Delta DIAView SCADA System

User Manual

Smarter. Greener. Together.

www.deltaww.com

Contents

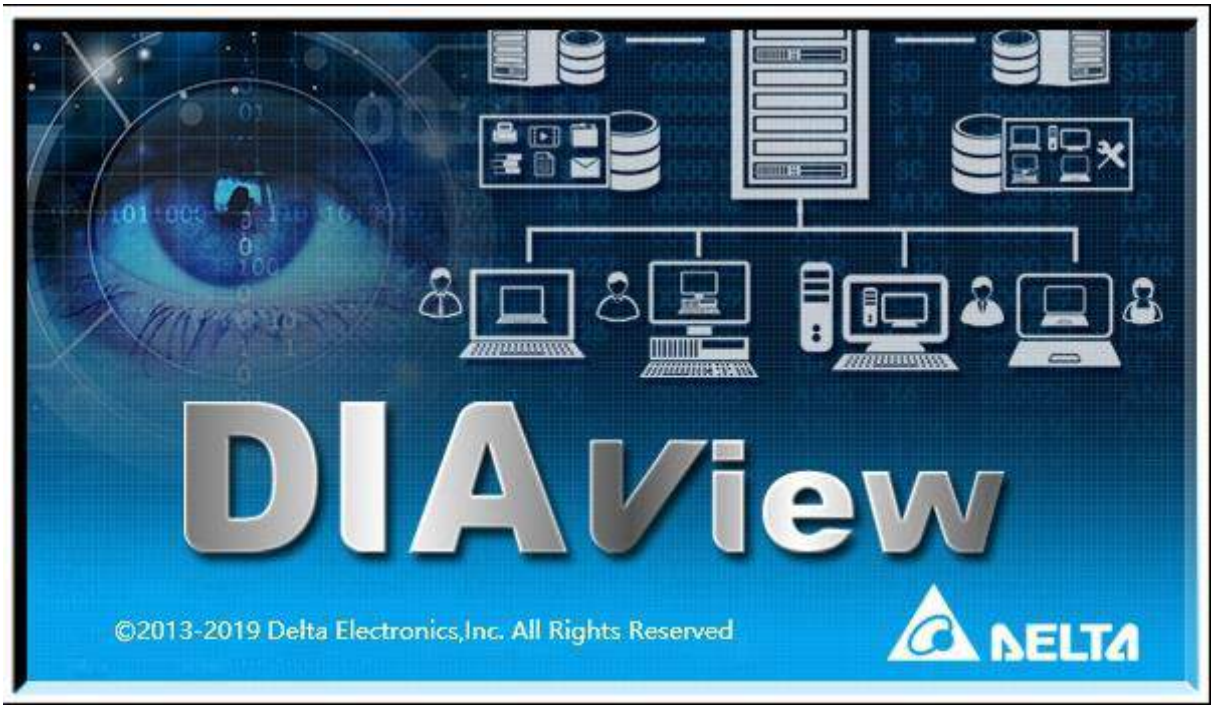
1. Delta DIAView SCADA introduction

1.Background

With the development of industrial and social economy, information technology based on computers and high speed Internet is becoming more and more mature and advanced; its application range includes industrial manufacturing, import and export trading, transportation, power energy, oil, chemical industry, metallurgy and the everyday life, culture and entertainment of people etc. Information technology is also gradually developing towards automation and intelligence, promoting productivity revolution and increasing people's living standards.

The SCADA (Supervisory Control And Data Acquisition) system is a data acquisition and monitoring system. It is a computer-based automated production process control and management system that can monitor and control the executing devices on-site in order to realize functions including data acquisition, device control, measurement, parameter control, various signal alarms and generate charts and reports etc. The SCADA system has significant effects for increasing the reliability, safety, economic benefits and work efficiency of industrial production and management control; it reduces personnel burden and labor, realizing automation and modernization.

Since releasing its first inverter in 1995, Delta continues to increase the reliability and precision of its products, and focused on innovative development of inverters and the power quality control, sensing, control and motion control fields. As its product lines become more and more complete, Delta also integrated the PLC, HMI, inverter, network communication devices, video and other industrial automation products manufactured by itself to provide comprehensive industrial automation solutions. With the gradual promotion and global distribution of Delta's own brand development strategy, and to further increase Delta's influence in the industrial automation field, adapt to the customers' need for Delta's industrial automation products and market development features, Delta developed its own brand of SCADA system - - the Delta Industrial Automation View system, abbreviated as DIAView.



2.Function features

- Based on the Windows operating system with wide, flexible and reliable applications.
- Supports the communication and networking of domestic and international mainstream PLC, DCS, PAC and IPC devices.
- Based on the .NET Framework; uses WPF (Windows Presentation Foundation) technology to display beautiful images and fully supports the XML technology.
- Friendly user interface, rich graphic elements, easy operations, easy configurations, easy usage and easy maintenance.
- Distributed architecture with powerful scalability.
- Stable industrial communication design.
- Powerful alarm management.
- Easy-learning and easy-to-use scripting language (VBScript).
- Highly efficient recipe management.
- Reliable user management.

2. DIAView Installation

2.1 DIAView configuration requirements

Configuration item	Minimum HW/SW system requirement
CPU	2GHz or above
Memory	2GB or above
Hard disk	20GB or above
Display	1024 x 768 resolution or above
Operation system	Windows 7 32bit/64bit (Professional/Ultimate/Enterprise)
	Windows 8 32bit/64bit (Professional/Ultimate/Enterprise)
	Windows 10 32bit/64bit (Professional/Ultimate/Enterprise)
	Windows Server 2008 32bit/64bit
	Windows Server 2012 64bit
	Language: English,Simplified Chinese, Traditional Chinese
System Privilege	Windows users must have administrator privileges
Execution Platform	Installation of Microsoft .NET Framework4.5 or above is required
Database	SQL Server 2008 R2 or above

Note : Users must prevent the computer system from entering standby modes or dormant state while DIAView runtime environment executes in order to prevent "Unrecognized Encryption -dongle" exception.

2.2 DotNet installation

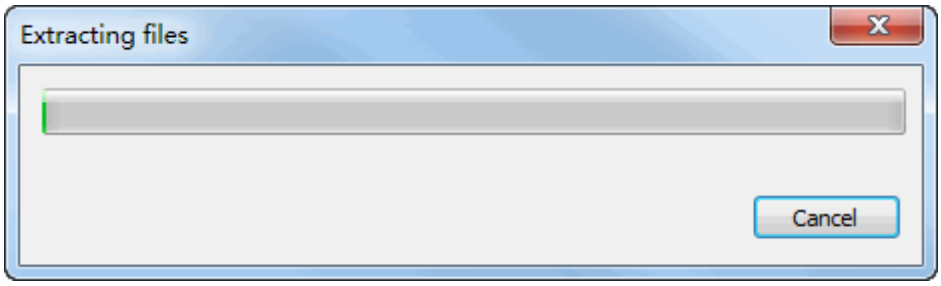
DIAView is developed based on the Microsoft .NET Framework environment; .NET Framework4.0 or above must first be installed before using DIAView. Installation steps are as follows (Taking Windows7 Professional Edition as the example):

Step 1: Start up the computer, click to start the DIAViewSetup.exe installer, as shown in the figure below:

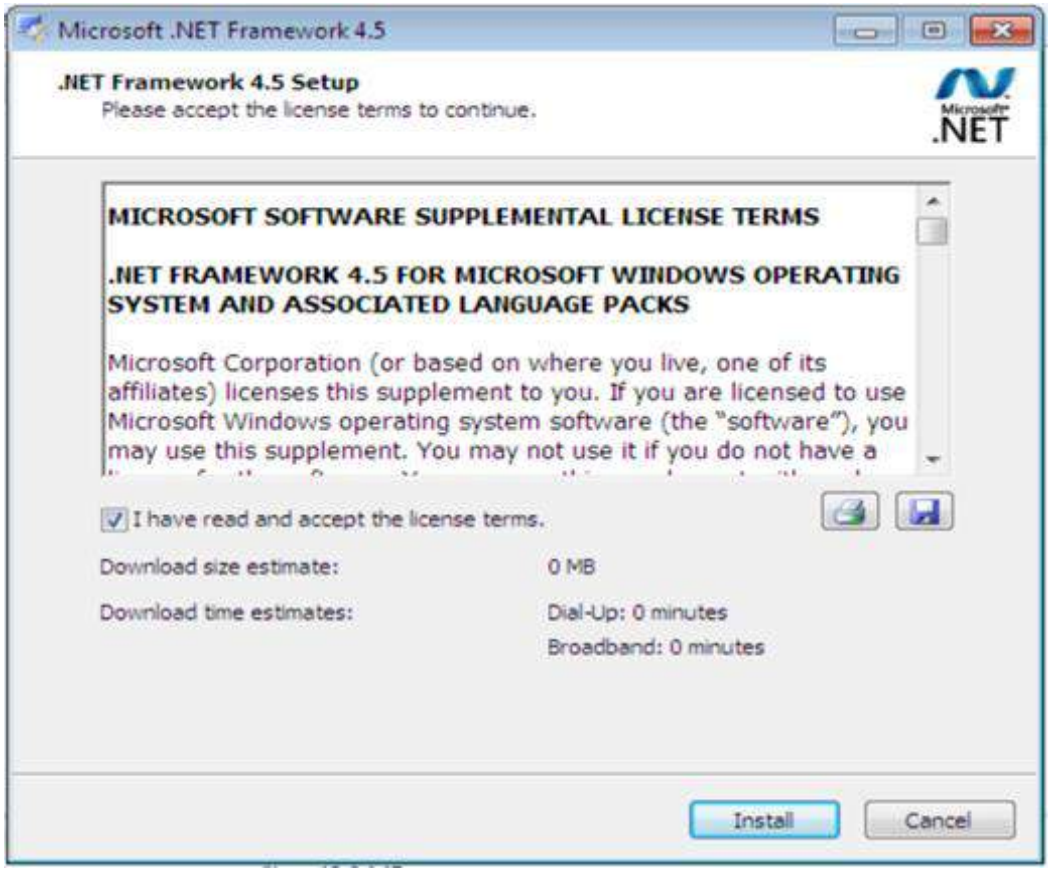
Once the installation program has started, it will enter the install navigation interface; there is a list of buttons on the right of the install navigation interface, when you move the mouse cursor onto one of the buttons, the text of the button will change from white to yellow. The functions of each button are as follows:

- Installation guide: Includes the system requirements for installing DIAView, problems that might encounter during the installation and the features of the new version.
- DIAView Full: Installing DIAView main program or DIAView Client.
- DIAView Client: Installing DIAView Client.
- SenseLock (Encryption -dongle): Installs an encryption -dongle driver.
- .Net Framework: Installs Microsoft .NET Framework (version 4.5 and above).
- OPCServer:Installing OPCServer.

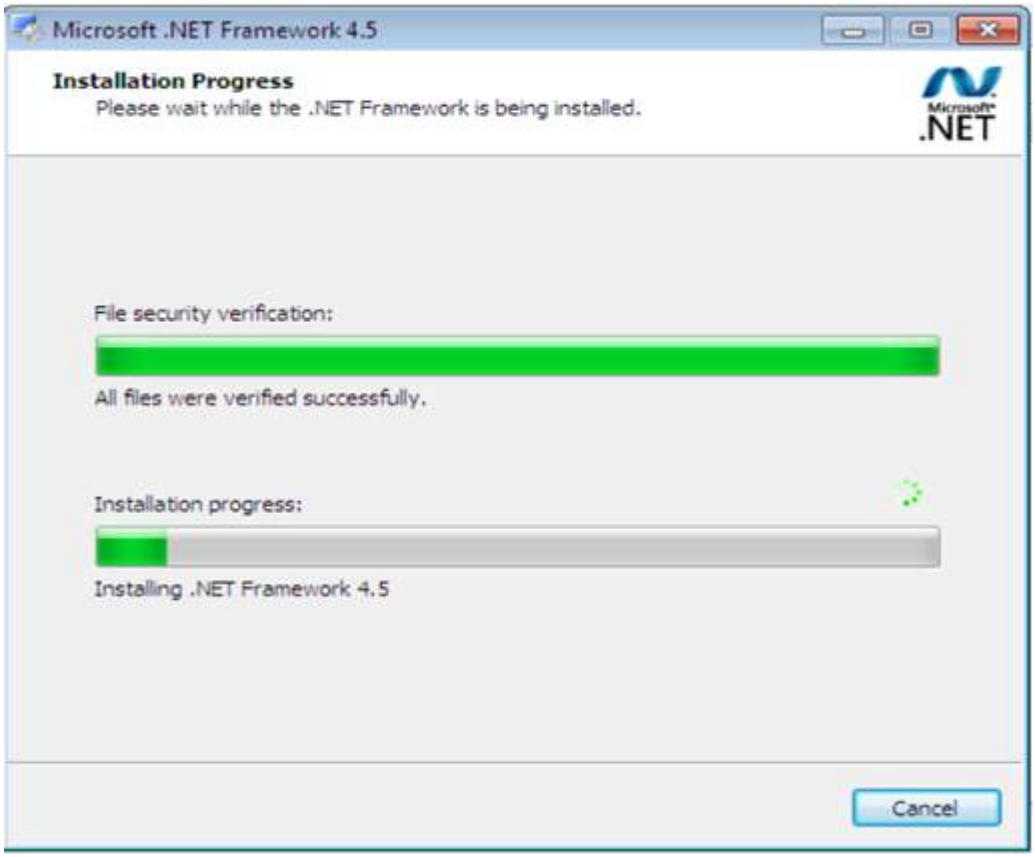
Step 2: Click the ".Net" button to enter the installation interface of Microsoft .NET Framework, as shown in the figure below:



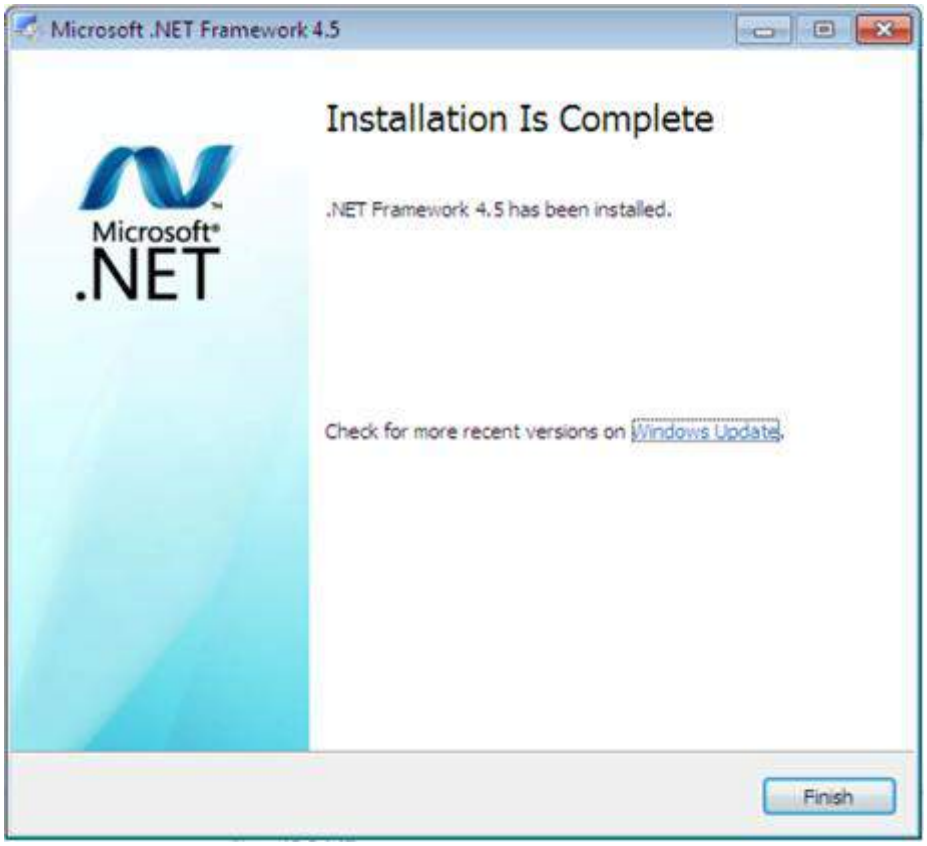
Step 3: The “-License Terms-” dialog will appear as shown in the figure below. If the user agrees to the terms in this agreement, check the “-I have read and accept the license terms-” checkbox and then press the “-Install-” button to continue the installation. If user does not agree, press the “-Cancel-” button to exit the installation program:



Step 4: Select accept license agreement and then press the “-Install-” button to continue installation and enter the installation stage:



Step 5: When installation is complete, the dialog shown in the figure below will appear; click the "-Finish-" button to complete the installation of -.NET Framework:

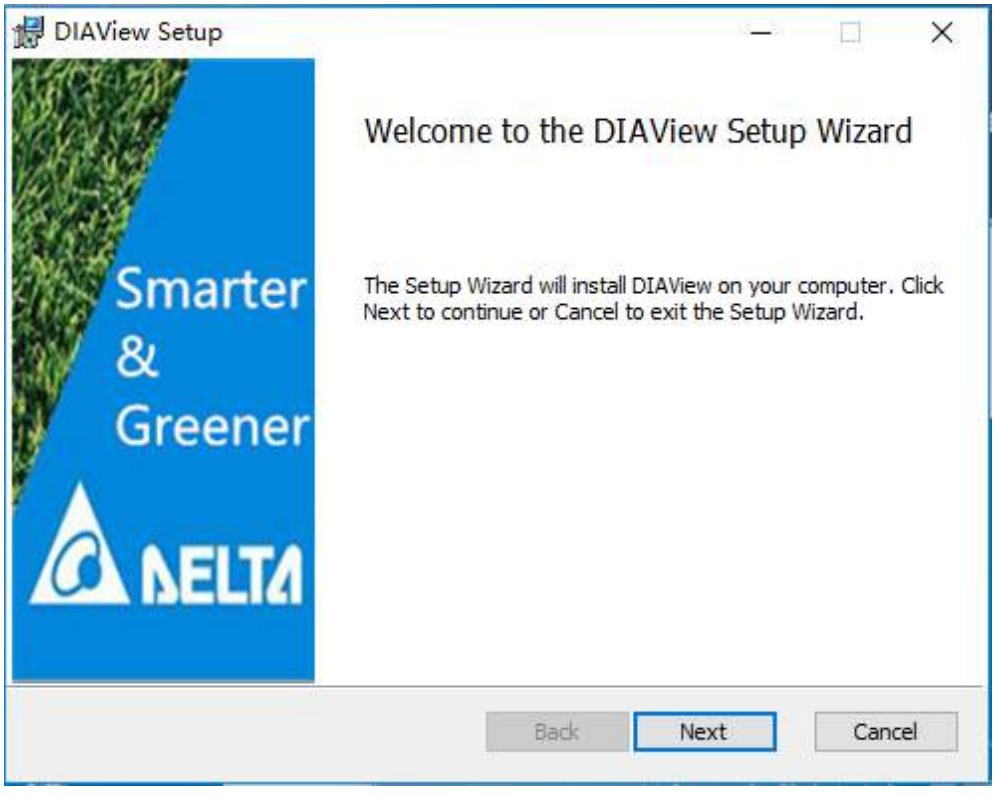


2.3 Install DIAView

Installation steps are as follows (Using Windows7 Professional Edition as the example):

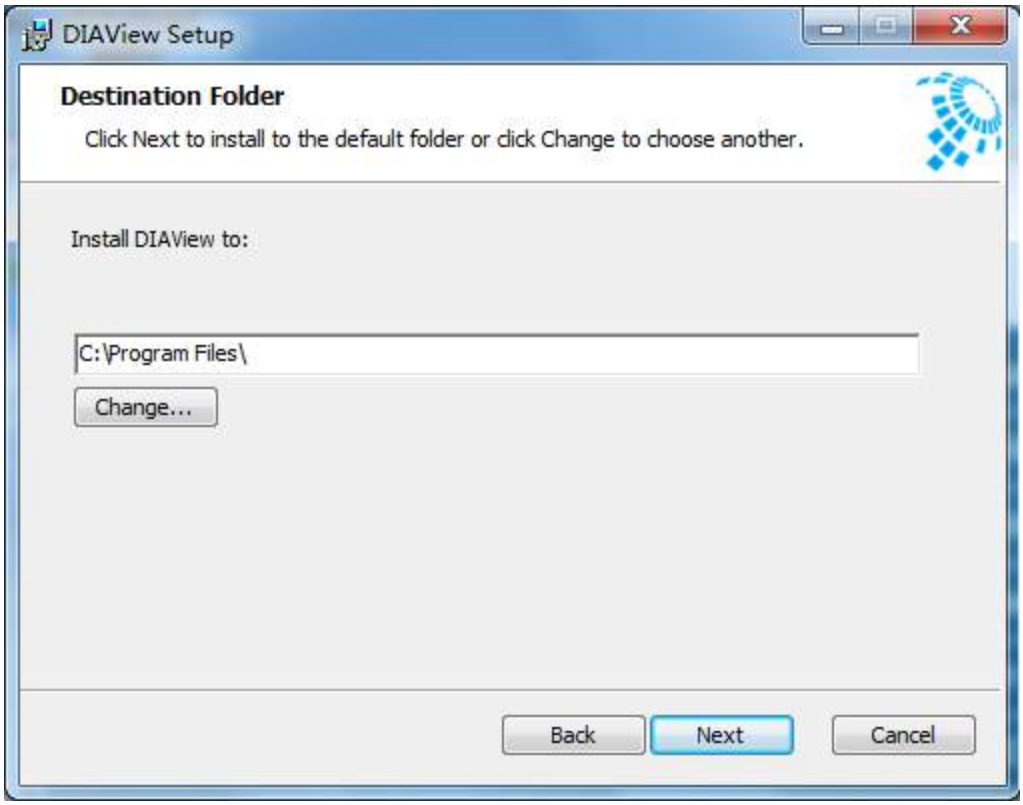
Step 1: Start the computer and click the DIAViewSetup.exe file to execute the installation program, as shown in the figure below:

Step 2: Click the “DIAView Full” or “DIAView Client” button to enter the DIAView installation interface, as shown in the figure below:



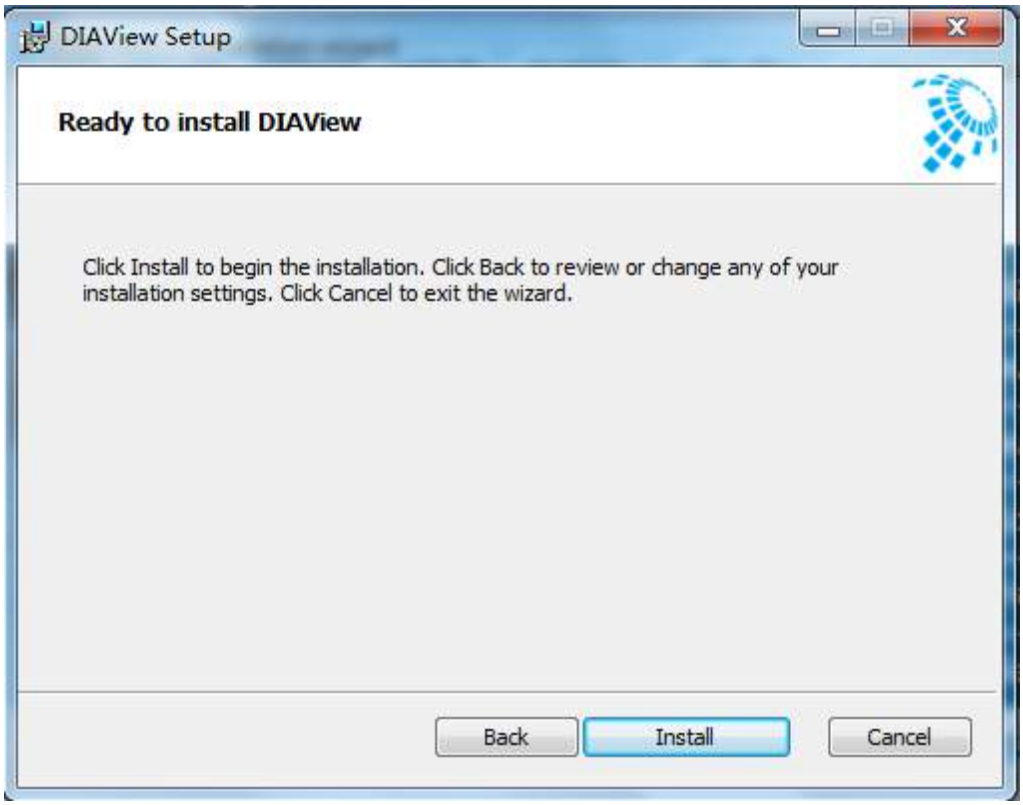
Step 3: Click “-Next-” and the “-User License Agreement-” dialog will appear. The content of the dialog is the legal agreement between “-Delta Electronics,Inc.” and the DIAView software user. If the user agrees to the terms in this agreement, check the “-I accept the terms in the license agreement-” checkbox and then press the “-Next-” button to continue the installation; users can also press the “-Print-” button to print this agreement. If user does not agree, press the “-Cancel-” button to exit the installation program:

Step 4: Select accept license agreement and then press the “-Next-” button to continue installation; the select installation folder dialog will appear:

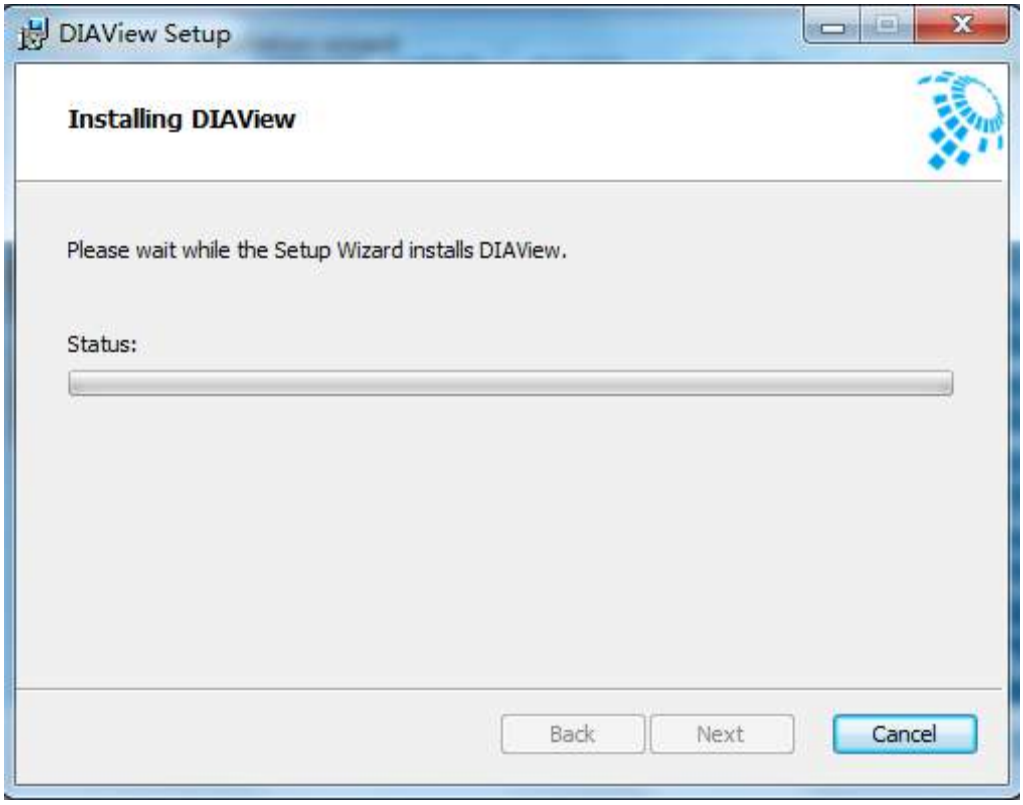


The system's -default installation path is: C:\Program Files\DIAView\, if the user wants to change the installation folder, press the "-Change-" button to select the installation path.

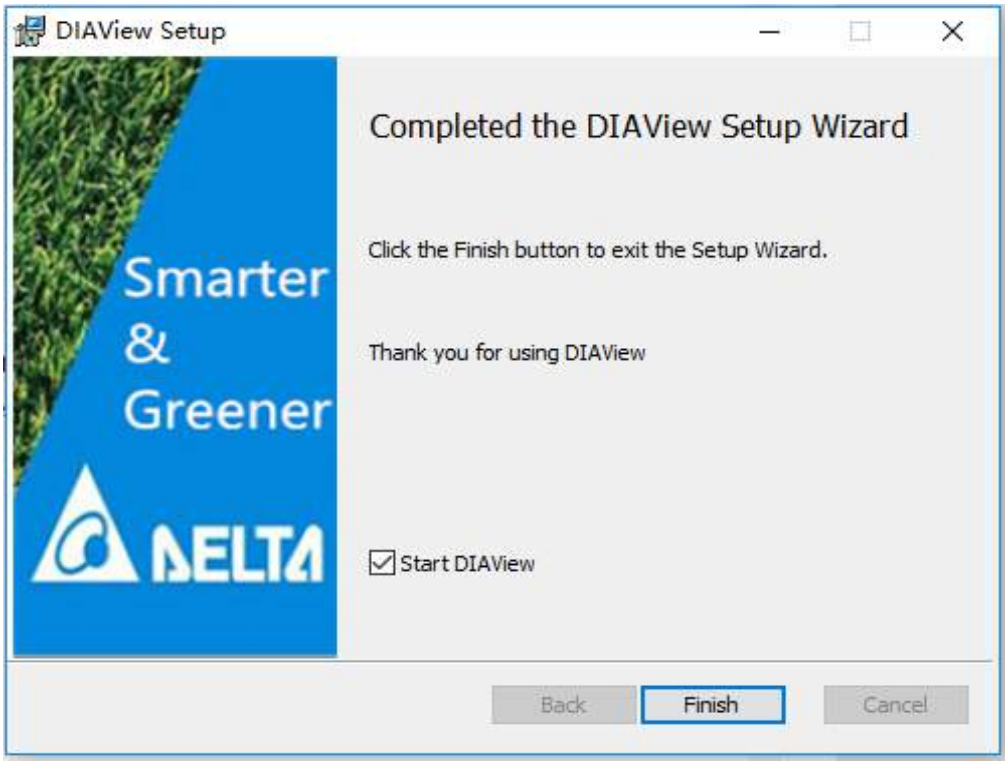
Step 5: Press "-Next-" and the installation program will analyze the installation environment and then the following window will appear:



Step 6: Click “-Install-” to start installing the DIAView main program; the installation progress window will appear:



Step 7: When installation is complete, the dialog shown in the figure below will appear; click the "-Finish-" button to complete the installation of the DIAView software. If user checked the "-Start DIAView-"checkbox, then the DIAView software will start when the "-Finish-" button is pressed:

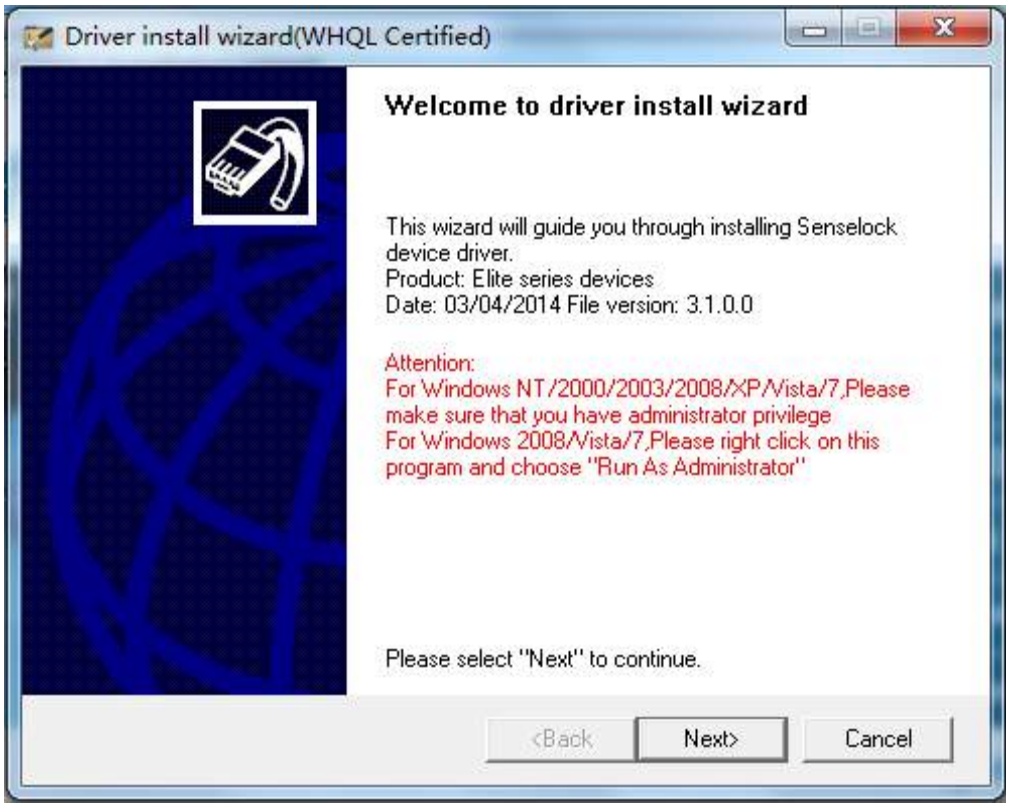


2.4 Install senselock driver

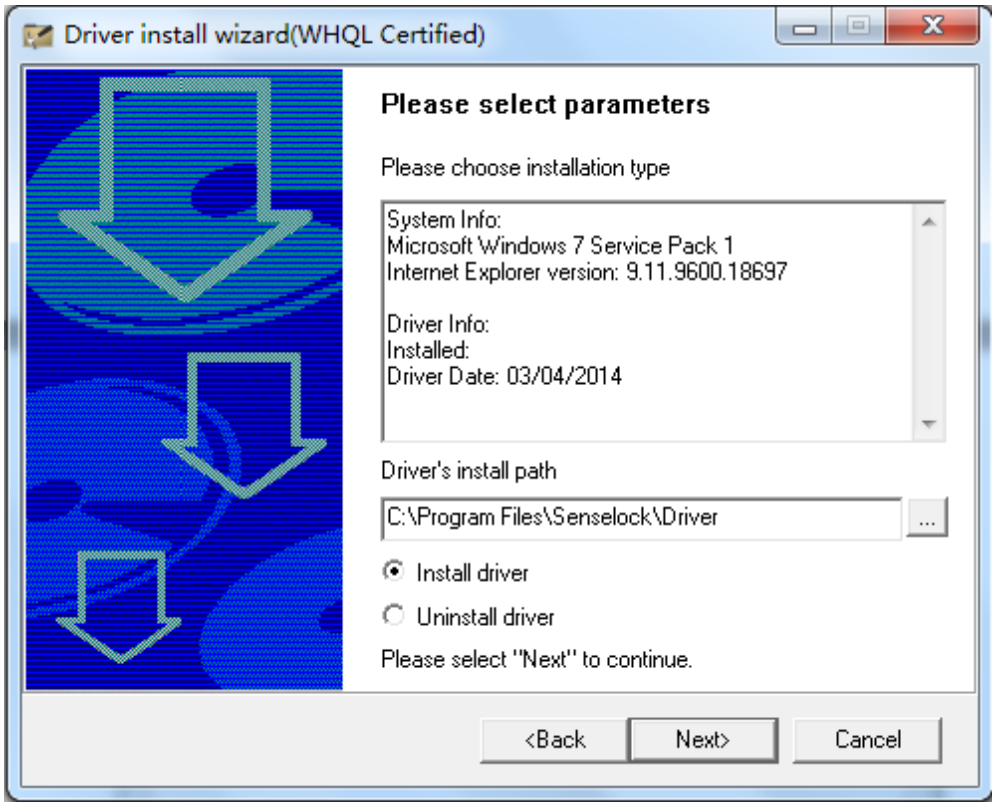
Installation steps are as follows (Using Windows7 Professional Edition as the example):

Step 1: Start the computer and click the DIAViewSetup.exe file on the disc to execute the installation program, as shown in the figure below:

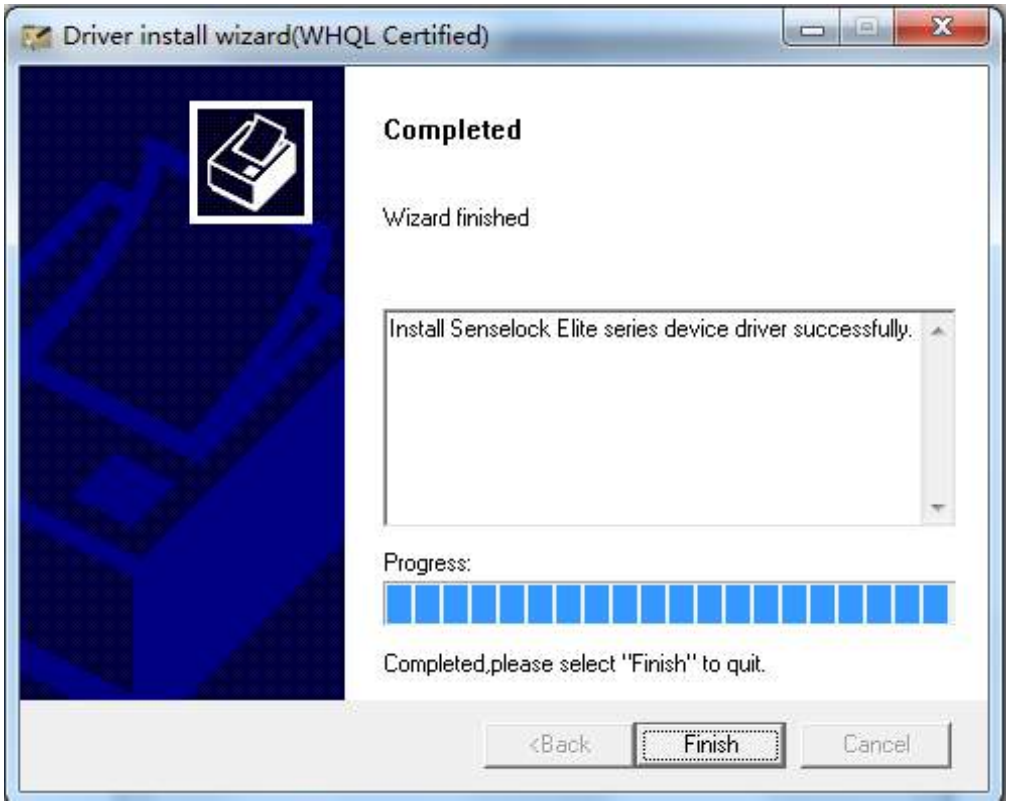
Step 2: Click the Senselock button to enter the installation navigation interface, as shown in the figure below:



Step 3: Click “-Next-” and select install parameters:



Step 4: Click the “-Next-” button to enter the program installation stage; when installation is complete, the screen will appear as shown in the figure below:



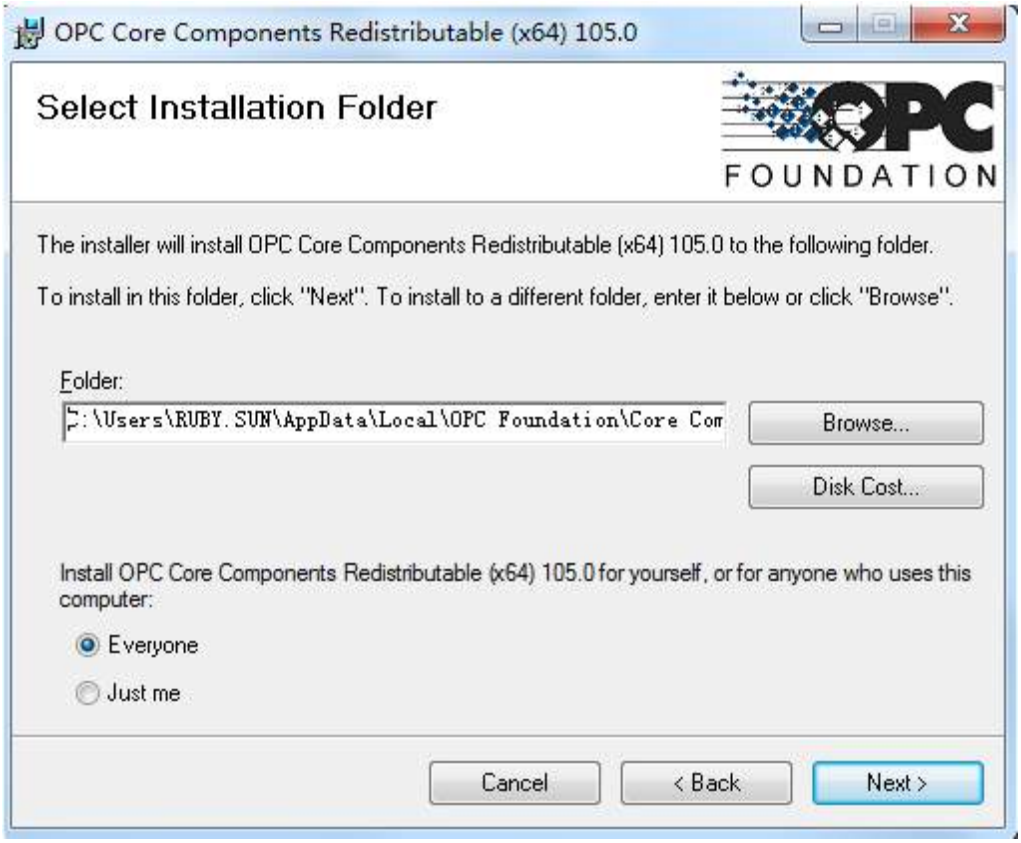
2.5 Install OPC server

Installation steps are as follows (Using Windows7 Professional Edition as the example):

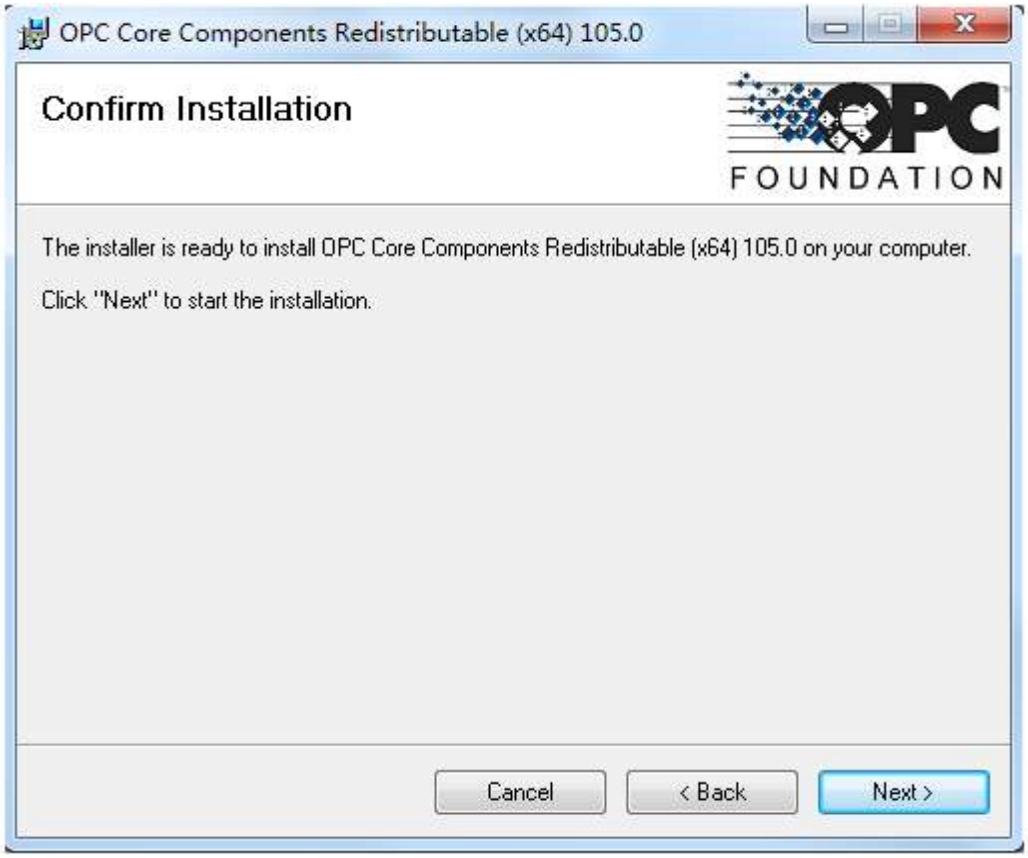
Step 1: Start the computer and click the DIAViewSetup.exe file to execute the installation program, as shown in the figure below:

Step 2: Click the “OPC Server” button to enter the OPC Server installation interface, as shown in the figure below:

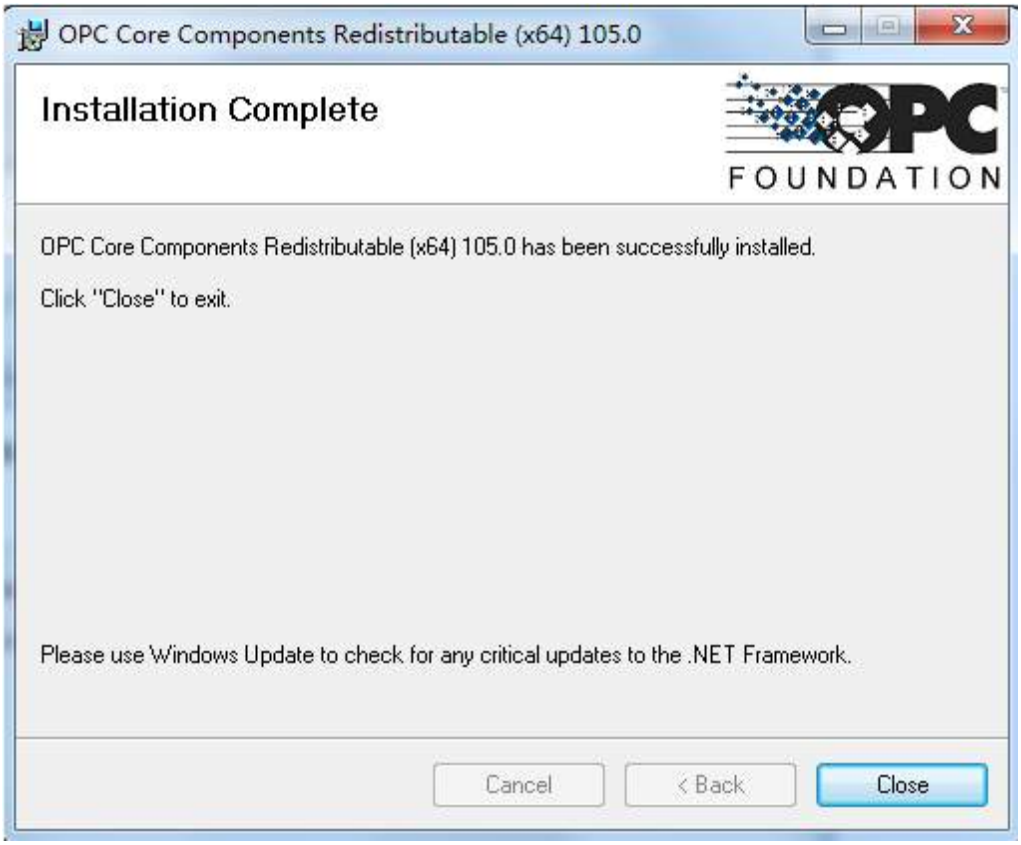
Step 3: Select "I Agree" then click "next" button,enter the following dialog:



Step 4: Click "next" button,enter the following dialog:



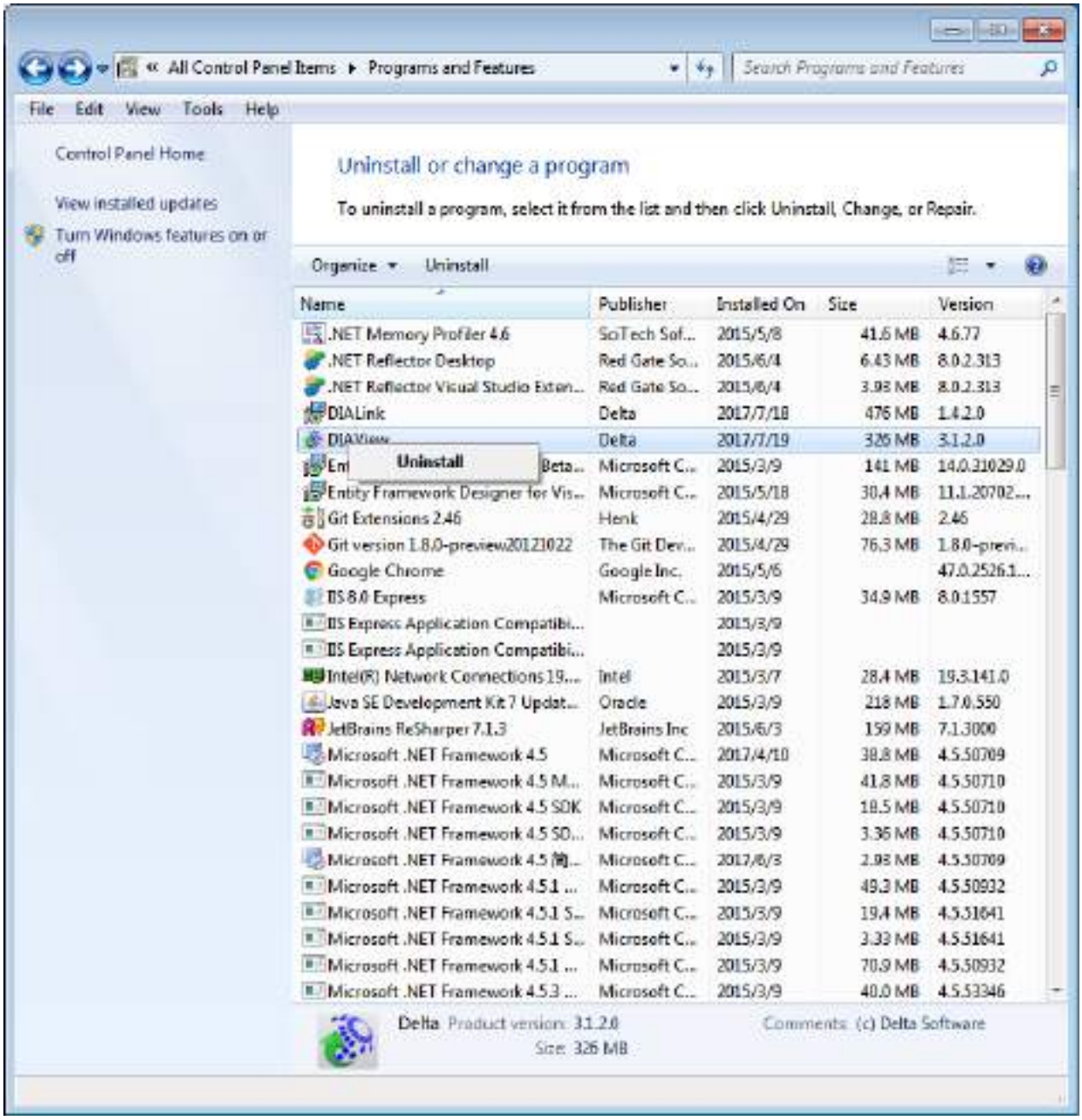
Step 5: Click "next" button,enter the installation program, when installation is complete, the screen will appear as shown in the figure below:



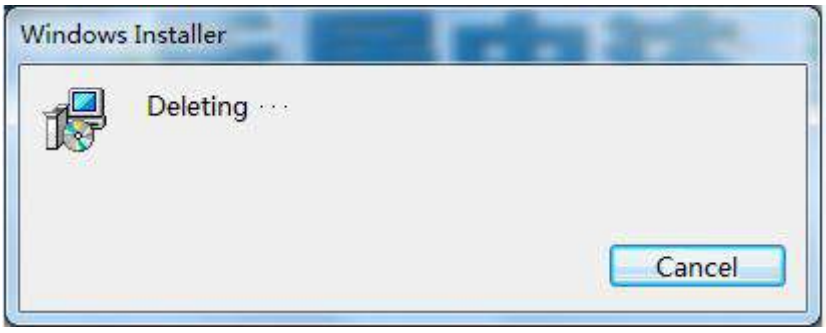
2.6 Uninstalling DIAView main program and senselock driver

1. There are three ways to uninstall the DIAView main program; the steps are as follows (using Windows 7 Professional Edition as the example):

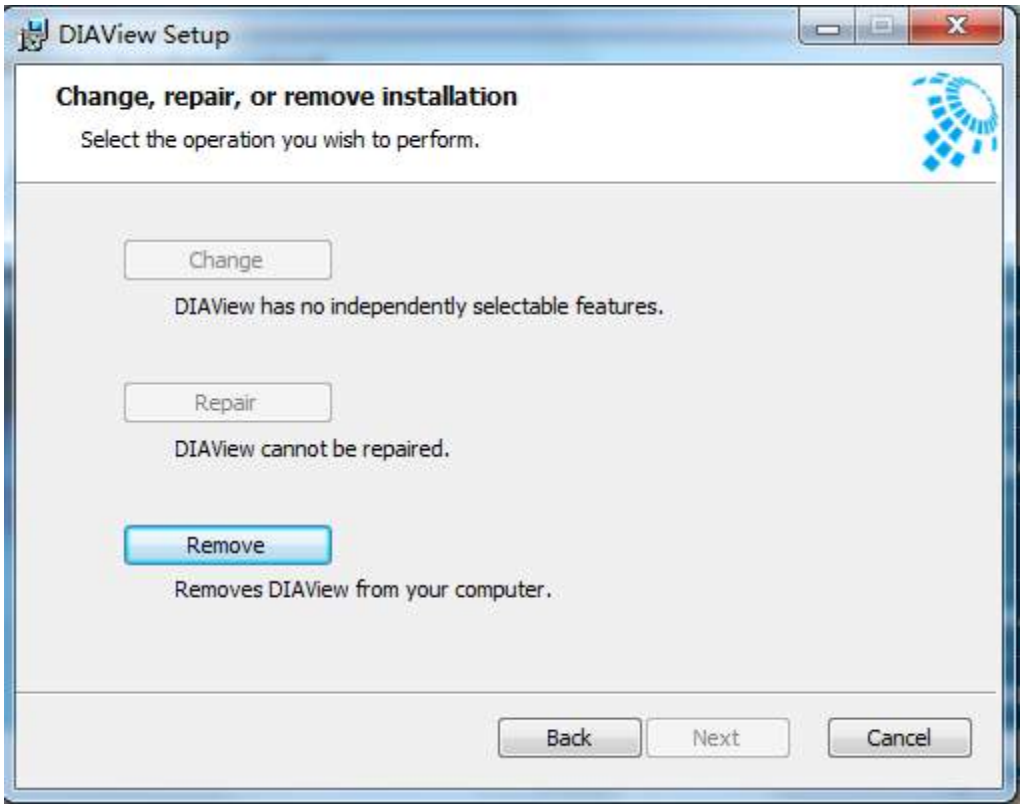
Method 1: Go to the “Control Panel” and select “Programs and Features,” then select “DIAView” and use your mouse to right-click on it. The right-click menu will appear as shown in the figure below:



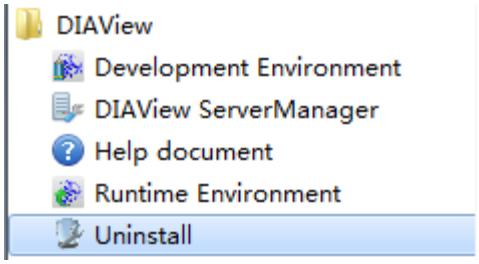
Select “-Uninstall-” from the right-click function menu and the system will automatically uninstall the program, as shown in the figure below:



Method 2: Click the DIAViewSetup.exe file on the disc to execute the installation program. Select “DIAView” from the navigation window that pops up, and then select “-Delete-” from the next screen that appears; the system will automatically uninstall the program:

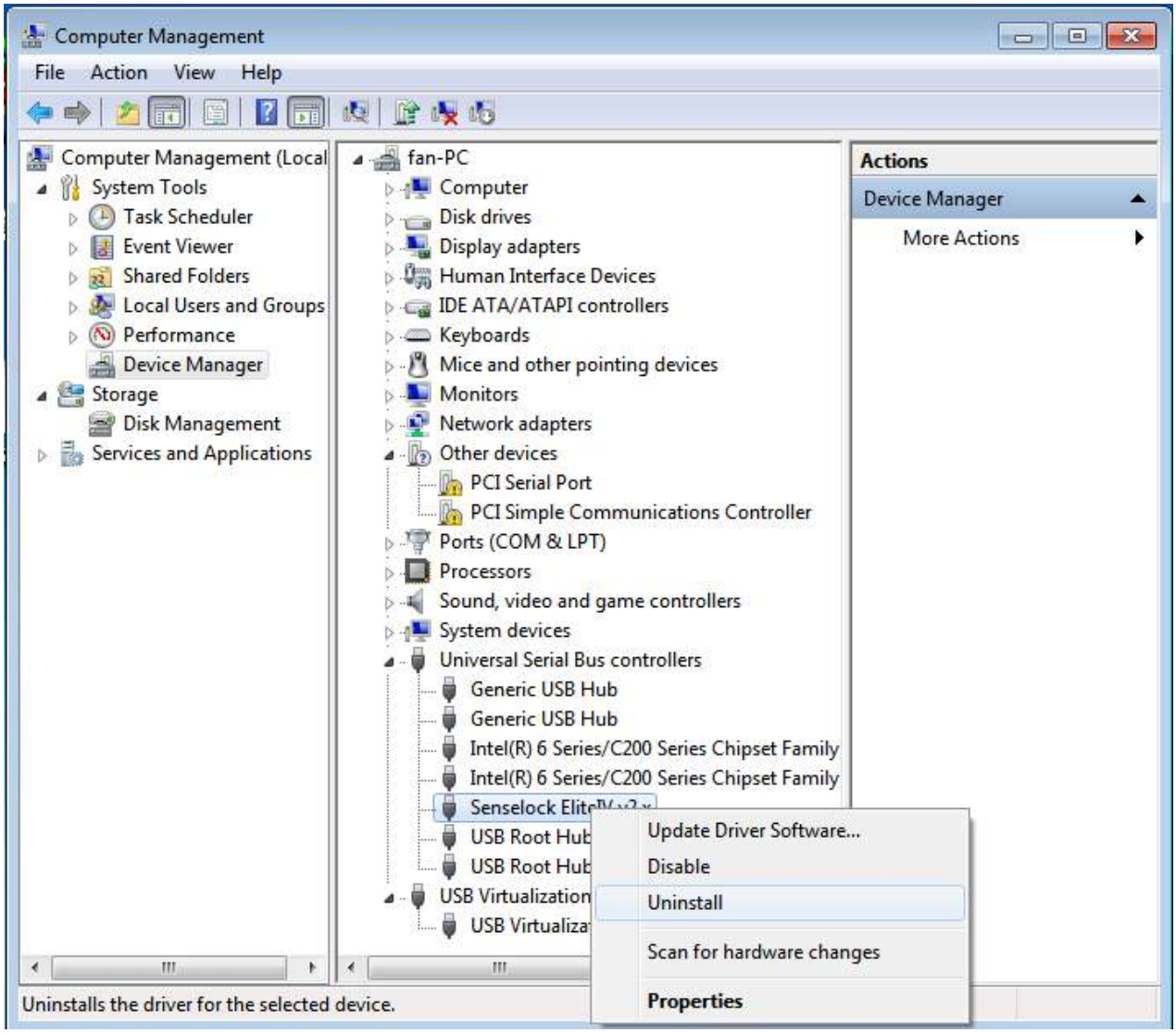


Method 3: Open the “Start” menu → All Programs → DIAView → Uninstall:



2. Uninstalling Senselock driver (the USB license key must be inserted):

Open “Computer Management”→ search for “Senselock EliteIV v2.x” in “Device Manager” and the right-click on it:

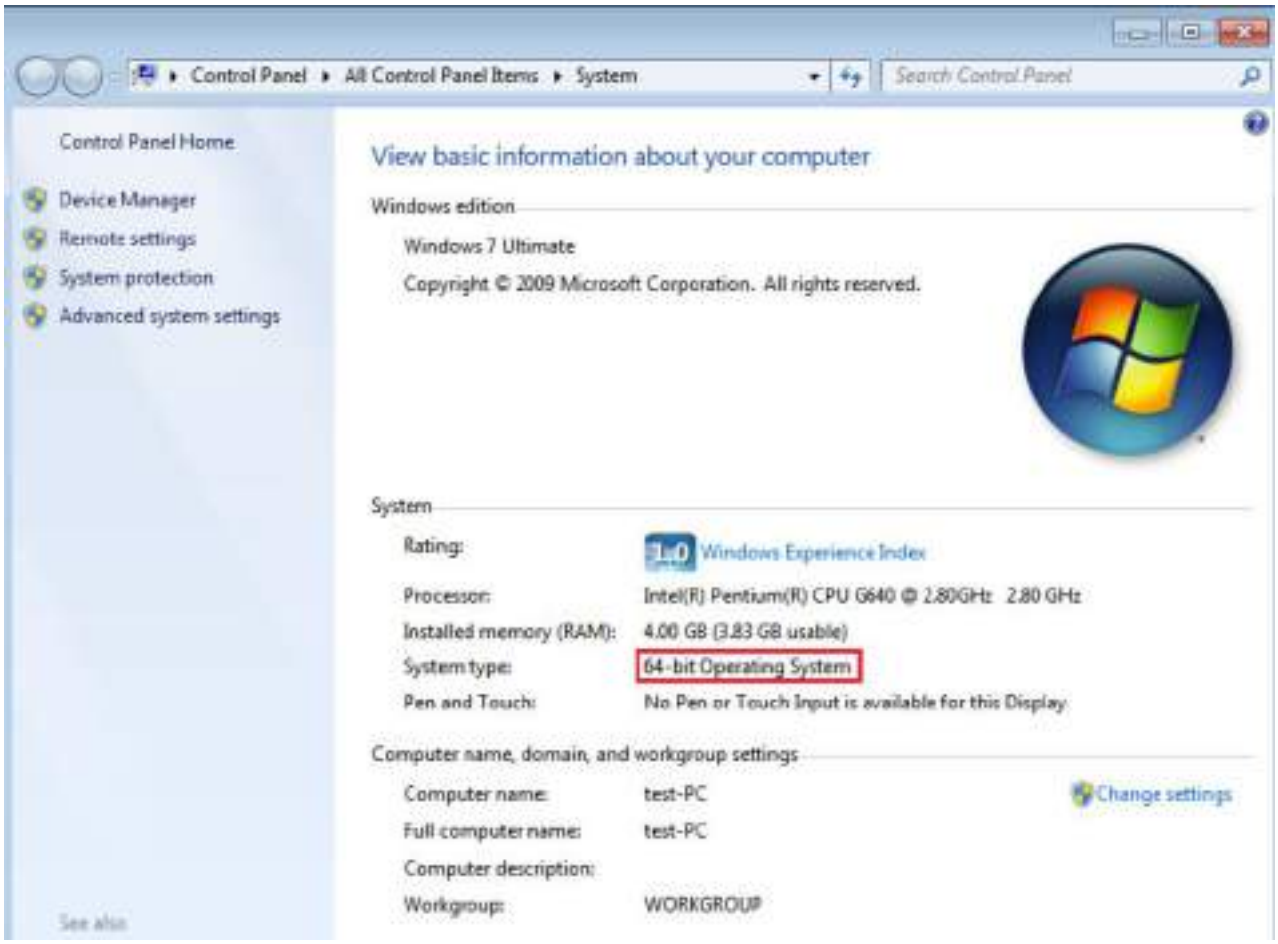


Select “-Uninstall-” and the “-Confirm device uninstall-” window will appear; check “-Delete driver software for this device-” and then click “-OK-” to uninstall it:

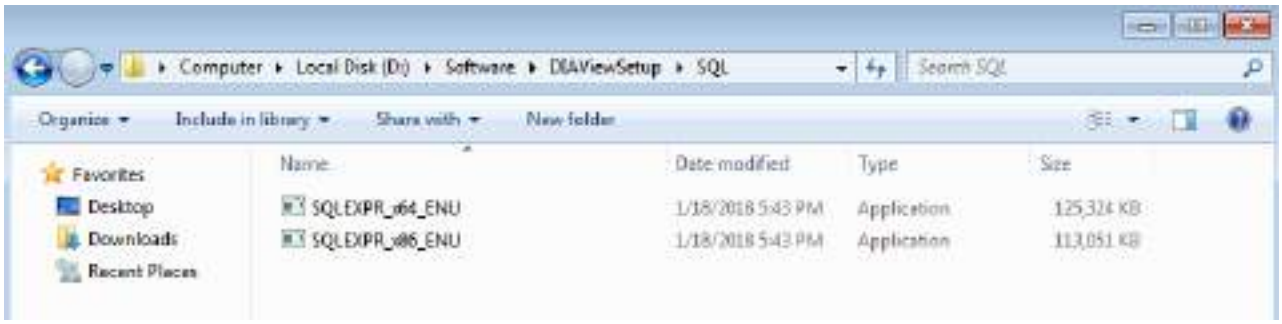
2.7 Install SQL server database

DIAView adds new "History Variable" function and optimize the query efficiency, use new version of "History Variable" need install SQL Server database and database version can't under the SQL Server 2008 R2. DIAView package has contain SQL Server 2008 R2 package(language version: en,x86 and x64), if user want to install other language version or higher version, please go to microsoft website(https://www.microsoft.com) to download. Installation steps are as follows (Taking Windows7 Professional Edition as the example):

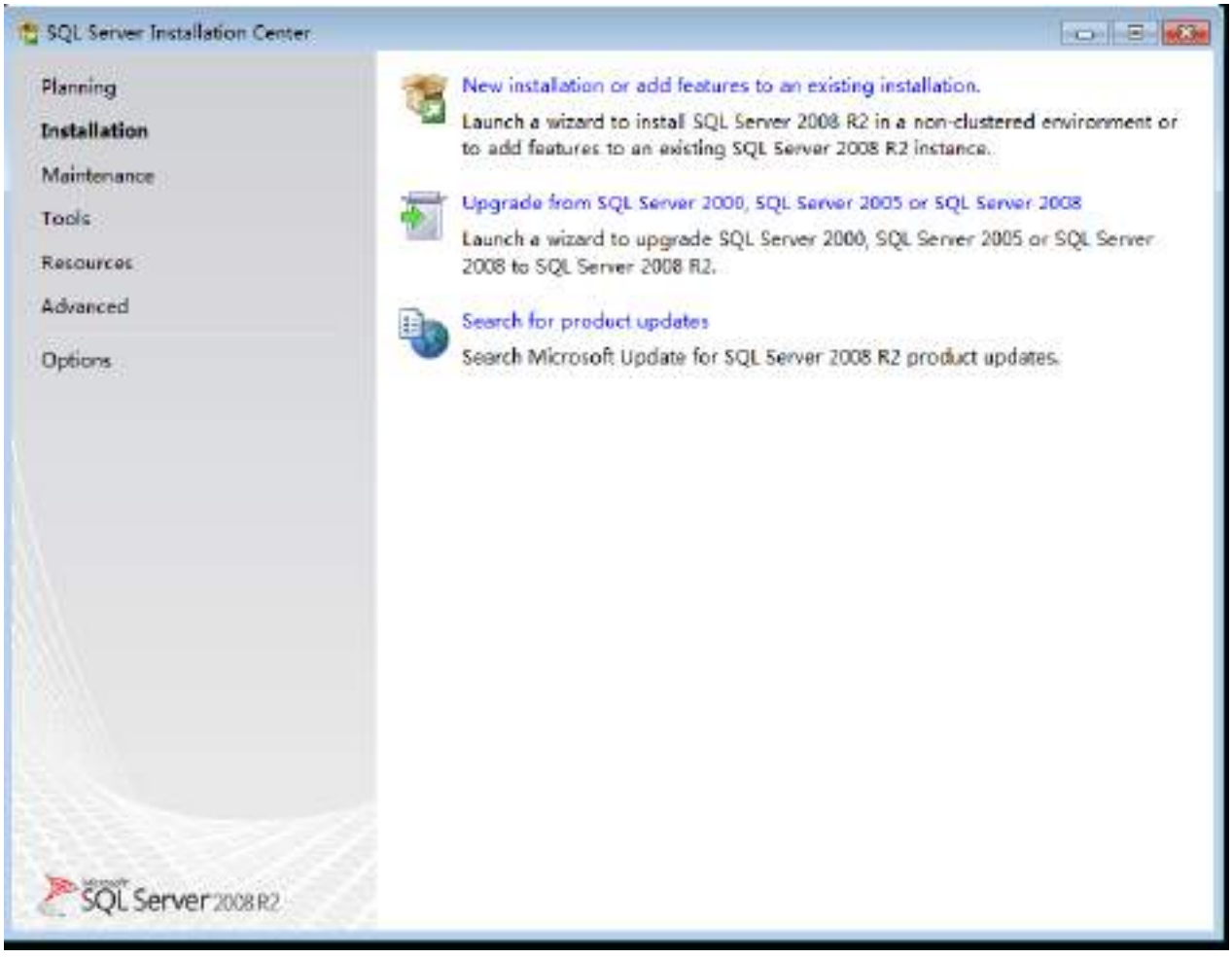
Step 1: Right_click "Computer", click "properies", go to computer system view, check the "system type", as shown in the figure below:



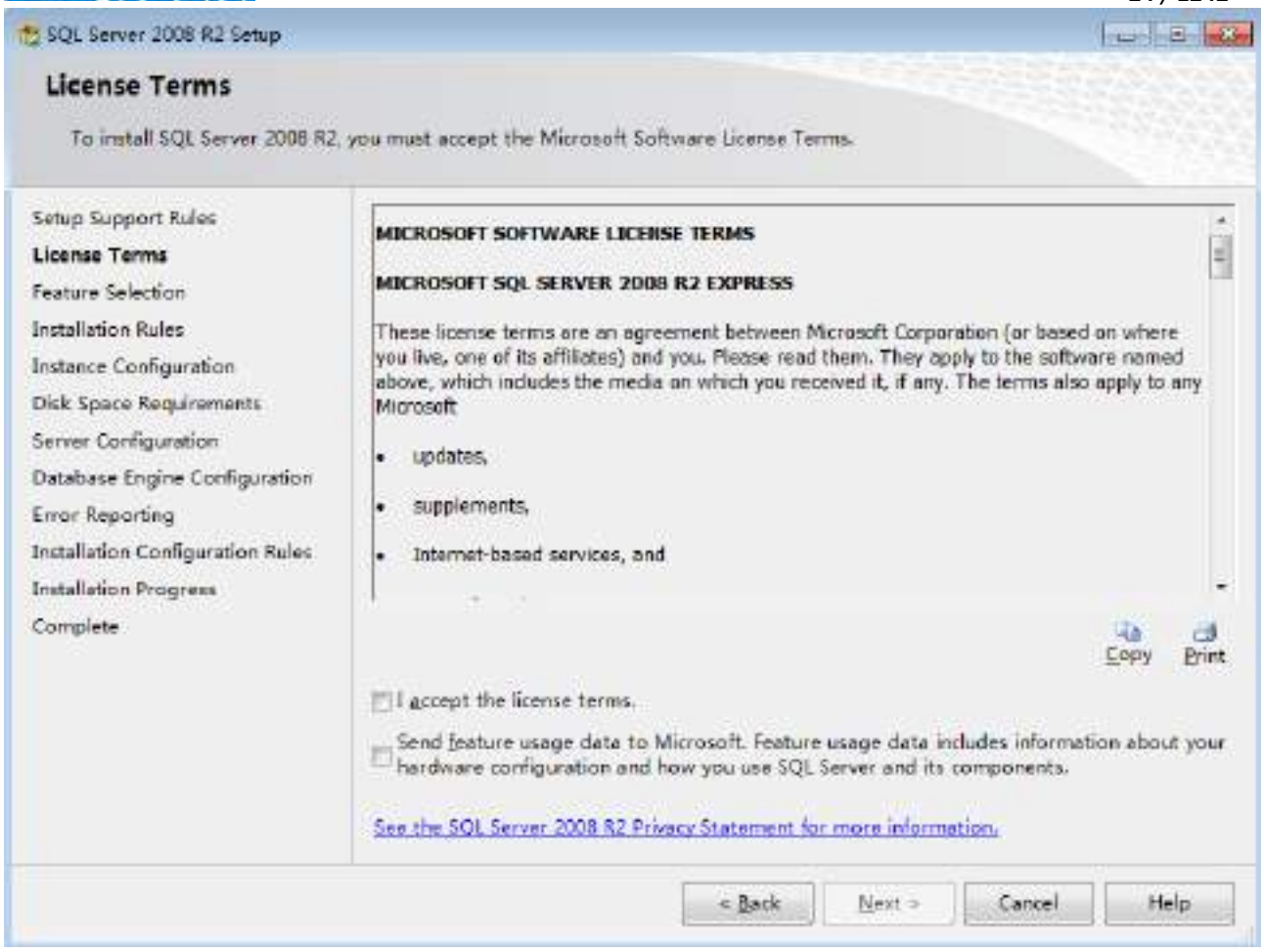
Step 2: Install SQL Server package in "sql" folder at DIAView package, if your computer System type is 32-bit Operating System, install package in "x86" folder, if your computer System type is 64-bit Operating System, install package in "x64" folder, as shown in the figure below:



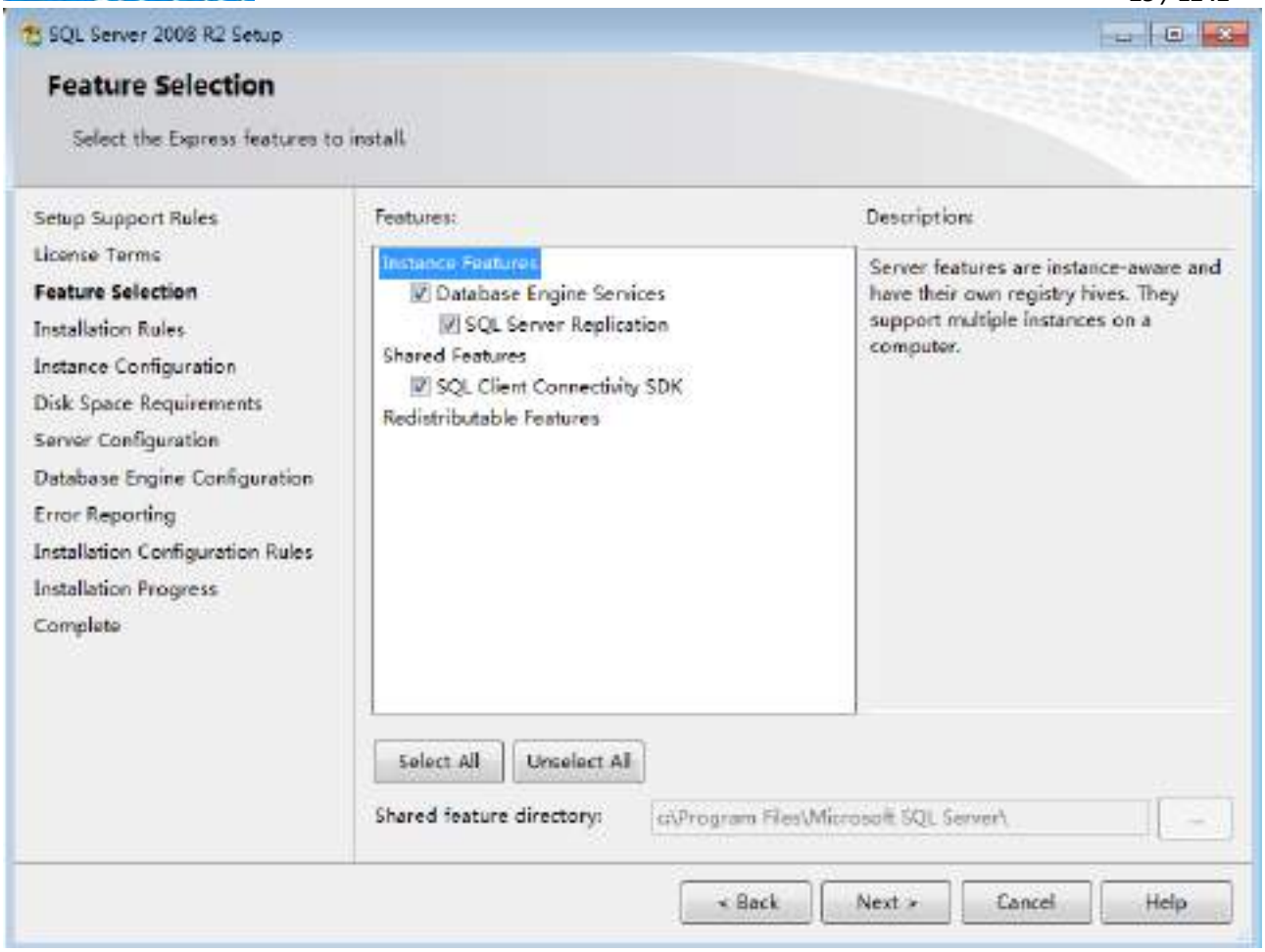
Step 3: Click SQL Server package, go to installation view, as shown in the figure below:



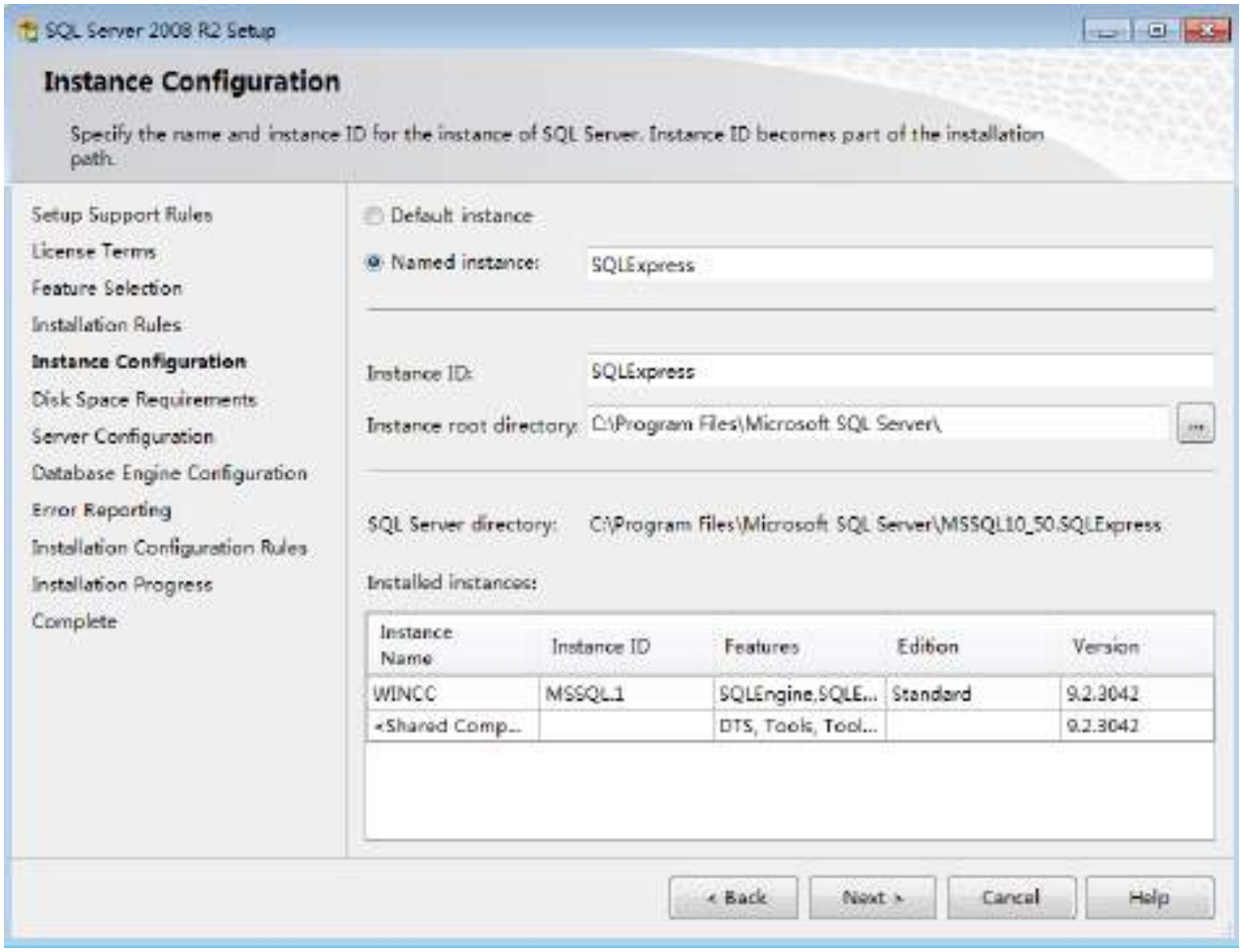
Step 4: Select "Installation" node and click "New installation or add features to an existing installation", go to "Lisence Terms" view, as shown in the figure below:



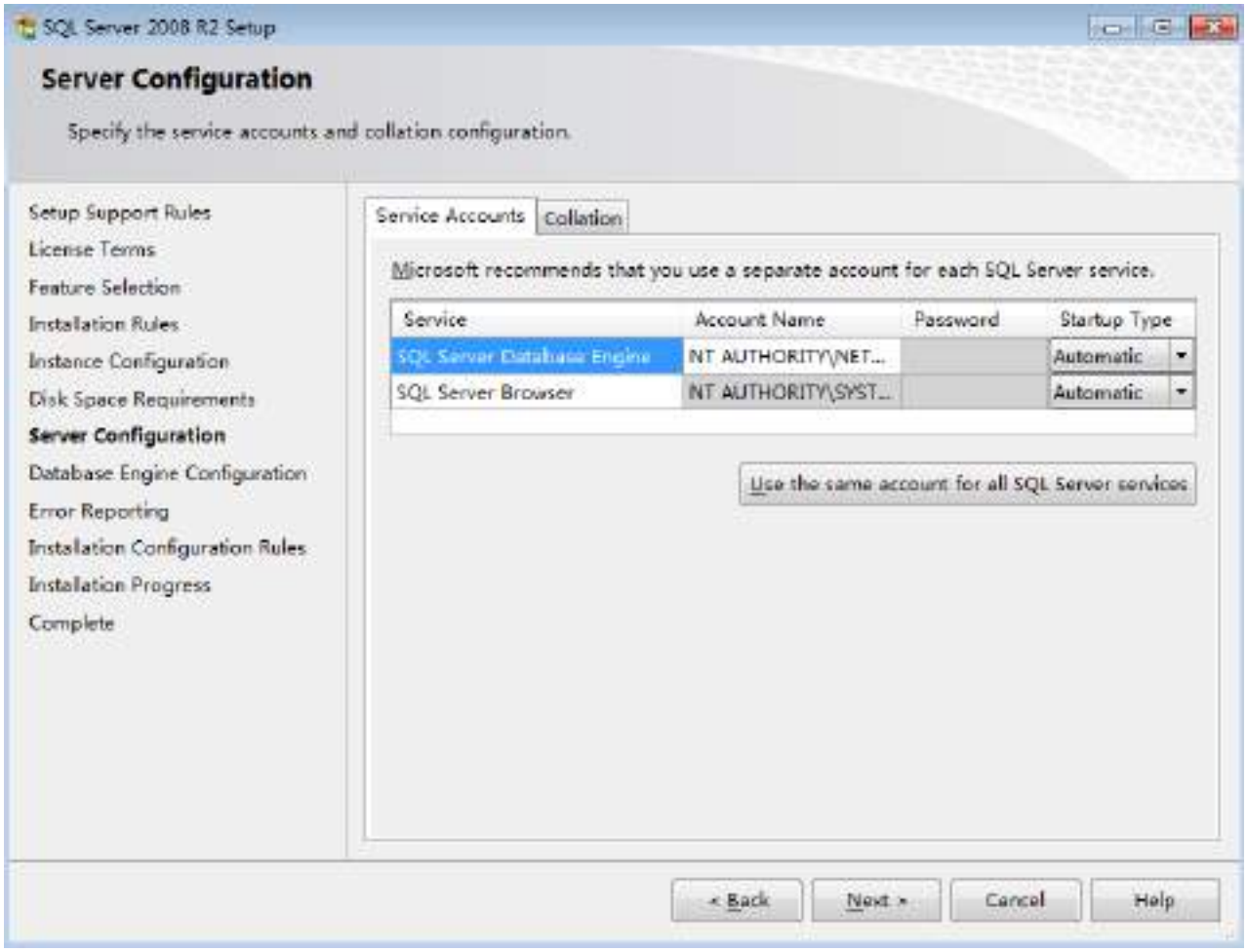
Step 5: Select "I accept the license terms" after read license and click "Next", go to "Feature Selection" view, as shown in the figure below:



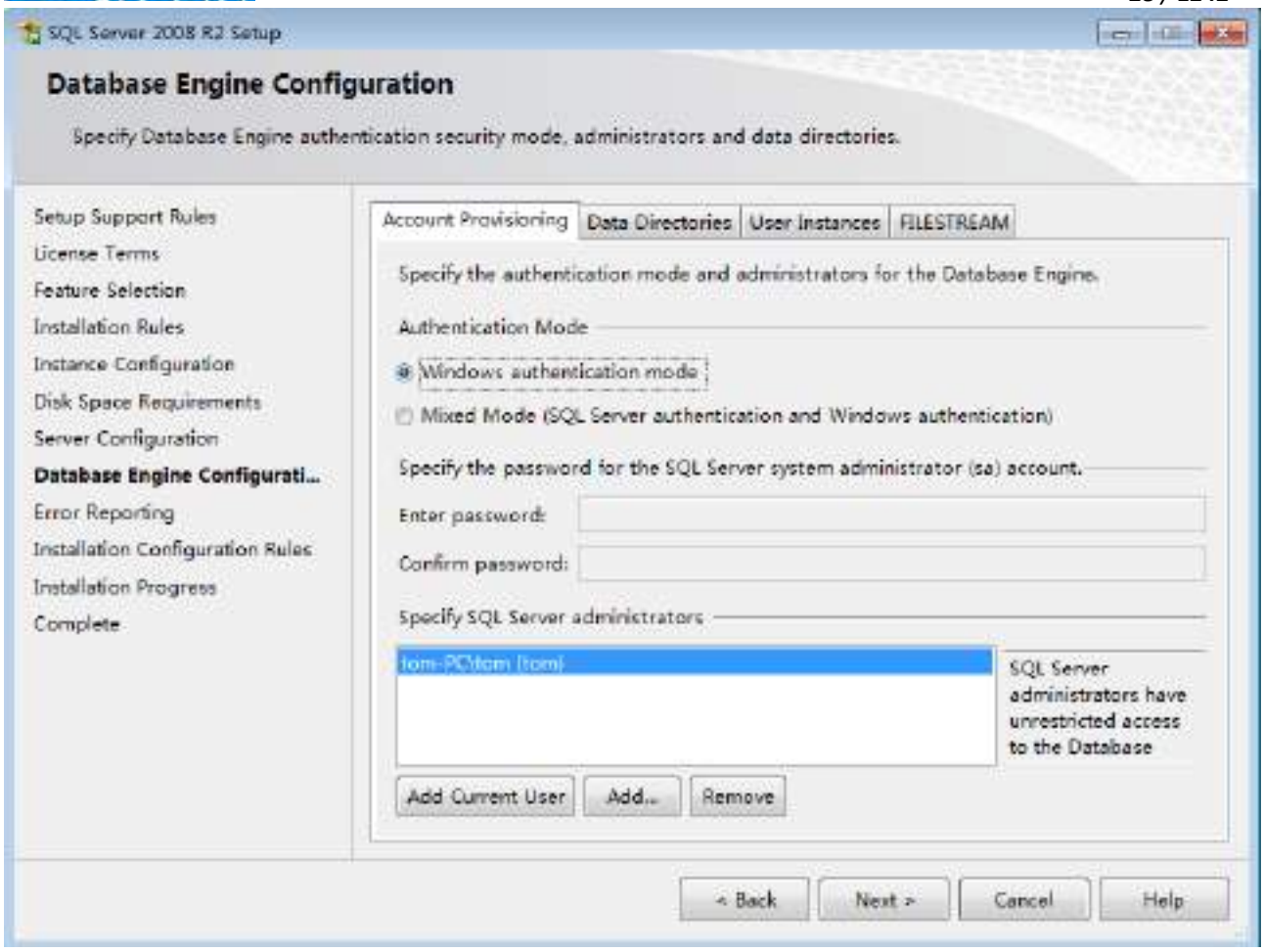
Step 6: In this view, suggert install all, click "Next", go to "Instance Configuration" view, as shown in the figure below:



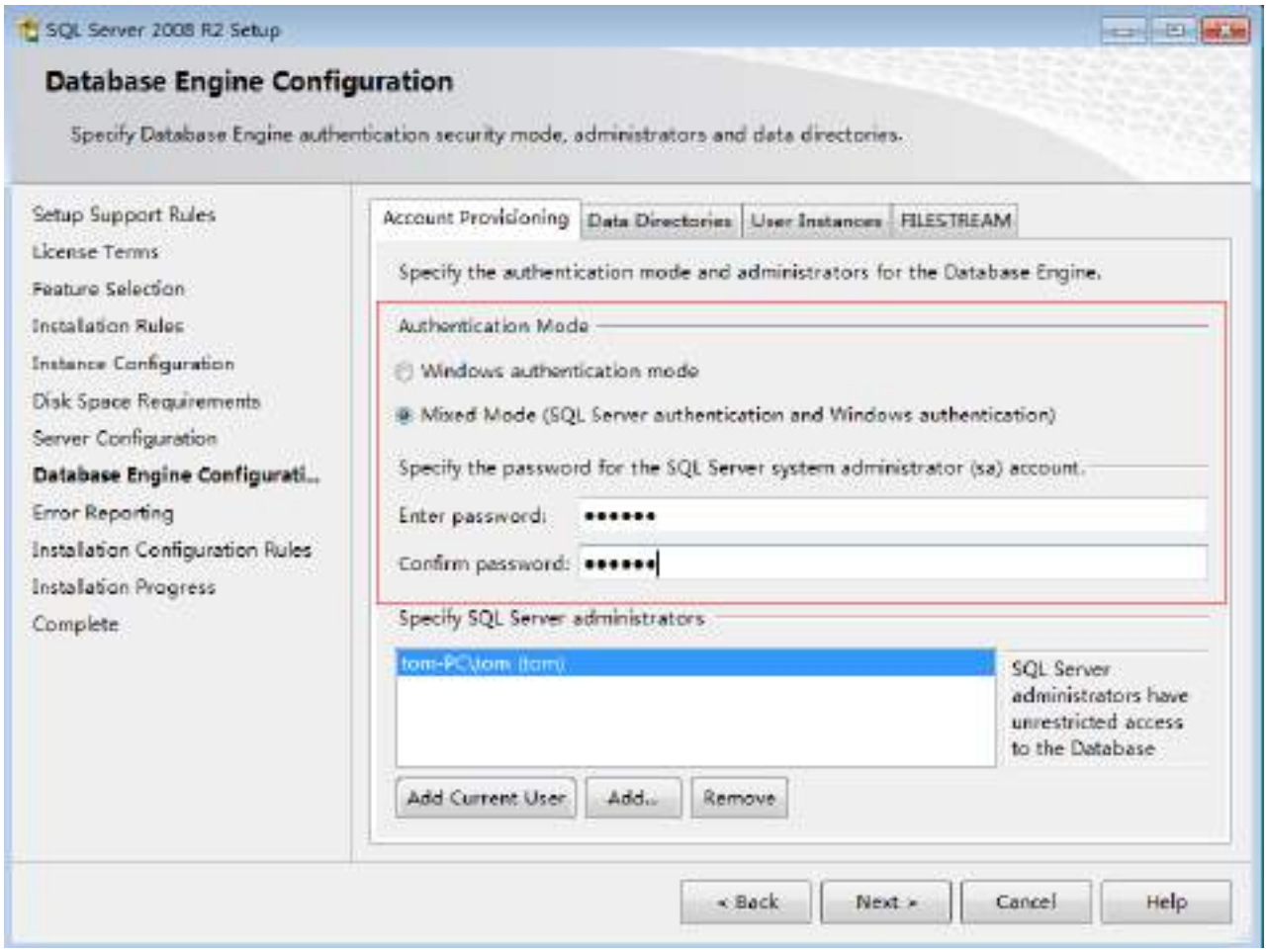
Step 7: Click "Next", go to "Server Configuration" view after check the hard dist space, as shown in the figure below:



Step 8: Click "Next", go to "Database Engine Configuration" view, as shown in the figure below:

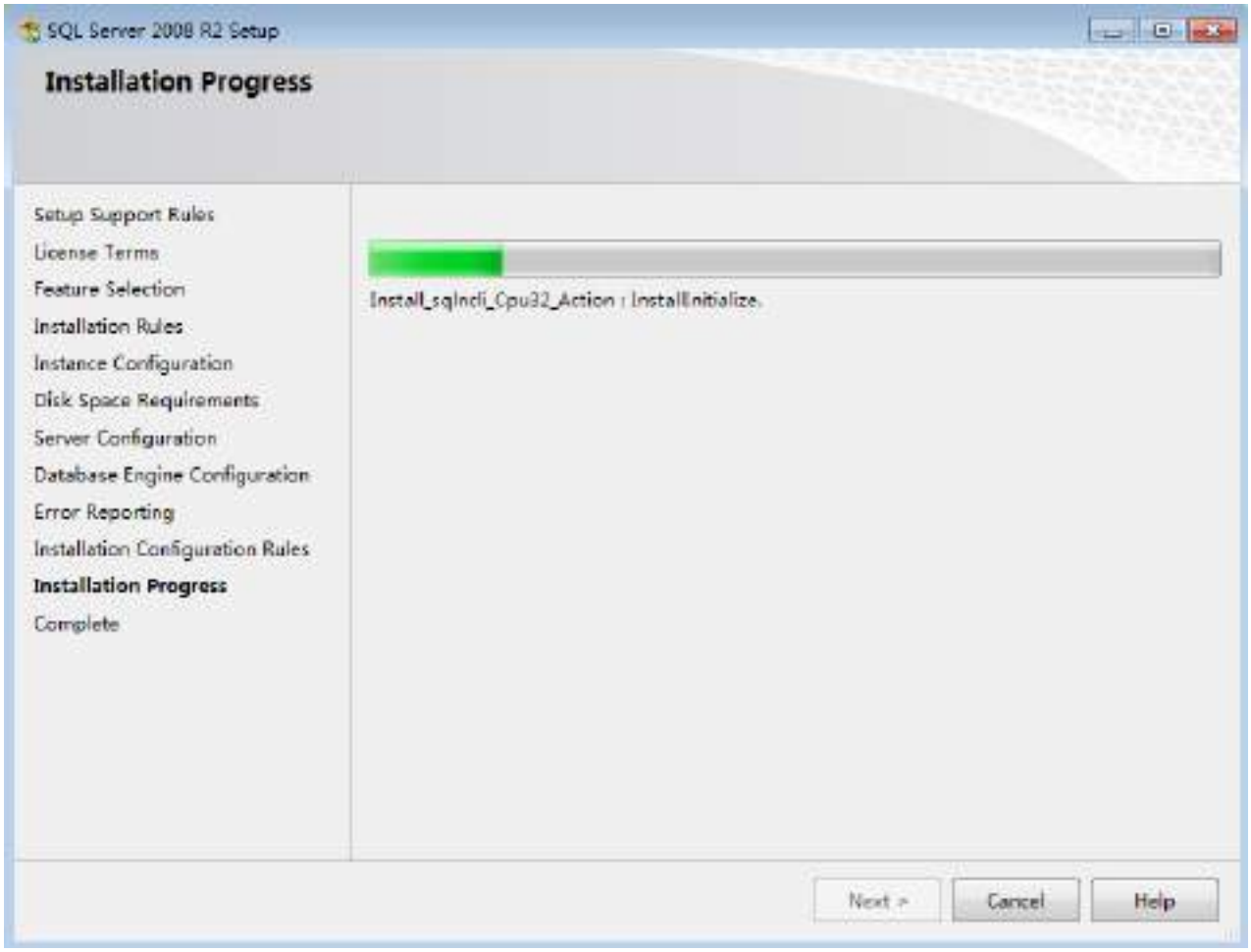


Step 9: In "Database Engine Configuration" view, suggest select "Mixed Mode" at "Authentication Mode", create a SQL Server account(usrename: sa), other computer can connect your SQL Server service use account, as shown in the figure below:

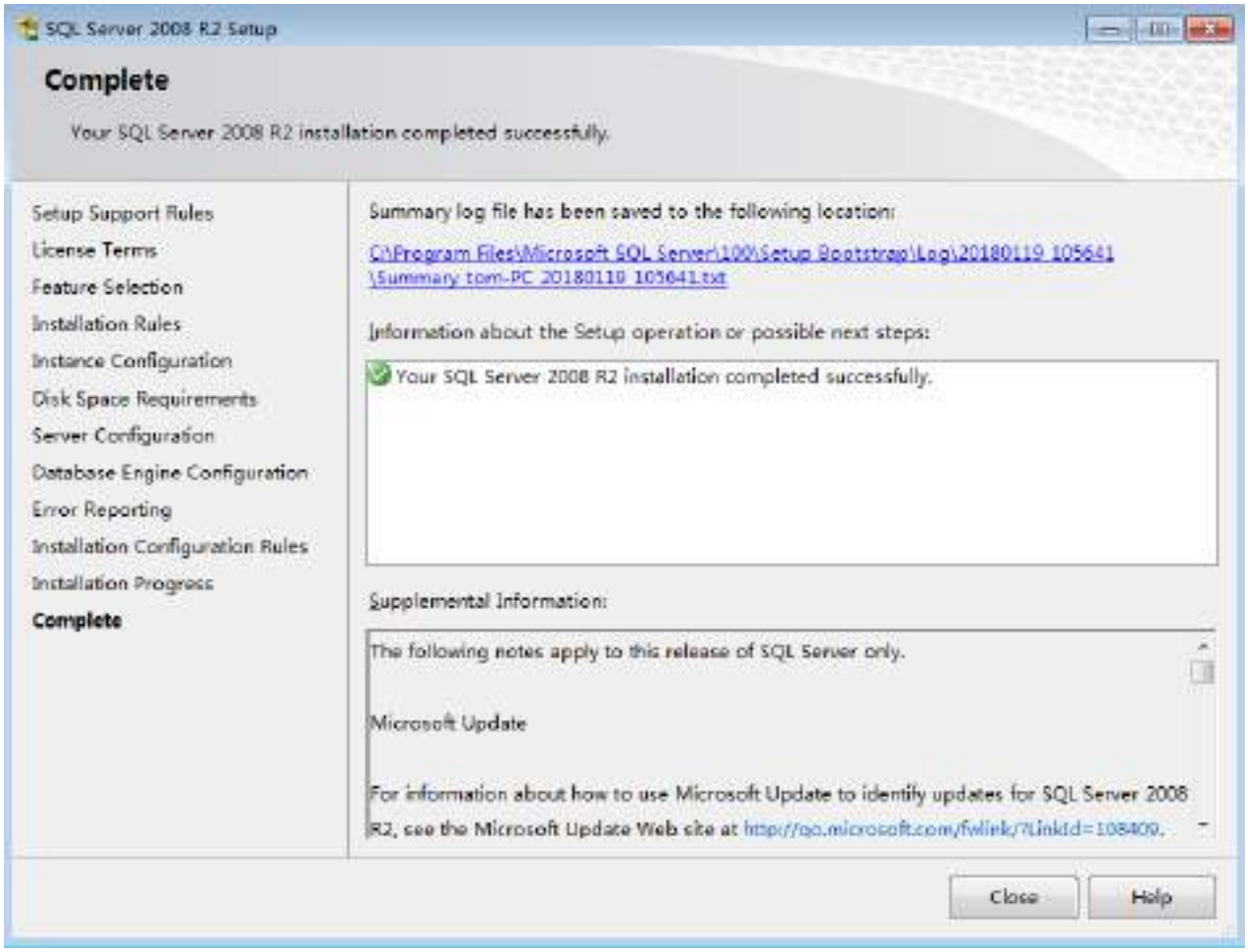


Step 10: Click "Next" and go to "Error Reporting" view, as shown in the figure below:

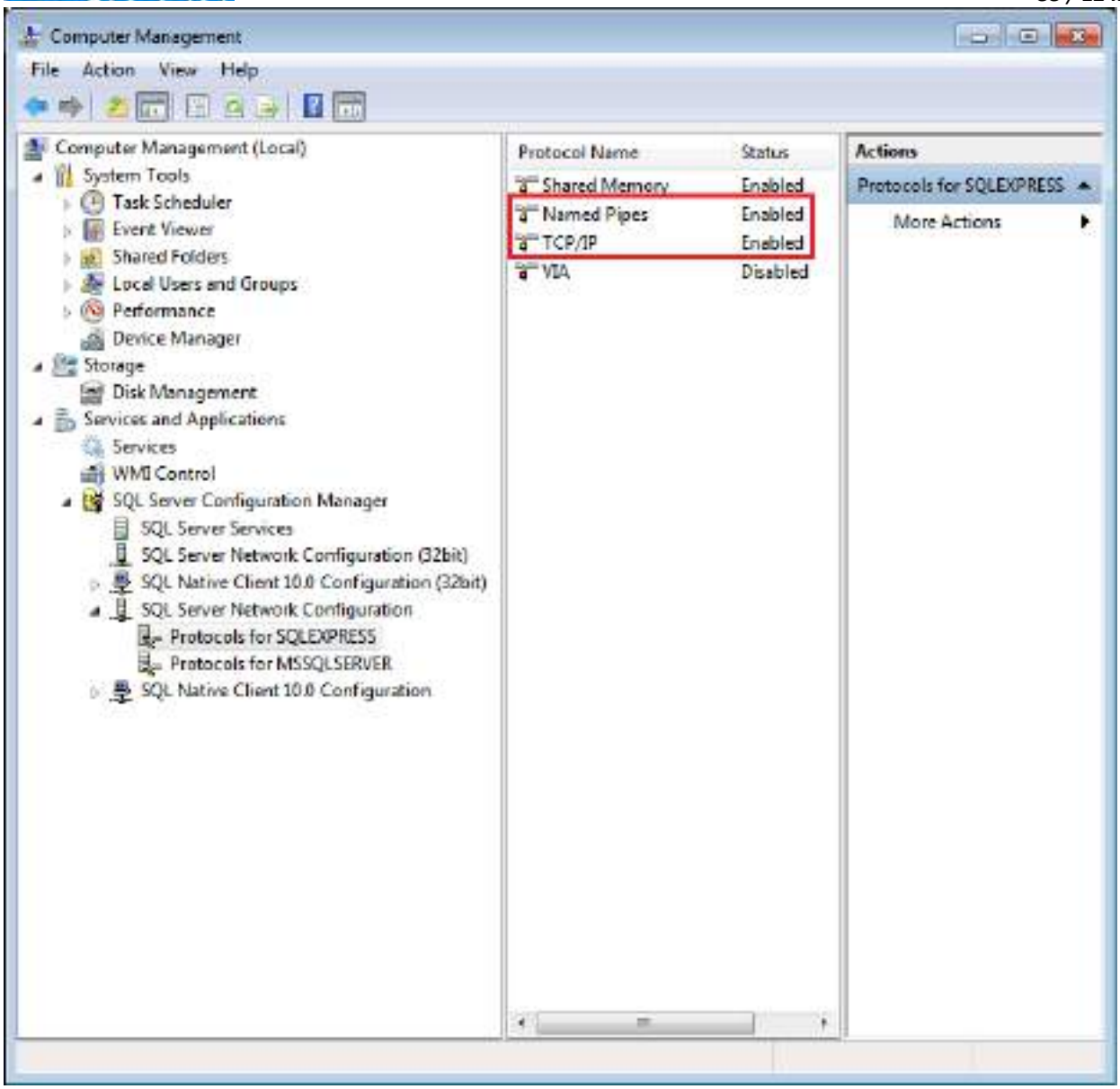
Step 11: Click "Next", go to "Installation Progress" view, start installing, as shown in the figure below:



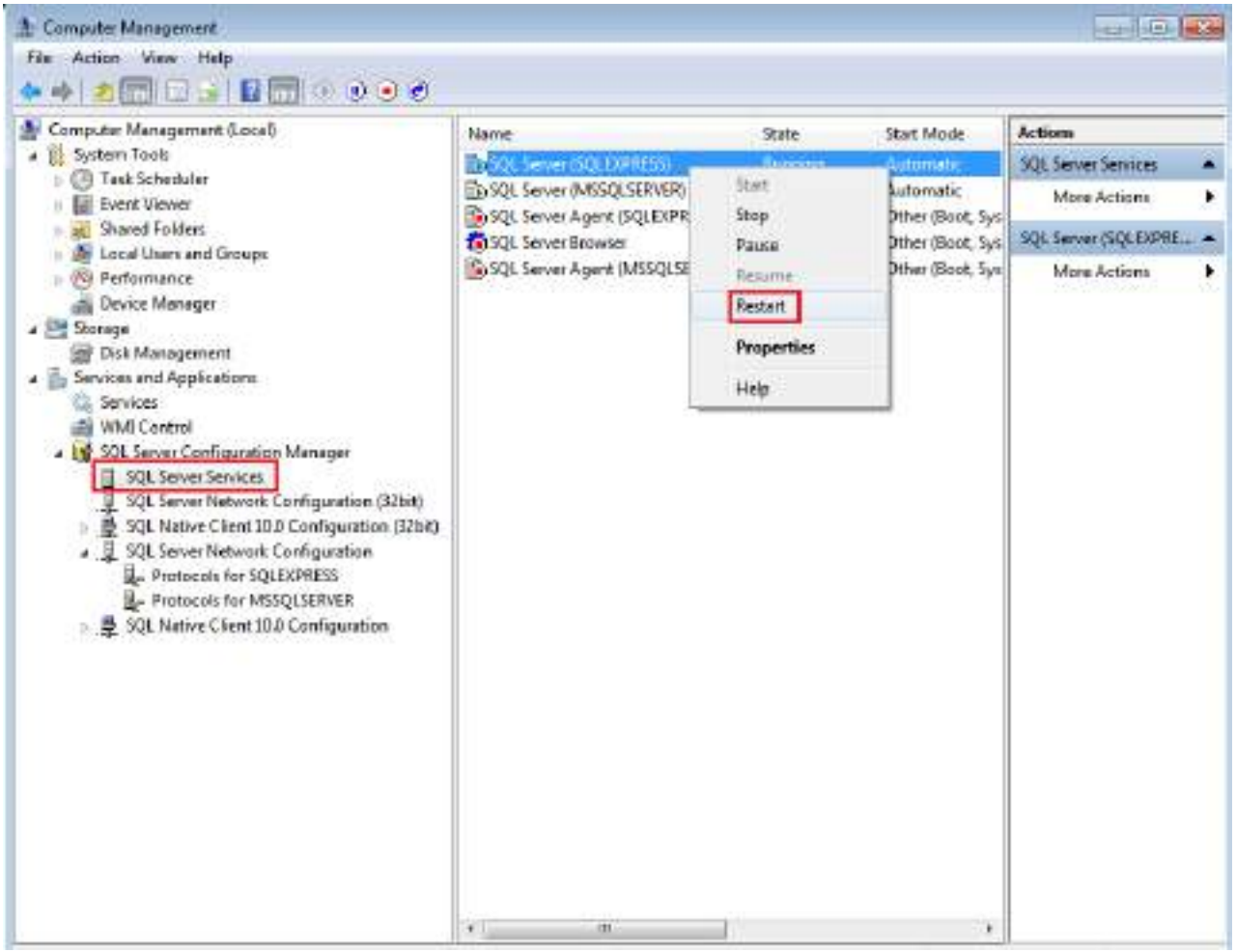
Step 12: If install success, as shown in the figure below:



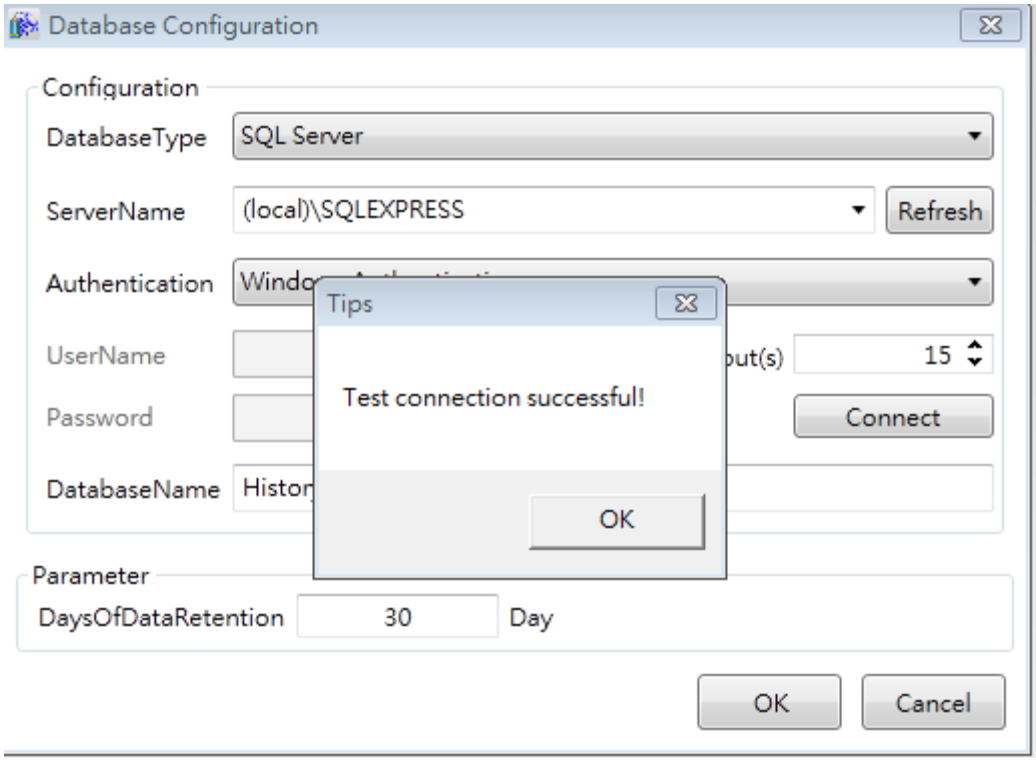
Step 13: Click "Close" after install success. If user want other computer connect this SQL Server service, right_click computer, click "Manage", go to Computer Management view, click "Services and Applications" , click "SQL Server Configuration Manager", click "SQL Server Network Configuration", select SQL Server service we installed(default: SQLEXPRESS), check the status of "Named Pipes" and "TCP/IP" is enabled, if status is disabled, right_click and click "Enabled",as shown in the figure below:



Step 14: Click "SQL Server Services", right_click service, select "Restart",as shown in the figure below:



Step 15: Open Dvstudio after install DIAView, right_click "History Variable", select "Database Configuration", click "Connect Test", check SQL Server databse has installed correctly, as shown in the figure below:



3. DIAView Development Environment

3.1 Common terms and concepts

This chapter introduces common technical terms used in the DIAView software to help users properly learn how to use the DIAView software.

- **Project**

A project is a set of automated application system developed with the DIAView software according to user requirement; it mainly includes a collection of the development of automated monitoring and control systems and its configuration information. A project includes graphic window interface, IO communication, alarm, variable and other parts, and the project developer can perform centralized management for each part of the project.Project also can be called "Engineering".

- **Development environment and runtime environment**

The DIAView software is composed of the two parts: Development environment and Runtime environment.

The development environment is the operating platform for system users to perform project design and development, including designing graphical interface, setting IO communication parameters

and configuration functions etc.

The user creates projects through the development environment, and then creates IO communication and channels, and defines variable dictionaries in the project. Users also use a sketchpad to draw the simulation pictures of field sceneries,configure animations, events, alarm information, configure reports and curves etc.; they can even write back-end scrips according to the data and calculation formula to perform logic operations.

The runtime environment is the platform where project are dynamically executed; it provides an execution display and control interface. The system user uses the runtime environment to perform real-time monitoring.

- **IO communication, channel,and device**

IO communication refers to collecting data information of on-site equipment, and use communication medias to read/write the information from/to equipment or save it to the system database to further analyze and process.

In which the communication media is what is called a “channel”; it is the bridge for the DIAView software and other equipment to connect and achieve data information transmission.

Equipment: This is the hardware products that the DIAView software uses to perform data acquisition; common equipment includes: PLC programmable logic controllers, inverters, boards, digital instrumentations, smart controllers and monitoring probes etc.

IO communication connections currently supported by the DIAView software includes serial port and Ethernet; supported communication interface standards include: Modbus TCPIP, Modbus RTU, Modbus ASCII, OPC, Delta DVP TCPIP, Delta DVP RTU, Delta DVP ASCII, Delta AH TCPIP, Delta AH RTU, Delta AH ASCII and simulators etc.

- **Variable dictionary and variable**

The variable dictionary is a collection of variables in the project; it manages the following variables;

Variables are values of the project that can change at any time; they are important participants for system data information handshakes.

The variable dictionary of the DIAView software can perform group management for variables; multiple variable groups can be created and multiple variables can be created under a variable group.

- **Window**

The window is a core component of the DIAView software for users to draw field simulation pictures, configure parameters and display information. It is the basis to realize real-time visualization of the DIAView software -- it provides various drawing elements including basic graphic group, window controls group and advanced controls group etc., system's built-in graphics and the customized graphics library that users can add on their own; it also provides windows that allow configuring of image object properties, animations and events. In the DIAView software, one windows is one screen.

- **Property**

In the DIAView software, properties describes the features or characteristic information that all objects in a window; for example name, size and background color etc

- **Animation**

Animation refers to the process of setting the properties of an graphic object, including the size, color, rotation angle and position, to change according to the changes in the variable data while the project is executing.

- **Event**

An event refers to the process of specifying the properties of an graphic object, including the size, color, rotation angle and position, to change with mouse or keyboard operations.

- **Alarm and alarm variable**

Alarms refer to related notifications or warnings (such as: E-mail or sound) triggered to remind related personnel to pay attention when

a certain data or status of the system is over or under the default value during the project execution process. The alarm is composed of alarm

variables and alarm configurations.

Alarm variables: Generates alarm information variables, sets the related variables that the alarm variables must connect to, and sets

the alarm level and related alarm configuration information. The DIAView software performs group management for alarm variables; each

alarm variable belongs to a certain alarm group.

Alarm configuration: Configures and alarm method--E-mail or sound alarm.

- **Operation variable**

Refers to saving the operation record to the database when the values of certain variables in the variable dictionary changed due to user operations in order to query and check user operations.

- **History record**

Refers to saving the changing record of certain variable values in the variable dictionary and defines the recording method for data statistics and analysis.

- **Security zone**

Refers to dividing and setting the graphic objects, control units and equipment etc. in the project to specific security levels so that only users in that security level can access them. It is a method to control user rights in order to guarantee the reliability and security of the DIAView software execution.

- **User**

System operators and administrators with related operating authorities set for the project development and execution. Different authorities can be set for different users in order to achieve specialization so that the DIAView software and execute securely.

- **Recipe and element**

Recipes refer to the ratio of raw materials and automated management of process control provided for specific production processes.The

use of recipes can increase the efficiency of automated production.

The recipe of the DIAView software is composed of two parts: recipe item and recipe ingredient.

The recipe ingredients are the main components of the recipe item. The recipe ingredient is controlled by the engineering variables of the

engineer. The adjustment of recipe ratio is completed by changing the variable values.

- **User script**

The user program of the DIAView software is mainly divided into “condition program” , “time program” and "Global Function".

Condition program refers to executing related program functions when the conditions set matches the engineering variables.

Time program refers to executing related program functions when the system time reaches the time condition set for the program.

Global function refers to user can package the reusable script code, and use it in other script.

- **DatabaseAccess and DatabaseAccessItem**

DatabaseAccess provides users with the ability to access external databases and read and write external database data.

The DatabaseAccessItem represents connection information for connecting to an external database.

- **Global**

Global refers to user can change content of text and image in window by switch language.

- **Project configuration**

Project configuration includes the configuring of the database, configuring of the execution screen and the configuration of the starting screen. Users can achieve handshakes between real-time data of the project and other commercial databases through engineering configuration by setting related configurations of the execution environment of the DIAView software

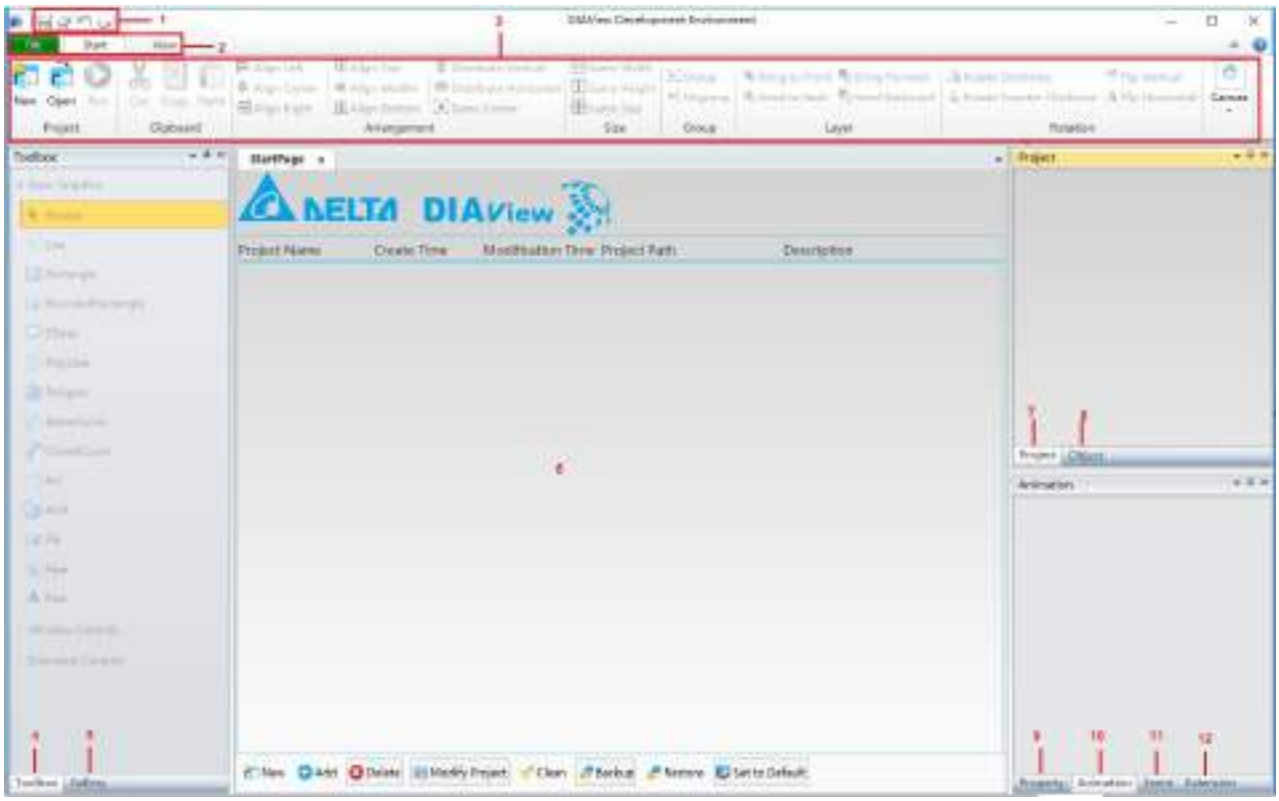
- **Runtime**

Runtime refers to dynamically running the project completed in the development environment in the runtime environment according to the development design requirements and configured parameters, achieving real-time data acquisition of the system, dynamic display of the screen, real-time response of commands, automatic display of reports and manual dispatching and control in order to exert the functions and effects of the various parts of the project.

