Creator

User Manual

Creator Software User Manual

This is the User Manual for the Creator software that can be used to design SmartView series and the HA-401 products, and is intended for system integration, programming and maintenance personnel.

Warranty

All products manufactured by ICP DAS are warranted against defective materials for a period of one year from the date of delivery to the original purchaser.

Warning

ICP DAS assumes no liability for damages consequent to the use of this product. ICP DAS reserves the right to change this manual at any time without notice. The information furnished by ICP DAS is believed to be accurate and reliable. However, no responsibility is assumed by ICP DAS for its use, or for any infringements of patents or other rights of third parties resulting from its use.

Copyright

Trademarks

Names are used for identification purposes only and may be registered trademarks of their respective companies.

License

This license allows the software to be used, modified and backed up on a single machine. This software, or any copy thereof, in whole or in part, may not be reproduced, transferred or distributed in any form or by any means, including electronic, mechanical, photocopying, recording or otherwise, without the prior written permission of ICP DAS.

Technical Service

If you encounter any difficulties or have any questions, feel free to contact your local agent or contact ICPDAS by email at: service@icpdas.com or service.icpdas@gmail.com. We will respond within 2 working days.

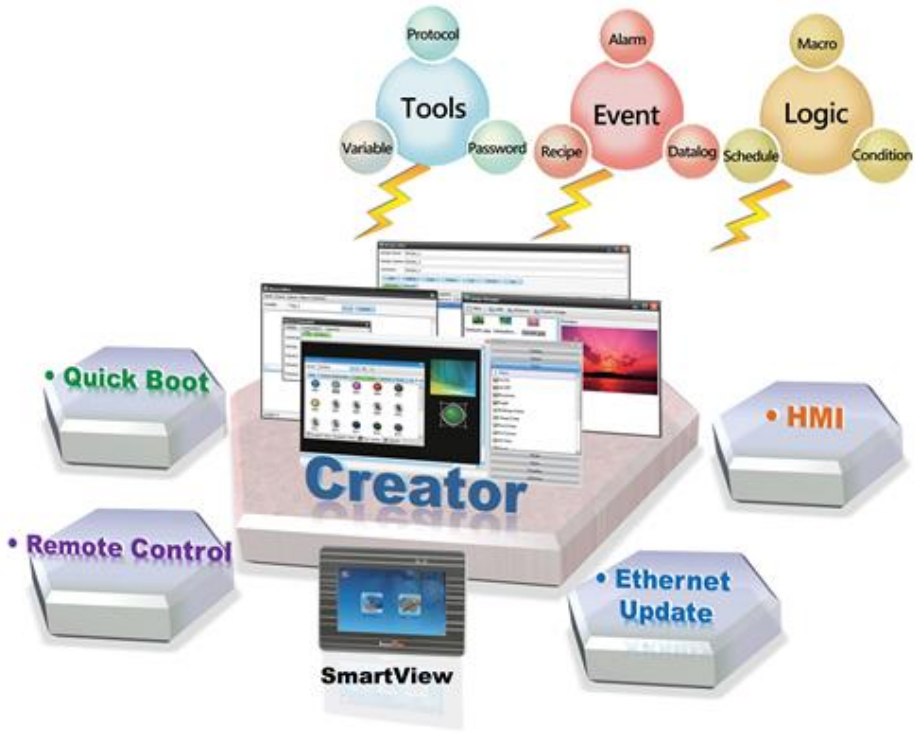
For more product information, refer to the ICPDAS website at: www.icpdas.com.

Last Revised by Janice Hong, R&D dept., ICP DAS.

Table of Contents

Chapter 1 Introduction to Creator

Creator is a professional development toolkit especially designed for the **SmartView (or HA-401) series** of devices produced by ICP DAS. **Creator** can be used to integrate several commonly used PLC communication protocols, providing rich and flexible object editing tools that allow easy navigation and adjustment of window management, repeated import and export of data in order to shorten the development process, simple uploading or updating of SmartView (or HA-401) projects using the built-in TCP transport, and the construction of control systems, ranging from a small and simple local control/monitoring application to management systems for large buildings, factories, and engine rooms, etc.



1.1 Features

- Supports commonly used PLC Communication Protocols
- Easy to create HMI Projects without the need for complex coding
- Diverse range of HMI Objects and Functions, including:
  - Alarms, Schedules, Recipes, Data Logging, Macros, etc.
- Update Projects via Ethernet
- Supports MQTT
- Online/Offline Simulation
- Supports Multiple languages (Traditional Chinese/Simplified Chinese/English)

Chapter 2 Introduction to the SmartView Series of Devices

The **SmartView** series of devices from ICP DAS combines a RISC-based CPU board, a TFT LCD touch screen and a ruggedized flat panel computer, and includes a wide range of software, such as HMI and MQTT, which are perfect for a variety of control and HMI applications. The **SmartView** series provides a variety of connection options, including Gigabit Ethernet, RS-232 and RS-485 ports.

The operating system is pre-installed in the onboard Flash memory, and Remote I/O expansion is available using ICP DAS Ethernet I/O modules and RS-485 I/O modules. The SmartView series is designed for panel-mount installation. The front panel is NEMA 4/IP65 rated, meaning that it can withstand sprayed water, humidity, and extreme dust, and can be operated over a wide ambient temperature range of -10°C to +60°C. The fan-less design provides the ultimate in reliability with no moving parts.

2.1 Hardware Features

The SmartView series delivers the most comprehensive configuration and remote system upgrade solutions to meet specific application requirements. The following is an overview of the hardware and software features that have been designed to simplify installation, configuration, and application.

- TFT LCD: 7", 10.4", or 15"
- Active M2M Transmission Mechanism: MQTT
- Makes connectivity for the “internet of things” and mobile devices easily
- Hard Real-time Capability
- Power over Ethernet (PoE)
- NEMA 4/IP65 Compliant Front Panel
- Ultra-Rugged Construction and Reliable Design
- Operating Temperature: -10 to +60°C

2.2 Specifications

The following is a summary of the specifications for the SmartView series of devices.

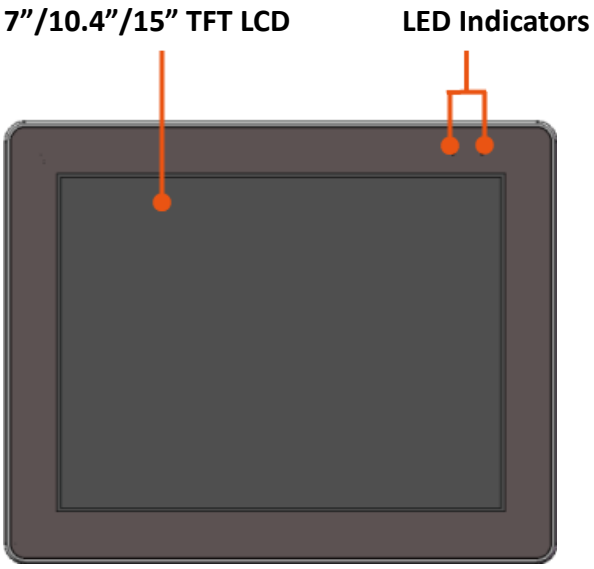
Models	S V - 2 2 0 1	S V - 4 2 0 1	S V - 6 2 0 1
LCD Display			
Size	7” (16:9)	10.4” (4:3)	15” (4:3)
Resolution	800 x 480	800 x 600	1024 x 768
Brightness (cd/m2)	400		
Contrast Ratio	500 : 1		
LED Backlight Life	20,000	50,000	
Touch Panel	4-wire, resistive type; Light Transmission: 80%	5-wire, resistive type; Light Transmission: 80%	
LED Indicators	2 (PWR, Run)		
System Software			
OS	Linux		
CPU Module			
CPU	RISC-based		
SDRAM	512 MB		
MRAM	128 KB		
Flash	256 MB		
Memory Expansion	microSD socket (Supports up to 32 GB)	SD socket (Supports up to 32 GB)	
EEPROM	16 KB		
RTC (Real Time Clock)	Yes		
Hardware Serial Number	Yes, 64-bit Hardware Serial Number		
Rotary Switch	Yes (0 to 9)		
Audio	Earphone-out		

Models	S V - 2 2 0 1	S V - 4 2 0 1	S V - 6 2 0 1
Communication Interface			
Ethernet	1x RJ-45, 10/100/1000 BaseTX		
COM1	RS-232/485 (DB9 Connector); 2500 VDC isolated		
COM2	RS-232/485 (DB9 Connector); 2500 VDC isolated		
COM3	-	RS-485 (Terminal Block, Data+, Data-); 2500 VDC isolated	
Mechanical			
Dimensions (W x L x H)	213 x 148 x 44 (mm)	291 x 229 x 54 (mm)	381 x 305 x 65 (mm)
Installation	Panel Mounting		
Ingress Protection	Front Panel: NEMA 4 /IP65		
Environment			
Operating Temperature	-10 to +60°C		
Storage Temperature	-20 to +70°C		
Ambient Relative Humidity	10 to 90% RH (Non-condensing)		
Power			
Power from Terminal Block	Yes, +12 to +48 VDC		
Power from PoE	Yes, IEEE 802.3af		
Isolation	-	1 kV	
Consumption	6 W	13 W	

2.3 Hardware Overview

The SmartView series of devices is equipped with a variety of interfaces and peripherals that can be integrated with external systems. The following is an overview of the integrated components.

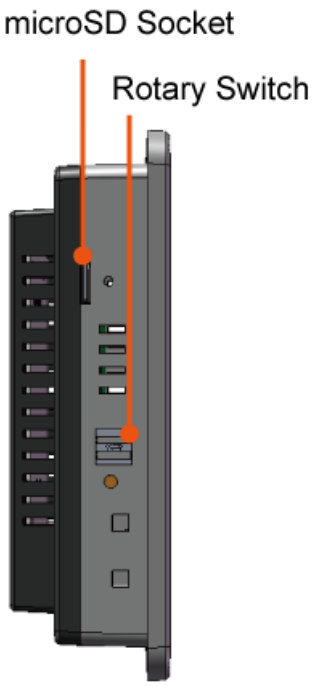
Front View



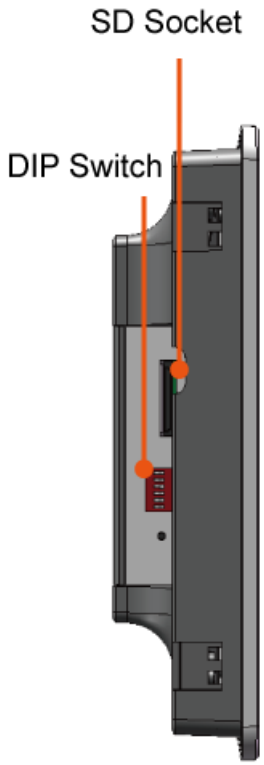
<table>
<tr><th>Item</th><th>Description</th></tr>
<tr><td>LED Indicators</td><td>The SmartView device contains two system LED indicators that display the status of the device, as shown below.
<table>
<tr><th>LED Indicator</th><th>Color (ON State)</th><th>Description</th></tr>
<tr><td>PWR</td><td>Green</td><td>The power is on</td></tr>
<tr><td>RUN</td><td>Red</td><td>The power is on and the OS is running</td></tr>
</table></td></tr>
</table>

Side View

SV-2201



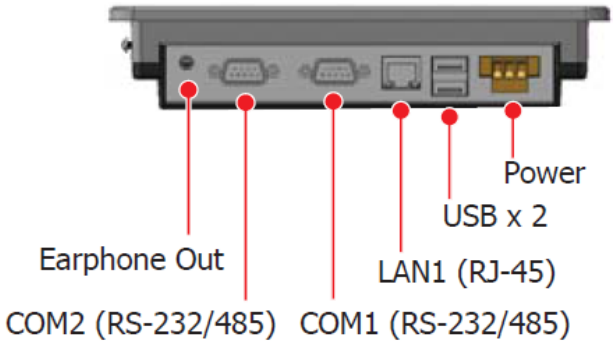
SV-4201 / SV-6201



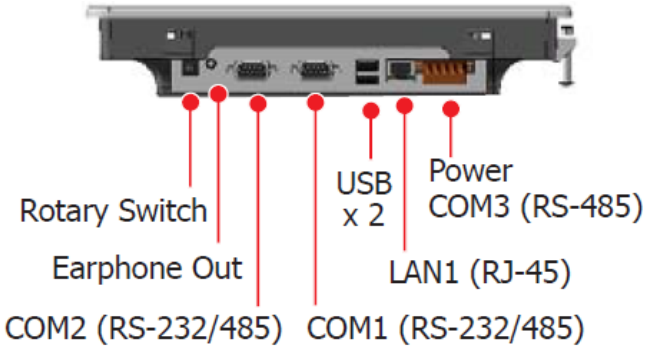
Item	Description
microSD Socket	The microSD socket enables the SmartView operating system to be restored via a microSD card, or to expand the memory to up to 32 GB.
Rotary Switch	The Rotary Switch is the operating mode selector switch that provides seven functions, and is used to select the operating mode for the SmartView device. 
SD Socket	The SD socket enables the SmartView operating system to be restored via an SD card, or to expand the memory to up to 32 GB.
DIP Switch	The DIP Switch is a binary switch. Each individual DIP Switch represents a unique value, which is combined to form the device address.

Bottom View

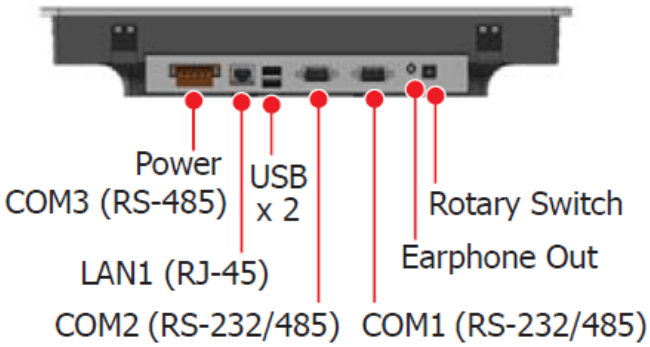
SV-2201

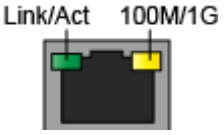
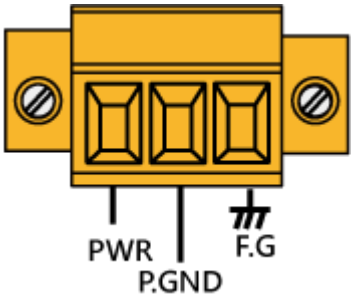
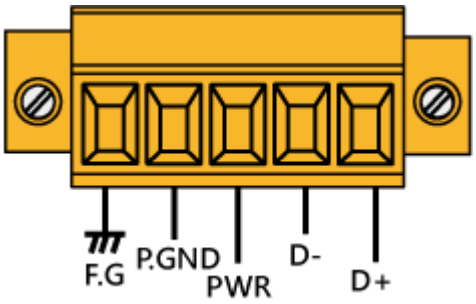


SV-4201



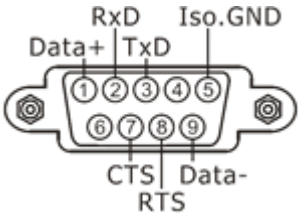
SV-6201



<table>
<tr><th>Item</th><th>Description</th></tr>
<tr><td>Earphone-Out</td><td>The SmartView series of devices includes an earphone output jack that can be used to output audio.</td></tr>
<tr><td>Rotary Switch</td><td>The Rotary Switch is an operating mode selector switch that provides seven functions which can be used to select the operating mode for the SmartView device.
  </td></tr>
<tr><td>LAN Port</td><td>The SmartView series of devices includes an Ethernet port that can be connected to an external computer or other device via an Ethernet cable.
 
<table>
<tr><th>LED Indicator</th><th>State (Color)</th><th>Description</th></tr>
<tr><td rowspan="2">100M/1G</td><td>ON (Orange)</td><td>Network Speed: 1 G</td></tr>
<tr><td>OFF</td><td>Network Speed: 10/100 M</td></tr>
<tr><td rowspan="3">Link/Act</td><td>ON (Green)</td><td>The Link is active</td></tr>
<tr><td>OFF</td><td>The Link is inactive</td></tr>
<tr><td>Blinking(Green)</td><td>Network activity</td></tr>
</table></td></tr>
<tr><td>Power</td><td>The SmartView series of devices includes either a 3- or 5-pin terminal block, depending on the model. Two pins are used for power input and the other 1 or 3 pins are used for the Frame Ground/COM3, as illustrated in the diagrams below.
<u>SV-2201</u>
 
<u>SV-4201 / SV-6201</u>
  </td></tr>
</table>

COM1 (RS-232/RS-485)

The COM1 port is a 9-pin RS-232/RS-485 connector. The pin arrangement for the COM1 port is shown in the diagram.



Port Type: Male
Baud Rate: 115200, 57600, 38400, 19200, 9600, 4800, 2400, 1200 bps
Data Bits: 5, 6, 7, 8
Parity: None, Even, Odd, Mark (Always 1), Space (Always 0)

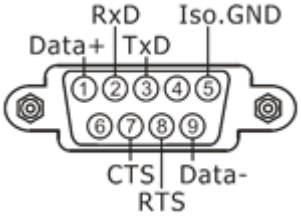
The COM1 port can be configured as either RS-232 or RS-485, although only one can be selected at a time and the configuration depends on the pin connections, as described below:

RS-232 (RXD, TXD, CTS, RTS and GND)
RS-485 (Data+ and Data-)

Note that no software configuration or hardware jumper is needed.

COM2 (RS-232/RS-485)

The COM2 port is a 9-pin RS-232/RS-485 connector. The pin arrangement for the COM2 port is shown in the diagram.



Port Type: Male
Baud Rate: 115200, 57600, 38400, 19200, 9600, 4800, 2400, 1200 bps
Data Bits: 5, 6, 7, 8
Parity: None, Even, Odd, Mark (Always 1), Space (Always 0)

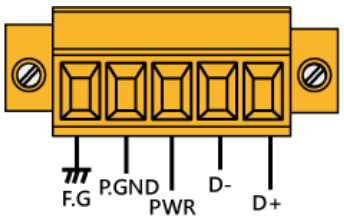
The COM2 port can be configured as either RS-232 or RS-485, although only one can be selected at a time and the configuration depends on the pin connections as described below:

RS-232 (RXD, TXD, CTS, RTS and GND)
RS-485 (Data+ and Data-)

Note that no software configuration or hardware jumper is needed.

COM3 (2-wire RS-485) for SV-4201 / SV-6201

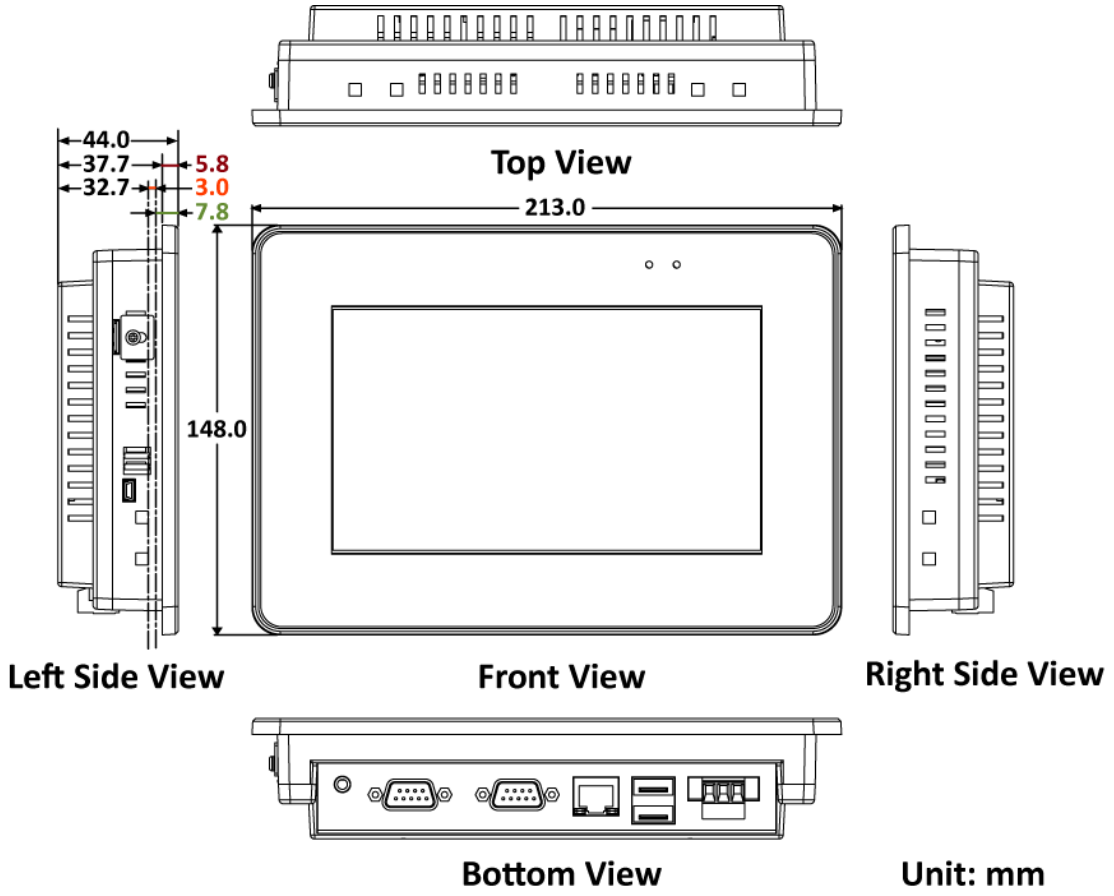
The COM3 port is a 2-wire RS-485 connector. The connections for the COM3 port are shown in the diagram.

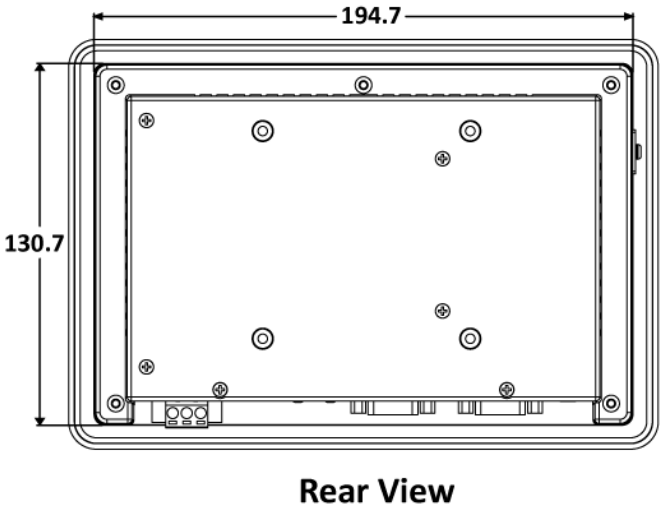


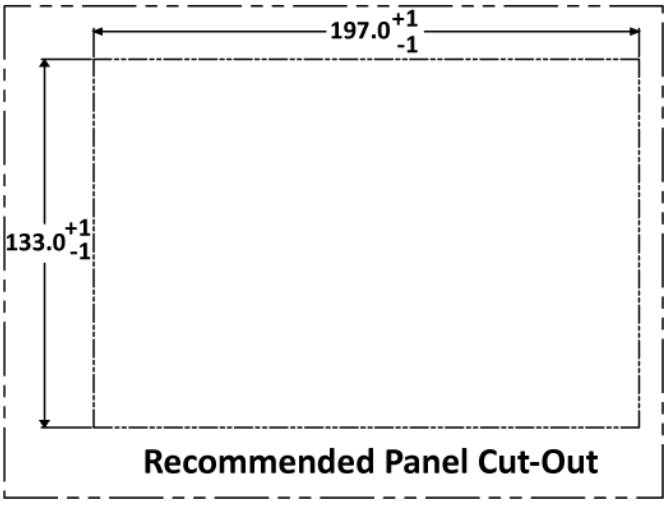
2.4 Dimensions

The diagrams below provide an overview of the dimensions and the panel cutout that can be used to define any enclosure specifications. Be sure to leave room for potential expansion if other components may be integrated into the system at a later date. All dimensions are in millimeters.

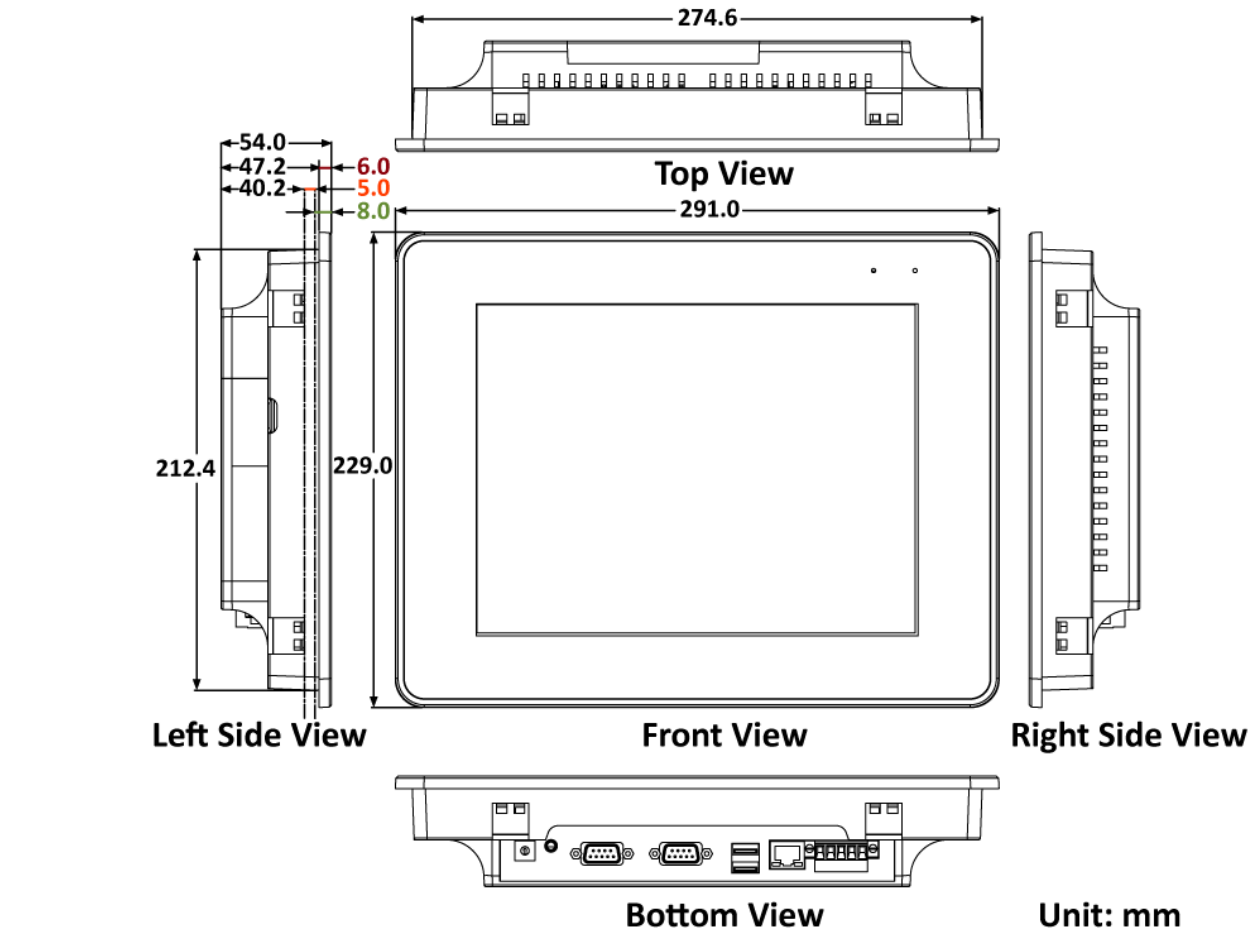
SV-2201

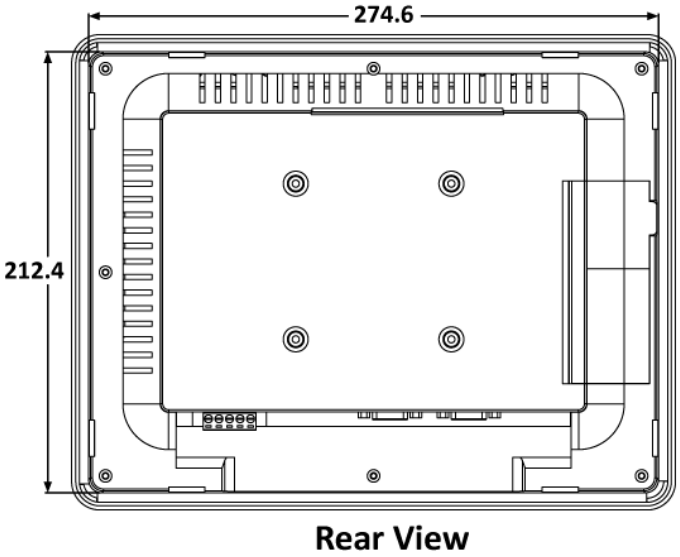


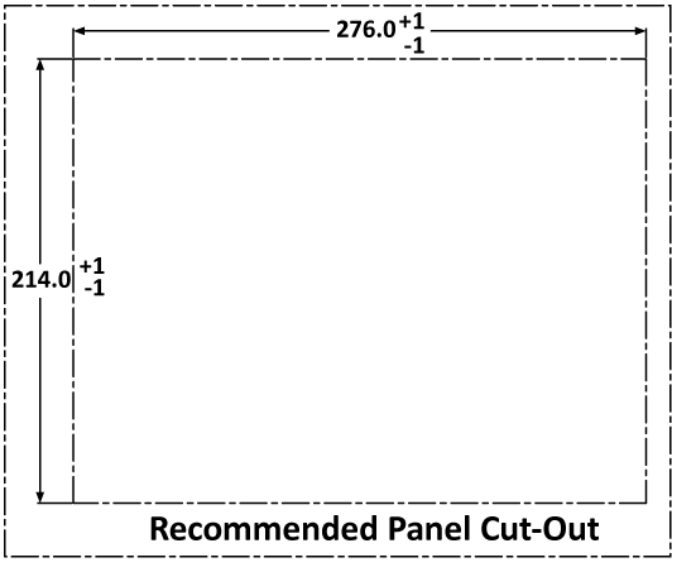




SV-4201





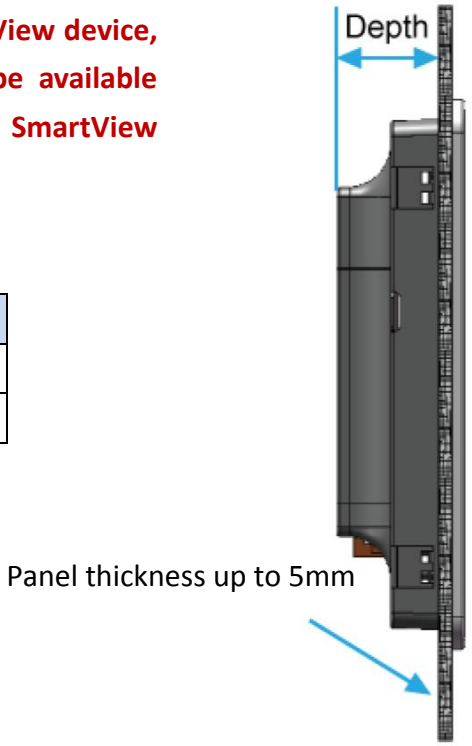


2.5 Mounting the Hardware

The SmartView series of devices can be mounted on a panel with a maximum thickness of 5 mm. Adequate access space should be made available at the rear of the instrument panel for wiring and servicing purposes.

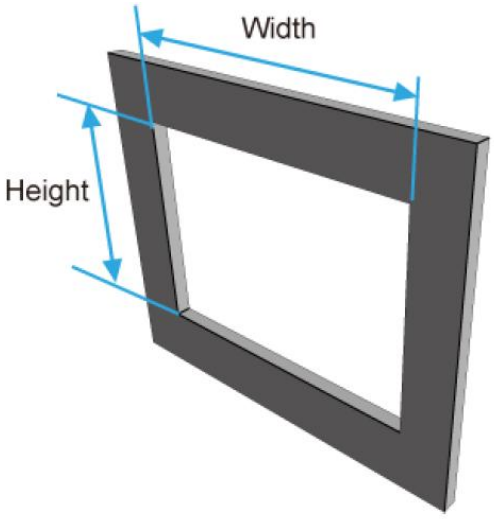
To ensure proper ventilation for the SmartView device, a minimum of 50 mm of space should be available between the top and bottom edges of the SmartView device and the enclosure panel.

Models	Depth
SV-2201	32.7 mm
SV-4201	40.2 mm



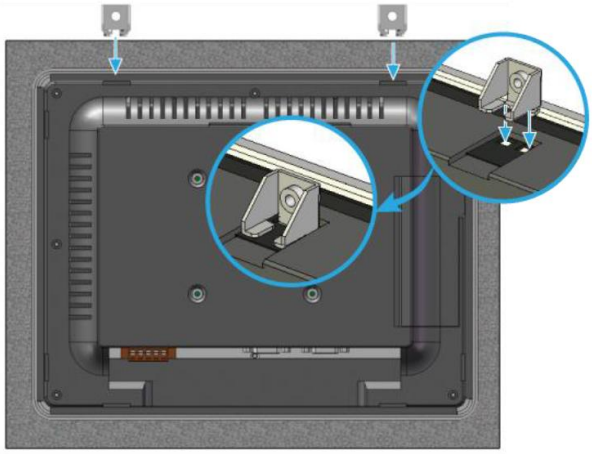
1. Cut a hole in the panel to the specified size. The dimensions of the panel cut-out for the SmartView device are shown below.

Model	Panel Cutout (Width x Height)
SV-2201	197 mm x 133 mm, ± 1 mm
SV-4201	276 mm x 214 mm, ± 1 mm

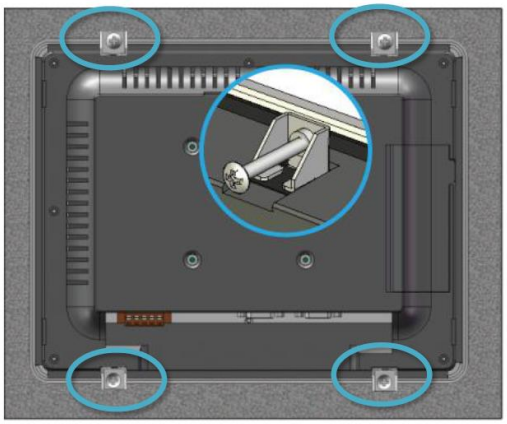


2. Attach the SmartView device to the cutout hole.

3. Insert the mounting clips for the panel into the upper and lower ventilation holes.



4. Attach the mounting clips to the panel using a screw.



Recommended Screw Torque: 3.4 to 4.5 kgf-cm.

Chapter 3 Software Installation

The following provides details related to the installation of the Creator software, including the recommended operating system and hardware specifications.

3.1 Hardware and Software Specifications

Before installing the Creator software, ensure that both the hardware specifications and operating system are sufficient to allow the efficient performance of the software.

3.1.1 Recommended Operating Systems

- Microsoft Windows 7 or later
- Microsoft .Net Framework version 3.5 or later

3.1.2 Recommended Hardware Specifications

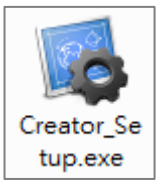
- CPU: 1.8 GHz or better
- Memory: Minimum of 1Gb RAM
- Hard Disk: At least 40G of free space
- Display: Full-color display that supports a resolution of 800*600 or better

3.2 Installation Procedure

3.2.1 Installing from the companion CD

- Automatic:
  After inserting the companion CD into the CD-ROM, Windows will automatically launch the installation file. Note that if the “Autorun” feature is not enabled in Windows, this function will not work and the manual method described below must be used.

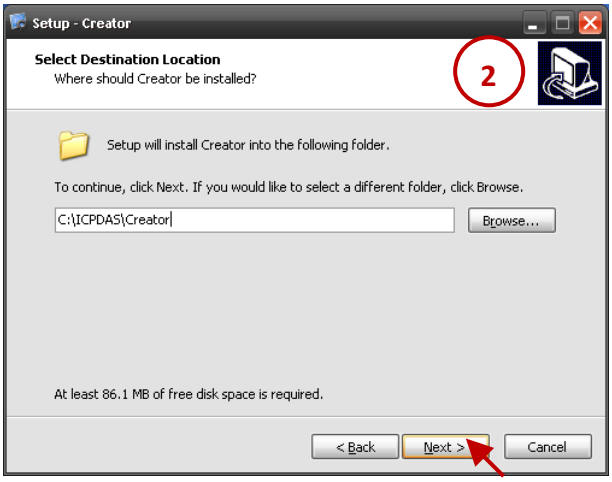
- Manual:
  Double-click the Creator_Setup.exe file that can be found in the root directory of the CD-ROM to launch the installation file.



3.2.2 Setup Wizard

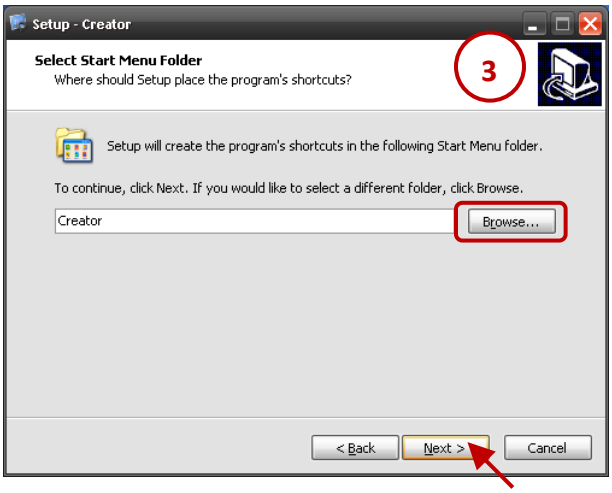
Install the software **Creator** by following the instructions given in the Setup Wizard.

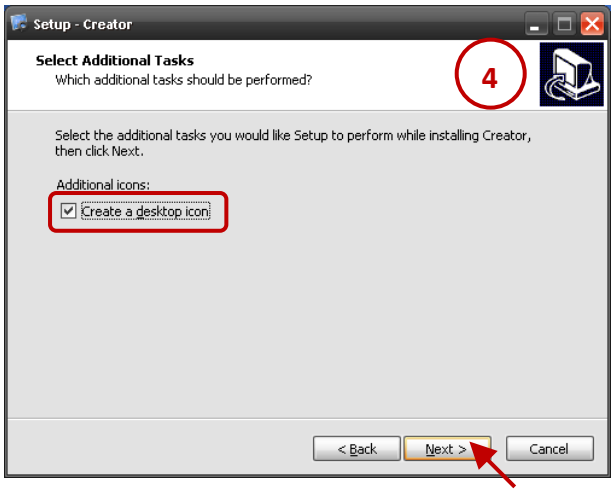
1. Click the **Next** button to begin the installation process.



2. On the **Select Destination Location** screen, either click the **Next** button to install Creator into the default folder, or click the **Browse…** button to select an alternate folder, and then click the **Next** button to continue.

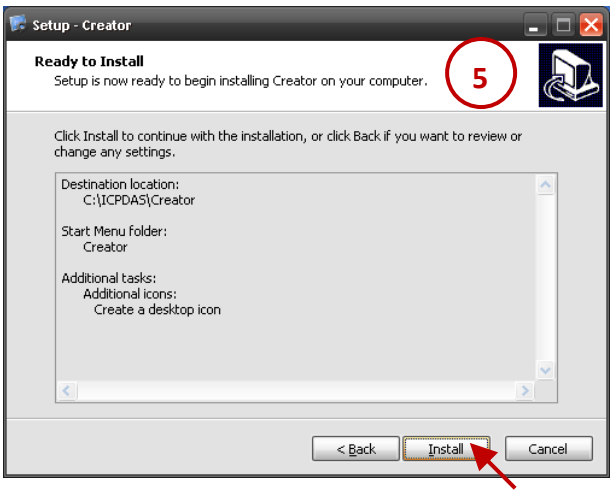
3. On the **Select Start Menu Folder** screen, either allow the shortcuts to be created in the default Start Menu folder, or click the **Browse…** button to select an alternate folder, and then click the **Next** button to continue.

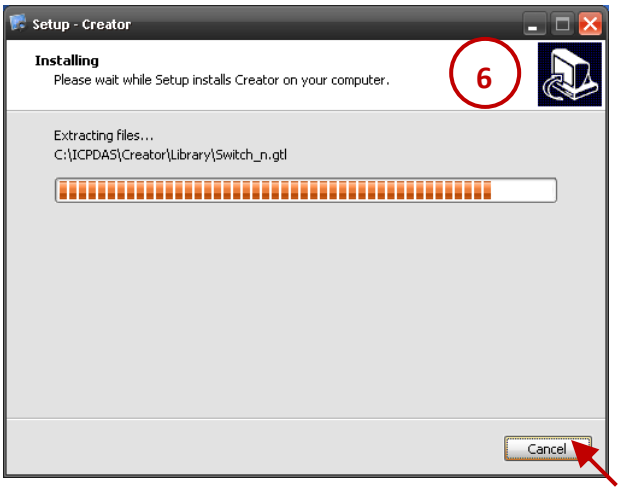




4. On the **Select Additional Tasks** screen, check the **Create a desktop icon** checkbox to create a shortcut on the desktop, and then click the **Next** button to continue.

5. On the **Ready to Install** screen, verify that the settings are correct, and then click the **Install** button to begin the installation.



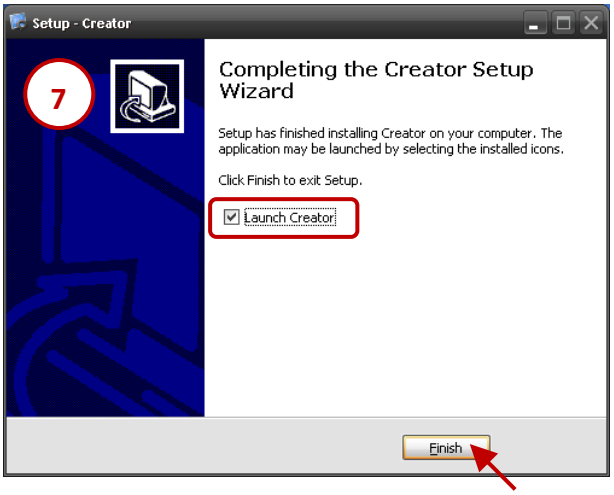


6. The **Setup Wizard** will display a progress bar to indicate the status of the installation process. Click the **Cancel** button to stop the installation if necessary.

7. Once the installation has been completed, click the **Finish** button to exit the Setup Wizard.

Note:

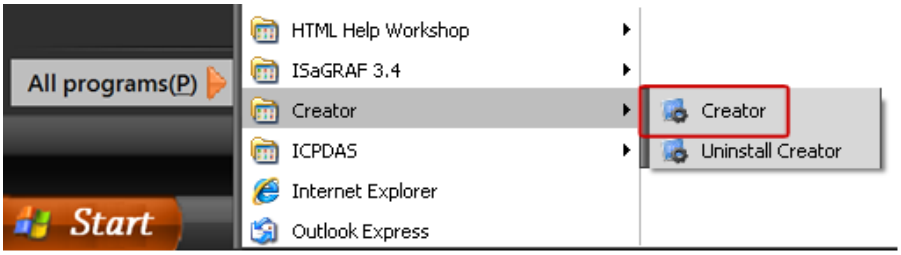
To automatically launch Creator once installation is complete, check the **Launch Creator** checkbox.



3.2.3 Execute the Creator software

To launch Creator, double-click the shortcut that was created on the desktop.

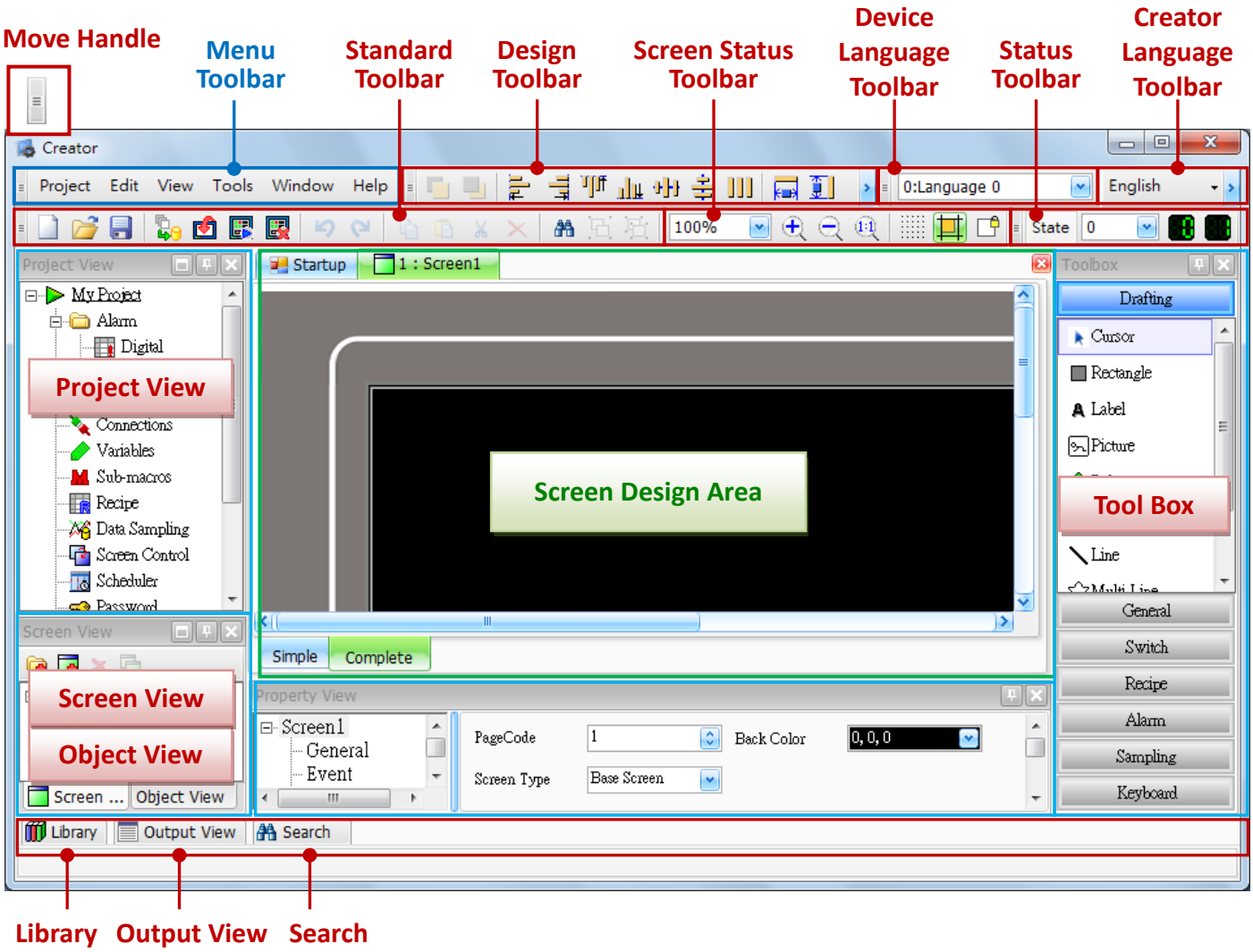
Alternatively click the **Start** button, and then point to **All programs**. Point to the **Creator** folder and then click **Creator**.



Chapter 4 Overview of the Software Interface

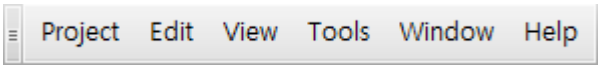
4.1 Program Window Interface

The diagram below provides an overview of the Creator interface, including the various menus and view areas, each of which is described in more detail below.

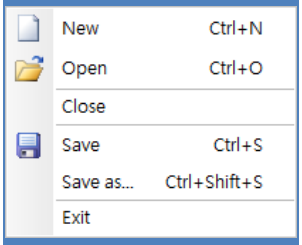
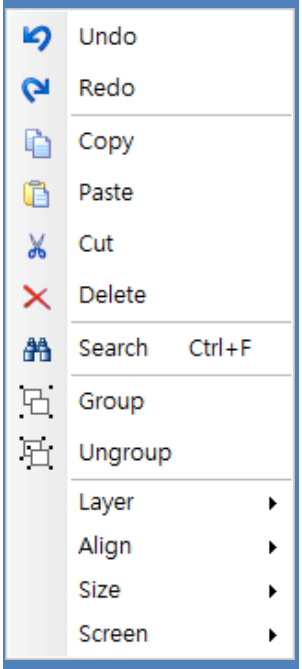


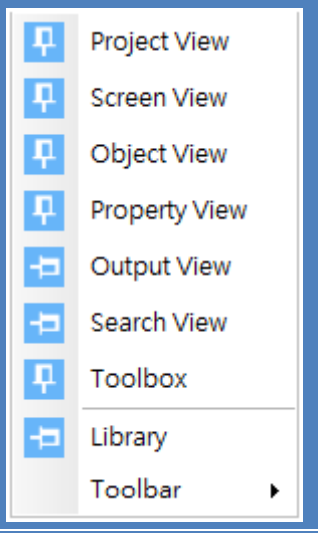
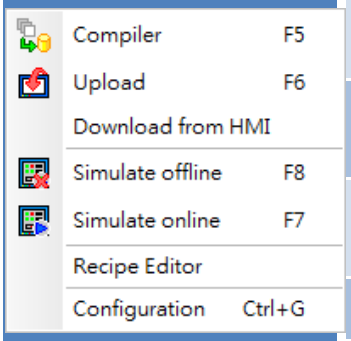
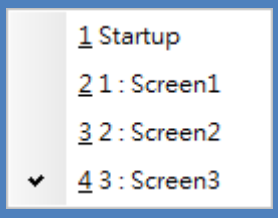
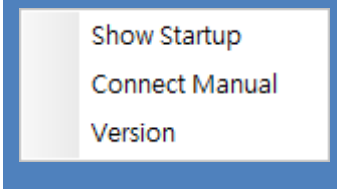
Menu Toolbar	The Menu Toolbar contains the six main functions of the Creator application. The tree structure can be expanded by clicking the menu.
Tools Toolbar	The **Tools Toolbar** contains icons for the most commonly used functions. Icons for other functions are also contained in the toolbars for the Design, Language, Status, Object, View and Toolbox functions. Hovering the mouse over the icon will display instructions for using the function.
Screen Design Area	The Screen Design Area is the main working area in the Creator application, and is used for designing, editing, and viewing the HMI screen and functions

4.2 Menu Toolbar



The **Menu Toolbar** provides access to the major features and commands for the Creator software. The following is an overview of the options available in the Menu Toolbar, together with a description of each.

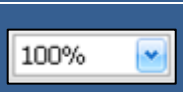
Function Menu	Item	Description
Project 	**New (Ctrl+N)**	Used to create a new project
	Open(Ctrl+O)	Used to open an existing project
	Close	Used to close the currently open project
	Save(Ctrl+S)	Used to save the currently open project
	Save as… (Ctrl+Shift+S)	Used to save the currently open project using a different name
	Exit	Used to exit the Creator software
Edit 	**Undo**	Used to reverse the previous action. Note that this option only applies to the screen design area.
	Redo	Used to restore the previous undo action. Note that this option only applies to the screen design area.
	Copy	Used to copy the selected object
	Paste	Used to paste the object that was copied or cut
	Cut	Used to cut the selected object
	Delete	Used to delete the selected object
	Search (Ctrl+F)	Used to search for a specific object or variable on the screen, or in the project.
	Group	Used to combine the selected objects as a group
	Ungroup	Used to restore elements combined in a group into individual objects
	Layer	Used to move an object to the front, the back, or up or down one layer
	Align	Used to align the objects
	Size	Used to set several objects to the same width, height, or size. (Note that the size of multiple objects can be adjusted simultaneously by clicking and holding the Ctrl key while selecting the objects, and then clicking one of the objects as the target to change all of the selected objects to the same size.)
	Screen	Used to save the screen as an a image file

Function Menu	Item	Description
View 	**Project View**	Used to open the Project function panel
	Screen View	Used to open the Screen panel
	Object View	Used to open the Object function panel
	Property View	Used to open the Property function panel
	Output View	Used to open the Output function panel
	Search View	Used to open the Search function panel
	Toolbox	Used to open the Toolbox panel
	Library	Used to open the Library function panel
	Toolbar	Used to open or hide the Toolbar
Tools 	**Compiler (F5)**	Used to compile a Creator project
	Upload (F6)	Used to upload a Creator project to the SmartView device
	Download from HMI	Used to download a Creator project from the HMI device (Available Soon)
	Simulate Offline (F8)	Used to simulate a project without needing to connect to the PAC
	Simulate Online (F7)	Used to simulate a project by directly connecting to the PAC
	Recipe Editor	Used to open a Recipe (.rp) file for editing
	Configure (Ctrl+G)	Used to configure the user interface and design surface
Window 		Used to switch between the different currently open windows ● After clicking the “Window” menu, the Startup window and a list of all the currently open windows will be displayed. Click the name of a window to switch to that window. ● The checked item is the window currently being viewed.
Help 	**Show Startup**	Used to switch to the startup screen
	Connect Manual	Used to open the Creator PLC Connection Manual, which is stored in C:\ICPDAS\Creator\Manual by default
	Version	Used to show the version information for the Creator software

4.3 Standard Toolbar

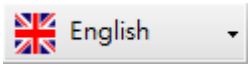
The **Standard Toolbar** contains shortcut buttons for the most commonly used tasks required by users of the Creator software. Hovering the mouse over the icon will display instructions for using the function

The following is an overview of the options available in the Standard Toolbar, together with a description of each.

	Used to create a new project		Used to delete the currently selected object
	Used to open an existing project		Used to search for an object or variable
	Used to save the current project		Used to group the currently selected objects into a single element
	Used to compile a Creator project		Used to separate previously grouped elements into individual objects
	Used to upload a Creator project to the SmartView device		The Zoom tool for the screen design area
	Used to simulate the project in Online mode		Used to increase the zoom percentage
	Used to simulate the project in Offline mode		Used to decrease the zoom percentage
	Used to reverse the previous action		Used to restore the size of the screen area to 100%
	Used to restore the previous undo action		Used to show or hide the gridlines on the screen area
	Used to copy the currently selected object		Used to align an object to the gridlines. (See Section 4.17.2 Design Surface for more details)
	Used to paste the object currently present in the clipboard		
	Used to cut the currently selected object		Used to lock an object in position

4.4 Creator Language Toolbar

The **Creator Language Toolbar** provides the ability to manage the language used for the Creator interface. Note that the user can change the language settings anytime.



The following is an overview of the language options available in the Creator Language Toolbar.

English	Sets the Creator interface display language to English
Chinese (Traditional)	Sets the Creator interface display language to Traditional Chinese
Chinese (Simplified)	Sets the Creator interface display language to Simplified Chinese

4.5 Arrange Toolbar

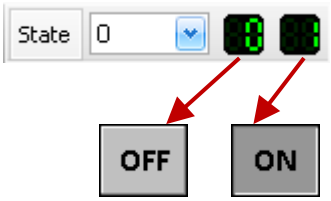
The **Arrange Toolbar** contains shortcut buttons that provide the ability to arrange and manage the positioning of screen objects contained in a project, including editing object layers, alignment, and size, etc., as well as other adjustments.

The following is an overview of the options available in the Arrange Toolbar, together with a description of each.

	Used to move the currently selected object to the top level		Used to align the vertical centers of the currently selected objects
	Used to move the currently selected object forward by one level		Used to align the horizontal centers of the currently selected objects
	Used to move the currently selected object backward by one level		Used to equally distribute the currently selected objects in a horizontal direction
	Used to move the currently selected object to the bottom level		Used to equally distribute the currently selected objects in a vertical direction
	Used to align the left edges of the currently selected objects		Used to scale the currently selected objects so that they are the same width
	Used to align the right edges of the currently selected objects		Used to scale the currently selected objects so that they are the same height
	Used to align the top edges of the currently selected objects		Used to scale the currently selected objects so that they are the same size
	Used to align the bottom edges of the currently selected objects	Note that the size of multiple objects can be adjusted simultaneously by clicking and holding the Ctrl key while selecting the objects, and then clicking one of the objects as the target to change all of the selected objects to the same size.	

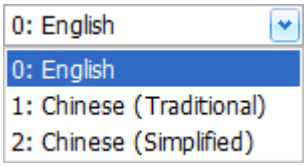
4.6 Screen State Toolbar

The **Screen State Toolbar** is used to preview all objects on the screen by changing the value of the state to 0 (OFF) or 1 (ON). See Section 8.3 Switch for more details.



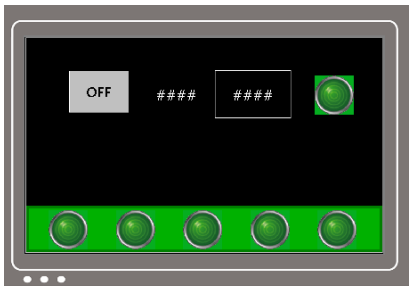
4.7 Device Language Toolbar

The **Device Language Toolbar** is used to adjust the language settings for a device when previewing the screen, and is used in situations such as previewing an object in different languages and text fonts. See Section 11.3 Language for more details.



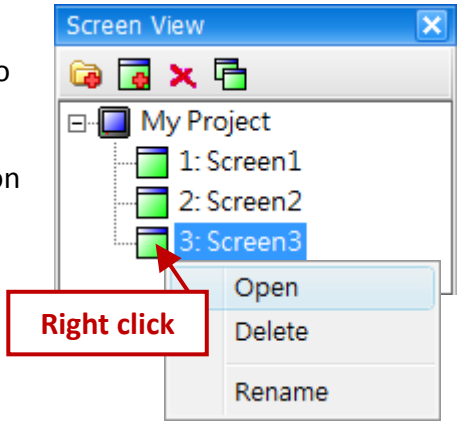
4.8 Screen Design Area

The **Screen Design Area** is the area used for designing and previewing the HMI screen that will be displayed on the SmartView device. A customized interface can be constructed for an HMI project by adding, editing, or deleting objects.



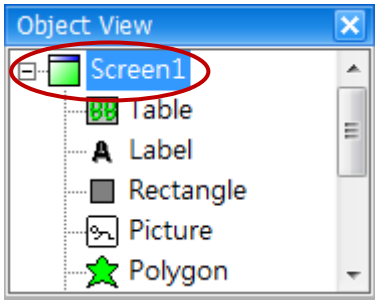
4.9 Screen View

The **Screen View** panel is used to display all the HMI screens related to the current project. HMI screens can be added, edited, deleted, or copied, and folders can be created to classify the screens. Right-click on a screen item can open, delete, or rename this screen.

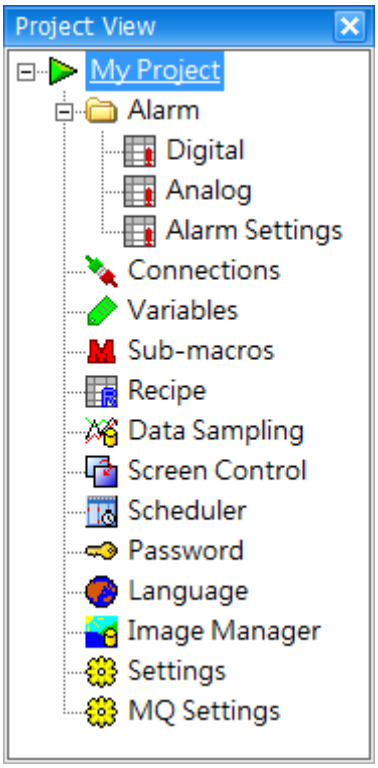


4.10 Object View

The **Object View** panel provides a list of all objects contained on the current screen. Double-click an object can locate it on the screen, and press the F2 key can rename the object.



4.11 Project View



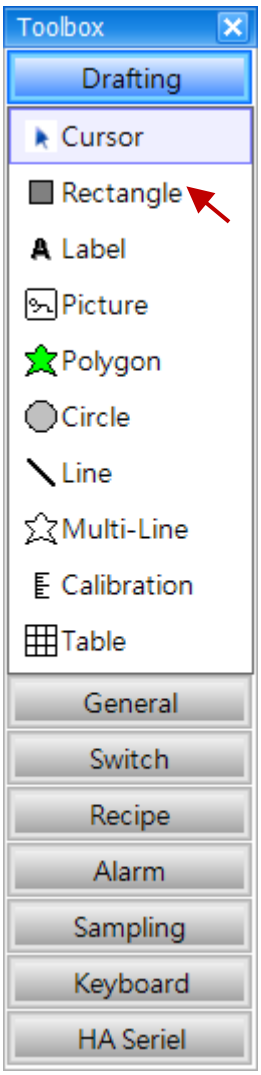
The **Project View** panel provides quick access to all the functions that are available once you have created a project. Each of which is described in more detail below.

Double-clicking any items will open their own settings screen allowing the parameters for the function to be edited. Each of which is described in more detail in the following sections.

The following is an overview of each type of function available for use in a project.

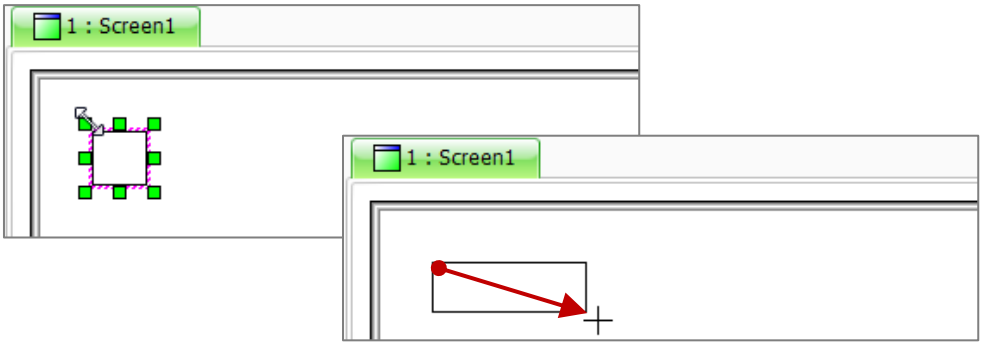
	Digital	Used to add and/or edit a Digital Alarm function		Screen Control	Used to add and/or edit a Screen Control function
	Analog	Used to add and/or edit an Analog Alarm function		Scheduler	Used to add and/or edit a Scheduler function
	Alarm Settings	Used to edit the settings for an Alarm message function		Password	Used to add and/or edit a Password function
	Connections	Used to add and/or edit a Connection function		Language	Used to configure the display languages
	Variables	Used to add and/or edit a Variable function		Image Manager	Used to manage the image library
	Sub-Macro	Used to add and/or edit a Sub-Macro function		Settings	Used to configure the properties of a project
	Recipe	Used to add and/or edit a Recipe function		MQ Settings	Used to configure the MQTT settings
	Data Sampling	Used to add and/or edit a Data Sampling function			

4.12 Toolbox



The **Toolbox** panel provides quick access to all the tools that are available once you have created a project. The Toolbox contains a number of different categories, including Drafting, General, Switch, Recipe, Alarm, Sampling, Keyboard, and etc. Each of which are described in more detail below.

To add an object to the HMI screen, click the name of the object in the relevant category of the Toolbox to select it, and then click the desired position on the Screen Design Area to place it. Or click and hold the left mouse button to drag a proper size for this object to add it on the screen, and then set the attributes in the Property View window. Refer to Chapter 8 Basic Object Usage for more details regarding the use of objects.

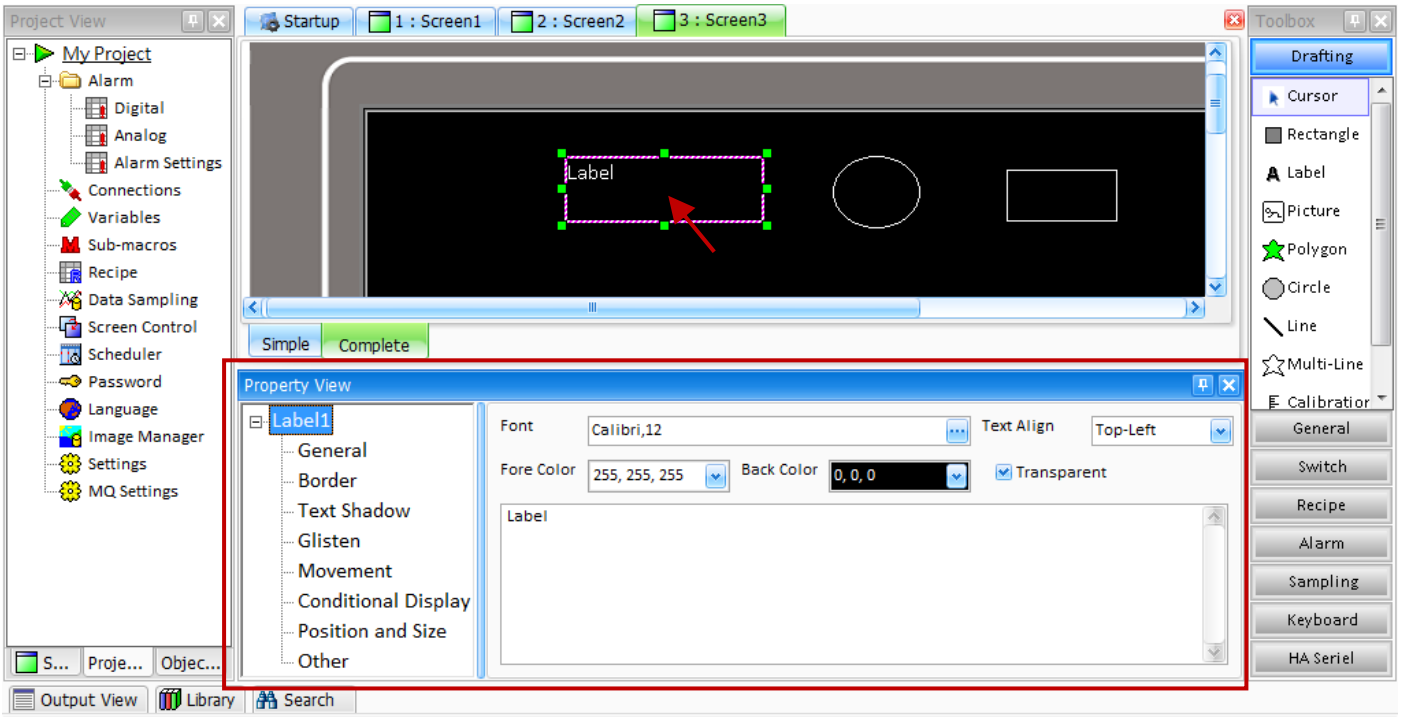


The following is an overview of each category of tool that is available from the Toolbox.

Drafting	This category includes tools that can be used to add items such as static graphics, static text, and display types, etc.
General	This category includes tools that can be used to display items such as dynamic values, dynamic graphics, the date and time, and form charts, etc.
Switch	This category includes tools that allow actions such as basic control, manual triggering, multistage control, and page control, etc.
Recipe	This category includes the operating objects needed for a specific recipe
Alarm	This category includes tools specifically designed for alarms, such as alarm viewers and alarm log operations, etc.
Sampling	This category includes tools that allow the creation of data sampling views, and real-time/history trend charts, etc.
Keyboard	This category includes the operating objects needed for the on-screen keyboard.

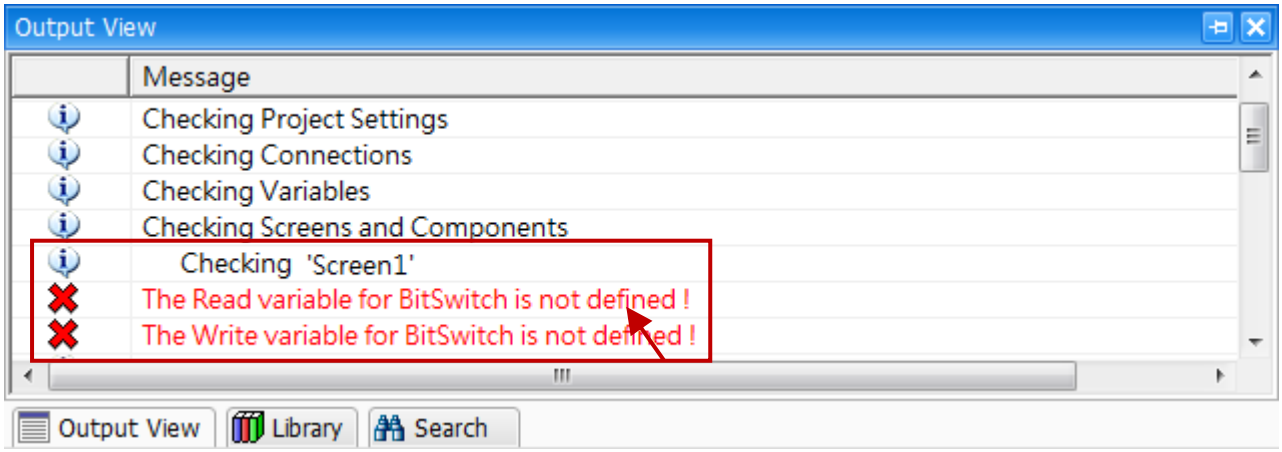
4.13 Property View

The **Property View** panel is used to edit the properties of a specific object, allowing a number of parameters and attributes to be manipulated and controlled, such as the font size and color, any images and borders, and text effects, etc., and will be automatically displayed when an object is added or selected. More details regarding the Property View for specific objects can be found in the Chapter 8.



4.14 Output View

The **Output View** panel is used to display any messages that may be generated while compiling, uploading or simulating a project. Double-click the error message can automatically locate the object on the screen, e.g., Screen1 in this example.

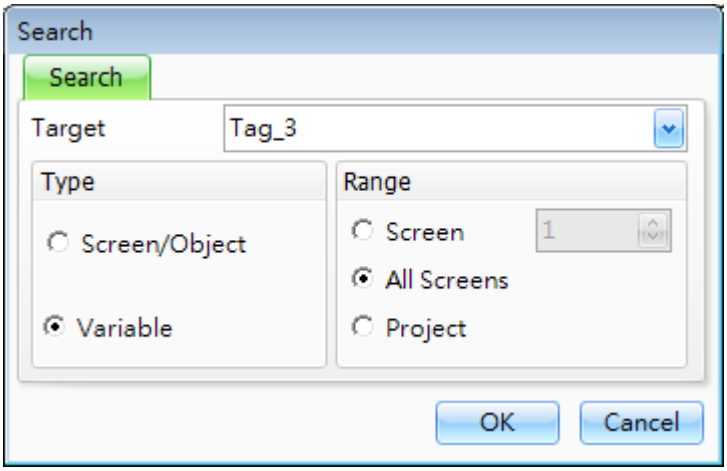


4.15 Search

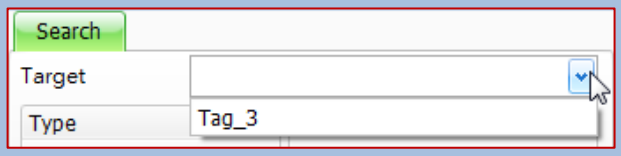
The **Search** option allows a search to be performed to locate specific screens, objects, or variables within a project based on the range set in the Search window. To access the Search function, either click the **Search** tool button, or click **Search** from the **Edit** menu in the **Menu Toolbar**.

4.15.1 Search Settings

After clicking the **Search** button, the **Search** dialog box will be displayed. Enter the relevant details and then click the **OK** button to execute the search.

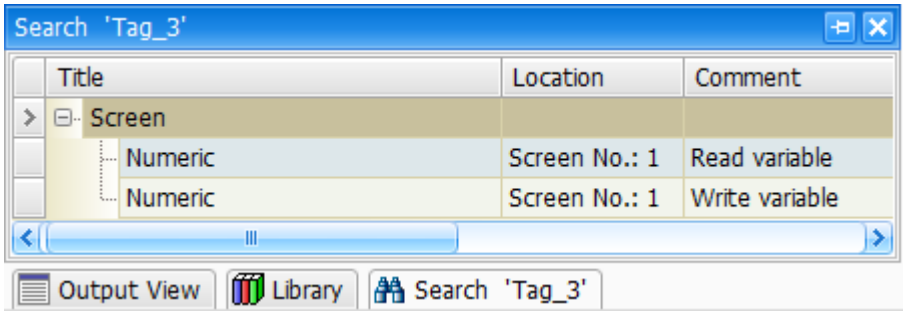


The following is an overview of the parameters contained in the **Search** dialog box.

Target	Used to enter the full name of the screen, object, or variable to be searched for. Note that the search keyword is case sensitive, so the keyword “Variable” will produce different results to a search for the keyword “variable”. And, all the keywords that have been searched for previously will be displayed in the drop-down menu. 
Type	Used to select the search type, which can be either a screen or an object, or a variable.
Range	Used to select the search range. If the “Screen” option is selected, the search will only be performed on the specified screen. Selecting “All Screens” will perform a search on all HMI screens contained in the project. Selecting “Project” will perform a search of the entire project, including the screens and variables, etc.

4.15.2 Search Results View

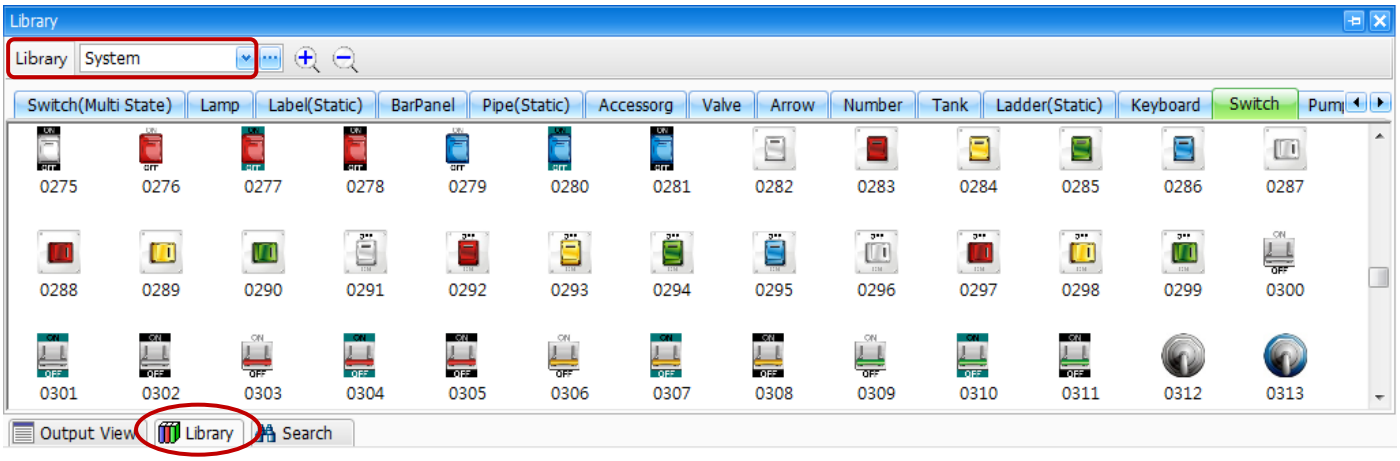
Once a search has been completed, the results will be displayed in the **Search Results** View. Double-click an entry to open the location of the item indicated in the result.



4.16 Object Library

The Creator Library provides a convenient method of managing components and functions. The Library can be accessed by clicking the **Library** button, or by clicking **Library** from the **View** menu in the **Menu Toolbar**.

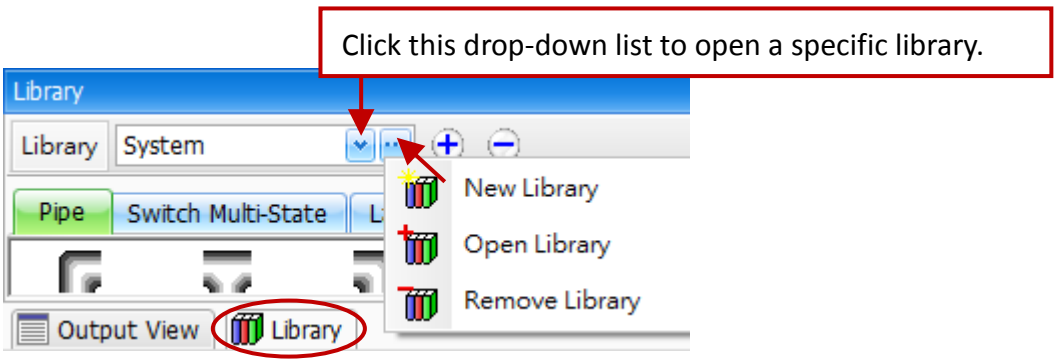
The default **System** library contains a variety of categories of pre-configured objects that can be used in a project. Custom libraries can also be created to store commonly used objects so that they can be easily added to a project the next time it is needed. This approach helps to avoid the need to repeatedly configure the properties of an object each time it is used.



4.16.1 Working with the Creator Library

To open the Creator Library view, click the **Library** tab at the bottom of the Creator interface.

The Library drop-down menu will contain a list of all currently existing libraries, including the default **System** library and any custom libraries that may have been created and saved.



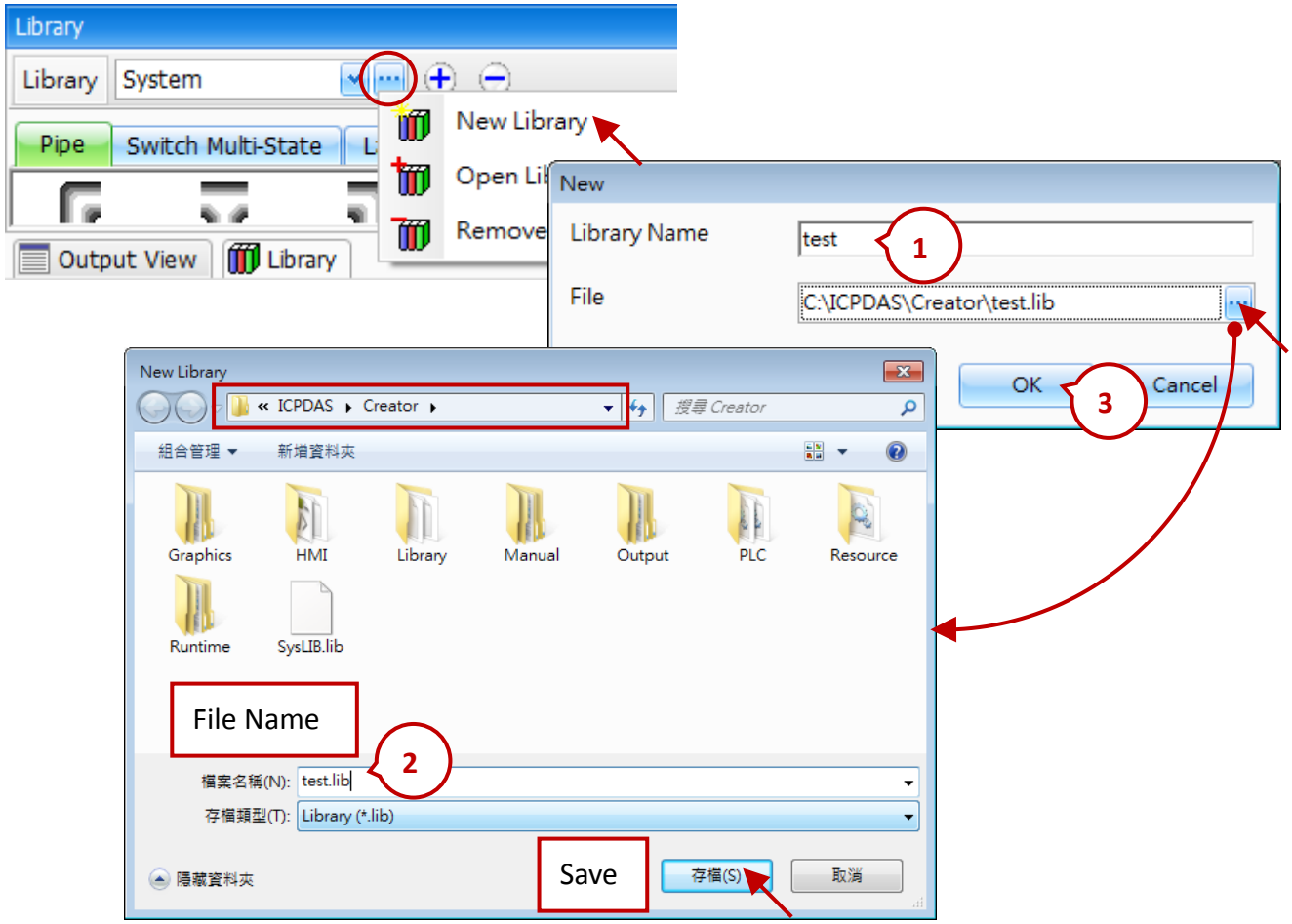
To open a specific library, select an option from the Library drop-down menu. Alternatively, click the **Tool** (…) button to create a new library, open an existing library, or remove a library.

The following is an overview of the options available in the Creator Library.

Library: System		Used to select a library, , including the default System and any custom libraries that may have been created and saved
(Tool)	New Library	Used to create a new library
	Open Library	Used to open an existing library
	Remove Library	Used to delete the currently open library, not including “System”
		Used to increase the display size of the object icons in the Library
		Used to reduce the display size of the object icons in the Library

4.16.2 Creating a new Library

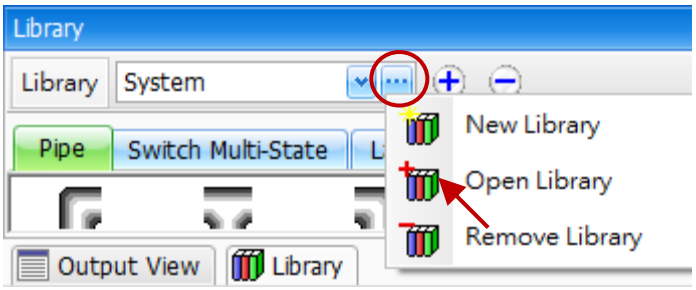
To create a new library, click the **Tool** (…) button in the Library panel and select **New Library** from the shortcut menu to open the **New** dialog box.



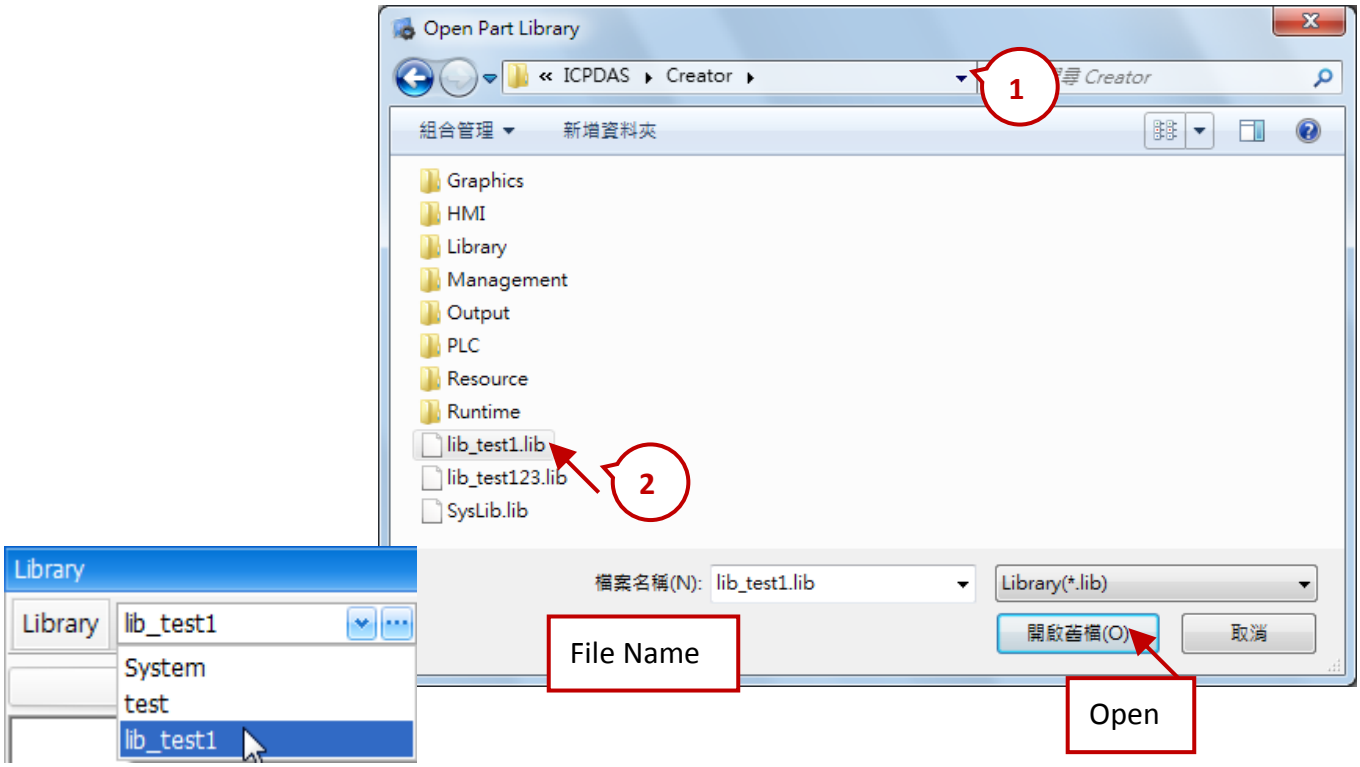
1. Enter a name for the library in the Library Name text field and click the Browse (…) button for assigning a path where the library file will be stored.
2. Locate the required folder, enter a file name, and click the Save button to create a new lib file.
3. Click the OK button in the New dialog box to create the new Library option.

4.16.3 Opening an Existing Library

To open an existing library, click the **Tool** (**…**) button in the Library panel and select **Open Library** from the shortcut menu.

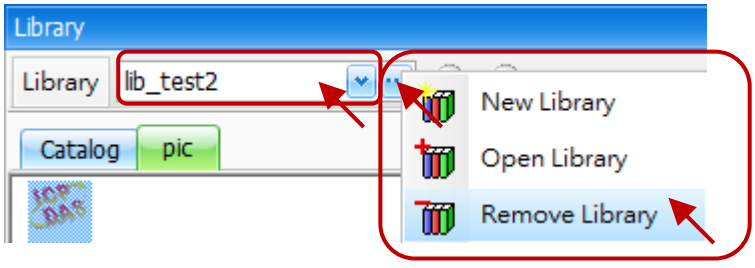


1. Locate the folder containing the desired .lib file.
2. Select the .lib file and then click the Open button. The Library will then be displayed in the Library view.



4.16.4 Removing a Library

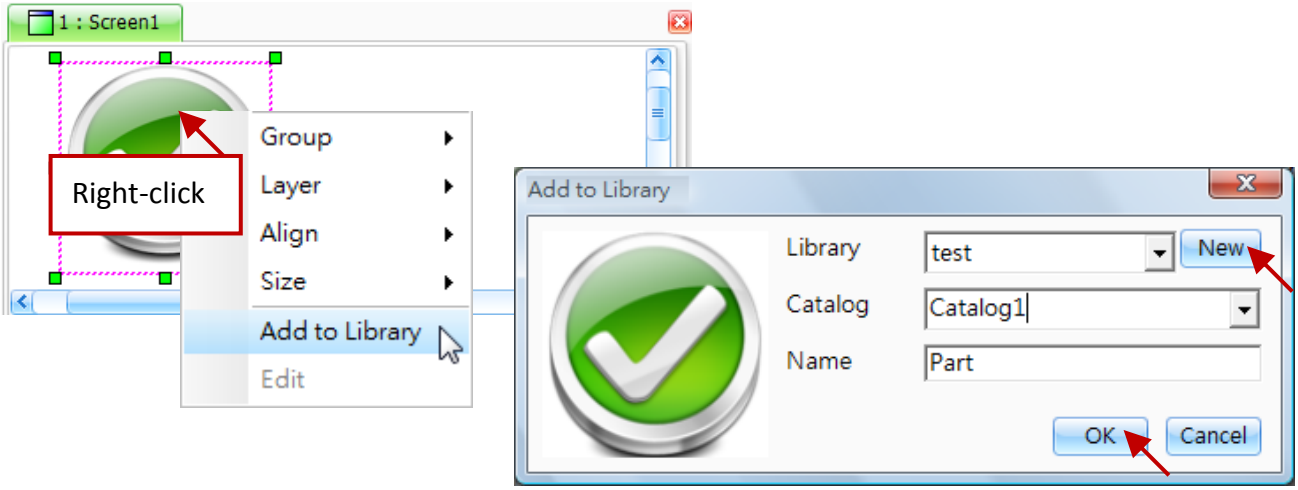
To remove an existing Library, select the name of the library from the **Library** drop-down menu, and then click the **Tool** (**…**) button and select **Remove Library** from the shortcut menu. This library option will then be deleted from the drop-down menu. Note that this library can be opened again.



4.16.5 Adding an Object to the Library

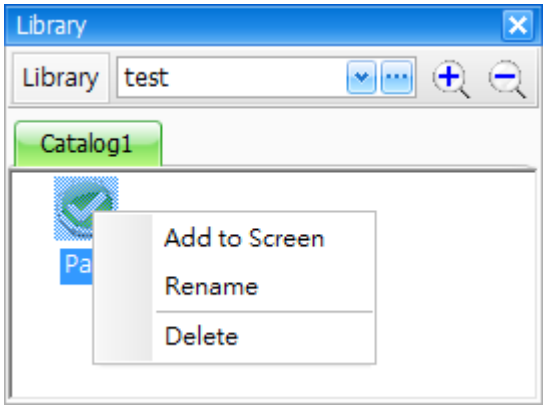
Once an object has been placed on the **Screen Design Area**, and its properties have been configured, the object can be added to a library for future use. To add an object to a library, click the object to select it, right-click the object to display the shortcut menu, and then click the **Add to Library** option.

In the **Add to Library** dialog box, click the **New** button to create a new library (or select an existing library name from the **Library** drop-down menu). Enter a **Catalog** name (or select an existing catalog name from its drop-down menu), and then enter a name for the object. Click the **OK** button to save the object in the library.



4.16.6 Managing an Object

Once an object has been added to a library, it can be inserted to the screen, renamed or deleted by right-clicking the object and selecting the required action from the shortcut menu.



Note: You can also use it by dragging the object to the Screen Design Area.

The following is an overview of the options available in the shortcut menu.

Add to Screen	Used to place an object on the Screen Design Area. The selected object will be placed in the top left corner of the currently opened screen.
Rename	Used to rename the object
Delete	Used to delete an object from the Library. Click this option to delete the object.

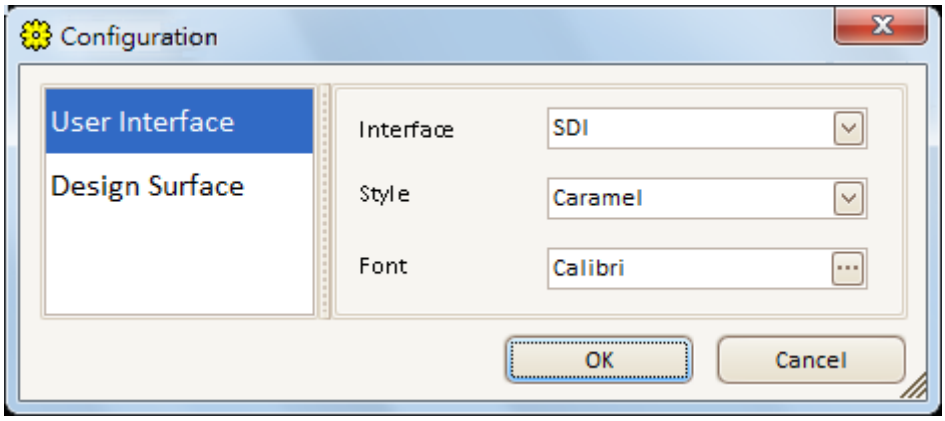
4.17 Configuration

The **Configuration** dialog box is used to select the basic options for the Creator Interface, and includes options for the User Interface and the Design Surface, each of which is described in more detail below.

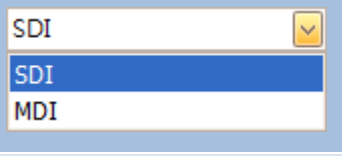
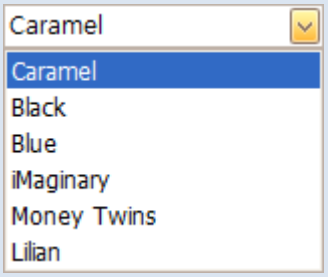
To access the Configuration page, click **Configuration** from the **Tools** menu in the **Menu Toolbar.**

4.17.1 User Interface

The **User Interface** dialog box allows parameters such as the Interface type, style, and font to be adjusted.



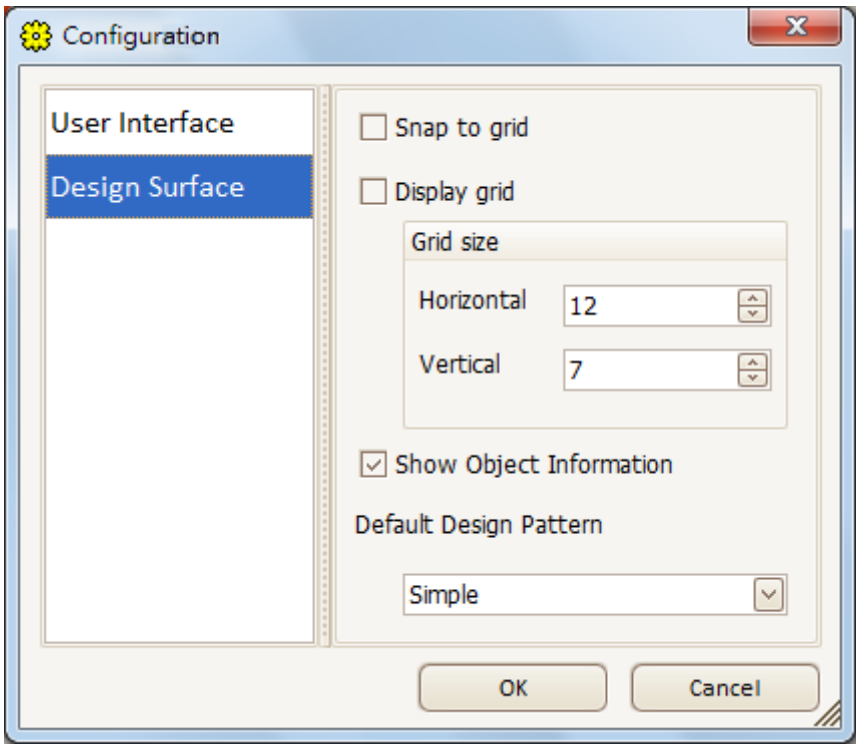
The following is an overview of the options available in the **User Interface** dialog box.

Interface	Used to change the type of interface, and offers two options: **SDI**: Single document interface **MDI**: Multiple document interface 
Style	Used to change the color and style of the user interface. Six options are available: Caramel, Black, Blue, iMaginary, Money Twins, and Lilian. 
Font	Used to configure the font displayed for the interface

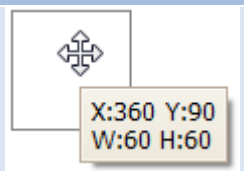
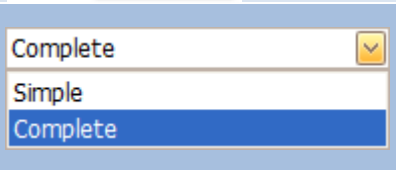
To configure the User Interface, select an interface type from the **Interface** drop-down menu, then select a style and font from the respective drop-down menus, and then click OK.

