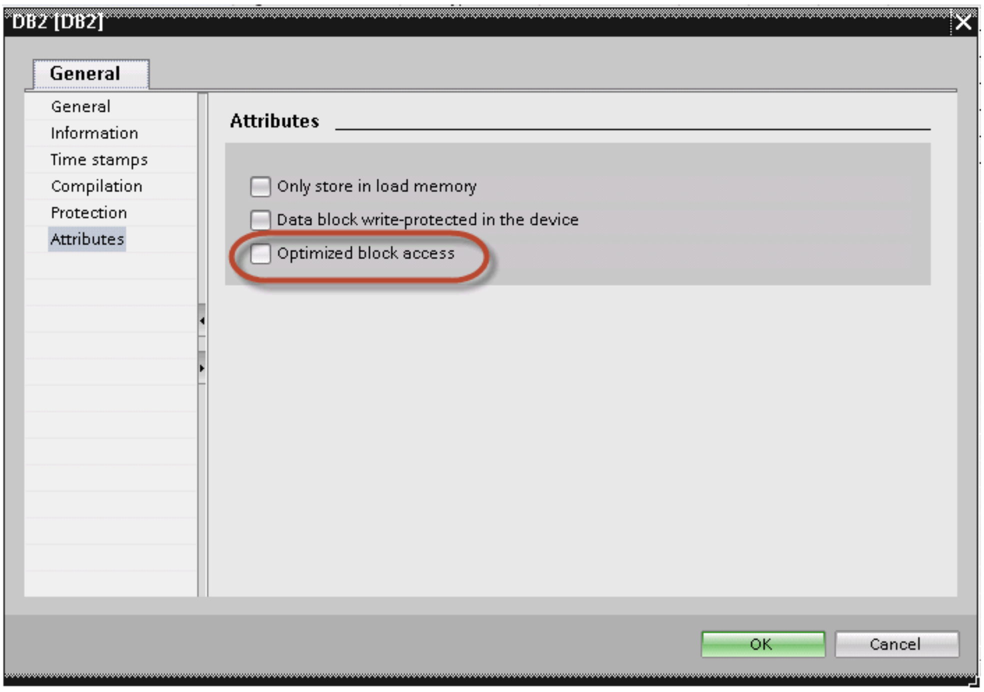
Exporting Program blocks

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**.



DB2 [DB2]

General

- General
- Information
- Time stamps
- Compilation
- Protection
- Attributes

Attributes

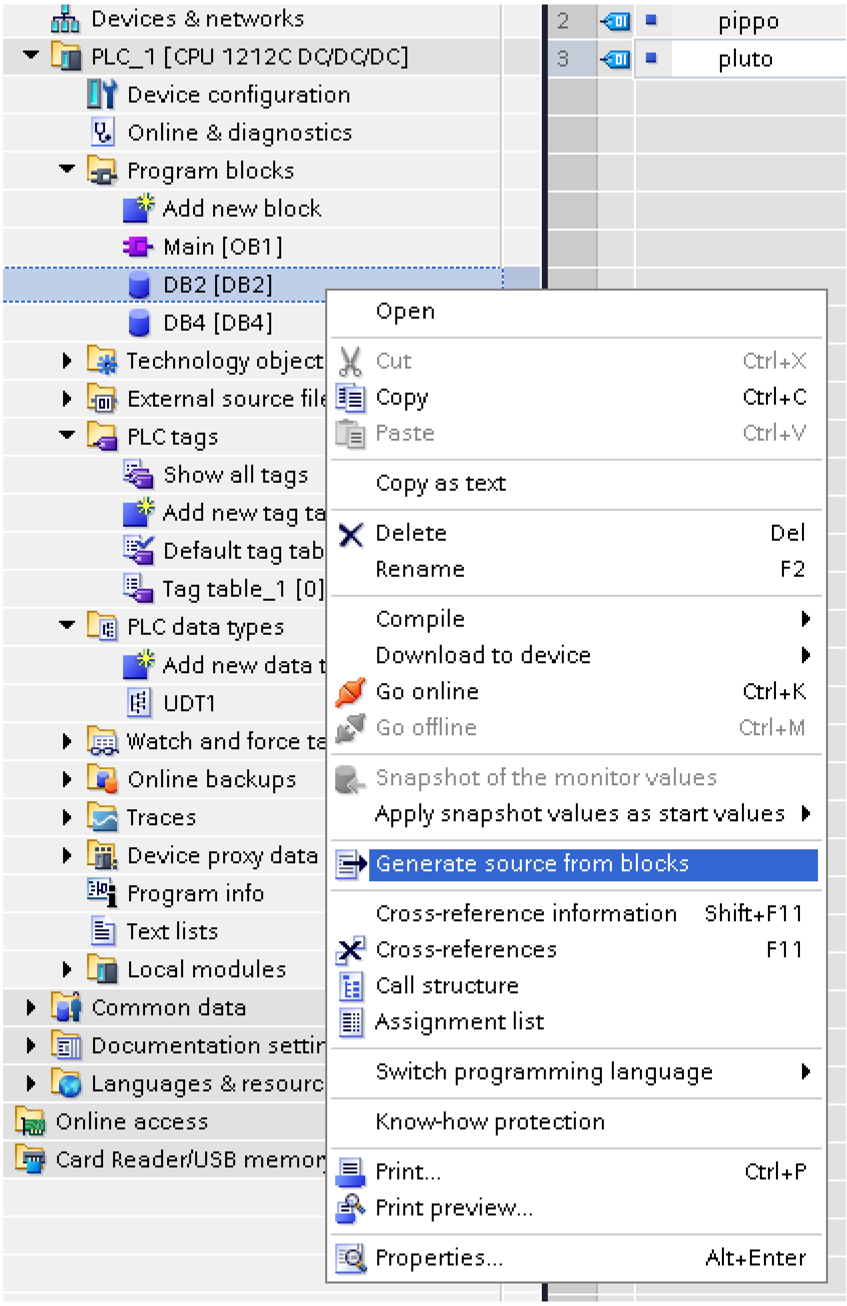
□ Only store in load memory

□ Data block write-protected in the device

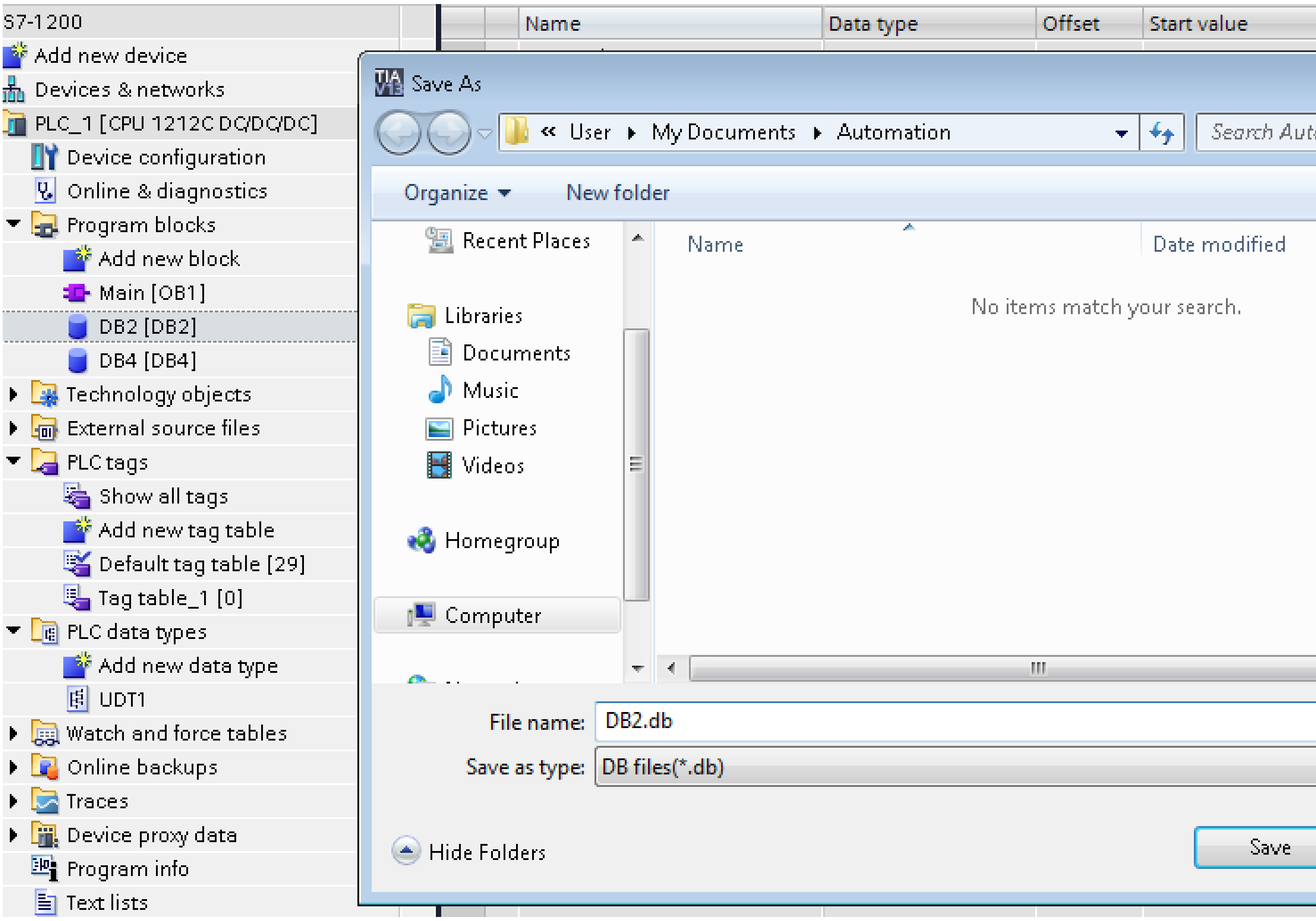
□ Optimized block access

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.



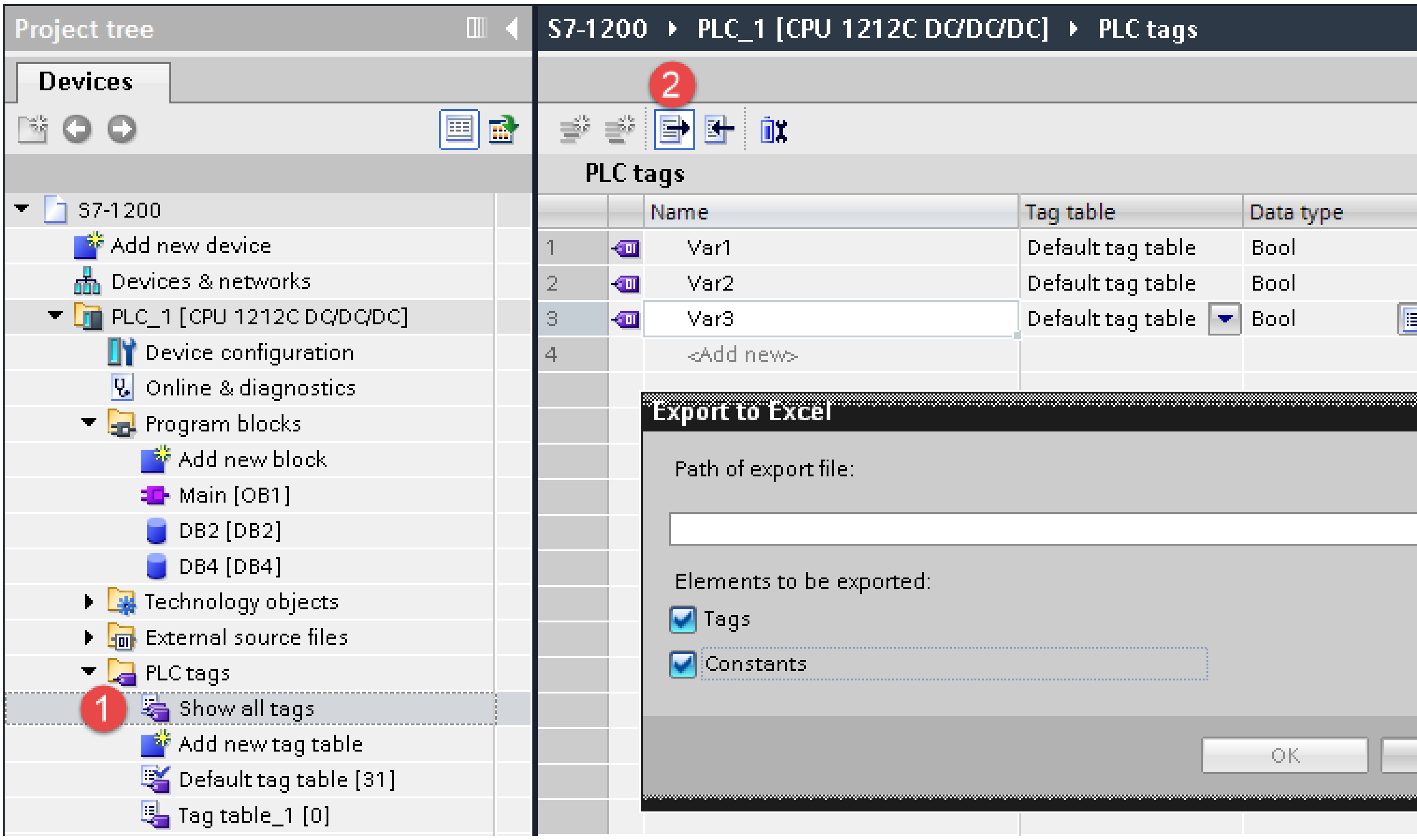
File name: DB2.db

Save as type: DB files(*.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.



		Name	Tag table	Data type
1		Var1	Default tag table	Bool
2		Var2	Default tag table	Bool
3		Var3	Default tag table	Bool
4		<Add new>		

Export to Excel

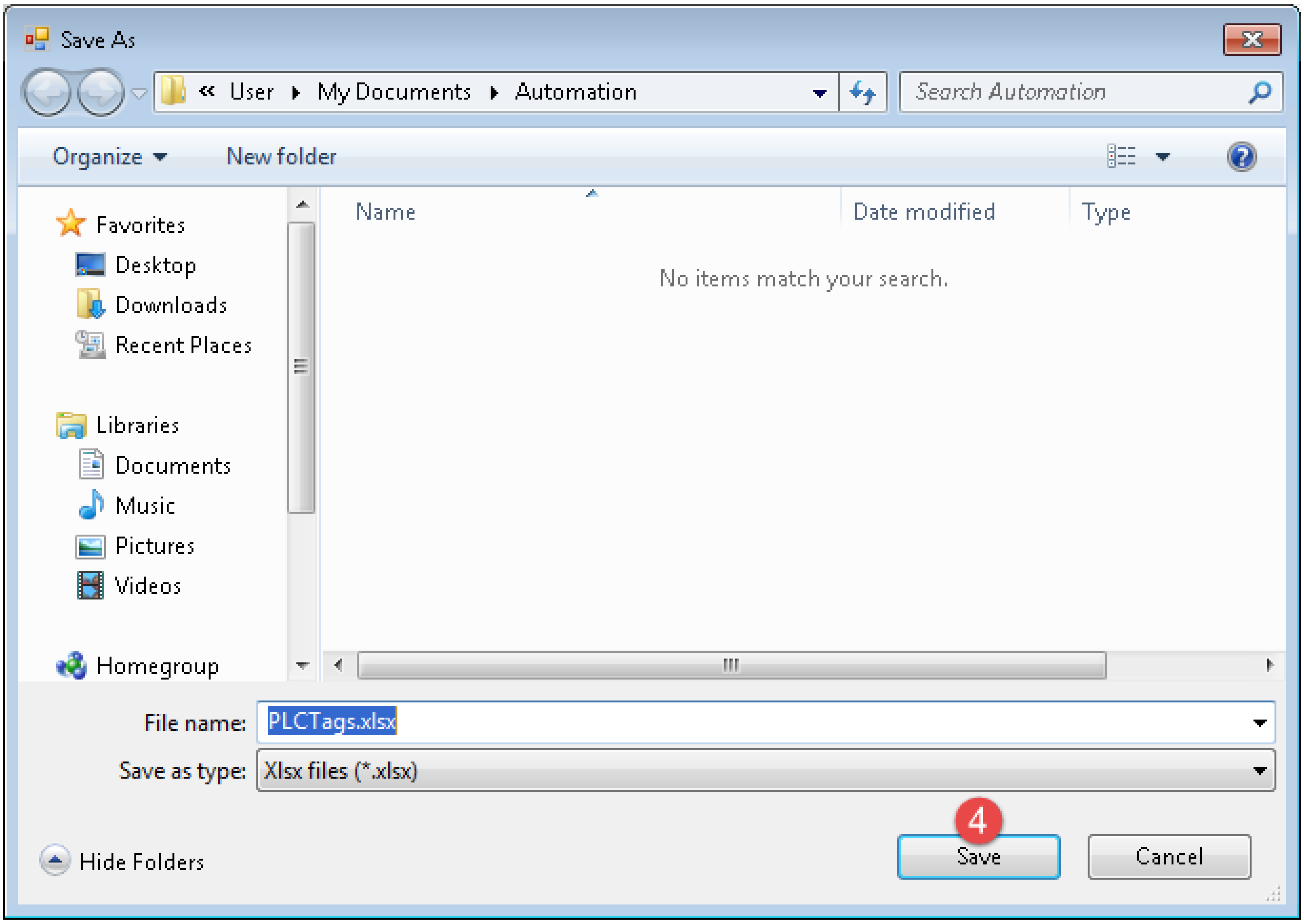
Path of export file:

Elements to be exported:

☑ Tags

☑ Constants

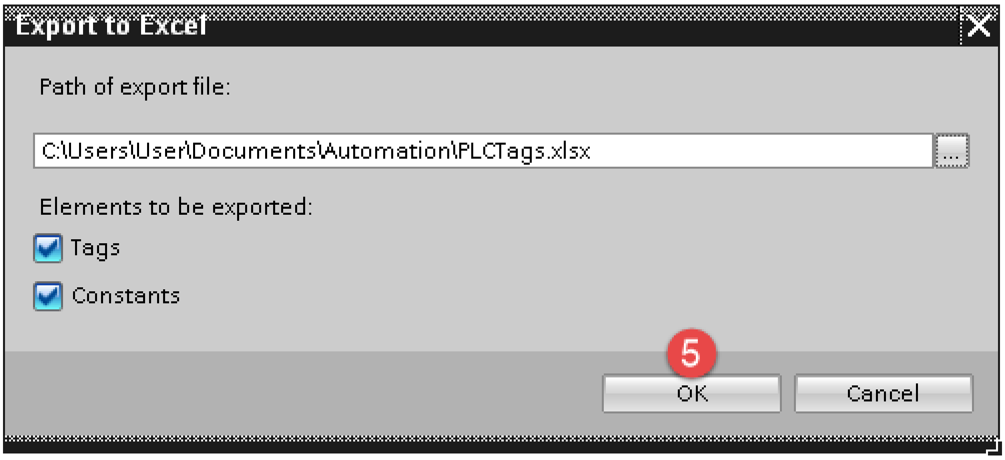
4. Click **Save** to confirm.



File name: PLCTags.xlsx

Save as type: Xlsx files (*.xlsx)

5. Click **OK** to export.



Path of export file:

C:\Users\User\Documents\Automation\PLCTags.xlsx

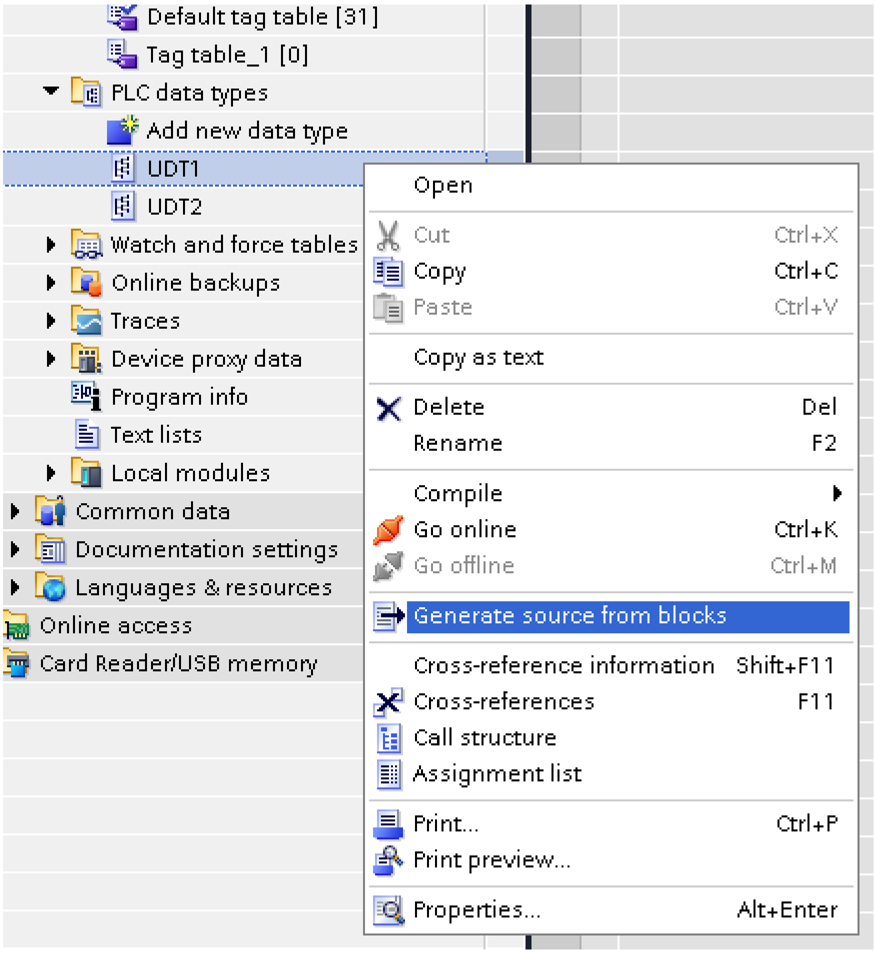
Elements to be exported:

☑ Tags

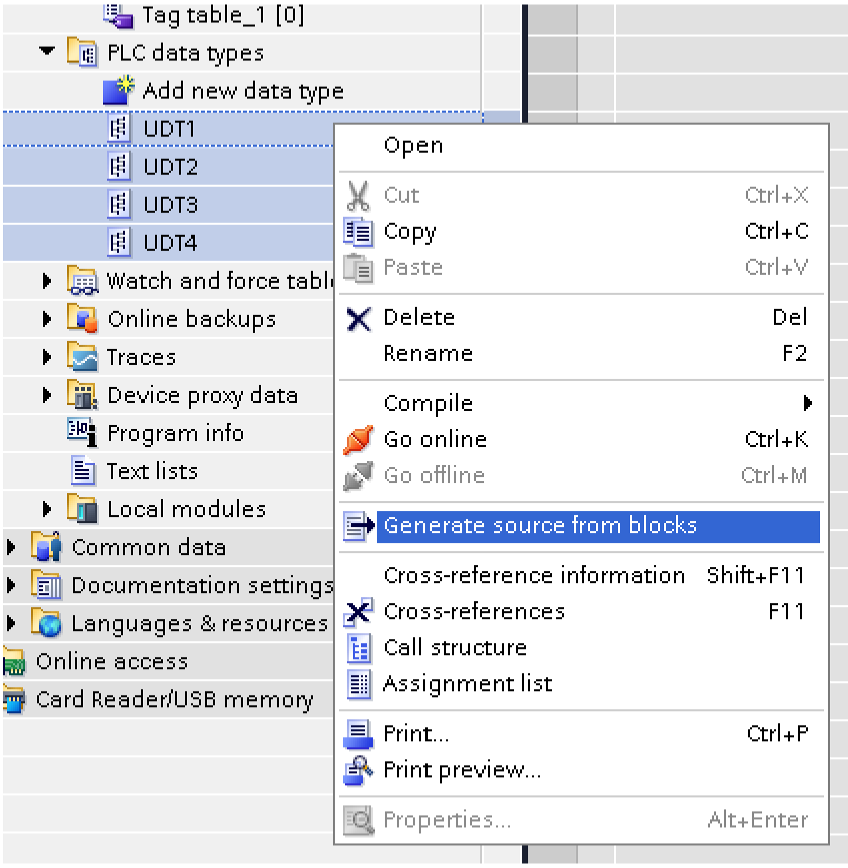
☑ Constants

Exporting PLC data types

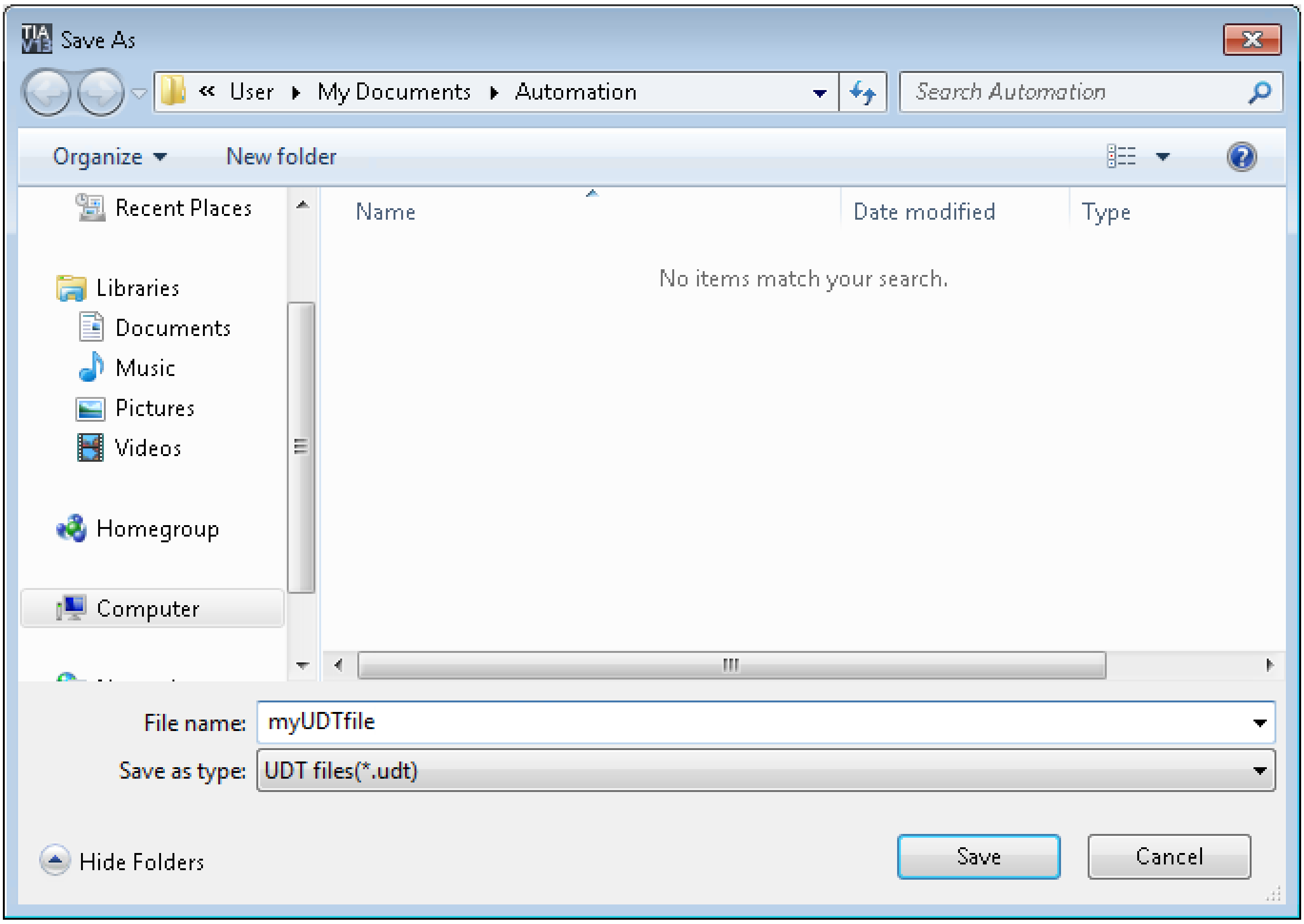
To create the file, expand **PLC data types** item from TIA Portal project tree and right click on the user defined structure. Then click on **Generate source from blocks**.



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.



File name: myUDTfile

Save as type: UDT files(*.udt)

This file will content all the PLC data types and it can be used for importing tags in Tag Editor.

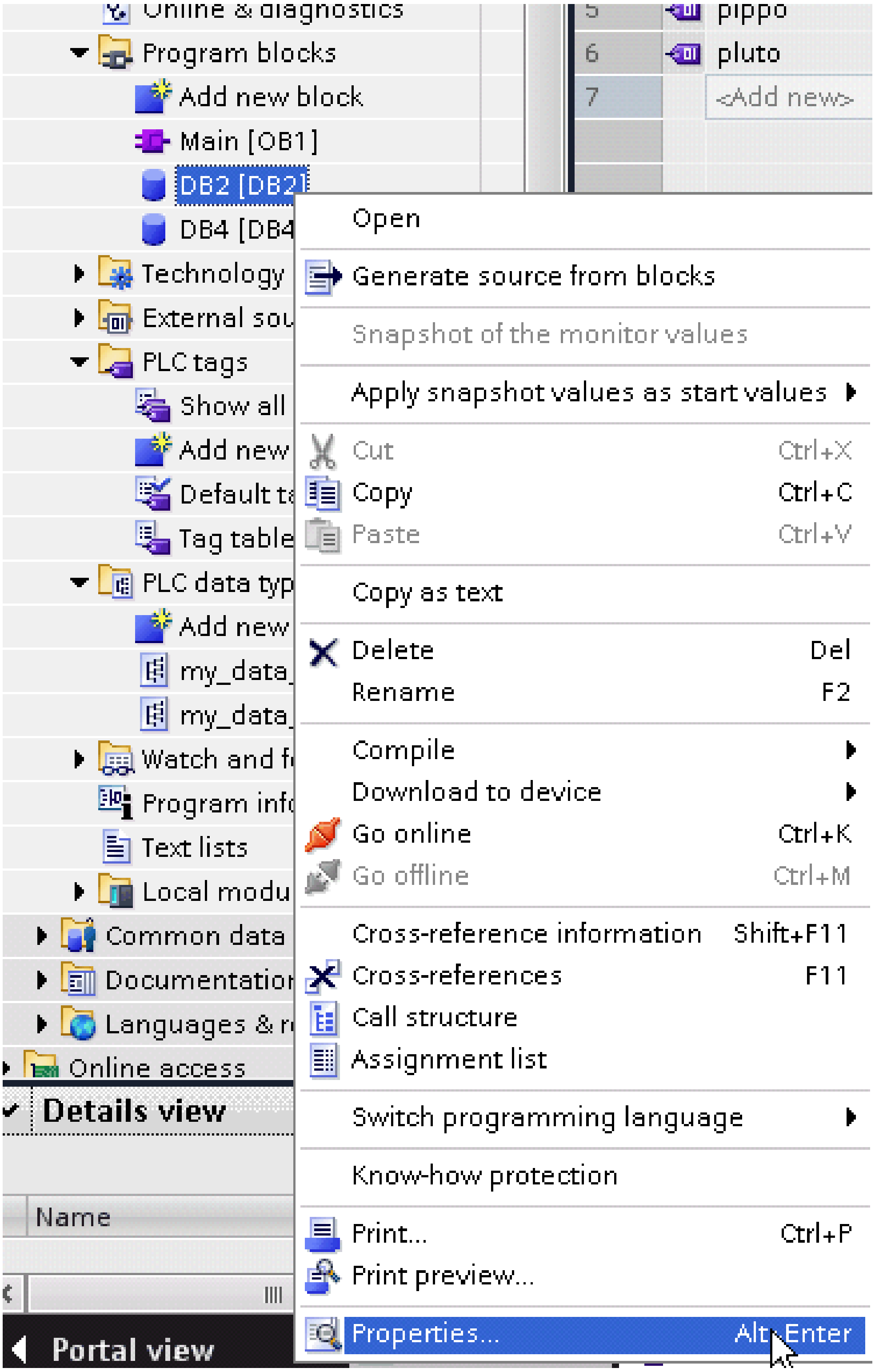
Check **Tag Import** chapter for more details.