3.2 DIAView development environment introduction

3.2.1 DIAView development environment framework

Start the DIAView software development environment and enter the main window of the system, as shown in the figure below:



Composition of the DIAView software development environment:

1. Quick tool bar: Provides frequently used buttons for project development.

2. Menu bar: Provides various basic operation functions for project development.

3. Tool buttons: Shortcut buttons for operating commands.

4. Toolbox: Provides frequently used basic graphic elements and control units for drawing.

5. Gallery: Frequently used graphic elements built-in the system and containers for users to freely expand customized graphics.

6. Sketchpad area: The area for project management, graphic image drawing and editing.

7. Project window: Tree index display window of the project; displays various compositions of the project and provides various portals for operation and configuration functions.

8. Object browser window: Displays all object members in the window.

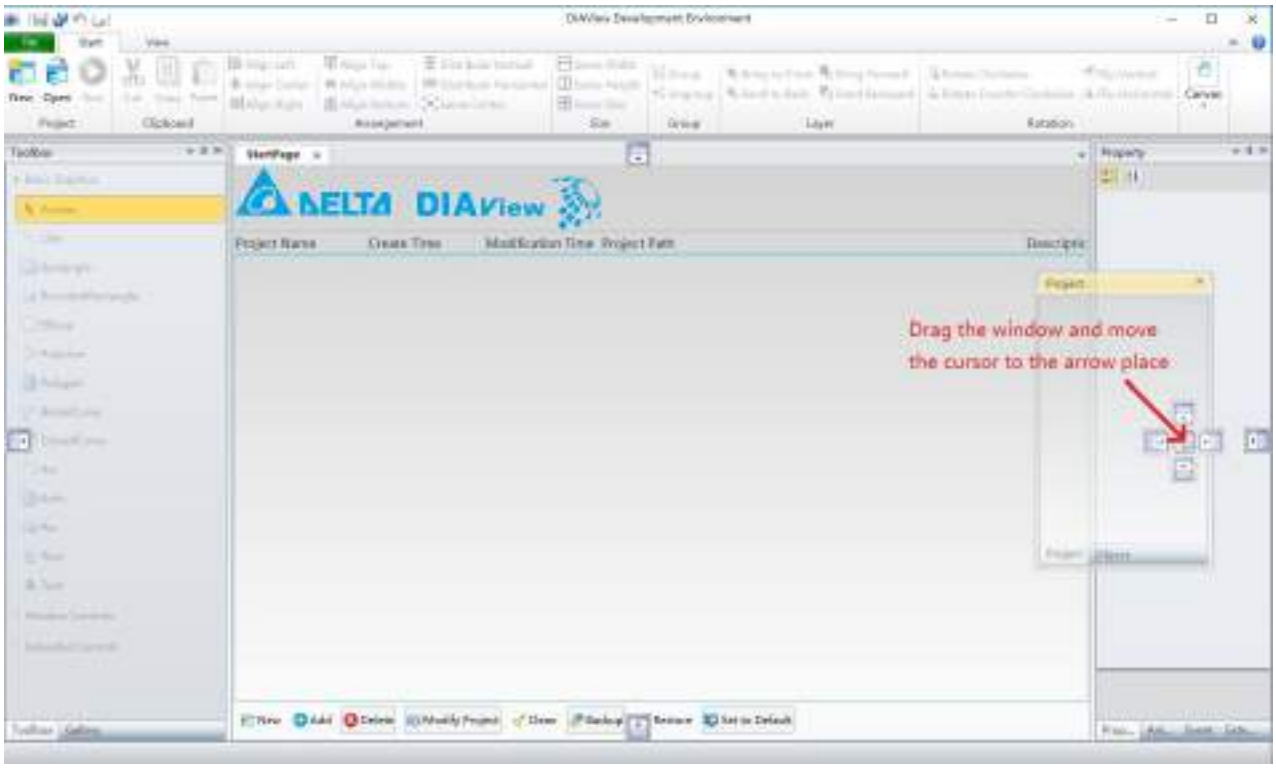
9. Property window: Displays the properties of the object; provides an operating portal to modify properties.

10. Animation window: Portal to configure animations for objects.

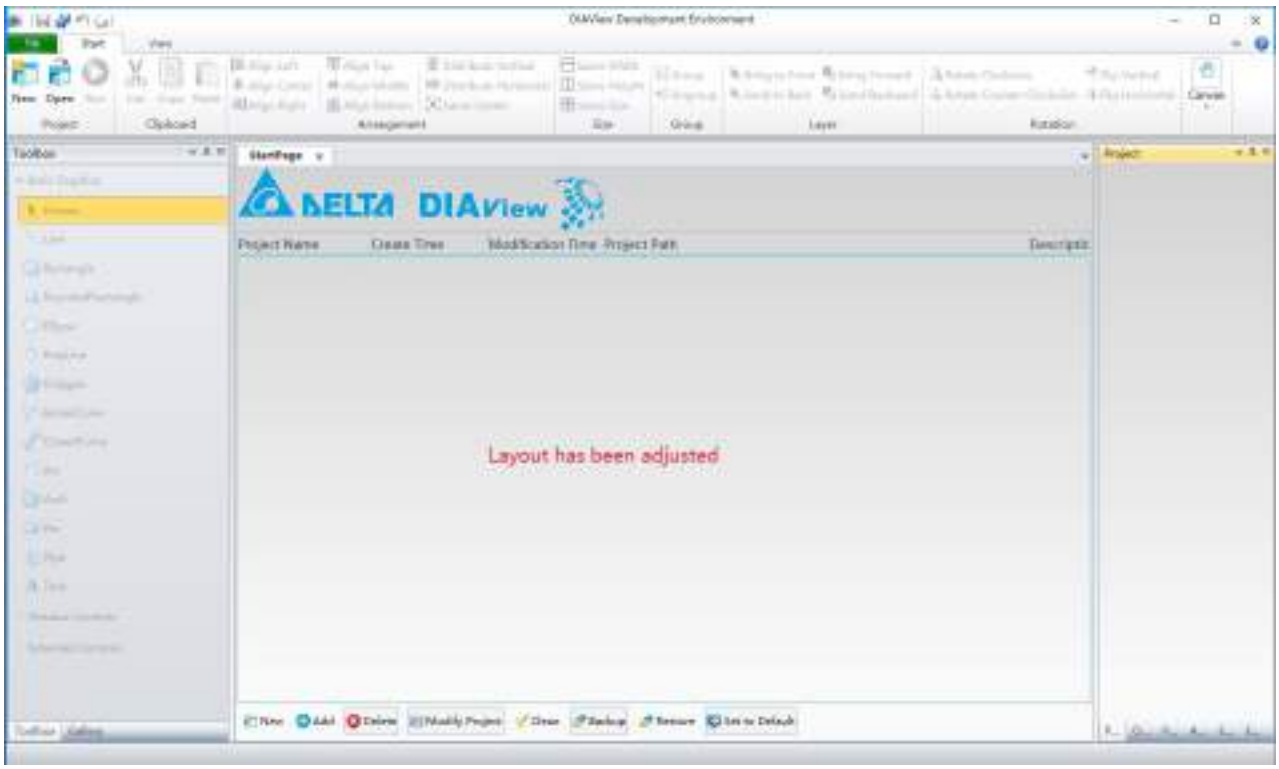
11. Event window: Portal to configure events for objects.

12. Extension window: Portal to configure extended properties for objects.

The windows of the DIAView software development environment can also be rearranged by dragging them with the mouse, as shown in the figure below:



The rearranged windows are as shown in the figure below:

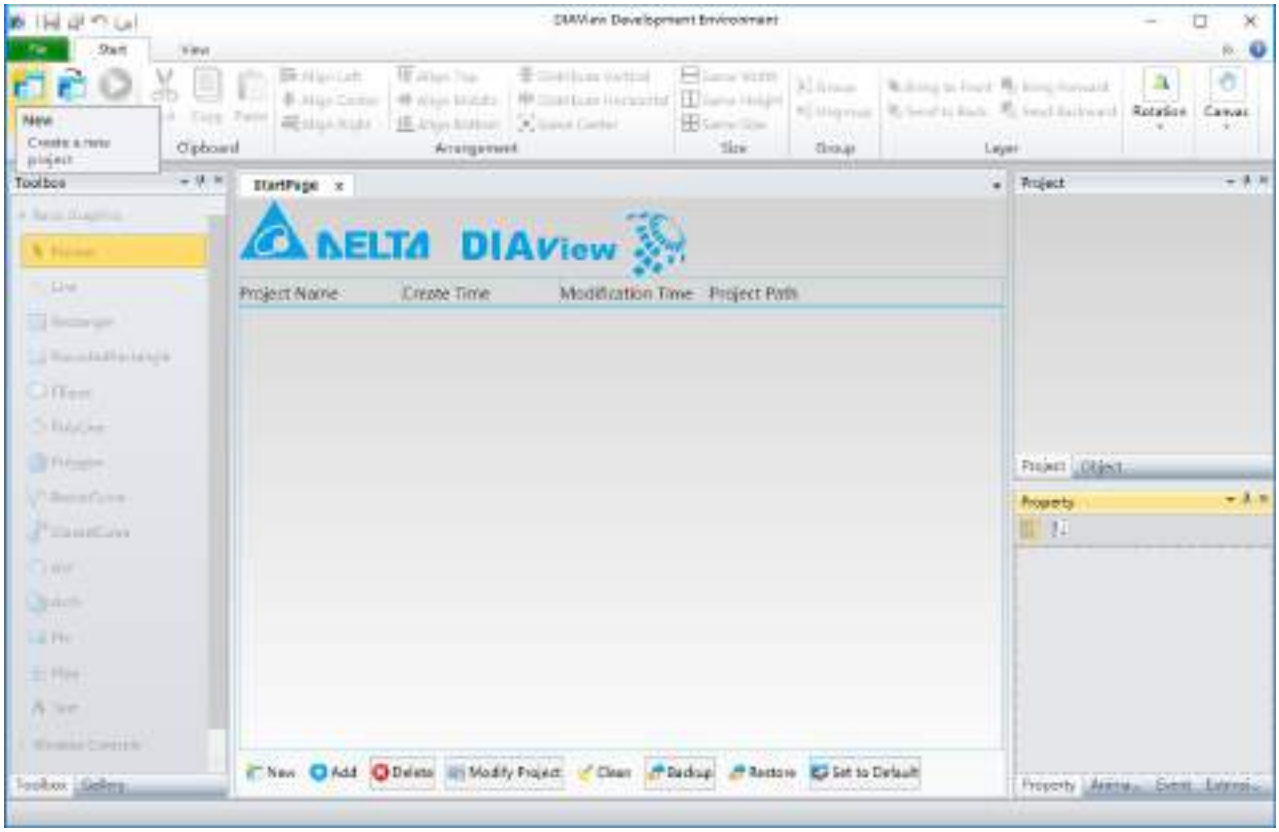


3.2.2 New project

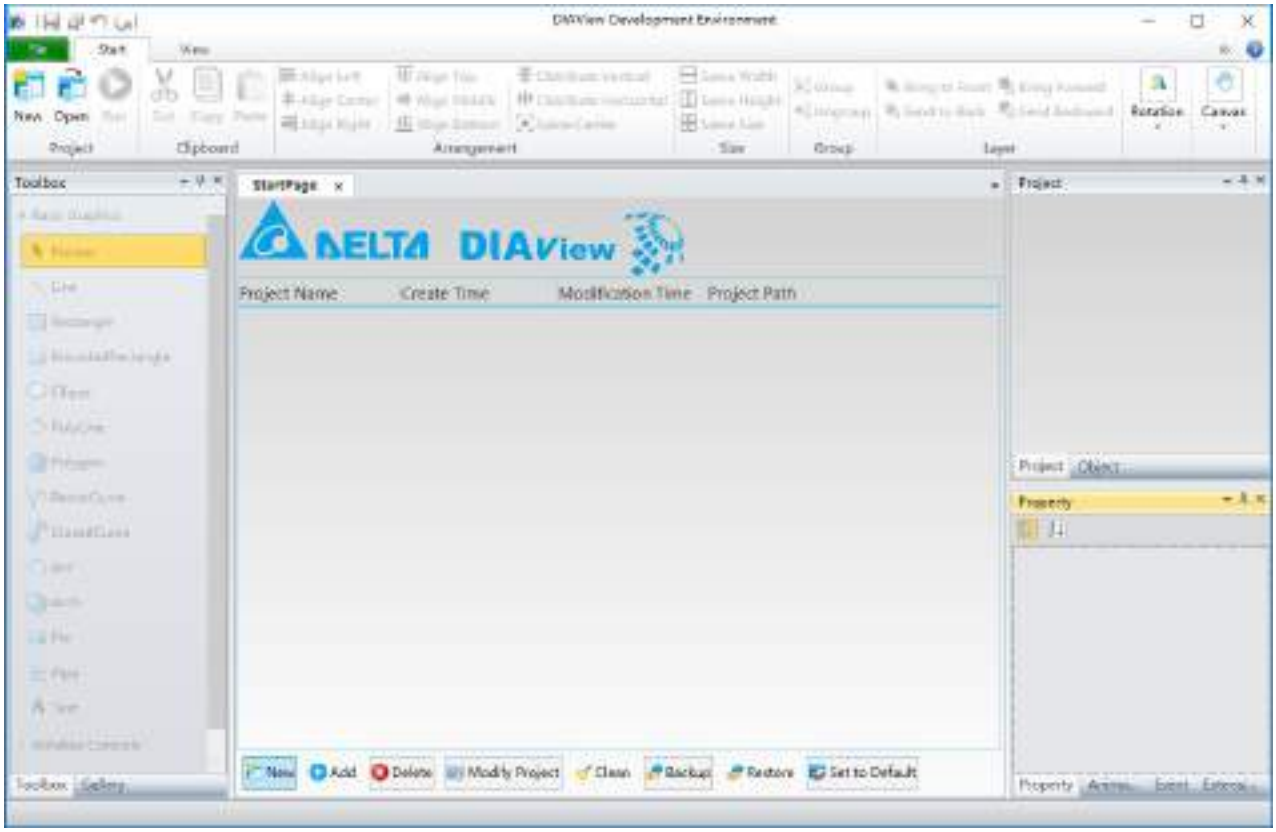
There are four main ways to add new projects in the DIAView software development environment.

Method 1: Select the “File” menu of the DIAView software development environment and then press “New”.

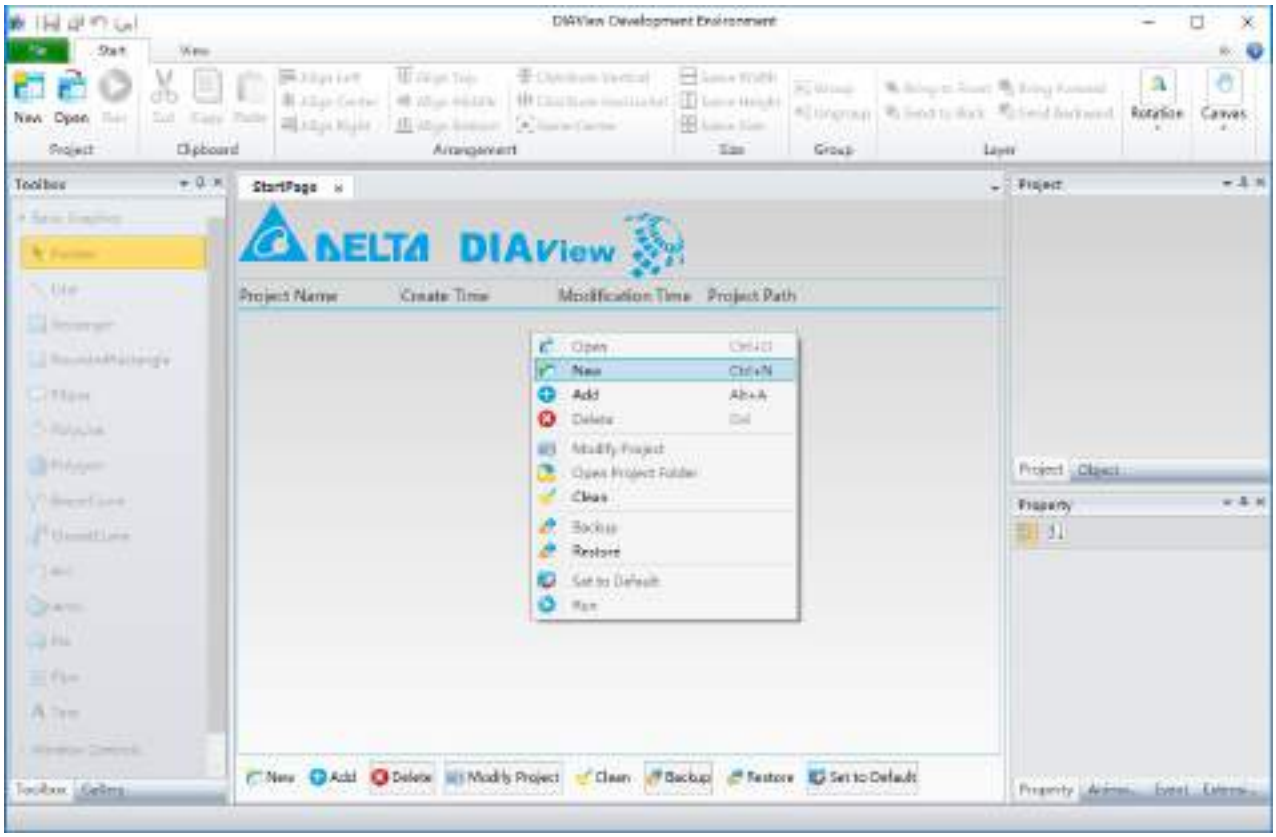
Method 2: Select the “Start” menu of the DIAView software development environment and then press “New”.



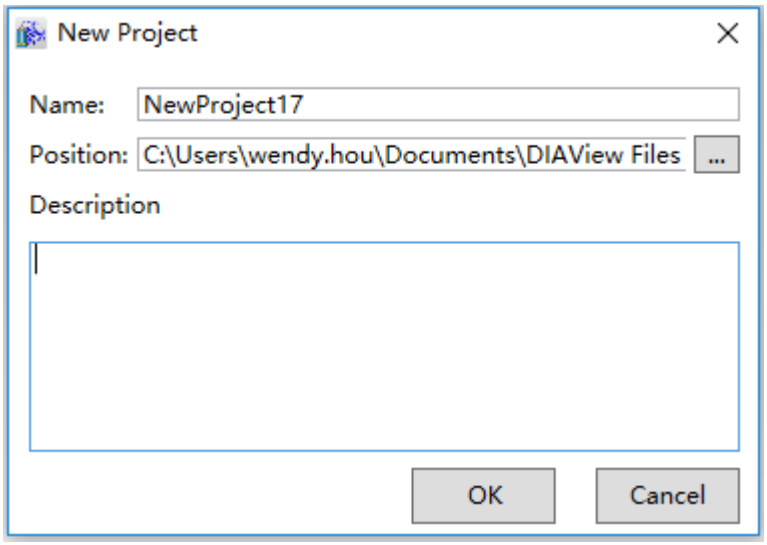
Method 3: Press the “New” button in the toolbar on the starting page of the DIAView software development environment to add a new project.



Method 4: Right-click the mouse on the starting page of the DIAView software development environment and then select “New” from the right-click menu to add a new project.



Clicking the add project button adds a new project and related information on project configuration, as shown in the figure below:



Name: Name defined for the project (required); the system will choose a name by default but the user can also change it. The project name customized by the user must comply with the naming rules of the VB Script scripting language:

1. The name must start with a letter.

2. Spaces, periods, exclamation marks and special characters such as @$#&*? etc. cannot be used in the name.

3. The length of the name cannot exceed 255 characters.

4. The name cannot be repeated with the object and function names in the VB Script language.

5. The name of the project cannot be repeated as other created projects placed under the same project folder.

Position: Specified the storage path of the project; press the  button to select the path.

Description: Allows users to define project-related explanation or description information.

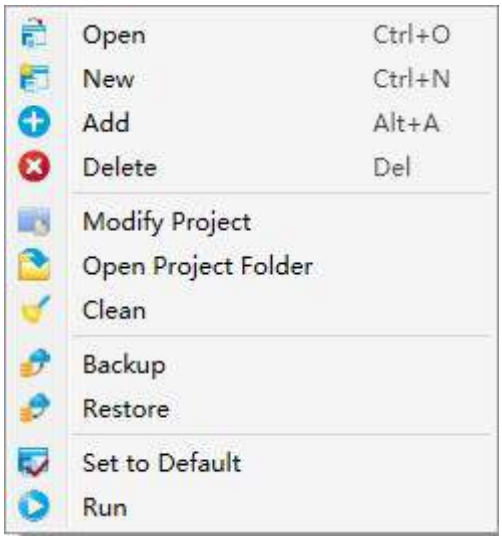
3.2.3 Manage project

To manage project refers to performing centralized management of the projects in the DIAView software to make it easier for users to perform related operations to the projects. There are two methods to manage the projects in the DIAView software: First is through the starting page of the development environment and second is through operating the function menu (please refer to “3.3.1 File menu”).

Start page

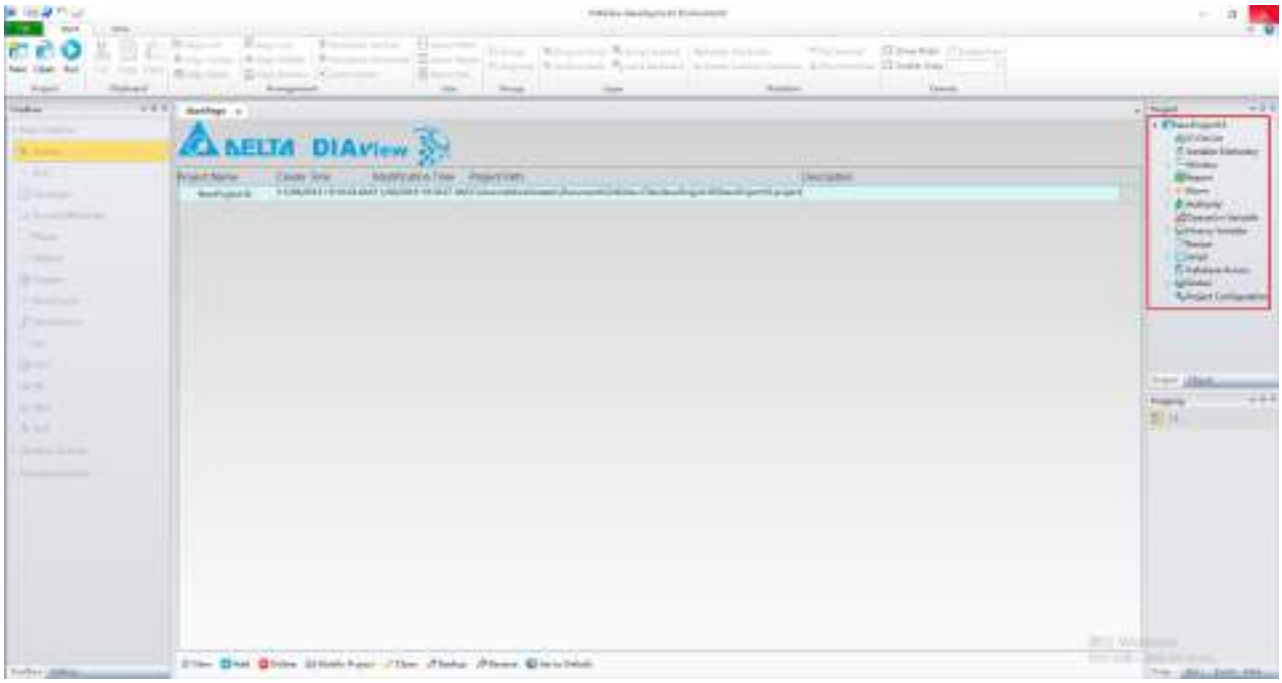
Open the DIAView software development environment and the starting page will be opened on the main interface by default. Project item information will be opened if a project was previously being developed, including project name, time of creation, modification time, resolution, project path and project description information. Operations such as add new, add, delete, clear, backup, restore and set project as preset project etc. can be performed on the starting page, as shown in the figure below:

Right-click menu can also be performed to do related operations on the starting page, as shown in the figure below:



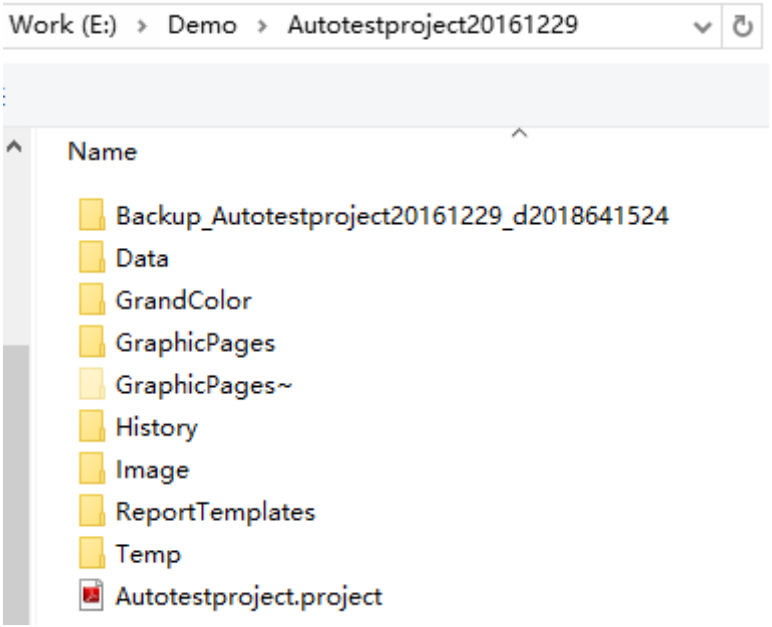
Introductions to the buttons and right-click menu item functions on the start page:

1. Open: Opens the project selected from the project list on the starting page; this can also be achieved by double-clicking the row of project to be opened in the project list. Once the project is opened, the project tree index will be displayed in the project window to the right as shown in the figure below:

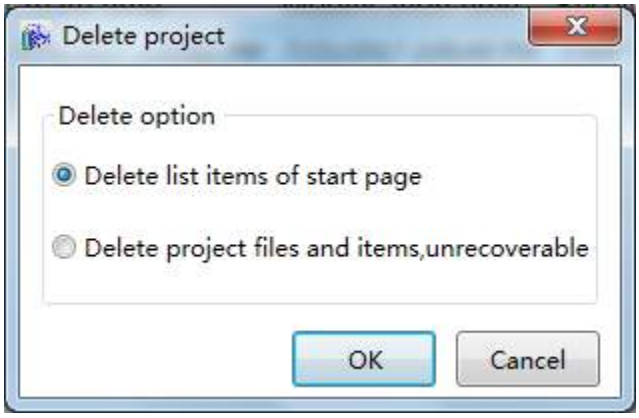


2. New: Adds a new project; the created project will be added to the project list on the starting page, and be opened in the project window. Please refer to “3.2.2 New project”.

3. Add: Adds a project that already exists under the specified path into the starting page project list. Click the “Add” button or right-click menu to open the folder selection box:



4. Delete: Deletes the selected project on the starting page project list. Click the “Delete” button or right-click menu to open the project deletion selection box:

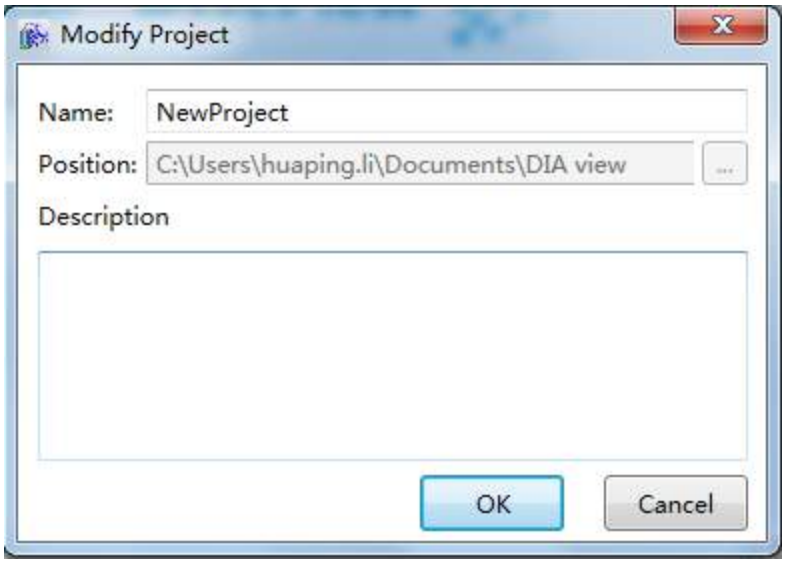


There are two delete options:

Delete list items of start page: Deletes the project information from the starting page, but the project file is not deleted.

Delete project file and item, unrecoverable: Deletes the project from the starting page list and also deletes the project file completely.

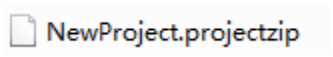
5.Modify project: Modify project name and description, click "Modify Project" button or right-click menu to open the modifying project box:



6.Open project folder: Opens the folder where the project is located

7.Clean: Clears the project information on the starting page list. It will be cleared if the project file was already deleted but still exists in the project list.

8.Backup: Backup: Backs up the selected project. The backed up project file will have the extension ".projectzip", as shown in the figure below:



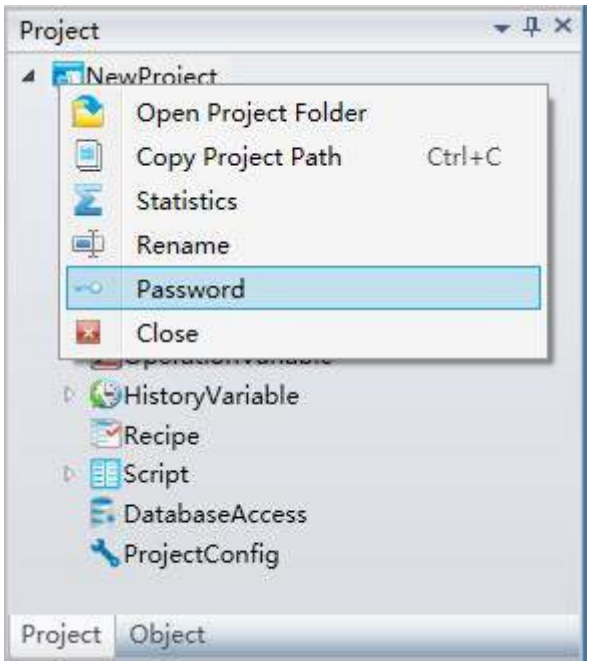
9.Restore: Restores the backed up project file.

10.Set to default: Selects the selected project as the preset project. This project will be executed when the DIAView execution environment is enabled.

11.Runtime: Executes the selected project.

Once a project is successfully created, the project tree index will be displayed in the project window to the right. Right-click project name and the right-click menu will display the functions.

"Open project folder", "Statistics", "Rename", " Password" and "Close" as shown in the figure below:



1.Open project folder: Open the folder where the project is located.

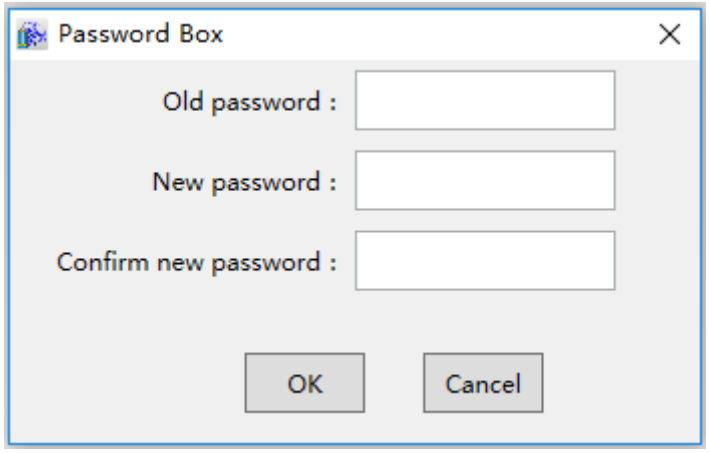
2.Copy Project Path: Copy the project's path.

3.Statistics: Show the statistics about IO points and the variable dictionary of the project.

4.Rename: Renames the project name.

5.Password: Sets a password for the project so that it can only be opened when the correct password is entered.

Right-click the mouse on the root of the project in the project tree index and then click “Password” to open the “Set and change password” window, as shown in the figure below:



The meanings of each setting in the configuration window are as follows:

Old password: Enter the original password of the project (if setting the project password for the first time, the preset original password is empty.

New password: Enter the new password.

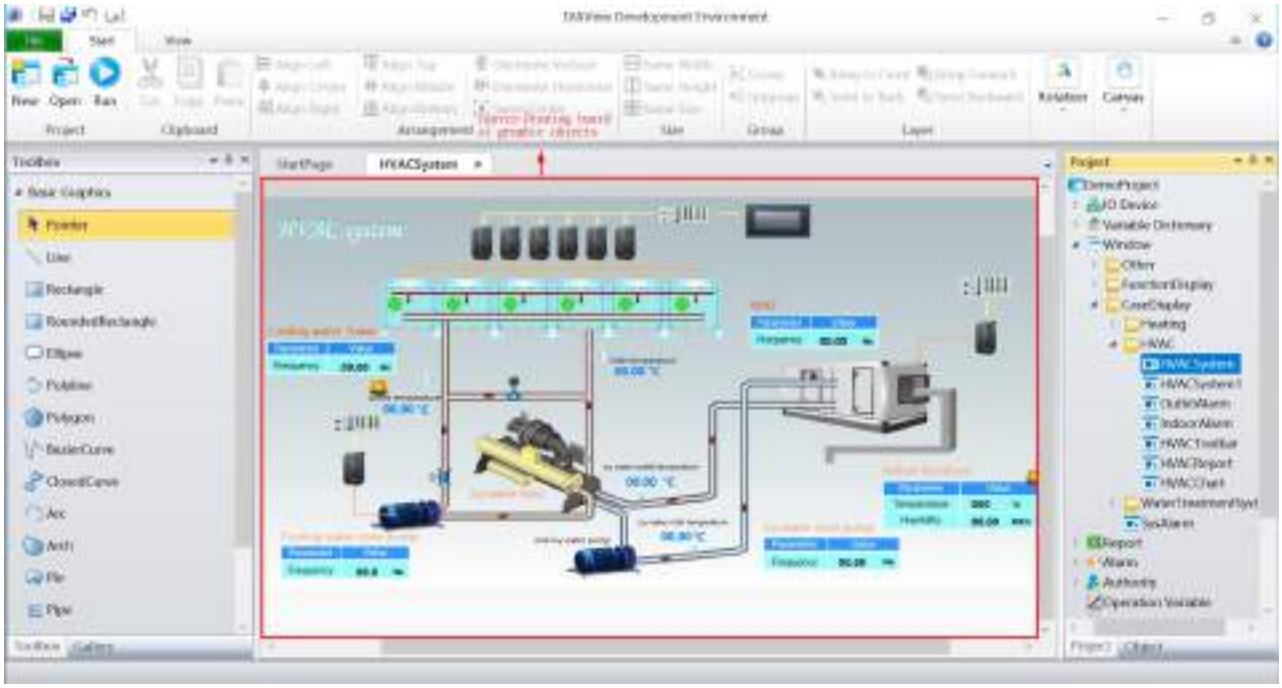
Confirm new password: Enter the new password again.

Opening password-protected projects requires entering the password, as shown in the figure below:

6.Close: Closes the currently opened project.

3.2.4 GUI development window introduction

After adding a project in the DIAView software development environment, right-click on the “Window” node in the project tree index to the right →“Add window” and the system will create a window that allows graphic drawing and editing, as shown in the figure below:



Its components are based on “3.2.1 Framework composition of the development environment”.

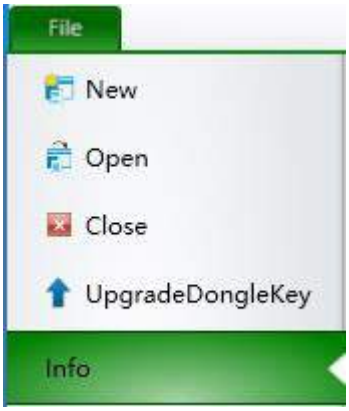
3.3 DIAView quick tool bar introduction

The quick tool bar provide some common used operations to users,such as save,undo,redo etc.The functions of each button are as follows:

Button	Command	Description
	Save	Saves the sketchpad information currently editing in the graphic interface development window.
	Save All	Saves all sketchpad and other configuration information of the current project.
	Undo	Reverse cancels the previous operation starting from the last operation.
	Redo	Reverse cancels the previous operation starting from the last “Undo.”

