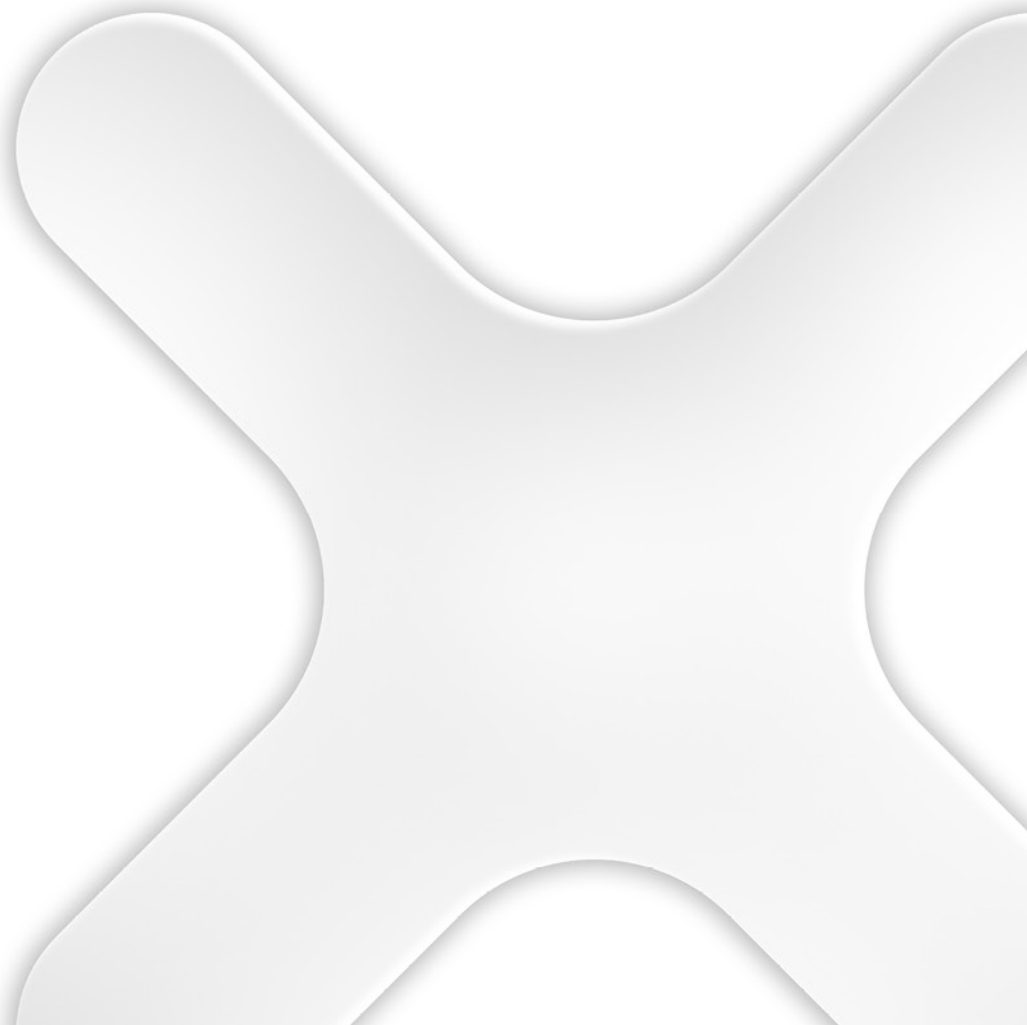


# JMobile Studio

User Manual



© 2009-2018 Exor International S.p.A.

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.

Microsoft®, Win32, Windows®, Windows XP, Windows Vista, Windows 7, Windows 8, Visual Studio are either registered trademarks or trademarks of the Microsoft Corporation in the United States and other countries. Other products and company names mentioned herein may be the trademarks 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

<b>1 Getting started</b>	<b>1</b>	Managing overlapping widgets	60
Assumptions	2	Grouping widgets	61
What's new	2	Changing multiple widgets properties	68
Installing the application	4	Changing fill color property according to tag values	69
<b>2 Runtime</b>	<b>9</b>	<b>6 Project properties</b>	<b>71</b>
HMI device basic settings	10	Runtime	72
Context menu options	10	Plug-in	77
Built-in SNTP service	13	Project	78
<b>3 Runtime on PC</b>	<b>15</b>	Web	82
Typical installation problems	17	Events	83
<b>4 My first project</b>	<b>21</b>	<b>7 The HMI simulator</b>	<b>85</b>
The workspace	22	Data simulation methods	86
Creating a project	22	Simulator settings	86
Communication protocols	24	Launching and stopping the simulator	87
Designing a page	26	<b>8 Transferring the project to HMI device</b>	<b>89</b>
The Widget Gallery	27	Download to HMI device	90
Data field widget	29	Update package	93
Adding tags	32	The Runtime loader	95
Attaching widget to tags	34	Upload projects	96
Dialog pages	36	<b>9 System Variables (Attach To)</b>	<b>97</b>
Exporting tags	37	Alarms variables	99
Importing tags	37	Buzzer variables	99
Tag find and rename	40	Communication variables	100
Tag find and replace	42	Database variables	100
<b>5 Programming concepts</b>	<b>45</b>	Daylight Saving Time variables	101
Data types	46	Device variables	102
"Attach to" parameters	46	Dump information variables	103
Formula	52	FTP client variables	104
Events	56	Keypad variables	105
Widgets positioning	59		

Network variables .....	105	Remote Client actions .....	183
PLC variables .....	105	System actions .....	184
Printing variables .....	106	Tag actions .....	196
Remote Client variables .....	107	Trend actions .....	198
Version variables .....	108	Text Editor actions .....	202
Screen variables .....	108	User management actions .....	202
SD card variables .....	108	Widget actions .....	205
Server variables .....	109	<b>12 The HMI Client .....</b>	<b>209</b>
Time variables .....	109	Client application on PC .....	210
Touch screen variables .....	110	Client application on HMI .....	211
USB drive variables .....	111	Settings and time zone options .....	212
User management variables .....	111	<b>13 Using the integrated FTP server .....</b>	<b>217</b>
<b>10 System Variables (Protocol) .....</b>	<b>113</b>	FTP settings .....	217
Protocol Editor Settings .....	114	<b>14 Using VNC for remote access .....</b>	<b>219</b>
Default variables .....	114	Starting VNC server on WinCE devices .....	220
HandHeld variables .....	129	Starting VNC server on Linux devices .....	221
HandHeld - wireless variables .....	133	Starting VNC viewer .....	221
Direct I/O variables .....	139	<b>15 Corvina Cloud .....</b>	<b>223</b>
PLCM09 variables .....	144	<b>16 Alarms .....</b>	<b>225</b>
Services variables .....	148	Alarms Editor .....	226
Retentive Memory variables .....	151	Remote alarms acknowledge .....	228
Tag Import .....	156	Alarm state machine .....	229
<b>11 Actions .....</b>	<b>159</b>	Setting events .....	230
Alarm actions .....	160	Active Alarms widget .....	232
Database actions .....	161	Alarms History widget .....	237
Event actions .....	164	Managing alarms at run time .....	237
MultiLanguage actions .....	164	Enable/disable alarms at run time .....	238
Keyboard actions .....	165	Displaying live alarm data .....	238
Media Player actions .....	167	Exporting alarm buffers to .csv files .....	239
FTP actions .....	167	Exporting alarm configuration .....	239
Page actions .....	170	<b>17 Recipes .....</b>	<b>243</b>
Print actions .....	177	Managing recipes .....	243
Recipe actions .....	178	Configuring a recipe widget .....	246



Recipe status .....	247	Recurring schedule .....	290
Uploading/downloading a recipe .....	248	Configuring location for schedules .....	292
Backup and restore recipes data .....	249	Configuring the Scheduler widget .....	293
<b>18 Trends .....</b>	<b>251</b>	Scheduling events at run time .....	294
Data logging .....	252	<b>23 21 CFR Part 11 Compliance .....</b>	<b>297</b>
Exporting trend buffer data .....	254	x.509 Certificate .....	302
Trend widgets .....	255	21 CFR Part 11 - Tips .....	306
History trends .....	257	<b>24 User management and passwords .....</b>	<b>309</b>
Trend widget properties .....	259	Enable/disable security management .....	310
Trend widget gestures .....	260	Configuring groups and authorizations .....	310
Values outside range or invalid .....	261	Modifying access permissions .....	311
Showing trend values .....	262	Assigning widget permissions from page view .....	316
Scatter diagram widget .....	263	Configuring users .....	317
Table trend widget .....	264	Default user .....	318
<b>19 Data transfer .....</b>	<b>269</b>	Managing users at run time .....	318
Data transfer editor .....	270	Force remote login .....	319
Exporting data to .csv files .....	272	<b>25 Audit trails .....</b>	<b>321</b>
Data transfer limitations and suggestions .....	272	Enable/disable audit trail .....	322
<b>20 Offline node management .....</b>	<b>275</b>	Electronic Signature .....	323
Offline node management process .....	276	Table audit widget .....	326
Manual offline node management process .....	276	Exporting audit trail as .csv files .....	327
Manual offline configuration .....	276	Configure audit events .....	328
Automatic offline node detection .....	277	<b>26 Reports .....</b>	<b>331</b>
<b>21 Multi-language .....</b>	<b>279</b>	Adding a report .....	332
The Multi-language editor .....	281	Configuring text reports .....	332
Changing language .....	282	Configuring graphic reports .....	333
Multi-language widgets .....	282	Print triggering events .....	334
Exporting/importing multi-language strings .....	284	Default printer .....	335
Changing language at run time .....	286	<b>27 Screen saver .....</b>	<b>339</b>
Limitations in Unicode support .....	286	<b>28 Backup/restore of Runtime and project .....</b>	<b>341</b>
<b>22 Scheduler .....</b>	<b>289</b>	<b>29 Keypads .....</b>	<b>343</b>
Creating a schedule .....	290	Creating and using custom keypads .....	345
HighResolution schedule .....	290		

Deleting or renaming custom keypads .....	347	BACnet widget .....	386
Keypad type .....	347	Browser widget .....	386
Keypad position .....	348	Canvas Widget .....	387
<b>30 External keyboards .....</b>	<b>349</b>	Combo Box widget .....	390
Search and filter .....	351	Consumption Meter widget .....	394
Displayed keys .....	351	Control list widgets .....	395
Removing action associations .....	351	DateTime widget .....	399
Keyboard layout .....	352	Gesture area widget .....	400
Enable/disable keyboard .....	352	IP Camera widgets .....	405
Associating actions to keys .....	352	Javascript function block widget .....	408
<b>31 Tag cross reference .....</b>	<b>355</b>	Media Player widgets .....	410
Updating data in the Tag Cross Reference pane .....	358	Multistate Image widget .....	412
<b>32 Indexed addressing .....</b>	<b>359</b>	Multistate Image Multilayer widget .....	413
Creating an indexed addressing set .....	360	Network Adapters widget .....	415
Using indexed tag set in pages .....	363	RSS Feed widget .....	415
<b>33 Storing data to external databases .....</b>	<b>365</b>	Scrolling RSS Feed widget .....	416
Installing SQL4Automation .....	367	Table widget .....	417
Configuring SQL4Automation .....	367	TextEditor widget .....	430
Configuring the HMI project .....	369	Variables widget .....	432
Transfer data with JavaScript .....	370	<b>36 Custom widgets .....</b>	<b>435</b>
Database tables .....	371	Creating a custom widget .....	436
Custom tables .....	372	Adding properties to a custom widget .....	438
Connection Limits .....	372	Using structured tags .....	441
<b>34 OPC UA Server .....</b>	<b>375</b>	JavaScript in custom widgets .....	443
Features .....	375	User's Gallery .....	446
Network .....	375	<b>37 Sending an email message .....</b>	<b>449</b>
Authentication .....	375	Configuring the email server .....	450
Using x.509 Certificates .....	376	Configure emails .....	450
Using self-signed certificates .....	379	<b>38 JavaScript .....</b>	<b>453</b>
Using external certificates .....	383	JavaScript editor .....	455
Alarm map .....	384	Execution of JavaScript functions .....	455
<b>35 Special widgets .....</b>	<b>385</b>	Events .....	457
		Widget events .....	458

Page events .....	460	Testing the Web project .....	536
System events .....	461	Downloading the Web project .....	537
Objects .....	463	Web connectivity issues .....	538
Widget class objects .....	463	Web supported features .....	539
Widget properties .....	464	Troubleshooting and FAQ .....	543
Widget methods .....	467	Privacy .....	543
Page object .....	468	<b>42 License activation of HMI device</b>	
Page object properties .....	468	<b>software modules .....</b>	<b>545</b>
Page object methods .....	469	Activate the device .....	546
Group object .....	471	Save a license .....	547
Group object methods .....	471	Import a license .....	547
Project object .....	473	<b>43 Updating system components in HMI</b>	
Project object properties .....	473	<b>devices .....</b>	<b>549</b>
Project object methods .....	473	Update of system components from the	
Project object widgets .....	482	application .....	550
Print reports object .....	483	Settings .....	551
State object .....	486	<b>44 Protecting access to HMI devices .....</b>	<b>553</b>
State object methods .....	486	Changing password on HMI device .....	554
Keywords .....	487	Ports and firewalls .....	554
Global functions .....	488	<b>45 Tips and tricks to improve performance .....</b>	<b>557</b>
Handling read/write files .....	488	Static Optimization .....	558
Sign in from JavaScript .....	491	FAQ on Static Optimization .....	561
Limitations in working with widgets in		Page caching .....	562
JavaScript .....	493	Image DB .....	562
Debugging of JavaScript .....	494	Precaching .....	562
		FAQ on precaching .....	562
<b>39 Handling Gestures .....</b>	<b>497</b>	<b>46 Functional specifications and</b>	
<b>40 System Settings .....</b>	<b>499</b>	<b>compatibility .....</b>	<b>565</b>
Linux Devices .....	500	Table of functions and limits .....	566
WinCE Devices .....	524	HMI devices capabilities .....	567
<b>41 Web access .....</b>	<b>533</b>	Compatibility .....	569
Supported platforms and browsers .....	534	Converting projects between different HMI	
Generating page for Web access .....	534	devices .....	569
Platform specific Home pages .....	536		



# 1 Getting started

---

JMobile Studio is a software application designed to create graphical HMI pages. JMobile Studio 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 JMobile Studio.

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

---

<b>Assumptions .....</b>	<b>2</b>
<b>What's new .....</b>	<b>2</b>
<b>Installing the application .....</b>	<b>4</b>

# Assumptions

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

## What's new

### What's new on v2.8

- Custom widgets and inheritance of widget properties
- Enhanced JM4Web function coverage including user management, custom widgets, canvas widget. Improved responsive design based on grid layout.
- Support for application compliance with 21 CFR part 11 including electronic signatures and security with certificates
- Calculation of expressions with tag values
- New alarm and trend widgets based on table widget
- Audit trail widget
- Enhanced support for multitouch and gestures
- Tag find with rename and replace functions
- Remove unused tags from project
- Source timestamps in tag data acquisition for OPC UA sources
- Video player for Linux-based HMI devices
- Trend dump when full
- Direct Serial protocol for creating custom communication protocols
- Direct Socket protocol for creating custom communication protocols
- Extended project limits for high-end HMI devices
- Runtime performance enhancement with faster boot time
- Status variables for CODESYS PLC
- Remote HMI Client software for Linux-based HMI devices
- Complete BACnet support with BACnet alarms and trends
- Added support for new EXOR products (JSmart, eXware, eX FB)

## What's new on v2.6

- Grid layout
- Table widget
- Canvas widget
- Javascript Function Block Widget
- Structured tags support for custom widgets
- "Show all" option for Alarm History widget (Enhanced the Alarm History widget to be able to show all alarms)
- "Attach to ..." support for properties in e-mail server configuration
- JM4Web support for trends
- JM4Web support for recipes
- BACnet protocol support for Calendar/Scheduler
- Revision of target limits
- Extend OPC-UA server to support Alarms and Trends
- FTP Client
- Enhanced Combo Box to work in full screen with pictures support
- Added support for manually edit and change data links
- Enhanced the Data Transfer to support alias (indexed tags)
- When required, the "Download to target" invoke the "update runtime" while downloading project.
- New System Variable to check the status of dumped files

## What's new on v2.4

- Project Limits check when converting a project (not just on download as before)
- Disable unavailable protocols based on device selected
- Disable SD card selection in devices where it is not available
- IP Widget for setting networks parameters.
- Introduced "WebPageRequest" to control page shown from a PLC
- New project property to set the runtime background color
- OPC-UA Server support
- SQL4Automation support
- JS support on JM4Web
- Multilanguage support on JM4Web
- Alarm state strings can be customized and translated
- PDF viewer support for Linux products
- Support for embedded custom JS in Widget gallery
- Gestures support
- Introduce push engine for data sent/received. Now JM4Web can refresh data and widget up to 10 times a second
- Redesign of Manage Target to update multiple HMI device
- New option to enable alarms from PLCs
- New option to execute a user macro action on alarm
- Possibility to group and filter alarms

## What’s new in comparison to v2.0.0

- Support of Windows 10 32-/64-bit
- User’s gallery for customized widgets
- Maximum number of tags for JMobile PC Runtime: 10000
- German online-help / manual

## Installing the application

JMobile Suite installation contains:

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

## JMobile Studio system requirements

JMobile Studio has the following system requirements:

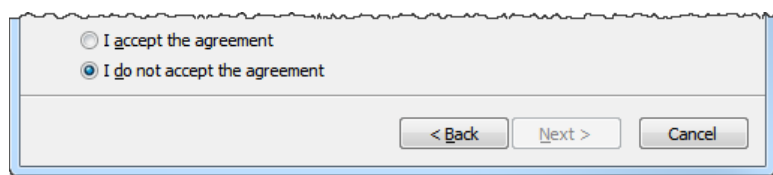
<b>Operating System</b>	Windows XP Professional Windows XP Embedded Windows Embedded Standard (WES 2009) Windows Server 2003 Windows Vista Business/Ultimate Windows 7 Professional Windows Embedded Standard 7 Windows 8 Windows 10
<b>Storage</b>	500 MB Minimum
<b>RAM</b>	512 MB
<b>Other</b>	One Ethernet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

## Installation procedure

To install JMobile Suite:



1. Run JMobile Suite setup and click **Next**.
2. Read the JMobile Studio Software License and accept the agreement.



3. Follow the instructions on the screen. The default location for the c software is *C:\Program Files\Exor\JMobile Suite*, 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 JMobile Studio icon on your desktop. A JMobile Suite group is automatically added to the **Start** menu by the installation procedure.



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

## Trial version

JMobile Studio 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**.



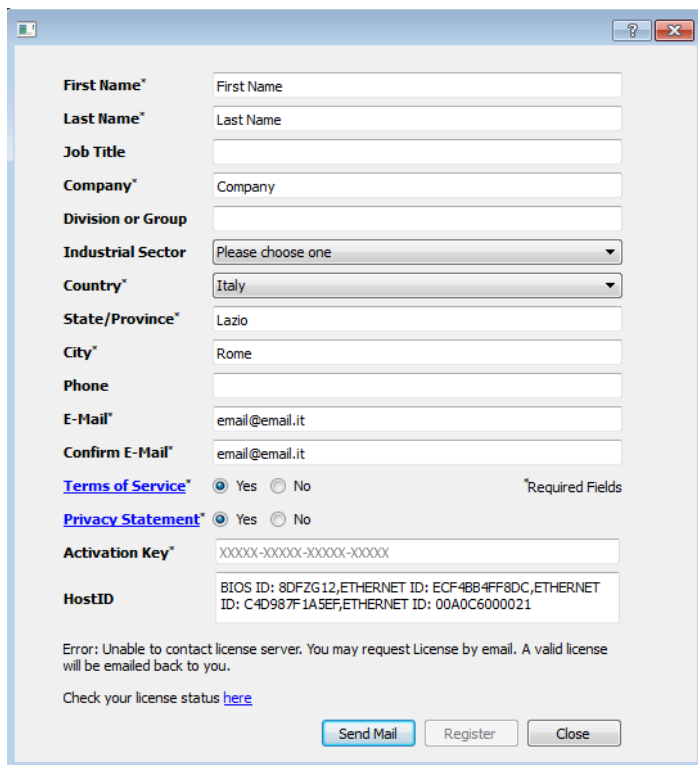
Note: The registration process requires an Internet connection. Ports TCP 80 and 443 are used for activation.

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%\Exor`

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

If JMobile Studio 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.



Registration form for JMobile Studio. Fields include:

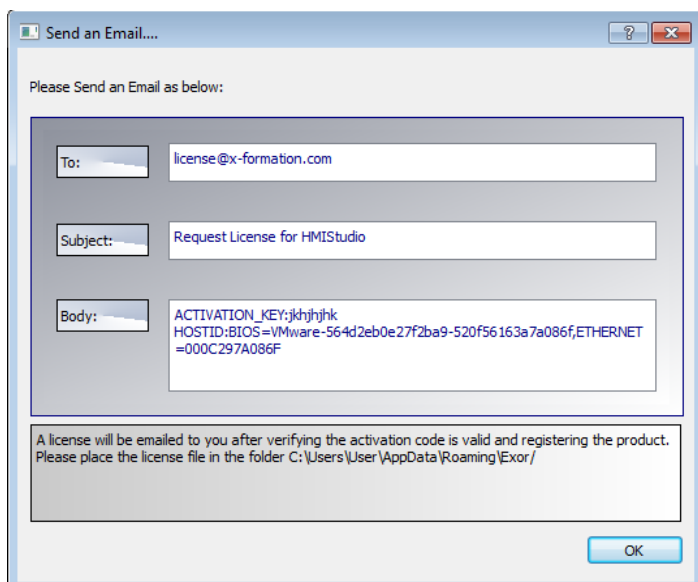
- First Name\*
- Last Name\*
- Job Title
- Company\*
- Division or Group
- Industrial Sector (Please choose one)
- Country\* (Italy)
- State/Province\* (Lazio)
- City\* (Rome)
- Phone
- E-Mail\* (email@email.it)
- Confirm E-Mail\* (email@email.it)
- Terms of Service\* (Yes/No)
- Privacy Statement\* (Yes/No)
- Activation Key\* (XXXXX-XXXXX-XXXXX-XXXXX)
- HostID (BIOS ID: 8DFZG12, ETHERNET ID: ECF4BB4FF8DC, ETHERNET ID: C4D987F1A5EF, ETHERNET ID: 00A0C6000021)

Error: Unable to contact license server. You may request License by email. A valid license will be emailed back to you.

Check your license status [here](#)

Buttons: Send Mail, Register, Close

Pressing the “Send Mail” button the JMobile Studio will display this form:



Send an Email....

Please Send an Email as below:

To: license@x-formation.com

Subject: Request License for HMISudio

Body: ACTIVATION\_KEY:jkjhjhkh  
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\

OK

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 JMobile Suite

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

Three installation scenarios are possible:

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

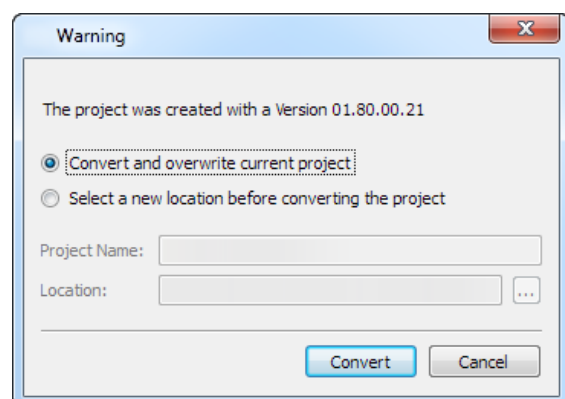
If you try to install a second instance of an already installed version of JMobile Suite, a warning message is displayed.

Multiple JMobile Suite installations share a common workspace folder, each sub-folder includes the version number, for example *C:\Program Files\Exor\JMobile Suite 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 JMobile Studio project (.jpr file) created with an older version of the software JMobile Studio asks to convert the project to the current version:



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



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

## Multilanguage for JMobile Studio

JMobile Studio is available in multiple languages. All languages are installed by default as part of JMobile Suite.

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

## Crash reports

A crash report dialog appears whenever JMobile Studio 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

---

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

---

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

# HMI device basic settings

HMI devices are delivered from factory without Runtime. If no Runtime is installed on the device, see ["The Runtime loader" on page 95](#) for details.

## Runtime modes

The JMobile 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.

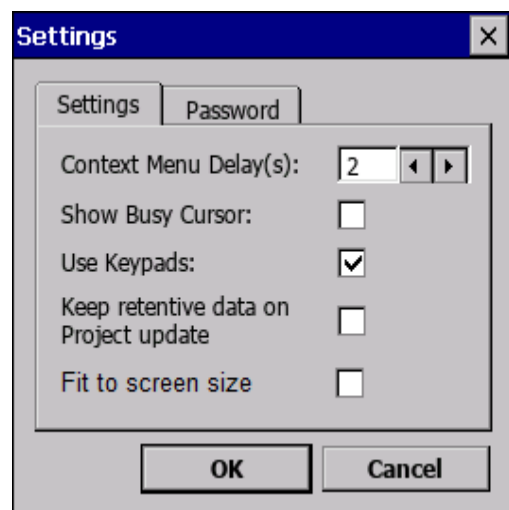
### Zoom In/Out

Select view size at run time

### Pan Mode

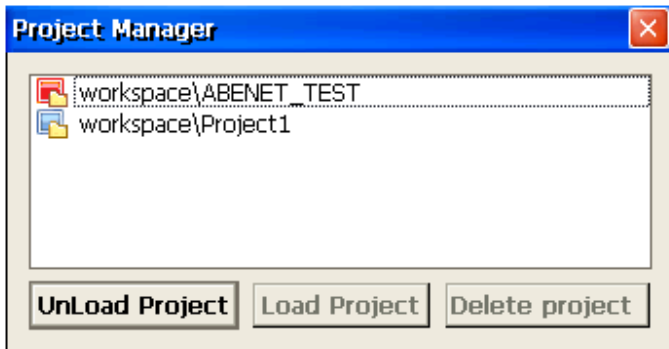
Enables/disables pan mode after a zoom in

## Settings



Main parameters	Description
<b>Context Menu Delay (s)</b>	Context menu activation delay. Range: 1–60 seconds.
<b>Show Busy Cursor</b>	Display an hourglass when the system is busy
<b>Use keypads</b>	Display keypads when user touches a data entry field. Set to <b>disable</b> when an external USB keyboard is connected to the device.
<b>Keep retentive data on project update</b>	Preserve the content of the retentive data at project download or update.
<b>Fit to screen size</b>	Adapts the view to the screen size
<b>Password</b>	Define password protected operations amongst the following: <ul style="list-style-type: none"> <li>• Download Project/Runtime</li> <li>• Upload project</li> <li>• Board management (BSP Update)</li> </ul> See " <a href="#">Protecting access to HMI devices</a> " on page 553 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

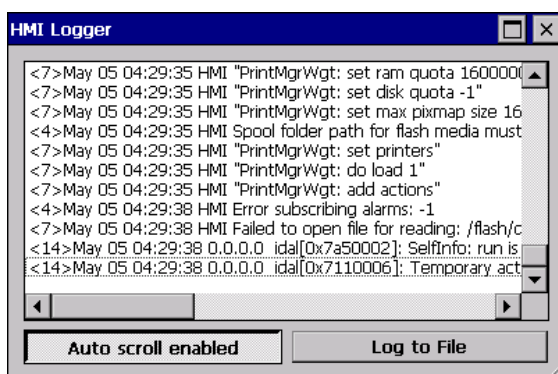
This function loads update packages from an external USB drive. See ["Update package" on page 93](#) for details.

## Backup

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

## Logging

This function displays a log of system operations.



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 499](#) for details.

## HMI Cloud Enabled

Open the HMI Cloud Enabler.



Command available only for HMI devices based on Windows CE platform.

On HMI devices based on the Linux platform the Service Cloud can be enabled from the "Service" tab of the ["Linux Devices" on page 500](#).

## Developer tools

Utility functions for debugging at run time. It is visible only if enabled in the Project Properties (see ["Developer tools" on page 74](#) 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. The system searches the SNTP server when turned on, or once a week if the HMI device is not turned off.

Use HMI device ["System Settings" on page 499](#) to configure the service.



*Availability: BSP v1.76 ARM / 2.79 MIPS or higher*



## 3 Runtime on PC

JMobile 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.

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

### JMobile PC Runtime system requirements

JMobile PC Runtime as the following minimum system requirements:

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

### Installing Runtime

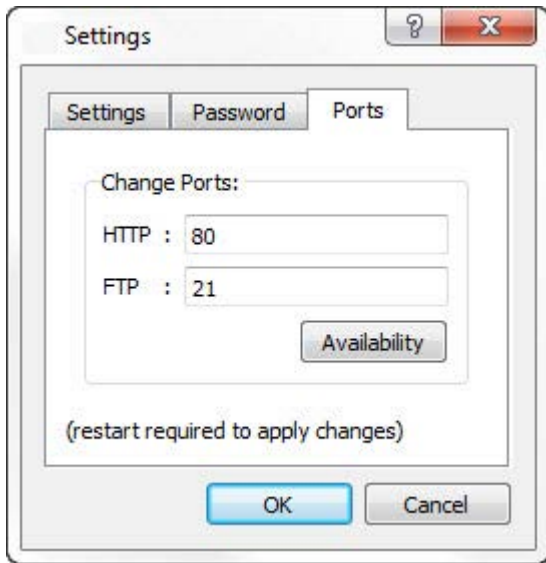
JMobile PC Runtime could be distributed as a component of the JMobile Suite or as a standalone application. When installing the software make sure that you select the **Runtime PC** option in the **Select Components** dialog.

### Multiple instances of JMobile PC Runtime

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



## Licensing

JMobile 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.

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



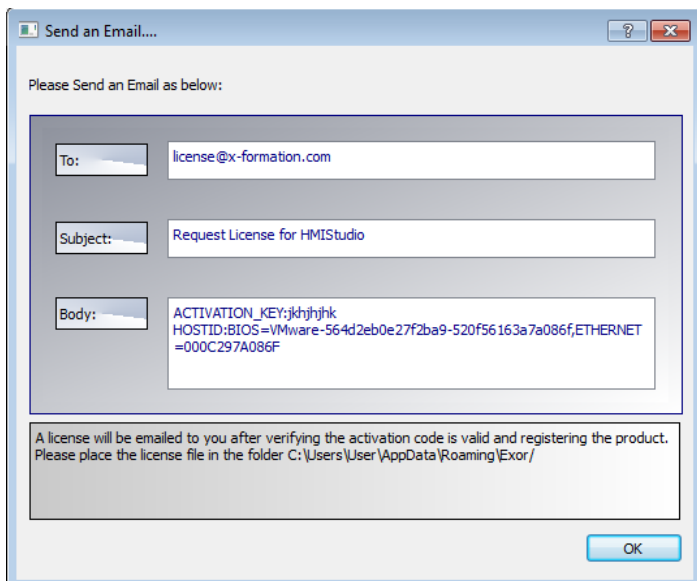
Note: the registration process requires an Internet connection. Ports TCP 80 and 443 are used for activation.

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%\Exor*

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

If JMobile 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.



See ["Installing the application" on page 4](#) for instructions on how to verify the activation status.

## Limitations

The following features are not supported in JMobile 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 565](#) for more details.

## Fullscreen mode

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

To switch to full screen:

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

## The workspace folder

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

`%appdata%\Exor\[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.

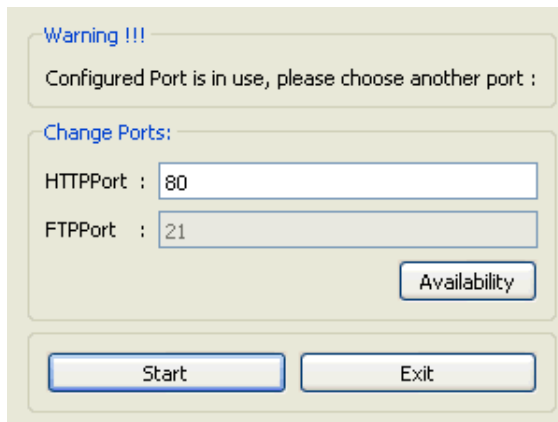
See ["Protecting access to HMI devices" on page 553](#) for details.

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

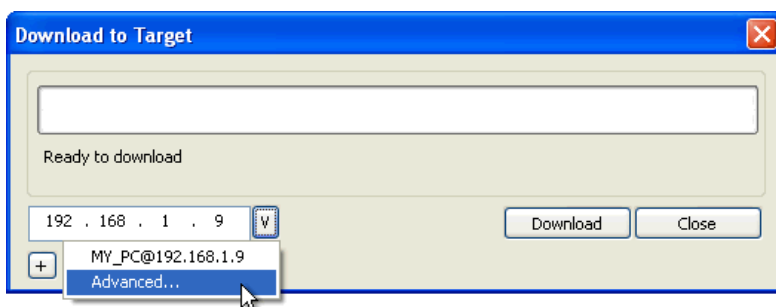
## Issues with port numbers

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

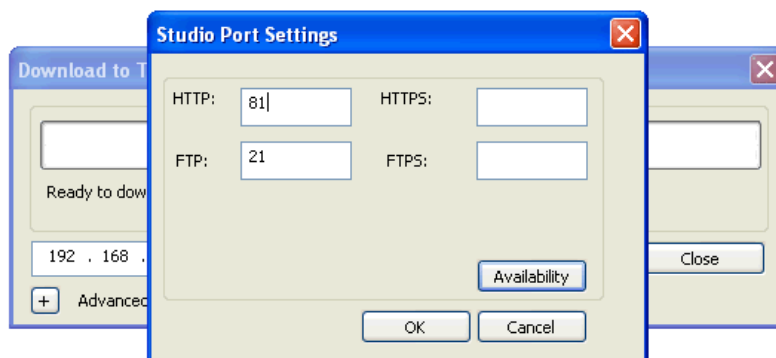


Make sure that when you change this port you also change the port used for download to HMI device in JMobile Studio.

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



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



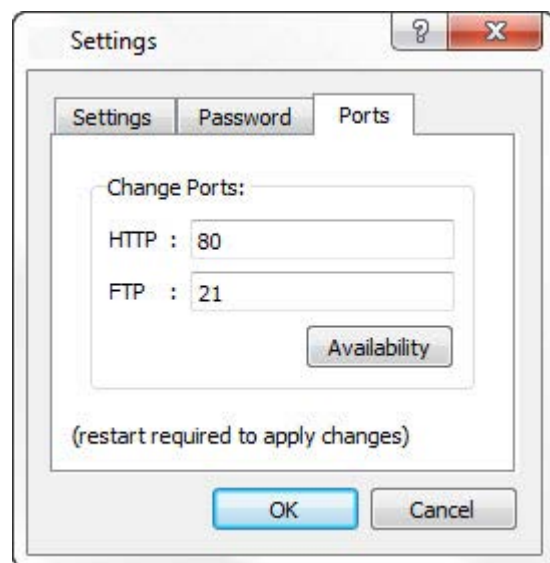
3. Click **OK** to confirm: you can now download you project to the JMobile PC Runtime.

## Restoring port information

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



## Bypassing firewall or antivirus blocks

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

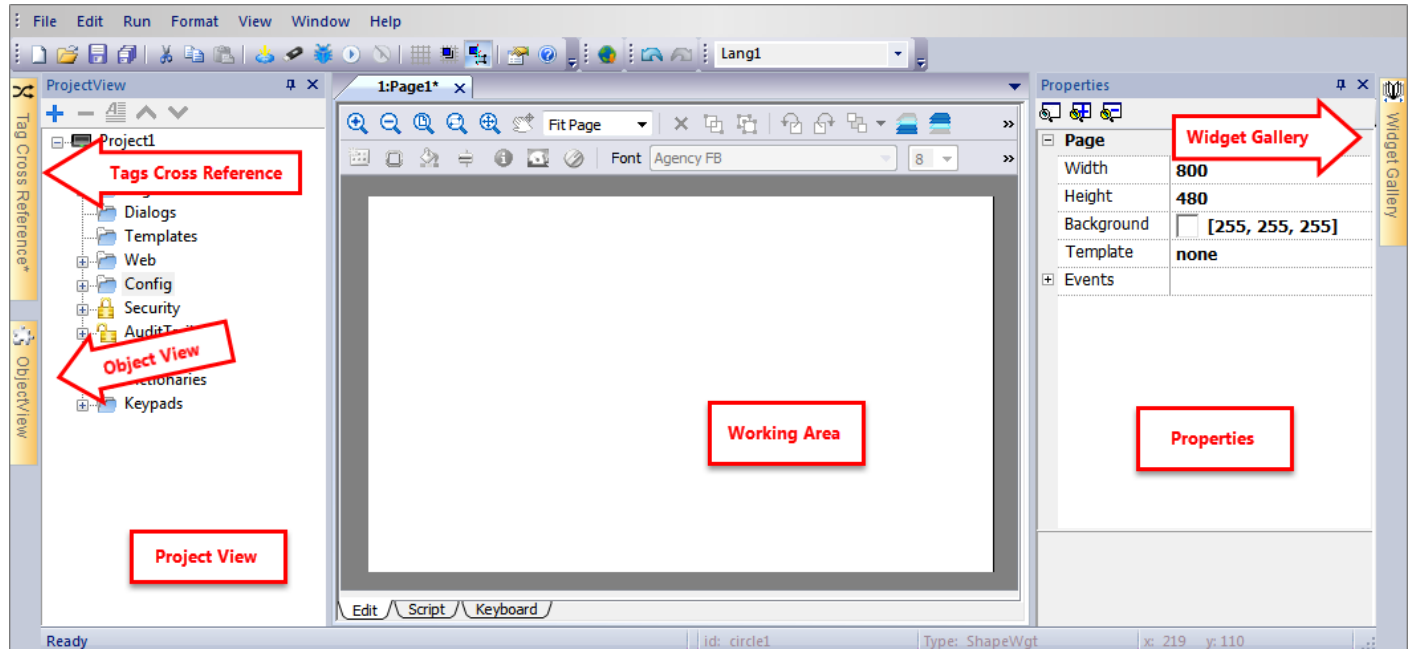
---

<b>The workspace .....</b>	<b>22</b>
<b>Creating a project .....</b>	<b>22</b>
<b>Communication protocols .....</b>	<b>24</b>
<b>Designing a page .....</b>	<b>26</b>
<b>The Widget Gallery .....</b>	<b>27</b>
<b>Data field widget .....</b>	<b>29</b>
<b>Adding tags .....</b>	<b>32</b>
<b>Attaching widget to tags .....</b>	<b>34</b>
<b>Dialog pages .....</b>	<b>36</b>
<b>Exporting tags .....</b>	<b>37</b>
<b>Importing tags .....</b>	<b>37</b>
<b>Tag find and rename .....</b>	<b>40</b>
<b>Tag find and replace .....</b>	<b>42</b>

# The workspace

## Workspace areas

JMobile Studio workspace is divided into the following main areas:



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



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. Click **Finish** to complete the Wizard.

## Portrait rotation exceptions

The following elements are not rotated in portrait mode.

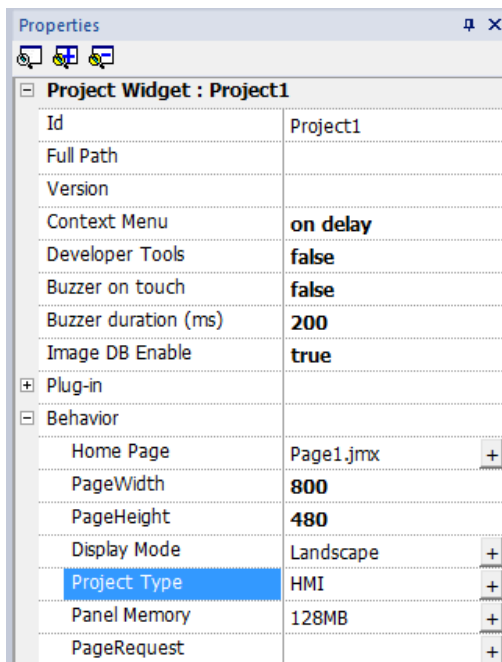
Element	Description
Operating system dialogs	System settings and system dialog
ContextMenu and related dialogs	Project Manager, About, Settings, Logging, Backup
Video	IPCamera, MediaPlayer
JavaScript	Alert and Print function
Dialog pages	"Title" of dialog pages
Scheduler	Dialogs for data entry
Macro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



HMI devices based on Linux platform can be rotated from the BSP (see "*Displays*" tab from the "*System Settings*" on page 499 "*Linux Devices*" on page 500 page) 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.



## Copying, moving, renaming a project

JMobile Studio 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.

## Communication protocols

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

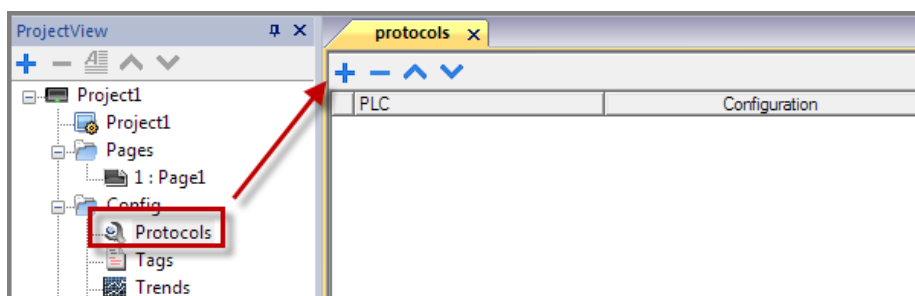
Device communication drivers are configured in the **Protocol Editor**. You can add up to the maximum number of protocols as specified in Table of functions and limits. Variable and System Variables are not counted as protocols.



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

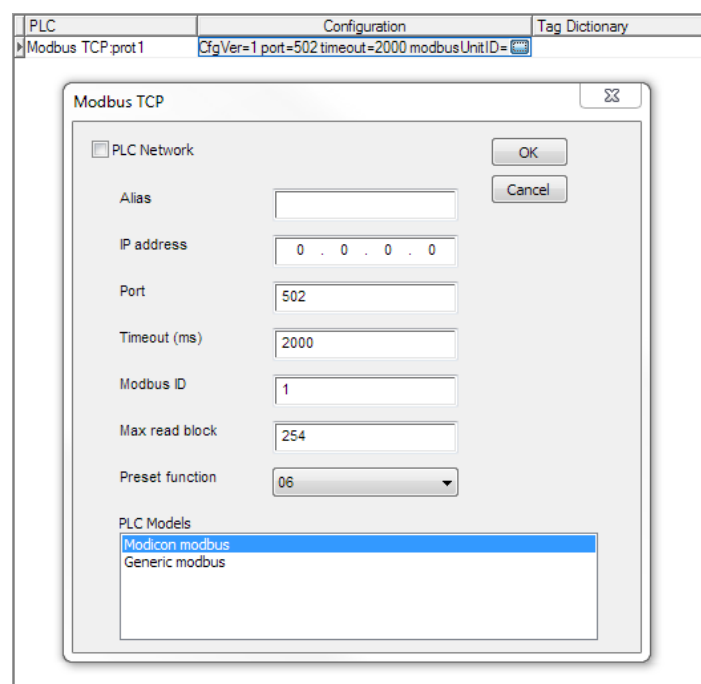
1. Click +.



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.



## Protocol parameters

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

Parameter	Description
<b>Dictionaries</b>	Tags imported for the protocol. See " <a href="#">Importing tags</a> " on page 37 for details.
<b>Enable Offline AlgorithmOffline Retry Timeout</b>	See " <a href="#">Automatic offline node detection</a> " on page 277 for details.

Parameter	Description
Version	Protocol version available in JMobile Studio for selected HMI device.

## Designing a page

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

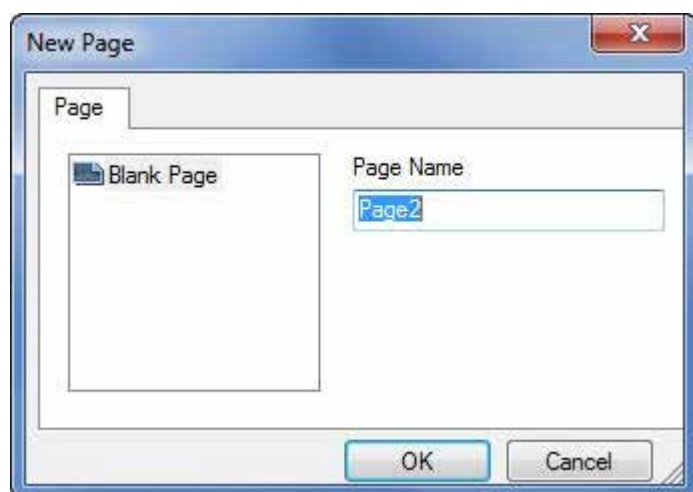
When a project is created, the first page is automatically added and shown in the **Page Editor**.

### 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.



### Importing a page

When importing a page JMobile Studio 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 run time. 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.

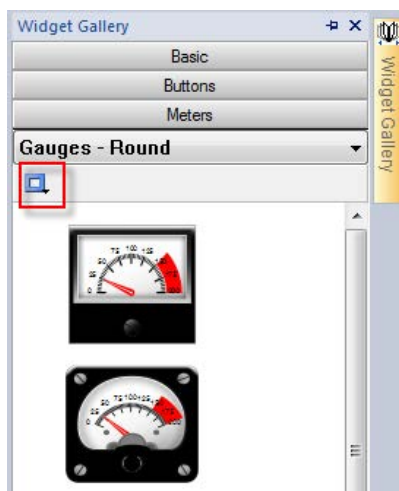
## Changing the appearance of a widget

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.





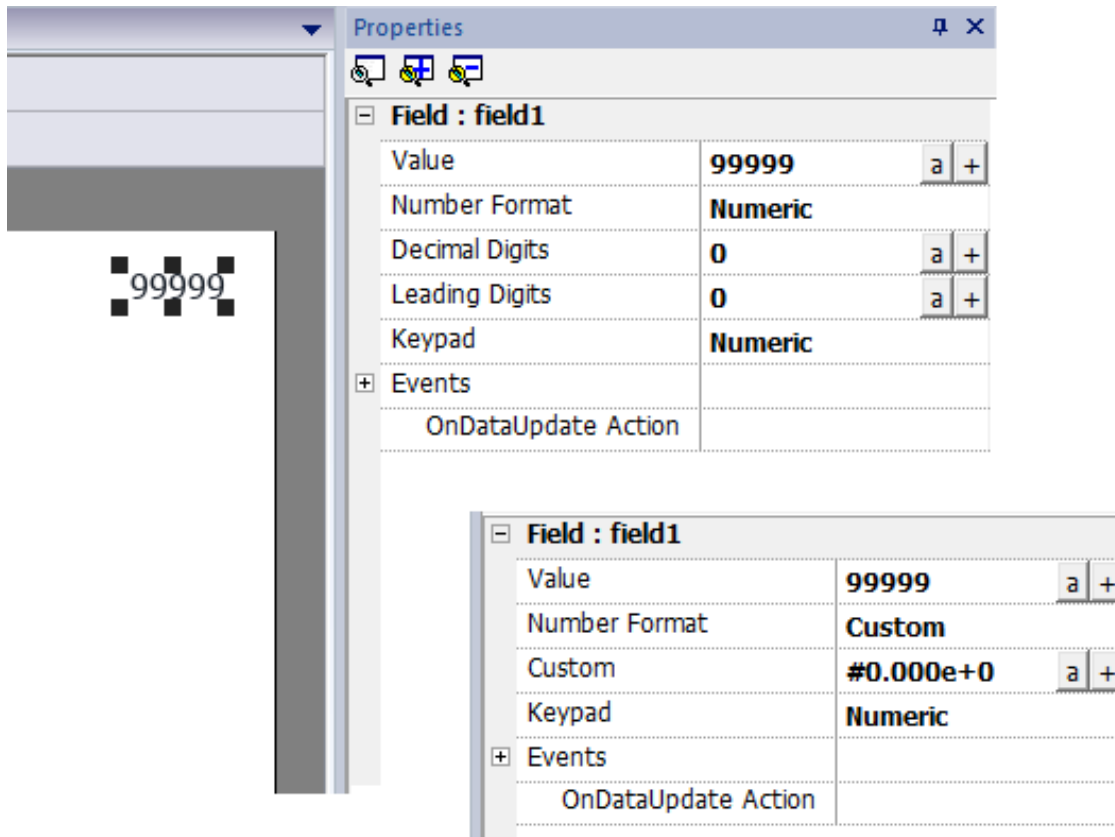
## Data field widget

The most common widget is the data field widget that give the possibility to display value of tags.

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



## Field properties



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

Property	Description
<b>Decimal Digits</b>	Number of decimal digits to show (not available on all format types)
<b>Leading Digits</b>	Number of leading digits to show (not available on all format types)
<b>Custom</b>	String that define the number format to use (available only when selected Number Format is CUSTOM)
<b>Keypad</b>	Keypad type to pop up to edit the tag value <ul style="list-style-type: none"> <li>• None</li> <li>• Alphabetic, Numeric, Etc. Pop up a predefined keypad or a user keypad (see <a href="#">"Keypads" on page 343</a>)</li> <li>• Wheel Keypad will not be displayed . Wheel can be used to increment/decrement the numeric value</li> <li>• Macro Keypad will not be displayed . Keyboard macro can be used to enter keys (see <a href="#">"Keyboard actions" on page 165</a>)</li> </ul>
<b>Events</b>	
<b>OnDataUpdate Action</b>	Commands list to execute any time the tag value changes (See <a href="#">"Actions" on page 159</a> for the available commands)

Some properties have a couple of buttons:



Enter edit mode:  
you can directly type the tag name to use



Attach to tag:  
the dialog where select the tag to use will be opened

## Custom Formats

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

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

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

## Examples

To display	As	Place Holder
123	0123	000#

To display	As	Place Holder
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

## Adding tags

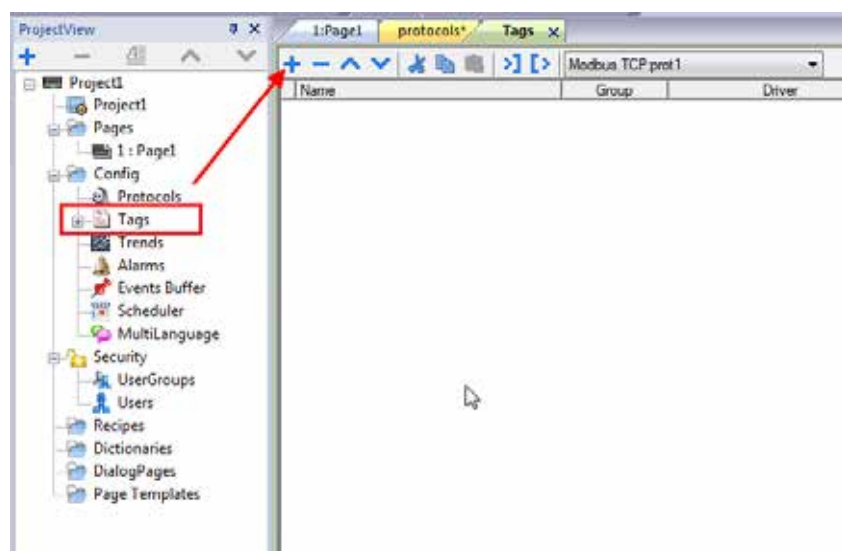
JMobile Studio uses tag names to access all device data. All fields and reference locations in the device need to be assigned a tag name to be used in the HMI project.

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 46](#) for details.

## Tag editor

Path: **ProjectView > Tags**




## Adding a tag


1. Click **+** and enter the required data.
2. Select the Address from the communication protocol address dialog: new tags are named Tag1, Tag2, ....
3. Click on the tag name to rename it.



## Tag properties

See specific protocol documentation for details.

Property	Description
<b>Name</b>	<p>Unique tag name at project level. Primary key to identify information in the runtime tag database.</p> <p> <b>WARNING: Duplicate tag names are not allowed.</b></p>
<b>Groups</b>	Group names associated to a tag
<b>Driver</b>	Communication protocol
<b>Address</b>	<p>Controller memory address.</p> <p>To edit click on the right side of the column to get the dialog box where you can enter the address information.</p>
<b>Encoding</b>	Encoding type for string data type (UTF-8, Latin1, UTF-2 and UTF-16)
<b>Comment</b>	Tag description
<b>Simulator</b>	Tag behavior during simulation. Several profiles are available.
<b>Scaling</b>	<p>Conversion applied to tag before database storage.</p> <p><b>By formula</b> = defined as a linear transformation.</p> <p><b>By range</b> = defined as a range conversion.</p>

The below properties will be visible only after select the “Show Advance Columns” mode from the tag editor toolbar..

Property	Description
<b>PLC Tag Name</b>	<p>Original PLC tag name, used to match tags used by HMI application (Tag Name) and tags exported from PLC</p> <p>R/W only in advanced view to allow for adjustments in case tag import errors.</p>
<b>Rate (ms)</b>	<p>Tag refresh time. Default: 500ms.</p> <p>When the refresh rate is set to “Manual”, the HMI device will not read the tag from the remote device automatically in background. Tag is read and refreshed into the database only by explicitly required from the “ForceReadTag” action or using the forceRefresh option into the JavaScript getTag().</p> <p> <b>WARNING: Tags refresh rate is the maximum refresh rate. Actual refresh rate depends on: communication type (serial, fieldbus, Ethernet), protocol, amount of data exchanged.</b></p>

Property	Description
R/W	<p>R/W tag attribute (R/W, R or W).</p> <p> Note: The content of Write Only tags is always written and never read. When communication is not active, the content of these tags may not be available in widgets.</p>
Active	<p>Update mode.</p> <p><b>false</b> = tags are read from controller only when required by the HMI device.</p> <p><b>true</b> = tags are continuously read even if not required by the displayed page.</p> <p> <b>Important: Leave this value set to false for higher communication performance.</b></p>

## 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.

## 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.

# Attaching widget to tags

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

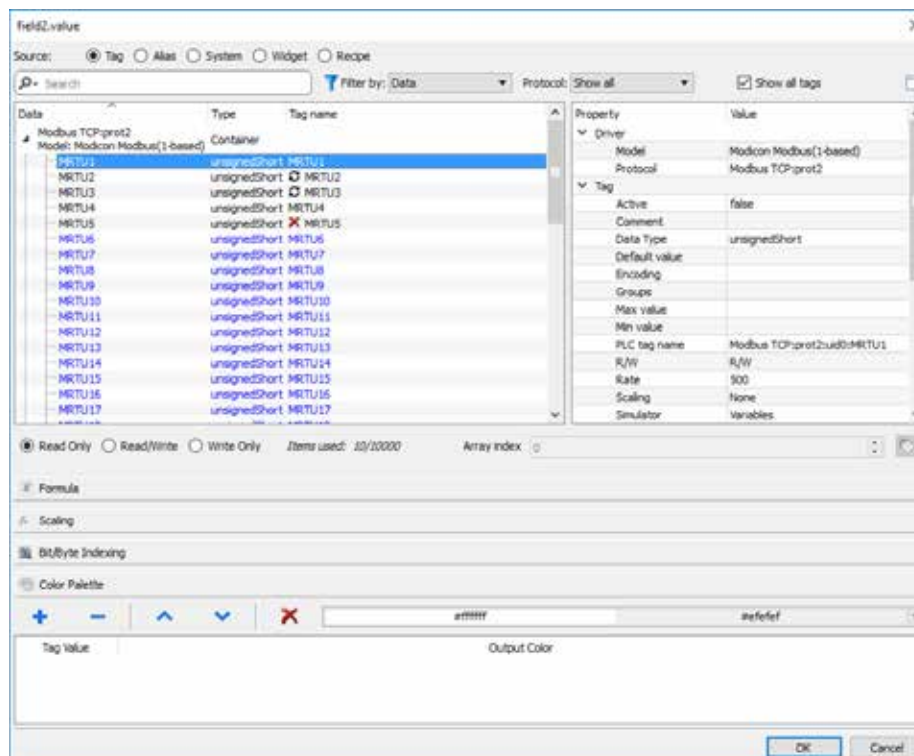
## Data sources

Elements to which an object property can be attached:

Data source	Description
Tag	Tag defined in the Tag Editor
Alias	Indexed tag address
System	Predefined system tags (see <a href="#">"System Variables (Attach To)" on page 97</a> )
Widget	Connect to a widget property (for example, value of a slider widget)
Recipe	Data from the Recipe Manager (see <a href="#">"Recipes" on page 243</a> )

## Attaching a property to a tag

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



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



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

See [""Attach to" parameters" on page 46](#) 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)

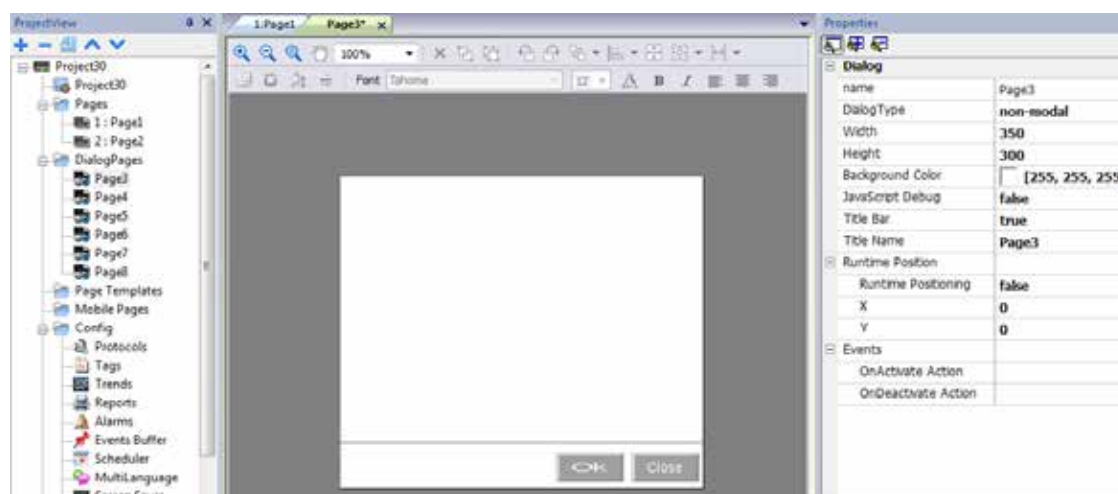
## Dialog pages

Path: **ProjectView**> **Dialogs**

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

### Main dialog properties

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



### Maximum number of dialogs

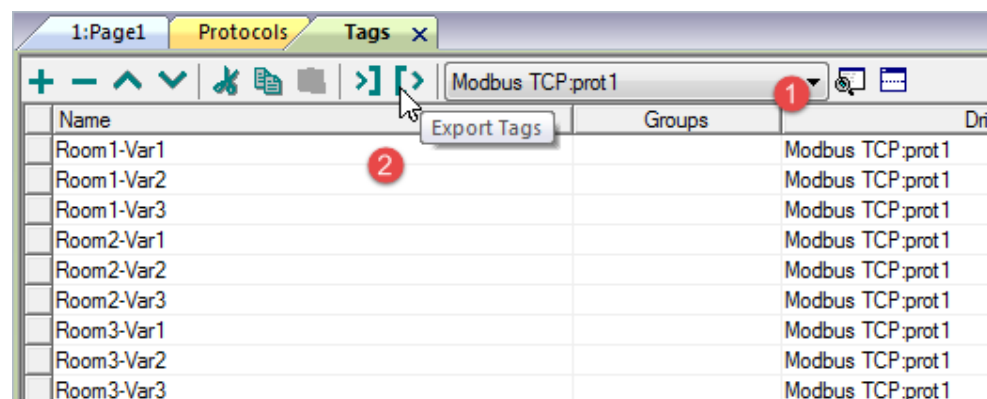
Maximum number of open dialogs is defined in ["Functional specifications and compatibility"](#) on page 565.

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



## Exporting tags

Path: **ProjectView > Tags**



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

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

## Importing tags

### Introduction

Some protocols allow you to import tags stored in a comma separated file (.csv or other formats).

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 cause communication errors. See ["Limitations in Unicode support" on page 286](#)**



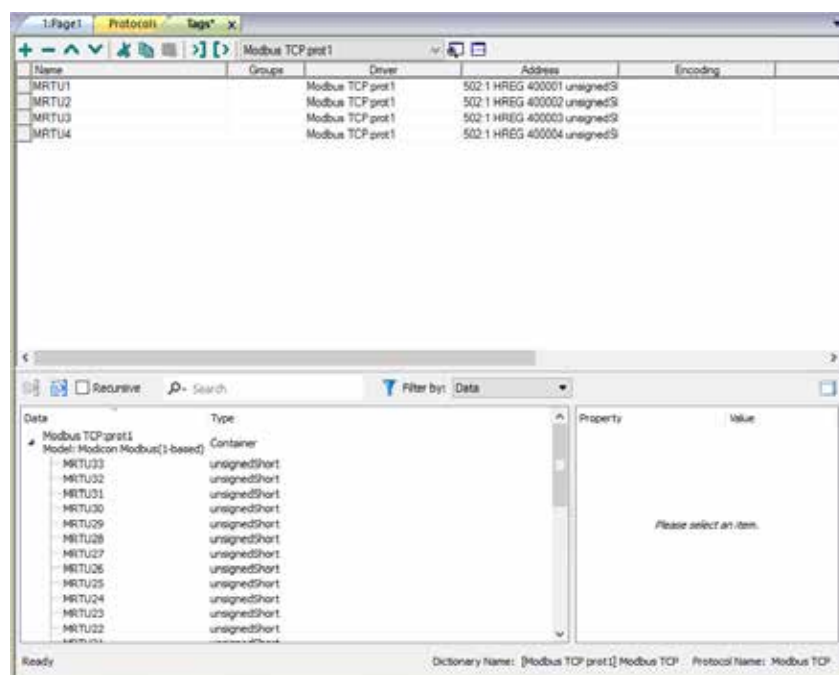
Note: When importing tags, character "." in tag names is replaced with "/" . The protocol will use the correct syntax when communicating to the PLC.

### Dictionaries

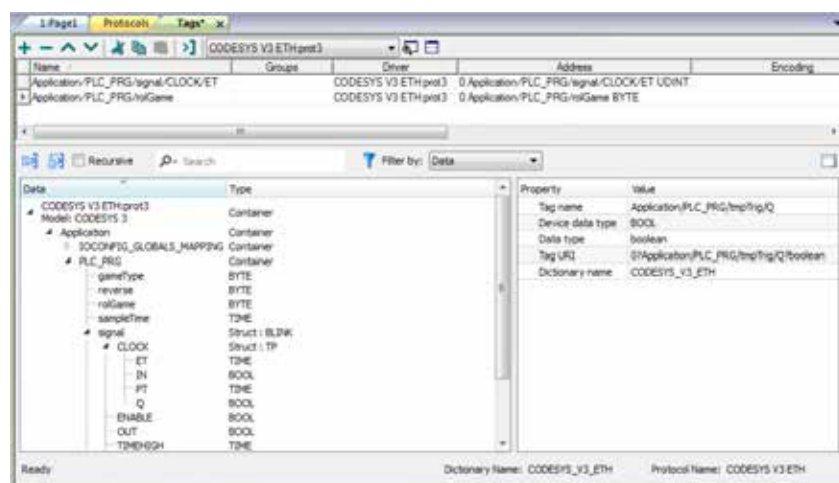
Path: **ProjectView > Dictionaries**

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.

## Linear view



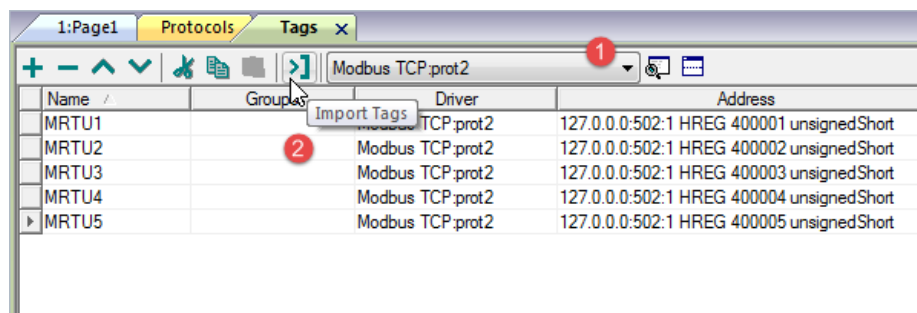
## Hierarchical view



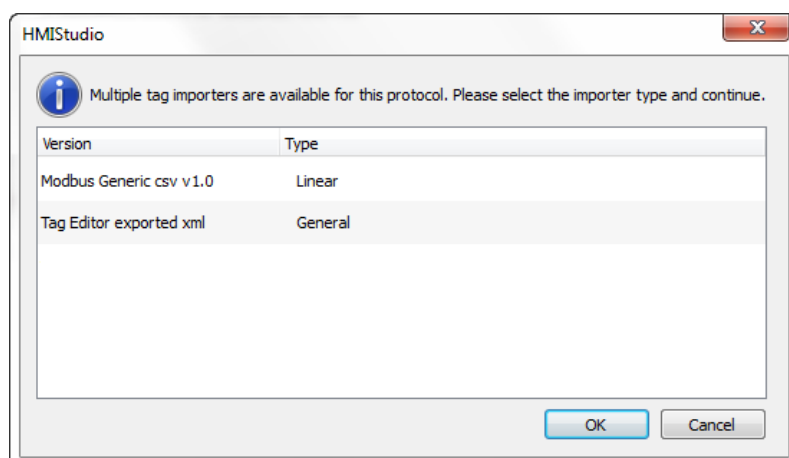
## Importing tags

To import tags from an external file:

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



2. Click the **Import Tags** button: the select file dialog appears. A dialog to choose the importer type appears.



3. Select the file: a list of tags is shown in a linear or hierarchical view.
4. To import tags, select one or more tags or a node (hierarchical view only) and click the **Import tag** button: tags are copied to the project and listed in the upper window section.

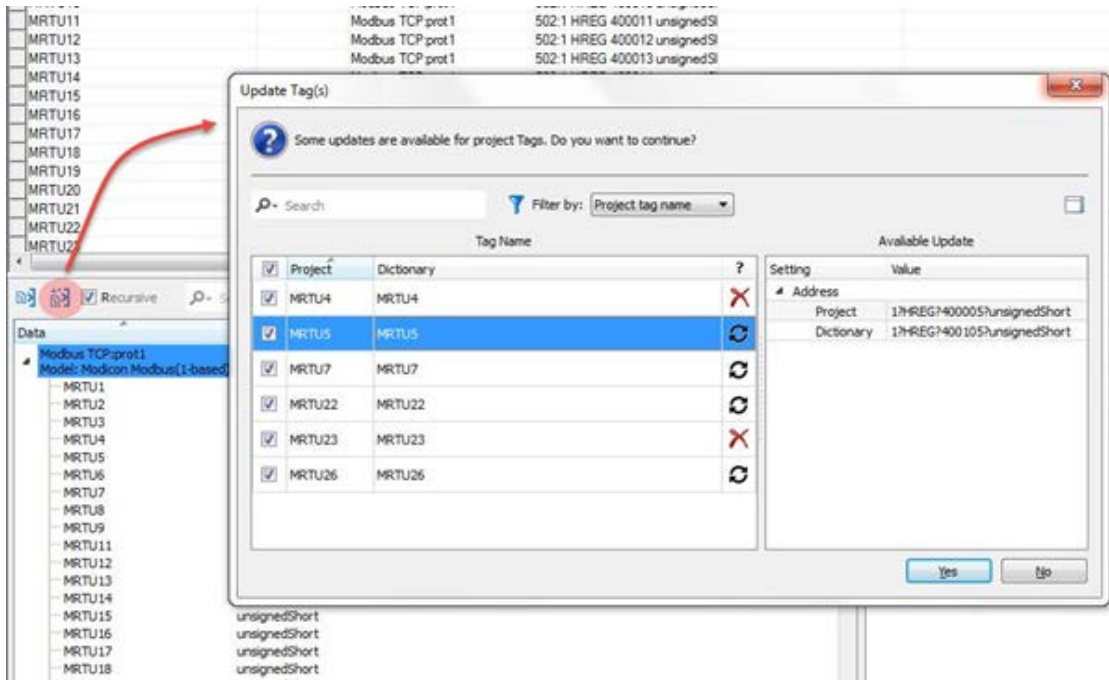
Parameter	Description
<b>Recursive</b>	All elements of the structure are imported into separate tags.



Note: 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.

## Updating the imported tags

Using the Update Tag(s) command you can re-import tags. A dialog allows you to select the tags to be reimported:



These tags need to be updated. A list of differences between project and dictionary is displayed.

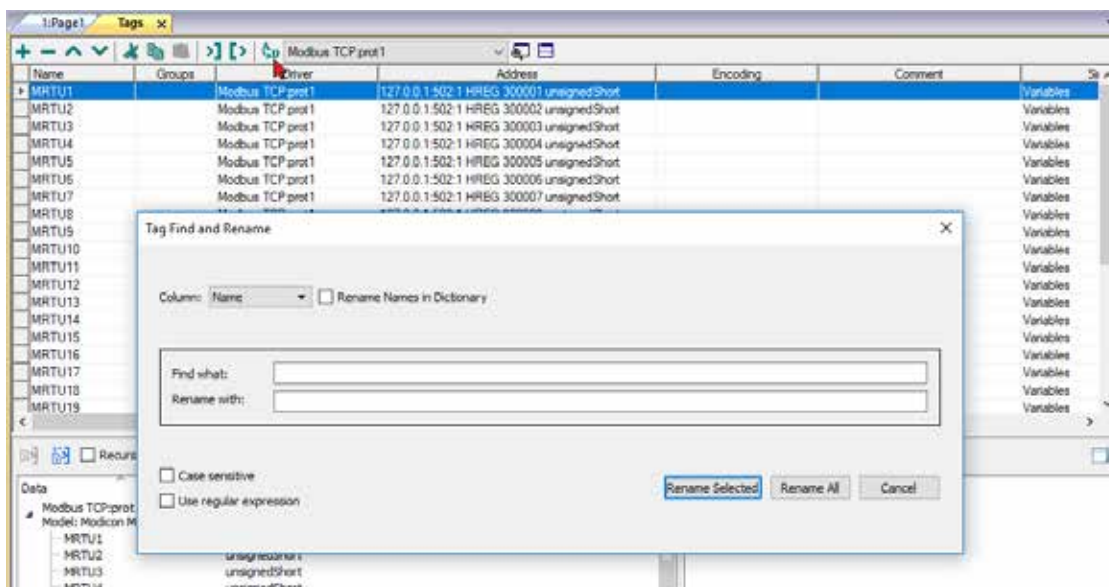




These tags are no longer available in the dictionary. If updated, these tags will be removed from the project.

## Tag find and rename

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

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



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

## 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**

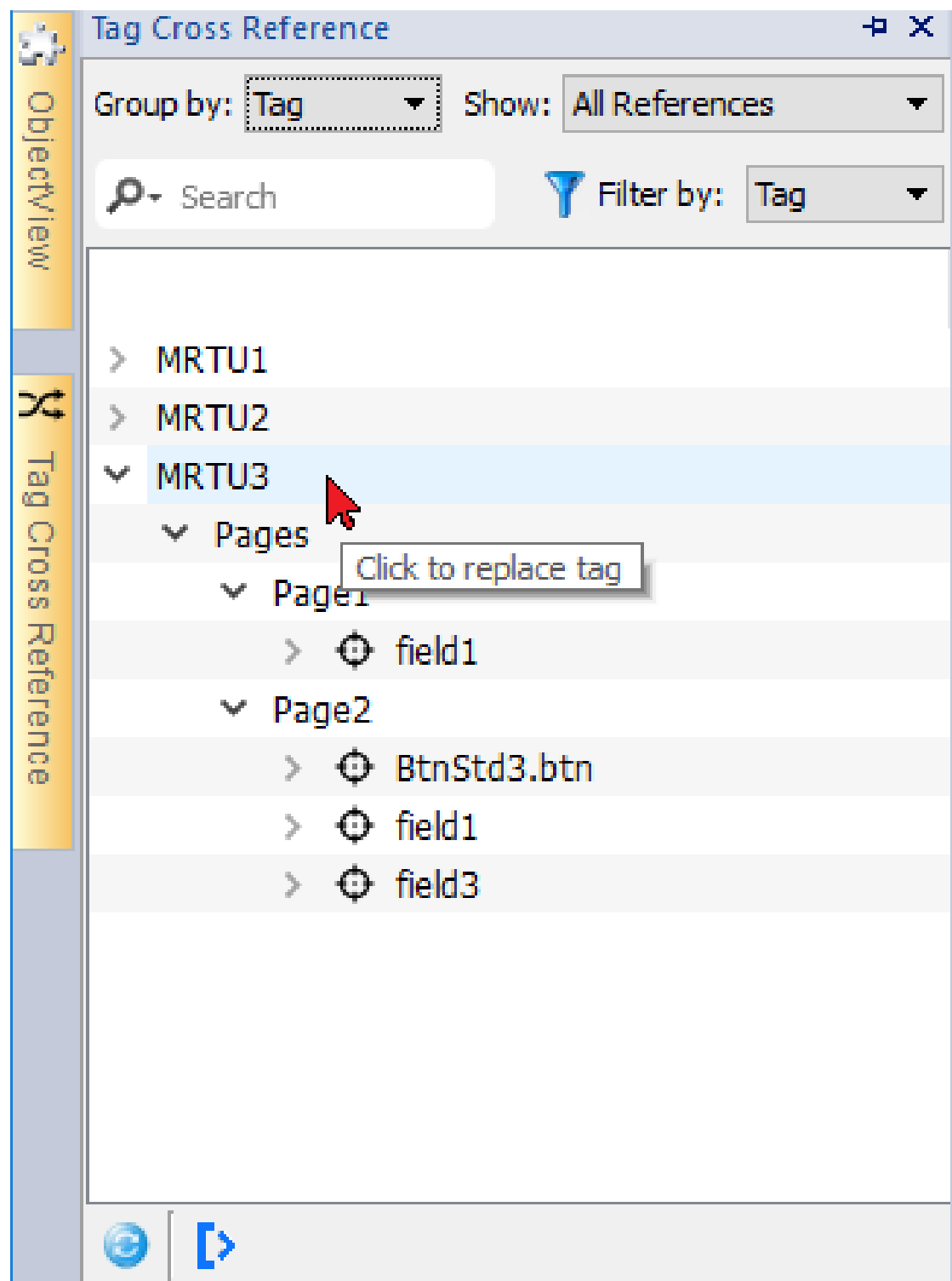
Where

- (.\*)** 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

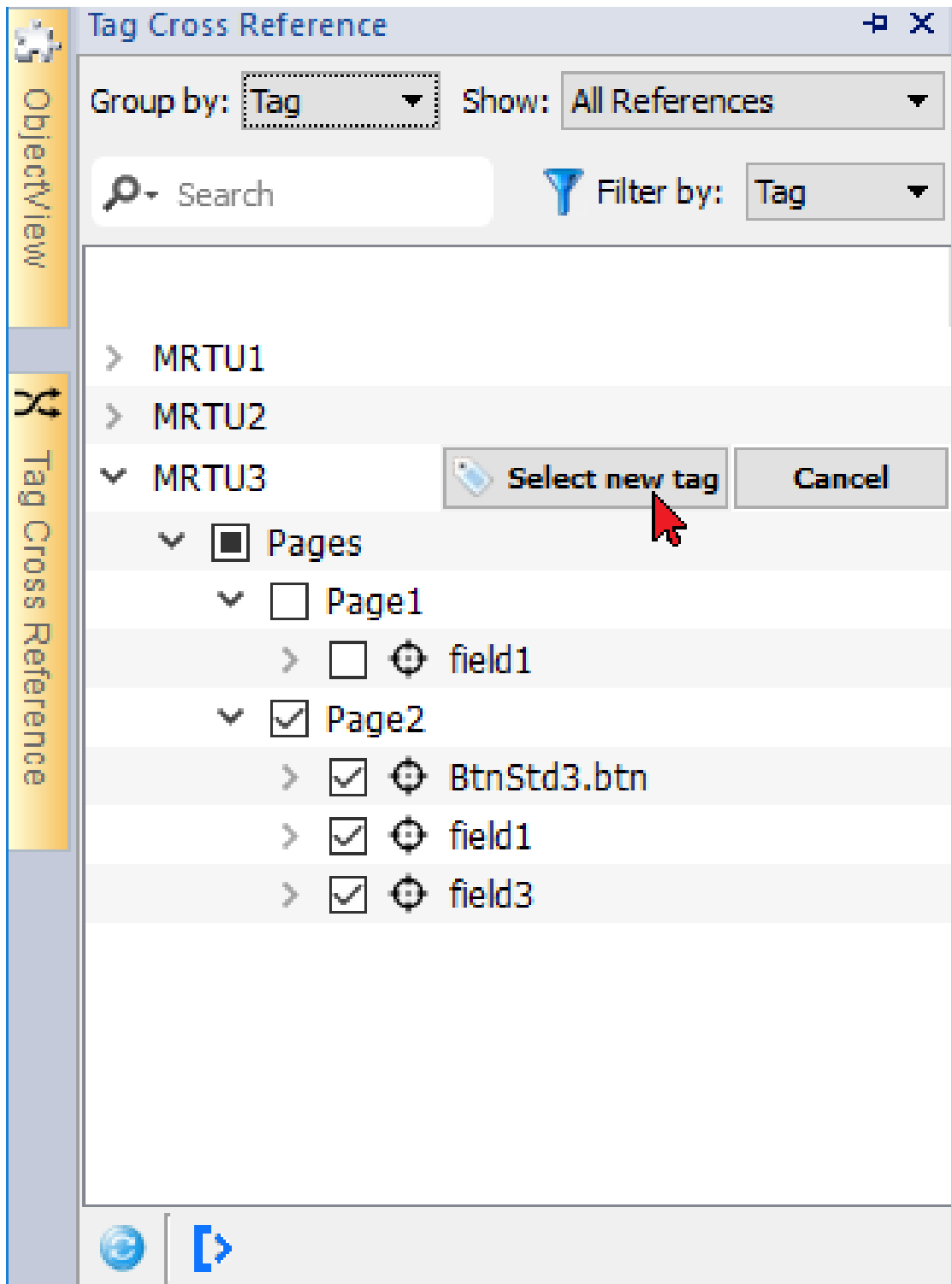
Using this feature you can search all occurrence of a tag inside the project and replace it with another tag.

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



## 5 Programming concepts

---

Programming for JMobile Studio is based on a few basic concepts and behaviors.

---

<b>Data types</b> .....	<b>46</b>
<b>"Attach to" parameters</b> .....	<b>46</b>
<b>Formula</b> .....	<b>52</b>
<b>Events</b> .....	<b>56</b>
<b>Widgets positioning</b> .....	<b>59</b>
<b>Managing overlapping widgets</b> .....	<b>60</b>
<b>Grouping widgets</b> .....	<b>61</b>
<b>Changing multiple widgets properties</b> .....	<b>68</b>
<b>Changing fill color property according to tag values</b> .....	<b>69</b>

# Data types

When creating a tag you have to specify its properties. Data type are specific to JMobile Studio, 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	Description
<b>boolean</b>	One bit data (0..1)
<b>byte</b>	Signed 8 bit data (-128..127)
<b>double</b>	IEEE double-precision 64-bit floating point type ( $\pm 2.2\text{e-}308$ ... $\pm 1.79\text{e}308$ )
<b>float</b>	IEEE single-precision 32-bit floating point type ( $\pm 1.17\text{e-}38$ ... $\pm 3.40\text{e}38$ )
<b>int</b>	Signed 32 bit data (-2.1e9 ... 2.1e9)
<b>short</b>	Signed 16 bits data (-32768..32767)
<b>string</b>	Characters coded according to selected format
<b>time</b>	Time data
<b>unsignedByte</b>	Unsigned 8 bit data (0..255)
<b>unsignedInt</b>	Unsigned 32 bit data (0 ... 4.2e9)
<b>unsignedShort</b>	Unsigned 16 bit data (0..65535)
<b>uint64</b>	Unsigned 64 bit data (0...264 – 1)

## "Attach to" parameters

### Object properties

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

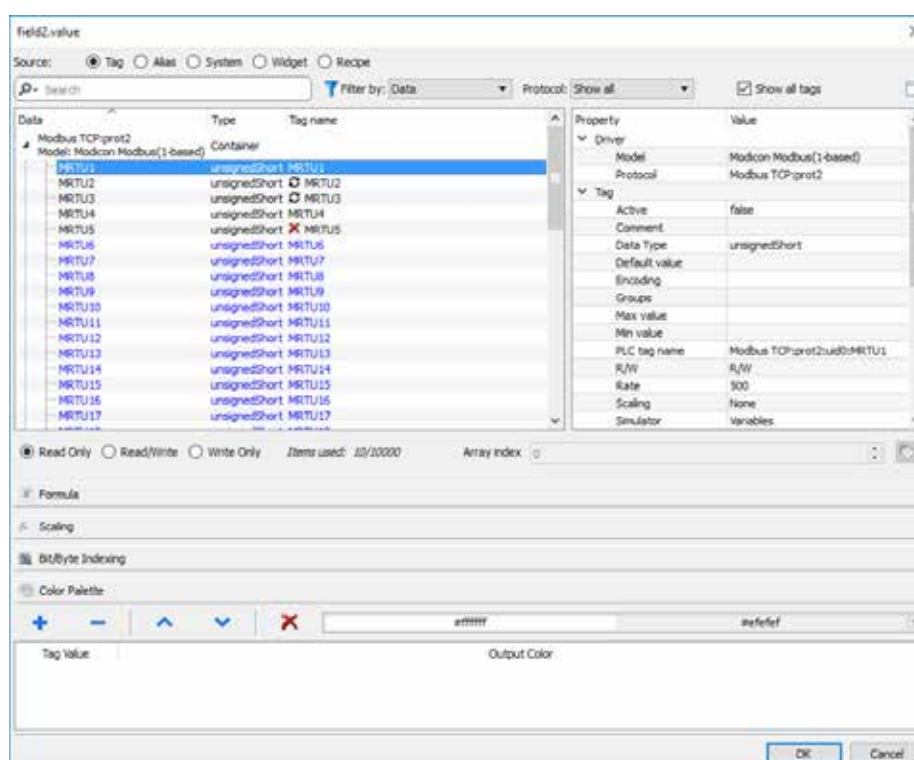


The page toolbar shows only the most common object properties, while the property pane shows all the properties in a basic or advanced view.

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

## Attaching a property to a tag

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



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

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

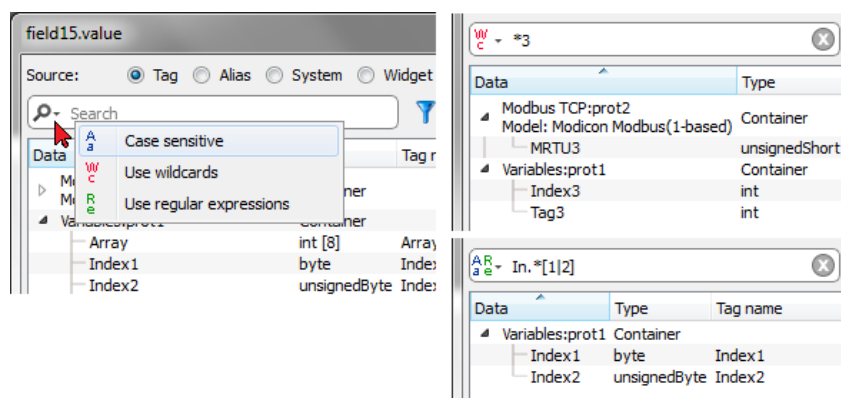
## Data sources

Elements to which an object property can be attached:

Data source	Description
Tag	Tag defined in the Tag Editor
Alias	Indexed tag address
System	Predefined system tags (see " <a href="#">System Variables (Attach To)</a> " on page 97)
Widget	Connect to a widget property (for example, value of a slider widget)
Recipe	Data from the Recipe Manager (see " <a href="#">Recipes</a> " on page 243)

## Advanced search

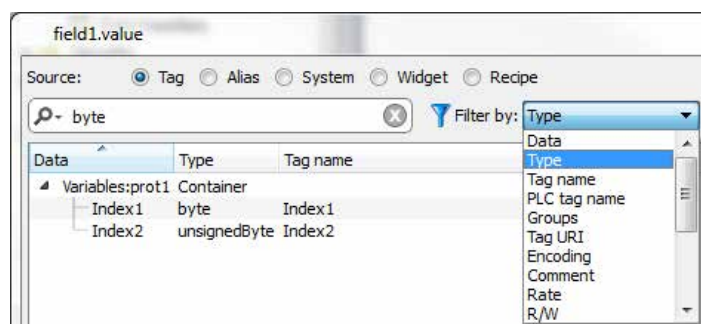
Various syntax options can be applied to search box:



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 <a href="https://en.wikipedia.org/wiki/Regular_expression">https://en.wikipedia.org/wiki/Regular_expression</a> for additional details regarding regular expressions.

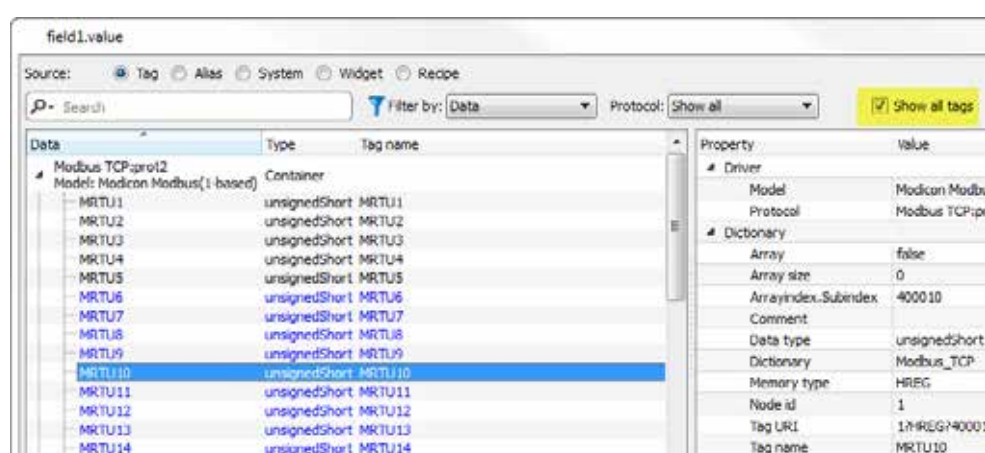
## Filtering tags

Choose various tag filter criteria:

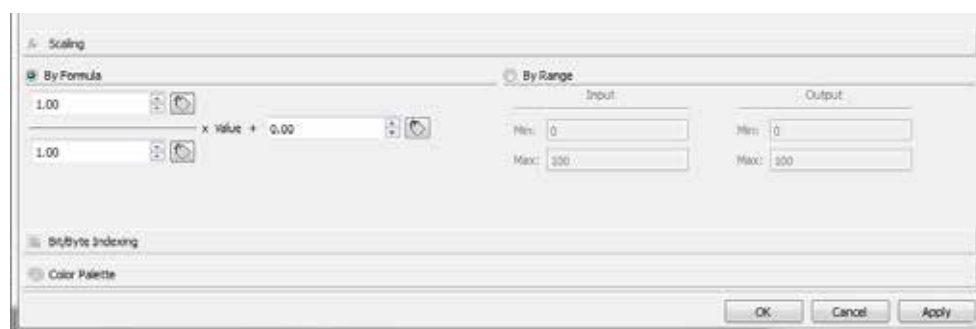


## Showing dictionary tags

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



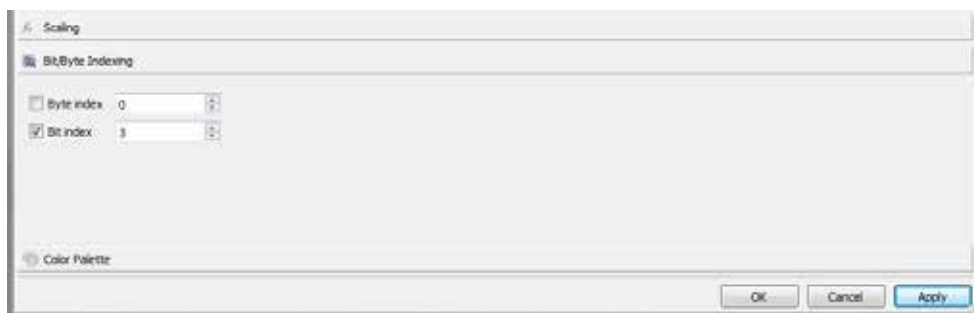
## Converting tag value



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

## Extract tag bit/byte based on index

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

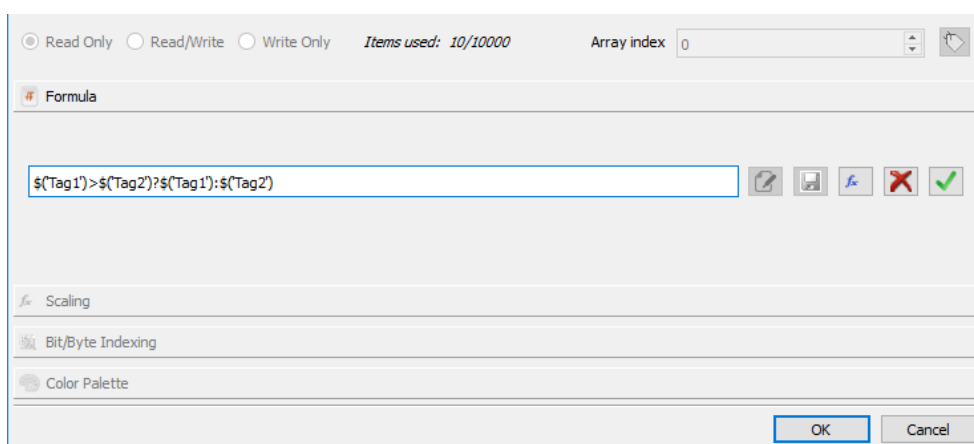


## Use a formula to calculate the value to use

Allow to use a formula to calculate the value to use. See ["Formula" on page 52](#) chapter for additional details.

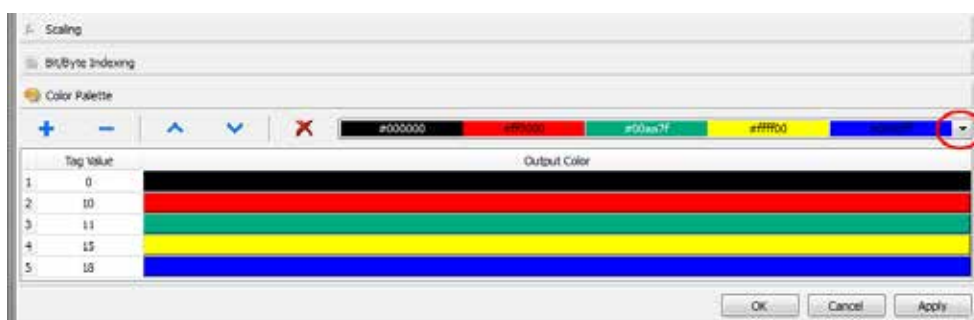


Note that using a formula the datalink will be ReadOnly



## Mapping tag values to color

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



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

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

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



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



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

## Datalink Serialization

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

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



The image shows two screenshots of the 'Properties' pane for a field named 'field1'. The top screenshot shows the 'Value' field set to '99999' and the 'Keypad' set to 'Numeric'. The bottom screenshot shows the 'DataLink' field set to 'Tag1 | R/W | ScaleXForm(1,10,0)' and the 'Access Type' set to 'R/W'.

The data link format is:

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

on in case of formula:

= <formula>

Example:

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

## Formula

A formula is an expression made of:

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

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

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

## Commands



Enter edit mode



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



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



Removing the entered formula



Confirming the entered formula





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

Example of formulas are:

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

## Syntax for formula

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

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

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

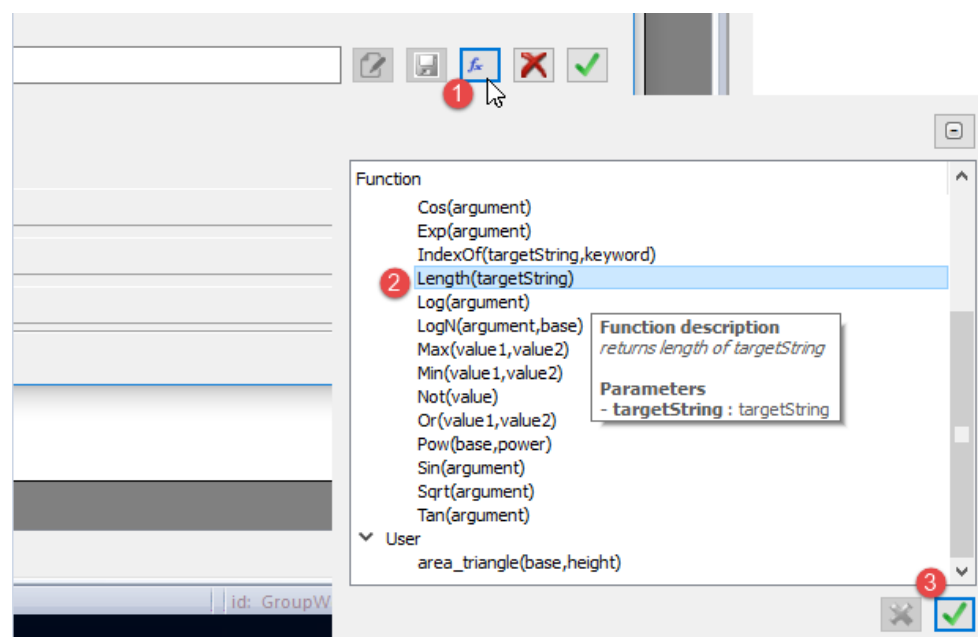
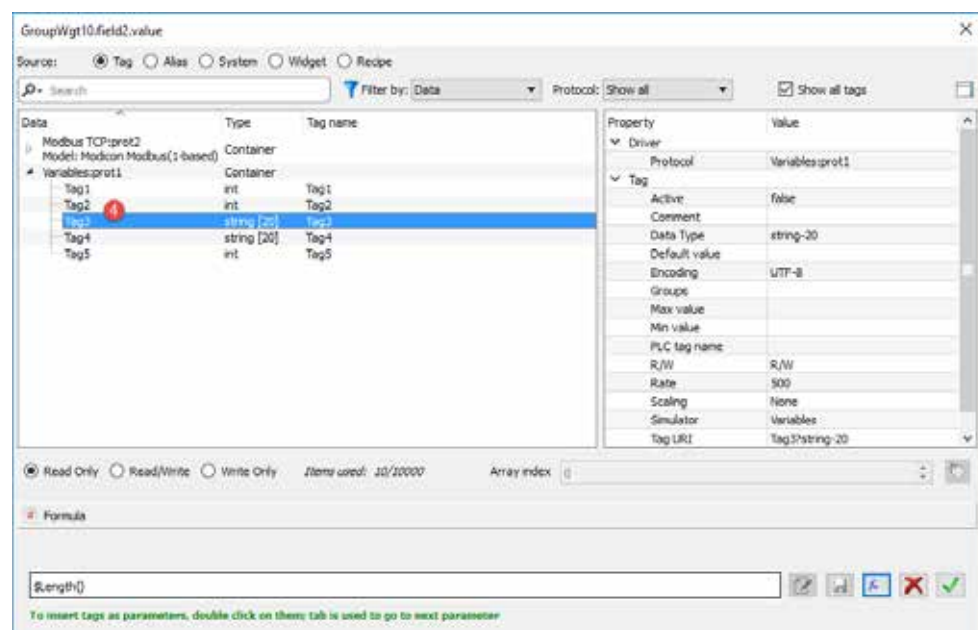
Logical Operators	
<b>&amp;&amp;</b>	AND
<b>  </b>	OR
<b>!</b>	NOT

Compare Operators	
<b>&lt;</b>	Greater than
<b>&lt;=</b>	Less than or equal to
<b>&gt;</b>	Less than
<b>&gt;=</b>	Greater than or equal to
<b>==</b>	Equal to
<b>!=</b>	Not equal to

## Use predefined formula from the library

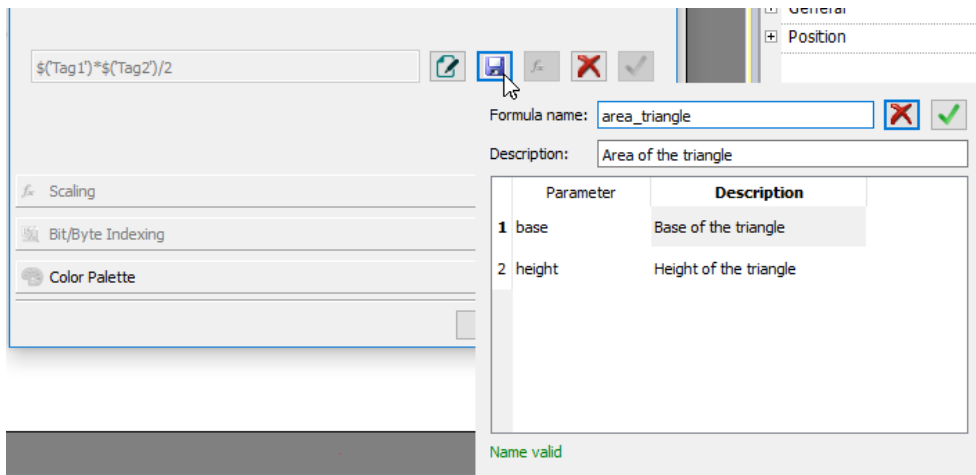
To insert a project or a predefined formula:

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

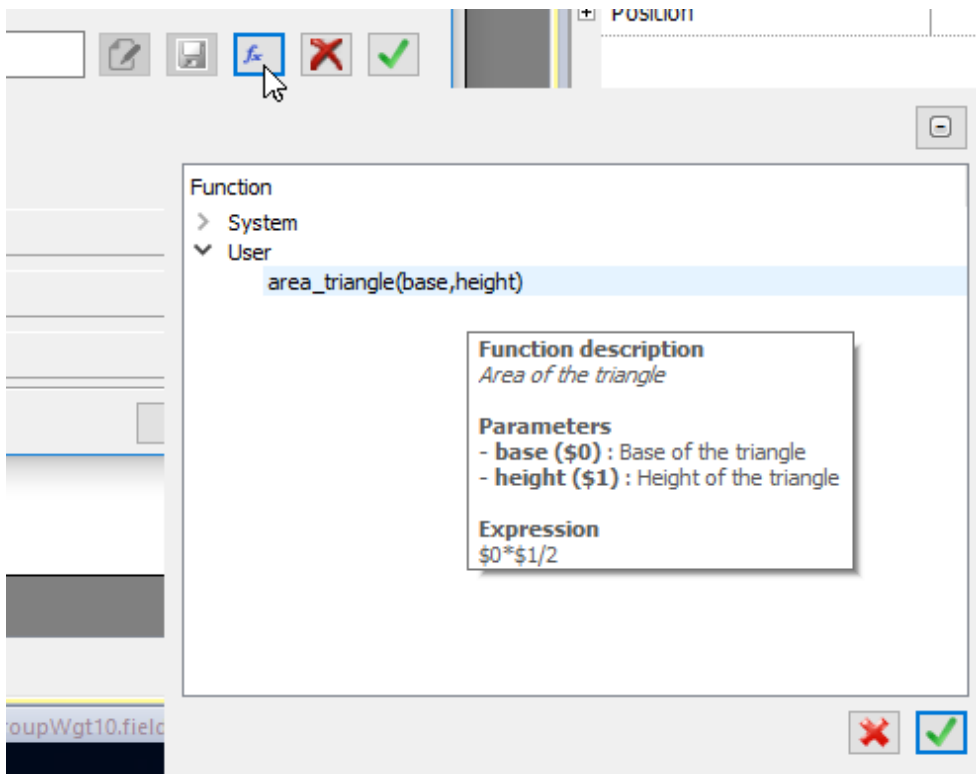


## Add user formulas into the library

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



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



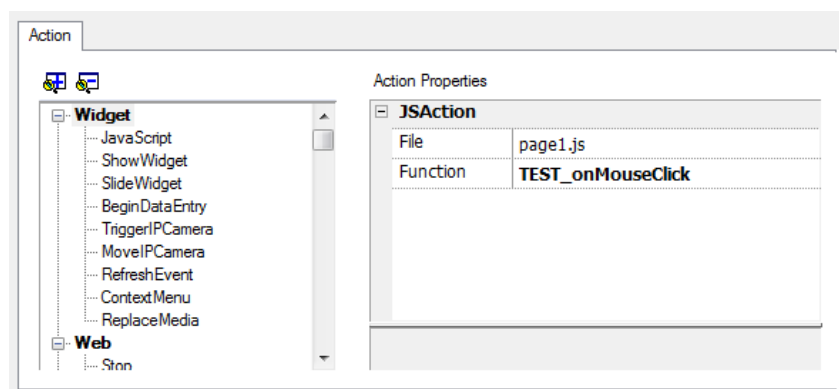
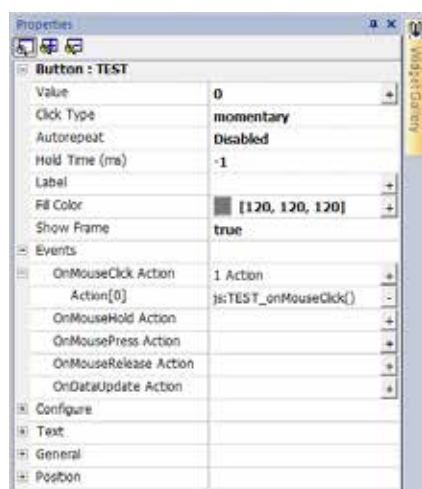
## Events

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

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

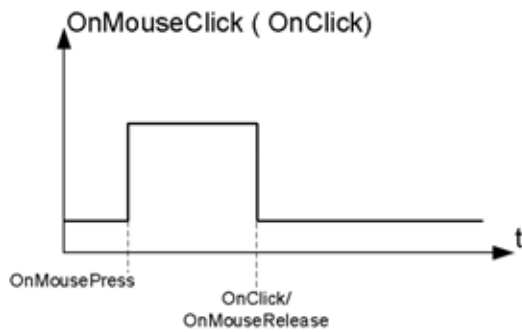
You can attach one or more actions to an event, so that they will be executed whenever the event occurs.

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



## OnClick / OnMouseClicked

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



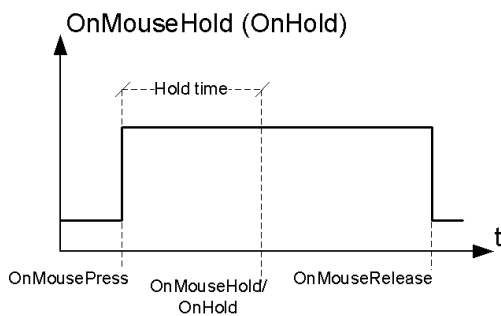
## OnHold/OnMouseHold

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 71.

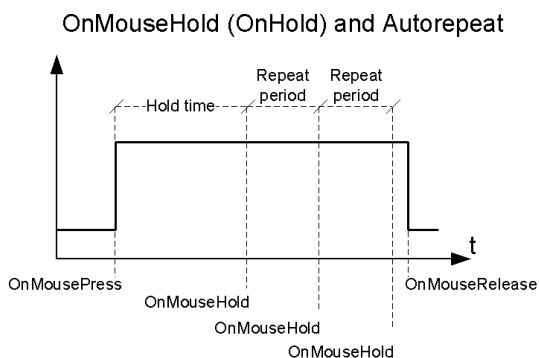


Note: If **Hold Time** is set to -1 for the widget, the project **Hold Time** value will be used.

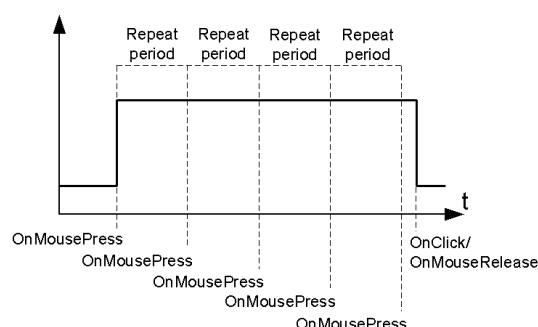


## Autorepeat

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



### OnMousePress and Autorepeat



## OnWheel

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

## OnActivate

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

## OnDataUpdate

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



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

# Widgets positioning

You can position widgets in the page using two methods:

- Snap to Grid
- Snap to Object

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

## Snap to Grid

*Path: View > Snap to Grid*

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

## Setting grid properties

*Path: View > Properties*

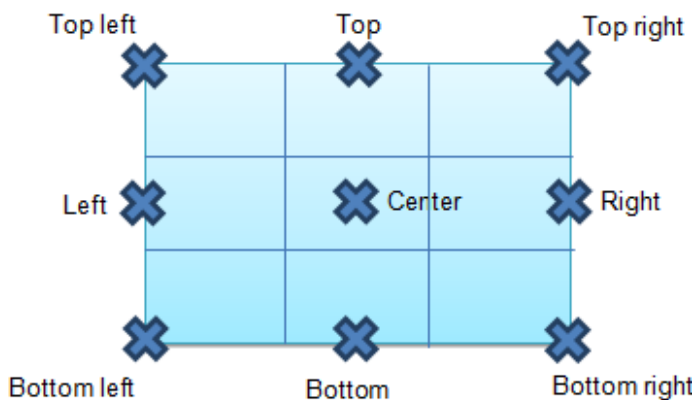
Parameter	Description
Spacing X	Space in pixel between two lines/dots on the X axis
Spacing Y	Space in pixel between two lines/dots on the Y axis
Type	Grid type (dot or line)
Color	Grid color

## Snap to Object

*Path: View > Snap to Object*

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

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

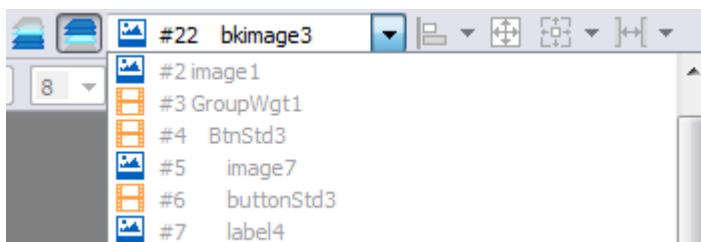


An algorithm finds a matching hot point among the near widgets hot points matching either the x or the y coordinates of the source snap point. For line widgets, the source snap points are the terminal points of the line.