4.17.2 Design Surface

The **Design Surface** dialog box allows parameters such as the grid size, hints, and design pattern to be adjusted.



The following is an overview of the options available in the **Design Surface** dialog box

Snap to grid	Used to automatically snap to the grid when add an object to the screen.
Display grid	Used to show or hide the grid on the design screen
Grid size	Used to configure the horizontal and vertical size of the grid The valid range is 4 to 64 pixels
Show Object Information	Used to show the coordinates and size for an object when dragging it on the screen 
Default Design Pattern	Used to select the default design pattern for the screen, which can be either Simple or Complete. 

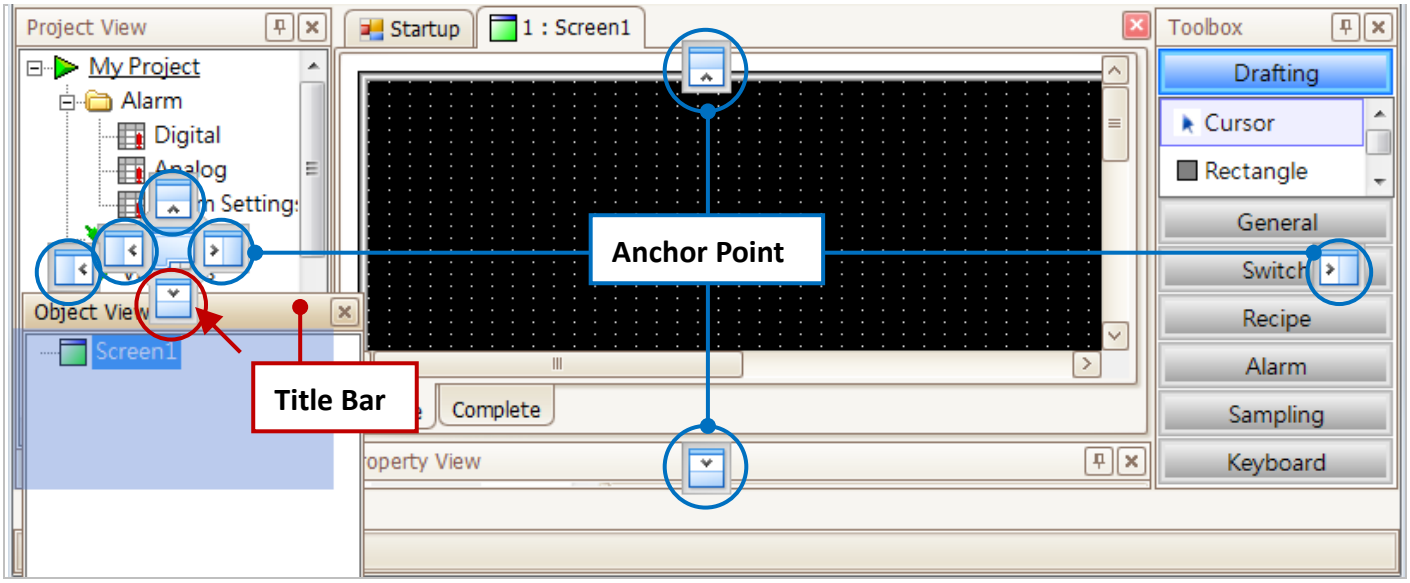
To configure the Design Surface, check the checkboxes for the relevant items, adjust the grid size or the design pattern as necessary, and then click the OK button. If the Design Pattern settings have been changed, Creator must be restarted before they will take effect.

4.18 Customizing the Interface

The Creator interface can be customized depending on individual requirements, and each panel can be positioned to suit personal preference. If the arrangement of the panels is adjusted, the current positions will be saved when Creator is closed.

4.18.1 Repositioning a Panel

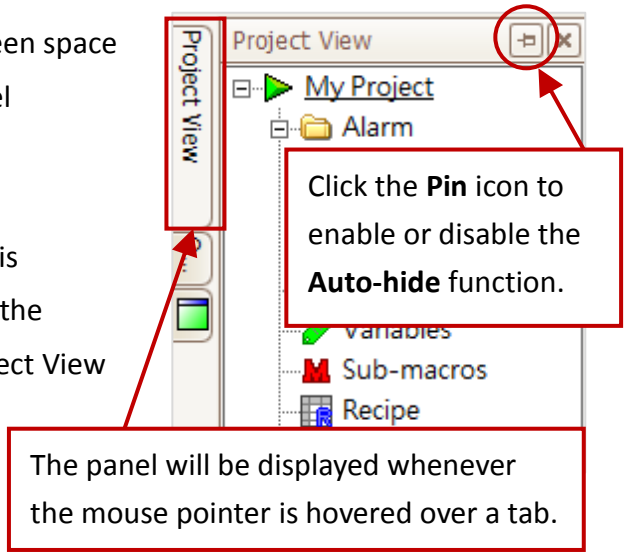
To move a panel to a new position, click and hold the **Title Bar** for the panel and drag it so that it is close to any **Anchor Point**, as illustrated below, and then release the mouse button. The panel will then be relocated to that area of the screen.



4.18.2 Auto-hiding a Panel

The **Auto-hide** function allows you to gain a little more screen space by configuring Creator to automatically hide a specific panel whenever the panel is not active.

The panel will be reactivated whenever the mouse pointer is positioned over the tab for an inactivated panel. To enable the **Auto-hide** function, click the **Pin** icon at the top of the Project View window, as illustrated.



Chapter 5 Project Setup

The following provides details of how to create a new project, including choosing the HMI model and configuring the connection parameters, etc.

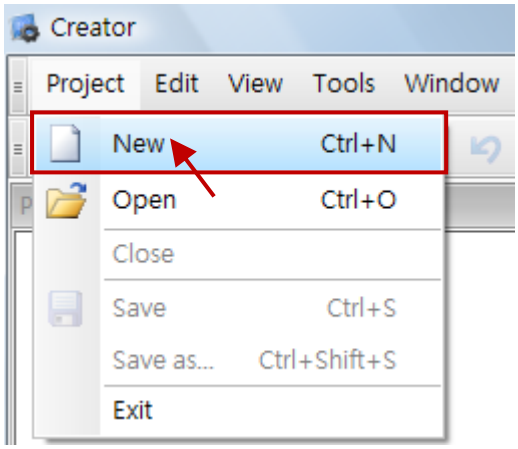
5.1 Creating a new Project

Follow the instructions described below to create a new project in the Creator software.

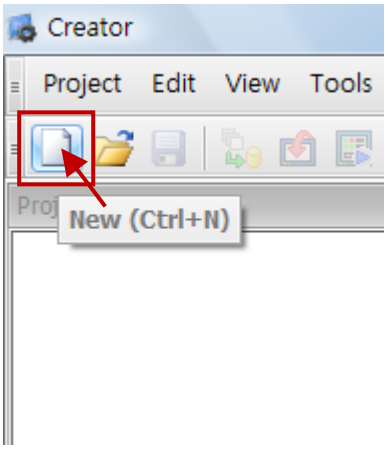
5.1.1 Create a New Project

A new project can be created using a variety of methods, each of which is described below.

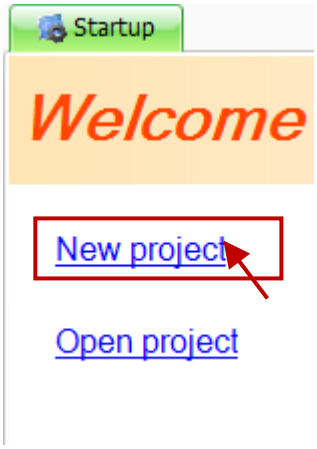
From the **Project** menu, click the **New** item to begin creating a new project. Alternatively, click the **New** button in the **Standard** toolbar, or click the **New project** item on the **Startup** screen.



or

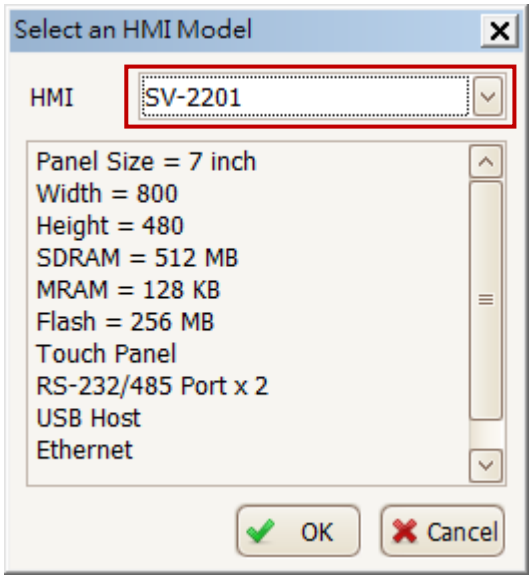


or



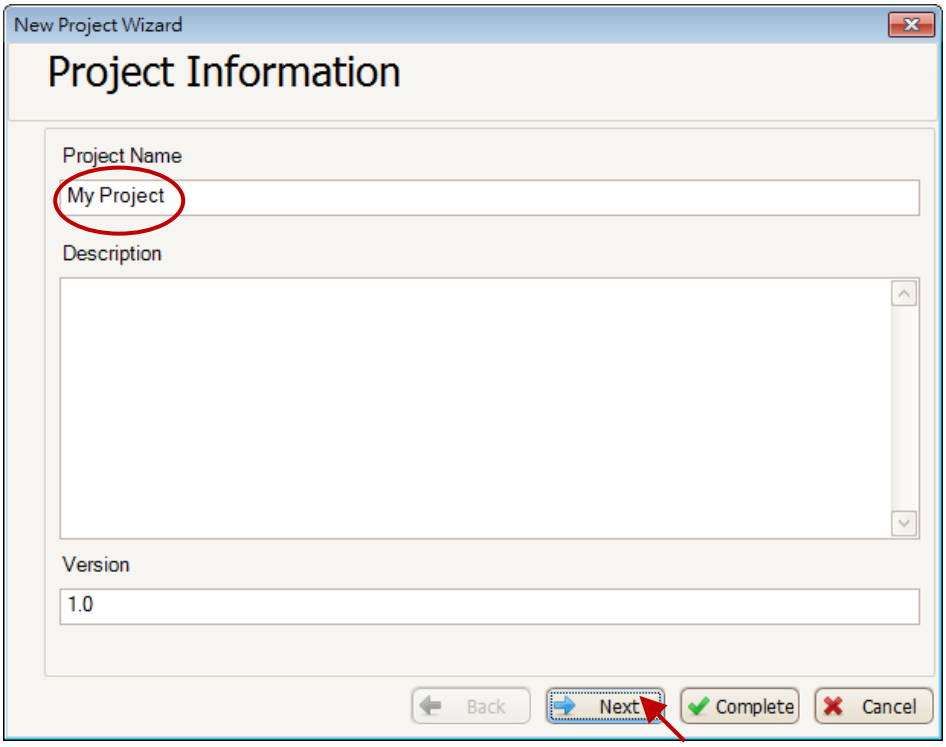
5.1.2 Select the SmartView Model

A dialog box will be displayed allowing the SmartView Series HMI Model to be selected. Choose an appropriate option from the **HMI** drop-down menu and then click the **OK** button. The **New Project Wizard** will then be displayed.



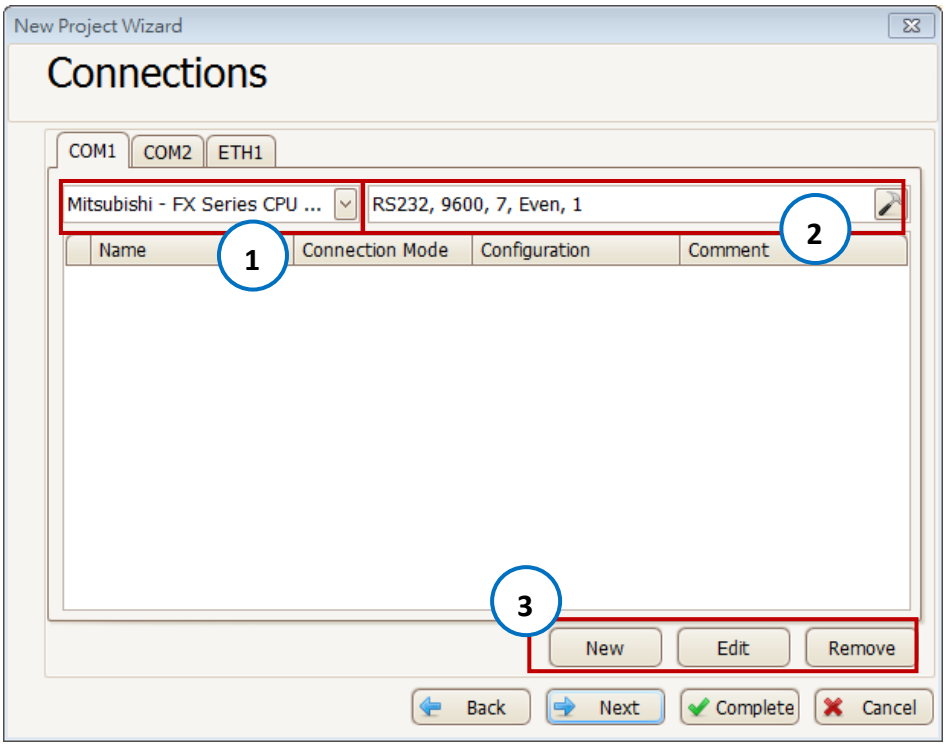
5.1.3 Enter the Project Information

In the **New Project Wizard**, enter a name for the project in the **Project Name** field, and then enter an appropriate description and version number, if necessary, in the respective fields. Click the **Next** button to continue.

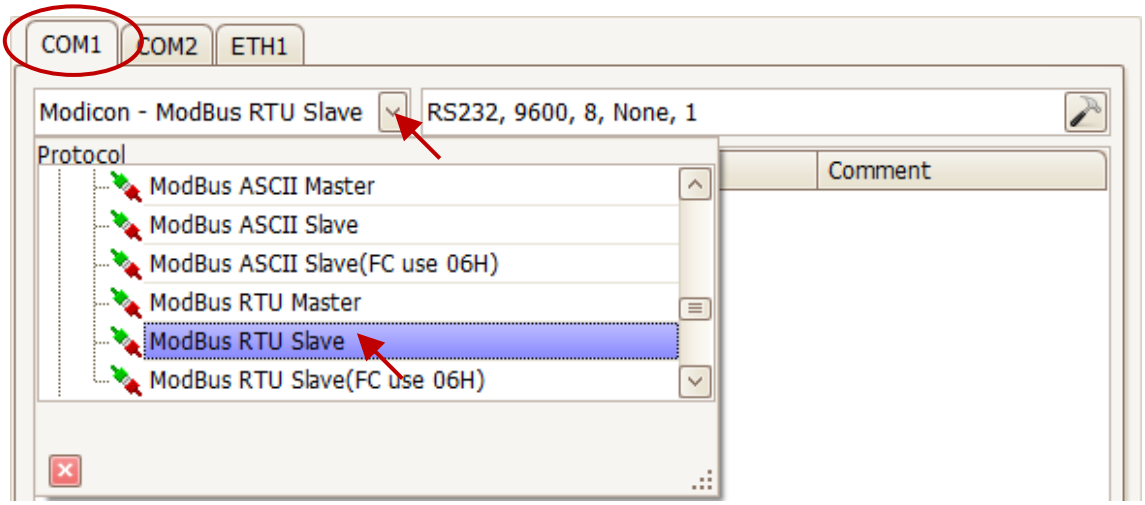


5.1.4 Configuring the Connection

On the "**Connections**" dialog box, configure the communication protocol and other parameters for the connection by entering the relevant information in the respective fields, each of which is described in more detail below.

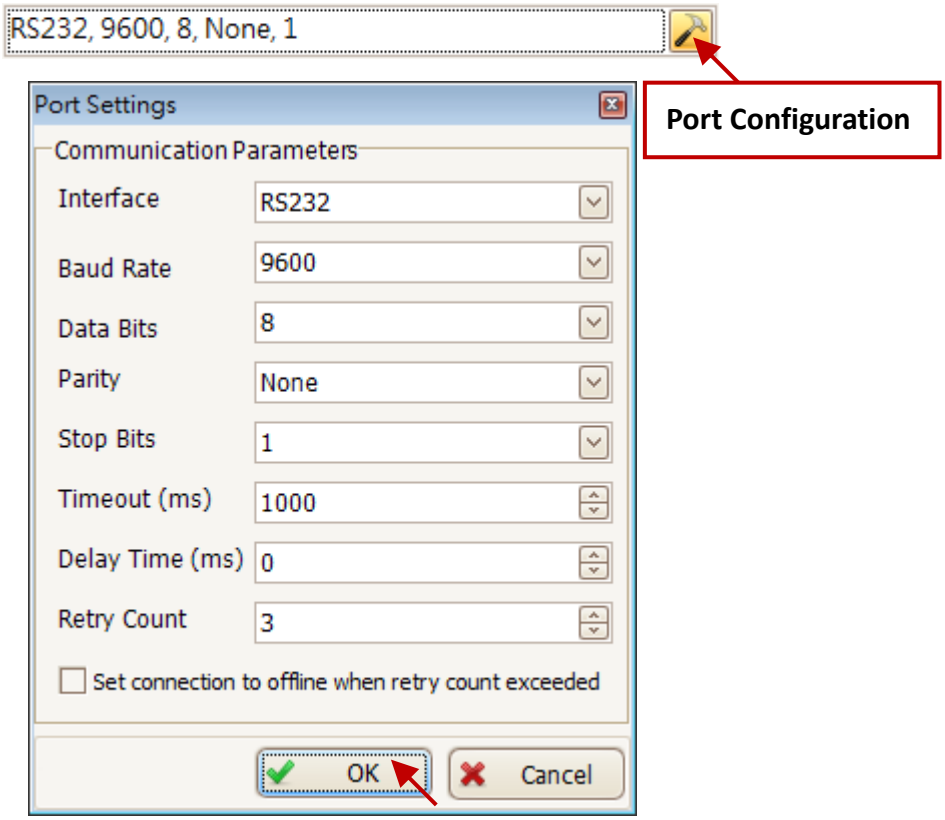


1. Select the communication protocol for the SmartView to connect to PLC device by first selecting the tab for the COM port, and then selecting the appropriate option from the drop-down menu.



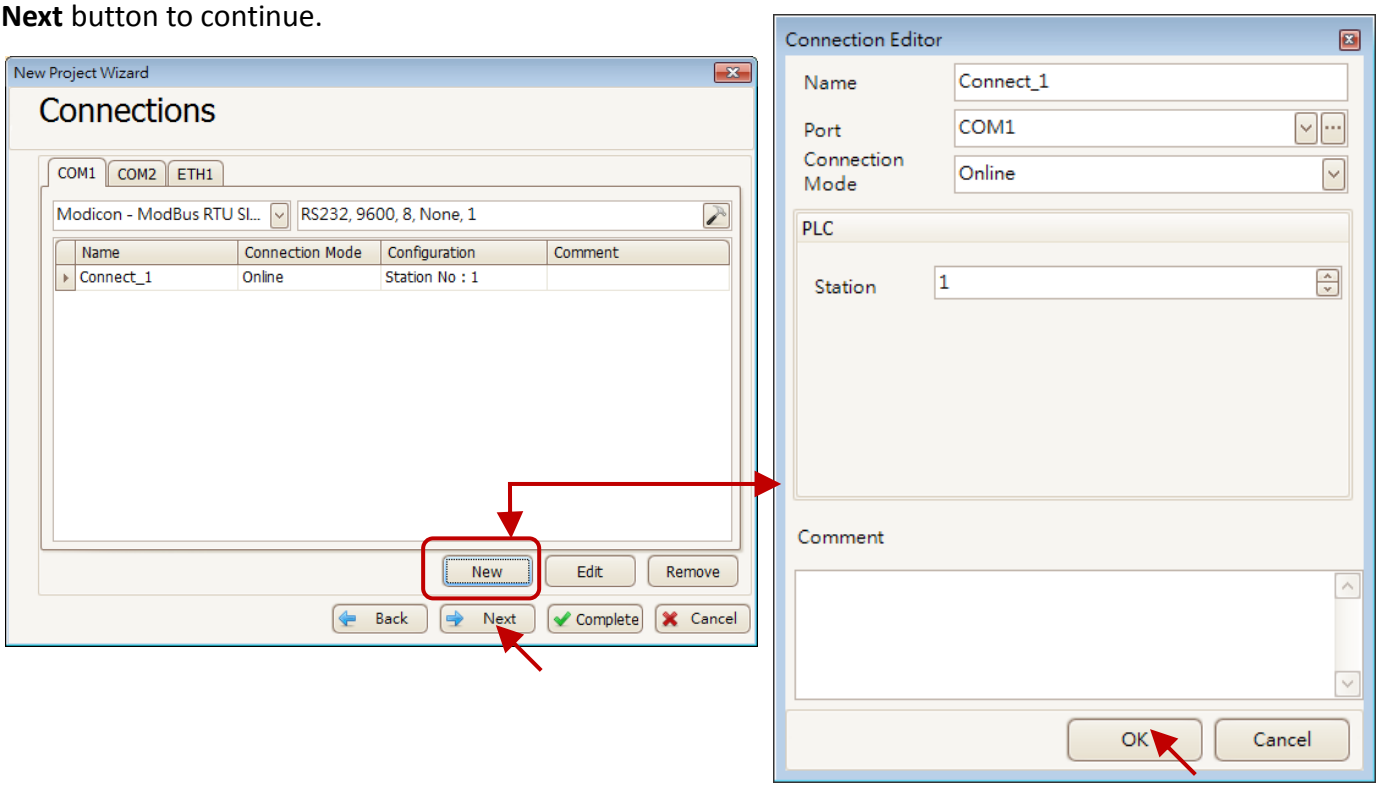
2. Configure the communication parameters to be used for the port by clicking the **Port Configuration** button to open the **Port Settings** dialog box.

   In the **Port Settings** dialog box, select the required values for the device to be connected from the respective drop-down menus, then click the **OK** button to continue.



3. Create a new connection

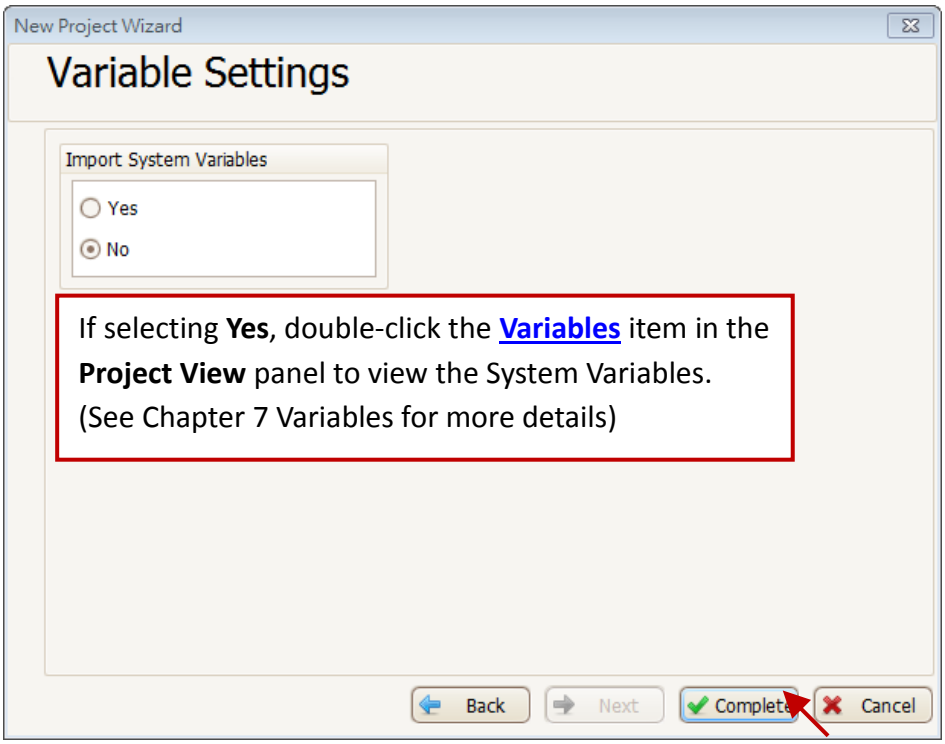
Initially, no connections will be listed in the **Connections** dialog box, so a new connection must be created. Click the **New** button to open the **Connection Editor** dialog box. In the **Connection Editor** dialog box, enter a name for the new connection and configure the communication parameters, including the COM Port and the Station number (Net ID), and then click the **OK** button to close the dialog box. Once the connection settings have been completed, the new connection will be listed in the **Connections** dialog box. Click the **Next** button to continue.



5.1.5 Importing System Variables

After clicking the **Next** button from the previous step, the **Variable Settings** dialog box will be displayed.

To import any system variables, click the **Yes** option button in the **Import System Variables** section of the **Variable Settings** page, or click **No** if the variables are not going to be imported, and then click the **Complete** button to finish creating the project.



5.2 Compiling the project

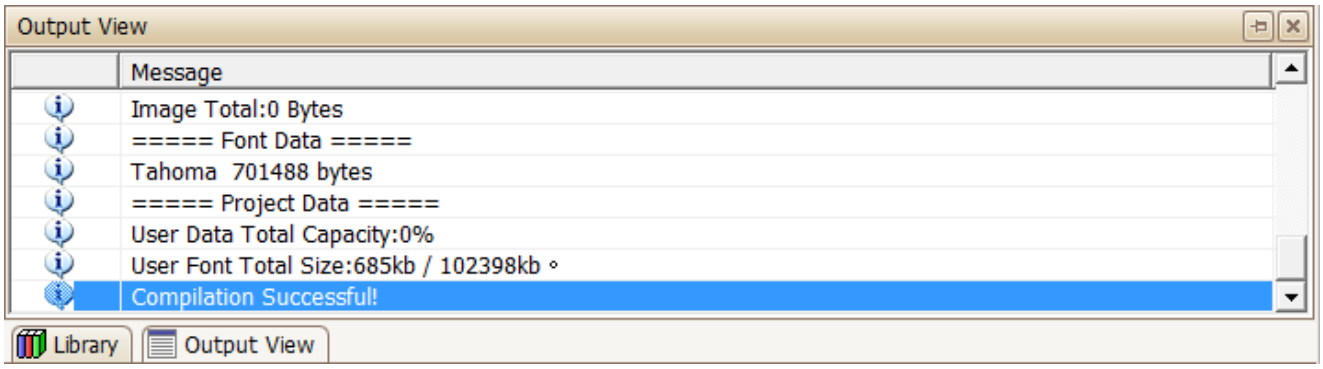
Before a project can be transferred and used on a SmartView device, it must be compiled. The following provides instructions for how to perform this task.

5.2.1 Launch the Compiler

To begin compiling the project, either click the Compiler button in the **Standard Toolbar** to launch the built-in compiler or, alternatively, click the **Compiler** item from the **Tools** menu.

5.2.2 Display the Compilation Results

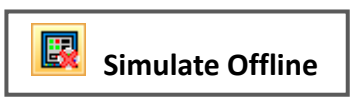
As the project is being compiled, the results of the compilation process will be automatically displayed in the pop-up **Output View** window. If there are any errors in the code, an error message will be displayed in red. Double-click the error message to automatically locate the object on the screen so that it can be corrected. See Section 4.14 Output View for more details. Once the error has been corrected, compile the project again. If further errors are encountered, repeat the process until no errors are found.



Once the project is successfully compiled, the message "Compilation Successful" will be displayed at the end of the output stream.

5.3 Simulating a Project Offline

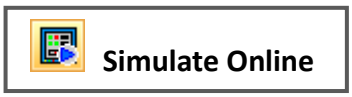
Creator includes a built-in simulator that allows a project to be simulated on a Host PC without the need to be physically connected to the destination PAC controller by using the Host PC as an HMI platform. This can be particularly advantageous when the target device is not available while developing a project.



To begin the simulation, either click the **Simulate Offline** button in the **Standard Toolbar** to launch the simulator or, alternatively, click the **Simulate Offline** item from the **Tools** menu.

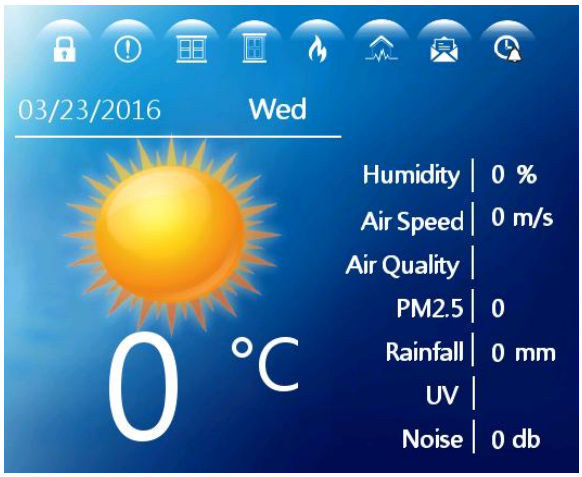
5.4 Simulating a Project Online

Creator includes a built-in simulator that allows a project to be simulated on a Host PC that is already connected to the destination PAC by using the Host PC as an HMI platform. This can be advantageous when the target device is connected to the Host PC while developing a project. The online simulation can be performed continuously for up to 30 minutes.

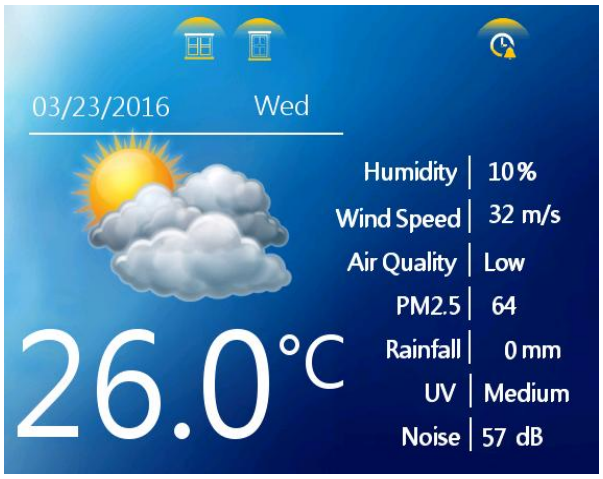


To begin the simulation, either click the **Simulate Online** tool icon button in the **Standard Toolbar** to launch the simulator or, alternatively, click the **Simulate Online** item from the **Tools** menu.

(**Simulate Offline**)



(**Simulate Online**)



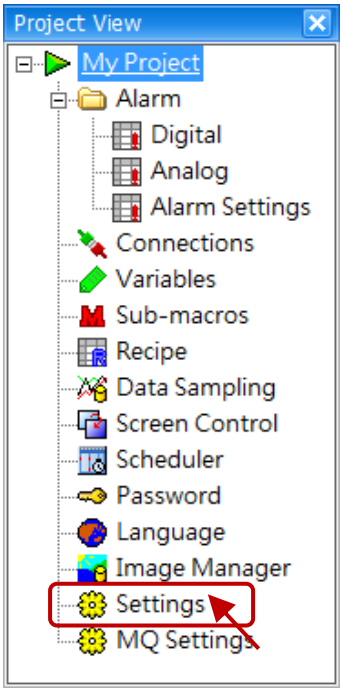
5.5 Uploading a Project

Creator provides the ability to upload a project to a SmartView device via the Ethernet, or via a Serial port (See Section 5.6.5 Upload Tab for more details about the settings).

To upload a project to a SmartView device, either click the **Upload** button in the **Standard Toolbar,** or, alternatively, click the **Upload** item from the **Tools** menu.

Once uploaded, the HMI pages will be displayed on the screen of the SmartView device.

5.6 Configuring the Project Settings

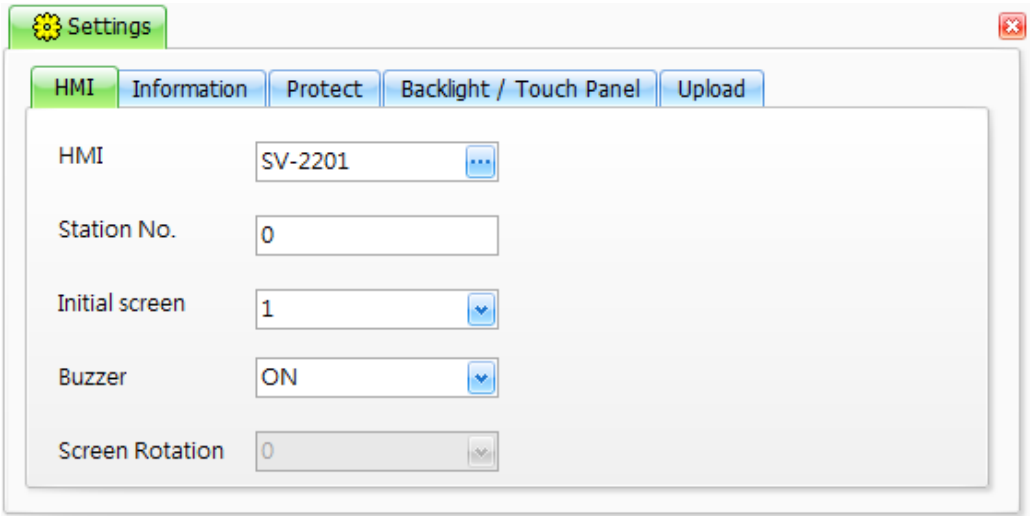


The **Settings** function in the **Project View** panel allows the settings to be configured for items such as the HMI (Model and Station No., etc.), the project information, password protection, the touch panel and backlight parameters, as well as the project upload parameters, including the connection, mode, IP address and font etc.

To configure the settings for a project, click the **Settings** function from the **Project View** panel. The **Settings** page contains five tabs, HMI, Information, Protect, Backlight/Touch Panel, and Upload. More details related to these tabs are provided in the following sections.

5.6.1 HMI Tab

The HMI tab contains options that allow the settings for the target HMI device to be configured, each of which are described in more detail below.



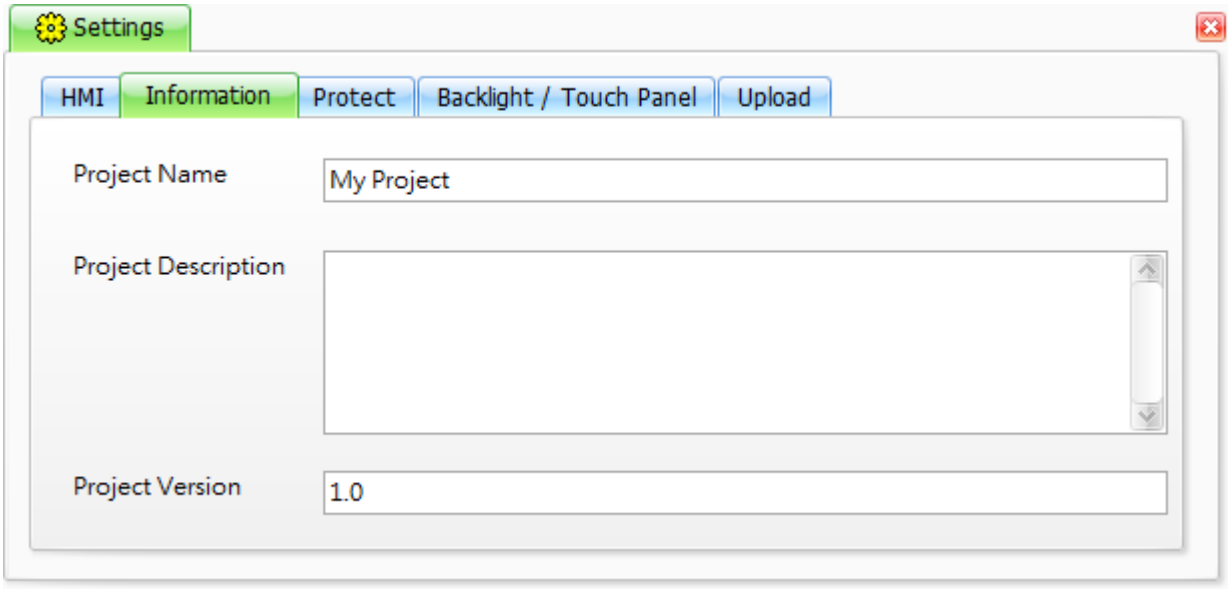
The following is an overview of the options available in the **HMI** tab.

HMI	Used to specify the target model for the project
Station No.	Used to specify the local station number for the PAC, and the valid range is 0 to 255
Initial Screen	Used to specify which screen will be set as the home page for the HMI project
Buzzer	Used to specify the initial state of the buzzer switch
Screen Rotation	Used to specify whether or not the screen can be rotated. Note that this function will be available soon.

To configure the settings for a specific HMI device, select a target model from the **HMI** drop-down menu, and then enter the device ID in the **Station No**. field. Select the required values from the **Initial Screen**, and **Buzzer** drop-down menus to complete the settings.

5.6.2 Project Information Tab

The **Information** tab contains options that allow important information about the project to be specified, each of which is described in more detail below.



The following is an overview of the options available in the **Information** tab.

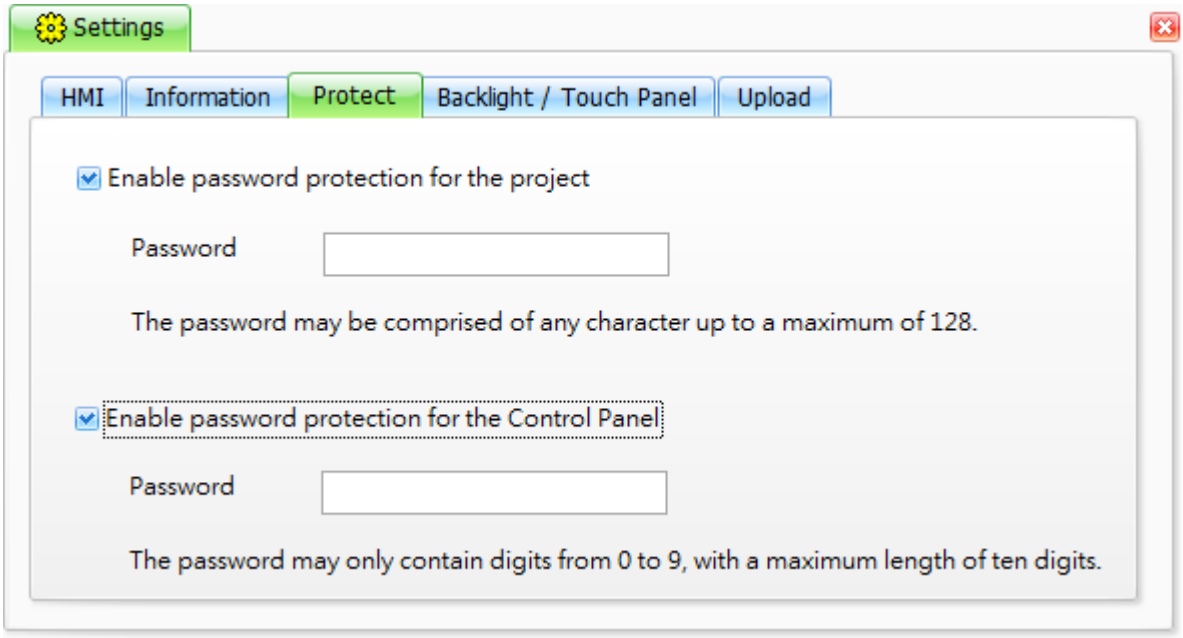
Project Name	Used to assign a name for the project
Project Description	Used to enter a description for the project
Project Version	Used to specify a version number for the project

Enter a name for the project in the **Project Name** field, and then, if necessary, enter an appropriate description and version number in the respective fields.

Note that completing these details is not compulsory, but doing so may be helpful for the future development of a project.

5.6.3 Protect Tab

The **Protect** tab contains options that allow password protection to be added to the project, or to enable protection that restricts unauthorized access to the Control Panel.



The following is an overview of the options available in the **Protect** tab.

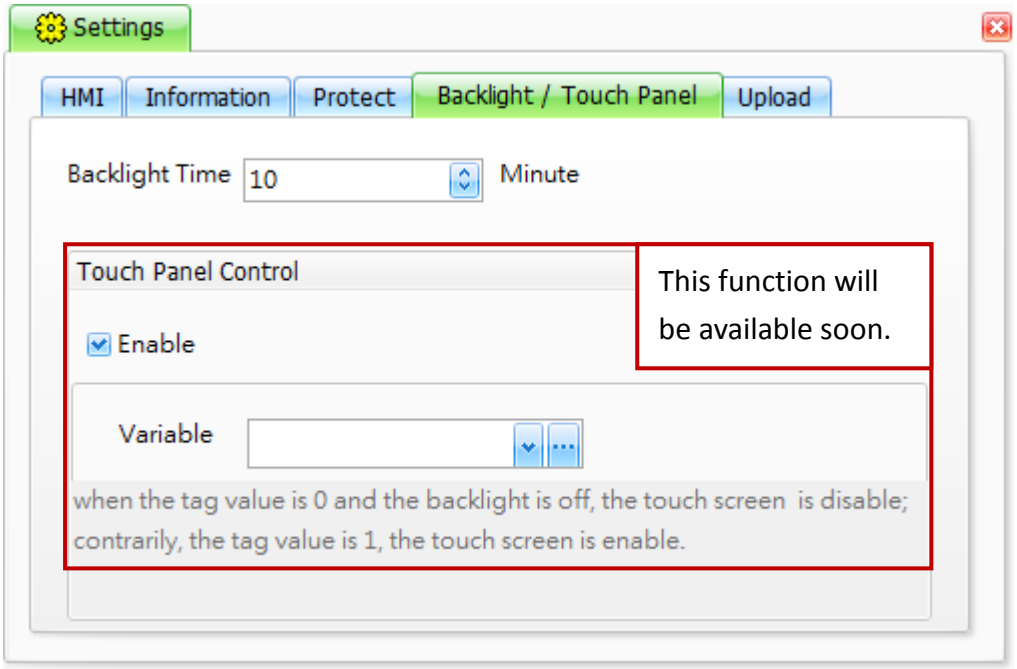
Enable password protection for the project	Used to enable or disable password protection for the project
Password	Used to assign a password for the project
Enable password protection for the Control Panel	Used to enable or disable password protection for accessing the Control Panel
Password	Used to assign a password to prevent unauthorized access to the Control Panel

To enable password protection for the project, check the checkbox and enter a password. Note that the password may be comprised of any character, with a maximum of 128 characters.

To restrict unauthorized access to the Control Panel, check the checkbox and enter a password. Note that this password may only be comprised of digits from 0 to 9, with a maximum of ten digits.

5.6.4 Backlight/Touch Panel Tab

The **Backlight/Touch Panel** tab contains options that allow the duration that the backlight on an HMI device is active to be adjusted, or to control the touch screen functionality by using a variable. For more details related to the usage of variables, see Chapter 7 Variables.

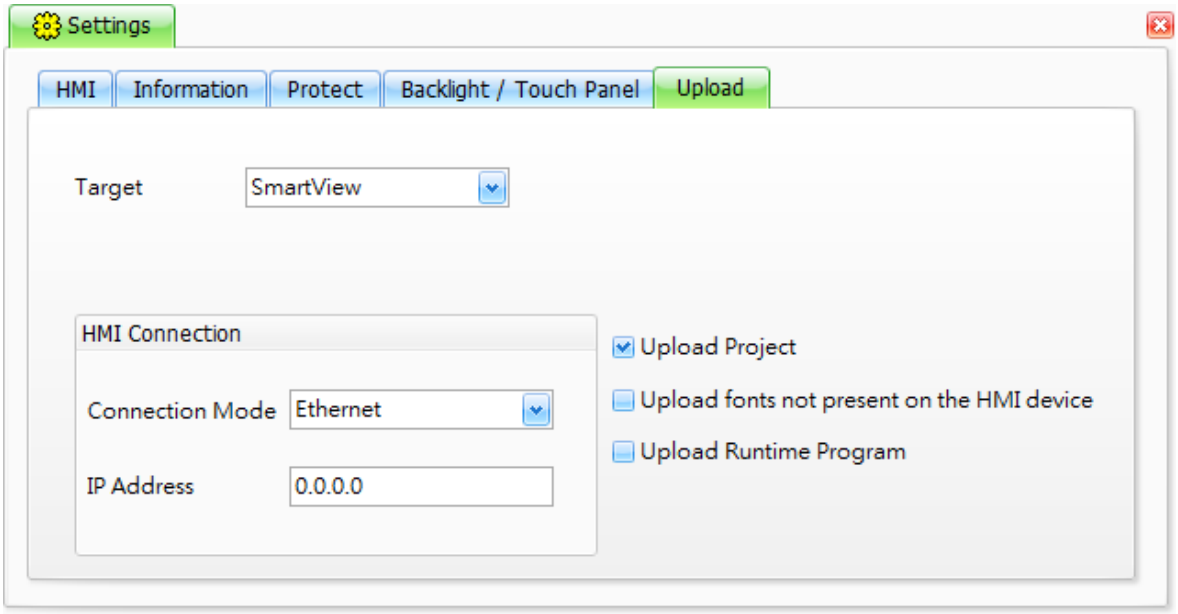


The following is an overview of the options available in the **Backlight/Touch Panel** tab.

Backlight Time		Used to specify the duration that the backlight is active
Touch Panel Control	**Enable**	Used to enable or disable the control functions for the touch panel
	Variable	Used to assign the control variables, where: 0: The Backlight is OFF 1: The Backlight is ON For more details related to the usage of variables, see Chapter 7 Variables.

5.6.5 Upload Tab

The **Upload** tab contains options that are used to configure the connection for an HMI project, as well as to enable or disable the various options relating to the project. See Section 5.5 Uploading a Project for more details.



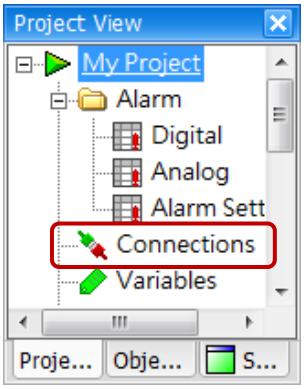
The following is an overview of the options available in the **Upload** tab.

Target		Used to select the Upload target device, which can be a SmartView device
HMI Connection	**Connection Mode**	Used to select the connection mode, which can be Ethernet
	IP Address	Used to specify the IP Address of the target device. Note that this option is only applicable to Ethernet connection mode
Upload Project		Used to specify whether or not the project should be uploaded
Upload fonts not present on the HMI device		Used to specify whether or not any fonts that are not present on the HMI device should be uploaded with the project
Upload Runtime Program		Used to specify whether or not the latest Runtime program should be uploaded with the project if the runtime program does not currently exist on the target

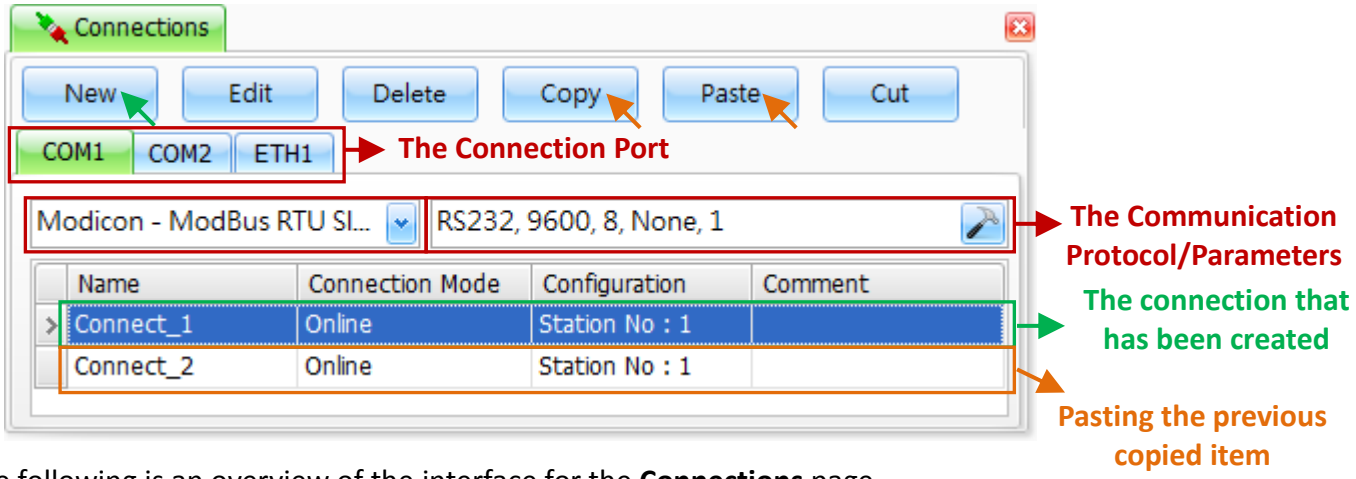
Click the **Upload** tab in the **Settings** dialog box and then select the target and the connection mode from the respective drop-down menus. Enter the IP address in the **IP Address** text field and enable or disable the various options relating to the project as necessary.

Chapter 6 Connections

The **Connections** function is used to create, edit, and manage the communication ports, protocols and parameters that are used for a specific project, which will be described in more detail below. The available COM ports that will be displayed completely depend on the specific SmartView model being used. The protocol and parameters for the SmartView ports used to connect to the PLC controller should be configured according the requirements of the specific PLC.



To access the **Connections** page, double-click the **Connections** function in the **Project View** panel. The following is an overview of the **Connections** interface.

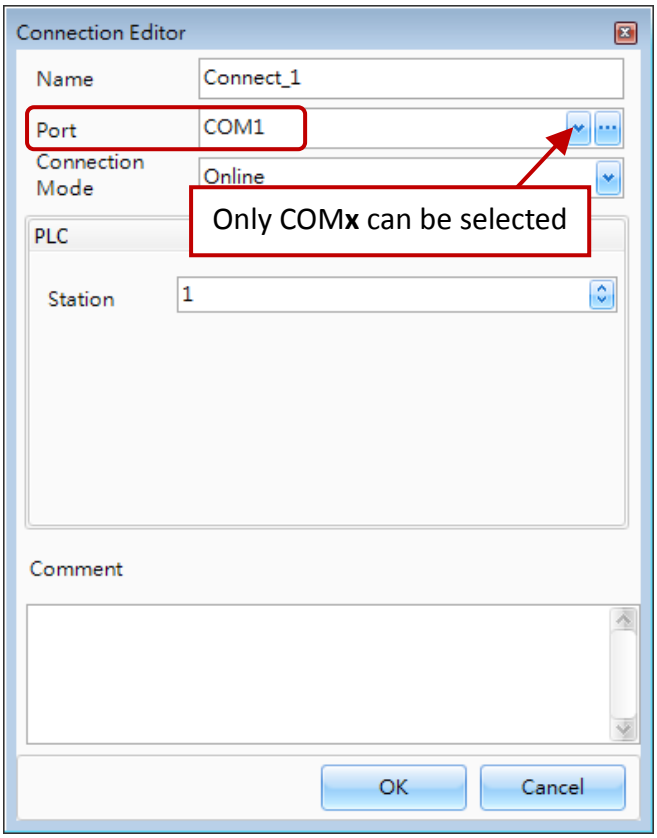


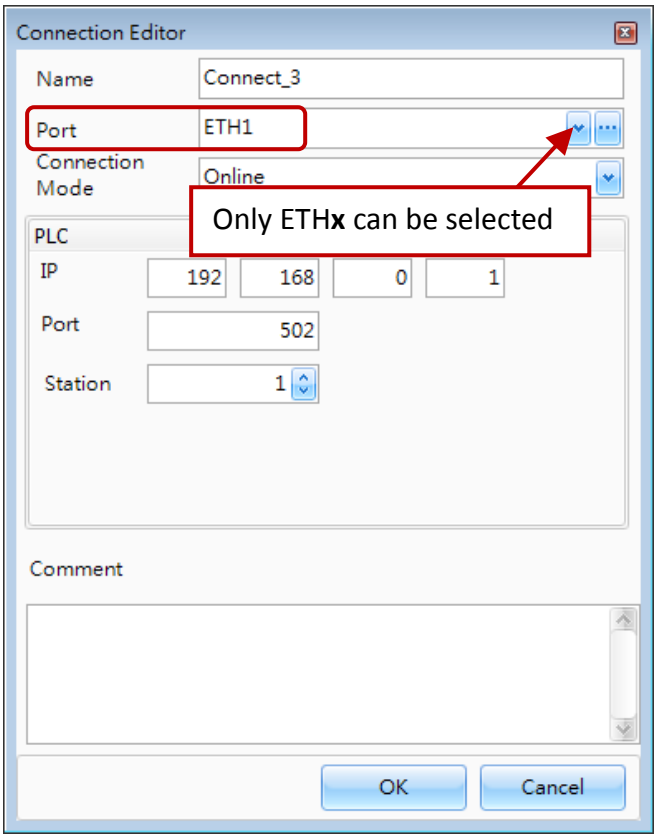
The following is an overview of the interface for the **Connections** page.

New	Used to create a new connection. See Section 6.1 Connection Editor for more details
Edit	Used to open the Connection Editor dialog box for the selected connection allowing the details to be edited. See Section 6.1 Connection Editor for more details
Delete	Used to delete the selected connection item(s)
Copy	Used to copy all settings for the selected connection item(s) (without the name)
Paste	Used to paste the connection item(s) that was copied
Cut	Used to cut the selected connection item(s). Note that if you click the Paste button later, the connection information will be pasted to the last item.
Connection Port	Used to specify the port type to be assigned to the connection, which can be either COM (Serial Port) or ETH (Ethernet Port). The buttons that will be displayed depend on the model selected. **Note that if the model used for the project is changed, any settings that are not supported by the new model will be deleted.**
Communication Protocol	Used to specify the communication protocol to be assigned to the connection. Note that only one protocol can be selected for each port, with the following limitations: ● A maximum of **32** connections can be created when connected to a Slave device. ● A maximum of **1** connection can be created when connected to a Master device.
Communication Parameters	Used to specify the communication parameters to be assigned to the connection. Click the **Port Configuration** button ( ) to open the Port Settings dialog box.

6.1 Connection Editor

The **Connection Editor** dialog box is used to assign the properties for a new connection, or to edit the details for an existing connection. To open the **Connection Editor** dialog box, click either the **New** or the **Edit** button in the **Connection** page, as illustrated in the diagram above.





The following is an overview of the options available in the **Connection Editor** dialog box.

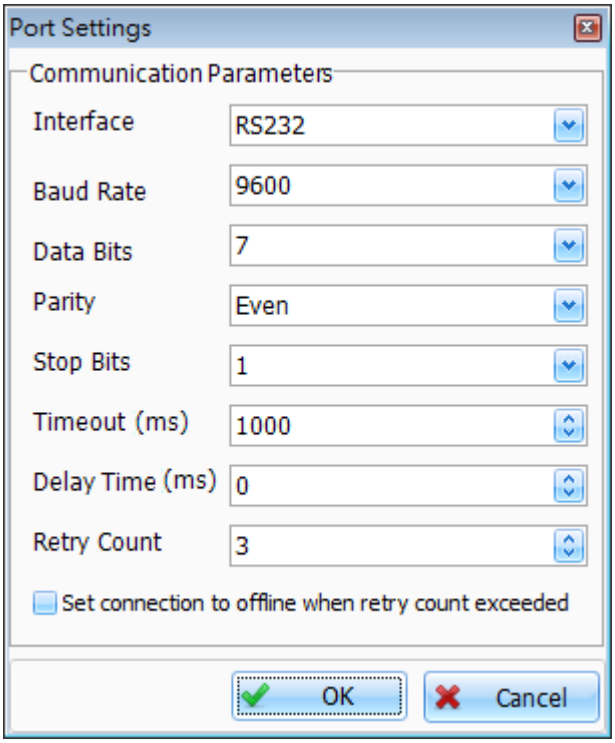
Name		Used to assign a name for the connection
Port		Used to specify which port is assigned to the connection. Note that the settings cannot be directly switched between COM ports and Ethernet ports
Connection Mode		Used to specify the initial connection mode when uploading the settings to the HMI device: **Online**: The initial connection mode is “Online”. **Offline**: The initial connection mode is “Offline”. Click the **Port Configuration** button (…) to open the Port Settings dialog box. Note that the status of the connection mode can be changed using an object function. Refer to Section 8.3.4 Connection Button for more details
PLC	**Station**	Used to specify the Station ID of the remote device
	IP Address	Used to specify the IP address of the remote device
	Port	Used to specify the TCP port number to be used, normally set as 502
Comments		Used to add any comments related to the connection

6.2 Communication Parameters

The **Port Settings** dialog box is used to assign the properties of a new connection, or to edit the details for an existing connection. The parameters displayed in the dialog depend on the type of connection, as illustrated below.

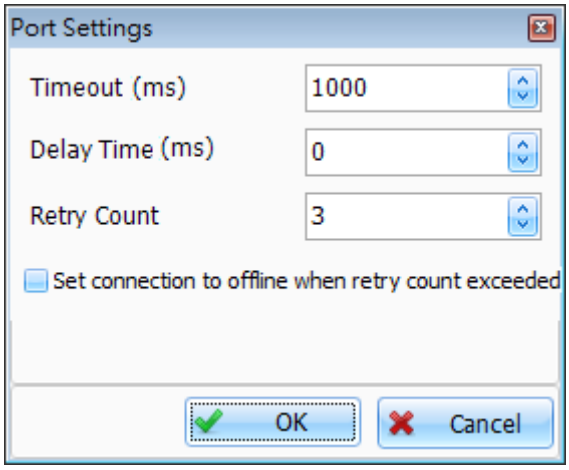
The following is an overview of the options available in the **Port Settings** dialog box.

Serial Port



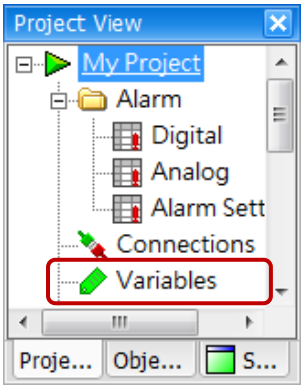
Interface	Used to specify the interface type and can be either RS232, RS422, or RS485
Baud Rate	Used to specify the Baud Rate to be used for the port
Data Bits	Used to specify the Data length to be used for the port
Parity	Used to specify the checksum mode to be used for the port
Stop Bits	Used to specify the Stop bit length to be used for the port
Timeout	Used to specify the timeout duration before a connection failure occurs
Delay Time	Used to specify the Delay time between commands
Retry Count	Used to specify the number of connection attempts that should be made when the connection fails
Set connection to offline when retry count exceeded	
Used to automatically set the connection to Offline Mode if the number of connection attempts exceeds the configured retry count value	

Ethernet Port



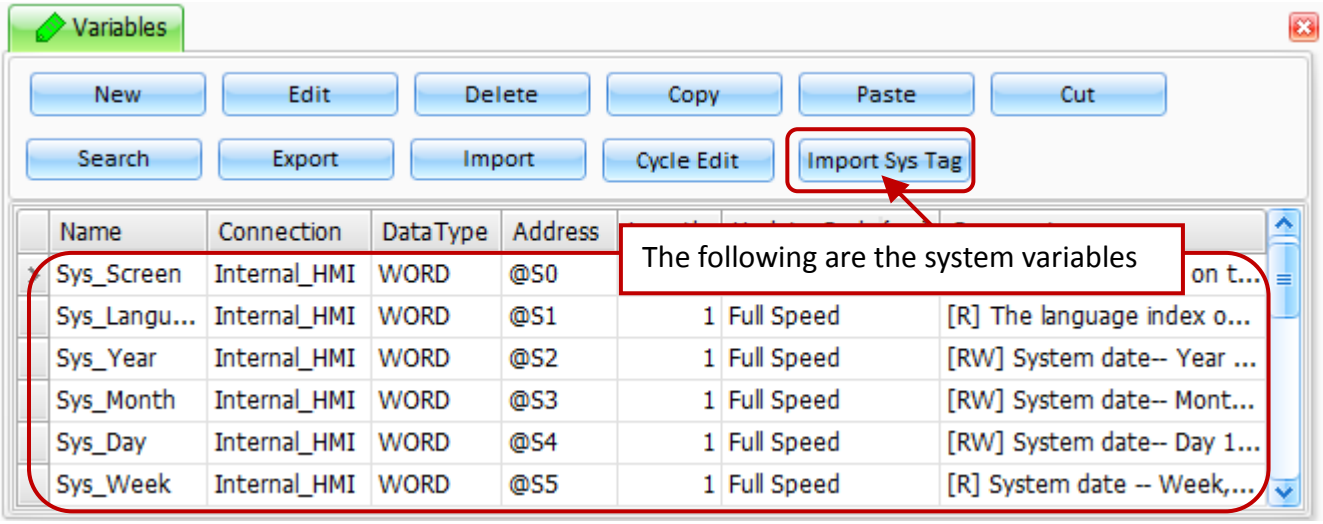
Timeout (ms)	Used to specify the timeout duration before a connection failure occurs
Delay Time (ms)	Used to specify the Delay time between commands
Retry Count	Used to specify the number of connection attempts that should be made if the connection fails
Set connection to offline when retry count exceeded	
Used to automatically set the connection to Offline Mode if the number of connection attempts exceeds configured retry count value	

Chapter 7 Variables



The **Variables** function is used to create, edit, and manage the variables that are used in a specific project, and will be described in more detail below.

To access the Variable page, double-click the **Variables** function in the **Project View** panel.



The following is an overview of the options available on the **Variables** page.

New	Used to create a new variable
Edit	Used to edit a selected variable item
Delete	Used to delete the selected variable item(s)
Copy	Used to copy the selected variable item(s) to the system clipboard
Paste	Used to paste the variable item(s) from the system clipboard
Cut	Used to cut the selected variable item(s) to the system clipboard
Search	Used to search the entire project for references to the variable. Select the name of a variable to be used as a keyword.
Export	Used to export all variable(s) to a CSV file
Import	Used to import all variable(s) from a CSV file
Cycle Edit	Used to specify a fixed period of time that defines how often the variable reads data from a device
Import Sys Tag	Used to import commonly used built-in system variables

7.1 Description of a Variable

A variable is one of the most basic units in a Control System. Depending on functional requirements, variables can be used to access any temporary values related to the connected devices, as well as for internal event simulation and logic operations. The data acquired via the variable can then be used to achieve different purposes using HMI objects, such as alarms, data sampling, recipes, and so on.

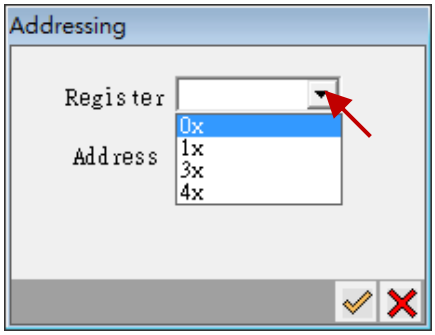
7.1.1 The Usage of Variables

- **Data Readability:**
  All variables can be named according to the devices that are used so that it's much easier to identify the purpose of the variable and to maintain the Creator project.

- **Easy Template Editing:**
  If a device that is being used needs to be replaced for any reason, there is no need to amend all the functions that use the variable. Simply modify the address used by the variable.

- **Prioritized Adjustment:**
  The update speed for variables can be adjusted to achieve the most efficient communication.

7.1.2 Variable Category

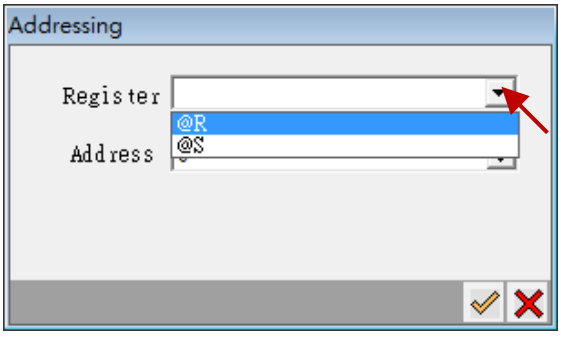


- **External Variables:**
  External variables are used to read/write the external data. The address used in the variable must be set to the Modbus address of the connected device.

- **Internal Variables:**
  Internal variables are used for the internal use of the HMI device. The address used in the variable is defined by the user based on one of two registers:

  @S Register: Used to temporarily store the value(s) for the system variable
  @R Register: Used to temporarily store the value(s) for the user-defined variable

- **Array Variables:**
  Array Variables are used for variables where the data length is greater than "1" (e.g., using **ten** WORDs).



7.1.3 Variable Types

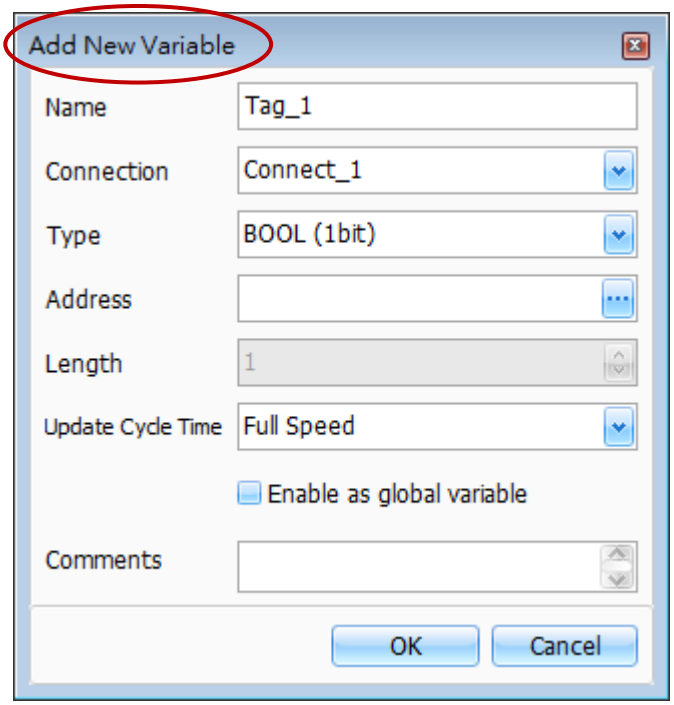
Each variable used in Creator has a specific type, which determines the size and layout of the memory used for the variable, the range of values that can be stored within that memory, and the set of operations that can be applied to the variable.

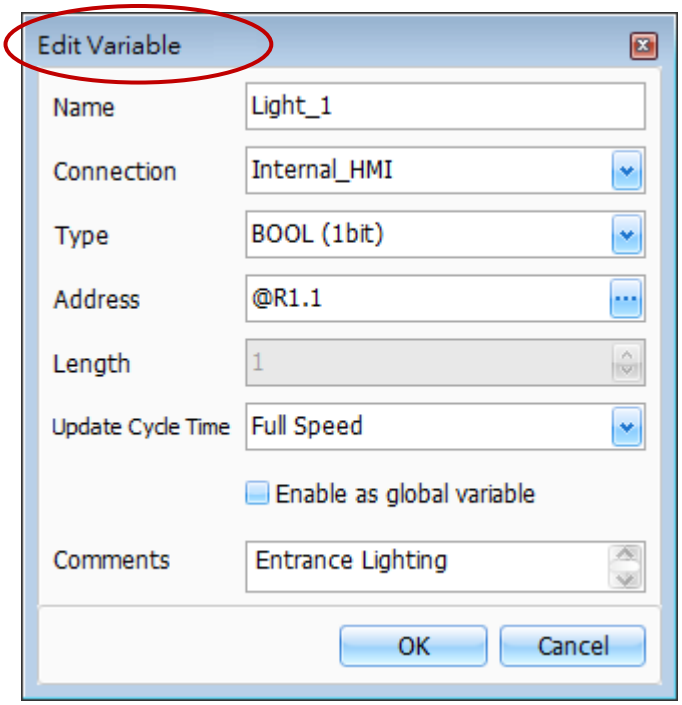
The following is an overview of the types of variable that can be used in Creator.

Type	Number of Bits	Valid Value Range
BOOL	1	0, 1
BYTE	8	0 to 255
CHAR	8	-128 to 127
WORD	16	0 to 65535
INT	16	-32768 to 32767
DWORD	32	0 to 4294967295
LONG	32	-2147483648 to 2147483647
FLOAT	32	-3.4e38 to +3.4e38 (IEEE Floating-Point)

7.2 Creating and Editing Variables

To create a new variable, click the **New** button on the **Variable** page to open the **Add New Variable** dialog box as noted in Chapter 7, which allows the parameters to be configured for items such as the connection type, the variable type, and the address, etc., each of which are described in more detail below.





7.2.1 Add New Variable

To create a new variable, enter a unique name for the Tag in the **Name** text field of the **Add New Variable** dialog box, as the figure above. Select a connection method from the **Connection** drop-down menu, and then select a type for the Tag from the **Type** drop-down menu. In the **Address** text field, click the **Addressing** (...) button to select a specific address or simply enter the address of the tag. Note that the address style will be different based on the **Connection** settings. Enter a data **Length** for the variable, and then select an time option from the **Update Cycle Time** drop-down menu. If desired, check the **Enable as global variable** item and enter any comments in the **Comments** text box. Click the **OK** button to save the changes.

Edit Variable

To edit an existing variable, amend any details in the **Edit Variable** dialog box as necessary, and then click the **OK** button to save the changes.

The following is a description of each of the parameters contained in the Add New variable or the Edit variable dialog box:

- **Name**

  The **Name** parameter is used to specify a user-defined name for the variable.

  Note: It is recommended that a different name be defined for each variable as Creator uses Macro names that are based on the variable name.

- **Connection**

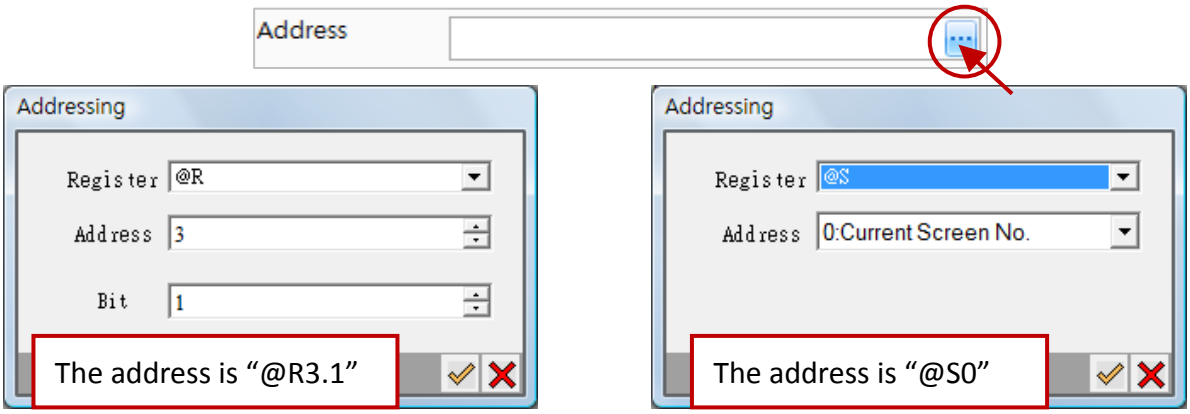
  The **Connection** parameter is used to specify the connection to be accessed by the variable. Select either an existing connection which has previously been configured in the Connections tab (see Chapter 6 Connections), or selects the **Internal_HMI** option to access an internal variable.

- **Type**

  The **Type** parameter is used to specify the data type for the variable and can be selected as any one of BOOL, BYTE, CHAR, WORD, INT, DWORD, LONG or FLOAT types. Refer to Section 7.1.3 Variable Type for more details.

- **Address**

  The **Address** parameter is used to specify the reference address when accessing the device data, and can be either the Communication address or the Internal address, each of which are described below.



A. Communication address

The **Communication address** is used to specify the address for accessing data from an external device. The format of the address style will be different depending on the communication protocol selected (e.g., “b1” or “0x2”).

B. Internal address

The **Internal address** is used to specify the address of the internal variable.

@S (System parameters)

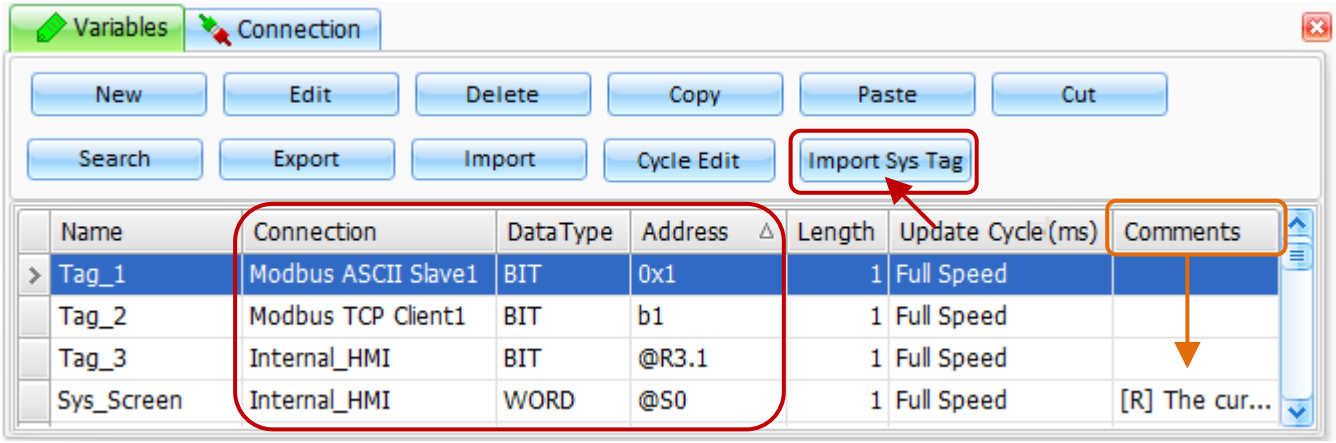
There are a number of built-in system variables that can be imported from the **Variables** page (see Chapter 7 **Variables**), each of which is shown in the table below. For more details regarding each system variable, check the **Comment** field, as illustrated in the diagram below.

The following is an overview of the system variables available for import to the Creator project.

@S0:	Sys_Screen	@S6:	Sys_Hour	@S14:	Sys_LoginUser
@S1:	Sys_Language	@S7:	Sys_Minute	@S15:	Sys_LoginPower
@S2:	Sys_Year	@S8:	Sys_Second	@S20:	Sys_RecipeTableID
@S3:	Sys_Month	@S11:	Sys_BacklightTimer	@S21:	Sys_RecipeRecordIndex
@S4:	Sys_Day	@S12:	Sys_AutoLogout	@S23:	Sys_CpuLoading
@S5:	Sys_Week	@S13:	Sys_Buzzer		

@R (User-defined parameters)

The @R parameter is used to define the starting address of a variable (unit: Byte). If the data type is BOOL (Bit), the address will be displayed with a decimal number (i.e., Bit0 to Bit7), for example, @R5.7 denotes that the variable Address is set to “5”, the Bit is set to “7”.



- **Length**

  The **Length** parameter is used to define how much data exists within a variable. If the length is set to greater than 1, it will be defined as an Array Variable. For example, assume that the starting address is set to **@R100,** the data type is set to **"WORD" (2 Bytes)**, and the length is set to **3**, it denotes that the access range will be **@R100 to @R105**.

Array Index	Configured Address
0	**@R100, @R101** (the WORD data type uses **2** Bytes)
1	**@R102, @R103** (the WORD data type uses **2** Bytes)
2	**@R104, @R105** (the WORD data type uses **2** Bytes)

Note: Array variable may not be supported by certain PLC devices, or if the data type is set to"BOOL" as the length is fixed as "1".

- **Update Cycle Time**

  The **Update Cycle** Time parameter is used to specify the cycle time in milliseconds (ms) for updating the variable values. When set to Full Speed, the update will be performed without any delay. To add additional time options, click the **Cycle Edit** button, see Section 7.3 Cycle Edit for more details.

- **Enable as global variable**

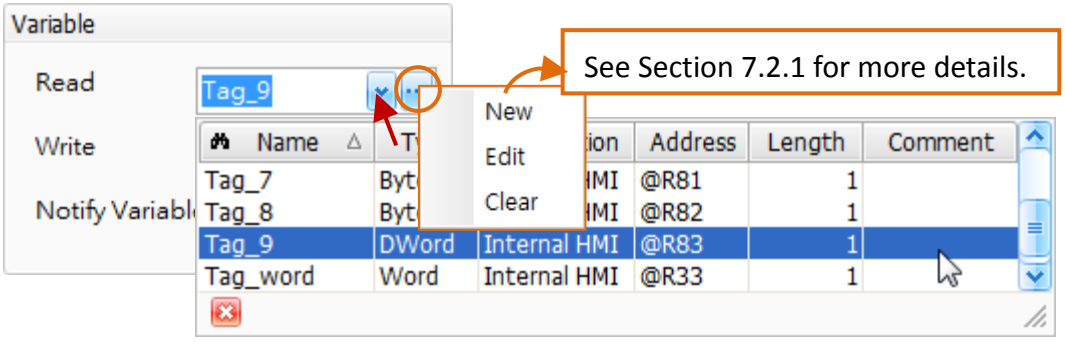
  The **Enable as global variable** checkbox is used to enable the variable as a global variable so that any object functions which reference this variable will operate correctly whatever page the HMI screen is switched to.

- **Comments**

  The **Comments** parameter is used to enter an appropriate description for the variable.

7.2.2 Using Variables

When configuring the object in the **Property View** panel, most of them must be assigned a Read/Write variable. As illustrated in the diagram below, you can click the button (▾), and choose a variable which has been defined in the **Variables** page previously. Alternatively, you can click the button (…) to add or edit or clear the variable. For more details about the objects, see Section 8.2 General Objects.



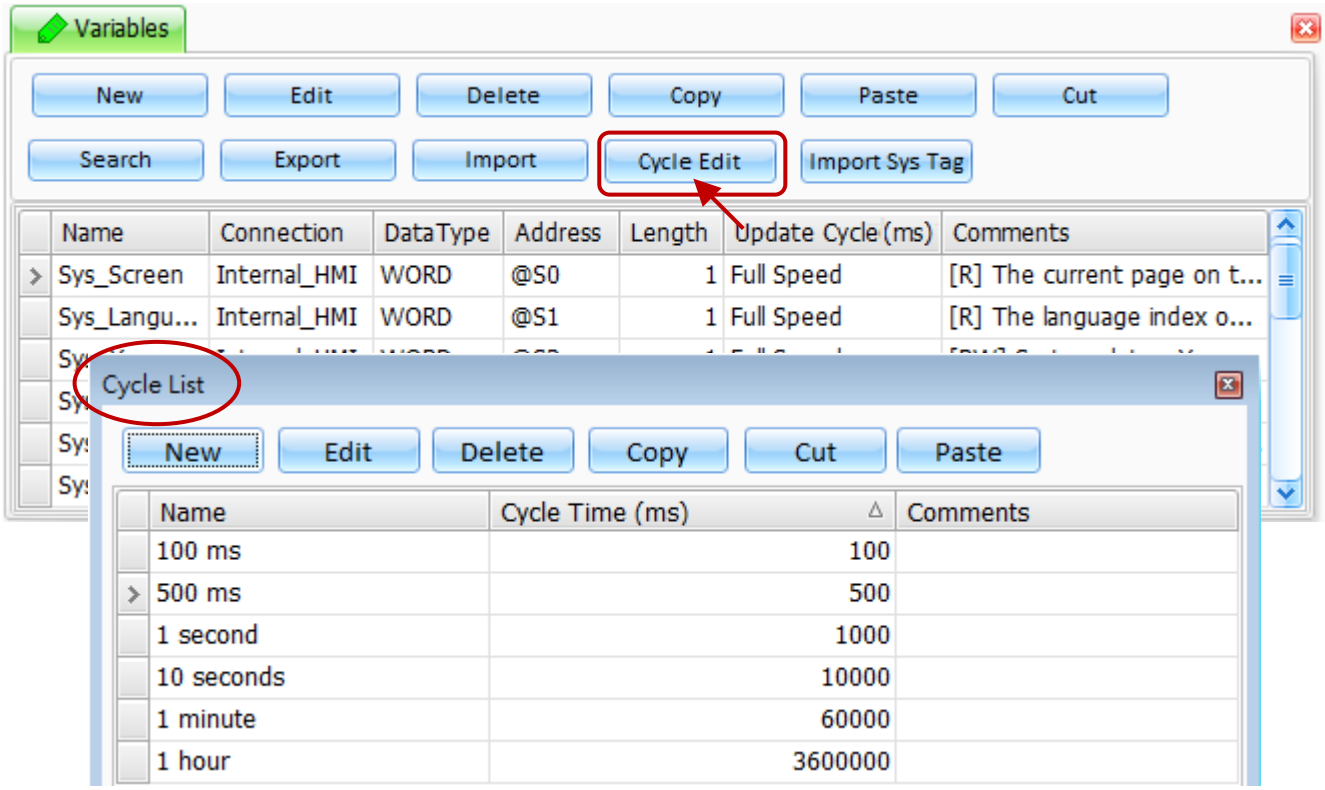
7.3 Cycle Edit

The **Update Cycle Time** parameter defines how often the variable reads data from a device and is a fixed period of time. It is recommended that the cycle time be configured to a value that is suitable for the conditions.

For example, when measuring a temperature that changes slowly, it would be more beneficial to set the cycle time to a value greater than 1 second. However, when monitoring a motor where the measurement changes rapidly and it is critical that the current value is known quickly, it would be advantageous to set the cycle time to a lower value such as 100 ms.

Creator provides a range of time options by default, such as 100 ms, 500 ms, and 1 second, etc., but custom update cycle times can also be configured using the **Cycle List** dialog box.

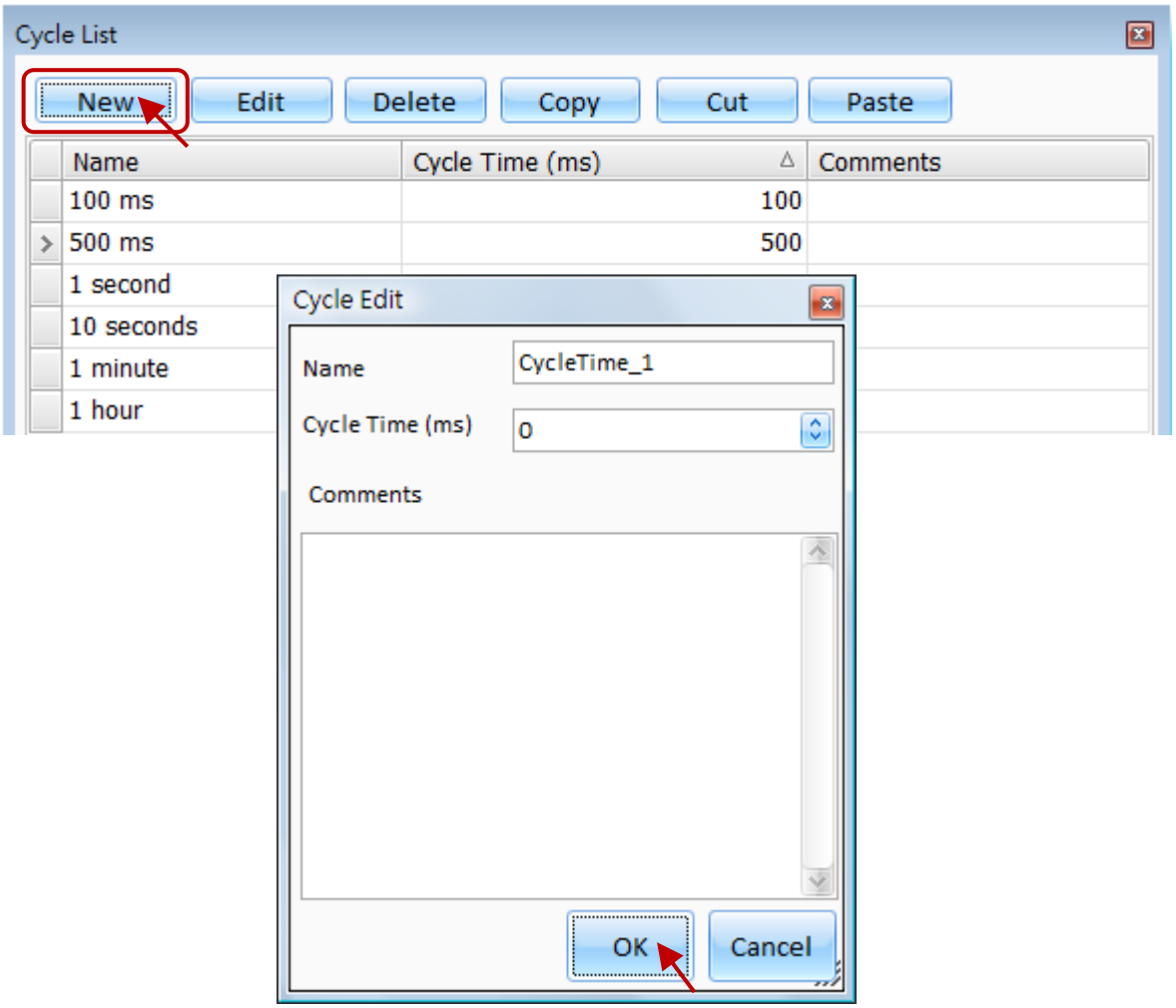
To access the **Cycle List** dialog box, click the **Cycle Edit** button on the **Variables** page.



The following is an overview of the options available in the **Cycle List** dialog box.

New	Used to create a new Cycle Time	**Copy**	Used to copy the selected Cycle Time (s)
Edit	Used to edit the selected Cycle Time	**Cut**	Used to cut the selected Cycle Time (s)
Delete	Used to delete the selected Cycle Time (s)	**Paste**	Used to paste the copied or cut Cycle Time (s)

To add a new Cycle Time item, click the **New** button in the **Cycle List** dialog box to display the **Cycle Edit** dialog box.



The following is an overview of the options available in the **Cycle Edit** dialog box.

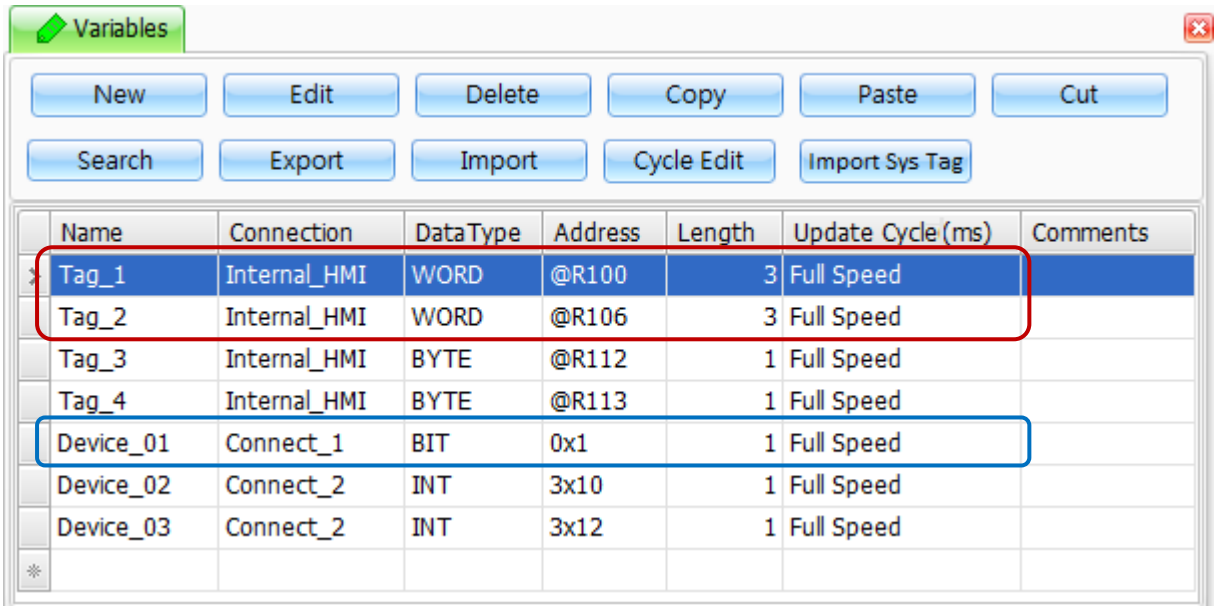
Name	Used to specify a name for the new Cycle List item
Cycle Time	Used to specify the cycle time in milliseconds (ms). The valid range is from0 to 3,600,000 ms
Comments	Used to enter a description for the Cycle List item

Enter a name for the new Cycle Time in the **Name** text field, and then enter a Cycle time in milliseconds in the **Cycle Time** field. If desired, enter any comments in the **Comments** text box, and then click the **OK** button to save the changes.

7.4 Array Variables

The Array variable can be used to access multiple data at one time. Therefore, if the data length of the variable is greater than 1, the variable will be referred to as an Array Variable. For example, assume that the starting address is set to **@R100,** the data type is set to **"WORD" (2 Bytes)**, and the length is set to **3**, it denotes that the access range will be **@R100 to @R105**.

Array Index	Configured Address
0	**@R100, @R101** (the WORD data type uses **2** Bytes)
1	**@R102, @R103** (the WORD data type uses **2** Bytes)
2	**@R104, @R105** (the WORD data type uses **2** Bytes)



7.5 Suggestions for Using Variables

The speed that the data is updated on the HMI screen is dependent on the traffic on the communication port. If there is heavy traffic, the update speed will be slower. If a large number of variables are used to display data on the HMI screen, the loading on the communication port will become greater. The Creator provides an optimization function, but is only applicable when the addresses of the variables are continuous.

For example, suppose a screen exists that uses the three addresses @R0, @R100, and @R200. As the addresses are not continuous, the Creator will generate three communication records. However, if the addresses are set to @R0, @R2 and @R4, the sequence becomes continuous, meaning that the system will only generate a single communication record. Moreover, if you use the external variable (e.g., "Device_01" as the figure above), the address of the variable is depending on the connected device.

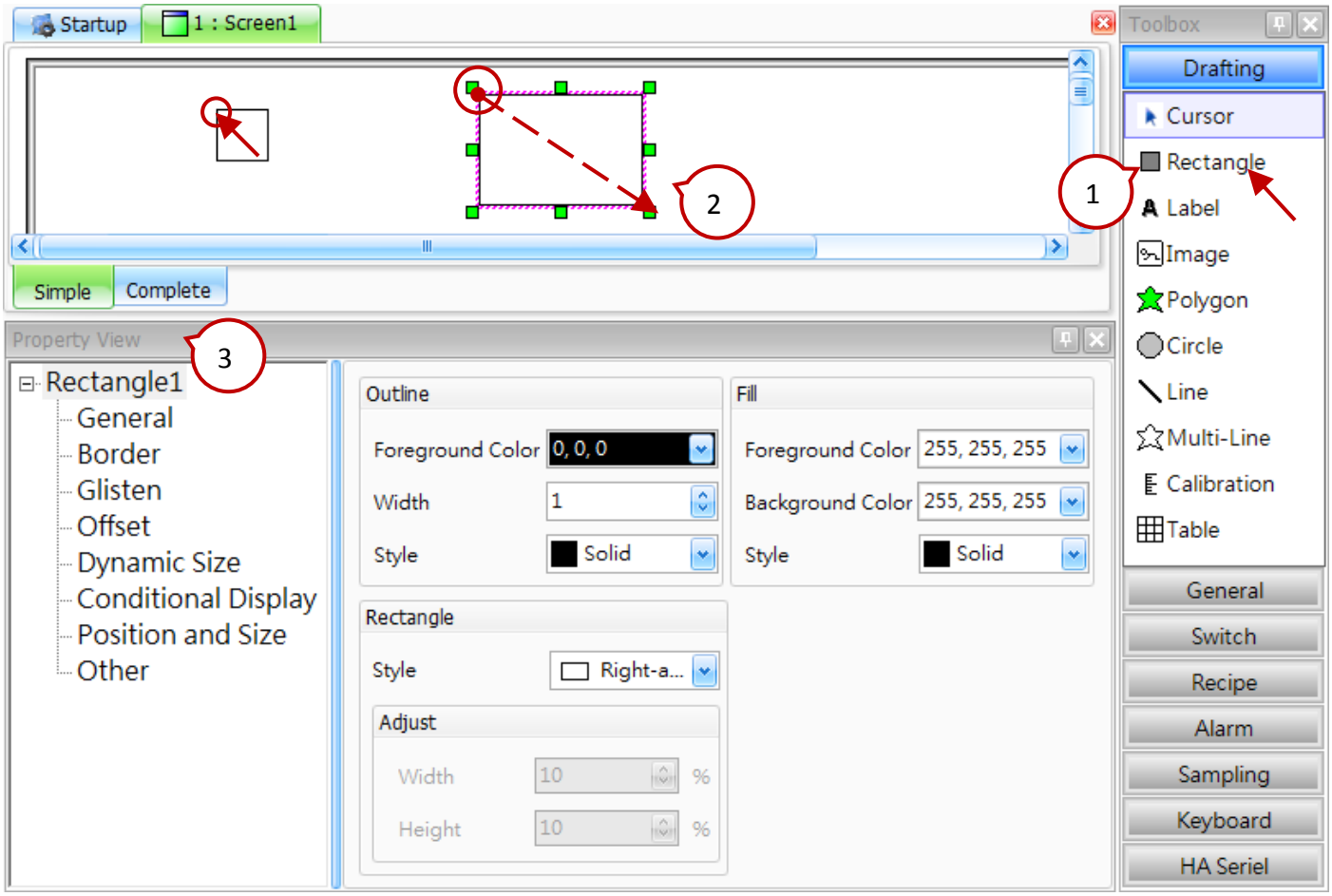
Chapter 8 Basic Object Usage

The **Toolbox** contains four categories of basic objects, including Drafting, General, Switch, and Keyboard. These objects can be used to transfer data to the graphical user interface on the HMI screen via either an external and/or an internal variable. Objects can also be used to perform control various processes. The functions for the advanced objects found in the Toolbox, which include Recipe, Alarm and Data Sampling objects, need to be configured before the object can be used. For more details related to the usage of a specific object type, refer to the descriptions provided in the sections below.

To create an object, first click the name of the desired object in the relevant category of the Toolbox. Position the mouse cursor on the Screen Design Area, and then click and hold the left mouse button to drag the object until it is the desired size and shape, as illustrated in the diagram below, and then release the mouse button.

Alternatively, click the name of the object in the relevant category of the Toolbox to select it, and then click the desired position on the Screen Design Area to place it with the default size.

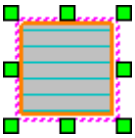
To edit the attributes of the object at a later date, click the object to open the Property View panel for that object.



