Technical Manual VisuControl

Technical Manual VisuControl

Technical Manual VisuControl

Contents

1 Getting Started with VisuControl Software

VisuControl software is a application that allows you to create graphical HMI pages. VisuControl 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 VisuControl.

This document describes how to use the VisuControl application, and is divided into chapters that represent the key operations of VisuControl. 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 VisuControl software to design control panel applications that run on VisuControl Touchpanels.
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 VisuControl on a PC

1.2.1 System Requirements

VisuControl 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. Unzip VisuControlLite (or VisuControlPro if you have purchased a license key). Install VisuControl and follow the instructions on the screen.

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

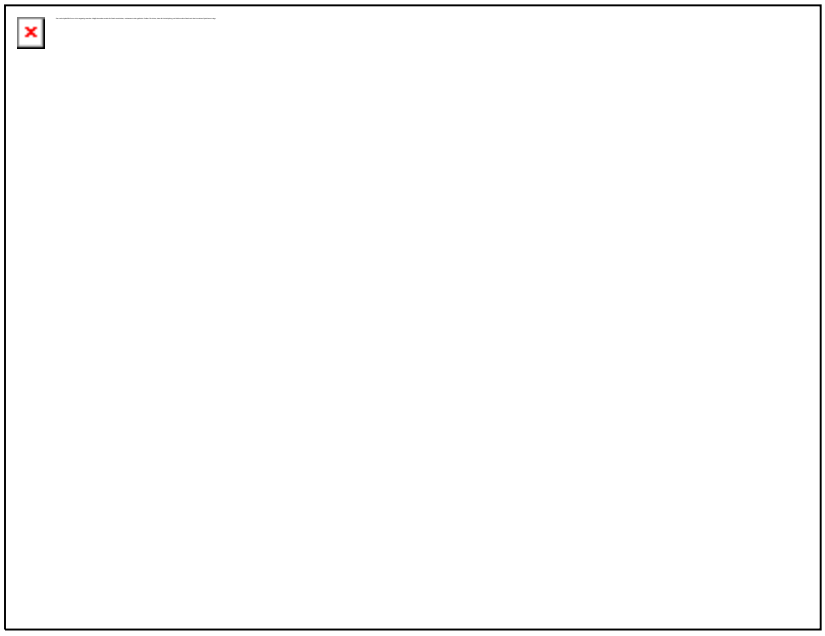

Figure 1

Click Next

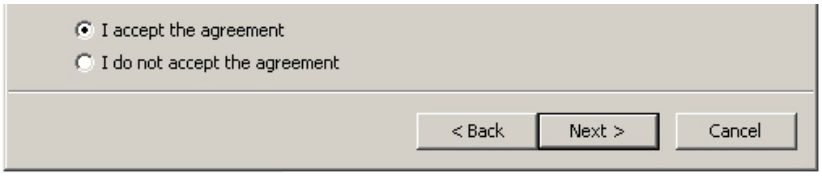

Figure 2

Read the VisuControl software license and accept the agreement; then follow the instructions on the screen. The default location for the VisuControl software is "C:\Program Files\MDT\VisuControl". Default installation path can be changed depending on need.

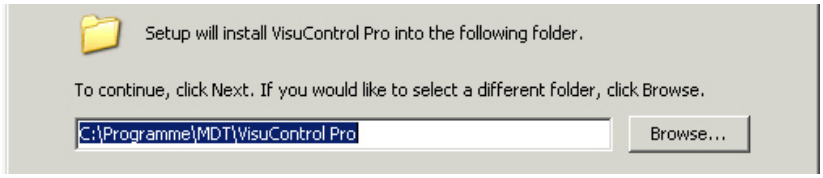

Figure 3

The installation procedure will create a program group called “VisuControl” in the Start menu. A VisuControl icon can be added to the desktop. VisuControl

Figure 4

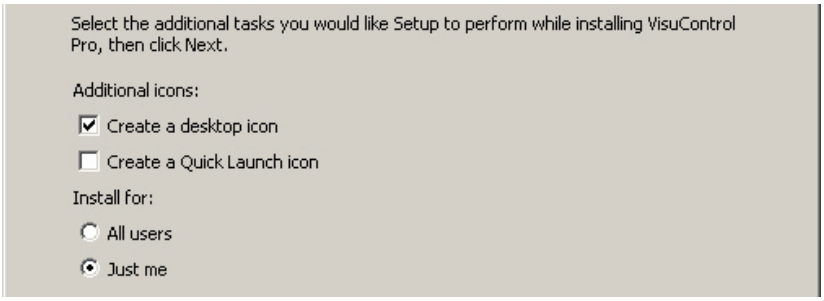

Figure 5

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

1.3 VisuControl Package

VisuControl contains the following package as part of installation.

VisuControl	VisuControl 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.
VisuControl Client	VisuControl Client is a light-weight application that can be used on Windows computers to remotely manage an application running on an HMI.
VisuControl HMI Runtime	VisuControl HMI Runtime is a standalone application that runs on the VisuControl Touchpanels.

2 The VisuControl HMI runtime

The HMI runtime is designed to support different platforms and different operating systems.All the panels are running today on the base of the Windows CE operating system (Version 6 R3).
The operating system and all its options are built around the minimum set of requirements of the HMI runtime; there is no option to get direct access to the Operating system settings as all the needed ones are managed via the runtime itself or via VisuControl.

Later in this document you will find more information about how to install the runtime components and how to manage the update of other system components (firmware) on the units, but always with a dedicated interface which prevents a direct access to the operating system, often a source of complexity.

2.1 Runtime Modes

The HMI runtime is composed by two logic units: one is called “server”, the other one is called “client”.

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

The client unit is the part which is responsible for the visualization process: using the data collected at server side to render on the display as graphical information.

The server unit of the HMI runtime can be in one of two operating modes: Configuration Mode and Operation Mode.

Configuration Mode: the server is idle; activity has not started for one of the following reasons:
- system files are missing
- activity stopped by the execution of the system call-“EnterCFGMode” (available in the list of actions programmable for an event); as an example, this is the condition in which the communication protocol does not run

Operation Mode: the server is active; it is operating according to the settings defined by the system files and by the application project.

The server operating mode is independent of the client side operating mode; you may have a visualization running but server activity stopped.

3 Basic Unit Settings

The device basic settings are available from the System Settings Menu, which is accessible through the VisuControl Context Menu if the panel has already the runtime installed or using the dedicated button if the unit is in loader mode (see for this the chapter “The Runtime Loader” below in this document.

Press and hold your finger on an empty area of the screen for few seconds, until the context menu appears as shown in the figure below.

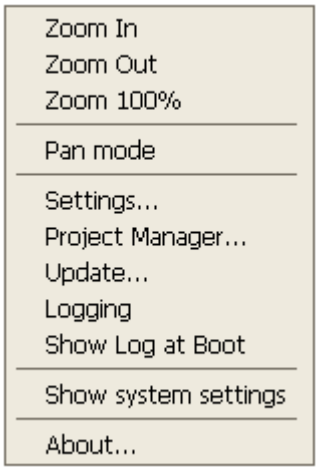

Figure 6

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
Plugin List	Provides a list of the plug-in modules installed and recognized by the system; this option may not be supported by all platforms and all versions.
Note:	Settings selected and confirmed with the OK button in the upper right corner of the dialog are automatically saved to the registry.

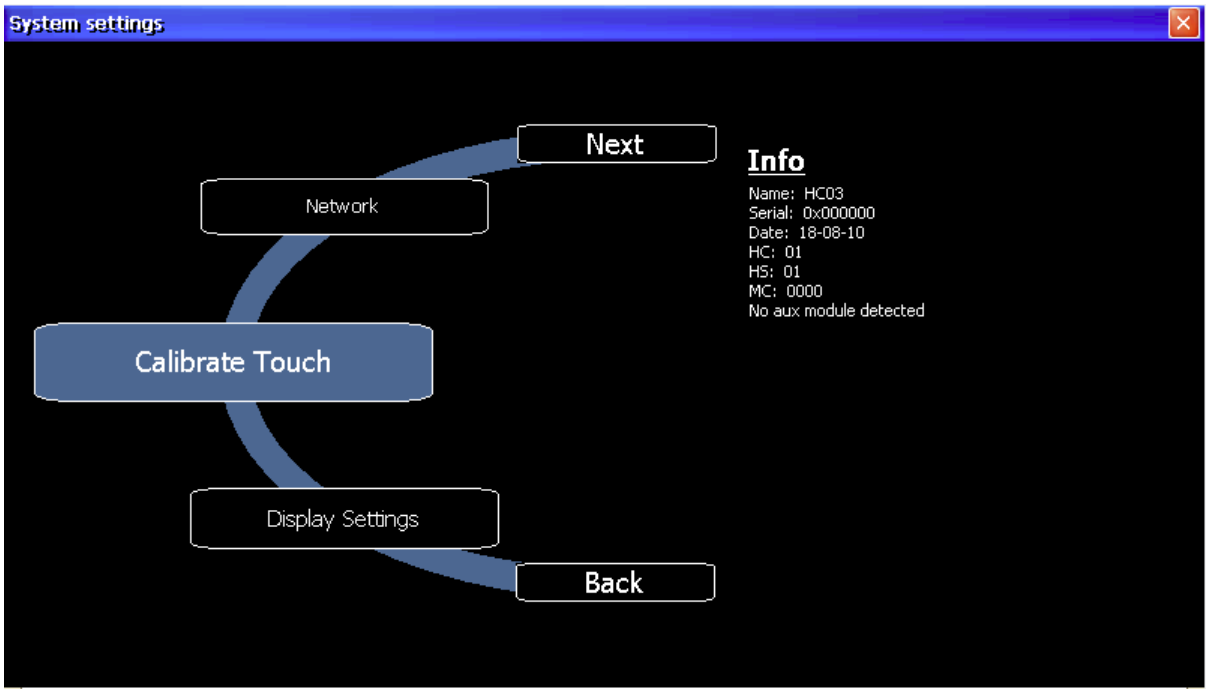

Figure 7

3.1 Other Context menu options

The context menu has several other options as follows:

Zoom In/Out/100% Allow to zoom the view at runtime

Pan Mode Enables/Disables the pan mode; works only when you have first activated a zoom in

Settings Provides options to set the Context menu activation delay, to enable/disable the visualization of the mouse pointer, to show the busy cursor (hourglass) when the system is busy with a long operation, to enable/disable the use of an external physical keyboard. The figure below shows how the Settings dialog looks on the unit.

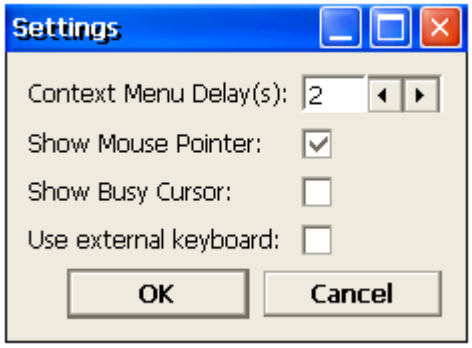

Figure 8

Project Manager When activated, a dialog box will appear (see figure below) providing options to unload (de-activate) the current project, load (activate) another project present on the panel memory, or delete a project. Please note that projects can be deleted only after they are unloaded. If you click on a project name different than the active one, the option "Load project" will first unload the running application and then automatically activate the new one.

Note: Project manager is supported starting from runtime version 1.60 and above. For older versions please refer to “Manage Target” chapter in this document to know about how to switch projects on device from VisuControl software.

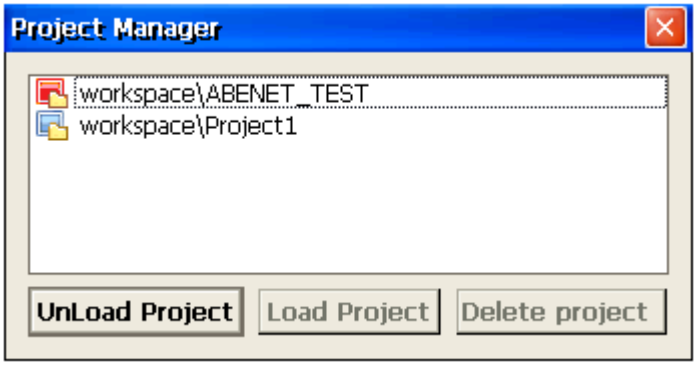

Figure 9

Update When activated, the panel verifies first the presence of an external USB pen drive inserted in the panel USB port, and later for the presence on its root folder of the update package. See the “Updating Runtime from USB Pen Drive“ chapter in this document for further information

Logging Enables you to display a trace of the system operation log; may be very useful in case there is a need to debug a problem of any nature. The following figure shows a case in which the system reports a communication error; the decoding of the reported information may not be immediate, but you can always use the option “Log to file” to save the dialog context to a file that can be later provided to Technical Support for investigation. The log file is called “logger.txt” and it saved to the folder “..\var\log” on the panel flash disk. The file can be retrieved from the panel using an FTP client.

The “Log TO FILE” Option is saved and retained among power cycles; when not needed any more it must be manually deactivated.

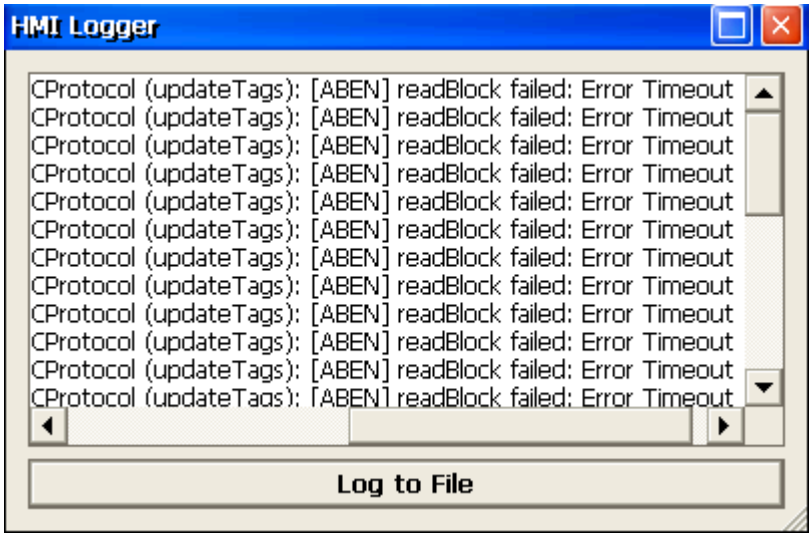

Figure 10

Show log at boot	Enables the logger at start up; if the “Log to file” option has been enabled, the files are saved, in this case, since the startup phase.
About	Shows information about the runtime version

3.2 Built in SNTP Service

The VisuControl Touchpanels Operating System features 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 VisuControl Project

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

4.1 Creating a New Project

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

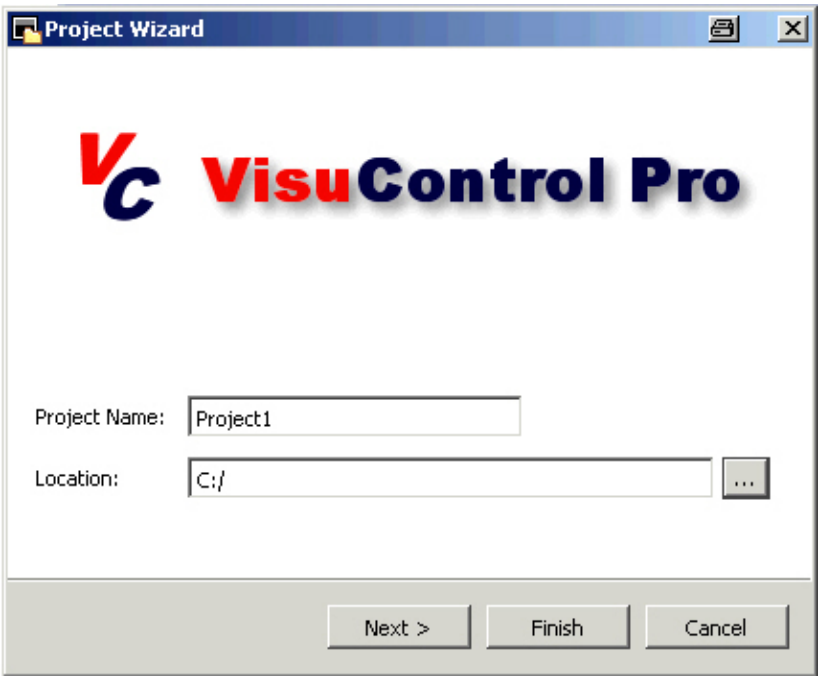

Figure 11

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

VisuControl 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 VisuControl 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.

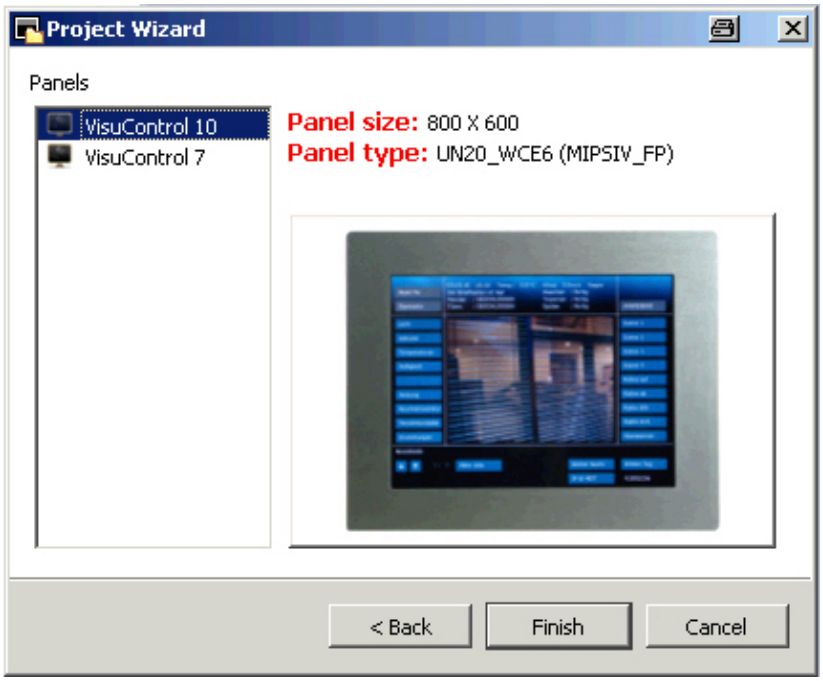

Figure 12

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

Per each model two radio buttons are available to select the orientation: landscape (default) or portrait. In portrait mode the panel is rotated 90° clockwise.

Note: Portrait mode has some limitations. All the System settings dialogs do not support the portrait mode and they do appear still as horizontal.

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.

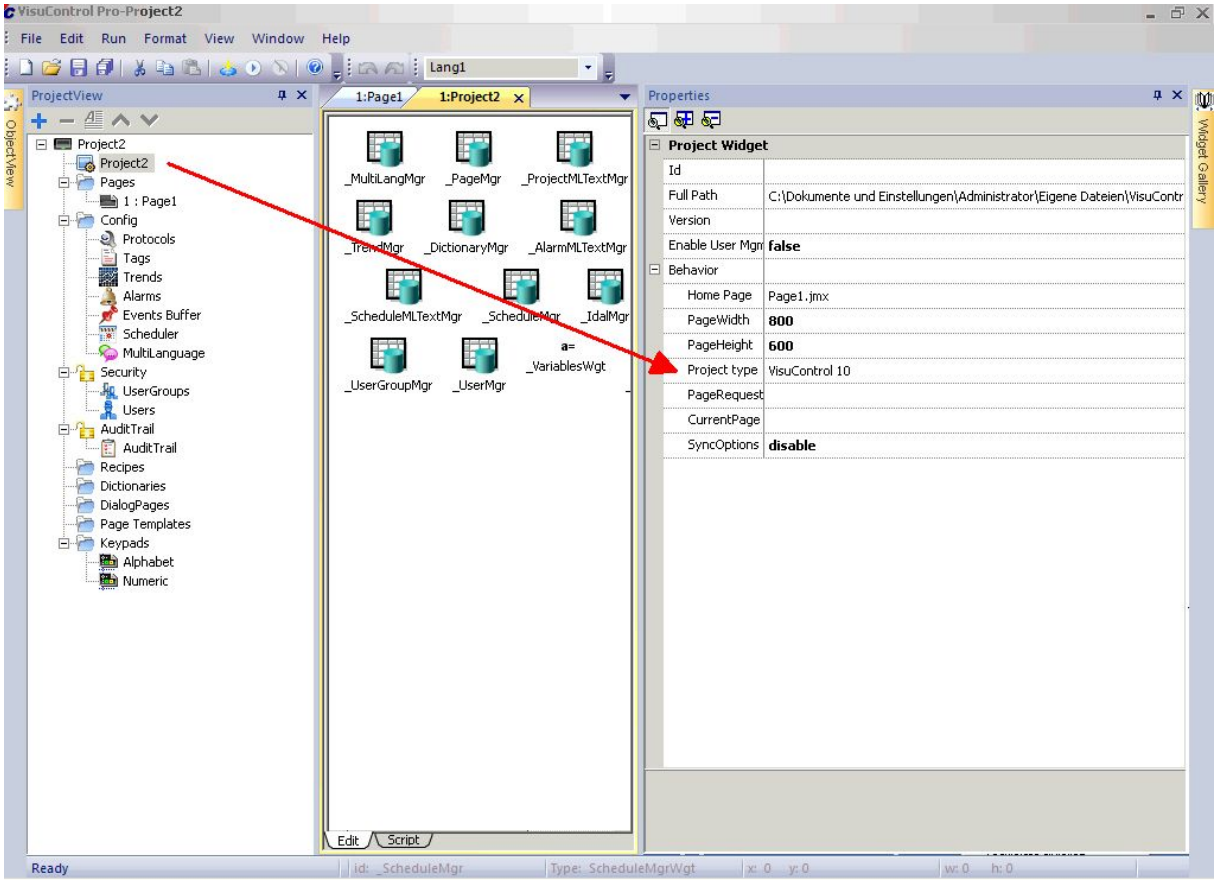

Figure 13

4.2 Workspace

The VisuControl workspace is divided into three main areas.

On the left-hand side of the window, you will find the Project View and Object View (as a sliding tab). 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.

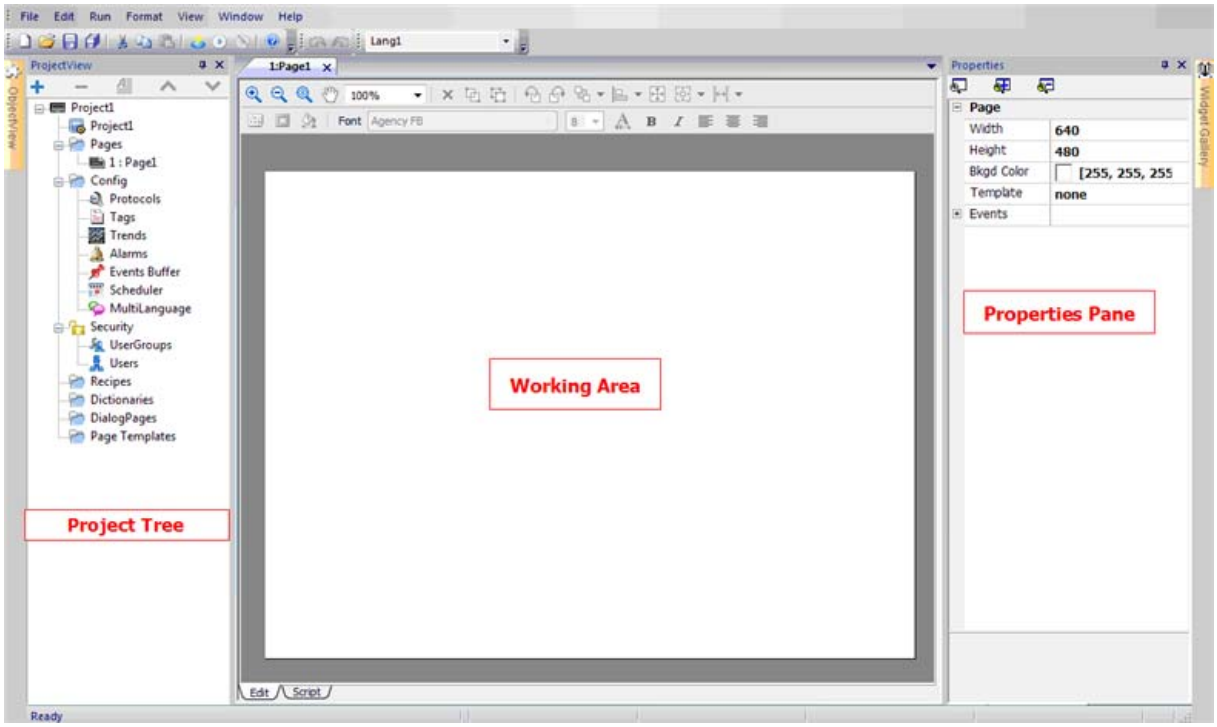

Figure 14

Note: The VisuControl workspace layout can be freely changed at any time; any change is saved and maintained among VisuControl activations. In case you need to reset the workspace to the original default layout, there is the command called “Reset and Restart” from the File menu.

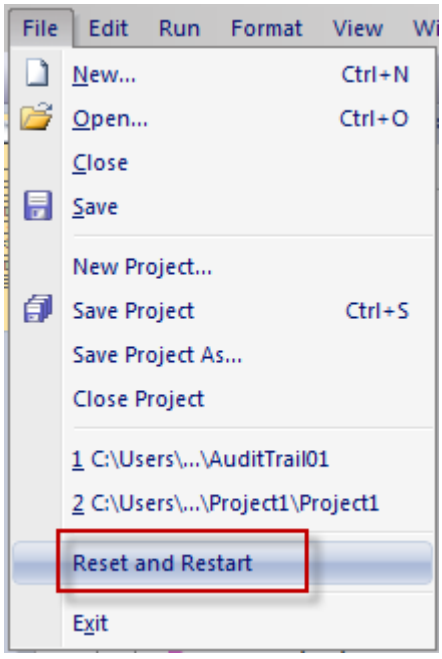

Figure 15

4.3 Select the KNX Communication Protocol

Device Communication drivers are configured in the “Protocol Editor", which is accessible from the project tree (as shown in below figure). 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.

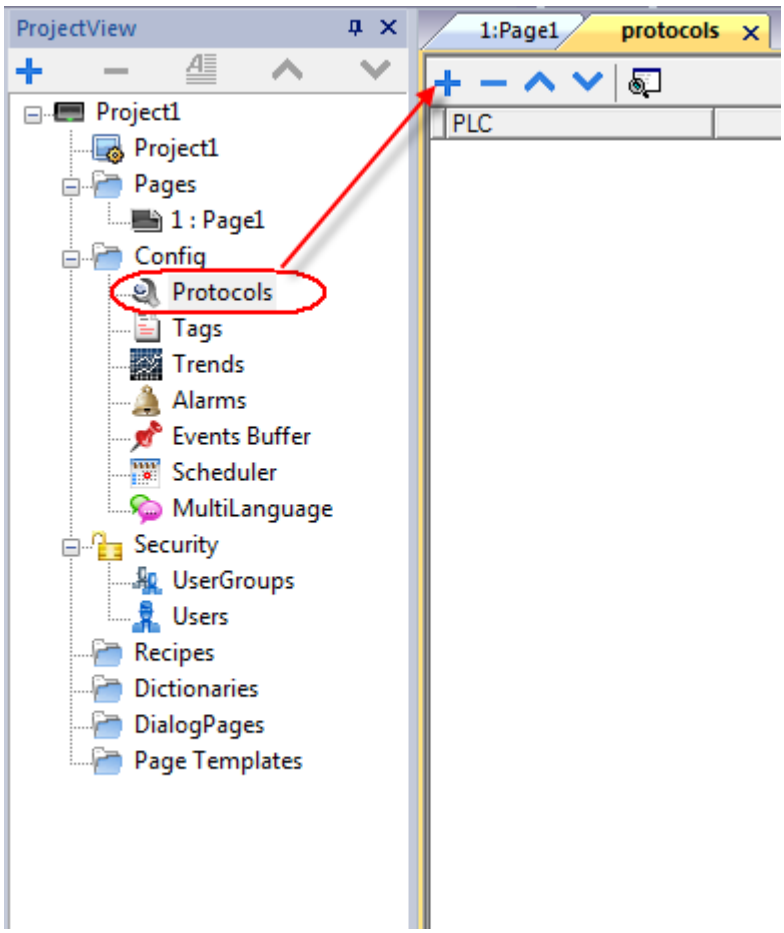

Figure 16

4.3.1 Configuring the Driver

The combo box shows the KNX drivers. Once the 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 below figure)

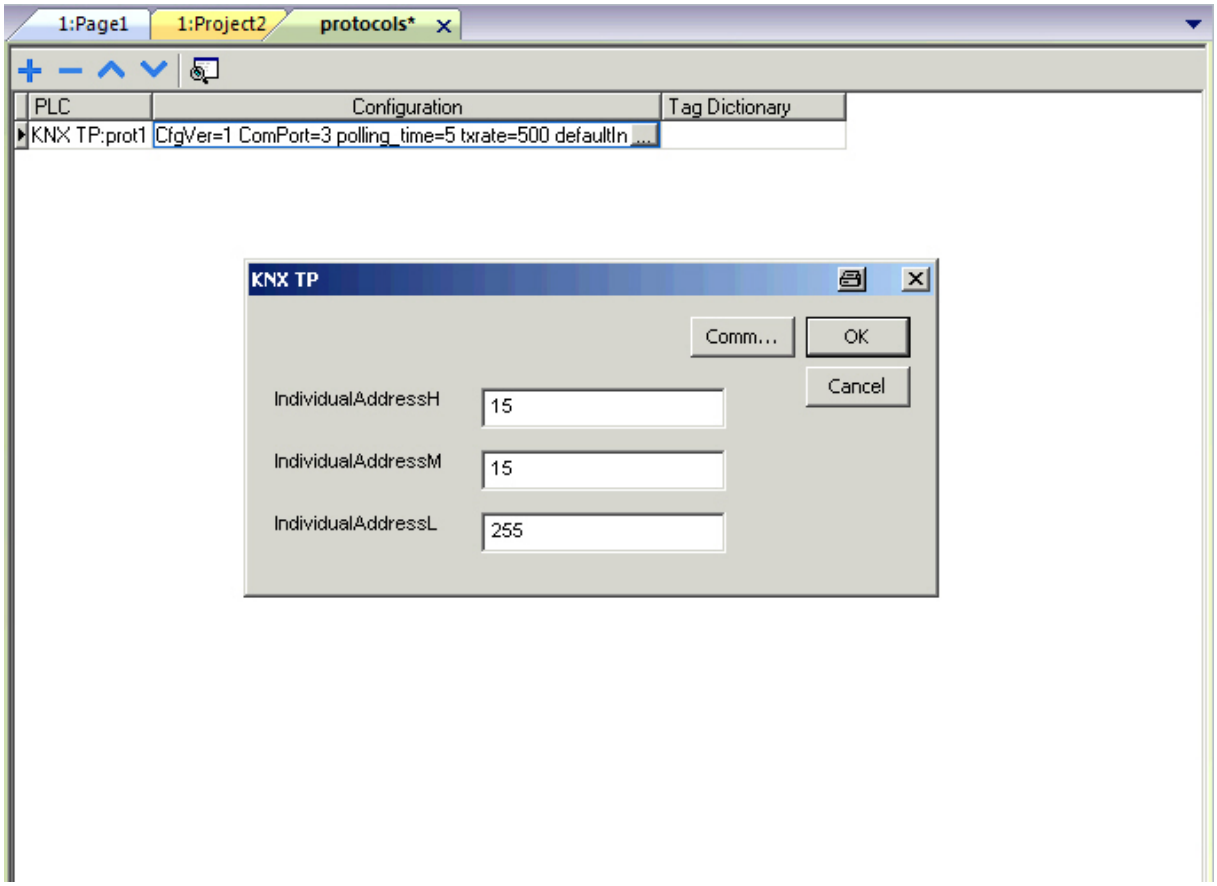

Figure 17

The other two Columns in the Protocol Editor are:

Tag Dictionary This shows the tags that have been imported for a Particular Protocol.

Prefix This Specifies an alias property for each Protocol. Names can be given to each Protocol.The tag name must be always unique at project level; often it may happen that the same tags, from the same symbol file have to be used for two different controllers. Since having tags with same name is not supported, you can use this feature to automatically add a prefix to the imported tag to make them unique at project level.

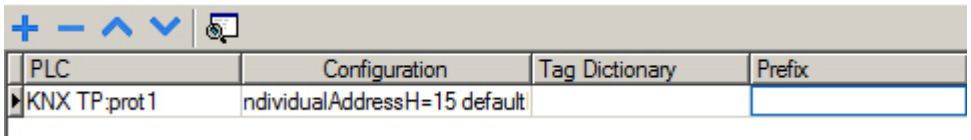

Figure 18

4.4 Add the Tags

VisuControl 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 below figure).

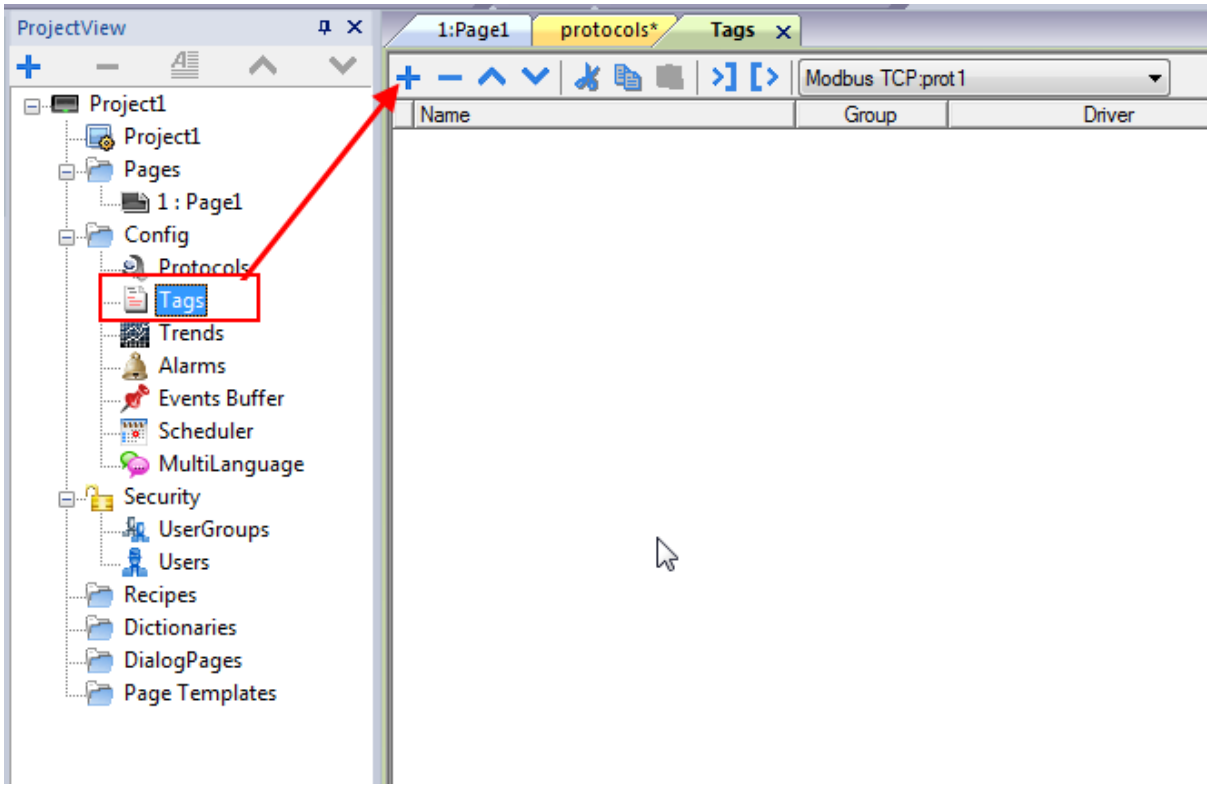

Figure 19

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 VisuControl provides the Tag Import feature, first select the Protocol from the filter button and then click on the Import button (as shown in below figure).

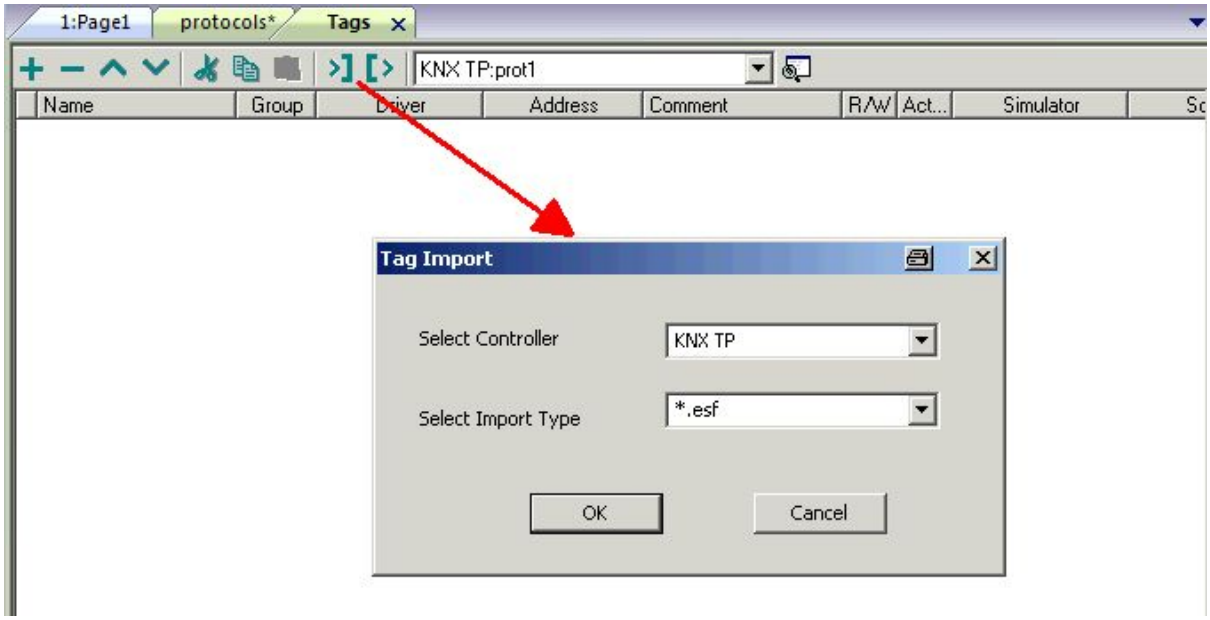

Figure 20

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 VisuControl Tag Editor

The tool in VisuControl designed to handle tags' creation and managing is called Tag Editor.
For 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.

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 "Attach To" 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 you 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 external software communicates with the local VisuControl runtime requesting sets of data that must be independent from the currently displayed screen.
The VisuControl web server publishes 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 you to define the users' groups and assign tags to them.

Driver Specifies the communication protocol (KNX) 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 information.

Comment Allows you to add a description of the tag.

R/W This option determines if the tag must be managed as Read only (R), Write only (W) or Read/Write (R/W).

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. 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 that you leave this parameter to false to avoid unexpected results in terms of overall device performances.

Simulator VisuControl provides off-line simulation. The behavior of each tag during simulation mode can be specified here by choosing between several profiles as shown in below figure.

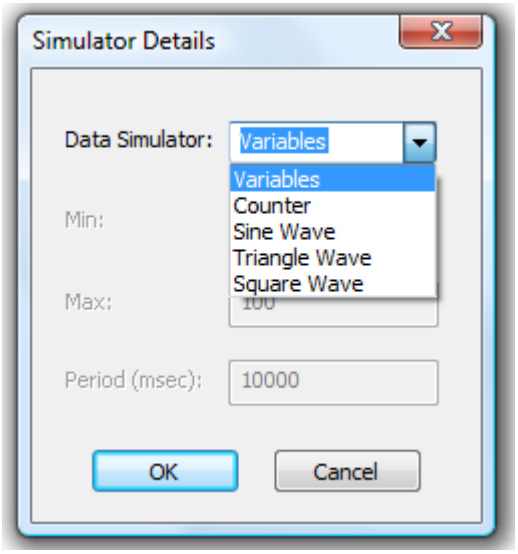

Figure 21

Scaling Tags' values are normally transferred "as they are" from the protocol to the real time tag database. You can specifically apply scaling to the tag values before they are stored in the database. The available scaling options are shown in the figure below. Scaling can be specified in terms of linear relationship as formula or as range.

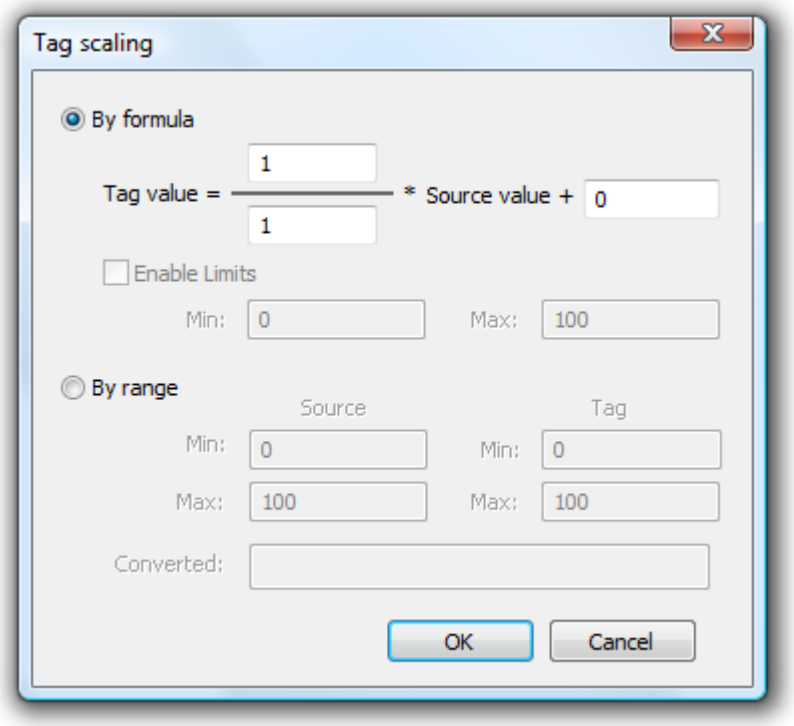

Figure 22

4.4.2 VisuControl Data Types

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

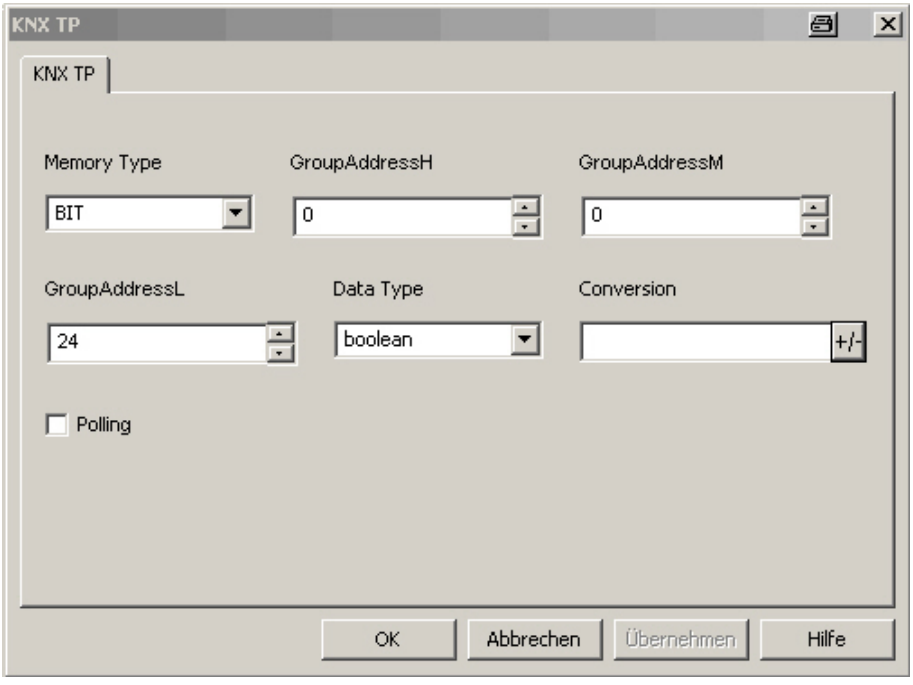

Figure 23

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

Data Type	Description
string	Character strings. The characters are coded in UTF-8 format.
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

Note: String coding in UTF-8 format is supported from runtime version V1.80

4.5 Design 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.

To add a new page, right click on the Page node from the project tree and select "Insert new page". A dialog box will appear asking for the name of the new page.

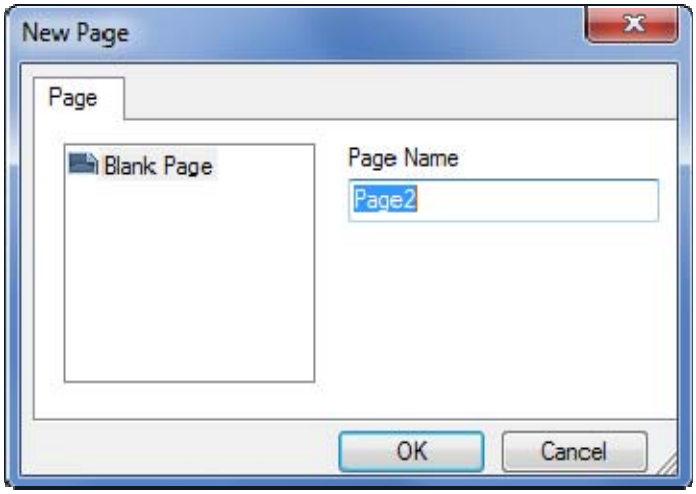

Figure 24

4.5.1 Import a Page

A page can also be imported from one Project to another. By right clicking on the page folder in the Project View, there is an option named “Page Import”. Please refer to the Figure below.

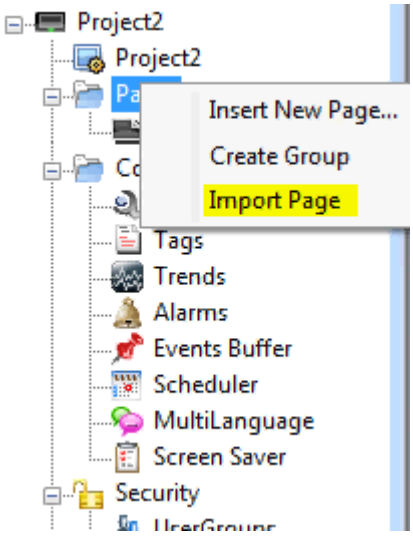

Figure 25

After selecting a page to be imported from the desired project, when you click OK, you get a warning message in VisuControl as given in the Figure below.

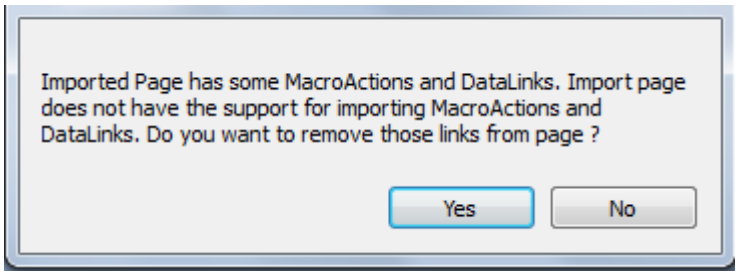

Figure 26

Please note that there are some Macro actions and Datalinks in the imported page. Page Import can support only import of page and the widgets in it, but not the Macro actions and datalinks attached to the widget. By selecting “Yes” all the datalinks and the Macro actions attached to the widgets will be removed. Only the Widgets will remain. By selecting “No” the macro actions and the datalinks will remain attached to the widgets, but it may not function properly during runtime unless and until the tags associated to the Macro and datalinks are created in the new project.

Note: The import page can be done between projects made in the same version of VisuControl. If the versions are different then a warning message will pop up to save the project in the new version, then again try to import the page.

4.6 The VisuControl 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 below figure).

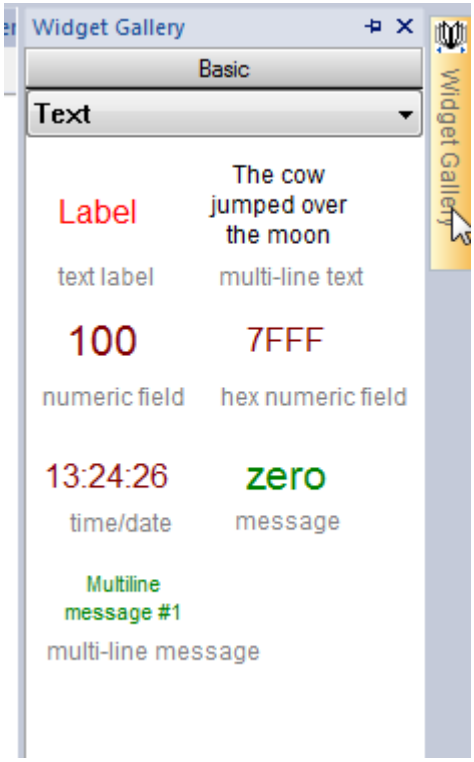

Figure 27

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 the property pane 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).

Below figure shows the Widget style button for round gauges.

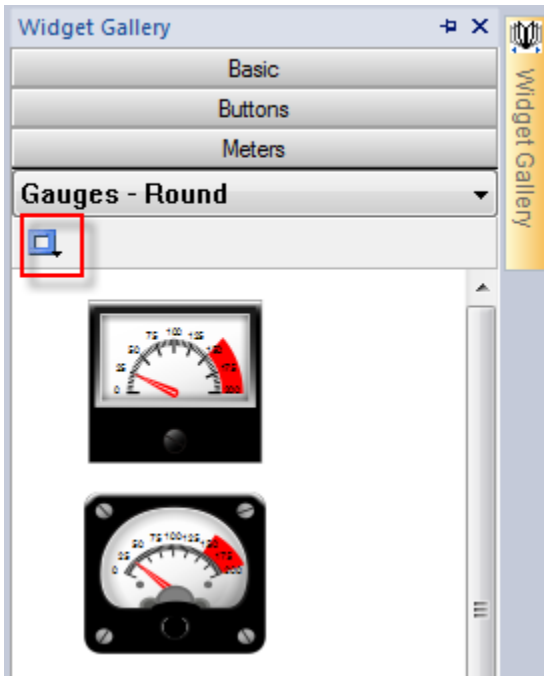

Figure 28

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 the page, the object can still be subject to additional style changes.
This is done using the Page Toolbar shown in below figure.

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

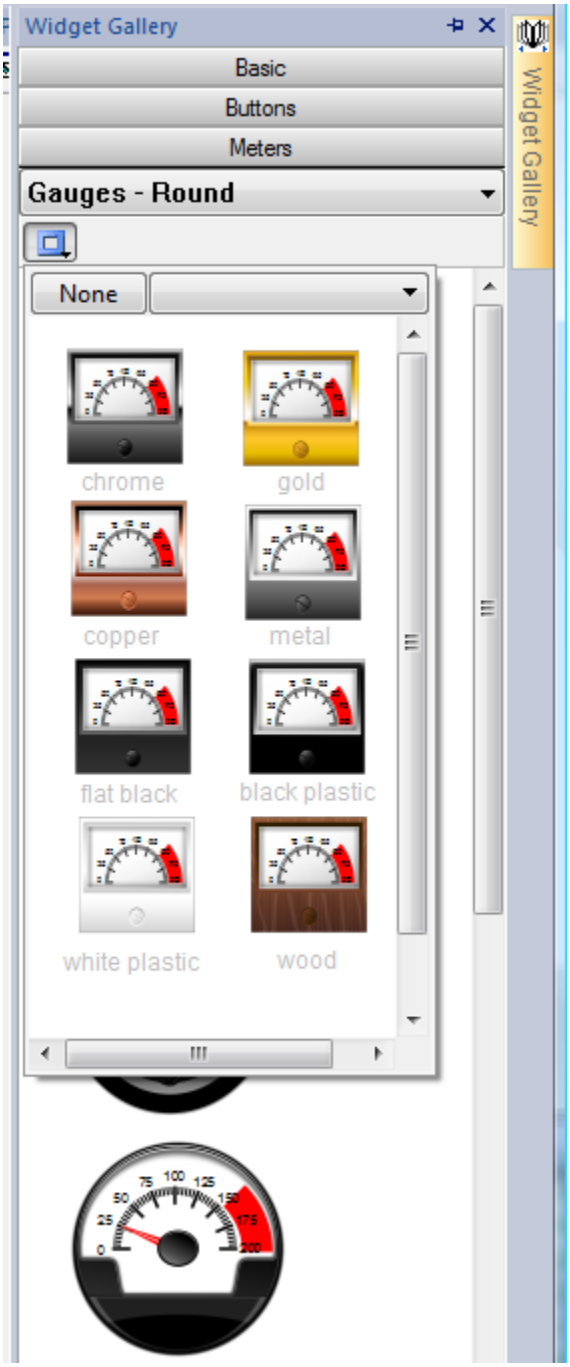
Widget Gallery
Basic
Buttons
Meters
Gauges - Round
None
chrome
gold
copper
metal
flat black
black plastic
white plastic
wood
Widget Gallery

Figure 29

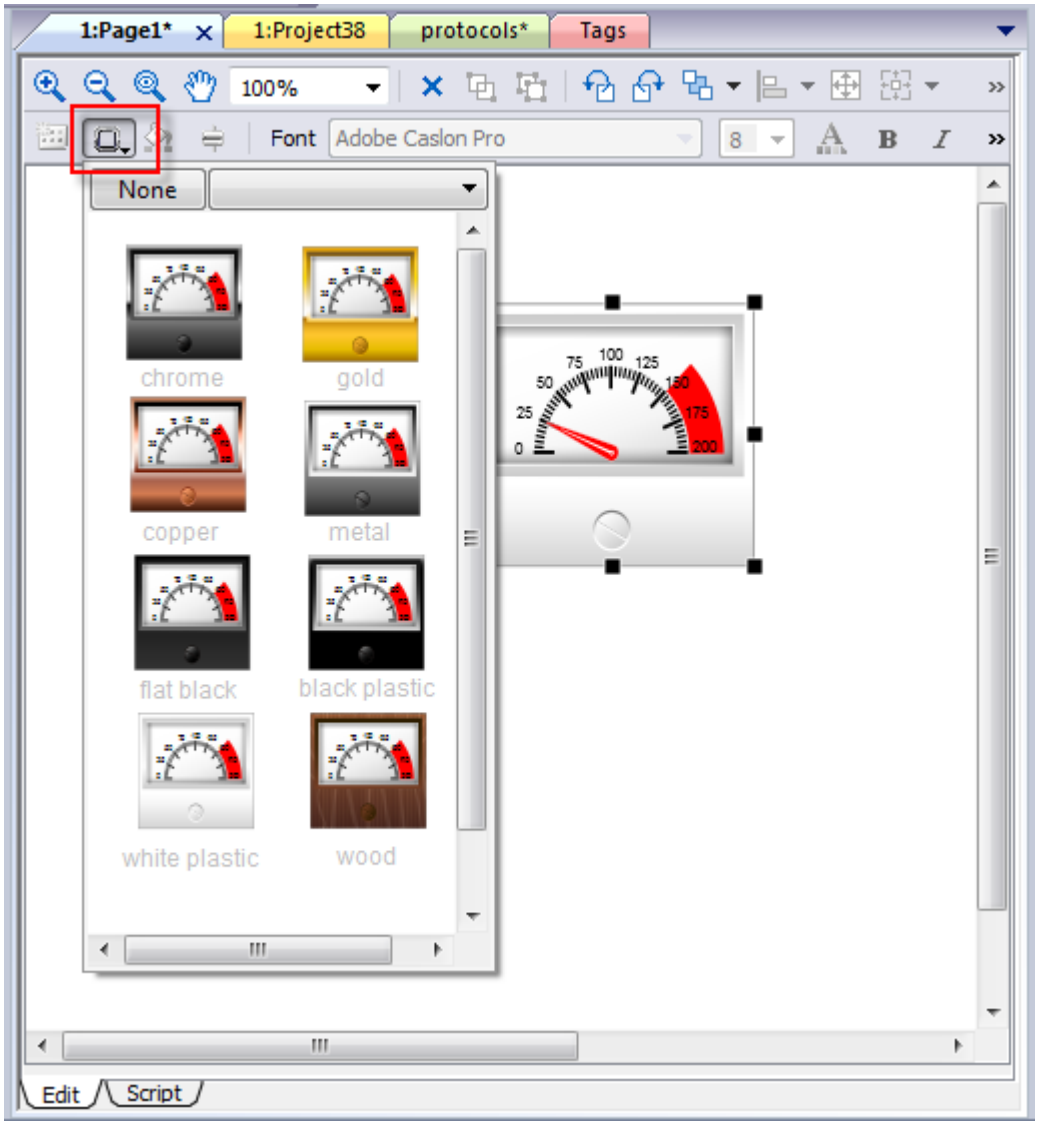

Figure 30

4.7 "Attach To" and dynamic properties

VisuControl 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. Click on this button and select the item Attach To… from the menu (as shown in below figure).

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.

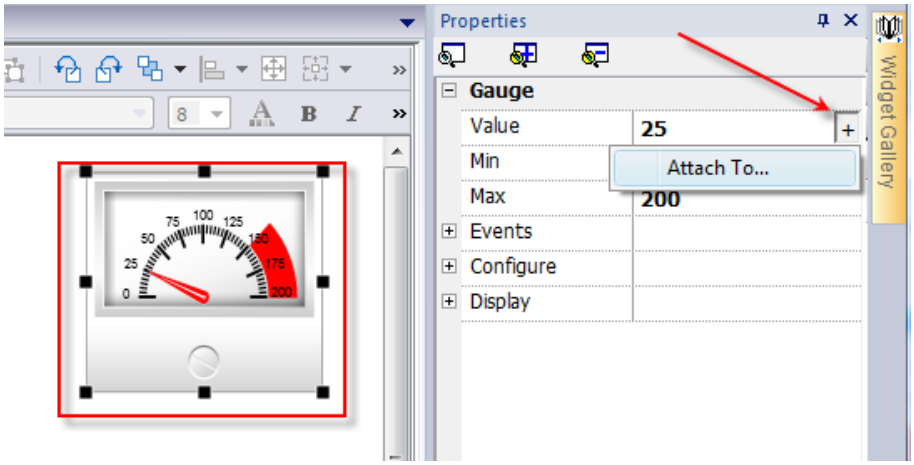

Figure 31

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 above figure. Select the Attach To… menu item and a dialog will be displayed as shown in below figure.

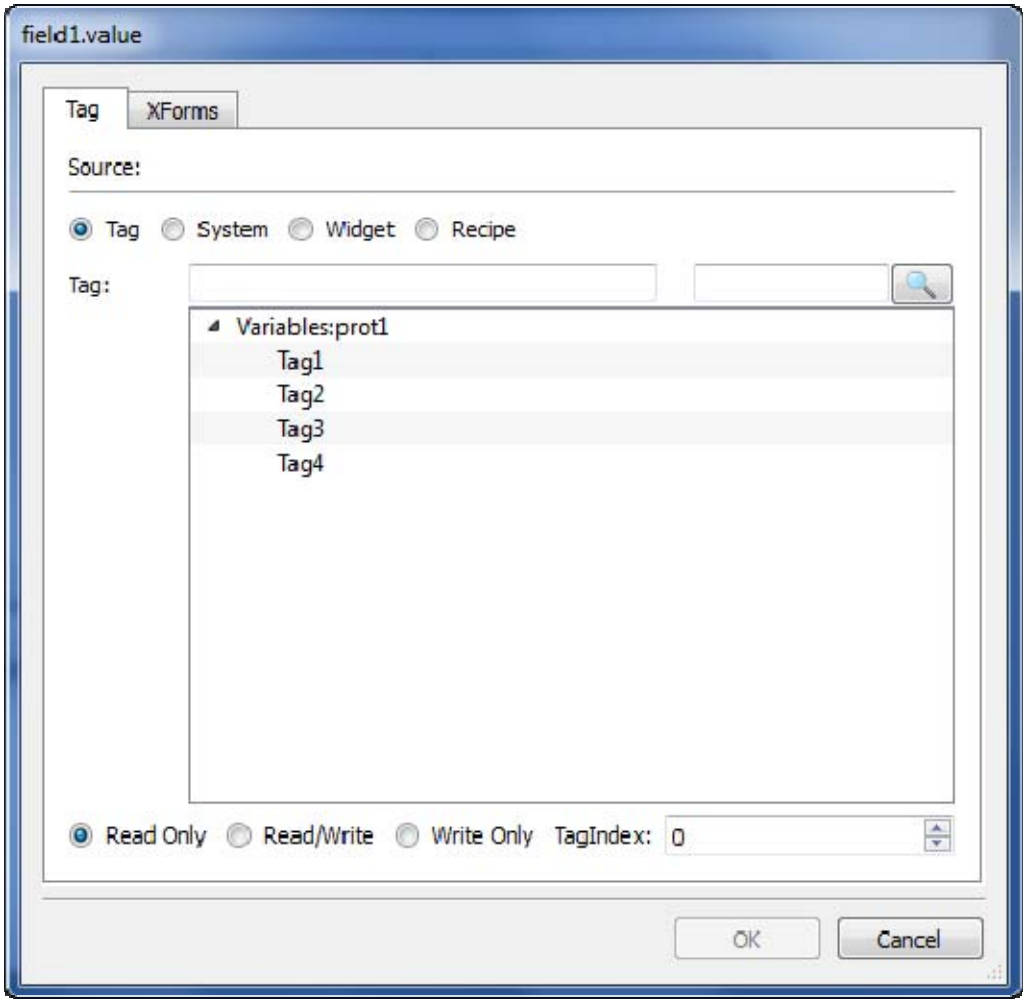

Figure 32

When attaching a Tag, you can attach four types of data sources: Tag, System Tag, another object property (called a Widget) and Recipes.

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.

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 To… popup menu. You can also right-click on the object and select the Attach To… 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.

Note: a specific chapter later in this document describes in detail the “Attach to” concept

4.8 Testing the Project

With VisuControl you can test the project functionality before downloading it to the Target device. VisuControl 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 below figure. 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.

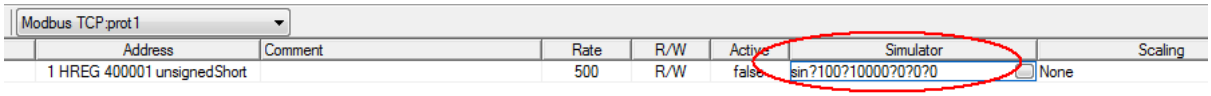

Figure 33

The VisuControl Simulator is launched from VisuControl. 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 VisuControl 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 Transferring the Project to Target

The VisuControl project can be transferred to the VisuControl Touchpanel by using the “Download to Target" item in the Run Menu.

Note: The project can be transferred to the device via Ethernet or via USB pen drive. In case the project is downloaded to the unit via Ethernet, the panel must have assigned a valid IP address, static or dynamic via DHCP server. Please see the chapter “Unit basic settings” for further information about how to assign an IP address to the panel.

The Download to Target dialog is shown in the figure below.
Once the panel has a valid IP assigned, it will become discoverable on the local network. Click on the “discovery” button as indicated in the figure below and click on the desired IP address.

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 needs to be transferred to the Target device. When updating a Target, VisuControl provides the option “Download only changes" to transfer only the modified files to the device.

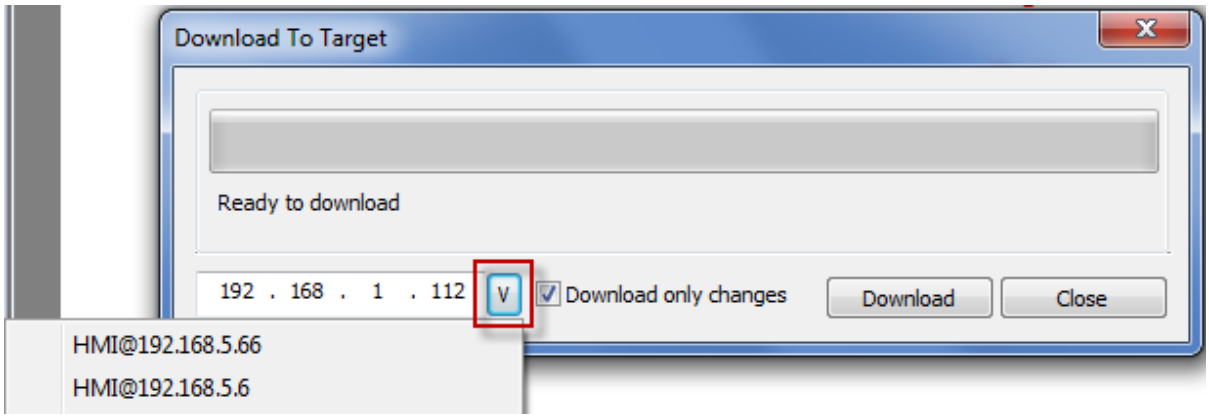

Figure 34

When transferring a project, VisuControl is using a combination of an HTTP and FTP connection. The HTTP connection is used to issue to the target device commands like “turn in transfer mode” or “unload running project”; the FTP session is instead used to transfer the files to the panel flash disk.

The Default port for HTTP connections on the Target is set as 80. However, the user can change the port number to a different value. To set the port number, click on the **Run>Manage Target**, then click on **Target Setup on the dialog.** The HTTP, FTP port or HTTPS, FTPS port can be set assign for the target.

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

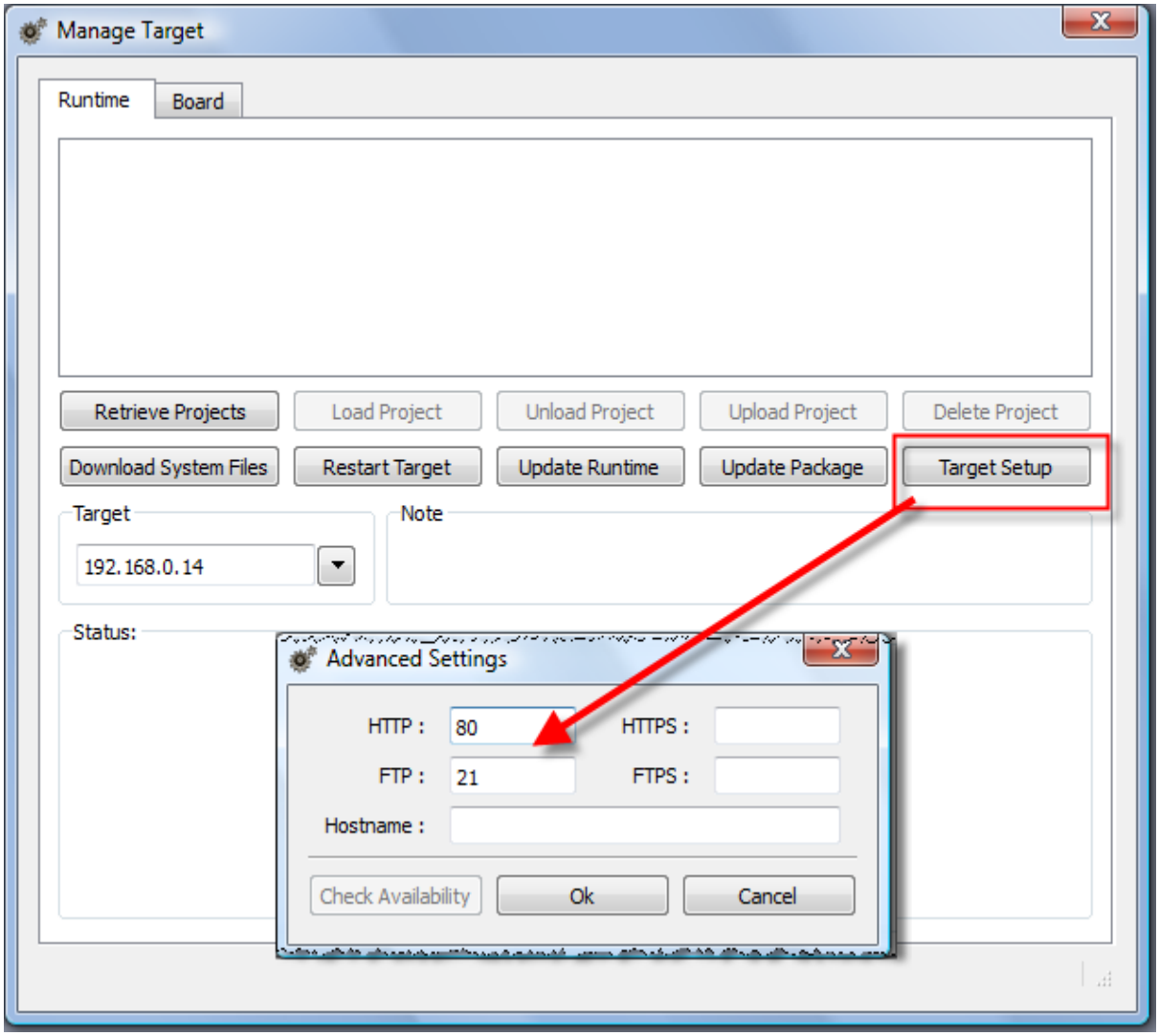

Figure 35

Note: Transferring a project after the above settings will result in a change of the default configuration. At the next download the new ports will be used on target and the new ports will have to be specified in VisuControl to match the new selection.

In the download dialog, click on Advanced Menu and set VisuControl as the port.

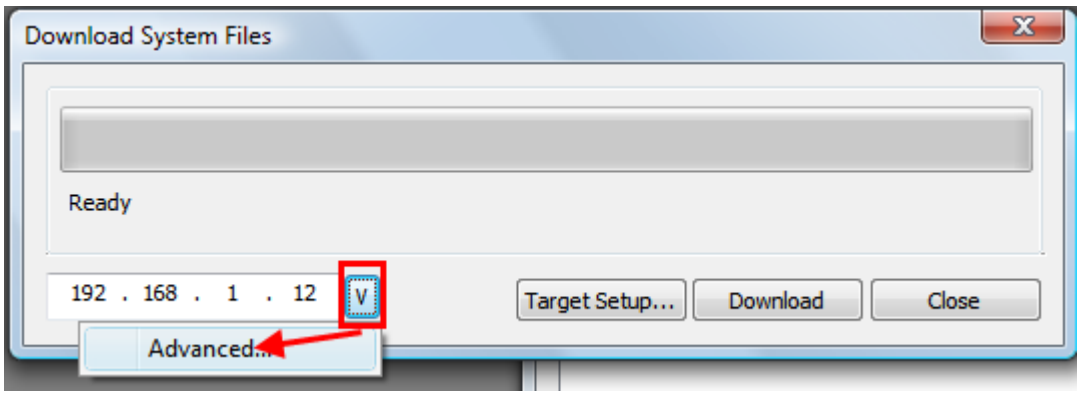

Figure 36

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

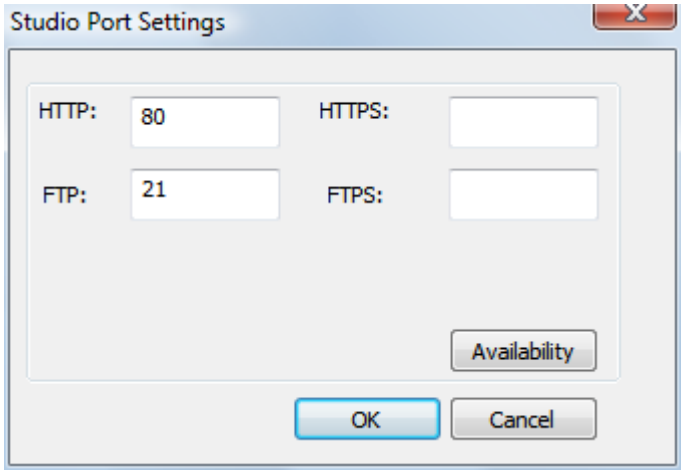

Figure 37

The HMI system also supports the ability to transfer a project to the panel device using a USB pen drive. To transfer the project via an external USB flash disk, you need first to create the so called “Update Package”.

From the “Run” menu click on “Manage Target” and select then “Update Package” as shown in the following figure.