## Managing overlapping widgets

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

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



**Important: Correct ordering of widgets is essential for run time 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

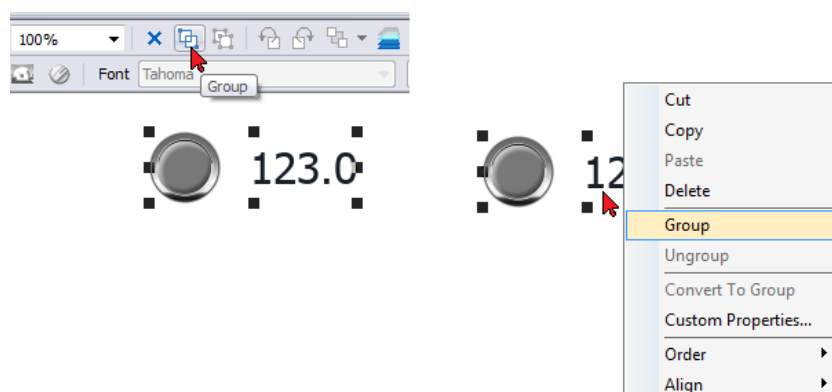
The toolbar allows to:

- 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:

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

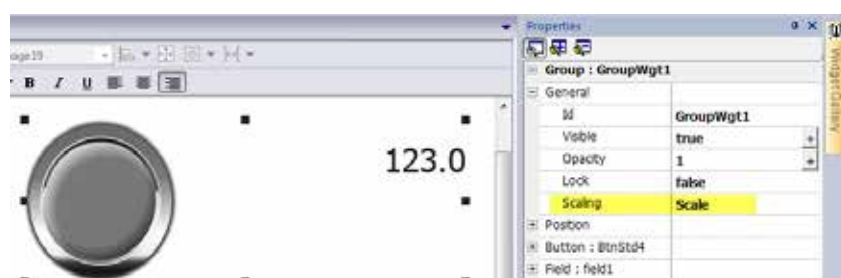


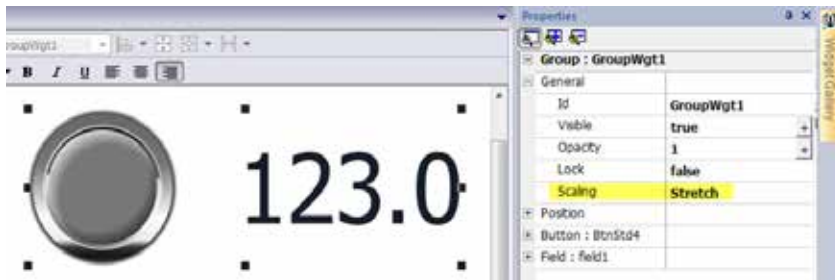
Tip: Double click to enter the group editing mode. In group mode only the group widgets are editable and selectable. All other widgets are partially hidden

## Resizing grouped widgets

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

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





## Grid Layout Group

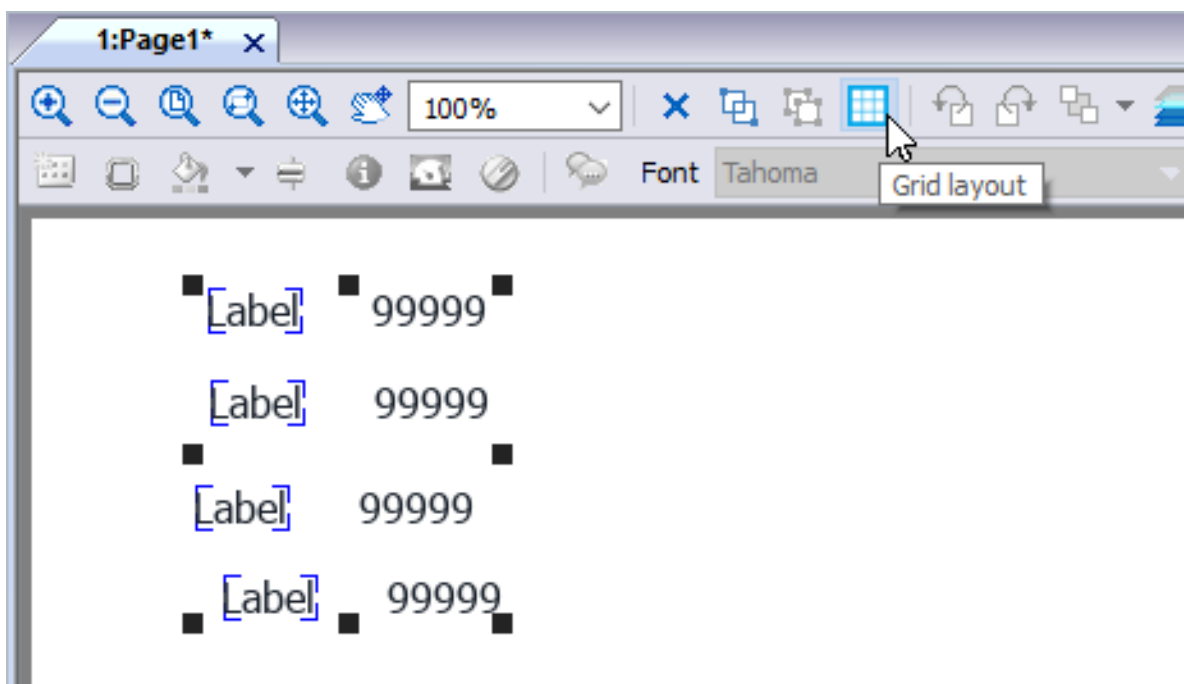
The grid layout add the possibility to configure the spatial relationships among the widgets of the group.

To create a grid layout:

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

or

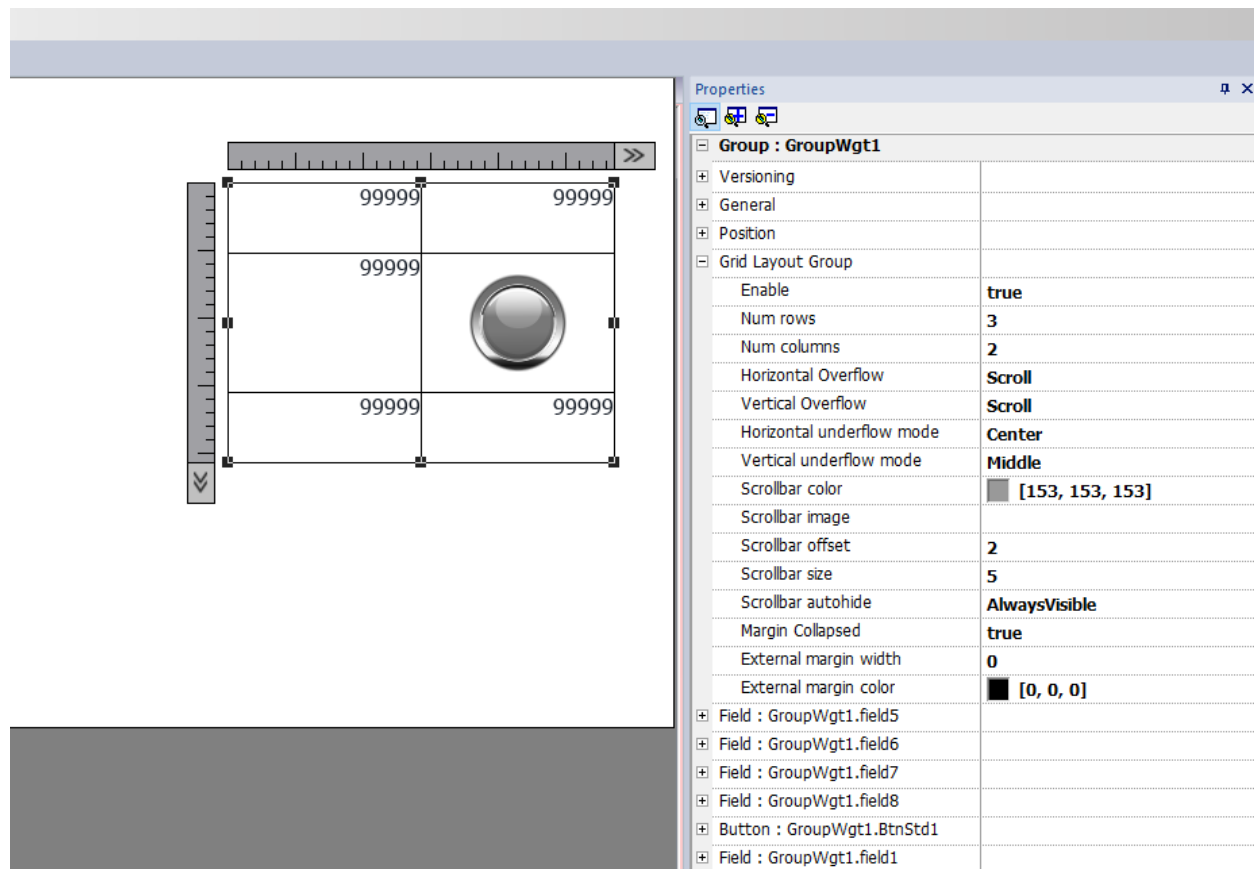
- Select the widgets that will be inside the table and click the "Grid Layout" button on page toolbar. The selected widgets will be aligned and collected inside a group with the grid layout property enabled.



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


- Grid properties
- Rows, Columns Properties
- Cells Properties

## Grid Properties



The screenshot shows a visual editor for a grid widget. The grid is 3 rows by 2 columns. The top row contains two text fields with the value '99999'. The middle row contains a button with a circular icon. The bottom row contains two text fields with the value '99999'. To the right of the grid is a 'Properties' panel for the selected widget, 'Group : GroupWgt1'.

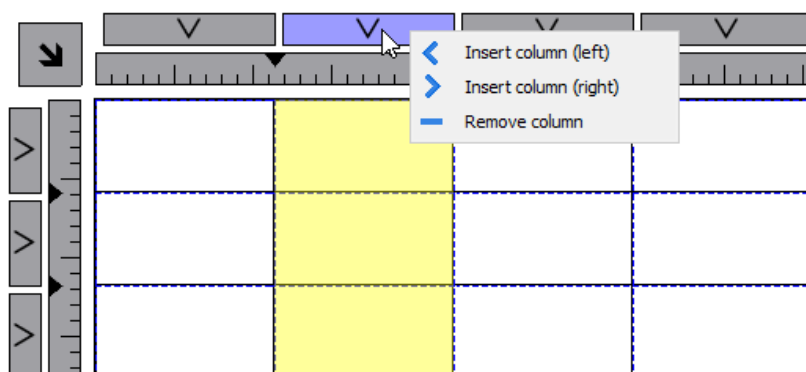
Group : GroupWgt1	
Versioning	
General	
Position	
Grid Layout Group	
Enable	true
Num rows	3
Num columns	2
Horizontal Overflow	Scroll
Vertical Overflow	Scroll
Horizontal underflow mode	Center
Vertical underflow mode	Middle
Scrollbar color	[153, 153, 153]
Scrollbar image	
Scrollbar offset	2
Scrollbar size	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
<b>Enable</b>	<p>Enable the grid layout.</p> <p>A grid will be generate around the widgets of the group</p>
<b>Num rows</b> <b>Num columns</b>	<p>Number of rows and columns of the grids.</p> <p> Rows and columns can be removed only if their cells are empty .</p>
<b>Horizontal overflow</b> <b>Vertical overflow</b>	<p>This parameter define the behavior of the grid when it is too small to contain all rows and columns.</p> <ul style="list-style-type: none"> <li>• <b>Hidden</b> Rows and columns that do not fit into the grid are not displayed</li> <li>• <b>Visible</b> The grid can not be made smaller than the minimum size required to contain all defined rows and columns</li> <li>• <b>Scroll</b> When the grid is too small to hold all the defined rows and columns, the scroll bars can be used to shift the content of the grid.</li> </ul>

Parameter	Description
<b>Horizontal underflow</b> <b>Vertical underflow</b>	This parameter defines the behavior of the grid when it is larger than the size defined for the rows and columns <ul style="list-style-type: none"> <li>Blocked The grid can not be made larger than the maximum size of rows and columns</li> <li>Left, Center, Right - Top, Middle, Bottom Defines the position of the widgets when cells are bigger than the maximum defined sizes</li> </ul>
<b>Scrollbar color</b> <b>Scrollbar image</b> <b>Scrollbar offset</b> <b>Scrollbar size</b> <b>Scrollbar autohide</b>	Parameters to define look and position of the scroll bars
<b>Margin collapsed</b>	Collapse all left-right and top-bottom margin using the parameters of the stroke with greater width.
<b>External margin width</b> <b>External margin color</b>	External margin parameters

### Add or remove rows or columns

To add or remove rows or columns, double click over the grid to enter in edit mode and right click over column or row selector to open the context menu.



### Merge or split rows or columns

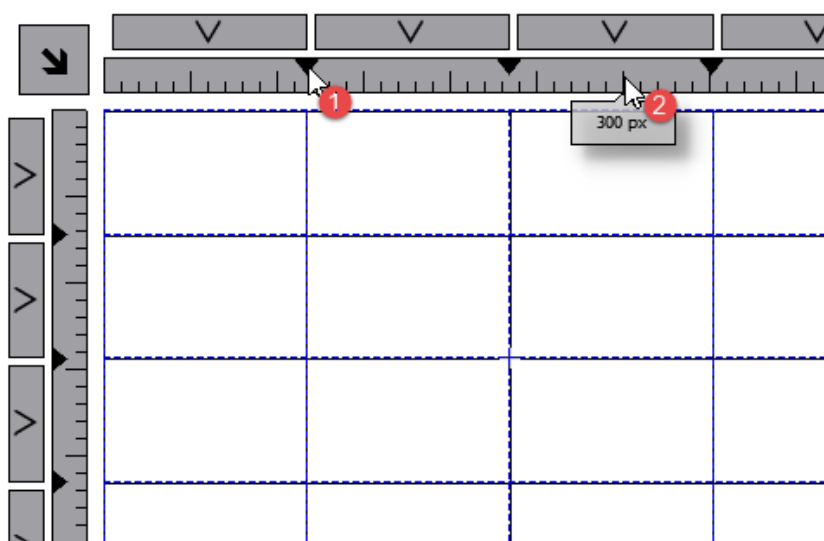
To merge or split rows or columns, double click over the grid to enter in edit mode and move the cursor over the ribbons:

- Double click the black triangle to merge the two adjacent rows or columns (1)



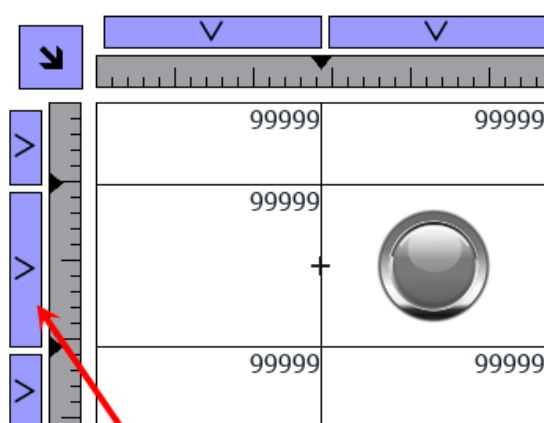
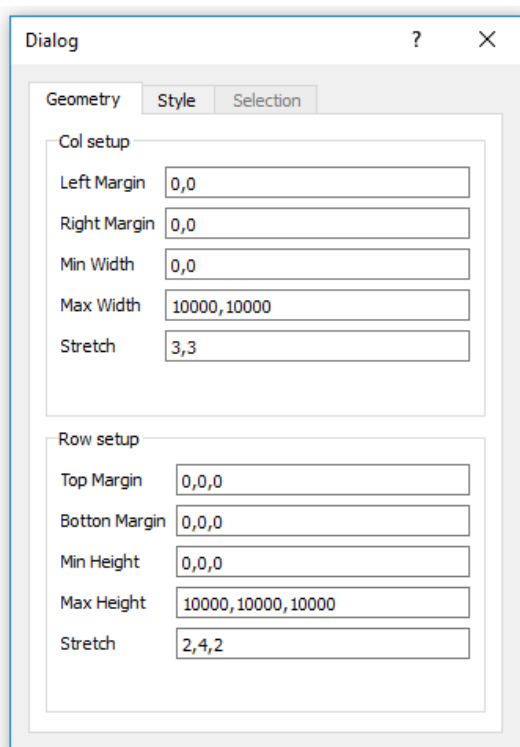
Note that merge is possible only with an empty row or column.

- Double click on ribbon to split the selected row or column (2)



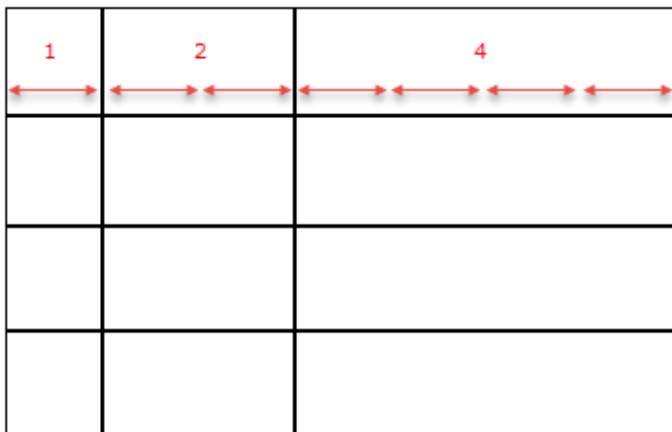
## Rows, Columns Properties

Row and columns properties are available inside a pop up dialog after clicking on the row and column selectors, that are visible after double clicking the group of widgets.

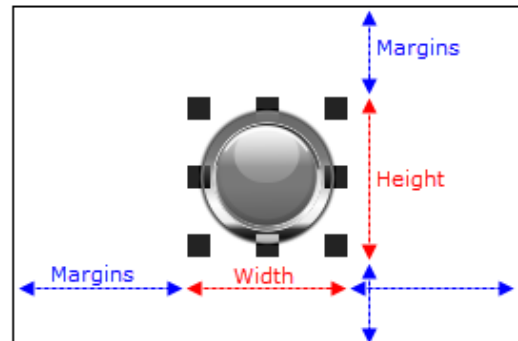


Rows and Columns Selectors

## Stretch



## Margins



### 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 height	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 417 for details)

Parameter	Description
<b>Foreground color</b>	Colors that the row assume when it is selected  The list of colors is related with row templates. First color is for row template 0, second color is for row template 1, and so on.
<b>Background color</b>	
<b>Stroke color</b>	

## 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.

The screenshot shows the JMobile Studio interface. On the left, a grid widget is displayed with a 3x2 layout. The top row contains two cells with the text '99999'. The middle row contains a cell with '99999' and a cell with a button icon. The bottom row contains two cells with the text '99999'. A selection tool is visible on the left side of the grid. On the right, the Properties panel is open, showing the properties for the selected button widget. The panel is titled 'Button : GroupWgt1.BtnStd1' and contains the following properties:

- 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
  - Bottom Margin: 15
  - Preserv aspect ratio: true
  - Aspect ratio: 1:1
  - Background: none

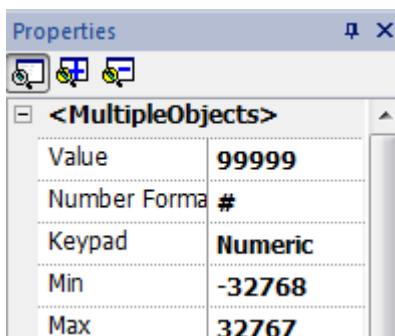
Parameter	Description
Horizontal underflow Vertical underflow	<p>This parameter defines the behavior of the widget when the cell is larger than the size defined for widget.</p> <ul style="list-style-type: none"> <li>Inherited Inherits the value used for the row or column</li> <li>Left, Center, Right - Top, Middle, Bottom Defines the position of the widgets when cells are bigger than the maximum defined sizes</li> </ul>
Max width Max height	Overwrite global grid parameters
Left margin Right margin Top margin Bottom margin	<p>Overwrite global grid parameters</p> <p>Additional pixels that are added to the total margin.</p>
Preserve aspect ratio	Preserve aspect ration of the widget
Aspect ratio	Available only when " <i>Preserve aspect ratio</i> " is true
Background	Background color of cell

## Changing multiple widgets properties

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

To change properties:

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



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



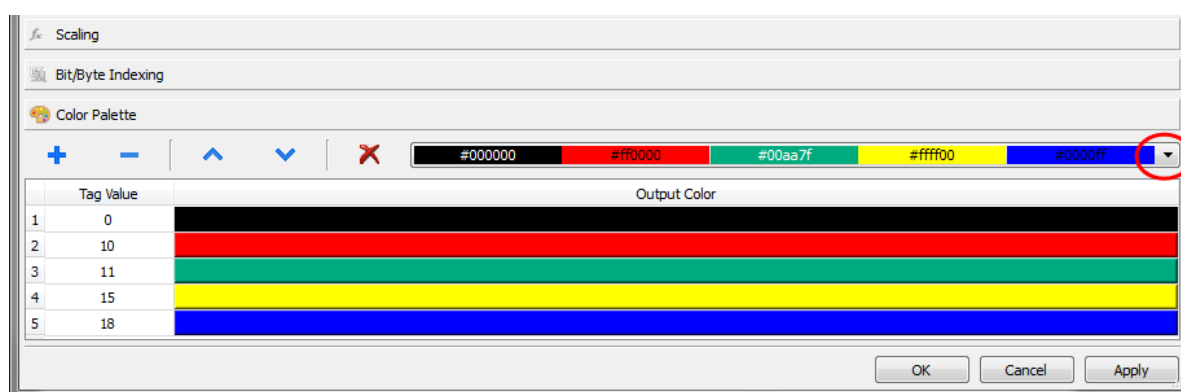
## Changing fill color property according to tag values

JMobile Studio allows to change the color property of a widget dynamically, based on tag values in two ways:

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

### Changing color property using ColorPalette

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



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

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

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



Note: This feature can be applied to all the objects available in the Widget gallery that have a color property. The run-time 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.

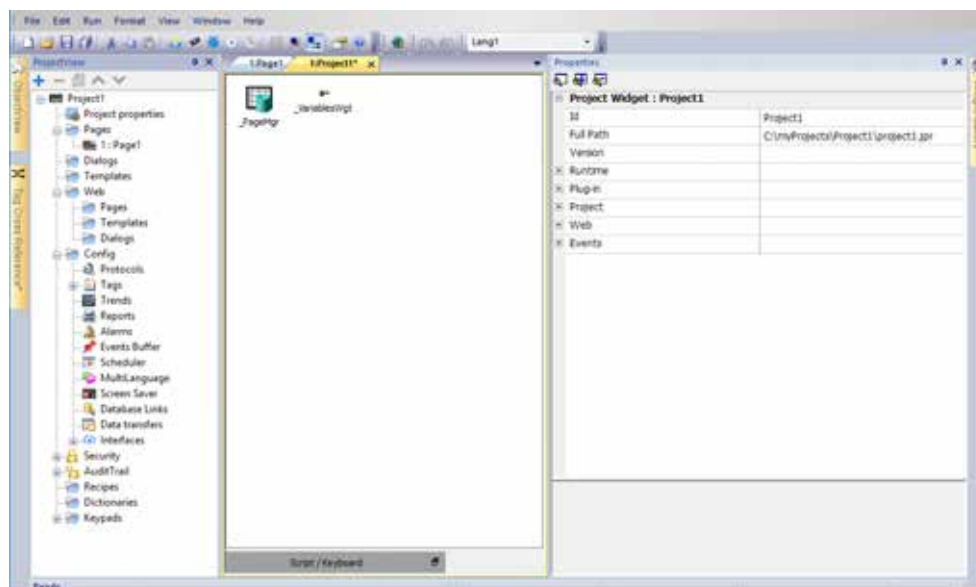


## 6 Project properties

Project properties contain settings for the project.

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

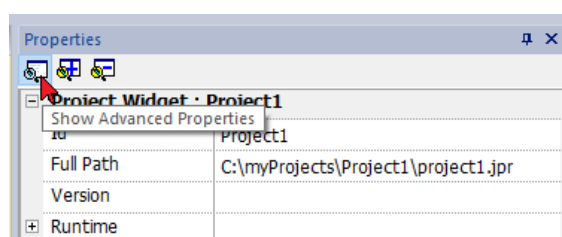
The project **Properties** pane contains a list of project level user-configurable data.



### Basic and advanced properties

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

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







### Available properties

Property	Description
<b>Id</b>	Project name (read only)
<b>Full Path</b>	Project path (read only)
<b>Version</b>	The Version field is available for users to report the project version.
<b>+ Runtime</b>	Properties related with the application runtime. See <a href="#">"Runtime" on the next page</a> for details.

Property	Description
+ Plug-In	Optional modules. See " <a href="#">Plug-in</a> " on page 77 for details.
+ Project	Properties related with the project. See " <a href="#">Project</a> " on page 78 for details.
+ Web	Properties related with the web interface. See " <a href="#">Web</a> " on page 82 for details.
+ Events	Global events. See " <a href="#">Events</a> " on page 83 for details.

## Runtime

Property	Description
Context Menu	<p>Define how context menu should appear in the HMI project.</p> <p><b>on delay</b> = context menu appears touching/pressing and holding for a few seconds an empty area of the runtime screen, or via <b>Context menu</b> action</p> <p><b>on action</b> = context menu appears only via <b>Context menu</b> action.</p> <p>See "<a href="#">Widget actions</a>" on page 205 for details.</p>
Developer Tool	Enable/disables a collection of runtime debugging utility tools.
Buzzer on Touch	<p>Enables buzzer when touching a widget on HMI device screen.</p> <p>Supported widgets:</p> <ul style="list-style-type: none"> <li>• buttons</li> <li>• hotspots</li> <li>• needles</li> <li>• fields</li> <li>• external keys</li> <li>• combo boxes</li> <li>• tables items</li> <li>• control list items</li> </ul> <p> <i>On Windows CE panels, available for from v1.76 ARM / 2.79 MIPS.</i></p>
Buzzer duration	Default 200 ms
Keyboard	Enables the use of keyboard macros at run time when using external keyboards.
JavaScript Debug	Enables the JavaScript debugger at run time for the current project.
Allow JS Remote Debugger	<p>Enables JavaScript remote debugger for current project.</p> <p> Remote debugging not supported on HMI Client.</p>

Property	Description
<b>Image DB enable</b>	<p>Activates an engine used by the Runtime to optimize project performance.</p> <p> <b>WARNING: This property should only be disabled by technical support for debugging purposes since this might reduce performance at run time.</b></p>
<b>FreeType Font Rendering</b>	<p>Switches to FreeType the font rendering used by JMobile Studio and runtime.</p> <p> 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.</p>
<b>Communication icon delay (ms)</b>	<p>Delay before display the communication error icon (default is 0 mSec)</p> <p>The special value -1 is meaning always disabled</p>
<b>Fast Boot</b>	<p>When fast boot is enabled and the User Interface is started before the background server</p> <ul style="list-style-type: none"> <li>• Default: User Interface is loaded after the background server is ready to use</li> <li>• Fast UI: User Interface is loaded before loading the background server</li> </ul>

## 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 503](#))

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

Example:

```

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

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

```

}



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

## Developer tools

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

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

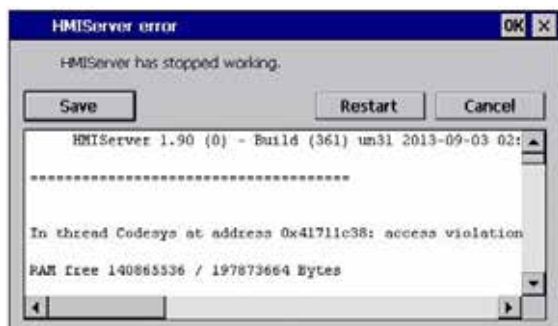
### Developer tools list

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

### Watchdog

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

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



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

## CPU Statistics

```

2014-04-25 23:02:48, up: 0:08:27, idle: 24 *
Period 2110 ms (overhead 69ms)
+   Thread      ID Prio    ms kernel/  user
   Codesys     78839810  0      8      0/    8
Other threads < 5ms
RAM free 125833216 / 194211840 Bytes (diff: 0)
ImageDB size ~2MB, free 4MB / 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         51     22     189
AlmMgr   0      0          0      0      0
MODR     0      0        122     11     15
VI       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    5/     5
Time lst update : +  195    3/     0
Time gfx creation: +  198  300/   133
      OnLoad    :      241/    94
Time rendering  : +  535  390/   387
ImageDB cache 15 hit/0 miss(0 ms, cpu: 0 ms)

Page "TemplatePagel.jmx":
Time init/start : +   60  133/    86
Time lst update : +  195    2/     0
Time gfx creation: +  459  27/    27
      OnLoad    :     9/     9
ImageDB cache 28 hit/0 miss(0 ms, cpu: 0 ms)

(Tap-tap to change position)

```

Profiling option	Description
<b>Enable timelog</b>	Enable timelog capture. Timing will be visible inside the "Timelog summary" window.
<b>Save timelog to file</b>	<p>Saves a report of profile details and the time spent loading a project and its pages into a timelog.txt file. This file can be exported and shared for further analysis.</p> <p> <b>Important: The execution of this function may reduce page change performance.</b></p>
<b>Overlay OnLoad times</b> <b>Overlay Rendering times</b>	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 Win32 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
<b>Select overlay color</b>	Select the overlay color to use

#### Timelog data

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

Times are provided in couples: wall time/CPU time. Wall time is the absolute time required by this part which can be higher than the actual CPU time required since higher priority threads are also running (for instance protocols). The start time



column refers to the page load start time. It can be used to track the actual time required to load a page, since partial times only refer to the most time critical functions and do not include other times that often contribute significantly to the total time.

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

## FreeType font rendering

New projects use the FreeType font engine as default. Projects created with older versions of JMobile Studio 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 Win32 platform.

Property	Description
<b>WebKit</b>	Module required by WebBrowser widget
<b>TextEditor</b>	Module required by TextEditor widget
<b>PDF Reader</b>	PDF Reader
<b>VNC Server</b>	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 JMobile Studio 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 Win32 platform.

## Project

These properties define various elements of page behavior.

Property	Description
Home Page	The first page loaded at run time (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 <a href="#">"User management and passwords" on page 309</a> 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.   <b>WARNING:</b> Starting from v2, the JMobile 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	See the below chapter
Hold Time Autorepeat Time	Defines the values for hold time and auto repeat time for buttons and external keyboards.   <b>Note:</b> These properties can be redefined for each button or key in their widget property table.
Hide Project Loading at boot	When hidden, the splash screen stay on the screen until the application is ready to run.
Target Zoom Factor	It is the zoom factor of the HMI device that will be applied when project is loaded at run time.

Property	Description						
	<table> <tr> <td><b>Range</b></td><td>0.3–2.9</td></tr> <tr> <td><b>Default value</b></td><td>1 = no zoom</td></tr> </table>	<b>Range</b>	0.3–2.9	<b>Default value</b>	1 = no zoom		
<b>Range</b>	0.3–2.9						
<b>Default value</b>	1 = no zoom						
<b>Background color option</b>	<p>When the defined page is smaller of the entire display area, colorize the area that is not covered from the page (for example when page is Zoom Out)</p> <table> <tr> <td><b>None</b></td><td>Old mode, color is white (default)</td></tr> <tr> <td><b>Selected color</b></td><td>Color to use</td></tr> <tr> <td><b>Page background</b></td><td>Auto adjust color based on background of template or of page</td></tr> </table>	<b>None</b>	Old mode, color is white (default)	<b>Selected color</b>	Color to use	<b>Page background</b>	Auto adjust color based on background of template or of page
<b>None</b>	Old mode, color is white (default)						
<b>Selected color</b>	Color to use						
<b>Page background</b>	Auto adjust color based on background of template or of page						
<b>Signature</b>	<p>Algorithm to use to signing</p> <ul style="list-style-type: none"> <li>• sha256</li> <li>• sha1</li> </ul>						
<b>Gesture Passthru Enabled</b>	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.						
<b>Gesture Passthru Delay (ms)</b>	When enabled, the gesture events are passed to underlying widgets after this delay (see <a href="#">"Gesture events pass thru" on page 402</a> for details)						
<b>Gesture Multitouch</b>	<p>Enable multi touch gestures</p> <ul style="list-style-type: none"> <li>• false</li> <li>• true (default)</li> </ul> <p> This property give the possibility to disable the multi touch gestures. This could be useful to avoid problems with old projects that were not designed to manage the multi touch gestures.</p>						
<b>On Access Denied</b>	<p>When user try to use a widget that is locked from the security configuration to read-only (e.g. a field or a button), a padlock icon is shown for a couple of seconds to highlight that the widget is not accessible.</p> <ul style="list-style-type: none"> <li>• None</li> <li>• Show Icon</li> </ul>						
<b>ComboBox View Mode</b>	<p>Select the visualization mode of all the Combo Box widgets of the project (see <a href="#">"Combo Box widget "full screen" mode with images" on page 391</a> for details)</p> <ul style="list-style-type: none"> <li>• Context Classic view with drop-down menus</li> <li>• Full screen Enhanced view with configurable texts and images that will pop up in the middle of the screen for easy scroll and selection.</li> </ul>						

## PageRequest, CurrentPage and SyncOptions

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

The following properties can be customized:

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

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

Set properties as follows:

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

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

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

Set properties as follows:

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

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

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

Set properties as follows:

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

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

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

Set properties as follows:

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

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

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

Set properties as follows:

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

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

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

Set properties as follows:

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

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

# Web


Property	Description	
<b>Web Inactivity Timeout</b>	Defines a timeout for JM4Web client. When the timeout expires without any activity the current user is logged out.	
	<b>Range</b>	1–86400 s (from 1 s to 24 h)
	<b>Default value</b>	600 s
	<b>Values</b>	0 = disabled
<b>Web Icon</b>	The favorite icon associate at the web pages	
<b>Refresh Time</b>	Defines the refresh time for the communication between the runtime and JM4Web clients.	
	<b>Range</b>	500–10000 ms
	<b>Default value</b>	3000 ms
<b>Browser Optimization</b>	<b>true</b>	Web engine optimization enable (default)
	<b>false</b>	Web engine optimization disable (useful for old browsers that not support the web engine optimization)
<b>Enable Global JavaScript for remote</b>	Define if the JavaScript code defined inside the Project Properties, general triggered from Alarms and Schedulers events, have to run only on local HMI device or even on remote clients.	
	<b>None</b>	Will not be executed on remote clients (run only inside the local HMI device)
	<b>Client</b>	Will be executed on HMI Client
	<b>Web</b>	Will be executed on Web client
	<b>Both</b>	Will be executed on both HMI Client and Web clients
<b>Max Bandwidth (Kbs)</b>	Limit for maximum data sent by server (useful for old slow browsers). Set to 0 to use all the available bandwidth (default)	
<b>Web clients connection mode</b>	<b>Auto</b>	The connection mode is selected by the client (default)
	<b>SSE</b>	Force the Server-Sent Events mode
	<b>Long Polling</b>	Force the Long-polling mode
<b>WebPageRequest</b>	You can synchronize pages shown on the JM4Web Clients from a controller such as a PLC.	
	Page to be shown on the JM4Web Client.	

Property	Description
	Attached tag must contain an integer value within the range of the available project pages and must be available at least as a Read resource.
<b>Web Communication icon delay (ms)</b>	<p>Delay before display the communication error icon (default is 0 mSec)</p> <p>The special value -1 is meaning always disabled</p>



The `project.getClientType()` can be used to retrieve the running client type. See "[Project object](#)" on page 473 inside JavaScript chapter for additional details.

## Events

Property	Description
<b>OnWheel</b>	<p>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.</p> <p>Attach this property to a change of wheel event and use an action like <b>BiStep</b> to increase/decrease a tag value.</p> <p> <b>The project's OnWheel Action is executed only when the OnWheel Action will not overwritten from the loaded page.</b></p>





# 7 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.

---

<b>Data simulation methods</b> .....	<b>86</b>
<b>Simulator settings</b> .....	<b>86</b>
<b>Launching and stopping the simulator</b> .....	<b>87</b>

# Data simulation methods

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

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

See ["Adding tags" on page 32](#) for details.

## Simulator settings

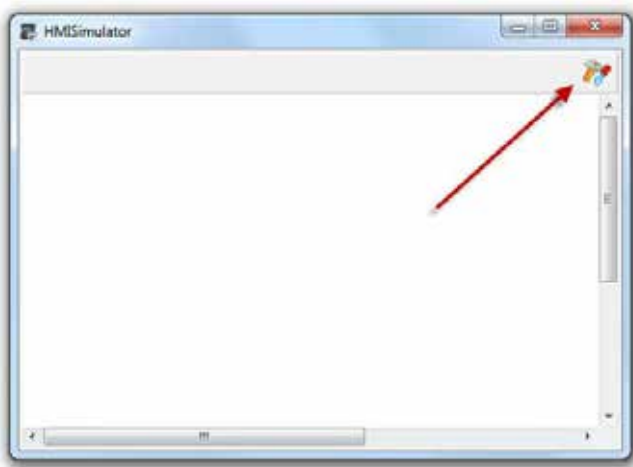
The Simulator works by default with simulated protocols. It can also work with real protocols (Ethernet or serial protocols)



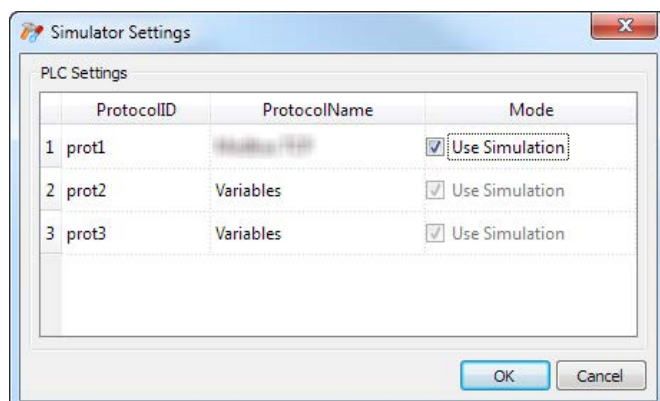
Note: For protocols not supporting communication with external devices, such as the Variables protocol, this option is always disabled.

## Changing simulated protocols

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



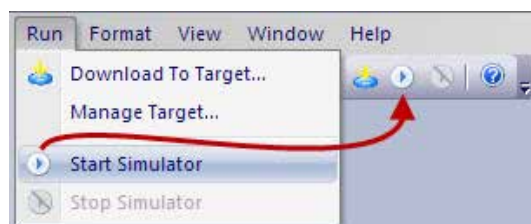
2. Select **Use Simulation** to use simulated protocols, otherwise real protocols will be used for communication with external devices.



## Launching and stopping the simulator

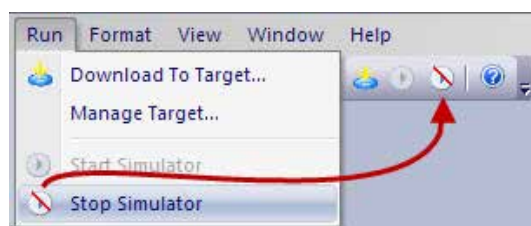
To launch the simulator:

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.





# 8 Transferring the project to HMI device

---

To transfer the JMobile Studio project to the target HMI device you can use:

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

---

<b>Download to HMI device</b> .....	<b>90</b>
<b>Update package</b> .....	<b>93</b>
<b>The Runtime loader</b> .....	<b>95</b>
<b>Upload projects</b> .....	<b>96</b>

# Download to HMI device

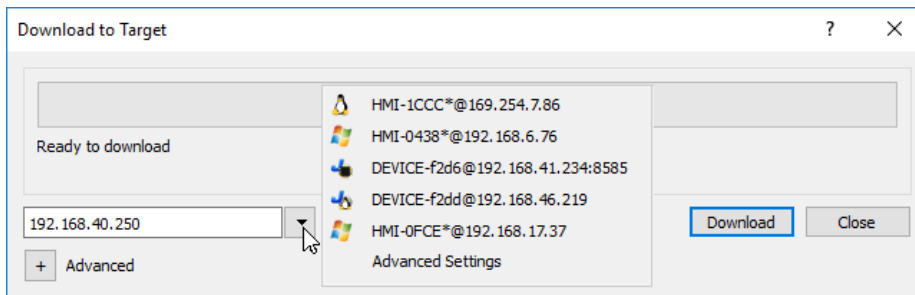
## Path: Run> Download to Target

This function transfers project and JMobile HMI Runtime via Ethernet .



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

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

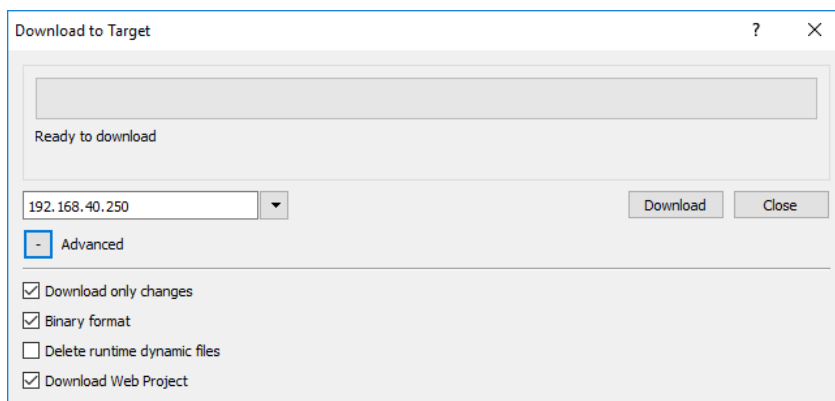


You can even enter the IP address manually or, if available, the host name provided by a DNS server. Using a service tool like Bonjour, Linux-based HMI devices can be discovered using their hostname (e.g HMI-0d37.local). Bonjour is a trademark of Apple inc.



3. Click **Download**: JMobile Studio will switch the HMI device to Configuration Mode and transfer the files.

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

## Advanced options



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

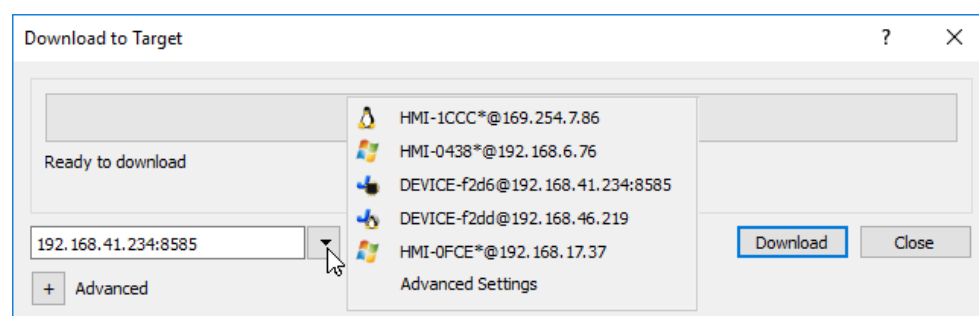
Option	Description
<b>Delete runtime dynamic files</b>	<p>Modified configuration of recipes, users, schedulers, etc. done at run time will be deleted and overwritten by the configuration defined in the project.</p> <p> <b>CAUTION: This operation cannot be undone, deleted dynamic files cannot be restored.</b></p> <p> <b>CAUTION: Dynamic files are not deleted if stored on external devices (USB or SD Cards).</b></p>
<b>Download Web Project</b>	Download the JM4Web pages to HMI device.

When transferring a project, JMobile Studio 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

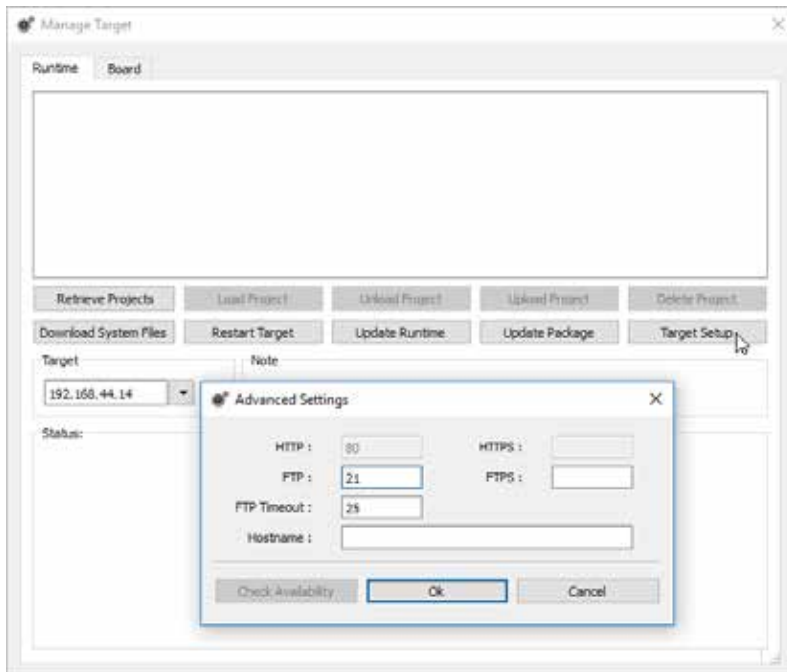
Using the “Advanced Settings” option, you can define the ports to use, but generally, you do not need to enter this information because HMI devices will provide the ports to use inside the panesI list.



## Changing HMI device connection settings

Path: **Run> Manage Target**

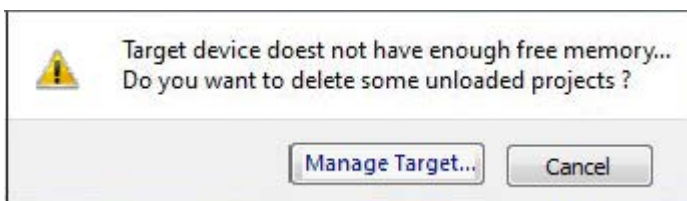
1. Click **Target Setup**: the **Advanced Settings** dialog is displayed. Default port for HTTP connections on the HMI device is port 80.



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

## Managing big projects

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



To free more memory:

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



## Update package

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

### Creating an update package

**Path:** *Run> Update Package*

**Update Package**

Target: [Dropdown]

☒ HMI Runtime ☐ HMI Client

☒ Project

☒ HMI Runtime & Plug-In

☒ Binary format

☒ Web Project

☐ Set Target Password [Browse]

☐ User Files

[Text Field] [Browse]

☐ Encrypted

Location : [Text Field] [Browse]

[Create] [Cancel]

Option	Description
<b>Target</b>	HMI device type. Selected automatically if the project is open.
<b>Application Selector</b>	Select the application to insert inside the UpdatePackage.zip <ul style="list-style-type: none"> <li>• HMI Runtime</li> <li>• HMI Client (Available only on Linux devices)</li> </ul>

Option	Description
<b>Project</b>	Adds open project to update package.
<b>HMI Runtime &amp; Plug-In</b>	HMI Runtime is added to the update package. If the project is open the required plugins are also added to update package.
<b>Binary Format</b>	Download files using binary format.
<b>Web Project</b>	Download the JM4Web pages to HMI device.
<b>Set Target Password</b>	Sets password to perform critical tasks (for example, project download/upload , board management)  See " <a href="#">Protecting access to HMI devices</a> " on page 553.
<b>User Files</b>	Selects files to be copied to the QTHM folder of HMI device. Max size 5 MB
<b>Encrypted</b>	Enables encryption of update package so that it can only be unzipped by the HMI Runtime.
<b>Location</b>	Location of update package.



**Important:** When create a package with the HMI Runtime application, always include both project and the runtime. If you need to use an old project with the latest Runtime version, convert the project first. See "[Installing the application](#)" on page 4 for details.

### Example of user's file location

Computer:

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

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

WinCE devices:

*/Flash/QtHmi*

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

Linux devices:

*/mnt/data/hmi/qthmi*

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

## Loading an update package

Path: from the context menu > **Update**

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



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

## The Runtime loader

HMI devices are delivered from factory without Runtime.

When you power up the device for the first time, the Runtime Loader window is displayed (see ["System Settings" on page 499](#) 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 JMobile Studio

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



See ["Transferring the project to HMI device" on page 89](#) 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 93](#)
2. Plug the USB drive in the device and follow the instructions for the type of device (see ["System Settings" on page 499](#) 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 JMobile Studio is running.

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

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



Upload could be password protected. See ["Protecting access to HMI devices" on page 553](#) for details.

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

A copy of the project is saved in:

`C:\Users\username\Documents\JMobile Studio\workspace\Uploaded\RuntimeIPAddress\workspace\ProjectName`



*Note: If the upload operation fails, check firewall settings the computer where JMobile Studio is running.*

# 9 System Variables (Attach To)

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

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



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

<b>Alarms variables</b>	99
<b>Buzzer variables</b>	99
<b>Communication variables</b>	100
<b>Database variables</b>	100
<b>Daylight Saving Time variables</b>	101
<b>Device variables</b>	102
<b>Dump information variables</b>	103
<b>FTP client variables</b>	104
<b>Keypad variables</b>	105
<b>Network variables</b>	105

---

<b>PLC variables .....</b>	<b>105</b>
<b>Printing variables .....</b>	<b>106</b>
<b>Remote Client variables .....</b>	<b>107</b>
<b>Version variables .....</b>	<b>108</b>
<b>Screen variables .....</b>	<b>108</b>
<b>SD card variables .....</b>	<b>108</b>
<b>Server variables .....</b>	<b>109</b>
<b>Time variables .....</b>	<b>109</b>
<b>Touch screen variables .....</b>	<b>110</b>
<b>USB drive variables .....</b>	<b>111</b>
<b>User management variables .....</b>	<b>111</b>

## Alarms variables

Number of alarms of the requested type.


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



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

## Buzzer variables

Adjust buzzer behavior.

Variable	Description	Data type
<b>Buzzer Setup</b>	<p>0 = disabled  1 = enabled (buzzer sounds as audible on any touchscreen event)  2 = buzzer status controlled by <b>Buzzer Control</b> system variable or by <b>Buzzer on Touch</b> property inside the <a href="#">"Project properties"</a> on page 71</p>  <p><b>Buzzer on touchscreen (Setup=1) is not available on Linux platforms. See "Buzzer on Touch" property in alternative.</b></p>	int
<b>Buzzer Control</b>	<p>0 = buzzer off  1 = buzzer on  2 = buzzer blink</p>	int
<b>Buzzer Off Time</b>	Duration in milliseconds of off time when blink has been selected. Default = 1000. Range: 100–5000.	int
<b>Buzzer On Time</b>	Duration in milliseconds of on time when blink has been selected. Default = 1000. Range: 100–5000.	int

## Communication variables

Communication status between HMI device and controllers.

Variable	Description	Data type
<b>Protocol Communication Status</b>	<p>Summarize the status of the communication protocols.</p> <p>0 = No protocol running, protocol drivers might not have been properly downloaded to the HMI device.</p> <p>1 = Protocols loaded and started, no communication error.</p> <p>2 = At least one communication protocol is reporting an error.</p>	int Read only
<b>Protocol Error Message</b>	<p>Communication error with error source.</p> <p>For example: "[xxxx]" where "xxxx" is the protocol abbreviation, the error source.</p> <p>Multiple acronyms appear in case of multiple error sources. Blank when no errors are reported.</p>	ASCII string Read only
<b>Protocol Error Count</b>	Number of communication errors occurred since last reset. Reset value with Reset Protocol Error Count action, see <a href="#">"System actions"</a> on page 184.	int Read only

## Database variables

Database connection status .



Variable	Description	Data type
<b>Database link error message</b>	Last detected error description	string read only
<b>Database link status</b>	0 = Undefined (not yet initialized) 1 = OnLine (ready) 2 = OffLine (not available) 3 = Transfer in progress 4 = Error	int read only
<b>Database link error count</b>	Errors counter. Increased after each error.	int read only

Each database variable is an array where index select the database link connection (Range 1-10)



Note: Variables are updated only when any database connector action is executed.



Note: These variables are available as tags from the "System Variables" protocol..

## Daylight Saving Time variables

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





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




Variable	Description
<b>Standard Offset</b>	Offset in minutes when standard time is set, with respect to GMT (for example: -8x60 = -480 minutes).
<b>Standard Week</b>	Week in which the standard time starts (for example: First = 1).
<b>Standard Month</b>	Month in which the standard time starts. Range: 0–11. (for example: November = 10).
<b>Standard Day</b>	Day of week in which the standard time starts (for example: Sunday = 0).
<b>Standard Hour</b>	Hour in which the standard time starts (for example: 02 = 2).
<b>Standard Minute</b>	Minute in which the standard time starts (for example: 00 = 0).
<b>DST Offset</b>	Offset in minutes when DLS time is set, with respect to GMT

Variable	Description
<b>DST Week</b>	Week in which the DLS time starts
<b>DST Month</b>	Month in which the DLS time starts. Range: 0–11.
<b>DST Day</b>	Day of week in which the DLS time starts
<b>DST Hour</b>	Hour in which the DLS time starts
<b>DST Minute</b>	Minute in which the DLS time starts

## Device variables

Device settings and operating status information.

Variable	Description	Data type
<b>Available System Memory</b>	Free available RAM memory in bytes.	uint64 read only
<b>Backlight Time</b>	Activation time in hours of the display backlight since production of the device.	unsignedInt read only
<b>Battery LED</b>	<p>Enables/disables the low battery LED indicator (when available).</p> <p><b>0</b> = disabled <b>1</b> = enabled</p> <p> Not available on Linux platforms (find the platform of your device at "<a href="#">HMI devices capabilities</a>" on page 567)</p>	int
<b>Battery Timeout</b>	<p>Reserved</p> <p> Not available on Linux platforms (find the platform of your device at "<a href="#">HMI devices capabilities</a>" on page 567)</p>	int
<b>Display Brightness</b>	<p>Returns and adjusts brightness level.</p> <p>Even when set to 0, the backlight is still on and the <b>Backlight Time</b> counter increases.</p> <p>Range: 0–255</p> <p>On WinCE device only: When set to a low light level (0..3), the backlight stays lit to a higher level for 8 seconds to allow the user to make the adjustments and then is switched-off.</p>	int
<b>External Timeout</b>	Non-operational time after which the display backlight is automatically turned off. The backlight is automatically turned on when the user touches the screen.	int

Variable	Description	Data type
	<p><b>-1 =</b> Switch off backlight and disable touch (switch display off). <b>Backlight Time</b> counter is stopped.</p> <p> On Linux devices requires BSP v1.0.324 or higher.</p> <p><b>-2 =</b> Switch off backlight but not disable touch. If touch is pressed, event is not passed to applications but screen saver exit and backlight return on.</p> <p> Available only on Linux devices. Requires BSP v1.0.324 or higher.</p> <p><b>0 =</b> Switch backlight on (switch display on)</p> <p><b>1..n =</b> Timeout, in seconds, for switch off backlight (screen saver timer)</p> <p> The timeout value is rounded to multiples of one minute (60, 120, 180, etc,) in all Linux devices.</p> <p>Find the platform of your device at <a href="#">"HMI devices capabilities" on page 567</a></p>	
<b>Flash Free Space</b>	Free space left in internal Flash memory.	uint64 read only
<b>Manufacturer Code</b>	Internal code that identify the HMI type	unsignedInt read only
<b>System Font List</b>	List of system fonts	string read only
<b>System Mode</b>	<p>Runtime operation status.</p> <p><b>1 =</b> booting</p> <p><b>2 =</b> configuration mode</p> <p><b>3 =</b> operating mode</p> <p><b>4 =</b> restart</p> <p><b>5 =</b> shutdown</p>	int
<b>System UpTime</b>	Time the system has been powered since production of the unit (hours).	unsignedInt read only

## Dump information variables

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

Variable	Description	Data type
<b>Dump Error Message</b>	Return error message if any error occurs during the dump operation	string read only
<b>Dump Archive Status</b>	0 = initial default state 1 = operation triggered 2 = operation complete successfully 3 = operation completed with errors	int read only
<b>Dump Recipe Status</b>	0 = initial default state 1 = operation triggered 2 = operation complete successfully 3 = operation completed with errors	int read only
<b>Dump Trend Status</b>	0 = initial default state 1 = operation triggered 2 = operation complete successfully 3 = operation completed with errors	int read only
<b>Reset Recipe Status</b>	0 = initial default state 1 = operation triggered 2 = operation complete successfully 3 = operation completed with errors	int read only
<b>Restore Recipe Status</b>	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
<b>FTP Current Command</b>	Last FTP command	string read only
<b>FTP Error Message</b>	Last FTP error message	string read only
<b>FTP Progress</b>	Download/upload progress (0/100%)	short read only
<b>FTP Status</b>	Status of last FTP command: <ul style="list-style-type: none"> <li>0 = idle</li> <li>1 = active</li> <li>2 = done</li> <li>3 = error</li> </ul>	short read only

## Keypad variables

Keypad status.

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

## Network variables

Device network parameters.

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

## PLC variables

Status of CODESYS V2 system integrated in HMI devices.

Variable	Description	Data type
<b>PLC Status</b>	Status of integrated CODESYS V2.  0 = RUN  1 = PROGRAM NOT LOADED (program not loaded in memory or CODESYS module not running because license missing)  2 = STOP (program loaded but not running)	int  read only
<b>Get CopyCodesysProject Action Status</b>	Status of CopyCodesysProject action related to integrated CODESYS V2.  0 = ACTION_NOT_CALLED 1 = ACTION_IN_PROGRESS 2 = ACTION_COMPLETED 3 = ACTION_ABORTED_CHK_FILE_MISSING 4 = ACTION_ABORTED_PRG_FILE_MISSING 5 = ACTION_ABORTED_SDB_FILE_MISSING 6 = ACTION_ABORTED_MUTIPLE_CHK_FILES_FOUND 7 = ACTION_ABORTED_MUTIPLE_PRG_FILES_FOUND 8 = ACTION_ABORTED_MUTIPLE_SDB_FILES_FOUND 9 = ACTION_ABORTED_INCONSISTENT_FILE_NAMES 10 = ACTION_ABORTED_UNABLE_TOMAKE_TARGET_DIR 11 = ACTION_ABORTED_COPY_FAILED 12 = ACTION_ABORTED_CODESYS_MODULE_NOT_PRESENT	int  read only

## Printing variables


Information on printing functions.

Variable	Description	Data type
<b>Completion percentage</b>	Percentage of completion of current print job.  Range: 0–100	read only
<b>Current disk usage</b>	Folder size in bytes where PDF reports are stored.  If <i>Flash</i> has been selected as <i>Spool media type</i> , this value corresponds to <i>reportspool</i> .	read only
<b>Current job</b>	Name of the report the job is processing. Current job is the following:  • [report name] for a <b>Graphic Report</b>	read only

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

## Remote Client variables

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

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

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

Variable	Description	Data type
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	<b>0</b> = idle, action is not in use or completed <b>1</b> = file download in progress <b>2</b> = error	int (32 bit) read only
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	<b>0</b> = idle, action is not in use or completed <b>1</b> = file upload in progress <b>2</b> = error	int (32 bit) read only

## Version variables

Operating System and runtime version.

Variable	Description	Data type
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 <b>LockScreen</b> action, " <a href="#">Page actions</a> " on page 170)
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
<b>SD Card FreeSpace</b>	Available space on card in bytes	uint64 read only
<b>SD Card Name</b>	Name of SD card	string read only
<b>SD Card Size</b>	Size in bytes of the card plugged in the slot	uint64 read only
<b>SD Card Status</b>	0 = SD card unplugged 1 = SD card plugged	int read only

## Server variables

Server status.



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

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

## Time variables

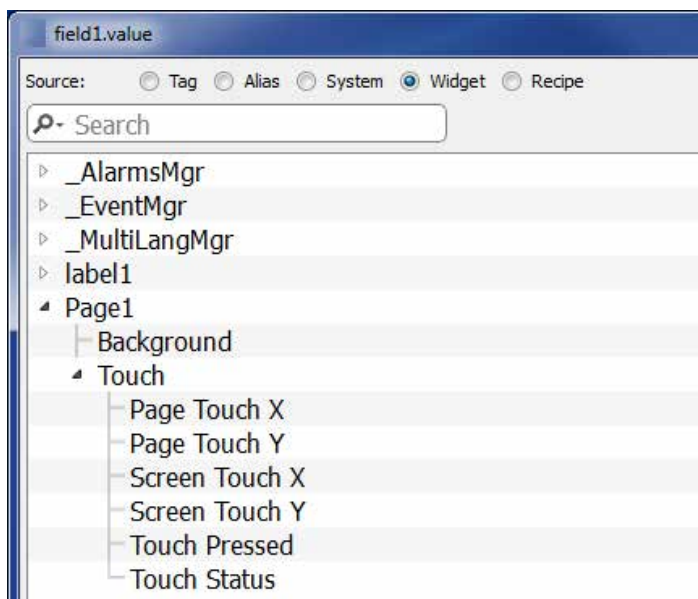
System time expressed in UTC format.

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

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

## Touch screen variables

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



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

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

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

## USB drive variables

Information on the external USB drive connected to the device.

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

## User management variables

Information on users and groups.