8.1 Drafting Objects

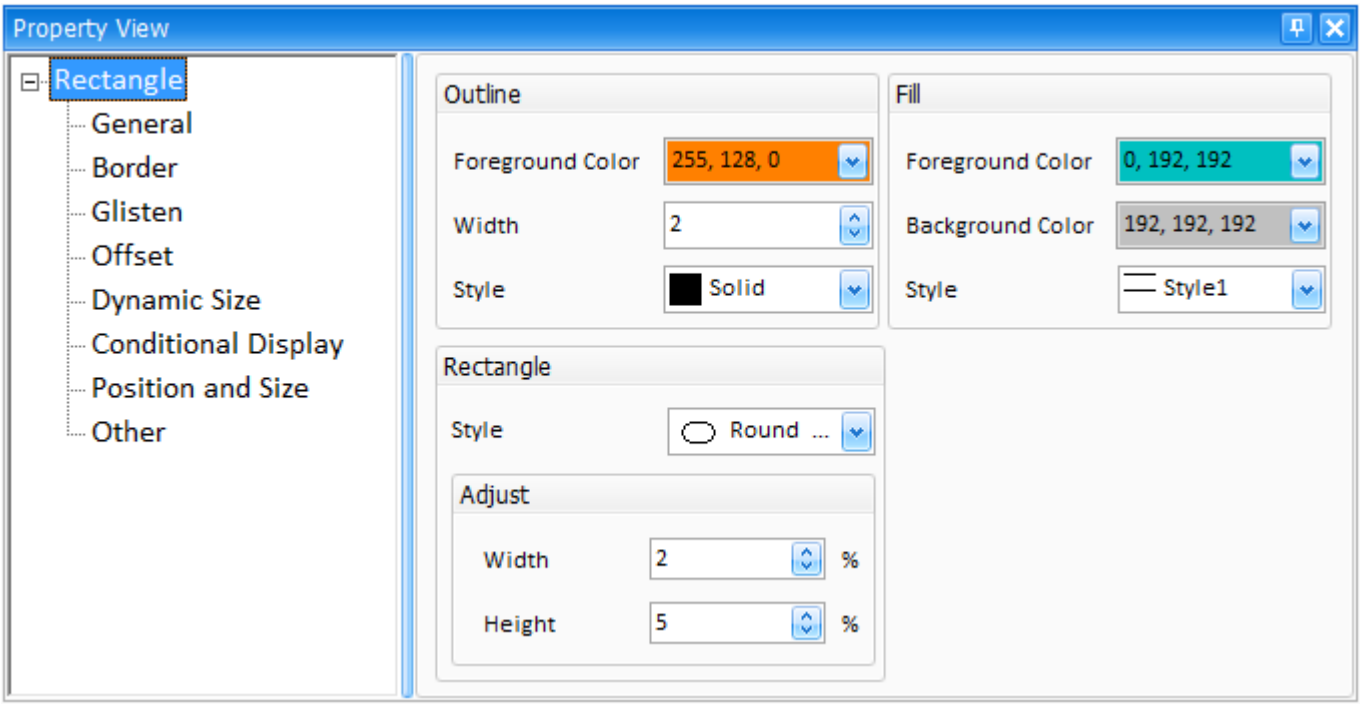
The **Drafting** category of the Toolbox includes a variety of objects that can be used to create and edit general geometric figures, calibration functions, tables, and other graphic type HMI objects, each of which are described in more detail in the following sections.

8.1.1 Rectangle



The **Rectangle** object can be used to create a rectangular shape on the screen. The attributes for the **Rectangle** object, including the size, style, and the color, can be adjusted as desired. See Chapter 8 Basic Object Usage for details of how to create an object.

After creating a **Rectangle** object, the **Property View** panel for the object will be automatically displayed allowing the various attributes of the object to be configured. The following is a description of how to configure the properties for the **Rectangle** object.



- General Properties

  The **General Properties** dialog box is used to configure the display style for the **Rectangle** object, including the outline color and width, the fill color and the style of the rectangle.

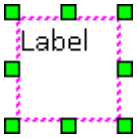
The following is an overview of the options available in the **General Properties** section of the Property View for the **Rectangle** object.

Outline	**Foreground Color**	Used to specify the color to be used for the outline of the **Rectangle** object

	Width	Used to specify the width to be used for the outline of the **Rectangle** object in pixels
	Style	Used to specify the style to be used for the outline of the **Rectangle** object, which can be either None or Solid
Fill	**Foreground Color**	Used to specify the foreground color of the **Rectangle** object
	Background Color	Used to specify the background color of the **Rectangle** object
	Style	Used to specify the pattern style of the Fill for the **Rectangle** object, which can be None (i.e., transparent), Solid, or Style1 to Style 52
Rectangle	**Style**	Used to specify the style of the **Rectangle** object, which can be Right-angled, Round Rectangle, or Slice Rectangle
	Adjust	Used to adjust the width and the height of the Rounded or Slice type **Rectangle** object

- See Section 8.5 Common Attributes for details about the others attributes of the **Label** object. Border (1), Glisten, Offset, Dynamic Size, Conditional Display, Position and Size, and Other.

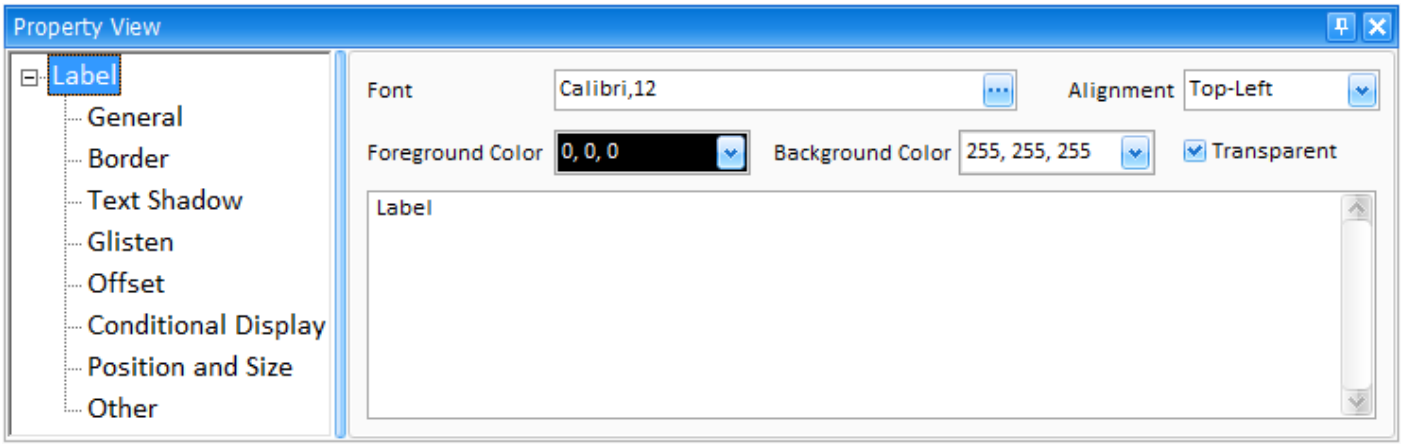
8.1.2 Label



The **Label** object can be used to add text to the screen. The attributes for the **Label** object, including the size and style of the font, as well as the alignment, can be adjusted as desired.

The text can be displayed in different languages based on the language of the current Operating System, see Section 11.3 Language for more details. See Chapter 8 Basic Object Usage for details of how to create an object.

After creating a **Label** object, the **Property View** panel for the object will be automatically displayed allowing the various attributes of the object to be configured. The following is a description of how to configure the properties for the **Label** object.



- Generally Properties

  The **General Properties** dialog box is used to configure the display style for the **Label** object, including the font for the text, the foreground and background color, and the alignment, etc.

The following is an overview of the options available in the **General Properties** section of the Property View for the **Label** object.

Font	Used to specify the font style and size for the text that will appear on the **Label** object
Foreground Color	Used to specify the color of the text to be used for the **Label** object
Background Color	Used to specify the background color of the **Label** object
Alignment	Used to specify the position of the text within the border of the **Label** object
Transparent	Used to specify whether or not the background color will be shown as transparent when displayed on the screen
Label	Used to specify the text that will be displayed on the **Label** object

- See Section 8.5 Common Attributes for details about the others attributes of the **Label** object. Border (1), Text Shadow, Glisten, Offset, Conditional Display, Position and Size, and Other.

8.1.3 Image (Picture)

The **Image** object can be used to add an image to the screen. The attributes for the **Image** object, including the size and alignment of the image, can be adjusted as desired.
See Chapter 8 Basic Object Usage for details of how to create an object.

After creating a **Image** object, the **Property View** panel for the object will be automatically displayed allowing the various attributes of the object to be configured. The following is a description of how to configure the properties for the **Image** object.

- General Properties

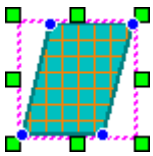
  The **General Properties** dialog box is used to load or clear an image that is displayed on the screen for the **Image** object, and to configure the size, alignment, and transparency, etc.

The following is an overview of the options available in the **General Properties** section of the Property View for the **Image** object.

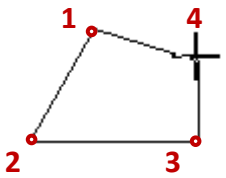
Load	Used to load an image from the **Image Manager**. See Section 11.4 **Image Manager** for more details.
Clear	Used to clear the currently loaded image
Auto Size	Used to specify whether or not the image should be automatically stretched to fit the size of the **Image** object
Using Transparent	Used to specify whether or not the selected color will be shown as transparent when displayed on the screen
Alignment	Used to specify the position of the image within the border of the **Image** object

- See Section 8.5 Common Attributes for details about the others attributes of the **Image** object. Border (1), Glisten, Offset, Dynamic Size, Conditional Display, Position and Size, and Other.

8.1.4 Polygon

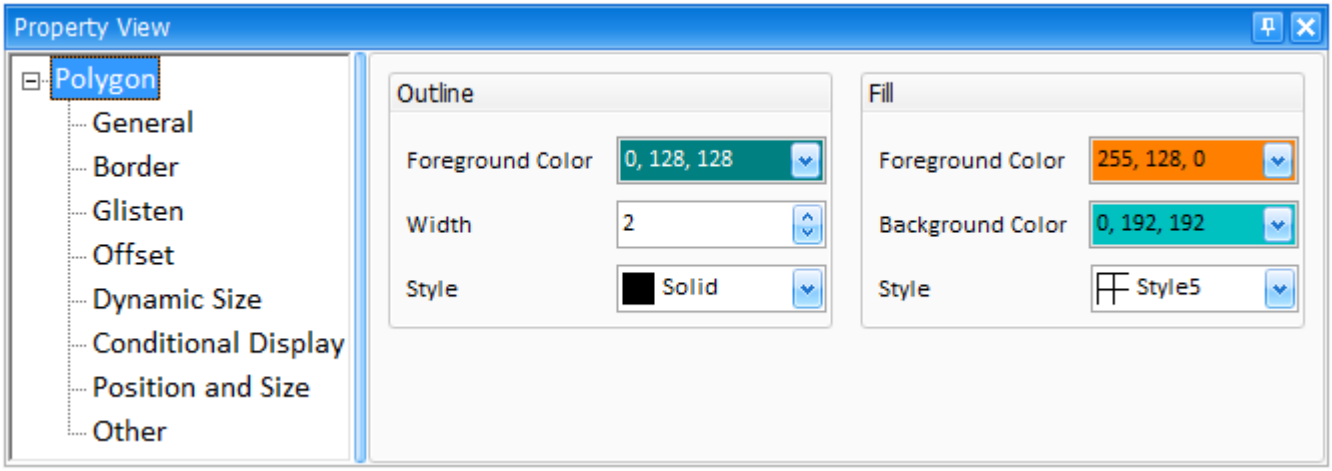


The **Polygon** object can be used to create a polygon shape on the screen. The attributes for the polygon, including the color, fill, and style, can be adjusted as desired.
See Chapter 8 Basic Object Usage for details of how to create an object.



The process for creating a **Polygon** object is slightly different to that of other objects because you need to click the desired position on the Screen Design Area several times to create multiple paths. The polygon consists of four nodes, meaning that four paths will be created. After creating the position of the initial node, continue clicking to create the nodes for the other corners, and double-clicking the ending nodes to complete the **Polygon** object. The polygon can be modified later by adjusting the position of an individual node.

After creating a **Polygon** object, the **Property View** panel for the object will be automatically displayed allowing the various attributes of the object to be configured. The following is a description of how to configure the properties for the **Polygon** object.



- General Properties

  The **General Properties** dialog box is used to configure the display style for the **Polygon** object, including the outline color and width, and the fill style of the polygon.

The following is an overview of the options available in the **General Properties** section of the Property View for the **Polygon** object.

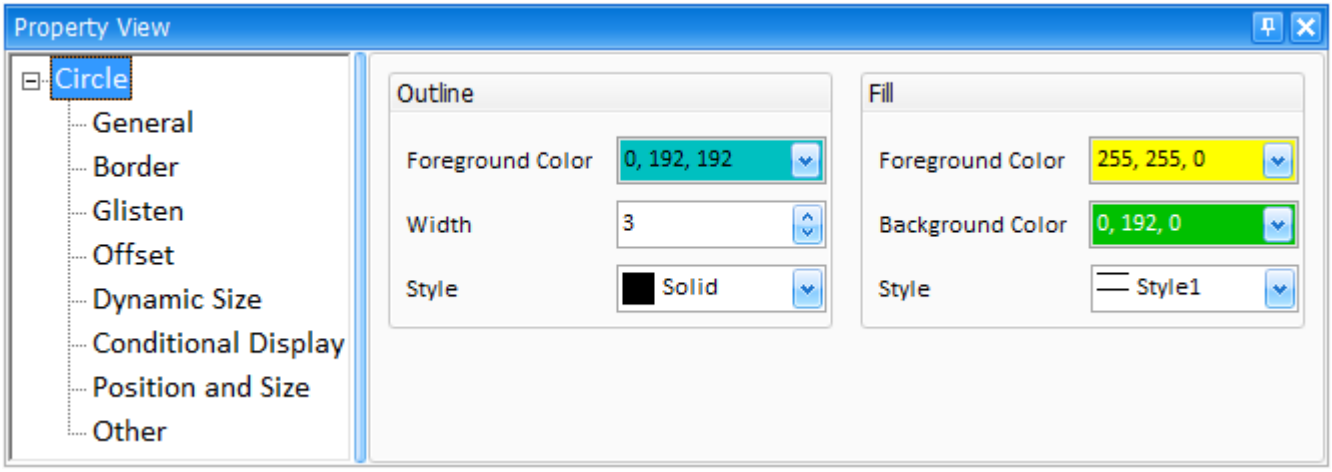
Outline	**Foreground Color**	Used to specify the color to be used for the outline of the **Polygon** object
	Width	Used to specify the width for the outline of the **Polygon** object in pixels
	Style	Used to specify the style for the outline of the **Polygon** object, which can be either None or Solid
Fill	**Foreground Color**	Used to specify the foreground color of the **Polygon** object
	Background Color	Used to specify the background color of the **Polygon** object
	Style	Used to specify the pattern style for the Fill for the **Polygon** object, which can be None (i.e., transparent), Solid, or Style1 to Style 52

- See Section 8.5 Common Attributes for details about the others attributes of the **Polygon** object. Border (1), Glisten, Offset, Dynamic Size, Conditional Display, Position and Size, and Other.

8.1.5 Circle

The **Circle** object can be used to create a circular shape on the screen. The attributes for the circle, including the color, width, and style, can be adjusted as desired.
See Chapter 8 Basic Object Usage for details of how to create an object.

After creating a **Circle** object, the **Property View** panel for the object will be automatically displayed allowing the various attributes of the object to be configured. The following is a description of how to configure the properties for the **Circle** object.



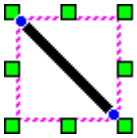
- General Properties
  The **General Properties** dialog box is used to configure the display style for the **Circle** object, including the outline color and width, and the fill style for the circle.

The following is an overview of the options available in the **General Properties** section of the Property View for the **Circle** object.

Outline	**Foreground Color**	Used to specify the color to be used for the outline of the **Circle** object
	Width	Used to specify the width for the outline of the **Circle** object in pixels
	Style	Used to specify the style for the outline of the **Circle** object, which can be either None or Solid
Fill	**Foreground Color**	Used to specify the foreground color of the **Circle**
	Background Color	Used to specify the background color of the **Circle** object
	Style	Used to specify the pattern style for the Fill for the **Circle** object, which can be None (i.e., transparent), Solid, or Style1 to Style 52

- See Section 8.5 Common Attributes for details about the others attributes of the **Circle** object.
  Border (1), Glisten, Offset, Dynamic Size, Conditional Display, Position and Size, and Other.

8.1.6 Line



The **Line** object can be used to create a single line on the screen. The attributes for the line, including the color, width, and style, can be adjusted as desired.
See Chapter 8 Basic Object Usage for details of how to create an object.

After creating a **Line** object, the **Property View** panel for the object will be automatically displayed allowing the various attributes of the object to be configured. The following is a description of how to configure the properties for the **Line** object.

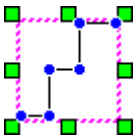
- General Properties
  The **General Properties** dialog box is used to configure the display style for the **Line** object, including the outline color and width, and the style of the line.

The following is an overview of the options available in the **General Properties** section of the Property View for the **Line** object.

Outline	**Foreground Color**	Used to specify the color to be used for the **Line** object
	Width	Used to specify the width of the **Line** object in pixels
	Style	Used to specify the style to be used for the **Line** object, which can be either None or Solid

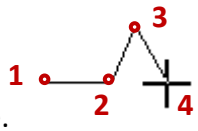
- See Section 8.5 Common Attributes for details about the others attributes of the **Line** object.
  Border (1), Glisten, Offset, Dynamic Size, Conditional Display, Position and Size, and Other.

8.1.7 Multi-Line



The **Multi-Line** object can be used to create an abstract shape on the screen. The attributes for the Multi-Line object, including the color, width, and style, can be adjusted as desired.
See Chapter 8 Basic Object Usage for details of how to create an object.

The process for creating a **Multi-Line** object is slightly different to that of other objects, because you need to click the desired position on the Screen Design Area several times to create multiple paths. After creating the position of the initial node, continue clicking to create the other nodes and the paths between them, and double-clicking the ending nodes to complete the **Multi-Line** object. The **Multi-Line** object can be modified later by adjusting the position of an individual node.



After creating a **Multi-Line** object, the **Property View** panel for the object will be automatically displayed allowing the various attributes of the object to be configured. The following is a description of how to configure the properties for the **Multi-Line** object.

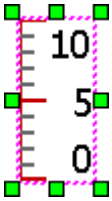
- General Properties
  The **General Properties** dialog box is used to configure the display style for the **Multi-Line** object, including the outline color and width, and the style of the multi-line shape.

The following is an overview of the options available in the **General Properties** section of the Property View for the **Multi-Line** object.

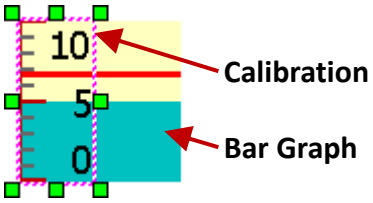
Outline	**Foreground Color**	Used to specify the color to be used for the **Multi-Line** object
	Width	Used to specify the width of the **Multi-Line** object in pixels
	Style	Used to specify the style to be used for the **Multi-Line** object, which can be either None or Solid

- See Section 8.5 Common Attributes for details about the others attributes of the **Multi-Line** object. Border (1), Glisten, Offset, Dynamic Size, Conditional Display, Position and Size, and Other.

8.1.8 Calibration

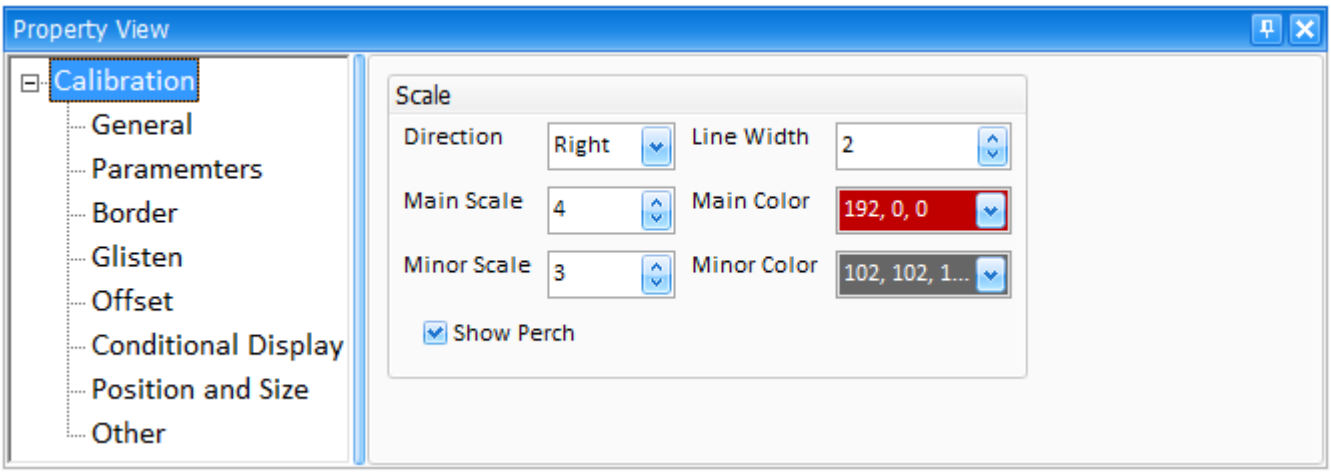


The **Calibration** object is used in conjunction with the **Bar Graph** object to create a visual representation of calibration data. The attributes for the **Calibration** object, including the color and style of the Scale and the Values used on the Bar Graph can be adjusted as desired.



See Chapter 8 Basic Object Usage for details of how to create an object. For more details related to the Bar Graph object, see Section 8.2.8 Bar Graph.

After creating a **Calibration** object, the **Property View** panel for the object will be automatically displayed allowing the various attributes of the object to be configured. The following is a description of how to configure the properties for the **Calibration** object.



- General Properties

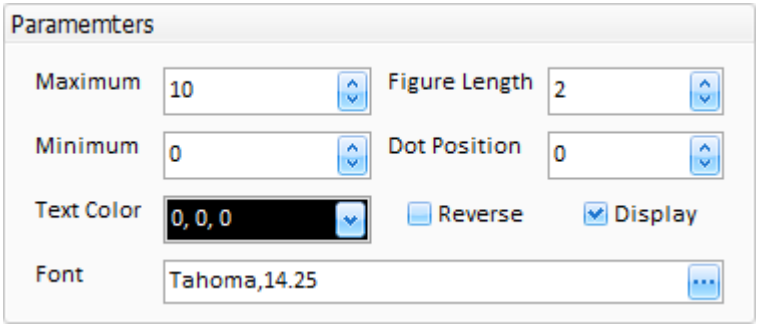
  The **General Properties** dialog box is used to configure the display style for the **Calibration** object, including the direction, and the color, width and intervals for the scales of the calibration.

The following is an overview of the options available in the **General Properties** section of the Property View for the **Calibration** object.

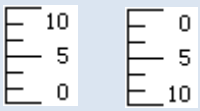
Direction	Used to specify the direction of the scale, and can be set to Right, Up, Left, and Down
Main Scale	Used to specify the interval used for the main scale. The valid range is 2 to 100
Minor Scale	Used to specify the interval used for the minor scale. The valid range is 0 to 100
Line Width	Used to specify the width of the lines used for the scale in pixels
Main Color	Used to specify the color used for the main scale
Minor Color	Used to specify the color used for the minor scale
Show Perch	Used to enable whether the vertical axis is displayed for the scale

- Parameters Properties

  The **Parameters Properties** dialog box is used to configure the attributes for the scale values of the **Calibration** object, including the maximum and minimum values, the maximum number of digits, and the color and font used for the text.

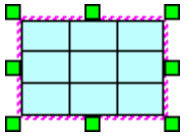


The following is an overview of the options available in the **Parameters** section of the Property View for the **Calibration** object.

Maximum	Used to specify the maximum value for the calibration scale
Minimum	Used to specify the minimum value for the calibration scale
Figure Length	Used to specify the maximum number of digits that can be displayed for the values on the calibration scale. For example, if Maximum is set to 123, and the Figure Length is set to 2, the scale value will be displayed as 23. In this case, if the Display Length is set to 3 or above, the scale value will still be displayed as 123.
Dot Position	Used to specify the position of the decimal point (for example, if Maximum is set to 10, the Figure Length is set to 2, and the Dot Position is set to 1, the scale value will be displayed as 1.0)
Text Color	Used to specify the color used for the value displayed on the calibration scale
Reverse	Used to specify whether or not the direction of the value displayed on the calibration scale should be reversed. For example, when reversed, the sequence 0 – 5 – 10 will be displayed as 10 – 5 – 0 
Display	Used to specify whether or not the value is to be displayed
Font	Used to specify font style and size used for the text on the calibration scale

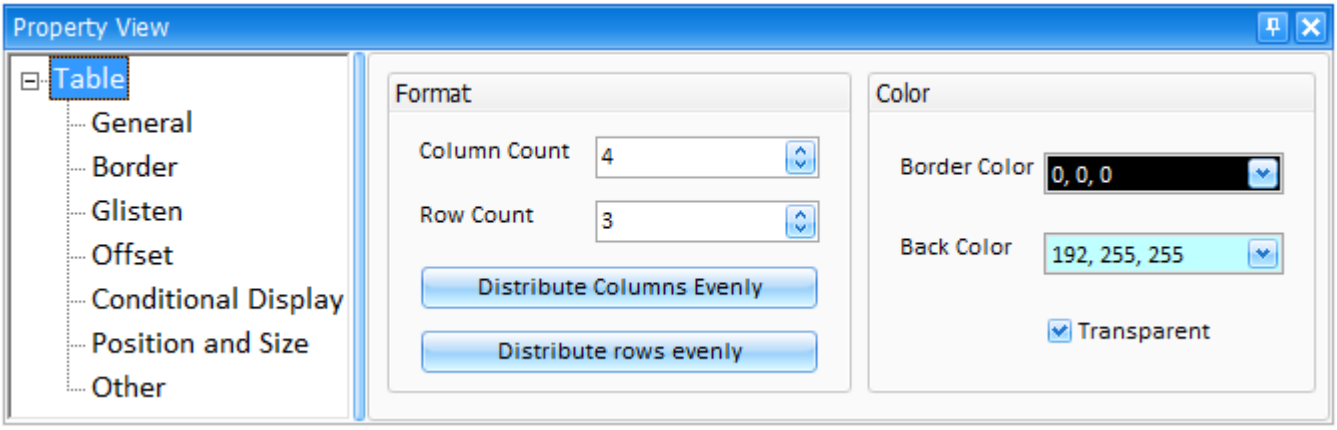
- See Section 8.5 Common Attributes for details about the others attributes of the **Calibration** object. Border (1), Glisten, Offset, Conditional Display, Position and Size, and Other.

8.1.9 Table



The **Table** object can be used in conjunction with all kinds of objects to display data in a table format. The attributes for the **Table** object, including the number of columns and rows to be included in the table, as well as the color and width of the border, can be adjusted as desired.

See Chapter 8 Basic Object Usage for details of how to create an object. After creating a **Table** object, the **Property View** panel for the object will be automatically displayed allowing the various attributes of the object to be configured. The following is a description of how to configure the properties for the **Table** object.



- General Properties

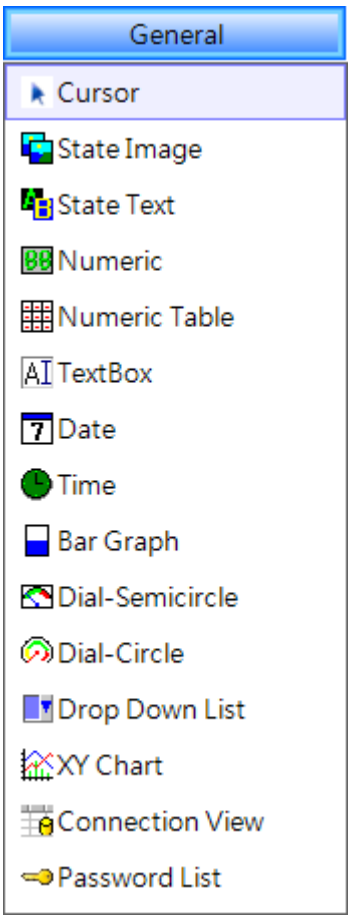
  The **General Properties** dialog box is used to configure the display style for the **Table** object, including the number of columns and rows, and the colors to be used for the table.

The following is an overview of the options available in the **General Properties** section of the Property View for the **Table** object.

Column Count	Used to specify the number of columns in the table
Row Count	Used to specify the number of rows in the table
Distribute Columns Evenly	Used to evenly distribute the width of the columns
Distribute Rows Evenly	Used to evenly distribute the height of the rows
Border Color	Used to specify the color to be used for the border of the table
Back Color	Used to specify the background color to be used for the table
Transparent	Used to specify whether or not the background color will be set to transparent

- See Section 8.5 Common Attributes for details about the others attributes of the **Table** object. Border (1), Glisten, Offset, Conditional Display, Position and Size, and Other.

8.2 General Objects

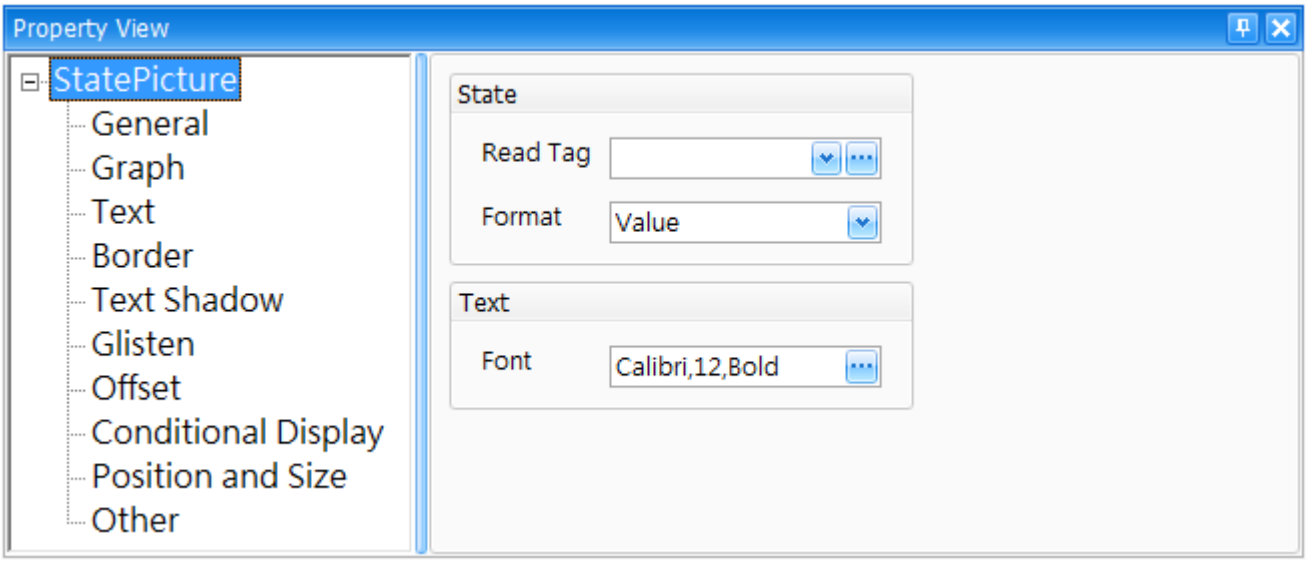


The **General** category of the Toolbox includes a variety of HMI objects that can be used to create and edit state images and text, text and numeric displays, date and time functions, and bar graphs and charts, etc., each of which are described in more detail in the following sections.

8.2.1 State Image

The **State Image** object can be used to graphically display a variable value using either an image or a text. The attributes for the State Image object, including the size and alignment of the image or text, can be adjusted as desired. See Chapter 8 Basic Object Usage for details of how to create an object.

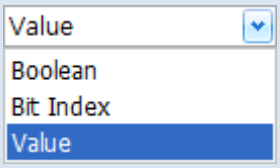
After creating a **State Image** object, the **Property View** panel for the object will be automatically displayed allowing the various attributes of the object to be configured. The following is a description of how to configure the properties for the **State Image** object.



- General Properties

  The **General Properties** dialog box is used to configure the state variable to be used for the **State Image** object, as well as the format, together with the font and size for the text to be displayed. For more details related to the usage of variables, see Chapter 7 Variables.

The following is an overview of the options available in the **General Properties** section of the Property View for the **State Image** object.

<table>
<tr><td rowspan="2">State</td><td>Read Tag</td><td>Used to specify which variable is used to read data
For more details related to the usage of variables, see <u>Chapter 7 Variables</u>.</td></tr>
<tr><td>Format</td><td>Used to specify the format that will be used to display the state value, as described below.
 
<u>Boolean:</u>
If the variable value is equal to 0, the state value will be 0.
If the variable value is not equal to 0, the state value will be 1.

For example, if the value of a variable which data type is WORD (16-bit) is 32, the picture or text will be displayed depends on the setting of state value 1. (See Graph and Text Properties for more details)
State Value Range: 0, 1

<u>Bit Index:</u>
If the variable value is equal to 2^n, the state value will be n.
If the variable value is not equal to 2^n, no picture/text will be displayed.

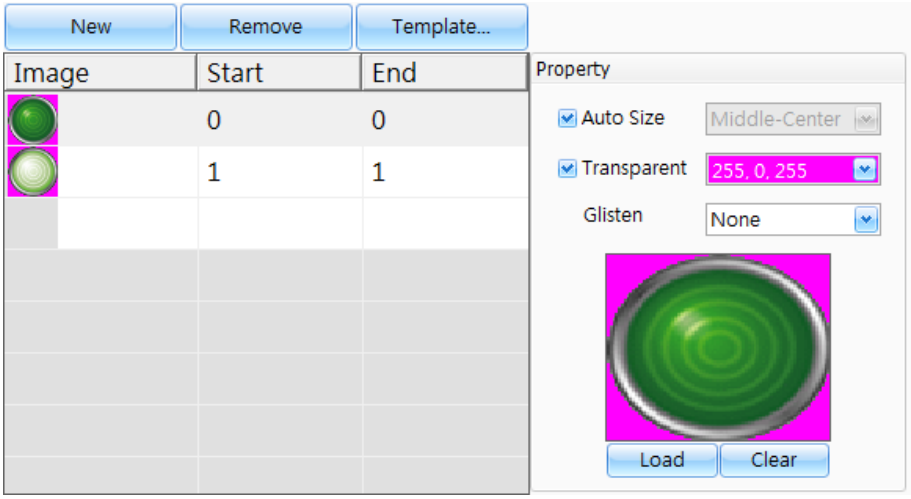
For example, if the value of a variable which data type is BYTE (8-bit) is 32 (i.e., 2^5) the picture or text will be displayed depends on the setting of state value 5.
MSB LSB
<table><tr><td>2^7</td><td>2^6</td><td>2^5</td><td>2^4</td><td>2^3</td><td>2^2</td><td>2^1</td><td>2^0</td></tr><tr><td>0</td><td>0</td><td>1</td><td></td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table>State Value Range: 0 to 31

<u>Value:</u>
The variable value is the state value.

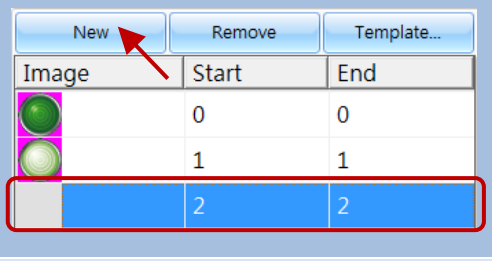
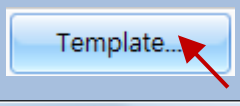
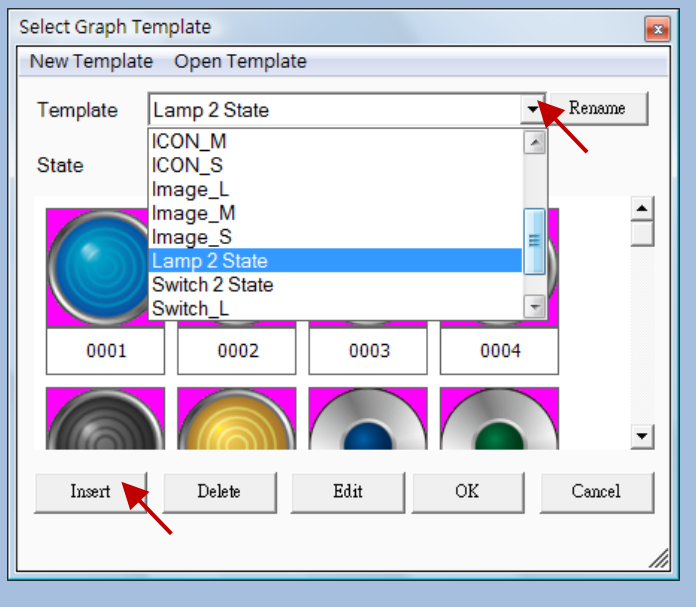
For example, if the value of a variable which data type is DWORD (32-bit) is 32, the picture or text will be displayed depends on the setting of state value 32.
State Value Range: 0 to 2147483647</td></tr>
<tr><td>Text</td><td></td><td>Used to specify the font and size for the text to be displayed</td></tr>
</table>

- Graph Properties

  The **Graph Properties** dialog box is used to configure the attributes for the state graphic used for the **State Image** object.

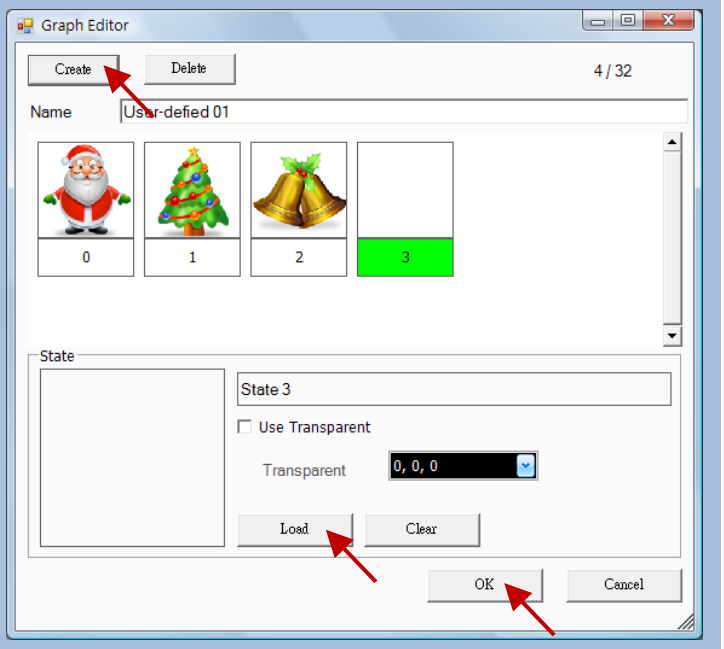


The following is an overview of the options available in the **Graph Properties** section of the Property View for the **State Image** object.

New	Used to add a new image item, and then you can double click the Image, Start, or End field to configure an image or state values 
Remove	Used to remove the selected image item(s)
Template	Used to add/open a template, or add a state image in the template. **<u>To create a user-defined state image</u>**  1. Click the **Template** button to go to the **Select Graph Template** dialog box. 2. Select one of image types from the **Template** drop-down menu. 3. Click the **Insert** button to go to the **Graph Edit** dialog box. You can enter a name for this state image. 

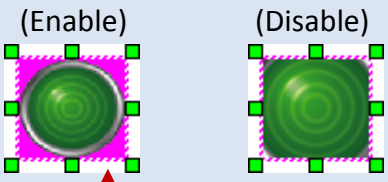
4. Click the **Create** button to add a State item, and then click the **Load** button to load an image.

5. Click the **OK** button to save the settings.



Auto Size

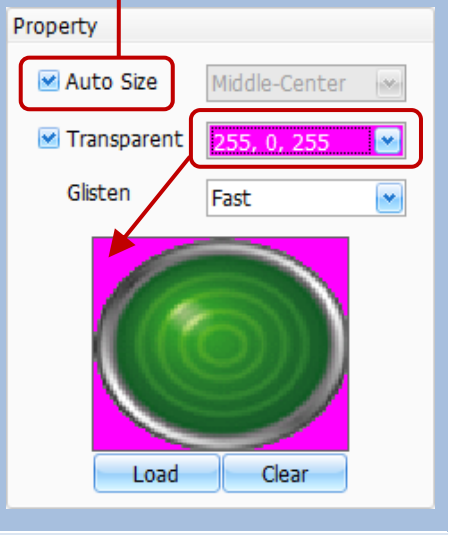
Used to specify whether or not the image should be automatically stretched to fit the size of the object.



Transparent

Used to specify whether or not the background color of image will be shown as transparent when displayed on the screen.

Note that only when the set color and the background color of image are the same, it will be displayed as transparent



Glisten

Used to specify the Glisten speed for images based on state values

Load

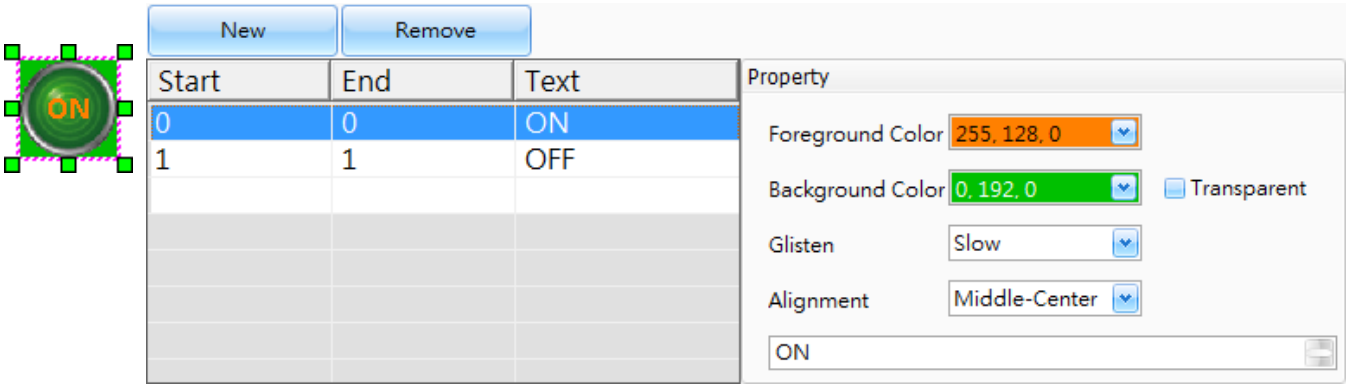
Used to select an image to be used from the **Image Manager**.
See Section 11.4 “Image Manager” for more details.

Clear

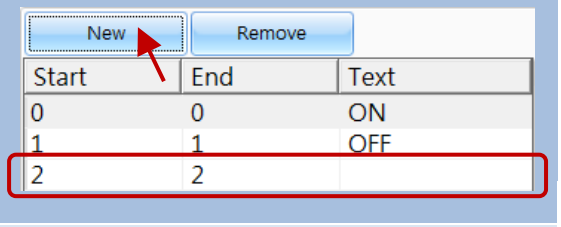
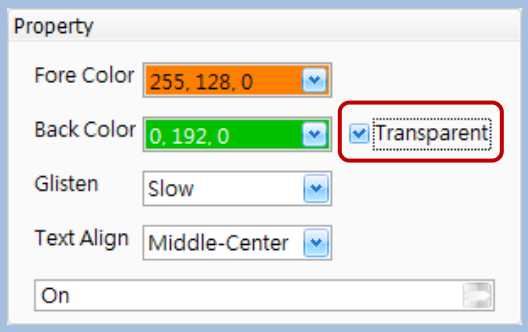
Used to clear the currently loaded image

- Text Properties

  The **Text Properties** dialog box is used to configure the attributes of the text used for the **State Image** object.

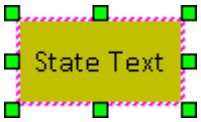


The following is an overview of the options available in the **Text Properties** section of the Property View for the **State Image** object.

New	Used to add a new text item, and then you can double click the Start, End or Text field to configure state values or the text 
Remove	Used to remove the selected text item(s)
Foreground Color	Used to specify the color to be used for the text
Background Color	Used to specify the color to be used for the background of the object
Transparent	Used to specify whether or not the background color will be shown as transparent when displayed on the screen  
Glisten	Used to specify the Glisten speed for text based on state values
Alignment	Used to specify the position of the text within the border of the object
Text	Used to specify the text to be displayed. Click a text item and enter the text.

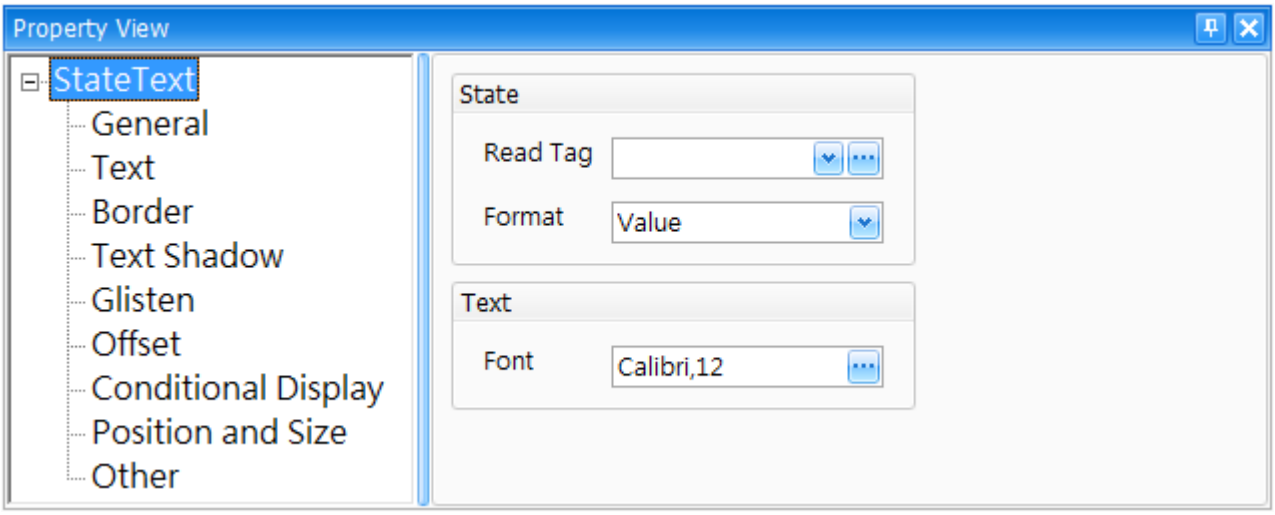
- See Section 8.5 Common Attributes for details about the others attributes of the **State Image** object. Border (1), Glisten, Offset, Conditional Display, Position and Size, and Other.

8.2.2 StateText



The **State Text** object can be used to display a variable value using a text message. The attributes for the State Text object, including the size and style of the font, as well as the alignment, can be adjusted as desired. See Chapter 8 Basic Object Usage for details of how to create an object.

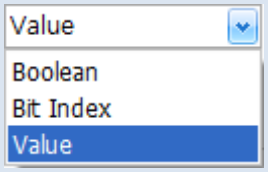
After creating a **State Text** object, the **Property View** panel for the object will be automatically displayed allowing the various attributes of the object to be configured. The following is a description of how to configure the properties for the **State Text** object.



- General Properties

  The **General Properties** dialog box is used to configure the state variable to be used for the **State Text** object, as well as the format, together with the font and size for the text to be displayed. For more details related to the usage of variables, see Chapter 7 Variables.

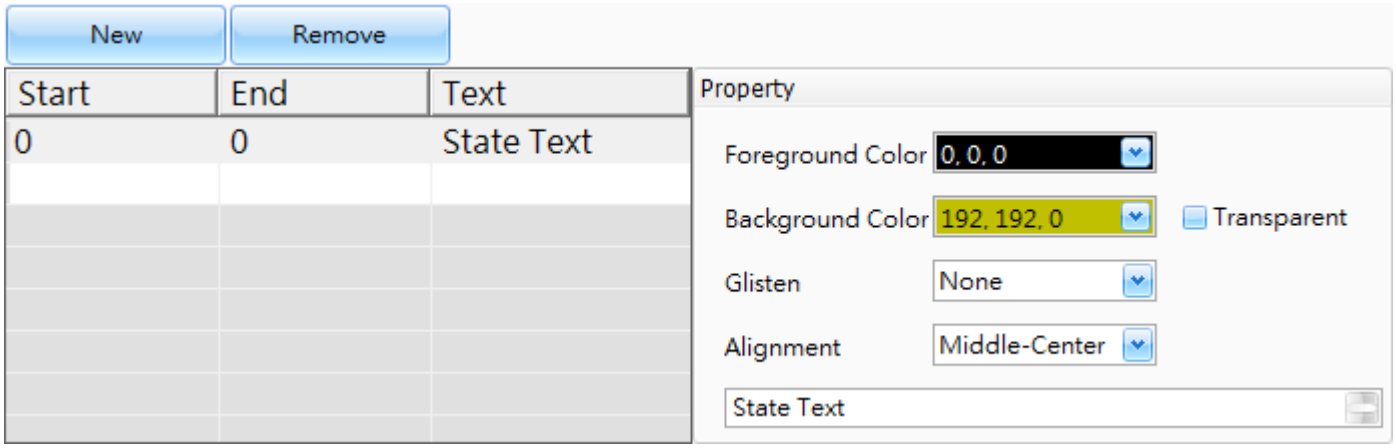
The following is an overview of the options available in the General Properties section of the Property View for the **State Text** object.

State	**Read Tag**	Used to specify which variable is used to read data For more details related to the usage of variables, , see Chapter 7 Variables
	Format	Used to specify the format that will be used to display the state value, as described below.  **Boolean:** If the variable value is equal to 0, the state value will be 0. If the variable value is not equal to 0, the state value will be 1.

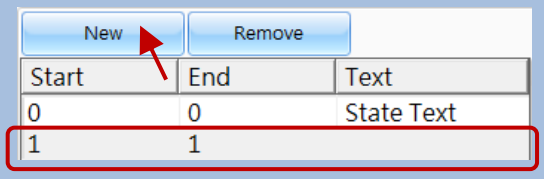
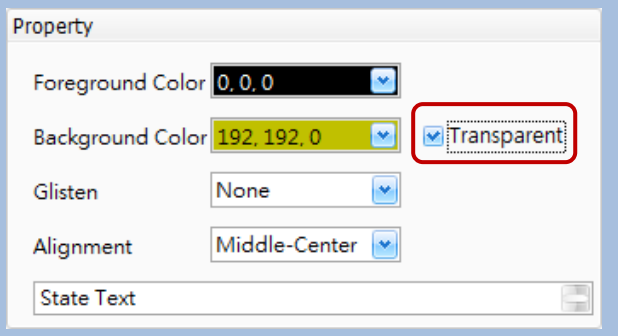
		For example, if the value of a variable which data type is WORD (16-bit) is 32, the text will be displayed depends on the setting of state value 1. (See Text Property for more details) State Value Range: 0, 1
State	**Format**	**Bit Index:** If the variable value is equal to 2^n, the state value will be n. If the variable value is not equal to 2^n, no text will be displayed. For example, if the value of a variable which data type is BYTE (8-bit) is 32 (i.e., 2^5) the text will be displayed depends on the setting of state value 5. MSB LSB 2^7 | 2^6 | **2^5** | 2^4 | 2^3 | 2^2 | 2^1 | 2^0 0 | 0 | **1** | 0 | 0 | 0 | 0 | 0 State Value Range: 0 to 31
		Value: The variable value is the state value. For example, if the value of a variable which data type is DWORD (32-bit) is 32, the text will be displayed depends on the setting of state value 32. State Value Range: 0 to 2147483647
Text	**Font**	Used to specify the font and size for the text to be displayed

- Text Properties

  The **Text Properties** dialog box is used to configure the attributes of the text used for the **State Text** object.

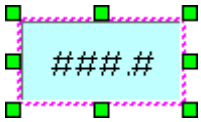


The following is an overview of the options available in the **Text Properties** section of the Property View for the **State Text** object.

New	Used to add a new text item, and then you can double click the Start, End or Text field to configure state values or the text 
Remove	Used to remove the selected text item(s)
Foreground Color	Used to specify the color to be used for the text
Background Color	Used to specify the color to be used for the background of the object
Transparent	Used to specify whether or not the background color will be shown as transparent when displayed on the screen  
Glisten	Used to specify the Glisten speed for text based on state values
Alignment	Used to specify the position of the text within the border of the object
Text	Used to specify the text to be displayed. Click a text item and enter the text.

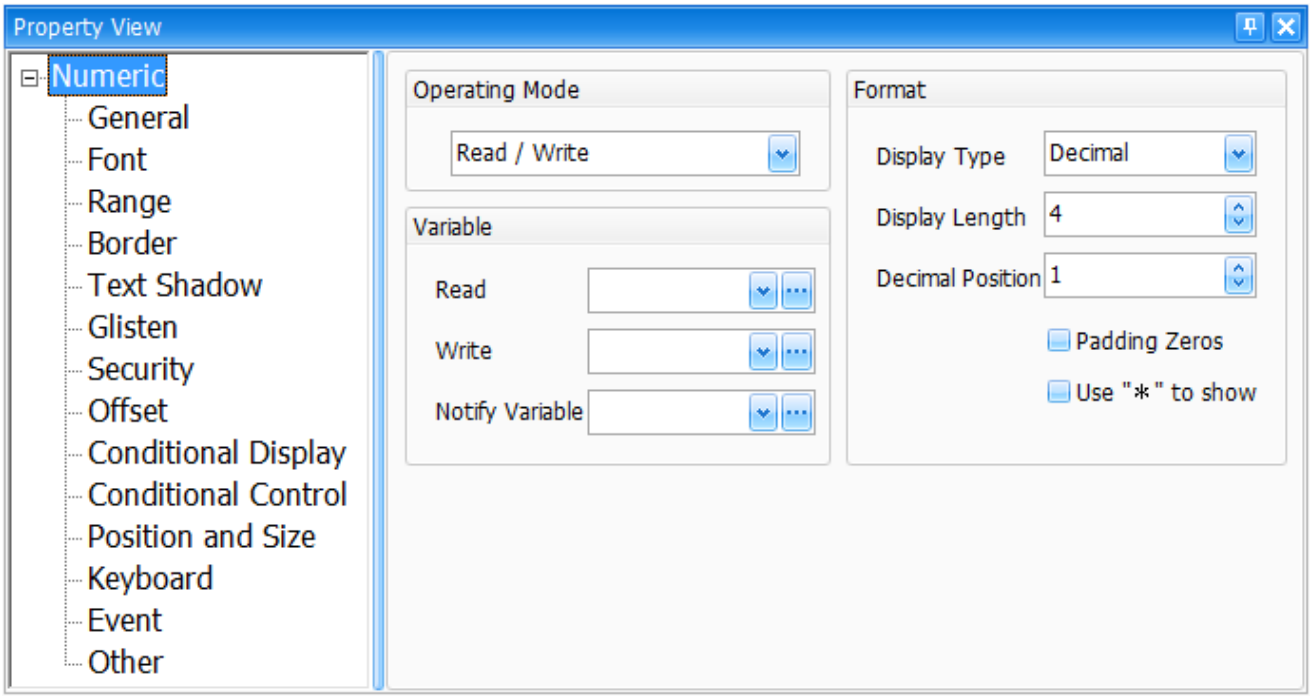
- See Section 8.5 Common Attributes for details about the others attributes of the **State Text** object. Border (1), , Text Shadow, Glisten, Offset, Conditional Display, Position and Size, and Other.

8.2.3 Numeric



The **Numeric** object can be used to read/write the variable value, or convert the read value depends on the display Type. The attributes for the Numeric object, including the size and style of the font, as well as the alignment, can be adjusted as desired.
See Chapter 8 Basic Object Usage for details of how to create an object.

After creating a **Numeric** object, the **Property View** panel for the object will be automatically displayed allowing the various attributes of the object to be configured. The following is a description of how to configure the properties for the **Numeric** object.



- General Properties

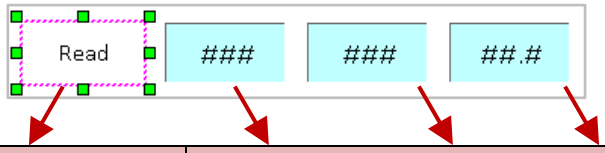
  The **General Properties** dialog box is used to configure the operating mode, the variable attributes, and the display format to be used for the **Numeric** object. For more details related to the usage of variables, see Chapter 7 Variables.

The following is an overview of the options available in the **General Properties** section of the Property View for the **Numeric** object.

Operating Mode		Used to configure the operating mode of a variable, and can be set to “Read” or “Read / Write”. Section
Variable	**Read**	Used to specify the variable to be read
	Write	Used to specify the variable to be write
	Notify Variable	Used to specify the variable which used to display the state of data write. The state value of the variable will be set to 1 when a data write occurs.

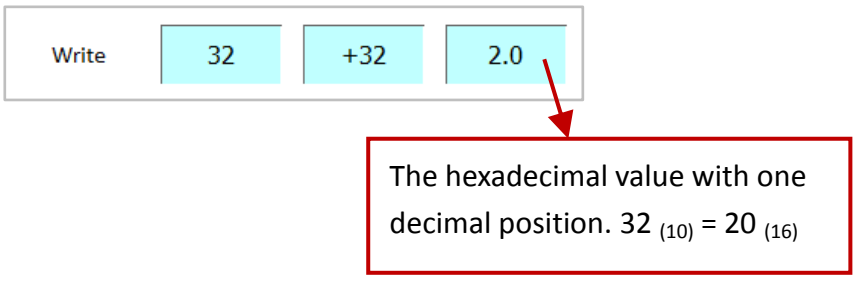
Format	**Display Type**	Used to specify the format to be used for the variable value. Five options are available: Decimal, Signed Decimal (i.e., the Read/Write data can be a positive or negative value), Octal, Hexadecimal, and BCD Code
	Display Length	Used to specify the number of digits that will be displayed
	Decimal Position	Used to specify the position of the decimal point (read from the right). E.g., when using a BYTE variable, the valid value range is 0 to 255. If the Decimal Position = 2, and the Display Type is Decimal, the valid range will become 0.00 to 2.55.
	Padding Zeros	Used to specify whether or not to add leading zeros to the insufficient number of digits. E.g., if the Display Length = 3, the result for a value of "55" will be displayed as "055".
	Use * to Show	Used to specify whether * will be displayed rather than the actual value. This is usually used for passwords or other sensitive information

For example, to add one **State Text** and three **Numeric** objects on the screen.



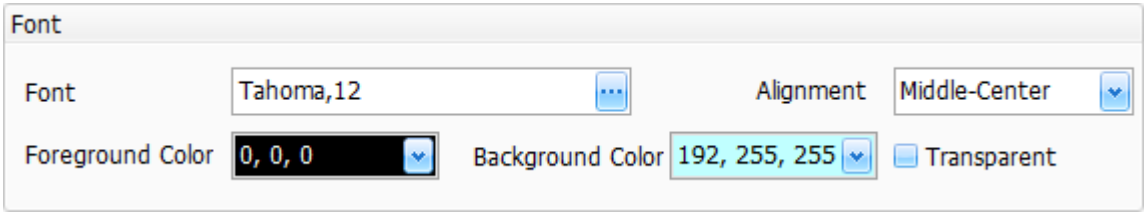
State Text	Numeric (1) / Numeric (2) / Numeric (3)
General Properties: Read Variable: **Tag1** (Data Type: BOOL) **Text Properties:** To display "Read" if the state value is equal 0. To display "**Write**" if the state value is equal **1**.	**General Properties:** Operating Mode: **"Read / Write"** Read / Write Variable: **Tag2** (Data Type: INT) Notify Variable: **Tag1** (Data Type: BOOL) Display Type: **Decimal / Signed Decimal / Hexadecimal** Display Length: **3** Decimal Position: 0 / 0 / **1**

If the **Numeric** (1) is set to **32**, i.e., 20 $_{(Hex.)}$, the **State Text** will be shown as **Write**, the **Numeric** (2) will be shown as **+32**, and the **Numeric** (3) will be shown as 2.0.



- Font Properties

  The **Font Properties** dialog box is used to configure the attributes of the font used for the **Numeric** object.

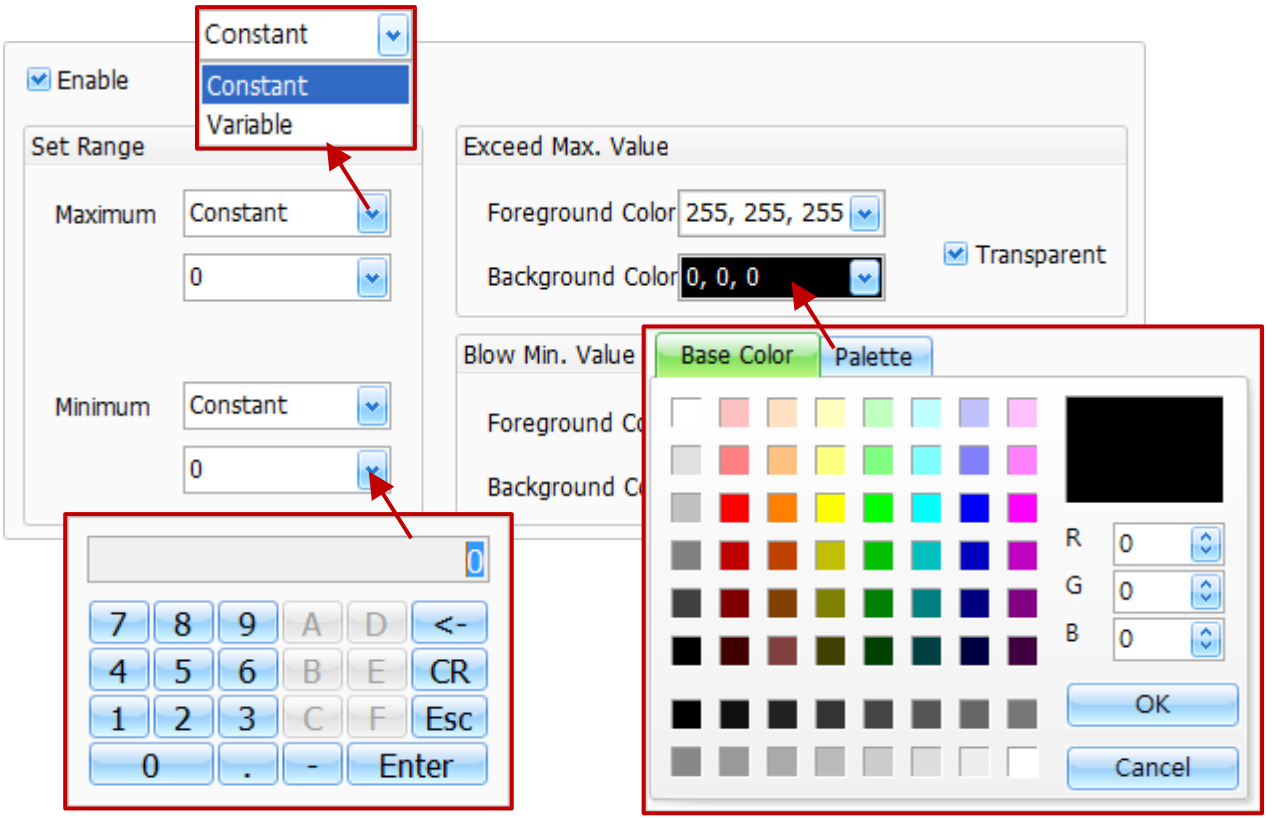


The following is an overview of the options available in the **Font Properties** section of the Property View for the **Numeric** object.

Font	Used to specify the font and size used for the read value
Alignment	Used to specify the position of the read value within the border of the object
Foreground Color	Used to specify the color to be used for the read value
Background Color	Used to specify the color to be used for the background of the object
Transparent	Used to specify whether or not the background color will be shown as transparent when displayed on the screen

- Range Properties

  The **Range Properties** dialog box is used to configure the maximum and minimum values and the alarm color for the **Numeric** object, which will be activated if the range of value is exceeded.

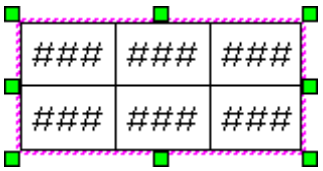


The following is an overview of the options available in the **Range** section of the Property View for the **Numeric** object.

Set Range	**Maximum**	Used to specify the maximum value with a constant or a variable
	Minimum	Used to specify the minimum value with a constant or a variable
Exceed Max. Value	**Foreground Color**	Used to specify the color to be used for the exceeding value
	Background Color	Used to specify the color to be used for the background of the object
	Transparent	Used to specify whether or not the background color will be shown as transparent when displayed on the screen
Below Min. Value	**Foreground Color**	Used to specify the color to be used for the exceeding value
	Background Color	Used to specify the color to be used for the background of the object
	Transparent	Used to specify whether or not the background color will be shown as transparent when displayed on the screen

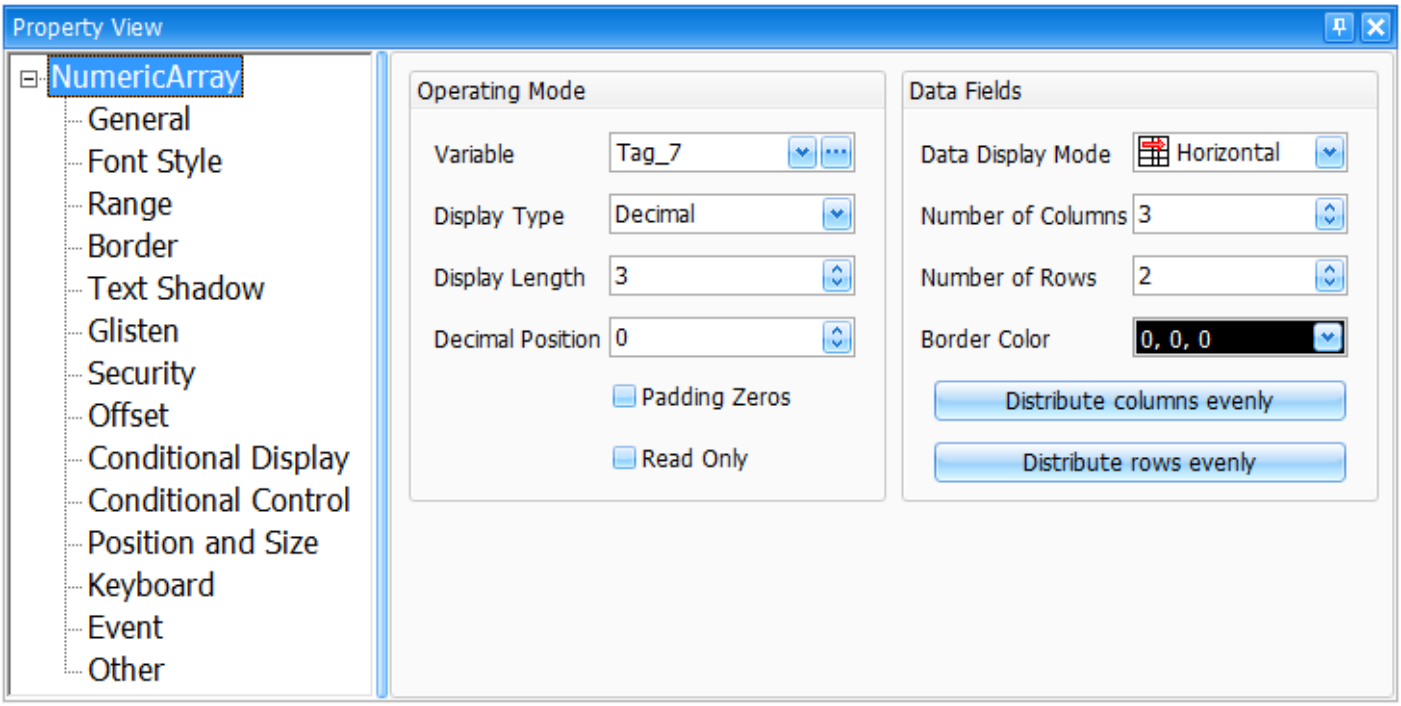
- See Section 8.5 Common Attributes for details about the others attributes of the **Numeric** object. Border (1), Text Shadow, Glisten, Security, Offset, Conditional Display, Conditional Control, Position and Size, Keyboard, Event, and Other.

8.2.4 Numeric Table



The **Numeric Table** object can be used to read/write values of an array variable in a table format. The attributes for the **Numeric Table** object, including the number of columns and rows to be included in the table, as well as the color and width of the border, etc., can be adjusted as desired.

See Chapter 8 Basic Object Usage for details of how to create an object. After creating a **Numeric Table** object, the **Property View** panel for the object will be automatically displayed allowing the various attributes of the object to be configured. The following is a description of how to configure the properties for the **Numeric Table** object.



- General Properties

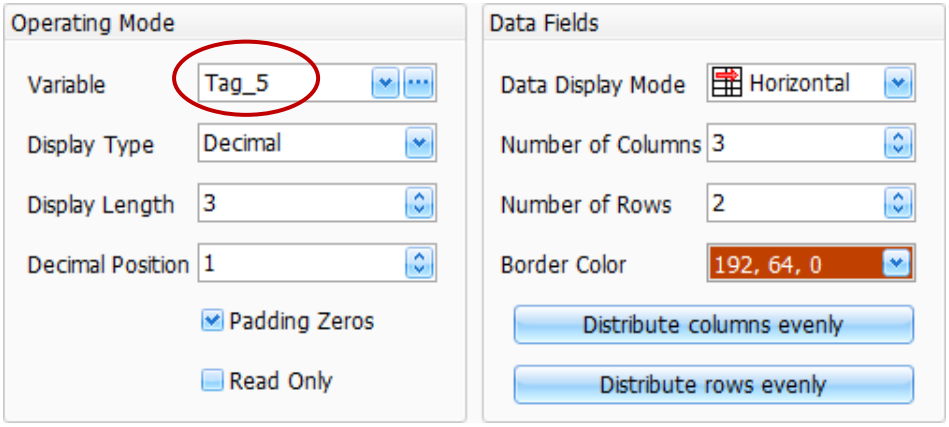
  The **General Properties** dialog box is used to configure the attributes for the variable to be used in the **Numeric Table** object and the display format for the data. For more details related to the usage of variables, see Chapter 7 Variables.

The following is an overview of the options available in the **General Properties** section of the Property View for the **Numeric Table** object.

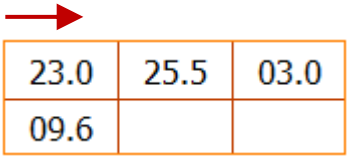
Operating Mode	**Variable**	Used to specify an Array variable Note that if the data length of the variable is greater than “1”, it called Array Variable (see Section 7.4 for more details)

	Display Type	Used to specify the format to be used for the variable value Five options are available: Decimal, Signed Decimal (i.e., the Read/Write data can be a positive or negative value), Octal, Hexadecimal, and BCD Code
	Display Length	Used to specify the number of digits that will be displayed
	Decimal Position	Used to specify the position of the decimal point (read from the right). E.g., when using a BYTE variable, the valid value range is 0 to 255. If the Decimal Position = 2, and the Display Type is Decimal, the valid range will become 0.00 to 2.55.
	Padding Zeros	Used to specify whether or not to add leading zeros to the insufficient number of digits. E.g., if the Display Length = 3, the result for a value of "55" will be displayed as "055".
	Read Only	Used to specify whether or not the variable is only for read data
Column	**Data Display Mode**	Used to specify the direction in which the array data is displayed: **1**: Display the data horizontally from left to right **2**: Display the data vertically from top to bottom
	Number of Columns	Used to specify the number of columns in the table
	Number of Rows	Used to specify the number of rows in the table
	Border Color	Used to specify the color for the border of the table
	Distribute Columns evenly	Used to equally distribute the width of the columns in the table
	Distribute Rows evenly	Used to equally distribute the height of the rows in the table

In this case, we configure the "Tag_5" as an Array variable, the data length is **4**, and the data type is Byte.



Note that you can write data in the range of 0.0 to 25.5 in the numeric table.

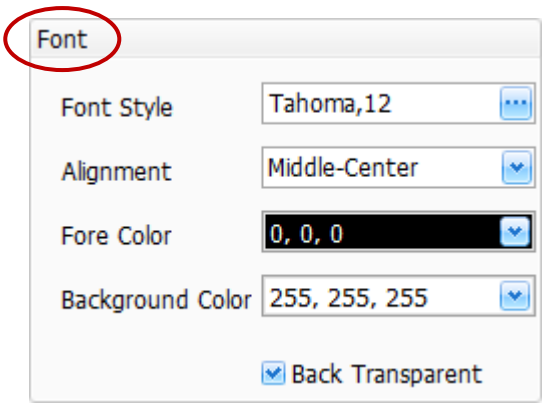


23.0	25.5	03.0
09.6		

- Font Style Properties

  The **Font Style Properties** dialog box is used to configure the attributes of the text used for the **Numeric Table** object.

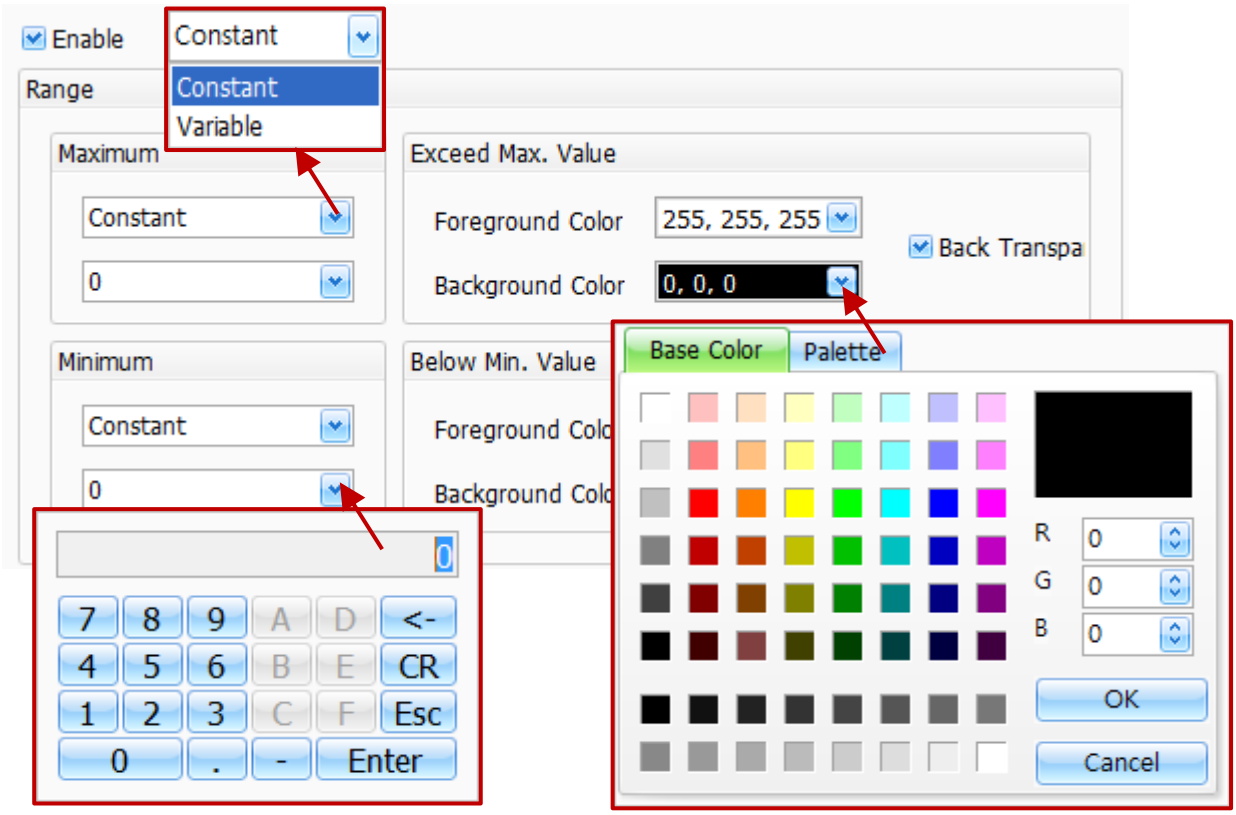
  The following is an overview of the options available in the **Font Style Properties** section of the Property View for the **Numeric Table** object.



Font Style	Used to specify the font and size used for the read value in the table
Alignment	Used to specify the position of the read value within the border of the object
Fore Color	Used to specify the color to be used for the read value in the table
Background Color	Used to specify the color to be used for the background of the object
Back Transparent	Used to specify whether or not the background color will be shown as transparent when displayed on the screen

- Range Properties

  The **Range Properties** dialog box is used to configure the maximum and minimum values and the alarm color for the **Numeric Table** object, which will be activated if the range of value is exceeded.

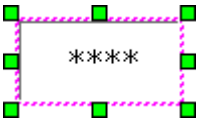


The following is an overview of the options available in the **Range** section of the Property View for the **Numeric Table** object.

Set Range	**Maximum**	Used to specify the maximum value with a constant or a variable
	Minimum	Used to specify the minimum value with a constant or a variable
Exceed Max. Value	**Fore Color**	Used to specify the color to be used for the exceeding value
	Back Color	Used to specify the color to be used for the background of the object
	Back Transparent	Used to specify whether or not the background color will be shown as transparent when displayed on the screen
Below Min. Value	**Fore Color**	Used to specify the color to be used for the exceeding value
	Back Color	Used to specify the color to be used for the background of the object
	Back Transparent	Used to specify whether or not the background color will be shown as transparent when displayed on the screen

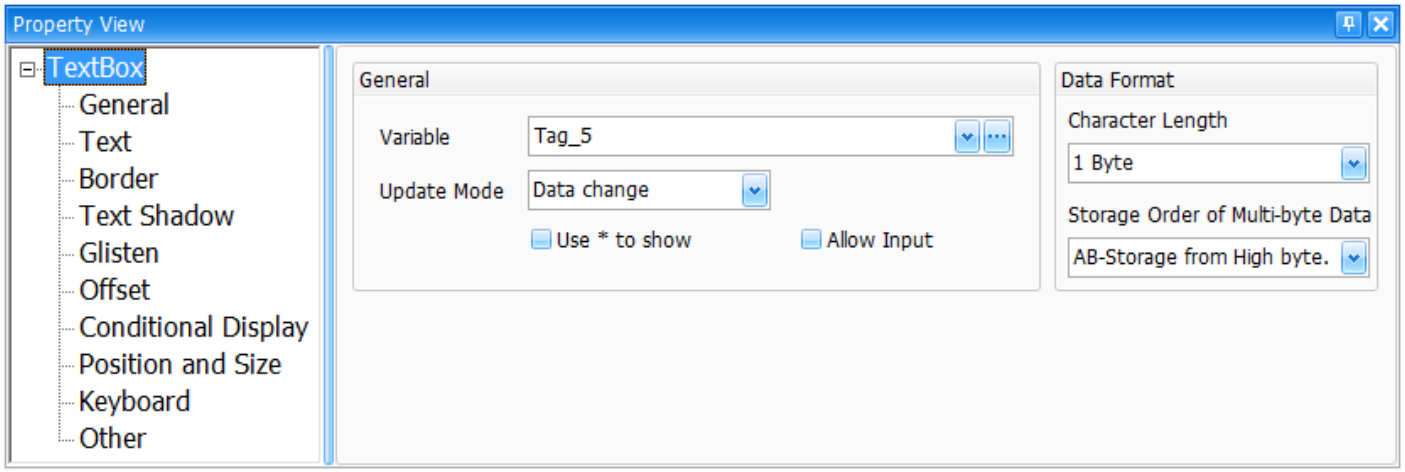
- See Section 8.5 Common Attributes for details about the others attributes of the **Numeric Table** object. Border (1), Text Shadow, Glisten, Security, Offset, Conditional Display, Conditional Control, Position and Size, Keyboard, Event, and Other.

8.2.5 Text Box



The **Text Box** object can be used to display or input a variable value as an ASCII character. The attributes for the **Text Box** object, including the size and style of the font, as well as the alignment, can be adjusted as desired. See Chapter 8 Basic Object Usage for details of how to create an object.

After creating a **Text Box** object, the **Property View** panel for the object will be automatically displayed allowing the various attributes of the object to be configured. The following is a description of how to configure the properties for the **Text Box** object.



- General Properties

  The **General Properties** dialog box is used to configure the update mode and data format of the variable used for the **Text Box** object. For more details related to the usage of variables, see Chapter 7 Variables.

The following is an overview of the options available in the **General Properties** section of the Property View for the **Text Box** object.

General	**Variable**	Used to specify the access variable to be used, which can also be an array variable
	Update Mode	Used to specify the update mode, where: **1**: Change data **2**: Bit detonate (Rising Edge) (0 → 1) **3**: Bit detonate (Falling Edge) (1 → 0) **4**: Bit state change (Rising/Falling Edge) (0 → 1 or 1 → 0)
	Use * to Show	Used to specify whether or not * will be displayed rather than the actual value. This is usually used for passwords or other sensitive information.
	Allow Input	Used to specify whether or not input functions are enabled

Data Format	**Character Length**	Used to specify the amount of memory occupied by each character
	Storage Order of Multi-byte Data	Used to specify the storage order of multi-byte data, where: **AB**: Big Endian The value will be stored from the High byte to the Low byte. E.g., if the variable (Type=Word, Length=1) value is 4142 $_{(Hex)}$, and the character length is set to 1 Byte, the order that the value will be stored in the memory is 41$_{(Hex)}$, 42$_{(Hex)}$. The read data for this object will be AB$_{(ASCII)}$

Variable Value	Memory Address		TextBox Value
4142$_{(Hex)}$	E.g. 1000	E.g. 1001	**A**B$_{(ASCII)}$
	41 $_{(Hex)}$	42$_{(Hex)}$	

In this case, if the character length is set to 2 Byte, the read data for this object will be **A**$_{(ASCII)}$. If you write data to this object, e.g., C$_{(ASCII)}$, , the value will be 4300 $_{(Hex)}$

BA: Little Endian

The value will be stored from the High byte to the Low byte.

E.g., if the variable (Type=Word, Length=1) value is 4142 $_{(Hex)}$, and the character length is set to 1 Byte, the order that the value will be stored in the memory is 42$_{(Hex)}$, 41$_{(Hex)}$.

The read data for this object will be BA$_{(ASCII)}$

Variable Value	Memory Address		TextBox Value
41**42**$_{(Hex)}$	E.g. 1000	E.g. 1001	**B**A$_{(ASCII)}$
	42 $_{(Hex)}$	41$_{(Hex)}$	

In this case, if the character length is set to 2 Byte, the read data for this object will be **B**$_{(ASCII)}$. If you write data to this object, e.g., C$_{(ASCII)}$, , the value will be 0043 $_{(Hex)}$

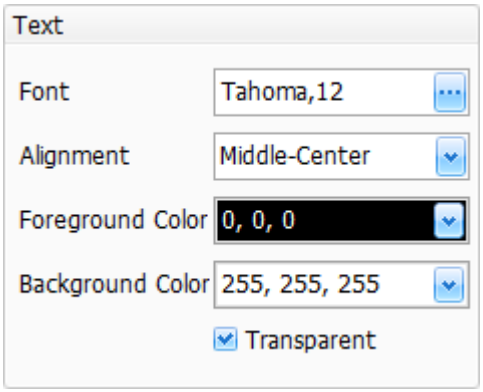
※ The formula for the maximum character length to be displayed and the input is as follows:

The maximum character length **=** the Variable length (in Bytes) **/** the Character length (in Bytes)

The Variable length **=** the Type length of the variable ***** the Array length of the Variable

- Text Properties

  The **Text Properties** dialog box is used to configure the attributes of the text used for the **Text Box** object.

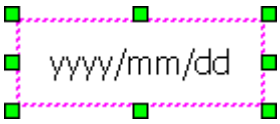


The following is an overview of the options available in the **Text Properties** section of the Property View for the **Text Box** object.

Font	Used to specify the font and size used for the read value in the table
Alignment	Used to specify the position of the read value within the border of the object
Foreground Color	Used to specify the color to be used for the read value in the table
Background Color	Used to specify the color to be used for the background of the object
Transparent	Used to specify whether or not the background color will be shown as transparent when displayed on the screen

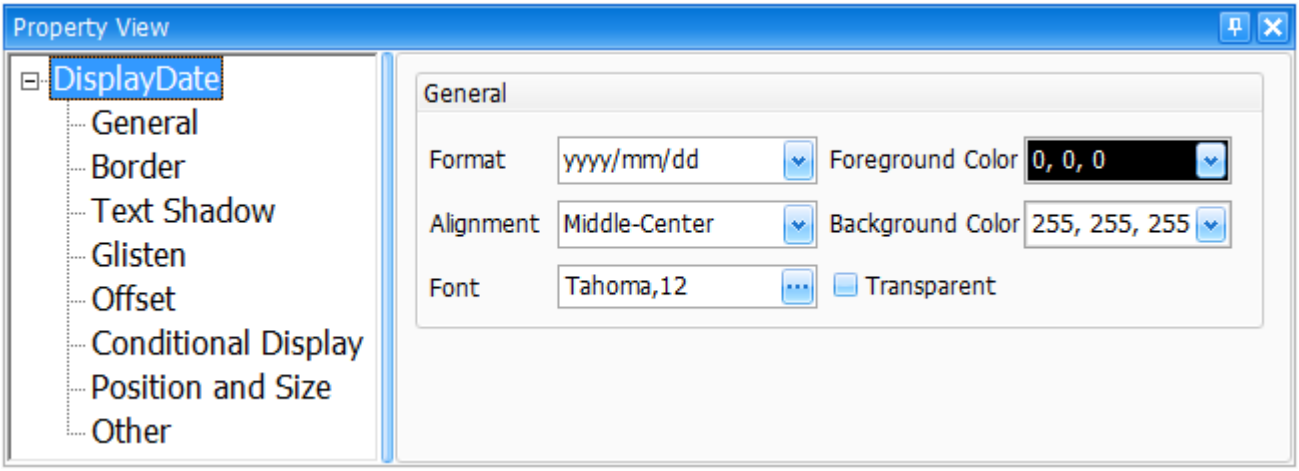
- See Section 8.5 Common Attributes for details about the others attributes of the **Text Box** object. Border (1), Text Shadow, Glisten, Security, Offset, Conditional Display, Position and Size, Keyboard, and Other.

8.2.6 Date



The **Date** object can be used to display the current system date. The attributes for the Date object, including the size and style of the font, as well as the alignment, can be adjusted as desired.

See Chapter 8 Basic Object Usage for details of how to create an object. After creating a **Date** object, the **Property View** panel for the object will be automatically displayed allowing the various attributes of the object to be configured. The following is a description of how to configure the properties for the **Date** object.



- General Properties

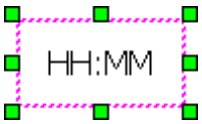
  The **General Properties** dialog box is used to configure the display format for the **Date** object, including the font and colors, etc.

The following is an overview of the options available in the **General Properties** section of the Property View for the **Date** object.

Format	Used to specify the format used to display the date, where: **1**: yyyy/mm/dd (Year, Month, Day) **2**: dd/mm/yyyy (Day, Month, Year) **3**: mm/dd/yyyy (Month, Day, Year)
Alignment	Used to specify the position of the Date text within the border of the object
Font	Used to specify the style and size of the font used for the Date text
Foreground Color	Used to specify the color to be used for the Date text
Background Color	Used to specify the color to be used for the background of the Date object
Transparent	Used to specify whether or not the background will be set to transparent

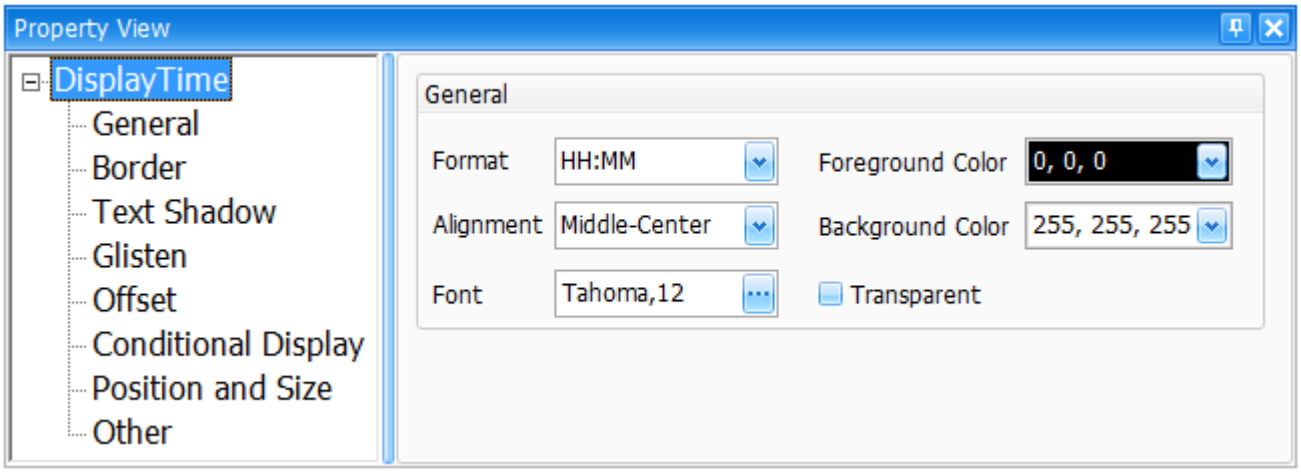
- See Section 8.5 Common Attributes for details about the others attributes of the **Date** object. Border (1), Text Shadow, Glisten, Offset, Conditional Display, Position and Size, and Other.

8.2.7 Time



The **Time** object can be used to display the current system time. The attributes for the Date object, including he size and style of the font, as well as the alignment, can be adjusted as desired.

See Chapter 8 Basic Object Usage for details of how to create an object. After creating a **Time** object, the **Property View** panel for the object will be automatically displayed allowing the various attributes of the object to be configured. The following is a description of how to configure the properties for the **Time** object.



- General Properties

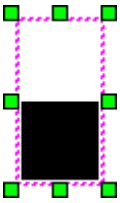
  The **General Properties** dialog box is used to configure the display format for the **Time** object, including the font and colors, etc.

The following is an overview of the options available in the **General Properties** section of the Property View for the **Time** object.

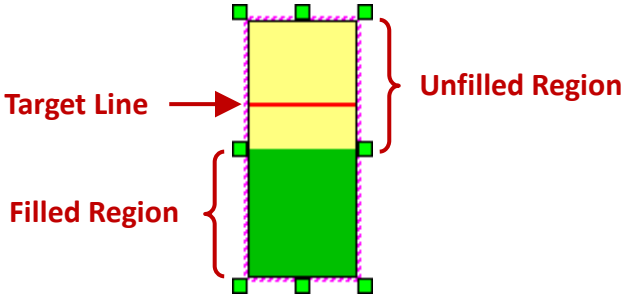
Format	Used to specify the format used to display the time, where: **1**: HH:MM (Hours : Minutes) **2**: HH:MM:SS (Hours : Minutes : Seconds)
Alignment	Used to specify the position of the Time text within the border of the object
Font	Used to specify the style and size of the font used for the Time text
Foreground Color	Used to specify the color to be used for the Time text
Background Color	Used to specify the color to be used for the background of the Time object
Transparent	Used to specify whether or not the background will be set to transparent

- See Section 8.5 Common Attributes for details about the others attributes of the **Time** object. Border (1), Text Shadow, Glisten, Offset, Conditional Display, Position and Size, and Other.

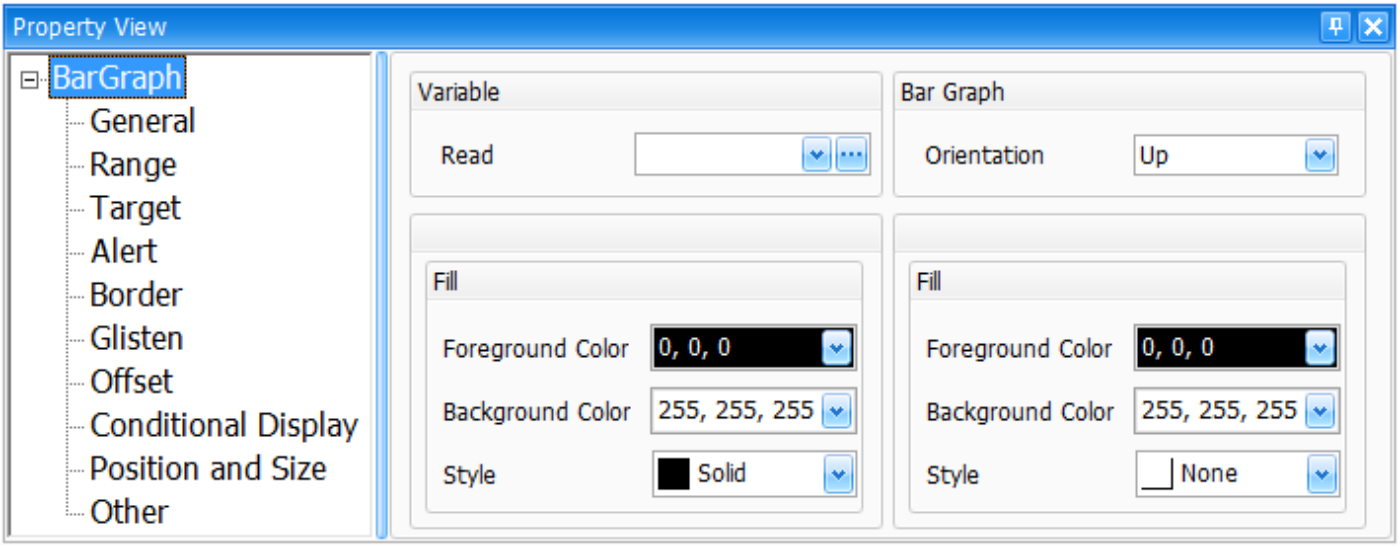
8.2.8 Bar Graph



The **Bar Graph** object can be used to display a variable value as a bar graph. The attributes for the Bar Graph object, including the size and style of the font, as well as the alignment, can be adjusted as desired. See Chapter 8 Basic Object Usage for details of how to create an object.



After creating a **Bar Graph** object, the **Property View** panel for the object will be automatically displayed allowing the various attributes of the object to be configured. The following is a description of how to configure the properties for the **Bar Graph** object.



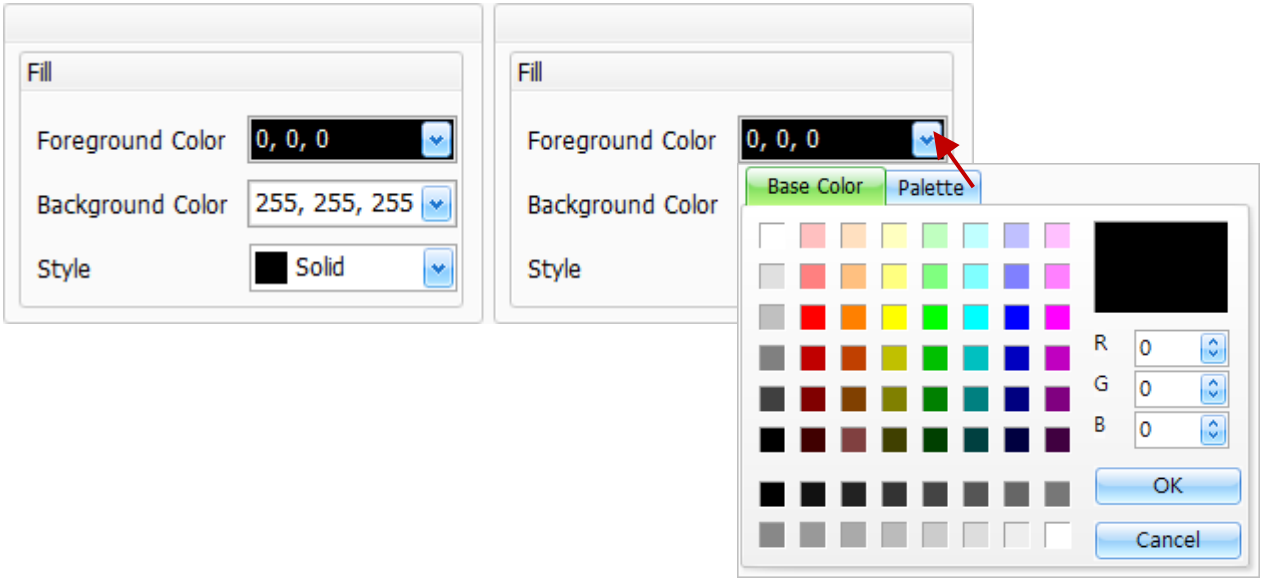
- General Properties

  The **General Properties** dialog box is used to configure the variable attributes and format for the **Bar Graph** object, including the font and colors, etc. For more details related to the usage of variables, see Chapter 7 Variables.