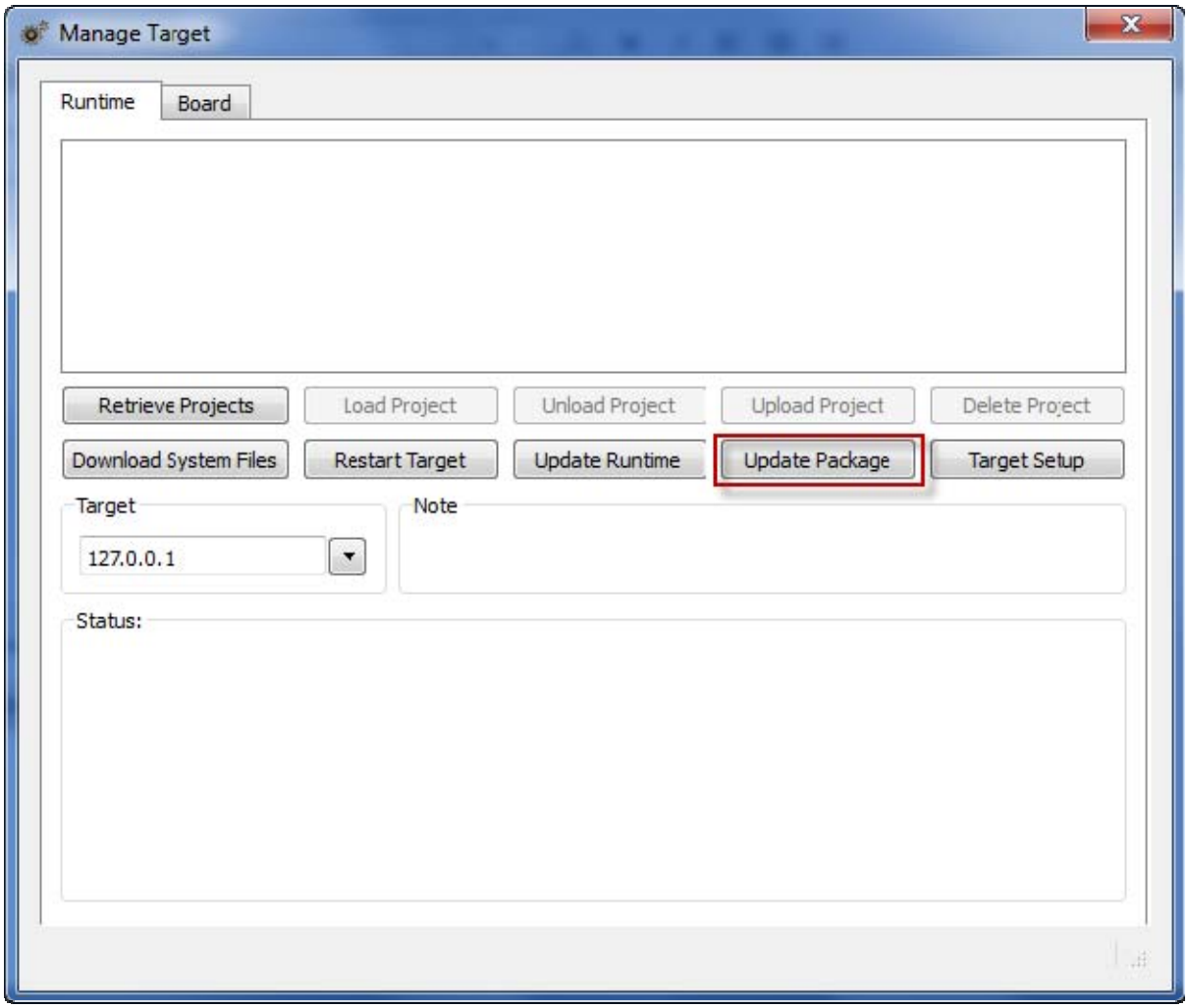

Figure 38

VisuControl will show a dialog where you are asked to specify the Target type, what to update (we are transferring a project, so only the first check box has to be marked), the path where to store the result of the operation (it can be directly the root folder of a USB pen drive plugged to a USB port on the PC), the compression options (recommended to save space) and encryption option.

Encryption option, when marked, creates a zip file protected by a password only known to VisuControl and runtime. This password makes sure the zip file is not opened by un-authorized personnel and its contents not changed for any reason. The password an not bei Change. Once ready, click on the “Create” button to generate the update package.

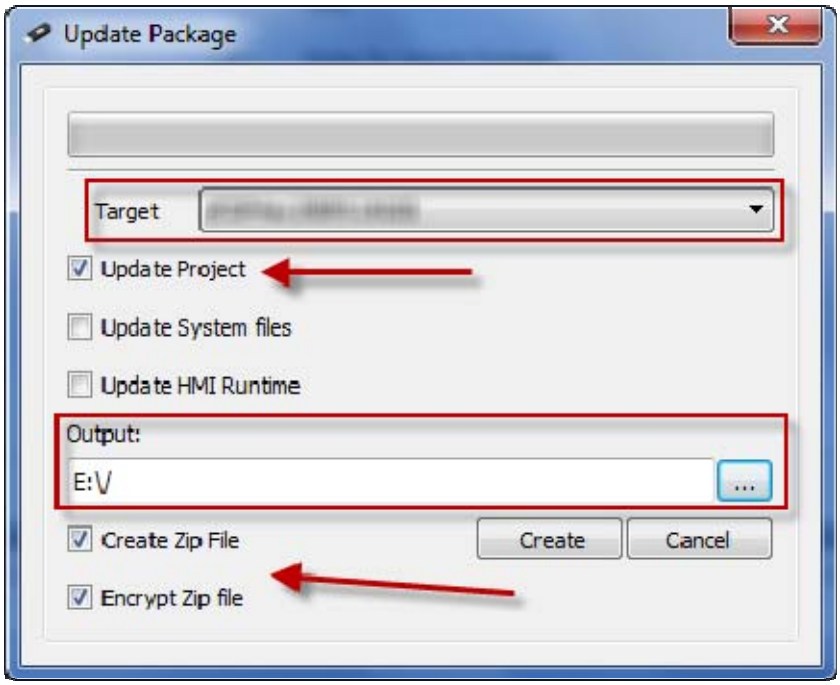

Figure 39

Assuming you have stored the package on the root folder of a USB drive, remove the drive from the PC, plug into the panel, activate the context menu by holding your finger for a few seconds on the screen (see also "Basic Unit Settings")and select "Update…" as shown in the following figure.

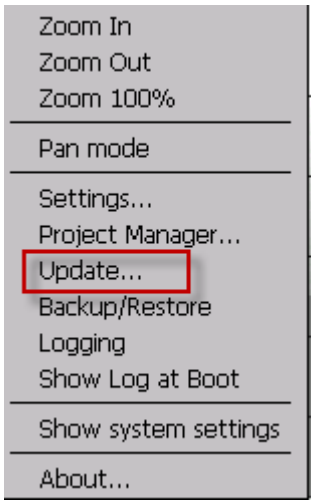

Figure 40

The system will automatically check for the update package on the root of the USB drive and ask to proceed with the update according to the following figure.

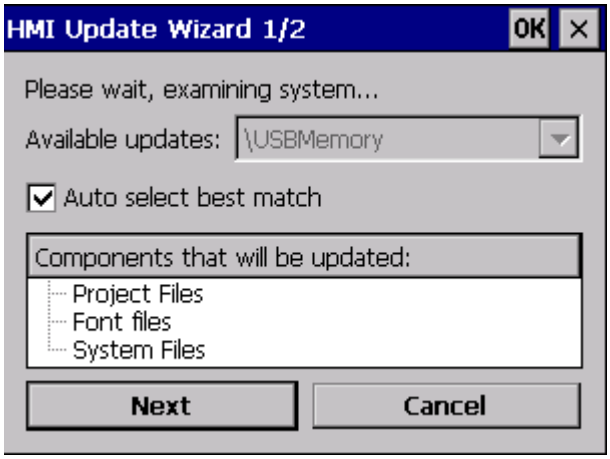

Figure 41

Mark the “Auto select best match” check box and click on the “Next” button.
The rest is automatically done by the system.

Note: The Update Package can be also used to update the panel runtime. For this please see the chapter “Updating Runtime from USB Pen Drive“

4.9.1 When Target Flash Memory is Low.

While trying to download a project to the Target if the project size is almost near or greater than the free space available in the flash memory, then it’s not possible to download the project directly. The difference of project size and available free memory should be at least 2MB.

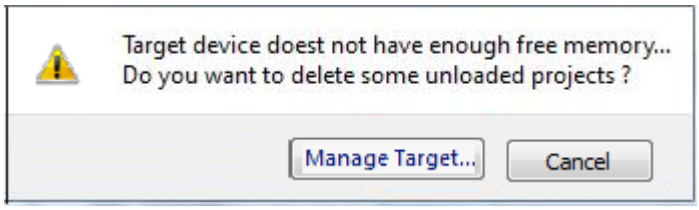

Figure 42

While clicking Download, a warning message will pop up mentioning that the Target memory is low and whether you need to delete some projects as in the figure given above. Soon after you click “Manage Target”, Manage Target window will open showing all the available projects in the Target. Deleting the unwanted projects from the target creates more memory space, hence making it possible to download the current project. By pressing Cancel, the dialog will close, and the download operation aborted.

Note: the automatic check for available space for project download is a feature available from VisuControl and runtime version 1.80.

4.9.2 The Runtime Loader

The explanations provided in the previous chapters are valid when using a panel with runtime system already installed.
The HMI devices are delivered from the factory without the runtime. When you power up the unit for the first time, it starts with the "**Runtime Loader**" screen as shown below.

Figure 43

When you click on "System setting" you can activate the System menu in User mode, where you can set the IP address of the panel. See the chapter "System settings tool" for additional information on this tool.

Once the IP address is assigned and the panel is connected to a valid network, the easiest way to install the runtime is to download a project from VisuControl. See the chapter "Transferring the Project to Target" for additional information.

The normal download procedure in VisuControl is able to recognize the needs of transferring the runtime and the process is automatically started.

As soon as the panel IP is selected from the list of available units in the network, VisuControl will in fact recognize the need for transferring the runtime, providing the information as shown in the following figure.

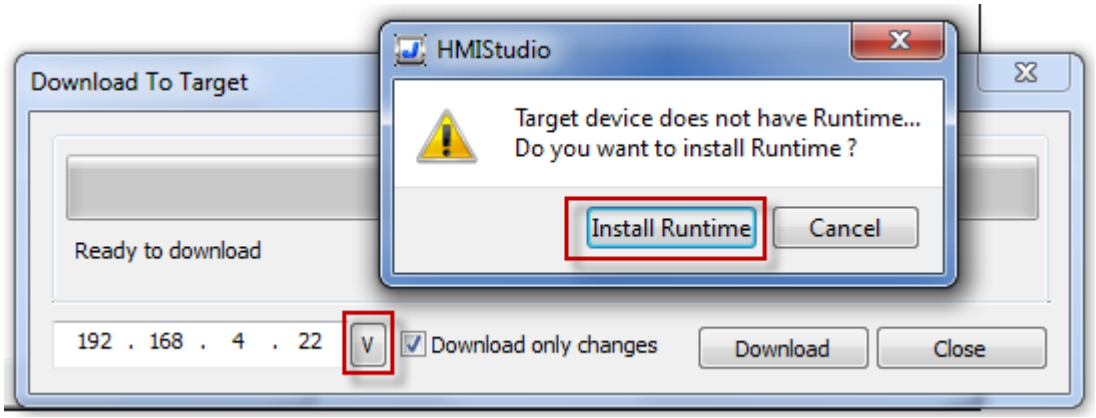

Figure 44

Just click on Install runtime button to proceed.
The process will automatically go through the required steps, ending with the project download.

On an off-the-shelf unit the runtime can be installed also using an USB pen drive.

Prepare the Update Package according to the instructions provided in the chapter "Transferring the Project to Target" and make sure to mark all the check boxes for the HMI Runtime and System files components, too, as shown in the following figure.

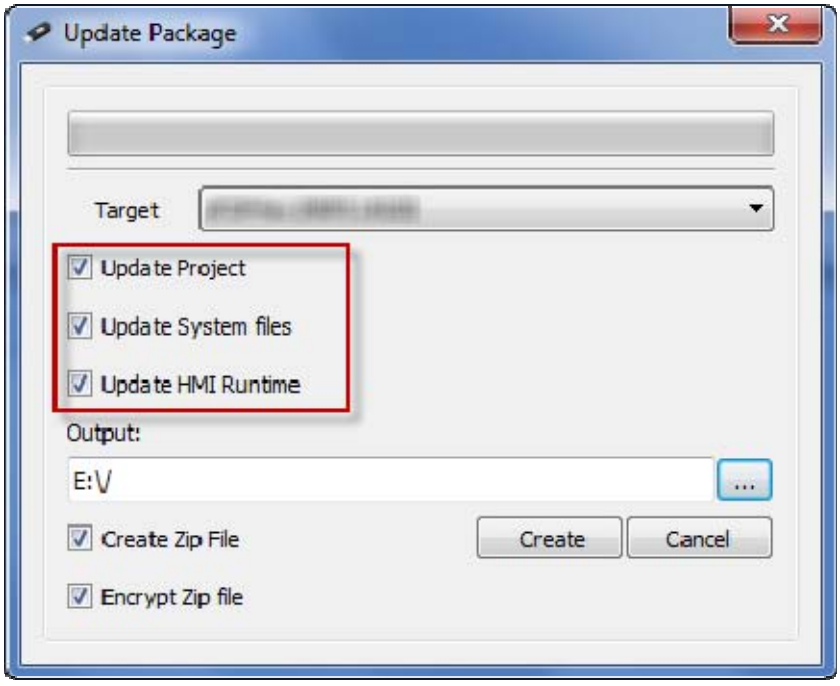

Figure 45

Then plug the USB pen drive in the panel and click on the “Transfer from disk” button as below.

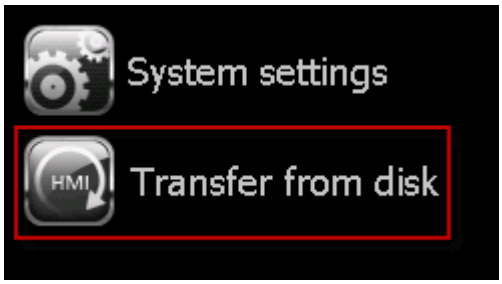

Figure 46

Follow then the instructions on the screen.

4.10 Using VisuControl Client

The VisuControl Client provides remote access to the VisuControl Touchpanel, and is included in the VisuControl installation.

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

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

4.10.1 Time zone options for Client

The Windows remote client provides an additional option to handle the visualization of the timestamp information of a project.

From the "Settings" dialog you now have access to a set of new options shown in the next figure:

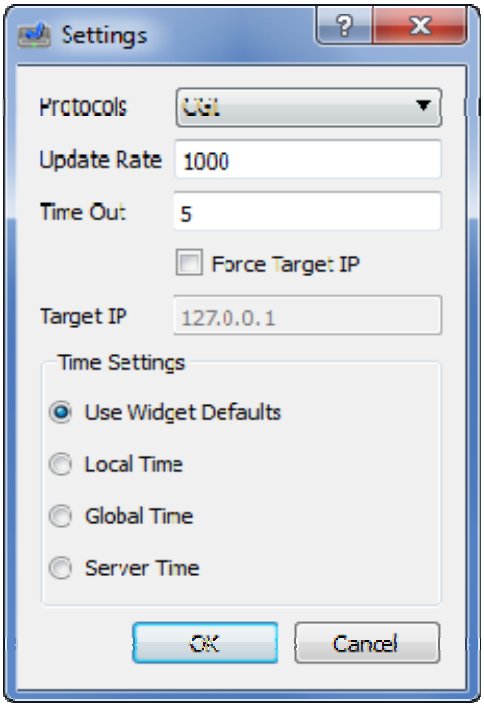

Figure 47

The Time settings information is used by the client to adapt the widget timestamp information according to the desired behavior.

Widget Defaults	Allows you to display the time information per each widget according to the widget settings provided at programming time in VisuControl.
Local Time	Translates all the timestamp information of the several widgets used in the project into the PC local time where the client is installed.
Global Time	Provides all the timestamp information with reference to the UTC time (Coordinated Universal Time).
Server Time	Translates all the timestamp information of the widgets used in the project in order to show the time as it is at panel side.
Note:	to be properly used this feature requires you to set the panel RCT with the correct zone and DST (Daylight Saving Time) options.

4.11 Using the integrated FTP Server

The HMI runtime system is featuring an integrated FTP server that can be used to get access to the internal flash disk data.

Note: folders present on the Flash disk external to the runtime directory are not accessible via FTP; external USB drive and SD Storage Card are not accessible via FTP.

You can use any standard FTP client program to connect to the panel FTP server. The FTP server responds to the standard port 21 when using as host the IP address assigned to the panel.

Note: The FTP server supports only ONE connection at a time; if you are using an FTP client which is configured to multiply the connections to the server in order to speed up the transfer operation, you will need to disable this feature in the client program or set to 1 the maximum number of connections per session.

The FTP server is configured by default to accept incoming connection from the following account:

User name: admin
Password: admin

FTP permissions and account information can be changed from the “UserGroups” under the “Security” item of the project folder as shown in the following figure.

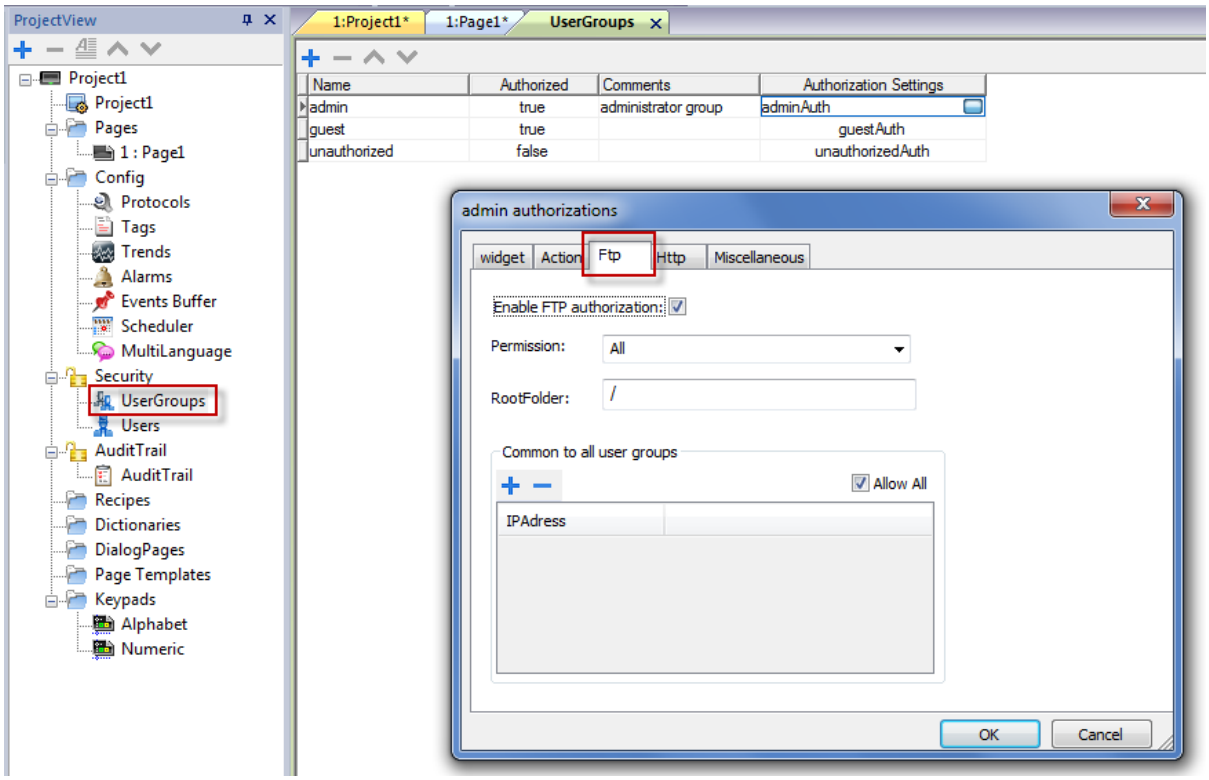

Figure 48

Additional information can be found later in this document in the chapter “FTP Authorizations”

4.12 Using ActiveX Client for Internet Explorer

In the standard distribution of VisuControl, the VisuControl and VisuControl ActiveX Client is 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 need to download the Download Microsoft Visual C++ 2008 Redistributable Package (x86) from the Microsoft web site.

4.12.1 Copy ActiveX into the Target device

The ActiveX component is distributed with the VisuControl installation package. The related files are located in the Runtime folder of the VisuControl installation directory. The files, “HMIAX.cab" and “HMIClientAX.html", should be copied into the workspace folder of the Target device, where Runtime is installed. The file copy can be done using the panel FTP server as described in the dedicated chapter.

4.12.2 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.

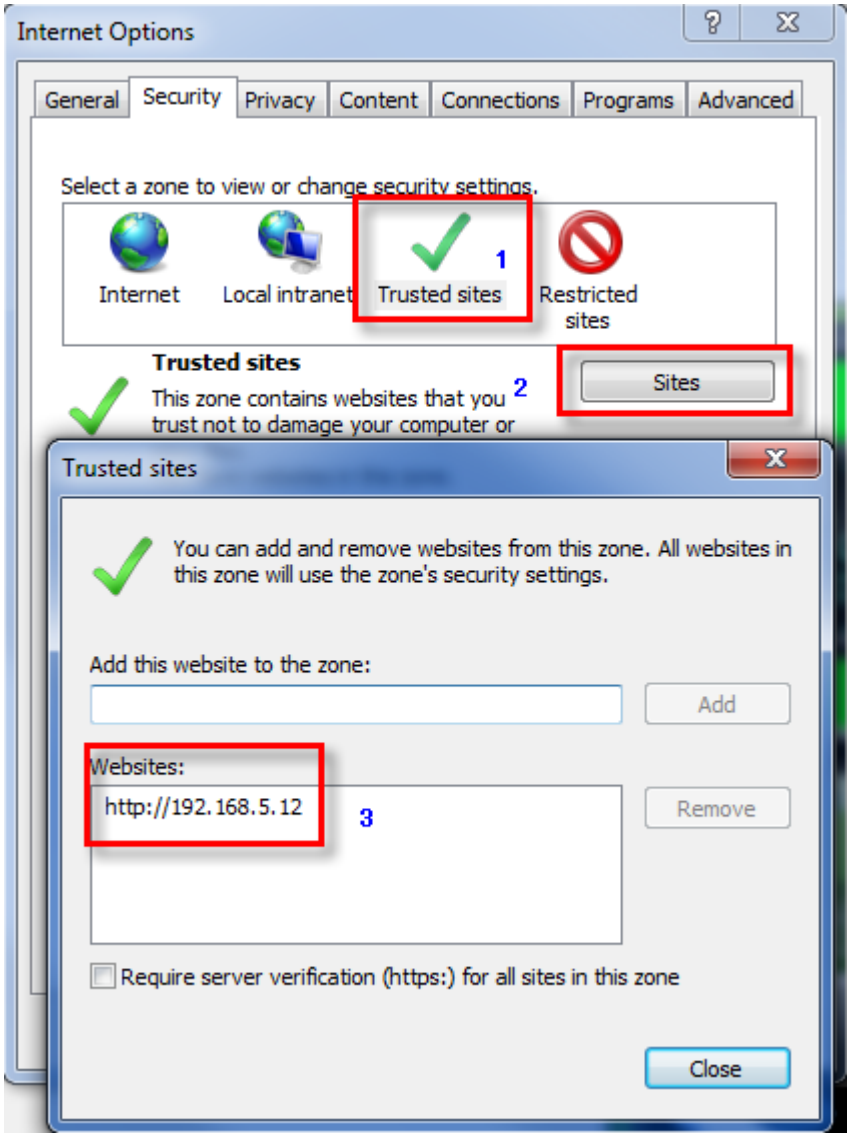
Internet Options
General
Security
Privacy
Content
Connections
Programs
Advanced
Select a zone to view or change security settings.
Internet
Local intranet
Trusted sites
1
Restricted sites
Trusted sites
This zone contains websites that you trust not to damage your computer or
2
Sites
Trusted sites
You 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:
Add
Websites:
http://192.168.5.12
3
Remove
Require server verification (https:) for all sites in this zone
Close

Figure 49

4.12.3 Security Setting for Trusted Site Zone

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

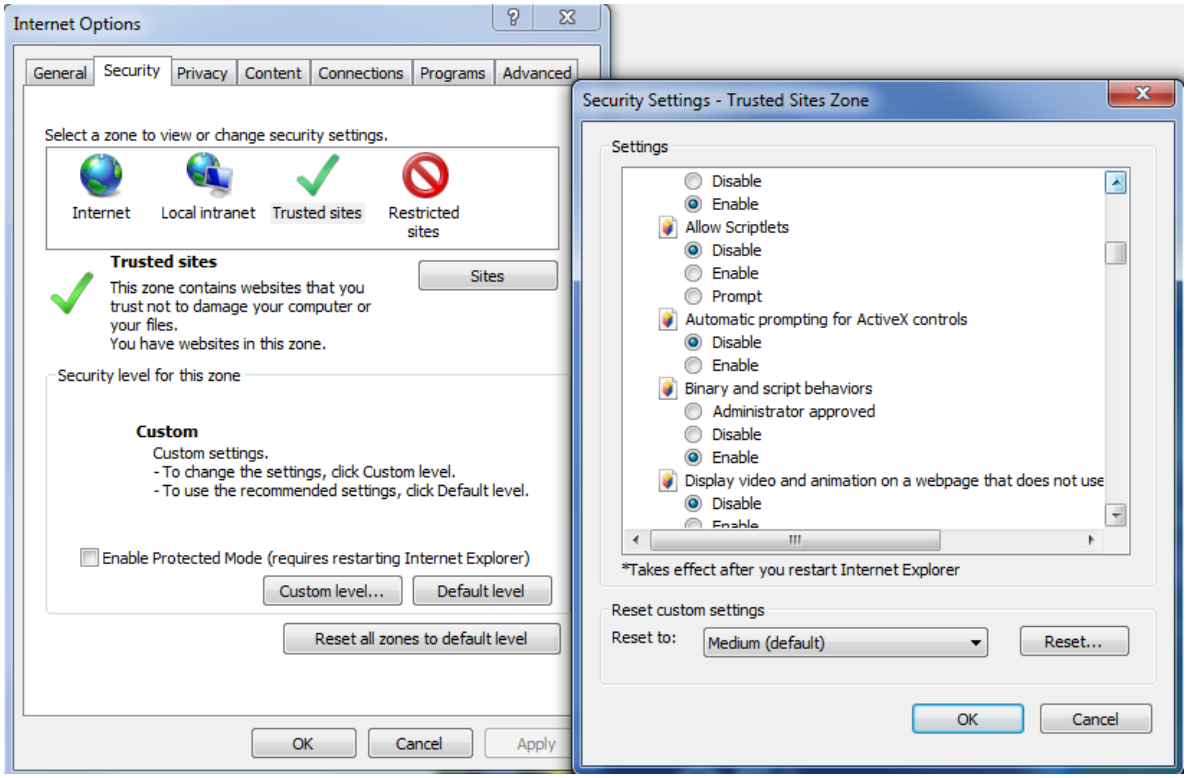

Figure 50

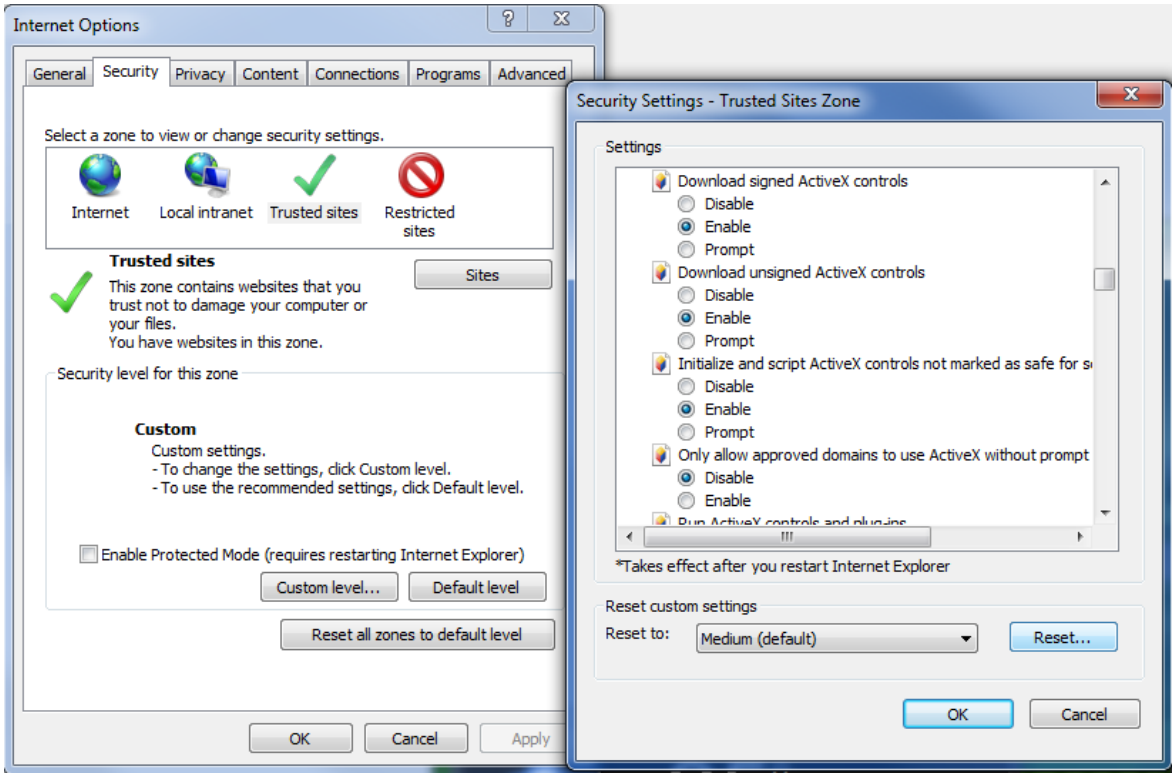

Figure 51

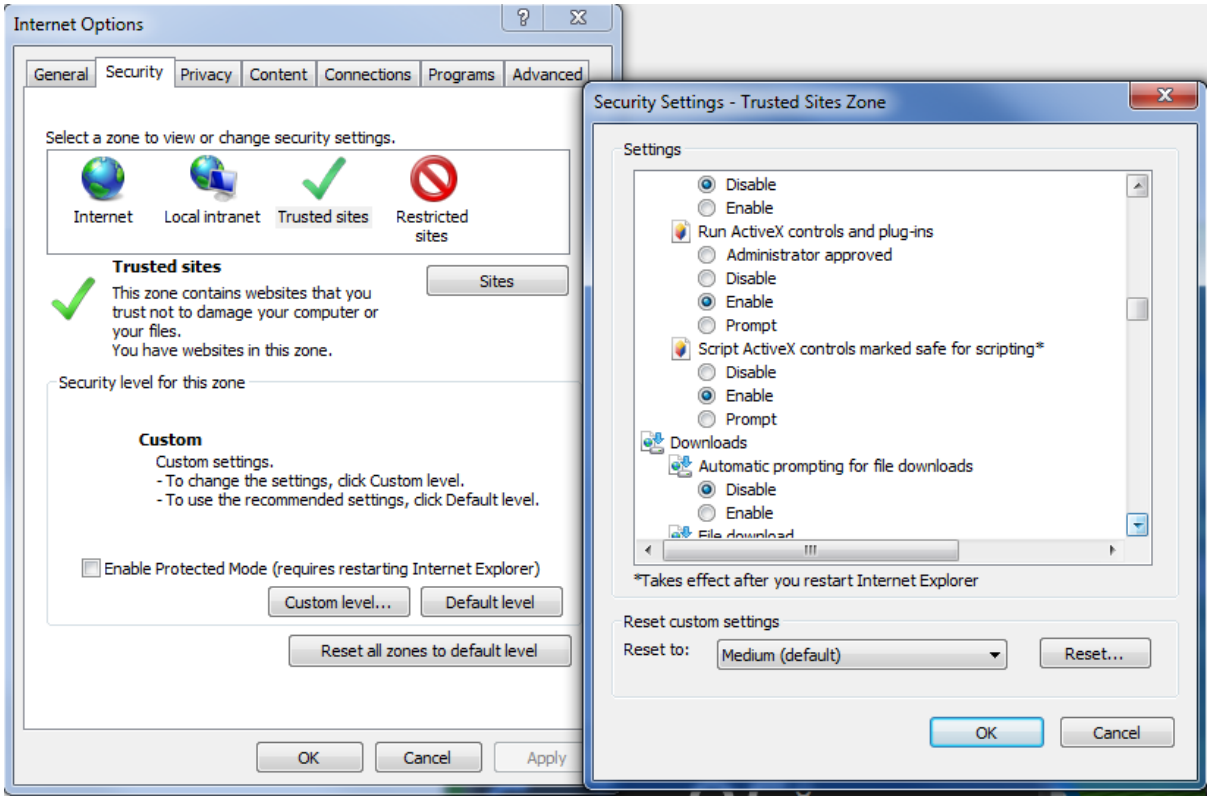

Figure 52

4.12.4 Install Active X on Internet Explorer

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

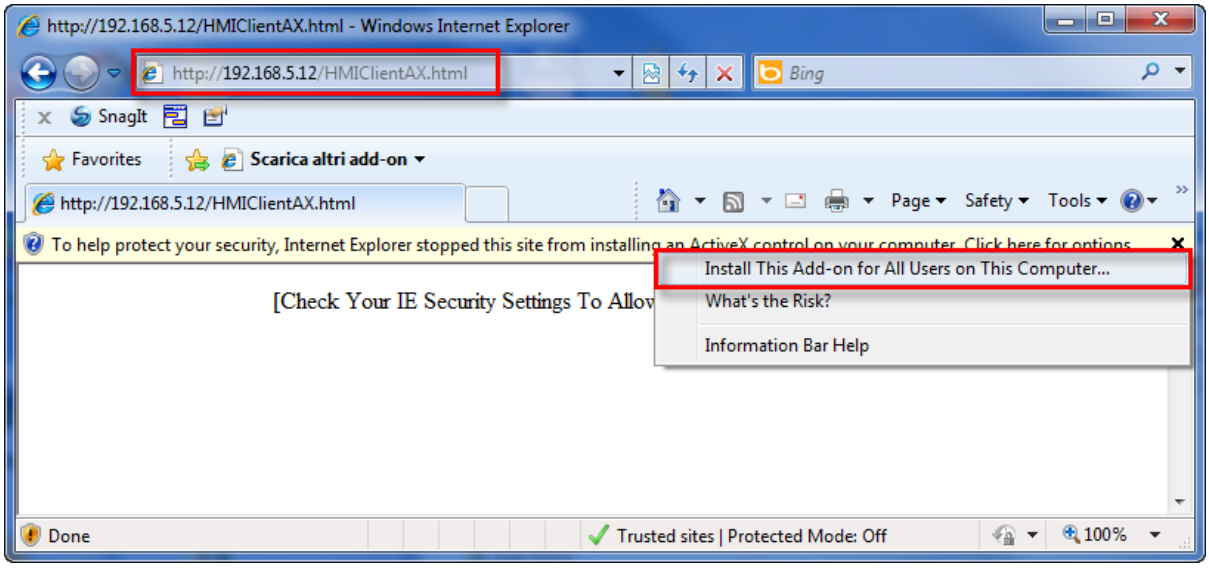

Figure 53

In case 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.

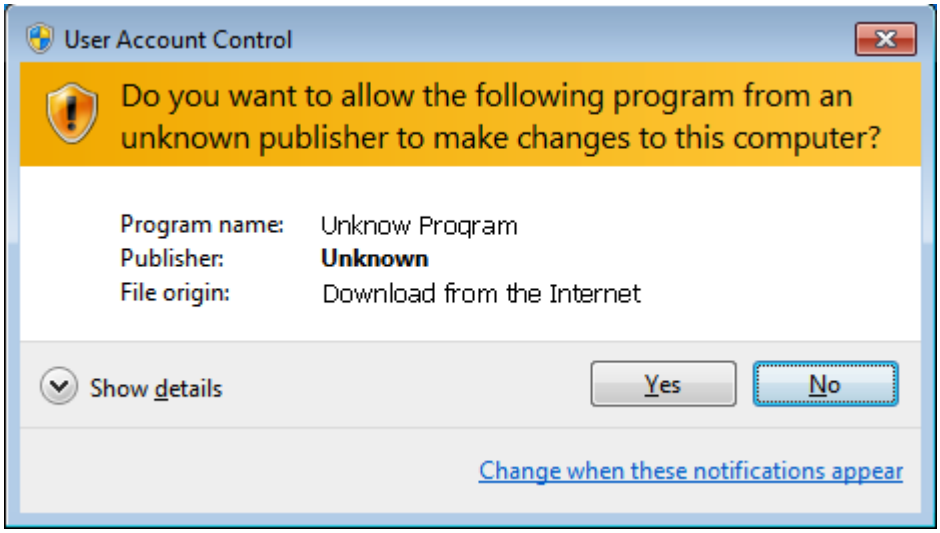

Figure 54

4.12.5 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\MDT\VisuControl\Runtime\HMIClientAX

4.12.6 ActiveX information

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

5 Basic Programming Concepts in VisuControl

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

5.1 Attach to

In VisuControl the basic programming techniques are used 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 55

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 you to both 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 ability to "Attach" the property to a second element. This operation is done using the "Attach to" dialog shown in below figure.

The Attach to dialog has two tabs. The first is called "Tag" and allows you 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 you to set the access type.
The TagIndex selection is used in case of arrays to determine the array element.

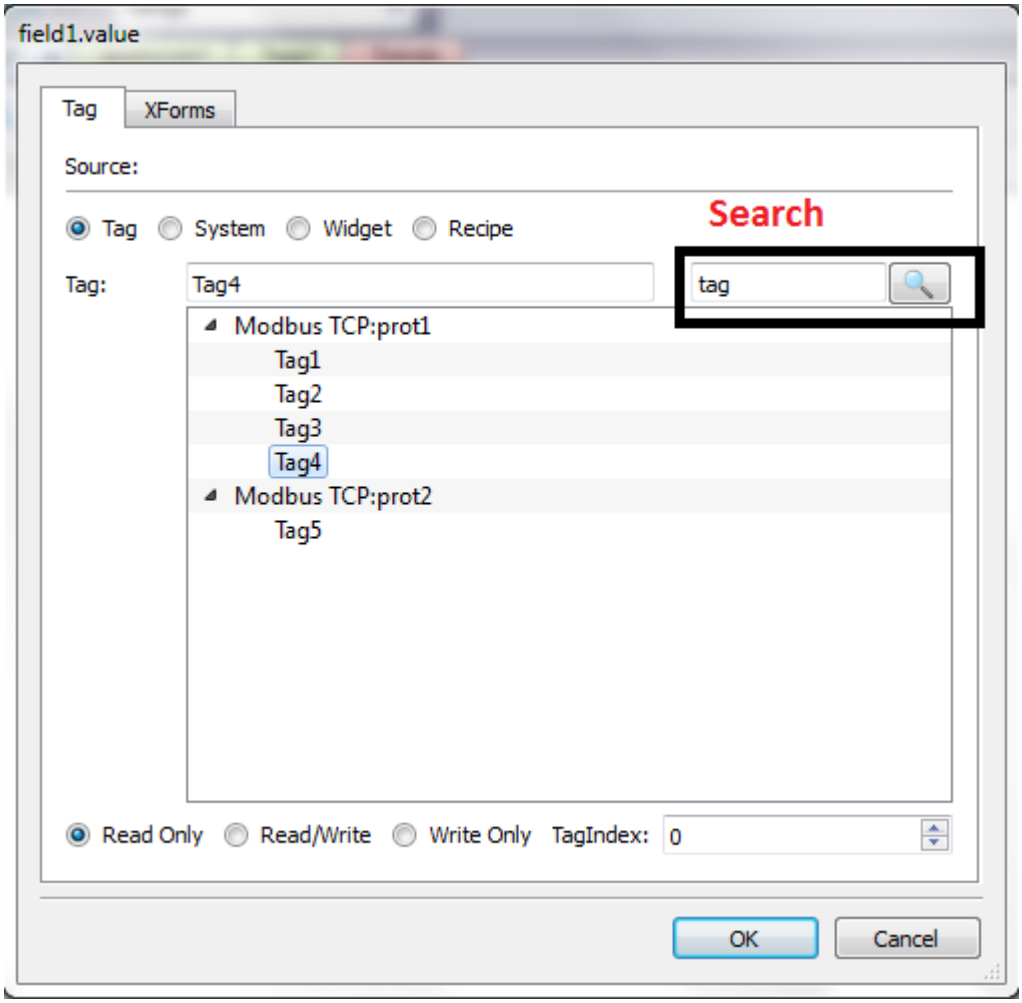

Figure 56

While adding Tags, the Protocols used in the Project are shown in the tag Dialog and when expanding each protocol the corresponding tags can be seen. The tags will be arranged in alphabetical order inside each Protocol.
There is an Option to search the tag to be attached by its name as shown in the Figure above. This makes it easy to find Tags.
The search can be done in two ways: first you can start typing the tag name in the left box: this will “jump” into the list to the first tag starting with the entered characters; second, you can type in the search box any part of a tag name; this will automatically apply a filter to the view so that only the tags which contain the entered characters are displayed.

Note: tag sorting, tag grouping and tag filtering are features available starting from version 1.80

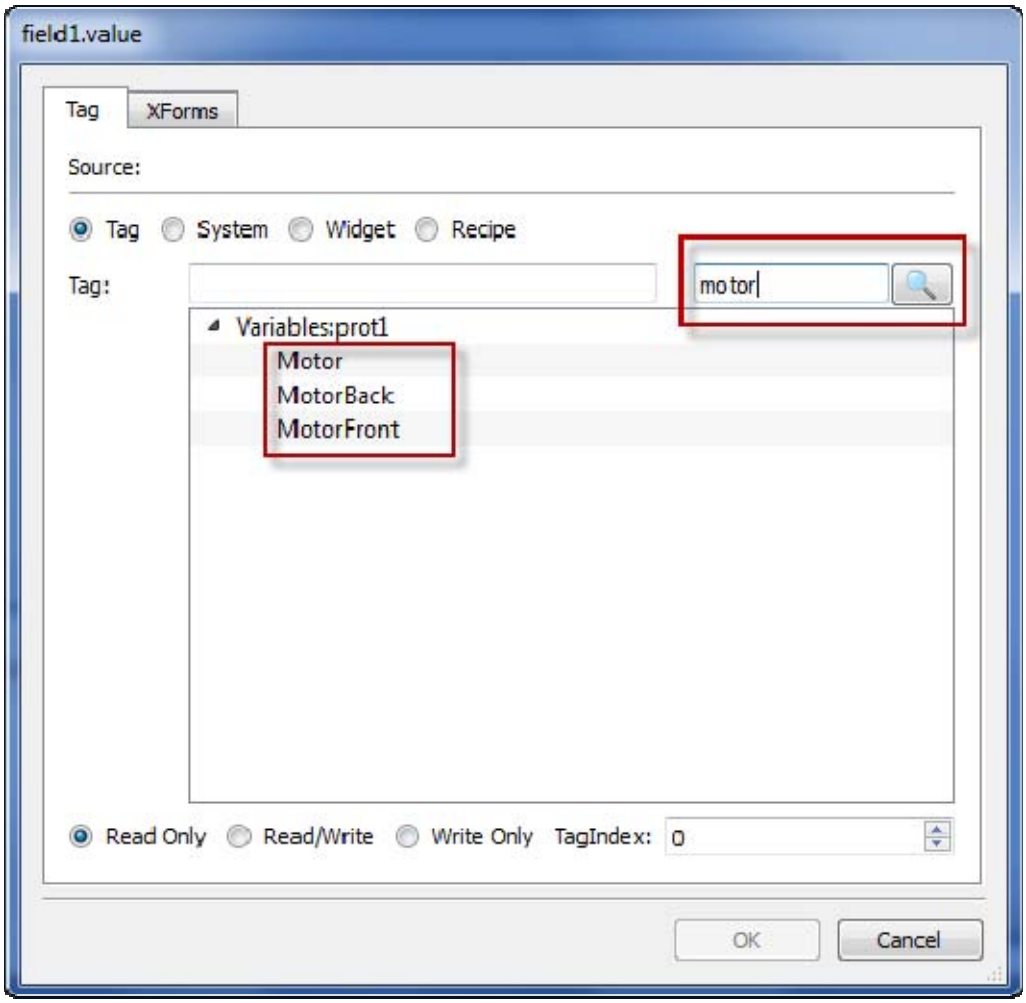

Figure 57

The second tab is called "XForms" and it is shown in below figure. The XForms tab allows to apply transformations to the numeric value of the source element before it 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.

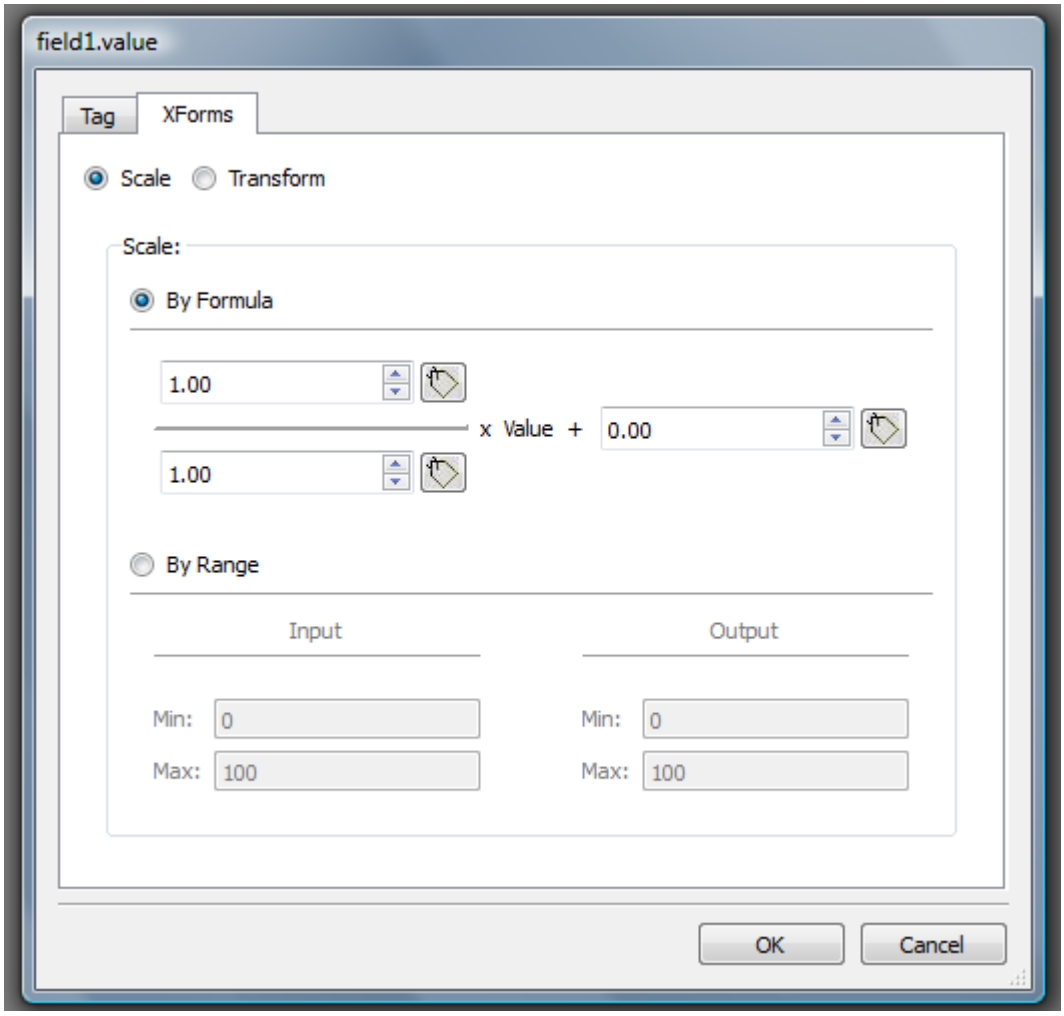

Figure 58

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

Bit and Byte index Allows extracting a single bit or byte content from a word depending on the specified bit or byte number.

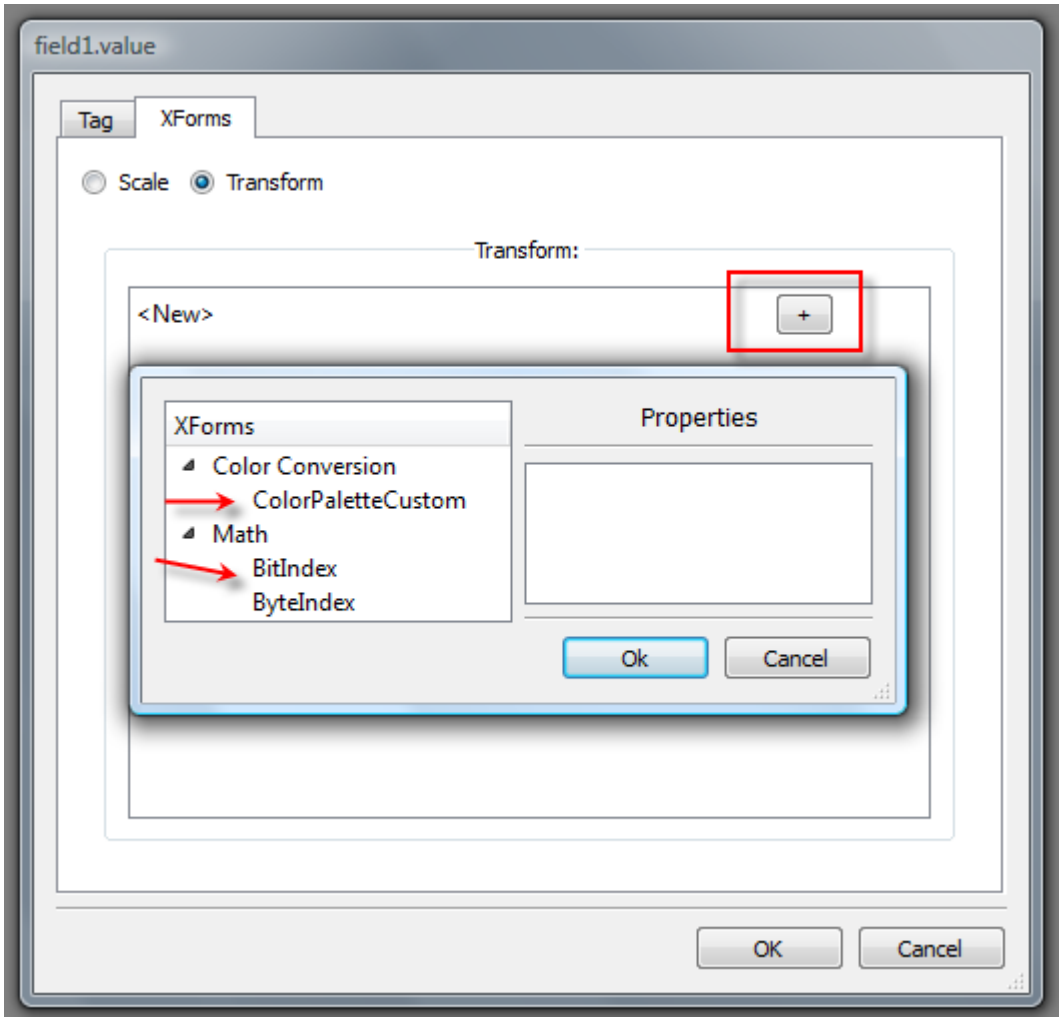
field1.value
Tag
XForms
Scale
Transform
Transform:
<New>
+
XForms
Color Conversion
ColorPaletteCustom
Math
BitIndex
ByteIndex
Properties
Ok
Cancel
OK
Cancel

Figure 59

5.2 Events

In a VisuControl 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 exited.
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 one of the following executed actions to the event:
A specific action coming from a list of predefined actions
A piece of JavaScript code
Below figure shows an example of an action activated by pressing a button.

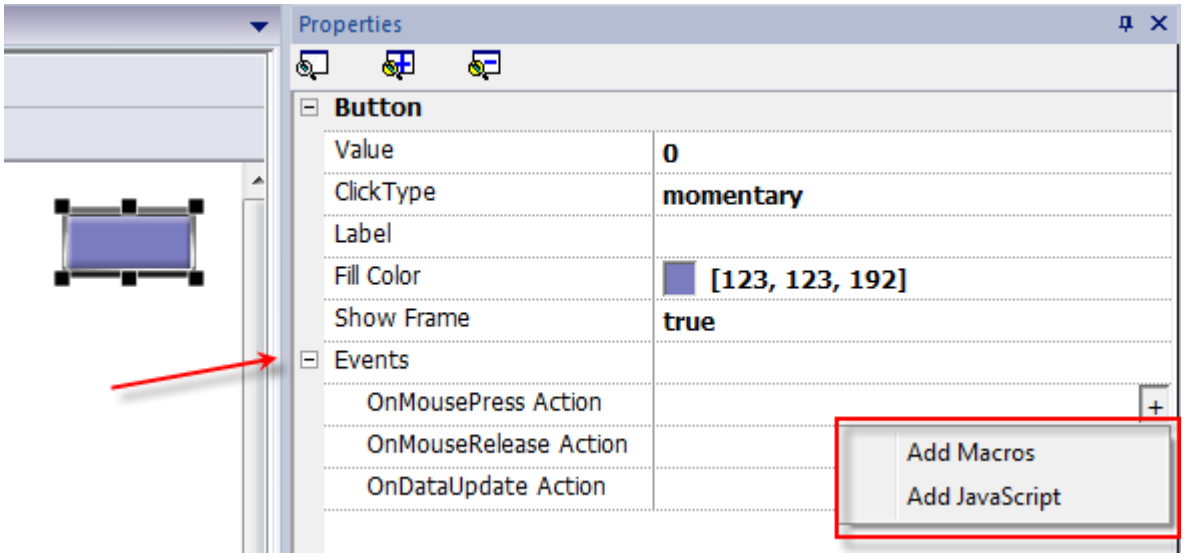

Figure 60

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

6 System Variables

System variables are special system tags containing generic information 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.

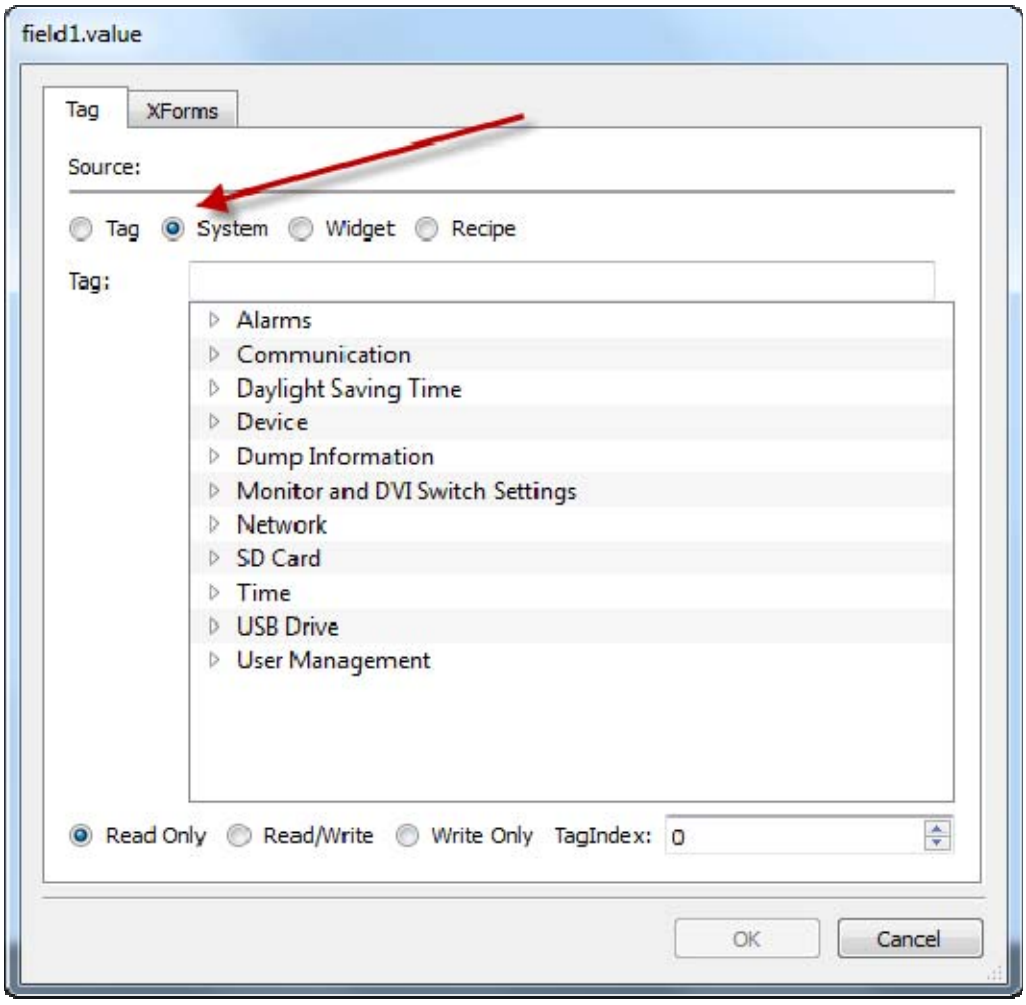

Figure 61

System variables are divided into categories. The relevant ones are listed below.

Note: Specific system variables may be available and useful only in special and dedicated applications including specific use of hardware; not documented variables have to be considered as “dedicated”; please contact Technical Support for further information.

6.1 Alarms

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

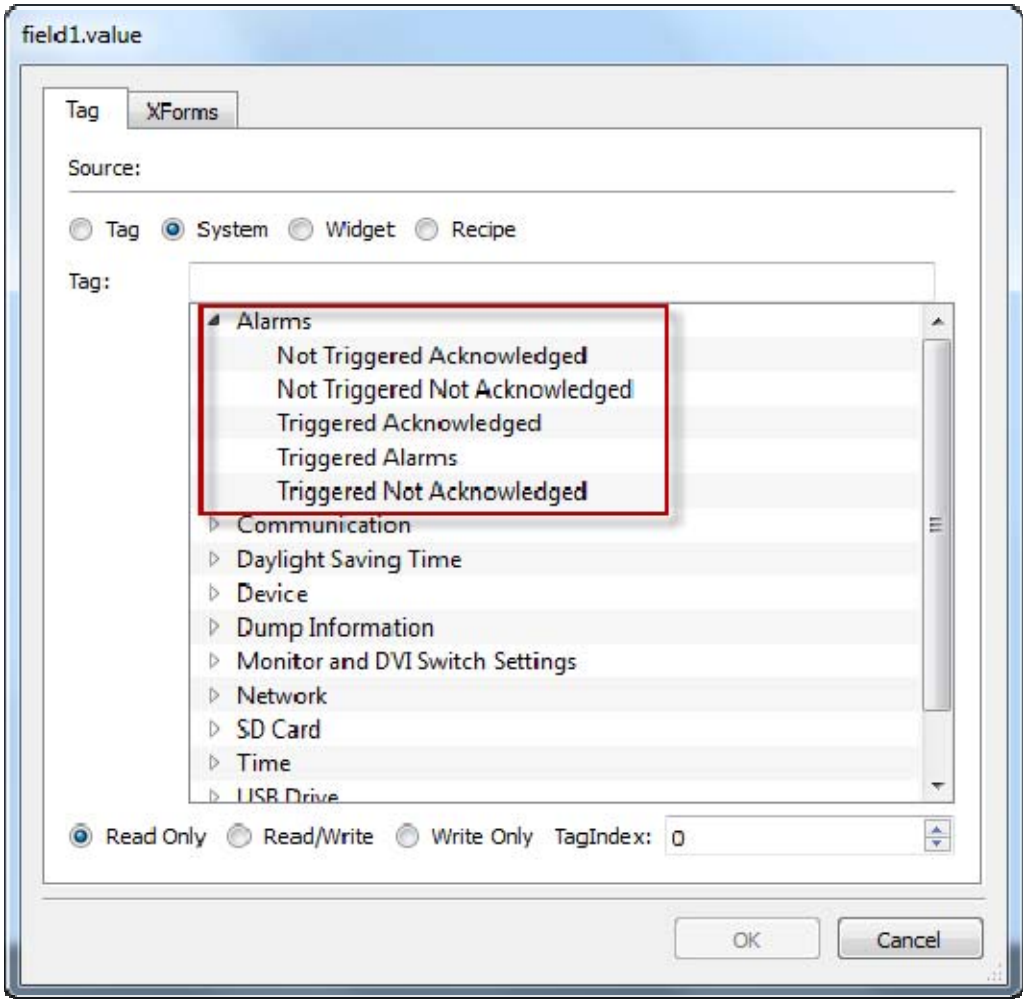

Figure 62

All these system variables are Int type (32 bit), read only.

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.

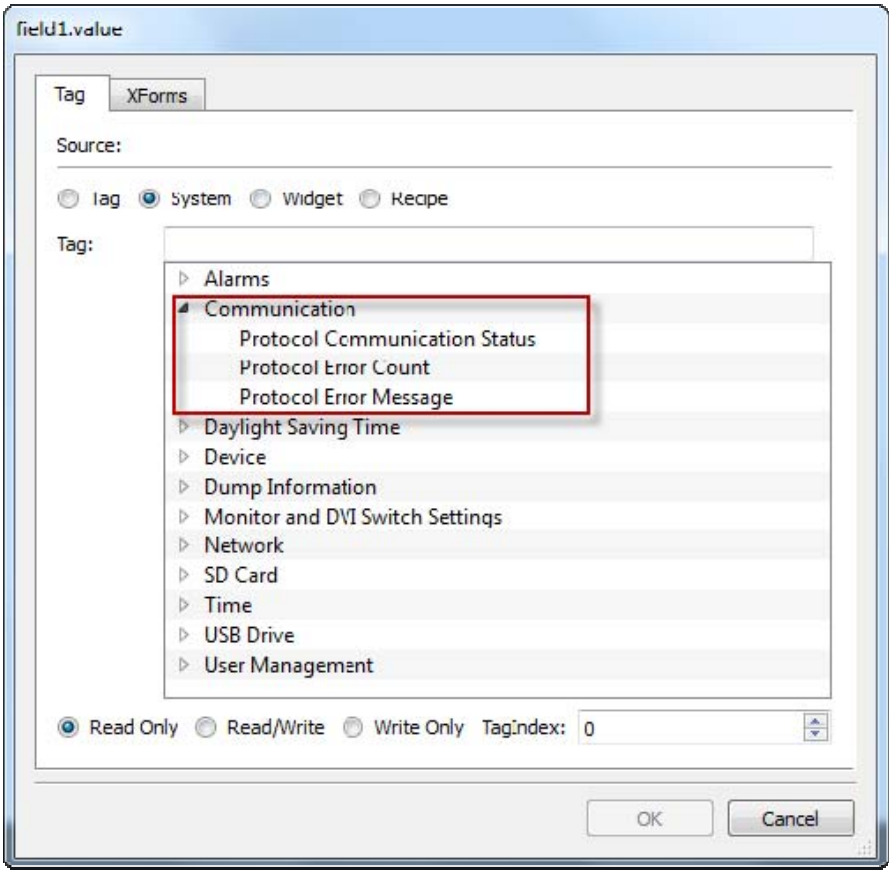

Figure 63

Protocol Communication Status The variable is read only int (32 bit), 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 brackets to recognize the source of the error in case of multiple protocol configuration.The variable is a read only string.

Protocol Error Count This variable returns the number of communication errors that occurred since the last time it was reset. The variable is a read only integer. The reset of this is only possible using the dedicated Action "Reset Protocol Error Count"

6.3 Daylight Savings Time

The variables in this category return information about the system clock and allow adjusting it from the application. They contain information on the "local" time. All the variables are read/write; this means that a change on them directly works changing the system RTC. All the variables are bytes (8 bit) except for the DLS and Standard Offset that are shorts (16 bit)

Standard time is the "solar time" and other is Daylight Savings 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)

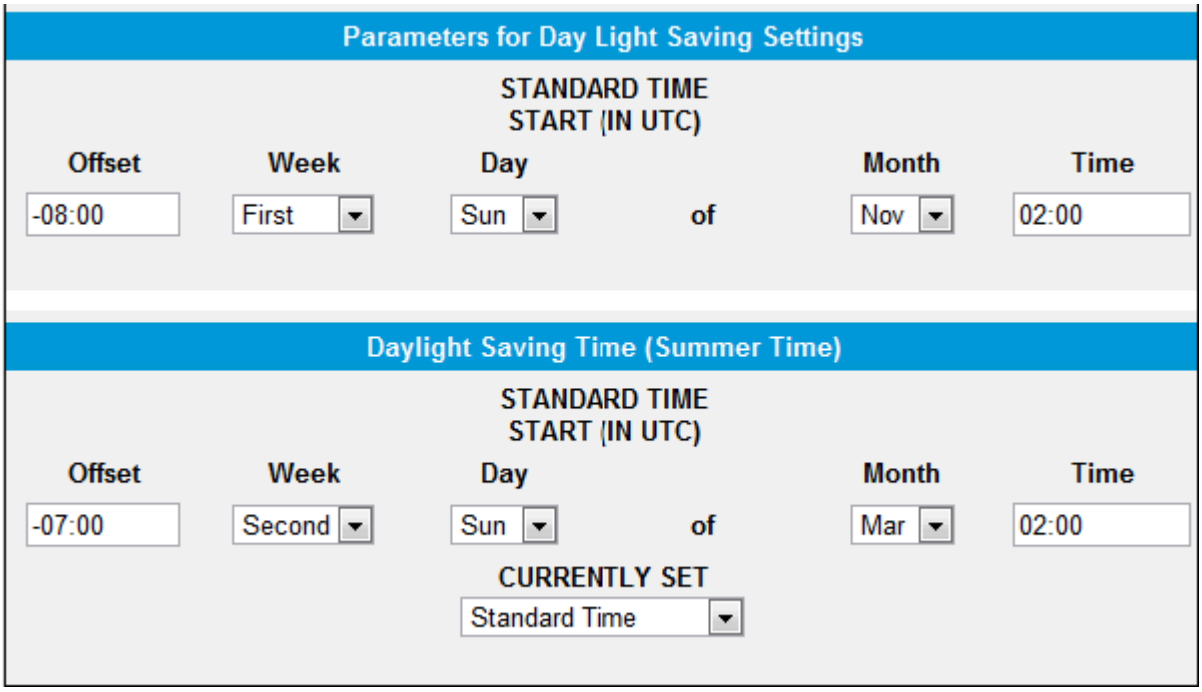
Parameters for Day Light Saving Settings
STANDARD TIME
START (IN UTC)
Offset
Week
Day
Month
Time
-08:00
First
Sun
of
Nov
02:00
Daylight Saving Time (Summer Time)
STANDARD TIME
START (IN UTC)
Offset
Week
Day
Month
Time
-07:00
Second
Sun
of
Mar
02:00
CURRENTLY SET
Standard Time

Figure 64

6.4 Device

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

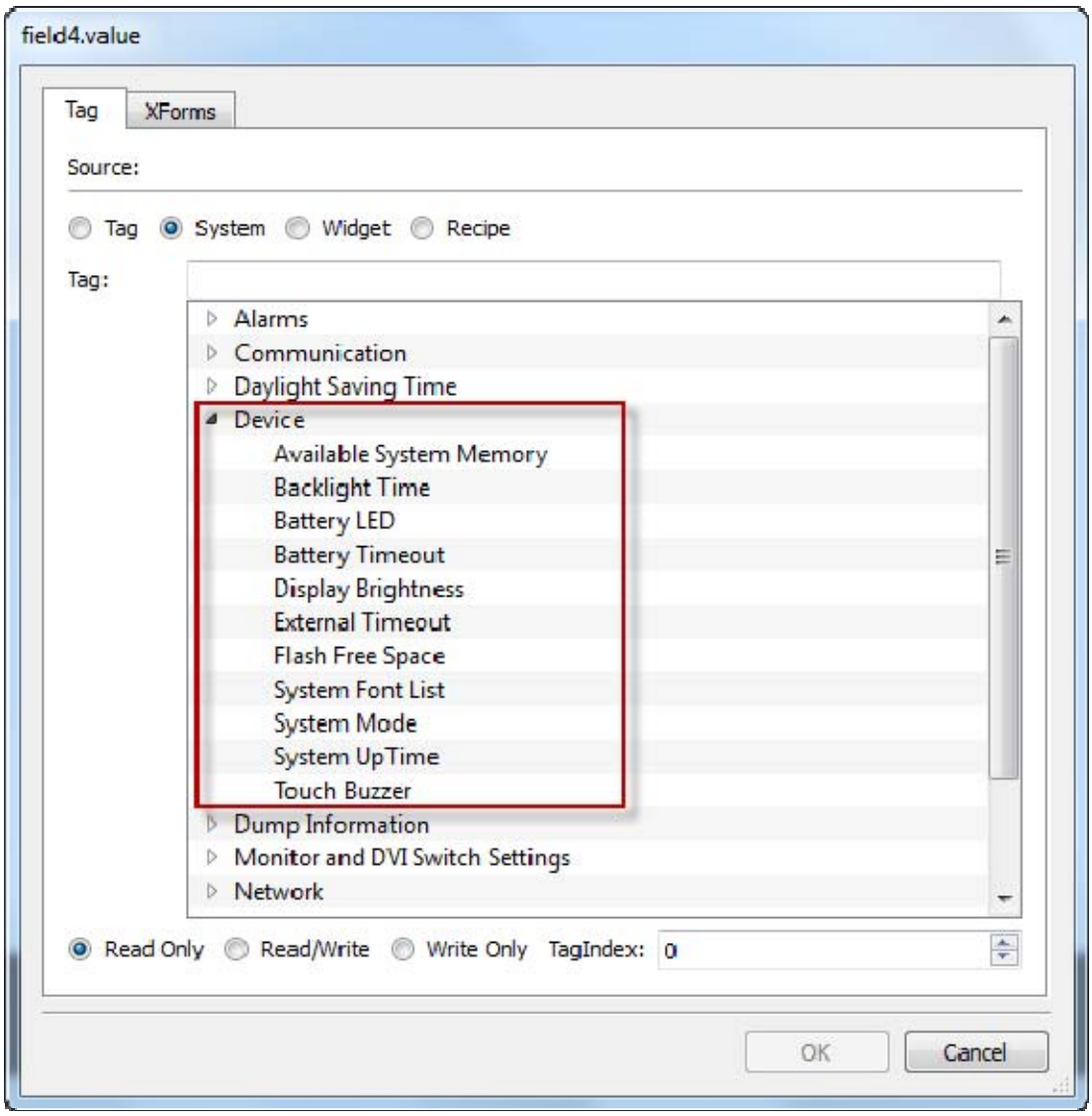

Figure 65

Available System Memory: Returns the free available RAM memory in bytes; it is a 64 bit data; it is a read only variable.

Backlight Time: Returns the activation time in hours of the display backlight lamp since production of the unit; it is a read only variable.

Battery LED Allows you to enable/disable the use of the front LED indicator to report the low level battery status. It can assume values 0 (disabled) or 1 (enabled).

Battery Timeout Reserved.

Display Brightness	This variable is an integer 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.
External Timeout	Allows setting the non operational time after which the display backlight is automatically turned off. The backlight is then automatically turned on when user presses on the touch screen. The variable is an int of R/W type.
Flash Free Space	Returns the free space left in the device internal flash.
System Font List	List of system fonts. The variable is a read only string.
System Mode	Returns a value informing the operation status of the runtime - the possible values are: 1: Operating mode 2: Configuration mode 3: Restart
System UpTime:	Returns the total time in hours in which the system has been powered since production of the unit. It is a read only variable.
Touch Buzzer	Allows you to enable/disable the touch audible feedback. It can assume values 0 (disabled) or 1 (enabled).

6.5 Dump information

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

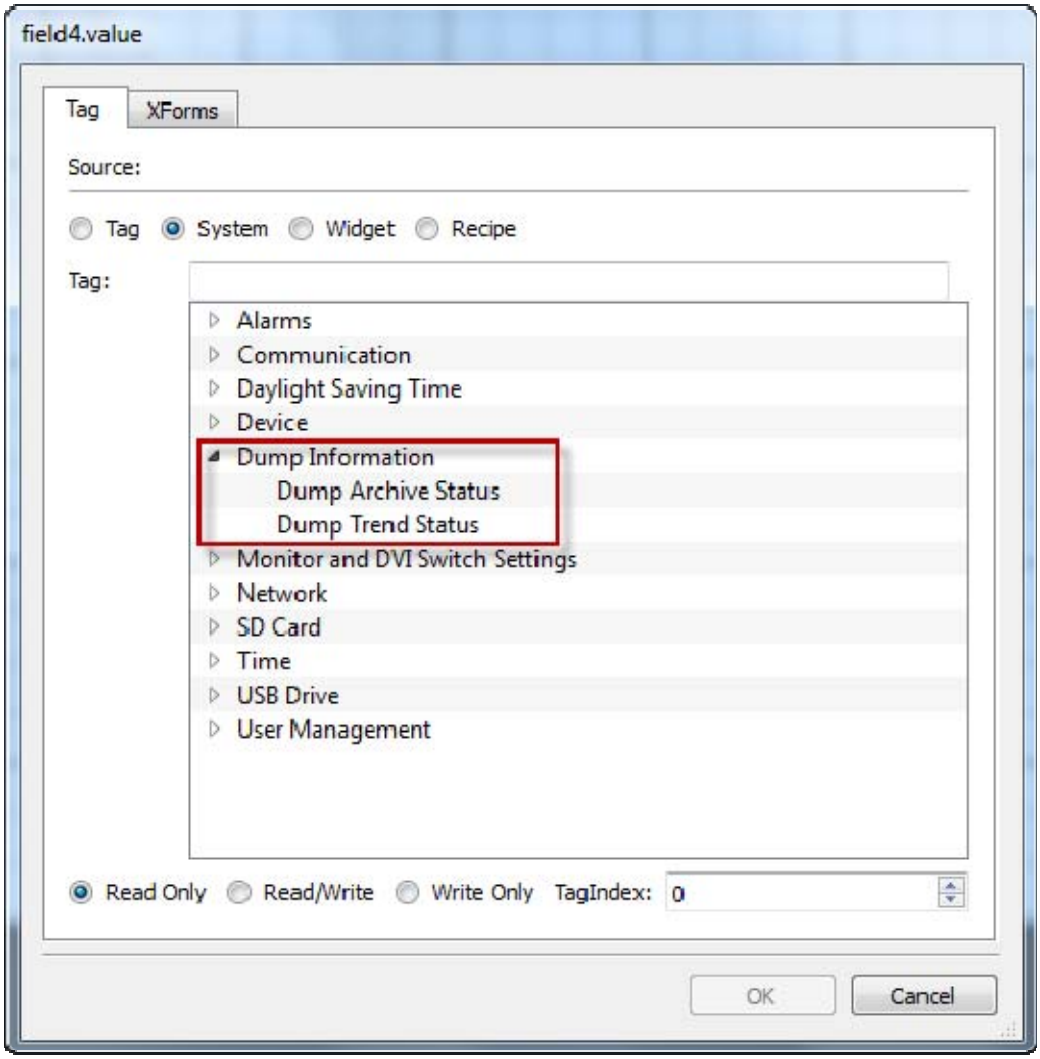

Figure 66

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.