Export using TIA Portal v10, v11, v12

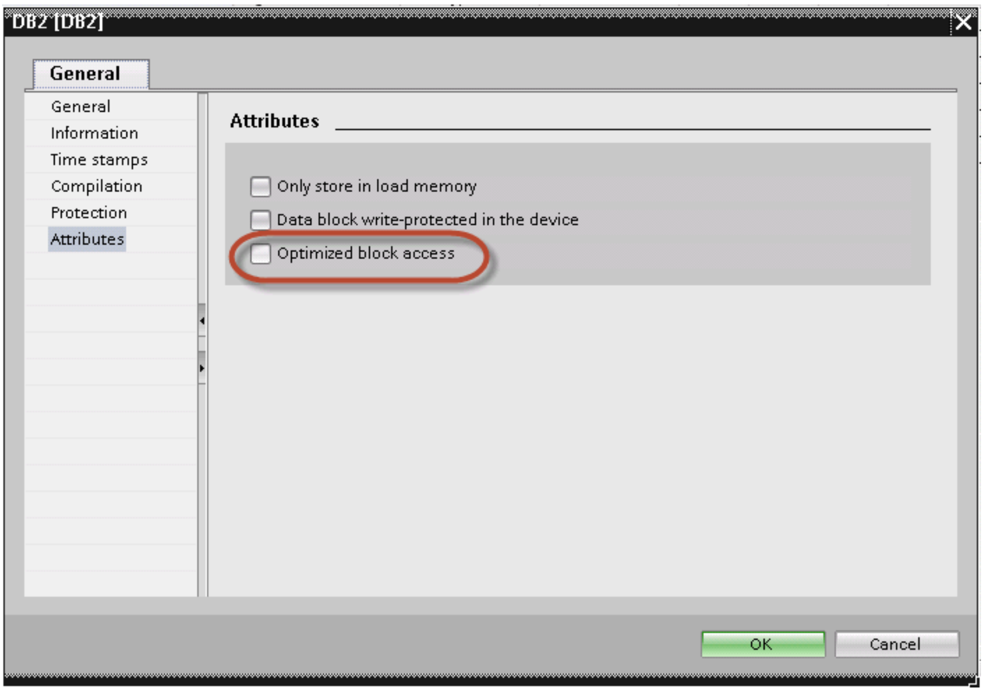
Exporting Program blocks

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**.

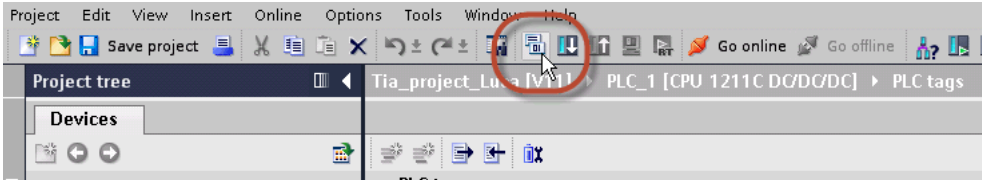


□ Only store in load memory

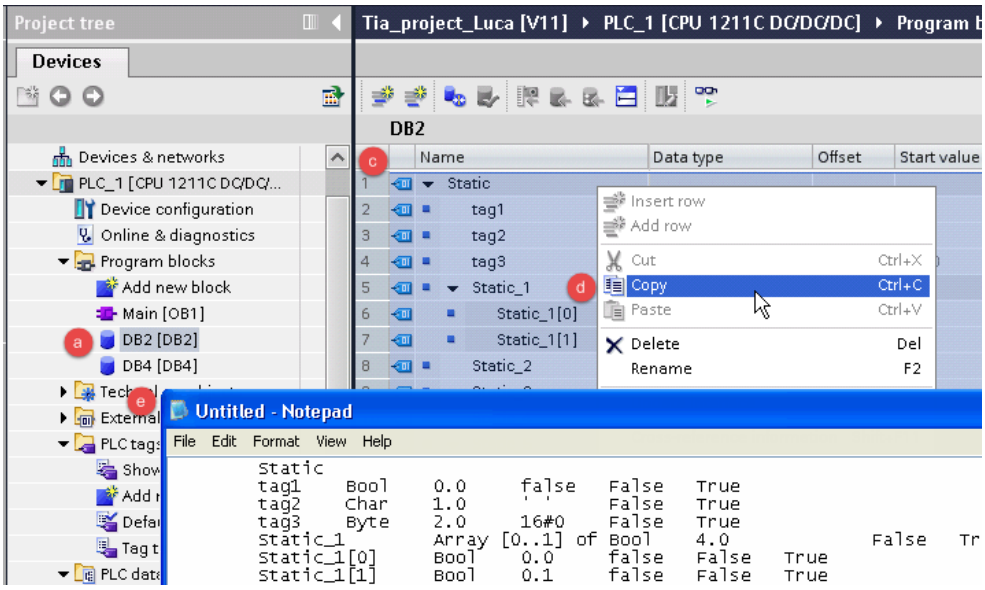
□ Data block write-protected in the device

□ 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.



DB2

	Name	Data type	Offset	Start value
1	Static			
2	tag1			
3	tag2			
4	tag3			
5	Static_1			
6	Static_1[0]			
7	Static_1[1]			
8	Static_2			

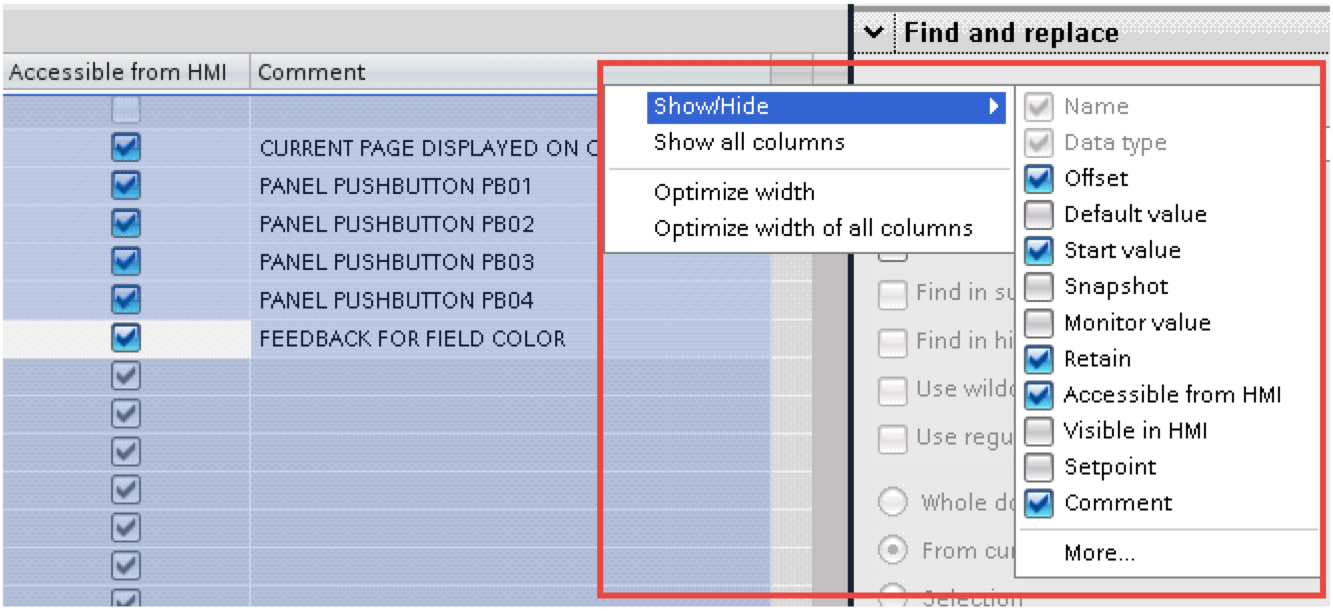
```
        Static
        tag1    Bool    0.0     false   False   True
        tag2    Char    1.0     ' '     False   True
        tag3    Byte    2.0     16#0    False   True
        Static_1        Array [0..1] of Bool    4.0             False   Tr
        Static_1[0]     Bool    0.0     false   False   True
        Static_1[1]     Bool    0.1     false   False   True
```

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



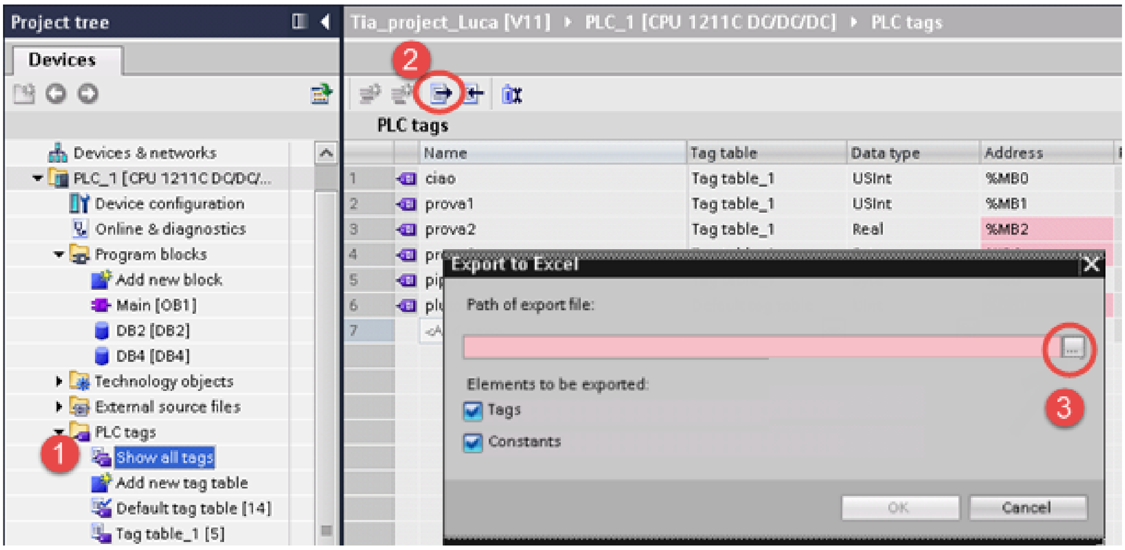
Accessible from HMI	Comment
□	
☑	CURRENT PAGE DISPLAYED ON C
☑	PANEL PUSHBUTTON PB01
☑	PANEL PUSHBUTTON PB02
☑	PANEL PUSHBUTTON PB03
☑	PANEL PUSHBUTTON PB04
☑	FEEDBACK FOR FIELD COLOR
☑	
☑	
☑	
☑	
☑	
☑	

☑ Name
☑ Data type
☑ Offset
□ Default value
☑ Start value
□ Snapshot
□ Monitor value
☑ Retain
☑ Accessible from HMI
□ Visible in HMI
□ Setpoint
☑ Comment
More...

Exporting PLC tags

An Excel file refers to PLC tags.

1. Double-click **Show all tags**: the tag table is displayed.



PLC tags

	Name	Tag table	Data type	Address
1	ciao	Tag table_1	USInt	%MB0
2	prova1	Tag table_1	USInt	%MB1
3	prova2	Tag table_1	Real	%MB2

Export to Excel

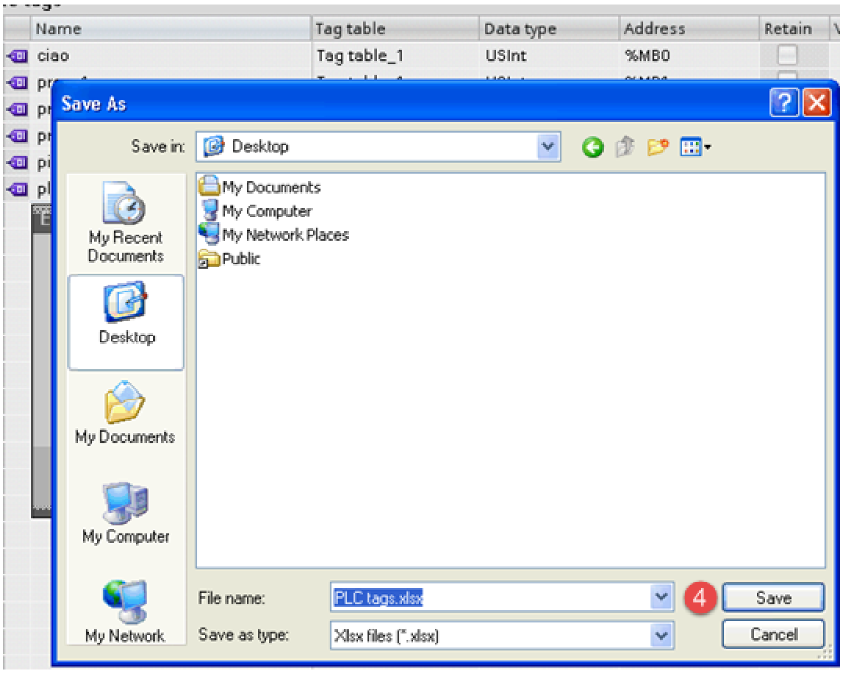
Path of export file:

Elements to be exported:

☑ Tags

☑ Constants

2. Click the **Export** button and browse for path file.
3. Define file name.
4. Click **Save** to confirm.



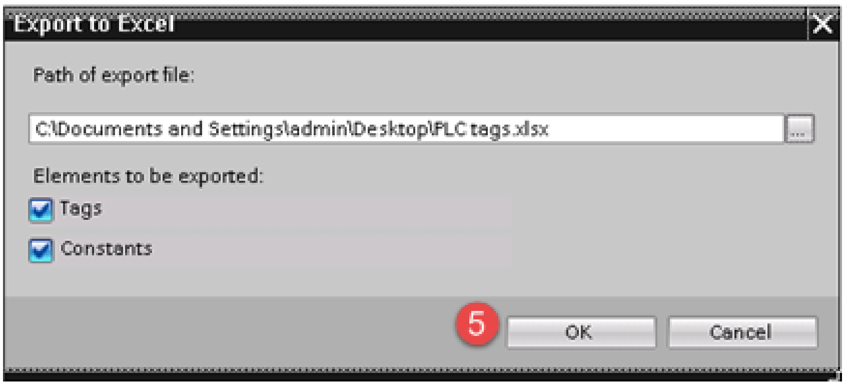
Name	Tag table	Data type	Address	Retain
ciao	Tag table_1	USInt	%MB0	□

Save in: Desktop

File name: PLC tags.xlsx

Save as type: Xlsx files (*.xlsx)

5. Click **OK** to export.



Export to Excel

Path of export file:

C:\Documents and Settings\admin\Desktop\PLC tags.xlsx

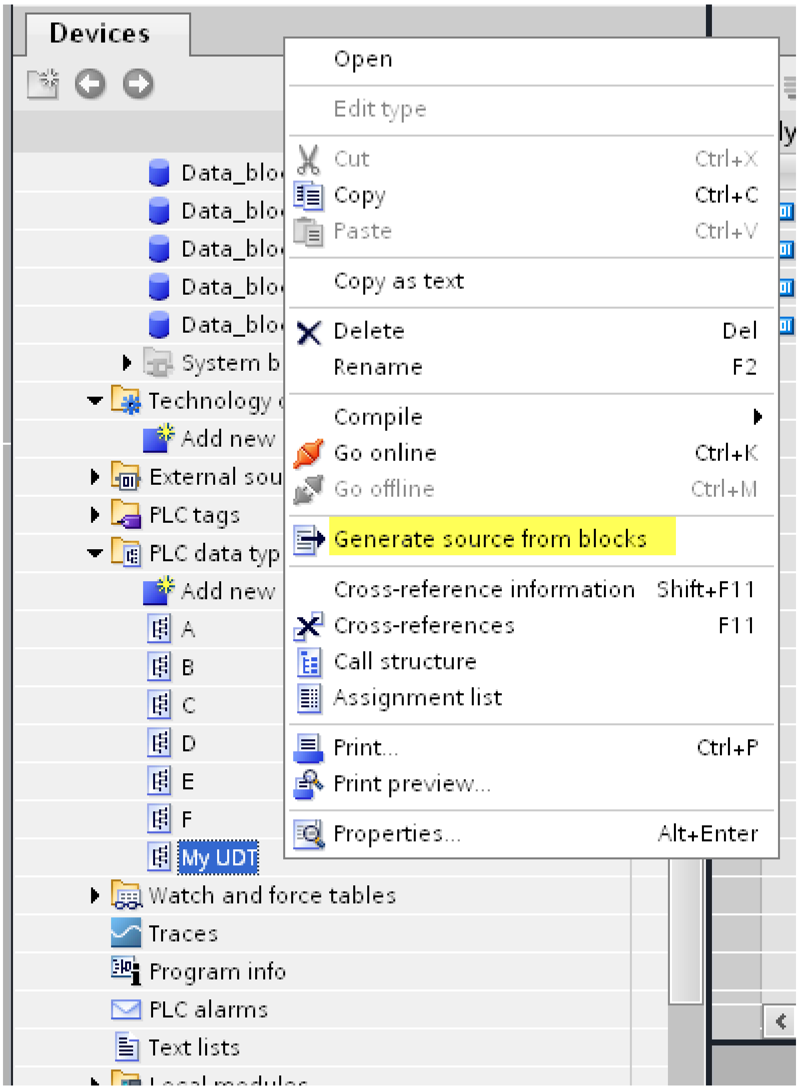
Elements to be exported:

☑ Tags

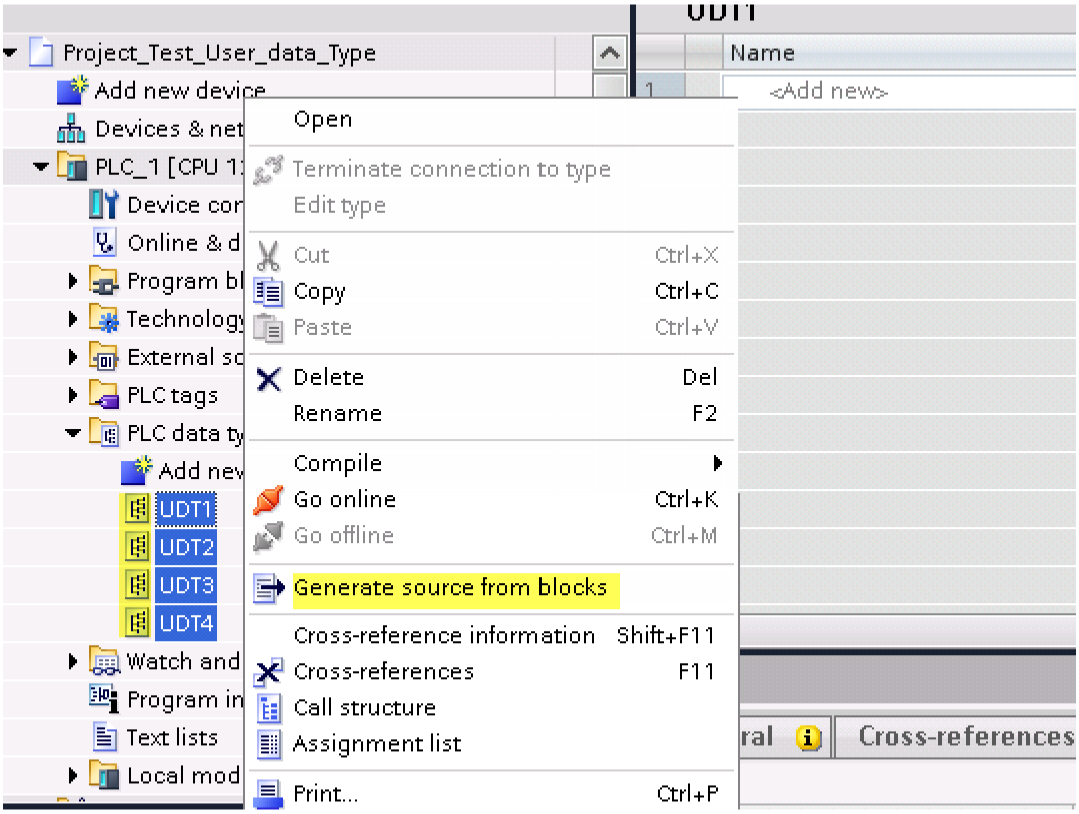
☑ Constants

Exporting PLC data types

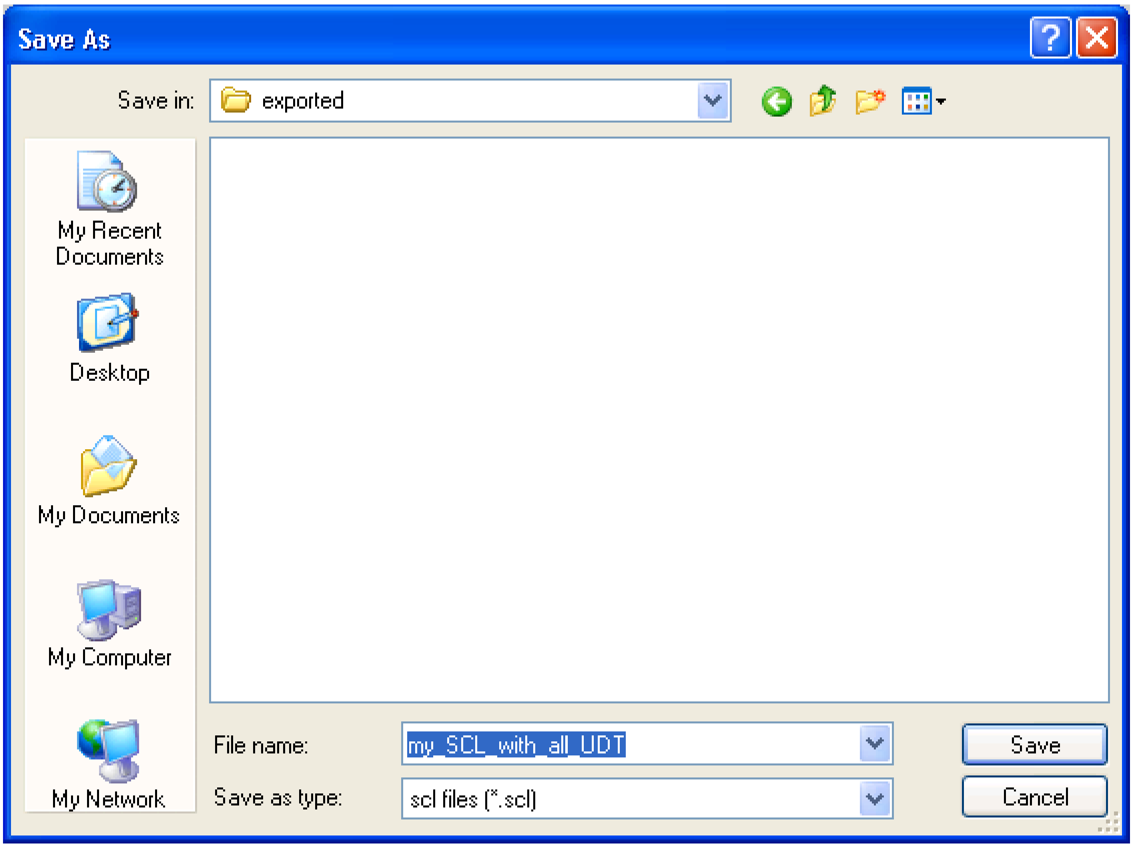
To create the file, expand **PLC data types** item from TIA Portal project tree and right click on the user defined structure. Then click on **Generate source from blocks**.



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.



Save As

Save in: exported

File name: my_SCL_with_all_UDT

Save as type: scl files (*.scl)

This file will content all the PLC data types and it can be used for importing tags in Tag Editor.

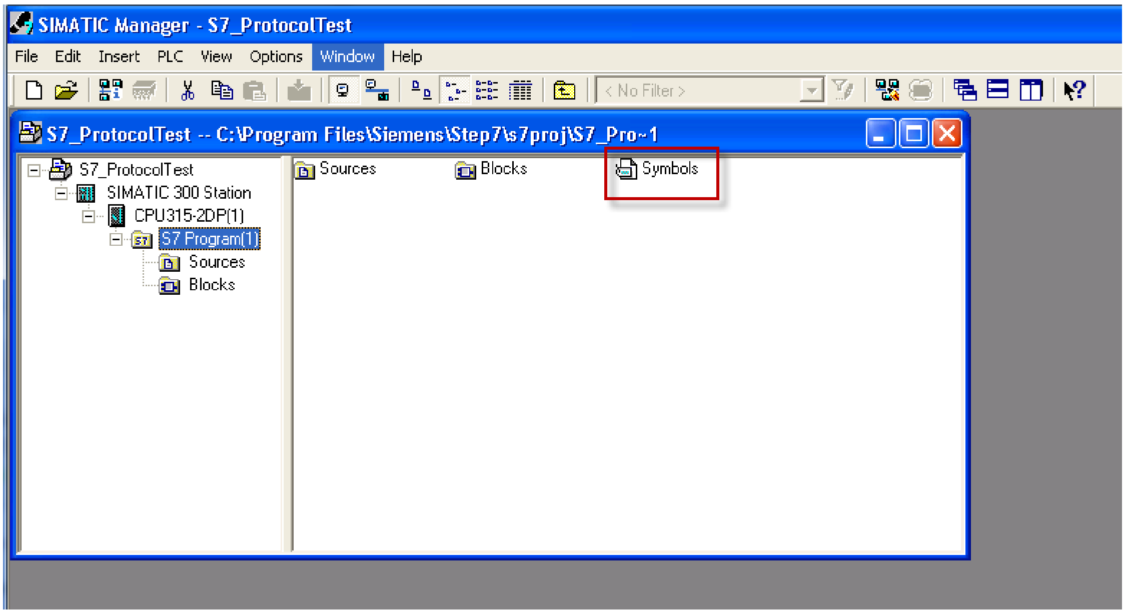
Check **Tag Import** chapter for more details.

Export using STEP7

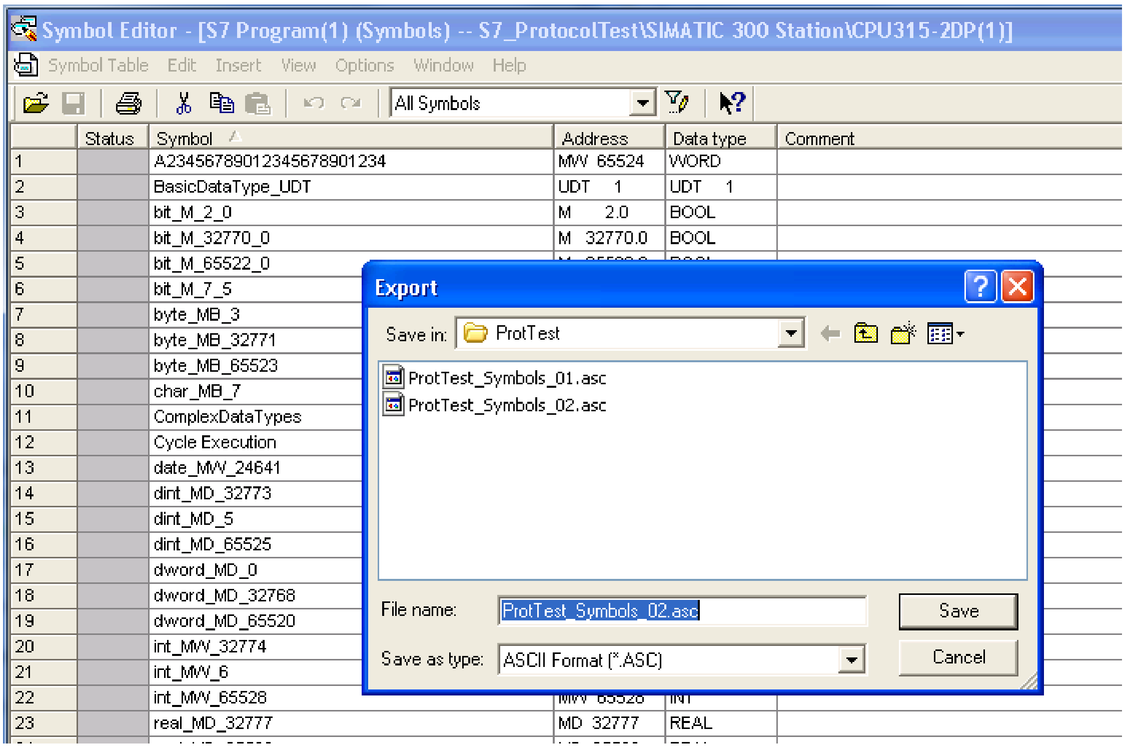
The Simatic S7 ETH Tag importer accepts symbol files (ASCII format .asc) and source files (.awl extension) created by the Simatic Step7. The symbol file can be previously exported using the Step7 symbol table utility.

Exporting Symbols table

Symbol files (.asc) can be exported from the symbol table utility.


SIMATIC Manager - S7_ProtocolTest
File Edit Insert PLC View Options Window Help
< No Filter >
S7_ProtocolTest -- C:\Program Files\Siemens\Step7\s7proj\S7_Pro~1
S7_ProtocolTest
SIMATIC 300 Station
CPU315-2DP(1)
S7 Program(1)
Sources
Blocks
Sources
Blocks
Symbols

1. From the **Symbol Table** menu in the Symbol Editor choose **Export**.
2. Assign a name and save the symbol table as ASCII file.