The following is an overview of the options available in the **General Properties** section of the Property View for the **Bar Graph** object.

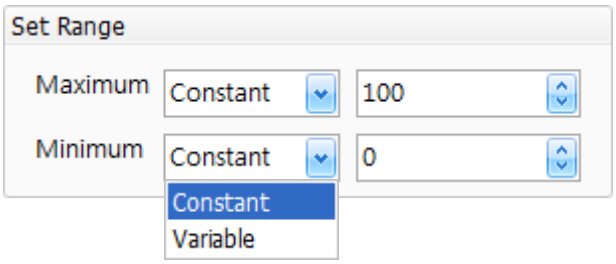
Variable	**Read**	Used to specify the variable to be read
Bar Graph	**Orientation**	Used to specify the orientation of the bars in the Bar Graph and can be Up, Down, Left, or Right

Fill (Set Filled Region)	**Foreground Color**	Used to specify the base color and the palette to be used for the foreground color of the filled region
	Background Color	Used to specify the base color and the palette to be used for the background of the filled region
	Style	Used to specify the style of the fill
Fill (Set Unfilled Region)	**Foreground Color**	Used to specify the base color and the palette to be used for the foreground color of the unfilled region
	Background Color	Used to specify the base color and the palette to be used for the background of the unfilled region
	Style	Used to specify the style of the fill



- Range Properties

  The **Range Properties** dialog box is used to configure the maximum and minimum values of the scales used for the **Bar Graph** object by using the constant or the variable. For more details related to the usage of variables, see Chapter 7 Variables.

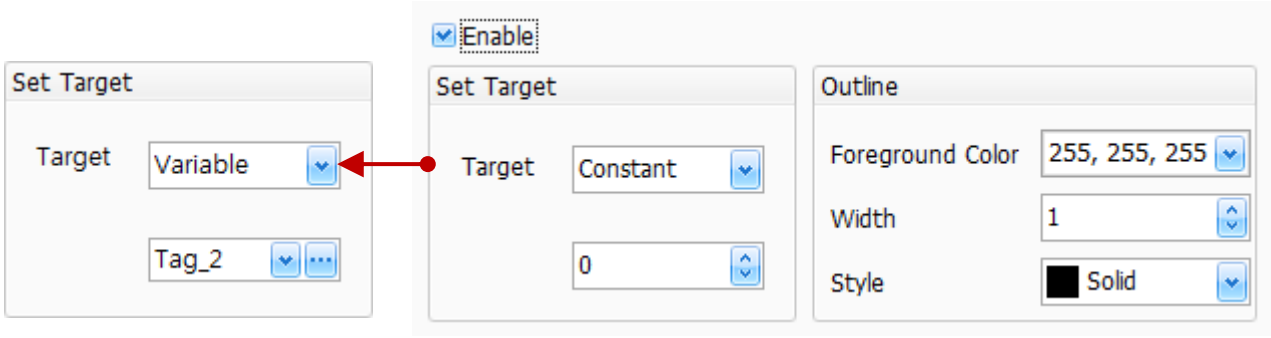


The following is an overview of the options available in the **Range Properties** section of the Property View for the **Bar Graph** object.

Maximum	Used to specify the maximum value for the Bar Graph scale
Minimum	Used to specify the minimum value for the Bar Graph scale

- Target Properties

The **Target Properties** dialog box is used to configure the target value and the line style for the **Bar Graph** object.

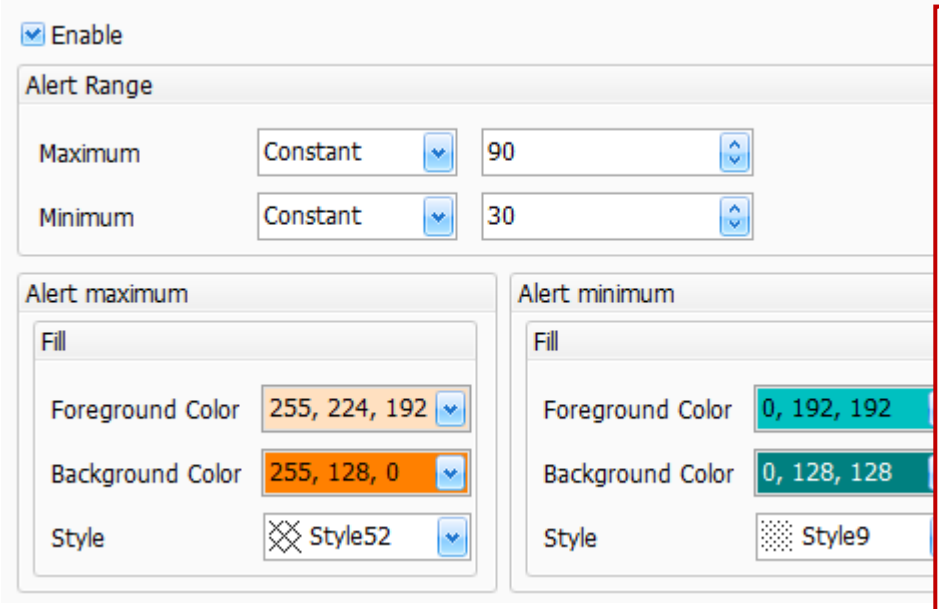


The following is an overview of the options available in the **Target Properties** section of the Property View for the **Bar Graph** object.

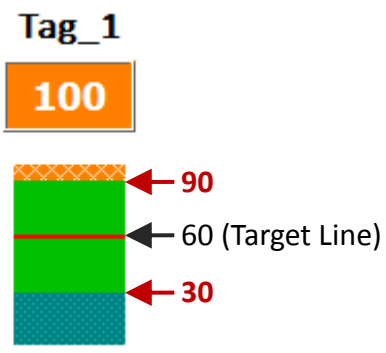
Set Target	**Target**	Used to specify the way to set the target value, and can be set to Constant or Variable
	Value	Used to specify a value or a variable as a target value
Outline	**Foreground Color**	Used to specify the color to be used for the target line
	Width	Used to specify the width to be used for the target line
	Style	Used to specify the style to be used for the target line

- Alert Properties

The **Alert Properties** dialog box for the **Bar Graph** object is used to configure the alert range and the display style of the object if the maximum or minimum value of Alert is exceeded.



E.g., if the read value is 100, the Bar Graph will be displayed as the figure below:



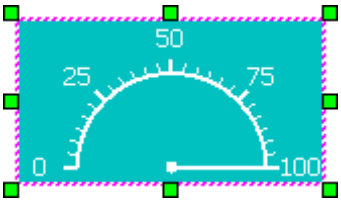
The following is an overview of the options available in the **Alert Properties** section of the Property View for the **Bar Graph** object.

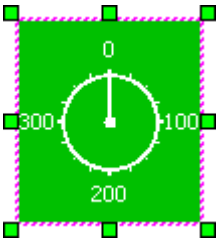
Set Alert Range	**Maximum**	Used to specify the maximum value for the Alert Range
	Minimum	Used to specify the minimum value for the Alert Range
Alert Maximum	**Foreground Color**	Used to specify the foreground color be displayed when the maximum alert value is exceeded
	Background Color	Used to specify the background color be displayed when the maximum alert value is exceeded
	Style	Used to specify the style of the fill
Alert Minimum	**Foreground Color**	Used to specify the foreground color be displayed when the minimum alert value is exceeded
	Background Color	Used to specify the background color be displayed when the minimum alert value is exceeded
	Style	Used to specify the style of the fill

- See Section 8.5 Common Attributes for details about the others attributes of the **Bar Graph** object. Border (1), Glisten, Offset, Conditional Display, Position and Size, and Other.

8.2.9 Dial-Semicircle / Dial-Circle

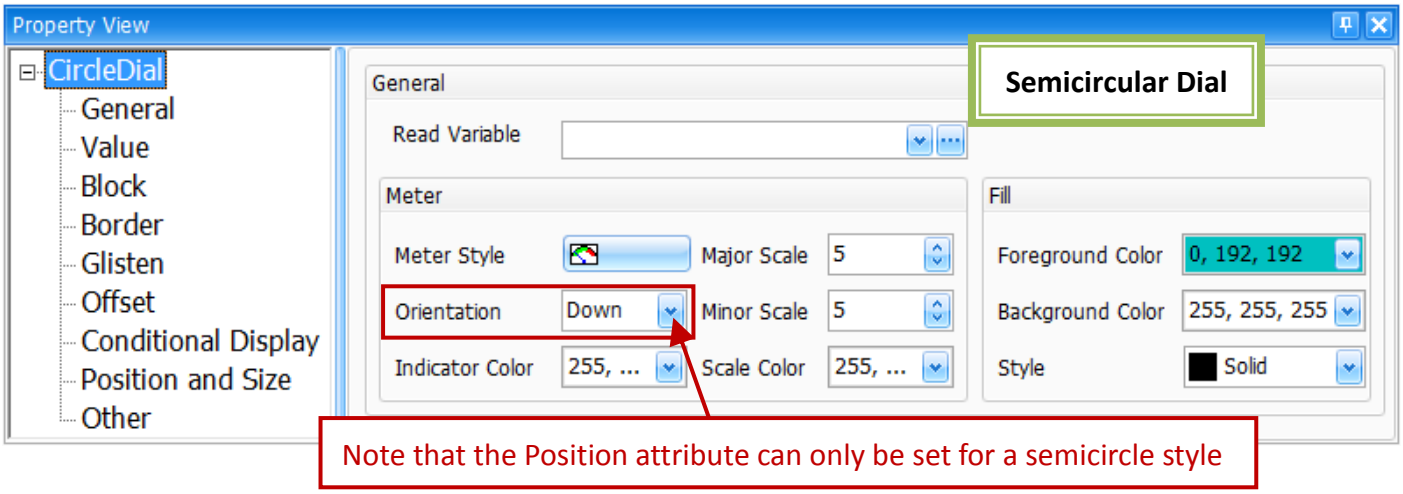
The **Circular/Semicircular Dial** object can be used to graphically display a variable value as either a circular or a semicircular Dial- meter. The attributes for the Circular/Semicircular Dial object, including the size and style of the font, as well as the alignment, can be adjusted as desired. See Chapter 8 Basic Object Usage for details of how to create an object.

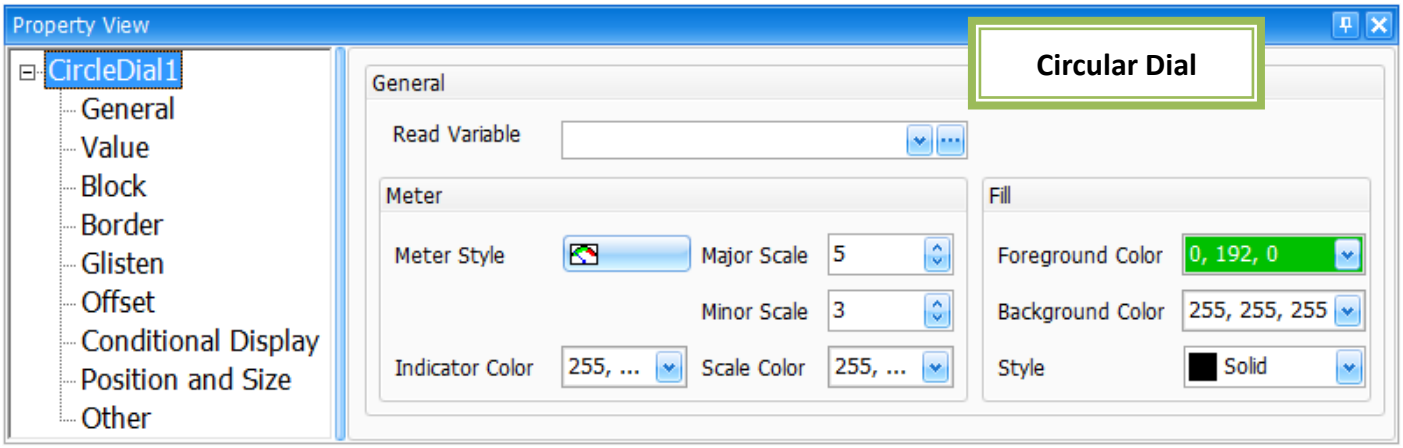
(Semi-circular Dial) 



(Circular Dial)

After creating a **Circular/Semicircular Dial** object, the **Property View** panel for the object will be automatically displayed allowing the various attributes of the object to be configured. The following is a description of how to configure the properties for the **Circular/Semicircular Dial** object.

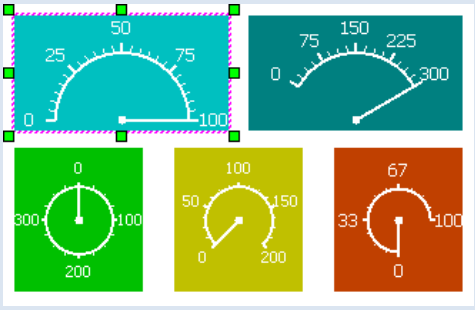
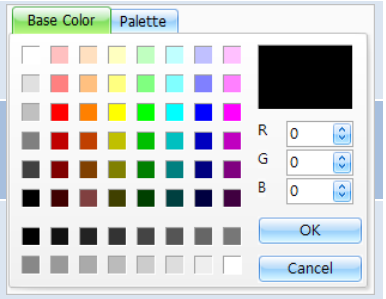




- General Properties

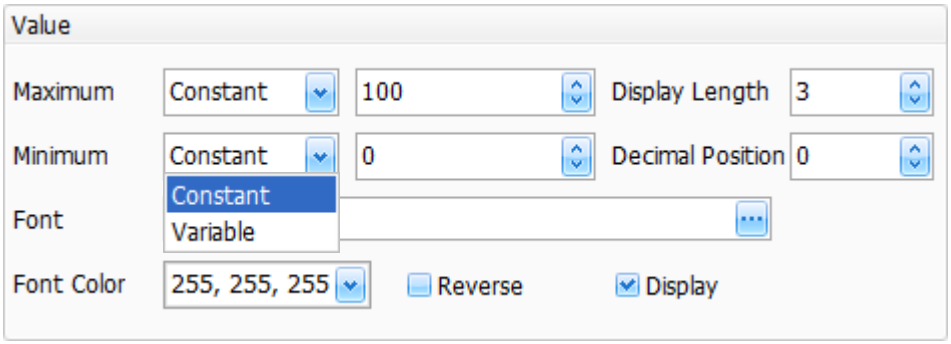
  The **General Properties** dialog box is used to configure the variable attributes and style to be used for the meter of the **Circular/Semicircular Dial** object.

The following is an overview of the options available in the **General Properties** section of the Property View for the **Circular/Semicircular Dial** object.

General	**Read Variable**	Used to specify the variable to be read. For more details related to the usage of variables, see Chapter 7 Variables.
Meter	**Meter Style**	Used to specify the style of meter to be used, depending on the type of dial selected. ● For a Semicircle style meter **1.** 180 degrees **2.** 120 degrees ● For a Circle style meter **1.** 360 degrees **2.** 270 degrees  
	Orientation	Used to specify the base position of the meter: Left, Right, Up, Down **Note that the Position parameter can only be set for a semicircle style dial**
	Indicator Color	Used to specify the color of the indicator
	Major Scale	Used to specify the interval used for the main scale
	Minor Scale	Used to specify the interval used for the minor scale
	Scale Color	Used to specify the color to be used for the scales
Fill	**Foreground Color**	Used to specify the base color and the palette to be used for the foreground color 
	Background Color	Used to specify the base color and the palette to be used for the background color
	Style	Used to specify the style of the fill, and can be selected from: None, Solid, and Style1 to Style52

- Value Properties

The **Value Properties** dialog box is used to configure the values to be displayed for the scale of the meter together with the format.

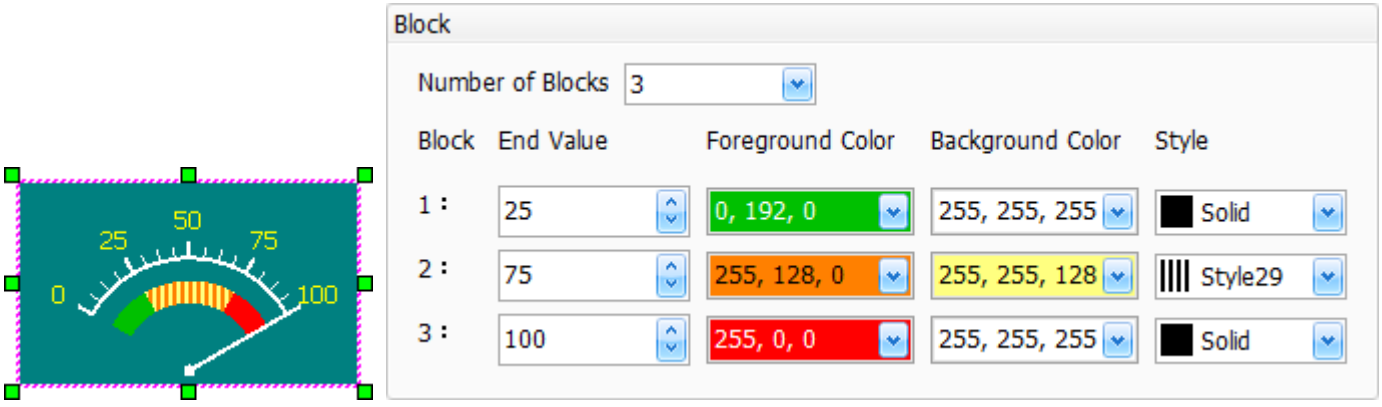


The following is an overview of the options available in the **Value Properties** section of the Property View for the **Circular/Semicircular Dial** object.

Maximum	Used to specify the maximum value for the scale with a constant or a variable
Minimum	Used to specify the minimum value for the scale with a constant or a variable
Display Length	Used to specify the number of digits that will be displayed
Decimal Position	Used to specify the position of the decimal point (read from the right). E.g., if the Display Length = 3 and the Decimal Position = 1, then the displayed value “100” (or 50) will be shown as “10.0” (or 5.0)
Font	Used to specify the type and size of the font to be used for the scale value
Font Color	Used to specify the color of the font to be used for the scale value
Reverse	Used to specify that the value will be shown in reverse order
Display	Used to specify whether or not the values will be displayed

- Block Properties

The **Block Properties** dialog box is used to configure a meter block and its style. Note that a maximum of five blocks can be configured.

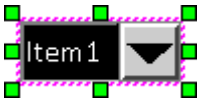


The following is an overview of the options available in the **Block Properties** section of the Property View for the **Circular/Semicircular Dial** object.

Number of Blocks	Used to specify the number of blocks to be used for the meter
Block End Value	Used to specify the ending value for the color-coded blocks
Foreground Color	Used to specify the foreground color of the fill style to be used for the block
Background Color	Used to specify the background color of the fill style to be used for the block
Style	Used to specify the style of the fill

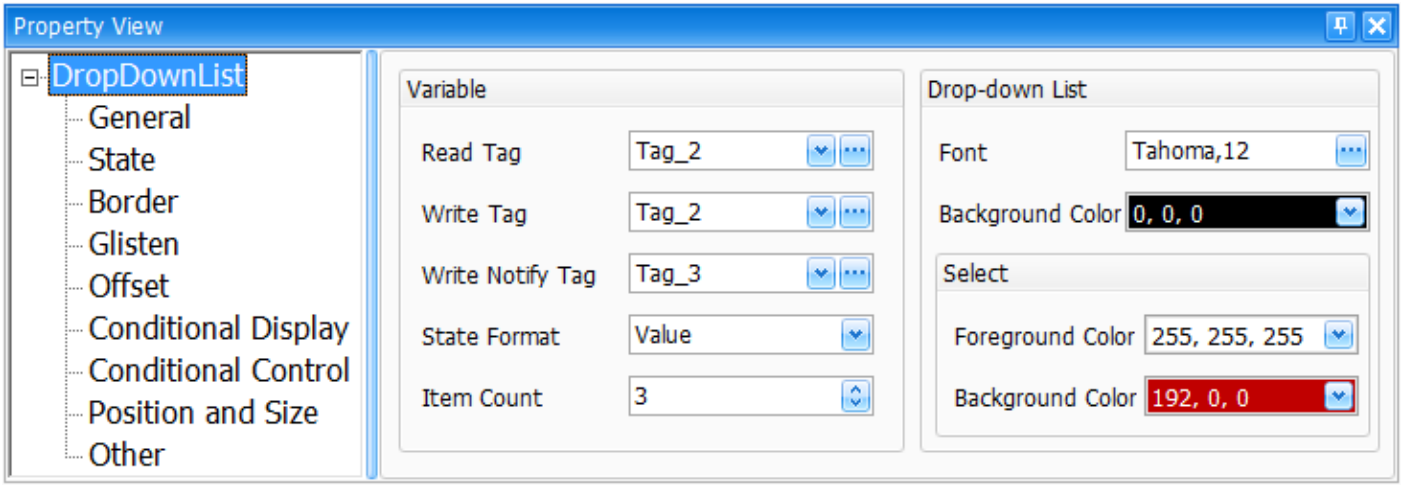
- See Section 8.5 Common Attributes for details about the others attributes of the **Circular/Semicircular Dial** object. Border (1), Glisten, Offset, Conditional Display, Position and Size, and Other.

8.2.10 Drop Down List



The **Drop-down List** object can be used to display a prescribed number of variable values as a drop-down list. The attributes for the Drop-down List object, including the size and style of the font, as well as the alignment, can be adjusted as desired. See Chapter 8 Basic Object Usage for details of how to create an object.

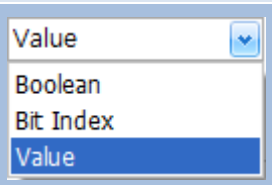
After creating a **Drop-down List** object, the **Property View** panel for the object will be automatically displayed allowing the various attributes of the object to be configured. The following is a description of how to configure the properties for the **Drop-down List** object.



- General Properties

  The **General Properties** dialog box is used to configure the variables and the state values to be used for the **Drop-down List** object, together with the attributes such as font, size and color for the text to be displayed. For more details related to the usage of variables, see Chapter 7 Variables.

  The following is an overview of the options available in the **General Properties** section of the Property View for the **Drop-down List** object.

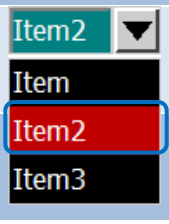
Variable	**Read Tag**	Used to specify which variable is used to read data
	Write Tag	Used to specify which variable is used to write data
	Write Notify Tag	Used to specify the write notification variable and set it to 1.
	For more details related to the usage of variables, see Chapter 7 Variables.	
	State Format	Used to specify the format that will be used to display the state value, as described below.  **Boolean:** If the variable value is equal to 0, the state value will be 0. If the variable value is not equal to 0, the state value will be 1.

<table>
<tr><td rowspan="4">Variable</td><td rowspan="3">State Format</td><td>For example, if the value of a variable which data type is WORD (16-bit) is 32, the text will be displayed depends on the setting of state value 1. (See State Property for more details)

State Value Range: 0, 1</td></tr>
<tr><td><u>Bit Index:</u>
If the variable value is equal to 2^n, the state value will be n.
If the variable value is not equal to 2^n, no text will be displayed.

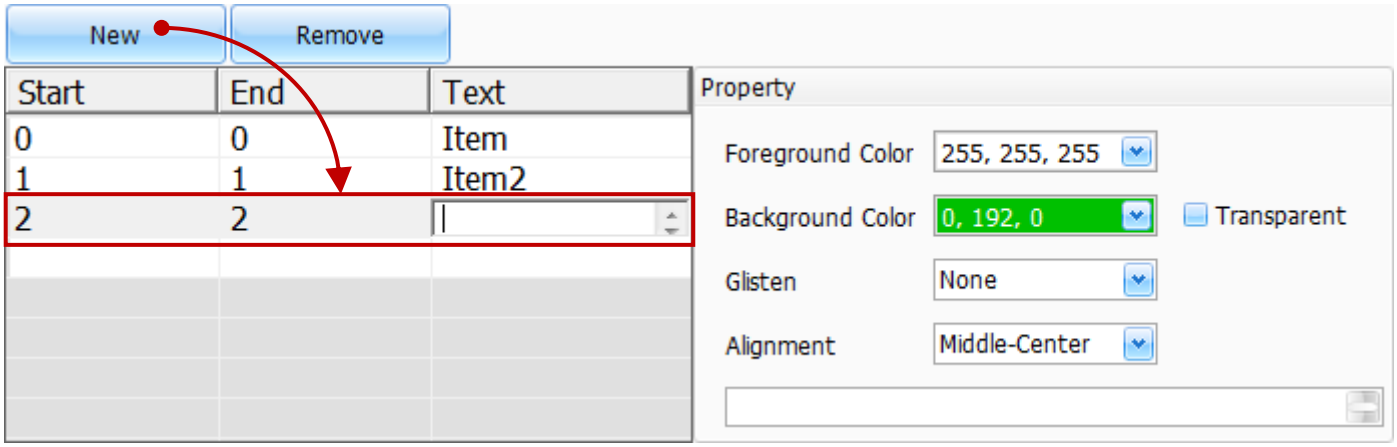
For example, if the value of a variable which data type is BYTE (8-bit) is 32 (i.e., 2^5), the text will be displayed depends on the setting of state value 5.
MSB LSB
<table><tr><td>2^7</td><td>2^6</td><td>2^5</td><td>2^4</td><td>2^3</td><td>2^2</td><td>2^1</td><td>2^0</td></tr><tr><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table>
State Value Range: 0 to 31</td></tr>
<tr><td><u>Value:</u>
The variable value is the state value.

For example, if the value of a variable which data type is DWORD (32-bit) is 32, the text will be displayed depends on the setting of state value 32.

State Value Range: 0 to 2147483647</td></tr>
<tr><td>Item Count</td><td>Used to specify the number of visible rows to be shown first in the drop-down list (e.g., 3)</td></tr>
<tr><td rowspan="2">Drop-down List</td><td>Font</td><td>Used to specify the font and size for the text to be displayed in the drop-down list</td></tr>
<tr><td>Background Color</td><td>Used to specify the color to be used for the background of the drop-down list</td></tr>
<tr><td rowspan="2">Select</td><td>Foreground Color</td><td rowspan="2">Used to specify the color of the text to be used for the selected item in the drop-down list
Used to specify the color of the background to be used for the selected item in the drop-down list
  </td></tr>
<tr><td>Background Color</td></tr>
</table>

- State Properties

  The **State Properties** dialog box is used to configure the state display and the text for the drop-down list.

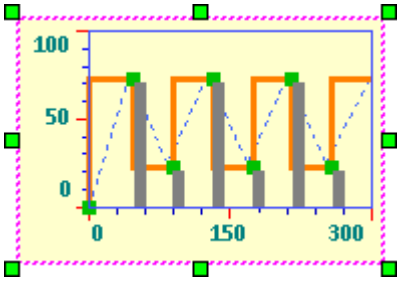


The following is an overview of the options available in the **State Properties** section of the Property View for the **Drop-down List** object.

New	Used to add an item, and then you can set the text to be displayed when the variable value meets the Start/End values.
Remove	Used to remove the selected item
Foreground Color	Used to specify the text color, e.g., “Item3” as the figure 
Background Color	Used to specify the color to be used for the background of the text
Transparent	Used to specify whether or not the background color will be shown as transparent when displayed on the screen 
Glisten	Used to specify the Glisten speed for the selected item
Alignment	Used to specify the position of the text within the border of the object

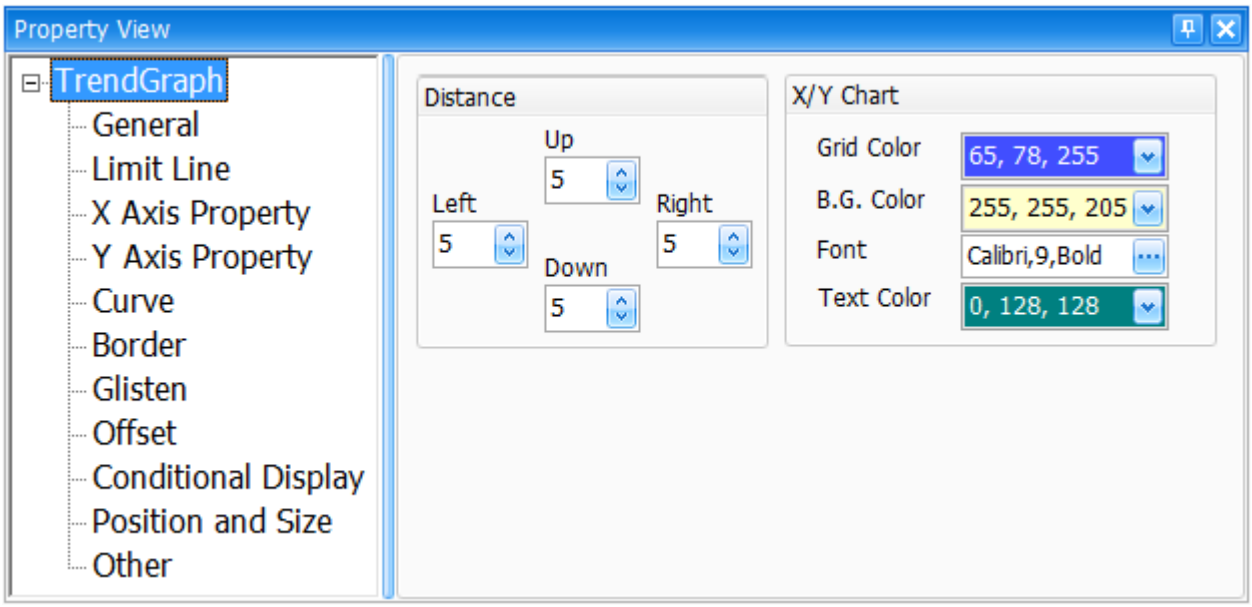
- See Section 8.5 Common Attributes for details about the others attributes of the **Drop-down List** object. Border (1), Glisten, Offset, Conditional Display, Conditional Control, Position and Size, and Other.

8.2.11 Trend Graph (XY Chart)



The **Trend Graph** object can be used to display a range of variable values as a Trend Graph. The attributes for the Trend Graph object, including the size and style of the font, as well as the color, can be adjusted as desired.
See Chapter 8 Basic Object Usage for details of how to create an object.

After creating a **Trend Graph** object, the **Property View** panel for the object will be automatically displayed allowing the various attributes of the object to be configured. The following is a description of how to configure the properties for the **Trend Graph** object.



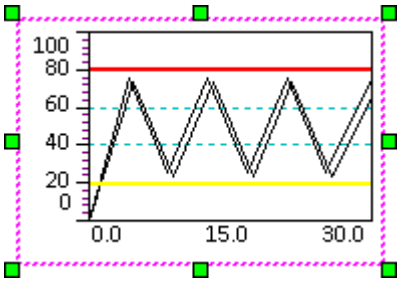
- General Properties

  The **General Properties** dialog box is used to configure the display format for the trend graph.

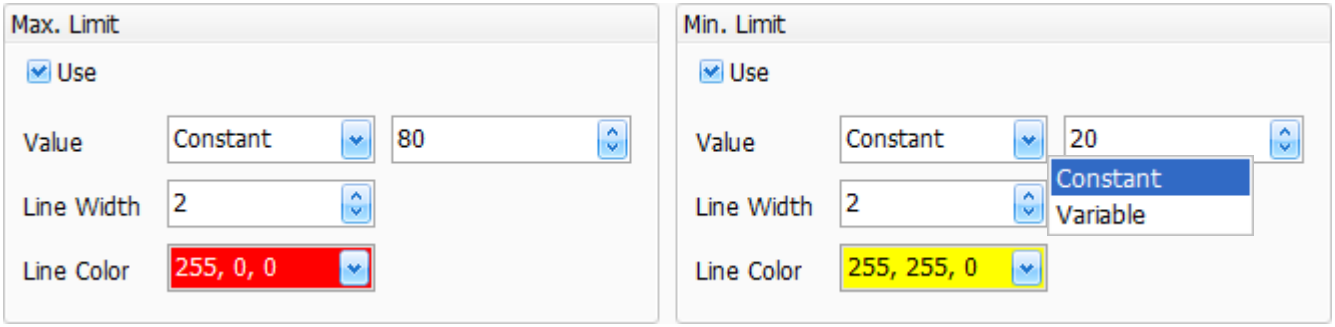
The following is an overview of the options available in the **General Properties** section of the Property View for the **Trend Graph** object.

Distance (pixels)	Used to specify the up/down/left/right margin for the trend graph
Grid Color	Used to specify the color of the grid lines for the trend graph
B.G. Color	Used to specify the color of the background to be used for the trend graph
Font	Used to specify the type and size of the font to be used for the text on the trend graph
Text Color	Used to specify the color of the text for the trend graph

- Limit Line Properties



The **Limit Line Properties** dialog box is used to configure maximum and minimum limits for the trend graph, including the limit value based on either a constant or a variable, the width of the line and the color. For more details related to the usage of variables, see Chapter 7 Variables.

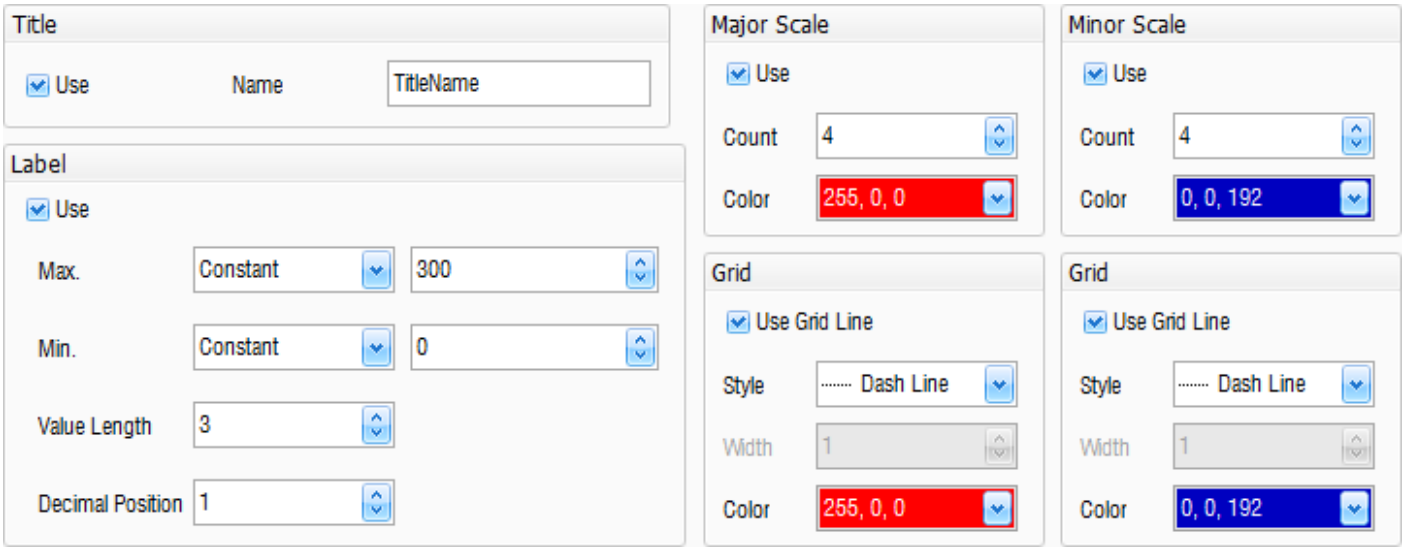


The following is an overview of the options available in the **Limit Line Properties** section of the Property View for the **Trend Graph** object.

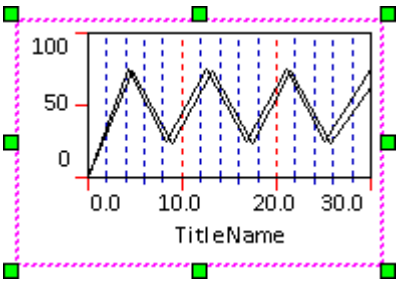
Max. Limit Line	**Value**	Used to specify the upper limit for the y-axis of the trend graph
	Line Width	Used to specify the width of the line in pixels
	Line Color	Used to specify the color of the line to indicate the maximum limit
Min. Limit Line	**Value**	Used to specify the lower limit for the x-axis of the trend graph
	Line Width	Used to specify the width of the line in pixels
	Line Color	Used to specify the color of the line to indicate the minimum limit

- X-axis Properties

The **X-axis Properties** dialog box is used to configure the title, the label, the major and minor scale, and the grid lines for the X-axis of the **Trend Graph** object.



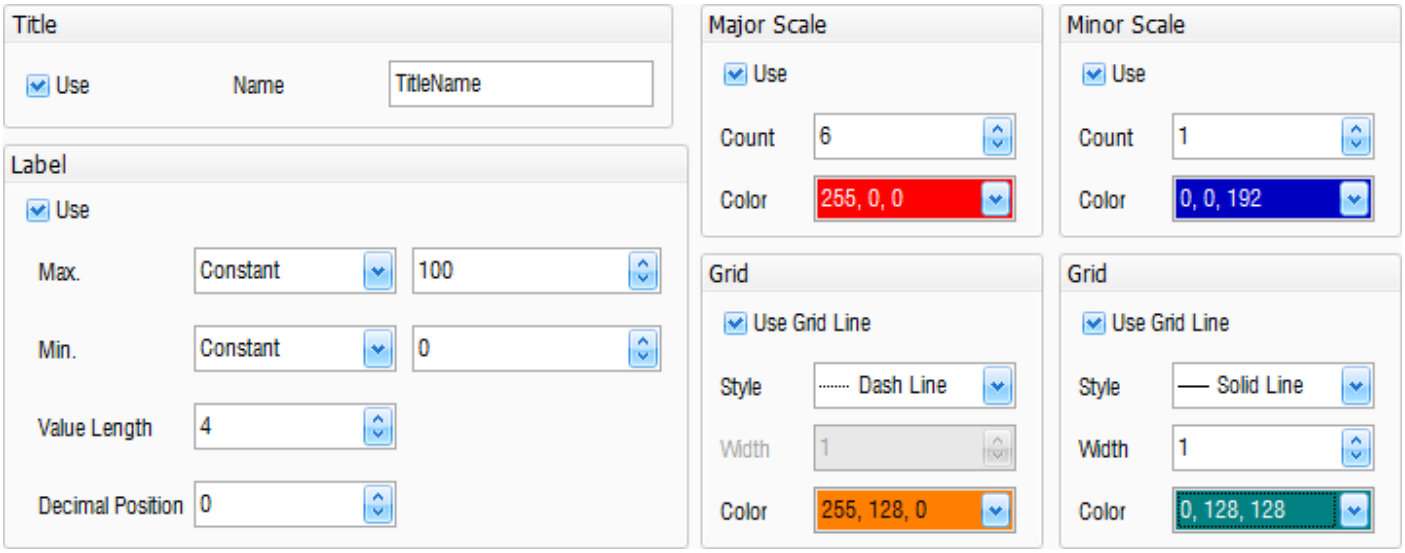
The following is an overview of the options available in the **X-axis Properties** section of the Property View for the **Trend Graph** object.



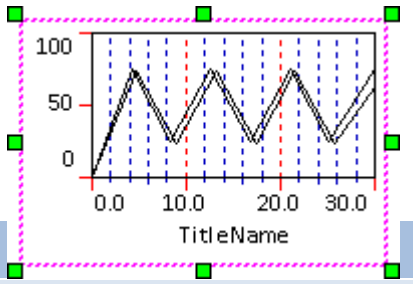
Title	**Name**	Used to specify a title for the X-axis
Label	**Max**	Used to specify the maximum value to be displayed for the X-axis
	Min	Used to specify the minimum value to be displayed for the X-axis
	Value Length	Used to specify the number of digits that will be displayed
	Decimal Position	Used to specify the position of the decimal point (read from the right). E.g., if the Display Length = 3, and the Decimal Position = 1, then the value “300” will be displayed as “30.0”
Major Scale	**Count**	Used to specify the interval to be used for the major scale of the X-axis
	Color	Used to specify the color to be used for the major scale of the X-axis
Minor Scale	**Count**	Used to specify the interval to be used for the minor scale of the X-axis
	Color	Used to specify the color to be used for the minor scale of the X-axis
Grid (Major Scale)	**Use Grid Lines**	Used to specify whether or not grid lines should be displayed for the major scale of the X-axis
	Style	Used to specify the style of the grid to be used for the major scale of the X-axis
	Width	Used to specify the width of the grid lines to be used for the major scale of the X-axis
	Color	Used to specify the color of the grid lines to be used for the major scale of the X-axis
Grid (Minor Scale)	**Use Grid Lines**	Used to specify whether or not grid lines should be displayed for the minor scale of the X-axis
	Style	Used to specify the style of the grid to be used for the minor scale of the X-axis
	Width	Used to specify the width of the grid lines to be used for the minor scale of the X-axis
	Color	Used to specify the color of the grid lines to be used for the minor scale of the X-axis

- Y-axis Properties

  The **Y-axis Properties** dialog box is used to configure the title, the label, the major and minor scale, and the grid lines for the Y-axis of the **Trend Graph** object.



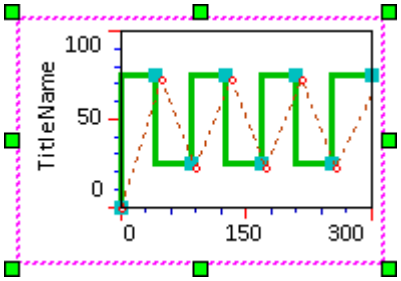
The following is an overview of the options available in the Y-axis Properties section of the Property View for the Trend Graph object.

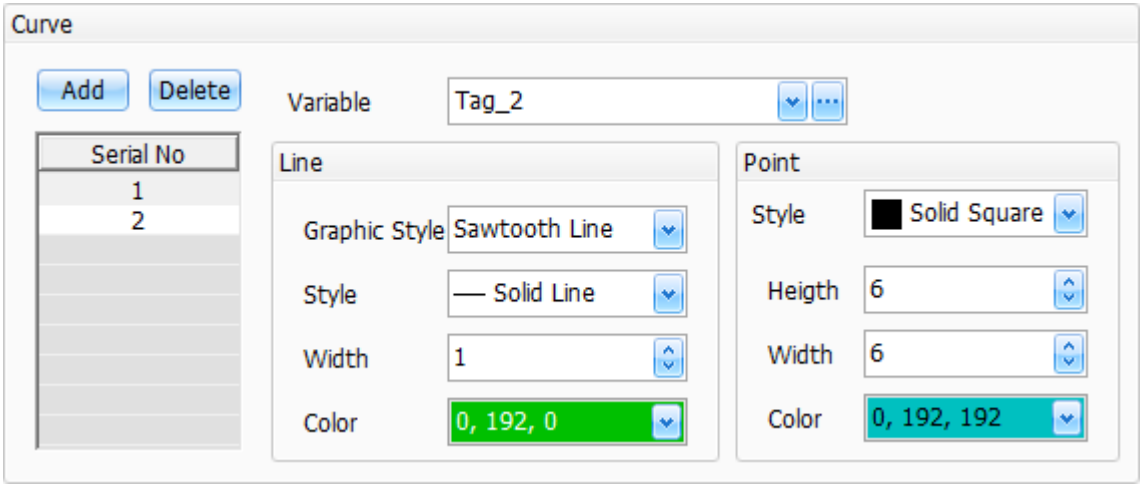


Title	**Name**	Used to specify a title for the Y-axis
Label	**Max**	Used to specify the maximum value to be displayed for the Y-axis
	Min	Used to specify the minimum value to be displayed for the Y-axis
	Display Length	Used to specify the number of digits that will be displayed
	Decimal Position	Used to specify the position of the decimal point (read from the right). E.g., if the Display Length = 4, and the Decimal Position = 1, then the value “100” will be displayed as “10.0”
Major/Minor Scale	**Count**	Used to specify the interval to be used for the Major/Minor scale of the Y-axis
	Color	Used to specify the color to be used for the Major/Minor scale of the Y-axis
Grid (Major/Minor Scale)	**Use Grid Lines**	Used to specify whether or not grid lines should be displayed for the Major/Minor scale of the Y-axis
	Style	Used to specify the style of the grid to be used for the Major/Minor scale of the Y-axis
	Width	Used to specify the width of the grid lines to be used for the Major/Minor scale of the Y-axis
	Color	Used to specify the color of the grid lines to be used for the Major/Minor scale of the Y-axis

- Curve Properties

  The **Curve Properties** dialog box allows the attributes and style used to display the curve for the **Trend Graph** object to be configured, and is based on a variable. For more details related to the usage of variables, see Chapter 7 Variables.





The following is an overview of the options available in the **Curve Properties** section of the Property View for the **Trend Graph** object.

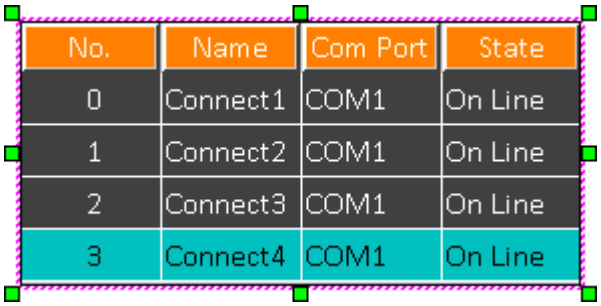
Add		Used to add a curve to the trend graph (e.g., No.3, No.4, etc.)
Delete		Used to remove one or more curve from the trend graph
Variable		Used to specify the variable to be used for the curve. For more details related to the usage of variables, see Chapter 7 Variables
Line Set	**Graphic Style**	Used to specify the style to be used for the curve, and can be selected from: Curve, Sawtooth Line, or Bar Chart
	Style	Used to specify the line style to be used for the curve, and can be selected from: Solid Line or Dash Line
	Width	Used to specify the width to be used for the curve
	Color	Used to specify the color to be used for the curve
Point Set	**Style**	Used to specify the style to be used for the trace point
	Height	Used to specify the height to be used for the trace point
	Width	Used to specify the width to be used for the trace point
	Color	Used to specify the color to be used for the trace point

- See Section 8.5 Common Attributes for details about the others attributes of the **Trend Graph** object. Border (1), Glisten, Offset, Conditional Display, Position and Size, and Other.

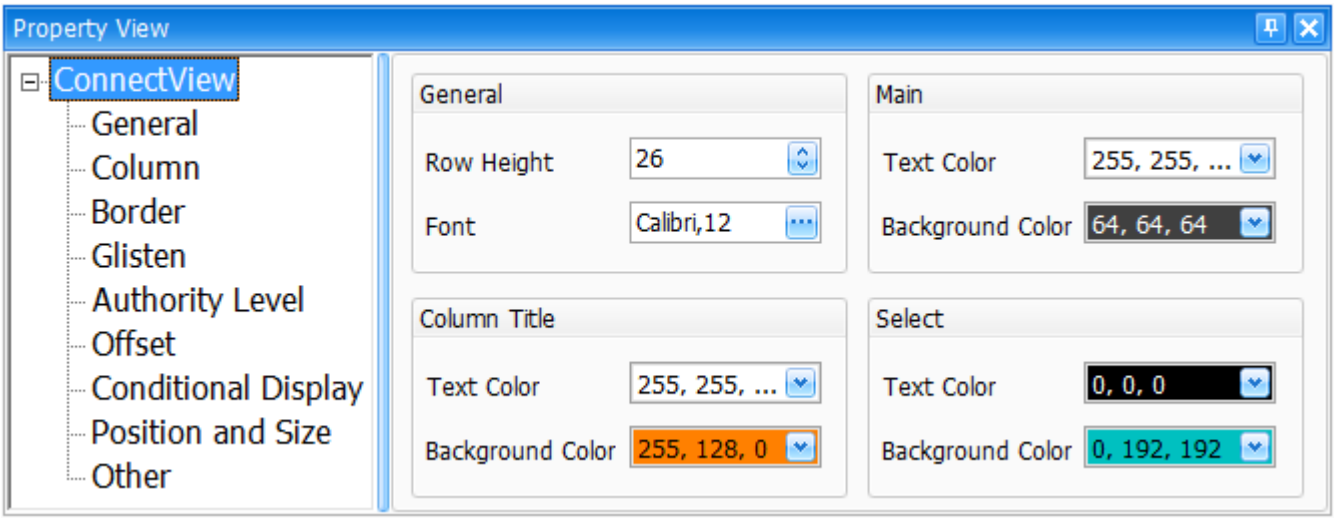
8.2.12 Connection View (Available soon)

The **Connection View** object can be used to display connection data that allows the status of the connection to be monitored and controlled. See Chapter 8 Basic Object Usage for details of how to create an object.

After creating a **Connection View** object, the **Property View** panel for the object will be automatically displayed allowing the various attributes of the object to be configured.



The following is a description of how to configure the properties for the **Connection View** object.



- General Properties

  The **General Properties** dialog box is used to configure the display format for the **Connection View** object.

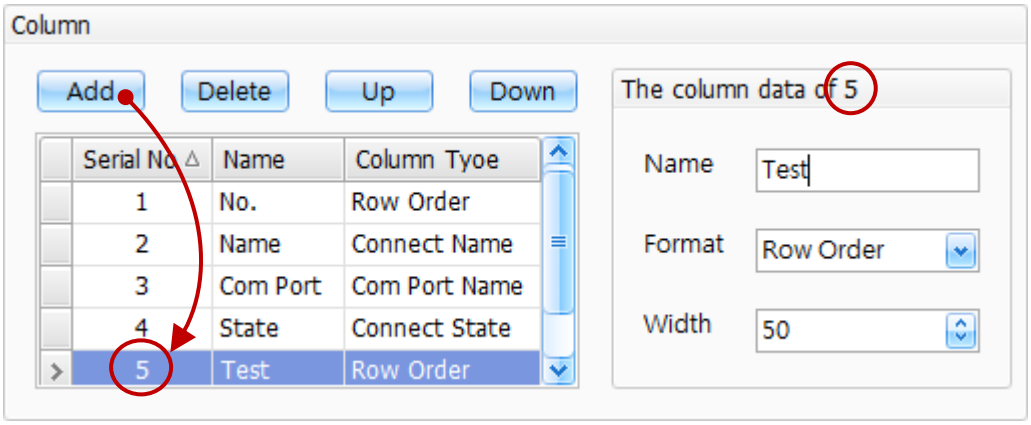
The following is an overview of the options available in the **General Properties** section of the Property View for the **Connection View** object.

General	**Row Height**	Used to specify the height of the table row
	Font	Used to specify the type and size of the font to be used for the text
Column Title	**Text Color**	Used to specify the color of the text to be used for the title
	Background Color	Used to specify the color of the background to be used for the title
Main	**Text Color**	Used to specify the color of the text to be used for the list data
	Background Color	Used to specify the color of the background to be used for data cells

Select	**Text Color**	Used to specify the color to be used for the list data text when it has been selected
	Background Color	Used to specify the color of the background of the data cell when it has been selected

- Column Properties

  The **Column Properties** dialog box is used to configure the attributes of the data table for the **Connection View** object, including the number of columns, the width, and the format.



The following is an overview of the options available in the **Column Properties** section of the Property View for the **Connection View** object.

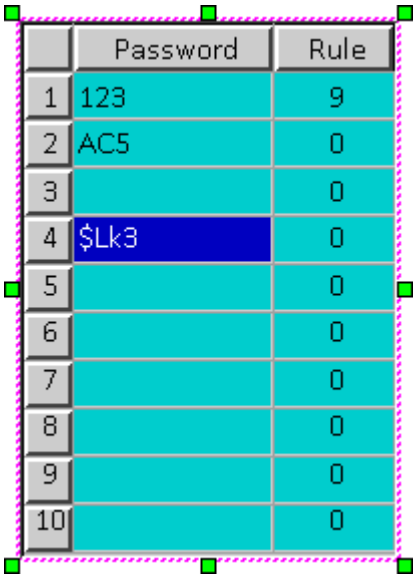
Column	**Add**	Used to add a specific column to the data table
	Delete	Used to delete a specific column from the data table
	Up	Used to move the selected row up, i.e., move the specific column to the left in the data table
	Down	Used to move the selected row down, i.e., move the specific column to the right in the data table
Column Data	**Name**	Used to specify name for the column, which will be used for the column headings in the data table
	Format	Used to specify the format of the column, and can be selected from: Row Order, Connection Name, COM Port, and Connection State.
	Width	Used to specify the width of the column

- See Section 8.5 Common Attributes for details about the others attributes of the object.

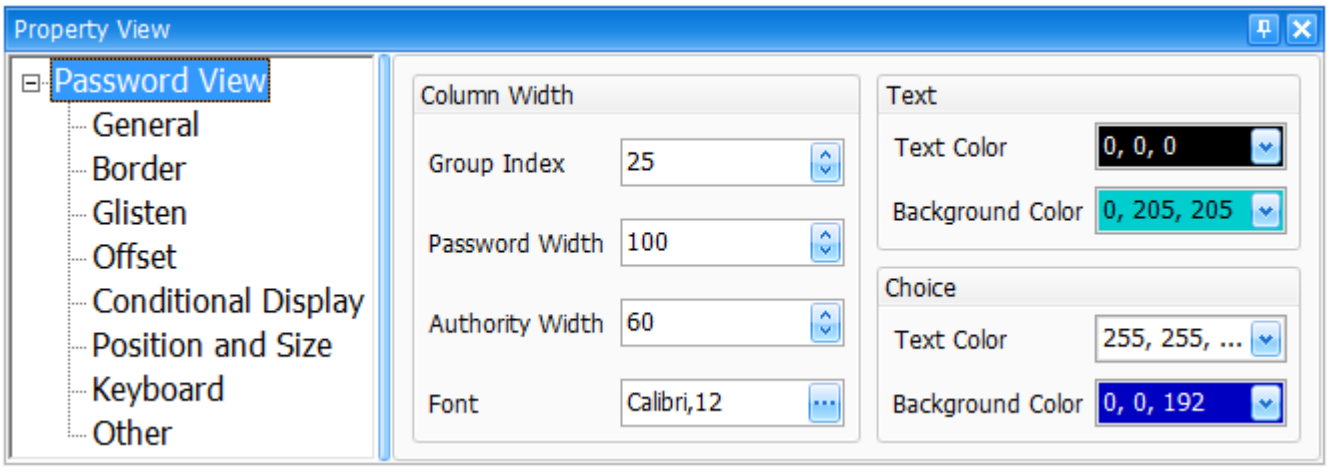
  Border (1), Glisten, Offset, Authority Level (Security), Conditional Display, Position and Size, and Other.

8.2.13 Password List

The **Password List** object can be used to display a list of passwords for the specific authority level that has been configured in the Password screen, and can also be edited in real-time. Only passwords for the current and lower level(s) relative the authority level of the current user will be displayed. Passwords for higher level(s) will not be shown in the list. See Chapter 8 Basic Object Usage for details of how to create an object.



After creating a **Password List** object, the **Property View** panel for the object will be automatically displayed allowing the various attributes of the object to be configured. The following is a description of how to configure the properties for the **Password List** object.



- General Properties

  The **General Properties** dialog box is used to configure the display style and format for the **Password List** object.

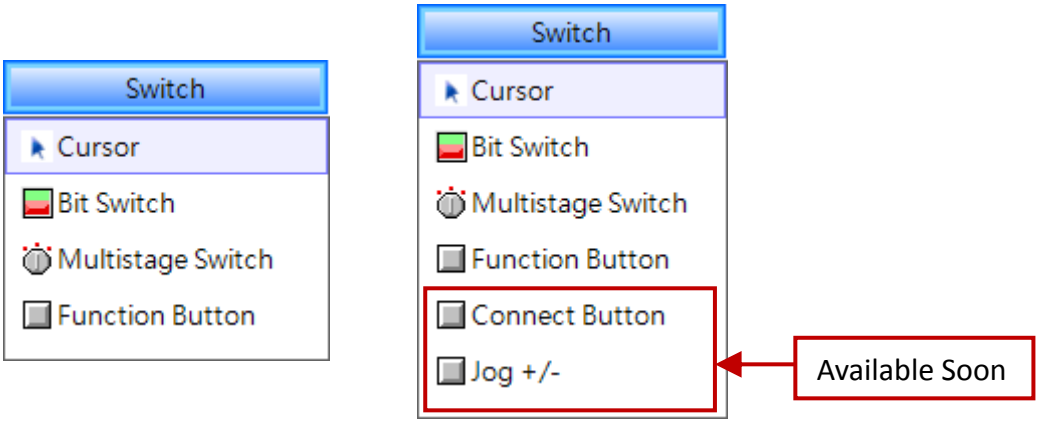
The following is an overview of the options available in the **General Properties** section of the Property View for the **Password List** object.

Column Width (Pixels)	**Group Index**	Used to specify the width of the Index column (See Section 11.2 Password for the “Group Index”)
	Password Width	Used to specify the width of the Password column
	Authority Width	Used to specify the width of the Rule column
	Font	Used to specify the type and size of the font to be used for the text.
Text	**Text Color**	Used to specify the color to be used for the text
	Background Color	Used to specify the color of the background to be used for table cells
Choice	**Text Color**	Used to specify the color of the text when the table cell has been selected
	Background Color	Used to specify the color of the background for the table cell that has been selected

- See Section 8.5 Common Attributes for details about the others attributes of the object. Border (1), Glisten, Offset, Conditional Display, Position and Size, Keyboard, and Other.

8.3 Switch Objects

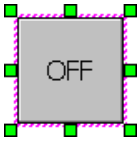
The **Switch** category of the Toolbox includes a variety of objects that can be used to add switch functionality for single action switches, such as a light switch, multistage switches, or Jog switches. These switches can then be used to perform increment/decrement operations, such as on/off, in/out, previous/next, or up/down, etc., each of which are described in more detail in the following sections.



The **Function Button** object is provided in each of the **Switch, Recipe, Alarm** and **Sampling** categories and can be used to execute up to 16 functions, such as recipe transmission, changing screens, or incrementing and decrementing variable values, etc. When the button is clicked, the functions that have been defined will be executed in sequence. The functions that can be defined for the **Switch Function Button** object include:

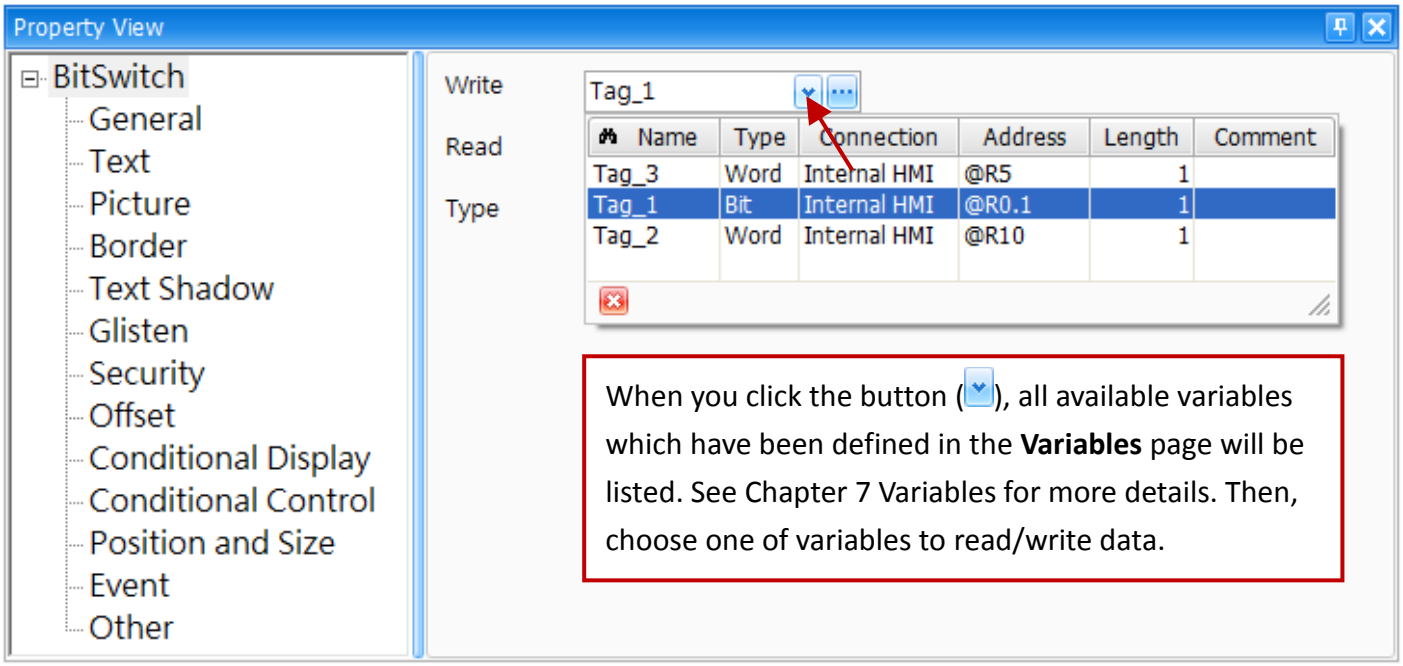
Set ON/Set OFF	Momentary/Toggle	Change Screens	Change Language
Complex State	Jog+/-	Login/Logout	Show Calculator
Show /Hide Window	Write Constant/Set Value	Return to the Previous Screen	

8.3.1 Bit Switch (Set ON/OFF)



The **Bit Switch** object can be used to control the state of a bit variable (ON/OFF), switching between two states, where OFF is 0 and ON is 1. The **Bit Switch** object allows four action types, including Set ON, Set OFF, Momentary, and Toggle , each of which are described in more detail in the following sections.

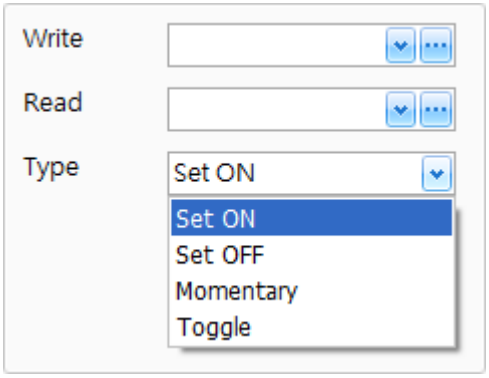
See Chapter 8 Basic Object Usage for details of how to create an object. After creating a **Bit Switch** object, the **Property View** panel for the object will be automatically displayed allowing the various attributes of the object to be configured. The following is a description of how to configure the properties for the **Bit Switch** object.



- General Properties

  The **General Properties** dialog box is used to configure the variable to be used for the **Bit Switch** object, as well as the switch type.

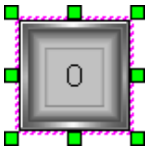
  The following is an overview of the options available in the **General Properties** section of the Property View for the **Bit Switch** object.



Write	Used to specify the Write variable to be used. For more details related to the usage of variables, see Chapter 7 Variables.
Read	Used to specify the Read variable to be used. For more details related to the usage of variables, see Chapter 7 Variables.
Type	Used to specify the action performed by the switch, where: 1. **Set ON**: Clicking the switch will set the variable to ON. 2. **Set OFF**: Clicking the switch will set the variable to OFF. 3. **Momentary**: Clicking and holding the switch will set the variable to ON, but releasing it will set the variable to OFF. 4. **Toggle**: Clicking the switch will alternate the variable between ON and OFF

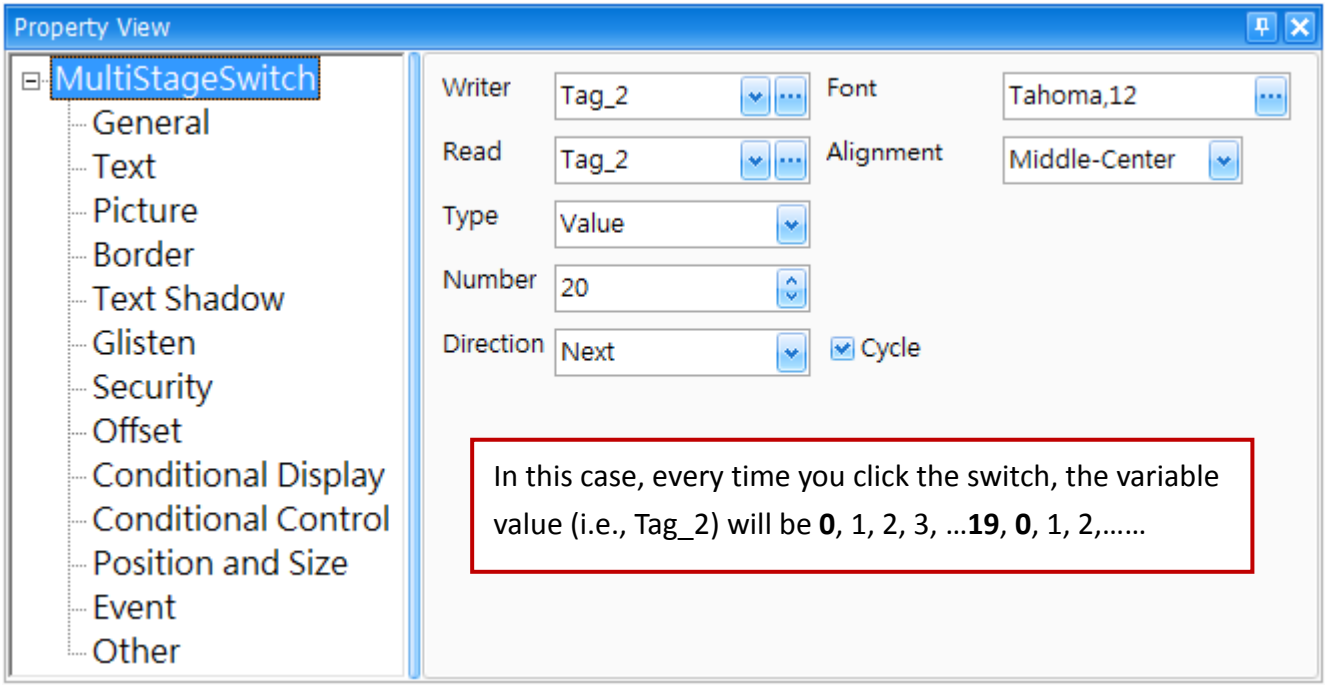
- See Section 8.5 Common Attributes for details about the others attributes of the **Bit Switch** object. Text, Image, Border (2), Text Shadow, Glisten, Security, Offset, Conditional Display, Conditional Control, Position and Size, Event, and Other.

8.3.2 Multistage Switch



The **Multistage Switch** can be used for the multistage sequence control. The Multistage Switch object provides multiple state properties compared to the Bit Switch, and can be used to configure the status for the different speeds of a motor or a multi-action mode switch.

See Chapter 8 Basic Object Usage for details of how to create an object. After creating a **Multistage Switch** object, the **Property View** panel for the object will be automatically displayed allowing the various attributes of the object to be configured. The following is a description of how to configure the properties for the **Multistage Switch** object.



- General Properties

  The **General Properties** dialog box is used to configure the variables for the **Multistage Switch** object together with the switch actions. For more details related to the usage of variables, see Chapter 7 Variables.

The following is an overview of the options available in the **General Properties** section of the Property View for the **Multistage Switch** object.

Write	Used to specify the Write variable to be used. For more details related to the usage of variables, see Chapter 7 Variables.
Read	Used to specify the Read variable to be used. For more details related to the usage of variables, see Chapter 7 Variables.

Type

Bit Index:

The initial value of the variable is "0", and the variable value must be 2^n, the state value is n.

For example,

1) If the value of a BYTE (8-bit) variable is 32 (i.e., 2^5), the state value is 5.

MSB LSB

2^7=128	2^6=64	**2^5=32**	2^4=16	2^3=8	2^2=4	2^1=2	2^0=1
0	0	**1**	0	0	0	0	0

2) If the Stages is set to "3", the Direction is set to "Next", and the Cycle is checked, then the variable value will be 0 → 1 → 2 → 4 → 1 → 2 → 4 when you press the button.

Value:

The initial value of the variable is "0", and the variable value is the state value.

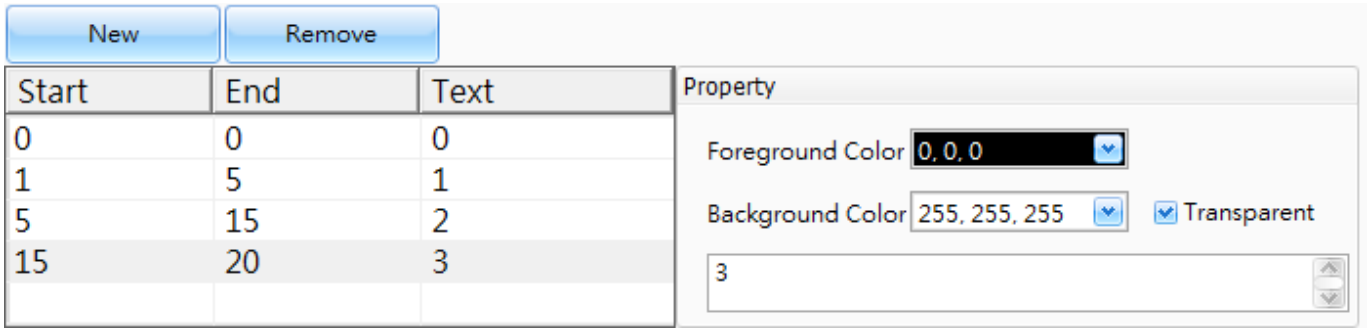
For example,

1) If the value of a BYTE (8-bit) variable is 32, the state value is 32.
2) If the **Number** is set to "**4**", the **Direction** is set to "**Before**", and the **Cycle** is checked, then the value will be 0 → 3 →2 → 1 → 0 → 3 → 2 → 1 when you press the button.

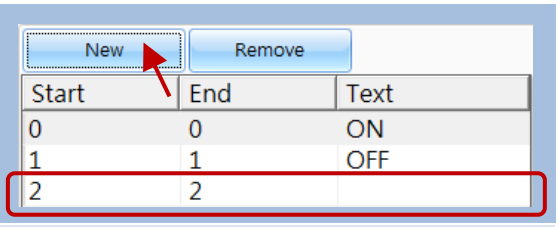
Number	Used to specify the number of stages to be used for the switch
Direction	Used to specify the sequence in which the action will be performed: **Next**: Pressing the switch will advance the action to the next stage **Before**: Pressing the switch will return the action to the previous stage Note that you can define the actions based on the variable values in the Event properties.
Cycle	Used to specify whether or not the actions are to be repeated as a continuous cycle, where: **Next**: The actions will be performed in the sequence stage1→stage2→stage3→stage1 **Before**: The actions will be performed in the sequence stage 1→stage 3→stage 2→stage 1
Font	Used to specify the type and size of the font to be used for the text.
Alignment	Used to specify the position of the text within the border of the object

- Text Properties

The **Text Properties** dialog box is used to configure the color and format of the text to be used for the **Multistage Switch** object.

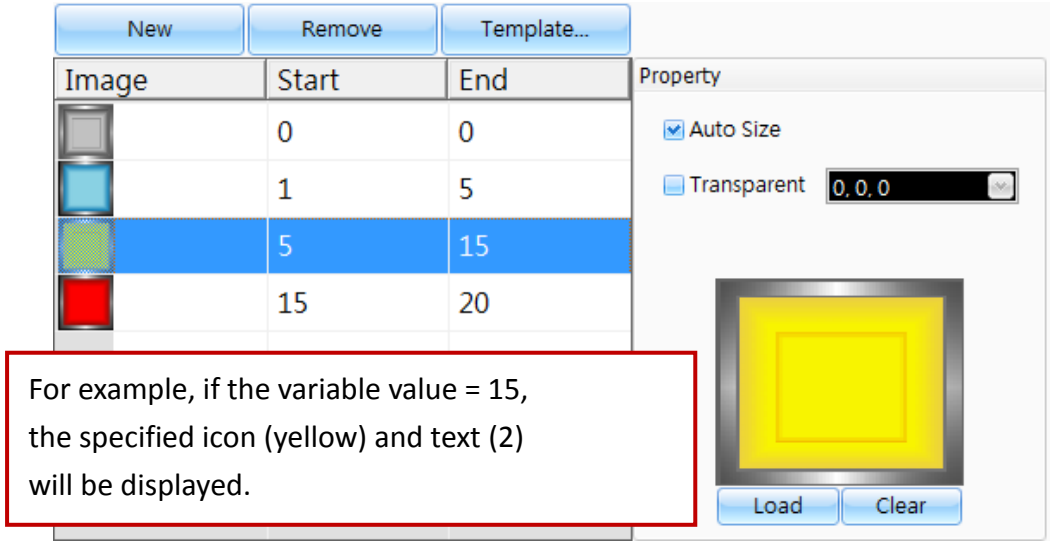


The following is an overview of the options available in the **Text Properties** section of the Property View for the **Multistage Switch** object.

New	Used to add a new text item, and then you can double click the Start, End or Text field to configure values or the text 
Remove	Used to remove the selected text item(s)
Foreground Color	Used to specify the color to be used for the text
Background Color	Used to specify the color to be used for the background of the object
Transparent	Used to specify whether or not the background color will be shown as transparent when displayed on the screen

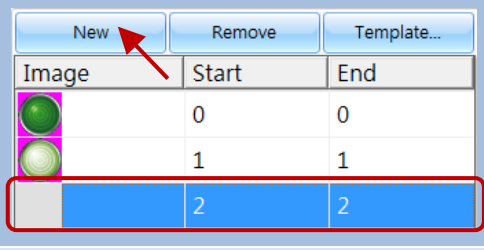
- Picture Properties

The **Picture Properties** dialog box is used to configure the images used by the **Multistage Switch** object to represent the conditions at different stages.

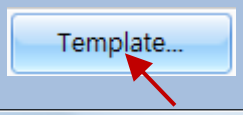


The following is an overview of the options available in the **Picture Properties** section of the Property View for the **Multistage Switch** object.

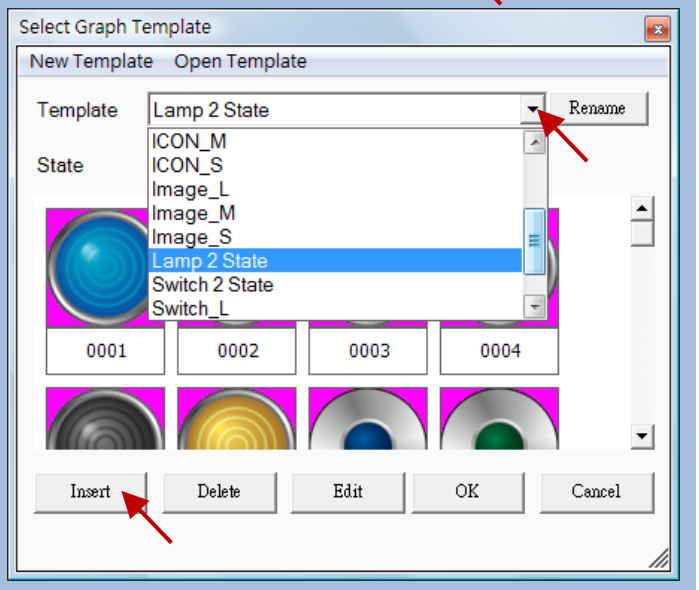
New	Used to add a new image item, and then you can double click the Image, Start, or End field to configure an image or values
Remove	Used to remove the selected image item(s)
Template	Used to add/open a template, or add a state image in the template.



<u>To create a user-defined state image</u>

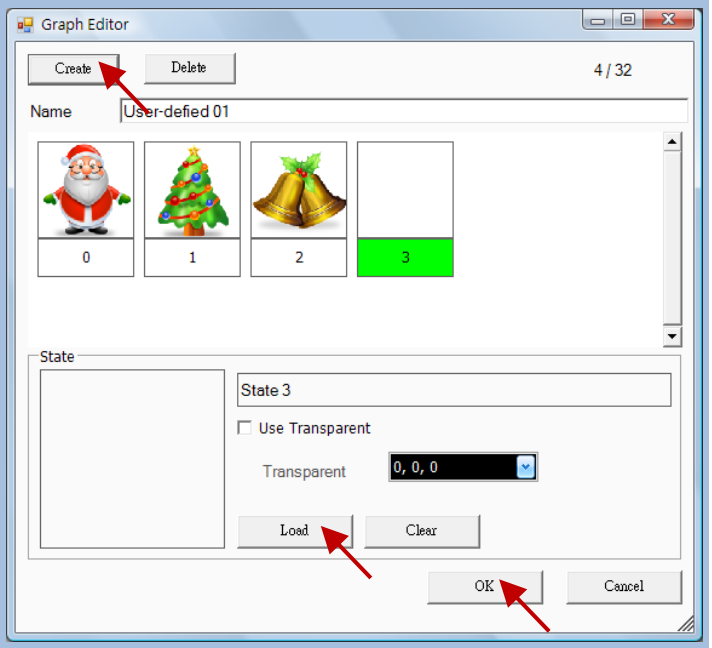


1. Click the **Template** button to go to the **Select Graph Template** dialog box.



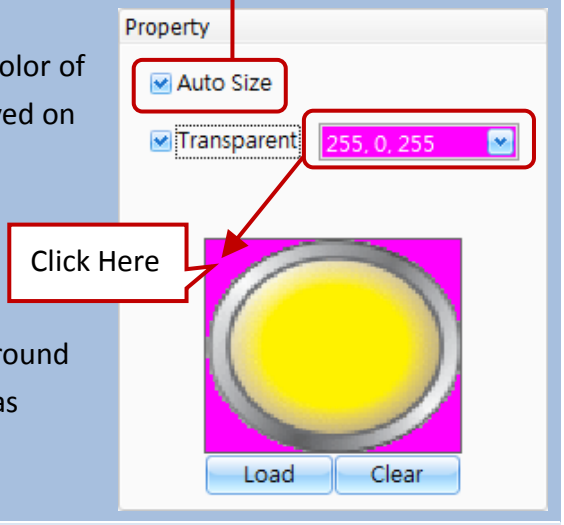
2. Select one of image types from the **Template** drop-down menu.

3. Click the **Insert** button to go to the **Graph Edit** dialog box. You can enter a name for this state image.



4. Click the **Create** button to add a State item, and then click the **Load** button to load an image.

5. Click the **OK** button to save the settings.

Auto Size	Used to specify whether or not the image should be automatically stretched to fit the size of the object 
Transparent	Used to specify whether or not the background color of image will be shown as transparent when displayed on the screen. Note that only when the set color and the background color of image are the same, it will be displayed as transparent 
Load	Used to select an image to be used from the **Image Manager**. See Section 11.4 “Image Manager” for more details.
Clear	Used to clear the currently loaded image

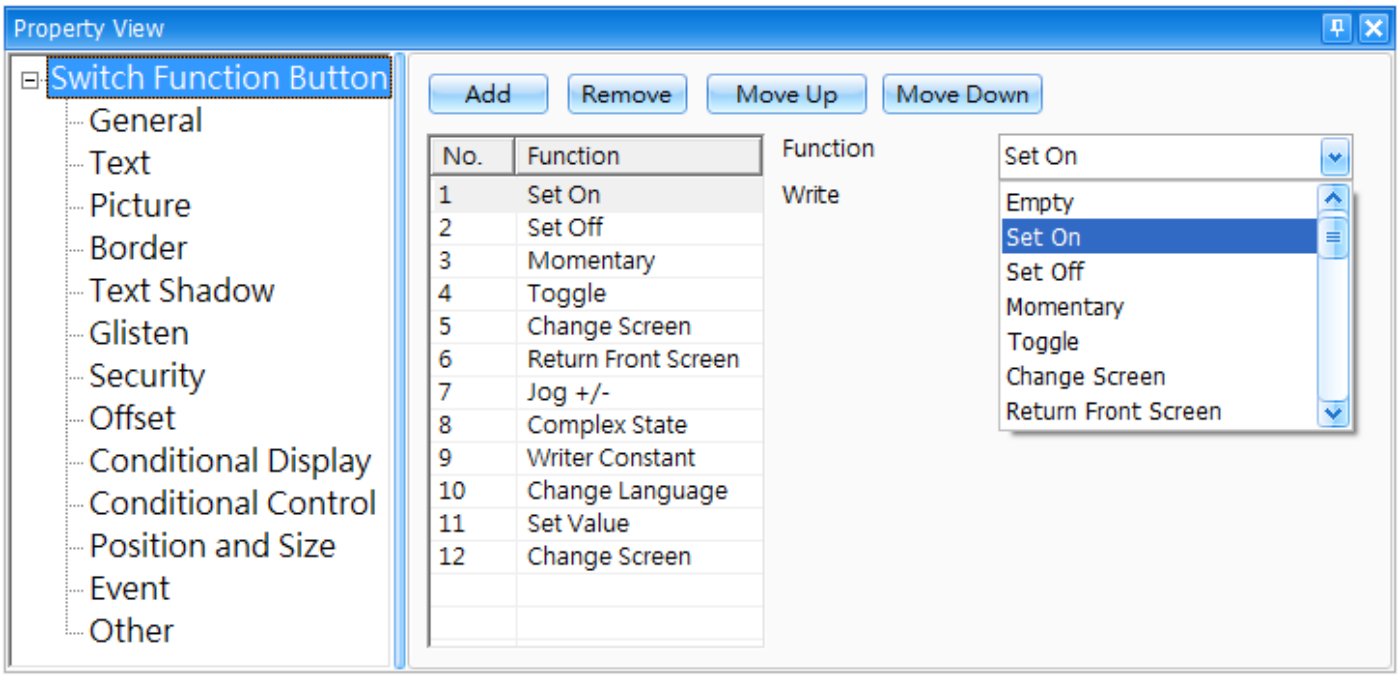
- See Section 8.5 Common Attributes for details about the others attributes of the **Multistage Switch** object. Border (2), Text Shadow, Glisten, Offset, Conditional Display, Conditional Control, Position and Size, Event, and Other.

8.3.3 Switch Function Button

The Function Button can be used to execute a wide range of functions, such as transmitting recipes, changing screens, or incrementing and decrementing values, etc. When a button is pressed, any functions that have been defined will be executed sequentially, and a maximum of 16 functions can be implemented. The **Function Button** object is available in the each of the **Switch, Recipe, Alarm**, and **Sampling** categories.

The **Switch Function Button** object provides the ability to set a variable to either ON or OFF, set a switch action to momentary or toggle, change screens, increment or decrement values, facilitate complex state switching, read and write constant values or set a specific value, change languages, log in or out from the system, show or hide specific windows, and display other applications such as a calculator, etc.

See Chapter 8 Basic Object Usage for details of how to create an object. After creating a **Switch Function Button** object, the **Property View** panel for the object will be automatically displayed allowing the various attributes of the object to be configured. The following is a description of how to configure the properties for the **Switch Function Button** object.



- General Properties

  The **General Properties** dialog box is used to configure the actions that can be performed by the **Switch Function Button** object.

The following is an overview of the options available in the **General Properties** section of the Property View for the **Switch Function Button** object.

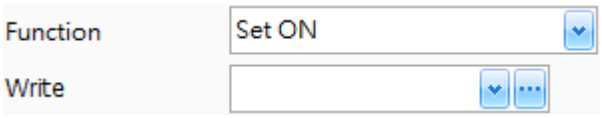
Add	Used to add a function. The default option is the “Change Screen” function A maximum of up to 16 functions can be used for one button
Remove	Used to remove a selected function
Move Up	Used to move the selected item up
Move Down	Used to move the selected item down
Function	Used to select the function from the Function drop-down menu

Introduction of Functions:

The following is an overview of the functions that can be used in conjunction with the **Switch Function Button** object.

■ Set ON

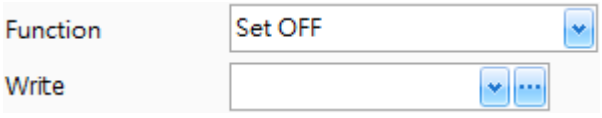
The **Set ON** function is used to set the status of a specified variable to ON when the button is clicked. For more details related to the usage of variables, see Chapter 7 Variables.



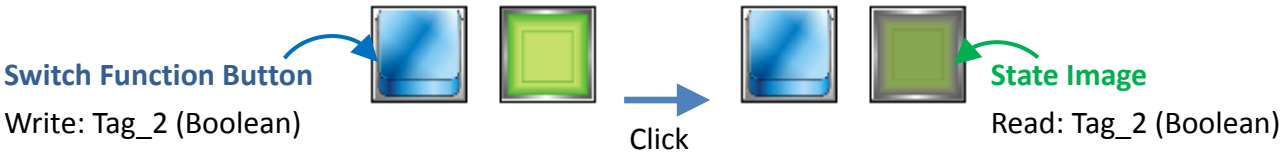
To use this function, select the **Set ON** option from the Function drop-down menu and then select the required variable from the drop-down menu, or click the **Browse** (…) button to add/edit/clear a variable.

■ Set OFF

The **Set OFF** function is used to set the status of a specified variable to OFF when the button is clicked. For more details related to the usage of variables, see Chapter 7 Variables.

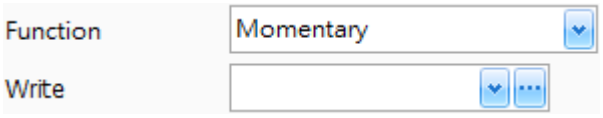


To use this function, select the **Set OFF** option from the Function drop-down menu and then select the required variable from the drop-down menu, or click the **Browse** (…) button to add/edit/clear a variable.

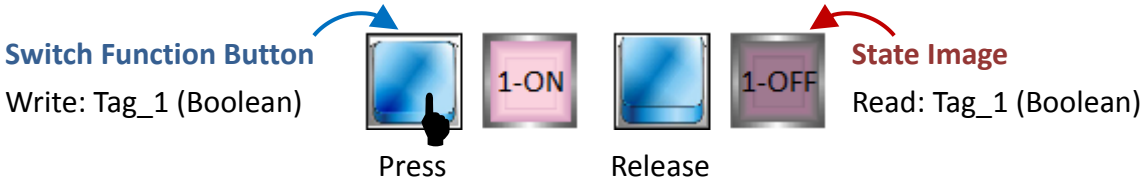


■ Momentary

The **Momentary** function is used to set the button action to Momentary so that the status of the specified variable will be set to ON when the button is clicked, but will be set to OFF when the button is released. For more details related to the usage of variables, see Chapter 7 Variables.



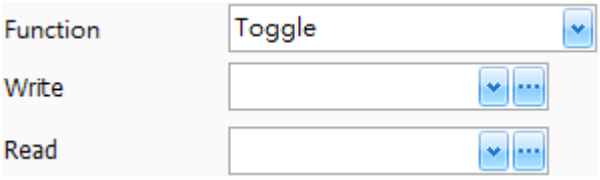
To use this function, select the **Momentary** option from the Function drop-down menu, and then select the required variable from the drop-down menu, or click the **Browse** (…) button to add/edit/clear a variable.



■ Toggle

The **Toggle** function is used to set the button action to Toggle so that when the button is clicked, the Read value will be inverted (e.g., ON to OFF) and assigned to the specified Write variable.

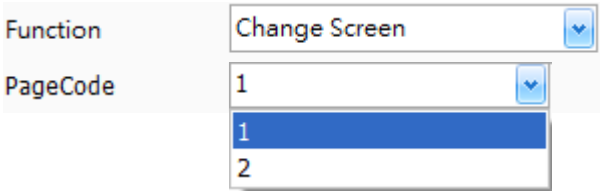
For more details related to the usage of variables, see Chapter 7 Variables.



To use this function, select the **Toggle** option from the Function drop-down menu, and then select the required variable from the drop-down menu, or click the **Browse** (…) button to add/edit/clear a variable.

■ Change Screen

The **Change Screen** function is used to transfer to the assigned screen.



To use this function, select the **Change Screen** option from the Function drop-down menu, and then select the required **Screen no.** from the PageCode drop-down menu. (See Section 4.9 Screen View)

■ Return Front Screen

The **Return Front Screen** function is used to return to the previous screen.

To use this function, select the **Return Front Screen** option from the Function drop-down menu.

■ Jog +/- (Available soon)

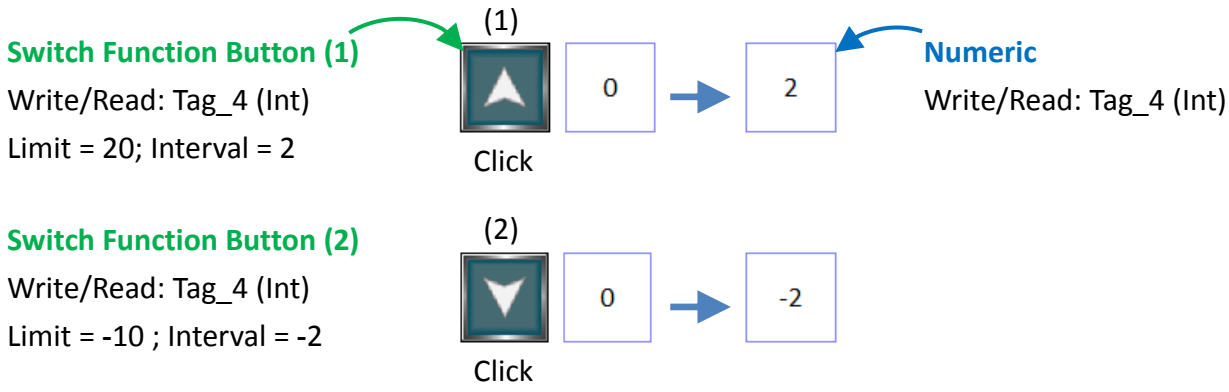
The **Jog +/-** function is used to progressively increment or decrement the value of a variable based on a specific increment interval, and write the new value to a specified variable.

The function provides the same functionality as for the Jog Button. For more details related to the usage of variables, see Chapter 7 Variables.

The following is an overview of the options available for the **Jog +/-** function.

Write	Used to specify the variable to be written. For more details related to the usage of variables, see Chapter 7 Variables.
Read	Used to specify the variable to be read. For more details related to the usage of variables, see Chapter 7 Variables.
Limit	Used to specify the maximum or minimum values for the increment
Interval	Used to specify the increment interval value

To use this function, select the **Jog +/-** option from the Function drop-down menu, select the required **Read** and **Write** variables from the respective drop-down menus, or click the **Browse** (…) button to add/edit/clear a variable, and then enter a value for both the **Limit** and **Interval** options.

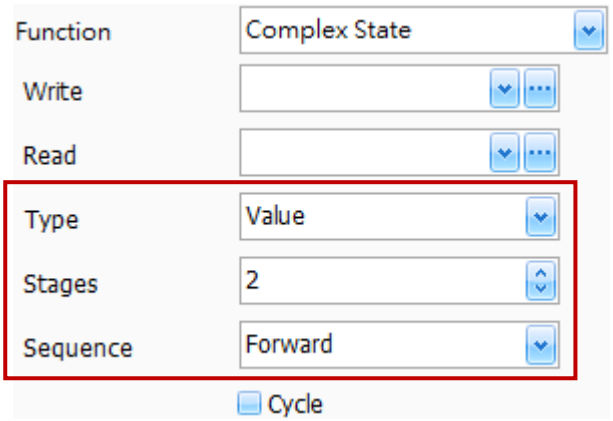


■ Complex State

The **Complex State** function can be used for the multistage sequence control, for example, to configure the different speeds of a motor or a multi-action mode switch.

The user can assign the variable type (i.e., **Bit Index** or **Value**) and the number of stages for multistage control. The variable value starts at 0, and will be changed according to the Sequence setting when the button is clicked.

This function provides the same functionality as for a multistage switch. For more details related to the usage of variables, see Chapter 7 Variables.



The following is an overview of the options available for the **Complex State** function.

Write	Used to specify the Write variable to be used.
Read	Used to specify the Read variable to be used.
Type	**Bit Index:** The initial value of the variable is "0", and the variable value must be 2^n, the state value is n. For example, if the value of a BYTE (8-bit) variable is 32 (i.e., 2^5), the state value is 5.

MSB LSB

2^7=128	2^6=64	**2^5=32**	2^4=16	2^3=8	2^2=4	2^1=2	2^0=1
0	0	**1**	0	0	0	0	0

1) If the **Stages** is set to "**3**", the **Sequence** is set to "**Forward**", and the **Cycle** is checked, then
the variable value will be 0 > **1** > **2** > **4** > 1 > 2 > 1 when you press the button.
2) If the **Stages** is set to "**3**", the **Sequence** is set to "**Reverse**", and the **Cycle** is checked, then the variable value will be 0 > **4** > **2** > **1** > 4 > 2 > 1 when you press the button.

Value:

The initial value of the variable is "0", and the variable value is the state value.
For example, if the value of a BYTE (8-bit) variable is 32, the state value is 32.

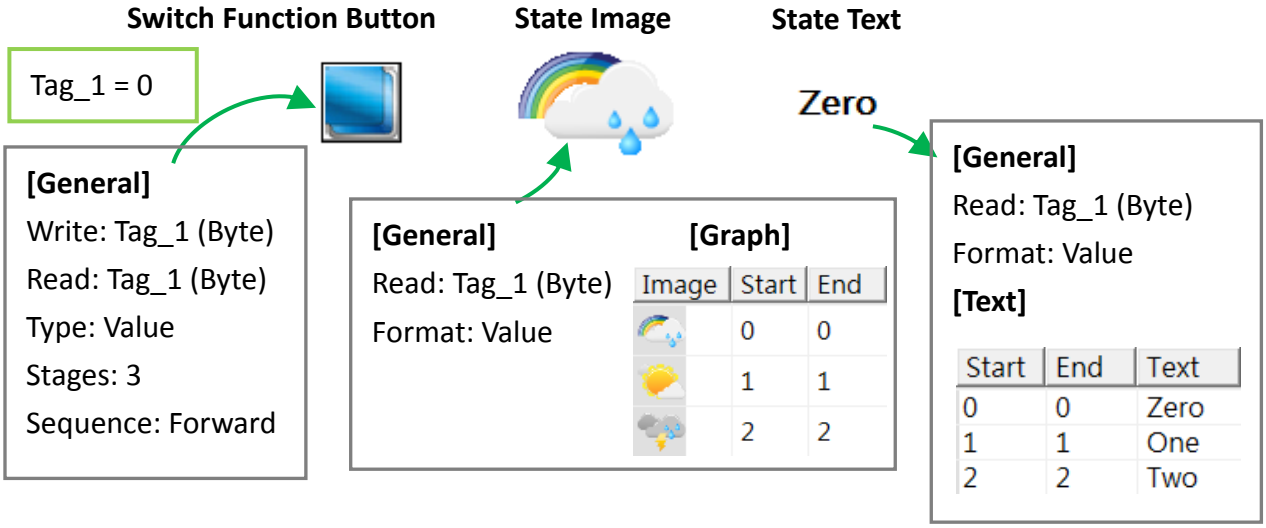
1) If the **Stages** is set to "**3**", and the **Sequence** is set to "**Forward**", and the **Cycle** is checked, then the variable value will be **0** > **1** > **2** > 0 > 1 > 2 when you press the button.
2) If the **Stages** is set to "**3**", the **Sequence** is set to "**Reverse**", and the **Cycle** is checked, then the variable value will be **0** > **2** > **1** > 0 > 2 > 1 when you press the button.

