Tech-note

JMobile Suite User Manual

This document contains information for JMobile Suite on-line help, accessible from JMobile Studio\Help command

Sitek S.p.A.
PTn343
Ver. 1.26

Contents

1 Getting Started with JMobile Software

JMobile Studio is a software application that allows you to create graphical HMI pages. JMobile Studio uses a drag-and-drop system that makes it easy to create displays. The same features found in many popular Windows applications are also available in JMobile Studio.

This document describes how to use the JMobile Studio application, and is divided in chapters that represent the key operations of JMobile Studio. Each chapter is presented in a standalone manner, allowing you to jump from chapter to chapter, depending on the task you wish to perform.

1.1 Assumptions

We assume that those reading this manual are using JMobile Suite software to design control panel applications that run on UniOP panels, Series 400.
We also assume that you have a basic understanding of PCs, Microsoft Windows, and the type of network environment in which you will run the application.

1.2 Installing Mobile Suite on a PC

1.2.1 System Requirements

JMobile Studio has the following system requirements:

Windows XP –SP2 or SP-3, Windows Vista SP1 or SP2, Windows 7.
100 MB of disk space
Minimum of 512 MB RAM
Ethernet interface

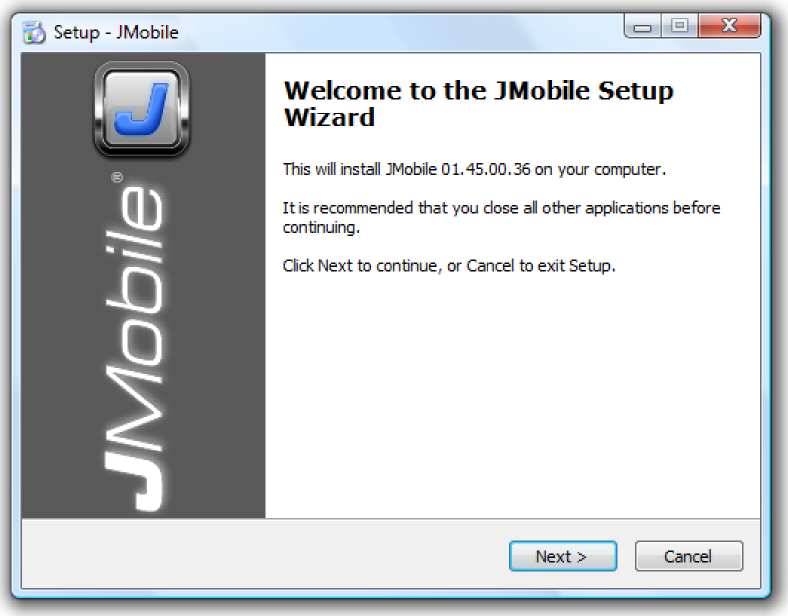
1.2.2 Installation

Insert the CD-ROM into the CD-ROM drive. If your system has Auto run enabled, the JMobile Suite installation will start automatically, otherwise run the JMobile Studio setup application.

Note: When running Windows 7 operating system, the installation must be done from users with Administration privileges.

Click on the Start button and select Run from the popup menu.
Type D:\setup in the box if you running the installation from the CD drive (if your CD-ROM drive is not drive D, replace with the appropriate letter).

Follow then the instructions on the screen.



Welcome to the JMobile Setup Wizard

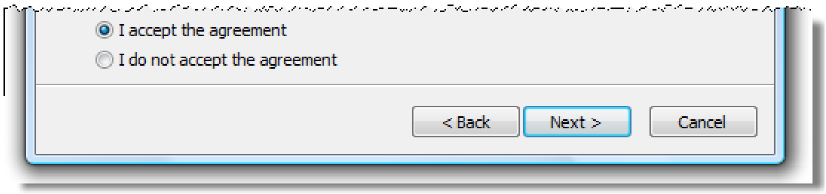
This will install JMobile 01.45.00.36 on your computer.

It is recommended that you close all other applications before continuing.

Click Next to continue, or Cancel to exit Setup.

Figure 1

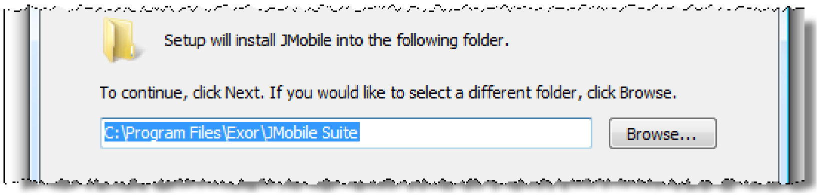
Click Next



● I accept the agreement
○ I do not accept the agreement

Figure 2

Read the JMobile Studio Software License and accept the agreement; follow then the instructions on the screen.
The default location for the JMobile Studio software is "C:\Program Files\Exor\JMobile suite". Default installation path can be changed depending on needs.



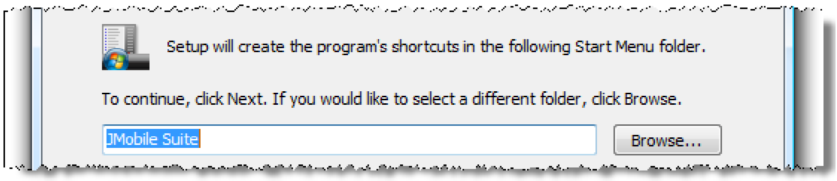
Setup will install JMobile into the following folder.

To continue, click Next. If you would like to select a different folder, click Browse.

C:\Program Files\Exor\JMobile Suite

Figure 3

The installation procedure will create a program group called “JMobile Suite" in the Start menu. A JMobile Studio icon can be added on the desktop. JMobile

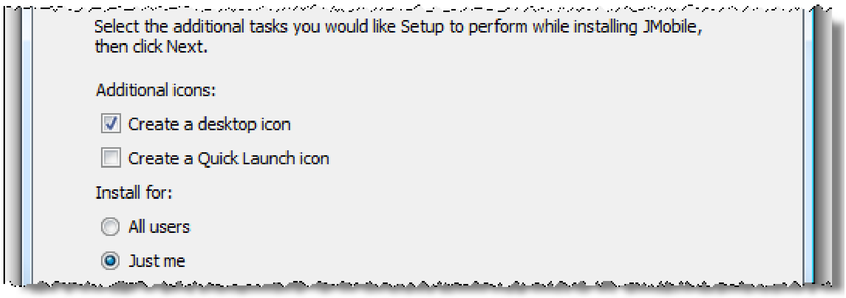


Setup will create the program's shortcuts in the following Start Menu folder.

To continue, click Next. If you would like to select a different folder, click Browse.

JMobile Suite

Figure 4



Select the additional tasks you would like Setup to perform while installing JMobile, then click Next.

Additional icons:

☑ Create a desktop icon

□ Create a Quick Launch icon

Install for:

○ All users

● Just me

Figure 5

After installing the JMobile Studio, you can run the Studio application by using the desktop icon or from Start – All programs- JMobile Studio.

1.3 JMobile Package

The JMobile Suite contains the following package as part of installation.

JMobile Studio:JMobile Studio is an application for designing custom HMI projects in a user-friendly manner, along with a variety of options in its built-in library, the Widget Gallery.

JMobile Windows Client: JMobile Client is a light-weight application that can be configured for mobile devices, HMI panels etc. Through the JMobile client, the device can be connected to the JMobile server.

JMobile HMI Runtime: JMobile UniOP Runtime is a standalone application that runs on the UniOP HMI's. It is provided in Studio to allow run time to update on Target devices.

2 Installing JMobile Runtime on UniOP Panels

All UniOP panels, Series 400, are produced and sold with JMobile Runtime pre-installed. JMobile Runtime cannot be turned off or deactivated.

Note: The following chapter explains how the JMobile Runtime can be installed on a UniOP Windows CE unit, in the case of an emergency situation or in the case that the unit flash disk requires formatting for some reason.

2.1 UniOP eTOP400 series Target Setup

Once JMobile Suite is installed on your PC, the Target files for the various operating systems will be in a folder called “**Runtime**", located in the JMobile Suite installation directory. Inside the **Runtime** folder, you will find these three subfolders: **UN20_WCE6 (MIPSIV_FP)**, **HMI** and **HMI Client**. Copy the “**UN20_WCE6 (MIPSIV_FP)**" folder to the HMI flash disk to use the Runtime server on HMI.

By default, the folder is in the following location: **C:\Program Files\Exor\JMobile Suite\Runtime**

To copy the files into the Target, you can use a USB pen-drive or Microsoft ActiveSync Connection (serial), or you can copy directly from the panel using the Ethernet connection, accessed through a shared folder on the PC (where JMobile Studio has been installed, and containing the “Runtime-UN20_WCE6 (MIPSIV_FP)" files).

Note: The folder MUST be renamed on the panel flash disk as, “qthmi".

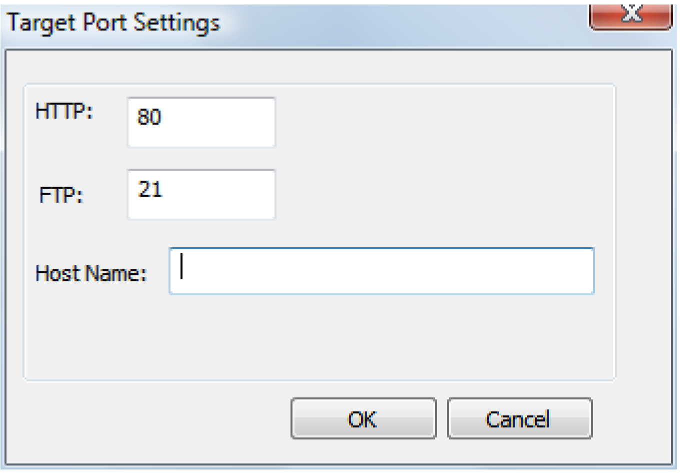
Note: JMobile Server requires an active Ethernet connection in order to properly operate. The Target unit must be accessible from the PC, at the location where JMobile Studio is installed, and the proper network parameters must be assigned to the panel.

Once the files are copied to the Target, start the executable file called “HMIce.exe".

2.2 Target Setup (Optional)

The Default port for the Target is set as 80. However, the user can set the port address for HMI. To set the port address, from JMobile Studio, click on the **Run-Download System Files** menu item, then click on **Target Setup.**

The Host Name can be defined by the user, in the appropriate box in the Target Port pop-up.



Target Port Settings

HTTP: 80

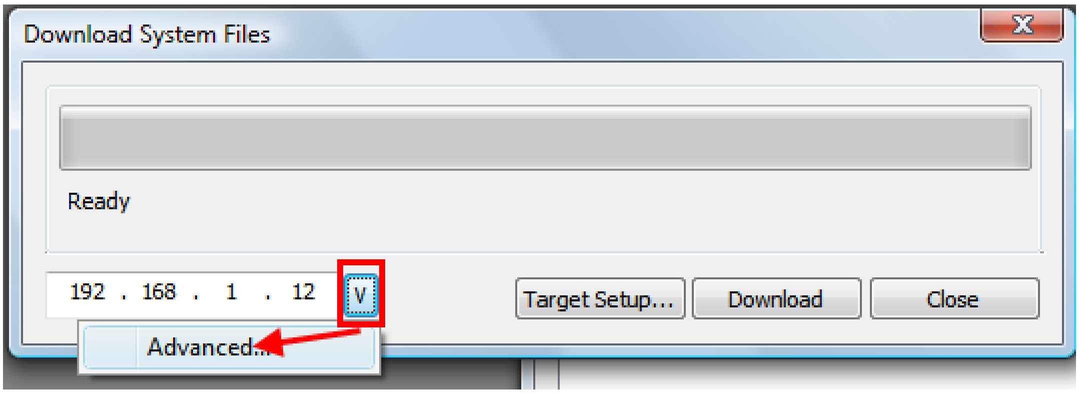
FTP: 21

Host Name:

Figure 6

2.2.1 Studio Port Setting

The Default Port for the server is 80. If the Target is running in a different port, Studio should still listen to the Target port. As such, you need to match the studio port with the Target port. In the download dialog, click on Advanced Menu and set Studio as the port.



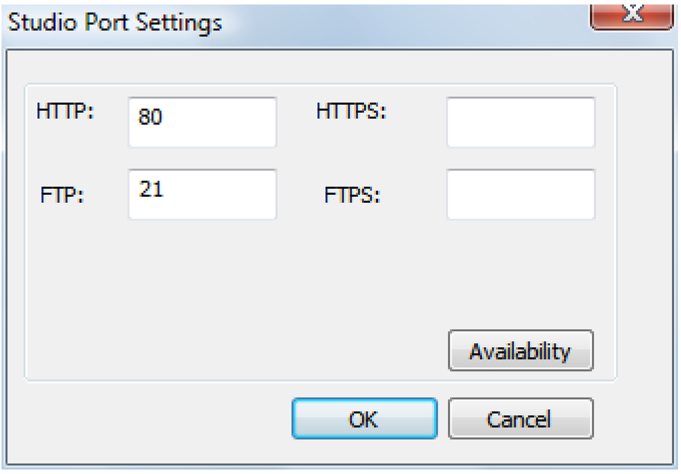
Download System Files

Ready

192 . 168 . 1 . 12

Figure 7

Set the HTTP/HTTPS port and FTP/FTPS port of the Target. They represent the port numbers used by the FTP(S) and the HTTP(S) server on Target. This is useful whenever default ports are, for some reason, in use by other applications or services, or if your local network requires using a different port setting.



Studio Port Settings

HTTP: 80

HTTPS:

FTP: 21

FTPS:

Figure 8

2.3 JMobile Server Runtime Modes

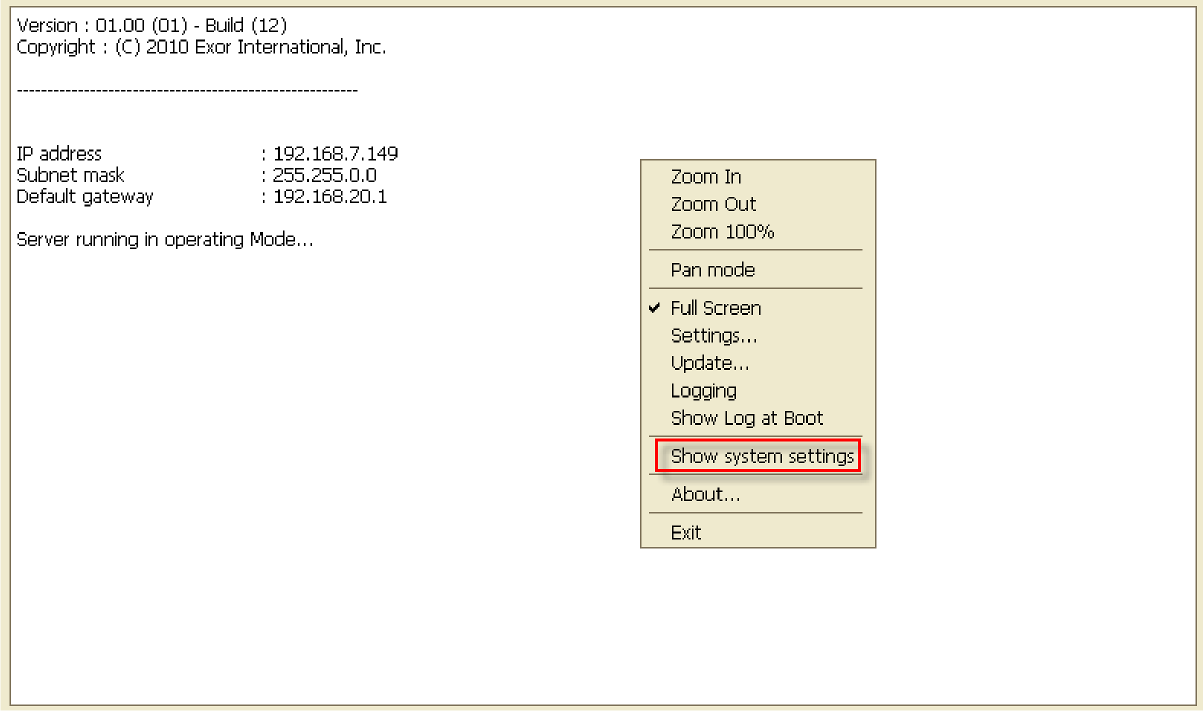
The JMobile Server supports the following two Operating Modes:

Configuration Mode: system is running with or without system files, but no project is running.
Operation Mode: server is running an application.

3 Basic Unit Settings: System Settings

System basic settings are available from Control Panel, which is accessible through the JMobile Configuration Mode.

Press and hold your finger on the screen for few seconds, until the context menu appears.



```
Version : 01.00 (01) - Build (12)
Copyright : (C) 2010 Exor International, Inc.

------------------------------------------------------------


IP address                    : 192.168.7.149
Subnet mask                   : 255.255.0.0
Default gateway               : 192.168.20.1

Server running in operating Mode...
```

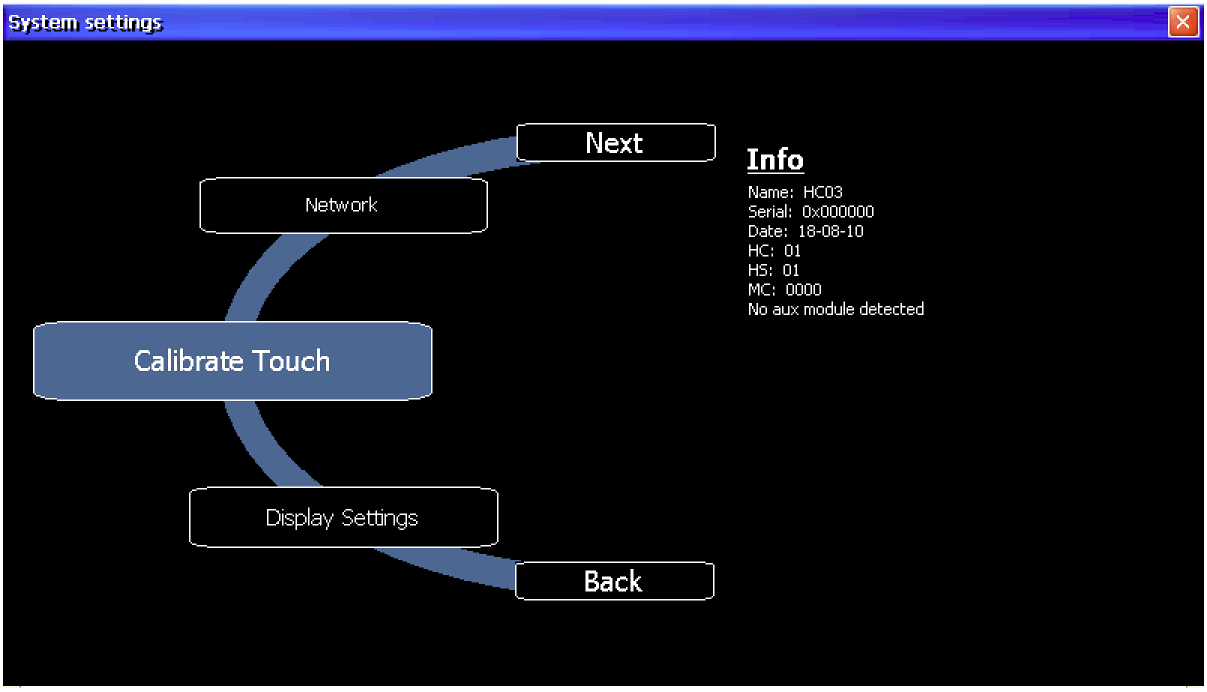
Figure 9

Touch on **Show system settings** to access the system settings tools.

The System settings tool is a rotating menu through which you can scroll using the “Next" and “Back" buttons. It includes the following entries:

Calibrate Touch	To calibrate the touch screen if needed
Display settings	Backlight and Brightness control
Time	Internal RTC settings
BSP Settings	Operating system version Unit operating timers: power up and activated backlight timers Buzzer control Battery LED control
Network	IP settings

Note: Settings selected and confirmed with the OK button in the upper right corner of the dialog are automatically saved to the registry.



Info

Name: HC03
Serial: 0x000000
Date: 18-08-10
HC: 01
HS: 01
MC: 0000
No aux module detected

Figure 10

3.1 Built-in SNTP Service

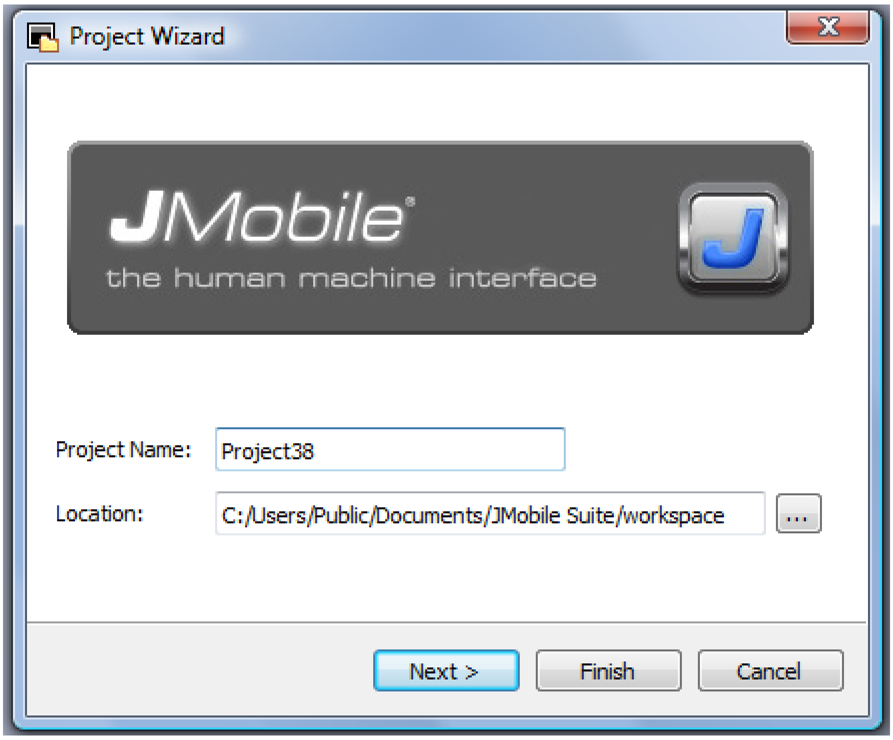
The UniOP Panels Operating System feature an integrated SNTP (**S**imple **N**etwork **T**ime **P**rotocol) that synchronizes the internal RTC panel whenever the predefined server is available.
The server addresses are hard-coded and cannot be changed by the user. The system searches for the following servers:
time.windows.com
tock.usno.navy.mil
SNTP servers are checked at power up, or once per week if the panel is not powered off.

4 My First JMobile Project

This section describes the steps to create a simple JMobile project.

4.1 Creating a New Project

To create a new project click on “File>>New Project" menu item.



Project Wizard

Project Name: Project38

Location: C:/Users/Public/Documents/JMobile Suite/workspace

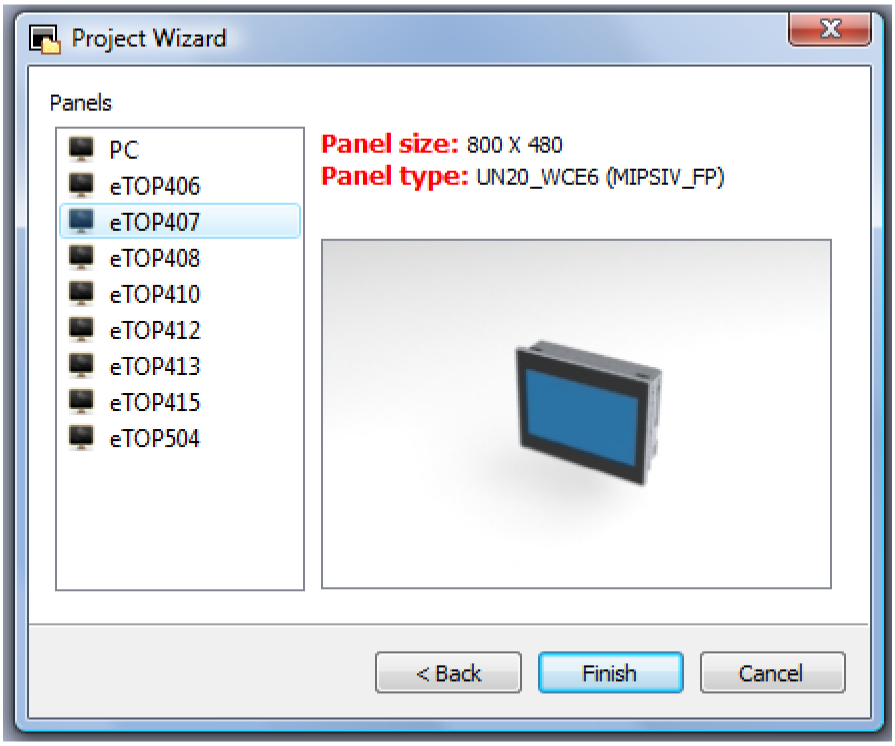
Figure 11

The Project Wizard dialog will appear, asking for a project name and a path where the corresponding project folder will be stored.

JMobile projects are stored in a folder that has the same name as the project. This folder contains all the project files. To move, copy or backup a project, you can simply move or copy the project folder and all its contents to the desired location.

Note: DO NOT rename the JMobile Project folders manually. If you need to rename a project, use the **File-Save Project As** function.

Click Next to go the panel selection dialog.



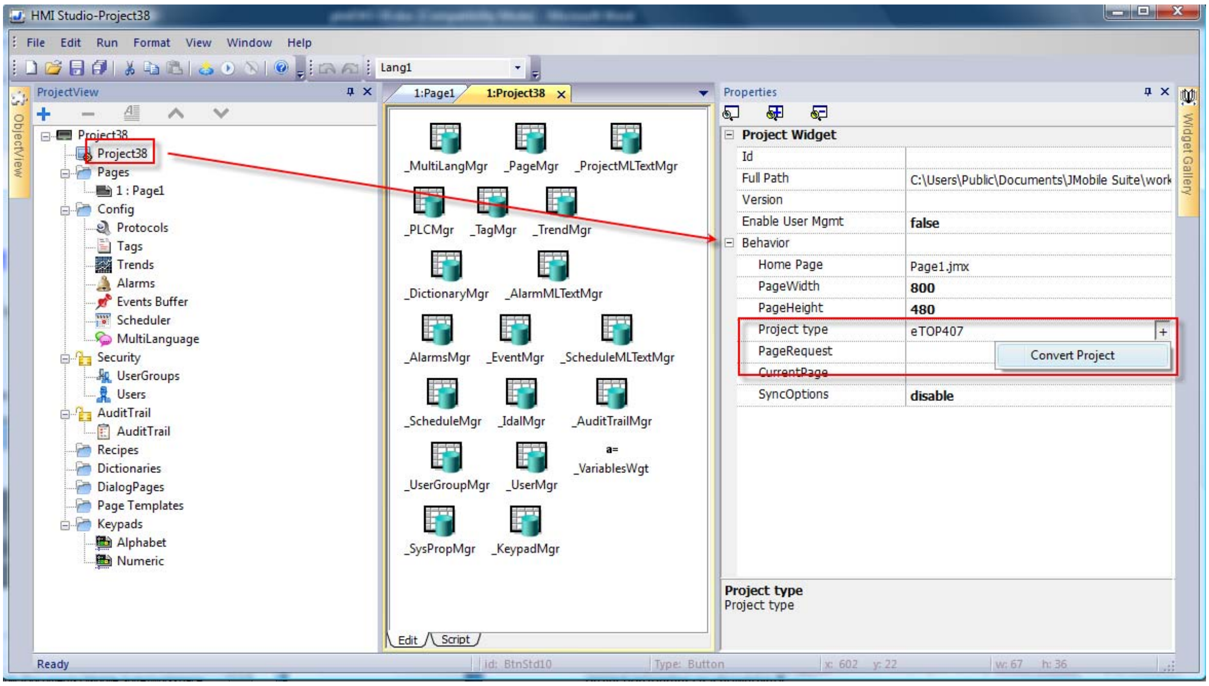
Panel size: 800 X 480
Panel type: UN20_WCE6 (MIPSIV_FP)

Figure 12

The panel selection is shown in Figure 12.
Here, you can scroll through a list of available units to select the panel model you are working with.

Click Finish to complete the Wizard.

Once the panel is chosen, you can convert the project to any other model, using the project properties portion of the screen, as shown below in Figure 13.



Project Widget

Id	
Full Path	C:\Users\Public\Documents\JMobile Suite\work
Version	
Enable User Mgmt	false
Behavior	
Home Page	Page1.jmx
PageWidth	800
PageHeight	480
Project type	eTOP407
PageRequest	
CurrentPage	
SyncOptions	disable

Project type
Project type

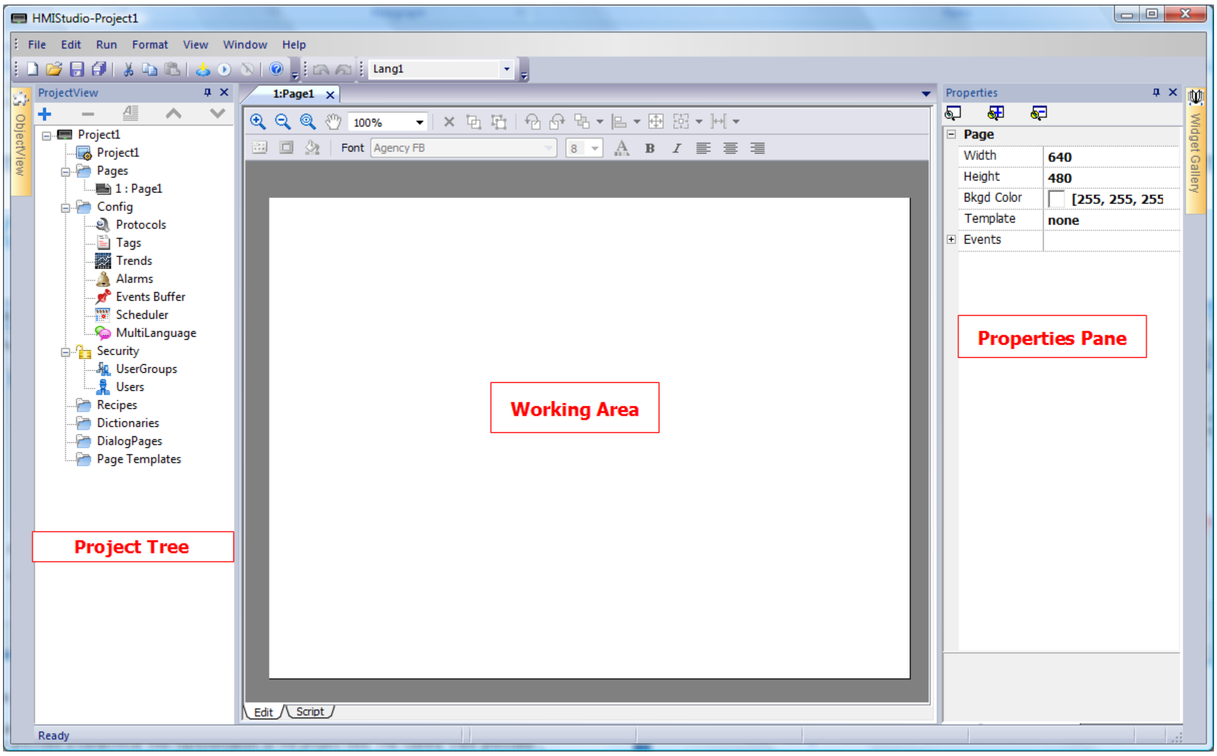
Figure 13

4.2 Workspace

The JMobile Studio workspace is divided into three main areas.
On the left-hand side of the window, you will find the Project View and Object View. Project View presents the project files in the form of a hierarchical Project Tree. Object View lists the Widgets with the corresponding ID's used in the page.

The center area is the main working space and is where editors create the HMI display and configure project data. Editor Views are indicated in a tab, at the top of the center area. You can quickly switch between different Editor Views by clicking on the desired tab.

The right part of the window shows the properties for the selected object, and on the very right side the Widget Gallery can be found as a slide in pane. The Widget Gallery provides a large library of symbols and graphics. When an object is selected, the object visual settings can be changed by changing the various properties in the Property View.



Page
Width 640
Height 480
Bkgd Color [255, 255, 255
Template none
Events

Figure 14

4.3 Step 1—Select the Communication Protocols

Device Communication drivers are configured in the “Protocol Editor", which is accessible from the project tree (as shown in Figure 15). Double click on the Protocols icon in the Project Tree view to open the Protocol Editor.

To add a driver, click on the “+" Icon and select the driver from the list in the controller field.

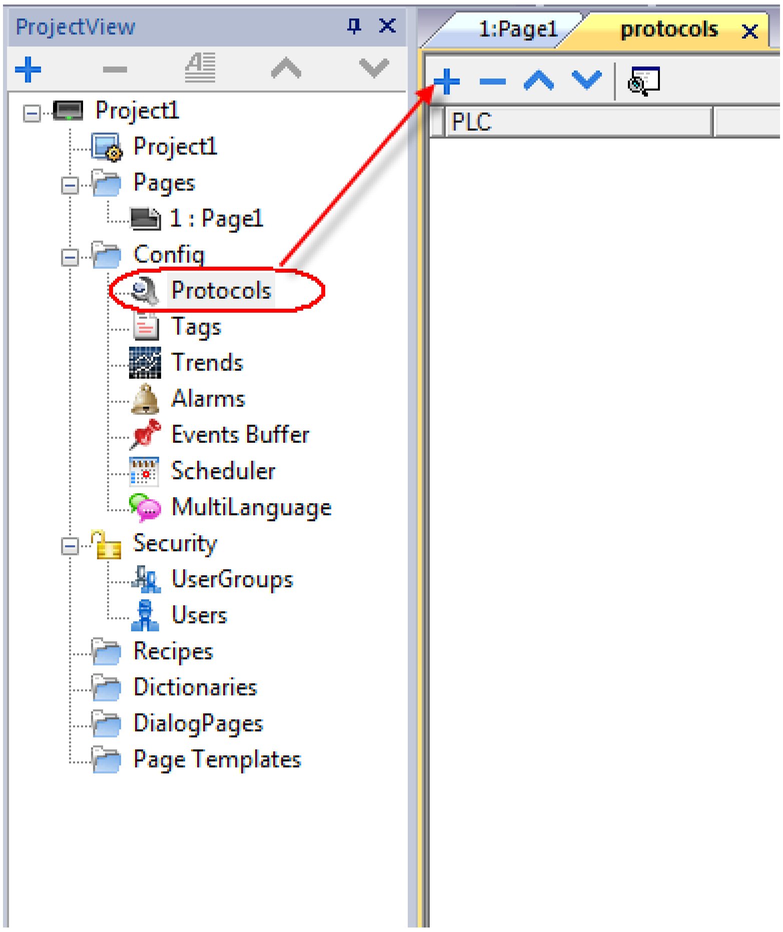
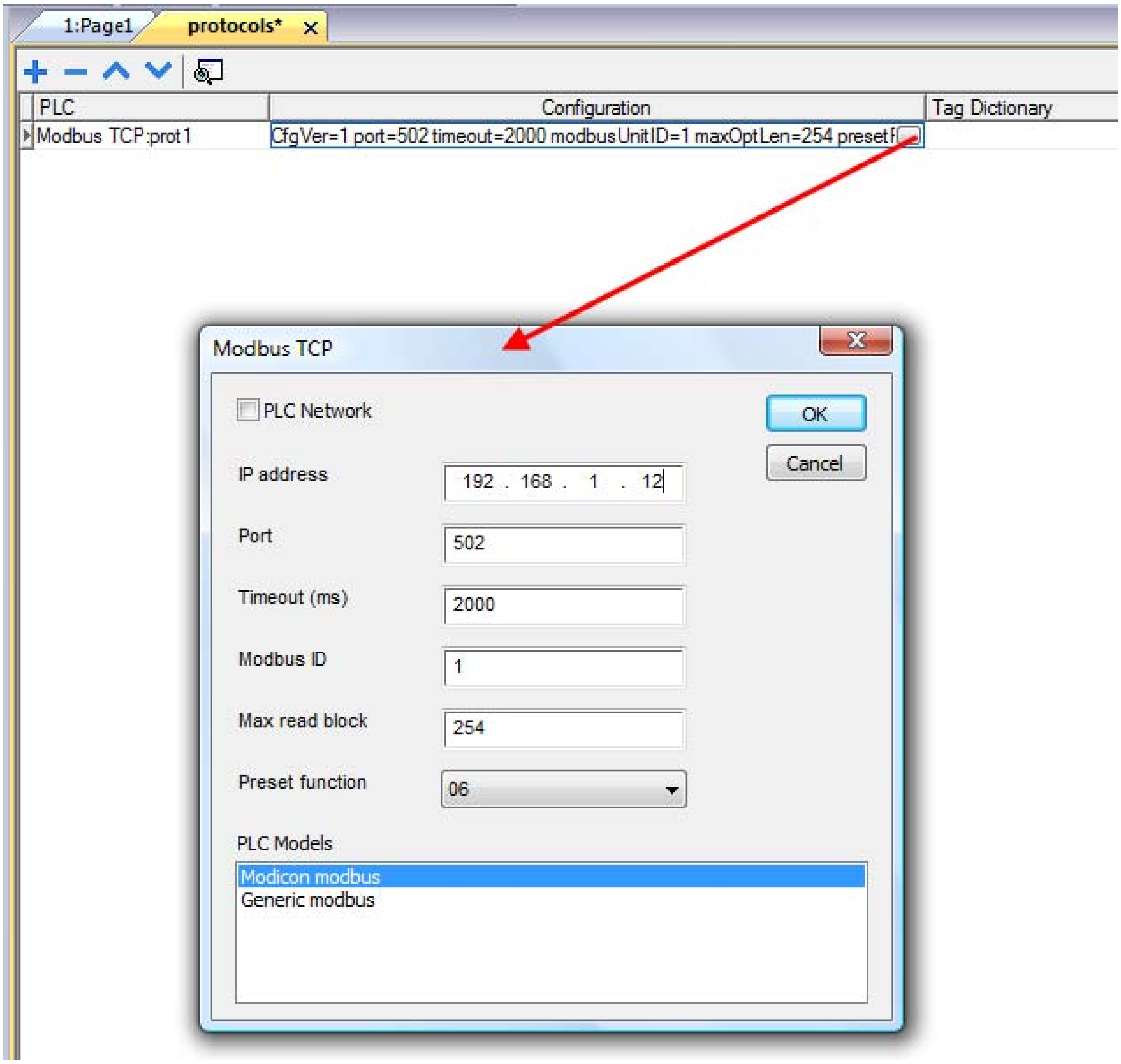


Figure 15

The combo box shows the list of available drivers. Once a driver is selected, configure the driver by clicking on the browse button in the Configuration field. A configuration dialog will be displayed, allowing you to set the parameters of the driver (as shown in Figure 16)

As an example, to create a project for Modbus TCP, you would select the Modbus TCP driver and then configure the communication parameters by selecting the browse button in the Configuration column.



PLC	Configuration	Tag Dictionary
Modbus TCP:prot1	CfgVer=1 port=502 timeout=2000 modbusUnitID=1 maxOptLen=254 preset	

Modbus TCP

□ PLC Network

IP address: 192 . 168 . 1 . 12

Port: 502

Timeout (ms): 2000

Modbus ID: 1

Max read block: 254

Preset function: 06

PLC Models:
- Modicon modbus
- Generic modbus

Figure 16

JMobile supports multiple protocol configurations. By repeating the steps previously outlined, you can add up to two protocols in the protocol editor.

4.4 Step 2 – Add the Tags

JMobile 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. To assign Tags, double click on the Tags icon in the Project View and the Tag Editor will be displayed (as shown in Figure 17).

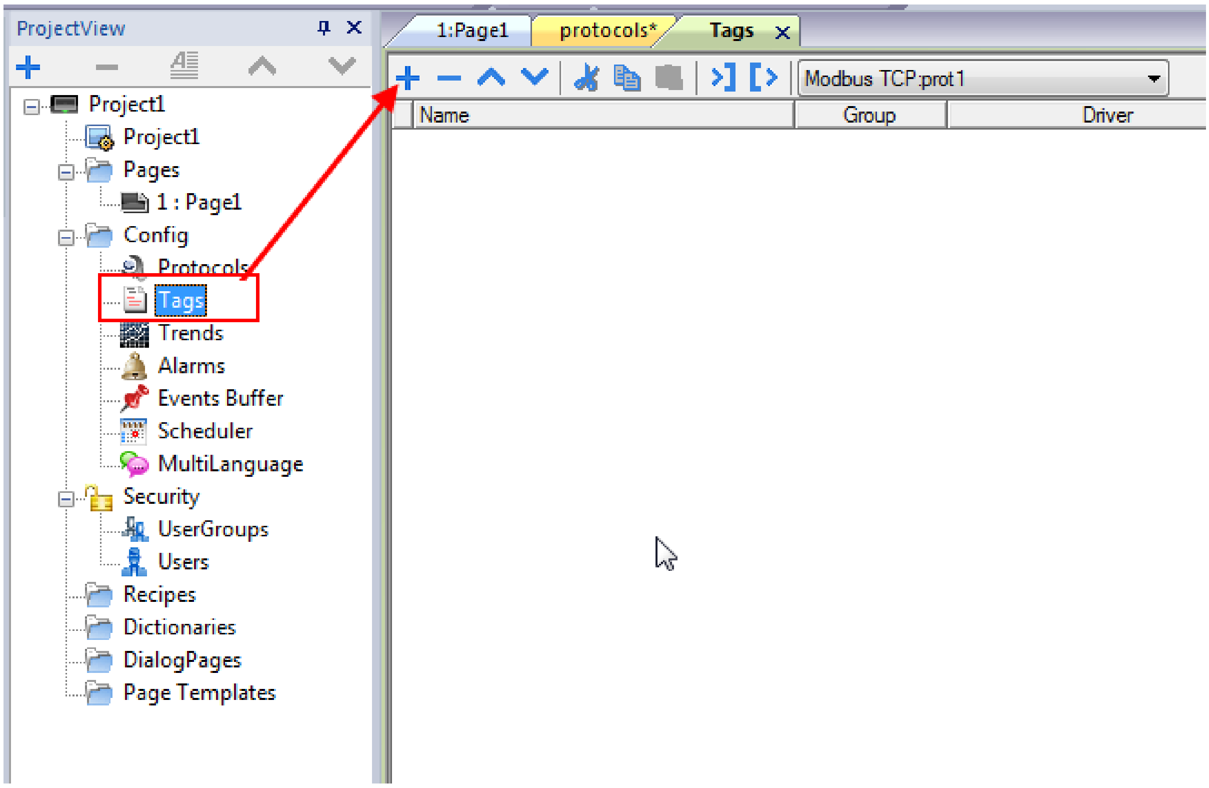


Figure 17

To add a new Tag, click on the “+" icon, and Select the Address from the Communication protocol address dialog. When Tags are initially added, these Tags are named Tag1, Tag2, etc., by default. The user can rename the Tag with the appropriate name by clicking once on the Tag name.
Tag Editor in JMobile Studio provides the Tag Import feature, which is available based on the protocol selected. Not all protocols support Tag Import.
If the protocol does support this feature (see specific Protocol documentation), first select the Protocol from the filter button and then click on the Import button (as shown in Figure 18).

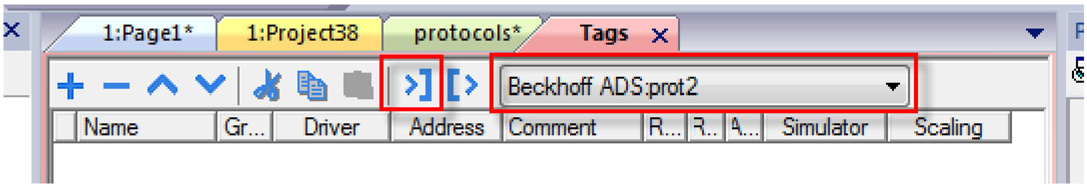


Figure 18

You will see the dialog that corresponds to the protocol selected, which prompts you to browse for the symbol file. The symbol file is exported by the controller programming software.

4.4.1 JMobile Tag Editor

The tool in JMobile Studio designed to handle tags' creation and managing is called Tag Editor. Per each tag the tag editor allows you to specify several properties.

Name
This is the unique name at project level of the tag. This is the primary key used to identify the information in the internal runtime tag database. Note that you cannot use the same tag name even if you are referring to different communication protocols.

Group

After the tags have been defined in the Tag Editor, they are used in the project screens by attaching them to the widgets' properties (see chapter 5.1 for a complete explanation).

Per each screen the system is able to identify which tags are used in the specific page and identifies them as part of the "page group". This allows an easy handling at run time of the requests made by the communication protocol to the connected controller(s): only the tags included in the displayed page are queued for retrieval from the controller memory.

This mechanism is fully automatic and there is no intervention required by the user.

The tag editor allows anyway to define groups of tags not belonging to a specific page but for instance grouped according to their logical meaning.

We can call these groups "Users' group". Users' groups have no meaning for the local visualization, but they are very useful when an external software communicate with the local JMobile runtime requesting sets of data that must be independent from the currently displayed screen.

The JMobile web server publishes in fact a set of communication interfaces that can be used from a 3rd part application to interface with the local tag database and read the tags according to their grouping.

The group column allows to define the users' groups and assign tags to them.

Driver

Specifies the communication protocol for which the tag is going to be defined.

Address

This shows the PLC controller memory address. To edit it, click on the right side of the column to get the dialog box where you can enter the address informations.

Comment

Allows to add a description of the tag.

R/W

If you need to specify already at tag editor level that a certain tag must never be written you can set this property to R. Any write operation to this tag will be ignored. This is useful when you need to be sure that the tag is never written.

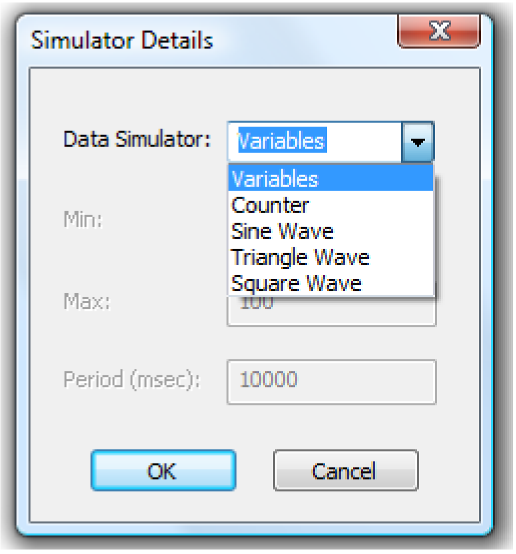
Active

As explained above tags are grouped per page and if needed in users' groups. By default tags are not active; this means they are automatically activated by the runtime when the visualization requires them. Anyway, you can force the system to continuously read a certain tag even if not present in current page by setting its Active property to true.

We recommend to leave this parameter to false to avoid unexpected results in terms of overall device performances.

Simulator

JMobile Studio provides off-line simulation. The behaviour of each tag during simulation mode can be specified here by choosing between several profiles as shown in Figure 19.



Simulator Details

Data Simulator: Variables

- Variables
- Counter
- Sine Wave
- Triangle Wave
- Square Wave

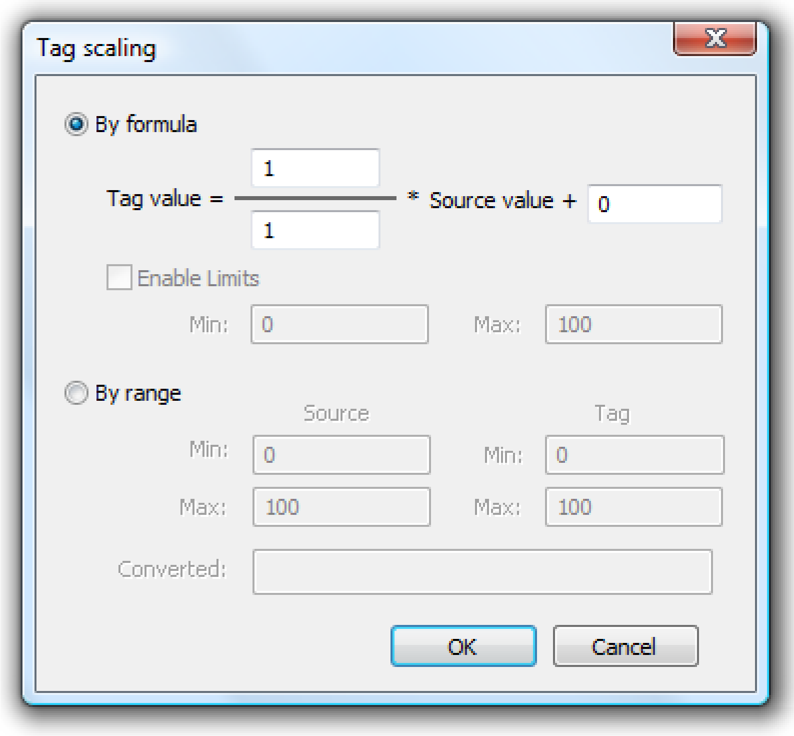
Min:

Max: 100

Period (msec): 10000

Figure 19

Scaling
Tags' values are normally transferred "as they are" from the protocol top the real time tag database. Anyway, you can specify to apply a scaling to the tag values before they are stored in the database. The available scaling options are shown in Figure 20. Scaling can be specified in terms of linear relationship as formula or as range.



Tag scaling

● By formula

Tag value = 1 / 1 * Source value + 0

□ Enable Limits

Min: 0 Max: 100

○ By range

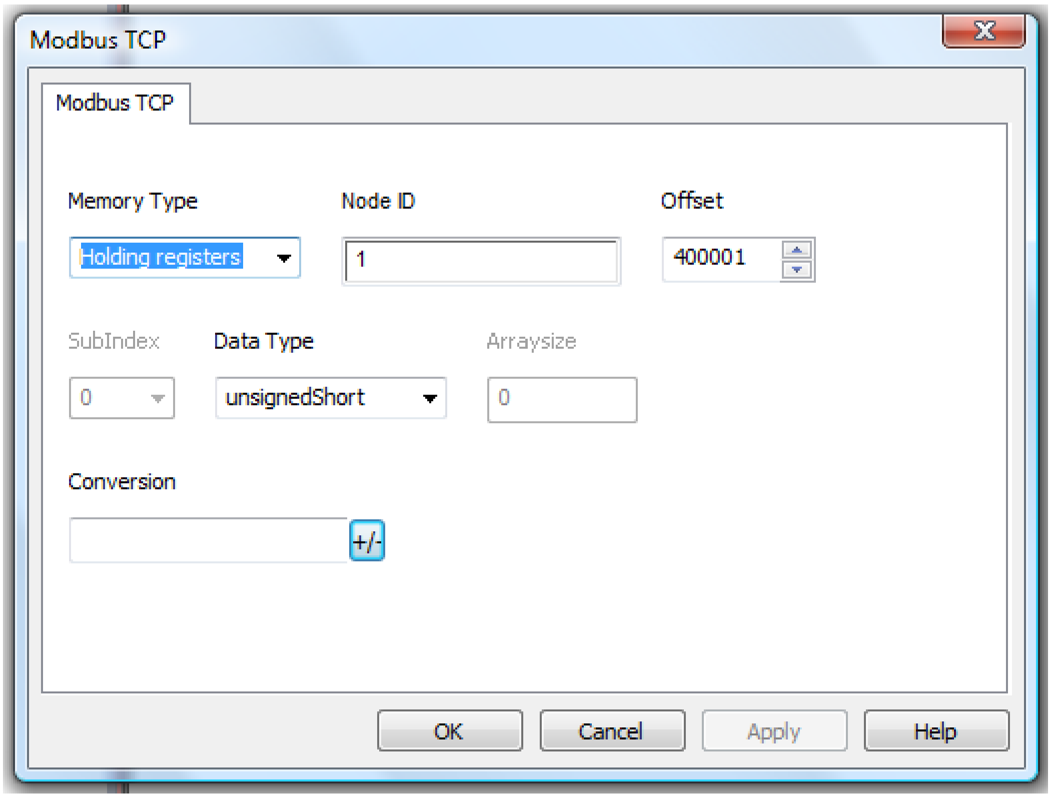
	Source		Tag
Min:	0	Min:	0
Max:	100	Max:	100

Converted:

Figure 20

4.4.2 JMobile Data Types

When creating a Tag, JMobile shows a dialog in which you need to specify Tag details. The Tag’s Memory Types are specific for the selected Protocol.



Memory Type: Holding registers

Node ID: 1

Offset: 400001

SubIndex: 0

Data Type: unsignedShort

Arraysize: 0

Conversion:

Figure 21

The Tag's Data Type must be selected from the list of available JMobile Data Type, according to the JMobile internal representation you need for the selected controller address. JMobile Data Types are summarized in the following table.

Data Type	Description
string	String represents character strings.
boolean	Boolean is one bit data
float	Float corresponds to the IEEE single-precision 32-bit floating point type
double	Double corresponds to IEEE double-precision 64-bit floating point type
binary	Binary represents arbitrary binary data
int	Int is signed 32 bit data
short	Short is signed 16 bits data
byte	Byte is signed 8 bits data
unsignedInt	UnsignedInt is unsigned 32 bit data
unsignedShort	UnsignedShort is unsigned 16 bit data
unsignedByte	UnsignedByte is unsigned 8 bit data
time	Time data
boolean [ ]	Array of Boolean
byte [ ]	Array of byte
short [ ]	Array of short
int [ ]	Array of int
unsignedbyte [ ]	Array of unsignedbyte
unsignedshort [ ]	Array of unsignedshort
unsignedint [ ]	Array of unsignedint
float [ ]	Array of float
double [ ]	Array of double
time [ ]	Array of time

Table 1

4.5 Step 3 – Create a Page

When a project is created, a page is automatically added to the project and shown in the Page Editor. To add objects to a page, simply drag and drop the object from the Widget Gallery to the page.

4.6 The JMobile Widgets Gallery

The Gallery is adjacent to the Property View panel and can be opened by clicking on the Widget Gallery tab (as shown in Figure 22).

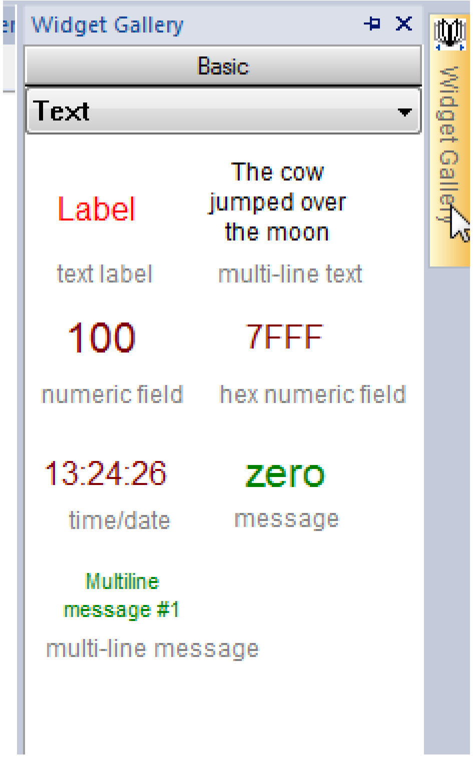


Figure 22

Select the desired object from the Widget Gallery, then drag and drop it to the page. To change the appearance of the object, select the desired property from Property View and change the property setting.
All the HMI objects required to make an application are collected in the Widget Gallery. The Widget Gallery is accessible as a slide in pane from the right side of the workspace (as explained in the previous chapter).
The gallery is divided into several categories, each with collections of different types of objects. Click on a category to display its sub-categories.
For each sub-category, the gallery offers the option of applying different styles to the objects within that category (when possible).
Figure 23 shows the Widget style button for round gauges.

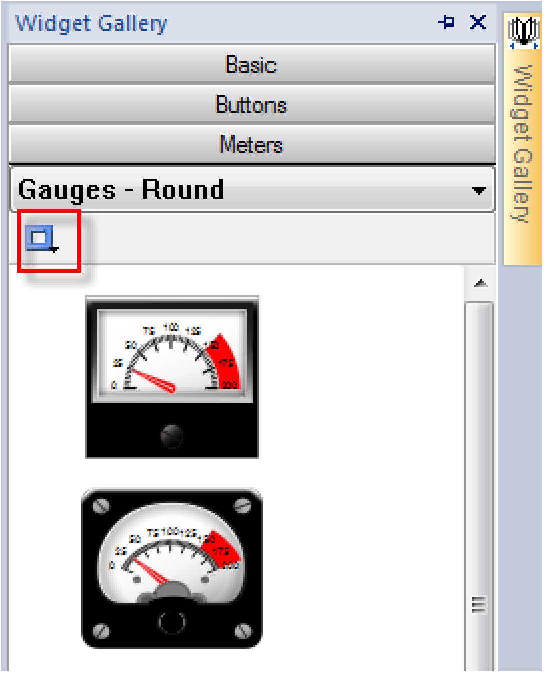


Figure 23

Clicking on the style button will display the available styles for the current object.
Select one of the available styles to apply it to the gallery objects.
The object will then be inserted on the page, with the new applied style.
Once on page, the object can still be subject to additional style changes.
This is done using the Page Toolbar shown in Figure 24.

Note: style change may not be available for all the widgets

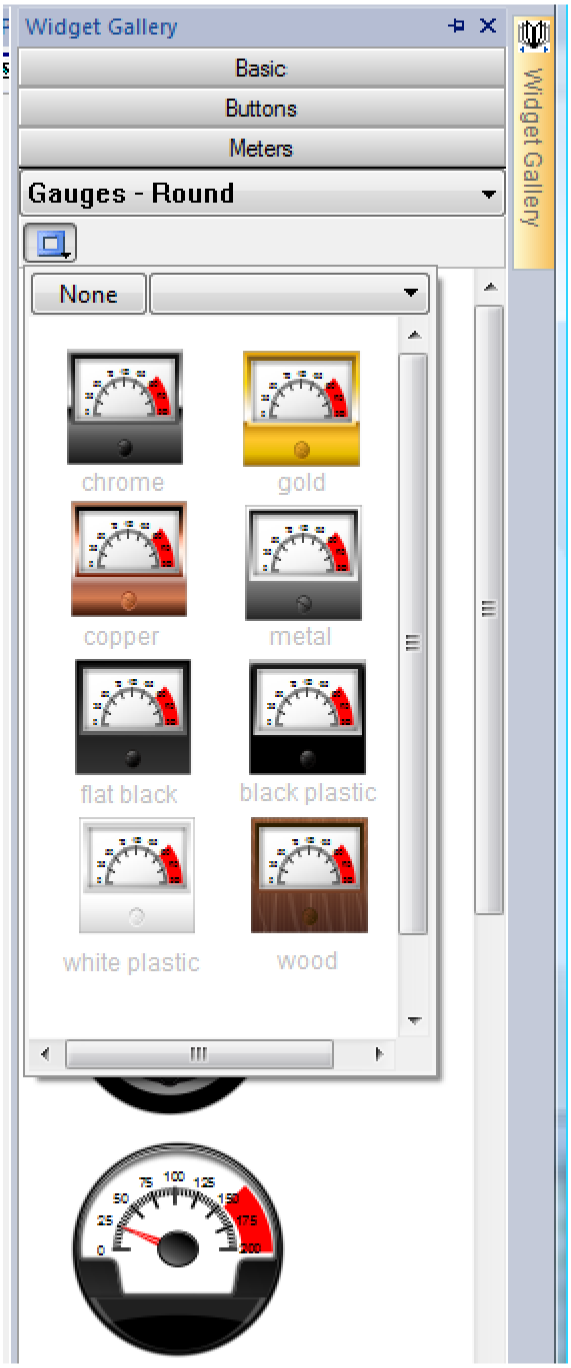

Widget Gallery
Basic
Buttons
Meters
Gauges - Round
None
chrome
gold
copper
metal
flat black
black plastic
white plastic
wood
Widget Gallery
0
25
50
75
100
125
150
175
200

Figure 24

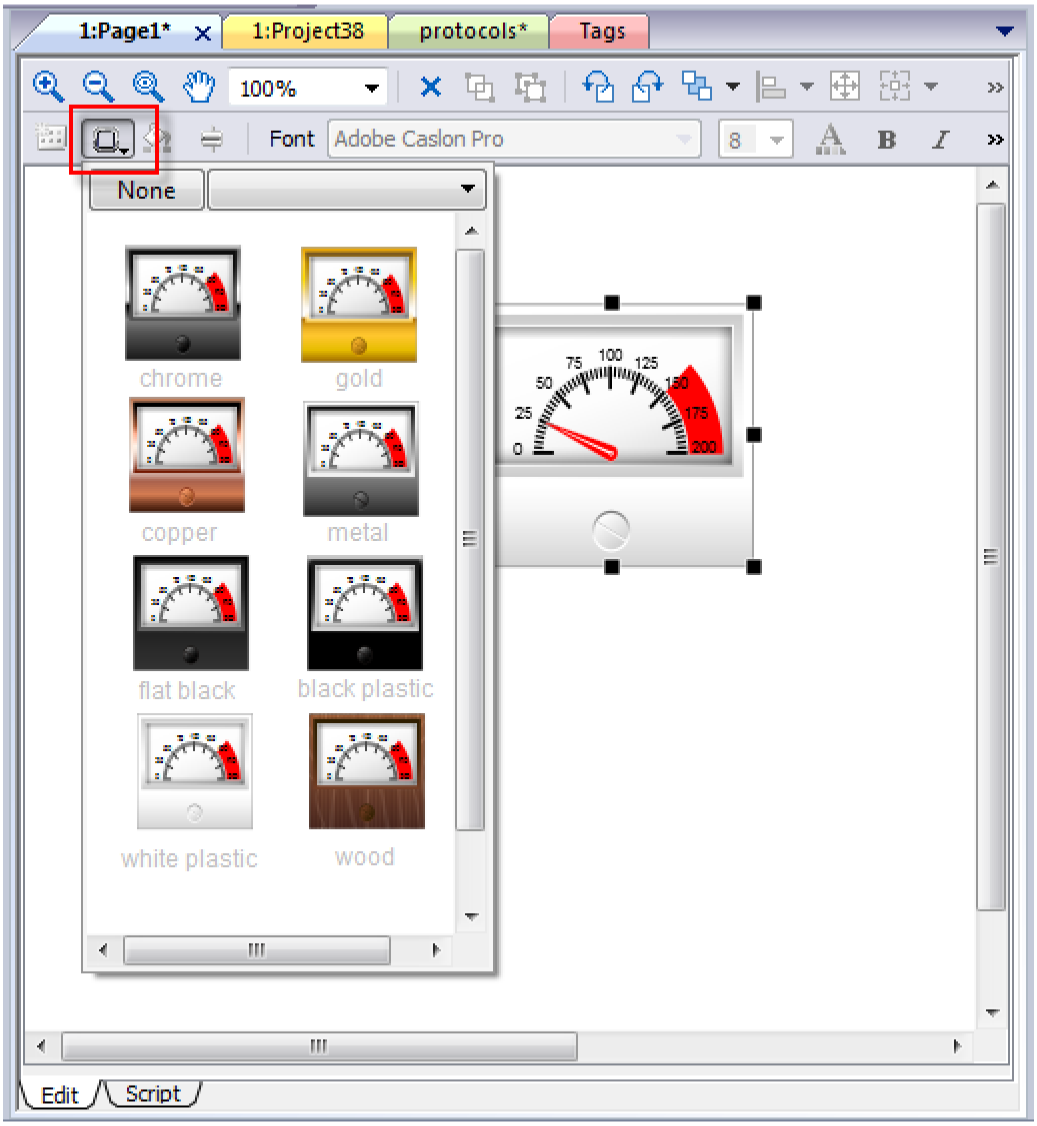


Figure 25

4.7 Step 4 – Attach Data to Objects

JMobile Studio allows simple binding between Tags and Widget Properties. Many different Widget Properties can be attached to a Tag, which allows you to control the device and animate objects based on live data.

To attach a Tag to a property, click on the property in Property view. A + button will be displayed on the right side of the property field. Click on this button and select the item Attach Tag from the menu (as shown in Figure 26).

For example, when working with a gauge object, the most common action taken by the programmer is to attach a Tag to the needle, so that the value of the Tag referenced in the controller memory is represented by the needle movement.

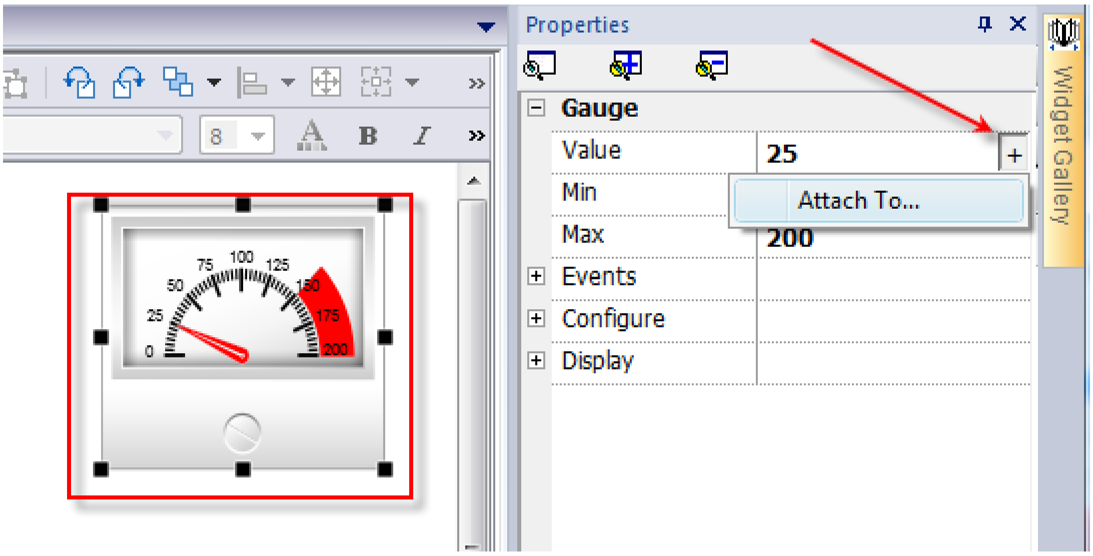
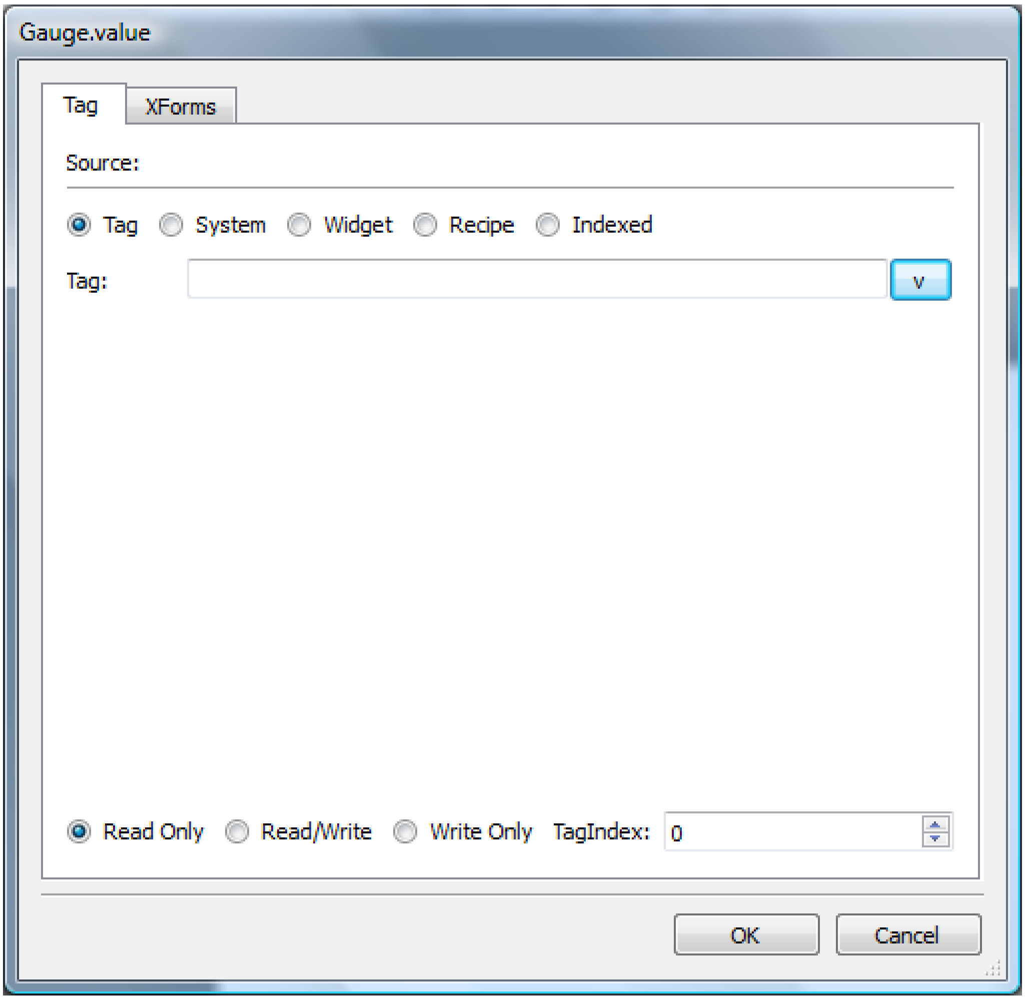


Figure 26

To attach the Tag to the needle, single click on the object to display its properties in the Property view. Locate the “Value" property and click on the + button on the right part of the field as shown in Figure 26. Select the Attach Tag menu item and a dialog will be displayed as shown in Figure 27.


Gauge.value

Tag | XForms

Source:

● Tag ○ System ○ Widget ○ Recipe ○ Indexed

Tag:

● Read Only ○ Read/Write ○ Write Only TagIndex: 0

Figure 27

When attaching a Tag, you can attach five types of data sources: Tag, System Tag, another object property (called a Widget), Recipes and Indexed types.
Select the ‘Tag’ source type to attach to a Tag defined in the Tag editor.
Select the ‘System’ source type to attach to a system Tag
Select the ‘Widget’ source type to attach to data from another object.
Select the ‘Recipe’ source type to attach to Recipe data from Recipe manager.
Select the ‘Indexed’ source type to attach to Indexed references

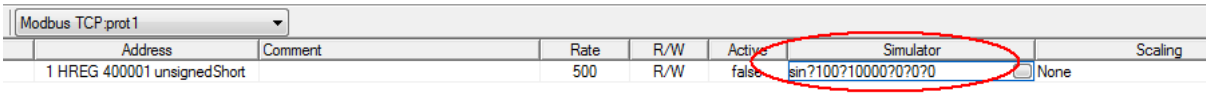
Now select the Tags from the Tag Name combo and Click OK.

Tags can be attached to many different properties of the object. You can attach a Tag to a different property by selecting the property in the Property view and clicking on the Attach Tag popup menu. You can also right-click on the object and select the Attach Tag menu item. The “Attach To" dialog will be displayed and you can select the desired property from a list on the right part of the dialog.

4.8 Step 5 – Testing the Project

With JMobile you can test the project functionality before downloading it to the Target device. JMobile provides an internal simulator that generates data and simulates the Target operation.

When defining Tag values, the Tag Editor also includes a field to select a method for *simulating* the data as shown in Figure 28. Tag values can be simulated in the following ways:
Variables: The data is stored in a variable in the simulator. This variable holds the value of the Tag so the client can read and write to the Tag value.
Counter: A count value is incremented from 1 to 1000. When the counter reaches 1000, the value is reset to 0 and the counter restarts.
Sine Wave: A sine wave value is generated and written to the Tag value. The Min, Max and Period values of the Sine wave can be defined for each Tag.
Triangle Wave: A triangle wave value is generated and written to the Tag value. The Min, Max and Period values of the wave can be defined for each Tag.
Square Wave: A square wave value is generated and written to the Tag value. The Min, Max and Period values of the Sine wave can be defined for each Tag.



Modbus TCP:prot1

Address	Comment	Rate	R/W	Active	Simulator	Scaling
1 HREG 400001 unsignedShort		500	R/W	false	sin?100?10000?0?0?0	None

Figure 28

The JMobile Simulator is launched from JMobile Studio. Select the Run-Start Simulator menu item to start the Simulator. At this point, the simulator is running locally on the PC in the same way that JMobile runtime runs on a panel or Target device.

In Preview mode, the project runs the same way it would run on the panel. The Controller date is provided in the simulator. You can click buttons, change pages, view live data and test the project before downloading it to the panel.

To stop the simulator, select the item Run-Stop Simulator from the menu.

4.9 Step 6 - Transferring the Project to Target

After successfully downloading system files, the HMICE is ready to run the projects.
The JMobile project can be transferred to the JMobile Server Target system from JMobile Studio using the “Download to Target" item in the Run Menu.

The Download to Target dialog is shown in Figure 29. Click on the **Download** button to start the process. The system will switch the Target to Configuration mode and transfer the files. When the download operation is completed, the Target is automatically switched to Operation mode and the downloaded project is started.

Any time a project is changed, the modified files need to be transferred to the Target device. When updating a Target, JMobile provides the option “Download only changes" to transfer only the modified files to the device.

Project folders can be transferred to and from JMobile Servers using a standard FTP client program. This allows the project to be managed and updated in a simple and uniform manner.

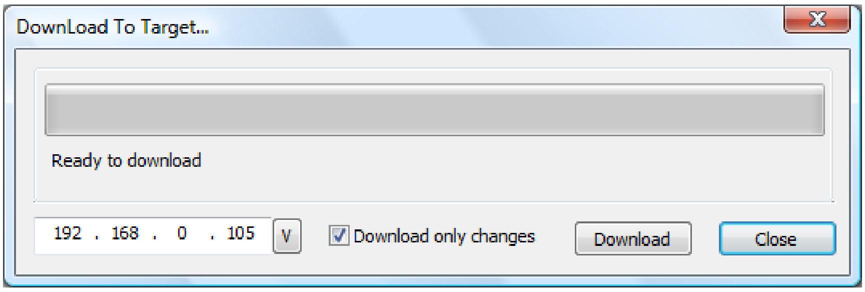


Figure 29

4.10 Using JMobile Client

The JMobile Client provides remote access to the JMobile Server, and is included in the JMobile Suite installation. The JMobile client consists of a simple standalone application; although it uses the same graphic rendering system as the server, it relies on a specified JMobile Server for live data.

JMobile Client for Windows is available in the Runtime folder of the JMobile Suite root folder. Execute the JMobile Client application from the Runtime folder or from the start up menu (JMobile Suite-Windows Client). Client will open in a browser-like style. Type the server IP address in the address bar (for example: http://192.168.1.12). JMobile Client will connect to the server and the same application will be loaded to JMobile Client with the same graphical interface.

JMobile Client acts as remote client and communicates to the server, sharing with the local visualization those Tag values that are maintained updated by the communication protocol.

4.11 Using ActiveX Client for Internet Explorer

In the standard distribution of JMobile Suite, a JMobile Windows Client and JMobile ActiveX Client are provided.
ActiveX components are NOT installed by default to the Target devices, in order to save space in the flash memory.

Note: This ActiveX requires Microsoft Visual C++ 2008 Redistributable Package (x86) installed in your system. You may want to download the Download Microsoft Visual C++ 2008 Redistributable Package (x86) from the Microsoft web site.

4.11.1.1 Copy ActiveX into the Target device

The ActiveX component is distributed with the JMobile Suite installation package. The related files are located in the Runtime folder of the JMobile Suite installation directory. The files, “HMIAX.cab" and “HMIClientAX.html", should be copied into the workspace folder of the Target device, where Runtime is installed.