Symbol Editor - [S7 Program(1) (Symbols) -- S7_ProtocolTest\SIMATIC 300 Station\CPU315-2DP(1)]
Symbol Table Edit Insert View Options Window Help
All Symbols
Export
ProtTest_Symbols_01.asc
ProtTest_Symbols_02.asc
Save
Cancel

	Status	Symbol	Address	Data type	Comment
1		A23456789012345678901234	MW 65524	WORD	
2		BasicDataType_UDT	UDT 1	UDT 1	
3		bit_M_2_0	M 2.0	BOOL	
4		bit_M_32770_0	M 32770.0	BOOL	
5		bit_M_65522_0			
6		bit_M_7_5			
7		byte_MB_3			
8		byte_MB_32771			
9		byte_MB_65523			
10		char_MB_7			
11		ComplexDataTypes			
12		Cycle Execution			
13		date_MW_24641			
14		dint_MD_32773			
15		dint_MD_5			
16		dint_MD_65525			
17		dword_MD_0			
18		dword_MD_32768			
19		dword_MD_65520			
20		int_MW_32774			
21		int_MW_6			
22		int_MW_65528			
23		real_MD_32777	MD 32777	REAL	

Save in: ProtTest

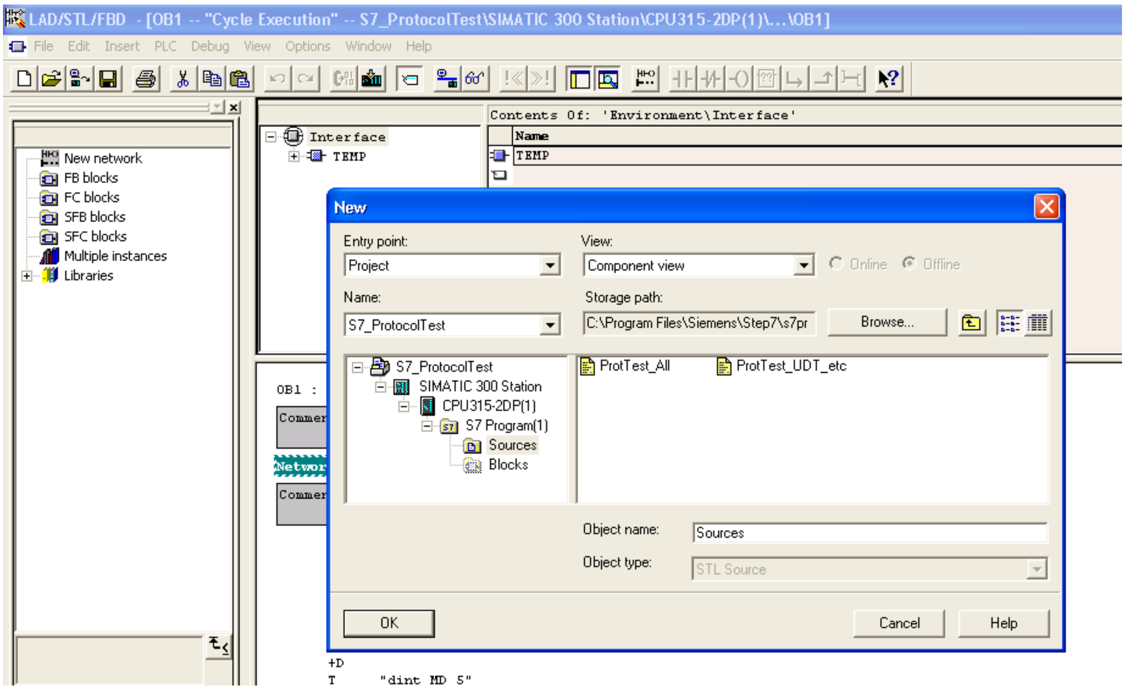
File name: ProtTest_Symbols_02.asc

Save as type: ASCII Format (*.ASC)

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:



LAD/STL/FBD - [OB1 -- "Cycle Execution" -- S7_ProtocolTest\SIMATIC 300 Station\CPU315-2DP(1)\...\OB1]

New

Entry point: Project

View: Component view ○ Online ● Offline

Name: S7_ProtocolTest

Storage path: C:\Program Files\Siemens\Step7\s7pr

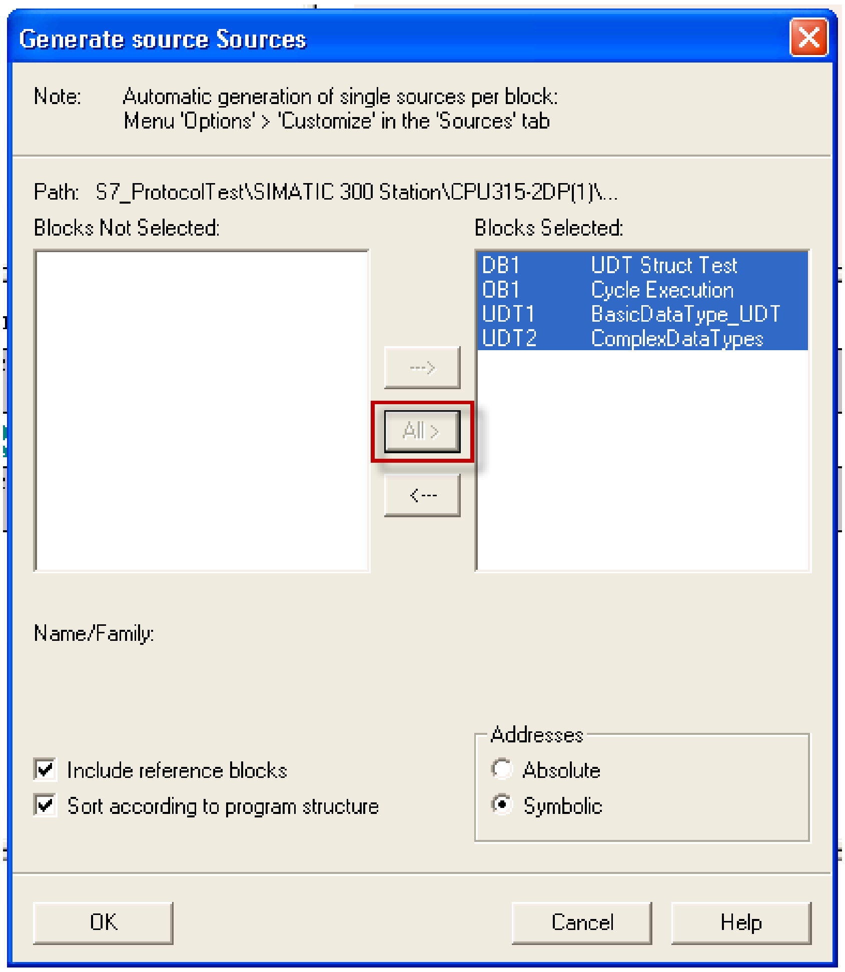
- S7_ProtocolTest
  - SIMATIC 300 Station
    - CPU315-2DP(1)
      - S7 Program(1)
        - Sources
        - Blocks

ProtTest_All ProtTest_UDT_etc

Object name: Sources

Object type: STL Source

1. Assign a name, "Sources" in the example, and click **OK**: the **Generate source Sources** dialog is displayed.



Generate source Sources

Note: Automatic generation of single sources per block:
Menu 'Options' > 'Customize' in the 'Sources' tab

Path: S7_ProtocolTest\SIMATIC 300 Station\CPU315-2DP(1)\...

Blocks Not Selected:

Blocks Selected:

DB1	UDT Struct Test
OB1	Cycle Execution
UDT1	BasicDataType_UDT
UDT2	ComplexDataTypes

Name/Family:

☑ Include reference blocks

☑ Sort according to program structure

Addresses

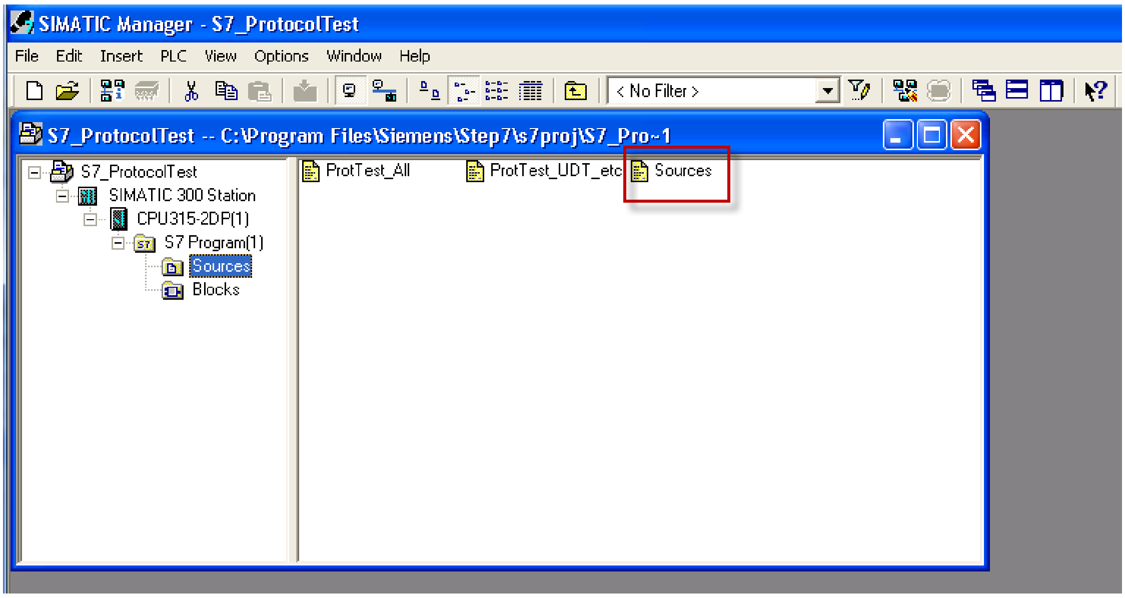
○ Absolute

● Symbolic

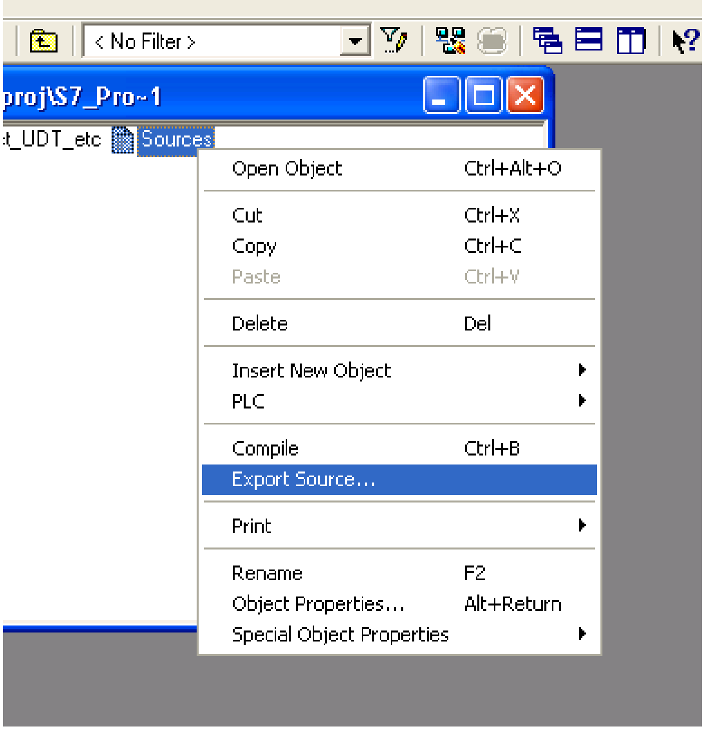
2. Click **All >** to generate source for all blocks.
3. Select the following options:

- **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.


SIMATIC Manager - S7_ProtocolTest
File Edit Insert PLC View Options Window Help
< No Filter >
S7_ProtocolTest -- C:\Program Files\Siemens\Step7\s7proj\S7_Pro~1
S7_ProtocolTest
SIMATIC 300 Station
CPU315-2DP(1)
S7 Program(1)
Sources
Blocks
ProtTest_All
ProtTest_UDT_etc
Sources

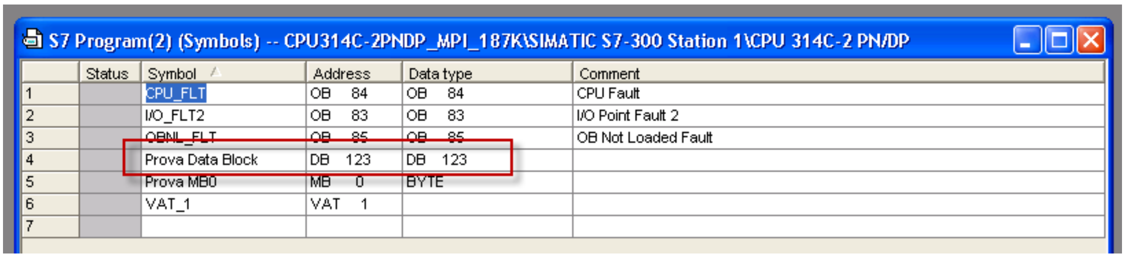
5. Right click on the object and select **Export Sources**.



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.



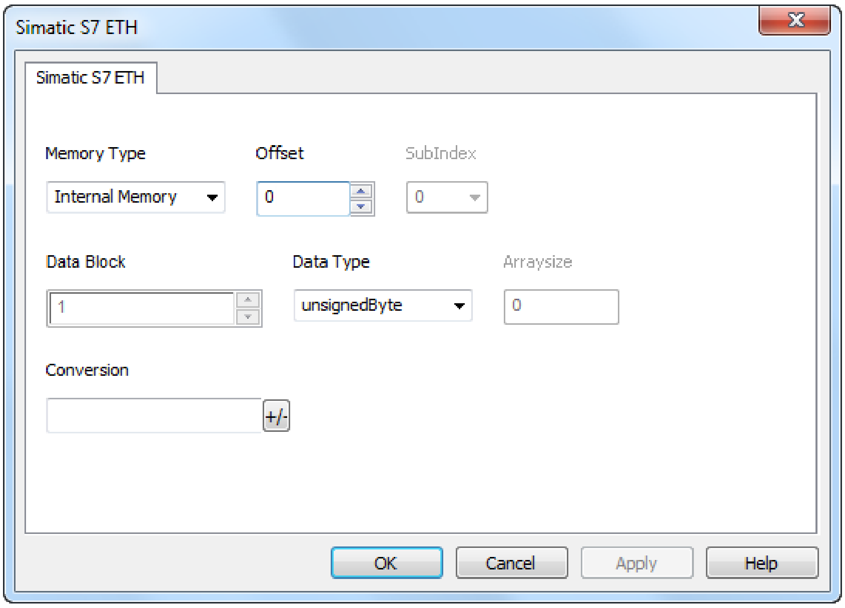
S7 Program(2) (Symbols) -- CPU314C-2PNDP_MPI_187K\SIMATIC S7-300 Station 1\CPU 314C-2 PN/DP

	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 MB0	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

Memory Type: Internal Memory

Offset: 0

SubIndex: 0

Data Block: 1

Data Type: unsignedByte

Arraysize: 0

Conversion:

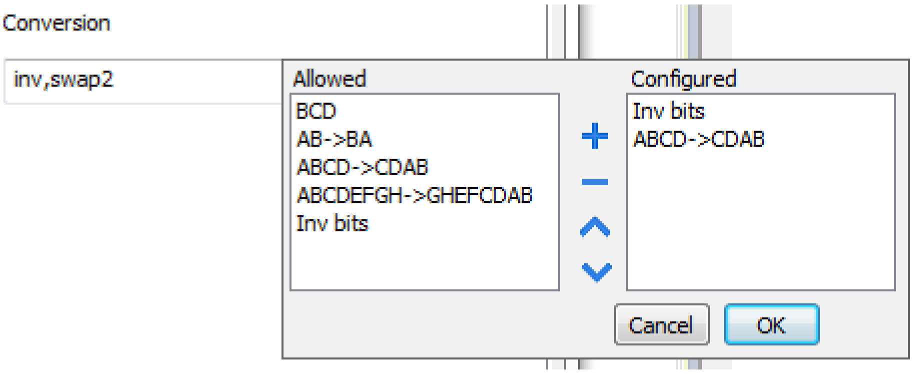
<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td><b>Memory Type</b></td><td>Area of PLC where tag is located.
<table>
<tr><th>Data Type</th><th>Simatic Type</th></tr>
<tr><td>Internal Memory</td><td>M</td></tr>
<tr><td>Data Block</td><td>DB</td></tr>
<tr><td>Input</td><td>I (E)</td></tr>
<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>
</table></td></tr>
<tr><td><b>Offset</b></td><td>Offset address where tag is located.</td></tr>
<tr><td><b>SubIndex</b></td><td>Resource offset within the register.</td></tr>
<tr><td><b>Data Block</b></td><td>Data block number for Data Block Memory Type.</td></tr>
<tr><td><b>Data Type</b></td><td>Available data types:
<ul>
<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>
See "Programming concepts" section in the main manual.
Note: To define arrays, select one of Data Type format followed by square brackets.</td></tr>
</table>

<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td><b>Arraysize</b></td><td>

- 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.</td></tr>
<tr><td><b>Conversion</b></td><td>

Conversion to be applied to the tag.



Conversion

inv,swap2

Configured

Inv bits
ABCD->CDAB

Depending on data type selected, the **Allowed** list shows one or more conversions, listed below.

<table>
<tr><th>Value</th><th>Description</th></tr>
<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 -> 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 -> CDAB</b></td><td>Swap bytes of a word.
<i>Example:</i>
9ACC → CC9A (in hexadecimal format)
39628 → 52378 (in decimal format)</td></tr>
<tr><td><b>ABCDEFGH -></b></td><td>Swap bytes of a double word.</td></tr>
</table>
</td></tr>
</table>

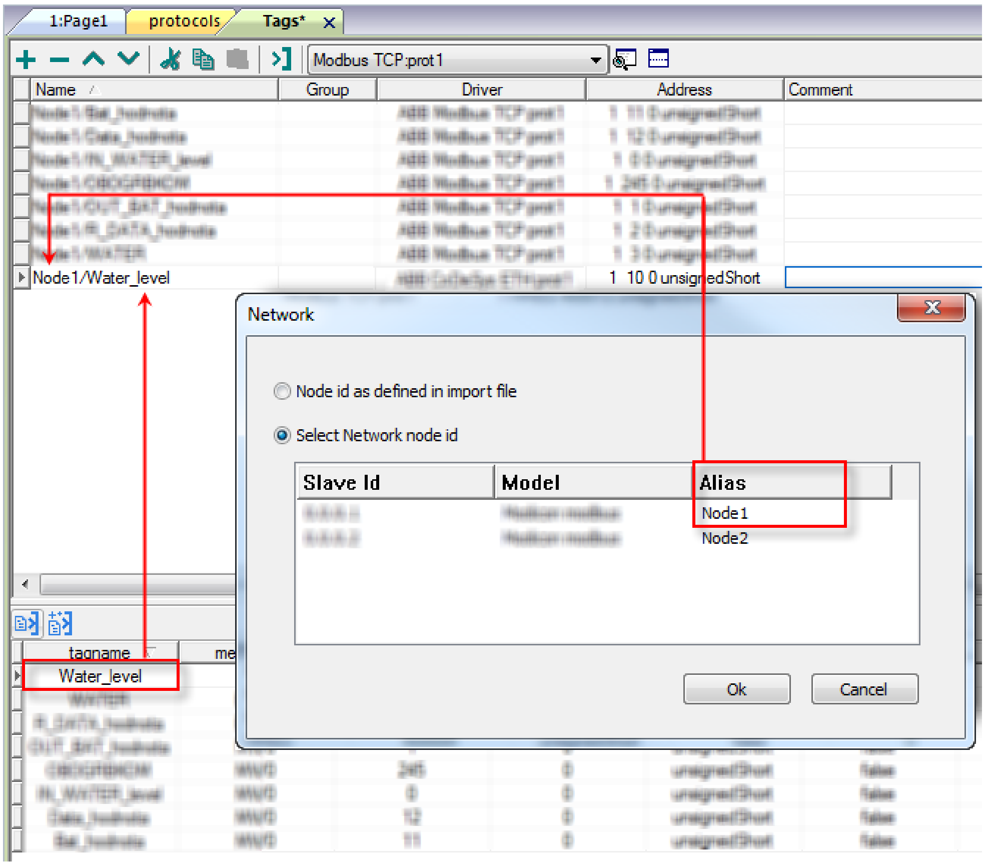
<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td><b>GHEFCDAB</b></td><td><i>Example:</i>
32FCFF54 → 54FFFC32 (in hexadecimal format)
855441236 → 1426062386 (in decimal format)</td></tr>
<tr><td><b>ABC...NOP -> OPM...DAB</b></td><td>Swap bytes of a long word.
Example:
142.366 → -893553517.588905 (in decimal format)
0 10000000110
0001110010111011011001000101101000011100101011000001
→
1 10000011100
1010101000010100010110110110110010110110000100111101
(in binary format)</td></tr>
<tr><td><b>BCD</b></td><td>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)</td></tr>
<tr><td><b>S5timer(BCD)</b></td><td>Used to support S5timer. Check <b>Simatic S5timer special data type</b> for more details.</td></tr>
<tr><td><b>S5timer(BIN)</b></td><td>Legacy transformation for S5timer in binary format.</td></tr>
</table>
<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>
</td></tr>
</table>

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.



Slave Id	Model	Alias
		Node1
		Node2

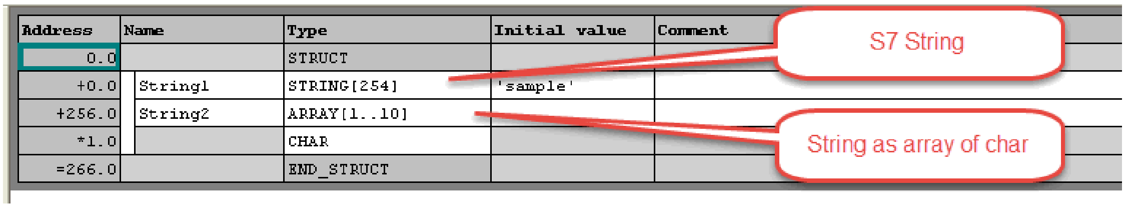
Note: Aliasing tag names is only available for imported tags. Tags added manually in the Tag Editor cannot have the Alias prefix in the tag name.
The Alias string is attached at the time of tag import. If you modify the Alias string after the tag import has been completed, there will be no effect on 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.

String data type

In Simatic S7 PLC two different types of tags manage string variables:

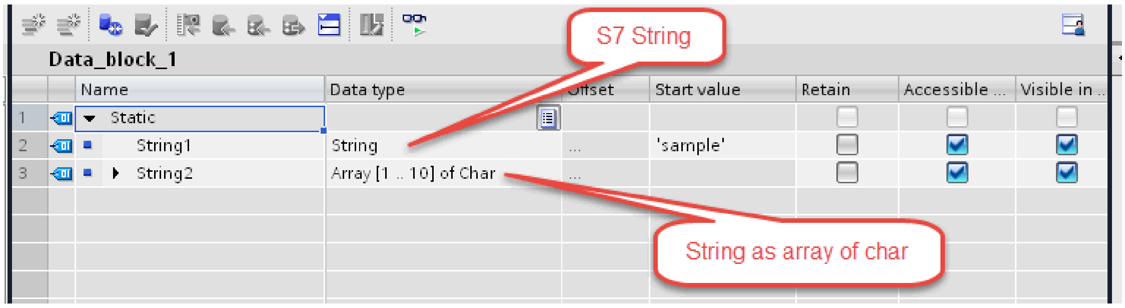
- as Array [1..xx] of characters,
- as String[xx].

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		

TIA Portal string declaration is shown in this example:



Data_block_1

	Name	Data type	Offset	Start value	Retain	Accessible ...	Visible in ...
1	Static				☐	☐	☐
2	String1	String	...	'sample'	☐	☑	☑
3	String2	Array [1 .. 10] of Char	...		☐	☑	☑

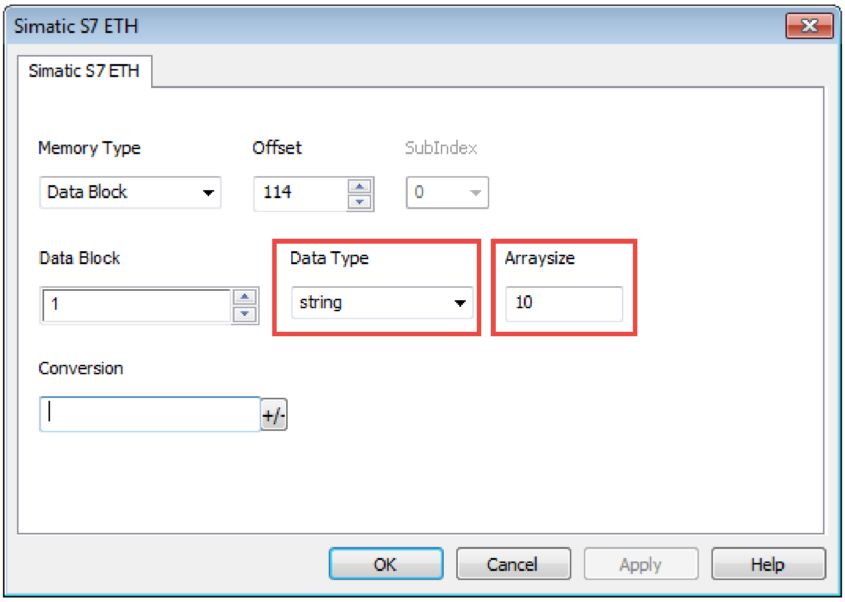
S7 String

String as array of char

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.



Memory Type: Data Block

Offset: 114

SubIndex: 0

Data Block: 1

Data Type: string

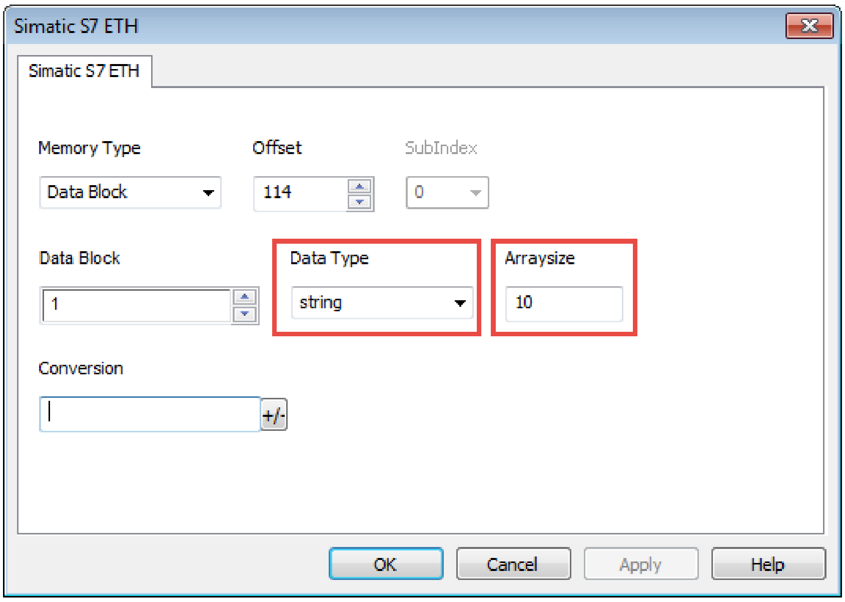
Arraysize: 10

Conversion:

2. Select **string** as **Data Type.**
3. Enter string length in **Arraysize**.
4. Click **OK** to confirm.

To add a string data type:

1. Press the **+** in the Tag Editor.



Memory Type: Data Block

Offset: 114

SubIndex: 0

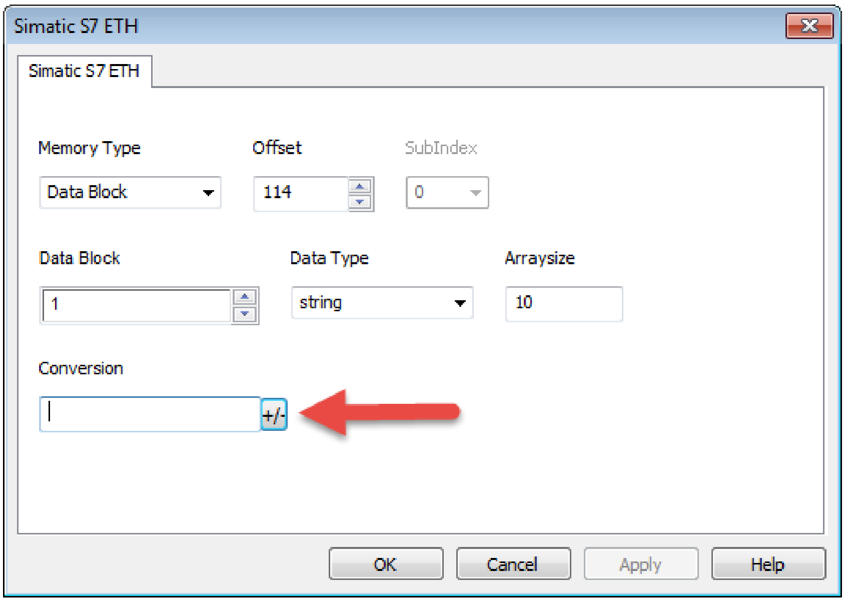
Data Block: 1

Data Type: string

Arraysize: 10

Conversion:

2. Select **string** as **Data Type.**
3. Enter string length in **Arraysize**.
4. Click **+/-** to open the Conversion dialog.