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

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

### JavaScript

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

Example:

```
var sysVar = project.getWidget( "_SysPropMgr" );

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

# 10 System Variables (Protocol)

---

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



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

---

<b>Protocol Editor Settings</b> .....	<b>114</b>
<b>Default variables</b> .....	<b>114</b>
<b>HandHeld variables</b> .....	<b>129</b>
<b>HandHeld - wireless variables</b> .....	<b>133</b>
<b>Direct I/O variables</b> .....	<b>139</b>
<b>PLCM09 variables</b> .....	<b>144</b>
<b>Services variables</b> .....	<b>148</b>
<b>Retentive Memory variables</b> .....	<b>151</b>
<b>Tag Import</b> .....	<b>156</b>

# Protocol Editor Settings

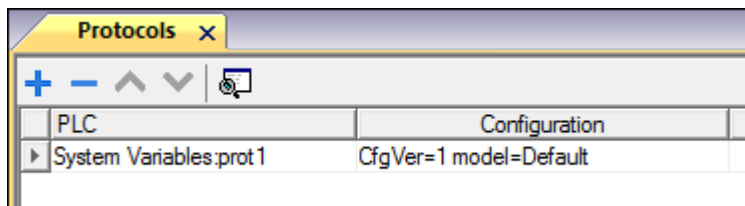
## Adding a protocol

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.



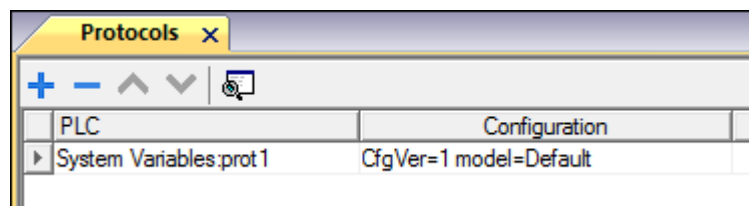
## Default variables

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

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

## Protocol Editor Settings

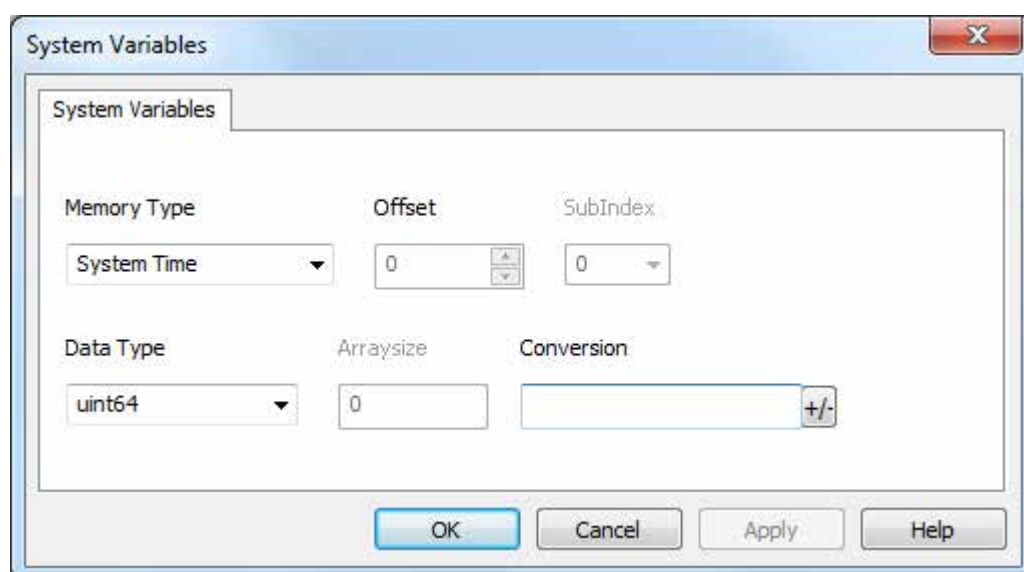
From PLC Model list of Protocol Editor dialog, select Default.



## Tag Editor Settings


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

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











Element	Description		
Memory Type	Represents the system variable to which the Tag refers to.		
	The below section shows the full list of possible system variables, grouped by category.		
	Alarms Variables		
	Variable Name	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	Alarm 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: acknowledgment not required	int read only
Number of triggered not acknowledged	Alarm condition active; awaiting acknowledgment	int read only	






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

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


Element	Description																																										
	<div> Each database variable is an array where index select the database link connection (Range 1-10) Variables are updated only when any database connector action is executed</div> <div><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>Standard Offset</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>Standard Week</td><td>Week in which the standard time starts (for example: First = 1)</td><td>int read only</td></tr><tr><td>Standard Month</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>Standard Day</td><td>Day of week in which the standard time starts (for example: Sunday = 0)</td><td>int read only</td></tr><tr><td>Standard Hour</td><td>Hour in which the standard time starts (for example: 02 = 2)</td><td>int read only</td></tr><tr><td>Standard Minute</td><td>Minute in which the standard time starts (for example: 00 = 0)</td><td>int read only</td></tr><tr><td>DST Offset</td><td>Offset in minutes when DLS time is set, with respect to GMT</td><td>int read only</td></tr><tr><td>DST Week</td><td>Week in which the DLS time starts</td><td>int read only</td></tr><tr><td>DST Month</td><td>Month in which the DLS time starts. Range: 0–11</td><td>int read only</td></tr><tr><td>DST Day</td><td>Day of week in which the DLS time starts</td><td>int read only</td></tr><tr><td>DST Hour</td><td>Hour in which the DLS time starts</td><td>int read only</td></tr><tr><td>DST Minute</td><td>Minute in which the DLS time starts</td><td>int read only</td></tr></table></div>	Daylight Saving Time Variables			Variable Name	Description	Data Type	Standard Offset	Offset in minutes when standard time is set, with respect to GMT (for example: -8x60 = -480 minutes)	int read only	Standard Week	Week in which the standard time starts (for example: First = 1)	int read only	Standard Month	Month in which the standard time starts. Range: 0–11. (for example: November = 10)	int read only	Standard Day	Day of week in which the standard time starts (for example: Sunday = 0)	int read only	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
Daylight Saving Time Variables																																											
Variable Name	Description	Data Type																																									
Standard Offset	Offset in minutes when standard time is set, with respect to GMT (for example: -8x60 = -480 minutes)	int read only																																									
Standard Week	Week in which the standard time starts (for example: First = 1)	int read only																																									
Standard Month	Month in which the standard time starts. Range: 0–11. (for example: November = 10)	int read only																																									
Standard Day	Day of week in which the standard time starts (for example: Sunday = 0)	int read only																																									
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																																									

Element	Description																								
	<div> All variables are read only: they cannot be used to update the system clock.</div> <table><tr><th colspan="3">Device Variables</th></tr><tr><th>Variable Name</th><th>Description</th><th>Data Type</th></tr><tr><td>Available System Memory</td><td>Free available RAM memory in bytes</td><td>uint64 read only</td></tr><tr><td>Backlight Time</td><td>Activation time in hours of the display backlight since production of the device</td><td>unsignedInt read only</td></tr><tr><td>Battery LED</td><td>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 "<a href="#">HMI devices capabilities</a>" on page 567)</td><td>int</td></tr><tr><td>Battery Timeout</td><td>Reserved   Not available on Linux platforms (find the platform of your device at "<a href="#">HMI devices capabilities</a>" on page 567)</td><td>int</td></tr><tr><td>Display Brightness</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>External Timeout</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  -1 = Switch off backlight and disable touch (switch display off). <b>Backlight Time</b> counter is stopped.</td><td>int</td></tr></table>	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 (when available)  0 = disabled  1 = enabled   Not available on Linux platforms (find the platform of your device at " <a href="#">HMI devices capabilities</a> " on page 567)	int	Battery Timeout	Reserved   Not available on Linux platforms (find the platform of your device at " <a href="#">HMI devices capabilities</a> " on page 567)	int	Display Brightness	Returns and adjusts brightness level.  When set to a low light level (0..3), the backlight stays lit to a higher level for 8 seconds to allow the user to make the adjustments and then is switched-off.  Even when set to 0, the backlight is still on and the Backlight Time counter increases. Range: 0–255	int	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). <b>Backlight Time</b> counter is stopped.	int
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 (when available)  0 = disabled  1 = enabled   Not available on Linux platforms (find the platform of your device at " <a href="#">HMI devices capabilities</a> " on page 567)	int																							
Battery Timeout	Reserved   Not available on Linux platforms (find the platform of your device at " <a href="#">HMI devices capabilities</a> " on page 567)	int																							
Display Brightness	Returns and adjusts brightness level.  When set to a low light level (0..3), the backlight stays lit to a higher level for 8 seconds to allow the user to make the adjustments and then is switched-off.  Even when set to 0, the backlight is still on and the Backlight Time counter increases. Range: 0–255	int																							
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). <b>Backlight Time</b> counter is stopped.	int																							

Element	Description	
	<b>Device Variables</b>	
	<b>Variable Name</b>	<b>Description</b>
		<p> On Linux devices requires BSP v1.0.324 or higher.</p> <p><b>-2</b> = Switch off backlight but not disable touch. If touch is pressed, event is not passed to applications but screen saver exit and backlight return on.</p> <p> Available only on Linux devices. Requires BSP v1.0.324 or higher.</p> <p><b>0</b> = Switch backlight on (switch display on)</p> <p><b>1..n</b> = Timeout, in seconds, for switch off backlight (screen saver timer)</p> <p> The timeout value is rounded to multiples of one minute (60, 120, 180, etc,) in all Linux devices.</p> <p>Find the platform of your device at "<a href="#">HMI devices capabilities</a>" on page 567</p>
	<b>Flash Free Space</b>	Free space left in internal Flash memory
		uint64 read only
	<b>Manufacturer Code</b>	Code number that identifies the HMI
		short read only
	<b>System RAM Usage</b>	Current RAM memory used from HMI, expressed in byte
		uint64 read only
	<b>System Font List</b>	List of system fonts
		string read only
	<b>System Mode</b>	Runtime operation status
		<p><b>1</b> = booting</p> <p><b>2</b> = configuration mode</p> <p><b>3</b> = operating mode</p> <p><b>4</b> = restart</p>
		int



Element	Description		
	<b>Device Variables</b>		
	<b>Variable Name</b>	<b>Description</b>	<b>Data Type</b>
		<b>5</b> = shutdown	
	<b>System UpTime</b>	Time the system has been powered since production of the unit (hours)	unsignedInt read only
	<b>Dump information Variables</b>		
	<b>Variable Name</b>	<b>Description</b>	<b>Data Type</b>
	<b>Dump Error Message</b>	Return error message if any error occurs during the dump operation	string read only
	<b>Dump Archive Status</b>	<b>0</b> = initial default state <b>1</b> = operation triggered <b>2</b> = operation complete successfully <b>3</b> = operation completed with errors	int read only
	<b>Dump Recipe Status</b>	<b>0</b> = initial default state <b>1</b> = operation triggered <b>2</b> = operation complete successfully <b>3</b> = operation completed with errors	int read only
	<b>Dump Trend Status</b>	<b>0</b> = initial default state <b>1</b> = operation triggered <b>2</b> = operation complete successfully <b>3</b> = operation completed with errors	int read only
	<b>Reset Recipe Status</b>	<b>0</b> = initial default state <b>1</b> = operation triggered <b>2</b> = operation complete successfully <b>3</b> = operation completed with errors	int read only
	<b>Restore Recipe Status</b>	<b>0</b> = initial default state <b>1</b> = operation triggered <b>2</b> = operation complete successfully <b>3</b> = operation completed with errors	int read only

Element	Description		
	<b>Network Variables0</b>		
	<b>Variable Name</b>	<b>Description</b>	<b>Data Type</b>
	<b>Gateway</b>	Gateway address of the main Ethernet interface of HMI	string read only
	<b>IP Address</b>	IP address of the main Ethernet interface of HMI	string read only
	<b>Mac ID</b>	MAC ID of the main Ethernet interface of HMI	string read only
	<b>Network Adapter Parameters</b>	JSON string that can be use to read or update the network adapters parameters	string
	<b>Network Status</b>	Contains the result of the last operation required by writing inside the Adapter Parameters. It is updated after each write operation. <ul style="list-style-type: none"> <li>• Empty string is meaning no errors</li> <li>• Last error descriptions</li> </ul>	string read only
	<b>Subnet Mask</b>	Subnet Mask of the main Ethernet interface of HMI	string read only

Element	Description		
	<b>PLC Variables</b>		
	<b>Variable Name</b>	<b>Description</b>	<b>Data Type</b>
	<b>Get CopyCodesysProject Action Status</b>	Provides status of CopyCodesysProject action. <b>0</b> = Action not called <b>1</b> = Action in progress <b>2</b> = Action completed <b>3</b> = Action aborted (CHK file missing) <b>4</b> = Action aborted (PRJ file missing) <b>5</b> = Action aborted (SDB file missing) <b>6</b> = Action aborted (multiple CHK files found) <b>7</b> = Action aborted (multiple PRJ files found) <b>8</b> = Action aborted (multiple SDB files found) <b>9</b> = Action aborted (inconsistent file names) <b>10</b> = Action aborted (unable to make directory) <b>11</b> = Action aborted (copy failed) <b>12</b> = Action aborted (Codesys module not present)	int  read only
	<b>PLC Status</b>	Provides status of internal PLC. <b>0</b> = Run mode <b>1</b> = Stop mode <b>2</b> = Reload from file <b>3</b> = Reset Warm <b>4</b> = Reset Cold <b>5</b> = Reset Hard	int
	 PLC system variables refers to CODESYS V2 internal PLC		



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

Element	Description																																																
	<div> All variables refer to server, not to HMI Client.</div> <div><table><tr><th colspan="3">Time Variables</th></tr><tr><th>Variable Name</th><th>Description</th><th>Data Type</th></tr><tr><td>Day Of Month</td><td>Range: 1–31</td><td>int</td></tr><tr><td>Day of Week</td><td>Range: 0 = Sunday, .. , 6 = Saturday</td><td>int</td></tr><tr><td>Hour</td><td>Range: 0–23</td><td>int</td></tr><tr><td>Minute</td><td>Range: 0–59</td><td>int</td></tr><tr><td>Month</td><td>Range: 1–12</td><td>int</td></tr><tr><td>Second</td><td>Range: 0–59</td><td>int</td></tr><tr><td>System Time</td><td>The same as UTC time. It can also be set as date/time for this variable</td><td>unsignedInt</td></tr><tr><td>Year</td><td>Current Year</td><td>int</td></tr></table><div> System time expressed in UTC format</div><div><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>USB Drive FreeSpace</td><td>Available space in bytes</td><td>uint64 read only</td></tr><tr><td>USB Drive Name</td><td>Name of USB device</td><td>string read only</td></tr><tr><td>USB Drive Size</td><td>Size in bytes of the device plugged in the USB port</td><td>uint64 read only</td></tr><tr><td>USB Drive Status</td><td>0 = USB Drive unplugged 1 = USB Drive plugged</td><td>int read only</td></tr></table></div></div>	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	USB Drive Variables			Variable Name	Description	Data Type	USB Drive FreeSpace	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
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																																															
USB Drive Variables																																																	
Variable Name	Description	Data Type																																															
USB Drive FreeSpace	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																																															

Element	Description															
	<b>Version Variables</b>															
	<table><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></table>	Variable Name	Description	Data Type	<b>Main OS version</b>	Version of Main OS	string read only	<b>Runtime version</b>	Version of Runtime	string read only						
	Variable Name	Description	Data Type													
	<b>Main OS version</b>	Version of Main OS	string read only													
	<b>Runtime version</b>	Version of Runtime	string read only													
	<b>Virtual Com Switch Variables</b>															
	<table><tr><th>Variable Name</th><th>Description</th><th>Data Type</th></tr><tr><td><b>VCS status</b></td><td>Provides status of VCS service.  <b>0</b> = Service enabled <b>1</b> = Client connected in interleaved mode <b>2</b> = Client connected in exclusive mode <b>3</b> = Service disabled (default)</td><td>unsignedByte read only</td></tr><tr><td><b>VCS disable</b></td><td>Provides manual override of VCS service.  <b>0</b> = VCS service enabled <b>1</b> = VCS service disabled (default)</td><td>boolean</td></tr><tr><td><b>VCS port</b></td><td>Provides current listening TCP port on HMI by VCS service</td><td>unsignedShort</td></tr></table>	Variable Name	Description	Data Type	<b>VCS status</b>	Provides status of VCS service.  <b>0</b> = Service enabled <b>1</b> = Client connected in interleaved mode <b>2</b> = Client connected in exclusive mode <b>3</b> = Service disabled (default)	unsignedByte read only	<b>VCS disable</b>	Provides manual override of VCS service.  <b>0</b> = VCS service enabled <b>1</b> = VCS service disabled (default)	boolean	<b>VCS port</b>	Provides current listening TCP port on HMI by VCS service	unsignedShort			
	Variable Name	Description	Data Type													
	<b>VCS status</b>	Provides status of VCS service.  <b>0</b> = Service enabled <b>1</b> = Client connected in interleaved mode <b>2</b> = Client connected in exclusive mode <b>3</b> = Service disabled (default)	unsignedByte read only													
	<b>VCS disable</b>	Provides manual override of VCS service.  <b>0</b> = VCS service enabled <b>1</b> = VCS service disabled (default)	boolean													
<b>VCS port</b>	Provides current listening TCP port on HMI by VCS service	unsignedShort														
<b>Data Type</b>	<p>Each system variable has a specific data type, described in above tables.</p> <p>The following table shows the details of any data type used for system variables.</p> <table><tr><th>Data Type</th><th>Memory Space</th><th>Limits</th></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></table>	Data Type	Memory Space	Limits	<b>short</b>	16-bit data	-32768 ... 32767	<b>int</b>	32-bit data	-2.1e9 ... 2.1e9	<b>unsignedByte</b>	8-bit data	0 ... 255	<b>unsignedShort</b>	16-bit data	0 ... 65535
Data Type	Memory Space	Limits														
<b>short</b>	16-bit data	-32768 ... 32767														
<b>int</b>	32-bit data	-2.1e9 ... 2.1e9														
<b>unsignedByte</b>	8-bit data	0 ... 255														
<b>unsignedShort</b>	16-bit data	0 ... 65535														

Element	Description										
	Data Type	Memory Space	Limits								
	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									
Arraysize	<p>In case of string Tag, this property represents the maximum number of bytes available in the string Tag.</p> <p>Note: number of bytes corresponds to number of string chars if Encoding property is set to UTF-8 or Latin1 in Tag Editor.</p> <p>If Encoding property is set to UCS-2BE, UCS-2LE, UTF-16BE or UTF-16LE one char requires 2 bytes.</p>										
Conversion	<p>Conversion to be applied to the tag.</p> <div><p>Conversion</p><div><div>inv,swap2</div><div><div>Allowed</div><div>BCD AB-&gt;BA ABCD-&gt;CDAB ABCDEFGH-&gt;GHEFCBAB Inv bits</div><div><div>+</div><div>-</div><div>^</div><div>v</div></div><div><div>Configured</div><div>Inv bits ABCD-&gt;CDAB</div><div>CancelOK</div></div></div></div><p>Depending on data type selected, the list <b>Allowed</b> shows one or more conversion typs.</p><table><tr><th>Value</th><th>Description</th></tr><tr><td>Inv bits</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>Negate</td><td><b>neg</b>: Set the opposite of tag value.  <i>Example:</i> 25.36 → -25.36</td></tr><tr><td>AB → BA</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></table></div>			Value	Description	Inv bits	<b>inv</b> : Invert all the bits of the tag.  <i>Example:</i> 1001 → 0110 (in binary format) 9 → 6 (in decimal format)	Negate	<b>neg</b> : Set the opposite of tag value.  <i>Example:</i> 25.36 → -25.36	AB → BA	<b>swapnibbles</b> : Swap nibbles in a byte.  <i>Example:</i> 15D4 → 514D (in hexadecimal format) 5588 → 20813 (in decimal format)
Value	Description										
Inv bits	<b>inv</b> : Invert all the bits of the tag.  <i>Example:</i> 1001 → 0110 (in binary format) 9 → 6 (in decimal format)										
Negate	<b>neg</b> : Set the opposite of tag value.  <i>Example:</i> 25.36 → -25.36										
AB → BA	<b>swapnibbles</b> : Swap nibbles in a byte.  <i>Example:</i> 15D4 → 514D (in hexadecimal format) 5588 → 20813 (in decimal format)										

Element	Description	
	Value	Description
	<b>ABCD → CDAB</b>	<b>swap2:</b> Swap bytes in a word.  <i>Example:</i> 9ACC → CC9A (in hexadecimal format) 39628 → 52378 (in decimal format)
	<b>ABCDEFGH → GHEFCBAB</b>	<b>swap4:</b> Swap bytes in a double word.  <i>Example:</i> 32FCFF54 → 54FFFC32 (in hexadecimal format) 855441236 → 1426062386 (in decimal format)
	<b>ABC...NOP → OPM...DAB</b>	<b>swap8:</b> Swap bytes in a long word.  <i>Example:</i> 142.366 → -893553517.588905 (in decimal format) 0 10000000110 000111001011101101100100010110100001110010101100000 1 → 1 10000011100 101010100001010001011011011011001011011000010011110 1 (in binary format)
	<b>BCD</b>	<b>bcd:</b> Separate byte in two nibbles, read them as decimal (from 0 to 9)  <i>Example:</i> 23 → 17 (in decimal format) 0001 0111 = 23 0001 = 1 (first nibble) 0111 = 7 (second nibble)
Select conversion and click +. The selected item will be added to list <b>Configured</b> .  If more conversions are configured, they will be applied in order (from top to bottom of list <b>Configured</b> ).  Use the arrow buttons to order the configured conversions.		

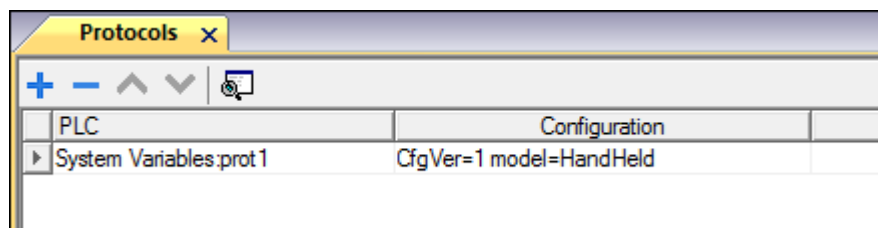
## HandHeld variables

System Variables - HandHeld protocol allows to create Tags that point to H3 device system variables.

This protocol is automatically added in a new project if H3 device is selected from project creation wizard.

### Protocol Editor Settings

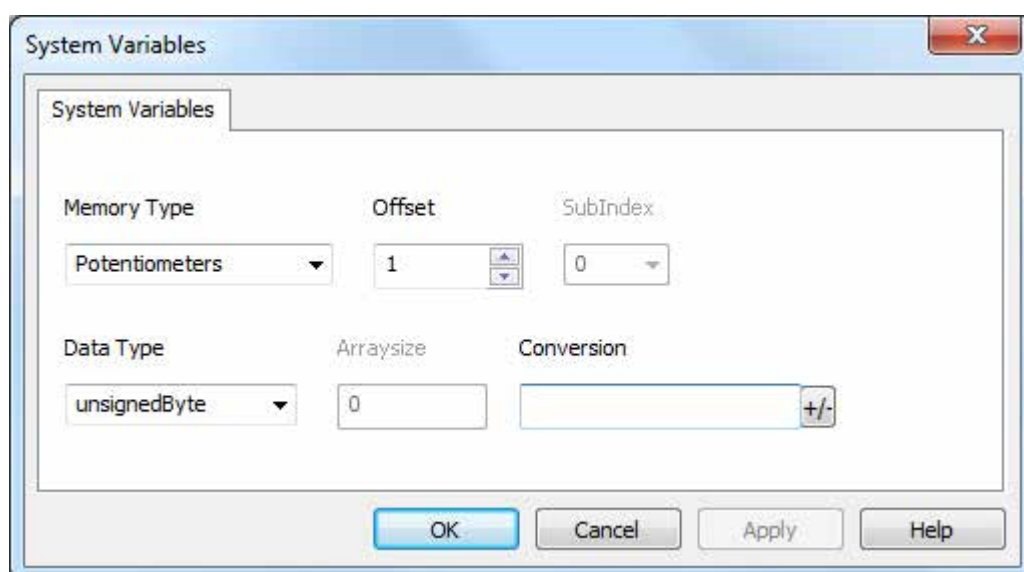
From PLC Model list of Protocol Editor dialog, select HandHeld.



## Tag Editor Settings

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

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



Element	Description		
Memory Type	Variable	Description	Data Type
	Potentiometers	Represents the value of physical potentiometer (Range: 0-255).  Offset property identifies the potentiometer:  1 = right potentiometer  2 = left potentiometer	unsignedByte  read only
	Wheel position	Represents the physical wheel absolute position.	int
	Note: Above Tags are automatically added in a new project if H3 device is selected from project creation wizard. For other memory types, refer to <i>System Variables - Default</i> chapter		
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	

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

<b>Conversion</b>	<p>Conversion to be applied to the tag.</p> <div data-bbox="284 600 1197 956" data-label="Image"> </div> <p>Depending on data type selected, the list <b>Allowed</b> shows one or more conversion types.</p> <table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td><b>Inv bits</b></td><td> <p><b>inv</b>: Invert all the bits of the tag.</p> <p><i>Example:</i>            1001 → 0110 (in binary format)            9 → 6 (in decimal format)</p> </td></tr> <tr> <td><b>Negate</b></td><td> <p><b>neg</b>: Set the opposite of tag value.</p> <p><i>Example:</i>            25.36 → -25.36</p> </td></tr> <tr> <td><b>AB → BA</b></td><td> <p><b>swapnibbles</b>: Swap nibbles in a byte.</p> <p><i>Example:</i>            15D4 → 514D (in hexadecimal format)            5588 → 20813 (in decimal format)</p> </td></tr> <tr> <td><b>ABCD → CDAB</b></td><td> <p><b>swap2</b>: Swap bytes in a word.</p> <p><i>Example:</i>            9ACC → CC9A (in hexadecimal format)            39628 → 52378 (in decimal format)</p> </td></tr> <tr> <td><b>ABCDEFGH →</b></td><td> <p><b>swap4</b>: Swap bytes in a double word.</p> <p><i>Example:</i></p> </td></tr> </table>	Value	Description	<b>Inv bits</b>	<p><b>inv</b>: Invert all the bits of the tag.</p> <p><i>Example:</i>            1001 → 0110 (in binary format)            9 → 6 (in decimal format)</p>	<b>Negate</b>	<p><b>neg</b>: Set the opposite of tag value.</p> <p><i>Example:</i>            25.36 → -25.36</p>	<b>AB → BA</b>	<p><b>swapnibbles</b>: Swap nibbles in a byte.</p> <p><i>Example:</i>            15D4 → 514D (in hexadecimal format)            5588 → 20813 (in decimal format)</p>	<b>ABCD → CDAB</b>	<p><b>swap2</b>: Swap bytes in a word.</p> <p><i>Example:</i>            9ACC → CC9A (in hexadecimal format)            39628 → 52378 (in decimal format)</p>	<b>ABCDEFGH →</b>	<p><b>swap4</b>: Swap bytes in a double word.</p> <p><i>Example:</i></p>
Value	Description												
<b>Inv bits</b>	<p><b>inv</b>: Invert all the bits of the tag.</p> <p><i>Example:</i>            1001 → 0110 (in binary format)            9 → 6 (in decimal format)</p>												
<b>Negate</b>	<p><b>neg</b>: Set the opposite of tag value.</p> <p><i>Example:</i>            25.36 → -25.36</p>												
<b>AB → BA</b>	<p><b>swapnibbles</b>: Swap nibbles in a byte.</p> <p><i>Example:</i>            15D4 → 514D (in hexadecimal format)            5588 → 20813 (in decimal format)</p>												
<b>ABCD → CDAB</b>	<p><b>swap2</b>: Swap bytes in a word.</p> <p><i>Example:</i>            9ACC → CC9A (in hexadecimal format)            39628 → 52378 (in decimal format)</p>												
<b>ABCDEFGH →</b>	<p><b>swap4</b>: Swap bytes in a double word.</p> <p><i>Example:</i></p>												



Element	Description	
	Value	Description
	<b>GHEFCDAB</b>	32FCFF54 → 54FFFC32 (in hexadecimal format) 855441236 → 1426062386 (in decimal format)
	<b>ABC...NOP -&gt; OPM...DAB</b>	<b>swap8</b> : Swap bytes in a long word.  Example: 142.366 → -893553517.588905 (in decimal format) 0 10000000110 000111001011101101100100010110100001110010101100000 1 → 1 10000011100 101010100001010001011011011001011011000010011110 1 (in binary format)
	<b>BCD</b>	<b>bcd</b> : Separate byte in two nibbles, read them as decimal (from 0 to 9)  <i>Example:</i> 23 → 17 (in decimal format) 0001 0111 = 23 0001 = 1 (first nibble) 0111 = 7 (second nibble)

Select conversion and click +. The selected item will be added to list **Configured**.

If more conversions are configured, they will be applied in order (from top to bottom of list **Configured**).

Use the arrow buttons to order the configured conversions.

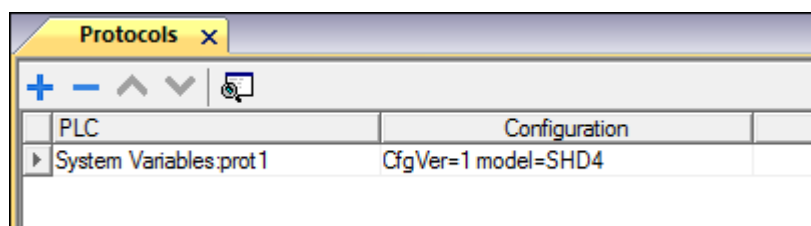
## HandHeld - wireless variables

System Variables - HandHeld - wireless protocol allows to create Tags that point to H4 device system variables.

This protocol is automatically added in a new project if H4 device is selected from project creation wizard.

### Protocol Editor Settings

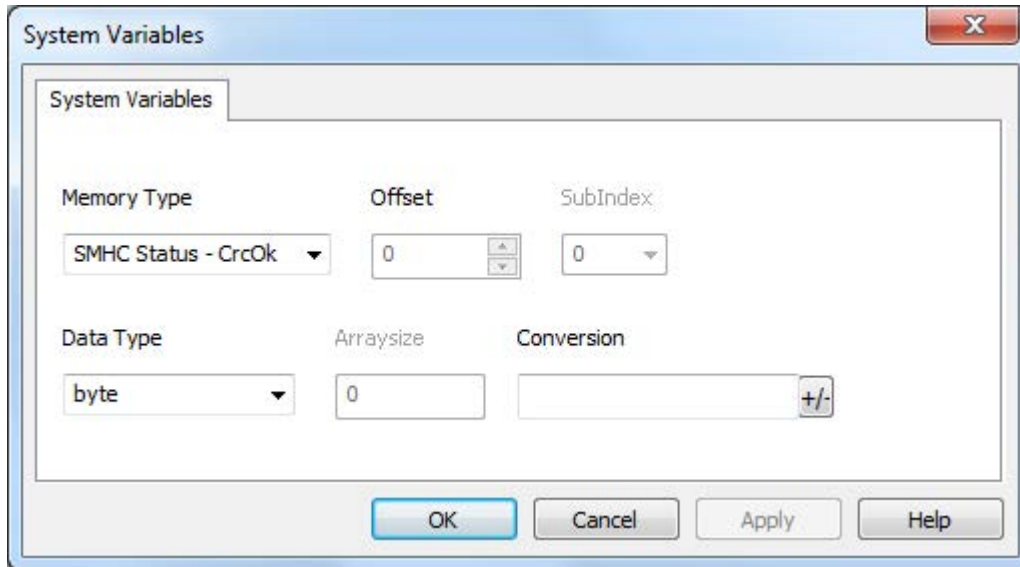
From PLC Model list of Protocol Editor dialog, select HandHeld - wireless.



## 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.



The image shows a 'System Variables' dialog box with a title bar containing a close button (X). The dialog has a tab labeled 'System Variables'. Inside, there are two rows of settings. The first row has 'Memory Type' (a dropdown menu showing 'SMHC Status - CrcOk'), 'Offset' (a numeric input field with '0' and up/down arrows), and 'SubIndex' (a dropdown menu showing '0'). The second row has 'Data Type' (a dropdown menu showing 'byte'), 'Arraysize' (a numeric input field with '0'), and 'Conversion' (an empty field with a '+/-' button). At the bottom, there are four buttons: 'OK', 'Cancel', 'Apply', and 'Help'.

Element	Description		
Memory Type	Variable	Description	Data Type
	<b>Battery Charging Status</b>	Value indicating battery charging status. <b>0</b> = power by battery <b>1</b> = external power, battery in charge <b>2</b> = external power, battery charging complete	unsignedByte read only
	<b>Battery Level</b>	Value indicating the battery charge percentage. (Range: 0-100). <b>0</b> = battery low <b>100</b> = battery fully charged	unsignedByte read only
	<b>Bluetooth Coupled</b>	Coupling status between H4 and remote safety module. <b>0</b> = not coupled <b>1</b> = coupled	boolean read only
	<b>Bluetooth Signal Strength</b>	Bluetooth RSSI level. (Range: 0-100). <b>0</b> = minimum quality <b>100</b> = maximum quality	byte read only
	<b>Emergency</b>	Status of the Emergency Stop button. <b>0</b> = button released <b>1</b> = button pressed	boolean read only
	<b>Selector Position</b>	Position of the selector switch (Range: 0-15)	unsignedByte read only
	<b>Token ID</b>	TOKEN ID value if token is present/connected, otherwise 0. 0 is returned also in case of error.	uint64 read only
	<b>Token Present</b>	Presence of the token. <b>0</b> = token not inserted <b>1</b> = token inserted	boolean read only
	<b>Token Valid</b>	Status of the token. <b>0</b> = token not valid	boolean read only

Element	Description		
	Variable	Description	Data Type
		<b>1</b> = token inserted and valid	
	<b>Potentiometers</b>	Represents the value of physical potentiometer (Range: 0-255).  Offset property identifies the potentiometer: <b>1</b> = right potentiometer <b>2</b> = left potentiometer	unsignedByte  read only
	<b>Vibration</b>	Value indicating vibration status.  <b>0</b> = vibration disabled <b>1</b> = vibration enabled	boolean  read only
	<b>Wheel position</b>	Represents the physical wheel absolute position.	int
	<b>WIFI Signal Strength</b>	Value indicating the level of the current Wi-Fi network (Range: 0-100).  <b>0</b> = no signal <b>100</b> = max signal level	byte  read only
	<b>SMHC Configuration</b>	SMCH security module system variables	-
	<b>SMHC Status</b>		
	<b>SMWL Configuration</b>	SMWL security module system variables	-
	<b>SMWL Status</b>		
	Note: Above Tags are automatically added in a new project if H4 device is selected from project creation wizard. For other memory types, refer to <i>System Variables - Default</i> chapter		
<b>Data Type</b>	Each system variable has a specific data type, described in above tables.  The following table shows the details of any data type used for system variables.		
	Data Type	Memory Space	Limits
	<b>boolean</b>	1-bit data	0 ... 1
	<b>byte</b>	8-bit data	-128 ... 127

Element	Description																								
	<table><tr><th>Data Type</th><th>Memory Space</th><th>Limits</th></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>uint64</td><td>64-bit data</td><td>0 ... 1.8e19</td></tr><tr><td>string</td><td colspan="2">Array of elements containing character code defined by selected encoding</td></tr></table>	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	
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																								
Arraysize	<p>In case of string Tag, this property represents the maximum number of bytes available in the string Tag.</p> <p>Note: number of bytes corresponds to number of string chars if Encoding property is set to UTF-8 or Latin1 in Tag Editor.</p> <p>If Encoding property is set to UCS-2BE, UCS-2LE, UTF-16BE or UTF-16LE one char requires 2 bytes.</p>																								
Conversion	<p>Conversion to be applied to the tag.</p> <div><p>Conversion</p><div><div>inv,swap2</div><div><div>Allowed</div><div>BCD AB-&gt;BA ABCD-&gt;CDAB ABCDEFGH-&gt;GHEFCADB Inv bits</div><div><div>+</div><div>-</div><div>^</div><div>v</div></div><div><div>Configured</div><div>Inv bits ABCD-&gt;CDAB</div><div>CancelOK</div></div></div></div><p>Depending on data type selected, the list <b>Allowed</b> shows one or more conversion typs.</p><table><tr><th>Value</th><th>Description</th></tr><tr><td>Inv bits</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>Negate</td><td><b>neg</b>: Set the opposite of tag value.</td></tr></table></div>	Value	Description	Inv bits	<b>inv</b> : Invert all the bits of the tag.  <i>Example:</i> 1001 → 0110 (in binary format) 9 → 6 (in decimal format)	Negate	<b>neg</b> : Set the opposite of tag value.																		
Value	Description																								
Inv bits	<b>inv</b> : Invert all the bits of the tag.  <i>Example:</i> 1001 → 0110 (in binary format) 9 → 6 (in decimal format)																								
Negate	<b>neg</b> : Set the opposite of tag value.																								

Element	Description	
	Value	Description
		<i>Example:</i> 25.36 → -25.36
	<b>AB → BA</b>	<b>swapnibbles:</b> Swap nibbles in a byte.  <i>Example:</i> 15D4 → 514D (in hexadecimal format) 5588 → 20813 (in decimal format)
	<b>ABCD → CDAB</b>	<b>swap2:</b> Swap bytes in a word.  <i>Example:</i> 9ACC → CC9A (in hexadecimal format) 39628 → 52378 (in decimal format)
	<b>ABCDEFGH → GHEFC DAB</b>	<b>swap4:</b> Swap bytes in a double word.  <i>Example:</i> 32FCFF54 → 54FFFC32 (in hexadecimal format) 855441236 → 1426062386 (in decimal format)
	<b>ABC...NOP → OPM...DAB</b>	<b>swap8:</b> Swap bytes in a long word.  <i>Example:</i> 142.366 → -893553517.588905 (in decimal format) 0 10000000110 000111001011101101100100010110100001110010101100000 1 → 1 10000011100 101010100001010001011011011001011011000010011110 1 (in binary format)
	<b>BCD</b>	<b>bcd:</b> Separate byte in two nibbles, read them as decimal (from 0 to 9)  <i>Example:</i> 23 → 17 (in decimal format) 0001 0111 = 23 0001 = 1 (first nibble) 0111 = 7 (second nibble)

Select conversion and click +. The selected item will be added to list **Configured**.

If more conversions are configured, they will be applied in order (from top to bottom of list **Configured**).

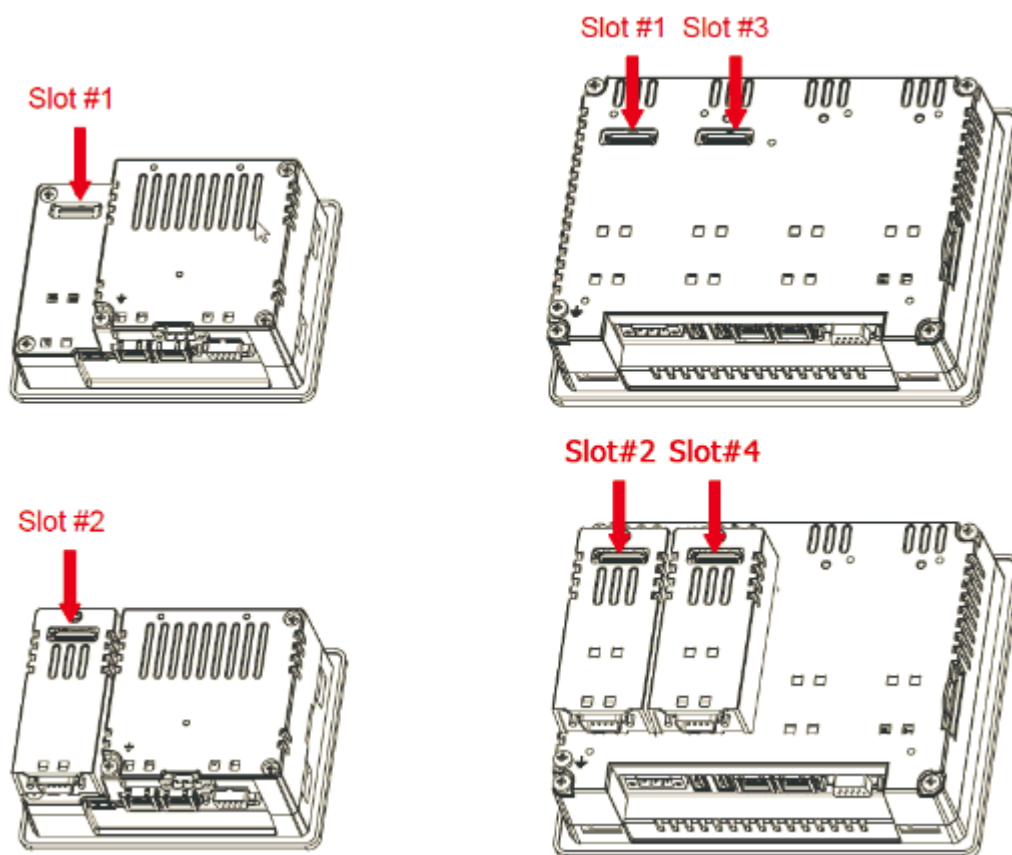
Use the arrow buttons to order the configured conversions.

## 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

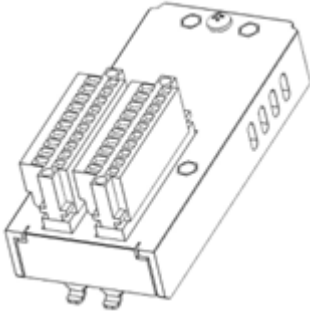
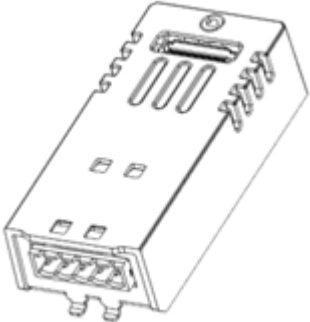
Most HMI offer 1 or 2 slots for connecting optional plug-in modules. Slot numbers are referred in the programming software for configuration of plug-in modules. Numbering of plug-in slots is shown in figure.



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.

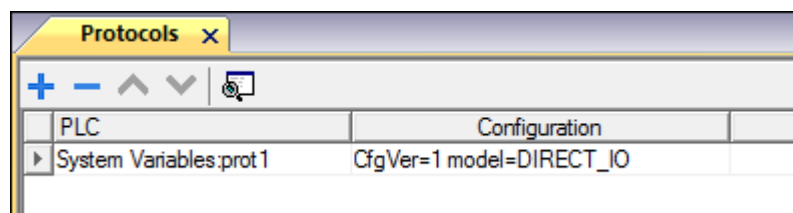
### Plug-in I/O Modules details

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

Module	I/O configuration	Image
8 Digital Inputs / 6 Digital Outputs / 1 Relay Output	<ul style="list-style-type: none"> <li>8 optically isolated digital inputs</li> <li>6 optically isolated digital outputs</li> <li>1 relay output</li> </ul>	
2 Relay Outputs	<ul style="list-style-type: none"> <li>2 relay output</li> </ul>	

## Protocol Editor Settings

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

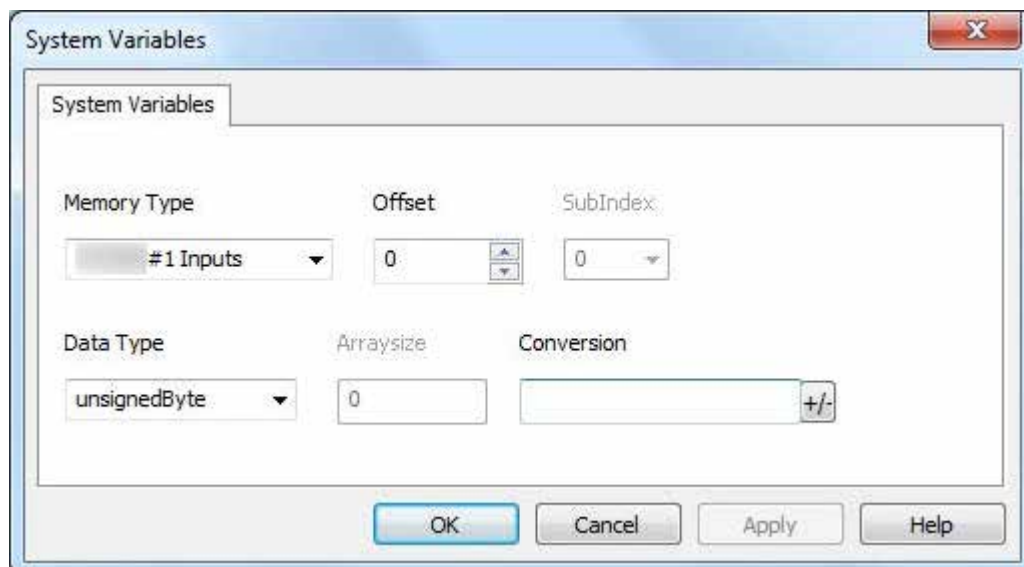


## Tag Editor Settings

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.







The screenshot shows a dialog box titled "System Variables" with a standard Windows-style title bar (minimize, maximize, close buttons). The dialog contains a tab labeled "System Variables". Inside the tab, there are two rows of controls. The first row has three controls: "Memory Type" (a dropdown menu showing "#1 Inputs"), "Offset" (a numeric input field showing "0" with up/down arrow buttons), and "SubIndex" (a dropdown menu showing "0"). The second row has three controls: "Data Type" (a dropdown menu showing "unsignedByte"), "Arraysize" (a numeric input field showing "0"), and "Conversion" (a text input field with a "+/-" button to its right). At the bottom of the dialog, there are four buttons: "OK", "Cancel", "Apply", and "Help".

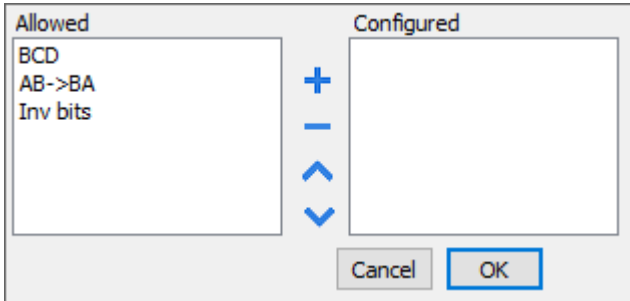
Memory Type	Offset	SubIndex
#1 Inputs	0	0

Data Type	Arraysize	Conversion
unsignedByte	0	

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

Element	Description																	
	Memory Type	Range	Description															
	Outputs	0 - 6	Refers to <b>8 Digital Inputs / 6 Digital Outputs / 1 Relay Output</b> I/O module															
			Bit #	Description	0	Digital Output 1	1	Digital Output 2	2	Digital Output 3	3	Digital Output 4	4	Digital Output 5	5	Digital Output 6	6	Relay Output
			Bit #	Description														
			0	Digital Output 1														
			1	Digital Output 2														
			2	Digital Output 3														
			3	Digital Output 4														
			4	Digital Output 5														
			5	Digital Output 6														
6	Relay Output																	
0 - 1	Refers to <b>2 Relay Outputs</b> I/O module	<table><tr><th>Bit #</th><th>Channel #</th></tr><tr><td>0</td><td>Relay Output 1</td></tr><tr><td>1</td><td>Relay Output 2</td></tr></table>	Bit #	Channel #	0	Relay Output 1	1	Relay Output 2										
		Bit #	Channel #															
		0	Relay Output 1															
		1	Relay Output 2															
Data Type	Data Type	Memory Space	Limits															
	unsignedByte	8-bit data	0 ... 255															
	boolean[]	1-bit data array	0 ... 1 (for each element)															
	<div> 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.</div> <div> 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<pre>13 14 function BtnStd1_ButtonWgt_onMouseClicked(me, eventInfo) { 15 16     project.setTag("Outputs",1,2); //tagName , tagValue, index 17 }</pre></div>																	
Arraysi ze	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																	

Element	Description								
	Variable<convert to text and insert new text here>								
<b>Conversion</b>	<p>Conversion to be applied to the tag.</p>  <p>Depending on data type selected, the <b>Allowed</b> list shows one or more conversions, listed below.</p> <table border="1"> <thead> <tr> <th>Value</th><th>Description</th></tr> </thead> <tbody> <tr> <td><b>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 -&gt; BA</b></td><td>           Swap nibbles of a byte.   <i>Example:</i>            15D4 → 514D (in hexadecimal format)            5588 → 20813 (in decimal format)         </td></tr> <tr> <td><b>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> </tbody> </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>	Value	Description	<b>BCD</b>	Separate the byte in two nibbles, and reads them as decimal (from 0 to 9)  <i>Example:</i> 23 → 17 (in decimal format) 0001 0111 = 23 0001 = 1 (first nibble) 0111 = 7 (second nibble)	<b>AB -&gt; BA</b>	Swap nibbles of a byte.  <i>Example:</i> 15D4 → 514D (in hexadecimal format) 5588 → 20813 (in decimal format)	<b>Inv bits</b>	Invert all the bits of the tag.  <i>Example:</i> 1001 → 0110 (in binary format) 9 → 6 (in decimal format)
Value	Description								
<b>BCD</b>	Separate the byte in two nibbles, and reads them as decimal (from 0 to 9)  <i>Example:</i> 23 → 17 (in decimal format) 0001 0111 = 23 0001 = 1 (first nibble) 0111 = 7 (second nibble)								
<b>AB -&gt; BA</b>	Swap nibbles of a byte.  <i>Example:</i> 15D4 → 514D (in hexadecimal format) 5588 → 20813 (in decimal format)								
<b>Inv bits</b>	Invert all the bits of the tag.  <i>Example:</i> 1001 → 0110 (in binary format) 9 → 6 (in decimal format)								

## PLCM09 variables

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

## Protocol Editor Settings

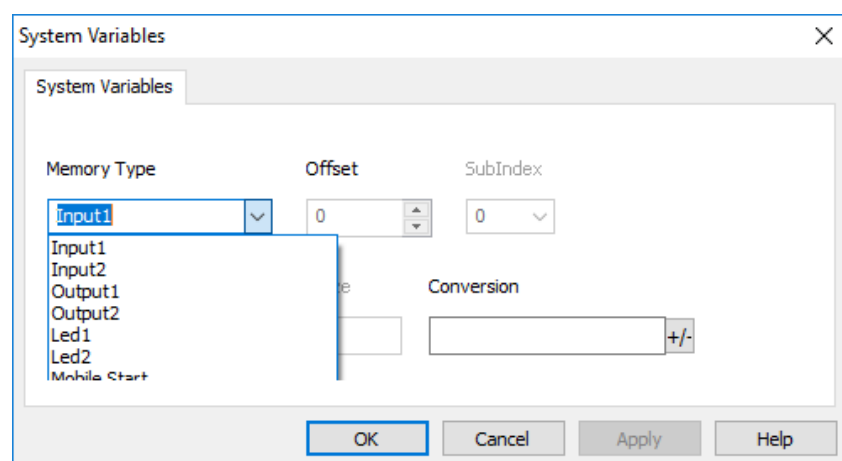
Path: **ProjectView** > **Protocols**


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




## Tag Editor Settings

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

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



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

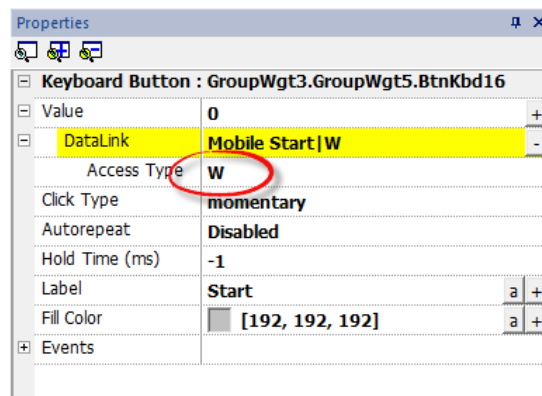
Element	Description	Data type
	 you rewrite it.	
<b>Mobile Stop</b>	<p>Write 1 inside this tag to force the wireless mode to stop.</p>  This is a write only variable, command will executed any time you rewrite it.	boolean Write Only
<b>Mobile Restart</b>	<p>Write 1 inside this tag to force the wireless mode to restart.</p>  This is a write only variable, command will executed any time you rewrite it.	boolean Write Only
<b>Mobile State</b>	<p>Mobile connection state</p> <ul style="list-style-type: none"> <li>0 IDLE</li> <li>1 STARTING</li> <li>10 RUNNING</li> <li>100 CONNECTING</li> <li>200 CONNECTED</li> <li>300 STOPPING</li> <li>-1 GENERICERROR</li> <li>-10 SYSTEMERROR</li> <li>-100 MODEMNOTFOUND</li> <li>-101 MODEMBUSY</li> <li>-110 MODEMCOMM</li> <li>-120 MODEMTIMEOUT</li> <li>-130 MODEMERROR</li> <li>-200 SIMMISSING</li> <li>-300 PINREQUIRED</li> <li>-301 NEWPINREQUIRED</li> <li>-310 PINERROR</li> </ul>	int Read Only

Element	Description	Data type
	-320 PUKREQUIRED -330 PUKERROR -400 ROAMINGBLOCKED -500 BADCREDENTIALS	
<b>Mobile Signal</b>	Mobile signal quality (0-100)	byte Read Only
<b>Mobile Operator</b>	Mobile operator name (e.g. 'Vodafone')	string[8] Read Only
<b>Mobile Access Technology</b>	Mobile access technology -1 N/A 0 GSM (2G) 2 UTRAN (2G) 3 GSM W/EGPRS (2G) 4 UTRAN W/HSDPA (3G) 5 UTRAN W/HSUPA (3G) 6 UTRAN W/HSDPA and HSUPA (3G)	int Read Only
<b>Mobile Registration Status</b>	Mobile registration status -1 N/A 0 Not registered. Wireless Modem is not currently searching a new operator to register. 1 Registered on home network. 2 Not registered. Wireless Modem is currently searching a new operator to register. 3 Registration denied. 4 Unknown 5 Registered on roaming	int Read Only

Element	Description	Data type
<b>Mobile RX/TX</b>	Number of received/transmitted bytes	unsignedInt[2] Read Only
<b>Mobile Start Time (Sec)</b>	When mobile connection was started (in seconds since epoch)	unsignedInt Read Only



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



## JavaScript (Mobile Connection State)

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

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

# Services variables

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

## 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.



The screenshot shows the 'System Variables' dialog box. The 'Memory Type' dropdown menu is open, displaying a list of variables: 'Vnc Start', 'Vnc Stop', 'Vnc State', 'Vnc Restart', and 'boolean'. The 'Offset' is set to 0, 'SubIndex' is 0, 'Arraysize' is 0, and 'Conversion' is set to '+/-'. The 'OK' button is highlighted with a blue border.

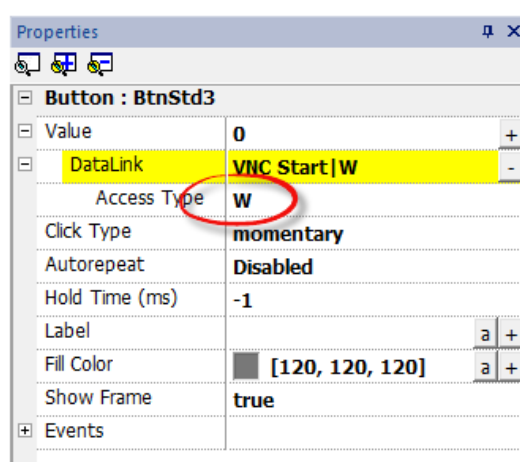
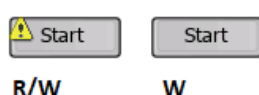
VNC status variables are supported only from Linux devices with BSP version 1.0.344 or greater. (See [x"HMI devices capabilities" on page 567x](#))

Element	Description	Data type
<b>VNC Start</b>	Write 1 inside this tag to force the VNC server to start.  This is a write only variable, command will executed any time you rewrite it.	boolean Write Only
<b>VNC Stop</b>	Write 1 inside this tag to force the VNC server to stop.  This is a write only variable, command will executed any time you rewrite it.	boolean Write Only
<b>VNC Restart</b>	Write 1 inside this tag to force the VNC server to restart.  This is a write only variable, command will executed any time you rewrite it.	boolean Write Only
<b>VNC State</b>	VNC server state  0 IDLE 1 STARTING 10 RUNNING 100 CONNECTING 200 CONNECTED -1 ERROR	int Read Only

Element	Description	Data type
<b>Cloud Start</b>	Write 1 inside this tag to force the cloud connection to start.  This is a write only variable, command will executed any time you rewrite it.	boolean Write Only
<b>Cloud Stop</b>	Write 1 inside this tag to force the cloud connection to stop.  This is a write only variable, command will executed any time you rewrite it.	boolean Write Only
<b>Cloud Restart</b>	Write 1 inside this tag to force the cloud connection to restart.  This is a write only variable, command will executed any time you rewrite it.	boolean Write Only
<b>Cloud State</b>	Cloud connection state  0 IDLE 1 STARTING 10 RUNNING 100 CONNECTING 200 CONNECTED -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.



## 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.

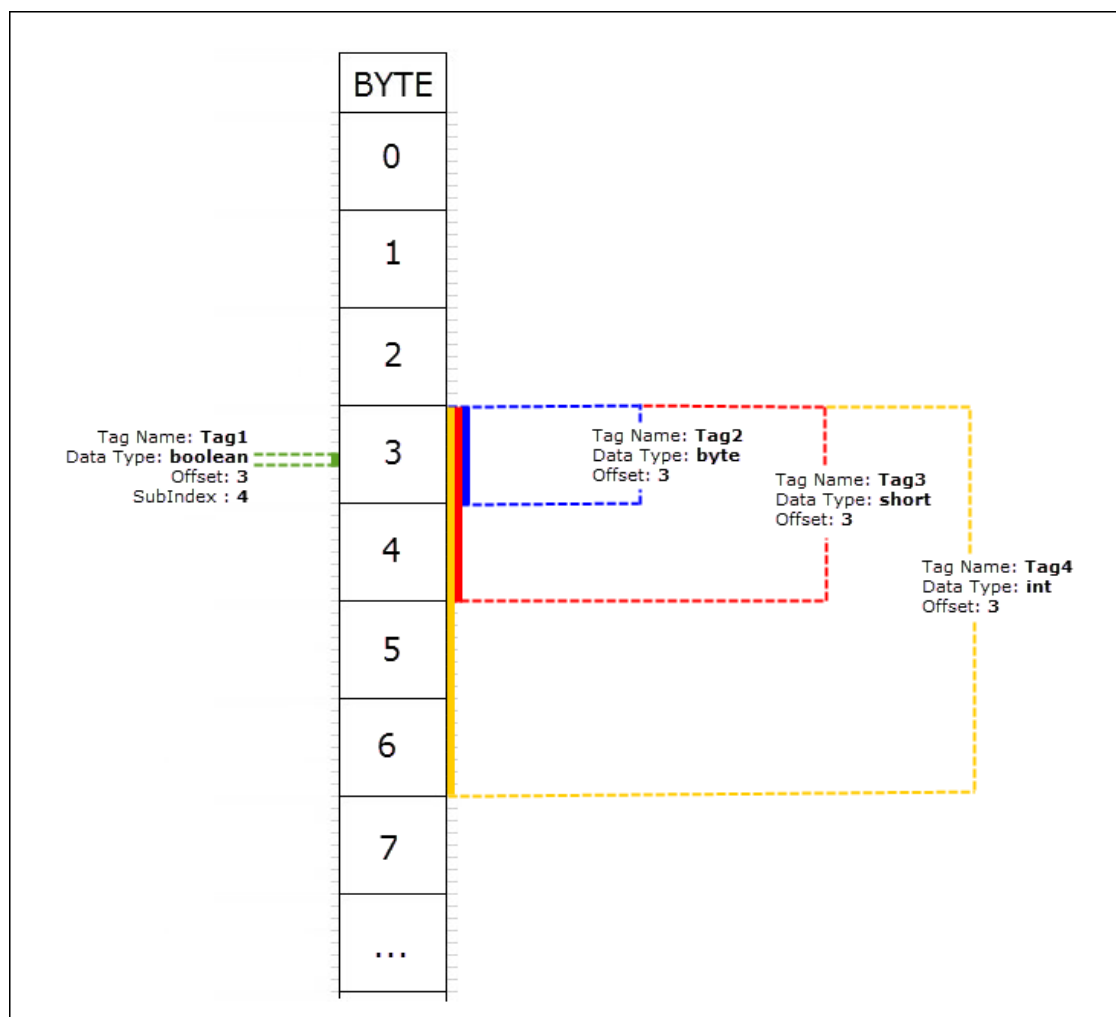
The physical support for retentive memory is based on FRAM technology.



**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.

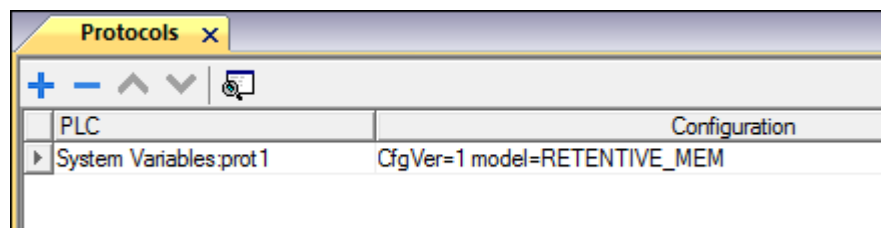


### Retentive memory vs. recipes storage

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

## Protocol Editor Settings

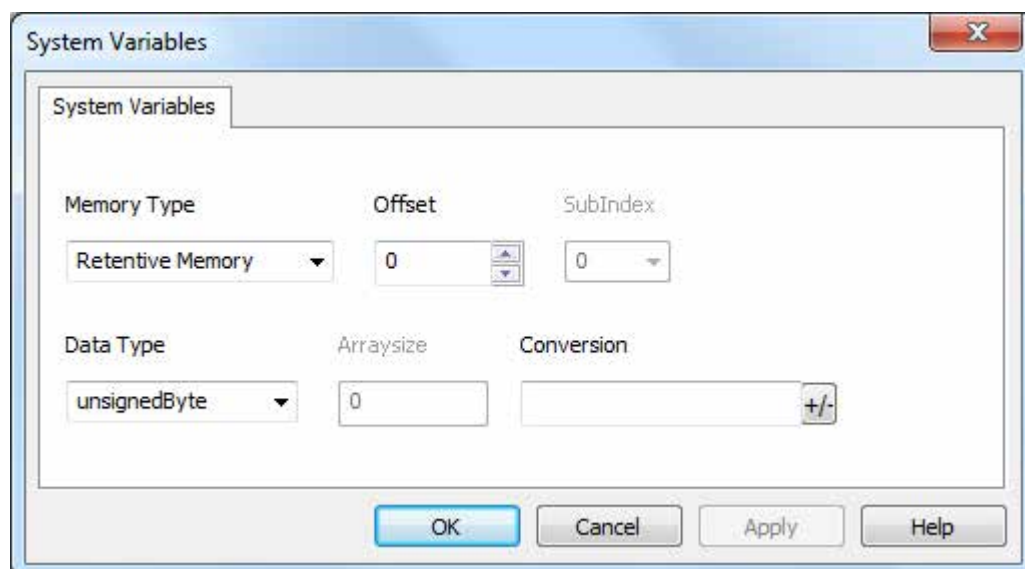
From PLC Model list of Protocol Editor dialog, select Retentive Memory.




## Tag Editor Settings

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

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



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

Element	Description		
	Data Type	Memory Space	Limits
	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	
	binary	Arbitrary binary data	
<div> Note: to define arrays. select one of Data Type format followed by square brackets like "byte []", "short[]"...</div>			
Arraysizes	<div><ul style="list-style-type: none"><li>• In case of array tag, this property represents the number of array elements.</li><li>• In case of string tag, this property represents the maximum number of bytes available in the string tag.</li></ul><p>Note: number of bytes corresponds to number of string characters if Encoding property is set to UTF-8 or Latin1 in Tag Editor. If Encoding property is set to UCS-2BE, UCS-2LE, UTF-16BE or UTF-16LE one character requires 2 bytes.</p></div>		
Conversion	<div>Conversion to be applied to the tag.</div> <div><div>Conversion</div><div>inv,swap2</div><div><div>Allowed</div><div>BCD AB-&gt;BA ABCD-&gt;CDAB ABCDEFGH-&gt;GHEFCADB Inv bits</div><div><div>+</div><div>-</div><div>^</div><div>v</div></div><div><div>Configured</div><div>Inv bits ABCD-&gt;CDAB</div><div>CancelOK</div></div></div></div>		

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

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

## Cleaning Retentive Memory

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



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

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



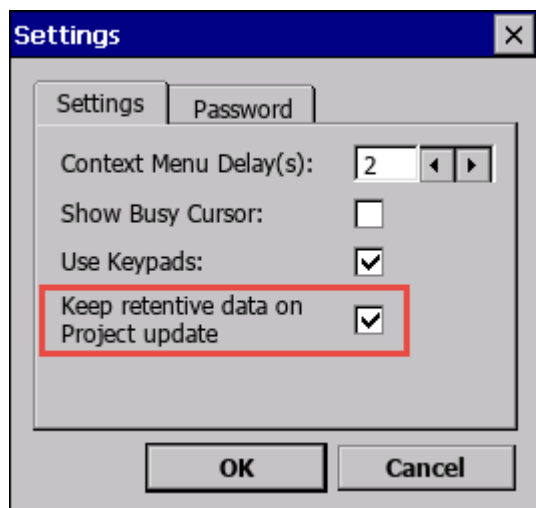
JavaScript interface for this action is:

`project.clearRetentiveMemory();`

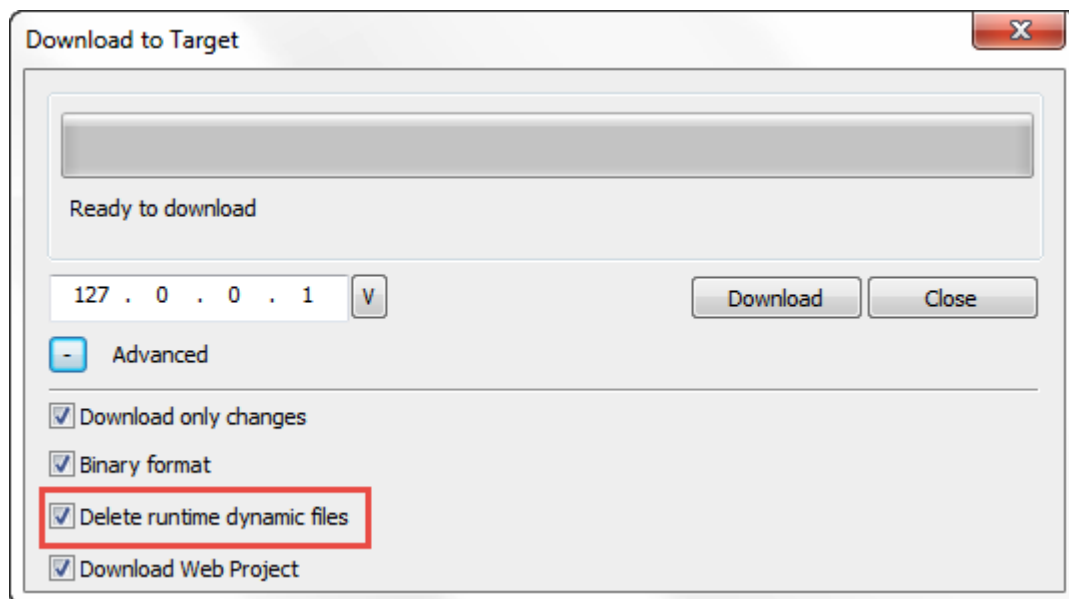
## Preserving Retentive Memory at project download

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

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

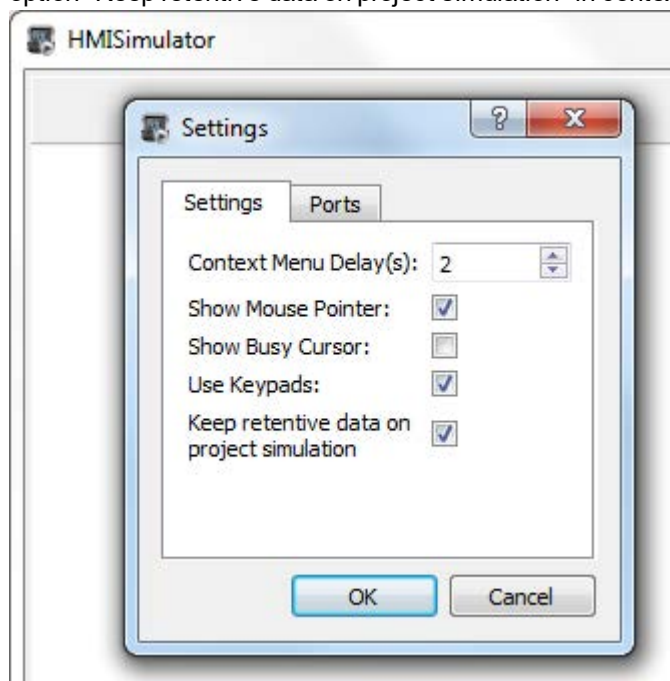


This setting will be ignored if **Delete runtime dynamic files** option is selected from *Download to Target* window.



## Preserving Retentive Memory in Simulator

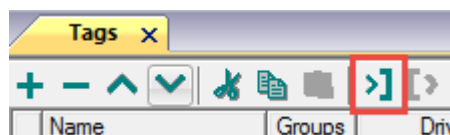
Simulator of JMobile Studio supports the retentive memory. To enable retentive memory during project simulation use the option "Keep retentive data on project simulation" in context menu.



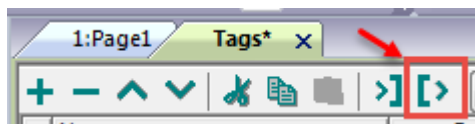
## Tag Import

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.

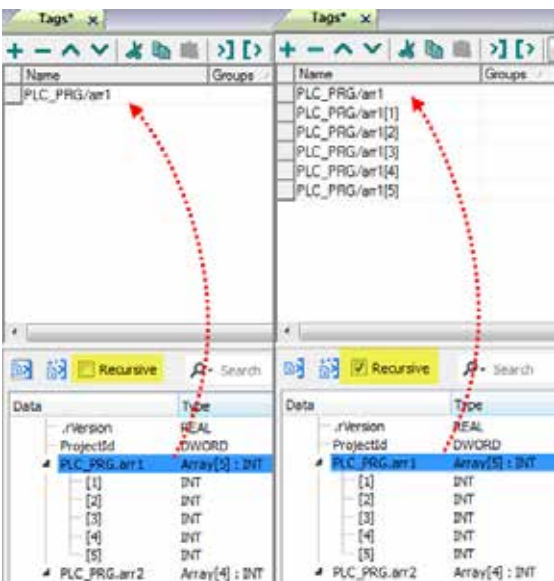


Once the importer has been selected, locate the symbol file and click **Open**.

Tags included in the symbol file are listed in the tag dictionary. The tag dictionary is displayed at the bottom of the screen.



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

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

# 11 Actions

---

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

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

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

---


<b>Alarm actions</b> .....	<b>160</b>
<b>Database actions</b> .....	<b>161</b>
<b>Event actions</b> .....	<b>164</b>
<b>MultiLanguage actions</b> .....	<b>164</b>
<b>Keyboard actions</b> .....	<b>165</b>
<b>Media Player actions</b> .....	<b>167</b>
<b>FTP actions</b> .....	<b>167</b>
<b>Page actions</b> .....	<b>170</b>
<b>Print actions</b> .....	<b>177</b>
<b>Recipe actions</b> .....	<b>178</b>
<b>Remote Client actions</b> .....	<b>183</b>
<b>System actions</b> .....	<b>184</b>
<b>Tag actions</b> .....	<b>196</b>
<b>Trend actions</b> .....	<b>198</b>
<b>Text Editor actions</b> .....	<b>202</b>
<b>User management actions</b> .....	<b>202</b>
<b>Widget actions</b> .....	<b>205</b>

# Alarm actions

Mainly used to acknowledge or reset alarms.


## SelectAllAlarms

Selects all alarms.

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

## SelectAlarm

Select a specif alarm.

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

## AckAlarm

Acknowledge a specific alarm or all selected alarms.

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

Acknowledges selected alarms.

## ResetAlarm

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

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

## EnableAlarms

Enable or disable a specific alarm or all selected alarms.

Disabled alarms will not generate alarm events.

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

## Database actions

### DBInit



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

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

Action Properties

<b>DBInit</b>	
Link Name	<b>myRemoteDB</b>
Custom SQL query	
<b>Link Name</b> Database link name	

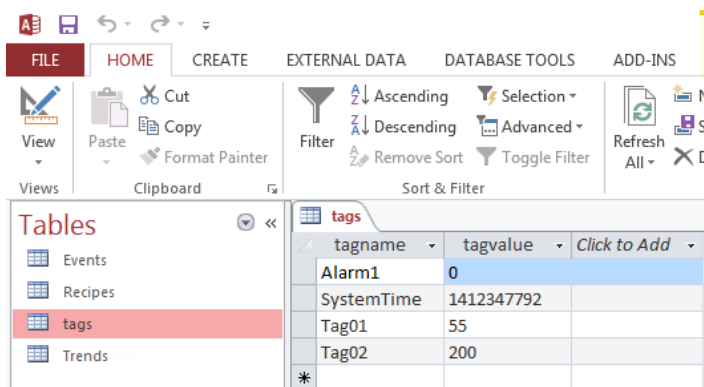
Use **Custom SQL query** parameter to define the pages to be created. Leave empty to generate default table names



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

## JavaScript Interface

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



## DBWriteTags, DBReadTags

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

Action Properties

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

Action Properties

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

## JavaScript Interface

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

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

## DBWriteGroups, DBReadGroups

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

Action Properties

DBWriteGroups	
Link Name	myRemoteDB
Custom SQL query	
Group names	Group1
Link Name Database link name	

Action Properties

DBReadGroups	
Link Name	myRemoteDB
Custom SQL query	
Group names	Group1
Link Name Database link name	

## JavaScript Interface

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

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

## DBWriteTrend

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

Action Properties

DBWriteTrends	
Link Name	myRemoteDB
Custom SQL query	
Trend names	Trend1
Duration	10 min
Link Name	
Database link name	

## JavaScript Interface

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

## DBWriteEvents

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

Action Properties

DBWriteEvents	
Link Name	myRemoteDB
Custom SQL query	
Buffer	AlarmBuffer1
Duration	1 hour
Buffer	
Select Event buffer	

Action Properties

DBWriteEvents	
Link Name	myRemoteDB
Custom SQL query	
Buffer	AuditTrail
Duration	1 hour
Buffer	
Select Event buffer	

## JavaScript Interface

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

## DBWriteRecipes, DBReadRecipes

Transfer the recipe data to/from the remote database.

Action Properties

DBWriteRecipes

Link Name	myRemoteDB
Custom SQL query	
Recipe names	Recipe1 +

**Recipe names**  
Recipe names seperated by semicolon(;)

Action Properties

DBReadRecipes

Link Name	myRemoteDB
Custom SQL query	
Recipe names	Recipe1 +

**Recipe names**  
Recipe names seperated by semicolon(;)

## JavaScript Interface

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

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

## DBResetErrors

Reset all the three status variables of the selected database link. ["Database variables" on page 100](#).

Action Properties

DBResetErrors

Link Name	myRemoteDB
-----------	------------

**Link Name**  
Database link name

## JavaScript Interface

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

# 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

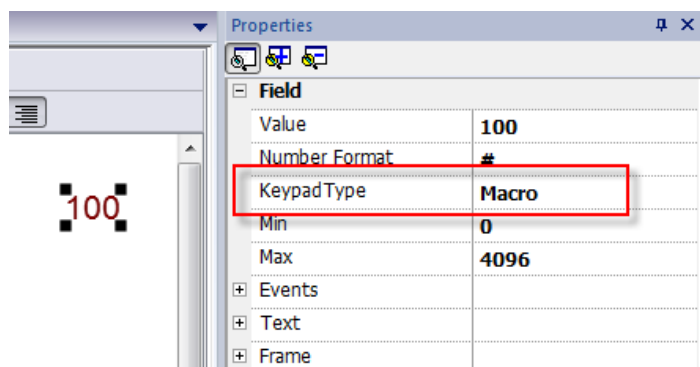
Sets the language used. The selected language will be applied at run time to all applicable widgets.

## 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**.

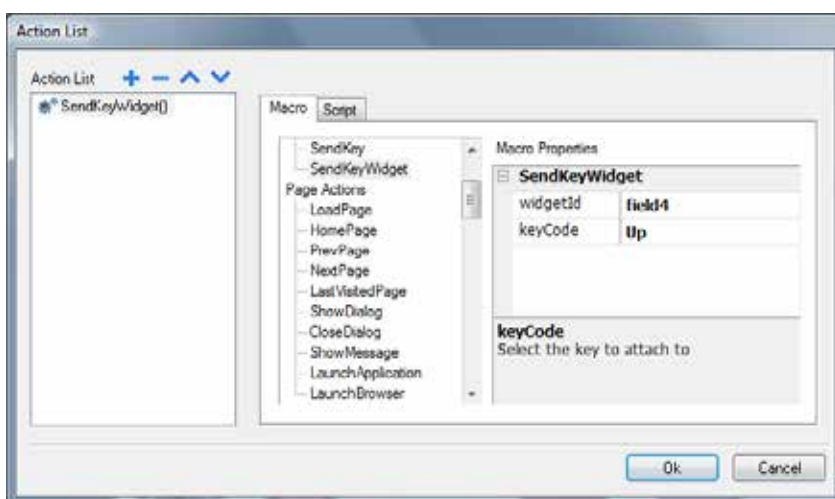
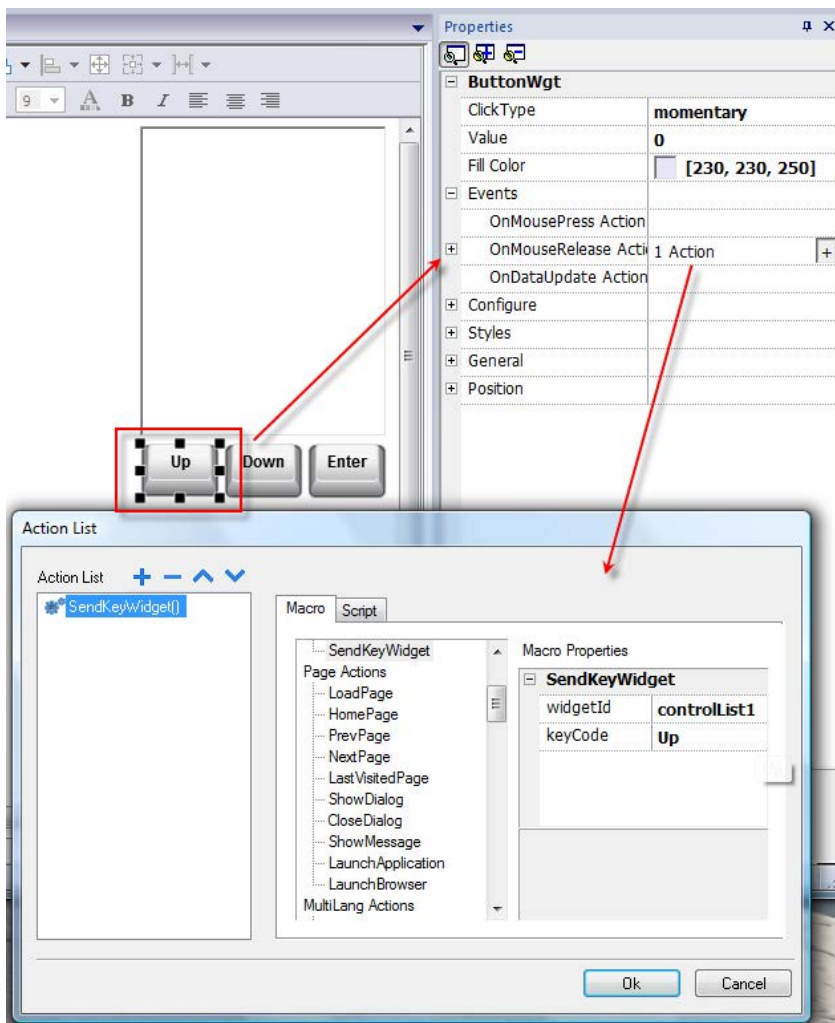


### SendKeyWidget

Sends one character to a specific widget.

#### Example

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



## ShowKeyPad

Shows the default operating system touch keypad.

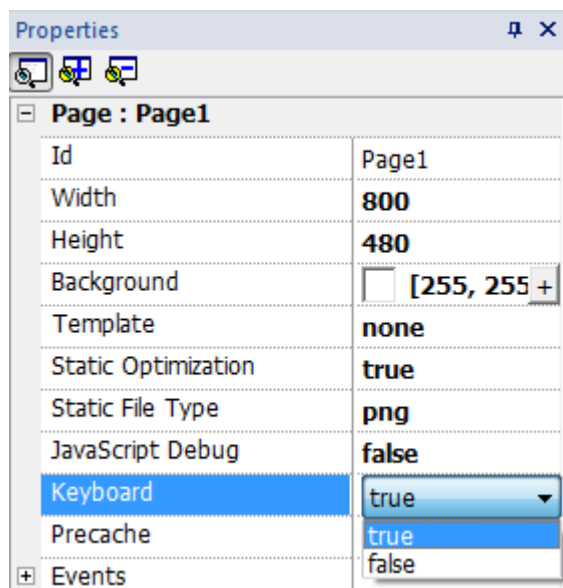


Note: might not be supported by all operating systems.

## 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.



## Media Player actions

Interact with the Media Player widget at run time.

Action	Description
<b>PlayMedia</b>	Starts playing the video.
<b>StopMedia</b>	Stops the video.
<b>ReloadMedia</b>	Restarts video from the beginning.
<b>PauseMedia</b>	Pauses the video.
<b>BrowseMedia</b>	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
<b>FtpConfig</b>	Configuring the FTP parameters
<b>FtpRemoteFileName</b>	File name on the remote FTP server to download (source)
<b>FtpLocalFileName</b>	File name on local HMI device (destination)

## ftpPUT

Upload files to a remote FTP server

Parameter	Description
<b>FtpConfig</b>	Configuring the FTP parameters
<b>FtpLocalFileName</b>	File name on local HMI device (source)
<b>FtpRemoteFileName</b>	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 104](#) for details).

## FTP Server Configuration

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

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

Click + to add more FTP servers configuration.



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

## FTP JavaScript Interface

### ftpConfig

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

Set the FTP parameters to use on next FTP calls

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

### ftpGET

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

Download files from a remote FTP server

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

### ftpPUT

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

Upload files to a remote FTP server

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

Example:

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

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

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

## Page actions

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

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

### LoadPage

Go to the selected page of the project.

### HomePage

Go to the home page.

You can set the home page in the **Behavior** section of the **Project Widget**, see ["Project" on page 78](#)

### PrevPage

Go to the previous page.

### NextPage

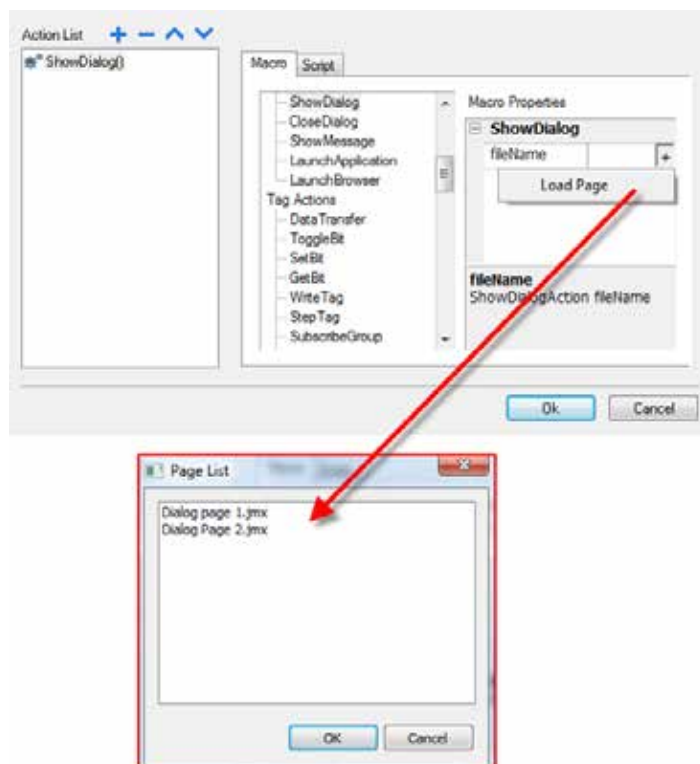
Go to the next page.

### LastVisitedPage

Go to the previously displayed page

### ShowDialog

Opens a dialog page defined in the project.

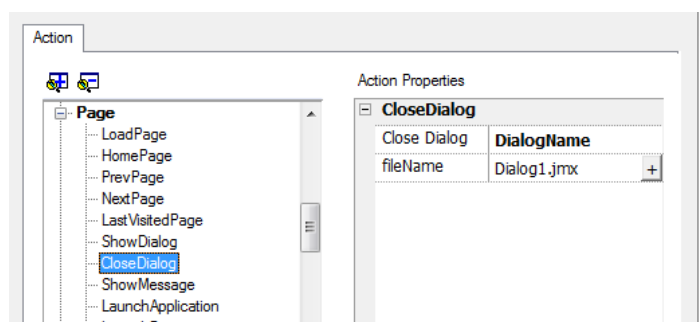


## CloseDialog

Close dialog pages.



Note: This action is applicable only to dialog pages.



## CloseDialog options

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

## JavaScript Interface

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

Where *DialogID*:

<b>All</b>	Closes all open dialogs
<b>Selected</b>	Closes only active dialog
<b>DialogName.jmx</b>	Closes dialog specified as <b>fileName</b> parameter

## Examples

Example	Behavior
<code>project.closeDialog("All");</code>	All open dialogs are closed
<code>project.closeDialog("Selected");</code>	The selected dialog is closed
<code>project.closeDialog("Dialog1.jmx");</code>	All instances of Dialog1 are closed

The function `project.closeDialog()`; without parameter works as `project.closeDialog("Selected");`.

## ShowMessage

Displays a popup message. Enter the text of the message to be displayed.

## LaunchApplication

Launches an external application.

Parameter	Description
<b>App Name</b>	Executable name with extension (for example, "notepad.exe" to run Notepad)
<b>Path</b>	Application path.
<b>Arguments</b>	Application specific arguments (for example, <code>\flash\qthmi\Manual.pdf</code> to open the document "Manual.pdf")
<b>Single Instance</b>	Argument to start the application in a single instance or multiple instances.  When single instance is selected, the system first verifies whether the application is already running; if so, then the application is brought to the foreground, if not, then the application is launched.
<b>FlushRuntimeCache</b>	Flush all runtimes cache to free as more ram as possible before running the application.



Note: Arguments with spaces must be quoted (for example, `"\Storage Card\Manual.pdf"`)

Example:



LaunchApplication	
Application Name	\Windows\cmd.exe
Executable path	
arguments	/c "\Flash\New Folder\test.bat" Par1 Par2
Single Instance	true

## LaunchBrowser

Opens the default web browser. You can define URL address as argument.



Note: Only works on platforms having a native web browser (for example, on Windows CE PRO with Internet Explorer enabled).

## LaunchVNC

Starts VNC server and opens the configuration.



Macro available only for HMI devices based on Windows CE platform.

On HMI devices based on Linux platform the VNC service can be enabled from the "Service" tab of the "Linux Devices" on page 500 BSP v1.0.44 or higher required.

See "Plug-in" on page 77 to include it on Windows CE devices.

## LaunchPDFViewer

Starts PDF Viewer.

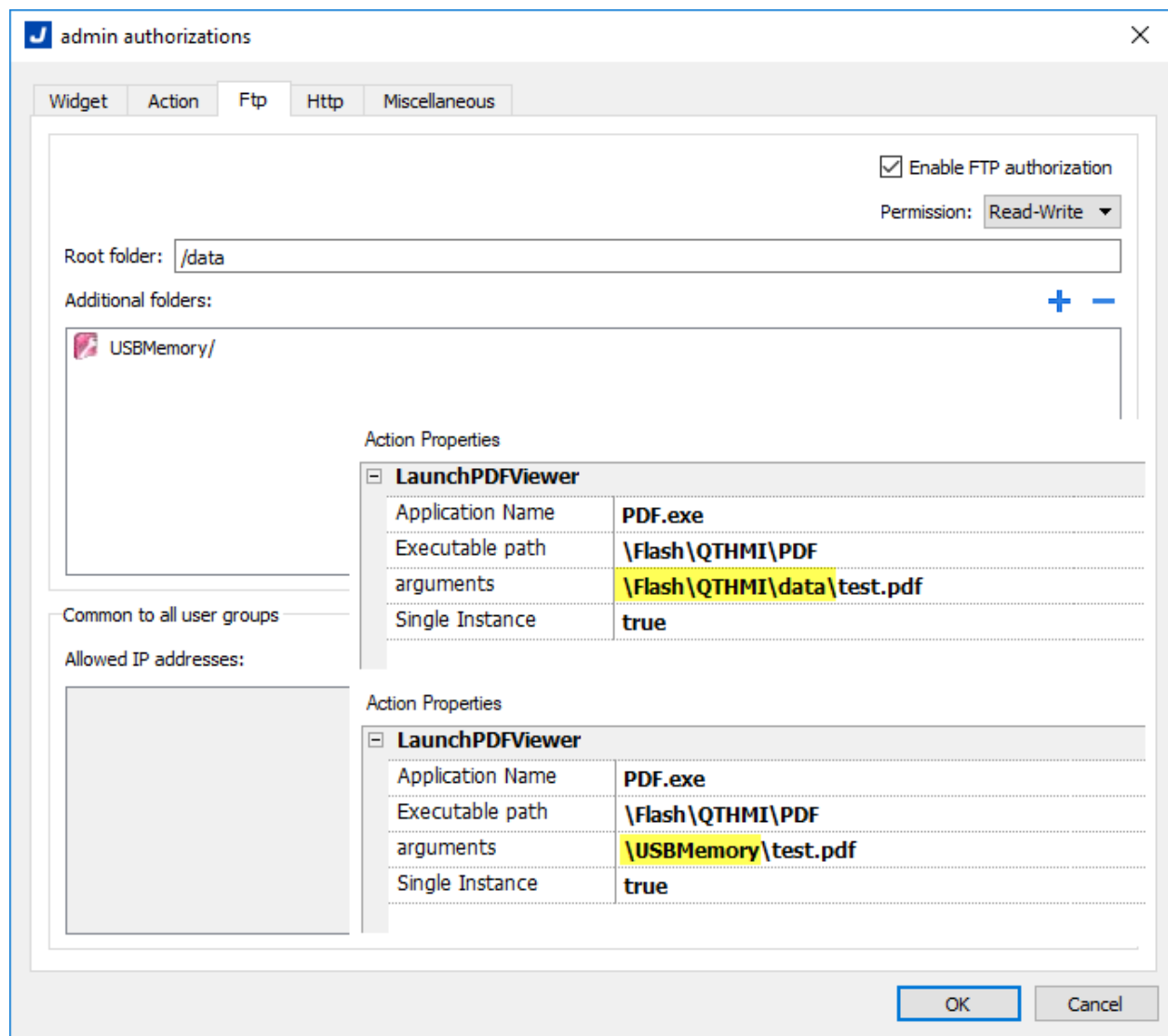


On WCE devices, only works on devices that include PDF Viewer. See "Plug-in" on page 77 to include it on Windows CE devices.

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 567).

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



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**

Widget Action **Ftp** Http Miscellaneous

☒ 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

OK Cancel

**-hide-open-button** (available only on Linux devices)

Using this option, the icon to open a different file will be removed from the PDF toolbar (to restrict navigation to PDF file already opened and passed via command line).

#### Action Properties

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

## LaunchUpdater

Updates project and runtime from an external device.

Use **Path** parameter to specify the folder that will contain the update package file. Leave the path parameter empty if you prefer select the file manually on the HMI device when the macro is invoked.

When the LaunchUpdater macro is executed, the below dialog is showed on HMI device



Note: Not supported in devices based on Win32.

## JavaScript Interface

*project.launchUpdater(strPath)*

## Examples

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

## LaunchHMICloudEnabler

Open the HMI Cloud Enabler.



Macro available only for HMI devices based on Windows CE platform.

On HMI devices based on the Linux platform the Service Cloud can be enabled from the "Service" tab of the "Linux Devices" on page 500.

## 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 108

## LoadProject

Unload current project and load the selected project inside the HMI device.

The project name has to be specified using relative path, as for the below example:

Action Properties

LoadProject	
projectName	workspace/project2/project2.jpr

## LastVisitedProject

Unload current project and return to previous project

## Print actions

Manages print tasks.

### PrintGraphicReport

Prints a graphic report.

Parameter	Description
<b>reportName</b>	Assigns a name to the report
<b>silent</b>	<b>false</b> = allows to set printer properties at run time
<b>fileName</b>	File name (available only for PDF reports) Supported placeholders: <ul style="list-style-type: none"> <li>• %n = Report name</li> <li>• %p = Project name</li> <li>• %y = Year, %M = Month, %d = Day</li> <li>• %h = Hour, %m = Minutes, %s = Seconds.</li> </ul>
<b>folderPath</b>	Folder Path (available only for PDF reports) Note that the pathname of the arguments field uses native OS format (see <a href="#">"HMI devices capabilities" on page 567</a> ). <ul style="list-style-type: none"> <li>• On WinCE devices Path for USB Device is "\\USBMemory"</li> <li>• On Linux devices Path for USB Device is "/mnt/usbmemory" "testFolder" will be inside "/mnt/data/hmi/qthmi/deploy/testFolder"</li> </ul>

### PrintText

Prints a string.

Parameter	Description
<b>text</b>	String to be printed
<b>silent</b>	<b>false</b> = allows to set printer properties at run time

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 567](#))

## PrintBytes

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

Parameter	Description
<b>bytes</b>	Exadecimal string to print
<b>silent</b>	<b>false</b> = allows to set printer properties at run time

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 567](#))

## EmptyPrintQueue

Flushes the current printing queue. If executed while executing a job, the queue is cleared at the end of the job.

## PausePrinting

Puts the current printing queue on hold. If executed while executing a job, the queue is paused at the end of the job.

## ResumePrinting

Restarts a queue previously put on hold.

## AbortPrinting

Stop the execution of the current job and removes it from the queue. If the queue has another job, then, after aborting, the next job starts.

# Recipe actions

Used to program recipe management.

## DownloadRecipe

Copy recipe data from HMI device flash memory to the controller (e.g. PLC, local variable, depending on the protocol).

Parameter	Description
<b>RecipeName</b>	Name of recipe to download
<b>RecipeSet</b>	Number of recipe set to copy. <b>curSet</b> = download currently selected recipe set

## UpLoadRecipe

Saves recipe data from the controller (e.g. PLC, local variable, depending on the protocol) to the device Flash Memory.

Parameter	Description
<b>RecipeName</b>	Name of recipe to upload
<b>RecipeSet</b>	Number of recipe set to copy. <b>curSet</b> = upload currently selected recipe set

## WriteCurrentRecipeSet

Sets the selected recipe as current recipe set.

Parameter	Description
<b>RecipeName</b>	Name of recipe to set as current recipe
<b>RecipeSet</b>	Recipe set to define as current recipe set

## DownLoadCurRecipe

Downloads current set of recipe data to the controller.

No parameter is required.

## UploadCurRecipe

Uploads set of controller data to current recipe set.

No parameter is required



## ResetRecipe

Restores factory settings for recipe data. Original recipe data will overwrite uploaded recipes

Select the recipe that you want to reset to factory data.

## DumpRecipeData

Dumps recipe data to internal or external storage. Data is saved in .csv format.



Parameter	Description
<b>RecipeName</b>	Name of recipe to dump
<b>FilePath</b>	<p>Destination folder</p> <ul style="list-style-type: none"> <li>• Internal = <code>\Flash\QTHMI\workspace\Dump</code></li> <li>• USB drive = <code>\USBMemory</code></li> <li>• SD Card = <code>\Storage Card</code></li> <li>• Public Network = <code>\\&lt;hostname or IP&gt;\sharePath</code></li> <li>• Private Network = <code>\\&lt;username&gt;:&lt;password&gt;@&lt;hostname or IP&gt;\sharePath</code></li> </ul> <p> Note: supported formats for external memory are FAT or FAT32 (NTFS format is not supported).</p> <p> Note: Private networks are supported only from Linux devices with BSP 1.0.25 and above.</p>
<b>FileName</b>	<p>Tag that specifies a filename.</p> <p>The below wildcards are supported</p> <ul style="list-style-type: none"> <li>• %r = Recipe name</li> <li>• %d = Dataset name</li> </ul> <p>Example: %r_%d</p>
<b>DateTimePrefixFileName</b>	<b>true</b> = the dumped file will have date and time as prefix to its name (for example D2012_01_01_T10_10_recipe1.csv)
<b>TimeSpec</b>	<p>Time format:</p> <ul style="list-style-type: none"> <li>• <b>Local</b> = the time values exported are the time of the HMI device.</li> <li>• <b>Global</b> = the time values exported are in UTC format.</li> </ul>

## RestoreRecipeData

Restores previously saved recipe data.

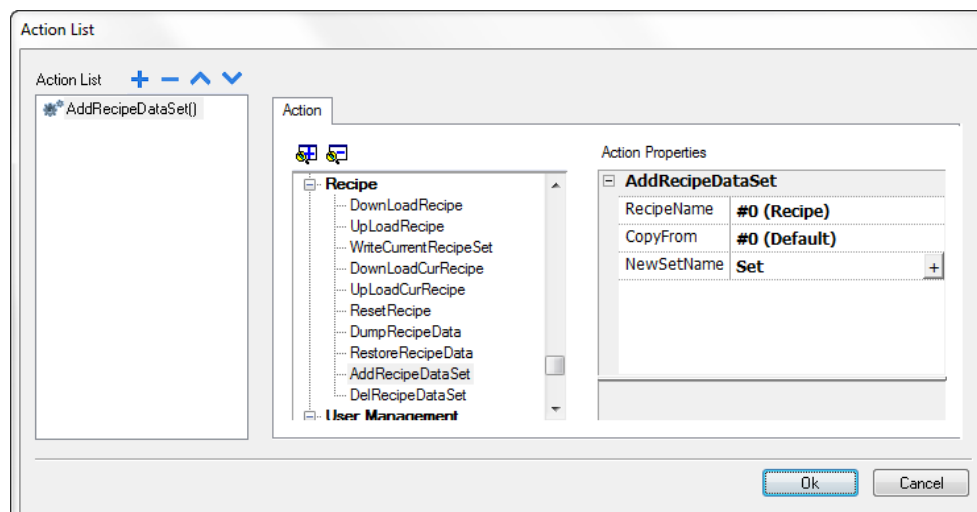
Parameter	Description
<b>RecipeName</b>	<p>Recipes to restore:</p> <ul style="list-style-type: none"> <li>• AllRecipes Data of all recipes will be replaced with the data read from the external file</li> <li>• CurrentRecipe Only the data of the current selected recipe will be replaced with the data read from the external file</li> </ul>
<b>RecipeDataSet</b>	<p>Available only when RecipeName=CurrentRecipe.</p> <p>Select the data sets to restore:</p>



Parameter	Description
	<ul style="list-style-type: none"> <li>AllRecipeDataSet All data set will restored</li> <li>curSet Only the data set of the current selected data set will restore</li> </ul>
<b>Restore Type</b>	<p>Available only when RecipeDataSet=AllRecipeDataSet.</p> <p>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</p> <ul style="list-style-type: none"> <li>Replace All data sets that are inside the device are removed and replaced with the data sets from the csv file</li> <li>Match Replace only the data set inside the device that have the same data set id</li> <li>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)</li> </ul>
<b>FilePath</b>	<p>Source folder</p> <ul style="list-style-type: none"> <li>Internal = <i>\Flash\QTHMI\workspace\Dump</i></li> <li>USB drive = <i>\USBMemory</i></li> <li>SD Card = <i>\Storage Card</i></li> <li>Public Network = <i>\\&lt;hostname or IP&gt;\sharePath</i></li> <li>Private Network = <i>\\&lt;username&gt;:&lt;password&gt;@&lt;hostname or IP&gt;\sharePath</i></li> </ul> <p> Note: supported formats for external memory are FAT or FAT32 (NTFS format is not supported).</p> <p> Note: Private networks are supported only from Linux devices with BSP 1.0.25 and above.</p>
<b>FileName</b>	Attached tag from which read the file name at run time.
<b>BrowseForFile</b>	<p><b>true</b> = shows the Open dialog to browse the file to read.</p> <p><b>false</b> = no dialog is shown,</p>

## AddRecipeDataSet

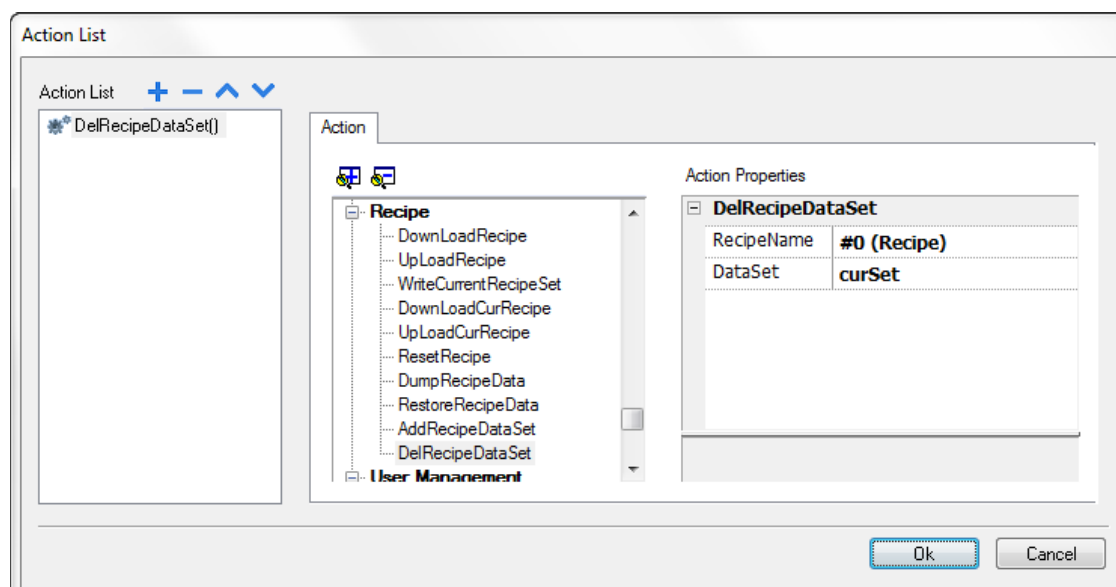
Adds a new dataset to the selected recipe. The new dataset is appended at the end of the already defined datasets.



Parameter	Description
<b>RecipeName</b>	Recipe where the dataset is added.
<b>CopyFrom</b>	Dataset from where parameters values are copied from to initialize the new dataset
<b>NewSetName</b>	Name of new dataset. Here you can use a tag reference.

## DelRecipeDataSet

Deletes a dataset from the selected recipe. Deleting a dataset will rearrange the position number of the datasets that follow.



Parameter	Description
<b>RecipeName</b>	Recipe where the dataset is to be deleted.
<b>DataSet</b>	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 HMI Client to access remote files via FTP.



**Important: Enable FTP support and give all necessary user rights to the folders used to transfer files.**

### UploadToHMI

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

Parameter	Description
<b>Destination</b>	Destination path on HMI device for file upload
<b>Filter</b>	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
<b>Source</b>	Source path on the HMI device for file download
<b>Filter</b>	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
<b>dirPath</b>	Source path on the HMI device for file download/upload
<b>strFilter</b>	File extensions of the files to be displayed separated by commas (for example, *.txt)

Return values:

<b>True</b>	Transfer successful
<b>False</b>	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
<b>TrendName</b>	Name of historical trend to store
<b>FolderPath</b>	<p>Destination folder:</p> <ul style="list-style-type: none"> <li>Internal = <code>\Flash\QTHMI\workspace\Dump</code></li> <li>USB drive = <code>\USBMemory</code></li> <li>SD Card = <code>\Storage Card</code></li> <li>Public Network = <code>\\&lt;hostname or IP&gt;\sharePath</code></li> <li>Private Network = <code>\\&lt;username&gt;:&lt;password&gt;@&lt;hostname or IP&gt;\sharePath</code></li> </ul> <p> Note: supported formats for external memory are FAT or FAT32 (NTFS format is not supported).</p> <p> Note: Private networks are supported only from Linux devices with BSP 1.0.25 and above.</p>
<b>FileFormat</b>	<p><b>Binary</b> = the buffer is dumped in binary format (a .dat file and .inf file). Both these files are then required to convert data in .csv format by an external utility.</p> <p><b>Compatibility CSV</b> = the buffer is dumped to the specified location as a .csv file format compatible with versions 1.xx</p> <p><b>Compact CSV</b> = the buffer is dumped to the specified location as a .csv file using a newer format</p> <p>See <a href="#">"Exporting trend buffer data" on page 254</a></p>
<b>DateTimePrefix</b>	<b>true</b> = the dumped file will have date and time as prefix to its name (for example D2012_01_01_T10_10_Trend1.csv)

Parameter	Description
<b>TimeSpec</b>	Time format: <ul style="list-style-type: none"> <li>• <b>Local</b> = the time values exported are the time of the HMI device.</li> <li>• <b>Global</b> = the time values exported are in UTC format.</li> </ul>
<b>FileName</b>	Enabled when the DateTimePrefixFileName=true The below wildcards are supported <ul style="list-style-type: none"> <li>• %n = Trend name</li> <li>• %y = Year</li> <li>• %M = Month</li> <li>• %d = Day</li> <li>• %h = Hour</li> <li>• %m = Minutes</li> <li>• %s = Seconds</li> </ul> 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 254](#)

Parameter	Description
<b>Select Fields</b>	Select the columns to export inside the dumped file. Available columns are: <ul style="list-style-type: none"> <li>• Date and Time</li> <li>• Date</li> <li>• Time</li> <li>• Value</li> <li>• Quality</li> </ul> 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: <ul style="list-style-type: none"> <li>• "" (<i>Empty string = all available fields</i>)</li> <li>• "DateTime,Value,Quality"</li> <li>• "Date,Time,Value"</li> </ul>
<b>Select Curves</b>	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.

Parameter	Description																																						
	<p>Example:</p> <ul style="list-style-type: none"> <li>• Empty string or "All curves" will export all datasets</li> <li>• "Name1,Name2,Name3"</li> <li>• "Name1,Name3"</li> </ul>																																						
<b>Date Format</b>	<p>Select the Date and Time format</p> <p>Using "Attach to tag" is possible define the date format at runtime through a string</p> <table> <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>	Date Placeholder		<b>d</b>	The day as number without a leading zero (1 to 31)	<b>dd</b>	The day as number with a leading zero (01 to 31)	<b>ddd</b>	The abbreviated localized day name (e.g. 'Mon' to 'Sun')	<b>dddd</b>	The long localized day name (e.g. 'Monday' to 'Sunday').	<b>M</b>	The month as number without a leading zero (1-12)	<b>MM</b>	The month as number with a leading zero (01-12)	<b>MMM</b>	The abbreviated localized month name (e.g. 'Jan' to 'Dec').	<b>MMMM</b>	The long localized month name (e.g. 'January' to 'December').	<b>yy</b>	The year as two digit number (00-99)	<b>yyyy</b>	The year as four digit number	Time Placeholder		<b>h</b>	The hour without a leading zero (0 to 23 or 1 to 12 if AM/PM display)	<b>hh</b>	The hour with a leading zero (00 to 23 or 01 to 12 if AM/PM display)	<b>m</b>	The minute without a leading zero (0 to 59)	<b>mm</b>	The minute with a leading zero (00 to 59)	<b>s</b>	The second without a leading zero (0 to 59)	<b>ss</b>	The second with a leading zero (00 to 59)	<b>zzz</b>	The millisecond with leading zero
Date Placeholder																																							
<b>d</b>	The day as number without a leading zero (1 to 31)																																						
<b>dd</b>	The day as number with a leading zero (01 to 31)																																						
<b>ddd</b>	The abbreviated localized day name (e.g. 'Mon' to 'Sun')																																						
<b>dddd</b>	The long localized day name (e.g. 'Monday' to 'Sunday').																																						
<b>M</b>	The month as number without a leading zero (1-12)																																						
<b>MM</b>	The month as number with a leading zero (01-12)																																						
<b>MMM</b>	The abbreviated localized month name (e.g. 'Jan' to 'Dec').																																						
<b>MMMM</b>	The long localized month name (e.g. 'January' to 'December').																																						
<b>yy</b>	The year as two digit number (00-99)																																						
<b>yyyy</b>	The year as four digit number																																						
Time Placeholder																																							
<b>h</b>	The hour without a leading zero (0 to 23 or 1 to 12 if AM/PM display)																																						
<b>hh</b>	The hour with a leading zero (00 to 23 or 01 to 12 if AM/PM display)																																						
<b>m</b>	The minute without a leading zero (0 to 59)																																						
<b>mm</b>	The minute with a leading zero (00 to 59)																																						
<b>s</b>	The second without a leading zero (0 to 59)																																						
<b>ss</b>	The second with a leading zero (00 to 59)																																						
<b>zzz</b>	The millisecond with leading zero																																						

Parameter	Description
	<b>Time Placeholder</b>
	<b>z</b> The millisecond
	<b>AP</b> Use AM/PM display. AP will be replaced by either "AM" or "PM".
	<b>ap</b> Use am/pm display. ap will be replaced by either "am" or "pm".
<b>Language</b>	Select the language to use.



Note: execution of the DumpTrend action will automatically force a flush to disk of the data temporarily maintained in the RAM memory. See ["History trends" on page 257](#) 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 JMobile Studio 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 maintained because now there is the possibility to dump trend buffer directly in .csv format.**

### .csv file structure

The resulting .csv file has five columns

Column	Description
<b>Data Type</b>	Data type of sampled tag: 0 = empty 1 = boolean 2 = byte 3 = short 4 = int 5 = unsignedByte 6 = unsignedShort 7 = unsignedInt 8 = float 9 = double
<b>Value</b>	Value of the sample
<b>Timestamp (UTC)</b>	Timestamp in UTC format
<b>Sampling Time(ms)</b>	Sampling interval time in milliseconds
<b>Quality</b>	Tag value quality. Information coded according the OPC DA standard and stored in a byte data (8 bits) defined in the form of three bit fields; Quality, Sub status and Limit status.  The eight quality bits are arranged as follows: QQSSSSL. For a complete and detailed description of all the single fields, please refer to the OPC DA official documentation.

### Commonly quality values

The most commonly used quality values returned by the HMI acquisition engine are:

Quality Code	Quality	Description
0	BAD	The value is bad but no specific reason is given
4	BAD	Specific server problem with the configuration. For example, the tag has been deleted from the configuration file (tags.xml).
8	BAD	No value may be available at this time, for example the value has not been provided by the data source.
12	BAD	Device failure detected
16	BAD	Timeout before device response.
24	BAD	Communication failure



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 <b>Limits</b> field indicates which limit has been exceeded but the value can move farther out of this range.
85	UNCERTAIN	Returned value outside its defined limits defined. In this case the <b>Limits</b> field indicates which limit has been exceeded but the value can move farther out of this range. The value has 'pegged' at some lower limit.
86	UNCERTAIN	Returned value outside its defined limits defined. In this case the <b>Limits</b> field indicates which limit has been exceeded but the value can move farther out of this range. The value has 'pegged' at some higher limit
87	UNCERTAIN	Returned value outside its defined limits defined. In this case the <b>Limits</b> field indicates which limit has been exceeded but the value can move farther out of this range. The value is a constant and cannot move.
192	GOOD	-




## DeleteTrend



Deletes saved trend data.

Define the name of the trend from which you want to delete logs.

## DumpEventArchive

Stores historical alarm log and audit trail data to external drives, such as USB memory or SD card.

Parameter	Description
<b>EventArchive</b>	Name of buffer to dump data
<b>FolderPath</b>	<p>Destination folder</p> <ul style="list-style-type: none"> <li>Internal = <code>\Flash\QTHMI\workspace\Dump</code></li> <li>USB drive = <code>\USBMemory</code></li> <li>SD Card = <code>\Storage Card</code></li> <li>Public Network = <code>\\&lt;hostname or IP&gt;\sharePath</code></li> <li>Private Network = <code>\\&lt;username&gt;:&lt;password&gt;@&lt;hostname or IP&gt;\sharePath</code></li> </ul> <p> Note: supported formats for external memory are FAT or FAT32 (NTFS format is not supported).</p> <p> Note: Private networks are supported only from Linux devices with BSP 1.0.25 and above.</p>
<b>DumpConfigFile</b>	Dump the description files of the archives
<b>DumpAsCSV</b>	<p><b>true</b> = the buffer is dumped to the specified location as a .csv file</p> <p><b>false</b> = the buffer is dumped in binary format (a .dat file and .inf file). Both these files are then required to convert data in .csv format by an external utility.</p>
<b>DateTimePrefix</b>	<b>true</b> = the dumped file will have date and time as prefix to its name (for example D2012_01_01_T10_10_alarmBuffer1.csv)
<b>timeSpec</b>	<p>Time format:</p> <ul style="list-style-type: none"> <li><b>Local</b> = the time values exported are the time of the HMI device.</li> <li><b>Global</b> = the time values exported are in UTC format.</li> </ul>
<b>csv Cols</b>	<p>Select the columns to dump into the .csv file.</p> <p> Available only when the EventArchive is an alarms buffer</p>
<b>FileName</b>	<p>The below wildcards are supported</p> <ul style="list-style-type: none"> <li>%n = Event archive name</li> <li>%y = Year</li> <li>%M = Month</li> <li>%d = Day</li> <li>%h = Hour</li> <li>%m = Minutes</li> <li>%s = Seconds</li> </ul>

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

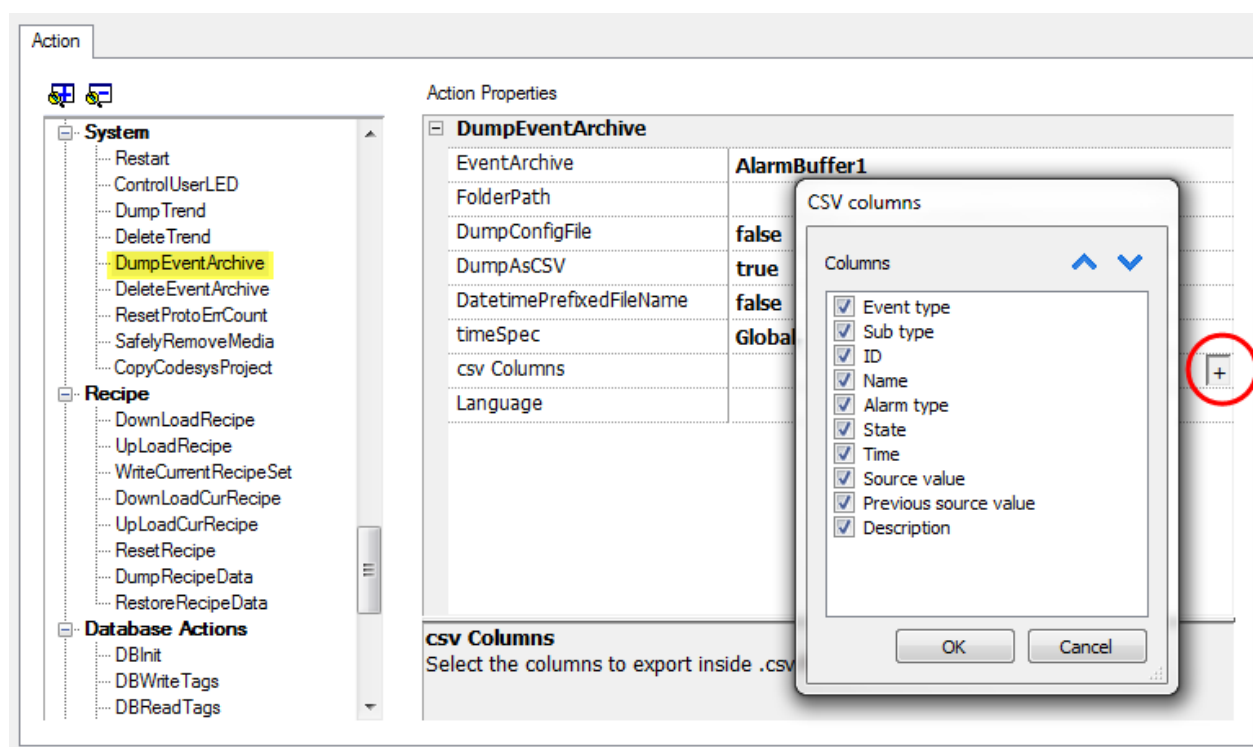
## Dumping in CSV Format

DumpAsCSV = true

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



Note: available only for Alarms buffers.



## Dumping in BINARY Format

DumpAsCSV = false

When exporting Event buffers in binary format and **DumpConfigFile** is set to true (recommended settings), there are two folders:

- **data**, containing data files,
- **config**, containing configuration files for .csv conversion.

Once the two folders are copied from the USB drive to the computer disk, the folder structure will be:

```
\config\
    alarms.xml
    eventconfig.xml
\data\
    AlarmBuffer1.dat
    AlarmBuffer1.inf
\
AlarmBufferReader.exe
```

### To convert dump files to .csv

The AlarmBufferReader.exe tool is stored in the *Utils* folder of the JMobile Studio 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.

## 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 97 for details.

## SafelyRemoveMedia

Provides for safe removal of SD card or USB drive from HMI.

## CopyCodesysProject

Copies the CODESYS 2.3 project files (.prg, .chk and .sdb) from the source path to the device CODESYS folder.

Files are automatically renamed to DEFAULT.CHK, DEFAULT.PRG, BOOT.SDB if needed.

After copy the CODESYS module is stopped, reloaded and started again.

Parameters	Description
<b>Source Path</b>	Project path into external storage (for example, \USBMemory\Codesys)
<b>Copy Symbols</b>	<b>true</b> = copies .sdb symbol file required by the CODESYS 2 ETH protocol

### To generate CODESYS project files

1. Run **Project > Rebuild All**: an updated .sdb symbol file is generated.
2. Run **Online > Create boot project**: the .chk and .prg file are generated.

This action can be used to transfer a CODESYS project from one HMI device to another. In this case copy these files from the HMI running CODESYS project:

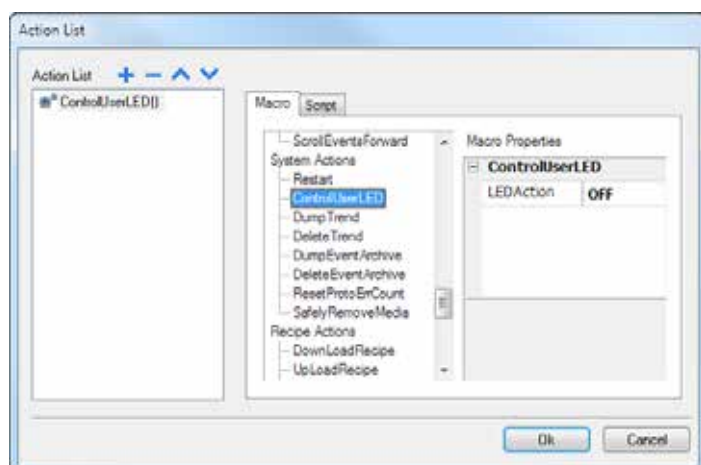
- default.chk
- default.prg
- boot.sdb

Verify **PLC** and **CopyCodesysProject** action status using the following PLC system variables:

- **PLC Status**
- **Get CopyCodesysProject Action**

## ControlUserLED




Sets the user LED behavior.



Not available on Linux platforms (find the platform of your device at ["HMI devices capabilities" on page 567](#))

## 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
<b>EventArchive</b>	Name of buffer to dump data
<b>FolderPath</b>	<p>Destination folder</p> <ul style="list-style-type: none"> <li>• Internal = <code>\Flash\QTHMI\workspace\Dump</code></li> <li>• USB drive = <code>\USBMemory</code></li> <li>• SD Card = <code>\Storage Card</code></li> <li>• Public Network = <code>\\&lt;hostname or IP&gt;\sharePath</code></li> <li>• Private Network = <code>\\&lt;username&gt;:&lt;password&gt;@&lt;hostname or IP&gt;\sharePath</code></li> </ul> <p> Note: supported formats for external memory are FAT or FAT32 (NTFS format is not supported).</p> <p> Note: Private networks are supported only from Linux devices with BSP 1.0.25 and above.</p>
<b>FileName</b>	<p>The below wildcards are supported</p> <ul style="list-style-type: none"> <li>• %n = Event archive name</li> <li>• %y = Year</li> <li>• %M = Month</li> <li>• %d = Day</li> <li>• %h = Hour</li> <li>• %m = Minutes</li> <li>• %s = Seconds</li> </ul> <p>Example: <code>%n\%y%M%d\%h%m%s</code></p>
<b>Format</b>	<p>Format of the output file</p> <ul style="list-style-type: none"> <li>• CSV</li> </ul>
<b>Signed</b>	<p>Generate the file signature.</p> <p> The algorithm to use to signing is defined inside the project properties parameters See "<a href="#">Project</a>" on page 78 for the available algorithms</p>
<b>TimeSpec</b>	<p>Time format:</p> <ul style="list-style-type: none"> <li>• <b>Local</b> = the time values exported are the time of the HMI device.</li> <li>• <b>Global</b> = the time values exported are in UTC format.</li> </ul>
<b>PeriodMode</b>	Defines the time window to export

Parameter	Description
	<ul style="list-style-type: none"> <li>• All events</li> <li>• Today</li> <li>• Yesterday</li> <li>• Last week</li> <li>• Last month</li> <li>• Current week</li> <li>• Current month</li> <li>• Custom</li> </ul> <p>The additional parameters "periodFrom" and "periodTo" will be shown</p>
<b>Separate Date and Time</b>	Uses two separate columns for Date and Time
<b>Date Format</b>	Select the Date and Time format

### Signed file

When the "Signed file" parameter is true, two files will be added in addition to filename.csv:

- filename.csv.sign  
The file signature will ensure that the records within the file filename.csv file have not been altered
- ssl-HMI.crt  
A copy of the certificate of the HMI device required to verify the authenticity of the report.

Name	Date modified	Type	Size
 AuditTrail-1413.csv	28/03/2018 16:13	Microsoft Excel Comma Separated Values File	1 KB
 AuditTrail-1413.csv.sign	28/03/2018 16:13	SIGN File	1 KB
 ssl-HMI.crt	28/03/2018 16:16	Security Certificate	2 KB

For more information about the certificate and how to verify signed files, see ["x.509 Certificate"](#) on page 302.

For more information about the exported information see ["Exporting audit trail as .csv files"](#) on page 327.

## LogMessage

Add a message into the audit trail buffer.

This macro give the possibility to developer to decide to keep track of some events (e.g. when a button is pressed, when a page is activate, etc.) into the audit trail. The attach to tag to have the possibility to define the message to log at runtime is supported.

Parameter	Description
<b>EventArchive</b>	Name of the audit buffer where add the message
<b>Message</b>	Message to add inside the audit buffer

## DeleteOldFiles

This macros delete files older that a give number of days.

In PC there is no restriction in using path. In panels it is allowed in dynamic media and data partition (/Flash in WinCE and /mnt/data in Linux )



**It will be developer responsibility to configure the application to avoid the possibility to delete system files.**

Parameter	Description
<b>FolderPath</b>	Folder where search the files to delete
<b>FileTypes</b>	List of files to delete separate by comma. Wildcard are supported Example: *.png,*.jpg
<b>OlderDays</b>	Minimum number of days without changes

## Tag actions

Interacts with tags.

### DataTransfer

Exchanges data between:

- two controllers,
- registers within a controller,
- from system variables to controllers,
- from controllers to system variables

The various tag types include a controller tag, a system variable, a recipe tag and widget property.

### ToggleBit

Toggles a bit value of a tag.

**BitIndex** allows you to select the bit to be toggled: toggling requires a read-modify-write operation; the read value is inverted and then written back to the tag.

### SetBit

Sets the selected bit to "1".

**BitIndex** allows you to select the bit position inside the tag.

### ResetBit

Resets the selected bit to "0"

**BitIndex** allows you to select the bit position inside the tag.

### WriteTag

Writes constant values to the controller memory. Specify tag name and value.



## StepTag

Increments or decrements tag value.

Parameter	Description
TagName	Name of tag to increase/decrease
Step	Step value
Do not step over limit	Enables step limit
Step Limit	Value of step limit, if enabled.

## BiStep

This action is similar to the StepTag action but the direction Increment/Decrement is automatically chosen by the rotation of the Wheel. Tag value will be increased when the Wheel is rotated clockwise. Tag value will be decreased in when the Wheel is rotated counterclockwise.

Property	Description
TagName	Name of Tag on which execute BiStep Tag action
Step	Value to be added/subtracted to Tag at every wheel rotation (depends on Event step property)
Event step	<p>This property allows to chose if adding/subtracting step values at every single wheel step, or at every rotation event.</p> <p><b>false</b> = The step value is added/subtracted to the Tag at every rotation event.  <i>Example: rotate the wheel performing 5 wheel steps in a single event, Tag will be increased/decreased by 1.</i></p> <p><b>true</b> = The step value is added/subtracted to the Tag at every single wheel step.  <i>Example: rotate the wheel performing 5 wheel steps in a single rotation, Tag will be increased/decreased by 5.</i></p>
Do not step over limit	If true, enables lower and upper limits, which represents the lower and the higher value that the Tag can assume due a BiStep Tag action
LowerLimit	If "Do not step over limit" is true, this property represents the lower value that the Tag can assume due a BiStep Tag action
UpperLimit	If "Do not step over limit" is true, this property represents the higher value that the Tag can assume due a BiStep Tag action



Available only inside 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
<b>Protocol ID</b>	Unique identifier of selected protocol
<b>NodeID</b>	Node identifier in selected protocol. Can be attached to a tag.
<b>Enable</b>	Node communication status:  <b>False</b> = disabled  <b>True</b> = enabled  When attached to a tag, tag = 0 means <b>False</b>

## BACnetClearPriority

Refer to the BACnet manual inside the “Communication Drivers” folder for a detailed description of BACnet actions.

## BACnetClearAllPriorities

Refer to the BACnet manual inside the “Communication Drivers” folder for a detailed description of BACnet actions.

## BACnetSetPriority

Refer to the BACnet manual inside the “Communication Drivers” folder for a detailed description of BACnet actions.

## ClearRetentiveMemory

When set to 0, clears the content of the Retentive Memory.

## ForceReadTag

Force a refresh of the specified tag from the remote controller.

# Trend actions

Used for Live Data Trends and Historical Trends Widget.

## RefreshTrend

Refreshes the **Trend** window.

It can be used in any Trends/Graphs widgets. Specify the widget as a parameter for the action.

## ScrollLeftTrend

Scrolls the **Trend** window to the left side, by one-tenth (1/10) of the page duration.



Note: with the real-time trends pause the trend using the **PauseTrend** action, or the window will be continuously shifted to the current value.

## ScrollRightTrend

Scrolls the **Trend** window to the right side, by one-tenth (1/10) of the page duration.



Note: with the real-time trends pause the trend using the **PauseTrend** action, or the window will be continuously shifted to the current value.

## PageLeftTrend

Scrolls the **Trend** window by one-page. For example, if the page size is 10 minutes, then use the **PageLeftTrend** action to scroll the trend left for 10 minutes.

## PageRightTrend

Scrolls the **Trend** window by one-page. For example, if the page size is 10 minutes, then use the **PageRightTrend** action to scroll the trend right for 10 minutes.

## PageDurationTrend

Sets the page duration of the **Trend** window.

Define trend name and page duration.



Note: you can set page duration at run time 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.

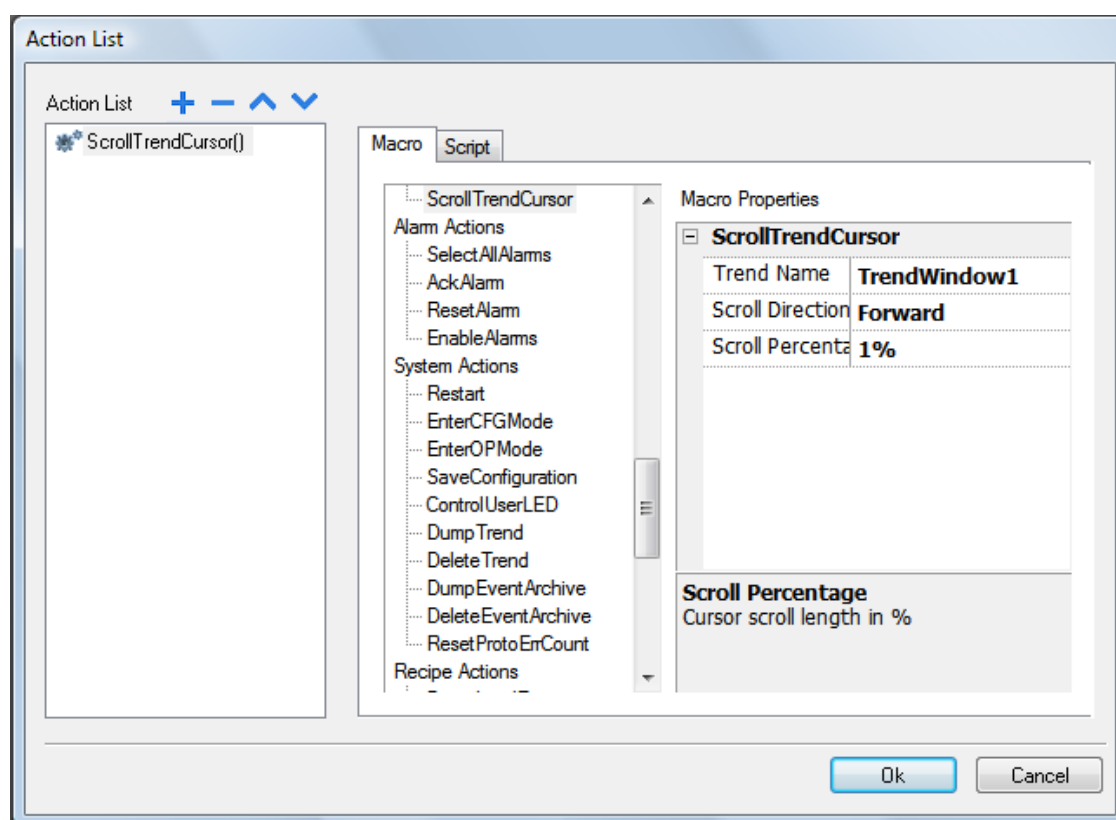
When the graphic cursor is enabled, the scrolling of the trend is stopped.

The **ScrollCursor** action moves the graphic cursor over the curves, or over the entire **Trend** window.

## ScrollTrendCursor

Scrolls the trend cursor backward or forward.

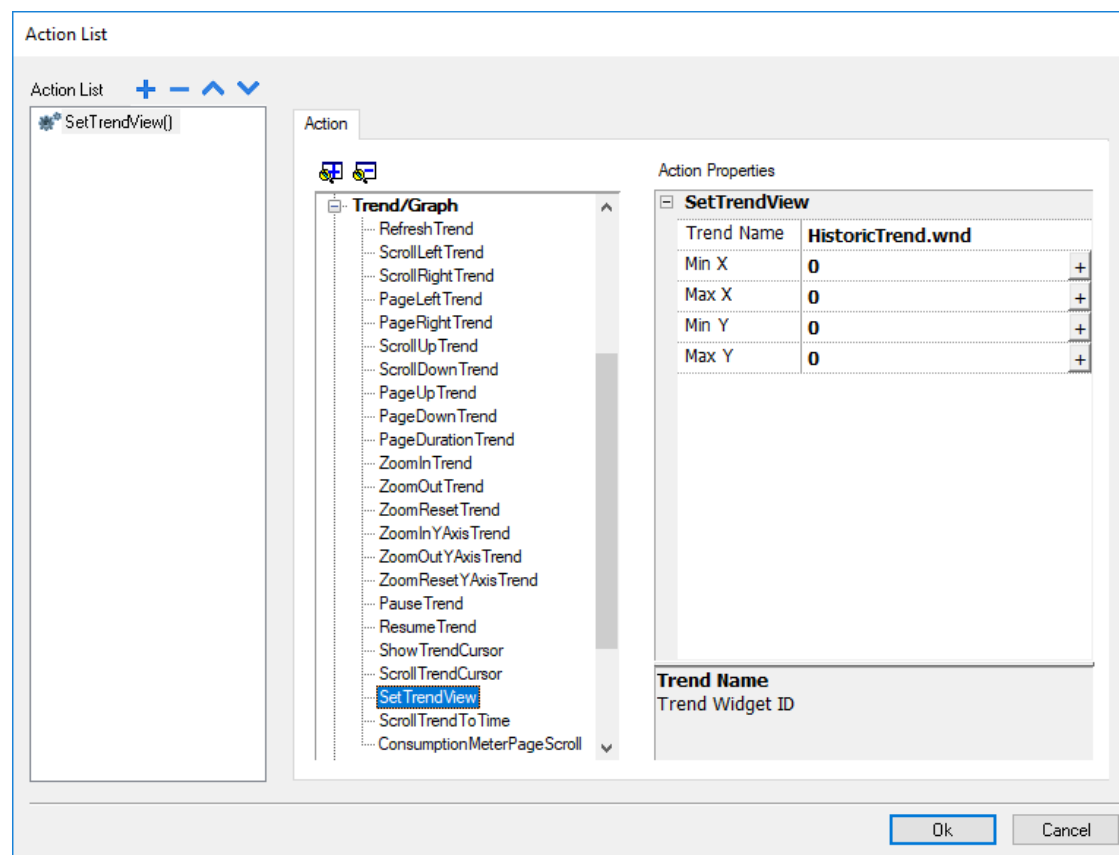
The Y cursor value will display the trend value at the point of the cursor. Scrolling percentage can be set at 1% or 10%. The percentage is calculated on the trend window duration.



## SetTrendView

Use this macro to change the axis ranges of the trend view.

When both Min X=0 and Max X=0, the static values defined inside the properties of widget are used. The same for the Y axe.



## ScrollTrendToTime

Scrolls the **Trend** window to a specified point in time.

Use this action when you need to scroll to a specific position in a trend window when a specific event occurred.

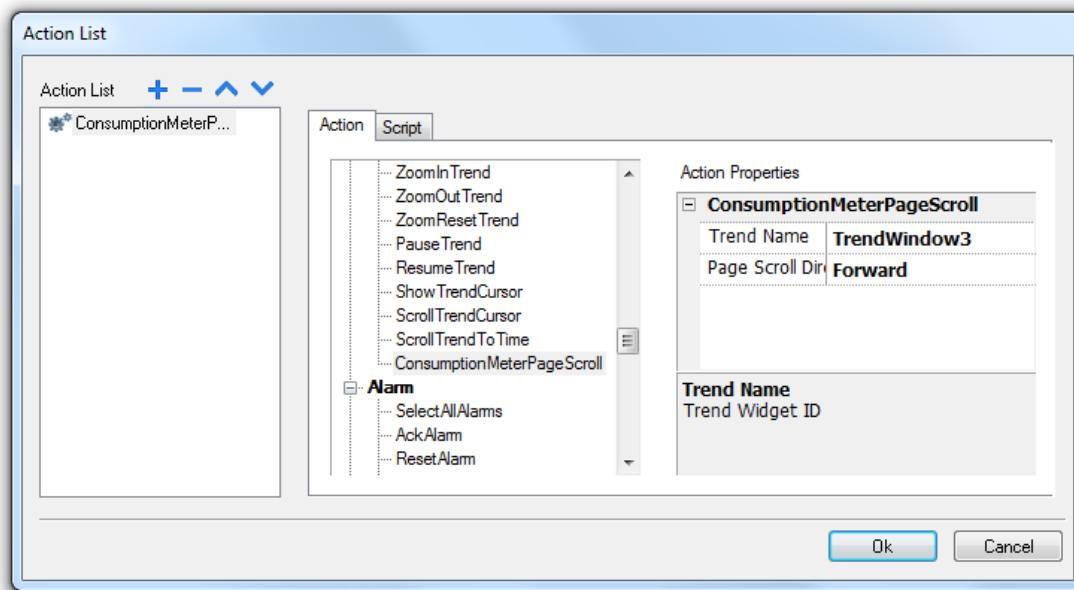
### Example

1. Configure an action for an event (for example, an alarm) that executes a data transfer of the system time into a tag.
2. Select that tag as **ScrollTrendToTime** parameter: the trend windows will be centered at the time when the event was triggered.

## ConsumptionMeterPageScroll

Scrolls the page backward or forward in a Consumption Meter widget.

Parameter	Description
<b>Trend Name</b>	Trend widget ID (for example, TrendWindow3)
<b>Page Scroll Direction</b>	Direction of page scrolling (Forward/backward)



## Text Editor actions

Macros used to interact with the TextEditor widget.

Reference to ["TextEditor widget" on page 430](#) for details

## User management actions

User management and security settings.

### LogOut

Logs off the current user. The default user is then automatically logged in. If no default user has been configured, the login window is displayed.

### SwitchUser

Switches between two users without logging off the logged user: the user login dialog appears. User can click **Back** to go back to the previously logged user.

User name:

Password:

☐ Show password

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 run time: a dialog appears.

---

User name:	<input type="text" value="user3"/>
Password:	<input type="password" value="*****"/> <input type="checkbox"/> Show password
Group:	<input type="text" value="admin"/>
Comments:	<input type="text"/>

---

Password must contain number:	<input type="checkbox"/>
Password must contain special character:	<input type="checkbox"/>
User must change his initial password:	<input type="checkbox"/>
Enable logoff time:	<input type="checkbox"/>
Inactivity logoff time:	<input type="text" value="0"/> min

---

<input type="button" value="Add"/>	<input type="button" value="Cancel"/>
------------------------------------	---------------------------------------

---

## DeleteUser

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

Deletes a user at run time: a dialog appears.

No parameter is required.

User name:

Group:

## EditUsers

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

Edits user settings.

User name:

Password:  ☐ Show password

Group:

Comments:

---

Password must contain number: ☐

Password must contain special character: ☐

User must change his initial password: ☐

Enable logoff time: ☒

Inactivity logoff time:  min

## DeleteUMDynamicFile

Deletes the dynamic user management file. Changes made to users settings at run time 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
<b>Widget</b>	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
<b>Widget</b>	Widget to slide
<b>Direction</b>	Sliding direction
<b>Speed</b>	Transition speed of sliding widget
<b>X Distance</b>	Travel distance of X coordinate in pixels
<b>Y Distance</b>	Travel distance of Y coordinate in pixels
<b>Slide Limit</b>	Enable/Disable movement limits of the widget with respect to the x, y coordinates
<b>X Limit</b>	Limit position of slide action for x coordinate
<b>Y Limit</b>	Limit position of slide action for y coordinate
<b>Toggle Visibility</b>	Show/hide widget at the end of each slide action
<b>Image Widget</b>	Image displayed during slide action

### BeginDataEntry

Displays a keypad and starts data entry on a data field without touching the widget itself. This action can be used to activate data entry using a barcode scanner.

## Java Script Interface

```
project.beginDataEntry(wgtName [, pageName])
```

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

## TriggerIPCamera

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 405](#) 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 237](#) 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 71](#)

## ReplaceMedia

Replaces existing media files with new files from USB/SD card. Can be used to replace video files of MediaPlayer widgets, or images of project.



Note: New media files must have same name and format of the files to be replaced.

Parameter	Description
<b>Media Type</b>	Type of file to update
<b>Device</b>	Device where new media files are supplied
<b>sourcePath</b>	Folder where new media files are stored (for example, "\USBMemory")
<b>Image Resize</b>	Resizes new images to the size of images to be replaced. Not applicable to video files.
<b>Silent</b>	Replaces media automatically. As default a dialog is displayed for the user to specify file location.

### Java Script Interface

```
void replaceMedia(var sourcePath, var bSilent, var Device, var nMediaType, var  
bResize)
```