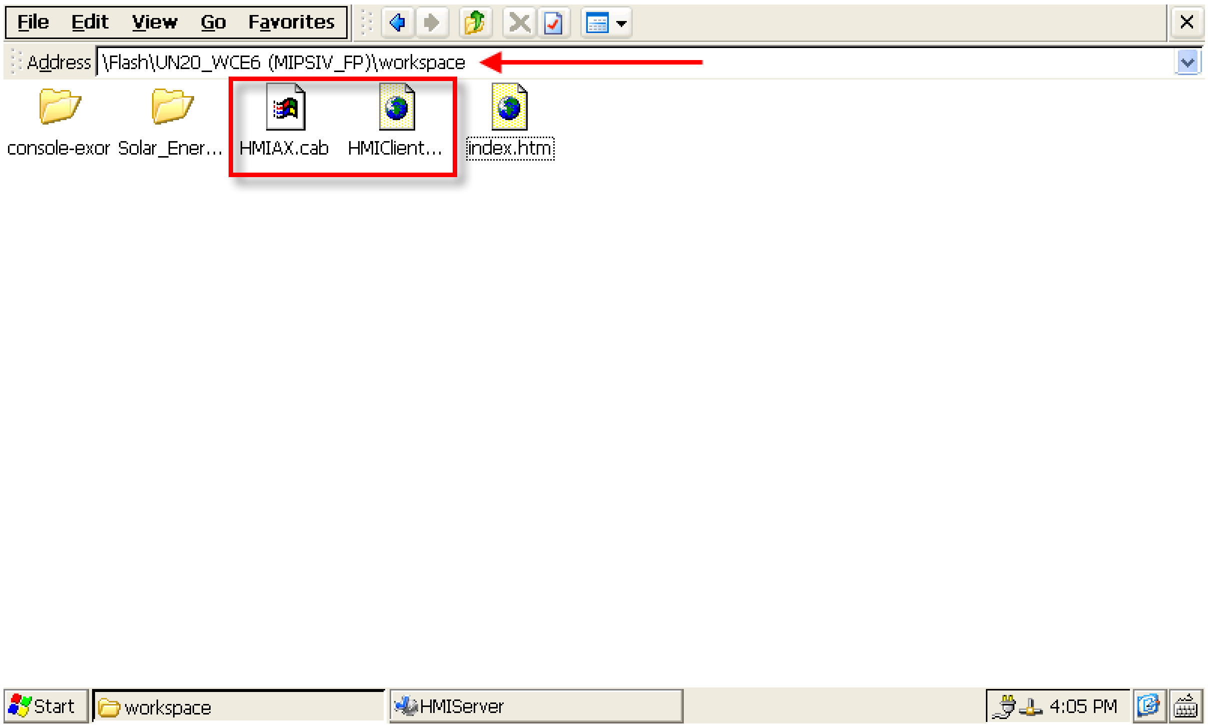
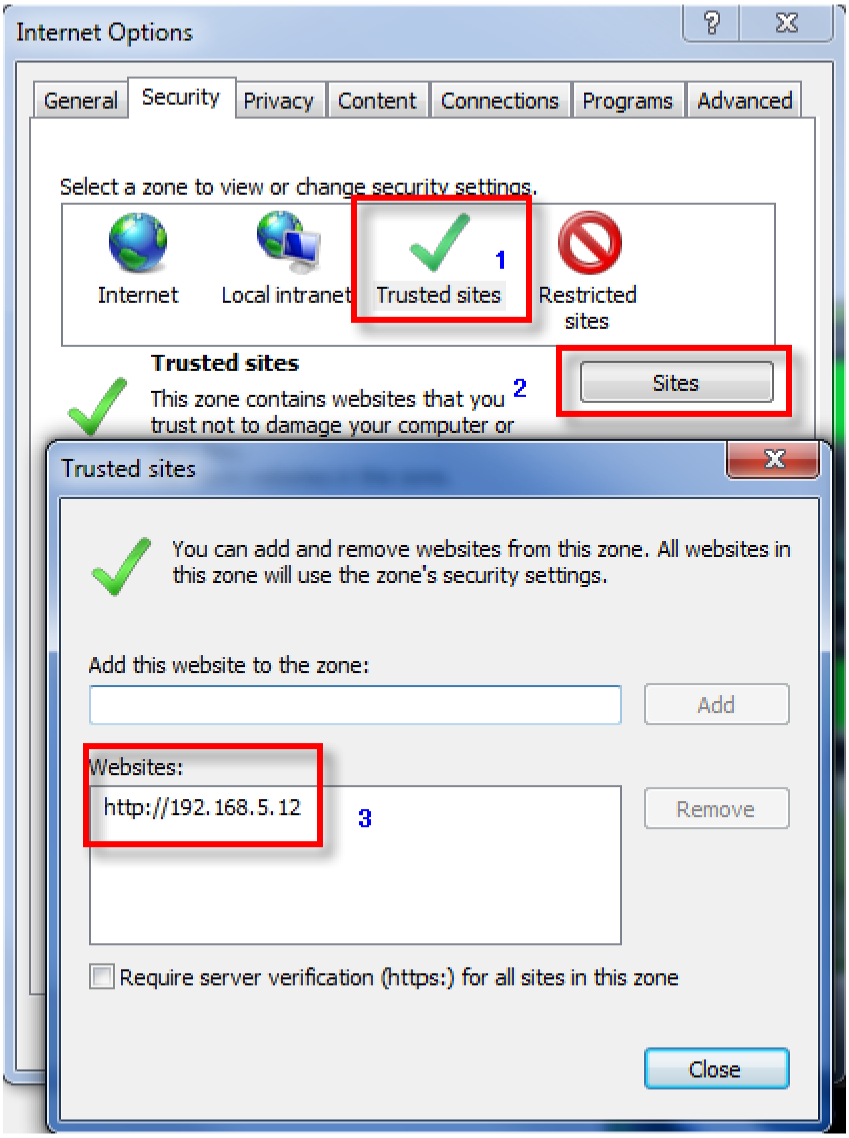


Figure 30

4.11.1.2 Internet Explorer Settings

Internet Explorer settings must be changed, adding the panel’s IP to the list of the trusted sites.
In Tools – Internet Option Security tab choose, “Trusted sites". Then click on the “Sites" button.
Type in the IP address to the Target device, at the location where the ActiveX component has been installed and will be loaded to the browser.



Select a zone to view or change security settings.

Trusted sites

This zone contains websites that you trust not to damage your computer or

You can add and remove websites from this zone. All websites in this zone will use the zone's security settings.

Add this website to the zone:

Websites:

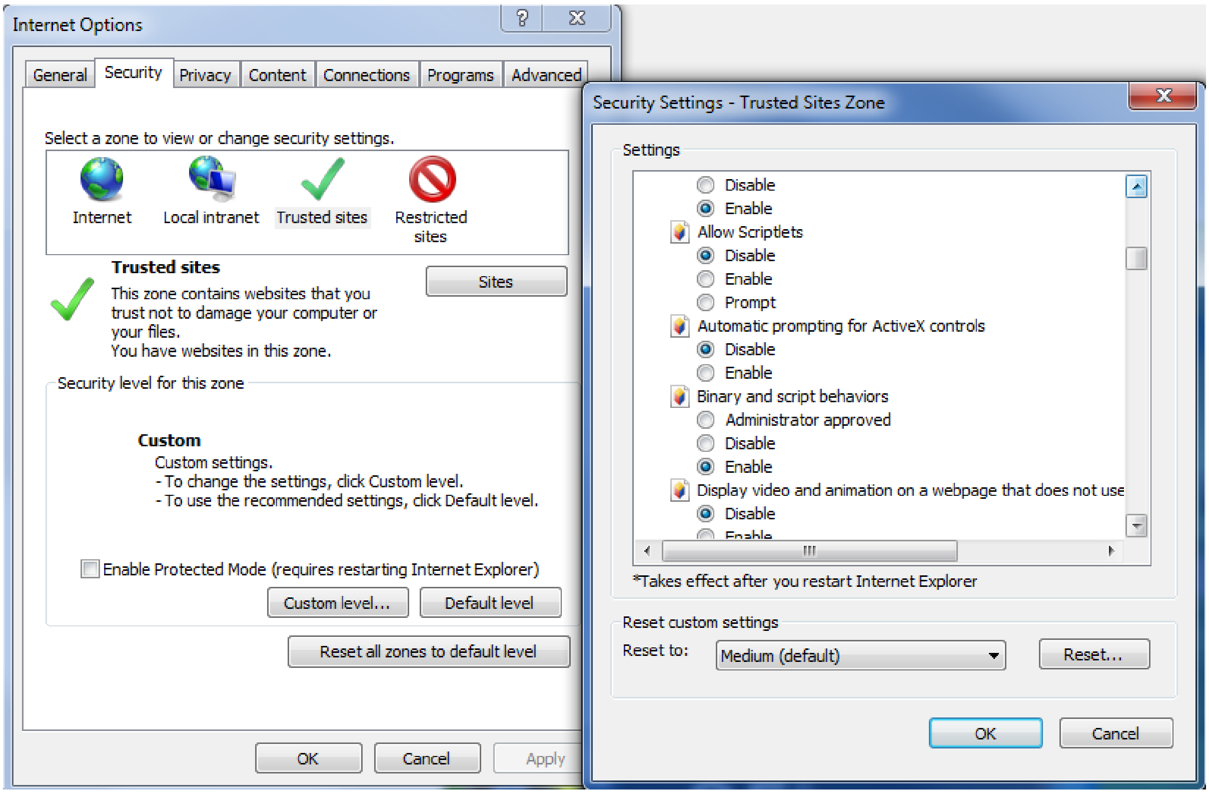
http://192.168.5.12

☐ Require server verification (https:) for all sites in this zone

Figure 31

4.11.1.3 Security Setting for Trusted Site Zone

Set your Internet Explorer Browser as seen in the following images:



Internet Options

General | Security | Privacy | Content | Connections | Programs | Advanced

Select a zone to view or change security settings.

Internet
Local intranet
Trusted sites
Restricted sites

Trusted sites

This zone contains websites that you trust not to damage your computer or your files.
You have websites in this zone.

Security level for this zone

Custom
Custom settings.
- To change the settings, click Custom level.
- To use the recommended settings, click Default level.

□ Enable Protected Mode (requires restarting Internet Explorer)

Security Settings - Trusted Sites Zone

Settings

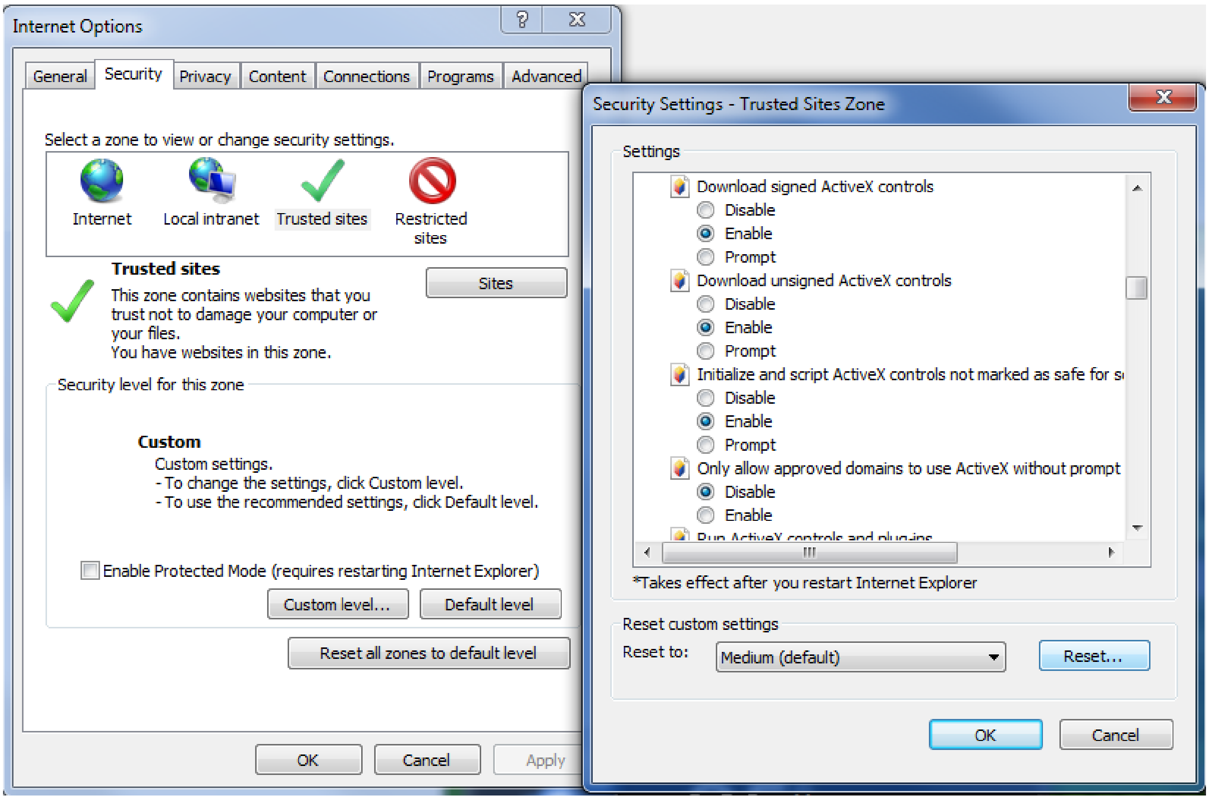
○ Disable
● Enable
Allow Scriptlets
● Disable
○ Enable
○ Prompt
Automatic prompting for ActiveX controls
● Disable
○ Enable
Binary and script behaviors
○ Administrator approved
○ Disable
● Enable
Display video and animation on a webpage that does not use
● Disable
○ Enable

*Takes effect after you restart Internet Explorer

Reset custom settings

Reset to: Medium (default)

Figure 32



Internet Options

General | Security | Privacy | Content | Connections | Programs | Advanced

Select a zone to view or change security settings.

Internet
Local intranet
Trusted sites
Restricted sites

Trusted sites

This zone contains websites that you trust not to damage your computer or your files.
You have websites in this zone.

Security level for this zone

Custom
Custom settings.
- To change the settings, click Custom level.
- To use the recommended settings, click Default level.

□ Enable Protected Mode (requires restarting Internet Explorer)

Security Settings - Trusted Sites Zone

Settings

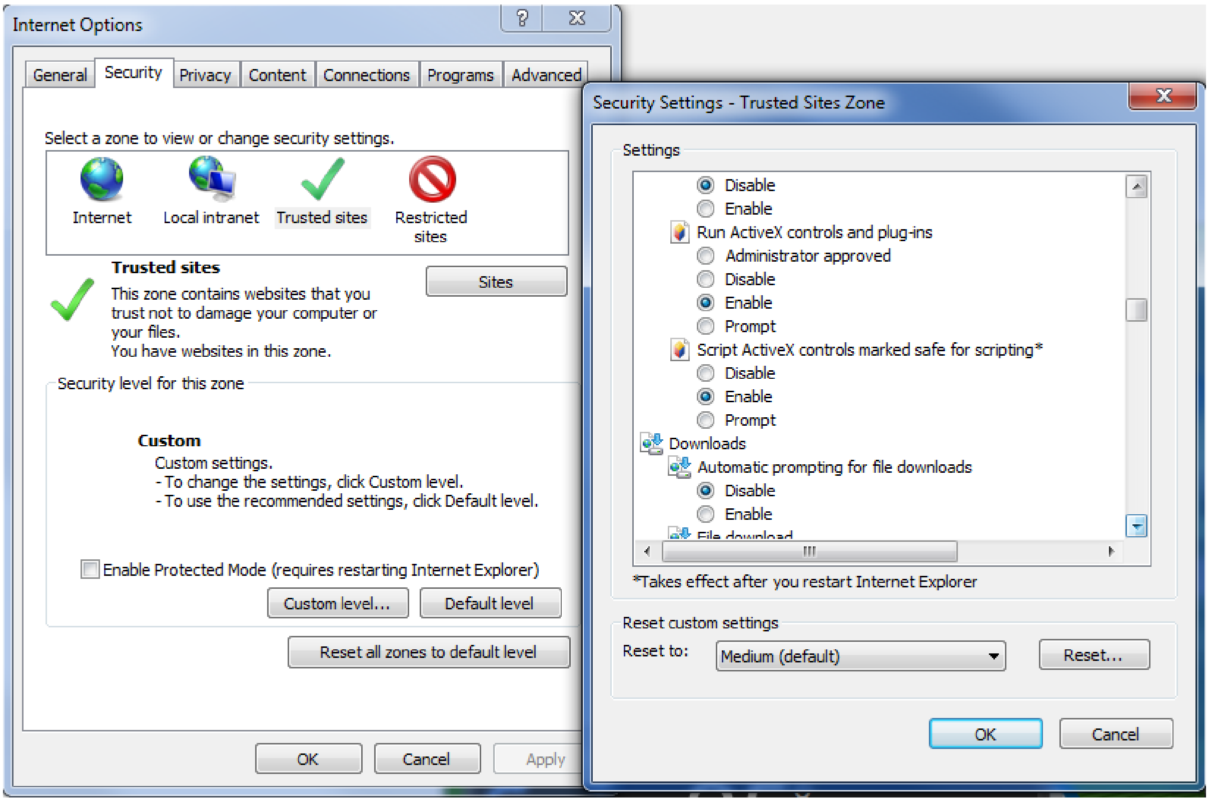
Download signed ActiveX controls
○ Disable
● Enable
○ Prompt
Download unsigned ActiveX controls
○ Disable
● Enable
○ Prompt
Initialize and script ActiveX controls not marked as safe for s
○ Disable
● Enable
○ Prompt
Only allow approved domains to use ActiveX without prompt
● Disable
○ Enable
Run ActiveX controls and plug-ins

*Takes effect after you restart Internet Explorer

Reset custom settings

Reset to: Medium (default)

Figure 33



Internet Options

General | Security | Privacy | Content | Connections | Programs | Advanced

Select a zone to view or change security settings.

Internet | Local intranet | Trusted sites | Restricted sites

Trusted sites

This zone contains websites that you trust not to damage your computer or your files.
You have websites in this zone.

Sites

Security level for this zone

Custom

Custom settings.
- To change the settings, click Custom level.
- To use the recommended settings, click Default level.

□ Enable Protected Mode (requires restarting Internet Explorer)

Custom level... | Default level

Reset all zones to default level

OK | Cancel | Apply

Security Settings - Trusted Sites Zone

Settings

- ● Disable
- ○ Enable
- Run ActiveX controls and plug-ins
  - ○ Administrator approved
  - ○ Disable
  - ● Enable
  - ○ Prompt
- Script ActiveX controls marked safe for scripting*
  - ○ Disable
  - ● Enable
  - ○ Prompt
- Downloads
  - Automatic prompting for file downloads
    - ● Disable
    - ○ Enable
  - File download

*Takes effect after you restart Internet Explorer

Reset custom settings

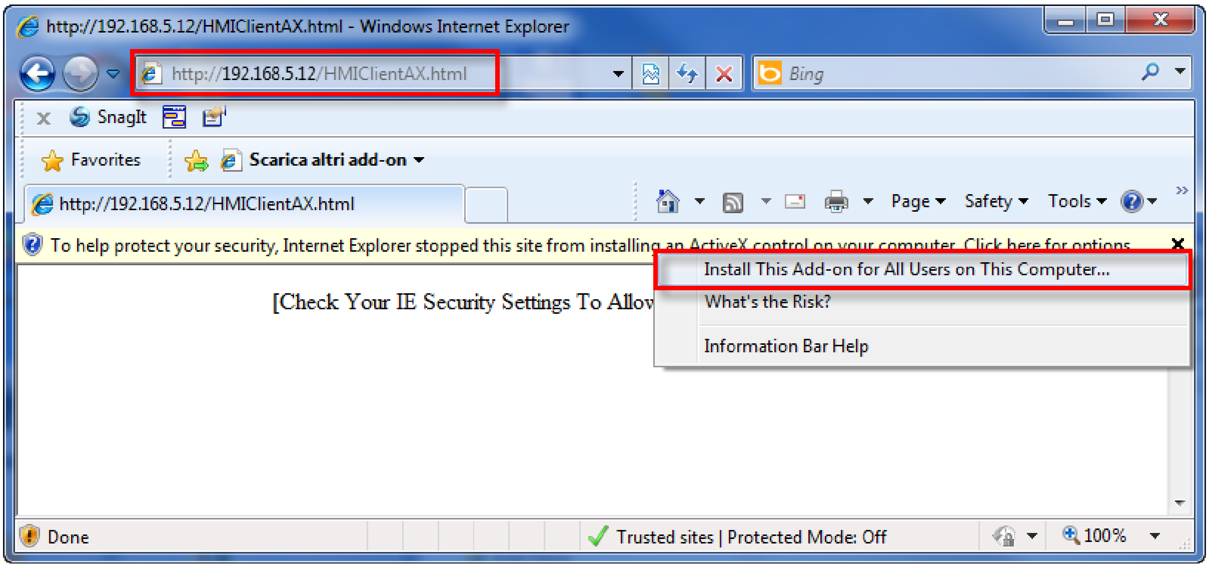
Reset to: Medium (default) | Reset...

OK | Cancel

Figure 34

4.11.1.4 Install Active X on Internet Explorer

In Internet Explorer, allow the installation of the ActiveX component when the question pops up in your browser.



http://192.168.5.12/HMIClientAX.html - Windows Internet Explorer

http://192.168.5.12/HMIClientAX.html

To help protect your security, Internet Explorer stopped this site from installing an ActiveX control on your computer. Click here for options...

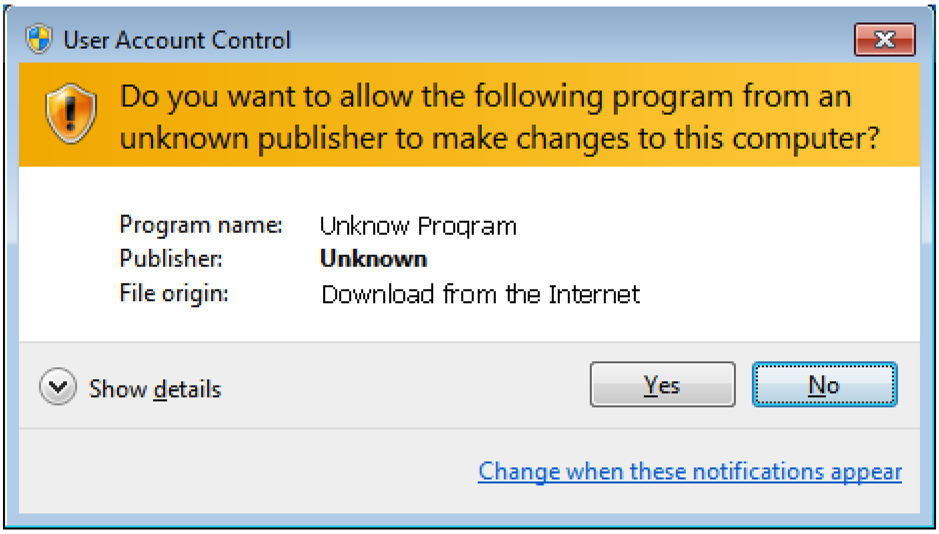
- Install This Add-on for All Users on This Computer...
- What's the Risk?
- Information Bar Help

[Check Your IE Security Settings To Allov

Done | Trusted sites | Protected Mode: Off | 100%

Figure 35

In case of you are using a Vista or Windows 7 operating system, you need to click on Yes on User Account Control, as shown in the following picture.



User Account Control

Do you want to allow the following program from an unknown publisher to make changes to this computer?

Program name: Unknow Program
Publisher: **Unknown**
File origin: Download from the Internet

Figure 36

4.11.1.5 Uninstalling Active X

To remove the ActiveX component from your system, you must delete it from the computer. By default, the component is installed in the following folder:
C:\Program Files\Exor\HMIClientAX

4.11.1.6 ActiveX information

The ActiveX is able to show projects at a maximum pixel resolution of 1200 x 800.

5 Basic Programming Concepts in JMobile

The programming guidelines for JMobile are based on a few basic concepts, which are recurrent in many parts of the system

5.1 Attach to

In JMobile the basic programming technique consist in configuring the properties for an object in page.
Objects' properties can be changed at programming time or configured to be dynamic.

To change a property you can use the page toolbar or the property pane which shows the properties available for the selected object.

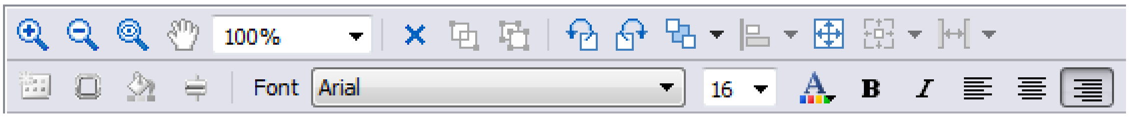
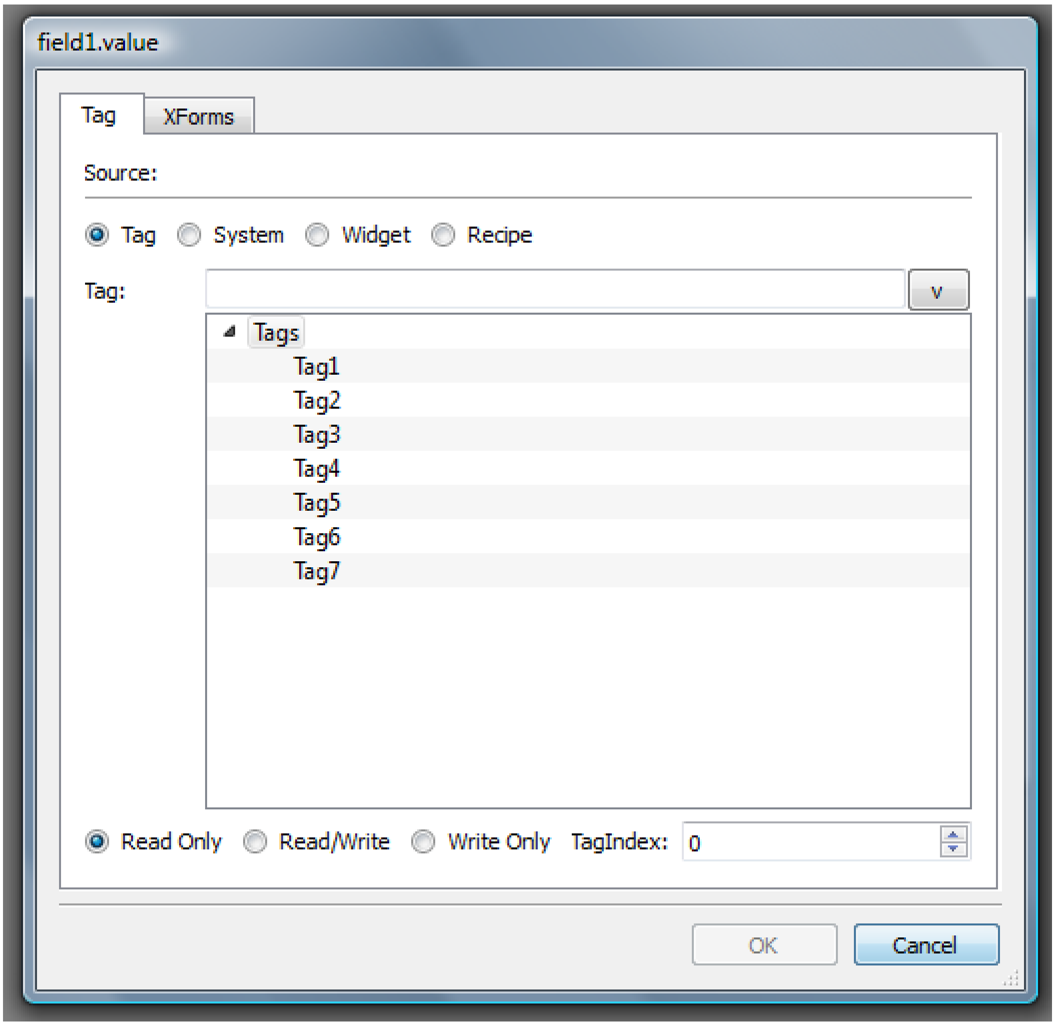


Figure 37

The page toolbar permits a quick change at programming time of the most commonly used object's properties.
When you need a complete view of all the properties of a certain object you need to use the property pane.
The property pane allows both to change a property at programming time and to attach the property to a dynamic element.
From the Property Pane, when you click on the right side of a property cell, you get the possibility to "Attach" the property to a second element. This operation is done using the "Attach to" dialog shown in Figure 38.

The Attach to dialog has two tabs. The first is called "Tag" and allows to attach the property to an element. The "source" can be selected using the radio buttons.
The element to which the property can be attached is:
- A Tag
- A System Variable (see chapter System Variables for an explanation of the meanings of all System Variables)
- A property from another Widget
- An element of a Recipe

The radio buttons at the bottom allow to set the access type.
The TagIndex selection is used in case of arrays to determine the array element.



field1.value

Source:

● Tag ○ System ○ Widget ○ Recipe

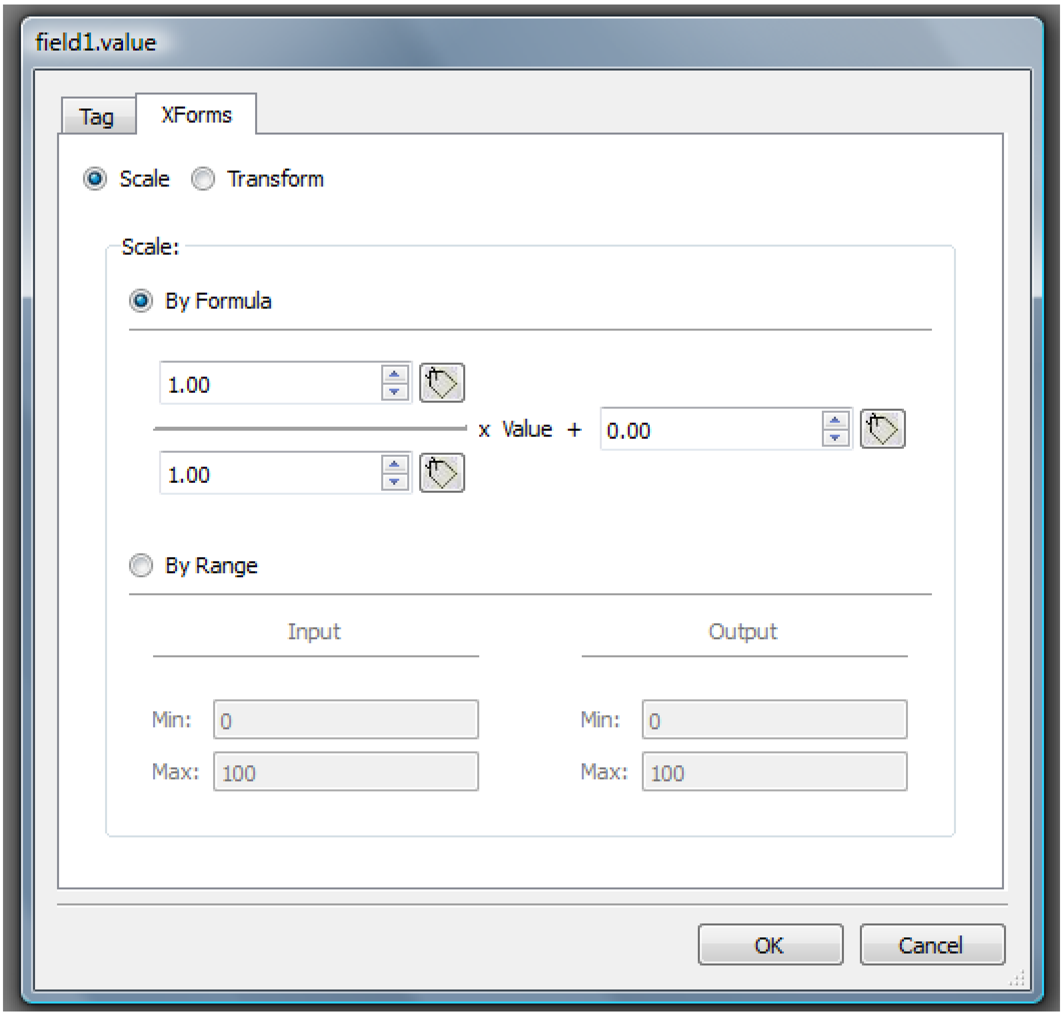
Tag:

Tags
Tag1
Tag2
Tag3
Tag4
Tag5
Tag6
Tag7

● Read Only ○ Read/Write ○ Write Only TagIndex: 0

Figure 38

The second tab is called "XForms" and it is shown in Figure 39.
The XForms tab allows to apply transformations to the numeric value of the source element before is applied to the property.
Transformations can be simple linear relationships or generic transformations.
Linear scaling can be configured when selecting the "Scale" radio button and they can be specified in terms of a formula or "By range". In case the range mode is selected, you just need to specify the input and output range while the system will automatically calculate the factors for the formula.



field1.value

Tag | XForms

● Scale ○ Transform

Scale:

● By Formula

1.00 / 1.00 x Value + 0.00

○ By Range

Input: Min: 0, Max: 100

Output: Min: 0, Max: 100

Figure 39

Special transformations are available when you click on the "Transform" radio button. Currently supported transformations are: color conversion and bit/byte index.

Color conversion allows to define a map between numeric values of the tag and colors to be assigned to the property.

Bit or Byte index transformation allows extracting from a word the single bit or byte contents depending on the specified bit or byte number.

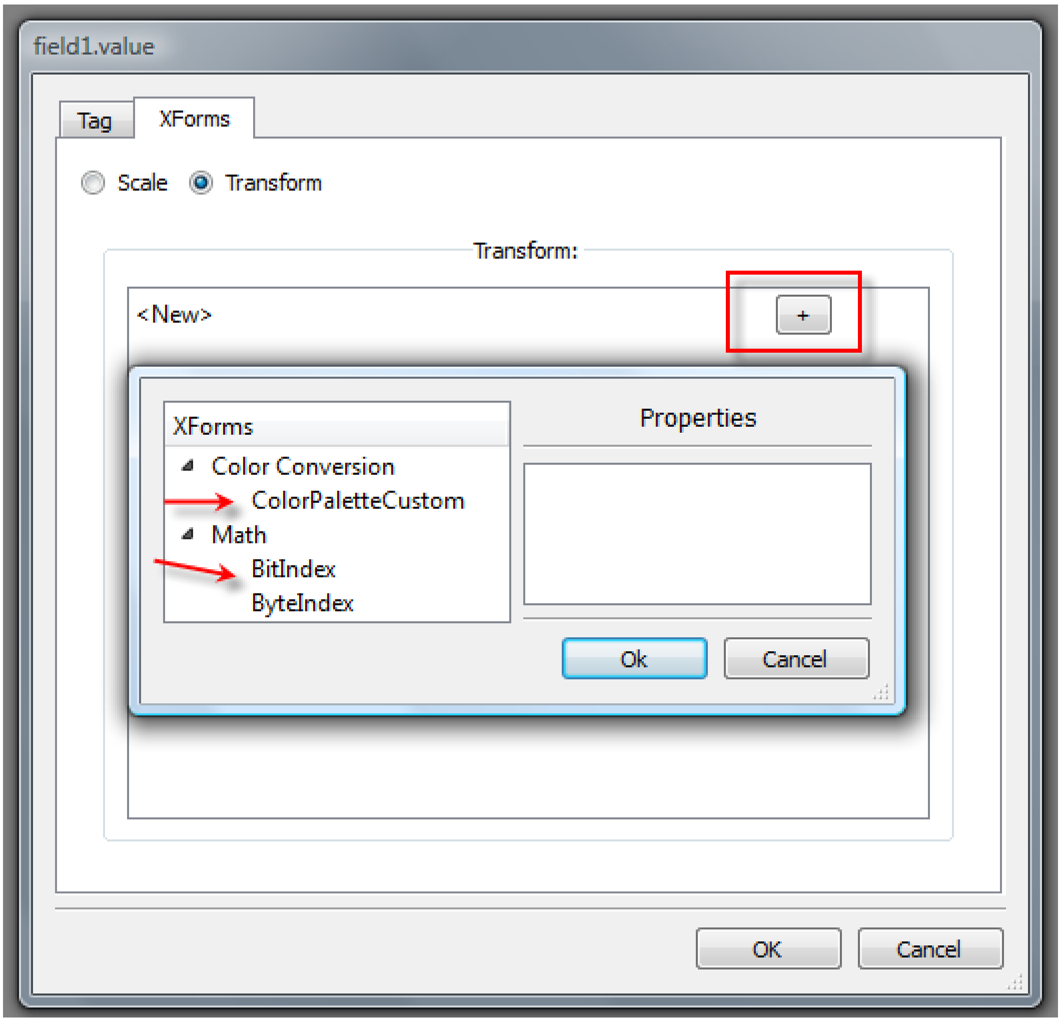


Figure 40

5.2 Events

In a JMobile system, an Event is generated under the following conditions:

When a button is pressed.
When a button is released.

When the visualization part of a Widget changes—this also includes the case of numeric fields with a Tag attached; the visualization will be updated because the linked Tag has changed its value.

When a page is entered.
When a page is left.
When the visualization component of at least one object in a certain page changes.
When an alarm is triggered.
When the scheduler engine is triggered because of a time condition.

Whenever the system generates an event, it is also possible to attach on of the following executed actions to the event:
A specific action coming from a list of predefined actions
A piece of JavaScript code
Figure 41 shows an example of an action activated by pressing a button.

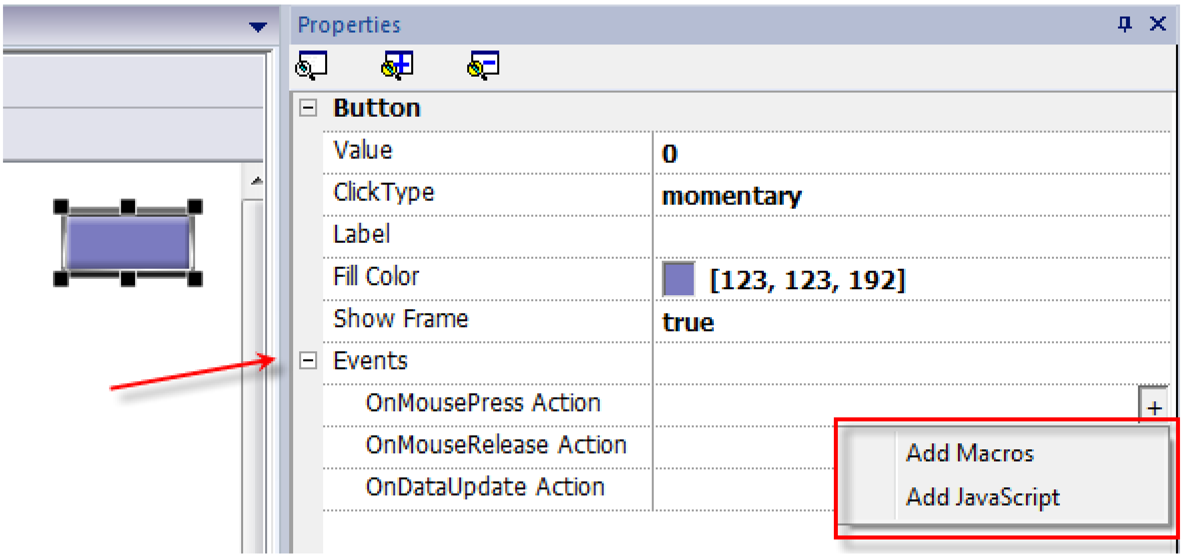
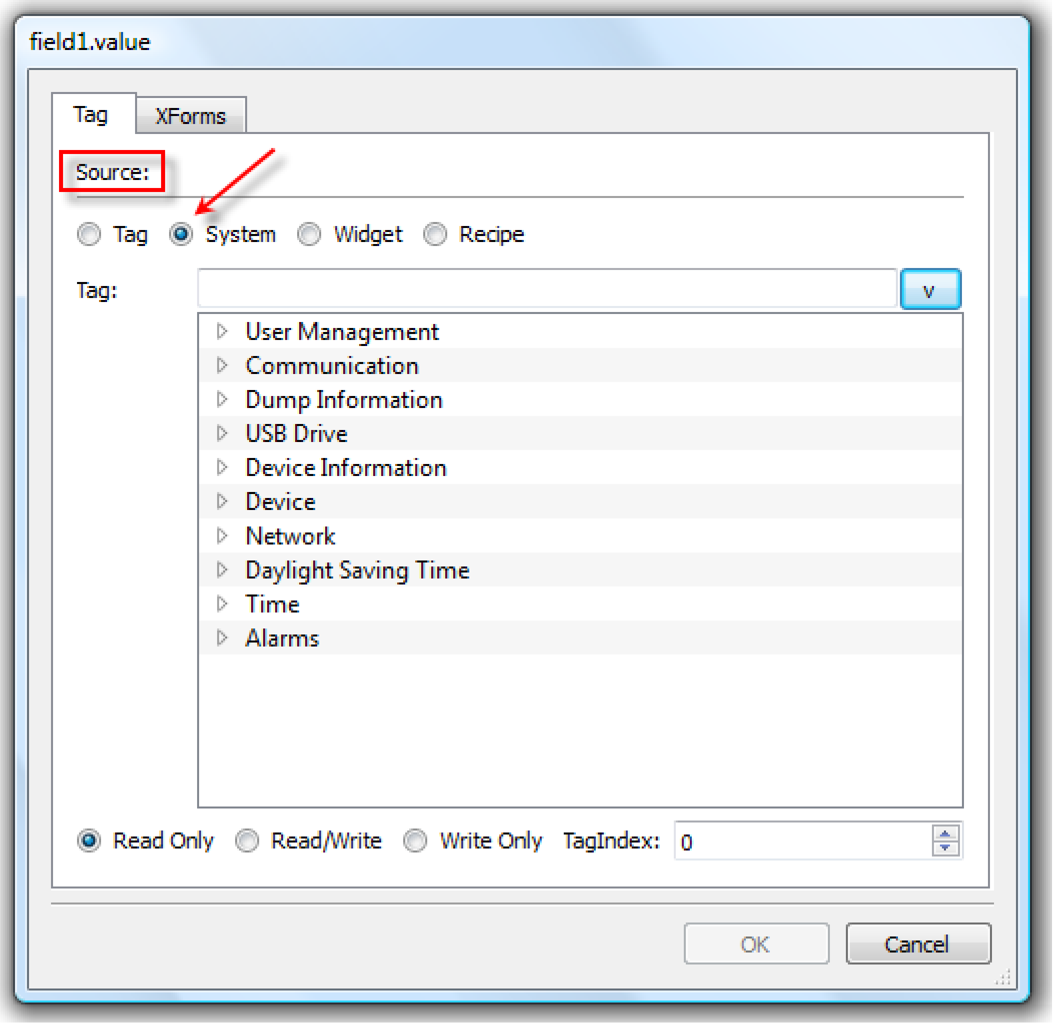


Figure 41

By associating actions to events, the JMobile programmer carries out program interactions with the interface.

6 System Variables

System variables are special system tags containing generic informations about the runtime and its operation.
System variables are available in the Attach to dialog from the "Source selection" as shown in the following picture.



field1.value

Source: ○ Tag ● System ○ Widget ○ Recipe

● Read Only ○ Read/Write ○ Write Only TagIndex: 0

Figure 42

System variables are divided in categories.

6.1 User Management

These system variables return information about users and groups.

This Client ID
The variable is valid with reference to the Client scope. Local and remote clients connected to the same "server" (same runtime) get a unique ID returned by this variable.

This Client User-Name

Name of the user logged to the client where the system variable is displayed

This Client Group-Name
Name of the group to which the current logged user belongs to.

No Of Remote-Clients Alive
Number of remote clients connected to the server.

6.2 Communication

These variables return information about the status of the communication between the target device and the controllers configured in the Protocol Editor.

Protocol Communication Status
The variable can assume 3 values:

0: No protocol running; it may happen if the protocol driver has not be properly downloaded to the target system
1: Protocol has been properly loaded and started; no communication errors
2: At least one communication protocol is reporting an error

Protocol Error Message
This variable returns an ASCII string containing a description of the actual communication error, the communication protocol acronym is reported between square brakets to recognize the source of the error in case of multiple protocols configuration.

Protocol Error Count
This variable returns the number of communication errors occurred.

6.3 Dump information

These variables return information about the status of the copy process to external drives (USB) for trend and archive buffers

Dump Trend Status
Returns value 1 during the copy process of the trend buffers. If the copy duration time is less than one second, the system variable does not change its value.

Dump Archive Status
Returns value 1 during the copy process of the archive buffers. If the copy duration time is less than one second, the system variable does not change its value.

6.4 USB Drive

The variables in this category return information about the external USB drive connected to the panel, the available variables are: USB Drive Name, USB Drive Size and USB Drive FreeSpace.

6.5 Device Information

The variables in this category return generic information about the hardware target device. Several informations are available in this category.

6.6 Device

The variables in this category can be used to adjust specific device settings and get operational information.

Display Brightness
This variable is of R/W type. Its range goes from 0 to 255. It can be used to check brightness level and adjust it from the application. Typical use is when connected to a slider widget.

Battery Timeout
Not used. Reserved for future enhancements.

External Timeout
Allows setting the not operational time after which the display backlight is automatically turned off. The backlight is then automatically turned on when user presses on the touchscreen.

Touch Buzzer
Allows to enable/disable the touch audible feedback.

Battery LED
Allows to enable/disable the use of the front LED indicator to report the low level battery status.

System Mode
Returns a value informing on the operation status of the runtime, the possible values are:

1: Operating mode
2: Configuration mode
3: Restart

System Font List
List of system fonts.

Flash Free Space
Returns the free space left in the device internal flash.

6.7 Network

The system variables in this category allow to show and set network device parameters. Except for the MAC ID, they are all of R/W type.

6.8 Daylight Saving Time

The variables in this category return information about the system clock and allow adjust it from the application.
They contain information of the "local" time.

Standard time is the "solar time" and other is Day light saving time.

Standard offset
This represents the offset in minutes when standard time is set, with respect to GMT. (with respect to the picture it is -8*60 = - 480 minutes)

Standard week
This is the week in which the Standard time starts (w.r.t. the picture it is First = 1).

Standard Month

This is the month in which the standard time starts (range of the variable is [0 -11] so w.r.t. the picture it is November = 10)

Standard Day
This is the day of week in which the standard time starts (w.r.t. the picture it is Sunday = 0)

Standard hour
This is the hour in which the standard time starts (w.r.t. the picture into Time field it is 02 = 2)

Standard minute
The minute in which the standard time starts (w.r.t. the picture into Time field it is 00 = 0)

Dst offset
This represents the offset in minutes when Dls time is set, with respect to GMT. (w.r.t. the picture it is -7*60 = - 420 minutes)

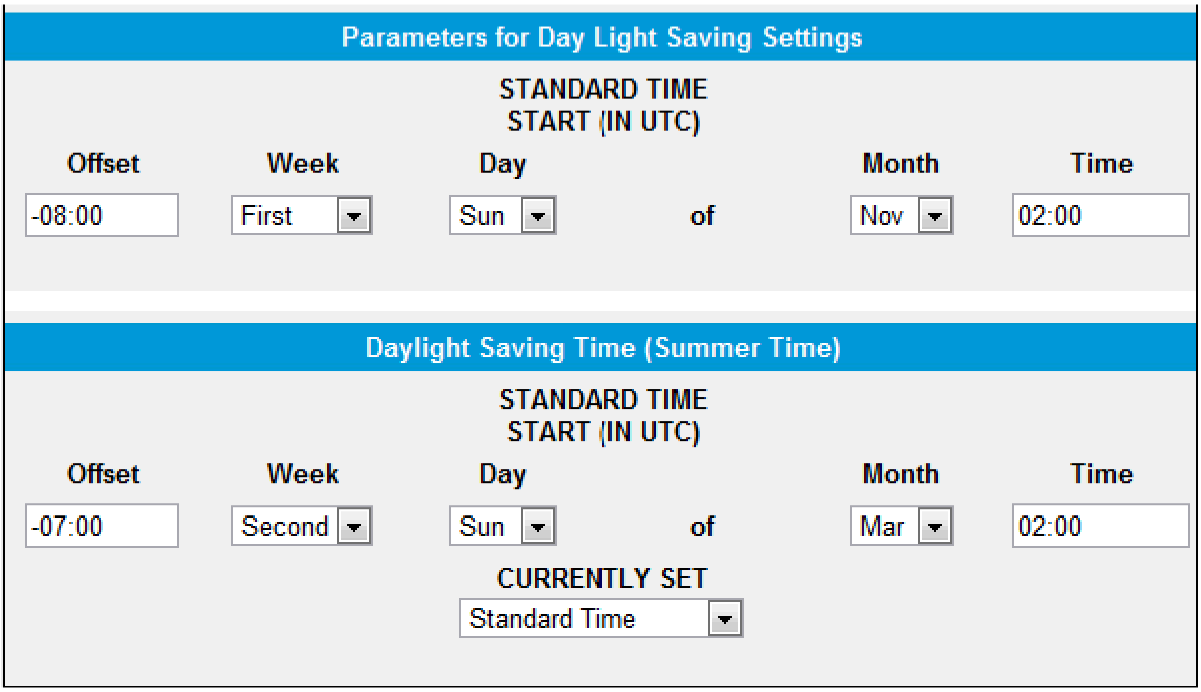
Dst week
This is the week in which the Dls time starts (w.r.t. the picture it is Second = 2).

Dst Month
This is the month in which the Dls time starts (range of the variable is [0 -11] so w.r.t. the picture it is March = 2)

Dst Day
This is the day of week in which the Dls time starts (w.r.t. the picture it is Sunday = 0)

Dst hour
This is the hour in which the Dls time starts (w.r.t. the picture into Time field it is 02 = 2)

Dst minute
This is the minute in which the Dls time starts (w.r.t. the picture into Time field it is 00 = 0)



Parameters for Day Light Saving Settings

STANDARD TIME START (IN UTC)

Offset	Week	Day		Month	Time
-08:00	First	Sun	of	Nov	02:00

Daylight Saving Time (Summer Time)

STANDARD TIME START (IN UTC)

Offset	Week	Day		Month	Time
-07:00	Second	Sun	of	Mar	02:00

CURRENTLY SET: Standard Time

Figure 43

6.9 Time

The variables in this category return information about the System Time expressed in UTC format.

6.10 Alarms

The variables in this category return information about the actual number of alarms divided per status.

7 Working with Actions

7.1 Widget Actions

7.1.1 Show Widget

The Show Widget macro allows you to Show or Hide the page Widgets. In the Macro properties, select the Widget you want to show or hide, then set the Show properties as follows: False to Hide and True to Show Widget.

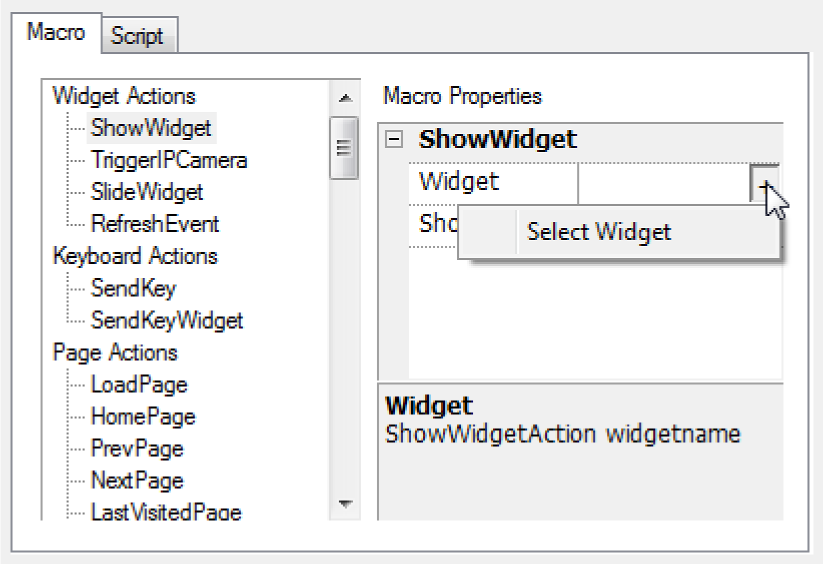
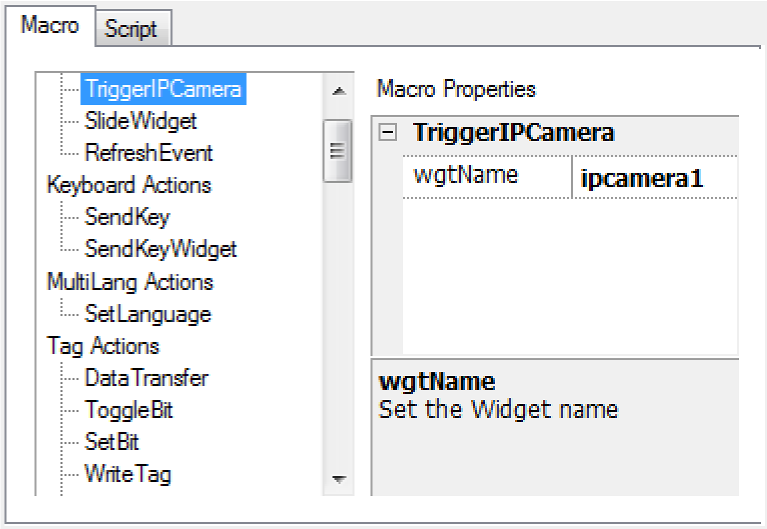


Figure 44

7.1.2 Trigger IP Camera

The Trigger IP Camera macro allows you to prompt the IP Camera to capture. Select the IP Camera from the Macro properties to trigger a capture from the IP Camera.



Macro Properties

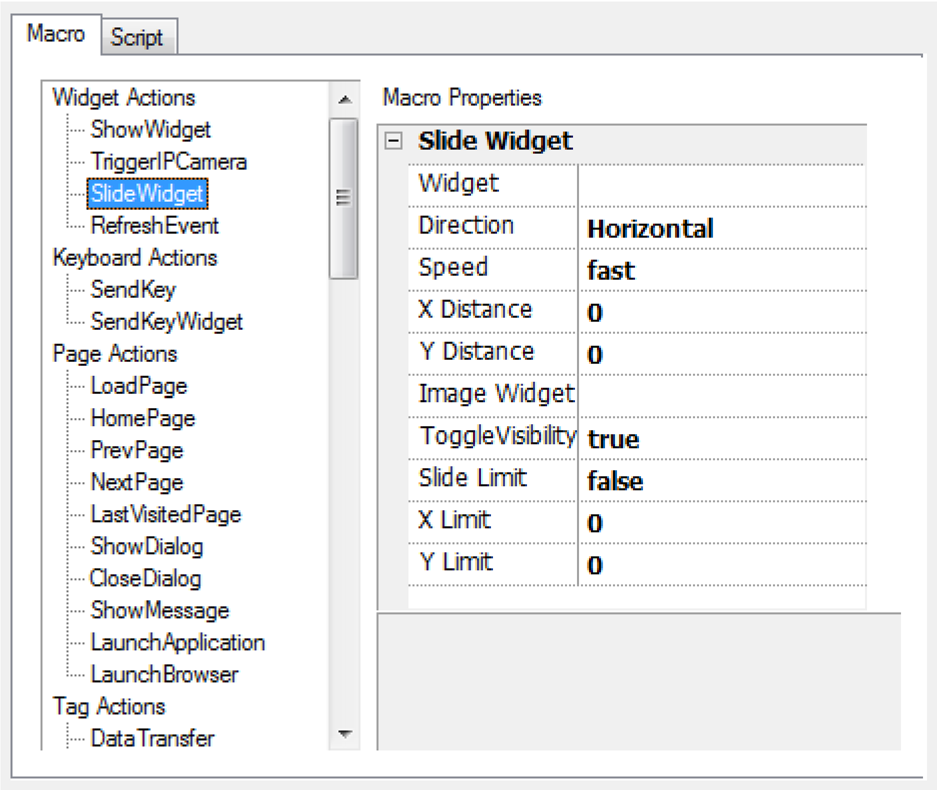
TriggerIPCamera	
wgtName	ipcamera1

wgtName
Set the Widget name

Figure 45

7.1.3 Slide Widget

The Slide Widget macro allows you to show the sliding effect of a Widget, or of a Widget group, in HMI Runtime.



Macro Properties

Slide Widget	
Widget	
Direction	Horizontal
Speed	fast
X Distance	0
Y Distance	0
Image Widget	
ToggleVisibility	true
Slide Limit	false
X Limit	0
Y Limit	0

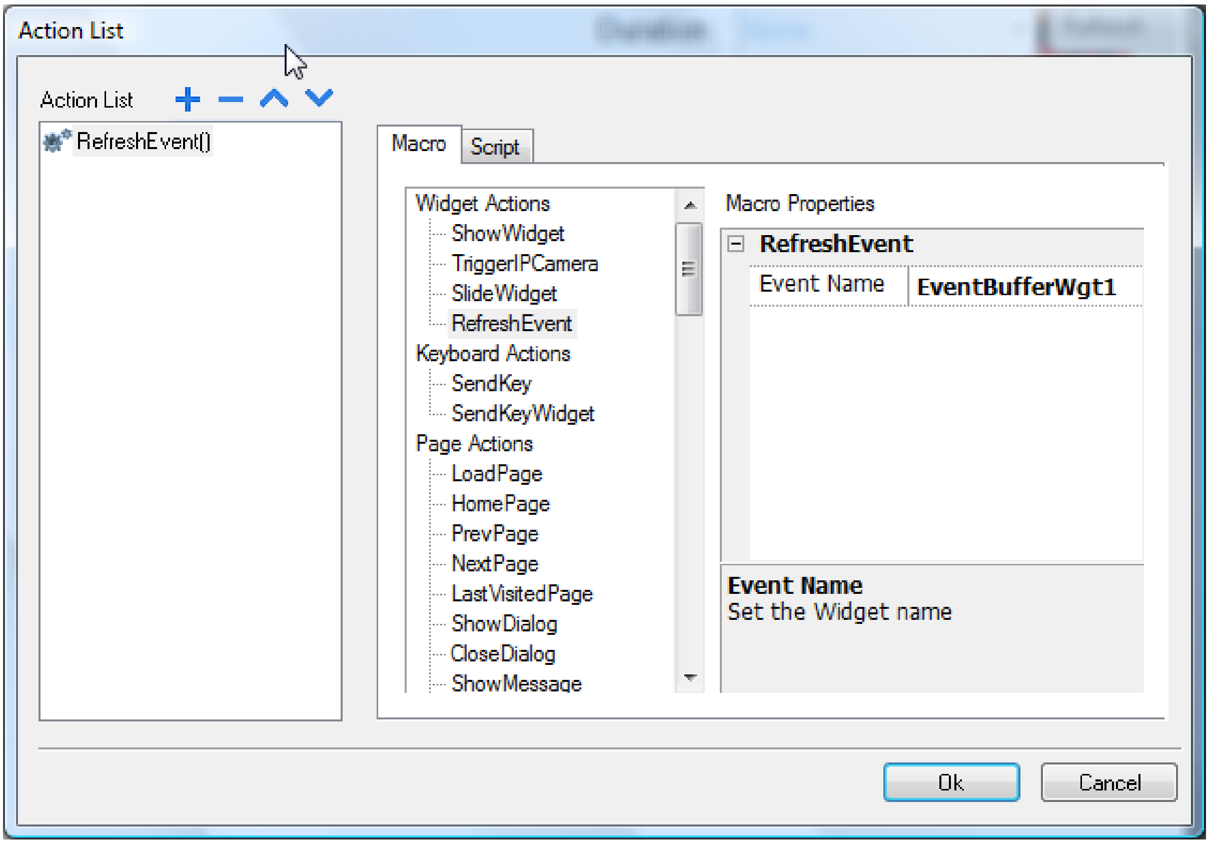
Figure 46

Widget	The Widget to slide
Direction	Sliding Direction
Speed	The transition speed of the sliding Widget
X Distance	The travel distance of the X coordinate of Pixel

Y Distance The travel distance of the Y coordinate of Pixel Image Widget

7.1.4 Refresh Event

The Refresh Event macro allows you to refresh the selected Events Widget.



RefreshEvent

Event Name	EventBufferWgt1

Event Name
Set the Widget name

Figure 47

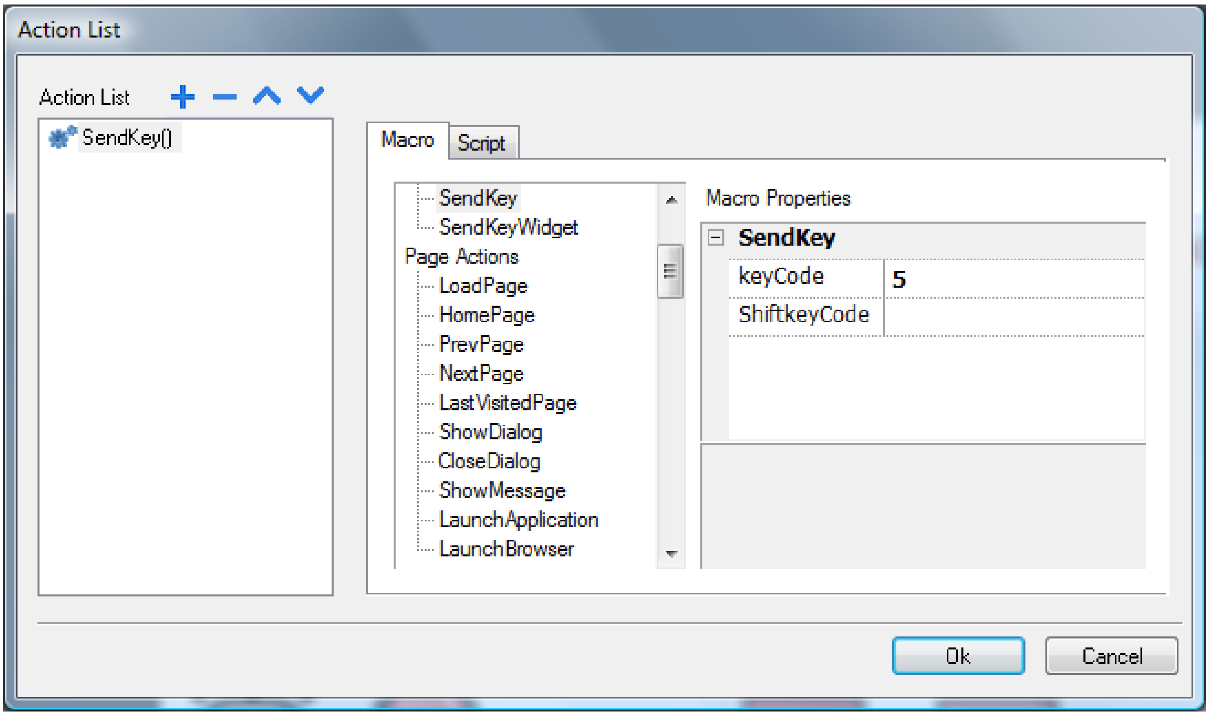
7.2 Keyboard Macro Actions

The Keyboard macro actions include Send Key and Send Key Widget.

7.2.1 Send Key

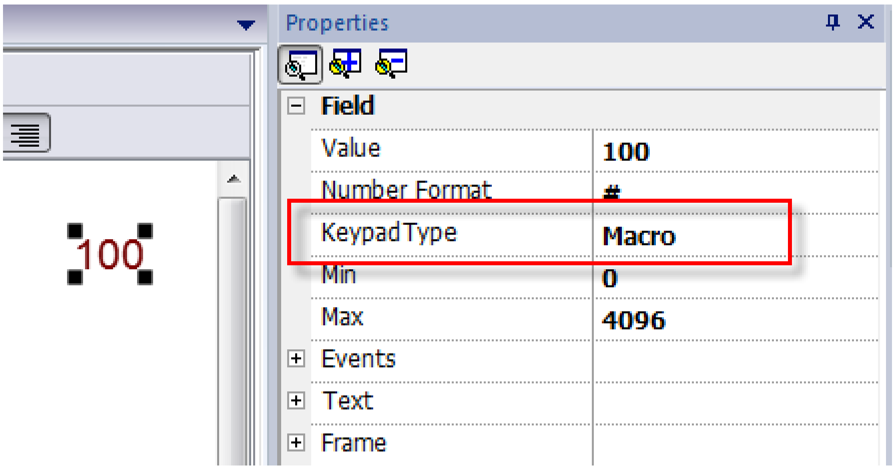
The Send Key macro is used to enter the predefined character to the Read/Write Widget. Define the predefined key code and Shift key code to the Macro actions property. In Runtime, first click the R/W numeric Widget, then execute the Macro to send the predefined keys to the Numeric Widget. The action works on the field currently being edited.

Note: To use the Send Key macro, you must define the keypad type as “Macro" in the Numeric Widget properties (as shown in Figure 52).



SendKey	
keyCode	5
ShiftkeyCode	

Figure 48



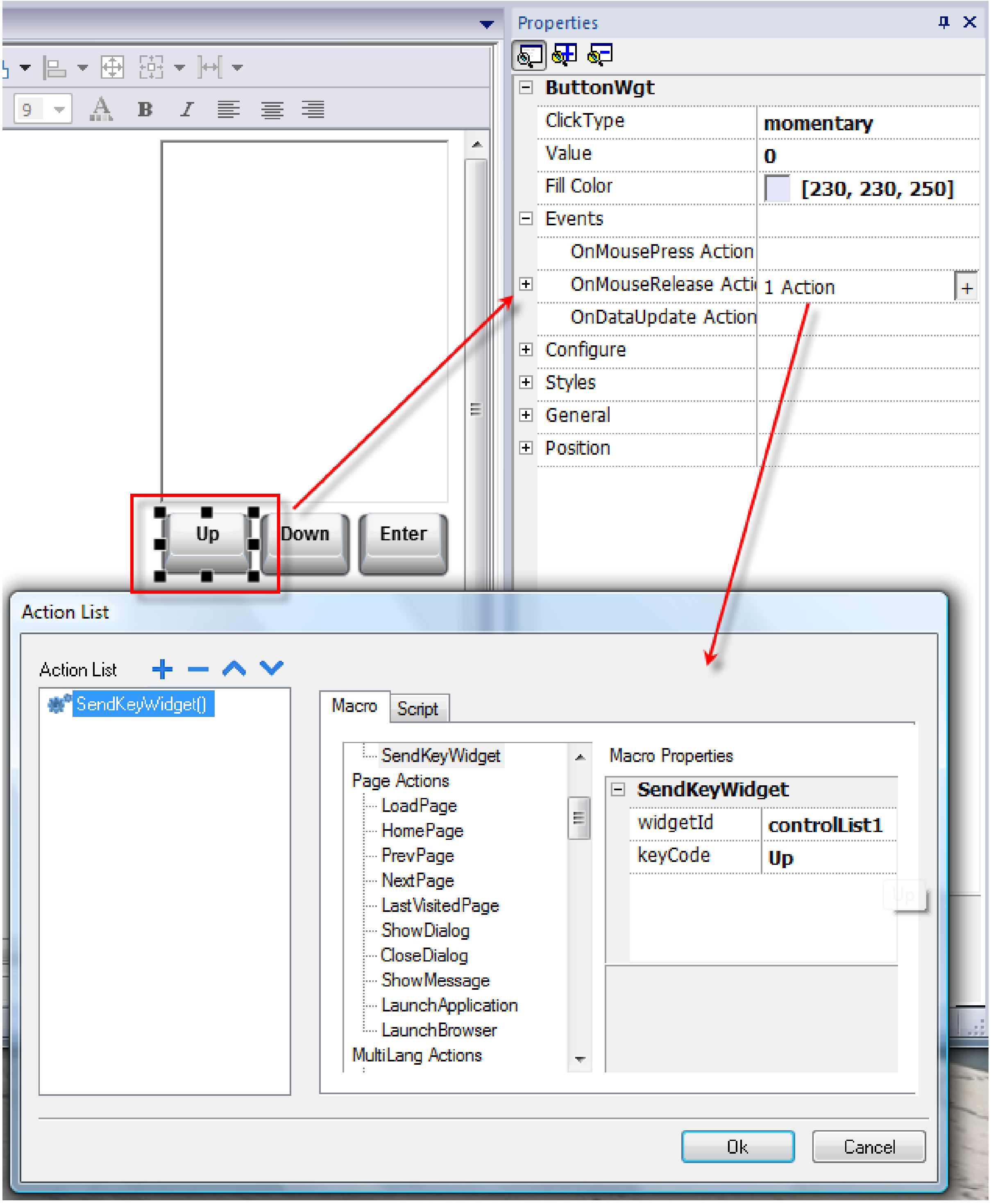
Field	
Value	100
Number Format	#
KeypadType	Macro
Min	0
Max	4096
Events	
Text	
Frame	

Figure 49

7.2.2 Send Key Widget

The Send Key Widget macro is used to enter the predefined character for a specific Widget. To use the macro, define the Widget ID and the key code into the Macro Properties.

Control List Widget (available in the Advanced category of the Widgets Gallery) is a good example of how this macro command can be used. Here Up and Down buttons have been implemented using the Send Key Widget macro. See Figure 53 for reference.



ButtonWgt

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

Action List: SendKeyWidget()

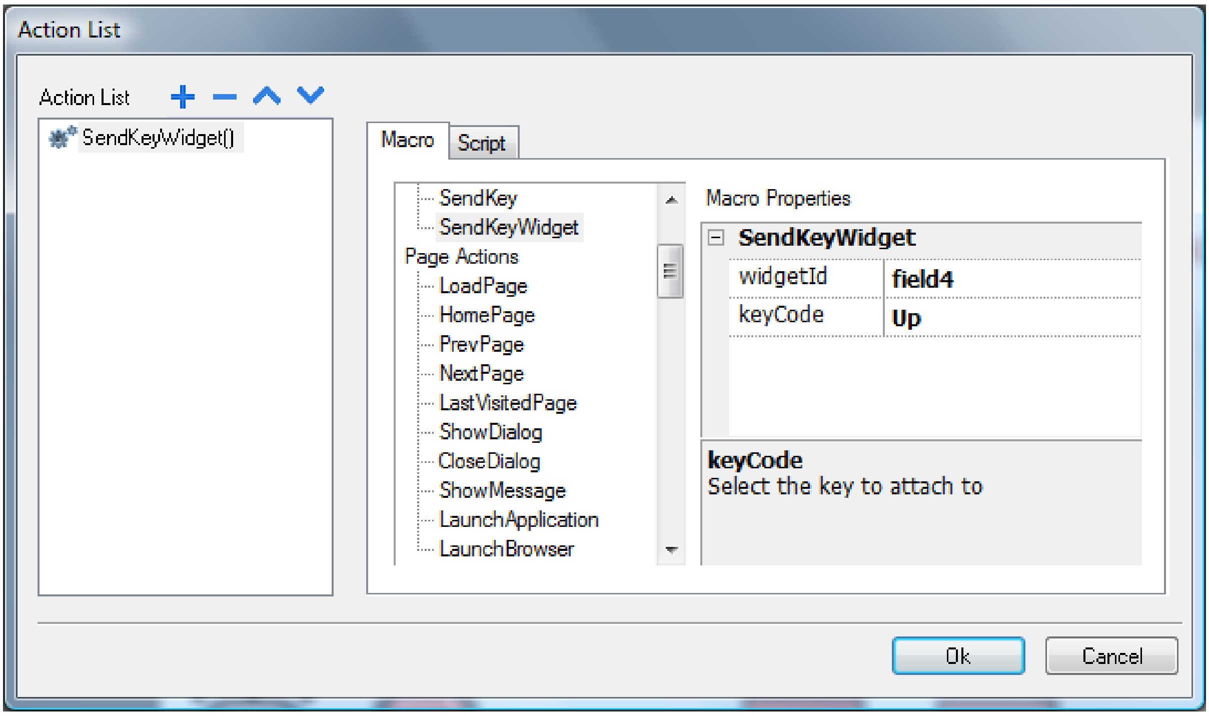
Macro Properties

SendKeyWidget

widgetId	controlList1
keyCode	Up

Figure 50

Note: To use the Send Key Macro, you need to define the keypad type as “Macro" in the Numeric Widget properties.



SendKeyWidget

widgetId	field4
keyCode	Up

keyCode
Select the key to attach to

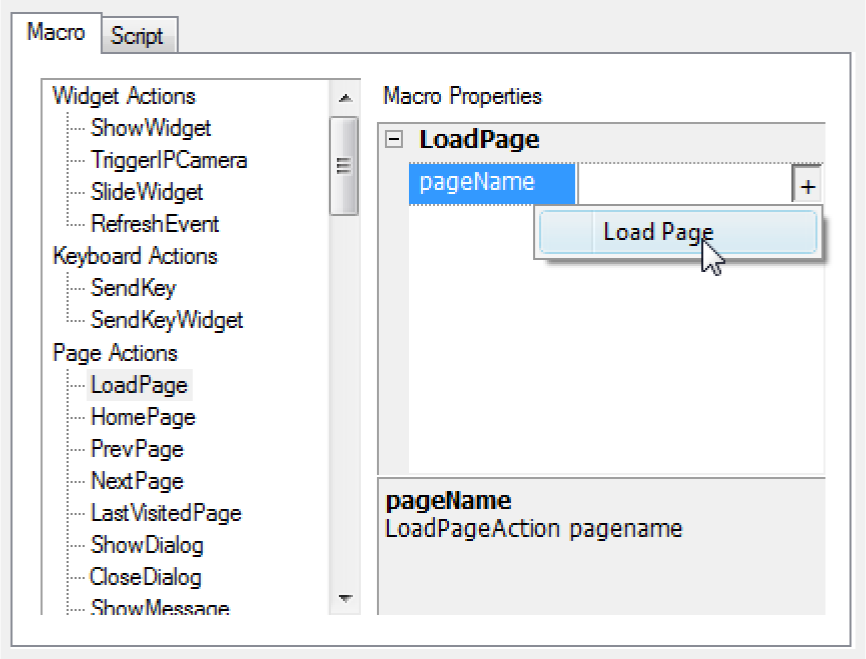
Figure 51

7.3 Page Actions

The Page Actions macro is used for Page Navigation and Load-specific pages. Please note that the Page Action macro will not be available for Mouse Press actions. The Page Actions macro is available for Alarms, Schedulers and Mouse Release Events.

7.3.1 Load Page

The Load page macro allows you to load the selected page when the macro is executed.