Memory Type: Data Block

Offset: 114

SubIndex: 0

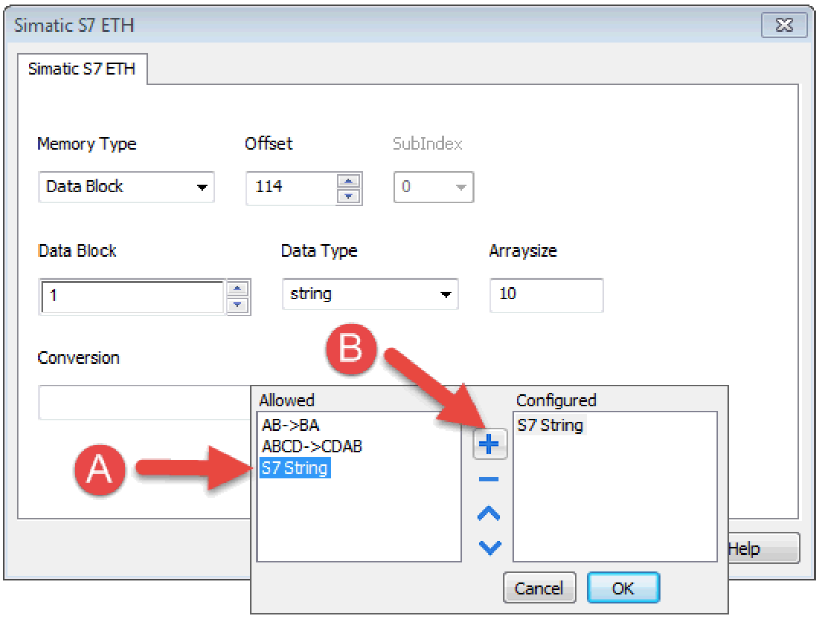
Data Block: 1

Data Type: string

Arraysize: 10

Conversion:

5. In the conversion dialog select the **S7 String** conversion type.



Simatic S7 ETH

Memory Type: Data Block
Offset: 114
SubIndex: 0
Data Block: 1
Data Type: string
Arraysize: 10
Conversion

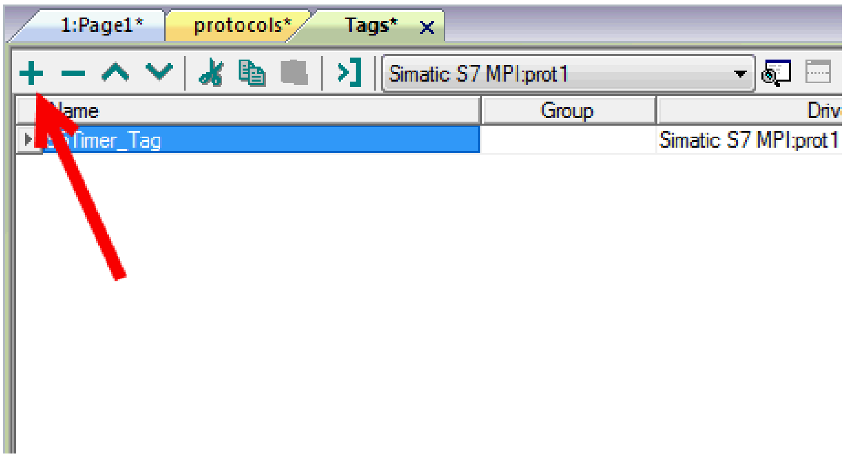
Allowed: AB->BA, ABCD->CDAB, S7 String
Configured: S7 String

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

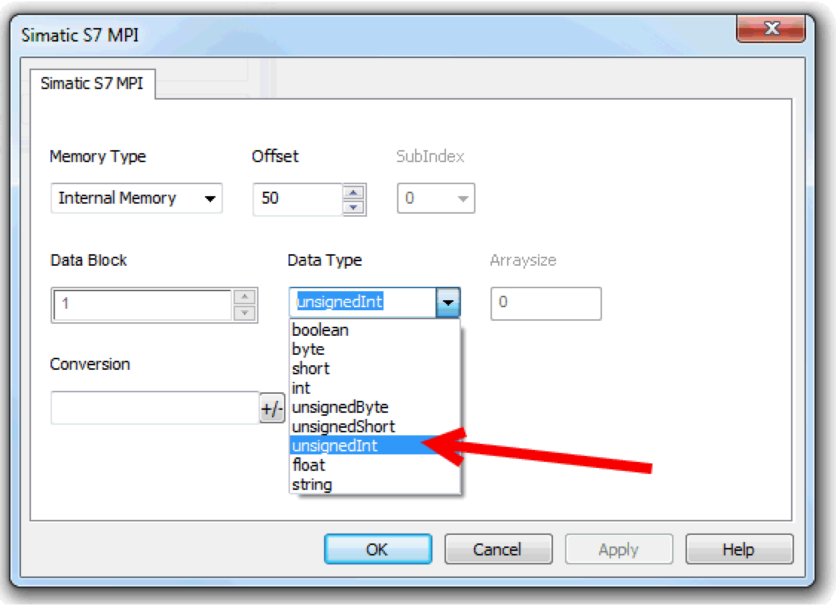
Simatic drivers support a special data type, the S5Timer data type.
The tag must be configured with a specific data type and a conversion must be applied to the tag to correctly read/write a Simatic S5Timer Variable.

1. In the Tag Editor click **+** to add a tag.



Name	Group	Driv
Timer_Tag		Simatic S7 MPI:prot1

2. Select **unsignedInt** as **Data Type.**


Simatic S7 MPI

Memory Type: Internal Memory

Offset: 50

SubIndex: 0

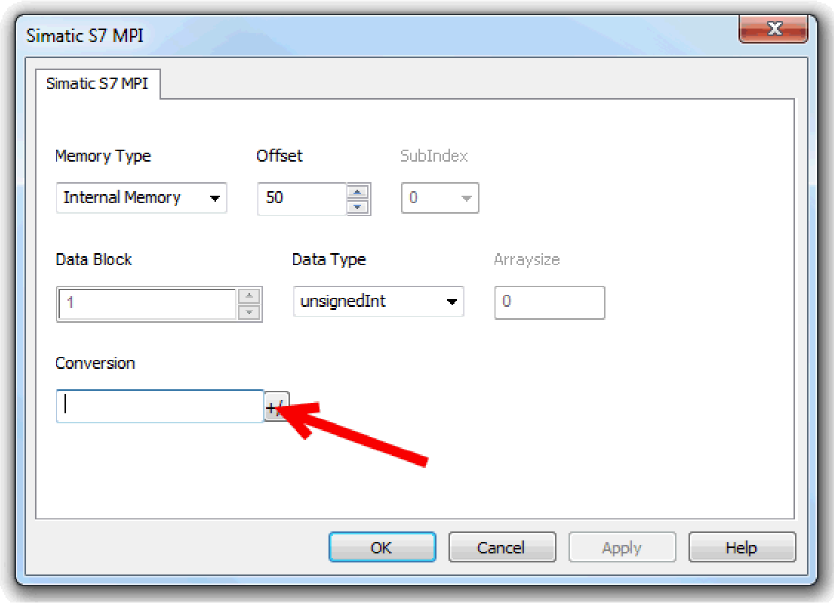
Data Block: 1

Data Type: unsignedInt

Arraysize: 0

Conversion:

3. Click **+/-** to open the Conversion dialog.


Simatic S7 MPI

Memory Type: Internal Memory

Offset: 50

SubIndex: 0

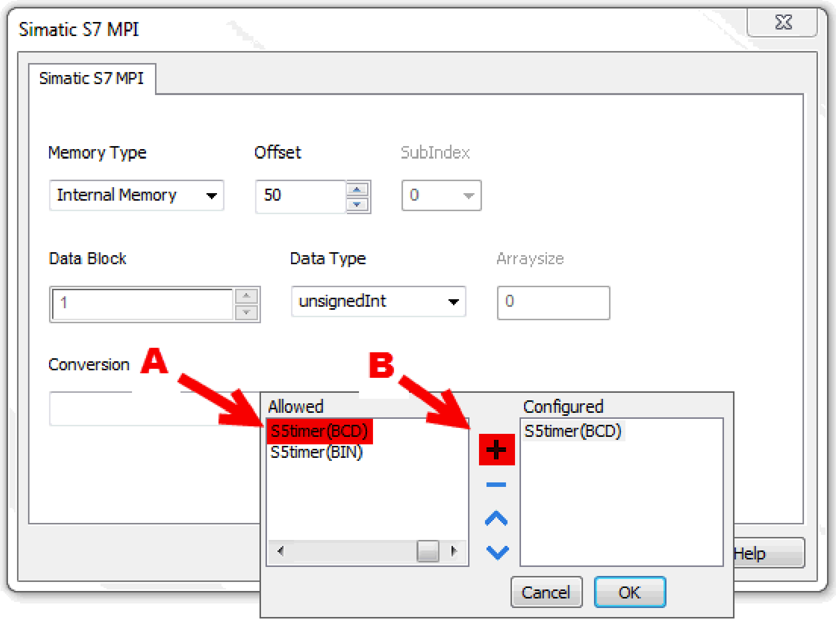
Data Block: 1

Data Type: unsignedInt

Arraysize: 0

Conversion:

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.

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.

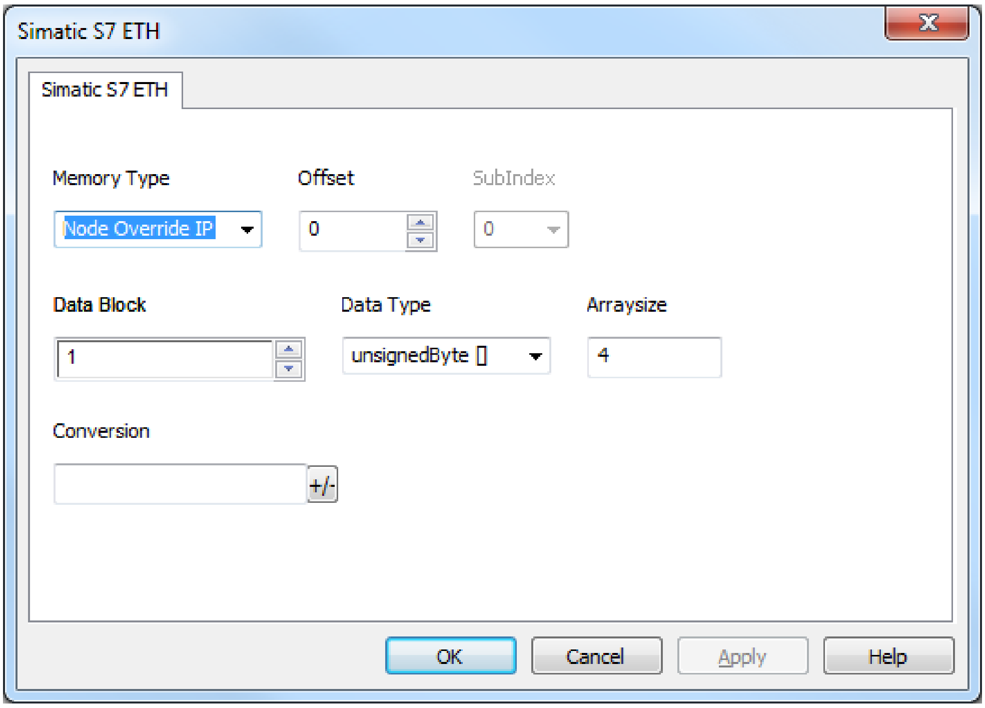
Node Override IP	PLC operation
0.0.0.0	Communication with the controller is stopped, no request frames are generated anymore.
Different from 0.0.0.0	It is interpreted as node IP override and the target IP address is replaced runtime with the new value.

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.



Memory Type: Node Override IP

Offset: 0

SubIndex: 0

Data Block: 1

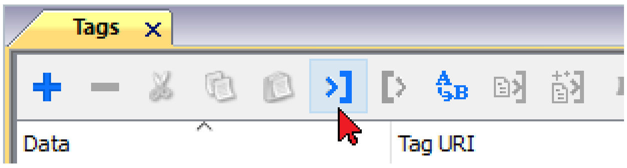
Data Type: unsignedByte []

Arraysize: 4

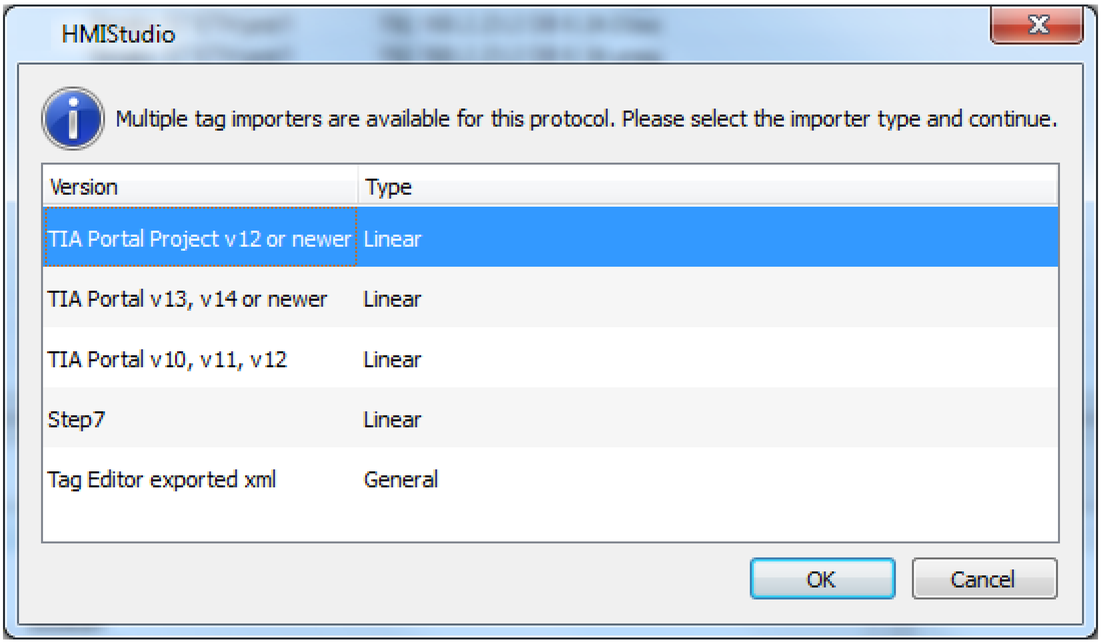
Conversion:

Tag Import

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

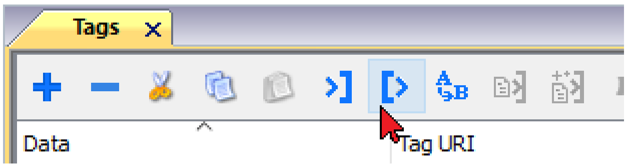


The following dialog shows which importer type can be selected.



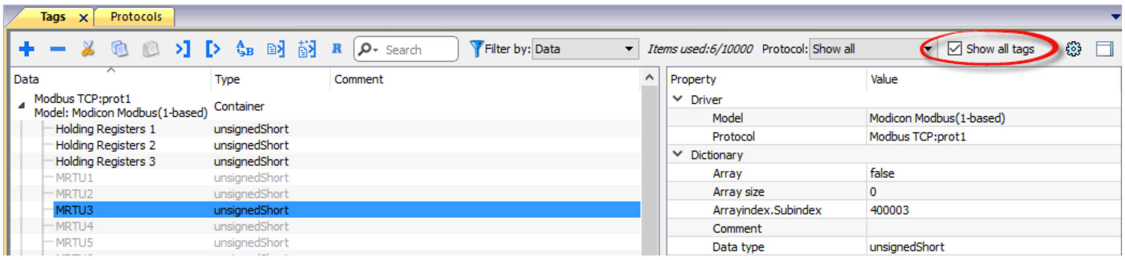
Multiple tag importers are available for this protocol. Please select the importer type and continue.

Version	Type
TIA Portal Project v12 or newer	Linear
TIA Portal v13, v14 or newer	Linear
TIA Portal v10, v11, v12	Linear
Step7	Linear
Tag Editor exported xml	General

Importer	Description
TIA Portal Project v12 or newer **Linear**	Allows to import the whole TIA Portal project file using **.apxx** 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.
TIA Portal v13, v14 or newer **Linear**	Allows to import: • Program blocks using **.db** file • PLC tags using **.xlsx** file • PLC data types using **.udt** file Check **Export using TIA Portal v13, v14 or newer** for more details. All variables will be displayed at the same level.
TIA Portal v10, v11, v12 **Linear**	Allows to import: • Program blocks using **.tia** file • PLC tags using **.xlsx** file • PLC data types using **.scl** file Check **Export using TIA Portal v10, v11, v12** for more details. All variables will be displayed at the same level.
Step7 **Linear**	Allows to import: • Symbols table **.asc** file • Sources using **.awl** file Check **Export using STEP7** for more details. 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. 

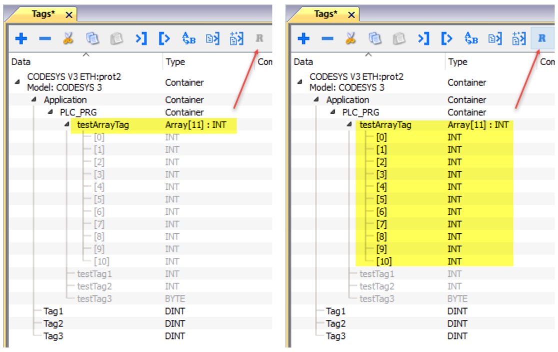
Once the importer has been selected, locate the symbol file and click **Open**.

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.



Data	Type	Comment
Modbus TCP:prot1 Model: Modicon Modbus(1-based)	Container	
Holding Registers 1	unsignedShort	
Holding Registers 2	unsignedShort	
Holding Registers 3	unsignedShort	
MRTU1	unsignedShort	
MRTU2	unsignedShort	
MRTU3	unsignedShort	
MRTU4	unsignedShort	
MRTU5	unsignedShort	

Property	Value
Driver	
Model	Modicon Modbus(1-based)
Protocol	Modbus TCP:prot1
Dictionary	
Array	false
Array size	0
Arrayindex.Subindex	400003
Comment	
Data type	unsignedShort

Toolbar item	Description
	Import Tag(s). Select tags to be imported and click on this icon to add tags from tag dictionary to the project
	Update Tag(s). Click on this icon to update the tags in the project, due a new dictionary import.
R	Check this box to import all sub-elements of a tag. Example of both checked and unchecked result:  Data | Type | Con CODESYS V3 ETH:prot2 Model: CODESYS 3 | Container Application | Container PLC_PRG | Container testArrayTag | Array[11] : INT [0] | INT [1] | INT [2] | INT [3] | INT [4] | INT [5] | INT [6] | INT [7] | INT [8] | INT [9] | INT [10] | INT testTag1 | INT testTag2 | INT testTag3 | BYTE Tag1 | DINT Tag2 | DINT Tag3 | DINT
Search Filter by: Tag name	Searches tags in the dictionary basing on filter combo-box item selected.

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
NAK	The controller replies with a not acknowledge.	-
Timeout	A request is not replied within the specified timeout period.	Check if the controller is connected and properly configured to get network access.

Error	Cause	Action
Invalid response	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.
General Error	Unidentifiable error. Should never be reported.	Contact technical support.

Simatic S7 MPI

HMI products support direct Siemens MPI communication without any additional module.

The driver supports the standard communication speed 187Kbit/s.

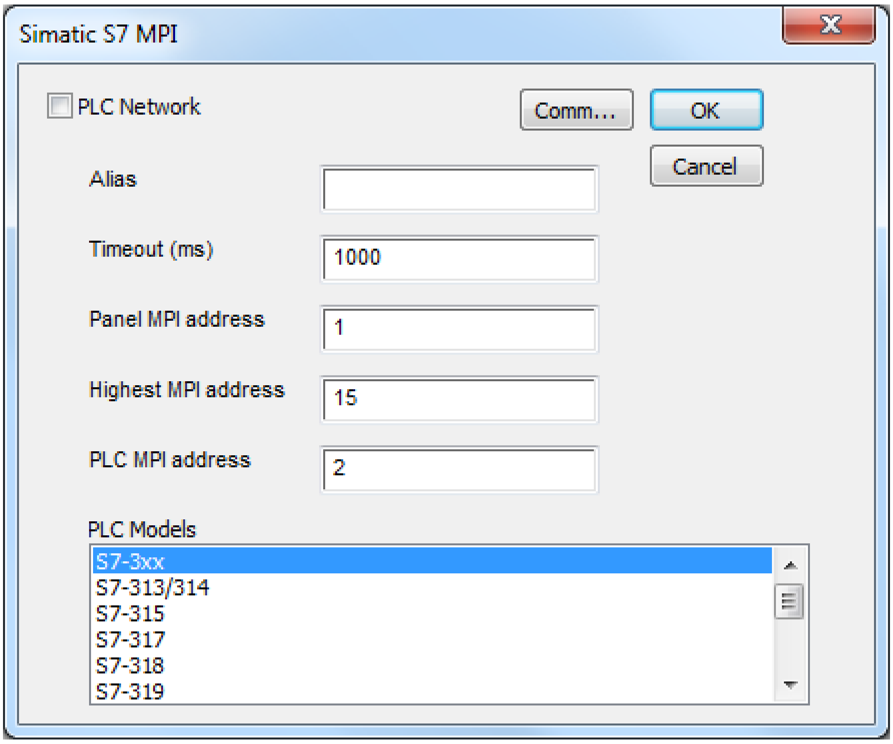
here is a minimum requirement also for the version of operating system running in the HMI (this is normally referenced as BSP version). See in user manual how to read the BSP version with the System Settings menu. The minimum requirements are shown in the following table.

Platform	BSP Version
UN30/31	v1.38 or newer
UN65/UN71	v1.0.300 or newer
UN60/UN70	v1.0.413 or newer
UN73	v1.0.142 or newer

Protocol Editor Settings

Add [+] a driver in the Protocol editor and select the “Simatic S7 MPI” protocol from the list of available protocols.

The protocol type can be selected from the dedicated combo box in the dialog.


Simatic S7 MPI

□ PLC Network

Alias

Timeout (ms) 1000

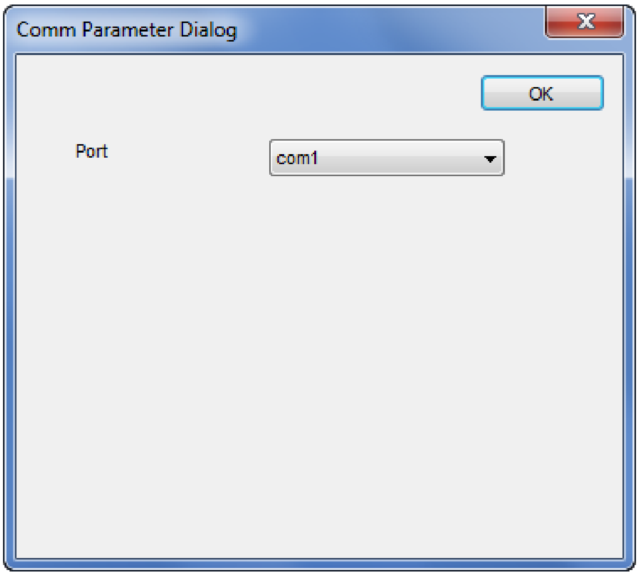
Panel MPI address 1

Highest MPI address 15

PLC MPI address 2

PLC Models

S7-3xx
S7-313/314
S7-315
S7-317
S7-318
S7-319

Element	Description
Alias	Name to be used to identify nodes in the plc network configuration. The name will be added as a prefix to each tag name imported for each network node.
Timeout (ms)	Defines the time inserted by the protocol between two retries of the same message in case of missing response from controller. Value is expressed in milliseconds.
Panel MPI Address	MPI node number assigned to the device.
Highest MPI Address	The highest node number in the MPI network where the device is operating and communicating.
PLC MPI Address	The MPI address of the controller to which the device needs to communicate.
PLC Models	List of compatible controller models. Make sure to select the correct PLC model in this list when configuring the protocol.
Comm…	Click on this button to configure the serial port on the device to be used as MPI port (see example in the following figure)  Comm Parameter Dialog Port: com1 Communication parameters for Simatic S7 MPI are fixed at: • Baud rate=187500 • Parity=Even • Data=bits8 • Stop=bit1 On UN20: • com1 is the HMI port labeled “PLC”, • com2 is the HMI port labeled “PC/Printer”

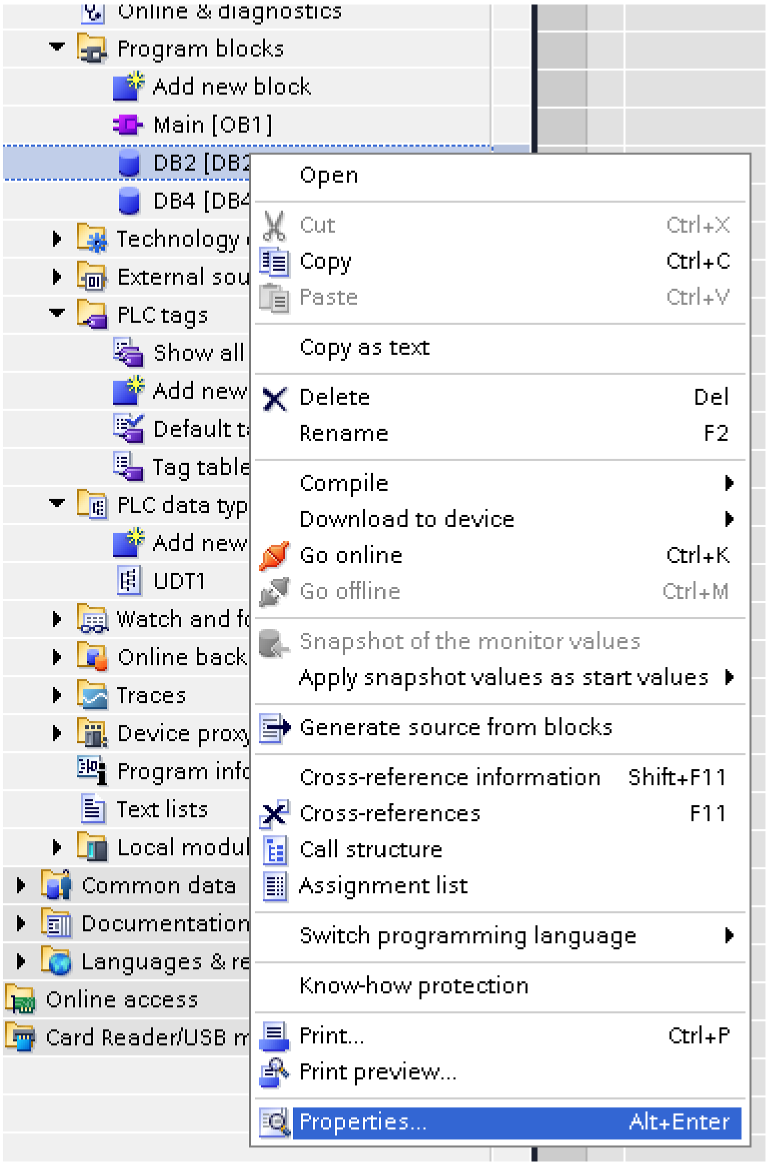
Element	Description
	On UN31 or UN30: • com1 is the integrated serial port, • com2 is an add-on module plugged in Slot#1 or #2 • com3 is an add-on module plugged in Slot#3 or #4 Note: The connection between device and PLC can be made with the following two options: 1. Creating a custom cable following the scheme provided with document CA255 “*eTOP400/500 serie PLC Port to MPI Port*” 2. Using a standard MPI cable with ADP-0001 “*MPI wiring adapter*”
PLC Network	The protocol supports connection to multiple controllers. To enable this option, check the "PLC Network" check box and enter the configuration per each controller node.

Direct Import of TIA Portal project

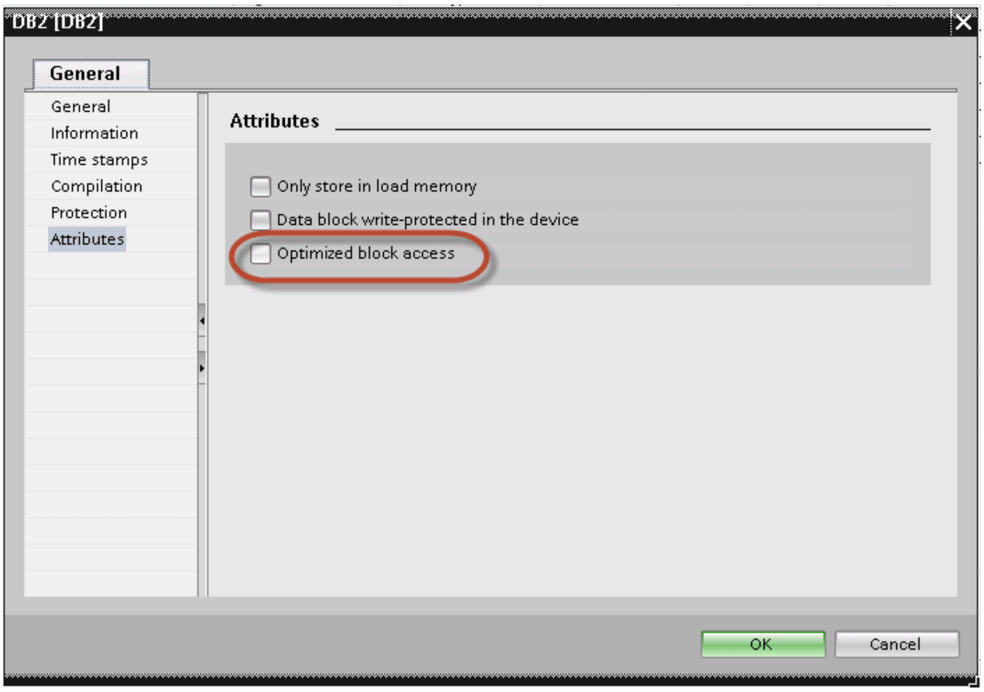
It is possible to import TIA Portal variables directly from TIA Portal project, by selecting "TIA Portal Project v12 or newer" from import selection (refer to "Tag Import" chapter).

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**.



Attributes

□ Only store in load memory

□ Data block write-protected in the device

□ Optimized block access

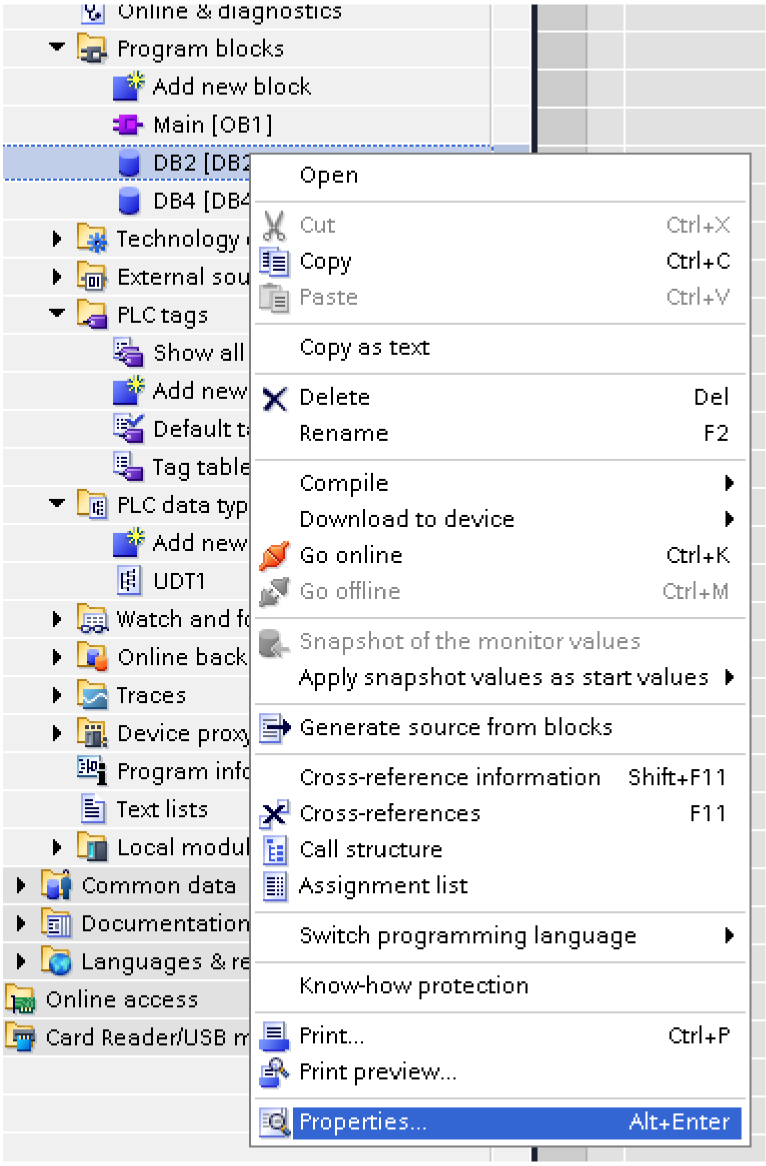
Note: If the options **Optimized block access** is not enabled (checkbox grayed out) this might mean that the Data Block is an "instance DB" linked to an "optimized access FB".