6.6 Network

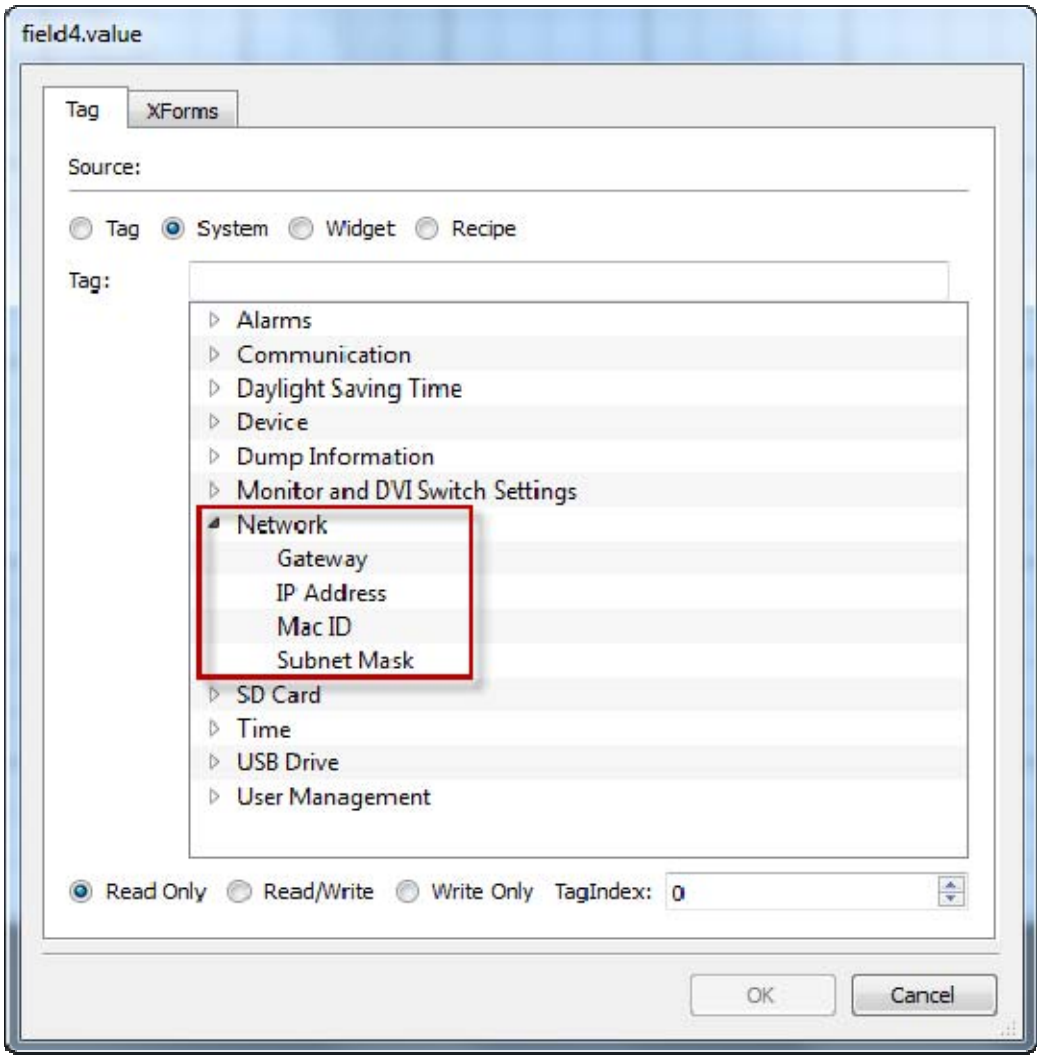

Figure 67

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

6.7 Time

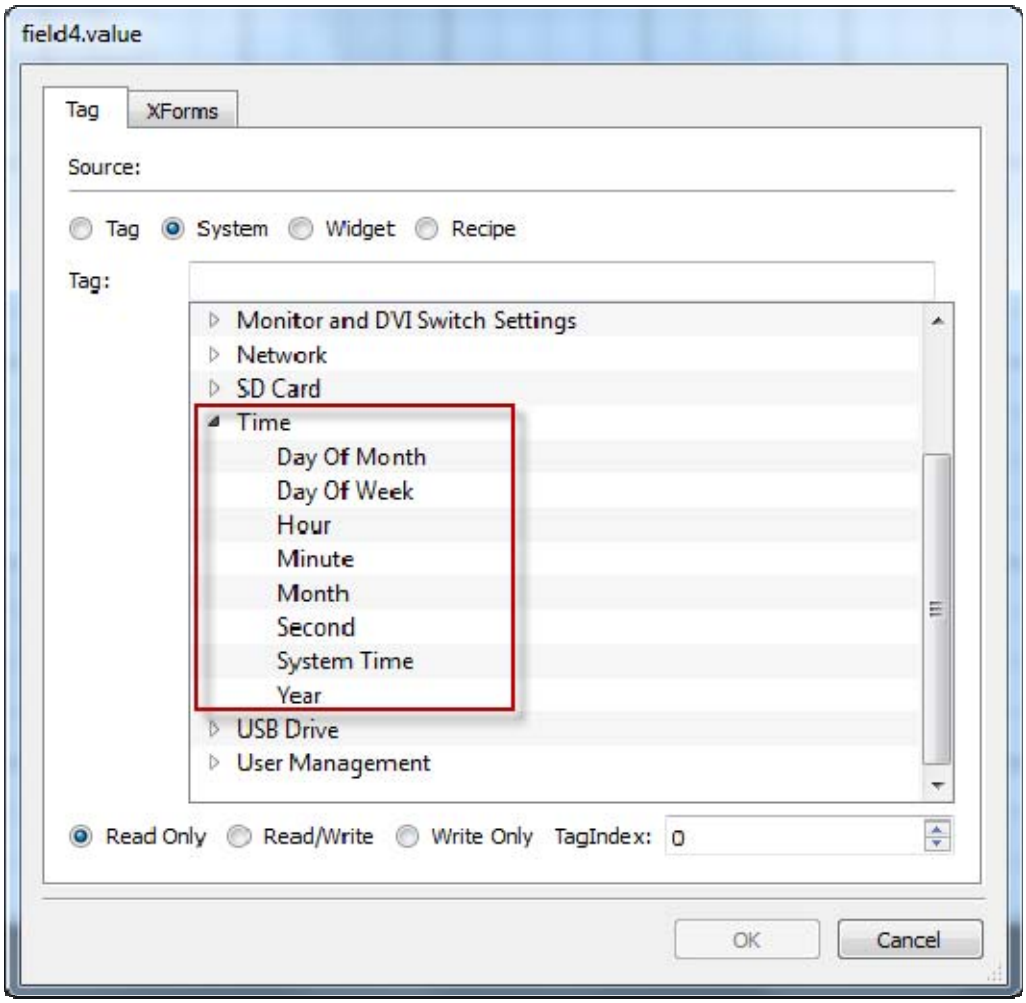

Figure 68

The variables in this category return information about the System Time expressed in UTC format. They are all int (32 bits) of read/write type, except for the System time which is a 64 bit variable, still of read/write type. This is actually the UTC time which also is available as date/time from the other variables.

6.8 User Management

These system variables return information about users and groups.

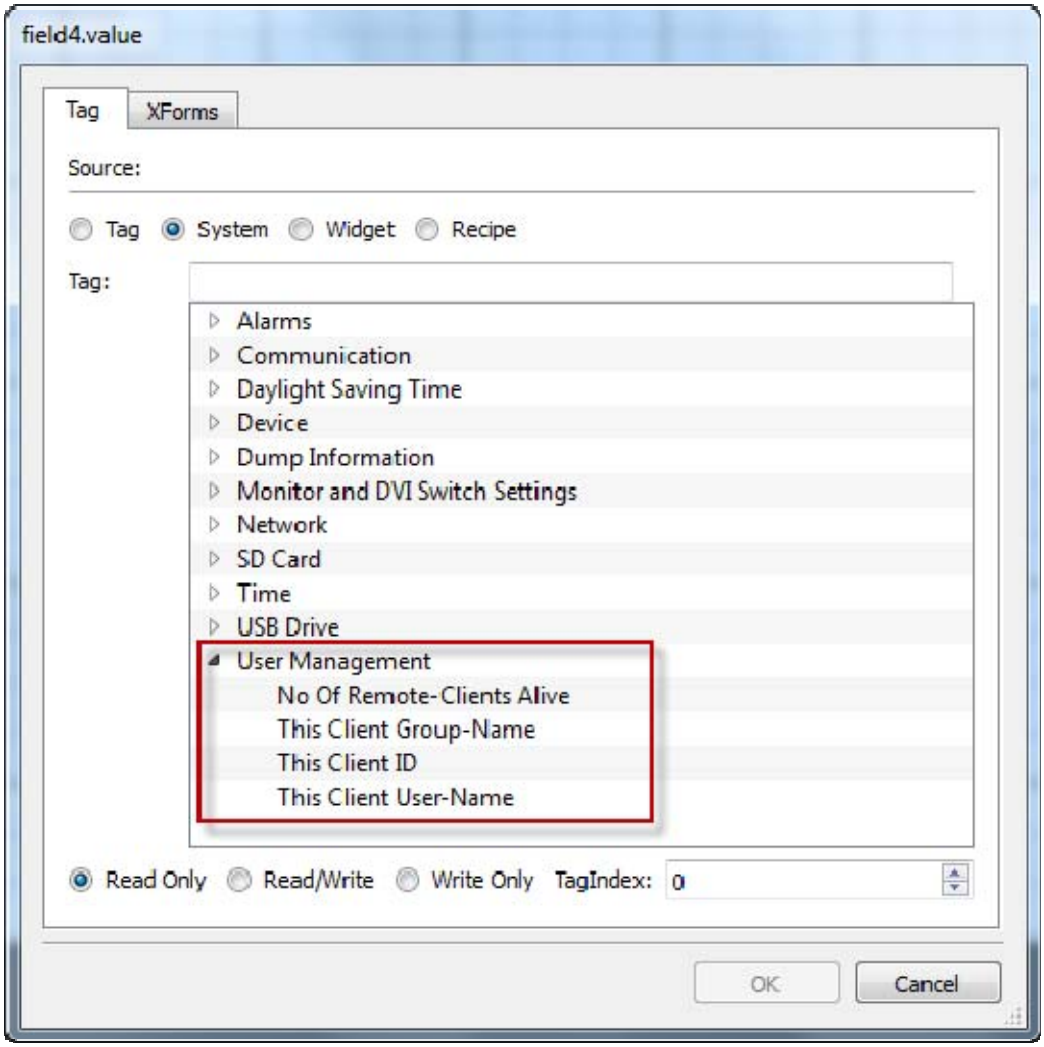

Figure 69

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 is a read only short (16 bit)

This Client User-Name — Name of the user logged to the client where the system variable is displayed. This is a read only string.

This Client Group-Name — Name of the group to which the current logged user belongs to. This is a read only string.

No Of Remote-Clients Alive — Number of remote clients connected to the server. This is a read only short (16 bit)

6.9 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”, “USB Drive Status” and USB Drive free space.

They are 64 bit variables, except the drive name which is a string. All the variables are read only.

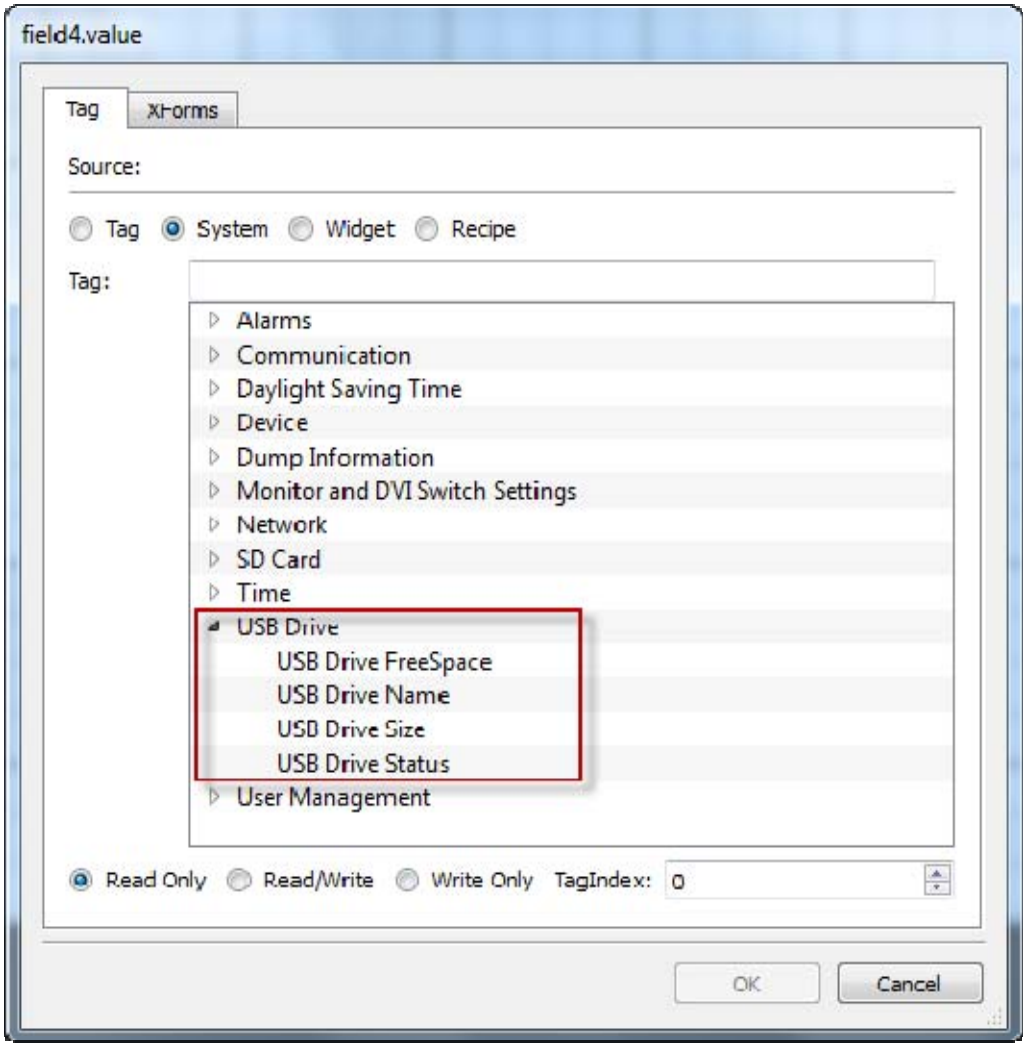

Figure 70

6.10 SD Card

The variables in this category return information about the external SD Card plugged into the panel. The available variables are: “SD Card Name”, “SD Card Size”, “SD Card Status” and “SD Card free space”.

They are 64 bit variables, except the drive name which is a string. All the variables are read only.

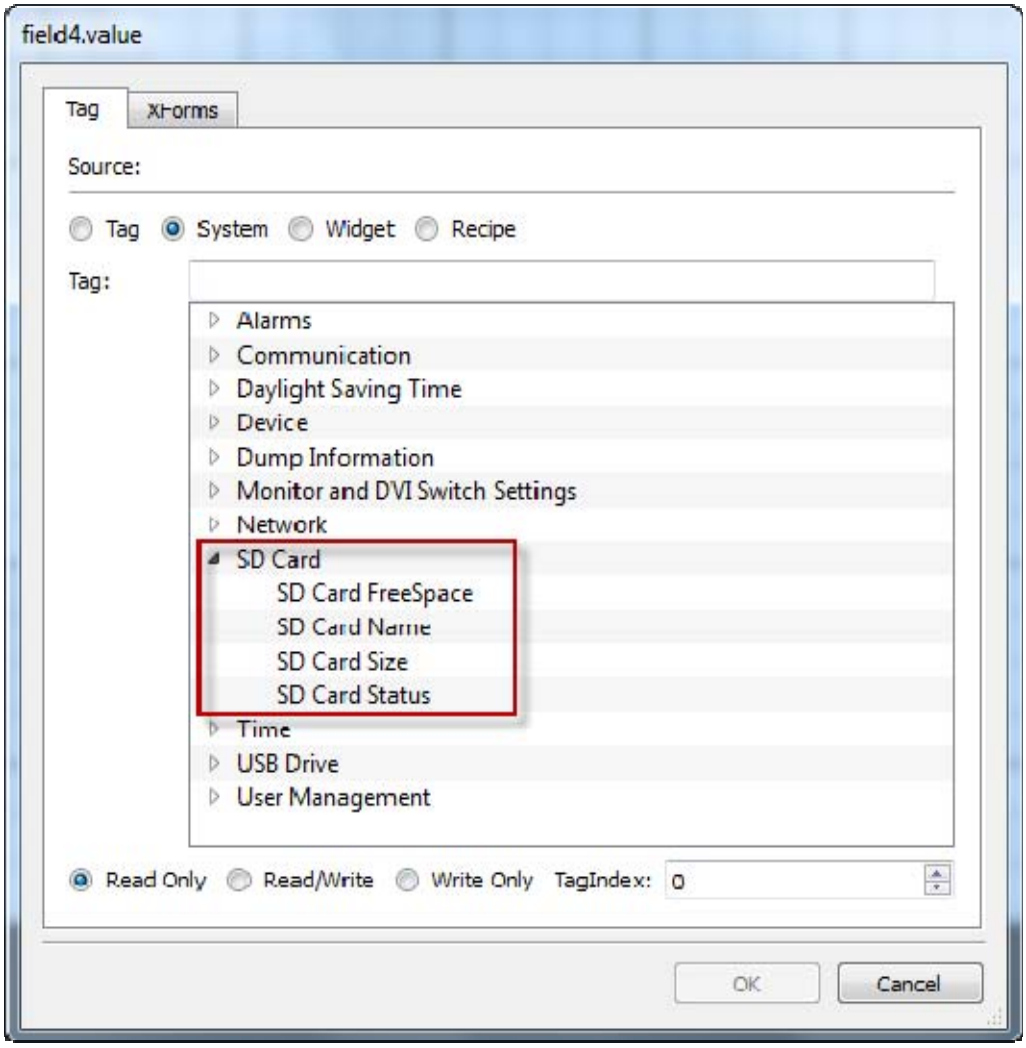

Figure 71

7 Working with Actions

Actions are used to interact with the system; they can be executed when events are triggered. When considering events generated by buttons (pressed or released) not all the actions are available for both states.

In case the selected action is not supported for the actual state, VisuControl will report a warning message as shown in the following figure.

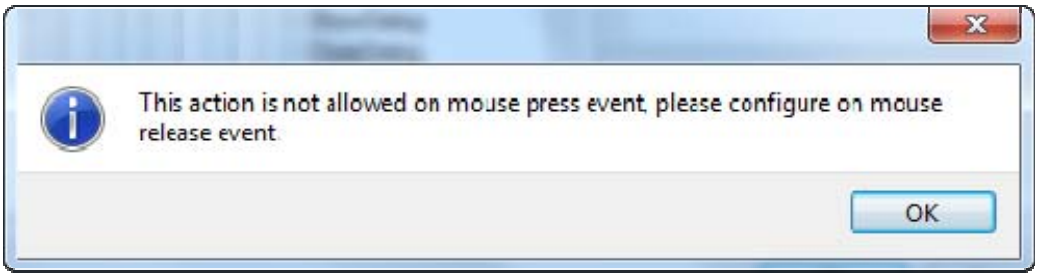

Figure 72

7.1 Widget Actions

The following chapter will include the description of a set of actions dedicated to handling widget visibility and control.

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.

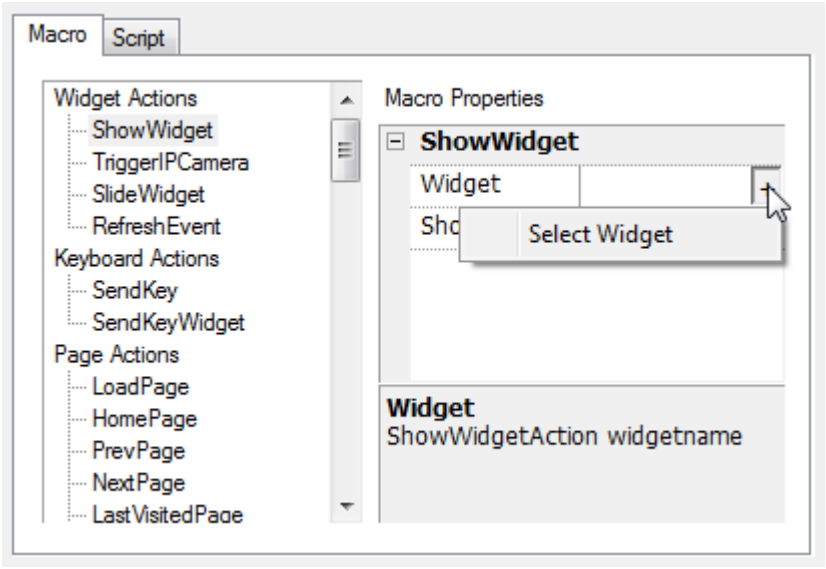

Figure 73

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.

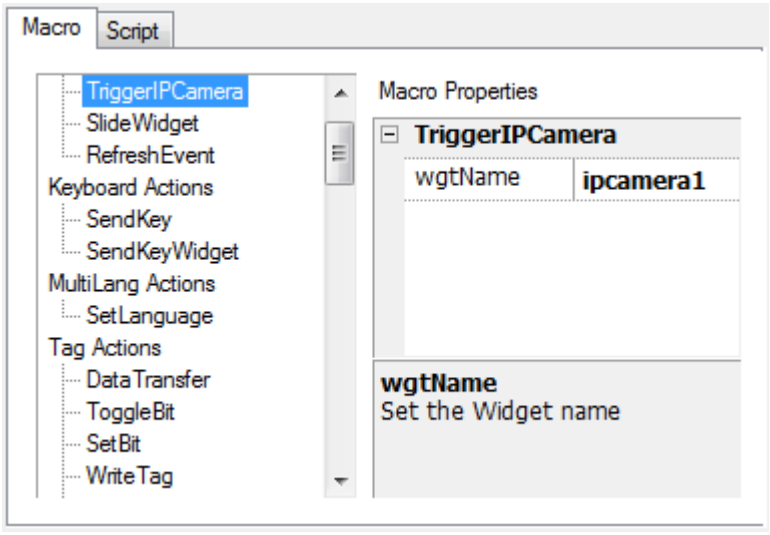

Figure 74

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.

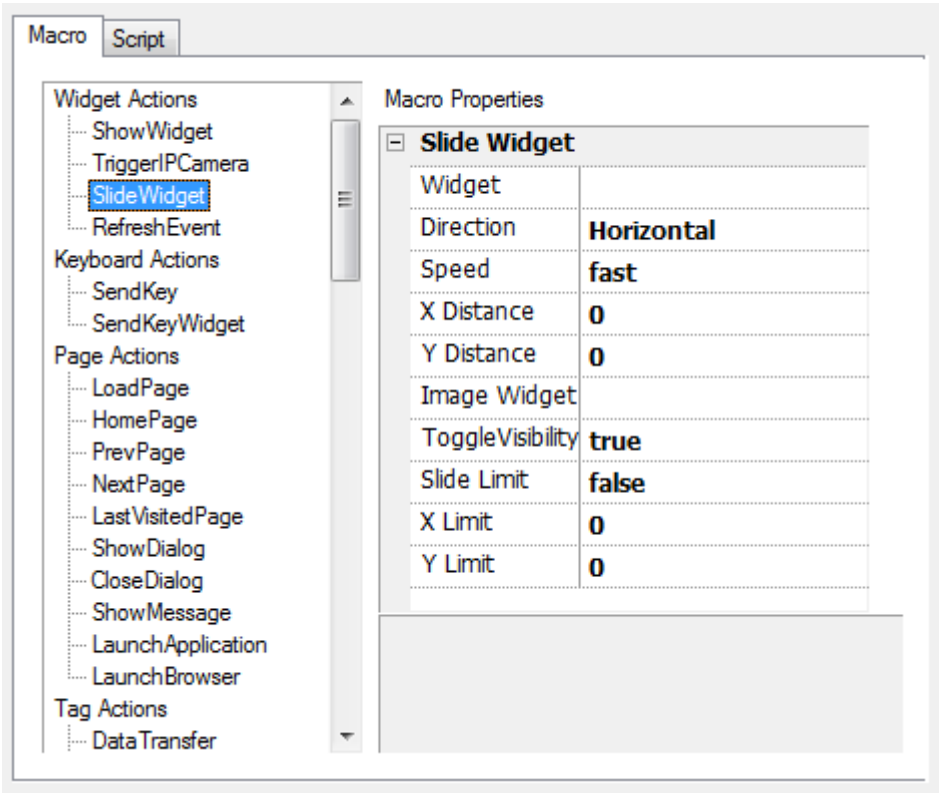

Figure 75

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
Slide Limit	Enable/Disable limiting the movement with Respect to the Coordinates (X and Y) of the Widget.
X Limit	When specified, allows to stop automatically the slide action when the widget reaches the specified position.
Y Limit	When specified allows stopping automatically the slide action when the widget reaches the specified position.
Toggle Visibility	Toggle the Visibility of the Widget at the end of each Slide action.
Image Widget	Allows to show an Image during the movement; the specified image will be shown during the Slide Operation between start and end point of the movement.

7.1.4 Refresh Event

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

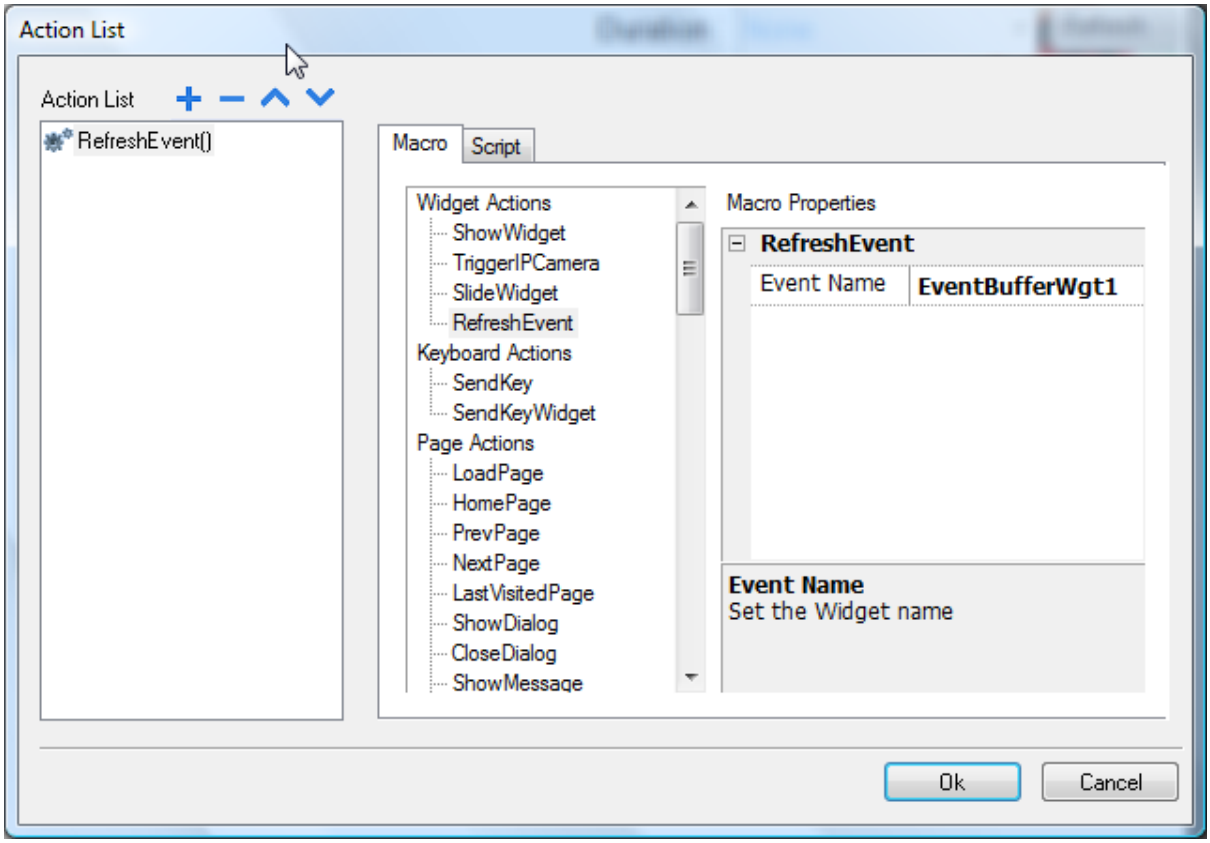

Figure 76

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 49).

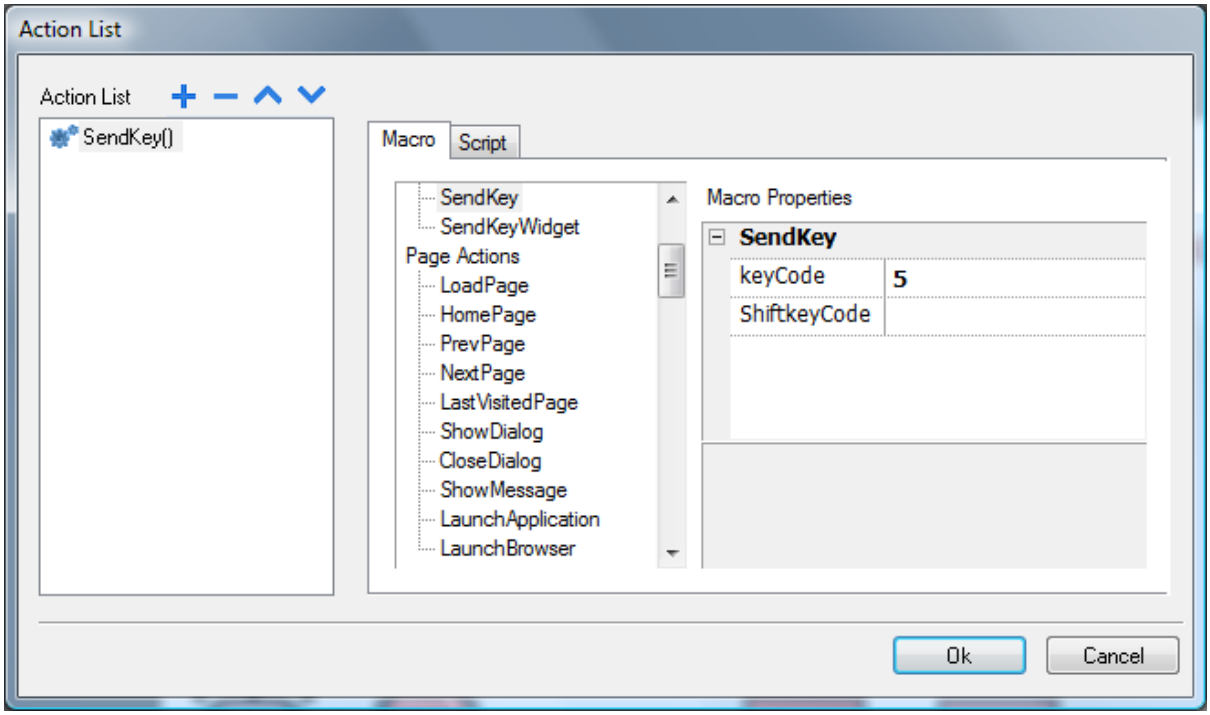

Figure 77

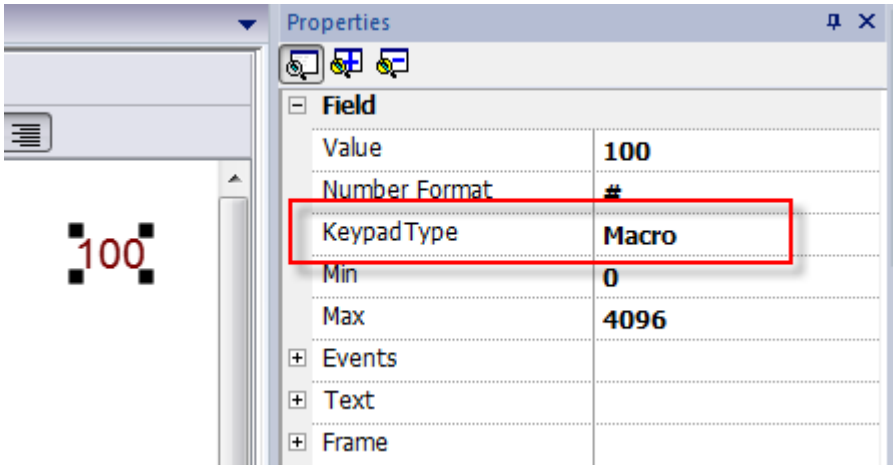

Figure 78

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 the figure below for reference.

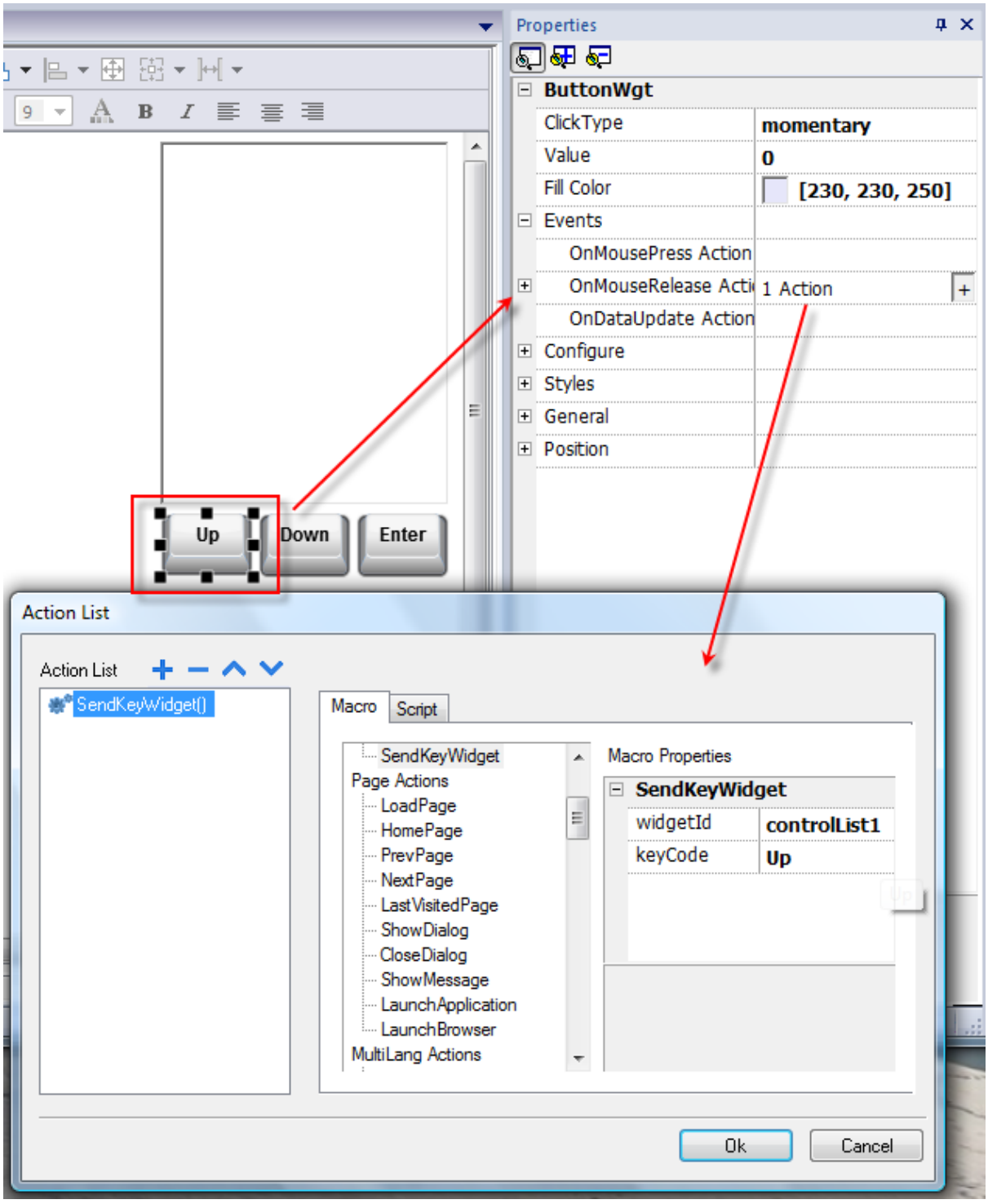

Figure 79

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

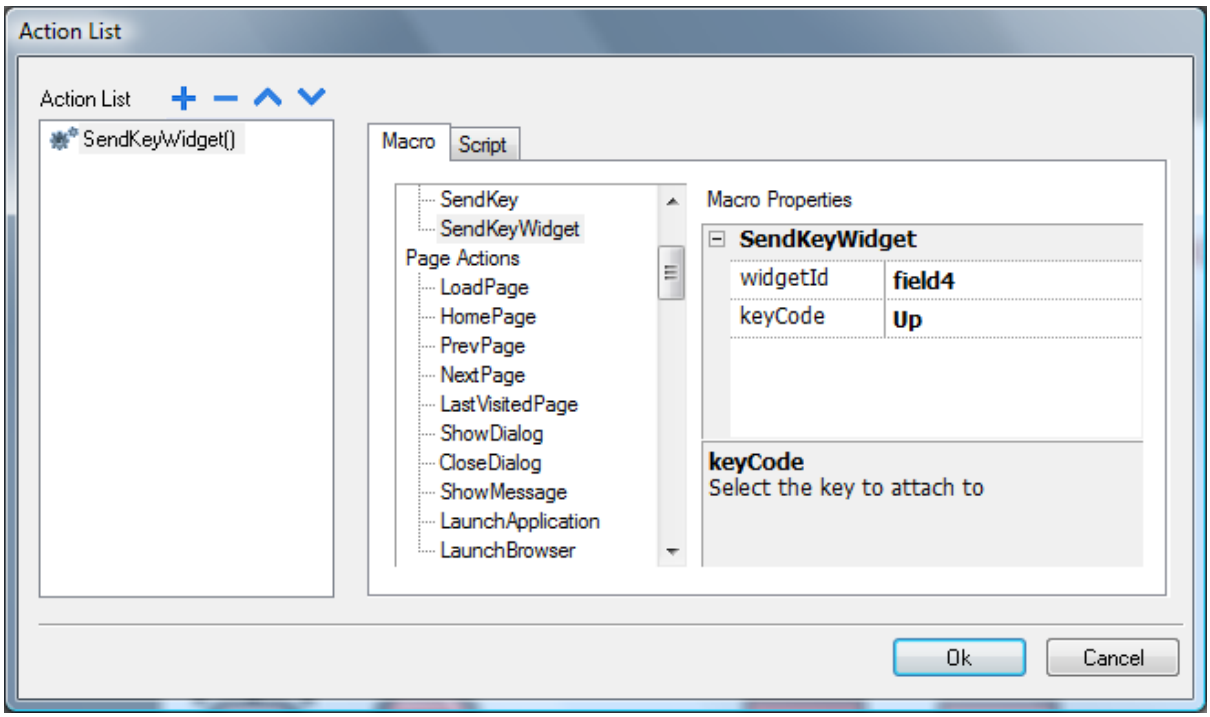

Figure 80

7.3 Page Actions

The Page Actions macro is used for Page Navigation and Load-specific pages. Please note that the

Note: Page Actions are programmable ONLY in the released state.

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.

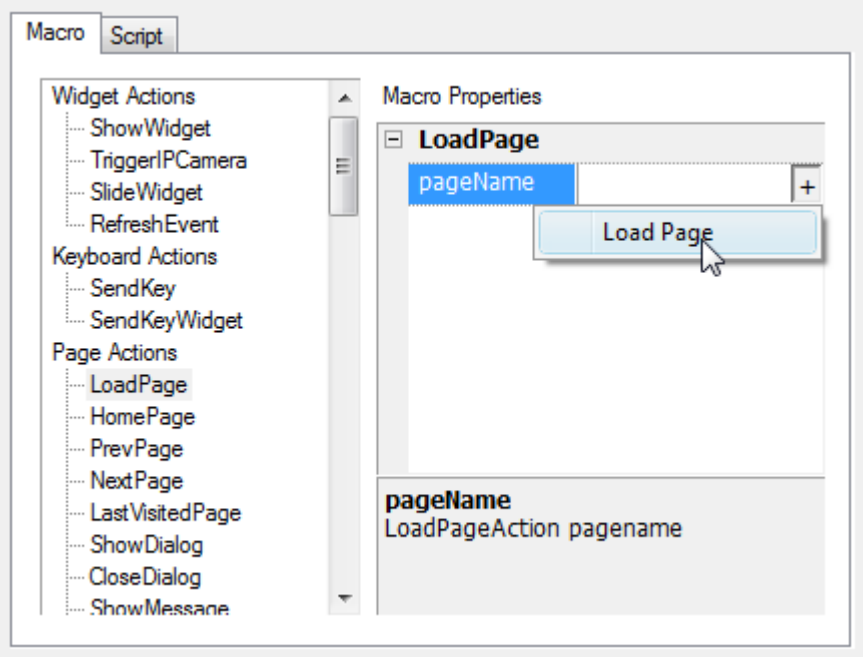

Figure 81

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 below figure).

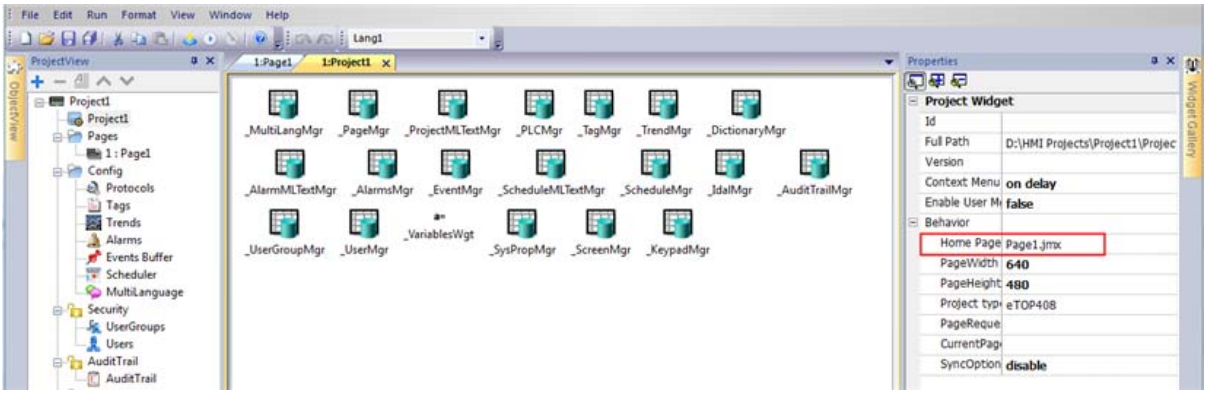

Figure 82

7.3.3 Previous Page

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

7.3.4 Next Page

The Next Page macro allows you to navigate the 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.

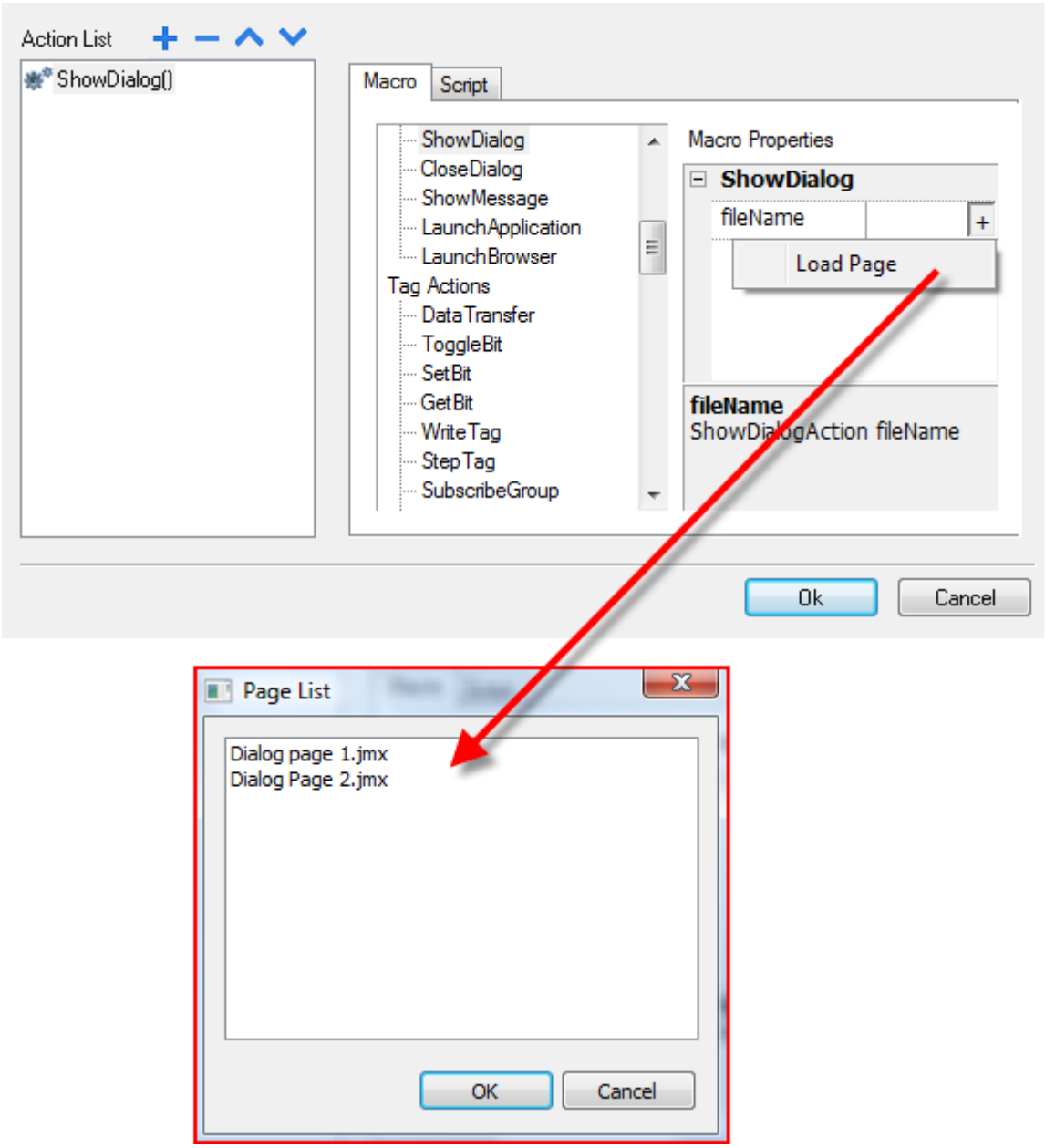

Figure 83

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 below figure).

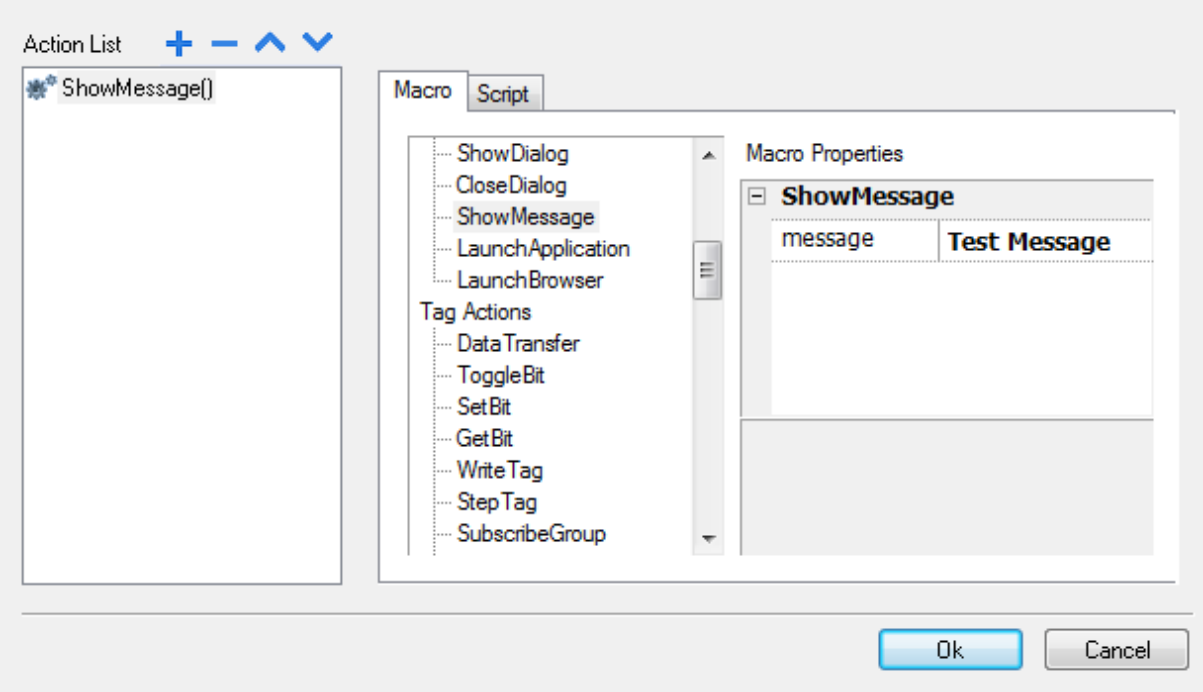

Figure 84

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 Instance

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

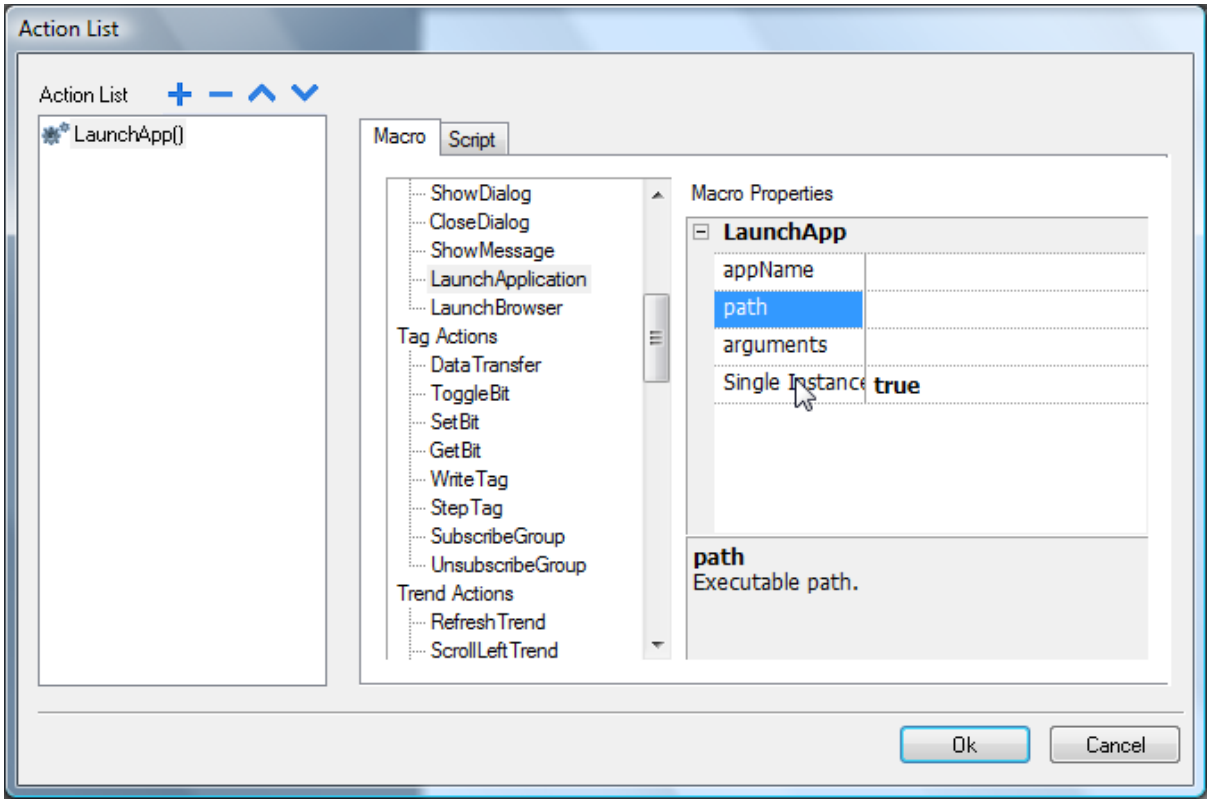

Figure 85

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.

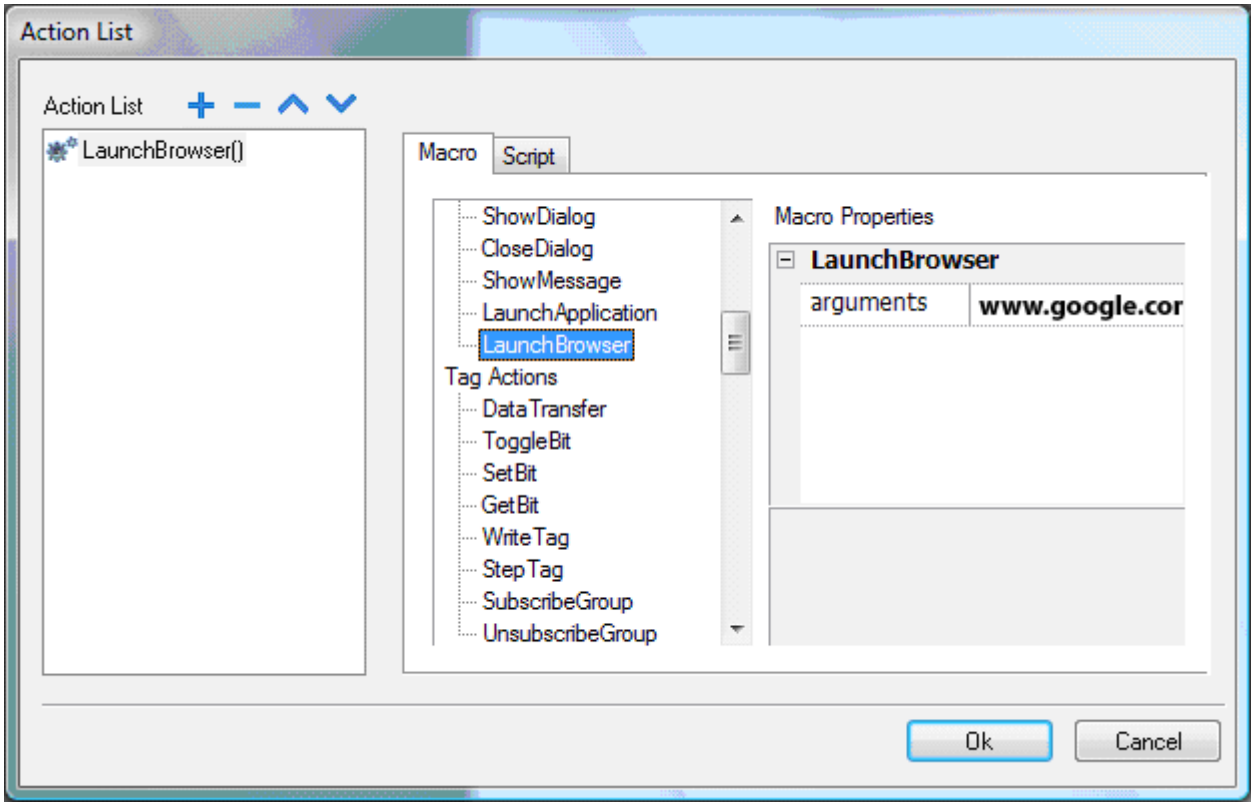

Figure 86

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.

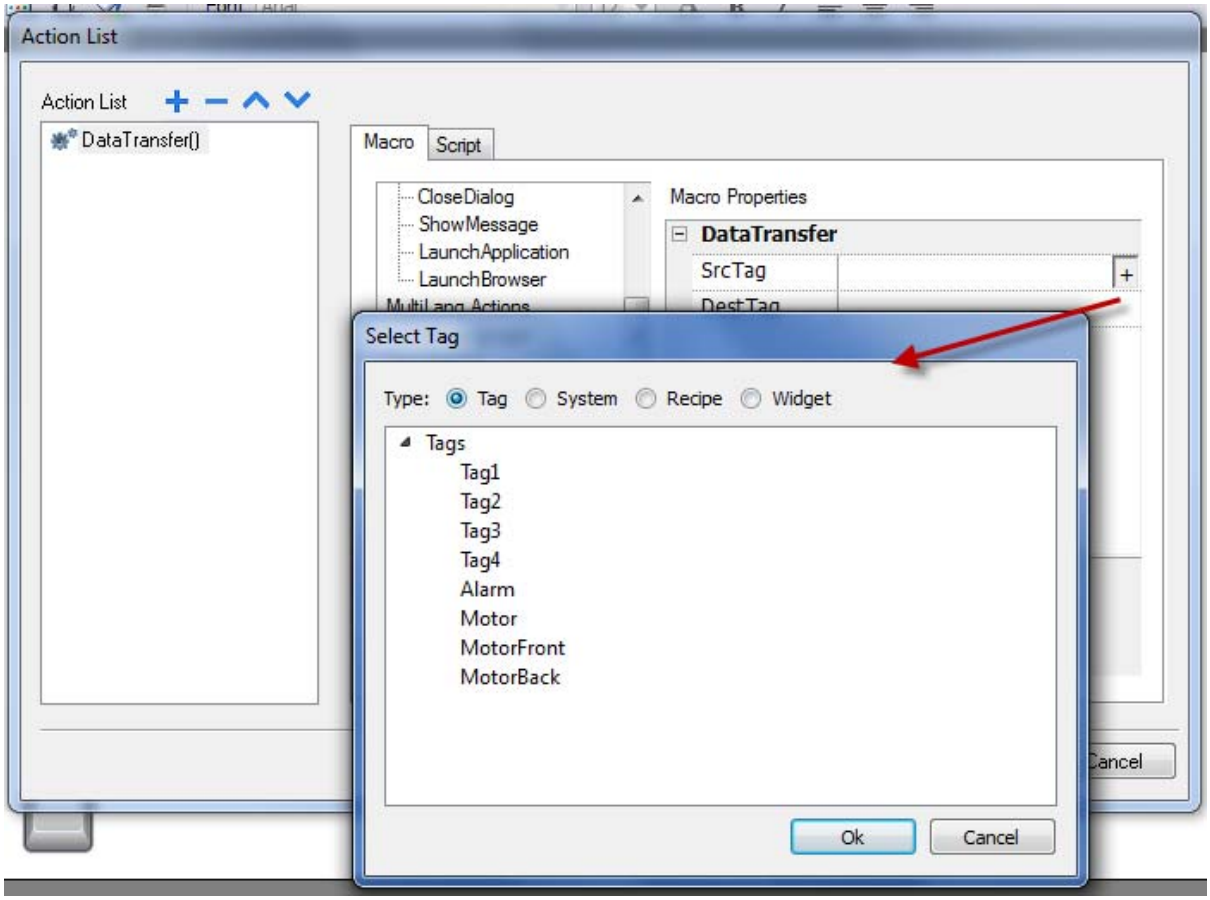

Figure 87

7.4.2 Toggle Bit

Toggle Bit macros allow you to “Toggle" (meaning set or reset) a Bit of a tag. The Bit Index allows you to select the Bit to be inverted: this implies a read-modify-write operation; the read value is inverted and then written back to the controller tag.

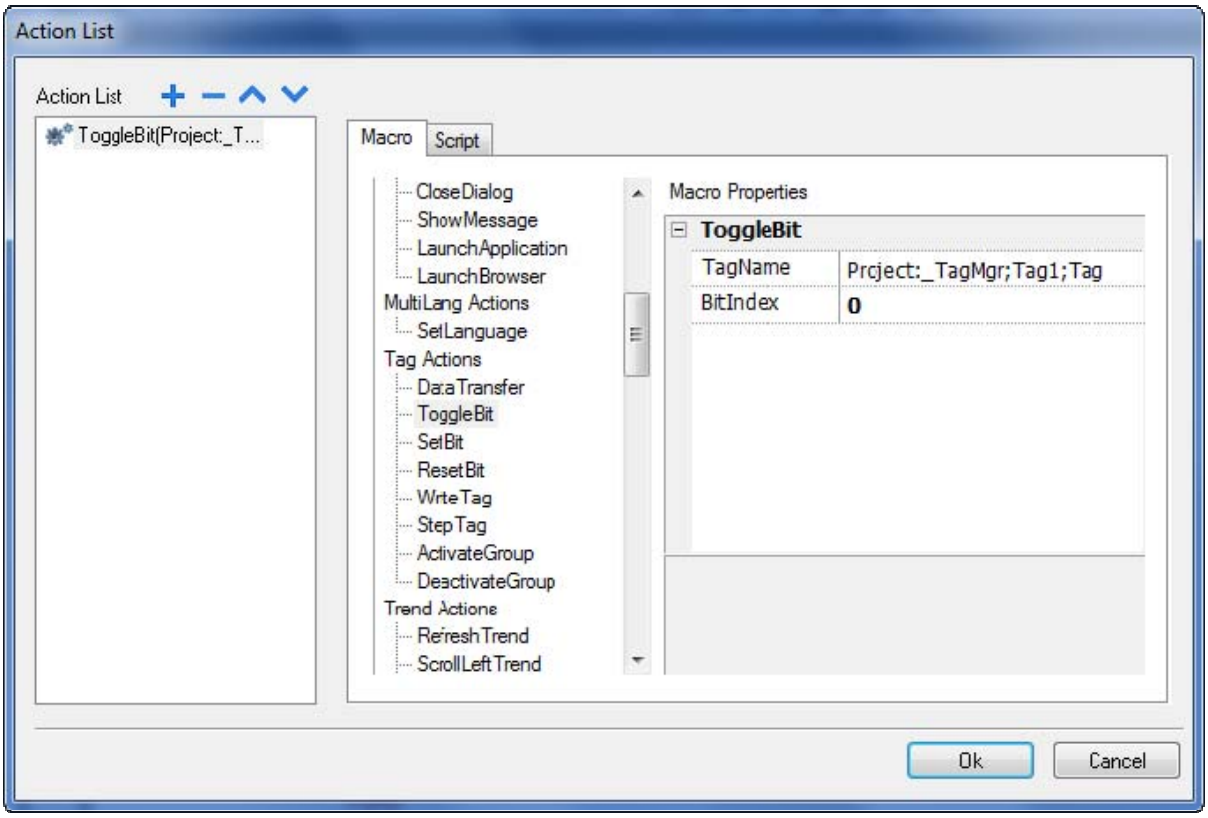

Figure 88

7.4.3 Set Bit

The Set Bit macro allows you to set the selected bit. When the macro is executed, the selected bit value is set to “1". The BitIndex property allows you to select the bit position inside the Tag.

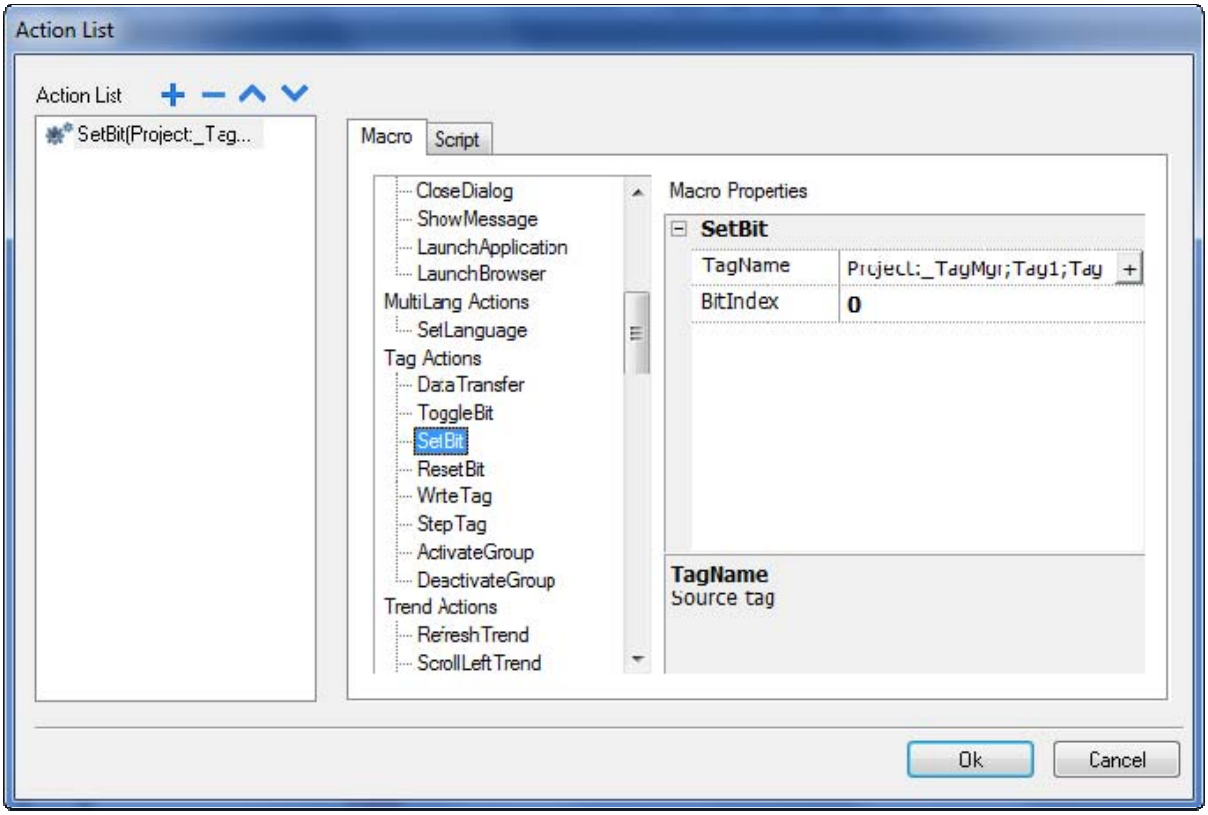

Figure 89

7.4.4 Reset Bit

The Reset Bit macro allows you to reset the selected bit. When the macro is executed, the selected bit value is set to “0”. The BitIndex property allows you to select the bit position inside the Tag.

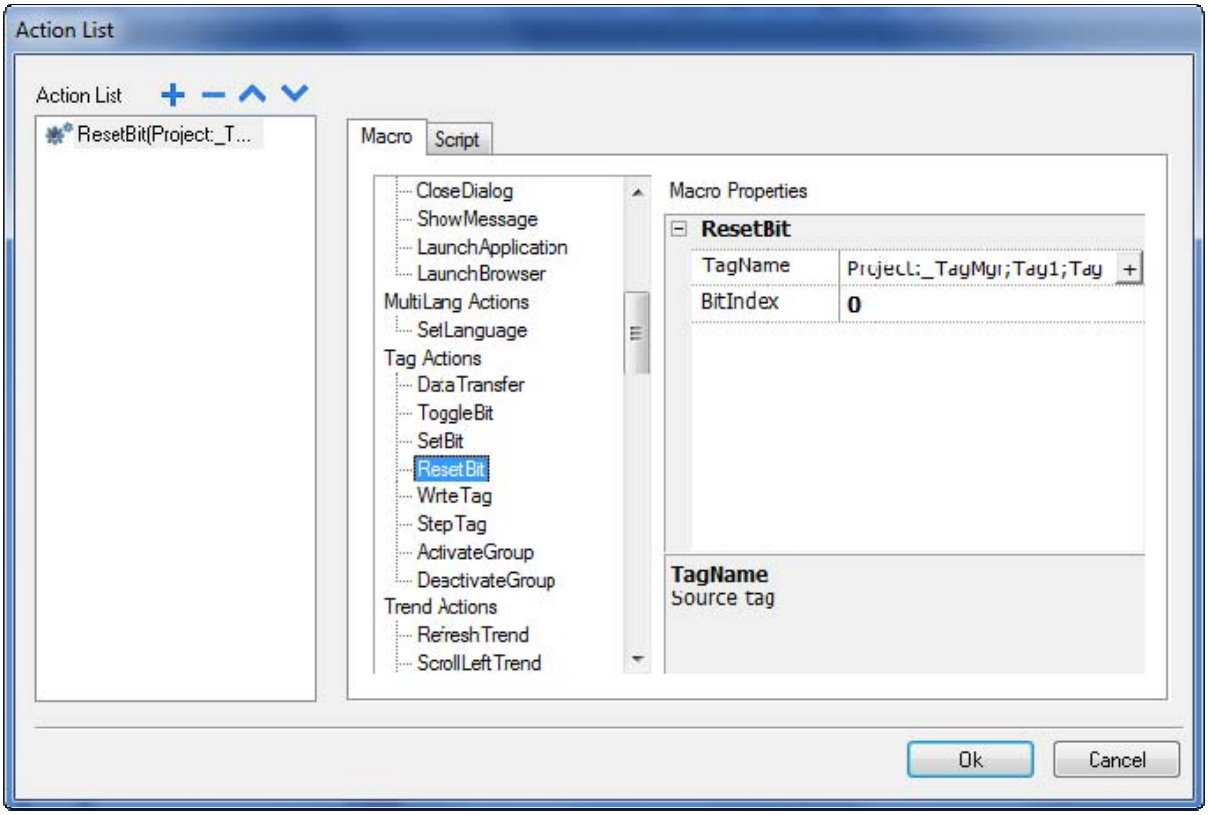

Figure 90

7.4.5 Write Tag

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

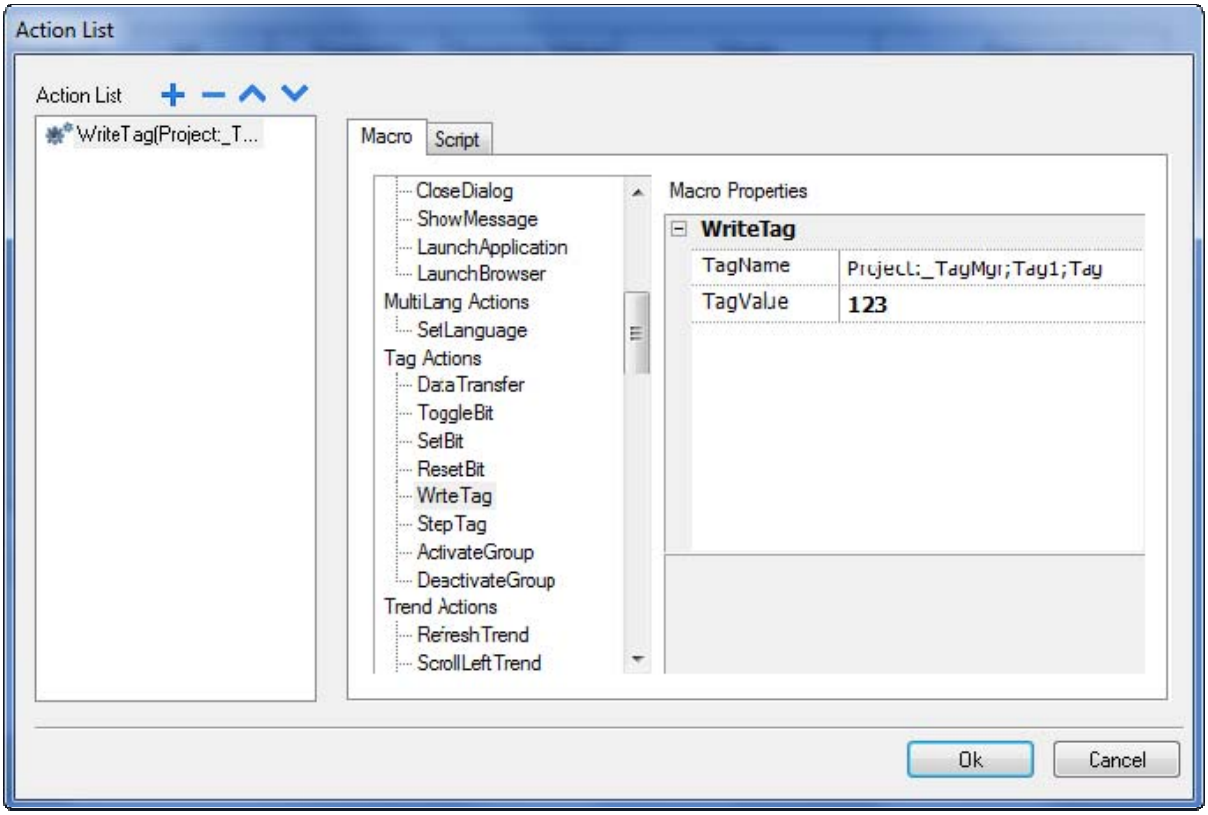

Figure 91

7.4.6 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.