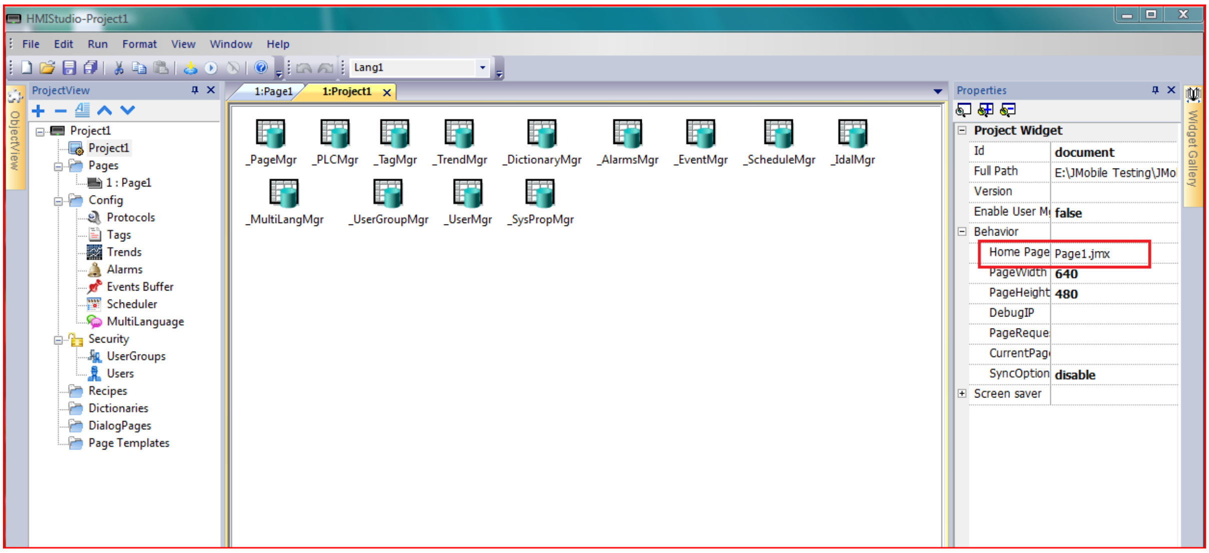


pageName
LoadPageAction pagename

Figure 52

7.3.2 Home Page

The Home Page property allows you to specify the home page. By default, the Home Page is the first page. However, you can change the Home Page in the project configuration Properties. To change the Home Page, double click on the Project name item in Project View. Once in Properties, choose the Home Page (as shown in Figure 56).



Project Widget

Id	document
Full Path	E:\JMobile Testing\JMo
Version	
Enable User M	false
Behavior	
Home Page	Page1.jmx
PageWidth	640
PageHeight	480
DebugIP	
PageReque	
CurrentPag	
SyncOption	disable
Screen saver	

Figure 53

7.3.3 Previous Page

The Previous Page macro allows you to navigate HMI Runtime to the previous page.

7.3.4 Next Page

The Next Page macro allows you to navigate HMI Runtime to the next page.

7.3.5 Last Visited Page

The Last Visited page macro allows you to load the page previously displayed on HMI Runtime.

7.3.6 Show Dialog Page

Show Dialog Page allows you display the Dialog Pages defined in the project. After the execution of this macro, HMI Runtime displays the specified Dialog Page.

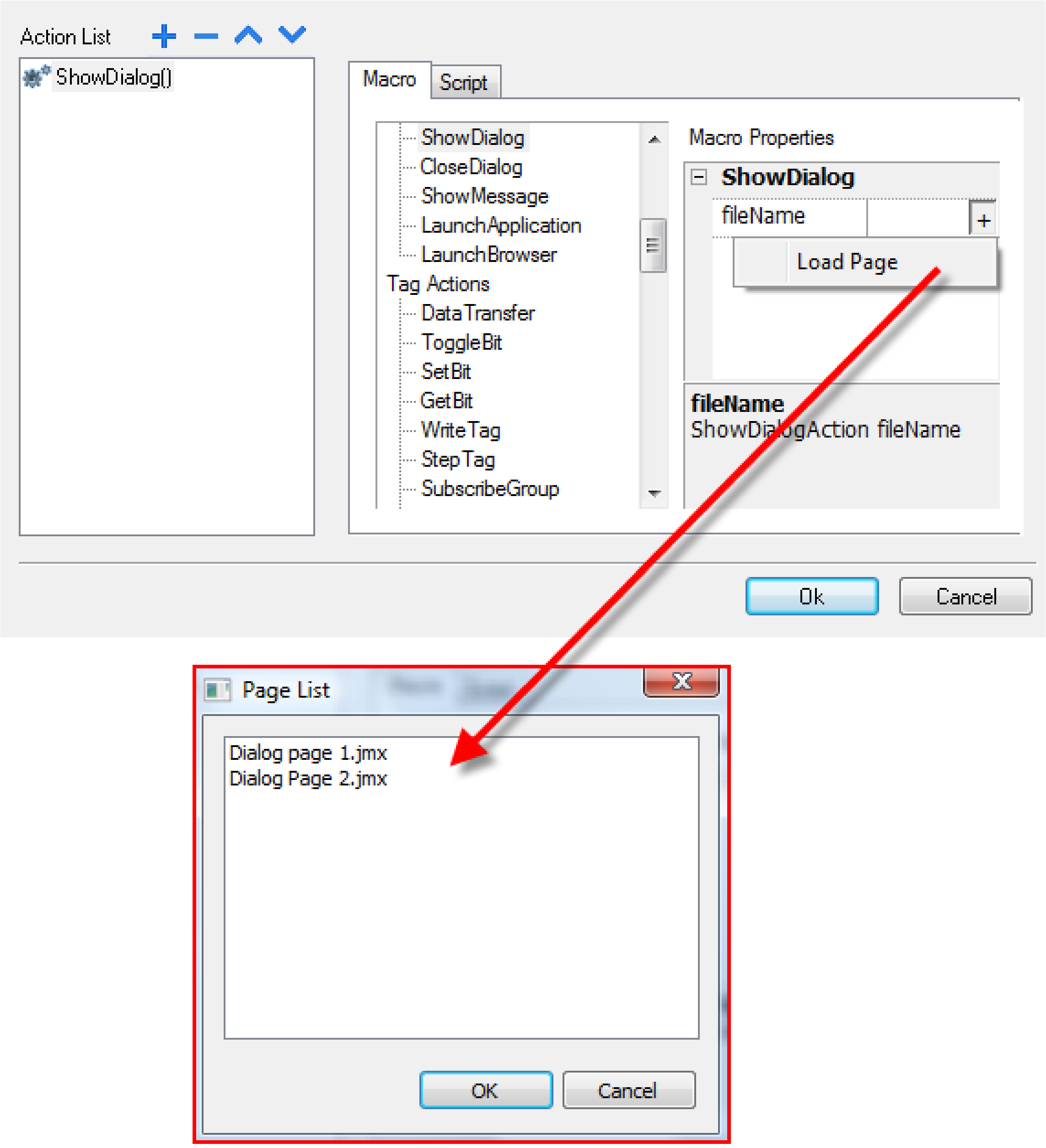


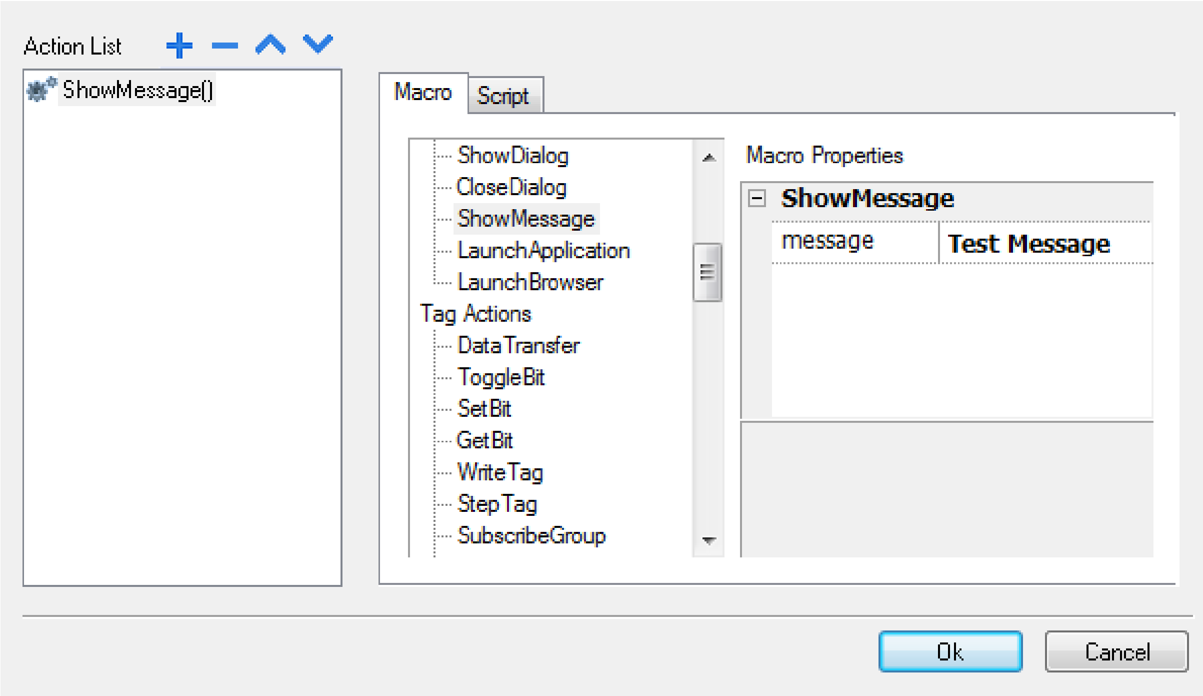
Figure 54

7.3.7 Close Dialog

The Close Dialog page is applicable only on Dialog pages. The Close Dialog page allows you to close the dialog page currently displayed.

7.3.8 Show Message

The Show Message macro allows you to display warning message popups when the macro is executed. Type the message that you wish to have displayed while executing the macro (as shown in Figure 55).



message	Test Message

Figure 55

7.3.9 Launch Application

The Launch Application, in the Macro tab, allows the user to launch an external third party application when the macro is executed. To configure the external launch application, the following inputs must be provided in order to execute the specific application.

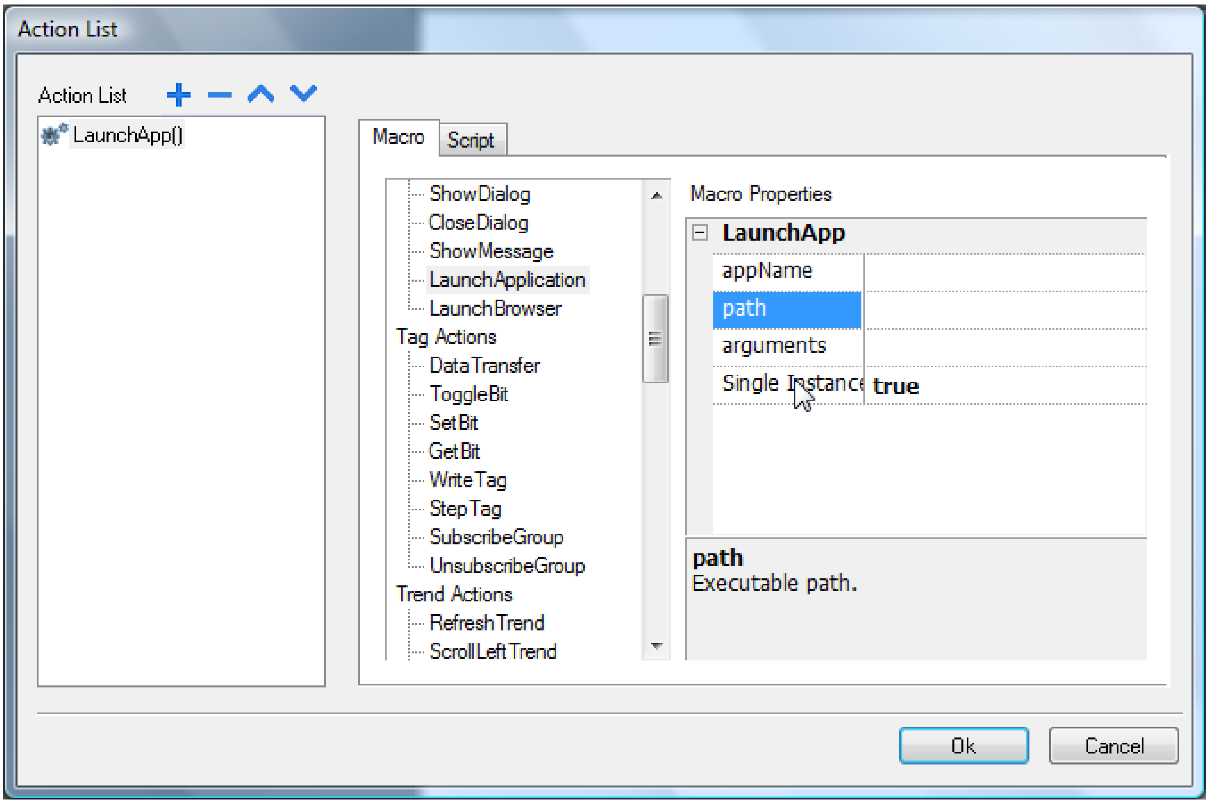
App Name The executable file name with extension. For example, if you want to run notepad application, the argument should be “notepad.exe"

Path The application path; when the target platform is Windows CE, the path has to be specified without the volume information; if for instance we need to launch an application from the Windows folder, the path will be indicated as "\windows"

Arguments Some external applications need arguments to be passed. For example, to open a notepad file, specify the file name so that, while launching the application, the file name set in the argument is loaded on the application.

Single This argument allows the application to start in single instances or

Instance multiple instances. When single instance is selected the system first verifies whether the application is already running. If it is running, then the application gets the focus (the operating system puts it in foreground to user attention); if it is not running, then the application is launched.



path
Executable path.

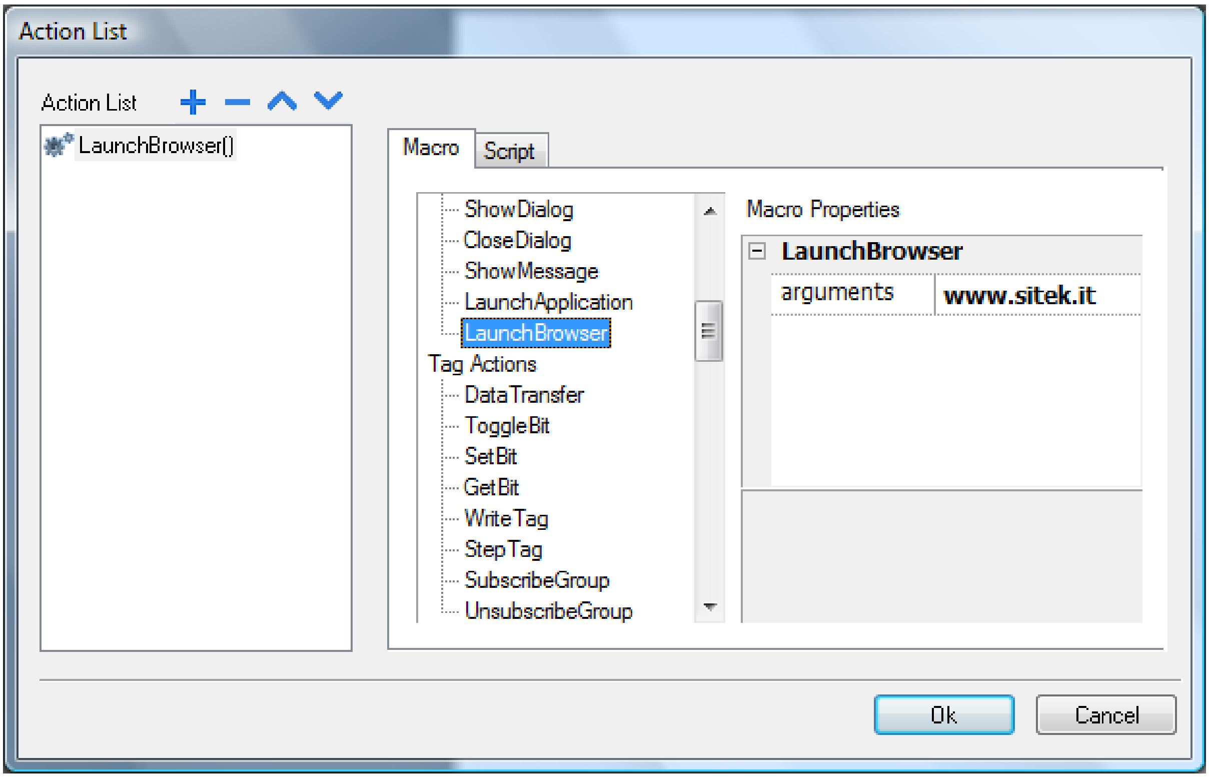
Figure 56

7.3.10 Launch Browser

Launch Browser will launch the default web browser. You can define the URL address of the webpage in the arguments.

Note: Not all the UniOP platforms are equipped with a default web browser. Contact Technical Support for additional information.

I



LaunchBrowser	
arguments	www.sitek.it

Figure 57

7.4 Tag Actions

The Tag Actions macros are used to interact with the application's Tags.

7.4.1 Data Transfer

Data Transfer macros allow you to exchange data between two controllers, between registers within a controller, or from system variables to controllers (and vice versa). " SrcTag" refers to the source Tag and "DestTag" refers to Destination Tag. The various Tag types include a Controller Tag, a System Tag, a Recipe Tag and Widget Property.

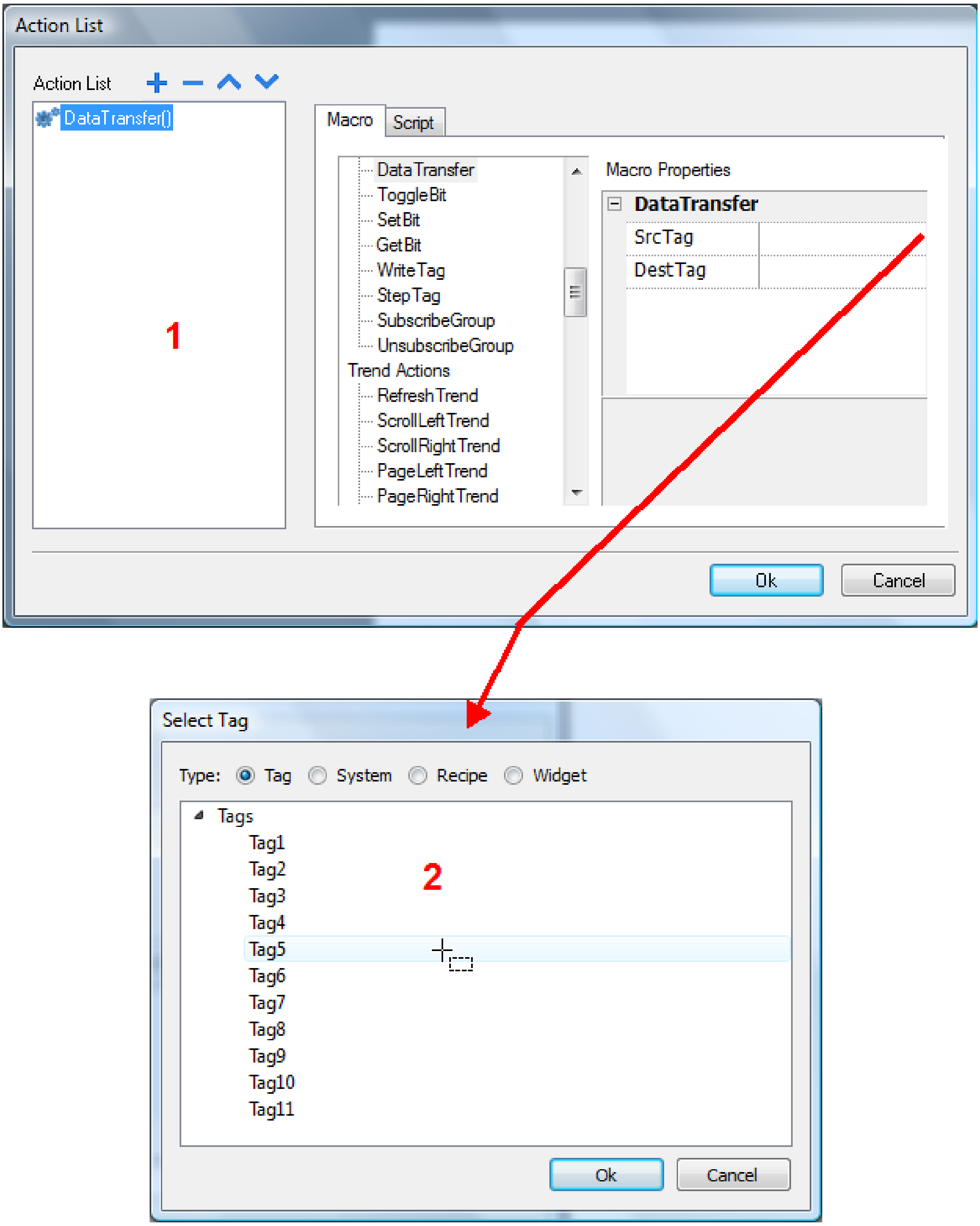
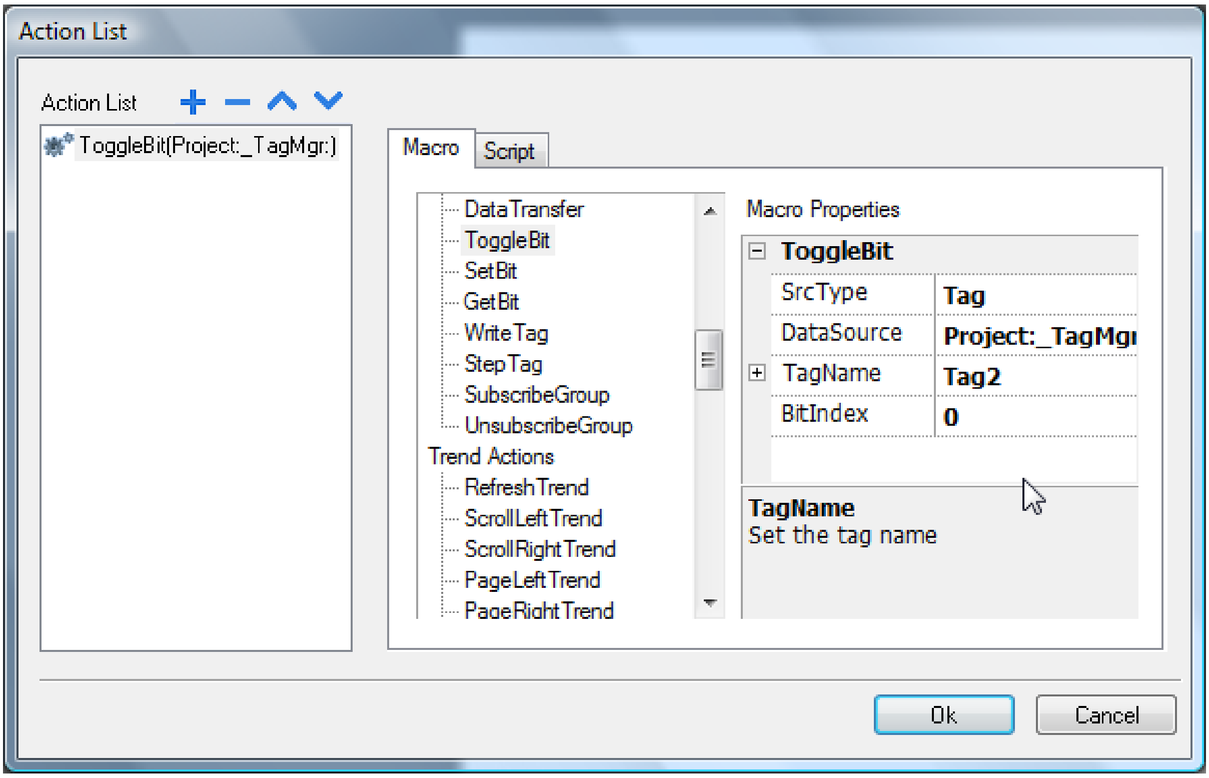


Figure 58

7.4.2 Toggle Bit

Toggle Bit macros allow you to “Toggle" (meaning set or reset) a Bit of a word. The Bit Index allows you to select the Bit to be Toggled.



ToggleBit

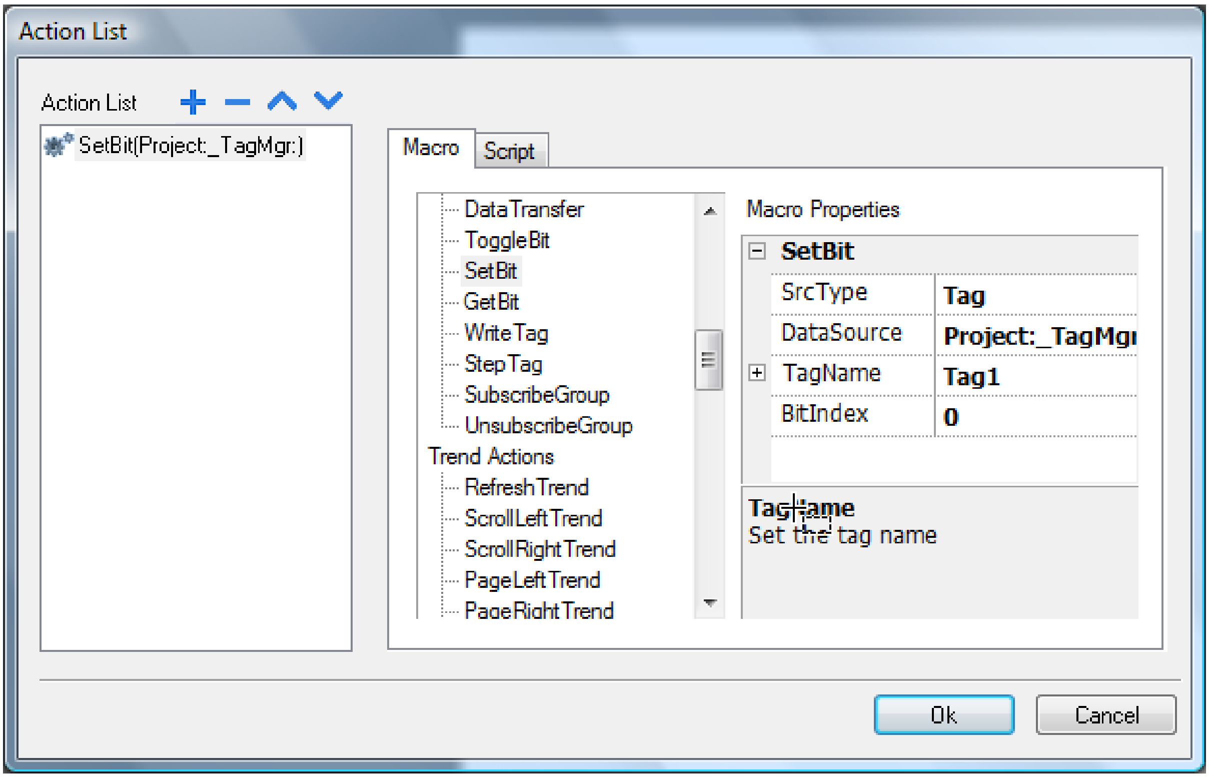
SrcType	Tag
DataSource	Project:_TagMgr
TagName	Tag2
BitIndex	0

TagName
Set the tag name

Figure 59

7.4.3 Set Bit

The Set Bit Index macro allows you to set the selected Bit. When the macro is executed, the selected Bit value is set to “1".



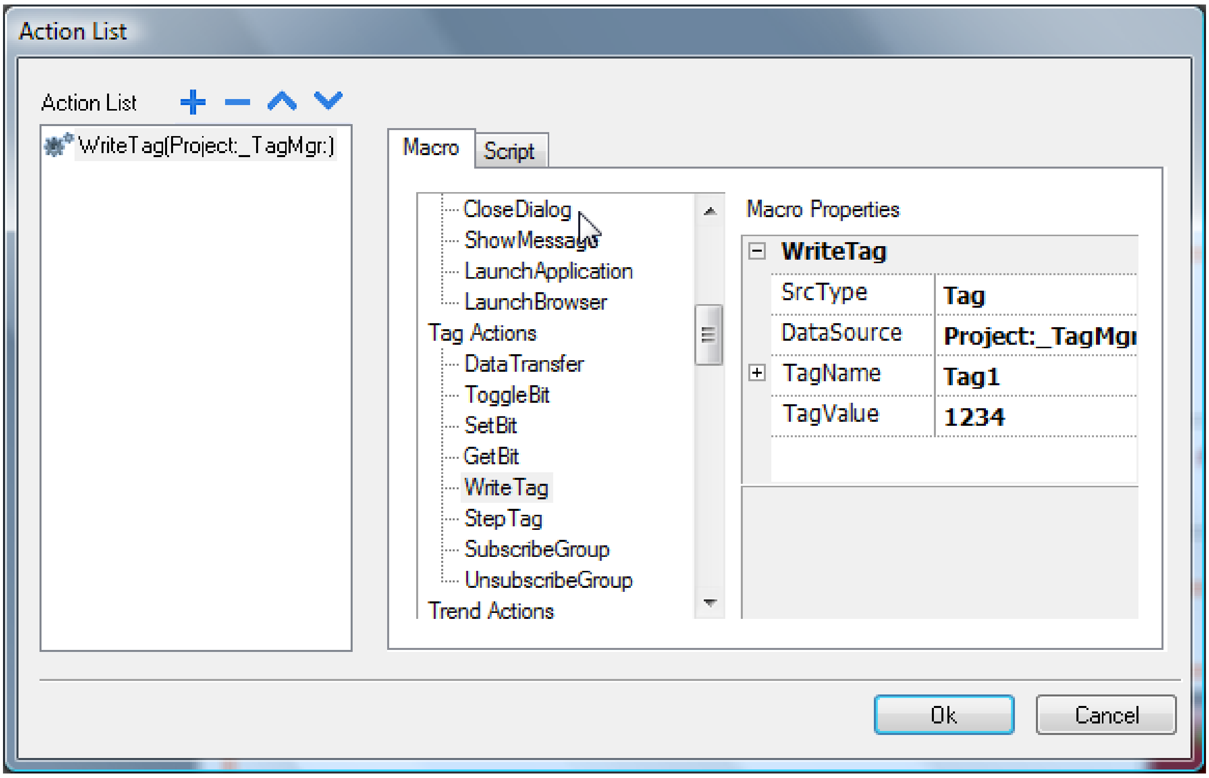
SetBit	
SrcType	Tag
DataSource	Project:_TagMgı
TagName	Tag1
BitIndex	0

TagName
Set the tag name

Figure 60

7.4.4 Write Tag

This command allows you to write constant values for the controller memory. In the action list, specify the Tag name and the constant value to be written.



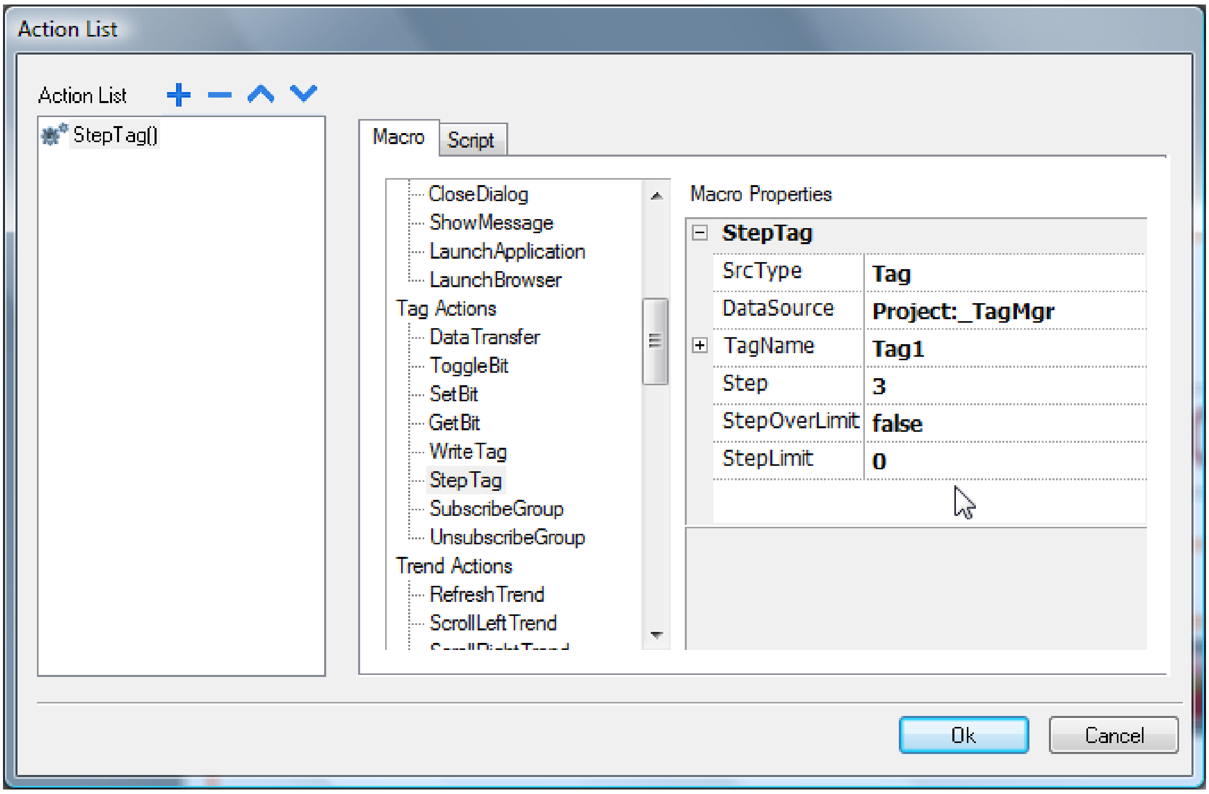
WriteTag	
SrcType	Tag
DataSource	Project:_TagMgı
TagName	Tag1
TagValue	1234

Figure 61

7.4.5 Step Tag

The Step command allows you to increment or decrement, in steps, the content of a Tag value.

SrcType :	The Source of the Tag
Data Source:	It depends on the “SrcType"
Tag Name:	Tag Name that you want to Step
Step:	Step value
Step Over Limit:	Step Limit enable
Step Limit:	If the Step Over Limit is True, then the macro will work until the Tag value reaches the specified level.



StepTag	
SrcType	Tag
DataSource	Project:_TagMgr
TagName	Tag1
Step	3
StepOverLimit	false
StepLimit	0

Figure 62

7.5 Trend Actions

Trend actions are used for both Live Data Trends and the Historical Trends Widget.

7.5.1 Refresh Trend

The Refresh Trend macro is used to refresh the Historical Trend window. You have to specify the Trend Widget in the Macro properties.

7.5.2 Scroll Left Trend

The Scroll Left Trend window is used to scroll the Trend window to the left side, by a one-tenth (1/10) page duration. For the Live Data Trend, the Scroll Left Trend works when the Trend is paused.

7.5.3 Scroll Right Trend

The Scroll Right Trend window is used to scroll the Trend window to the right side, by a one-tenth (1/10) page duration. For the Live Data Trend, the Scroll Right Trend works when the Trend is paused.

7.5.4 Page Left Trend

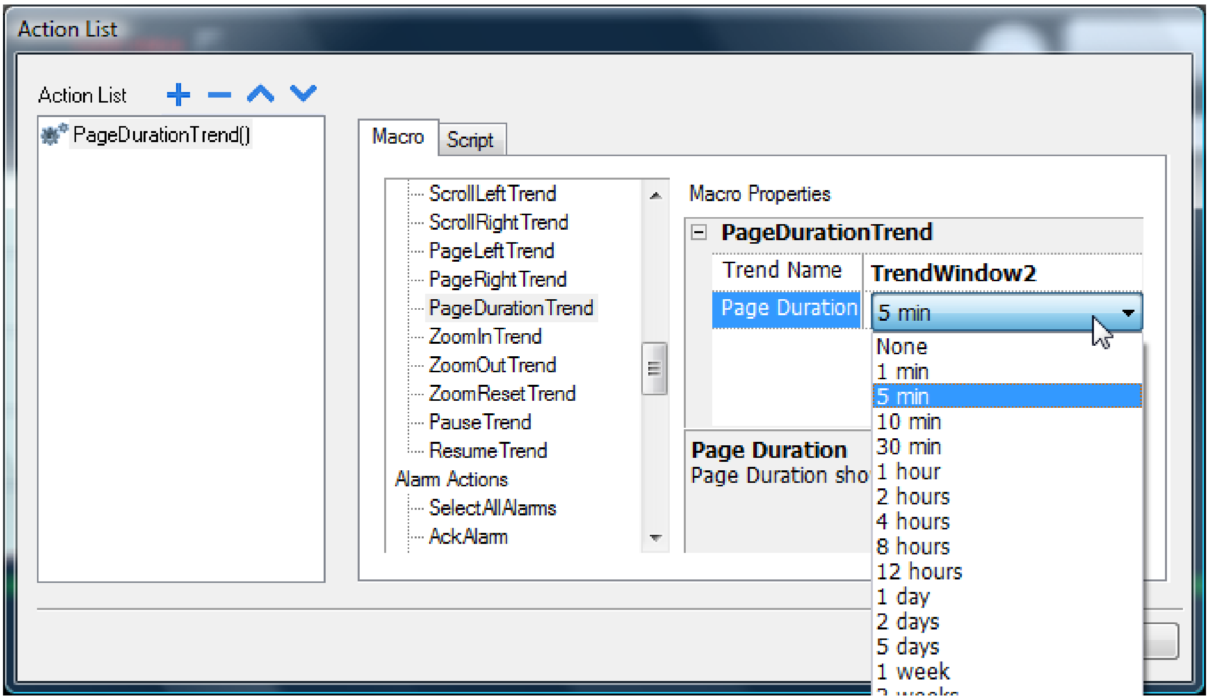
The Page Left Trend allows you to scroll the Trend window by a one-page duration. For example, if the page duration 10 minutes, then, with the Page Left Trend macro you can scroll the trend left by 10 minutes.

7.5.5 Page Right Trend

The Page Right Trend allows you to scroll the Trend window by a one-page duration. For example, if the page duration is 10 minutes, then, with the Page Right Trend macro, you can scroll the trend right by 10 minutes.

7.5.6 Page Duration Trend

The Page Duration macro is used to set the page duration of the Trend window. In Macro Properties, you must define the Trend Window and Duration.



PageDurationTrend

Trend Name: TrendWindow2

Page Duration: 5 min

Figure 63

Note: You can also use a combo box Widget to select the page duration in Runtime.

7.5.7 Zoom In Trend

Zoom In Trend macro allows you to reduce the page duration.

7.5.8 Zoom Out Trend

Zoom Out Trend macro allows you to make the page duration longer.

7.5.9 Zoom Reset Trend

Zoom Reset macro allows you to reset the zoom level back to the original zoom level.

7.5.10 Pause Trend

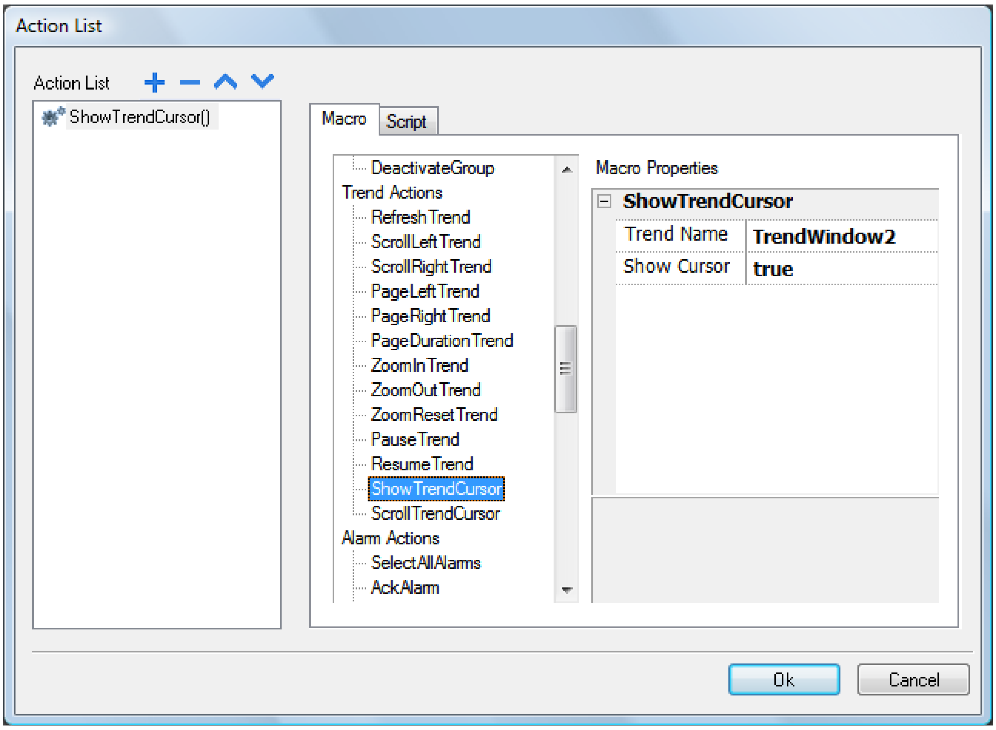
Pause Trend macro allows you to stop plotting the Trend curves in the Trend window. The Trend logging operation is not stopped from the panel when this macro command is used.

7.5.11 Resume Trend

Resume Trend macro allows you to resume a Trend plotting you previously paused. After executing the Resume Trend macro, the Trend window will start to plot the data to the Trend once again.

7.5.12 Show Trend Cursor

The Trend Cursor allows the user to ascertain the value of the curve at a given point on the X-Axis. Use Trend macros to activate the Trend Cursor. In Runtime, upon executing the macro, a Vertical Line (Cursor) will display in the Trend Widget. When the Graphic Cursor is enabled, the scrolling of the Trend is stopped. You can implement Scroll Cursor macros to move the Graphic Cursor over the curves, or to move the entire Trend window.



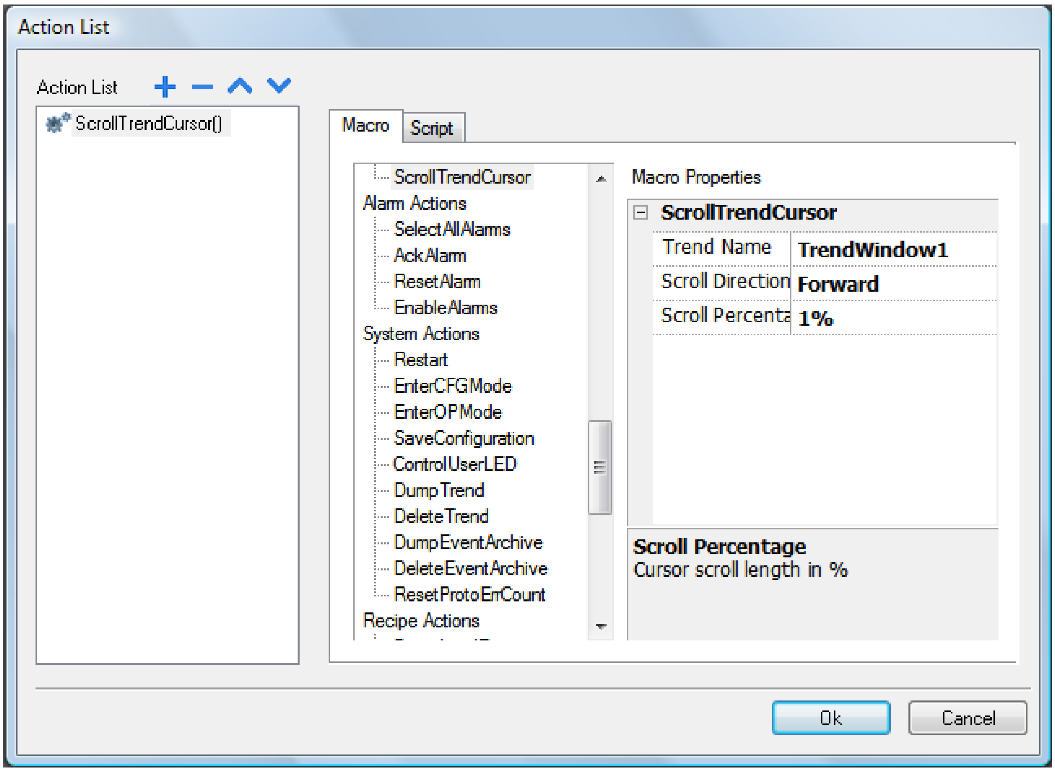
ShowTrendCursor

Trend Name	TrendWindow2
Show Cursor	true

Figure 64

7.5.13 Scroll Trend Cursor

The Scroll Trend Cursor allows the user to scroll through the Trend Cursor in forward or reverse, thereby moving the Cursor to the point at which you want the value of the Trend. The Y-Cursor value will display the Trend value at the point of the cursor. The scrolling percentage can be set at 1% or 10%. The percentage is calculated based on the Trend window duration.



ScrollTrendCursor	
Trend Name	TrendWindow1
Scroll Direction	Forward
Scroll Percenta	1%

Scroll Percentage
Cursor scroll length in %

Figure 65

7.6 Alarm Actions

Alarm Actions are macros used to acknowledge or reset the Alarms. The actions listed here can be used to build a custom Widget for the Alarms display; you can observe an example of how these are used in the default Alarm Widget, available in the Widget gallery.

7.6.1 Select All Alarms

This macro allows you to select all the Alarms in the Alarm Widget.

7.6.2 Ack Alarm

The Ack Alarm macro allows for acknowledging the selected Alarms.

7.6.3 Reset Alarm

The Reset Alarm macro allows you to reset the selected acknowledged Alarms.

7.7 System Actions

The System Actions macro allows you to use the system properties in Runtime.

7.7.1 Restart

The Restart system macro allows you to restart JMobile Runtime. After executing the macro, the JMobile server goes to configuration mode and restarts Runtime.

7.7.2 Enter Configuration Mode

This macro is used to switch the JMobile Runtime to Configuration mode (“Config").

7.7.3 Enter Operation Mode

This macro is used to switch the JMobile Runtime to Operation Mode.

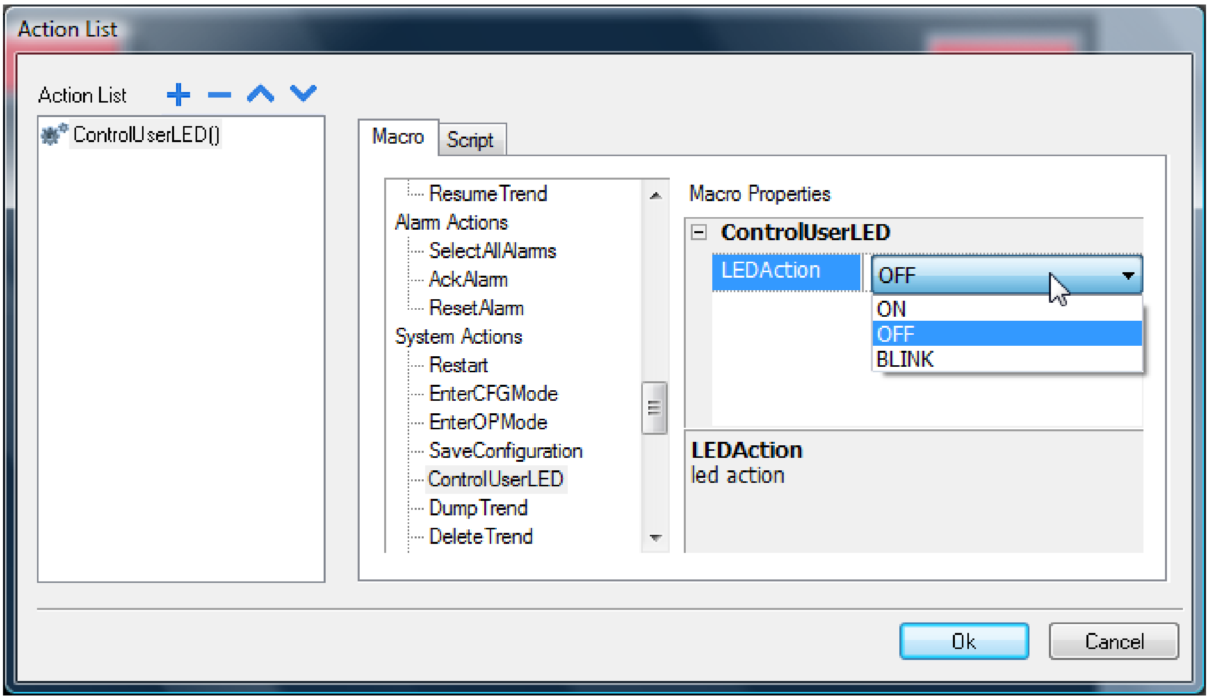
Note: All the macros will work only when Runtime runs in Operation Mode. If Runtime is in Configuration Mode, the macros will not work.

7.7.4 Save Configuration

The Save Configuration macro allows you to save Runtime settings, like schedulers, time, page, duration of Trends, etc., to the system files.

7.7.5 Control User LED

This allows the switching ON, OFF or blinking of the User LEDs. In the action properties, the LED action can be set to OFF, ON or Blink.



ControlUserLED

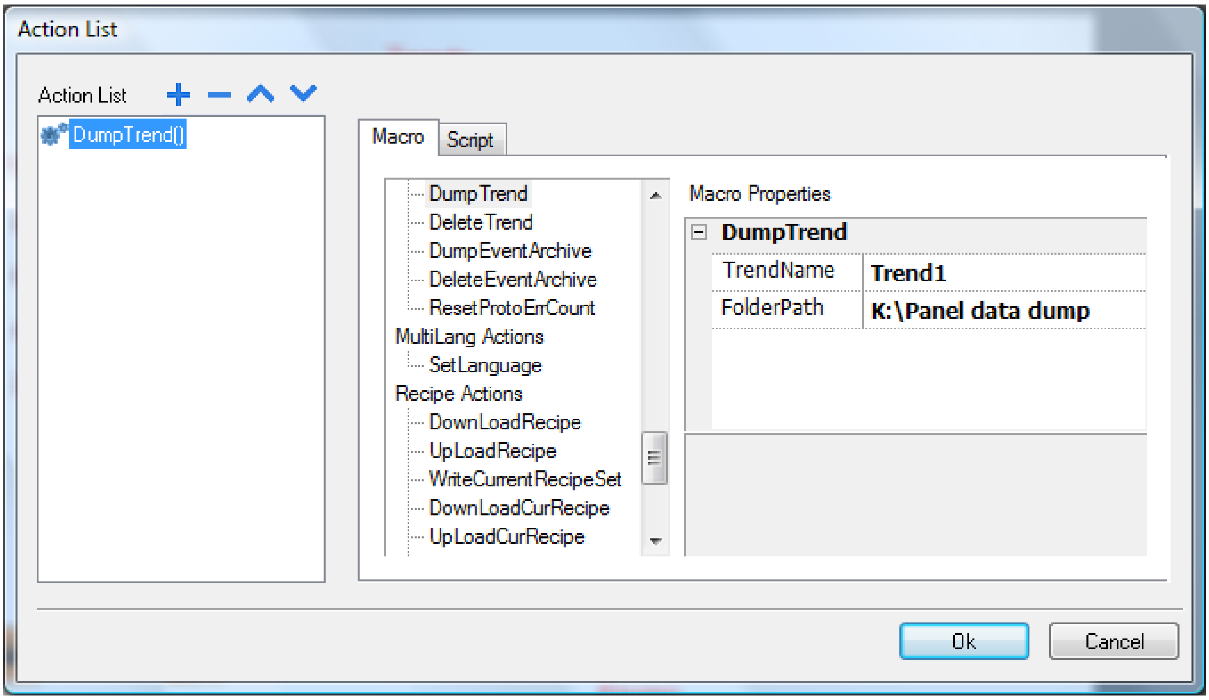
LEDAction	OFF

LEDAction
led action

Figure 66

7.7.6 Dump Trend

The Dump Trend macro is used to store the Historical Trend data to external drives, such as a hard drive or USB memory. In macro properties, you must configure the Historical Trend name you want to store and the destination folder path. If you use a USB pen drive plugged into the USB port, the path will be "\USBMemory", followed by the specified folder in the memory (or left empty for root folder).



DumpTrend

TrendName	Trend1
FolderPath	K:\Panel data dump

Figure 67

The dumped Trend file comes in a binary format. An external utility is then required to convert it to a CSV format.
The utility is called “TrendEdit.exe"
The TrendEdit utility can be invoked using a batch file with the following syntax:

```
trendedit -r TREND\Trend1 csv\Trend1.csv 1
```

The resulting CSV file has 5 columns with the following meaning:

```
DataType,Value,Timestamp(UTC),SamplingTime(ms),Quality
```

Where:
DataType is a code that informs about the data type of the sampled Tag according to the following rules:

0: Empty
1: Boolean
2: Byte
3: Short
4: Int
5: Unsigned Byte
6: Unsigned Short
7: Unsigned Int
8: Float
9: Double

Value is the value of the tag sampled
Timestamp(UTC) is the timestamp in UTC format
SamplingTime(ms) is the sampling interval time in milliseconds
Quality informs about the tag value quality according to the following rules:

0: Bad
1: Uncertain
2: Not used
3: Good

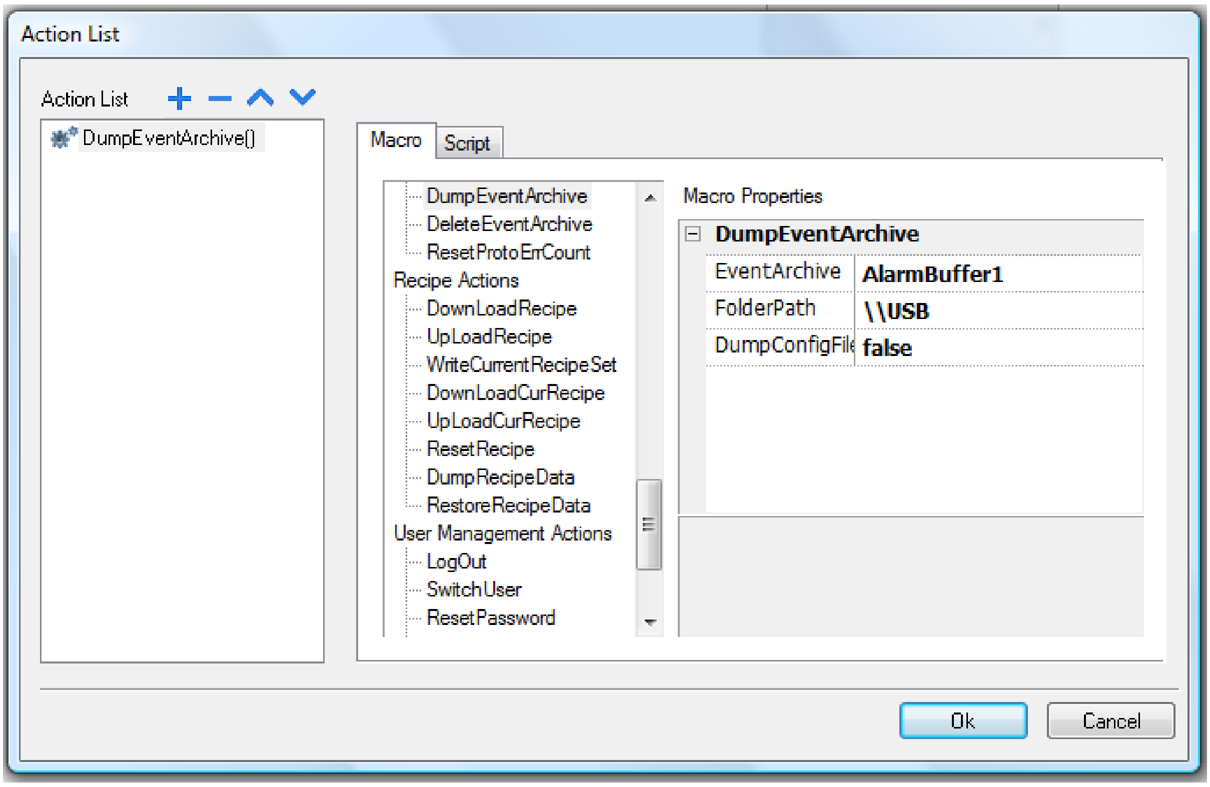
7.7.7 Delete Trend

The Delete Trend macro allows you to delete saved Trend data from the file. In macro properties, define the Trend name from which you want to delete the Trend logs.

7.7.8 Dump Event Archive

The Dump Event Archive macro is used to store Historical Alarm log and Audit Trails data to external drives, such as a hard drive or USB memory. In macro properties, you need to configure the Event buffer name that you want to store and the destination folder path, the DumpConfigFile property must be set to true when you plan to convert to CSV the dumped files.

The Events archive is dumped in a binary format; an external utility is then required to convert it to a CSV format.
Depending on the buffer type a specific utility has to be used. The utility called “AlarmBufferReader.exe” have to be used for Alarm buffers while the utility called “ArchiveEdit.exe" have to be used for Audit Trail buffers.



DumpEventArchive

EventArchive	AlarmBuffer1
FolderPath	\\USB
DumpConfigFil	false

Figure 68

7.7.9 Delete Event Archive

The Delete Event Archive macro allows you to delete saved Historical Alarms log data from the file. In macro properties, define the Event buffer name that you want to delete from the Trend logs.

7.7.10 Reset ProtoErr Count

The Reset Protocol Err Count macro is used to reset the protocol error count on the system variables.

7.8 Multi Lang Actions

The Multi Language macros are used to select and modify the current display languages.

7.8.1 Set Language

The Set Language macro allows you to set the current display language. In macro properties, enter into the Language index. In Runtime, while executing the macro, the selected language will be applied to all applicable Widgets.

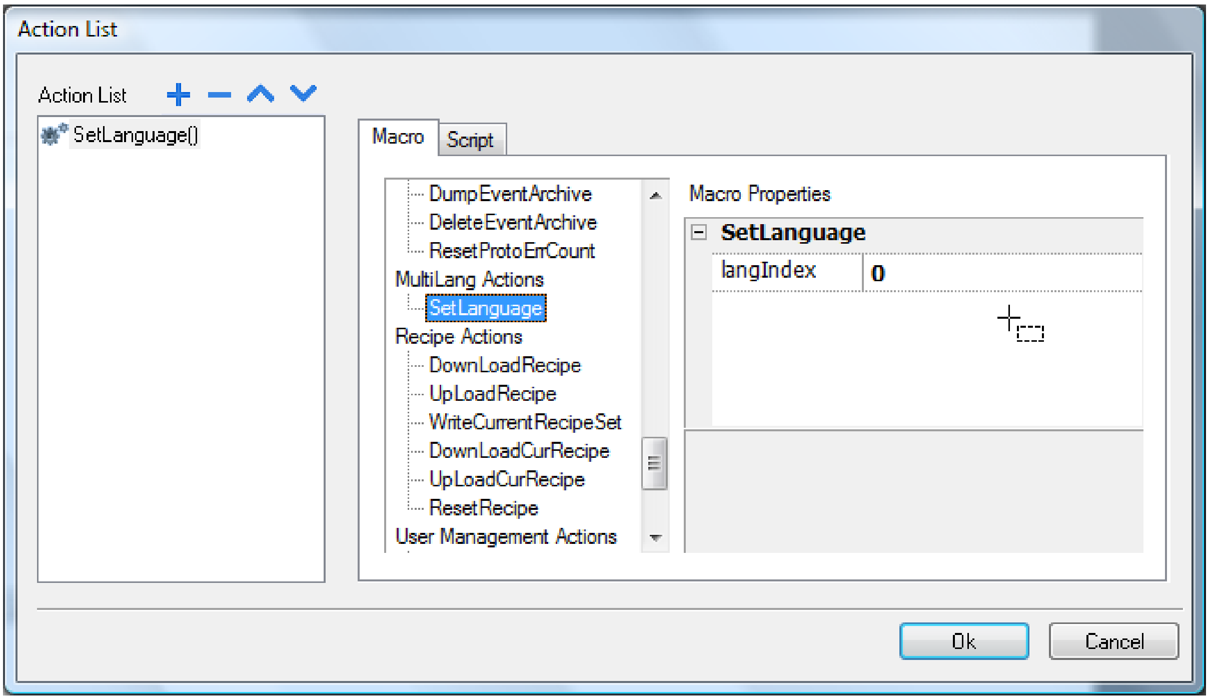


Figure 69

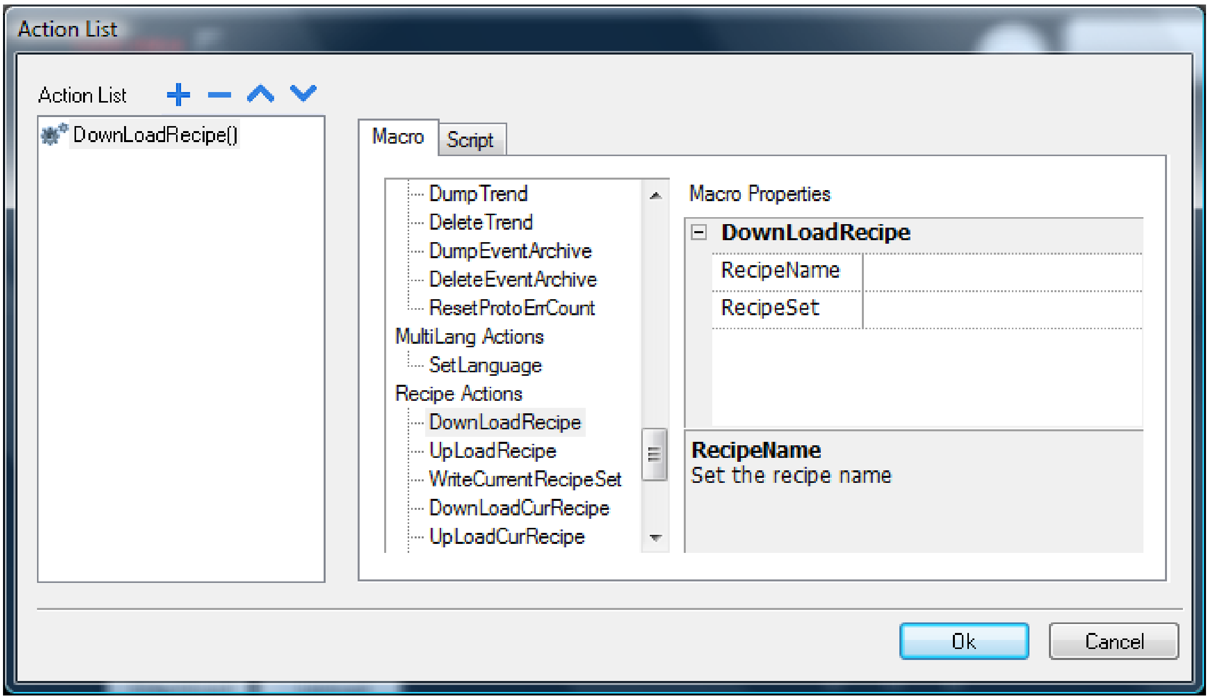
Note: The Language index for a language can be found in the Multilanguage editor.

7.9 Recipe Actions

The Recipe Actions macros are used in Recipes.

7.9.1 Download Recipe

The Download Recipe macro allows you to transfer the set of Recipe data to controller Tags. In macro properties, select the Recipe in the Recipe Name field and select the Recipe set you want to download. To download a selected Recipe set, select “curset" in the Recipe set.



RecipeName
Set the recipe name

Figure 70

7.9.2 Upload Recipe

The Upload Recipe macro allows you to transfer the set of controller data to the Recipe data. In macro properties, select the Recipe in the Recipe Name and select the Recipe set that you want to upload. To upload a selected Recipe set, select “curset" in the Recipe set.

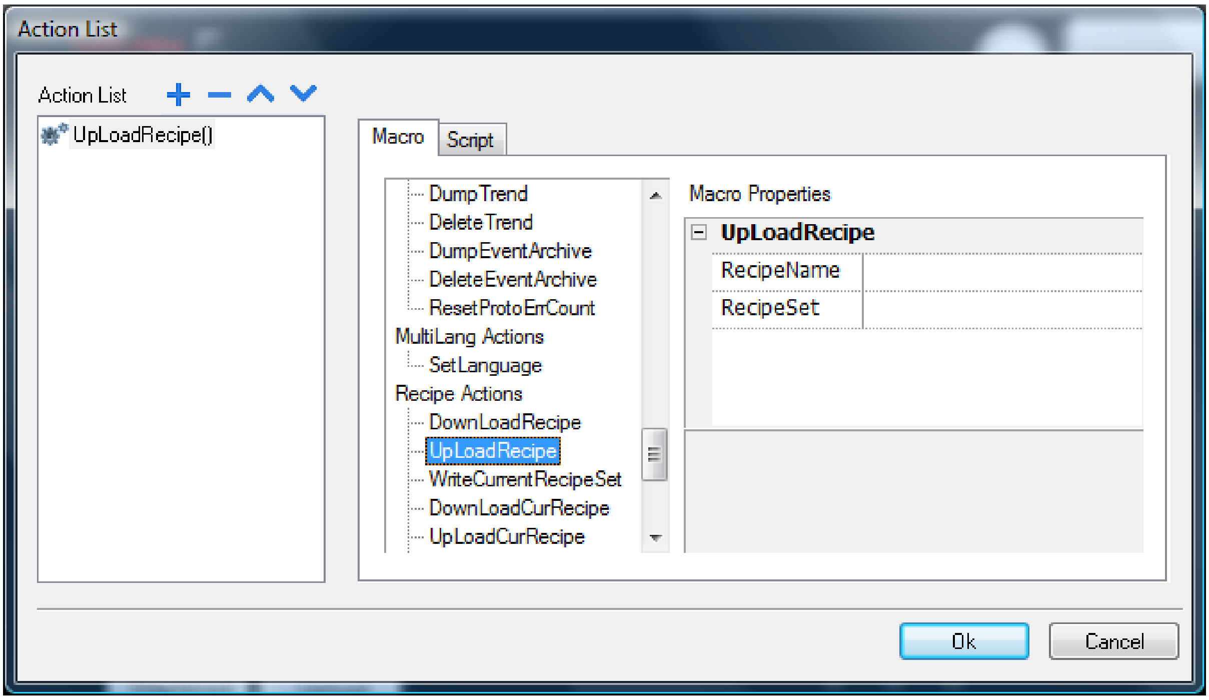
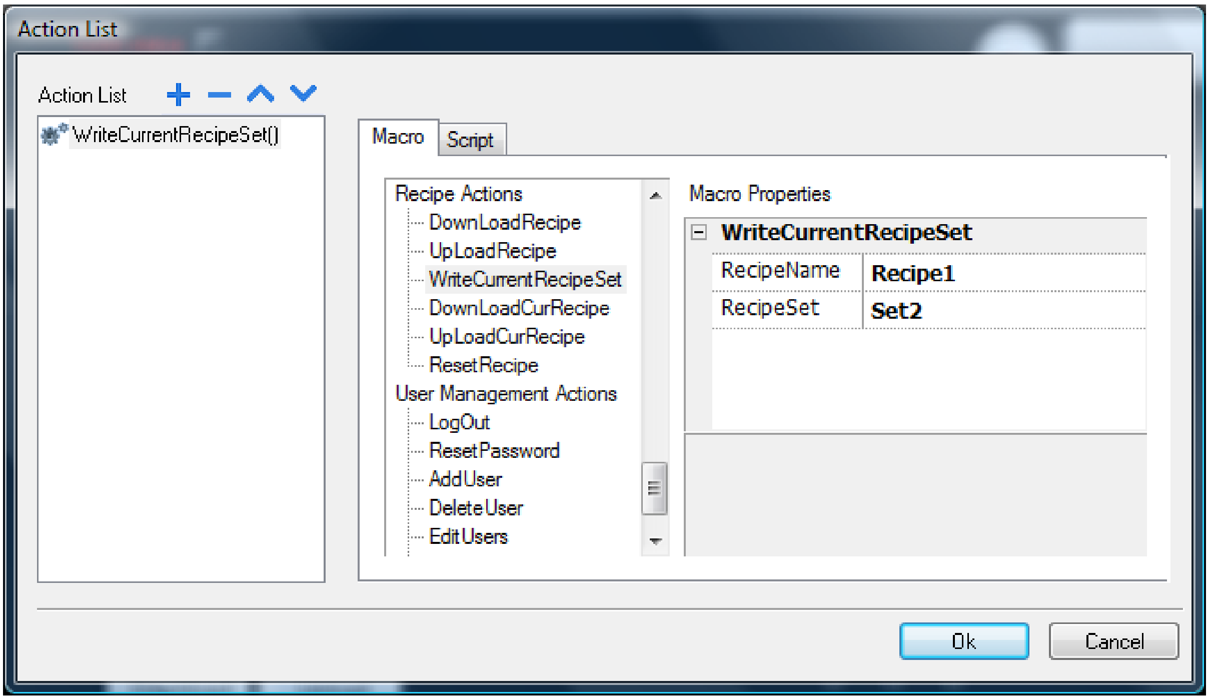


Figure 71

7.9.3 Write Current Recipe

The Write Current Recipe macro allows you to set the selected Recipe as current Recipe Set. In macro properties, select the Recipe and Recipe Set you want to set as the Current Recipe in Runtime.



WriteCurrentRecipeSet

RecipeName	Recipe1
RecipeSet	Set2

Figure 72

7.9.4 Download Current Recipe

The Download Current Recipe macro allows you to transfer the set of Recipe data to the controller Tags. No parameter is required to set this in the macro parameters. This will download the currently selected Recipes and Recipe set to the controller Tags.

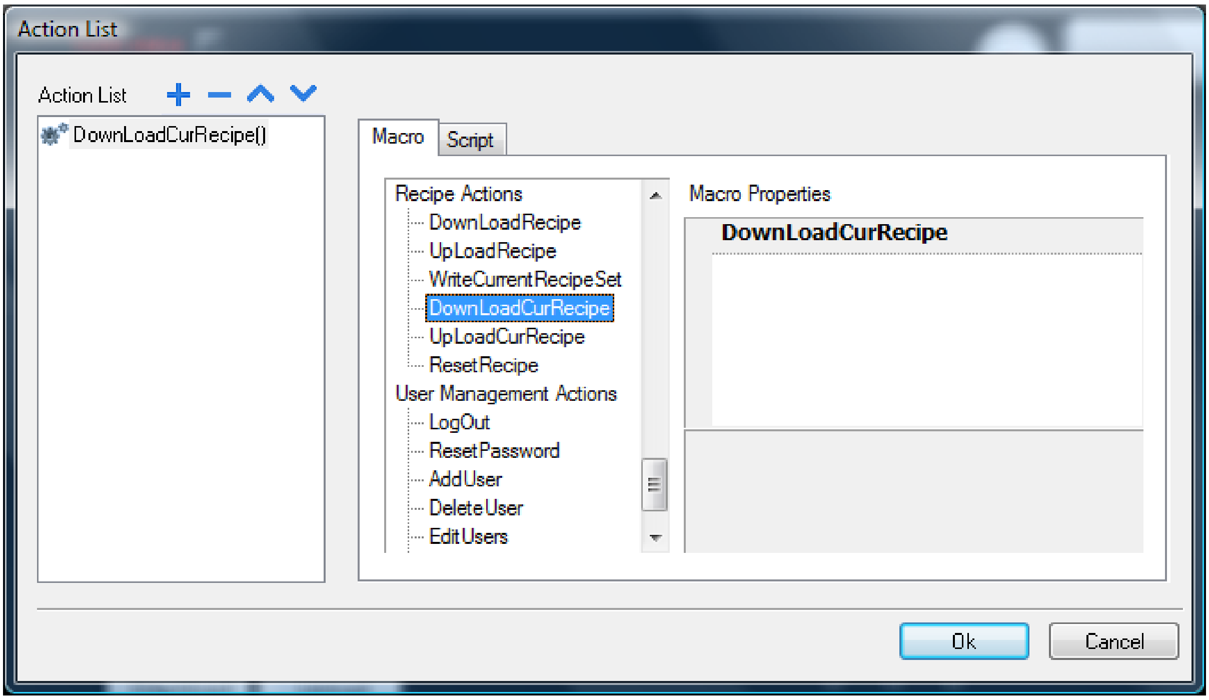


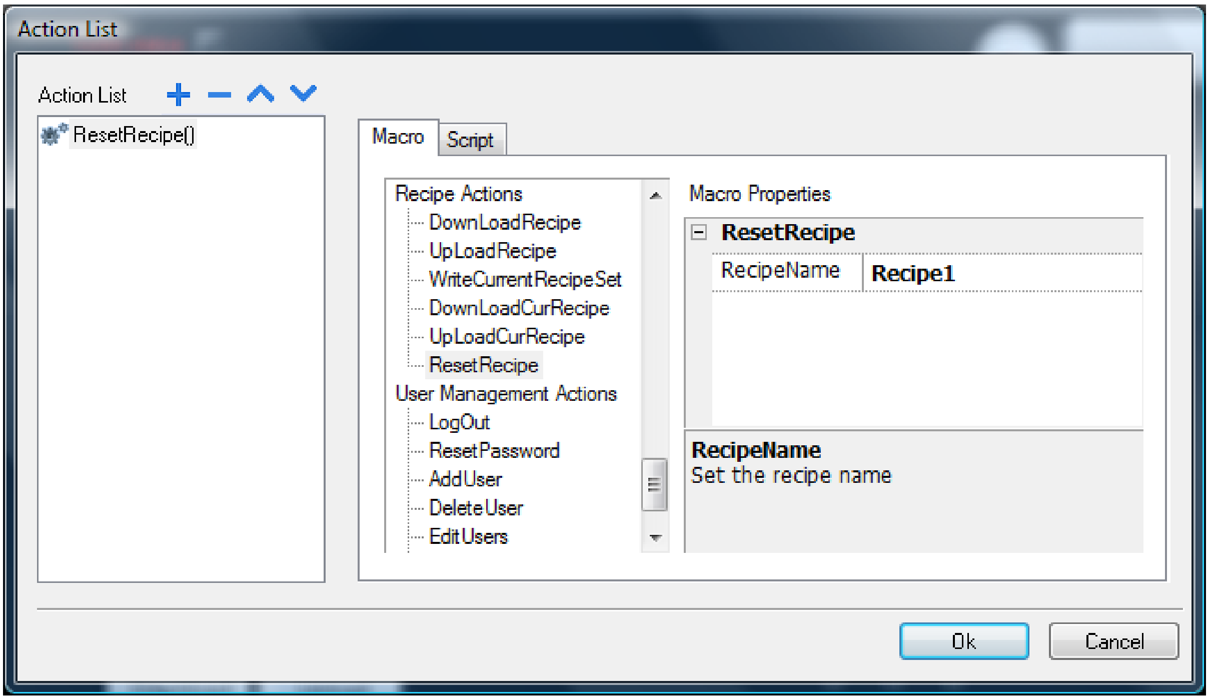
Figure 73

7.9.5 Upload Current Recipe

The Upload Current Recipe macro allows you to transfer the set of controller Tags data to Recipes. No parameter is required to set this in the macro parameters. This will upload the currently selected Recipes from the controller Tags.

7.9.6 Reset Recipe

The Reset Recipe macro allows you to restore the factory settings for the Recipe data. The uploaded Recipes will be replaced with the original Recipe data. In the macro property, select the Recipe that you want to reset to factory settings.