Export using TIA Portal v13, v14 or newer

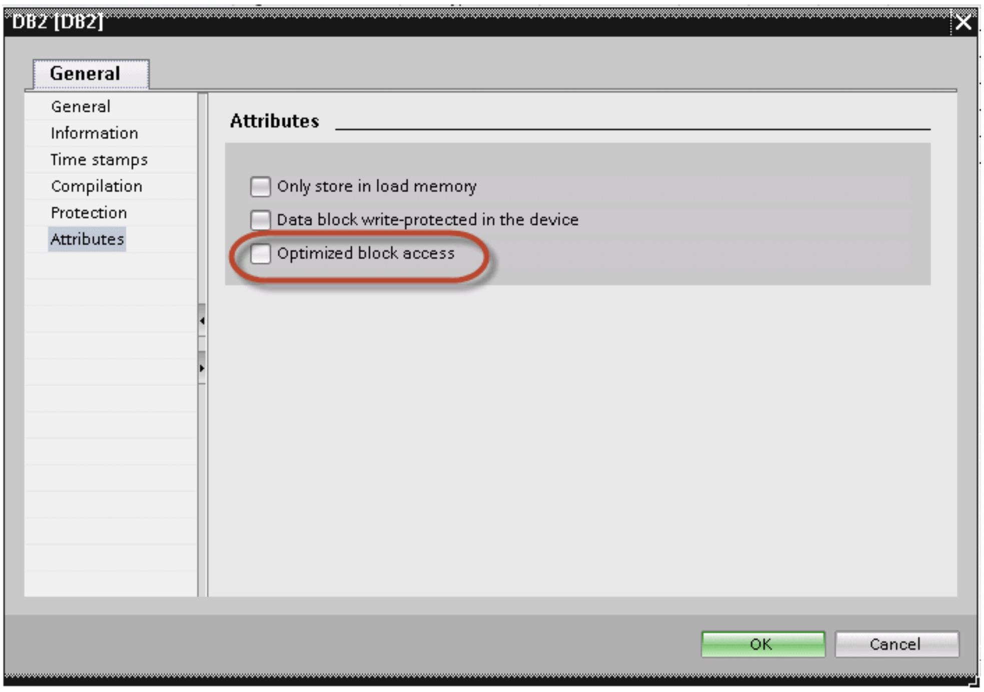
Exporting Program blocks

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**.


DB2 [DB2]

General

Attributes

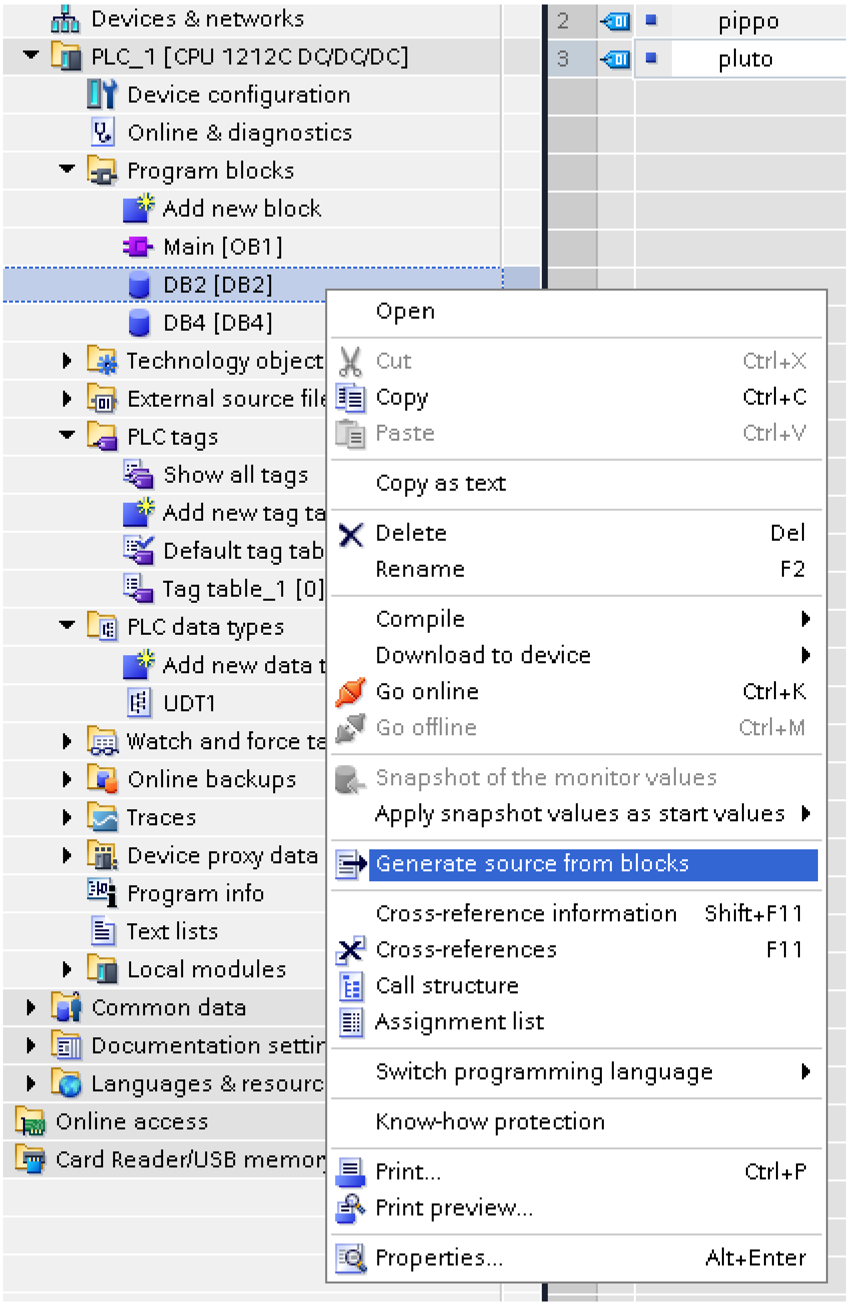
□ Only store in load memory

□ Data block write-protected in the device

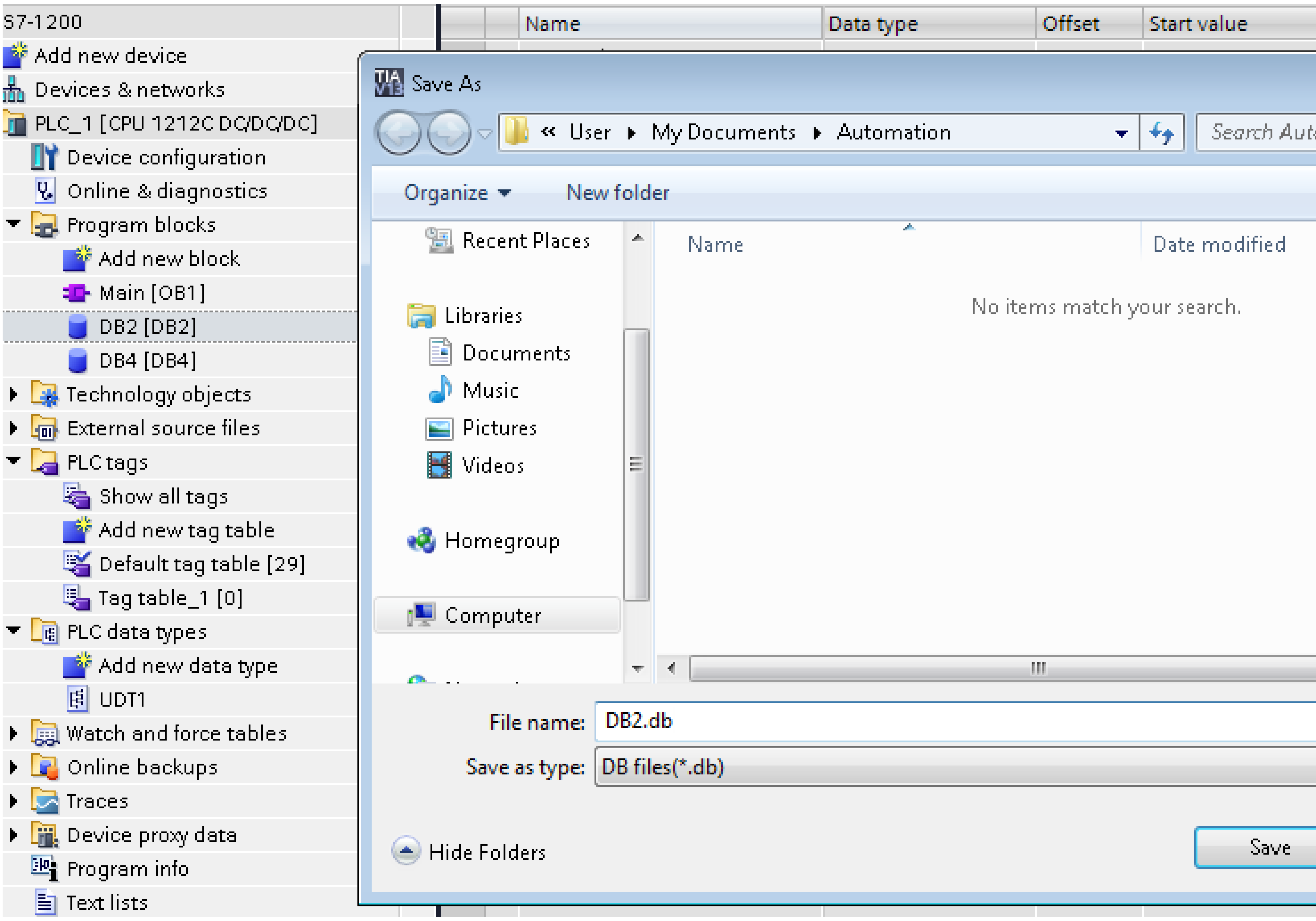
□ Optimized block access

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.



No items match your search.

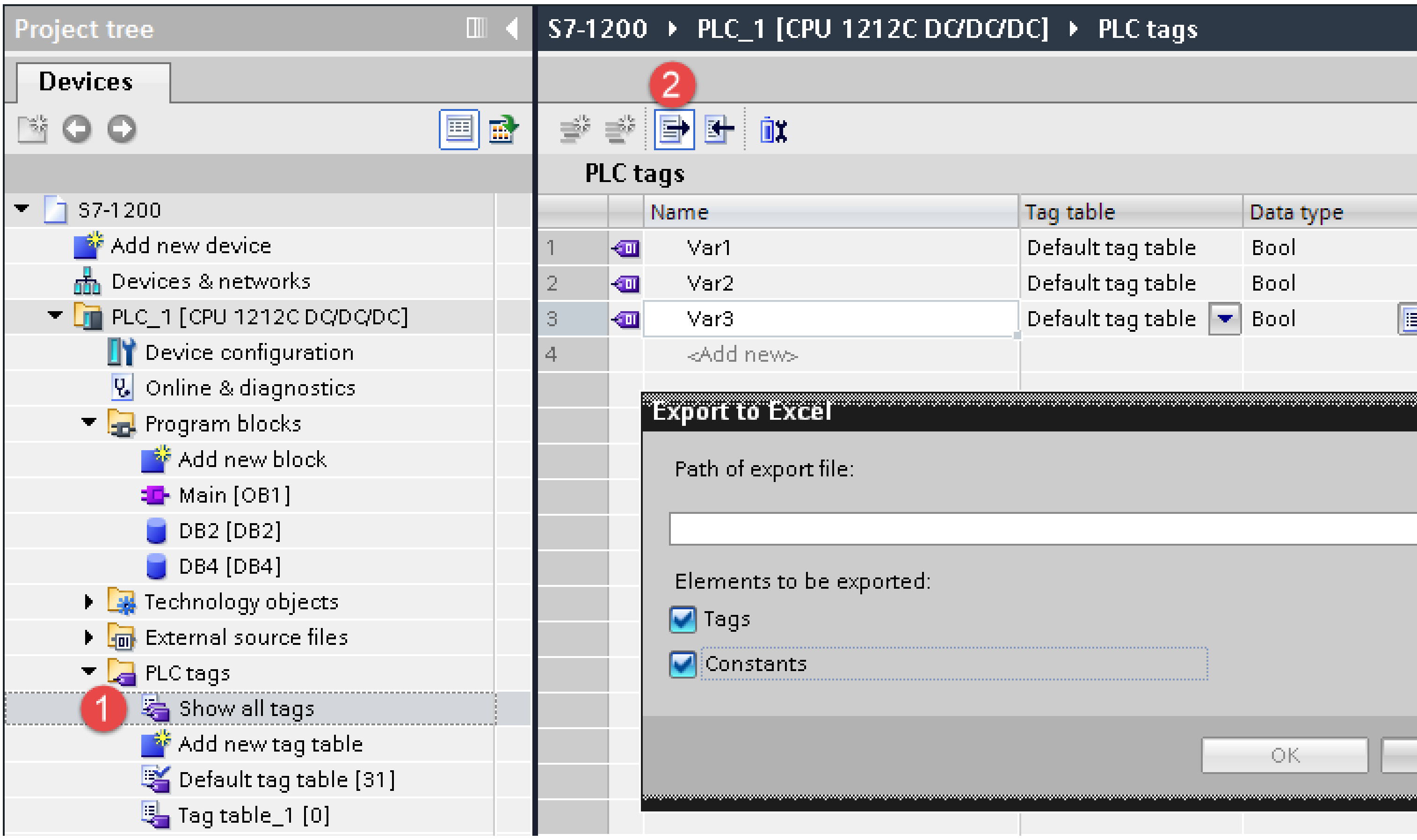
File name: DB2.db

Save as type: DB files(*.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.



		Name	Tag table	Data type
1		Var1	Default tag table	Bool
2		Var2	Default tag table	Bool
3		Var3	Default tag table	Bool
4		<Add new>		

Export to Excel

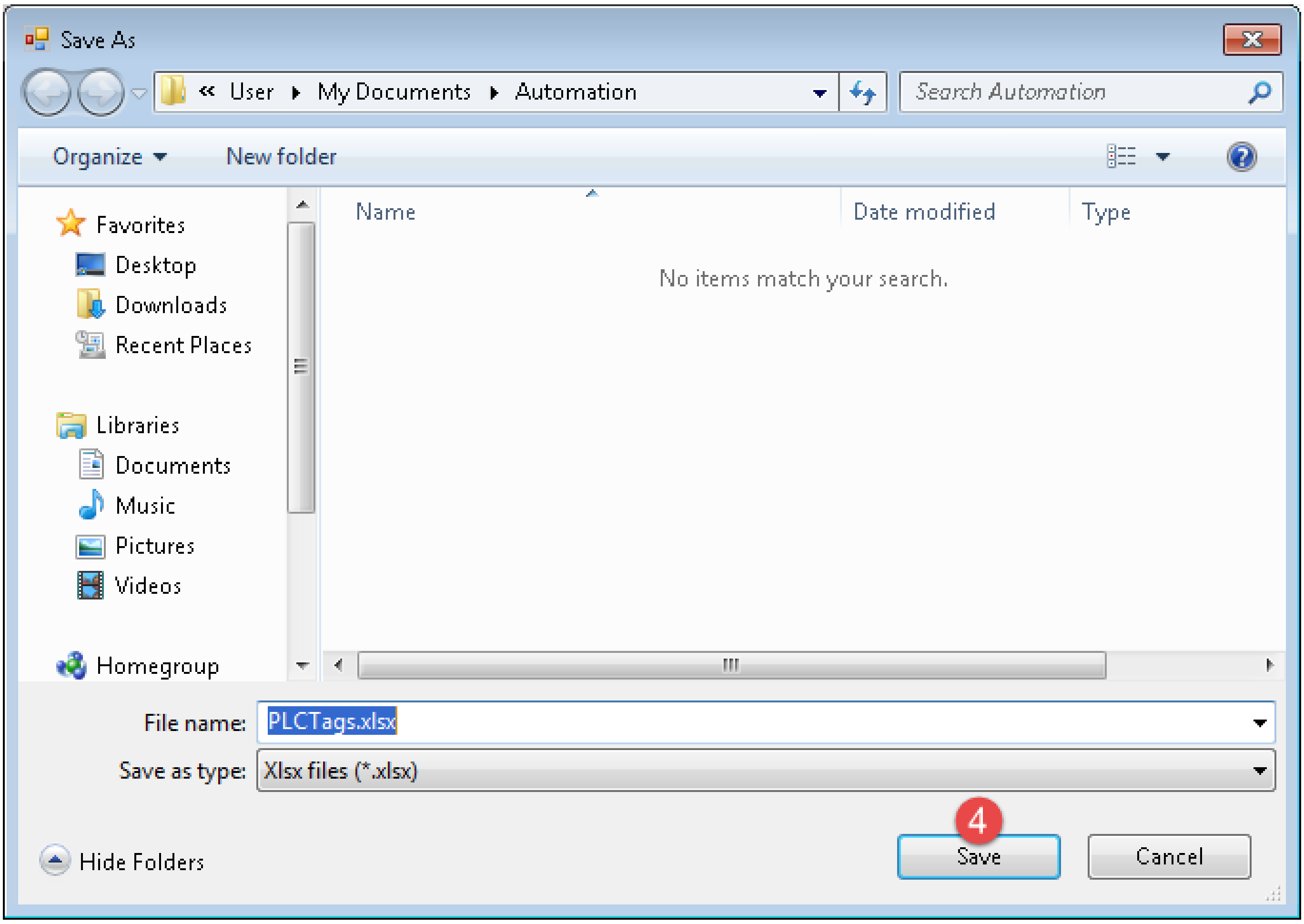
Path of export file:

Elements to be exported:

☑ Tags

☑ Constants

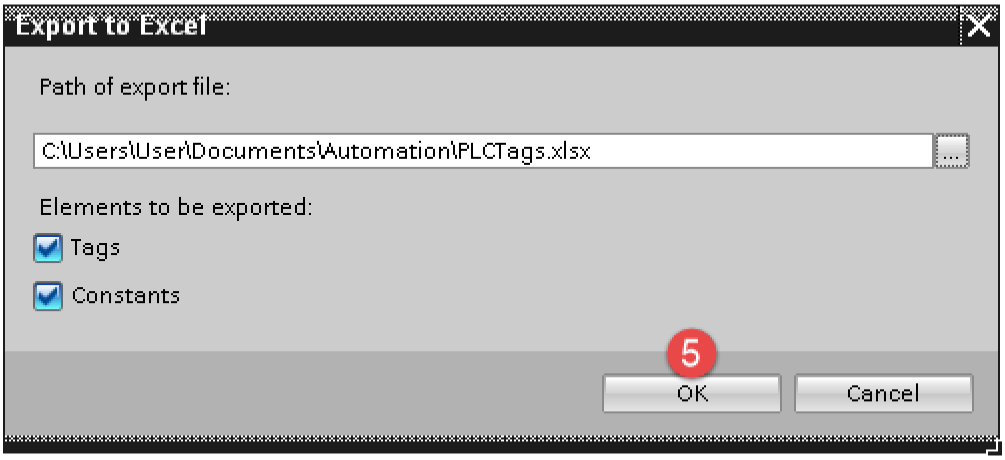
4. Click **Save** to confirm.



File name: PLCTags.xlsx

Save as type: Xlsx files (*.xlsx)

5. Click **OK** to export.



Path of export file:

C:\Users\User\Documents\Automation\PLCTags.xlsx

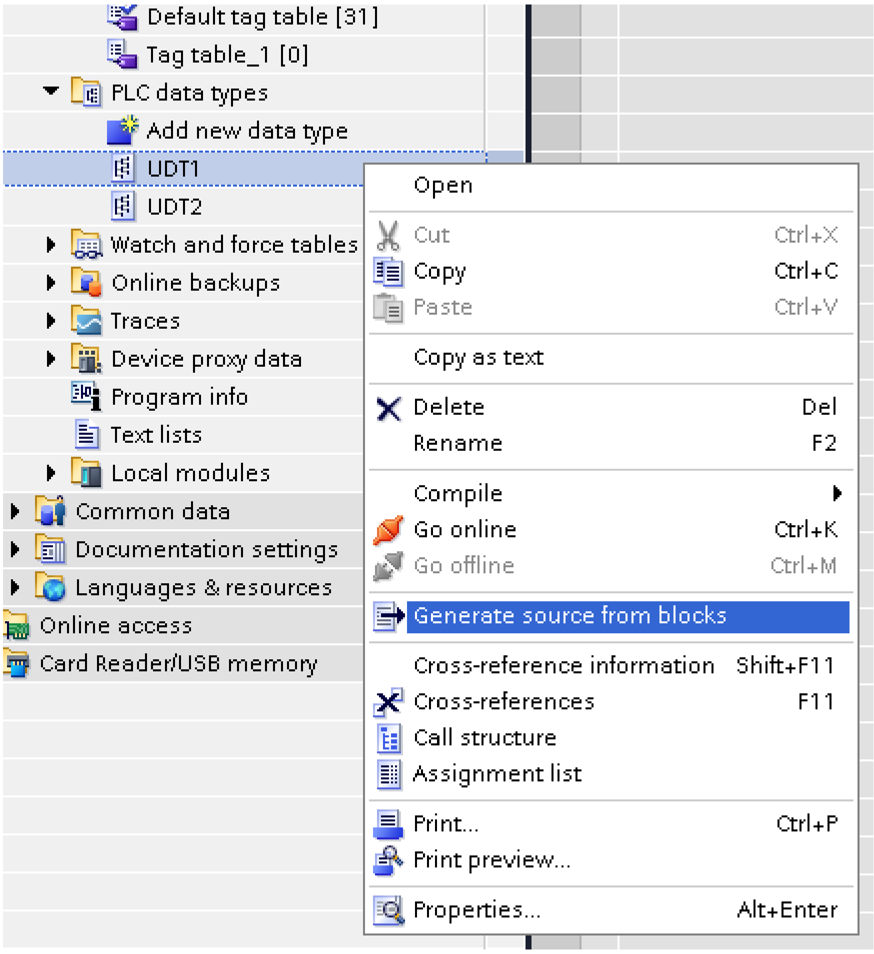
Elements to be exported:

☑ Tags

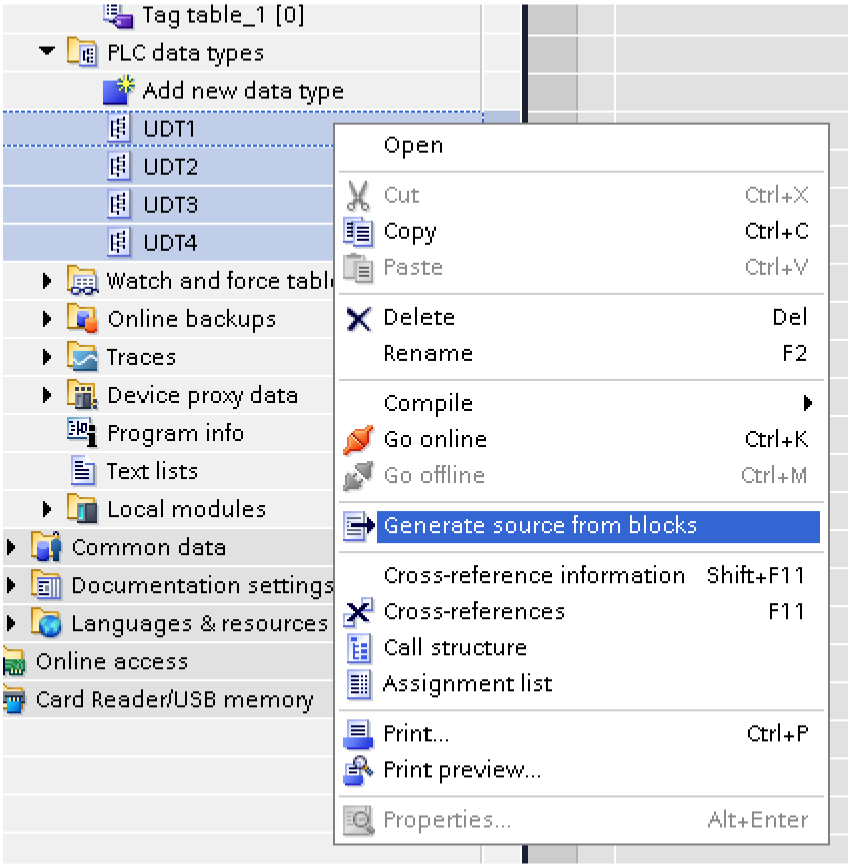
☑ Constants

Exporting PLC data types

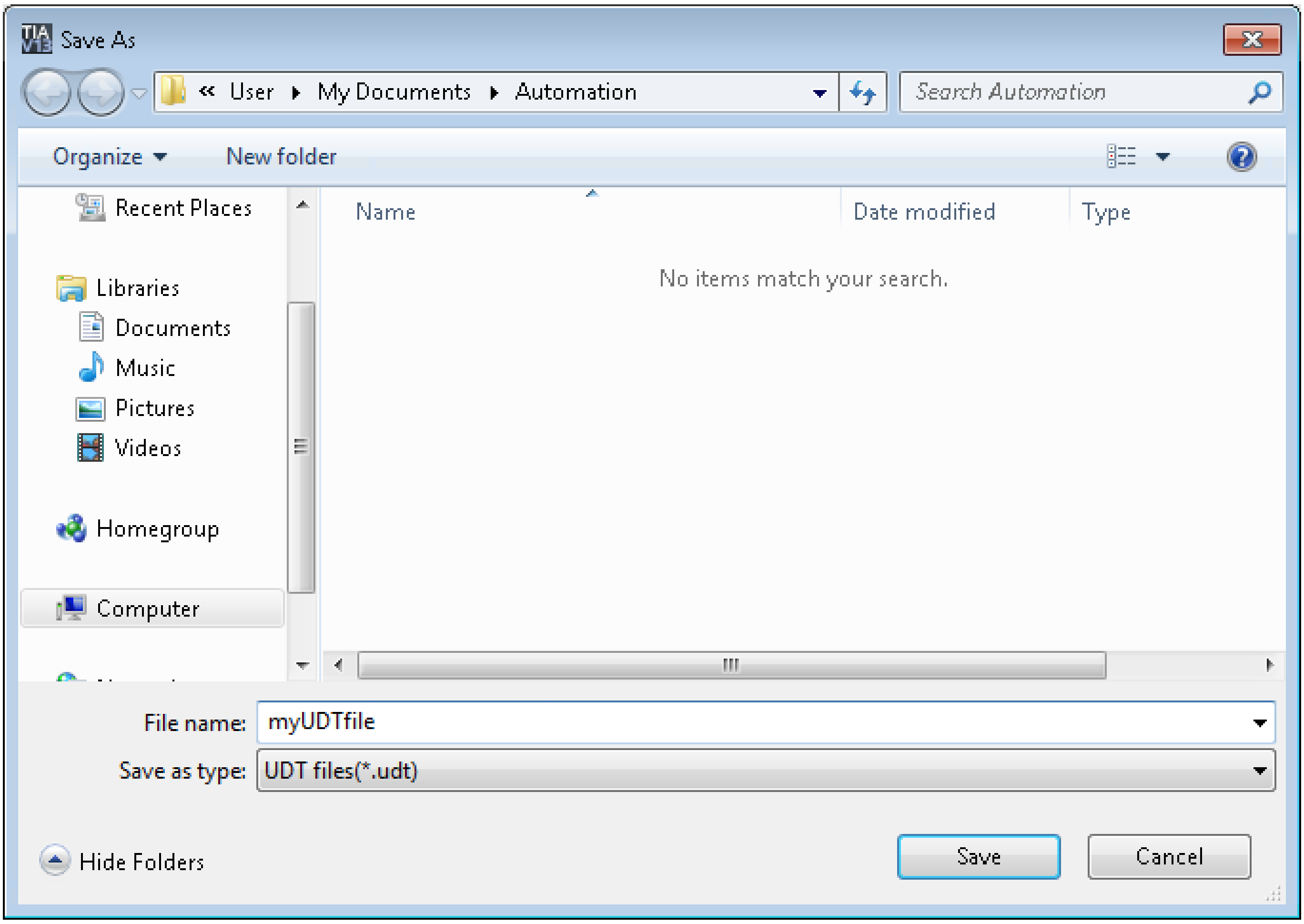
To create the file, expand **PLC data types** item from TIA Portal project tree and right click on the user defined structure. Then click on **Generate source from blocks**.



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.



File name: myUDTfile

Save as type: UDT files(*.udt)

This file will content all the PLC data types and it can be used for importing tags in Tag Editor.

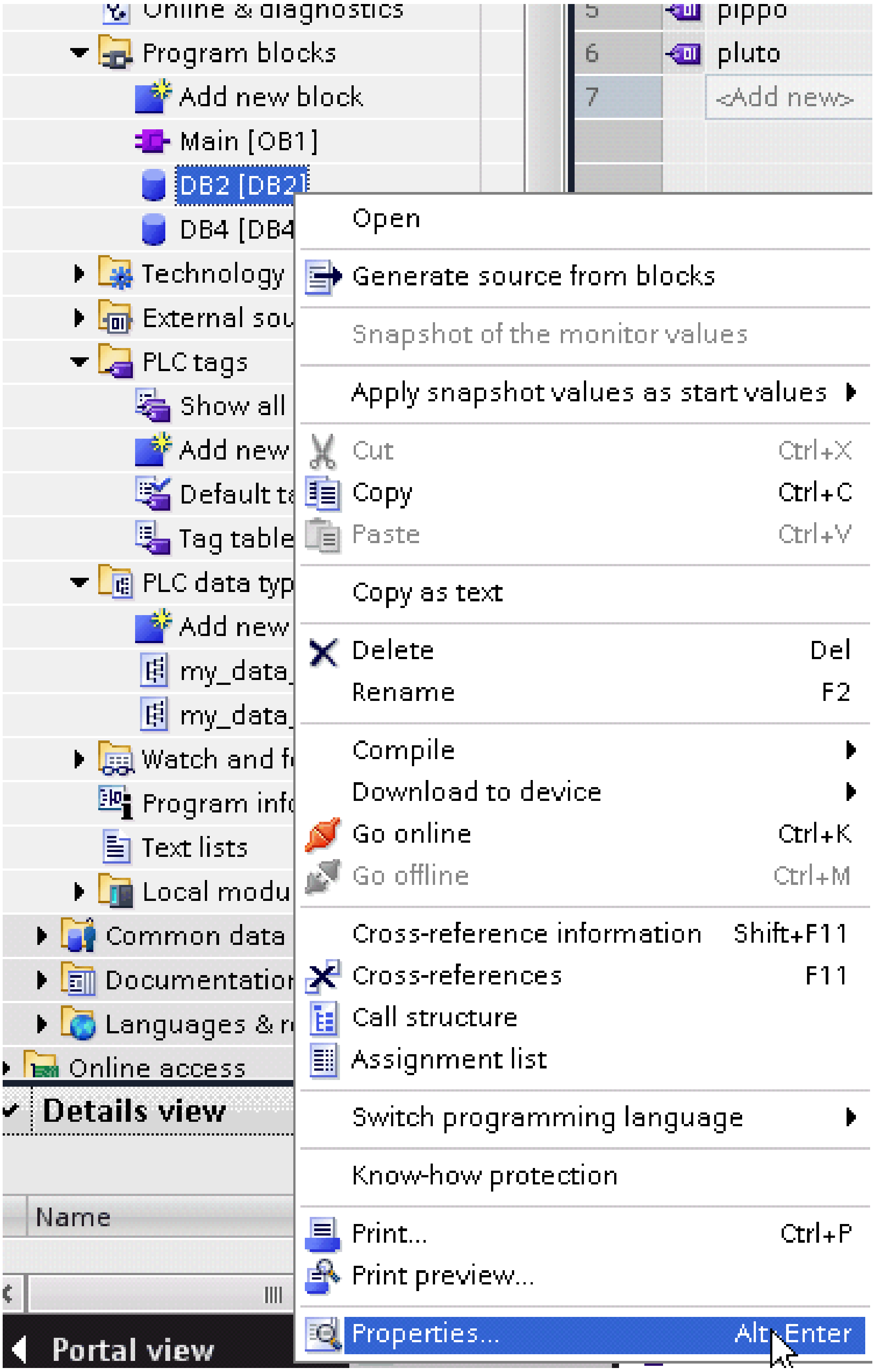
Check **Tag Import** chapter for more details.