Stages	Used to specify the number of stages to be used for the switch.
Sequence	Used to specify the sequence in which the action will be performed: **Forward**: Pressing the button will advance the action to the next stage **Reverse**: Pressing the button will return the action to the previous stage Note that you can define the actions based on the variable values in the Event properties.
Cycle	Used to specify whether or not the actions are to be repeated as a continuous cycle If the Stages = "4", the Type = "value", the actions will be performed in these sequence **Forward: 0** > **1** > **2** > **3** > 0 > 1 (click the Switch Function Button 5 times) **Reverse: 0** > **3** > **2** > **1** > 0 > 3 (click the Switch Function Button 5 times)

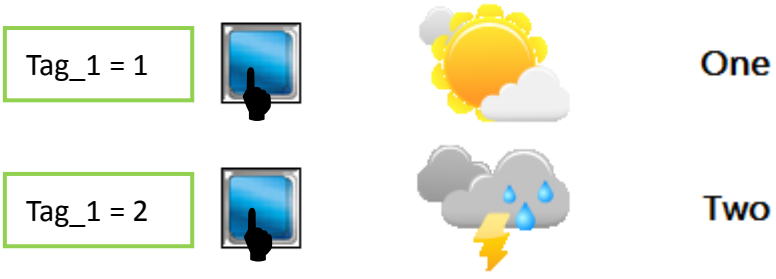
To use this function, select the **Complex State** option from the Function drop-down menu, select the required **Read** and **Write** variables from the respective drop-down menus, or click the respective **Browse** (…) button to add/edit/clear a variable. Select a **Type** from the Type drop-down menu, and then enter a value for the **Stage** option and specify a **Sequence** from the Sequence drop-down menu. To specify whether the function is to be repeated as a continuous cycle, check the **Cycle** checkbox.

The following will take a simple example. The user can use this function according to the application needs.

1) dd and configure three objects on the screen as the following settings.

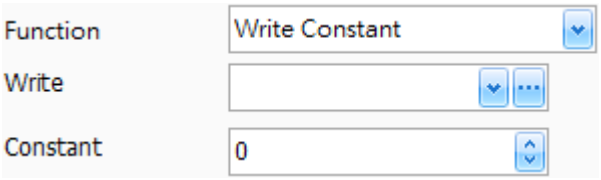


2) Click the **Simulate Offline** button ( or "F8") on the toolbar, and then click the **Function Button** to view the change.

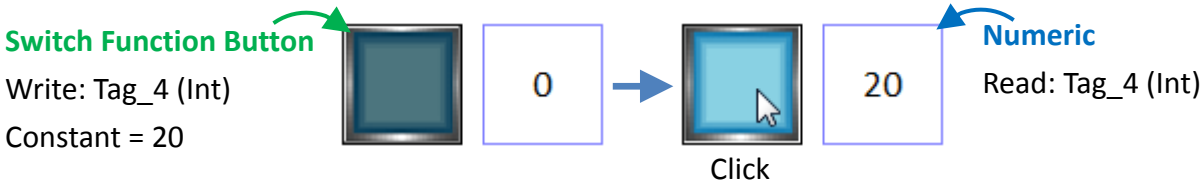


■ Write Constant

The **Write Constant** function is used to write a constant value to a specific variable. For more details related to the usage of variables, see Chapter 7 Variables.



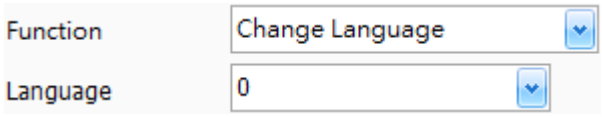
To use this function, select the **Write Constant** option from the Function drop-down menu, select the required **Write** variable from the drop-down menu, or click the **Browse** (…) button to add / edit / clear a variable, and then input a value for the **Constant** from the Constant spin-edit menu.



■ Change Language

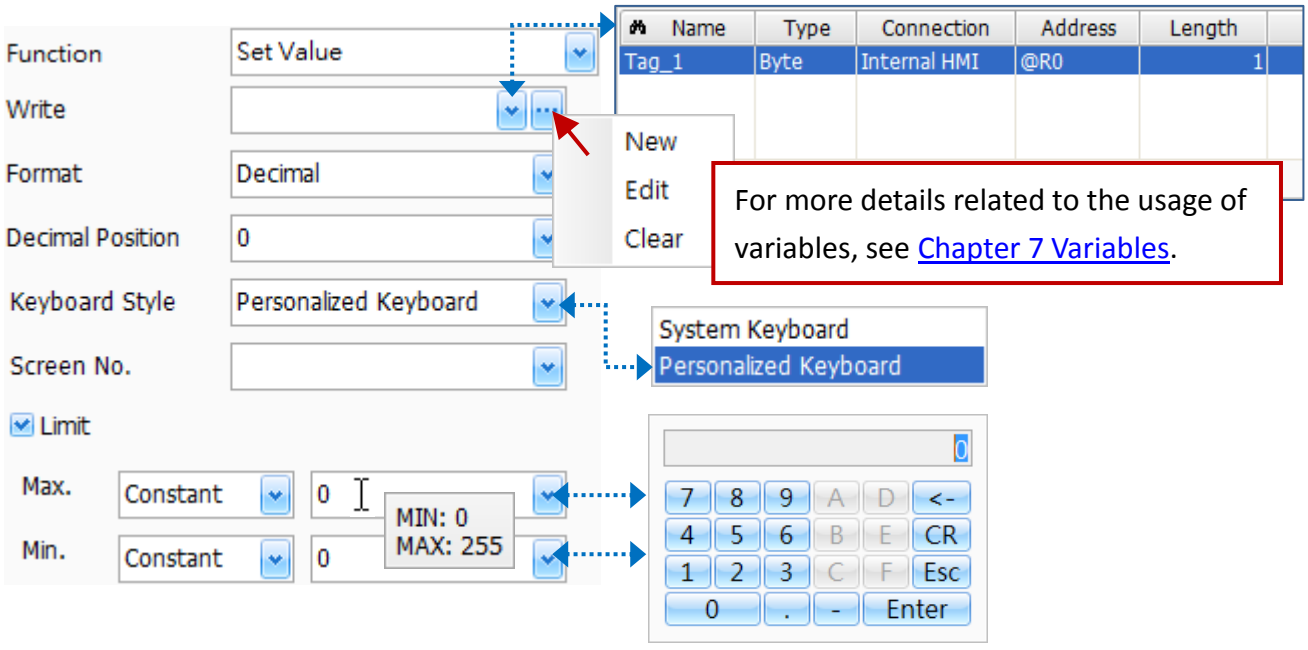
The **Change Language** function is used to change the current language settings.

To use this function, select the **Change Language** option from the Function drop-down menu, and then select the required **Language Code** from the Language drop-down menu, which can be selected from English, Traditional or Simplified Chinese. See Section 11.3 Language for more details related to the Language setting.



■ Set Value

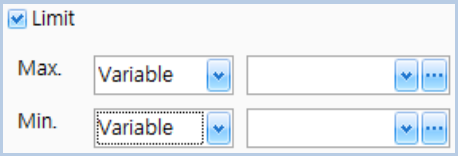
The **Set Value** function is used to write an input value to a specific variable.



The following is an overview of the options available for the **Set Value** function.

Write	Used to specify the variable to be written
Format	Used to specify the value format and can be set to either Decimal or Hexadecimal
Decimal Position	Used to specify position of the decimal point (read from the right). E.g., if the decimal position is set to "1", the input value is "100", and then "10.0" will be written to the specified variable
Keyboard Style	Used to specify the keyboard style, which can be configured as either the system keyboard or a customized keyboard (see Section 8.4 Keyboard Objects)
Screen No.	Used to specify the page number of the Window screen for the customized keyboard, see Section 8.4 Keyboard Objects for more details
Limit	Used to specify whether or not a maximum and minimum value should be used
Max.	Used to specify the maximum limit value based on a specified constant or variable
Min.	Used to specify the minimum limit value based on a specified constant or variable

Note: If the write variable is **Tag_1** (**Byte**) and the Max./Min. is a Constant, the valid range (**0 ~ 255**) will be shown when the user moves the cursor over the drop-down menu (see the figure above). If the Max./Min. is a Variable, the related drop-down menu will be appeared.



To use this function, select the **Set Value** option from the Function drop-down menu, select the required **Write** variable from the Write drop-down menu, or click the **Browse** (…) button to add/ edit/clear a variable, and then select a **Format, Decimal Position, Keyboard Style** and **Screen No.** (for the customized keyboard) from the respective drop-down menus. To specify whether a maximum and minimum value should be used, check the **Limit** checkbox, select the Constant or Variable for writing value, and then enter the maximum and minimum values or specify a variable from the respective drop-down menus.

■ Login

The **Login** function is used to configure the login authority level for a user.

■ Logout

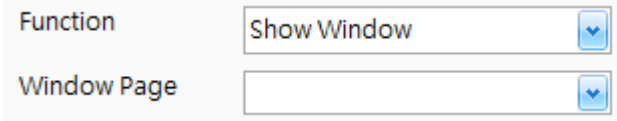
The **Logout** function is used to configure the logout authority level for a user.

See **Section 11.2 Password** to configure the password for each authority level. After clicking the **Login** button on the HMI screen, you can have access permissions depends on the input password. "9" is the highest level which allows the user to operate all objects on the screen. After completing operations, click the **Logout** button to back to the operating status with level 0.

To use this function, select the **Login/Logout** option from the Function drop-down menu.

■ Show Window

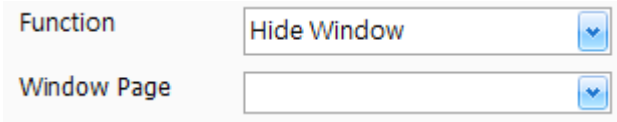
The **Show Window** function is used to open a customized keyboard on the screen.



To use this function, select the **Show Window** option from the Function drop-down menu, and then select the required page number of the Window screen from the Window Page drop-down menu.

■ Hide Window

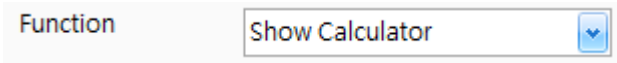
The **Hide Window** function is used to close a customized keyboard on the screen.



To use this function, select the **Hide Window** option from the Function drop-down menu, and then select the required page number of the Window screen from the Window Page drop-down menu.

■ Show Calculator

The **Show Calculator** function is used to display the embedded calculator tool.



To use this function, select the **Show Calculator** option from the Function drop-down menu.

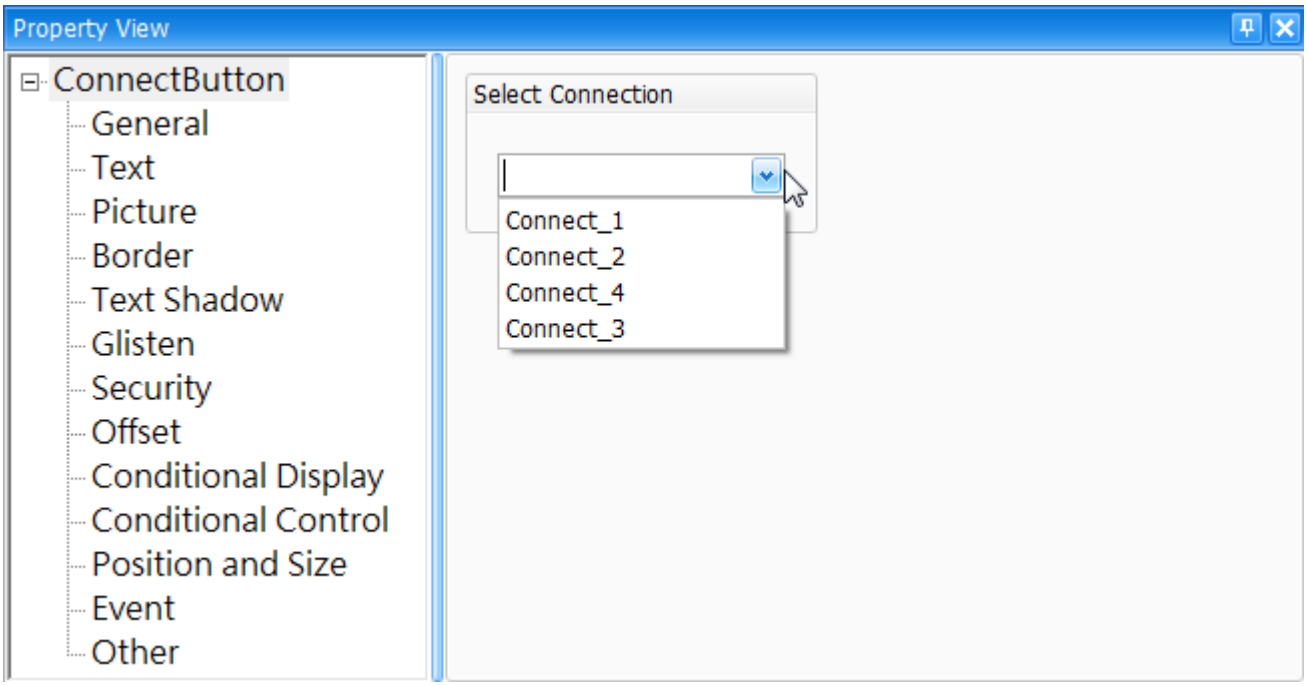
The user can see Section 9.2.6 for the **Recipe** function, see Section 9.6.2 for the **Alarm** function, and see Section 9.8.3 for the **Sampling** function.

- See Section 8.5 Common Attributes for details about the others attributes of the **Switch Function Button** object. Text, Image, Border (2), Text Shadow, Glisten, Security, Offset, Conditional Display, Conditional Control, Position and Size, Event, and Other.

8.3.4 Connect Button (Available soon)

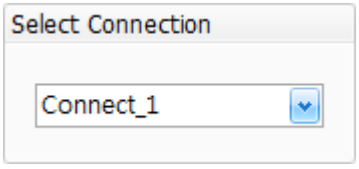
The **Connect Button** object can be used to control whether or not the specified connection is set to either Online or Offline. See Chapter 8 Basic Object Usage for details of how to create an object.

After creating a **Connect Button** object, the **Property View** panel for the object will be automatically displayed allowing the various attributes of the object to be configured. The following is a description of how to configure the properties for the **Connect Button** object.



- General Properties

  The **General Properties** dialog box is used to select which of the existing connections is to be controlled. See Chapter 6 Connections for more details on settings.

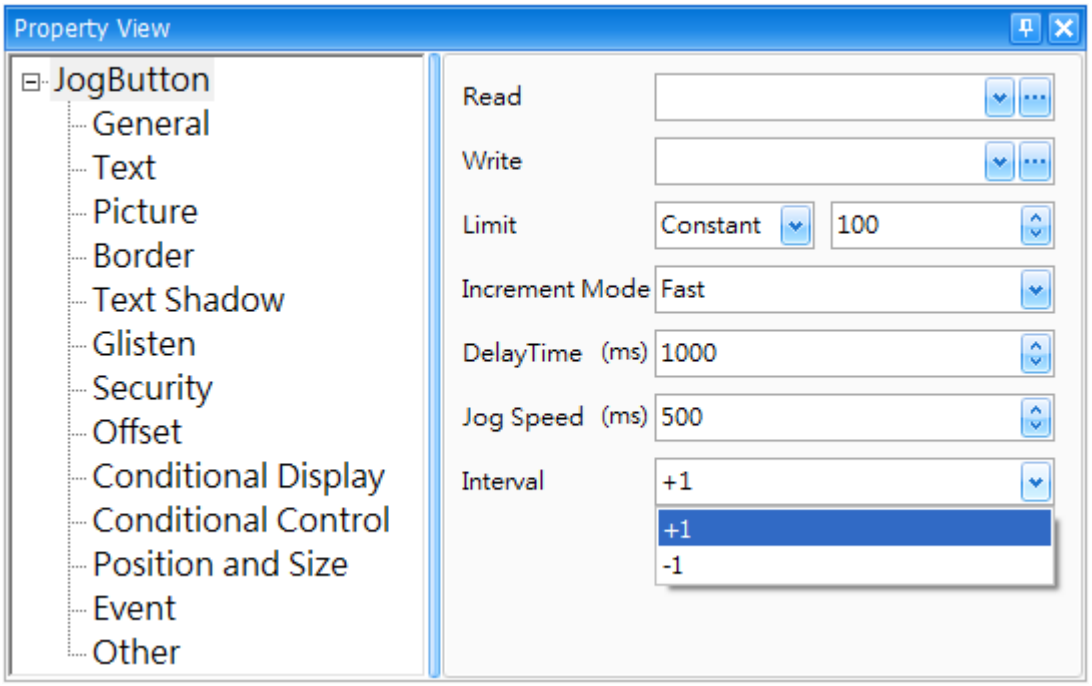


- See Section 8.5 Common Attributes for details about the others attributes of the **Connect Button** object. Text, Image, Border (2), Text Shadow, Glisten, Security, Offset, Conditional Display, Conditional Control, Position and Size, Event, and Other.

8.3.5 Jog (+ / -) Button (Available soon)

The **Jog Button** object can be used to progressively increment or decrement the value of a variable, and then write the new value to a specific variable. See Chapter 8 Basic Object Usage for details of how to create an object.

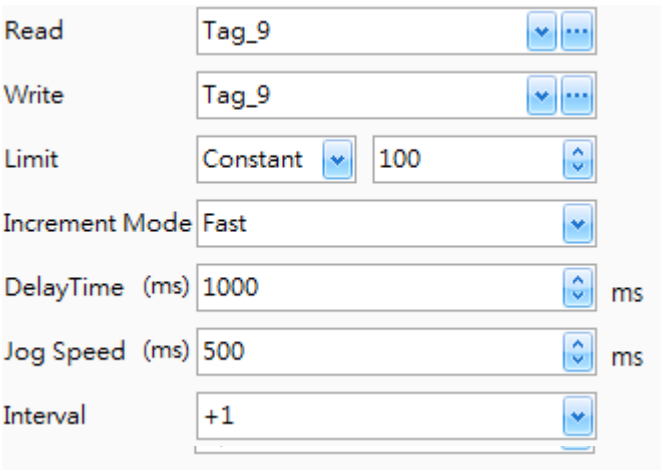
After creating a **Connect Button** object, the **Property View** panel for the object will be automatically displayed allowing the various attributes of the object to be configured. The following is a description of how to configure the properties for the **Connect Button** object.



- General Properties

  The **General Properties** dialog box is used to configure the attributes for the **Jog Button** object, allowing the value of a variable to be progressively incremented or decremented based on a specific increment interval, and then write the new value to a specified variable.

  For more details related to the usage of variables, see Chapter 7 Variables.



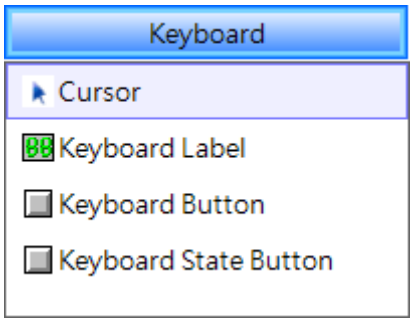
The following is an overview of the options available in the **General Properties** section of the Property View for the **Jog Button** object.

Read	Used to specify the variable to be read.
Write	Used to specify the variable to be written.
Limit	Used to specify the maximum or minimum limit based on the Constant or Variable
Increment Mode	Used to specify the Increment Mode when the button is pressed (or held down): Fast Mode: Only increments the interval value by +1 or -1 Fixed Mode: User defined interval value.
Delay Time	Used to specify how long will it starts accumulating values when the button is held down
Jog Speed	Used to specify the jog speed when the button is held down
Interval	Used to specify the increment interval value

- See Section 8.5 Common Attributes for details about the others attributes of the **Jog Button** object. Text, Image, Border (2), Text Shadow, Glisten, Security, Offset, Conditional Display, Conditional Control, Position and Size, Event, and Other.

8.4 Keyboard Objects

The **Keyboard** category of the **Toolbox** includes objects that can only be added to the Window-type screen and can be used for designing the customized keyboard that will be displayed on the HMI screen.

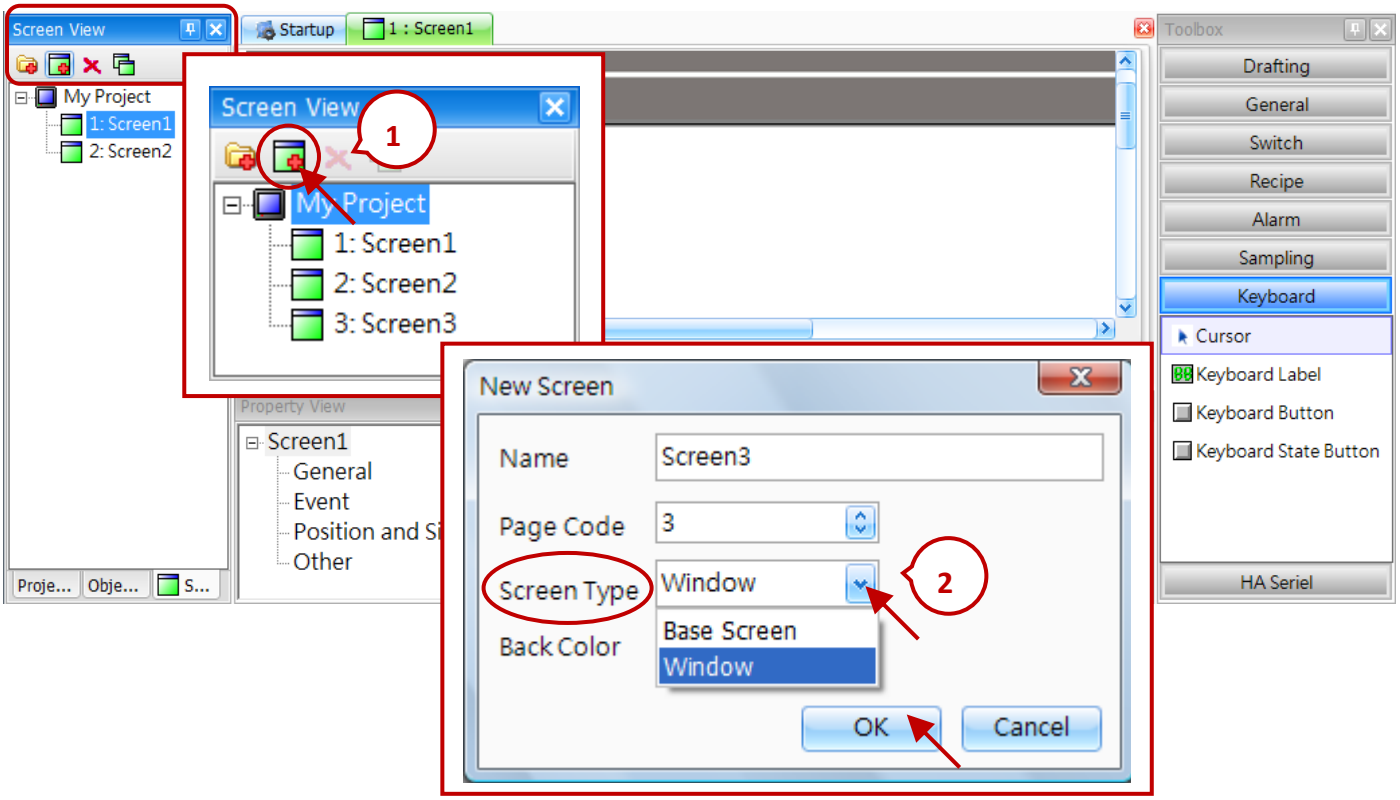


Some of objects come with the keyboard function, such as Numeric, Numeric table, Text-Box, Password List, Recipe Numeric /Text, etc. The user can choose the system keyboard or the personalized keyboard in the Keyboard Properties dialog box of these objects.

For using the customized keyboard, first the user needs to create a Window-type screen and design a layout for the on-screen keyboard. See steps below:

To create a Window screen

1. Click the **New screen** button in the **Screen View** panel to open the New Screen dialog box.
2. Choose **Window** in the **Screen Type** drop-down menu, and then click the **OK** button.

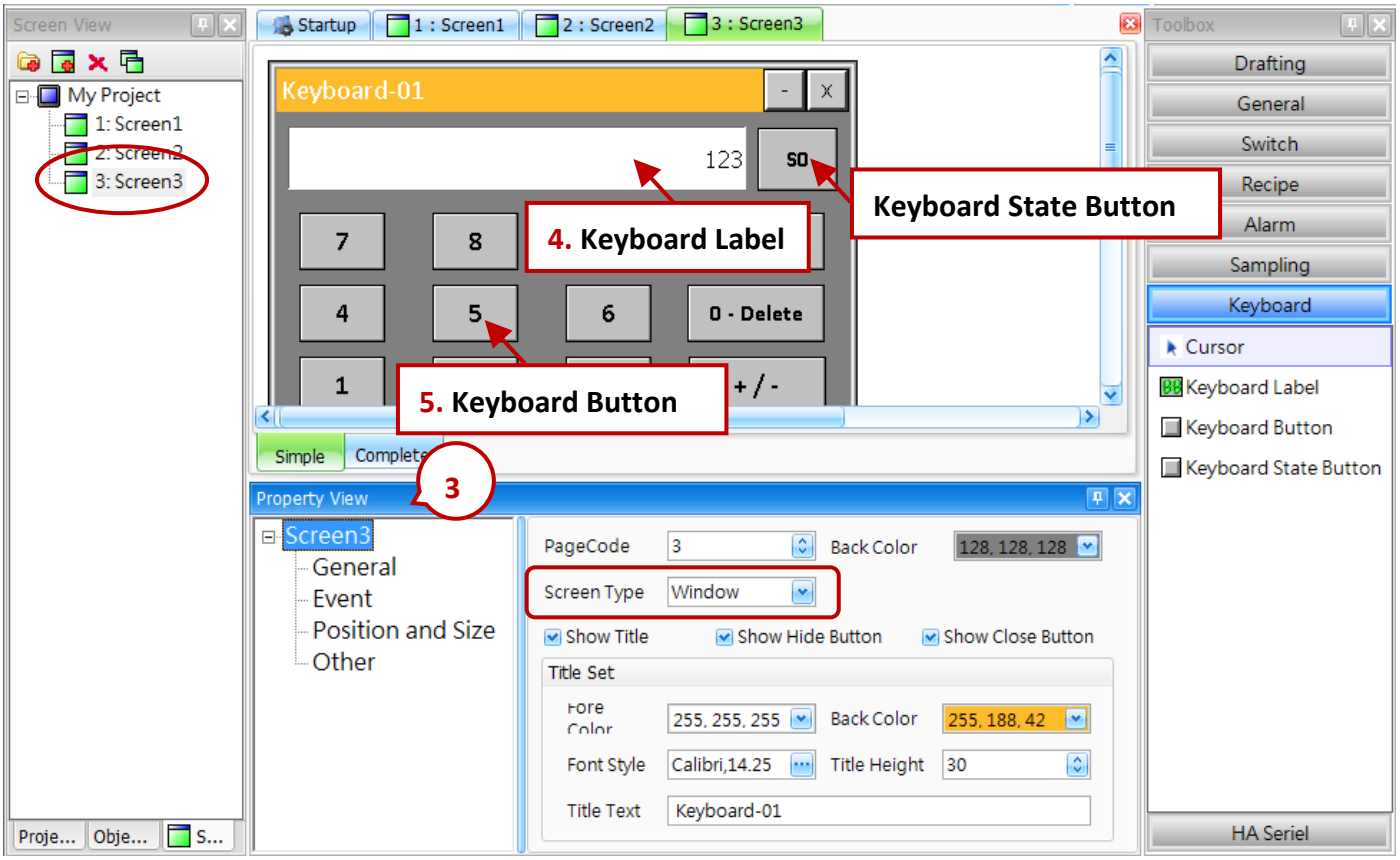


To configure a keyboard layout

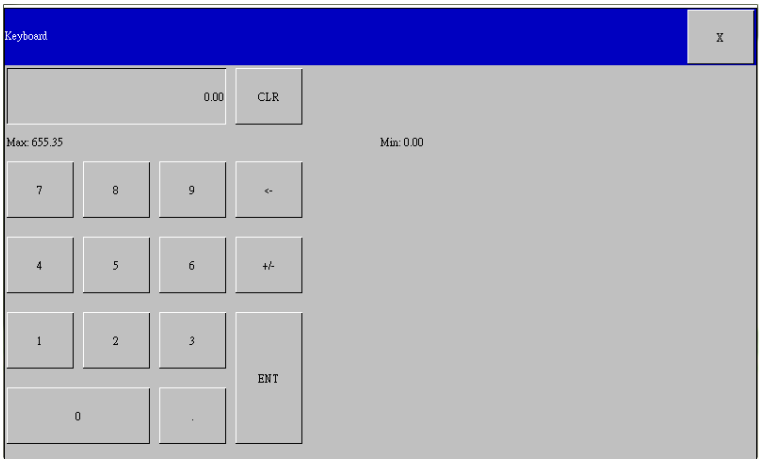
To create the Keyboard object, click the name of the object in the **Keyboard** category of the **Toolbox** to select it, and then either drag out the proper size of shape on the screen or click the desired position on the screen to place the object. To open the **Property View** for that object, click the object and then configure its properties.

3. In the Window screen, configure the appearance for the keyboard in the Property View panel.
4. Add the Keyboard Label object for the numeric input. See Section 8.4.1.
5. Add the Keyboard Button object as a numeric button. See Section 8.4.2.

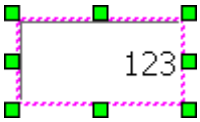
You can also add the **Keyboard State Button** object (see Section 8.4.3), if necessary. Each of these objects will be described in more detail in the following sections.



The system Keyboard:



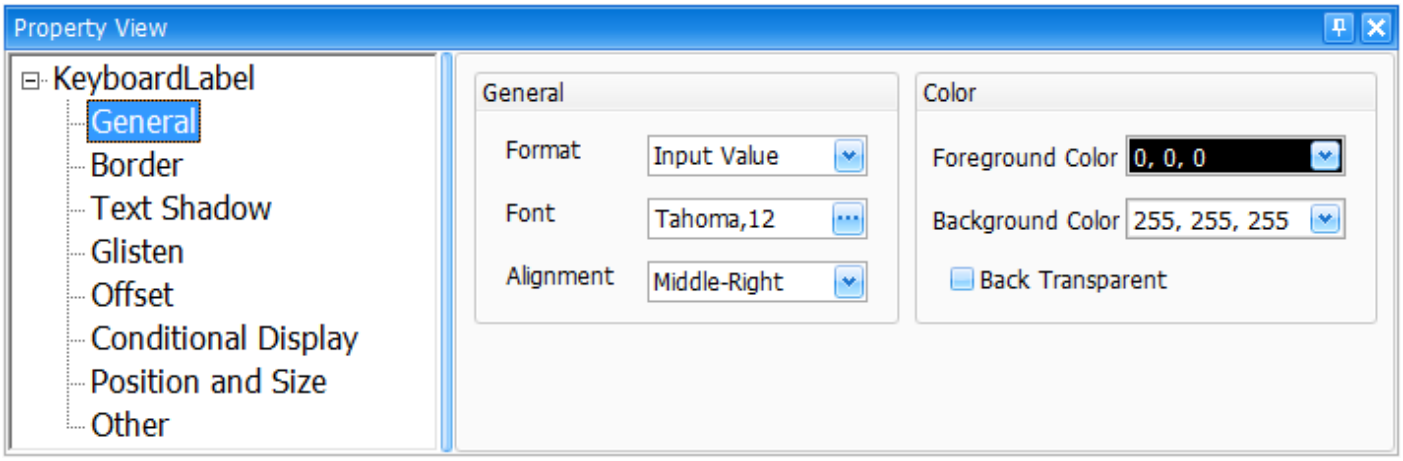
8.4.1 Keyboard Label



The **Keyboard Label** object can be used to display the current numeric value that has been entered via the user-defined keyboard. See Chapter 8 Basic Object Usage for details of how to create an object.

Note that this object can only be added to the Window-type screen, see Section 8.4 for more details.

After creating a **Keyboard Label** object, the **Property View** panel for the object will be automatically displayed allowing the various attributes of the object to be configured. The following is a description of how to configure the properties for the **Keyboard Label** object.



- General Properties

  The **General Properties** dialog box is used to configure the display format to be used for the **Keyboard Label** object, including the font, text alignment, and colors, etc.

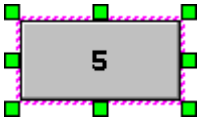
The following is an overview of the options available in the **General Properties** section of the Property View for the **Keyboard Label** object.

General	**Format**	Used to specify the display style, where **Input Value:** Displays the current input value **Minimum:** Displays the minimum value in the input range **Maximum:** Displays the maximum value in the input range Note that the value of the input range is based on the data type of the variable to be specified in the object. See 7.1.3 Variable Type for more details, and see Section 8.5 (M) Keyboard Properties to find out which objects are available.

General	**Font**	Used to specify the type and size of the font to be used for the text shown on this object
	Alignment	Used to specify the horizontal and vertical position of the text within the border of the object
Color	**Foreground Color**	Used to specify the color of the text to be used for the object
	Background Color	Used to specify the color of the background to be used for the object
	Return Front Screen	Used to specify whether or not the background color will be shown as transparent when displayed on the screen

- See Section 8.5 Common Attributes for details about the others attributes of the **Keyboard Label** object. Border (1), Text Shadow, Glisten, Offset, Conditional Display, Position and Size, and Other.

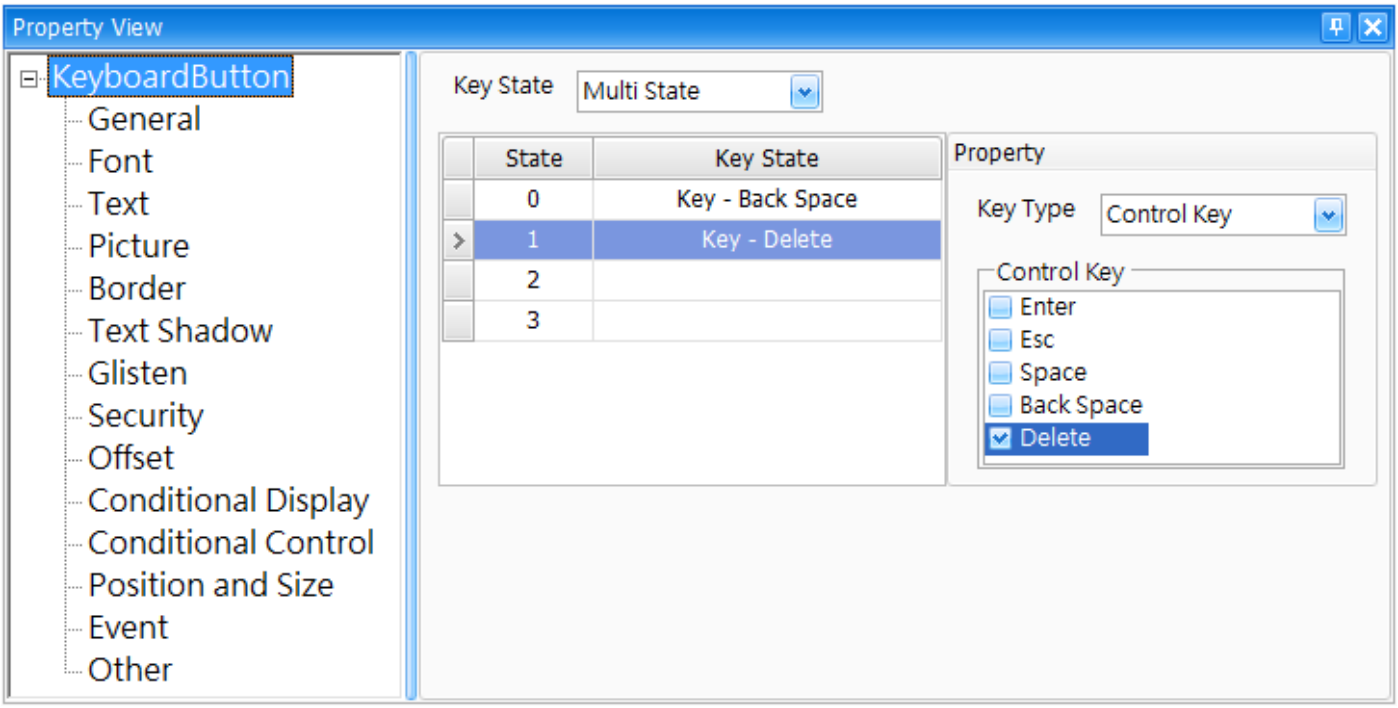
8.4.2 Keyboard Button



The **Keyboard Button** object can be used to create the numeral key or the function key on the user-defined keyboard. See Chapter 8 Basic Object Usage for details of how to create an object.

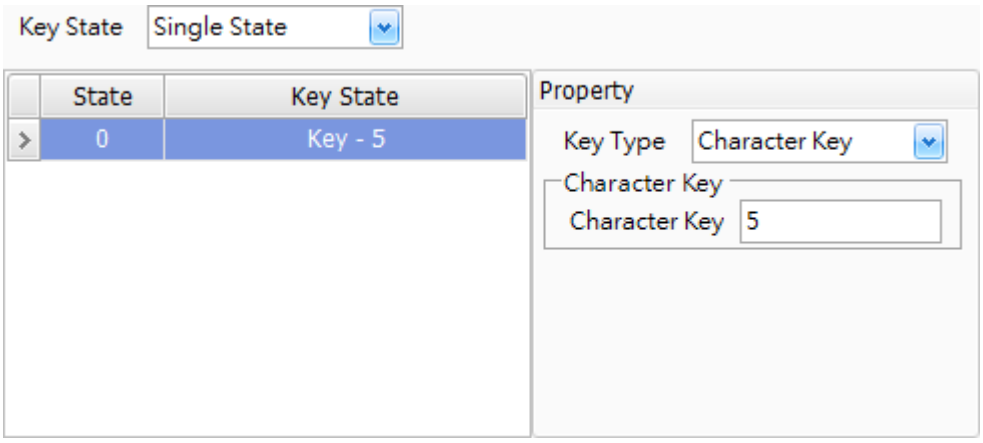
Note that this object can only be added to the Window-type screen, see Section 8.4 for more details.

After creating a **Keyboard Button** object, the **Property View** panel for the object will be automatically displayed allowing the various attributes of the object to be configured. The following is a description of how to configure the properties for the **Keyboard Button** object.



- General Properties

  The **General Properties** dialog box is used to configure the key state or type to be used for the **Keyboard Button** object.

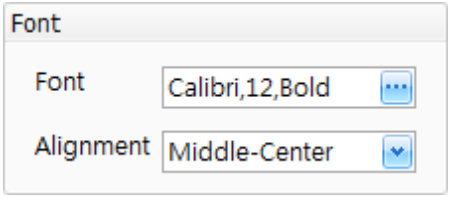


The following is an overview of the options available in the **General Properties** section of the Property View for the **Keyboard Button** object.

Key State		Used to specify the type of button, where: **1**. Single State **2**. Multiple States. A maximum of 4 functions can be assigned that can be switched by using the **Keyboard State Button** object.
State		Used to indicate the index number for the State
Key State (function)		Used to indicate the function that has been assigned
Property	**Key Type**	Used to specify the type of key, where: **1**. Control Key **2**. Character Key
	Control Key	Used to specify the function when the Key Type attribute has been set to Control Key, where: **1**. Enter **2**. Esc **3**. Space **4**. Backspace **5**. Delete
	Character Key	Used to configure the input for a single character when the Key Type attribute has been set to Character Key.

- Font Properties

  The **Font Properties** dialog box is used to configure the type and size of the font to be used for the text on the **Keyboard Button** object, as well as the alignment.



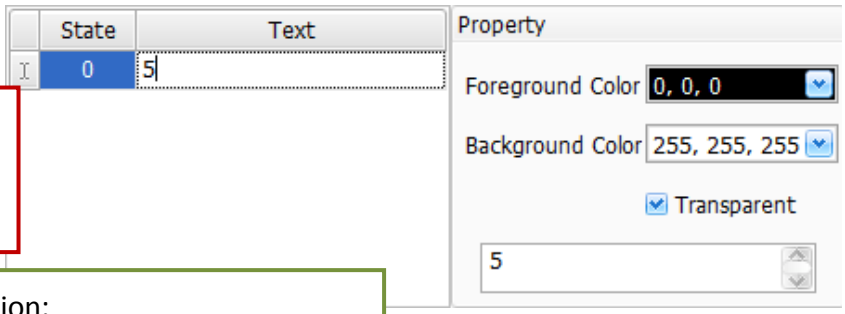
The following is an overview of the options available in the **Font Properties** section of the Property View for the **Keyboard Button** object.

Font	Used to specify the font and size of the text displayed on the key
Alignment	Used to specify the horizontal and vertical alignment of the text displayed on the key within the border of the object

- Text Properties

  The **Text Properties** dialog box is used to configure the text to be displayed for the **Keyboard Button** object**.** There are two state types - Single state and multiple states.

If you choose the **Single State** option:

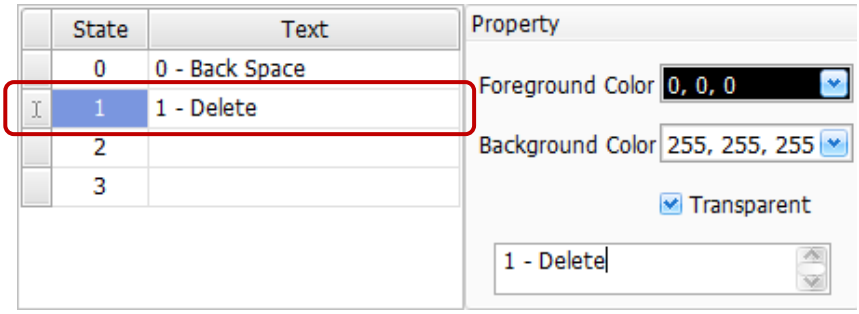


General Properties - Key State:

1. Single State

2. Multiple states (support 4 states)

If you choose the **Multiple states** option:



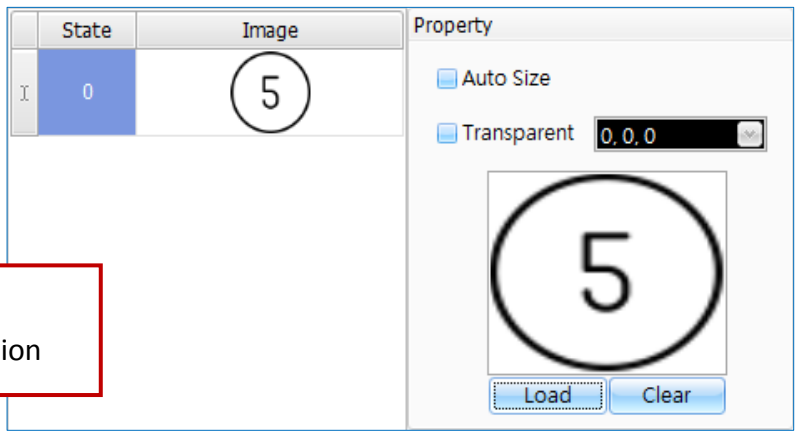
The foreground color, background color and background transparency attributes can be individually configured by selecting the required entry and editing the attributes as necessary.

The following is an overview of the options available in the **Text Properties** section of the Property View for the **Keyboard Button** object.

State		Used to indicate the index number for the State
Text		Used to display a preview of the input text or edit the text
Property	**Foreground Color**	Used to specify the color to be used for the text
	Background Color	Used to specify the color of the background to be used for the key
	Transparent	Used to specify whether or not the background color will be shown as transparent when displayed on the screen
	Text	Used to enter the text to be displayed.

- Picture Properties

  The **Picture Properties** dialog box is used to configure single or multiple images to be used for the **Keyboard Button** object.



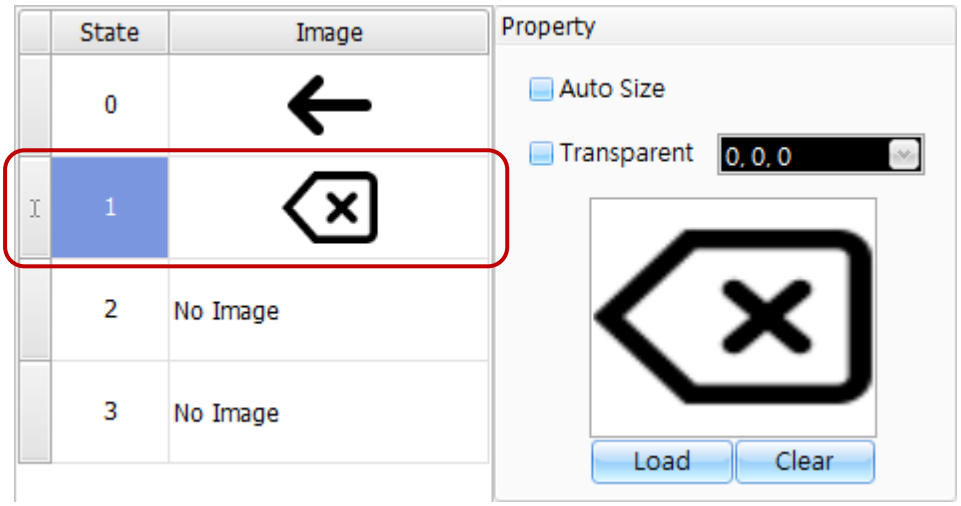
General Properties - Key State

If you choose the **Single State** option

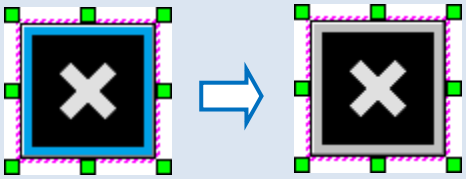
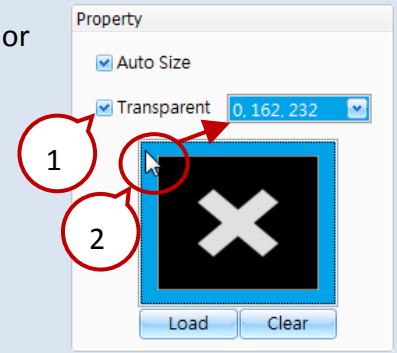
General Properties - Key State

If you choose the **Multiple states** option (supports 4 states):

The foreground color, background color and background transparency attributes can be individually configured by selecting the required entry and editing the attributes as necessary.

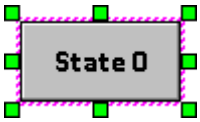


The following is an overview of the options available in the **Picture Properties** section of the Property View for the **Keyboard Button** object.

State		Used to indicate the index number for the State
Image		Used to display a preview of the image or mouse double-click it to load the image from the Image Manager window (see Section 11.4 for more details)
Property	**Auto Size**	Used to specify whether or not the image should be automatically stretched to fit the size of the object
	Transparent	Used to specify whether or not the specified color of the image will be shown as transparent when displayed on the screen  
	Load	Used to select an image from the **Image Manager** to be used for the button. See Section 11.4 “Image Manager” for more details.
	Clear	Used to clear the currently loaded image

- See Section 8.5 Common Attributes for details about the others attributes of the **Keyboard Button** object. Border (2), Text Shadow, Glisten, Security, Offset, Conditional Display, Conditional Control, Position and Size, Event, and Other.

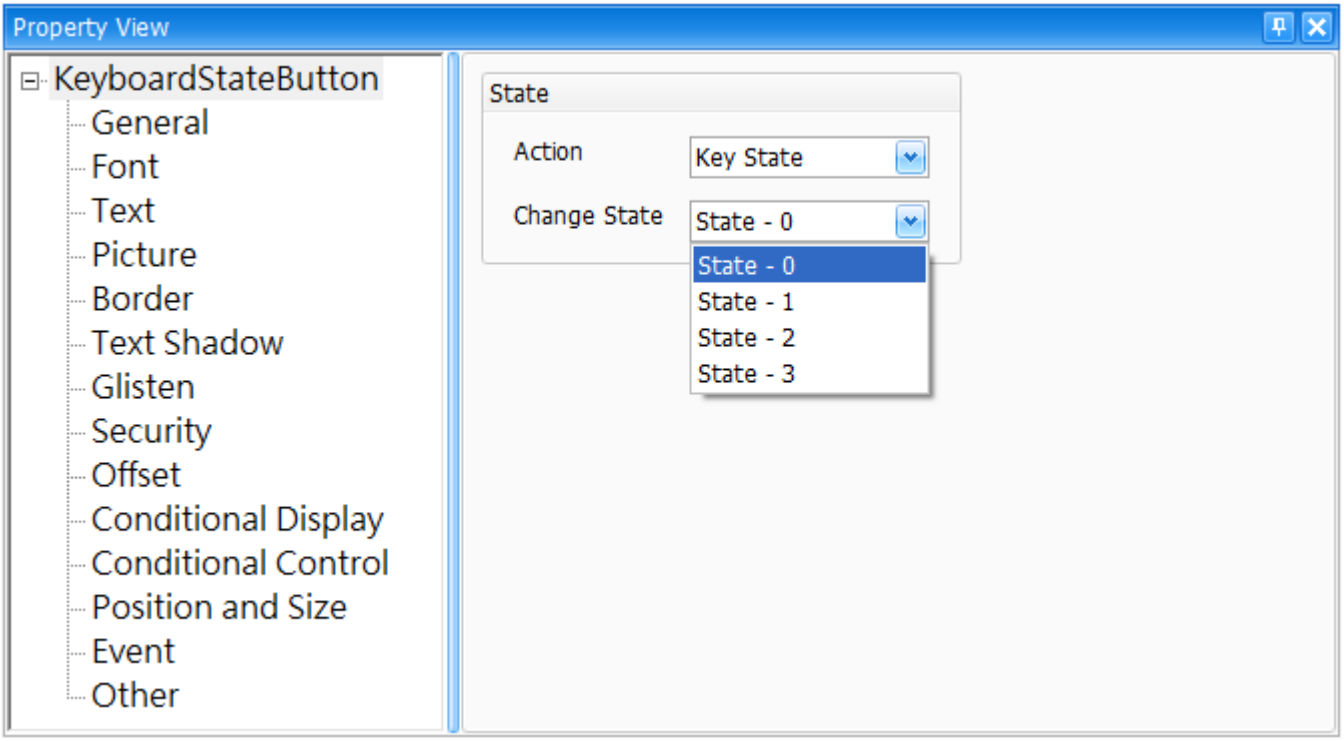
8.4.3 Keyboard State Button



The **Keyboard State Button** object can be used to switch states when the key state of the Keyboard Button (see Section 8.4.2) is set as multi-state. See Chapter 8 Basic Object Usage for details of how to create an object.

Note that this object can only be added to the Window-type screen, see Section 8.4 for more details.

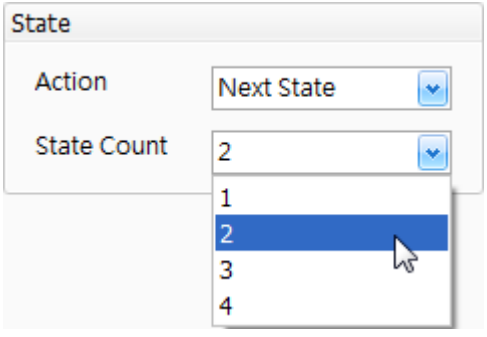
After creating a **Keyboard State Button** object, the **Property View** panel for the object will be automatically displayed allowing the various attributes of the object to be configured. The following is a description of how to configure the properties for the **Keyboard State Button** object.



- General Properties

  The **General Properties** dialog box is used to configure the action that will be performed when the keyboard state is changed.

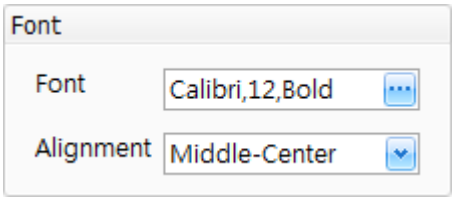
  The following is an overview of the options available in the **General Properties** section of the Property View for the **Keyboard State Button** object.



State	**Action**	Used to specify the way to switch the keyboard states, where: **1**. Key State (Switch to the assigned state) **2**. Next State (Switch to the next state) **3**. Previous State (Switch to the previous state)
	Change State	Used to specify which State No. will be switched to. Note that the State 0 to State 3 are defined in the Keyboard Button object, see Section 8.4.2 for more details
	State Count	Used to specify the number of states will be switched when the Action is set to Next State or Front State

- Font Properties

  The **Font Properties** dialog box is used to configure the type and size of the font to be used for the text on the **Keyboard State Button** object, as well as the alignment.

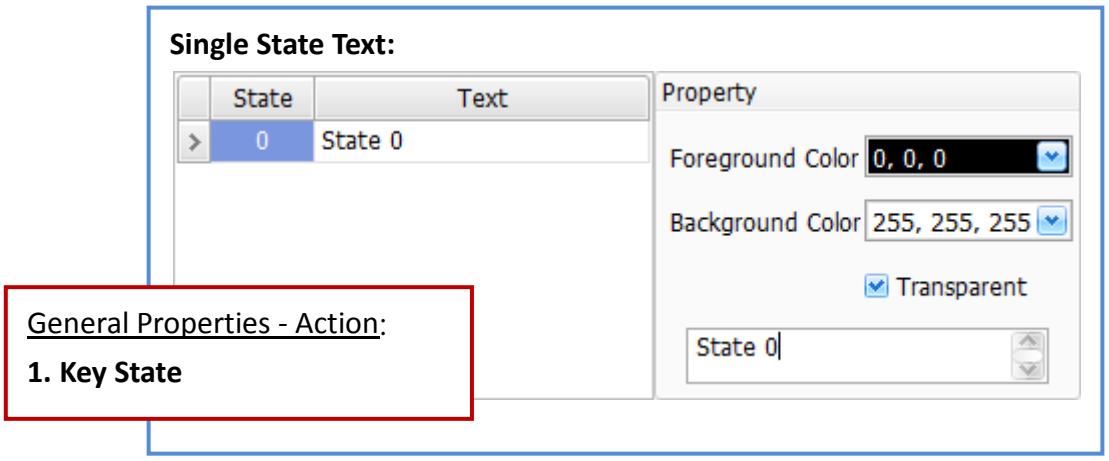
  

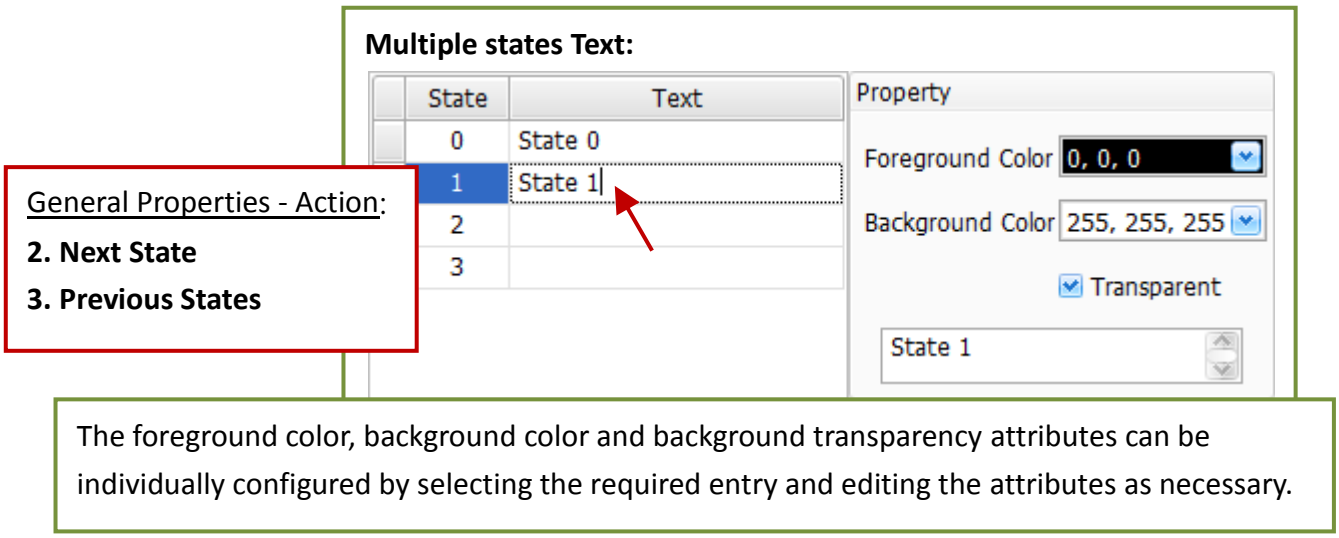
  The following is an overview of the options available in the **Font Properties** section of the Property View for the **Keyboard State Button** object.

Font	Used to specify the font and size of the text displayed on the key
Alignment	Used to specify the horizontal and vertical alignment of the text displayed on the key within the border of the object

- Text Properties

  The **Text Properties** dialog box is used to configure the text to be displayed for the **Keyboard State Button** object. Single or multiple states text can be configured.

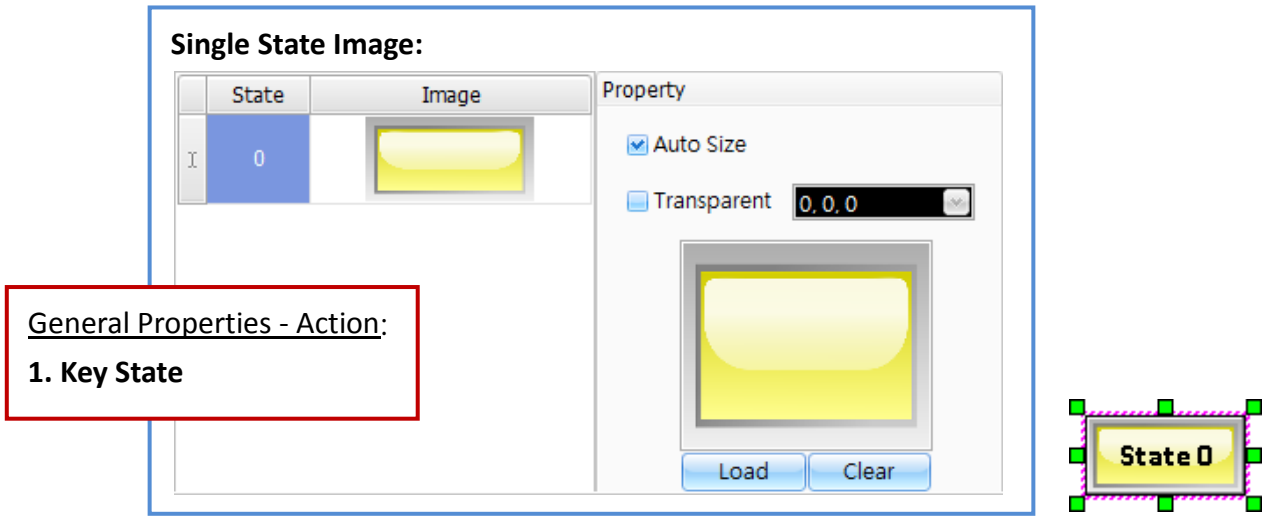


The following is an overview of the options available in the **Text Properties** section of the Property View for the **Keyboard State Button** object.

State		Used to indicate the index number for the State
Text		Used to display a preview of the input text or edit the text
Property	**Foreground Color**	Used to specify the color to be used for the text
	Background Color	Used to specify the color of the background to be used for the key
	Transparent	Used to specify whether or not the background color will be shown as transparent when displayed on the screen
	Text	Used to enter the text to be displayed.

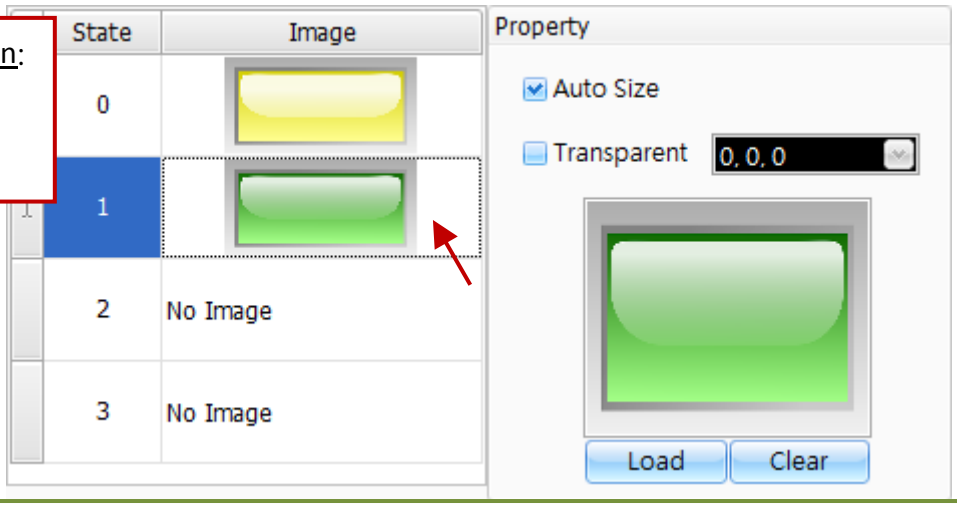
- Picture Properties

  The **Picture Properties** dialog box is used to configure single or multiple images to be used for the **Keyboard State Button** object.



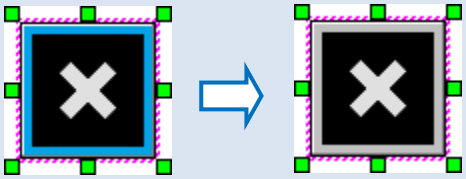
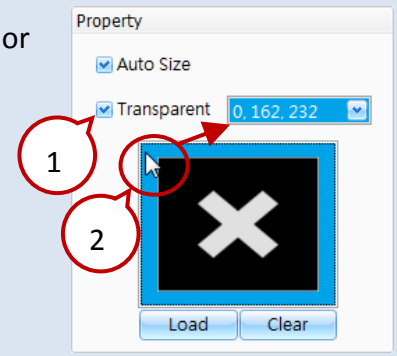
General Properties - Action:
2. Next State
3. Previous States

Multiple states Picture:



The foreground color, background color and background transparency attributes can be individually configured by selecting the required entry and editing the attributes as necessary.

The following is an overview of the options available in the **Picture Properties** section of the Property View for the **Keyboard State Button** object.

State		Used to indicate the index number for the State
Image		Used to display a preview of the image or mouse double-click it to load the image from the Image Manager window (see Section 11.4 for more details)
Property	**Auto Size**	Used to specify whether or not the image should be automatically stretched to fit the size of the object
	Transparent	Used to specify whether or not the specified color of the image will be shown as transparent when displayed on the screen  
	Load	Used to select an image from the **Image Manager** to be used for the button. See Section 11.4 “Image Manager” for more details.
	Clear	Used to clear the currently loaded image

- See Section 8.5 Common Attributes for details about the others attributes of the **Keyboard State Button** object. Border (2), Text Shadow, Glisten, Security, Offset, Conditional Display, Conditional Control, Position and Size, Event, and Other.

8.5 Common Attributes

The Toolbox contains eight categories of objects, including 8.1 Drafting Objects, 8.2 General Objects, 8.3 Switch Objects, 8.4 Keyboard Objects, 9.2 Recipe Objects, 9.6 Alarm Objects and 9.8 Sampling Objects. All objects need to be configured before they can be used. This section provides the introductions to attributes that are common to all objects, each of which is described in the following sections.

The following table shows the Section numbers for all objects which is convenient for you to go back to view the description for individual objects.

Toolbox – 8.1.x Drafting Objects:

1. Rectangle	2. Label	3. Image	4. Polygon	5. Circle
6. Line	7. Multi-Line	8. Calibration	9. Table	

Toolbox – 8.2.x General Objects:

1. State Image	2. StateText	3. Numeric	4. Numeric Table	5. Text Box
6. Date	7. Time	8. Bar Graph	9. Dial-Semicircle / Dial-Circle	
10. Drop Down List	11. Trend Graph	12. Connection View	13. Password List	

Toolbox – 8.3.x Switch Objects:

1. Bit Switch	2. Multistage Switch	3. Function Button	4. Connect Button	5. Jog (+ / -) Button

Toolbox – 8.4.x Keyboard Objects:

1. Keyboard Label	2. Keyboard Button	3. Keyboard State Button

Toolbox – 9.2.x Recipe Objects:

1. Recipe Numeric	2. RecipeText	3. Recipe List / Record List
4. Record View	5. Recipe Table View	6. Function Button (Recipe)

Toolbox – 9.6.x Alarm Objects:

1. Alarm View	2. Function Button (Alarm)

Toolbox – 9.8.x Sampling Objects:

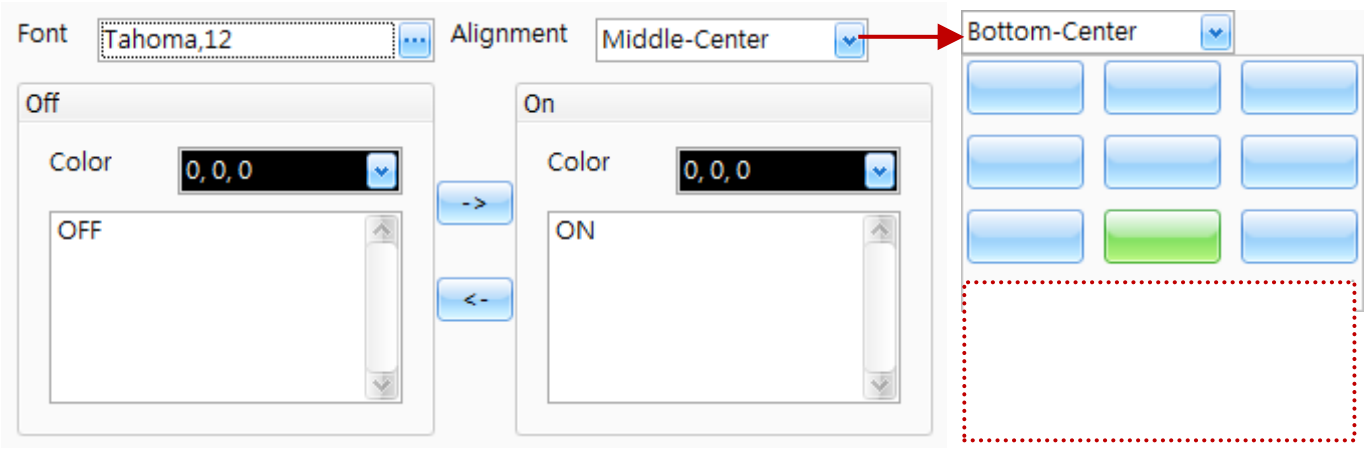
1. Real-time / History Trend Chart	2. Data Sampling View	3. Function Button (Sampling)

A. Text Properties

The **Text Properties** dialog box is used to configure the font, color, and alignment of the text used for the object in both the ON and OFF states.

For Objects:

Switch (Bit Switch, Function Button, Connect Button, Jog (+ / -) Button), Recipe (Function Button (Recipe)), Alarm (Function Button (Alarm)), and Sampling (Function Button (Sampling))

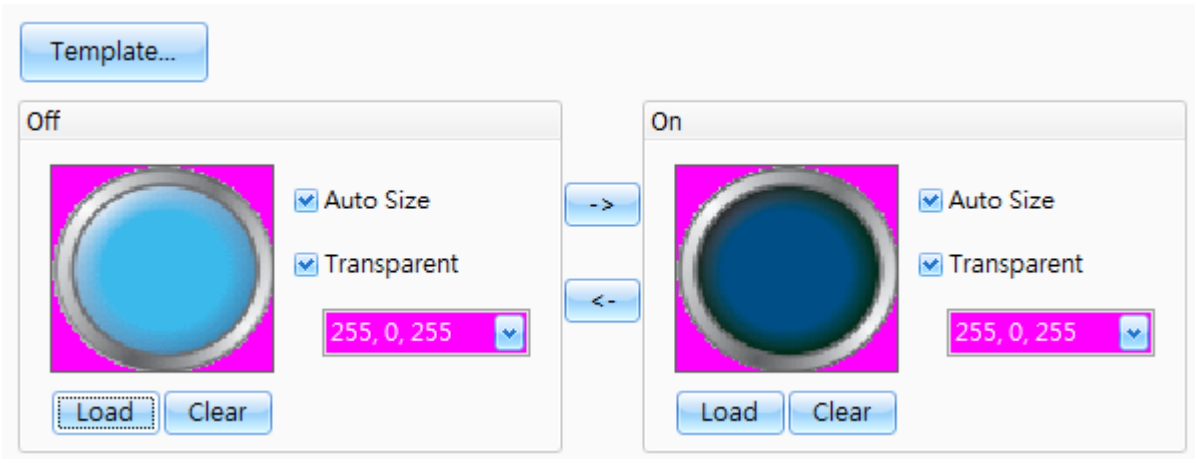


The following is an overview of the options available in the **Text Properties** section of the Property View for the object.

Font		Used to specify the type and size of the font to be used for the text of the button
Alignment		Used to specify the horizontal and vertical position of the text on the button within the border of the object
OFF	**Color**	Used to specify the text and its color to be shown when the button is in the OFF state
ON	**Color**	Used to specify the text and its color to be shown when the button is in the ON state

B. Image/Picture Properties

The **Image Properties** dialog box is used to configure the image used to indicate the status of the object in both the ON and OFF states.



For Objects:

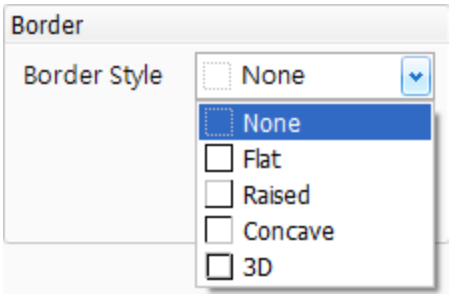
Switch (Bit Switch, Function Button, Connect Button, Jog (+ / -) Button), Recipe (Function Button (Recipe)), Alarm (Function Button (Alarm)), and Sampling (Function Button (Sampling))

The following is an overview of the options available in the **Image Properties** section of the Property View for the object.

Template		Used to select a template for the button
OFF / ON	**Auto Size**	Used to specify whether or not the image should be stretched to fit the size of the object when the button is in the OFF/ON state
	Transparent	Used to specify whether or not the specific color will be shown as transparent when the button is in the OFF/ON state
	Color	Used to specify which color will be set to transparent when the button is in the OFF/ON state
	Load	Used to select or add an image from the **Image Manager** to be used when the button is in the ON state. See Section 11.4 “Image Manager” for more details.
	Clear	Used to clear the currently loaded image

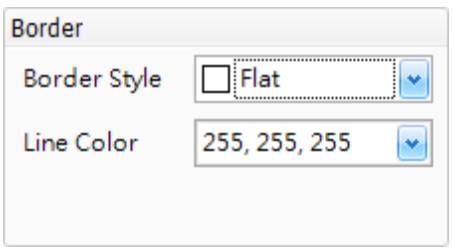
C. Border Properties (1)

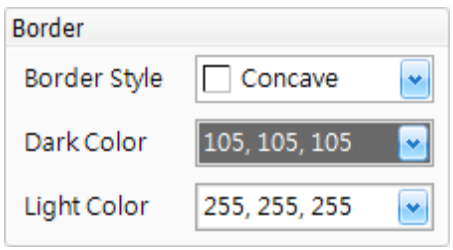
The **Border Properties** dialog box is used to configure the border style and the line color for the object. Five options are available: None, Flat, Raised, Concave, and 3D.

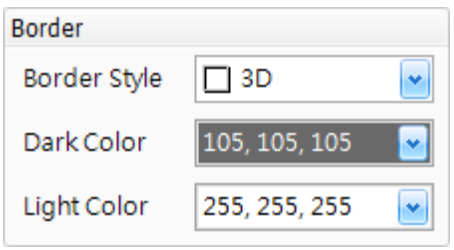


For Objects:

Drafting (All), General (All), Keyboard (Keyboard Label), Recipe (Recipe Numeric, RecipeText, Recipe List / Record List, Record View, Recipe Table View), Alarm (Alarm View), and Sampling (Real-time / History Trend Chart, Data Sampling View)







The following is an overview of the options available in the **Border Properties** section of the Property View for the object.

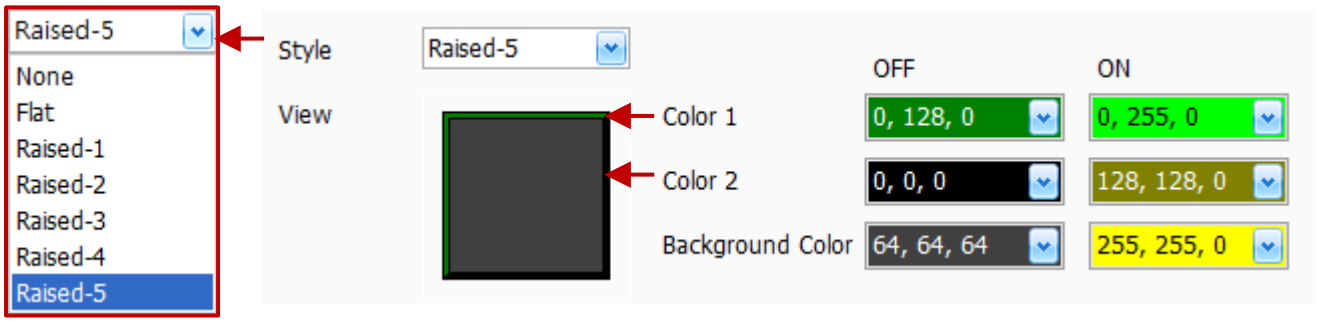
Border Style	Used to specify the style of the border. Five options are available: None, Flat, Raised, Concave, and 3D
Line Color	Used to specify the border color if the Border Style is set to Flat
Dark Color	Used to specify the dark border color of the object if the Border Style is set to Raised, Concave, or 3D
Light Color	Used to specify the light border color of the object if the Border Style is set to Raised, Concave, or 3D

D. Border Properties (2)

The **Border Properties** dialog box is used to configure the border style for the object.

For Objects:

Switch (All), Keyboard (Keyboard Button, Keyboard State Button), Recipe (Function Button (Recipe)), Alarm (Function Button (Alarm)), and Sampling (Function Button (Sampling))

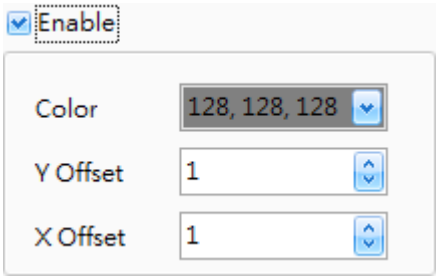


The following is an overview of the options available in the **Border Properties** section of the Property View for the object.

Style		Used to specify the style used for the border, and can be selected from None, Flat, Raised-1, Raised-2, Raised-3, Raised-4, Raised-5
View		Used to display a preview of the button **Notice:** you can click the button to view the color changes.
OFF / ON	**Color 1**	Used to specify the color of the border on the top-left side when the button is in the OFF/ON state
	Color 2	Used to specify the color of the border on the bottom-right when the button is in the OFF/ON state
	Background Color	Used to specify the fill color of the background when the button is in the OFF/ON state

E. Text Shadow Properties

The **Text Shadow Properties** dialog box is used to enable the text shadow for the object.



For Objects:

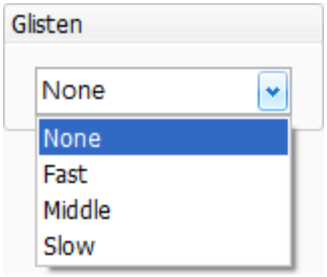
Drafting (Label), General (State Image, StateText, Numeric, Numeric Table, Text Box, Date, Time), Switch (All), Keyboard (All), Recipe (Recipe Text, Function Button (Recipe)), Alarm (Function Button (Alarm)), and Sampling (Function Button (Sampling))

The following is an overview of the options available in the **Text Shadow Properties** section of the Property View for the object.

Color	Used to specify the color to be used for the text shadow
X Offset	Used to specify how far the text shadow will be displaced in the horizontal direction in pixels
Y Offset	Used to specify how far the text shadow will be displaced in the vertical direction in pixels

F. Glisten Properties

The **Glisten Properties** dialog box is used to configure the glisten speed for the object. Four options are available: None, Slow, Medium, and Fast.



For Objects:

Drafting (All), General (All), Switch (All), Keyboard (All), Recipe (All), Alarm (All), and Sampling (All)

G. Security (Authority Level) Properties

The **Security Properties** dialog box for the object is used to configure the operating authority level for a user, where 0 is the lowest authority and 9 is the highest.

The user can create the password for each authority level on the **Password** page in the **Project View** panel; see Section 11.2 for more details. If the login authority level for the object is insufficient, a warning dialog box will be automatically displayed.

For Objects:

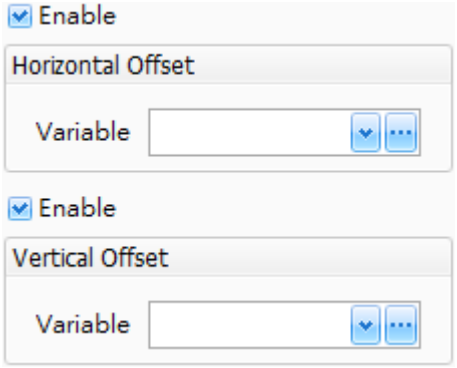
General (Numeric, Numeric Table, Connection View), Switch (All), Keyboard (Keyboard Button, Keyboard State Button), Recipe (Recipe Numeric, Recipe List / Record List, Record View, Function Button (Recipe)), Alarm (Function Button (Alarm)), and Sampling (Function Button (Sampling))

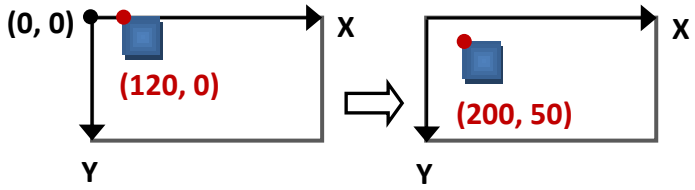
H. Offset Properties

The **Offset Properties** dialog box is used to configure the horizontal and vertical movement offset values for the object, which are controlled via a variable. For more details related to the usage of variables, see Section 7.2.2 Using Variables.

For Objects:

Drafting (All), General (All), Switch (All), Keyboard (All), Recipe (All), Alarm (All), and Sampling (All)



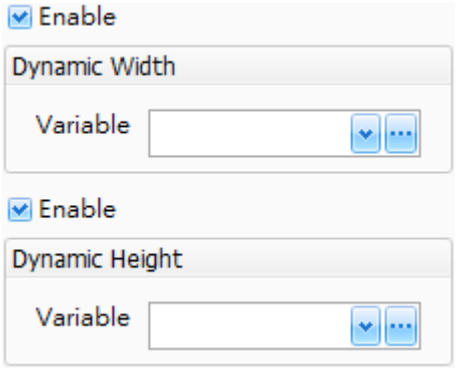


I. Dynamic Size Properties

The **Dynamic Size Properties** dialog box is used to enable the function that allows the width and/or height of the object to be changed dynamically during runtime, and is controlled via a variable. For more details related to the usage of variables, see Section 7.2.2 Using Variables.

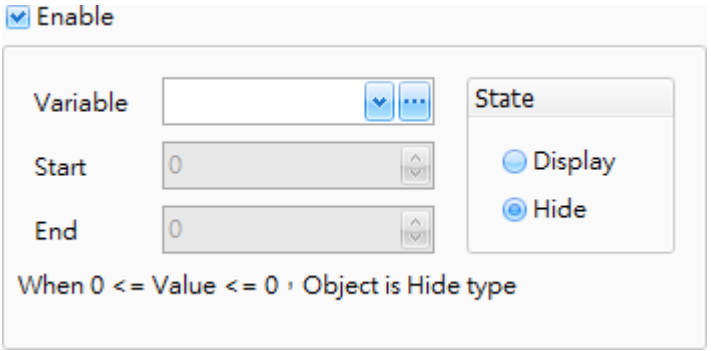
For Objects:

Drafting (Rectangle, Image, Polygon, Circle, Line, Multi-Line)



J. Conditional Display Properties

The **Conditional Display Properties** dialog box enables the function used to display or hide the object in specific conditions to be configured, and is controlled via a variable. For more details related to the usage of variables, see Section 7.2.2 Using Variables.



As the figure shows, when Start ≦ Value ≦ End, the object will be hidden.

For Objects:

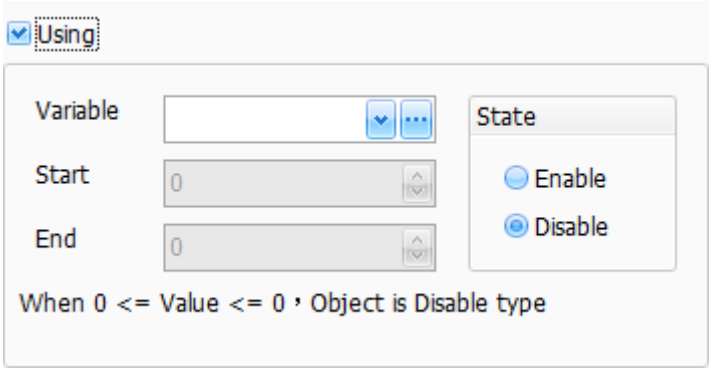
Drafting (All), General (All), Switch (All), Keyboard (All), Recipe (All), Alarm (All), and Sampling (All)

The following is an overview of the options available in the **Conditional Display Properties** section of the Property View for the object.

Variable	Used to specify the variable to be used for the conditional display.
Start	Used to specify the starting value for the conditional display. Note that the End value must be greater than or equal to the Start value.
End	Used to specify the ending value for the conditional display.
State	Used to specify whether the object to be displayed or hidden in specified conditions.

K. Conditional Control Properties

The **Conditional Control Properties** dialog box enables the function used to enable or disable the application function of the object in specific conditions to be configured, and is controlled via a variable. For more details related to the usage of variables, Section 7.2.2 Using Variables.



As the figure shows, when Start ≦ Value ≦ End, the function of the object will be disabled.

For Objects:

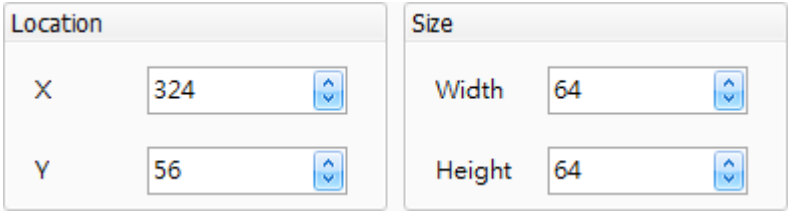
General (Numeric, Numeric Table, Drop Down List), Switch (All), Keyboard (Keyboard Button, Keyboard State Button), Recipe(Recipe Numeric, RecipeText, Function Button (Recipe)), and Sampling (Function Button (Sampling))

The following is an overview of the options available in the **Conditional Control Properties** section of the Property View for the object.

Variable	Used to specify the variable to be used for the conditional control.
Start	Used to specify the starting value for the conditional control. Note that the End value must be greater than or equal to the Start value.
End	Used to specify the ending value for the conditional control.
State	Used to specify whether the function to be enabled or disabled in specified conditions.

L. Position and Size Properties

The **Position and Size Properties** dialog box is used to configure the relative position and size of the object.



For Objects:

Drafting (All), General (All), Switch (All), Keyboard (All), Recipe (All), Alarm (All), and Sampling (All)

The following is an overview of the options available in the **Position and Size Properties** section of the Property View for the object.

Location (pixels)	**X**	Used to specify the horizontal coordinate position for the object
	Y	Used to specify the vertical coordinate position for the object
Size (pixels)	**Width**	Used to specify the width of the object
	Height	Used to specify the height of the object

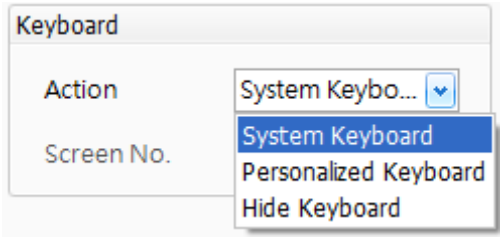
Note that the coordinates for the upper left corner of the screen design area are (0,0).

M. Keyboard Properties

The **Keyboard Properties** dialog box is used configure the keyboard function that will be activated when a object is clicked.

For Objects:

General (Numeric, Numeric Table, Text Box, Password List) and Recipe (Recipe Numeric, RecipeText, Recipe List / Record List, Record View)

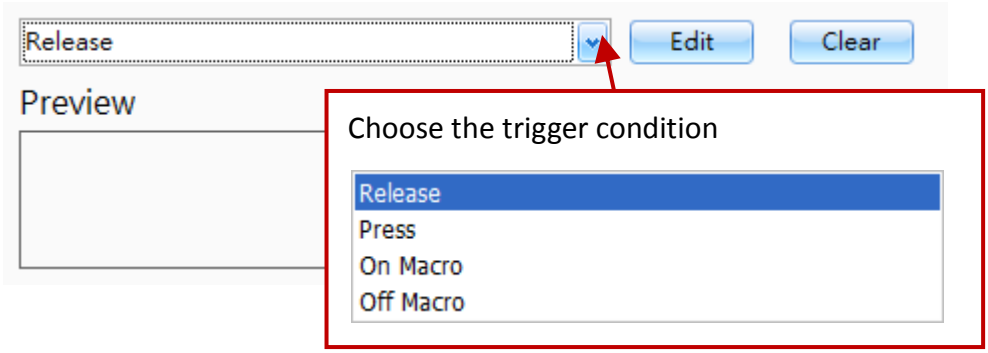


The following is an overview of the options available in the **Keyboard Properties** section of the Property View for the object.

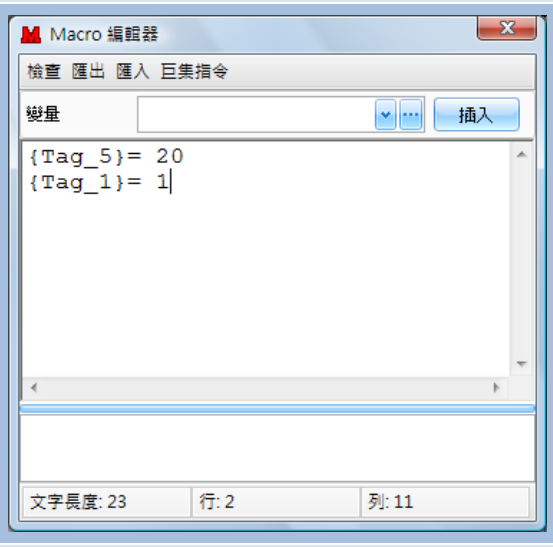
Action	**System Keyboard**	Used to enable an on-screen keyboard with a default style
	Personalized Keyboard	Used to enable the on-screen keyboard with the customized style. See Section 8.4 Keyboard Objects for more details
	Hide Keyboard	Used to disable the keyboard function
Screen No.		Used to specify the page number of the Window screen

N. Event Properties

The **Event Properties** dialog box is used to edit the macro procedure which will be used when the **Setting** condition (i.e., Release, Press, On Macro, or Off Macro) of the object is triggered. For more details related to the usage of Macros, see the 10-B Macros Section.



The following is an overview of the options available in the **Event Properties** section of the Property View for the object.

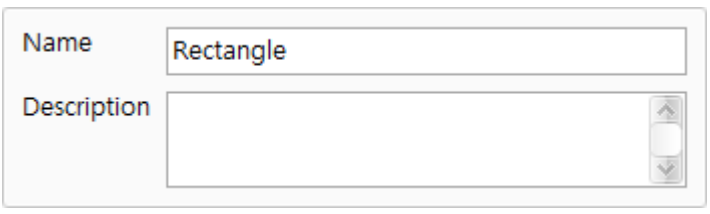
Setting/Release/Press/On Macro/Off Macro: Used to choose the trigger condition for this object.	
Setting	General (Numeric, Numeric Table) ➢ "Setting" means that the macro commands will be triggered after writing data to the variable. See General Properties.
Release **Press** **On Macro** **Off Macro**	Switch (Bit Switch) ➢ Four types of trigger conditions can be selected; When the state of the button is Release, Press, On, or Off, the customized macro commands will be triggered.
Release **Press**	Switch (Multistage Switch, Function Button, Connect Button, Jog (+ / -) Button), Recipe (Function Button (Recipe)), Alarm (Function Button (Alarm)), and Sampling (Function Button (Sampling)) ➢ Two types of trigger conditions can be selected; When the state of the button is Release, Press, the customized macro commands will be triggered.
Edit	Used to edit the macro command. When you click the **Edit** button as the figure above, the **Macro Editor** dialog box will be displayed, and then you can edit the macro command. See Section 10.6 Macro Editor for more details. 
Clear	Used to clear macro commands displayed in the Preview text box. For more details related to the usage of Macros, see Section 10-B Macros.
Preview	Used to preview macro commands. For more details related to the usage of Macros, see Section 10-B Macros.

O. Other Properties

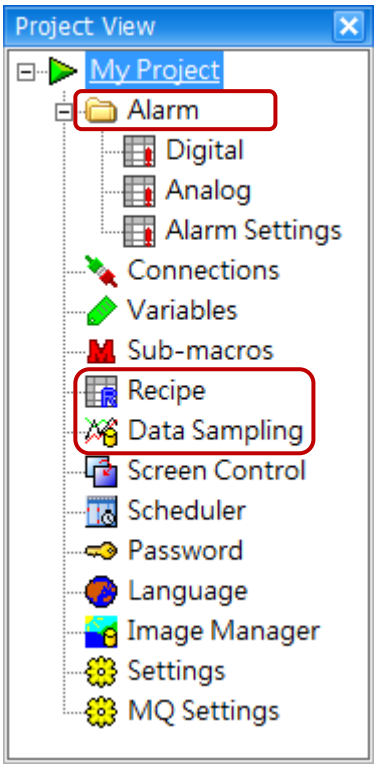
The **Other Properties** dialog box is used to assign a user-defined name and description for the object.

For Objects:

Drafting (All), General (All), Switch (All), Keyboard (All), Recipe (All), Alarm (All), and Sampling (All)



Chapter 9 Advanced Function and Object Usage



The Creator software provides a number of advanced functions and objects that can be used to perform enhanced operations. These are separated into two categories: **Function Management**, which can be used to create and manage the functions that are found in the **Project View** panel (see Section 4.11 "Project View" for details), and **Object Usage**, which can be used to add objects from the **Toolbox** (see Section 4.12 "Toolbox" for details) that use functions related to monitoring and control of the HMI screen.

The process for implementing the advanced functions is to first use the Recipe, Alarm, and Data Sampling functions that are accessed via the **Project View** panel to manage the advanced functions, and then use the Recipe, Alarm, and Data Sampling objects located in the **Toolbox** to design the HMI screen.

9 – (A) Recipe (Available soon)

A **Recipe** is usually used to configure and store device processes or machine parameters, which can then be downloaded from or uploaded to an HMI device. The recipe is stored in the internal memory, meaning that data can be retained after the device is powered off. There are two parts to the structure of a Recipe, the Element, and the Record, each of which are described below.

Element	**Record**
In a recipe, the device processes or machine parameters are defined as **Elements** and a maximum of 512 elements can be created for each recipe. An Element is used to set a mapping variable that can be assigned as a BIT, BYTE, CHAR, WORD, INT, DWORD, LONG or FLOAT data type.	A **Record** is composed of a number of Elements, and a maximum of 32768 records can be created for each recipe.

For example, the parameters related to a paper cutting process include the Product Name, and the Width (Element1) and the Length (Element2) of the cut. In this example, there are five different products based on the size of the paper to be cut. The structure of the recipe is shown on the right:

元素 記錄

名稱	Paper Width	Paper Length
A3	29.7	42
A4	21	29.7
A5	14.8	21
B4	25.7	36.4
B5	18.2	24.7

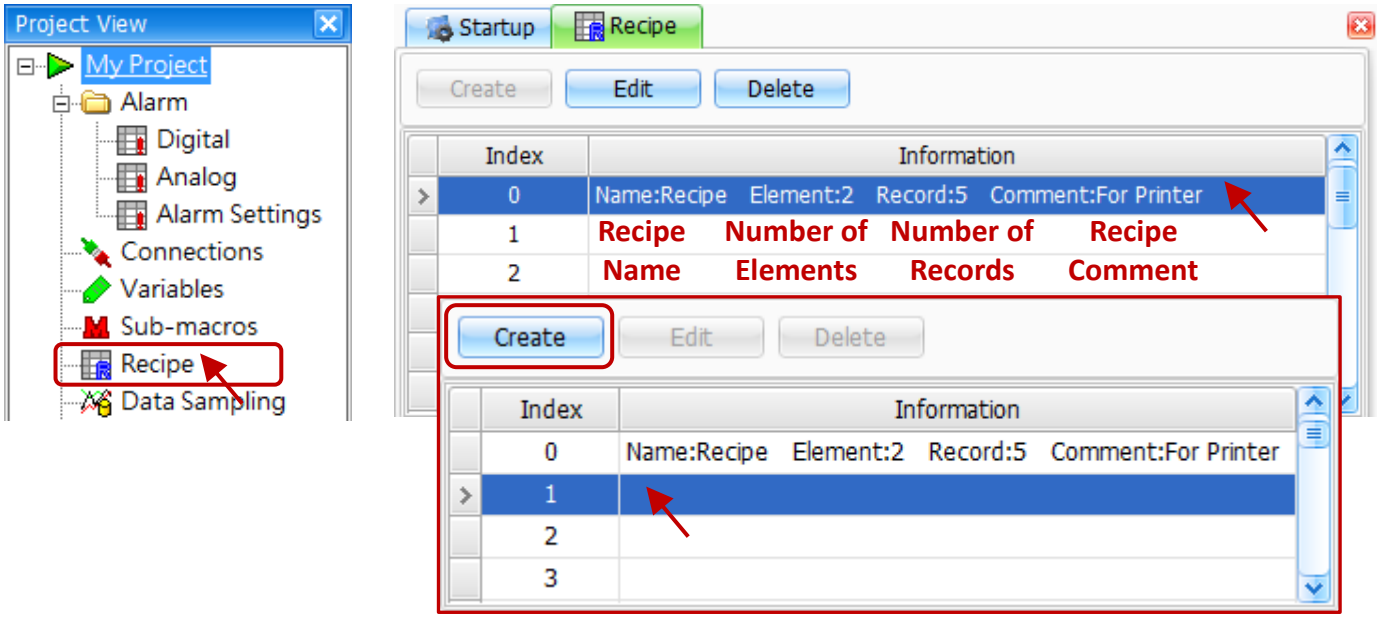
9.1 Recipe Management

The Recipe function is used to create, edit, and manage a recipe, and a maximum of 20 recipes can be created. The basic structure of a recipe includes two components, an Element and a Record, which can be configured in the Recipe Editor dialog box, which is described in subsequent sections below.

In this example, suppose that the operators of a paper manufacturing facility wish to create a recipe to automate their paper cutting process. There are five different sizes of paper, including A3, A4, A5, B4 and B5, and the process parameters for the different sizes include the length and the width. The recipe for this process can be constructed in the following manner:

Recipe	Include a user-defined name for the recipe, e.g., “Paper Size Recipe”.
Element	Specify the Elements to be included, e.g., Paper Width and Paper Length
Record	Provide the details of the paper size (A3, A4, A5, B4, B5), including the width and length for each size.