3.4 DIAView main menu introduction

3.4.1 File menu

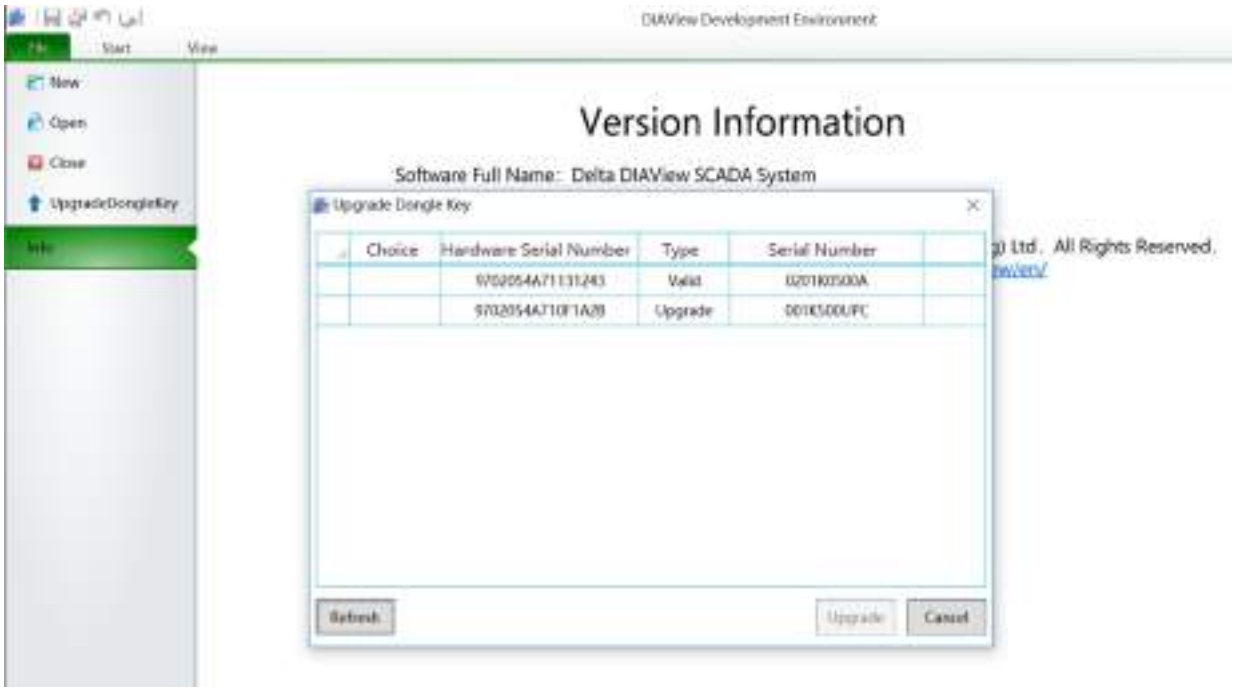


The file menu is used to manage the project; its function items and functions are as follows:

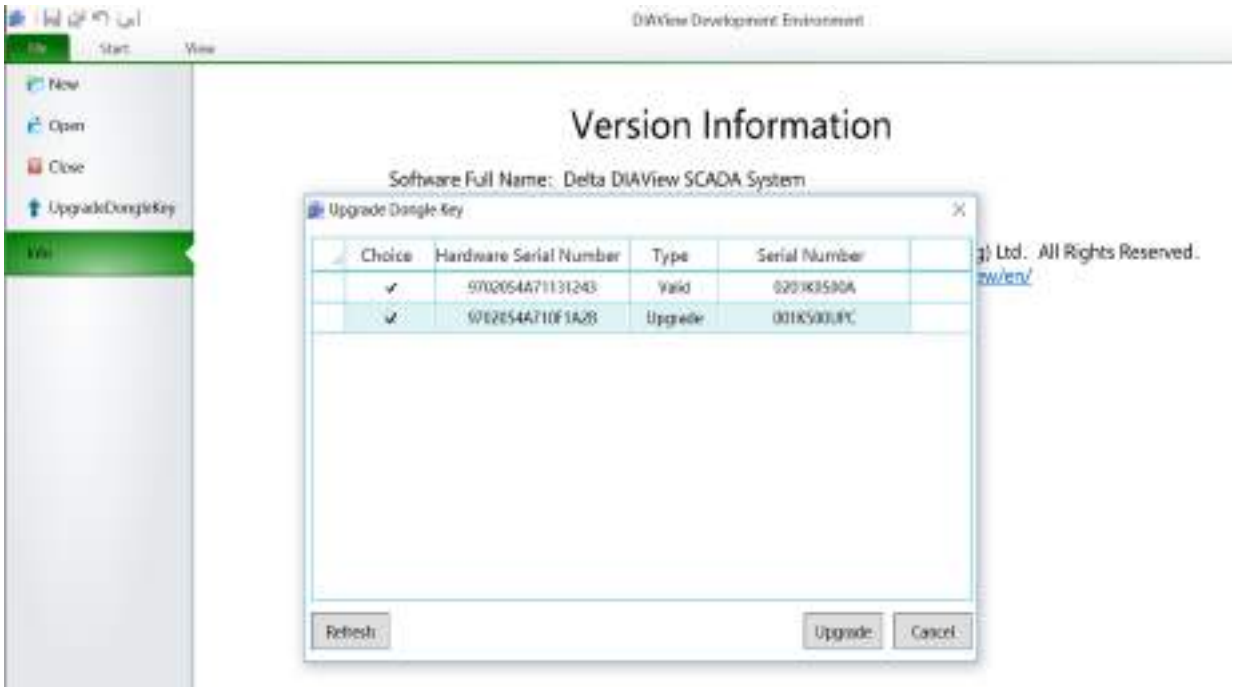
- **New:** Create a new project.
- **Open:** Open a created project in the development environment.
- **Close:** Close an opened project in the development environment.
- **UpgradeDongleKey:** Upgrade the dongle key of the DIAView , the specific steps are as follows:

Example 1: Network version 5 client 1000 point dongle upgrade to 10 client 1500 point

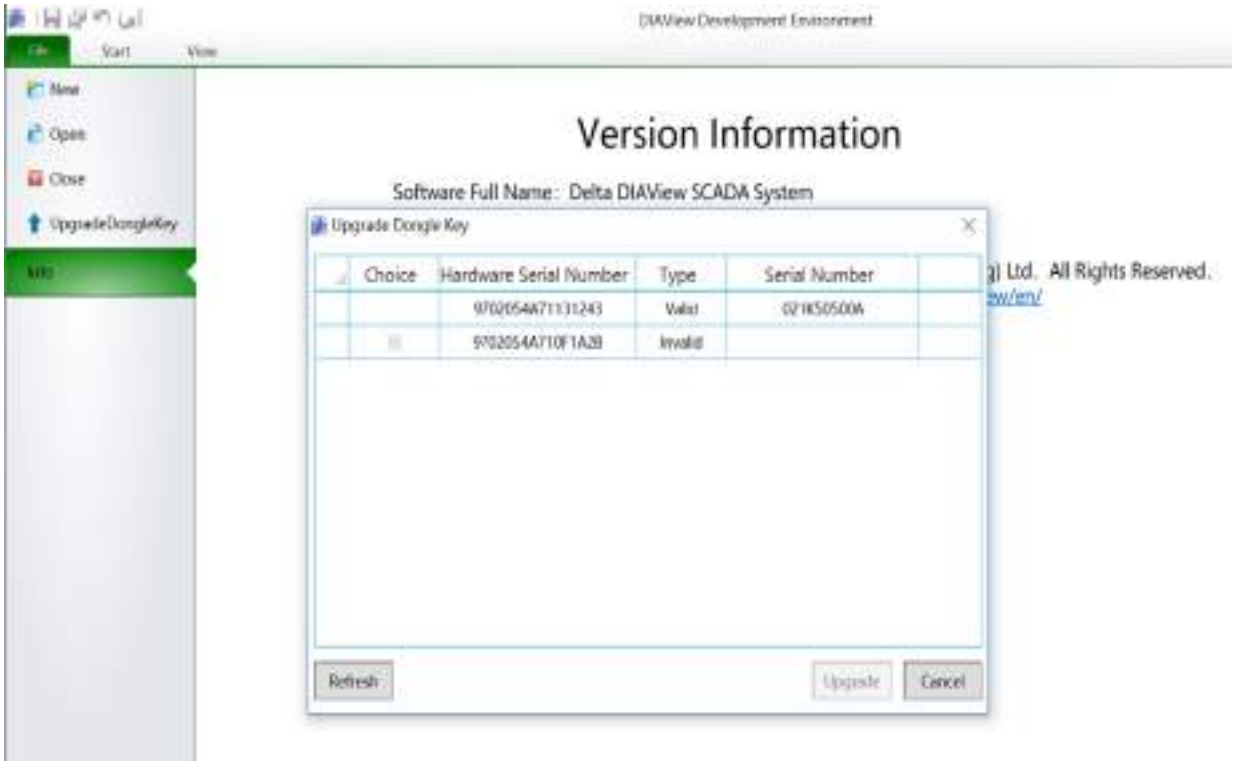
Step 1:Insert network version 5 client 1000 point dongle and 1500 point upgrade dongle into the computer, click upgrade dongle from the file menu, and open the upgrade dongle configuration window, as shown in the figure below:



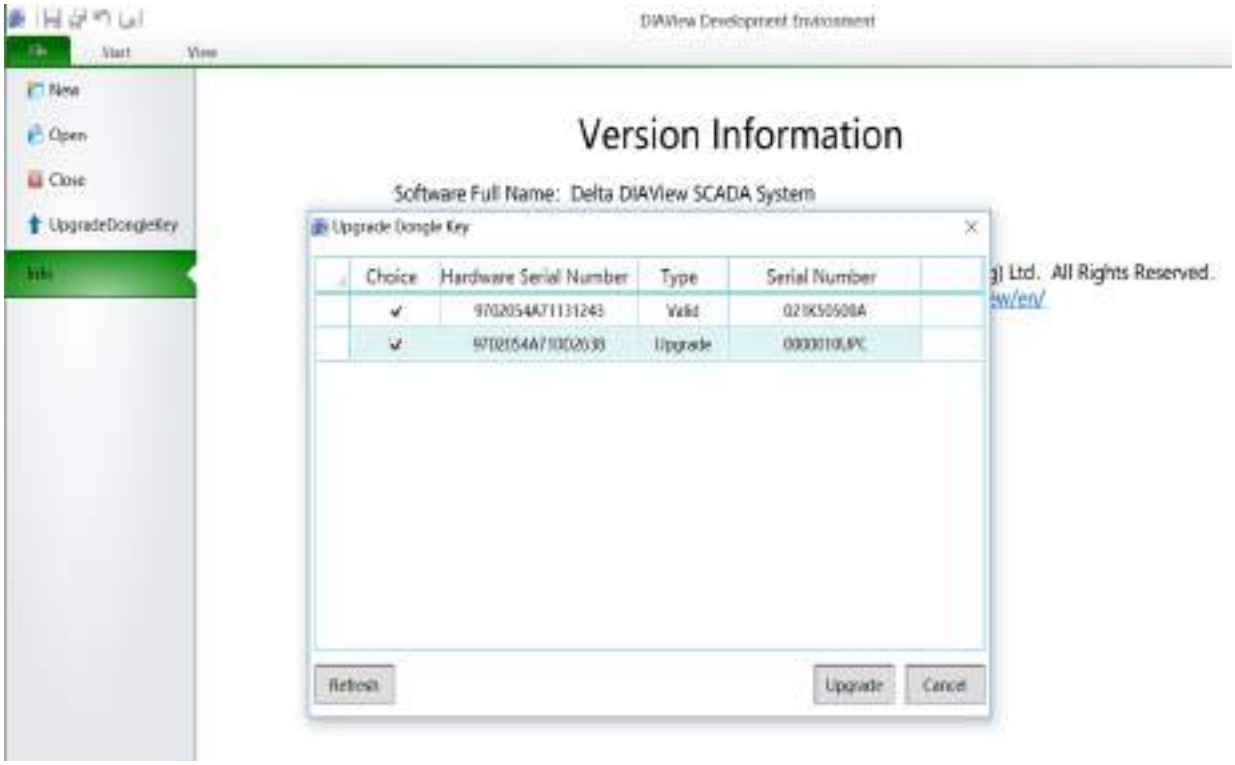
Step 2:Select the options of dongle and upgrade dongle respectively, and click the merge upgrade button at the bottom right to upgrade the dongle data point (unplug the dongle during the upgrade process), as shown in the figure below:



Step 3:After merging and upgrading, a new 1500 encryption dog is generated for 5 clients. The upgraded encryption dog can only be used once and then fails, as shown in the figure below:



Step 4:Network version 5 client 1500 point dongle and 10 client upgrade dongle into the computer, in the upgrade dongle window select the dongle and upgrade dongle to upgrade the combination, as shown in the figure below:



Step 5:Merge and upgrade to generate a new 10 client 1500 point dongle, as shown in the figure below:

- **Info:** View the related version information of the DIAView.

3.4.2 Start menu

The function of the start menu is mainly used for project management and layout operations of the location, layer and arrangement of the graphic objects in the graphic interface development. The availability of each button is determined by the selected graphic object and operations in the sketchpad. When the mouse cursor is placed on top of the button, the system will display the reminder window with the description of this button function below the mouse. The functions of each button are as follows:

Button	Command	Description
	New	Create a new project
	Open	Open a created project in the development environment.
	Run	Execute the project currently opened in the development environment.
	Cut	Cut one or multiple selected graphic objects from the sketchpad to the clipboard.
	Copy	Copy one or multiple selected graphic objects from the sketchpad to the clipboard.
	Paste	Paste one or multiple graphic objects on the clipboard to the sketchpad.
	Align Left	Align two or more selected graphic objects in the sketchpad to the left side of the graphic object that was selected first.
	Align Center	Align two or more selected graphic objects in the sketchpad to the same vertical line of the geometric center of the graphic object that was selected first.
	Align Right	Align two or more selected graphic objects in the sketchpad to the right side of the graphic object that was selected first.
	Align Top	Align two or more selected graphic objects in the sketchpad to the top side of the graphic object that was selected first.
	Align Middle	Align two or more selected graphic objects in the sketchpad to the same horizontal line of the geometric center of the graphic object that was selected first.
	Align Bottom	Align two or more selected graphic objects in the sketchpad to the bottom side of the graphic object that was selected first.
	Distribute Vertical	Distribute three or more selected graphic objects in the sketchpad evenly in the vertical direction.
	Distribute Horizontal	Distribute three or more selected graphic objects in the sketchpad evenly in the horizontal direction.

	Same Center	Align two or more selected graphic objects in the sketchpad to the center of the graphic object that was selected first.
	Same Width	Make two or more selected graphic objects in the sketchpad have the same width.
	Same Height	Make two or more selected graphic objects in the sketchpad have the same height.
	Same Size	Make two or more selected graphic objects in the sketchpad have the same width and height.
	Group	Groups the selected multiple graphic objects into a new graphic when there are multiple graphic objects in the sketchpad.
	Ungroup	Splite the group graphic into separated graphic objects.
	Bring to Front	Move the selected graphic object at the top layer when there are multiple graphic objects stacked together in the sketchpad.
	Send to Back	Move the selected graphic object at the bottom layer when there are multiple graphic objects stacked together in the sketchpad.
	Bring Forward	Move the selected graphic object up one layer when there are multiple graphic objects stacked together in the sketchpad.
	Send Backward	Move the selected graphic object down one layer when there are multiple graphic objects stacked together in the sketchpad.
	Rotate Clockwise	Rotates the selected graphic object in the sketchpad 90 degrees clockwise using the center of the graphic object as the origin.
	Rotate Counter Clockwise	Rotates the selected graphic object in the sketchpad 90 degrees counterclockwise using the center of the graphic object as the origin.
	Flip Vertical	Flip the selected graphic object in the sketchpad vertically using the Xaxis of the center of the graphic object as the axis of symmetry.
	Flip Horizontal	Flip the selected graphic object in the sketchpad horizontally using the Yaxis of the center of the graphic object as the axis of symmetry.
	Show Ruler	Display or hide the framework ruler.

Enable	Enable Snap	When enable snap function, draw and move objects by the grid.
	Enable Pan	When enable pan function, handshape tool will be displayed to move the board.
100%	Zoom Ratio	Adjuest the zoom ratio of the canvas.

3.4.3 View menu

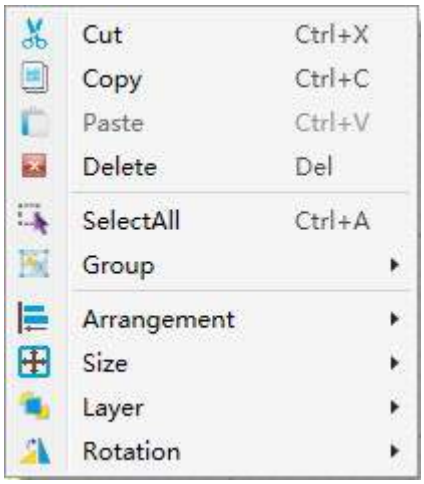
The view menu is used to open the configuration windows including the property window and project tree index etc. under the development environment; its button functions are as follows:

Button	Command	Description
	Start Page	Open the starting page window
	Project	Open the project folder window.
	Object	Open the object browser window.
	Property	Open the property window.
	Extension	Open the extended property window.
	Animation	Open the animation window.
	Event	Open the event window.
	Toolbox	Open the toolbox.
	Gallery	Open the gallery
	Error list	Display compilation error messages.

	Save layout	Save the window layout
	Reset layout	Restore the window layout

3.5 DIAView right-click menu introduction

Right-clicking the mouse in the sketchpad of the DIAView software graphic interface development window will open the right-click menu, as shown in the figure below:



The functions of the sketchpad right-click menu are as follows:

Command	Description
Cut	Cut the selected graphic object onto the clipboard.
Copy	Copy the selected graphic object onto the clipboard.
Paste	Paste the graphic object on the clipboard onto the sketchpad.
Delete	Remove the selected graphic object from the sketchpad.
SelectAll	Select all of the graphic objects in the sketchpad.
Group	Make the multiple selected base graphic objects as a new group graphic object or split the group graphic object to individual sub graphic objects.
Arrangement	Align or distribute multiple graphic objects according to different methods.
Size	Resize the graphic object all of the graphic objects according to different methods.
Layer	Change the layer of graphic objects according to

	different methods.
Rotation	Rotate the graphic objects according to different methods.

3.6 DIAView shortcut key introduction

Shortcut key is also called hot key. It uses specific buttons or button combinations to execute an operation or command to replace certain tasks with the mouse; it is convenient and fast to use. Shortcut key available for the DIAView software is as shown in the table below:

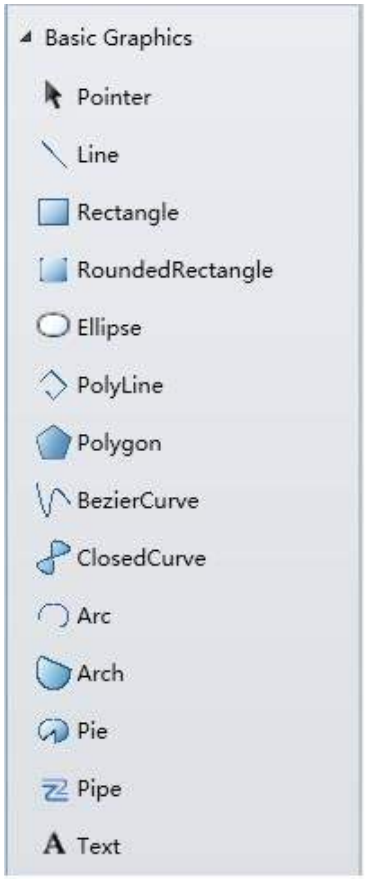
Shortcut key	Function/command
Ctrl ＋ O	Open project
Ctrl ＋ N	Create new project
Ctrl ＋ R	Execute the project
F11	Maximize/restore runtime environment
Ctrl ＋ C	Copy
Ctrl ＋ X	Cut
Ctrl ＋ V	Paste
Ctrl ＋ D	Advanced copy; equivalent to the combination of Ctrl ＋ C and Ctrl ＋ V.
Ctrl ＋ Z	Undo
Ctrl ＋ Y	Redo
Ctrl ＋ S	Save
Ctrl＋ G	Group
Ctrl ＋ U	Cancel group
Delete	Delete
↑	Move up
↓	Move down
←	Move left
→	Move right

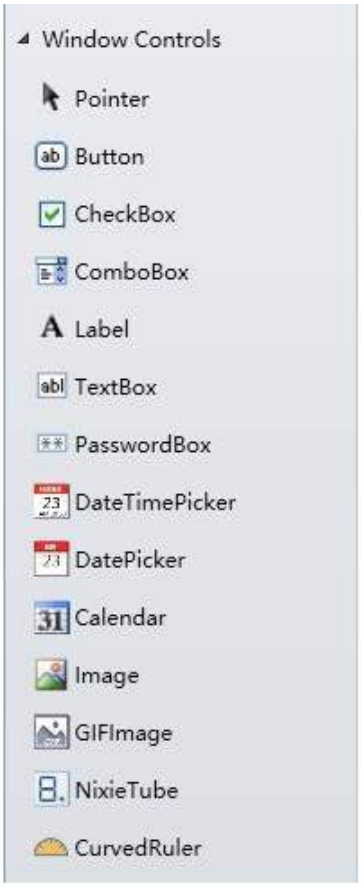
3.7 DIAView tool window introduction

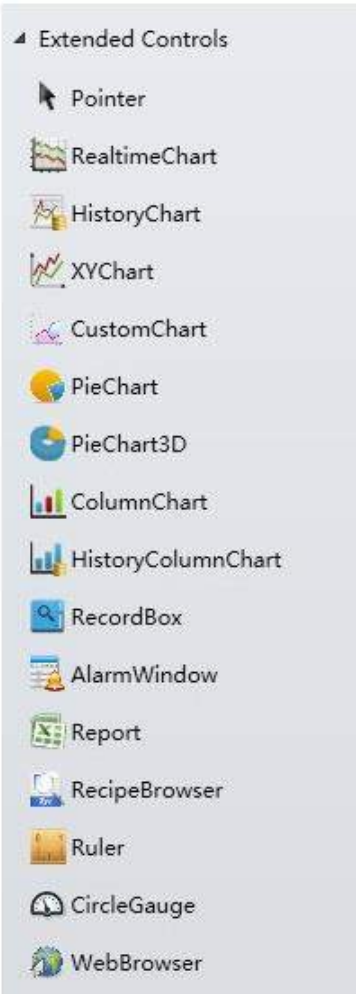
The tool window of the DIAView software refers to the “Toolbox” in the graphic development window. They provide basic graphic elements, graphic control units and frequently used graphic components and models required for graphic drawing. Users can also develop and define graphic models on their

own and add it to the graphic library. Drawing is also the basis for SCADA project development.

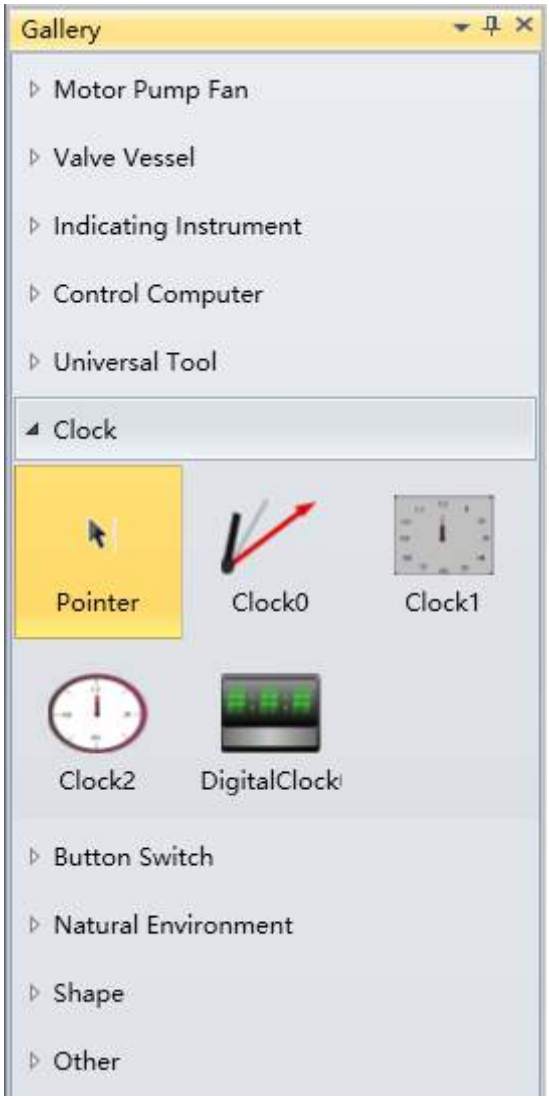
Basic graphics, Window controls, Extended controls in the “Toolbox” are as follows:





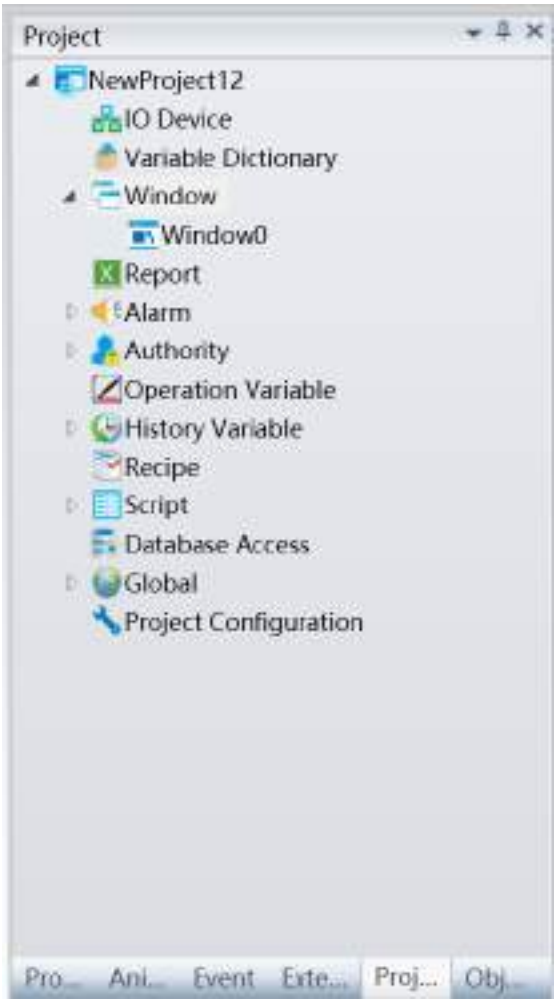


The Gallery is as follows:



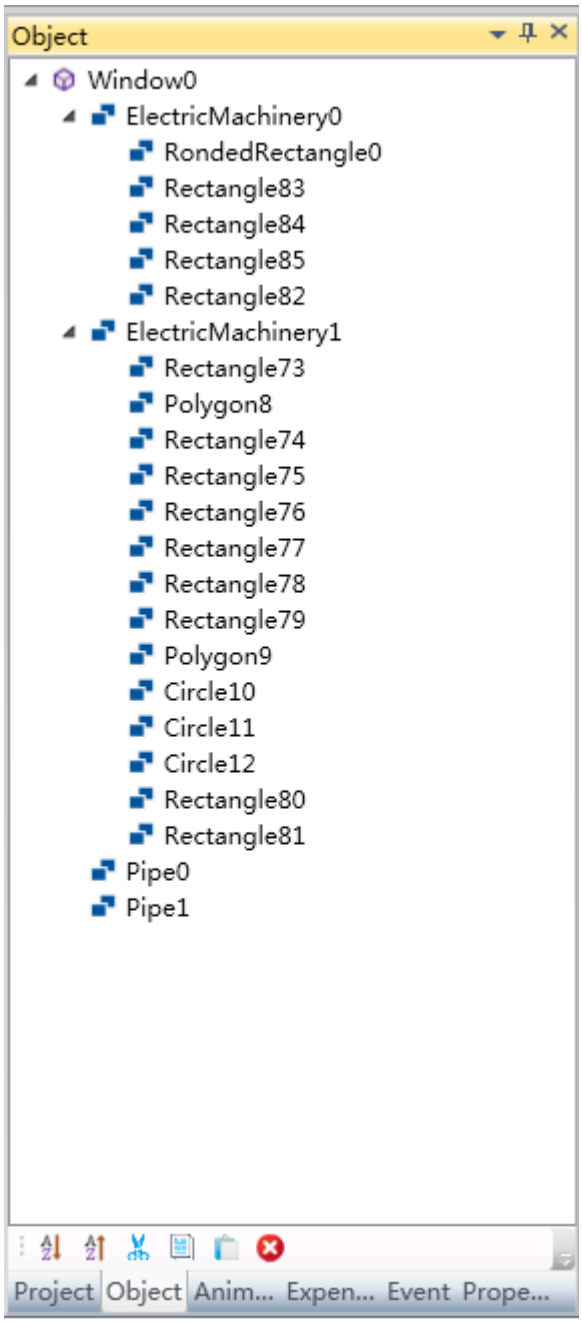
3.8 DIAView project window introduction

The project window is the project tree index display window. It lists all of the function nodes of the project and each node is for configuring project related functions and administrators. Click a certain node and the node will be selected. Then, users can double-click on it or use the right-click menu to perform related configuration operations. If related sub-items are already configured for this node, there will be a "□ " symbol in front of the node. Click this symbol to expand the sub-items and then double-click the sub-item to open it in the sketchboard and edit configuration operations of the sub-item content, as shown in the figure below:



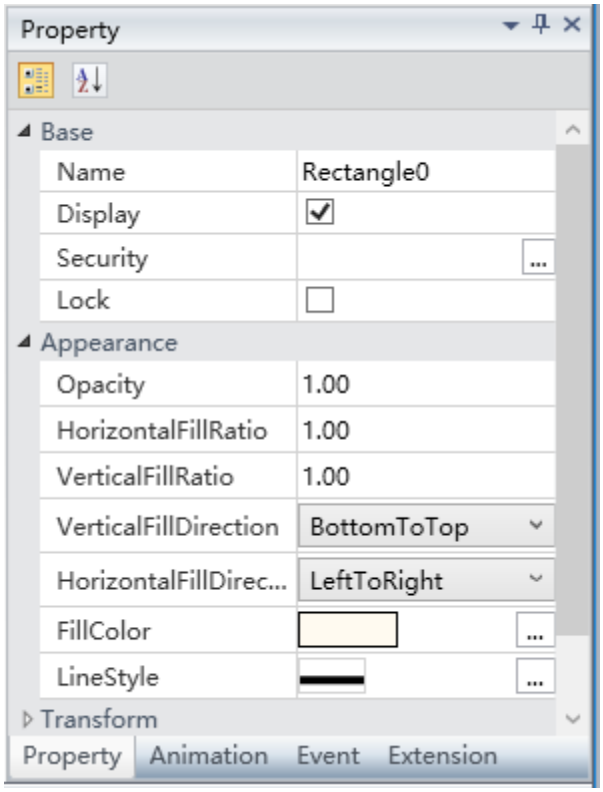
3.9 DIAView object window introduction

The object window is mainly used to display the current graphic components of the sketchboard in the opened window. When a certain component is selected, the corresponding graphic objects of this component are also selected in the sketchboard, as shown in the figure below:



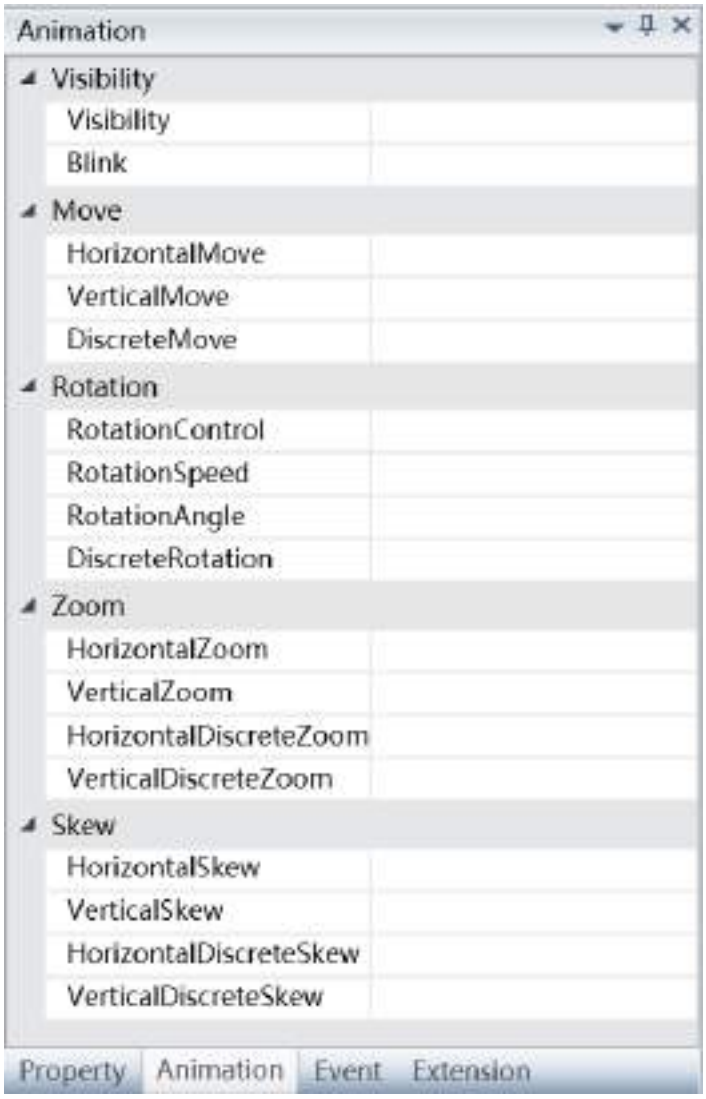
3.10 DIAView property window introduction

The property window manages related properties of objects. When any graphic is selected in the sketchboard, this window will display its corresponding properties in real-time. Properties include the name, position, size and color etc. of the graphic object. The user can modify and configure the property values in the property dialog, as shown in the figure below:



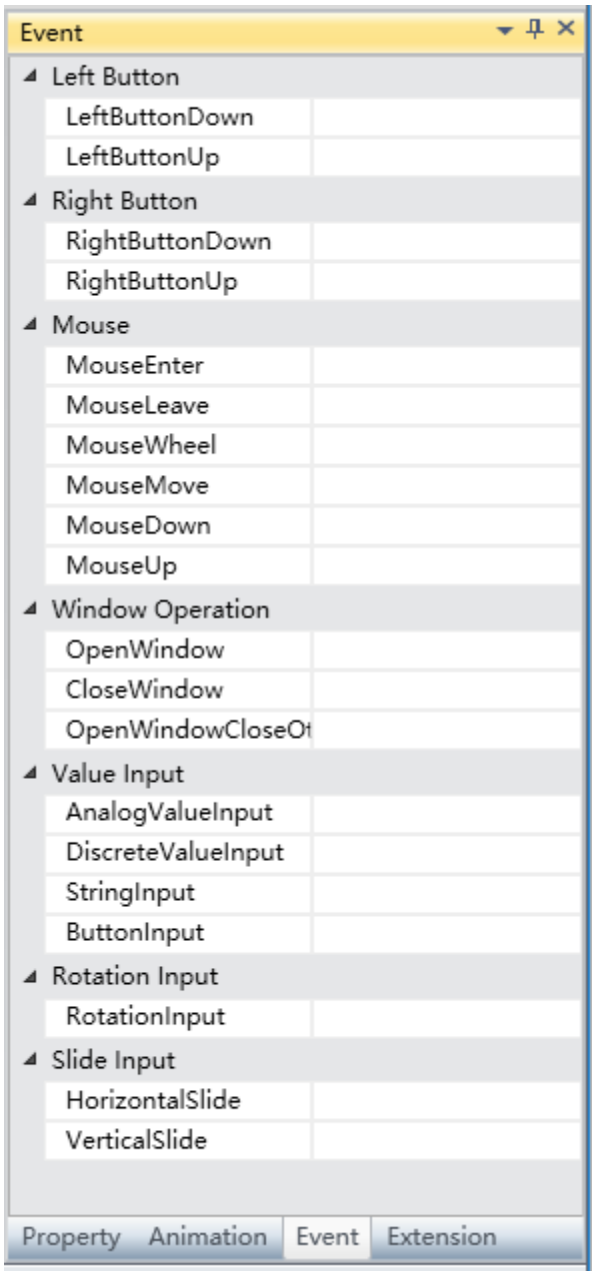
3.11 DIAView animation window introduction

The animation window is used to manage the animation configuration information of graphic objects. Users can add or delete animations for the graphic. Animation configuration requires connecting to related properties and variable data, as shown in the figure below:



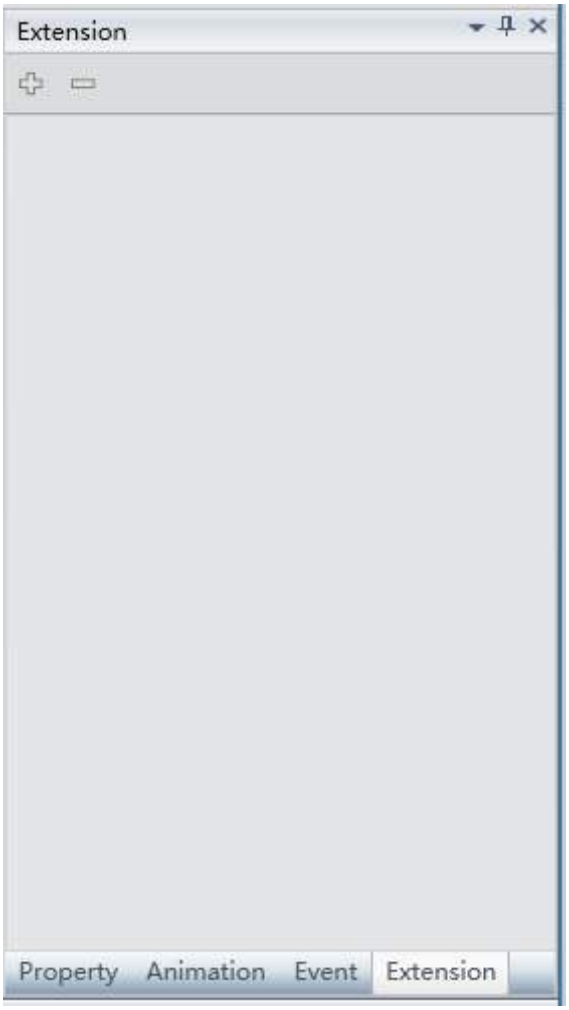
3.12 DIAView event window introduction

The event window is used to manage the event configuration information of graphic objects. Users can add or delete graphic events. Event configuration requires connecting to related mouse and keyboard events and variable data, as shown in the figure below:



3.13 Extended property window introduction

Extended property allows user to configure self-defined property for group graphics,in this window,users can not only add or delete extended property but also can rename, select type, set default value, add description and relate script with the property in the popups, as shown in the figure below:



4. Development Step Instruction

In order to quickly and skillfully use the DIAView software to develop projects efficiently, this chapter introduces the common steps used to develop projects with the DIAView software. Users may refer to these steps for common project developments, but since there are different project applications, users should select the development steps suitable for their specific needs.

Common steps of project development with DIAView are as follows:

Step 1: Execute the DIAView software development environment.

Step 2: Add project: Click the “Add project item from the “File” or “Start menu in the main interface of the development environment and configure information including the project name and project storage path etc., and then the system will generate and display the initial tree index in the “(Project)” window.

Step 3: Define variables: Variables are medias used to record the changes of certain data of the DIAView software in real-time. It dynamically displays the data source and display form of the controlled field site data. Therefore, all variables participating in the information handshake of the project must be defined.

Step 4: Create IO Communication: Create a physical data channel between the upper and lower machines of the DIAView software and the monitoring and control field equipment, and set the information including connection method, communication parameters and communication variables etc.

Step 5: Create window: This means creating the graphic interface for the system. Draw the graphics for field simulation pictures and configure related information including properties and charts etc.

Step 6: Define animations and events: Defines the changes and actions for the graphic objects in the window to take when its properties changes according to the data acquired during project execution.

Step 7: User program configuration: Defines condition programs and set the project to perform script programs for specific function when related data changes, and configure time program to define operation programs to execute when the screen starts/stops or at specific times.

Step 8: Alarm configuration: Sets the warning value of the range of variable value changes, and specify the alarm level and related alarm configurations when the range of the changed variable value is lower than or exceeded the warning value.

Step 9: History record setting: Specifies certain more important variables and records their value changes in order to use the “ (History chart)” or “ (Report)” to connect these variables to perform data summary and analysis.

Step 10: Project configuration: Sets the related parameters for the connection between the DIAView software and third-party databases in order to achieve handshakes between the real-time project data and other commercial databases, and sets the project execution options and start screen.

Step 11: Security zone setting: Sets the project operating authority. Divides the various components in the project into different security zones. Objects have different operating authorities in different security zones

Step 12: User setting: Sets the account and authority (security level) of the system operators.

Step 13: Runtime: Executes the project and view whether the screen effects and functions meet the design requirements. If it does, then it can be executed and used, or else please redesign starting from Step 3 to perfect the project according to your need.

5. IO Communication

5.1 Overview

IO communication refers to the DIAView software connecting to a southbound equipment through specific communication protocols to perform data acquisition and transmission. It has high reliability, stability, instantaneity and powerful data processing capability.

The IO communication of the DIAView software has the following features:

- Supports various connection methods;
- Supports standard Modbus communication protocol;
- Able to test the connection status of the equipment directly during communication configuration;
- Supports communication status feedback and dynamic starting and stoping of devices;
- Supports OPC service;
- Has comprehensive troubleshooting strategies;
- Supports equipment communication gateway function to save hardware gateway costs;
- The collection of Delta's communication driver can perfectly match all Delta series products, making communication parameter configurations easier, more flexible and more efficient.

5.2 Channel and device

A channel is the data transmission media during the IO communication process; for example: cable circuit interfaces, cable mounts, hub devices and software channels etc. Channels supported by the DIAView software includes: serial port channel, Ethernet channel, OPC channel and simulator channel.

Device is the physical object for the DIAView software to perform IO communication. The DIAView software can read data from the device and also write data into the device.

The Device is divided into hardware device and software device, including programmable logic controllers (PLC), inverters, modules, digital instrumentation, smart controllers, monitoring sensors, distributed control systems (DCS), loop controllers, remote terminal units (RTU) and third-party software etc.

Specific device and example:

Device	Example

PLC	Delta PLC, Modicon Siemens,Mitsubishi,Rockwell etc.
OPC	Third-party OPC servers
Simulator	Simulator
Power Meter	Delta Power Meter

5.3 Driver

The driver is a set of program that can operate the device. It determine show to operate the device register or device memory, and it is an important bridge for communication between the DIAView software and field devices. Different device manufacturers support different device drivers, but they generally support standard communication protocols.

Protocols currently supported by the DIAView software includes: Modbus, OPC, PPI,MPI,PROFIBUS,Ethernet/IP,FINS, HostLink etc.