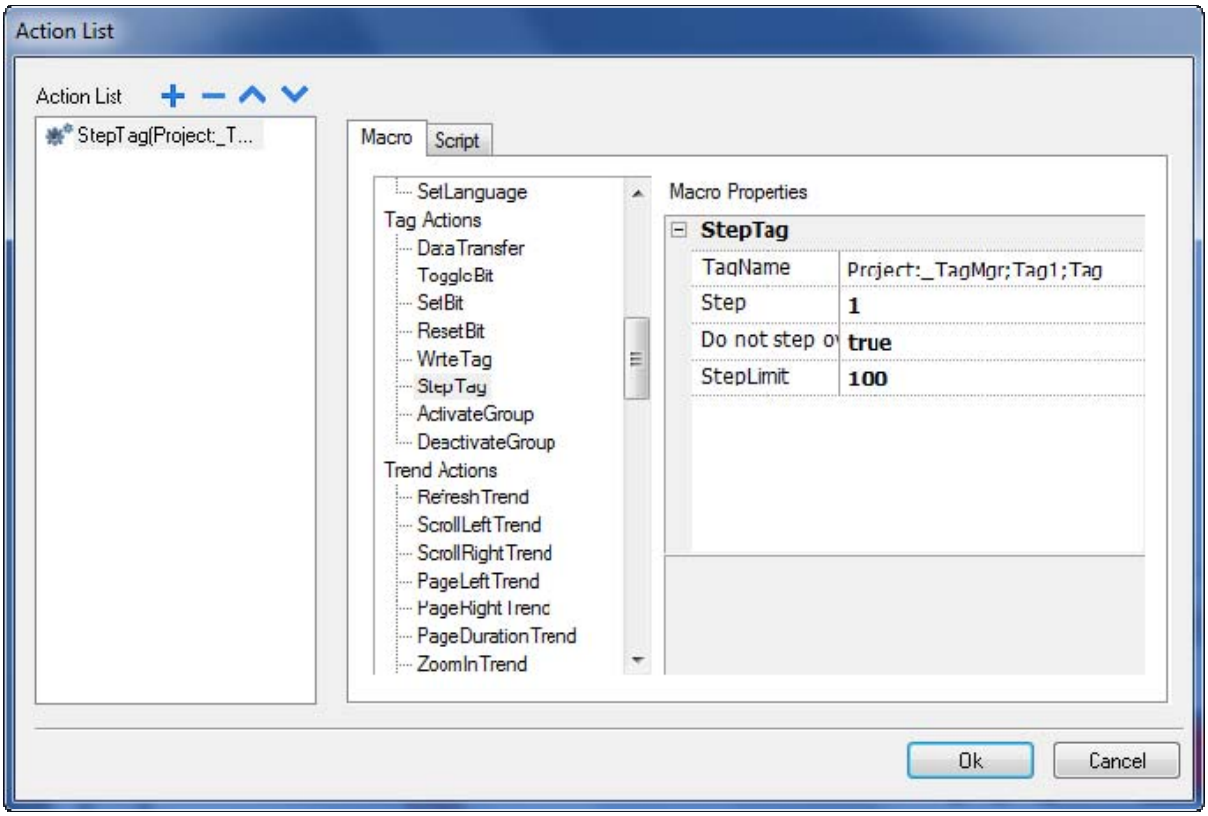

Figure 92

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 Scroll to Time Trend

The Scroll to time trend is used to scroll the Trend Window to a Particular point in time. When we execute this macro the Trend Window will move to the Time assigned in the Macro Action.

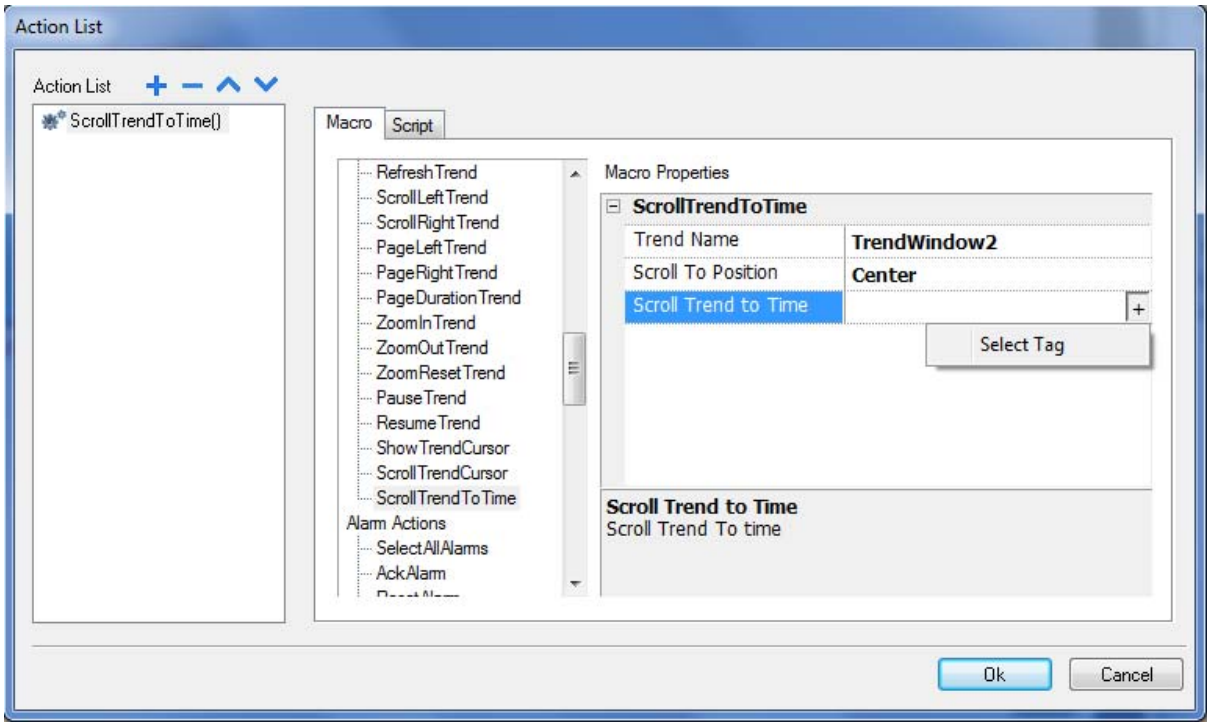

Figure 93

This Action may be very useful when you need to scroll at a specific time in a trend window based on the time at which a certain event occurs. This can be achieved by configuring an Action for that alarm (event) that executes a Data Transfer of the system time into a Tag; when selecting that tag as “Scroll trend to time” parameter (see above figure) the trend windows will be centered at the time in which the event has been triggered.

7.5.5 Page Left Trend

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

7.5.6 Page Right Trend

The Page Right Trend allows you to scroll the Trend window by 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 for 10 minutes.

7.5.7 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.

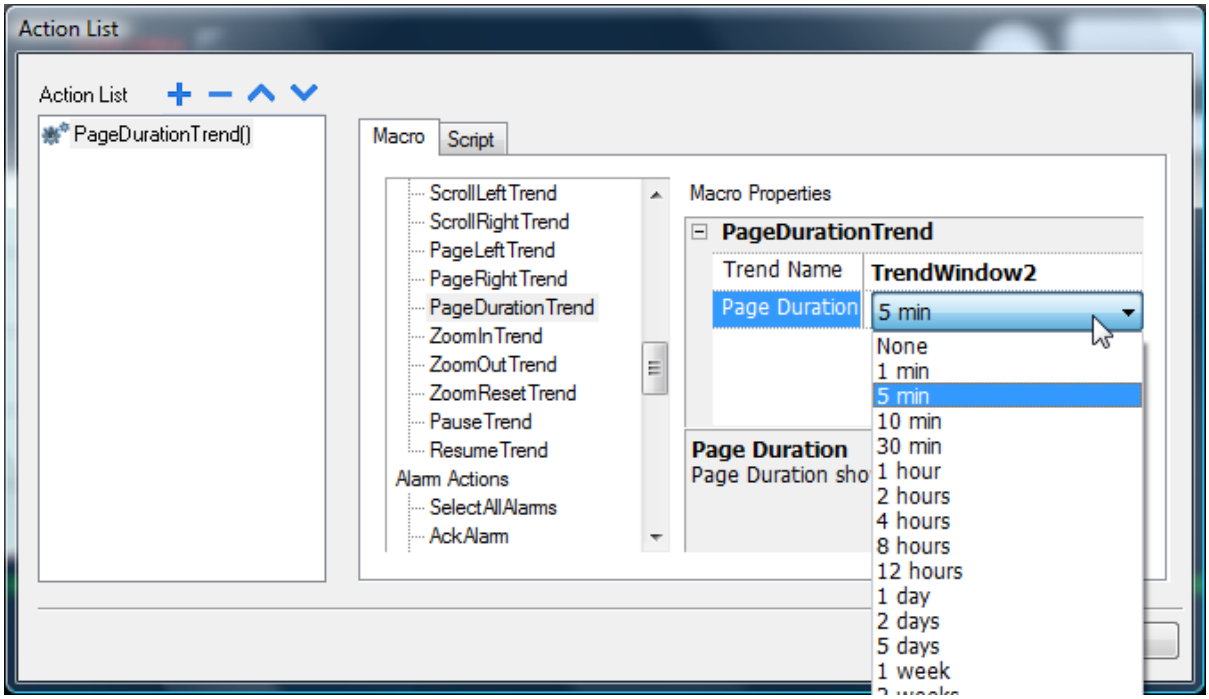

Figure 94

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

7.5.8 Zoom In Trend

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

7.5.9 Zoom Out Trend

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

7.5.10 Zoom Reset Trend

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

7.5.11 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.

7.5.12 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.

7.5.13 Show Trend Cursor

The Trend Cursor allows the user to know the value of the curve at a given point on the X-Axis. Use this macro 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.

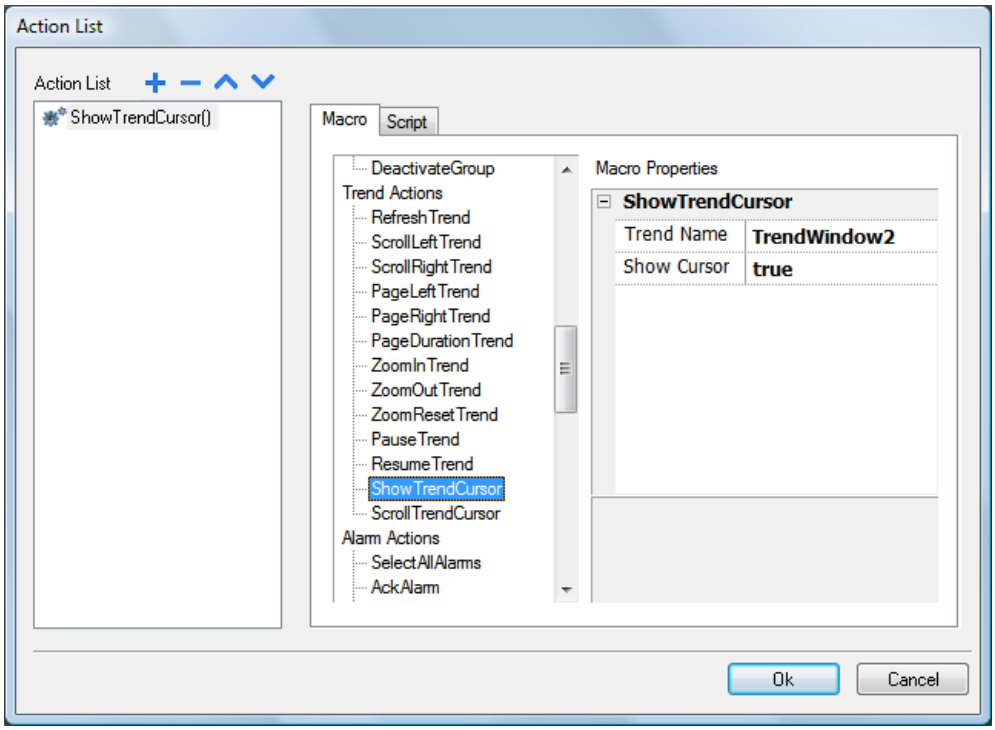

Figure 95

7.5.14 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.

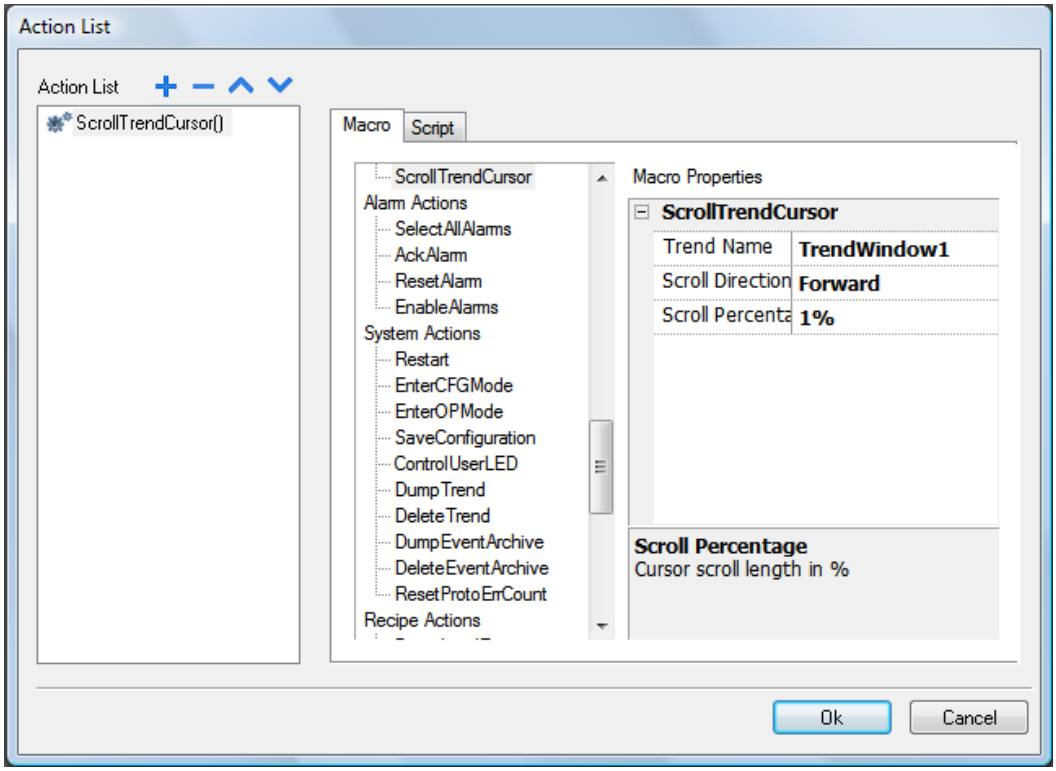

Figure 96

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.

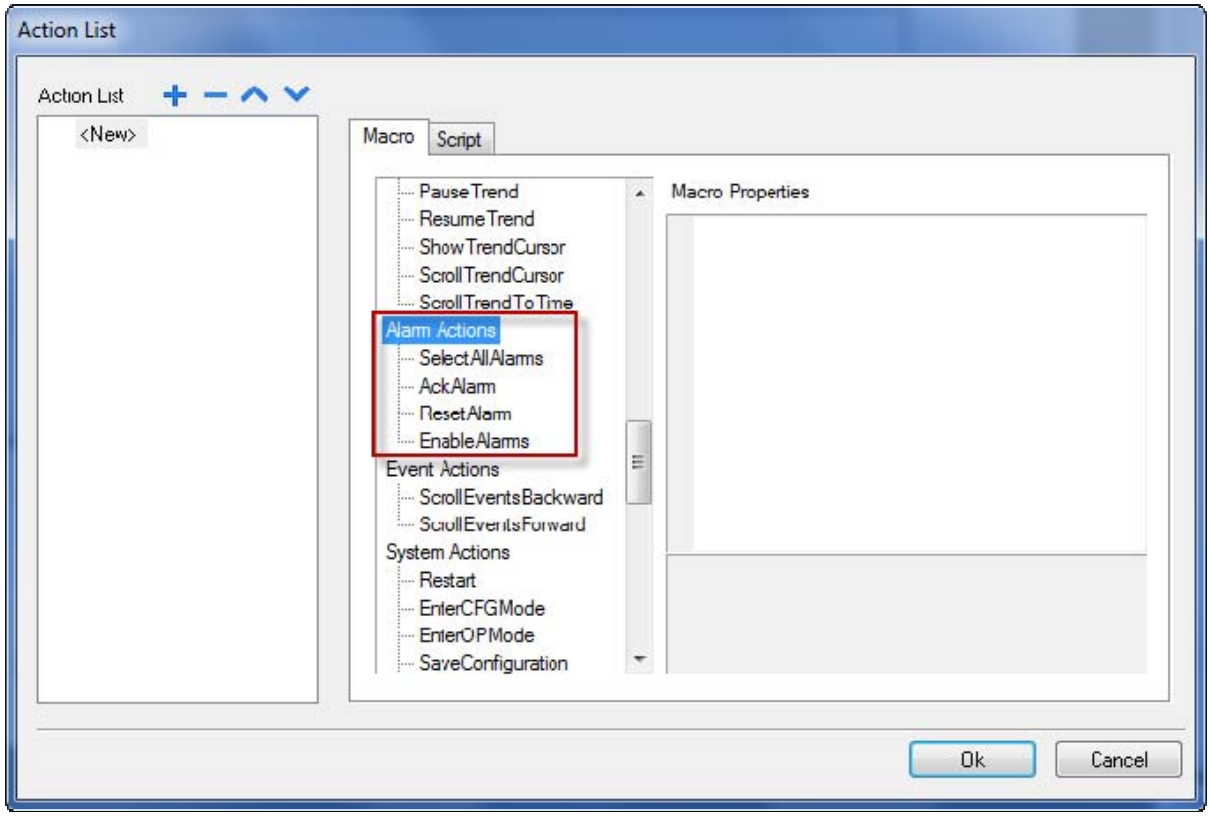

Figure 97

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.

7.6.4 Enable Alarms

The Enable Alarm Action is used within the “Save” button of the Alarm widget; it is required to properly save at runtime the changes done in the “Enable” check boxes from the “Enable” column in the alarm widget.

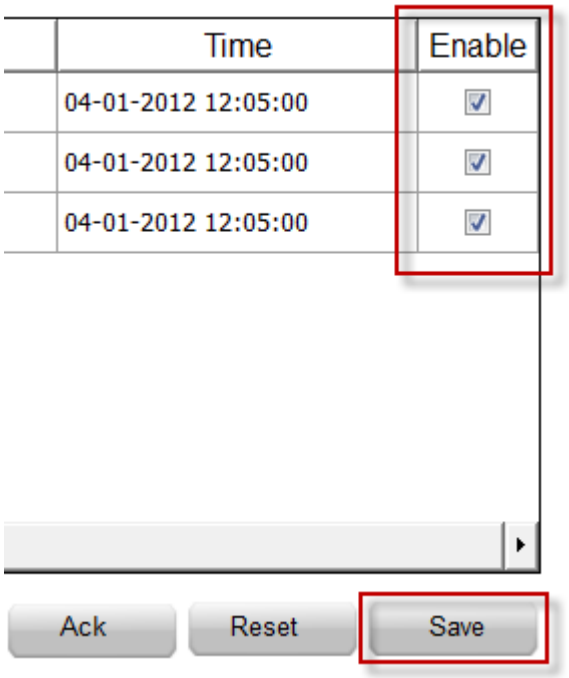

Figure 98

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 VisuControl Runtime. After executing the macro, the VisuControl server goes to configuration mode and restarts Runtime.

7.7.2 Enter Configuration Mode

This macro is used to switch the VisuControl runtime to Configuration mode (“Config").

7.7.3 Enter Operation Mode

This macro is used to switch the VisuControl 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 LED. In the action properties, the LED action can be set to OFF, ON or Blink.

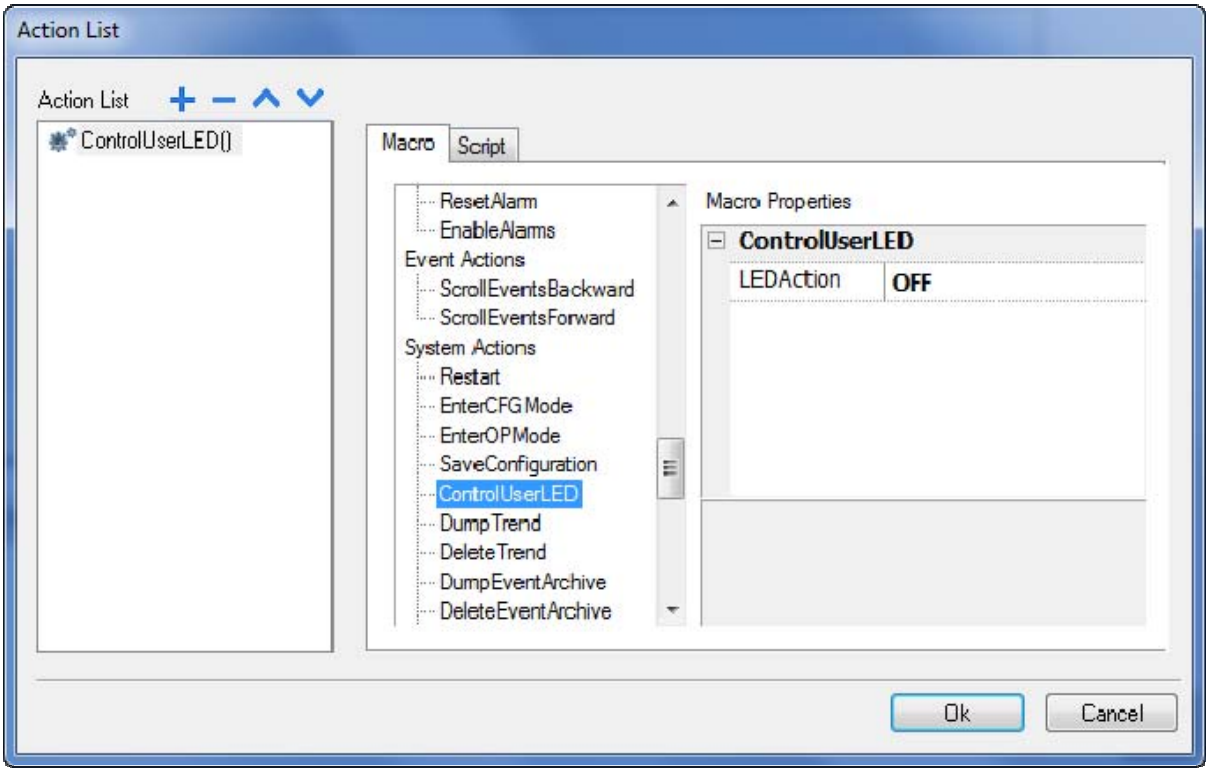

Figure 99

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 the 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.

Note: The execution of the Dump action will automatically force a flush to disk of the data temporarily maintained in the RAM memory. See the chapter “Trend Editor” for further information about the policy used to save sampled data to disk

Note: The external drives plugged on the USB port of the panel must be FAT or FAT32 formatted. NTFS format is not supported.

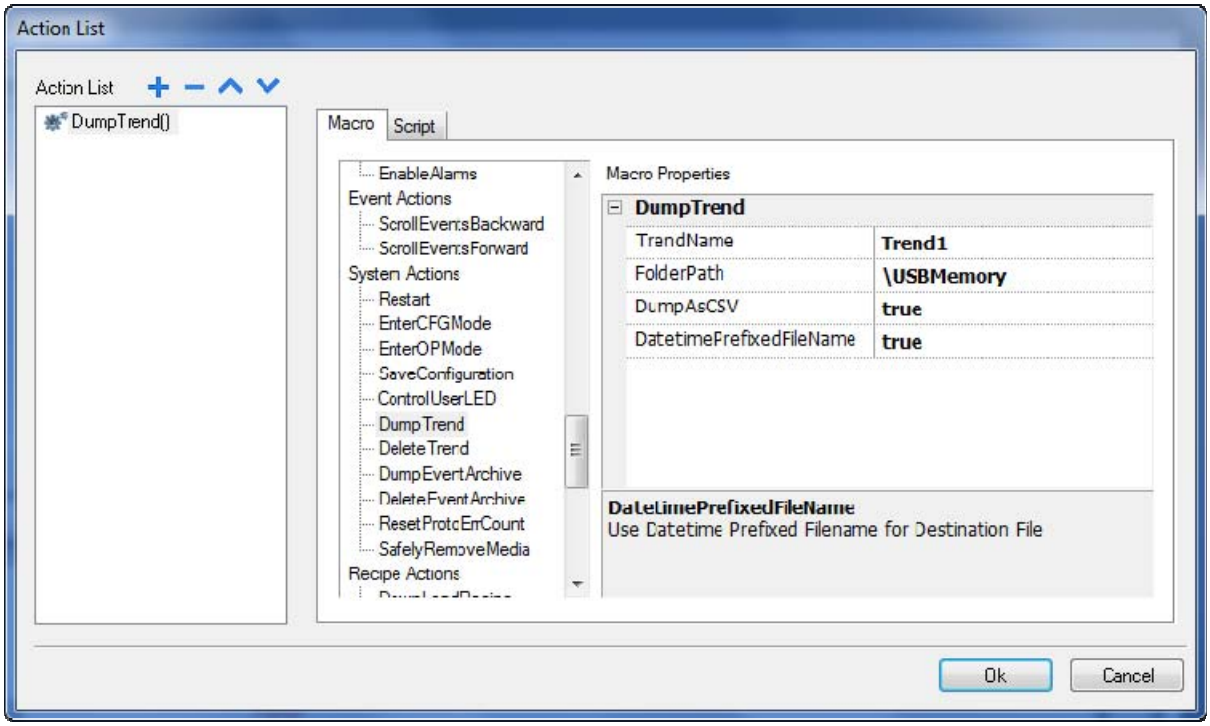

Figure 100

DumpAsCSV — If this option is set true, then the buffer will be directly dumped in the specified location as a *.CSV file in the format specified below. If it is set as False, then the dumped Trend file comes in a binary format; the result of the dump operation is actually a couple of files, one with extension .dat and one with extension .inf. An external utility is then required to convert it to a CSV format. The two above mentioned files are both required by the utility to operate the conversion.

DateTimePrefixFileName — When this Option is enabled the dumped File will have the Date and Time as Prefix to the name of the File. For example if we are making a Dump at 10.10AM on 1-1-2012, then the file name will look like D2012_01_01_T10_10_Trend1.csv.
[DYear_Month_day_THour_Minute_Filename]
This helps to know the Time at which the Dump was made and also to identify which one is the latest.

Note: — The utilities required to convert the dumped files to CSV are available in the VisuControl folder called “Utils” available under the directory where the software is installed

The utility needed to convert trend buffers is called “TrendBufferReader.exe".

The TrendBufferReader.exe utility can be invoked using a batch file with the following syntax:

TrendBufferReader -r Trend1 Trend1.csv 1

where Trend1 is the name of the trend buffer without extension as dumped (original file name is trend1.dat) and Trend1.csv is the name desired for the output file.

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. The information is coded according the OPC DA standard; the information is stored in a byte data (8 bits) currently defined in the form of three bit fields; Quality, Substatus and Limit status.
The 8 Quality bits are arranged as follows:
QQSSSSLL

For a complete and detailed description of all the single fields, please refer to the OPC DA official documentation. Here below we report the most commonly used quality values returned by the HMI acquisition engine:

Quality Code	Quality	Description
0	BAD	The value is bad but no specific reason is known
4	BAD	There is some server specific problem with the configuration. For example the tag is question has been deleted from the configuration file (tags.xml).
8	BAD	This quality may reflect that no value is available at this time, for reasons like the value may have not been provided by the data source.
12	BAD	A device failure has been detected
16	BAD	Timeout occurred before device responded.
24	BAD	Communications have failed.
28	BAD	There are no data found to provide upper or lower bound value (trend interface specific flag).
32	BAD	No data have been collected (i.e. archiving not active. Trend interface specific flag).
64	UNCERTAIN	There is no specific reason why the value is uncertain.
65	UNCERTAIN	There is no specific reason why the value is uncertain. (The value has ‘pegged’ at some lower limit)

66	UNCERTAIN	There is no specific reason why the value is uncertain. (The value has ‘pegged’ at some high limit.)
67	UNCERTAIN	There is no specific reason why the value is uncertain. (The value is a constant and cannot move.)
84	UNCERTAIN	The returned value is outside the limits defined for it. Note that in this case the “Limits” field indicates which limit has been exceeded but the value can move farther out of this range.
85	UNCERTAIN	The returned value is outside the limits defined for it. Note that in this case the “Limits” field indicates which limit has been exceeded but the value can move farther out of this range. (The value has ‘pegged’ at some lower limit)
86	UNCERTAIN	The returned value is outside the limits defined for it. Note that in this case the “Limits” field indicates which limit has been exceeded but the value can move farther out of this range. (The value has ‘pegged’ at some high limit.)
87	UNCERTAIN	The returned value is outside the limits defined for it. Note that in this case the “Limits” field indicates which limit has been exceeded but the value can move farther out of this range. (The value is a constant and cannot move.)
192	GOOD	

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 export Historical Alarm log and Audit Trail data to external drives, such as a hard drive or USB memory.

Note: The external drives plugged on the USB port of the panel must be FAT or FAT32 formatted. NTFS format is not supported.

In the action properties, you need to configure the Event buffer name that you want to dump and the destination folder path, the DumpConfigFile property must be set to true when you plan to convert the dumped files to CSV.

DumpAsCSV If this option is set true, then files will be directly dumped in the location in *.CSV format. If it is set as False, then the Events archive is dumped in a binary format; an external utility is then required to convert it to a CSV format.

DateTimePrefixFileName When this Option is enabled then dumped File will have the Date and Time as prefix to the name of the file.
For example if we are making a Dump at 10.10AM on 1-1-2012, then the file name will look like
D2012_01_01_T10_10_alarmBuffer1.csv.
[DYear_Month_day_THour_Minute_Filename]
This helps to know the Time at which the Dump was made and also to identify which one is the latest.

Note: This option is only supported when exporting to CSV directly

When exporting Alarm buffers in binary format assuming the DumpConfigFile option is set to true (recommended settings) the result of the dump action execution is 2 folders; one is called “data” and it contains the data files, the second one is called “config” and it does contain the configuration files needed by the utility to reconstruct the entire information for proper conversion to CSV.

Once the two folders are copied from the root of the USB drive to the computer disk, the folder structure will look as follows:

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

Note: the utility is distributed in VisuControl under the ..\MDT\VisuControl\Utils folder.

The AlarmBufferreader can be called from command line with the following syntax:

AlarmBufferReader AlarmBuffer1 FILE ./AlarmBuffer1.csv

Where AlarmBuffer1 is the name of the dumped .dat file without extension and AlarmBuffer1.csv is the desired output file name.

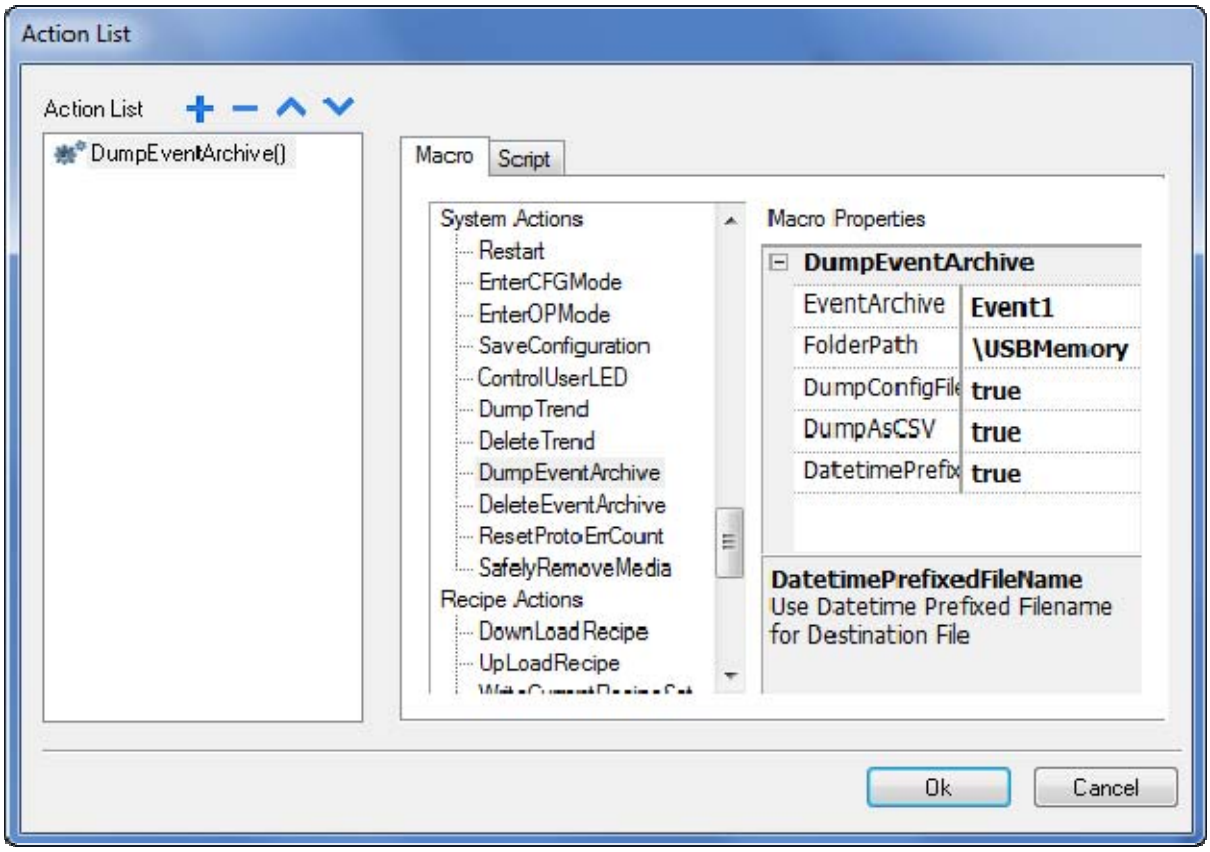

Figure 101

The utility called “AuditTrailBufferReader.exe" has to be used for Audit Trail buffers.

Note: The action has to be configured with the option DumpConfigFile set to true.

The result of the dump is a directory structure similar to the one generated for Events.
The utility can be called from the command line according to the following syntax:

AuditTrailBufferReader AuditTrail FILE ./AuditTrail.csv

Where AuditTrail is the name of the dumped buffer without extension and AuditTrail1.csv is the desired output file name.

7.7.9 Delete Event Archive

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

7.7.10 Reset ProtoErr Count

The Reset Protocol Err Count macro is used to reset the Protocol Error Count System Variable. See the chapter “System Variables“ for further information about system variables.

7.8 Multi Language 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 in the Language index. In Runtime, while executing the macro, the selected language will be applied to all applicable Widgets.

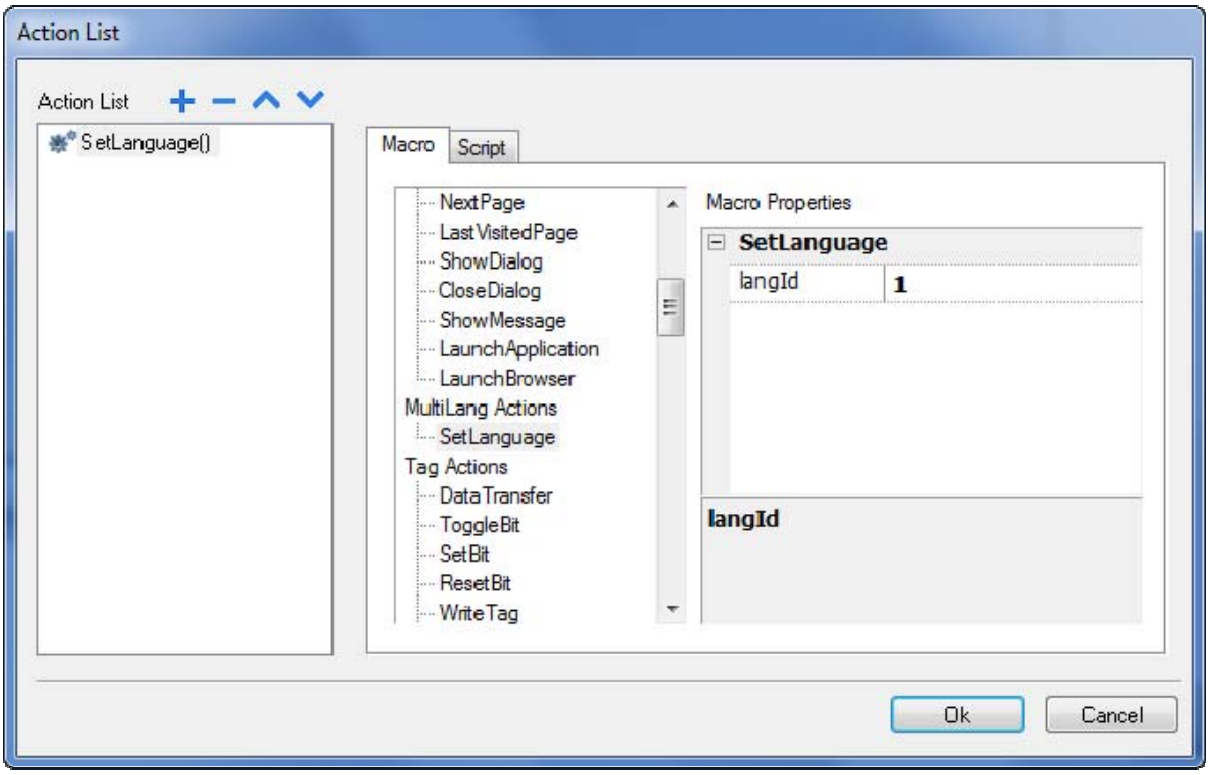

Figure 102

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.

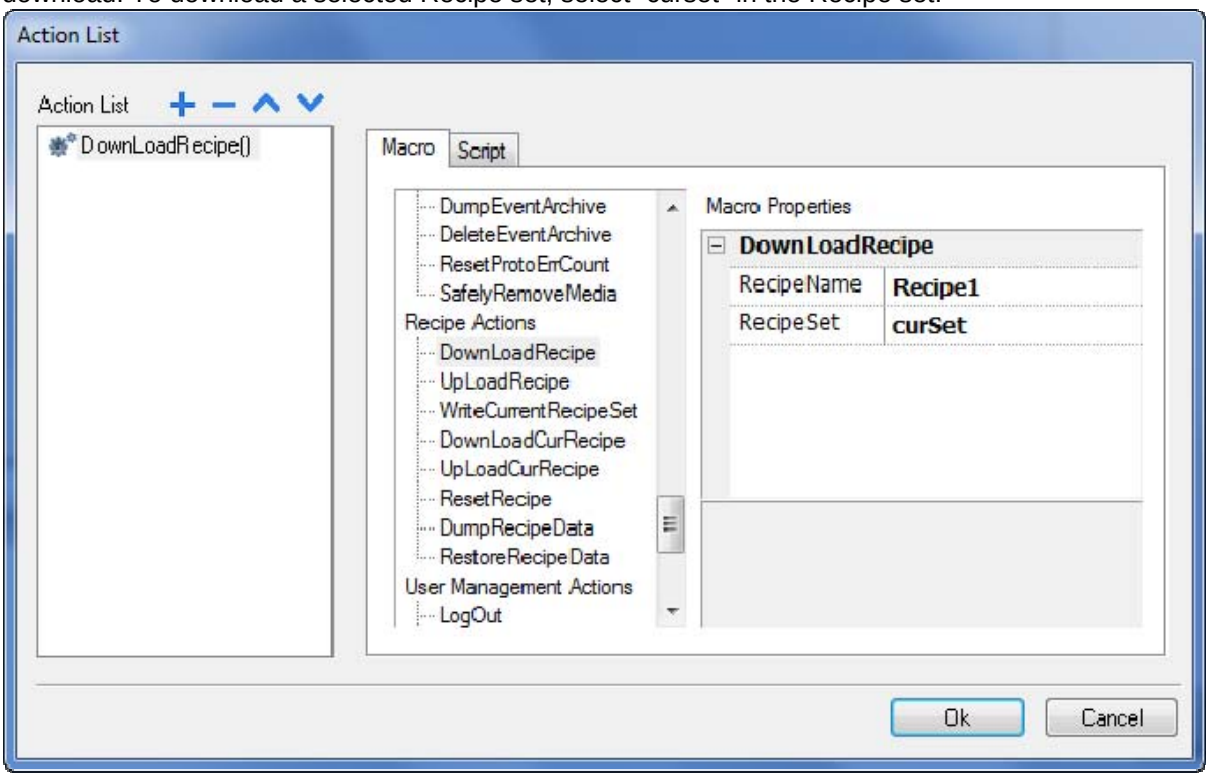

Figure 103

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.

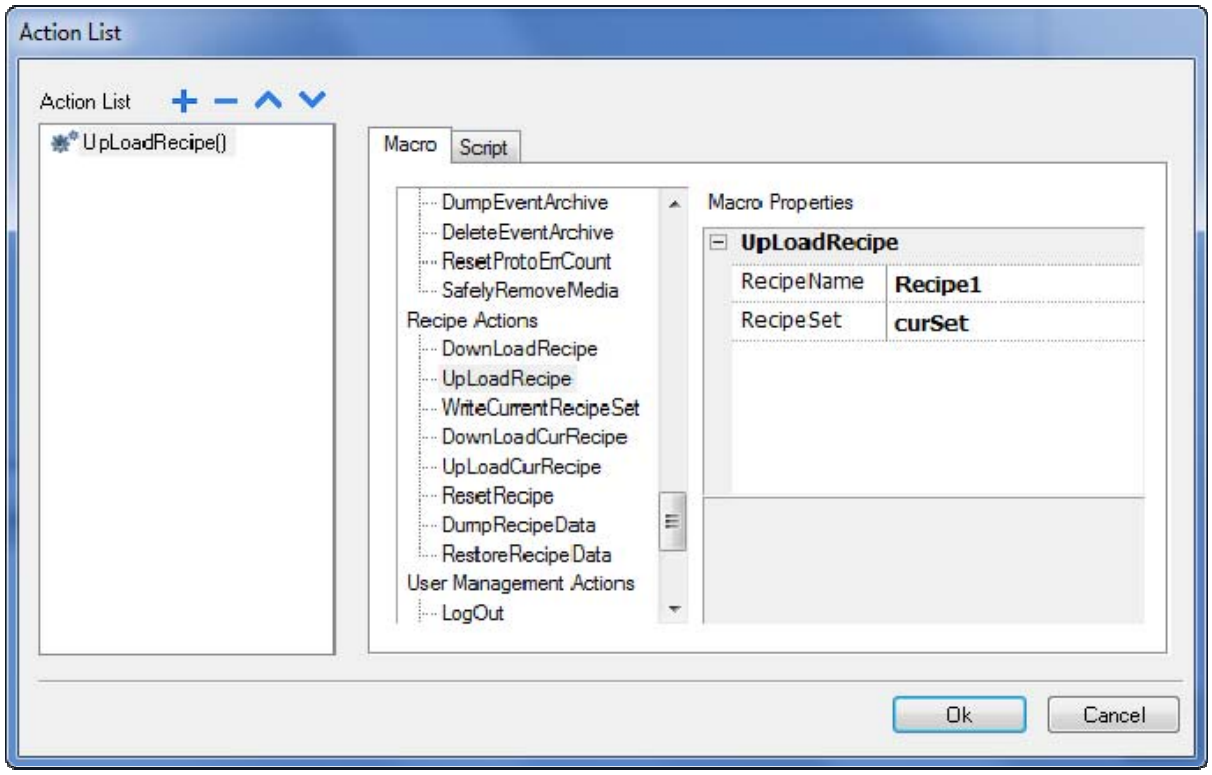

Figure 104

7.9.3 Write Current Recipe Set

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.

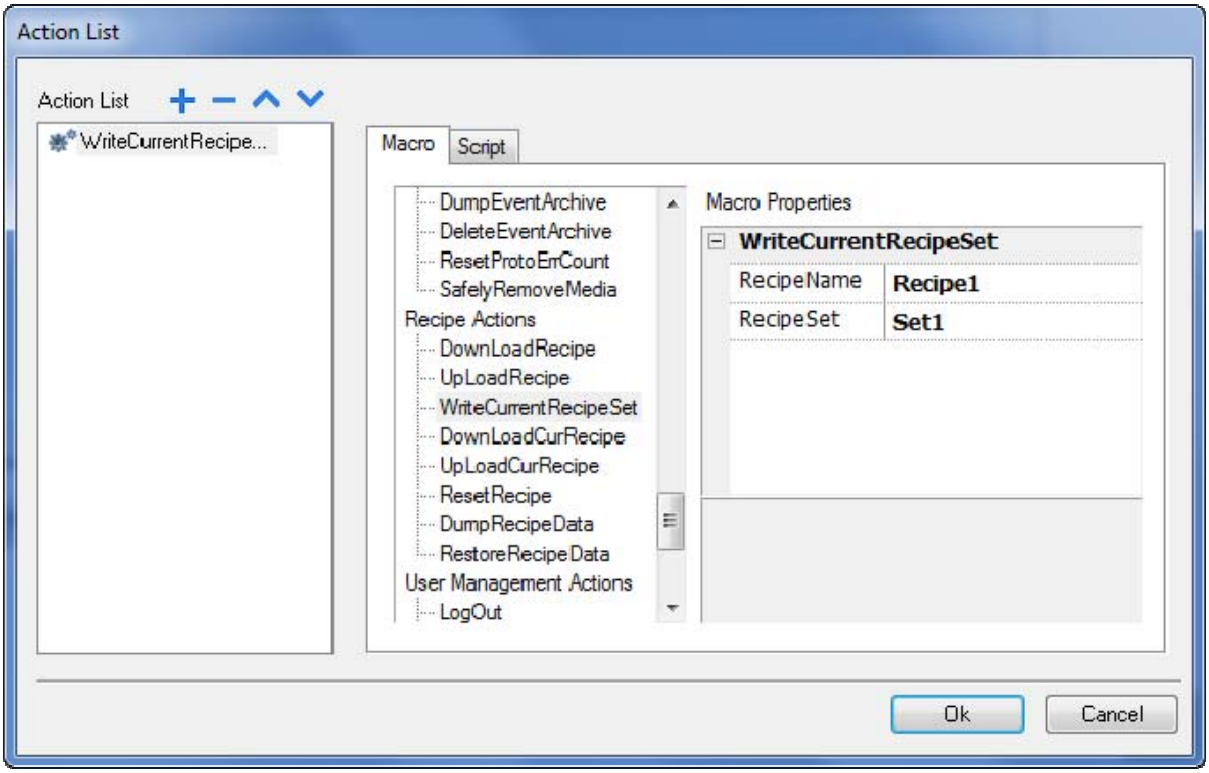
Action List
Action List
WriteCurrentRecipe...
Macro
Script
DumpEventArchive
DeleteEventArchive
ResetProtoErrCount
SafelyRemoveMedia
Recipe Actions
DownLoadRecipe
UpLoadRecipe
WriteCurrentRecipeSet
DownLoadCurRecipe
UpLoadCurRecipe
ResetRecipe
DumpRecipeData
RestoreRecipeData
User Management Actions
LogOut
Macro Properties
WriteCurrentRecipeSet
RecipeName
Recipe1
RecipeSet
Set1
Ok
Cancel

Figure 105

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.

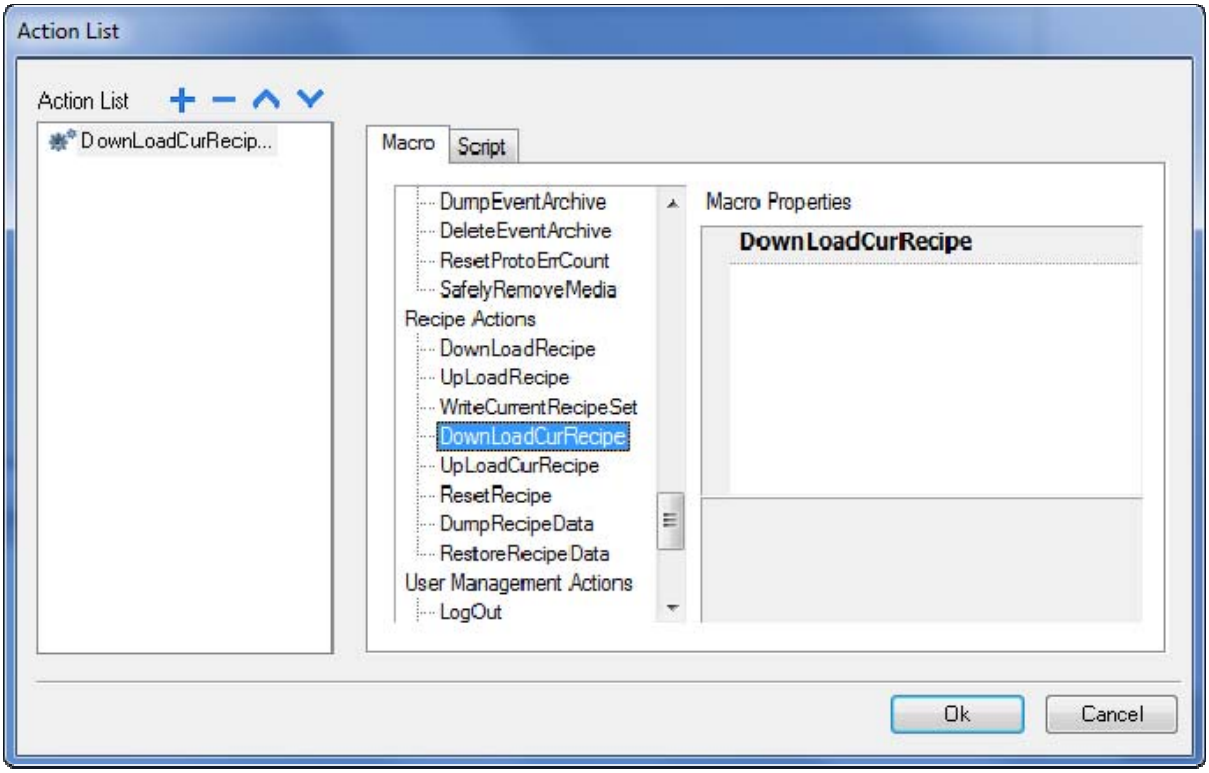

Figure 106

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.

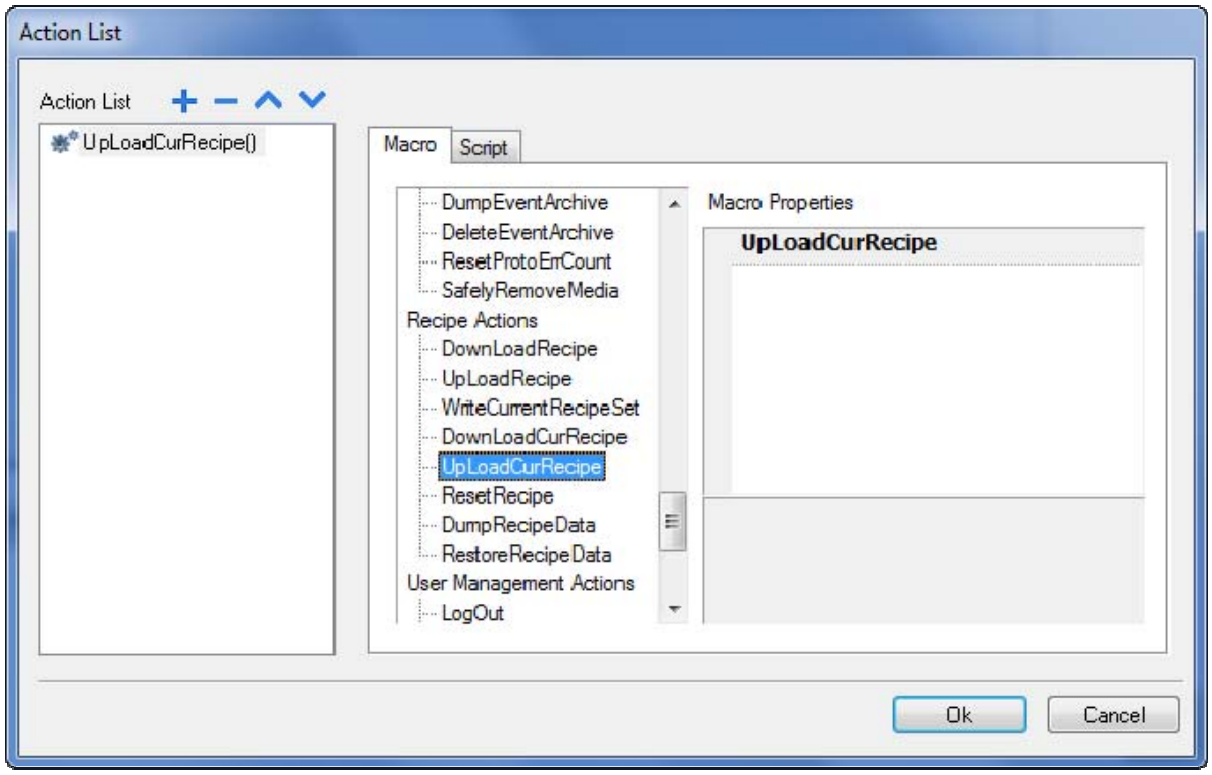

Figure 107

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.

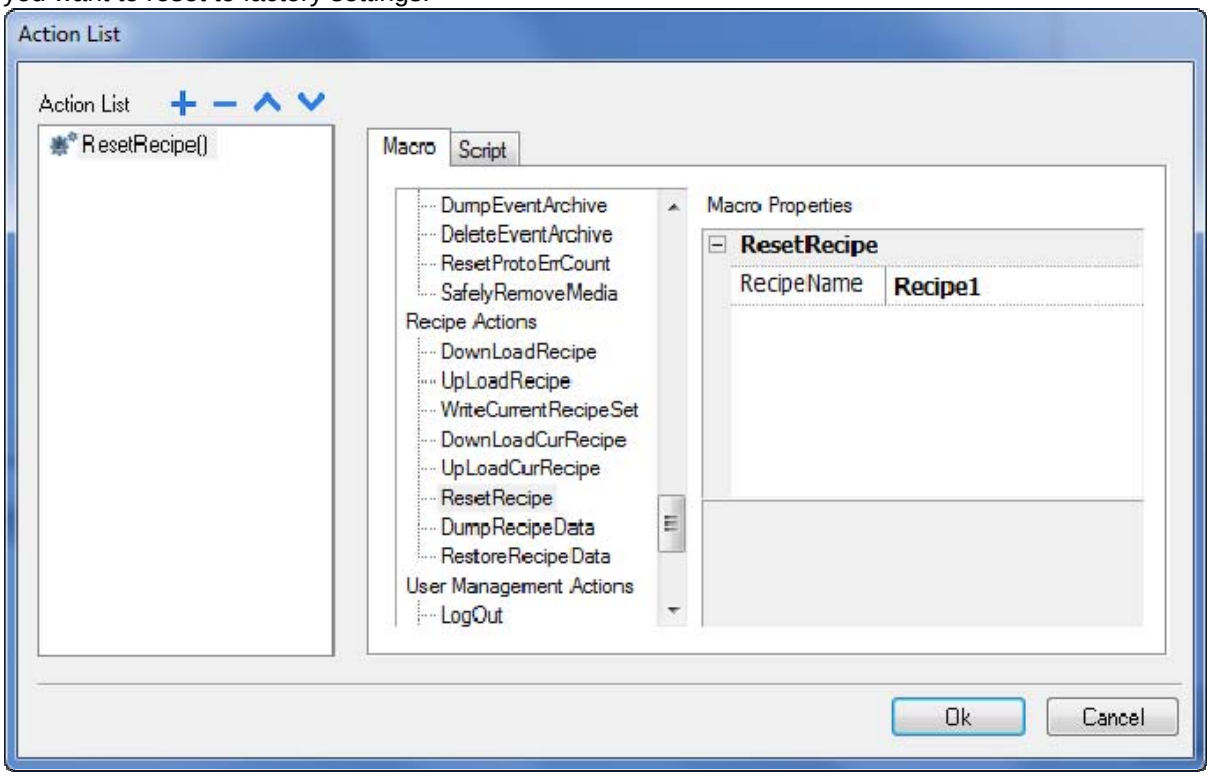

Figure 108

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.

Note: The external drives plugged on the USB port of the panel must be FAT or FAT32 formatted. NTFS format is not supported.

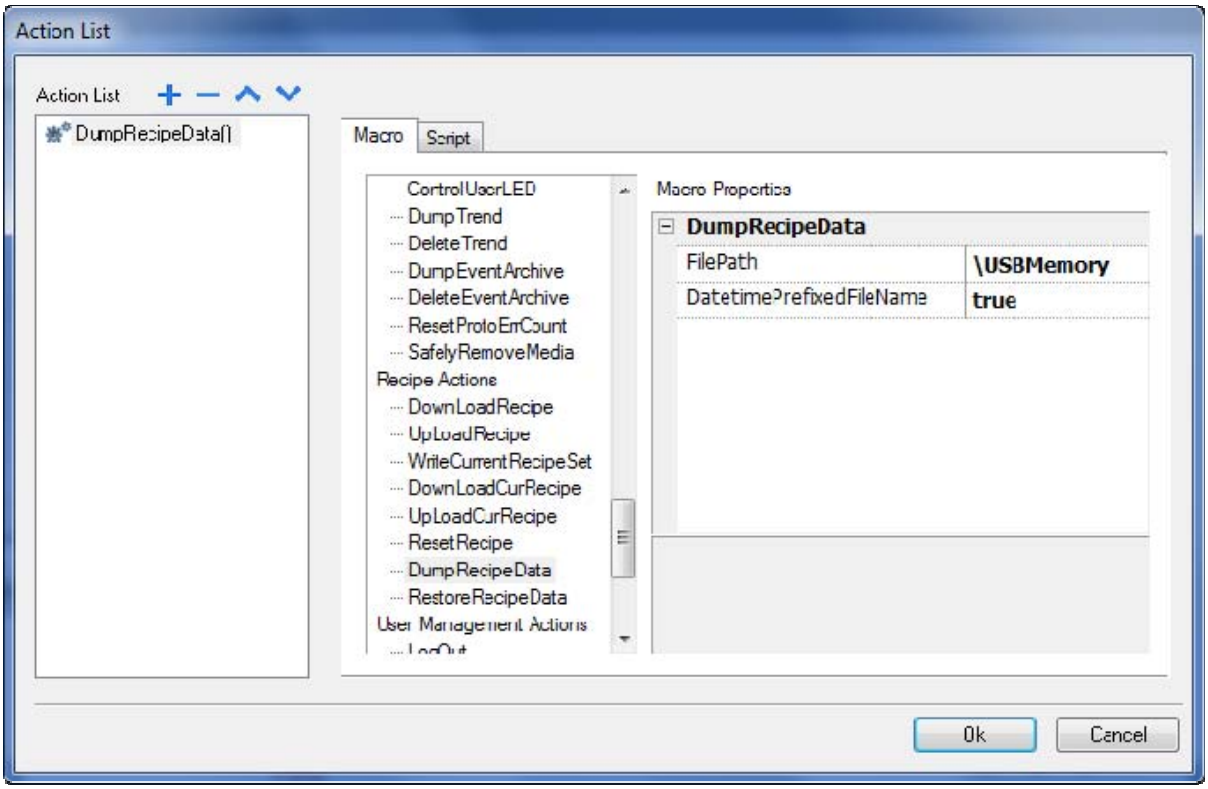

Figure 109

DateTimePrefixFileName When this Option is enabled then the dumped file will have the Date and Time as prefix of the name of the File.

For example: if we are making a Dump at 10.10AM on 1-1-2012, then the file name will look like D2012_01_01_T10_10_recipe1.csv. [DYear_Month_day_THour_Minute_Filename]
This helps to know the Time at which the Dump was made and also to identify which one is the latest.

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. In the figure below the example shows the case of a USB external pen drive.

Note: The external drives plugged on the USB port of the panel must be FAT or FAT32 formatted. NTFS format is not supported.

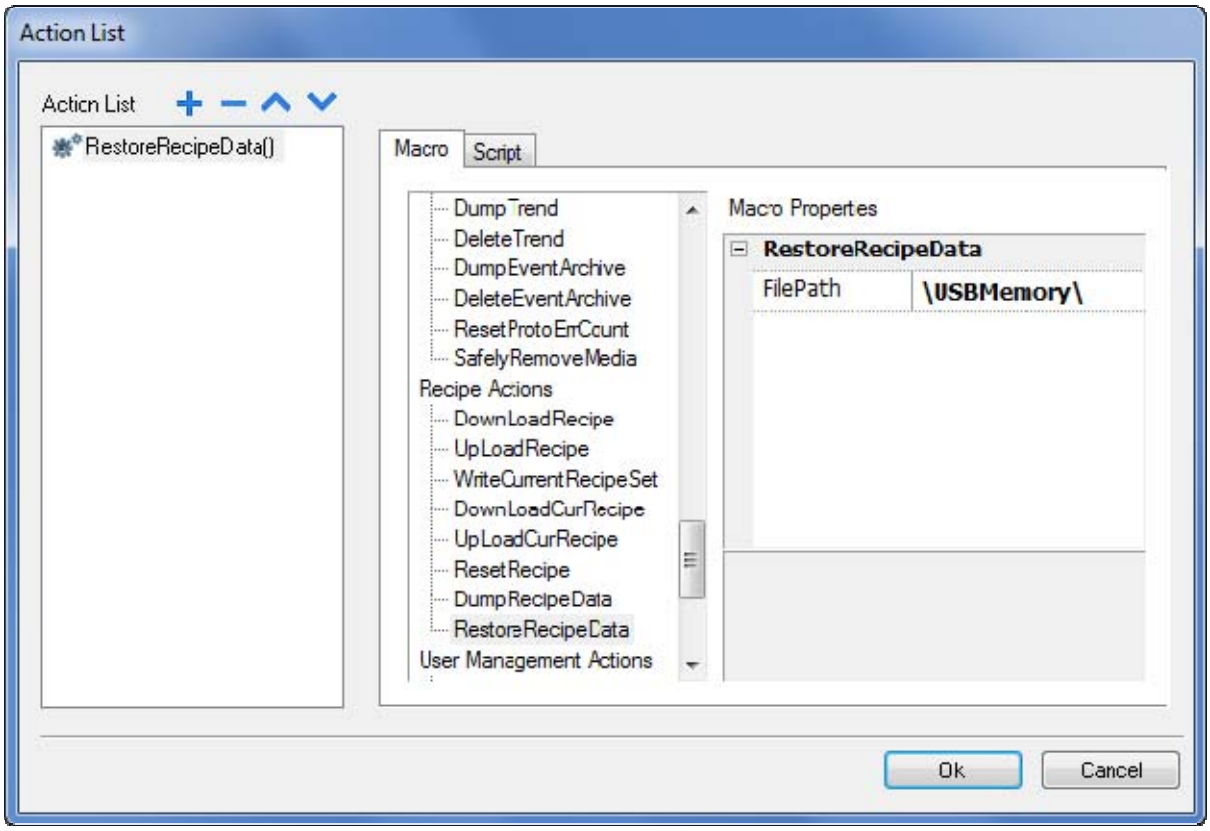

Figure 110

7.9.9 User Management Actions

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

7.9.10 LogOut

The Logout macro allows you to log off the current user in Runtime. After executing the Logout macro, the HMI behavior depends on the fact that a Default user is configured or not in the project.
If there is a Default user, the Logout automatically logs in the Default user. If there is not a Default user or you log out from the Default user, then the log in screen is shown.

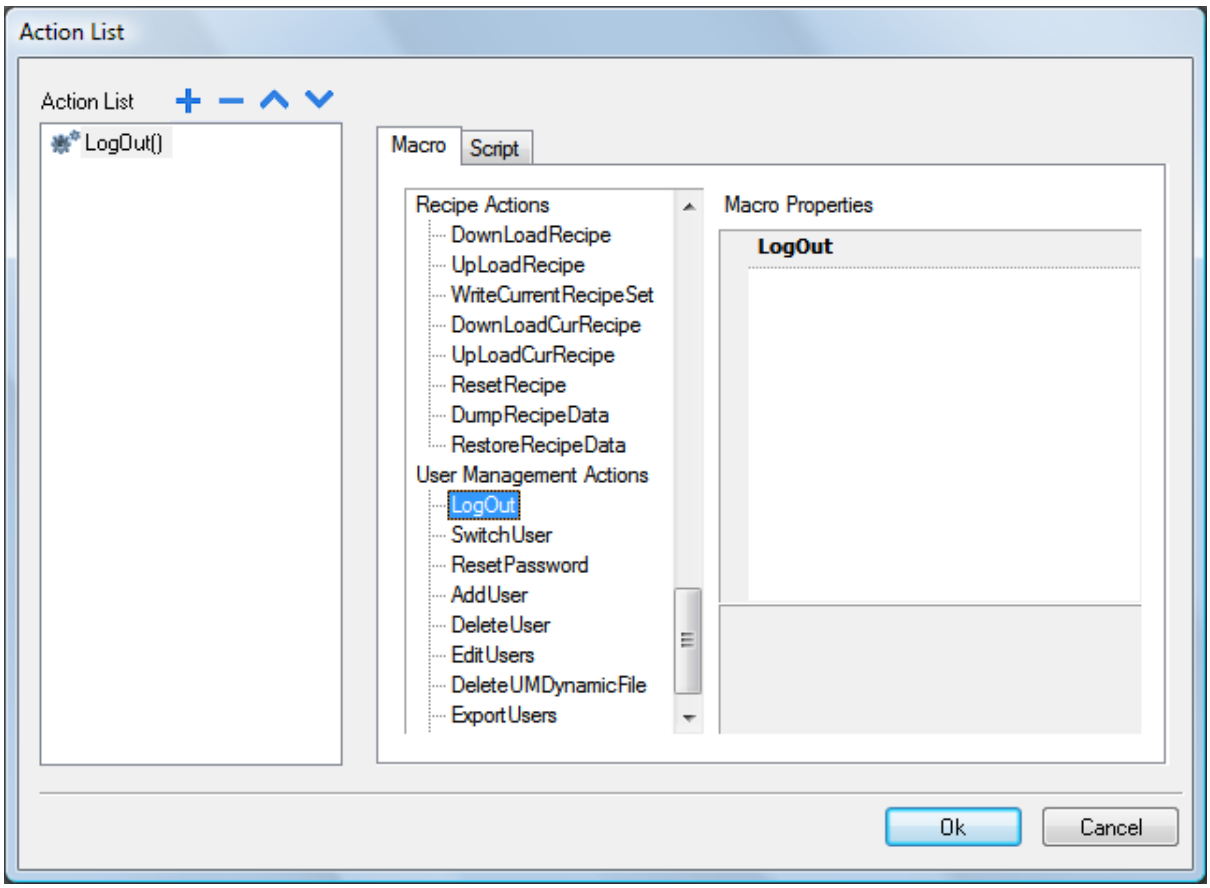

Figure 111

7.9.11 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 onto the system.

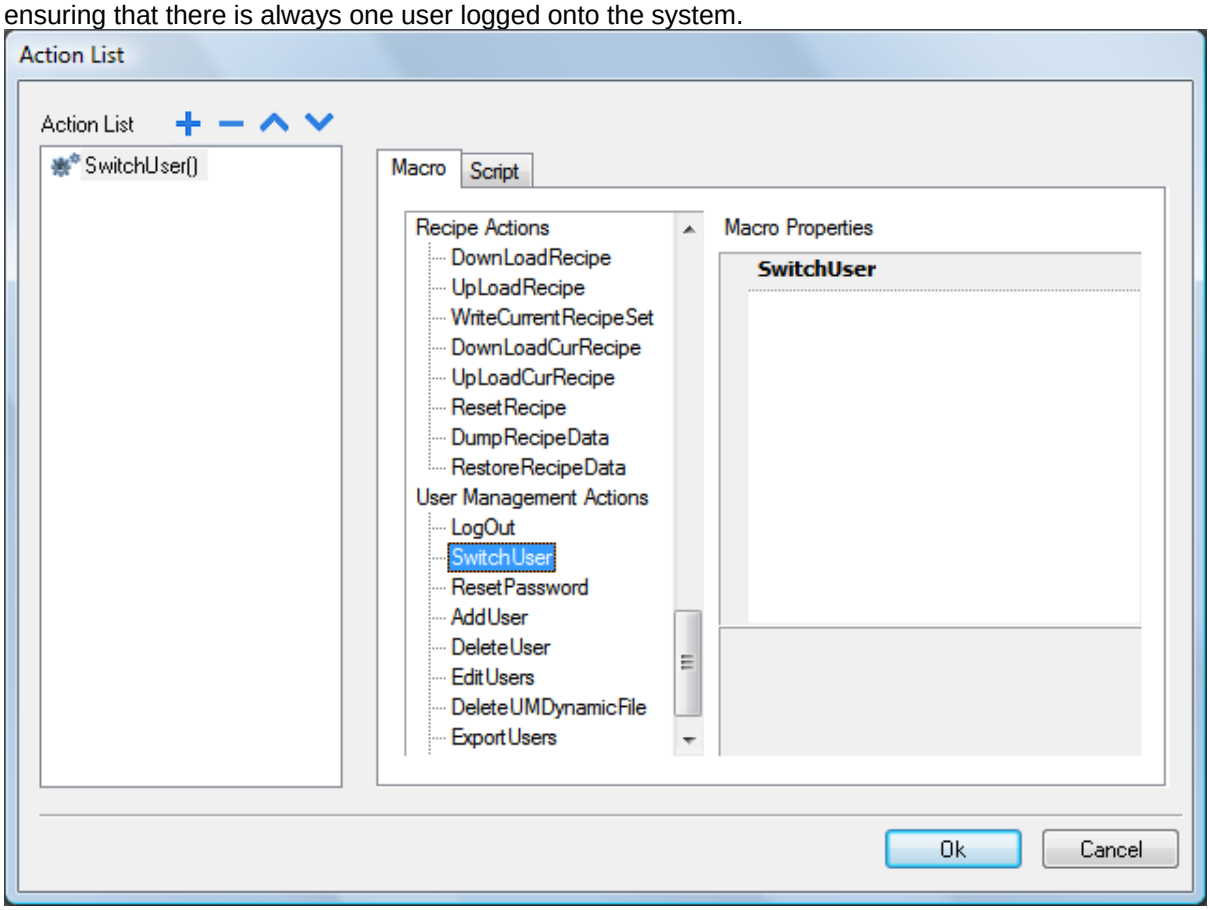

Figure 112

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

Figure 113

7.9.12 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.

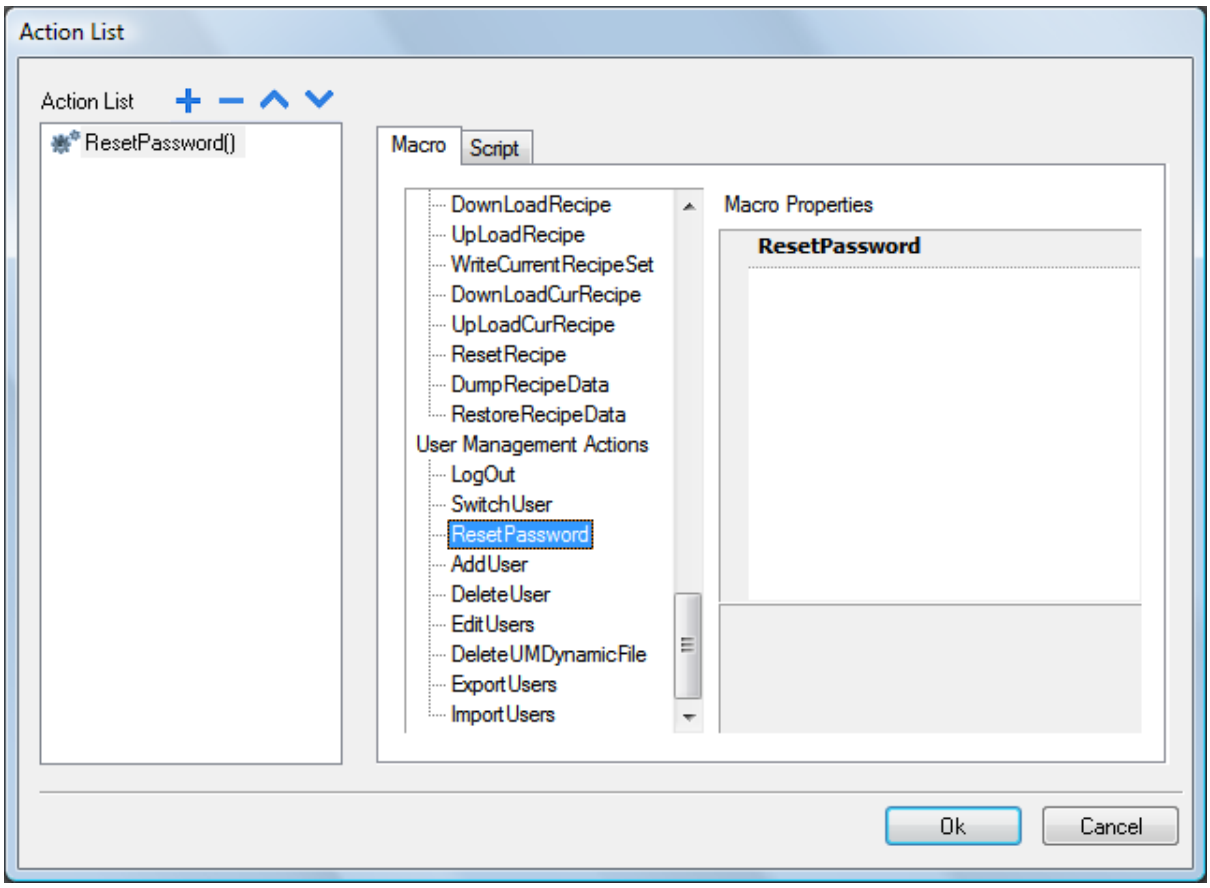

Figure 114

7.9.13 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 below figure.

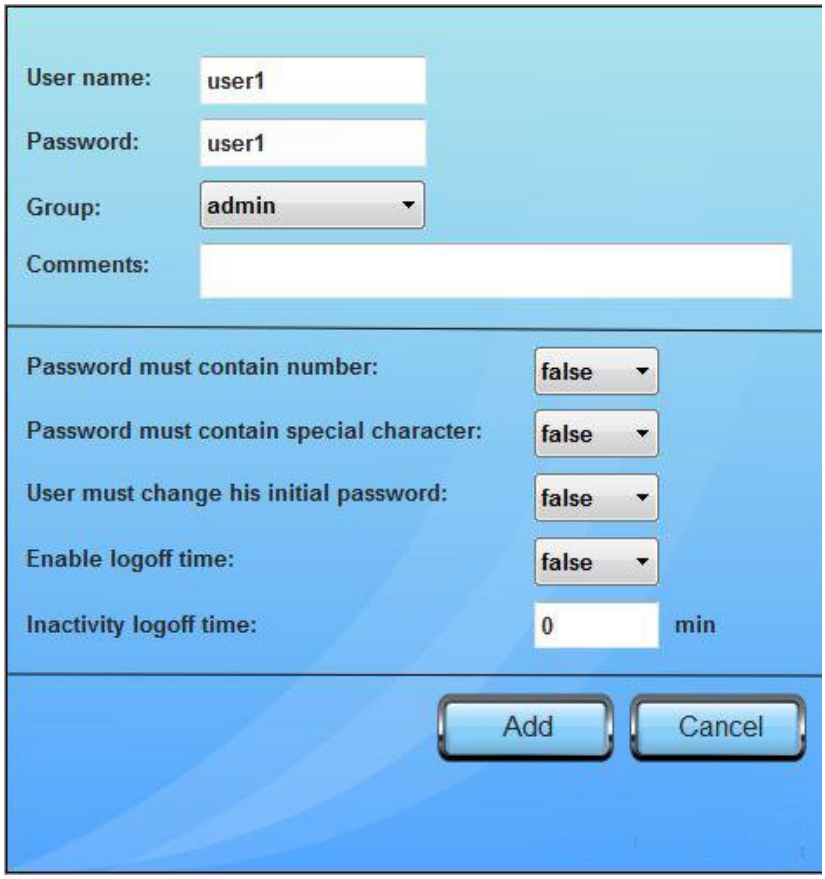

Figure 115

7.9.14 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 below figure.