ResetRecipe

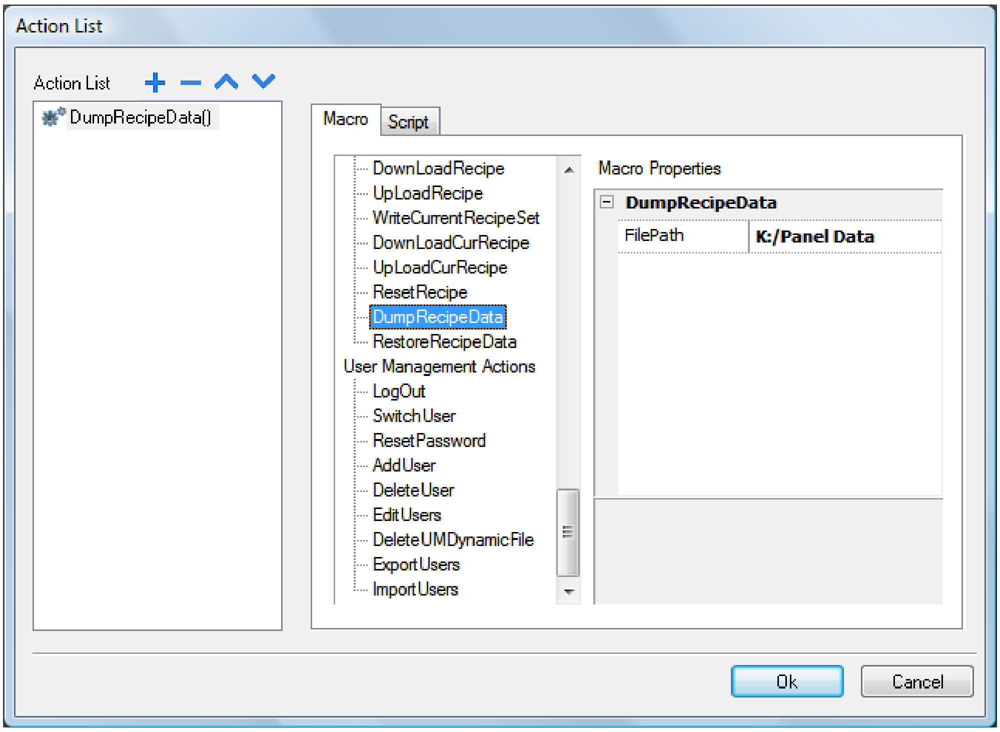
RecipeName	Recipe1

RecipeName
Set the recipe name

Figure 74

7.9.7 Dump Recipe Data

The Dump Recipe Data macro allows you to dump the Recipe to a USB Stick or other memory system, in CSV format. In macro properties, define the file location at which the dumped file is to be saved.
Recipe data is saved in CSV format.



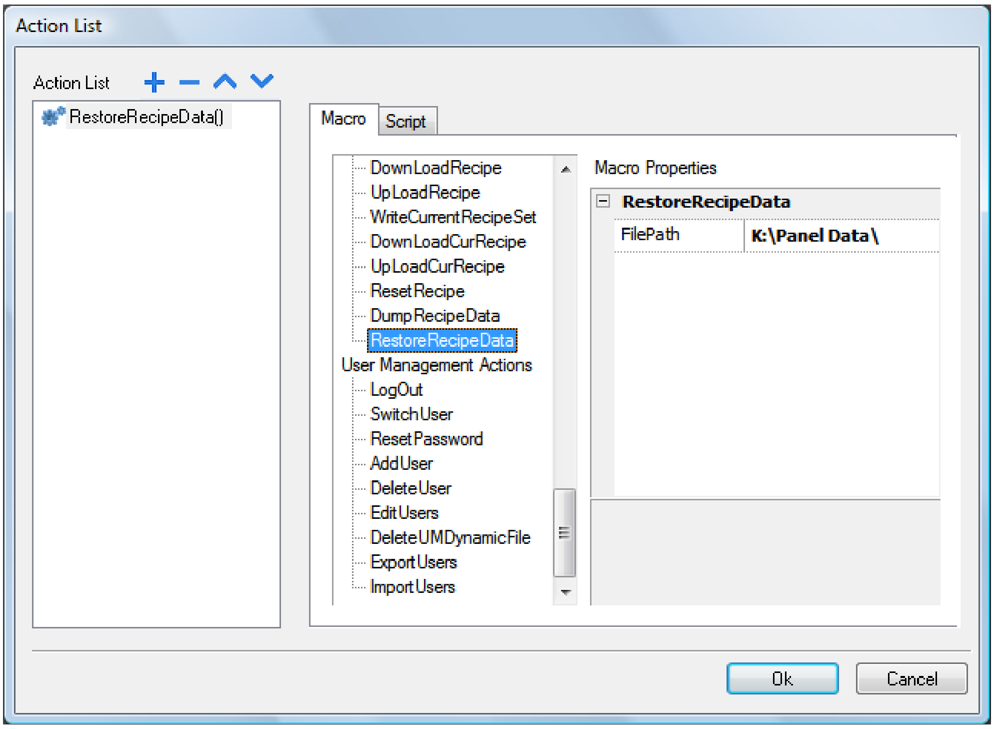
DumpRecipeData

FilePath	K:/Panel Data

Figure 75

7.9.8 Restore Recipe Data

The Restore Recipe Data macro allows you to restore the Recipe data using the saved Recipe data file from the USB Stick or from another system memory. In macro properties, provide the file path of the Recipe files.



RestoreRecipeData

FilePath	K:\Panel Data\

Figure 76

7.10 User Management Actions

The User Management Actions macros can be utilized for user management and security settings in Runtime.

7.10.1 LogOut

The Logout macro allows you to log off the current user in Runtime. After executing the Logout macro, HMI Runtime will start up the Logon page. No parameter is required to set this macro.

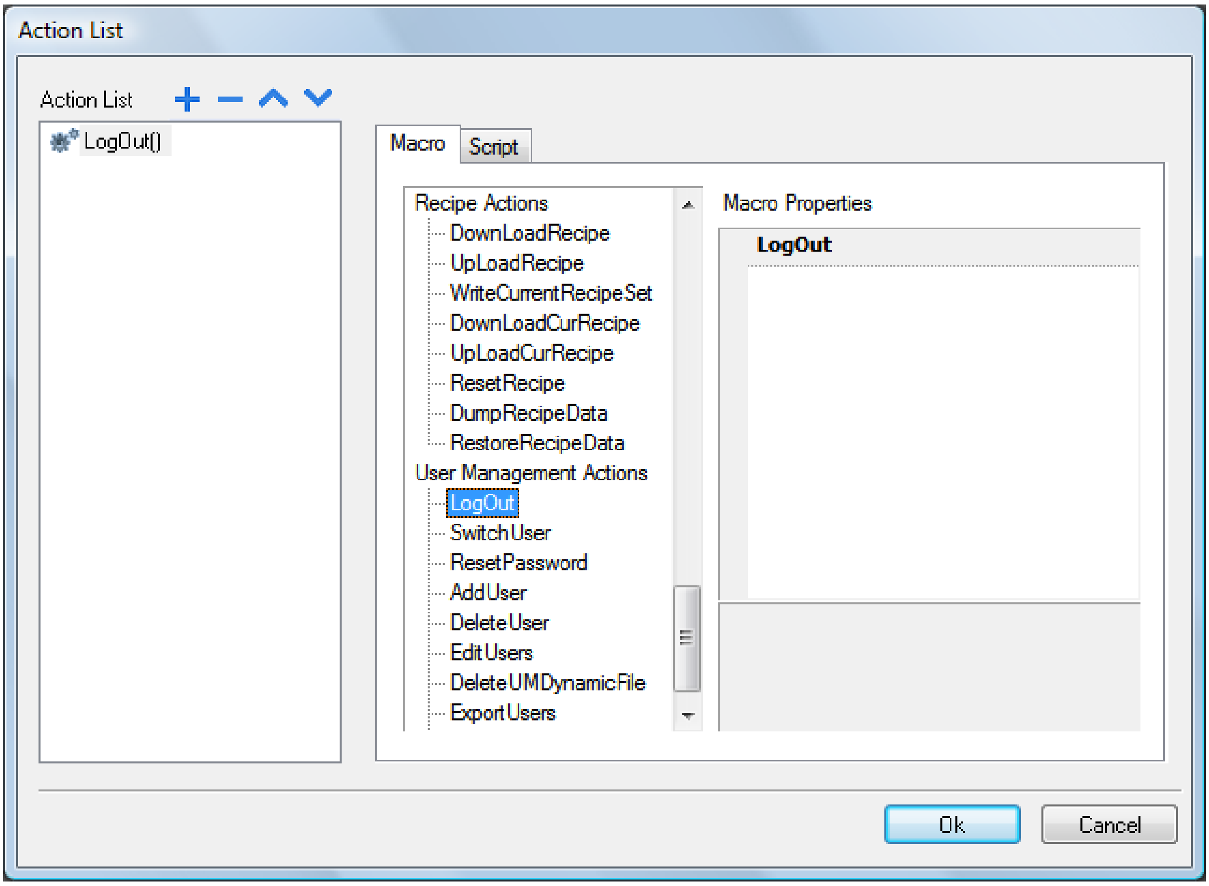


Figure 77

7.10.2 Switch User

The Switch user macro allows you to switch between two users without logging out the logged-in user. The server continues running with the previously logged-in user, until the next user logs in. This means, after executing the Switch User macro, Runtime will display the User Login template. Internally, however, the server runs with the previously logged-in user. This action is useful for ensuring that there is always one user logged into the system.

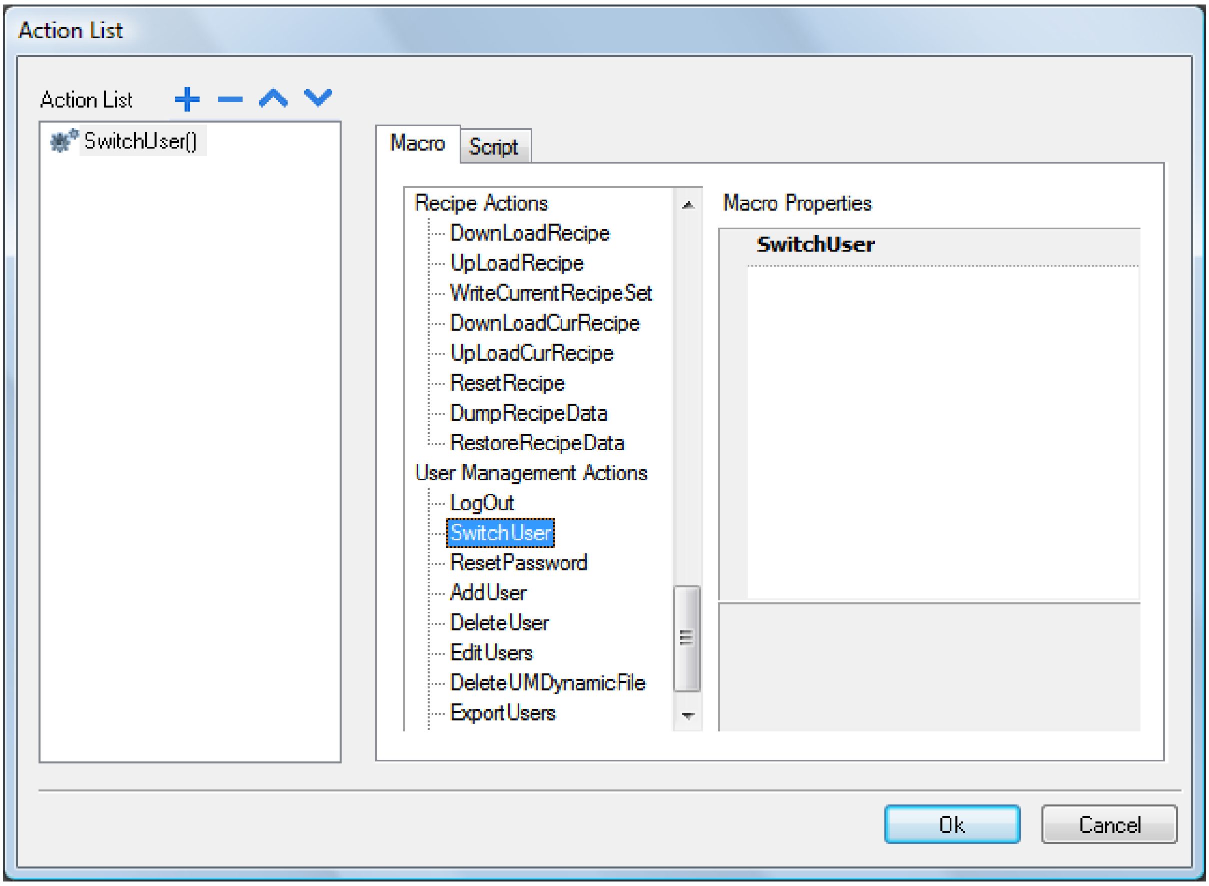


Figure 78

Click on the “Back" button to go back to the previously logged-in user.

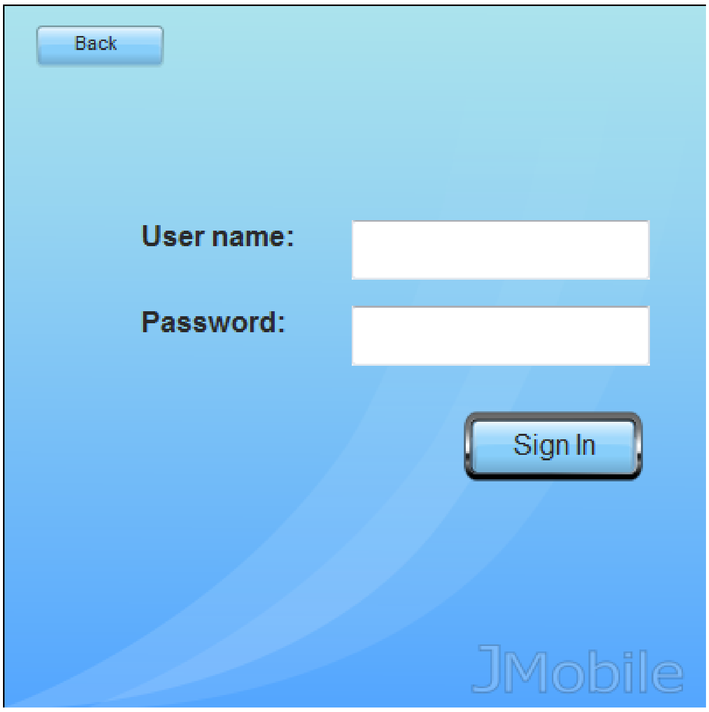


Figure 79

7.10.3 Reset Password

The Reset Password macro allows the current user to restore his or her original password; this macro will restore factory settings for the current user's password. No parameter is required to set this macro.

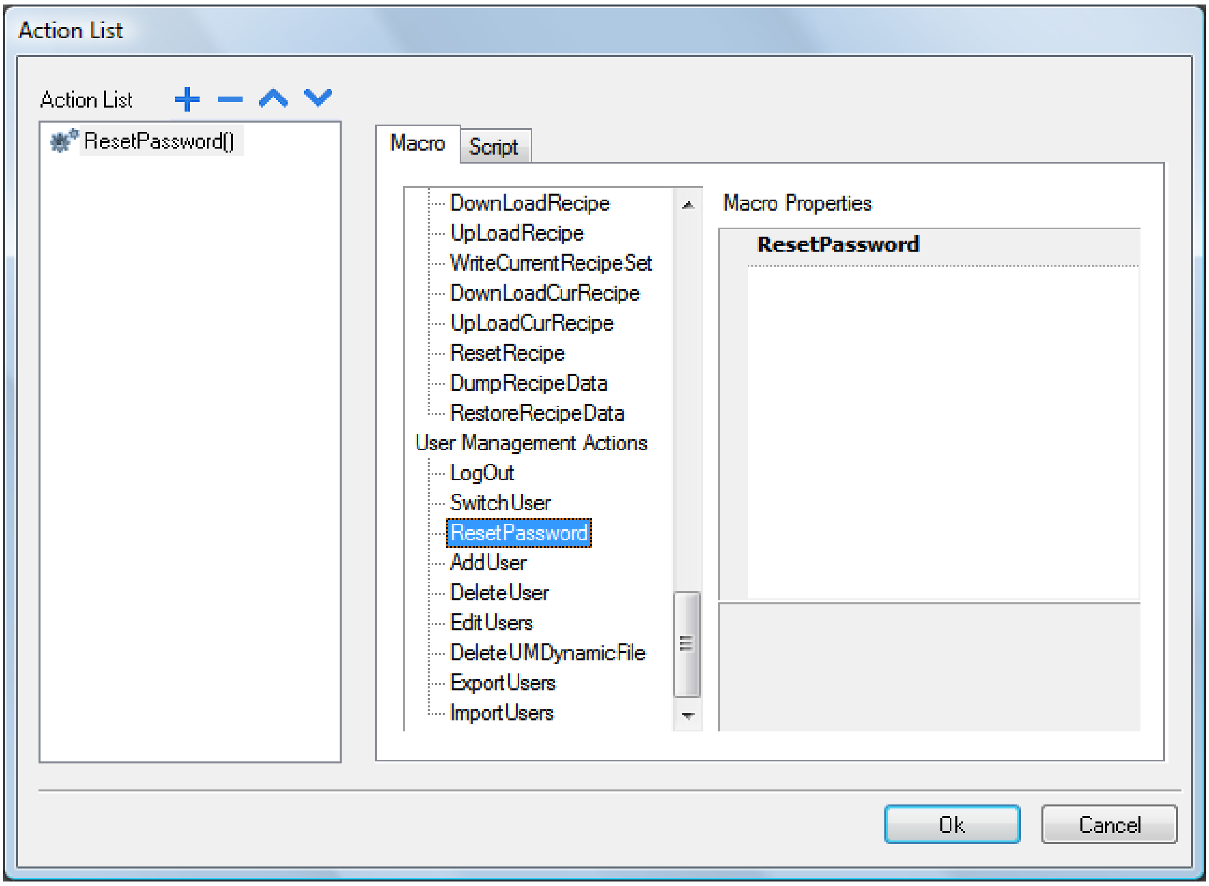
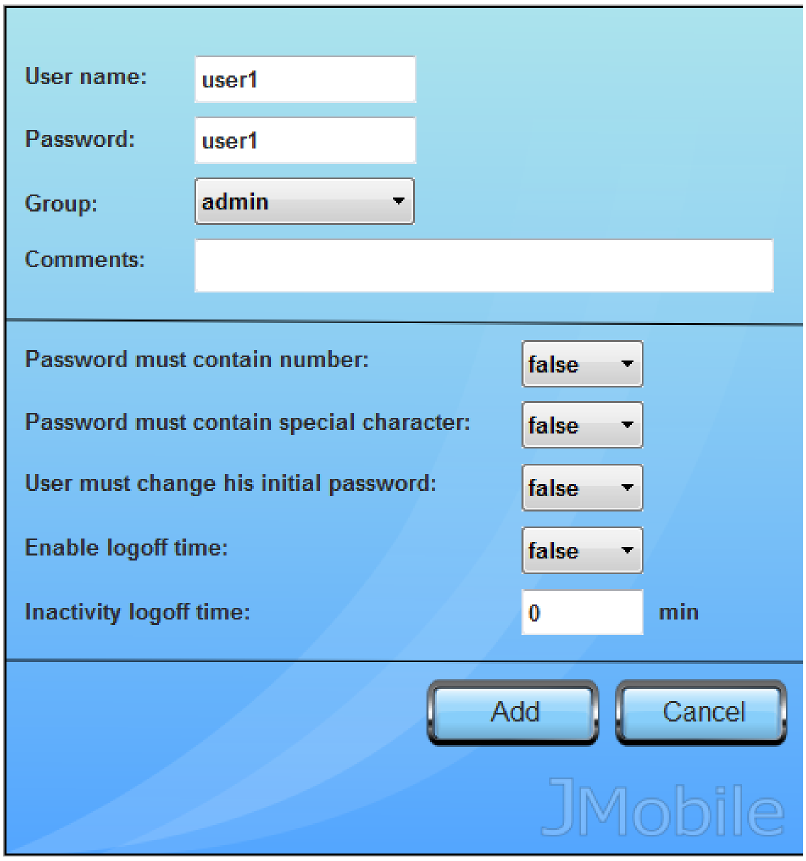


Figure 80

7.10.4 Add User

This macro is used to add users in Runtime. When this macro is executed, a template page pops up, where parameters for the user can be set. These parameters include Username, Password, Group, Comments, flags like 'password must contain numbers', 'password must contain special character', 'user must change his initial password', 'enable logoff time' and 'Inactivity Logoff Time'. The User Log is shown in Figure 79.



User name: user1

Password: user1

Group: admin

Comments:

Password must contain number: false

Password must contain special character: false

User must change his initial password: false

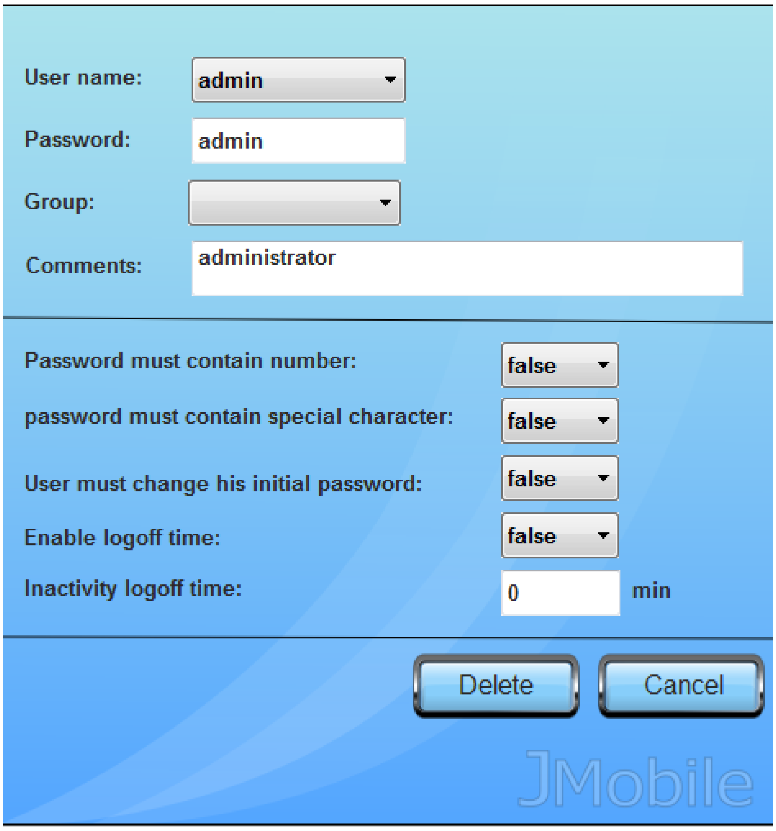
Enable logoff time: false

Inactivity logoff time: 0 min

Figure 81

7.10.5 Delete User

This macro is used to delete users at Runtime. Upon executing this macro, a template page will pop up where you can select the user you wish to delete. No parameters are required to set this macro. After executing the macro, the Delete User form will be displayed, as shown in Figure 82.



User name: admin

Password: admin

Group:

Comments: administrator

Password must contain number: false

password must contain special character: false

User must change his initial password: false

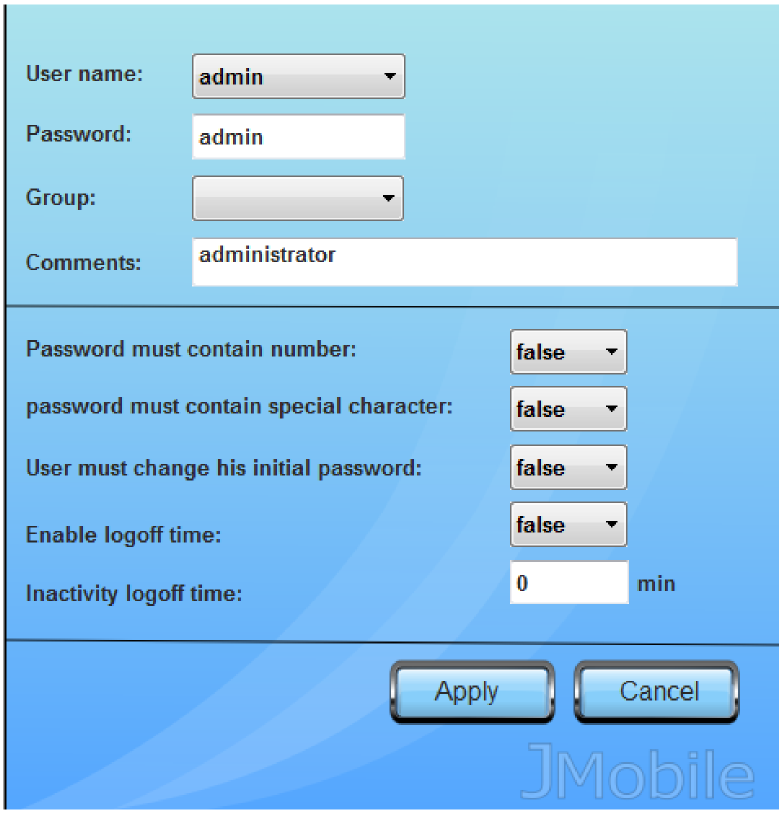
Enable logoff time: false

Inactivity logoff time: 0 min

Figure 82

7.10.6 Edit Users

This macro is used to edit users in Runtime. When executing this macro, a template page pops up. Here you can select a user and modify this user's parameters (such as Username, Password, Group, Comments, flags like 'password must contain numbers', 'password must contain special character', 'user must change his initial password', 'enable logoff time' and Inactivity Logoff Time). After executing the macro, a User Edit form will pop up, as shown in Figure 83.



User name: admin

Password: admin

Group:

Comments: administrator

Password must contain number: false

password must contain special character: false

User must change his initial password: false

Enable logoff time: false

Inactivity logoff time: 0 min

Figure 83

7.10.7 Delete UM Dynamic File

This macro allows you to delete the dynamic user management file. This means that the users created, edited, or deleted in Runtime will be erased, and the server will restore the settings originally downloaded from JMobile Studio.

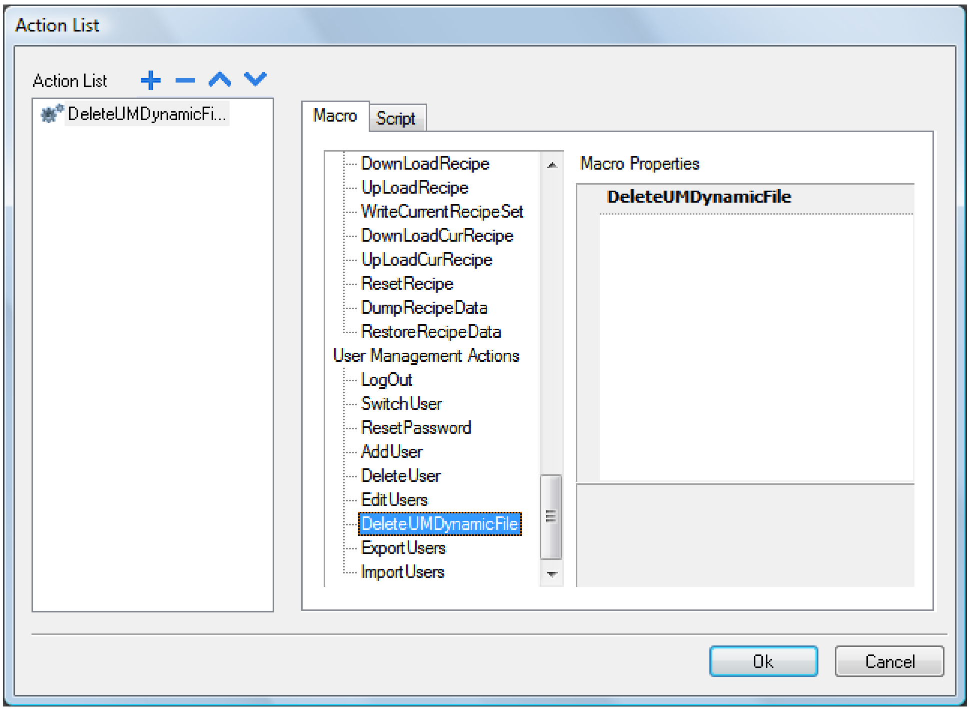
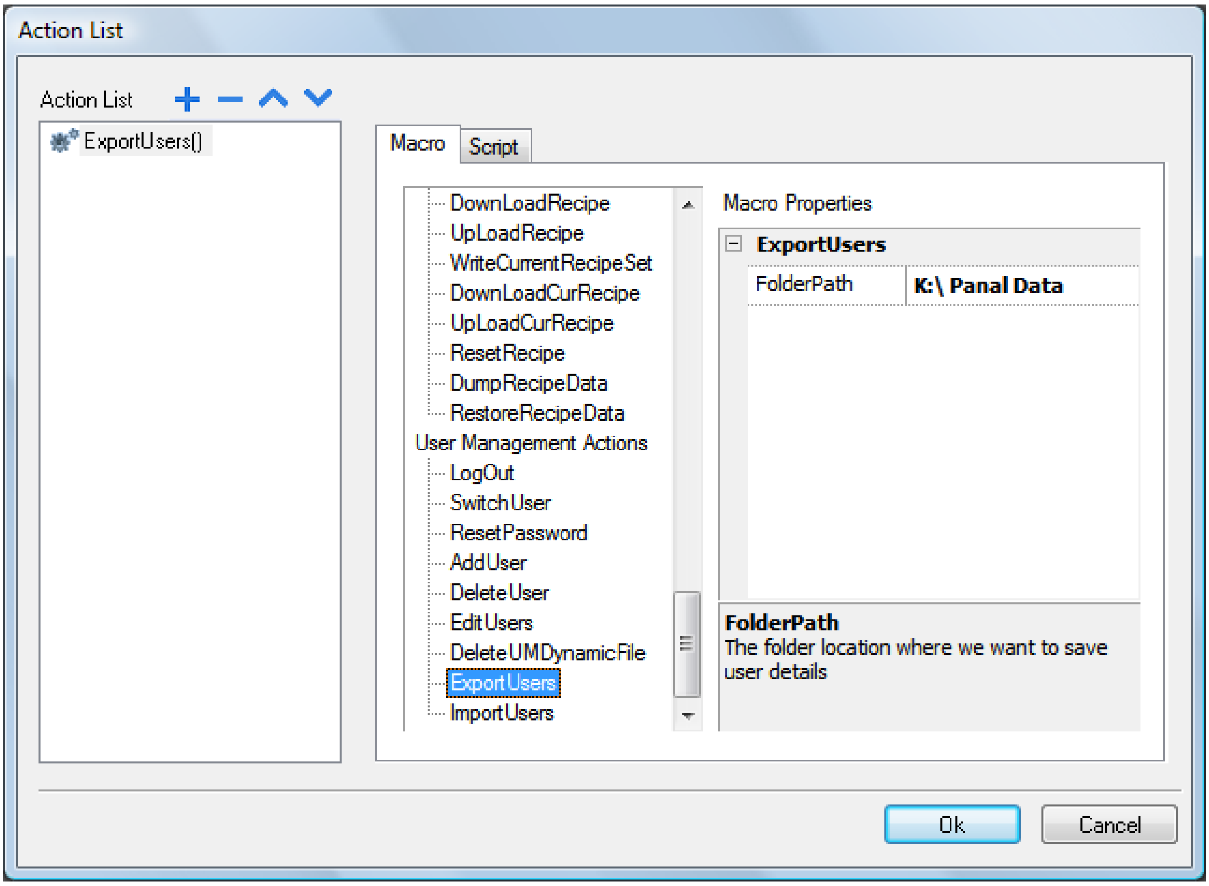


Figure 84

7.10.8 Export Users

This macro allows for exporting user details to an xml file (usermgnt_user.xml). User details will be in an encrypted form. In this macro property, the destination folder path must be set to the location where the usermgnt_user.xml file is saved.
If using a USB pen drive plugged in to the USB port, the path will be “\USBMemory", followed by the specified folder in the memory (or left empty for root folder).
Since the file is encrypted, there is no way to edit the user configuration from this exported file. This action is most useful for making a backup to be used for a later restore (see next chapter).



ExportUsers

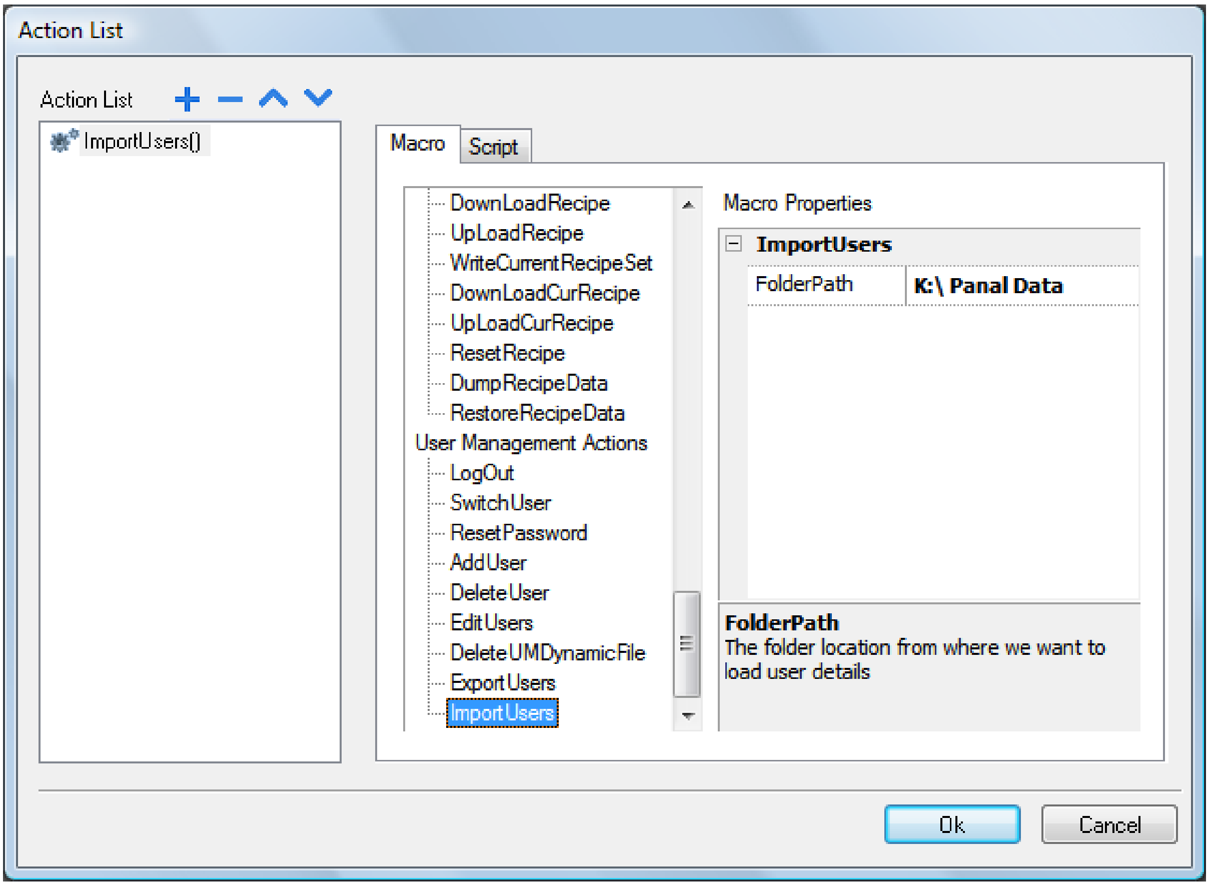
FolderPath: K:\ Panal Data

FolderPath
The folder location where we want to save user details

Figure 85

7.10.9 Import Users

This macro allows for importing user details from an xml file named, “usermgnt_user.xml". The folder path where the usermgnt_user.xml file is located must be specified within the Macro Properties.
If using a USB pen drive plugged in to the USB port, the path will be “\USBMemory", followed by the specified folder in the memory (or left empty for root folder).



ImportUsers

FolderPath K:\ Panal Data

FolderPath
The folder location from where we want to load user details

Figure 86

8 Working with Alarms and the Historical Alarms List

The Alarm feature is designed to provide alerts through pop-up messages, typically to issue a warning, or to indicate any abnormal conditions or any malfunctions on the machine. Whenever a Bit goes high, or the value of a Tag crosses the limit of deviation defined in the Alarm Configuration, the respective Alarm message(s) will be displayed in a special dialog. Or, alternatively, you can program certain macro actions to be executed when the Alarms are triggered.
Please note that, in JMobile, there is no default action associated to a triggered Alarm. The visualization of a specific page containing the Alarm Widget is optional, and the specific action executed when the trigger condition is verified can be any one of the actions found on the Action list.

The Alarm Configuration of an Alarm determines whether or not the alarm has to be acknowledged. It can also be used to determine how the Alarm appears when displayed on the HMI device(like Background and foreground color). Alarm Configuration also determines whether, and when, the corresponding alarm is logged in the Event list.

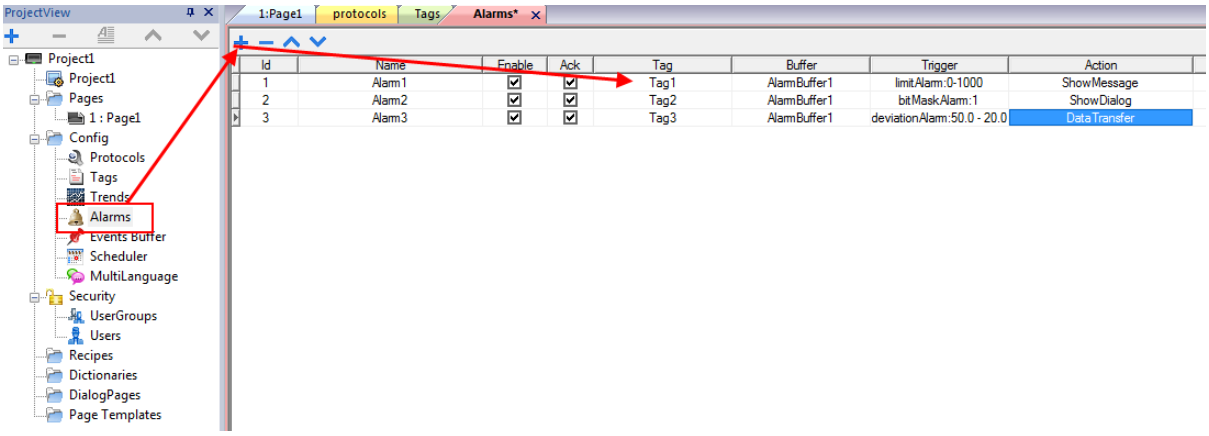
For Alarms displaying critical or hazardous operating and process statuses, a stipulation can be made requiring the plant operator to acknowledge the Alarm.

The Alarm window is configured in the screen's template and, thus, is a component of all screens of a project. More than one Alarm can be displayed simultaneously, depending on its configured size. An Event can trigger the closing and reopening of the Alarm window.

Please note that, in JMobile, working with Alarms is similar to working with Events. In general, there is no absolute need have a pop-up dialog when an Alarm is triggered. Any “background" action (from the list of available actions) can be associated with this Event.

8.1 Alarm Configuration Editor.

In the Project Workspace, double click on Alarms to open the Editor. Then add the Alarms by clicking the “+" button. If the Alarm requires ‘Acknowledge’, then click on the ‘Acknowledge; check box to enable the Acknowledgement.



Id	Name	Enable	Ack	Tag	Buffer	Trigger	Action
1	Alarm1	☑	☑	Tag1	AlarmBuffer1	limitAlarm:0-1000	ShowMessage
2	Alarm2	☑	☑	Tag2	AlarmBuffer1	bitMaskAlarm:1	ShowDialog
3	Alarm3	☑	☑	Tag3	AlarmBuffer1	deviationAlarm:50.0 - 20.0	DataTransfer

Figure 87

8.2 Name

Specifies the name of the specific Alarm.

8.3 Enable

A user can enable or disable the triggering of particular alarms. Alarms can be enabled or disabled on Runtime as well (for more information, please see Chapter 8.16).

8.4 Acknowledgment

For an Alarm that needs to be acknowledged by the operator (when the alarm is triggered), select the check box to enable the Acknowledgment. If checked, an operator is required to acknowledge this Alarm any time it is triggered.

8.5 Tag

Attach the Tag in which the Alarm shall periodically check the Tag value, so that the respective alarm(s) is triggered when this deviates from its limits. (The Alarm function will refer to the value of this Tag, or to the state of a Bit, in the case of Bitmask, to determine when to trigger the Alarm.)

8.6 Buffer

Specifies the Buffer file to which the Alarm history will be saved.

8.7 Trigger

This selection determines the type triggering condition for an alarm. Three Alarm types are available, each one depends on different conditions for triggering the Alarm.

8.7.1 Limit Alarm

A Limit Alarm is triggered when the monitored Tag value goes OUTSIDE of its given boundaries (low limit and high limit).

Note: When the Tag value is equal to its low or high limit, the alarm is not triggered.

8.7.2 Bitmask Alarm

A Bitmask will be given in a hexadecimal format. To get a valid trigger, the Bit wise AND the operation with Bitmask, along with the Tag value corresponding to that Alarm, must yield True (any and all of the corresponding Bits of the Tag value must match with that of the Bitmask). When the Bitmask Alarm is selected, you can specify one or more Bit positions inside the Tag. When one of the Bits is set, the alarm is triggered. The Bit position must be given in hexadecimal format; if more Bits are specified, each position must be separated by a “,".

Note: Bitmask is a position, so it starts from zero (0).

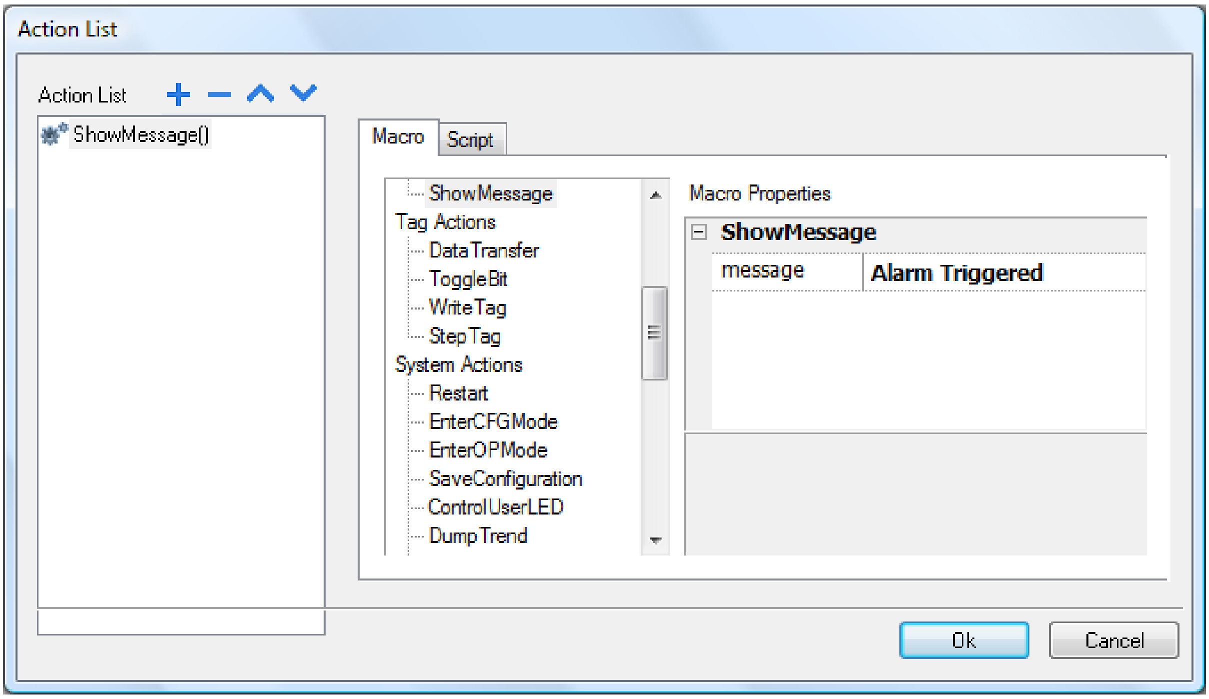
8.7.3 Deviation Alarm

For the Deviation Alarm, a predefined “set point", as well as a value for “deviation" will be given. If the percentage of deviation of the Tag value from the set point exceeds this deviation, then the trigger condition becomes True.

$$(Value_{now} - SetPoint) > \left(\frac{deviation}{100} * SetPoint\right)$$

8.8 Action

Define the action(s) to be executed for the specific Alarm.
Actions are executed by default when the specified trigger condition becomes True. Additional conditions can be specified in the “Events" configuration (in the last column of the Alarm editor, as explained in chapter 8.11.3).



ShowMessage

message	Alarm Triggered

Figure 88

8.9 Foreground and Background Colours

You can modify the Foreground (FG) and Background (BG) colors of the alarm, which will both apply to the Alarm Widget.

8.10 Severity and Priority

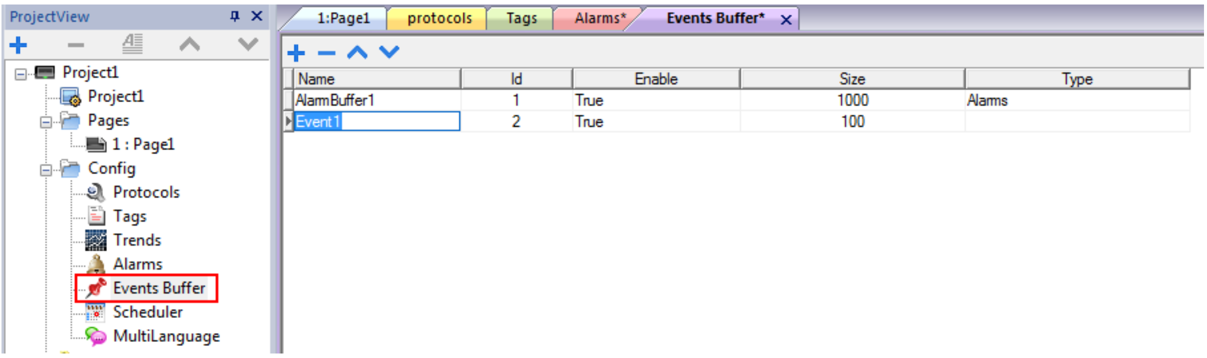
A user can indicate the Severity and Priority of the various Alarms. If multiple Alarms are triggered simultaneously, the actions will be executed based on these Priority and Severity settings.

8.11 Events Types

These options allow you to specify conditions relating to following matters: when the Alarms' events are to be logged, when the Alarms' Widget View is to be refreshed or updated by the system, and some particular options for action execution.

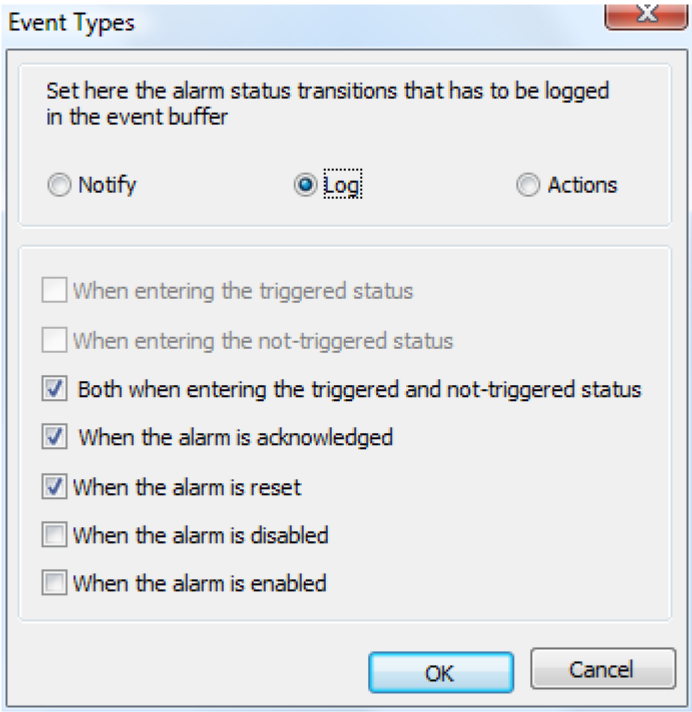
8.11.1 Log Events

The Alarm Events History can be accessed by logging in, in a dedicated buffer called "Event Buffer"; to configure the Event Buffer, double click on "Buffers" in the Configuration Editor (as shown below in Figure 71).



Name	Id	Enable	Size	Type
AlarmBuffer1	1	True	1000	Alarms
Event1	2	True	100	

Figure 89


Event Types

Set here the alarm status transitions that has to be logged in the event buffer

○ Notify ● Log ○ Actions

□ When entering the triggered status

□ When entering the not-triggered status

☑ Both when entering the triggered and not-triggered status

☑ When the alarm is acknowledged

☑ When the alarm is reset

□ When the alarm is disabled

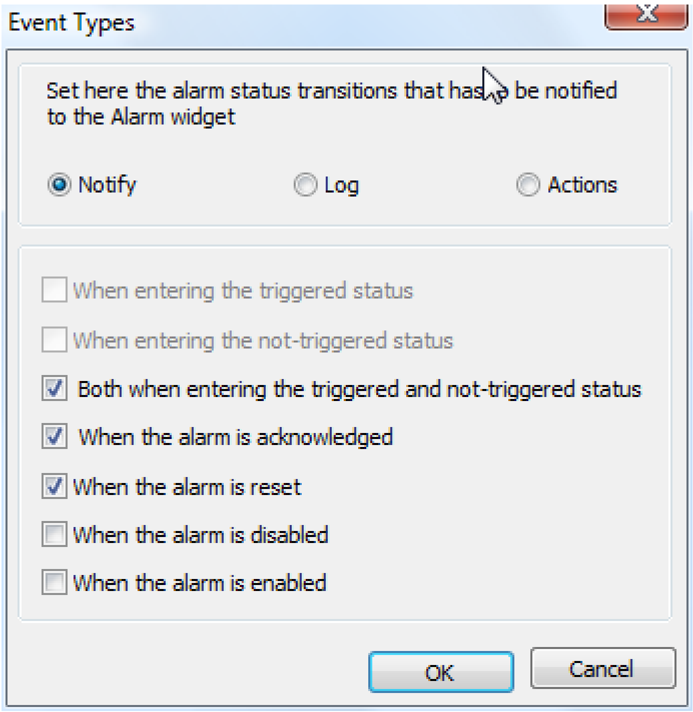
□ When the alarm is enabled

Figure 90

First select the "Log" radio button in the dialog (as shown in Figure 89). The list below this represents a set of conditions in which you may want to store the specific event in the Alarm History Buffer. Click the check boxes corresponding to the application requirements.

8.11.2 Notify

The user can choose the conditions under which the Alarms should be notified in the Alarms Widget. This specifically refers to the default Alarm Widget, available in the Widget gallery. The user can decide when the Widget will be notified of a change to the Alarm Status. We recommend leaving the default settings here, and changing only those necessary for specific application requirements.


Event Types

Set here the alarm status transitions that has to be notified to the Alarm widget

● Notify ○ Log ○ Actions

□ When entering the triggered status

□ When entering the not-triggered status

☑ Both when entering the triggered and not-triggered status

☑ When the alarm is acknowledged

☑ When the alarm is reset

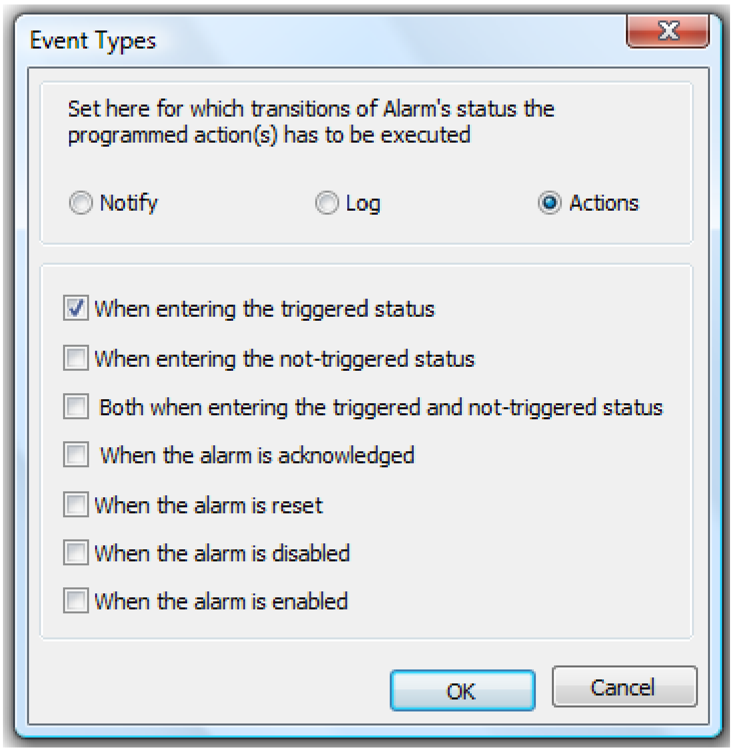
□ When the alarm is disabled

□ When the alarm is enabled

Figure 91

8.11.3 Action Enable

The user can specify the conditions under which the action(s), configured for the specific Alarm, must be executed.



Event Types

Set here for which transitions of Alarm's status the programmed action(s) has to be executed

○ Notify ○ Log ● Actions

☑ When entering the triggered status

□ When entering the not-triggered status

□ Both when entering the triggered and not-triggered status

□ When the alarm is acknowledged

□ When the alarm is reset

□ When the alarm is disabled

□ When the alarm is enabled

Figure 92

By default, the actions are executed only when the Alarm enters the triggering condition; you may change this by configuring the system to execute the configured action also for the other alarms statuses available.

8.12 Configure Alarms Widget

You can insert the Alarms Widget in a page to see the status, acknowledge or reset the alarms. Simply drag and drop the Active Alarms Widget from the Advanced Gallery page.

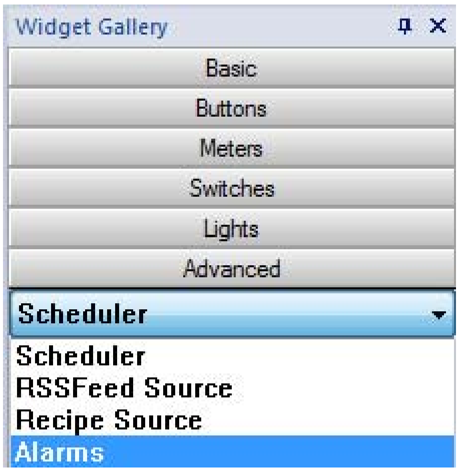


Figure 93

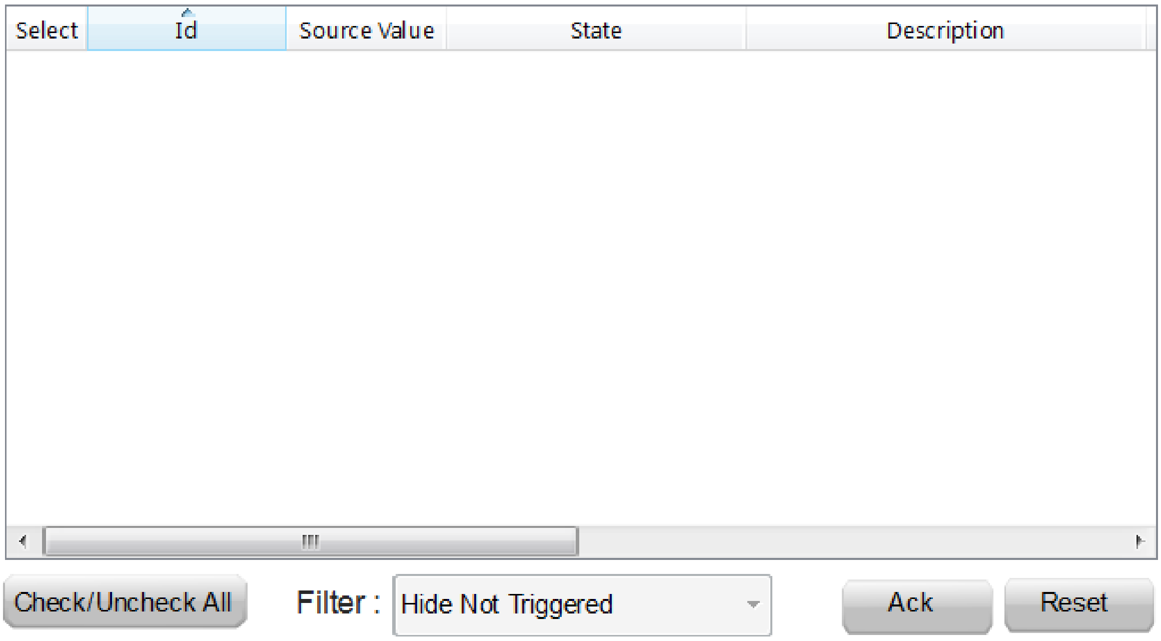


Figure 94

The Alarm Widget will display the Alarms in Runtime.

8.13 Enable / Disable Column Sorting

You can enable or disable the column sorting option, available at Runtime for the Alarms Widget by clicking on the column header. The sorting order is based on the string sorting.

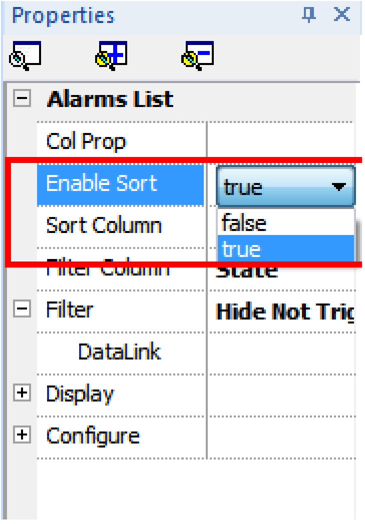
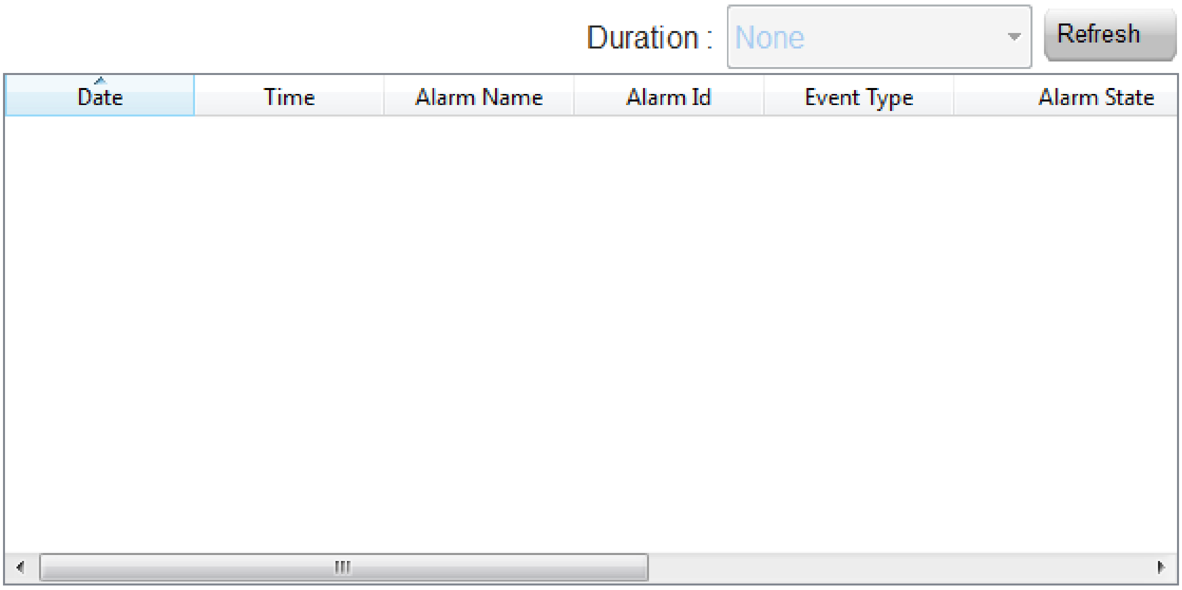


Figure 95

8.14 Configure Alarms History Widget

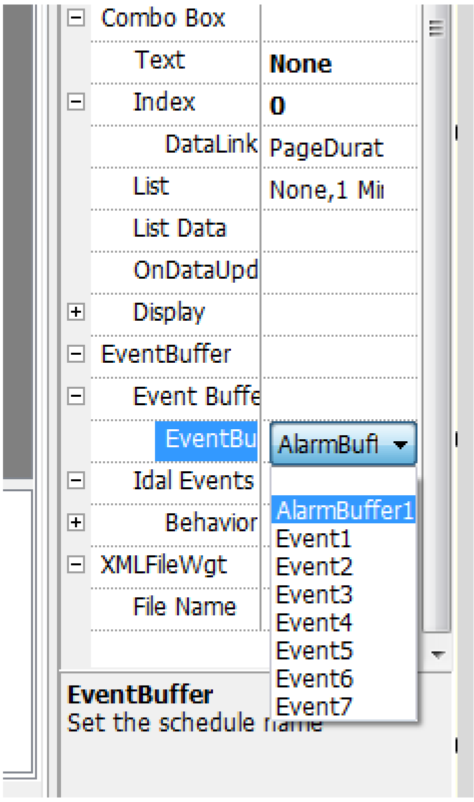
JMobile automatically logs the Alarm list based on the Flag Settings set in the Alarms Editor, under “Log Event Types". To see the Historical Alarm list, you must configure the Alarms History Widget (from the Advanced Gallery page) on a dedicated page.



Duration : None

Figure 96

The selection of the Event Buffer is available in the property panel (as shown in the figure).



Combo Box
Text None
Index 0
DataLink PageDurat
List None,1 Mi
List Data
OnDataUpd
Display
EventBuffer
Event Buffe
EventBu AlarmBufl

EventBuffer
Set the schedule name

Figure 97

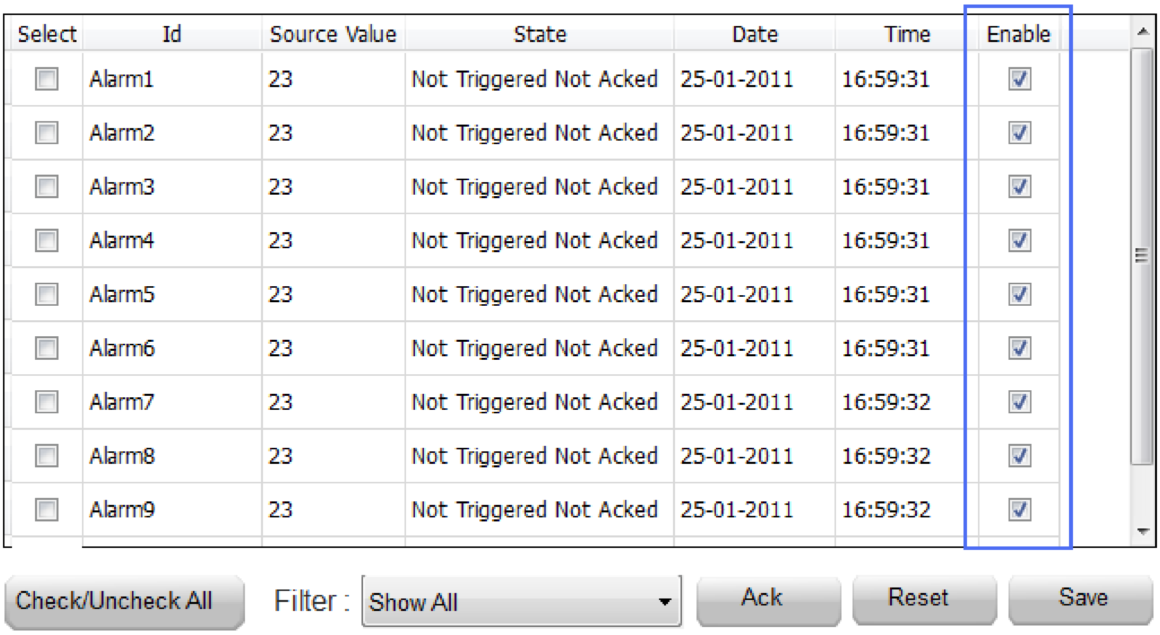
Note: For each of the different Alarm Buffers, there is a specific Event Widget that must be configured for the project; the current version of the Event List Widget does not allow you to switch between buffers.

8.15 Managing Alarms in Run-time.

When an Alarm is triggered, the Alarm will be displayed in the Active Alarms Widget. The Widget allows you to acknowledge and reset the Alarm.
The Alarm display can be filtered by “Hide Not Triggered", “Show All" and other custom filters.
Please note that the visualization of the Alarm Widget is not automatic; if the Widget has been placed at a certain page, then, for each Alarm, you must add a dedicated action that displays on the page when the Alarm Triggering condition becomes True.

8.16 Enable / Disable Alarms in Run-time

You can enable or disable the Alarms in run-time. If you want to disable an alarm, just uncheck the Alarms from the Enable column in the Alarm Widget and execute the “Save" command. This way the alarm will not get triggered and the disabled Alarm is unsubscribed.



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

Filter : Show All

Figure 98

Later, if you want to re-enable the Alarm, select the Alarm and check the Enable check box. Then execute the Save command. The Alarm will now be subscribed and subject to being triggered.

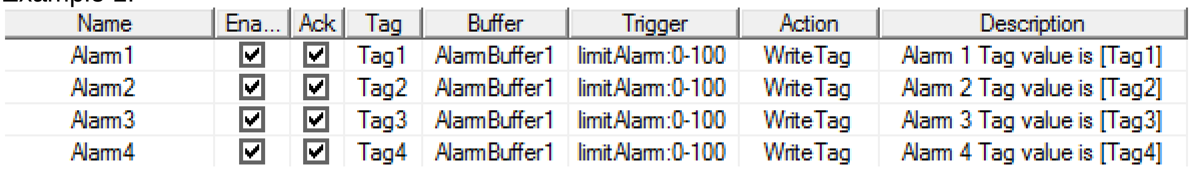
8.17 Live Data in the Active Alarms Widget

This feature is used to view the live Tag data value inside the Alarm Description. It is applicable only for the “Active Alarms" Widget.

The Alarm description string, defined under the Alarm Editor, must contain markers to identify one or more of the Tags. Square brackets are used as markers along with the Tag name. See Figure 99 and Figure 100.

In the Active Alarms Widget, you can view the live Tag values in the description column. The Widget automatically refreshes and shows the current values of the Tags in the Widget.
To configure the live data visualisation in the Alarm Widget, follow a simple syntax rule. The Tags to be included must be specified in the description, including the Tag names in square brackets: start with “[“, and end with “]" as well. An example is shown below.

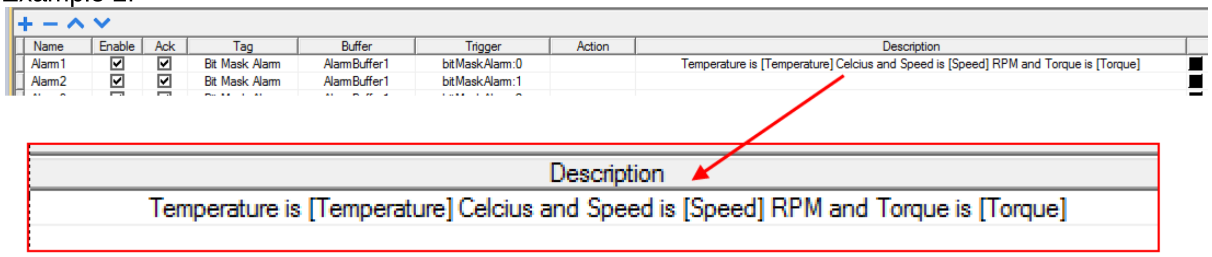
Example 1:



Name	Ena...	Ack	Tag	Buffer	Trigger	Action	Description
Alarm1	☑	☑	Tag1	AlarmBuffer1	limitAlarm:0-100	WriteTag	Alarm 1 Tag value is [Tag1]
Alarm2	☑	☑	Tag2	AlarmBuffer1	limitAlarm:0-100	WriteTag	Alarm 2 Tag value is [Tag2]
Alarm3	☑	☑	Tag3	AlarmBuffer1	limitAlarm:0-100	WriteTag	Alarm 3 Tag value is [Tag3]
Alarm4	☑	☑	Tag4	AlarmBuffer1	limitAlarm:0-100	WriteTag	Alarm 4 Tag value is [Tag4]

Figure 99

Example 2:



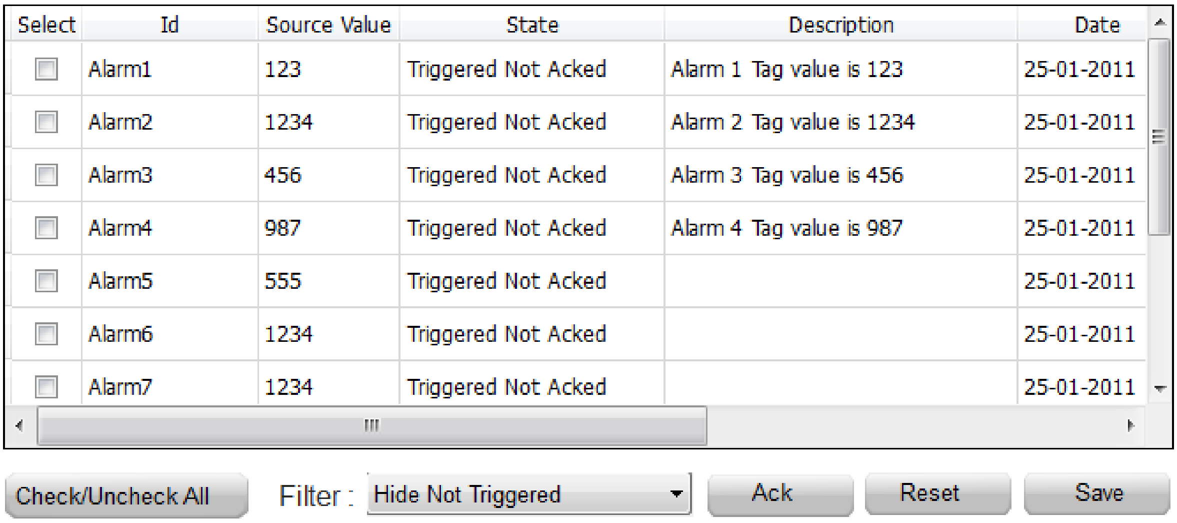
Name	Enable	Ack	Tag	Buffer	Trigger	Action	Description
Alarm1	☑	☑	Bit Mask Alarm	AlarmBuffer1	bitMaskAlarm:0		Temperature is [Temperature] Celcius and Speed is [Speed] RPM and Torque is [Torque]
Alarm2	☑	☑	Bit Mask Alarm	AlarmBuffer1	bitMaskAlarm:1		

Description
Temperature is [Temperature] Celcius and Speed is [Speed] RPM and Torque is [Torque]

Figure 100

On Runtime, the markers and Tag name will be replaced by the actual value of the Tag. Output will be displayed as shown in Figure 101 and Figure 102.

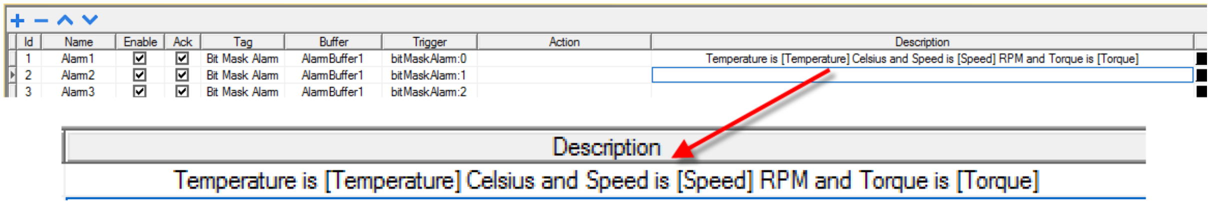
Example 1:



Select	Id	Source Value	State	Description	Date
□	Alarm1	123	Triggered Not Acked	Alarm 1 Tag value is 123	25-01-2011
□	Alarm2	1234	Triggered Not Acked	Alarm 2 Tag value is 1234	25-01-2011
□	Alarm3	456	Triggered Not Acked	Alarm 3 Tag value is 456	25-01-2011
□	Alarm4	987	Triggered Not Acked	Alarm 4 Tag value is 987	25-01-2011
□	Alarm5	555	Triggered Not Acked		25-01-2011
□	Alarm6	1234	Triggered Not Acked		25-01-2011
□	Alarm7	1234	Triggered Not Acked		25-01-2011

Figure 101

Example 2:



Id	Name	Enable	Ack	Tag	Buffer	Trigger	Action	Description
1	Alarm1	☑	☑	Bit Mask Alarm	AlarmBuffer1	bitMaskAlarm:0		Temperature is [Temperature] Celsius and Speed is [Speed] RPM and Torque is [Torque]
2	Alarm2	☑	☑	Bit Mask Alarm	AlarmBuffer1	bitMaskAlarm:1		
3	Alarm3	☑	☑	Bit Mask Alarm	AlarmBuffer1	bitMaskAlarm:2		

Description
Temperature is [Temperature] Celsius and Speed is [Speed] RPM and Torque is [Torque]

Figure 102

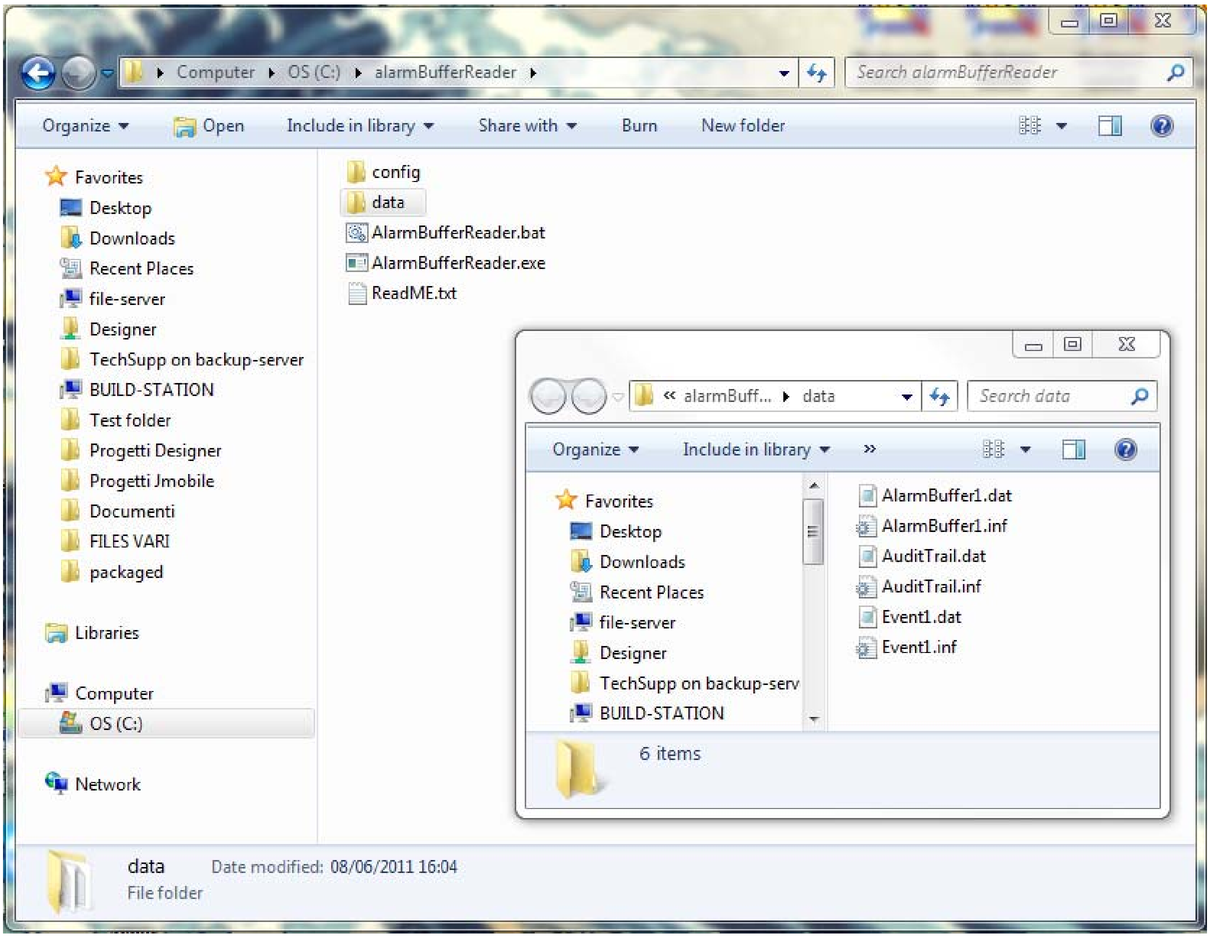
The Alarm Widget will be refreshed periodically, to maintain the Tag value updated.