Frequently used drivers for the IO Server of the DIAView software are as shown in the table below:

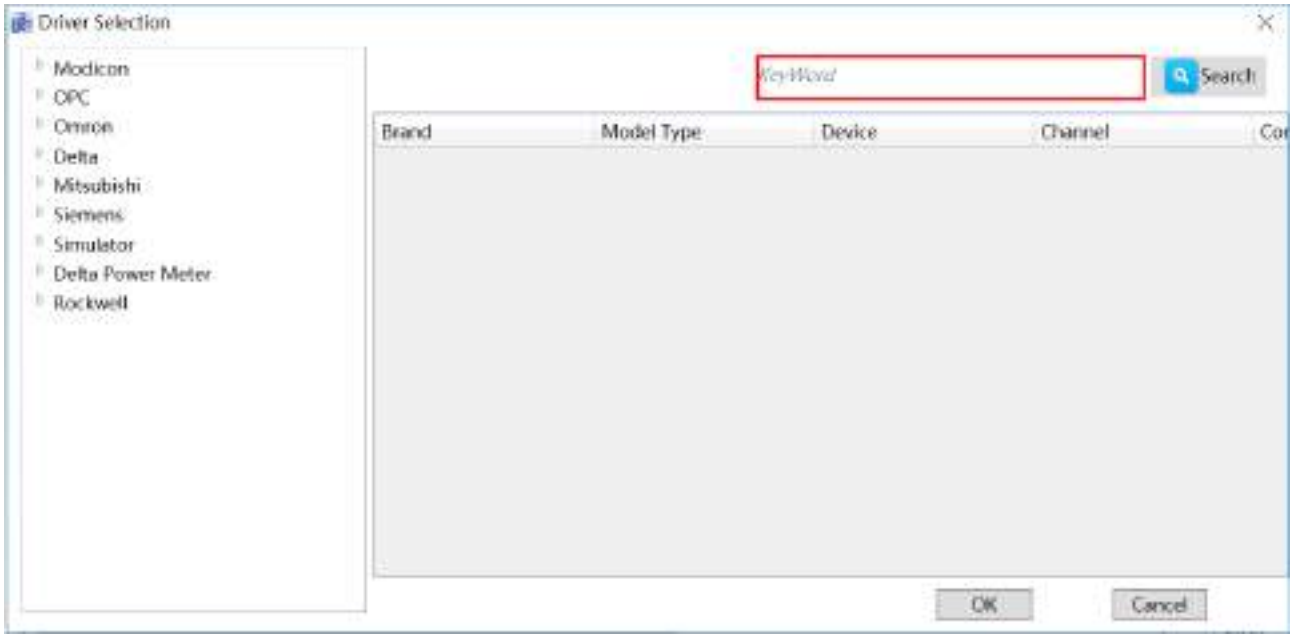
Driver type	Example
Modicon	Modbus TCP Modbus Serial RTU Modbus Serial ASCII
OPC	OPC、OPCUA
Omron	FINS TCP FINS ASCII HostLink ASCII
Delta	Delta AS series:TCP、RTU、ASCII Delta AH series:TCP、RTU、ASCII Delta DVP series:TCP、RTU、ASCII

	DIALink、DIALinkCNC
Mitsubishi	Mitsubishi FX Serial Mitsubishi ProFX Serial Mitsubishi Q Serial Mitsubishi Q EtherNet Mitsubishi FX EtherNet
Siemens	S7300 TCP S71200 TCP S7300 MPI S7200 TCP S7200 PPI S7200 Smart TCP
Simulator	Simulator
Delta Power Meter	DPM-C530 Ethernet DPM-C530 Serial
Rockwell	Rockwell Controllogix EtherNet

➢**When creating a new driver, you can quickly find the driver through the retrieval function:**

Step 1: Right-click on the “IODevice” node in the project management area and select” New Device”; the menu selection in the figure below will appear:

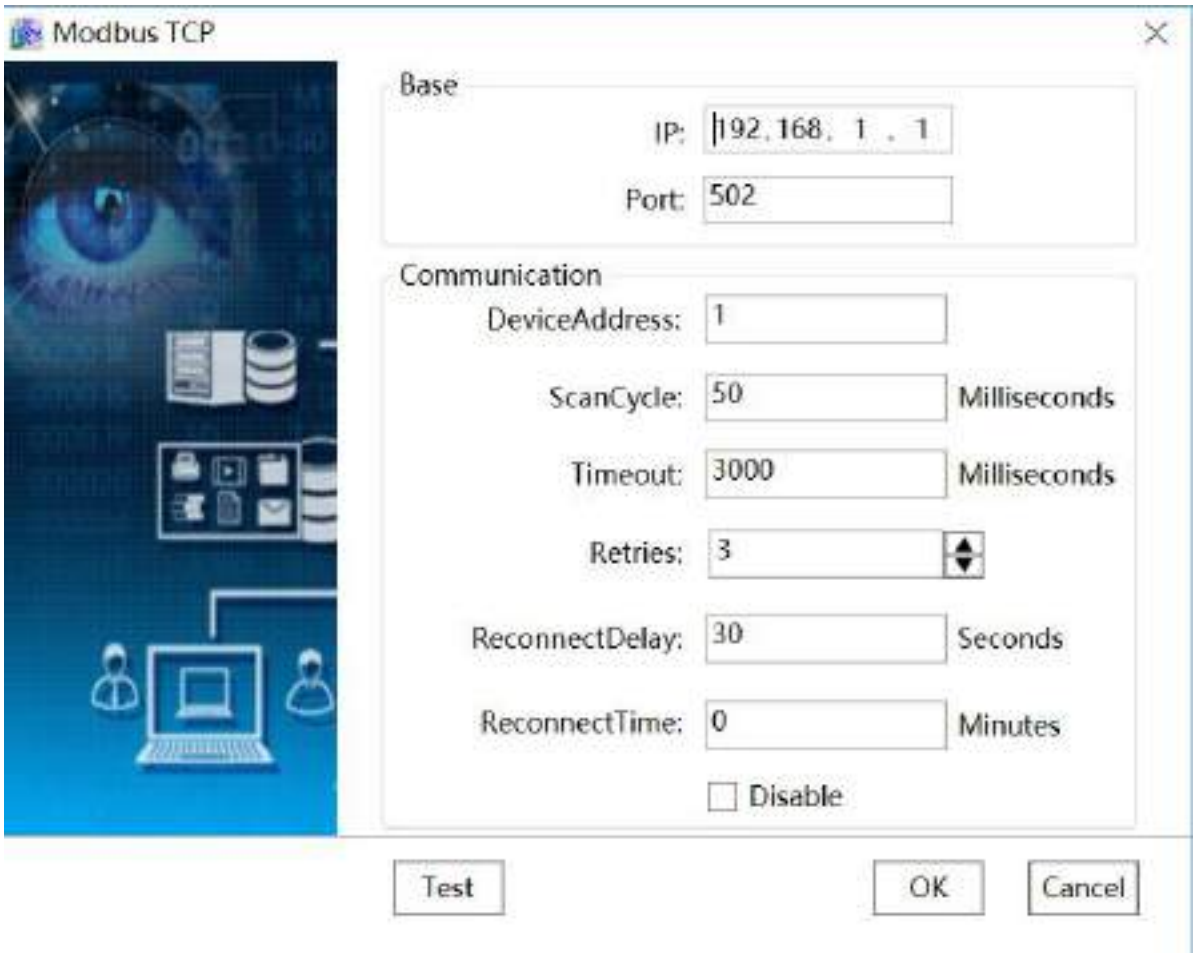
Step 2: After clicking the search button, the IO communication driver search box as shown in the following figure. Enter the driver name in the keyword box and click the search button to find the desired driver.



5.4 Troubleshooting strategies

When the communication is interrupted or disconnected during system execution,the DIAView software will try to restore the communication between devices by processing following the troubleshooting strategies.

The troubleshooting strategies set for the DIAView software is as shown in the figure below:



Communication property:

Communication property:

Property	Description
DeviceAddress	Address number of the connected device
ScanCycle	The scan cycle of the IO communication data is preset to 100 milliseconds.

Timeout	If the device did not respond for an extended period of time, it will be determined to have communication timeout according to the time length set. The timeout length is preset to 1000 milliseconds
Retries	This is the number of times the system will try to reconnect to the device when there is communication timeout. The default number of retries is 3 times.
ReconnectDelay	This option is used to set the period of time of delay before trying to reconnect again when the number of retries have reached the limit and reconnect was unsuccessful. The reconnect delay time is default to 30 seconds.
ReconnectTime	Default to 0 minutes: No time limit, which means it will continually try to reconnect with the device; Other value: sets the reconnect time length. If the system cannot connect and receive responds from the device within the reconnect time limit, it will discard the connection to that device.
Disable	The current configuration equipment will be disabled from communicating with the DIAView software.

5.5 IO variable

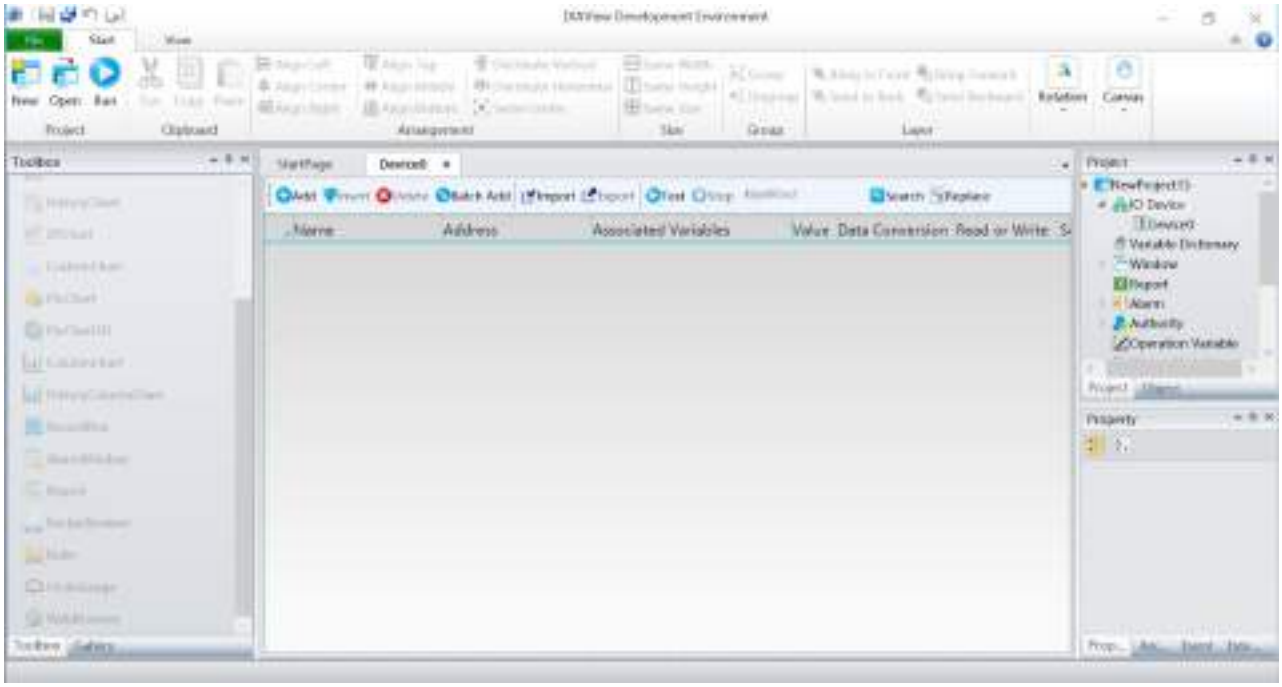
The IO variables are the external device variables created in the DIAView software. They are acquired directly from the register addresses inside related devices. Association mapping between the IO variables and the project variables inside the DIAView software is created to allow the DIAView software to connect with devices to perform information handshake.

The IO parameters are related to device register addresses. They have different access mode including read/write, read-only and write-only, and are able to acquire linear data and root operations.

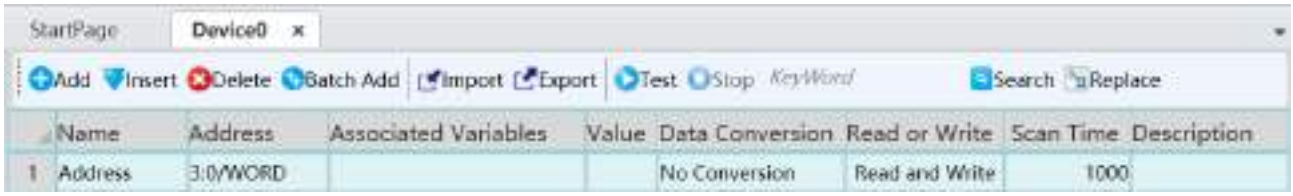
1. Creating IO variable

Example 1: Creating IO variables based on the "Serial port device"

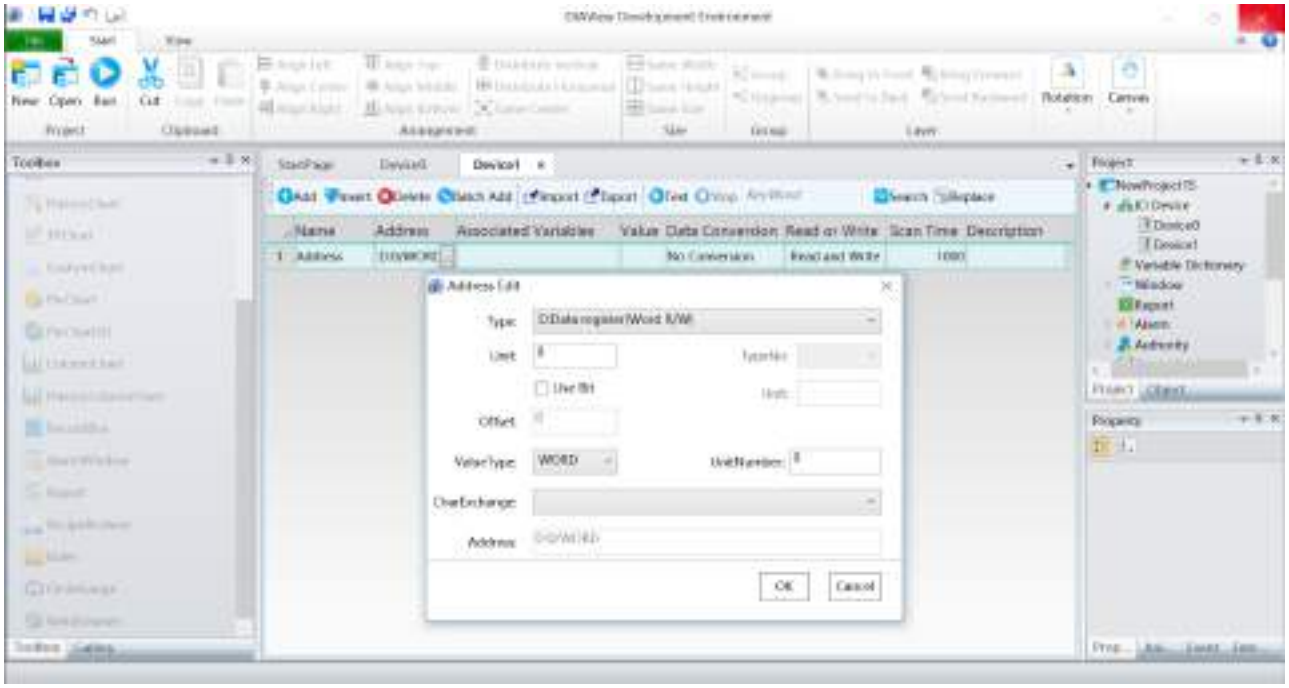
Step 1: Double-click the created “Serial port device” under the “IO Communication” node of the project tree index to open the serial port device IO variable configuration window, as shown in the figure below:



Step 2: Click the “Add ” button to add IO variable:



Step 3: Rename the IO variable name “Address” to “D0” and click the button in the Address field to open the address property configuration window to configure the address:



Parameter descriptions

Register type

Register type	Read/Write Mode	Description	Remark
CB: Counter (Bit)	Read/write	Counter bit device	DVP，AH500
CW: Counter (Word)	Read/write	Counter character device	DVP，AH500
HCB: High Speed Counter(Bit)	Read/write	32 bit high speed counter bit	AH500
HCDW: 32Bit Counter(DWord)	Read/write	32 bit high speed counter	AH500
D: Data Register	Read/write	Data register character	DVP，AH500
SED: DR (SE Device)	Read/write	Data register character	DVP 12SE
S: StepSpot	Read/write	Stepping point relay bit device	DVP，AH500
SM: Special Auxiliary Relay(Bit)	Read/write	Special auxiliary flag bit device	AH500

SR: Special Data Register(Word)	Read/write	Special data register	AH500
TB: Timer (Bit)	Read/write	Timer bit device	DVP，AH500
TW: Timer (Word)	Read/write	Timer character device	DVP，AH500
X: External Input Relay	Read only	Input relay bit device	DVP，AH500
XB: External Input Relay(Bit)	Read only	Input relay bit device	AH500
XW: External Input Relay(Word)	Read only	Input relay character device	AH500
Y: External Output Relay	Read/write	Output relay bit device	DVP，AH500
YB: External Output Relay(Bit)	Read/write	Output relay bit device	AH500
YW: External Output Relay(Word)	Read/write	Output relay character device	AH500
M: Auxiliary Relay	Read/write	Auxiliary relay bit device	DVP，AH500
E: Pointer	Read/write	Indexed register	AH500

Value type

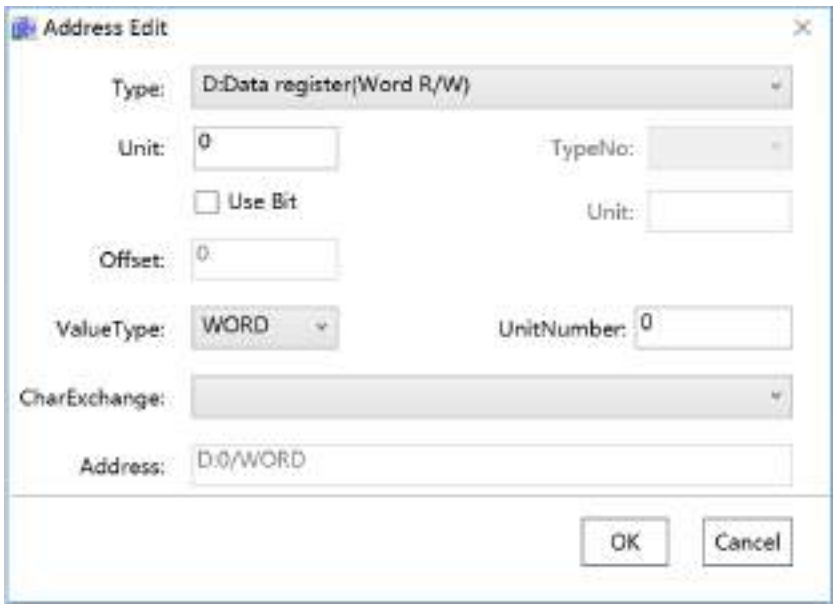
Value type	Description
BYTE	Byte
UBYTE	Unsigned byte
WORD	Word
UWORD	Unsigned word
DWORD	Double word
UDWORD	Unsigned double word
FLOAT	Single-precision floating point
DOUBLE	Double-precision floating point

STRING	String

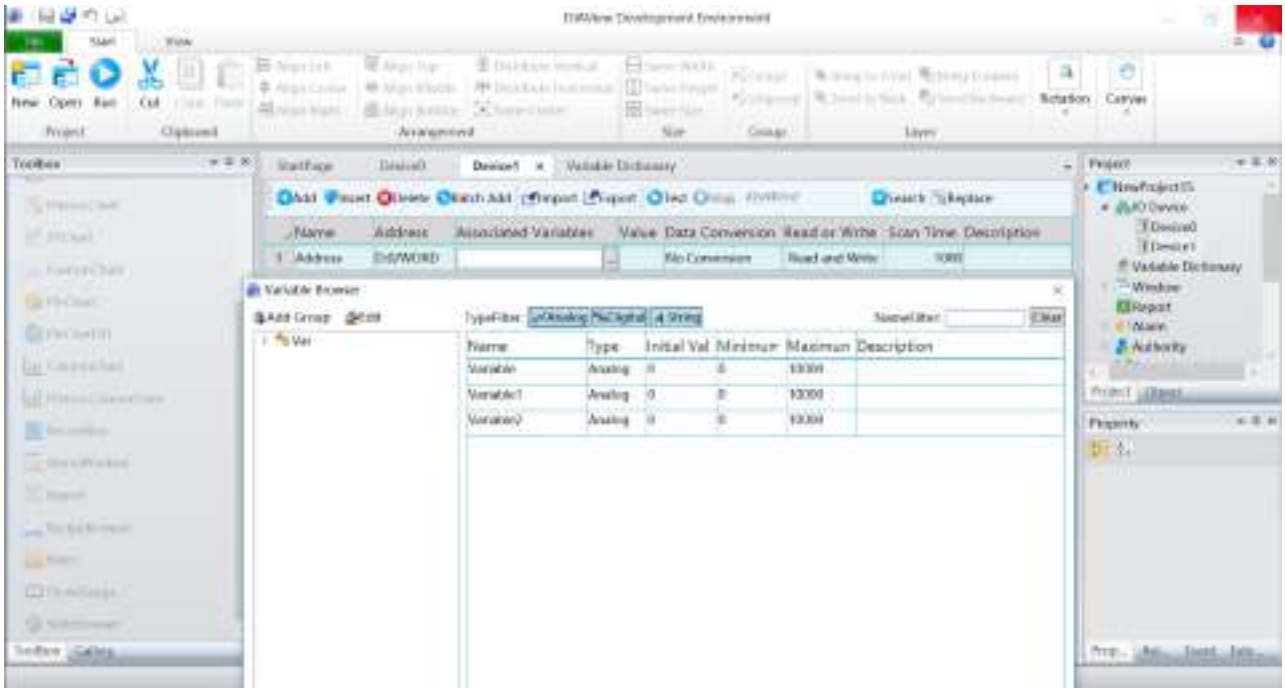
Character exchange

Byte exchange	Name
No Swap (01) No Swap (0123) No Swap (01234567)	No exchange
Swap BYTE (10)	Byte exchange (Word)
Swap BYTE (3210) Swap WORD BYTE (1032) Swap WORD (2301)	Byte exchange (DWord)
Swap BYTE (76543210) Swap WORD BYTE (10325476) Swap DWORDBYTE (32107654) Swap WORD (67452301) Swap DWORDWORD (23016745) Swap DWORD (45670123)	Byte exchange (Double)

Step 4: Select register type: “D:WORD R/W (D: Data register (WORD R/W))”, unit number: 0, keep the default value for other items, as shown in the figure below:



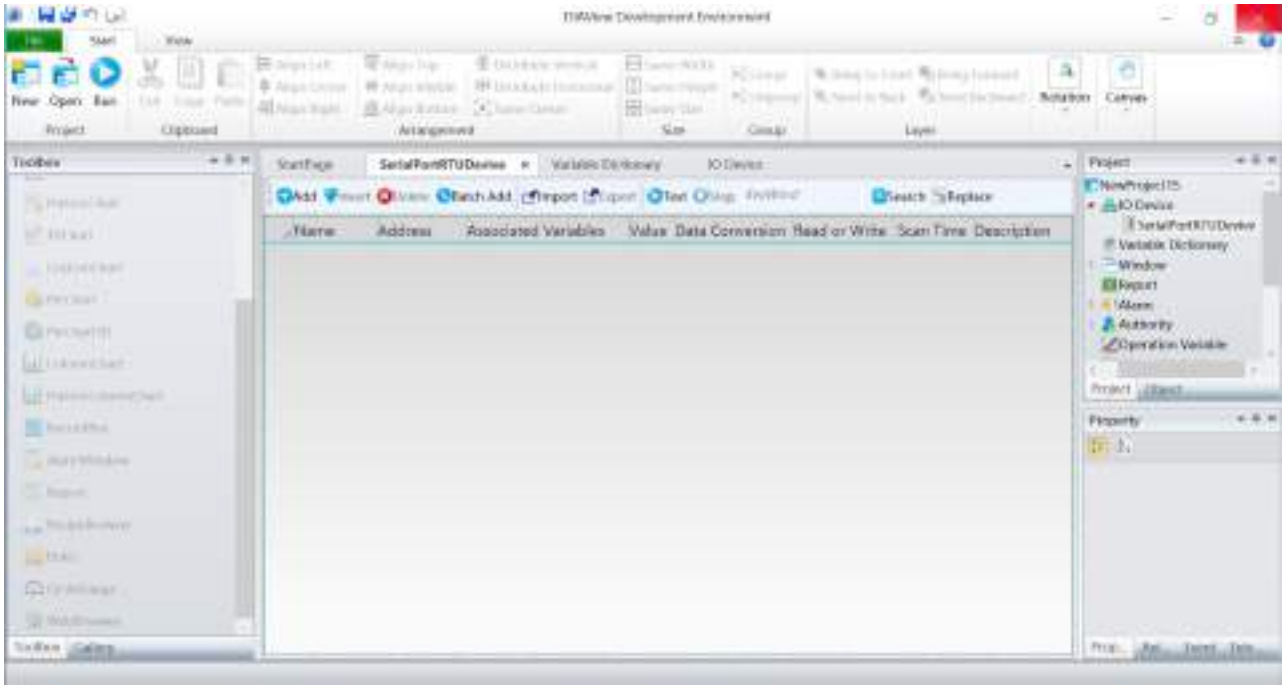
Step 5: Click the button in the【Variable】field to open the variable browser and select the previously created internal project variables, to complete the association mapping between the IO variables and internal project variables:



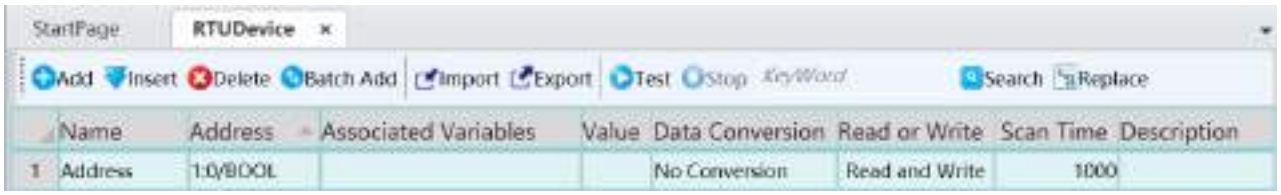
Step 6: Use default values for the bits in other fields; this completes the creation of an IO variable:

Example 2: Creating IO variables based on the "Serial port RTU device"

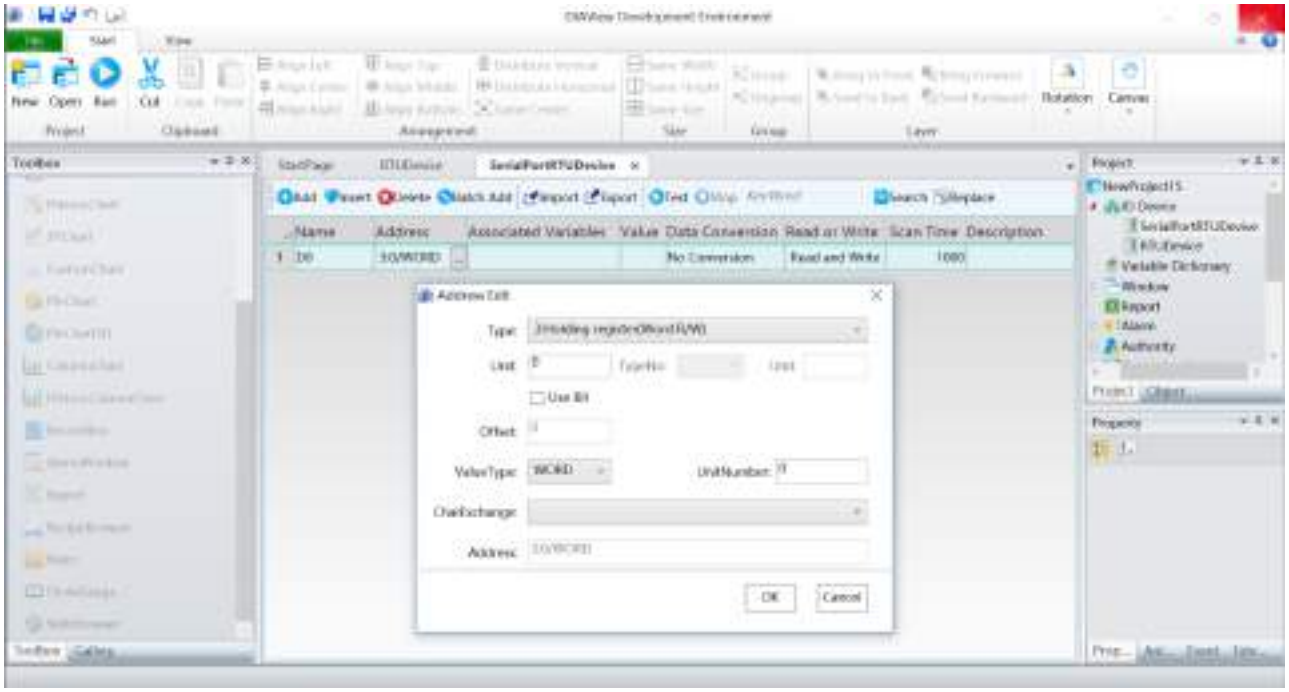
Step 1: Double-click the created “Serial port RTU device” under the “IO Communication” node of the project tree index to open the serial port RTU device IO variable configuration window, as shown in the figure below:



Step 2: Click the “Add” button to add IO variable



Step 3: Rename the IO variable name “Address” to “D0” and click the button in the " Address" field to open the address property configuration window to configure the address Parameter descriptions:



Register type：

Register	Description	Property	Function code	Default value
1	Coil output	Read/write	1,5	Bool
2	Digital input	Read only	2	Bool
3	Holding register	Read/write	3,16	Word
4	Analog value input	Read only	4	Word
5	Expanded output register	Read/write	21	Word
6	Expanded input register	Read only	20	Word

Value type:

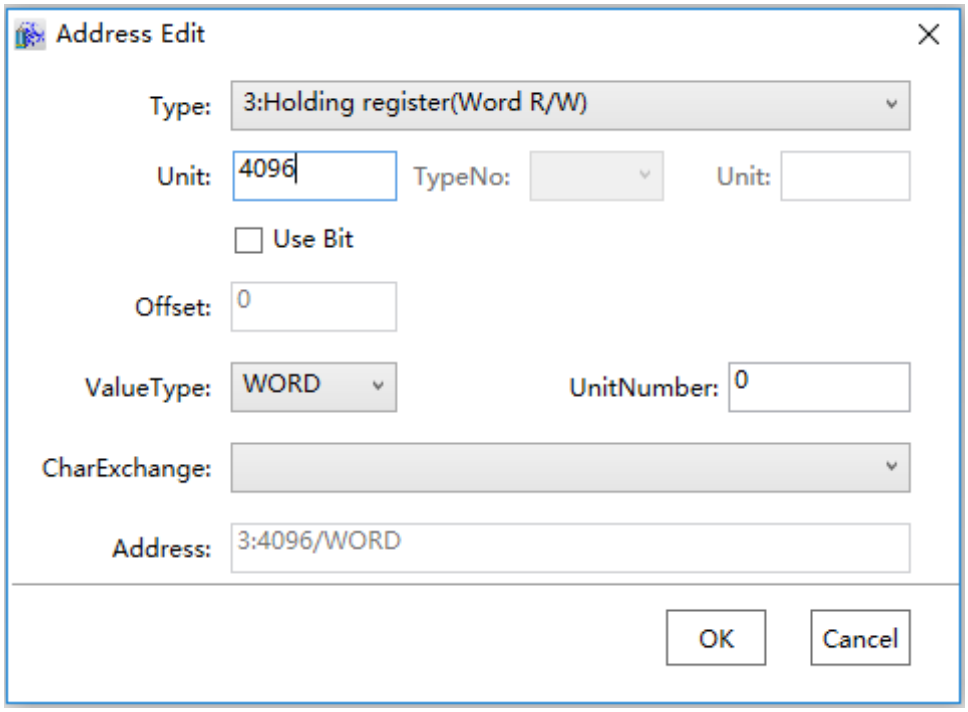
Value type	Description
BYTE	Byte
UBYTE	Unsigned byte
WORD	Word
UWORD	Unsigned word

DWORD	Double word
UDWORD	Unsigned double word
FLOAT	Single-precision floating point
DOUBLE	Double-precision floating point
STRING	String

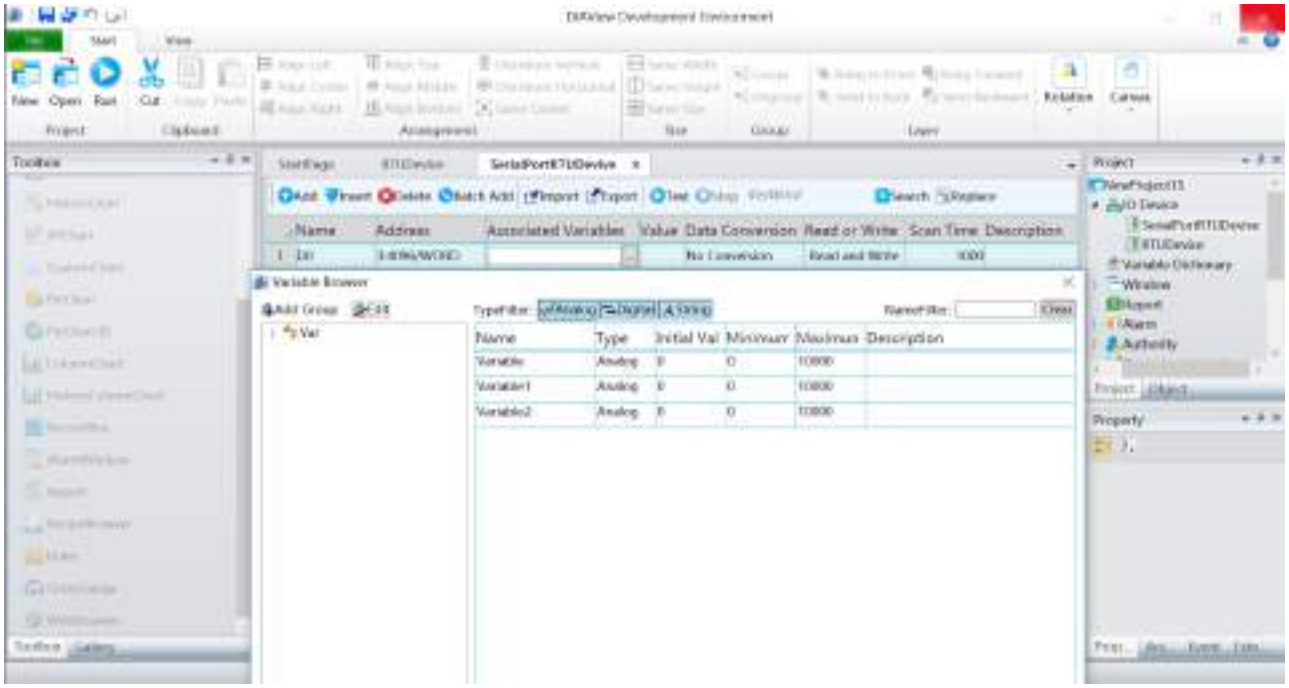
Character exchange:

Byte exchange	Name
No Swap (01)	No exchange
Swap BYTE (10)	Byte exchange (Word)

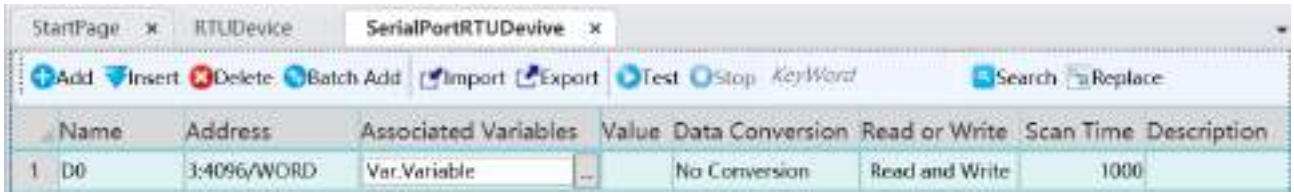
Step 4: Select register type: “3 R/W,3,16 (3: Holding register (R/W,3,16))”, unit number: 4096, keep the default value for other items, as shown in the figure below:



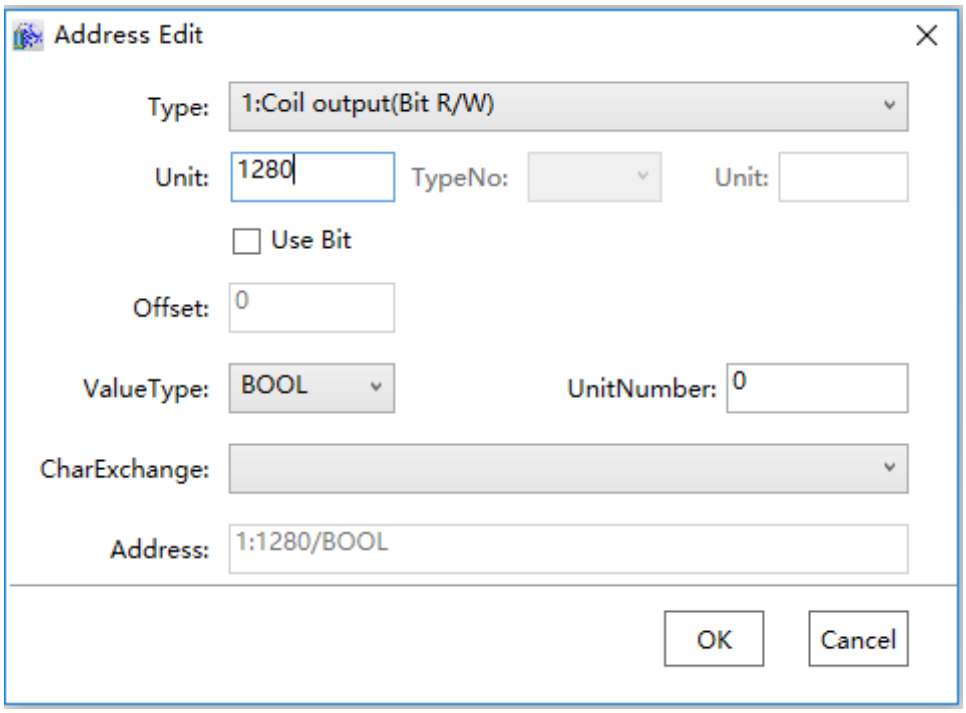
Step 5: Click the button in the "Variable" field to open the variable browser and select the “Analog value" type variables:



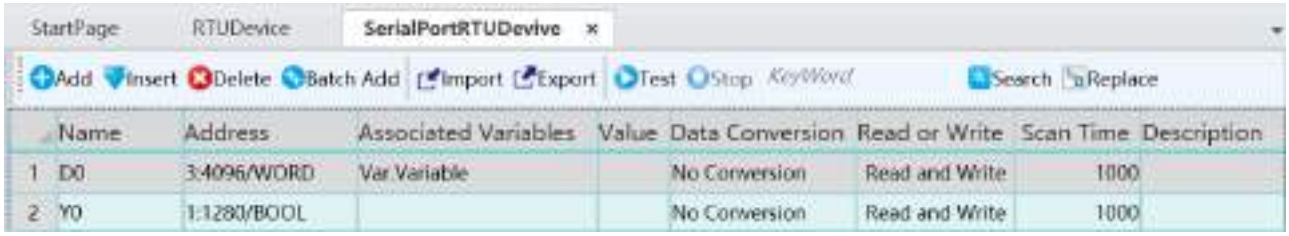
Step 6: Use preset values for the bits in other fields; this completes the creation of an IO variable D0:



Step 7: Create another IO variable Y0: The other steps are the same as creating D0, only its "Address" field parameters are as shown in the figure below:

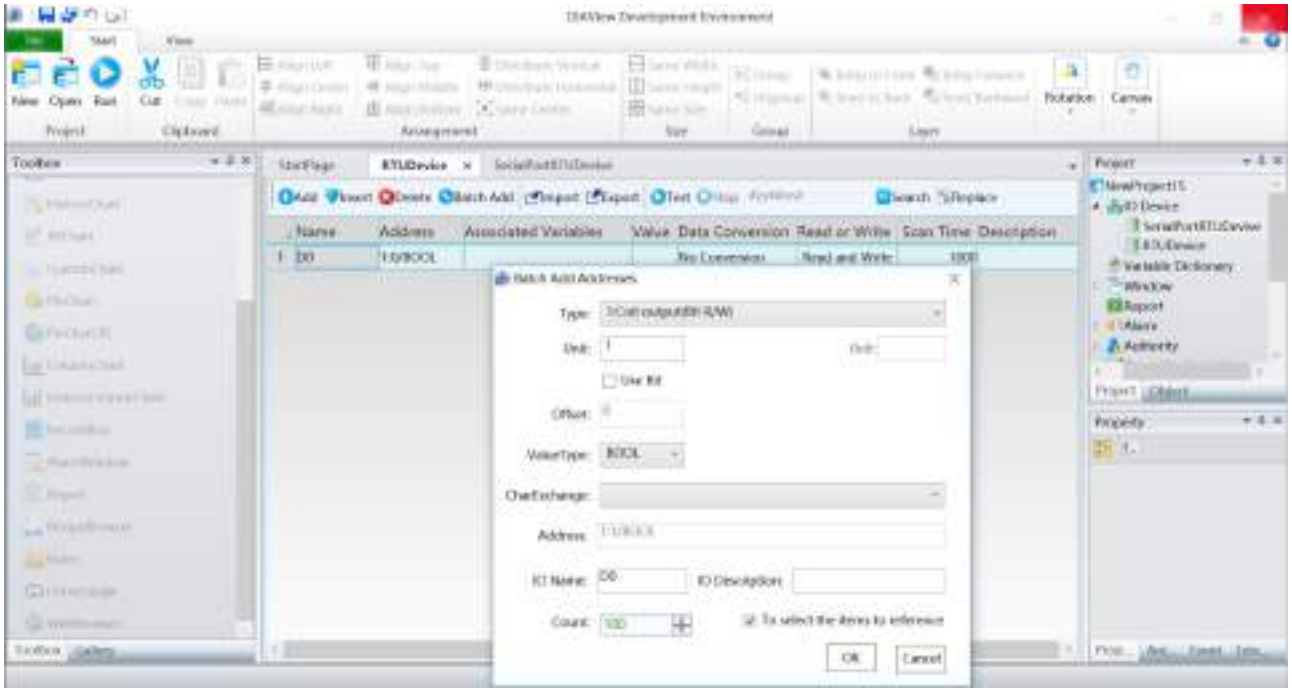


Step 8: Complete the creation of the two IO variables D0 and Y0, as shown in the figure below:



2.Creating IO variables in batches

Step 1: Click the “Batch Add " button on the IO variable configuration window toolbar of the “Serial port device”:



Meanings of configuration items:

Type: Sets the register type;

Use Bit: Whether to use a specified bit of the access communication address;

Offset: Relative to the size of the register's starting address;

Address: Sets the starting value of the address index for the batch creation of IO variables;

IO Name: The name of the IO variable; this name will be used as the basis with gradually increasing numbers-at-the-end during batch add;

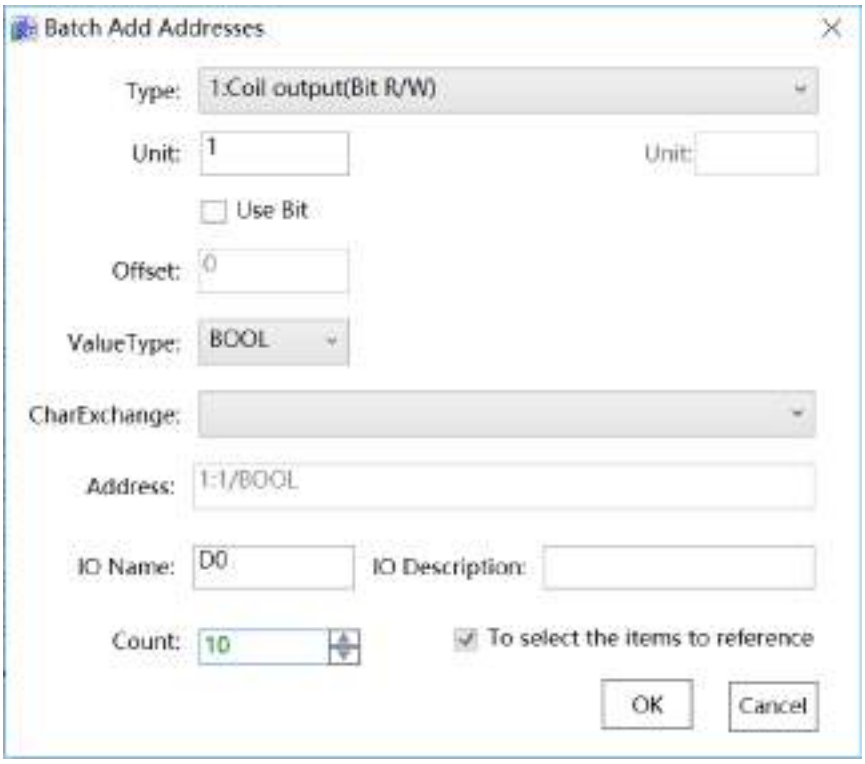
IO Description: Explanation information on the IO variable;

Count: The number for creating IO variables in batches;

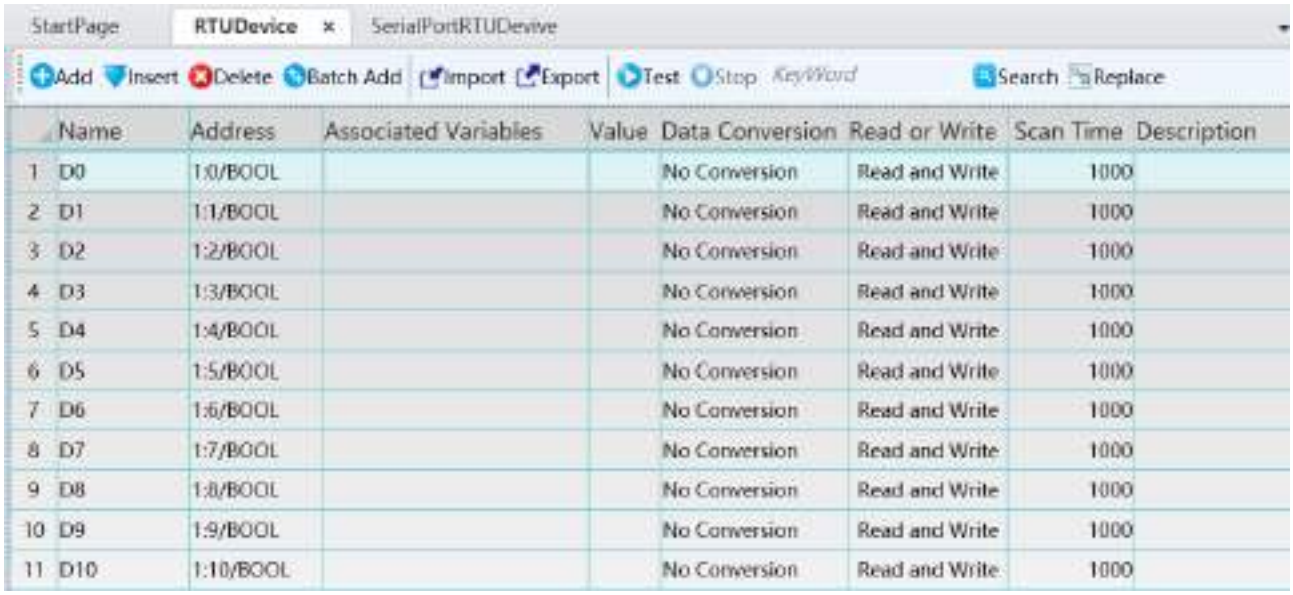
To selected the item to reference(not used): If selected, the IO name will use the selected IO variable name from the IO variable table as the basis and gradually increase; the register type will be the same as the selected IO variable type and the starting address will gradually increase from the selected item's address. If not selected, the IO name will gradually increase using the name preset by the system, the register type will be the preset item and the starting address is 0.

Not supported to batch the type of 'STRING'

Step 2: Set "count" as: 10, and other terms remain unchanged:



Step 3: Batch add 10 IO variables D1~D10, as shown in the figure below:



	Name	Address	Associated Variables	Value	Data Conversion	Read or Write	Scan Time	Description
1	D0	1:0/BOOL			No Conversion	Read and Write	1000	
2	D1	1:1/BOOL			No Conversion	Read and Write	1000	
3	D2	1:2/BOOL			No Conversion	Read and Write	1000	
4	D3	1:3/BOOL			No Conversion	Read and Write	1000	
5	D4	1:4/BOOL			No Conversion	Read and Write	1000	
6	D5	1:5/BOOL			No Conversion	Read and Write	1000	
7	D6	1:6/BOOL			No Conversion	Read and Write	1000	
8	D7	1:7/BOOL			No Conversion	Read and Write	1000	
9	D8	1:8/BOOL			No Conversion	Read and Write	1000	
10	D9	1:9/BOOL			No Conversion	Read and Write	1000	
11	D10	1:10/BOOL			No Conversion	Read and Write	1000	

Step 4: Associate the variables and set the bits in other fields; this completes the batch creation of IO variables.

3、Importing and exporting IO variables

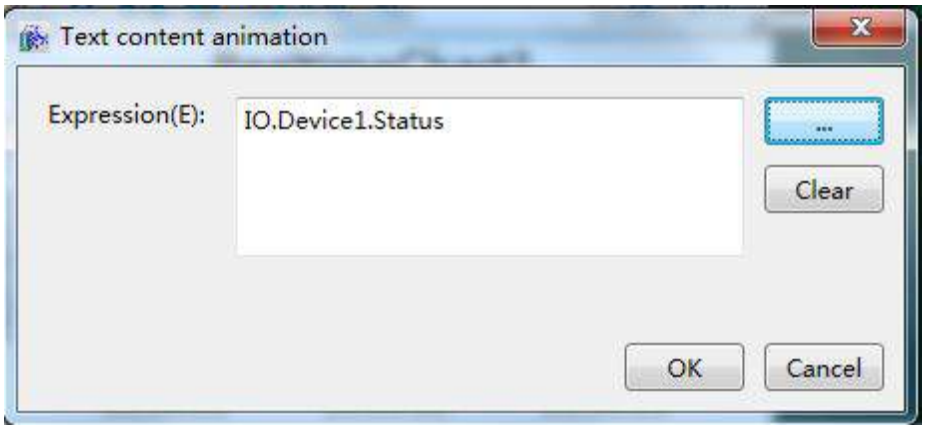
The IO variables in the DIAView software can be imported or exported into Excel; The excel file format must follow the rule of DIAView I/O variable table.

5.6 Communication status

There are three types of communication status: Connected, disconnected, stopped.

➢ **View device communication status:**

Draw a basic “text” graphic in any window of an existing project and configure “Expression(E)” in the Text animation window, as shown in the figure below:



The communication status of the “serial port device” can be checked once executed; please refer to the table blow:

Status	Flag bit status value
Communication normal	IO.serial port device.Status = 0
Communication disconnected	IO.serial port device.Status = 2
Communication stopped	IO.serial port device.Status =2

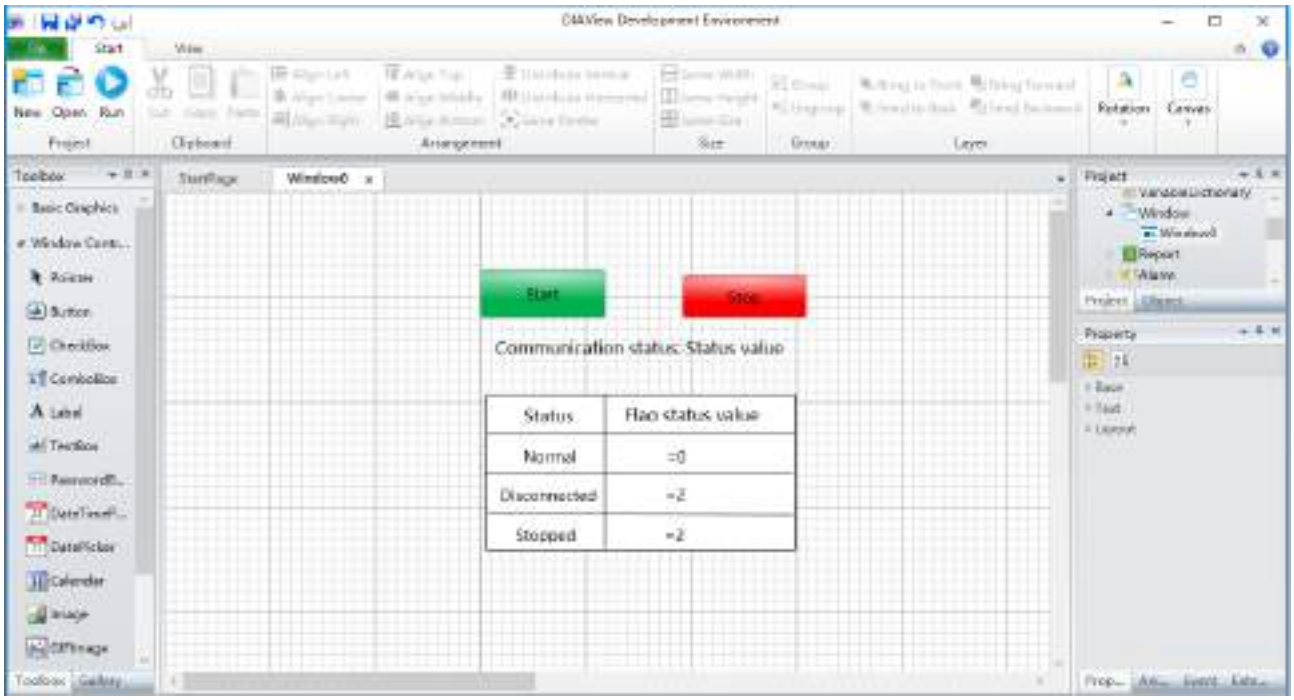
5.7 Communication control

There are three types of communication status: connected, disconnected and stopped. In real applications, the communication status returned by the DIAView software can be used to determine whether the connected device communication is normal, and dynamically control to start or stop its communication when necessary.

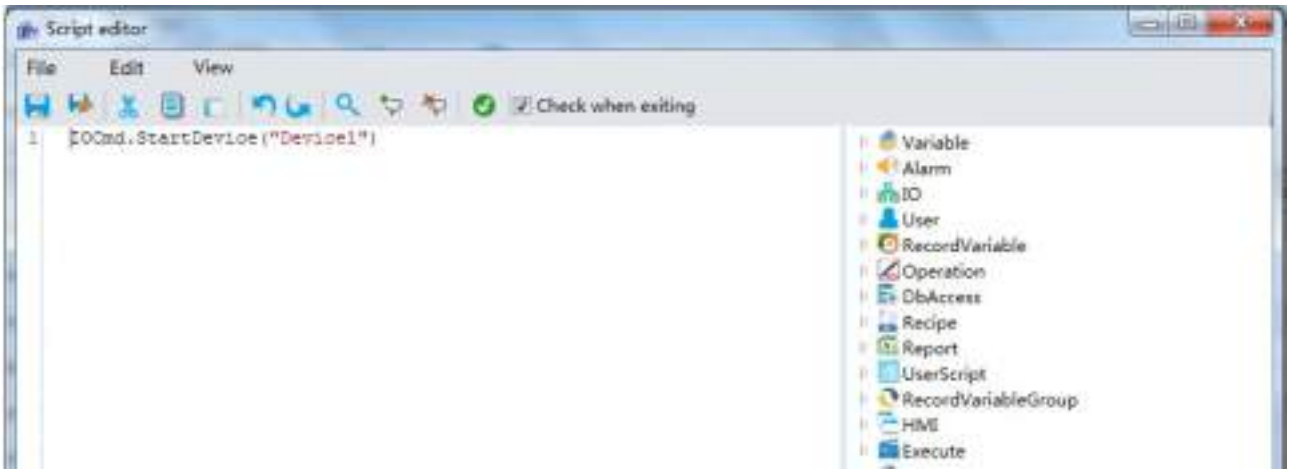
Device communication status flag can be changed to perform control to the communication status,, that is to change the property value of the device Status ， please refer to the example below:

- **Control device communication status:**

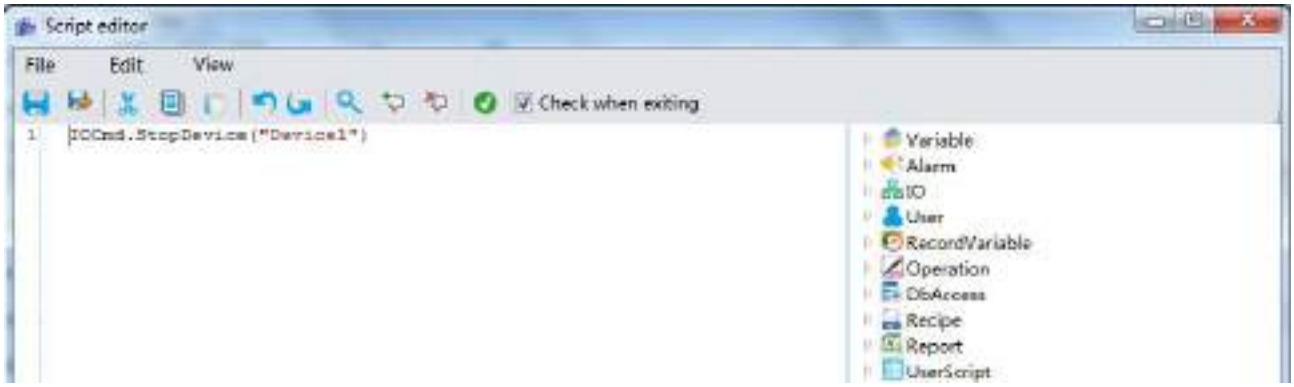
Step 1: Draw two buttons and two texts in the window of an existing project, as shown in the figure below:



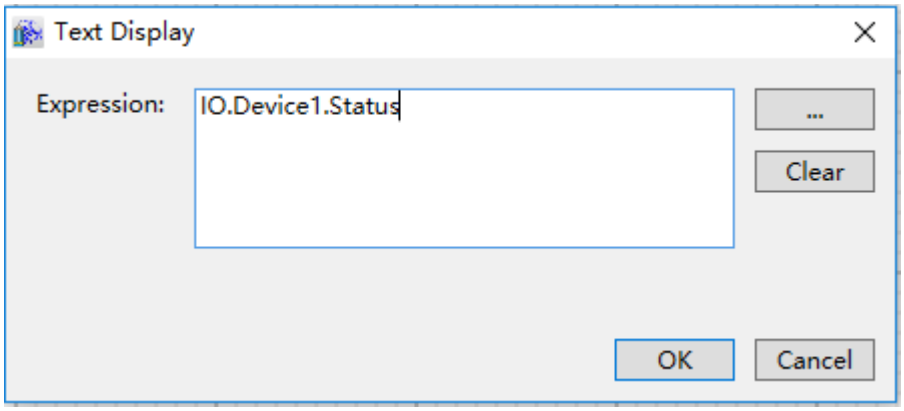
Step 2: Configure a “Left-click” event for the “Start button”, as shown in the figure below:



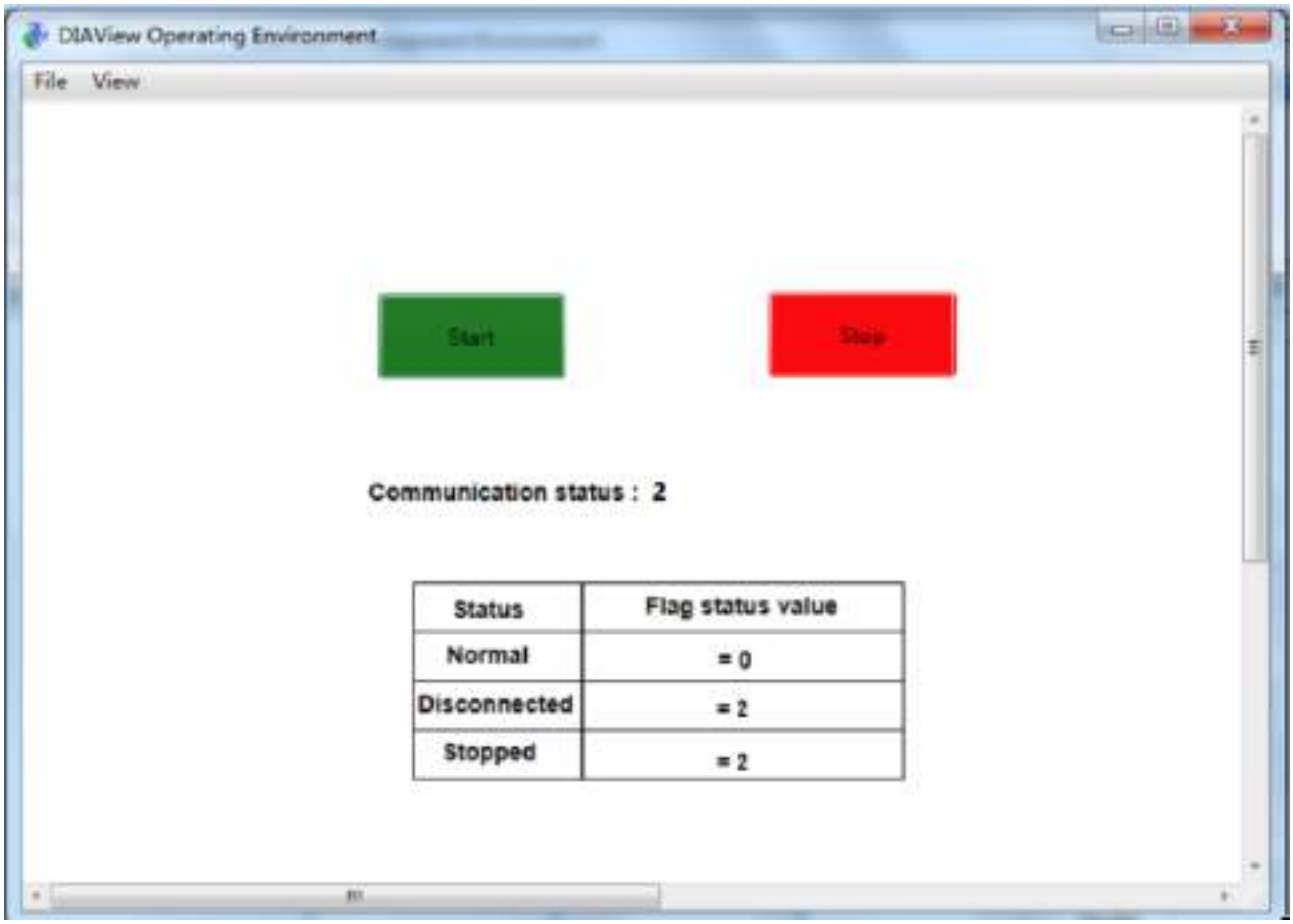
Step 3: Configure a “Left-click” event for the “Stop” button, as shown in the figure below:



Step 4: Configure a “Expression(E)” for the “Status value” text, as shown in the figure below:

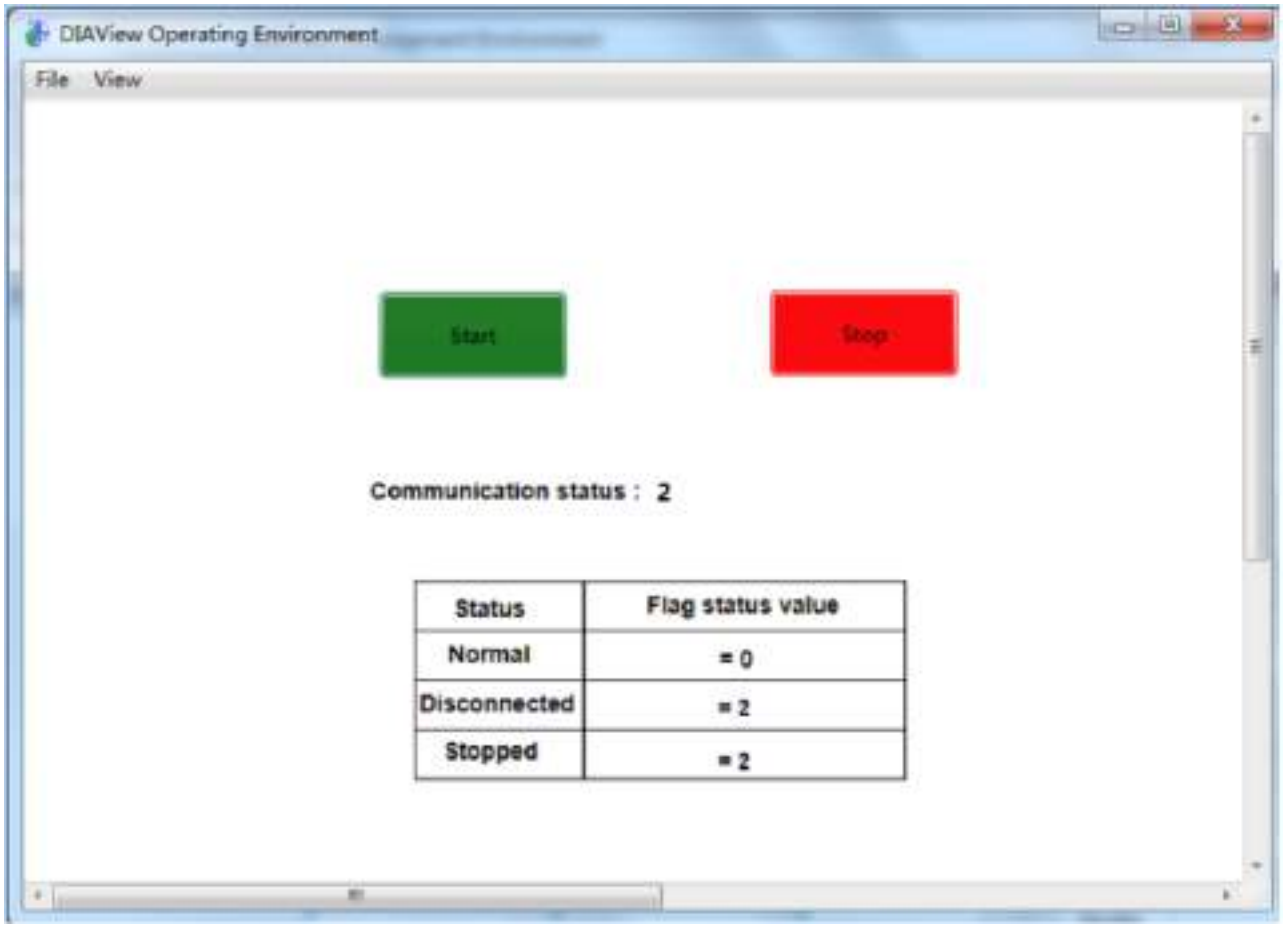


Step 5: Once executed the initial communication status of the “Device0” is as follows:



Step 6: The communication status is as follows after connects to the machine and reconnection mechanism:

Step 7: The communication status of “Device0” is as follows after clicking the “Stop” button:



Step 8: The communication status of “Device0” is as follows after clicking the “Start” button:

5.8 PLC

5.8.1 Modicon

Modicon is the sub-brand of Schneider Electric,its Chinese name is "Mo Di Kang".

DIAView software supports the communication of Modicon PLC,the protocols include Modicon RTU,Modicon ASCII,Modicon TCP.

Supporting communication network interface: Ethernet and serial.

✧ **Noted:** Modicon address description,as shown in the table below:

Register	Range	Default type	Description
1	1~65536	Bool	Coil output
2	1~65536	Bool	Digital input

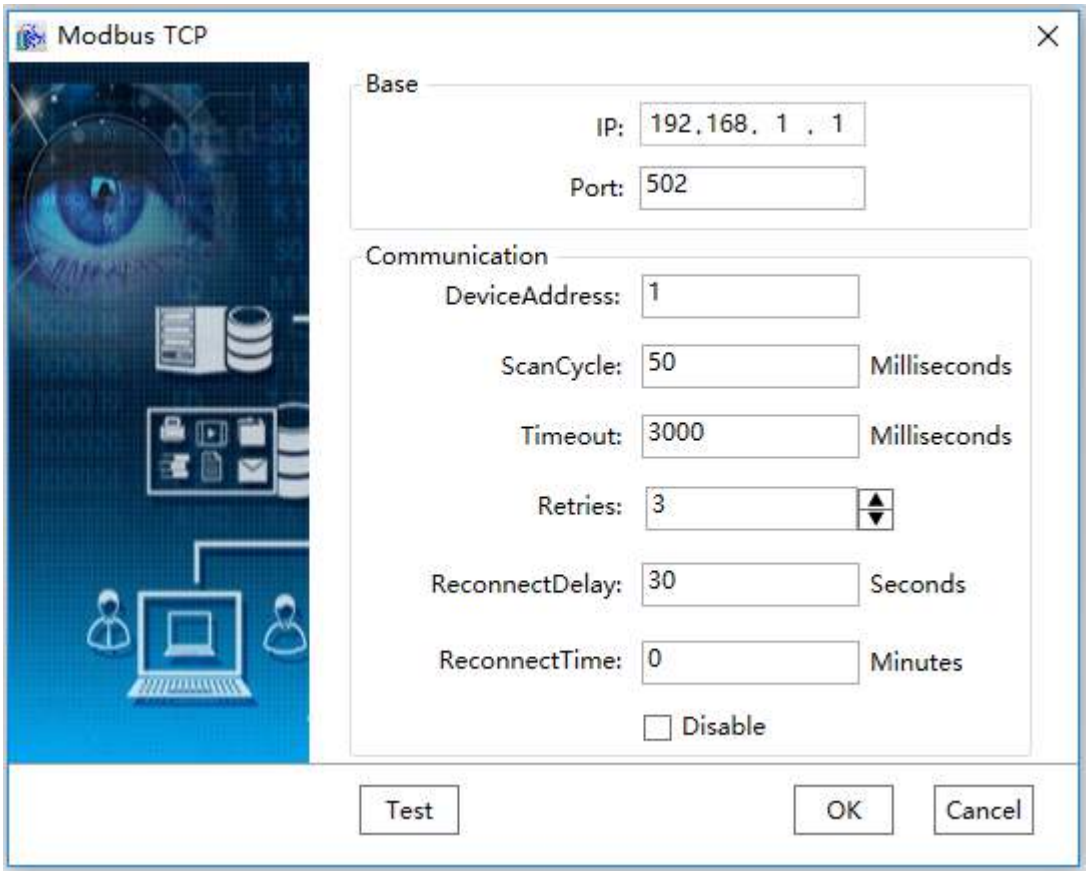
3	1~65536	Word	Holding register
4	1~65536	Word	Analog output
5	1~65536	Word	Extended output register
6	1~65536	Word	Extended input register

5.8.1.1 Modbus TCP

DIAView can communicate with modicon PLC by computer Ethernet port Modbus protocol.

Example 1:

Please refer to Delta PLC TCP communication configuration

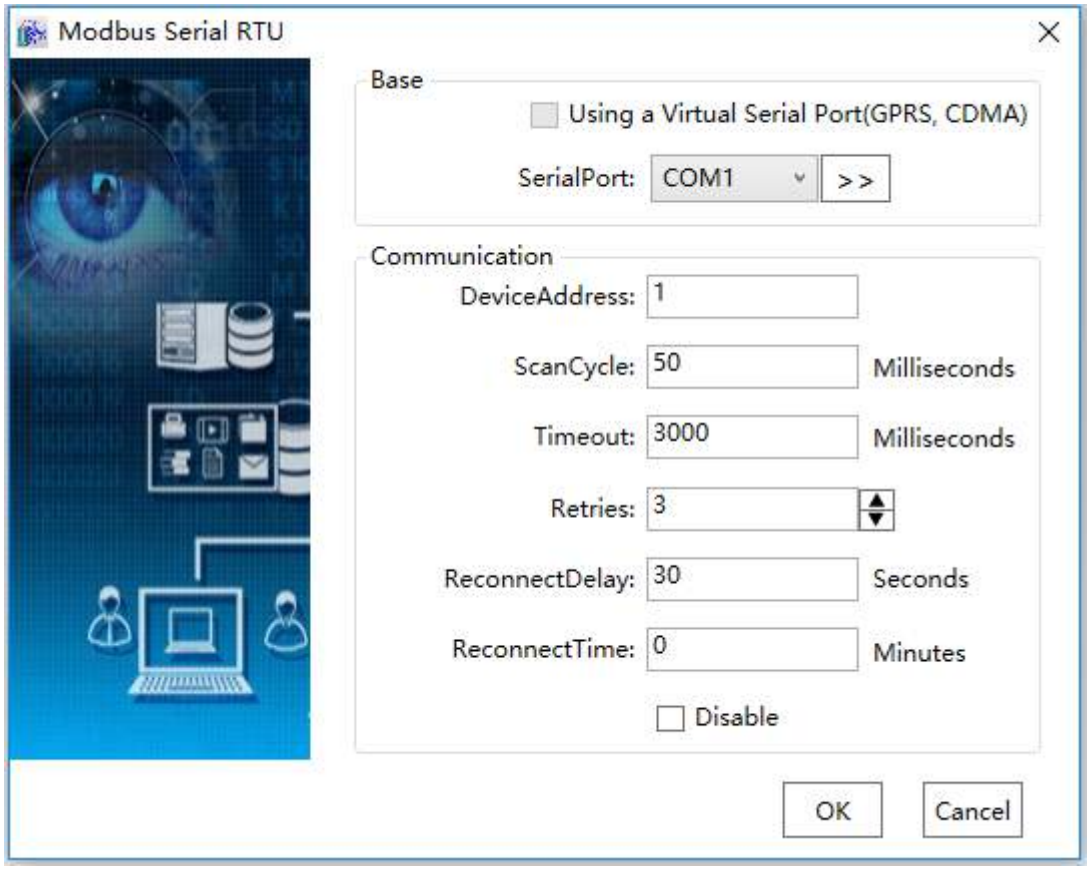


5.8.1.2 Modbus Serial RTU

DIAView can communicate with modicon PLC by computer serial port Modbus protocol, with the way of RTU encoding.

Example 1:

Please refer to the Delta PLC serial port communication configuration

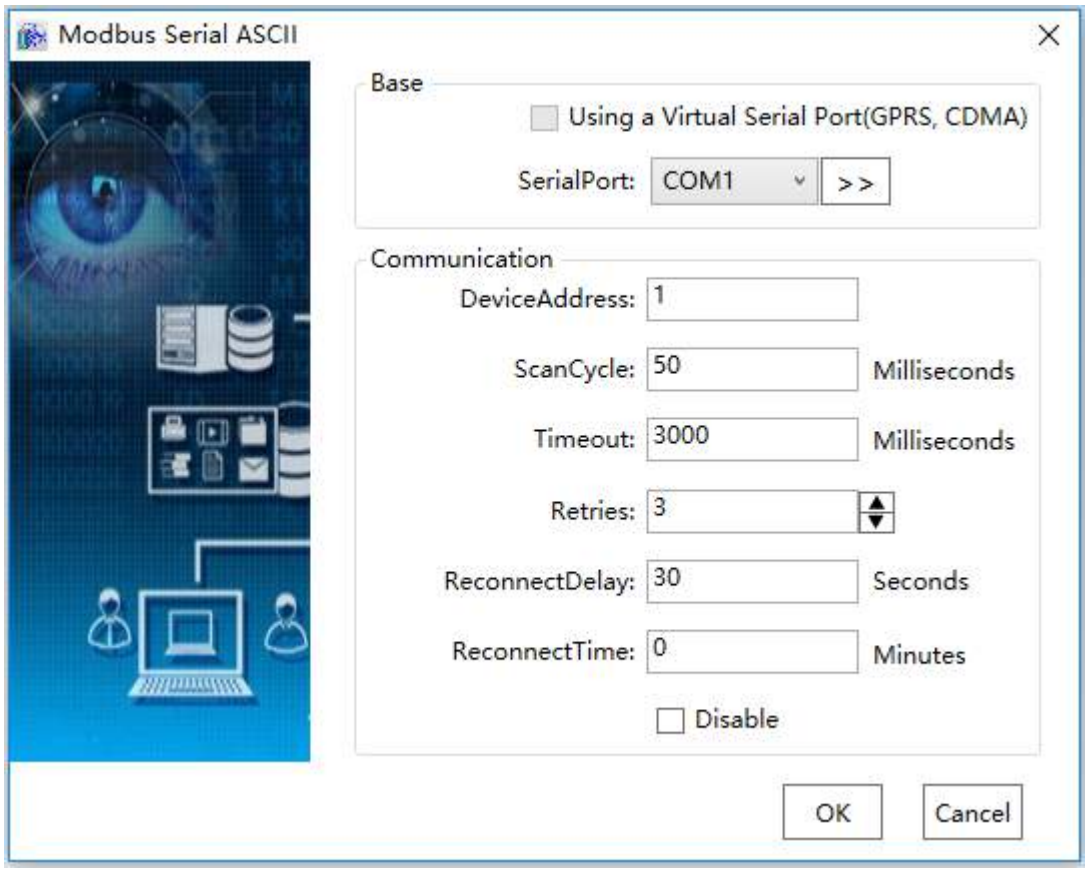


5.8.1.3 Modbus Serial ASCII

DIAView can communicate with modicon PLC by computer serial port Modbus protocol, with the way of ASCII encoding.

Example 1:

Please refer to Delta PLC serial port communication configuration.



5.8.2 Omron

DIAView software supports the communication of Omron PLC,the protocols include FINS TCP, FINS ASCII, HostLink ASCII.

Supporting device: CS/CJ(CP) series,CV series(Untested).

Supporting command format: C-mode command and FINS command.

Supporting communication network interface: Ethernet and serial.

5.8.2.1 FINS TCP

The equipments that current test environment requires are Omron SYSMAC CJ2M CPU32

and Omron SYSMAC CP1H.