Figure 116

7.9.15 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 below figure.

Figure 117

7.9.16 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 VisuControl.

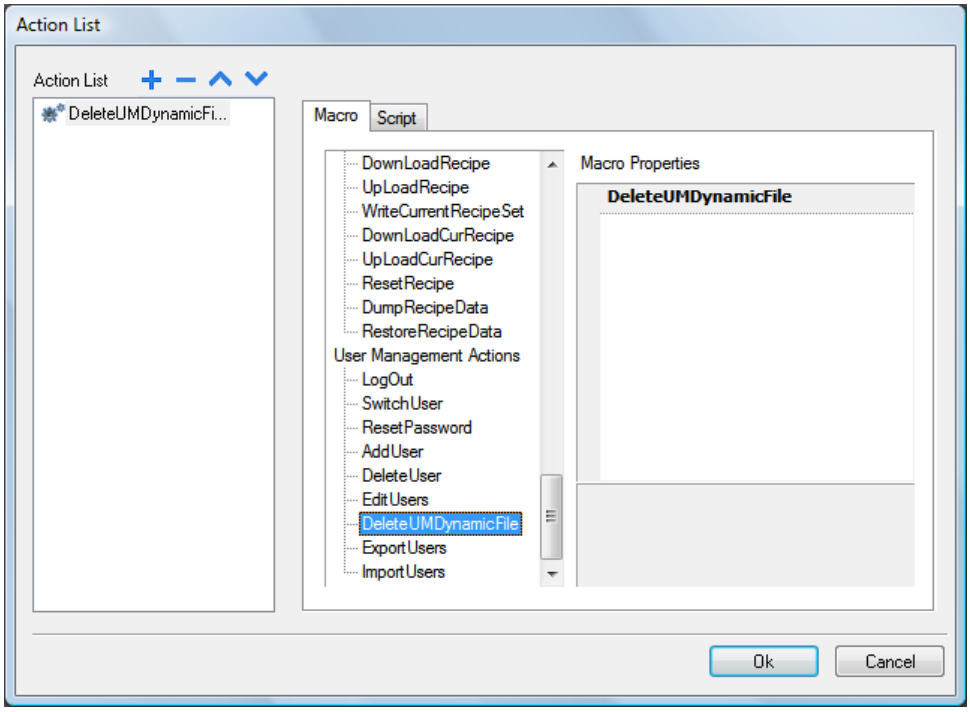

Figure 118

7.9.17 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).

Note: The external drives plugged on the USB port of the panel must be FAT or FAT32 formatted. NTFS format is not supported.

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).

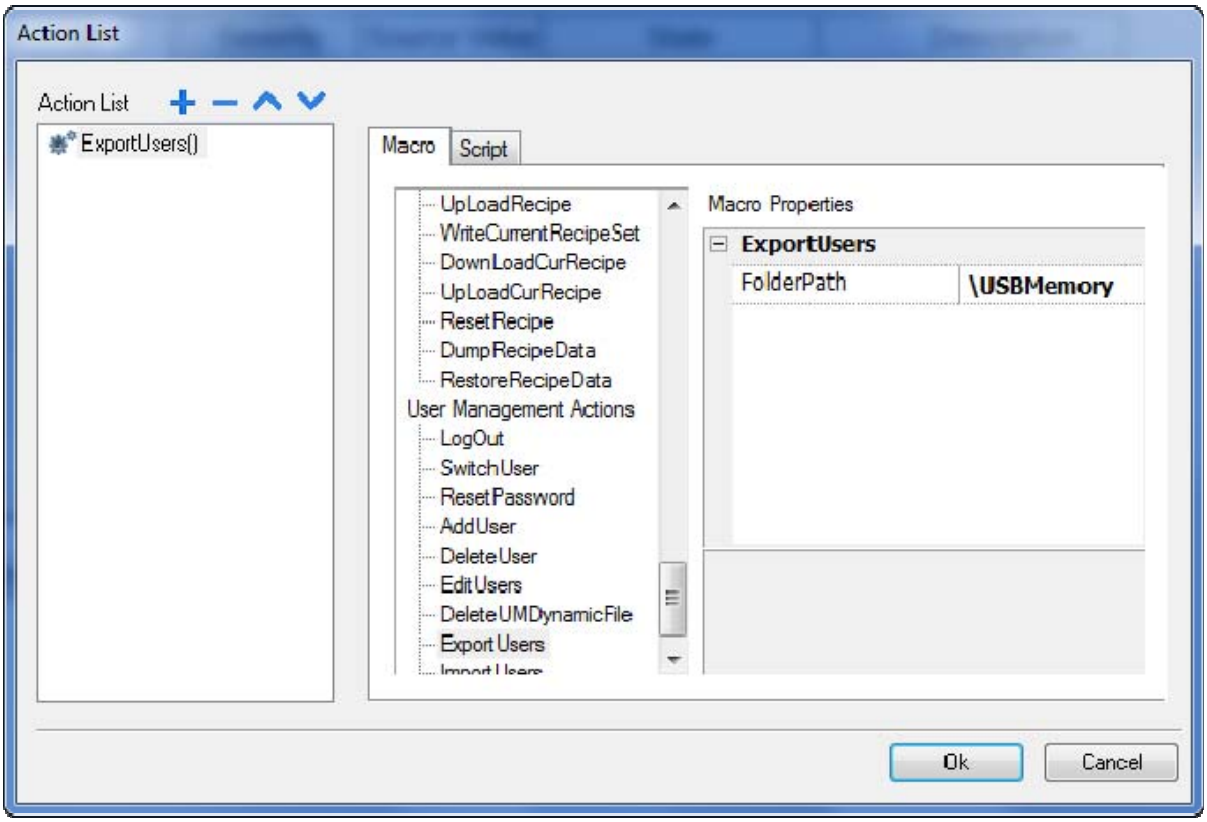

Figure 119

7.9.18 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 into the USB port, the path will be “\USBMemory", followed by the specified folder in the memory (or left empty for root folder).

Note: The external drives plugged on the USB port of the panel must be FAT or FAT32 formatted. NTFS format is not supported.

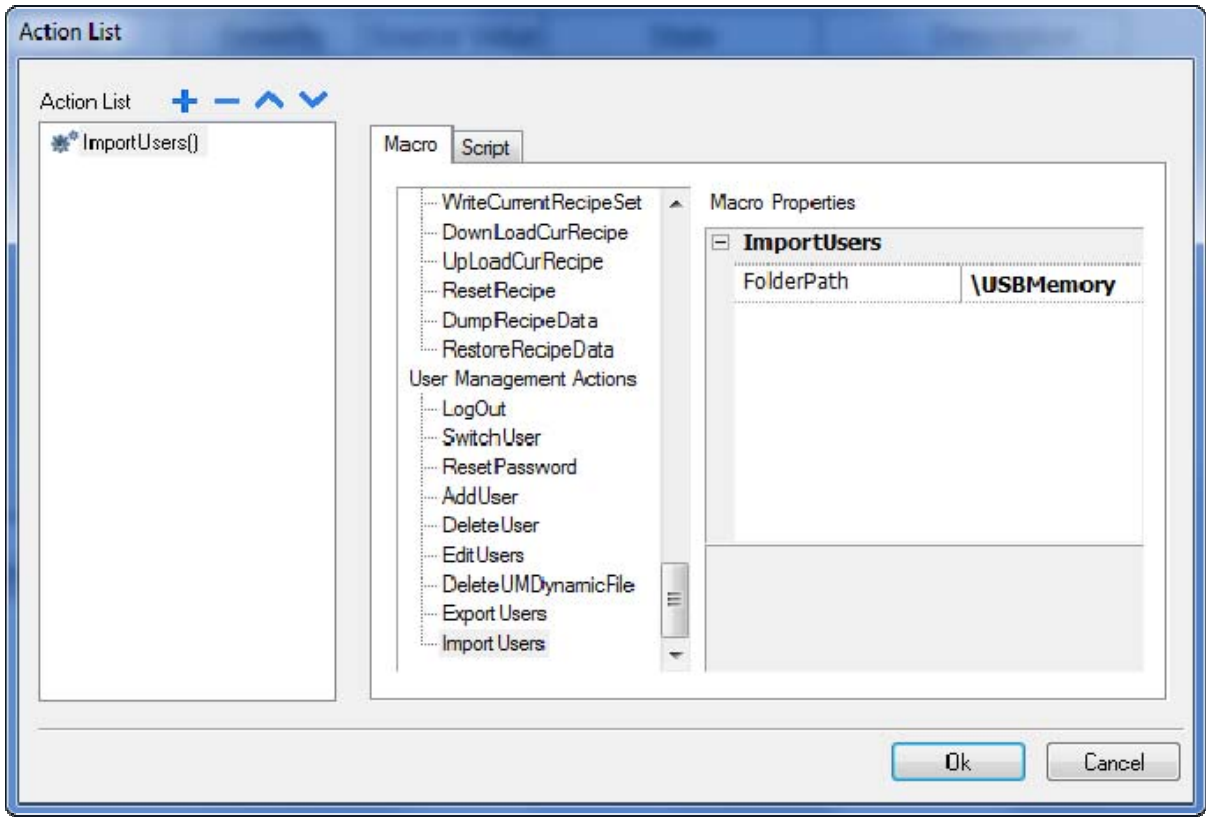

Figure 120

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 VisuControl, 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 status, a stipulation can be made requiring the plant operator to acknowledge the Alarm.

The Alarms are configured in the alarm manager and, thus, are a component of all screens of a project. More than one Alarm can be displayed simultaneously in the alarm widget, depending on its configured size. An Event can trigger the closing and reopening of the Alarm window.

Please note that, in VisuControl, 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.

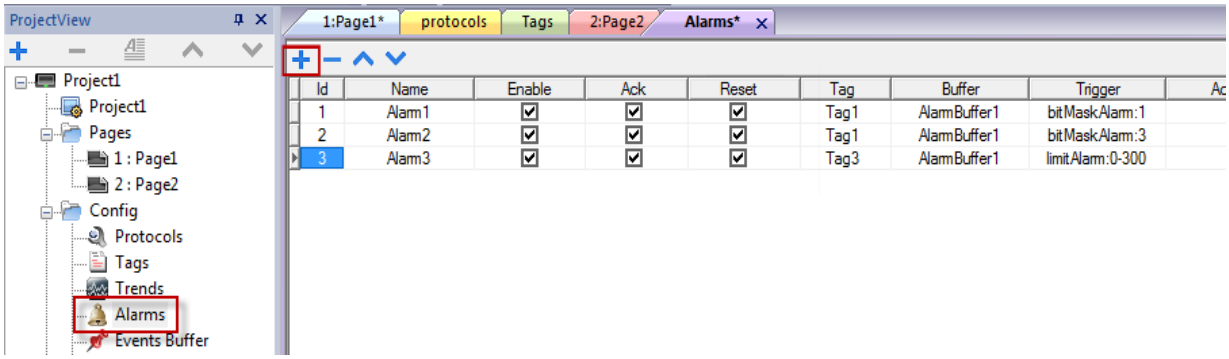

Figure 121

8.2 Name

Specifies the name of the 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 Enable / Disable Alarms in Run-time).

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, before it will be cleared from the Active alarm widget.

8.5 Reset

The Alarms' editor includes an column called "Reset". This check box, specific to each alarm, works in conjunction with the acknowledge check box. After an alarm requiring acknowledgment has been acknowledged, it will be cleared from the alarm list. If the reset check box is checked, the alarm will continue to be listed in the alarm list, as “Not Triggered Acked”, until the Reset button present in the alarm widget is pressed.

8.6 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.)

8.7 Buffer

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

8.8 Trigger

This selection determines the triggering condition for an alarm.
Three Alarm types are available:

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.

Bitmask Alarm — To get a valid trigger, the bitwise AND operator compares each bit of the bitmask with the Tag value corresponding to that Alarm. If both bits are on, the alarm is set to true. 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 decimal format; if more Bits are specified, each position must be separated by a ",".

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

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.

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

8.9 Alarms' State Machine

The HMI system implements an alarm State Machine which is described by the following figure. The graph includes states and transitions between them according to the selected options and desired behavior.

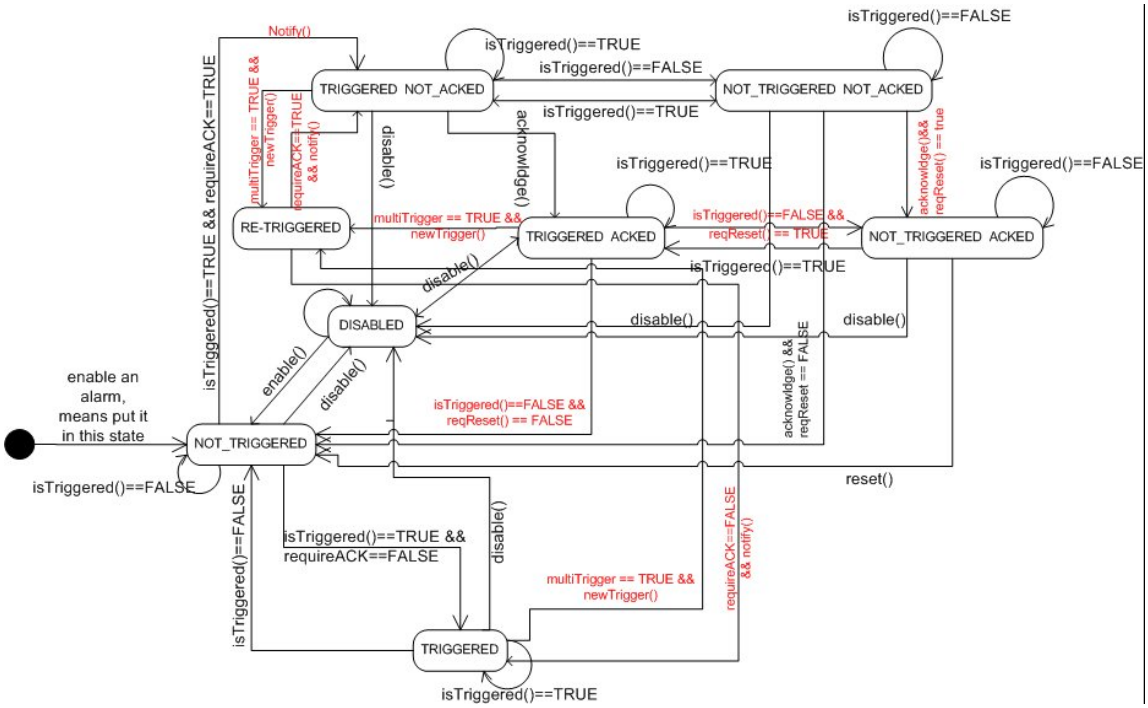

Figure 122

8.10 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 Action Enable).

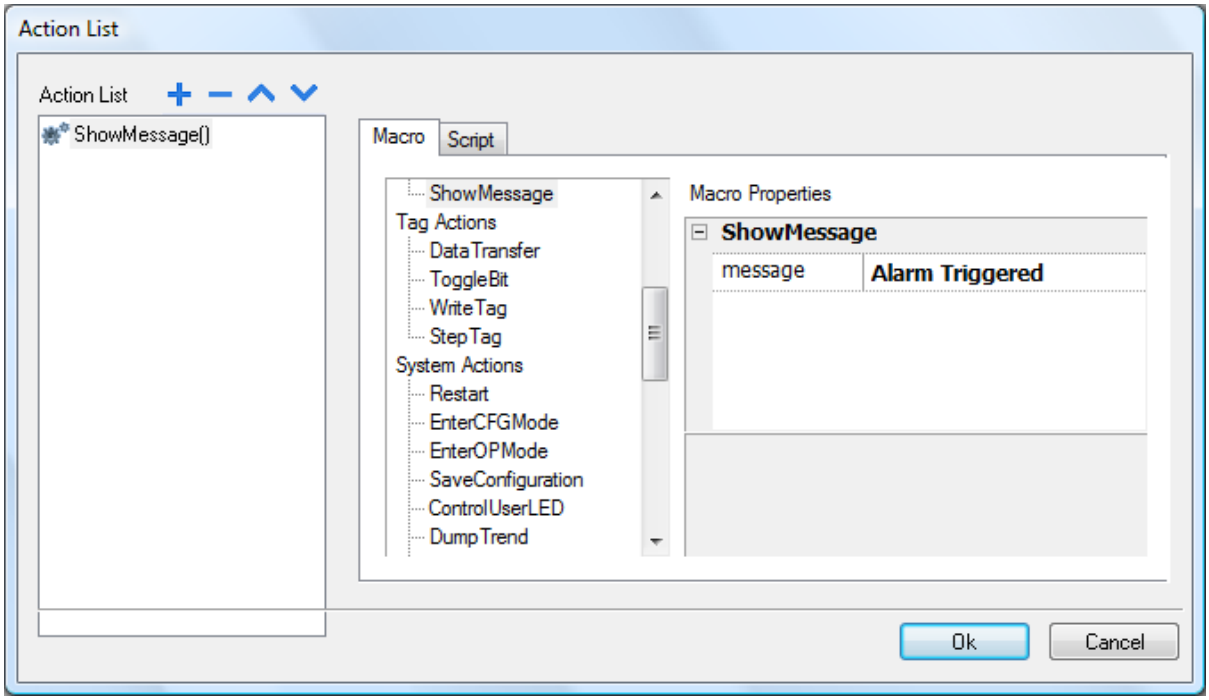

Figure 123

8.11 Description

This is the description for the alarm condition. The Alarm description is normally a text; this text supports the multiple language features. The text can be a combination between static and dynamic parts, where the dynamic ones are eventually some tag values. Please see the chapter “Live Data in Alarm Description” for further information about this feature.

8.12 Foreground and Background Colors

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

8.13 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.

8.14 Event 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.

8.14.1 Log Events

Select the "Log" tab in the dialog box (as shown in above figure). 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.

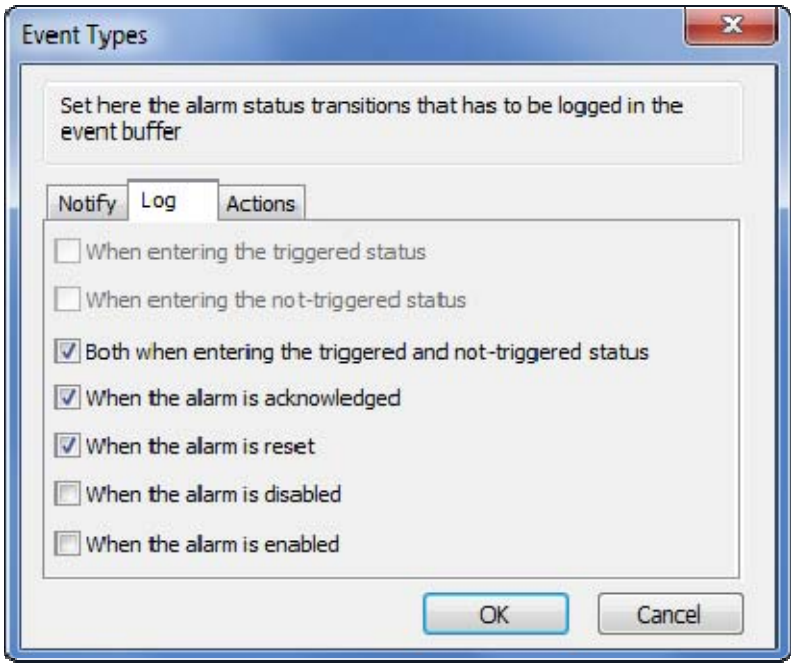

Figure 124

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 in below figure). Here there is an Option of Selecting the storage type. There are three Options: Flash , USBMemory, SD Card. For SD Card and USBMemory, a specific path has to be provided where the files have to be stored.

Note: An external USB pen drive is referenced by the operating system as "USBMemory"; an SD card as "Storage Card".

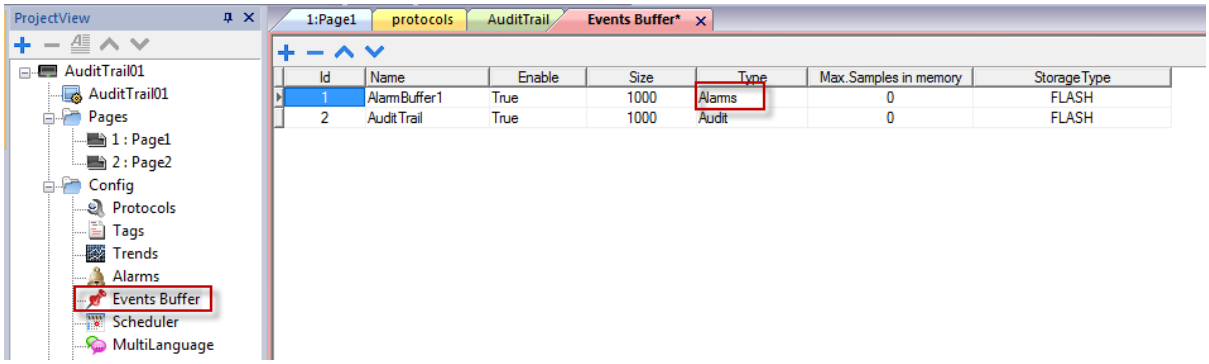

Figure 125

When the events' buffers are stored in the internal flash disk, the systems provides to save the file on disk every 5 minutes. The Event Buffer editor provides per each buffer an option called "**Max Samples in Memory**". This parameter allows a certain degree of freedom in the control of the number of events retained in RAM before saving them to the file on disk. With this parameter it is possible in fact to decide exactly after how many events the data is written on disk. The parameter can be used ONLY when an external disk storage is selected: When the internal Flash disk is used, the parameter is automatically force to "0" (not used).

Note: if the application requires fast saving of events, we suggest to use an external storage device with adequate support for frequent write operations. It is strongly recommended to avoid fast writing to internal flash disk.

8.14.2 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.

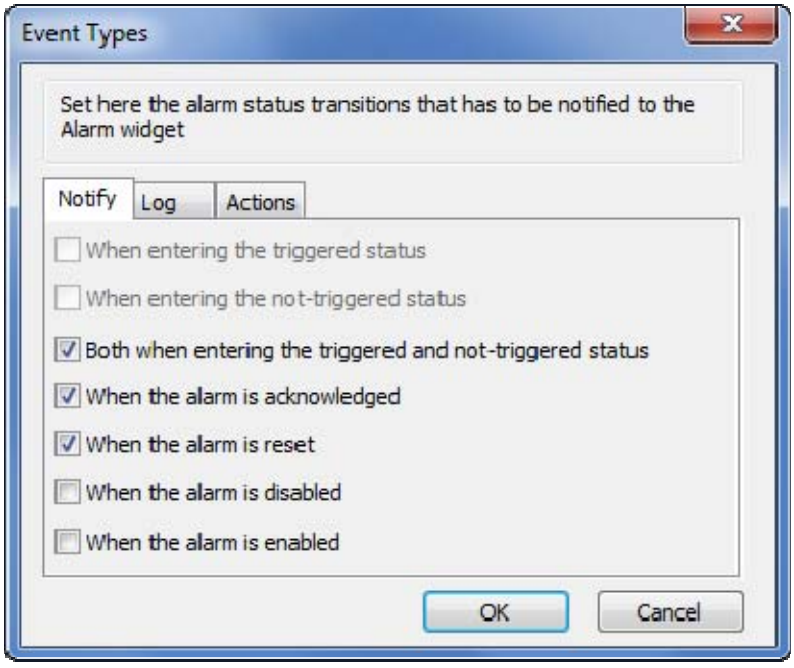

Figure 126

8.14.3 Action Enable

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

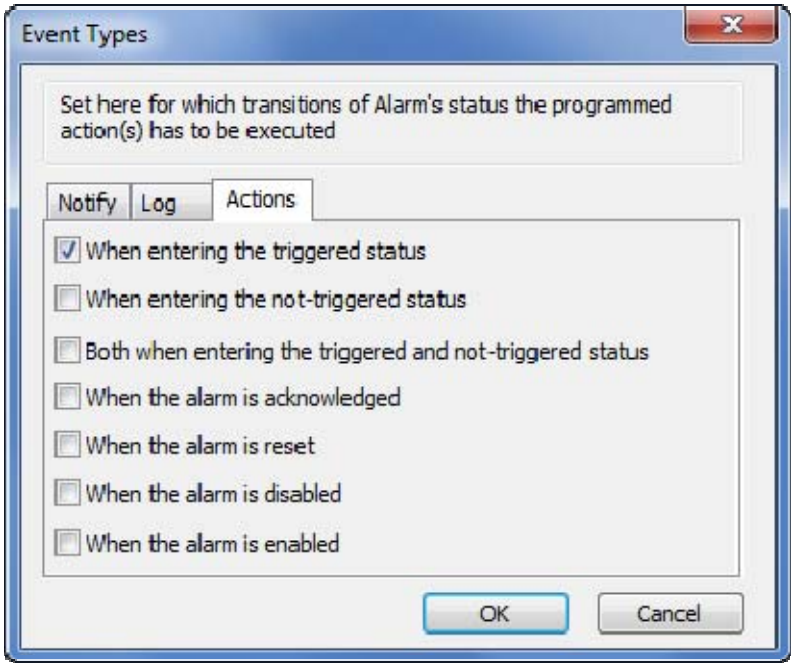

Figure 127

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.

8.15 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.

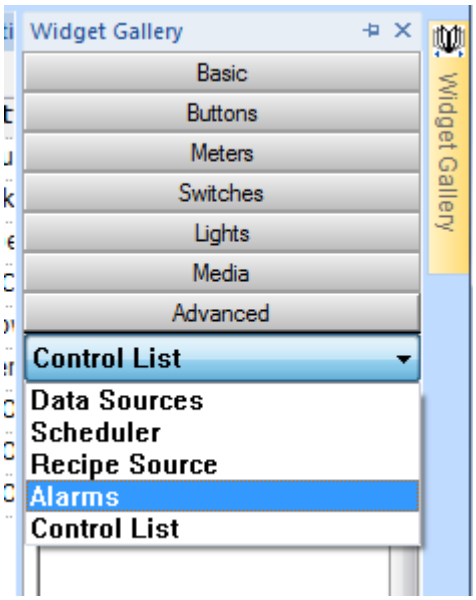

Figure 128

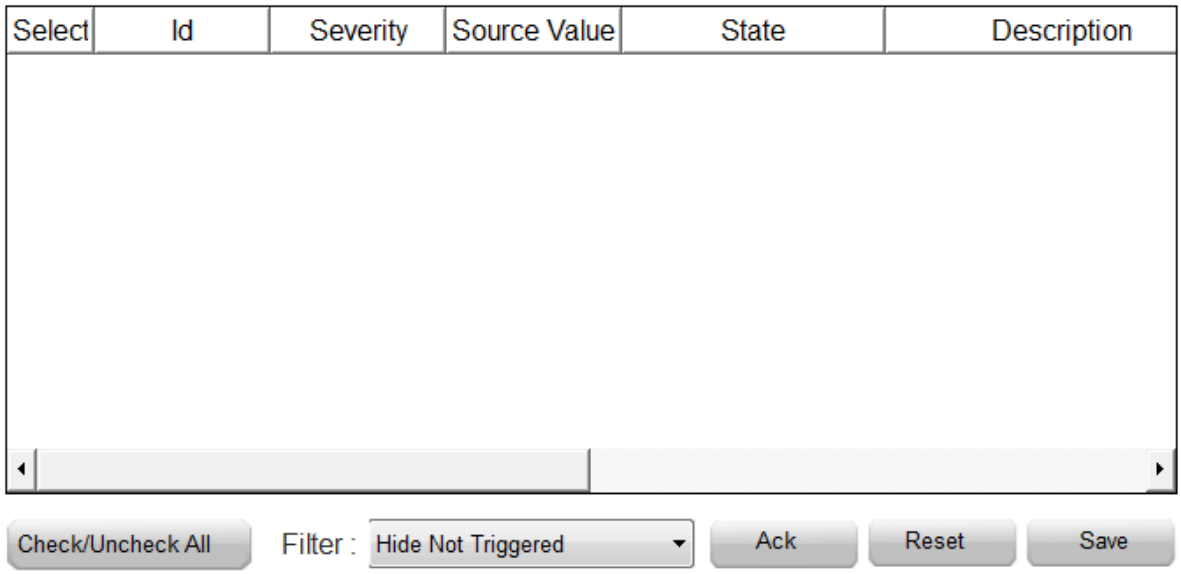

Figure 129

The Alarm Widget will display the Alarms in Runtime.

8.16 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.

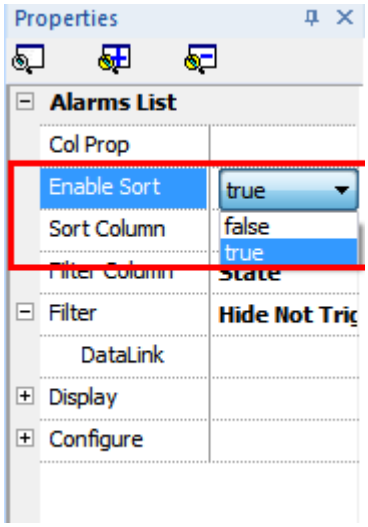

Figure 130

8.17 Configure Alarms History Widget

VisuControl 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.

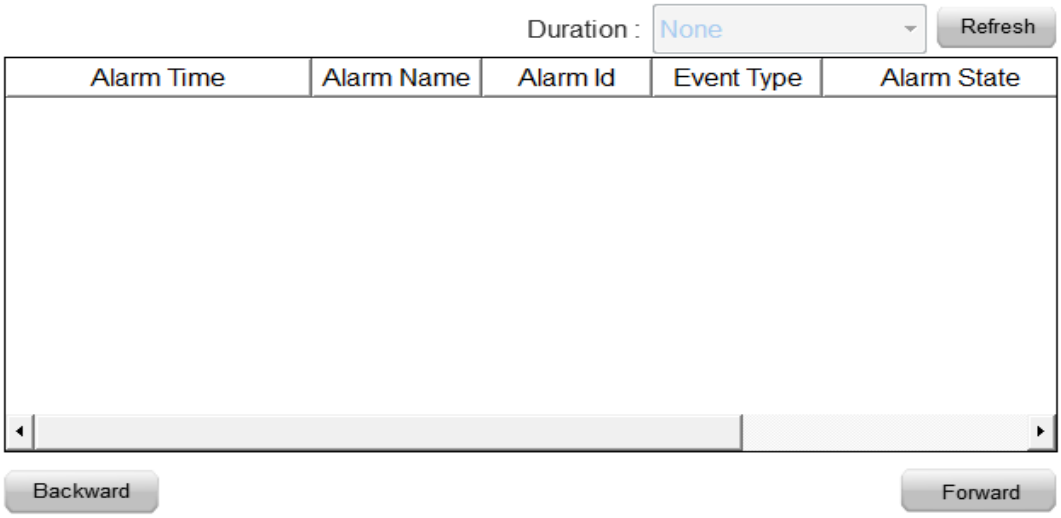

Figure 131

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

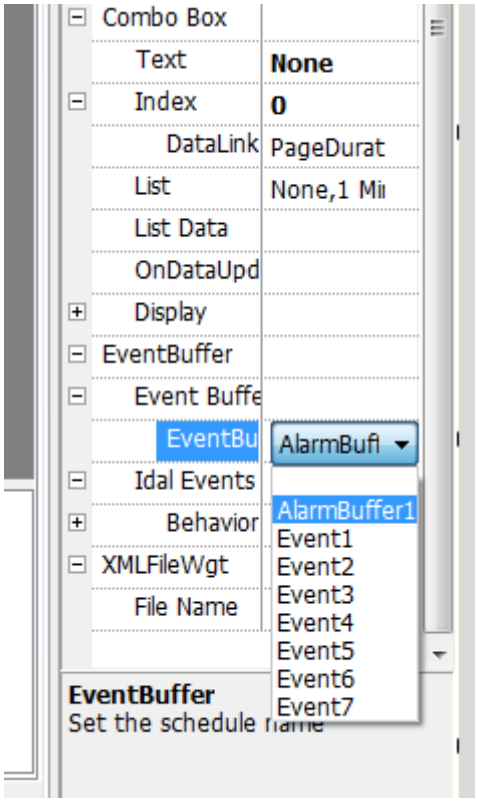

Figure 132

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

8.18 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 on a certain page, when an alarm is active, you must add a dedicated action that will go the page showing the Alarm widget.

8.19 Enable / Disable Alarms at Runtime

You can enable or disable the alarms at 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.

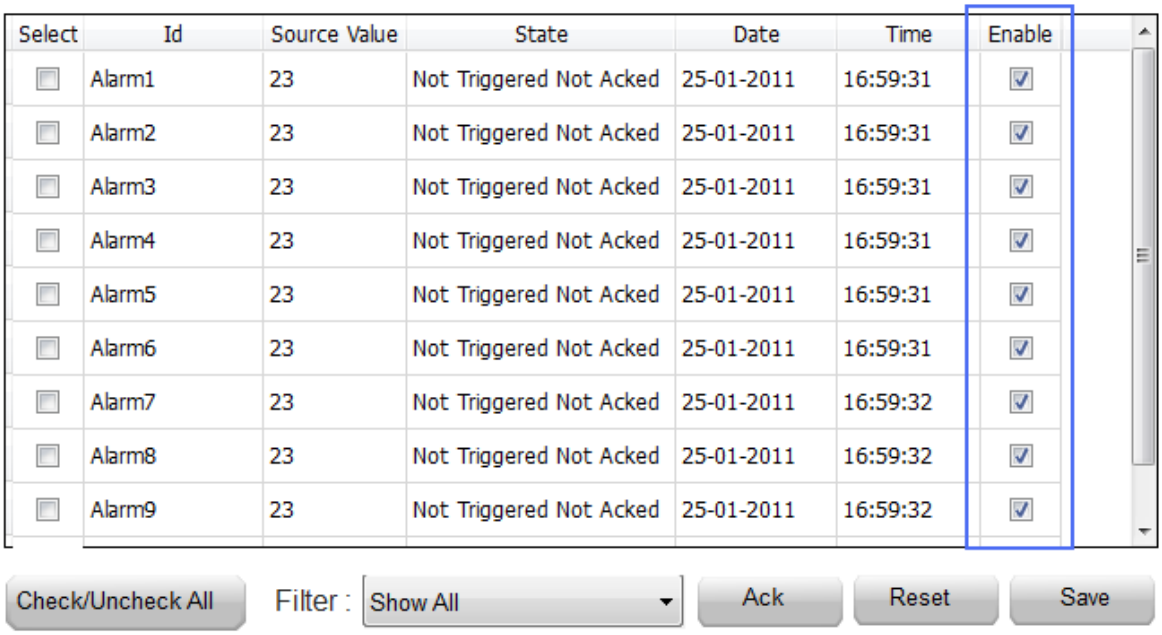

Figure 133

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.

8.20 Live Data in Alarms Widget

This feature is used to view the live Tag data value inside the alarm description. It is applicable for both Active Alarms and History Alarms widget.

To configure the live data visualization in the Alarm Widget, follow a simple syntax rule.
The Tags to be included must be specified in the alarm description string, including the Tag names in square brackets:

[Tag name]

An example is shown below.

Id	Name	Enable	Ack	Reset	Tag	Buffer	Trigger	Action	Description
1	Alarm1	☑	☑	☑	Tag1	AlarmBuffer1	bitMaskAlarm:	ShowDialog	Alarm 1 Tag Value is [Tag1]
2	Alarm2	☑	☑	☑	Tag1	AlarmBuffer1	bitMaskAlarm:1	ShowDialog	Alarm 2 Tag Value is [Tag2]
3	Alarm3	☑	☑	☑	Tag1	AlarmBuffer1	bitMaskAlarm:1	ShowDialog	Alarm 3 Tag Value is [Tag3]

Figure 134

During Runtime in the Alarm Widget, the markers and Tag name will be replaced, in the description column, by the actual value of the Tag. The Widget automatically refreshes and shows the current values of the Tags in the Widget.

In History Alarm Widget it will show the value of the Tag at the moment the alarm was triggered.
Into the CSV file resulting from the Dump of the alarms events list, the Tag values can be seen in the description column.

Output will be displayed as shown in figure below.

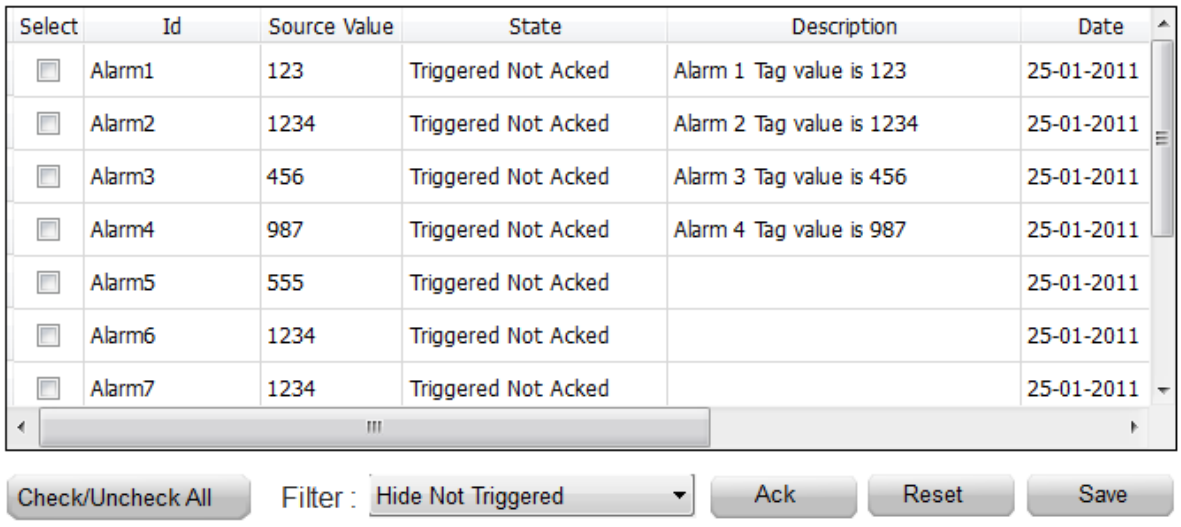

Figure 135

8.21 Exporting Alarm buffers as CSV file

The historical alarm list (the event buffer) can be exported using the specific action called “DumpEventArchive”. The action is described in the chapter called “Dump Event Archive”. Please refer to that chapter for information about how to export the data.

Note: the tag values included in the Alarms’ description are also included in the event description stored in the event buffer; the tags are sampled at the moment in which the alarm is triggered and that is the value recorded and included in the description; in the Alarm description, displayed by the Alarm widget, the value may change because it is constantly updated, but no additional values are recorded.

9 Working with Recipes

Data can be stored 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 a 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 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.

Currently, the Recipe data is configured in VisuControl, the user can specify default values for each element of the data records. On Runtime, data can be edited; the new data is saved to a new and separated data file, different from the original one containing the default values. Any change to recipe data is file on disk. 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.

Note: the reset recipe data to default values there is a dedicated action called “Reset recipes”; see below in this chapter for further information.

The User can also select where the Recipe needs to be stored. There are three options for this: FLASH, USB, SD CARD. User can select any one.

You can configure Recipes by adding the required controller data items to a page from the 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 VisuControl 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. The newly added Recipe item will be added in the project workspace.

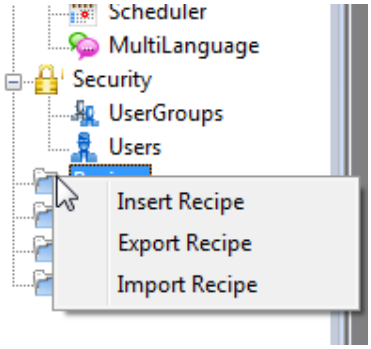

Figure 136

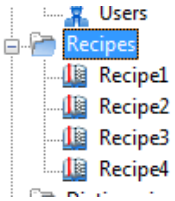

Figure 137

Double Click on the Recipes to open the Recipe Editor, as shown in below figure.
Add the recipe elements by clicking the “+" button, and then link the tags to the recipe element.

By clicking on the button for “Storage Type” you can select where to store recipe data. See the figure below

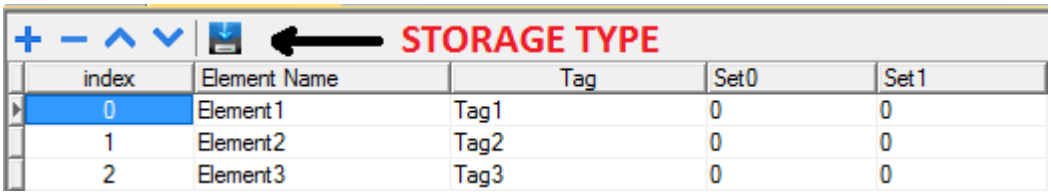

Figure 138

A dialog in which the selection can be made will open. See the figure below.
For USB and SD card you must provide the folder location path.

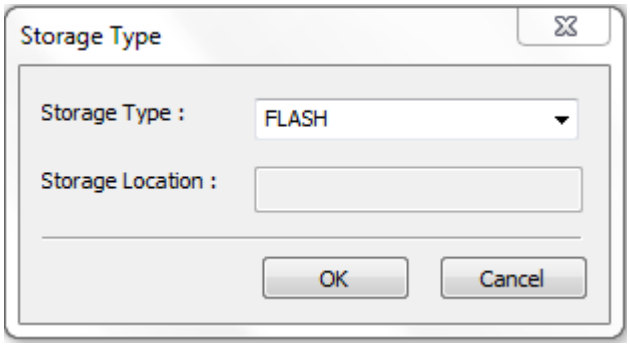

Figure 139

Note: an external USB pen drive is referenced by e operating system as “USBMemory”; an SD card as “Storage Card”.

9.2 Configuring Recipes Set on the Page

The number of parameter sets can be changed in the Number of sets field in the property pane. From there you can also change the name of each Recipe set.
Recipe values for all the parameter sets can be entered into the Recipe Editor window.

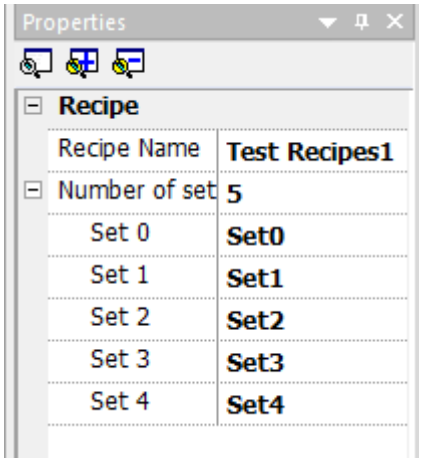

Figure 140

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. Below figure shows an example of a Tag attached to a Recipe field.

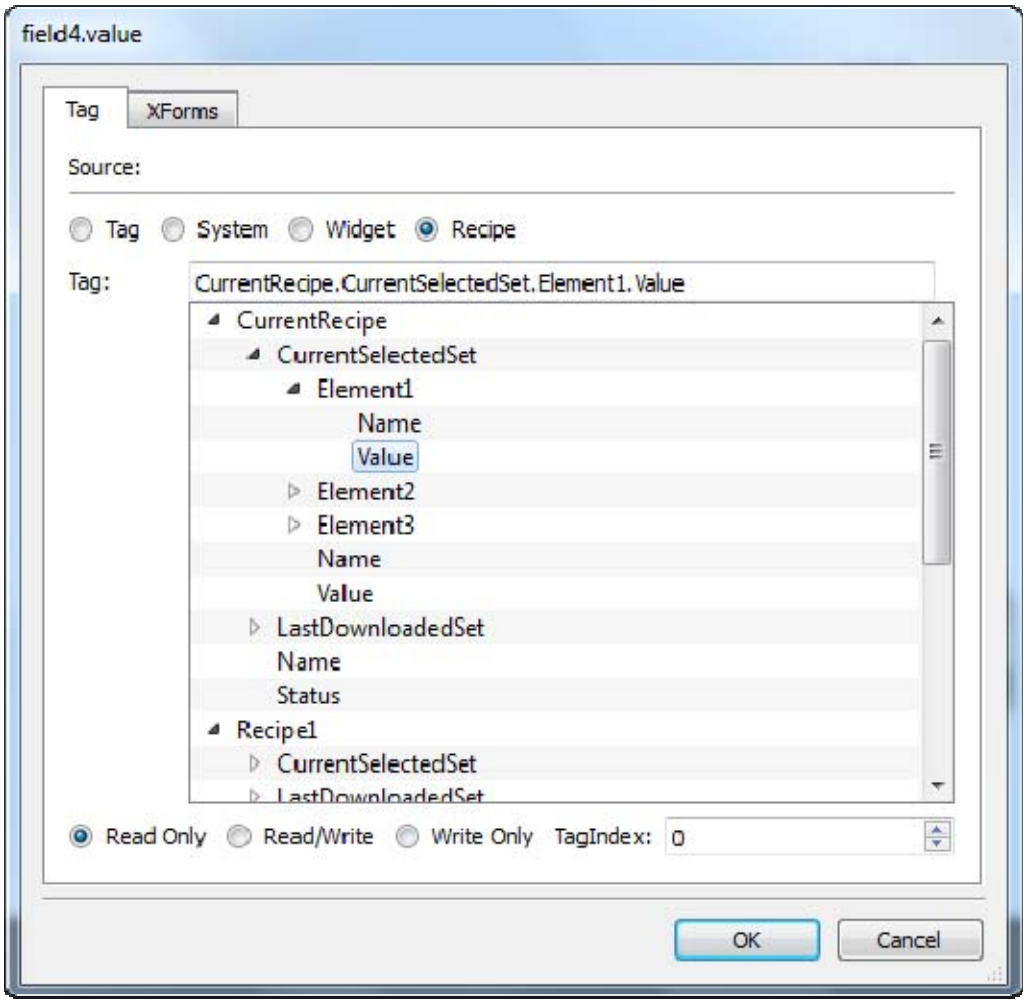

Figure 141

The Attach to Dialog allows you to attach to the numeric field 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 fields are defined as Read/Write, 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 142

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 actions’ list.

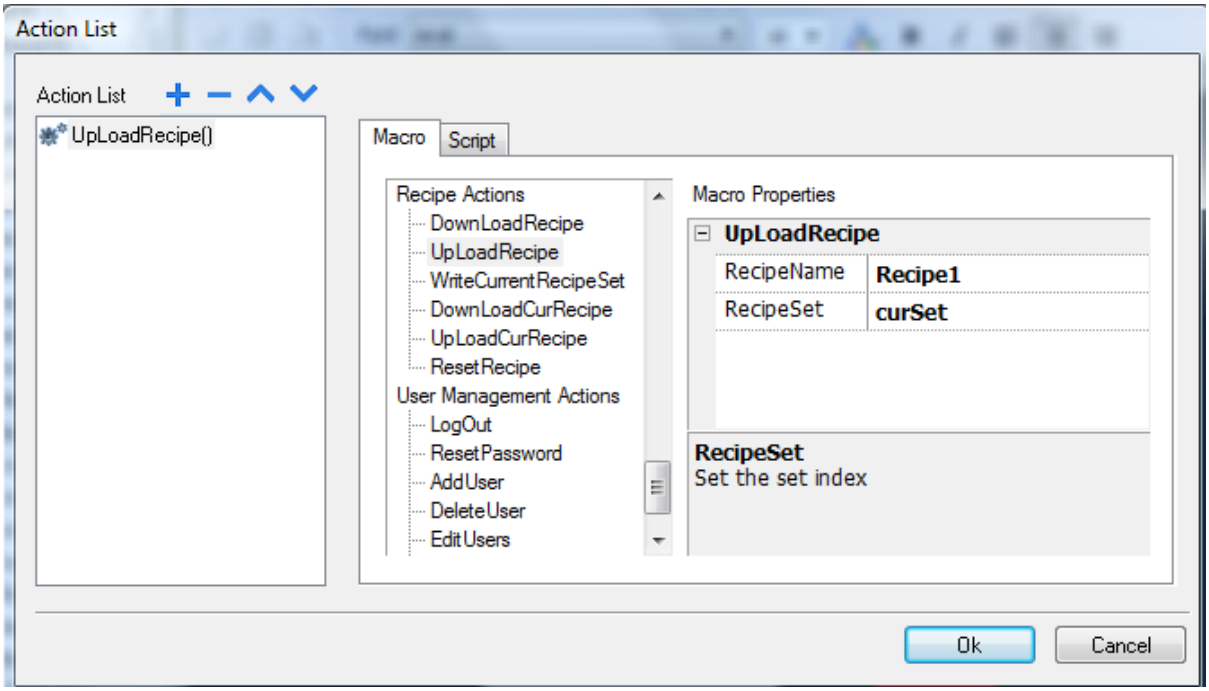

Figure 143

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 setting: recipe values as originally set in 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 " Configuring Recipe Widget for Runtime Execution") 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 "Configure Recipe Transfer Macros". 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. Below figure shows an example of the Recipe project.

Figure 144

9.7.3 Backup and Restore of recipes data

The Recipe data stored on the HMI device can be exported for backup purposes and later restored to the system. Please refer to the chapter "Dump Recipe Data"and "Restore Recipe Data" for further information.

10 Working with Trends in VisuControl

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 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.

To configure the RealTime Trend, just drag and drop the RealTime Trend Widget from the Basic > Trends category of the gallery.

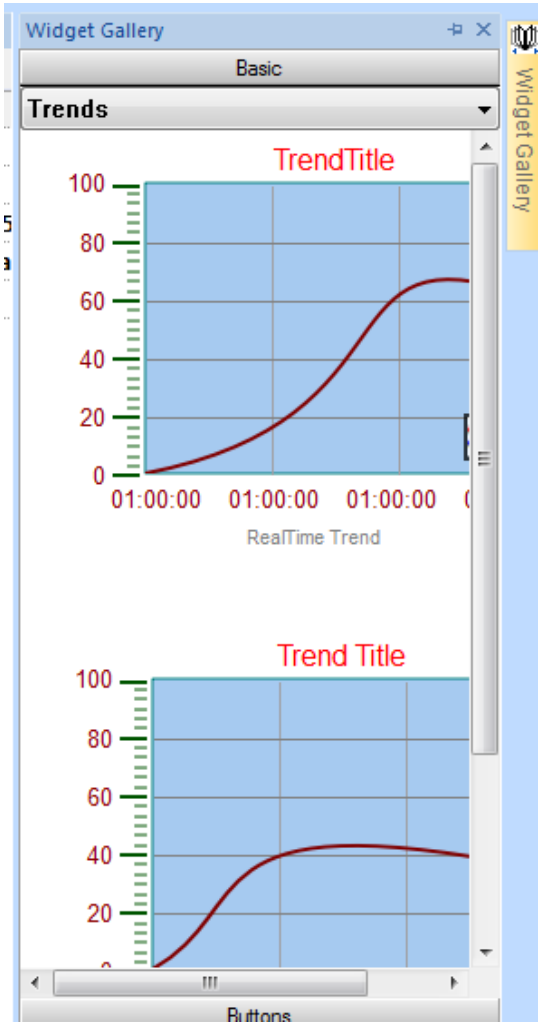

Figure 145

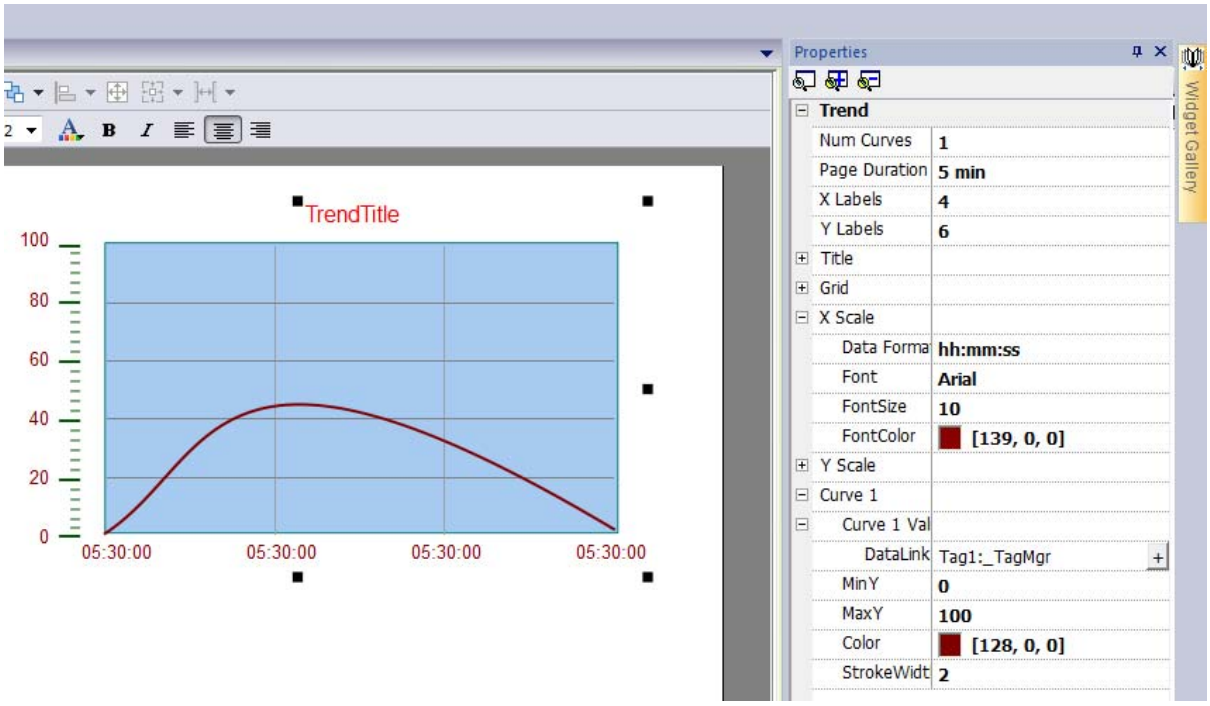

Figure 146

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

Number of curves (Num Curves) This property allows you to configure the number of trend curves in the Trend window. A maximum of 5 curves can be configured in a Trend window.

Page duration Here you can set the time range of the X-Axis. However, you can dynamically change the page duration in Runtime with the Date Time combo widgets, thereby attaching to the Trend window page duration properties.

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

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

Title These properties allow you to modify the Trend title and font properties (like font size, label, etc.)

Curve x These properties allows you define the Tag or the Trend buffer that will be plotted into the trend window.
Scaling can be applied to the Tag values. To apply scaling, use the X Forms attached to dialog.
You can set the Minimum or Maximum of the curves. You can also attach a Tag to these minimum and maximum properties. This enhances the ability 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.

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.

The first step in creating a History Trend is to create a Trend Buffer.
The purpose of the trend buffer is to save a sequence of values of a specified Tag in order to record the state of the tag while time changes. Once values are stored in the buffer a dedicated widget, called History Trend viewer, can be used to represent the curve in a graphical format.
The History Trend viewer is available in the widget gallery under the Basic>Trends category as shown by the following figure.

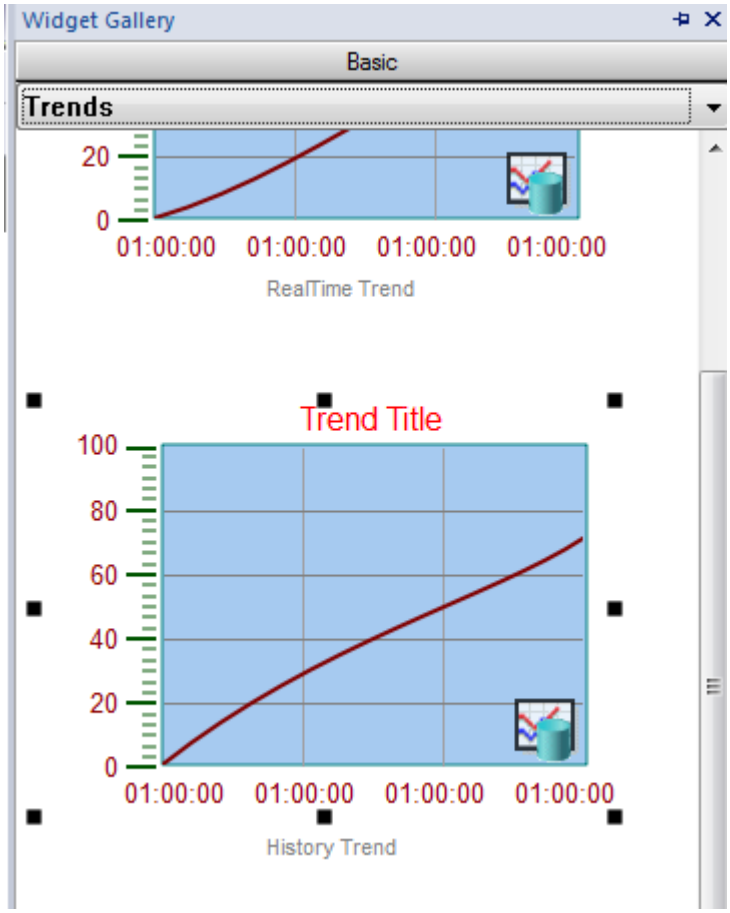

Figure 147

In the History Trend widget the start time of the Trend window will be the current time and stop time will be the current time + duration of the window.
The plot starts from the right end of the Trend window as in the figure below. The graph will be automatically refreshed during a certain interval of time, till the stop time. When the curve reaches the stop time then the graph will scroll left and the update of the curve will continue until it reaches again the right side of the viewer; at that moment a new scroll is automatically done and the process repeats.

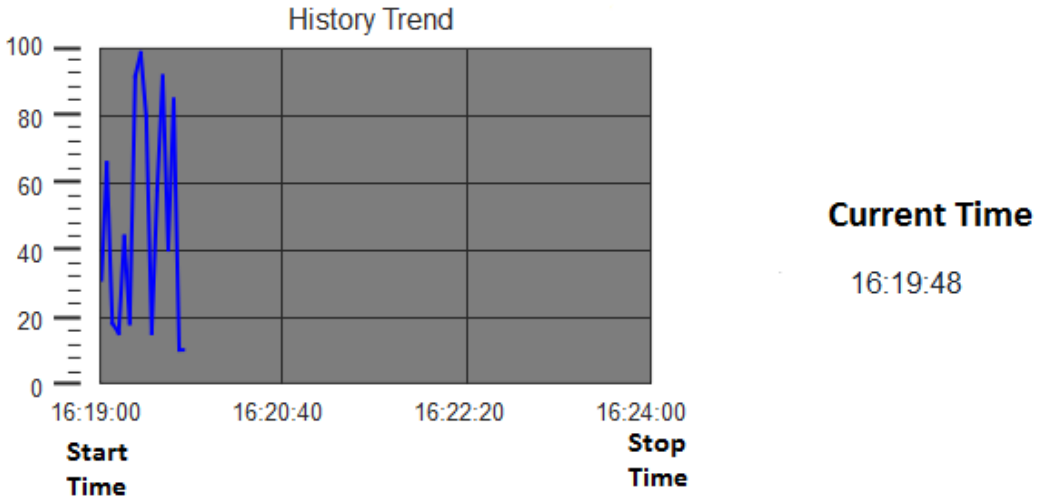

Figure 148

10.2.1 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 in the format of a data file. There is an option to store these data files on the Flash Disk, USB Memory or SD card. We can select where to store the data file for each trend buffer.

Note: an external USB pen drive is referenced by the operating system as "USBMemory"; an SD card as "Storage Card".

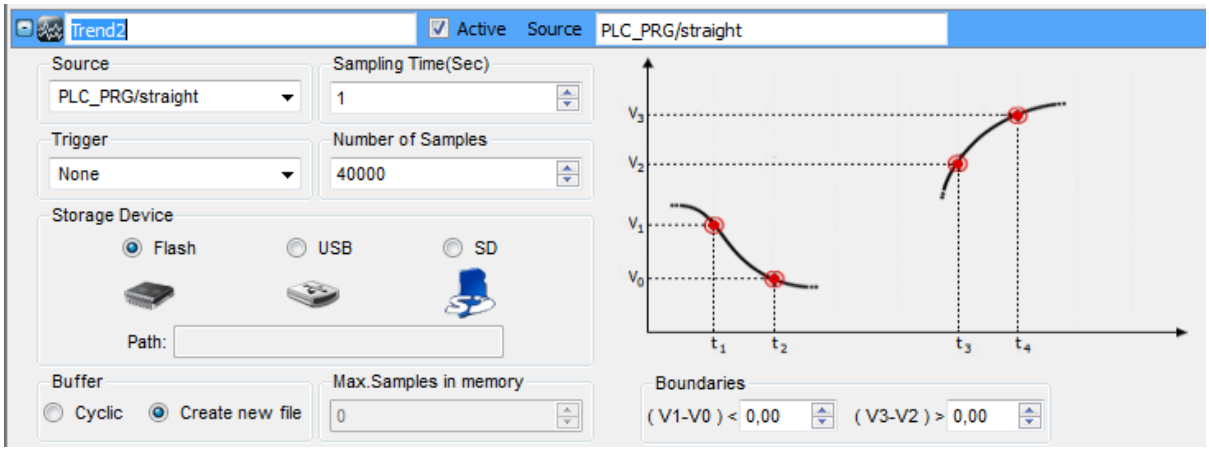

Figure 149

In the Project View pane, double click Trends to open the Trend Editor. Then add the trend buffer, by selecting the "+ Add" button on the editor. By clicking "+" near each trend buffer, the corresponding buffer configuration is expanded.

The “Total memory Space” bar shows how much memory has been used by the trend buffers configured so far. The max number of samples allowed for a project is 2400000. The memory use is the percentage of this number. As in the Figure above, suppose the total number of samples used in the project is 80000. Then the total memory used will be shown as 3%. This is calculated by the formula

$$Total\ Memory\ Space\ = \frac{Total\ Number\ of\ Samples\ used\ in\ the\ Project}{Max\ Number\ of\ Samples\ allowed\ for\ a\ Project} * 100$$

As we Increase the number of samples, the percentage of usage also increases and this will be shown in the bar.

Note: The maximum number of trend buffers per project is 30 with a total number of 2400000 samples. The maximum number of samples configurable for a single trend buffer is anyway limited to 200000.

The following are the properties of each trend buffer in the Trend Editor:

Trend Name This defines the trend buffer name, which will appear when you define the buffer to a trend window property pane. A default name is assigned by the system; the name can be modified by the user.

Active Specifies if the trend runs by default when system starts up.

Note: The trend buffers can not be activated during Runtime.

Source This combo list allows selecting 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 specifies the sampling period in seconds.

Trigger When the Trigger tag is specified, the source tag is not sampled on a cyclical basis but 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.

Number of Samples This represents the buffer size expressed in samples.

Storage Device This is an option to select where the trend buffer data file will be stored.Here three options are provided: Flash, USBMemory and SD card. For Storing in USBMemory or SD card. The exact path has to be given in the space provided.

Note: Please be sure that the USB pen drive or SD card has been properly inserted in the Panel before executing the Dump action.

Note: an external USB pen drive is referenced by the operating system as “USBMemory”; an SD card as “Storage Card”.

Buffer	This is an option to select how the buffer sampling has to be executed and how the data files have to be created. Two options are available: Cyclic and Create a new file... If Cyclic is selected then the trend data is organized in the form first in first out (FIFO). Once the buffer gets full, the oldest values will be erased to create space for storing the new values. If Create a new file is selected, when the buffer gets full, it stops storing the data and it will create a new file where the new trend buffer values will get stored. The older values will not get erased. The buffer values will be stored in different files.
Max Samples in Memory	This parameter allows a certain degree of freedom in the control of the number of samples retained in RAM before saving them to the file on disk. **When the selected storage disk is the internal flash memory, the data stored in RAM is flushed to disk every 5 minutes. This time is fixed and can not be changed.** When selecting an external memory storage device, the parameter becomes editable and specifies the number of samples retained in memory before saving to disk.
Note:	if the application requires fast saving, use an external storage device with adequate support for frequent write operations.
Boundaries	This specifies 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 based on a trigger tag value change, the boundaries are applied to the trigger tag value.

10.3 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 page, drag and drop the “History Trend" widget to the page.

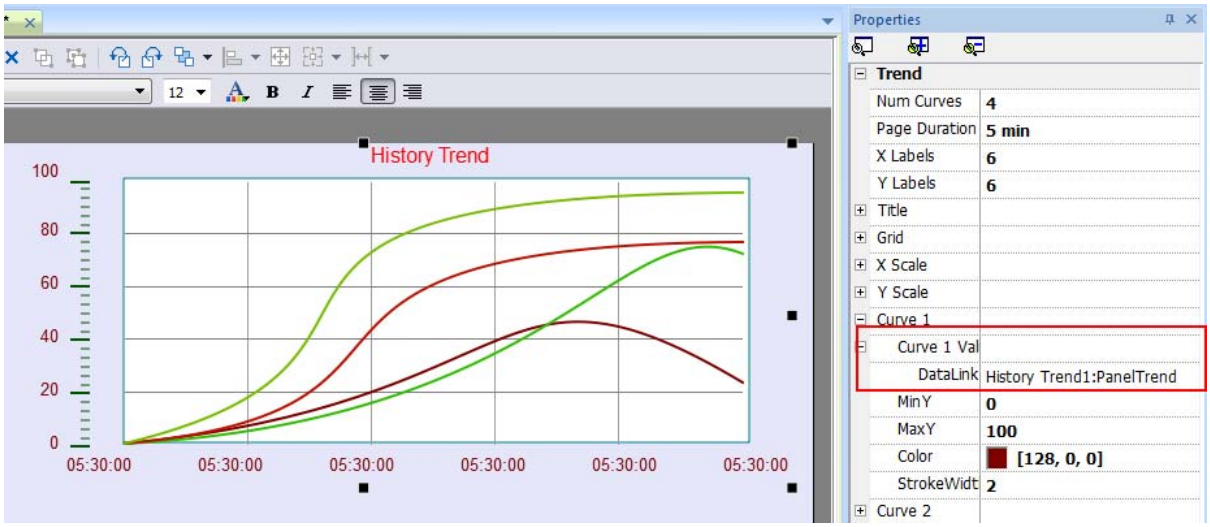

Figure 150

Then, in the property pane of the Trend window, attach the trend buffer to be plotted in the trend window (as shown in below figure).

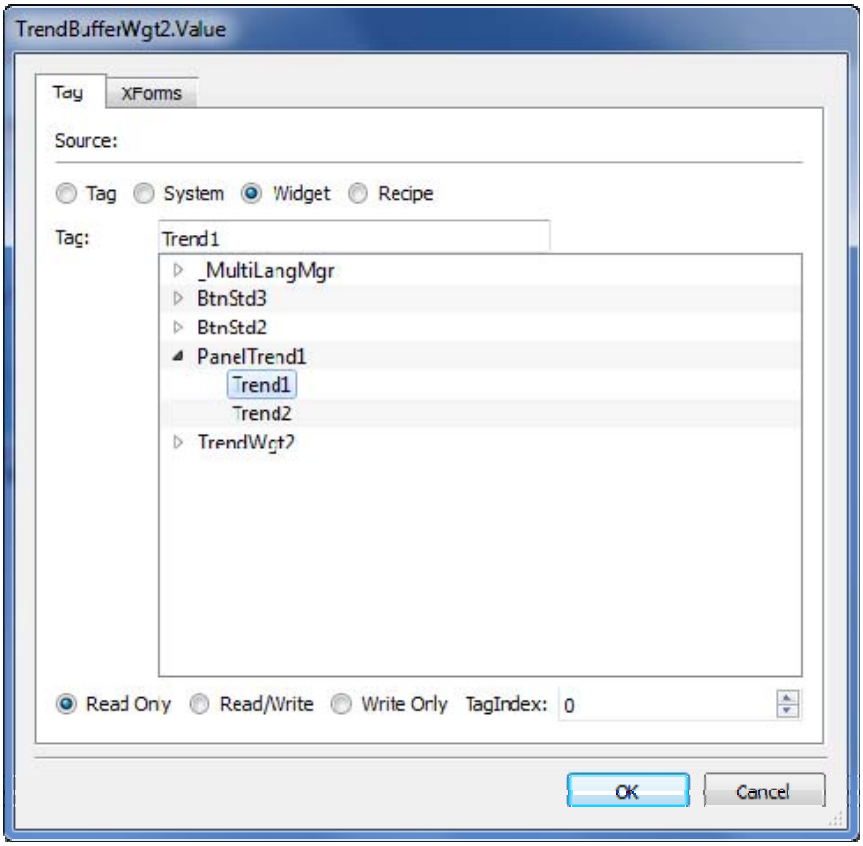

Figure 151

10.4 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, number of samples, etc.

In the “Curve x” category there is one property called “Request Samples” as shown in below figure.

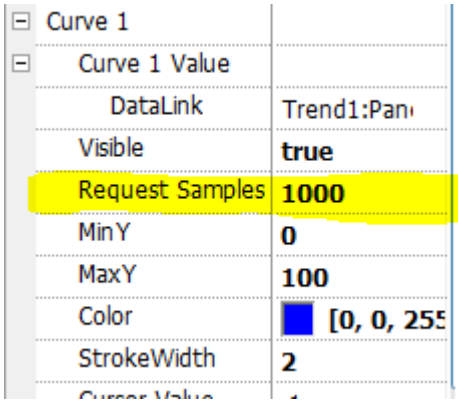

Figure 152

This property represents the maximum numbers of samples read by the widget from the buffer data file; this block size can be adjusted to fine tune performances in trend viewer refresh especially when working with remote clients; the default value is normally a good compromise for most cases.

10.5 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 to the get the value of the Curve at the particular instant in time.

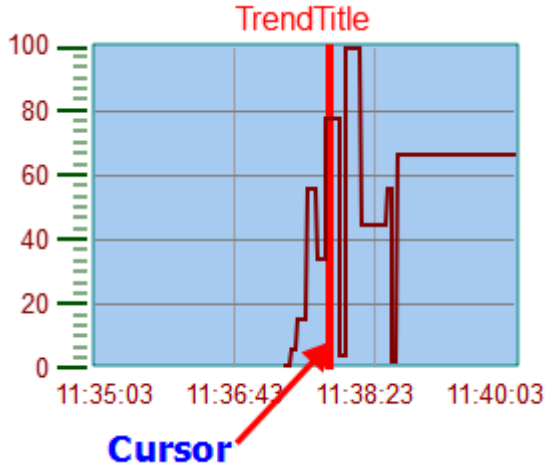

Figure 153

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 below figure). This is the Y axis Value of the Cursor.

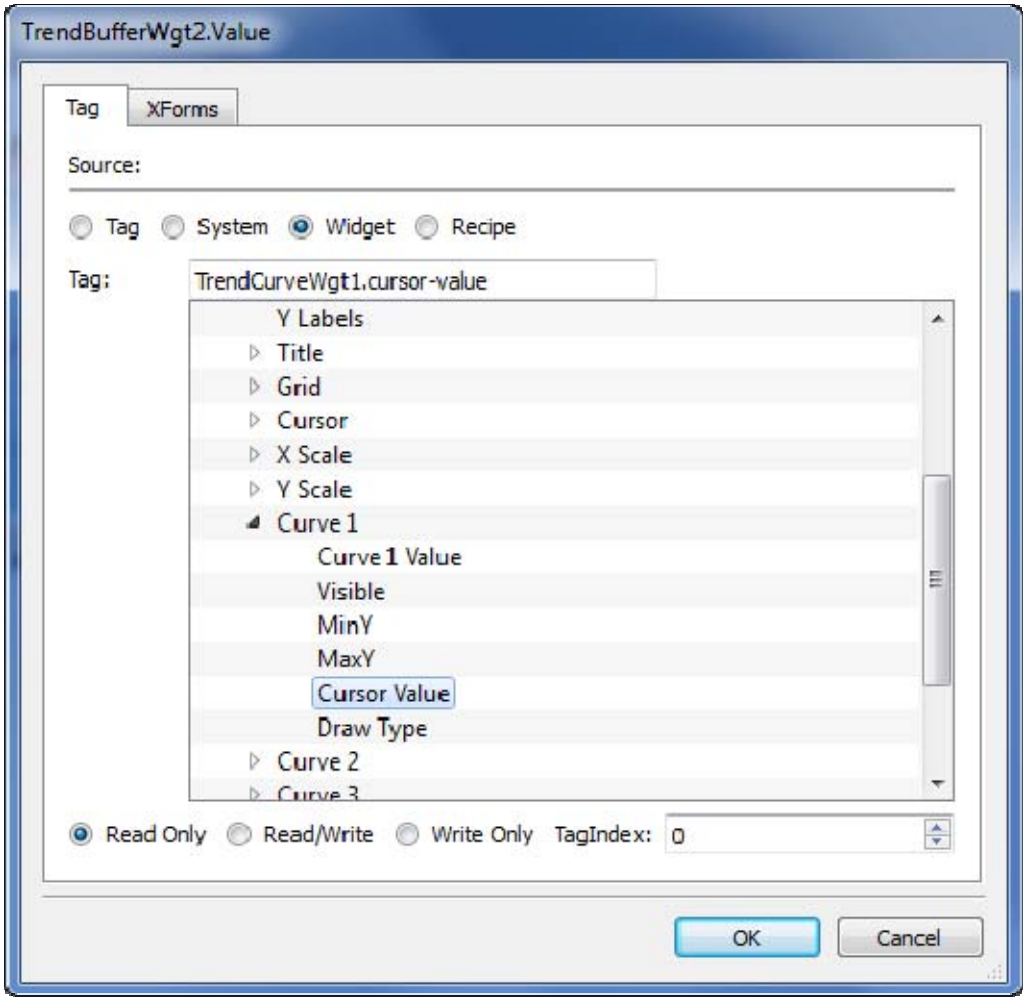

Figure 154

To get the Time at the Particular Point where the Cursor is placed, define a numeric field and attach to the “Widget Tag” as shown in the Figure Below.