To access the Recipe page, double-click the **Recipe** function in the **Project View** panel. On the Recipe page, click an empty row in the table and click the Create button to create a new Recipe, and you can also double-click any row to create or edit a Recipe. See next Section for details.

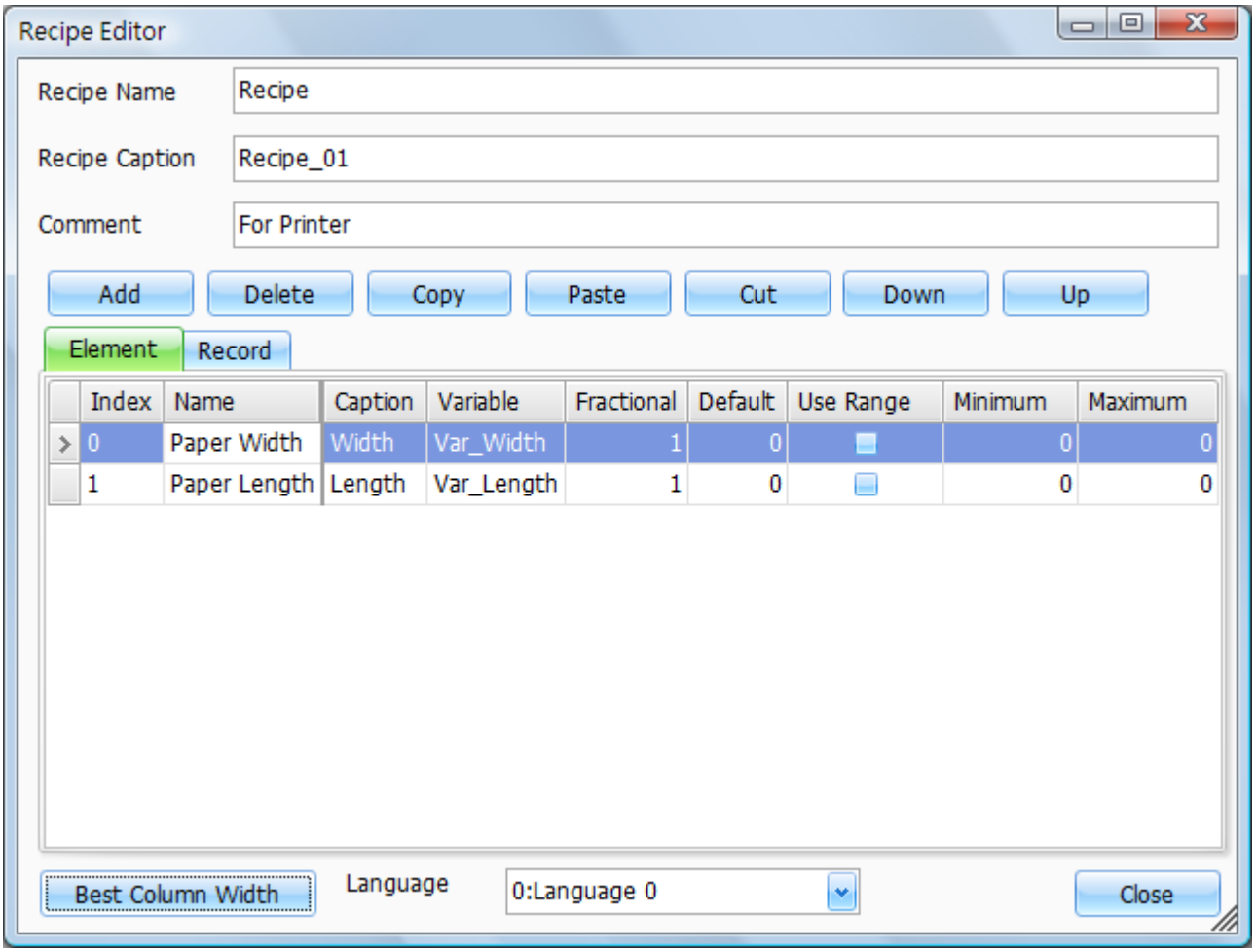


The following is an overview of the options available on the **Recipe** page for the Recipe function.

Create	Used to create a recipe. A maximum of 20 recipes can be created.
Edit	Used to edit a recipe
Delete	Used to delete a recipe
Index	Used to indicate the index number of the Recipe
Information	Used to indicate the details of the Recipe

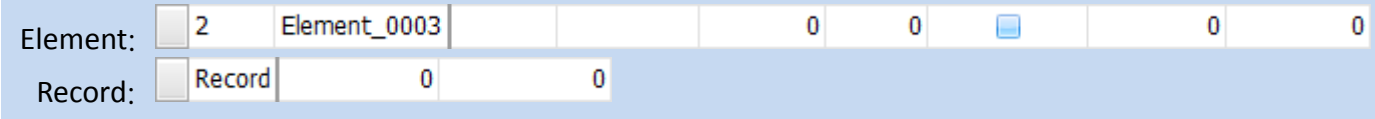
9.1.1 Recipe Editor

The Recipe Editor dialog box is used to add, edit, and manage the Elements and Records for a recipe, as well as configure the parameters of the recipe. On the Recipe page (see the previous page), double-click any row in the table (Index 0 in this example) to open the **Recipe Editor** dialog box, and then enter the details for the Name, Caption of the Recipe and a Comment. The usage of the Element and Record are described in subsequent sections below



The following is an overview of the options available in the **Recipe Editor** dialog box.

Recipe Name	Used to specify a name for the recipe
Recipe Caption	Used to specify the text that will be displayed on the HMI screen Note that multiple languages can be used. See the Language item below
Comment	Used to add a description for the recipe
Element Tab	Used to access the Element data page
Record Tab	Used to access the Record data page
Add	Used to add a new Element or Record



Delete	Used to delete an Element or Record
Copy	Used to copy an Element or Record

Copy (Index 1 / B5) and Paste (will be pasted to the last row)

Element:

1	Paper Length	Length	Var_Length	1	0	☐	0	0
2	Element_0000	Length	Var_Length	1	0	☐	0	0

Record:

B5	20	26
B5	20	26

Paste	Used to paste an Element or Record
Cut	Used to cut an Element or Record

Cut (Index 1) and Paste (will be pasted to the last row)

Element:

1	Element_0000	Length	Var_Length	1	0	☐	0	0
2	Paper Length	Length	Var_Length	1	0	☐	0	0

1	Paper Length	Length	Var_Length	1	0	☐	0	0
2	Element_0000	Length	Var_Length	1	0	☐	0	0

Down	Used to move the selected Element or Record down one row
Up	Used to move the selected Element or Record up one row
Best Column Width	Used to minimize the column width depends on characters in each cells

Element | Record

Index	Name	Caption	Variable	Fractional
0	Paper Width	Width	Var_Width	1
1	Paper Length	Length	Var_Length	1

Element | Record

Index	Name	Caption	Variable	Fractional	Default	Use Range	Minimum	Maximum
0	Paper Width	Width	Var_Width	1	0	☐	0	0
1	Paper Length	Length	Var_Length	1	0	☐	0	0

Language	Used to configure or change the Caption text after selecting the specific language, see Section 11.3 Language for more details on language setting

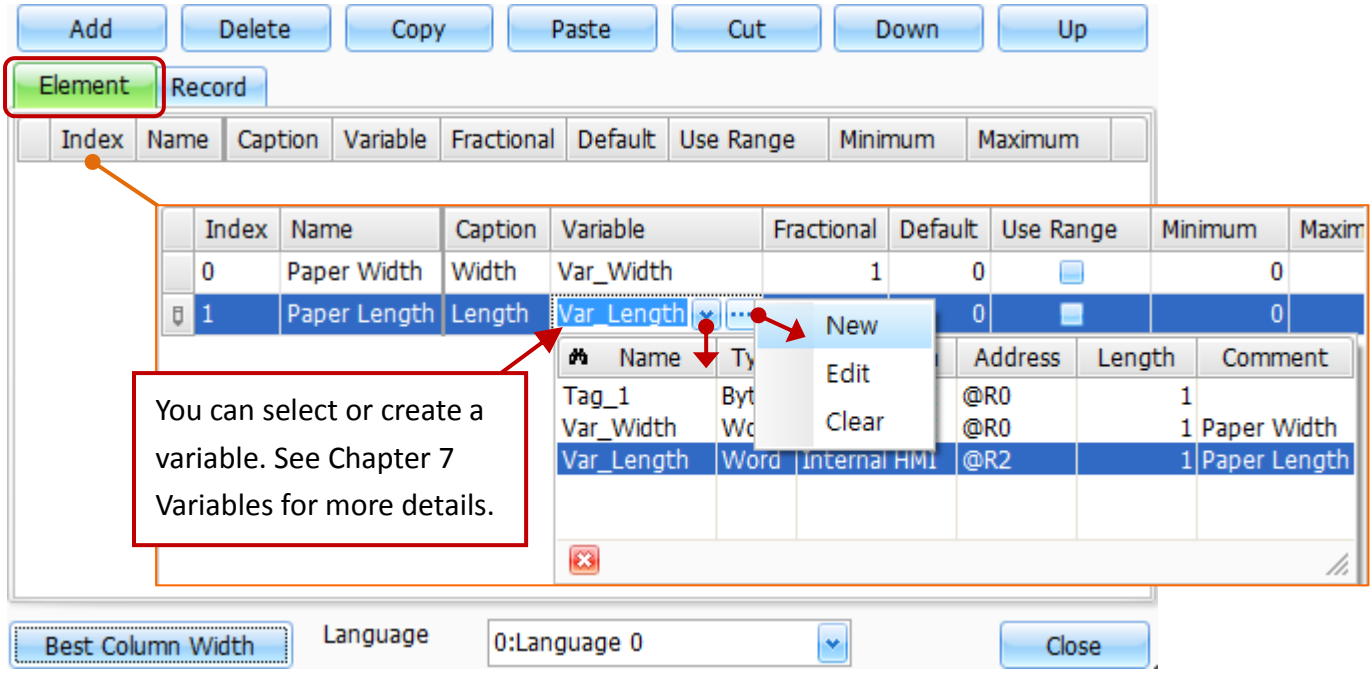
0:English

0:English
1:Chinese (T)
2:Chinese (S)

Index	Name	Caption	Variable	Fractional	Default	Use Range	Minimum	Maximum
0	Paper Width	寬 (W)	Var_Width	1	0	☐	0	0
1	Paper Length	長 (L)	Var_Length	1	0	☐	0	0

9.1.2 Element Tab

In a recipe, the device processes or machine parameters are defined as **Elements** and a maximum of 512 Elements can be created for each recipe. The user can set the mapping variable for each element, and the data type can be assigned as a BIT, BYTE, CHAR, WORD, DWORD, INT, LONG or FLOAT.



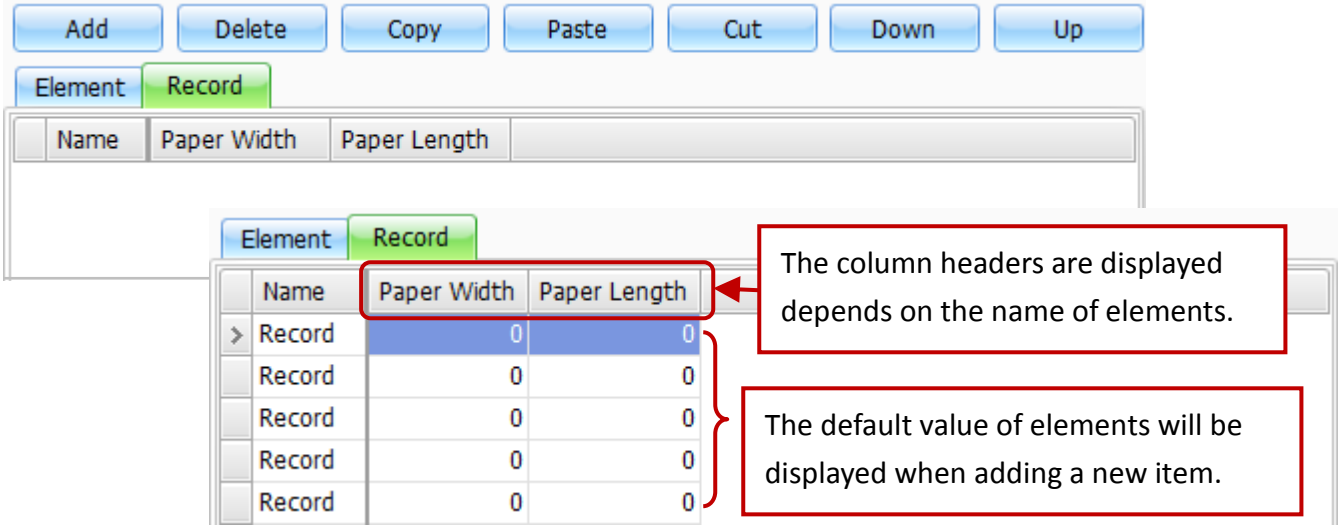
The following is an overview of the options available in the **Element** tab of the **Recipe Editor** dialog box.

Index	Used to indicate the index number of the Element
Name	Used to indicate the name of the Element, which must be unique in the same recipe
Caption	Used to specify the name that will be displayed on the HMI screen. Multiple languages can be set after you choose the Language option, described on the previous page.
Variable	Used to specify the variable for data access. Note that only numeric type variables are supported. The Data type can be assigned as BIT, BYTE, CHAR, WORD, DWORD, INT, LONG or FLOAT. See Chapter 7 “Variable” for more details of how to set a variable.
Fractional	Used to specify the position of the decimal point
Default	Used to specify the default settings for an Element
Use Range	Used to specify whether the input range limitations should be enabled for the Element
Maximum	Used to specify the maximum value for the input range
Minimum	Used to specify the minimum value for the input range

To create the Element Data, first double-click the Recipe that has been created to open the **Recipe Editor** dialog box (see Section 9.1.1). On the **Element** tab, click the **Add** button to create a new Element. Enter the details for the Name, Caption, Variable and Fractional, etc., for the Element (Paper Width in this example) and then repeat this process for any additional Elements, for instance, Paper Length in this example.

9.1.3 Record Tab

A **Record** is composed of a number of Elements, and a maximum of 32768 records can be created for each recipe. The Record Tab is used to create the Records related to a specific Element that has been previously created.

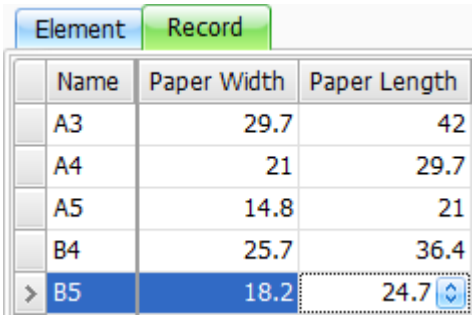


The following is an overview of the options available in the **Record** tab of the **Recipe Editor** dialog box.

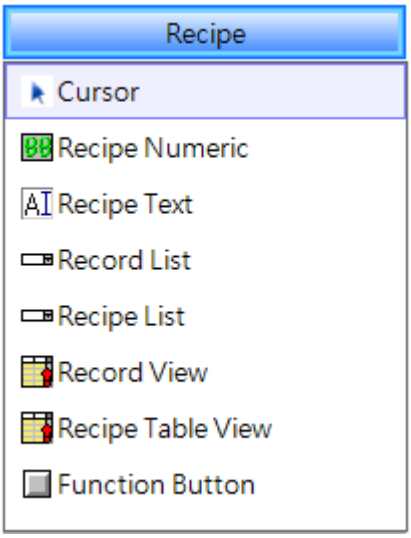
Name Column	Used to specify the name of the Record
Element Column(s)	Used to specify the value of the Element. The number of Element columns that are displayed is based on the number of Elements that were created on the Element page (see Section 9.1.2 “Element Tab” for details)

In this example, there are five different sizes of paper, including A3, A4, A5, B4 and B5. To create the Record Data for the different sizes of paper, click the **Record** Tab in the **Recipe Editor** dialog box, and then click the **Add** button to create a new Record. Enter a name for the Record in the Name field and then enter the relevant values for both the Paper Width and Paper Length Elements in the respective fields. Repeat this process for all the required Records.

The recipe for this automating the paper cutting process can be configured as the figure below:

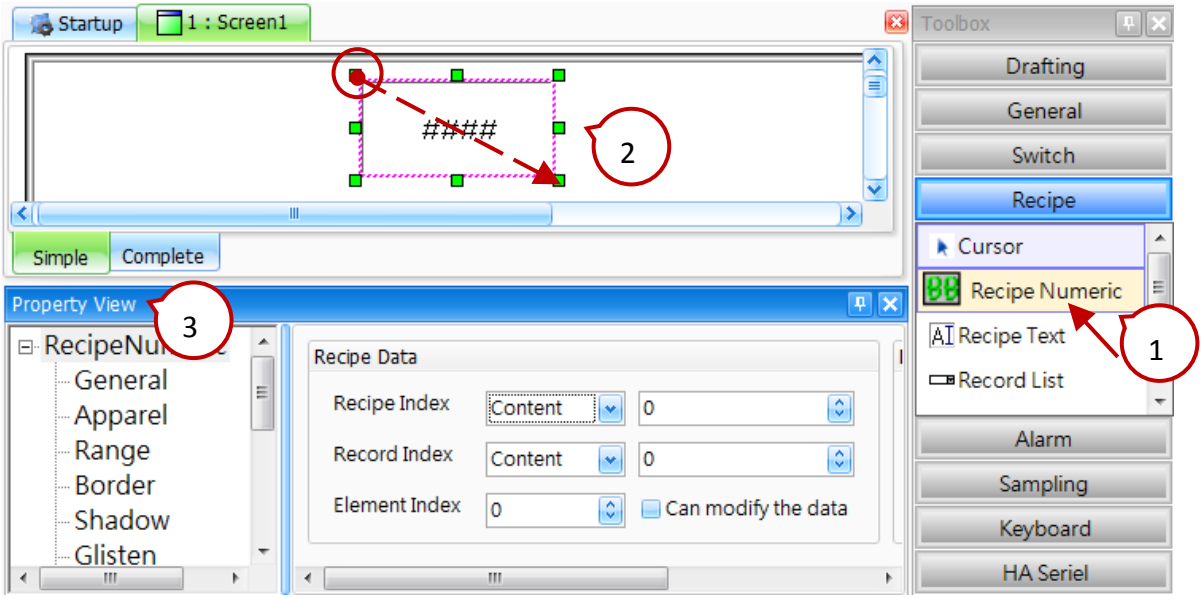


9.2 Recipe Objects



The **Recipe** category of the Toolbox includes a variety of objects that can be used to view or modify the data for elements or records related to a specific recipe on the HMI screen, each of which are described in more detail in the following sections.

To create an object, first click the name of the desired object in the **Recipe** category of the Toolbox. Position the mouse cursor on the Screen Design Area, and then click and hold the left mouse button to drag the object until it is the desired size and shape, as illustrated in the diagram below, and then release the mouse button. After creating a object, the **Property View** panel for the object will be automatically displayed allowing the various attributes of the object to be configured.

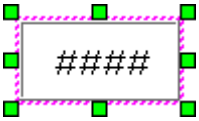


The **Function Button** object is provided in each of the **Switch, Recipe, Alarm** and **Sampling** categories and can be used to execute up to 16 functions, such as recipe transmission, changing screens, or incrementing and decrementing variable values, etc.

When the button is clicked, the functions that have been defined will be executed in sequence. The **Recipe** functions that can be configured in the **Function Button** object include:

Recipe – Add	Recipe – Remove	Recipe – Upload	Recipe – Download
Recipe – Export	Recipe – Import	Recipe – First Record	Recipe –Prior Record
Recipe – Next Record	Recipe – Last Record		

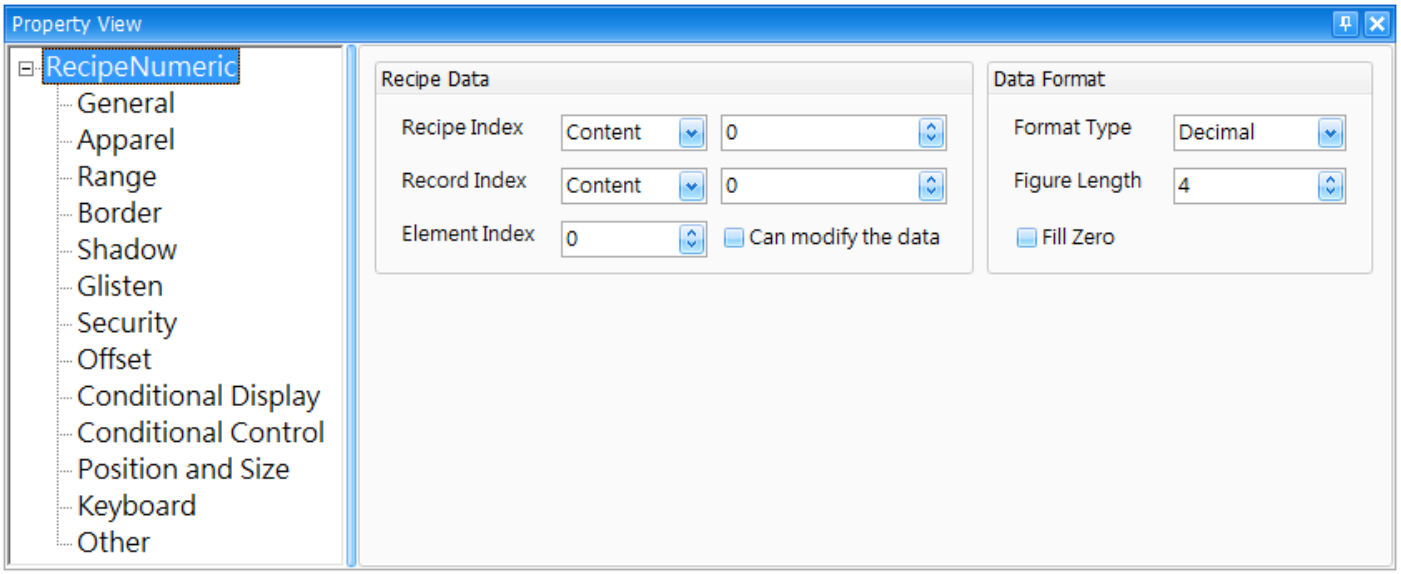
9.2.1 Recipe Numeric



Before using the **Recipe Numeric** object, you need to configure the **Recipe** function in the **Project View** panel, see Chapter 9 for more details. This object is used to display the numerical data for the recipe record, see Section 9.1.3 Record Tab for more details.

By using this object, you can also input value for the specified record, and the recipe will be directly updated the modified data, which will still be retained even after the device is switched off.

See Section 9.2 Recipe Objects for details of how to create an object. After creating a **Recipe Numeric** object, the **Property View** panel for the object will be automatically displayed allowing the various attributes of the object to be configured. The following is a description of how to configure the properties for the **Recipe Numeric** object.



- General Properties

  The **General Properties** dialog box is used to configure the display data and format to be used for the **Recipe Numeric** object.

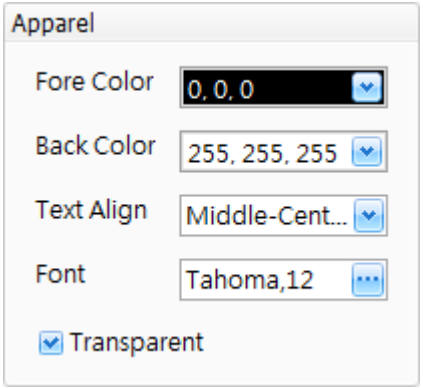
The following is an overview of the options available in the **General Properties** section of the Property View for the **Recipe Numeric** object.

Recipe Data	**Recipe Index**	Used to specify the index number of the recipe to be displayed Note that if the index number is invalid, the data will be displayed as blank
	Record Index	Used to specify the index number of the record to be displayed Note that if the index number is invalid, the data will be displayed as blank
Recipe Data	**Element Index**	Used to specify the index number of the element to be displayed Note that if the element number is invalid, the data will be displayed as blank

	Can modify the data	Used to specify whether or not the function that allows the data to be modified is enabled
Data Format	**Format**	Used to specify the format to be used to display the data: **Decimal**: The data will be displayed in decimal format **Signed Decimal**: The data will be displayed in decimal format, with a "+" or "-" sign **Octal**: The data will be displayed in octal format **Hexadecimal**: The data will be displayed in hexadecimal format **BCD**: The data will be displayed in BCD code format
	Figure Length	Used to specify the maximum number of digits that will be displayed, including any digits following the decimal point. Note that if the length of the data is greater than the value that is set, the data will be displayed as "*". E.g., if the **Figure Length** is set to 2, the result for a value of "25.7" will be displayed as "**"
	Fill Zero	Used to specify whether the padding zeroes would be added if the length of the data is less than the **Figure Length**. E.g., if the **Figure Length** is set to 4, the result for a value of "25.7" will be displayed as "025.7"

- Apparel Properties

  The **Apparel Properties** dialog box is used to configure the display format for the **Recipe Numeric** object.

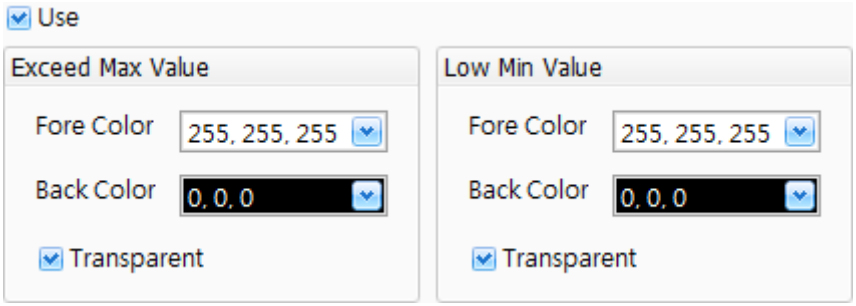


  The following is an overview of the options available in the **Apparel Properties** section of the Property View for the **Recipe Numeric** object.

Fore Color	Used to specify the color for the text
Back Color	Used to specify the background color for the object
Text Align	Used to specify the position of the text in relation to the border of the object
Font	Used to specify the type and size of the font to be used for the text.
Transparent	Used to specify whether or not the background color will be shown as transparent when displayed on the screen

- Range Properties

  The **Range Properties** dialog box is used to configure how the format of the **Recipe Numeric** object will change if the maximum or minimum values are exceeded.

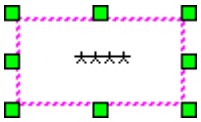


The following is an overview of the options available in the **Range Properties** section of the Property View for the **Recipe Numeric** object.

Use		Used to enable the Range function
Exceed Max. Value	**Fore Color**	Used to specify the color for the text
	Back Color	Used to specify the background color for the object
	Transparent	Used to specify whether or not the background color will be shown as transparent when displayed on the screen
Low Min. Value	**Fore Color**	Used to specify the color for the text
	Back Color	Used to specify the background color for the object
	Transparent	Used to specify whether or not the background color will be shown as transparent when displayed on the screen

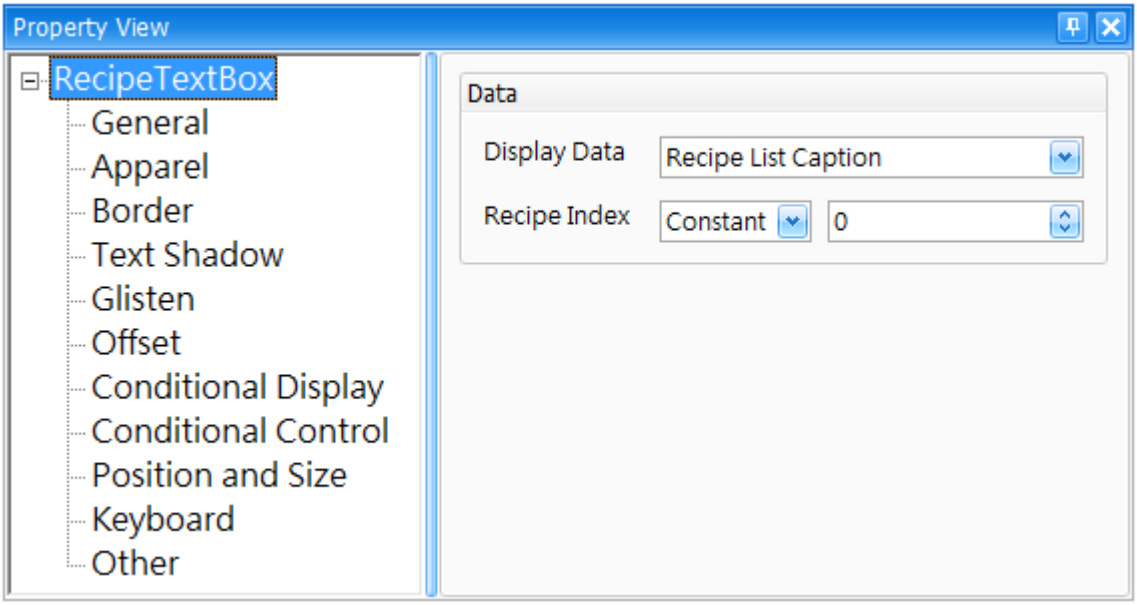
- See Section 8.5 Common Attributes for details about the others attributes of the **Recipe Numeric** object. Border (1), Text Shadow, Glisten, Security, Offset, Conditional Display, Conditional Control, Position and Size, Keyboard, and Other.

9.2.2 Recipe Text



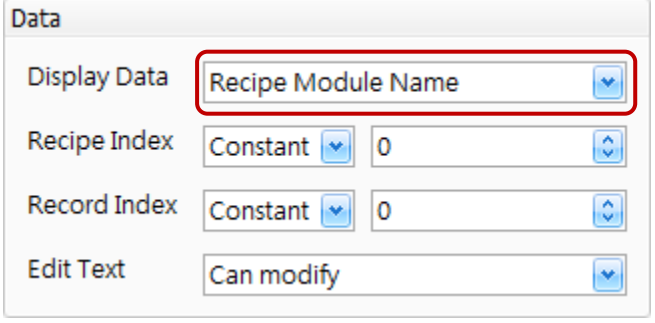
Before using the **Recipe Text** object, you need to configure the **Recipe** function in the **Project View** panel, see Chapter 9 for more details. This object is used to display the Recipe Caption (e.g., Recipe_01, see Section 9.1.1), the Element Caption (e.g., Width, see Section 9.1.2), or the Record Name (e.g., B5, see Section 9.1.3) which can be configured to allow or forbid the display name changed. The recipe will then be updated based on the data that has been modified, and the data will still be retained even after the system has been powered off.

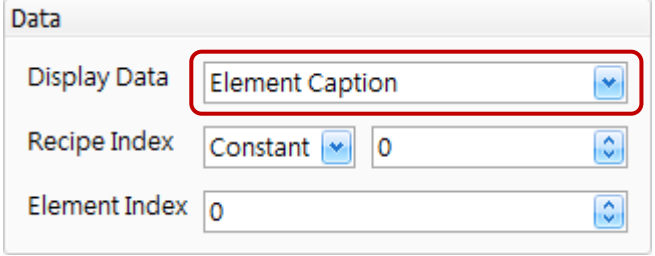
See Section 9.2 Recipe Objects for details of how to create an object. After creating a **Recipe Text** object, the **Property View** panel for the object will be automatically displayed allowing the various attributes of the object to be configured. The following is a description of how to configure the properties for the **Recipe Text** object.



- General Properties

  The **General Properties** dialog box is used to specify the caption or name to be displayed for the **Recipe Text** object.



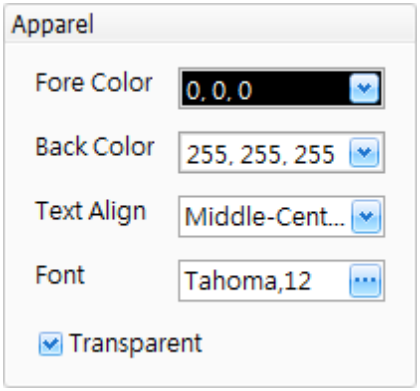


The following is an overview of the options available in the **General Properties** section of the Property View for the **Recipe Text** object.

Display Data	Used to select the data to be displayed, which can be the recipe caption, the element caption or the record name
Recipe Index / Record Index	Used to assign an index number for the Recipe or the Record, which can be either a constant or a variable Note that if the index number is invalid, the data will be displayed as blank
Element Index	Used to assign an index number for the Element Note that if the index number is invalid, the data will be displayed as blank
Edit Text	Used to specify whether or not the display name is allowed to be modified To edit the name, click the object on the HMI screen and then enter a new name

- Apparel Properties

  The **Apparel Properties** dialog box is used to configure the display format for the **Recipe Text** object.



The following is an overview of the options available in the **Apparel Properties** section of the Property View for the **Recipe Text** object.

Fore Color	Used to specify the color for the text
Back Color	Used to specify the background color for the object
Text Align	Used to specify the position of the text in relation to the border of the object
Font	Used to specify the type and size of the font to be used for the text.
Transparent	Used to specify whether or not the background color will be shown as transparent when displayed on the screen

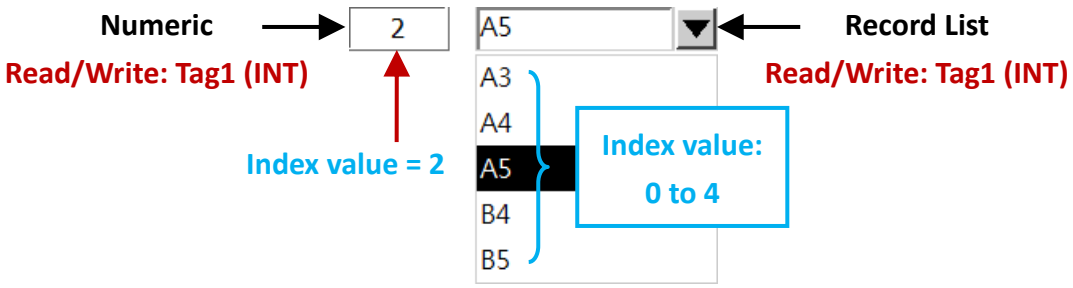
- See Section 8.5 Common Attributes for details about the others attributes of the **Recipe Text** object. Border (1), Text Shadow, Glisten, Offset, Conditional Display, Conditional Control, Position and Size, Keyboard, and Other.

9.2.3 Recipe List / Record List

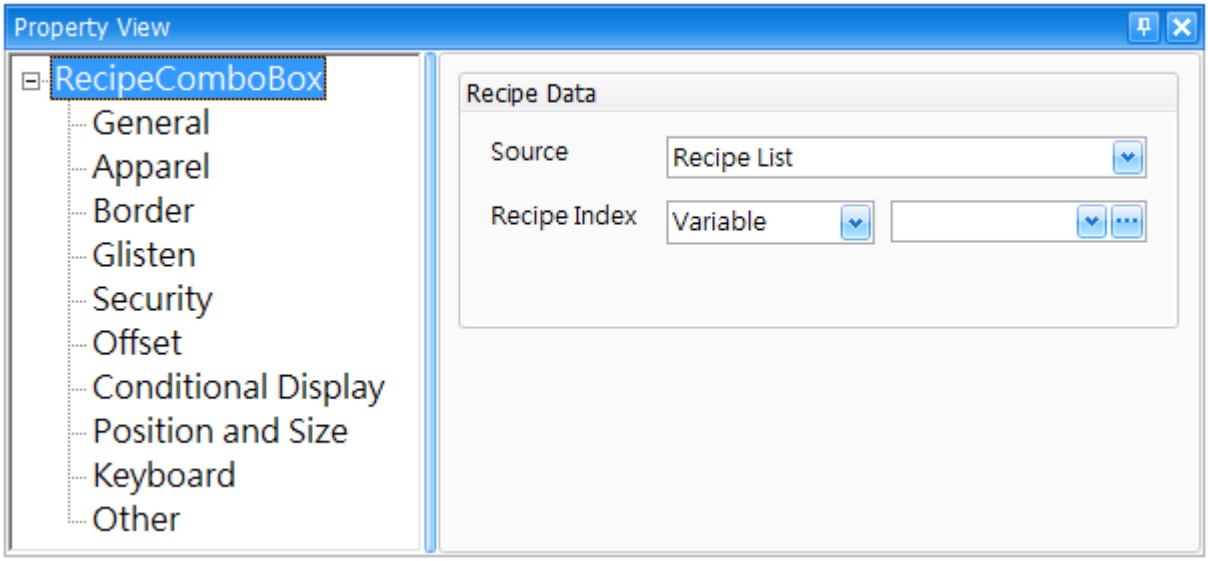
Before using the **Recipe List/Record List** object, you need to configure the **Recipe** function in the **Project View** panel, see Chapter 9 for more details.

The **Recipe List/Record List** object is used to create a list of recipes or records. When a recipe or a record is selected from the list, its index value will be written to the specified variable. If the variable is set to a valid recipe/record index, the corresponding caption of the recipe (or the name of the record) will be displayed. However, if the index value is invalid, a blank row will be displayed.

For example, suppose there are five size of paper, including A3, A4, A5, B4 and B5, the index value of records is 0 to 4, and the record data has been created, see Section 9.1.3 Record Tab. The record list will be shown as the figure.



See Section 9.2 Recipe Objects for details of how to create an object. After creating a **Recipe List/Record List** object, the **Property View** panel for the object will be automatically displayed allowing the various attributes of the object to be configured. The following is a description of how to configure the properties for the **Recipe List/Record List** object.

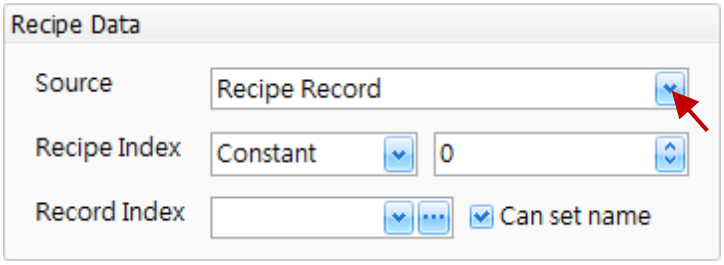


- General Properties

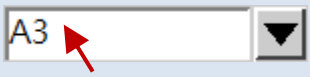
  The **General Properties** dialog box is used to configure the attributes for the **Recipe List/Record List** object. The attributes that will be displayed depend on whether the source is defined as either Recipe List or Record List.

The following is an overview of the options available in the **Recipe Data** dialog box in the **General Properties** section of the Property View for the **Recipe List/Record List** object when the **Recipe List** option is selected.

Source	Used to specify the display source, which can be either Recipe List or Recipe Record
Recipe Index	Used to assign an index number for the Recipe, which can be either a constant or a variable. Note that if the index is invalid, the data will be displayed as blank

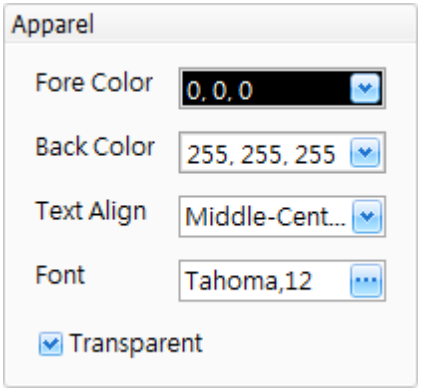


The following is an overview of the options available in the **Recipe Data** dialog box in the **General Properties** section of the Property View for the **Recipe List/Record List** object when the **Record List** option is selected.

Source	Used to specify the display source, which can be either Recipe List or Recipe Record
Recipe Index	Used to assign an index number for the Recipe, which can be either a constant or a variable. Note that if the index is invalid, the data will be displayed as blank
Record Index	Used to assign a variable for the Record. If one of the records is selected when the HMI device is operating, the index number will be written to this variable. Note that if the index is invalid, the data will be displayed as blank
Can set name	Used to specify whether or not the name of the record is allowed to be modified To edit the name, click the name and then enter a new name 

- Apparel Properties

  The **Apparel Properties** dialog box is used to configure the display format for the **Recipe List/Record List** object.

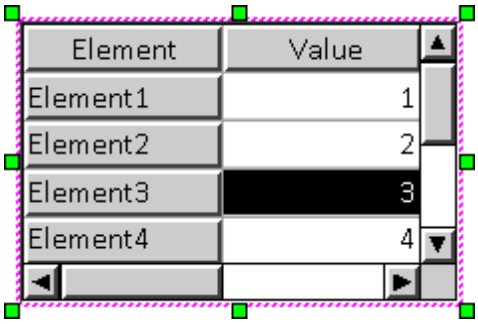


The following is an overview of the options available in the **Apparel Properties** section of the Property View for the **Recipe List/Record List** object.

Fore Color	Used to specify the color for the text
Back Color	Used to specify the background color for the object
Text Align	Used to specify the position of the text in relation to the border of the object
Font	Used to specify the type and size of the font to be used for the text.
Transparent	Used to specify whether or not the background color will be shown as transparent when displayed on the screen

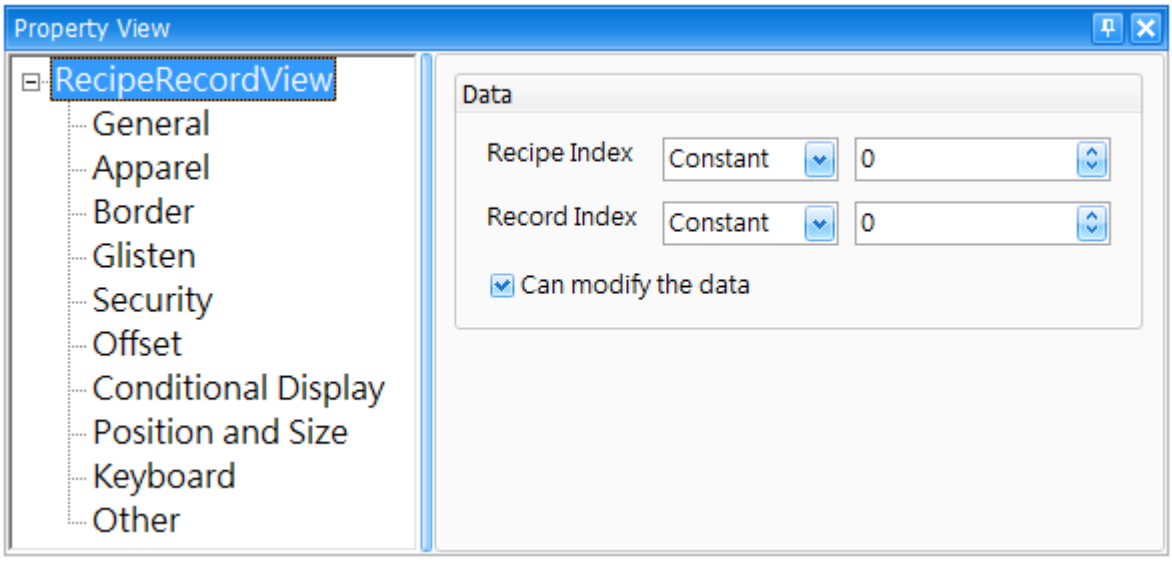
- See Section 8.5 Common Attributes for details about the others attributes of the **Recipe List/Record List** object. Border (1), Glisten, Security, Offset, Conditional Display, Position and Size, Keyboard, and Other.

9.2.4 Record View



The **Record View** object is used to view the data of all the Elements contained in a record in a table format. Once the Recipe and Record index numbers have been assigned, the data (see Section 9.1.1 for more details) will be automatically displayed. Click a Value displayed in the table allows the Element data to be directly modified.

See Section 9.2 Recipe Objects for details of how to create an object. After creating a **Record View** object, the **Property View** panel for the object will be automatically displayed allowing the various attributes of the object to be configured. The following is a description of how to configure the properties for the **Record View** object.



- General Properties

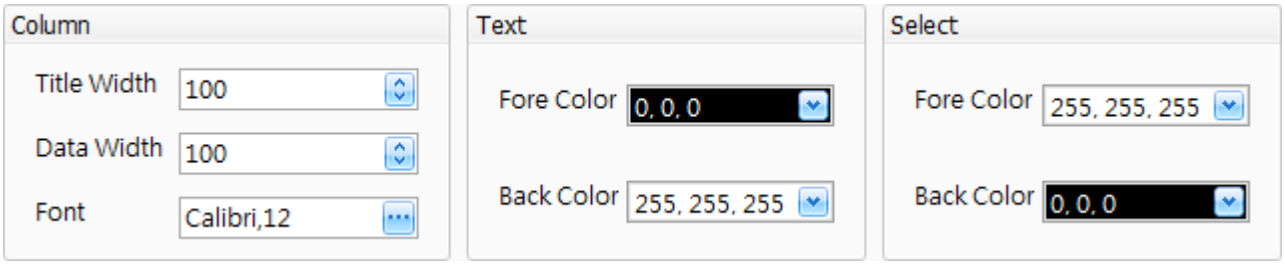
  The **General Properties** dialog box is used to configure the attributes for the **Record View** object.

The following is an overview of the options available in the **General Properties** section of the Property View for the **Record View** object.

Recipe Index	Used to assign an index number for the Recipe or the Record, which can be either a constant or a variable. Note that if the index is invalid, the data will be displayed as blank
Record Index	
Can modify the data	Used to specify whether or not the data is allowed to be modified To edit the data, click the item and then enter a new value.

- Apparel Properties

  The **Apparel Properties** dialog box is used to configure the display format for the **Record View** object.

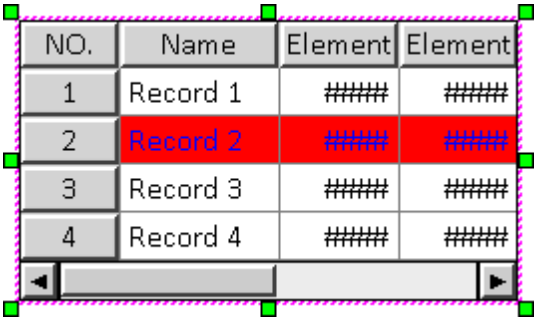


The following is an overview of the options available in the **Apparel Properties** section of the Property View for the **Record View** object.

Column	**Title Width**	Used to specify the width of the Element column
	Data Width	Used to specify the width of the Value column
	Font	Used to specify the type and size of the font to be used for the text in the table
Text	**Fore Color**	Used to specify the text color to be used for the data
	Back Color	Used to specify the color to be used for the background of all data cells
Select	**Fore Color**	Used to specify the text color to be used for the selected data cell
	Back Color	Used to specify the color to be used for the background of the selected data cell

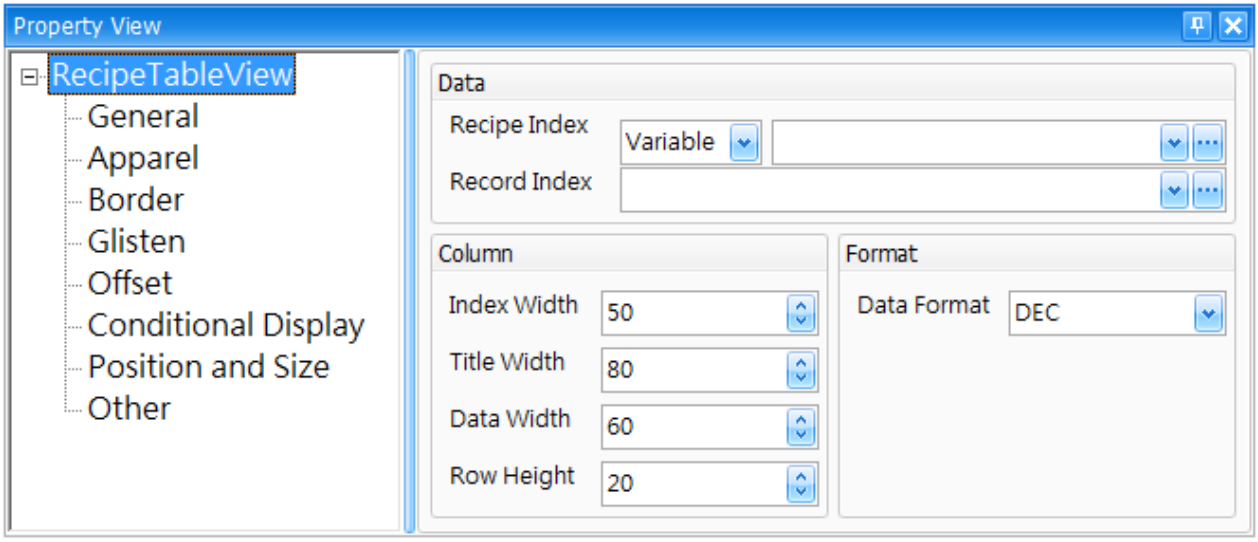
- See Section 8.5 Common Attributes for details about the others attributes of the **Record View** object. Border (1), Glisten, Security, Offset, Conditional Display, Position and Size, Keyboard, and Other.

9.2.5 Recipe Table View



The **Recipe Table View** object is used to view the data of all the Elements and Records contained in a Recipe in a table format. Once a valid Recipe index number has been assigned, the data related to the recipe will be automatically displayed. Clicking a Record displayed in the table allows the index number to be written to the relevant variable. Note that this object is only used to view the Recipe data which cannot be modified.

See Section 9.2 Recipe Objects for details of how to create an object. After creating a **Recipe Table View** object, the **Property View** panel for the object will be automatically displayed allowing the various attributes of the object to be configured. The following is a description of how to configure the properties for the **Recipe Table View** object.



- General Properties

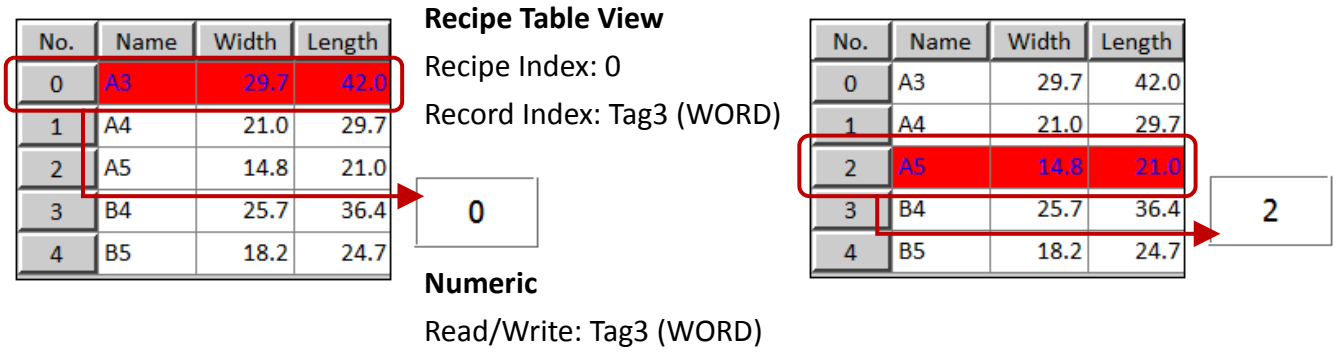
  The **General Properties** dialog box is used to configure the attributes and format of the displayed recipe for the **Recipe Table View** object.

The following is an overview of the options available in the **General Properties** section of the Property View for the **Recipe Table View** object.

Data	**Recipe Index**	Used to assign an index number for the Recipe, which can be either a constant or a variable. Note that if the index is invalid, the data will be displayed as blank
	Record Index	Used to retrieve the index number of the currently selected Record. When a record is selected, the index number for the Record will be written to the variable.
Column	**Index Width**	Used to specify the width of the column used for the index number

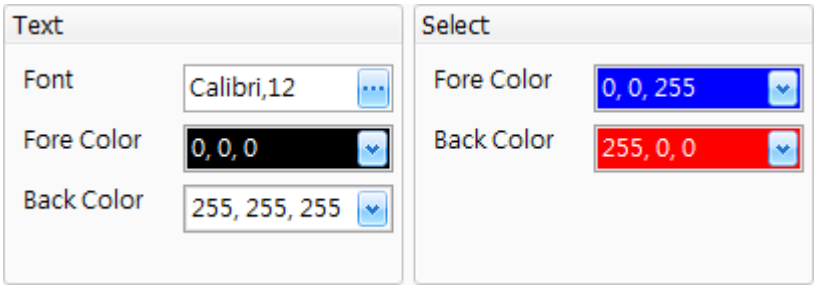
	Title Width	Used to specify the width of the column used for the name of the Record
	Data Width	Used to specify the width of the column used for the Recipe Element
	Row Height	Used to specify the height of each row. If the height is lower less than the size of the font, the size of the font will be used as the height of the row.
Format	**Data Format**	Used to configure the format used to display the data, which can be either decimal, Hexadecimal or BCD.

For example, we set the Recipe Index as “0”, see Section 9.1 for more details on how to create a new Recipe, and assign the variable “Tag3” with the data type “WORD” (see Chapter 7) to the Record Index field. In addition, we create the Numeric object to read/write the variable “Tag3”. If you select No. 0 in the Recipe table, the Numeric object will display as “0”, and if you select No. 2 in the **Recipe table**, the **Numeric** object will display as “2”.



- Apparel Properties

  The **Apparel Properties** dialog box is used to configure the display format for the **Recipe Table View** object.



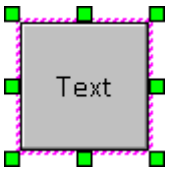
The following is an overview of the options available in the **Apparel Properties** section of the Property View for the **Recipe Table View** object.

Text	**Font**	Used to specify the type and size of the font to be used for the text in the table
	Fore Color	Used to specify the text color to be used for the data
	Back Color	Used to specify the color to be used for the background of all data rows
Select	**Fore Color**	Used to specify the text color to be used for the selected data row
	Back Color	Used to specify the color to be used for the background of the selected data row

- See Section 8.5 Common Attributes for details about the others attributes of the **Recipe Table View** object. Border (1), Glisten, Offset, Conditional Display, Position and Size, and Other.

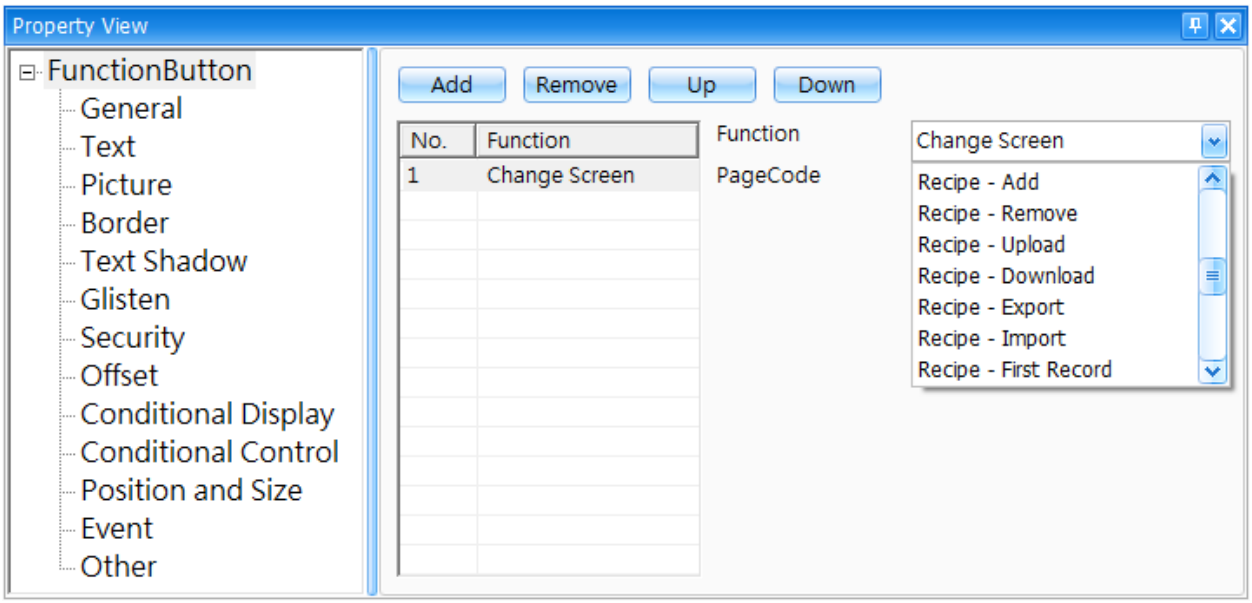
9.2.6 Recipe Function Button

Function Buttons can be used to execute a wide range of functions, such as transmitting recipes, changing screens, or incrementing and decrementing values, etc. When a button is pressed, any functions that have been defined will be executed sequentially, and a maximum of 16 functions can be implemented. The **Function Button** object is available in the each of the **Switch, Recipe, Alarm**, and **Sampling** categories.



The **Recipe Function Button** object provides the ability to add or remove a Recipe Record, upload or download a Recipe Record, import or export a Recipe, as well edit the Recipe Records, etc.

See Section 9.2 Recipe Objects for details of how to create an object. After creating a **Recipe Table View** object, the **Property View** panel for the object will be automatically displayed allowing the various attributes of the object to be configured. The following is a description of how to configure the properties for the **Recipe Table View** object.



- General Properties

  The **General Properties** dialog box is used to configure the actions that can be performed by the **Recipe Function Button** object.

The following is an overview of the options available in the **General Properties** section of the Property View for the **Recipe Function Button** object.

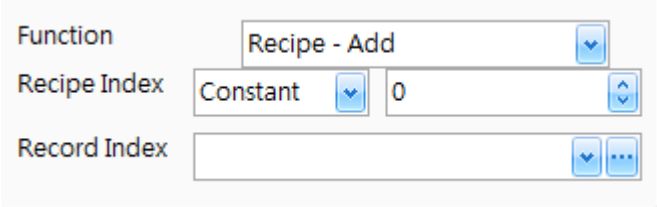
Add	Used to add a function. A maximum of up to 16 functions can be used for one button
Remove	Used to remove a selected function
Move Up	Used to move the selected item up
Move Down	Used to move the selected item down
Function	Used to select the function from the Function drop-down menu

Recipe Functions:

The following is an overview of the functions that can be used in conjunction with the **Recipe Function Button** object.

■ Recipe - Add

The **Recipe - Add** function is used to add a new Record in the specified Recipe, and then the index number of the Record will be written to the referenced variable.



The following is an overview of the options available for the **Recipe - Add** function.

Recipe Index	Used to assign an index number for the Recipe, which can be either a constant or a variable. See Section 9.1 Recipe Management for more details.
Record Index	Used to assign a variable for the Record. After clicking this function button, the index number will be written to this variable. See Chapter 7 Variables for more details.

For example, we set the Recipe Index as “0”, and then assign the variable “Tag6” with the data type “BYTE” to the Record Index field. As the following conditions, when you click the Recipe function button, a Recipe Record will be added, and the index number of the record will be written to the variable “Tag6”.

Recipe Table View
Recipe Index: 0
Record Index: Tag6 (BYTE)

No.	Name	Width	Length
0	A3	29.7	42.0
1	A4	21.0	29.7
2	A5	14.8	21.0
3	B4	25.7	36.4
4	B5	18.2	24.7
5	New For	0.0	0.0

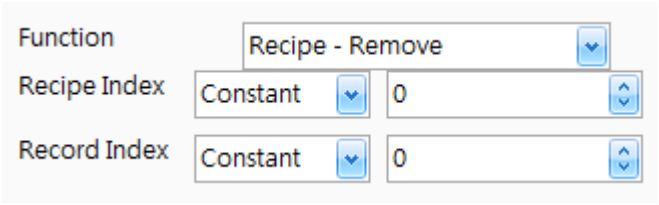
Recipe Function Button
Recipe Index: 0
Record Index: Tag6 (BYTE)

Numeric
Read: Tag6 (BYTE)

■ Recipe - Remove

The **Recipe - Remove** function is used to remove or delete the specified Recipe record.

The following is an overview of the options available for the **Recipe - Remove** function.



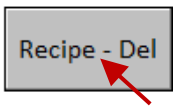
Recipe Index	Used to assign the index number of the Recipe to be used, which can be either a constant or a variable. See Section 9.1 Recipe Management for more details.
Record Index	Used to assign the index number of the Record to be removed, which can be either a constant or a variable. See Chapter 7 Variables for more details.

For example, we set the Recipe Index as “0”, and set the Record Index as “2”. When you click the Recipe function button, the Record (No.2) will be removed.

Recipe Table View
Recipe Index: 0

No.	Name	Width	Length
0	A3	29.7	42.0
1	A4	21.0	29.7
2	A5	14.8	21.0
3	B4	25.7	36.4
4	B5	18.2	24.7

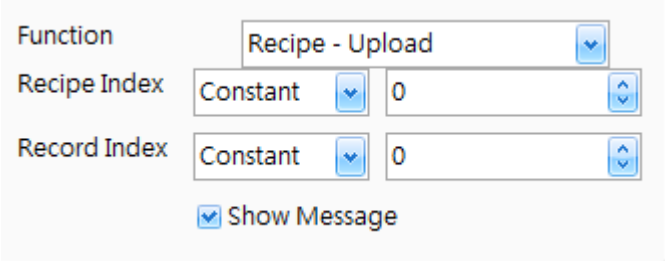
Recipe Function Button
Recipe Index: 0
Record Index: 2



No.	Name	Width	Length
0	A3	29.7	42.0
1	A4	21.0	29.7
2	B4	25.7	36.4
3	B5	18.2	24.7

■ Recipe - Upload

The **Recipe - Upload** function is used to upload the latest data from the connected device (refer to Chapter6 Connections and Section 7.2 Editing Variables) so that the current Recipe record can be updated.



The following is an overview of the options available for the **Recipe - Upload** function.

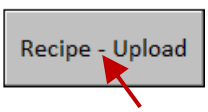
Recipe Index	Used to specify the index number of the Recipe to be used, which can be either a constant or a variable. See Section 9.1 Recipe Management for more details.
Record Index	Used to specify the index number of the Record to be updated, which can be either a constant or a variable. See Chapter 7 Variables for more details.
Show Message	Used to specify whether or not the confirmation dialog box will be displayed after you click the button

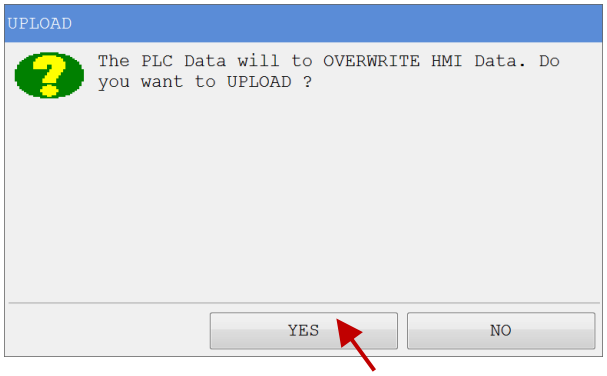
For example, we set the Recipe Index as “0”, set the Record Index as “3”, and check the **Show Message** checkbox. When you click the Recipe function button, the confirmation dialog box will be displayed. Make sure you want to overwrite the data, and click “Yes” button to update the Record (No.3).

Recipe Table View
Recipe Index: 0

No.	Name	Width	Length
0	A3	29.7	42.0
1	A4	21.0	29.7
2	A5	14.8	21.0
3	B4	25.7	36.4
4	B5	18.2	24.7

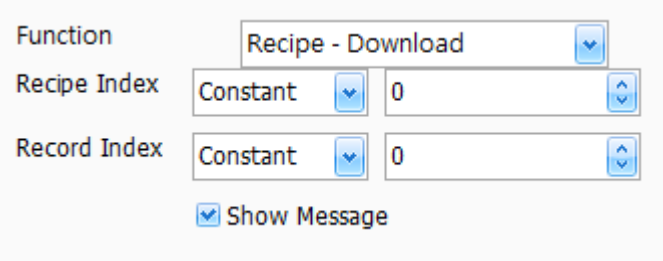
Recipe Function Button
Recipe Index: 0
Record Index: 3





■ Recipe - Download

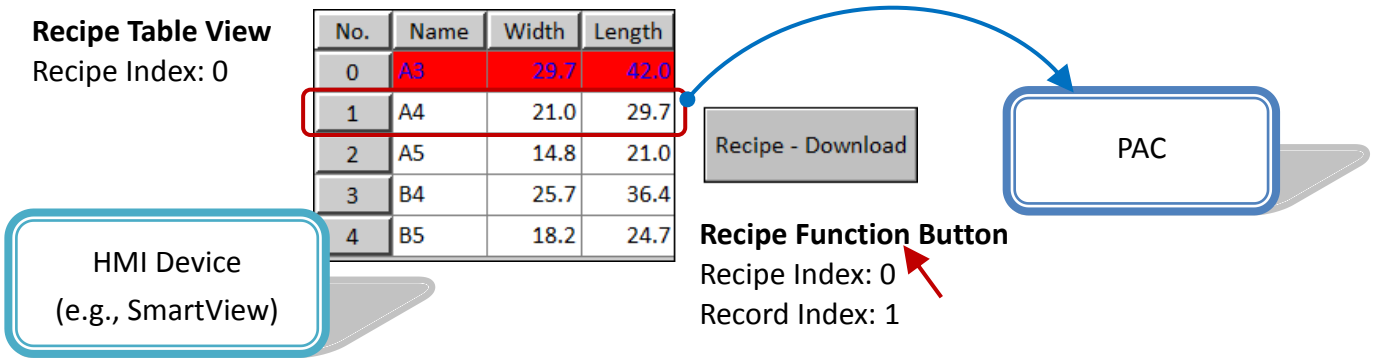
The **Recipe - Download** function is used to download the current data of the Recipe Record to the connected device (refer to Chapter6 Connections and Section 7.2 Editing Variables) so that it can be executed.



The following is an overview of the options available for the **Recipe - Download** function.

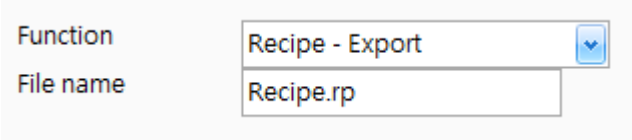
Recipe Index	Used to specify the index number of the Recipe to be used, which can be either a constant or a variable. See Section 9.1 Recipe Management for more details.
Record Index	Used to specify the index number of the Record to be downloaded, which can be either a constant or a variable. See Chapter 7 Variables for more details.
Show Message	Used to specify whether or not the confirmation dialog box will be displayed after you click the button

For example, we set the Recipe Index as “0”, and set the Record Index as “1”. When you click the Recipe function button, the Record (No.1) will be downloaded to the PAC.



■ Recipe - Export

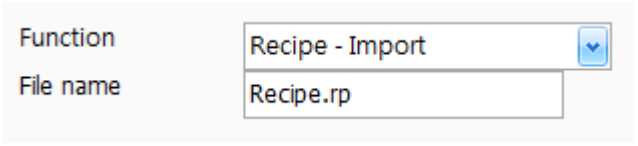
The **Recipe - Export** function is used to save all recipes as a file and then export it to the storage memory using the specified file name.



To use this function, select the **Recipe - Export** option from the Function drop-down menu, and then enter a name for the **File** in the File Name text field.

■ Recipe - Import

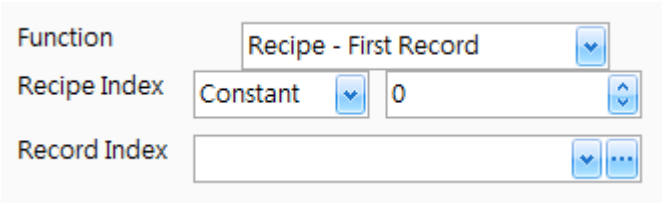
The **Recipe - Import** function is used to import a recipe file from the storage memory to replace the current Recipe(s), see Section 9.1 for more details about the Recipe(s).



To use this function, select the **Recipe - Import** option from the Function drop-down menu, and then enter a name for the **File** in the File Name text field.

■ Recipe – First Record

The **Recipe - First Record** function is used to select the first record, and the index number "0" will be written to the specified variable.



The following is an overview of the options available for the **Recipe - First Record** function.

Recipe Index	Used to specify the index number of the Recipe to be used, which can be either a constant or a variable. See Section 9.1 Recipe Management for more details.
Record Index	Used to specify the index number of the Record to be set, which can be either a constant or a variable. See Chapter 7 Variables for more details.

In this case, each time you click the Recipe function button, the first record will automatically be selected, and the variable (e.g., Tag6) will be set to 0.

Recipe Table View

Recipe Index: 0

No.	Name	Width	Length
0	A3	29.7	42.0
1	A4	21.0	29.7
2	A5	14.8	21.0
3	B4	25.7	36.4
4	B5	18.2	24.7

Recipe Function Button

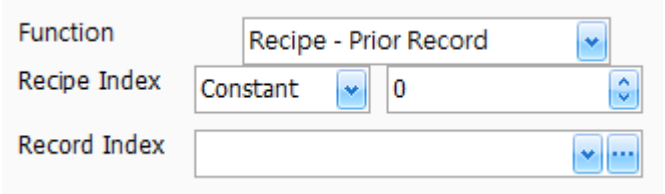
Recipe Index: 0

Record Index: Tag6 (BYTE)

No.	Name	Width	Length
0	A3	29.7	42.0
1	A4	21.0	29.7
2	A5	14.8	21.0
3	B4	25.7	36.4
4	B5	18.2	24.7

■ Recipe – Prior Record

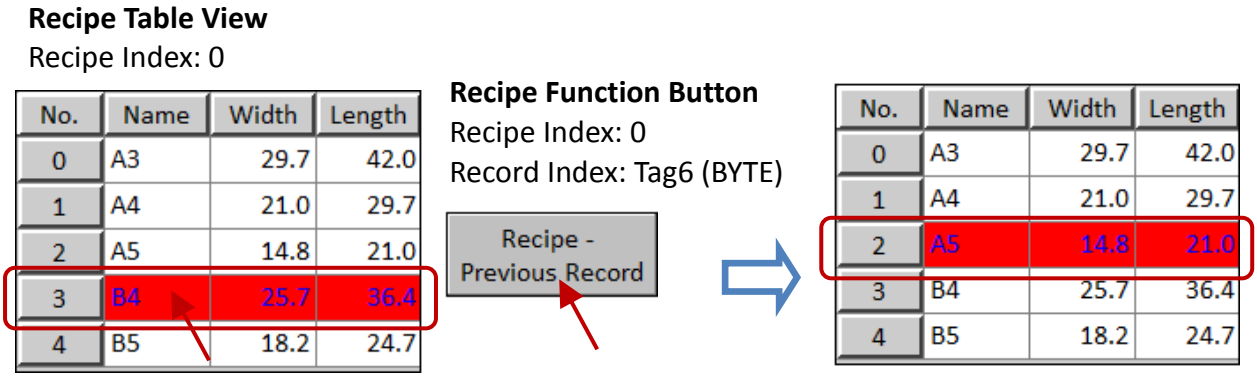
The **Recipe - Prior Record** function is used to select the previous record. If the value of the variable for the Record Index is greater than 0, then the value will be decremented by 1; if the value is less than or equal to 0, then the value will be set to 0.



The following is an overview of the options available for the **Recipe - Prior Record** function.

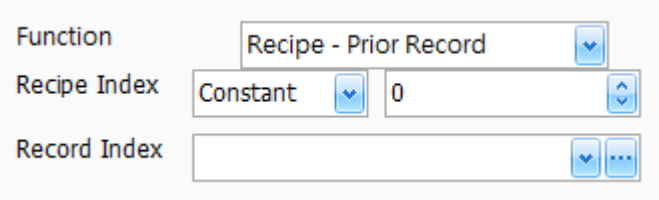
Recipe Index	Used to specify the index number of the Recipe to be used, which can be either a constant or a variable. See Section 9.1 Recipe Management for more details.
Record Index	Used to specify the index number of the Record to be set, which can be either a constant or a variable. See Chapter 7 Variables for more details.

In this case, when you click the Recipe function button, the previous record will automatically be selected, and the index number (e.g., 2) will be written to the variable (e.g., Tag6).



■ Recipe – Next Record

The **Recipe - Next Record** function is used to select the next record. If the value of the variable for the Record Index is less than the total number of Records, then the value will be incremented by 1; if the value is greater than the total number, the value will not be changed.



The following is an overview of the options available for the **Recipe - Next Record** function.

Recipe Index	Used to specify the index number of the Recipe to be set, which can be either a constant or a variable. See Section 9.1 Recipe Management for more details.
Record Index	Used to specify the index number of the Record to be set, which can be either a constant or a variable. See Chapter 7 Variables for more details.

In this case, when you click the Recipe function button, the next record will automatically be selected, and the index number (e.g., 4) will be written to the variable (e.g., Tag6).

Recipe Table View
Recipe Index: 0

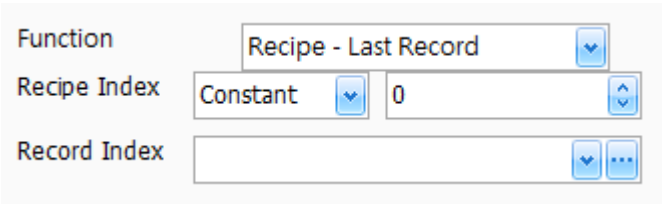
No.	Name	Width	Length
0	A3	29.7	42.0
1	A4	21.0	29.7
2	A5	14.8	21.0
3	B4	25.7	36.4
4	B5	18.2	24.7

Recipe Function Button
Recipe Index: 0
Record Index: Tag6 (BYTE)

No.	Name	Width	Length
0	A3	29.7	42.0
1	A4	21.0	29.7
2	A5	14.8	21.0
3	B4	25.7	36.4
4	B5	18.2	24.7

■ Recipe – Last Record

The **Recipe - Last Record** function is used to set the value of the variable for the Record Index to the maximum value for the Record Index.



The following is an overview of the options available for the **Recipe - Last Record** function.

Recipe Index	Used to specify the index number of the Recipe to be set, which can be either a constant or a variable. See Section 9.1 Recipe Management for more details.
Record Index	Used to specify the index number of the Record to be set, which can be either a constant or a variable. See Chapter 7 Variables for more details.

In this case, when you click the Recipe function button, the last record will automatically be selected, and the index number (e.g., 4) will be written to the variable (e.g., Tag6).

Recipe Table View
Recipe Index: 0

No.	Name	Width	Length
0	A3	29.7	42.0
1	A4	21.0	29.7
2	A5	14.8	21.0
3	B4	25.7	36.4
4	B5	18.2	24.7

Recipe Function Button
Recipe Index: 0
Record Index: Tag6 (BYTE)

No.	Name	Width	Length
0	A3	29.7	42.0
1	A4	21.0	29.7
2	A5	14.8	21.0
3	B4	25.7	36.4
4	B5	18.2	24.7

- See Section 8.5 Common Attributes for details about the others attributes of the **Recipe Function Button** object. Text, Picture, Border (2), Glisten, Security, Offset, Conditional Display, Conditional Control, Position and Size, Event, and Other.

9 – (B) Alarms (Available soon)

Creator provides the ability to configure an Alarm object so that if the condition of the device becomes abnormal, such as when the temperature being monitored is too high or too low, etc., an alarm will be triggered. An alarm notification will then be displayed via on-screen messages, which can be either a text message or a visual warning, to inform the on-site operators that a problem has occurred, and to assist in troubleshooting the error, while also providing a record of the messages.

Alarm Features:

1. Up to 1000 alarm records can be displayed
2. All alarm records are stored in the retained memory so that they will not be cleared when the device is rebooted
3. Both Digital and Analog alarms can be recorded
4. The alarm scan cycle is one second
5. Up to 5000 Digital and Analog alarm records can be configured
6. Two types of alarm messages (scrolling text or static text) can be used to inform the triggering alarm
7. After an alarm is triggered, the HMI screen can be automatically switched to the alarm screen
8. Alarm records can be exported to the storage memory in a CSV file format by using the Alarm Function Button object

There are two components to the **Alarm** functionality, including Alarm Function Management and Alarm Object usage, each of which is described in more detail below:

Alarm Function Management:

This component is used to configure a function to be used in conjunction with an alarm, and including digital and analog alarms, together with the alarm message settings, as described below:

Digital	Used to configure an alarm that will be triggered based on digital data
Analog	Used to configure an alarm that will be triggered based on analog data
Alarm Settings	Used to configure any message that will be displayed when an alarm is triggered

For more details related to the usage of Alarm functions, see Section 9.3 Digital Alarm Management, 9.4 Analog Alarm Management, and 9.5 Alarm Settings.

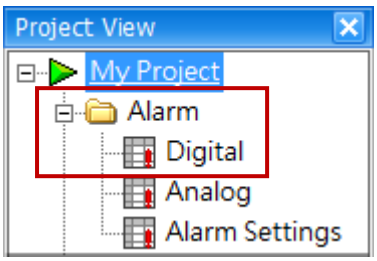
Alarm Objects:

This component is used to add an Alarm object to the HMI screen, and includes the Alarm View object and the Alarm Function Button object, which can be used to view or to export, or to remove Alarm records.

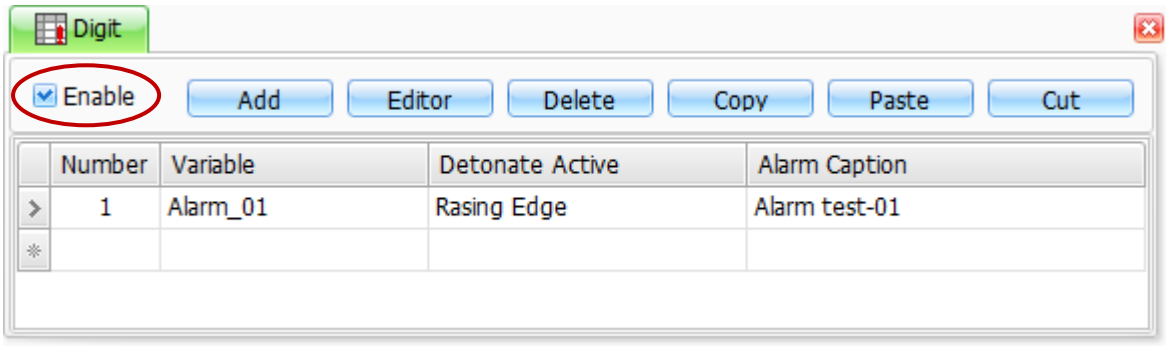
For more details related to the usage of Alarm objects, see Section 9.6 Alarm Objects.

9.3 Digital Alarm Management

The **Digital** function is used to configure an alarm that will be triggered based on digital data.



To open the **Digital** page, double-click the **Digital** item of the **Alarm** folder in the **Project View** panel. Check the **Enable** checkbox to allow the digital alarm functions to be added, edited, and managed.



The following is an overview of the options available on the **Digital** page.

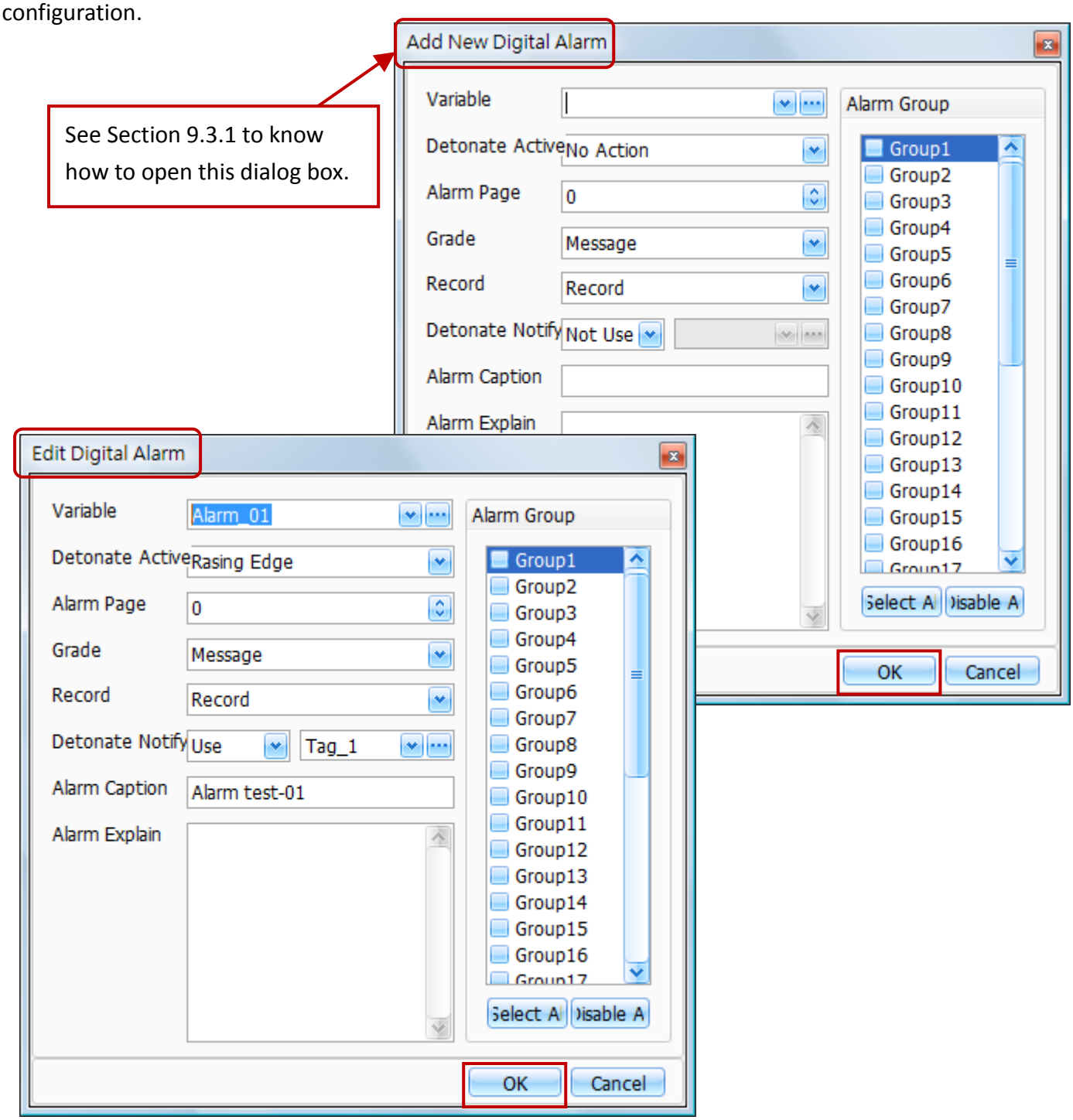
Enable	Used to enable the Digital Alarm function
Add	Used to add the digital alarm (See Section 9.3.1 “Adding a new Digital Alarm” for more details)
Editor	Used to edit the selected alarm (see Section 9.3.2 “Editing an existing Digital Alarm” for more details)
Delete	Used to delete the selected alarm
Copy	Used to copy the selected alarm data
Paste	Used to paste the alarm data that was copied or cut
Cut	Used to cut the selected alarm data

9.3.1 Adding a new Digital Alarm

The **Add New Digital Alarm** dialog box is used to configure a new Digital Alarm. As illustrated in the figure above, click the **Add** button on the **Digital** page to open the **Add New Digital Alarm** dialog box. Refer the following table to enter the details for the relevant attributes in the respective fields, and then click the **OK** button to save the new Alarm.

9.3.2 Editing an existing Digital Alarm

The **Edit Digital Alarm** dialog box is used to modify the attributes for an existing Digital Alarm. As illustrated in Section 9.3, click the **Editor** button on the **Digital** page to open the **Edit Digital Alarm** dialog box. Edit any of the details for the relevant atttributes as necessary, and then click the **OK** button to save the new configuration.

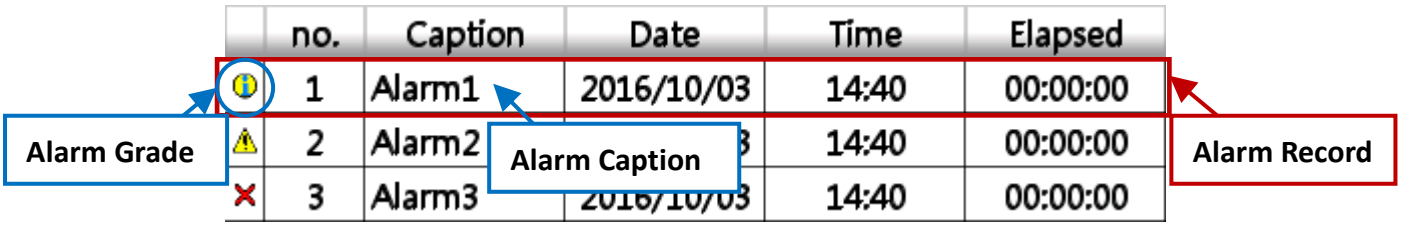


The following is an overview of the options available in either the **Add New Digital Alarm** or the **Edit Digital Alarm** dialog box.

Variable	Used to specify the variable that will be used to control the Alarm. For more details related to the usage of variables, see the Variables Section.

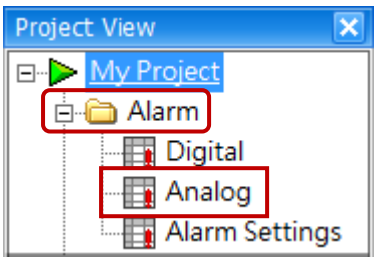
Detonate Active	Used to specify the trigger conditions for the Alarm, where: **No Action**: No alarm actions will be triggered **Rising Edge**: The Alarm will be triggered when the value of the variable changes from 0 to 1 **Falling Edge**: The Alarm will be triggered when the value of the variable changes from 1 to 0 **Rising and Falling Edges**: The Alarm will be triggered when the value of the variable changes from either 1 to 0 or from 1 to 0 **High Level (ON)**: The Alarm will be triggered when the value of the variable is 1 **Low Level (OFF)**: The Alarm will be triggered when the value of the variable value is 0 For more details related to the usage of variables, see the Variables Section.
Alarm Page	Used to assign a specific HMI screen to be displayed if an alarm is triggered. Setting a value of 0 means that the screen will not be changed.
Grade	Used to identify the alarm level using an icon, which will be displayed on the Alarm View object, and can be defined as: **Message** ( ), **Warning** ( ), and **Error** ( )
Record	Used to specify whether or not the triggered alarm will be recorded in the Alarm View object, see Section 9.6.1 for more details
Detonate Notify	Used to specify whether or not a notification will be issued when the alarm is triggered, which is configured using a variable. For more details related to the usage of variables, see the Variables Section
Alarm Caption	Used to specify a caption for the alarm which will be displayed on the Alarm View object
Alarm Explain	Used to provide an explanation of the alarm, which will be displayed when the alarm record is double-clicked
Alarm Group	Used to add this alarm to the specified group(s)

The following is an Alarm View object; see Section 9.6.1 for more details.

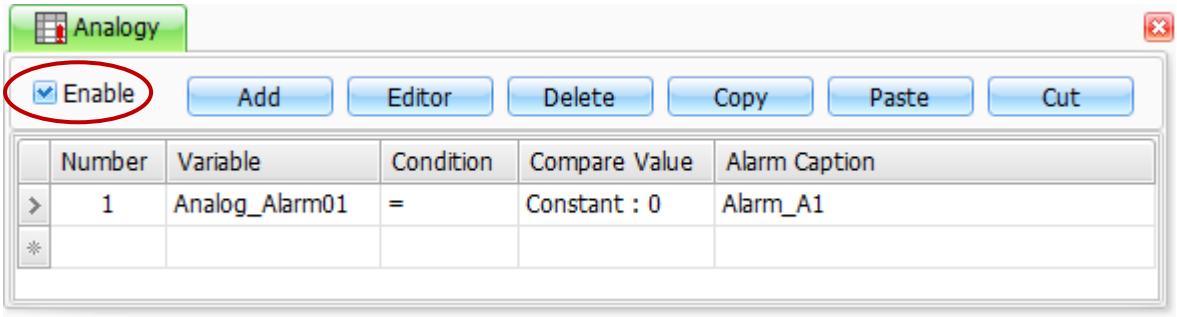


9.4 Analog Alarm Management

The **Analog** function is used to configure an alarm that will be triggered based on analog data.



To open the **Analog** page, double-click the **Analog** item of the **Alarm** folder in the **Project View** panel. Check the **Enable** checkbox to allow the digital alarm functions to be added, edited, and managed.



The following is an overview of the options available on the **Analog** page.

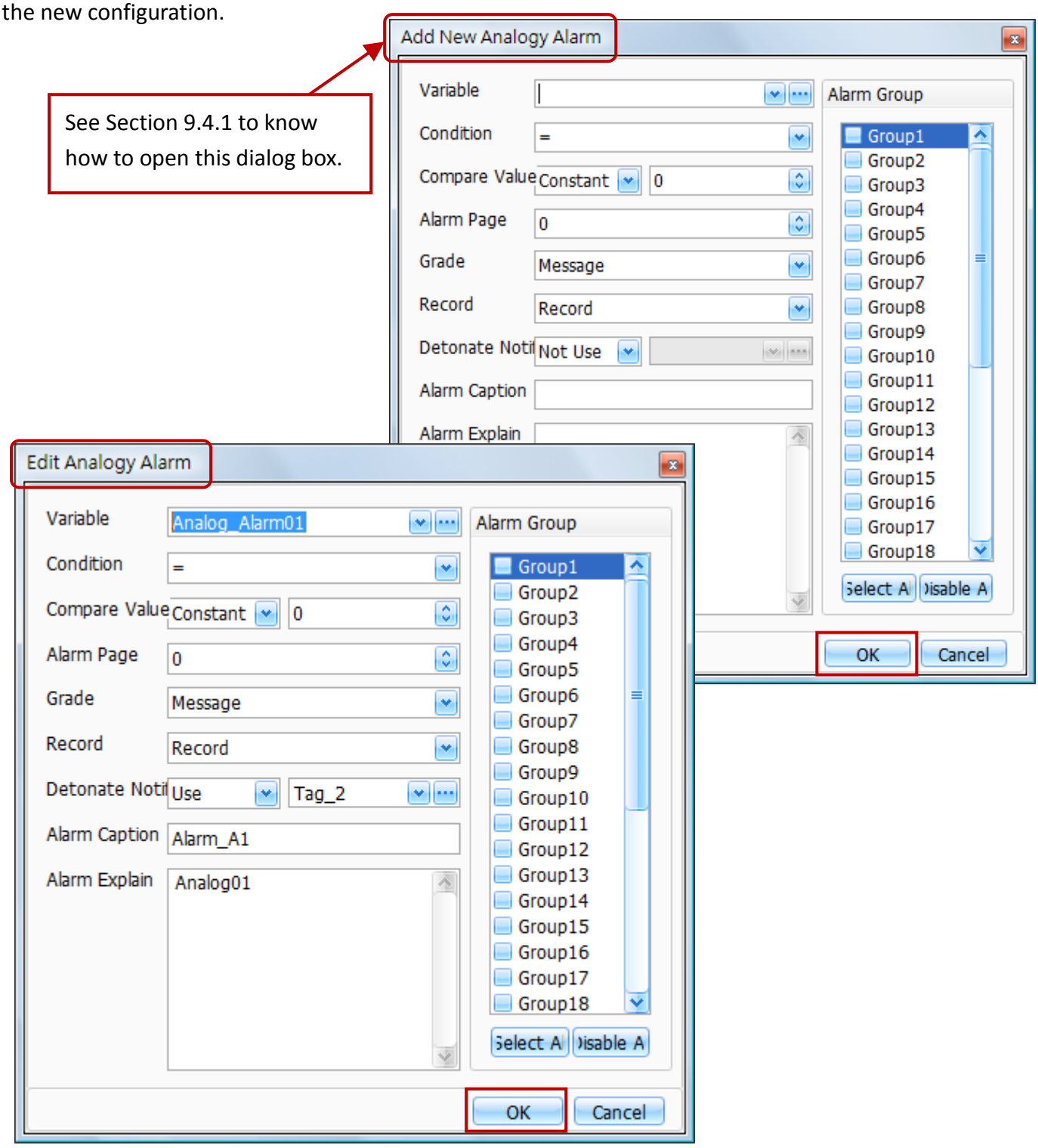
Enable	Used to enable the Analog Alarm function
Add	Used to add the analog alarm (See Section 9.4.1 “Adding a new Analog Alarm” for more details)
Editor	Used to edit the selected alarm (see Section 9.4.2 “Editing an existing Analog Alarm” for more details)
Delete	Used to delete the selected alarm
Copy	Used to copy the selected alarm data
Paste	Used to paste the alarm data that was copied or cut
Cut	Used to cut the selected alarm data

9.4.1 Adding a new Analog Alarm

The **Add New Analog Alarm** dialog box is used to configure a new Analog Alarm. As illustrated in the figure above, click the **Add** button on the **Analog** page to open the **Add New Analog Alarm** dialog box. Enter the details for the relevant attributes in the respective fields, and then click the **OK** button to save the new Alarm.

9.4.2 Editing an existing Analog Alarm

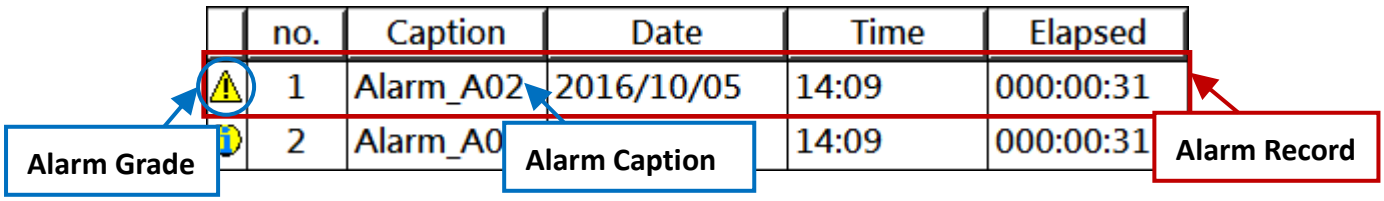
The **Edit Analog Alarm** dialog box is used to modify the attributes for an existing Analog Alarm. As illustrated in Section 9.4, click the **Editor** button on the **Analog** page to open the **Edit Analog Alarm** dialog box. Edit any of the details for the relevant atttributes as necessary, and then click the **OK** button to save the new configuration.



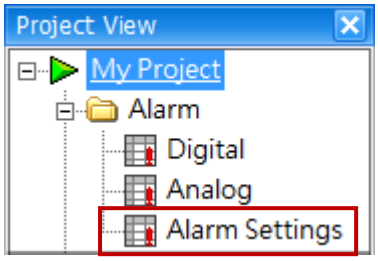
The following is an overview of the options available in either the **Add New Analog Alarm** or the **Edit Analog Alarm** dialog box.