```
project.replaceMedia("Images", true, "\\USBMemory", 1, true);
```



# 12 The HMI Client

---

HMI Client is a standalone application which provides remote access to the JMobile HMI Runtime, and is included in the JMobile Suite. The HMI Client uses the same graphic rendering system as the runtime in the HMI devices, it relies on a specified JMobile HMI Runtime as server for live data.

HMI Client acts as a remote client and communicates to the server, sharing the local visualization with the tag values that are maintained or updated by the communication protocol.



HMI projects contain properties indicating which page is currently displayed on the HMI and can force the HMI to switch to a specific page. You can use these properties to synchronize pages showed on the HMI device and HMI Client or to control an HMI device with a PLC. See ["Project" on page 78](#) for details.



**To avoid unexpected behavior:**

- be sure to use the same version of the JMobile 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 72](#))

---

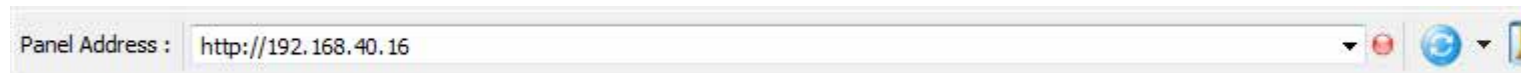
Client application on PC .....	210
Client application on HMI .....	211
Settings and time zone options .....	212

# Client application on PC

To run the HMI Client application on PC:

1. From the **Start** menu > **JMobile Suite** > **HMI Client**: the client opens in a browser-like style window.
2. Type the server/device IP address in the address bar (for example: `http://192.168.1.12`): HMI Client will connect to the server and the same graphical application running on the device will be loaded in the client window.

## The Client application toolbar



Element	Description
HMI server address	IP address of the remote HMI device (e.g. 192.168.0.1:80)
Connection status	Network request status. Red during data exchange.
Reload from cache	Reloads project
BookMark	Bookmarks preferred pages and reload them.
Settings	Opens <b>Settings</b> dialog

## Reload options

Option	Description
F5	Reloads project from cache
Shift + F5	Downloads project to client

## Transferring files to a remote HMI device

You can upload and download files to and from a remote HMI device using two dedicated actions. These actions can only be used from a remote HMI Client and access remote files via FTP.

See ["Remote Client actions" on page 183](#) and ["Remote Client variables" on page 107](#).



**Important: Enable FTP support and give all necessary user rights to the folders used to transfer files.**

## Workspace

Project files are uploaded from the device and stored in HMI Client into the following cache folder.

`%appdata%\Exor\[build number]\client\cache`

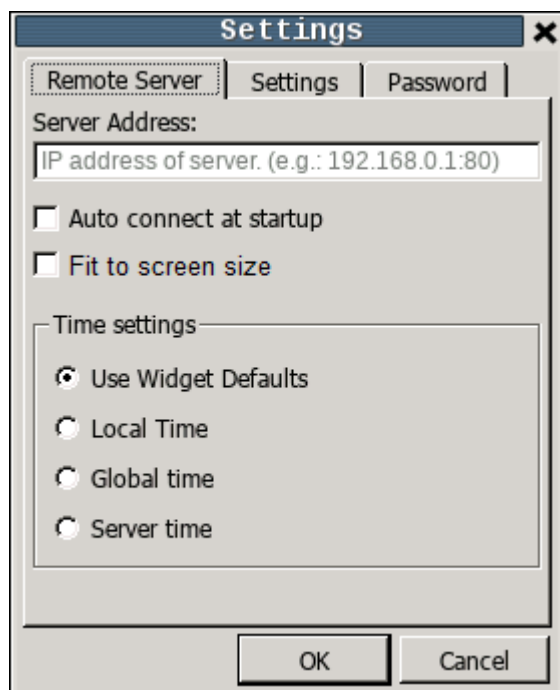
where:

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

## Client application on HMI

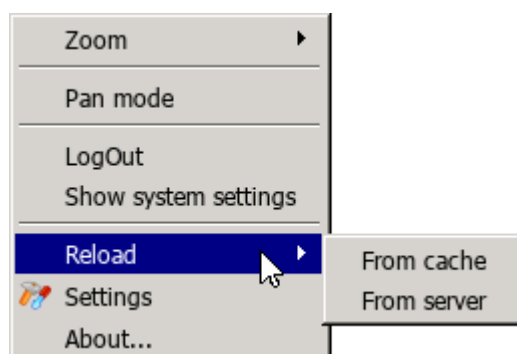
To run the HMI Client application on Linux HMI device:


1. From the **Run > Update Package** menu, create an Update Package and install the HMI Client application in to the HMI device (see ["Update package" on page 93](#) for additional information)
2. Type the server/device IP address in the Setting dialog that will be available when HMI device start (for example: <http://192.168.1.12>): HMI Client will connect to the server and the same graphical application running on the device will be loaded in the client window.



## Context Menu

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



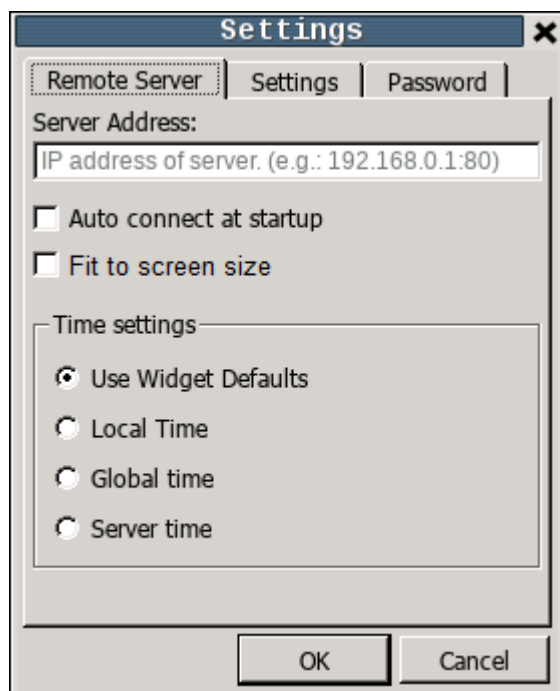
Option	Description
<b>Zoom</b>	Select view size at run time <ul style="list-style-type: none"> <li>• Zoom In</li> <li>• Zoom Out</li> <li>• Zoom 100%</li> </ul>
<b>Pane Mode</b>	Enables/disables pan mode after a zoom in
<b>Logout</b>	Logs off the current user.
<b>Show system settings</b>	Allow the HMI settings and the management of system components. See " <a href="#">System Settings</a> " on page 499 for details.
<b>Reload</b>	Reload remote project <ul style="list-style-type: none"> <li>• From cache</li> <li>• From server</li> </ul>
<b>Setting</b>	Open the HMI Client Settings. See " <a href="#">Settings and time zone options</a> " below for details  Could be password protected
<b>About</b>	Shows information about the HMI Client version.

## Settings and time zone options

In the **Settings** dialog you can configure client settings and decide how to display project time stamp information.



## Remote Server



### Connection settings

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

### Time settings

Parameter	Description
<b>Use Widget Defaults</b>	Displays time information according to the widget settings.
<b>Local Time</b>	Translates all timestamps in the project into the computer local time where the client is installed.
<b>Global Time</b>	Translates all timestamps in the project into UTC format.
<b>Server Time</b>	Translates all timestamps in the project into the same used by HMI device/server in order to show the same time.



**Important: Make sure you set the HMI RTC correct time zone and DST options.**

## Settings

The screenshot shows a 'Settings' dialog box with three tabs: 'Remote Server', 'Settings' (which is active), and 'Password'. The 'Settings' tab contains several configuration options. At the top, 'Context Menu Delay(s)' is set to 2. Below it, 'Show Busy Cursor' and 'Use Keypads' are both checked. The 'FTP settings' section shows the 'Port' set to 21. The 'HTTP settings' section includes a 'Protocols' dropdown set to 'CGI', 'Update Rate (s)' set to 1, 'Timeout (s)' set to 5, and two unchecked checkboxes for 'Reuse connection' and 'Enable compression'. 'OK' and 'Cancel' buttons are located at the bottom of the dialog.

### Interface Settings



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

### FTP settings

Parameter	Description
<b>Port</b>	FTP communication port

### HTTP settings

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

Parameter	Description
	 <p>Note: When enabled, this option may cause high latency if the proxy server does not immediately terminate old requests thus saturating connection sockets. This is often the case with 3G connections.</p>
Enable compression	<p>Compresses data to reduce download times. Default = disabled.</p>  <p><b>CAUTION: enabling this option could causes excessive CPU overhead.</b></p>
Time Settings	Used by the client to adapt the widget time stamp information.

## 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



# 13 Using the integrated FTP server

JMobile 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

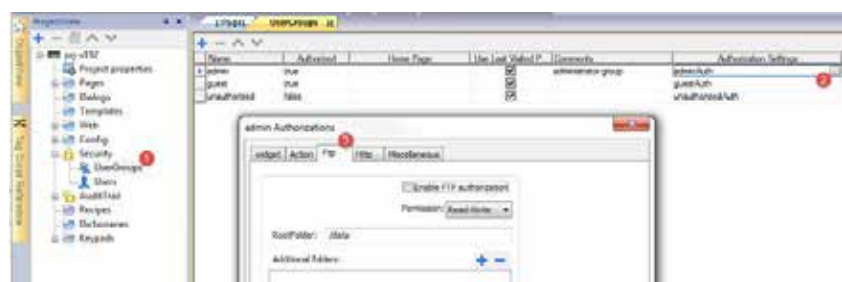
When User Management/Security is disabled use the following credentials for incoming connections:

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.



See ["Configuring groups and authorizations" on page 310](#) for details.



# 14    Using VNC for remote access

---

VNC is a remote control software which allows you to see and control the HMI application remotely using your local mouse and keyboard.

Remote access is particularly useful for administration and technical support. In order to use it you need to:

- start a server in the HMI device
- install a viewer on the remote device

---

<b>Starting VNC server on WinCE devices .....</b>	<b>220</b>
<b>Starting VNC server on Linux devices .....</b>	<b>221</b>
<b>Starting VNC viewer .....</b>	<b>221</b>

# Starting VNC server on WinCE devices

VNC server is a plug-in. It can be enabled and downloaded as part of the Runtime. ["Plug-in" on page 77](#).

## 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.

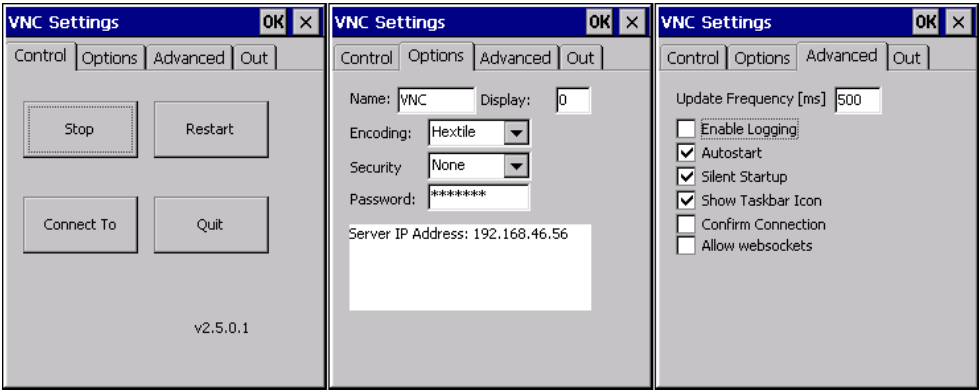
## Starting/stopping the VNC server



The VNC server is located in the folder `\Flash\qthm\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 71](#)

## VNC Options dialog

From the **VNC Options** dialog you can perform several tasks.



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

The screenshot shows the 'System Settings' app with the 'Services' tab selected. On the left, a list of settings categories is visible: Language, System, Logs, Date & Time, Network, Services (selected), Management, Display, Restart, Authentication, and EXIT. The main area shows 'Service Settings' with a list of services and their status (Enabled/Disabled). The 'VNC Service' is highlighted, and its specific settings are shown on the right.

Service Name	Status
Avahi Daemon	Enabled
Cloud Service	Disabled
Router Service	Enabled
SNMP Server	Disabled
SSH Server	Enabled
System Logger	Enabled
VNC Service	Enabled

**VNC Service Settings:**

Port	5900
Multiple clients	Enabled
View-only	Disabled
Encryption (compatible clients)	Disabled
Authentication	Disabled

An 'Edit' button is located at the bottom right of the VNC Service settings panel.

## Starting VNC viewer

No VNC viewer is provided as part of JMobile Suite.

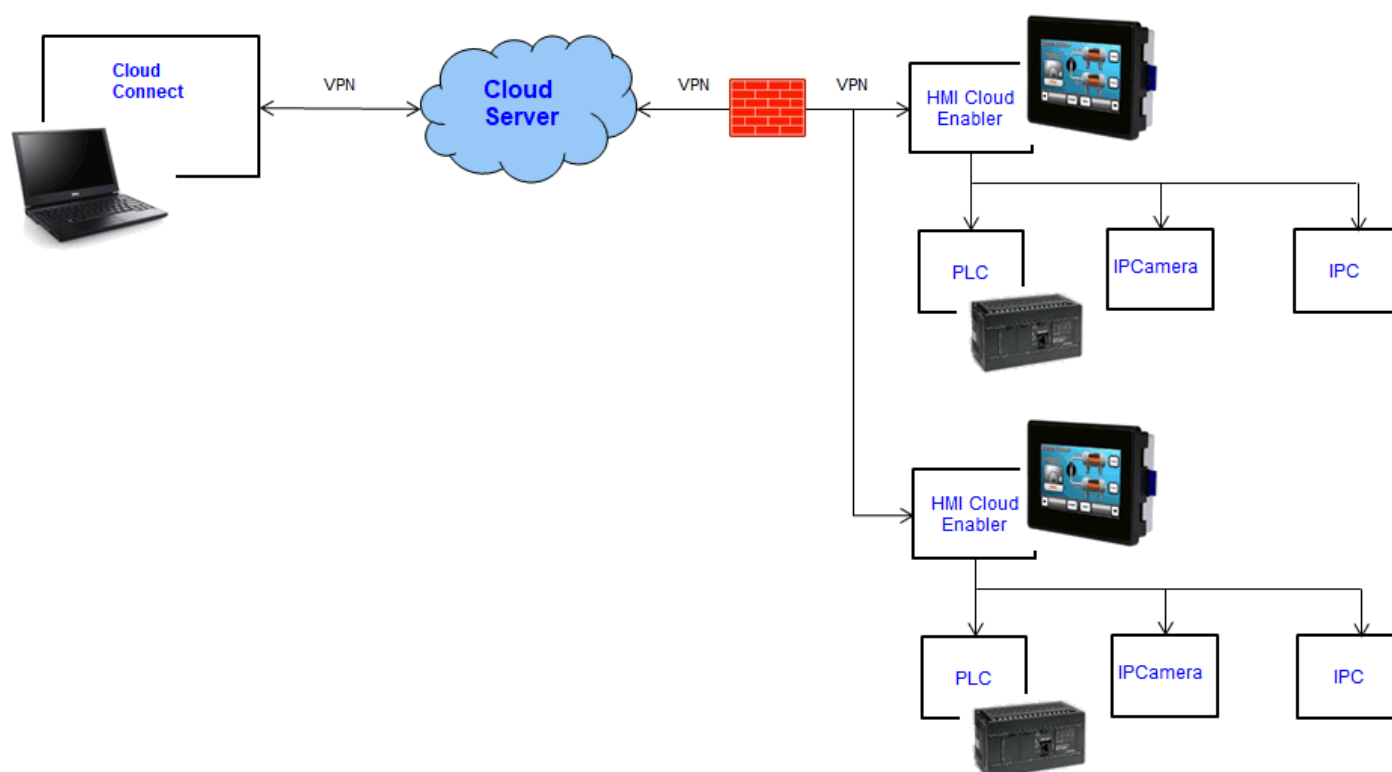
Many compatible VNC viewers are available for free download (for example, TightVNC).



# 15 Corvina Cloud

Corvina Cloud is a VPN-based solution that allows a seamless connection of diverse remote devices, called endpoints, to a centralized server through gateways. Users who have access to the Corvina Cloud can easily reach the gateways and the endpoints, provided they have the necessary access rights, using a PC application called Corvina Cloud Connect.

The diagram below presents a possible setup of the various components of the infrastructure, showing how they are interconnected:



See Corvina Cloud reference manual at [corvinacloud.exorint.net](http://corvinacloud.exorint.net) for additional details.



# 16 Alarms

---

The alarms handling system has been designed to provide alerts through pop-up messages, typically to display warning messages indicating any abnormal condition or malfunction in the system under control.

Whenever a bit changes, or the value of a tag exceeds a threshold set in the alarm configuration, a message is displayed. Specific actions can also be programmed to be executed when an alarm is triggered.



**Important: No default action is associated with any alarm.**

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

Alarms are configured in the Alarms Configuration Editor and, thus, are available for all the pages of the project. An alarm widget can display more than one alarm at a time, if sized appropriately. You can trigger the opening or closing of the Alarm window with an event.

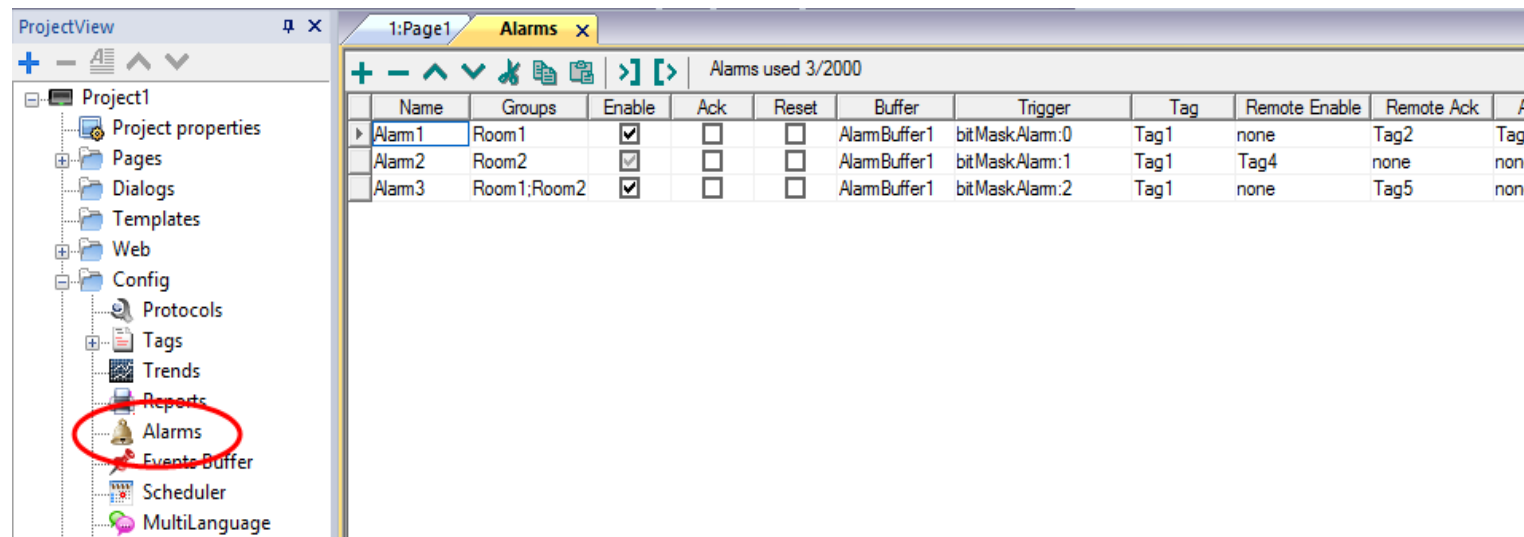
You work with alarms in the same way as you work with any other event. You may not want to display a dialog when an alarm is triggered and you can associate to it any other available action.

---

<b>Alarms Editor</b> .....	<b>226</b>
<b>Remote alarms acknowledge</b> .....	<b>228</b>
<b>Alarm state machine</b> .....	<b>229</b>
<b>Setting events</b> .....	<b>230</b>
<b>Active Alarms widget</b> .....	<b>232</b>
<b>Alarms History widget</b> .....	<b>237</b>
<b>Managing alarms at run time</b> .....	<b>237</b>
<b>Enable/disable alarms at run time</b> .....	<b>238</b>
<b>Displaying live alarm data</b> .....	<b>238</b>
<b>Exporting alarm buffers to .csv files</b> .....	<b>239</b>
<b>Exporting alarm configuration</b> .....	<b>239</b>


# Alarms Editor

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



## Adding an alarm

Click **+** to add an alarm.

Parameter	Description
<b>Name</b>	Name of alarm
<b>Groups</b>	Groups associated with the alarm. They can be used in widgets display filters.
<b>Enable</b>	Enable/disable triggering of alarm.  Alarms can be enabled or disabled at run time as well (see " <a href="#">Enable/disable alarms at run time</a> " on page 238 for details).
<b>Ack</b>	Enable/disable acknowledgment of alarm, if selected the operator must acknowledge the alarm once triggered to remove it from the <b>Active Alarm</b> widget.
<b>Reset</b>	Used with the <b>Ack</b> option, if selected, acknowledged alarms stay in the alarm list, labeled as <b>Not Triggered Acked</b> , until the operator presses the <b>Reset</b> button in the alarm widget.
<b>Buffer</b>	Buffer file where the alarm history will be saved.
<b>Trigger</b>	Triggering condition depending on alarm type: <ul style="list-style-type: none"> <li><b>limitAlarm</b>: alarm triggered when tag value exceeds its limits. The alarm is not triggered if the value reaches the limits.</li> <li><b>valueAlarm</b> alarm is triggered when tag value is equal to the configured value</li> </ul>

Parameter	Description
	<ul style="list-style-type: none"> <li>• <b>bitMaskAlarm</b>: 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 ",".</li> <li>• <b>deviationAlarm</b>: alarm triggered if the percentage of deviation of the tag value from the set point exceeds a set deviation. <math display="block"> Value_{now} - SetPoint  &gt; \left( \frac{deviation}{100} \times SetPoint \right)</math> </li> <li>• <b>protAlarm:BACN</b> : alarm is triggered from the remote BACnet device. <div data-bbox="384 689 453 763" data-label="Image"></div> This trigger mode is available only when the protocols configuration contains a BACnet device. Refer to the BACnet manual inside the "Communication Drivers" folder for a detailed description of BACnet alarm events. </li> </ul>
<b>Tag</b>	<p>Tag whose value will trigger the alarm when it exceeds the set limits.</p> <p>The alarm can refer to the value of this tag, or to the state of a bit if <b>bitMaskAlarm</b> has been selected as trigger.</p>
<b>Remote Enable</b>	<p>Tag used by the PLC to enable/disable the alarm.</p> <ul style="list-style-type: none"> <li>• Changing the enable status from the Alarms Widget will change the tag value</li> <li>• When the tag cannot be read (e.g. communication error) the alarm is disabled</li> <li>• No tags related to the alarm are refreshed when alarm is disabled.</li> </ul> <div data-bbox="300 1160 379 1227" data-label="Image"></div> <p>Tip: It could be useful to enable the logging of the alarm's enable flag</p> <div data-bbox="416 1227 1177 1861" data-label="Image"></div>
<b>Remote Ack</b>	<p>Tag used by the PLC to acknowledge the alarm. A transition of this tag from 0 to a non zero value is considered an acknowledgment request.</p>

Parameter	Description
	<p>Leave empty if remote acknowledgment is not required.</p> <p>See <a href="#">"Remote alarms acknowledge" below</a> for details.</p>
<b>Ack Notify</b>	<p>Tag used by the HMI device to notify when the alarm is acknowledged from the device or from the PLC.</p> <p>0 = set to this value when alarm is triggered</p> <p>1 = set to this value when alarm is acknowledged.</p>
<b>Action</b>	<p>Actions executed when the alarm is triggered. Additional conditions can be specified in the <b>Events</b> column.</p> <p>See <a href="#">"Setting events" on page 230</a> for details.</p>
<b>User Action</b>	<p>Actions executed when user press the action button in the active alarm widget.</p> <p>See <a href="#">""Active Alarms widget" on page 232</a> for details.</p>
<b>Description</b>	<p>Alarm description. This text supports the multiple language features and can be a combination of static and dynamic parts, where the dynamic portion includes one or more tag values.</p> <p>See <a href="#">"Displaying live alarm data" on page 238</a> for details.</p>
<b>Custom Field #</b>	<p>It is an additional alarm description that can be used to show additional information inside the alarms widgets. For example, could be an index to use to show a picture related with the alarm.</p>
<b>Color</b>	<p>Foreground and background colors of alarm rows based on the status of alarm.</p>
<b>AckBlink</b>	<p>Blinking for triggered alarms. If selected the alarm rows blinks until acknowledged. Only effective if <b>Ack</b> is selected.</p>
<b>Severity</b>	<p>Severity of the alarm. If multiple alarms are triggered simultaneously, actions will be executed based on severity settings.</p> <p>0 = not important</p> <p>1 = low</p> <p>2 = below normal</p> <p>3 = normal</p> <p>4 = above normal</p> <p>5 = high</p> <p>6 = critical</p>
<b>Events</b>	<p>Conditions in which the alarms are notified, logged or printed.</p> <p>See <a href="#">"Setting events" on page 230</a> for details.</p>

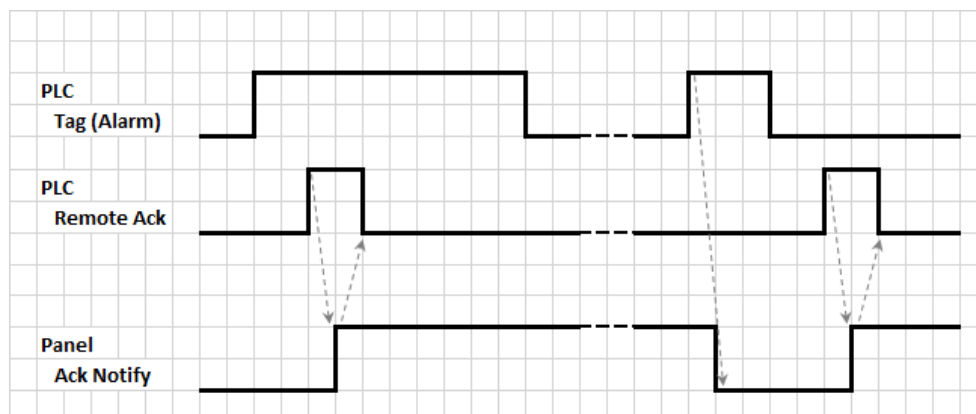
## Remote alarms acknowledge

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.



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** column > **Log** tab

Set conditions for which you want to store the specific event in an alarm history buffer.

The alarm history is logged in the Event Buffer.

## Executing actions

Path: **ProjectView** > **Config** > **Alarms** > **Events** column > **Actions** tab

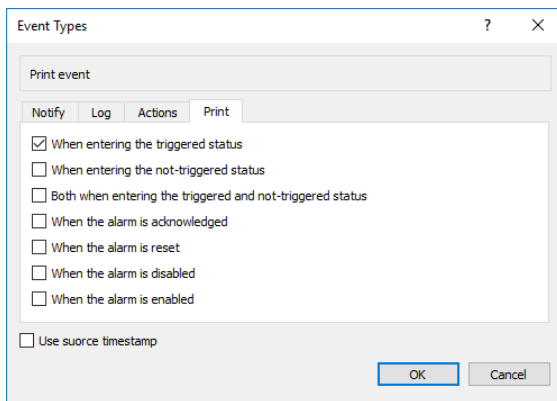
Set conditions under which the action(s), configured for the specific alarm, must be executed.

By default, actions are executed only when the alarm is triggered; other alarm states can also be set to execute actions.

## Print events

Path: **ProjectView** > **Config** > **Alarms** > **Events** column > **Print** tab

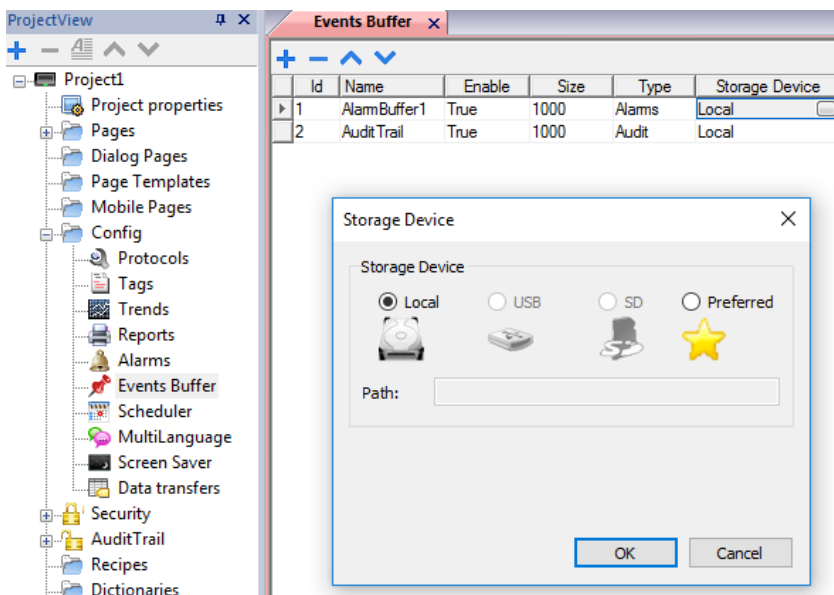
Set conditions for which you want to print the specific event



## Setting storage device

Path: **ProjectView** > **Config** > **Events Buffer** > **Storage Device** tab

1. Open the **Storage Device** dialog.
2. Select a device for event data storage.



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

### Use source timestamp

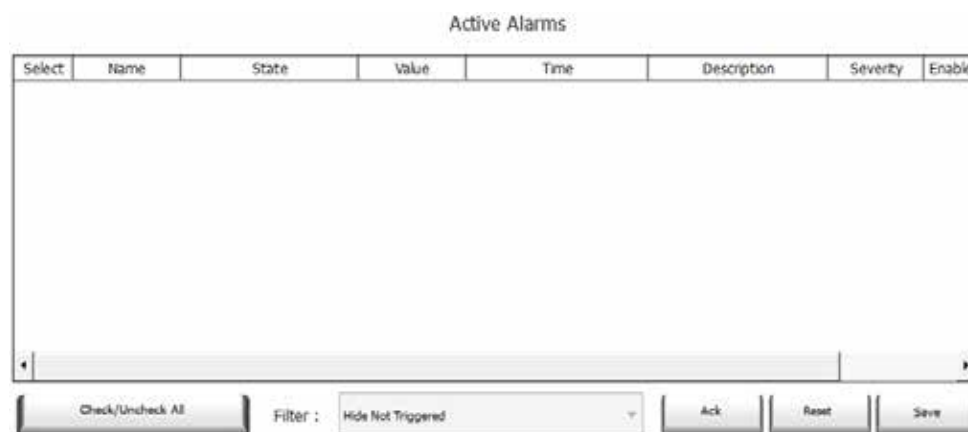
Events are stored with the timestamp of when the HMI device detects the event. When "Use source timestamp" is selected, the events are stored with the timestamp received from the remote device.



Available only for device's protocols that support this feature (OPC UA Client and BACnet)

## Active Alarms widget

You can insert the **Active Alarms** widget in a page to display the alarms and to acknowledge, reset or enable/disable them.



## Alarm filters

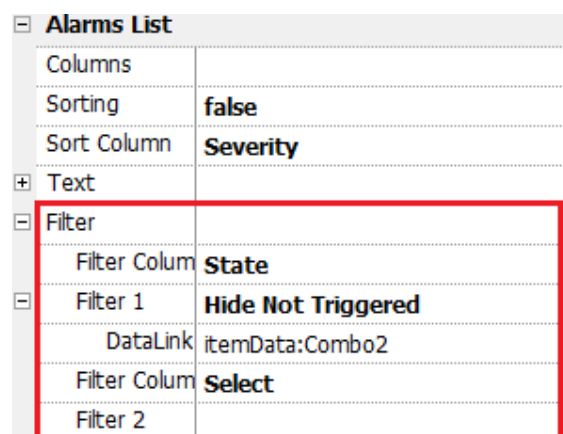
Path: **ActiveAlarm** widget > **Properties** pane > **Filter**

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.



## Setting filters

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

To set one of the two available filters:

1. Select **Filter Column 1** and choose the value to filter for (e.g.: Name, State, Time, Groups)
2. In **DataLink** attach a combo box widget. Use Shift+ left-click to select the combo box.
3. In the **Properties** pane select list property and open dialog to customize combo box values
4. In the combo box configuration dialog, specify **String List** and the regular expression to filter values.

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

### Filters first example

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

The screenshot shows the 'Active Alarms' widget interface and its configuration properties. The widget has a table with columns 'State' and 'Value'. Below the table are buttons for 'Ack', 'Reset', and 'Save', and a dropdown menu currently showing 'Not Triggered'. The 'Properties' panel on the right is expanded to show the 'Alarms List' configuration. Under the 'Filter' section, 'Filter Column 1' is set to 'State' and 'Filter 1' is set to '^((Not Triggered Aced|I'. 'Filter Column 2' is set to 'Value' and 'Filter 2' is set to '10'. These two settings are circled in red.

### 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]

The screenshot shows the 'Active Alarms' widget interface and its configuration properties. The widget has a table with columns 'State' and 'Value'. Below the table are buttons for 'Ack', 'Reset', and 'Save', and a dropdown menu currently showing 'Not Triggered'. The 'Properties' panel on the right is expanded to show the 'Alarms List' configuration. Under the 'Filter' section, 'Filter Column 1' is set to 'State' and 'Filter 1' is set to '^((Not Triggered Aced|I'. 'Filter Column 2' is set to 'Severity' and 'Filter 2' is set to '[3-6]'. These two settings are circled in red.

### Filters third example

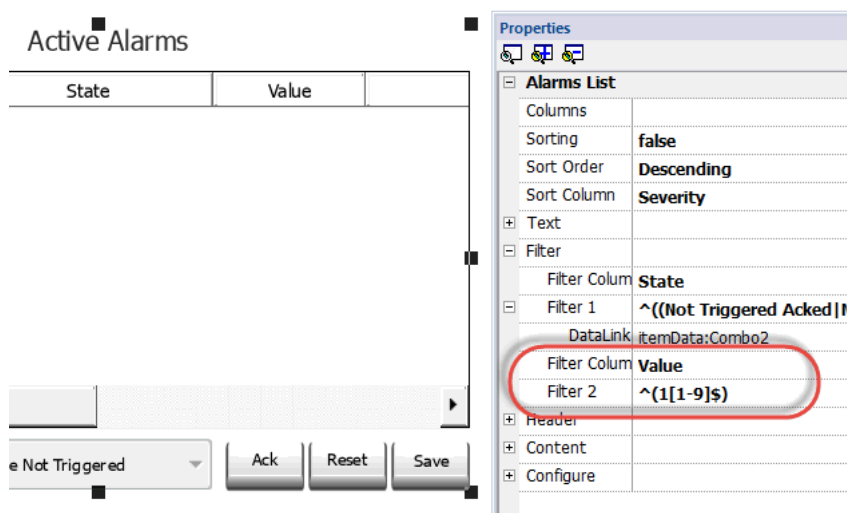
You want to show all alarms matching a value from 11 to 19. Then properties settings: **Filter column 2** = Severity, **Filter 2** = ^(1[1-9]\$)

Meaning:

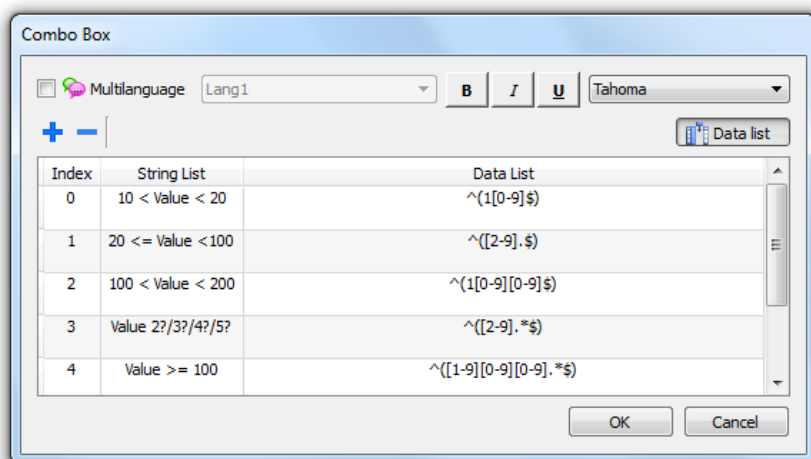
^ = match must starts from the beginning of the string

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

\$ = end of the comparison.



## Filters expression examples



Filter by	String list	Data list
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 run time 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 226](#) 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

Check/Uncheck All Filter : Show All Ack Reset Save



**WARNING:** If you are using an older converted project, you have to substitute the old Active Alarms Widget with the new one from the Widgets gallery



Note: The image can be modified from the Columns property of the Active Alarms widget

Table Column Editor

Columns + - ^ v

- Action
- Select
- Enable
- Name
- Groups
- State
- Value
- Time
- Description
- Severity

Col 0 Info

Header	Action
Value	alUserAction
Width	100
Type	Image
Visible	true
Image path	images\action.png

OK Cancel

Alarms List : ActiveAlarms

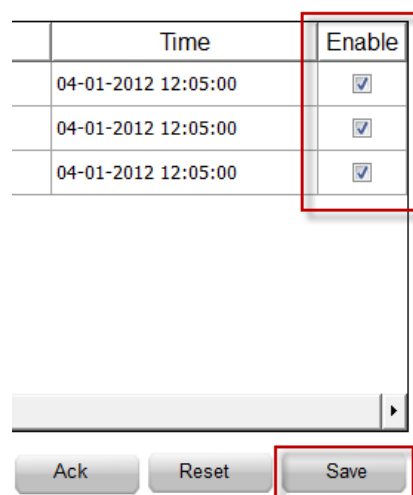
Columns	
Sorting	false
Sort Order	Descending
Sort Column	Severity
Text	
Filter	
Header	
Content	
Configure	
General	
Position	

## Enable/Disable Alarms

At runtime the Alarms Widget can be used to enable or disable the alarms.

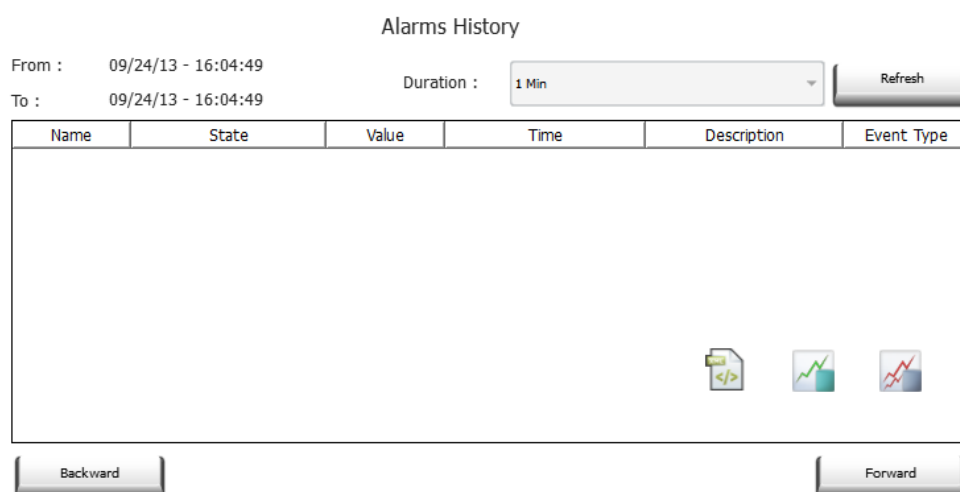


Saves changes made in the **Enable** column in the alarm widget. This action is used with the **Save** button in the alarm widget.



## Alarms History widget

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



## Attaching widget to buffer

Path: **AlarmHistory** widget> **Properties** pane> **Buffer** > **EventBuffer**

In **Properties** pane > **Event** select the **Event Buffer** from which the alarm list is retrieved

## Managing alarms at run time

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 232](#) 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 run time

You can enable or disable the alarms at run time.

To enable an alarm select the **Enable** option in the alarm widget.

Disabled alarms are not triggered and therefore not displayed at run time.

Select	M	Source Value	State	Date	Time	Enable
<input type="checkbox"/>	Alarm1	23	Not Triggered Not Acked	25-01-2011	16:59:31	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Alarm2	23	Not Triggered Not Acked	25-01-2011	16:59:31	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Alarm3	23	Not Triggered Not Acked	25-01-2011	16:59:31	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Alarm4	23	Not Triggered Not Acked	25-01-2011	16:59:31	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Alarm5	23	Not Triggered Not Acked	25-01-2011	16:59:31	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Alarm6	23	Not Triggered Not Acked	25-01-2011	16:59:31	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Alarm7	23	Not Triggered Not Acked	25-01-2011	16:59:32	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Alarm8	23	Not Triggered Not Acked	25-01-2011	16:59:32	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Alarm9	23	Not Triggered Not Acked	25-01-2011	16:59:32	<input checked="" type="checkbox"/>

Check/Uncheck All Filter: Show All Ack Reset Save



Note: Alarms can be configured to be enable/disable even from the PLC. See [Alarm Configuration Editor](#) for details.

## Displaying live alarm data

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

Both in the Active Alarms and in the History Alarms widget you can set the alarm description to display live tag data.

Id	Name	Enable	Ack	Reset	Tag	Buffer	Trigger	Action	Description
1	Alarm1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Tag1	AlarmBuffer1	bitMaskAlarm:	ShowDialog	Alarm 1 Tag Value is [Tag1]
2	Alarm2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Tag1	AlarmBuffer1	bitMaskAlarm:1	ShowDialog	Alarm 2 Tag Value is [Tag2]
3	Alarm3	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Tag1	AlarmBuffer1	bitMaskAlarm:1	ShowDialog	Alarm 3 Tag Value is [Tag3]

To show the tag value, set a placeholder in **Description** entering the tag name in square brackets, for example "[Tag1]". At run time, in **Description** column of Active Alarms widget the current value of the tag will be displayed.

There are two type of placeholders

- [TagName]  
When alarm is triggered, tag value is read and continuously updated
- [!TagName]  
When alarm is triggered, tag value is read and frozen

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)



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

Use '\', even when the tag label contains square brackets. For example, to display the live tag value of tag

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

## Example of Alarm widget

Select	Id	Source Value	State	Description	Date
<input type="checkbox"/>	Alarm1	123	Triggered Not Acked	Alarm 1 Tag value is 123	25-01-2011
<input type="checkbox"/>	Alarm2	1234	Triggered Not Acked	Alarm 2 Tag value is 1234	25-01-2011
<input type="checkbox"/>	Alarm3	456	Triggered Not Acked	Alarm 3 Tag value is 456	25-01-2011
<input type="checkbox"/>	Alarm4	987	Triggered Not Acked	Alarm 4 Tag value is 987	25-01-2011
<input type="checkbox"/>	Alarm5	555	Triggered Not Acked		25-01-2011
<input type="checkbox"/>	Alarm6	1234	Triggered Not Acked		25-01-2011
<input type="checkbox"/>	Alarm7	1234	Triggered Not Acked		25-01-2011

Filter :

Number of live tags that can be used inside each alarm's description depends on size of used tags. JMobile Studio 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	

## Exporting alarm buffers to .csv files

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

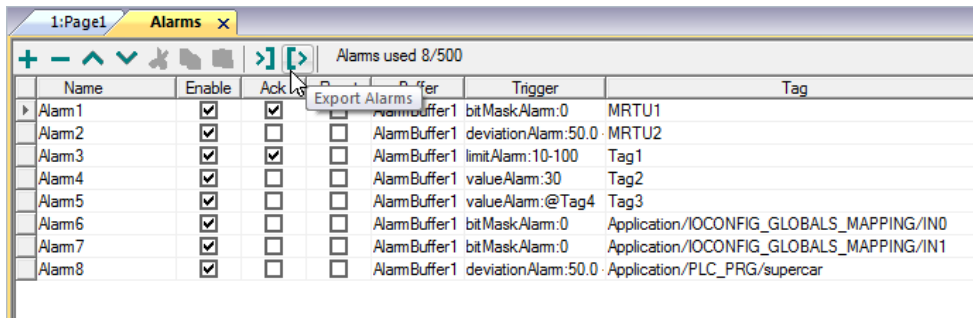
See "[System actions](#)" on page 184 for details.



**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

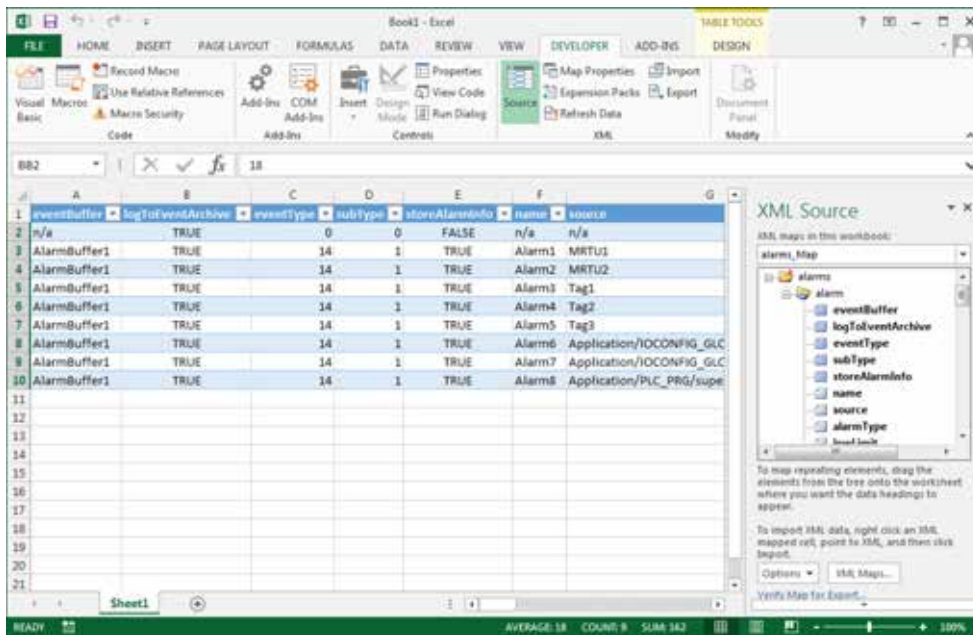
Path: **ProjectView** > **Config** > double-click **Alarms**



Name	Enable	Ack	AlarmBuffer	Trigger	Tag
Alarm1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	AlarmBuffer1	bitMaskAlarm:0	MRTU1
Alarm2	<input checked="" type="checkbox"/>	<input type="checkbox"/>	AlarmBuffer1	deviationAlarm:50.0	MRTU2
Alarm3	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	AlarmBuffer1	limitAlarm:10-100	Tag1
Alarm4	<input checked="" type="checkbox"/>	<input type="checkbox"/>	AlarmBuffer1	valueAlarm:30	Tag2
Alarm5	<input checked="" type="checkbox"/>	<input type="checkbox"/>	AlarmBuffer1	valueAlarm:@Tag4	Tag3
Alarm6	<input checked="" type="checkbox"/>	<input type="checkbox"/>	AlarmBuffer1	bitMaskAlarm:0	Application/IOCONFIG_GLOBALS_MAPPING/IN0
Alarm7	<input checked="" type="checkbox"/>	<input type="checkbox"/>	AlarmBuffer1	bitMaskAlarm:0	Application/IOCONFIG_GLOBALS_MAPPING/IN1
Alarm8	<input checked="" type="checkbox"/>	<input type="checkbox"/>	AlarmBuffer1	deviationAlarm:50.0	Application/PLC_PRG/supercar

Click the **Export Alarms** button: the alarms configuration table is exported into an .xml file.

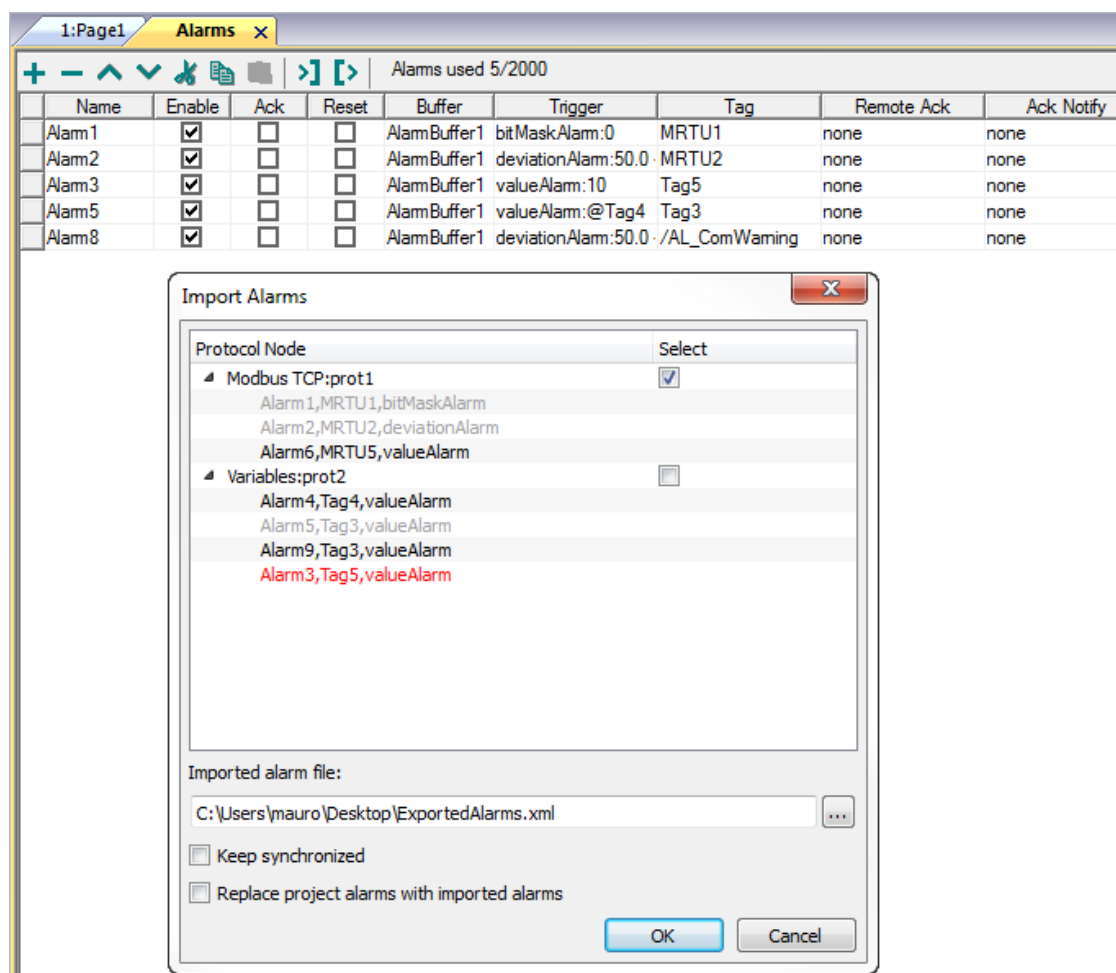
You can edit the resulting .xml file using third part tools (for example, Microsoft Excel).



eventBuffer	logToEventArchive	eventType	subType	storeAlarmInfo	name	source
n/a	TRUE	0	0	FALSE	n/a	n/a
AlarmBuffer1	TRUE	14	1	TRUE	Alarm1	MRTU1
AlarmBuffer1	TRUE	14	1	TRUE	Alarm2	MRTU2
AlarmBuffer1	TRUE	14	1	TRUE	Alarm3	Tag1
AlarmBuffer1	TRUE	14	1	TRUE	Alarm4	Tag2
AlarmBuffer1	TRUE	14	1	TRUE	Alarm5	Tag3
AlarmBuffer1	TRUE	14	1	TRUE	Alarm6	Application/IOCONFIG_GLC
AlarmBuffer1	TRUE	14	1	TRUE	Alarm7	Application/IOCONFIG_GLC
AlarmBuffer1	TRUE	14	1	TRUE	Alarm8	Application/PLC_PRG/supe

## Importing alarm configuration

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



1. Click the **Import Alarms** button and select the .xml file from which to import the alarms configuration: the **Import Alarms** dialog is displayed.
2. Select the group of alarms to import and click **OK** to confirm.

Differences are highlighted in the **Import Alarms** dialog using different colors

Color	Description
Black	This is a new alarm and it will be imported
Red	This alarm has not been found and will be removed (only if check "Replace project alarms with imported alarms" is checked)
Blue	This alarm has been modified and will be updated.
Gray	This alarm is already part of the project and will be skipped.

## Automatic synchronization

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

Whenever changes occur in the alarms configuration, the file will be automatically updated in silent mode.



Tip: Enable this function when the alarm file is managed by a different tool (for example, PLC programming software) as well as by JMobile Studio.

# 17 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 JMobile Studio workspace; the user can specify default values for each element of the data records. In JMobile 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 JMobile 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 JMobile HMI Runtime.

See ["Recipe actions" on page 178](#) 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.

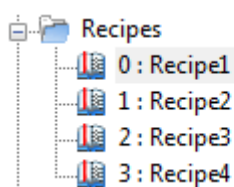
Managing recipes .....	243
Configuring a recipe widget .....	246
Recipe status .....	247
Uploading/downloading a recipe .....	248
Backup and restore recipes data .....	249

## Managing recipes

### Creating a recipe

To create a recipe for your project:

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



## Recipe editor

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

index	Element Name	Tag	Fill Tank 1	Fill Tank 3	Fill Tank 5	Fill Tank 7	Fill Tank 1	Empty Tank	Empty Tank	Empty Tank 75	Em
0	Home Valve	Recipe_HomeV: 1	1	1	1	1	0	0	0	0	0
1	Truck Valve	Recipe_TruckV: 0	0	0	0	0	1	1	1	1	1
2	Fill Flow Meter	Recipe_FillFlow: 15	35	50	75	100	75	50	25		15
3	Empty Flow Meter	Recipe_EmptyFl: 0	0	0	0	0	25	50	75		85
4	Chemical1	Recipe_Chemic: 0	0	0	0	0	0	0	0		0
5	Chemical2	Recipe_Chemic: 0	0	0	0	0	0	0	0		0

## Configuring recipe properties

In the **Properties** pane of each recipe you set the following parameters:

Parameter	Description
<b>Recipe Name</b>	Name of the recipe
<b>Number of sets</b>	Number of values sets for each recipe element. Each set has a different configurable name.

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

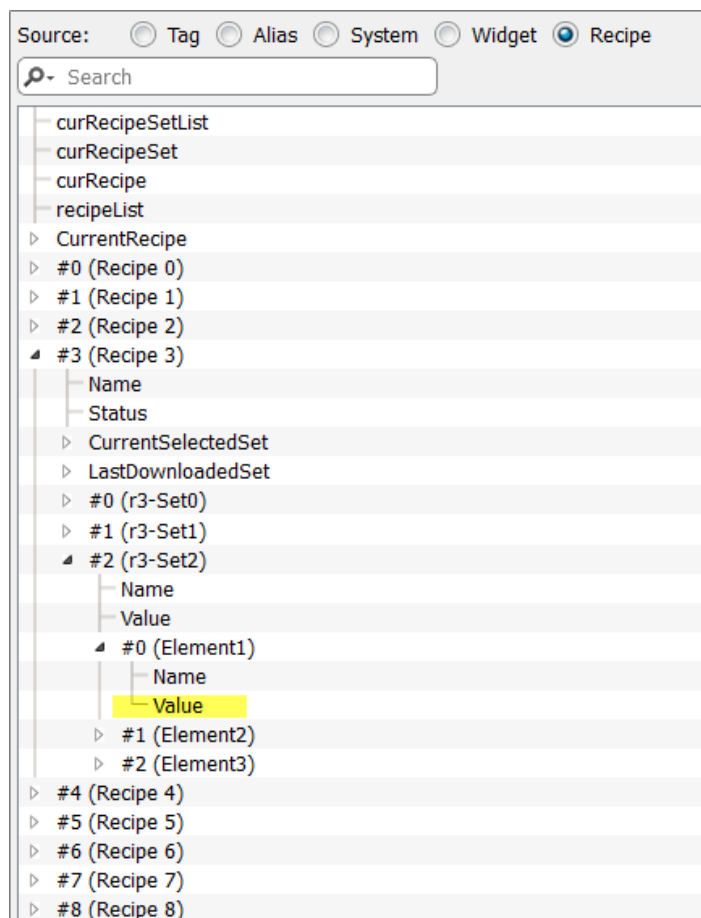
## Setting up a recipe

1. Click **+** to add an element of the recipe.
2. Link the tags to each recipe element.

## Defining recipe fields

Create a recipe field in the page using a numeric widget and attaching it to a recipe item after selecting Recipe as the Source.





In the **Attach to** dialog you have the choice of all the different recipe variables, such as:


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

When numeric widgets are defined as read/write, the default recipe data can be edited at run time. These new values are stored in a separate file as modified recipe data.



Note: Since JavaScript API functions are used, the recipe elements and sets can be referenced by name or by position. To avoid ambiguity between names and index, the names of the recipe elements and sets must include at least one alphanumeric character.

## Storing recipe data

In the Recipe Editor click the storage type icon  to select where to store recipe data: the **Storage Device** dialog is displayed.



For USB drive and SD card storage you can provide the folder location.



**WARNING:** Recipe configuration files are created automatically when the project is saved and stored in the data subfolder of the project. To use external storage devices, you need to copy this folder into the external device. Note that you have the responsibility to manage the data folder inside external devices. Even dynamic files are not deleted when project is updated using the “Delete dynamic file” option.



**Important:** You can add a subfolder but you must not rename the "data" subfolder.

## Import/Export recipes

To import/export the recipes configuration of your project:

In **ProjectView** right-click **Recipes** and select **Export Recipe** or **Import Recipe**

The following formats are supported for import:

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



Note: Use the Unicode Text file format when you import a file modified using Microsoft® Excel®.

## Configuring a recipe widget

You can choose one of the two recipe widgets available in the **Widget Gallery**:

- **Recipe set:** allows you to select a recipe set for upload or download. See ["Uploading/downloading a recipe" on page 248](#)
- **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.

**Recipe Set**

Recipe Set

Download

Upload

**Recipe Menu**

Recipe

Recipe Set

Download

Upload

## Configuring the Recipe Set widget

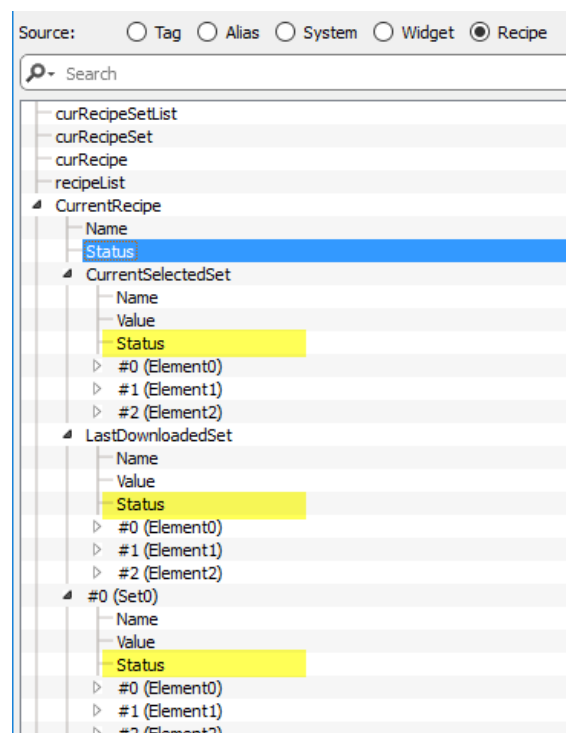
In the **Properties** pane of each **Recipe Set** widget set the following parameter:

Parameter	Description
Recipe Name	Name of the recipe

## Recipe status

Each recipe contains two kinds of status parameters

- Recipe Status (blue in the below picture)  
Give information regarding the last download or upload operation
- DataSet Status (yellow in the below picture)  
Give information of modified datasets



## Recipe Status

After every recipe upload or download, or recipe set modification, the **Recipe Status** parameters contain a value with the result of the operation.

Code	Function	Description
0	Set modified	Selected set changed
1	Download triggered	Download request triggered
2	Download Done	Download action completed
3	Download Error	Error during download (for example, unknown set, unknown recipe, controller not ready, Tags write failed etc.)
4	Upload triggered	Upload request triggered
5	Upload done	Upload action completed
6	Upload Error	Error during upload - same as for download
7	General Error	General error (for example, data not available)

## DataSet Status

The status of each data set indicates that it has been changed. This information may be useful to not forget to download the recipe to synchronize the PLC. Both download or upload operations will reset the **DataSet Status** to 0.

Code	Function	Description
0	Synchronized	User synchronized PLC with the dataset values
1	Modified	User modified some values of the dataset



Note: After a device startup or a recipe reset/restore, all status values will reset to 0.

# Uploading/downloading a recipe

## Uploading a recipe

You upload a recipe to an HMI device using a recipe widget and the **UpLoadRecipe**, **UpLoadCurRecipe** action in one of the following ways:

- attach the action to an event of a button or a switch (see [""Attach to" parameters" on page 46](#) for details)
- configure the action in an alarm action list (see ["Alarm actions" on page 160](#) for details)
- configure the action in a scheduler action list (see ["Scheduling events at run time" on page 294](#) 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 178](#)

## 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 178](#) for details.



# 18 Trends

---

Trends allow you to sample and record the values of specified tags according to specific sampling conditions. The trend function includes trend acquisition and trend display.

Trend acquisition parameters are set in the Trend editor so that data can be stored. Stored data can then be displayed in a graphical format using a trend widget.

---

<b>Data logging</b> .....	<b>252</b>
<b>Exporting trend buffer data</b> .....	<b>254</b>
<b>Trend widgets</b> .....	<b>255</b>
<b>History trends</b> .....	<b>257</b>
<b>Trend widget properties</b> .....	<b>259</b>
<b>Trend widget gestures</b> .....	<b>260</b>
<b>Values outside range or invalid</b> .....	<b>261</b>
<b>Showing trend values</b> .....	<b>262</b>
<b>Scatter diagram widget</b> .....	<b>263</b>
<b>Table trend widget</b> .....	<b>264</b>

# Data logging

Data can be logged and stored to HMI memory. Data logging allows you to store the values of a group of tags all at the same time to a buffer. Data logging can be triggered by a timer or by a dedicated tag. Logged data can be exported to a .csv file or displayed using the historical trend widget. Logged data can be saved locally on a USB device or SD card, or on any available custom network folder.



**WARNING:** The operation with removable memory devices (USB Flash drives, SD memory cards) containing a very large number of files may result in a decrease of system performance.



**WARNING:** The max number of files inside a SD memory card depends on the type of formatting (e.g. FAT32 max 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.



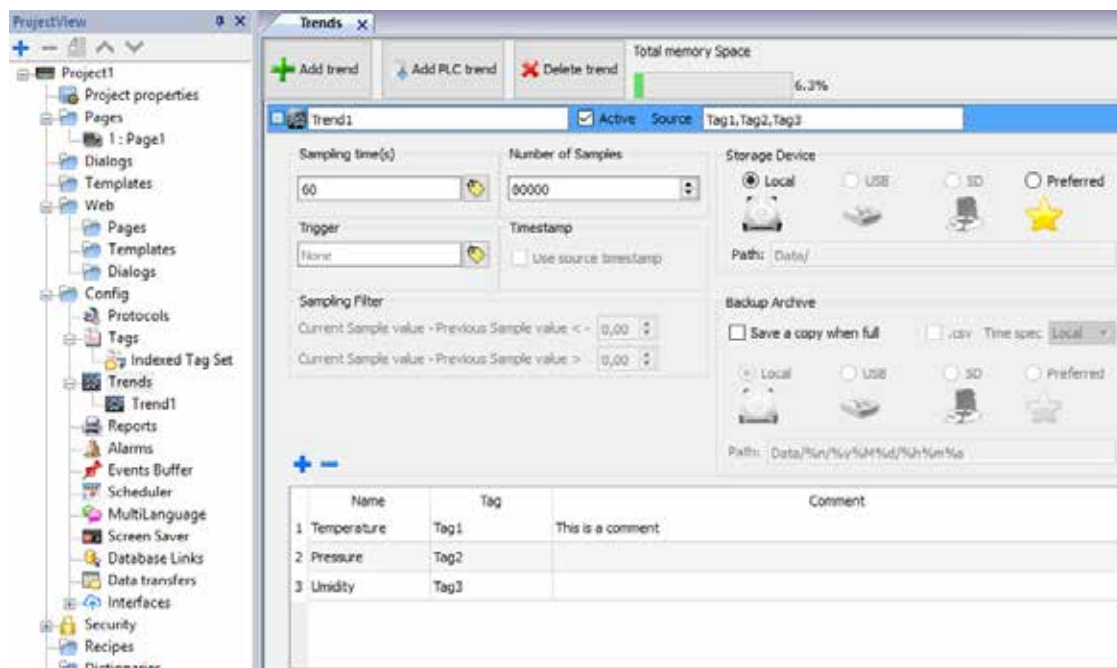
**WARNING:** If the data/time is moved back, the samples with invalid date/time are removed from the trend buffer. When system detects that data/time is invalid (e.g. battery low), a popup is shown to advise the user and the date/time of the last sample is used to avoid losing data.

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.





Element	Description
<b>Total memory Space</b>	<p>Memory currently used by the trend buffer. See <a href="#">"Table of functions and limits" on page 566</a> for maximum number of samples allowed for project.</p> <p>This percentage is calculated as follows:</p> $\text{Total Memory Space} = \frac{\text{Total Number of Samples used in the Project}}{\text{Max Number of Samples allowed for a Project}} * 100$
<b>Trend Name</b>	Name of trend that will be displayed in the window property pane.
<b>Active</b>	<p>When enabled, the trend runs by default at system startup.</p> <p> Note: Trends cannot be activated at run time.</p>
<b>Source</b>	Tags sampled by the trend.
<b>Sampling Time (s)</b>	Sampling interval in seconds.
<b>Trigger</b>	<p>Tag triggering the sample. When the value of this tag changes, a sample is collected.</p> <p> Note: Trigger and Source can refer to the same tag.</p>
<b>Timestamp</b>	<p>When checked, use the time stamp provided from the remote device.</p> <p>Available:</p> <ul style="list-style-type: none"> <li>only for device's protocols that support this feature (OPC UA Client and BACnet)</li> <li>when trend buffer is configured to with a single tag</li> </ul>
<b>Number of Samples</b>	Buffer size.
<b>Storage Device</b>	Where trend buffer data will be stored.
<b>Backup Archive</b>	<p>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.</p> <p><b>.csv:</b> Backup data using text CSV format.</p> <p><b>Time Spec:</b> Dump to CSV the time in local or UTC</p> <p><b>Path:</b> Where trend buffer data will be copied</p> <p>The below wild cards are supported</p> <ul style="list-style-type: none"> <li>%n = Trend name</li> <li>%y = Year</li> </ul>

Element	Description
	<ul style="list-style-type: none"> <li>• %M = Month</li> <li>• %d = Day</li> <li>• %h = Hour</li> <li>• %m = Minutes</li> <li>• %s = Seconds</li> </ul>
<b>Sampling Filter / Trigger Filter</b>	<p>This parameter allows to specify a dynamic filter if required.</p> <p>When sampling is done on time basis the offset is applied to the sampled Tag value. If the new value exceeds the specified limits the new value is considered valid and stored, otherwise the new record will retain the previous saved value.</p> <p>When sampling is done on trigger the offset is applied to the trigger Tag value. If the trigger Tag value change exceeds the specified limits a new sample is taken and stored, otherwise no sampling will be done.</p>
<b>Sampled tags table</b>	<p><b>Name:</b> name of trend</p> <p><b>Tag:</b> tag to be sampled.</p> <p><b>Comment:</b> trend description</p>

## Exporting trend buffer data

Use the **DumpTrend** action to export trend buffer data to a .csv file. See "[DumpTrend](#)" on page 184 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	<b>Timestamp</b>	<b>Tag1</b>	<b>4 Tag2</b>	<b>8 Tag3</b>			<b>3</b>
2		<b>Value</b>	<b>Quality</b>	<b>Value</b>	<b>Quality</b>	<b>Value</b>	<b>Quality</b>
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	<b>Date</b>	<b>Time</b>	<b>Name1(int)</b>	<b>Quality</b>	<b>Name2(int)</b>	<b>Quality</b>	<b>Name3(boolean)</b>	<b>Quality</b>
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

## Trend widgets

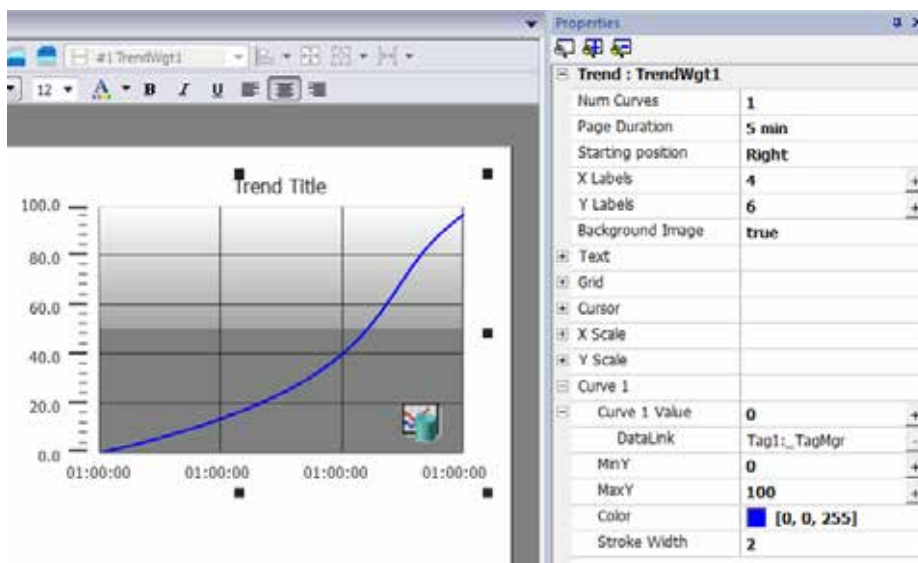
Data logged by the HMI device can be displayed in graphical format using trend widgets.

### RealTime trend widget

The real-time trend widget can be used to display the changes of value of a tag. Data is not stored in a trend buffer and cannot be retrieved for later analysis.


To display a real-time trend:

1. Drag and drop the **RealTime Trend** widget from the widget gallery to the page.



2. Attach the tag that you want to sample to the **Curve *n* Value**. Data is always plotted against time.

## RealTime trend widget properties

Property	Description
<b>Num Curves</b>	Number of trend curves to be displayed (Max. 5)
<b>Page Duration</b>	Time range of the x-axis.  Tip: You can attach a <b>Date Time</b> combo widget to the <b>Page Duration</b> property and dynamically change page duration at run time.
<b>Starting Position</b>	Specifies the starting point of the curve when the page is opened.
<b>X Labels</b>	Number of ticks on the x-axis scale
<b>Y Labels</b>	Number of ticks in the y-axis scale.
<b>Text</b>	Trend title and font properties (font size, label, etc.)
<b>Grid</b>	Properties of grid presentation (colors)
<b>Cursor</b>	Properties of cursor presentation (enable and color)
<b>X Scale</b>	Properties of X Scale presentation
<b>Y Scale</b>	Properties of Y Scale presentation
<b>Curve "n"</b>	Tag that will be plotted in the trend widget. See <a href="#">"Trend widget properties" on page 259</a> for details.  You can set the minimum and maximum of the curves ( <b>MinY</b> , <b>MaxY</b> ). You can attach a tag to minimum and maximum properties. This enhances the ability to change the minimum and maximum values dynamically at run time.

## Scaling data

Tag values can be scaled using the X Forms in the **Attach to** dialog. See [""Attach to" parameters" on page 46](#) for details.

## History trends

Trend data stored in trend buffers 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.

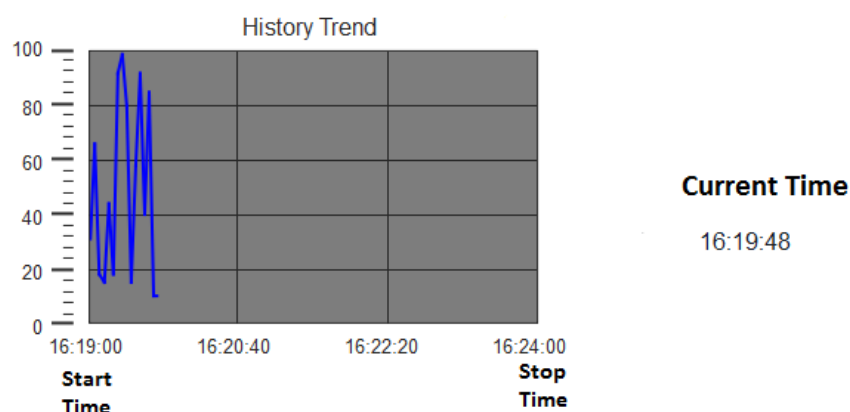
See ["Data logging" on page 252](#) for details on how to create a trend buffer.

### History Trend widget

History Trend widget displays in graphical format the content of a trend buffer.

Start time is the current time and stop time will be the current time plus the duration of the window. The curve starts from the left and progresses to the right, data is automatically refreshed during a certain interval time, until the stop time.

When the curve reaches the stop time, the curve will scroll left and the update of the curve will continue until it again reaches the stop time. At that moment a new scroll is automatically performed and the process repeats.

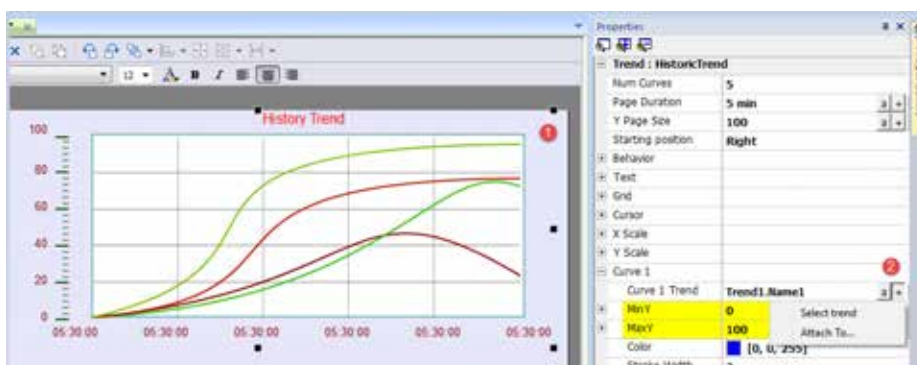


History trends require a proper configuration of trend buffer.

See ["Data logging" on page 252](#) for details on how to work in the Trend editor.

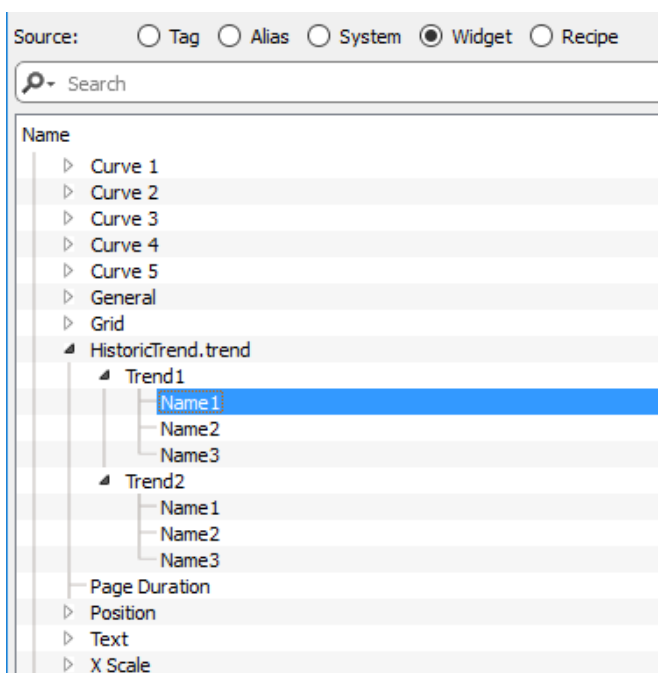
## Configuring the History Trend widget

1. From the **Trends/Graphs** section of the **Widget Gallery**, drag and drop the **History Trend** widget to the page.



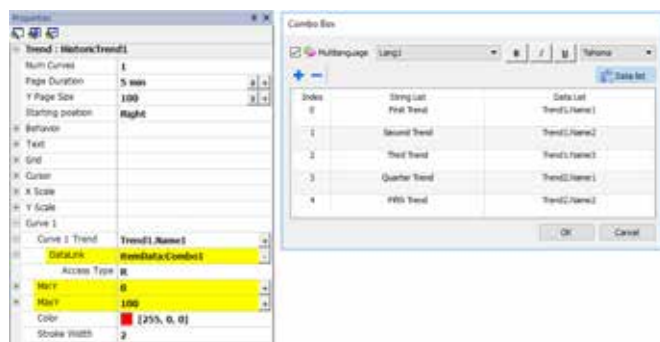
2. In the **Properties** pane, attach the trend buffer to be plotted in the widget.

You can **Select trend** directly from the HistoricTrend widget



or use the **Attach To...** tag to identify the tag that will contains the trend name to show. Using the attach to tag, the curve to show can be select from the JMobile HMI Runtime

For example, connect to `CurrentData` property of a combo box widget configured with the list of trends to show



## Trend widget properties

Some Trend widget properties are only available when the Properties pane is in Advanced view.

### Request Samples

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



Tip: You normally do not need to modify the default value. Adjust it to fine tune performances in the trend widget refresh, especially when working with remote clients.

### Color bands

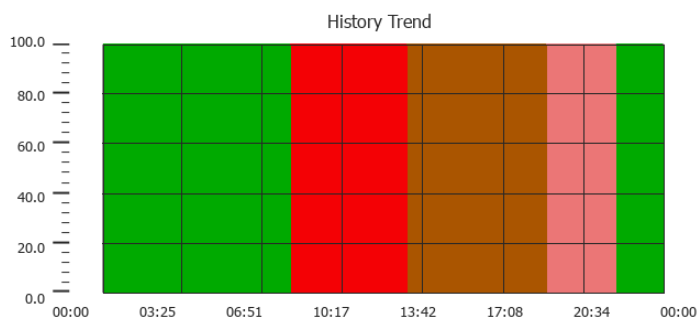
Use the color bands configuration to customize your graphs background, for example to make certain days or hours stand out (weekends, night hours, etc.).

1. In the **Properties** pane, in **Color Bands** property click **+**: the **Configure Bands** window appears.
2. Click **+** to add as many colors you need.
3. Select multiple cells and click on a color band to assign the color to the selected range of cells.



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

## Calendar color bands example



## 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**



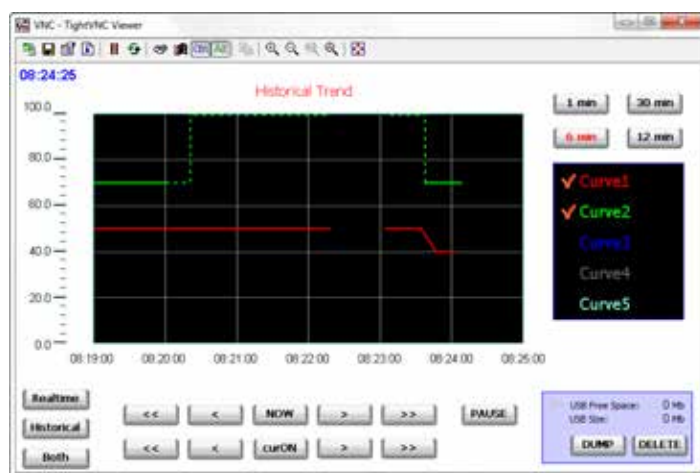
Note: In order to support gestures on Y axis, Min/Max properties of the trend widget must be linked to Min/Max values of Behavior parameters (default for new trends).



Properties		
<b>Trend : RealtimeTrend</b>		
Num Curves	1	
Page Duration	5 min	+
Y Page Size	100	+
Starting position	Right	
<b>Behavior</b>		
Min Y	0	+
Max Y	100	+
X Labels	4	+
Y Labels	6	+
Background Image	true	
<b>Text</b>		
<b>Grid</b>		
<b>Cursor</b>		
<b>X Scale</b>		
<b>Y Scale</b>		
Min	0	+
DataLink	y0:RealtimeTrend.wnd	-
Access Type	R	
Max	100	+
DataLink	y1:RealtimeTrend.wnd	-
Access Type	R	

## Values outside range or invalid

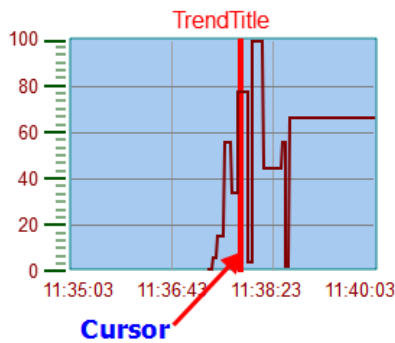
When trend value goes beyond the limits set for the trend widget, a dotted line is displayed. When the value of the tag is not available, for example the controller device is offline, no curve is drawn.



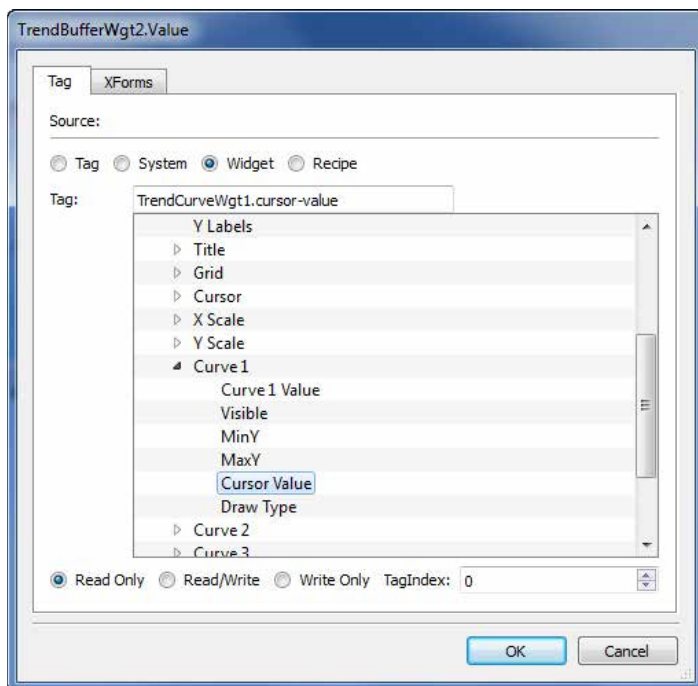
## Showing trend values

Trend cursor displays the trend value at a specific point.

Use the actions **ShowTrendCursor** and **ScrollTrendCursor** to enable the trend cursor and move it to the required point to get the value of the curve at that particular point in time.

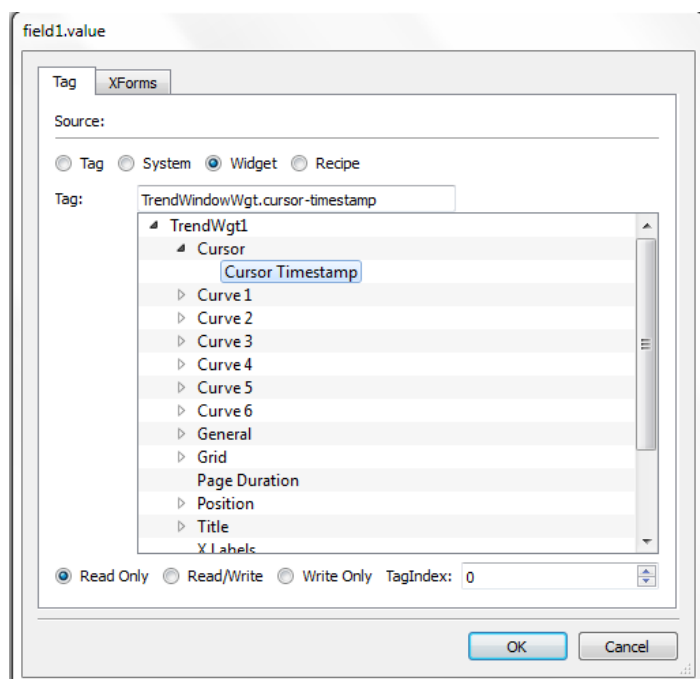


To display the value of the trend cursor on the page, define a numeric field and attach it to the **Cursor Value** widget tag.



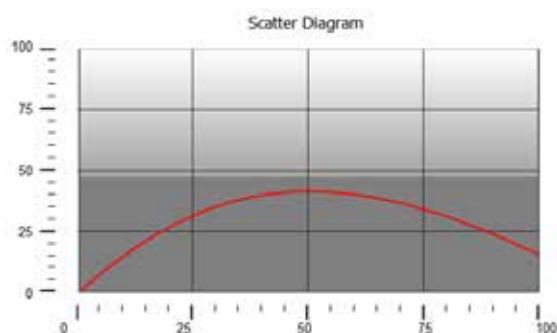
In this example the Y axis value of the cursor is displayed.

To display the trend timestamp at the position of the cursor, define a numeric field and attach it to **Cursor Timestamp** widget tag.



## 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.

## 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      **Refresh**

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

**Backward**      **Forward**

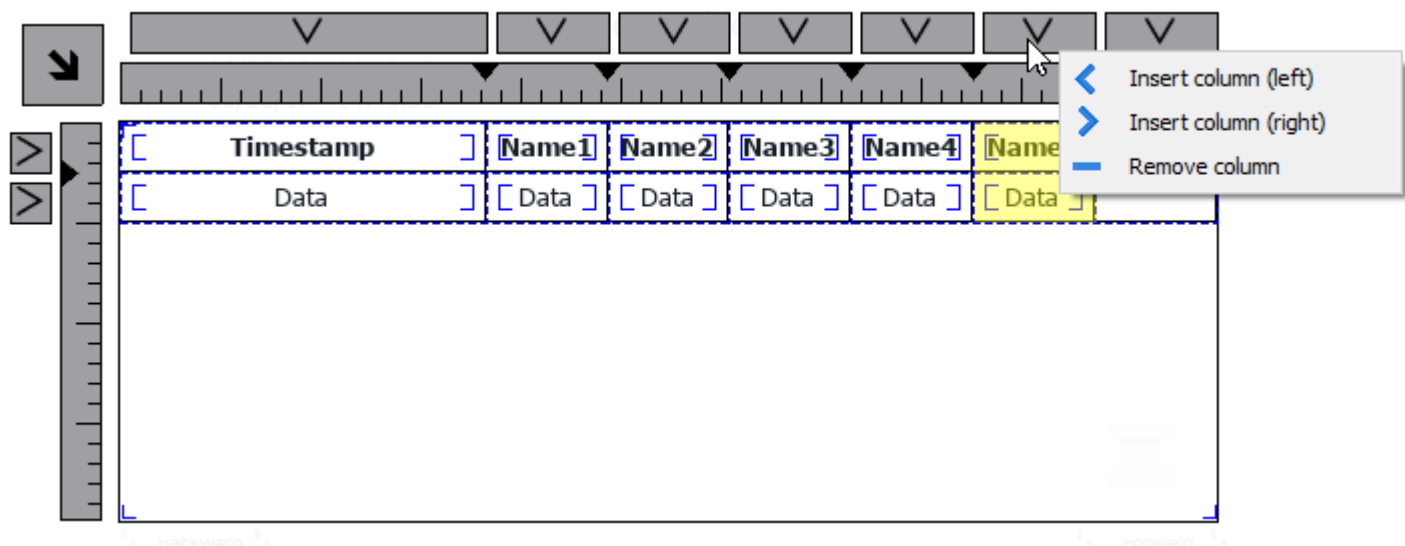
### Buttons:

- **REFRESH**  
Retrieve trend data from internal buffer and refresh table view
- **BACKWARD/FORWARD**  
Move the display window forward or backward as specified in the duration parameter

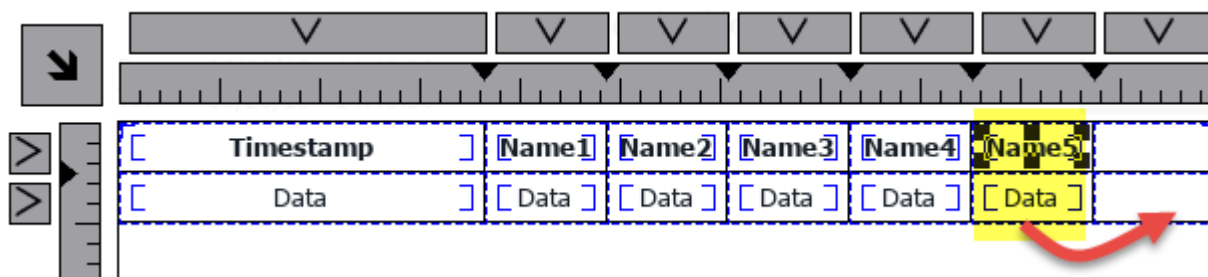
Parameter	Description
<b>TrendName</b>	Trend Buffer from which the samples are retrieved (see <a href="#">"Data logging" on page 252</a> )
<b>Heading</b>	Heading label
<b>Page Duration</b>	Initial value of time window to show
<b>Time Spec</b>	Time format: <ul style="list-style-type: none"> <li>• <b>Local</b> = show the time values of the HMI device.</li> <li>• <b>Global</b> = show the time values using UTC format.</li> </ul>
<b>Date Format</b>	Select the Date and Time format
<b>Table Layout</b>	Defines the characteristics of the scroll bar and allows to remove the header of the table

## Adding or removing trend columns

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



Copy and past fields from another column



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

The screenshot illustrates the configuration of a Table trend widget in JMobile Studio. The main canvas shows a table with columns labeled Name1 through Name6 and Data. A red circle with the number 1 highlights the Name6 column. The Properties panel on the right shows the widget's configuration, including Text, DataLink, Access, and Events. The DataLink is set to Name5:TrendTableNew.TrendSrcWgt. The Source panel at the bottom shows the widget's structure, with a red circle and the number 3 highlighting the Name6 property under TrendTableNew.TrendSrcWgt.

**Properties Panel:**

Text : TrendTableNew.TableWgt.label13	
<b>Text</b>	Name5
<b>DataLink</b>	Name5:TrendTableNew.TrendSrcWgt
<b>Access</b>	R
<b>Events</b>	

**Source Panel:**

- TrendTableNew.TableWgt.label13.text
  - Source: ☐ Tag ☐ Alias ☐ System ☒ Widget ☐ Recipe
  - Search
  - Name
    - \_AlarmsMgr
    - \_EventMgr
    - \_MultiLangMgr
    - Page1
    - TrendTableNew
      - General
        - Heading
        - Page Duration
      - Position
      - TrendTableNew.TrendSrcWgt
        - Name1
        - Name2
        - Name3
        - Name4
        - Name5
        - Name6**
        - TimeStamp

Figure 18-1: TrendTableNew widget configuration

The figure shows the configuration of the TrendTableNew widget in JMobile Studio. The main window displays the widget's visual representation, which includes a table with columns labeled Name1 through Name5 and a Data column. A red circle with the number 4 highlights the Data column.

The Properties panel on the right shows the configuration for the selected widget (Text : TrendTableNew.TableWgt.label14). The DataLink property is set to Name5:TrendTableNew.TrendSrcWgt, and the Access Type is set to R.

The TrendTableNew widget configuration dialog is shown below, with a red circle and the number 6 highlighting the Name6 property.

**TrendTableNew.TableWgt.label14.text**

Source: ☐ Tag ☐ Alias ☐ System ☒ Widget ☐ Recipe

Search

**Name**

- ▷ \_AlarmsMgr
- ▷ \_EventMgr
- ▷ \_MultiLangMgr
- ▷ Page1
- ▲ TrendTableNew
  - ▷ General
    - Heading
    - Page Duration
  - ▷ Position
    - ▲ TrendTableNew.TrendSrcWgt
      - Name1
      - Name2
      - Name3
      - Name4
      - Name5
      - Name6**
      - TimeStamp





# 19 Data transfer

---

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

---

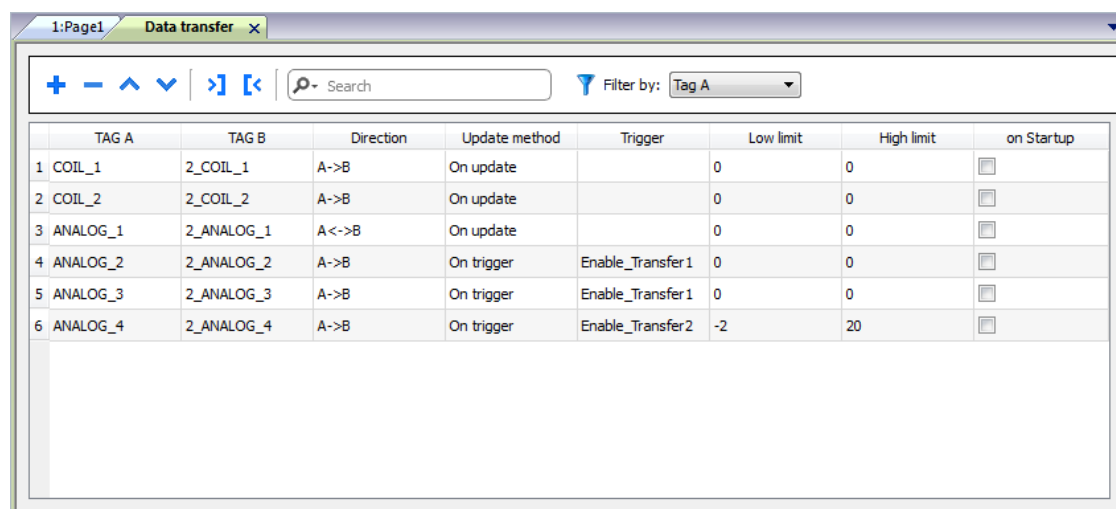
<b>Data transfer editor</b> .....	<b>270</b>
<b>Exporting data to .csv files</b> .....	<b>272</b>
<b>Data transfer limitations and suggestions</b> .....	<b>272</b>

# Data transfer editor

Path: **ProjectView** > **Config** > double-click **Data transfer**

Use the Data transfer editor to map transfer rules.

Each line in the Data transfer editor defines a mapping rule between two tags. Define more mapping rules if you need different direction, update method or trigger.



	TAG A	TAG B	Direction	Update method	Trigger	Low limit	High limit	on Startup
1	COIL_1	2_COIL_1	A->B	On update		0	0	<input type="checkbox"/>
2	COIL_2	2_COIL_2	A->B	On update		0	0	<input type="checkbox"/>
3	ANALOG_1	2_ANALOG_1	A<->B	On update		0	0	<input type="checkbox"/>
4	ANALOG_2	2_ANALOG_2	A->B	On trigger	Enable_Transfer1	0	0	<input type="checkbox"/>
5	ANALOG_3	2_ANALOG_3	A->B	On trigger	Enable_Transfer1	0	0	<input type="checkbox"/>
6	ANALOG_4	2_ANALOG_4	A->B	On trigger	Enable_Transfer2	-2	20	<input type="checkbox"/>







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

## Data transfer toolbar

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

## Data transfer parameters

Parameter	Description
<b>TAG A/ TAG B</b>	Pair of tags to be mapped for exchanging through the HMI device.
<b>Direction</b>	Transfer direction.  <b>A-&gt;B</b> and <b>B-&gt;A</b> : Unidirectional transfers, values are always copied from one tag and sent to the other tag in the specified direction.  <b>A&lt;-&gt;B</b> : Bidirectional transfer, values are transferred to and from both tags.
<b>Update Method</b>	<b>On trigger</b> : Data transfer occurs when the value of the tag set as trigger changes above or below the values set as boundaries. Limits are recalculated on the previous tag value, the same that triggered the update.

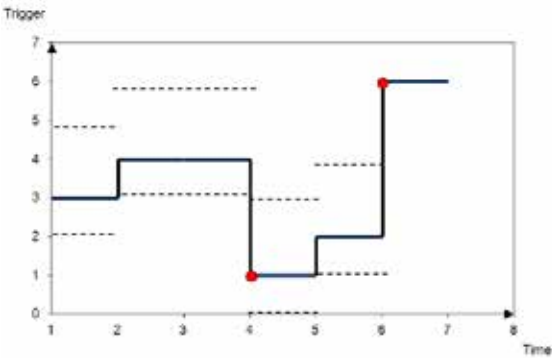
Parameter	Description
	<p> Note: This method applies only to unidirectional transfers (A-&gt;B or B-&gt;A).</p> <p><b>On Update:</b> Data transfer occurs whenever the value of the source tag changes.</p> <p> Note: This method applies both to unidirectional and to bidirectional transfers (A-&gt;B, B-&gt;A and A&lt;-&gt;B).</p> <p> Note: The Runtime cyclically monitors source tags changes (trigger tag when using On Trigger or tags to transfer when using On Update) based on Tag editor <b>Rate</b> parameter. If <b>Rate</b> setting for source Tag is 500 ms (default), the system checks for updates every 500 ms.</p> <p> Note: Changes on source tags faster than <b>Rate</b> may be not detected .</p>
<b>Trigger, High limit, Low limit</b>	<p>Tag that triggers the data transfer process. When this tag changes its value outside the boundaries set as <b>High limit</b> and <b>Low limit</b>, data transfer is started. The range of tolerance is recalculated according to the specified limits on the tag value which triggered the previous update. No action is taken if the change falls within the limits.</p> <p>This mechanism allows triggering data transfers only when significant variations of the reference values occur.</p> <p><b>Low limit</b> is less or equal to zero.</p> <p> Note: If both <b>Low limit</b> and <b>High limit</b> are set to "0", data transfer occurs whenever the value of the trigger tag changes.</p>
<b>on Startup</b>	<p>When selected, data transfer is forced:</p> <ul style="list-style-type: none"> <li>• on HMI startup if the quality of the source tag is good</li> <li>• after communication errors, when the associate device nodes return active</li> </ul> <p>See "<a href="#">Objects</a>" on <a href="#">page 463</a> for details on quality.</p> <p> <b>Important: Data transfers executed on startup may have major impact on the HMI device boot time. Enable this option only when necessary.</b></p>

### Example of limit setting

**High limit** = 1,9

**Low limit** = - 0,9

• = points where the data transfer is triggered





# Exporting data to .csv files

Configuration information for data transfers can be exported to a .csv file.

## Example of data transfer settings in .csv file

A	B	C	D	E	F	G	H	I	J
COIL_1	2_COIL_1	A->B	On update		0	0	data1	true	1
COIL_2	2_COIL_2	A->B	On update		0	0	data2	true	1
ANALOG_1	2_ANALOG_1	A<->B	On update		0	0	data3	true	1
ANALOG_2	2_ANALOG_2	A->B	On trigger	Enable_Transfer1	0	0	data4	true	1
ANALOG_3	2_ANALOG_3	B->A	On trigger	Enable_Transfer1	0	0	data5	true	1
ANALOG_4	2_ANALOG_4	A->B	On trigger	Enable_Transfer2	-10	20	data6	true	1

Column	Description
A to G	Same data as in the Data transfer editor
H	Unique identifier automatically associated to each line. <div> <b>Important: When you edit the .csv file and you add any extra line, make sure you enter a unique identifier in this column.</b></div>
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.



# 20 Offline node management

---

When one of the controllers communicating with the HMI device goes offline, communication performance of the system may eventually decrease.

The offline node management feature recognizes offline controllers and removes them from communication until they come back online.

Additionally, if you know that any of the controllers included in the installation is going to be offline for a certain time, you can manually disable it to maximize system performance.



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

---

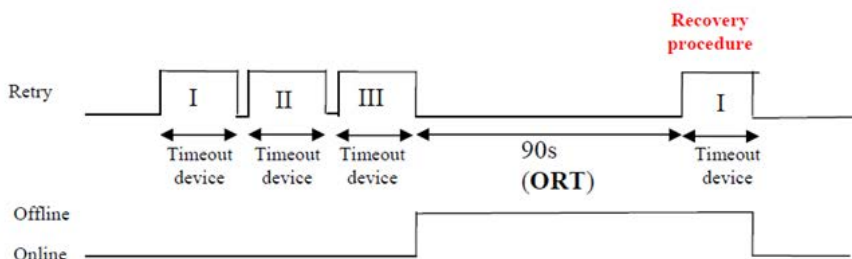
<b>Offline node management process .....</b>	<b>276</b>
<b>Manual offline node management process .....</b>	<b>276</b>
<b>Manual offline configuration .....</b>	<b>276</b>
<b>Automatic offline node detection .....</b>	<b>277</b>

# Offline node management process

Steps of the process are:

- The system communicates normally with a certain device. When the device is not responding to a communication request, the system will repeat the request twice before declaring the device offline.
- When a device is offline, the system sends communication requests to the device with a longer interval, called Offline Retry Timeout. If the device answers to one of these requests, the system declares it online and restarts normal communication.

The diagram shows the three communication attempts and the recovery procedure that starts when the Offline Retry Timeout is elapsed.



## Manual offline node management process

Offline node management can be done manually. When a specific device is online and it is communicating normally you can:

- use an action to declare the device offline: the system stops communication with the device.
- use an action to declare the device online: the system restarts normal communication with the device.

## Manual offline configuration

When you know that some devices in communication with the HMI device are going to remain offline for a certain period of time, you can exclude them from communication using the **EnableNode** action.



**WARNING:** All disabled device nodes will remain disabled if the same project is downloaded on the device, on the other hand, if a different project is downloaded, all disabled devices will be re-enabled. The same happens with a package update.



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



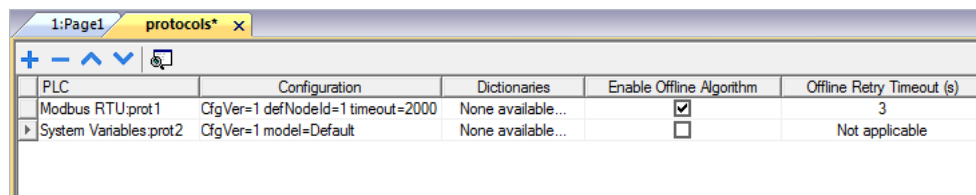
**Note:** When using the action **EnableNode** to force a device node back online, communication will start immediately.



## Automatic offline node detection

When a device is not answering to communication requests, it is de-activated. The HMI device stops sending requests to this device. After three seconds, the HMI device sends a single command to check if device is available, if so the communication is restarted, otherwise it is disabled for another timeout interval.

Default settings can be modified in Protocol editor.



PLC	Configuration	Dictionaries	Enable Offline Algorithm	Offline Retry Timeout (s)
Modbus RTU:prot1	CfgVer=1 defNodeId=1 timeout=2000	None available...	<input checked="" type="checkbox"/>	3
System Variables:prot2	CfgVer=1 model=Default	None available...	<input type="checkbox"/>	Not applicable



Note: Not all protocols support this feature.

Parameter	Description
<b>Enable Offline Algorithm</b>	Enables offline management for the protocol
<b>Offline Retry Timeout</b>	Interval in seconds for the retry cycle after a device has been deactivated. Range: 1–86.400 seconds (24h).



# 21 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.

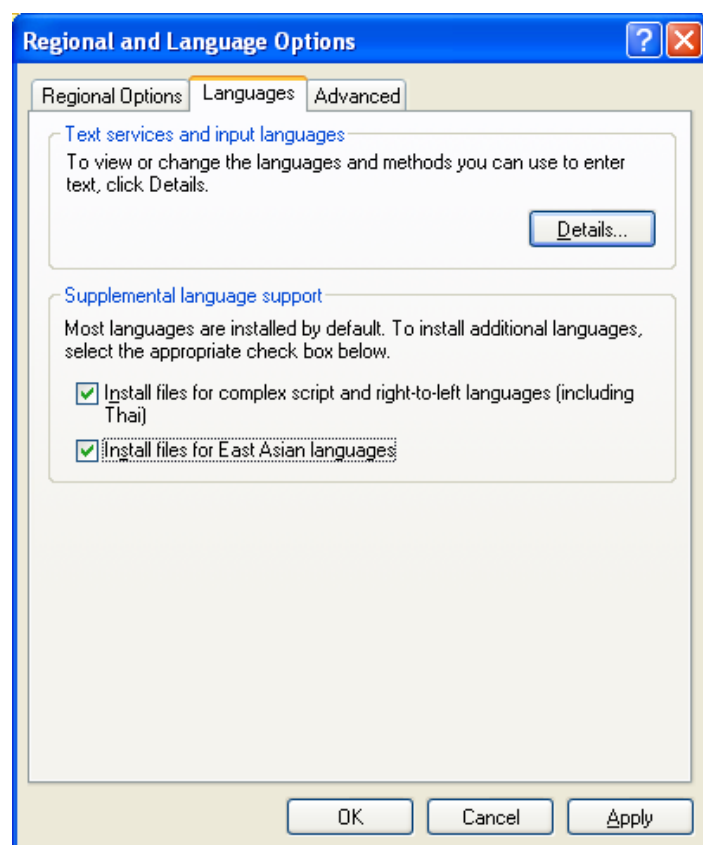
Multi-language feature can be used to define languages and character sets in a project. JMobile Studio also extends the TrueType Fonts provided by Windows systems to provide different font faces associated with different character sets.

JMobile Studio also allows you to provide strings for each of the languages supported.

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



**Important: In Windows XP operating systems you have to install the support for complex script and East Asian languages.**



## Supported fonts for Simplified Chinese

For Simplified Chinese, the following fonts are supported:

Font name	Font file
Fangsong	simfang.ttf
Arial Unicode MS	ARIALUNI.TTF
Kaiti	simkai.ttf
Microsoft Yahei	msyh.ttf
NSImSun	simsun.ttc
SimHei	simhei.ttf
Simsun	simsun.ttc

## Supported fonts for Traditional Chinese

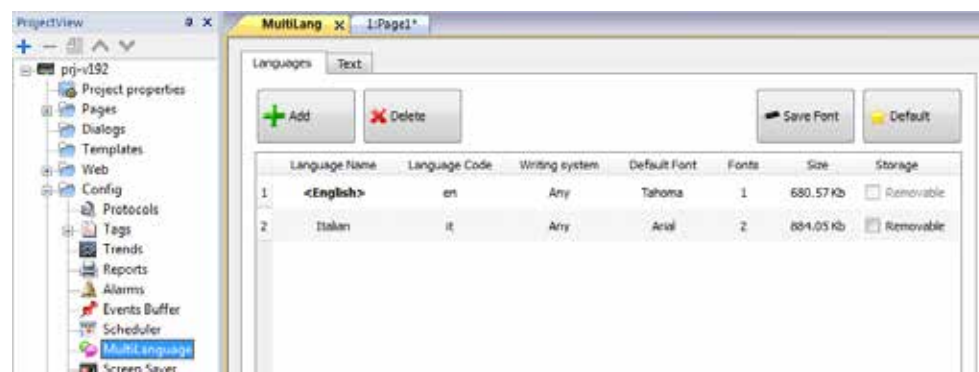
For Traditional Chinese, the following fonts are supported:

Font name	Font file
DFKai-SB	kaiu.ttf
Microsoft Sheng Hai	msjh.ttf
Arial Unicode MS	ARIALUNI.TTF
MingLiU	mingliu.ttc
PMingLiU	mingliu.ttc
MingLiU_HKSCS	mingliu.ttc



The Multi-language editor .....	281
Changing language .....	282
Multi-language widgets .....	282
Exporting/importing multi-language strings .....	284
Changing language at run time .....	286
Limitations in Unicode support .....	286

# The Multi-language editor

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



## Language settings

Parameter	Description
<b>Language Name</b>	Name identifying the language in the project.
<b>Language Code</b>	ISO 639 language code identifier, used to match language items when importing resources from external xml files.
<b>Writing system</b>	Select the set of fonts to be used with the language
<b>Default Font</b>	Default font for project's widgets.  Note: When you choose a new font you are prompted to replace the font used in the widgets you already created.
<b>Fonts</b>	Number of fonts associated with the selected language.
<b>Size</b>	Memory used to store font files.
<b>Storage</b>	Location of file fonts is a removable external memory.  Tip: Store large font files on removable memory to free memory requirements in the HMI device.

## Adding a language

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

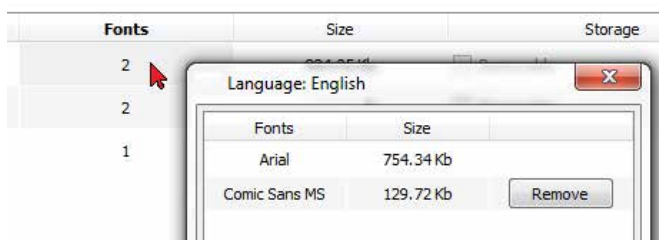


**Important:** Font files configured to be stored on removable memory must be provided to the final user to complete font installation on the HMI device.

## Removing fonts

To remove fonts no longer needed:

1. Click on the font number in the Multi-language editor: a dialog with the list of the used fonts is displayed.

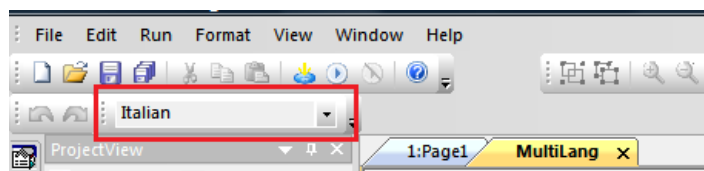


2. Select the fonts to be removed and click **Remove**: removed fonts are replaced with the default font.

## Changing language

### Changing language during page design

A combo box is available for changing language during page design. If no texts appears, please check **Text** tab in the Multilanguage editor and insert missing string.



## Multi-language widgets

Multi-language support is available for objects such as buttons, static text, messages, alarm descriptions and pop-up messages.

### Multi-language for label widgets

Double-click on a text widget in a page to open the **Text** dialog.



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

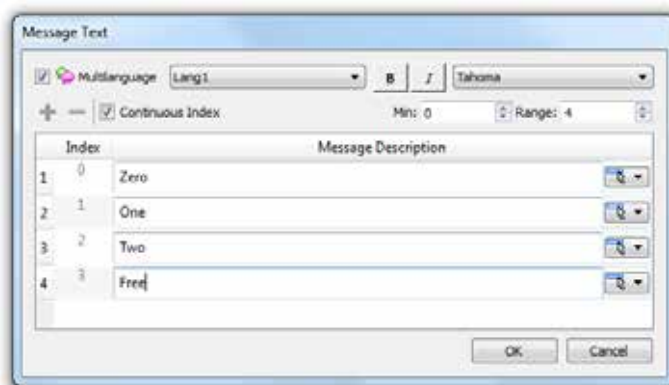


Note: Bold, italic and color properties set here for the widget are applied to all languages .

Parameter	Description
<b>Multilanguage</b>	Enable/disable multi-language function for the widget.
<b>Choose text from other widget</b>	Click on button to browse existing message strings in project to pick text for the widget.

## Multi-language for message widgets

Double-click on a message widget in a page to open the **Message Text** dialog.

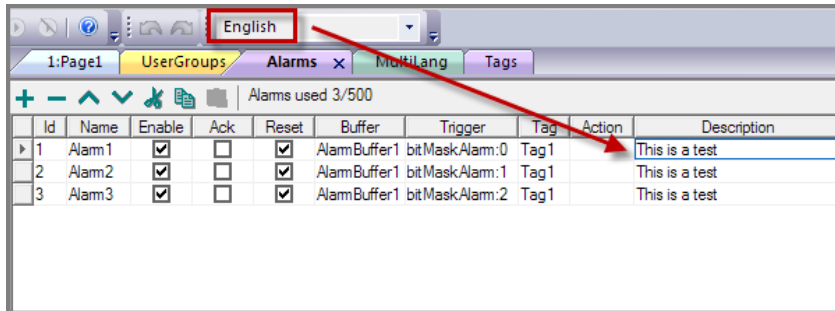


Parameter	Description
<b>Multilanguage</b>	Enable/disable multi-language function for the widget.
<b>Continuous Index</b>	Index for the widget is set of contiguous numbers (example 3, 4,5,6)
<b>Min</b>	Starting number for index
<b>Range</b>	Number of messages
<b>Choose text from other widget</b>	Click on button to browse existing message strings in project to pick text for the widget.

## Multi-language for alarm messages

To add a multi-language strings for alarm messages:

1. Open the Alarm editor.
2. Select a language using the language combo box.
3. Enter the text for the alarm in the **Description** column.

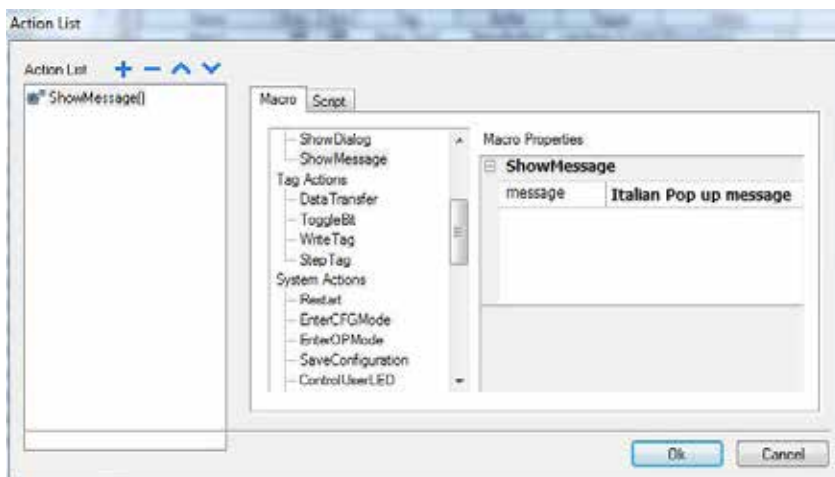


Tip: Text labels with alarm states displayed by alarms widgets can be translated or personalized through the Multilanguage text editor.

## Multi-Language for pop-up messages

To add a multi-language pop-up message:

1. Select a language from the language combo box.
2. Add the Page action **ShowMessage** and enter the text in the selected language.



## Exporting/importing multi-language strings

The easiest way to translate a project into multiple languages is to export all texts to a .csv file, translate the resulting document and then import the translated text back into the project.



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

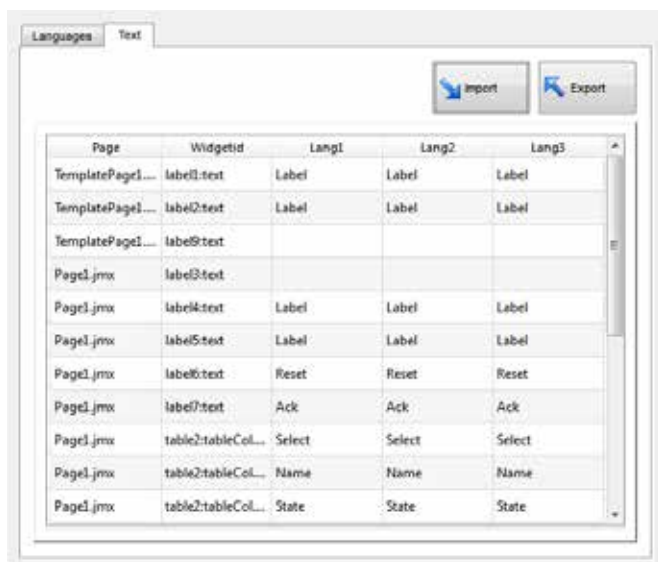


## Exporting and reimporting strings

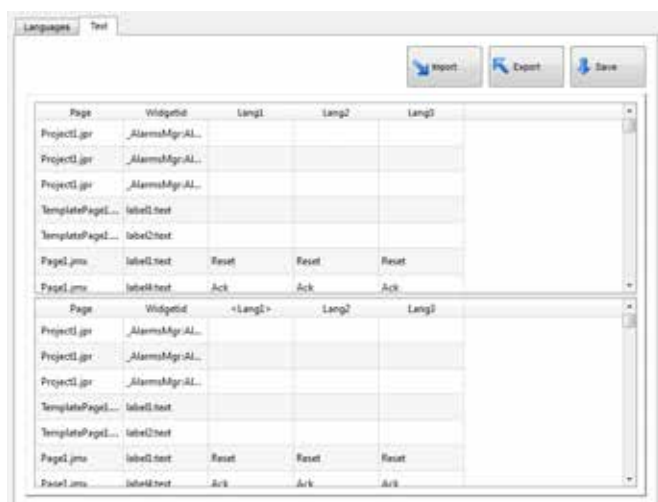
Path: **ProjectView** > **Config** > double-click **MultiLanguage**

To export and re-import multi-language strings:

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



**Important:** Set all languages that will be used in the project before exporting the file. This will guarantee that the exported file will contain all columns and language definitions.



2. Once the strings have been translated, click **Import** to re-import them into the project: strings are imported matching the widget ID and the page number of each widget.
3. Click **Save** to save the new widget data.



**Note:** To change the separator used in the exported file, change the regional settings of your computer. When importing, the separator information is retrieved from the file; if not found, the default character "," is used.

## Import constraints

The following formats are supported for import:

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



Note: Use the Unicode Text file format when you import a file modified using Microsoft® Excel®.

## Changing language at run time

### 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 164](#).



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 ["curlLangCode" on page 482](#) for additional details.

### Missing fonts

When you change language, if the required fonts are not available in the device memory, a pop-up message prompts you to insert the memory card containing the missing fonts. At the end of the operation you can remove the memory card.



## Limitations in Unicode support

JMobile Studio 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
<b>Protocol editor</b>	Alias	ASCII [32..126]	(space), , ; . < * > '
<b>Tag editor</b>	Name	ASCII [32..126]	. \ / * ? : > <   " & # % ; =
	Group	ASCII [32..126]	<New> \ / * ? : > <   " & # % ;
	Comment	Unicode	
<b>Trends</b>	Name	ASCII [32..126]	\ / * ? : > <   " & # % ;
<b>Printing Reports</b>	Name	ASCII [32..126]	\ / * ? : > <   " & # % ;
<b>Alarms</b>	Name	ASCII [36..126]	\ / * ? : > <   " & # % ;
	Description	Unicode	[] - for live tags, \ escape seq for [ and \
<b>Events</b>	Buffer Name	ASCII [32..126]	\ / * ? : > <   " & # % ;
<b>Scheduler</b>	Name	ASCII [32..126]	\ / * ? : > <   " & # % ;
<b>Languages</b>	Language Name	ASCII [32..126]	\ / * ? : > <   " & # % ;
	Texts in widgets	Unicode	-
	Texts from import files	Unicode	-
<b>User Group</b>	Group Name	a-z A-Z _	admin,guest,unauthorized
	Comments	Unicode	-
<b>User</b>	Name	ASCII [32..126]	\ / * ? : > <   " & # % ;
	Password	Unicode	-
	Comment	Unicode	-
<b>Recipes</b>	Name	ASCII [32..126]	\ / * ? : . > <   " & % ; ,
	Set Name	ASCII [32..126]	\ / * ? : . > <   " & % ; ,
	Element name	ASCII [32..126]	\ / * ? : . > <   " & % ; ,
<b>General</b>	Project Name	A-Z,a-z,0-9,-, _	"PUBLIC", "readme", "index.html"
	Page Name	A-Z,a-z,0-9,-, _	-
	Dialog Page Name	A-Z,a-z,0-9,-, _	-
	Template Page Name	A-Z,a-z,0-9,-, _	-

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	-

## 22 Scheduler

---

JMobile Studio 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 run-time user interface that allows the end-user to change settings for each schedule. You do this adding a **Scheduler** widget to a page of your project and configuring it to fit user scheduling needs.

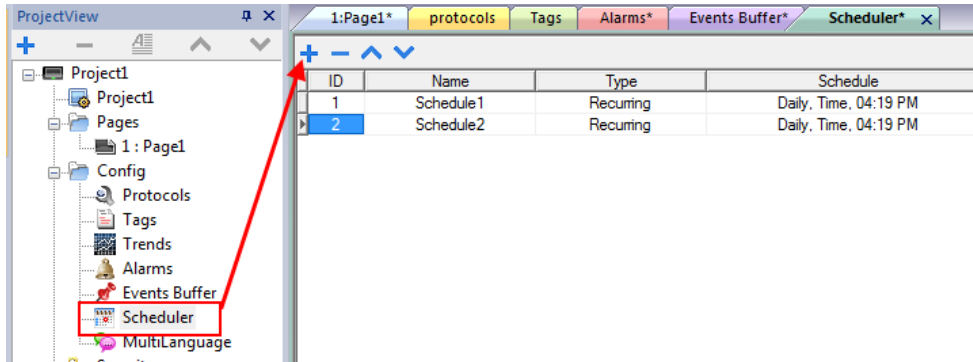
---

<b>Creating a schedule</b> .....	<b>290</b>
<b>HighResolution schedule</b> .....	<b>290</b>
<b>Recurring schedule</b> .....	<b>290</b>
<b>Configuring location for schedules</b> .....	<b>292</b>
<b>Configuring the Scheduler widget</b> .....	<b>293</b>
<b>Scheduling events at run time</b> .....	<b>294</b>

# Creating a schedule

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

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



## Schedule parameters

Parameter	Description
<b>ID</b>	Unique code assigned automatically to the schedule
<b>Name</b>	Name of schedule
<b>Type</b>	Type of schedule: <ul style="list-style-type: none"> <li><b>Recurring</b>, see "<a href="#">Recurring schedule</a>" below for details.</li> <li><b>HighResolution</b>, see "<a href="#">HighResolution schedule</a>" below for details</li> </ul>
<b>Schedule</b>	Scheduler settings and options. See " <a href="#">Recurring schedule</a> " below for details.
<b>Action</b>	Actions to be executed at the scheduled time
<b>Priority</b>	Priority level for the event. If two schedules occur at the same time, the event with the higher priority will be executed first.

## HighResolution schedule

The **HighResolution** schedule is used to perform actions that need to be repeated at specified intervals. The interval between executions is set in milliseconds in the **Schedule** column.





Note: You cannot change at run time the settings of this type of schedule. If you need to change the action time settings at run time, 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 run time.

## Recurring scheduler parameters

Parameter	Description
<b>Type</b>	Frequency of the scheduled actions
<b>Mode</b>	Specific settings required by each scheduler type
<b>Condition</b>	<p>Boolean tag (true/false) to activate the specified actions at the moment the timer is triggered. Actions will be executed if tag = true. By default, actions are executed when the timer is triggered.</p> <p> Note: Only tags attached to the Boolean data type are shown.</p>
<b>Actions</b>	<p>Actions to be executed by the schedule.</p> <p> <b>Important: Actions and schedule parameters cannot be modified at run time</b></p>
<b>Date</b>	Date when the scheduled actions will be executed
<b>Time/Offset</b>	<p>This field display one of the following:</p> <p><b>Time</b> = when the scheduled actions will be executed</p> <p><b>Offset</b>= delay or advance with respect to the selected mode.</p>
<b>Location</b>	Reference location to calculate sunset/sunrise time.
<b>weekdays</b>	Days of the week in which the scheduled actions will be executed.
<b>On startup</b>	Executes schedule at start up
<b>Enable schedule</b>	Enables/disables the schedule
<b>Execute only at startup</b>	Executes the schedule only once at start up

## Schedule type options

Option	Description
<b>By Date</b>	Actions are executed on the specified date and time.
<b>Daily</b>	Actions are executed daily at the specified time.
<b>Every</b>	Actions are executed with the specified interval (Range: 1 s–1 day)
<b>Hourly</b>	Actions are executed every hour at the specified minute.
<b>Monthly</b>	Actions are executed every month at the specified date and time.

Option	Description
<b>Weekly</b>	Actions are executed every week on the specified weekday(s) and time.
<b>Yearly</b>	Actions are executed every year at the specified date and time.

## Schedule mode options

Option	Description
<b>Time</b>	Depends on the schedule type. Allows you to specify date/time/week data.
<b>Random10</b>	Actions are executed in the time interval of 10 minutes before or after the set time. For example, if set time is 10:30, actions are executed any time between 10:20 and 10:40.
<b>Random20</b>	Actions are executed in the time interval of 20 minutes before or after the set time. For example, if set time is 10:30, actions are executed any time between 10:10 and 10:50.
<b>Sunrise+</b>	Actions are executed with a specified delay after sunrise. The delay is set in minutes/hours and sunrise time is location specific.
<b>Sunrise-</b>	Actions are executed with a specified advance before sunrise. The advance is set in minutes/hours and sunrise time is location specific.
<b>Sunset+</b>	Actions are executed with a specified delay after sunset. The delay is set in minutes/hours and sunset time is location specific.
<b>Sunset-</b>	Actions are executed with a specified advance before sunset. The advance is set in minutes/hours and sunset time is location specific.

See "[Configuring location for schedules](#)" 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 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.



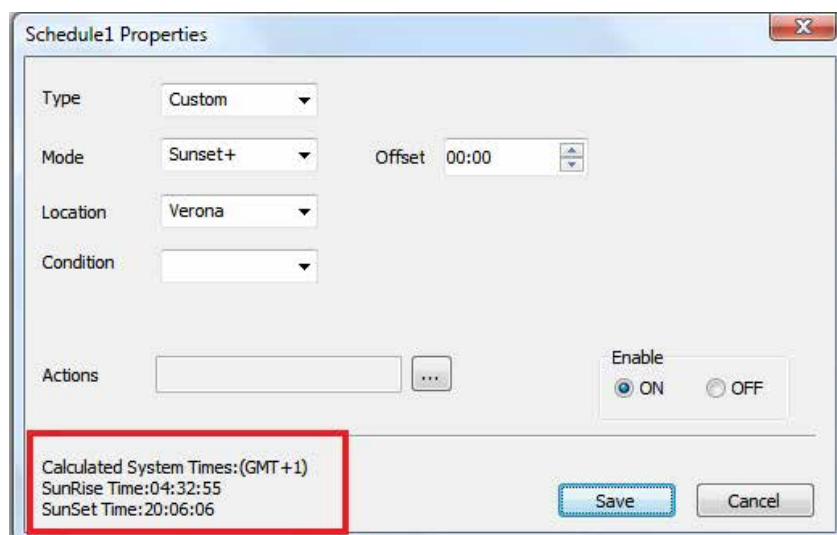
## Location files position

Application	Location file path
JMobile Studio	<i>Exor\JMobile Suite\languages\shared\studio\config\Target_Location.xml</i>
HMI Devices	<i>Exor\JMobile Suite\runtime\&lt;HW Platform&gt;\config\Target_Location.xml</i>
Simulator	<i>Exor\JMobile Suite\simulator\config\Target_Location.xml</i>
JMobile PC Runtime	<i>Exor\JMobile Suite\server\config\Target_Location.xml</i>

For example, the information for the city of Verona (IT) is shown below:

```
<file city="Verona" latitude="45.44" longitude="10.99" utc="1"/>
```

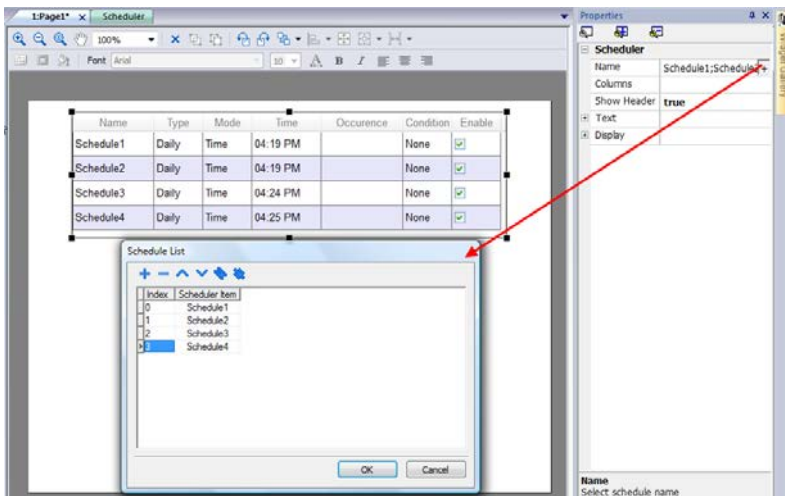
Location information is also displayed in the dialog together with sunset and sunrise times.



## Configuring the Scheduler widget

To display scheduler data on a page:

1. Drag and drop a **Scheduler** widget from the widget gallery into the page.
2. In the **Properties** pane, click + for the **Name** parameter: the **Schedule List** dialog is displayed.
3. Add all the schedules you want to display in the page.



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

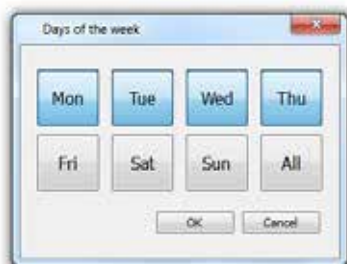
## Scheduler settings

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

## Scheduling events at run time

At run time you can modify the following scheduling parameters.

Name	Type	Mode	Time	Occurrence	Condition	Enable
Schedule1	By Date	Time	11:01	JUN 20, 2013	None	<input checked="" type="checkbox"/>
Schedule3	Monthly	Sunrise+	11:01	Day : 3	None	<input checked="" type="checkbox"/>
Schedule4	Weekly	Rando...	16:19	MTWTFSS	None	<input checked="" type="checkbox"/>
Schedule5	Yearly	Time	01:00			
Schedule6	Custom	Time	01:16			



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

See ["Recurring schedule"](#) on page 290 for details on schedule parameters.



## 23 21 CFR Part 11 Compliance

JMobile Studio 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 JMobile Studio 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	JMobile Studio 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.	<p>Reports generated by JMobile Studio can be signed using x.509 Certificates. A certificate that includes the public key, necessary to verify the signature of reports, will be exported with the report.</p> <p>References:</p> <ul style="list-style-type: none"> <li>• <a href="#">"SaveEventArchive" on page 194</a></li> <li>• <a href="#">"Scheduler" on page 289</a></li> </ul>
11.10(b)	The ability to generate accurate and complete copies of records in both human readable and electronic form suitable for inspection, review, and copying by the agency. Persons should contact the agency if there are any questions regarding the ability of the agency to perform such review and copying of the electronic records.	<p>Application developer can select the resources (process values, alarms, etc.) whose changes will be tracked to the audit trail. Each change of the selected resources will be recorded with the name of the operator doing the change. The audit trail reports can be exported to .csv files.</p> <p>References:</p> <ul style="list-style-type: none"> <li>• <a href="#">"Enable/disable audit trail" on page 322</a></li> <li>• <a href="#">"Exporting audit trail as .csv files" on page 327</a></li> <li>• <a href="#">"SaveEventArchive" on page 194</a></li> </ul>
11.10(c)	Protection of records to enable their accurate and ready retrieval throughout the records retention period.	<p>Applications can be developed to self-generate signed reports to external memory or network folders at predefined interval (e.g. at the end of the day) or when circular buffer is full. User is responsible to keep these reports saved for the retention period.</p> <p>References:</p> <ul style="list-style-type: none"> <li>• <a href="#">"SaveEventArchive" on page 194</a></li> <li>• <a href="#">"Scheduler" on page 289</a></li> </ul>
11.10(d)	Limiting system access to authorized individuals.	Application developer is responsible for the

Chapter	Description	JMobile Studio compliance level (v2.8)
		<p>appropriate security configuration of the application.</p> <p>References:</p> <ul style="list-style-type: none"> <li>• <a href="#">"User management and passwords" on page 309</a></li> </ul>
11.10(e)	Use of secure, computer-generated, time-stamped audit trails to independently record the date and time of operator entries and actions that create, modify, or delete electronic records. Record changes shall not obscure previously recorded information. Such audit trail documentation shall be retained for a period at least as long as that required for the subject electronic records and shall be available for agency review and copying.	<p>Audit trail records are stored using a circular buffer (this is to ensure that the device will not run out of memory). Audit trails cannot be modified by the operator. Each record contains a sequential number to easily check the presence of all records. The application can be developed to save/export a copy of the data at regular intervals (e.g. at the end of each day); operator is responsible for storing copy of reports in a safe place.</p> <p>References:</p> <ul style="list-style-type: none"> <li>• <a href="#">"SaveEventArchive" on page 194</a></li> <li>• <a href="#">"Scheduler" on page 289</a></li> <li>• <a href="#">"Exporting audit trail as .csv files" on page 327</a></li> </ul>
11.10(f)	Use of operational system checks to enforce permitted sequencing of steps and events, as appropriate.	<p>Macros or JavaScript can be used to configure command sequences in the application.</p>
11.10(g)	Use of authority checks to ensure that only authorized individuals can use the system, electronically sign a record, access the operation or computer system input or output device, alter a record, or perform the operation at hand.	<p>The HMI application can be configured</p> <ul style="list-style-type: none"> <li>• to be accessible only after user sign in with its own password</li> <li>• objects can be configured to be available or not available depending on the user who logged in to the system</li> <li>• resources can be configured to require a password confirmation before be modified</li> </ul> <p>References:</p> <ul style="list-style-type: none"> <li>• <a href="#">"User management and passwords" on page 309</a></li> <li>• <a href="#">"Electronic Signature" on page 323</a></li> </ul>
11.10(h)	Use of device (e.g., terminal) checks to determine, as appropriate, the validity of the source of data input or operational instruction.	<p>Resources can be configured to be accessible only from selected user groups. List of allowed IP address can be configured from the User Management settings.</p> <p>References:</p> <ul style="list-style-type: none"> <li>• <a href="#">"Modifying access permissions" on page 311</a></li> </ul>

Chapter	Description	JMobile Studio compliance level (v2.8)
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.	JMobile Studio 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, responsibility, or authorship) associated with the signature.	All records will be added to the audit trail with time stamp and user id of logged user.  References: <ul style="list-style-type: none"> <li>• <a href="#">"Exporting audit trail as .csv files" on page 327</a></li> <li>• <a href="#">"Table audit widget" on page 326</a></li> </ul>
11.50(b)	The items identified in paragraphs (a)(1), (a)(2), and (a)(3) of this section shall be subject to the same controls as for electronic records and shall be	

Chapter	Description	JMobile Studio compliance level (v2.8)
	included as part of any human readable form of the electronic record (such as electronic display or printout).	
11.70	Electronic signatures and handwritten signatures executed to electronic records shall be linked to their respective electronic records to ensure that the signatures cannot be excised, copied, or otherwise transferred to falsify an electronic record by ordinary means.	Application developer is responsible for avoiding using the macros that permit the import/export of user passwords.
11.100(a)	Each electronic signature shall be unique to one individual and shall not be reused by, or reassigned to, anyone else.	System will ensure that two users with the same id cannot be defined. It is user responsibility to avoid removal and reassignment of the same user id to a different user.
11.100(b)	Before an organization establishes, assigns, certifies, or otherwise sanctions an individual's electronic signature, or any element of such electronic signature, the organization shall verify the identity of the individual.	User responsibility.
11.100(c)	<p>Persons using electronic signatures shall, prior to or at the time of such use, certify to the agency that the electronic signatures in their system, used on or after August 20, 1997, are intended to be the legally binding equivalent of traditional handwritten signatures.</p> <p>(1) The certification shall be submitted in paper form and signed with a traditional handwritten signature, to the Office of Regional Operations (HFC-100), 5600 Fishers Lane, Rockville, MD 20857.</p> <p>(2) Persons using electronic signatures shall, upon agency request, provide additional certification or testimony that a specific electronic signature is the legally binding equivalent of the signer's handwritten signature.</p>	User responsibility.
11.200(a)	<p>(a) Electronic signatures that are not based upon biometrics shall:</p> <p>(1) Employ at least two distinct identification components such as an identification code and password.</p>	JMobile Studio Security functions are based on the combination Username/ Password.
	(i) When an individual executes a series of signings during a single, continuous period of controlled system access, the first signing shall be executed using all electronic signature components; subsequent signings shall be executed using at least	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.



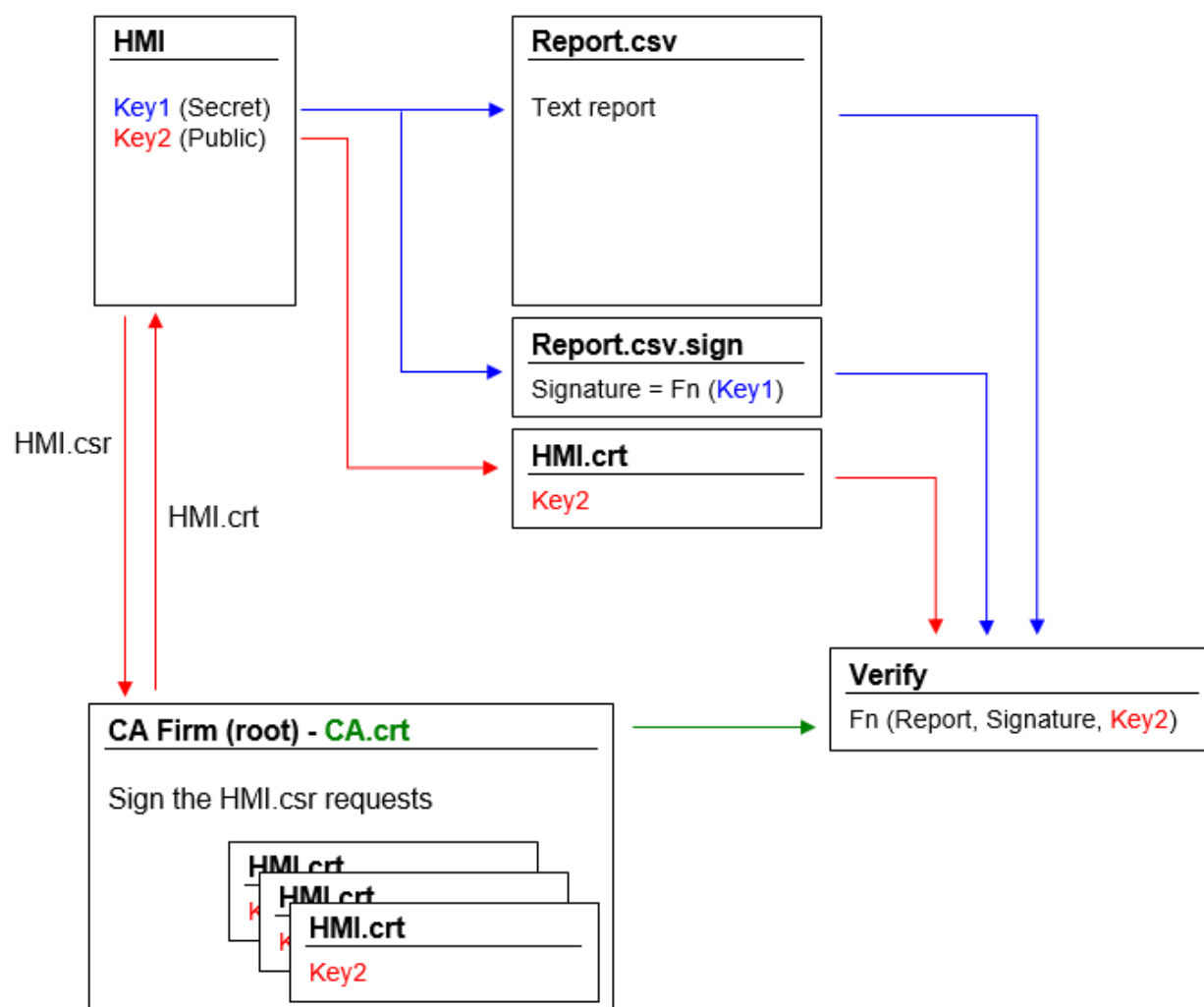
Chapter	Description	JMobile Studio compliance level (v2.8)
	<p>one electronic signature component that is only executable by, and designed to be used only by, the individual.</p> <p>(ii) When an individual executes one or more signings not performed during a single, continuous period of controlled system access, each signing shall be executed using all of the electronic signature components.</p>	<p>References:</p> <ul style="list-style-type: none"> <li>• <a href="#">"User management and passwords" on page 309</a></li> <li>• <a href="#">"Electronic Signature" on page 323</a></li> </ul>
	<p>(2) Be used only by their genuine owners; and</p> <p>(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.</p>	<p>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.</p> <p>References:</p> <ul style="list-style-type: none"> <li>• <a href="#">"Configuring users" on page 317</a></li> </ul>
11.200(b)	Electronic signatures based upon biometrics shall be designed to ensure that they cannot be used by anyone other than their genuine owners.	JMobile Studio 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).	<p>System can be configured to force each users to define a new and different password after a configurable number of days</p> <p>References:</p> <ul style="list-style-type: none"> <li>• <a href="#">"Configuring users" on page 317</a></li> </ul>
11.300(c)	Following loss management procedures to electronically deauthorize lost, stolen, missing, or otherwise potentially compromised tokens, cards, and other devices that bear or generate identification code or password information, and to issue temporary or permanent replacements using suitable, rigorous controls.	<p>Users can change their password at any time. Administration can redefine each user's password and force them to redefine at the first login.</p> <p>References:</p> <ul style="list-style-type: none"> <li>• <a href="#">"User management actions" on page 202</a></li> <li>• <a href="#">"Configuring users" on page 317</a></li> </ul>

Chapter	Description	JMobile Studio compliance level (v2.8)
11.300(d)	Use of transaction safeguards to prevent unauthorized use of passwords and/or identification codes, and to detect and report in an immediate and urgent manner any attempts at their unauthorized use to the system security unit, and, as appropriate, to organizational management.	Failed logging attempts are logged to audit trail.
11.300(e)	Initial and periodic testing of devices, such as tokens or cards, that bear or generate identification code or password information to ensure that they function properly and have not been altered in an unauthorized manner.	User is responsible for ensuring appropriate measures.

## x.509 Certificate

To ensure authenticity of reports generated by HMI devices, JMobile HMI Runtime can generate reports with signed files to verify the authenticity and the integrity of the generated reports.

JMobile HMI Runtime uses asymmetric cryptography keys to sign files and x.509 standard to manage public key certificates. The picture shows the architecture.



## Workflow

- 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.
- The macro `SaveEventArchive` can be used to generate signed reports (see "[SaveEventArchive](#)" on page 194 for additional details)
- Users can use the public key and the signed file to verify a report is authentic and has not been modified after it has been generated.

### Authenticity of the public key

Before verifying a report (Report.csv) with its signed file (Report.csv.sign), users must be sure of the authenticity of the public key (HMI.crt).

The public key can be signed by a Certificate Authority (CA) that guarantees its authenticity. As an alternative, the public key can be taken directly from the HMI device

## The internal x.509 certificate files

You can generate a self-signed certificate with a dedicated function in System Settings. You can also export or import the internal certificate file.

Users can personalize the information contained in the certificate. They can keep the self-signed certificate or retrieve the "Certificate Signed Request" to submit to a Certificate Authority for the validation. Then the certificate signed by the Certificate Authority can be imported inside the HMI device. (See the x.509 Certificate section inside the ["System Settings" on page 499](#))

You can import certificates with or without private key. When the certificate contains the private key, the internal private key of the HMI device will be substituted with the key found in the certificate. Certificates are encoded base64..



You can import inside each HMI device the same certificate file to have a unique public certificate file for all your HMI devices.

Example of a certificate with both public and private keys.