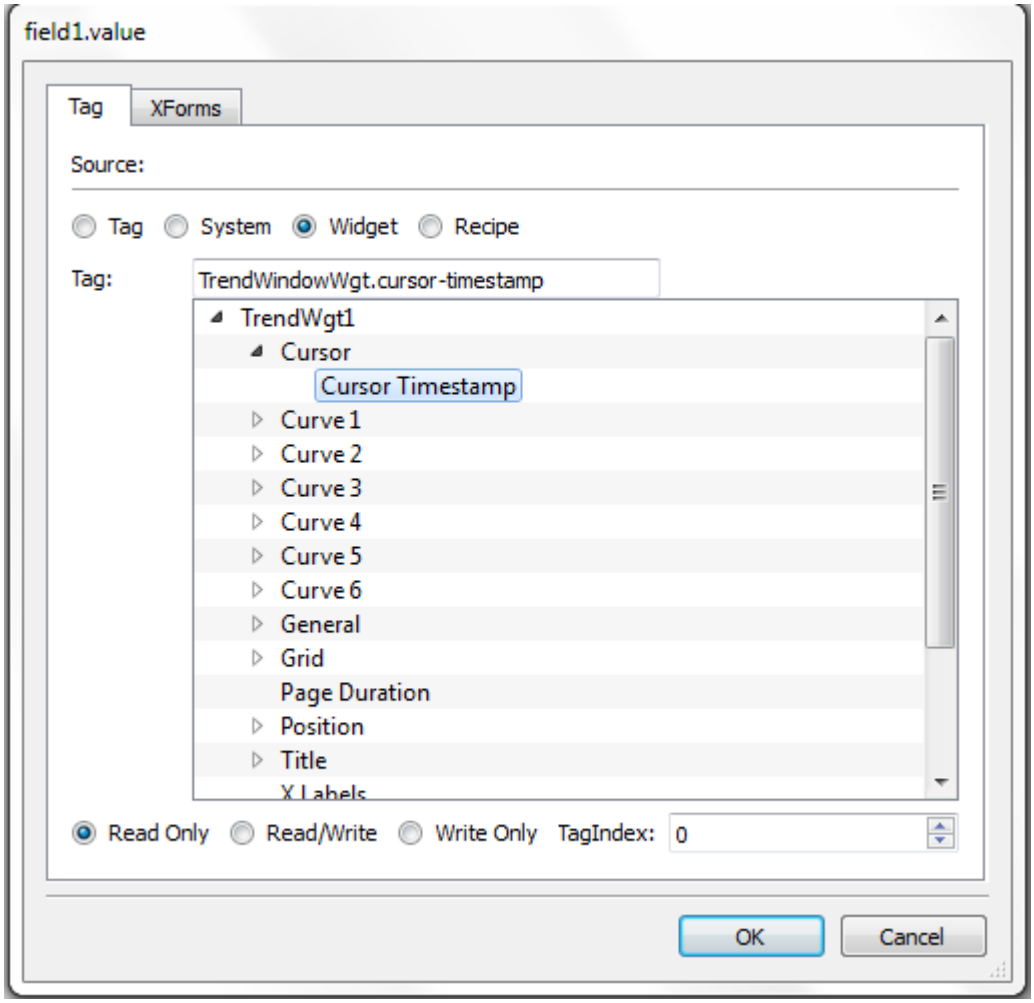

Figure 155

The Widget tag shown in the figure above represents the X axis cursor value for the trend window.

10.6 Exporting trend buffer data to CSV file

The trend buffers stored in the selected media can be exported to CSV file using dedicated actions. Please refer to the chapter “Dump Trend” for further information.

11 Working with Multi-Language in VisuControl

A true Multilanguage feature has been implemented in VisuControl 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. VisuControl also extends the TrueType Fonts (in short TTF) provided by Windows systems to provide different font faces associated with different character sets.

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

Note: In Windows XP operating systems, for the proper operation of the multiple language editor in the VisuControl, you will need to install the support for complex script and East Asian languages as shown in below figure.

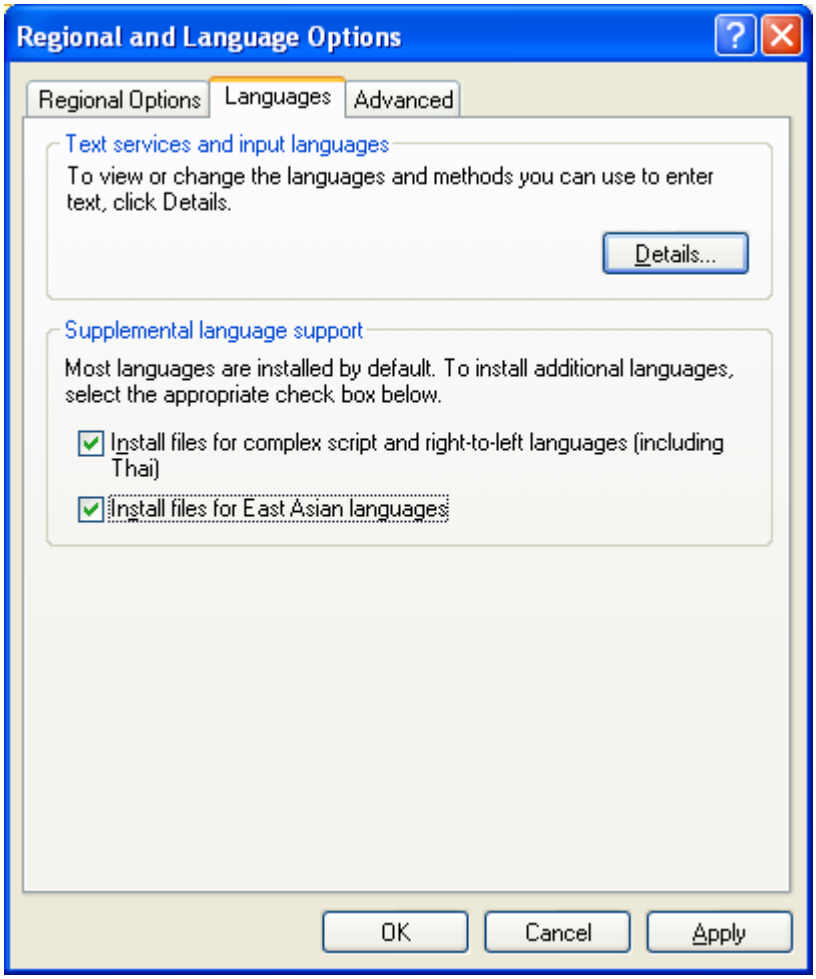

Figure 156

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

For Simplified Chinese, VisuControl 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, VisuControl 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.

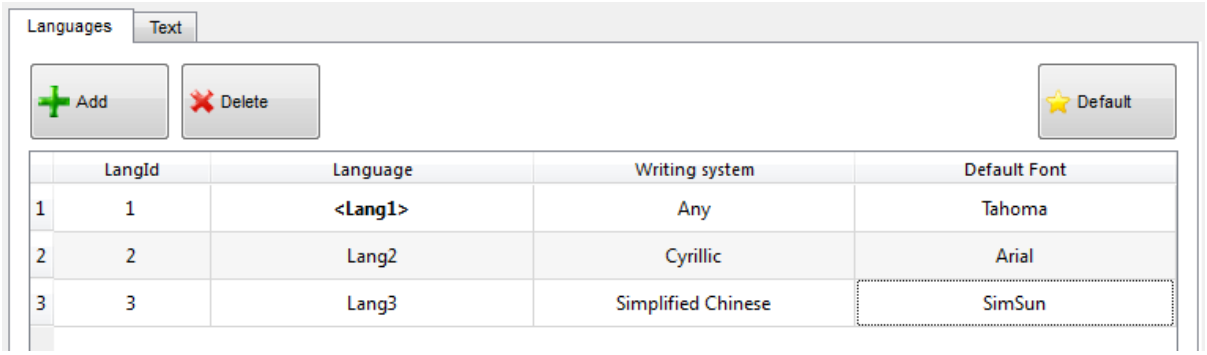

Figure 157

11.1.1 Language Display Combo

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

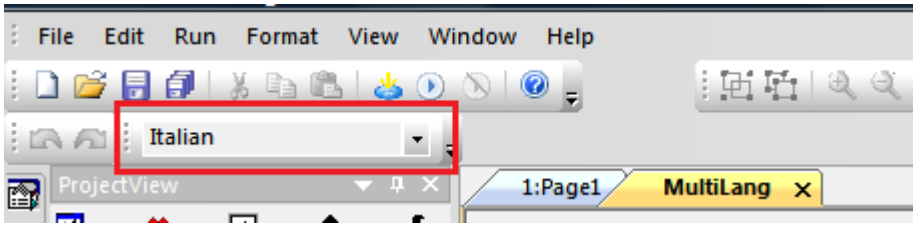

Figure 158

11.2 Multi Language Widget

Multilanguage support is available for different objects, like push buttons, static text, message, 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 below will open. Here, you can edit the text for the selected language 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. You can also use the export and import features, as described in chapter: Export and Import of Multilanguage Strings.

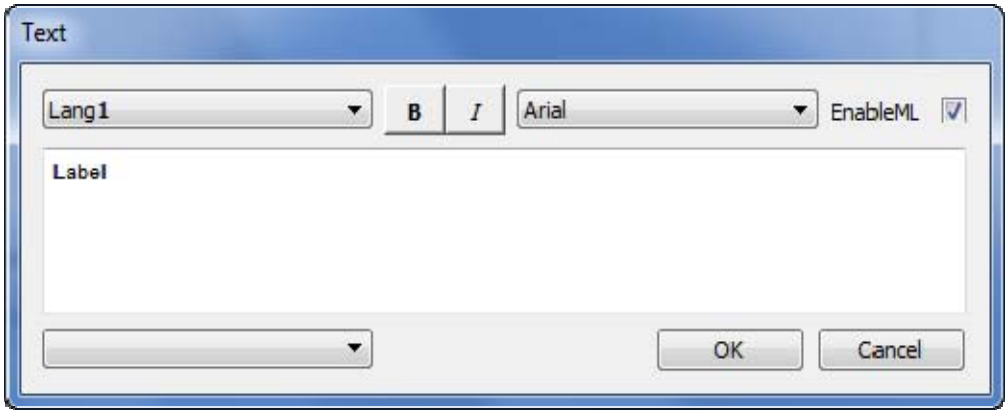

Figure 159

11.2.2 Multi Language for Message Widget

VisuControl 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 box and enter the message description for the selected language. You can also use the export and import features, as described in chapter: Export and Import of Multilanguage Strings.

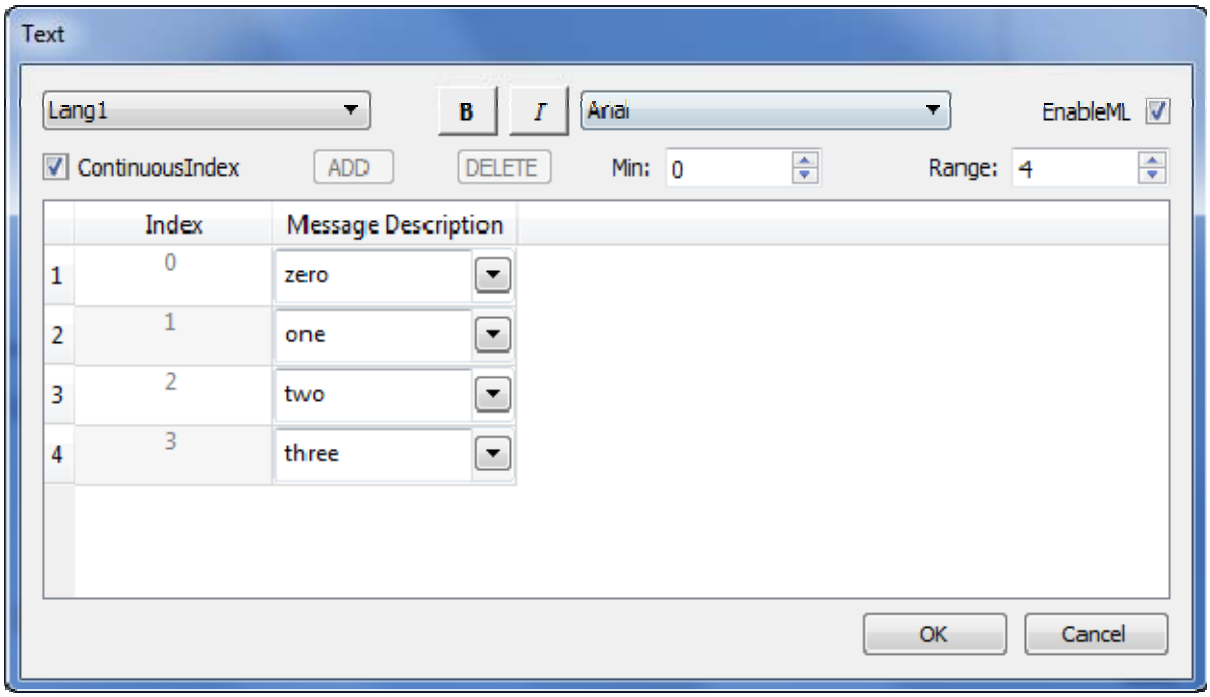

Figure 160

11.2.3 Multi Language for Alarm Messages

VisuControl 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: Export and Import of Multilanguage Strings.

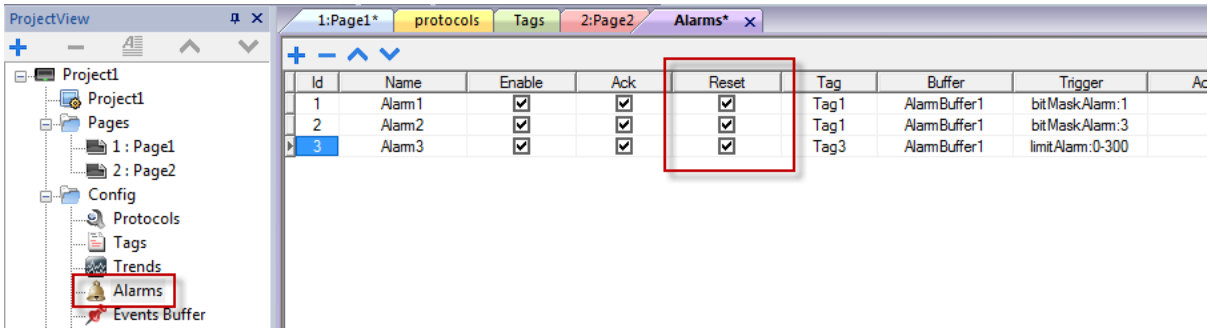

Figure 161

11.2.4 Multi Language for Popup Messages

For the popup message macro, you can define the Multilanguages. 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 below figure).

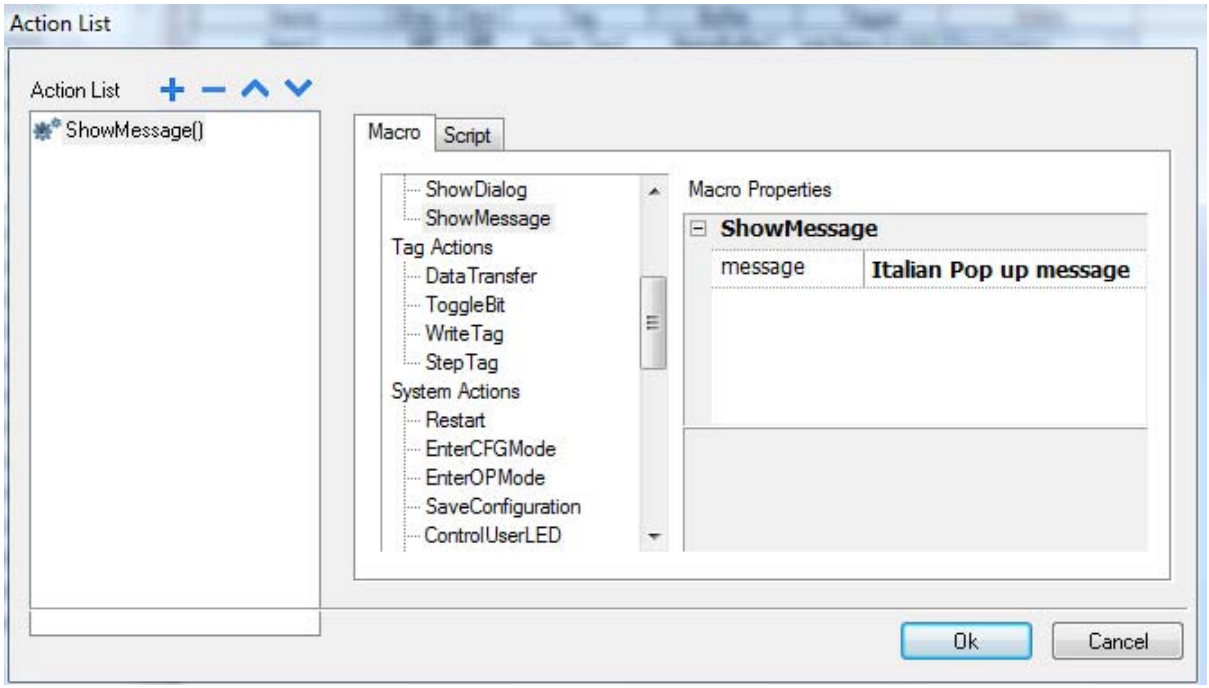

Figure 162

11.3 Export and Import of Multilanguage Strings

The easiest way to translate a project into multiple languages is to use the Export feature, exporting all text to an external file. The translation can be executed in that documents, then using the Import, bring all text for all languages back into the project.

The Multilanguage strings will be exported in CSV file format, then you can modify the strings with an external editor, and import it back to the VisuControl.The CSV file exported by VisuControl is coded in Unicode. To edit it, you need a specific tool that supports a CSV 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 VisuControl. Click “Save" button to save the text.

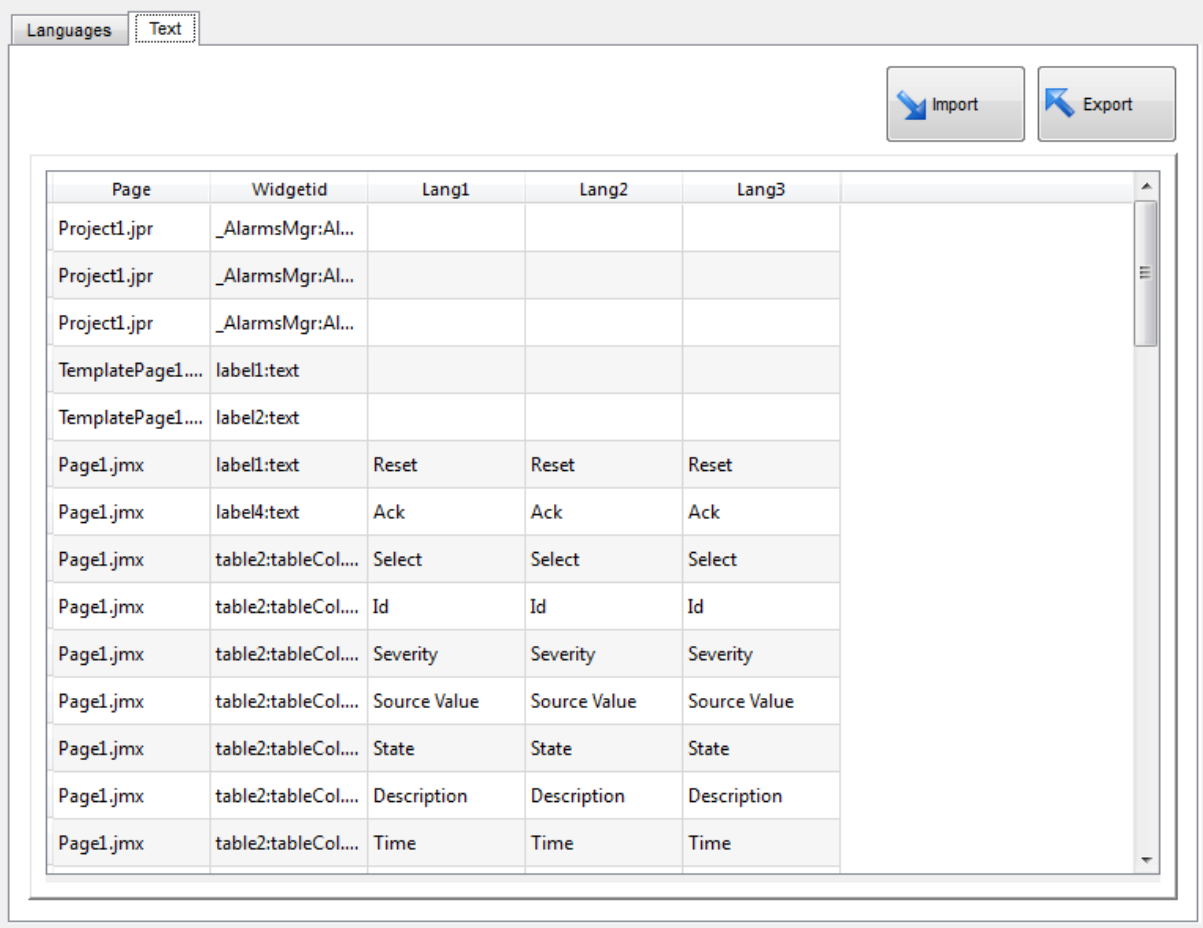

Figure 163

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.

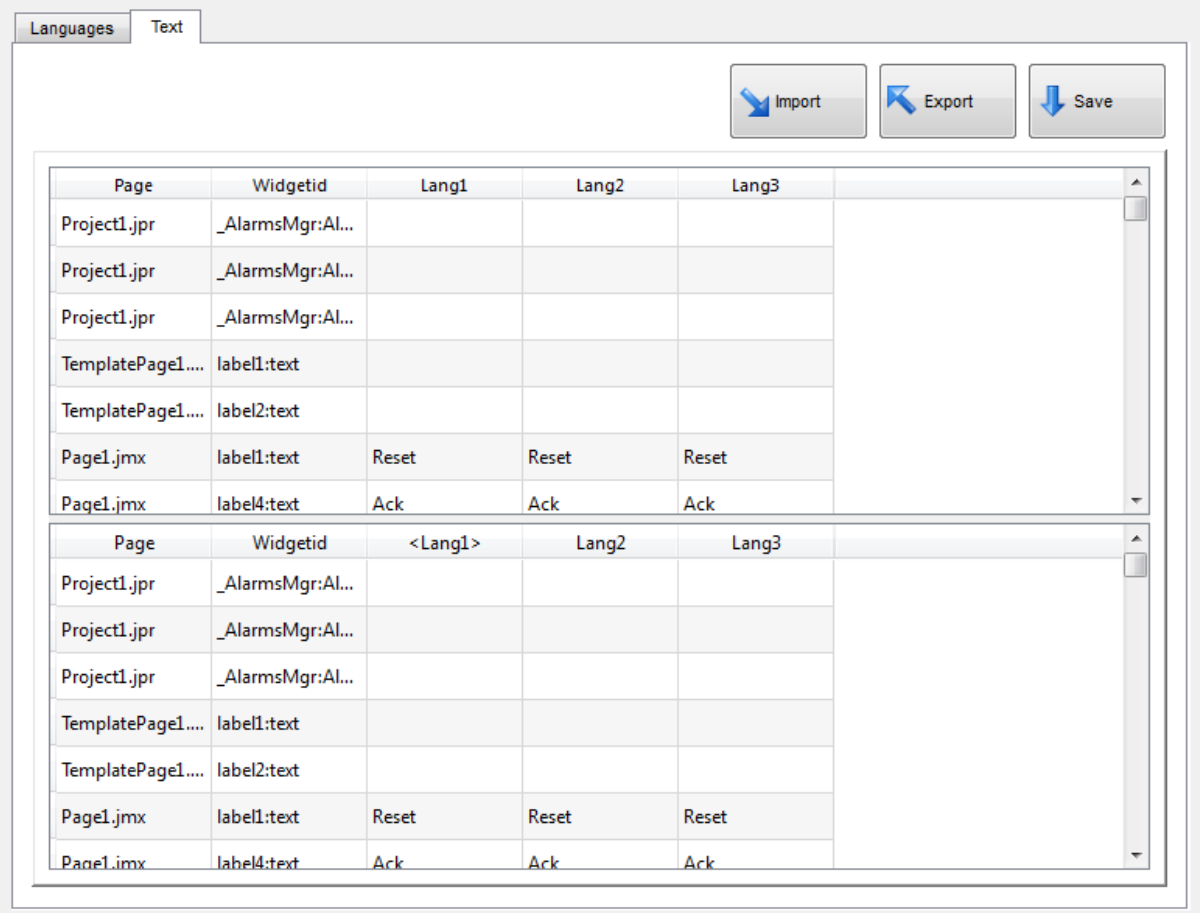

Figure 164

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.

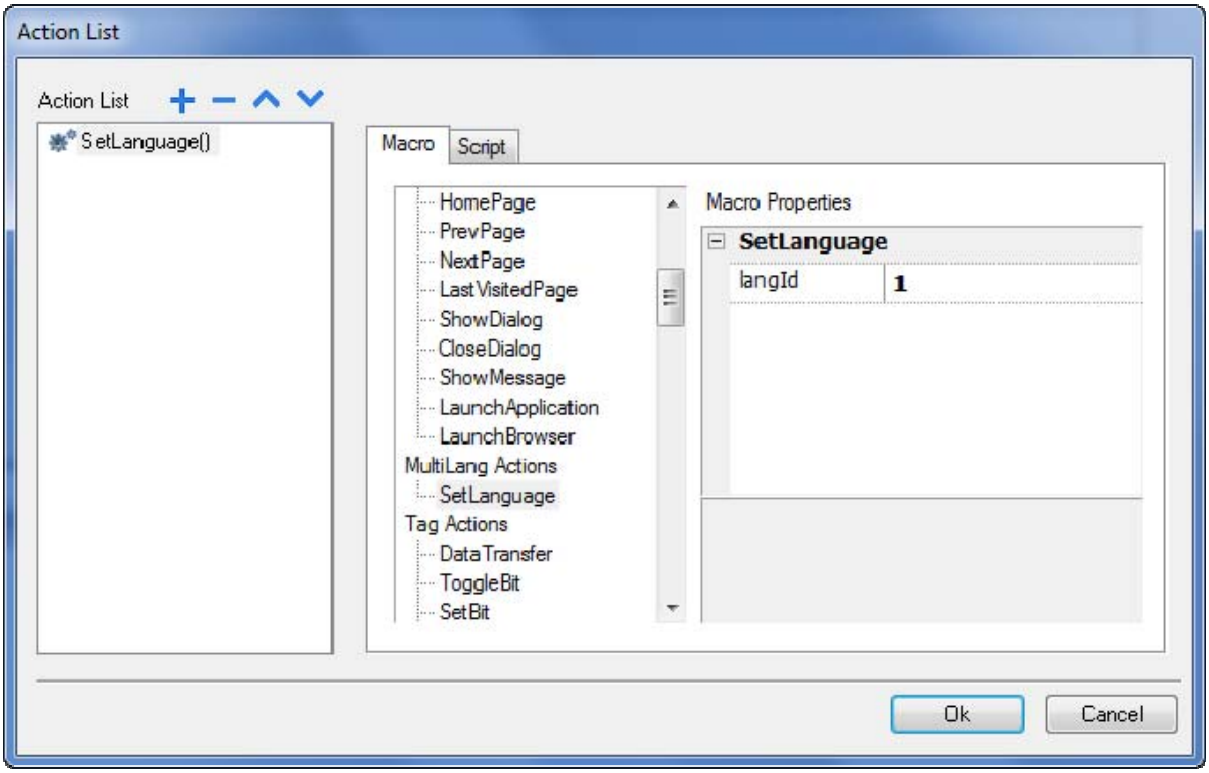

Figure 165

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 the Scheduler

The VisuControl System provides a scheduler engine that can be easily configured to program the execution of specific actions at repeated intervals, on time 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 to run on the panel. This includes selecting the actions to perform when the scheduled event is activated.

The first step is performed using the Scheduler Editor, in VisuControl.
The second step is to create a Runtime user interface that allows the end-user to change settings per each defined scheduler. For example the Runtime 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. This can be done by dragging and dropping a predefined Scheduler 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 Scheduler Editor is accessible from the “Scheduler" voice of the ProjectView pane (as shown in below figure).

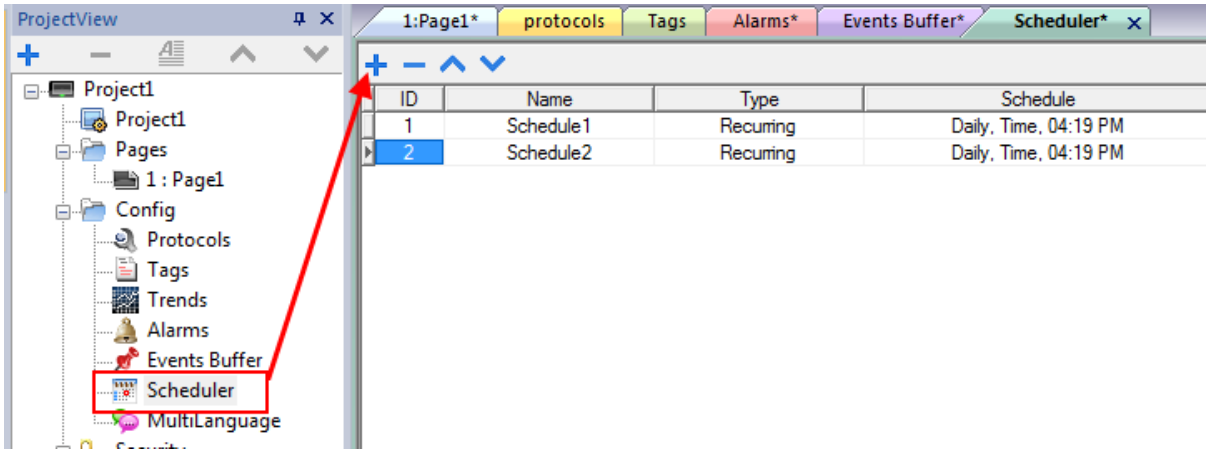

Figure 166

Click on the “+" symbol to add a schedule item. Scheduler can be of two different types as listed below and shown in below figure:

Recurring Scheduler
High Resolution scheduler

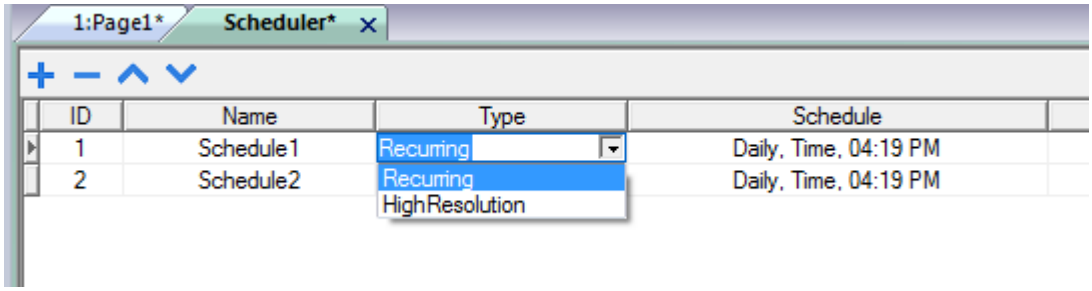

Figure 167

Name Allows you to define the Schedule name.

Type Allows you to select the type of scheduler.

Schedule Allows you to select different scheduler options, which are described in chapters Recurrence Scheduler and Type.

Action Allows you to define macros, which have to be executed at the scheduled time.

Priority 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.

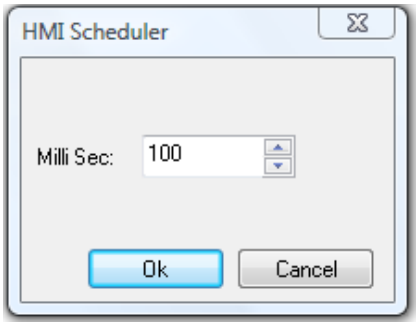

Figure 168

Note: The High Resolution scheduler cannot be changed during run-time. 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.

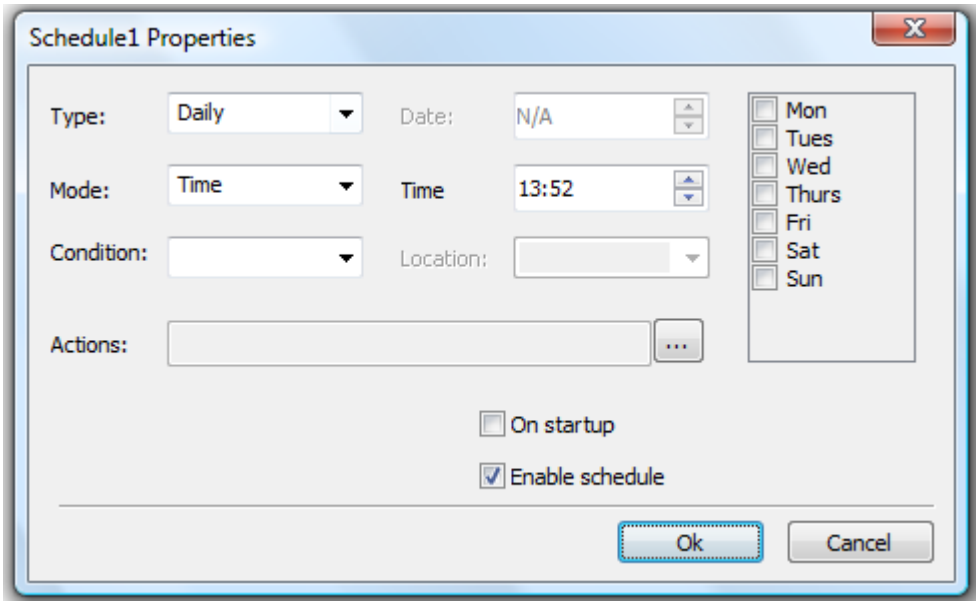

Figure 169

Each Scheduler can be configured to run once at Startup (when the “On Startup” check-box in marked). Additionally, you can specify ,by default, the scheduler to be enabled or not at first run by using the “Enable scheduler” check box.

12.4 Type

The Type combo allows you to select the type of the Schedulers (as shown in below figure). However, you can change the type for the scheduler at anytime during the Runtime, as described in chapter: Schedule the Events During Runtime.

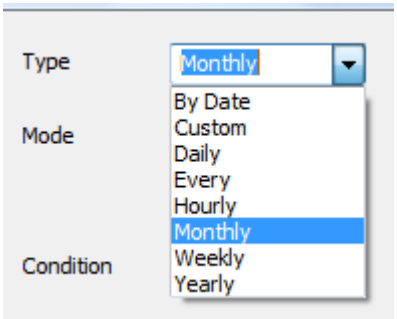

Figure 170

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 are 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: Mode.).

12.4.3 Every Schedule

The Every Scheduler is much like the High Resolution scheduler, with the possibility of changing it in the Runtime. The “Every" Scheduler allows you to execute 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 the 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 savings time to be set on a specific date every year, set the date and time of the starting date for the daylight savings 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.

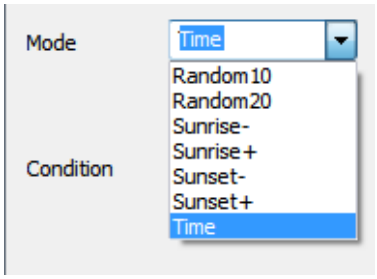

Figure 171

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 VisuControl

In VisuControl, 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, VisuControl automatically calculates the sunrise and sunset time.

In theVisuControl 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 VisuControl\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, VisuControl displays the city name in the place combo list, and you can see the sunrise and sunset time on the dialog (as in the below figure).

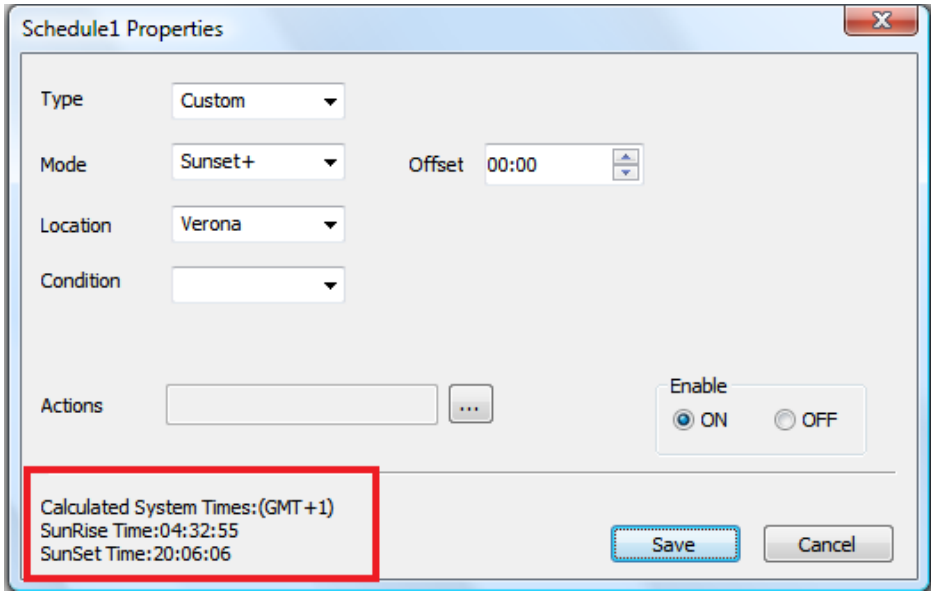

Figure 172

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- Scheduler, 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 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.

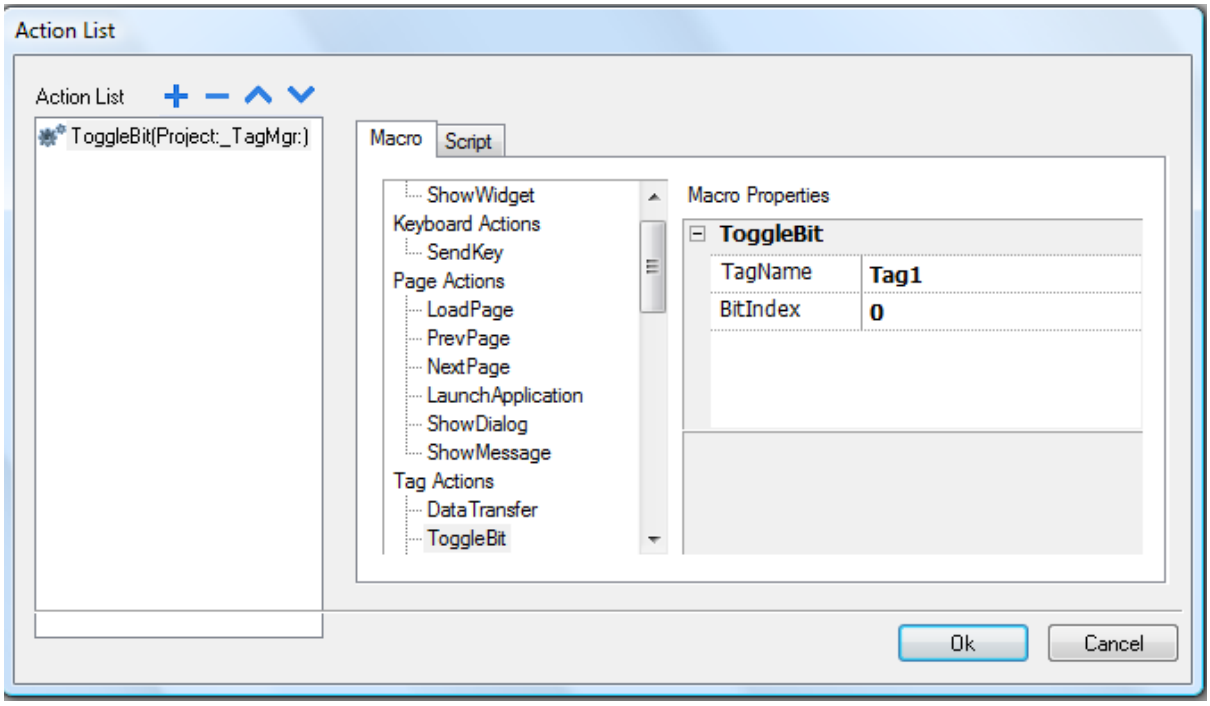

Figure 173

Note: The Actions should be programmed in VisuControl. Actions 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 below figure) where you can add the schedule from the list.

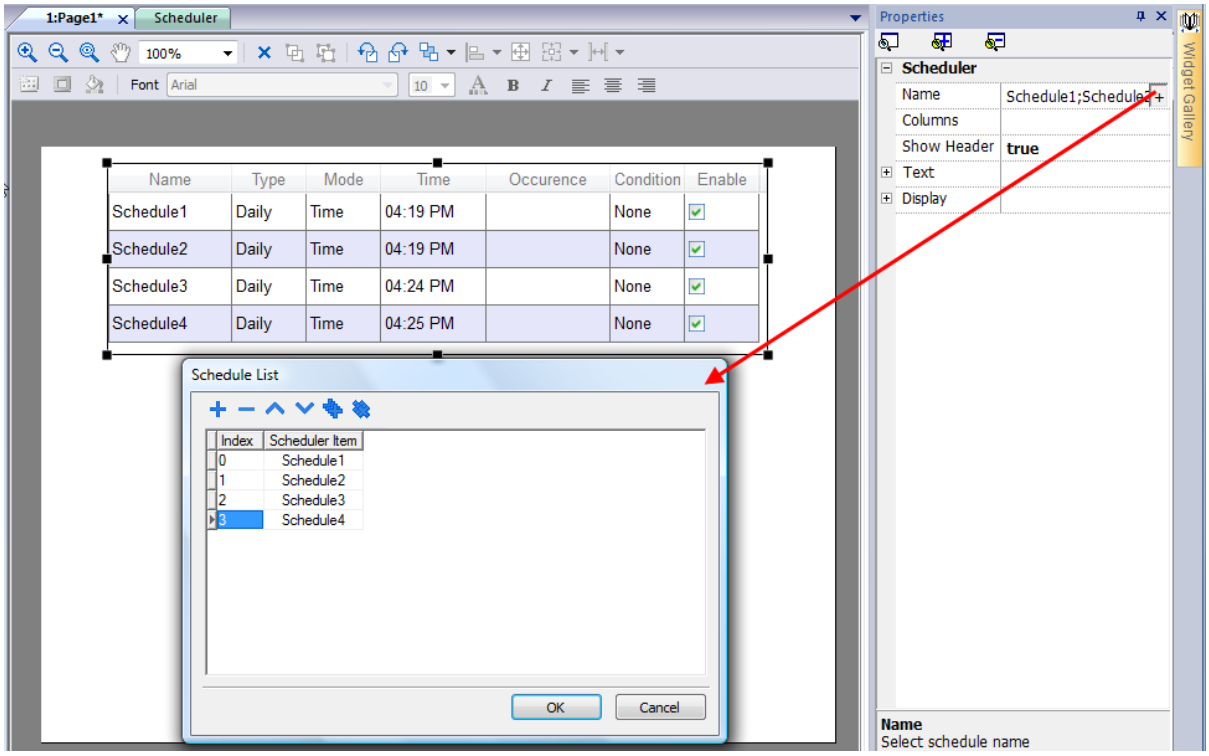

Figure 174

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 Configuring the Schedule Interface for Run-time Interaction), 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 modes as described in the dedicated chapter.

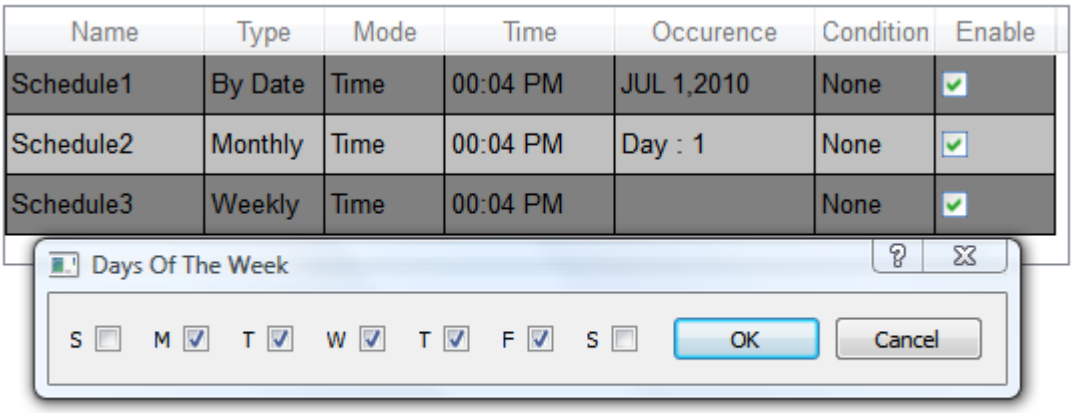

Figure 175

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.

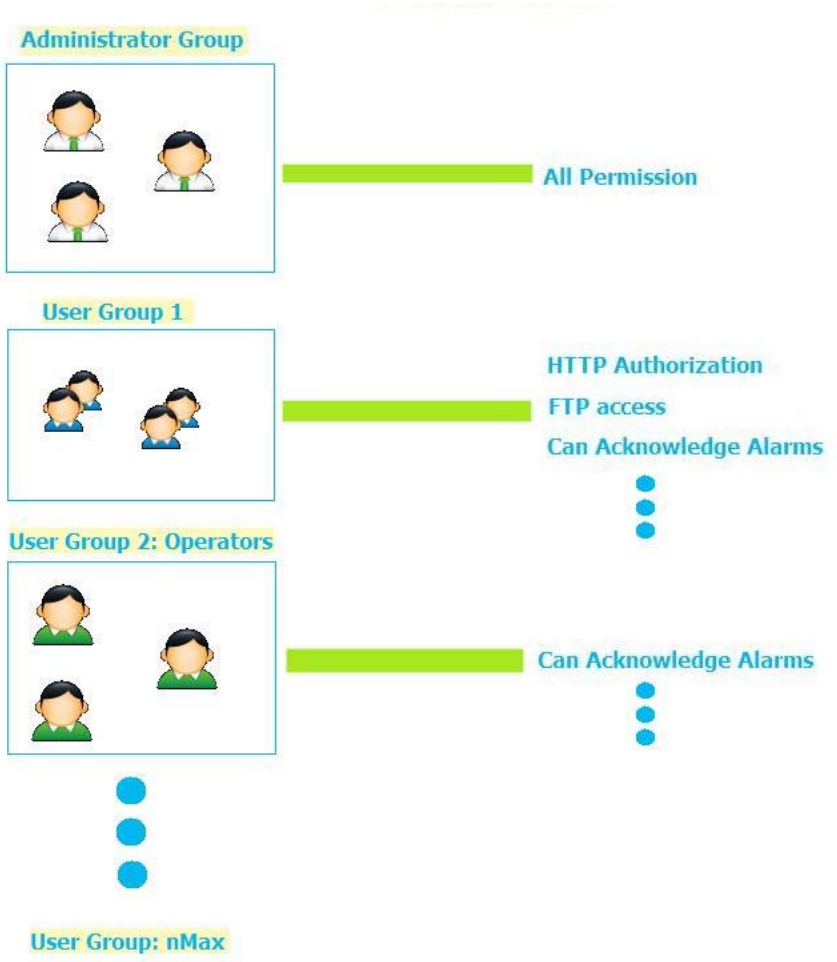

Figure 176

13.1 Configuring Security Options

The section below describes how to configure security-related settings in VisuControl.

Note: To enable or disable the user management feature, right click on the “Security" folder from the Project View and set Enable or Disable. See the following figure as reference.

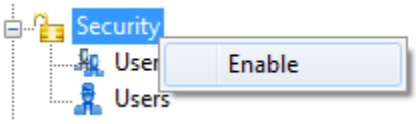

Figure 177

13.2 Configuring Groups and Authorizations in VisuControl

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

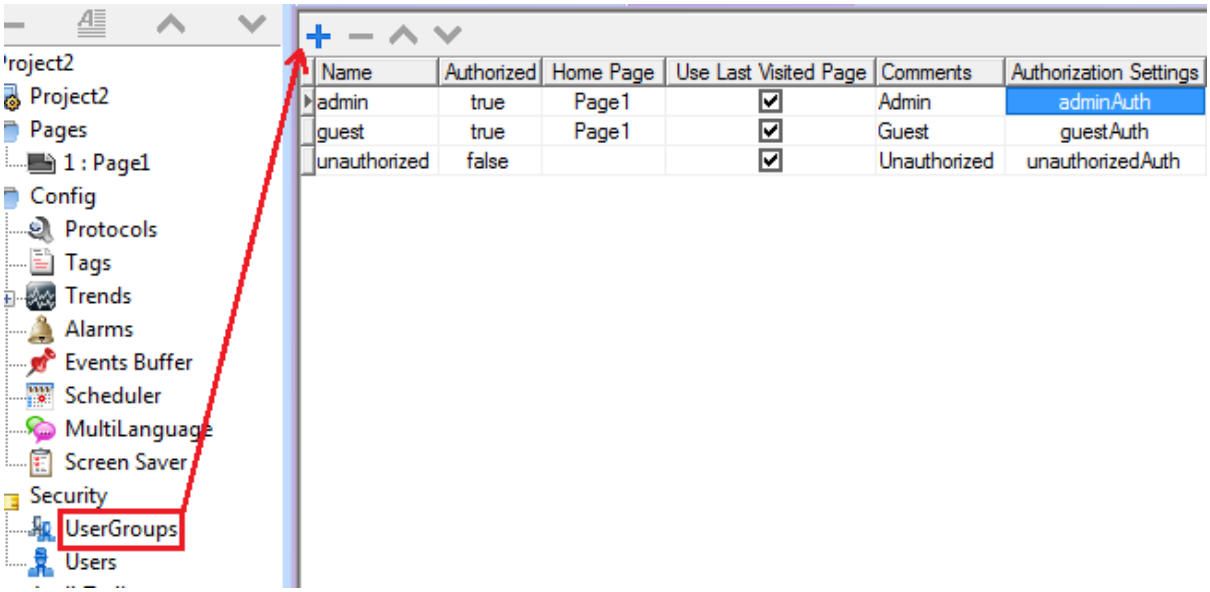

Figure 178

New User groups can be added by clicking the “+” Button. Three predefined groups are available by default, these groups cannot be deleted and their names cannot be changed. Predefined group’s authorizations and comment fields can be instead changed according to application requirements. For each group of users you can assign a home page. This means that, whenever a user from this User Group is logged in, the selected Home page for that group will appear.

There is one additional option called ‘Use Last Visited Page’. If enabled, and a new user logs in, the page visited by the previous user will be displayed. If this page is for some reason, not “accessible” in terms of rights for the current user, the system reverts to the default project home page.

13.3 Modifying the Access Permission of Groups

To modify and assign the permissions, click the browse button on the Authorization Setting column. The Admin Authorizations dialog will open, here you have several tabs for the several available options.

13.3.1 Widget Permissions

The following figure shows the dialog where you can change the widgets' permissions.

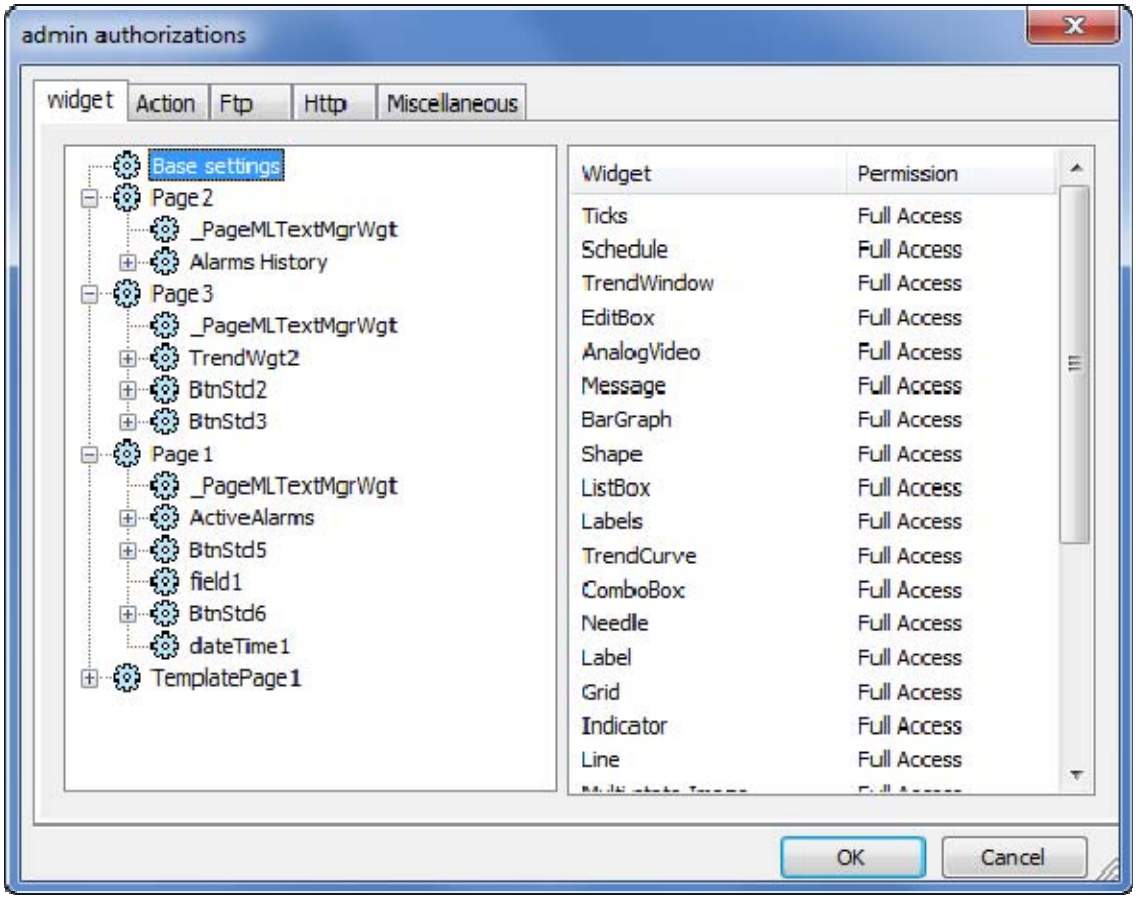

Figure 179

For the widget, the possible options are: Full-Access, Read-Only and Hide.
When you click on “Base settings” the right part of the dialog shows the permissions that will be valid as default and at project level.

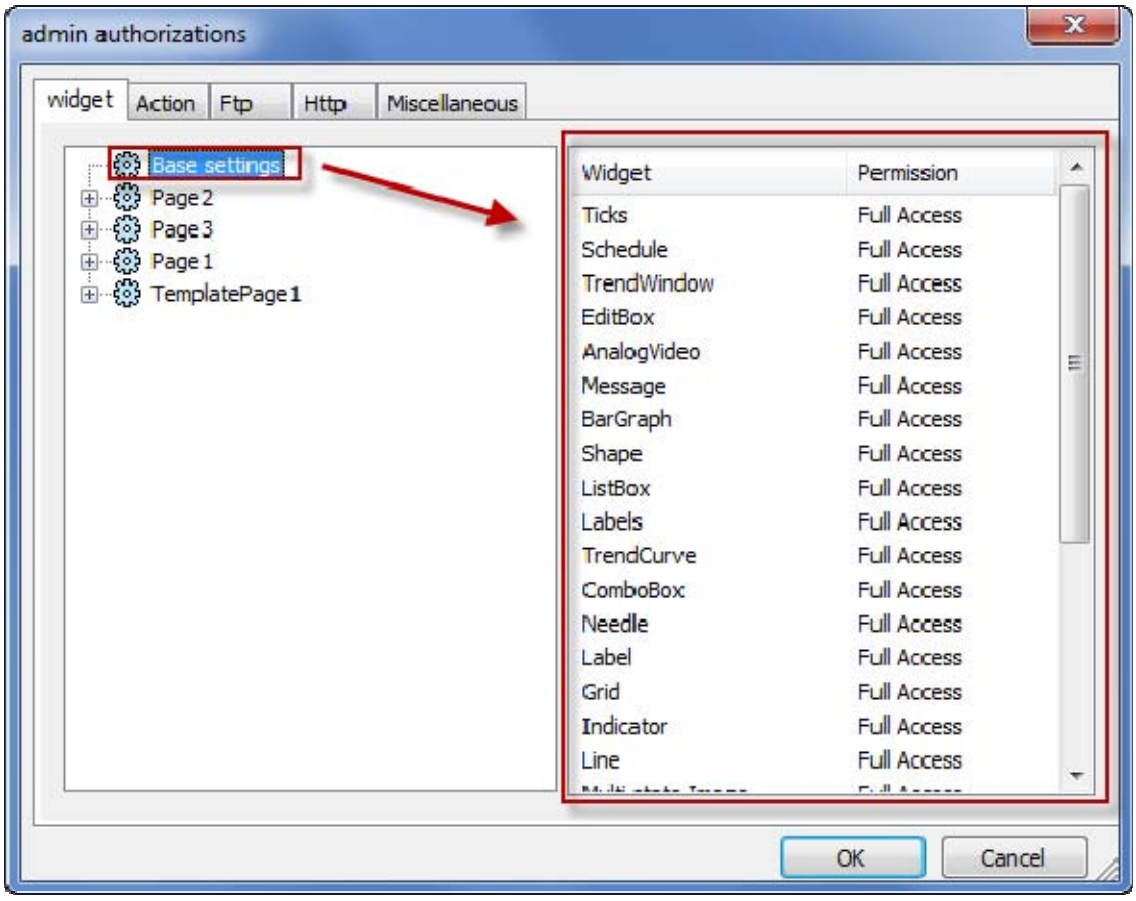

Figure 180

The widgets 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.

Permissions can be given at three levels, Project level, Page level and Widget level.
In the Tree structure the permission for a page can be set as Full Access, Hide or Read Only.
All the widgets in this page will take the settings that have been assigned to the page with a type of hierarchy logic..

Suppose the page permission is set as 'Read Only', then all the widgets in the page will have the permission as "Read Only". On selecting a widget inside the page from the tree structure, you can see that the permission is given as "Use Base Settings". It means that it takes the permission given to the page.(Read Only). The widget permission takes the priority as follows:

Low priority	- Basic settings (widget settings in general for the project)
Medium priority	- Page settings (settings for all the widgets of a particular page)
High priority	- Widget settings (individual widget's or its group/parent widget's permission of any page).

For example suppose a widget is set "Read Only" permission at project Level and it is given "Full Access" at page Level then the page Level Settings will be taken.

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

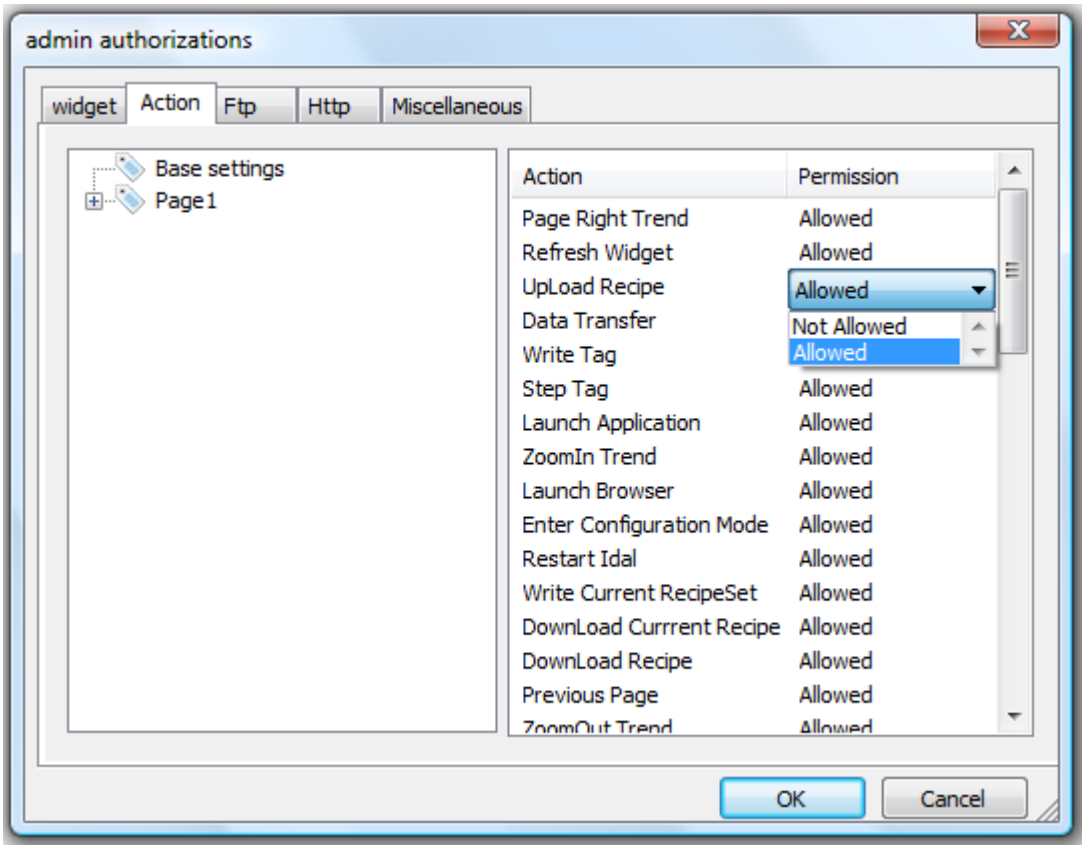

Figure 181

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 page, and widget 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

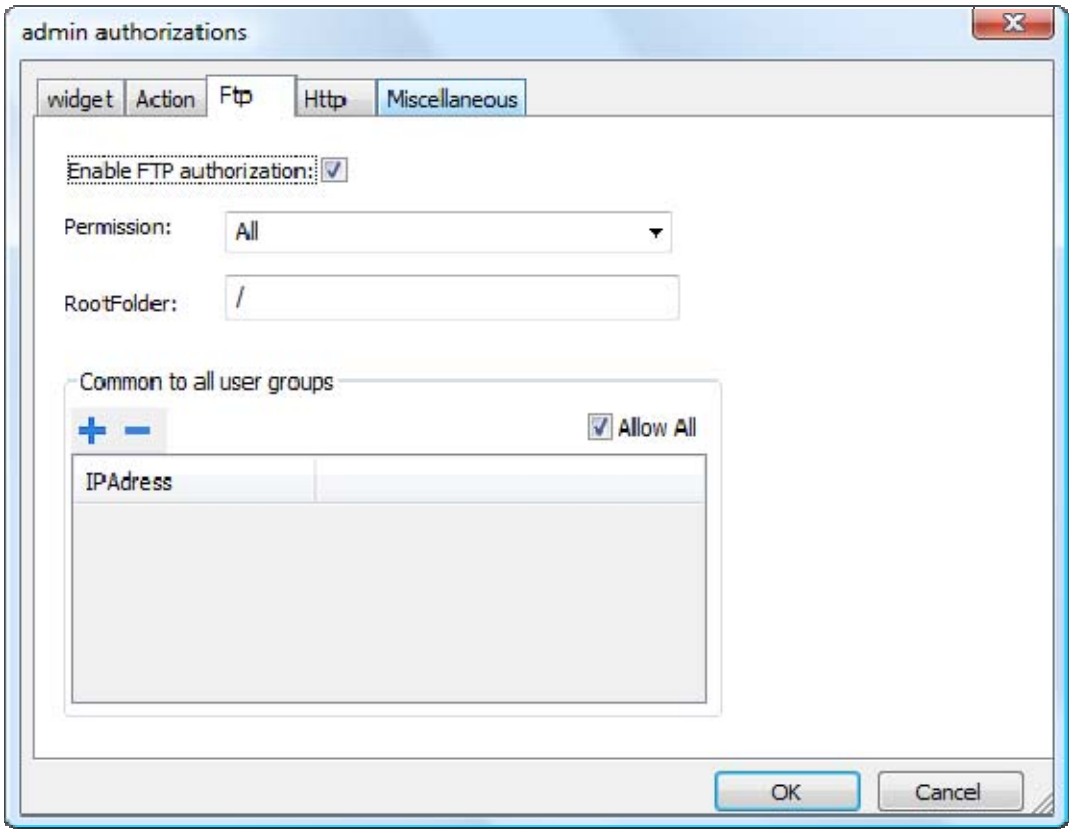

Figure 182

13.3.4 HTTP Authorizations

The HTTP authorization dialog allows you 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 VisuControl web server.

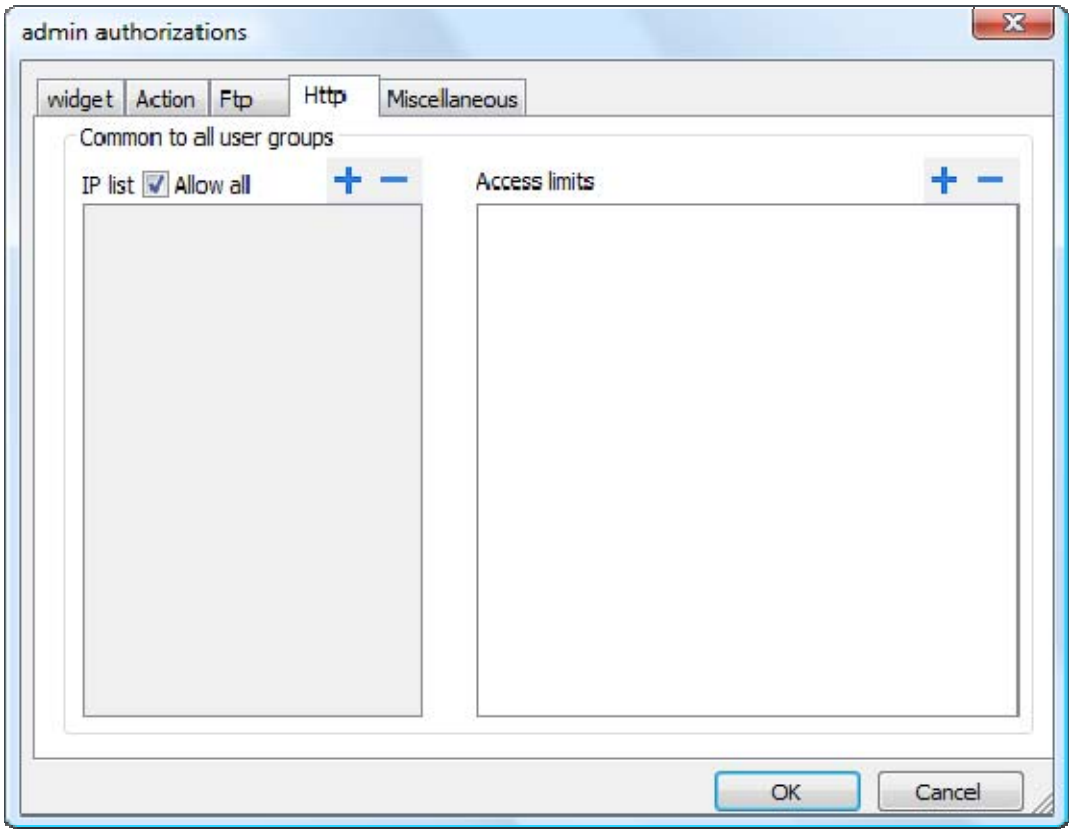

Figure 183

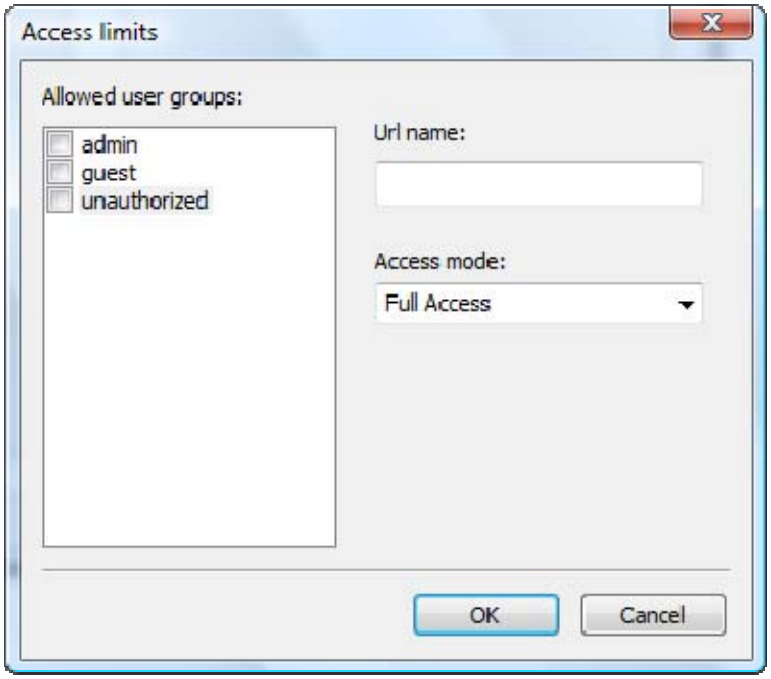

Figure 184

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.

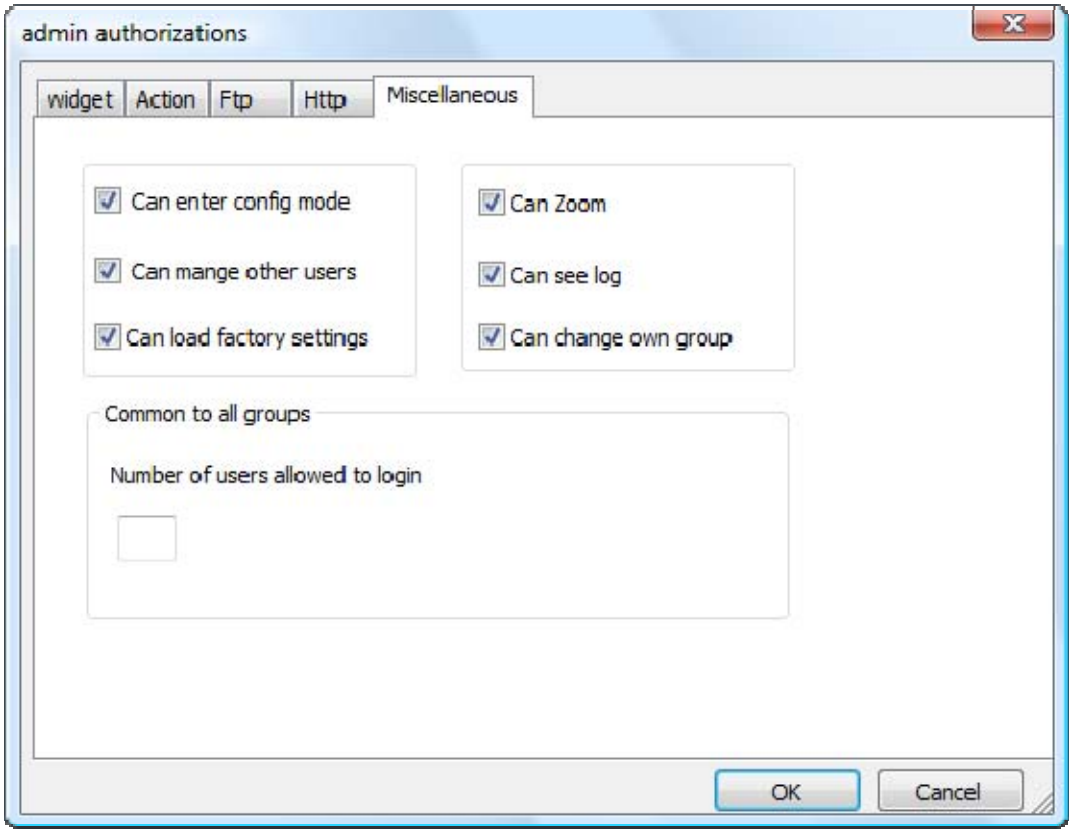

Figure 185

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, has priority on the base settings.

13.4 Configuring Users in VisuControl

To configure users in VisuControl, 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.

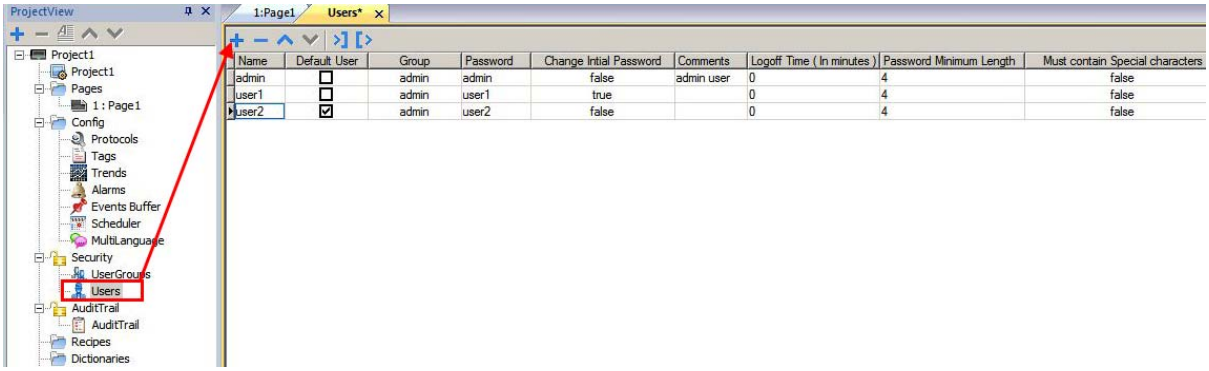

Figure 186

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 VisuControl 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 VisuControl, 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).