Variable	Used to specify the variable that will be used to control the alarm. For more details related to the usage of variables, see the Variables Section.
Condition	Used to specify the method used to perform the comparison of values, where: **=**: An alarm will be triggered if the value of the variable is equal to the value being compared **>**: An alarm will be triggered if the value of the variable is larger than the value being compared **>=**: An alarm will be triggered if the value of the variable is larger than or equal to the value being compared **<**: An alarm will be triggered if the value of the variable is smaller than the value being compared **<=**: An alarm will be triggered if the value of the variable is smaller than or equal to the value being compared
Compare Value	Used to specify the value to be used to compare the condition action, which is configured using either a constant or a variable. For more details related to the usage of variables, see the Variables Section.
Alarm Page	Used to assign a specific HMI screen to be displayed if an alarm is triggered. Setting a value of 0 means that the screen will not be changed.
Grade	Used to identify the alarm level using an icon, which will be displayed on the Alarm View object, and can be defined as: **Message** ( ), **Warning** ( ), and **Error** ( )
Record	Used to id whether or not the message will be recorded in the Alarm View object, see Section 9.6.1 for more details
Detonate Notify	Used to specify whether or not a notification will be issued when the alarm is triggered, which is configured using a variable. For more details related to the usage of variables, see the Variables Section
Alarm Caption	Used to specify a caption for the alarm which will be displayed on the Alarm View object
Alarm Explain	Used to provide an explanation of the alarm, which will be displayed when the alarm record is double-clicked
Alarm Group	Used to add this alarm to the specified group(s)

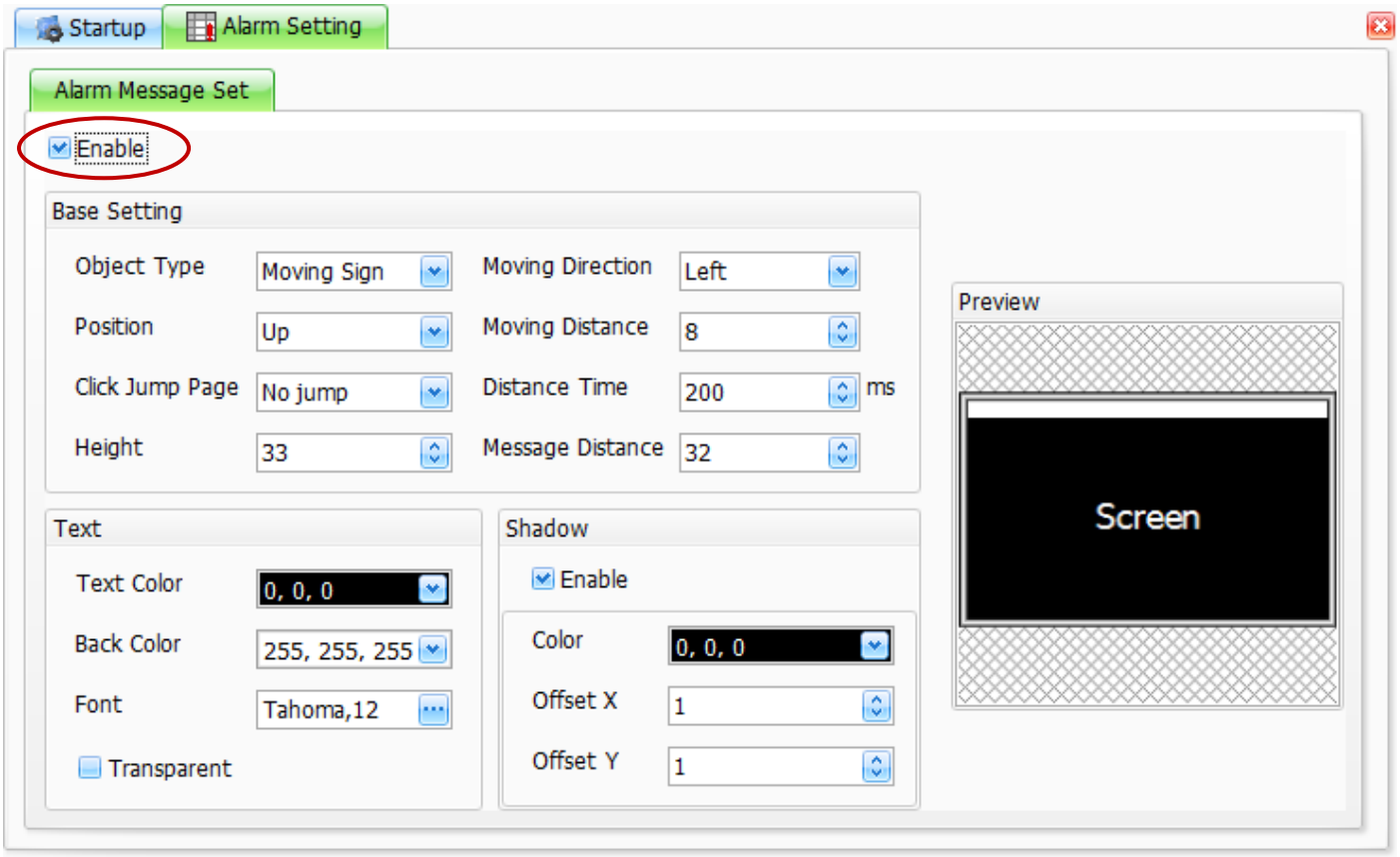
The following is an Alarm View object; see Section 9.6.1 for more details.



9.5 Alarm Settings



To open the Alarm Settings page, double-click the **Alarm Settings** item of the **Alarm** function in the **Project View** panel. Check the **Enable** checkbox to activate the attributes that can be used to configure the display style for the alarm message. Note that the message will not be displayed again when the triggering **condition** of the alarm is not satisfied (see Section 9.3 and 9.4).

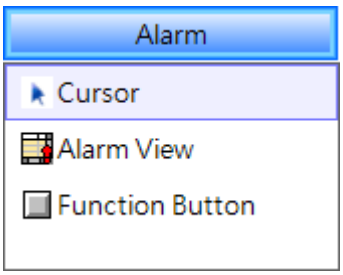


The following is an overview of the options available in the **Alarm Settings** page.

Enable		Used to enable or disable the alarm message
Base Settings	**Object Type**	Used to specify the type of display, where: **Static**: The alarm message will be displayed as a static text **Scrolling**: The alarm message will be displayed as a scrolling text
	Position	Used to specify the position of the message and can be either up or down
	Click Jump page	Used to specify whether or not the HMI will display an assigned page when the alarm is triggered Note that all page numbers created in the Screen View panel will be automatically listed in the drop-down menu.

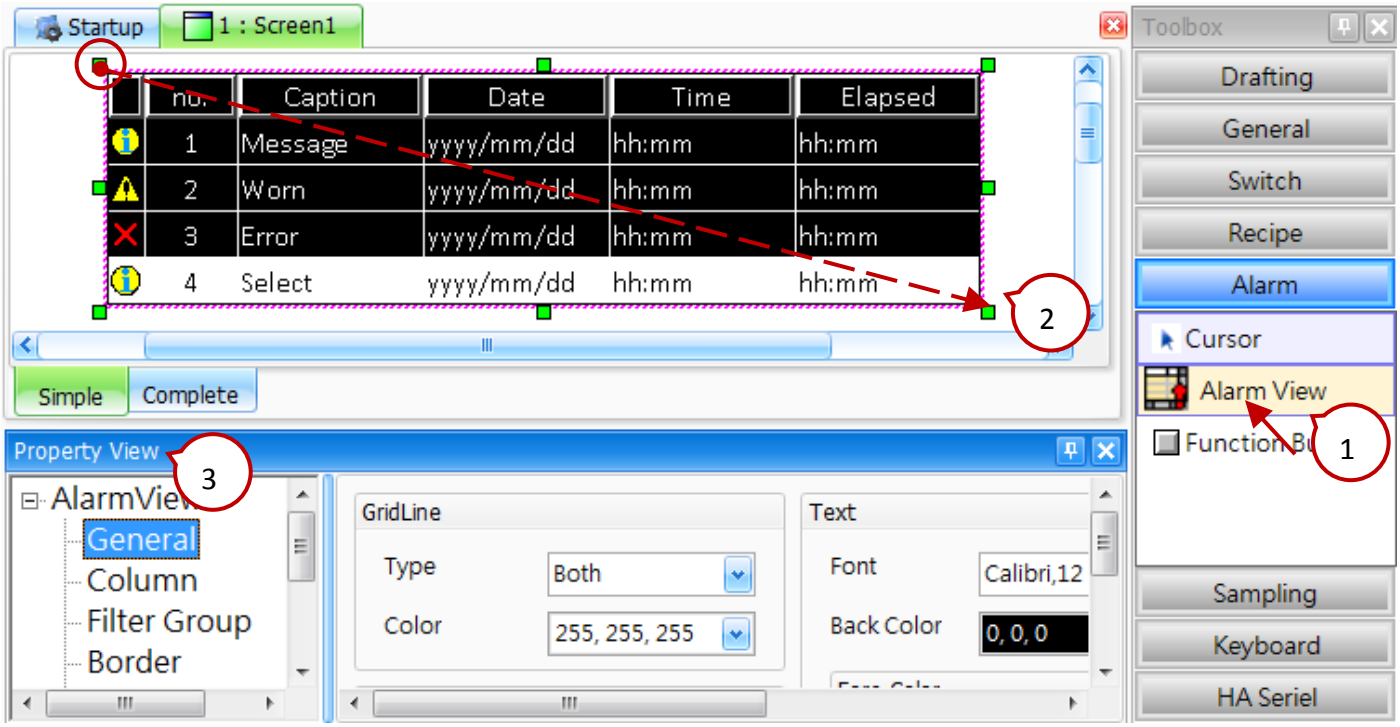
Base Settings	**Height**	Used to specify the distance of the message in relation to the screen
	Moving Direction	Used to specify the scroll direction for the message, and can be selected from Left, Right, Up, and Down. Note that this option is only applicable when the Object Type attribute is set to Scrolling.
	Moving Distance	Used to specify the distance the message will move in pixels. Note that a maximum of 32 pixels can be set, and this option is only applicable when the Object Type attribute is set to Scrolling.
	Distance Time	Used to specify the interval between movements, in milliseconds. Note that a maximum of 10000 ms can be set, and this option is only applicable when the Object Type attribute is set to Scrolling.
	Message Distance	Used to specify the amount of blank space between each instance that the message is displayed. Note that a maximum of 64 pixels can be set, and this option is only applicable when the Object Type attribute is set to Scrolling.
Text	**Text Color**	Used to specify the color to be used for the alarm message text
	Back Color	Used to specify the color of the background to the alarm message text
	Font	Used to specify the type and size of the font to be used for the alarm message text
	Transparent	Used to specify whether or not the background color will be set to transparent
Shadow	**Enable**	Used to specify whether the text shadow attribute is enabled or disabled
	Color	Used to specify the color to be used for the text shadow
	Offset X	Used to specify how far the text shadow will be displaced in the horizontal direction in pixels
	Offset Y	Used to specify how far the text shadow will be displaced in the vertical direction in pixels
Preview		Used to provide a preview of the alarm message text

9.6 Alarm Object



The **Alarm** category of the Toolbox includes a variety of objects that can be used to display the alarm and any messages on the HMI screen, or to export or remove the alarm record(s), each of which are described in more detail in the following sections.

To create an object, first click the name of the desired object in the **Alarm** category of the Toolbox. Position the mouse cursor on the Screen Design Area, and then click and hold the left mouse button to drag the object until it is the desired size and shape, as illustrated in the diagram below, and then release the mouse button. After creating a object, the **Property View** panel for the object will be automatically displayed allowing the various attributes of the object to be configured.



The **Function Button** object is provided in each of the Switch, Recipe, Alarm and Sampling categories and can be used to execute up to 16 functions, such as transmitting recipes, changing screens, or incrementing and decrementing variable values, etc. When the button is clicked, the functions that have been defined will be executed in sequence. The functions that can be defined for the **Alarm Function Button** object include:

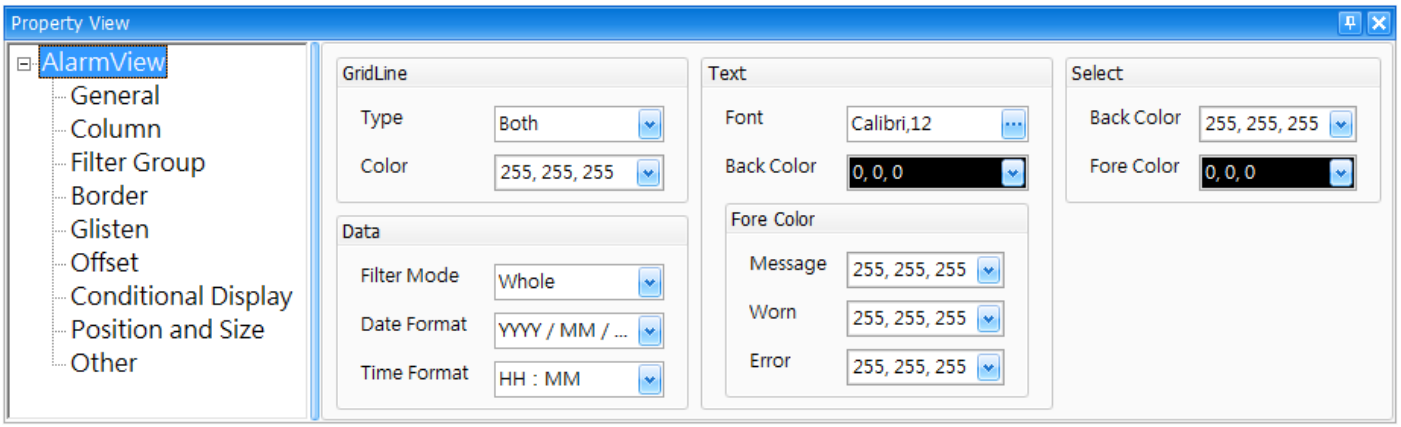
- Export Alarm
- Remove Alarm

9.6.1 Alarm View

Before using the **Alarm View** object, you need to configure the Alarm function in the Project View panel; see 9-(B) Alarm section for more details. This object can be used to display a list of alarm records in the form of a table, including the type of alarm, the time and date of the alarm, as well as the duration of the alarm.

	No.	Caption	Date	Time	Elapsed
ⓘ	1	Alarm_D1	2016/10/06	14:58	00:01:15
⚠	2	Alarm_A1	2016/10/06	14:59	00:00:26
✕	3	Alarm_A2	2016/10/06	14:59	00:00:10

See Section 9.6 Alarm Objects for details of how to create an object. After creating a **Alarm View** object, the **Property View** panel for the object will be automatically displayed allowing the various attributes of the object to be configured. The following is a description of how to configure the properties for the **Alarm View** object.



- General Properties

  The **General Properties** dialog box is used to configure the display attributes for the **Alarm View** object, including the color and type for the table borders, and the format of the data.

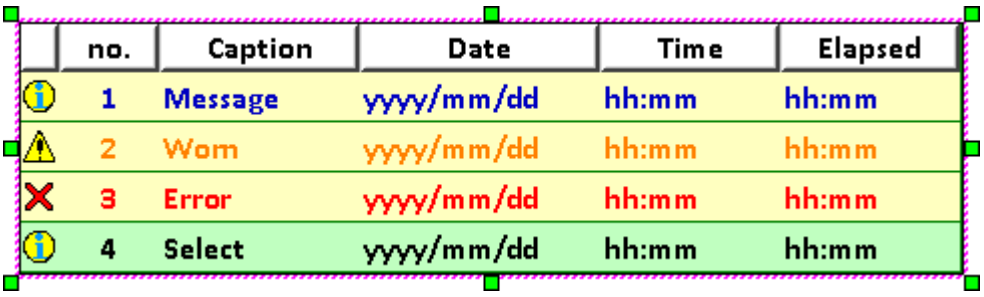
The following is an overview of the options available in the **General Properties** section of the Property View for the **Alarm View** object.

Borders	**Type**	Used to specify borders to the alarm table, where: 1. **Nil**: No Border 2. **Horizontal:** Add all horizontal borders 3. **Vertical:** Add all vertical borders 4. **Both:** Add all boards
	Color	Used to specify the color of the borders for the alarm table

Data	**Filter Mode**		Used to specify the filter mode for displaying specific types of alarm, where: 1. **Whole**: Displays all alarm records 2. **Show Today**: Displays only today's alarm records 3. **Show Touch off**: Displays only those records that still meet the triggering condition
	Date Format		Used to specify the format of the date, where: 1. yyyy/mm/dd (Year, Month, Day) 2. dd/mm/yyyy (Day, Month, Year) 3. mm/dd/yyyy (Month, Day, Year)
	Time Format		Used to specify the format of the time, where: 1. HH:MM (Hours:Minutes) 2. HH:MM:SS (Hours:Minutes:Seconds)
Text	**Font**		Used to specify the type and size of the font to be used for the text
	Back Color		Used to specify the background color for alarm records
	Fore Color	**Message**	Used to specify the color of the text to be displayed for standard messages (see Section 9.3.2 or 9.4.2, the Grade option)
		Warning	Used to specify the color of the text to be displayed for warning messages (see Section 9.3.2 or 9.4.2, the Grade option)
		Error	Used to specify the color of the text to be displayed for error messages (see Section 9.3.2 or 9.4.2, the Grade option)
Select	**Back Color**		Used to specify the background color for the alarm record when it has been selected
	Fore Color		Used to specify the color of the alarm record when it has been selected

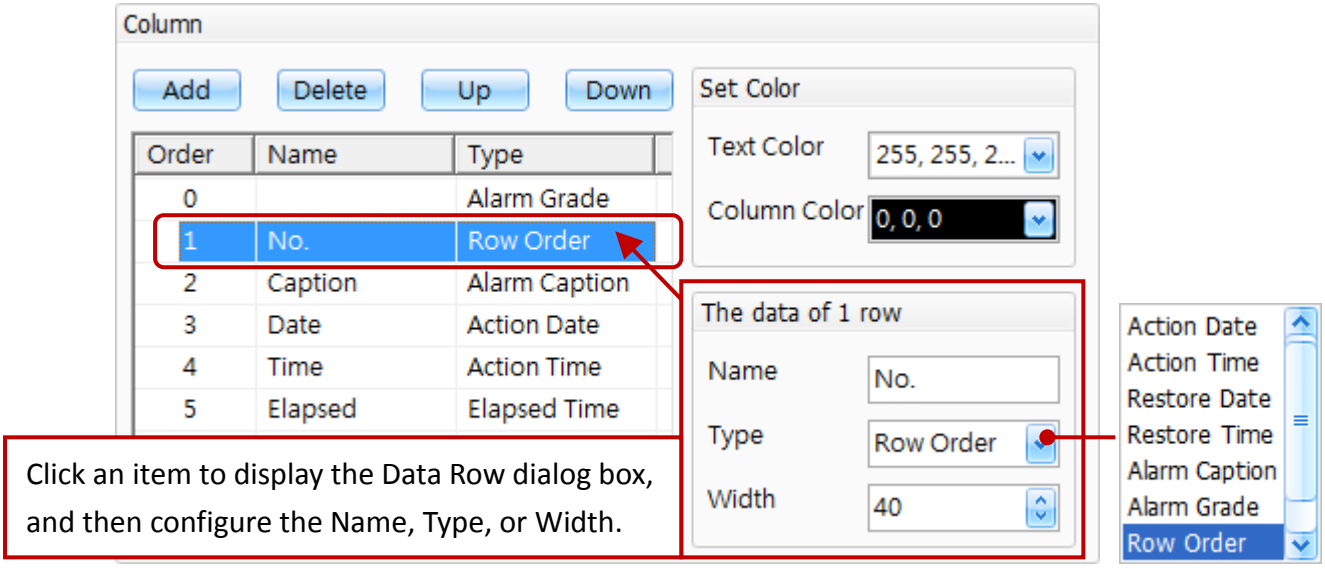
Give it a try: (Click the F8 key to perform the offline simulation)

- **Borders –** Type: Horizontal
  Borders – Color: Green
- **Select -** Background Color: Green
  Select - Foreground Color: Black
- **Text -** Foreground Color
  – Message: Blue
  – Warning: Orange
  – Error: Red



- Column Properties

  The **Column Properties** dialog box for the **Alarm View** object is used to configure the column headings and style, as well as to specify which kinds of data should be displayed in the alarm table.



The following is an overview of the options available in the **Column Properties** section of the Property View for the **Alarm View** object.

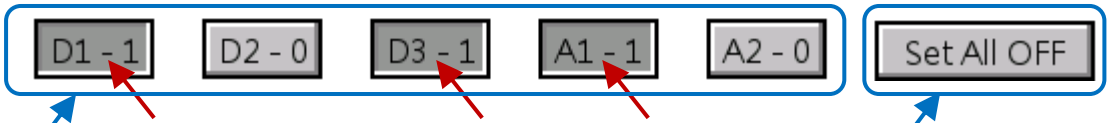
Column	**Add**	Used to add column to the alarm table (up to a maximum of 8)
	Delete	Used to delete the selected column(s) from the alarm table To select multiple column, click the first row you want to select, and hold down the Shift key (or the Ctrl key), and then click the last row (or click each of the other columns you want to select).
	Up	Used to move the selected data row up
	Down	Used to move the selected data row down
Set Color	**Text Color**	Used to specify the color of the text
	Column Color	Used to specify the background color to be used for the column
The Data of n Row	**Name**	Used to specify a name for the column, which will be used for the column headings
	Type	Used to specify the type of the column **Action Data:** Used to indicate the date that the alarm was triggered **Action Time:** Used to indicate the time that the alarm was triggered **Restore Date:** Used to indicate the date when the alarm was reset **Restore Time:** Used to indicate the time when the alarm was reset

The Data of n Row	Type	**Alarm Caption:** Used to display a caption for the alarm **Alarm Grade:** Used to display an icon for the different alarm types: (See Section 9.3.2 or 9.4.2 the Grade option) Message Warning Error **Row Order:** Used to indicate the order in which the alarm messages will be displayed **Elapsed Time:** Used to display the duration between when the alarm is triggered and when the alarm is reset **Alarm Value:** Used to display the value of the variable that will cause the alarm to be triggered
	Width	Used to specify the width of the selected column

Give it a try:

Digital Alarm / Analog Alarm:
Alarm Caption (Type, Variable):
Alarm_D1 (Message, D01), Alarm_D2 (Warning, D02), Alarm_D3 (Error, D03)
Alarm_A1 (Warning, A01), Alarm_A2 (Error, A02)

No.	Caption	Trigger Date	Trigger Time	Reset Time	Duration	Trigger Value
1	Alarm_D1	2016/10/12	11:04		00:02:27	1
2	Alarm_D3	2016/10/12	11:05	11:06	00:01:00	1
3	Alarm_A1	2016/10/12	11:05		00:01:33	1
4	Alarm_D3	2016/10/12	11:06		00:00:33	1



Bit Switch:
Write/Read (Bit): D01, D02, D03
Write/Read (WORD): A01, A02
Type: Toggle (Click to ON or OFF)

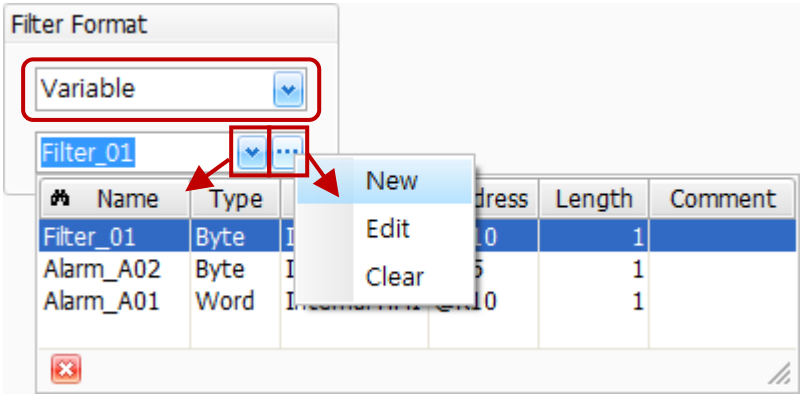
Function Button:
Function: Set OFF (5)
Write: D01, D02, D03, A01, A02

- Filter Group Properties

  The **Filter Group Properties** dialog box for the **Alarm Viewer** object is used to configure the filter for displaying alarm records and can be defined using either a group or a variable.

  When the **Group** Filter format is selected, a Dialog box containing a list of groups will be displayed. Select the required groups to filter the data based on that group. Note that you must specify the group number for each alarm first, see Section 9.3.2 or 9.4.2 for more details.

  When the **Variable** Filter format is selected, a dialog box will be displayed that allows the required variable to be selected. For more details related to the usage of variables, see Chapter 7 Variables.



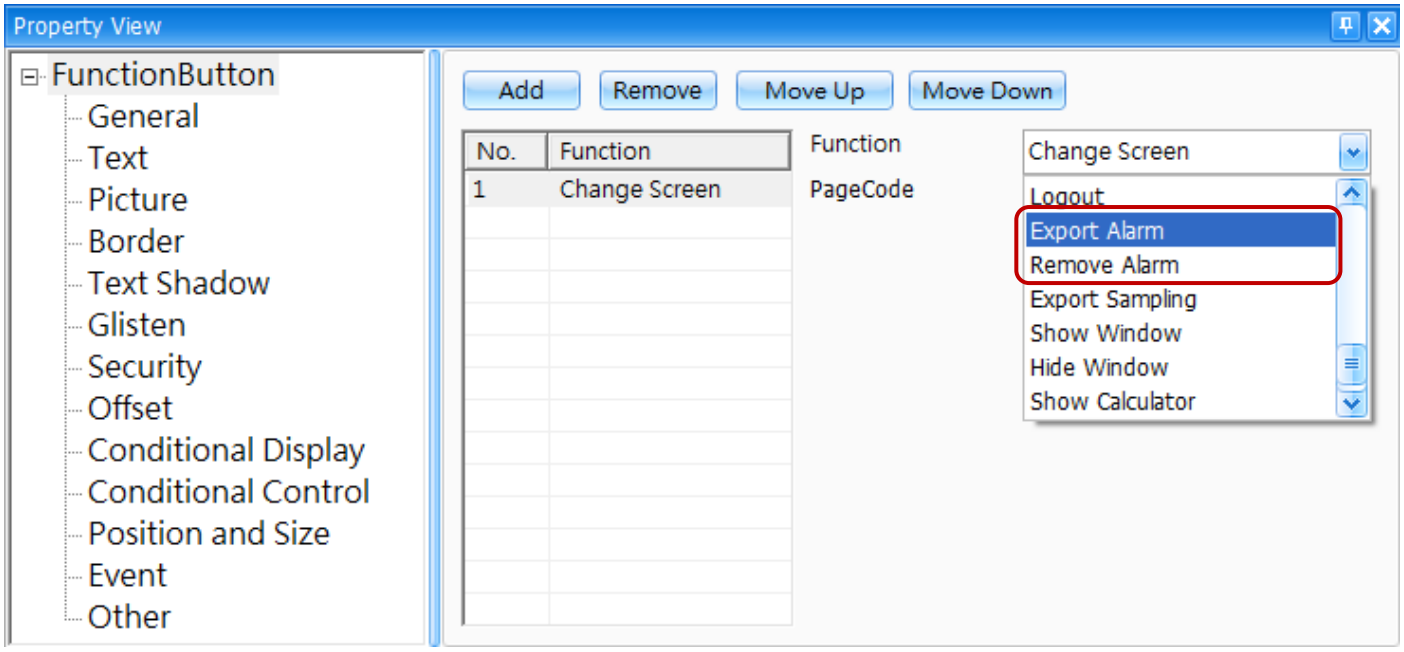
- See Section 8.5 Common Attributes for details about the others attributes of the **Alarm Viewer** object. Border (1), Glisten, Offset, Conditional Display, Position and Size, and Other.

9.6.2 Alarm Function Button

Function Buttons can be used to execute a wide range of functions, such as transmitting recipes, changing screens, or incrementing and decrementing values, etc. When a button is pressed, any functions that have been defined will be executed sequentially, and a maximum of 16 functions can be implemented. The Function Button object is available in the each of the Switch, Recipe, Alarm, and Sampling categories.

The **Alarm Function Button** object provides the ability to export an alarm or remove an alarm. See Section 9.6 Alarm Objects for details of how to create an object. After creating a **Alarm Function Button** object, the **Property View** panel for the object will be automatically displayed allowing the various attributes of the object to be configured.

The following is a description of how to configure the properties for the **Alarm Function Button** object.



- General Properties

  The **General Properties** dialog box is used to configure the actions that can be performed by the **Function Button** object.

The following is an overview of the options available in the **General Properties** section of the Property View for the **Function Button** object.

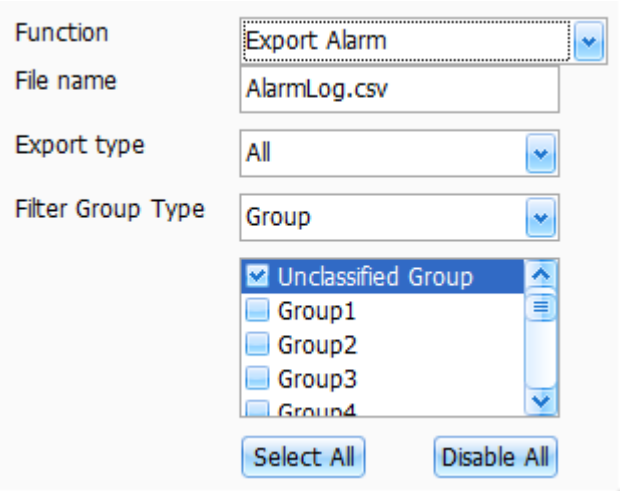
Add	Used to add a function A maximum of up to 16 functions can be used for one button
Remove	Used to remove a selected function
Move Up	Used to move the selected item up
Move Down	Used to move the selected item down
Function	Used to select the function from the Function drop-down menu

Alarm Functions:

The following is an overview of the functions that can be used in conjunction with the **Alarm Function Button** object.

■ Export Alarm

The **Export Alarm** function is used to export the alarm history records to the storage memory using the specified file name. The **Export Type** option allows the choice between exporting all records, or only the records for the current day.



The following is an overview of the options available for the **Export Alarm** function.

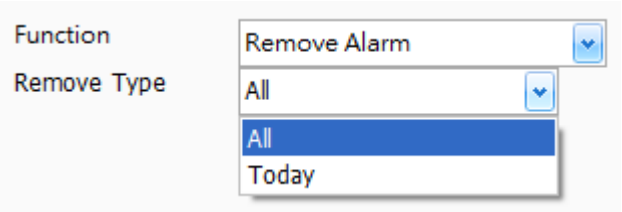
File Name	Used to specify the name to be used for the storage file
Export Type	Used to specify the type of records to be stored, and can be selected from: **All**: All alarms that have been logged since the device was started will be stored **Today**: Only the alarm records relating to the current date will be stored
Filter Group Type	Used to specify the way to store the records, and can be selected from: **Group:** The alarm records will be stored based on the selected group number Note that you must specify the group number for each alarm first, see Section 9.3.2 or 9.4.2 for more details. **Variable:** The alarm records will be stored based on the value of the variable. For more details related to the usage of variables, see the Variables Section
Select All	Used to select all Groups
Disable All	Used to deselect all currently selected Groups

To use this function, select the **Export Alarm** option from the Function drop-down menu, enter a name for the **File** in the File Name text field, select **All** records or **Today**'s records from the Export Type drop-down menu, and then select the **Group** way or the **Variable** way from the Filter Group Type drop-down menu.

If the Group option is selected, specify which Groups should be included in the Alarm History by selecting the required groups from the Groups section. If the **Variable** option is selected, select an existing variable from the drop-down menu, or click the **Browse** (…) button to add a new variable.

■ **Remove Alarm**

The **Remove Alarm** function is used to remove the alarm records that have been confirmed (or reset). The Remove Type option allows the choice between removing all confirmed records, or only removing those confirmed records for the current day.



To use this function, select the **Remove Alarm** option from the Function drop-down menu, and then select **All** records or **Today**'s records from the Remove Type drop-down menu. When you click this button on the HMI screen, all confirmed records or those confirmed records for the current day will be removed.

- See Section 8.5 Common Attributes for details about the others attributes of the **Alarm Function Button** object. Text, Picture, Border (2), Glisten, Security, Offset, Conditional Display, Conditional Control, Position and Size, Event, and Other.

9 – (C) Data Sampling

Data sampling is used to access the current values of specific events or attributes and can be configured to detonate at a fixed time or following a predefined event. After reading the values from the device, the data is stored in the internal memory, meaning that the data can be retained after the device is powered off.

Data Sampling Features:

1. Provides support for the **Trend Chart** object that can be used to display data using a line graph style.
2. Provides support for the **Data Sampling View** object that can be used to displays data values using a table style.
3. Provides the ability to perform **Data Sampling** using either **Time Detonation** or **Bit Detonation** modes.
4. Provides the ability to export sampling data to a storage memory in a CSV file format using the Data Sampling Function Button.
5. The total size of the memory allocated for the data sampling is 512 KB.

There are two components to the **Data Sampling** functionality, including Data Sampling Function Management and Data Sampling Object usage, each of which is described in more detail below:

Data Sampling Function Management:

This component is used to configure a Data Sampling function that can be used to create, design, and manage Data Sampling functionality.

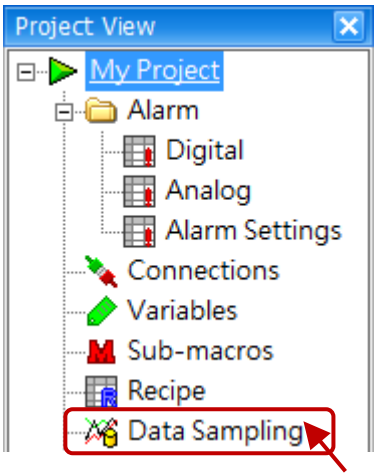
To access the **Data Sampling** page, double-click the **Data Sampling** option in the **Project View** panel.

Data Sampling Objects:

This component is used to add a Data Sampling object to the HMI screen, and includes options such as a Real-Time/History Trend Chart, the Data Sampling View, and the Data Sampling Function Button.

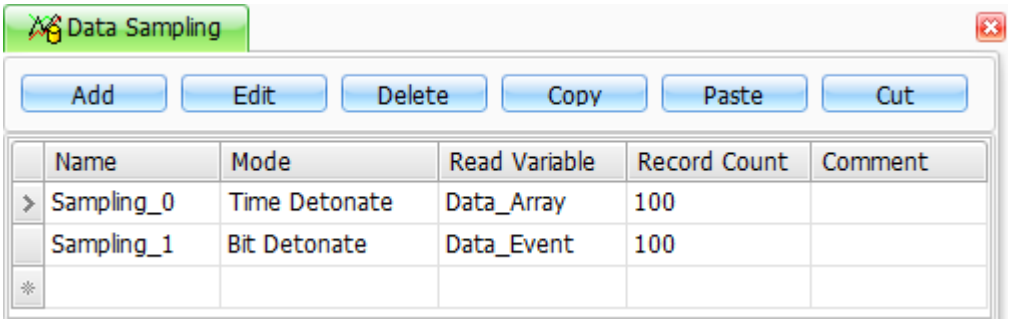
To access these **Data Sampling Objects**, click the name of the desired object in the **Sampling** category of the Toolbox. See Section 9.8 for more details.

9.7 Data Sampling Management



The Data Sampling function is used to configure the functions to be used in conjunction with the data sampling object, such as the sampling mode, the record mode, the overflow setting, etc.

To open the **Data Sampling** page, double-click the **Data Sampling** item in the **Project View** panel. On the **Data Sampling** page, you can create, edit and manage the Data Sampling function. Note that a maximum of 32767 data can be recorded, and the maximum size for each is 512 KB.



The following is an overview of the options available on the **Data Sampling** page.

Add	Used to add the Data Sampling Function (see Section 9.7.1 “Adding a new Data Sampling Function” for more details)
Edit	Used to edit the selected Data Sampling Function (see Section 9.7.2 “Editing an existing Data Sampling Function” for more details)
Delete	Used to delete the selected Data Sampling Function
Copy	Used to copy the selected Data Sampling Function data
Paste	Used to paste the Data Sampling Function data that was copied or cut
Cut	Used to cut the selected Data Sampling Function data

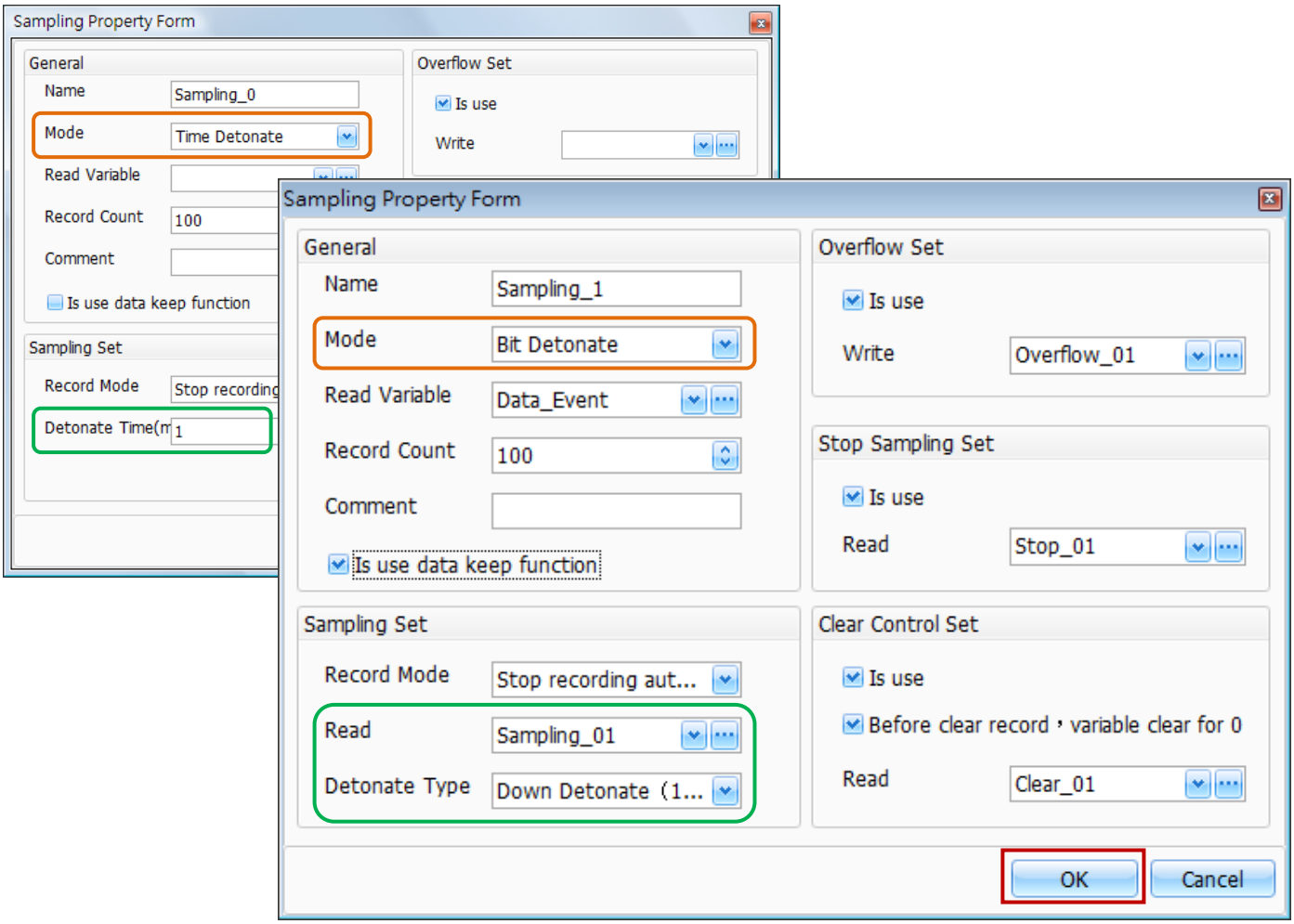
9.7.1 Adding a new Data Sampling Function

The **Sampling Property Form** dialog box is used to configure a new Data Sampling Function. As illustrated in the figure above, click the **Add** button on the **Data Sampling** page to open the **Sampling Property Form** dialog box.

Refer the following table in the Section 9.7.2 to enter the details for the relevant attributes in the respective fields, and then click the **OK** button to save the new Data Sampling Function.

9.7.2 Editing an existing Data Sampling Function

The **Sampling Property Form** dialog box is used to modify the attributes for an existing Data Sampling Function. As illustrated in Section 9.7, click the **Edit** button on the **Data Sampling Management** page to open the **Sampling Property Form** dialog box.

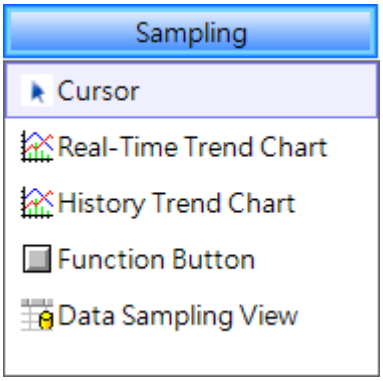


The following is an overview of the options available in the **Sampling Property Form** dialog box.

General	**Name**	Used to specify a name for the Data Sampling Function
	Mode	Used to specify the sampling method, where: **Time Detonate**: The data will be collected at a fixed period of time **Bit Detonate**: The data will be collected when the status of the specified Bit variable is TRUE, see the **Sampling Settings - Detonate Time** as below
	Read Variable	Used to specify the Read variable to be used, which can also be an array variable. See Chapter 7 Variables for more details.
	Record Count	Used to specify the maximum number of sampling records, which can be in the range of 1 to 32767
	Comment	Used to provide a description of the Data Sampling Function

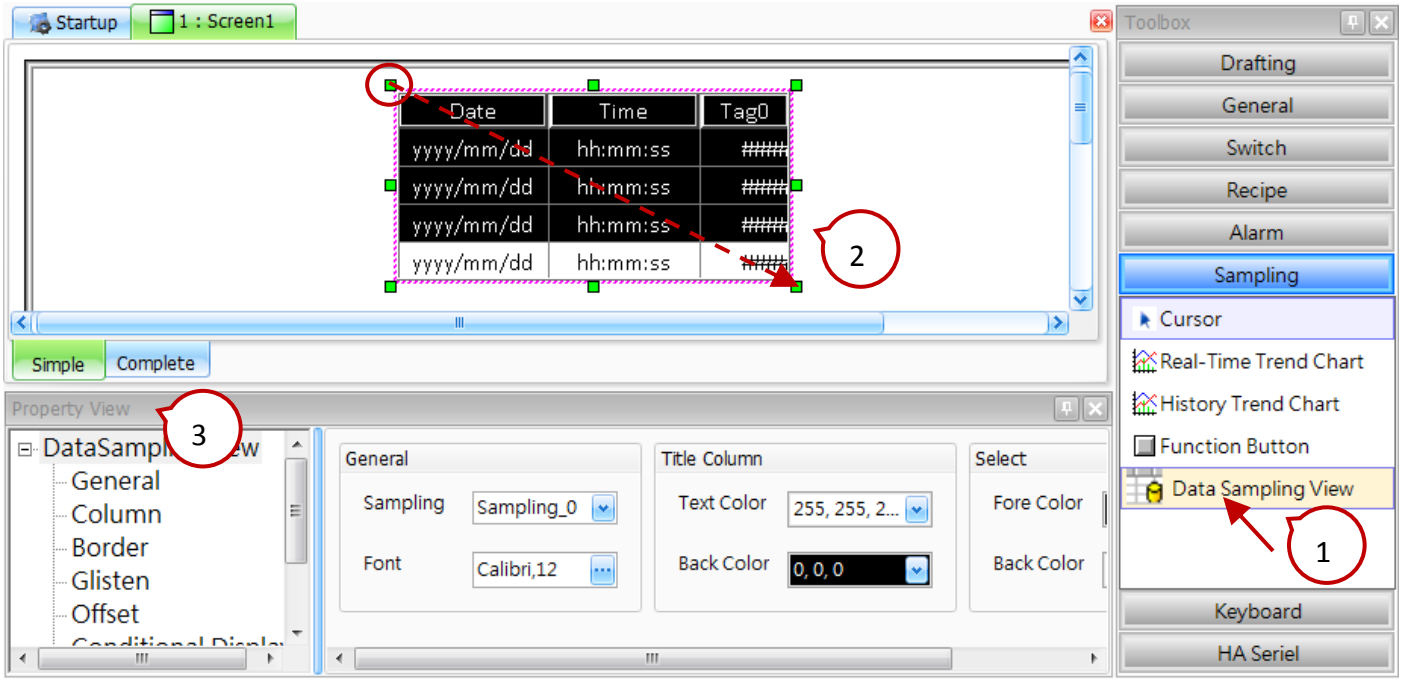
	Is use data keep function	Used to enable the Retain Data function so that the data can be retained after the system is powered off
Sampling Set	**Record Mode**	Used to specify the action to be taken if a data sampling overflow occurs: **Stop Recording Automatic:** The Data Sampling function will automatically stop when a data overflow occurs **First In, First Out (FIFO)**: When a data overflow occurs, the data sampling function will remove the earliest record and add the new record at the end of the list
	Detonate Time	Used to specify the sampling time when the Time Detonate option is selected as the Data Sampling mode
	Read	Used to specify the variable that is to be used to control the Data Sampling function when the Bit Detonate option is selected as the Data Sampling mode. See Chapter 7 Variables for more details.
	Detonate Type	Used to specify the triggered conditions for the Data Sampling Function when the Bit Detonate option is selected as the Data Sampling mode **Down Detonate**: The value of the Variable changes from 1 to 0 **Up Detonate**: The value of the Variable changes from 0 to 1 **Up/Down Detonate:** The value of the Variable changes from 0 to 1 or from 1 to 0. Only the Bit variable can be used, see Chapter 7 Variables for more details.
Overflow Set	**Is use**	Used to enable the Overflow control function
	Write	Used to set the Write variable to 1 when the number of data sampling records has reached the maximum. Only the Bit variable can be used, see Chapter 7 Variables for more details.
Stop Sampling Set	**Is use**	Used to enable the Stop Sampling control function
	Read	Used to specify that sampling should be stopped if the variable value is equal to 0. Only the Bit variable can be used, see Chapter 7 Variables for more details.
Clear Control Set	**Is use**	Used to enable the Clear control function
	Before clear record, variable clear for 0 Used to set the variable to 0 after removing the data sampling records	
	Read	Used to clear the data sampling records if the value of the variable is equal to 1. Only the Bit variable can be used, see Chapter 7 Variables for more details.

9.8 Sampling Objects



The Sampling Objects category of the Toolbox includes a variety of objects that can be used to display items such as Trend Charts and Tables on an HMI screen, and to record processes), each of which are described in more detail in the following sections.

To create an object, first click the name of the desired object in the **Sampling** category of the Toolbox. Position the mouse cursor on the Screen Design Area, and then click and hold the left mouse button to drag the object until it is the desired size and shape, as illustrated in the diagram below, and then release the mouse button. After creating a object, the **Property View** panel for the object will be automatically displayed allowing the various attributes of the object to be configured.



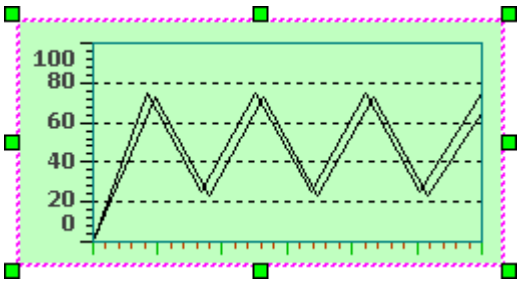
The Function Button object is provided in each of the Switch, Recipe, Alarm and Sampling categories and can be used to execute up to 16 functions, such as transmitting recipes, changing screens, or incrementing and decrementing variable values, etc. When the button is clicked, the functions that have been defined will be executed in sequence. The functions that can be defined for the **Sampling Function Button** object include:

- Export Sampling Data

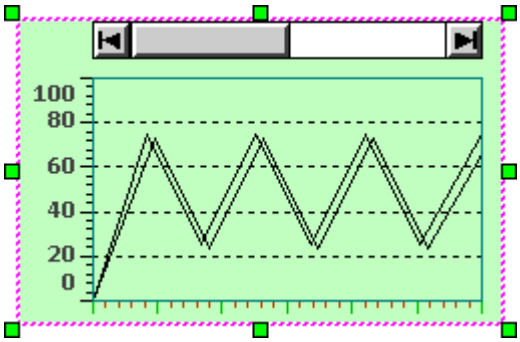
9.8.1 Real-time/History Trend Chart

Before using the **Real-time/History Trend Chart** object, you need to configure the **Data Sampling Function** in the Project View panel; see Section 9.7 **Data Sampling Management** for more details. The **Real-time/ History Trend Chart** object can be used to periodically read data, display the value, and present the data as a line graph based on the numerical distribution.

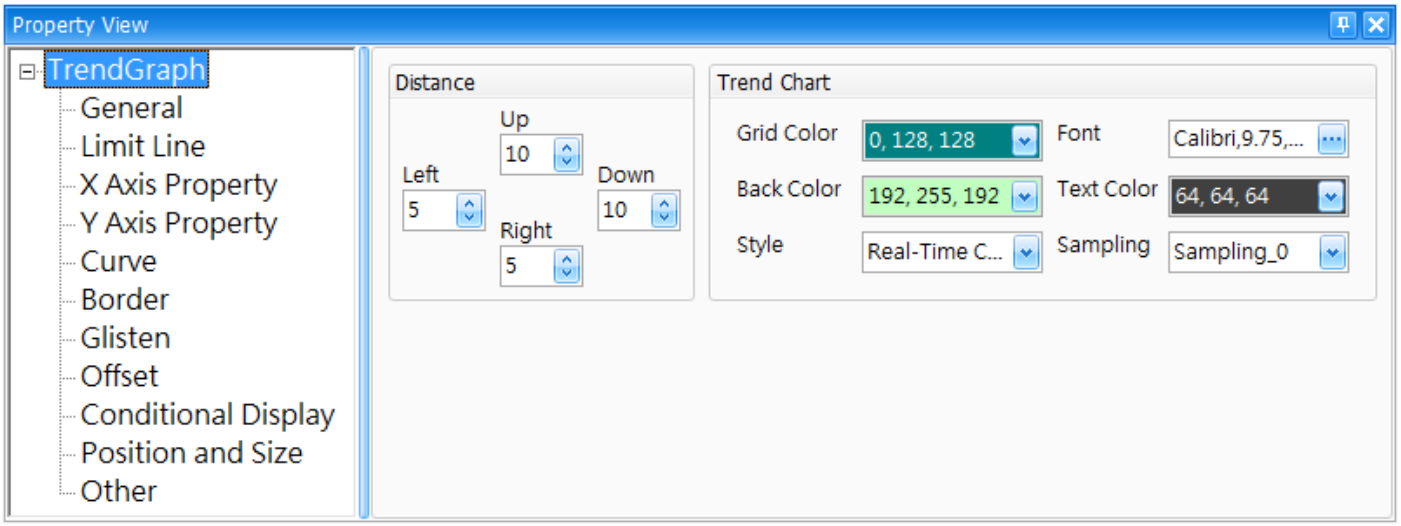
(Real-time Graph)



(History Line Graph)



The **Real-Time Trend Chart** can be used to instantly update the existing state, whereas the History Trend Chart can be used for later analysis based on long-term data sampling. See Section 9.8 Sampling Objects for details of how to create an object. After creating a **Real-time/ History Trend Graph** object, the **Property View** panel for the object will be automatically displayed allowing the various attributes of the object to be configured. The following is a description of how to configure the properties for the **Real-time/ History Trend Graph** object.



- General Properties

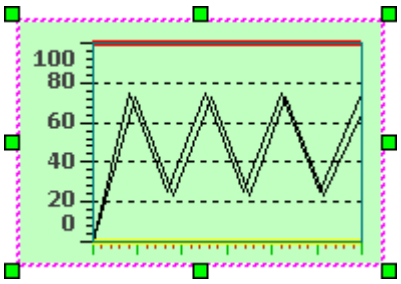
  The **General Properties** dialog box is used to configure the display format and the attributes for the **Real-time/History Trend Chart** object.

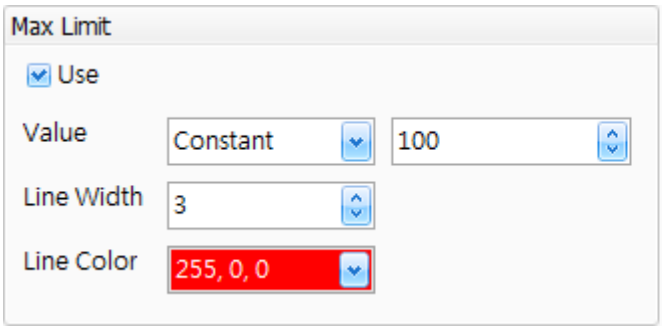
The following is an overview of the options available in the **General Properties** section of the Property View for the **Real-time/History Trend Chart** object.

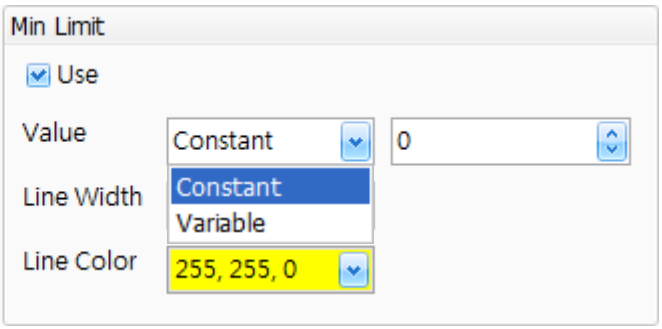
Distance	**Up/Down Left/Right**	Used to specify the position of the Line Graph in relation to the border
Line Graph	**Grid Color**	Used to specify the color of the grid for the Line Graph
	Back Color	Used to specify the background color of the Line Graph
	Style	Used to specify which type of trend chart will be used, where: **1.** Real-time Line Graph **2.** Historical Line Graph
	Font	Used to specify the type and size of the font to be used for the text
	Text Color	Used to specify the color of the text
	Sampling	Used to specify which data sampling function will be used See Section 9.7 Data Sampling Management for more details

- Limit Line Properties

  The **Limit Line Properties** dialog box is used to configure maximum and minimum limits of the **Real-time/History Trend Chart**, including the location of the limit line (based on either a constant or a variable), the width of the line and the color. For more details related to the usage of variables, see Chapter 7 Variables.





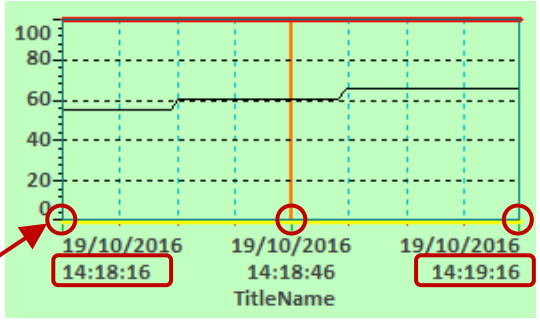


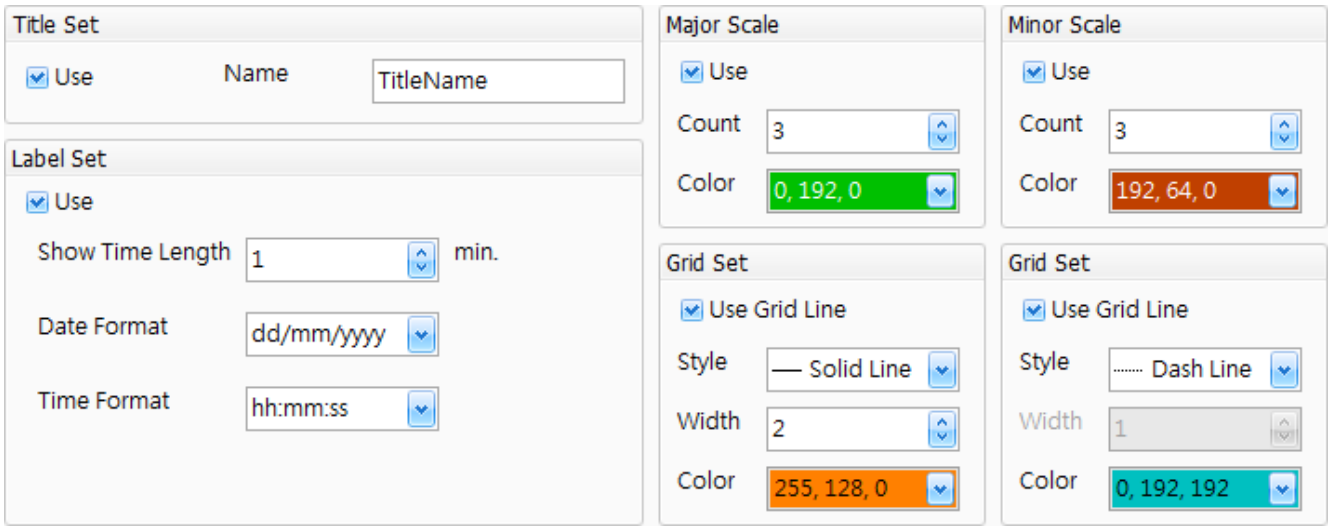
The following is an overview of the options available in the **Limit Line Properties** section of the Property View for the **Real-time/History Trend Chart** object.

Max. Limit Line	**Value**	Used to specify the upper limit for the Y-axis of the line graph
	Line Width	Used to specify the width of the line in pixels
	Line Color	Used to specify the color of the line to indicate the maximum limit
Min. Limit Line	**Value**	Used to specify the lower limit for the Y-axis of the line graph
	Line Width	Used to specify the width of the line in pixels
	Line Color	Used to specify the color of the line to indicate the minimum limit

- X-axis Properties

  The **X-axis Properties** dialog box is used to configure the title, the label, the major and minor scale, and the grid lines for the X-axis of the **Real-time/History Trend Chart** object.





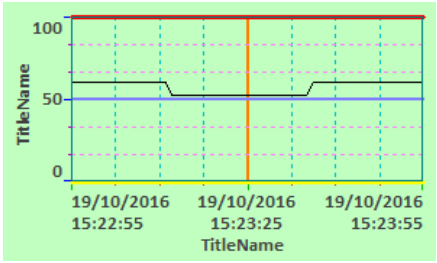
The following is an overview of the options available in the **X-axis Properties** section of the Property View for the **Real-time/History Trend Chart** object.

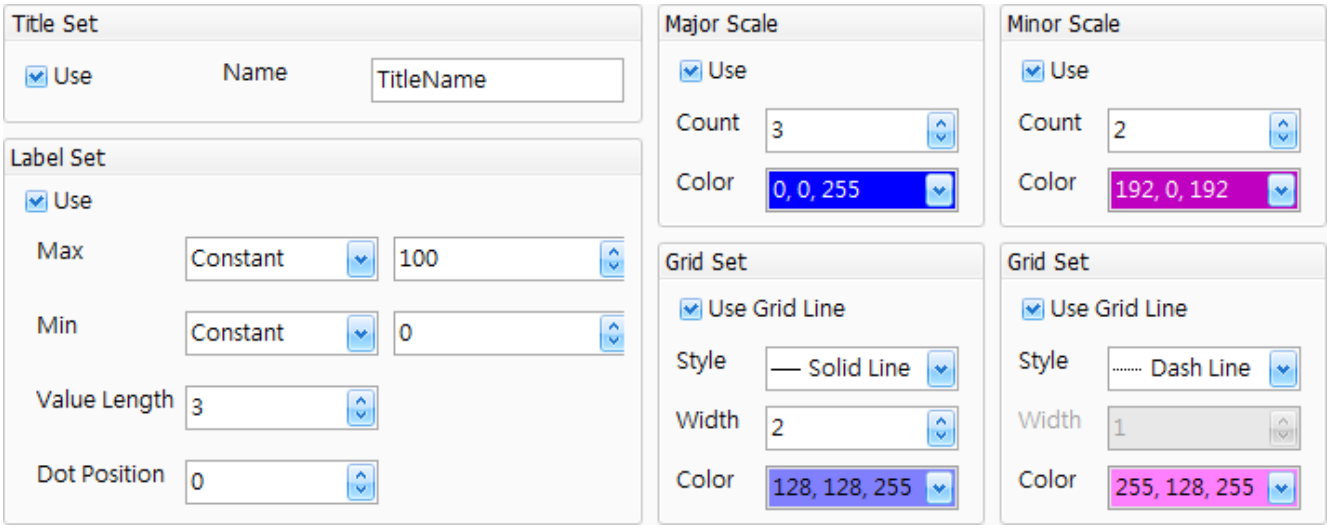
Title Set	**Name**	Used to specify a title for the X-axis
Label Set	**Show Time Length**	Used to specify the duration of the data line is displayed
	Date Format	Used to specify the format of the date, where: 1. None 2. yyyy/mm/dd (Year, Month, Day) 3. mm/dd/yyyy (Month, Day, Year) 4. dd/mm/yyyy (Day, Month, Year)
	Time Format	Used to specify the format of the time, where: 1. None 2. hh:mm:ss (Hours:Minutes:Seconds) 3. hh:mm (Hours:Minutes) 4. mm:ss (Minutes:Seconds)
Major Scale	**Count**	Used to specify the number of the major scale on the X-axis
	Color	Used to specify the color used for the major scale of the X-axis
Minor Scale	**Count**	Used to specify the number of scale between major scales on the X-axis
	Color	Used to specify the color used for the minor scale of the X-axis

Grid Set (Major Scale)	**Use Grid Line**	Used to specify whether grid lines should be displayed for the major scale of the X-axis
	Style	Used to specify the style of the grid for the major scale of the X-axis
	Width	Used to specify the width of the grid lines for the major scale of the X-axis
	Color	Used to specify the color of the grid lines for the major scale of the X-axis
Grid Set (Minor Scale)	**Use Grid Line**	Used to specify whether grid lines should be displayed for the minor scale of the X-axis
	Style	Used to specify the style of the grid for the minor scale of the X-axis
	Width	Used to specify the width of the grid lines for the minor scale of the X-axis
	Color	Used to specify the color of the grid lines for the minor scale of the X-axis

- Y-axis Properties

  The **Y-axis Properties** dialog box is used to configure the title, the label, the major and minor scale, and the grid lines for the Y-axis of the **Real-time/History Trend Chart** object.





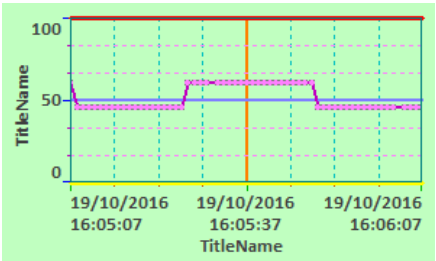
The following is an overview of the options available in the Y-axis Properties section of the Property View for the **Real-time/History Trend Chart** object.

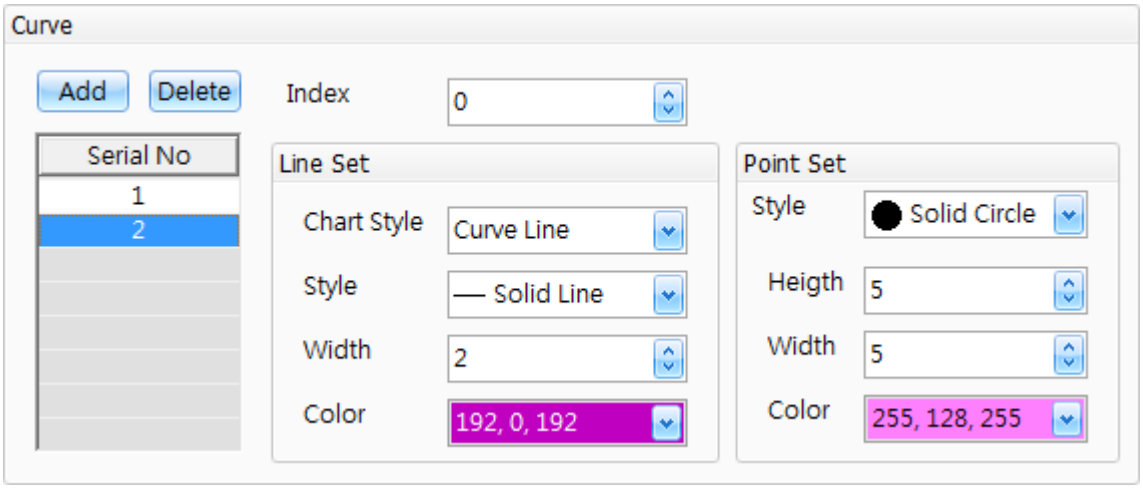
Title Set	**Name**	Used to specify a title for the Y-axis
Label Set	**Max**	Used to specify the maximum value to be displayed for the Y-axis
	Min	Used to specify the minimum value to be displayed for the Y-axis
	Value Length	Used to specify the number of digits that will be displayed
	Dot Position	Used to specify the position of the decimal point (read from the right). E.g., if the **Value Length** = 4, and the **Dot Position** = 1, then the value “100” will be displayed as “10.0”

Major Scale	**Count**	Used to specify the number of the major scale on the Y-axis
	Color	Used to specify the color used for the major scale of the Y-axis
Minor Scale	**Count**	Used to specify the number of scale between major scales on the Y-axis
	Color	Used to specify the color used for the minor scale of the Y-axis
Grid Set (Major Scale)	**Use Grid Line**	Used to specify whether grid lines should be displayed for the major scale of the Y-axis
	Style	Used to specify the style of the grid for the major scale of the Y-axis
	Width	Used to specify the width of the grid lines for the major scale of the Y-axis
	Color	Used to specify the color of the grid lines for the major scale of the Y-axis
Grid Set (Minor Scale)	**Use Grid Line**	Used to specify whether grid lines should be displayed for the minor scale of the Y-axis
	Style	Used to specify the style of the grid for the minor scale of the Y-axis
	Width	Used to specify the width of the grid lines for the minor scale of the Y-axis
	Color	Used to specify the color of the grid lines for the minor scale of the Y-axis

- Curve Properties

The **Curve Properties** dialog box is used to configure the attributes and style used to display the curve for the **Real-time/History Trend Chart** object, and is based on a variable. For more details related to the usage of variables, see the Variables Section.





The following is an overview of the options available in the **Curve Properties** section of the Property View for the **Real-time/History Trend Chart** object.

Add		Used to add a curve to the trend graph (e.g., No.3, No.4, etc.)
Delete		Used to remove one or more curve from the trend graph
Index		Used to specify the index number for the data when using the Array variable. E.g., if the data length of the variable is 3, the valid range is 0 to 2. See Chapter 7.4 Array Variables for more detail.
Line Set	**Chart Style**	Used to specify the style to be used for the curve, and can be selected from: Curve, Sawtooth Line, or Bar Chart
	Style	Used to specify the line style to be used for the curve, and can be selected from: Solid Line or Dash Line
	Width	Used to specify the width to be used for the curve
	Color	Used to specify the color to be used for the curve
Point Set	**Style**	Used to specify the style to be used for the trace point
	Height	Used to specify the height to be used for the trace point
	Width	Used to specify the width to be used for the trace point
	Color	Used to specify the color to be used for the trace point

- See Section 8.5 Common Attributes for details about the others attributes of the **Real-time/History Trend Chart** object. Border (1), Glisten, Offset, Conditional Display, Position and Size, and Other.

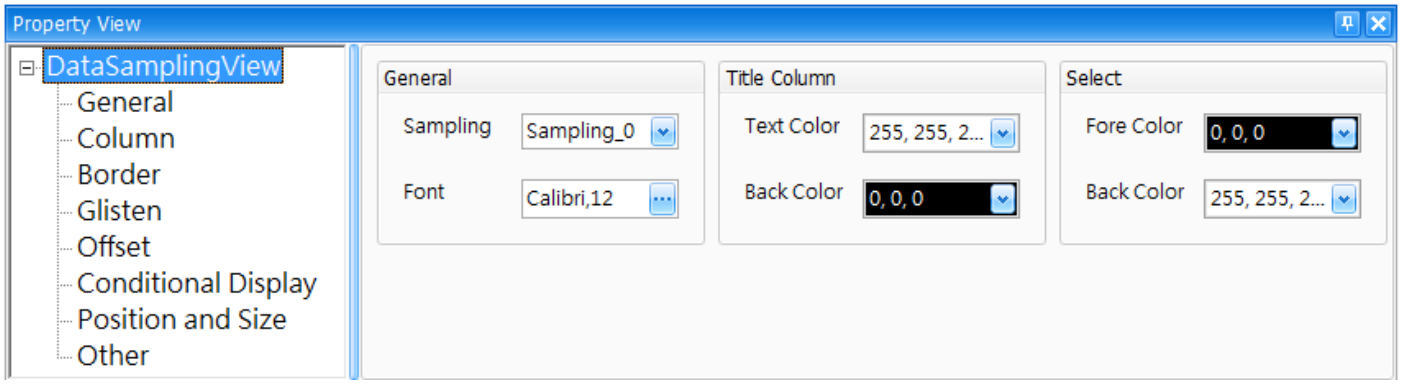
9.8.2 Data Sampling View

Before using the **Real-time/History Line Graph** object, you need to configure the **Data Sampling Function** in the Project View panel; see Section 9.7 **Data Sampling Management** for more details.

The **Data Sampling View** object can be used to graphically display numerical of sampling data in a table format. The number of columns and rows to be included in the table, as well as the color and width of the border, etc., can be adjusted as desired.

Date	Time	Tag0
20/10/2016	15:32:02	60
20/10/2016	15:32:01	60
20/10/2016	15:32:00	60
20/10/2016	15:31:59	55

See Section 9.8 Sampling Objects for details of how to create an object. After creating a **Data Sampling View** object, the **Property View** panel for the object will be automatically displayed allowing the various attributes of the object to be configured. The following is a description of how to configure the properties for the **Data Sampling View** object.



- General Properties

  The **General Properties** dialog box is used to configure the display format and attributes to be used for the **Data Sampling View** object.

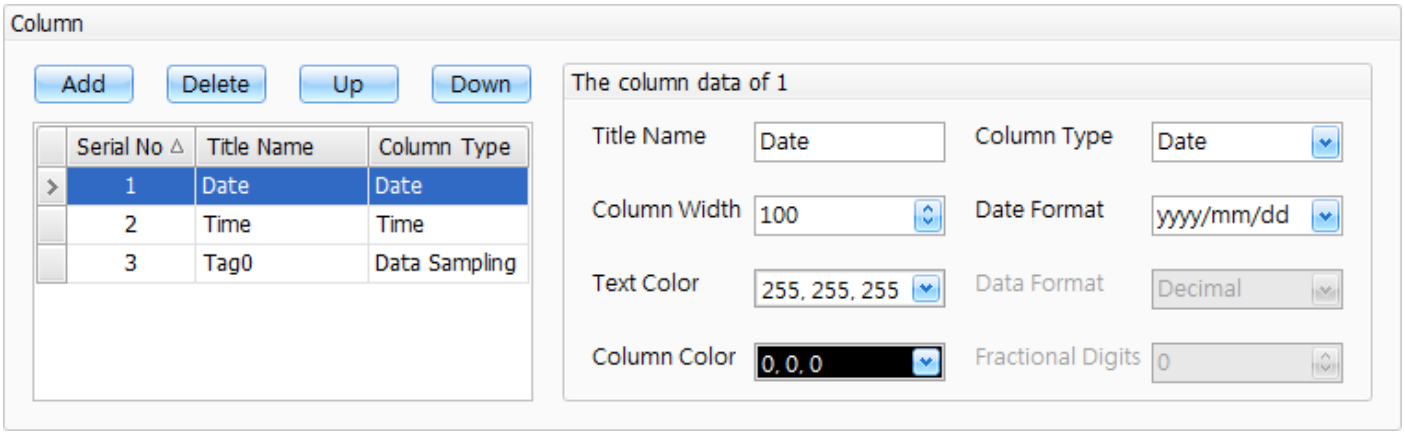
The following is an overview of the options available in the **General Properties** section of the Property View for the **Data Sampling View** object.

General	**Sampling**	Used to specify which data sampling function will be used See Section 9.7 Data Sampling Management for more details
	Font	Used to specify the type and size of the font to be used for the text
Title Column	**Text Color**	Used to specify the color of the text to be used for the title row
	Back Color	Used to specify the color of the background to be used for the title row

Select	**Fore Color**	Used to specify the color to be used for the record text when it has been selected
	Back Color	Used to specify the color of the background for the record text when it has been selected

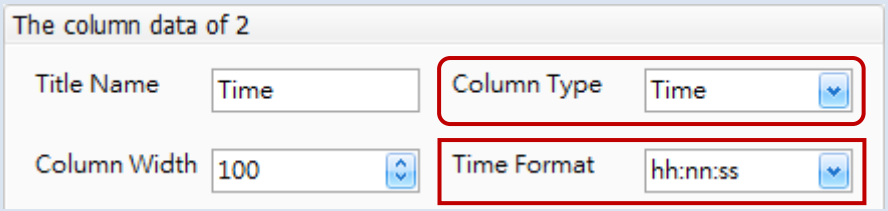
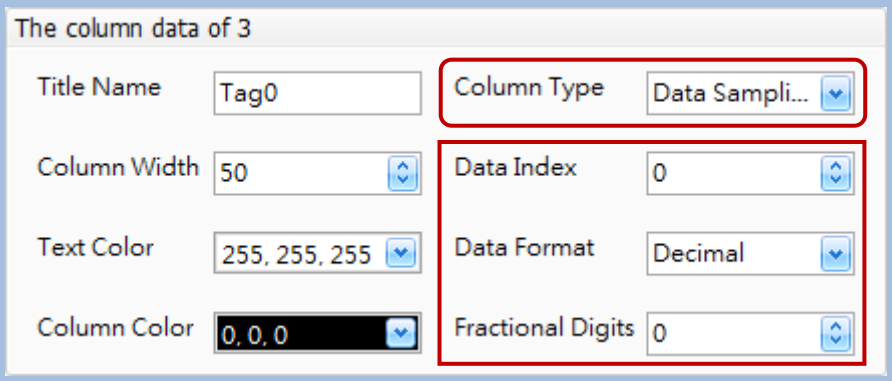
- Column Properties

  The **Column Properties** dialog box is used to configure the column format and attributes of the data table for the **Data Sampling View** object, including the date, time, the column width, and the format.



The following is an overview of the options available in the **Column Properties** section of the Property View for the **Data Sampling View** object.

Column	**Add**	Used to add a column to the data table
	Delete	Used to delete a column from the data table
	Up	Used to move the selected column left in the data table
	Down	Used to move the selected column right in the data table
Column Data	**Name**	Used to specify the column headings depends on the Column Type
	Column Width	Used to specify the width of the column
	Text Color	Used to specify the color to be used for the data text
	Column Color	Used to specify the color of the background to be used for the data column
	Column Type	Used to specify the column type to be used, where: **1**. Date **2**. Time **3**. Data Sampling

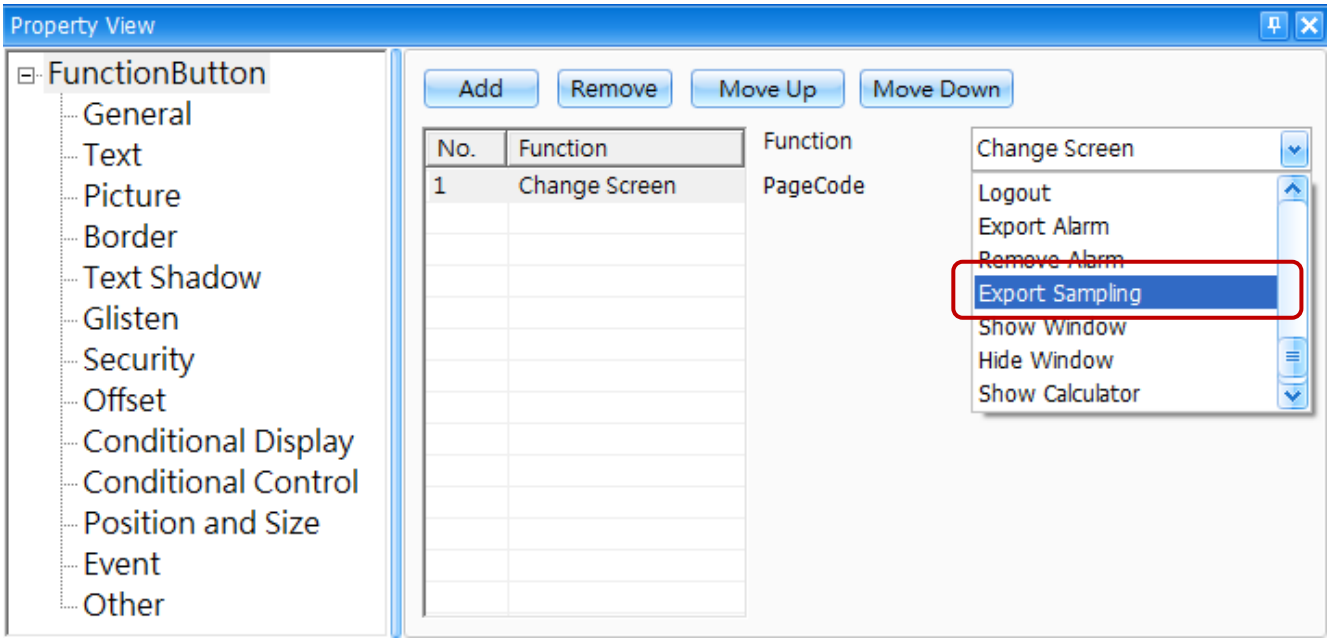
Column Data	**Date Format**	Used to specify the format used to display the date when the Date option is selected as the Column Type, where: **1**. yyyy/mm/dd (Year, Month, Day) **2**. dd/mm/yyyy (Day, Month, Year) **3**. mm/dd/yyyy (Month, Day, Year)
	Time Format	Used to specify the format used to display the time when the Time option is selected as the Column Type, where: **1.** hh:mm:ss (Hours:Minutes:Seconds) **2.** hh:mm (Hours:Minutes) **3.** mm:ss (Minutes:Seconds) 
	Data Index	Used to specify the index number for the data when the Data Sampling is selected as the Column Type  Note that if using the Array variable, and if the data length of the variable is 3, the valid range is 0 to 2. See Chapter 7.4 Array Variables for more detail.
	Data Format	Used to specify the format to be used for the data when the Data Sampling option is selected as the Column Type, where: **1**. Decimal **2**. Hexadecimal **3.** BCD
	Fractional Digits	Used to specify the number of decimal places to be displayed when the Data Sampling option is selected as the Column Type

- See Section 8.5 Common Attributes for details about the others attributes of the **Data Sampling View** object. Border (1), Glisten, Offset, Conditional Display, Position and Size, and Other

9.8.3 Sampling Function Button

Function Buttons can be used to execute a wide range of functions, such as transmitting recipes, changing screens, or incrementing and decrementing values, etc. When a button is pressed, any functions that have been defined will be executed sequentially, and a maximum of 16 functions can be implemented. The Function Button object is available in the each of the Switch, Recipe, Alarm, and Sampling categories.

The **Sampling Function Button** object provides the ability to export sampling data to a storage memory in a CSV file format. See Section 9.8 Sampling Objects for details of how to create an object. After creating a **Sampling Function Button** object, the **Property View** panel for the object will be automatically displayed allowing the various attributes of the object to be configured. The following is a description of how to configure the properties for the **Sampling Function Button** object.



- General Properties

  The **General Properties** dialog box is used to configure the actions that can be performed by the **Function Button** object.

The following is an overview of the options available in the **General Properties** section of the Property View for the **Function Button** object.

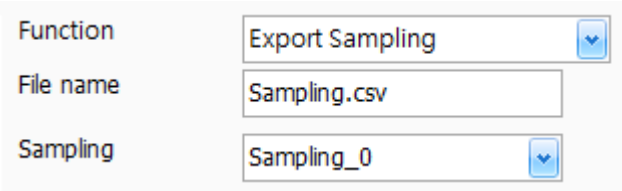
Add	Used to add a function A maximum of up to 16 functions can be used for one button
Remove	Used to remove a selected function
Move Up	Used to move the selected item up
Move Down	Used to move the selected item down
Function	Used to select the function from the Function drop-down menu

Data Sampling Function:

The following is an overview of the functions that can be used in conjunction with the **Sampling Function Button** object.

■ Export Sampling Data

The **Export Sampling Data** function is used to export the sampling records to the storage memory, e.g., a SD card, using the specified file name.



To use this function, select the **Export Sampling Data** option from the **Function** drop-down menu, enter a name for the file in the **File Name** text field, and then select the source of the sampling data (see Section 9.7 Data Sampling Management for more details) from the **Sampling** drop-down menu.

● See Section 8.5 Common Attributes for details about the others attributes of the **Recipe Function Button** object. Text, Picture, Border (2), Glisten, Security, Offset, Conditional Display, Conditional Control, Position and Size, Event, and Other.

Chapter 10 Scheduler and Macro Functions

10 – (A) Scheduler

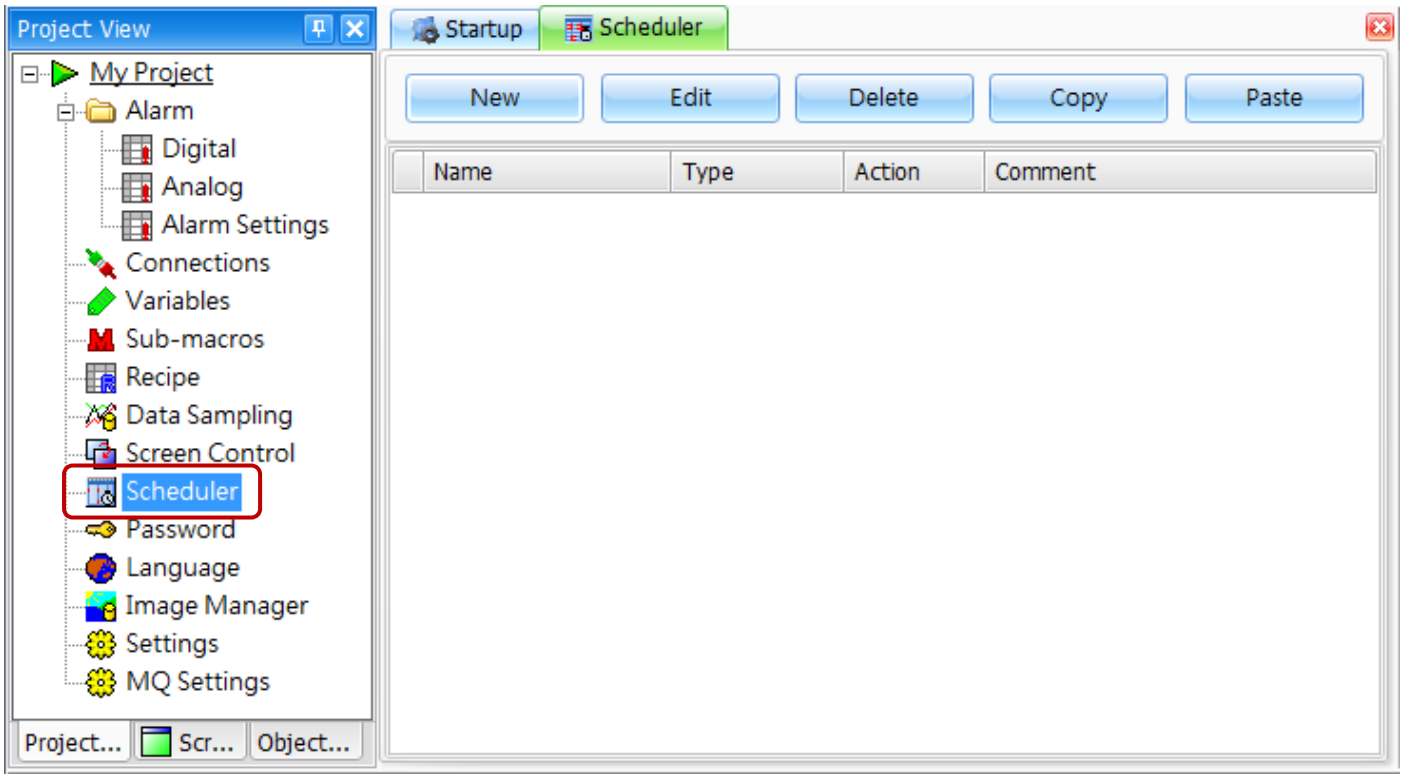
The **Scheduler** function can be used to instruct the SmartView device to execute a specific task once a particular action performed by the HMI has been completed, which can be defined via either a timer control or a trigger event. The structure that must be defined for each Scheduler event includes:

Trigger:

Used to specify the condition(s) that will trigger the Scheduler action.

Action:

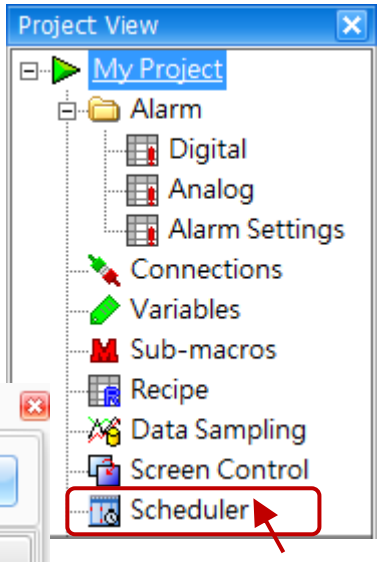
Used to specify the action that will be performed after the Scheduler condition is triggered.



Note:

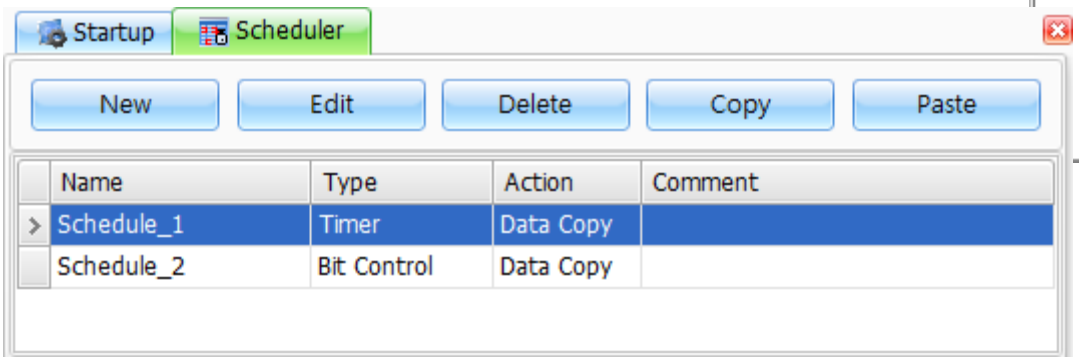
Although there is no limit to the maximum number of Scheduler events that can be configured, if too many are assigned, or if an event is executed too frequently, the burden on the system may be too great and result in low efficiency. Consequently, the necessity and usage of Scheduler events should be carefully evaluated before creating an event so as to avoid this problem.

10.1 Scheduler Management



The Scheduler function is used to create, configure, and manage the Scheduler event, including the trigger and action properties for the event.

To access the Scheduler page, double-click the **Scheduler** function in the **Project View** panel and the Scheduler page will be displayed.



The following is an overview of the options available for the **Scheduler** function in the **Property View** panel.

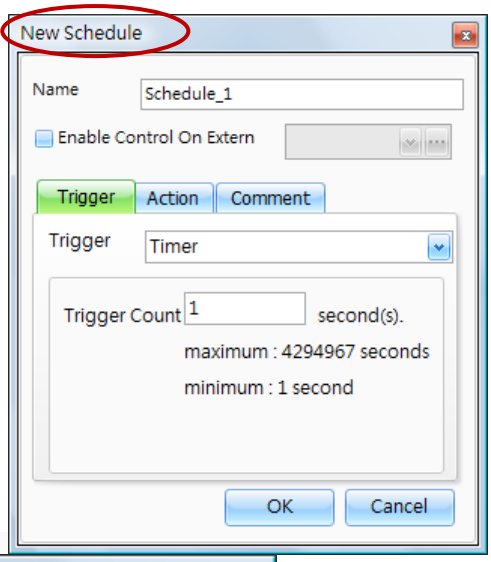
New	Used to add a new Scheduler event
Edit	Used to edit the selected Scheduler event
Delete	Used to delete the selected Scheduler event
Copy	Used to copy the selected Scheduler event
Paste	Used to paste the selected Scheduler event
Name	Used to specify a name for the Scheduler event, see Section 10.2
Type	Used to specify the type of trigger condition, see Section 10.2
Action	Used to specify the action that will be performed after the Scheduler event is triggered, see Section 10.2
Comments	Used to add an optional description for the Scheduler event, see Section 10.2

10.2 Adding /Editing a Schedule

Add a new Schedule

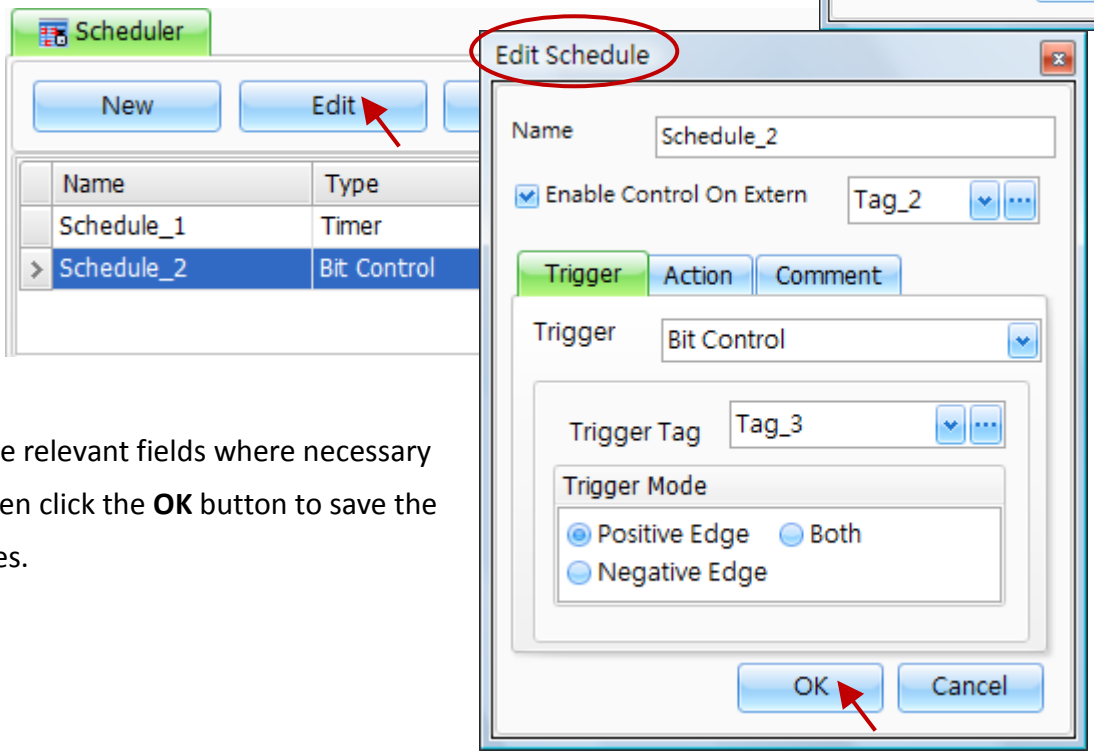
The **New Schedule** dialog box is used to configure a new Schedule event. Click the **New** button on the **Scheduler** page to open the **New Schedule** dialog box.

When the **New Schedule** dialog box is first opened, the **Trigger** tab will be displayed by default. Two other tabs, **Action** and **Comment**, are also available, and are described in the relevant sections below.



Edit an existing Schedule

The **Edit Schedule** dialog box is used to modify the attributes for an existing Schedule event. Click the **Edit** button on the **Scheduler** page to open the **Edit Schedule** dialog box.



Edit the relevant fields where necessary and then click the **OK** button to save the changes.

The following is an overview of the options available in the **Trigger** tab in the **Edit Schedule** dialog box.

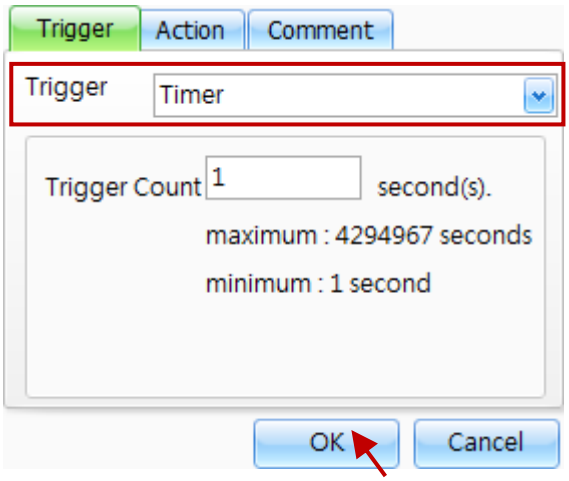
Name	Used to specify a name for the Scheduler event
Enable Control On External	Used to specify whether or not a variable will be used to control the Scheduler event
Trigger Tab (default)	Used to configure the trigger condition for the Scheduler event. For more details, see Section 10.2.1 “Trigger Tab” below.
Action Tab	Used to configure the action that will be executed after the Scheduler event is triggered. For more details, see Section 10.2.2 “Action Tab” below.
Comment Tab	Used to add a description for the Scheduler event. For more details, see Section 10.2.3 “Comment Tab” below.

10.2.1 Trigger Tab

The **Trigger** tab in the **Edit Schedule** dialog box is used to configure the type of Trigger that will be used for the Scheduler event. Two options are available, **Timer** and **Bit Control**, each of which is described in more detail below.

- Timer

When the **Timer** option is selected, the action will be triggered within a fixed period of time, which is specified using the **Trigger Count** text field. Note that the maximum cycle time is 4,294,967 seconds, and the minimum is 1 second.

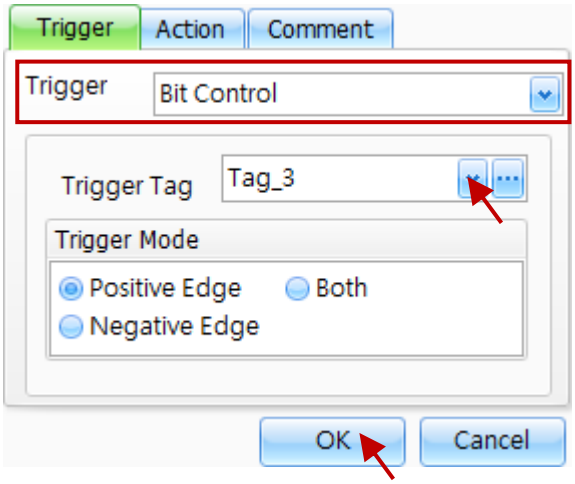


Enter the required Trigger Count value in the **Trigger Count** text field, and then click the **OK** button to save the changes.

- Bit Control

When the **Bit Control** option is selected, the action will be triggered when there is a change in the state of the variable, as follows:

1. Positive Edge: The action will be triggered if the state of the variable changes from 0 to 1
2. Negative Edge: The action will be triggered if the state of the variable changes from 1 to 0
3. Both: The action will be triggered if any change occurs in the state of the variable



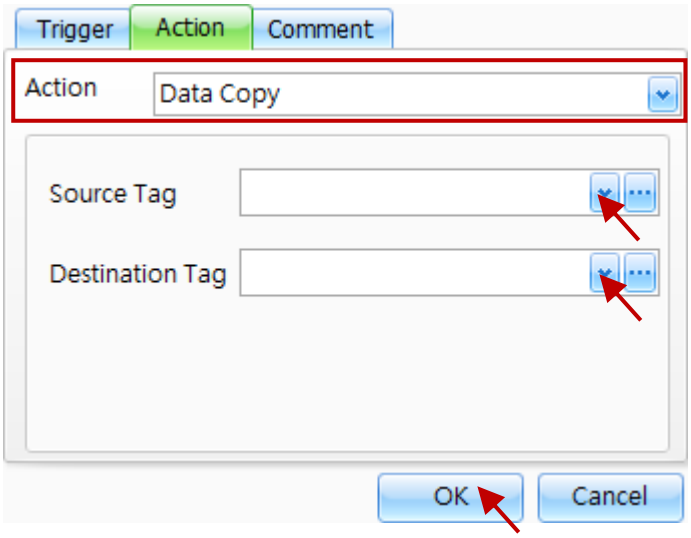
Select a **Trigger Tag** from the drop-down menu, and then click the option button for the required **Trigger Mode**. Click the **OK** button to save the changes. For more details related to the usage of variables, see the Variables Section.