Export using TIA Portal v10, v11, v12

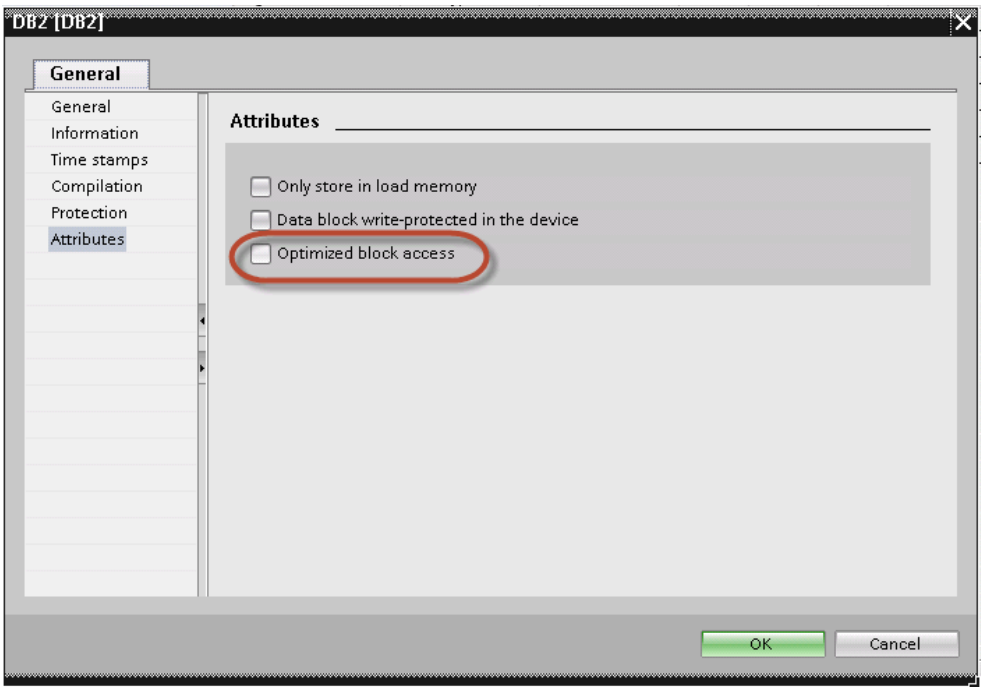
Exporting Program blocks

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**.



DB2 [DB2]

General

- General
- Information
- Time stamps
- Compilation
- Protection
- Attributes

Attributes

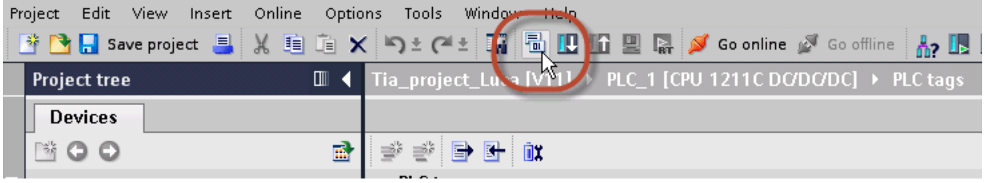
☐ Only store in load memory

☐ Data block write-protected in the device

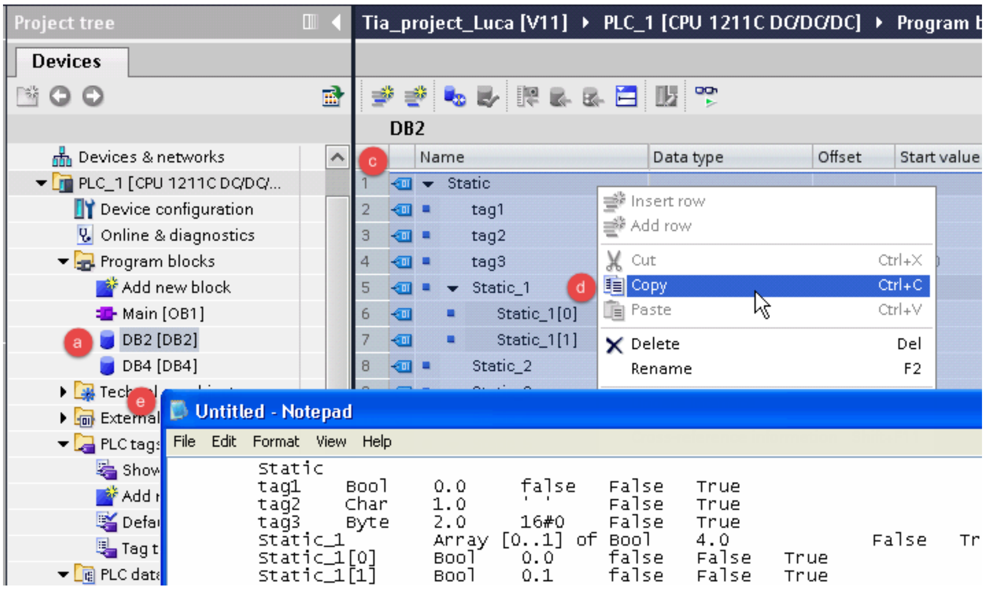
☐ Optimized block access

OK | Cancel

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.



DB2

	Name	Data type	Offset	Start value
1	Static			
2	tag1			
3	tag2			
4	tag3			
5	Static_1			
6	Static_1[0]			
7	Static_1[1]			
8	Static_2			

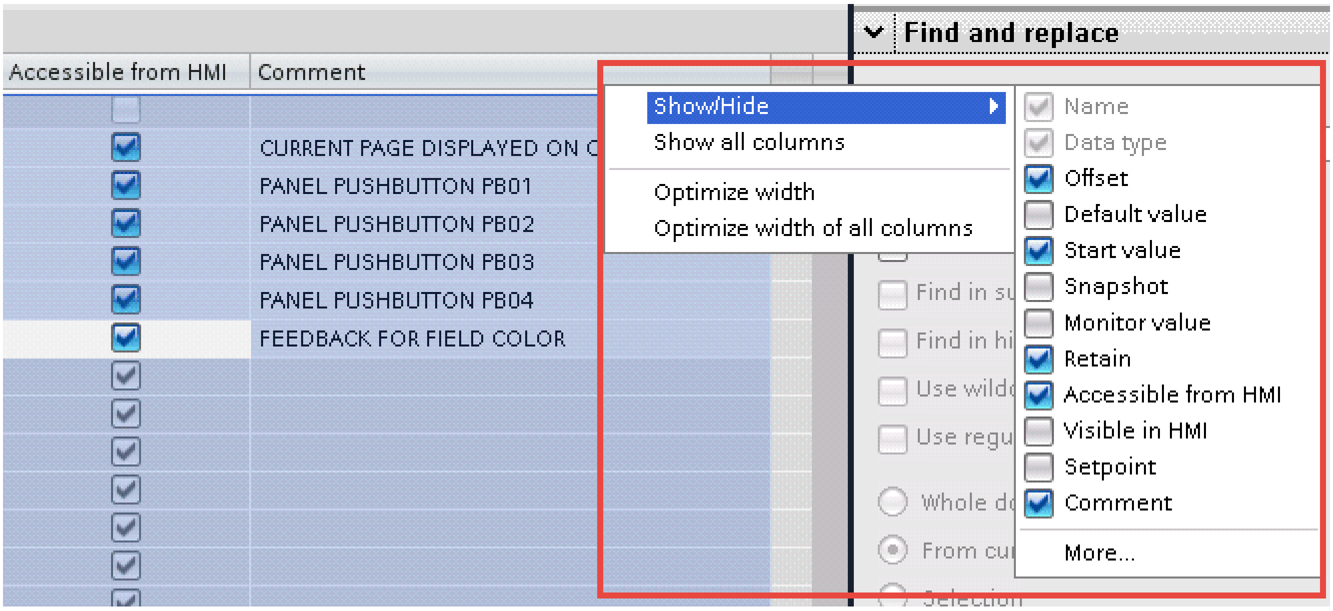
```
        Static
        tag1     Bool     0.0      false    False    True
        tag2     Char     1.0      ' '      False    True
        tag3     Byte     2.0      16#0     False    True
        Static_1          Array [0..1] of Bool     4.0                False    Tr
        Static_1[0]       Bool     0.0      false    False    True
        Static_1[1]       Bool     0.1      false    False    True
```

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



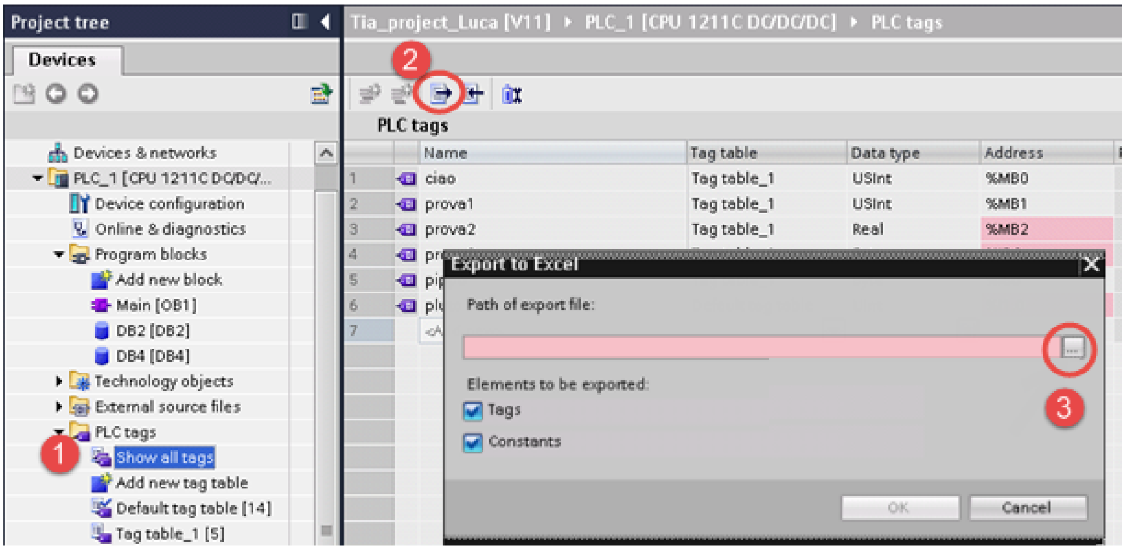
Accessible from HMI	Comment
□	
☑	CURRENT PAGE DISPLAYED ON C
☑	PANEL PUSHBUTTON PB01
☑	PANEL PUSHBUTTON PB02
☑	PANEL PUSHBUTTON PB03
☑	PANEL PUSHBUTTON PB04
☑	FEEDBACK FOR FIELD COLOR
☑	
☑	
☑	
☑	
☑	
☑	

☑ Name
☑ Data type
☑ Offset
□ Default value
☑ Start value
□ Snapshot
□ Monitor value
☑ Retain
☑ Accessible from HMI
□ Visible in HMI
□ Setpoint
☑ Comment

Exporting PLC tags

An Excel file refers to PLC tags.

1. Double-click **Show all tags**: the tag table is displayed.



	Name	Tag table	Data type	Address
1	ciao	Tag table_1	USInt	%MB0
2	prova1	Tag table_1	USInt	%MB1
3	prova2	Tag table_1	Real	%MB2
4	pr			
5	pi			
6	pl			
7	<A			

Export to Excel

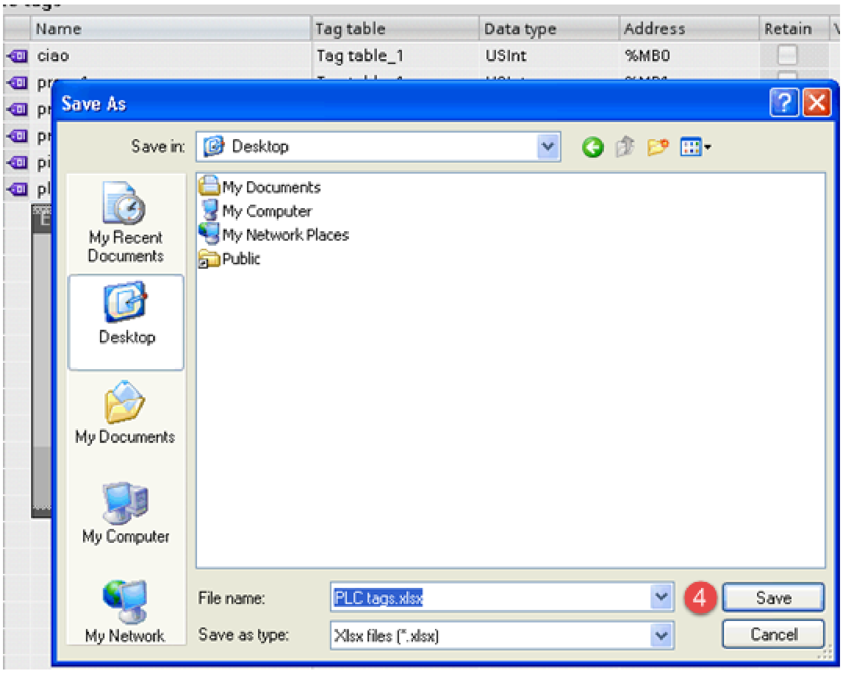
Path of export file:

Elements to be exported:

☑ Tags

☑ Constants

2. Click the **Export** button and browse for path file.
3. Define file name.
4. Click **Save** to confirm.



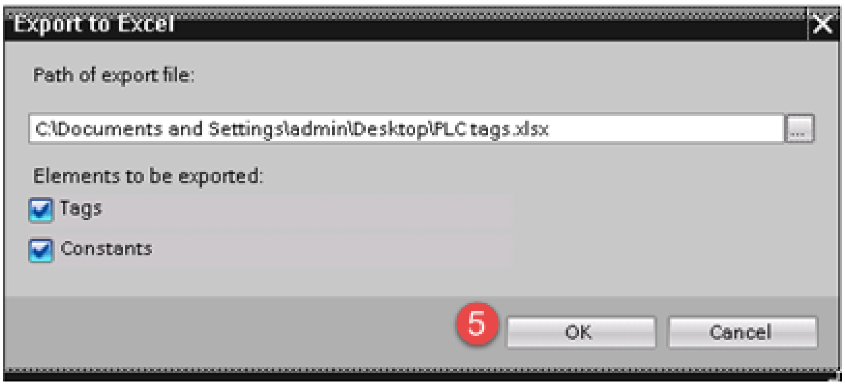
Name	Tag table	Data type	Address	Retain
ciao	Tag table_1	USInt	%MB0	□

Save in: Desktop

File name: PLC tags.xlsx

Save as type: Xlsx files (*.xlsx)

5. Click **OK** to export.



Export to Excel

Path of export file:

C:\Documents and Settings\admin\Desktop\PLC tags.xlsx

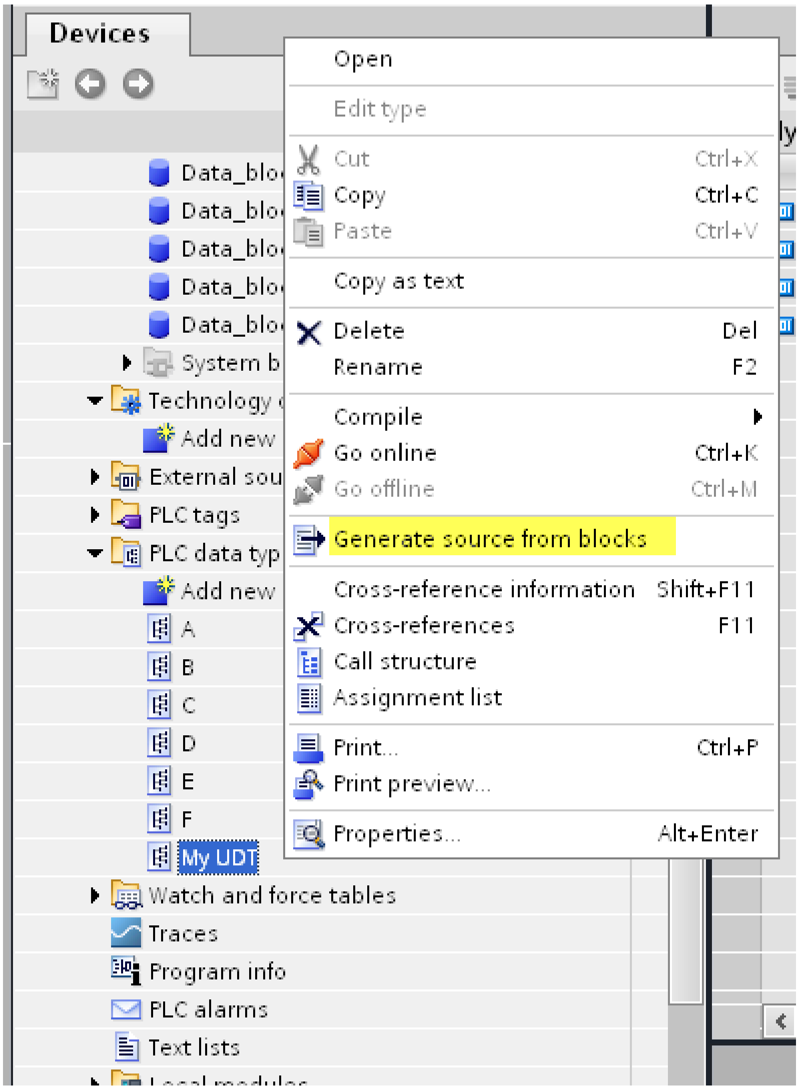
Elements to be exported:

☑ Tags

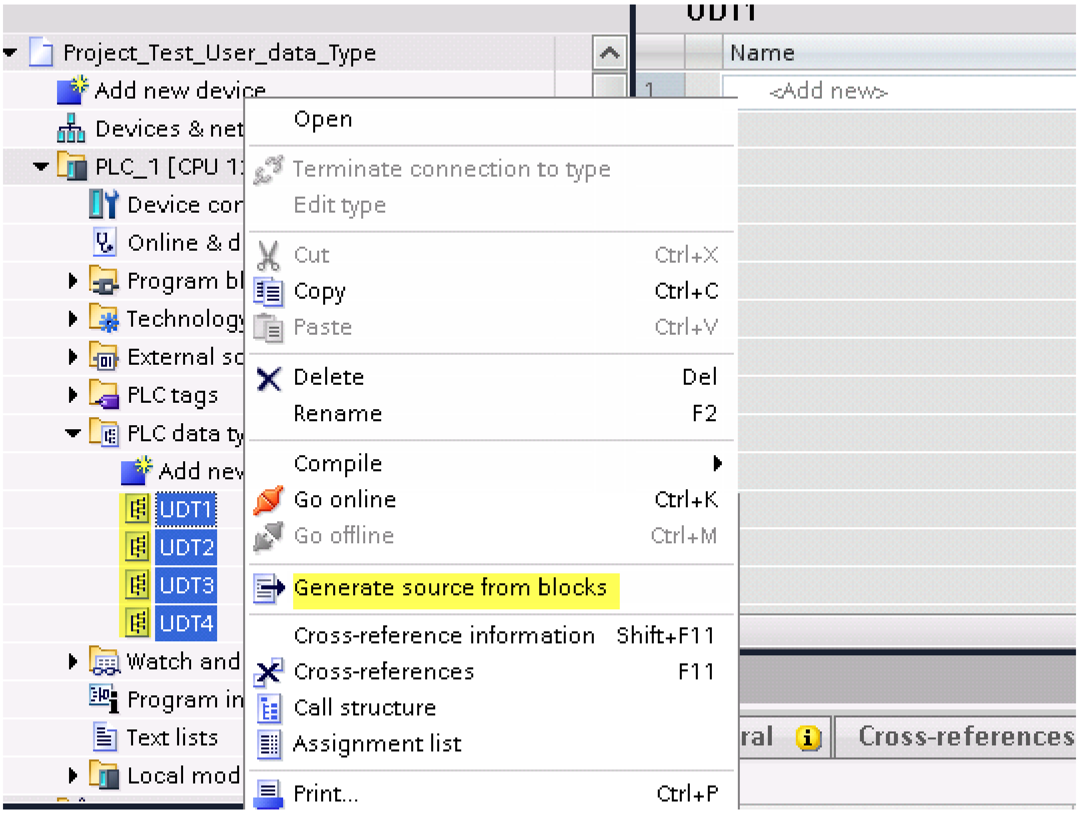
☑ Constants

Exporting PLC data types

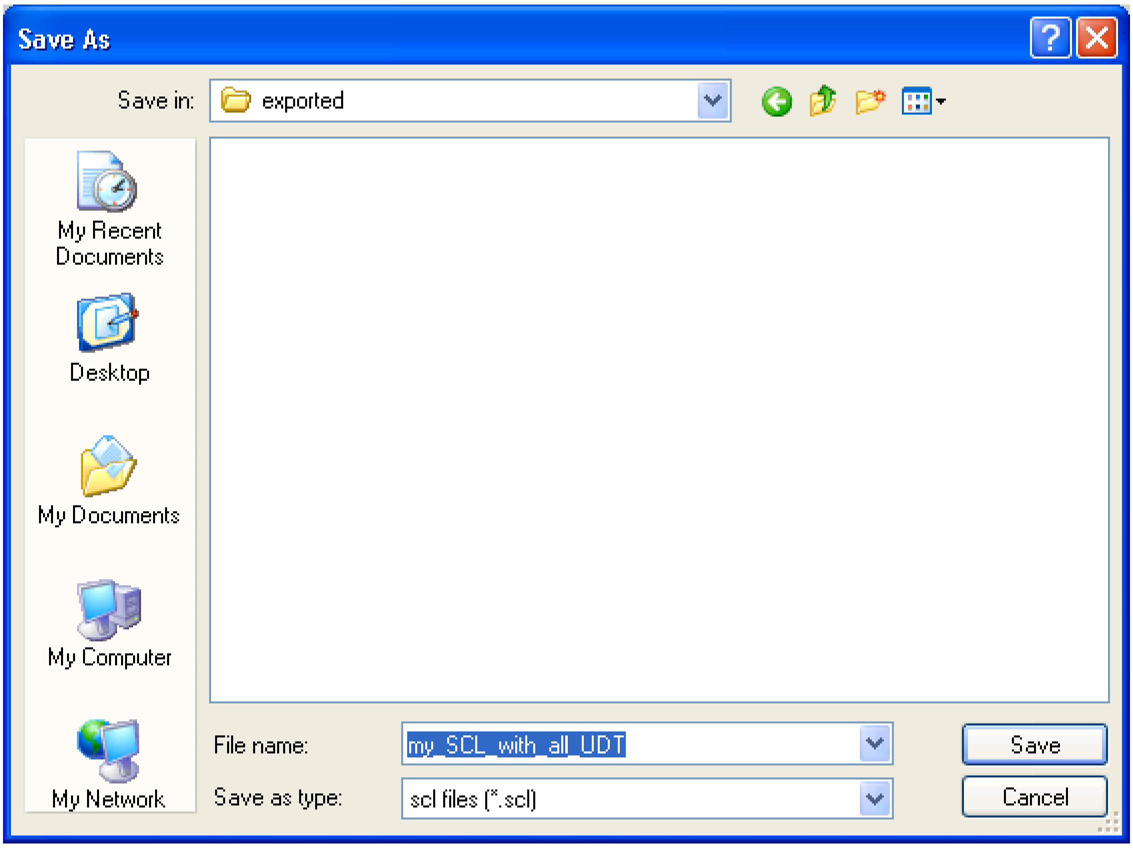
To create the file, expand **PLC data types** item from TIA Portal project tree and right click on the user defined structure. Then click on **Generate source from blocks**.



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.



File name: my_SCL_with_all_UDT

Save as type: scl files (*.scl)

This file will content all the PLC data types and it can be used for importing tags in Tag Editor.

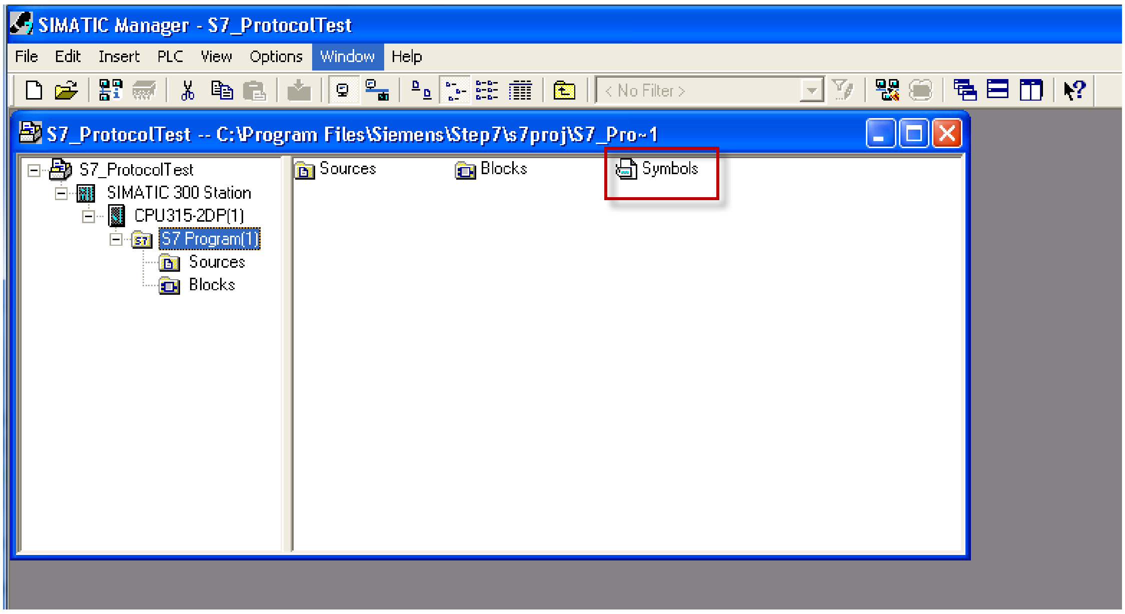
Check **Tag Import** chapter for more details.

Export using STEP7

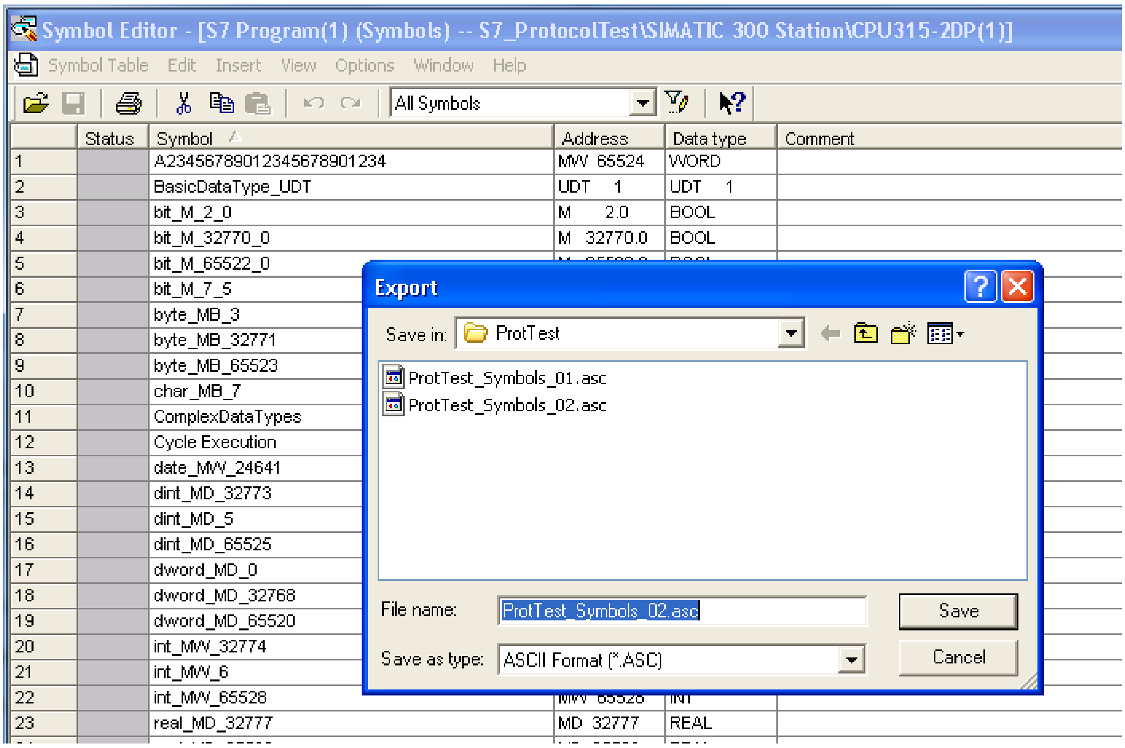
The Simatic S7 ETH Tag importer accepts symbol files (ASCII format .asc) and source files (.awl extension) created by the Simatic Step7. The symbol file can be previously exported using the Step7 symbol table utility.

Exporting Symbols table

Symbol files (.asc) can be exported from the symbol table utility.


SIMATIC Manager - S7_ProtocolTest
File Edit Insert PLC View Options Window Help
< No Filter >
S7_ProtocolTest -- C:\Program Files\Siemens\Step7\s7proj\S7_Pro~1
S7_ProtocolTest
SIMATIC 300 Station
CPU315-2DP(1)
S7 Program(1)
Sources
Blocks
Sources
Blocks
Symbols

1. From the **Symbol Table** menu in the Symbol Editor choose **Export**.
2. Assign a name and save the symbol table as ASCII file.