Note: the system supports up to 5 tags as live data per each alarm's description

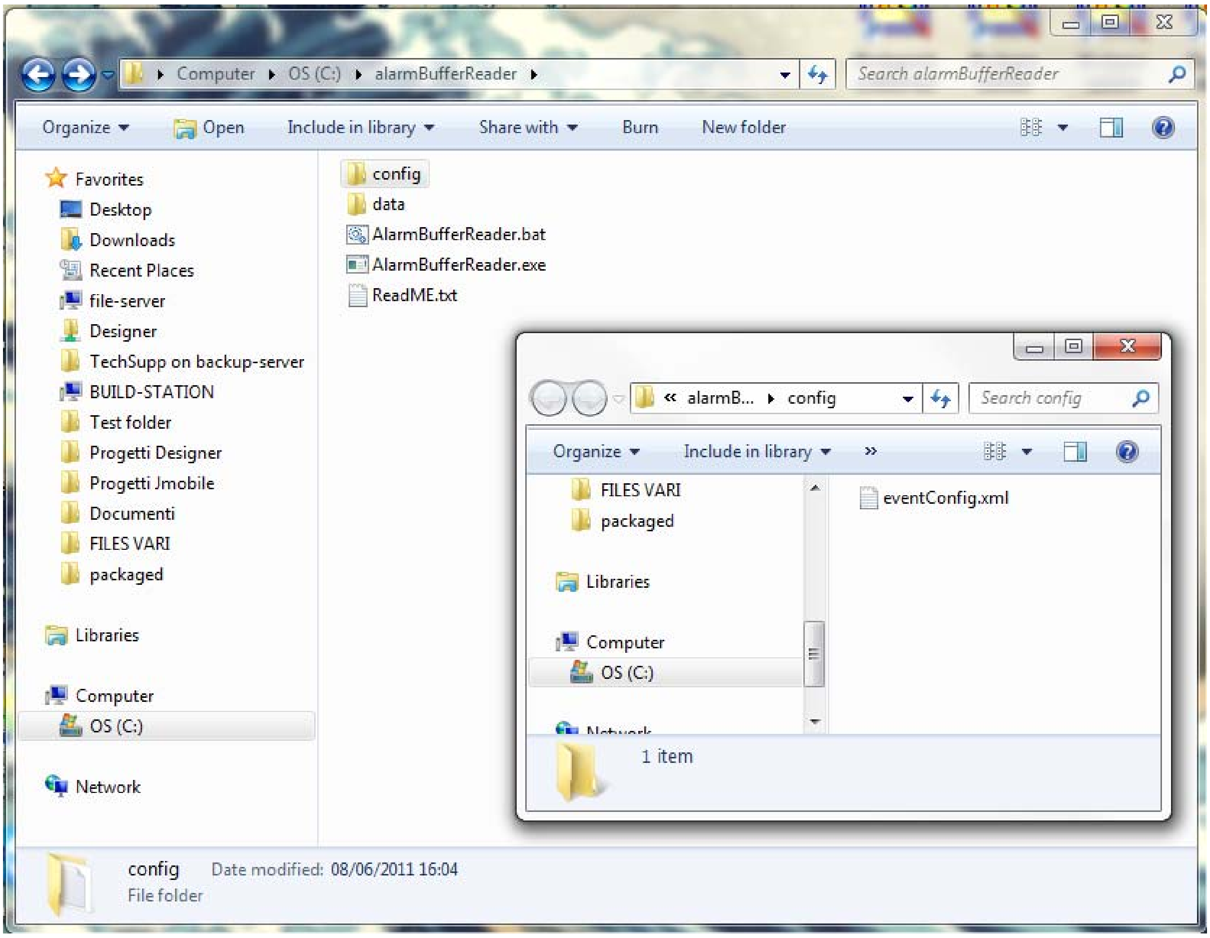
8.18 Exporting Alarm buffers as CSV file

You can convert the Alarm buffer data to a “.csv" file, the utility needed is called AlarmBufferReader available in the ..\Exor\JMobile Suite\utils folder.
To convert the buffer data, follow these steps.

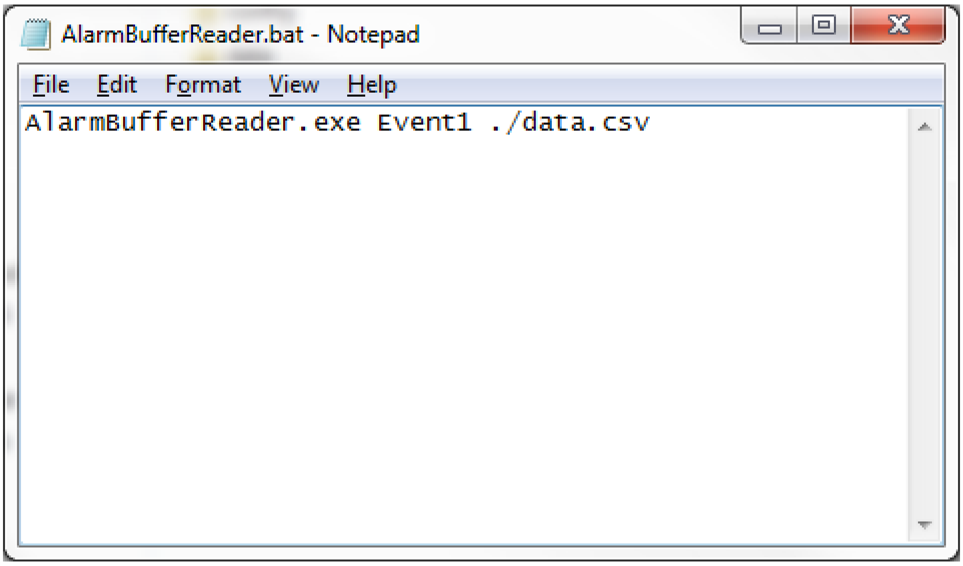
Take the .DAT and .INF dumped files and copy them into the DATA folder that you will find into the utility folder as shown in the figure below.



Take eventConfig.xml file and copy it into the CONFIG folder that you will find into the utility folder as shown in the figure below.



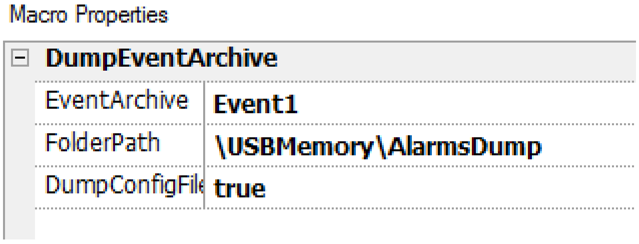
Now edit the AlarmBufferReader.bat file, you can do it with a simple text editor like Notepad,



```
AlarmBufferReader.exe Event1 ./data.csv
```

The string into the .bat file is made of three parts, each one separated by the next one with a space:

AlarmBufferReader.exe - This is the name of the program file to be executed.
Event1 - This is the name of the archive that will be Exported in CSV, the name is reported into the Properties of the DumpEventArchive macrosee the figure below,



Macro Properties

DumpEventArchive	
EventArchive	Event1
FolderPath	\USBMemory\AlarmsDump
DumpConfigFil	true

./data.csv – This is the name of the CSV file that will be created by the utility. In this example the file created will be named data.csv

Save the changes and close the editor, double click then on the bat file to run it, the utility will be executed and the file created.

9 Working with Recipes

Data can be stored on the flash disk of the panel device, in the format of a data file. This data can be written to the controller, and, conversely, the data can be read from the controller and saved back on the panel storage media. This concept is normally referred as "Recipes", and it offers you powerful way to extend the capabilities of the controller. This is especially true for controllers that have a limited amount of internal memory.

The Recipe memory is the physical storage (the flash disk) for the Recipes. The “Recipe Tag" block basically identifies the “current Recipe": from the Recipe memory, you take one Recipe data record and designate it “current/active Recipe". Then, you operate transfers over this, to or from the controller. These Tags can be displayed on the page.

At the moment the Recipe data is configured in Studio, the user can specify default values for each element of the data records. On Runtime, data can be edited and saved back on a new data file, separate from the original one containing the default values. The use of a separate data file on 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 on Runtime.

You can configure Recipes by adding the required controller data items to a page in Recipe Widget. A Recipe can be associated with a particular page and is composed of all the Recipe data items on that page. Recipe data items contain all the information associated with normal controller data items; but, rather than the data being read and written directly to the controller during the course of normal operation, the data is instead read from and written to the panel memory that is reserved for the data item.

This chapter describes how to configure and use the Recipes in JMobile Studio/server application.

9.1 Recipe Configuration Editor.

In the Project View pane, select Recipes and right click. Then choose Insert Recipe if you want to create a new Recipe set. Then newly added Recipe item will be added in the project workspace.

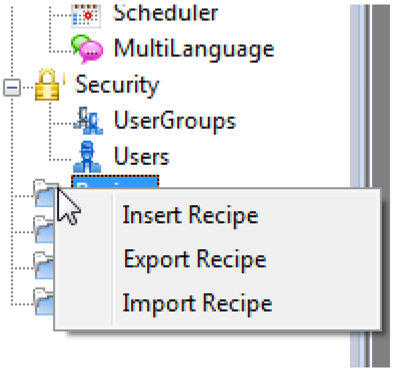


Figure 103

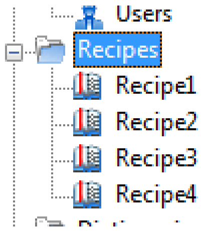
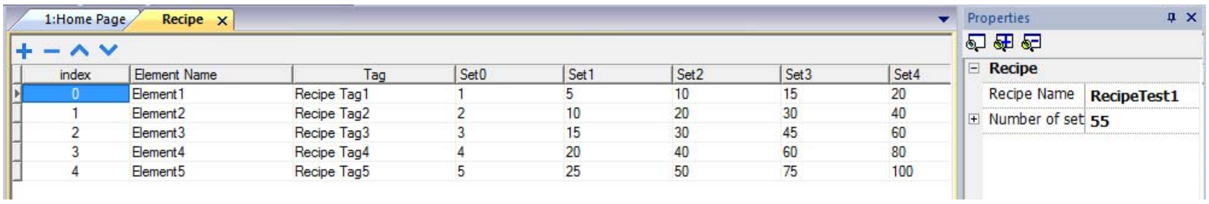


Figure 104

Double Click on the Recipes to open the Recipe Editor, as shown in Figure 105.
Add the Recipe Elements by clicking the “+" button, and then link the Tags to the Recipe element.



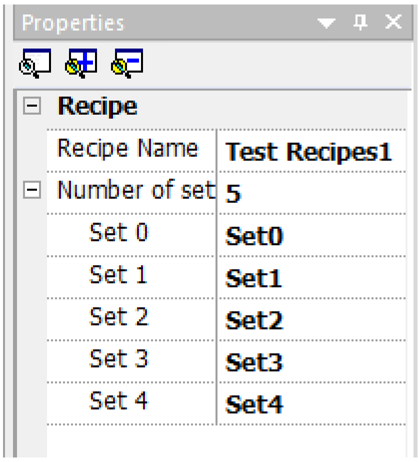
index	Element Name	Tag	Set0	Set1	Set2	Set3	Set4
0	Element1	Recipe Tag1	1	5	10	15	20
1	Element2	Recipe Tag2	2	10	20	30	40
2	Element3	Recipe Tag3	3	15	30	45	60
3	Element4	Recipe Tag4	4	20	40	60	80
4	Element5	Recipe Tag5	5	25	50	75	100

Recipe

Recipe Name	RecipeTest1
Number of set	55

Figure 105

9.2 Configuring Recipes Set on the Page



Recipe

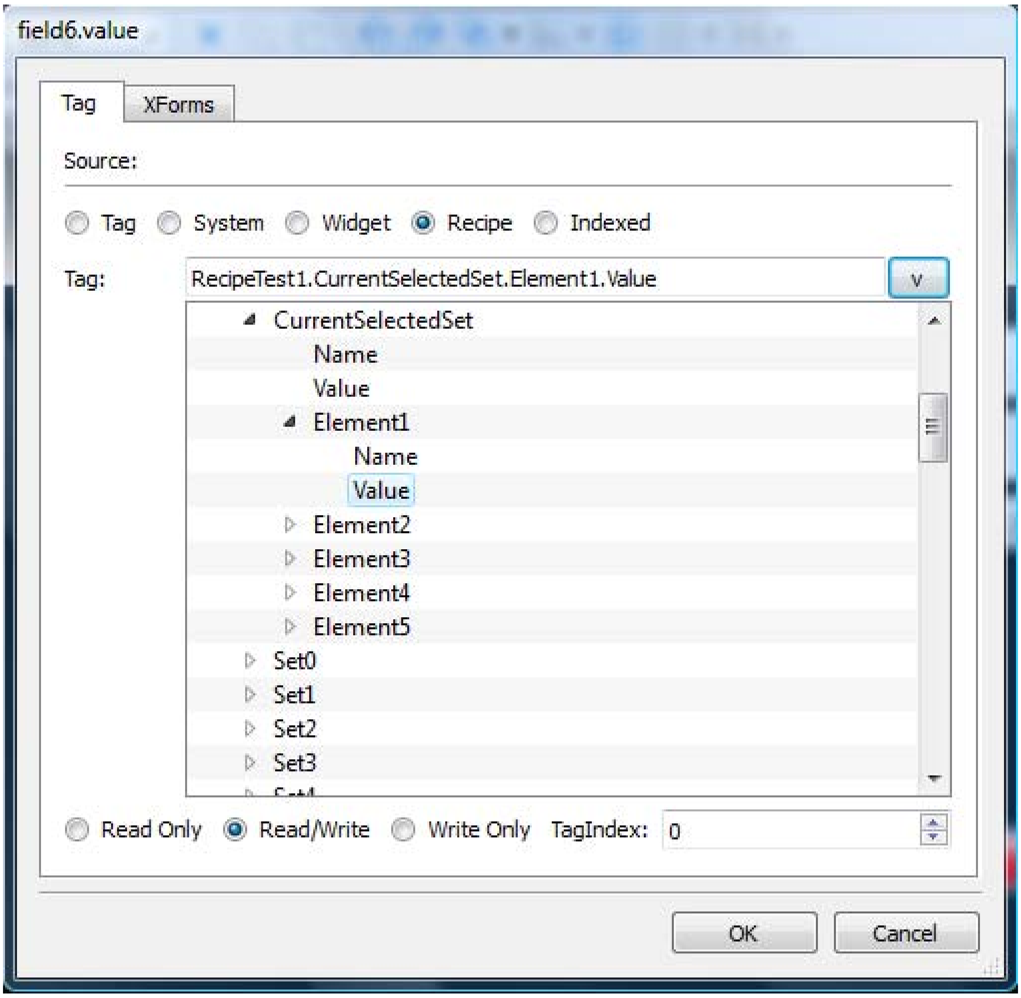
Recipe Name	Test Recipes1
Number of set	5
Set 0	Set0
Set 1	Set1
Set 2	Set2
Set 3	Set3
Set 4	Set4

Figure 106

The number of parameter sets can be changed in the Number of sets field in the property pane. Recipe values for all the parameter sets can be entered into the Recipe configuration window. You can rename the Recipe set in the property pane as shown in Figure 104.

9.3 Defining Recipe Fields

The user can define the Recipe field on the page by using the numeric field Widget from the gallery and attach the Tags from the Recipe data source. Figure 107 shows an example of a Tag attached to a Recipe field.



field6.value

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

Tag: RecipeTest1.CurrentSelectedSet.Element1.Value

○ Read Only ● Read/Write ○ Write Only TagIndex: 0

Figure 107

The Attach to Dialog allows you to attach to the numeric field with all the different Recipe variables, such as:

Current Recipe ->Current selected Recipe set-> Element -> value (or) name,
Selected Recipe -> Selected Set0 -> Element -> Value (or) Name
Selected Recipe list
Currently selected Recipe list
Recipe Status
Among others…..

When the numeric files are defined of a Read/Write type property, the default Recipe data can be edited on Runtime. As explained in the Introduction, these new values are stored in a separate file as modified Recipe data.

9.4 Recipe Status

After every Recipe Upload or Download, or Recipe set modification, the Recipe status parameters display a value.
The following are the values and conditions for the Recipe status system variable.

0 - Set modified - current selected set changed.
1 - Download triggered - triggered a download request

2 - Download Done - download action completed.
3 - Download error - error occurred when doing download - errors like unknown set, unknown Recipe, controller not ready, Tags write failed etc.
4 - Upload triggered - triggered an upload request
5 - Upload done - upload action completed
6 - Upload Error - error occurred when doing upload - errors similar to download errors.
7 - General Error - errors like connection lost.

9.5 Configuring Recipe Widget for Runtime Execution

Two default Recipe Widgets are available in the advanced Widget Gallery category.
The “Recipe Set" Widget allows you to select a Recipe set for the upload and download operation. If you have more than one Recipe in the project, then the “Recipe Menu" Widget can be directly used to manage all the Recipes in a single Widget, listing Recipes and selecting the sets for each Recipe.

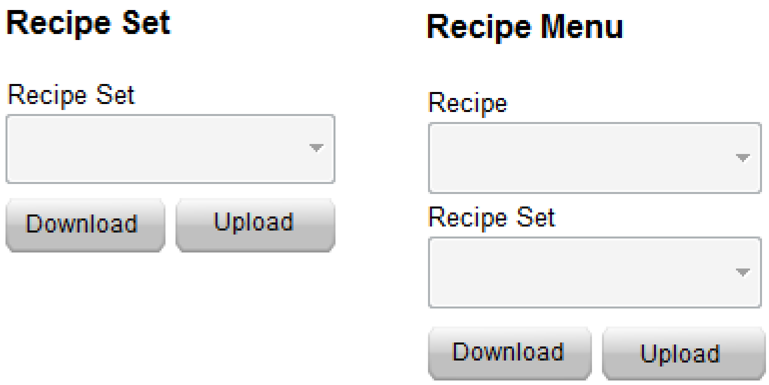


Figure 108

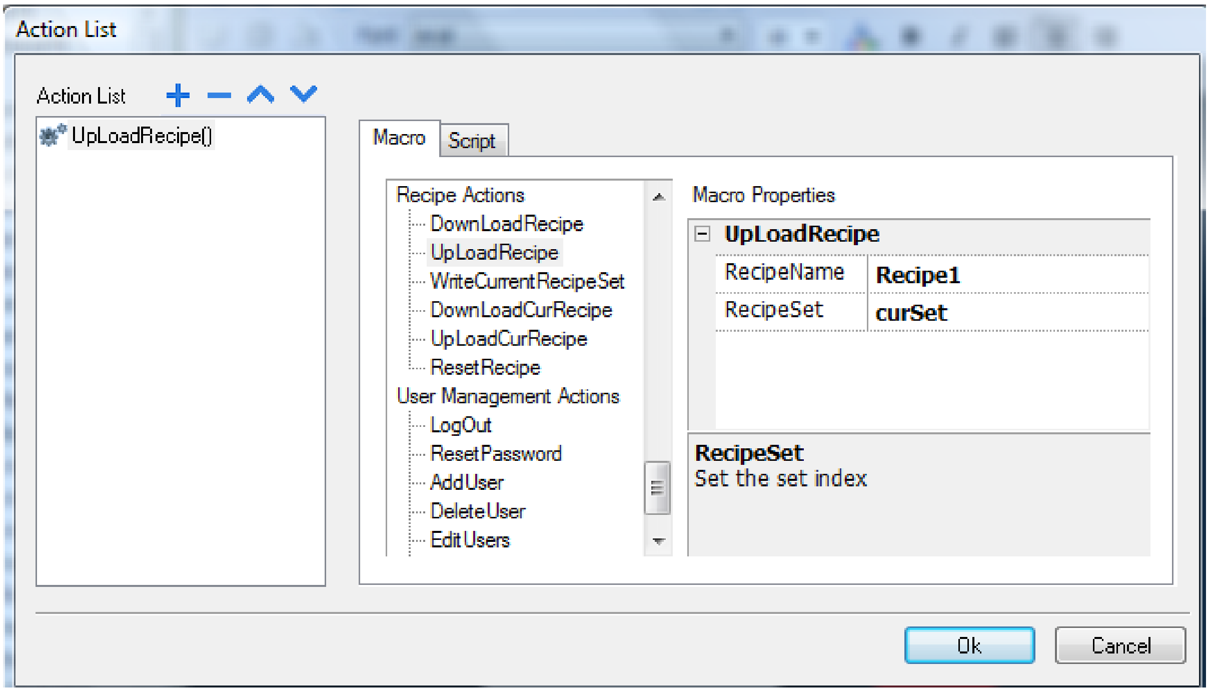
9.6 Configure Recipe Transfer Macros.

The Recipe transfer action can be completed through the action list dialog. The transfer of Recipes can be achieved by any of the following methods:

Using a click action on a button or switch.

Configuring the action from Alarms’ action list.

Using the Schedulers and timers Event actions’ list.



UpLoadRecipe

RecipeName: Recipe1

RecipeSet: curSet

RecipeSet
Set the set index

Figure 109

9.6.1 Download Recipe

The “Download Recipe" action allows you to transfer the selected set of Recipe data from the panel memory to the live Tag data in the controller memory. To do this, the user has to configure the Download Recipe action, select the Recipe name (in the action property) and set the Recipe set you want to download. If you select “CurSet", the currently selected Recipe set will be downloaded.

9.6.2 Upload Recipe

The “Upload Recipe" action allows you to transfer the set of live Tag data from the controller memory to the panel Recipe memory; the Recipe data will be replaced with the new values. To do this, the user must configure the Upload Recipe macro, and, in the macro property, select the Recipe name and Recipe set you want to upload to. If you select “CurSet", the uploaded data will be stored in the currently selected Recipe set.

9.6.3 Download Current Recipe

The DownLoadCurRecipe action allows you to transfer the current Recipe set data from the HMI memory to the live Tag data in the controller. To do this, the user must configure the DownLoadCurRecipe macro. It is not necessary to define any parameters in the macro properties. The “current" Recipe is the one selected by the Recipe Menu Widget.

9.6.4 Upload Current Recipe

The UpLoadCurRecipe action allows you to transfer the set of live Tag data from the controller to the current Recipe memory, in the HMI. The Recipe data will be replaced with the new values. To do this, the user must configure the UpLoadCurRecipe macro. It is not necessary to define any parameters in the macro properties. The “current" Recipe is the one selected by the Recipe Menu Widget.

9.6.5 Reset Recipe

The ResetRecipe macro allows you to restore Recipes to their factory settings, that is the Recipe configuration originally set for the project. By executing the macro, the factory Recipe settings will be set back to the default values originally specified at the time of programming, thereby erasing the modified Recipe data file.

9.7 Upload or Download of Recipes During Run-time

9.7.1 Recipe Download Through Recipe Widget in Run-time

Drag and drop the Recipe Widget (as described in Chapter 8.5) into the project to execute the Recipe transfer in run-time. Select the Recipe from the drop down box, and select the Recipe set from the set dropdown list. Then press the “Download" button to download the current selected Recipe set, or press the “Upload" button to upload the current selected Recipe set.

9.7.2 Recipe Download or Upload Through Recipe Transfer Macro in Run-time

The Recipes can be Downloaded or Uploaded through the Recipe transfer macro. Define the macro button as described in chapter 9.6. In run-time, execute the macro (if the macro is programmed with a push button, then press the button). The Recipes data will then be transferred to the controller, or uploaded from the controller, depending on the action programmed. Figure 110 shows an example of the Recipe project.

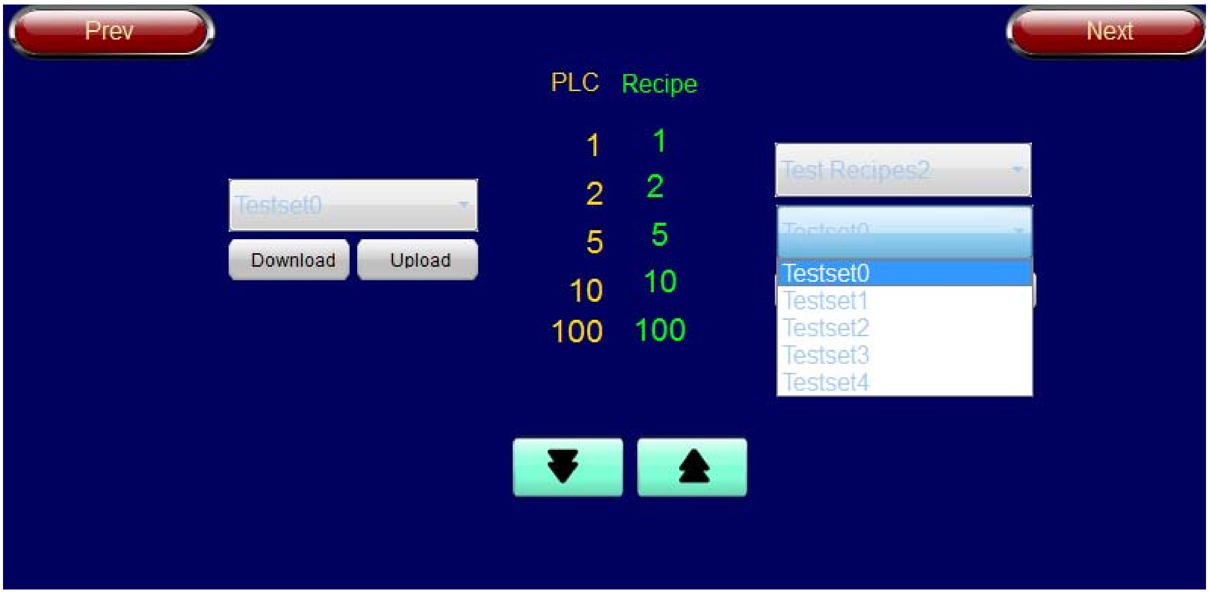


Figure 110

10 Working with Trends in JMobile

Trending means to sample and record values of a specified Tag according to sampling conditions (normally, the time).
Trending is divided into two main parts: Trend acquisition and Trend viewing. Trend acquisition (Trend Editor) collects the data into a database. The Trend viewer (Trend Widget) displays the data from this database in a graphical format.

10.1 Types of Trends Logging

In JMobile we have two types of Trend mechanisms: Real-Time data Trending and Historical data Trending.

10.1.1 Real-Time Trend

In real time Trend, the data will be presented directly in the Trend window, and the changes to the live data can be seen directly in the format of a curve on the Trend window. Users can manage the process by seeing the Trend on the HMI. The real time Trend Widget is just a viewer for a Tag, and it does not refer to any saved data in any buffer. Any curve plotted is lost when the page containing the Widget is changed.

10.2 History Trend

If you want to analyze the data in the future, you will need to store the Trend data somewhere for later review. For this reason, we have introduced the History Trend. When you select History Trend, you can store date information with reference to time.

10.3 Configuring RealTime Trend

To configure the RealTime Trend, just drag and drop the RealTime Trend Widget from the Gallery.

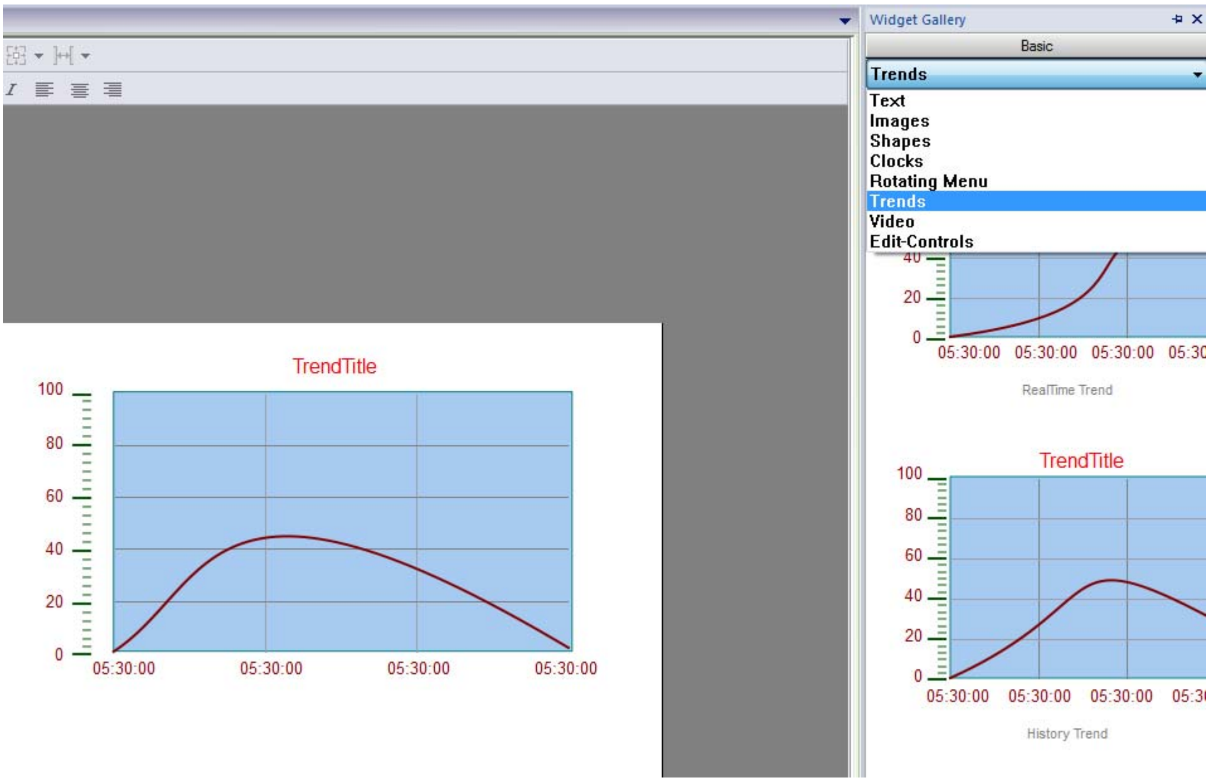
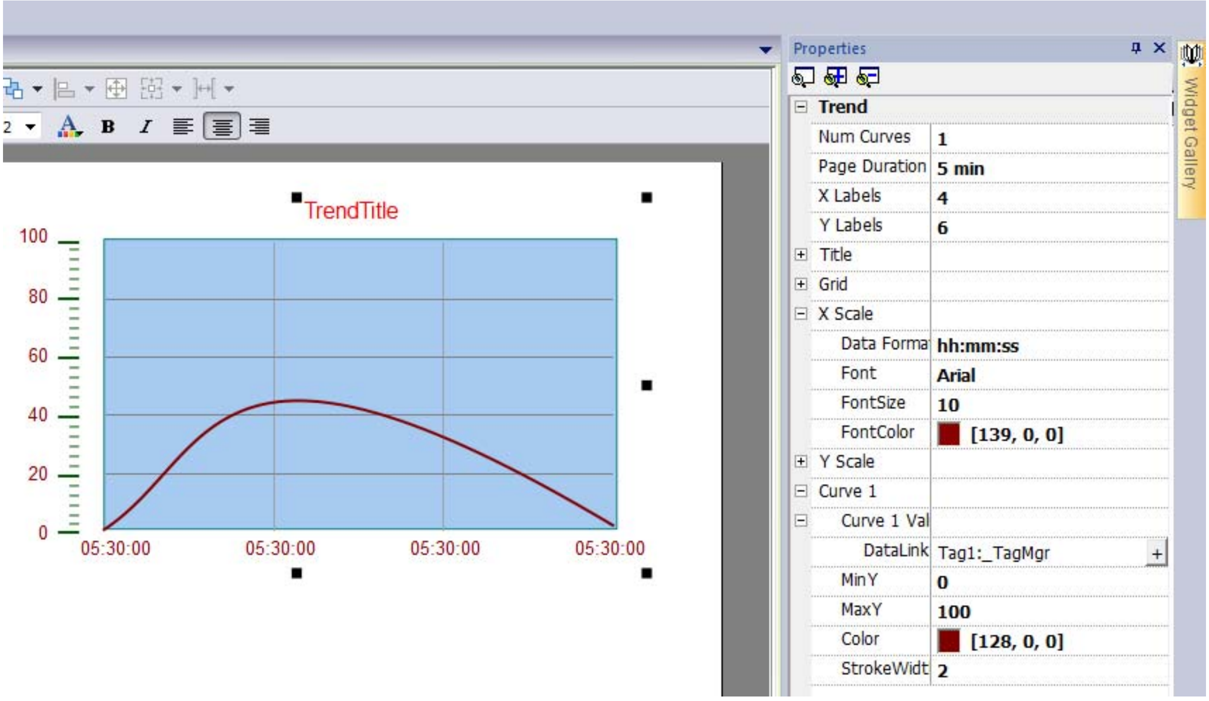


Figure 111



Trend	
Num Curves	1
Page Duration	5 min
X Labels	4
Y Labels	6
Title	
Grid	
X Scale	
Data Forma	hh:mm:ss
Font	Arial
FontSize	10
FontColor	[139, 0, 0]
Y Scale	
Curve 1	
Curve 1 Val	
DataLink	Tag1:_TagMgr
MinY	0
MaxY	100
Color	[128, 0, 0]
StrokeWidt	2

Figure 112

Select the Trend Widget and, in the properties pane, attach to the “Curve Value" property of the Tag for which you want the data to be plotted against time.

The Number of Curves property allows you to configure the number of Live Trend curves in the Trend window. A maximum of 5 Curves can be configured in a Trend window.

The page duration property sets the time range of the X-Axis. However, you can dynamically change the page duration in Run Time with the Combo Widgets, thereby attaching to the Trend window page duration properties.

X Labels property decides the number of Labels in the X-axis scale.

Y Labels property decides the number of Labels in the Y-axis scale.

A title property allows you to modify the Trend title and font properties (like font size, label, etc.)

The A curve property allows you to set the Minimum or Maximum of the curves. You can also attach a Tag to these minimum and maximum properties. This enhances the possibility to change the min and max dynamically in the run-time.

Also you can modify the properties, such as colors, update time, number of samples, etc. of the Trend curves through the properties view.

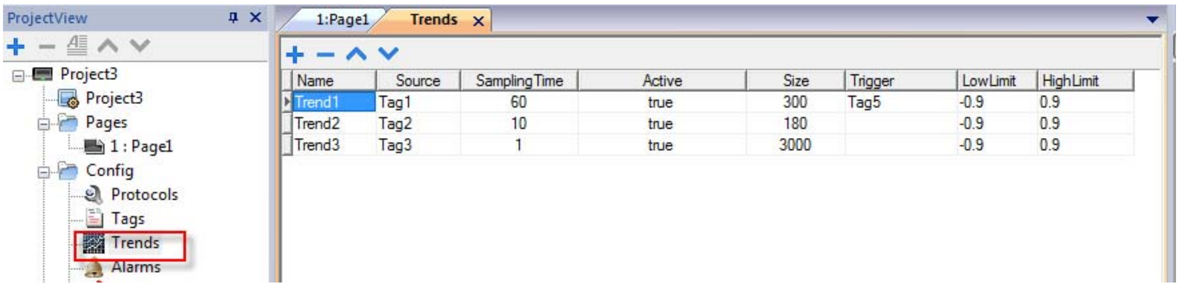
Scaling allows for a linear transformation of the data being sampled. To apply scaling, use the X Forms attached to dialog.

10.4 Configuring History Trend

The first step in creating a History Trend is to create a Trend Buffer.
Historic Trend display: Sets a fixed sampling time or triggers, from controller to specified registers' read value, and saves to the HMI data logger. After a long time constantly sampling, the HMI will transform the sampling values to display on the screen.

10.4.1 Trend Editor

Historical Trends require a proper configuration of Trend data buffer. Trends' buffers are configured from the Trend editor.
The Trend Buffers are stored on the flash disk in the format of a data file.



Name	Source	SamplingTime	Active	Size	Trigger	LowLimit	HighLimit
Trend1	Tag1	60	true	300	Tag5	-0.9	0.9
Trend2	Tag2	10	true	180		-0.9	0.9
Trend3	Tag3	1	true	3000		-0.9	0.9

Figure 113

In the Project View pane, double click Trends to open the Trend editor. Then add the Trend Buffer, by selecting the “+" button on the editor.

The following are the properties for the Trend editor:

Name
The name is the Trend Buffer name, which will appear when you define the buffer to a Trend window property pane.

Source Tag
This combo list allows to select the Tag which is sampled by the Trend manager system.

Sampling Time
Samples are collected and stored in the disk data file on a cyclical basis, or based on a trigger condition. Default sampling condition is the time; the Sampling Time simply specifies the sampling period in seconds.

Active
Specifies is the Trend runs by default when system starts up. This property has always to be activated.

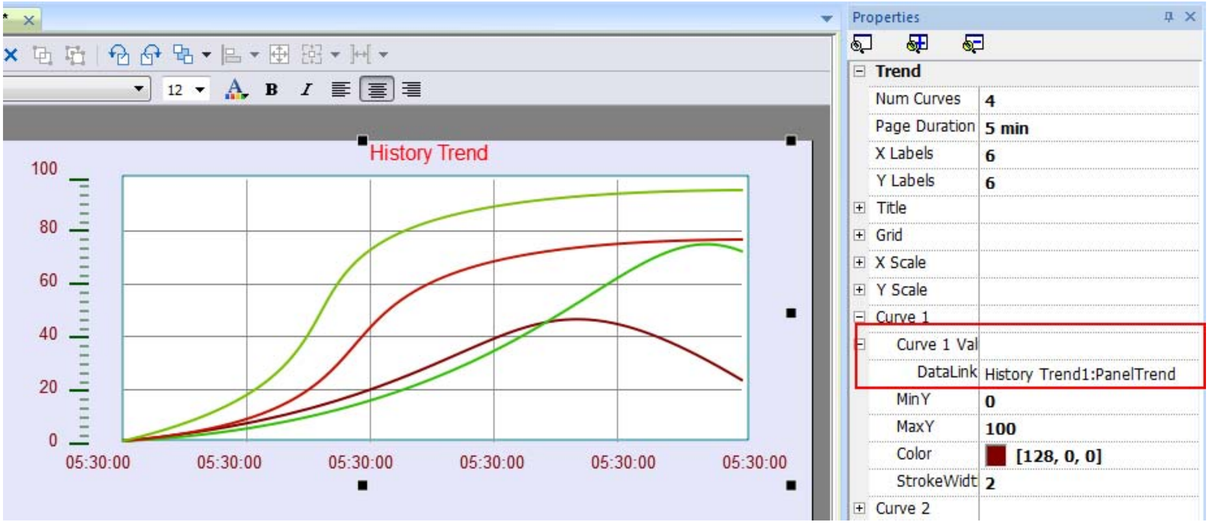
Size
This represents the buffer size expressed in samples.
The Trend data are stored in a file organized as a first-in first-out (FIFO) buffer. Once the buffer is full, older sample values will be erased to allow the new sample values to be stored.
The stored data in the disk can be erased at any time by using the proper DeleteTrend macro command.

Trigger Tag
When the Trigger Tag is specified, the source Tag is not sampled on a cyclical basis but rather on the Trigger Tag value change. In any case, the samples are plotted with respect to the time. The Trigger Tag and source can be the same.
A Trend condition is triggered (source Tag sampled) when the change in the value of the Trigger Tag (compared with previous value) goes out of given boundaries (Low Limit and High Limit).

Low Limit and High Limit
Low and High Limits specify the boundaries. When the triggering condition is the time, a new sample is considered significant (and then stored) only if its value, in comparison with the last saved value, goes out from the specified boundaries.
In case the triggering condition is an additional Tag value change, the boundaries are applied to the Trigger Tag.

10.5 Configuring Trend Window for History Trends

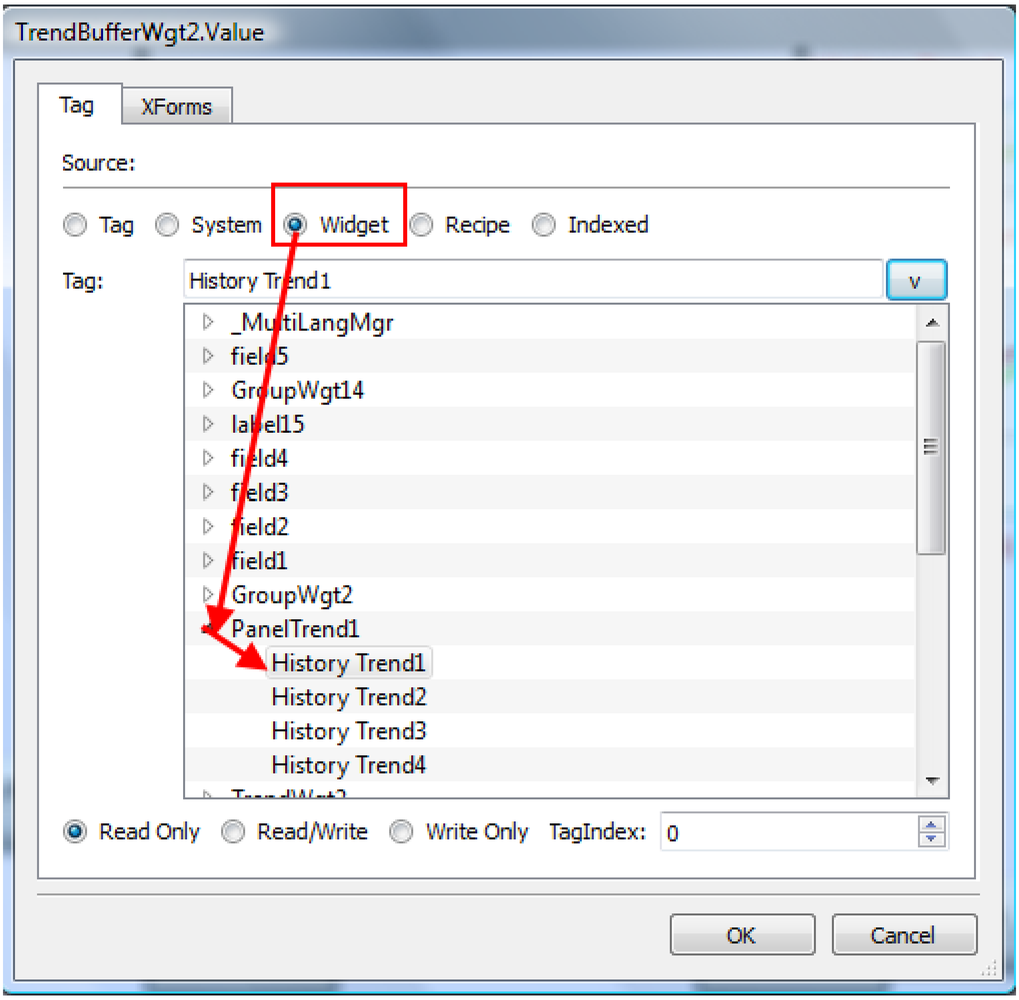
The History Trend Widget (Trend window) is the area used to display the Trend Buffer in a curve format. After configuring the Trend Buffer in Trend editor, you can use the Historical Trend viewer widget to plot the Trend curve on the screen. From the Trend Gallery, drag and drop the “History Trend" Widget to the page.



Trend	
Num Curves	4
Page Duration	5 min
X Labels	6
Y Labels	6
Title	
Grid	
X Scale	
Y Scale	
Curve 1	
Curve 1 Val	
DataLink	History Trend1:PanelTrend
MinY	0
MaxY	100
Color	🟥 [128, 0, 0]
StrokeWidt	2
Curve 2	

Figure 114

Then, in the property pane of the Trend window, attach the History Trend Buffer (as shown in Figure 115).


TrendBufferWgt2.Value

Source:

○ Tag ○ System ● Widget ○ Recipe ○ Indexed

Tag: History Trend1

- _MultiLangMgr
- field5
- GroupWgt14
- label15
- field4
- field3
- field2
- field1
- GroupWgt2
- PanelTrend1
  - History Trend1
  - History Trend2
  - History Trend3
  - History Trend4

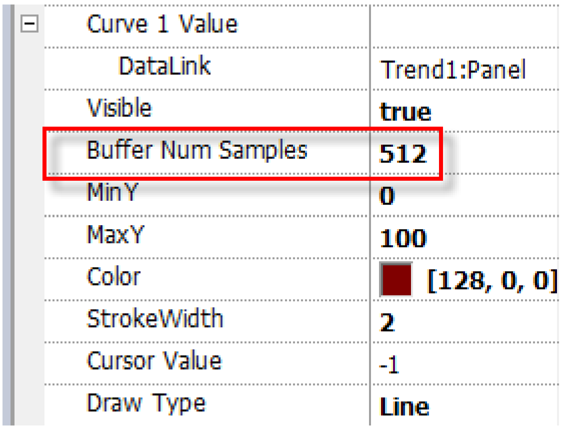
● Read Only ○ Read/Write ○ Write Only TagIndex: 0

Figure 115

10.6 Properties for Trend Window (Advanced View)

With the help of the property pane of the Trend window, you can customize the Trend window properties, such as, X Axis time, Y Axis value, number of Trend curves, changes to the labels, grids, and number of samples, etc.

In the Curve category there is one property called Buffer Num Samples as showed in Figure 116.



⊟ Curve 1 Value	
DataLink	Trend1:Panel
Visible	true
Buffer Num Samples	512
MinY	0
MaxY	100
Color	■ [128, 0, 0]
StrokeWidth	2
Cursor Value	-1
Draw Type	Line

Figure 116

This property represents the maximum numbers of samples plotted by the widget and must be set according to the buffer size.

10.7 Trend Cursor

The Trend Cursor allows you to see the Trend value at a point. Use Show Trend Cursor macro and Scroll Trend Cursor macro to enable the Trend cursor and move it to the required point.

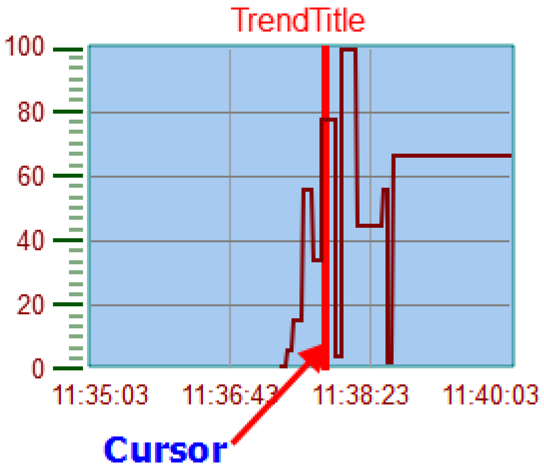
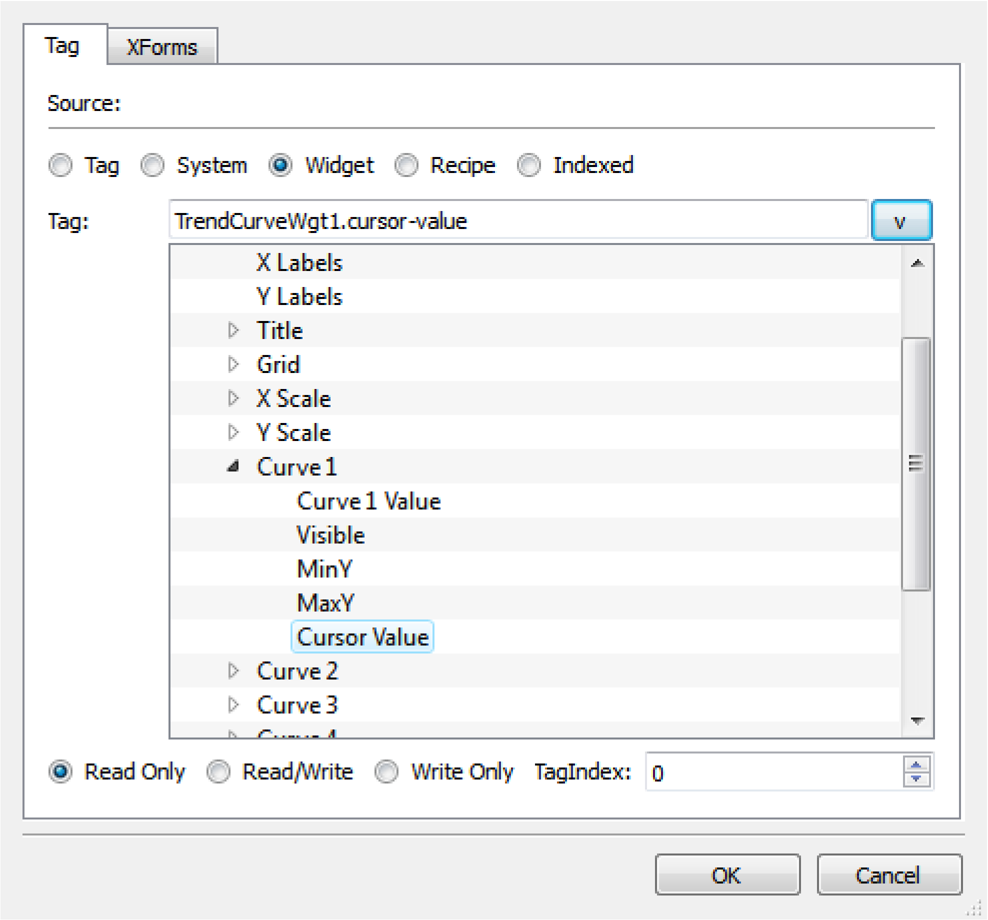


Figure 117

To display the value of the Trend Cursor on the page, define a numeric field and attach the Cursor Value Widget Tag (as shown in Figure 118).



Tag | XForms

Source:

○ Tag ○ System ● Widget ○ Recipe ○ Indexed

Tag: TrendCurveWgt1.cursor-value

● Read Only ○ Read/Write ○ Write Only TagIndex: 0

Figure 118

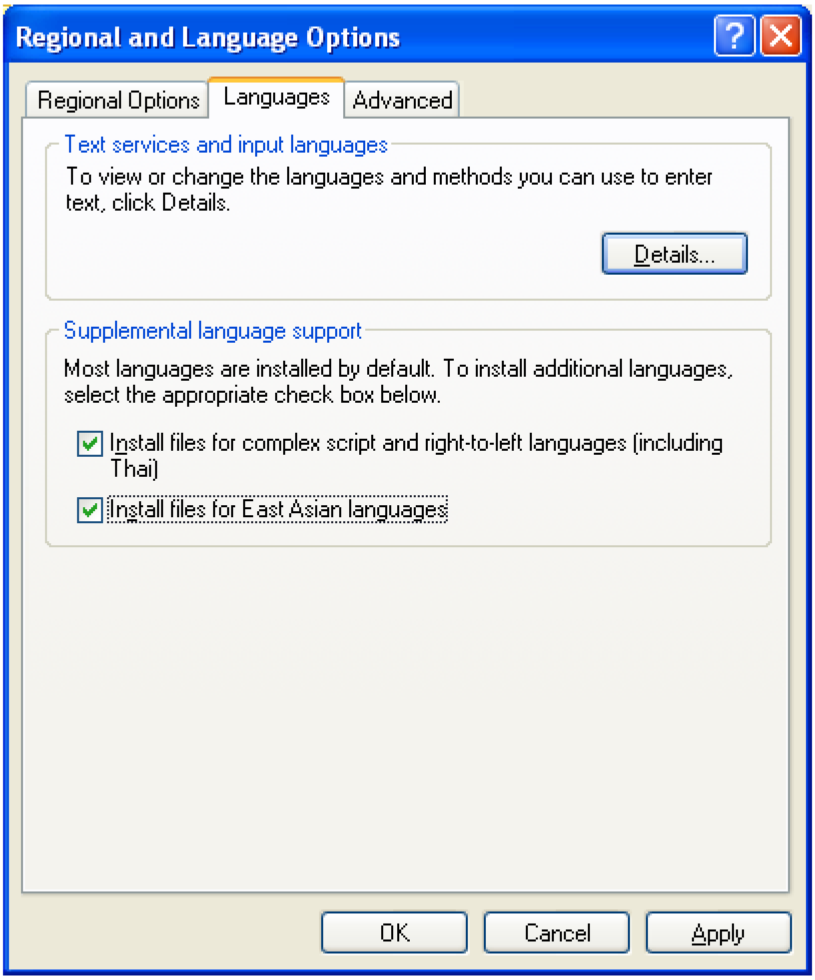
11 Working with Multi-Language in JMobile

A true Multilanguage feature has been implemented in JMobile through Code Pages support from the Microsoft Windows systems. The Multilanguage feature handles different code pages for the different languages. A code page (or a script file) is a collection of letter shapes used inside each language.

The Multilanguage feature can be used for a project by defining languages and character sets. JMobile also extends the TrueType Fonts (in short TTF) provided by Windows systems to provide different font faces associated with different character sets.

JMobile has features to allow users to provide strings for each of the languages.
When in edit mode, JMobile Studio provides support to change the display language from language combo. This helps users see the page's look and feel at Design Timtime itself.

Note: In Windows XP operating systems, for the proper operation of the multiple language editor in Studio, you need to install the support for complex script and East Asian languages (as shown in Figure 119).



Text services and input languages

To view or change the languages and methods you can use to enter text, click Details.

Supplemental language support

Most languages are installed by default. To install additional languages, select the appropriate check box below.

☑ Install files for complex script and right-to-left languages (including Thai)

☑ Install files for East Asian languages

Figure 119

JMobile is actually supporting a restricted set of fonts for the Chinese languages.

For Simplified Chinese, JMobile supports the following fonts:

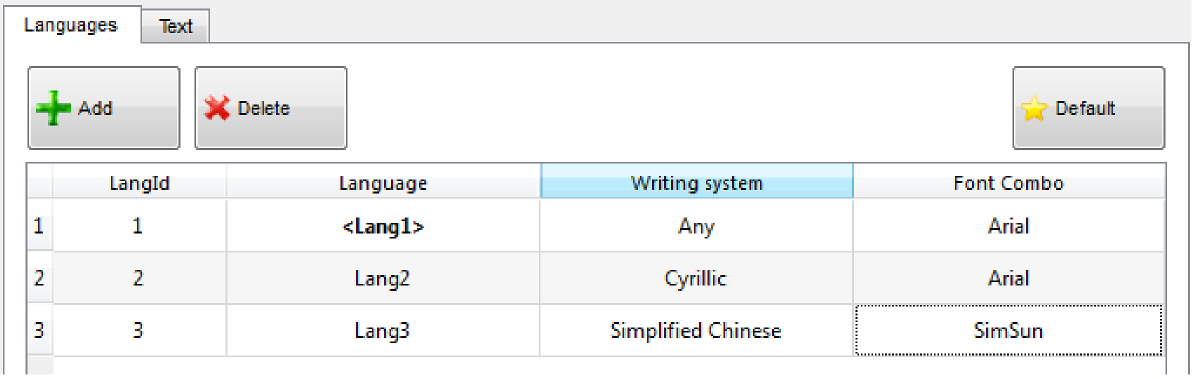
1. Fangsong - simfang.ttf
2. Arial unicode MS - ARIALUNI.TTF
3. Kaiti - simkai.ttf
4. Microsoft Yahei - msyh.ttf
5. NSImsun - simsun.ttc
6. SimHei - simhei.ttf
7. Simsun - simsun.ttc

For the Traditional Chinese language, JMobile supports instead the following fonts:

1. DFKai-SB - kaiu.ttf
2. Microsoft Sheng Hai - msjh.ttf
3. Arial unicode MS - ARIALUNI.TTF
4. MingLiU - mingliu.ttc
5. PMingLiU - mingliu.ttc
6. MingLiU_HKSCS - mingliu.ttc

11.1 Add a Language to Project

To add a language to a project, launch Multilanguage from the ProjectView pane. Click the “Add" button to add the language, then select the writing system and the default font used by all the “table like" Widgets (such as alarms or events). Use the “Default" button to set the default language used when the Runtime starts.



	LangId	Language	Writing system	Font Combo
1	1	<Lang1>	Any	Arial
2	2	Lang2	Cyrillic	Arial
3	3	Lang3	Simplified Chinese	SimSun

Figure 120

11.1.1 Language Display Combo

This combo can be used to change language at design phase. This helps users to view the page in different supported languages at design time itself.

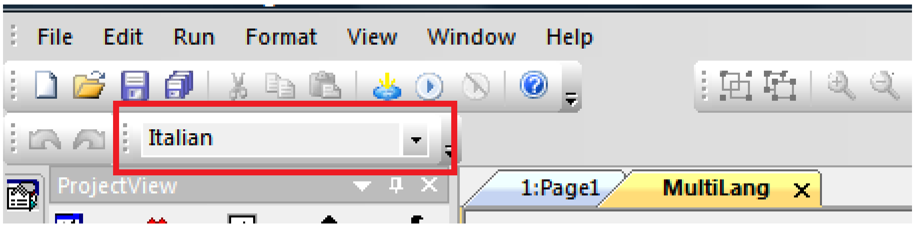


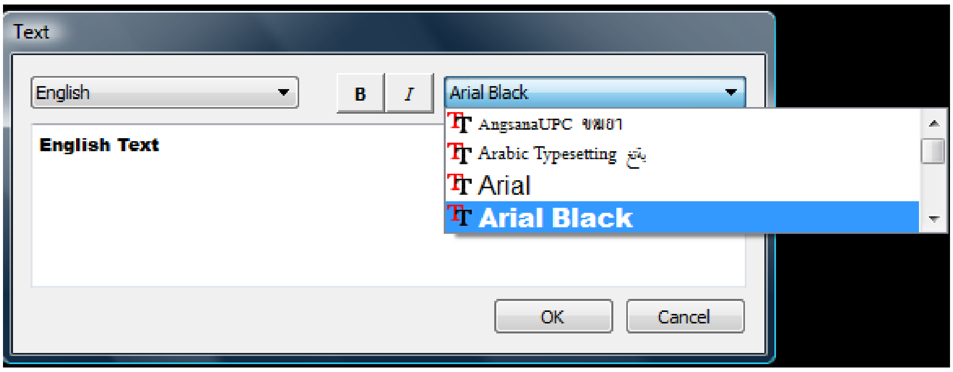
Figure 121

11.2 Multi Language Widget

Multilanguage support is available for different objects, like push buttons, Static Text, Message and Alarm description and the pop-up messages.

11.2.1 Multi Language for Static Text Widget

When you double click a text Widget on the page, the dialog shown in Figure 122 will open. Here, you can edit the text for the selected Languages and select the font.
The Bold, Italic and color properties are set for all the languages globally for the Widget. Text for each of the languages can be given, by selecting the language from combo.

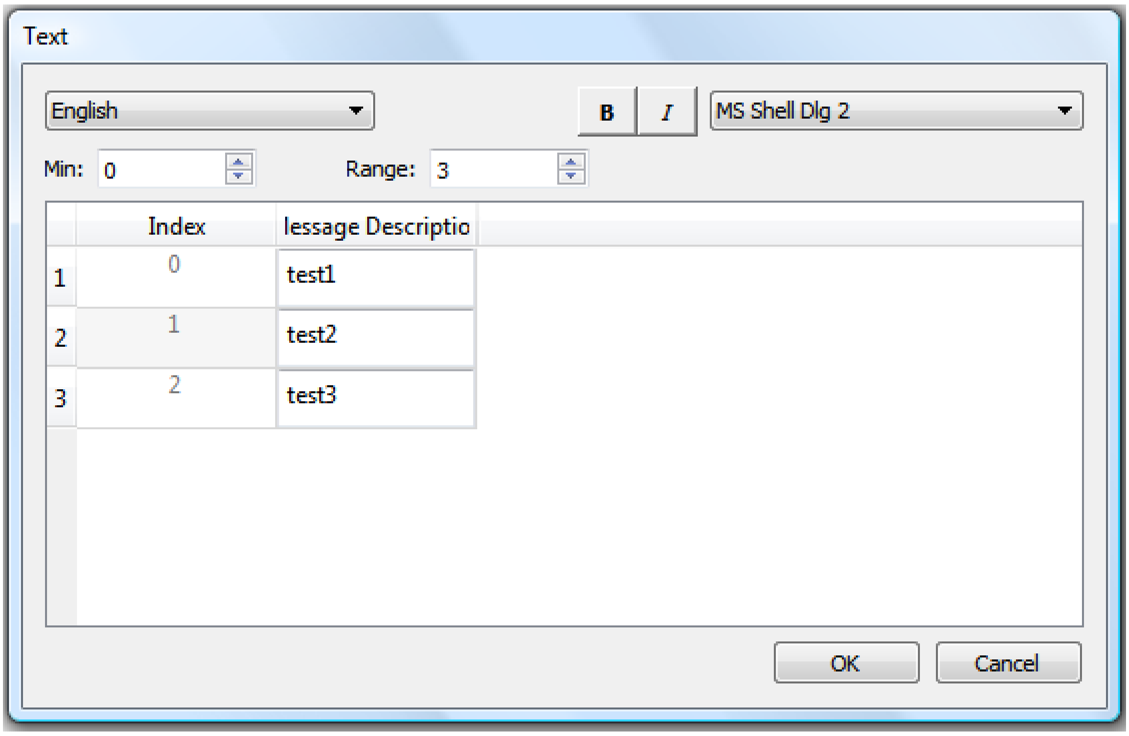


English Text

Figure 122

11.2.2 Multi Language for message Widget

JMobile allows you to use the Multilanguage in the message Widget. After you drag and drop a message Widget, select the language from the Language combo and enter the message description for the selected language. Do the same for all the languages or use Export and import strings as described in chapter: Export and Import of Multilanguage Strings.



English

MS Shell Dlg 2

Min: 0

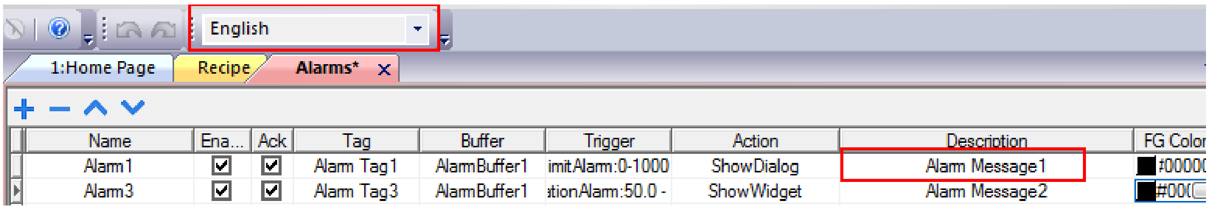
Range: 3

	Index	lessage Descriptio
1	0	test1
2	1	test2
3	2	test3

Figure 123

11.2.3 Multi Language for Alarm Messages

JMobile allows you to use Multilanguage for Alarm Messages.
To add a Multilanguage string for an Alarm message, open the alarm editor, select the language list from the tool bar (Language combo) and add the alarm messages. You can also use the export and import features, as described in chapter 11.3.

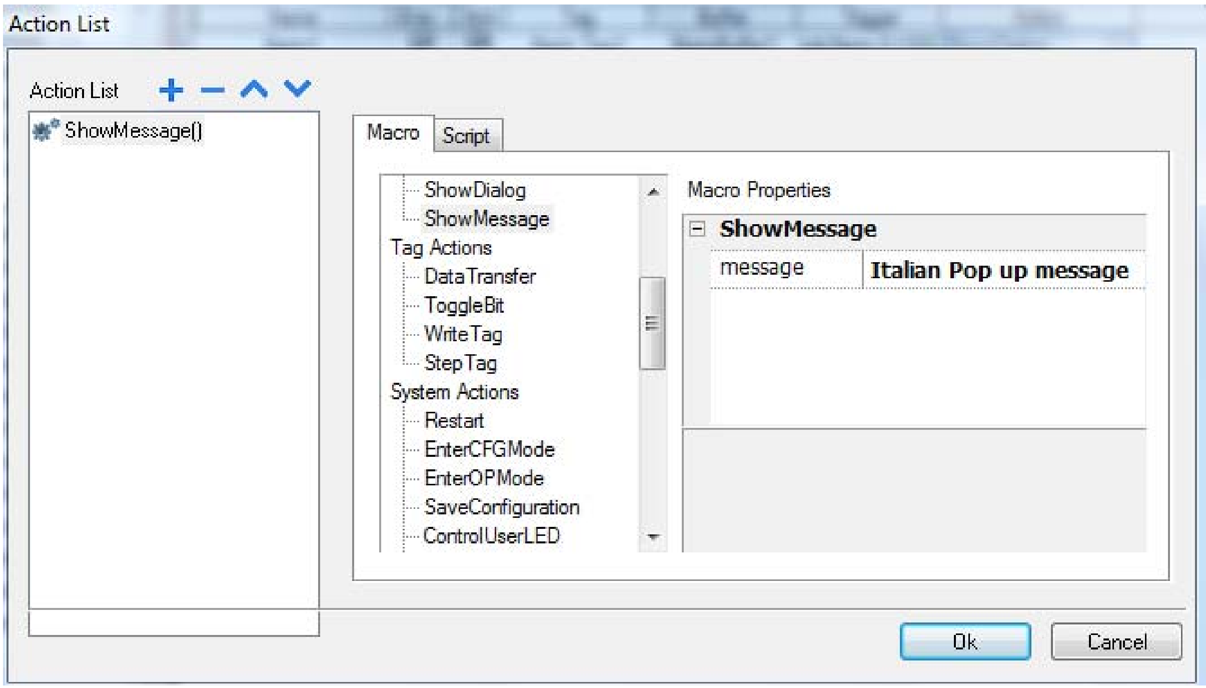


Name	Ena...	Ack	Tag	Buffer	Trigger	Action	Description	FG Color
Alarm1	☑	☑	Alarm Tag1	AlarmBuffer1	imitAlarm:0-1000	ShowDialog	Alarm Message1	■ #0000(
Alarm3	☑	☑	Alarm Tag3	AlarmBuffer1	ationAlarm:50.0 -	ShowWidget	Alarm Message2	■ #00(

Figure 124

11.2.4 Multi Language for Popup Messages

For the popup message macro, you can define the Multi Languages. To do this, you first need to select the language from language list combo, and then enter the message in the show message macro (as shown in Figure 125).



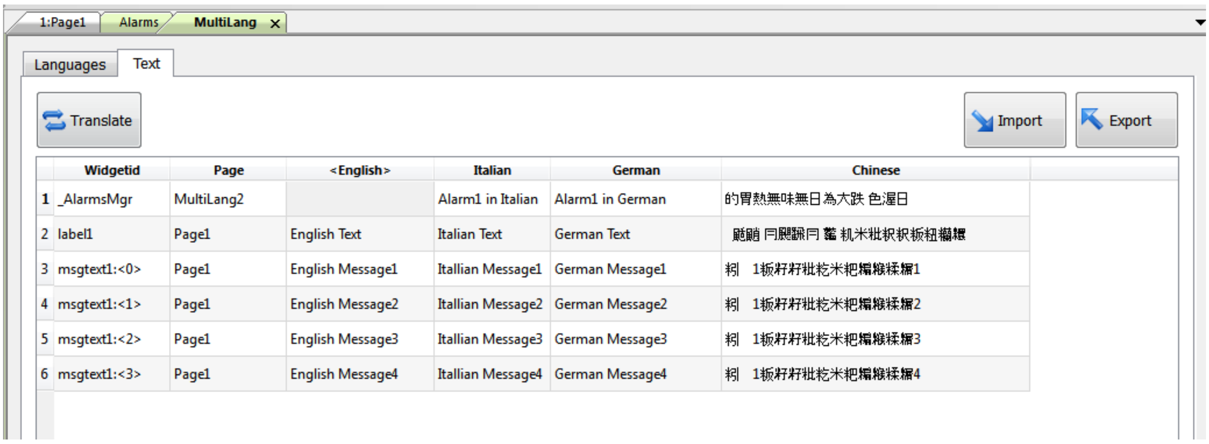
ShowMessage

message: Italian Pop up message

Figure 125

11.3 Export and Import of Multilanguage Strings

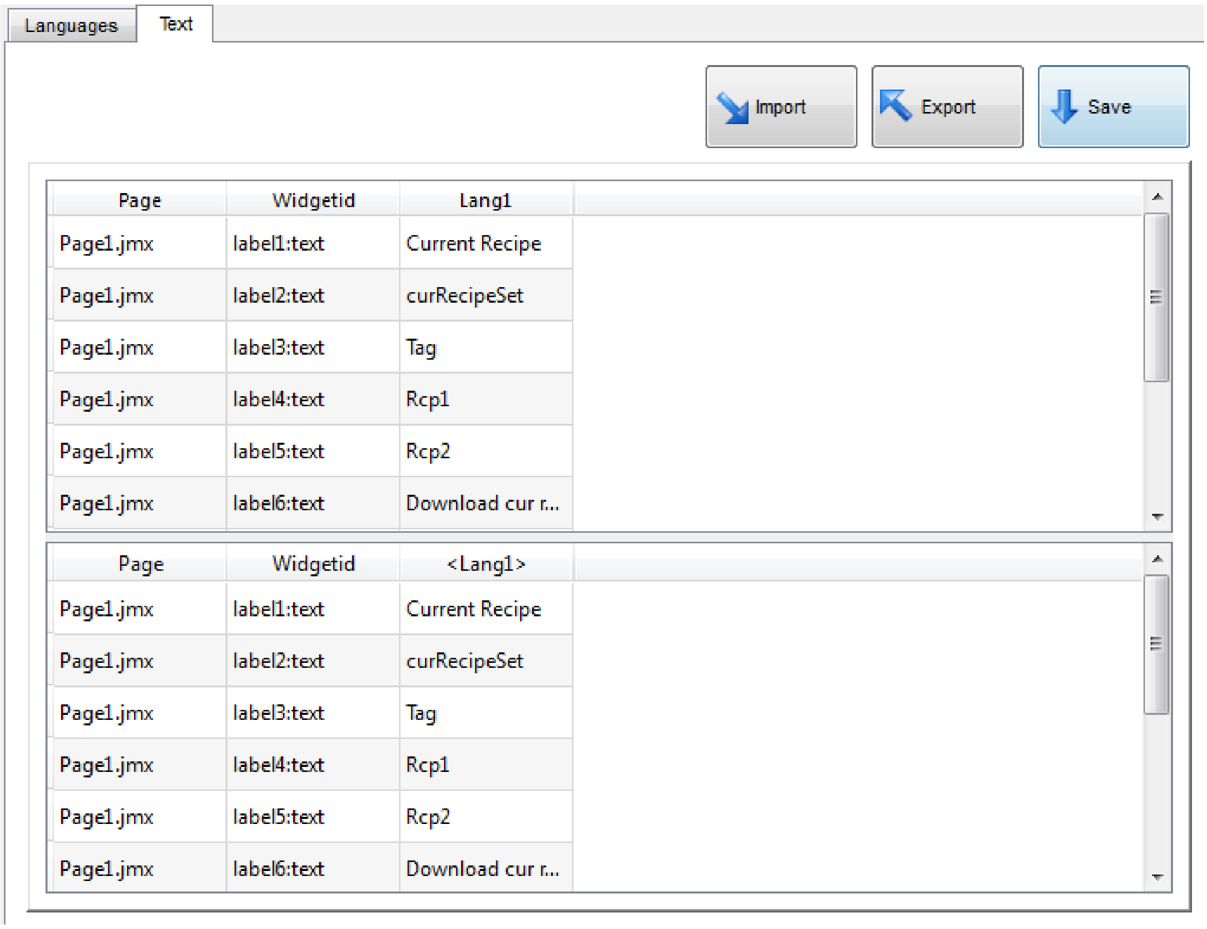
The easiest way to translate a project into multiple languages is most likely by Exporting to an external file, completing a translation operated with external tool and Importing the new string files back.
You can Export the Multilanguage strings in the CSV file format. First modify the string, from an external editor, and then Import it back to the JMobile Studio.
The CSV file exported by Studio is coded in Unicode. To edit it, you need a specific tool that supports a CCSV file encoded in a Unicode format.
To Export the Multilanguage string, open the Multilanguage editor and switch to “Text" View. Then, click the “Export" button and save the exported CSV file. You can then modify the exported CSV file and “Import" back to JMobile Studio. Click “Save" button to save the text.



	Widgetid	Page	<English>	Italian	German	Chinese
1	_AlarmsMgr	MultiLang2		Alarm1 in Italian	Alarm1 in German	的胃熱無味無日為大跌 色渥日
2	label1	Page1	English Text	Italian Text	German Text	颩颵 冃颵颵冃 [illegible] 机米粃粎粎粄粈粬粿
3	msgtext1:<0>	Page1	English Message1	Itallian Message1	German Message1	粡 1粄籽籽粃粇米粑糄糠糅糬1
4	msgtext1:<1>	Page1	English Message2	Itallian Message2	German Message2	粡 1粄籽籽粃粇米粑糄糠糅糬2
5	msgtext1:<2>	Page1	English Message3	Itallian Message3	German Message3	粡 1粄籽籽粃粇米粑糄糠糅糬3
6	msgtext1:<3>	Page1	English Message4	Itallian Message4	German Message4	粡 1粄籽籽粃粇米粑糄糠糅糬4

Figure 126

The strings are imported matching the Widget ID and the page number of each Widget. To change the separator used in the exported file, please have the regional settings of your work PC changed. Upon importing, the separator information is retrieved from the file; if not found, the default of "," is used. Immediately after the Import, the modified strings will be displayed in the text tab. Once the user hits the button to "Save" the changes, the changes are saved to the internal Widgets.



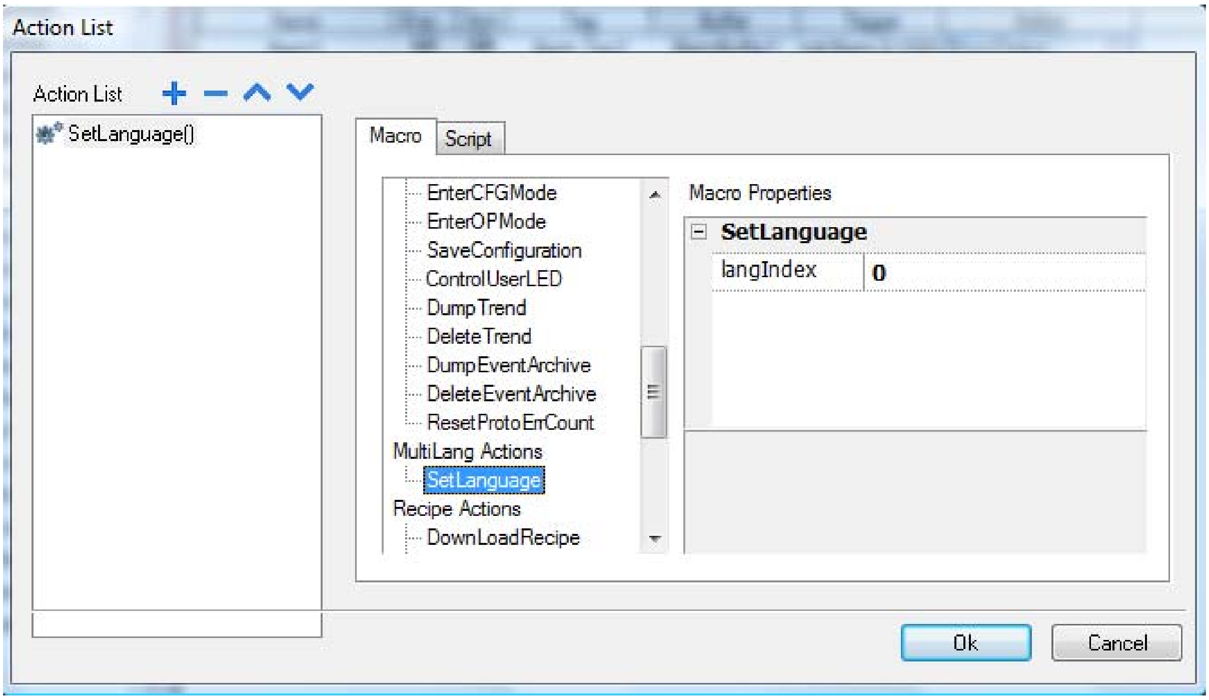
Page	Widgetid	Lang1
Page1.jmx	label1:text	Current Recipe
Page1.jmx	label2:text	curRecipeSet
Page1.jmx	label3:text	Tag
Page1.jmx	label4:text	Rcp1
Page1.jmx	label5:text	Rcp2
Page1.jmx	label6:text	Download cur r...