10.2.2 Action

The **Action** tab in the **Edit Schedule** dialog box is used to configure the action that will be executed after the Scheduler event is triggered. Four options are available, Copy Data, Run Macro, RTC Transmission, and RTC Transmission (Byte Format), each of which is described in more detail below. To configure an action, click the **Action** drop-down menu and select the required option.

- Copy Data

When the **Copy Data** option is selected, the current value for the source variable will be copied to the destination variable after the Scheduler event is triggered. Note that an Array variable (see Section 7.4 Array Variables for details) can also be copied.



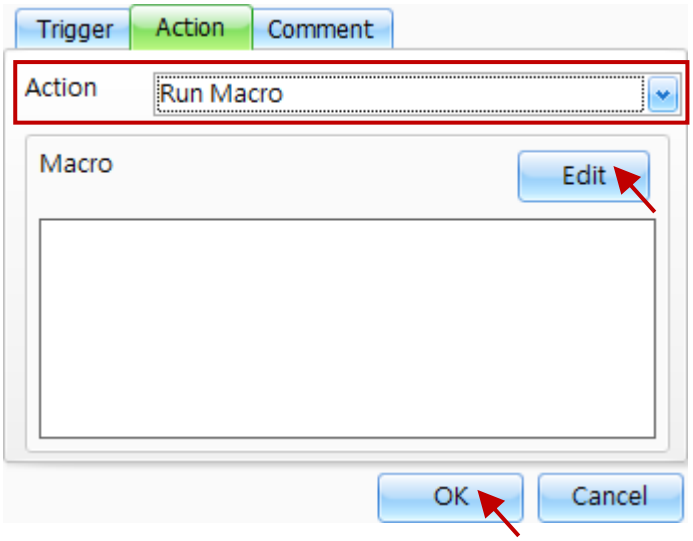
The following is an overview of the options available in the **Action** tab in the **Edit Schedule** dialog box when the **Copy Data** option is selected.

Source Tag	Used to specify the source variable that is to be used to copy the data. The source can also be configured as an Array variable, see Section 7.4 Array Variables for details
Destination Tag	Used to specify the destination variable. The destination can also be configured as an Array variable, see Section 7.4 Array Variables for details

Select the required **Source Tag** and **Destination Tag** from the respective drop-down menus, and then click the **OK** button to save the changes.

- Run Macro

When the **Run Macro** option is selected, the specified Macro program will be executed after the Scheduler event is triggered.

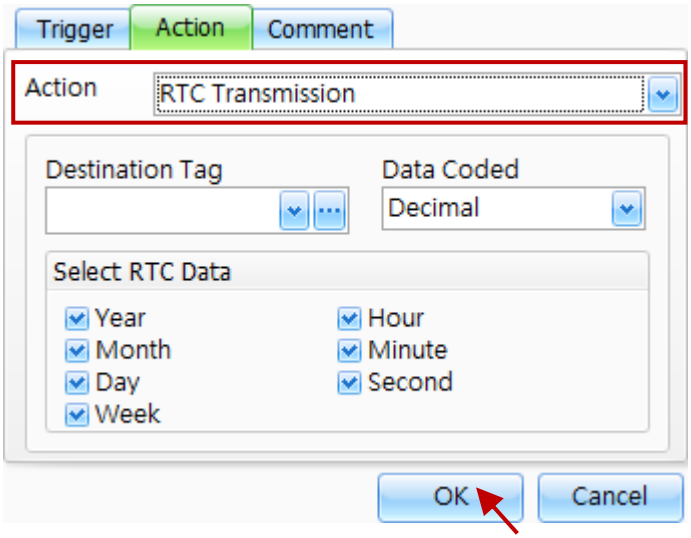


Click the **Edit** button to write a Macro program in the Macro Editor dialog box (see Section 10.6 for more details), and then click the **OK** button to save the changes. For more details related to the usage of Macros, see Section 10 – (B) Macros.

- RTC Transmission

When the **RTC Transmission** option is selected, the RTC data from the HMI device will be transmitted to the destination variable after the Scheduler event is triggered.

The following is an overview of the options available in the **Action** tab in the **Edit Schedule** dialog box when the **RTC Transmission** option is selected.



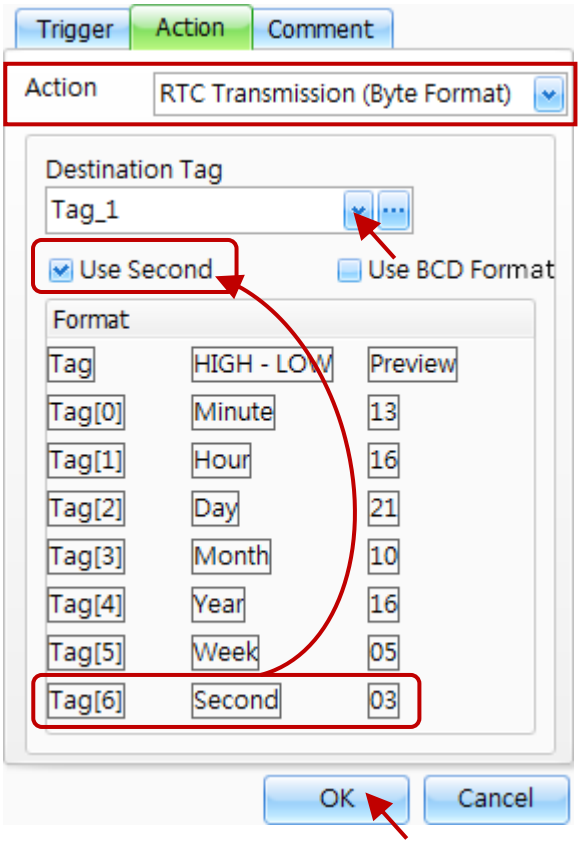
Destination Tag	Used to specify the destination variable. The destination can also be configured as an Array variable (see Section 7.4 Array Variables for details).
Data Coded	Used to specify the format that will be used to code the data, which can be either Decimal or BCD.
Select RTC Data	Used to specify which items are to be included in the RTC Data. If all data items in the **Select RTC Data** section are selected, the order that the data will be stored in the Array variable will be as follows: **Array Index** / **Data Order** 0 – Year 1 – Month 2 – Day 3 – Week 4 – Hours 5 – Minutes 6 – Seconds If any of the data items in the **Select RTC Data** section are not selected, the data item allocated to the next array index will be moved up the data order. Note that the length specified for the Array variable must be greater than the amount of stored data, or a compilation error will occur, which means that when all items are selected, the length must be set to at least 7.

- RTC Transmission

When the **RTC Transmission (Byte Format)** option is selected, the RTC data from the HMI device will be transmitted to the destination variable in Byte format after the Scheduler event is triggered.

The following is an overview of the options available in the **Action** tab in the **Edit Schedule** dialog box when the **RTC Transmission (Byte Format)** option is selected.

Destination Tag	Used to specify the destination variable. The destination can also be configured as an Array variable (see Section 7.4 Array Variables for details).
Use Seconds	Used to specify whether or not to add Seconds to the data
Use BCD Format	Used to specify whether or not to display data in BCD format
Format	Used to preview the result of the data

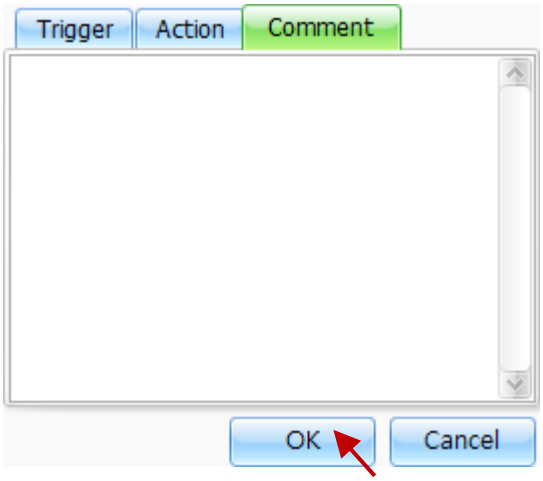


Note that the length specified for the Array variable must be greater than the amount of stored data, or a compilation error will occur, which means that when all items are selected, the length must be set to at least 7.

10.2.3 Comment

The **Comment** tab of the **Edit Schedule** dialog box is used to add an optional description for the Scheduler event.

Enter the desired information in the text field and then click the **OK** button to complete the settings.



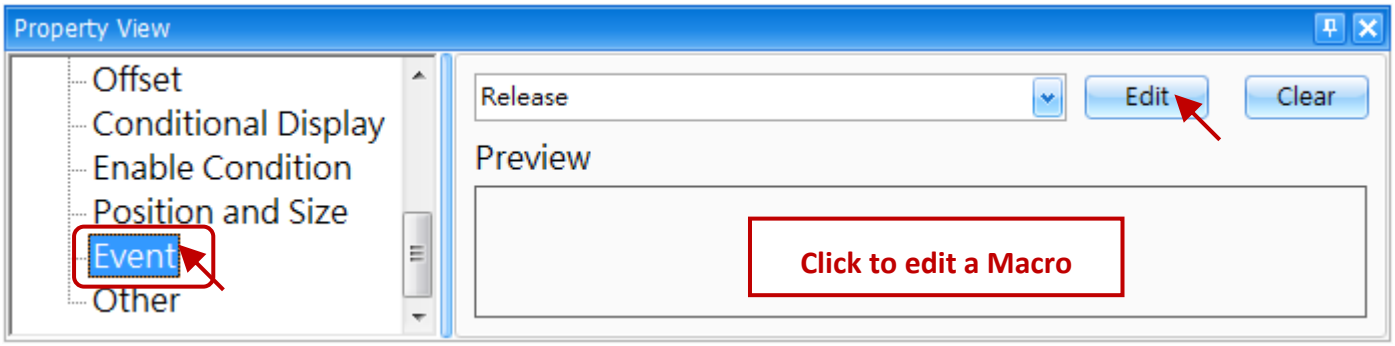
10 – (B) Macro

A Macro is a set of instructions that can be combined into a single program designed to perform specific control or logic operations. Importantly, attention should be paid to the rationality of the program logic when designing a Macro. For example, a Macro that generates an infinite loop may cause the system operation to become abnormal. In addition, a Macro that is configured to be executed too frequently or for too long may also cause lower overall effectiveness.

A Macro can be created and edited from several areas of the Creator application. For example, a Macro can be defined from the **Event** Property View panel for an object. A Macro can also be created by selecting **Run Macro** in the **Action** section in the **Edit Schedule** dialog box for the **Scheduler** function, or by selecting **Sub Macro** function form the **Project View** panel, each of which is described below.

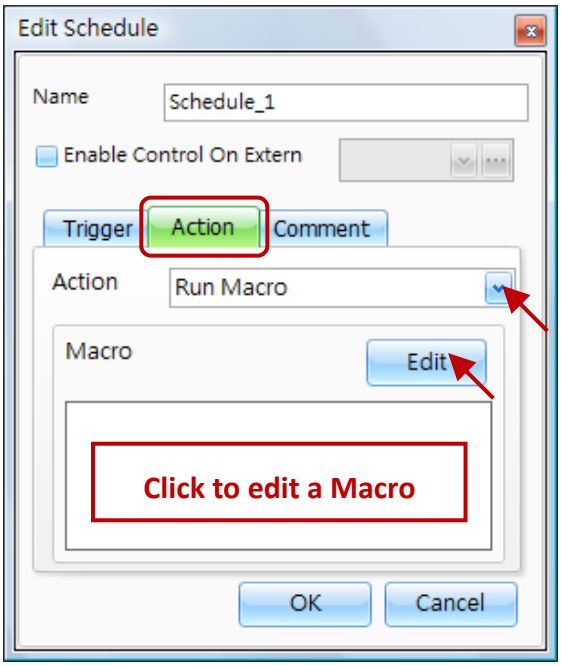
Creating a Macro from the Event dialog box for the object:

To create a Macro using the **Event** dialog box for an object, click the object and then click **Event** in the **Property View** for that object, as illustrated in the example below.



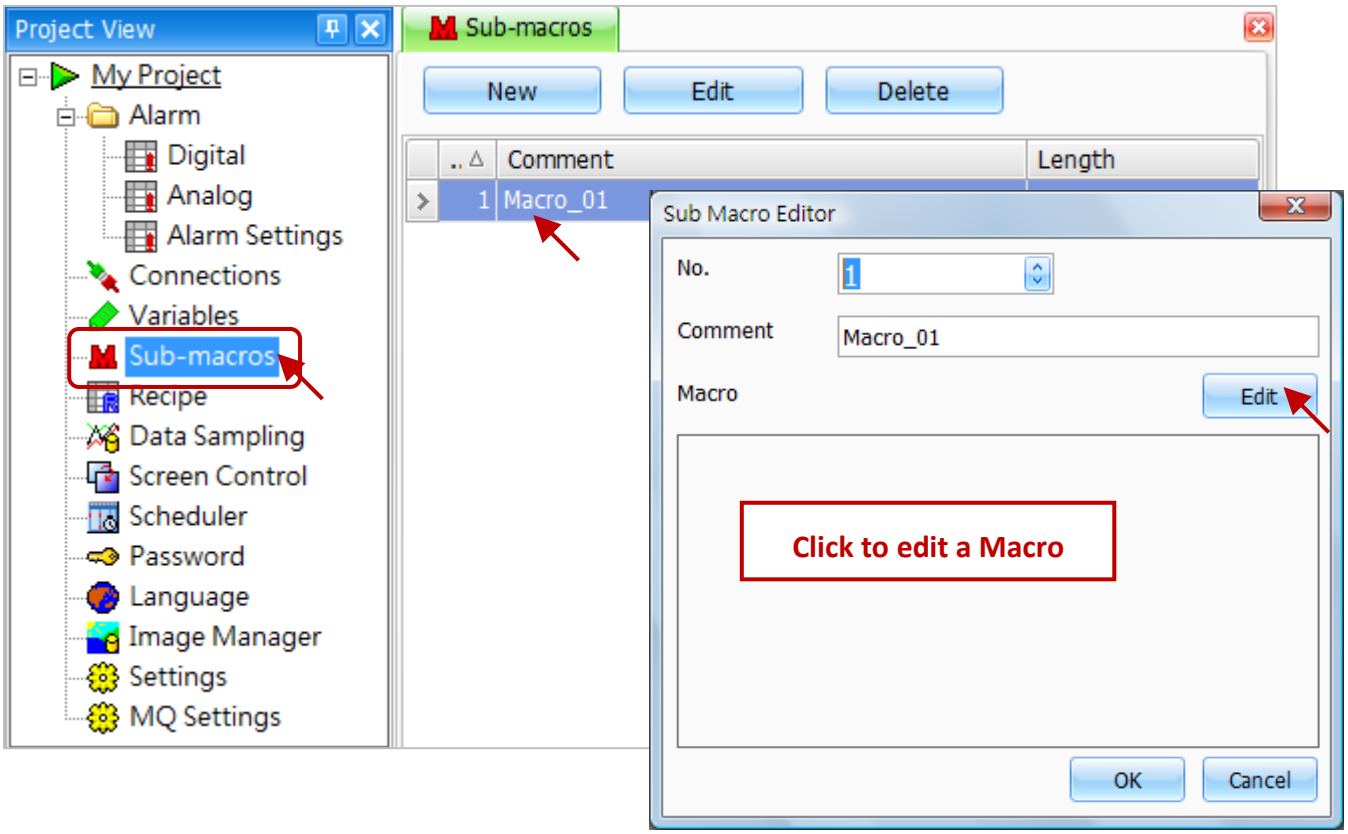
Creating a Macro using the Run Macro Action in the Scheduler function:

To create a Macro using the **Run Macro** Action in the Scheduler function, double-click the **Scheduler** function in the **Project View** panel to open the Scheduler page. Double-click any existing Scheduler event (or add a new event) to open the Edit/New Schedule dialog box, click the Action tab and then select the **Run Macro** option from the drop-down menu, as illustrated in the example.



Creating a Macro using the Sub Macro function in the Project View panel:

To create a Macro using the **Sub Macro** function, double-click the **Sub- macros** function in the **Project View** panel to open the **Sub-Macros** page. Click the **New** button to display the Sub Macro Editor dialog box, as illustrated in the example below.

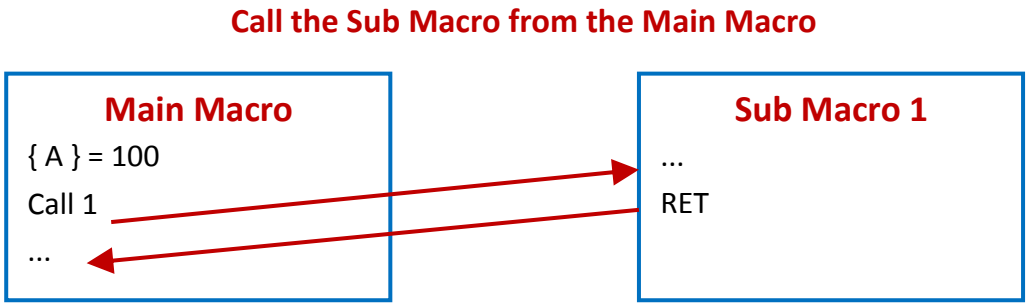


Note:

The method for editing a Macro is the same for each of the Event, Scheduler or Sub Macro options, and will be introduced in the following descriptions using the Sub Macro option as an example.

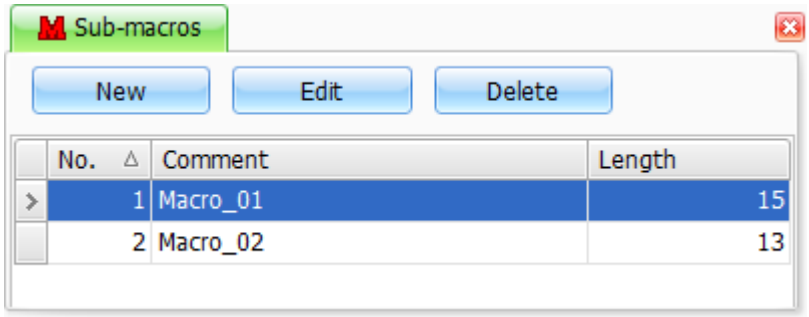
10.3 Sub Macro Overview

A Sub Macro is an independent Macro program that can be called from a main Macro program using a *Call* command. A Sub Macro can be instructed to repeatedly call other Sub Macros for up to three cycles, but after more than three cycles the *Call* command will no longer be executed. Note that a maximum of 512 Sub Macro items can be created.



10.4 Sub Macro Management

The Sub Macro function can be used to create a new Sub Macro function or can be used to edit an existing Sub Macro.



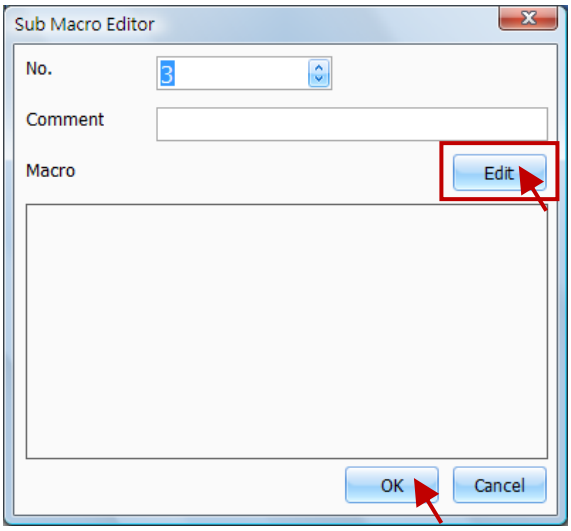
To create a **Sub Macro**, double-click the **Sub Macro** function in the **Project View** panel. The Sub-macros page will then be displayed.

The following is an overview of the options available on the **Sub Macro** page.

New	Used to create a new Sub Macro. A maximum of 512 Sub Macro items can be created.
Edit	Used to edit the details of a selected Sub Macro
Delete	Used to delete a selected Sub Macro

To create a new Sub Macro, click the **New** button to display the **Sub Macro Editor** dialog box.

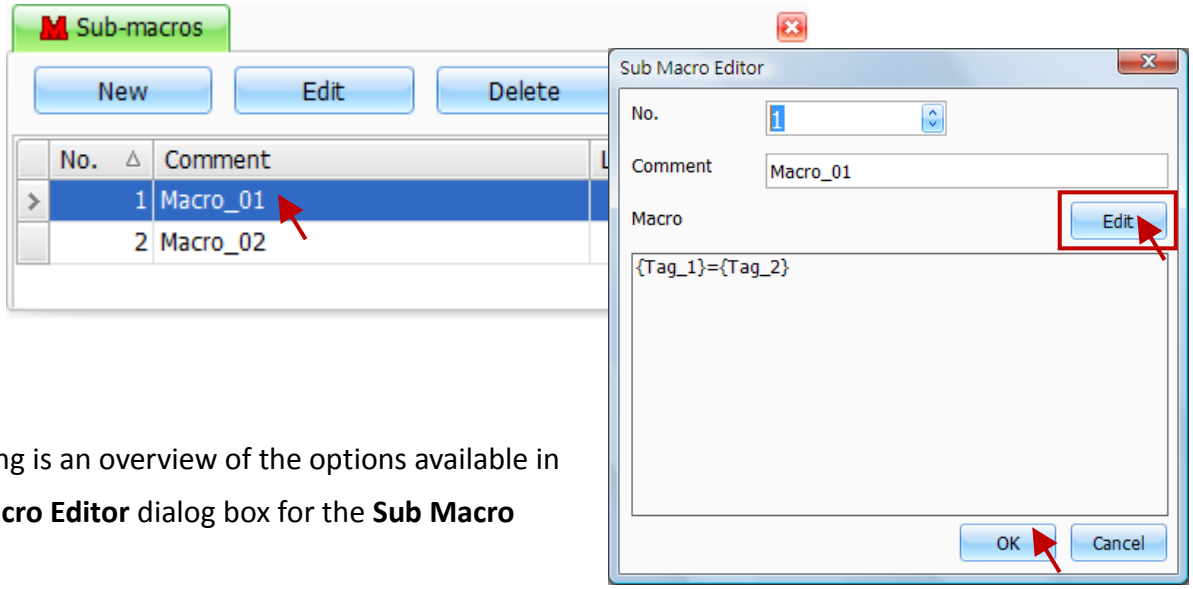
Click the **Edit** button to display the **Macro Editor** dialog box, see Section 10.6 Macro Editor for more details related to the usage. Enter the necessary commands and then click the **OK** button to save the changes.



See next section for an overview of the options available in the **Sub Macro Editor** dialog box for the **Sub Macro** function.

10.5 Editing a Sub Macro

The **Sub Macro Editor** dialog box is also used to edit an existing Sub Macro function. Double-click any existing Sub Macro function in the Sub-macros page to display the **Sub Macro Editor** dialog box.



The following is an overview of the options available in the **Sub Macro Editor** dialog box for the **Sub Macro** function.

No.	Used to specify the index number of the Sub Macro, which can be in the range of 1 to 512
Comment	Used to add a description for the Sub Macro, usually a name that can be easily recognized
Macro	Used to view and/or edit the instructions for the Sub Macro
Edit	Used to open the Macro Editor window. See Section 10.6 “Macro Editor” for more details

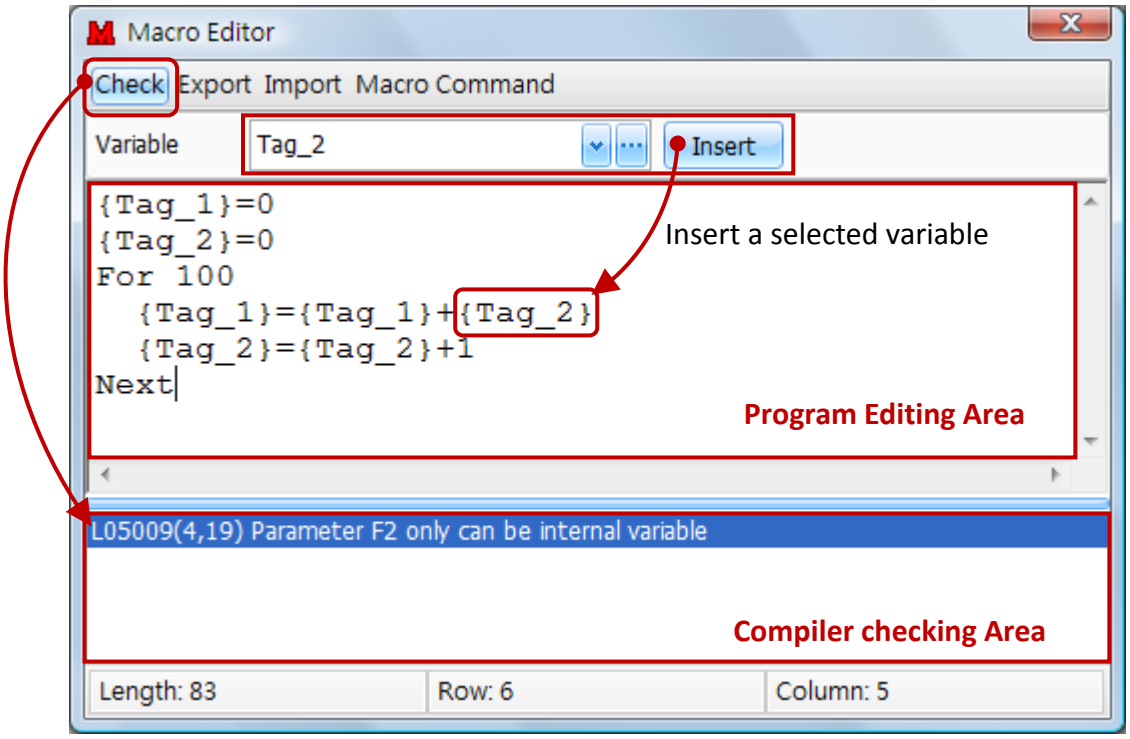
To edit an existing Sub Macro, click the **Edit** button to display the **Macro Editor** dialog box. Adjust any of the commands as necessary and then click the **OK** button to save the changes.

10.6 Macro Editor

The **Macro Editor** dialog box is used to write or edit a Macro program. To access the Macro Editor, click the **Edit** button in the **Sub Macro Editor** dialog box as noted before, or simply click anywhere in the **Macro** editing area and the Macro Editor window will be displayed allowing a Macro program to be written.

At the top of the Editor window is a menu that includes Check, Export, Import, and Macro Command functions, which can be used to quickly and accurately create a Macro.

To add a variable to the Macro program, click the Variable drop-down menu to select an existing variable, or click the "…" ( ) button to create a new variable, and then click the **Insert** button to insert the selected variable into the Macro code.

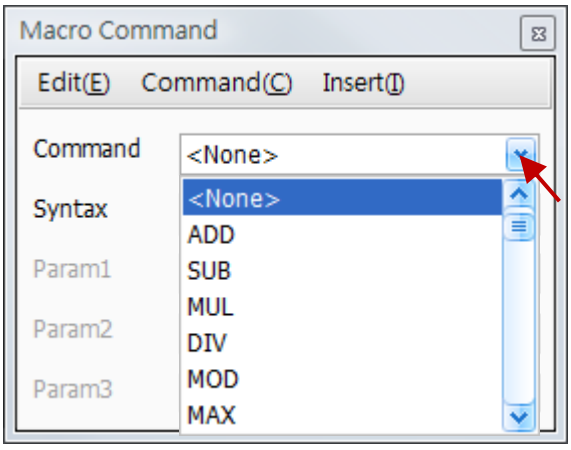


The following is an overview of the options available in the **Macro Editor** dialog box**.**

Check	Used to verify that the program is valid
Export	Used to export the Macro program as a text file in .txt format
Import	Used to import an existing Macro program in .txt format
Macro Command	Used to execute the Macro command
Variable	Used to select an existing variable, see Chapter 7 Variables for more details
	Used to create a new variable or edit/delete an existing variable
Insert	Used to insert the selected variable into the Macro code

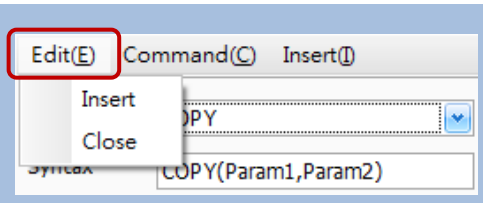
10.7 Macro Command Tool

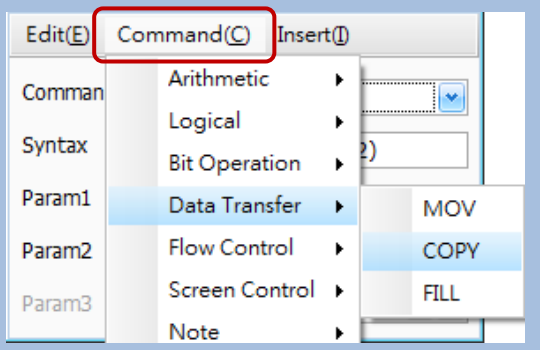
The **Macro Command** menu option in the **Macro Editor** dialog box provides a number of tools that can be used to prompt the syntax and parameters for a selected command, which can help prevent compilation errors.

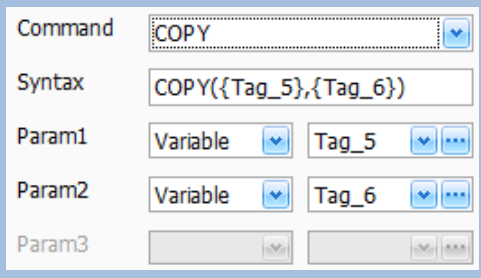


The following is an overview of the options available in the **Macro Command** dialog box

Edit	Used to insert the current command and close the Macro Command Tool window
Command	Used to select a command depends on the category, where: 1. Arithmetic (e.g., ADD, MAX, AVG, etc.) 2. Logical (e.g., AND, OR, BCD, etc.) 3. Bit Operation (e.g., BSET, BCLR, etc.). 4. Data Transfer (e.g., MOV, COPY, etc.) 5. Flow Control (e.g., IF, CALL, FOR, etc.) 6. Screen Control (e.g., SHOWWINDOW, etc.) 7. Note (i.e., " ; ")
Insert	Used to insert the current syntax into the Macro program editing area
Command	Used to select a command (unclassified)
Syntax	Used to preview or edit the syntax for the selected command
Param 1 to 3	Used to configure the variables or the constant to be used for the parameters







10.8 Using a Variables in a Macro

Accessing a variable in a Macro is as simple as adding braces to the name of the required variable, for example, {TAG}, where **TAG** is the name of the variable. Note that the name of the TAG is **NOT** case sensitive.

If the variable to be accessed is an external variable, the HMI device will immediately initiate a communication request, and will not process the next command until that command is completed. Consequently, when a data operation is being performed, it is recommended that any external variables are first copied to an internal variable, which can then be used to perform any calculations in order to increase the efficiency of the program execution.

10.9 Macro Commands

The following is an overview of the commands that can be performed using a Macro, and are broken down into seven categories, Arithmetic, Logical, Bit Operation, Data Transfer, Flow Control, Screen Control, and Notes.

Arithmetic	Logical	Bit Operation	Data Transfer	Flow Control	Screen Control	Notes
Section 10.9.1	Section 10.9.2	Section 10.9.3	Section 10.9.4	Section 10.9.5	Section 10.9.6	Section 10.9.7
ADD	AND	BSET	MOV	IF	SHOWWINDOW	;
SUB	OR	BCLR	COPY	ELSEIF	HIDEWINDOW	
MUL	XOR	BINV	FILL	ELSE	SHOWSCREEN	
DIV	NOT			ENDIF		
MOD	BCD			CALL		
MAX	BIN			RET		
MIN	SHL			LABEL		
AVG	SHR			GOTO		
SUM				FOR...NEXT		
				DELAY		

A detailed description of all commands available in each category is provided in the sections below, including the related syntax for the command and an example of its usage.

Note: The use of external variables is not supported by the majority of the Macros listed above.

10.9.1 Arithmetic Macro Commands

- **ADD**

Description:

This Macro Command is used to add variable F1 to variable F2, and then save the result to variable F0.

Syntax:

F0 = F1 + F2

Parameters	F0: Result , F1: Augend , F2: Addend			
Supported Variable Type		Constant	Internal Variable	External Variable
	F0		✔	
	F1	✔	✔	
	F2	✔	✔	

Example:

Variables:

Name: F0 ; **Address**: @R0 ; **Data Type**: WORD ; **Data Length**: 1

Name: F1 ; **Address**: @R100 ; **Data Type**: WORD ; **Data Length**: 1

F1 = 50 ; F2 = 100 (constant)

Contents:

{F0} = {F1} + 100

Result:

F0 = 150

- **SUB**

Description:

This Macro Command is used to subtract variable F2 from variable F1, and then save the result to variable F0.

Note that if F0 is **not** a signed variable and the result is negative, an overflow will occur and the resulting value will be incorrect.

Syntax:

F0 = F1 - F2

Parameters	F0: Result, F1: Minuend, F2: Subtrahend			
Supported Variable Type		Constant	Internal Variable	External Variable
	F0		✔	
	F1	✔	✔	
	F2	✔	✔	

Example:

Variables:

Name: F0 ; **Address**: @R0 ; **Data Type**: WORD ; **Data Length**: 1

Name: F1 ; **Address**: @R100 ; **Data Type**: WORD ; **Data Length**: 1

F1 = 50 ; F2 = 10 (constant)

Contents:

{F0} = {F1} - 10

Result:

F0 = 40

- **MUL**

Description:

This Macro Command is used to multiply variable F1 by variable F2, and then save the result to variable F0.

Syntax:

F0 = F1 * F2

Parameters	F0: Result, F1: Multiplicand, F2: Multiplier			
Supported Variable Type		Constant	Internal Variable	External Variable
	F0		✔	
	F1	✔	✔	
	F2	✔	✔	

Example:

Variables:

Name: F0 ; **Address**: @R0 ; **Data Type**: WORD ; **Data Length**: 1

Name: F1 ; **Address**: @R100 ; **Data Type**: WORD ; **Data Length**: 1

F1 = 50 ; F2 = 100 (constant)

Contents:

{F0} = {F1} * 100

Result:

F0 = 5000

- **DIV**

Description:

This Macro Command is used to divide variable F1 by variable F2, and then save the result to variable F0.

Syntax:

F0 = F1 / F2

Parameters	F0: Result, F1: Dividend, F2: Divisor			
Supported Variable Type		Constant	Internal Variable	External Variable
	F0		✔	
	F1	✔	✔	
	F2	✔	✔	

Example:

Variables:

Name: F0 ; **Address**: @R0 ; **Data Type**: WORD ; **Data Length**: 1

Name: F1 ; **Address**: @R100 ; **Data Type**: WORD ; **Data Length**: 1

F1 = 50 ; F2 = 5 (constant)

Contents:

{F0} = {F1} / 5

Result:

F0 = 10

- **MOD**

Description:

This Macro Command is used to divide variable F1 by variable F2, and then save the remainder to variable F0.

Syntax:

F0 = F1 % F2

Parameters	F0: Result, F1: Dividend, F2: Divisor			
Supported Variable Type		Constant	Internal Variable	External Variable
	F0		✔	
	F1	✔	✔	
	F2	✔	✔	

Example:

Variables:

Name: F0 ; **Address**: @R0 ; **Data Type**: WORD ; **Data Length**: 1

Name: F1 ; **Address**: @R100 ; **Data Type**: WORD ; **Data Length**: 1

F1 = 45 ; F2 = 10 (constant)

Contents:

{F0} = {F1} % 10

Result:

F0 = 5

- **MAX**

Description:

This Macro Command is used to compare the values of variable F1 and variable F2, and then save the larger value to variable F0.

Syntax:

F0 = MAX(F1, F2)

Parameters	F0: Result, F1: Comparing value, F2: Comparing value			
Supported Variable Type		Constant	Internal Variable	External Variable
	F0		✔	
	F1	✔	✔	
	F2	✔	✔	

Example:

Variables:

Name: F0 ; **Address**: @R0 ; **Data Type**: WORD ; **Data Length**: 1

F1 = 50 (constant) ; F2 = 100 (constant)

Contents:

{F0} = MAX(50, 100)

Result:

F0 = 100

- **MIN**

Description:

This Macro Command is used to compare the values of variable F1 and variable F2, and then save the smaller value to variable F0.

Syntax:

F0 = MIN(F1, F2)

Parameters	F0: Result, F1: Comparing value, F2: Comparing value			
Supported Variable Type		Constant	Internal Variable	External Variable
	F0		✔	
	F1	✔	✔	
	F2	✔	✔	

Example:

Variables:

Name: F0 ; **Address**: @R0 ; **Data Type**: WORD ; **Data Length**: 1

F1 = 50 (constant) ; F2 = 100 (constant)

Contents:

{F0} = MIN (50, 100)

Result:

F0 = 50

- **AVG**

Description:

This Macro Command is used to calculate the average of the elements contained in Array variable F1, and then save the result to variable F0. The Macro adds all the values contained in the Array variable F1, and then divides the total by variable F2. Note that the F2 value is the total number of elements contained in the array F1.

Syntax:

F0 = AVG (F1, F2)

Parameters	F0: Result, F1: Sum (Can be an Array), F2: Divisor			
Supported Variable Type		Constant	Internal Variable	External Variable
	F0		✔	
	F1	✔	✔	
	F2	✔	✔	

Note: The F2 value is the count for the number of values contained in the array.

Example:

Variables:

Name: F0 ; **Address**: @R0 ; **Data Type**: WORD ; **Data Length**: 1

Name: F1 ; **Address**: @R100 ; **Data Type**: WORD ; **Data Length**: 10

F1 = { 10, 20, 30, 40 ,50 ,60, 70, 80, 90, 100 }

F2 = 10 (This value is the count for the number of values contained in the array F1)

Contents:

{F0} = AVG(F1, 10)

Result:

F0 = 55

- **SUM**

Description:

This Macro Command is used to calculate the sum of F2 elements from the F1 variable Array, and then save the total to variable F0.

Syntax:

F0 = SUM(F1, F2)

Parameters	F0: Result, F1: Sum (Can be an Array) , F2: Length			
Supported Variable Type		Constant	Internal Variable	External Variable
	F0		✔	
	F1	✔	✔	
	F2	✔	✔	

Example:

Variables:

Name: F0 ; **Address**: @R0 ; **Data Type**: WORD ; **Data Length**: 1

Name: F1 ; **Address**: @R100 ; **Data Type**: WORD ; **Data Length**: 10

F1 = { 10, 20, 30, 40 ,50 ,60, 70, 80, 90, 100 }

F2 = 5 (constant)

Contents:

{F0} = SUM(F1, 5)

Result:

F0 = 150

10.9.2 Logical Macro Commands

- **AND**

Description:

This Macro Command is used to perform an AND operation on variables F1 and F2, and then save the result to variable F0.

Syntax:

F0 = AND(F1, F2)

Parameters	F0: Result, F1: Operator, F2: Operator			
Supported Variable Type		Constant	Internal Variable	External Variable
	F0		✔	
	F1	✔	✔	
	F2	✔	✔	

Example:

Variables:

Name: F0 ; **Address**: @R0 ; **Data Type**: WORD ; **Data Length**: 1

Name: F1 ; **Address**: @R100 ; **Data Type**: WORD ; **Data Length**: 1

F1 = 15 ; F2 = 5 (constant)

Contents:

{F0} = AND(F1, 5)

Result:

F0 = 5

- **OR**

Description:

This Macro Command is used to perform an OR operation on variables F1 and F2, and then save the result to variable F0.

Syntax:

F0 = OR(F1, F2)

Parameters	F0: Result, F1: Operator, F2: Operator			
Supported Variable Type		Constant	Internal Variable	External Variable
	F0		✔	
	F1	✔	✔	
	F2	✔	✔	

Example:

Variables:

Name: F0 ; **Address**: @R0 ; **Data Type**: WORD ; **Data Length**: 1

Name: F1 ; **Address**: @R100 ; **Data Type**: WORD ; **Data Length**: 1

F1 = 3 ; F2 = 5 (constant)

Contents:

{F0} = OR(F1, 5)

Result:

F0 = 7

- **XOR**

Description:

This Macro Command is used to perform an XOR operation on variables F1 and F2, and then save the result to variable F0.

Syntax:

F0 = XOR(F1, F2)

Parameters	F0: Result, F1: Operator, F2: Operator			
Supported Variable Type		Constant	Internal Variable	External Variable
	F0		✔	
	F1	✔	✔	
	F2	✔	✔	

Example:

Variables:

Name: F0 ; **Address**: @R0 ; **Data Type**: WORD ; **Data Length**: 1

Name: F1 ; **Address**: @R100 ; **Data Type**: WORD ; **Data Length**: 1

F1 = 3 ; F2 = 5 (constant)

Contents:

{F0} = XOR(F1, 5)

Result:

F0 = 6

- **NOT**

Description:

This Macro Command is used to perform a NOT operation on variable F1, and then save the result to variable F0.

Syntax:

F0 = NOT(F1)

Parameters	F0: Result, F1: Operator			
Supported Variable Type		Constant	Internal Variable	External Variable
	F0		✔	
	F1	✔	✔	

Example:

Variables:

Name: F0 ; **Address**: @R0 ; **Data Type**: WORD ; **Data Length**: 1

Name: F1 ; **Address**: @R100 ; **Data Type**: WORD ; **Data Length**: 1

F1 = FF00H

Contents:

{F0} = NOT(F1)

Result:

F0 = 00FFH

- **BCD**

Description:

This Macro Command is used to convert the contents of variable F1 to BCD code, and then save the result to variable F0.

Syntax:

F0 = BCD(F1)

Parameters	F0: Result, F1: Operator			
Supported Variable Type		Constant	Internal Variable	External Variable
	F0		✔	
	F1	✔	✔	

Example:

Variables:

Name: F0 ; **Address**: @R0 ; **Data Type**: WORD ; **Data Length**: 1

Name: F1 ; **Address**: @R100 ; **Data Type**: WORD ; **Data Length**: 1

F1 = 12

Contents:

{F0} = BCD(F1)

Result:

F0 = 0012H

- **BIN**

Description:

This Macro Command is used to convert the contents of variable F1 to Binary code, and then save the result to variable F0.

Syntax:

F0 = BIN(F1)

Parameters	F0: Result, F1: Operator			
Supported Variable Type		Constant	Internal Variable	External Variable
	F0		✔	
	F1	✔	✔	

Example:

Variables:

Name: F0 ; **Address**: @R0 ; **Data Type**: WORD ; **Data Length**: 1

Name: F1 ; **Address**: @R100 ; **Data Type**: WORD ; **Data Length**: 1

F1 = 0012H

Contents:

{F0} = BIN(F1)

Result:

F0 = 12

- **SHL**

Description:

This Macro Command is used to shift the contents of variable F1 to the left of the bit defined by variable F2, and then save the result to variable F0.

Syntax:

F0 = SHL(F1, F2)

Parameters	F0: Result, F1: Operator, F2: Offset			
Supported Variable Type		Constant	Internal Variable	External Variable
	F0		✔	
	F1	✔	✔	
	F2	✔	✔	

Example:

Variables:

Name: F0 ; **Address**: @R0 ; **Data Type**: WORD ; **Data Length**: 1

Name: F1 ; **Address**: @R100 ; **Data Type**: WORD ; **Data Length**: 1

F1 = 000FH ; F2 = 4 (constant)

Contents:

{F0} = SHL(F1, 4)

Result:

F0 = 00F0H

The following shows the left-shift operation performed on four bits.

Original: $000F_{(H)}$ = 0000 0000 0000 **1111**$_{(2)}$

Result: $00F0_{(H)}$ = 0000 0000 **1111** 0000 $_{(2)}$

- **SHR**

Description:

This Macro Command is used to shift the contents of variable F1 to the right of the bit defined by variable F2, and then save the result to variable F0.

Syntax:

F0 = SHR(F1, F2)

Parameters	F0: Result, F1: Operator, F2: Offset			
Supported Variable Type		Constant	Internal Variable	External Variable
	F0		✔	
	F1	✔	✔	
	F2	✔	✔	

Example:

Variables:

Name: F0 ; **Address**: @R0 ; **Data Type**: WORD ; **Data Length**: 1

Name: F1 ; **Address**: @R100 ; **Data Type**: WORD ; **Data Length**: 1

F1 = 1200H ; F2 = 8 (constant)

Contents:

{F0} = SHR(F1, 8)

Result:

F0 = 0012H

The following shows the right-shift operation performed on eight bits.

Original: $1200_{(H)}$ = **0001 0010** 0000 0000 $_{(2)}$

Result: $0012_{(H)}$ = 0000 0000 **0001 0010** $_{(2)}$

10.9.3 Bit Operation Macro Commands

- **BSET**

Description:

This Macro Command is used to set the state of variable F0 to ON. Note that only the Bit data type is supported.

Syntax:

F0 = (ON)

Parameters	F0: Result			
Supported Variable Type		Constant	Internal Variable	External Variable
	F0		✔	

Example:

Variables:

Name: F0 ; **Address**: @R0.5 ; **Data Type**: BIT ; **Data Length**: 1

F0 = 0

Contents:

{F0} = (ON)

Result:

F0 = 1

- **BCLR**

Description:

This Macro Command is used to set the state of variable F0 to OFF. Note that only the Bit data type is supported.

Syntax:

F0 = (OFF)

Parameters	F0: Result			
Supported Variable Type		Constant	Internal Variable	External Variable
	F0		✔	

Example:

Variables:

Name: F0 ; **Address**: @R0.5 ; **Data Type**: BIT ; **Data Length**: 1

F0 = 1

Contents:

{F0} = (OFF)

Result:

F0 = 0

- **BINV**

Description:

This Macro Command is used to reverse the contents of variable F1, and then save the result to variable F0. Note that only the Bit data type is supported.

Syntax:

F0 = BINV(F1)

Parameters	F0: Result, F1: Operator			
Supported Variable Type		Constant	Internal Variable	External Variable
	F0		✔	
	F1		✔	

Example:

Variables:

Name: F0 ; **Address**: @R0.5 ; **Data Type**: BIT ; **Data Length**: 1

F1 = 1 (constant)

Contents:

{F0} = BINV(F1)

Result:

F0 = 0

10.9.4 Data Transfer Macro Commands

- **MOV**

Description:

This Macro Command is used to assign the value of variable F1 to variable F0.

Syntax:

F0 = F1

Parameters	F0: Result, F1: Operator			
Supported Variable Type		Constant	Internal Variable	External Variable
	F0		✔	✔
	F1	✔	✔	✔

Example:

Variables:

Name: F0 ; **Address**: @R0 ; **Data Type**: WORD ; **Data Length**: 1

Name: F1 ; **Address**: @R100 ; **Data Type**: WORD ; **Data Length**: 1

F0 = 0

F1 = 1234

Contents:

{F0} = {F1}

Result:

F0 = 1234

COPY

Description:

This Macro Command is used to copy the contents of variable F1 to variable F0. Note that the data type and the length of the source and target variables must be the same.

Syntax:

COPY(F0, F1)

Parameters	F0: Target Variable, F1: Resource Variable			
Supported Variable Type		Constant	Internal Variable	External Variable
	F0		✔	✔
	F1		✔	✔

Example:

Variables:

Name: F0 ; **Address**: @R0 ; **Data Type**: WORD ; **Data Length**: 5

Name: F1 ; **Address**: @R100 ; **Data Type**: WORD ; **Data Length**: 5

F0 = { 0, 0, 0, 0, 0 }

F1 = { 10, 20, 30, 40, 50 }

Contents:

COPY(F0, F1)

Result:

F0 = { 10, 20, 30, 40, 50 }

- **FILL**

Description:

This Macro Command is used to fill the contents of variable F0 with the value from variable F1.

Note that if variable F0 is an Array, each element of the array will be set to the value of variable F1.

Syntax:

FILL(F0, F1)

Parameters	F0: Target Variable, F1: Fill Value			
Supported Variable Type		Constant	Internal Variable	External Variable
	F0		✔	✔
	F1	✔	✔	

Example:

Variables:

Name: F0 ; **Address**: @R0 ; **Data Type**: WORD ; **Data Length**: 5

F0 = { 10, 20, 30, 40, 50 }

F1 = 0 (constant)

Contents:

FILL(F0, 0)

Result:

F0 = { 0, 0, 0, 0, 0 }

10.9.5 Flow Control Macro Commands

- **IF**

Description:

This Macro Command is used to create a conditional function that can be used as an argument in any command that takes a function argument, where:

- If the result of the Condition is True, execute the Command.
- If the result of the Condition is False, o nothing and go to the end of the Command.

Syntax:

IF (Condition)

Command

Condition:

a. (F0==F1) Equal
b. (F0<>F1) Not Equal
c. (F0>F1) Greater Than
d. (F0>=F1) Greater Than or Equal
e. (F0<F1) Less Than
f. (F0<=F1) Less Than or Equal

Note: The IF command needs to be closed with the ENDIF command.

Command:

All other Macro commands.

Parameters	F0: Comparison Parameter, F1: Comparison Parameter			
Supported Variable Type		Constant	Internal Variable	External Variable
	F0	✔	✔	
	F1	✔	✔	

Example:

Variables:

Name: F0 ; **Address**: @R0 ; **Data Type**: WORD ; **Data Length**: 1

Contents:

```
{F0} = 200
IF ({F0} > 100)
   {F0} = 100
ENDIF
```

Result:

F0 = 100

- **ELSEIF**

Description:

This Macro Command is used to create a conditional function that can be used as an argument in any command that takes a function argument. The command is only executed if previous expressions in the IF command are False, where:

- If the result of Condition1 is True, execute Command1.
- If the result of Condition1 is False, then evaluate Condition 2.
- If the result of Condition2 is True, execute Command2.
- If the result of Condition2 is False, do nothing and end the evaluation.

Syntax:

IF (Condition 1)

Command 1

ELSEIF(Condition 2)

Command 2

Condition:

a. (F0==F1) Equal
b. (F0<>F1) Not Equal
c. (F0>F1) Greater Than
d. (F0>=F1) Greater Than or Equal
e. (F0<F1) Less Than
f. (F0<=F1) Less Than or Equal

Note: The IF command needs to be closed with the ENDIF command.

Command:

All other Macro commands

Parameters	F0: Comparison Parameter, F1: Comparison Parameter			
Supported Variable Type		Constant	Internal Variable	External Variable
	F0	✔	✔	
	F1	✔	✔	

Example:

Variables:

Name: F0 ; **Address**: @R0 ; **Data Type**: WORD ; **Data Length**: 1

Contents:

```
{F0} = 20
IF ({F0} > 100)
    {F0} = 100
ELSEIF({F0} < 50)
    {F0} = 0
ENDIF
```

Result:

F0 = 0

- **ELSE**

Description:

This Macro Command is used to create a conditional function that can be used as an argument in any command that takes a function argument. The function will execute a specified command if a particular condition is True and will execute a second command if the condition is false, where:

- If the result of the Condition is True, execute Command1.
- If the result of the Condition is False, execute Command 2 and end the evaluation.

Syntax:

```
IF (Condition)
    Command 1
ELSE
    Command 2
```

Condition:

a. (F0==F1) Equal
b. (F0<>F1) Not Equal
c. (F0>F1) Greater Than
d. (F0>=F1) Greater Than or Equal
e. (F0<F1) Less Than
f. (F0<=F1) Less Than or Equal

Note: The IF command needs to be closed with the ENDIF command.

Command:

All other Macro commands

Parameters	F0: Comparison Parameter, F1: Comparison Parameter			
Supported Variable Type		Constant	Internal Variable	External Variable
	F0	✔	✔	
	F1	✔	✔	

Example:

Variables:

Name: F0 ; **Address**: @R0 ; **Data Type**: WORD ; **Data Length**: 1

Contents:

```
{F0} = 50
IF ({F0} > 100)
    {F0} = 100
ELSE
    {F0} = 0
ENDIF
```

Result:

F0 = 0

- **ENDIF**

Description:

This Macro Command is used to mark the end of an IF command sequence. Note that the ENDIF command can only be used in conjunction with IF or IF ... ELSEIF commands and must always be used for every IF statement in the command otherwise an error will occur.

Syntax:

IF (Condition)

Command

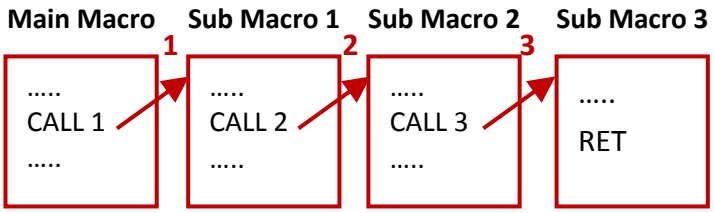
ENDIF

Note: The ENDIF command must always be used to close an IF command.

- **CALL**

Description:

This Macro Command is used to call a specified Sub Macro. A Sub Macro can be used to call Sub Macro itself or can also be used to call other Sub Macros. Recommend that not to call Sub Macro over three layers otherwise a program error may occur.



Syntax:

CALL(F0)

Parameters	F0: Sub Macro Number			
Supported Variable Type		Constant	Internal Variable	External Variable
	F0	✔	✔	

Example:

Variables:

Name: F0 ; **Address**: @R0 ; **Data Type**: WORD ; **Data Length**: 1

Contents:

```
CALL 5        ; Call Sub Macro number 5
CALL 0        ; No action taken as the Sub Macro number was invalid. Note that the base
                number for calling a Sub Macro is 1.
{F0} = 10
CALL {F0}     ; Call Sub Macro number 10
```

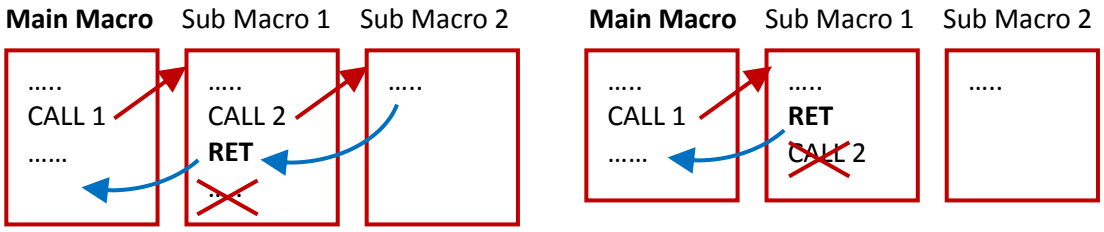
- **RET**

Description:

This Macro Command is used to instruct a Sub Macro to return to the original Macro, and is normally used in a Macro to indicate the end of the Macro.

Syntax:

RET



- **LABEL**

Description:

This Macro Command is used to assign a label for the program. Note that this command is used in conjunction with the GOTO command. See the example below.

Syntax:

LABEL:

Note that the LABEL can be any user-defined name.

- **GOTO**

Description:

This Macro Command is used to instruct the program to execute a command in the location specified by the LABEL command. Note that each GOTO command must be used in conjunction with a previously created LABEL command, and that LABEL must exist within the same Macro.

Syntax:

GOTO LABEL

Example:

Variables:

Name: var ; **Address**: @R0 ; **Data Type**: WORD ; **Data Length**: 1

Contents:

```
{var} = 0
LABEL1:    {var} = {var} + 1
    IF ({var} <> 100)
        GOTO LABEL1
    ENDIF
```

Note:
In this case, the program will perform **var = var+1** until **var** is equal to 100, so **var** = 0, 1, 2, …..99, and then perform **var** = 99+1 =100, then exit the program.

Result:

var = 100

- **FOR…NEXT**

Description:

This Macro Command is used to repeat a command the number of times specified by variable F0.

Note that the FOR…NEXT nested command can also be used.

Syntax:

FOR F0

    Command

NEXT

Parameters	F0: Loop Count, F1: Result			
Supported Variable Type		Constant	Internal Variable	External Variable
	F0	✔	✔	

Example:

Variables:

Name: F0 ; **Address**: @R0 ; **Data Type**: WORD ; **Data Length**: 1

Name: F1 ; **Address**: @R100 ; **Data Type**: WORD ; **Data Length**: 1

Contents:

{F1} = 0

{F0} = 10

For {F0}

    {F1} = {F1} + 1

NEXT

Result:

F1 = 55

- **DELAY**

Description:

This Macro Command is used to delay the execution time of the Macro or Sub Macro for the specified duration in milliseconds.

Syntax:

DELAY (F0)

Parameters	F0: Delay Time (ms)			
Supported Variable Type		Constant	Internal Variable	External Variable
	F0	✔	✔	

Example:

Variables:

Name: F0 ; **Address**: @R0 ; **Data Type**: WORD ; **Data Length**: 1

F0 = 100

Contents:

{F0} = 100

DELAY ({F0})

Note: In this case, the processing of the Macro will be delayed for 100 ms.

10.9.6 Screen Control Macro Commands

- **SHOWWINDOW**

Description:

This Macro Command is used to open an existing Window-type screen. This is useful when you want to display a dialog box on the HMI screen, see Section 8.4 Keyboard Objects for more details about the Window-type screen. Note that if a page number of Base-type screen is assigned, no action will be taken.

Syntax:

SHOWWINDOW(F0, 0)

Note: The second parameter is fixed as 0.

Parameters	F0: Page Number			
Supported Variable Type		Constant	Internal Variable	External Variable
	F0	✔	✔	

Example:

Variables:

Name: F0 ; **Address**: @R0 ; **Data Type**: WORD ; **Data Length**: 1

Contents:

{F0} = 3

SHOWWINDOW({F0}, 0)

Result:

A screen will be opened where the page number is “3” and the screen type is “Window”.

- **HIDEWINDOW**

Description:

This Macro Command is used to close an existing Window-type screen. See Section 8.4 Keyboard Objects for more details about the Window-type screen. Note that if a page number of Base-type screen is assigned, no action will be taken.

Syntax:

HIDEWINDOW(F0)

Parameters	F0: Page Number			
Supported Variable Type		Constant	Internal Variable	External Variable
	F0	✔	✔	

Example:

Variables:

Name: F0 ; **Address**: @R0 ; **Data Type**: WORD ; **Data Length**: 1

Contents:

{F0} = 3

HIDEWINDOW({F0})

Result:

A screen will be closed where the page number is “3” and the screen type is “Window”.

- **SHOWSCREEN**

Description:

This Macro Command is used to open an existing Base-type screen, e.g., the Home page. Note that if a page number of Window-type screen is assigned, no action will be taken. See Section 8.4 Keyboard Objects for more details about the Window-type screen.

Syntax:

SHOWSCREEN (F0)

Parameters	F0: Page Number			
Supported Variable Type		Constant	Internal Variable	External Variable
	F0	✔	✔	

Example:

Variables:

Name: F0 ; **Address**: @R0 ; **Data Type**: WORD ; **Data Length**: 1

Contents:

{F0} = 1

SHOWSCREEN({F0})

Result:

A screen will be opened where the page number is “1” and the screen type is “Base Screen”.

10.9.7 Note Macro Command

- **;**

Description:

This Macro Command is used to add a note or comment text to the Macro code.

Syntax:

; (comment)

Any text entered after the “;” character is considered as a note or comment.

Chapter 11 Other Functions and Optimized Design

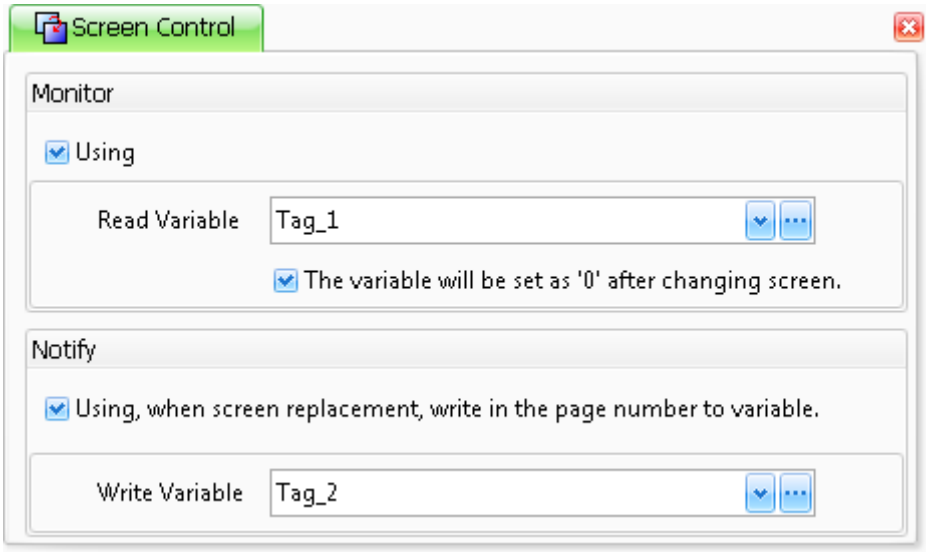
This section provides a description of the other important functions contained in the Project View panel, including the Screen Control, Password, Language and Image Manager objects. In addition, details of a number of optimized design considerations that can be implemented in a project are also provided.

11.1 Screen Control Functions

The **Screen Control** function is used to control aspects of the HMI screen, such as switching screens or accessing specific pages.

The Screen Control function uses a combination of Read and Write variables to switch between screens, and also to obtain the current page number.

For more details related to the usage of variables, see Chapter 7 Variables.



The following is an overview of the options available in the **Screen Control** page.

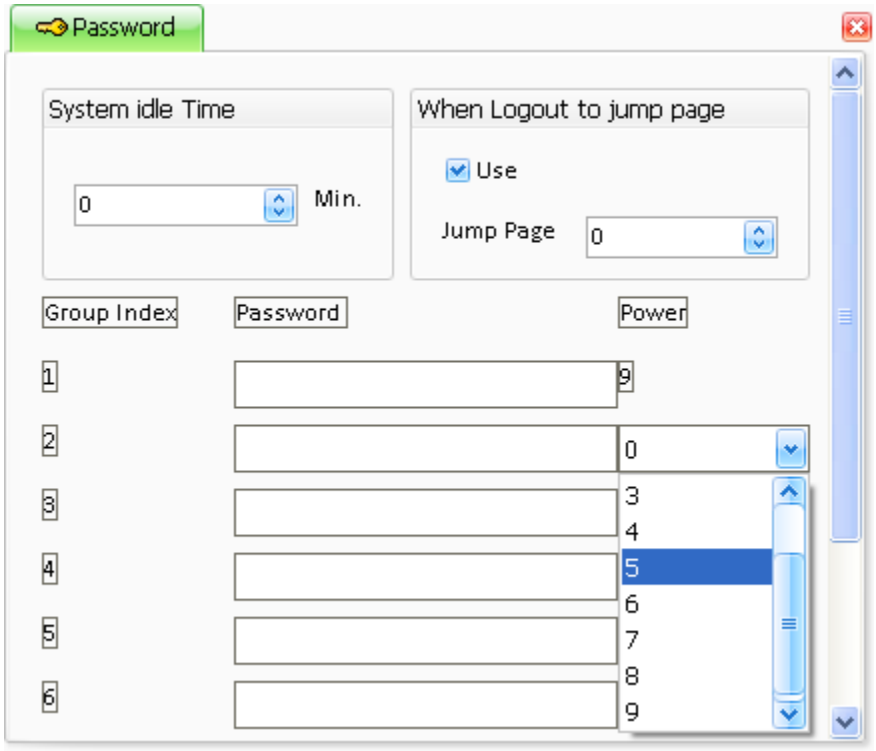
Monitor	**Use**	Used to enable whether or not to display an HMI screen specified by the value of a variable.
	Read Variable	Used to specify the Read variable that will be used to control the screen. See Chapter 7 Variables for more details
	Reset the variable to "0" after changing screens	Used to enable the function that specifies whether or not the Read variable is reset to 0 when changing to another screen.
Notify	**Use**	Used to specify whether or not the page number is written to the Write variable when changing to another screen.
	Write Variable	Used to specify the Write variable. See Chapter 7 Variables for more details

11.2 Password Function

The **Password** function is used to configure the different levels of password access permissions that can be specified to protect the functions in the project programs. A maximum of 10 password groups can be configured to provide a range of permission levels.

The lowest authority level is 0, which means that there are no restrictions on access permissions, whereas 9 is the highest authority and is intended for Admin level users.

If the authority level for the current user is not sufficient for the action to be performed, a login dialog box will automatically be displayed requesting that the user log in to the system using a password that has the required permissions.



The following is an overview of the options available in the **Password** page.

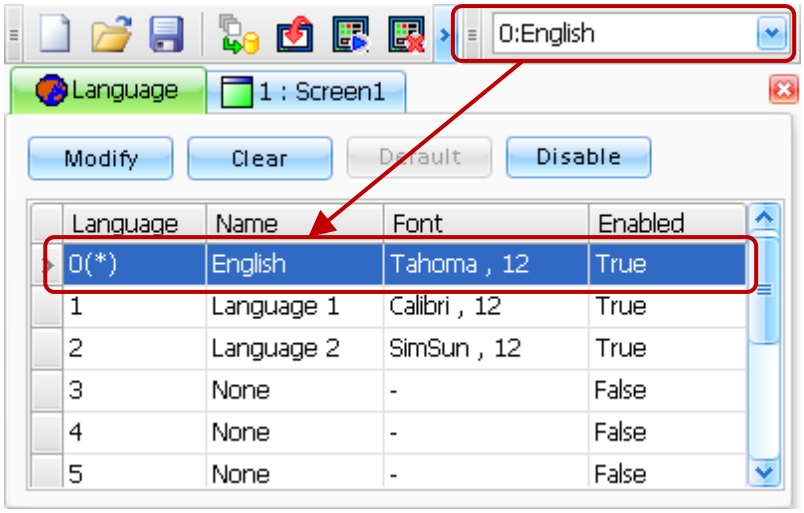
System Idle Time		Used to specify the duration of the system idle time in minutes. If there is no activity by the user within the specified period, the user will be automatically logged out from the system.
When Logout to jump page	**Use**	Used to enable the jump page function when the user is logged out
	Jump Page	Used to configure the page number to be displayed to when the user is logged out
Group Index		Used to indicate the index number of a password group. A maximum of 10 password groups can be configured for the different password authority levels.
Password		Used to specify the login password for a group
Power		Used to configure the authority level for a group

To configure the password, enter a password in the text field, and select a level from the drop-down menu. In addition, enter an appropriate system idle time and the page number, if necessary, in the respective fields.

11.3 Language

The **Language** function is used to configure the language options to be used for the HMI screens. The language can be selected from English, Traditional Chinese, Simplified Chinese, or other and a maximum of 10 different font style configurations can be assigned.

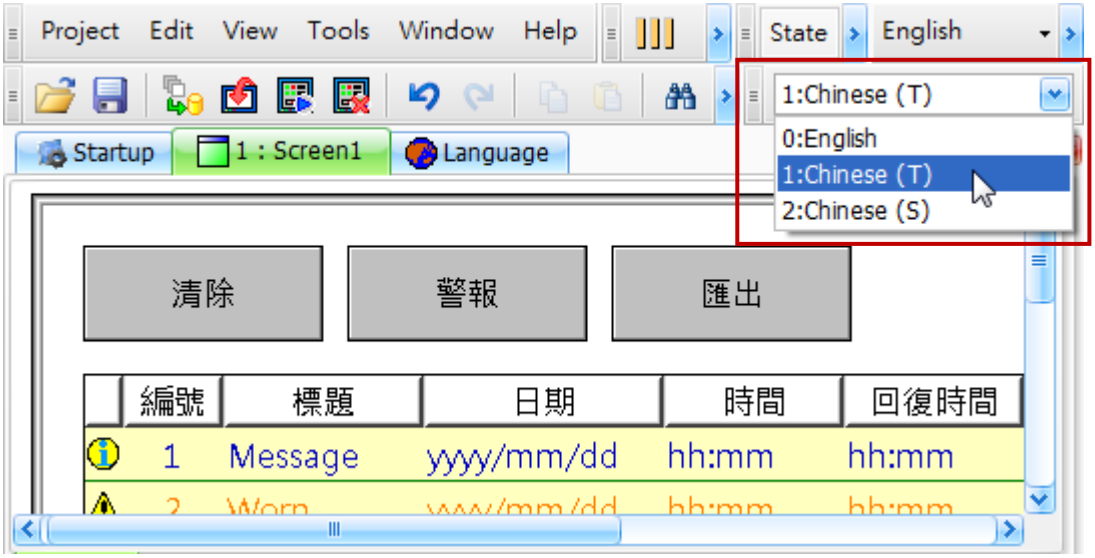
Note that you can edit the display text for any objects on the HMI screen after changing the language option.



The following is an overview of the options available in the **Language** page.

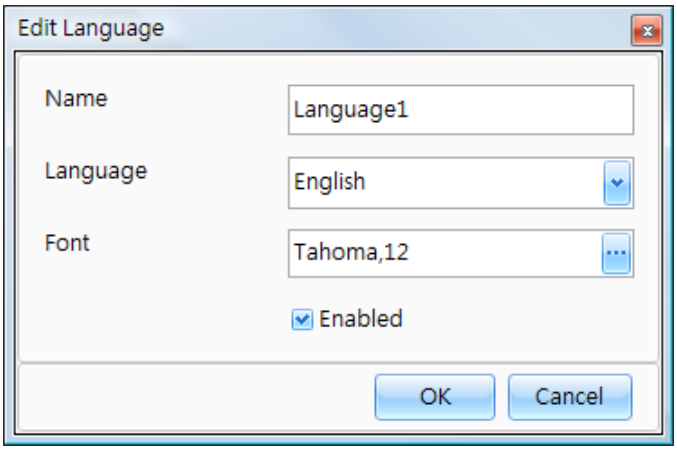
Modify	Used to modify the settings for the selected Language item
Clear	Used to clear the settings and parameters for the selected Language item
Default	Used to specify the default display language when the HMI screens is executing The default language will be indicated with an asterisk (*).
Enable/Disable	Used to enable or disable the selected language setting item

When the language option “1: Chinese (T)” is selected, the text on the HMI screen will be displayed in Chinese. You can also edit the text you want to display in the **Property View** panel of each object



After clicking the **Modify** button, the **Edit Language** dialog box will be displayed, allowing the language and style to be modified.

To configure the display language, enter a name for the Language in the **Name** field, and then select the language from the **Language** drop-down menu. Click the **Browse** button (**…**) from the **Font** text box to select the desired font and then click the **OK** button to save the changes.



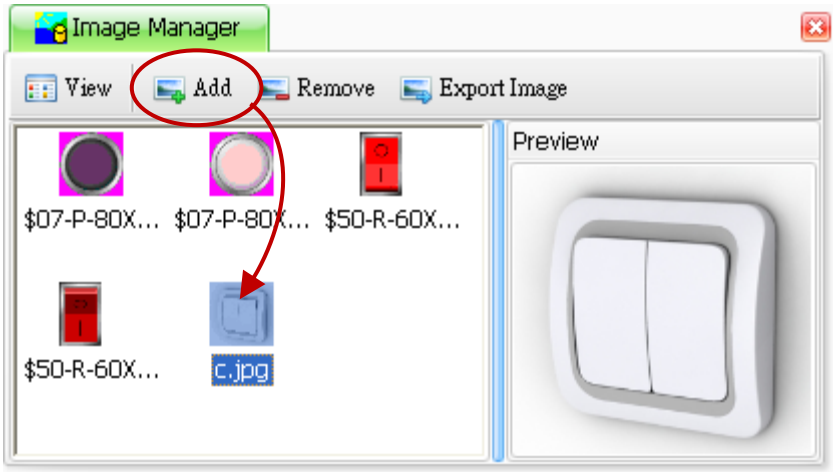
The following is an overview of the options available in the **Edit Language** dialog box.

Name	Used to assign a name for the language setting
Language	Used to specify the language to be displayed, which can be selected from English, Traditional Chinese, Simplified Chinese, or other
Font	Used to specify the style and size of the font
Enabled	Used to specify whether or not this language setting is enabled

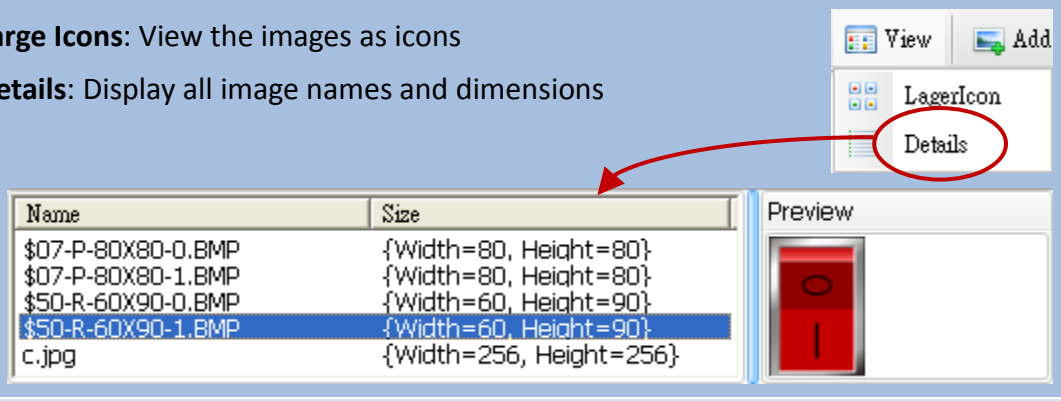
11.4 Image Manager Function

All the images used on the HMI screens will be displayed in the Image Manager page. In addition to the system default images that can be included in a Creator project, custom images can also be added, allowing a unique customized HMI project to be designed.

After clicking the Add button to add the custom image to the **Image Manager** page, you can load the image in the Picture properties dialog box of the Property View panel for the HMI object. See Section 8.5 (B) Picture properties for more details.

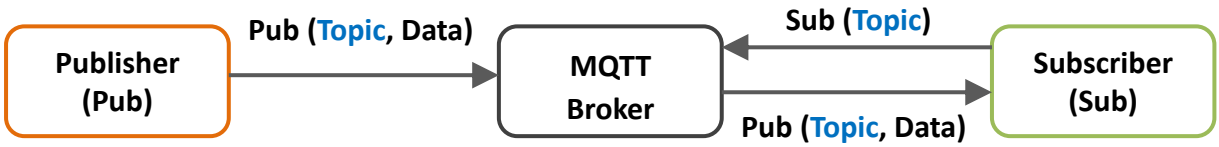


The following is an overview of the options available in the **Image Manager** dialog box.

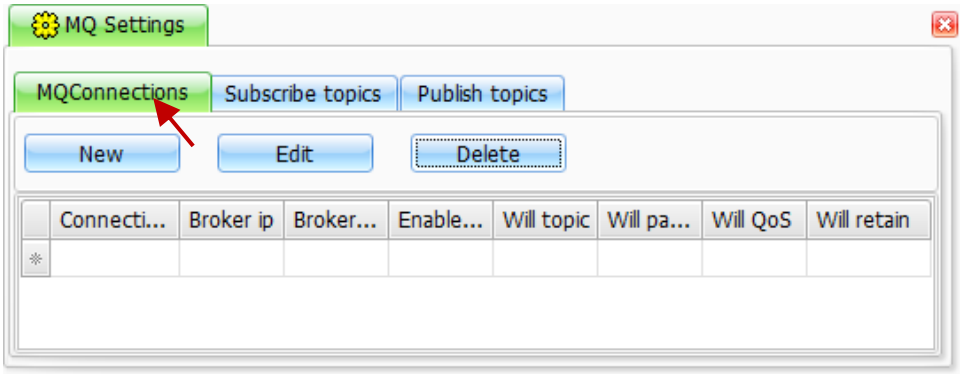
View	Used to specify the View Settings for the Image Manager and there are two options: **Large Icons**: View the images as icons **Details**: Display all image names and dimensions 
Add	Used to add an image to the Image Library. Note that only bmp and jpg format images are supported.
Remove	Used to remove the selected image from the library. Note that if a linked image is removed, i.e., an image that has been used in the project, it will no longer be displayed in the project.
Export Image	Used to export the selected image from the library

11.5 MQTT Function

The MQTT function in the Creator is only supported for the SmartView. MQTT is a machine-to- machine (M2M)/"Internet of Things" connectivity protocol. It was designed for publish /subscribe messaging between devices.



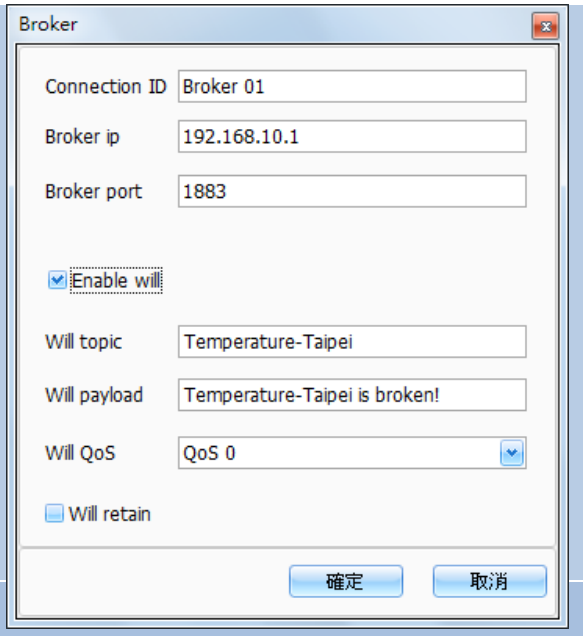
If a Publisher send a Topic message (e.g., "Temp1" , "25") to the Broker, all the Subscriber that has subscribed to this Topic (e.g., "Temp1") will receive this message.



The user needs to conduct the following settings when using MQTT communication:

- The Broker Settings

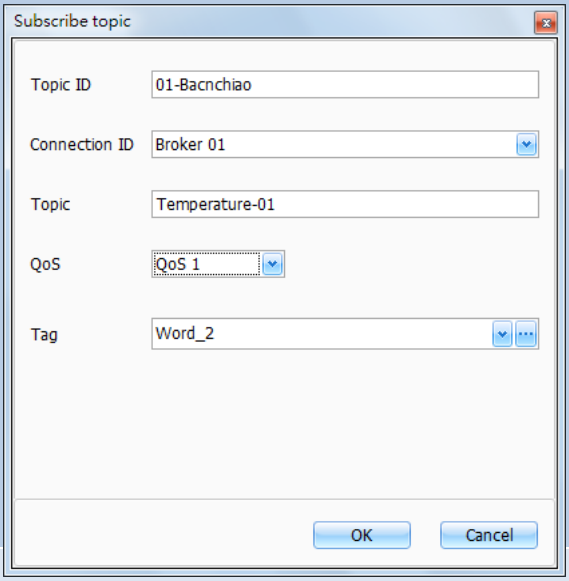
Connection ID	**Set the identifier for the Broker.**
Broker IP	Set the IP address of the Broker.
Broker Port	Set the Broker Port. (Defaults: "1883")
Enable Will	Enable the Will Mechanism.
Will Topic	Specify the topic for the Broker to publish when the Client is unexpectedly disconnect.
Will Payload	Set the topic message for the Broker to publish when the Client is unexpectedly disconnected.
Will QoS	Set the Quality of Service for the Will function. QoS 0: **At most once delivery.** Messages are delivered according to the best effort of the underlying network. No response and no retry mechanisms are defined in the protocol, so messages can get lost if the client unexpectedly disconnects or if the server fails, but, it's the fastest way to send a message.



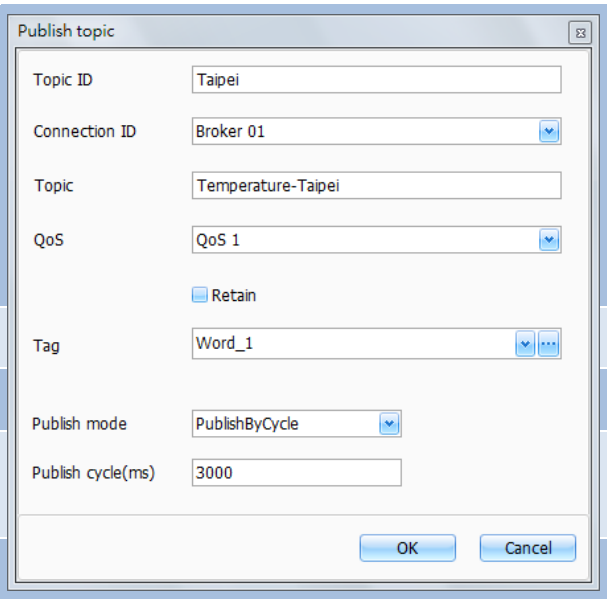
Will QoS	QoS 1: **At least once Delivery.** For this level of service, if there is no response from the Broker, the Client will resend to make sure the message is arrive but duplicates may occur. QoS 2: **Exactly once delivery.** This is the highest level of QoS. Additional protocol flows ensure that duplicate messages are not delivered to the receiving application.
Enable Retain	Whether to retain the Will message in the Broker.

- The Subscribe Topic Settings

Topic ID	**When the text of the subscribed topic is too long, the user can enter a short and easy-to-remember alias.** **Note: This field is required and the content must be unique. You can copy and paste the Topic text.**
Connection ID	Choose the needed Broker.
Topic	Set the subscribed topic.
QoS	Set the Quality of Service for the received message. QoS 0: **At most once delivery.** Messages are delivered according to the best effort of the underlying network. No response and no retry mechanisms are defined in the protocol, so messages can get lost if the client unexpectedly disconnects or if the server fails, but, it's the fastest way to send a message. QoS 1: **At least once Delivery.** For this level of service, if there is no response from the Broker, the Client will resend to make sure the message is arrive but duplicates may occur. QoS 2: **Exactly once delivery.** This is the highest level of QoS. Additional protocol flows ensure that duplicate messages are not delivered to the receiving application. **Note: The publisher and the subscriber can set different QoS on the same topic, but the real QoS depends on the lower level.** For example, if a client (Sub) subscribes a topic with **Qos 1**, then Client (Pub) with Qos **0:** It will send message with Qos **0.** Client (Pub) with Qos **2:** It will send message with Qos **1.**
Tag	Set the variable for receiving topic data.



- The Publish Topic Settings



Topic ID	When the text of the subscribed topic is too long, the user can enter a short and easy-to-remember alias. **Note:** This field is required and the content must be unique. You can copy and paste the Topic text.
Connection ID	Choose the needed Broker.
Topic	Set the published topic.
QoS	Set the Quality of Service for the received message. QoS 0: **At most once delivery.** Messages are delivered according to the best effort of the underlying network. No response and no retry mechanisms are defined in the protocol, so messages can get lost if the client unexpectedly disconnects or if the server fails, but, it's the fastest way to send a message. QoS 1: **At least once Delivery.** For this level of service, if there is no response from the Broker, the Client will resend to make sure the message is arrive but duplicates may occur. QoS 2: **Exactly once delivery.** This is the highest level of QoS. Additional protocol flows ensure that duplicate messages are not delivered to the receiving application. **Note: The publisher and the subscriber can set different QoS on the same topic, but the real QoS depends on the lower level.**
Enable Retain	Whether to retain the published message in the Broker.
Tag	Set the variable for sending topic data.
Publish mode	● Publish by cycle: Send periodically. ● Publish when tag value changed: Send if the data changes.
Publish cycle (ms)	Set the interval time if using publish mode 1.

11.6 Some Considerations for Optimizing the Design of a Project

When designing a project, there are a number of things that need to be taken into account to ensure that the final result performs in an optimal state. The following are some ideas that may help to improve the performance of the project.

- When using images, it would be preferable to avoid using the Auto Size functions to improve the performance of the project.
- It is recommended that the addresses for the variable are configured so that they are continuous as this helps to improve communication efficiency.
- It should be remembered that the more objects that exist on a screen, the slower the system speed will be. To mitigate the possibility of system overload, ICP DAS recommends that no more than 50 objects are present on a single screen.
- It is suggested that objects which need to be frequently updated are not placed so they overlap with other objects. This may help to improve the speed at which data is displayed.
- It is recommended that an appropriate update cycle time is specified for individual variables. For example, when measuring a temperature that changes slowly, it would be more beneficial to set the measurement cycle in the 500 to 1000 ms range. However, when monitoring the speed of a motor where the measurement changes rapidly, set the measurement cycle time to value in the range of 0 to 500 ms.
- When a macro needs to access an external variable, the system will immediately establish a connection, and the macro will not perform the next command until the external communication has been completed. Consequently, it is better to load the data of the external variables into the internal variables at once, and then perform any operations internally, as this can increase the efficiency of the macro.

Appendix 1: SmartView Operations

The following provides an overview of the operations that can be performed on the SmartView device, including how to execute a project either automatically or manually, together with details of the functions available in the Control Panel of the SmartView device.

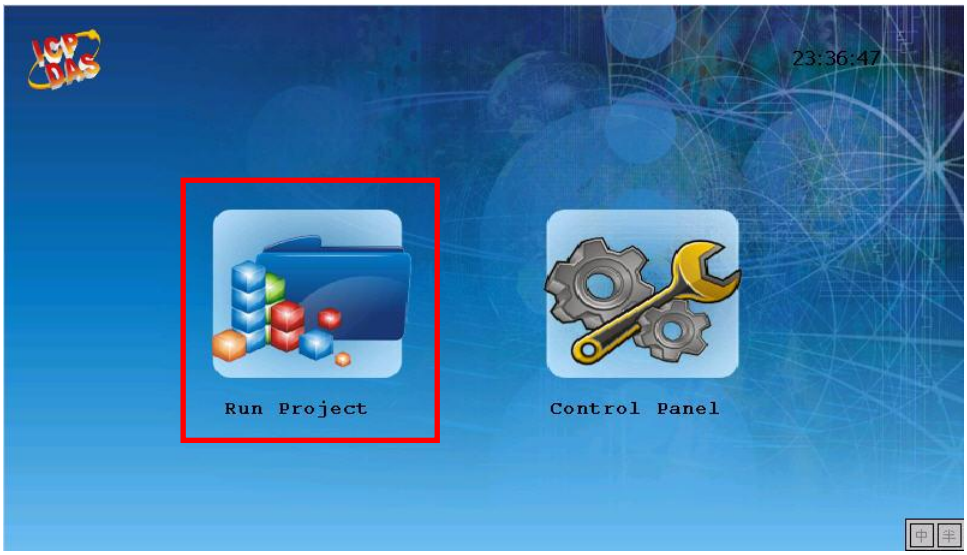
A. Executing a Project Automatically

By default, the SmartView will automatically load and run the project once it boots up.

Note: To prevent the project from automatically loading, tap anywhere on the screen as it is loading, as illustrated below.

B. Executing a Project Manually

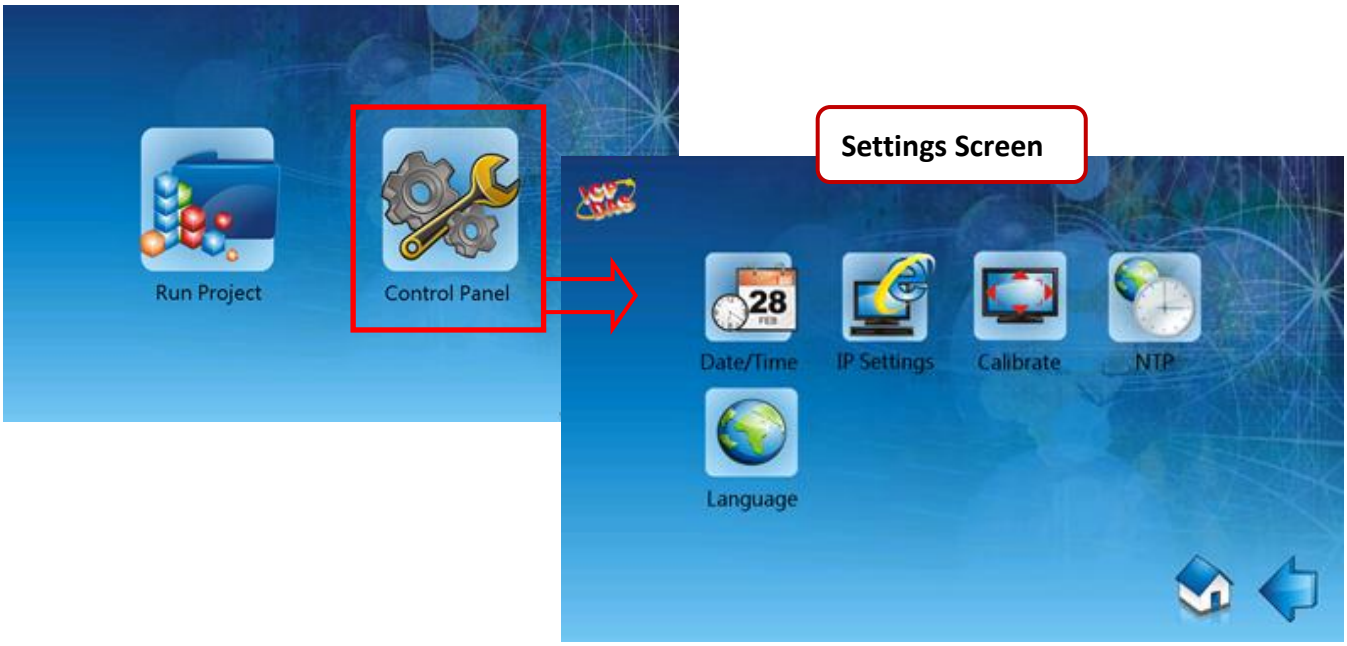
In some cases, you need to execute a project manually. For example, either after uploading a project to the SmartView, or after configuring most of functions in the Control Panel, the SmartView will not automatically run the project. Thus, you can tap the **Run Project** icon to manually execute a project.



C. Control Panel

The **Control Panel** is used to configure a variety of functions, each of which are described in more detail below, and include options such as changing the system date and time, configuring the IP settings and calibrating the sensitivity and accuracy of the touch screen, etc.

To access the Control Panel, tap the Control Panel icon and the settings screen will be displayed.



The following is an overview of the options available in the **Control Panel**.

	Used to adjust the system date and time settings
	Used to configure the IP address, etc. for the SV-x201 device
	Used to calibrate the sensitivity and accuracy of the touch screen
	Used to specify the NTP server and the time zone for the SV-x201 device
	Used to specify the language settings for the SV-x201 device

C.1 Date/Time Settings

The **Date/Time Settings** function is used to adjust the system date and time for the SV-x201 device.

To adjust the Date and Time Settings for the SV-x201 device, tap the **Date/Time Settings** icon in the **Control Panel** to open the Date/Time Settings screen.

After tapping the value you want to adjust, the Numeric Keyboard will be displayed as below. Input the value for the year (or month / day / hour / minutes / seconds) field, and then tap the **Enter** button to complete the setting, as illustrated in the figure below. Finally, click the **OK** button to apply the settings.

The following is an overview of the options available in the Date/Time Settings screen for the **Date/Time Settings** function in the **Control Panel**

Date	Used to adjust the system date
Time	Used to adjust the system time
OK	Used to apply the settings and exit the screen
Cancel	Used to exit the screen without saving the configuration settings

C.2 IP Settings

The **IP Settings** function is used to configure the IP address, Network Mask, Gateway address and DNS Server address for the SV-x201 device.

To configure the IP address, tap the **IP Settings** icon in the **Control Panel** to open the IP Settings screen. Enter the relevant details in the respective fields and then tap the **OK** button to apply the settings.

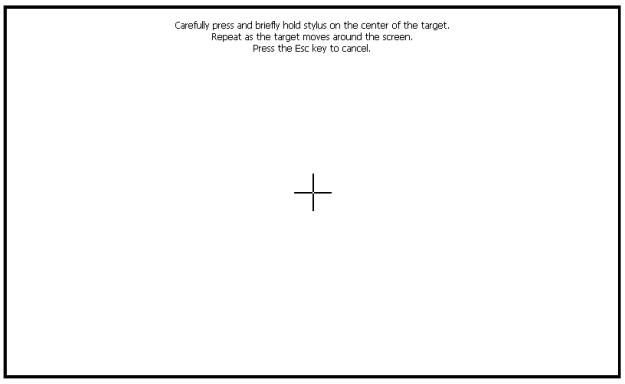
The following is an overview of the options available in the IP Settings screen for the **IP Settings** function in the **Control Panel**.

IP Address	Used to specify the IP address for the SV-x201 device
Network Mask	Used to specify the Network Mask for the SV-x201 device
Gateway	Used to specify the Gateway address for the SV-x201 device
DNS Server	Used to specify the address of the DNS Server for SV-x201 device
Obtain an IP address via DCHP	Used to specify whether or not the IP address for the project should be obtained via a DCHP server
OK	Used to apply the configuration settings and exit the screen
Cancel	Used to exit the screen without saving the configuration settings

C.3 Screen Calibration

The Screen Calibration function is used to calibrate the sensitivity of the touch screen and can be used to adjust the accuracy of the response to user input.

To calibrate the sensitivity and accuracy of the touch screen, tap the **Calibrate** icon in the **Control Panel** to open the Screen Calibration screen.



On the Calibration screen, tap and briefly hold the target (cross) in the center of the screen. Repeat this process as the target moves around the screen.

After completing the process, it will automatically return to the **Control Panel**.

C.4 NTP Server

The NTP (Network Time Protocol) Server can be used to automatically synchronize the system time of the SV-x201 device with a remote server.

To configure the NTP server, tap the **NTP** icon in the **Control panel** to open the NTP screen.

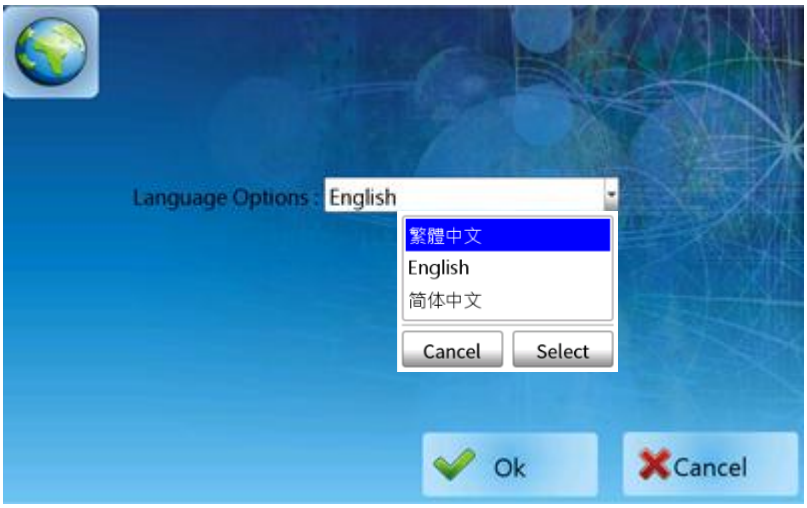
Choose one of the NTP server and the time zone, and check the Enable NTP checkbox and then tap the **OK** button to apply the settings.

The following is an overview of the options available in the NTP screen for the **NTP** function in the **Control Panel**.

NTP Server	Used to specify the NTP Server for updating the system time
Time Zone	Used to specify the Time Zone
Enable NTP	Used to enable the NTP function
OK	Used to apply the settings and exit the screen
Cancel	Used to exit the screen without saving the configuration settings

C.5 Language Settings

The **Language** function is used to configure the language used for the SV-x201 device and can be selected from English, Traditional Chinese, or Simplified Chinese.



To adjust the interface language, tap the **Language** icon in the **Control panel** to open the Language screen. Choose the desired language in the drop-down list, tap the **Select** button to select it, and then tap the **OK** button to apply the settings.

D. Exit the Project

To exit the project and return to the Home screen, follow the procedure described below:

1. Tap and hold the top left-hand corner of the screen (A).
2. Slide your finger to the bottom left-hand corner of the screen (B).
3. Release your finger to exit the project.