Symbol Editor - [S7 Program(1) (Symbols) -- S7_ProtocolTest\SIMATIC 300 Station\CPU315-2DP(1)]
Symbol Table Edit Insert View Options Window Help
All Symbols
Export
Save in: ProtTest
ProtTest_Symbols_01.asc
ProtTest_Symbols_02.asc
Save
Cancel

File name: ProtTest_Symbols_02.asc

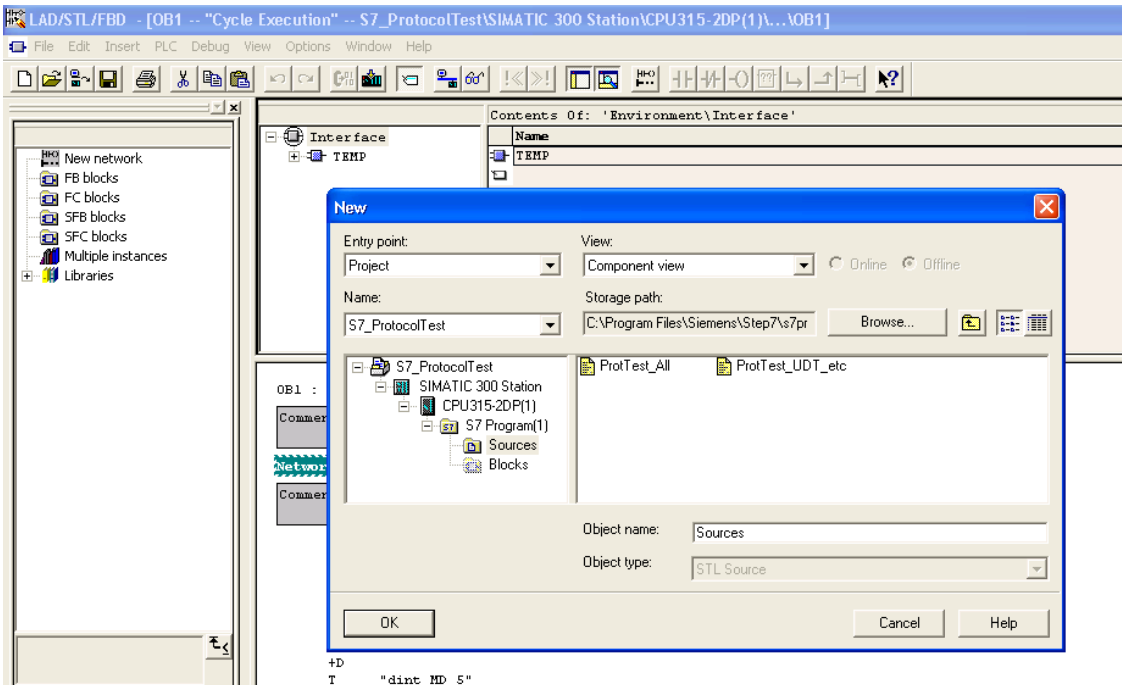
Save as type: ASCII Format (*.ASC)

	Status	Symbol	Address	Data type	Comment
1		A23456789012345678901234	MW 65524	WORD	
2		BasicDataType_UDT	UDT 1	UDT 1	
3		bit_M_2_0	M 2.0	BOOL	
4		bit_M_32770_0	M 32770.0	BOOL	
5		bit_M_65522_0			
6		bit_M_7_5			
7		byte_MB_3			
8		byte_MB_32771			
9		byte_MB_65523			
10		char_MB_7			
11		ComplexDataTypes			
12		Cycle Execution			
13		date_MW_24641			
14		dint_MD_32773			
15		dint_MD_5			
16		dint_MD_65525			
17		dword_MD_0			
18		dword_MD_32768			
19		dword_MD_65520			
20		int_MW_32774			
21		int_MW_6			
22		int_MW_65528	MW 65528	INT	
23		real_MD_32777	MD 32777	REAL	

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:



Entry point: Project

View: Component view ○ Online ● Offline

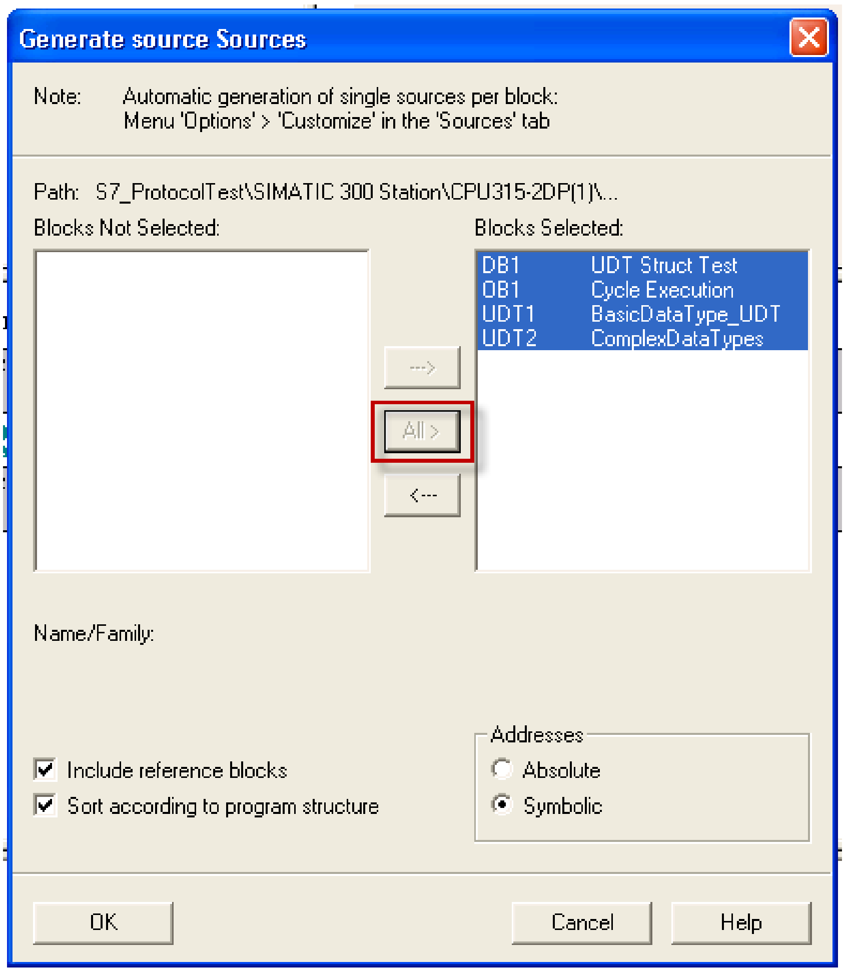
Name: S7_ProtocolTest

Storage path: C:\Program Files\Siemens\Step7\s7pr

Object name: Sources

Object type: STL Source

1. Assign a name, "Sources" in the example, and click **OK**: the **Generate source Sources** dialog is displayed.



Note: Automatic generation of single sources per block:
Menu 'Options' > 'Customize' in the 'Sources' tab

Path: S7_ProtocolTest\SIMATIC 300 Station\CPU315-2DP(1)\...

Blocks Not Selected:

Blocks Selected:

DB1	UDT Struct Test
OB1	Cycle Execution
UDT1	BasicDataType_UDT
UDT2	ComplexDataTypes

Name/Family:

☑ Include reference blocks

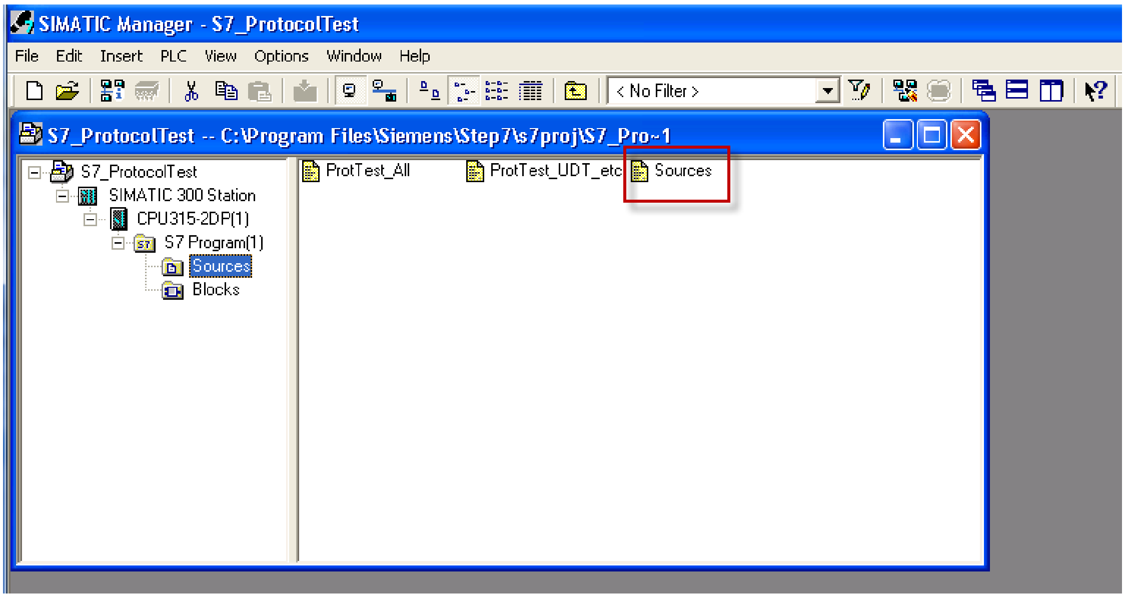
☑ Sort according to program structure

Addresses
○ Absolute
● Symbolic

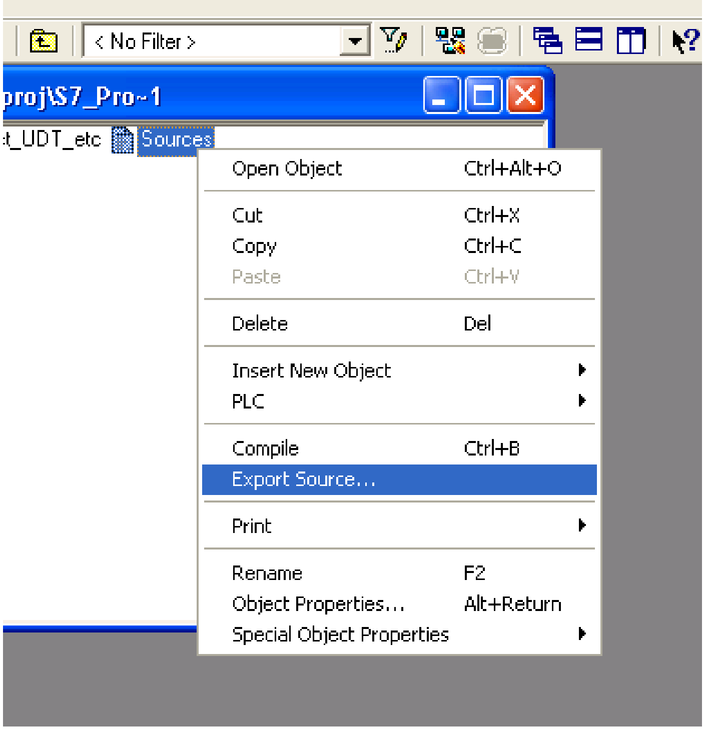
2. Click **All >** to generate source for all blocks.
3. Select the following options:

- **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.


SIMATIC Manager - S7_ProtocolTest
File Edit Insert PLC View Options Window Help
< No Filter >
S7_ProtocolTest -- C:\Program Files\Siemens\Step7\s7proj\S7_Pro~1
S7_ProtocolTest
SIMATIC 300 Station
CPU315-2DP(1)
S7 Program(1)
Sources
Blocks
ProtTest_All
ProtTest_UDT_etc
Sources

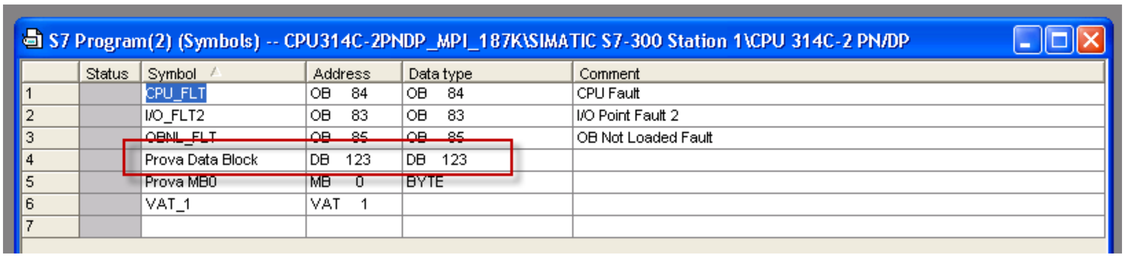
5. Right click on the object and select **Export Sources**.



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.



S7 Program(2) (Symbols) -- CPU314C-2PNDP_MPI_187K\SIMATIC S7-300 Station 1\CPU 314C-2 PN/DP

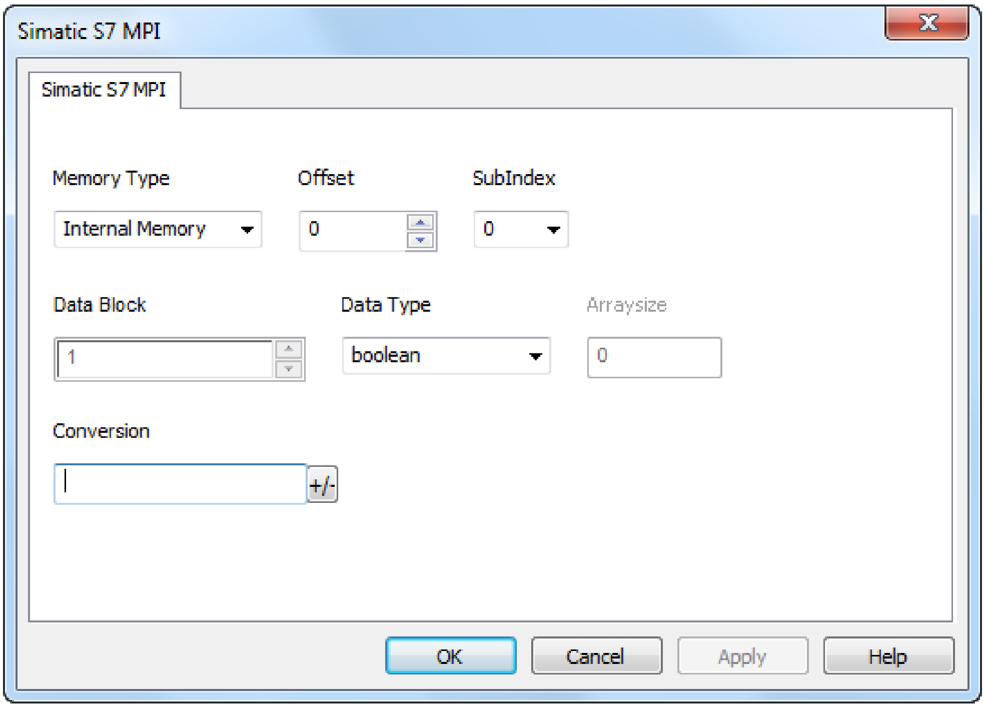
	Status	Symbol	Address	Data type	Comment
1		CPU_FLT	OB 84	OB 84	CPU Fault
2		I/O_FLT2	OB 83	OB 83	I/O Point Fault 2
3		OBNL_FLT	OB 85	OB 85	OB Not Loaded Fault
4		Prova Data Block	DB 123	DB 123	
5		Prova MB0	MB 0	BYTE	
6		VAT_1	VAT 1		
7					

Each entry enables the import filter to import the tags related to the specified data block.

Tag Editor Settings

Into Tag editor select the protocol “Simatic S7 MPI” from the list of defined protocols and add a tag using [+] button.

Tag settings can be defined using the following dialog:



Simatic S7 MPI

Memory Type: Internal Memory

Offset: 0

SubIndex: 0

Data Block: 1

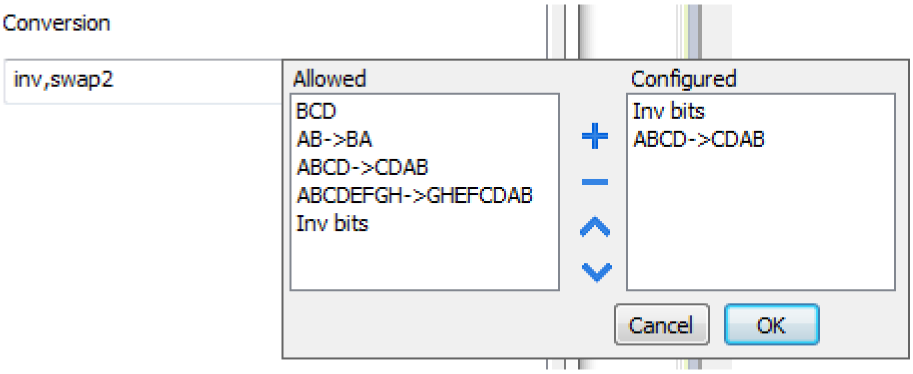
Data Type: boolean

Arraysize: 0

Conversion:

<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td><b>Memory Type</b></td><td>Area of PLC where tag is located.
<table>
<tr><th>Data Type</th><th>Simatic Type</th></tr>
<tr><td><b>Internal Memory</b></td><td>M</td></tr>
<tr><td><b>Data Block</b></td><td>DB</td></tr>
<tr><td><b>Input</b></td><td>I (E)</td></tr>
<tr><td><b>Output</b></td><td>O (A)</td></tr>
<tr><td><b>Timer value</b></td><td>T</td></tr>
<tr><td><b>Counter value</b></td><td>C</td></tr>
</table></td></tr>
<tr><td><b>Offset</b></td><td>Offset address where tag is located.</td></tr>
<tr><td><b>SubIndex</b></td><td>In case of Boolean data type, this is the offset of single bit.</td></tr>
<tr><td><b>Data Block</b></td><td>If Memory Type is “Data Block”, this will identify the DB number.</td></tr>
<tr><td><b>Data Type</b></td><td>
<table>
<tr><th>Data Type</th><th>Memory Space</th><th>Limits</th></tr>
<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>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>float</b></td><td>IEEE single-precision
32-bit floating point type</td><td>1.17e-38 ... 3.40e38</td></tr>
<tr><td><b>string</b></td><td colspan="2">Refer to “String data type channel”</td></tr>
</table>
ℹ Note: to define arrays, select one of Data Type format followed by square brackets like “byte[]”, “short[]”…</td></tr>
<tr><td><b>Arraysize</b></td><td>• 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.</td></tr>
</table>

<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td></td><td>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.</td></tr>
<tr><td><b>Conversion</b></td><td>Conversion to be applied to the tag.



Conversion: inv,swap2

Depending on data type selected, the <b>Allowed</b> list shows one or more conversions, listed below.

<table>
<tr><th>Value</th><th>Description</th></tr>
<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 -> 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 -> CDAB</b></td><td>Swap bytes of a word.
<i>Example:</i>
9ACC → CC9A (in hexadecimal format)
39628 → 52378 (in decimal format)</td></tr>
<tr><td><b>ABCDEFGH -> GHEFCDAB</b></td><td>Swap bytes of a double word.
<i>Example:</i>
32FCFF54 → 54FFFC32 (in hexadecimal format)
855441236 → 1426062386 (in decimal format)</td></tr>
</table>
</td></tr>
</table>

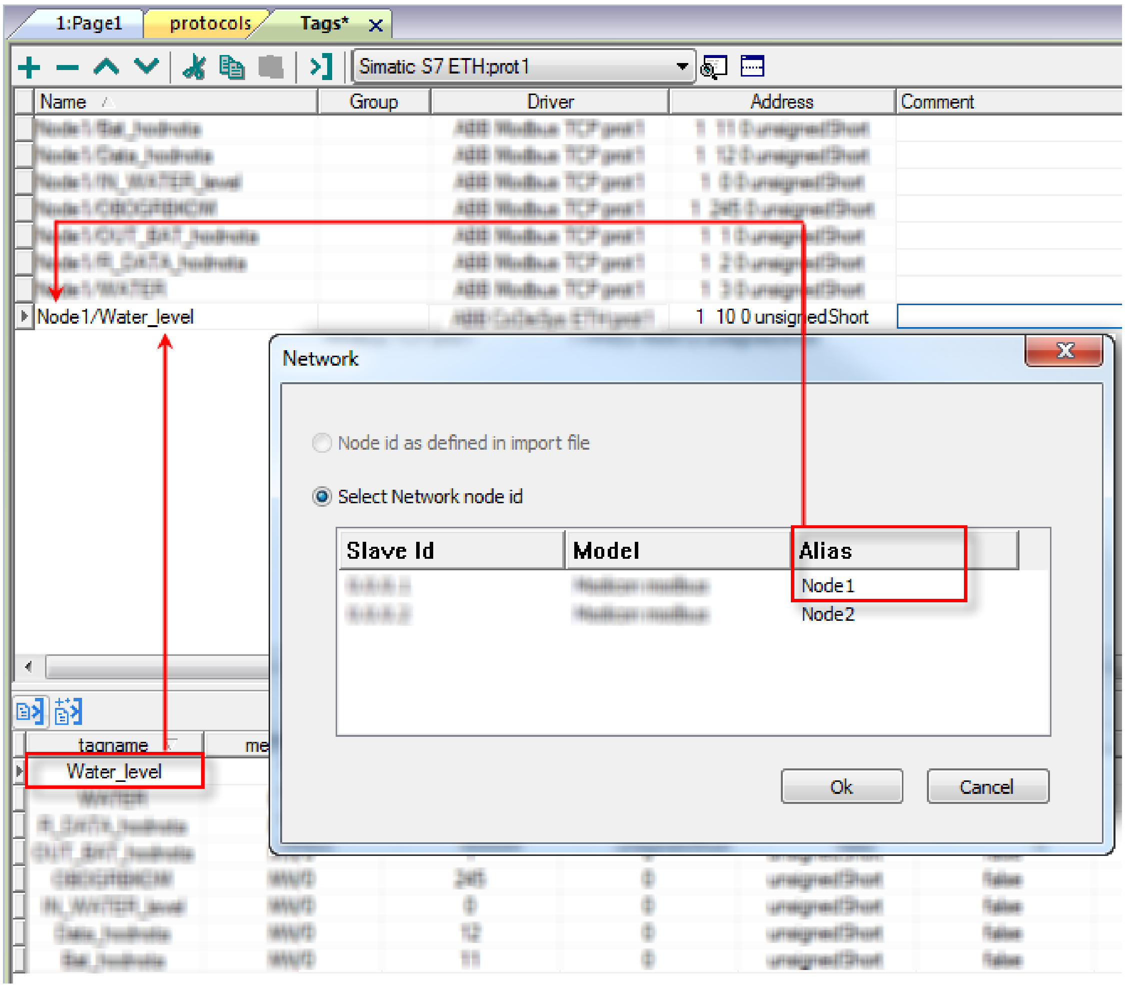
<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td><b>ABC...NOP -> OPM...DAB</b></td><td>Swap bytes of a long word.
Example:
142.366 → -893553517.588905 (in decimal format)
0 10000000110
0001110010111011011001000101101000011100101011000001
→
1 10000011100
1010101000010100010110110110110010110110000100111101
(in binary format)</td></tr>
<tr><td><b>BCD</b></td><td>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)</td></tr>
<tr><td><b>S5timer(BCD)</b></td><td>Used to support S5timer. Check <b>Simatic S5timer special data type</b> for more details.</td></tr>
<tr><td><b>S5timer(BIN)</b></td><td>Legacy transformation for S5timer in binary format.</td></tr>
</table>

Select the conversion and click on plus button. The selected item will be added on <b>Configured</b> list.
If more conversions are configured, they will be applied in order (from top to bottom of <b>Configured</b> list).
Use the arrow buttons to order the configured conversions.</td></tr>
</table>

Aliasing Tag Names in Network Configurations

Tag names must be unique at project level; it often happens that the same tag names have to be used for different controller nodes (for example when the HMI is connected to two devices that are running the same application). Since tags include also the identification of the node and Tag Editor does not support duplicate tag names, the import facility in Tag Editor has an aliasing feature that can automatically add a prefix to imported tags. With this feature tag names can be done unique at project level.

The feature works when importing tags for a specific protocol. Each tag name will be prefixed with the string specified by the “Alias”. As shown in the figure below, 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.



Simatic S7 ETH:prot1

Name	Group	Driver	Address	Comment
Node1/Water_level			1 10 0 unsignedShort	

Network

○ Node id as defined in import file

● Select Network node id

Slave Id	Model	Alias
		Node1
		Node2

Note: Aliasing tag names are only available when tags can be imported. Tags which are added manually in the Tag Editor do not need to have the Alias prefix in the tag name.

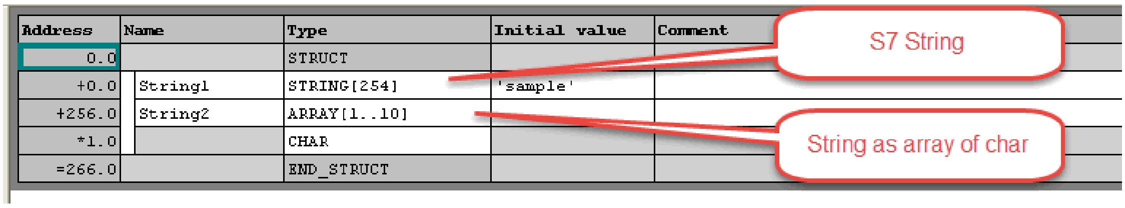
The Alias string is attached to the tag name only at the moment the tags are imported using Tag Editor. If Alias string is modified after the tag import has been completed, there will be no effect on the names already present in the dictionary. When the Alias string is changed and tags are imported again, all tags will be imported again with the new prefix string.

String data type

In Simatic S7 PLC it's possible to define two different types of tags to manage string variables.

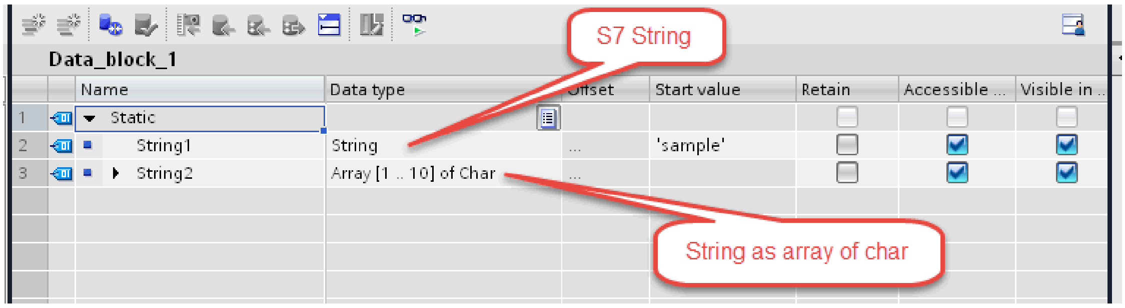
- as Array [1..xx] of Chars.
- as String[xx].

Step7 string declaration is showed in the following figure:



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		

TIA Portal string declaration is showed in the following figure:

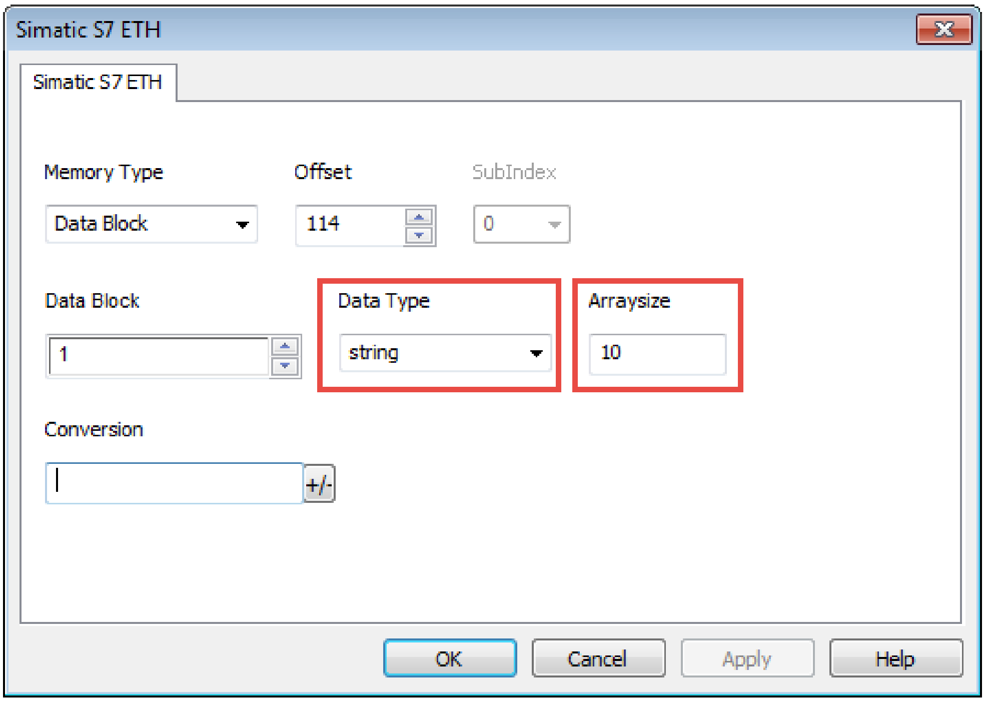


Data_block_1

	Name	Data type	Offset	Start value	Retain	Accessible ...	Visible in ...
1	▾ Static				□	□	□
2	String1	String	...	'sample'	□	☑	☑
3	▸ String2	Array [1 .. 10] of Char	...		□	☑	☑

Note: Usage of String[xx] data type is allowed but a specific Conversion must be applied to the tag. Anyway using tag importer to import tag dictionary from TIA Portal or Step7 string tags are automatically configured and no changes/conversion are needed.

To manually add an "Array [1..xx] of Chars" data type tag, press the [+] button in the Tag Editor, then select "string" as Data Type of the Tag and type the string length in the "Arraysize" field:



Simatic S7 ETH

Memory Type: Data Block

Offset: 114

SubIndex: 0

Data Block: 1

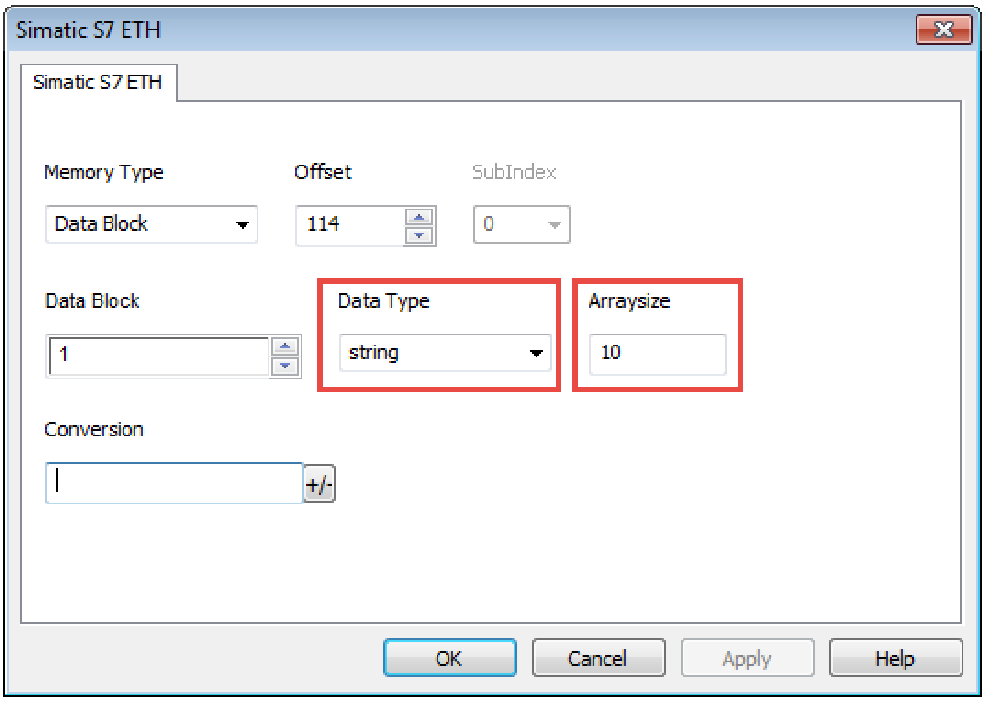
Data Type: string

Arraysize: 10

Conversion:

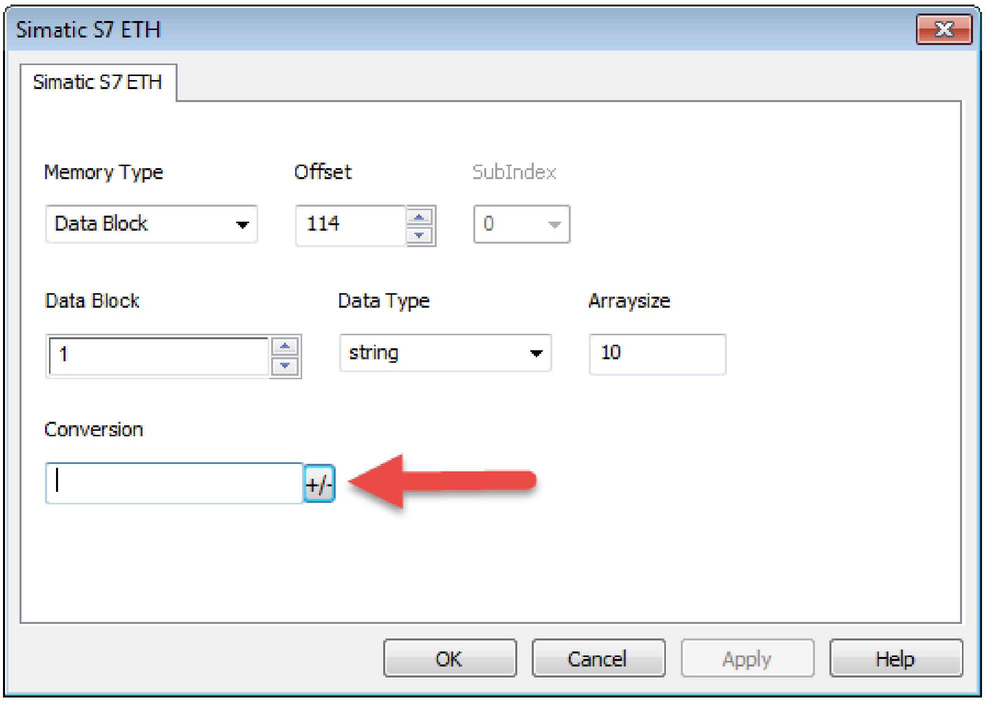
and confirm with OK button.