Page	Widgetid	<Lang1>
Page1.jmx	label1:text	Current Recipe
Page1.jmx	label2:text	curRecipeSet
Page1.jmx	label3:text	Tag
Page1.jmx	label4:text	Rcp1
Page1.jmx	label5:text	Rcp2
Page1.jmx	label6:text	Download cur r...

Figure 127

11.4 Change Languages at Runtime

After the project download, Runtime will start with the default language. However, you can change the language on Runtime using the “Set language" macro.



SetLanguage

langIndex	0

Figure 128

The language index corresponds to the language ID, as it can be read from the language configuration editor.

Note: After languages are changed on Runtime with the macro execution, the changed language is saved and retained for the next run.

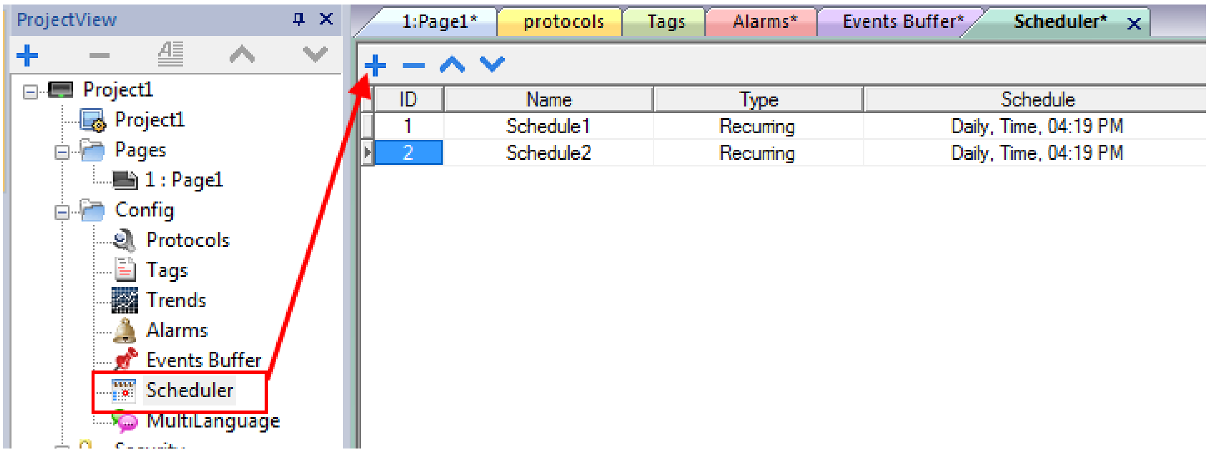
12 Working with Scheduler

The JMobile System provides a scheduler engine that can be easily configured to program the execution of specific actions at repeated intervals, on a daily or weekly basis.
Depending on your application, creating a schedule is typically performed by a 2-step process. The first step is to define the parameters of the schedule that run on the panel. This includes selecting the actions that occur when the scheduled event fires. The first step is performed using the Schedule Editor, in JMobile Studio.

The second step is to create a Runtime user interface that allows the end-user to change settings per each defined user. For example the Rutime user interface will allow the user to turn on a device at 5:00 pm, and turn the device off at 10:00 pm, every day. The second step is performed by dragging and dropping a predefined schedule Widget, from the Gallery, and placing it on the page. Once on the page, you can set the properties of the individual GUI elements to create the desired interface to be presented to the end-user.

12.1 Configuring the Scheduler Engine

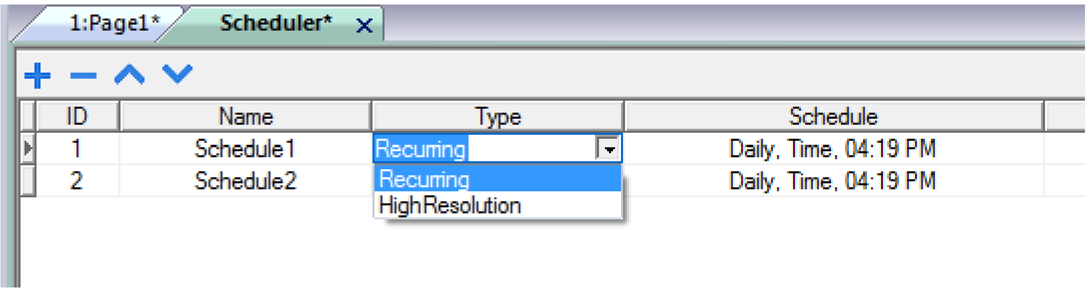
The configuration of the Scheduler Engine is done using the Scheduler Editor. The Schedule Editor is accessible from the “Config" folder of the ProjectView pane (as shown in Figure 129).



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

Figure 129

Click on the “+" symbol to add a schedule item. Scheduler can be of two different types as listed below and shown in Figure 130.
Recurring Scheduler
High Resolution scheduler



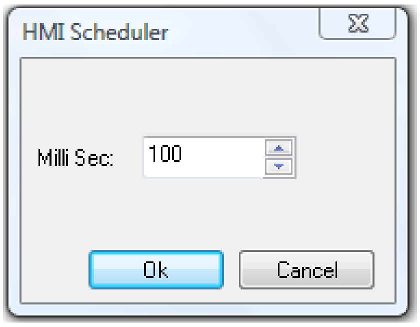
ID	Name	Type	Schedule
1	Schedule1	Recurring	Daily, Time, 04:19 PM
2	Schedule2		Daily, Time, 04:19 PM

Figure 130

The “Name" column is for the Schedule name.
The “Type" column allows you to select the type of scheduler.
The “Schedule" column allows you to select different scheduler options, which are described in chapters 12.3 and 12.4.
The “Action" column allows you to define macros, which have to be executed at the scheduled time.
The “Priority" column allows you set a priority level for the event. This is used in case two distinct schedules occur at the same time. The event with the higher priority will be executed before those of lower priority.

12.2 High Resolution

The High Resolution scheduler can be programmed to perform an action, or sequence of actions, repeatedly, at a specific duration. The High Resolution scheduler can be set in milliseconds. To configure the High Resolution scheduler, select “High Resolution" from the Type column and set the desired duration from the schedule column.



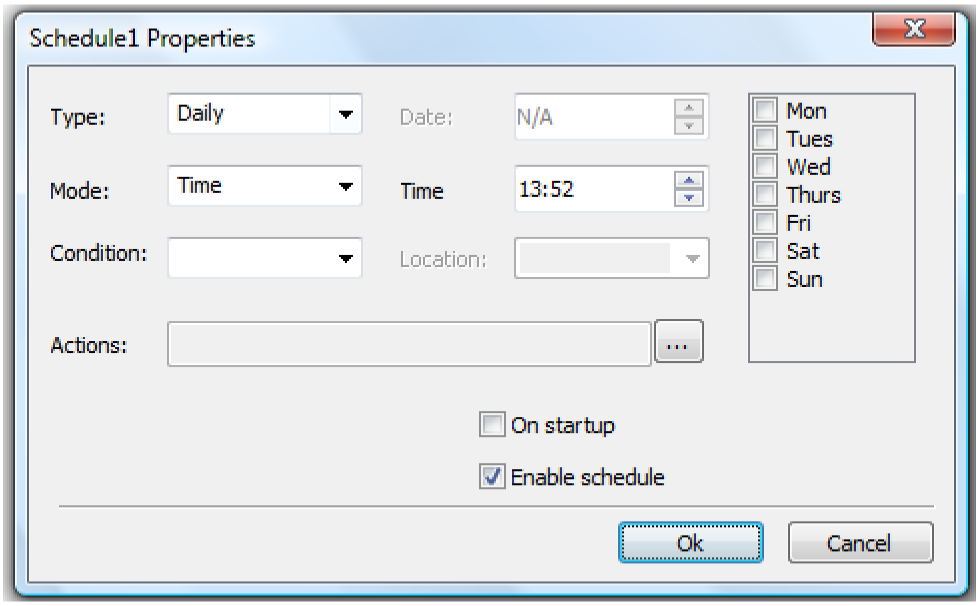
Milli Sec: 100

Figure 131

Note: The High Resolution scheduler cannot be changed during run-time. The High Resolution scheduler time is given in milliseconds. If user wants to change the schedule run-time, then use the Recurrence scheduler by selecting “Every", which is described in the following chapters. The minimum time resolution, when using a Recurrence scheduler in “Every" mode, is one second.

12.3 Recurrence Scheduler

The Recurrence Schedulers can be programmed to perform an action, or sequence of actions, and the schedule can be modified during Runtime.



Schedule1 Properties

Type: Daily

Date: N/A

Mode: Time

Time 13:52

Condition:

Location:

Actions:

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

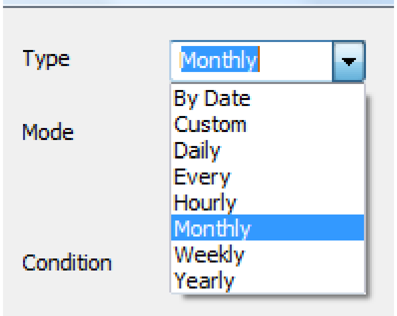
□ On startup

☑ Enable schedule

Figure 132

12.4 Type

The Type combo allows you to select the type of the Schedulers (as shown in Figure 133). However, you can change the type for the scheduler at anytime during the Runtime, as described in chapter 12.10.



Type Monthly

Mode

Condition

Figure 133

12.4.1 By Date

By Date scheduler allows you to define the schedule for the specific date and time when the actions shall be executed. To define the schedule by date scheduler, select “By Date" from the type combo and set the date and time.

12.4.2 Daily Schedule

Daily schedules define the execution of a set of actions on a daily basis by specifying the time of day in which the actions to be executed. To configure the Daily schedule, select Daily from the Type combo and set the time. For some of the Schedule types, we can further control them through different modes of Sunrise+, Sunrise-, Random, etc. (The Mode selection is described in chapter 12.5.).

12.4.3 Every Schedule

The Every Scheduler is much like the High Resolution scheduler, with the possibility of changing in in the Runtime. The “Every" Scheduler allows you to execute the macros with the specific time interval. The time interval can be set from 1 sec to 1 day (For example, data transfer from one protocol to another with an interval of every 5 minutes).

12.4.4 Hourly Schedule

The Hourly Schedules allow you to execute a set of actions on an Hourly basis, by specifying the minute in which the actions have to be executed. To configure the Hour schedule, select hourly from the Type combo and set the time of the macro execution. For example, values to be written to the controller on 5th Minute of every hour.

12.4.5 Monthly Schedule

The Monthly Schedules allow you to execute a set of actions on a Monthly basis, by specifying the day in which the actions have to be executed. To configure the Month schedule, select Monthly from the Type combo and set the day and time of the macro execution. For example, for values to be written to the controller at 4:35 on 5th of every month, a monthly scheduler can be programmed to perform the actions.

12.4.6 Weekly Schedule

Weekly schedules allow you to execute a set of actions on a Weekly basis by specifying the time and day(s) in which the actions have to be executed. To configure the Week’s schedule, select Weekly from the Type combo and select the days of the week and the time for the macro execution. For example, for an Alarm to be triggered at 6:00 AM on weekdays (Monday to Friday), the Weekly scheduler can be programmed to perform the same actions.

12.4.7 Yearly Schedule

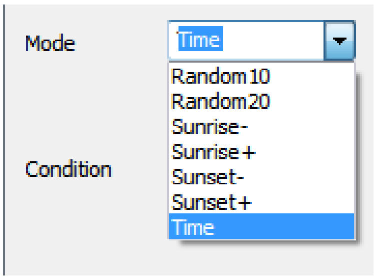
The Yearly schedule allows you to execute a set of actions once a year, specifying the date and time in which the actions have to be executed. To configure the Yearly Schedule, select Yearly from the Type combo and select the date and time for the macro execution. For example, for daylight saving time to be set on a specific date every year, set the date and time of the starting date for the daylight saving and define the macros to set the new time.

12.4.8 Custom

The Custom mode allows to specify a “one shot" scheduled action(s).

12.5 Mode

The Mode combo allows you to specify the mode of the schedule, like Time basis or Sunrise+, sunrise-, sunset+, sunset-, random, etc.



Mode: Time

Figure 134

12.5.1 Time

The Time mode allows you to set the schedule based on the Local time settings. For example, if you select Daily from the Type combo, Time from Mode combo, and set the time as 15:55:00, then the scheduled action will be executed at 3:55 PM.

12.6 Configuring Location in JMobile Suite

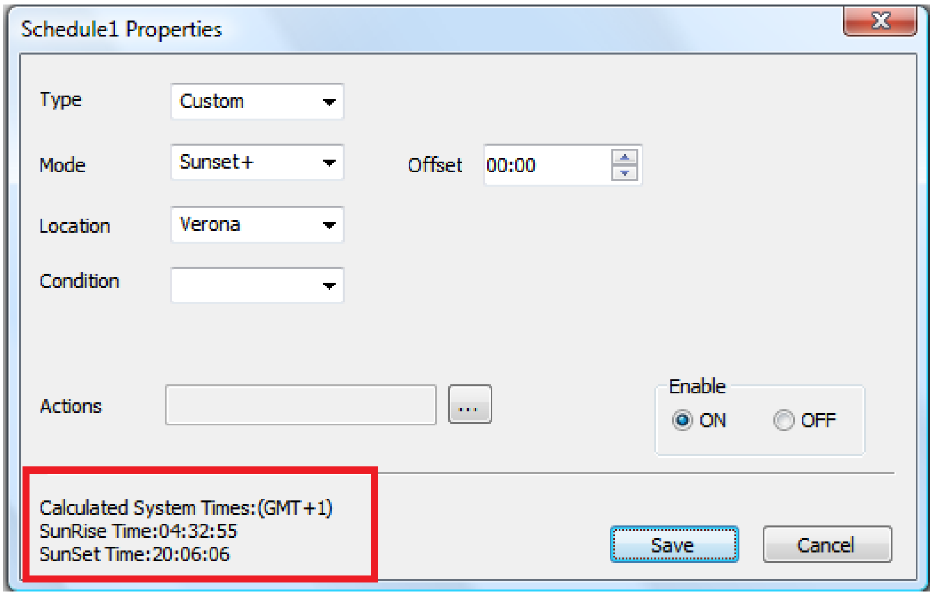
In JMobile Suite, we have a unique feature that schedules based on sunrise and sunset. Before you start the sunrise or sunset scheduler, you need to define the location. Based on the UTC location, the JMobile system automatically calculates the sunrise and sunset time.

In the studio installation, few locations are set by default. If your location does not show up under the list, you can add your location by entering the latitude, longitude and UTC information in the “Target_Location.xml" file under JMobile Suite\studio\config root folder.

For example, the information for Verona City is as shown below:

```
<file city="Verona" latitude="45.44" longitude="10.99" utc="1"/>
```

After entering the location information, the Studio displays the city name in the place combo list, and you can see the sunrise and sunset time on the dialog (as in the Figure 135).



Schedule1 Properties

Type: Custom

Mode: Sunset+

Offset: 00:00

Location: Verona

Condition:

Actions:

Enable: ● ON ○ OFF

Calculated System Times:(GMT+1)
SunRise Time:04:32:55
SunSet Time:20:06:06

Figure 135

12.6.1 Sunrise+

The Sunrise+ Scheduler allows you to make a schedule based on the sunrise. To configure a Sunrise+ Scheduler, select Sunrise+ from the Mode combo. Then, select your location and set the offset time. For example, if an alarm needs to be triggered 30 minutes after the sunrise, select the Sunrise+ mode, set the offset as 30 minutes and specify the actions to be executed.

12.6.2 Sunrise-

The Sunrise- Scheduler allows you to make a schedule based on the sunrise. To configure a Sunrise- Schedule,r select Sunrise- from the Mode combo, select your location and set the offset time. For example, if an alarm needs to be triggered 30 minutes before the sunrise, select the Sunrise- mode, set the offset as 30 minutes and specify the actions to be executed.

12.6.3 Sunset+

The Sunset+ Scheduler allows you to make a schedule based on the sunset. To configure a Sunset+ Scheduler, select Sunset+ from the Mode combo, select your location and set the offset time. For example, if an alarm needs to be triggered 30 minutes after the sunset, select the Sunset+ mode, set the offset as 30 minutes and specify the actions to be executed.

12.6.4 Sunset-

Sunset- Scheduler allows you to make a schedule based on the sunset. To configure a Sunset- Scheduler, select Sunset- from the Mode combo, select your location and set the offset time. For example, if an alarm needs to be triggered 30 minutes before the sunset, select the Sunset- mode, set the offset as 30 minutes and specify the actions to be executed.

12.6.5 Random10

The Random 10 Scheduler specifies that the triggering time is randomly affected by a factor of +/-10 minutes. To configure the Random 10, select Random from combo mode, set the time of the offset and define the actions in the macros. The action will be executed at the set time +/- 10 minutes

12.6.6 Random20

The Random 20 Scheduler specifies that the triggering time is randomly affected by a factor of +/-20 minutes. To configure the Random 20, select Random from combo mode, set the time on the offset and define the actions in the macros. The action will be executed at the set time +/- 20 minutes

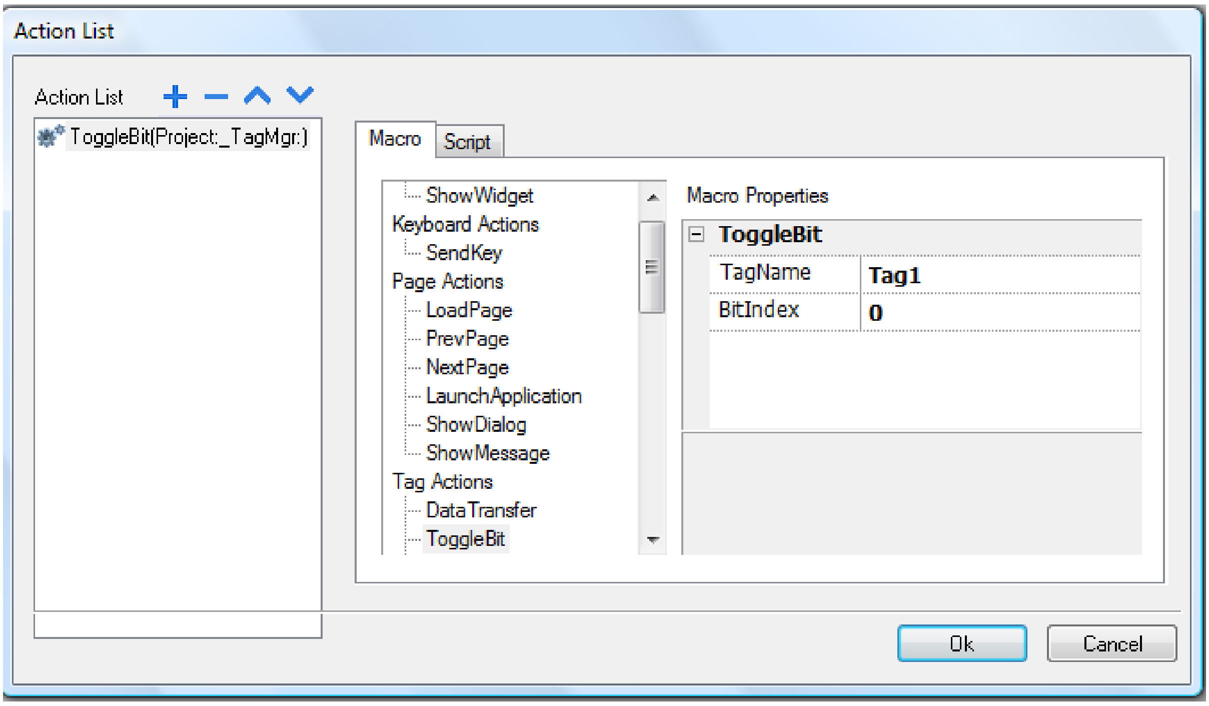
12.7 Condition

The Condition combo allows you to select a Boolean Tag (Yes/No) to be evaluated, before activating the specified actions, at the moment the timer is triggered. If Tag = True, actions will be executed, and if Tag = False, the actions will not be executed.
By default, there is “none" => the actions are executed when the timer is triggered.

Note: The condition combo will list only the Tag attached to the Boolean data type.

12.8 Actions

From the Action List dialog, you can add as many Actions as desired. The Actions will automatically be executed when the Schedule time occurs.



ToggleBit

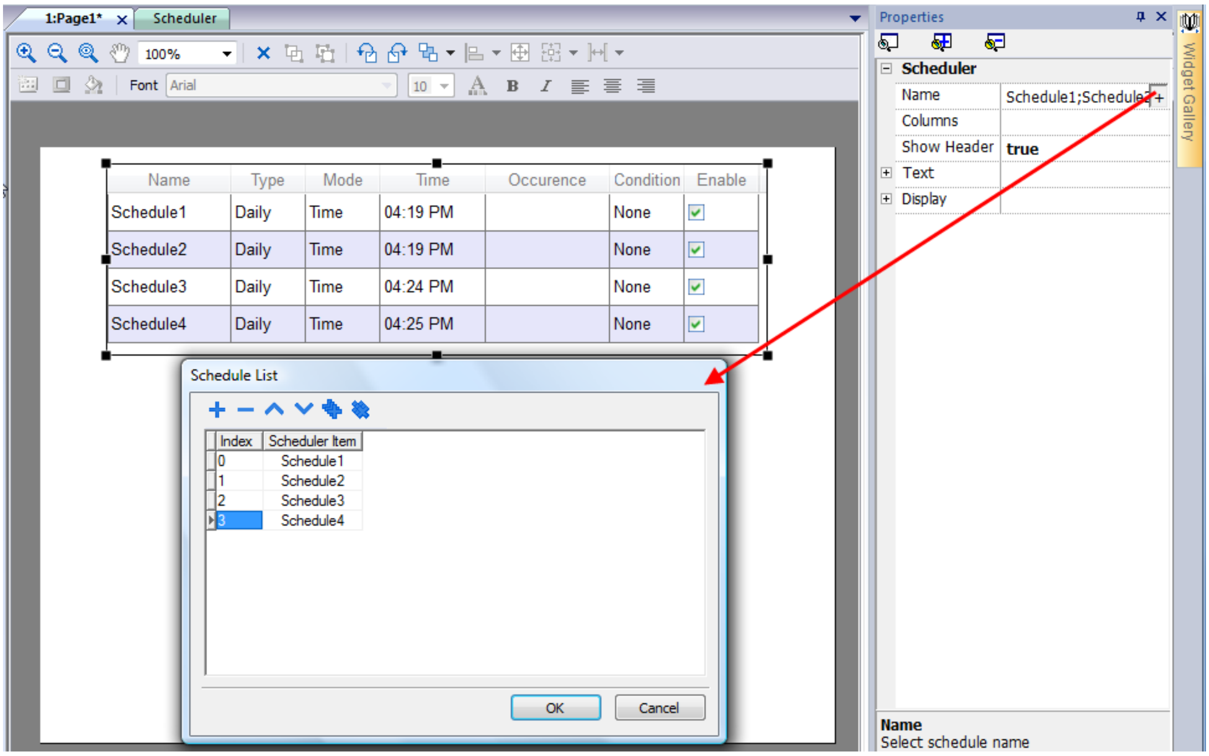
TagName	Tag1
BitIndex	0

Figure 136

Note: The Actions should be programmed in the Studio. It is only actions that cannot be modified in Runtime, all other scheduler parameters can be modified in Runtime (such as, type, mode, location, etc.)

12.9 Configuring the Schedule Interface for Run-time Interaction

The User Interface for run-time is the Widget called Scheduler. To add this to the project, just drag and drop it from the Advanced section of the Widget Gallery. Once the object is on the page, in order to select the Scheduler items to be displayed in the Widget, click on the + button of the “Name" property that is part of the Scheduler object. A Dialog page will open (as shown in Figure 137) where you can add the schedule from the list.



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

Index	Scheduler Item
0	Schedule1
1	Schedule2
2	Schedule3
3	Schedule4

Scheduler

Name	Schedule1;Schedule2+
Columns	
Show Header	true
Text	
Display	

Name
Select schedule name

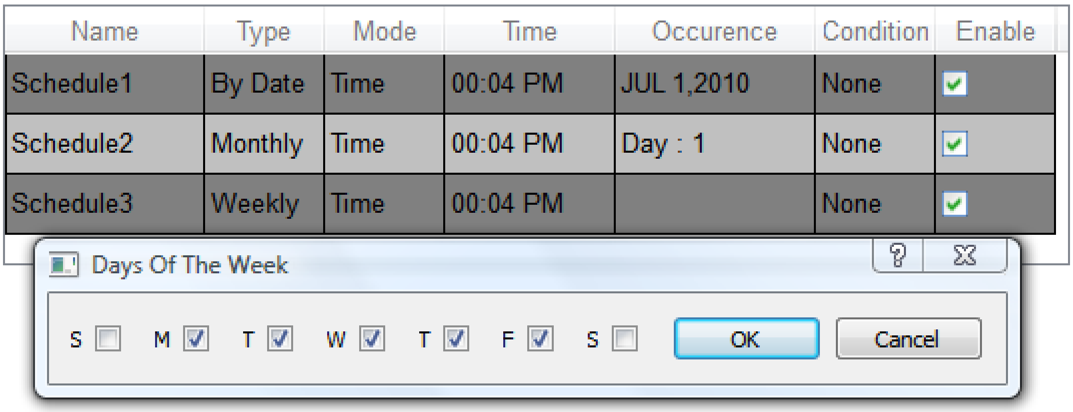
Figure 137

In the Properties pane, you can customize the scheduler Widget to adjust row colors, column width, show or hide column, etc.

12.10 Schedule the Events During Run-time

If you defined the scheduler GUI on a page (as described in chapter 125), then you can schedule the event, and modify this schedule, during run-time on the server.

In run-time, the user has the flexibility to change all possible types and change the possibility to mode as described in the dedicated chapter.



Name	Type	Mode	Time	Occurence	Condition	Enable
Schedule1	By Date	Time	00:04 PM	JUL 1,2010	None	☑
Schedule2	Monthly	Time	00:04 PM	Day : 1	None	☑
Schedule3	Weekly	Time	00:04 PM		None	☑

Days Of The Week

S □ M ☑ T ☑ W ☑ T ☑ F ☑ S □

Figure 138

12.11 Occurrence

The Occurrence column specifies the date selected by the type of column, as shown in the figure.

12.12 Condition

The Condition column lists the available Boolean Tags from the project. If a Tag is selected as a condition, then the scheduler will trigger only when the condition Tag value is 1—otherwise the scheduler will not trigger.

12.13 Enable

The Enable check box allows you to enable or disable the schedule. The scheduler will trigger when the enable check box is set. If you want to disable the scheduler temporarily, then uncheck the Enable check box.

13 User Management and Passwords

This chapter describes the requirements for user management and how to restrict access to various objects and/or operations only for certain authorized user groups. Users, user groups and authorizations are the 3 entities used for users' handling. Each user must be a member of a group. Users can be a member of just one group. Each group will have different types of authorizations and permissions assigned to them.
Authorizations and permissions for the groups are divided in two basic categories: Widgets' permissions (hide, read only, full access) and Actions' permission (allowed or not allowed). The proper combination of these will allow for the implementation of the necessary handling of security options for the application.

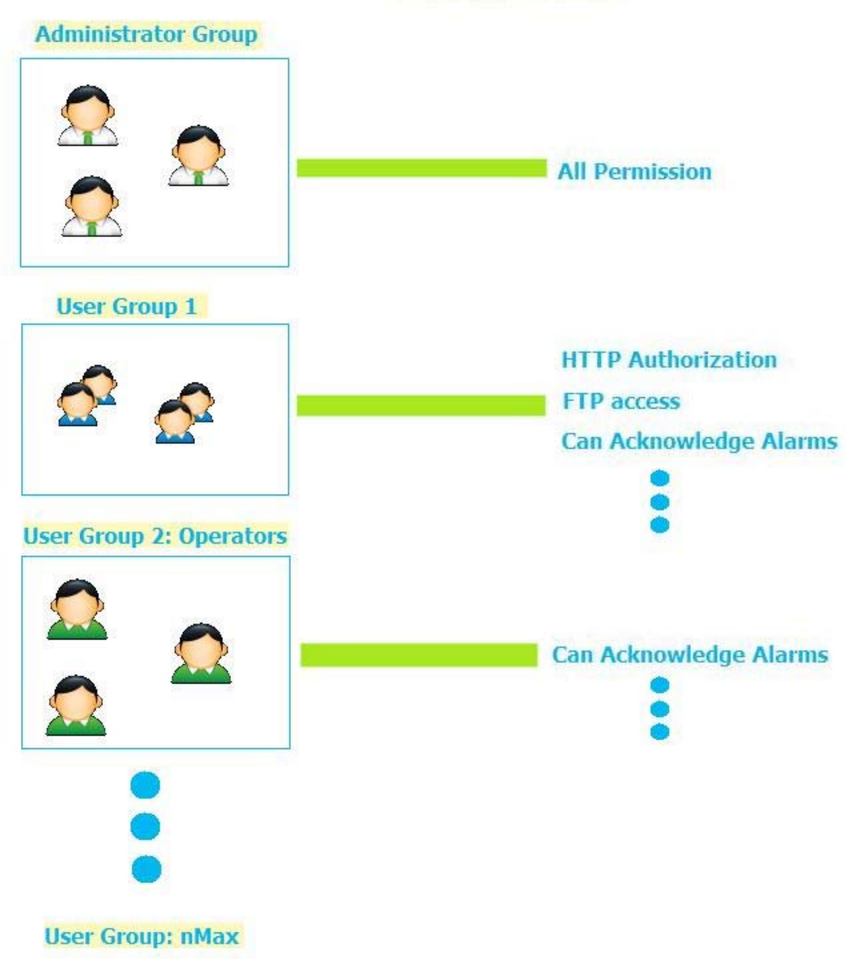


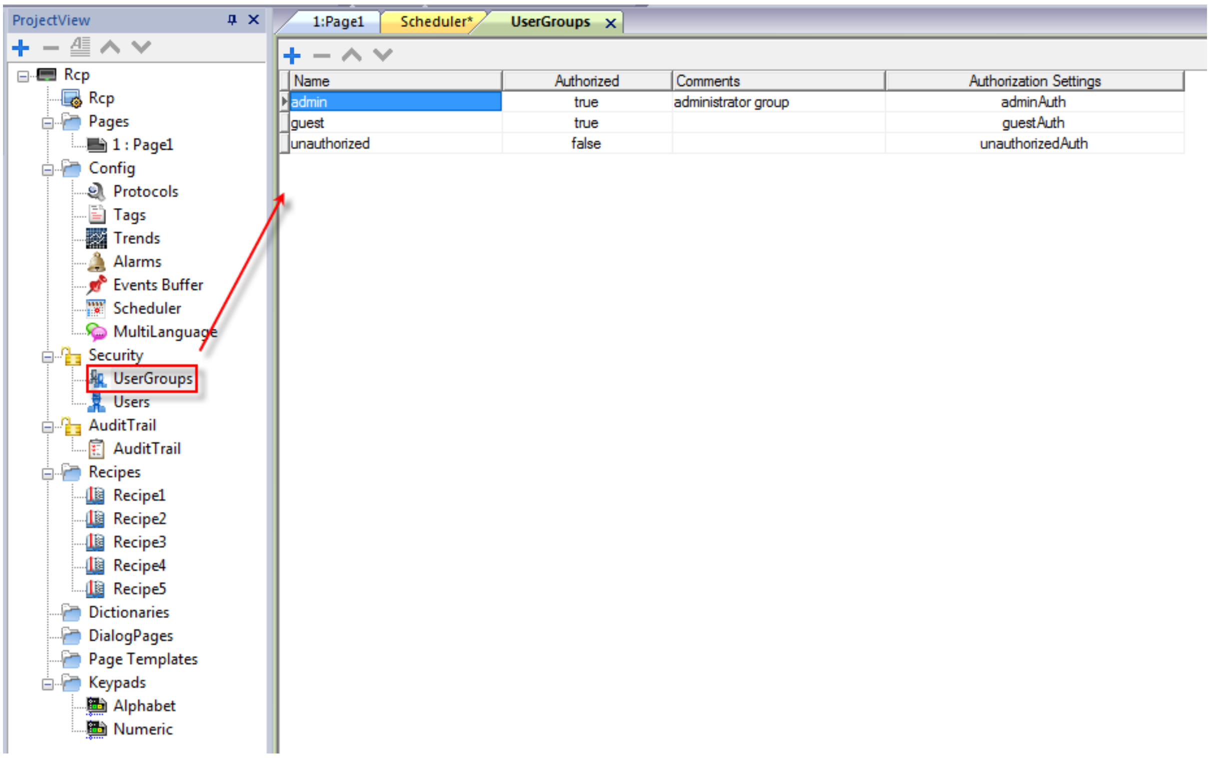
Figure 139

13.1 Configuring Authorizations, Groups & Users in Studio

The section below describes how to configure security-related settings in JMobile Studio.
To enable or disable the user management feature, right click on the "Security" folder from the Project View and set Enable or Disable.

13.2 Configuring Groups and Authorizations in Studio

Open the “User Groups" to configure and assign their authorization in JMobile studio.



Name	Authorized	Comments	Authorization Settings
admin	true	administrator group	adminAuth
guest	true		guestAuth
unauthorized	false		unauthorizedAuth

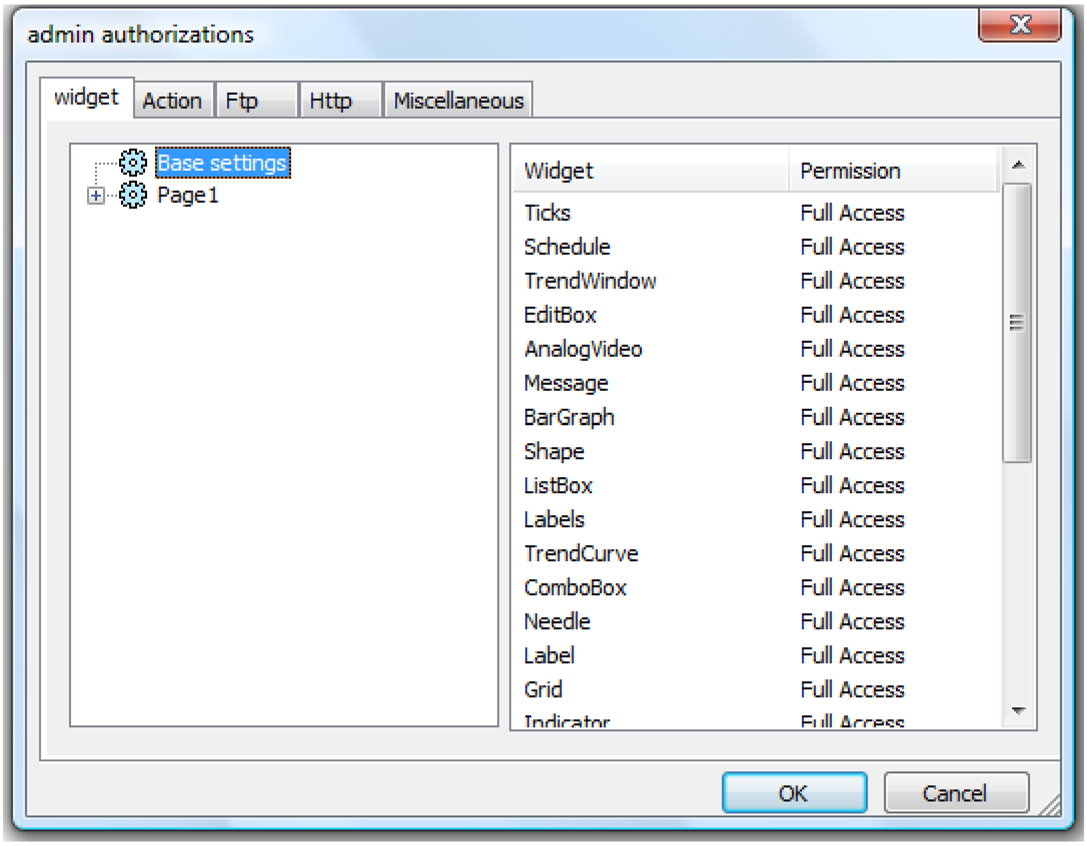
Figure 140

The three predefined groups cannot be deleted. Also, the name of the predefined groups cannot be edited.
However, their authorizations and comment fields can be edited.

13.3 Adding and Modifying the Access Permission of Groups

To add a user group, click on the “+" button and choose the name of the group.
To modify and assign the permissions, click the browse button on the Authorization Setting column, then you will see the dialog shown in the figure below.

13.3.1 Widget Permissions

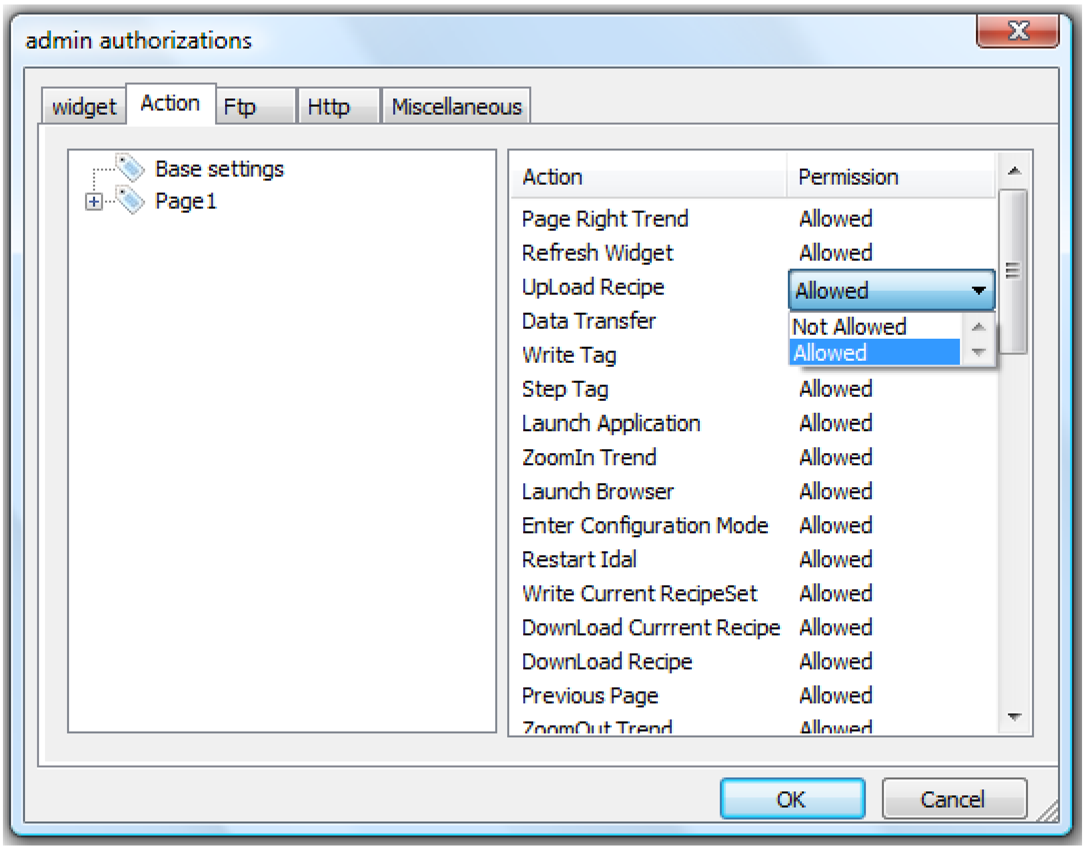


Widget	Permission
Ticks	Full Access
Schedule	Full Access
TrendWindow	Full Access
EditBox	Full Access
AnalogVideo	Full Access
Message	Full Access
BarGraph	Full Access
Shape	Full Access
ListBox	Full Access
Labels	Full Access
TrendCurve	Full Access
ComboBox	Full Access
Needle	Full Access
Label	Full Access
Grid	Full Access
Indicator	Full Access

Figure 141

In this dialog, it is possible to assign global access for the Widgets of the project. The possible choices are: Full-Access, Read-Only and Hide.
The Widget security settings applicable to all pages of a project will be edited by selecting Widgets node.
The properties window will show the different types of Widgets. These Widgets can be hidden, read-only or full-access mode.
The Widget security settings can be changed, not only globally, but also for each single Widget present within the project; all the Widgets can be reached from the tree structure on the left part of the Widget tab.
Later in the chapter, we explain how to modify permissions for a specific Widget directly from the page view (rather than locating the Widget from the tree view shown in the authorizations' dialog).

13.3.2 Action Permissions



Action	Permission
Page Right Trend	Allowed
Refresh Widget	Allowed
UpLoad Recipe	Allowed
Data Transfer	
Write Tag	
Step Tag	Allowed
Launch Application	Allowed
ZoomIn Trend	Allowed
Launch Browser	Allowed
Enter Configuration Mode	Allowed
Restart Idal	Allowed
Write Current RecipeSet	Allowed
DownLoad Currrent Recipe	Allowed
DownLoad Recipe	Allowed
Previous Page	Allowed
ZoomOut Trend	Allowed

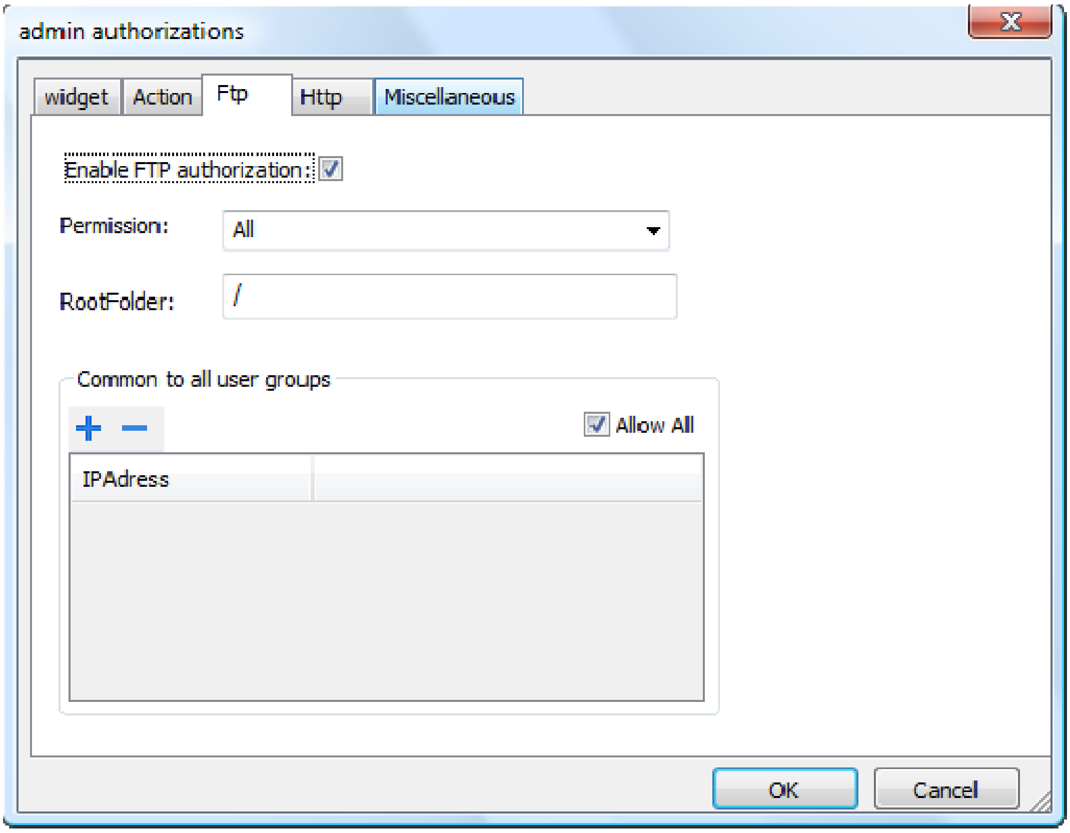
Figure 142

With this dialog, it is possible to assign the authorizations for the actions with respect to a project. The Access is either Allowed and Not allowed.
As for the Widgets, the authorizations can be assigned globally, but also for each single action, and programmed into the project.
Later in the chapter, we will explain how to modify permissions for a specific action directly from the page view (rather than locating the action from the tree view shown in the authorizations' dialog).

13.3.3 FTP Authorizations

Per each group you can set specific authorizations related to the use of the FTP server as shown in the following picture.
FTP permissions can be enabled or disabled. If enabled, you can specify from the "Permissions" combo box the access level selecting between All, Write, Read, Browse, and None.
The IP Address list access allows to specify from which IP an incoming FTP connection should be accepted.

Note: IP access list configuration is common to all groups



Enable FTP authorization: ☑

Permission: All

RootFolder: /

Common to all user groups

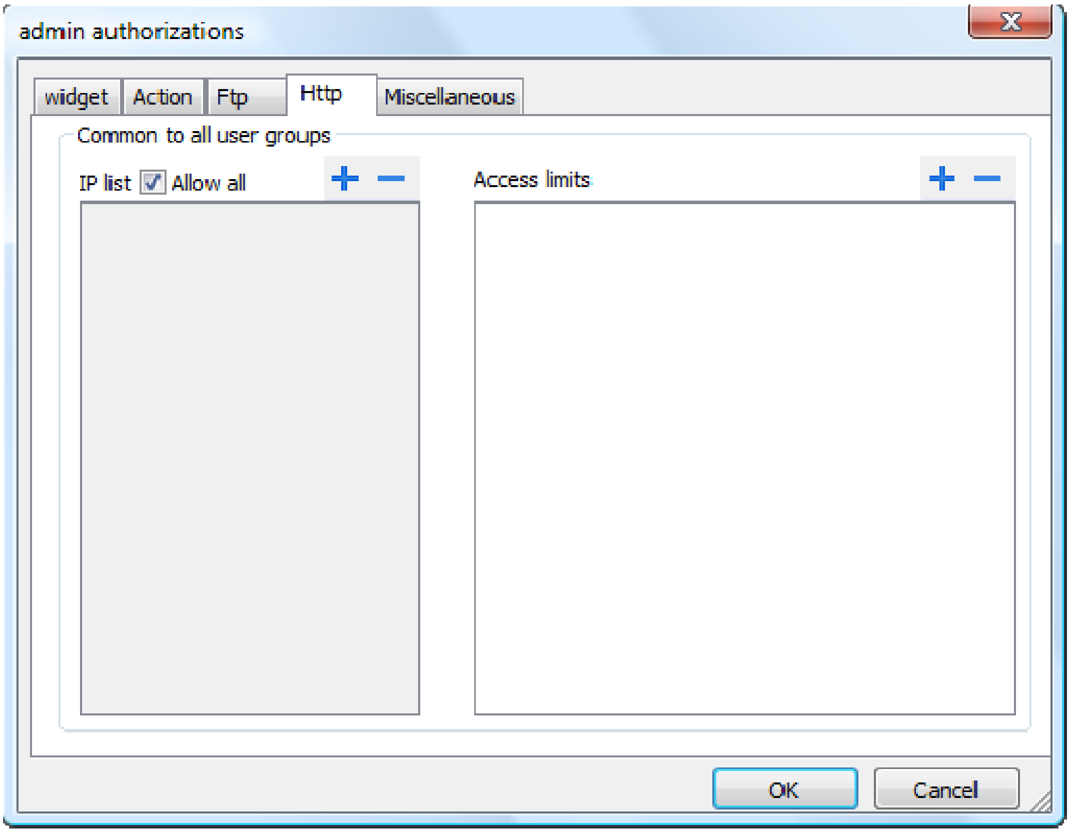
☑ Allow All

IPAdress

Figure 143

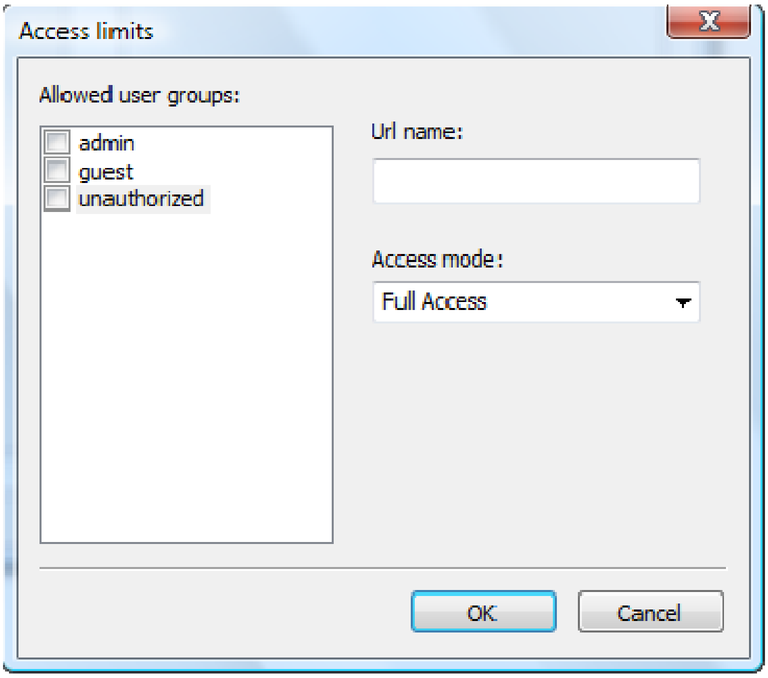
13.3.4 HTTP Authorizations

The HTTP authorization dialog allows to configure the IP access list, which is common to all groups. Additionally, the user can define specific access permissions to specific URL path within the JMobile web server.



☑ Allow all

Figure 144



Allowed user groups:

□ admin
□ guest
□ unauthorized

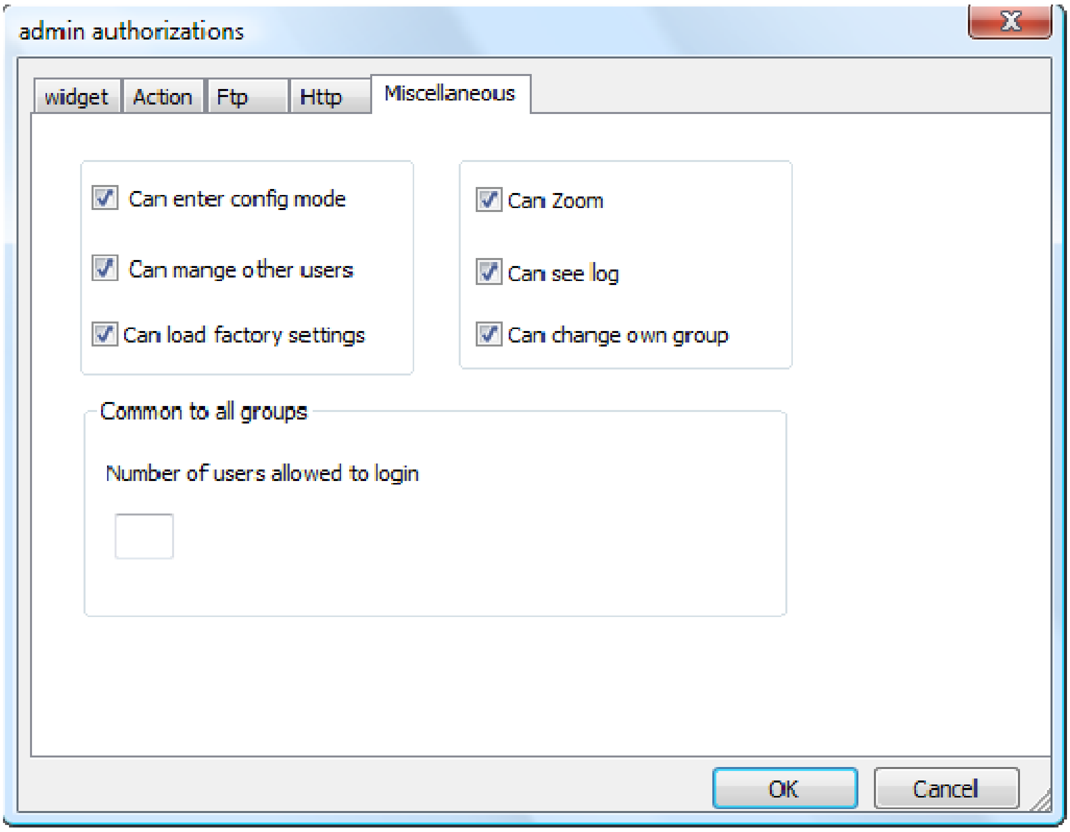
Url name:

Access mode: Full Access

Figure 145

13.3.5 Miscellaneous

The Miscellaneous tab contains different settings related to the several options as indicated in the following picture.
Please note that as indicated in the picture, some settings are related to the group, but some settings are global to all groups.



☑ Can enter config mode
☑ Can mange other users
☑ Can load factory settings
☑ Can Zoom
☑ Can see log
☑ Can change own group

Common to all groups

Number of users allowed to login

Figure 146

13.3.6 Access Priority

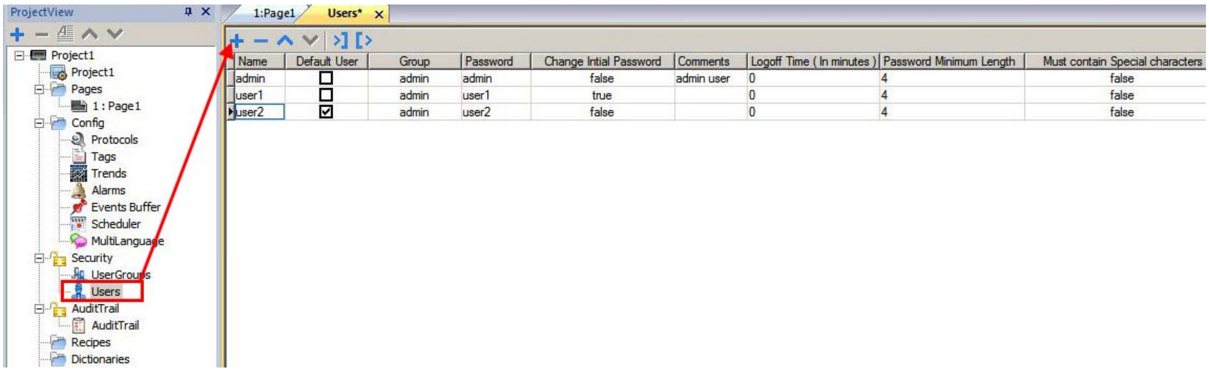
If the Access control is applied to a Widget, page and or even the Global Access, then the top priority goes to the Widget access.

Top Priority : Control from Widget
Medium Priority : Page Access or its Parent Access
Low Priority : Global Access

In other words, this means that “exceptions" configured for an action or a Widget, directly from the page view, have priority on the base settings.

13.4 Configuring Users in Studio

To configure users in JMobile Studio, double click on Users from the Project View, and then click on + to add a new user. A user named admin is already present by default, this user cannot be deleted.



Name	Default User	Group	Password	Change Intial Password	Comments	Logoff Time (In minutes)	Password Minimum Length	Must contain Special characters
admin	□	admin	admin	false	admin user	0	4	false
user1	□	admin	user1	true		0	4	false
user2	☑	admin	user2	false		0	4	false

Figure 147

Name: User Name

Default User: Identifies the user which is automatically logged-in by the system when starting, re-starting or after a logout; only one default user is allowed.

Group: Select user groups for the user.

Password:Enter the initial password for the user

Change Initial
Password: If True, the user is forced to change his password on first logon

Comments: Comments for the user

Logoff time (In Min): The user will be automatically logged off when there is no operation for the specific time in Runtime of the server. After Log off, the JMobile Server goes to default user.

Minimum Length: In Numbers, the minimum length of the password should be equal or greater than the set value.

Must Contain Special Characters: If True, the password should contain at least one special character

Must Contain Numbers: If True, the password should contain at least one numeric digit.

13.5 Default User

You can program a default user for a project. When the server starts or reboots, the Runtime is logged in with the default user. All the privilege settings of the default user will be activated in the system. If you want to log in as different user in Runtime, you can use either the Switch User macro or the Log Off macro.
The default user will automatically get logged in if any user (other than default user) logs off.

13.6 Assigning Widget Permissions from Page View

In JMobile, we can assign different security accesses to different users for a single Widget, also directly from project pages.
Select the Widget, then right click and select Security settings from the context menu. Next, choose the group and assign the security properties to access the Widget (as shown in figure).

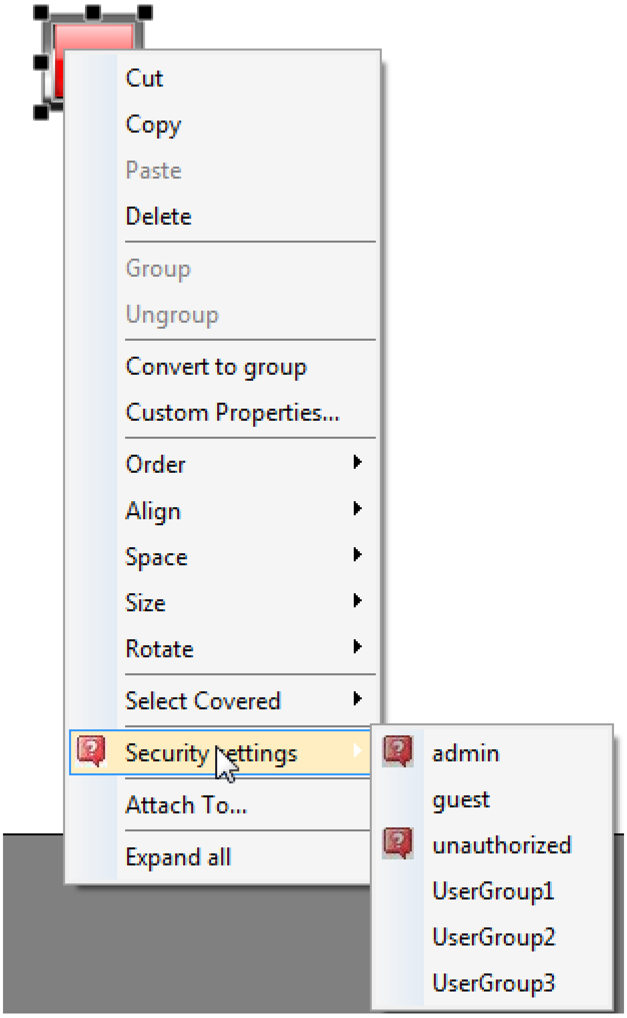
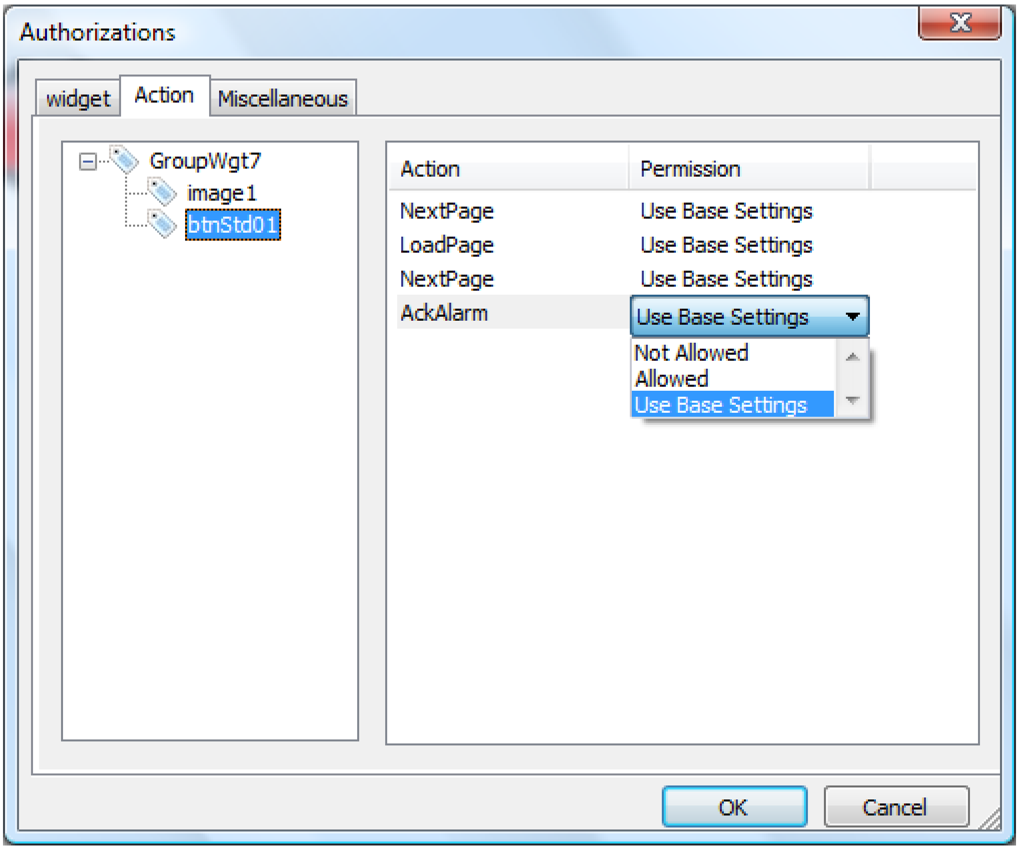

Cut
Copy
Paste
Delete
Group
Ungroup
Convert to group
Custom Properties...
Order
Align
Space
Size
Rotate
Select Covered
Security settings
Attach To...
Expand all
admin
guest
unauthorized
UserGroup1
UserGroup2
UserGroup3

Figure 148



Action	Permission
NextPage	Use Base Settings
LoadPage	Use Base Settings
NextPage	Use Base Settings
AckAlarm	Use Base Settings

Figure 149

13.7 Operation on Runtime

After starting the Target, the JMobile system will ask for User name and Password, based on the user, Runtime allows only the configured permissions for the logged user.
If a default user is specified within the project, the JMobile system will provide automatic login of that user without prompting for user login.
Please note that there are specific actions for user logout, user edit, user add, user remove and user switch.
In particular, please note that users can be edited, added or removed on Runtime using the specifications provided in the Action List, as shown in Figure 151.
All the users' information modified in Runtime is stored in a separated file, thereby preventing loss of the users' configuration in case of a new project download.
The proper action “DeleteUMDynamicFile" must be used if the modified user settings have to be deleted and you want to revert them all back to initial settings.

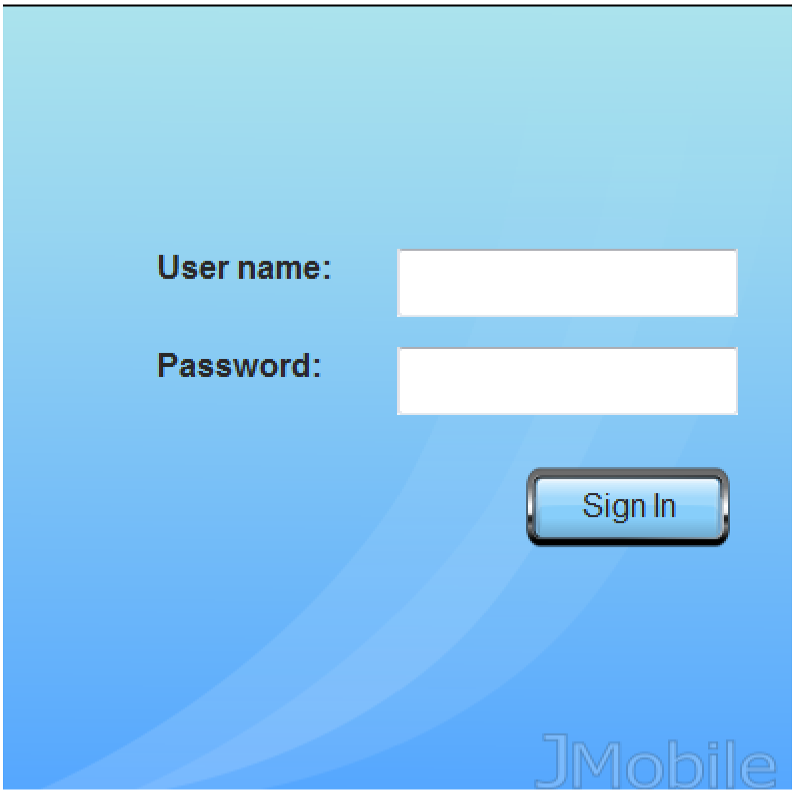


Figure 150

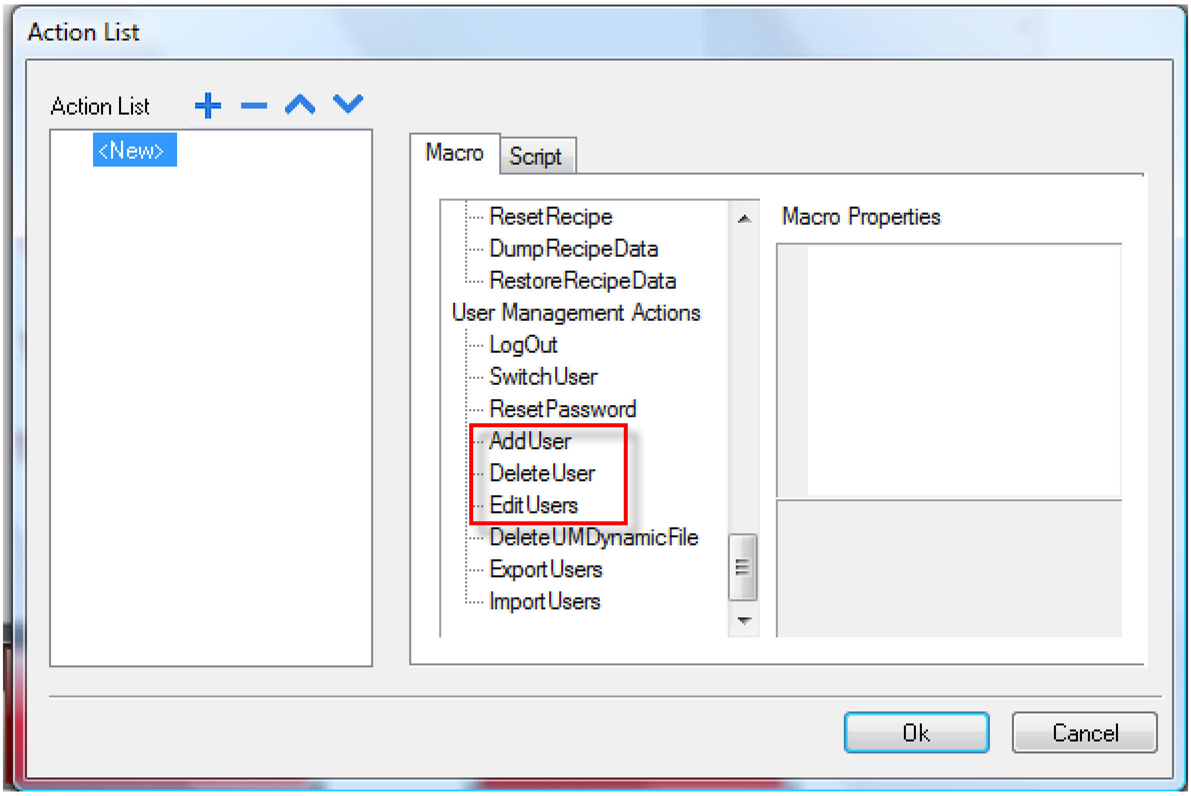


Figure 151

14 Audit Trails

JMobile supports Audit Trail functionality, which provides essential process tracking, user identification linked to time and the date of events logged facilitating recalls and/or rationalizing of your production processes.
The Audit Trail function provides flexible, tailor-made and easy-to-review event logs.

The Audit Trail (or audit log) is a chronological sequence of audit records, each containing evidence of the actions executed and the user that did these.
The Audit Trail can be enabled with or without user management. So it could access and supervise all actions from all users, and a normal user could not stop or change this.

14.1 Enable or Disable the Audit Trail

In the Project view pane, right click on the Audit Trail and click either enable or disable for the Audit Trail recording on Runtime. The padlock symbol in the tree informs you that, in the project, the Audit Trail is enabled or disabled. When the Audit Trail is enabled, and the padlock symbol turns locked otherwise, it stays open.

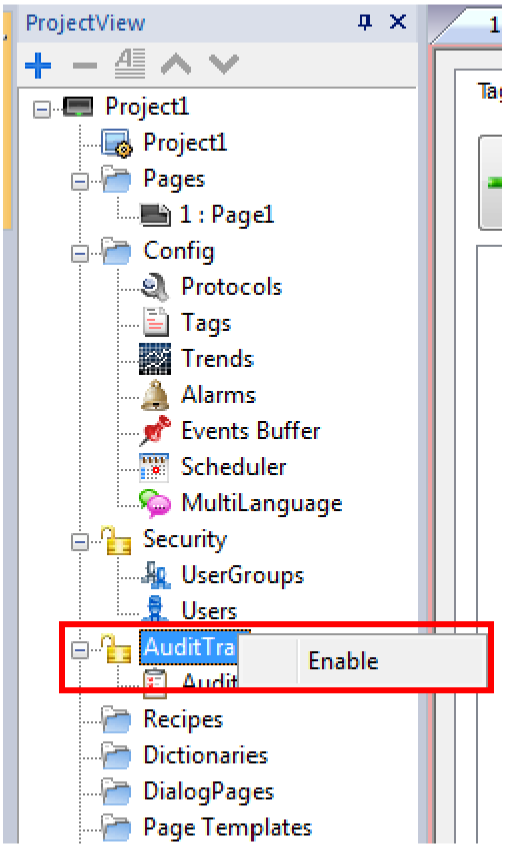
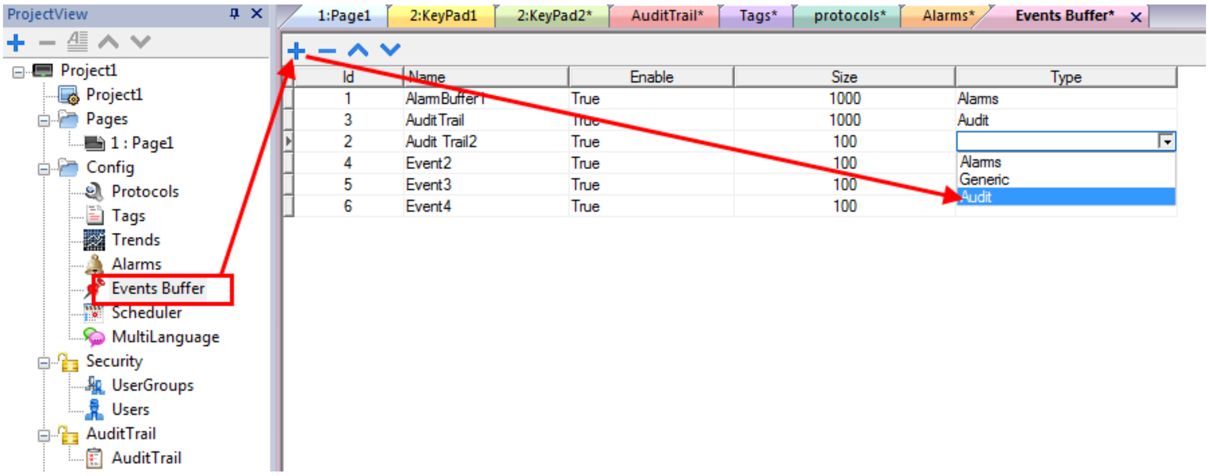


Figure 152

14.2 Configure Audit Events

You can have more than one set of Audit Records. To add to the Audit files, you need to configure the Events buffer.
Double click the Events buffer from the project workspace. Next, add the events buffer and set the file size, and ten select the Log type “Audit".

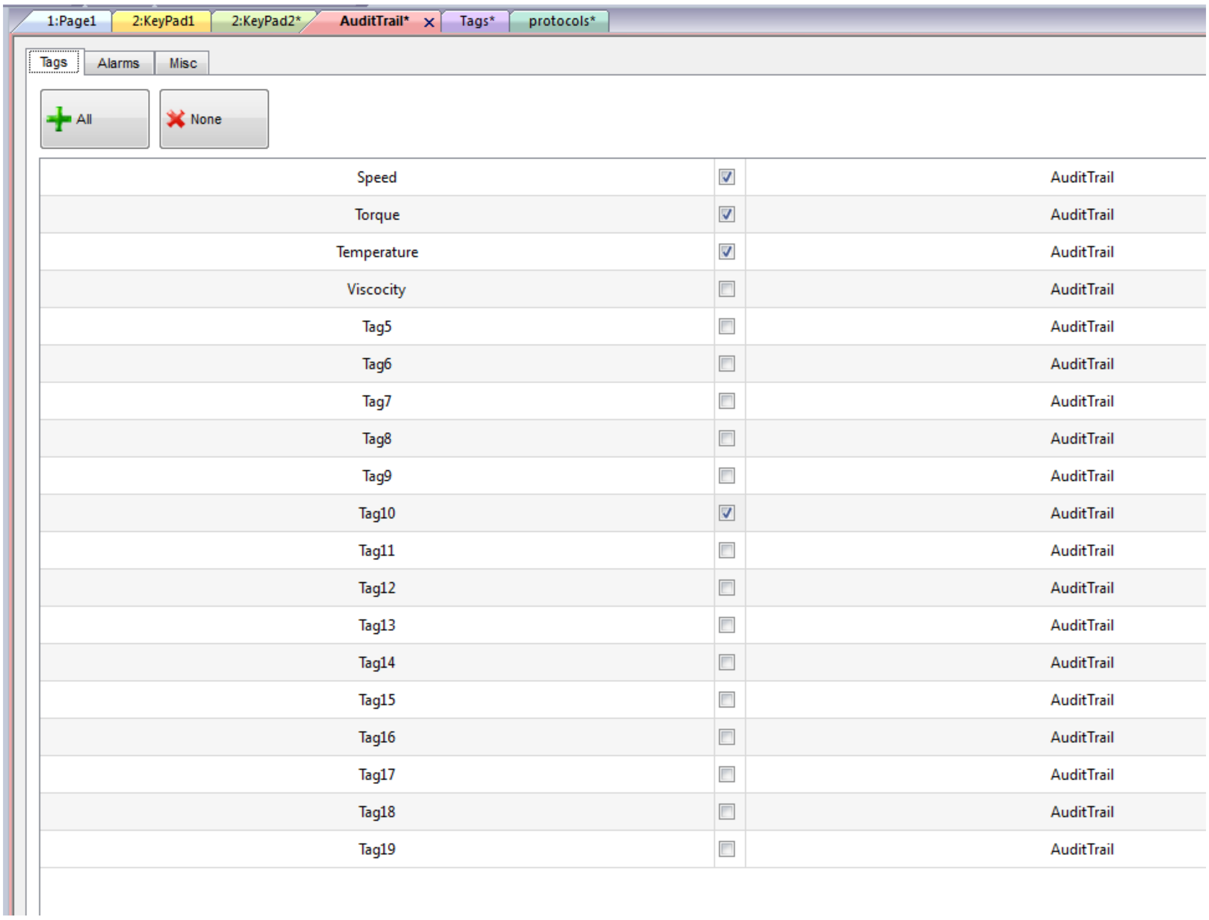


Id	Name	Enable	Size	Type
1	AlarmBuffer1	True	1000	Alarms
3	Audit Trail	True	1000	Audit
2	Audit Trail2	True	100	
4	Event2	True	100	
5	Event3	True	100	
6	Event4	True	100	

Figure 153

14.3 Configure Tags in the Audit Trail

For most of the cases, all the Tags specified in the project are not necessarily to be monitored. So, you can customize the Tags to be monitored by Audit Trail.