```

ssl-certificate.crt
1  -----BEGIN CERTIFICATE-----
2  MIIDBDCCAewCCQDcBYW7PYwJsDANBgkqhkiG9w0BAQsFADBEMQswCQYDVQQGEwJJ
3  VDEPMA0GA1UEBwwGVmVyb25hMRMwEQYDVQQKDApUZXRN0T2ZmaWNlMQ8wDQYDVQQD
4  DAZITUktMDQwHhcNMTCwNjI2MDgwOTQ1WhcNMTCwNjI2MDgwOTQ1WjBEMQswCQYD
5  VQGEwJVVDEPMA0GA1UEBwwGVmVyb25hMRMwEQYDVQQKDApUZXRN0T2ZmaWNlMQ8w
6  DQYDVQQDDAZITUktMDQwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCd
7  Nlp2kswcbLh4IxS6eeCgQ4EAUHCrpaz25YPfQ/un9/s0tejaa3Si3Pcqv3JqddJM8
8  mJE2aPF/+HhAEhtC+rv57TbgullUQJdoQpfoGChofpULforXZt2BfdWNx67plNoa
9  YM3ElaNtAKIW2o6S9HGEvlkf09XFLGkFgeMgC59+SejggucNT0m99m6fNa5910I7
10 UDJFinkC3bxtONj+WiL/iEZYkHXacaN9q06fx+2NfmiSsXGfPmSys5mocqo89tMa
11 TjyeF7jYpDccCpJ9pY4xRjRpicDKM7PabVoG/ascSMUUEXPE2R0W4UJ6bPaygD6
12 QLKCCq0BUi6/eUj0pnanaMBAAEWdQYJKoZIhvcNAQELBQADggEBAMLfIEXQOEjS
13 OpwVxzNxXmL/A6PLU5BK1hVYHb7ofb2Z37zn69vCn8ESglAFYK7QhkhJu3zAD+jH
14 fYBVKVdxfd3HS8EmcDWxpC6F2lfgqsSqepMRTbKbsSa053a7JsXtwnHVNfG6EBZV
15 8tqS1Gc4RwtJeVZJelUdmWSBD4Fc7asFeBCKqLrHJinz7buj3I4fLcyscTaMTBI9
16 fsE7poEpWvKc7NWtKY2glGG3AG6xONu3sEahcJ5k+UVdh/QQdAiCt3vG+JJ/owYU
17 sd30WI24pNzG/GUH9MbJyvI4ftA8IvEhGxHvi3xt7s1JnvYQDaghoEDhdtGvilOr
18 nJZ2FZOBCEI=
19  -----END CERTIFICATE-----
20  -----BEGIN RSA PRIVATE KEY-----
21  MIIEpAIBAAKCAQEA3D2adpLMHGy4eCMUunngoEOBAFBwkaWmeWD30P7p/f7NLXo2
22  mt0otz3Kr9yanXSTPjIRGwJxf/h4QBIbQvq7+e024LtZVECXaEKX6BgoaH6VC36K
23  l2bdgX3Vjceud6TaGmDNxJWjbQCifTqOkvRxxhL5ZH9PVxSxpBYHjIAuffkno4ILg
24  jU9Jvf2unzWuf2dC0lAyRSDZAt28bTjY/loi/4hGwJB12nGjfatOn8ftjX5okrFx
25  j55ksrOZqHKqPPbTgk48nhe42KQ3HAqSfaWOMUY0aXCJAjOz2mlaBv2rHEjFFOl
26  zxNkdFuFCemzwMoA+kCyggqtAVIuv3lI9KZ2pwIDAQABAoIBAGnamsuqrwDu5hGh
27  02H8GhUPvd/3ytIISujHyvgkwTf+FoII32y9uMe0pUy5/3y2v9v9/qm3P3djafJq
28  gb5Fprxx4dJFXJZaYi2U7U585lesmVqoHneCk/GeGlyH4zWlwo2xgNgBkkhgaIoR
29  zz0m0bachVz+SCD6wXUJpbMOW0FBw54oPL0XS/gD+76S9ET7xmQZAS5xV/w8Khht
30  PtjPftS8GKhqVIC9cMrrBrkuGQPrNrDaJMPsQDxrFp7POQm4+GivrUJ0FA9Vtx46
31  C5QhXqVps/BOD03mjeOcj2b/FqsvG7WCc5PWOAcCqStmDxl+DQZOIVFSTrE4kdpq
32  mNn/80kCgYEA88Xfmqg0ta83lpe9b6U0BaLvvs1gxgXmCmkyvK7Ru+iKyPUMzx+B+
33  BjGWeeiZuigmIhXfFu3eBs5xOgDrUxf9j55sJAFamljG4LTyun378RnOdA87fflq
34  rpf4oPKVfTrfXXz2keIg0eX2tD6Lsn3+MJwYqpefovxyJA3kPgCgV0CgYEA50H0
35  HQififZ22nApGpf/jJpU7hBLC45cSXwE2MX2I3rd3ptGwzKRo/lZks1bvQutqRln
36  slyEF+c9LCz6g7FYHJoewChLqCVfeZ9GxBzHeJloX2wmxDX18L4vmEDphwlcV8b3
37  ExHqUIMGuINHGelPIRILKESbTQU+OVHuNv443MCgYEA7rMKYhl1C6bYCsjowSMG
38  TqKembX84cqyl+zstp+EVbi99Usm0Lc4f/4cd6EQrp1Twbqi6YPgDdAmRQLTalp
39  e3FIOFVub4aQrOXgDEcC5bI8W57yxUr2JLjjYs5HHQoB4Dw5m0TOMFnS+enoxs3i
40  kly3Nowjz+fRCYFWN8kLVE0CgYEA43CLLK7ZcW9XKa2cNB00PElg8A4YMJJfk2nl
41  zKjNj1F9ujyO2NV4RYOsI+RSsFe3ARdJcS6xP200Tc8ixrh57VhCnAxFdGb1QpFy
42  oNgJGkf9zjPoMJsqykjSOHTG+CctqaqmPxxkkLScbIW4PPSn/U6KDPNHpVNOuQeO
43  hXHak58CgYBLW1719vgYhUisWc9Gd3mCSxpAb6y8RcyTgqF76K8v4MallPqFkEtD
44  0BaFt1A+PtMLk2ODTRH4XU18oc9eV+7VDFkPJ8T0A2VwjzjMgNAd+vKlm4nOEBTt
45  UhegY0k8yLxSl2vuYiVnHvKBIOF/G2ckwrxjO9KVE+SA45Ex0Px5qQ==
46  -----END RSA PRIVATE KEY-----

```

## JMobile PC Runtime

When using JMobile PC Runtime the certificate files can be found inside the folder:  
 %AppData%\Exor\<Version>\server\config\ssl-certificate

## Batch files to manage certificate files using the public OpenSSL-Win32 tool

Here are example of how certificate files can be managed using a public OpenSSL-Win32 library

Reference.: <https://www.openssl.org/>

### Creating self-signed certificates

```
@echo off
set OpenSSL="C:\Program Files (x86)\OpenSSL-Win32\bin\openssl.exe"
set CertificateName=HMIDeviceID

rem Generate an RSA key
%OpenSSL% genrsa -out ssl-certificate.key 2048

rem Creating Certificate Signing Requests
%OpenSSL% req -new -key ssl-certificate.key -out ssl-certificate.csr -subj
"/C=IT/L=Verona/O=CompanyName/OU=RD Team/CN=%CertificateName%"

rem Creating Certificate
%OpenSSL% x509 -req -days 3650 -in ssl-certificate.csr -signkey ssl-
certificate.key -out ssl-certificate.crt

pause
```

### Verify a signed report

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

rem Extract public key from the certificate
%OpenSSL% x509 -in HMI.crt -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
```



SaveEventArchive (see "[SaveEventArchive](#)" on page 194) generates signed reports using the "sha256" algorithm.

See project properties in "[Project](#)" on page 78 for other available algorithms.

## 21 CFR Part 11 - Tips

Suggestions to development a CFR11 compliant application

### 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,





## 24 User management and passwords

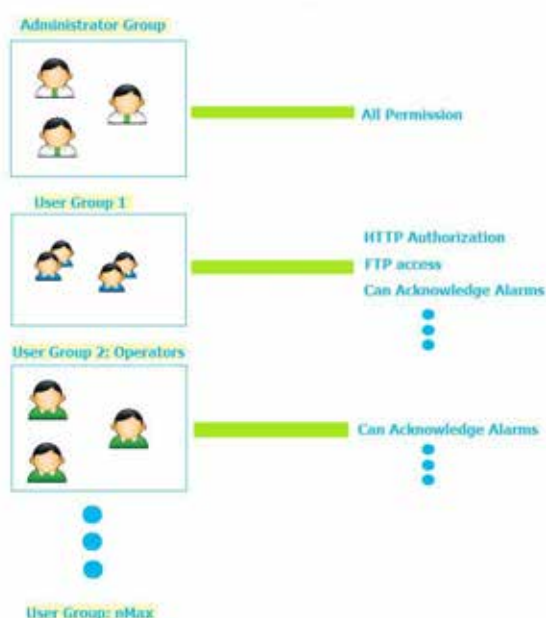
You can restrict access to various widgets and operations by configuring users, users groups and assigning specific authorizations to each group.

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.

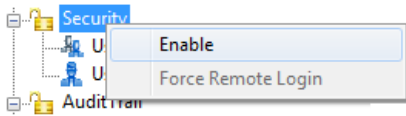


Enable/disable security management .....	310
Configuring groups and authorizations .....	310
Modifying access permissions .....	311
Assigning widget permissions from page view .....	316
Configuring users .....	317
Default user .....	318
Managing users at run time .....	318
Force remote login .....	319

# Enable/disable security management

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

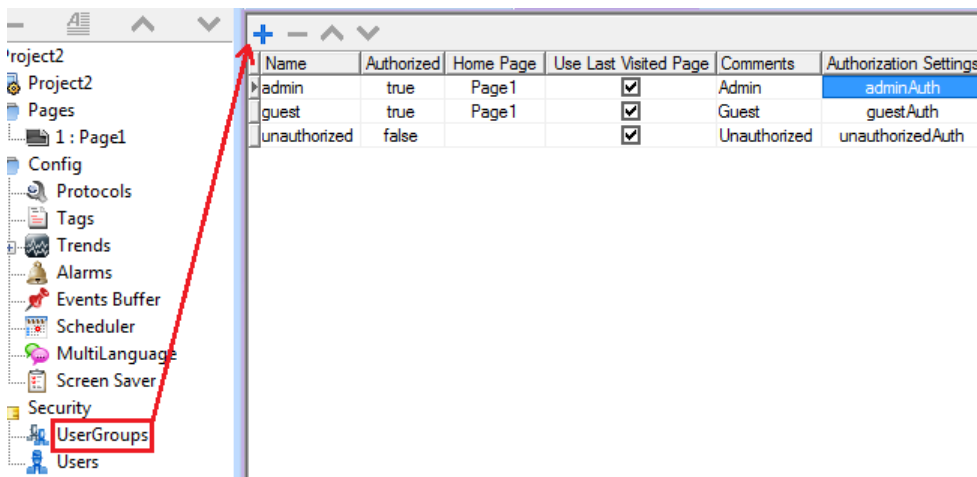
The padlock symbol indicates whether the function is enabled or disabled.



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

## Configuring groups and authorizations

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



Three predefined groups are available by default (**admin**, **guest** and **unauthorized**): they cannot be deleted nor renamed. You can, however, modify authorizations and other settings.

### Adding a user group

Click **+** to add user group.

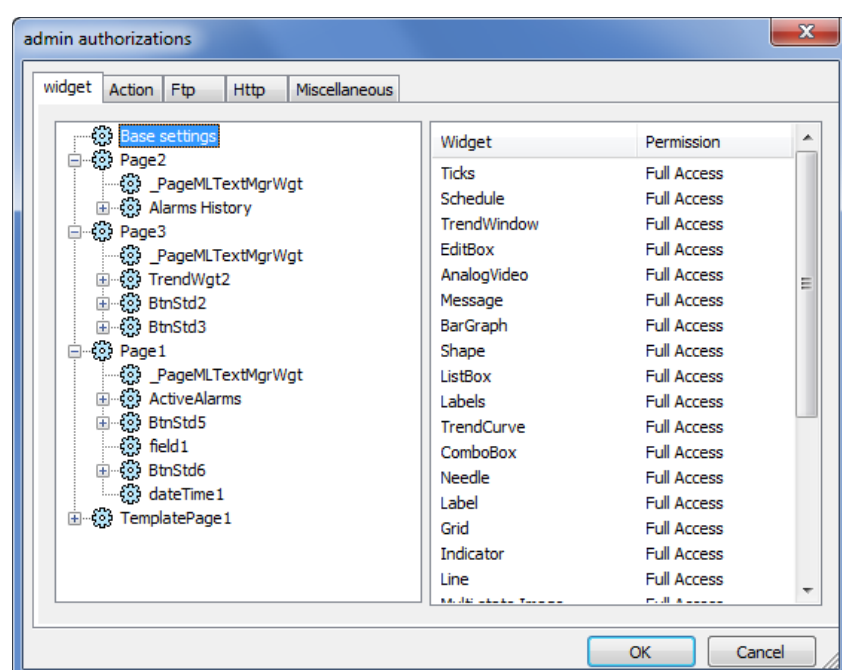
Parameter	Description
<b>Name</b>	Name of users group
<b>Authorized</b>	Authorization granted
<b>Home Page</b>	Page displayed when users belonging to this group log in
<b>Use Last Visited Page</b>	When selected, the last page displayed by the previous user will be displayed when users belonging to this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

Parameter	Description
<b>Comments</b>	Any comment or description for the group
<b>Authorization Settings</b>	Opens the Admin Authorization dialog to set access permissions. See " <a href="#">Modifying access permissions</a> " below for details.

## Modifying access permissions

*Path: **ProjectView** > **Security** > double-click **UserGroups** > **Authorization Settings** column*

Click the button: a dialog appears with a list of widgets and actions. You can modify access permissions for each one in the list.



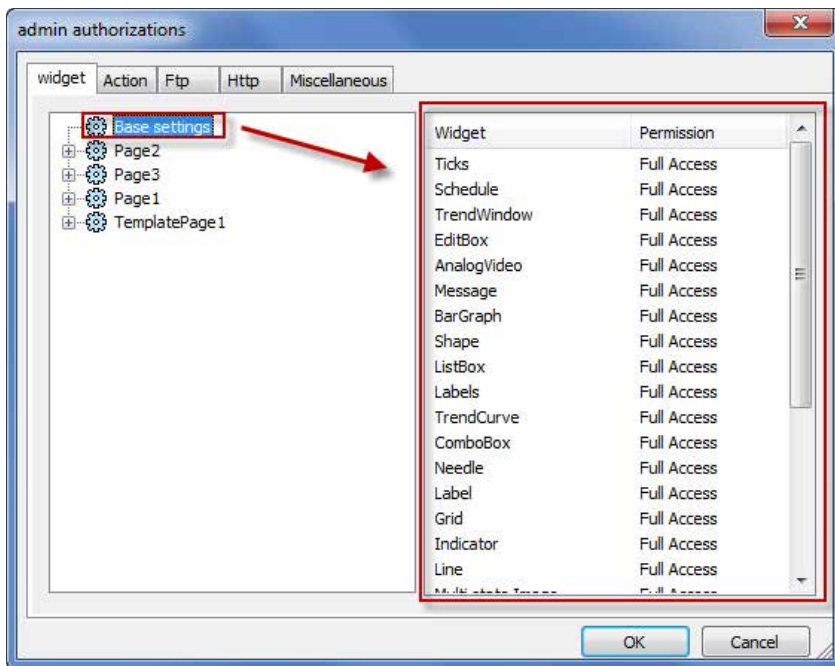
## Widget permissions

In the **Widget** tab you can define widget access options at project level, at page level or at widget level for all the widgets used in the project. Lower levels permission (for example, widget level) overrides higher levels (that is, page and project levels).

Use **Base settings** to set default permissions at project level.

Possible settings are:

- **Full Access** to enable read/write access to the widget
- **Read Only** to enable readonly access to the widget
- **Hide** to hide widget for selected group



## Changing a widget permission

To change access permission for an individual widget in a page of the project, navigate to that widget within its page on the right pane and customize its access options. Otherwise, all widgets take the permissions set at project or page level.

For example, if page permission for a widget is set at project level to **Read Only**, then all the same widgets will have permission **Read Only**. When you select a widget inside a page from the tree structure, permission is actually set to **Use Base Settings**. You can change this setting and modify access permissions only for this widget in this page.

## Access priority

Widget permissions are considered with the following priority:

Permission level	Priority
Project level - Basic settings	Low
Page level	Medium
Widget level	High

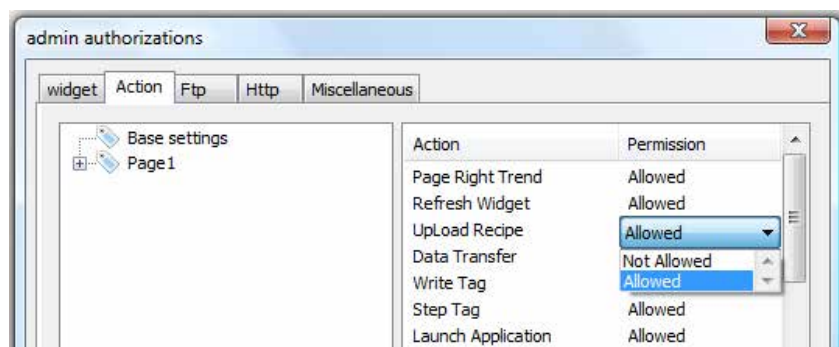
This allows you to specify exceptions for an action or a widget directly from the page view.

For example, if you set permissions for a widget at project level to Read Only and to Full Access at page level then the page level settings will prevail.

Access permissions can be modified directly from the project page. See ["Assigning widget permissions from page view" on page 316](#) for details.

## Action permissions

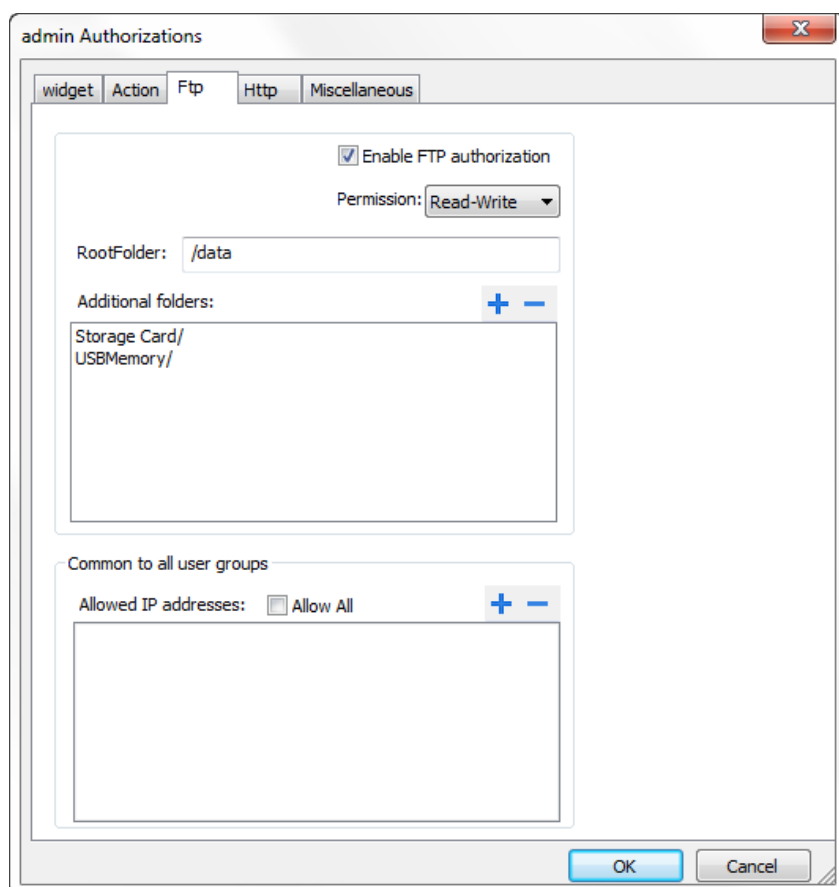
In the **Action** tab you can define action authorizations at project level, at page level or at widget level. Actions can be either **Allowed** or **Not Allowed**.




Action permissions can be modified directly from the project page. See ["Assigning widget permissions from page view"](#) on [page 316](#) for details.

## FTP authorizations

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



Element	Description
<b>Enable FTP authorization</b>	Enables the FTP function for the specific group
<b>Permission</b>	Type of permission: <ul style="list-style-type: none"> <li><b>Read-Only</b></li> </ul>

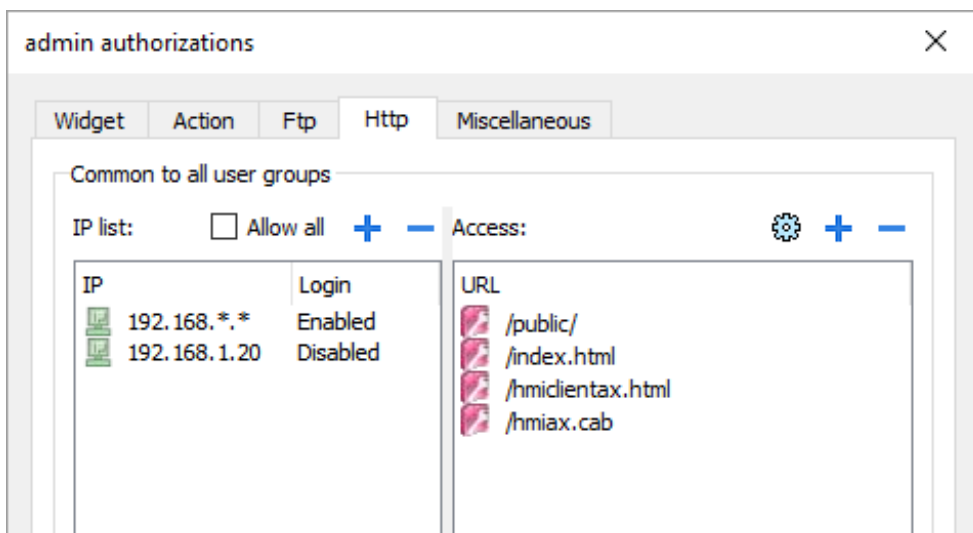
Element	Description
	<ul style="list-style-type: none"> <li>• <b>Read-Write</b></li> </ul>
<b>Root Folder</b>	Folder to be used as root for FTP access. This is a relative path.
<b>Additional folder</b>	Extra folders to be used as root for FTP access (for example, on USB drive or SD card)
<b>Allowed IP Addresses</b>	List of IP addresses from which FTP connection can be accepted. <div>  <b>Important: This setting is common to all users groups.</b> </div>

## HTTP authorizations

In the **HTTP** tab you set restrictions to HTTP access to the web server integrated in JMobile HMI Runtime.

Wildcards can be used to identify a range of IP addresses.

For example, the two following rules set the HMI device unit can only be accessed by all the IP addresses 192.168.\*.\* on your local network in which only the IP address of 192.168.1.20 can access the device without entering a login name.



Element	Description
<b>IP list</b>	IP addresses authorized to access the HTTP server. By default all.
<b>Login</b>	When disabled, the username and password are not required.
<b>Access limits</b>	List of resources for which access is limited

Effect of these settings depends on whether the option **Force Remote Login** has been selected. See "[Force remote login](#)" on page 319 for details.

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



**Important: This setting is common to all users groups.**

## Adding an HTTP configuration

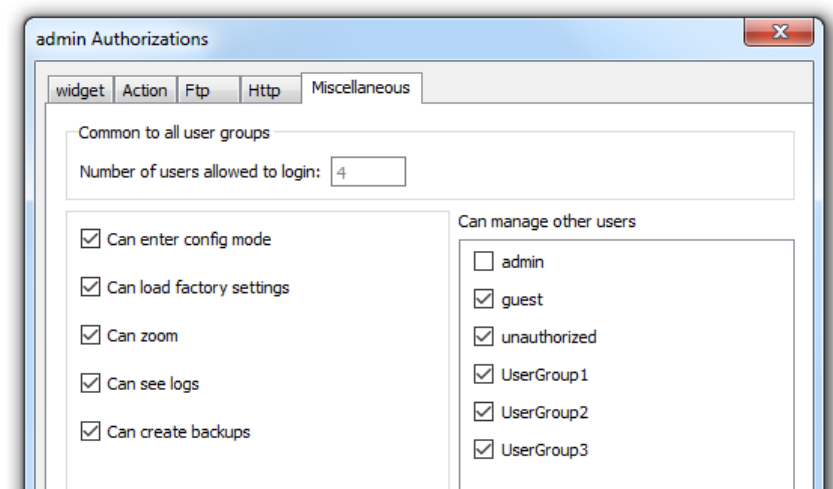
To add and configure a new access click **+**: the **Access limits** dialog is displayed.

To restore the default configuration click the **Set default access limits** icon. Default configuration allows access to the following:

- PUBLIC folder and Index.html

## Miscellaneous settings

In the **Miscellaneous** tab you can define various authorization settings.



**Note:** Some of these settings are group specific, while other are common to all groups.

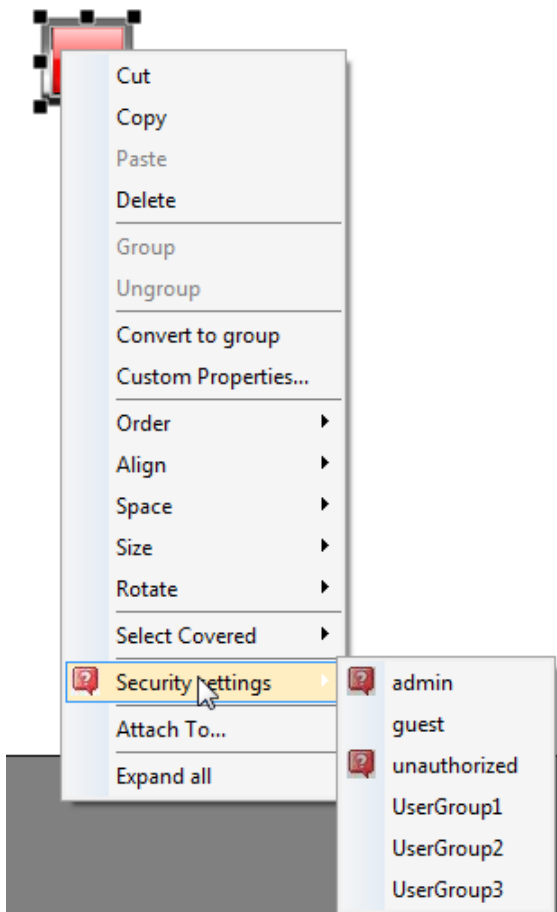
Option	Description
<b>Can enter config mode</b>	Enables switching from runtime to configuration mode. Normally used for maintenance.
<b>Can manage other users</b>	Gives super user privileges at run time to manage the select groups. Allows adding, deleting and modifying users' permissions.

Option	Description
<b>Can load factory settings</b>	Restores factory settings.
<b>Can zoom</b>	Enables zoom in/out in context menu at run time
<b>Can see log</b>	Allows user to see logs at run time
<b>Can create backup</b>	Allows user to backup project.
<b>Number of users allowed to login</b>	Maximum number of users that can be connected to the HMI Runtime at the same time. Default is 3.

## 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.



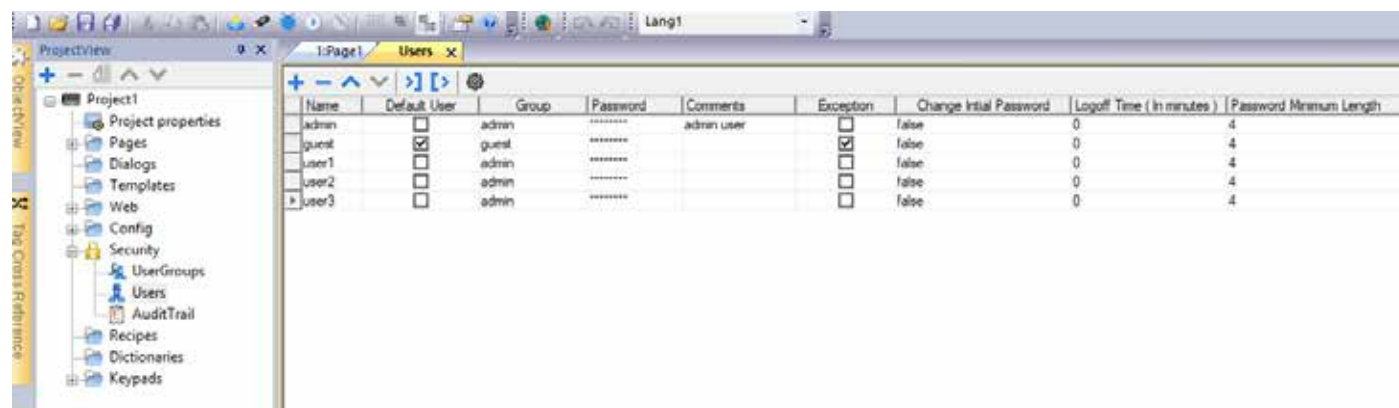
See ["Modifying access permissions" on page 311](#) for details.




# Configuring users

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

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



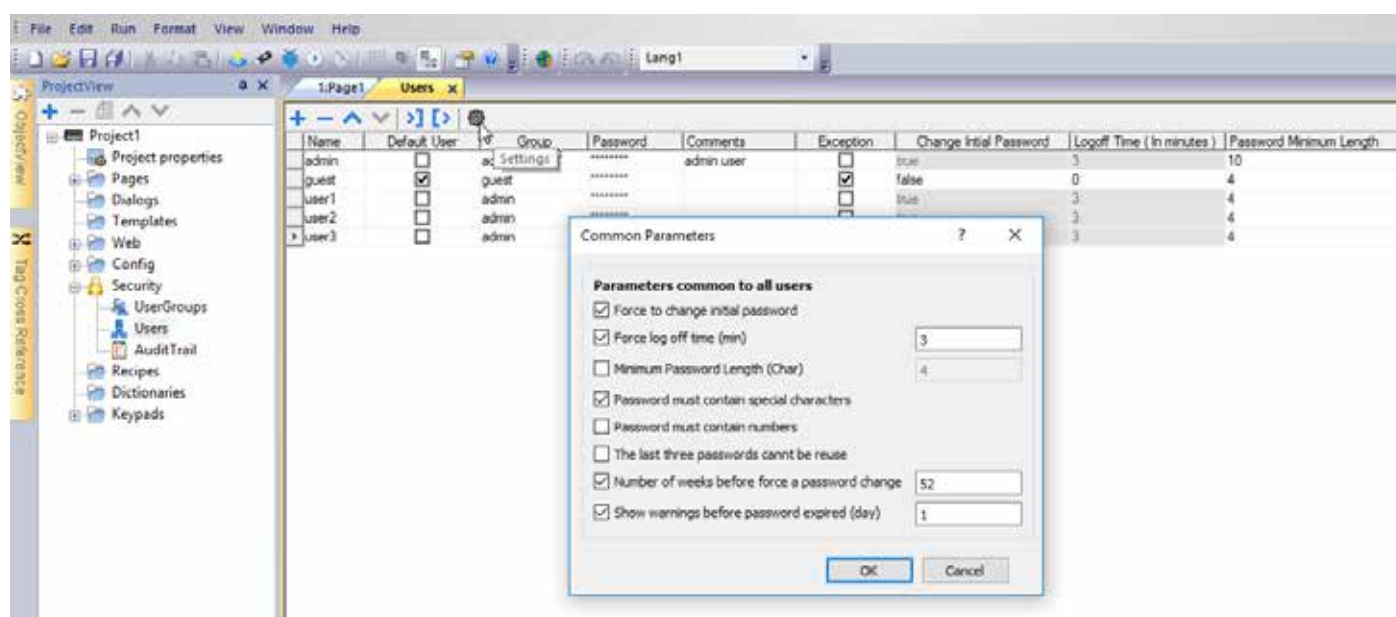
Parameter	Description
<b>Name</b>	User name
<b>Default User</b>	This user is automatically logged in when the system is started or after another user has logged off. Only one Default user can be set
<b>Group</b>	User group
<b>Password</b>	User password. Note that for security reasons the password will never be displayed  <b>Passwords are encrypted and cannot be retrieved not even for specialized technicians</b>
<b>Comments</b>	Further user description
<b>Exception</b>	Allows to change the values forced from the User Settings parameters
<b>Change Initial Password</b>	This user is forced to change his password at first log in
<b>Logoff time (minutes)</b>	Minutes of inactivity after which the user is logged off. Set to 0 to disable
<b>Password Minimum Length</b>	Minimum length of password
<b>Must Contain Special Characters</b>	Password must contain at least one special character
<b>Must Contain Numbers</b>	Password must contain at least one numeric digit
<b>Password cannot be reused</b>	The new password must be different from the last 3 used passwords

Parameter	Description
<b>Password aging (weeks)</b>	Number of weeks before forcing a password change (1/52 weeks)
<b>Warning (days)</b>	Show a warning message before password expires (1/30 days)

## Users Settings

From the Settings command, there is the possibility to define parameters values that will be common to all users.

Users with the Exception flag checked are not force to use the common parameters.



## Default user

You can define only one default user in a project. This is the user automatically logged in at system start up and when the currently logged user logs out or is logged out after time-out.

To log into JMobile HMI Runtime with a different user, use one of the actions:

- **SwitchUser**
- **LogOut**

See "[User management actions](#)" on page 202 for details.

## Managing users at run time

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

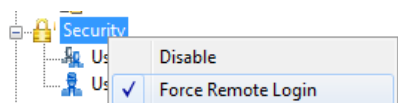
## Removing user data

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

- on HMI Runtime: execute the action DeleteUMDynamicFile
- with JMobile Studio: select the **Delete Dynamic Files** in the download dialog.

## Force remote login

Path: **ProjectView** > right-click **Security** > **Force Remote Login**



Select this option to force user to log in when using remote access via HMI Client. If not selected, remote access will use the same level of protection of local access.



**Important: This function only works when user management is enabled.**



**Tip:** 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 310](#) 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 311](#) for details on HTTP access limits.



## 25 Audit trails

---

The Audit trail is a chronological sequence of audit records. Each record contains information on the actions executed and the user that performed them.

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

---

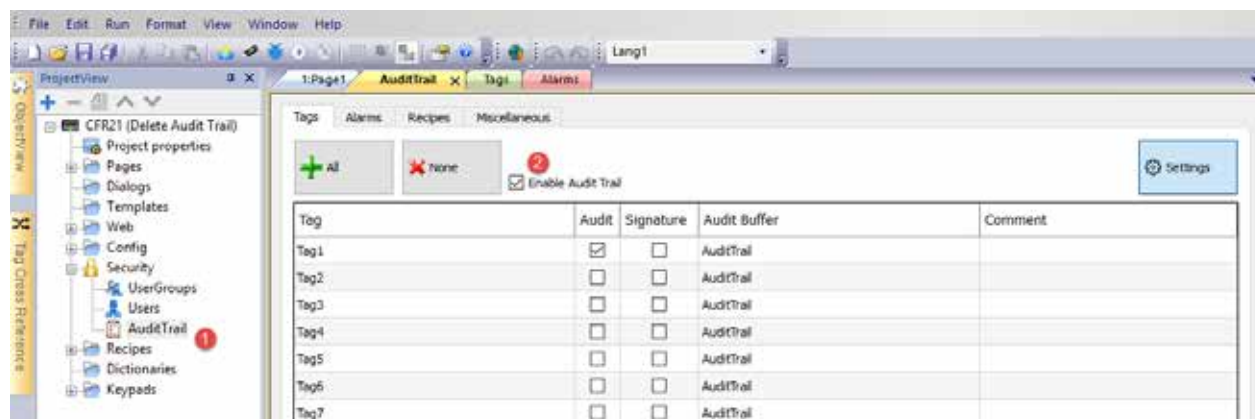
<b>Enable/disable audit trail .....</b>	<b>322</b>
<b>Electronic Signature .....</b>	<b>323</b>
<b>Table audit widget .....</b>	<b>326</b>
<b>Exporting audit trail as .csv files .....</b>	<b>327</b>
<b>Configure audit events .....</b>	<b>328</b>

# Enable/disable audit trail

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

Audit trail logging can be enabled from the “Enable Audit Trail” check box

When enabled, all changes to the selected resources will be logged to the audit buffer with the time stamp, user name that performed the operation and some additional information concerning the modified resource (e.g. new value and previous value for tags)



From the main tabs (Tags, Alarms, Recipes and Miscellaneous) of the Audit trail Editor you can switch between the list views of the available resources.

Parameter	Description
<b>Audit</b>	Enable tracking of the selected resource
<b>Signature</b>	The user password is required before allowing the resource to be modified from the user (see " <a href="#">Electronic Signature</a> " on the facing page to additional information)
<b>Audit Buffer</b>	Internal buffer where store the related audit events (see " <a href="#">Configure audit events</a> " on page 328 to additional information)
<b>Comment</b>	Comment space available for the developers

## Tags

- Keep track of when tag value changes.

## Alarms

- Keep track of when user acknowledges or resets an alarm event

## Recipes

- Keep track of when user downloads or uploads recipes

## Miscellaneous Resources

- User login details  
Keep track of when user login, logout or change password
- User management actions  
Keep track of when a user is added, removed or when the user properties are modified
- System actions  
Keep track of system actions (HMI Device Restart, Power On, Backup, Update, Download, enter in System Setting, open Project Manager)
- FTP actions  
Keep track of ftpGET, ftpPUT, OpenTextEditor, SaveTextEditor
- Buffer actions  
Keep track of dump and delete actions on alarms, audit or trends buffers

### LogMessage Macro

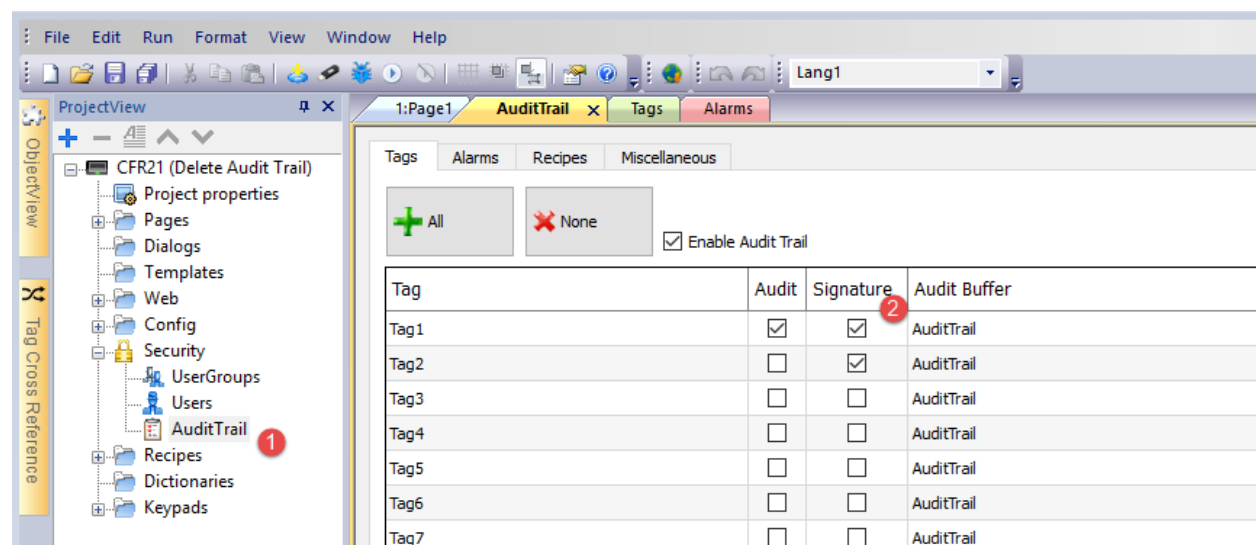
In addition of that, the LogMessage macro gives the possibility to define additional events to log to the audit trail buffer.

See "[LogMessage](#)" on page 195 for additional details.

## Electronic Signature

For each resources listed within the Audit Trail editor, it is possible configure the JMobile HMI Runtime to require the password confirmation before changing it. If the audit trail log is enabled, the user has the option of adding a comment that will be recorded within the Track Log.

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



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

# Confirm your password

Password:

\*\*\*\*\*

Comment:

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

Ok

Cancel



The introduced password will be not required again for the commands released in the next 10 Sec. The validity time can be modified from the Settings dialog.

[Settings](#)

Signature	Audit Buffer	Comment
<input checked="" type="checkbox"/>	AuditTrail	
<input checked="" type="checkbox"/>	AuditTrail	
<input type="checkbox"/>	AuditTrail	
<input type="checkbox"/>	AuditTrail	
<input type="checkbox"/>	AuditTrail	
<input type="checkbox"/>	AuditTrail	
<input type="checkbox"/>	AuditTrail	
<input type="checkbox"/>	AuditTrail	
<input type="checkbox"/>	AuditTrail	
<input type="checkbox"/>	AuditTrail	

Settings

Signature password validity (Sec):

OK
Cancel





# 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

Duration : 1 Hour

Refresh

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

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

Backward

Forward

### Buttons:

- **REFRESH**  
Retrieve trend data from internal buffer and refresh table view
- **BACKWARD/FORWARD**  
Move the display window forward or backward as specified in the duration parameter

### Filter:

Use the combo box to select the column where search for and the text filed on the right to enter the string to search to.

Parameter	Description
<b>AuditBuffer</b>	Event Buffer from which the event list is retrieved (see <a href="#">"Configure audit events" on page 328</a> )
<b>Heading</b>	Heading label
<b>Default Duration</b>	Initial value of time window to show
<b>Time Spec</b>	Time format: <ul style="list-style-type: none"> <li>• <b>Local</b> = show the time values of the HMI device.</li> <li>• <b>Global</b> = show the time values using UTC format.</li> </ul>

Parameter	Description
Date Format	Select the Date and Time format
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

## 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 194](#) 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 JMobile HMI Runtime application
Action	Action executed.

### Exported data file has the following content

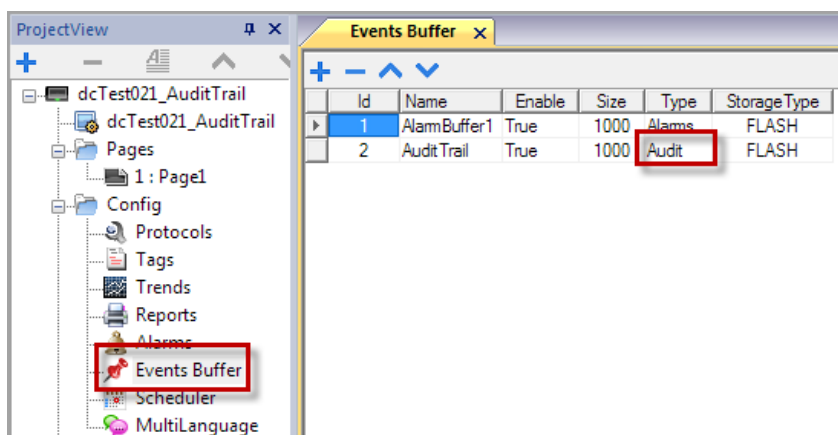
<b>Status</b>	Result of the executed action <ul style="list-style-type: none"> <li>• S_OK Action executed correctly</li> <li>• E_FAIL Action non executed</li> <li>• S_NEEDNOT_NOTIFY Action triggered (will be executed asynchronously)</li> </ul>
<b>Information</b>	Additional info related with the executed action.

## Configure audit events

You can have more than one set of audit records. You need to configure a dedicated event buffer.

### Creating an event buffer

Path: **ProjectView**> **Config**> double-click **Event Buffer**



1. In the **Event Buffer** editor, click **+**: a row is added to the table.
2. Select **Audit** for **Type**.
3. Configure buffer parameters.

Parameter	Description
<b>Id</b>	Buffer identification number
<b>Name</b>	Buffer name
<b>Enable</b>	Enable/disable logging
<b>Size</b>	Size of log file. Data is automatically saved to disk every 5 minutes.
<b>Type</b>	Type of events logged: <ul style="list-style-type: none"> <li>• <b>Alarms</b></li> </ul>

Parameter	Description
	<ul style="list-style-type: none"><li>• Audit</li><li>• Generic</li></ul>
Storage Device	Device where audit data will be stored



## 26 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, JMobile Studio provides access to a dedicated widget gallery featuring only widgets available for reports.

Reports format can be customized using predefined templates for page layout.



Note: Report printing is not supported by HMI Client.

---



<b>Adding a report .....</b>	<b>332</b>
<b>Configuring text reports .....</b>	<b>332</b>
<b>Configuring graphic reports .....</b>	<b>333</b>
<b>Print triggering events .....</b>	<b>334</b>
<b>Default printer .....</b>	<b>335</b>

# Adding a report

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

In **Reports** editor, click **Graphic Report** or **Text Report**: one new row is added to the table.

## Report types

Report type	Description
<b>Text Reports</b>	<p>Use for line-by-line printing of alarms.</p> <p>Only used for line printers.</p> <p>Text is sent to the printer without using any special driver.</p> <p> <b>Important: This printing mode requires using a physical port and only works on Windows CE platforms.</b></p>
<b>Graphic Reports</b>	<p>Contain graphical elements and may include complex widgets such as screenshots or alarms.</p> <p> <b>Important: Each printer requires a specific printer driver. See "<a href="#">Configuring graphic reports</a>" on the facing page for a list of supported printer drivers.</b></p>

## Configuring text reports

Use the **Reports** editor . **Paper Size** in number of characters.

### Setting printer options

Use printer options to control flush of pages on printer.

Printing starts either immediately or after a timeout. In printer options you can force flush as soon as a specific condition occurs, after a specified number of events, lines or seconds.



Note: Text reports do not support PDF format.

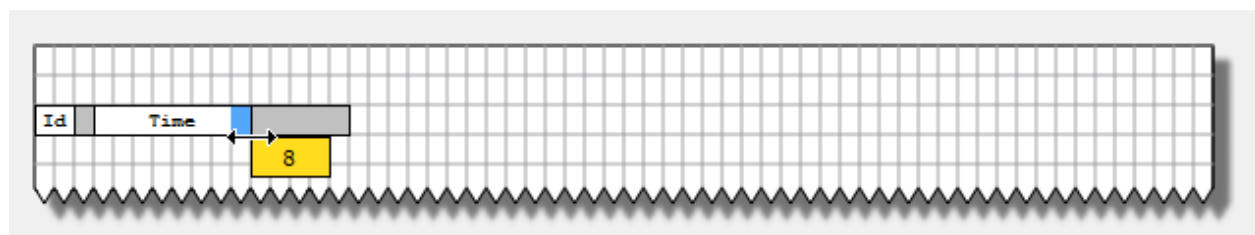
### Setting alarms layout

**Paper Size** is the width of paper in number of characters.

### Adding fields to the report

To add an item to the report, drag and drop it on the template page from the **Available fields** list.





Re-size the field using the mouse, a tool tip shows the dimension in number of characters.



Note: If the text does not fit in the dedicated space, the auto wrap is applied.

## Configuring graphic reports

Use the **Report** editor to configure graphic reports.



### Adding a report page

Click **+** to add a new page to the report layout.

When the mouse goes over a page, two icons are displayed and allow you to reorder or delete the pages.

### Modifying report page content

1. Double click on a page to edit its content: the **Graphic Report** editor appears.


Each page is divided in: header, footer and page body.

2. Double click on the area you want to edit: the edit area is shown in white, others are grayed out.

The Widget Gallery is context-sensitive and displays only the widgets available for the area you are editing.

### Widgets available for reports

Widgets that can be used for a graphic report:

Widget	Function
Page Number	Automatic page numbering
Screenshot	Screen capture of the page currently displayed by the HMI device. The report page is automatically resized to fit the HMI device page.   Note: The full screen is printed, including all open dialogs.
Alarm	Entire contents of the event buffer (default buffer is Alarm Buffer1).
Text	Widgets such as labels and numeric fields

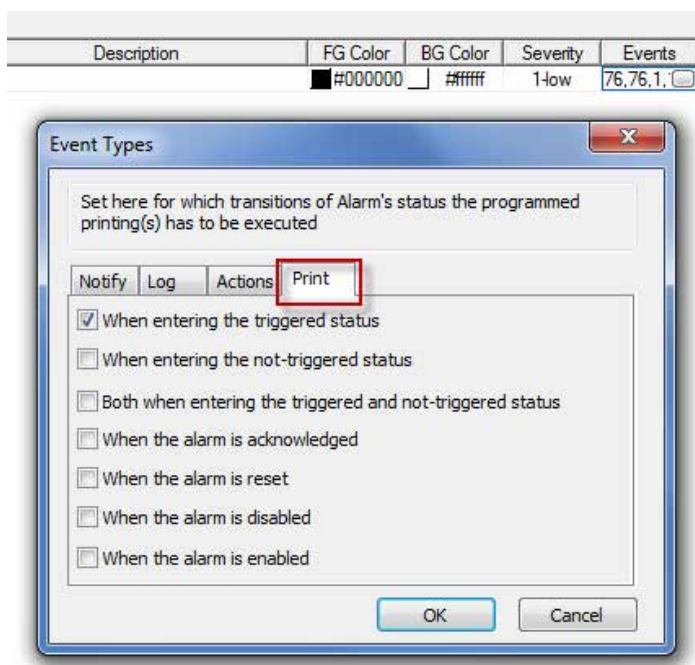
## Print triggering events

Report printing can be triggered by events.

### Configuring alarm printing

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

1. In the Alarms editor, open the **Event Types** dialog from the **Events** column.
2. In **Print** tab select all the conditions for which you want to trigger printing.

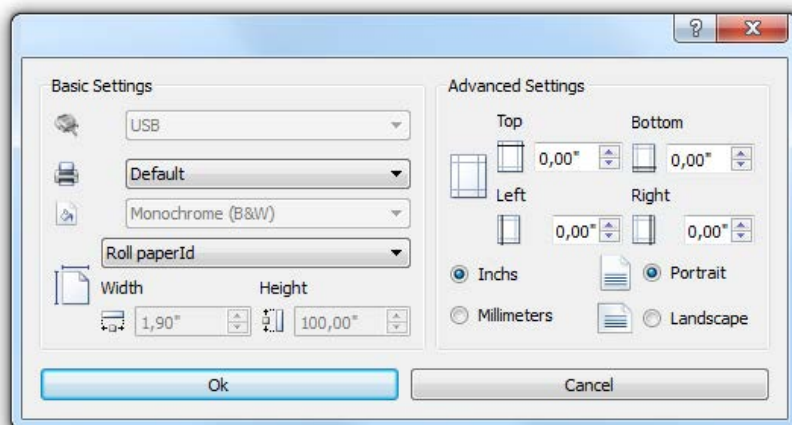


**Important:** Only one report can be set as Active alarm report in a project and it can be either a text report or a graphic report.

## Adjusting printer settings at run time

A graphic report printing can be started also using the action **PrintGraphicReport**.

Set the action property **silent** to **false** to have a pop-up dialog.



## Default printer

### Printer setting

You can set a default printer for all graphic reports. Each report can then be configured to use the default printer or any other printer available. Click **Printer Setting** button to set printer parameters.

For PDF printers you also define the folder where files are saved by using **Printed Files Location**.

### 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

Printer	Languages
<b>SPT-8</b>	SPT-8
<b>M1POS</b>	M1POS
<b>MP300</b>	MP300
<b>Zebra</b>	Zebra CPCL language
<b>Intermec PB42, PB50, PB51, PB2, PB3</b>	Intermec PB42/50/51/2/3 with ESC/P language
<b>Datamax Apex</b>	Datamax Apex

## Supported ports

The following ports are supported:


- LPT1 (USB printers)
- File (PDF)




Note: On Win32 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
<b>Custom</b>	Plus 4 Kube II	Yes	Yes
<b>Epson ESC/P 2</b>	Epson AcuLaser M2310	Yes	Simulate
<b>Epson LX (9-pin)</b>	Epson LX-300+II	No	Yes
<b>HP PCL 3</b>	HP LaserJet P2015dm	Yes	Simulate
	HP LaserJet 4700dtn	Yes	Yes
<b>HP PCL 3 GUI</b>	HP Deskjet 1010	Yes	No
	HP Deskjet D5560	Yes	No
	HP LaserJet 4700dtn	No	Yes
<b>HP PCL 5e</b>	HP LaserJet P2015dm	Yes	Simulate
	HP LaserJet 4700dtn		
<b>INTERMEC</b>	Intermec PB50 with ESC/P language with 4 inch roll paper.  Note: The HMI device crashes when trying to print on	Yes	Yes

Driver	Printer Model	Graphic	Line
	 Intermec PB50 printers in standby mode after a first successful print job.		
PDF	-	Yes	No



## 27 Screen saver

Screen saver can be used to display a slide show when the HMI device is not in use. The slide show starts after a timeout if none of the following events occur:

- touch of display
- mouse movement
- external keyboard key pressed

### Enabling the screen saver function

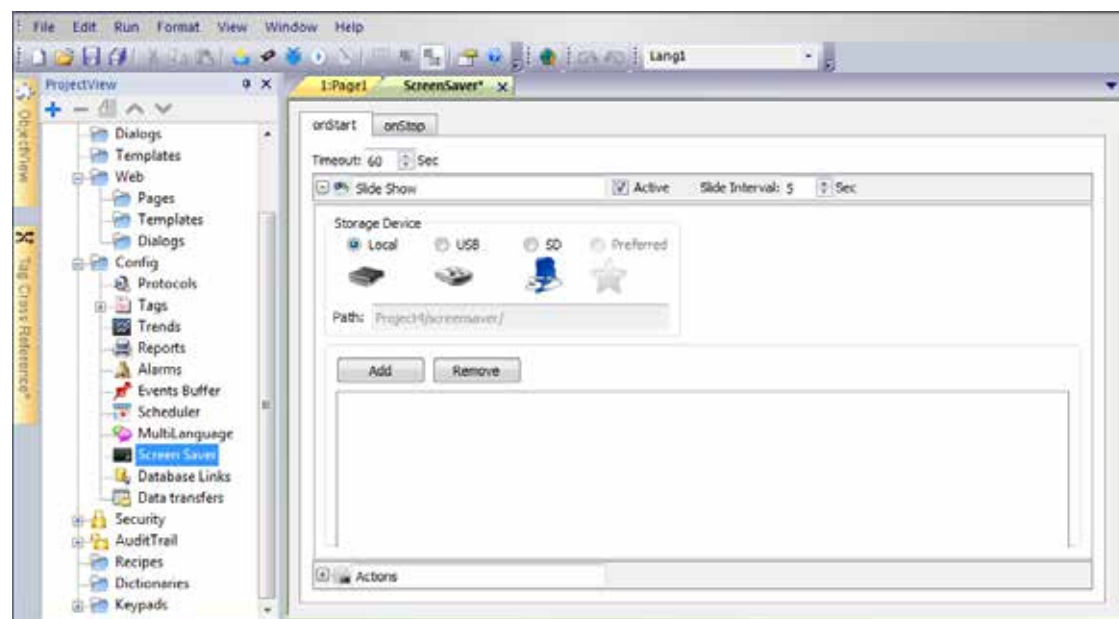
Path: **ProjectView** > **Config** > right-click **Screen Saver** > **Enable**



**Important:** You must enable the screen saver before you can configure it.


### Configuring a screen saver

Path: **ProjectView** > **Config** > double-click **Screen Saver**



### Slide show parameters

Parameter	Description
<b>Timeout</b>	Time after which the slide show starts
<b>Slide Interval</b>	Interval between slides

Parameter	Description
<b>Storage Device</b>	<p>Location of the images used in the slide show.</p> <p>Images stored locally are saved in <i>workspace\projectname\screensaver</i> and can be downloaded to the HMI device when the project is downloaded.</p> <p>Images stored on USB or SD devices are saved in a screensaver folder on the device itself.</p> <p> <b>Important: Only JPEG and PNG images are supported.</b></p>

## 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.



## 28 Backup/restore of Runtime and project

You can backup all the content of the HMI device, including

- JMobile 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 311](#) for details.



Note: Backup is not supported in Win32 / HMI Client.

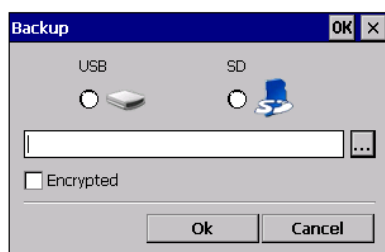
### Backup function

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 JMobile 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 JMobile 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 93](#) for details)
- or from the System Settings (see ["System Settings" on page 499](#) for details)

# 29 Keypads

Several keypads are provided by default in the JMobile Studio so that they can be used for data entry.

The alphabet keypad can be use associate with a string data type



The numeric keypad can be use associate with a numeric data type



The calendar keypad can be use associate with a date data type



Creating and using custom keypads .....	345
Deleting or renaming custom keypads .....	347
Keypad type .....	347

---

<b>Keypad position .....</b>	<b>348</b>
------------------------------	------------

## Creating and using custom keypads

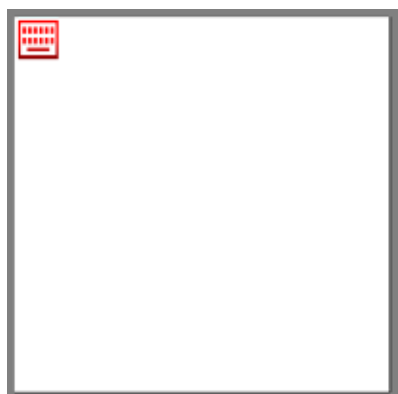
You can either create a new keypad or customize an existing one.

### Creating a keypad

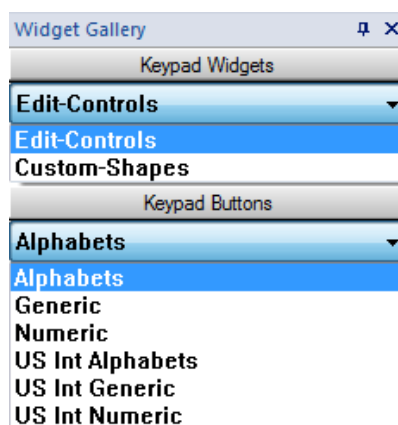
1. In **ProjectView**, right-click **Keypads** and select **Insert Keypad**: the **New Keypad** dialog is displayed.



2. Select one of the available keypads, or **Blank** to create a keypad from scratch. In this case a blank keypad is displayed.



3. Use the **Keypad Widgets** and **Keypad Buttons** from the Widget Gallery to create your custom keypad.

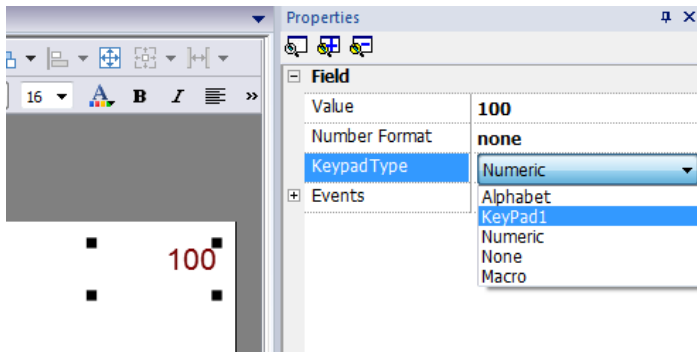


The keypad you create, as in this example, will be saved in the project folder.



## Attaching custom keypads to fields

Custom keypads can then be reused for any field where the **Keypad** property points to it as in this example.



## Tips and tricks with custom keypads

By default, any numeric widget (read/write numeric field) are assigned the numeric keypad.

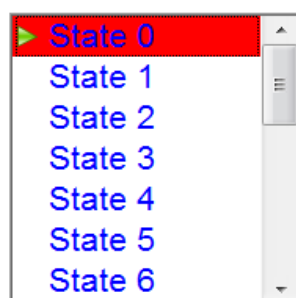
If you want to apply a customized version of the numeric keypad to all the numeric widgets you add to your project proceed as follows:

1. Create a new keypad and select **Numeric** as **Keypad** type. This will be a backup of the original settings for the numeric keypad.
2. Customize the default numeric keypad and save it. This customized version of the numeric keypad will now be assigned as default in the project.

See ["Deleting or renaming custom keypads"](#) on the facing page for details on how to rename a custom keypad.

## Up-down arrows keypad

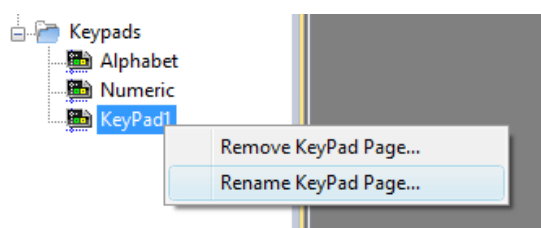
This type of keypad is particularly useful to move the cursor up and down within widget requiring this functionality. Here an example using a **Control List** widget. See ["Control list widgets"](#) on page 395 for details.



## Deleting or renaming custom keypads

In **ProjectView**, right-click on a custom keypad and select one of the options:

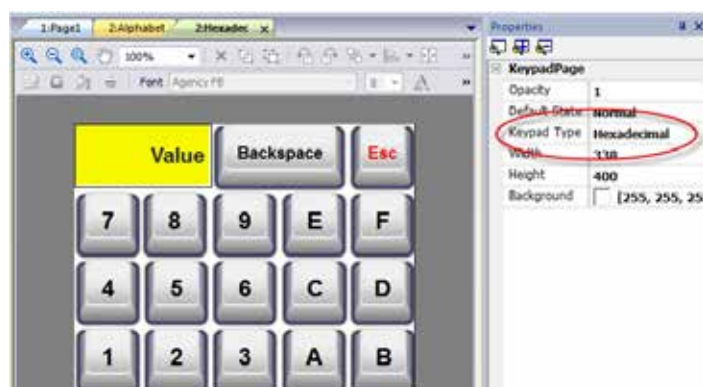
- **Remove KeyPad Page** to remove the keypad from the project
- **Rename Keypad Page** to rename the keypad.



## Keypad type

**Path:** **ProjectView** > **Keypads** > double-click a keypad > **Properties**

Set **Keypad Type** parameter for a keypad to define the type of data entry.



Keypad Type	Description
<b>Auto</b>	Default setting
<b>Decimal</b>	Only numeric keys are accepted. Entering 10, the keypad returns 10 that will be displayed as "10" if the attached field is numeric or ASCII, as 'A' if the attached field is hexadecimal.
<b>Hexadecimal</b>	Only hexadecimal keys are accepted. Entering 10, the keypad returns 16 that will be displayed as "16" if the attached field is numeric or ASCII, as "10" if the attached field is hexadecimal.
<b>Ascii</b>	All keys are enabled. Entering 1A, the keypad returns 1A that will be displayed as '1' if the attached field is numeric, as "1A" if the attached field is ASCII or if the attached field is hexadecimal.

## Keypad position

**Runtime Positioning** property of keypads can be used to define where keypads will appear in the screen.

Option	Description
<b>Automatic</b>	The best position is selected according to here data entry is required.
<b>Absolute</b>	X,Y coordinates are entered to identify the exact position
<b>Left-top</b> <b>Left-center</b> <b>Left-bottom</b> <b>Center-top</b> <b>Center-center</b> <b>Center-bottom</b> <b>Right-top</b> <b>Right-cente</b> <b>Right-bottom</b>	Predefined screen positions

Select the **Lock Keypad position** option if you don't want the keypad to be moved by dragging.



## 30 External keyboards

JMobile HMI Runtime has been designed to work with external keyboards connected via USB.

Keyboards can be used for:

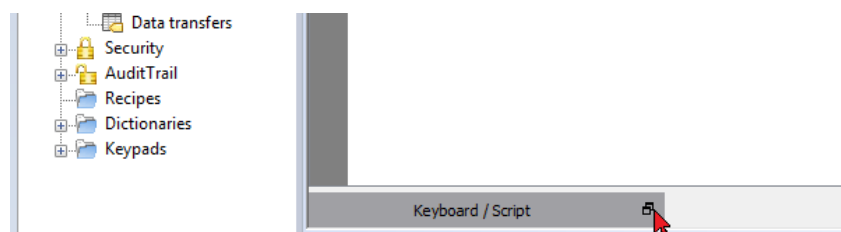
- data entry (default)
- execute actions mapped on specific keys

For example, the right arrow key **OnClick** event can be mapped to the **LoadPage** action.

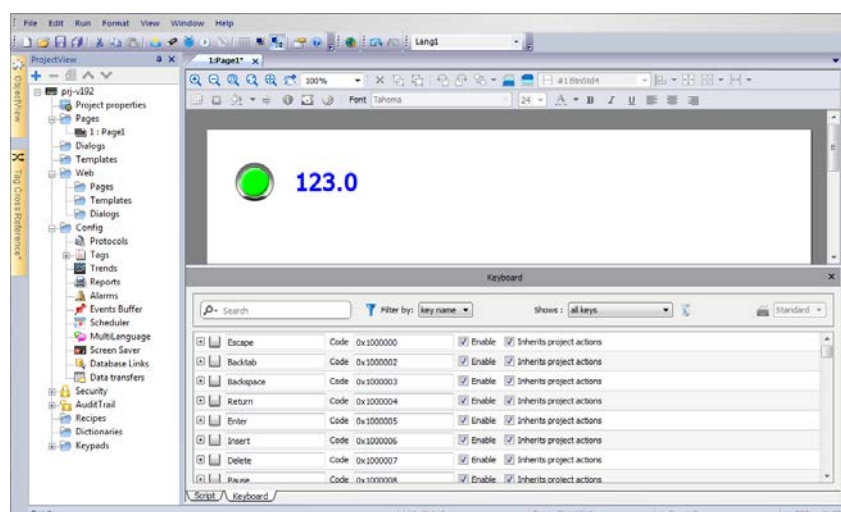
Keyboard can be programmed at project level so that settings will be inherited by all the pages. In each page you can then choose which key setting will be inherited from the project and which one you will customize for the specific page.

### 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.



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	JMobile HMI Runtime behavior
Checked	Unchecked	Action lists show the page actions (or nothing if the list is empty)	Only the page actions (if any) will be executed.
Checked	Checked	Action lists show the project actions only and cannot be edited	Only the configured project actions (if any) will be executed.
Unchecked	Checked	Inherits project actions check box and all action lists are disabled. Action lists show the project actions only.	No page or project action will be executed.
Unchecked	Unchecked	Inherits project actions check box and all action lists are disabled. Action lists show the project actions only.	No page or project action will be executed.

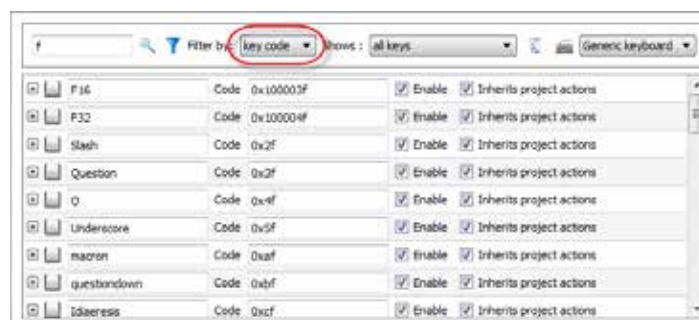
<b>Search and filter</b> .....	<b>351</b>
<b>Displayed keys</b> .....	<b>351</b>
<b>Removing action associations</b> .....	<b>351</b>
<b>Keyboard layout</b> .....	<b>352</b>
<b>Enable/disable keyboard</b> .....	<b>352</b>
<b>Associating actions to keys</b> .....	<b>352</b>

## Search and filter

To display a filtered set of keys, in **Filter by** select **key name** and type a letter in the search field: only the keys containing that letter in their name will be displayed in the Keyboard editor.



Alternatively, in **Filter by** select **key code** and type a letter in the search field: only the key containing that letter in their code will be displayed in the Keyboard editor.



## Displayed keys

You can easily select what keys will be listed in the Keyboard editor window. To display a limited set of keys, select an option in **Shows**.

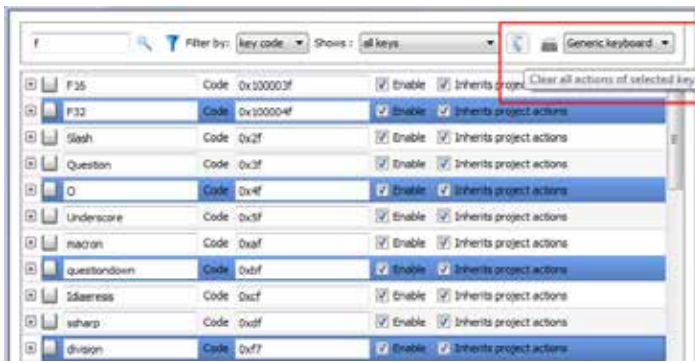
Option	Description
<b>all keys</b>	All keys available in the keyboard layout are listed
<b>modified keys</b>	Only the keys associated with actions at the page level are listed
<b>modified keys in project</b>	Only the keys associated with actions at project level are listed

## Removing action associations

To remove all the associations you created between keys and actions:

1. Select the keys for which you want to remove the association.
2. Click the **Clear all actions of selected keys** button.

If you are working at page level, page actions will be removed, if you are working at project level, project actions will be removed.

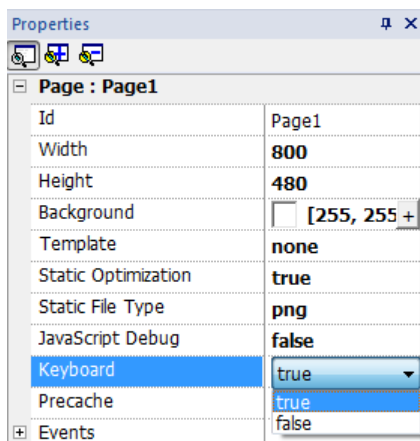


## Keyboard layout

Select the layout of the keyboard from the **Keyboard Layout** combo box. **Generic Keyboard** refers to a generic international keyboard layout.

## Enable/disable keyboard

You can enable/disable keyboard actions both at project and at page level. To enable keyboard actions, in the **Properties** pane set **Keyboard macro** to **true**.

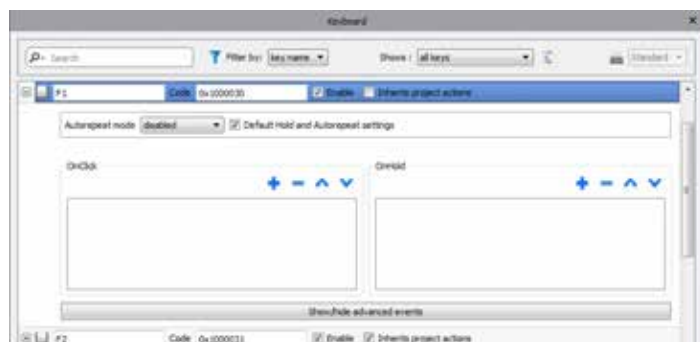


You can enable/disable keyboard actions also at run time using the KeyboardMacros action. See ["Keyboard actions" on page 165](#) for details.

## Associating actions to keys

You associate actions to a keys from the Keyboard editor.

1. Click **+** next to the key you want to program: the fields for key configuration are displayed.



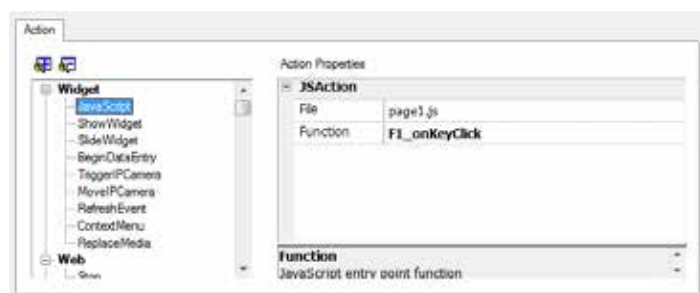
1. Click **+** to add actions.

You can associate actions both to the **OnClick** event and to the **OnHold** event.

See ["Events" on page 56](#) for details.



Note: Also JavaScript code can be associated to a key event.





## 31 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.

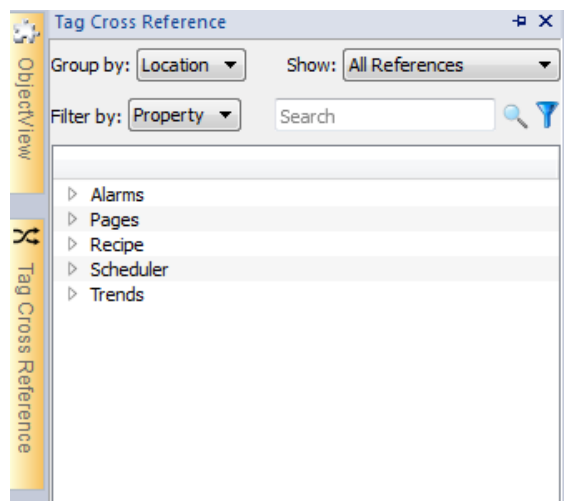
---

Updating data in the Tag Cross Reference pane .....	358
---	-----

## 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.



## Working in the Tag Cross Reference pane

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

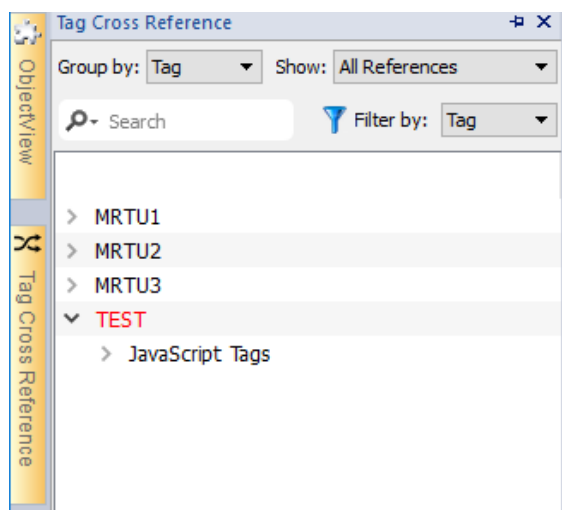
Element	Function
Group by	Groups tags by <b>Location</b> (alarms, pages, trends and so on) or <b>Tag</b> name
Show	Filters tags and displays: <ul style="list-style-type: none"><li>• <b>All Reference</b>: all tags</li><li>• <b>Invalid Tag Reference</b>: tags not listed in the Tag Editor.</li><li>• <b>Unused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

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

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

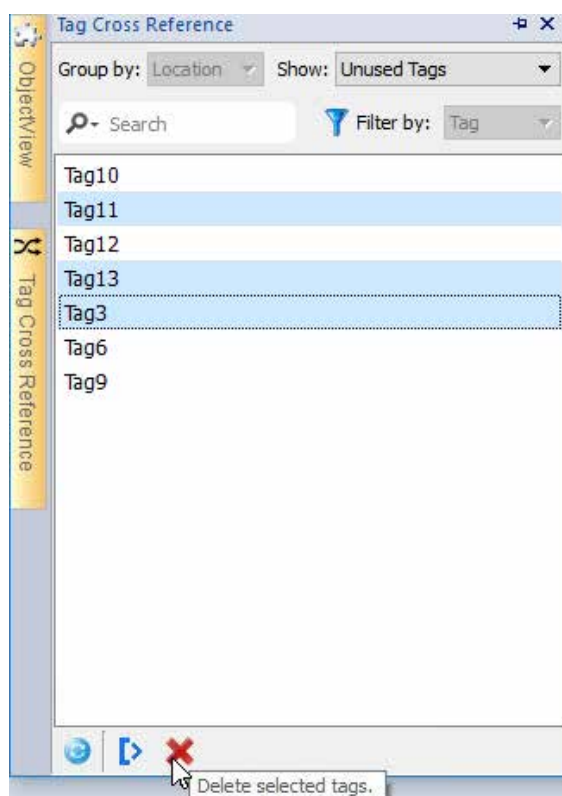
**Invalid tag references will be listed in red color:**





### Delete unused tags


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



**Be aware that eventually tags referenced inside JavaScript may not be found (depends on how the code was written). Even it is not a good practice to use 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

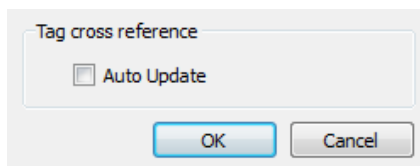
## Manual update

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 JMobile Studio **Properties** page.



Select the **Auto Update** option.

## Exporting data

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

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

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



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

## 32 Indexed addressing

---

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

---

<b>Creating an indexed addressing set .....</b>	<b>360</b>
<b>Using indexed tag set in pages .....</b>	<b>363</b>

# Creating an indexed addressing set

## Scenario

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

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

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

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

Room 1

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

## How to create an indexed tag set

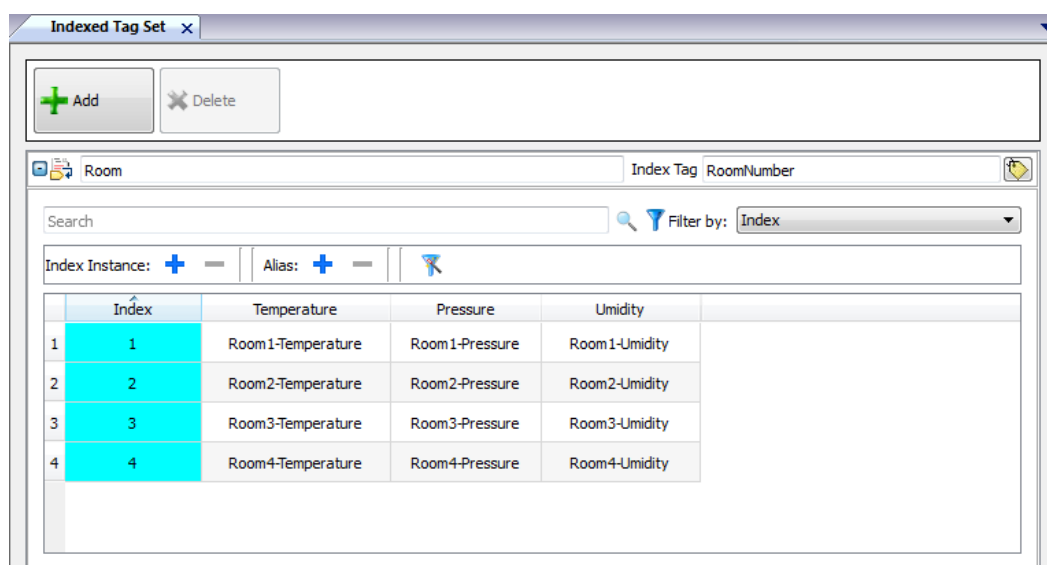
Path: **ProjectView> Tags**

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

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

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.



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

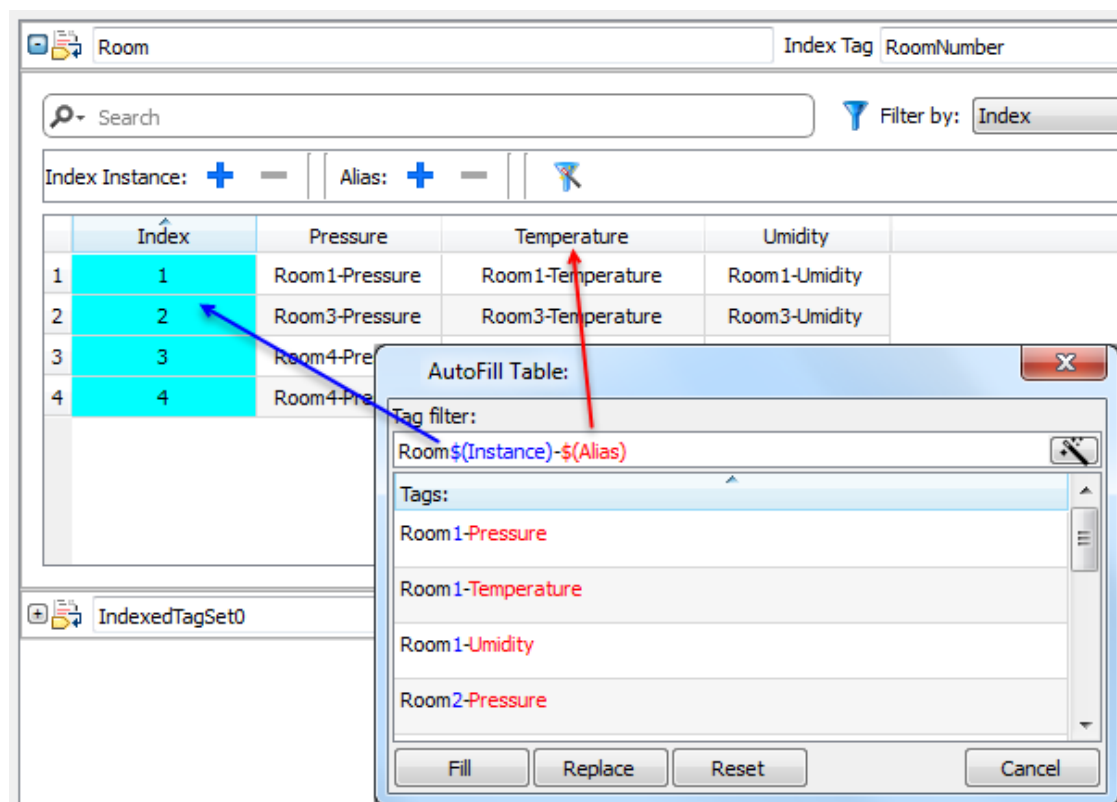


Note: To reference an array data type use the array index = -1

## 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.



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

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

## Autofill example

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

Room1-Temperature,

Room1-Pressure,

Room1-Humidity,

Room2-Temperature,

...

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

Room1-Temperature,

Room01-Pressure,

Room001-Humidity,


Room2-Temperature,

Room02-Pressure,

Room002-Humidity,

...

## Autofill table elements

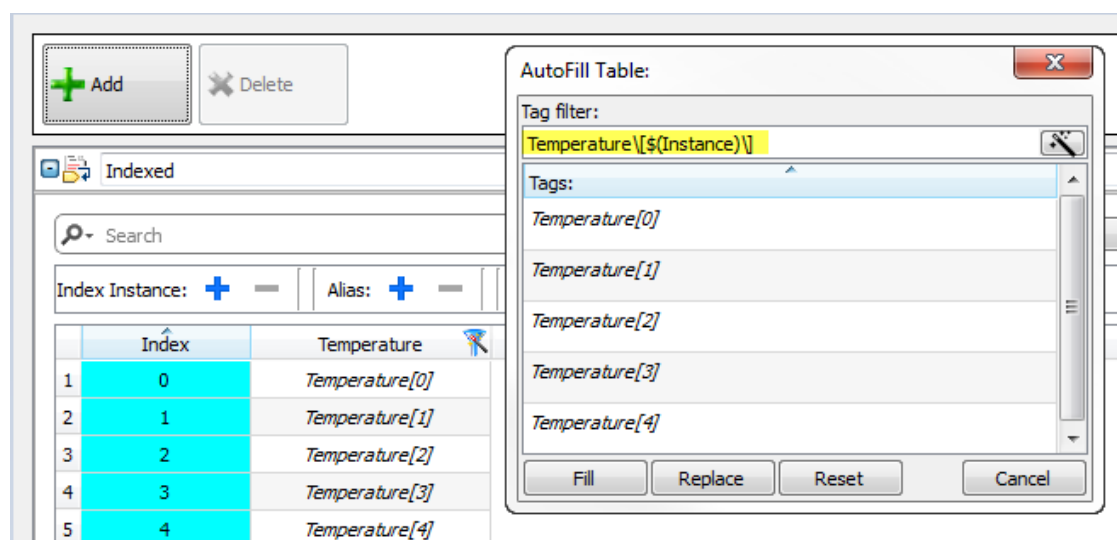
Element	Description
<b>Fill</b>	Fills in missing entries in the tag table using the set filter (if any). For example, when new instances or new aliases are added you can use this option to fill in the new entries.
<b>Replace</b>	Replace all table entries with those provided by the Autofill table.
<b>Reset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



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



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



## 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

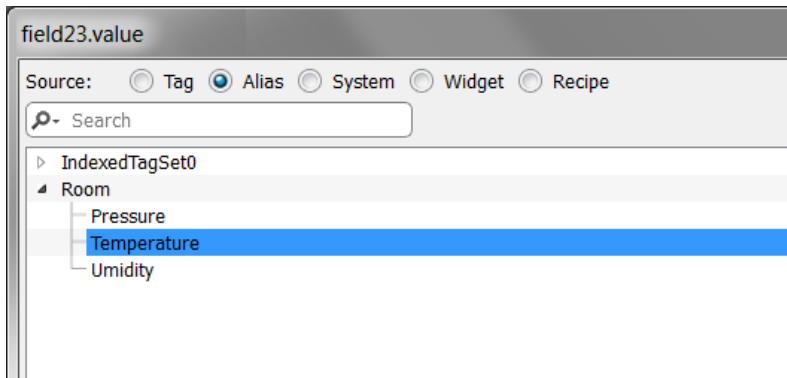
Temperature (°C)	21
Pressure	1
Umidity (%)	75

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**).

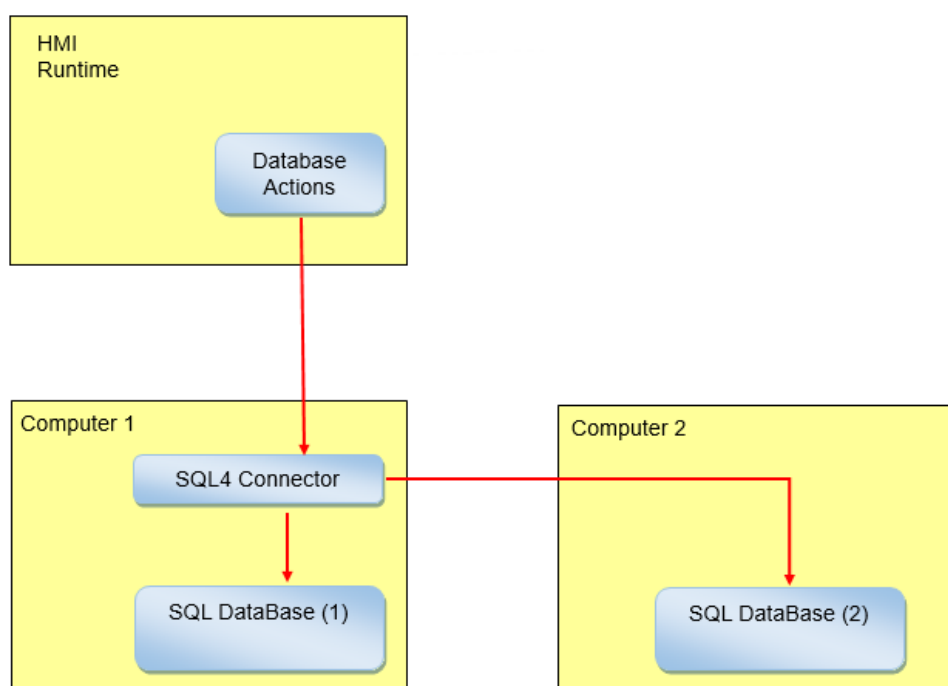




# 33 Storing data to external databases

JMobile Studio allow to connect to the SQL4Automation Connector, a software solution for the industrial usage. It connects HMI, PLC and robotic controls directly with SQL databases. HMI directly access SQL databases via the connector and can query data from tables, insert, change and delete data in tables by using SQL commands [structured query language].

The database site communicates by ODBC. Therefore all SQL databases can be integrated, which support an ODBC interface. The SQL syntax needs to be adapted to the given database, e.g. MS SQL Server, mySQL, MS Office Access, SQLite, Oracle, PostgreSQL...



To store data into an external database:

1. Install the SQL4Automation tool on the computer hosting the database or in a computer between the HMI device and the database.
2. Configure the SQL4Automation tool.
3. Create a project that use the dedicated DB actions to access at the external database.

<b>Installing SQL4Automation</b>	<b>367</b>
<b>Configuring SQL4Automation</b>	<b>367</b>
<b>Configuring the HMI project</b>	<b>369</b>
<b>Transfer data with JavaScript</b>	<b>370</b>
<b>Database tables</b>	<b>371</b>
<b>Custom tables</b>	<b>372</b>
<b>Connection Limits</b>	<b>372</b>



## Installing SQL4Automation

Download the latest version of SQL4automation and install it on the computer. Refer to [www.sql4automation.com](http://www.sql4automation.com) for details and download.

Procedures described in this document refer to SQL4Automation Connector Version 3.3.2.0

## Configuring SQL4Automation



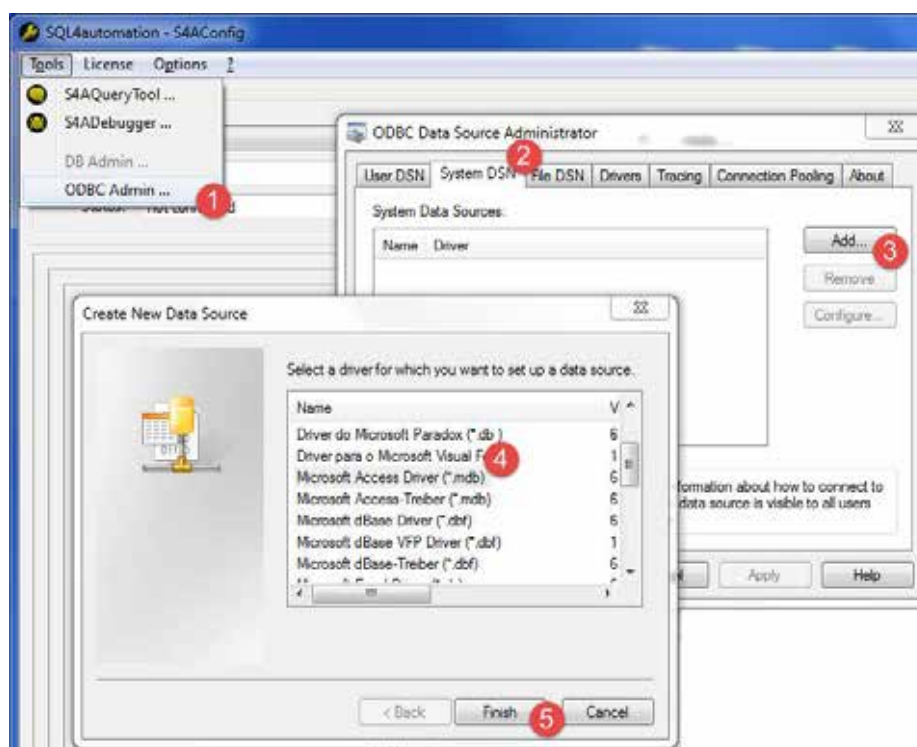
**Important: Refer to the SQL4Automation user manual for detailed configuration instruction.**

Here is a quick description of how to access to a MS Office Database (MS Access).

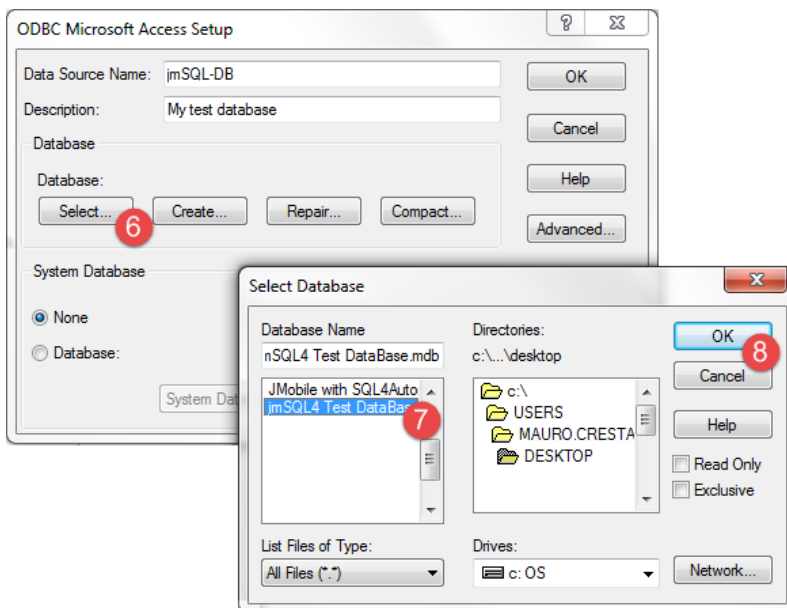
You must have the MS Office Suite installed on a computer and create an empty database using Microsoft Access.

Start SQL4Automation and follow the procedure to configure your SQL4Automation Connector:

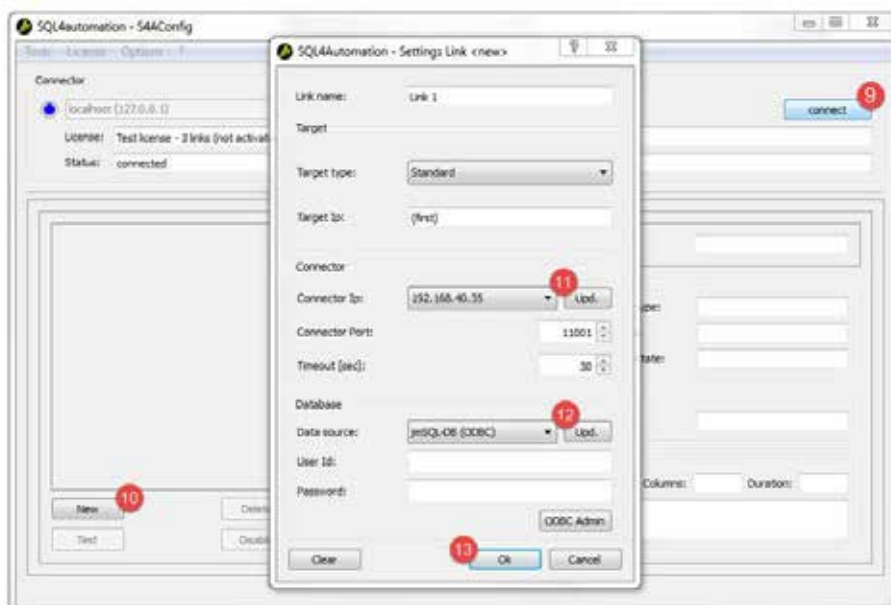
1. Select **ODBC Admin**: the **ODBC Data Source Administrator** dialog is displayed.
2. Select the **System DSN** tab.
3. Click **Add**: the **Create New Data Source** dialog is displayed.
4. Select the Microsoft Access Drive
5. Click **Finish** to confirm.