To manually add a "String[xx]" data type tag, press the [+] button in the Tag Editor, then select "string" as Data Type of the Tag and type the string length in the "Arraysize" field,


Simatic S7 ETH

Memory Type: Data Block
Offset: 114
SubIndex: 0
Data Block: 1
Data Type: string
Arraysize: 10
Conversion:

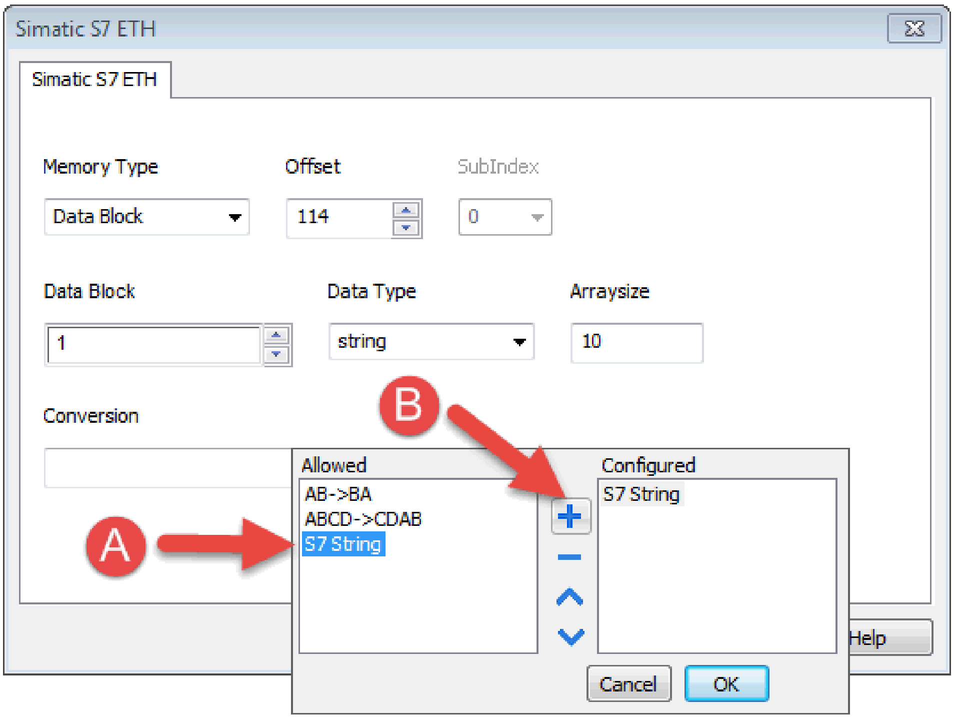
then click on [+/-] button to open the Conversion dialog.


Simatic S7 ETH

Memory Type: Data Block
Offset: 114
SubIndex: 0
Data Block: 1
Data Type: string
Arraysize: 10
Conversion:

Into conversion dialog:

- select the "S7 String" conversion type
- click on [+] button to add the conversion.



Simatic S7 ETH

Memory Type: Data Block
Offset: 114
SubIndex: 0
Data Block: 1
Data Type: string
Arraysize: 10
Conversion

Allowed: AB->BA, ABCD->CDAB, S7 String
Configured: S7 String

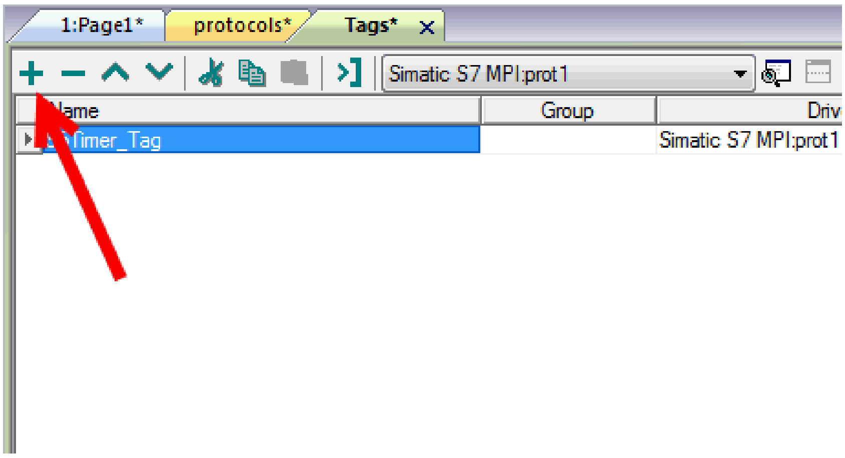
The conversion will be listed into the Configured window on the right.

Confirm with OK button.

Simatic S5timer data type

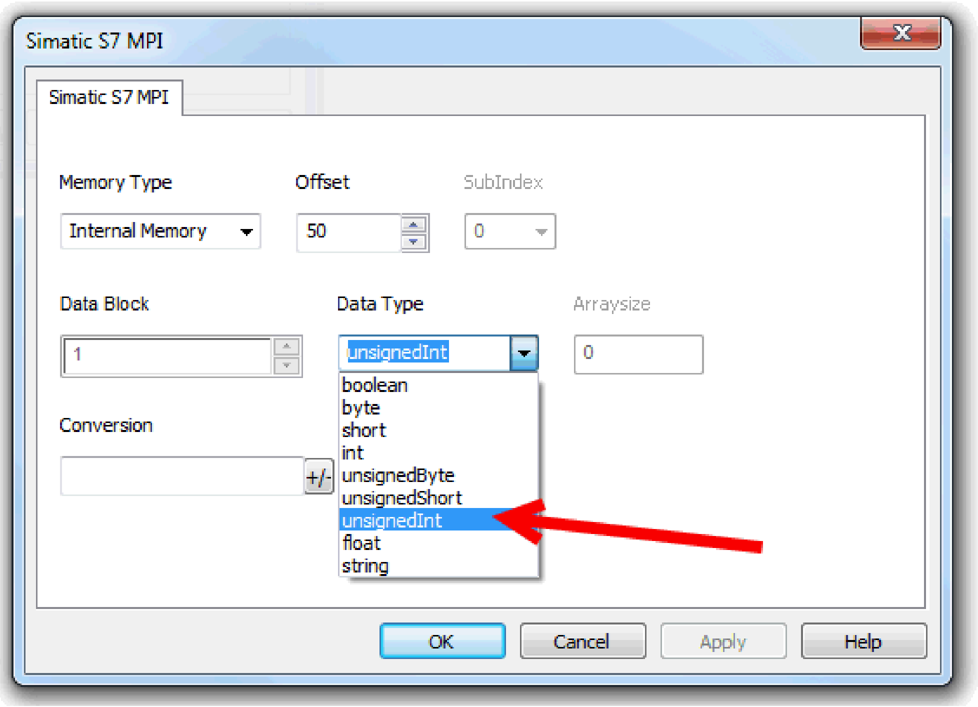
Simatic drivers support a special data type, called S5Timer.
The tag must be configured with a specific data type and a conversion must be applied to the Tag to correctly read/write a Simatic S5Timer Variable.

Open the Tag Editor and add a Tag pressing the Plus button.



Name	Group	Driv
Timer_Tag		Simatic S7 MPI:prot1

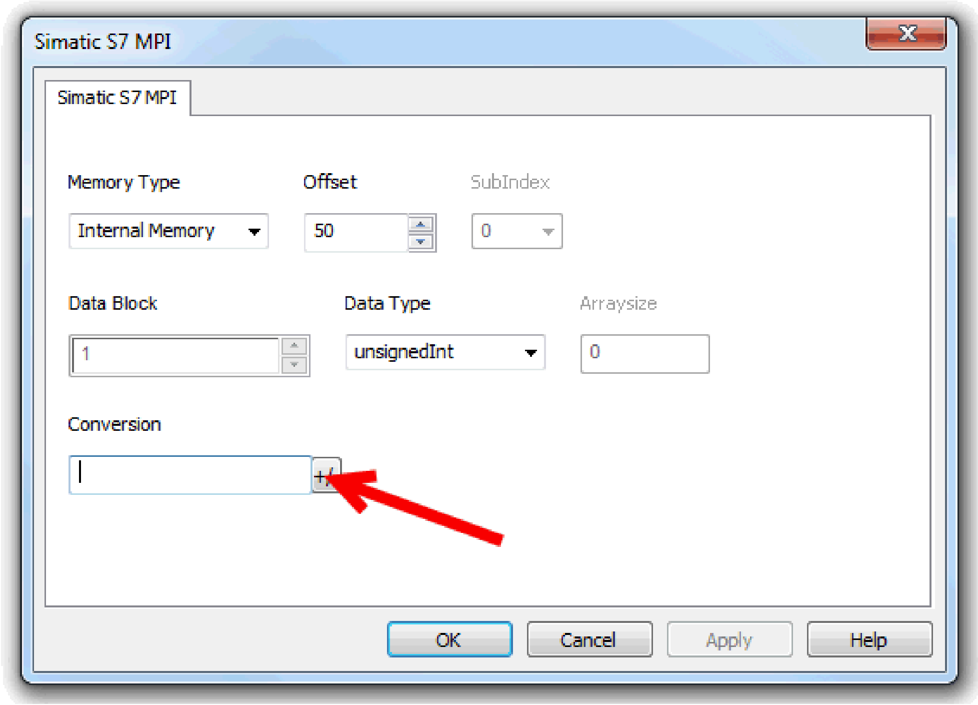
Select “unsignedInt” as Data Type of the Tag.



Simatic S7 MPI

Memory Type: Internal Memory
Offset: 50
SubIndex: 0
Data Block: 1
Data Type: unsignedInt
Arraysize: 0
Conversion:

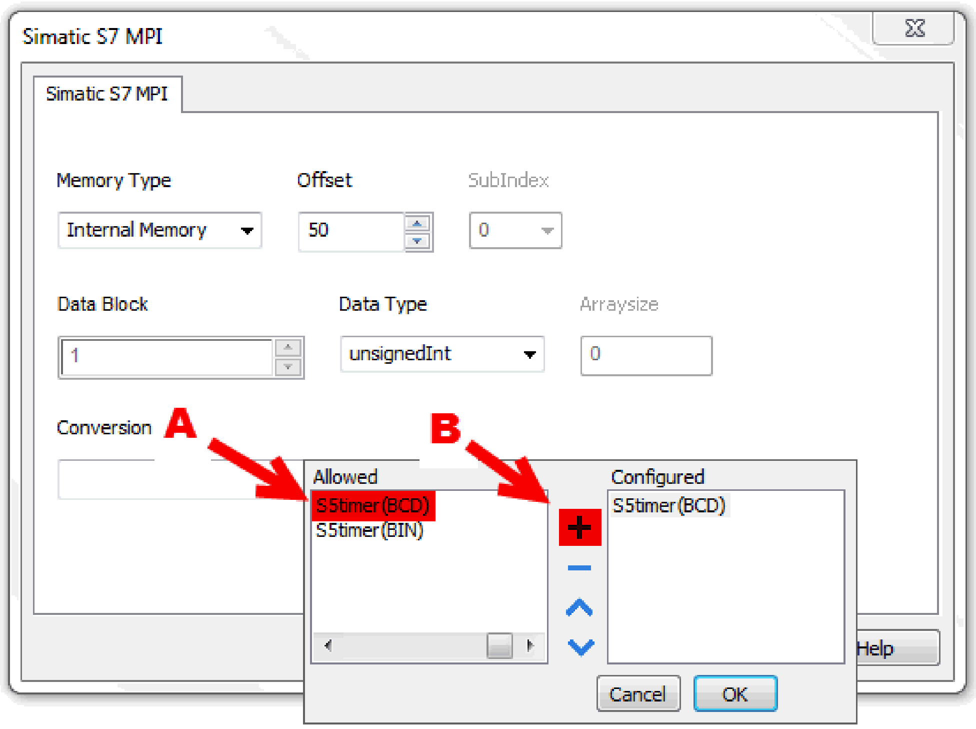
Click on +/- button to open the Conversion dialog.



Simatic S7 MPI

Memory Type: Internal Memory
Offset: 50
SubIndex: 0
Data Block: 1
Data Type: unsignedInt
Arraysize: 0
Conversion:

In the Conversion dialog select the S5timer(BCD) conversion type [A] then click on Plus button [B] to add the conversion, the configured conversion will be listed into the Configured window on the right. Then confirm with OK.



Memory Type: Internal Memory

Offset: 50

SubIndex: 0

Data Block: 1

Data Type: unsignedInt

Arraysize: 0

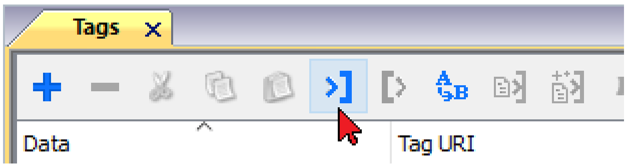
Conversion

Allowed: S5timer(BCD), S5timer(BIN)

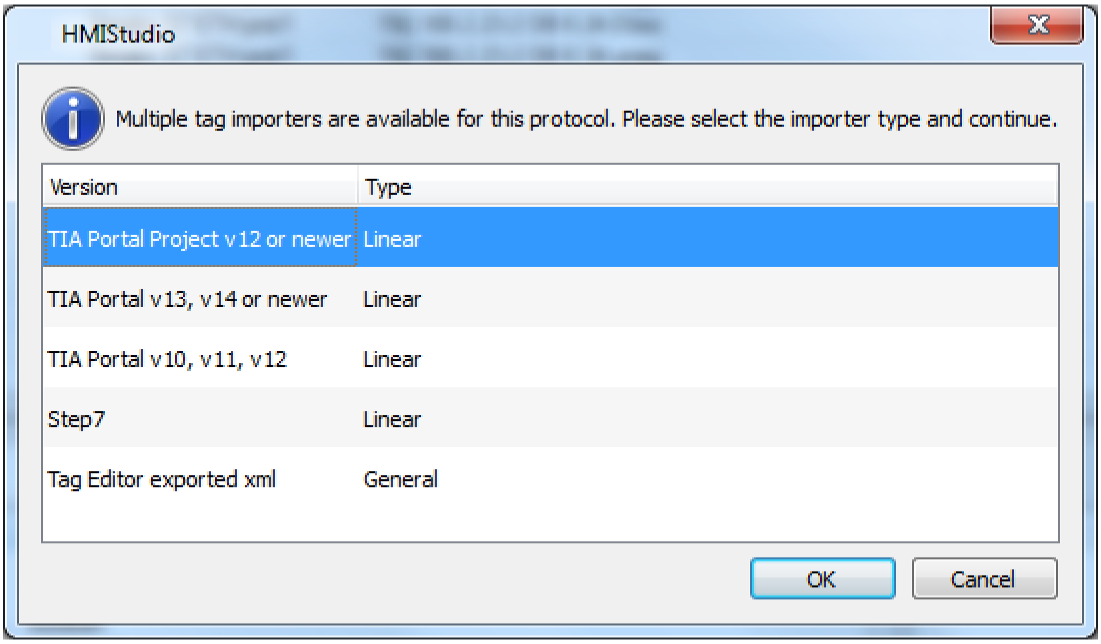
Configured: S5timer(BCD)

Tag Import

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

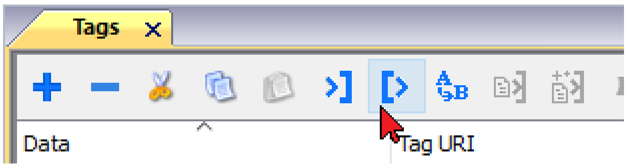


The following dialog shows which importer type can be selected.



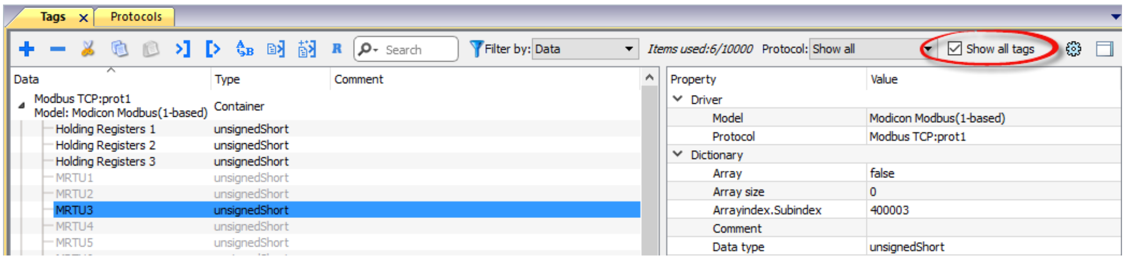
Multiple tag importers are available for this protocol. Please select the importer type and continue.

Version	Type
TIA Portal Project v12 or newer	Linear
TIA Portal v13, v14 or newer	Linear
TIA Portal v10, v11, v12	Linear
Step7	Linear
Tag Editor exported xml	General

Importer	Description
TIA Portal Project v12 or newer **Linear**	Allows to import the whole TIA Portal project file using **.apxx** 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.
TIA Portal v13, v14 or newer **Linear**	Allows to import: • Program blocks using **.db** file • PLC tags using **.xlsx** file • PLC data types using **.udt** file Check **Export using TIA Portal v13, v14 or newer** for more details. All variables will be displayed at the same level.
TIA Portal v10, v11, v12 **Linear**	Allows to import: • Program blocks using **.tia** file • PLC tags using **.xlsx** file • PLC data types using **.scl** file Check **Export using TIA Portal v10, v11, v12** for more details. All variables will be displayed at the same level.
Step7 **Linear**	Allows to import: • Symbols table **.asc** file • Sources using **.awl** file Check **Export using STEP7** for more details. 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. 

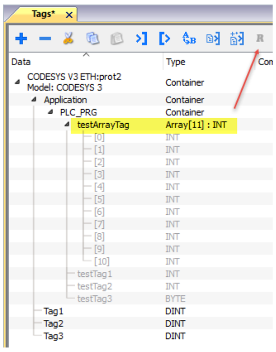
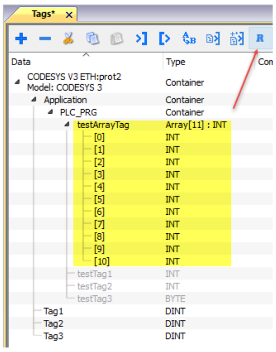
Once the importer has been selected, locate the symbol file and click **Open**.

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.



Data	Type	Comment
Modbus TCP:prot1 Model: Modicon Modbus(1-based)	Container	
Holding Registers 1	unsignedShort	
Holding Registers 2	unsignedShort	
Holding Registers 3	unsignedShort	
MRTU1	unsignedShort	
MRTU2	unsignedShort	
MRTU3	unsignedShort	
MRTU4	unsignedShort	
MRTU5	unsignedShort	

Property	Value
Driver	
Model	Modicon Modbus(1-based)
Protocol	Modbus TCP:prot1
Dictionary	
Array	false
Array size	0
Arrayindex.Subindex	400003
Comment	
Data type	unsignedShort

Toolbar item	Description
	Import Tag(s). Select tags to be imported and click on this icon to add tags from tag dictionary to the project
	Update Tag(s). Click on this icon to update the tags in the project, due a new dictionary import.
R	Check this box to import all sub-elements of a tag. Example of both checked and unchecked result:  
Search Filter by: Tag name	Searches tags in the dictionary basing on filter combo-box item selected.

Data	Type
CODESYS V3 ETH:prot2 Model: CODESYS 3	Container
Application	Container
PLC_PRG	Container
testArrayTag	Array[11] : INT
[0]	INT
[1]	INT
[2]	INT
[3]	INT
[4]	INT
[5]	INT
[6]	INT
[7]	INT
[8]	INT
[9]	INT
[10]	INT
testTag1	INT
testTag2	INT
testTag3	BYTE
Tag1	DINT
Tag2	DINT
Tag3	DINT

Communication status

The communication status can be displayed using the dedicated system variables. Please refer to the User Manual for further information about available system variables and their use.

The status codes supported for this communication driver are:

Error	Notes
NAK	Controller replies with a not acknowledge.
Timeout	Request is not replied within the specified timeout period; ensure the controller is connected and properly configured for network access

Error	Notes
Invalid response	The device did receive from the controller a response, but its format or its contents or its length is not as expected; ensure the data programmed in the project are consistent with the controller resources.
General Error	Error cannot be identified; 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

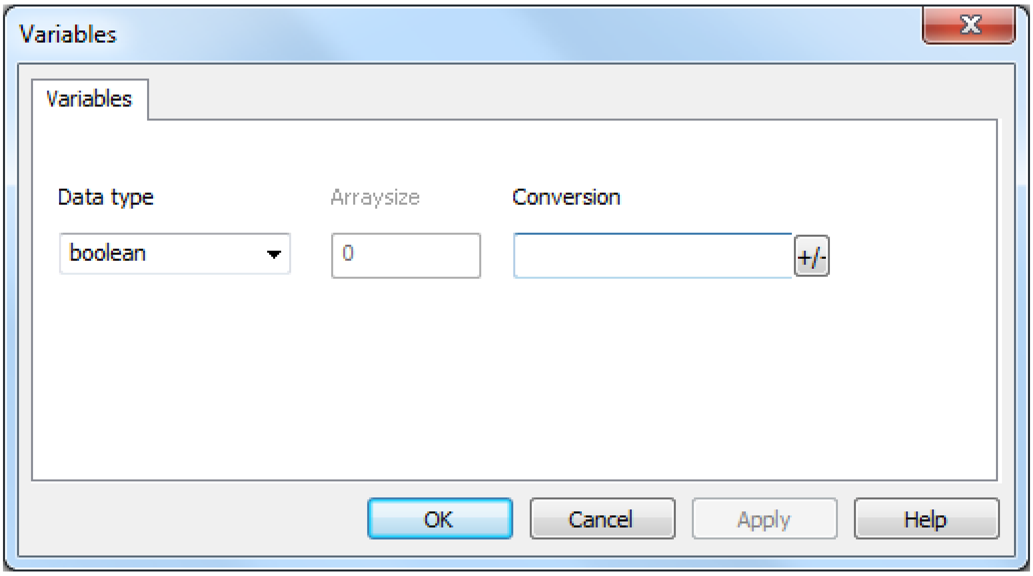
To configure the protocol:

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.

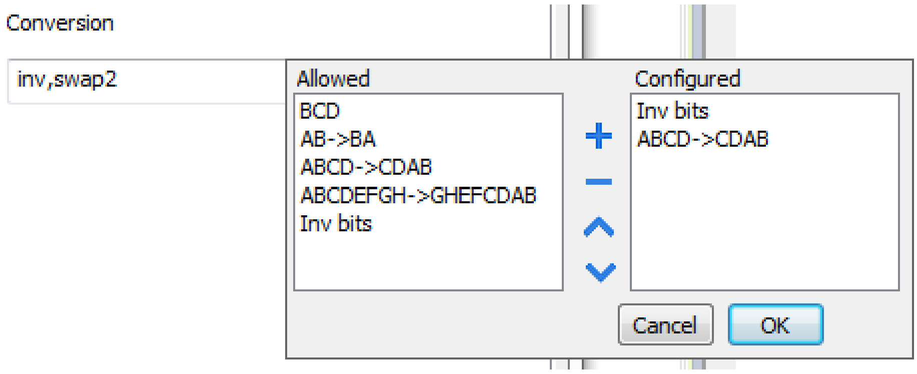


Data type: boolean

Arraysize: 0

Conversion:

<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td><b>Data Type</b></td><td>
<table>
<tr><th>Data Type</th><th>Memory Space</th><th>Limits</th></tr>
<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>
<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>
</table>
Note: to define arrays. select one of Data Type format followed by square brackets like “byte[]”, “short[]”…
</td></tr>
<tr><td><b>Arraysize</b></td><td>
• 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.
</td></tr>
<tr><td><b>Conversion</b></td><td>Conversion to be applied to the tag.</td></tr>
</table>

<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td></td><td>

Conversion
inv,swap2
Allowed
BCD
AB->BA
ABCD->CDAB
ABCDEFGH->GHEFCDAB
Inv bits
Configured
Inv bits
ABCD->CDAB

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

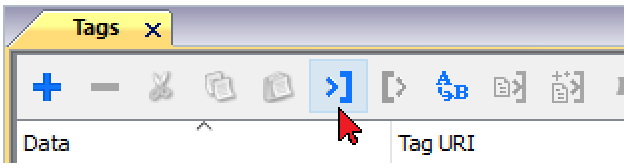
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td><b>Inv bits</b></td><td><b>inv</b>: Invert all the bits of the tag.
<i>Example:</i>
1001 → 0110 (in binary format)
9 → 6 (in decimal format)</td></tr>
<tr><td><b>Negate</b></td><td><b>neg</b>: Set the opposite of tag value.
<i>Example:</i>
25.36 → -25.36</td></tr>
<tr><td><b>AB -> 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 -> 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 -> 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 -> OPM...DAB</b></td><td><b>swap8</b>: Swap bytes in a long word.
Example:
142.366 → -893553517.588905 (in decimal format)
0 10000000110</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td></td><td>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>
</table>

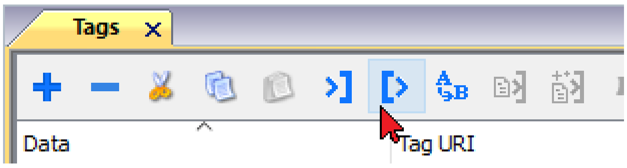
Select conversion and click +. The selected item will be added to list <b>Configured</b>.
If more conversions are configured, they will be applied in order (from top to bottom of list <b>Configured</b>).
Use the arrow buttons to order the configured conversions.</td></tr>
</table>

Tag Import

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

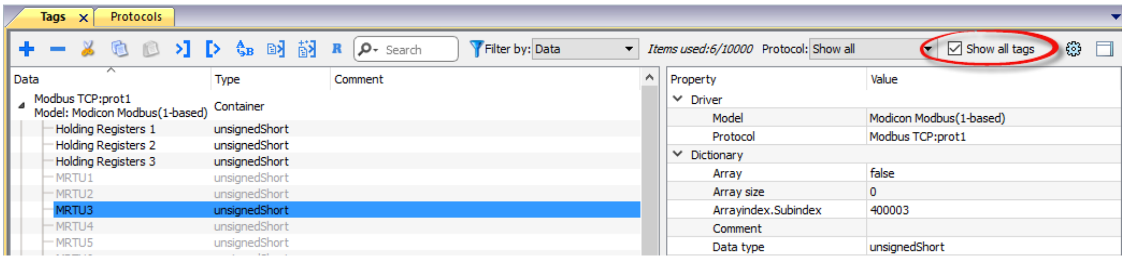


The system will require a generic XML file exported from Tag Editor by appropriate button.



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.

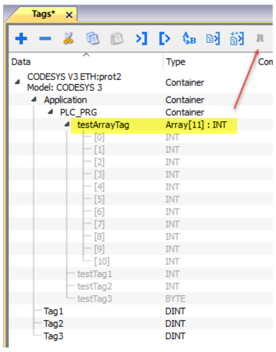
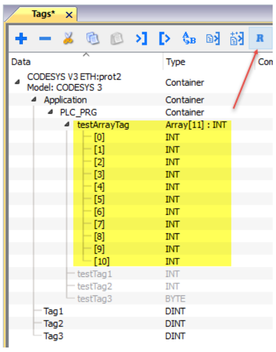


Tags | Protocols

Search | Filter by: Data | *Items used:6/10000* | Protocol: Show all | ☑ Show all tags

Data	Type	Comment
Modbus TCP:prot1 Model: Modicon Modbus(1-based)	Container	
Holding Registers 1	unsignedShort	
Holding Registers 2	unsignedShort	
Holding Registers 3	unsignedShort	
MRTU1	unsignedShort	
MRTU2	unsignedShort	
MRTU3	unsignedShort	
MRTU4	unsignedShort	
MRTU5	unsignedShort	

Property	Value
Driver	
Model	Modicon Modbus(1-based)
Protocol	Modbus TCP:prot1
Dictionary	
Array	false
Array size	0
Arrayindex.Subindex	400003
Comment	
Data type	unsignedShort

Toolbar item	Description
	Import Tag(s). Select tags to be imported and click on this icon to add tags from tag dictionary to the project
	Update Tag(s). Click on this icon to update the tags in the project, due a new dictionary import.
R	Check this box to import all sub-elements of a tag. Example of both checked and unchecked result: 
Search Filter by: Tag name	Searches tags in the dictionary basing on filter combo-box item selected.

Tags*

Data	Type
CODESYS V3 ETH:prot2 Model: CODESYS 3	Container
Application	Container
PLC_PRG	Container
testArrayTag	Array[11] : INT
[0]	INT
[1]	INT
[2]	INT
[3]	INT
[4]	INT
[5]	INT
[6]	INT
[7]	INT
[8]	INT
[9]	INT
[10]	INT
testTag1	INT
testTag2	INT
testTag3	BYTE
Tag1	DINT
Tag2	DINT
Tag3	DINT