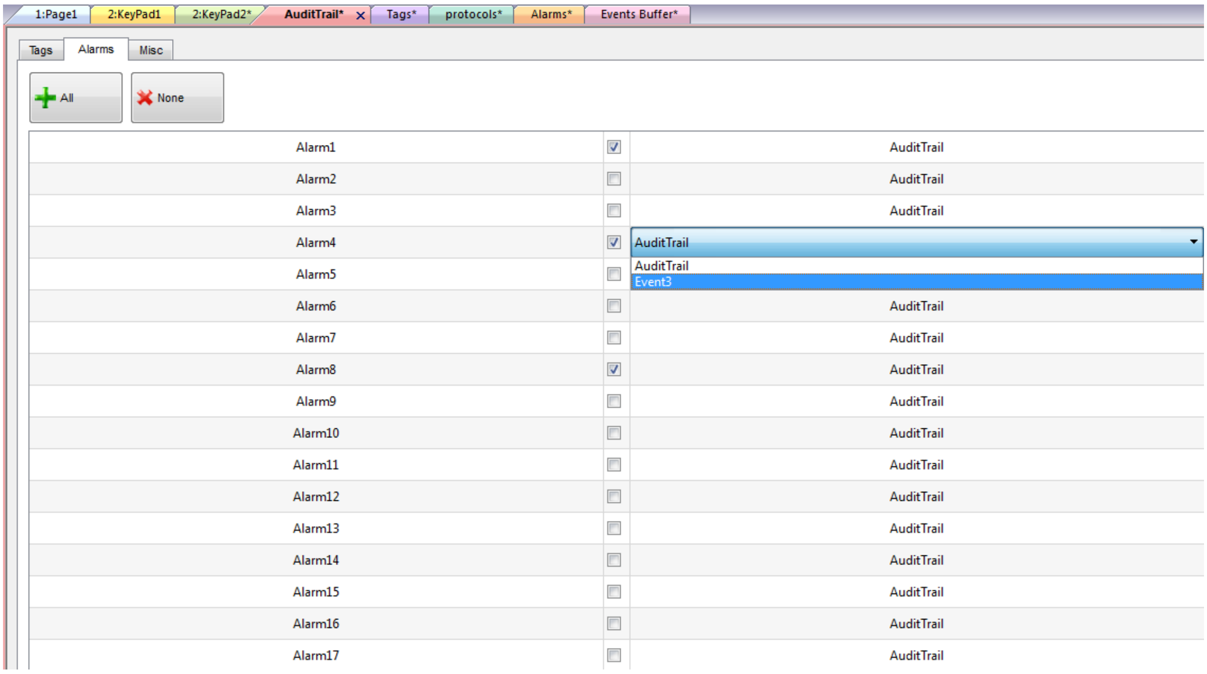
Speed	☑	AuditTrail
Torque	☑	AuditTrail
Temperature	☑	AuditTrail
Viscocity	□	AuditTrail
Tag5	□	AuditTrail
Tag6	□	AuditTrail
Tag7	□	AuditTrail
Tag8	□	AuditTrail
Tag9	□	AuditTrail
Tag10	☑	AuditTrail
Tag11	□	AuditTrail
Tag12	□	AuditTrail
Tag13	□	AuditTrail
Tag14	□	AuditTrail
Tag15	□	AuditTrail
Tag16	□	AuditTrail
Tag17	□	AuditTrail
Tag18	□	AuditTrail
Tag19	□	AuditTrail

Figure 154

In the Audit Trail editor, all the Tags are available for selection. You can select only the Tags to be monitored by Audit Trail. For each selected Tag, the Audit Trail will record the write operations to that Tag, together with the time stamp and user that activates the write.

14.4 Configure Alarms in the Audit Trail

Like Tags, you can specify the alarms to be monitored by Audit Trail. Double click the Audit Trail from the project workspace and click on the Alarms tab. Select the alarms you want to log in for during the Audit Trail. The Audit Trail for alarm will record and acknowledge the operation done by the logged-in user.

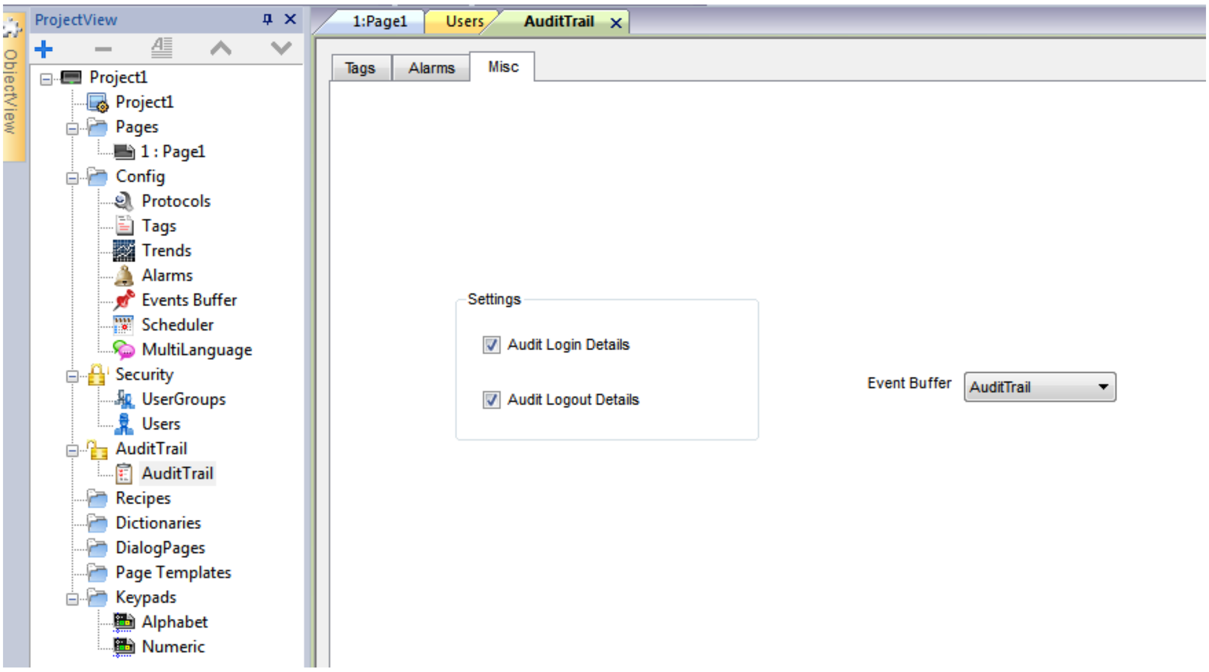


Alarm1	☑	AuditTrail
Alarm2	□	AuditTrail
Alarm3	□	AuditTrail
Alarm4	☑	AuditTrail AuditTrail Event3
Alarm5	□	
Alarm6	□	AuditTrail
Alarm7	□	AuditTrail
Alarm8	☑	AuditTrail
Alarm9	□	AuditTrail
Alarm10	□	AuditTrail
Alarm11	□	AuditTrail
Alarm12	□	AuditTrail
Alarm13	□	AuditTrail
Alarm14	□	AuditTrail
Alarm15	□	AuditTrail
Alarm16	□	AuditTrail
Alarm17	□	AuditTrail

Figure 155

14.5 Configure Login or Logout Details in Audit Trail.

Audit Trail can record information about user login and user logout events. These settings are available in the Misc. tab of the Audit Trail.



Settings

☑ Audit Login Details

☑ Audit Logout Details

Event Buffer: AuditTrail

Figure 156

14.6 Viewing Audit Trails in Run-time

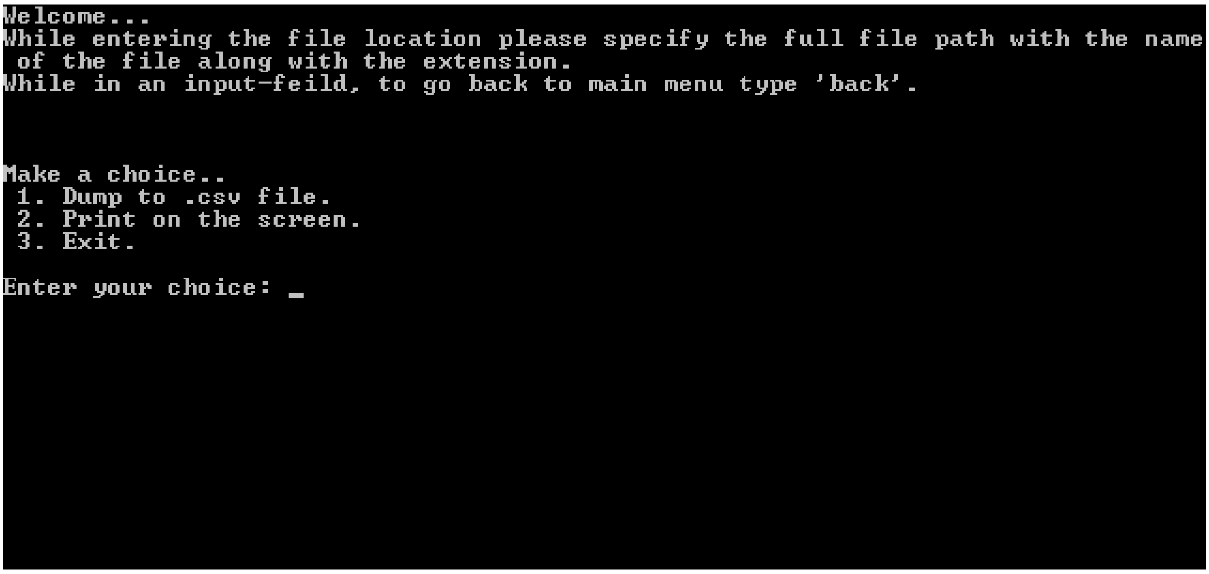
Audit Trail data cannot be displayed on Runtime, they are only available in the exported data file.

14.7 Viewing Audit Trails in MS Dos Prompt

An executable named 'ArchiveEdit.exe' has been provided to view the audit data.
The executable utility is available in the JMobile installation folder

To print the audit data in the screen, execute the ArchiveEdit.exe.
For example:
C:\Program Files\Exor\JMobile Suite\utils

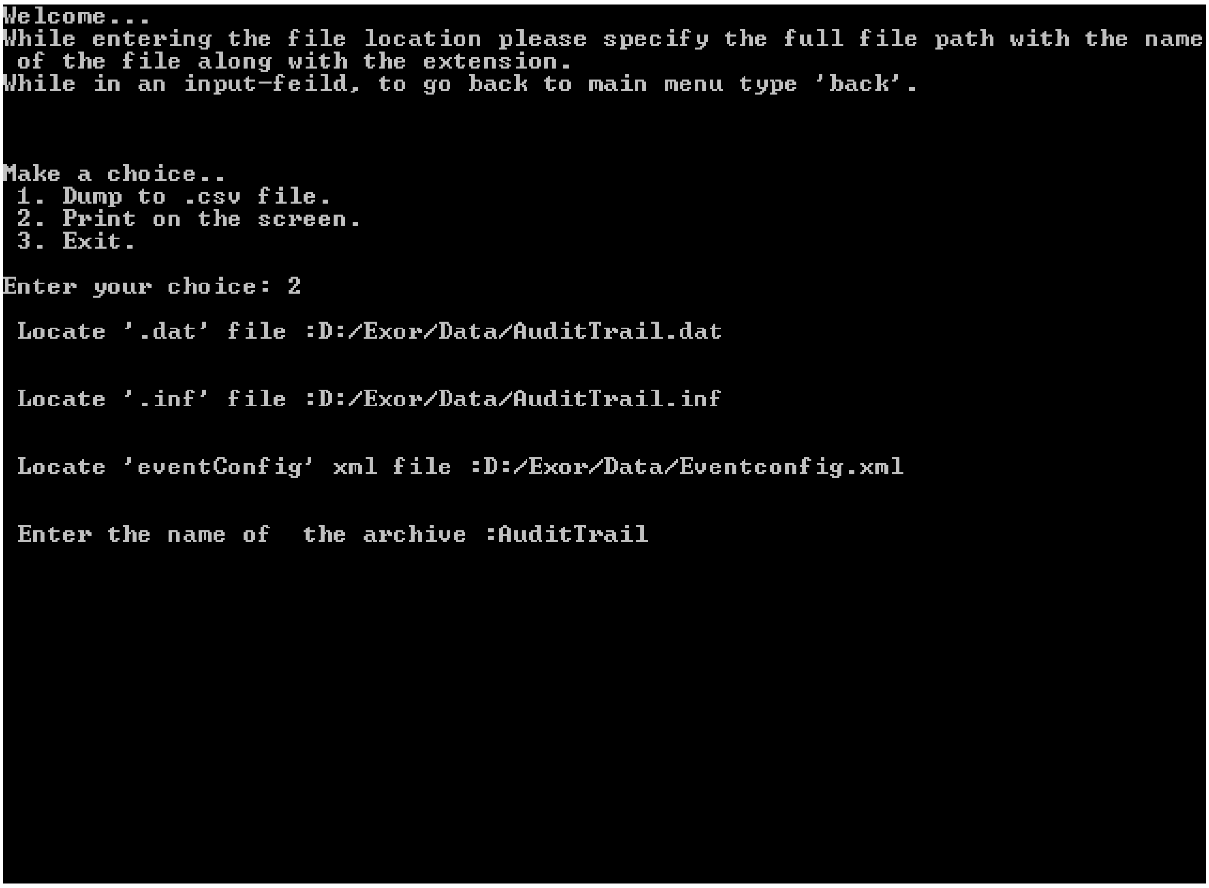
Run the exe via the MS Dos command line, with the command line argument identical to that provided below, or launching the executable file from the folder in Windows:



```
Welcome...
While entering the file location please specify the full file path with the name
 of the file along with the extension.
While in an input-feild, to go back to main menu type 'back'.



Make a choice..
 1. Dump to .csv file.
 2. Print on the screen.
 3. Exit.

Enter your choice: _
```

Figure 157

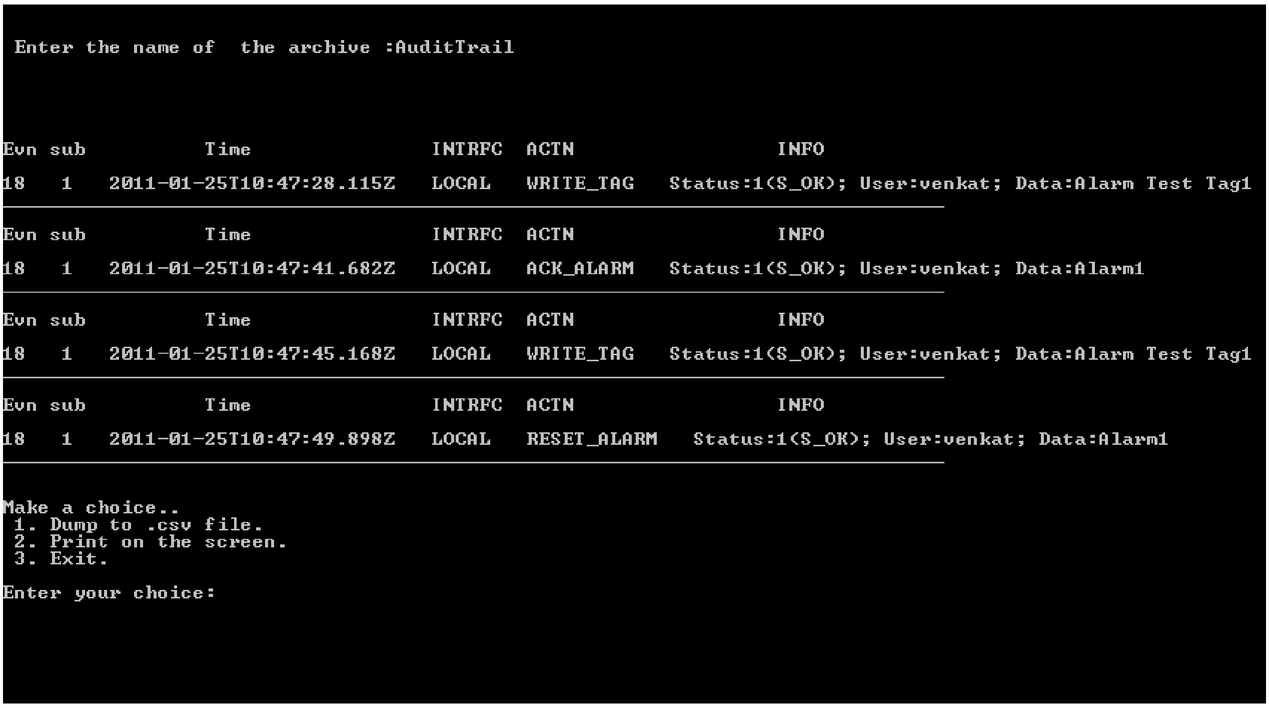
Enter your choice. If you want to see the audit data on the screen itself, type 2 hit Enter; then type the location of the dat file, inf file, and project eventconfig.xml file.



```
Welcome...
While entering the file location please specify the full file path with the name
 of the file along with the extension.
While in an input-feild, to go back to main menu type 'back'.


Make a choice..
 1. Dump to .csv file.
 2. Print on the screen.
 3. Exit.

Enter your choice: 2

 Locate '.dat' file :D:/Exor/Data/AuditTrail.dat


 Locate '.inf' file :D:/Exor/Data/AuditTrail.inf


 Locate 'eventConfig' xml file :D:/Exor/Data/Eventconfig.xml


 Enter the name of  the archive :AuditTrail
```

Figure 158

Provide the location of the Audit dat and inf files. The Audit dat and inf files will be available in the DATA folder in the JMobile Runtime folder.
Before these become available for copy, the Audit Trail buffer needs to be dumped to a specific folder; a dump can also be completed via the USB pen driver, plugged for the scope to the unit (that can be plugged to the USB port). To dump the Audit Trail buffer, use the DumpEventArchive action explained in chapter 7.7.8, specifying, upon programming the action, that this configuration file must be exported, as well (set to true the property “DumpConfigFile").
A sample of the Audit data you will see on the screen is shown in the following picture.

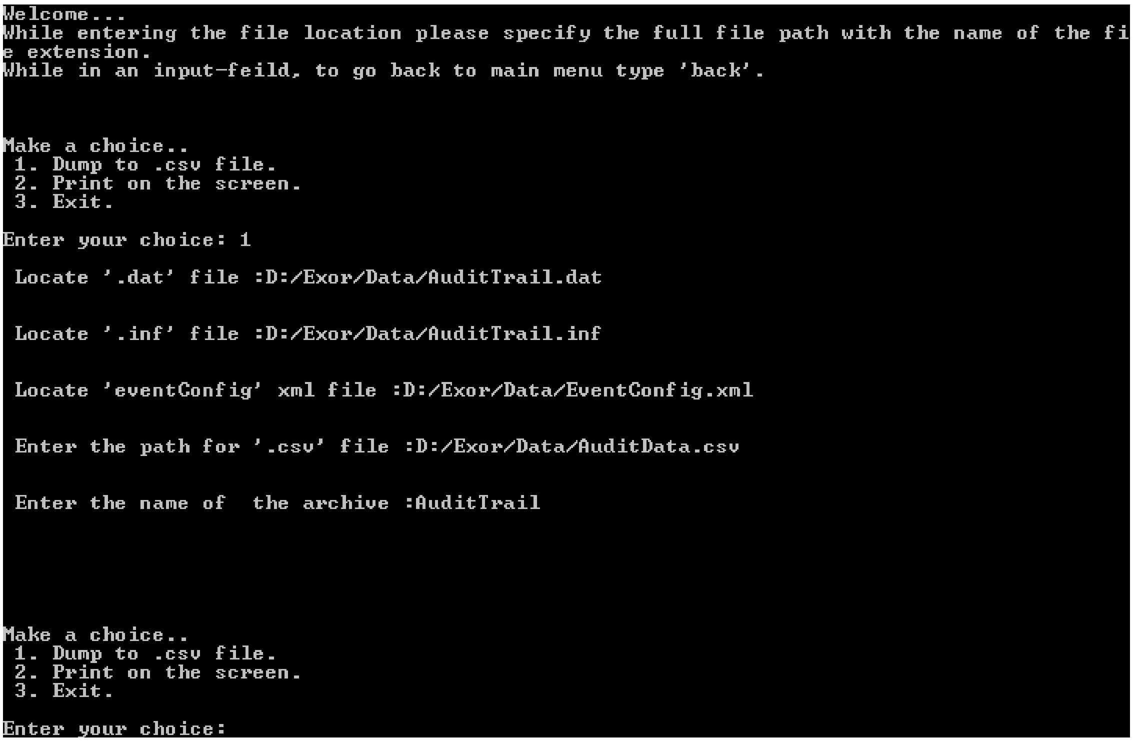


```
 Enter the name of  the archive :AuditTrail



Evn sub          Time               INTRFC  ACTN                 INFO
18   1   2011-01-25T10:47:28.115Z   LOCAL   WRITE_TAG   Status:1(S_OK); User:venkat; Data:Alarm Test Tag1
________________________________________________________________________________
Evn sub          Time               INTRFC  ACTN                 INFO
18   1   2011-01-25T10:47:41.682Z   LOCAL   ACK_ALARM   Status:1(S_OK); User:venkat; Data:Alarm1
________________________________________________________________________________
Evn sub          Time               INTRFC  ACTN                 INFO
18   1   2011-01-25T10:47:45.168Z   LOCAL   WRITE_TAG   Status:1(S_OK); User:venkat; Data:Alarm Test Tag1
________________________________________________________________________________
Evn sub          Time               INTRFC  ACTN                 INFO
18   1   2011-01-25T10:47:49.898Z   LOCAL   RESET_ALARM   Status:1(S_OK); User:venkat; Data:Alarm1
________________________________________________________________________________

Make a choice..
 1. Dump to .csv file.
 2. Print on the screen.
 3. Exit.

Enter your choice:
```

Figure 159

14.8 Exporting Audit Trail as CSV File

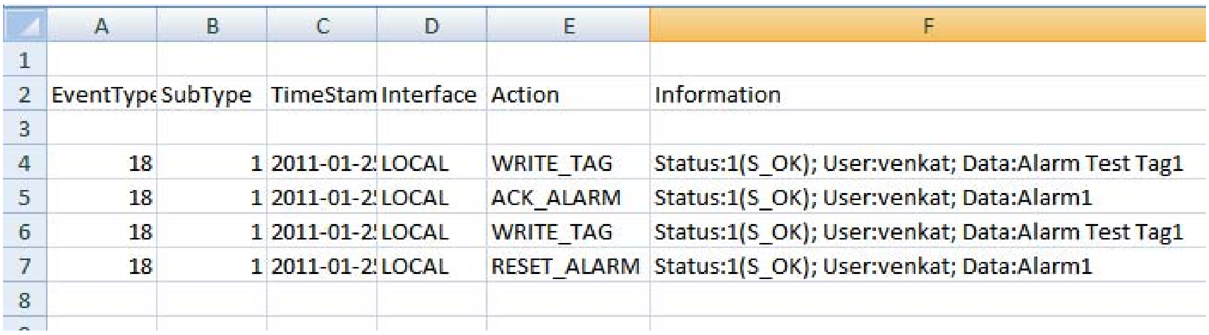
You can convert the audit data to a “.csv" file.
To convert the audit data, execute the ArchiveEdit.exe. Then, type ‘1’ into the command prompt and hit Enter. Enter the locations of the Dat file, inf file, project event config .xml file and csv file.



```
Welcome...
While entering the file location please specify the full file path with the name of the fi
e extension.
While in an input-feild, to go back to main menu type 'back'.


Make a choice..
 1. Dump to .csv file.
 2. Print on the screen.
 3. Exit.

Enter your choice: 1

 Locate '.dat' file :D:/Exor/Data/AuditTrail.dat


 Locate '.inf' file :D:/Exor/Data/AuditTrail.inf


 Locate 'eventConfig' xml file :D:/Exor/Data/EventConfig.xml


 Enter the path for '.csv' file :D:/Exor/Data/AuditData.csv


 Enter the name of  the archive :AuditTrail





Make a choice..
 1. Dump to .csv file.
 2. Print on the screen.
 3. Exit.

Enter your choice:
```

Figure 160

A sample of the exported csv file is shown in the following picture:



	A	B	C	D	E	F
1						
2	EventType	SubType	TimeStam	Interface	Action	Information
3						
4	18	1	2011-01-2	LOCAL	WRITE_TAG	Status:1(S_OK); User:venkat; Data:Alarm Test Tag1
5	18	1	2011-01-2	LOCAL	ACK_ALARM	Status:1(S_OK); User:venkat; Data:Alarm1
6	18	1	2011-01-2	LOCAL	WRITE_TAG	Status:1(S_OK); User:venkat; Data:Alarm Test Tag1
7	18	1	2011-01-2	LOCAL	RESET_ALARM	Status:1(S_OK); User:venkat; Data:Alarm1
8						

Figure 161

15 Custom Keypad

Keypads are used for data entry operations. In Studio there are three keypads provided by default. Numeric, Alphabet and Up-Down are shown in the following pictures:

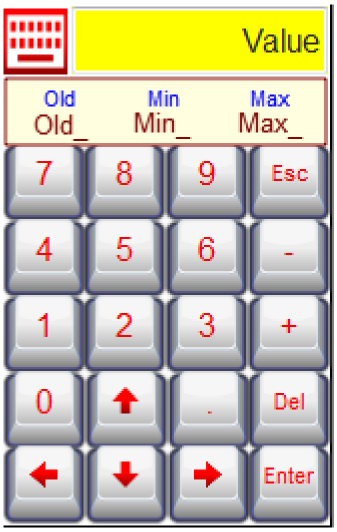


Figure 162 – Numeric keypad

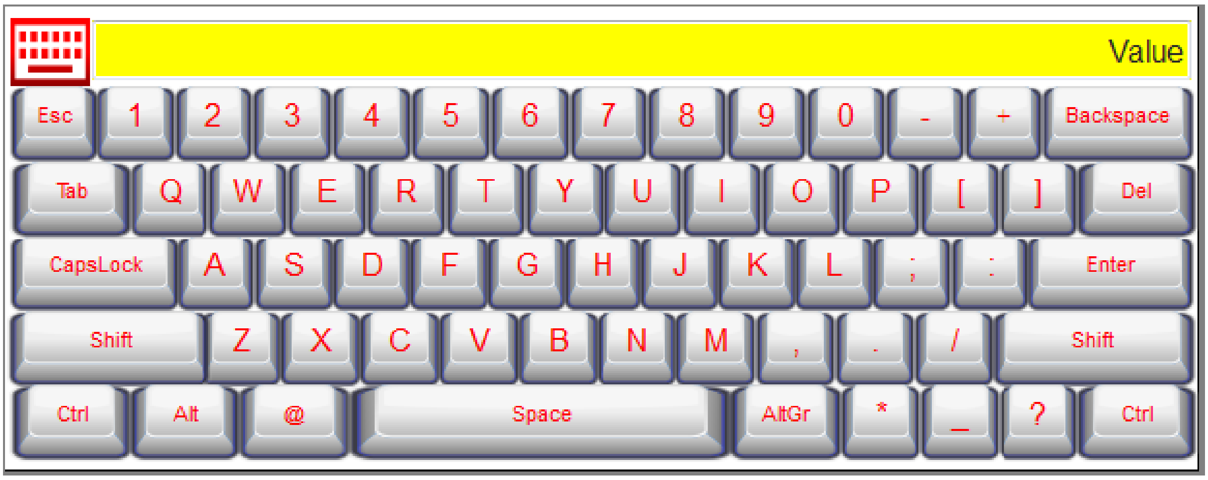

Value
Esc 1 2 3 4 5 6 7 8 9 0 - + Backspace
Tab Q W E R T Y U I O P [ ] Del
CapsLock A S D F G H J K L ; : Enter
Shift Z X C V B N M , . / Shift
Ctrl Alt @ Space AltGr * _ ? Ctrl

Figure 163 - Alphabet keypad

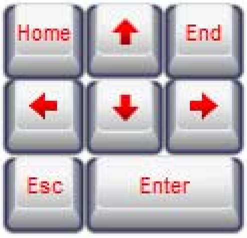


Figure 164 - Up Down Keypad

15.1 Creating and Using a Custom Keypad.

Right click on the Keypad folder, found under the “Project View" section of the workspace. A context menu, as shown in the below picture, will be displayed:

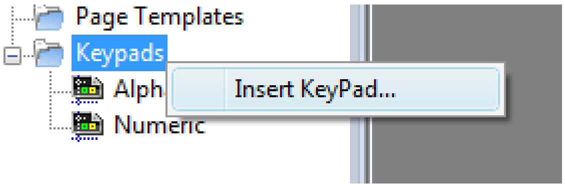
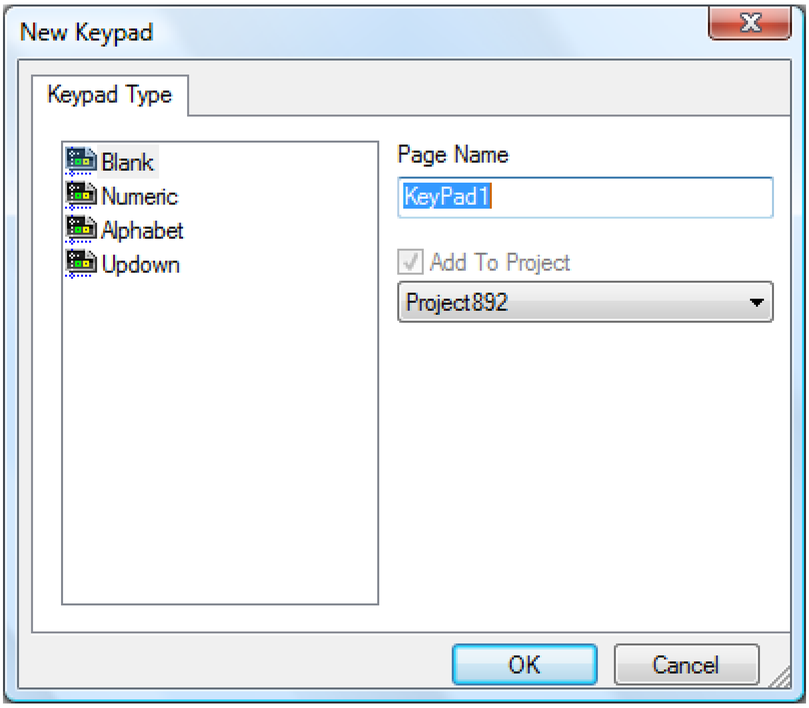


Figure 165

Clicking on the “Insert Keypad" will generate a pop-up with the “New Keypad" dialog, as shown below.



Page Name: KeyPad1

☑ Add To Project

Project892

Figure 166

The user can select any of the available keypads that are provided in the project template (the list shown on the left side) to create a custom keypad. If you need to create a keypad from scratch, then select the “Blank" option. This will insert a Blank Keypad, as shown below:

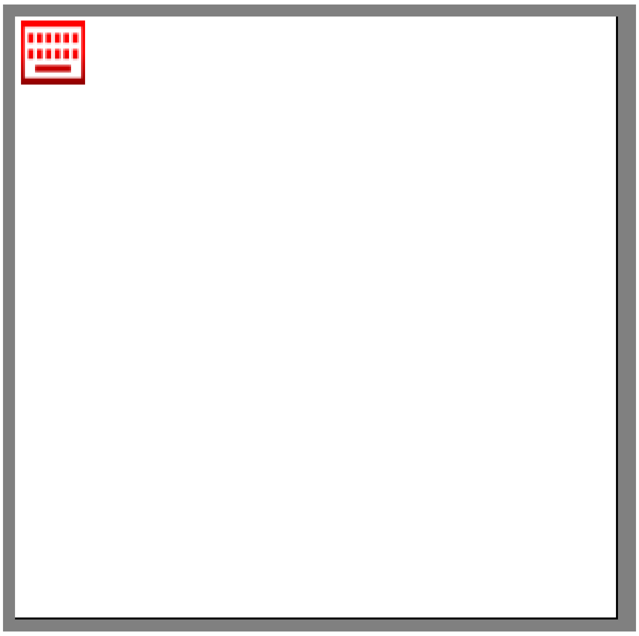


Figure 167.

The user can then use the Widgets available from the Keypad Widget gallery (as shown in the picture below) to create the custom keypad.

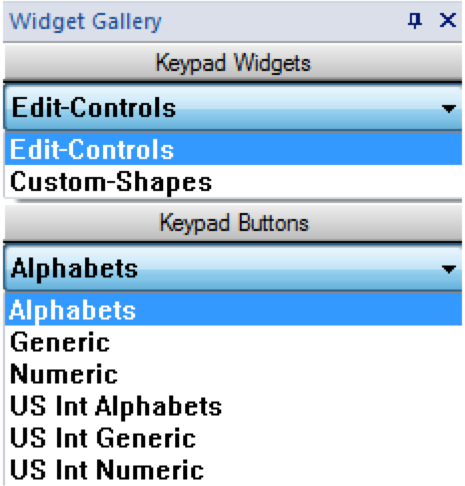


Figure 168

A sample custom-created keypad is shown below.
Newly created keypads will be saved in the project folder, together will all other project files.

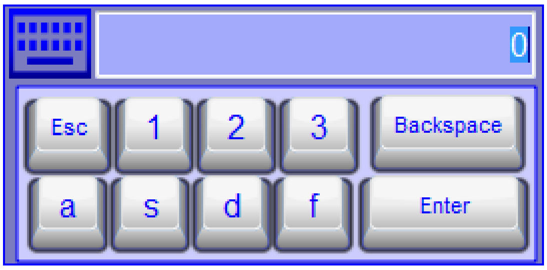
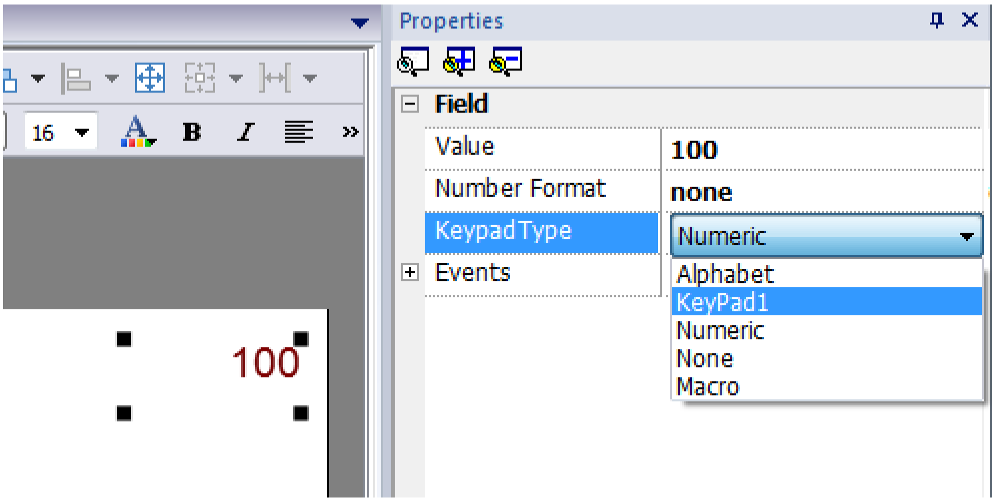


Figure 169

Once the custom keypad is created, it may be used for any specific field where the Keyboard Type property has been properly set by selecting the corresponding keypad from the property "Keypad Type" in the property pane (as shown below).



Field

Value	100
Number Format	none
KeypadType	Numeric
Events	

Figure 170

The Up-Down keypad is mainly used for moving cursors in Widgets and supporting this function. An example is the "Control List" is shown in the following picture:

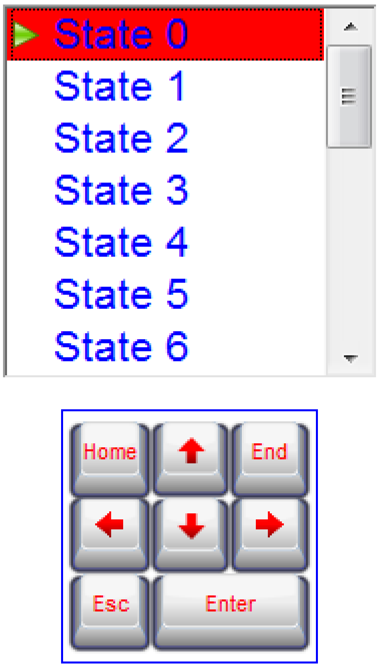


Figure 171

15.2 Deleting or Renaming Custom Keypad

Right click on the Keypad folder, found in the "Project View" section of the workspace. A context menu (as shown in the below figure) will be displayed.

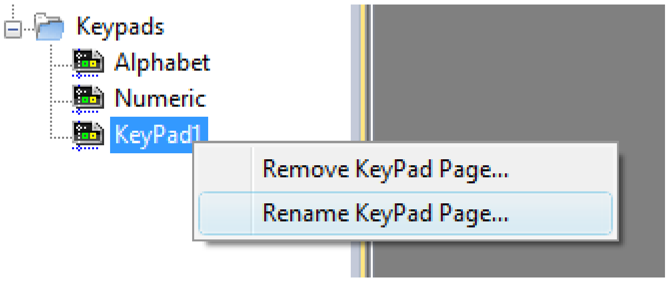


Figure 172

The user can choose the “Remove Keypad Page" option to remove the keypad page from the project, or the “Rename Keypad Page" option to rename the keypad.

16 Special widgets

16.1 RSS Feed widget

The RSS (**R**eally **S**imple **S**yndication) Feed widget allows to display on the screen your favourite RSS feeds directly from the internet.
The widget is available in the widget gallery under the Advanced category. In page it looks like shown in Figure 173.

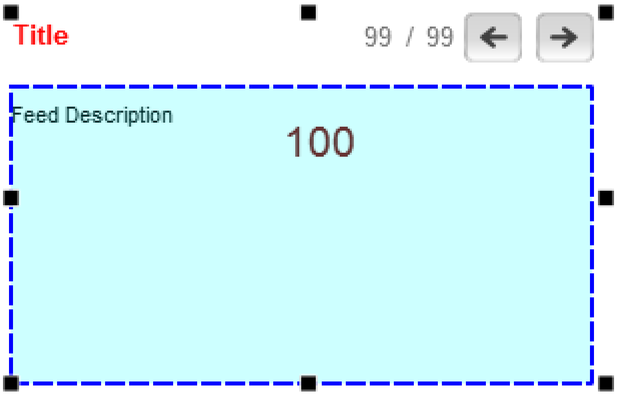
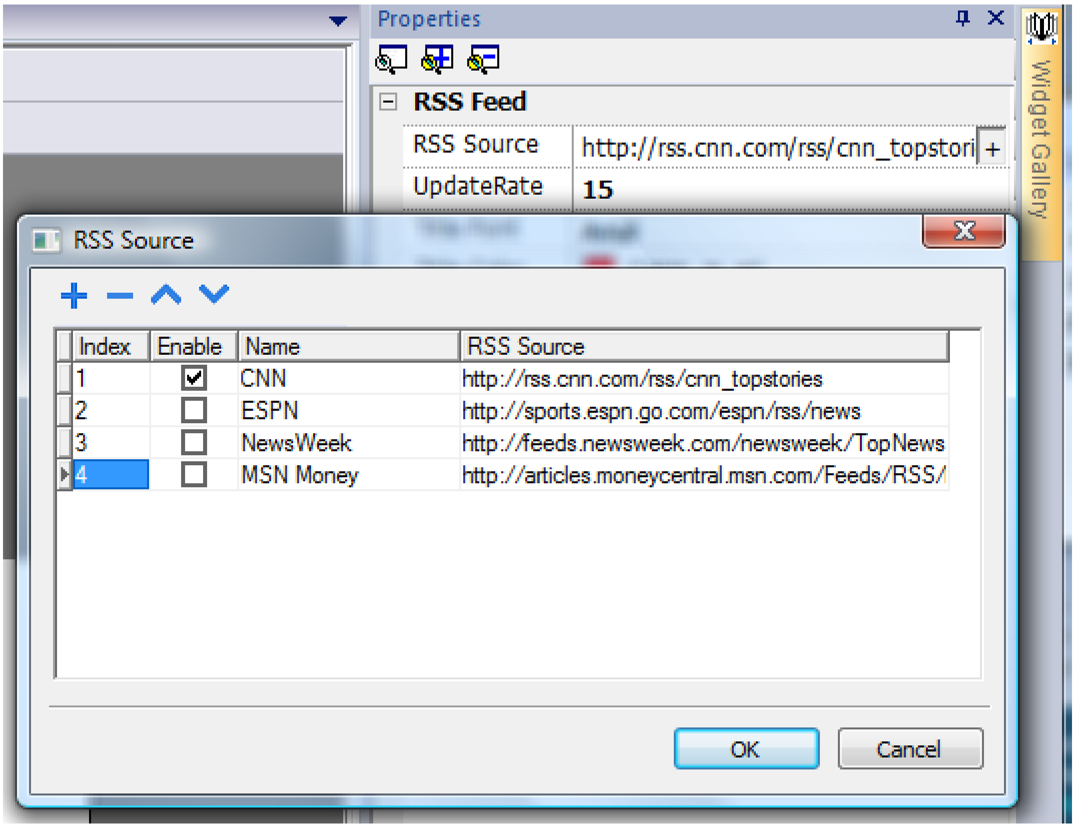


Figure 173

The RSS Feed widget has two important properties: RSS Source allows you to specify the feed URL, UpdateRate allows to specify the refresh time.
Properties are shown in Figure 174.

Note: Feeds sources are fixed and cannot be changed run time



RSS Feed

RSS Source	http://rss.cnn.com/rss/cnn_topstori
UpdateRate	15

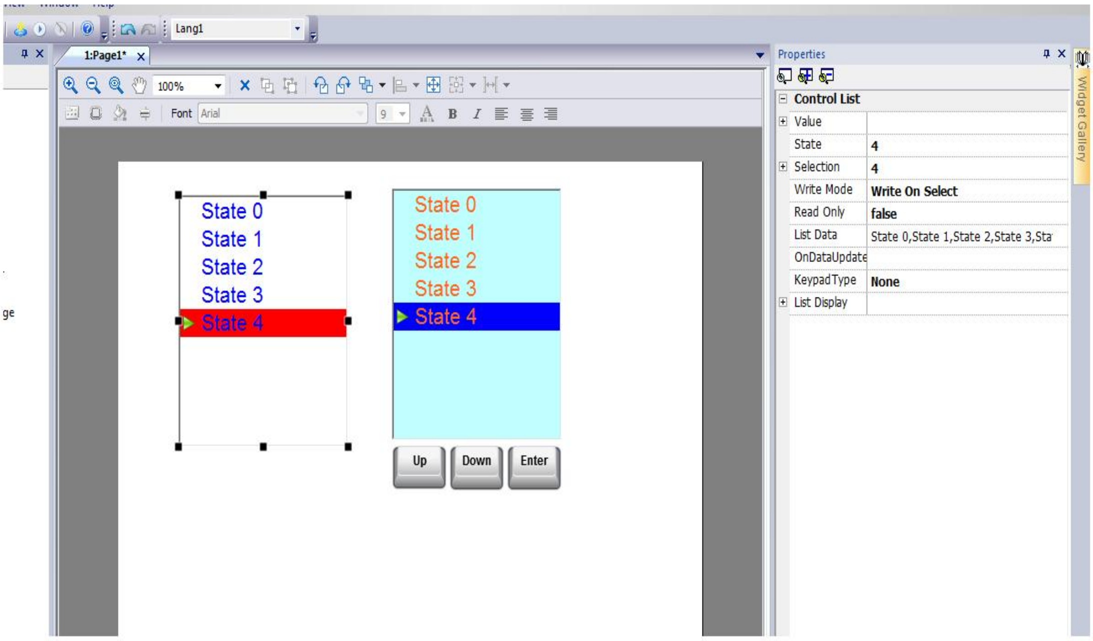
Index	Enable	Name	RSS Source
1	☑	CNN	http://rss.cnn.com/rss/cnn_topstories
2	□	ESPN	http://sports.espn.go.com/espn/rss/news
3	□	NewsWeek	http://feeds.newsweek.com/newsweek/TopNews
4	□	MSN Money	http://articles.moneycentral.msn.com/Feeds/RSS/

Figure 174

16.2 Control List Widget

JMobile provides Control List Widgets, a convenient way to represent the status associated to a particular process.
'States' can be added to the control list by selecting Add, and they can be removed by selecting Remove, which are both items from the property pane; in making this selection, it is possible to select any of the states that are listed.
The Control List Widget is available in the Widget Gallery under the Advanced category.

There are two types of control lists. One is a control list group, in which the up and down buttons are present on the control list itself. The state can be selected with the up and down buttons. The other type of control list has no pre-configured buttons in the group.

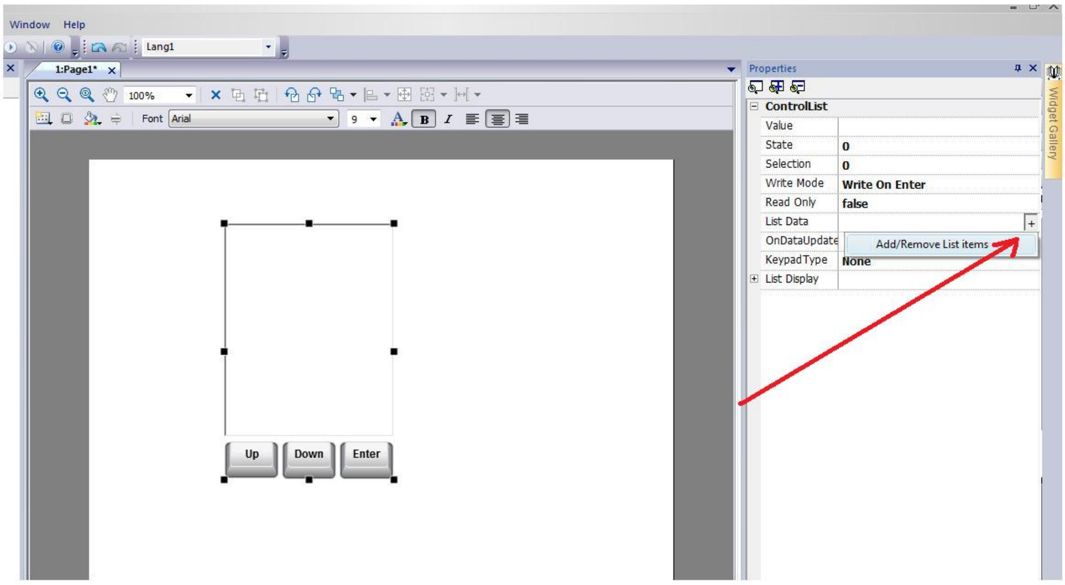


Control List	
Value	
State	4
Selection	4
Write Mode	Write On Select
Read Only	false
List Data	State 0,State 1,State 2,State 3,Sta
OnDataUpdate	
KeypadType	None
List Display	

Figure 175

16.2.1 State

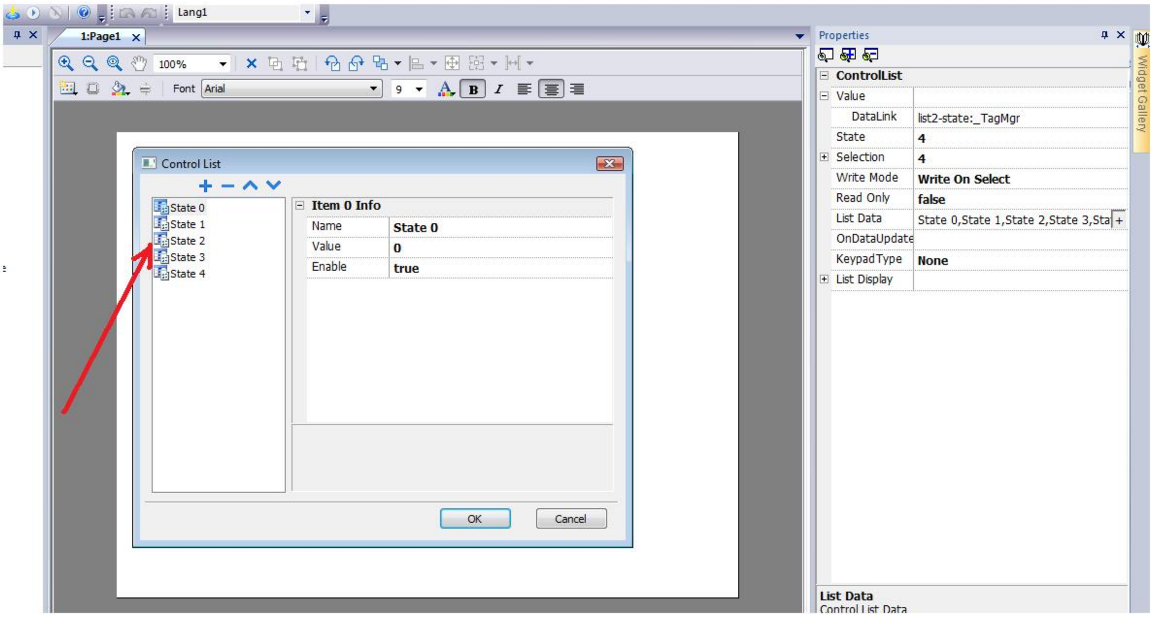
States are added by selecting Add/Remove List Items in the property pane. Any value can be assigned to a State; activating the State will result in a write operation to the Tag, which has been linked to the Value property of the Control List Widget.



ControlList

Value	
State	0
Selection	0
Write Mode	Write On Enter
Read Only	false
List Data	
OnDataUpdate	
KeypadType	None
List Display	

Figure 176



Item 0 Info

Name	State 0
Value	0
Enable	true

ControlList

Value	
DataLink	list2-state:_TagMgr
State	4
Selection	4
Write Mode	Write On Select
Read Only	false
List Data	State 0,State 1,State 2,State 3,Sta
OnDataUpdate	
KeypadType	None
List Display	

List Data
Control List Data

Figure 177

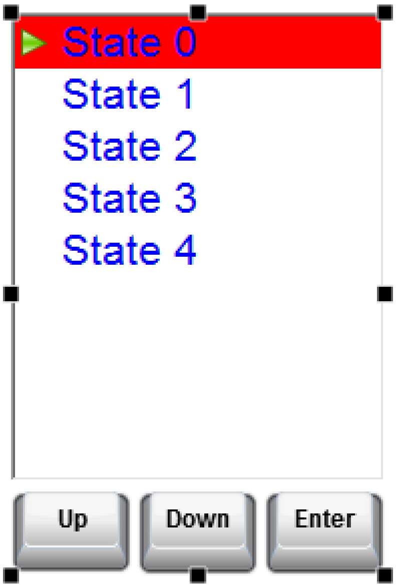


Figure 178

16.2.2 Selection

Selection shows which status is currently selected, and will appear as a highlight cursor moving up and down (according to the use of the defined keys). The Selection property can be attached, as well, to a Tag.
The small triangle on the left side of the list tells you what the current status is.
There are two modes for the control list:
Write on Select
Write on Enter

16.2.3 Write on Select

On Write on Select, the value will automatically be written when the status is selected.

16.2.4 Write on Enter

First select the state, and then press the enter key to write the status value to the Tag.

17 Working with Custom Widgets in JMobile

JMobile Studio includes a large library of predefined Widgets and graphic images. The symbol library includes predefined dynamic Widgets (such as buttons, lights, gauges, switches, Trends, Recipes, and dialog items), as well as static images (such as shapes, pipes, tanks, motors, etc.). With the symbol library, you can simply drag and drop a symbol on to the page, and then size it, move it, rotate it or transform it any way you want. All symbols are vector based, so they look good at any size.

Custom Widgets are Widgets created by the user and based on the various existing Widgets in the gallery. This Chapter describes how to create a customized Widget and assign it properties.

The advantage of the custom Widget that it builds a Widget out of any complexity, including several elements, and can decipher which properties have to be published and made available in the “custom Widget" Property pane.

17.1 Creating Custom Widget

The following steps describe how to create a custom Widget:

1. Select all the Widgets you want as part of the custom Widget and make them one group.
2. Right click on the group, and select "Convert To Widget" from the context menu.

A Conversion to Widget dialog is displayed below.

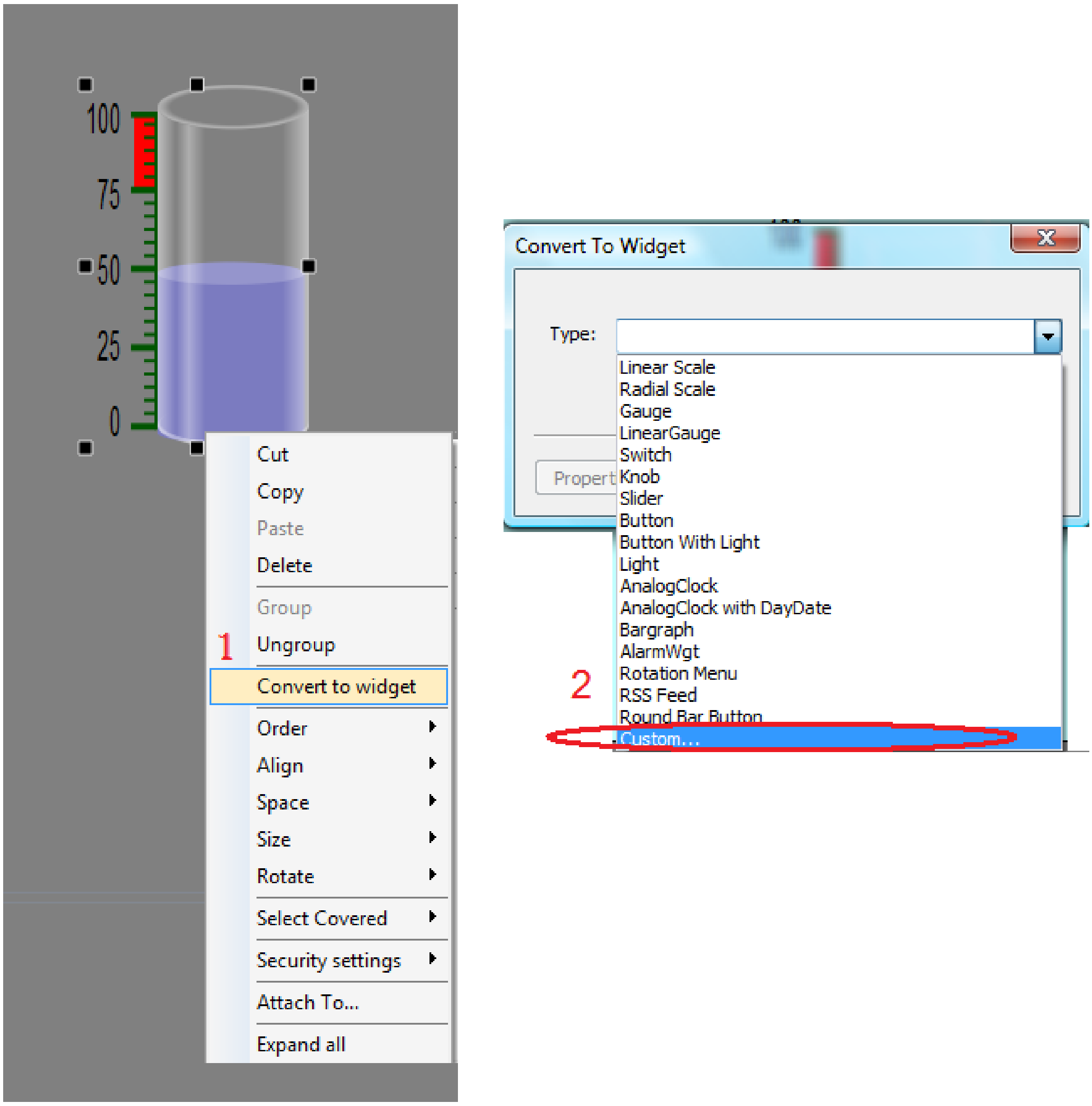
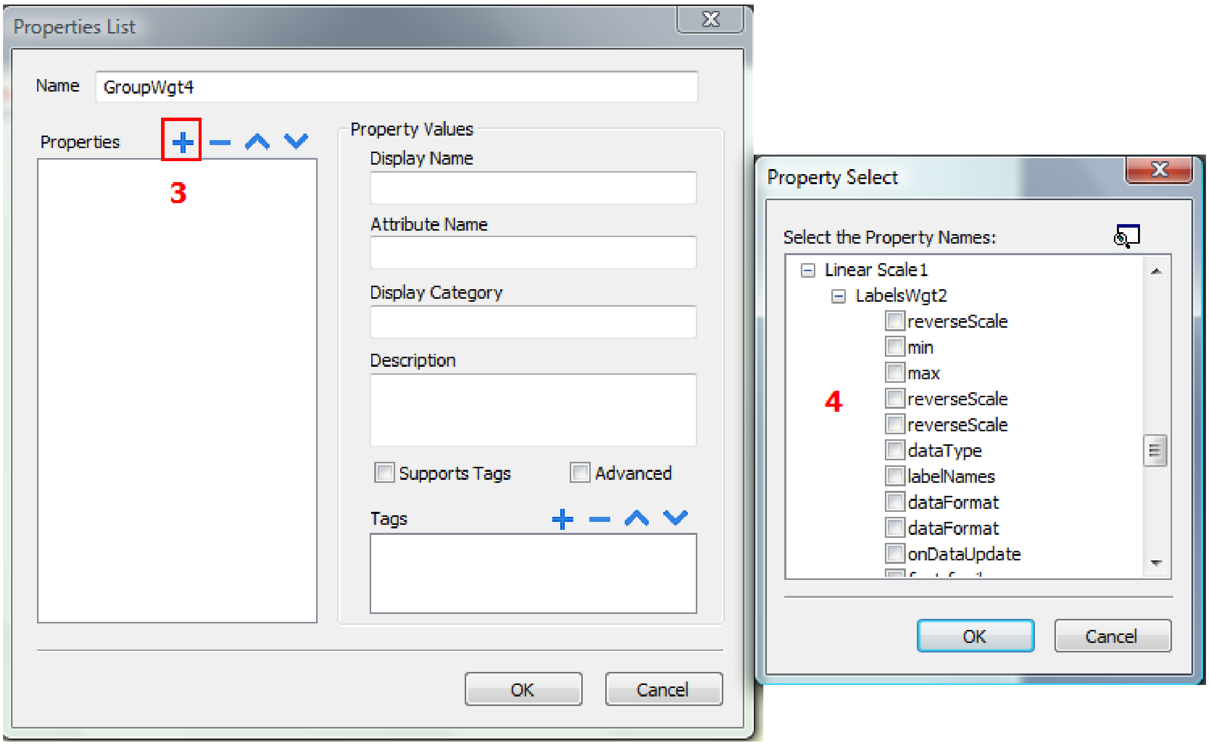


Figure 179

At this point, you can select existing custom Widget types (such as Knobs, Button with indicator light, etc.), or you can select "Custom" to create a new custom Widget type.

17.2 Adding the Properties

After creating the custom Widget, the next step is to add the property that will be published in the custom Widget property pane. The “Property Select" dialog shows all the applicable properties for the grouped Widget; this is basically a list of all the properties of all Widgets grouped together. You can select the properties by clicking the corresponding check box.



Name: GroupWgt4

Figure 180

Enter the name of the Custom Widget. This is the name that will appear in the Property view. The next step is to select the properties that will be displayed in the Property view. Click on the ‘+’ button above the ‘Properties’ list box, and a Property Select dialog will be displayed.

Note: The ConvertToWidget dialog shows “standard" custom Widget types. These types are defined in the gallery. The dialog, however, does not show types that are specifically created for a project.

17.2.1 Display Name

The 'Display Name' and 'Description' are names that will be shown in the Property view. You can change this value to set the information for each custom Widget property.

17.2.2 Attribute Name

The 'Attribute Name' is the name exposed by JMobile, to JavaScript functions and Attach Tag dialog. The default property name has the form 'WidgetType.name'; 'WidgetType' is the type of Widget; and 'name' is the Attribute Name. If you have more than one Widget of the same type, the Widget type name will be WidgetType01, WidgetType02, etc.

17.2.3 Display Category

The 'Display Category' is the category or group of the property in the Property view. All properties in the same category are shown together in the Property view. This allows you to organize the properties in the view. The Display Category allows you to view by category group, by clicking on either the Collapse or Expand button. For example, you can declare position properties, like the X coordinate, height, width properties in a single display category called Position.

17.2.4 Description

The Description property allows you to define the description and comments within the property; the information will be displayed in the property pane.

17.2.5 Advanced

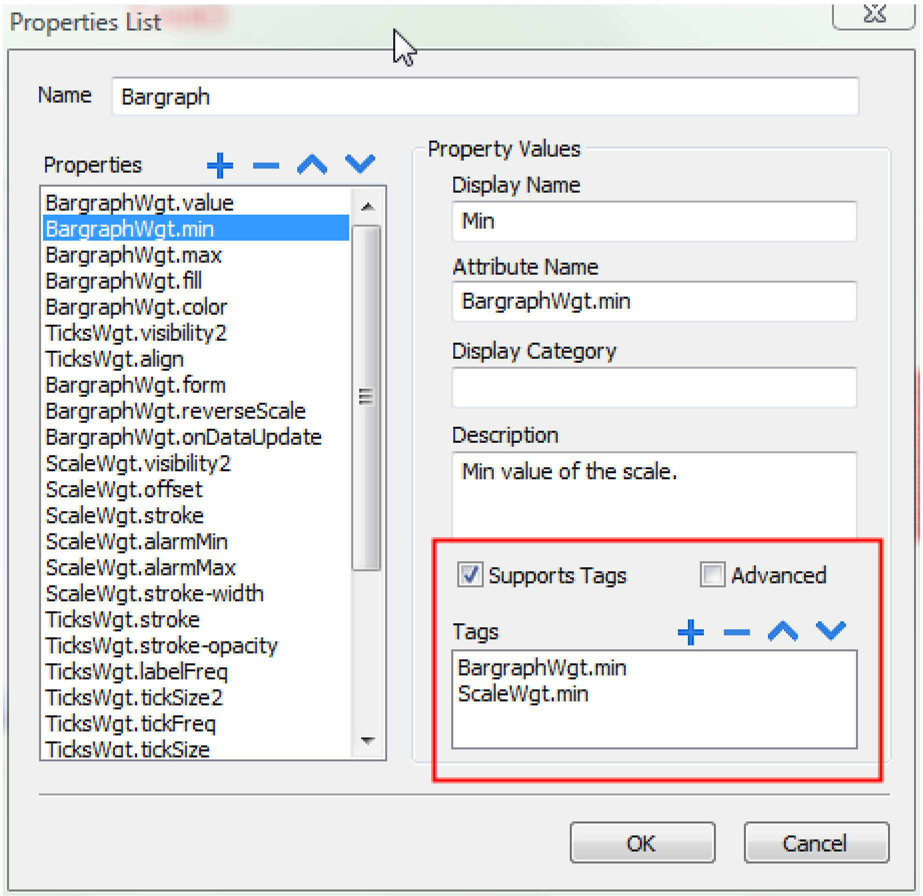
The properties are shown in either the “Normal" or “Advanced" mode. The “Advanced" check box allows you to specify whether each property should appear in the advanced, or in the simple property pane view mode.

17.2.6 Supports Tag

The “Supports Tags" checkbox must be marked if the property supports the “Attach to" attribute.

17.2.7 Tags

The “Tags" list box indicates the internal Tag name for the Widget. This internal Tag name is typically the same as the attribute name, however, this is not always the case. You can expose a different attribute name for your custom Widget. The Tag list is also used to combine Tags.
If you want to combine two or more properties into one, select the primary property in the Property List and click on the ‘+’ button above the Tags list box. The Property Select dialog will be displayed, and you can select the properties that should be combined. Note that this dialog is only shows the properties that should be combined (not all properties are shown in the Properties list). For example, to combine the 'min' property of the scale Widget and Bargraph Widget, click on the NeedleWgt min property and click on the BargraphWgt min property from the Property Select dialog. Click the OK button. Both attributes will be shown in the Tags list box, as shown in Figure 181.



Name: Bargraph

Properties:
BargraphWgt.value
BargraphWgt.min
BargraphWgt.max
BargraphWgt.fill
BargraphWgt.color
TicksWgt.visibility2
TicksWgt.align
BargraphWgt.form
BargraphWgt.reverseScale
BargraphWgt.onDataUpdate
ScaleWgt.visibility2
ScaleWgt.offset
ScaleWgt.stroke
ScaleWgt.alarmMin
ScaleWgt.alarmMax
ScaleWgt.stroke-width
TicksWgt.stroke
TicksWgt.stroke-opacity
TicksWgt.labelFreq
TicksWgt.tickSize2
TicksWgt.tickFreq
TicksWgt.tickSize

Property Values
Display Name: Min
Attribute Name: BargraphWgt.min
Display Category:
Description: Min value of the scale.
☑ Supports Tags ☐ Advanced
Tags:
BargraphWgt.min
ScaleWgt.min

Figure 181

You can arrange the order of the properties by clicking on the up or down button in the Property List.
To remove a property, select the property name and click on the delete button.
When a property is selected in the Property List, the property information is shown in the dialog.

17.3 Editing Custom Properties

If you want to change the properties of a Custom Widget after it has been created, you can simply right click on the Widget in the Page editor and select the "Custom Properties" menu item from the context menu. The Custom Properties dialog will be displayed and you can change the properties.

18 Sending E-mail From JMobile

The Send e-mail is an Action that can be programmed as trigger action for an alarm or for a timed scheduled action.
You can include Tags in the e-mail Body; upon executing the action, Tags value, instantaneously, will be detected by the system and included in the message body.

18.1 Send Email script

Send E-mail action is available under the script tab of the Action list for alarms and scheduler actions.

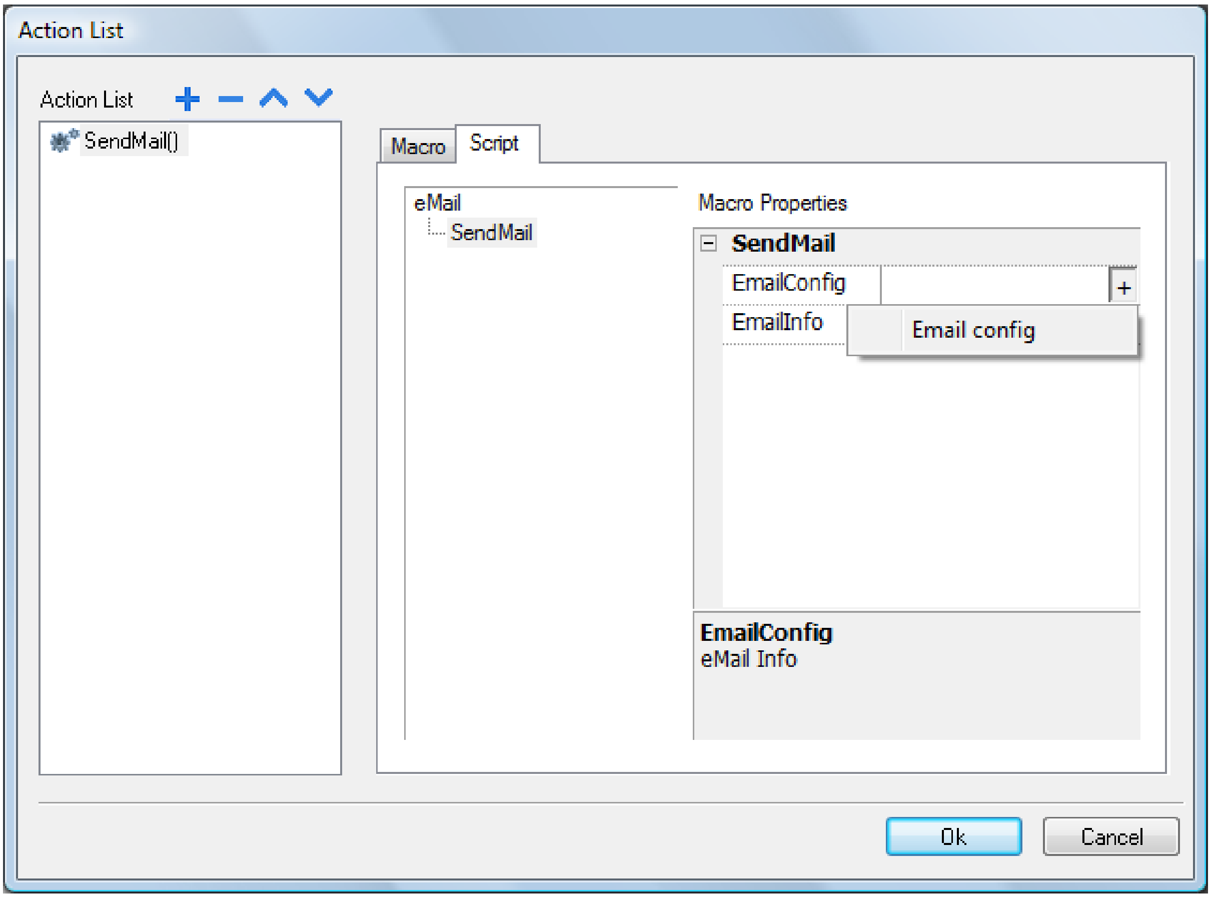
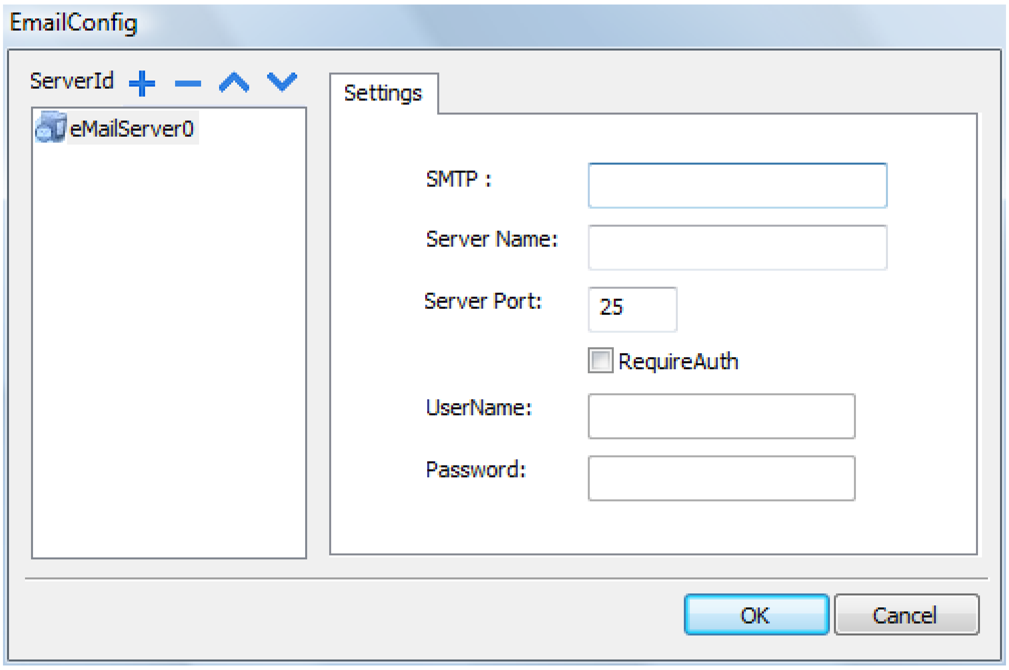


Figure 182

18.2 Configure E-mail Server

To configure the e-mail Server, you need to provide the following information: SMTP Server Name, SMTP Port, Sender, User Name and (if Authentication is required) Password.



SMTP :

Server Name:

Server Port: 25

□ RequireAuth

UserName:

Password:

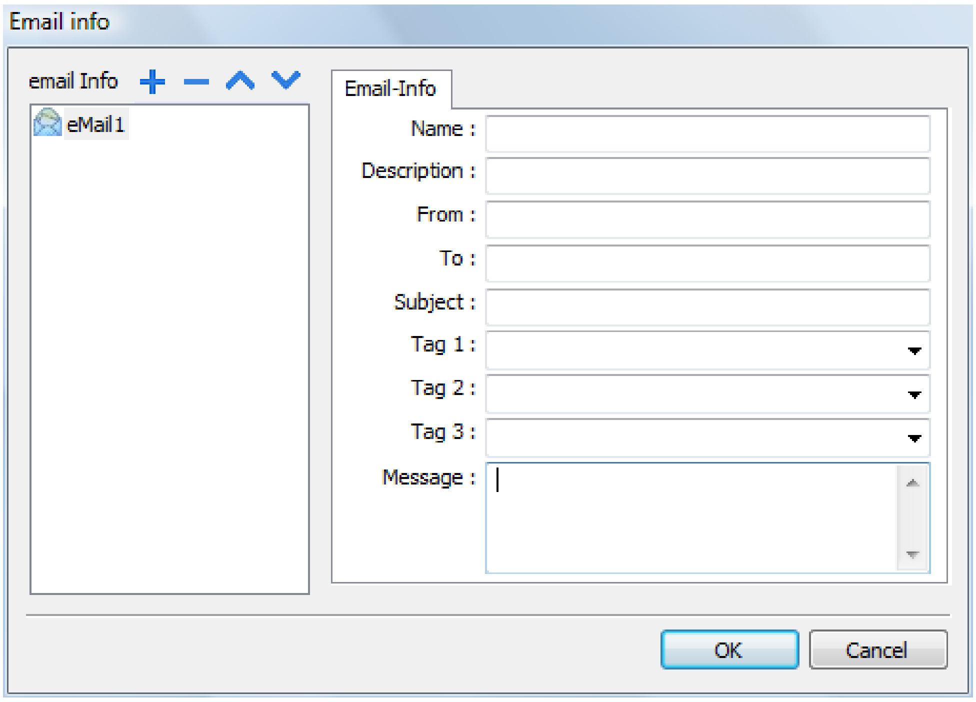
Figure 183

Note: You can add more E-mail servers by clicking the “+" button on the left hand side.

When you are writing, the system does not yet support any security login mechanism, such as SSL.

18.3 Configure E-mail Accounts

In the e-mail info, set recipient e-mail addresses. If you want to send to more recipients, separate the e-mail address with a semi-colon “;".