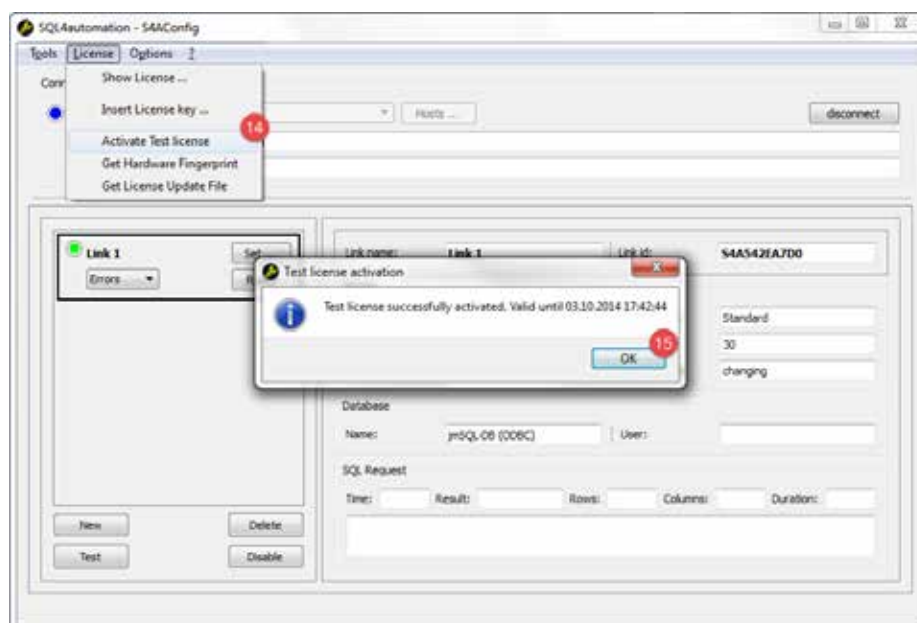
6. Enter **Data Source Name** and **Description** then click **Select**: the **Select Database** dialog is displayed.
7. Select your Access database.
8. Click **OK** to confirm.



9. At the first connection, click **New** to select your Data Source
10. Select the IP address of your computer. This will be the connection IP Address used from your HMI device.
11. Select the Data Source.
12. Click **OK** to confirm.
13. Click **Connect**.



14. Select **License> Activate Test License**: when the **Link 1** led turns green the procedure has been completed correctly.
15. Click **OK** to confirm.

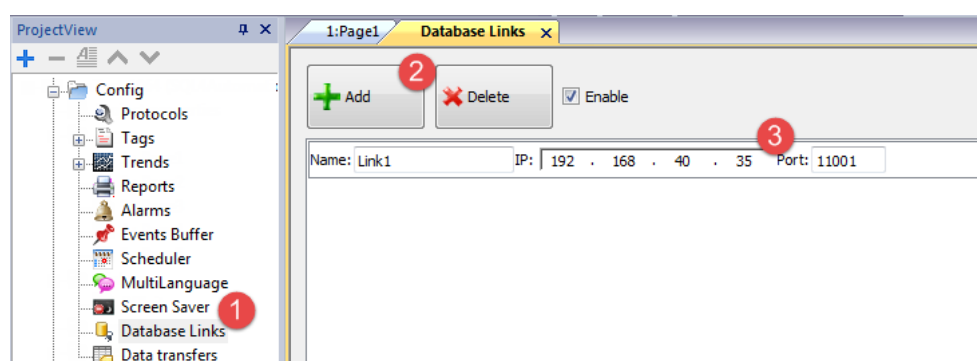


## Configuring the HMI project

Path: **ProjectView> Config > double-click Database Links**

To save a project data to an external database you need to create a link with the specific database

1. In the **Database Links** editor select **Enable** to enable the function.
2. click **Add** to create a new link.
3. Enter the IP Address the computer hosting the SQL4Automation Connector.



**Important:** The link name here is not necessarily the same defined inside the SQL4Automation Connector. But this is the name to be used in all actions using the remote database.

# Transfer data with JavaScript

Some actions used to transfer data from a HMI device to a remote database can be used as macros inside a JavaScript code as in the example below.

Status of database connection is available through system variable tags. See ["Database variables" on page 100](#).

Error status can be reset with actions. See ["Database actions" on page 161](#)

```
function myButton1_onMouseClicked(me, eventInfo) {
    var CustomSQL = ' ' ;
    var DatabaseLink ='Link1';
    project.dbInit(DatabaseLink, CustomSQL);
};

function myButton2_onMouseClicked(me, eventInfo) {
    var CustomSQL = ' ' ;
    var DatabaseLink ='Link1';
    var Tags ='Alarm1;SystemTime;Tag01;Tag02;';
    project.dbReadTags(DatabaseLink, CustomSQL, Tags);
};

function myButton3_onMouseClicked(me, eventInfo) {
    var CustomSQL = ' ' ;
    var DatabaseLink ='Link1';
    var Tags ='Alarm1;SystemTime;Tag01;Tag02;';
    project.dbWriteTags(DatabaseLink, CustomSQL, Tags);
};
```

## dbQuery

```
project.dbQuery(databaseLink, customSQL, dbCallback);
```

Using this query you can execute SQL Queries.

Parameter	Description
<b>databaseLink</b>	Link to the database to use
<b>customSQL</b>	String with the SQL query
<b>dbCallback()</b>	Function that will be call when query data are ready

## dbCallBack

```
project.dbCallBack(dbStatus, dbResponse);
```

Parameter	Description
<b>dbStatus</b>	0: no error found
<b>dbResponse</b>	<p>Query response. Table column names followed by its rows:</p> <p>In the example:</p> <p>TagName - Tagvalue</p> <p>Tag09 - 103</p> <p>Tag10 - 302</p>

```

1
2  function JS1_onMouseClicked(me, eventInfo) {
3
4      var customSQL = "SELECT Tagname, Tagvalue FROM Tags WHERE Tagname='Tag09' OR Tagname='Tag10' ORDER BY Tagname"
5      var databaseLink = "Link1";
6      project.dbQuery(databaseLink, customSQL, dbCallback)
7  };
8
9
10 function dbCallback(dbStatus, dbResponse){
11
12     alert("SQL Answer = " + dbResponse + "\ndbStatus =" + dbStatus);
13 };
14
15

```

## Database tables

Here the structure of the database tables used by the database actions.



Note: These tables can be generated on an empty database from the **DBInit** action.

### Table: Tags

<b>FieldName</b>	Text(255)	PRIMARY KEY
<b>TagValue</b>	Text(255)	

### Table: Trends

<b>Id</b>	Long Integer	PRIMARY KEY
<b>TrendName</b>	Text(255)	
<b>SampleTime</b>	Text(255)	
<b>TrendValue</b>	Text(255)	
<b>Quality</b>	Text(255)	
<b>RefreshTime</b>	Text(255)	

## Table: Recipes

<b>Recipe</b>	Text(255)	PRIMARY KEY
<b>SetName</b>	Text(255)	PRIMARY KEY
<b>ElementName</b>	Text(255)	PRIMARY KEY
<b>SetValue</b>	Text(255)	

## Table: Event

<b>Id</b>	Long Integer	PRIMARY KEY
<b>EventName</b>	Text(255)	
<b>SampledTime</b>	Text(255)	
<b>EventType</b>	Text(255)	
<b>EventSubTime</b>	Text(255)	
<b>EventValue</b>	Text(255)	

# Custom tables

SQL queries released from the DB actions are listed inside the project file config\dbconnector.xml.

Modify the commands defined inside this file to customize the SQL strings released from the DB actions and then get access to a different structured database.

## Example

```
CREATE TABLE myTagsTable (tagname VARCHAR(255) PRIMARY KEY, tagvalue VARCHAR(255))
UPDATE myTagsTable SET Tagvalue= '%_JMV' WHERE Tagname= '%_JMT'
INSERT INTO myTagsTable (Tagname, Tagvalue) Values ('%_JMT', '%_JMV')
```

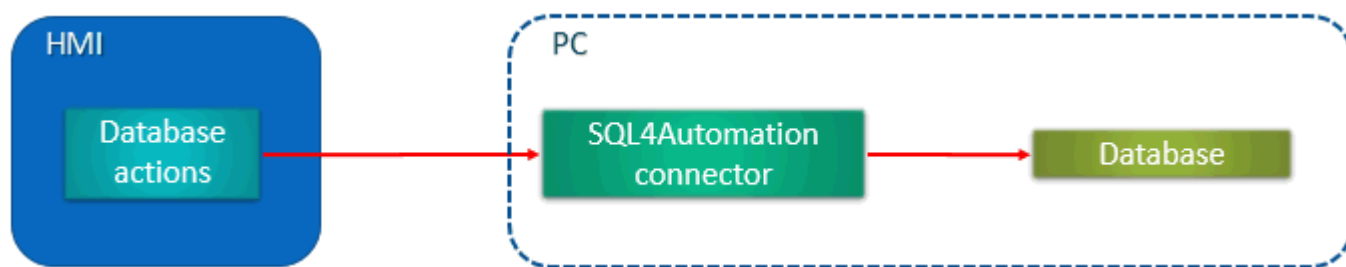
Where "%\_JMV" will be replaced with the tag value and "%\_JMT" with the tag name.

# Connection Limits

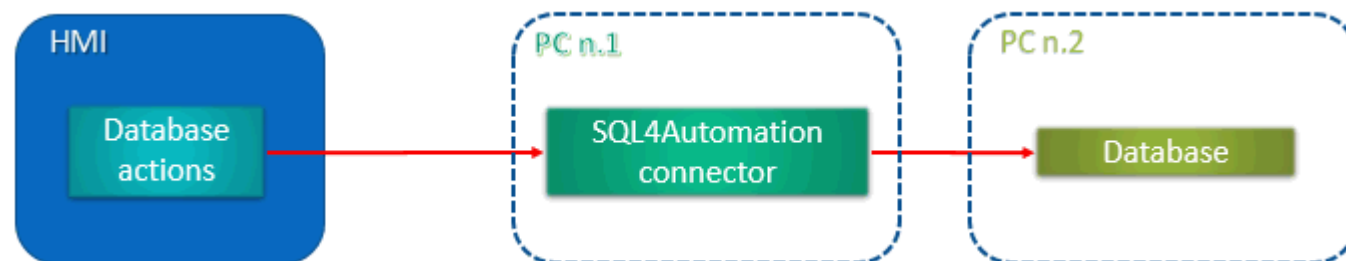
SQL4Automation is delivered as a USB dongle with a license for a predefined number of connections. Connections are called "Links" inside SQL4Automation Connector. The number of connections depends on the license you have purchased.

SQL4Automation Connector can be installed on the same Computer/Server running the database:





Or onto a separated Computer/Server:





## 34 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
<b>Enable OPC UA Server</b>	Main flag to activate OPC UA Server.  Data values defined in the HMI device are published by the OPC UA Server.

### Features

Parameter	Description
<b>Enable alarms</b>	Activates publication of real-time alarms data (active alarms).
<b>Enable historical alarms</b>	Activates publication of historical alarms data.
<b>Enable trends</b>	Activates publication of trends data.
<b>Tag groups</b>	Only tags belonging to selected groups will be available to the OPC UA Server.
<b>Alarm groups</b>	Only alarms belonging to selected groups will be available to the OPC UA Server.

### Network

Parameter	Description
<b>Node Name</b>	Enter node name or leave empty to use host name.
<b>Port</b>	Port number of OPC UA Server.   Port number proposed as default may be different from port used by OPC UA Client.


### Authentication

Select authentication options for OPC UA Server.



OPC UA Clients will be responsible for choosing, from available options, the most appropriate option to use according to their capabilities.

#### User authentication


Parameter	Description
<b>Anonymous</b>	Anonymous clients accepted.
<b>User/Password</b>	Authentication with user name is accepted.  Any valid user has unrestricted access to OPC UA Server (see <a href="#">"Configuring users" on page 317</a> ).

## Using x.509 Certificates


OPC UA provides a secure communication channel using digital certificates. Configurable levels of end-to-end security ensuring encryption, confidentiality and integrity of each message are available. When enabled, the server validates the client certificate and vice versa.



OPC UA Clients will be responsible for choosing, from available options, the most appropriate option to use according to their capabilities.

Security Mode	Description
<b>None</b>	Connection without certificate is allowed.  Not recommended in public networks.
<b>Sign</b>	OPC UA Client must provide its own certificate: communication through signed messages is allowed.
<b>SignAndEncrypt</b>	OPC UA Client must provide its own certificate: communication through signed and encrypted messages is allowed.

Security Policy	Description
<b>Basic128Rsa15</b>	Accepted encryption level (used only when Security Mode is active).
<b>Basic256</b>	Accepted encryption level (used only when Security Mode is active).
<b>Basic256Sha256</b>	Accepted encryption level (used only when Security Mode is active).

Parameter	Description
<b>Automatically trust any new clients</b>	All certificates provided from any OPC UA Clients are accepted.
<b>Trusted Certificates</b>	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
<b>Manufacturer name</b>	Human readable name of the manufacturer of the product.  OPC UA Client can retrieve this information from tag: <code>ServerName Objects Server ServerStatus BuildInfo ManufacturerName</code>
<b>Product name</b>	A human readable name for the product running in the server.  The OPC UA Client can retrieve this information from tag: <code>ServerName Objects Server ServerStatus BuildInfo ProductName</code>

## Certificate Parameters




Server certificate can be either generated automatically or by adding an existing certificate file.


### Automatically generate self-signed certificate

If auto generated certificate is enabled the certificate is regenerated after every change made by user to certificate parameters. The certificate is also replaced by any explicitly set certificate.

## Certificate parameters

Each certificate must contain information that should identify the certificate and its restrictions. If you have chosen to use a self-generated certificate, enter the information you want to be inside the self-generated certificate. Otherwise parameters are read from the certificate you have supplied.

Parameter	Description
<b>Server Name (Common Name)</b>	Name of the certificate (e.g. the device name).
<b>Organization</b>	Organization name
<b>Unit</b>	Organization unit  This field could be useful to differentiate different divisions within an organization.
<b>Location</b>	Locality field denotes the city where organization resides in
<b>State</b>	State or Province field specifies where the organization is physically located.  Content of State or Province field should not be abbreviated. For example, "CA" is not a valid state name. "California" is the proper state name.
<b>Country</b>	The X.509 naming scheme standard requires a 2-character country code.  Country code for the United States is US; country code for Italy is IT.

Parameter	Description
<b>Product URI</b>	A globally unique identifier for the server. Example: <i>"urn:NodeName:CompanyName:ServerName"</i>
<b>DNS Names IP Addresses</b>	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.  <b>The certificate will be valid only if the IP address where the OPC UA Server is running is included in this list.</b>
<b>Validity</b>	Period of validity of the certificate starting from creation date
<b>Key Length</b>	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=Italy/C=IT/L=Verona/O=CompanyName/OU=R&D Team/CN=OPCUAServer@%NodeName%

rem Creating Certificate (.pem)
echo subjectAltName=URI:urn:%NodeName%:CompanyName:OPCUAServer,IP:%IPAddress% >
san.txt
echo
keyUsage=digitalSignature,nonRepudiation,keyEncipherment,dataEncipherment,keyCertSign
>> san.txt
echo extendedKeyUsage=critical,serverAuth,clientAuth >> san.txt
echo authorityKeyIdentifier=keyid,issuer >> san.txt
echo basicConstraints=CA:TRUE >> san.txt
%OpenSSL% x509 -req -days 3650 -in server.csr -signkey server-key.pem -out
server.crt -extfile san.txt

rem Convert Certificate (.der)
%OpenSSL% x509 -in server.crt -outform der -out server.der

rem Not necessary files
del san.txt

pause
```

## Using self-signed certificates

This chapter is a step by step example that explains how to configure two HMI devices to communicate using self-signed certificates

### OPC UA Server

1. Create a simple project including a few tags
2. Open the OPC UA dialog and enable OPC UA Server. Be sure to enable tag groups (e.g. select “All”)
3. Enter in “IP addresses field” the IP address of the HMI device where OPC UA Server will run

☒ Automatically generate self signed certificate

Organization

Unit

Location

State

Country

DNS names

IP addresses

Validity  years

Key length

Certificate  ...

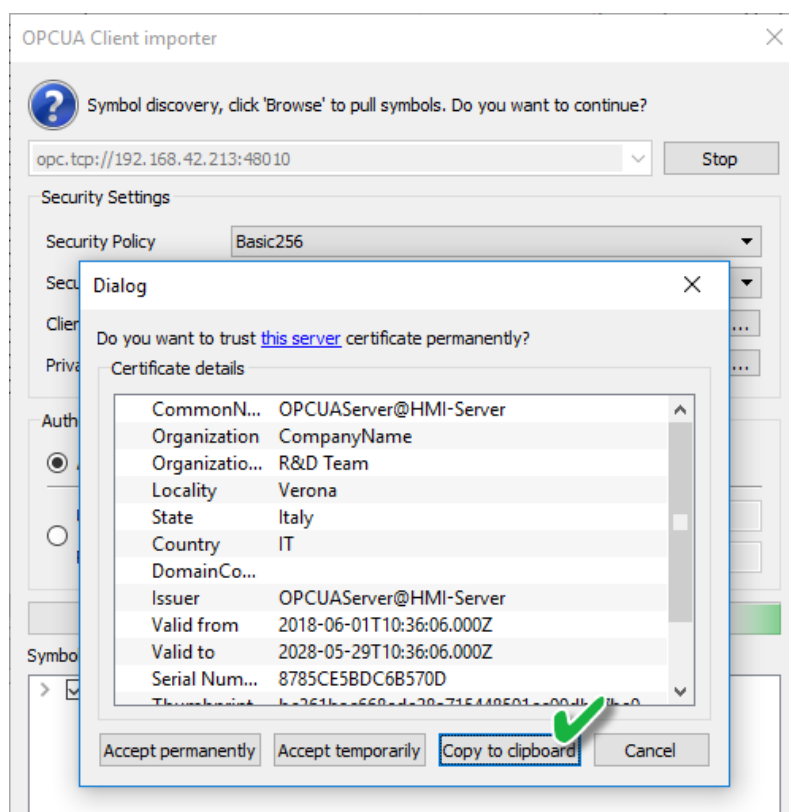
Private key  ...

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 certificate to clipboard as shown in figure. Paste it in the “Server Certificate” field of the protocol configuration dialog.



9. Repeat step 7, accept the Server OPC UA certificate, import some tags. 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 for later usage without asking you for confirmation.
10. Open again the protocol dialog box. Select the Security Policy = Basic256 and Security Mode = SignAndEncrypt
11. Download the project to the HMI device

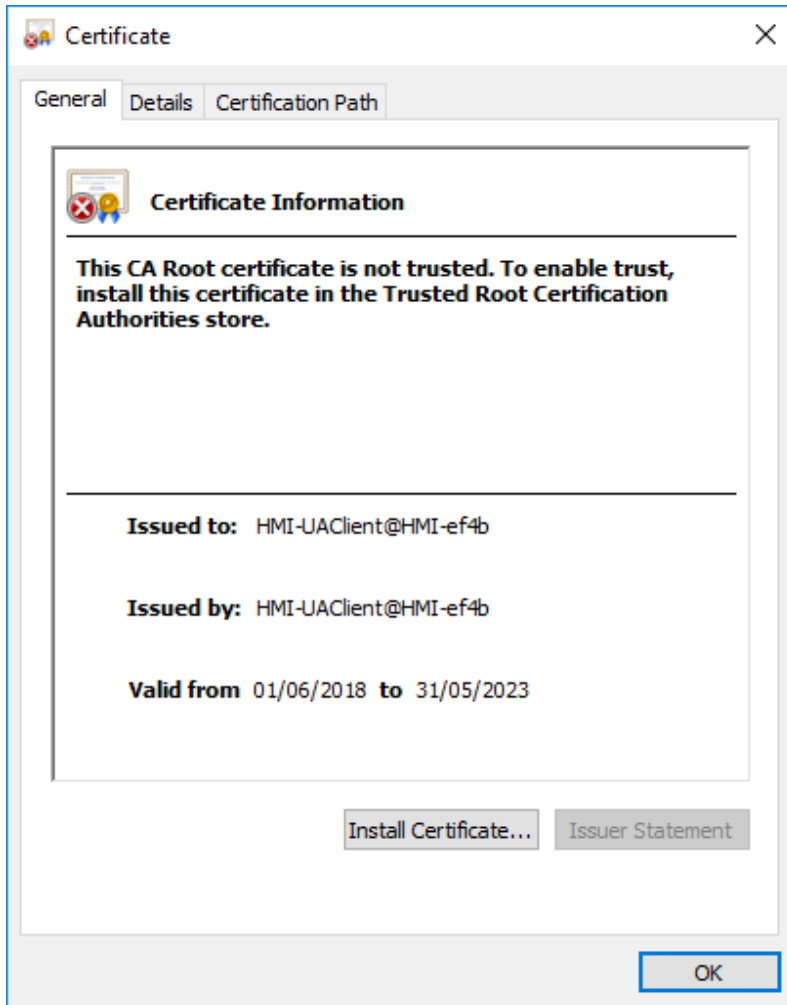
Since we left empty the “Server certificate” field in the OPC UA Client, the application has generated its own certificate and sent it to OPC UA Server but since the server does not know this certificate it rejects the connection request. Now we have to tell the server to trust these certificates. There are different ways to do it.

#### **Make rejected certificate trusted using FTP client:**

1. Connect to OPC UA device using an FTP client
2. Look inside the certificate folders and move the rejected certificate from the rejected folder to the trusted folder.

/workspace/<YourProjectName>/config/pkiserver/rejected  
 /workspace/<YourProjectName>/config/pkiserver/trusted/certs

You can double click the certificate file to open it and look to certificate parameters to be sure about the certificate you are validating



Now the communication will start

#### Make rejected certificate trusted using GDS tool:

1. Open the GDS tool and export its certificate
2. Open the project and add the certificate of the GDS tool to the Trusted Certificate list
3. Download the updates project to the HMI device

Now you can manage certificates using the tools in the HMI device.

## Using external certificates

This chapter is a step by step example explaining how to configure two HMI devices to communicate using external certificates.

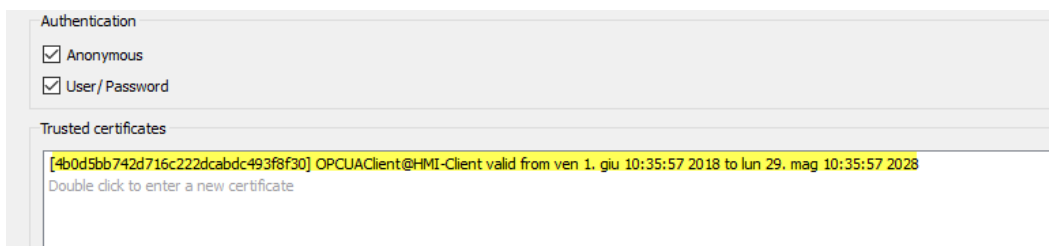
### Generate certificates

You can use the script given in this manual to generate a copy of your own certificates, one for OPC UA Server and another one for OPC UA Client.

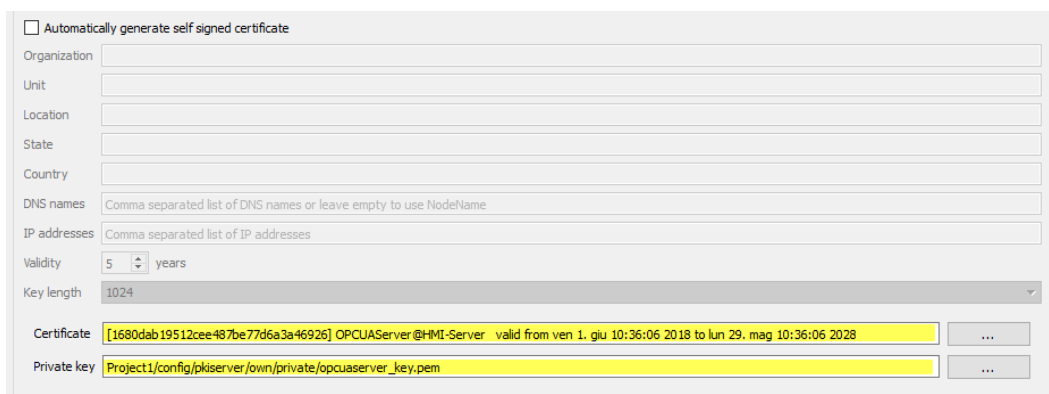
1. Install a OpenSSL-Win32 library (Reference.: <https://www.openssl.org/>)
2. Use the script ("[Script to generate a Certificate](#)" on page 378) to generate OPC UA Server certificate. Be sure to set the IPAddress variable with the IP Address of the HMI device where OPC UA Server will run before running the script.
3. Find in the OPC UA Client protocol manual a sample script to generate a certificate for the OPC UA Client protocol

### OPC UA Server

4. Create a simple project that using a few tags
5. Open the OPC UA dialog and enable OPC UA Server. Be sure to enable tag groups (e.g. select "All")
6. Add the client.der certificate to the Trusted Certificate area to enable the OPC UA Client to communicate with OPC UA Server



7. Remove the check on "*Automatically generate self-signed certificate*" and add the server certificate (server.der) and the server private key (server-key.pem)



8. Download the project to the HMI device


### OPC UA Client

9. Create a simple project
10. Add OPC UA Client protocol.
11. Enter the IP address of the remote OPC UA server and its port number (48010).
12. Open the ASCII version of the server certificate (server.crt), remove all Newline characters and then copy and paste the ASCII characters of your certificate inside the Server Certificate field.
13. Repeat the same with Client Certificate (client.crt) and Client private key (client-key.pem)
14. Select the Security Policy Basic256 and the Security Mode = SignAndEncrypt
15. Open tag editor and import tags. Select "OPC UA Discovery" mode
16. Accept OPC UA Server certificate, import some. Note that you can accept the certificate permanently or temporarily. If you accept the certificate permanently, a copy of the certificate is saved inside your computer to later usage without asking you for confirmation
17. Download the project to the HMI device

You will note that OPC UA Client is retrieving data from OPC UA Server using the given certificates.

## Alarm map

The alarm states are mapped to OPC UA states according to the following rules:

OPC UA Alarm state	JMobile Studio Alarm state
<b>Opcua.Alarm.Active</b>	True when alarm state is triggered <ul style="list-style-type: none"> <li>• TRIGGERED</li> <li>• TRIGGERED_NOT_ACKED</li> <li>• TRIGGERED_ACKED</li> </ul>
<b>Opcua.Alarm.Acked</b>	True when alarm acknowledgment is not required <ul style="list-style-type: none"> <li>• TRIGGERED_ACKED</li> <li>• NOT_TRIGGERED_ACKED</li> <li>• NOT_TRIGGERED</li> </ul>
<b>Opcua.Alarm.Retain</b>	True when alarm is pending <ul style="list-style-type: none"> <li>• TRIGGERED  </li> <li>• TRIGGERED_NOT_ACKED  </li> <li>• TRIGGERED_ACKED  </li> <li>• NOT_TRIGGERED_ACKED but a RESET is required</li> </ul>
<b>Opcua.Alarm.Confirmed</b>	True when alarm is returned (Not triggered, acknowledged and reset) <ul style="list-style-type: none"> <li>• NOT_TRIGGERED</li> </ul> <div style="display: flex; align-items: center;">  <p>This info is available only when alarm is configured to required a RESET</p> </div>

## 35 Special widgets

---

Widgets designed for special purposes are called special widgets and include control lists, date and time widgets, variable widgets and so on.

<b>BACnet widget</b> .....	<b>386</b>
<b>Browser widget</b> .....	<b>386</b>
<b>Canvas Widget</b> .....	<b>387</b>
<b>Combo Box widget</b> .....	<b>390</b>
<b>Consumption Meter widget</b> .....	<b>394</b>
<b>Control list widgets</b> .....	<b>395</b>
<b>DateTime widget</b> .....	<b>399</b>
<b>Gesture area widget</b> .....	<b>400</b>
<b>IP Camera widgets</b> .....	<b>405</b>
<b>Javascript function block widget</b> .....	<b>408</b>
<b>Media Player widgets</b> .....	<b>410</b>
<b>Multistate Image widget</b> .....	<b>412</b>
<b>Multistate Image Multilayer widget</b> .....	<b>413</b>
<b>Network Adapters widget</b> .....	<b>415</b>
<b>RSS Feed widget</b> .....	<b>415</b>
<b>Scrolling RSS Feed widget</b> .....	<b>416</b>
<b>Table widget</b> .....	<b>417</b>
<b>TextEditor widget</b> .....	<b>430</b>
<b>Variables widget</b> .....	<b>432</b>

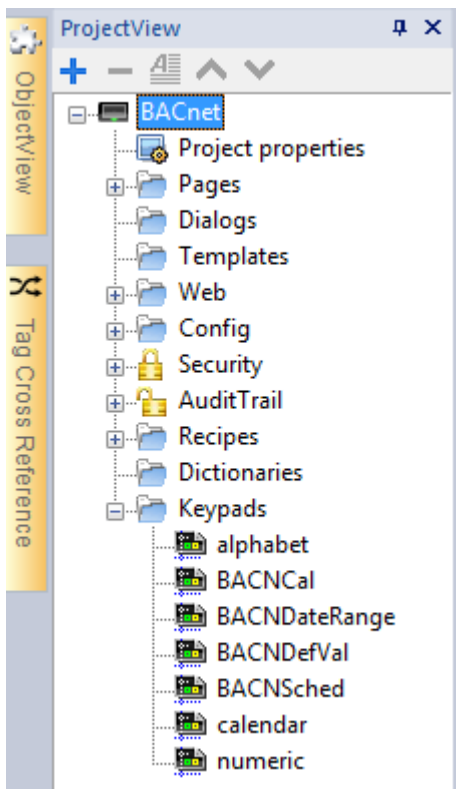
# BACnet widget

Path: **Widget Gallery**> **BACnet**

BACnet widgets are special widgets that let you interact with native BACnet objects.

- BACnet Calendar
- BACnet Scheduler
- BACnet Effective Period

These widgets are using special keypads that are added into the keypads folder when widgets are used. Generally, you do not need to take care of these keypads unless you want customized them.



Refer to the BACnet manual inside the “Communication Drivers” folder for a detailed description of BACnet special widgets.



**BACnet widgets are not supported on Web pages.**

# Browser widget

Path: **Widget Gallery**> **Media**> **Web Controls**

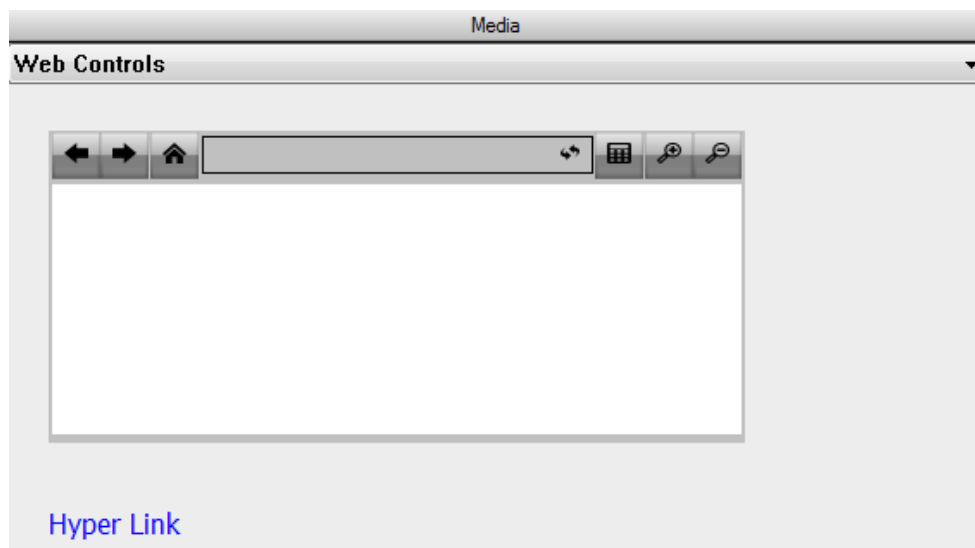
Use this widget to embed web pages into your HMI device pages. This is an HTML5 compatible browser widget based on the WebKit engine.



Note: The WebKit library is available as a plugin (see ["Plug-in" on page 77](#) for details) to download to the JMobile 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.**

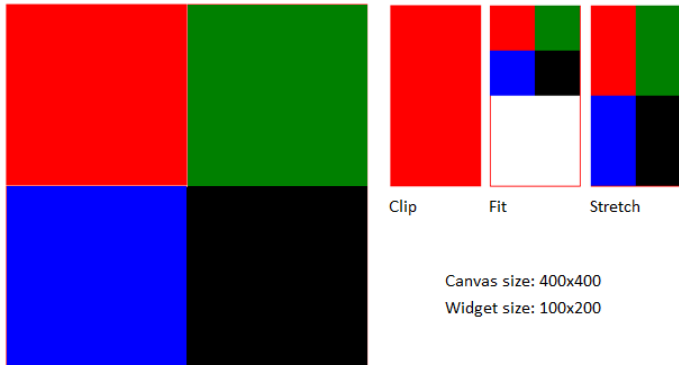

## Canvas Widget

Path: **Widget Gallery> Basic> Generic Canvas**

Canvas widget can be used to draw graphic via JavaScript scripting.



Note: the JavaScript methods are the same that are available for the HTML5 <canvas> tag

Parameter	Description
<b>Canvas Width</b> <b>Canvas Height</b>	<p>Canvas size.</p> <p>Note this is not the widget size. For example, the canvas size could be 500x500 pixels where the widget size could be 100x100 pixels. Draw Hint parameter will define how to stretch the canvas size to fit the widget size.</p>
<b>Draw Hint</b>	<p>Define how fit the canvas inside the widget size</p> <ul style="list-style-type: none"> <li>• <b>Clip</b> No Transformation is applied, coordinate system is not scaled and drawing is clipped inside the widget bounding rectangle.</li> <li>• <b>Fit to size</b> Fit to the widget size preserving the canvas model aspect ratio.</li> <li>• <b>Stretch</b> Fit to the widget size ignoring the canvas model aspect ratio.</li> </ul> <p>Example using a Canvas size larger than the widget size:</p>  <p>Canvas size: 400x400 Widget size: 100x200</p>
<b>Design Time Preview</b>	<p>Canvas preview inside JMobile Studio</p>  <p>Note the JavaScript code could use data not available inside JMobile Studio but only inside the HMI device</p>
<b>Auto Clear Background</b>	Automatic clear the background before draw canvas. When disabled, the painted items are persisted and is not necessary redraw everything from scratch.
<b>OnDraw Action</b>	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.
<b>OnMousePress Action</b> <b>OnMouseRelease Actions</b> <b>OnMouseDrag Actions</b>	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 QImageWgt using target and source rect)

- void drawImage(QObject \*pObjImage, qreal sx, qreal sy, qreal sw, qreal sh, qreal dx, qreal dy, qreal dw, qreal dh);
- void drawImage(QObject \*pObjImage, qreal dx, qreal dy);
- void drawImage(QObject \*pObjImage, qreal dx, qreal dy, qreal dw, qreal dh);
- void drawImage(const QVariant& image, int width, int height, const QString& format, qreal sx, qreal sy, qreal sw, qreal sh, qreal dx, qreal dy, qreal dw, qreal dh);

## // Pixel manipulation

- `ImageData createImageData(double sw, double sh);` //Empty Image
- `ImageData createImageData(ImageData fromImage);` //from another Image
- `ImageData createImageData(ArrayBuffer value);` //From arraybuffer
- `void putImageData(ImageData imgData, double dx, double dy);`
- `void putImageData(ImageData imagedata, double dx, double dy, double dirtyX, double dirtyY, double dirtyWidth, double dirtyHeight);`
- `ImageData getImageData(qreal sx, qreal sy, qreal sw, qreal sh);`

## Canvas JavaScript Example

The canvas is initially blank. To display something, a script first needs to access the rendering context and draw on it:

```
var ctx = me.context2d;
```

then you can use the canvas methods, as in the below example

```
function GenericCanvasWgt1_onDraw(me, eventInfo)
{
    var ctx = me.context2d;
    ctx.fillStyle = 'red';
    ctx.fillRect(0,0,250,250);
    ctx.fillStyle = 'green';
    ctx.fillRect(250,0,250,250);
    ctx.fillStyle = 'blue';
    ctx.fillRect(0,250,250,250);
    ctx.fillStyle = 'black';
    ctx.fillRect(250,250,250,250);
}

function GenericCanvasWgt1_onMouseDown(me, eventInfo)
{
    alert("X = " + eventInfo.posX + "\nY = " + eventInfo.posY );
}
```

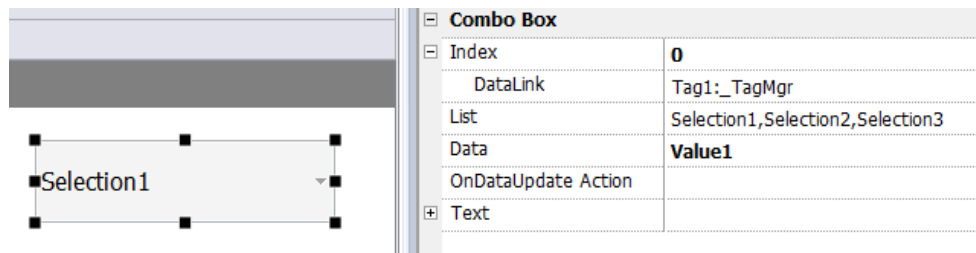
The update method can be used to dynamically redraw a canvas widget



```
function BtnStd1_btn_onMouseClicked(me, eventInfo)
{
    var myCanvasWidget = page.getWidget("GenericCanvasWgt1");
    myCanvasWidget.update();
}
```

## Combo Box widget

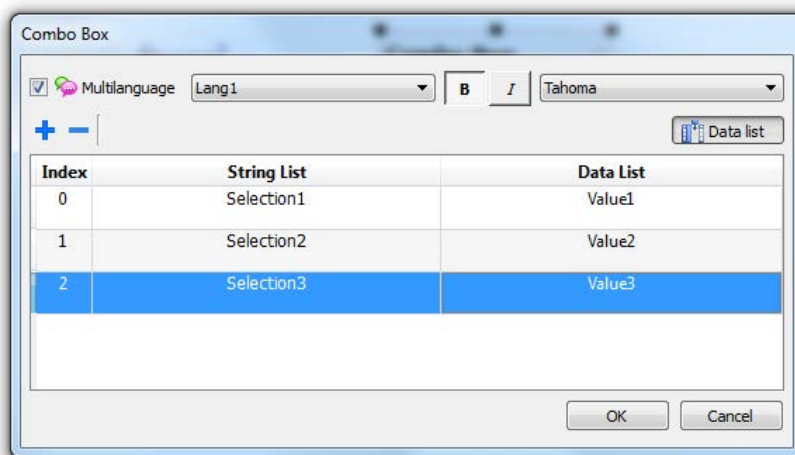
**Path: *Widget Gallery> Basic> Controls***

Use this widget as a selector widget or to filter rows in a table to display only the values selected in the combo box.



Parameter	Description
<b>Index</b>	Index of the selected item.
<b>List / String List</b>	Item strings in the combo box.  Note: This field is multi-language.
<b>Data / Data List</b>	Returns the value in the Data List column (as string) in the Data field of the widget.  Tip: Use this parameter to return a custom value based on an item selected in the combo box.
<b>Text</b>	Format of displayed text.

## Attaching data vs. attaching indexes



In many projects you may need to attach fields such as **Index** or **Data** to tags to know the values of the selected item in the combo box. Use:

- **Index:** to display the index (integer) of the selected item (0...n).
- **Data:** to display the data value (string) specified in the Data List column.

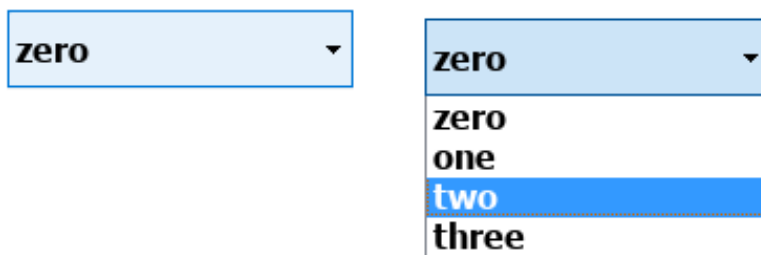
## Combo Box widget “full screen” mode with images

From the "Project properties" on page 71 the look and behavior of Combo Boxes can switch from Context mode to Full Screen mode

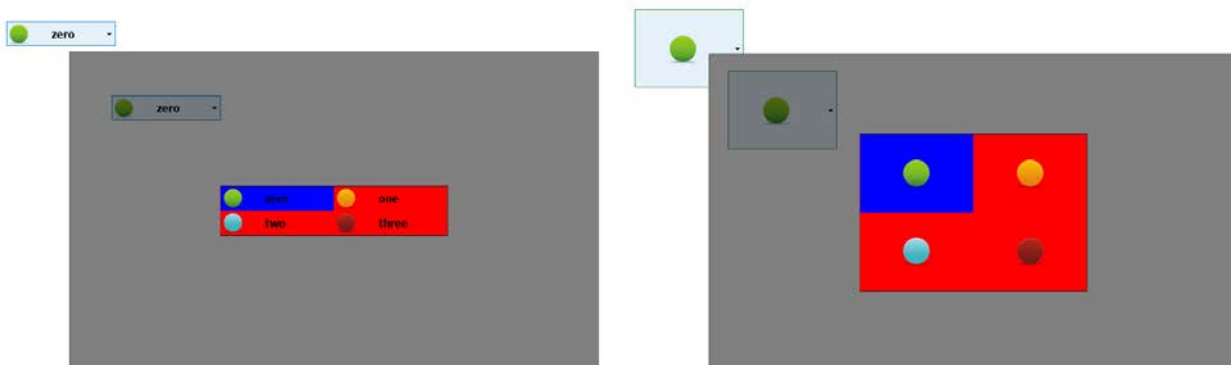
Path: **ProjectView**> double-click **Project properties**> **Properties pane**> **Project**> **ComboBox View Mode**

Parameter	Description
<b>ComboBox View Mode</b>	<p>Select the visualization mode of all the Combo Box widgets of the project</p> <p><b>Context</b> Classic view with drop-down menus</p> <p><b>Full screen</b> Enhanced view with configurable texts and images that will pop up in the middle of the screen for easy scroll and selection.</p>

**Context** view example

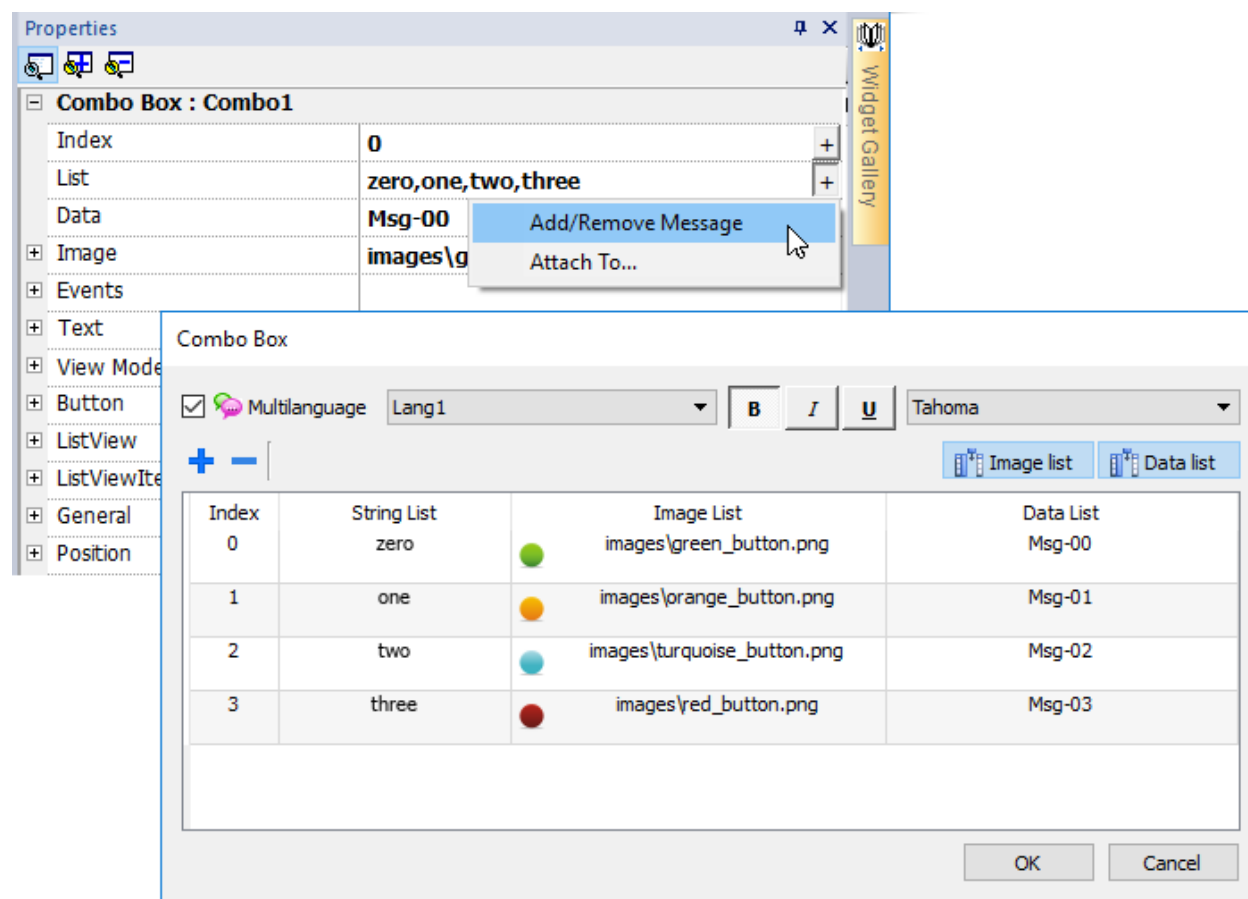


**Full screen** view example



## Additional parameters available in full screen mode

The additional "*Image List*" column will be available inside **Combo Box**> **List** parameter:



Note: Some properties are displayed only in advanced mode.

Parameter	Description
<b>Image</b>	Return, inside the attached tag, the file name of the selected image
<b>Button</b>	Define the look of the Combo Box <ul style="list-style-type: none"> <li>Show background = true Combo Box button is showed</li> <li>Show background = false Only image or text is showed</li> </ul>
<b>ListView</b>	Layout parameters of the Combo Box in edit mode
<b>ListViewItems</b>	Define the items type that will be inside the Combo Box <p>Image Mode:</p> <ul style="list-style-type: none"> <li>Only Text</li> <li>Only Images</li> <li>Text and Images</li> </ul>

# Consumption Meter widget

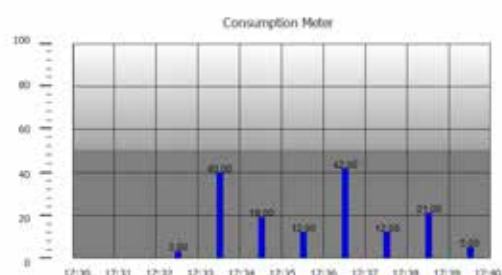
**Path:** *Widget Gallery > Basic > Trends/Graphs*

Use this widget to monitor a resource which is continuously increasing. The system reads the value of the resource and calculates the increment in a set range of time, the increment is then displayed in a bar-graph in a trend-like window.

Different colors can be used to used in the graph based on the time frame.



Tip: Use this widget to calculate the power consumption of a system.



Parameter	Description
<b>Value</b>	Resource monitored
<b>Graph Duration</b> <b>Graph Duration</b> <b>Duration Units</b>	Time period displayed in the window
<b>Bar Duration</b> <b>Bar Duration</b> <b>Units</b>	Time period represented by each bar in the graph
<b>Time Periods</b>	Assigns a specific color to highlight the increment of the monitored resource in a specified time period (minimum resolution = 1 hour).
<b>Color</b> <b>Bar Width</b>	Bar color and width
<b>Bar Value</b>	Show/Hide the value of each bar
<b>Consumption Meter</b>	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.



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.**

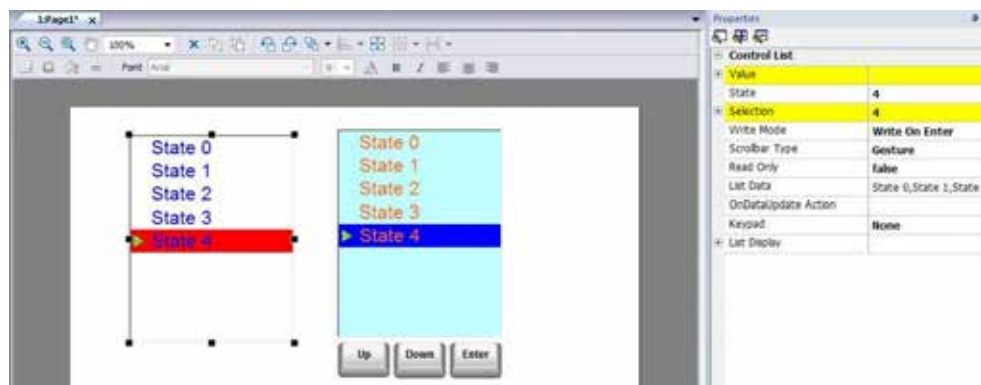
## Control list widgets

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.

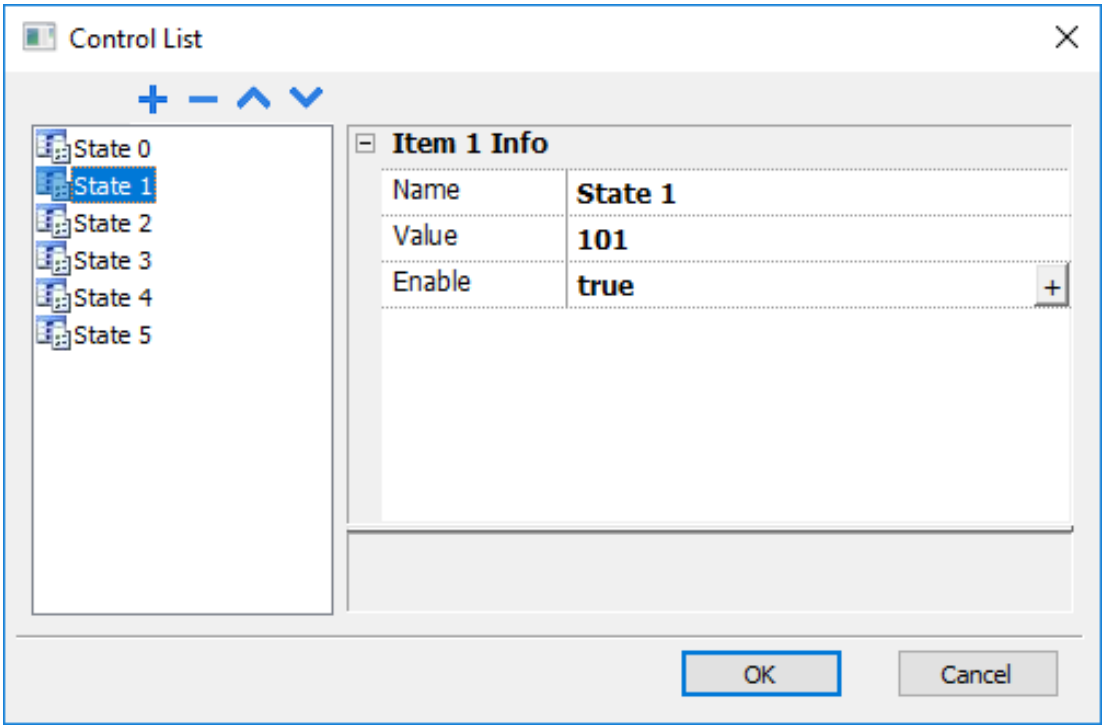
Two types of control lists are available:

- 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.



Parameter	Description
<b>Value</b>	The value corresponding to the status of the widget. If there is a tag attached to the value property, when loading the widget, the State will be aligned with the tag value.
<b>State</b>	State of widget. The widget highlights the item related to its State with a different background color (see "state color" in the properties of the widget).
<b>Selection</b>	State selection. The selected item will be displayed with a small triangle on the left side of the list.
<b>Write Mode</b>	Select the State update mode <ul style="list-style-type: none"> <li>• <b>Write On Select:</b> The state is updated automatically to be aligned with the cursor position.</li> <li>• <b>Write On Enter:</b> The status is updated with the cursor position only when the user presses enter</li> </ul>
<b>Scrollbars Type</b>	Select the scroll mode of the table <ul style="list-style-type: none"> <li>• <b>Gesture:</b> Pan gesture can be used to smoothly scroll the table.</li> <li>• <b>Scrollbar:</b> Use the scrollbar to scroll the table</li> </ul>

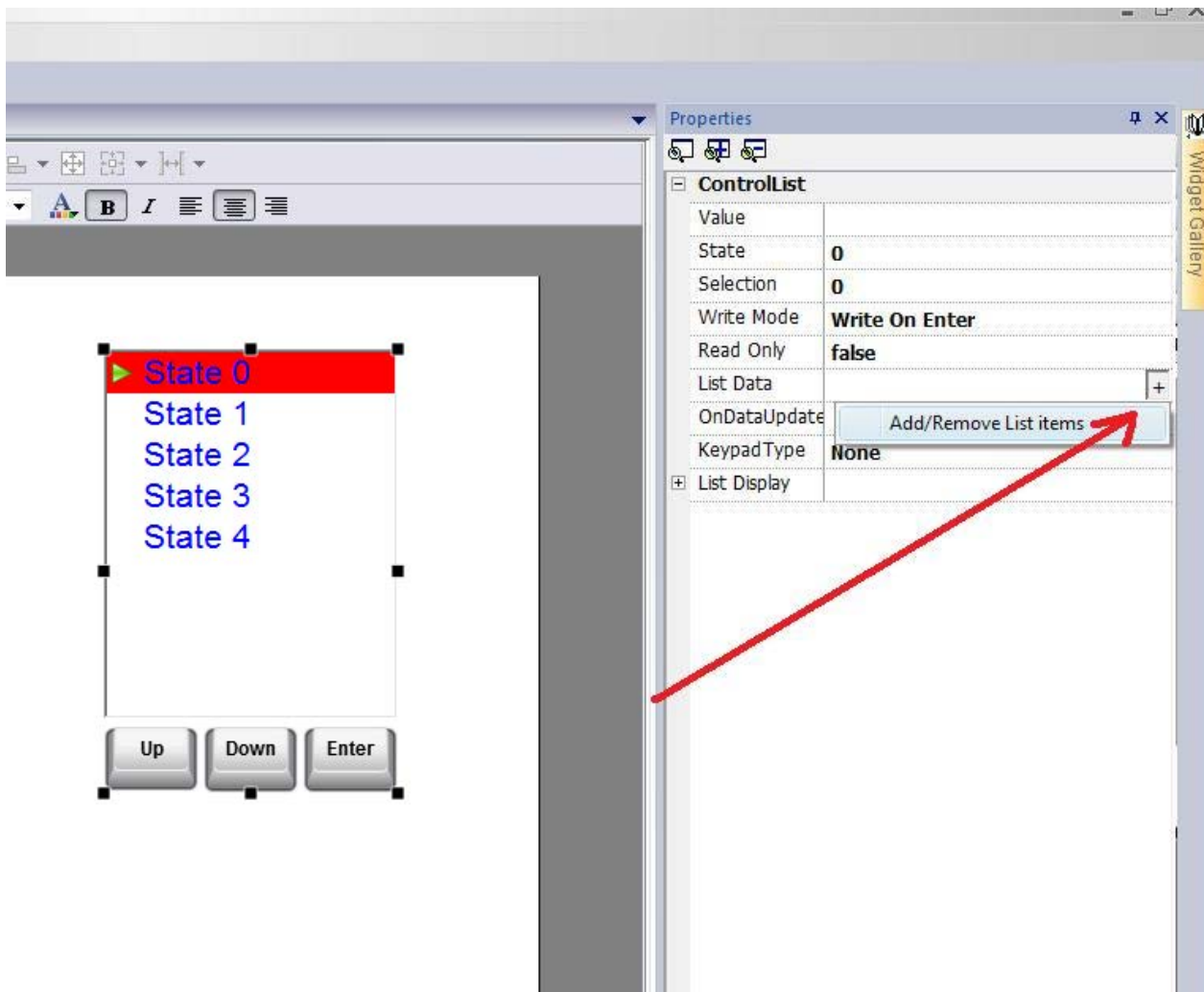


Parameter	Description
<b>Read Only</b>	Defines whether the list is only an indicator.
<b>List Data</b>	List of status items. Each item has a status name, a corresponding value and a flag that enables to display the item inside the widget. 

## Defining states

Add/remove states, that is items in the list, from the **List Data** property.

Any value can be assigned to a state. When you activate the state, by selecting the related item if in **WriteOnSelect** mode or selecting it and confirming with enter if **Write On Enter**, this will write the value assigned to state to the tag linked to the Control List widget **Value**.



## Manage list data items from JavaScript code

The list of data items can be modified, at runtime, from JavaScript code using the ***setProperty("listData", <NewControlList>)***. The below example shows how to modify the list of items

```
function SetItemsList_btn_onMouseClicked(me, eventInfo)
{
    var NewControlList = [{"OFF",100,true}, {"ON",101,true}, {"MAN",102,true},
    ["AUTO",103,true]};
    var ControlListWgt = page.getWidget("controlListBtn.controlList");
    ControlListWgt.setProperty("listData", NewControlList);
}
```

### Where

- *NewControlList* is an array with the items description
- *controlListBtn.controlList* is the ID of the Control List Widget to modify

The ***getProperty("listData")***, instead, will just return a comma separated string of just the names.

```
function Read_btn_onMouseClicked(me, eventInfo)
{
    var ControlListWgt = page.getWidget("controlListBtn.controlList");
    var ListData = ControlListWgt.getProperty("listData");
}
```

Where the result of ListData will be: "OFF,ON,MAN,AUTO"

### State

The **getProperty("state")** can be used to retrieve the State value. Here is an example of the JavaScript code

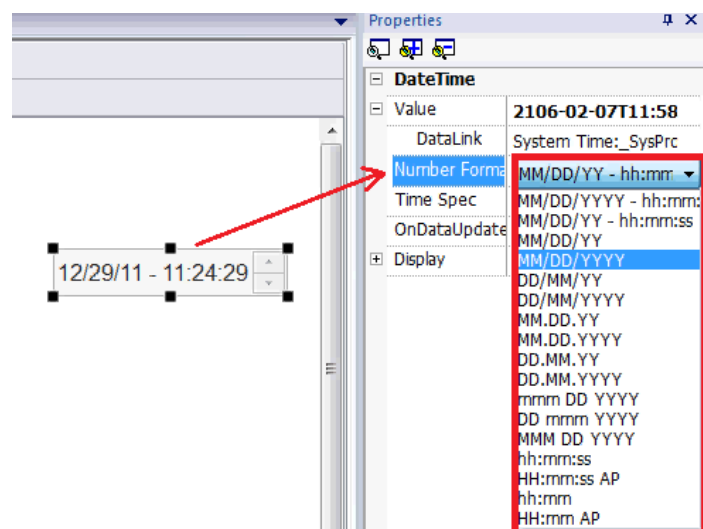
```
function controlListBtn_onDataUpdate(me, eventInfo)
{
    var ControlListWgt = page.getWidget("controlListBtn.controlList");
    var State = ControlListWgt.getProperty("state");
    project.setTag("State", State);
    return false;
}
```

## DateTime widget

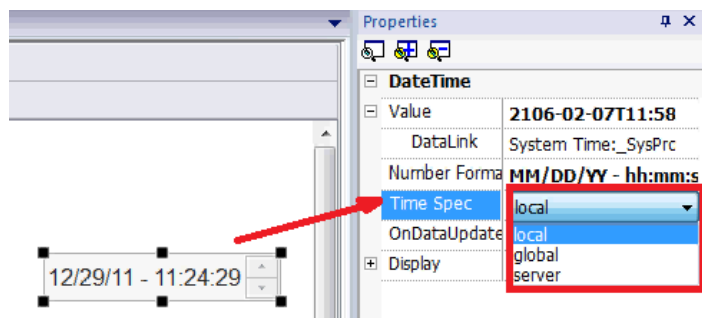
**Path:** *Widget Gallery* > *Basic* > *Controls*

Use this widget to display and edit current date and time .

In the **Properties** pane different formats are available for representing date and time.



For the **Time Spec** property select which time the widget will show at run time.



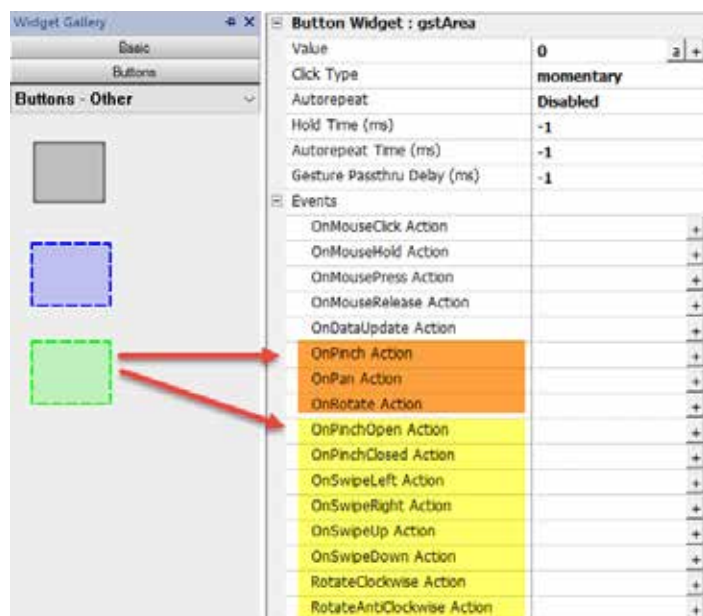
## Time options


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

## Gesture area widget

Path: **Widget Gallery**> **Buttons**> **Others**

Gesture Area Widget is a hotspot button that generates gesture events.



Gesture Events	Description
<b>OnSwipeLeft</b> <b>OnSwipeRight</b>	An event is release when swipe gesture is detected
<b>OnPinchOpen</b> <b>OnPinchClose</b>	An event is release when pinch gesture is detected
<b>RotateClockwise</b> <b>RotateAntiClockwise</b>	An event is release when rotate gesture is detected
<b>OnPan</b> <b>OnPinch</b> <b>OnRotate</b>	<p>A series of events released during the gesture.</p> <p>Only JavaScript can be used to service these events, through the JavaScript code the developer can manage the gestures events as he prefer.</p> <div>  <b>WARNING: Only multi touch HMI devices can generate OnPinch and OnRotate events</b> </div>

## OnPan

```
boolean onGesturePan(me, eventInfo)
```

This event occurs when one point inside the area has pressed and a linear movement has been detected.

Parameter	Description
<b>me</b>	Object triggering the event.
<b>eventInfo</b>	<p><b>id</b> = Gesture id; it is used to identify different gestures.</p> <p><b>running</b> = True except for last event delivered to notify gesture completion.</p> <p><b>dx</b> = Total X axis movement in screen pixel units from initial touch position .</p> <p><b>dy</b> = Total Y axis movement in screen pixel units from initial touch position.</p>

## OnPinch

```
boolean onGesturePinch(me, eventInfo)
```

This event occurs when two points inside the area have been pressed and a linear movement has been detected.

Parameter	Description
<b>me</b>	Object triggering the event
<b>eventInfo</b>	<p><b>id</b> = Gesture id; it is used to identify different gestures.</p> <p><b>running</b> = True except for last event delivered to notify gesture completion.</p> <p><b>dx</b> = Total X axis movement in screen pixel units from initial touch position. It represents the distance change between fingers. Positive value means that the</p>

Parameter	Description
	<p>distance is increasing; negative value means that the distance is decreasing. This amount may be used to control a zoom value.</p> <p><b>dy</b> = Total Y axis movement in screen pixel units (see dx).</p>

## OnRotate

```
boolean onGestureRotate(me, eventInfo)
```

This event occurs when two points inside the area have been pressed and a rotate movement has been detected.

Parameter	Description
<b>me</b>	Object triggering the event
<b>eventInfo</b>	<p><b>id</b> = Gesture id; it is used to identify different gestures.</p> <p><b>running</b> = True except for last event delivered to notify gesture completion.</p> <p><b>drot</b> = How many degrees (0/360) have been added since the previous event.</p> <p><b>trot</b> = Total degrees (0/360) of the entire movement.</p> <p>Positive numbers meaning clockwise rotation, negative anticlockwise rotation.</p>

## Gesture events pass thru

To use a widget (e.g. a button or a slider) covered from a gesture object, you have to keep pressed the widget 200 mSec to move the control to the underlying object. The time that must be waited for to send the command to the underlying object can be modified from the "Gesture Passthru Delay" parameter that is available in the advanced properties view.

Parameter	Description
<b>Gesture Passthru Enabled</b>	<p>Enable the possibility to pass gesture events to underlying widgets after a configurable delay. User has to keep pressed the finger and then execute the gesture.</p> <p><b>default</b> = Use the value defined in the project properties. See <a href="#">"Project" on page 78</a></p> <p><b>true</b> = Gesture passthru enabled</p> <p><b>false</b> = Gesture passthru disabled</p>
<b>Gesture Passthru Delay (ms)</b>	<p>The time that must be waited for to send the command to the underlying object</p> <p>0/500            mSec</p> <p>-1                Use the delay defined in the project properties. See <a href="#">"Project" on page 78</a></p>

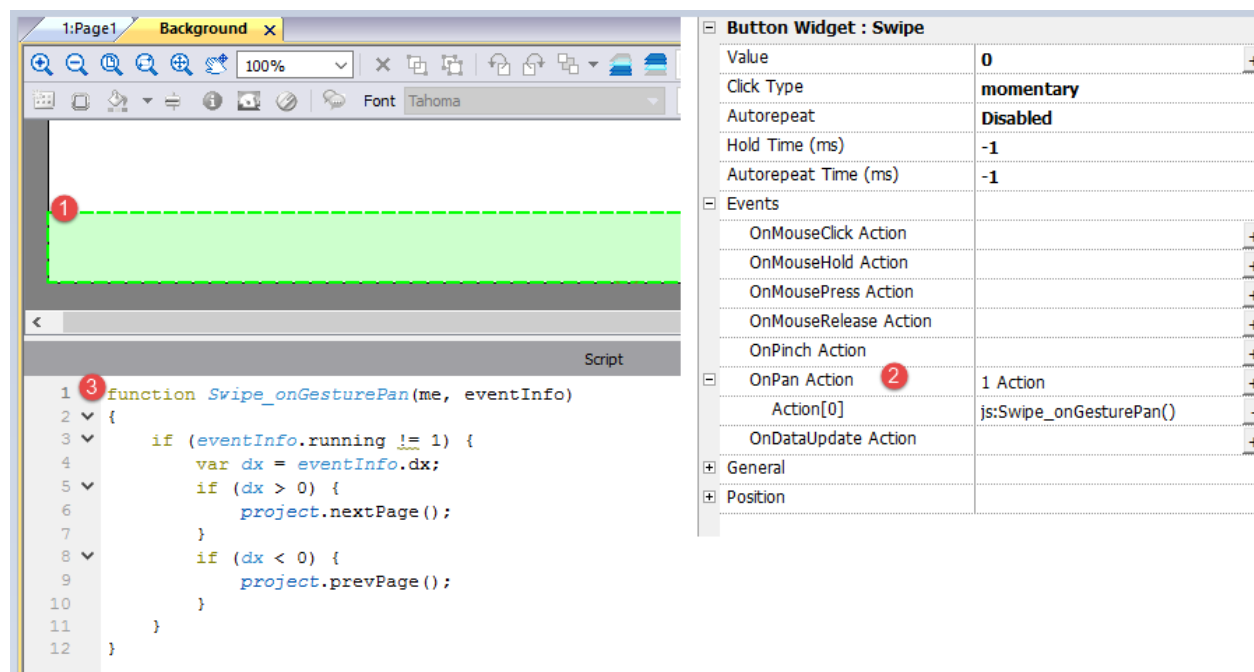
## Examples of using gesture events in association with JavaScript

Here some example of using gesture events in association with JavaScript code to identify gestures and program the requested actions

## Swipe Gesture

How to recognize a "swipe" gesture to change page in the application.

1. Put a Gesture area widget into the page
2. Configure the OnPan Action to trigger a JavaScript function
3. Write the JavaScript code that recognize and manage the swipe gesture



## Pinch Gesture

How to recognize a "pinch" gesture to resize an image.

1. Put a Gesture area widget into the page over the image
2. Configure the OnPinch Action to trigger a JavaScript function
3. Write the JavaScript code that recognize and manage the pinch gesture

The screenshot displays the JMobile Studio IDE with a canvas on the left showing a green area with a yellow light and a sign that says "UNDER CONSTRUCTION". A red box labeled '1' highlights the green area. On the right, the Properties panel shows the configuration for the "Button Widget : gstArea". The "Events" section is expanded, showing the "OnPinch Action" configured with "1 Action" and the action name "js:gstArea\_onGesturePinch()", which is highlighted with a red box labeled '2'. At the bottom, the JavaScript code for the widget is shown, with a red box labeled '3' highlighting the function definition.

**Properties Panel: Button Widget : gstArea**

Value	0
Click Type	momentary
Autorepeat	Disabled
Hold Time (ms)	-1
<b>Events</b>	
OnMouseClicked Action	+
OnMouseHold Action	+
OnMousePress Action	+
OnMouseRelease Action	+
OnPinch Action	1 Action
Action[0]	js:gstArea_onGesturePinch()
OnPan Action	+
OnDataUpdate Action	+

```

1  var pinchDX = 0;
2  var pinchDY = 0;
3  function gstArea_onGesturePinch(me, eventInfo)
4  {
5      if (eventInfo.running == 0) {
6          pinchDX = 0;
7          pinchDY = 0;
8      } else {
9          pinch_picture((eventInfo.dx-pinchDX), (eventInfo.dy-pinchDY));
10         pinchDX = eventInfo.dx;
11         pinchDY = eventInfo.dy;
12     }
13 };
14
15 function pinch_picture(dx, dy) {
16     var shape = page.getWidget("image1");
17     shape.x = shape.x - dx/2;
18     shape.y = shape.y - dy/2;
19     shape.width = shape.width + dx/2;
20     shape.height = shape.height + dy/2;
21 };
22

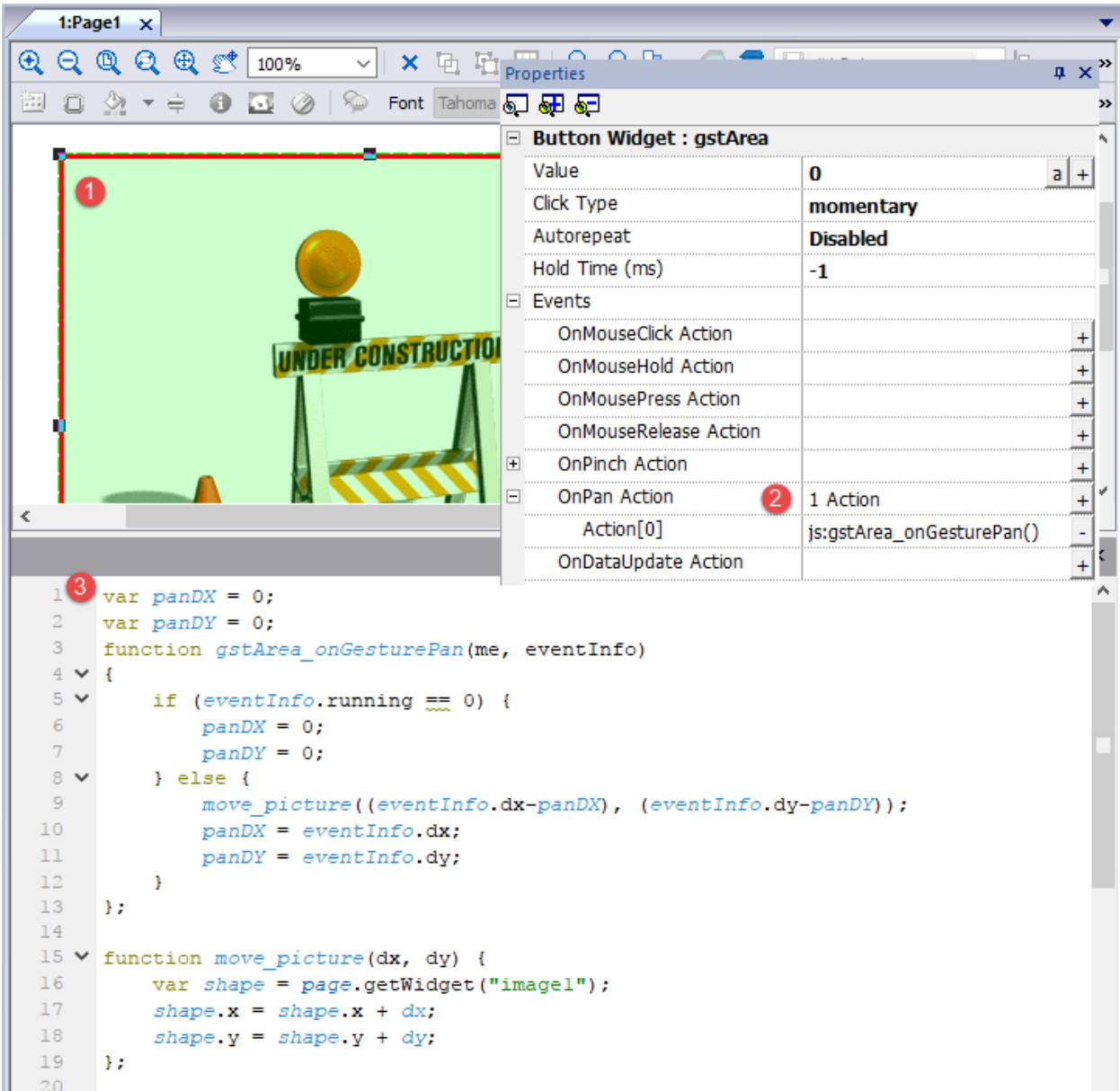
```

## Pan Gesture

How to recognize a "pan" gesture to move an image.

1. Put a Gesture area widget into the page over the image
2. Configure the OnPan Action to trigger a JavaScript function
3. Write the JavaScript code that recognize and manage the pan gesture





The screenshot shows the JMobile Studio interface. On the left, a widget is displayed with a green background and a yellow construction sign that says "UNDER CONSTRUCTION". A red dashed box highlights the widget, and a red circle with the number 1 is placed in the top-left corner. On the right, the Properties panel is open, showing the properties for the "Button Widget : gstArea". The properties include Value (0), Click Type (momentary), Autorepeat (Disabled), Hold Time (ms) (-1), and a list of Events. The "OnPan Action" is highlighted with a red circle and the number 2, and its value is set to "1 Action". The "Action[0]" is set to "js:gstArea\_onGesturePan()". Below the Properties panel, the code editor shows the implementation of the "gstArea\_onGesturePan" function, with a red circle and the number 3 highlighting the function definition.

```

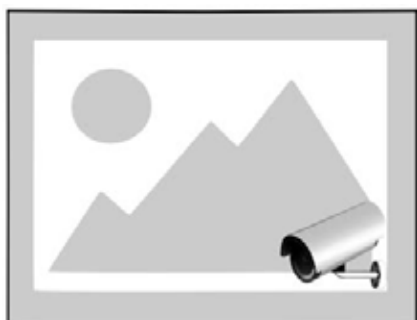
1 3 var panDX = 0;
2  var panDY = 0;
3  function gstArea_onGesturePan(me, eventInfo)
4  {
5      if (eventInfo.running == 0) {
6          panDX = 0;
7          panDY = 0;
8      } else {
9          move_picture((eventInfo.dx-panDX), (eventInfo.dy-panDY));
10         panDX = eventInfo.dx;
11         panDY = eventInfo.dy;
12     }
13 };
14
15 function move_picture(dx, dy) {
16     var shape = page.getWidget("image1");
17     shape.x = shape.x + dx;
18     shape.y = shape.y + dy;
19 };
20

```

## IP Camera widgets

Path: **Widget Gallery** > **Media** > **IP Camera**

Use these these widgets to show images captured from an IP Camera or a video stream.



Parameter	Description
<b>Camera URL</b>	URL of the IP Camera when used in JPEG format.
<b>Refresh Rate</b>	Number of JPEG images for second allowed. Max rate = 1 fps.
<b>User Name</b>	Name of user allowed to access the camera. Set this parameter when access to the camera is password protected.
<b>Password</b>	Password to access the camera.
<b>MJPEG Camera URL</b>	URL of MJPEG streaming (for example, http://192.168.0.1/video.cgi)

When this widget is used to stream HTTP MJPEG, **Camera URL** and **Refresh Rate** are ignored.

Performance of streaming is not fixed and depends on many factors such as: frame size, frame compression level, CPU of HMI device, quality of IPCamera. Based on these factors the widget can reach up to 25 fps.

You can add multiple IP Camera widgets, but this will reduce the frame rate for each widget.

## Supported IPCameras

The following IP Cameras have been tested so far:

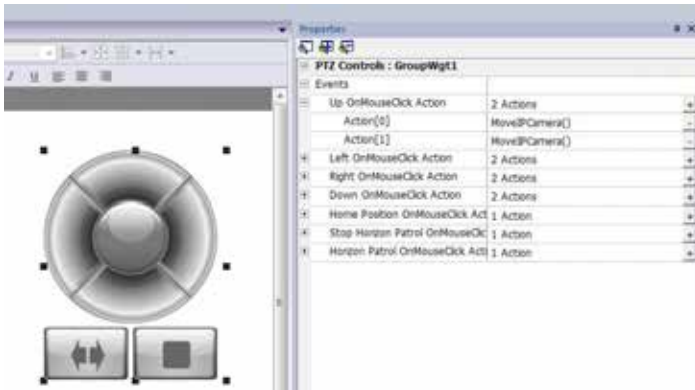
IPCamera	Protocol	URL
<b>Apexis APM-J901-Z-WS PTZ IP Camera</b>	MJPEG	http://{ip_address}/videostream.cgi
	HTTP	http://{ip_address}/snapshot.cgi
<b>AXIS M3027-PVE Network Camera</b>	MJPEG	http://{ip_address}/axis-cgi/mjpg/video.cgi
	HTTP	http://{ip_address}/axis-cgi/jpg/image.cgi
<b>DAHUA DH-IPC-HD2100P-080B 1.3mp Outdoor Vandalproof</b>	HTTP	http://{ip_address}:9988/onvif/media_service/snapshot
<b>D-Link DCS-5605 PTZ</b>	MJPEG	http://{ip_address}/video/mjpg.cgi
<b>D-Link DCS-900W IP Camera</b>	MJPEG	http://{ip_address}/video.cgi
<b>D-Link DCS-932L</b>	MJPEG	http://{ip_address}/video.cgi
<b>Edimax IC-7100P PTZ</b>	MJPEG	http://{ip_address}/mjpg/video.mjpg
	HTTP	http://{ip_address}/picture.jpg
<b>Foscam FI8916W</b>	MJPEG	http://{ip_address}/videostream.cgi
	HTTP	http://{ip_address}/snapshot.cgi
<b>Foscam FI9803 EP</b>	MJPEG	http://{ip_address}:88/cgi-bin/CGIStream.cgi?cmd=GetMJStream&usr={user}&pwd={pass}

IPCamera	Protocol	URL
		NOTE: <ul style="list-style-type: none"> <li>port 88 may be different as per IP Camera settings</li> <li>{user} = username defined into IP Camera settings</li> <li>{pass} = password defined into IP Camera settings</li> </ul>
<b>Hamlet HNIPCAM IP Camera</b>	MJPEG HTTP	http://{ip_address}/video.cgi http://{ip_address}/image.jpg
<b>MOXA VPort 254</b> (Rugged 4-channel MJPEG/MPEG4 industrial video encoder)	MJPEG HTTP	http://{ip_address}/moxa-cgi/mjpeg.cgi http://{ip_address}/moxa-cgi/getSnapShot.cgi?chindex=1
<b>NVS30 network video server</b>	MJPEG HTTP	http://{ip_address}:8070/video.mjpeg http://{ip_address}/jpg/image.jpg
<b>Panasonic WV-Series Network Camera</b>	MJPEG	http://{ip_address}/cgi-bin/mjpeg
<b>Ubiquiti UniFi Video Camera</b>	HTTP	http://{ip_address}:7080/images/snapshot/camera/{camera_guid}?force=true NOTE: <ul style="list-style-type: none"> <li>{camera_guid} can be found into IP Camera Webpage</li> <li>port 7080 may be different as per IP Camera settings</li> </ul>
<b>Zavio F3210 2MP Day &amp; Night Compact IP Came</b>	MJPEG HTTP	http://{ip_address}/stream?uri=video.pro3 http://{ip_address}/cgi-bin/view/image?pro_0 NOTE: <ul style="list-style-type: none"> <li>MJPEG video streaming can be configured selecting "video profile 3" with 640x480 resolution into IP Camera settings.</li> </ul>

## PTZ Controls widget

PTZ (pan-tilt-zoom) cameras are cameras capable of remote directional and zoom control.

The PTZ Controls widget uses the MoveIPCamera action to send HTTP/cgi commands to the PTZ IP Camera.



Parameter	Description
<b>Camera URL</b>	URL of IP Camera
<b>User Name</b>	Name of user allowed to access the camera. Set this parameter when access to the camera is password protected.
<b>Password</b>	Password to access the camera.
<b>Command</b>	Command to send to the PTZ controller (for example, decoder_control.cgi?command=0)

## 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

# 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.

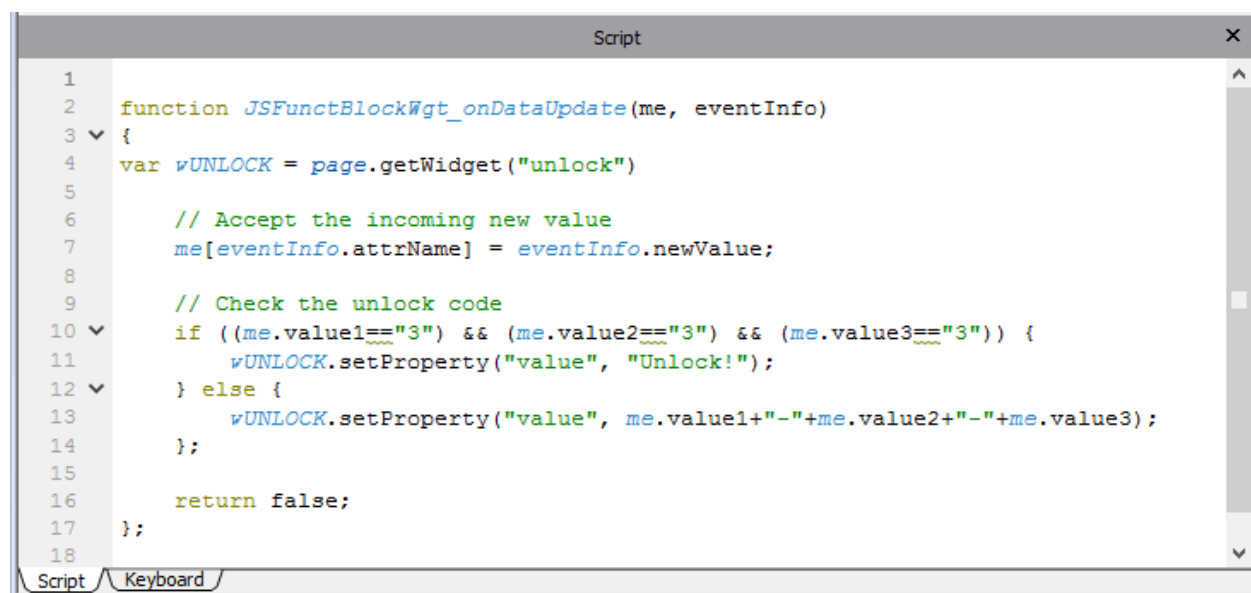
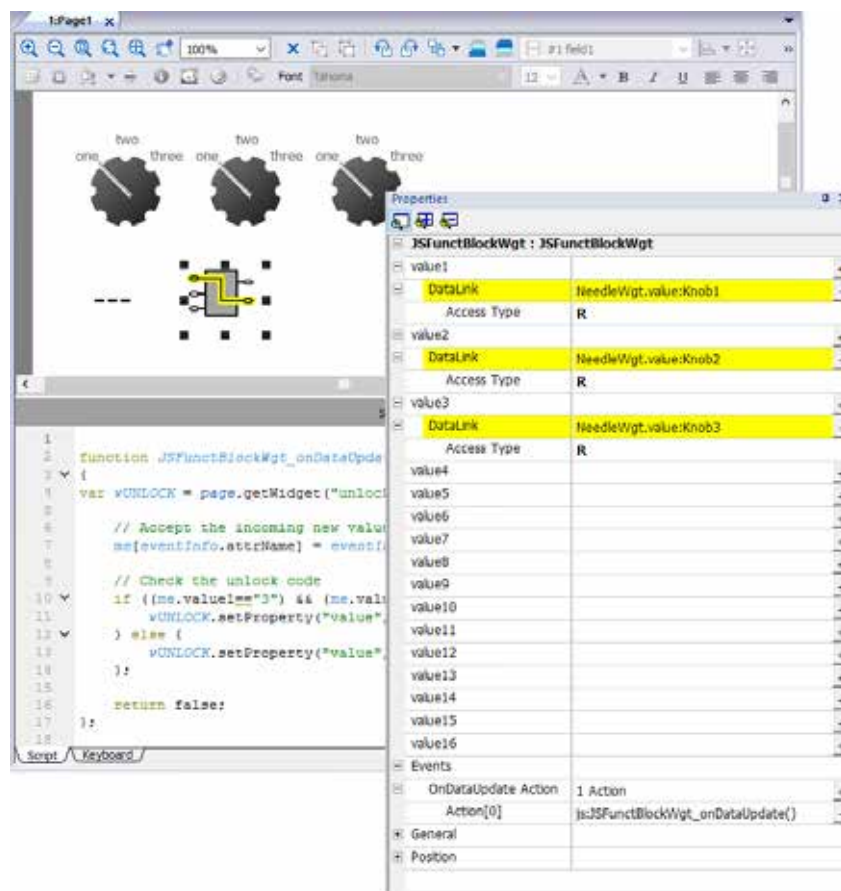
Parameter	Description
<b>value1</b> ... <b>value16</b>	Objects that will trigger the OnDataUpdate action.
<b>OnDataUpdate</b>	Action that will be executed when a change of an associated value is detected



Note: This widget is rendered only in JMobile Studio, and it is not rendered in the HMI device.

Example:

A Javascript code that check the combination lock of three selectors



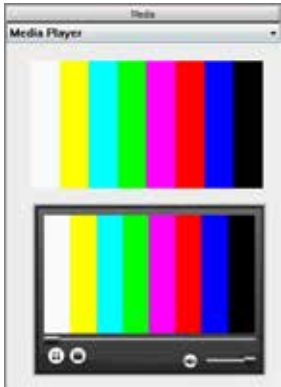
See "Widget events" on page 458 for the description of the onDataUpdate parameters

# Media Player widgets

**Path:** *Widget Gallery > Media > Media Player*

Use these widgets to play videos from a playlist. The video files can be stored on a USB drive, on the Flash card or an SD Card.

Two widgets are available: one includes a multimedia frame with buttons to play and stop the video, the other is a plain frame where the video is played without user control.



Parameter	Description
<b>Media Player List</b>	Open file browser of Windows to select the video files to collect into the play list. The selected files will be downloaded inside the HMI device with the project
<b>Loop Style</b>	<p>Define how the video is played.</p> <ul style="list-style-type: none"> <li>• <b>NoLoop</b>: plays all the videos in the playlist, then stops.</li> <li>• <b>LoopOne</b>: repeats the first video in the playlist.</li> <li>• <b>LoopAll</b>: repeats the entire playlist.</li> <li>• <b>Random</b>: plays the videos in a random order.</li> </ul>



Note: The Media Player widget only works with some HMI devices (see "[HMI devices capabilities](#)" on page 567). It doesn't work the HMI Client.



Note: You can have only one Media Player widget in a page.

## Supported video encoding

Two groups of codecs are supported:

- DSP based video codecs
- Software video codecs



List of HMI devices that support the DSP (video hardware acceleration) is available on "[HMI devices capabilities](#)" on page 567.

## DSP video codecs

These include:

- H264 using AVI/MP4 container, CABAC off and Level 3 (suggested)
- MPEG4 using MP4 container



*On WinCE devices BSP v1.55 or greater is required*  
*On Linux devices BSP v1.0.269 or greater is required*

## Software video codecs

This is only:

- Microsoft MPEG4 v3 using an AVI container.

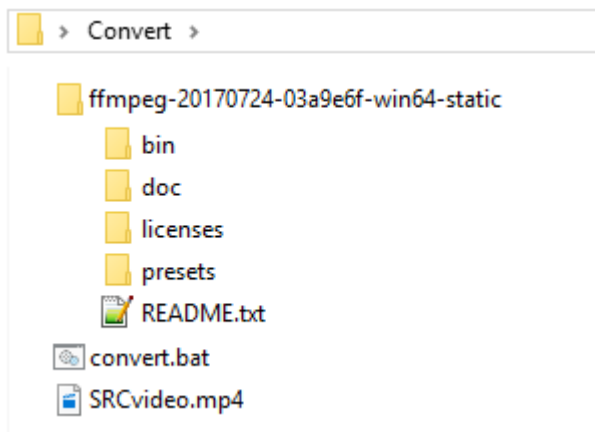


Be aware that video performance are depending from the chosen resolution, bit rate and device capabilities. If video rendering is not smooth, try to reduce the resolution or the bit rate of your video.

The videos encoded with Microsoft MPEG4 v3 are not using the hardware acceleration and have more limitations. To prevent the videos from running jerky, a maximum resolution of 640x512 pixels and a bit rate of 1300 kb/s are suggested. In addition, the size of the Media Player widget used on the page should have the same size as the videos in the play list, in order to avoid up scaling and down scaling. Audio is not supported.

## Converting a video

The FFMPEG ([www.ffmpeg.org](http://www.ffmpeg.org)) can be used to convert a video into the correct codec supported from the HMI device. Using the folder structure of the below picture, the following batch file could be used to convert any video file.



```

convert.bat
1 @echo off
2 set FFMPEG=ffmpeg-20170724-03a9e6f-win64-static\bin\ffmpeg.exe
3
4 %FFMPEG% -i SRCvideo.mp4 ^
5         -y ^
6         -an ^
7         -s 240x160 ^
8         -b:v 4200k ^
9         -maxrate 4200k ^
10        -c:v libx264 ^
11        -profile:v baseline ^
12        -level:v 3 ^
13        -bufsize 3000k ^
14        -minrate 0 ^
15        -f avi ^
16        -preset slow ^
17        HMIvideo.avi
18
19 pause
20

```

Now you can open the converted video with a standard video player, such as Windows Media Player and check the quality. You can add the resulting video to the play list of the Media Player widget.



Note : The FFMPEG tool is not distributed with the JMobile Suite.

## 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';

```

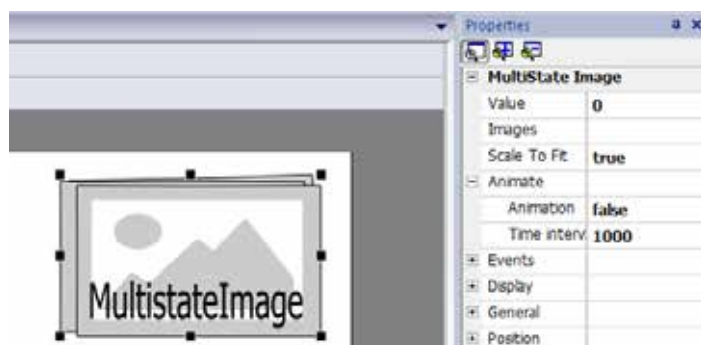
See "JavaScript " on page 453 for details on how to work with JavaScript.

## Multistate Image widget

*Path: Widget Gallery> Basic> Images*



Use this widget to display an image from a collection based on the value of a tag used as Index. You can use this widget also for simple animations.

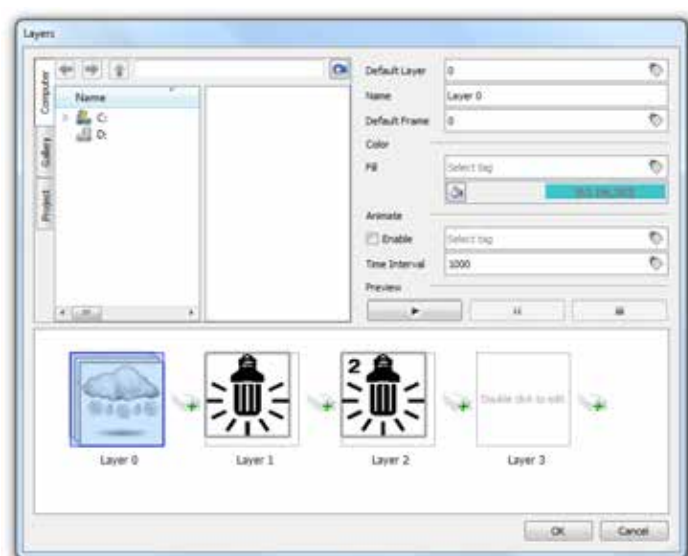


Parameter	Description
<b>Value</b>	Index of image to display. For example, set Value=0, to display the image with index 0 in the image collection.
<b>Images</b>	Images collection with associated index.
<b>Animate</b>	Set to true, to enable a slide show.
<b>Time interval</b>	Interval between images in the slide show.

## Multistate Image Multilayer widget

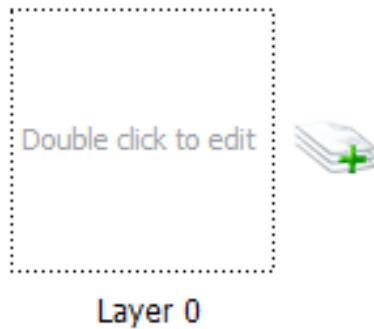
**Path:** *Widget Gallery> Basic> Images*

Use this widget to create different animations and select the most suitable at run time.

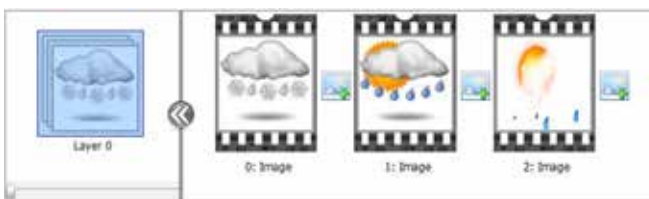


## Setting up widget layers

1. Open the **Layers** dialog from the **Properties** pane.
2. Click **+** to add as many layers as you need.



3. Double click on each layer to add as many images as you want to include in the layer.



4. Drag and drop images into the frame to add it to current layer.



5. Define widget properties.

Parameter	Description
<b>Default Layer</b>	Layer shown at run time.
<b>Name</b>	Name of selected layer.
<b>Default Frame</b>	Frame shown when current layer is displayed.
<b>Color / Fill</b>	Fill color for images of current layer.
<b>Animate</b>	Enables slide show for active layer. Animations can be started/stopped at run time attaching it to a tag.
<b>Time Interval</b>	Time interval of slide show, if enabled.
<b>Preview</b>	Slide show simulation.



Note: **Default Layer**, **Default Frame**, **Color** and **Fill** can be changed at run time, attaching the to a tag.

## Network Adapters widget

**Path:** *Widget Gallery > Basic > Control*

Use the IP Widget to set the network adapters parameters.

The dialog box titled "Network Adapter Parameters" contains the following fields and controls:

- A dropdown menu for the network adapter, currently showing "LAN16".
- A "Mac ID:" label followed by the text "00:50:56:C0:00:08".
- A "Use DHCP:" label followed by a dropdown menu set to "Yes".
- An "IP Address:" label followed by a text box containing "172.16.239.1".
- A "Subnet Mask:" label followed by a text box containing "255.255.255.0".
- A "Gateway:" label followed by a text box containing "0.0.0.0".
- "Cancel" and "Apply" buttons at the bottom.

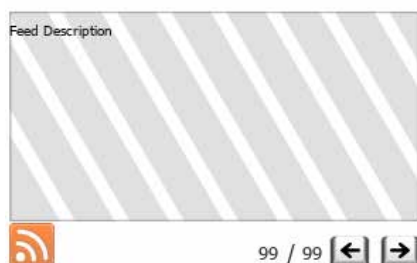
The system variable Network->Status contains the result of the last operation performed by the IP Widget (see "[Network variables](#)" on page 105 for details)


## RSS Feed widget

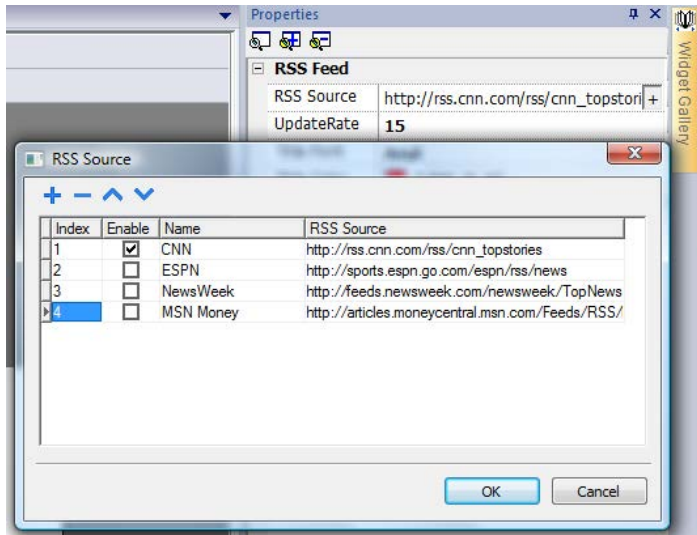
**Path:** *Widget Gallery > Media > RSSFeed Source*

Use this widget to display on the HMI device your favorite RSS feeds directly from the Internet.

RSSFeed



Parameter	Description
<b>RSS Source</b>	Feed URL  Note: Feed sources cannot be modified at run time.
<b>UpdateRate</b>	Refresh time



The RSS Feed widget has been specifically designed to work with Pocket Internet Explorer.

## Scrolling RSS Feed widget

**Path:** *Widget Gallery > Media > RSSFeed Scroll*

Use this version of the main RSS Feed widget to display highlights inside a text line using a smoothing scrolling text.

RSSFeed Scroll



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
<b>Scrolling</b>	Scrolling speed
<b>Title Separator</b>	Separator character between highlights

# 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:

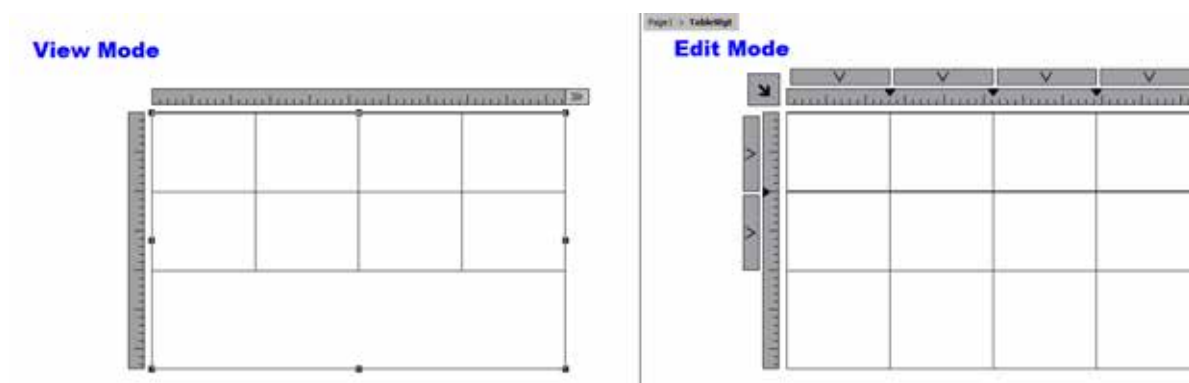
1. Put a table widget on the screen and configure the template of the table.
2. Add widgets into cells to configure one or more rows that will be used as row templates when the table will be filled with data provided from the data source.
3. Select a data source that will be used to fill the rows of 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

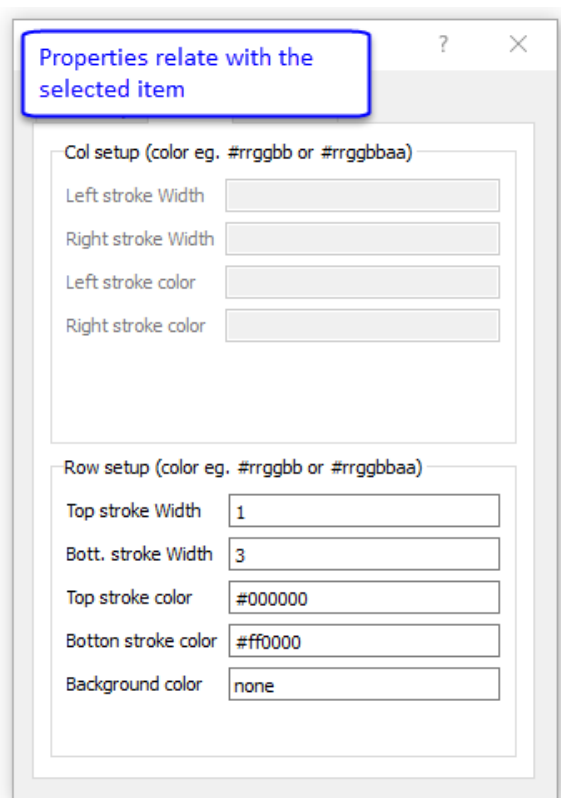
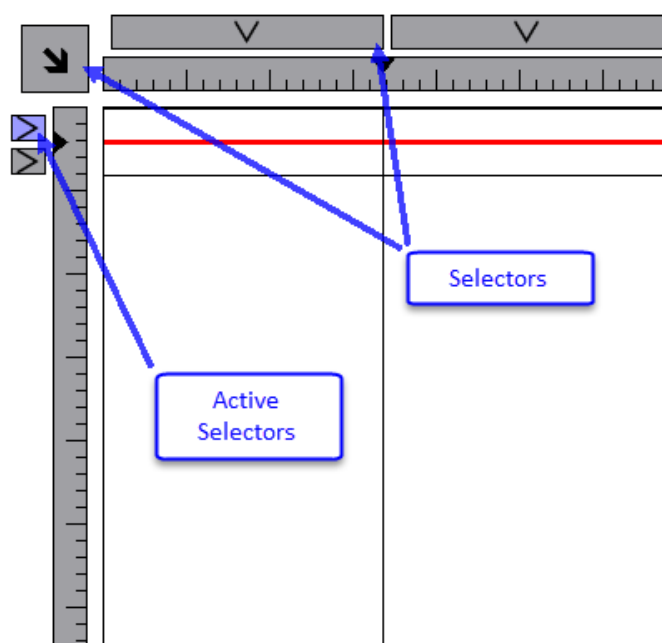
In view mode, you can configure the table layout. Drag and drop the table onto the page, resize the table, define number of template rows, number of columns and the main table properties.

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

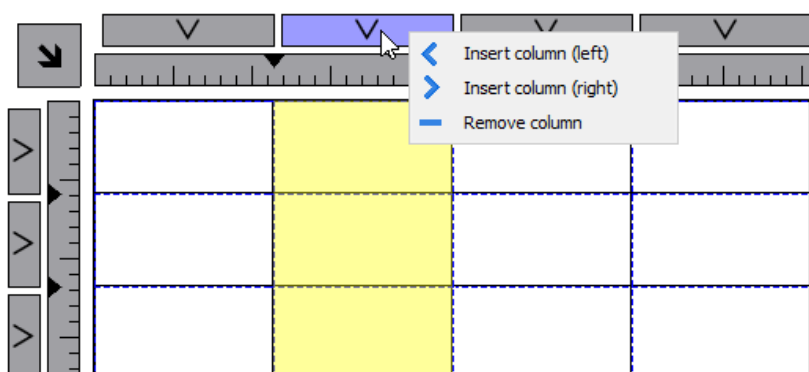
In edit mode, it is possible configure the format and the content of each cell of the table. Each row of the table will act as a row template.

To configure the look of the table, click on table's selectors to select the item to configure.



## 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

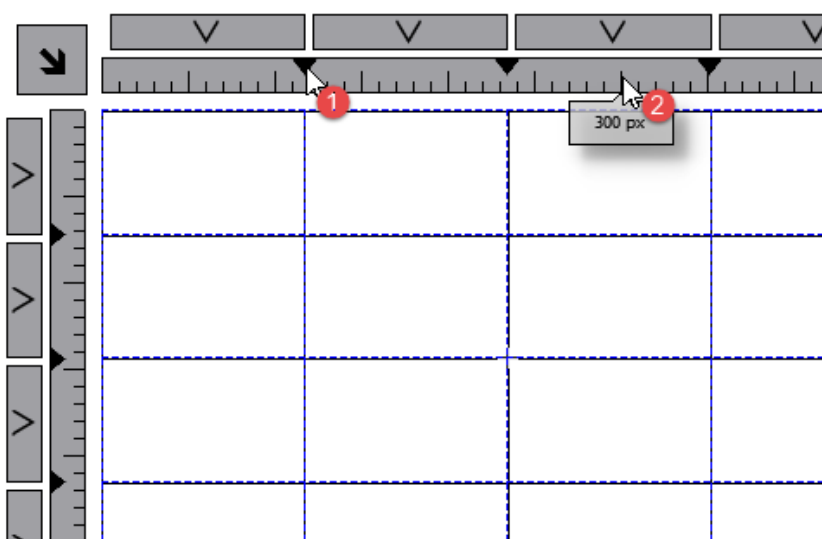
To merge or split rows or columns, double click over the grid to enter in edit mode and move the cursor over the ribbons:

- Double click the black triangle to merge the two adjacent rows or columns (1)

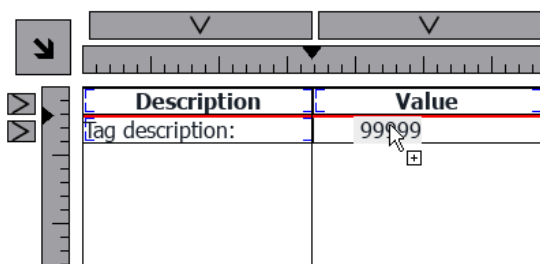


Note that merge is possible only with an empty row or column.

- 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.