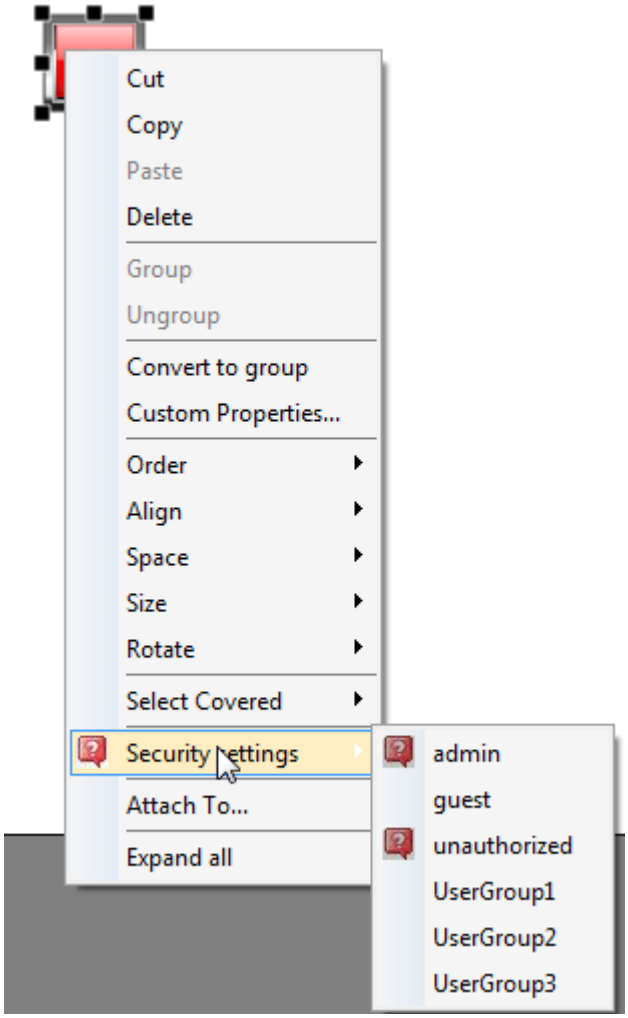

Figure 187

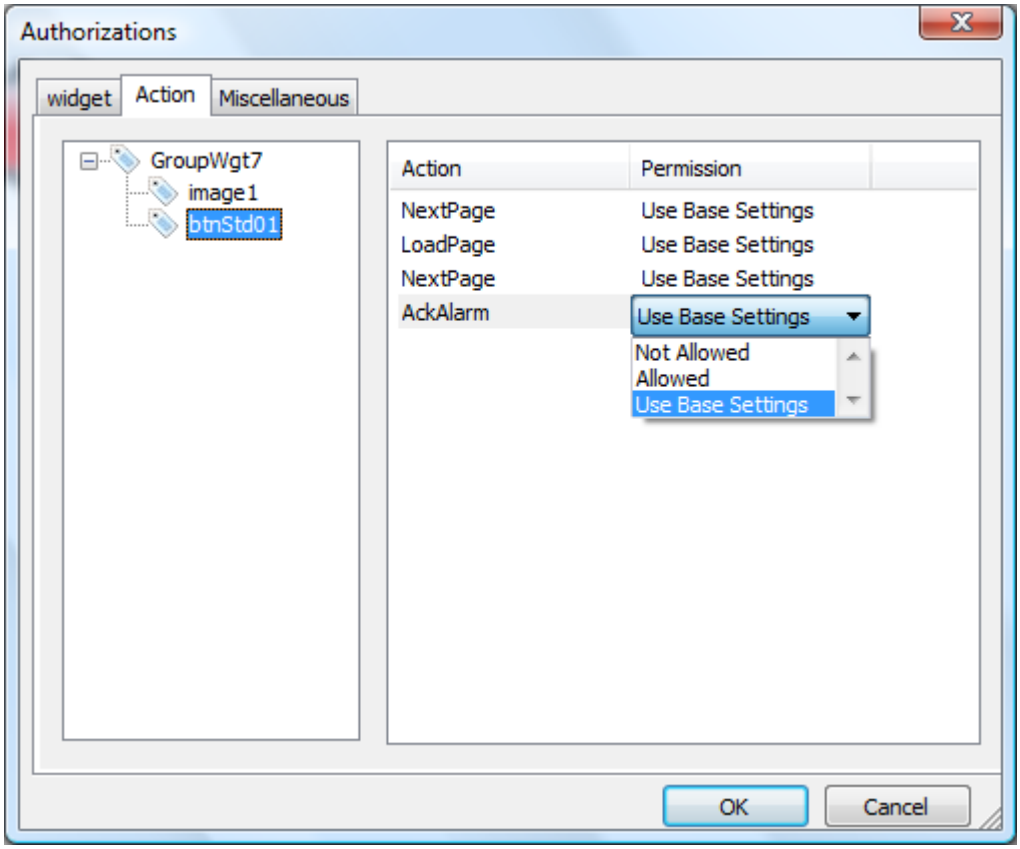

Figure 188

13.7 Operation on Runtime

After starting the Runtime, the VisuControl 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 VisuControl 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 below and explained in chapter User Management Actions.

All the users' information modified in Runtime are stored in a separate file, thereby preventing loss of the users' configuration in case of a new project downloads.
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 189

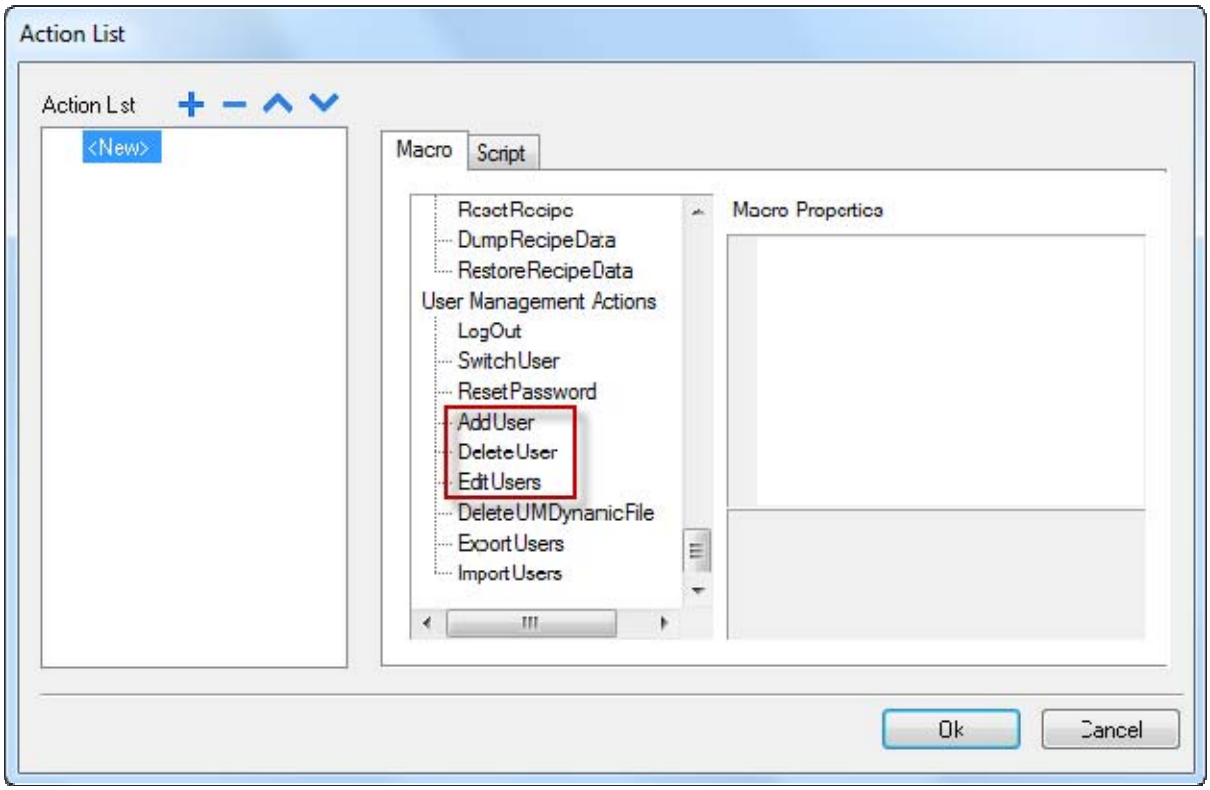

Figure 190

14 Audit Trails

VisuControl 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 them. The Audit Trail can be enabled with or without user management. So it can 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, the padlock symbol shows locked, otherwise, it stays open.

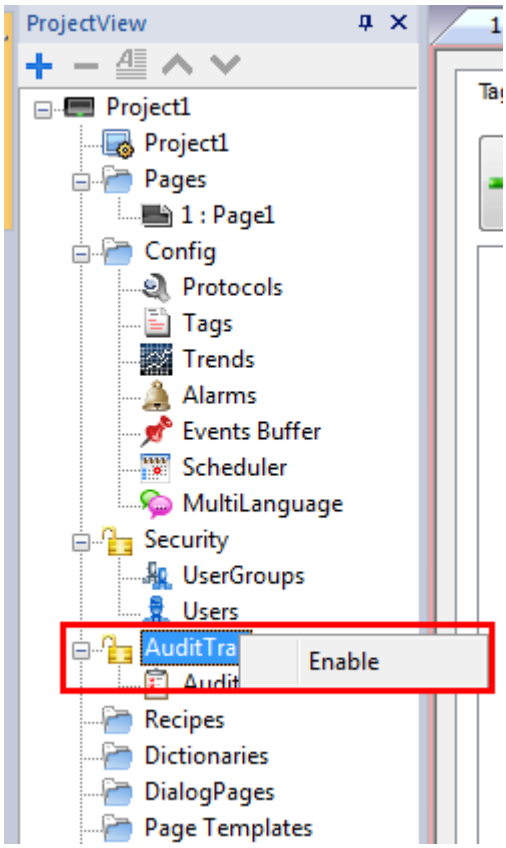

Figure 191

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 then select the log type “Audit".
Here there is an option for selecting the storage where the dumped Audit files have to be stored. There are three options: Flash , USBMemory, SD Card. For SD Card and USBMemory specific path has to be provided.

Note: an external USB pen drive is referenced by the operating system as “USBMemory”; an SD card as “Storage Card”.

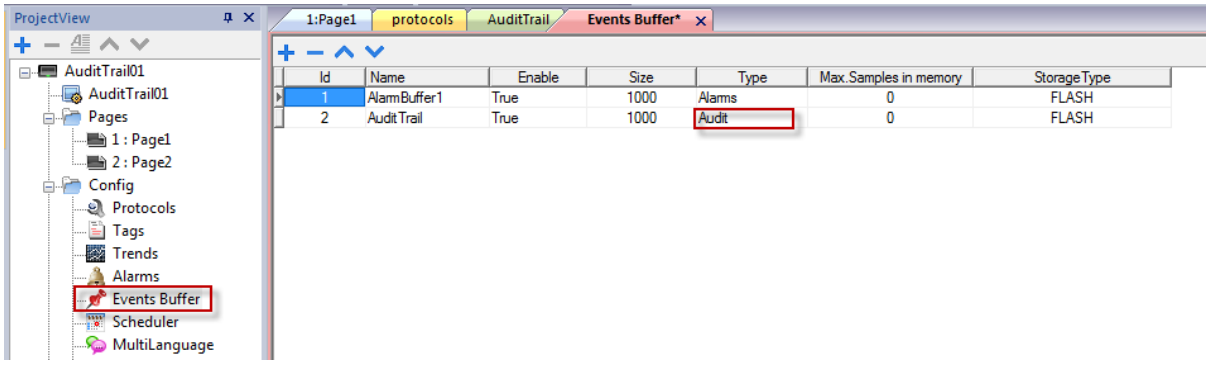

Figure 192

When the events’ buffers are stored in the internal flash disk, the systems provides to save the file on disk every 5 minutes.

The Event Buffer editor provides per each buffer an option called “**Max Samples in Memory**”.

This parameter allows a certain degree of freedom in the control of the number of events retained in RAM before saving them to the file on disk. With this parameter it is possible in fact to decide exactly after how many events the data is written on disk. The parameter can be used ONLY when an external disk storage is selected: When the internal Flash disk is used, the parameter is automatically force to “0” (not used).

Note: if the application requires fast saving of events, we suggest to use an external storage device with adequate support for frequent write operations. It is strongly recommended to avoid fast writing to internal flash disk.

14.3 Configure Tags in the Audit Trail

For most cases, all the tags specified in the project do not necessarily need to be monitored. You can customize the tags to be monitored by Audit Trail.

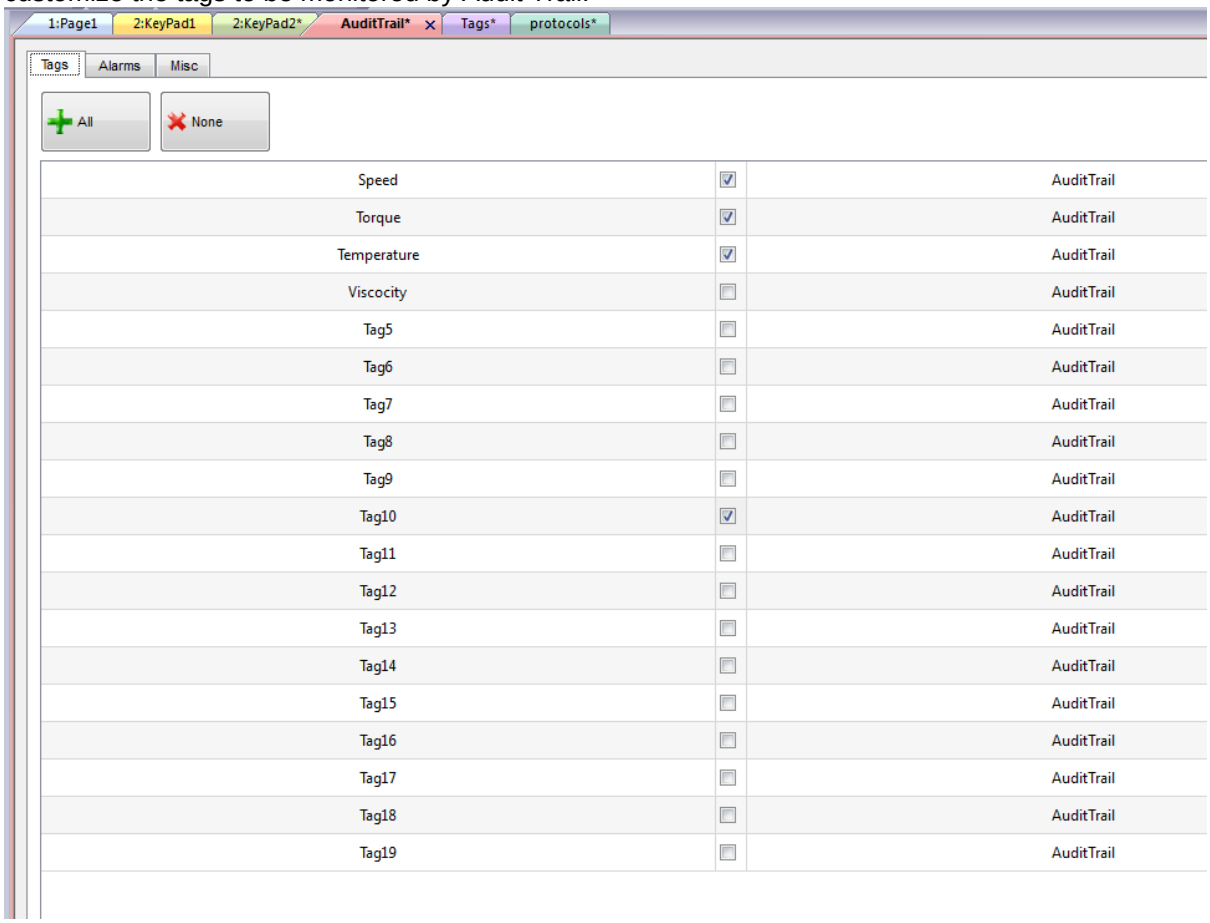

Figure 193

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 operation 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 the Audit Trail. Double click 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.

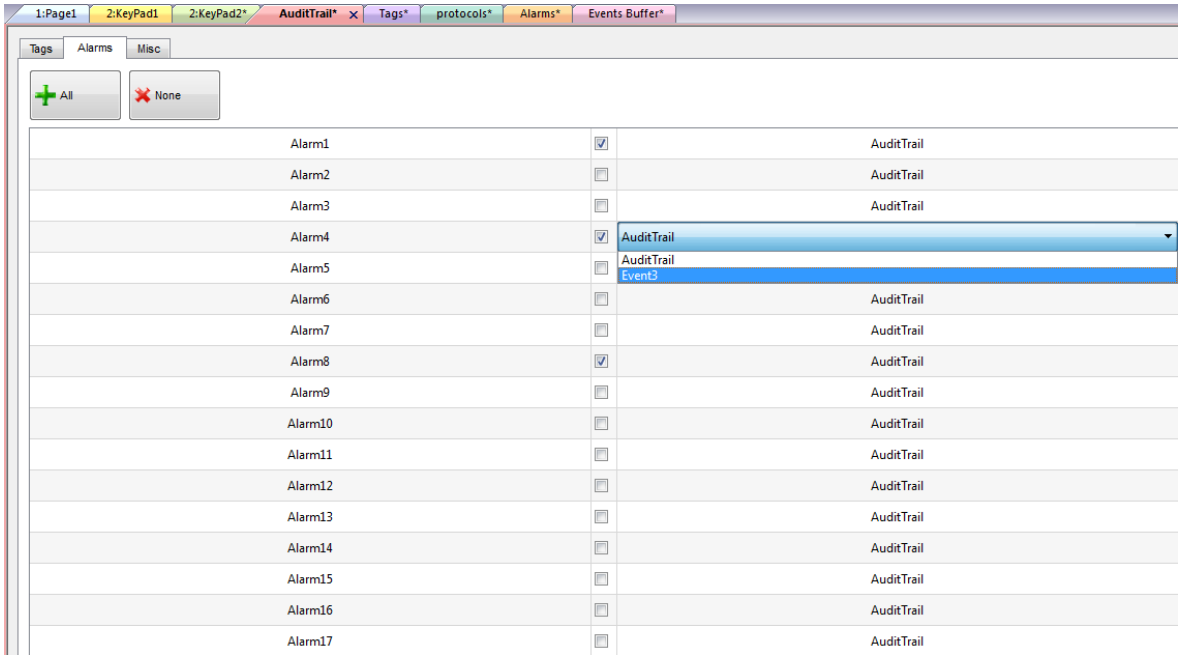

Figure 194

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.

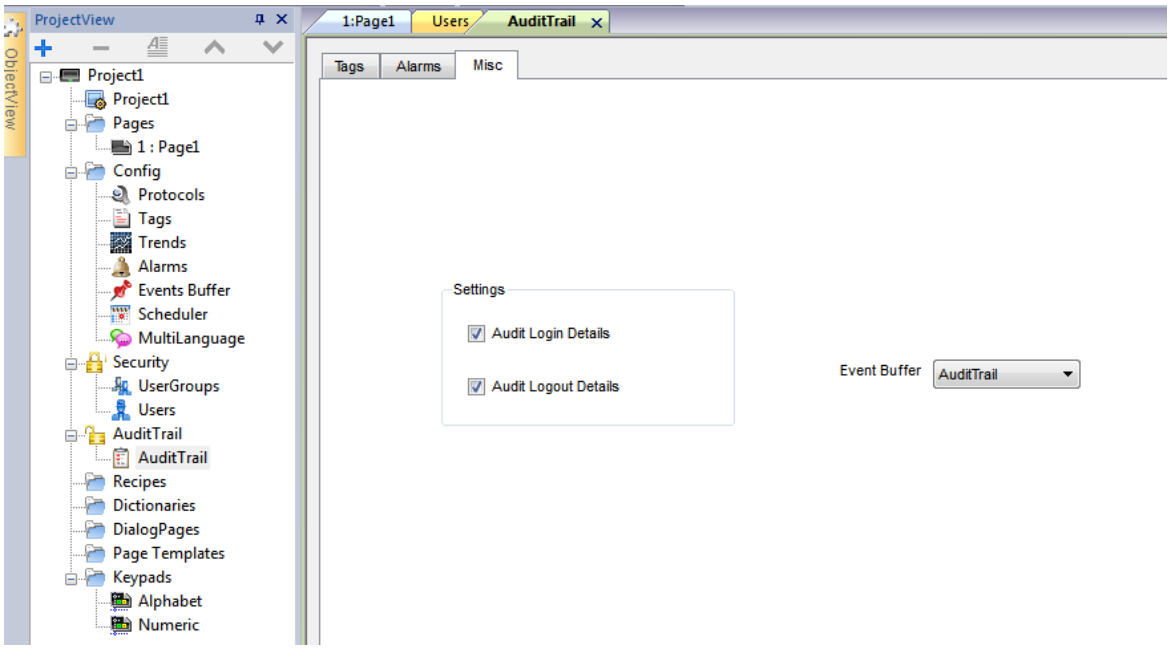

Figure 195

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 Exporting Audit Trail as CSV File

You can convert the audit data to a “.csv" file.
For detailed description look the explanation provided for the Dump Archive macro action, chapter Dump Event Archive.

15 Custom Keypad

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

Figure 196 – Numeric keypad

Figure 197 - Alphabet keypad

Figure 198 - Alphabet Small keypad

Figure 199 - Up Down Keypad

15.1 Creating and Using a Custom Keypad.

Keypads can be created from scratch following the described procedure. Note that you can also change the existing ones directly applying modifications to them.

Right click on the Keypads folder, from the Project View pane. A context menu, as shown in the below picture, will be displayed:

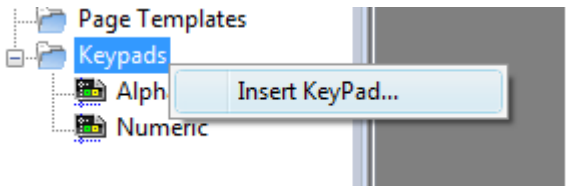

Figure 200

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

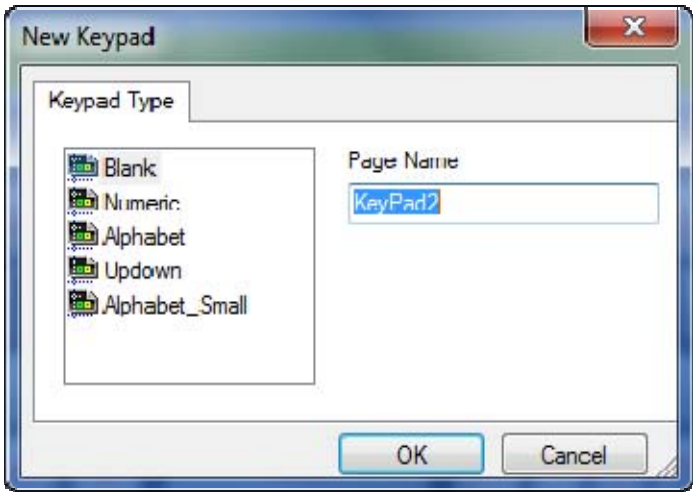

Figure 201

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:

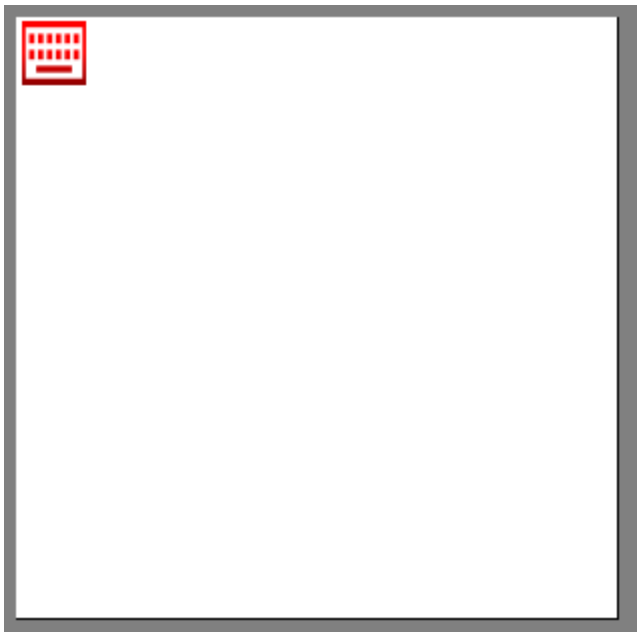

Figure 202.

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

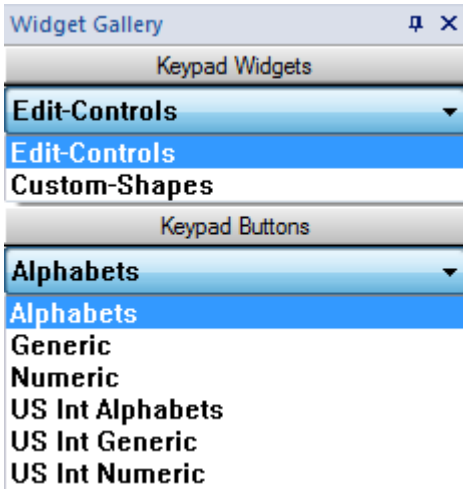

Figure 203

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 204

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.

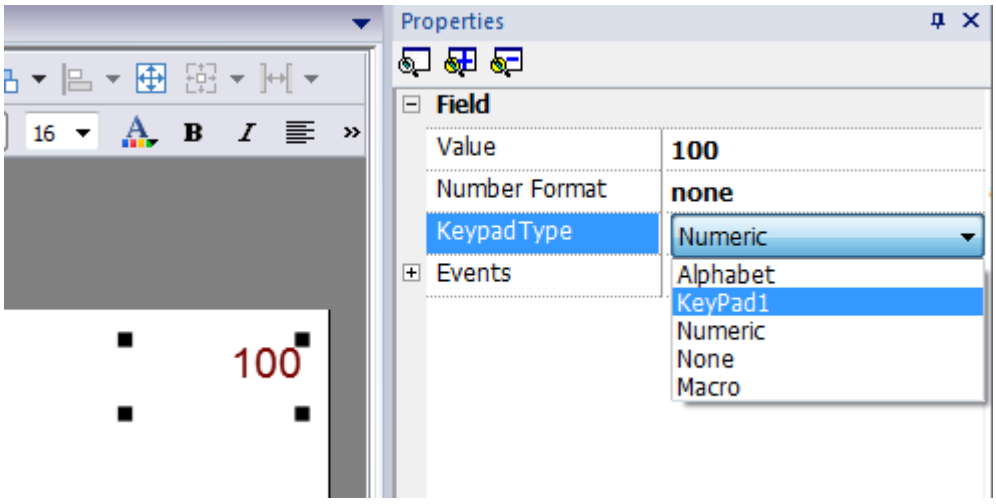

Figure 205

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

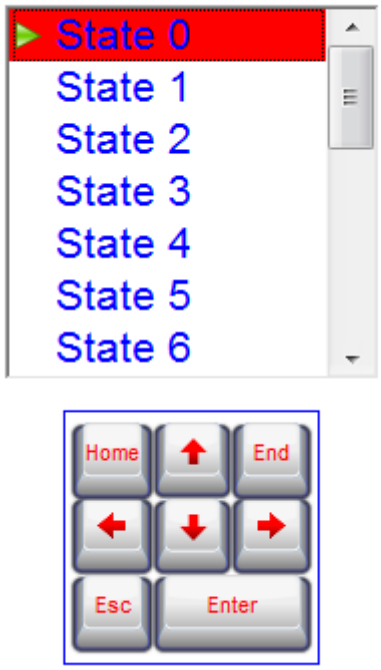

Figure 206

15.2 Deleting or Renaming Custom Keypad

Right click on the keypad you need to delete or rename, from the Project View pane. A context menu (as shown in the below figure) will be displayed.

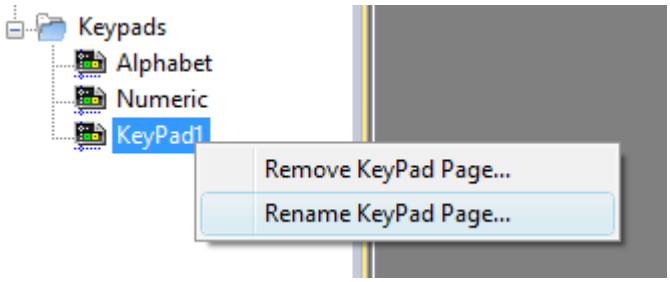

Figure 207

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 Date & Time Widget

Date Time widget is a widget that can be used to view and edit the current time and date at Runtime. The widget can be found in the Widget Gallery in Basic>Edit-Controls. See the figure below.
The widget value has to be attached to the System Time system variable. If you drag and drop the widget it automatically shows the current time, which it takes from the system time.

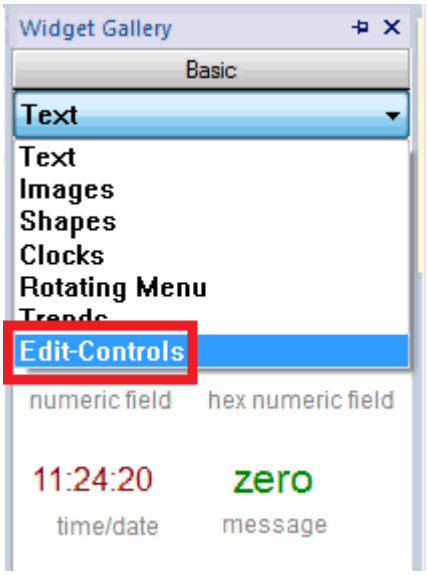

Figure 208

In the Property pane of the widget we can set the format of date and time as “Date only”, “Time only”, “Date and Time”, different formats for representing date and time are available as shown in figure below.

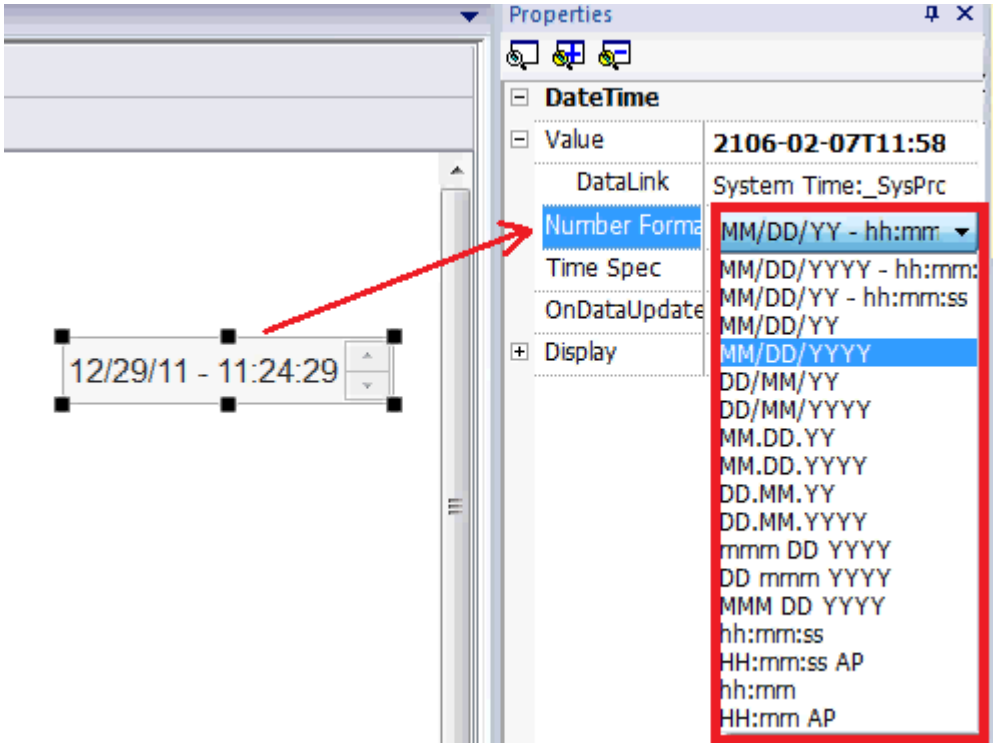

Figure 209

Time Spec option allows selecting which time the widget has to show during Runtime, three options are available for this property: Server, Local and Global.

To understand the difference between the options available for the “Time Spec” property, you need to recall the basic concepts behind the HMI system architecture. Please read the chapter “Server Runtime Modes” to get familiar with the HMI software architecture first.

If we select “Server” as Time Spec, the widget will show the time information as handled by the server side of the HMI system.
If we select Global as Time spec, it will show the Global Time (GMT).
If we select Local as Time Spec, it will show the Local Time in the Widget (the time of the target where the project is running). Please see the chapter “Time zone options for Client“ for additional information about how to interpret the time information when running the project on a remote client.

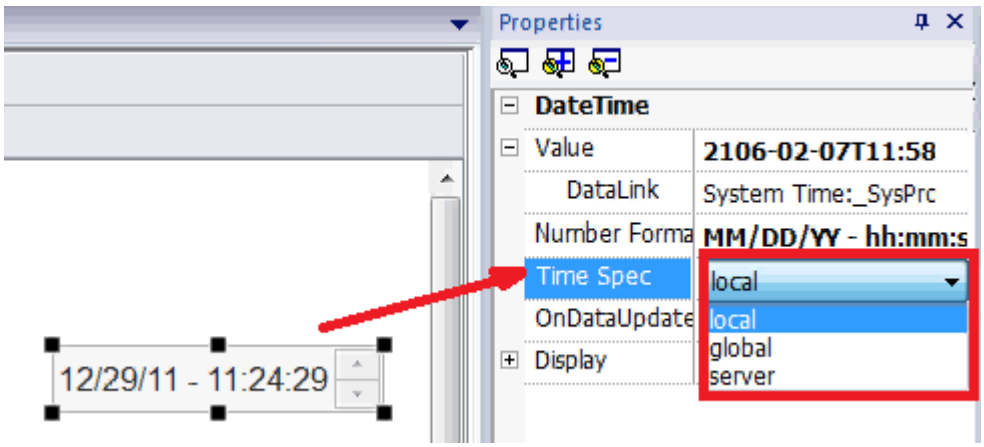

Figure 210

16.2 RSS Feed Widget

The RSS (**R**eally **S**imple **S**yndication) Feed widget allows to display on the screen your favorite RSS feeds directly from the internet.
The widget is available in the Widget Gallery under the Media category. On the page it looks as shown in below figure.

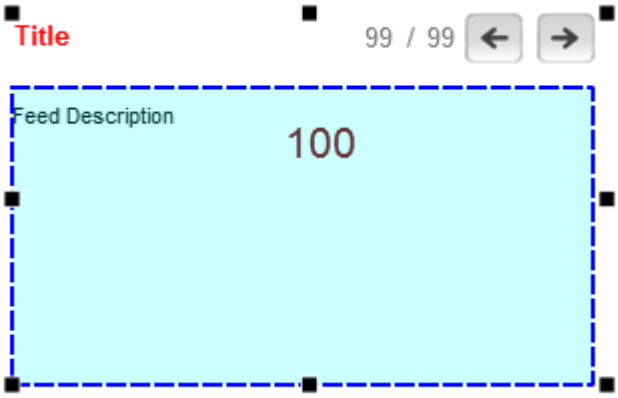

Figure 211

The RSS Feed widget main properties are:

RSS Source	allows you to specify the feed URL
UpdateRate	allows you to specify the refresh time.
	Properties are shown in below figure.
Note:	Feeds sources are fixed and cannot be changed in runtime

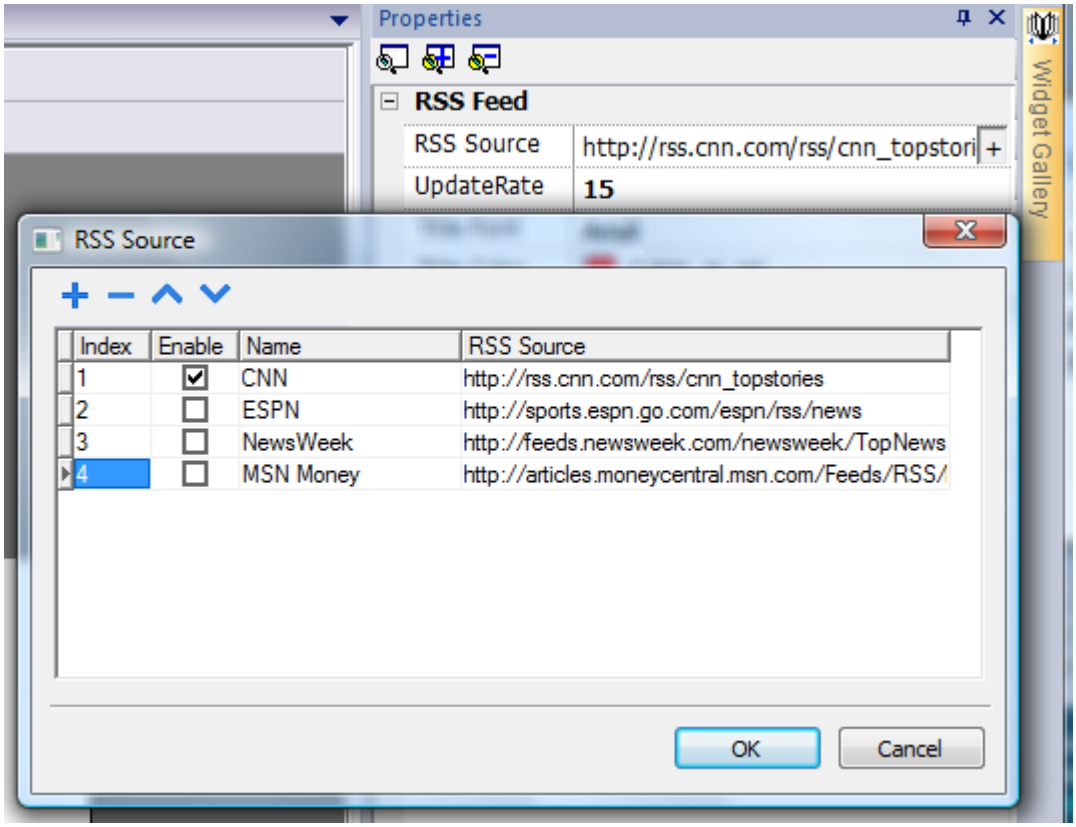

Figure 212

Note: the RSS widget is specifically designed to work on units where the Internet Explorer browser (Pocket Internet Explorer) is part of the operating system.

16.3 Control List Widget

VisuControl provides Control List widgets, a convenient way to represent the status associated to a particular process.
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.

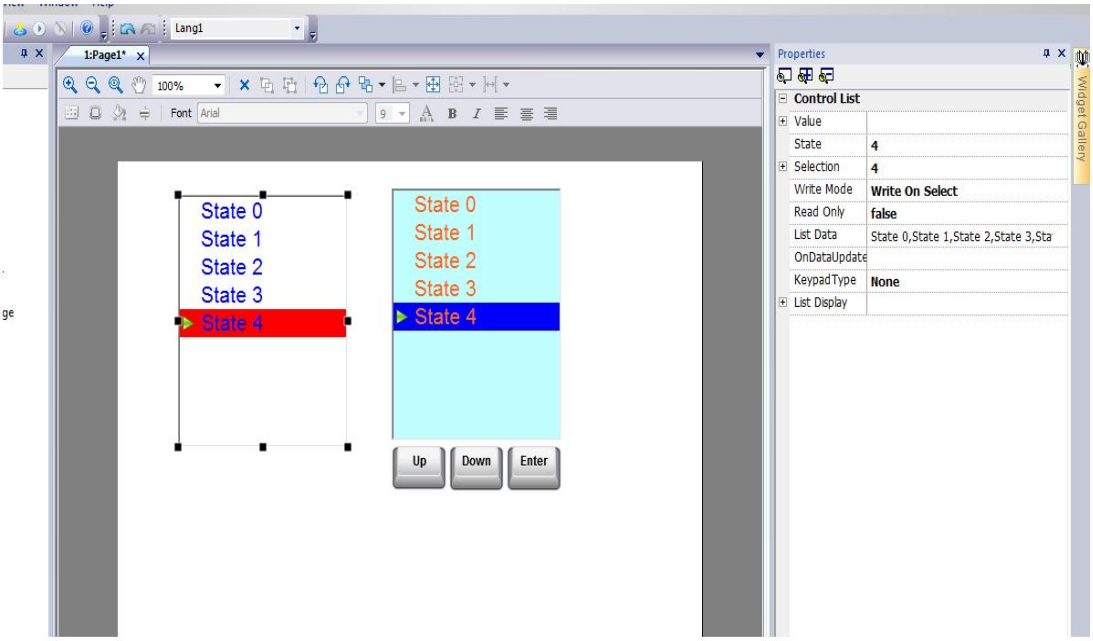

Figure 213

16.3.1 State

States are added by selecting Add/Remove List Items from the List Data voice 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.

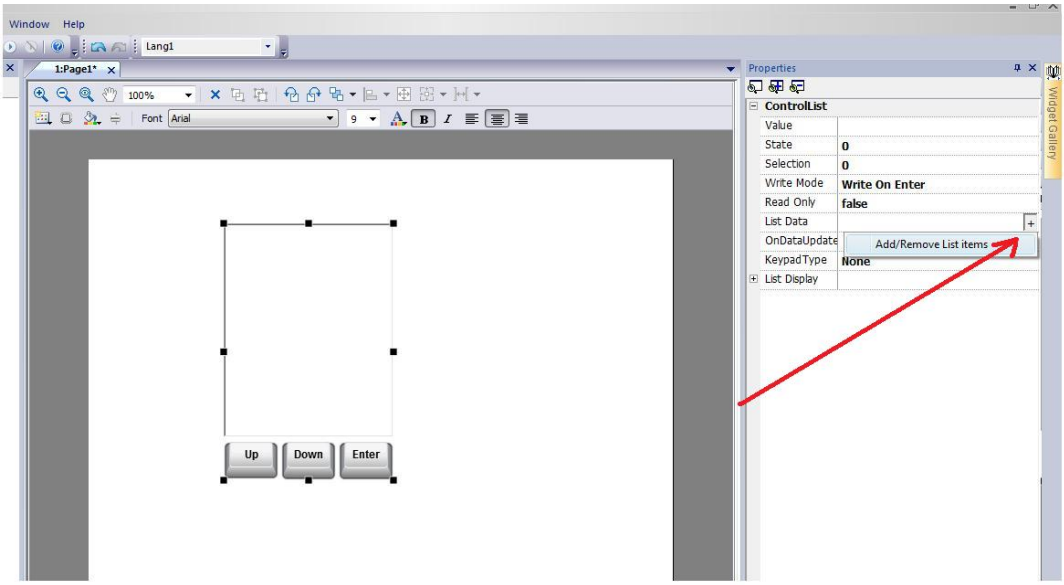

Figure 214

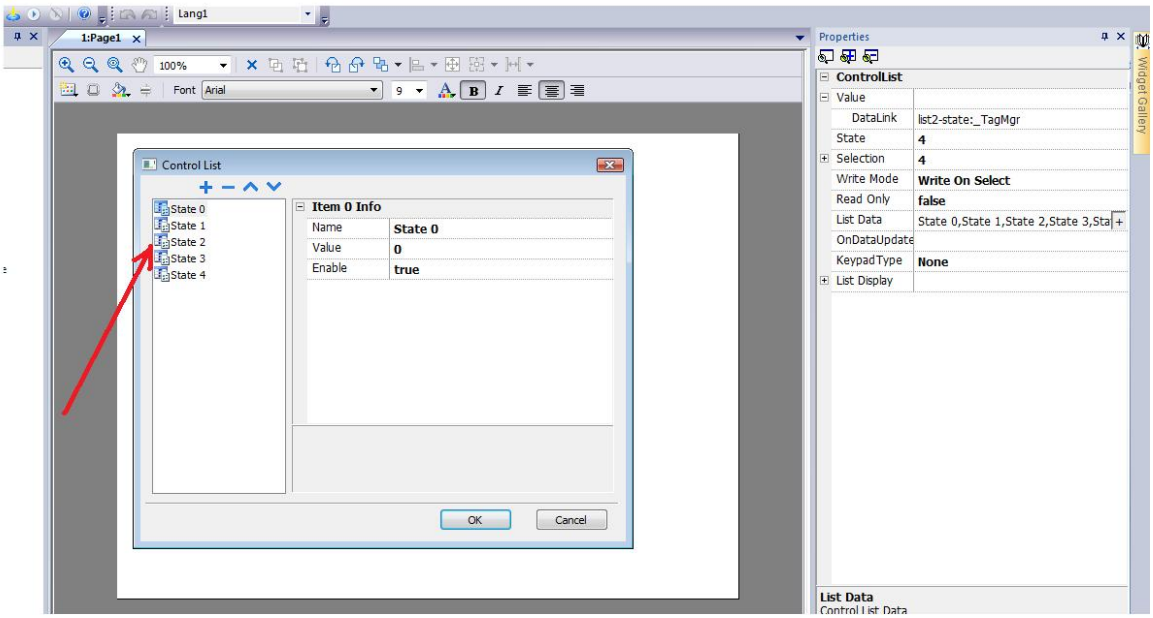

Figure 215

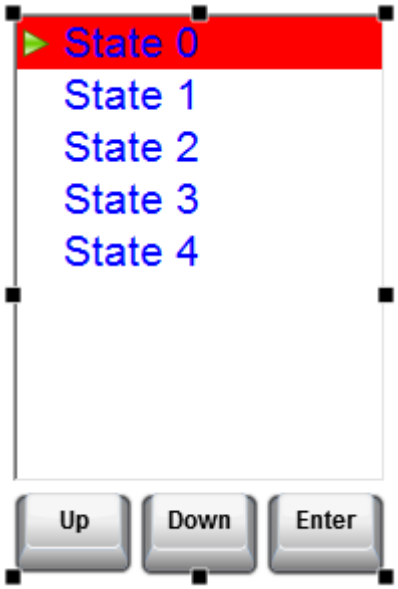

Figure 216

16.3.2 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 write modes for the control list: Write on Select and Write on Enter

16.3.3 Write on Select

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

16.3.4 Write on Enter

On Write on Enter you need first to select the state, and then press the enter key to write the status value to the Tag.

16.4 Variables Widget

The Variables widget is available in the advanced category under the Data Sources sub-category as shown in the following figure.

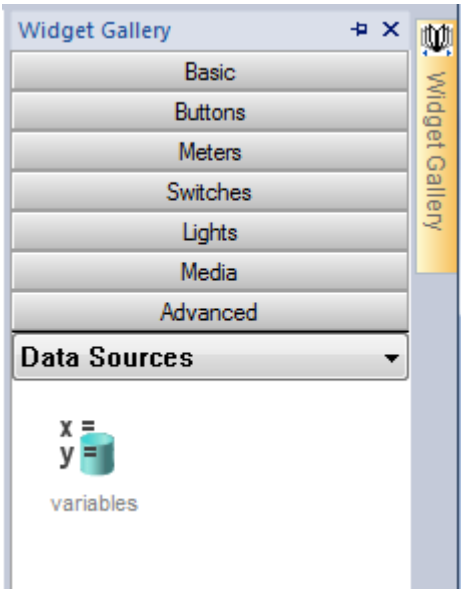

Figure 217

The purpose of variable widgets is to have some internal variables that can be used for certain operations such as data transfer or use in JavaScript. The variables are local to the page where the widget is inserted.

To insert the widget in page, just drag and drop it to any position on the page. This will display a placeholder to remember the widget is there, but it will not be visible at runtime.

We can create some variables and give some values in VisuControl as shown in the following figure.

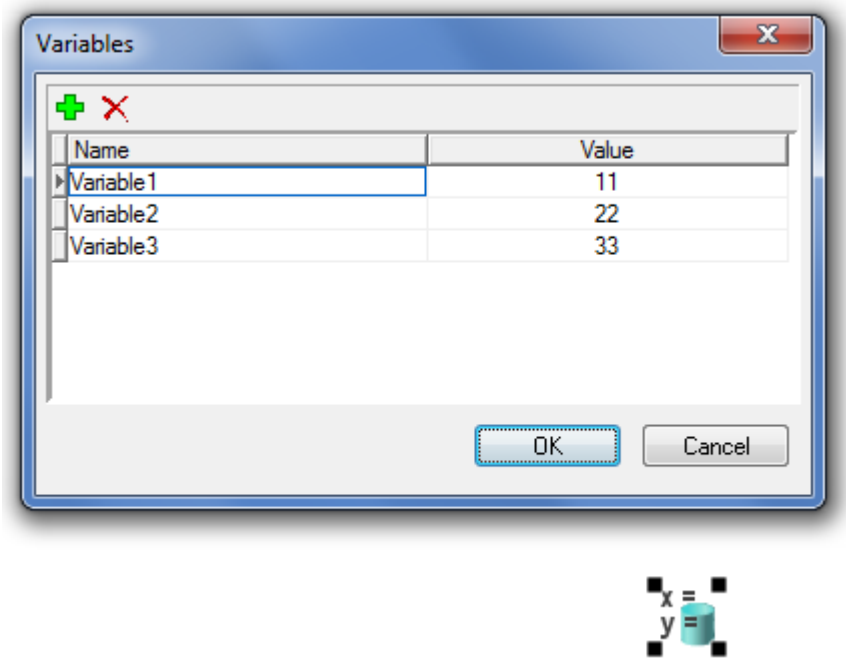

Figure 218

The configured Variables can be referenced from the Attach tag dialog once you click on the Widget source as showed in the following figure.

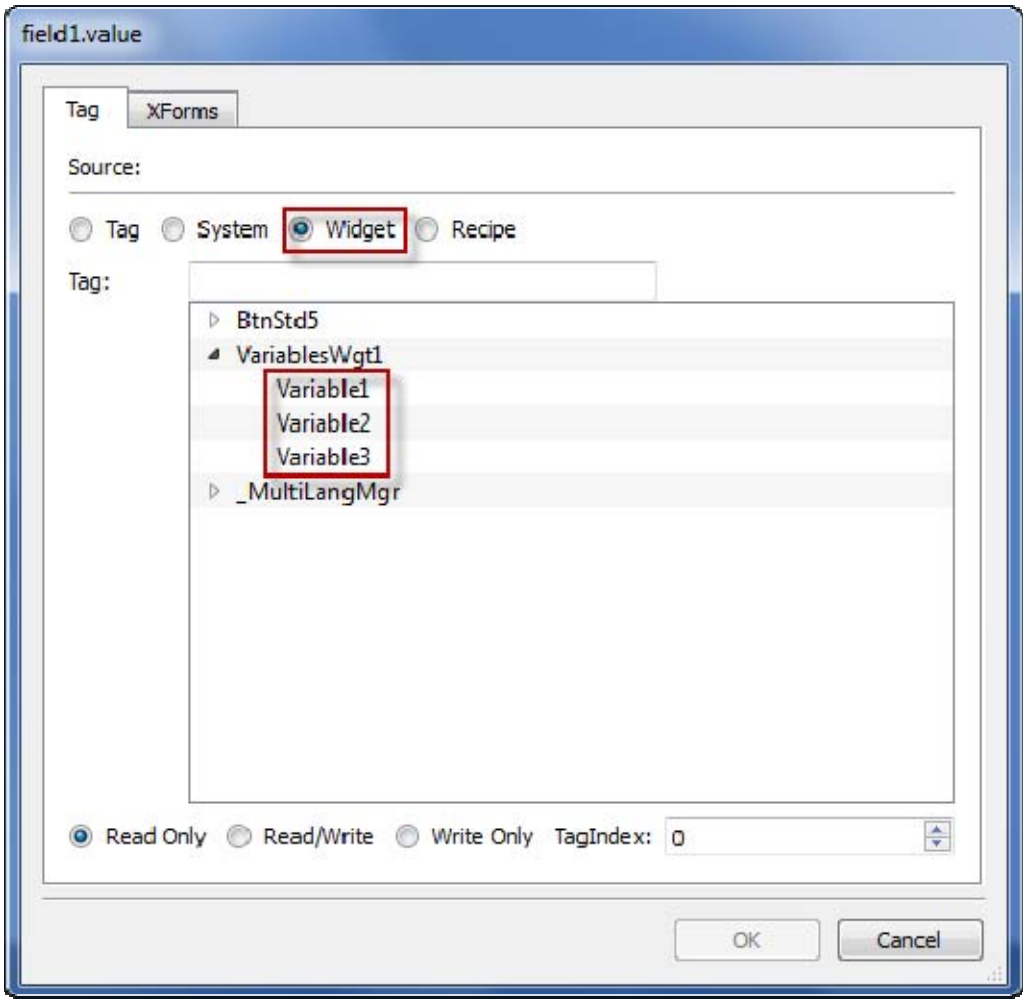

Figure 219

In case you need global variables, they can be configured from the project widget adding the desired variables to the global Variable Widget as shown in the following figure.

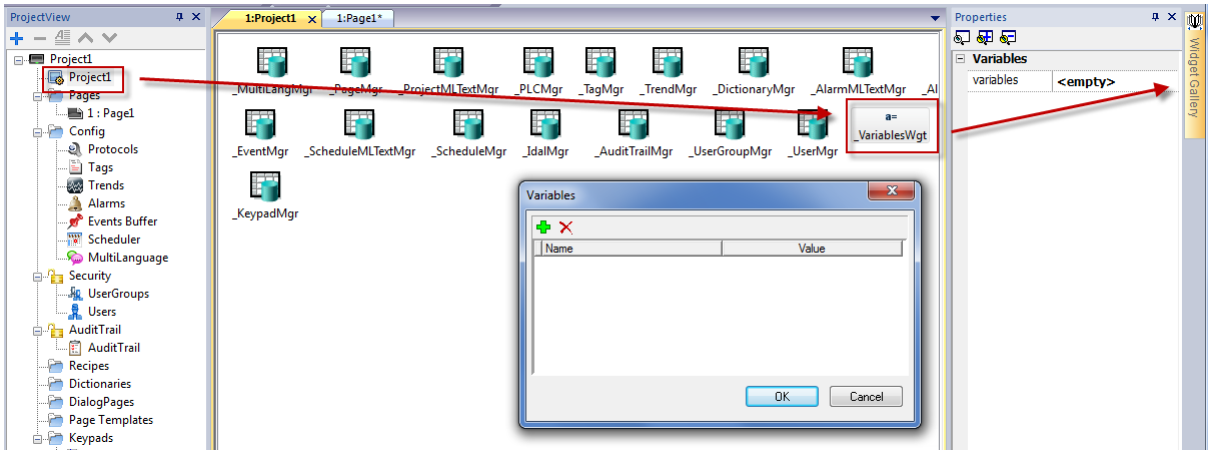

Figure 220

16.4.1 Using Variables in JavaScript

The Variables defined according to the explanation above can be also referenced in JavaScript with the following syntax:

For Local Variables

```
var varWgt = page.getWidget("_VariablesWgt");
var compVar = varWgt.getProperty("VariableName");
```

For Global Variables

```
var varWgt = project.getWidget("_VariablesWgt");
var compVar = varWgt.getProperty("VariableName");
```

16.5 Weather widget

The Weather widget is available in the media category of the widget gallery.

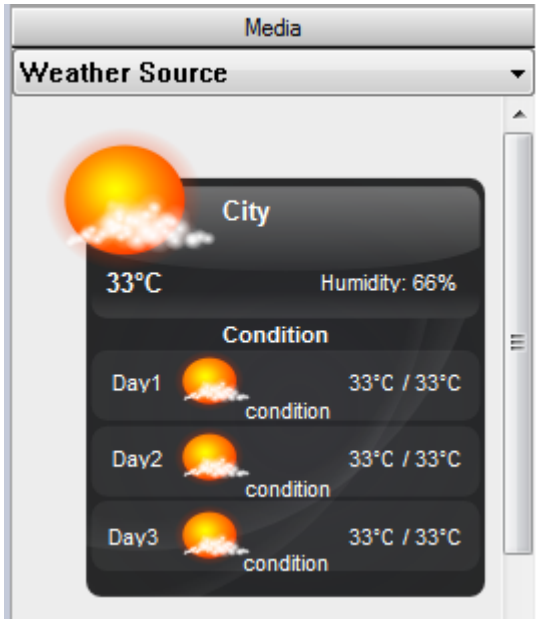

Figure 221

The widget can be used to retrieve from the Internet the weather conditions and display them on the screen. The weather widget uses the Google API to connect to the meteo services provided by Google.

To use the widget just drag and drop it to a project page.

Note: the weather information to be updated requires the panel to be connected to a network with access to a working internet connection

The widget properties are shown in the following figure.

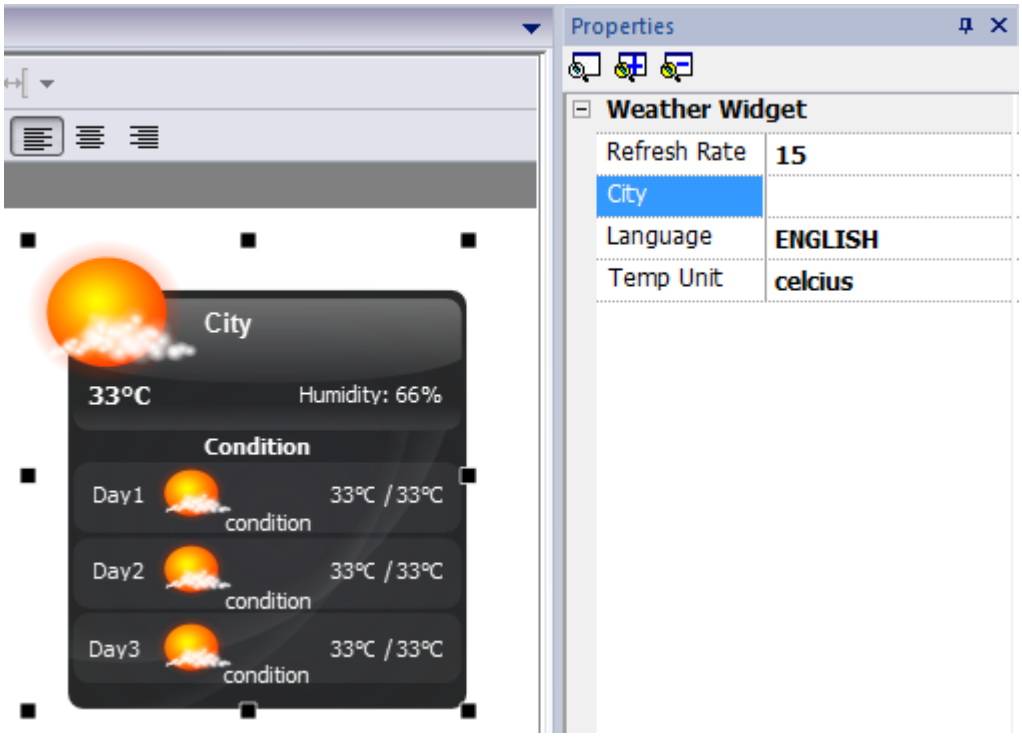

Figure 222

City — The widget supports all the cities from the Google database; you can specifiy a default city in the format of “city, country”.
City can be changed at runtime simply editing the field by touching it.

Refresh Rate — The refresh rate is the interval of time in minutes used by the widget to retrieve the meteo information.

Temp Unit — This property allows to specify the units to be used for the temperature visualization.

Language — This property specifies the language used by the interface. The supported languages are directly coming from the Google API. To change language you need to configure in page the dedicated combo box available at the bottom of the weather source category as shown in the following figure.

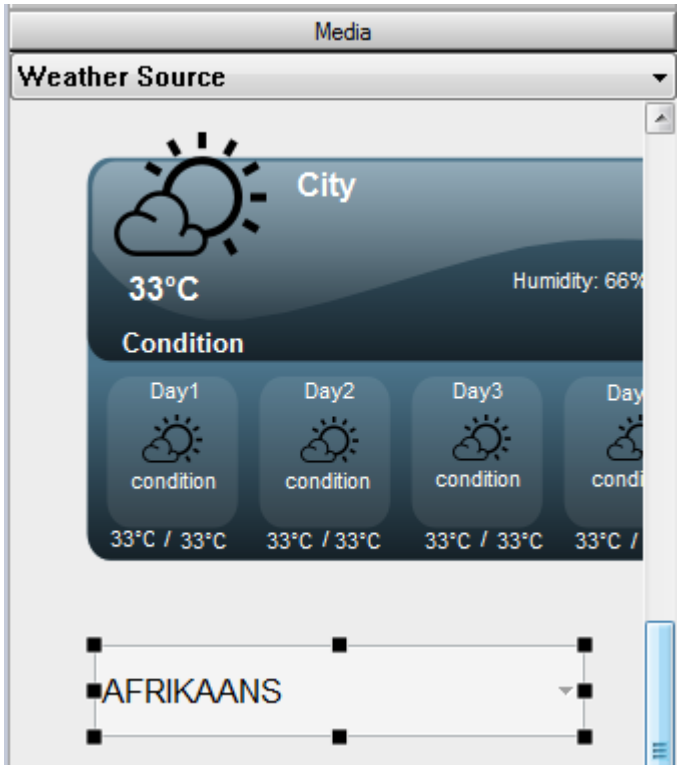

Figure 223

The “language” property of the weather widget has to be linked then to the Data property of the combo box as shown in the following figure.

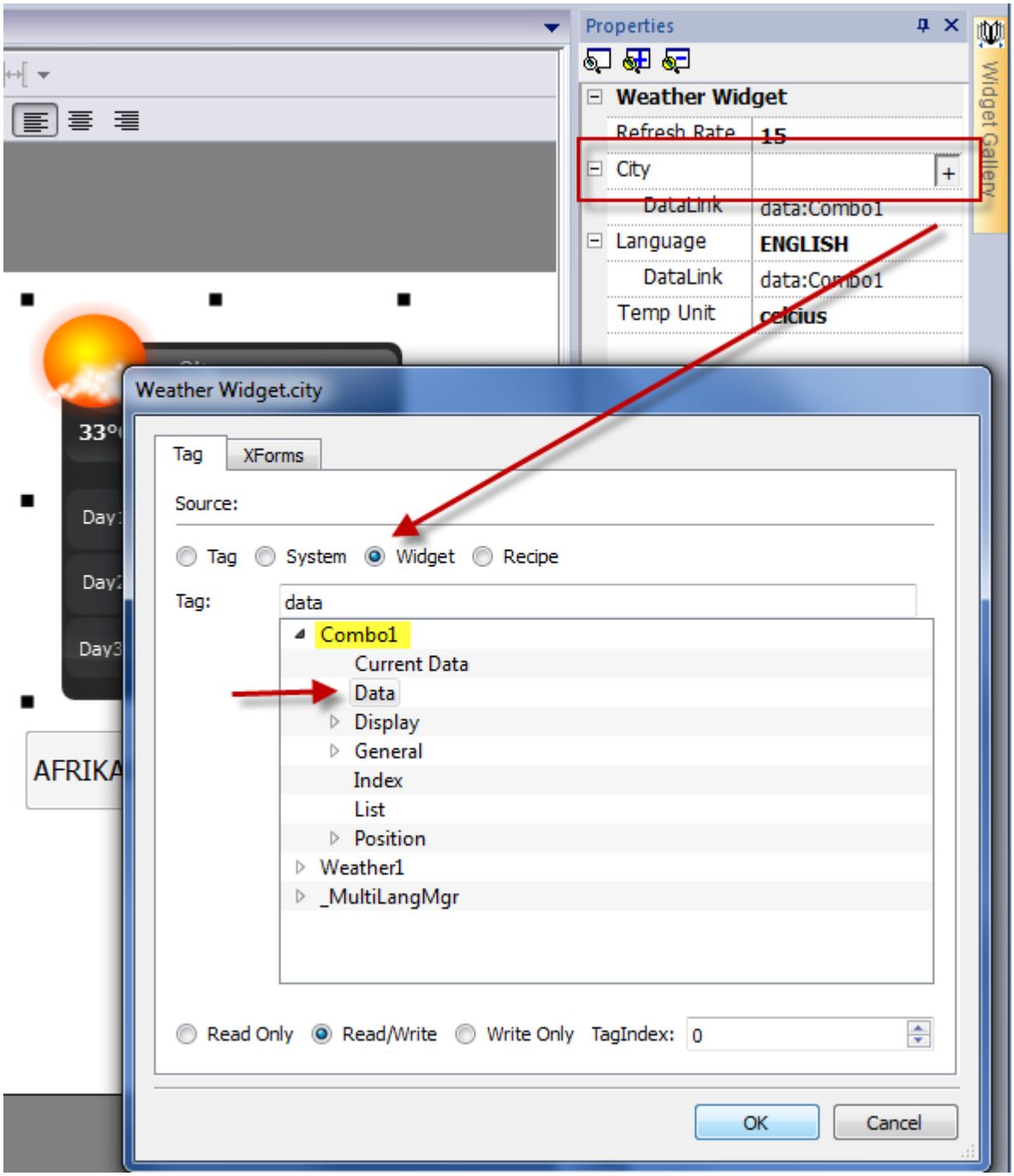
Properties
Weather Widget
Refresh Rate
15
City
DataLink
data:Combo1
Language
ENGLISH
DataLink
data:Combo1
Temp Unit
celcius
Widget Gallery
Weather Widget.city
Tag
XForms
Source:
Tag
System
Widget
Recipe
Tag:
data
Combo1
Current Data
Data
Display
General
Index
List
Position
Weather1
_MultiLangMgr
Read Only
Read/Write
Write Only
TagIndex:
0
OK
Cancel
AFRIKA

Figure 224

17 Working with Custom Widgets in VisuControl

VisuControl includes a large library that 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 is 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 a group.

2: Right click on the group, and select "Convert To Widget" from the context menu.

A Conversion to Widget dialog box is displayed below.

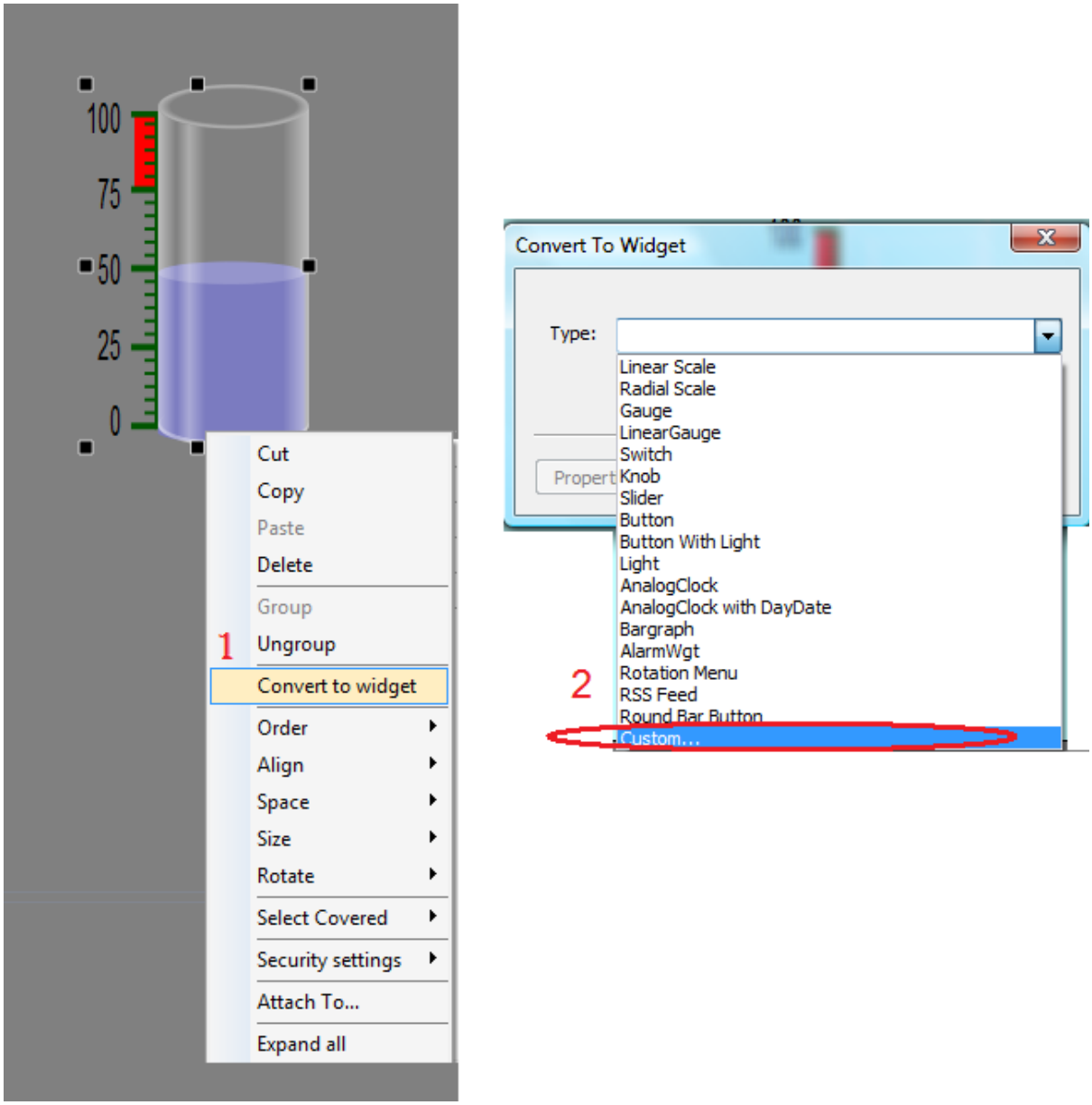

Figure 225

At this point, you can select existing custom widget types (such as Knobs, Button With 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.

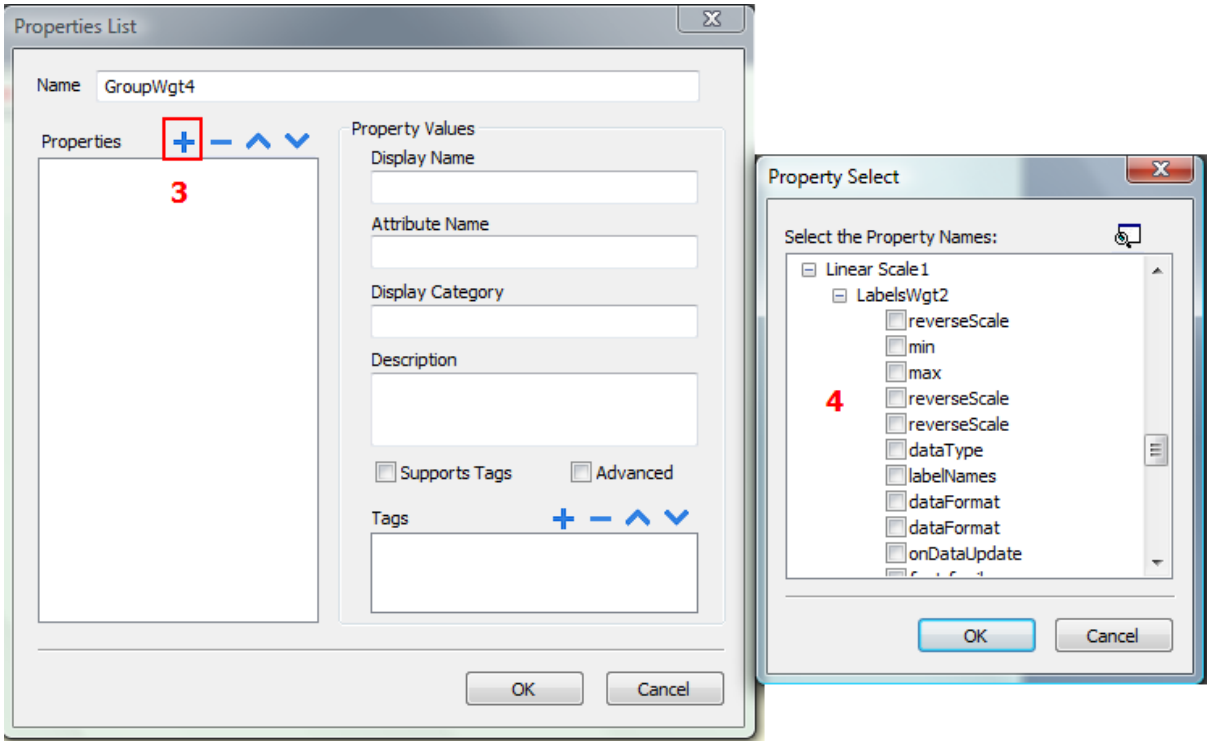

Figure 226

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 VisuControl, 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 assign 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 box 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 below figure.