Name :

Description :

From :

To :

Subject :

Tag 1 :

Tag 2 :

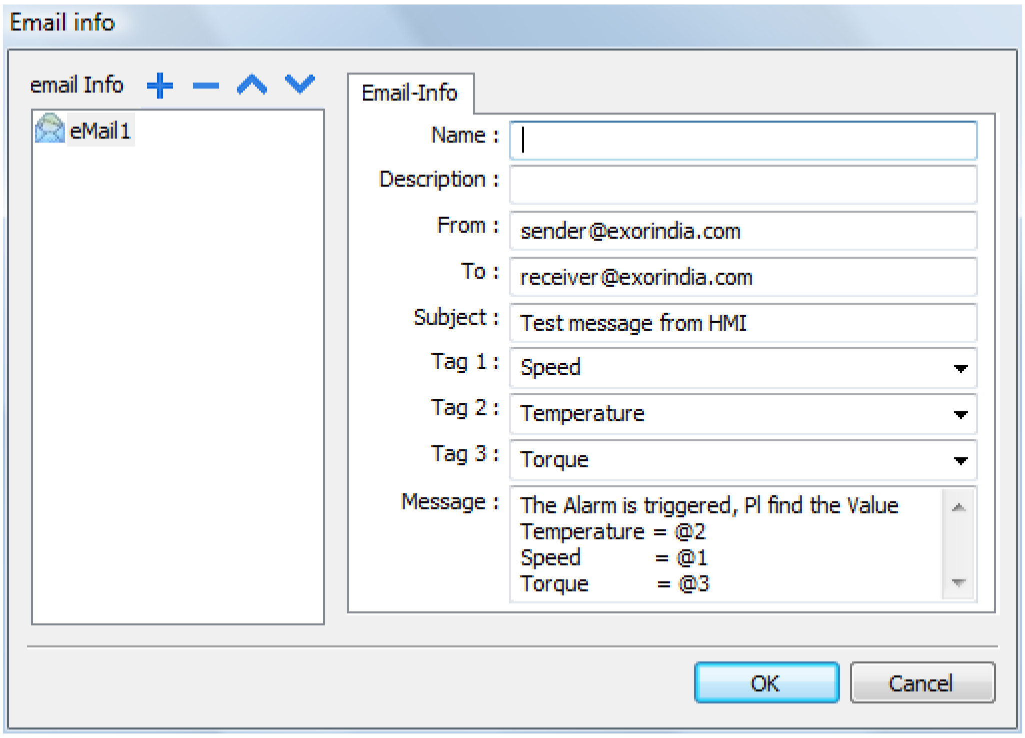
Tag 3 :

Message :

Figure 184

18.4 Sending Live Tag Data through Email

You can send live Tag data to the recipients within the email body. In e-mail Info, select the Tags you want to send in the Tag1, Tag2, Tag3. In the email body, use the keyword “@Tag" to display the Tag data.
An example is shown below.



Name :

Description :

From : sender@exorindia.com

To : receiver@exorindia.com

Subject : Test message from HMI

Tag 1 : Speed

Tag 2 : Temperature

Tag 3 : Torque

Message :
```
The Alarm is triggered, Pl find the Value
Temperature = @2
Speed       = @1
Torque      = @3
```

Figure 185

18.5 Limitation

A maximum number of 3 Tags is supported in each e-mail message body.

19 Working with Java script in JMobile

The purpose of this chapter is to describe the JavaScript interface implemented in JMobile. JMobile JavaScript is based on the ECMAScript scripting language, as defined in standard ECMA-262. Microsoft's Jscript and Firefox JavaScript are also based on the ECMAScript standard. If you are familiar with JavaScript, you can use the same type of commands in JMobile as you do in a web browser. If you are not familiar with the ECMAScript language, there are several existing tutorials and books that cover this subject, such as:

http://doc.trolltech.com/4.3/ecmascript.html

http://www.davidflanagan.com/javascript5/

This purpose of this document is not to explain JavaScript language, but rather to describe how JavaScript is used in the JMobile application.

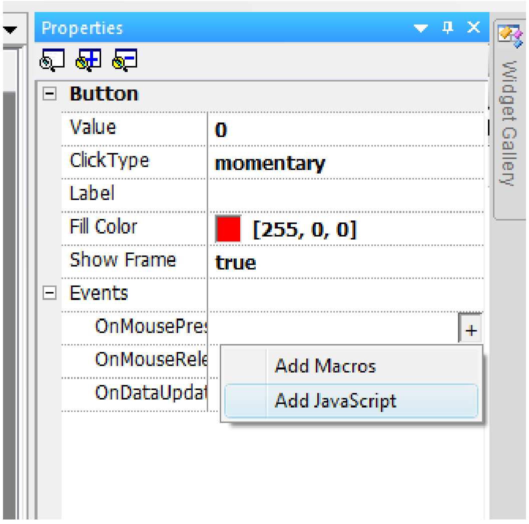
19.1 Execution

A JavaScript function is executed when an event occurs. For example, a user can define a script for the OnMousePress event and the JavaScript script will be executed when the button is pressed on the panel.
It is important to note that JavaScript functions are not executed in the same manner as certain other controller programming scripts, such as Ladder Logic. JavaScript functions are not executed at a given scan rate the whole time. JavaScript functions are only executed when the given event occurs. This approach minimizes the overhead required to execute logic on the panel.

JMobile provides a JavaScript engine running at client side. Each project page can contain scripting with scope local to the page where they are programmed. The project can also contain global scripts that can be executed by scheduler events or alarm events, but it is important to understand that the scripts are still executed at client side. In other words, having more than one client connected to the panel (for instance, an external PC running the Windows Client) means each client will run the same script, providing output results that depend on the input. Inputs provided to the different clients may be different.
This can be clarified, for instance, considering a situation in which the script acts based on a slider position, which can be different for the different clients.

19.2 Events

You can add the java scripts in the following events:
Page Actions
Widget on update
Button Events



Properties

Button	
Value	0
ClickType	momentary
Label	
Fill Color	🟥 [255, 0, 0]
Show Frame	true
Events	
OnMousePres	
OnMouseRele	
OnDataUpdat	

Figure 186

19.2.1 Widget Events

1) **bool onMousePress(me, eventInfo)**
This occurs when the button is pressed.
The code can terminate with a “Return TRUE" or “Return FALSE".
After terminating the code with “Return FALSE", the control is returned to the calling Widget that may launch other actions.
After terminating the code with TRUE, the control is NOT returned to the Widget and this makes sure that no additional actions are executed following the calling event.
The “eventInfo" parameter is reserved for future enhancements.

2) **bool onMouseRelease(me, eventInfo)**
This occurs when the button is released.
The code can terminate with a “Return TRUE" or “Return FALSE".
After terminating the code with “Return FALSE", the control is returned to the calling Widget that may launch other actions.
After terminating the code with TRUE, the control is NOT returned to the Widget and this makes sure that no additional actions are executed following the calling event.
The “eventInfo" parameter is reserved for future enhancements.

3) **bool onDataUpdate (me, eventInfo)**
This occurs when the data attached to the Widget changes.
The value is the data value passed to the Widget.
This event is triggered by the system <u>before</u> the value is passed to the Widget; this means the code programmed here can modify or alter the value before it is actually passed to the Widget.

The code can terminate with a “Return TRUE" or “Return FALSE".
After terminating the code with “Return FALSE", the control is returned to the calling Widget that may launch other actions.
After terminating the code with TRUE, the control is NOT returned to the Widget and this makes sure that no additional actions are executed following the calling event.

The “eventInfo" parameter can be used to get the following details:

eventInfo.oldValue: old value, widget value before the change.
eventInfo.newValue: new value, the value which will be updated to the widget.
eventInfo.attrName: the attribute on which the event is generated

eventInfo.index: attribute index if any, default = 0
eventInfo.mode: W when user is writing to the widget, R otherwise.

19.2.2 Page Events

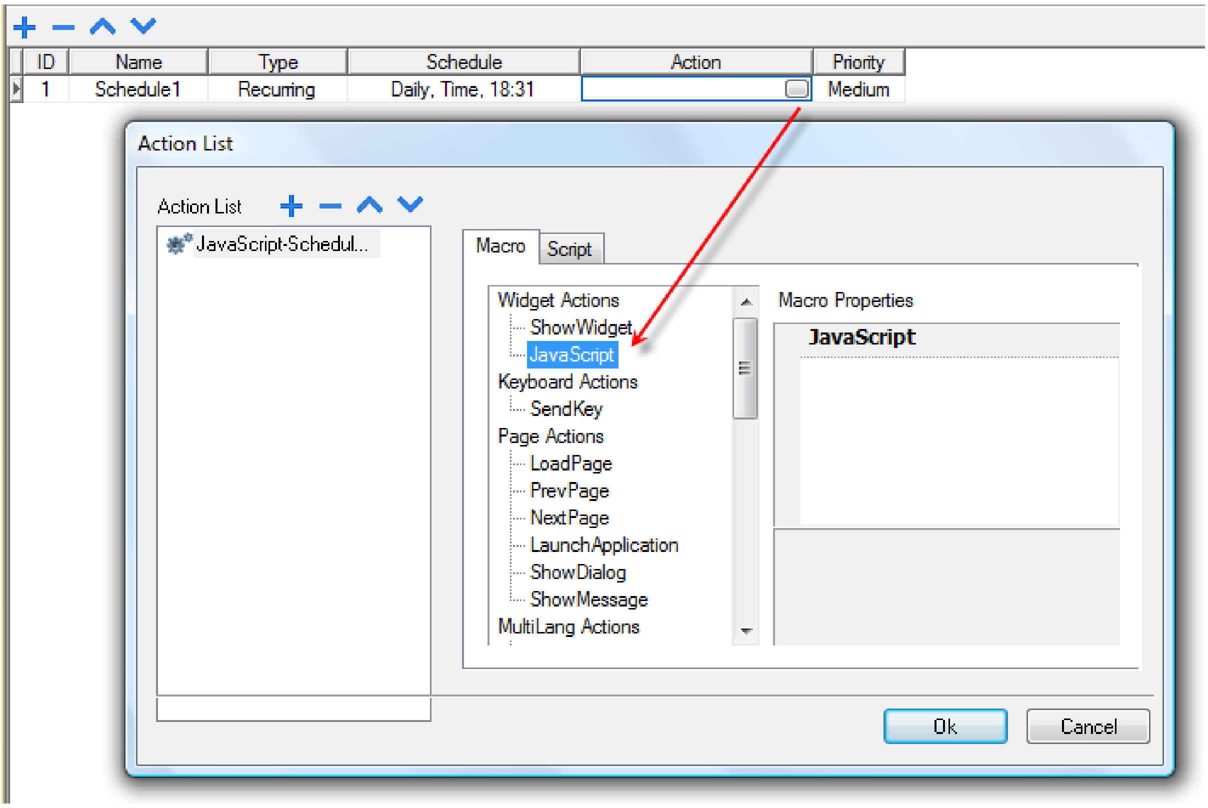
4) void onActivate(me, eventInfo)
This event occurs each time the page is shown.
This java script will execute when the page is Active. It means that, when the page is loaded, the script will execute.
The “eventInfo" parameter is reserved for future enhancements.

5) void onDeactivate(me, eventInfo)
This occurs when leaving the page.
The “eventInfo" parameter is reserved for future enhancements.

19.2.3 System Events

There are two types of system events, one is related to the scheduler, the other one related to the alarms.

Scheduler Event
The event occurs when triggered by the proper action available in the scheduler system as shown in Figure 187.



ID	Name	Type	Schedule	Action	Priority
1	Schedule1	Recurring	Daily, Time, 18:31		Medium

Figure 187

Alarm Event
The event occurs when triggered by a specific alarm condition and programmed in the proper action as shown in Figure 188.

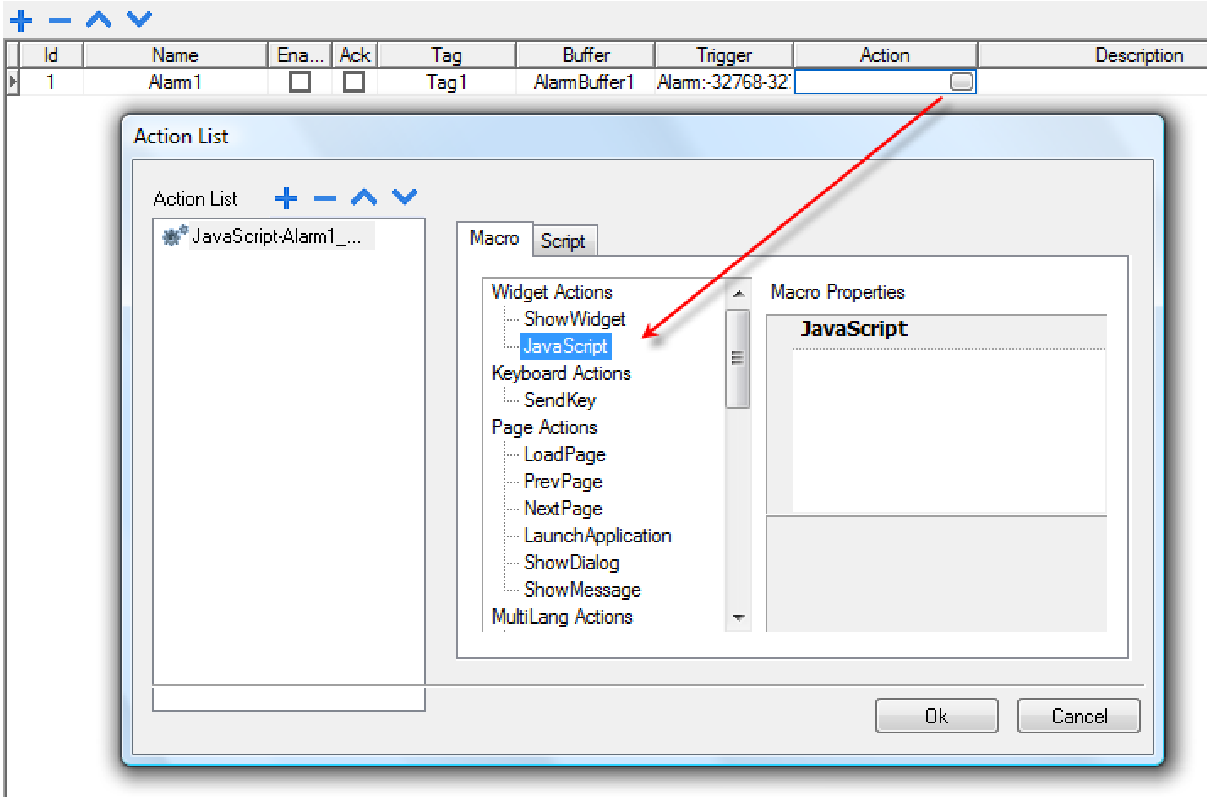
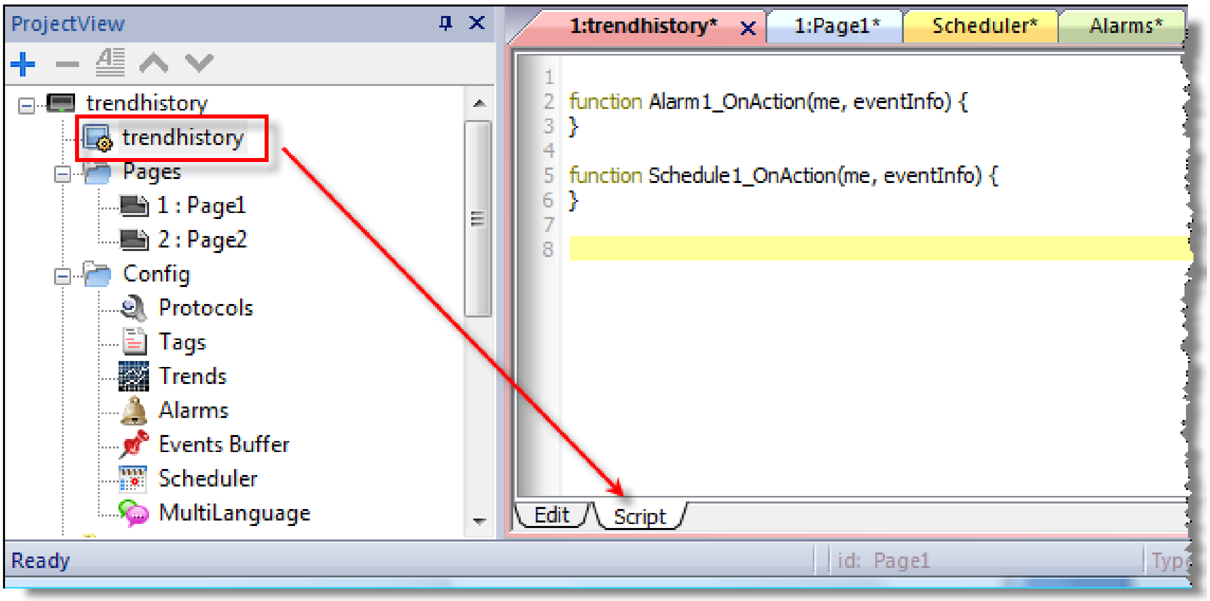


Figure 188

Once the system events are configured, the custom code for them can be edited from the global JavaScript editor interface, which is available from the Project view (double click on the project name icon) as shown in Figure 189.



```
function Alarm1_OnAction(me, eventInfo) {
}

function Schedule1_OnAction(me, eventInfo) {
}
```

Figure 189

19.3 Language Reference

The following sections describe the JavaScript functions and properties that are unique to the JMobile environment.

19.3.1 Objects

JMobile 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 page.

19.3.1.1 Widget

The Widget class is the base class for all elements on the page including the page element. Note that ‘Widget’ is not a specific element but a JavaScript class.

19.3.1.2 Properties:

name

gets or sets the name of the Widget. The name is a unique id for the Widget.

```
    Example:
        function btnStd04_onMouseRelease(me) {
        var wgt=page.getWidget("rect1");
        var name= wgt.name;
        }
```

x

gets the Widget ‘x’ position in pixels

```
    Example:
        function btnStd1_onMouseRelease(me) {
        var wgt=page.getWidget("rect1");
        wgt.x=10;
        }
```

y

gets or sets the Widget ‘y’ position in pixels

```
    Example:
        function btnStd1_onMouseRelease(me) {
        var wgt=page.getWidget("rect1");
        wgt.y=10;
        }
```

width

gets or sets the Widget width in pixels

```
    Example:
        function btnStd1_onMouseRelease(me) {
        var wgt=page.getWidget("rect1");
        wgt.width=10;
        }
```

height

gets or sets the Widget height in pixels.

```
Example:
   function btnStd1_onMouseRelease(me) {
   var wgt=page.getWidget("rect1");
   wgt.height=10;
   }
```

visible

gets or sets the Widget visible state.

```
Example:
   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

gets or sets the Widget value.

```
Example:
   function btnStd6_onMouseRelease(me) {
   var wgt=page.getWidget("field1");
   wgt.value=100;
   }
```

opacity

gets or sets the Widget opacity. Values are decimals from 0 to 1, where 1 is 100% opaque.

```
Example:
   function btnStd8_onMouseRelease(me) {
   var wgt=page.getWidget("rect1");
   wgt.opacity=0.5;
   }
```

rotation

gets or sets the rotation angle for the Widget. The rotation is done by degree and makes a clockwise rotation, starting at the East position.

```
Example:
   function btnStd9_onMouseRelease(me) {
   var wgt=page.getWidget("rect1");
   wgt.rotation=45;
   }
```

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.

19.3.1.3 Methods:

getWidget(wgtName)

returns the child Widget with the given name. If the child Widget does not exist, *null* is returned.

getProperty(TagName, index=0)

returns a property for the Widget with the given TagName and optional index value. Any property that is shown in the JMStudio Property view can be retrieved from the *getProperty* method. The index value is optional and only used for Widgets that support arrays.

setProperty(TagName, value, index=0)

sets a property for the Widget with the given TagName to the value specified in the *value* argument. Any property that is shown in the JMStudio Property view can be set by this method. The index value is optional and only used for Widgets that support arrays. The *setProperty* method returns a boolean true orfalse value to indicate if the property was set.

19.3.1.4 Page Object

This object references the current HMI page. The page is the top-level object of the screen.

19.3.1.5 Properties

backgroundColor

the page background color

```
Example:
   function btnStd11_onMouseRelease(me) {
   var prjpage=page;
   prjpage.backgroundcolor="rgb(128,0,0)";
   }
```

width

gets or sets the Widget width in pixels

```
Example:
   function btnStd05_onMouseRelease(me) {
   var prjpage=page;
   prjpage.width=800;
   }
```

height

gets or sets the Widget height in pixels

```
Example:
   function btnStd05_onMouseRelease(me) {
   var prjpage=page;
   prjpage.height=800;
   }
```

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.

19.3.1.6 Methods

getWidget(wgtName)

returns the child Widget with the given name. If the child Widget does not exist, null is returned.

19.3.1.7 Project Object

This object defines the project widget. The project widget is used to retrieve data about the project such a tags, alarms, recipes, schedules, tags, etc. There is one project widget for the project. The project can be reference through the *project* variable.

19.3.1.8 Methods

startPage

the page shown when the application is started

```
Example:
    var startPage = project.startPage;
    project.startPage = “Page2.jmx";
```

nextPage

the script executes the next page macro.

```
Example:
    project.nextPage();
```

prevPage

The script executes the Previous page macro.

```
Example:
    project.prevPage();
```

homePage

The script executes the Home page macro.

```
Example:
    project.homePage();
```

loadPage

The script executes to load the set page defined in the script.

```
Example:
    project.loadPage("Page5.jmx");
```

showDialog

The script executes to show the dialog page.

```
Example:
    project.showDialog("Dialog.jmx");
```

closeDialog

The script executes to close the currently-opened dialog page.

```
Example:
    project.closeDialog();
```

showMessage

The script executes to display the message popup.

```
Example:
    project.showMessage("Hi This is test message");
```

clearAllTimeouts

The script clears all the JavaScript timers started.

```
        Example:
           project.clearAllTimeouts();
```

19.3.1.9 Tag Object

This object defines a Tag. The Tag can be retrieved by calling project.getTag(*TagName*) where you find *TagName* , the name of the Tag.

Note: You are able to retrieve any Tag in the project; however the Tag value is only valid for the current page that is being displayed on the HMI.

19.3.1.10 Properties:

name

the Tag name

value

value of the Tag. This value is a variant type. It is up to the calling function to know how the Tag value can be correctly converted to other data types.

comment

comment for the Tag

19.3.1.11 Methods

getTag

returns the Tags value string of the given Tag name

Parameters
[in] A string of the tag name.
[in] A state object instance to be filled.
[in] An index if tag is array type. set -1 to return complete array.
[in] Callback function name if an async read is required. Default = “"

Return value
Tags value is returned. If tag is array type and index = -1 then the complete array is returned.

Remarks
For non array tags provide index as 0; this is required for providing backward compatibility with earlier getTag(strTagName, index) usage.

Example

```
        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();
        var value = project.getTag("Tag1", state,-1, "fnTagReady");
        function fnTagReady(tagName, tagState) {
           if (tagName=="Tag1") {
       var myValue = tagState.getValue();
   }
}
```

setTag
sets the given Tag in the project. Name and value are in a string.

Parameters
[in] A string with name of the tag.
[in] A variant instance of the tag value.
[in] An int with index for array tags. Default = 0; set to -1 to pass complete array

Return value
None.

Example

```
        var val = [1,2,3,4,5];
        project.setTag("Tag1", val, -1);
```

```
        var val = "value";
        project.setTag("Tag1", val);
```

19.3.1.12 Group Object

A group is a basic logical element that is associated with a set of logical tags. It provides an interface to enable the uniform operation on a set of logically connected tags.

19.3.1.13 Methods

getTag
Gets the tag specified by name from the group object.

Parameters
[in] String representing the tag name.

Return value
Variant value of the tag. Undefined is returned if tag is invalid.

Remarks
None.

Example

```
        var group = new Group();
        project.getGroup("GroupName", group);
        var value = group.getTag("Tag1");
```

getCount
Returns total number of tags in this group.

Parameters
None.

Return value
Integer number of tags.

Remarks
None.

Example

```
        var group = new Group();
        project.getGroup("GroupName", group);
        var value = group.getCount("Tag1");
```

getTags
Returns the list of all tags in group.

Parameters
None

Return value
Array of all tags in the group

Remarks
None.

Example

```
        var group = new Group();
        project.getGroup("enginesettings", group);
        var tagList = group.getTags();
        var i = 0;
        for(i=0;i<tagList.length; i++){
           var tagName = tagList[i];
           //do something…
        }
```

getGroup
Fast read method which gets the values of all tags in a group from GUI cache if the group is subscribed else from server.

Parameters
[in] name of the group
[in] group instance to be filled.
[in] callback function as string, that is called when group is ready.

Return value
Status (integer) 1 – success, 0 – fail.

Remarks
None.

Example

```
        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,
"fnGroupReady");
        function fnGroupReady(groupName, group)
        {
           var val = group.getTag("Tag1");
           if (val!=undefined) {
           // do something with the value
        }
   }
        }
```

19.3.1.14 State

Class for holding state of a variable acquired from the controlled environment. Beside value itself, it contains the timestamp indicating when the value is collected together with flags marking quality of the value.

19.3.1.15 Methods

getQualityBits
Returns an integer - a combination of bits indicating tag value quality.

Parameters
None.

Return value
Integer containing quality bits.

Remarks
None.

Example

```
        Var state = new State()
        var value = project.getTag("Tag1", state);
        var qbits = state.getQualityBits();
```

getTimestamp
Returns time the value was sampled.

Parameters
None.

Return value
Date object containing the timestamp.

Remarks
Date is a native JavaScript data type.
This is not yet supported on client side

Example

```
        var state = new State()
        var value = project.getTag("Tag1", state);
        var ts = state.getTimestamp();
```

isQualityGood
Returns whether value contained within this State object is reliable.

Parameters
None.

Return value
True if quality is good, false otherwise.

Remarks
None.

Example

```
        var state = new State()
        var value = project.getTag("Tag1", state);
        if(state.isQualityGood())
        {
           // do something…
        }
```

19.3.1.16 Recipe Object

19.3.1.17 Methods

getRecipeItem
Gets the value of the given recipe set element.

Parameters
[in] String representing the recipe name.
[in] String representing the recipe set, can be either the recipe set name or 0 based set index.
[in] String representing the recipe Element, can be either the element name or 0 based element index

Return value
Variant value of the recipe. Undefined is returned if invalid. If of type array, an array object type is returned.

Remarks
None.

Example

```
        Var value = project.getRecipeItem("recipeName", "Set", "Element");
```

setRecipeItem
Gets the value of the given recipe set element.

Parameters
[in] String representing the recipe name.

[in] String representing the recipe set, can be either the recipe set name or 0 based set index.
[in] String representing the recipe Element, can be either the element name or 0 based element index
[in] Variant type recipe set element value, can be an array type too.

Return value
None.

Remarks
None.

Example

```
        Var val = [2,3,4];
        project.setRecipeItem("recipeName", "Set", "Element", val);
```

downloadRecipe
Downloads the recipe set to corresponding tag.

Parameters
[in] String representing the recipe name.
[in] String representing the recipe set

Return value
None.

Remarks
None.

Example

```
        project.downloadRecipe("recipeName", "Set");
```

uploadRecipe
Uploads the value of tags into the provided recipe set.

Parameters
[in] String representing the recipe name.
[in] String representing the recipe set.

Return value
None.

Remarks
None.

Example

```
        project.uploadRecipe("recipeName", "Set");
```

19.3.2 Widgets

The following sections define visual Widgets that can be displayed on the page. All Widgets inherit the Widget object, which means that all properties and methods that are valid for the Widget object are also valid for the each visual Widget.

19.3.2.1 Field

This Widget defines a numeric field Widget on the page. Inherits the Widget class.

Note: You are able to retrieve any Widget on the page, by calling page.getWidget (wgtName) where wgtName is the name of the Widget.

19.3.2.2 Properties:

align
    defines the horizontal alignment. Values are: left, center, right

fontSize
    font size in pixels

fontColor
    font color as a Color field (see Color field definition later in this document)

fontBold
    indicates if the font is bold

min
    min input value for the field

max
    max input value for the field

19.3.2.3 Methods:

None

19.3.2.4 Label

This Widget defines a text label on the page. Inherits the Widget class.

19.3.2.5 Properties:

align
    defines the horizontal alignment. Values are: left, center, right

fontSize
    font size in pixels

fontColor
    font color as a Color field (see Color field definition later in this document)

fontBold

indicates if the font is bold

19.3.2.6 Methods:

None

19.3.2.7 Shape

This object defines a shape on the page. Inherits the Widget class.

19.3.2.8 Properties:

fill

defines the fill color. Type is a Color data type.

stroke

defines the stroke color

thickness

defines the stroke width

19.3.2.9 Methods:

None

19.3.2.10 Line

This object defines a shape on the page. Inherits the Widget class.

19.3.2.11 Properties:

stroke

defines the stroke color

thickness

defines the stroke width

19.3.2.12 Methods

None.

19.3.2.13 Image

This Widget defines an image on the page. Inherits the Widget class.

19.3.2.14 Properties:

fill

defines the fill color. The fill color is only applicable to SVG images.

stroke

defines the stroke color. The stroke color is only applicable to SVG images.

19.3.2.15 Methods:

None.

19.3.2.16 Multistate Image

This Widget defines a multi-state image on the page. Inherits the Image class.

19.3.2.17 Properties:

value

defines the current state index

19.3.2.18 Methods:

None.

19.3.2.19 Message Text

This Widget defines a message text on the page. Inherits the Widget class.

19.3.2.20 Properties:

value

current index value for the message being displayed

align

defines the horizontal alignment. Values are: left, center, right

fontSize

font size in pixels

fontColor

font color as a Color field (see Color field definition later in this document)

fontBold

indicates if the font is bold

19.3.2.21 Methods:

None.

19.3.2.22 Button

This Widget defines a button on the page. Inherits the Widget class.

19.3.2.23 Properties:

value

current value for the button state

```
   Example:
      Widget.value = 1;
```

type

button type

```
      Widget.type = "maintained";
      var type = Widget.type;
```

upColor

color for the button when up (for 2 color buttons)

```
      Widget.upColor = "rgb(100,100,100)";
      var upCol = Widget.upColor;
```

downColor

color for the button when down (for 2 color buttons)

```
      Widget.upColor = "rgb(100,100,100)";
      var upCol = Widget.downColor;
```

19.3.2.24 Methods:

None.

19.3.2.25 Bargraph

This Widget defines a bar graph on the page. Inherits the Widget class.

19.3.2.26 Properties

value

current value for the button state

```
      Widget.value = 10;
      var val = Widget.value;
```

type

bargraph type. Values: horizontal, vertical

```
      Widget.type = "horizontal";
      var type = Widget.type;
```

color

bargraph color

```
      Widget.color = "rgb(100,100,100)";
```

fill

bargraph background color

min

minimum value

max

maximum value

reverse

bargraph direction. Values: true means the bargraph goes in the opposite direction.

19.3.2.27 Methods:

None

19.3.2.28 Indicator

This Widget defines an indicator on the page. Inherits the Widget class.

19.3.2.29 Properties:

value

current value for the indicator state

numStates

number of states in the indicator

min

minimum value

max

maximum value

19.3.2.30 Methods:

None.

19.3.2.31 Scale

This Widget defines a scale on the page. Inherits the Widget class.

19.3.2.32 Properties:

type

type of scale. Values: horizontal, vertical , round

reverse

indicates if the scale is clockwise (for a radial scale)

min

minimum value

max

maximum value

startAngle

start angle for the radial scale

stopAngle

stop angle for the radial scale

19.3.2.33 Methods:

None.

19.3.2.34 IPCamera

This Widget defines the TCP/IP-based camera on the page. Inherits the Widget class.

19.3.2.35 Properties:

refreshRate

refresh rate for the images

```
        Widget.refreshRate = 1; //in seconds
```

ipaddress

IP address of the video source

```
        Widget.ipaddress = "http://192.168.0.1"
```

userName

username for access to the IP camera (if used by the given camera)

```
        Widget.userName = "user";
```

password

password for access to the IP camera (if used by the camera)

```
        Widget.password = "pass";
```

19.3.2.36 Methods:

refesh()

forces a refresh of the image

```
        Widget.refresh();
```

19.3.3 Keywords

Global objects are predefined objects that can be referenced by the names listed below. For example, in the current page, a name can be accessed in JavaScript with the following command:

var myName = page.name;

self

references the Widget attached to the event. For example, when the onDataUpdate event is called, the script can access the Widget object being updated by using the self variable.

```
    self.visible = true;
```

page

references the page object for the current page

project

references the project Widget

19.3.4 Global Functions

print(message)

prints a message to the debugger console window.

```
    Example:
             print("test message");
```

alert(message)

displays a popup dialog with the given message. The user must press the OK button in the dialog to continue with the execution of the script.

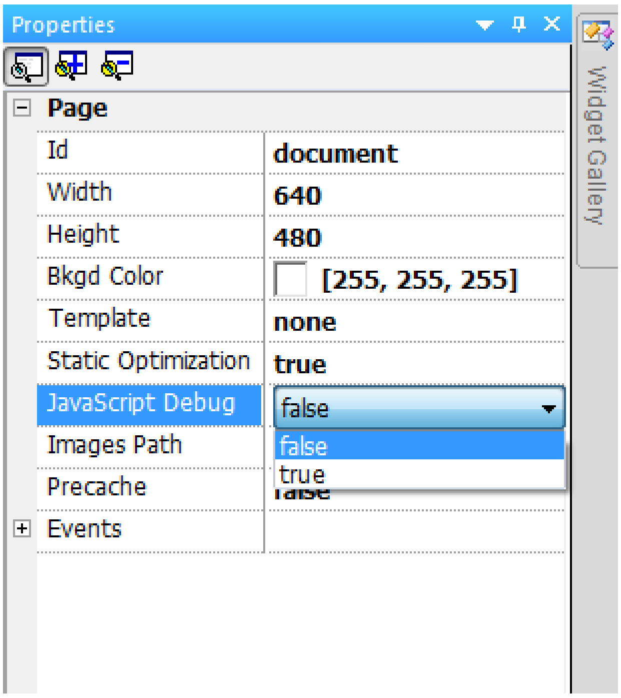
Note: The alert function is often used for debugging JavaScript routines.

Example:

alert("test message");

19.4 Debugging of Java Script

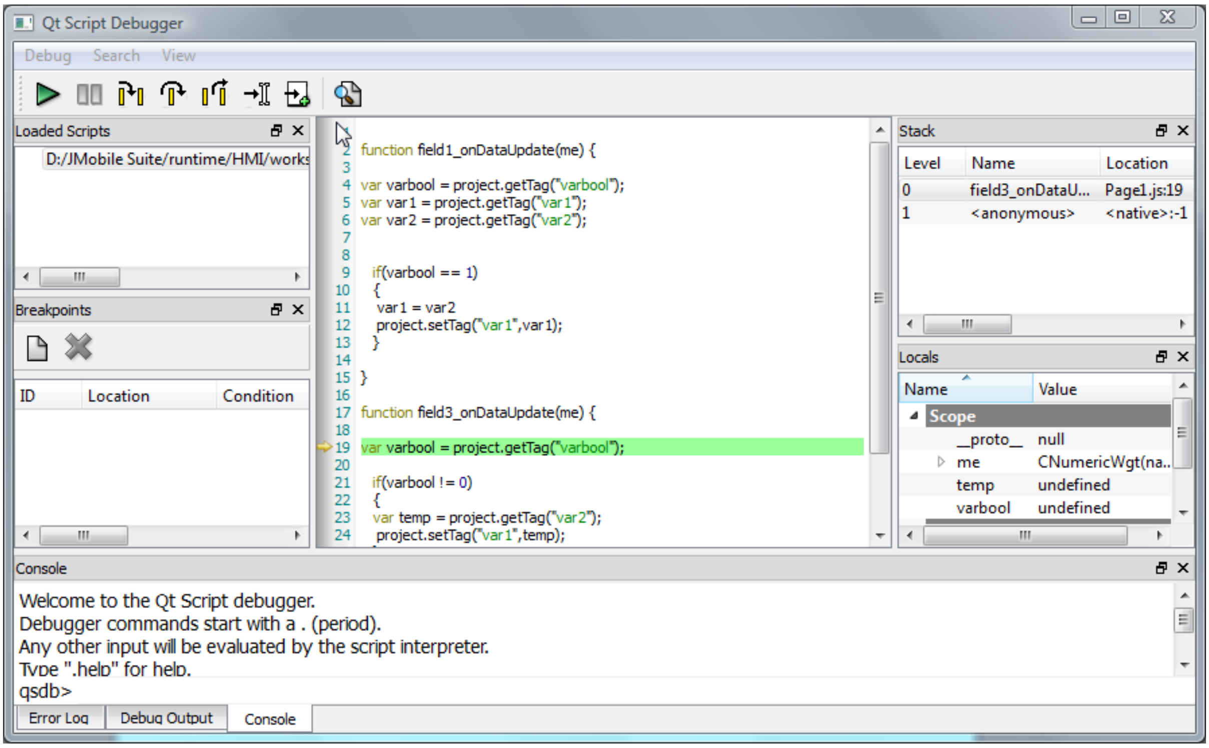
You can enable or disable the debugging mode to find and solve the script error. To enable the debugging mode, in the advanced page property, set JavaScript Debug to True as shown in the Figure 190.



Page	
Id	document
Width	640
Height	480
Bkgd Color	□ [255, 255, 255]
Template	none
Static Optimization	true
JavaScript Debug	false false true
Images Path	
Precache	false
Events	

Figure 190

Then in Runtime, when the events are called, the script debugger will show the debug information (as shown in Figure 191).



D:/JMobile Suite/runtime/HMI/works

ID	Location	Condition

```
function field1_onDataUpdate(me) {

var varbool = project.getTag("varbool");
var var1 = project.getTag("var1");
var var2 = project.getTag("var2");


  if(varbool == 1)
  {
   var1 = var2
   project.setTag("var1",var1);
  }

}

function field3_onDataUpdate(me) {

var varbool = project.getTag("varbool");

  if(varbool != 0)
  {
  var temp = project.getTag("var2");
   project.setTag("var1",temp);
```

Level	Name	Location
0	field3_onDataU...	Page1.js:19
1	<anonymous>	<native>:-1

Name	Value
Scope	
__proto__	null
me	CNumericWgt(na..
temp	undefined
varbool	undefined

```
Welcome to the Qt Script debugger.
Debugger commands start with a . (period).
Any other input will be evaluated by the script interpreter.
Type ".help" for help.
qsdb>
```

Figure 191

19.5 Limitations

Widgets cannot be instantiated from JavaScript. The Widgets can only be access and changed. If you need additional Widgets on the page, you can add hidden Widgets on the page, and show or position them from JavaScript.

20 Updating System Components for UniOP Panels

The eTOP Series 400 series products offer powerful tools for a complete upgrade of system components: Operating System, FPGA and run-time software.
This upgrade can be done using USB flash drives that include the new software modules, and running the proper procedure, described in detail in this document, on the device.

Each unit comes from the manufacturer with a label, "product code", which includes all the information related to factory settings (in terms of hardware, software and firmware components).

Product labeling is the first reference for learning the factory settings and version of the components installed at time of manufacturing.

The update tool on the panel also provides the user with detailed information on the components actually running in the system.

Note: Files required for upgrades depend on the product code. Using wrong files for upgrade may result in system malfunctions, and even render system unusable.

Note: files for upgrade are distributed on demand by technical support department

20.1 Product Code Description

The UniOP eTOP400 series products are identified by a product label, as shown in Figure 192 below. The label reports several pieces information, including the model name, the part number, the power supply, the date of production (in the following format: mm/yy) and two barcodes.

The first code is the version code, the second one the serial number (S.N.).

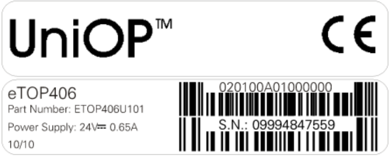


Figure 192

The versioning code is a 15-characte code, which contains the following information:

Typology/Series	2 chars – 00..ZZ	Identifies the series, for instance 400 series
HW type	2 chars – 00..ZZ	Identifies the carrier board and the CPU module
Display and Touch	2 chars – 00..ZZ	Identifies the combination of display and touch screen components
Operating System Type	1 char – A..9	Identifies the type of operating system, for instance Windows CE 6
BSP (Board Support Package) version	2 chars – 00..ZZ	Identifies the BSP version
Type and version of firmware components (boot loader, FPGA, etc..)	3 chars	Identifies the type of boot loader and the versions of the firmware components
Customer code (brand)	3 chars – 000…ZZZ	Identifies the custom brand

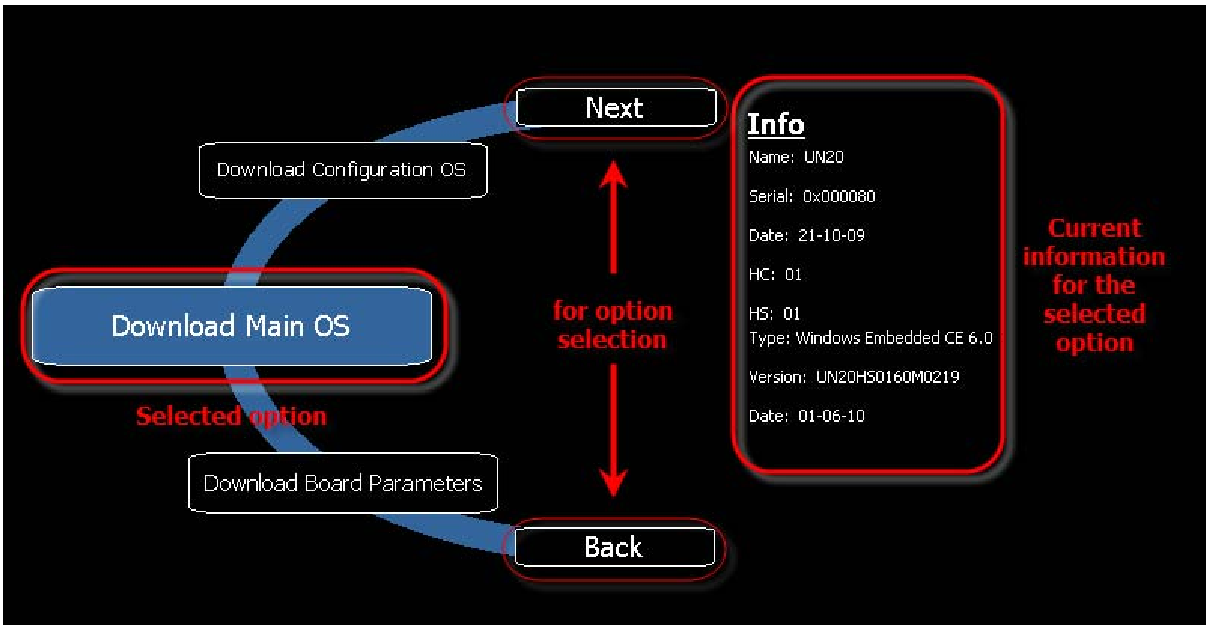
Table 2

Each of these fields is incrementally adjusted any time there is a change in the related component.

20.2 System Settings Tool

The System Settings Tool comes in the shape of a rotating menu, with navigation buttons at the top and bottom to scroll between the available options. The tool is shown in Figure 193.

On the left side, the several components and functions are highlighted and, for each of them, the right side ("Info" pane) shows the information about the current version (when applicable). In the picture below, the version of the Main OS component is shown.


Next
Download Configuration OS
Download Main OS
Selected option
Download Board Parameters
Back
for option selection
Current information for the selected option

Info

Name: UN20

Serial: 0x000080

Date: 21-10-09

HC: 01

HS: 01

Type: Windows Embedded CE 6.0

Version: UN20HS0160M0219

Date: 01-06-10

Figure 193

System Settings have two operating modes: **User Mode** and **System Mode**. The only difference between them is the number of available options.

"**User mode**" is the simplest possible interface where a generic user can get access to the basic settings of the panel:

Calibrate Touch: allows to calibrate the touch screen interface

Network: allows to change the options of the panel on-board network card

Time: allows to change the panel RTC options, including time zone and DST

Display settings: automatic backlight turnoff and brightness adjustment

BSP settings: allows to check the **BSP** (**B**oard **S**upport **P**ackage) version (example 2.37), check the operating hours timers for the unit and separately for the backlight, enable or disable the buzzer, and enable or disable the use of the "low battery" front LED indicator.

"**System Mode**" is the complete interface of the System Settings tool, where all the available options are available; in addition to the options available in the "User Mode", we have the following important options:

Format Flash: allows to format the internal panel flash disk

Resize Image Area: allows to resize the flash portion, reserved to store the splash screen image that is displayed by the unit at power up; default settings are normally ok for all the units

Download Configuration OS: allows to check the actual version and upgrade the back-up operating system (see below, in the next chapter, for additional details)

Download Main OS: allows to check actual version and upgrade the main operating system, see below in the next chapter for additional details

Download Splash Image: allows changing the splash screen image displayed by the unit at power up; the image should be provided in a specific format. We suggest you update Splash Screen Image directly from Studio software, which supports this feature starting from V 1. 50

Download Bootloader: allows for checking actual version of the system boot loader and upgrading it (see below for additional details)

Download Main FPGA: allows to check actual version and upgrade the main FPGA firmware (see below for additional details)

Download Safe FPGA: allows for checking actual version and upgrading the back-up (safe) copy of the FPGA Firmware (see below for additional details)

Download System Supervisor: allows for checking actual version and upgrading the system supervisor firmware responsible for RTC and power supply handling (see below for additional details)

Note: the System Settings tool also includes other options; not all options are described and documented at this time

The System Settings tool is accessible from the JMobile context menu by selecting the item "Show system settings".
When activated in this way, the System Settings tools always starts in "User Mode".

The JMobile context menu can be activated by pressing and holding down the free area on the screen, until the menu is displayed.

The eTOP400 series products also support a special procedure for accessing the System Settings tool; the special procedure is required when the System Settings is started in System Mode, or when the standard access procedure is not accessible for some reason.
When activated by special procedure, the System Settings tool always starts in "System Mode".

The special access to the System Settings tool can be activated with a tap-tap sequence over the touch interface during the power-up phase.
Tap-tap consists of a high frequency sequence of touch activations, done by the simple means of finger tapping the touch screen, performed during the power up and started immediately after the panel is powered.

20.2.1 List of Upgradable Components

The panels support the upgrade of the following components:

System Supervisor Firmware of the system supervisor controller
(sample file name: packaged_GekkoZigBee_v4.13.bin)

Note: IMPORTANT - The System Supervisor Component can be upgraded only if actual version on the panel is V4.13 or above. Version V4.08, V4.09, V4.10 and V4.11 MUST NOT be updated, they do not support automatic update from System Settings.

Main FPGA FPGA firmware
(sample file name: *h146xaf02r06.bin*)

Safe FPGA back-up copy of the Main FPGA that ensures unit booting in case of main FPGA corruption (may be after failed update)
(Sample file name: *h146xaf02r06.bin*)

Note: When updating FPGA firmware on the panel, the same file must be used for Main and Safe FPGA components

Boot-loader Loader to handles panel start-up
(sample file name: redboot_UN20HS010025.bin)

Main OS Main Operating System
(sample file name: mainos_UN20HS0160M0237.bin)

Configuration OS Back-up operating system that ensures units that are recovering as consequence of main operating system corruption (may be after a failed update)
(sample file name: configos_UN20HS0160C0237.bin)

20.3 Update of System Components from JMobile Studio

JMobile Studio provides a tool to update system components downloading them to the target device using the Ethernet communication interface.

Note: this feature is available starting from JMobile Studio V1.50 and above.

The tool is called Manage Target and it is available from the Run menu.

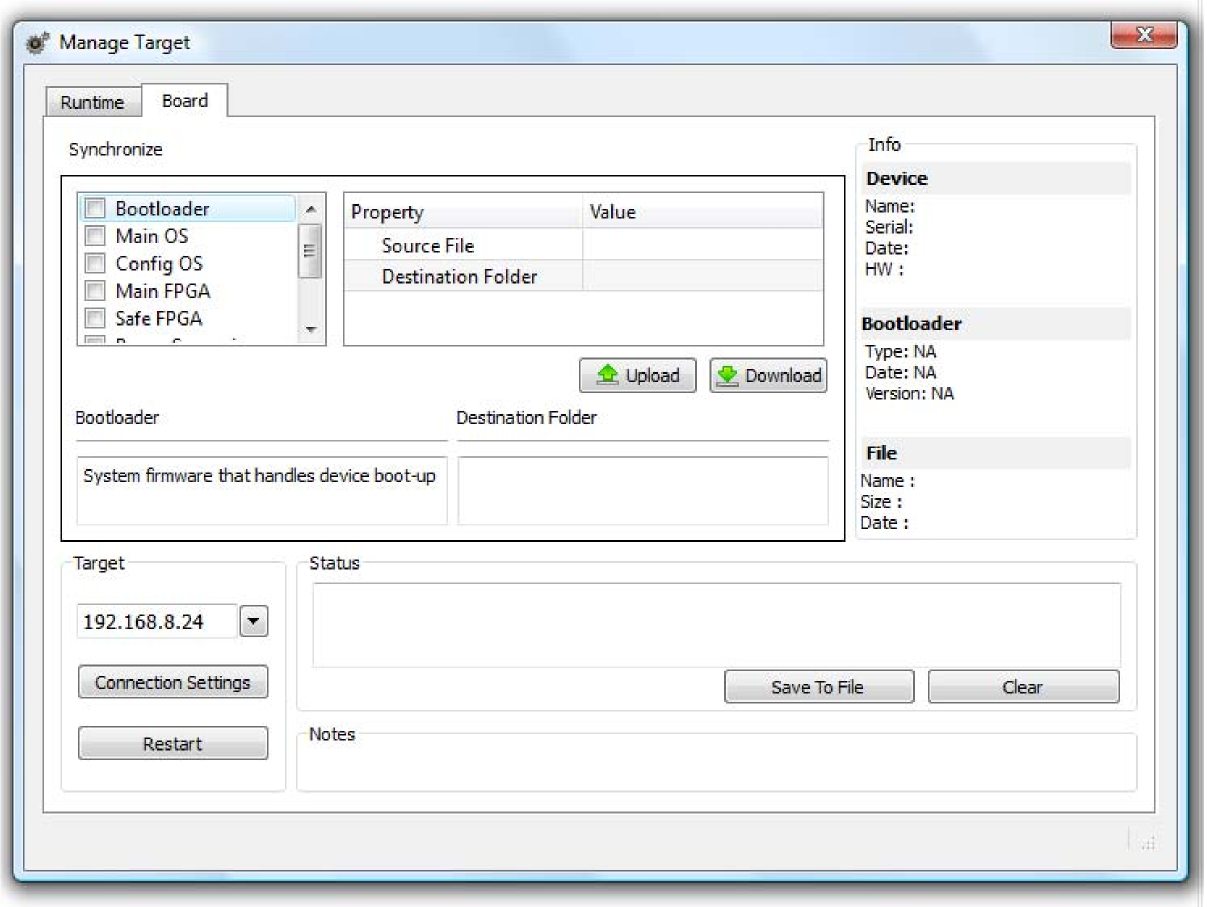


Figure 194

The Manage Target dialog has two tabs. Click on the "Board" tab to access to the board support tools.
The first step is to use the Target discovery function to locate the panel IP from the local network. Click on the little arrow symbol and identify the panel from the list of units recognized in the network. In case the panel is not listed, you can try a second time or type directly the IP in the box. Click then out of the box to accept the inserted IP. See Figure 195.

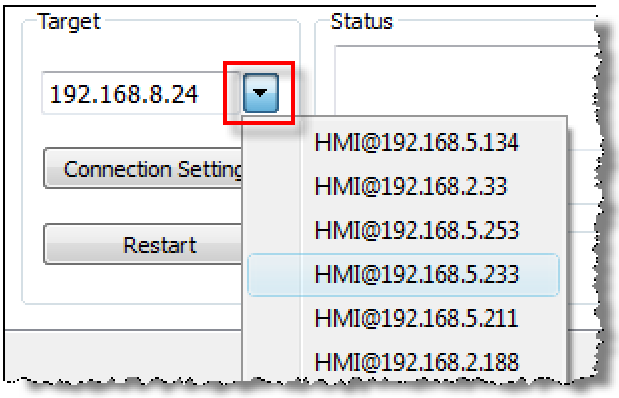
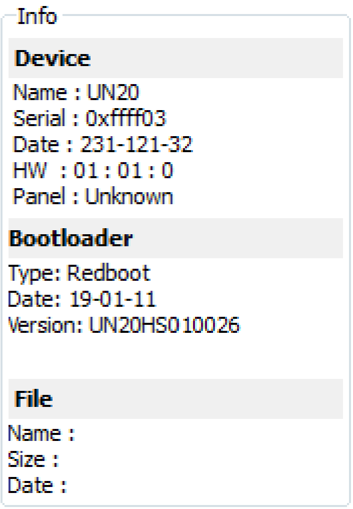


Figure 195

When the device is recognized the Info box shows the target details as shown as an example in Figure 196



Info

Device

Name : UN20
Serial : 0xffff03
Date : 231-121-32
HW : 01 : 01 : 0
Panel : Unknown

Bootloader

Type: Redboot
Date: 19-01-11
Version: UN20HS010026

File

Name :
Size :
Date :

Figure 196

Locate then in the component list the one you need to update, check the box and browse for the file from the "Source file" box as shown in Figure 197.

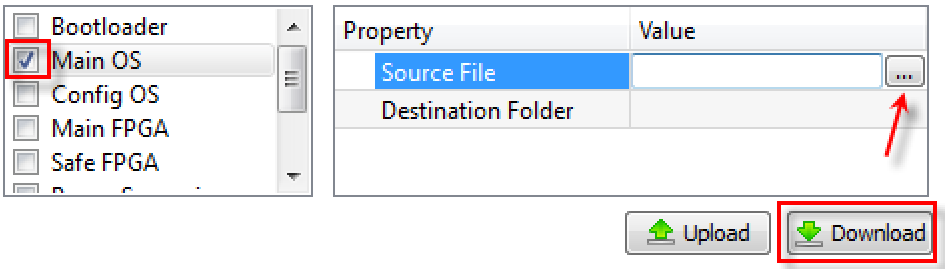


Figure 197

Click then download and check the progress from the Status box below.

Note: in the component selection you can mark more than one check box and provide the related file to be downloaded. The system will execute then the transfer of all the elements one next to the other and only at the end you will need to cycle the power of the system.

Manage target allows also to replace the default splash screen image shown by the devices during the power up phase. Image for the splash screen must be provided in bitmap format saved in RGB 565 format.

Note: Splash screen images must NOT be bigger than 500 KB and they must have black background. This ensures the best optical results.

20.4 Update of the System Components via USB Flash Drive

The upgradable components in a unit are listed below. For each component, a of the file name used when distributing the update is provided.

Note: IMPORTANT – only units with regular serial numbers can be upgraded; units marketed as "prototypes" or having "Prot." serial numbers **cannot** be upgraded

20.4.1 Upgrade Steps

Copy all files you need to upgrade in a USB Memory and plug this into the USB port of the panel.

Start the System Settings tool with the special procedure for getting this in "System Mode, and then locate the desired item in the rotating menu. Click directly on the item (the blue button with white label) and browse to locate the proper file stored on the pen drive (USBMemory). Figure 198 shows an example for the Main OS components.

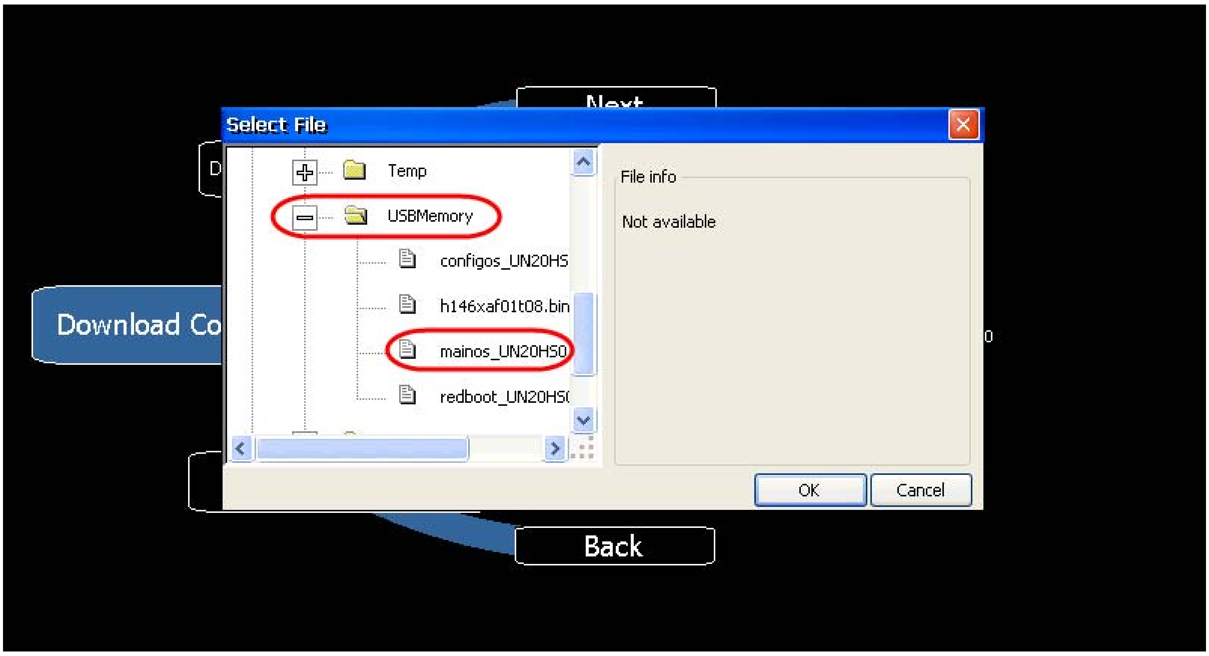


Figure 198

Note: Select te "Download" command to transfer files <u>to</u> the panel. Select the "Upload" command to get files <u>from</u> the panel.

Then follow the instructions on the screen to proceed with the update. A progress bar on the screen will inform you of the status of the operation.
The System components have to be upgraded in the following order, without powering down between one upgrade and the next one:

System Supervisor

Main FPGA

Safe FPGA

Bootloader

Main OS

Configuration OS

Note: When all the upgrades are completed, just power cycle the unit to return to normal operation. Power cycle is required only AFTER all the components have been updated.

20.5 Additional Steps for Units with BSP Older than V2.37

The next steps are ONLY required if the Main OS version currently installed is older than V2.37. With reference to the product version matters explained in chapter 20.1, the procedure described here is required for all the units with BSP version "01".

Figure 193 (above) shows how to read the Main OS version directly from the rotating menu in the Info pane (right side). The last 3 digits of the version string give the version number; in the picture the version is V2.19.

After the **Configuration OS** component is upgraded, power cycle the unit and recall again the System Setting tool in its "System Mode" with the tap-tap procedure.

Locate in the rotating menu the **Format Flash** command and execute it.
At first, the system will try a "quick format" of the flash disk. The quick format method is called "Partition Format".
If Partition Format fails for any reason the panel reports a dialog with the following message:
"Partition Format Failed – Doing Store Format – System may appear stuck for a while"
Confirm, then click OK to start the Store Format.

Note: Storing Format requires up to 15 minutes to complete. During this period the panel appears frozen. Do not the power cycle, and thereby leave the finish process. At the end, the system automatically restarts.

When format is completed, connect the network cable, enter the Windows CE Control Panel and double click on the "System Settings" icon.

Locate in the rotating menu the **Calibrate touch** item and start it.

Execute the calibration following the instruction on the screen using a stylus pen with thin but soft point.

Once the calibration process is completed, go back to Control Panel, locate the icon "Registry Save" and double click on it to store on the registry the calibration parameters.

20.5.1 JMobile Runtime Installation

Because of the flash format operation, the panel requires the installation of the JMobile Runtime at this point.

Install the JMobile Suite software on the computer and locate the "Runtime" folder under JMobile Suite Installation directory.
Copy the entire folder "UN20_WCE6 (MIPSIV_FP)" on the USB pen drive.
Rename the "UN20_WCE6 (MIPSIV_FP)" folder name in "**qthmi**". Please note the system is case sensitive, and the new name has to be with lower cases.
Insert the USB pen drive in the USB port, open "My Device", open "USBMemory" and copy the entire "qthmi" folder from USB to the Flash disk, immediately under the root: \Flash\qthmi.

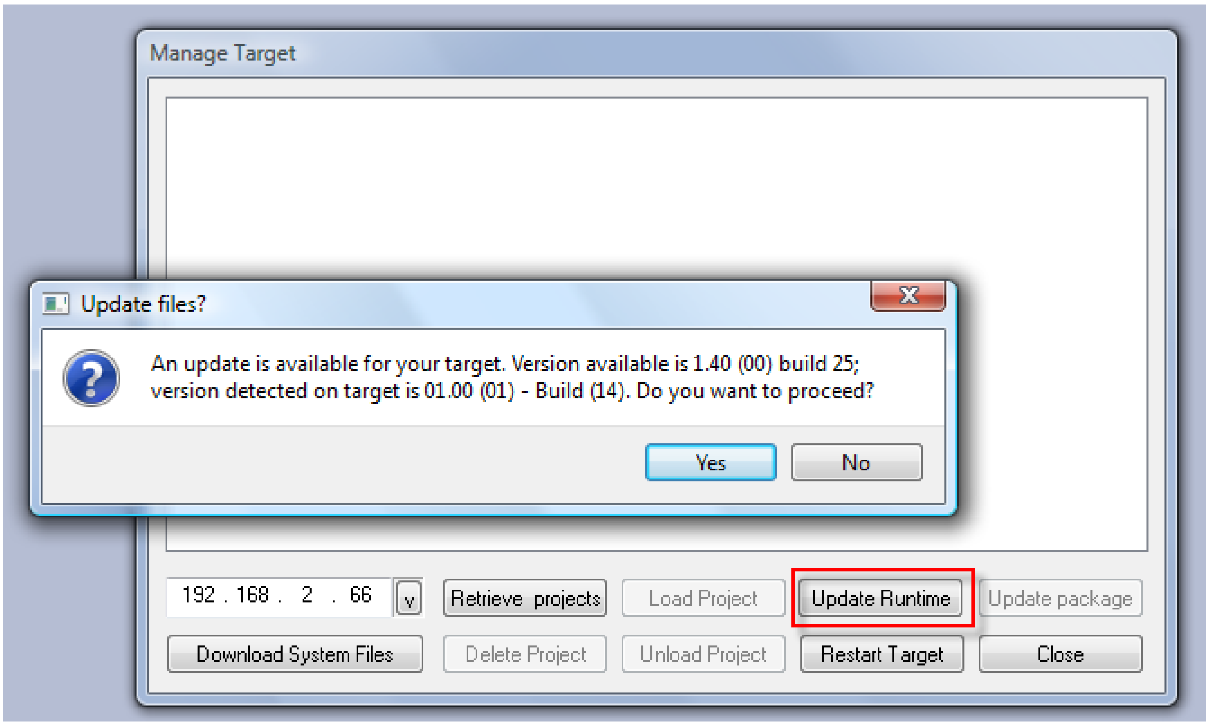
Cycle the power of the unit to return to normal operation; the panel will automatically start launching the JMobile Runtime during the boot up phase.

21 Updating Panel Runtime

Panel Runtime can be updated using a USB pen driver, or directly from Studio.

21.1 Updating Runtime from Studio

From the Run menu select the "Manage Target" option, then click on "Update Runtime", as shown in the following image.



Manage Target

Update files?

An update is available for your target. Version available is 1.40 (00) build 25; version detected on target is 01.00 (01) - Build (14). Do you want to proceed?

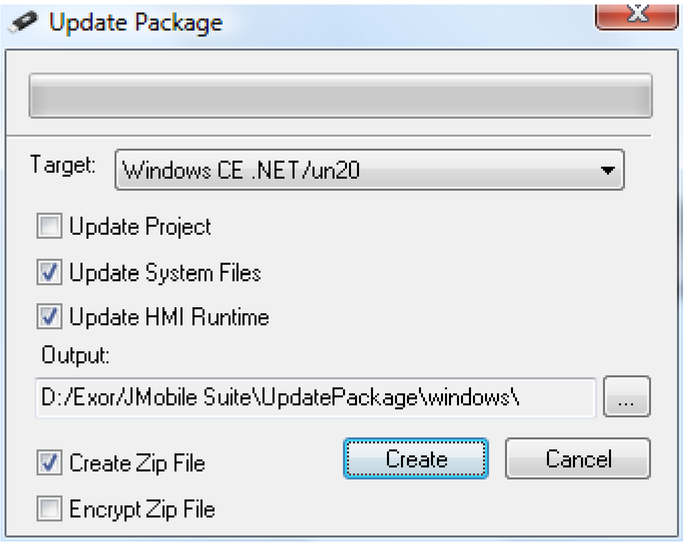
Figure 199

A confirmation message will inform about the current version installed and the version available in Studio.

Note: If Studio is installed on computer running Vista or Windows 7 operating systems, you need to start it "as Administrator" using a right click on the JMobile Studio icon and selecting "Run as Administrator"

21.2 Updating Runtime from USB Pen Drive

From the Run menu, select "Update package" to start the update package creation process.



Update Package

Target: Windows CE .NET/un20

□ Update Project

☑ Update System Files

☑ Update HMI Runtime

Output:

D:/Exor/JMobile Suite\UpdatePackage\windows\

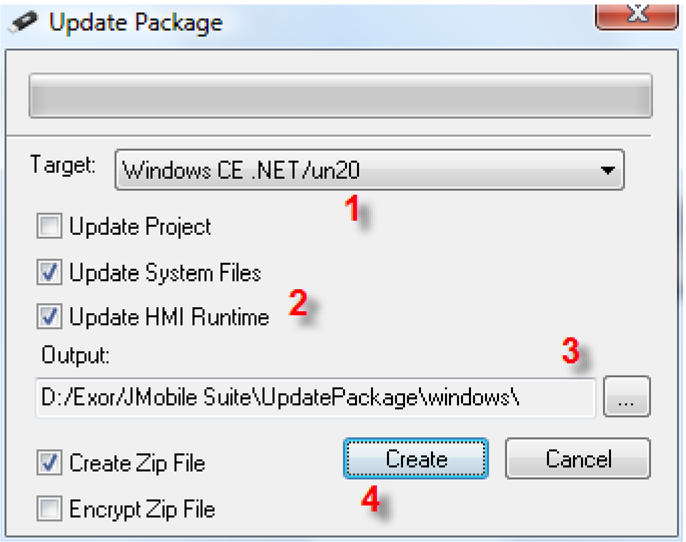
☑ Create Zip File

□ Encrypt Zip File

Figure 200

21.2.1 Package Creation

Select the Target from the Target drop down list. Select the components you need to update and specify the output directory. Then click on "Create".



Update Package

Target: Windows CE .NET/un20

□ Update Project

☑ Update System Files

☑ Update HMI Runtime

Output:

D:/Exor/JMobile Suite\UpdatePackage\windows\

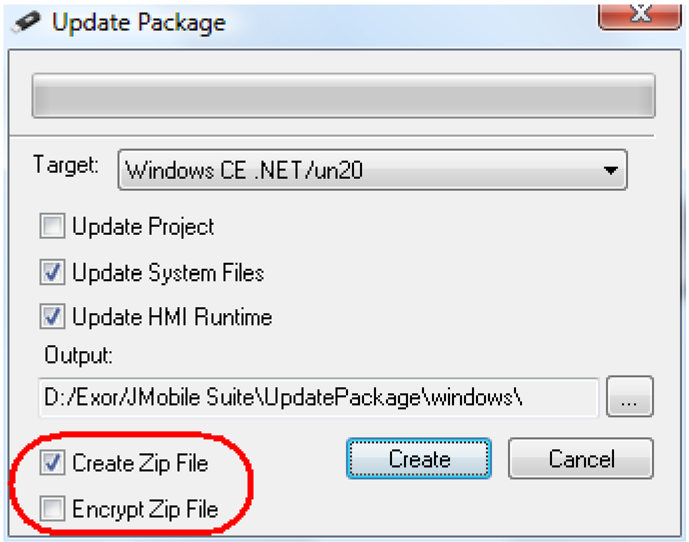
☑ Create Zip File

□ Encrypt Zip File

Figure 201

21.2.2 Zipped Package

The Package can be created in the zip format.
If Encrypt is checked, then a hard-coded password is used (64 bits) to encrypt the zip file.



Update Package

Target: Windows CE .NET/un20

☐ Update Project

☑ Update System Files

☑ Update HMI Runtime

Output:

D:/Exor/JMobile Suite\UpdatePackage\windows\

☑ Create Zip File

☐ Encrypt Zip File

Figure 202

21.2.3 Update Runtime

Copy the package created onto a USB Stick, or create the package directly on the pen drive, then plug the USB Stick into HMI.
Press and hold your finger on the screen for few seconds, until the context menu is shown.
Click on Update to activate the procedure.

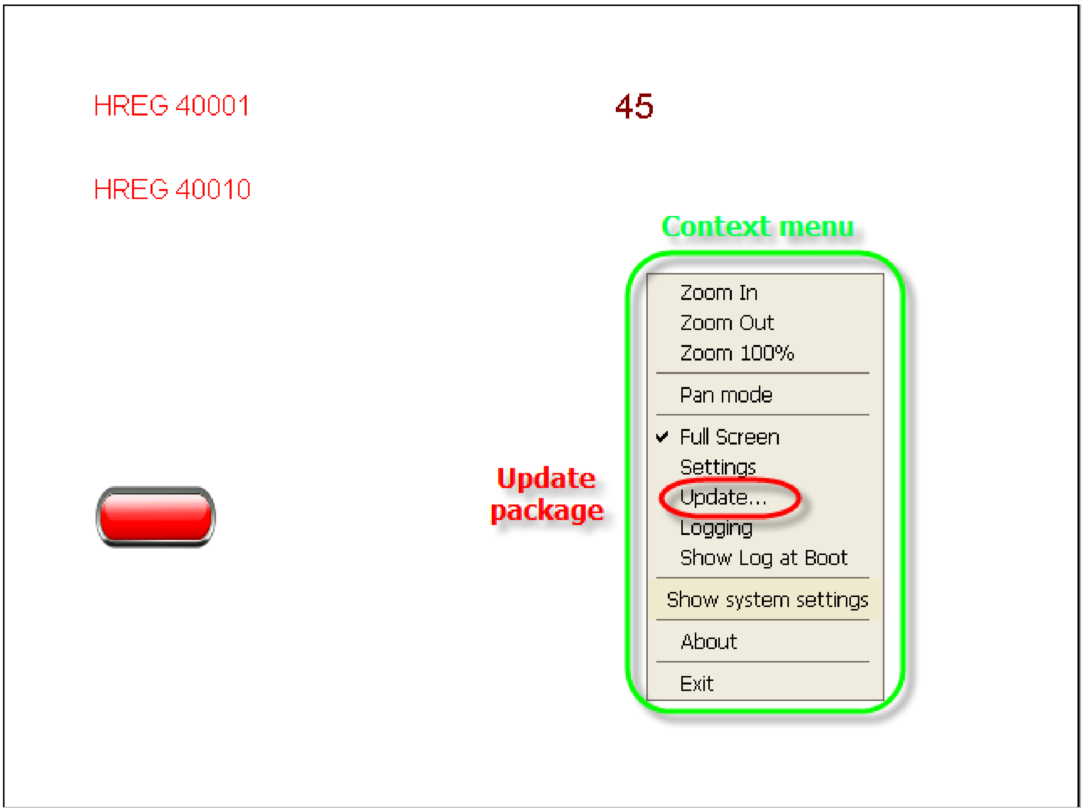
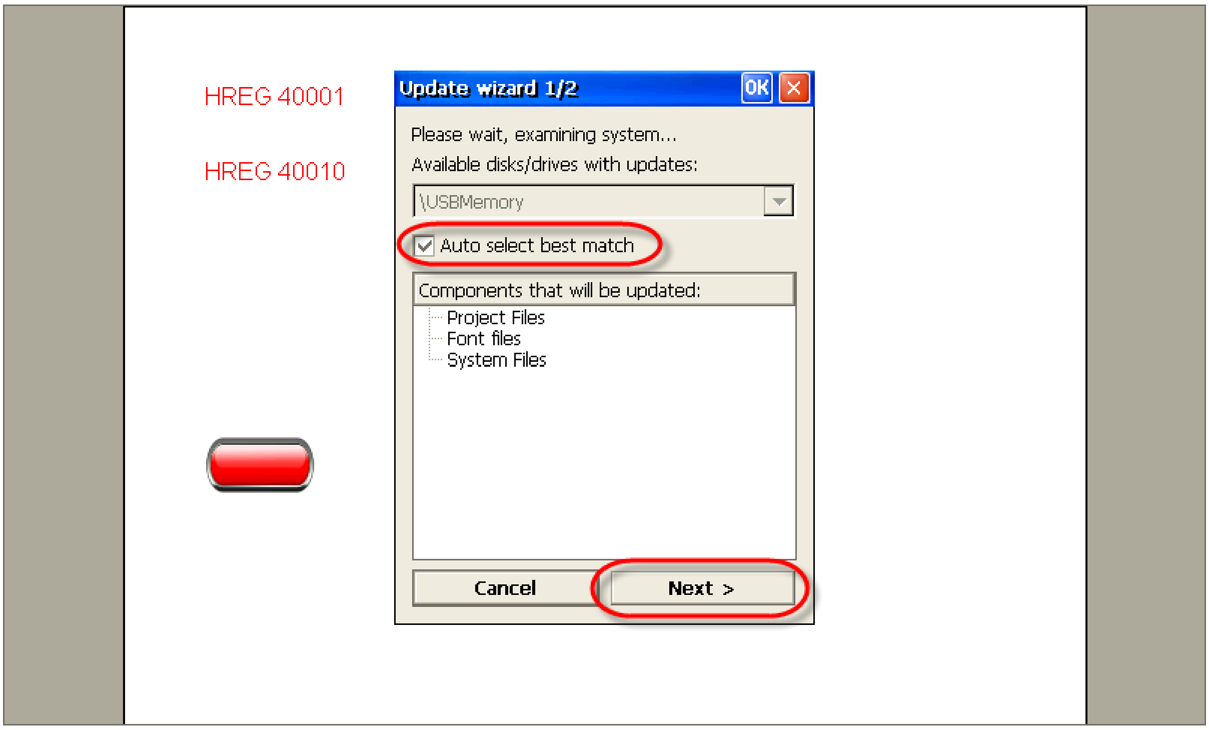


Figure 203

The wizard utility will start. Check "Auto select best match",and then click "Next".



Please wait, examining system...

Available disks/drives with updates:

\USBMemory

☑ Auto select best match

Components that will be updated:

- Project Files
- Font files
- System Files

Figure 204

Press on "Close" at the end of the operation.

22 JMobile Functional Specifications and Compatibility

The scope of this chapter is to provide a clear overview of the supported functions and related limitations for JMobile Studio and JMobile Runtim for the UniOP eTOP400 series.

What is listed below in this document is a safe limitation, above that which proper operation and state-of-the-art performance of the system is not guaranteed.

22.1 Table of Functions and Limits

Function \ Feature	Max allowed
Number of pages	1000 (max screen size 1280x800 pixels)
Number of Tags	10000
Number of dialog pages	20
Number of objects of any type in one page	2000
Number of Recipes	32
Number of parameter sets for a Recipe	32K
Number of elements per each Recipe	1000
Number of user groups	20
Number of users	50
Number of concurrent clients	4
Number of schedulers	30
Number of alarms	2000
Number of templates pages	50
Number of actions programmable per each button state	32
Number of Trend Buffers	30
Number of curves per Trend Widget	5
Number of curves per page	10
Number of samples per each Trend Buffer	200000
Number of messages in a message field	1024
Number of languages	16
Number of events per buffer	2048
Number of event buffers	4
JavaScript file size per each page	8KB
Size of project on disk	30MB

22.2 Compatibility

Starting from the first official release of JMobile V1.00 (00) we have applied the following policy for compatibility:

JMobile Studio version MUST be always aligned with JMobile Runtime on panel; the user has the responsibility to update Runtime components on the Target device together with any Studio update; a Runtime update can be done directly from Studio using the "Update Target" command available in the "Run\Manage Target" dialog

Any version of Studio newer than V1.00 (00) is able to open and properly handle projects created on an older version, but no older than V1.00 (00)

Projects created with older versions of Studio, but not older than V1.00 (00), opened with later versions and deployed to compatible Runtime, are ensured to maintain the performance and functionality just as before.

Compatibility between newer versions of Runtime and those projects created and deployed with older versions of Studio is not ensured.