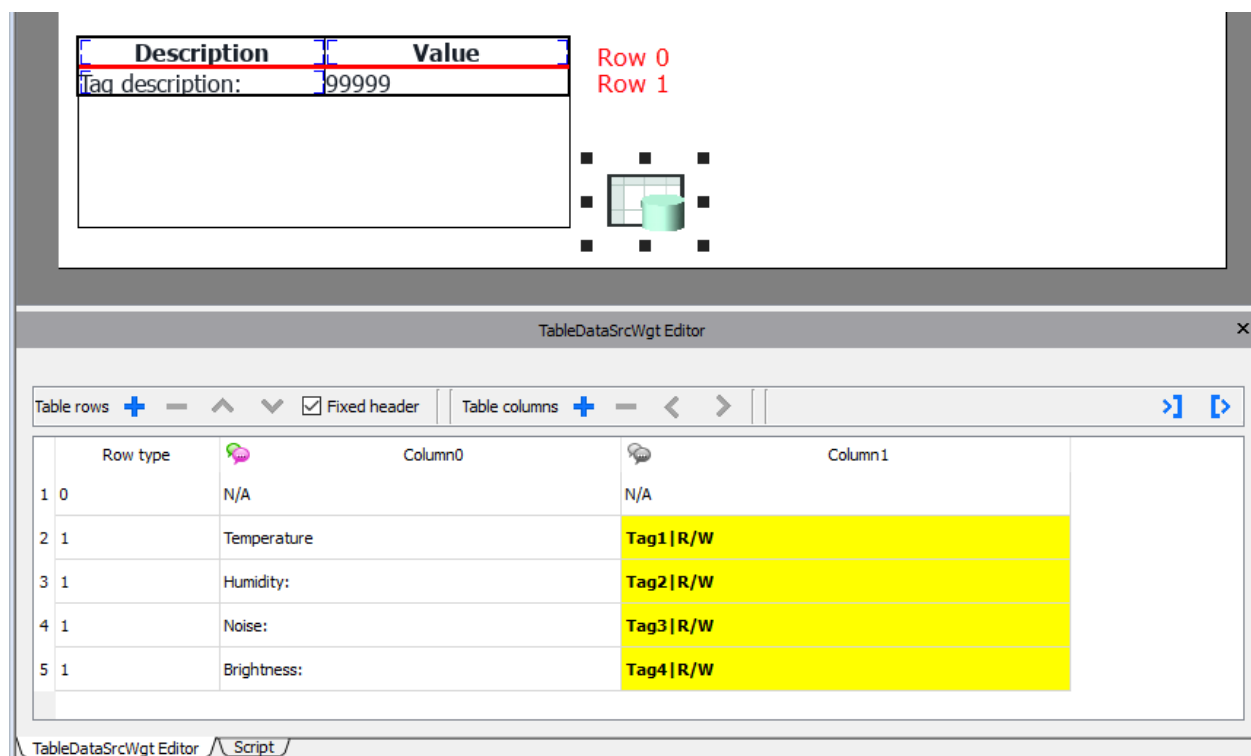
**Properties**

<b>TableGroupWgt : TableWgt</b>	
Current selected row	-1
<b>Table model</b>	
<b>DataLink</b>	model:TableDataSrcWgt
Access Type	R
<b>Grid Layout Group</b>	
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	2
External margin color	[0, 0, 0]
<b>Events</b>	
<b>General</b>	

Select the Data Source and inside the TableDataSrcWgt Editor add the rows and columns that are needed. In the following example, we have defined two row templates:



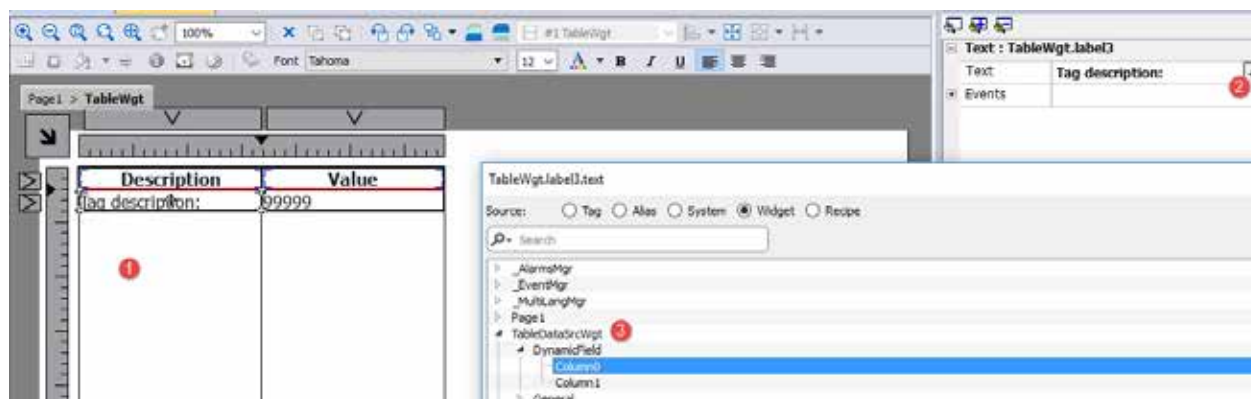
- Row 0  
Header of the table. Contains only static text.
- Row 1  
Template of rows with data. On the first column we added a label that will contain the description and on the second column a field that will contain the value.



Each row must be assigned a row type. The row will take on the format of the corresponding row template. Widgets that were placed in each cell of the row template will appear in rows of that type.

### Define links with data source

1. Double click over the Table widget to enter in edit mode and select a widget
2. Select the property that is to be read from the data source
3. Select the column of the data source that will provide the data



The below picture is showing how our example will be rendered at runtime

Description	Value
Temperature	111
Humidity:	222
Noise:	333
Brightness:	444

### Fixed Header


If you want the first row will be not scrollable, check the *"Fix Header"* check box on Data Source toolbox or set true the *"Show Header"* propert inside the Data Source properties panel (note the parameter is available only in advanced view).

### Multilanguage

To enable the Multilanguage support right click on the Multilanguage icon of the column. The icon will change color to indicate that the support is enabled.




Avoid enabling the Multilanguage support when not necessary to better performance.

Table rows + - ^ v <input checked="" type="checkbox"/> Fixed header		Table columns + - < >	
Row type	Column0		Column1
1 0	N/A		N/A
2 1	Temperature		Tag1   R/W
3 1	Humidity:		Tag2   R/W
4 1	Noise:		Tag3   R/W
5 1	Brightness:		Tag4   R/W

### Import/Export Data Source

The configuration of the Data Source can be imported/exported using xml files

Table rows + - ^ v <input checked="" type="checkbox"/> Fixed header		Table columns + - < >	
Row type	Column0		Column1
1 0	N/A		N/A
2 1	Temperature		Tag1   R/W

### JavaScript Object

In alternative to the Data Source Widget, for data to fill the table could be provided from a JavaScript Object. In this case, we have to fill an array of elements with the data to use and assign the array to the table widget.

```
var myTable = page.getWidget("TableWgt1");
```

```
myTable.model = model;
```

**model** is an array of elements with the table definition and data. The first element of the array will contain the template of the rows while the other elements will contain the data to fill in the rows of the table

```
model[0] = row_templates;    // row templates
model[1] = row_data1;        // data of the row1
model[2] = row_data2;        // data of the row2
model[3] = row_data3;        // data of the row3
model[4] = row_data4;        // data of the row4
model[5] = row_data4;        // data of the row5
```

The **row templates** is a multi dimensional array where each array defines the datalink of one template row.

On the below example, we have a template for two rows.

```
var row_templates = {
  _h : [
    [ [] , [] ], //rowType = 0
    [ ["text"] , ["value"] ] //rowType = 1
  ]
}
```

The first row has two columns that do not contain data links. We use this template for the header on the first row of our table.

The second row defines the template of one row with the “text” property of the widget into the first column and the “value” property of the widget into the second column. They will be dynamically filled using the data provided inside the model variable.

On the below example we define a **row of data**

```
var row_data = {
  _t : 1,
  _v : ["Temperature:", { _c : "dl" , s : "_TagMgr", a : "Tag1", i: 0, m : 2 }]
}
```

The first element is the row template to use while the second element is the array with the data to use. In our example “Temperature:” is the text to use inside the widget on the first column, while the other element is a datalink that will provide the value to fill the value property of the widget into the second column.

The datalink element:

Parameter	Description
<b>_c : "dl"</b>	Identify the element as a Datalink
<b>s : "_TagMgr"</b>	Specify the source of data is the Tag Manager
<b>a : "Tag1", i: 0, m:2</b>	Specify tag name and index (necessary when the tag is an array) and the read/write mode

Parameter	Description
	<ul style="list-style-type: none"> <li>• m=0 is Read Only</li> <li>• m=1 is Write Only</li> <li>• m=2 is Read/Write</li> </ul>

The below JavaScript code will generate the same table of the previous example using the Table Data Source Widget

```
var model = [];

var row_templates = {
  _h : [
    [ [] , [] ], //rowType = 0
    [ ["text"] , ["value"] ] //rowType = 1
  ]
}

var row_data1 = {
  _t : 0,
  _v : [],
  _h : true
}

var row_data2 = {
  _t : 1,
  _v : ["Temperature:", { _c : "dl" , s : "_TagMgr", a : "Tag1", i: 0, m : 2 }]
}

var row_data3 = {
  _t : 1,
  _v : ["Humidity:", { _c : "dl" , s : "_TagMgr", a : "Tag2", i: 0, m : 2 }]
}

var row_data4 = {
  _t : 1,
  _v : ["Noise:", { _c : "dl" , s : "_TagMgr", a : "Tag3", i: 0, m : 2 }]
}

var row_data5 = {
  _t : 1,
  _v : ["Brightness:", { _c : "dl" , s : "_TagMgr", a : "Tag4", i: 0, m : 2 }]
}

model[0] = row_templates;
model[1] = row_data1;
model[2] = row_data2;
model[3] = row_data3;
model[4] = row_data4;
model[5] = row_data5;
```

```
var myTable = page.getWidget("TableWgt1");
myTable.model = model;
```

Note the first row (row\_data1) contains the directive `_h: true` to avoid the first line will be scrollable.

```
var row_data1 = {
    _t : 0,
    _v : [],
    _h : true
}
```

## Multilanguage

A multi languages text can be entered using the below element:

```
{ _c : "ml" , mltext : { "en-US" : "Temperature:" , "it-IT" : "Temperatura:" } }
```

Parameter	Description
<code>_c : "ml"</code>	Identify the element as a Multilanguage text
<code>mltext : { ... }</code>	List of couples: "ID Language": "Text" Example: <ul style="list-style-type: none"> <li>• "en-US" : "Temperature:"</li> <li>• "it-IT" : "Temperatura:"</li> </ul>

Example:

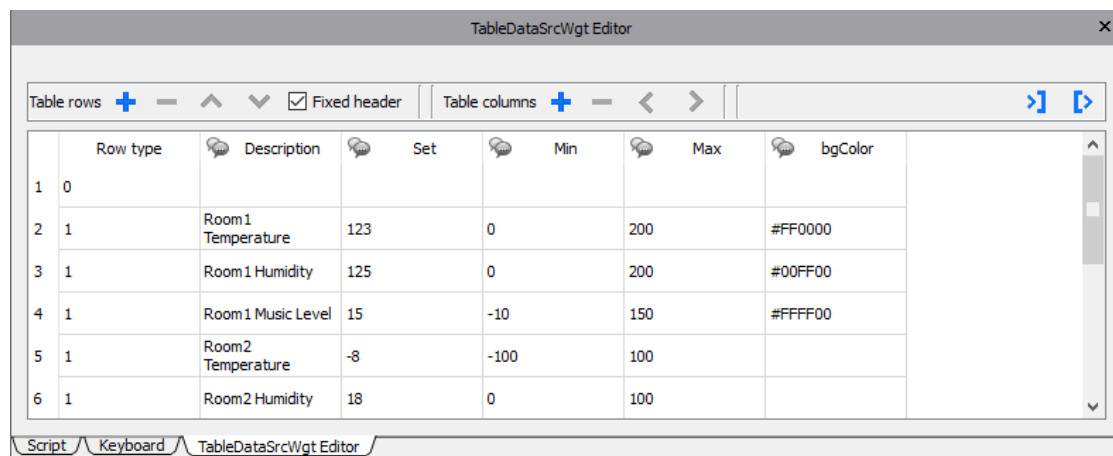
```
var row_data2 = {
    _t : 1,
    _v : [ { _c : "ml" , mltext : { "en-US" : "Temperature:",
                                   "it-IT" : "Temperatura:" } },
          { _c : "dl" , s : "_TagMgr", a : "Tag1", i: 0, m : 2 } ]
}
```

## Row background color

Using the Rows background parameter is possible define the column of the Data Source Widget that will contains the background color of the associate row.

To configure the background color of the rows of the table:

1. Add a new column inside the Data Source Widget to contain the background color of each row
2. Configure the "Row background" color parameter of the Table to point to the color column of the Data Source Widget



External margin width	1
External margin color	■ [0, 0, 0]
Rows background	
DataLink	bgColor:TableDataSrcWgt
Access Type	R
Table filter	
Events	

Filter:



Description	Set	Min	Max
Room1 Temperature	123	0	200
Room1 Humidity	125	0	200
Room1 Music Level	15	-10	150
Room2 Temperature	-8	-100	100
Room2 Humidity	18	0	100
Room2 Music Level	12	0	150
Room3 Temperature	15	0	150
Room3 Humidity	134	0	500

## Table Filter

Content visible inside the table can be filtered using the "Table Filter" property. On datalink you can use a formula (see ["Formula" on page 52](#) chapter for additional details) to define the criteria to use to filter the data.

Each row of the table will be visible only when the Datalink of the Table Filter return true value.

### Example 1

If you want choice to see only the rows that contain “something” inside the Description column, you can use the below formula:

```
= $Contains ($ ('Description:TableDataSrcWgt'), $ ('value:SearchOnTable'))
```

Where

- 'Description:TableDataSrcWgt' is a Dynamic Field of the Data Source Widget used from the table to identify the column to check
- 'value:SearchOnTable' is the value of a text field that will contains the string to search

Filter:  ✕

Description	Set	Min	Max
Room1 Temperature	123	0	200
Room1 Humidity	125	0	200
Room1 Music Level	15	-10	150
Room2 Temperature	-8	-100	100
Room2 Humidity	18	0	100
Room2 Music Level	12	0	150
Room3 Temperature	15	0	150
Room3 Humidity	134	0	500

Filter:  ✕

Room2 Temperature	-8	-100	100
Room2 Humidity	18	0	100
Room2 Music Level	12	0	150

Filter:  ✕

Room1 Temperature	123	0	200
Room2 Temperature	-8	-100	100
Room3 Temperature	15	0	150
Room4 Temperature	2	0	10

Properties

TableGroupWgt : TableWgt

Current selected row

-1

Table model

Grid Layout Group

Events

Table filter

DataLink

= \$Contains (\$ ('Description:TableDataSrcWgt'), \$ ('value:SearchOnTable'))

Access Type

R

General

Position

Text : TableWgt.label1

Text : TableWgt.label2

## 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

TableDataSrcWgt Editor

Table rows + - ^ v ☒ Fixed header Table columns + - < > >] <]

	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


Script Keyboard TableDataSrcWgt Editor



Properties	
<div> <div>TableGroupWgt : TableWgt</div> <div> <div>Current selected row</div> <div>-1</div> </div> <div> <div>Table model</div> <div>a +</div> </div> <div> <div>Grid Layout Group</div> <div></div> </div> <div> <div>Events</div> <div></div> </div> <div> <div>Table filter</div> <div>+ -</div> </div> <div> <div>DataLink</div> <div>enable:TableDataSrcWgt1 -</div> </div> <div> <div>Access Type</div> <div>R</div> </div> <div> <div>General</div> <div></div> </div> <div> <div>Position</div> <div></div> </div> <div> <div>Text : TableWgt.label1</div> <div></div> </div> <div> <div>Text : TableWgt.label2</div> <div></div> </div> </div>	

## 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

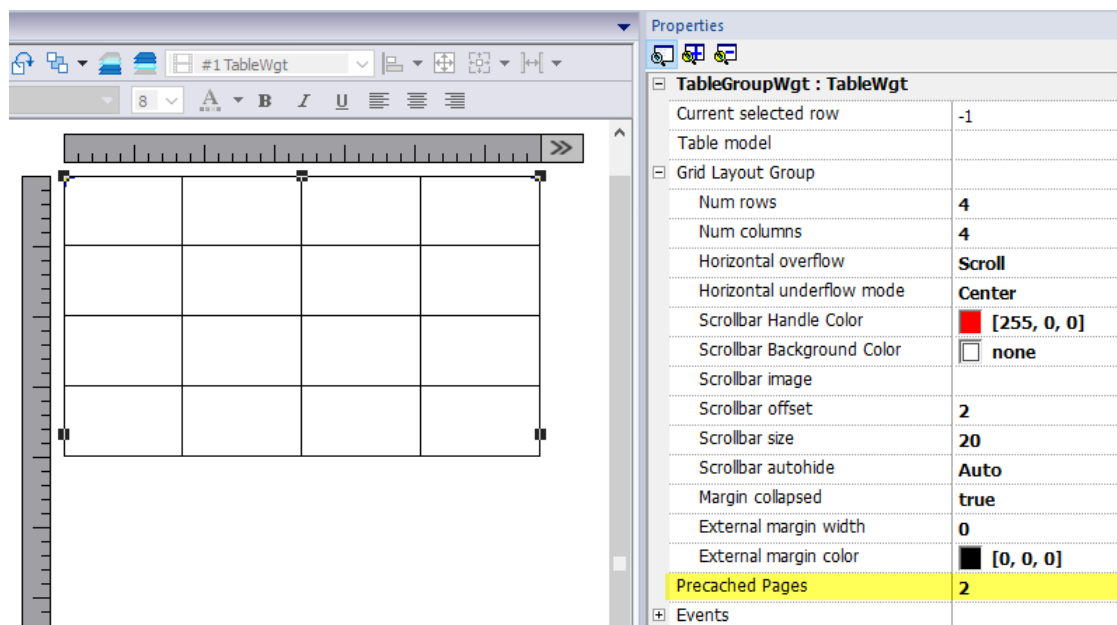
Normally the JMobile HMI Runtime retrieve only the data that will be visible into the display. To make table scrolling more pleasant, it may be useful to preload the data of the next and previous rows of the displayed ones. Using the Precached Pages parameter is possible define how many pages will be preloaded

- 0 = no pages preloaded
- N = number of pages to preload

Example:

Using a table with 4 rows and Precache Pages = 2

- Number of row to preload are 8 (2 pages x 4 rows)
- 4 rows before (to be ready to manage scroll table up)
- 4 row above (to be ready to manage scroll table down)



Precached Pages parameter is available only in Advanced Proprieties View mode

## TextEditor widget

Path: **Widget Gallery**> **Advanced**> **Editor**

Use this widget to edit text files. Widget can load the text file from the local HMI device or download the file from a remote device using an ftp connection.



Note: TextEditor widget is available as a plugin (see ["Plug-in" on page 77](#) for details) to download to the JMobile HMI Runtime only when required.



## Widget Buttons

Button	Description
Open	Load text file inside the TextEditor
Save	Save text file
Cancel	Remove all changes from last OPEN or SAVE command
Edit	Enter in edit mode
Insert	Insert a new line
Delete	Delete current line
Up/Down	Move cursor up/down

## Widget Properties

Option	Description												
Keyboard	<p>TextEditor widget has an embedded keyboard. When widget is used without the embedded keyboard, the alphabetic keyboard will be displayed when enter edit mode.</p> <ul style="list-style-type: none"> <li>• Hidden</li> <li>• Visible</li> </ul>												
FTP Config	<p>FTP parameters to download the text file from a remote FTP server. Leave this field empty to load the text file from the local HMI device.</p> <table> <tr> <th>Parameter</th><th>Description</th></tr> <tr> <td>FTP Address</td><td>FTP server IP Address</td></tr> <tr> <td>Server Port</td><td>Port for FTP connection (default = 21).</td></tr> <tr> <td>Authentication</td><td> <p>Select the FTP authentication to use:</p> <ul style="list-style-type: none"> <li>• Normal (Username and password required)</li> <li>• Anonymous</li> </ul> </td></tr> <tr> <td>User Name</td><td>Username of the remote FTP account</td></tr> <tr> <td>Password</td><td>Password of the remote FTP account</td></tr> </table>	Parameter	Description	FTP Address	FTP server IP Address	Server Port	Port for FTP connection (default = 21).	Authentication	<p>Select the FTP authentication to use:</p> <ul style="list-style-type: none"> <li>• Normal (Username and password required)</li> <li>• Anonymous</li> </ul>	User Name	Username of the remote FTP account	Password	Password of the remote FTP account
Parameter	Description												
FTP Address	FTP server IP Address												
Server Port	Port for FTP connection (default = 21).												
Authentication	<p>Select the FTP authentication to use:</p> <ul style="list-style-type: none"> <li>• Normal (Username and password required)</li> <li>• Anonymous</li> </ul>												
User Name	Username of the remote FTP account												
Password	Password of the remote FTP account												
File Name	File name to edit. When empty a file browser to load a local file will be opened												
Syntax Highlight Type	<p>Displays text in different colors according to the selected language</p> <ul style="list-style-type: none"> <li>• None</li> <li>• GCode</li> </ul>												

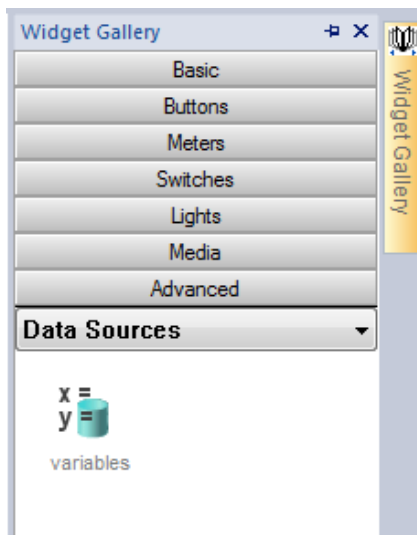
# Variables widget

**Path:** *Widget Gallery* > *Advanced* > *Data Sources*

Use this widget to add internal variables for operations such as data transfer or to be used in JavaScript programs.



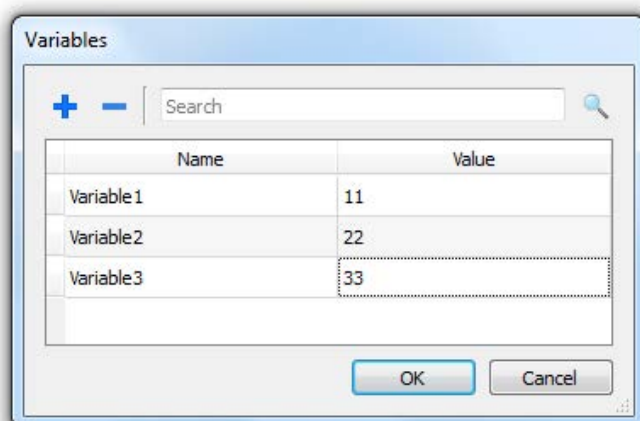
Note: The variables are local to the page where the widget has been inserted.



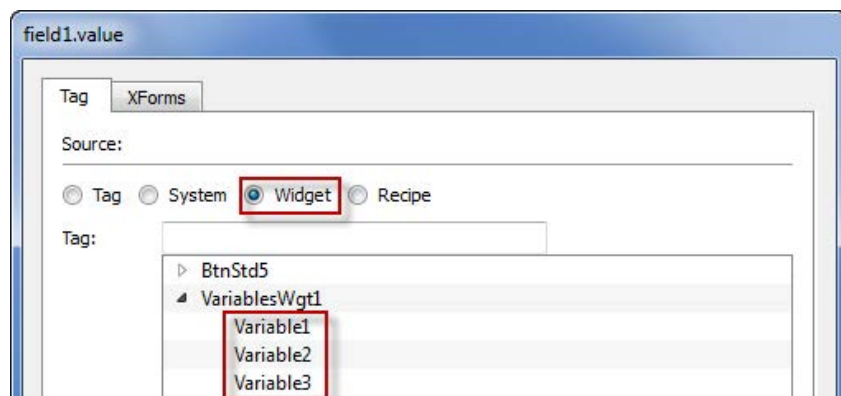
When you drag and drop this widget into your page, a placeholder will be displayed to indicate the widget location, but it will not be visible at run time.

## Setting the widget

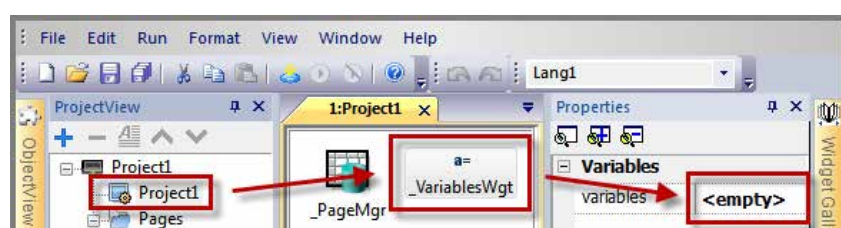
To create variables and assign values to them, open the **Variables** dialog from the **Variables** property in the **Properties** pane.



These variables can then be referenced from the **Attach tag** dialog, from the Page Editor.



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");
```



## 36 Custom widgets

---

JMobile Studio has a large widget library which includes predefined dynamic widgets (buttons, lights, gauges, switches, trends, recipes, and dialog items), as well as static images (shapes, pipes, tanks, motors).

You can drag and drop an object from the gallery to the page, and then size, move, rotate or transform it. All widgets in the gallery are vector based, so they do not lose definition when resized.

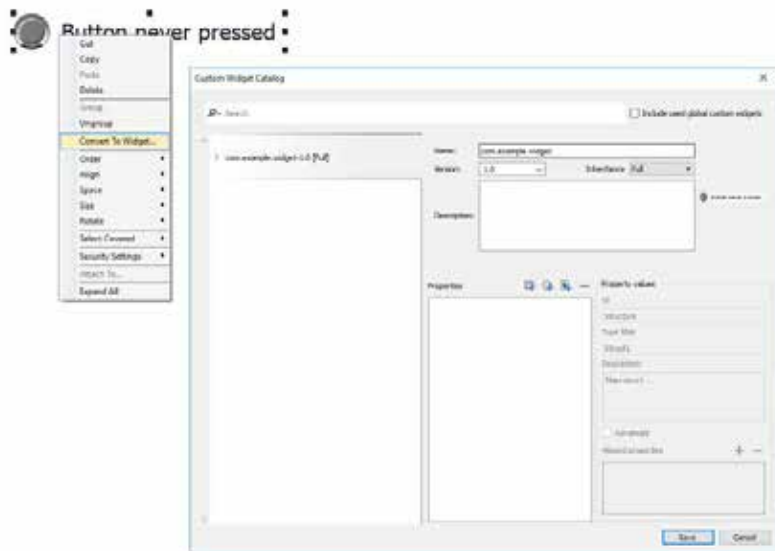
You can, however, modify any of the pre-defined widgets to create your own custom widget. Custom widgets can be made up of several elements only including the properties needed to their purpose.

---

<b>Creating a custom widget</b> .....	<b>436</b>
<b>Adding properties to a custom widget</b> .....	<b>438</b>
<b>Using structured tags</b> .....	<b>441</b>
<b>JavaScript in custom widgets</b> .....	<b>443</b>
<b>User's Gallery</b> .....	<b>446</b>

# Creating a custom widget

1. Drag and drop on a page all the widget you want to use to compose your custom widget.
2. Select and group them.
3. Right-click on the grouped object and select **Convert To Widget**: the **Custom Widgets Catalog** dialog is displayed.



Parameter	Description
<b>Include used custom widgets</b>	When checked, list all the widgets used inside the project. Even system widgets.
<b>Name</b>	You can define everything you prefer, but is common keep a name structure. The folder com.hmi is reserved for the system widgets
<b>Description</b>	Widget description.
<b>Version</b>	Widget version.  All widgets that share the same version share the properties defined from the Inheritance parameter.
<b>Inheritance</b>	Properties shared between widgets with the same version <ul style="list-style-type: none"> <li>• Full (both Graphic and Logic)</li> <li>• Only Graphic</li> <li>• Only Logic</li> <li>• Disable</li> </ul>

## Modify a custom widget

To modify a custom widget, simple double clicks the custom widget to enter in edit mode.



If the Inheritance flag is enabled, a lock icon will appear to warn you that you are adding changes that will be propagated to all the other custom widgets that share the same version. Click the padlock icon to confirm to enter in edit mode, padlock will be open. Click again when modifies are done.

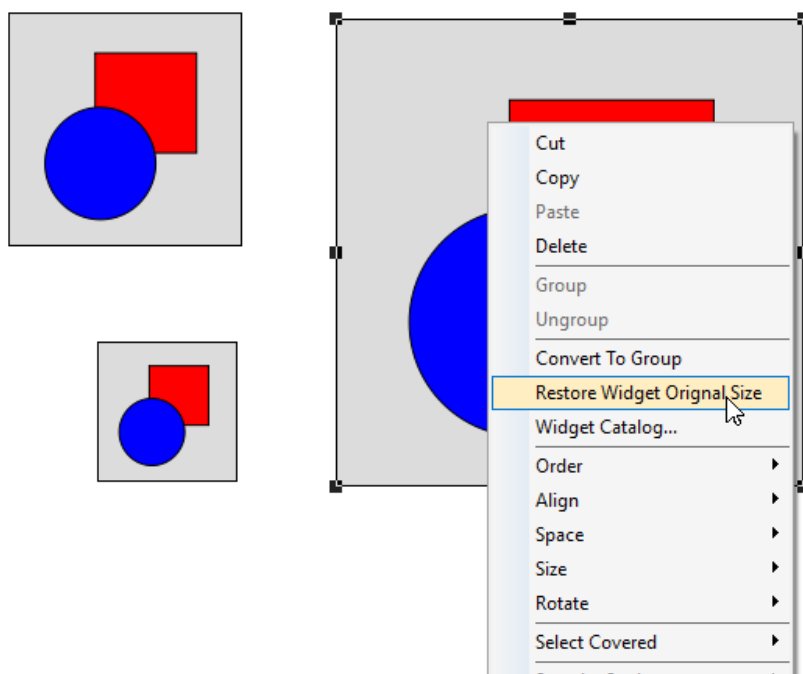


Padlock is showed only when the Inheritance is enabled.

## Resize a custom widget

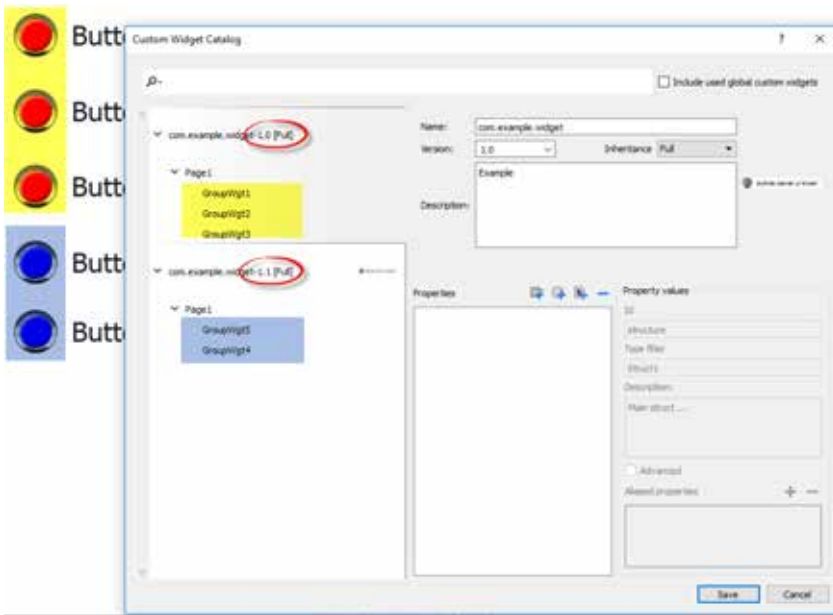
When sizes of custom widget is changed, the new sizes will not be propagated to the other widget instances.

“Restore Widget Original Size” command can be used.



## Share properties

When a custom widget is modified, all the modifies will be propagated to all the other custom widgets that share the same version and that are configured to inherit the widget properties.

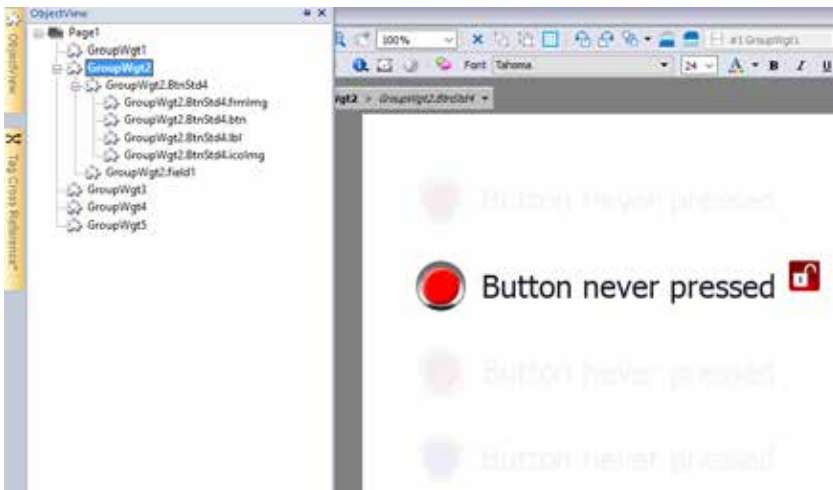


## Using widgets components

Widgets are usually made up of many parts, for example a button is a complex widget including two image widgets, a button widget and label.

To display a list of all the elements that are part of a widget, select the widget, open the padlock and open the **ObjectView** pane: all the element making up a complex widget are listed in hierarchical order.

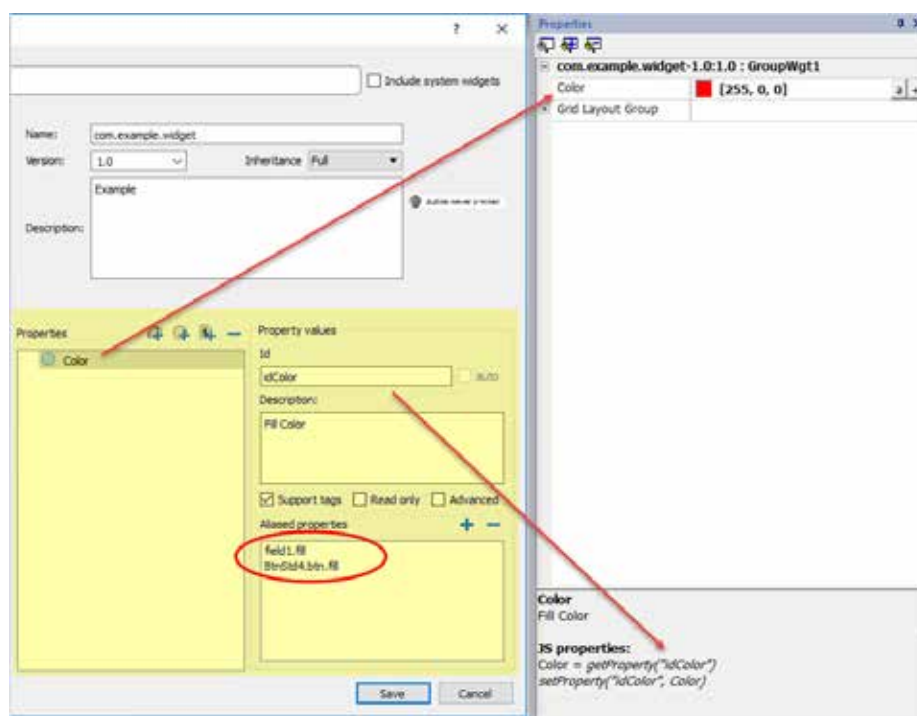
To select a single widget, select it directly from the **ObjectView** pane.



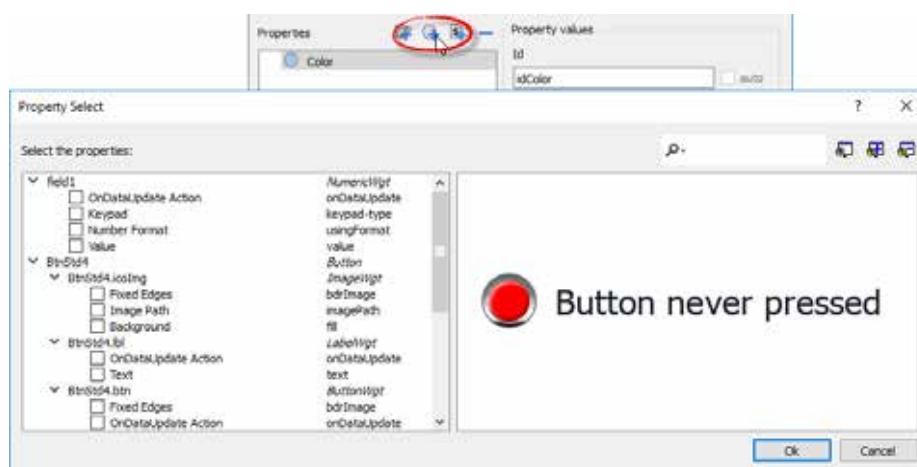
## Adding properties to a custom widget

When you create a custom widget, you need to define the properties that will be displayed for it in the **Properties** pane.

1. Right-click on the grouped object and select **Widget catalog**: the properties dialog is displayed.



2. Click **+** to open the **Property Select** dialog: this lists all the properties of all the grouped widgets.



3. Select the properties you want to define for your custom widget.
4. Define each property's details.



Note that you can create folders and use drag & drop to move or reorganize the **Properties** list

Parameter	Description
<b>Properties</b>	Name shown in the <b>Properties</b> pane.
<b>Description</b>	Any comment on the property to be displayed in the <b>Properties</b> pane.

Parameter	Description
<b>Id</b>	The name exposed by JMobile Studio, to JavaScript functions and Attach Tag dialog.
<b>Support Tags</b>	Specifies if the property supports the "Attach to" attribute.
<b>Read only</b>	Property exposed only in read mode
<b>Advanced</b>	Specifies whether each property should appear in the advanced, or in the simple view mode of the <b>Properties</b> pane.
<b>Aliased properties</b>	Internal properties linked with the exposed property

## Combining properties

To combine two or more properties:

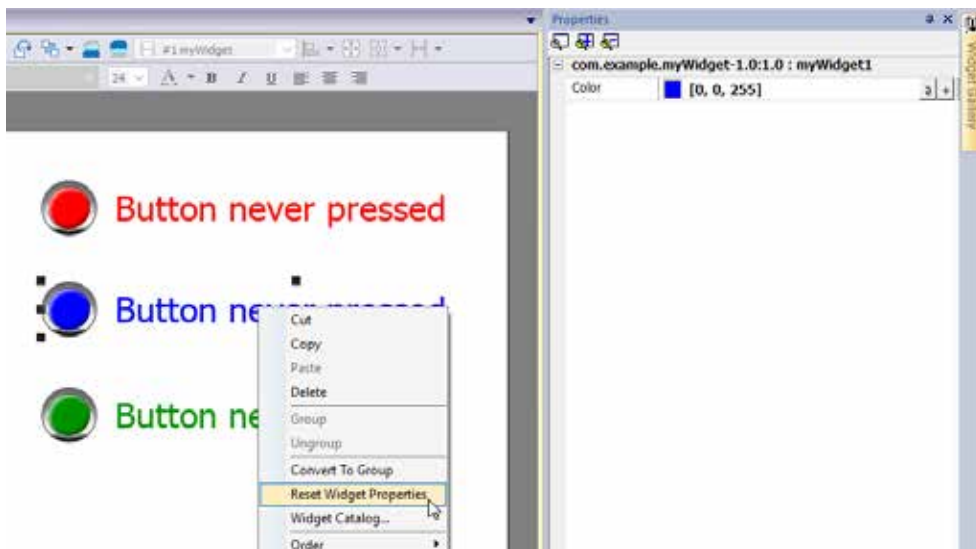
1. Select the primary property in the **Properties** list dialog.
2. Click **+** in the **Aliased properties** toolbar: the **Property Select** dialog is displayed.
3. Select the properties you want to combine.
4. Click **OK**: the combined attributes will be shown in the **Aliased properties** list box.

### Example

If you insert into a "Color" property the fill color of all widgets (e.g. filed1.fill and BtnStd4.btn.fill) when you set the exposed Color property of the custom widget all colors of the included widgets will changes.

## Reset Widget Properties

The "Reset Widget Properties" reset the modified properties values to original values.



## Using structured tags

A common problem using a widget that use many tags is the need to create instances of the widget by giving only the tag name of the structure that contains all the tags instead to configure each single tag.

For example, think about the below widget. It use four tags, the room name, temperature, humidity and pressure. If we want use two instances of this widget for two different rooms we have to configure eight tags, four tags for each room.

The screenshot displays two widget instances and their configuration. On the left, the 'Bathroom' widget shows Temperature: 23.0, Humidity: 52, and Pressure: 105. The 'Living room' widget shows Temperature: 21.0, Humidity: 22, and Pressure: 101. The 'Properties' panel for 'com.example.widget-1.0:1.0 : GroupWgt1' shows the following configuration:

Property	Value
Room	room1
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

The 'Select datatype for GroupWgt1.RoomID' dialog shows the following data structure:

Data	Type	Tag name
CODESYS V3 Ethernet	Container	
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.

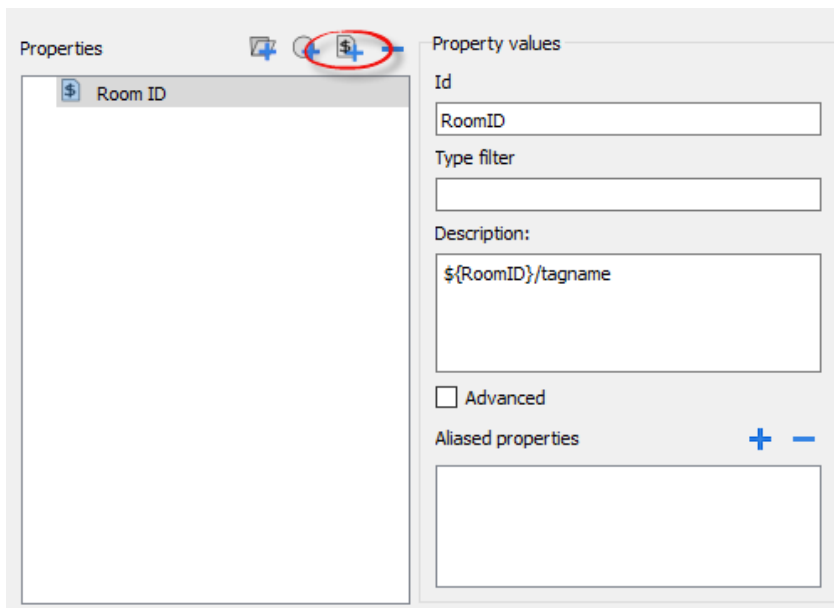
The screenshot displays two widget instances and their configuration. On the left, the 'Bathroom' widget shows Temperature: 23.0, Humidity: 52, and Pressure: 105. The 'Living room' widget shows Temperature: 21.0, Humidity: 22, and Pressure: 101. The 'Properties' panel for 'com.example.widget-1.0:1.0 : GroupWgt1' shows the following configuration:

Property	Value
Room ID	room1
Grid Layout Gr	

The 'Properties' panel for 'com.example.widget-1.0:1.0 : GroupWgt1' shows the following configuration:

Property	Value
Room ID	room2
Grid Layout Gr	

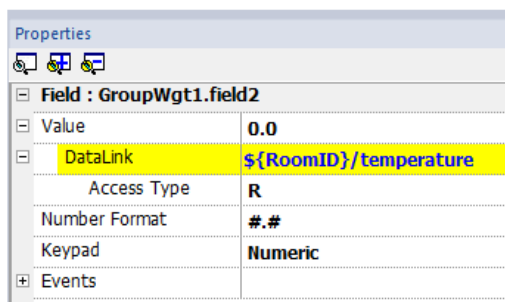
A "Parameter" field can be added inside the custom widget using the "Add Parameter" icon:



To configure the data links of the custom widget the keyword `${RoomID}` can be used to reference at the structure instance

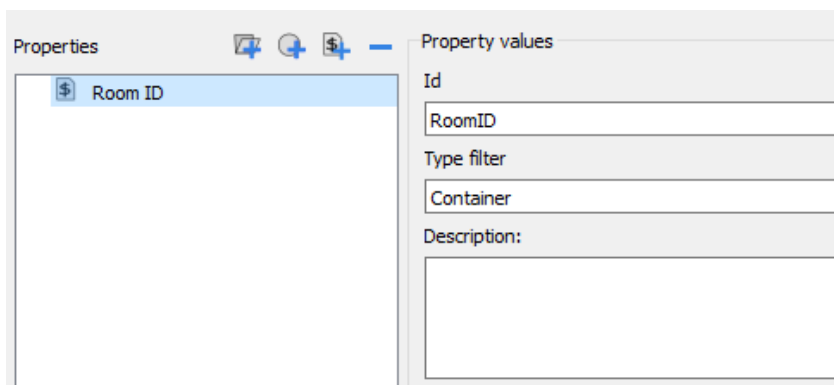
### Room

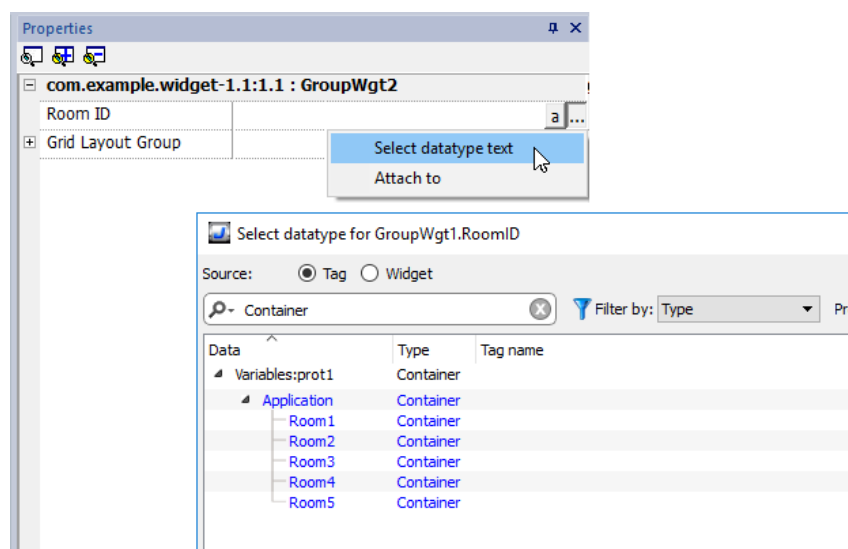
Temperature: 0.0  
Humidity: 0  
Pressure: 0



### Type filter

Typically, value of the parameter will be an element of a structured tag. Using the “*Type filter*” parameter, the “*Select datatype text*” will list filtered tags.





The “*Select datatype text*” will return a string while the “*Attach to*” will return a datalink to a tag that will contains the string to use.

## getParameter

From JavaScript you can read the parameters' value using the `getParameter()`

`object getParameter(paramID)`

Example:

```
var myWidget = page.getWidget("myWidget");
function BtnStd3_btn_onMouseClicked(me, eventInfo)
{
    alert("Room is: " + myWidget.getParameter("RoomID"));
}
```

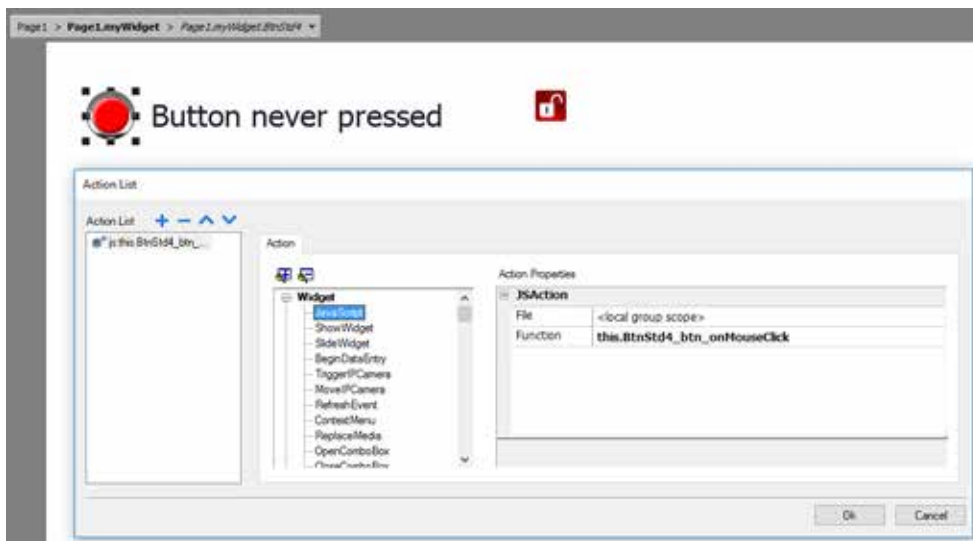


You can also use `getProperty()`, but `getParameter()` is more efficient to read custom widget parameters

## JavaScript in custom widgets

JavaScript functions can be embedded in custom widgets.

After doing a double click on the custom widget and clicked on the padlock, the edit mode is active and it is possible to associate the JavaScript code to the available events.




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.







If you cut and paste some instances of the custom widget of the above example and execute it, e.g. inside the simulator, you will obtain the below result.

 Tue Jan 31 2017 14:51:18

 Button never pressed

 Tue Jan 31 2017 14:51:12

 Tue Jan 31 2017 14:51:14

 Button never pressed

## onActivate property

To initialize the custom widget is possible to define the onActive property with an initializing function as for the below example.

The onActivate() function will be execute when the page is loading

```

1  /*!
2  javascript module: widget-1.0.js
3  javascript source file path: lib\com\example\widget-1.0\widget-1.0.js
4  */
5
6  this.wMSG = wgt.getWidget(wgt.id+".field1")
7
8  this.BtnStd1_btn_onMouseClicked = function (me, eventInfo)
9  {
10     var now = new Date();
11     this.wMSG.setProperty("value", now.toString().slice(0, 24));
12 }
13
14
15 this.onActivate = function()
16 {
17     this.wMSG.setProperty("value", "Button never pressed");
18 };
19 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_onMouseClicked = function (me, eventInfo)
{
    var now = new Date();
    this.wMSG.setProperty("value", now.toString().slice(0, 24));
}

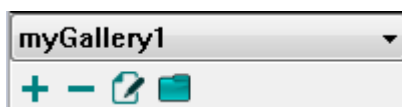
this.onActivate = function()
{
    this.wMSG.setProperty("value", "Button never pressed");
};
this.onActivate();





```

## User's Gallery

Widgets created from the developers can be saved inside the Widgets Gallery to be available during development of new projects.

### User widgets toolbar

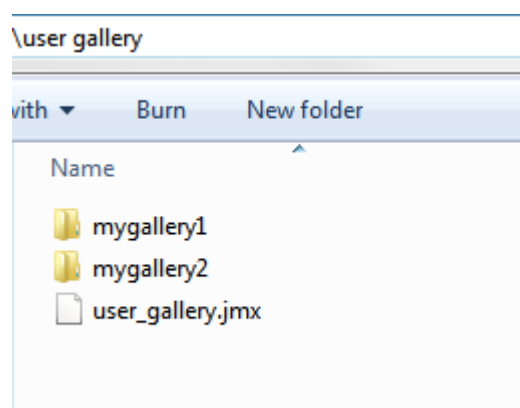


Command	Description
	Open the selected widgets folder into the JMobile Studio editor
	Add a new widgets folder
	Delete current selected folder
	Select the user widgets folder

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.

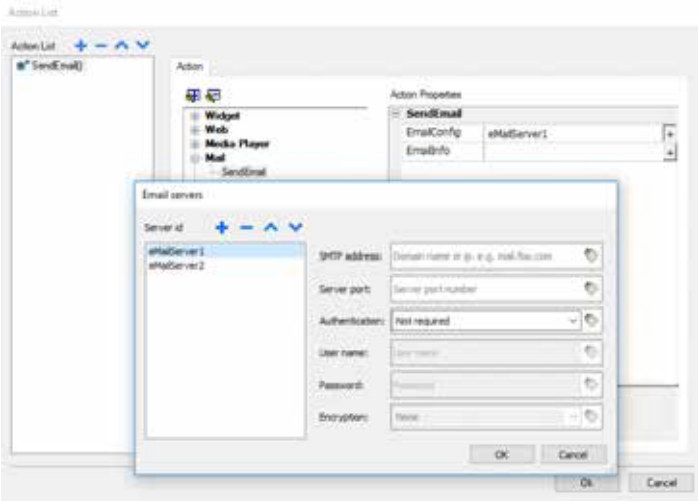




# 37    Sending an email message

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.



Configuring the email server .....	450
Configure emails .....	450

# Configuring the email server

To configure the email server, enter the following information for the **EmailConfig** setting:

Parameter	Description
<b>SMTP Address</b>	SMTP server address.
<b>Server Port</b>	Port for SMTP server connection (default = 25).
<b>Require Auth</b>	Select if the SMTP server requires authentication.
<b>User Name</b>	Username for sending mail using SMTP server.
<b>Password</b>	Password for sending mails using SMTP server.
<b>Encryption</b>	Encryption type (none or SSL).


Click **+** to add more email servers.



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

## Configure emails

Enter the following information for the **EmailInfo** setting:

Parameter	Description
<b>Name</b>	Optional, this information is only for the log.
<b>Description</b>	Optional, this information is only for the log.
<b>From</b>	Optional, sender email address (for example, John@domain.com).
<b>To</b>	Recipient e-mail addresses. To enter multiple addresses, separate them with a semi-colon.
<b>Subject</b>	Subject of email.
<b>Attachment</b>	<p>Path of the file to be sent as attachment. Only one attachment at a time can be sent.</p> <div>  <p>Note: The maximum size of the attachments is usually set by the SMTP server.</p> </div>
<b>Body</b>	<p>Main content of the email. Here you can insert live tags if you include them in square brackets.</p> <p>For example, a message body as "Tag1 value is [Tag1]", will be sent as "Tag1 value is 45", if the current value of Tag1 is 45.</p>



Tip: Attach a string tag to the **From**, **To** and **Subject** fields so that their value can be changed in the JMobile 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.

The screenshot shows a dialog box titled "Emails". On the left, under the "Drafts" header, there is a list box containing "eMail1" and a toolbar with icons for adding (+), removing (-), moving up (^), and moving down (v). To the right of the list box are form fields for "Name", "Description", "From", "To", "Subject", and "Attachment". Each of the "From", "To", and "Subject" fields has a small icon to its right. Below these fields is a large text area labeled "Message". At the bottom right of the dialog are "OK" and "Cancel" buttons.





## 38 JavaScript

---

The purpose of this section is to describe how JavaScript is used in the JMobile Studio applications, not to explain the JavaScript language.

JMobile Studio 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 JMobile Studio as you do in a web browser. If you are not familiar with the ECMAScript language, refer to:

<https://developer.mozilla.org/en/JavaScript>

---

<b>JavaScript editor</b> .....	<b>455</b>
<b>Execution of JavaScript functions</b> .....	<b>455</b>
<b>Events</b> .....	<b>457</b>
<b>Widget events</b> .....	<b>458</b>
<b>Page events</b> .....	<b>460</b>
<b>System events</b> .....	<b>461</b>
<b>Objects</b> .....	<b>463</b>
<b>Widget class objects</b> .....	<b>463</b>
<b>Widget properties</b> .....	<b>464</b>
<b>Widget methods</b> .....	<b>467</b>
<b>Page object</b> .....	<b>468</b>
<b>Page object properties</b> .....	<b>468</b>
<b>Page object methods</b> .....	<b>469</b>
<b>Group object</b> .....	<b>471</b>
<b>Group object methods</b> .....	<b>471</b>
<b>Project object</b> .....	<b>473</b>
<b>Project object properties</b> .....	<b>473</b>
<b>Project object methods</b> .....	<b>473</b>
<b>Project object widgets</b> .....	<b>482</b>
<b>Print reports object</b> .....	<b>483</b>
<b>State object</b> .....	<b>486</b>
<b>State object methods</b> .....	<b>486</b>
<b>Keywords</b> .....	<b>487</b>

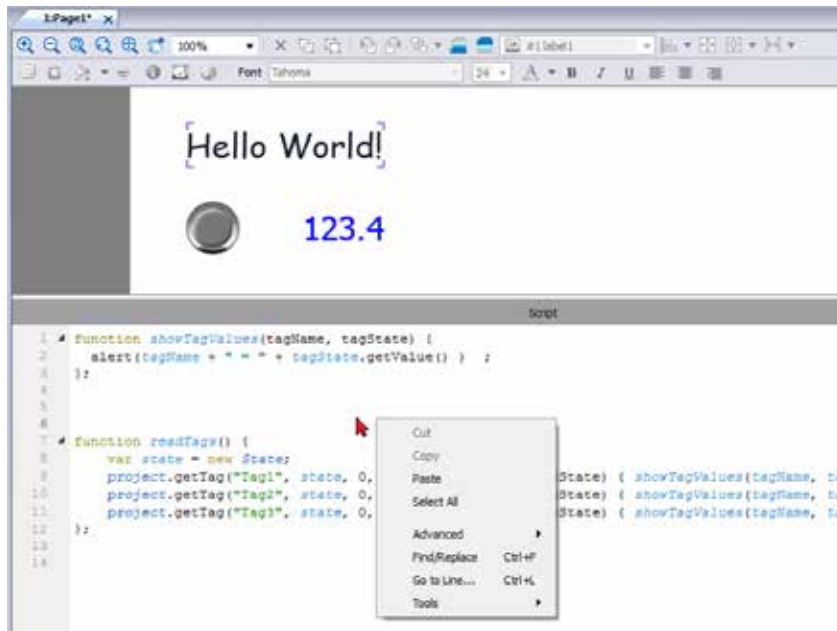
---

<b>Global functions .....</b>	<b>488</b>
<b>Handling read/write files .....</b>	<b>488</b>
<b>Sign in from JavaScript .....</b>	<b>491</b>
<b>Limitations in working with widgets in JavaScript .....</b>	<b>493</b>
<b>Debugging of JavaScript .....</b>	<b>494</b>

## JavaScript editor

JMobile Studio includes a powerful JavaScript editor.

Right-click in the editor to display available commands.



## Execution of JavaScript functions

JavaScript functions are executed when events occur. For example, a user can define a script for the OnMouseClicked event and the JavaScript script will be executed when the button is pressed on the HMI device.

JavaScript functions are executed only when the programmed event occurs and not cyclically. This approach minimizes the overhead required to execute logic in the HMI device.

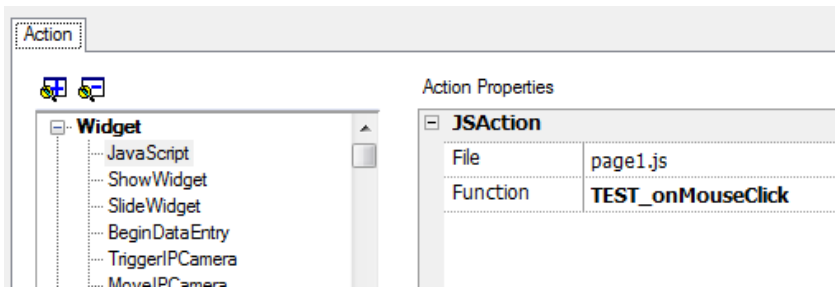
JMobile Studio provides a JavaScript engine running on the client side. Each project page can contain scripts having a scope local to the page where they are added; global scripts can be created to be executed by scheduler events or alarm events.

In both cases scripts are executed on the client. This means that if more than one client is connected to the HMI device (for external computer running the HMI Client), each client will run the same script, providing different output results depending on the input, since inputs provided to different clients may be different.

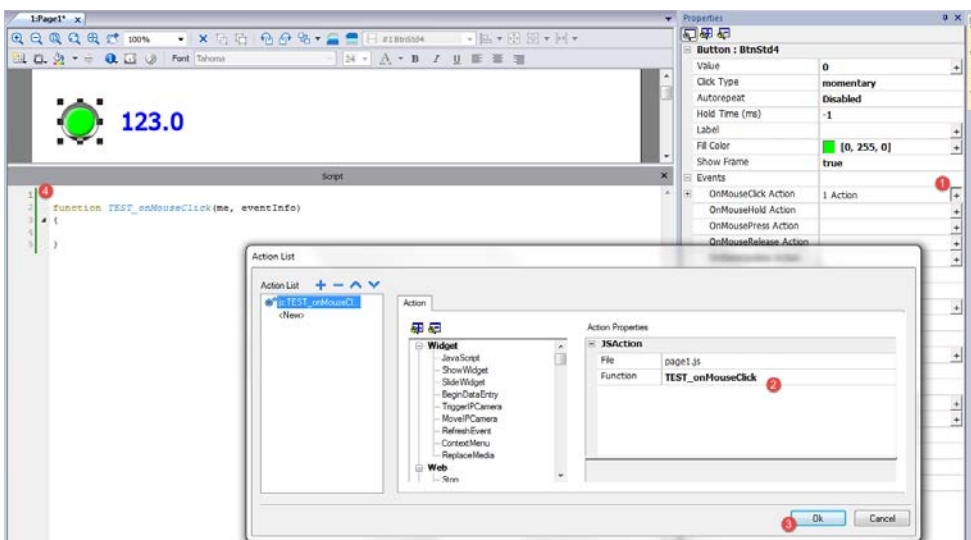
For example, if a script acts according to the position of a slider and this position is different on the different clients, the result of the script will be different on each client.

### JavaScript functions for page events

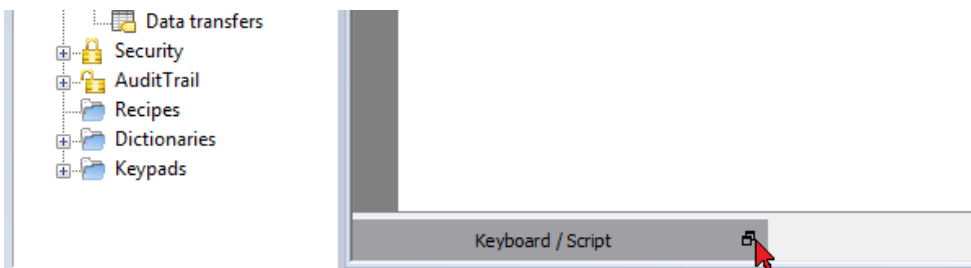
JavaScript editor will open when you add a JavaScript action inside an action list.



1. Select the event that will execute the action.
2. Add a **JavaScript** action from the **Widget** category.
3. Either leave the default function name, or type a new one.
4. Click **OK** to confirm: the JavaScript editor displays your function structure.



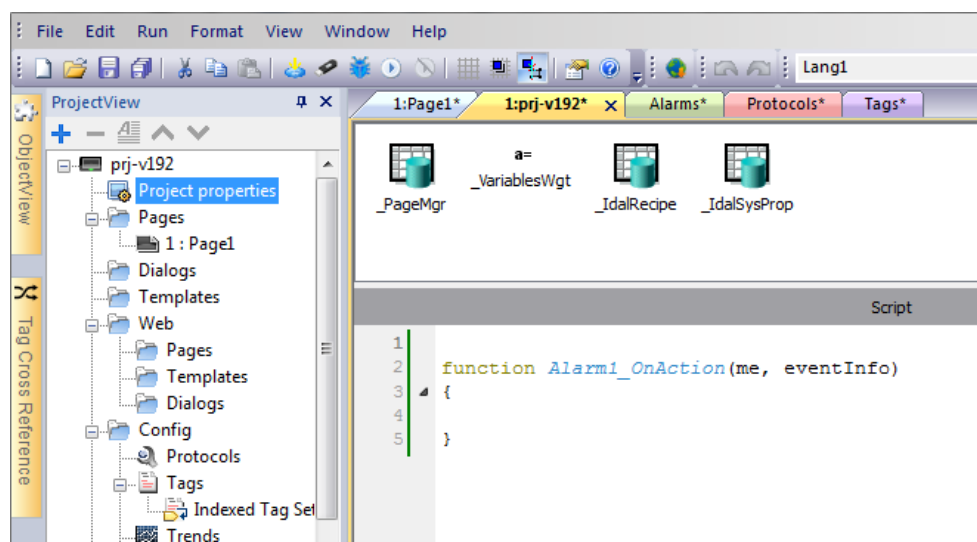
You can also open the JavaScript editor from the **Script** tab at the bottom of the workspace.



## JavaScript functions for alarms and scheduled events

JavaScript code associated with alarms and scheduled events and not associated with a specific page, can be edited from the main **Project properties** page.

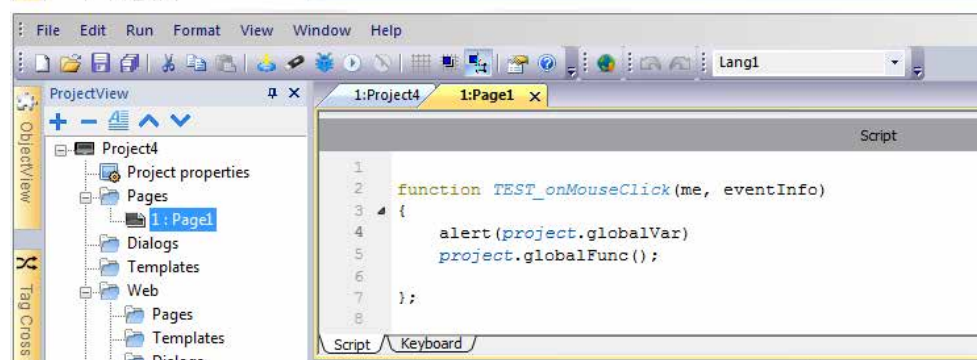
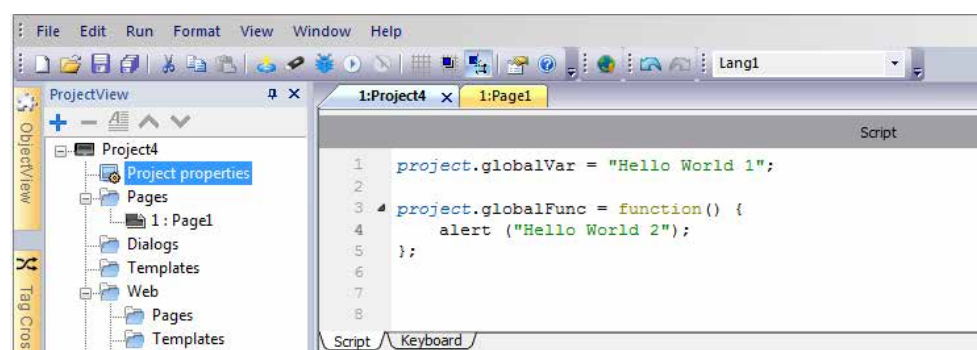
**Path:** *ProjectView* > double-click *Project properties*



Note: JavaScript actions are client actions so they are executed only when a client is logged in.

## Shared JavaScript code

The **project** global variable can be used to share JavaScript code between the pages. Variables are created/initialized from the main JavaScript code from the main **Project properties** page and can then be used from the project pages.



## Events

You can add JavaScript to the following categories of events:

- Widget events
- Page events
- System events

For events of type:

- OnMousePress
- OnMouseRelease
- OnMouseClicked
- OnWheel

JavaScript **eventInfo** parameter contains the following additional properties:

Parameter	Description
<b>eventInfo.posX</b>	Local mouse/touch X coordinate with respect to widget coordinates
<b>eventInfo.posY</b>	Local mouse/touch Y coordinate with respect to widget coordinates
<b>eventInfo.pagePosX</b>	Page X mouse/touch coordinate
<b>eventInfo.pagePosY</b>	Page Y mouse/touch coordinate
<b>eventInfo.wheelDelta</b>	<p>Mouse wheel delta. Integer value with sign representing the rotation direction.</p> <p>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.</p>

## Widget events

### onMouseClicked

```
void onMouseClick (me, eventInfo)
```

This event is available only for buttons and it occurs when the button is pressed and released quickly.

Parameter	Description
<b>me</b>	Object triggering the event
<b>eventInfo</b>	Details of triggered event

```
function buttonStd1_onMouseClicked(me, eventInfo) {
    //do something...
}
```

### onMouseHold

```
void onMouseHold (me, eventInfo)
```

This event is available only for buttons and it occurs when the button is pressed and released after the number of seconds set as **Hold Time** in the widget properties.

Parameter	Description
<b>me</b>	Object triggering the event
<b>eventInfo</b>	Details of triggered event

```
function buttonStd1_onMouseHold(me, eventInfo) {
    //do something...
}
```

## onMousePress

```
void onMousePress(me, eventInfo)
```

This event is available only for buttons and it occurs when the button is pressed.

Parameter	Description
<b>me</b>	Object triggering the event
<b>eventInfo</b>	Details of triggered event

```
function buttonStd1_onMousePress(me, eventInfo) {
    //do something...
}
```

## onMouseRelease

```
void onMouseRelease (me, eventInfo)
```

This event is available only for buttons and it occurs when the button is released.

Parameter	Description
<b>me</b>	Object triggering the event
<b>eventInfo</b>	Details of triggered event

```
function buttonStd1_onMouseRelease(me, eventInfo) {
    //do something...
}
```

## onDataUpdate

```
boolean onDataUpdate (me, eventInfo)
```

This event occurs when data attached to the widget changes.

Parameter	Description
<b>me</b>	Object triggering the event
<b>eventInfo</b>	<p>An object with the fields listed below (you can refer fields using “.” - dot notation)</p> <p><b>oldValue</b> = Widget value before the change</p> <p><b>newValue</b> = Value which will be updated to the widget</p> <p><b>attrName</b> = Attribute on which the event is generated</p> <p><b>index</b> = Integer attribute index if any, default = 0</p> <p><b>mode</b> = W when the user is writing to the widget. R in all others status.</p>

The event is triggered before the value is passed to the widget. A JavaScript code can intercept the event and decide to avoid to update the widget by return true value.



Note: if there are additional macros associate at the event, all macros will be execute regardless of the return value used inside the JavaScript code.

```
function buttonStd1_onDataUpdate(me, eventInfo) {
  if ( eventInfo.newValue > 100) {
    //do something...
    return true; // To avoid to continue and update
                // the widget (e.g. not update the linked tag)
  }
  return false; // To continue and update the widget
                // (e.g. update the linked tag)
}
```

## Page events

### onActivate

```
void onActivate( me, eventInfo )
```

This event occurs each time the page is displayed.

Parameter	Description
<b>me</b>	Object triggering the event
<b>eventInfo</b>	Reserved for future use

JavaScript will be executed when the page is active, that is when the page is loaded.

```
function Page1_onActivate(me, eventInfo) {
  //do something...
}
```



## onDeactivate

```
void onDeactivate( me, eventInfo )
```

This event occurs when leaving the page.

Parameter	Description
<b>me</b>	Object triggering the event
<b>eventInfo</b>	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
<b>me</b>	Object triggering the event
<b>eventInfo</b>	Details of triggered event

```
function Page1_onMouseWheelClock(me, eventInfo) {
    //do something...
}
```

# System events

System events can be related to:

- scheduler
- alarms
- a wheel device

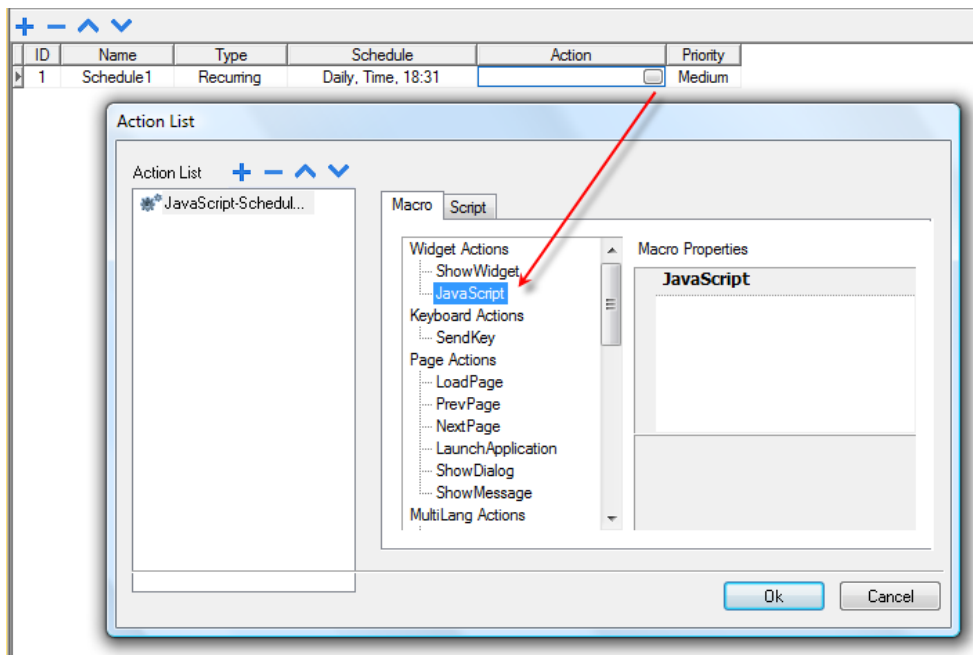


**Important: Make sure you do not duplicate JavaScript function names at page and project level. When a conflict happens, that is two functions with the same name in current page and at project level, the system execute the JavaScript callback at page level.**

When a JavaScript callback is not found in the current page, the system automatically searches for it at project level.

## Scheduler events

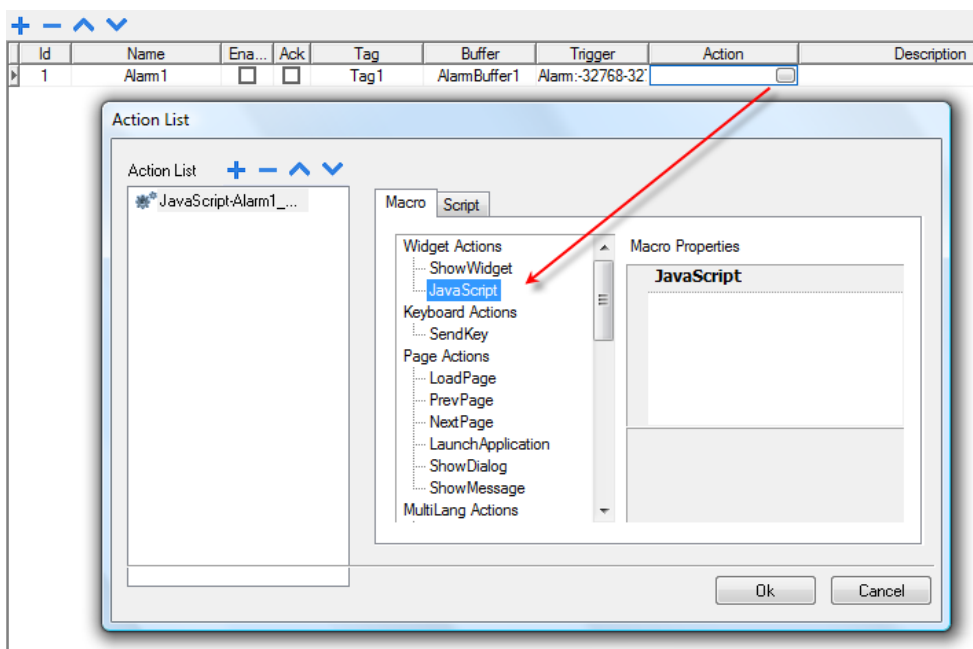
These events occur when triggered by the associated action in the scheduler.



You can edit the JavaScript from the **Project Properties** tab.

## Alarm events

These events occur when triggered by the associated alarm condition.



You can edit the JavaScript from the **Project Properties** tab.

## onWheel

```
void onMouseWheelClock( me, eventInfo )
```

This event occurs when a wheel device is moving (for example, a mouse wheel).

Parameter	Description
<b>me</b>	Object triggering the event
<b>eventInfo</b>	Details of triggered event

```
function Project1_onMouseWheelClock(me, eventInfo) {
    //do something...
}
```

## Objects

JMobile Studio uses JavaScript objects to access the elements of the page. Each object is composed of properties and methods that are used to define the operation and appearance of the page element. The following objects are used to interact with elements of the HMI device page:

Object	Description
Widget	This is the base class for all elements on the page including the page element
Page	This object references the current HMI device page. The page is the top-level object of the screen.
Group	This object associates a set of tags to allow uniform operation on a set of logically connected tags
Project	This object defines the project widget. The project widget is used to retrieve data about the project such as tags, alarms, recipes, schedules, tags and so on. There is only one widget for the project and it can be referenced through the project variable.
State	This object is the class holding the state of a variable acquired from the controlled environment. Beside the value itself, it contains the timestamp indicating when the value was collected and flags marking the quality of the value.

## Widget class objects

The Widget class is the base class for all the elements on a page including the page element.

Widget, in this case, is not used to indicate a specific screen object but a JavaScript class.

### Changing widget properties with JavaScript

If you want to change the properties of widgets with JavaScript set the widget property **Static Optimization** to **Dynamic**.



**Important:** If the widget property **Static Optimization** is not set to **Dynamic**, changes to properties will be ignored.

Whenever a call to `getWidget` fails, the remote debugger reports the following error:

*“Trying to access static optimized widget "label1". Disable widget static optimization to access widget from script.”*

This error is visible also using following code fragment:

```
var wgt;
try {
  wgt = page.getWidget('label1');
} catch(err) {
  alert("" + err);
}
```

## Widget properties

Some properties are common to all widgets.

### objectName

string objectName

Gets the name of the widget, a unique id.

```
function btnStd04_onMouseRelease(me) {
  var wgt = page.getWidget("rect1");
  var name = wgt.objectName;
}
```

(Available on web pages)

### x

number x

Gets or sets the widget 'x' position in pixels.

```
function btnStd1_onMouseRelease(me) {
  var wgt = page.getWidget("rect1");
  wgt.x = 10;
}
```

(Available on web pages)

### y

number y

Gets or sets the widget 'y' position in pixels.

```
function btnStd1_onMouseRelease(me) {  
    var wgt = page.getWidget("rect1");  
    wgt.y = 10;  
}
```

(Available on web pages)

## width

number width

Gets or sets the widget width in pixels.

```
function btnStd1_onMouseRelease(me) {  
    var wgt = page.getWidget("rect1");  
    wgt.width = 10;  
}
```

(Available on web pages)

## height

number height

Gets or sets the widget height in pixels.

```
function btnStd1_onMouseRelease(me) {  
    var wgt = page.getWidget("rect1");  
    wgt.height = 10;  
}
```

(Available on web pages)

## visible

boolean visible

Gets or sets the widget visible state.

```
function btnStd4_onMouseRelease(me) {  
    var wgt = page.getWidget("rect1");  
    wgt.visible = false;  
}  
  
function btnStd5_onMouseRelease(me) {  
    var wgt = page.getWidget("rect1");  
    wgt.visible = true;  
}
```

```
}
```

## value

number value

Gets or sets the widget value.

```
function btnStd6_onMouseRelease(me) {
    var wgt = page.getWidget("field1");
    wgt.value = 100;
}
```

## opacity

number opacity (range from 0 to 1)

Gets or sets the widget opacity. Values are decimals from 0 to 1, where 1 is 100% opaque.

```
function btnStd8_onMouseRelease(me) {
    var wgt = page.getWidget("rect1");
    wgt.opacity = 0.5;
}
```

(Available on web pages)

## rotation

number rotation (in degrees)

Gets or sets the rotation angle for the widget. The rotation is done clockwise and by degrees, starting at the East position.

```
function btnStd9_onMouseRelease(me) {
    var wgt = page.getWidget("rect1");
    wgt.rotation = 45;
}
```

(Available on web pages)

## userValue

string userValue

Gets or sets a user-defined value for the widget. This field can be used by JavaScript functions to store additional data with the widget.

```
function btnStd9_onMouseRelease(me) {
    var wgt = page.getWidget("rect1");
    wgt.userValue = "Here I can store custom data";
}
```

```
}
```

Every widget has some specific properties that you can access using dot notation. For an up-to-date and detailed list of properties you can use the JavaScript Debugger inspecting the widget methods and properties.

## Widget methods

Some methods are common to all widgets.

### getProperty

```
object getProperty( propertyName, [index] )
```

Returns a property.

Parameter	Description
<b>propertyName</b>	String containing the name of property to get
<b>index</b>	Index of the element to get from the array (default = 0)

Almost all properties that are shown in the JMobile Studio **Properties** pane can be retrieved using the `getProperty` method. The index value is optional and only used for widgets that support arrays.

```
function buttonStd1_onMouseRelease(me, eventInfo) {
    var shape = page.getWidget("rect2");
    var y_position = shape.getProperty("y");
}
```

```
function buttonStd2_onMouseRelease(me, eventInfo) {
    var image = page.getWidget("multistate1");
    var image3 = image.getProperty("imageList", 2);
    //...
}
```

(Available on web pages)

### setProperty

```
boolean setProperty( propertyName, value, [index] )
```

Sets a property for the widget.

## Parameters

Parameter	Description
<b>propertyName</b>	String containing the name of property to set
<b>value</b>	String containing the value to set the property.
<b>index</b>	Index of the element to set in the array (default = 0)

Almost all properties that are shown in the JMobile Studio **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
<b>wgtName</b>	String containing the widget name

## Return value

An object representing the widget. If the widget does not exist, null is returned.

```
function btnStd1_onMouseRelease(me) {  
    var my_button = page.getWidget("btnStd1");  
}
```

(Available on web pages)

## setTimeout

number setTimeout( functionName, delay )

Starts a timer to call a given function after a given delay.

Parameter	Description
<b>functionName</b>	String containing the name of function to call
<b>delay</b>	Delay in milliseconds

## Return value

A number corresponding to the timerID.

```
var duration = 3000;  
var myTimer = page.setTimeout("innerChangeWidth()", duration);
```

(Available on web pages)

## clearTimeout

void clearTimeout( timerID )

Stops and clears the timeout timer with the given timer.

Parameter	Description
<b>timerID</b>	Timer to be cleared and stopped

```
var duration = 3000;  
var myTimer = page.setTimeout("innerChangeWidth()", duration);  
// do something  
page.clearTimeout(myTimer);
```

(Available on web pages)

## setInterval

number setInterval( functionName, interval )

Starts a timer that executes the given function with the given interval.

Parameter	Description
<b>functionName</b>	String containing the name of function to call
<b>interval</b>	Interval in milliseconds

### Return value

A number corresponding to the timerID.

```
var interval = 3000;
var myTimer = page.setInterval("innerChangeWidth()", interval);
```

(Available on web pages)

## clearInterval

```
void clearInterval( timerID )
```

Stops and clears the interval timer with the given timer.

Parameter	Description
<b>timerID</b>	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)

# 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)

## Project object

This object defines the project widget. The project widget is used to retrieve data about the project such as tags, alarms, recipes, schedules, tags and so on. There is only one widget for the project and it can be referenced through the project variable.

## Project object properties

Properties to be set at project level.

### startPage

string startPage

Page shown when the project is started.

```
var startPage = project.startPage;  
project.startPage = "Page2.jmx";
```

## Project object methods

Methods to be used at project level.

### nextPage

void nextPage()

The script executes the Next page action.

```
project.nextPage();
```

(Available on web pages)

### prevPage

void prevPage()

The script executes the previous page action.

```
project.prevPage();
```

(Available on web pages)

### lastVisitedPage

void lastVisitedPage()

The script executes the last visited page action.

```
project.lastVisitedPage();
```

(Available on web pages)

## homepage

```
void homePage()
```

The script executes the Home page action.

```
project.homePage();
```

(Available on web pages)

## loadPage

```
void loadPage (pageName)
```

The script executes to load the set page defined in the script.

```
project.loadPage ("Page5.jmx");
```

(Available on web pages)



**WARNING:** When page change, all active time events are forced to removed and the JavaScript procedure will run until the end before switch to the new page.

## showDialog

```
void showDialog (pageName)
```

The script executes to show the dialog page.

```
project.showDialog ("Dialog.jmx");
```

(Available on web pages)

## closeDialog

```
void closeDialog()
```

The script executes to close the currently-opened dialog page.

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

(Available on web pages)

## showMessage

```
void showMessage ( message )
```

The script executes to display the message popup.

```
project.showMessage("Hi This is test message");
```

(Available on web pages)

## getGroup

```
number getGroup( groupName, groupInstance, [callback] )
```

Fast read method; this gets the values of all tags in a group.

Parameter	Description
<b>groupName</b>	String containing the name of the group
<b>groupInstance</b>	Group element to be filled
<b>callback</b>	String containing the name of the function to be called when the group is ready

### Return value

A number value that is the status: 1 for success, 0 for fail.

```
var group = new Group();
var status = project.getGroup ("enginesettings", group);
if (status == 1) {
    var value = group.getTag("Tag1");
    if (value!=undefined) {
        // do something with the value
    }
}
```

```
var g = new Group();
var status = project.getGroup ("enginesettings", g,
    function (groupName, group) { fnGroupReady(groupName, group);} );

function fnGroupReady(groupName, group) {
    var val = group.getTag("Tag1");
    if (val!=undefined) {
        // do something with the value
    }
}
```

(Available on web pages)

## getTag

```
object getTag( tagName, state, index, forceRefresh)
```

```
void getTag( tagName, state, index, callback, forceRefresh)
```

It returns the tag value or the complete array if index value is -1 of the given tagName.

Parameter	Description
<b>tagName</b>	String of tag name
<b>state</b>	State element to be filled
<b>index</b>	Index if the tag is of array type. -1 returns the complete array. Default = 0.
<b>callback</b>	Function name if an asynchronous read is required. Default = "".
<b>forceRefresh</b>	(Optional parameter) True = the Runtime will read an updated value of the tag directly from the device. Default is false.

### Return value

Tags value is returned. If tag is array type and index = -1 then the complete array is returned. For non-array tags provide index as 0.

```
var state = new State();
var value = project.getTag("Tag1", state, 0);
//
//for non array type
//tags index is not considered, so can be left as 0
//
if (value!=undefined) {
//...do something with s
}
```

```
var state = new State();
project.getTag("Tag1", state, -1,
    function(tagName, tagState) { fnTagReady(tagName, tagState); });
function fnTagReady(tagName, tagState) {
    if (tagName=="Tag1") {
        var myValue = tagState.getValue();
    }
}
```

(Available on web pages)

## setTag

```
number setTag( tagName, tagValue, [index], [forceWrite] )
```

Sets the given tag in the project. Name and value are in strings.



Parameter	Description
<b>tagName</b>	String of tag name
<b>tagValue</b>	Object containing the value to write
<b>index</b>	Index if the tag is of array type. -1 pass the complete array. Default = 0.
<b>forceWrite</b>	Boolean value for enabling force write of tags, the function will wait for the value to be written before it returns back. Default = false.

### Return value

Integer value for denoting success and failure of action when forceWrite is true. 0 means success and -1 means failure. If forceWrite is false, returned value will be undefined.

```
var val = [1,2,3,4,5];
var status = project.setTag("Tag1", val, -1, true);
if (status == 0) {
    // Success
} else {
    // Failure
}
```

```
var val = "value";
project.setTag("Tag1", val);
```

(Available on web pages)

## updateSystemVariables

```
void project.updateSystemVariables()
```

Force system variables to refresh.

```
project.updateSystemVariables()
```

## selectAllAlarms

```
void project.selectAllAlarms(bool selected)
```

Select/unselect all alarms

```
project.selectAllAlarms(true)
```

(Available on web pages)

## ackAlarms

```
void project.ackAlarms()
```

Acknowledge all selected alarms

```
project.selectAllAlarms(true);
project.ackAlarms();
project.selectAllAlarms(true);
```

(Available on web pages)

## resetAlarms

void project.resetAlarms()

Reset all selected alarms

```
project.selectAllAlarms(true);
project.resetAlarms();
project.selectAllAlarms(true);
```

(Available on web pages)

## enableAlarms

void project.enableAlarms()

Enable all selected alarms

```
project.selectAllAlarms(true);
project.enableAlarms();
project.selectAllAlarms(true);
```

(Available on web pages)

## getRecipeItem

object getRecipeItem (recipeName, recipeSet, recipeElement)

Gets the value of the given recipe set element.

Parameter	Description
<b>recipeName</b>	String representing the recipe name
<b>recipeSet</b>	String representing the recipe set, can be either the recipe set name or 0 based set index.
<b>recipeElement</b>	String representing the recipe Element, can be either the element name or 0 based element index.

### Return value

An object with the value of the recipe. undefined is returned if invalid. If of type array, an array object type is returned.

```
var value = project.getRecipeItem("recipeName", "Set", "Element");
```

## setRecipeItem

```
number setRecipeItem (recipeName, recipeSet, recipeElement, value )
```

Gets the value of the given recipe set element.

Parameter	Description
<b>recipeName</b>	String representing the recipe name
<b>recipeSet</b>	String representing the recipe set, can be either the recipe set name or 0 based set index.
<b>recipeElement</b>	String representing the recipe Element, can be either the element name or 0 based element index.
<b>value</b>	An object containing the value to store in the recipe. It can be an array type.

### Return value

Integer value for denoting success and failure of action. A '0' means success and '-1' means failure.

```
var val = [2,3,4];
project.setRecipeItem("recipeName", "Set", "Element", val);
if (status == 0) {
    // Success
} else {
    // Failure
}
```

## downloadRecipe

```
void downloadRecipe (recipeName, recipeSet )
```

Downloads the recipe set to the corresponding tag.

Parameter	Description
<b>recipeName</b>	String representing the recipe name
<b>recipeSet</b>	String representing the recipe set, can be either the recipe set name or 0 based set index.

```
project.downloadRecipe("recipeName", "Set");
```

## uploadRecipe

```
void uploadRecipe (recipeName, recipeSet )
```

Uploads the value of tags into the provided recipe set.

Parameter	Description
<b>recipeName</b>	String representing the recipe name
<b>recipeSet</b>	String representing the recipe set, can be either the recipe set name or 0 based set index.

```
project.uploadRecipe("recipeName", "Set");
```

## launchApp

```
void launchApp( appName, appPath, arguments, singleInstance)
```

Executes an external application.

Parameter	Description
<b>appName</b>	String containing the application name
<b>appPath</b>	String containing the application absolute path
<b>Arguments</b>	String containing the arguments to be sent to application
<b>singleInstance</b>	true = only single instance allowed, false = multiple instances allowed

Note the pathname's syntax depend from the native OS format (see ["HMI devices capabilities" on page 567](#)).

On **WinCE devices**, the pathname's syntax need double backslash character.

```
project.launchApp("PDF.exe", "\\Flash\\QTHMI\\PDF", "\\USBMemory\\file.pdf", "true");
```

On **Linux devices**, the pathname's syntax need slash character (even double slash character is permitted).

```
project.launchApp
("pdfViewer", "/mnt/data/hmi/qthmi/deploy", "/mnt/usbmemory/test.pdf", "true");
```

## getClientType

```
string getClientType()
```

Return the client type

Client Type	Description
local	Running on HMI device
remote	Running on HMI Client client
web	Running on Web client

```
var clientType = project.getClientType();
if (clientType=="web") {
    // Currently running on web client
} else if (clientType=="remote") {
    // Currently running on HMI Client
} else if (clientType=="local") {
    // Currently running on HMI Device
}
```

(Available on web pages)

## login

```
int project.login("username", "password")
```

Access to the system with the given credentials

```
var ReplyCode;
ReplyCode = project.login("admin", "admin");
if (ReplyCode != 0) {
    alert("Access denied");
}
```

### Return value

<b>0</b>	No Error
<b>1</b>	Error: You are not authorized.
<b>2</b>	Error: Connection lost with the Runtime.
<b>3</b>	Error: The username or password you entered is incorrect
<b>4</b>	Error: The password entered is incorrect
<b>5</b>	Error: Action cannot be executed
<b>6</b>	Error: Passwords do not match
<b>7</b>	Error: Password length too short
<b>8</b>	Error: Password must contain numbers
<b>9</b>	Error: Password must contain special characters
<b>10</b>	Error: Password must be different than previous passwords
<b>11</b>	Error : User already exist
<b>12</b>	Error: Password cannot be empty
<b>13</b>	Error: Your password has expired
<b>14</b>	Warning: Your password will expire soon

## logout

```
project.logout(AllowDefaultUser)
```

Exiting the system

```
project.logout();           // Logout even from default user
project.logout(true);       // Logout even from default user
project.logout(false);      // Logout only if not logged as default user
```

# Project object widgets

## getCurrentPageName

string getCurrentPageName()

Return the name of current active page

```
// Get PageMgr widget
var pageMgr = project.getWidget( "_PageMgr" );

// Show Current Page
var currentPageName = pageMgr.getCurrentPageName();
project.showMessage( "Current active page is: " + currentPageName );
```

(Available on web pages)

## hasPage

boolean hasPage(string pageName)

Return true if the page exist, false otherwise

```
// Get PageMgr widget
var pageMgr = project.getWidget( "_PageMgr" );

//Page exists
var pageExists = pageMgr.hasPage( "Page10" );
if (pageExists) {
    project.showMessage( "Page10 exists" );
} else {
    project.showMessage( "Hei Page10 not exists!" );
}
```

(Available on web pages)

## curLangCode

string curLangCode

Property of MultiLangMgr widget. Contains the code of the active language.

```
// Get MultiLangMgr widget
var MultiLangMgr = project.getWidget( "_MultiLangMgr" );

// Show curLangCode
var curLangCode = MultiLangMgr.curLangCode;
project.showMessage( "Current active language is: " + curLangCode );
```

## Print reports object

### printGfxReport

```
void printGfxReport( reportName, silentMode)
```

Prints the graphic report specified by reportName.

Parameter	Description
<b>reportName</b>	String containing the report name
<b>silentMode</b>	True = silent mode enabled. No printer settings dialog is displayed.

```
project.printGfxReport("Report Graphics 1", true);
```

### printText

```
void printText( text, silentMode)
```

Prints a fixed text.

Parameter	Description
<b>text</b>	String to print
<b>silentMode</b>	True = silent mode enabled. No printer settings dialog is displayed.

```
project.printText("Hello I Am Text Printing",true);
```

### printBytes

```
void printBytes( text, silentMode)
```

Prints a hexadecimal string representing data to print. For example, "1b30" to print < ESC 0 >

Parameter	Description
<b>text</b>	Hexadecimal string to print
<b>silentMode</b>	True = silent mode enabled. No printer settings dialog is displayed.

```
project.printText("Hello I Am Text Printing",true);
```

### 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.prinRAMQuota;  
project.setTag("prinRAMQuota",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);
```

# State object

This is the class holding the state of a tag acquired from the controlled environment.

## State object methods

Methods to be used with state objects.

### getQualityBits

```
number getQualityBits()
```

Returns an integer - a combination of bits indicating tag value quality.

```
var state = new State();
var value = project.getTag("Tag1", state, 0);
var qbits = state.getQualityBits();
```

(Available on web pages)

## getTimestamp

number getTimestamp()

Returns time the value was sampled.

### Return value

A number containing the timestamp (for example 1315570524492).



Note: Date is a native JavaScript data type.

```
var state = new State();
var value = project.getTag("Tag1", state, 0);
var ts = state.getTimestamp();
```

## isQualityGood

boolean isQualityGood()

Returns whether the value contained in this state object is reliable.

### Return value

A Boolean true if quality is good, false otherwise.

```
var state = new State();
var value = project.getTag("Tag1", state, 0);
if (state.isQualityGood()) {
    // do something...
}
```

(Available on web pages)

## Keywords

Global objects are predefined and can be referenced by the following names.

### page

object page

References the page object for the current page.

```
function btnStd04_onMouseRelease(me) {
    var wgt = page.getWidget("rect1");
    var name = wgt.objectName;
}
```

## project

object project

References the project widget.

```
var group = new Group();
project.getGroup("GroupName", group);
var value = group.getCount("Tag1");
```

# Global functions

## print

void print( message )

Prints a message to the HMI Logger window.

Parameter	Description
<b>message</b>	Message string

```
print("Test message");
```

## alert

void alert( message )

Displays a pop-up dialog with the given message. The user must press the **OK** button in the dialog to continue with the execution of the script.

Parameter	Description
<b>message</b>	Message string



Note: The alert function may be used for debugging JavaScript functions.

```
alert("Test message");
```

(Available on web pages)

# Handling read/write files

## Create folder

boolean fs.mkdir(strPath);

Creates a folder, if not already existing, in the specified path. Returns true on success and false if it fails.

Parameter	Description
<b>strPath</b>	Path string

## Remove folder

```
boolean fs.rmdir(dirPath);
```

Remove directory at strPath if exists and empty. Returns true on success and false if it fails.

Parameter	Description
<b>dirPath</b>	Folder string

## Read folder content

```
object fs.readdir(dirPath);
```

Reads the contents of a folder. Returns an array of the names of the files in the folder excluding '.' and '..'. Returns empty list if it fails.

Parameter	Description
<b>dirPath</b>	Folder string

## Read file

```
object fs.readFile(strfile [,strFlag]);
```

Opens the strFile file in read mode, reads its contents and returns it.

Parameter	Description
<b>strFile</b>	File name string
<b>strFlag</b>	Read file mode: "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
<b>strFile</b>	File name string
<b>fileData</b>	Data to be write on the file in byte array
<b>strFlag</b>	Write file mode: <ul style="list-style-type: none"> <li>"a": appends fileData to the end of the text file</li> </ul>

Parameter	Description
	<ul style="list-style-type: none"> <li>“r”: replaces the contents of the file with fileData</li> <li>“ab”: appends fileData to the end of the binary file</li> <li>“rb”: replaces the contents of the binary file with fileData</li> </ul>

Default flag is for writing text file in append and write mode. File path will be created if not present.

Returns -1 if write error occurs.

## Append file

```
int fs.appendFile(strFile, fileData);
```

If the files does not exist creates it, otherwise append to existing file. Returns the number of character written or -1 on error.

Parameter	Description
<b>strFile</b>	File name string
<b>fileData</b>	Data to be write on the file in byte array

## File exists

```
boolean fs.exists(strPath)
```

Returns true if the file or folder exists at strPath.

Parameter	Description
<b>strPath</b>	Path string

## Remove file

```
boolean fs.unlink(strPath)
```

Removes the given file at strPath from filesystem if exists. Returns true on success and false if it fails.

Parameter	Description
<b>strPath</b>	Path string

## File status

```
object fs.stat(strPath)
```

Retrieves information on the file/folder present at the specified path.

Parameter	Description
<b>strPath</b>	File/folder path string

```
var fileStats = var fs.stat(strPath)
```

<code>fileStats.isFile</code>	True if path is a file
<code>fileStats.isDir</code>	True if path is a folder
<code>fileStats.size</code>	Size in bytes of that file
<code>fileStats.atime</code>	Date object representing the last read access time
<code>fileStats.mtime</code>	Date object representing the last write access time
<code>fileStats.ctime</code>	Date object representing the creation time
<code>fileStats.perm</code>	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

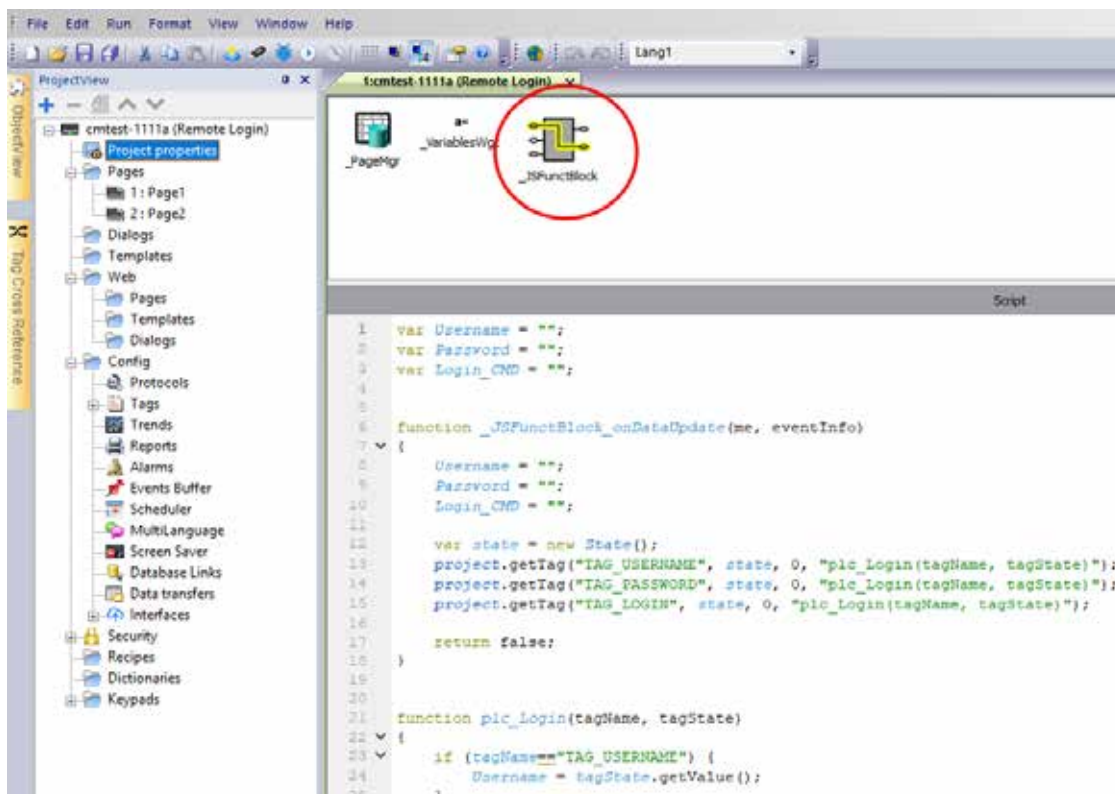
The TAG\_LOGIN will be the command code to execute.

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.





### 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 481](#)

## 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

JMobile Studio and JMobile 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 JMobile Studio)

## Enabling debugging

In the **Properties** pane of a page, set **JavaScript Debug** to **true**.

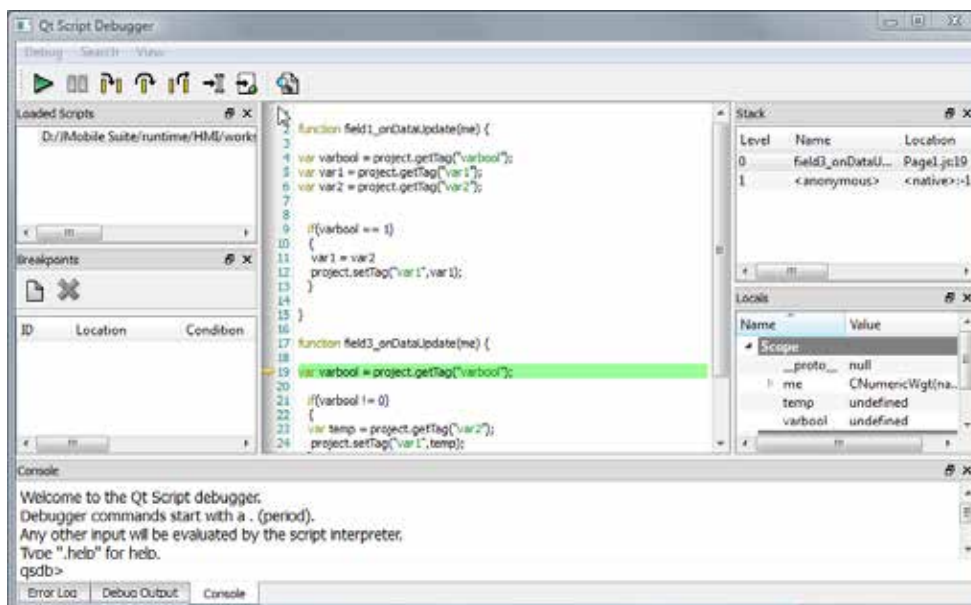
Project Widget	
Id	Project
Full Path	
Version	
Context Menu	on delay
Developer Tools	false
Keyboard	true
JavaScript Debug	true
Allow JavaScript Remote	true

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 JMobile HMI Runtime, when the events are called, the debugger will show the debug information. In the **Locals** pane you can inspect all variables and elements.



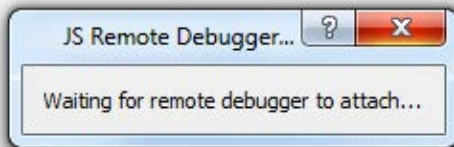
For a complete reference guide about JavaScript Debugger refer to :

<http://qt-project.org/doc/qt-4.8/qtscripdebugger-manual.html>

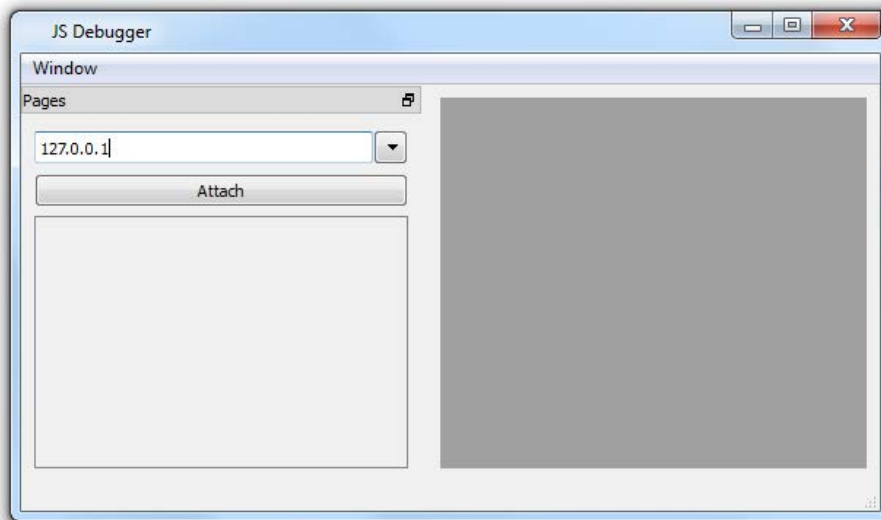
## Remote JavaScript Debugger

Path: **Run> Start JS Remote Debugger**

1. Set the **Allow JavaScript Remote** and the **JavaScript Debug** parameters in the project Properties to true in all the pages where debugging is required.
2. Download the project: the following message is displayed on the runtime.



3. In the **JS Debugger** window, select the IP of the HMI device and click **Attach** to connect the debugger to the HMI device.



Remote JavaScript debugger connects to JMobile HMI Runtime using port 5100/TCP.



Note: The Remote JavaScript debugger tool is not supported in HMI Client.

## JavaScript Memory Usage

When the memory exceeds the maximum, an out of memory exception is thrown with a custom message. Please note that we don't have a fine control over the actual memory usage so it is mainly a soft limit. Moreover we can't forbid the allocation (this will break the engine implementation), so exception is thrown only when the memory is already over the limit. Before raising the exception, a garbage collection is forced to see if some memory can be freed.

JavaScript memory limit can be accessed from the global object **\$EngineMemory**. The default is 16MB, which should be enough for the typical JavaScript usage (mainly control, without many allocations).

- `$EngineMemory.setLimit()`  
set maximum memory allowed for JavaScript (the default limit is 0x00FFFFFF)
- `$EngineMemory.getLimit()`  
get maximum memory allowed for JavaScript
- `$EngineMemory.getSize()`  
get currently used memory from JS (fastMallocStat)

### Test memory exception

To generate and test memory exception you can use the following snippet. Please note that we need to reset the memory limit to 0xffffffff to be able to run the alert, otherwise the memory allocations required to pop up the alert would fail.

```
try
{
    // Generate out-of-memory error
    var a = [];
    while(1)
    {
        a.push("a");
    };
} catch(e)
{
    // Ensure there is enough memory to pop up error message
    $EngineMemory.setLimit(0xffffffff);
    alert("Exception: " + e);
};
```

## 39 Handling Gestures

Some widgets have the capability to detect and manage pan and pinch gestures.

- Trends (see ["Trend widget gestures" on page 260](#) 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 400](#) for details)

For widgets based on table presentation, when the **Scrollbars Type** parameter has been set to "Gesture", the pan gesture is used to smoothly scroll the table.

- Alarms
- Control List



**WARNING: Pinch and Rotate gesture requires two fingers. They are available only with HMI devices supporting multi touch operation (see ["HMI devices capabilities" on page 567](#))**



**Tip:** Using multi touch HMI device you can implement safe commands by programming a command to be executed only when two buttons are pressed at the same time.



PAN



PINCH



ROTATE



# 40 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 JMobile Studio (see "[Updating system components in HMI devices](#)" on page 549 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.*

---

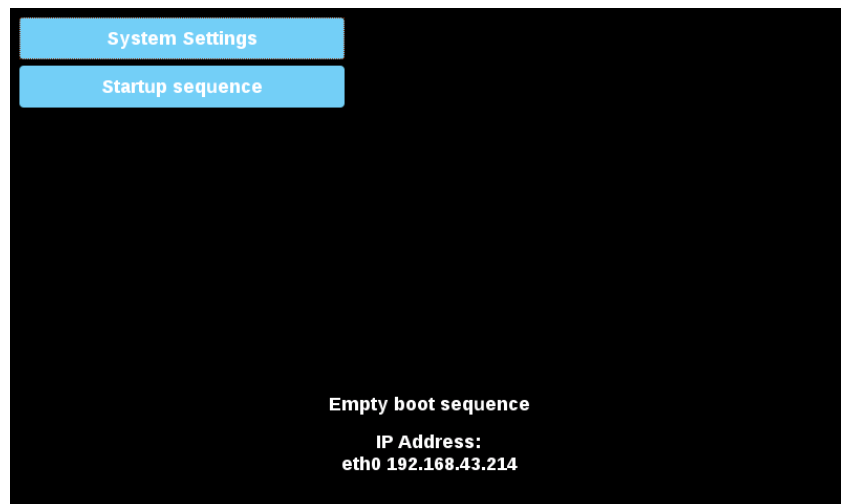
Linux Devices .....	500
WinCE Devices .....	524

# Linux Devices

Linux 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.