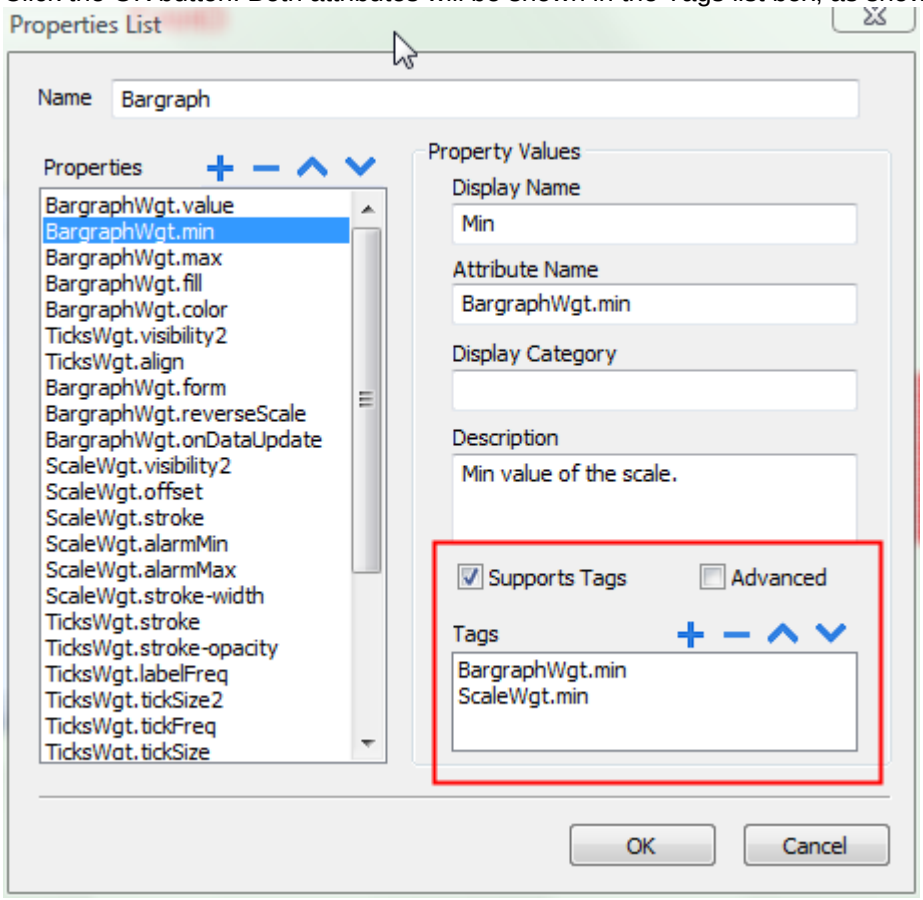

Figure 227

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 box.

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 VisuControl

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.

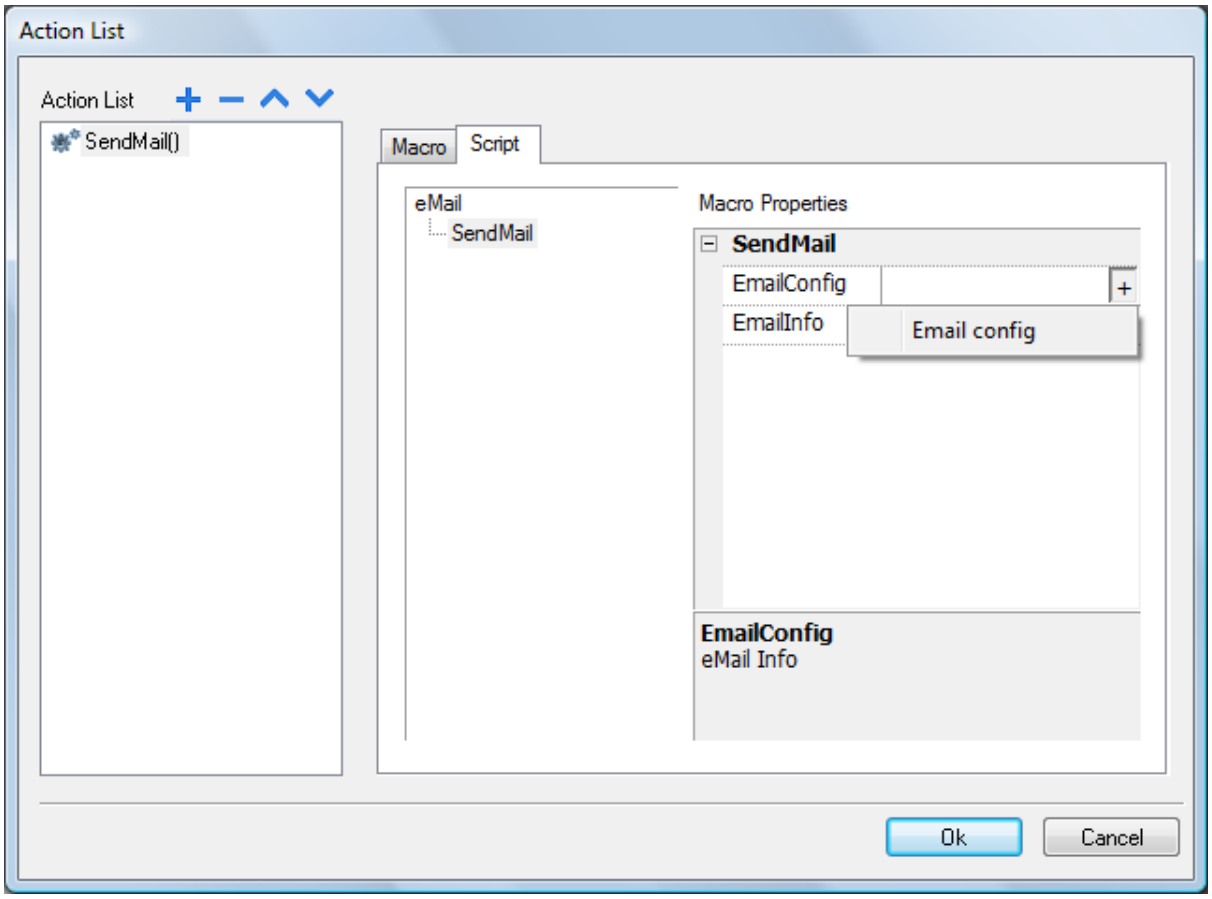

Figure 228

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.

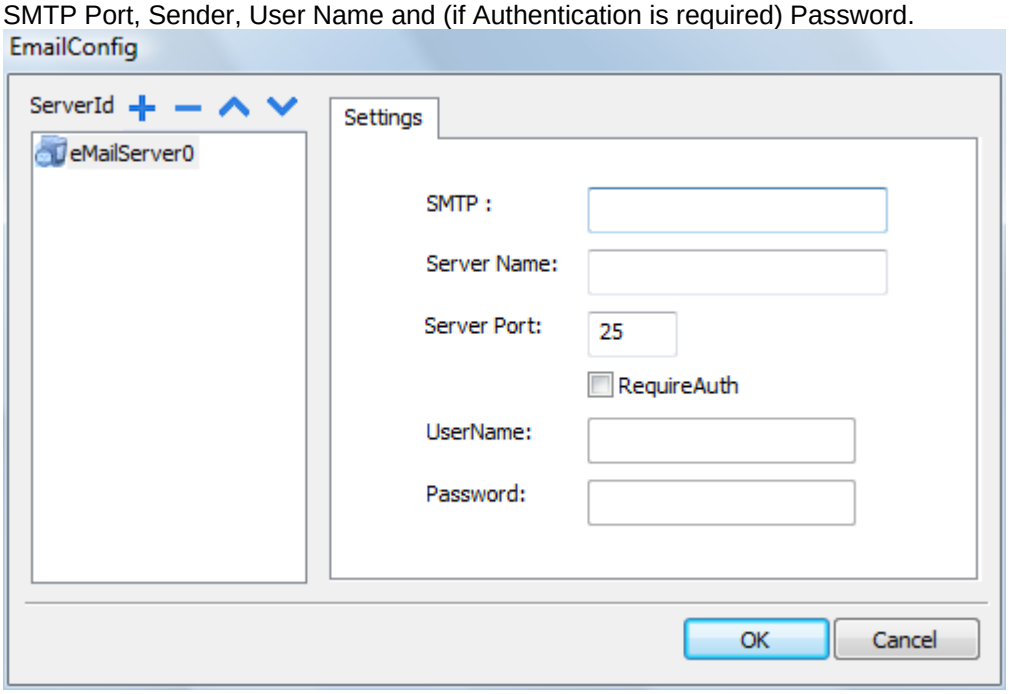

Figure 229

Note: You can add more E-mail servers by clicking the “+" button n 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 semicolon “;".

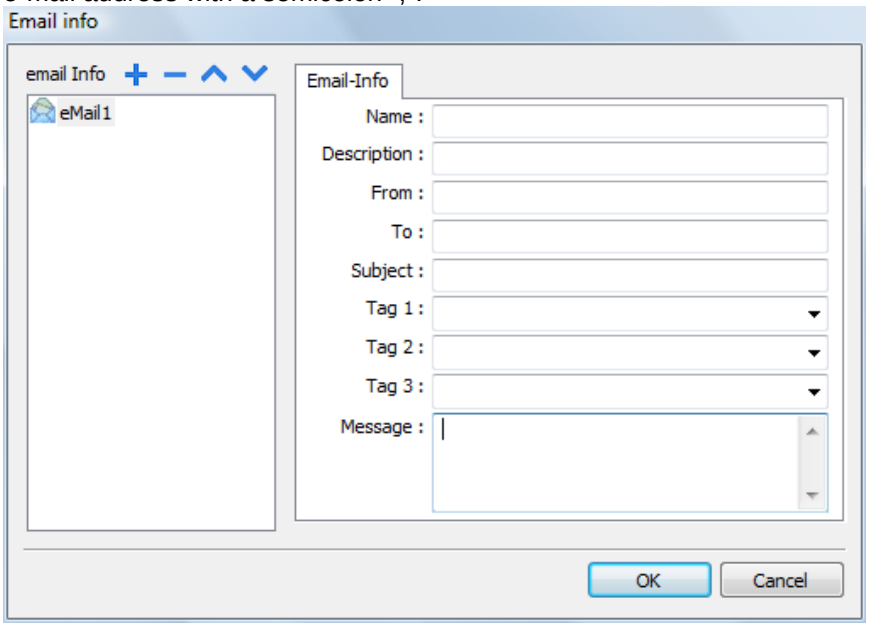

Figure 230

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. The “Tag” information that follows the “@” symbol represents the index of the tag as per the configuration made. In the example below, the “Temperature” tag has index 2; to insert the value of the “Temperature” tag in the mail message body, use the “@2” syntax.

An example is shown in the figure below.

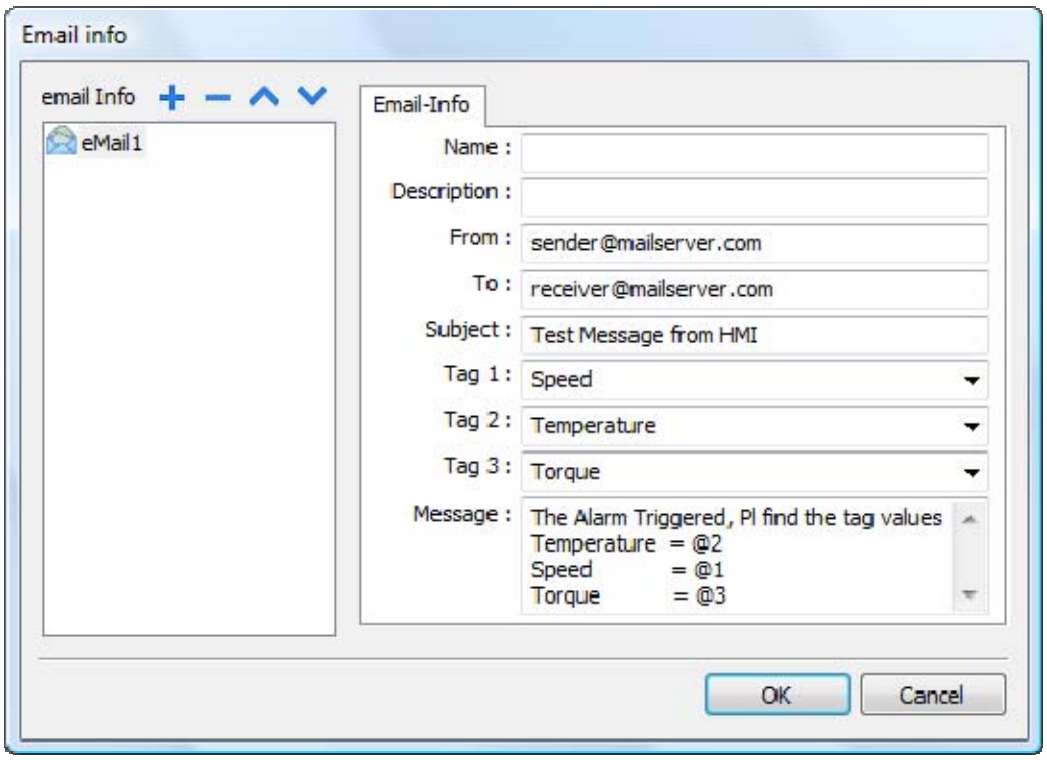

Figure 231

18.5 Limitation

A maximum number of 3 tags are supported in each e-mail message body.

19 Working with JavaScript in VisuControl

The purpose of this chapter is to describe the JavaScript interface implemented in VisuControl JavaScript is based on the ECMAScript programming language http://www.ecmascript.org , as defined in standard ECMA-262. Microsoft Chakra and Firefox SpiderMonkey JavaScript engines support the ECMAScript standard. If you are familiar with JavaScript, you can use the same type of commands in VisuControl 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/4.7/ecmascript.html

https://developer.mozilla.org/en/JavaScript

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

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, but they are only executed when the given event occurs. This approach minimizes the overhead required to execute logic on the panel.

VisuControl provides a JavaScript engine running at client side. Each project page can contain scripts 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 JavaScript in the following events:

Widget Events
Page Events
System Events

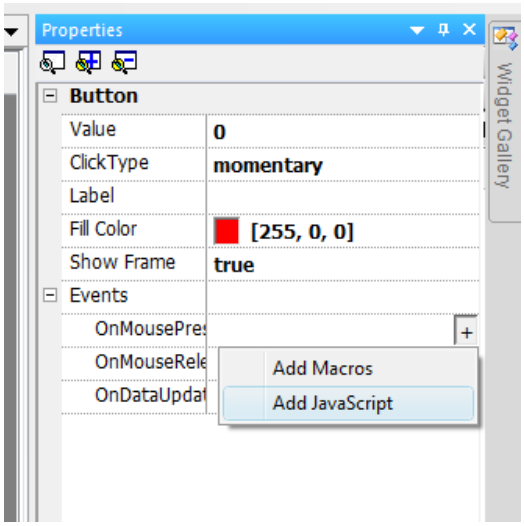

Figure 232

19.2.1 Widget Events

1) onMousePress
void onMousePress(me, eventInfo)

This event is available only for buttons and it occurs when the button is pressed.

Parameters
me The object that triggers the event.
eventInfo It is reserved for future enhancements.

```
function buttonStd1_onMousePress(me, eventInfo) {
    //do something…
}
```

2) onMouseRelease
void onMouseRelease (me, eventInfo)

This event is available only for buttons and it occurs when the button is released.

Parameters
me The object that triggers the event.
eventInfo It is reserved for future enhancements.

```
function buttonStd1_onMouseRelease(me, eventInfo) {
    //do something…
}
```

3) onDataUpdate
boolean onDataUpdate (me, eventInfo)

This occurs when the data attached to the Widget changes.

Parameters
me The object that triggers the event.
eventInfo An object with these fields (you can refer fields using "." - dot notation):
oldValue: The old value that is the widget value before the change.

newValue: The new value that is the value which will be updated to the widget.
attrName: The attribute on which the event is generated
index: An integer attribute index if any, default = 0
mode: W when user is writing to the widget, R otherwise.
This event is triggered by the system before 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.

This method is always called the first time you enter a page.

```
function buttonStd1_onDataUpdate(me, eventInfo) {
  if ( eventInfo.oldValue < 0) {
      //do something…
  }
  return false;
}
```

19.2.2 Page Events

4) onActivate
void onActivate(me, eventInfo)

This event occurs each time the page is shown.

Parameters
me The object that triggers the event.
eventInfo It is reserved for future enhancements.
This JavaScript will execute when the page is Active. It means that, when the page is loaded, the script will execute.

```
function Page1_onActivate(me, eventInfo) {
    //do something…
}
```

5) onDeactivate
void onDeactivate(me, eventInfo)

This occurs when leaving the page.

Parameters
me The object that triggers the event.
eventInfo It is reserved for future enhancements.

```
function Page1_onDeactivate(me, eventInfo) {
    //do something…
}
```

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 below figure.

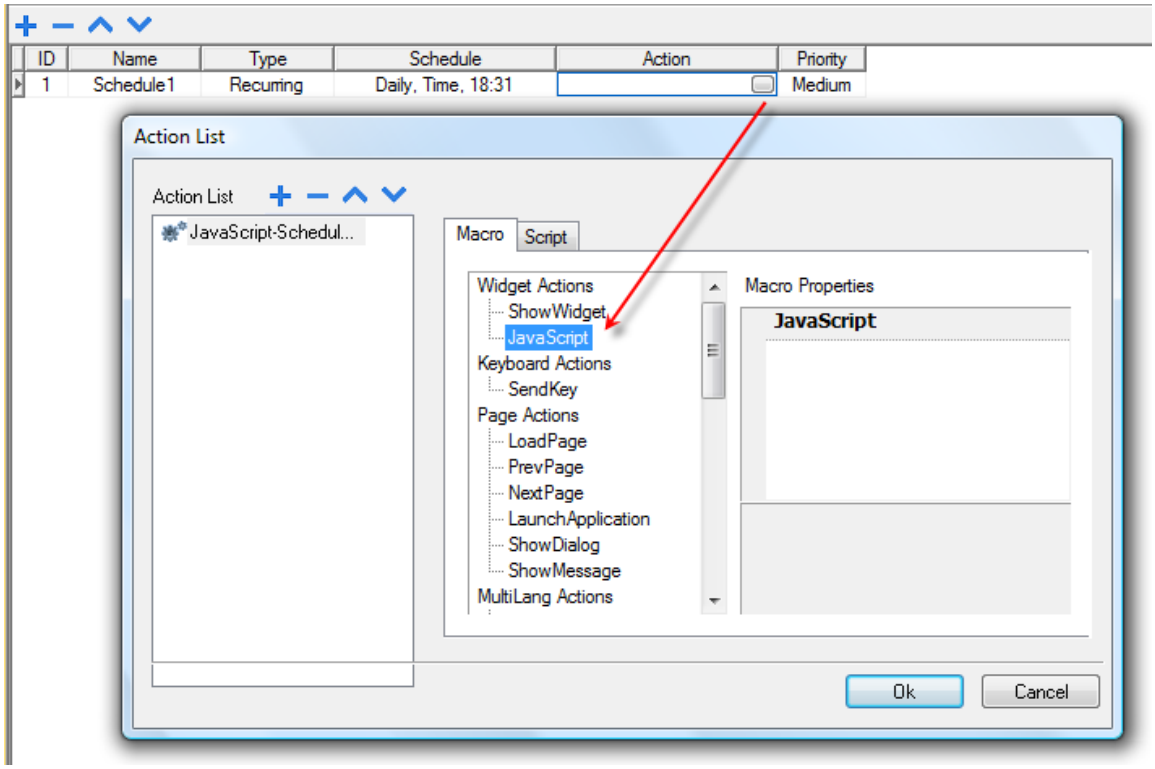

Figure 233

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

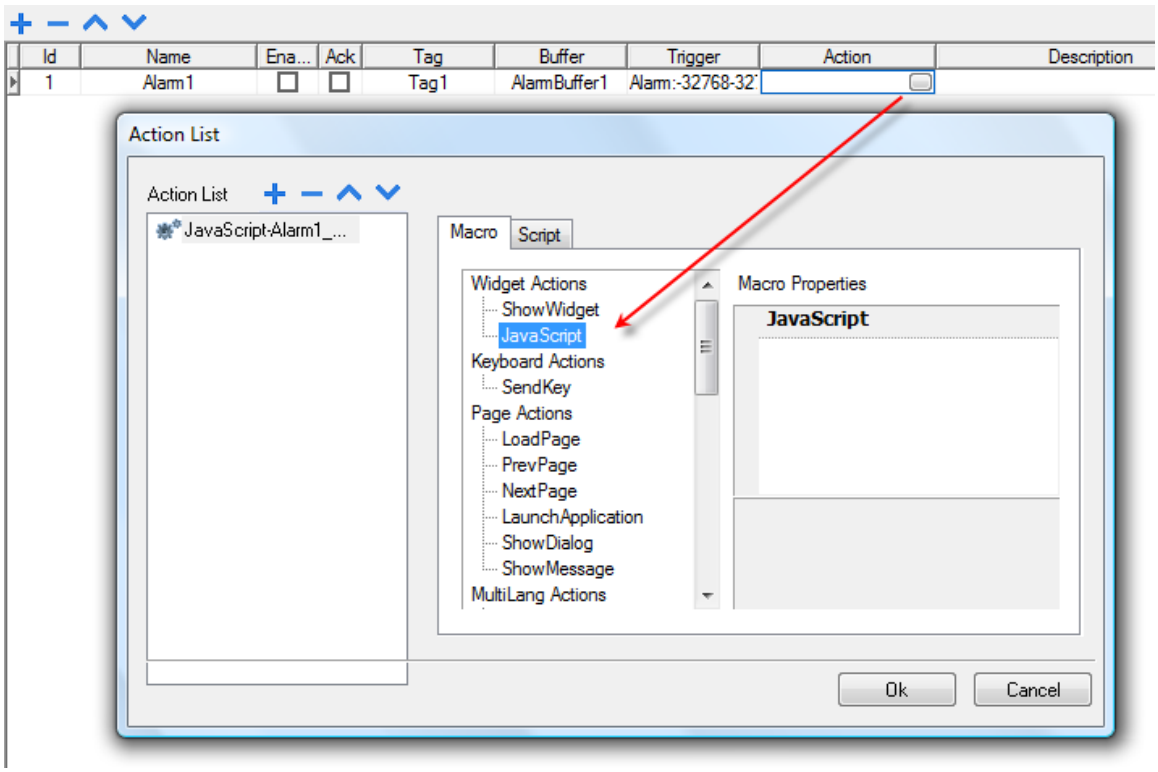

Figure 234

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 below figure.

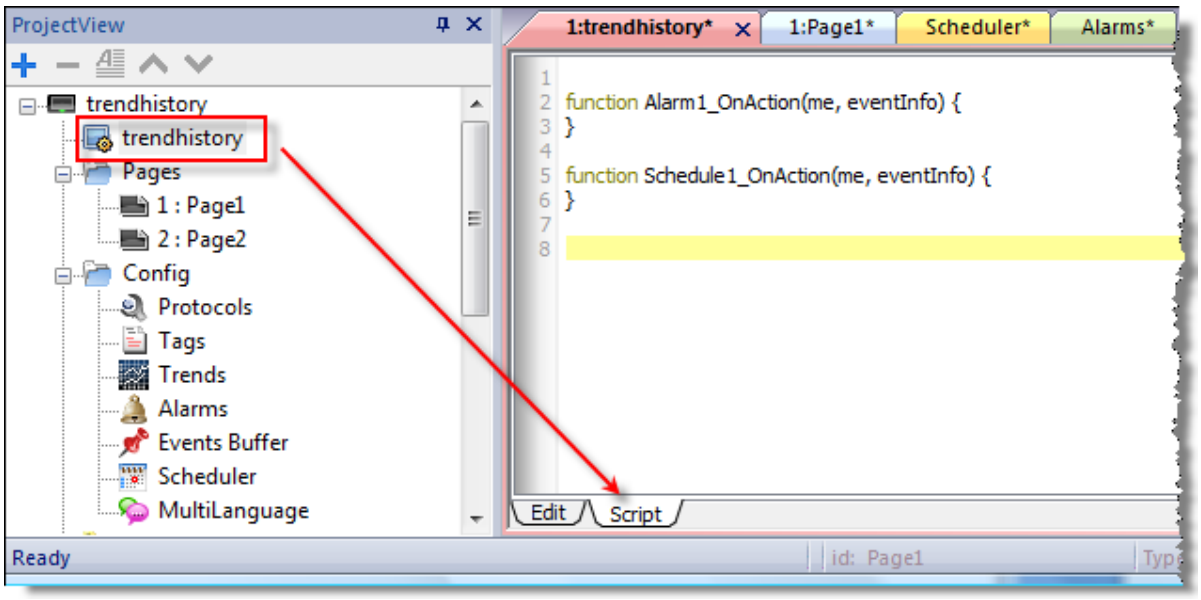

Figure 235

19.3 Language Reference

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

19.3.1 Objects

VisuControl 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

The following properties are common among all widgets.
Important! When you change the properties of widgets with JavaScript you have to set the page **Static Optimization** to **false** or set the widget **Static Optimization** to **Dynamic**, otherwise changes to properties will be ignored. You can find the option **Static Optimization** in the **Advance Properties Page**.

objectName

string objectName

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

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

x

number x

It gets or sets the Widget ‘x’ position in pixels.

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

y

number y

It gets or sets the Widget ‘y’ position in pixels.

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

width

number width

It gets or sets the Widget width in pixels.

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

height

number height

It gets or sets the Widget height in pixels.

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

visible

boolean visible

It gets or sets the Widget visible state.

```
function btnStd4_onMouseRelease(me) {
    var wgt = page.getWidget("rect1");
    wgt.visible = false;
}
function btnStd5_onMouseRelease(me) {
    var wgt = page.getWidget("rect1");
    wgt.visible = true;
}
```

value

number value

It gets or sets the Widget value.

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

opacity

number opacity (range from 0 to 1)

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

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

rotation

number rotation (in degrees)

It 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.

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

userValue

string userValue

It gets or sets a user-defined value for the Widget. This field can be used by JavaScript functions to store additional data with the Widget.

```
function btnStd9_onMouseRelease(me) {
    var wgt = page.getWidget("rect1");
    wgt.userValue = "Here I can store custom data";
}
```

Every widget has some specific properties that you can access using dot notation. For an up-to-date and detailed list of properties you can use the Qt Script Debugger inspecting the widget methods and properties.

19.3.1.3 Methods

getProperty

object getProperty(propertyName, [index])

Returns a property

Parameters

propertyName A string containing the name of property to get.
index The index of the element to get from the array. Default is 0.

Almost all properties that is shown in the VisuControl Property view can be retrieved from the getProperty method. The index value is optional and only used for Widgets that support arrays.

```
function buttonStd1_onMouseRelease(me, eventInfo) {
    var shape = page.getWidget("rect2");
    var y_position = shape.getProperty("y");
}
```

```
function buttonStd2_onMouseRelease(me, eventInfo) {
    var image = page.getWidget("multistate1");
    var image3 = image.getProperty("imageList", 2);
    //…
}
```

setProperty

boolean setProperty(propertyName, value, [index])

Sets a property for the Widget

Parameters

propertyName A string containing the name of property to set.
value A string containing the value to set the property.
index The index of the element to set in the array. Default is 0.

Almost all properties that are shown in the VisuControl Property view can be set by this method. The index value is optional and only used for Widgets that support arrays (for example a MultiState Image widget). The setProperty method returns a boolean value true or false to indicate if the property was set or not.

```
function buttonStd1_onMouseRelease(me, eventInfo) {
    var setting_result = shape.setProperty("y", 128);
    if (setting_result)
        alert("Shape returned to start position");
}
```

```
function buttonStd2_onMouseRelease(me, eventInfo) {
    var image = page.getWidget("multistate1");
    var result = image.setProperty("imageList", "Fract004.png", 2);
    //…
}
```

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

string backgroundColor (in format rgb(xxx, xxx, xxx) where xxx range from 0 to 255)

The page background color

```
function btnStd11_onMouseRelease(me) {
    page.backgroundColor = "rgb(128,0,0)";
}
```

width

number width

The Page width in pixels

```
function btnStd05_onMouseRelease(me) {
    var middle_x = page.width / 2;
}
```

height

number height

The Page height in pixels

```
function btnStd05_onMouseRelease(me) {
    var middle_y = page.height / 2;
}
```

userValue

string userValue

It gets or sets a user-defined value for the Widget. This field can be used by JavaScript functions to store additional data with the page.

```
function btnStd9_onMouseRelease(me) {
    page.userValue = "Here I can store custom data";
}
```

19.3.1.6 Methods

getWidget

object getWidget(wgtName)

It returns the Widget with the given name.

Parameters

wgtName A string containing the name of widget

Return value

An object representing the widget. If the widget does not exist, null is returned.

```
function btnStd1_onMouseRelease(me) {
    var my_button = page.getWidget("btnStd1");
}
```

setTimeout

number setTimeout(functionName, delay)

It starts a timer to execute a given function after a given delay once.

Parameters

functionName	A string containing the name of function to call.
delay	The delay in milliseconds.

Return value

It returns a number corresponding to the timerID.

```
var duration = 3000;
var myTimer = page.setTimeout("innerChangeWidth()", duration);
```

clearTimeout

void clearTimeout(timerID)

It stops and clear the timeout timer with the given timer.

Parameters

timerID	The timer to be cleared and stopped.

```
var duration = 3000;
var myTimer = page.setTimeout("innerChangeWidth()", duration);
// do something
page.clearTimeout(myTimer);
```

setInterval

number setInterval(functionName, interval)

It starts a timer that executes the given function at the given interval.

Parameters

functionName	A string containing the name of function to call.
interval	The interval in milliseconds.

Return value

It returns a number corresponding to the timerID

```
var interval = 3000;
var myTimer = page.setInterval("innerChangeWidth()", interval);
```

clearInterval

void clearInterval(timerID)

It stops and clears the interval timer with the given timer.

Parameters

timerID	The timer to be cleared and stopped.

```
var interval = 3000;
var myTimer = page.setInterval("innerChangeWidth()", interval);
// do something
page.clearInterval(myTimer);
```

clearAllTimeouts

void clearAllTimeouts()

It clears all the timers started.

```
Page.clearAllTimeouts();
```

19.3.1.7 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.

19.3.1.8 Methods

getTag

object getTag(TagName)

Gets the tag specified by TagName from the group object.

Parameters
TagName A string representing the tag name.

Return value
An object that is the value of the tag or if tag value is an array it returns the complete array. If you need to retrieve an element of the array, check the method getTag available in object Project. undefined is returned if tag is invalid.

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

getCount

number getCount()

Returns total number of tags in this group.

Return value
The number of tags.

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

getTags

object getTags()

Returns the list of all tags in group.

Return value
An array of all tags in the group.

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

19.3.1.9 Project Object

This object defines the project widget. The project widget is used to retrieve data about the project such as tags, alarms, recipes, schedules, tags and so on. There is only one widget for the project and it can be referenced through the project variable.

19.3.1.10 Properties

startPage

string startPage

The page shown when the application is started

```
var startPage = project.startPage;
project.startPage = "Page2.jmx";
```

19.3.1.11 Methods

nextPage

void nextPage()

The script executes the next page macro.

```
project.nextPage();
```

prevPage

void prevPage()

The script executes the Previous page macro.

```
project.prevPage();
```

homepage

void homePage()

The script executes the Home page macro.

```
project.homePage();
```

loadPage

void loadPage(pageName)

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

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

showDialog

void showDialog(pageName)

The script executes to show the dialog page.

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

closeDialog

void closeDialog()

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

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

showMessage

void showMessage(message)

The script executes to display the message popup.

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

getGroup

number getGroup(groupName, groupInstance, [callback])

Fast read method which gets the values of all tags in a group.

Parameters

groupName A string containing the name of the group.
groupInstance The group element to be filled.
callback A string containing the name of the function to be called when the group is ready.

Return value

A number value that is the status: 1 for success, 0 for fail.

```
var group = new Group();
var status = project.getGroup ("enginesettings", group);
if (status == 1) {
    var value = group.getTag("Tag1");
    if (value!=undefined) {
        // do something with the value
    }
}
```

```
var g = new Group();
var status = project.getGroup ("enginesettings", g, "fnGroupReady");
function fnGroupReady(groupName, group) {
    var val = group.getTag("Tag1");
    if (val!=undefined) {
        // do something with the value
    }
}
```

getTag

object getTag(tagName, state, index, [callback])

It returns the tag value or the complete array if index value is -1 of the given tagName.

Parameters

tagName A string of the tag name.
state The state element to be filled.
index An index if the tag is array type. -1 returns the complete array. Default is 0.
callback function name if an asynchronous 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.

```
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

number setTag(tagName, tagValue, index)

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

Parameters

tagName	A string of the tag name.
tagValue	An object containing the value to write.
index	An index if tag is array type. Set -1 to pass complete array. Default is 0.

Return value

Interger value for denoting success and failure of action. A ‘0’ means success and ‘-1’ means failure.

```
var val = [1,2,3,4,5];
var status = project.setTag("Tag1", val, -1);
if (status == 0) {
    // Success
} else {
    // Failure
}
```

```
var val = "value";
var status = project.setTag("Tag1", val);
if (status == 0) {
    // Success
} else {
    // Failure
}
```

getRecipeItem

object getRecipeItem (recipeName, recipeSet, recipeElement)

Gets the value of the given recipe set element.

Parameters

recipeName	A string representing the recipe name.
recipeSet	A string representing the recipe set, can be either the recipe set name or 0 based set index.
recipeElement	A string representing the recipe Element, can be either the element name or 0 based element index.

Return value

An object with the value of the recipe. undefined is returned if invalid. If of type array, an array object type is returned.

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

setRecipeItem

number setRecipeItem (recipeName, recipeSet, recipeElement, value)

Gets the value of the given recipe set element.

Parameters

recipeName A string representing the recipe name.
recipeSet A string representing the recipe set, can be either the recipe set name or 0 based set index.
recipeElement A string representing the recipe Element, can be either the element name or 0 based element index.
value An object containing the value to store in the recipe. It can be an array type too.

Return value

Interger value for denoting success and failure of action. A '0' means success and '-1' means failure.

```
var val = [2,3,4];
project.setRecipeItem("recipeName", "Set", "Element", val);
if (status == 0) {
    // Success
} else {
    // Failure
}
```

downloadRecipe

void downloadRecipe (recipeName, recipeSet)

Downloads the recipe set to corresponding tag.

Parameters

recipeName A string representing the recipe name.
recipeSet A string representing the recipe set, can be either the recipe set name or 0 based set index.

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

uploadRecipe

void uploadRecipe (recipeName, recipeSet)

Uploads the value of tags into the provided recipe set.

Parameters

recipeName A string representing the recipe name.
recipeSet A string representing the recipe set, can be either the recipe set name or 0 based set index.

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

19.3.1.12 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.

19.3.1.13 Methods

getQualityBits

number getQualityBits()

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

Return value
A number containing the quality bits.

```
var state = new State();
var value = project.getTag("Tag1", state, 0);
var qbits = state.getQualityBits();
```

getTimestamp

number getTimestamp()

Returns time the value was sampled.

Return value
A number containing the timestamp (for example 1315570524492).

Remarks
Date is a native JavaScript data type.

```
var state = new State();
var value = project.getTag("Tag1", state, 0);
var ts = state.getTimestamp();
```

isQualityGood

boolean isQualityGood()

It returns whether value contained within this State object is reliable.

Return value
A Boolean true if quality is good, false otherwise.

```
var state = new State();
var value = project.getTag("Tag1", state, 0);
if (state.isQualityGood()) {
    // do something…
}
```

19.3.2 Keywords

Global objects are predefined and always available objects that can be referenced by the names listed below.

page

object page

It references the page object for the current page.

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

project

object project

It references the project Widget.

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

19.3.3 Global Functions

print

void print(message)

It prints a message to the HMI Logger window.

Parameters

message A string containing the message to display.

```
print("Test message");
```

alert

void alert(message)

It 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.

Parameters

message A string containing the message to display.

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

```
alert("Test message");
```

19.4 Debugging of JavaScript

You can enable or disable the debugging mode to find, to solve the script errors or to inspect the properties and methods of elements. To enable the debugging mode, in the **Advanced Properties Page**, set **JavaScript Debug** to **True** as shown in the below figure.

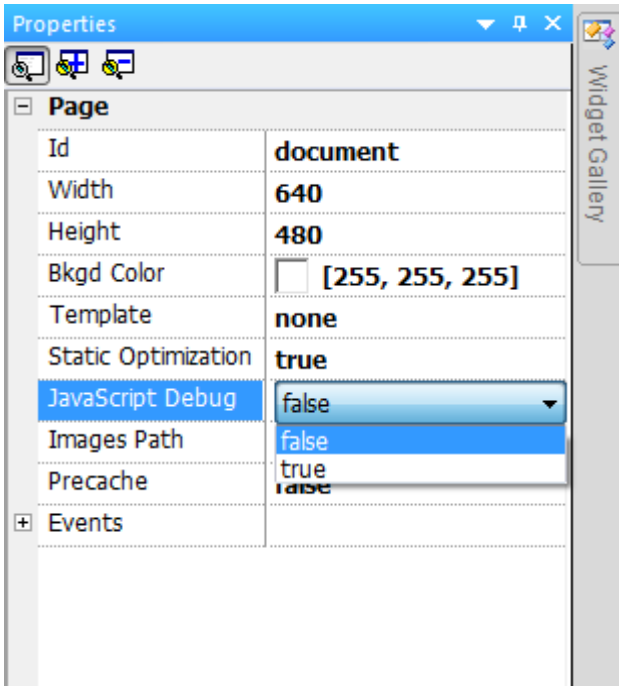

Figure 236

Then in Runtime, when the events are called, the script debugger will show the debug information (as shown in below figure). In the box **Locals** you can inspect all available variables and elements.

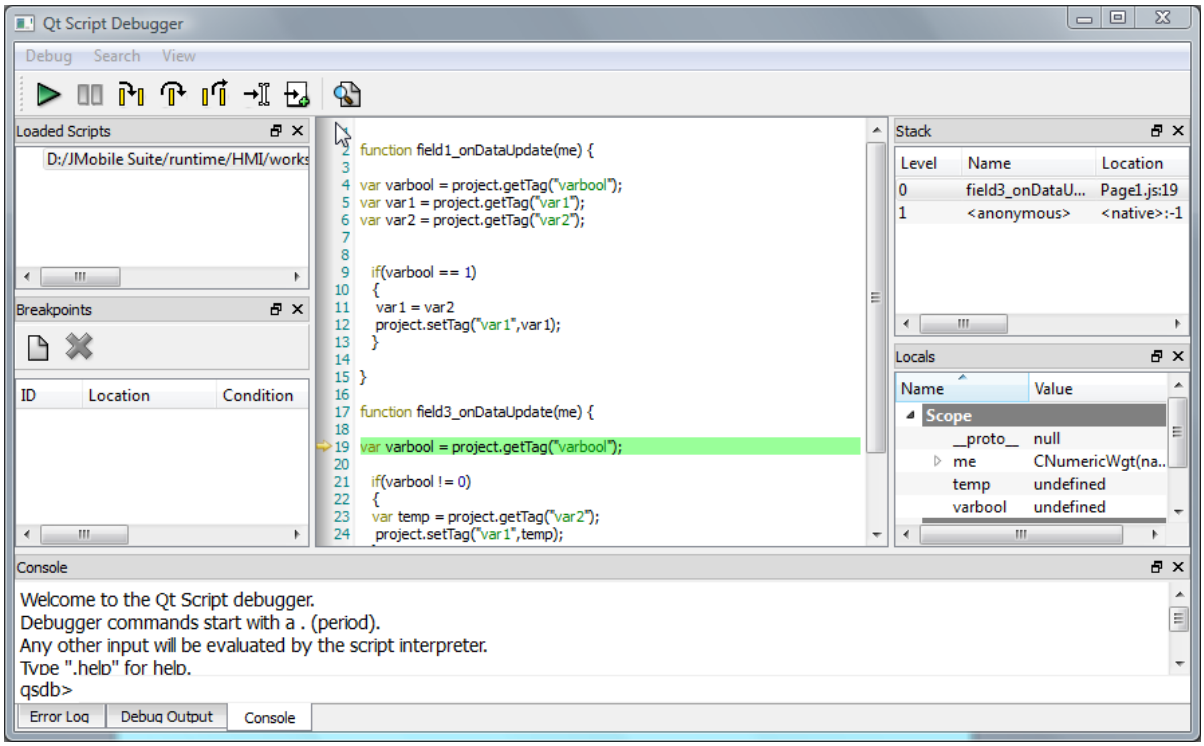

Figure 237

For a complete reference guide about Qt Script Debugger you can open the following link in your browser:

http://doc.troll.no/4.7/qtscriptdebugger-manual.html

19.5 Limitations

Widgets cannot be instantiated from JavaScript. The Widgets can only be accessed 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 VisuControl Touchpanels

Most of the system software components can be easily upgraded directly by the end users; this ensures a high degree of flexibility in providing updates and fixes to existing and running systems.

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 chapter, 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 manufacture.

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 may even render system unusable.

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

20.1 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 below figure.

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.

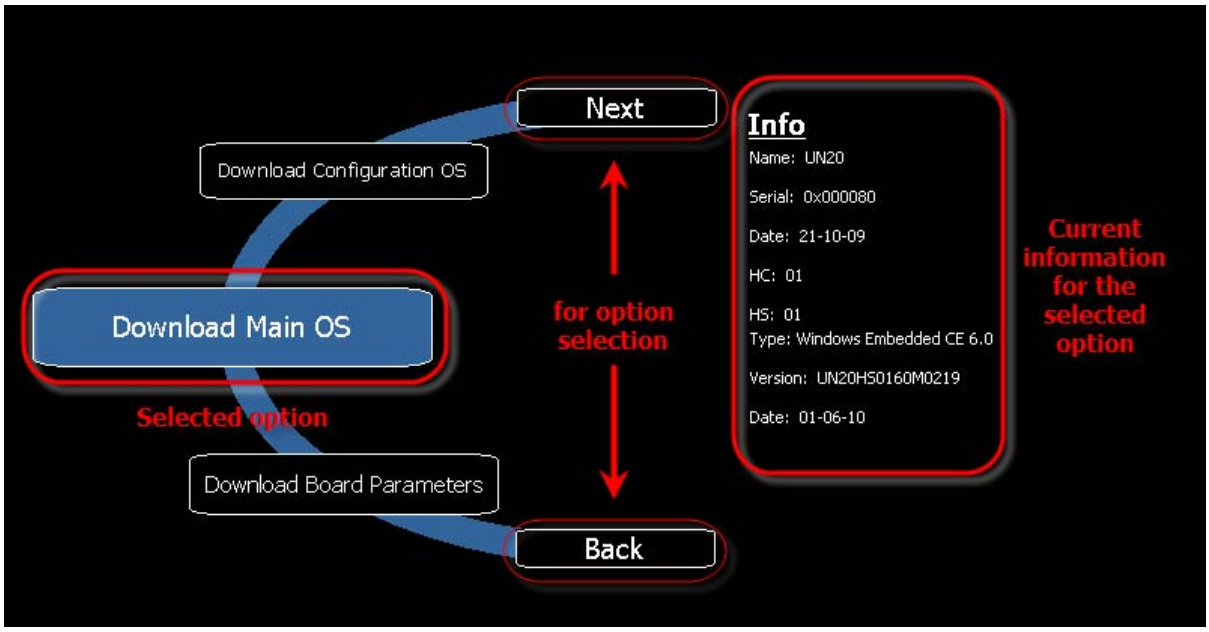

Figure 238

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 you to calibrate the touch screen interface
Plugin List	provides a list of the plug-in modules installed and recognized by the system; this option may not be supported by all platforms and all versions.
Network	allows you to change the options of the panel on-board network card
Time	allows you to change the panel RTC options, including time zone and DST
Display settings	automatic backlight turnoff and brightness adjustment
BSP settings	allows you to check the BSP (Board Support Package) 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, with all the available options; 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
Restore Factory Settings	has two options; the first (Uninstall HMI) removes from the unit the HMI runtime (if present); at next start the panel will behave as a brand new unit; this command does not reset settings like IP, brightness or RTC; the command may not be present in all platforms and versions; the second option (Clear System settings) allows to reset also the system parameters.
Resize Image Area	allows you 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 you 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 you 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 VisuControl software, which supports this feature starting from V 1. 50
Download Bootloader	allows for checking the actual version of the system boot loader and upgrading it (see below for additional details)

Download Main FPGA allows you to check the actual version and upgrade the main FPGA firmware (see below for additional details); this command may not be present in all platforms and versions.

Download Safe FPGA allows for checking the actual version and upgrading the back-up (safe) copy of the FPGA Firmware (see below for additional details);); this command may not be present in all platforms and versions.

Download System Supervisor allows for checking the 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

When executed in “System Mode” the System settings also provides the “BSP Settings”. Only when recalled from System Mode, the BSP settings shows an additional tab called “Password” as shown in the figure below.

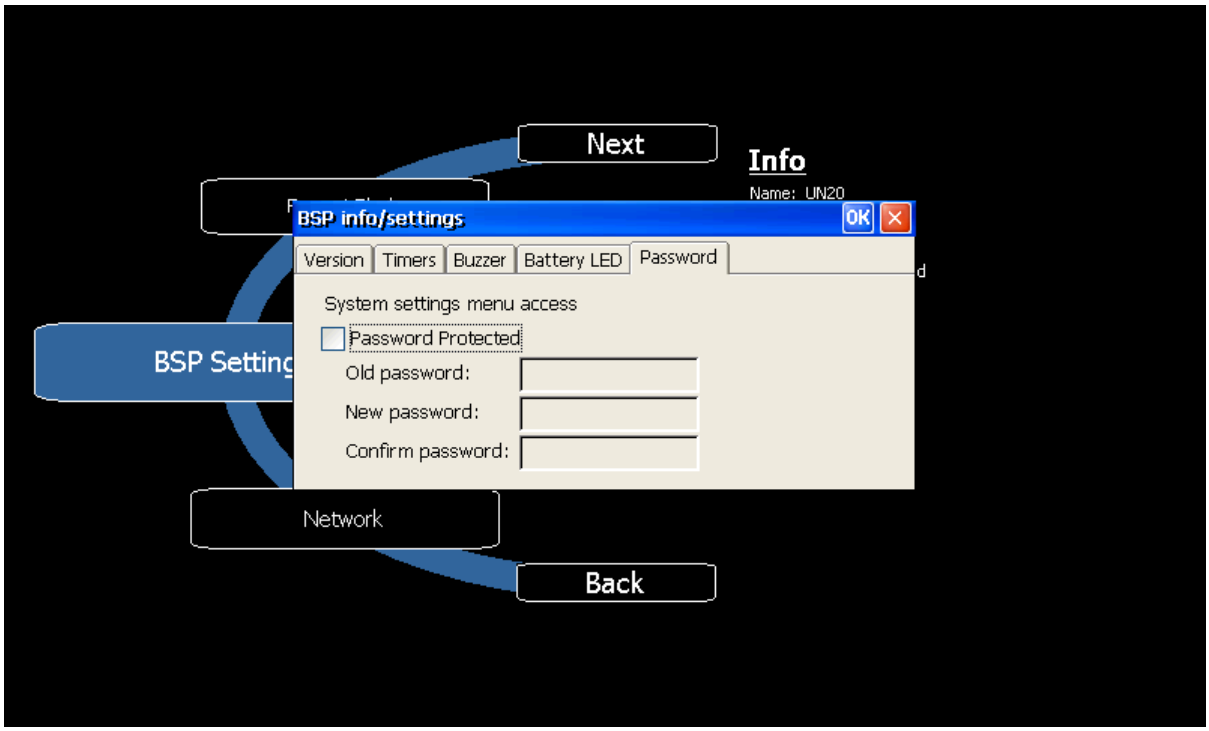

Figure 239

This function allows you to protect with password the access to the System settings in System mode so all the advanced and critical functions are not easily accessible to anyone.
To activate the protection, simply mark the check box “Password Protected” and specify the desired password as shown in the following figure.
The password must be at least 5 characters long.
If you are changing a password previously inserted or disabling the protection, you are asked to provide the old password first.

Note: please keep a note of the configured password in a safe place; there is no way to reset the password protection and in case it is lost the unit must be returned to the factory for proper reconditioning.

When the System settings menu is protected by a password, for each critical function that may compromise the proper system operation you try to execute, the HMI will prompt you to enter the password. If correct, the operation will proceed, if wrong, the operation will be aborted.

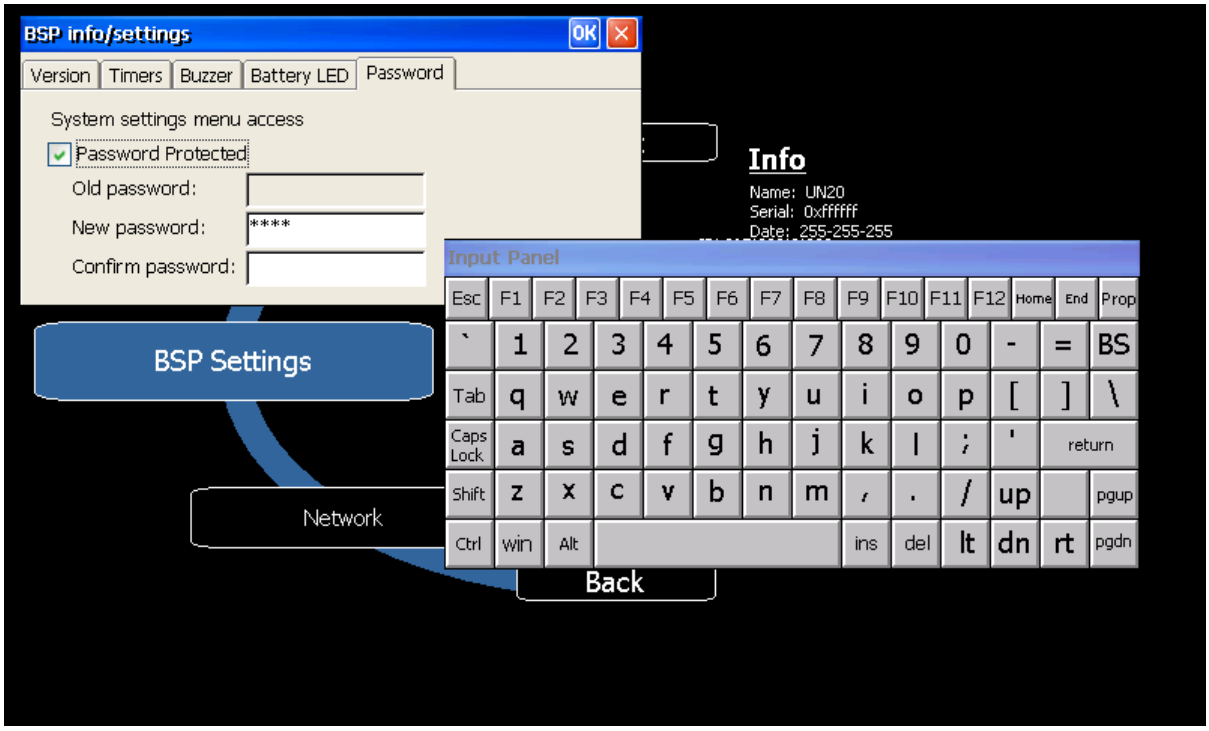

Figure 240

The System Settings tool is accessible from the VisuControl context menu by selecting the item "Show system settings". When activated in this way, the System Settings tool always starts in "User Mode".

The VisuControl context menu can be activated by pressing and holding down the free area on the screen, until the menu is displayed.

The VisuControl Touchpanel products also support a special procedure for accessing the System Settings tool; the special procedure is required to start the System Settings 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.

20.1.1 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 handle 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)

20.2 Update of System Components from VisuControl

VisuControl provides a tool to update system components downloading them to the target device using the Ethernet communication interface.

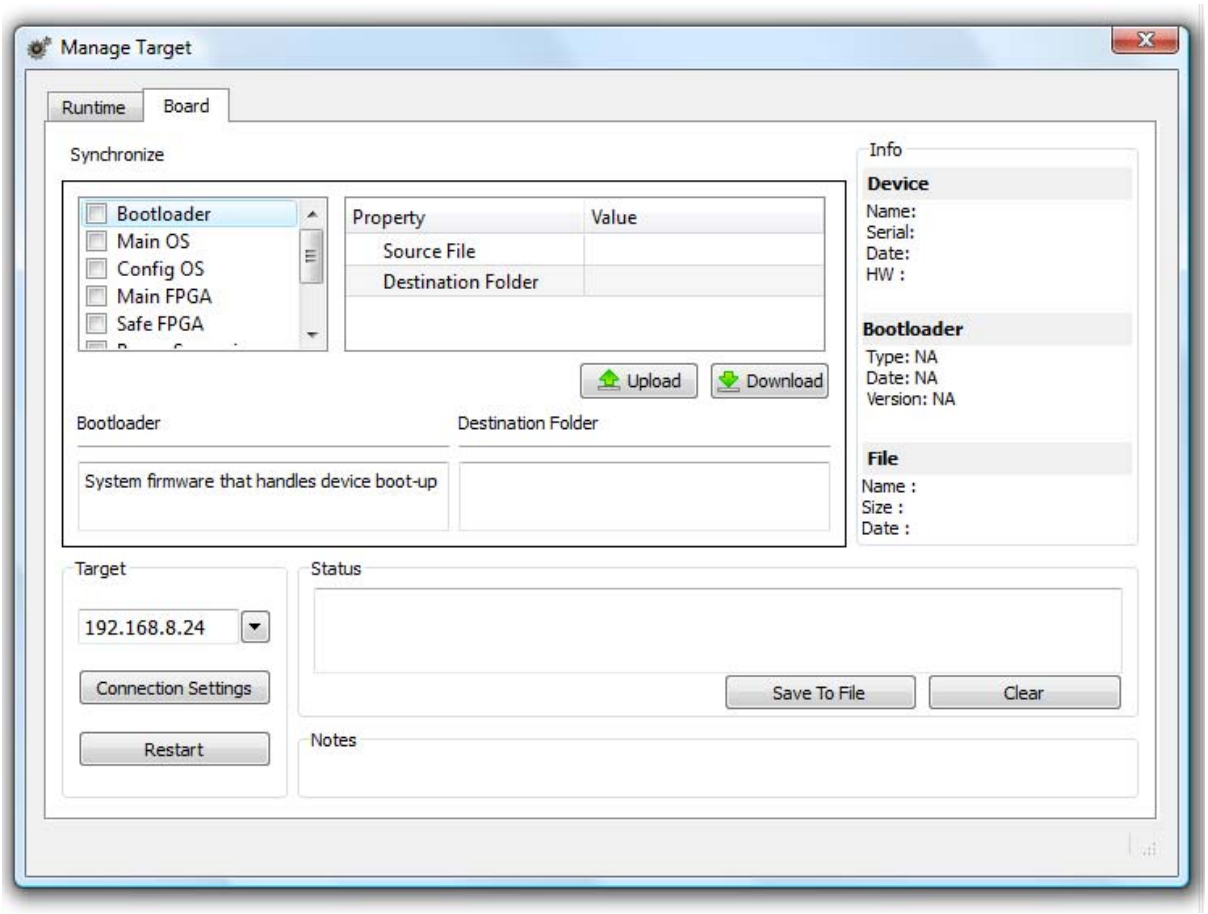

Figure 241

The Manage Target dialog has two tabs. Click on the "Board" tab to access 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. Then click out of the box to accept the inserted IP. See below figure.

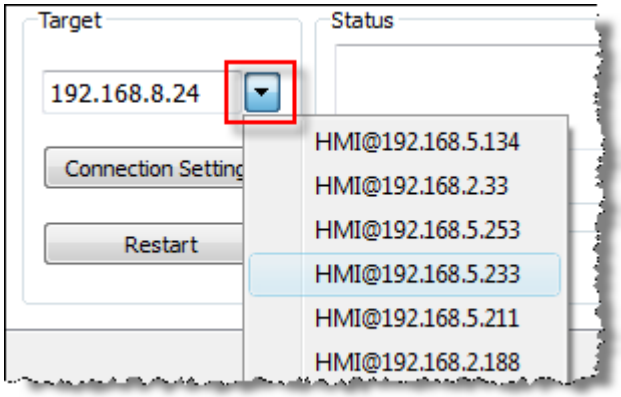

Figure 242

When the device is recognized the Info box shows the target details as shown as an example in below figure.

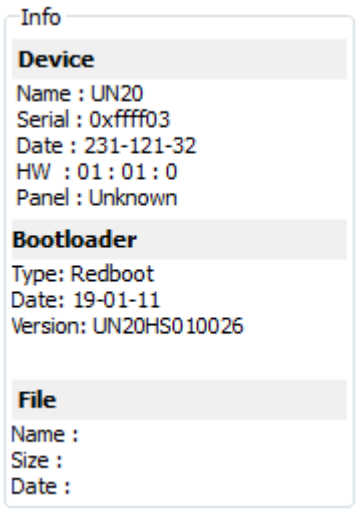

Figure 243

Locate 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 below figure.

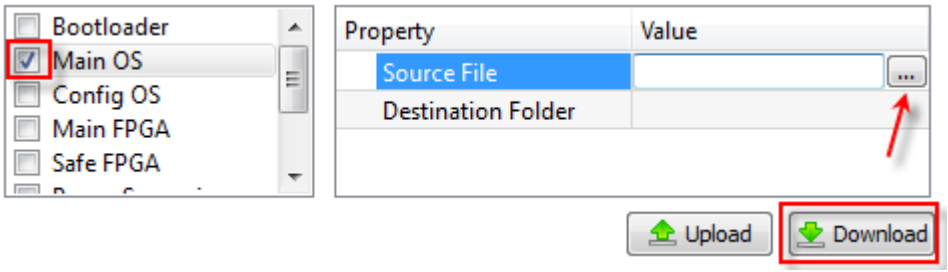

Figure 244

Then click 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 then execute the transfer of the entire elements one after the other and at the end you will need to cycle the power of the system.

Manage target allows you 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.

20.3 Update of the System Components via USB Flash Drive

The upgradable components in a unit are listed in the chapter "List of upgradable componets". Per each component a specific file (o set of files) is provided.
Files of system components are available on demand; please contact technical support for any specific needs.

Note: IMPORTANT – only units with regular serial numbers can be upgraded; units marketed as "prototypes" or having "Prot." serial numbers **cannot** be upgraded

20.3.1 Upgrade Steps

Copy all the 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). Below figure shows an example for the Main OS components.

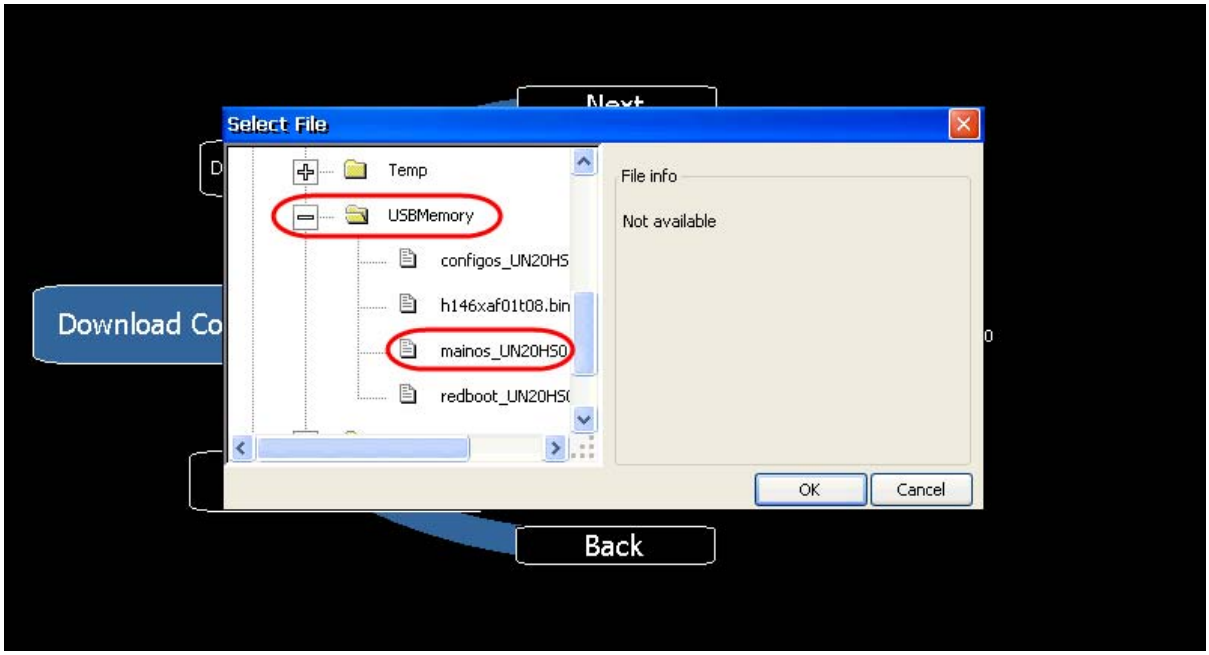

Figure 245

Note: Select the "Download" command to transfer files to the panel. Select the "Upload" command to get files from the panel.

Then follow the instructions on the screen to proceed with the update. A progress bar on the screen will inform you about the status of the operation.

Note: Upgrade procedure may change depending on the hardware revision or operating system version from which you start; please contact technical support offices for any detail about the exact sequence.

20.3.2 Updating Panel Runtime

This chapter assumes the panel is up and running with the runtime components installed. Please see the chapter “Transferring the Project to Target“ for any information related to the very first use of a brand new unit, just out of production.

Panel Runtime can be updated directly from VisuControl or using a USB pen drive.

20.3.3 Updating Runtime from VisuControl

From the Run menu select the "Manage Target" option, then click on "Update Runtime", as shown in the following image.

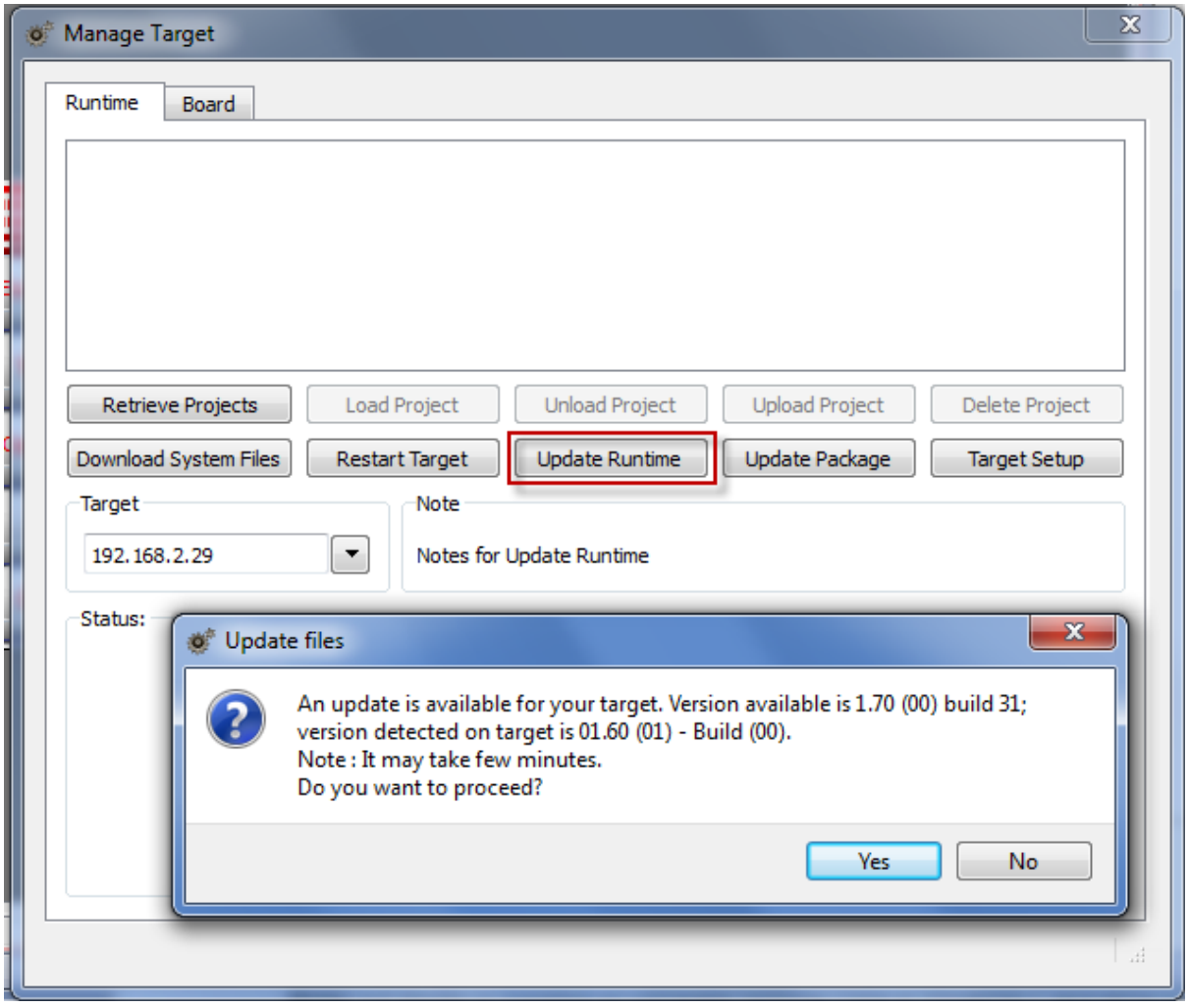

Figure 246

A confirmation message will inform you about the current version installed and the version available in VisuControl.

Note: If VisuControl is installed on computer running Vista or Windows 7 operating systems, you need to start it "as Administrator" using a right click on the VisuControl icon and selecting "Run as Administrator"

When using the “Download” facility of VisuControl to simply transfer a project to the panel, VisuControl is always checking in background if the existing runtime on target is compatible with the VisuControl version in use. In case the runtime on the panel is found not compatible with the actual version of VisuControl, you will be automatically asked to proceed to update the HMI runtime on the device. In case the runtime version of the panel is found not compatible, VisuControl will display a warning message as shown in the following figure.
You need to use the Manage Target tool to update the runtime as described in the previous chapter.

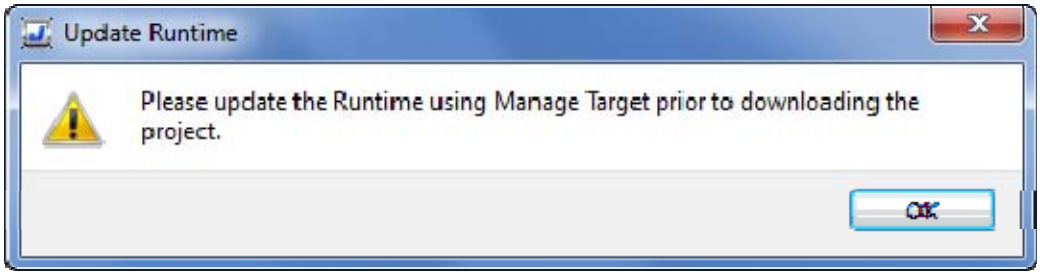

Figure 247

20.3.4 Updating Runtime from USB Pen Drive

From the Run menu, select "Update package" to start the update package creation process.

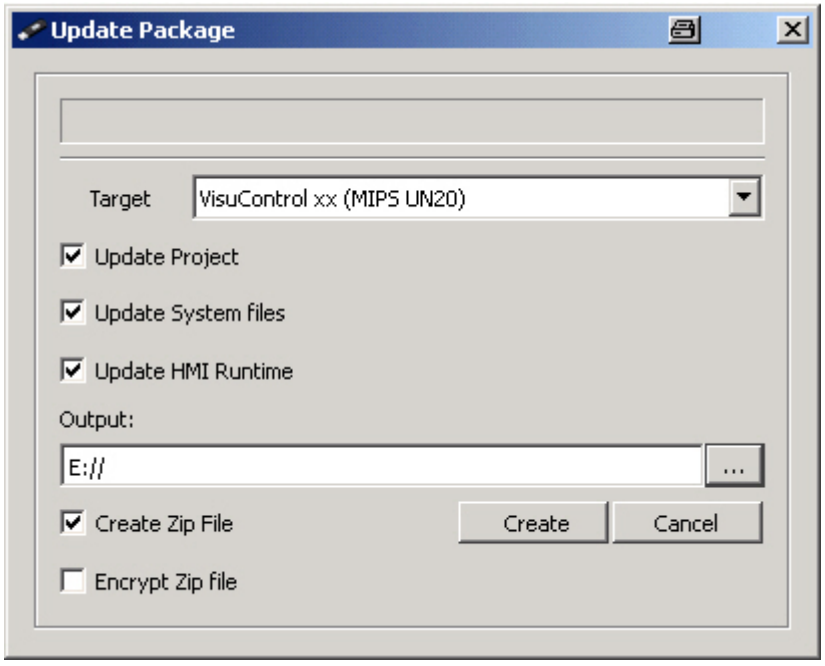

Figure 248

20.3.4.1 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".

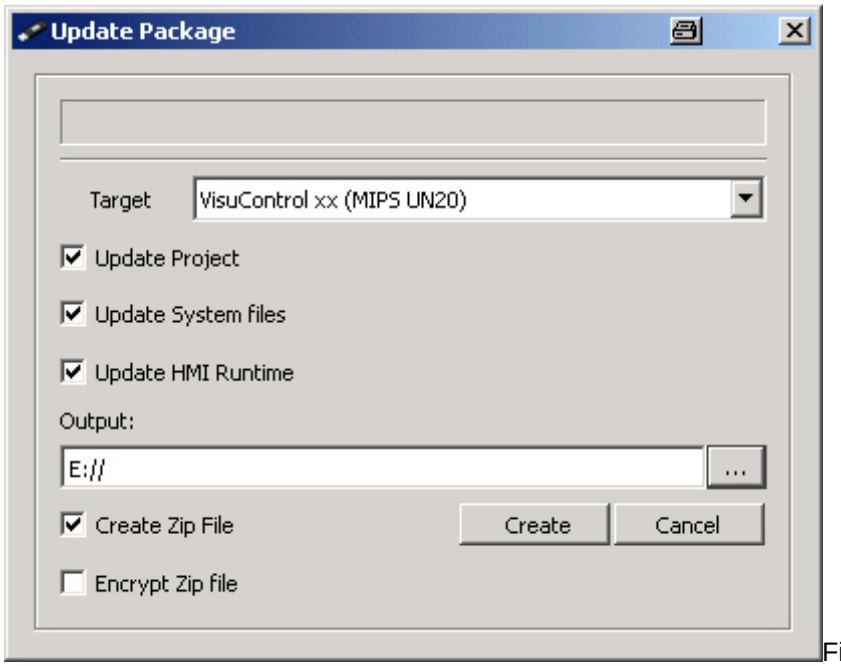

Figure 249

20.3.4.2 Zipped Package

The Package can be created in the zip format.
If Encrypt is checked, then a hard-coded password is used to encrypt the zip file.

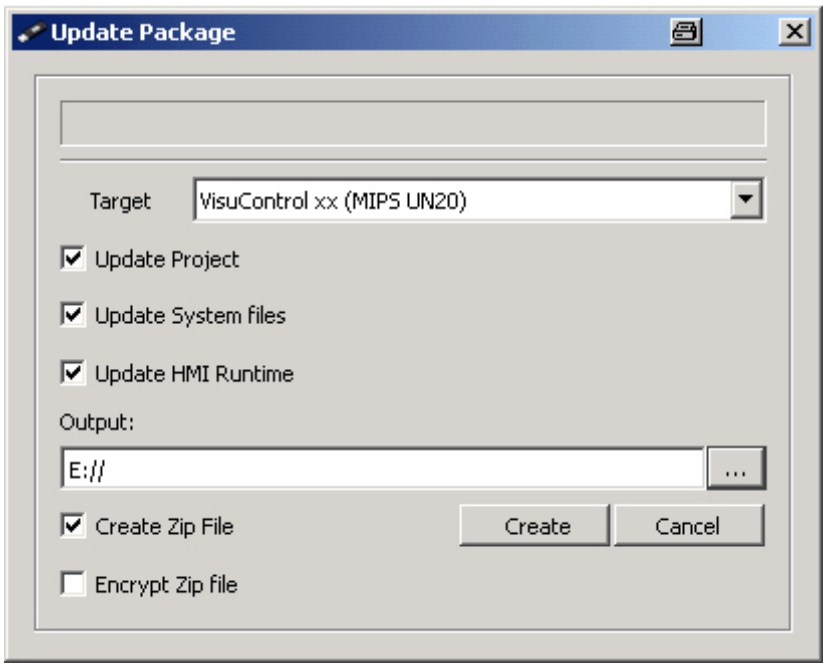

Figure 250

20.3.4.3 Transferring the update package on the target device

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.

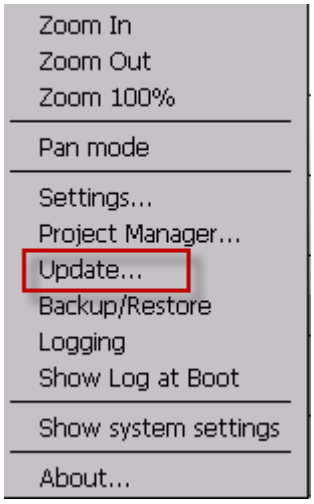

Figure 251

The wizard utility will start. Check "Auto select best match", and then click "Next".

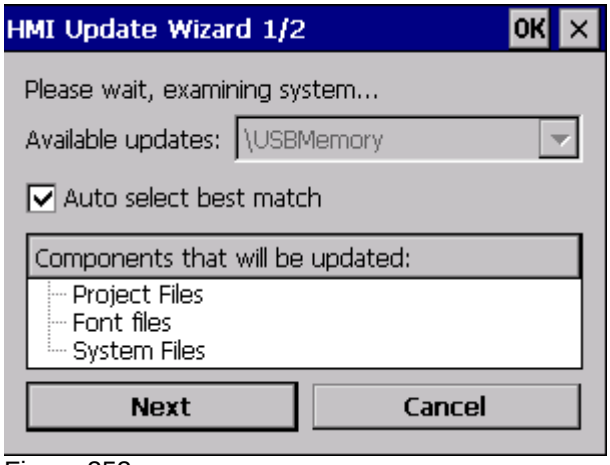

Figure 252

Press on "Close" at the end of the operation.

21 VisuControl Functional Specifications and Compatibility

The scope of this chapter is to provide a clear overview of the supported functions and related limitations for both programming software and HMI runtime system.

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.

21.1 Table of Functions and Limits

Function \ Feature	Max allowed
Number of pages	1000 (max screen size 1280x800 pixels)
Number of Widgets	2000
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 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 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 Trend Buffer	200000
Number of Trend Buffer Samples for a Project	2400000
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 page	8KB
Size of project on disk	30MB

21.2 Compatibility

Starting from the first official release of VisuControl V1.00 (00) we have applied the following policy for compatibility:

VisuControl version MUST be always aligned with VisuControl Runtime on panel; the user has the responsibility to update Runtime components on the Target device together with any VisuControl update; a Runtime update can be done directly from VisuControl using the "Update Target" command available in the "Run\Manage Target" dialog

Any version of VisuControl 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 VisuControl, 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 VisuControl is not ensured.