Runtime can be installed:

- Automatically, via Ethernet on first project download with JMobile Studio
- Manually via USB Memory, creating an “Update Package”. (See the ["Update package" on page 93](#) to create a runtime package)

## Install Runtime via Ethernet

To install Runtime via Ethernet follow the ["Download to HMI device" on page 90](#) 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:

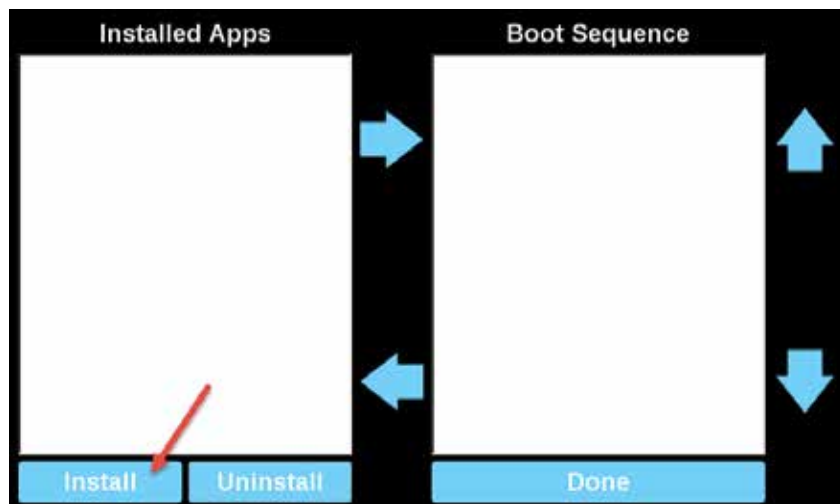
1. Create an Update Package from JMobile Studio and copy into an empty USB memory stick



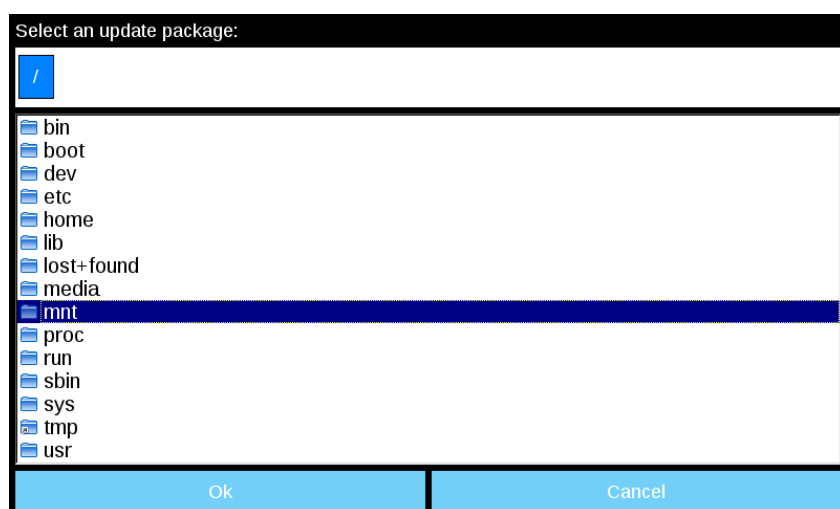


Note: File systems supported are FAT16/32 and Linux Ext2, Ext3 and Ext4.

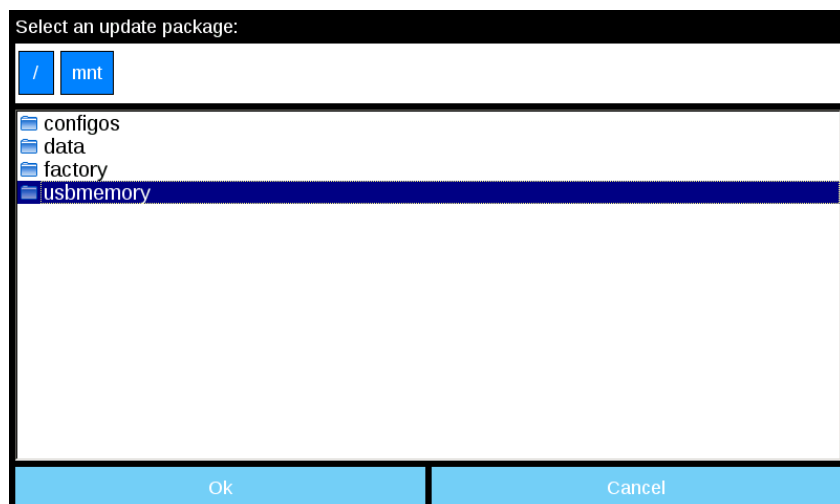
2. On HMI select [Startup sequence], then [Install]



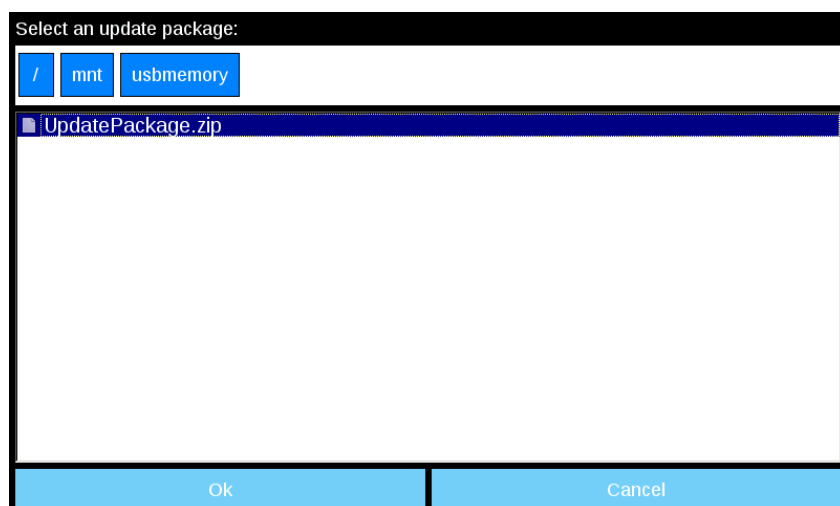
3. Double click on “mnt” to access this folder



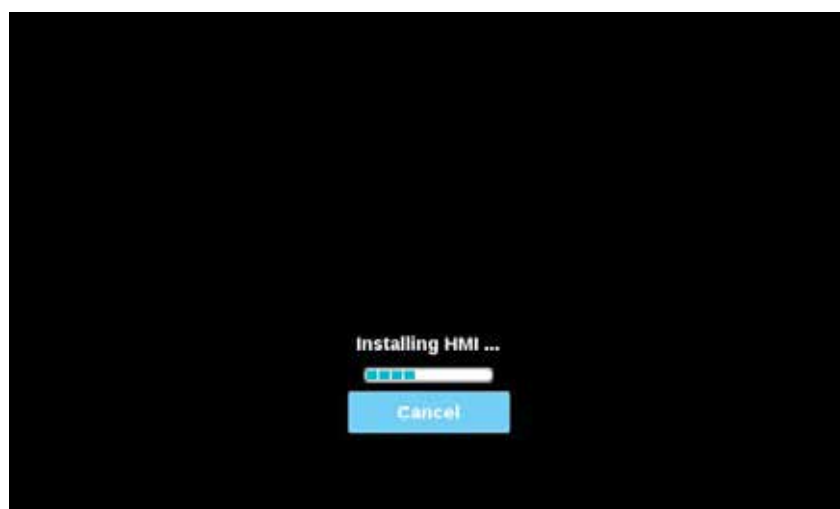
4. Then on “usbmemory”



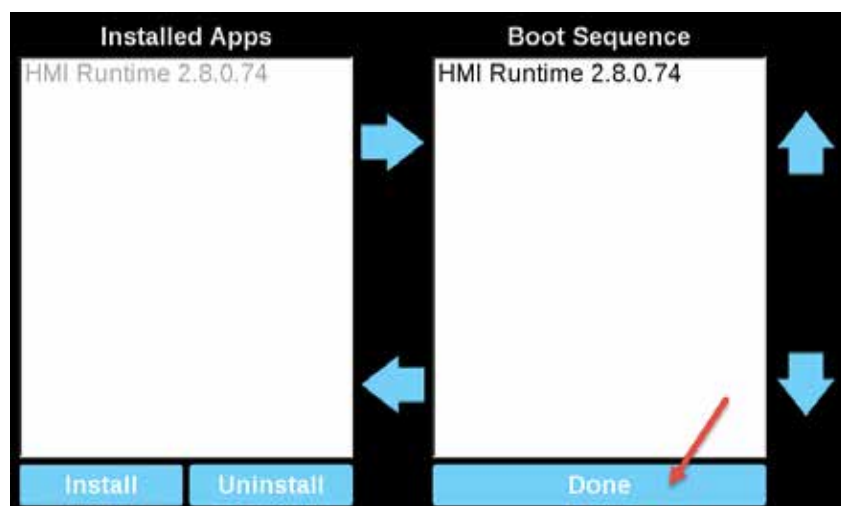
5. Select "UpdatePackage.zip" and confirm with [Ok]



6. The runtime installation begin



7. At the end press "Done" button



8. Then "Start HMI" button



## 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.

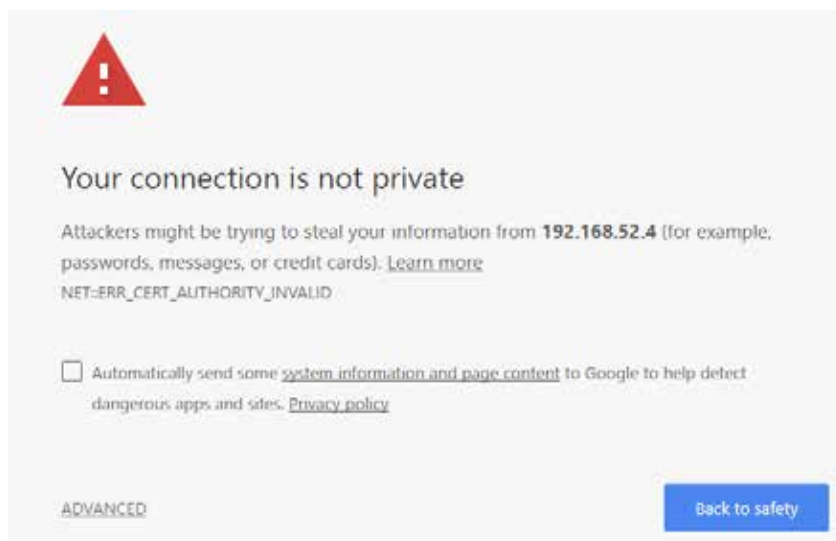
### System Setting access from Web browser

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.



Browse through the options available in the menu on the left: the active item is highlighted and related information is displayed on the right.



Default security protocols proposed by the HTTPS server in the Linux HMI device are:

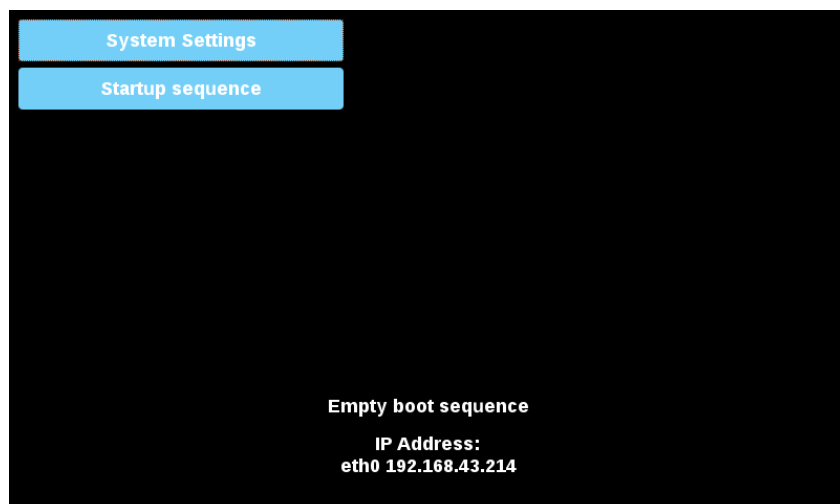
- SSLv3 256 bits ECDHE-RSA-AES256-SHA
- TLSv1 256 bits ECDHE-RSA-AES256-SHA



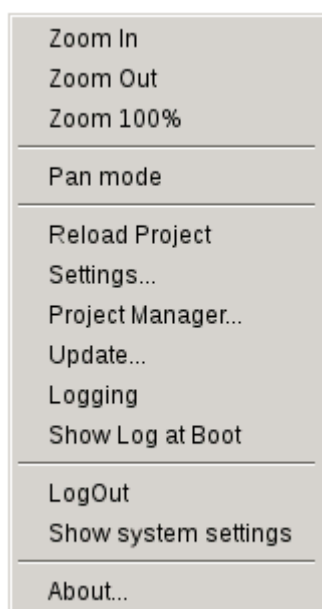
**WARNING: We discourage usage of CBC cyber suites in the context of SSL3 or TLSv1.0 connections since potentially affected by some vulnerabilities.**

### System Setting access from HMI device

When Runtime is not installed, the System Settings is accessible from the Runtime Loader screen,



When Runtime is installed the System Settings is accessible selecting “Show System Settings” option of Context Menu,



### Enter System Settings via tap-tap procedure

Tap-tap consists in a sequence of several touch activations by simple means of the finger tapping the touch screen performed during the power-up phase and started immediately after the HMI is powered on.



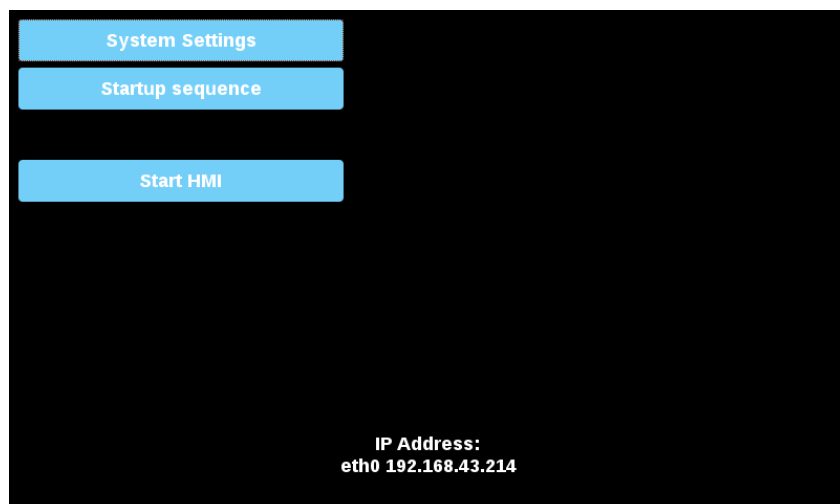
When “tap-tap detected” message appears on the top of the screen. Wait for 5 seconds (without touching the screen) to enter System Settings sub menu



Wait for 5 more seconds (without touching the screen) to enter Default Mode



Select "System Setting" from the HMI Default Mode screen



### System Settings Sections

To change system settings values, enter in edit mode by click the edit button on the right top.



The edit button is available only inside the dialogs that contains modifiable parameters.

## 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
<b>Current Timezone</b>	Timezone region
<b>Current Date Local Time</b>	Date and Time can set manually only when the Automatic Update is disabled.
<b>Automatic Update (NTP)</b>	<p>Enable to keep date and time synchronized from a remote server</p> <ul style="list-style-type: none"> <li>NTP Server Specify the Internet NTP Server address</li> </ul> <p> The NTP Client of the HMI Device is a complete implementation of the Network Time Protocol (NTP) version 4, but also retains compatibility with version 3, as defined by RFC-1305, and version 1 and 2, as defined by RFC-1059 and RFC-1119, respectively</p>

## Networks

Network parameters. Available parameter in edit mode:

Parameter	Description
<b>General Settings</b>	<p>Device hostname</p> <p>Avahi Hostname (see "<a href="#">Avahi Daemon</a>" below)</p>
<b>Network Interface</b>	<p>Network parameters of the available interfaces</p> <ul style="list-style-type: none"> <li>DHCP</li> <li>IP Address</li> <li>Net Mask</li> <li>Gateway</li> </ul>
<b>DNS</b>	<p>DNS Servers Generally provided from the DHCP servers, but can be modified in edit mode</p> <p>Search Domains Optional domains that will be used in concatenation with the provided urls</p>

## 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.

### 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

Network Interfaces

Name	Label	MAC	DHCP	Address	Netmask	Gateway	Bridged
eth0	WAN	00:30:d8:04:63:63	Enabled	192.168.57.98	255.255.0.0	192.168.20.1	
eth1	LAN1	00:30:d8:04:63:64	Enabled				
eth2	LAN2		Enabled				
tap0	VPN		Disabled				

Download to Target

Ready to download

myDevice.local

Download

Close

+ Advanced

Avahi Daemon runs on UDP port 5353

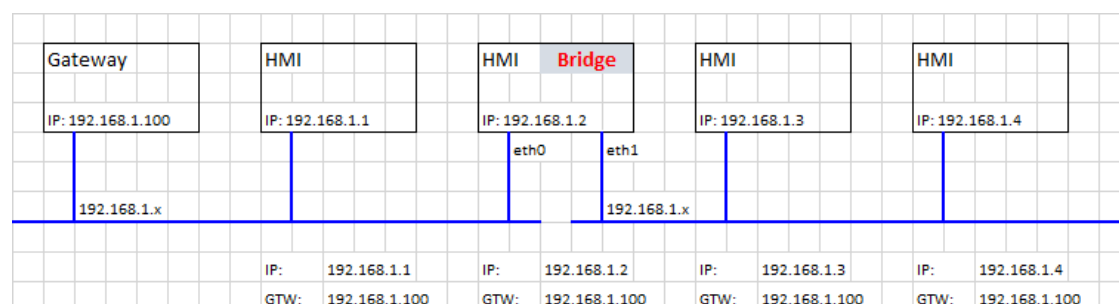


On Linux and Apple PCs, the Avahi service comes for free with the OS. On Windows PCs instead, you need to install an Avahi service to be able to reach the panel by his Avahi host name (e.g. you need to install the Apple Bonjour application - Bonjour is a trademark of Apple inc.).

## Bridge Service

Using the bridge service is possible connect together the WAN (eth0) network adapter with the LAN (eth1 or eth2) network adapters. When used, the two Ethernet adapters are bridged and both Ethernet adapters 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 (Note: Cloud scenario still requires Router Service to be active)



## Cloud Service

Allow to manage remote HMI devices connected to a centralized server through gateways.

See "Cloud Service" on page 514 for additional details.

## 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

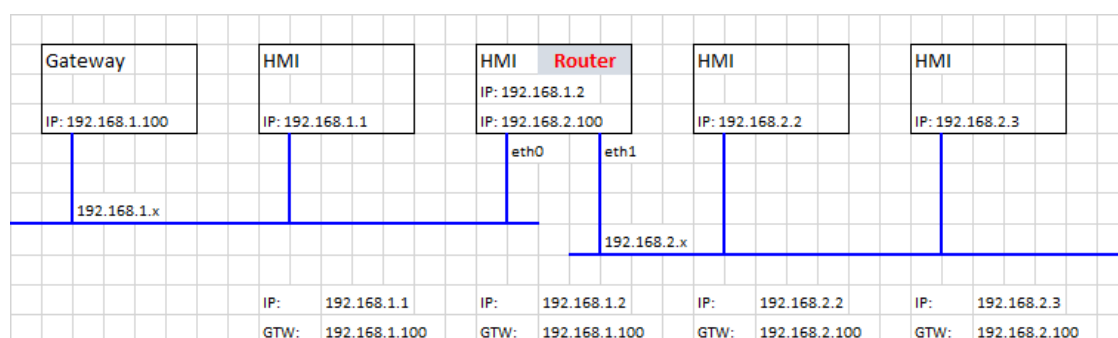


**Required BSP v1.0.242 or greater**

## Router Service

This service uses the IP Forwarding and the Network Address Translation to sharing the connections from WAN (eth0) towards LAN (eth1 or eth2): the connected endpoints may reach the same networks reachable by the gateway (including Internet if available)

With Cloud Service active, endpoints can be reached via the gateway's LAN port (please refer to ["Cloud Service" on page 514](#) for more information)



## 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.

**iReasoning MIB Browser**

Address: 192.168.57.98 | Advanced... | OID: .1.3.6.1.2.1.1.5.0 | Operations: Get | Go

**SNMP MIBs**

- iso.org.dod.internet
  - mgmt
    - mib-2
      - system
        - sysDescr
        - sysObjectID
        - sysUpTime
        - sysContact
        - sysName**
        - sysLocation
        - sysServices
      - interfaces
        - at
        - ip
        - icmp

**Result Table**

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

**sysName Details**

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



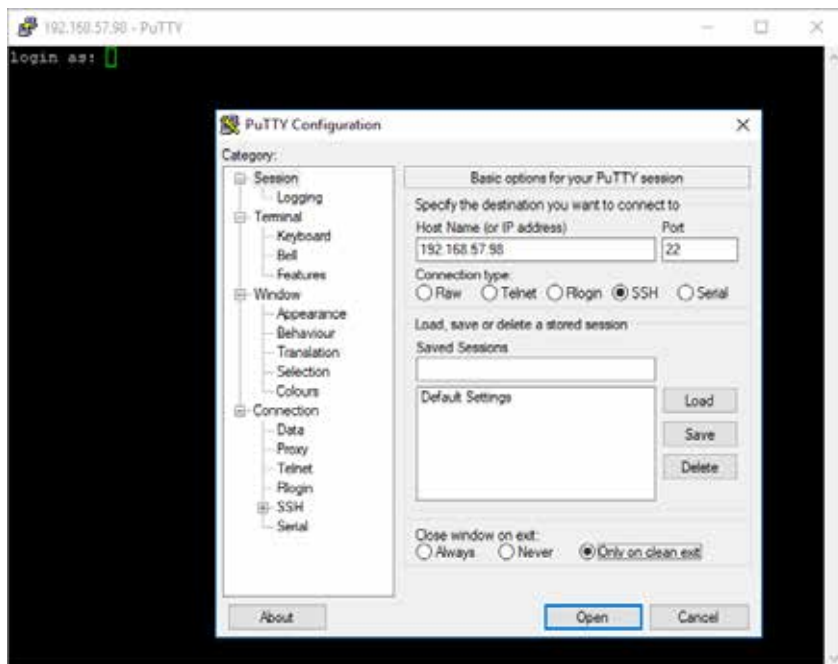
**This service is designed to be used during the development phase. For security reasons, remember to disable the service before switch to production.**

## 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 513](#) chapter to additional information.



SSH Server runs on TCP port 22



**This service is designed to be used during the development phase. For security reasons, remember to disable the service before switch to production.**

## VNC Service

VNC is a service that allows remote access to the display of the HMI device. VNC clients can be used to get the remote control of the HMI device.

- 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: don't 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 page 517 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 through 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
State	The state/region where your organization is located

Parameter	Description
<b>Location</b>	The city where your organization is located
<b>Country</b>	The two-letter ISO code for the country where your organization is location
<b>Valid (days)</b>	Validity of the certificate
<b>Key Length</b>	Number of bits of the key used from the cryptographic algorithm

Managed certificates are base64 encoding



**Required BSP v1.0.239 or greater**

## EXIT

Exit from the System Setting tool.

## Cloud Service

**Cloud Service** allows devices to connect to a Cloud Server, making them securely available to remote clients.



**BSP v1.0.117 or greater is required**

### Prerequisites

This service requires external access to the server for VPN setup (default port UDP/1194) and for self-configuration/other advanced features on TCP port 443 (Cloud Server mode only), so please check configuration and make sure no firewalls block such ports.

### Setup

If you need endpoints behind your gateway device to be reached, make sure Router Service is active and set it up as follows:

- WAN port (eth0) connected to the main network with Internet access (Cloud Server must be reachable from this network)
- LAN port (eth1) connected to one or more endpoint devices (newly-created private network)

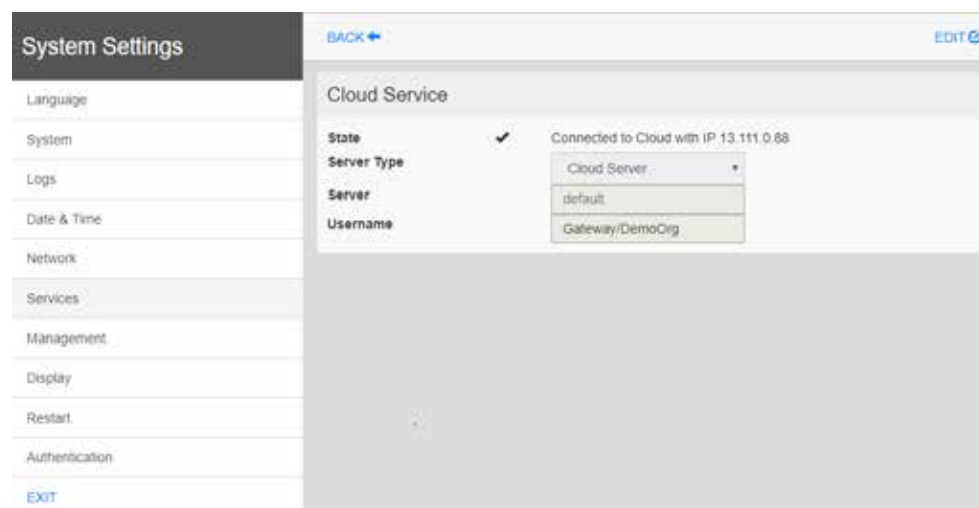


This functionality is automatically supported when using a Cloud Server, but will require extra manual setup for plain OpenVPN server.

### Configuration

Configuration options are available in the Services Menu of System Settings (they are visible only once the service is enabled). Two Server Types are currently available:

1. Cloud Server
2. OpenVPN



## Cloud Server

Cloud Server is a VPN-based solution that allows seamless connection of users with gateways and endpoints. It provides a full management infrastructure to make such process painfree.

Configuration is downloaded automatically from Cloud Server, so the only required parameters are Server (hostname or IP address), Username and Password.

## OpenVPN

This mode uses a standard OpenVPN configuration to connect devices.

### Case A: Configuration files provided

In remote access environments based on an OpenVPN server, system administrators normally supply a number of OpenVPN configuration files directly to end users.

In such case configuration is quite straight-forward since it requires only two simple steps:

1. browse and upload N files (this should include at least a main OpenVPN configuration file, but may also include server and/or client certificates in .pem, .p12 or other formats); make sure you select all necessary files in one shot by using platform-dependent multiselection;
2. select an appropriate Authentication type and insert credentials if they are required.

You're done! now press Save, wait a little while and you should see an updated connection status.

### Case B: No configuration files provided

If no configuration files have been provided by your system administrator, you will need to create the OpenVPN configuration file yourself.

### Sample 1: Username/Password

This sample uses:

- username/passsword-based authentication
- LZO compression and TAP device
- server running on UDP port 1194

*openvpn.conf*

```

client
dev tap
proto udp
remote testserver.whatever.com 1194
comp-lzo
ca cacert.pem
auth-user-pass

```

This configuration file only refers to one external file (*cacert.pem*), so:

1. upload the 2 files using the Browse option
2. insert your allocated Username and Password - note that the *auth-user-pass* option can also take a file argument, so you can even insert newline-separated username and password in a new file and specify its name here (not recommended); in such case you would select also your external file when browsing files and choose *None (from file)* Authentication method
3. Save and wait for State change

### Sample 2: Plain certificate

This sample uses:

- plain X509 certificate-based authentication
- LZO compression, TUN device, custom MTU and AES-128-CBC cipher
- server running on TCP port 1195

*openvpn.conf*

```

tls-client
dev tun
proto tcp
tun-mtu 1400
remote testserver.whatever.com 1195
pkcs12 mycert.p12
ca cacert.pem
cert client.pem
key client.key
cipher AES-128-CBC
comp-lzo
verb 4

```

This configuration refers to 3 files (*cacert.pem*, *client.pem*, *client.key*), so:

1. upload main *openvpn.conf* and external files (total 4), using the Browse option
2. since no passwords are required, choose *None (from file)* Authentication
3. Save and wait for State change

### Sample 3: Password-protected PKCS #12 certificate

This sample uses:

- certificate-based authentication (password-protected PKCS #12)
- other parameters same as Sample 2

*openvpn.conf*



```
[..]
pkcs12 mycert.p12
```

The PKCS #12 bundle normally contains both CA certificate client keypair, so this configuration file only refers to one external file (*mycert.p12*). Hence:

1. upload the 2 files using the Browse option
2. choose *Certificate* Authentication
3. insert the password which should be used to unencrypt the PKCS #12 bundle containing your certificate
4. Save and wait for State change

#### Sample 4: 2-factor authentication via password-protected PKCS #12 certificate + username/password

This sample uses:

- both certificate-based authentication (password-protected PKCS #12) and username/password
- other parameters same as Sample 2

*openvpn.conf*

```
[..]
pkcs12 mycert.p12
auth-user-pass
```

upload the 2 files using the Browse option

choose *Certificate + Username/Password* Authentication

insert *Username* and *Password* for PSK authentication

insert the *PKCS #12 Password*

Save and wait for State change

#### Links

Please refer to [OpenVPN documentation](#) for further details.

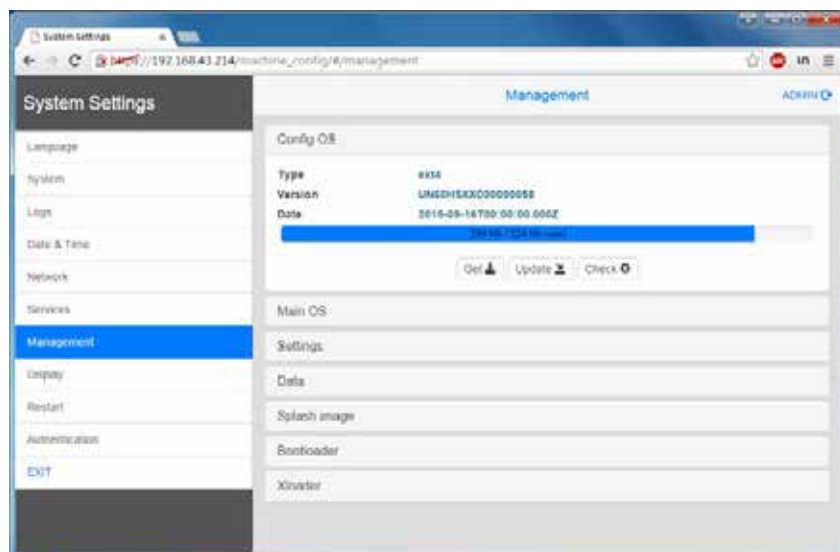
## Update System Components



**CAUTION:** Working in the Management area is a critical operation and, when not performed correctly, may cause product damages requiring service of the product. Contact technical support for assistance (the latest BSP files will be provided from tech support).

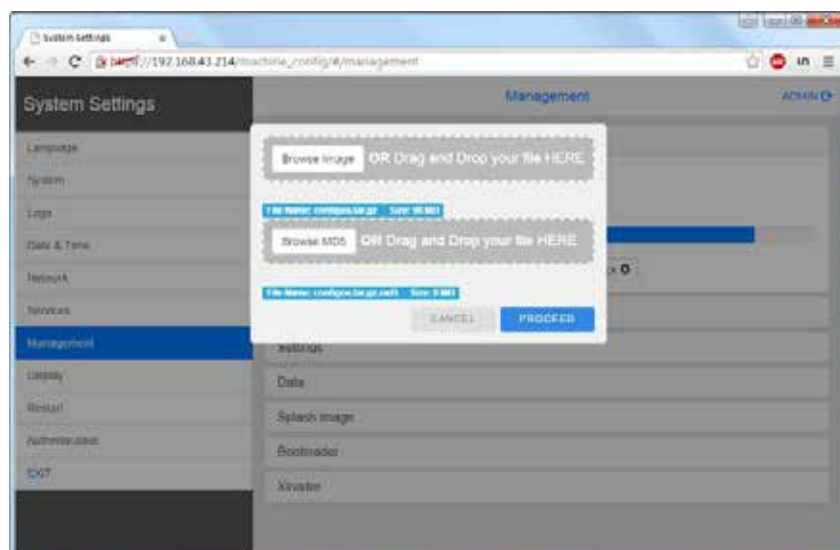
The system components of the Linux device can update locally using a USB memory key or remotely via web browser.

To update system components enter System Settings in Config OS mode via tap-tap procedure on HMI or open web browser to <https://<HMI-IP-address>> and select the "Management" section.



Expand the component to update and select [Update]

On the opened dialog, click [Browse Image], then select the “xxx-mainos-xxx.tar.gz” file. Click then on [Browse MD5] and select the “xxx-mainos-xxx.tar.gz.md5” file.



**Important:** Do not turn off the device while a system component is being upgraded.

At the end of the component update, restart HMI and leave it starting normally.

## Enter System Settings in Config OS mode via tap-tap procedure

System Setting in Config OS mode is available via tap-tap sequence, this mode can be accessed also when HMI is facing a software failure.

Tap-tap consist in a sequence of several touch activations by simple means of the finger tapping the touch screen performed during the power-up phase and started immediately after the HMI is powered on.



When “tap-tap detected” message appears on the top of the screen, press and hold the finger on touchscreen, to select “Restart: Config OS”



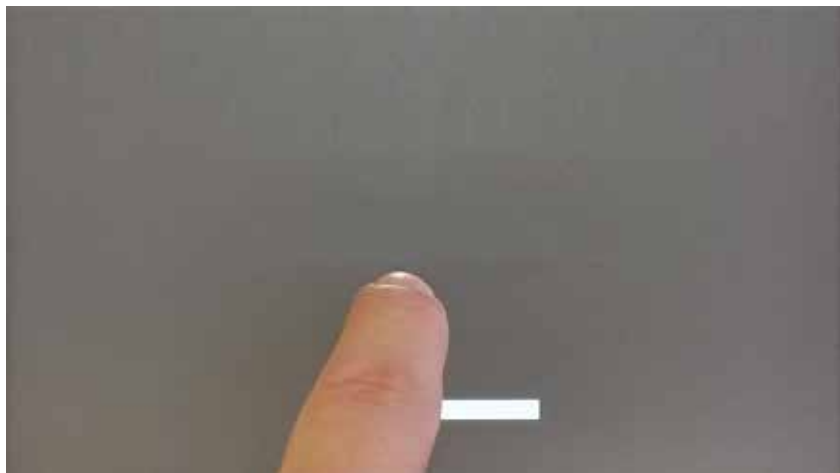
HMI will restart into System Settings in Config OS mode:



## Touchscreen calibration

System Setting Calibration allows to calibrate Touchscreen device, can be accessed by tap-tap procedure.

Tap-tap consists in a sequence of several touch activations by simple means of the finger tapping the touch screen performed during the power-up phase and started immediately after the HMI is powered on.



When “tap-tap detected” message appears on the top of the screen, wait for 5 seconds (without touching the screen) to enter System Settings sub menu



Press on touch screen, “Touchscreen calibration” voice will be highlighted in yellow, hold pressed for few seconds until touchscreen calibration procedure starts

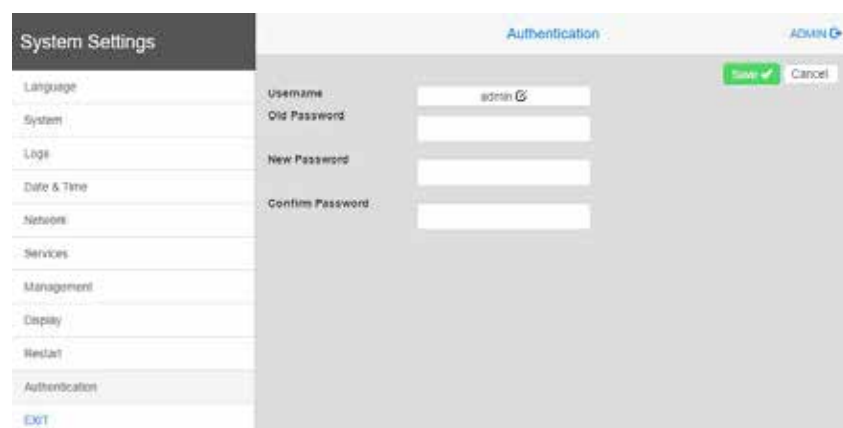


Follow the instructions on screen to complete the calibration procedure, system will prompt to touch specific points to calibrate the touchscreen device.

## Password protection

Internal password of the HMI device.

From the Authentication tab, inside the ["System Settings" on page 503](#), activate the edit mode and select the username to change the associated password.



Password for admin user can modified even from the context menu of the JMobile HMI Runtime (see ["Context menu options" on page 10](#) for details).

## 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



Wait for 5 more seconds (without touching the screen) to enter Default Mode

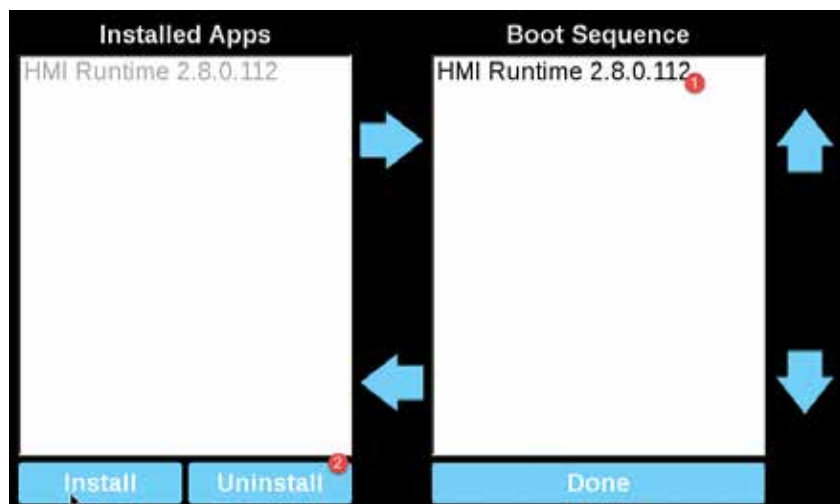


To uninstall the Runtime from HMI in Default Mode screen select [Startup Sequence]:

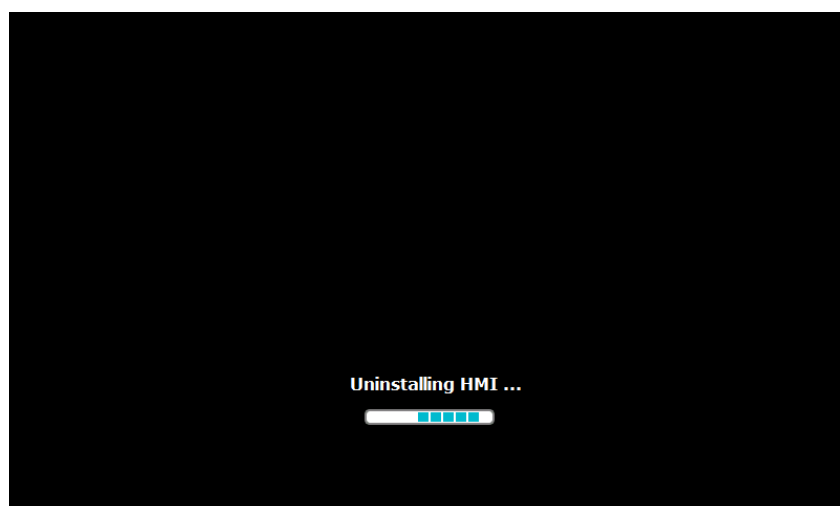


From the installed applications view:

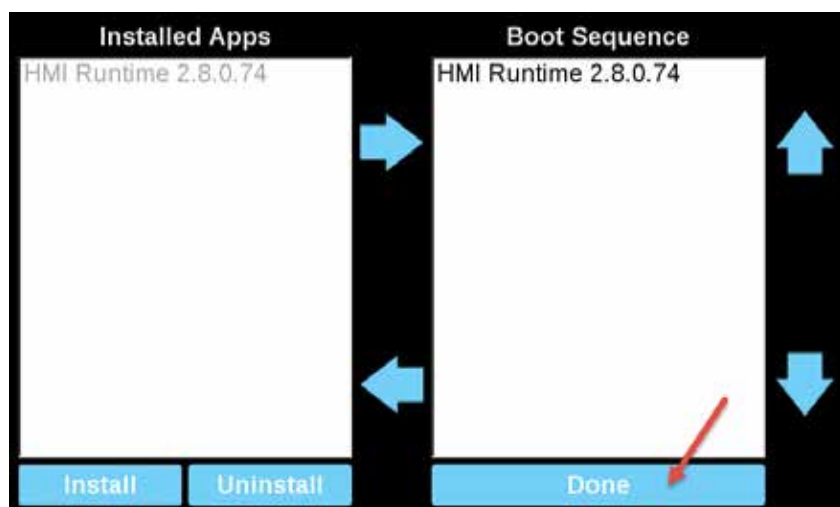
1. Select the Runtime you want to remove
2. Uninstall the selected Runtime



Runtime uninstall process will be performed:

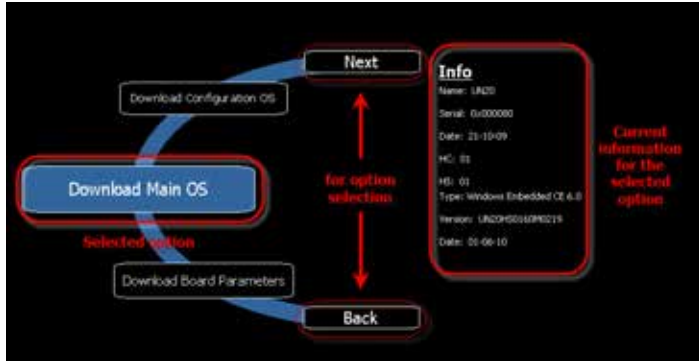


At the end press "Done" button



## WinCE Devices

The System Settings tool includes a rotating menu, and navigation buttons to scroll between the available options.



For each function and component on the left, the **Info** pane on the right displays all available information. In the example the version of the Main OS component is shown.

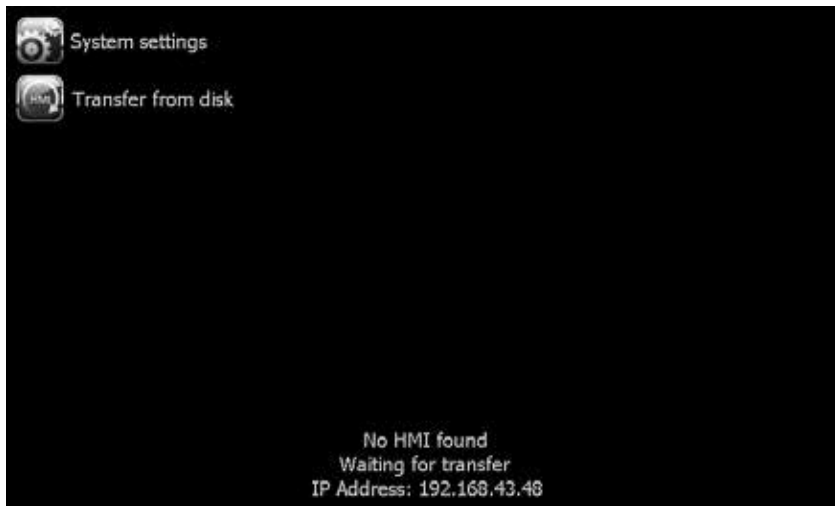
The System Settings tool can be used in two operating modes:

- User mode
- System mode.

For each mode different options are available.

## Runtime Installation

HMI devices are delivered from factory without Runtime, at first power up HMI shows the “Runtime Loader” screen.



Runtime can be installed:

- Automatically, via Ethernet on first project download with JMobile Studio
- 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 90](#) procedure.





**WARNING:** Runtime installation via Ethernet download requires the HMI to have a valid IP address.

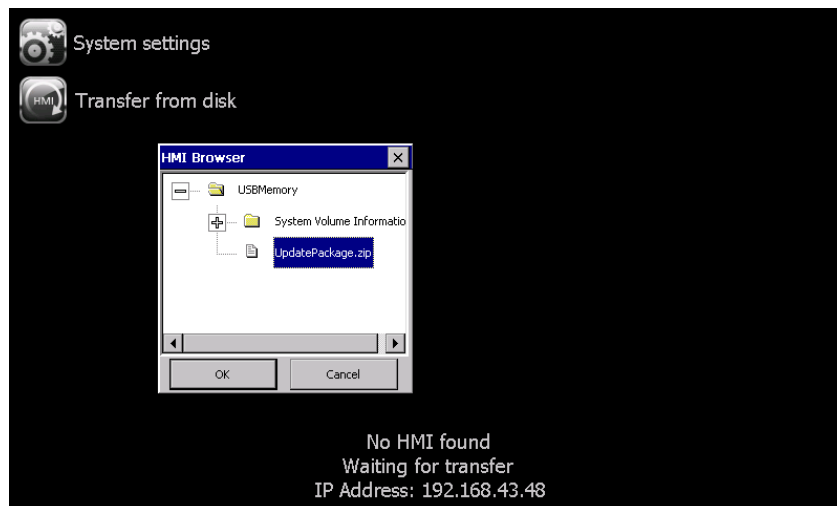
The IP address can be assigned in three ways:

- *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 JMobile Studio and copy into an empty USB memory stick
2. On HMI select [Transfer from disk] and select the UpdatePackage.zip to load.



## System Settings

System Settings has two operating modes:

- **User Mode**  
a simplified interface that gives users access to the basic settings of the HMI device.
- **System Mode**  
a full interface that gives users access to all the tool's options.

When you access the tool at runtime selecting "*Show system settings*" from the context menu, the tool is started by default in User Mode.





Note: Press and hold on a screen area without buttons or other touch sensitive elements to display the context menu.

To access System Mode:

- Execute a tap sequence on the touch screen during the power-up phase. A tap sequence is a high frequency sequence of touch activations executed immediately after the device has been powered.
- From the System Setting page in User Mode, restart the panel in Configuration OS mode

## Elements available in User Mode




Element	Description										
<b>Calibrate Touch</b>	Calibrate the touch screen										
<b>Display settings</b>	Control backlight inactivity timeout and brightness										
<b>Time</b>	Set HMI device date and time manually or configure NTP servers										
<b>Regional Settings</b>	Select or customize the regional setting parameters										
<b>BSP Settings</b>	Display operating system version and unit operating timers to control buzzer and battery led.										
<b>Network</b>	Sets IP address and other network settings										
<b>Plug-in List</b>	<p>List the plug-in modules installed and recognized by the system.</p> <p> Note: this option may not be supported by all platforms and all versions.</p>										
<b>Close</b>	Closes the system setting page										
<b>x.509 Certificate</b>	<p>You can personalize the certificate with the data of your Company and ask to a Certificate Authority to firm it.</p> <p>The procedure to personalize and firm your certificate is:</p> <ol style="list-style-type: none"> <li>1. Fill the necessary parameters, then push GENERATE button to generate a self-signed certificate with your data.</li> <li>2. Export the "Certificate Signed Request"</li> <li>3. Sent the "Certificate Signed Request" to a Certificate Authority to firm it (general this is a paid service)</li> <li>4. Import the signed certificate into the HMI device</li> </ol> <p><b>Certificate's parameters</b></p> <table> <tr> <th>Parameter</th><th>Description</th></tr> <tr> <td><b>Device Name</b></td><td>The name of your device</td></tr> <tr> <td><b>Organization</b></td><td>The legal name of your organization</td></tr> <tr> <td><b>Unit</b></td><td>The division of your organization handling the certificate</td></tr> <tr> <td><b>State</b></td><td>The state/region where your organization is located</td></tr> </table>	Parameter	Description	<b>Device Name</b>	The name of your device	<b>Organization</b>	The legal name of your organization	<b>Unit</b>	The division of your organization handling the certificate	<b>State</b>	The state/region where your organization is located
Parameter	Description										
<b>Device Name</b>	The name of your device										
<b>Organization</b>	The legal name of your organization										
<b>Unit</b>	The division of your organization handling the certificate										
<b>State</b>	The state/region where your organization is located										

Element	Description										
	<table> <tr> <th>Parameter</th><th>Description</th></tr> <tr> <td><b>Location</b></td><td>The city where your organization is located</td></tr> <tr> <td><b>Country</b></td><td>The two-letter ISO code for the country where your organization is location</td></tr> <tr> <td><b>Valid (days)</b></td><td>Validity of the certificate</td></tr> <tr> <td><b>Key Length</b></td><td>Number of bits of the key used from the cryptographic algorithm</td></tr> </table> <p>Managed certificates are base64 encoding</p> <p> <b>Required BSP v2.29 or greater</b></p>	Parameter	Description	<b>Location</b>	The city where your organization is located	<b>Country</b>	The two-letter ISO code for the country where your organization is location	<b>Valid (days)</b>	Validity of the certificate	<b>Key Length</b>	Number of bits of the key used from the cryptographic algorithm
Parameter	Description										
<b>Location</b>	The city where your organization is located										
<b>Country</b>	The two-letter ISO code for the country where your organization is location										
<b>Valid (days)</b>	Validity of the certificate										
<b>Key Length</b>	Number of bits of the key used from the cryptographic algorithm										
<b>Restart</b>	<p>Restart the HMI device</p> <ul style="list-style-type: none"> <li>• Main OS Restart the HMI device in the operating mode</li> <li>• Configuration OS Restart the HIM device with System Setting tool active in System Mode</li> </ul>										

## Elements available in System Mode

In addition to those available in User Mode, the following features are also available:

Element	Description
<b>Format Flash</b>	Formats the internal device flash disk. All projects and the JMobile HMI Runtime will be erased, returning the device to its factory settings.
<b>Restore Factory Settings</b>	<p>Restores factory settings as an alternative to Format Flash, in a more flexible way. The following options are available:</p> <p><b>Uninstall HMI:</b> removes the JMobile HMI Runtime (entire qthmi folder) at the next start the device will behave as a brand new unit. This command does not reset settings such as IP address, brightness or RTC.</p> <p><b>Clear System Settings:</b> resets system parameters (registry settings) and deletes the following files:</p> <p><i>\\Flash\\Documents and Settings\\system.hv</i></p> <p><i>\\Flash\\Documents and Settings\\default\\user.hv</i></p> <p><i>\\Flash\\Documents and Settings\\default.mky</i></p> <p><i>\\Flash\\Documents and Settings\\default.vol</i></p> <p>System Mode password is also reset.</p>

Element	Description
	<p><b>Clear Controller Application:</b> clears current folders used by CODESYS V2.3 and CODESYS V3 internal controllers for applications:</p> <ul style="list-style-type: none"> <li>• \Flash\QtHmi\RTS\APP\*. *</li> <li>• \Flash\QtHmi\RTS\VISU\*. *</li> <li>• \Flash\QtHmi\codesys\*</li> <li>• \Flash\\$SysData\$\codesys\*</li> </ul> <p><b>Clear sysdata settings:</b> clears \Flash\\$SysData\$ folder</p> <p> <i>Service call: To be used only by technical support to fix display problems.</i></p> <p> Note: Not all these options are available for all HMI devices and BSPs.</p>
<b>Resize Image Area</b>	Resizes the flash memory reserved to store the splash screen image displayed at power up. Default settings are normally suitable for all units.
<b>Download Configuration OS</b>	Checks and upgrades the current version of the operating system used in System Mode
<b>Download Main OS</b>	Checks and upgrades the current version of the main operating system
<b>Download Splash Image</b>	<p>Loads a new file for the splash screen image displayed by the unit at power up.</p> <p> Tip: Update the splash screen image directly from the JMobile Studio programming software.</p> <p>See "<a href="#">Update of system components from the application</a>" on page 550 for details.</p>
<b>Download Bootloader</b>	Checks and upgrades the current version of the system boot loader.
<b>Download Main FPGA</b>	Checks and upgrades the current version of the main FPGA file. This function may not be available for all platforms and versions.
<b>Download Safe FPGA</b>	Checks and upgrades the current version of the backup copy of the FPGA file. This function may not be available for all platforms and versions.

Element	Description
<b>Download System Supervisor</b>	Checks and upgrades the current version of the system supervisor firmware (used for the RTC and power supply handling).
<b>Upload Configuration OS</b> <b>Upload Main OS</b> <b>Upload Splash Image</b> <b>Upload Bootloader</b> <b>Upload Main FPGA</b> <b>Upload Safe FPGA</b> <b>Upload System Supervisor</b>	Copy the system files from the operator panel on the external device (usually an USB stick).

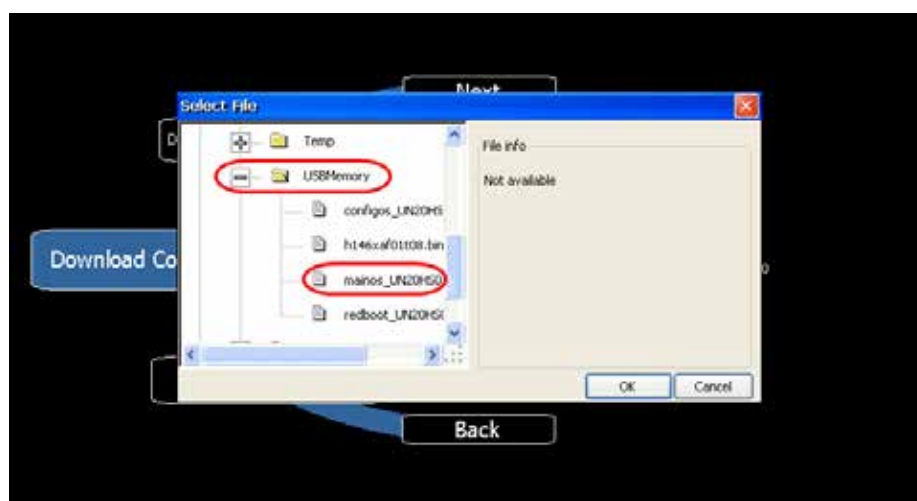
## Update System Components

System components can be updated using a USB flash drives. For each component, a couple of specific update files are provided.



Note: Upgrading procedures depend on hardware and operating system versions. Contact technical support for assistance.

1. Copy all the upgrade files you need to a USB drive and plug it into the USB port of the HMI device.
2. Start the System Settings tool in System Mode (see ["System Settings" on page 525](#) 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.

- Follow the instructions displayed to complete the update: the progress of the operation is displayed in a progress bar.



This operation may require a few minutes.



**Important: Do not turn off the device while a system component is being upgraded.**

## List of upgradable components

The HMI devices support the upgrade of the following components:

Component	Description
Application	The HMI Application and the HMI Runtime generated from the <b>Run&gt; Update Package</b> command
Main OS	Main Operating System
Configuration OS	Backup operating system that ensures units recovery in case of main operating system corruption
Splash	The initial screen shown during the startup of the HMI device
Bootloader	Loader to handle device startup
Main FPGA	FPGA firmware
Safe FPGA	Backup copy of the Main FPGA that ensures unit booting in case of main FPGA corruption <div>  <b>Important: Use the same file for updating Main and Safe FPGA components.</b> </div>
System Supervisor	Firmware of the system supervisor controller (for example: packaged_GekkoZigBee_v4.13.bin). <div> <i>The System Supervisor component can be upgraded from v4.13 or above.</i> </div> <div>  <b>Important: Do not try to update versions V4.08, V4.09, V4.10 and V4.11 since they do not support automatic update from System Settings.</b> </div>

## Touchscreen calibration

System Setting Calibration allows to calibrate Touchscreen device, can be accessed from System Settings

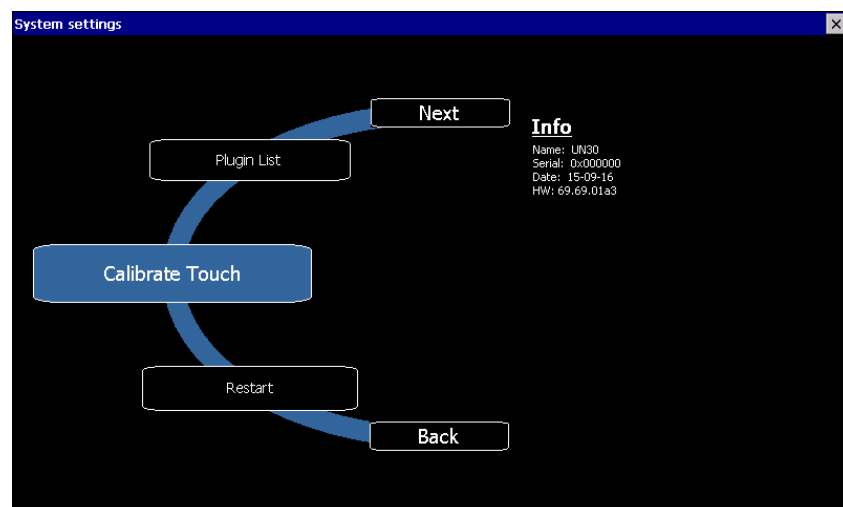
To access System Settings:

- Execute a tap sequence on the touch screen during the power-up phase. A tap sequence is a high frequency sequence of touch activations executed immediately after the device has been powered.

or

- Press and hold on an empty area of the screen for a few seconds to display the context menu.

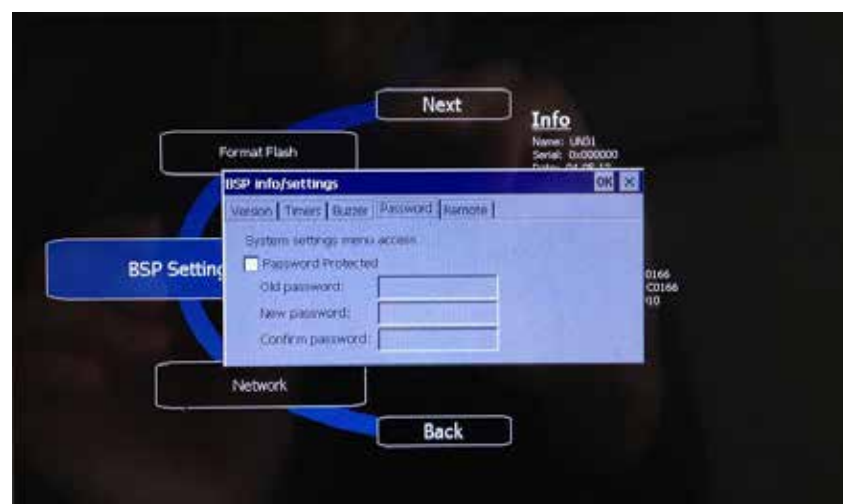
From the rotating menu, select “*Calibrate Touch*” and follow the instructions on screen to complete the calibration procedure, system will prompt to touch specific points to calibrate the touchscreen device.



## Password protection

Internal password of the HMI device can be defined from the System Settings in System Mode (see "[System Settings](#)" on [page 525](#) for details)

From the rotating menu, select “*BSP Settings*” and then the Password tab to open the set password dialog.



The password must be at least 5 characters long.



Leave “Old password” empty as default if target password is not set.



*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 525](#) for details.



# 41 Web access

---

JM4Web 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 JM4Web, users can create a web project to display at a remote location the same graphical display shown on the HMI device. JM4Web 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 JM4Web pages on the HMI device.

---

<b>Supported platforms and browsers</b> .....	<b>534</b>
<b>Generating page for Web access</b> .....	<b>534</b>
<b>Platform specific Home pages</b> .....	<b>536</b>
<b>Testing the Web project</b> .....	<b>536</b>
<b>Downloading the Web project</b> .....	<b>537</b>
<b>Web connectivity issues</b> .....	<b>538</b>
<b>Web supported features</b> .....	<b>539</b>
<b>Troubleshooting and FAQ</b> .....	<b>543</b>
<b>Privacy</b> .....	<b>543</b>

# Supported platforms and browsers

JM4Web 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

JM4Web works with all modern web browsers. The following browsers have been tested for compatibility with JM4Web:

- Mozilla Firefox 50+
- Microsoft Internet Explorer 11+
- Apple Safari 10+
- Google Chrome 52+



## Working with tablets or phones

JM4Web works with most tablet and phone devices. The following tablets have been tested for compatibility with JM4Web:

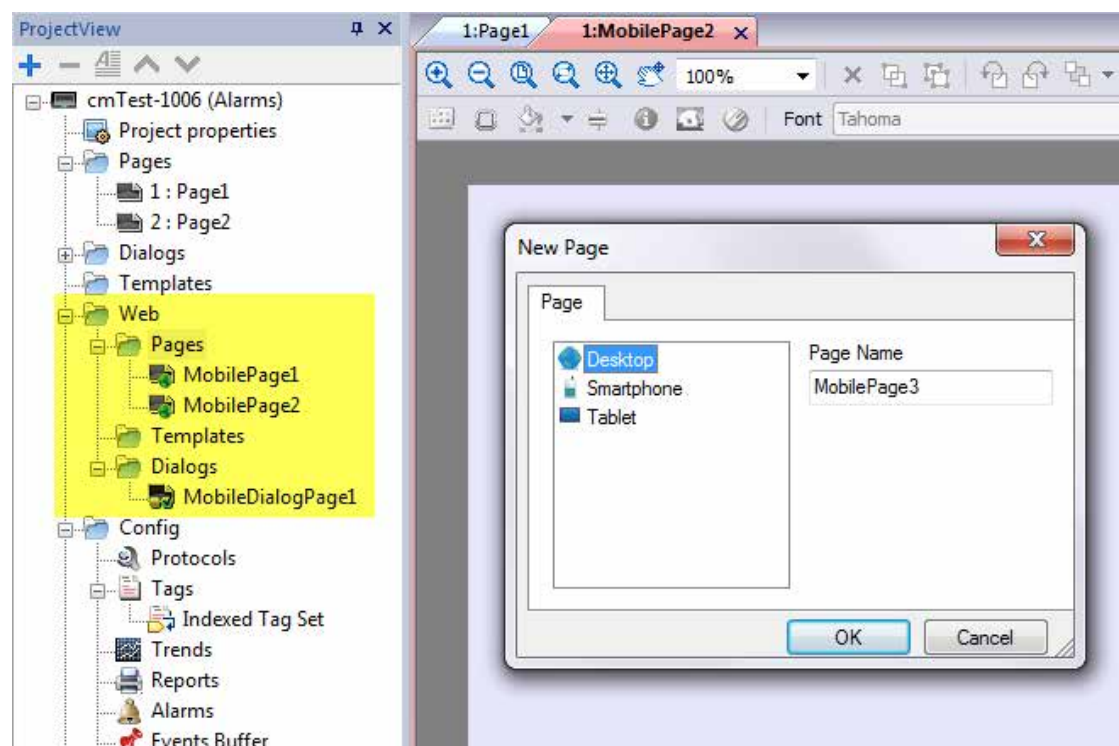
- iOS 10+ - Mobile Safari
- Android 7+ - Chrome for Android 55+



# Generating page for Web access

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

Right-click the **Pages** node and select **Insert Page** to add a web page.



Any widgets and features can be used in JMobile Studio; however, not all features are currently available in JM4Web. If the project includes a feature that is not available, JM4Web will still work correctly but the feature will not be available on the remote client device.

See ["Web supported features" on page 539](#) for a list of the features supported in JM4Web and of the existing limitations.

## Web page properties

In addition of the standard page's properties, there is an addition property

Property	Description
Fit to Screen Size	<p>How the page will adapted to the browser's viewport</p> <ul style="list-style-type: none"> <li>None</li> <li>Fit to Screen Simple modify the zoom level to adapted the page to viewport of the browser</li> <li>Responsive Design Smart modify the zoom level to adapted the page to viewport of the browser respecting the restrictions defined inside the grid layout</li> </ul>

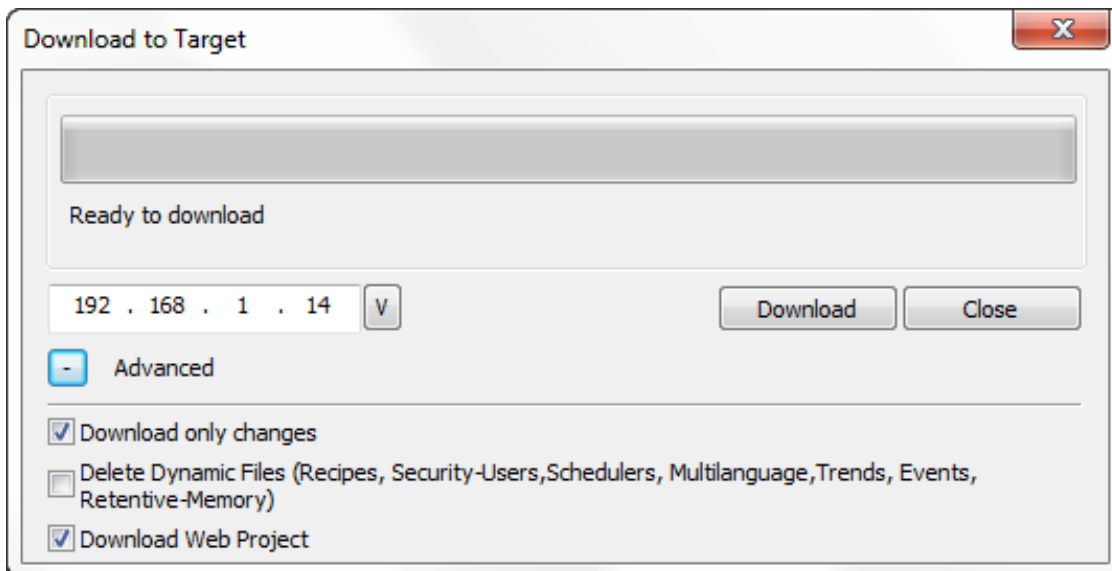
## Exporting pages

To select pages to export from the current project.

1. On the **Run** menu, click **Web Project Settings**: the **Web Project Settings** dialog is displayed.
2. Select the web pages you wish to export. By default all project and dialog pages are selected.
3. For each platform, select the home icon next to the page you want to define as the Home page. Only one Home page can be selected for each platform. All other home icons are grayed.



**WARNING:** When you download a project to the HMI device, make sure the **Download Web Project** option is selected.



## Platform specific Home pages

The Home Page of the JM4Web project defines the first page that is shown in the browser of each platform type and defines the starting point for your web project. Pages that can be accessed from home page depend on the how other pages are linked in the project.

For example, if you have designed a set of pages for a phone platform, set as a Home Page a page appropriately sized for a mobile phone display. Then include in this page only links to other phone pages: the user will only access phone pages when browsing the JM4Web project from a phone.

## Testing the Web project

You can test your JM4Web project using the online simulator opening a standalone web page directly from a browser.

### Testing with the online simulator

JMobile Studio includes an web server in the online simulator. You can start the simulator and access your JM4Web project from a web browser. The pages will be served from the simulator.

1. Create your project (see ["Generating page for Web access" on page 534](#)).
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 534](#) for a list of browser compatible with JM4Web).
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 JM4Web.
5. Test your project in the browser.



**Important:** If you make any changes to the project pages in JMobile Studio 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

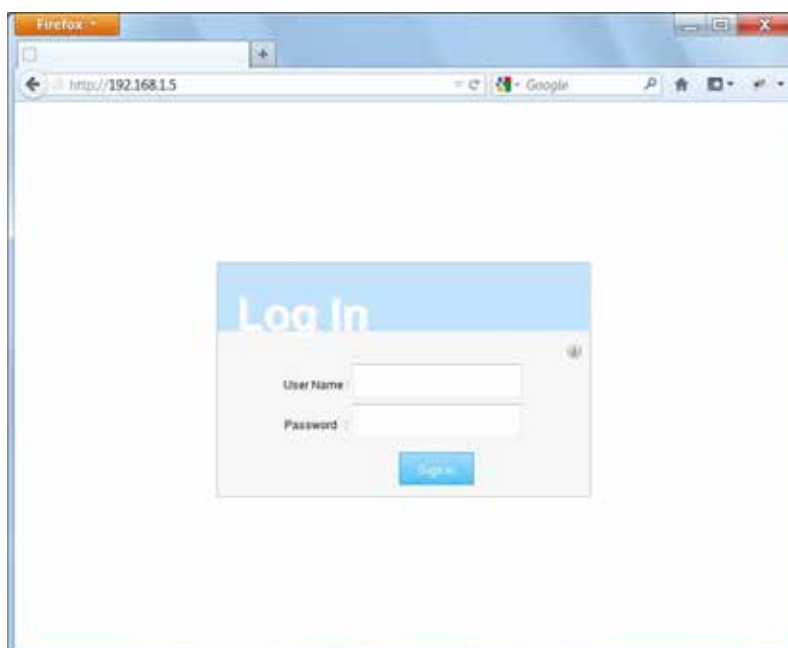
After testing the JM4Web pages, you can download the project to the desired HMI device.

The JM4Web project is downloaded together with the JMobile Studio project, see ["Download to HMI device" on page 90](#) for details.

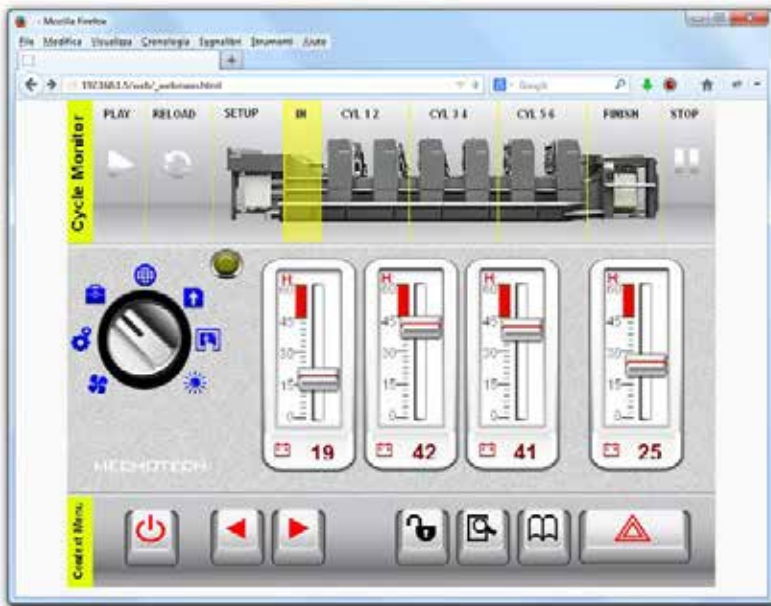
After the download process is completed, the HMI project automatically starts on the HMI device and the JM4Web project is ready to be used.

## Running JM4Web 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 309 for details on how to create credentials.

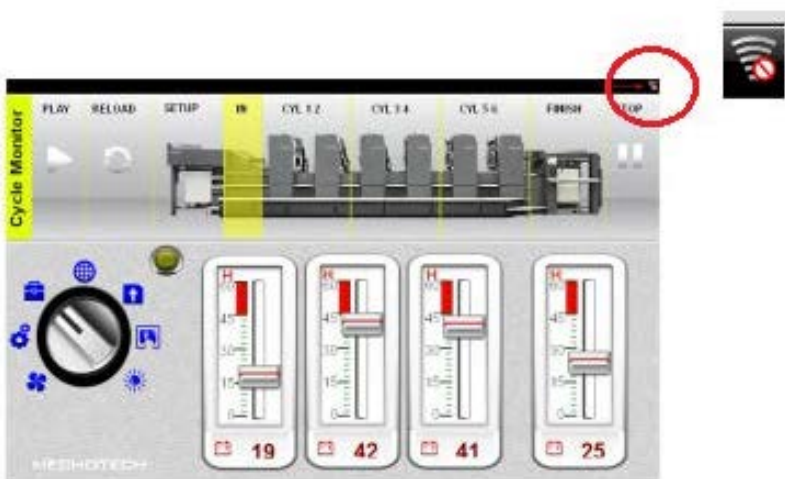
You can interact with the project using the browser in the same way you interact with a device when touching the screen: click buttons to change pages, view indicators and gauges, drag slider handles to change values, and so on. The JM4Web project will manage all communications with the web server while 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 JM4Web 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 JM4Web 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 107](#) for additional details.



Note: If you make changes in the JM4Web 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

JM4Web 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 JM4Web Project

The JM4Web 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 JM4Web 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 JM4Web cannot perform any server-bound communication.

## Web supported features

Currently not all JMobile Studio features are supported in JM4Web. Here a list of features supported and limitations, classified by category.



**When you copy and paste objects from standard pages to a web pages, make sure that all objects are supported in web pages. Eventually remove unsupported objects from the web page after paste.**

Category	Supported features	Limitations
Widgets	<ul style="list-style-type: none"> <li>• Basic (Text/Numeric, Images, Shapes, Trends/Graphs, Recipes, Controls, Alarms, Texture, JSFunctionBlock, Generic Canvas e Grid Layout)</li> <li>• Buttons</li> <li>• Meters</li> <li>• Switches</li> <li>• Lights</li> <li>• Media (IP Camera)</li> <li>• Icons</li> <li>• Factory Automation</li> </ul>	<ul style="list-style-type: none"> <li>• AttachToTag of system variables is not supported</li> <li>• Font files without web download permissions flag enabled are not loaded from the JM4Web</li> <li>• Widget properties with Attach to... dynamic behavior may not work for all properties supported by JMobile Studio.</li> <li>• Multistate Image Multi-Layer is not supported.</li> <li>• Alarm Color based on trigger condition is not supported in Web</li> <li>• Can not edit the Alarm widgets in runtime</li> <li>• The ComboBox View Mode in full screen is not supported</li> <li>• BACnet widgets are not supported</li> </ul>
Alarms	<ul style="list-style-type: none"> <li>• Alarms limits in JM4Web is the same of HMI device (500..2000 based on target)</li> </ul>	<ul style="list-style-type: none"> <li>• On Smartphone/Tablet (in general embedded devices) based on HW a user could expect performance problems with &gt; 500 alarms.</li> </ul>



Category	Supported features	Limitations
Actions	<ul style="list-style-type: none"> <li>• Widgets (Javascript)</li> <li>• Page (HomePage, LoadPage, NextPage, PrevPage, LastVisitedPage, ShowDialog, CloseDialog, ShowMessage, LaunchBrowser)</li> <li>• Multilanguage (SetLanguage)</li> <li>• Tag (WriteTag, StepTag, SetBit, ResetBit, ToggleBit)</li> <li>• Trend/Graph (RefreshTrend, ScrollLeftTrend, ScrollRightTrend, PageLeftTrend, PageRightTrend, ScrollUpTrend, ScrollDownTrend, PageUpTrend, PageDownTrend, PageDurationTrend, ZoomInTrend, ZoomOutTrend, ZoomResetTrend, ZoomInAxisTrend, ZoomOutAxisTrend, ZoomResetXAxisTrend, PauseTrend, ResumeTrend, ShowTrendCursor, ScrollTrendCursor, ScrollTrendToTime, setTrendView)</li> <li>• Alarm (ResetAlarm, AckAlarm, SelectAllAlarms, EnableAlarms)</li> <li>• System (DumpTrend, DeleteTrend, DeleteEventArchive)</li> <li>• Recipes (DownloadRecipe, UploadRecipe, WriteCurrentRecipeSet, DownloadCurRecipe, UploadCurRecipe, ResetRecipe, DumpRecipeData, RestoreRecipeData, AddRecipeDataSet, DelRecipeDataSet)</li> <li>• User Management (LogOut e ChangePassword)</li> </ul>	<ul style="list-style-type: none"> <li>• JavaScript is supported (see <a href="#">"JavaScript" on page 453</a> JavaScript chapter for a list of supported features)</li> <li>• Page actions are not supported in alarm trigger condition</li> <li>• The page's OnWheel event is not supported</li> </ul>
XForms	<ul style="list-style-type: none"> <li>• Formula</li> <li>• Scaling</li> <li>• Bit/Byte Index</li> <li>• ColorPalette</li> </ul>	
Keypads	<ul style="list-style-type: none"> <li>• Only numeric keypads widgets are supported.</li> </ul>	<ul style="list-style-type: none"> <li>• Custom keypads are not supported. The numeric keyboard will be displayed as numeric widgets with a read/write or write mode.</li> </ul>
Dialog Page	<ul style="list-style-type: none"> <li>• Supported, you can show them and close them based on the ShowDialog and CloseDialog actions.</li> </ul>	<ul style="list-style-type: none"> <li>• Dialog pages support only modal dialogs.</li> </ul>

Category	Supported features	Limitations
User Management	<ul style="list-style-type: none"> <li>The login mechanism verifies user credentials on the server. The user name and password are based on the user credentials defined in <a href="#">"User management and passwords" on page 309</a>.</li> </ul>	
Concurrent User Connections	<ul style="list-style-type: none"> <li>The web server in the HMI device supports three concurrent connections at a time.</li> </ul>	<ul style="list-style-type: none"> <li>If more than 3 connections are attempted from remote browsers, only the first 3 connections will be permitted.</li> </ul>

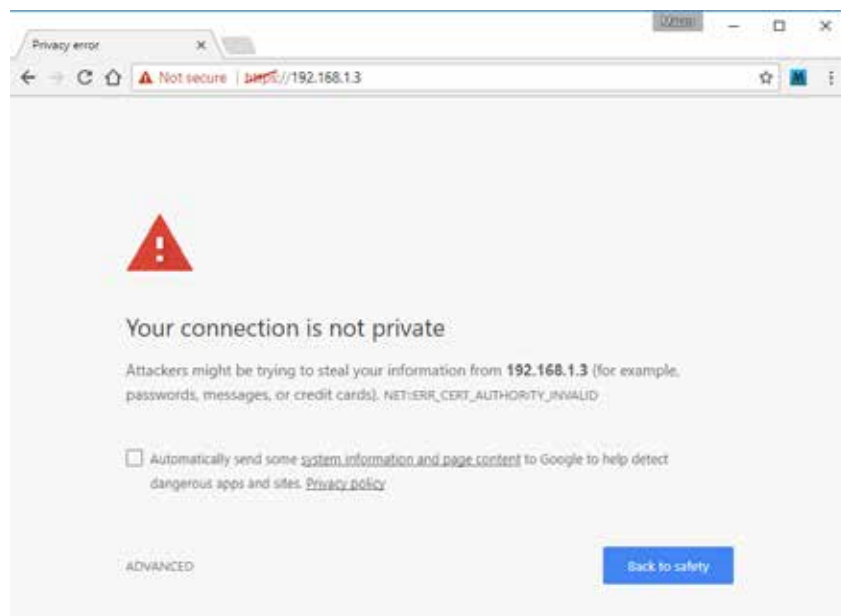
## Secure Socket Layer (HTTPS)

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.

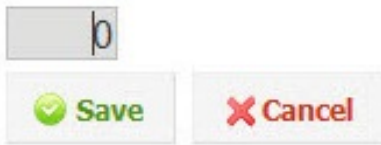
Simple click the ADVANCE button to continue.



## Working with keypads in JM4Web

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

JM4Web requires JavaScript to provide interactivity with the server and the user. JM4Web 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 JM4Web pages.

### Browser cache

JM4Web 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 JM4Web 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 JM4Web pages.

Caching is normally enabled by default, for optimal JM4Web performance make sure it has not been disabled.



Note: JM4Web 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 JM4Web 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.

## 42 License activation of HMI device software modules

---

You perform license related tasks for the HMI device in the Manage Target dialog of JMobile Studio.

---

<b>Activate the device</b> .....	<b>546</b>
<b>Save a license</b> .....	<b>547</b>
<b>Import a license</b> .....	<b>547</b>

# Activate the device

On each HMI device you need to activate the corresponding license. To do this you need an Internet connection.



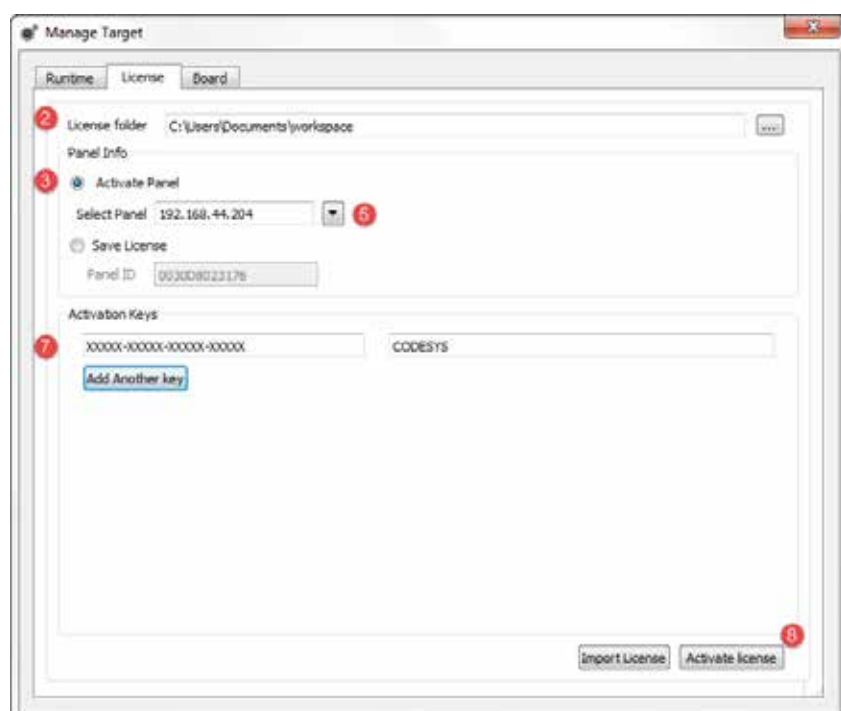
**Important: JMobile HMI Runtime must be installed in the device before you can activate your license.**

**Path: Run > Manage Target**

1. In the **Manage Target** dialog, click the **License** tab.
2. In **License folder** select the location of your backup license files.
3. Choose the **Activate Panel** option.
4. Select the panel.
5. Click **Select IP** button: all of the devices connected to the network are listed.
6. Select the device on which the license is to be activated.
7. Enter **Activation Key**.

Once you have entered an activation key, the **Add Another Key** button is enabled, and you can add another activation key.

8. Click **Activate license**: if the activation key is valid, the license file is downloaded to the HMI device and the license is activated.



The enabled features or status of the activation keys is displayed in the Feature/status list box.



**Note:** You need to restart the HMI device to enable the activated license.

## Save a license

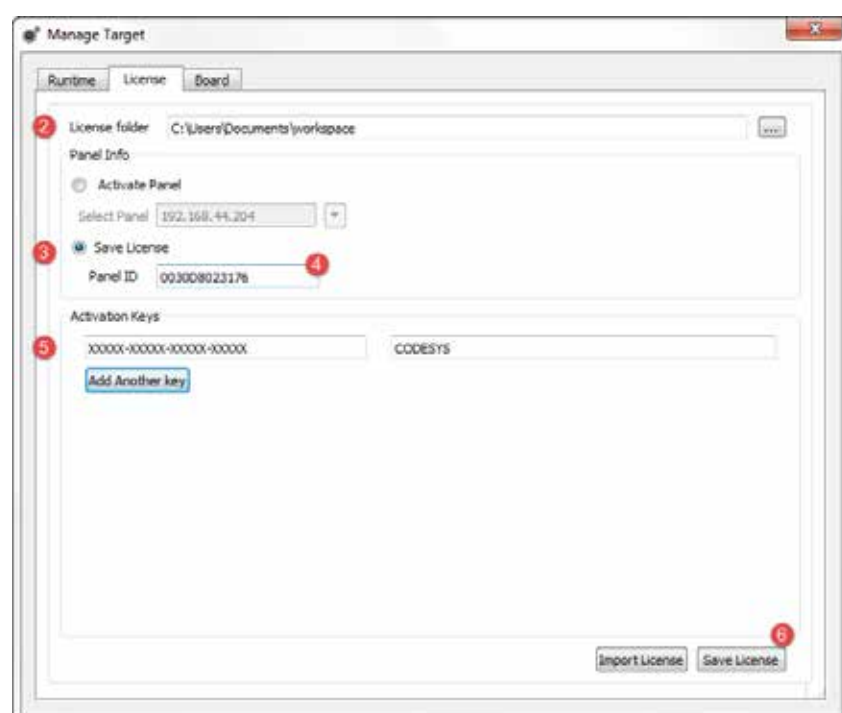
You may need to create a license file for later activation on the HMI device.

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

1. In the **Manage Target** dialog, click the **License** tab.
2. In **License folder** select the location of your backup license file.
3. Choose the **Save License** option.
4. Enter **Panel ID**.
5. Enter **Activation Key**.

Once you have entered an activation key, the **Add Another Key** button is enabled, and you can add another activation key.

6. Click **Save License**: if the activation key is valid, the license file is stored in the selected folder.



The enabled features or status of the activation keys is displayed in the Feature/status list box.

## Import a license

If no working Internet connection is available on the computer running JMobile Studio, you can neither save or activate a license. You must therefore import your license from another computer where you were able to download it.

This is a two-step procedure:

1. First you download and activate your license on a computer connected to the internet
2. Then you import the downloaded license to the HMI device.

## Downloading and activating the license

1. On a computer with Internet access, open a browser and go to <https://license.x-formation.com>
2. Enter your activation key (for example, 5BDI0-FORLS-HR67G-5BI5T).
3. Enter your HMI MAC ID in the field **HostID Value** as **Custom=<MAC\_ID>** (for example, Custom=0030D801DE27).
4. Click **Activate**: the license file is generated.
5. Click **Download**: the license file is saved to your computer.

## Importing and activating the license

*Path: Run> Manage Target*

1. In the **Manage Target** dialog, click the **License** tab.
2. Select the device if it is connected or enter the MAC ID if you are generating the license offline.
3. Click **Import license** and select the license file you created. Add more license files if more than one activation key was purchased.
4. Click **Activate license** to activate the licenses on the HMI device or **Save License** for offline license file generation.



Note: You need to restart the HMI device to enable the activated license.



# 43 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 499 for details)
- From JMobile Studio 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.*

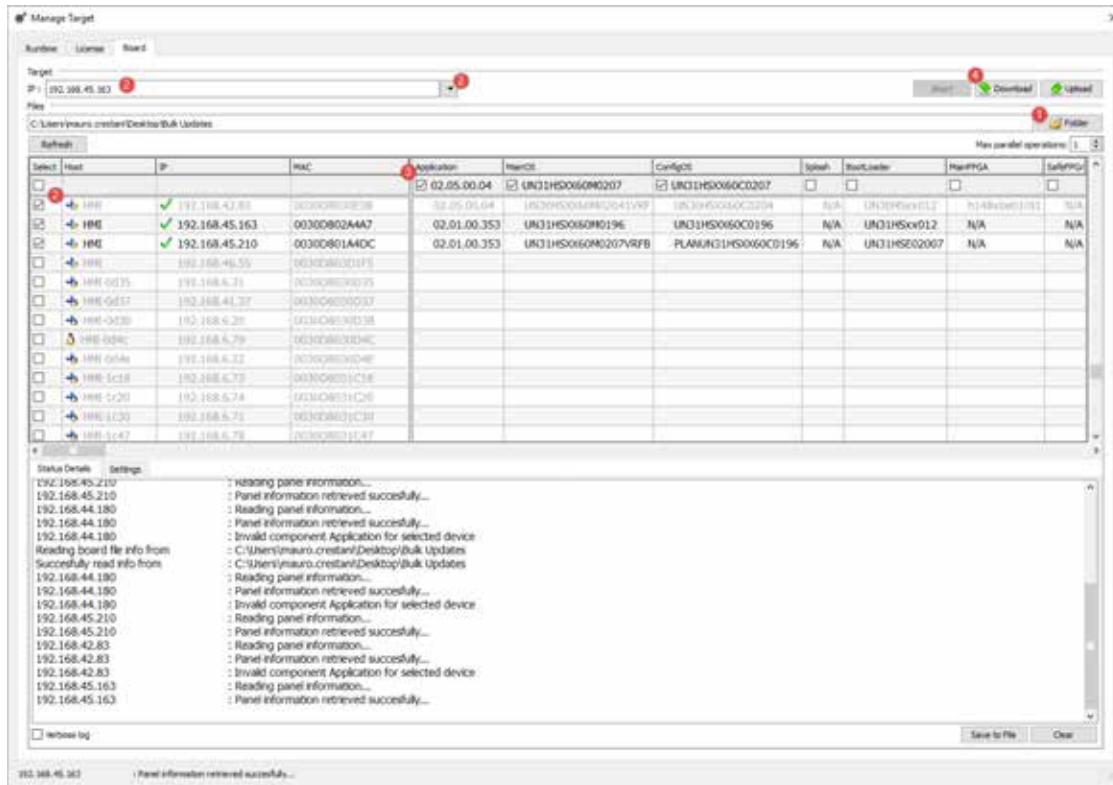
---

Update of system components from the application .....	550
Settings .....	551

# Update of system components from the application

You can download system components to a single HMI device or to a bulk of HMI devices of the same type using the Ethernet communication interface.

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



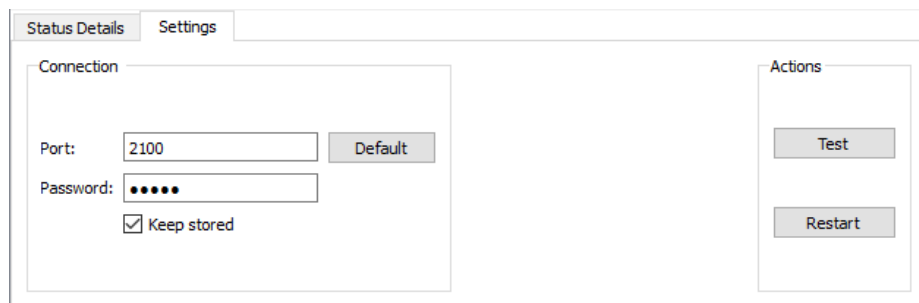
1. Select the folder that contains the files to download to the HMI device or where to upload files from the HMI device
2. Select one or more HMI device.
3. Select the components that you will download (or upload) to/from the devices
4. Start the Download to HMI or the Upload from the HMI operation

## Note:

- The tool is designed to update multiple HMI devices of the same type. Please avoid putting files for different device type into the same folder
- If the desired target IP is not listed, type it directly into the box. The discovery service is a broadcast service. When a remote connection is done via VPN or from external networks, it will not work and you will have to enter the address manually.
- Download of the selected components will be performed only to the compatible devices
- Based on your network and hardware capabilities you can increase the number of devices to update in parallel
- You need to restart the HMI device to finalize the update.

## Settings

From the **Settings** tab you can specify the Port and the Password parameters to use to communicate with the HMI devices. Leave Password empty if no password is set on the HMI device side.




**WARNING:** Bulk mode is working only with the HMI devices that have the same connection parameters

## Uploading a splash screen picture

You can replace the default splash screen image shown by the devices during the power up phase.

The image used as splash screen must comply with the following requirements:

Filename	splash.bmp
Format	Bitmap, RGB 565 format
Size	< 500 KB
Bitmap width	Even number (for example 430x239)

To upload the splash screen image:

1. Rename the new image splash.bmp and copy it in the source folder.
2. Select HMI devices
3. Click **Download**.



To ensure the best visual results, splash screen images must have a black background.



# 44 Protecting access to HMI devices

The following operations are password protected on the HMI device:

- JMobile HMI Runtime management: install JMobile HMI Runtime and update JMobile 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: 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 .....	554
Ports and firewalls .....	554

## Changing password on HMI device

To change the password on the HMI device, use one of the following methods:

- From the JMobile HMI Runtime context menu: **Settings> Password** tab.



- Use the **Set Target Password** function in update package: the password is updated by JMobile HMI Runtime just after the update process is completed.
- Using HMI device "[System Settings](#)" on [page 499](#) Tool



Leave "Old password" empty as default if target password is not set.



For Win32 JMobile HMI Runtime, password is saved into *Users\[username]\AppData\Roaming\Exor\buildNumber\server\config\RemoteUpdateConfig.xml*.

## Ports and firewalls

Here a list of all the ports used by JMobile Suite 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
1200/tcp	CODESYS 2.3	-	-	-
11740-11743/tcp 1217, 1740- 1743/udp	CODESYS 3	-	-	-

## Remote access

Remote access is required to connect to JMobile HMI Runtime using:

- HMI Client
- Internet Browser

## Runtime and project management ports

You use these ports to connect to JMobile 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 JMobile Studio.





# 45 Tips and tricks to improve performance

---

JMobile Studio allows great flexibility for a project designers.

Follow these guidelines to create projects that perform better in terms of boot time, page change and animations.

---

<b>Static Optimization</b> .....	<b>558</b>
<b>FAQ on Static Optimization</b> .....	<b>561</b>
<b>Page caching</b> .....	<b>562</b>
<b>Image DB</b> .....	<b>562</b>
<b>Precaching</b> .....	<b>562</b>
<b>FAQ on precaching</b> .....	<b>562</b>

# Static Optimization

Static optimization is a technique used in JMobile Studio to improve run-time 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 JMobile Studio, 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 run time.
- **Dynamic:** values or properties change at run time.



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, JMobile Studio 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.

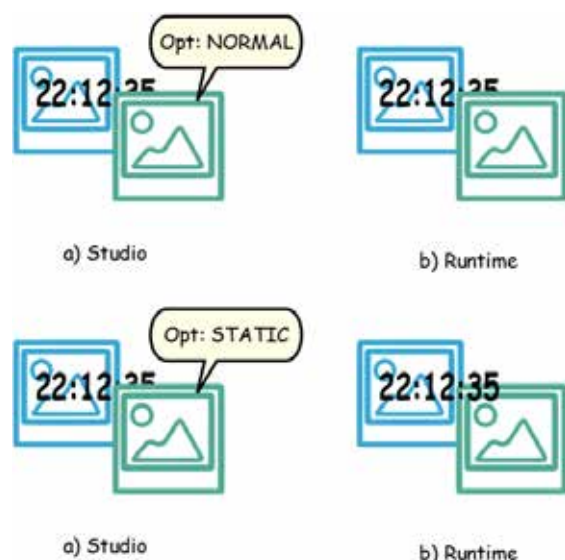
Finer control can be achieved setting the **Static Optimization** attribute of each single widget as follows:

- **Normal:** JMobile Studio 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 run time.

- **Dynamic:** The widget is not optimized at all. Use this flag when a static widget needs to be changed by Javascript.



## Tips for best performance

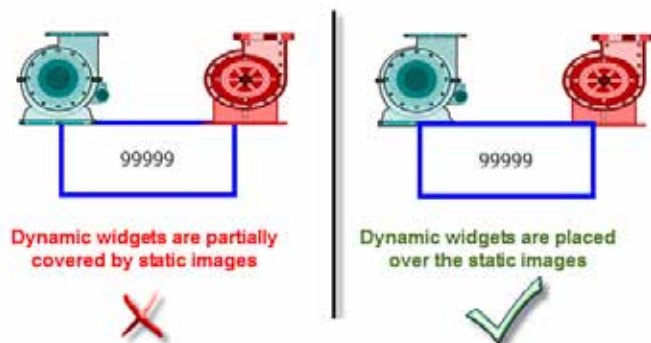
1. First of all: avoid placing static widgets over a dynamic widget. The overlapping area is computed considering the bounding rectangles of the widgets, that is the rectangles delimited by editing handles.
2. Don't use static optimization if your pages contain almost only dynamic objects. Static optimization 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 run time for dynamic images and "hides" the actual image size during editing.
6. Use **Size to fit** to make the widget to the real size of his contents.
7. If overlapping cannot be avoided make sure to place the static widgets in the back, that is behind the dynamic widget.
8. Choose the image file format based on the HMI device you are connecting to.

9. Avoid using too many widgets in a single page. Often widgets are placed outside the visible area or their transparency is controlled by a tag. Since widgets are loaded even if they are not visible, having too many widgets in a page can significantly slow down the page change time.
10. Split a page with many widgets into multiple pages with less widgets.
11. For popping up new graphic elements in a page, prefer dialog pages with controlled positioning to transparent widgets.
12. Check the *opt* folder to see if static optimization is working as expected, the widgets z-order might need to be adjusted.
13. Numeric fields are often used to run JavaScript code on OnDataUpdate event even if the widget doesn't need to be visible on the page. In this case place the widget outside the page visible area instead of making it invisible, altering font color or visibility property. In the latter case you might end up with many left over wedges.
14. Use a HotSpot button if you need a touch area to react to user inputs.
15. If you reuse a widget from the gallery or you create your own, remember to set the correct optimization properties. For example button widgets are dynamic widgets, if you use a button widget just for its frame it won't be optimized since the button widget is dynamic. If you just need the frame you should use the Up image.
16. With many pages having many dynamic widgets and using a common template:
  1. set template static optimization to **true**,
  2. set page static optimization to **false**, since the background is already provided by the template.

In this scenario the background image can be reused by many different pages thus saving memory space.

17. Do not use dynamic widgets, such as buttons, only for graphic purposes, when the button function is not needed, use image widgets instead to obtain the same graphical effect.

Here is an example of a correct and an incorrect use of static optimization.



## Supported image formats

JMobile Studio supports several raster formats like BMP, PNG, JPEG, TIFF and the vector format SVG. Here a list of pros and cons:

Image format	Pros	Cons
RASTER	<ul style="list-style-type: none"> <li>• Fast rendering</li> <li>• Well standardized</li> </ul>	<ul style="list-style-type: none"> <li>• Big file size</li> <li>• Fixed resolution</li> </ul>
VECTOR (SVG)	<ul style="list-style-type: none"> <li>• Small file size</li> <li>• Rescale without quality loss</li> <li>• Can handle dynamic properties</li> </ul>	<ul style="list-style-type: none"> <li>• Complex SVG images with many graphic items and layers can be slow to render.</li> <li>• Creating an optimized SVG is not simple.</li> <li>• Only Tiny 1.2 (<a href="http://www.w3.org/TR/SVGTiny12/">http://www.w3.org/TR/SVGTiny12/</a>) supported.</li> </ul>



Note: Scour software is free tool that can be used to remove foreign code from file (<http://www.codedread.com/scour/>).

## Static optimization of templates

Template pages can have large amounts of static content. However, static optimization cannot be applied to a template page, since where the template is used is based on the page design.

If a huge background image should be repeated in every page that uses the same template, this would increase the footprint of the device as the same static image would be created for each of the pages using the template page.

## FAQ on Static Optimization

**Q: In a page where there are a few identical widgets, in the *opt* folder I see a PNG for each one of them. If they are really identical, why should the software duplicate them instead of having just one PNG?**

A: The software does not know if static images are actually the same since each widget could have different settings/properties altering the actual rendering at run time.

**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 JMobile Studio(not using optimization) and JMobile HMI Runtime.

**Q: What will happen when no optimization is done in the software?**

A: Every single widget is rendered at run time. 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 JMobile 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, JMobile HMI Runtime proceeds as follows:

1. Page images are preloaded until 76 MB of memory space is still available (imageDBLowMem)
2. Pages where precache is set to **true** are preloaded until 64 MB of memory space is still available (pageCacheLowMemMax). The images of these pages are loaded in the RAM (into the Image DB).

When the project is ready:

1. Any new page visited is saved in the cache (RAM) with all related images until 40 MB of memory space is still available (pageCacheLowMemMin)
2. When a page change happens and space in RAM is critical (<40MB), the JMobile HMI Runtime starts emptying the cache (RAM) removing pages and related images until 64 MB of memory space is made available. JMobile HMI Runtime removes data stored in the cache in the following order:
  1. last visited pages and bigger and unused images (>320x240),
  2. if more memory is needed also the pages in precache and all images loaded in Image DB can be removed.





## 46 Functional specifications and compatibility

---

Here is an overview of the supported functions and related limitations. Limitations indicated here represent a safe limitation, beyond that proper operation and state-of-the-art performance of the system is not guaranteed.

---

<b>Table of functions and limits .....</b>	<b>566</b>
<b>HMI devices capabilities .....</b>	<b>567</b>
<b>Compatibility .....</b>	<b>569</b>
<b>Converting projects between different HMI devices .....</b>	<b>569</b>

## 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	50
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	50
Number of concurrent remote clients	4
Number of schedulers	30
Number of alarms	2.000 (See <a href="#">"HMI devices capabilities" on the facing page</a> )
Number of data transfers	1000
Number of templates pages	50
Number of actions programmable per button state	32
Number of Trend Buffers	30
Number of curves per trend widget	5
Number of curves per scatter diagram widget	10
Number of samples per trend buffer	200.000
Number of tags per trend buffer	200
Number of trend buffer samples for a project	1.200.000 (See <a href="#">"HMI devices capabilities" on the facing page</a> )
Number of messages in a message field	1024
Number of languages	12
Number of events per buffer	2.048
Number of event buffers	4

Function	Max limit
JavaScript file size per page	16 KB
Size of project on disk	60 MB (See <a href="#">"HMI devices capabilities" below</a> )
Number of indexed instances	100
Number of indexed alias	100
Number of indexed tag sets	30
Number of physical protocols	4
Number of reports	32
Number of reports pages	32
Max number of variables in variables widget	255
User folder size (UpdatePackage.zip)	5 MB
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 Samples	Curves per Trend Widget	Events Buffers Size	User Folder Size
Runtime PC	Win32		MPG4	Yes	Yes	60 MB	200	10.000	8	64 KB	64	2.400.000	10	10 K	na
eBI5504	WEC		na	Yes	na	30 MB	50	500	4	16 KB	32	1.200.000	5	2 K	10 MB
eBI5507	WEC		na	Yes	na	30 MB	50	500	4	16 KB	32	1.200.000	5	2 K	10 MB
eBI5510	WEC		MPG4/H264	Yes	Yes	60 MB	50	2000	4	16 KB	32	1.200.000	5	2 K	100 MB
eBI5513	WEC		MPG4/H264	Yes	Yes	60 MB	50	2000	4	16 KB	32	1.200.000	5	2 K	100 MB
eSMART04	Linux		na	Yes	Yes	60 MB	50	500	4	64 KB	32	1.200.000	5	2 K	100 MB
eSMART04M	Linux		MPG4	Yes	Yes	60 MB	50	2000	4	64 KB	32	1.200.000	5	10 K	100 MB
eSMART07M	Linux		MPG4	Yes	Yes	60 MB	50	2000	4	64 KB	32	1.200.000	5	10 K	100 MB
eSMART10	Linux		MPG4	Yes	Yes	60 MB	50	2000	4	64 KB	32	1.200.000	5	10 K	100 MB
eSMART107	Linux		MPG4/H264	No	Yes	240 MB	200	4.000	8	64 KB	64	2.400.000	10	10 K	100 MB
eTOP504	WEC		na	Yes	na	30 MB	50	500	4	16 KB	32	1.200.000	5	2 K	10 MB
eTOP506	WEC		na	Yes	na	30 MB	50	500	4	16 KB	32	1.200.000	5	2 K	10 MB
eTOP507	WEC		na	Yes	na	30 MB	50	500	4	16 KB	32	1.200.000	5	2 K	10 MB
eTOP507G	WEC		na	Yes	na	30 MB	50	500	4	16 KB	32	1.200.000	5	2 K	10 MB
eTOP507M	WEC		MPG4/H264	Yes	Yes	60 MB	50	2000	4	16 KB	32	1.200.000	5	2 K	100 MB
eTOP507MFB	WEC		MPG4/H264	Yes	Yes	60 MB	50	2000	4	16 KB	32	1.200.000	5	2 K	100 MB
eTOP507MG	WEC		MPG4/H264	Yes	Yes	60 MB	50	2000	4	16 KB	32	1.200.000	5	2 K	100 MB
eTOP510	WEC		MPG4/H264	Yes	Yes	60 MB	50	2000	4	16 KB	32	1.200.000	5	2 K	100 MB
eTOP510G	WEC		MPG4/H264	Yes	Yes	60 MB	50	2000	4	16 KB	32	1.200.000	5	2 K	100 MB
eTOP512	WEC		MPG4/H264	Yes	Yes	60 MB	50	2000	4	16 KB	32	1.200.000	5	2 K	100 MB
eTOP513	WEC		MPG4/H264	Yes	Yes	60 MB	50	2000	4	16 KB	32	1.200.000	5	2 K	100 MB
eTOP515	WEC		MPG4/H264	Yes	Yes	60 MB	50	2000	4	16 KB	32	1.200.000	5	2 K	100 MB
eTOP515G	WEC		MPG4/H264	Yes	Yes	60 MB	50	2000	4	16 KB	32	1.200.000	5	2 K	100 MB
eTOP605	WEC		na	Yes	na	30 MB	50	500	4	16 KB	32	1.200.000	5	2 K	10 MB
eTOP607L	Linux	Multi	MPG4	Yes	Yes	60 MB	50	2000	4	64 KB	32	1.200.000	5	10 K	100 MB
eTOP607M	WEC		MPG4/H264	Yes	Yes	60 MB	50	2000	4	16 KB	32	1.200.000	5	2 K	100 MB
eTOP610	WEC		MPG4/H264	Yes	Yes	60 MB	50	2000	4	16 KB	32	1.200.000	5	2 K	100 MB
eTOP610L	Linux	Multi	MPG4	Yes	Yes	60 MB	50	2000	4	64 KB	32	1.200.000	5	10 K	100 MB
eX705	Linux	Multi	MPG4	Yes	Yes	60 MB	50	2000	4	64 KB	32	1.200.000	5	10 K	100 MB
eX707	Linux	Multi	MPG4/H264	No	Yes	240 MB	200	4.000	8	64 KB	64	2.400.000	10	10 K	100 MB
eX707FB	Linux	Multi	MPG4/H264	No	Yes	240 MB	200	4.000	8	64 KB	64	2.400.000	10	10 K	100 MB
eX707G	Linux	Multi	MPG4/H264	No	Yes	240 MB	200	4.000	8	64 KB	64	2.400.000	10	10 K	100 MB
eX710	Linux	Multi	MPG4/H264	No	Yes	240 MB	200	4.000	8	64 KB	64	2.400.000	10	10 K	100 MB
eX710G	Linux	Multi	MPG4/H264	No	Yes	240 MB	200	4.000	8	64 KB	64	2.400.000	10	10 K	100 MB
eX715	Linux	Multi	MPG4/H264	No	Yes	240 MB	200	4.000	8	64 KB	64	2.400.000	10	10 K	100 MB
eX715FB	Linux	Multi	MPG4/H264	No	Yes	240 MB	200	4.000	8	64 KB	64	2.400.000	10	10 K	100 MB
eX721	Linux	Multi	MPG4/H264	No	Yes	240 MB	200	4.000	8	64 KB	64	2.400.000	10	10 K	100 MB
eXware703	Linux		MPG4	Yes	Yes	60 MB	50	2000	4	64 KB	32	1.200.000	5	10 K	100 MB
eXware705	Linux		MPG4	Yes	Yes	60 MB	50	2000	4	64 KB	32	1.200.000	5	2 K	100 MB
eXware707	Linux		MPG4/H264	Yes	Yes	240 MB	200	4.000	8	64 KB	64	2.400.000	10	10 K	100 MB
eXware707Q	Linux		MPG4/H264	Yes	Yes	240 MB	200	4.000	8	64 KB	64	2.400.000	10	10 K	100 MB
H3 wired	WEC		na	Yes	na	30 MB	50	500	4	16 KB	32	1.200.000	5	2 K	10 MB
H4 wireless	WEC		na	Yes	na	30 MB	50	500	4	16 KB	32	1.200.000	5	2 K	10 MB
JSmart705	Linux	Multi	MPG4/H264	No	Yes	240 MB	200	4.000	8	64 KB	64	2.400.000	10	10 K	100 MB
JSmart707	Linux	Multi	MPG4/H264	No	Yes	240 MB	200	4.000	8	64 KB	64	2.400.000	10	10 K	100 MB
JSmart710	Linux	Multi	MPG4/H264	No	Yes	240 MB	200	4.000	8	64 KB	64	2.400.000	10	10 K	100 MB
JSmart715	Linux	Multi	MPG4/H264	No	Yes	240 MB	200	4.000	8	64 KB	64	2.400.000	10	10 K	100 MB
JSmart721	Linux	Multi	MPG4/H264	No	Yes	240 MB	200	4.000	8	64 KB	64	2.400.000	10	10 K	100 MB

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 JMobile 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:

- JMobile Studio version must always be aligned with JMobile HMI Runtime on the device,
- the user is responsible for updating JMobile HMI Runtime components on the HMI device at any JMobile Studio update,
- the JMobile HMI Runtime update can be done directly from JMobile Studio using the Update Target command available in the Run\Manage Target dialog,
- projects created in a JMobile Studio version no older than V1.00 (00) can be opened and handled by any newer version,
- projects created with older versions of JMobile Studio, opened with later versions and deployed to compatible JMobile HMI Runtime, are ensured to maintain the performance and functionality,
- compatibility between newer versions of JMobile HMI Runtime and projects created and deployed with older versions of JMobile Studio is not ensured.



**Important: Do not edit projects with a version of JMobile Studio older than the one used to create them. It can result in a damage of the project and to JMobile HMI Runtime instability.**

## Converting projects between different HMI devices

Project conversion from different HMI device models is supported, however, some manual operations may be required if the project uses features not supported in the destination device.

### Guideline

Before converting a project have a look if some unsupported features are present (see ["HMI devices capabilities" on page 567](#)), 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 566](#) 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.





**JMobile Studio**  
User Manual

2.08  
2018-10-30

Copyright © 2009-2018

**Exor International S.p.A.**  
Via Monte Fiorino, 9  
37057 San Giovanni Lupatoto (Verona)  
Italy

info@exorint.com  
**phone:** +39 045 8750404  
**fax:** +39 045 8779023