The example that DIAView software communicates with equipments by Ethernet is as follows, FINS TCP and FINS ASCII (serial) configuration are similar, and the device address is the same:

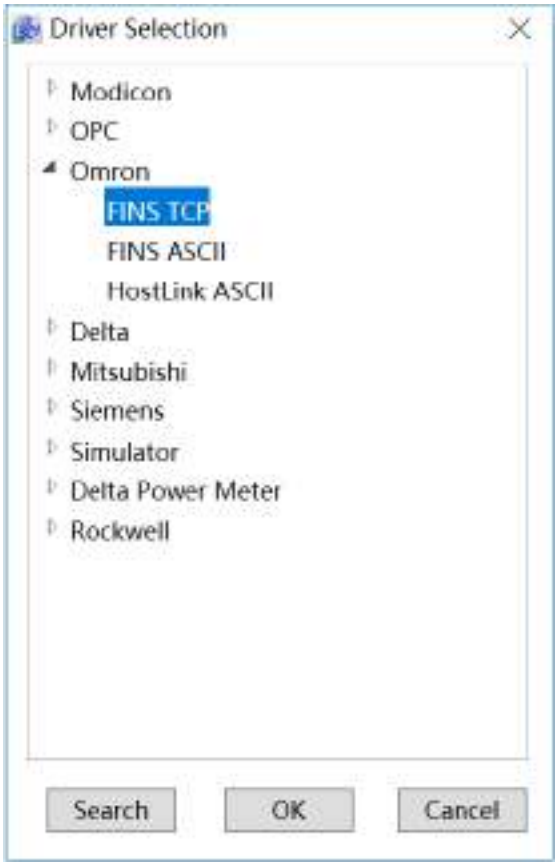
Example 1:

- **Creating communication between DIAView software and Omron CJ2M through Ethernet**

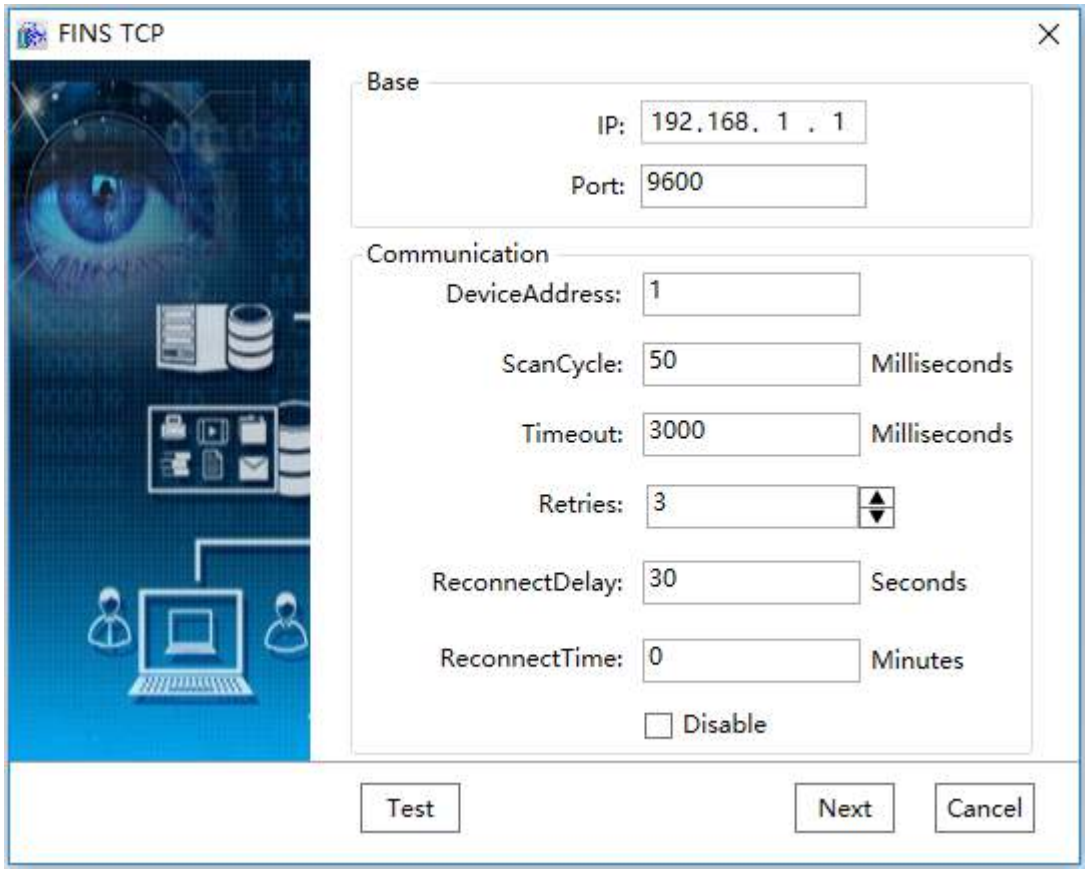
Step 1: Create the following hardware framework ,Omron programming software set CJ2M address as"192.168.1.5", the computer IP address is "192.168.1.200" (keep the PLC and the computer in the same LAN):

Step 2: In the project management area, right click on "IODevice "node and select "New Device".

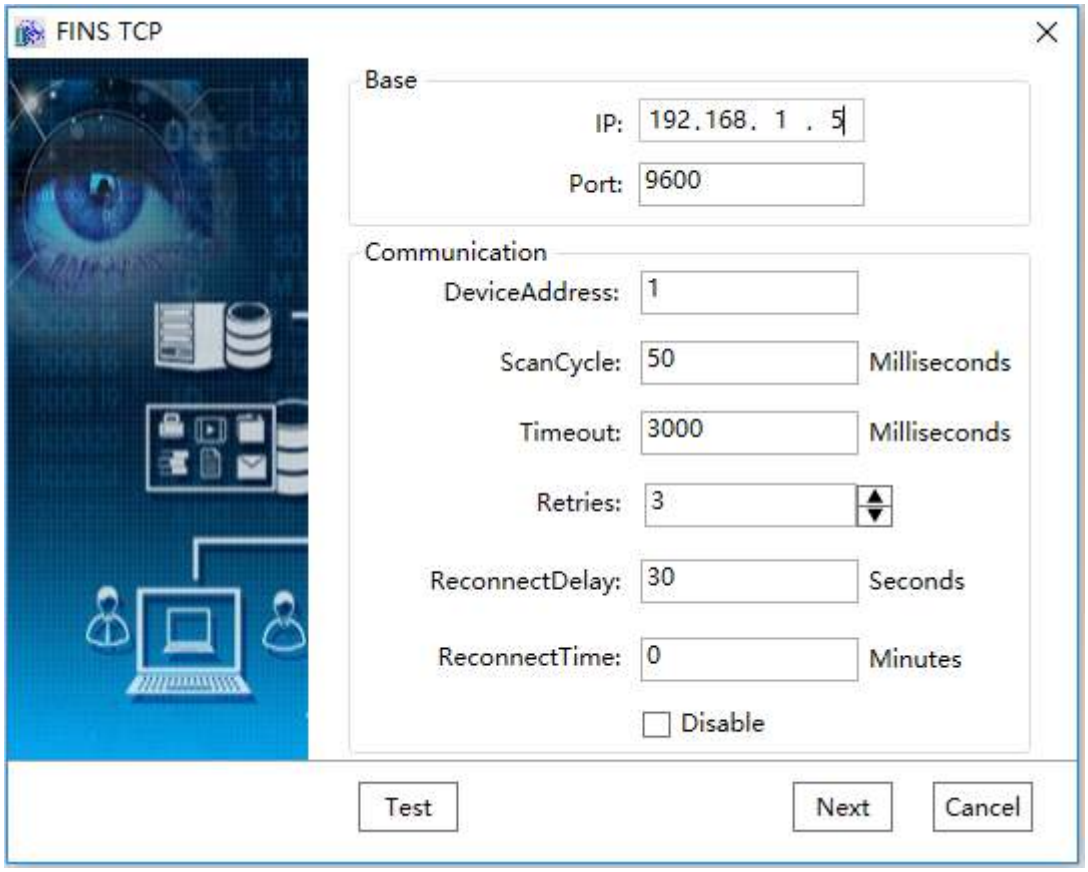
Step 3: Select "Omron" →"FINS TCP" in the driver selection window, as shown in the figure below:



Step 4: Configure the Omron FINS TCP communication parameters, as shown in the figure below:

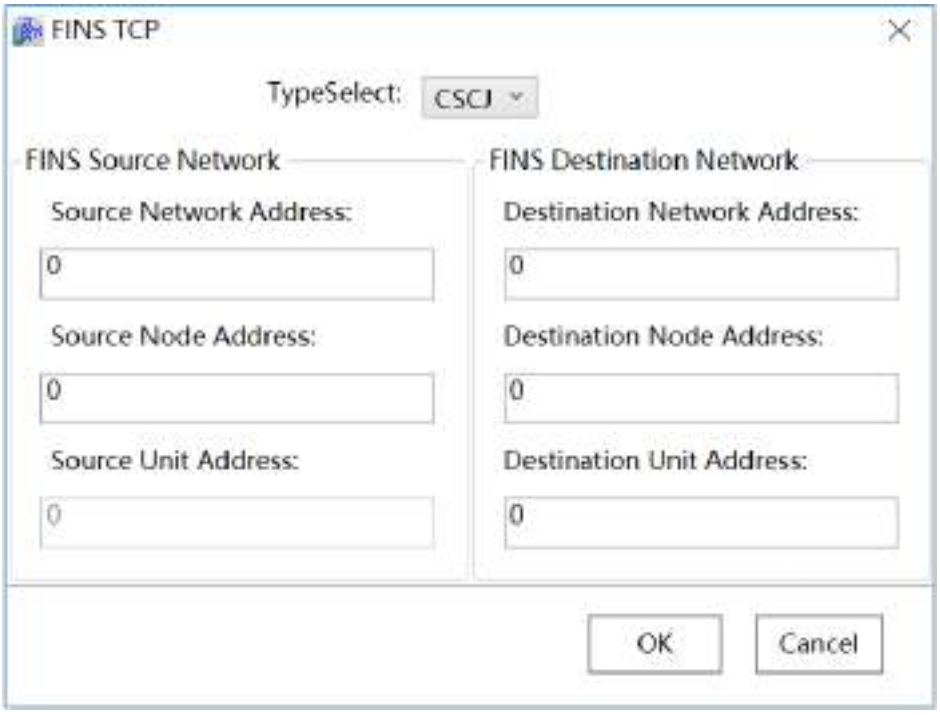


Step 5: Set the "IP" in the "Base" as 192.168.1.5, set the port number as 9600, as shown in the figure below:

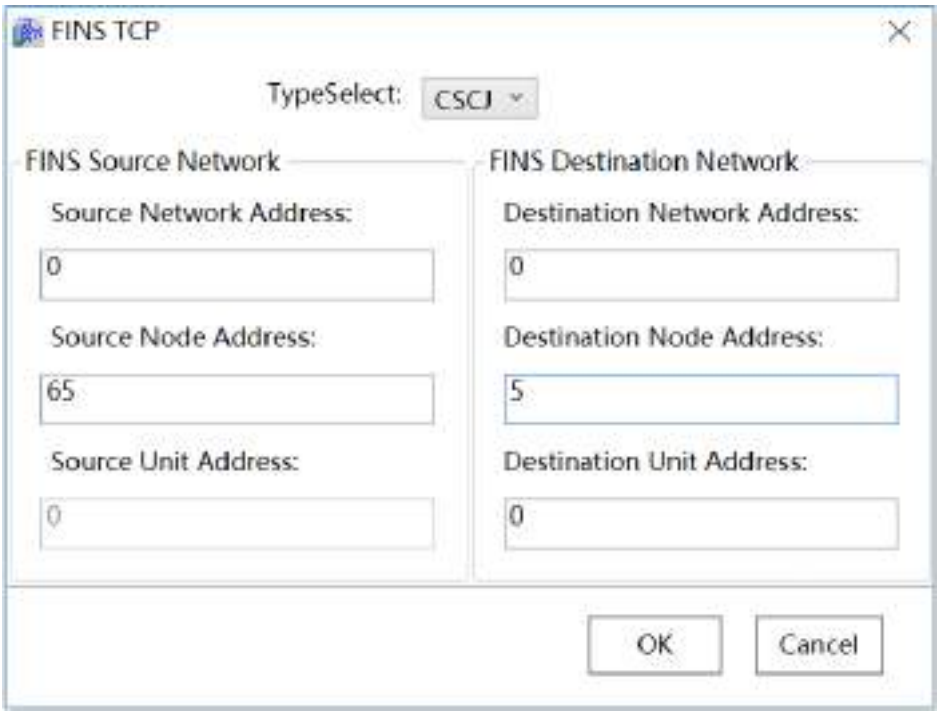


Step 6: Other options can be the default value, click "Test" button to test whether the connection is successful:

Step 7: Click “Next” button to go to the Omron FINS specific configuration interface, only including 2 models: CS/CJ (CP) and CV, as shown in the figure below:



Source node: 65, destination node: 5, other unchanged, as shown in the figure below:



Network Address:

If there is only one local network, then the network numbers are all set to 0, representing only one network. If there are multiple networks, in order to avoid conflict, then you must specify different network number at all levels, the range is 1-127.

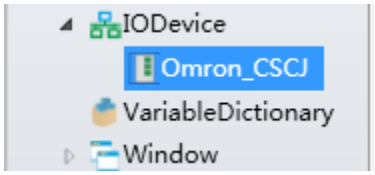
Node Address:

In the same level of the network, the node number of each connection node needs to be set as a different number, a node corresponds to a PLC. If it is an Ethernet network, the node number is the last field of the IP address generally.

Unit Address:

In the same PLC, each module unit number is different from each other, CPU unit number has always been 0, the rest of their own settings.

Step 8: Click the "OK" button when all parameters have been configured and the device with default name will appear under “IODevice" node of the project tree directory, and rename it as "Omron_CSCJ".



➢ **Noted:** FINS(CS/CJ):Omron address description is as shown in the table below:

Register	Range	Default type	Description
CIO_B	0~614315	Bool	/
CIO_W	0~6143	Word	/
WR_B	0~51115	Bool	/
WR_W	0~511	Word	/
HR_B	0~51115	Bool	/
HR_W	0~511	Word	/
AR_B	0~95915	Bool	00000~44715(only read)
			44800~95915(read/write)
AR_W	0~959	Word	000~447(only read)
			448~959(read/write)
DM_B	0~3276715	Bool	/

DM_W	0~32767	Word	/
EM_B	0~32767	Bool	EM bank 0 to bank F: E0_0000000 to 3276715 to EF_0000000 to 3276715 like : EM_B0:3276715
			EM bank 10 to bank 18: E0_0000000 to 3276715 to EF_0000000 to 3276715
			EM current bank: E0000000 to E3276715
EM_W	0~32767	Word	EM bank 0 to bank F: E0_00000 to 32767 to EF_00000 to 32767,like: EM_B0:32767
			EM bank 10 to bank 18: E0_00000 to 32767 to E18_00000 to 32767
			EM current bank: E00000 to E32767
TIM_PV	0~4095	Word	/
CNT_PV	0~4095	Word	/
IR_W	0~15	Word	/
DR_W	0~15	Word	/

5.8.2.2 FINS ASCII

Please refer to“FINS TCP” and other serial configuration.

The example that DIAView software communicates with equipments by serial port is as follows,using FA command:

Creating communication between DIALink and Omron CJ2M through serial port:

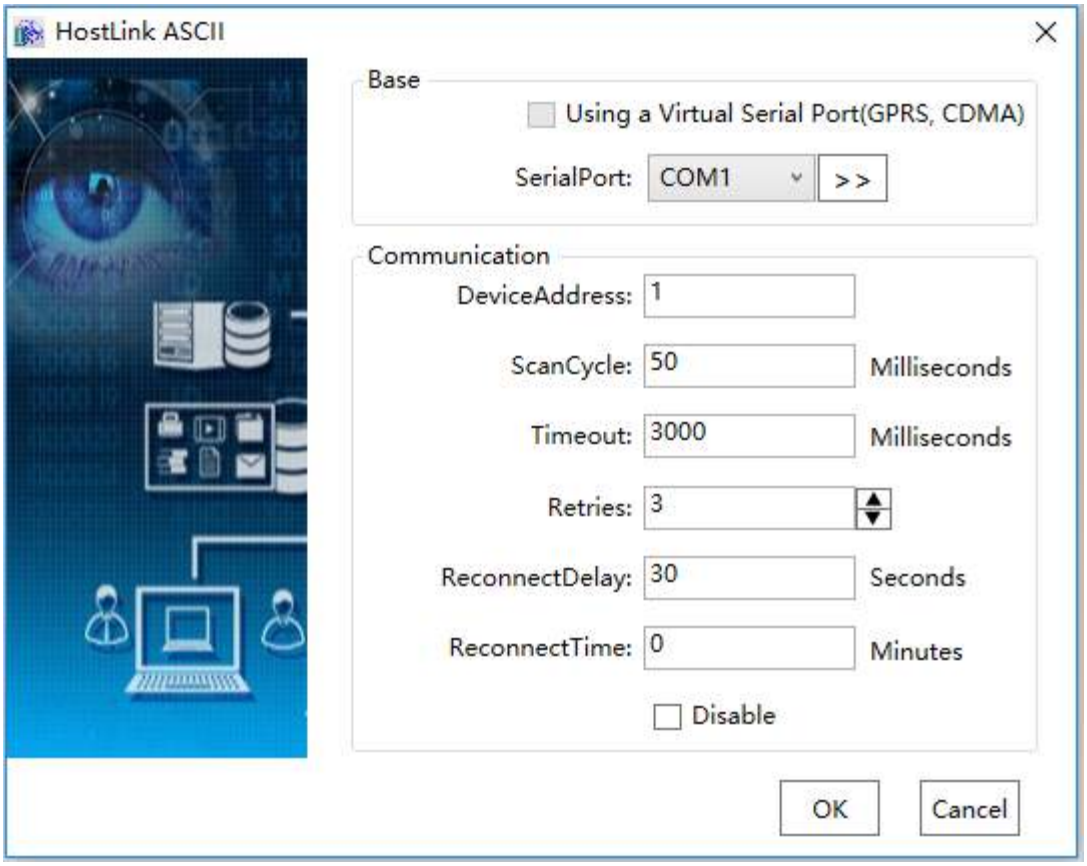
Example 1: “Omron CJ2M, HostLink ASCII”driver

Step 1: Create the following hardware framework, set CJ2M communication port parameters as “9600, 8,E,1,ASCII”, set Station number as 1, as shown in the figure below:

Step 2: In the project management area, right click on “IODevice “node and select "New Device":

Step 3: Select "Omron" → “HostLink ASCII” in the driver selection window, as shown in the figure below:

Step 4: Configure the Omron HostLink ASCII communication parameters, as shown in the figure below:



➢ **Noted:** HostLink:Omron address description, as shown in the table below:

Register	Range	Default type	Description
CIO	0~6143	Word	/
LR	0~199	Word	/
HR	0~511	Word	Holding relay
DM	0~9999	Word	Data memory
EM	0~9999	Word	/
TC_PV	0~4095	Word	(0~2047) Counter /(2048~4095) Counter
TC_STATUS	0~4095	Bool	Counter/Counter status flag
AR	0~959	Word	Auxiliary relay

5.8.2.3 HostLink ASCII

The example that DIAView software communicates with equipments by serial is as follows,using C-mode command:

➢ **Creating communication between DIALink and Omron CJ2M through serial port:**

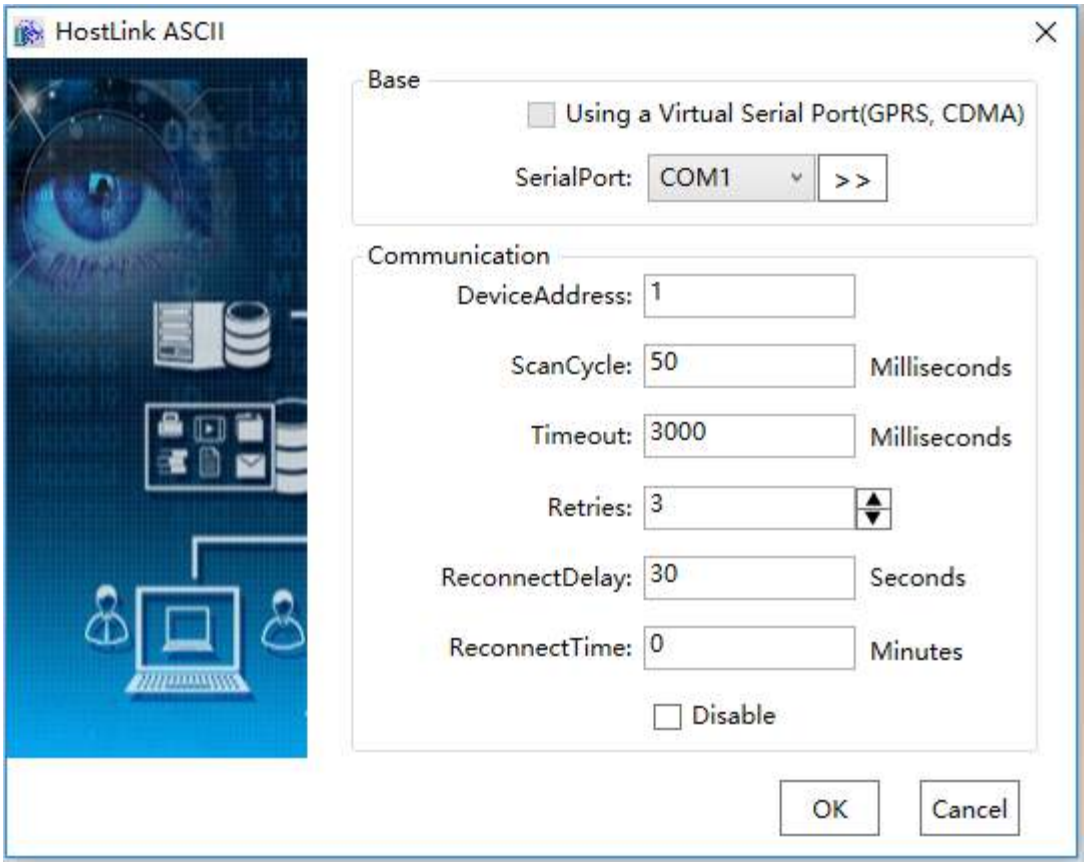
Example 1: "Omron CJ2M, HostLink ASCII"driver

Step 1: Create the following hardware framework, set CJ2M communication port parameters as "9600, 8,E,1,ASCII", set Station number as 1, as shown in the figure below:

Step 2: In the project management area, right click on “IODevice “node and select "New Device":

Step 3: Select "Omron" → “HostLink ASCII” in the driver selection window, as shown in the figure below:

Step 4: Configure the Omron HostLink ASCII communication parameters, as shown in the figure below:



➢ **Noted:** HostLink:Omron address description, as shown in the table below:

Register	Range	Default type	Description
CIO	0~6143	Word	/
LR	0~199	Word	/
HR	0~511	Word	Holding relay
DM	0~9999	Word	Data memory
EM	0~9999	Word	/
TC_PV	0~4095	Word	(0~2047) Counter /(2048~4095) Counter
TC_STATUS	0~4095	Bool	Counter/Counter status flag
AR	0~959	Word	Auxiliary relay

5.8.3 Delta

Delta PLC includes DVP series and AH series.

✧**Noted1:**Delta Modbus DVP address description, as shown in the table below:

Register	Range	Default type	Description
CB	0~255	Bool	Counter
CW	0~199	word	Counter
CDW	200~255	Double word	Counter
D	0~9999	Word	Data register
SED	10000~11999	Word	Data register
S	0~1023	Bool	Stepping relay
TB	0~255	Bool	Timer

TW	0~255	Word	Timer
X	0~377 (OCT)	Bool	External input relay
Y	0~377 (OCT)	Bool	External output relay
M	0~4095	Bool	Auxiliary relay

✧**Noted2:**Delta Modbus AS300 address description, as shown in the table below:

Register	Range	Default type	Description
CB	0~511	Bool	Counter
CW	0~511	Word	Counter
HCB	0~255	Bool	High-speed counter
HCDW	0~255	Double word	32 bit counter HC
D	0~ 29999	Word	Data register
S	0~2047	Bool	Stepping relay
SM	0~4095	Bool	Special auxiliary relay
SR	0~2047	Word	Special data register
TB	0~511	Bool	Timer
TW	0~511	Word	Timer
XB	0.0~63.15	Bool	Input relay
XW	0~63	Word	Input relay
YB	0.0~63.15	Bool	Output relay
YW	0~63	Word	Output relay
M	0~8191	Bool	Auxiliary relay
E	0~9	Word	Indirect specified regiser

✧**Noted3:**Delta Modbus AH address descirption, as shown in the table below:

Register	Range	Default type	Description
CB	0~2047	Bool	Counter
CW	0~2047	Word	Counter
HCB	0~63	Bool	High-speed counter
HCDW	0~63	Double word	32 bit counter HC
DB	0~ 65535	Bool	Data register
D	0~ 65535	Word	Data register
S	0~1023	Bool	Stepping relay
SM	0~2047	Bool	Special auxiliary relay
SR	0~2047	Word	Special data register
TB	0~2047	Bool	Timer
TW	0~2047	Word	Timer
XB	0.0~511.15	Bool	Input relay
XW	0~511	Word	Input relay
YB	0.0~511.15	Bool	Output relay
YW	0~511	Word	Output relay
M	0~8191	Bool	Auxiliary relay
E	0~31	Word	Indirect specified regiser

5.8.3.1 Delta AS300 TCP

The example that DIAView software communicates with equipments by Ethernet please refer to "5.8.3.5 DVP Modbus TCP".

5.8.3.2 Delta AS300 RTU & ASCII

The example that DIAView software communicates with equipments by serial port please refer to "5.8.3.6 DVP Modbus Serial RTU & ASCII".

5.8.3.3 AH Modbus TCP

The example that DIAView software communicates with equipments by Ethernet please refer to "5.8.3.5 DVP Modbus TCP".

5.8.3.4 AH Modbus Serial RTU & ASCII

The example that DIAView software communicates with equipments by serial port please refer to "5.8.3.6 DVP Modbus Serial RTU & ASCII".

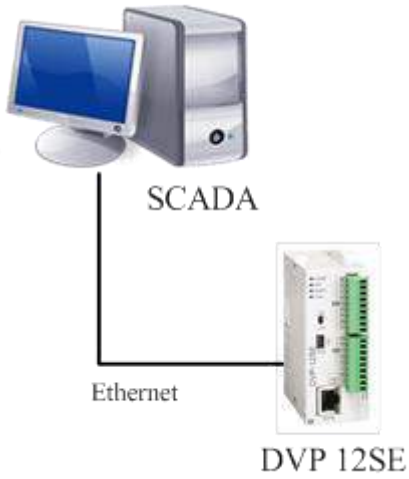
5.8.3.5 DVP Modbus TCP

DIAView software supports the communication of the equipment based on Modbus communication protocol standard by Ethernet (TCPIP).

The example that DIAView software communicates with equipments by Ethernet is as follows:

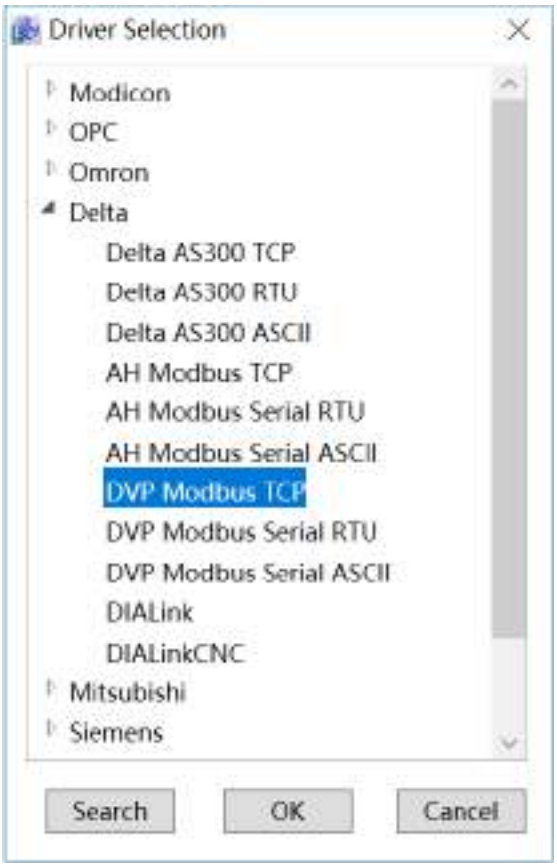
➢**Creating communication between DIAView software and Delta PLC—DVP 12SE through Ethernet:**

Step 1: Set up the hardware framework shown as the figure below, set 12SE IP address : “192.168.1.10”, the computer IP address :“192.168.1.200” (keep PLC and the computer in the same LAN).

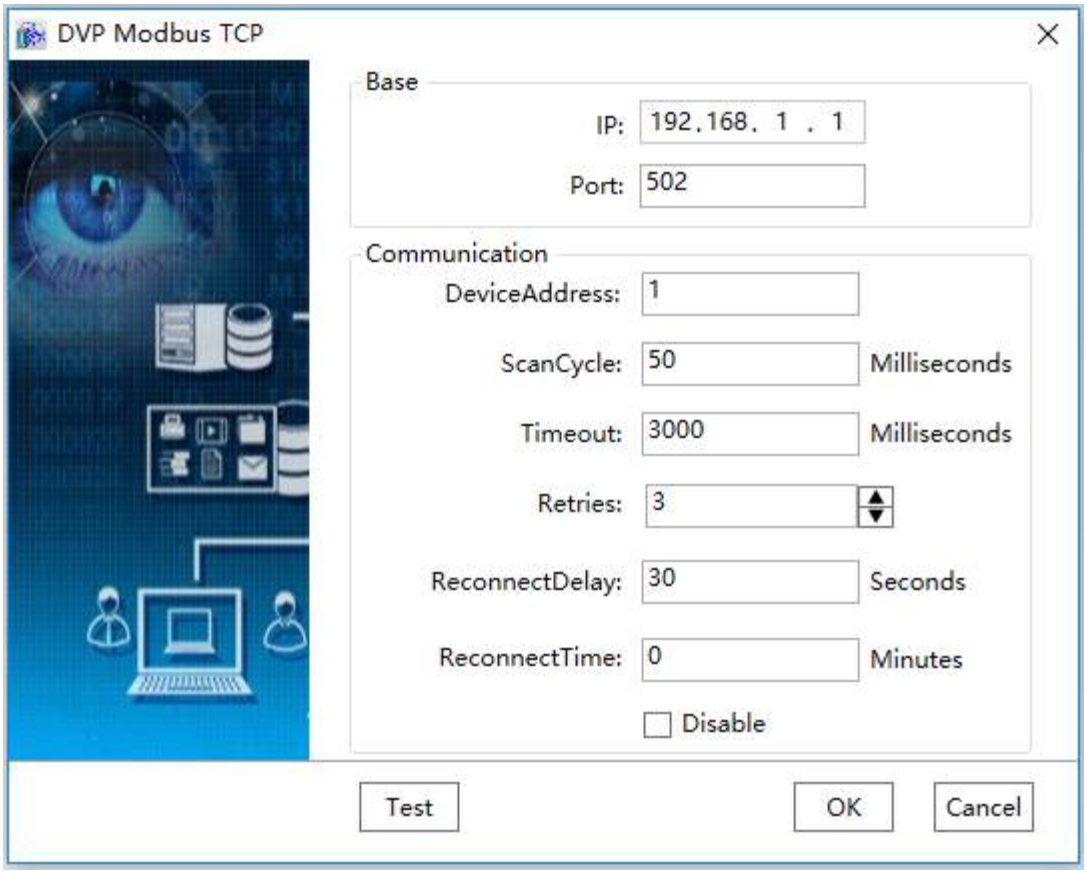


Step 2: Right-click on the “IODevice” node in the project management area and select” New Device”; the menu selection in the figure below will appear:

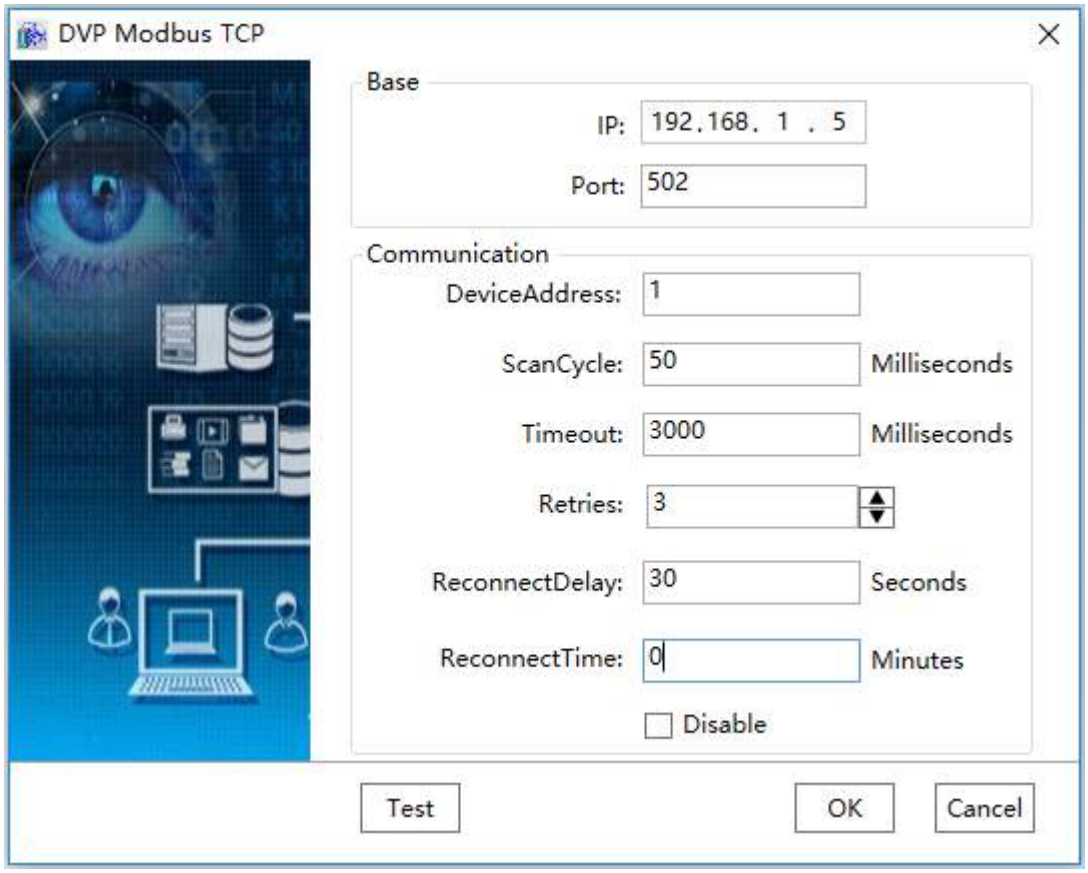
Step 3: Select “Delta”→ "DVP Modbus TCP” in the “Driver Selection” window, as shown in the figure below:



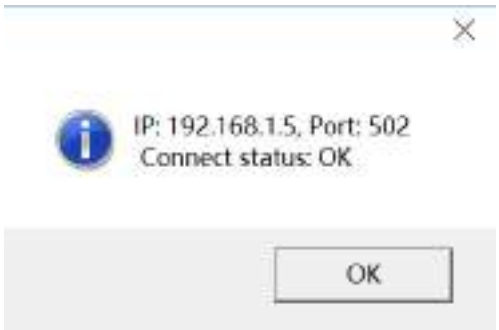
Step 4: Configure the Delta DVP TCP communication parameters, as shown in the figure below:



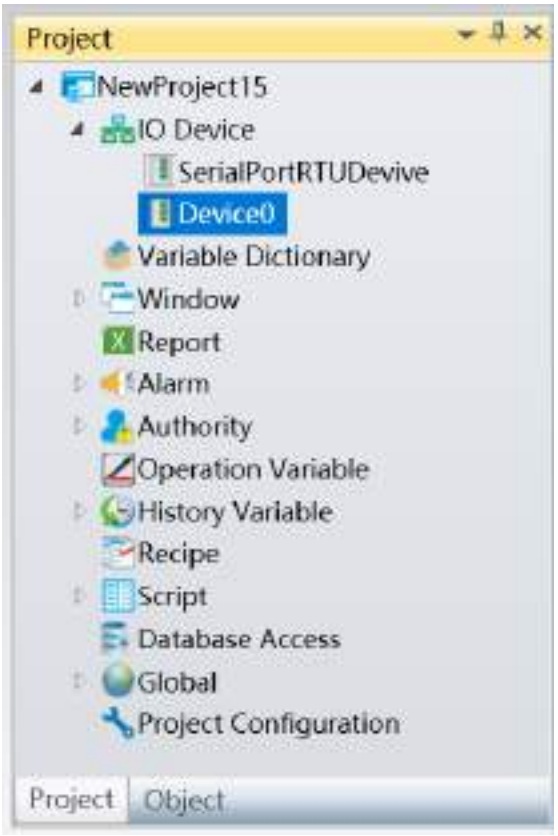
Step 5: Set the “IP” in the “Base” configuration as: 192.168.1.5 and the “Port” as: 502, as shown in the figure below:



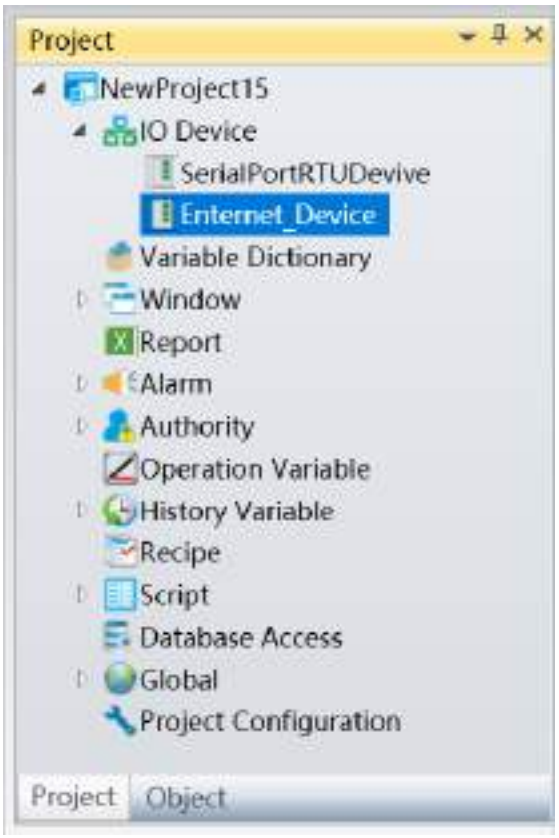
Step 6: Click the “Test” button to test whether the connection is successful, as shown in the figure below:



Step 7: click the “OK” button when all parameters have been configured and the device with the default name “Device0” will appear under the “IODevice” node, as shown in the figure below:



Step 8: Rename the newly-built IO communication device as "Ethernet_Device" and complete Ethernet driver communication configuration, as shown in the figure below:



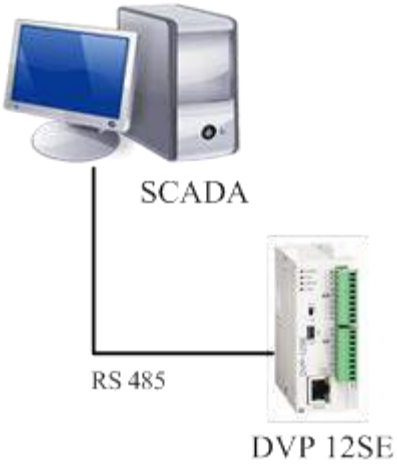
5.8.3.6 DVP Modbus Serial RTU & ASCII

The example that DIAView software communicates with equipments by serial port is as follows:

➢ **Creating communication between DIAView software and Delta PLC-DVP 12SE by serial port:**

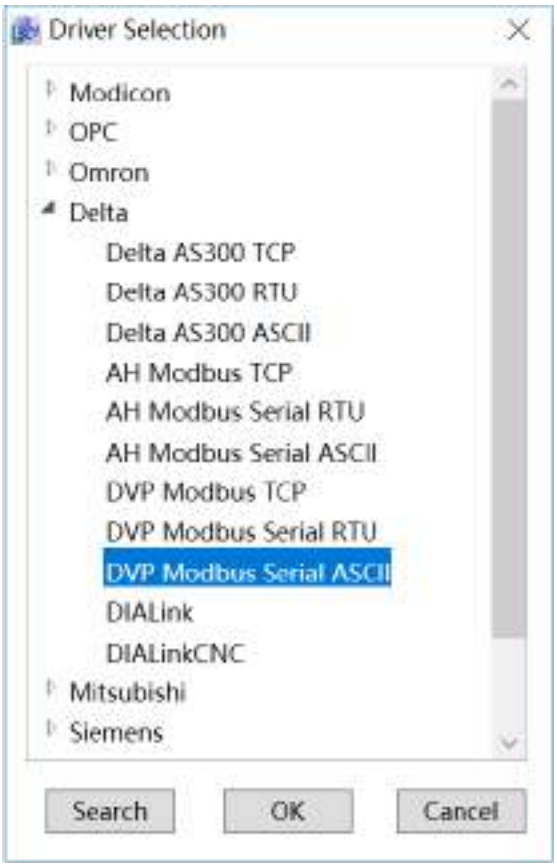
Example 1: Taking “Delta DVP ASCII”driver as an example

Step 1: Set up the hardware framework shown as the figure below, set 12SE COM2 communication data as 9600,7,E,1,ASCII, Station1:

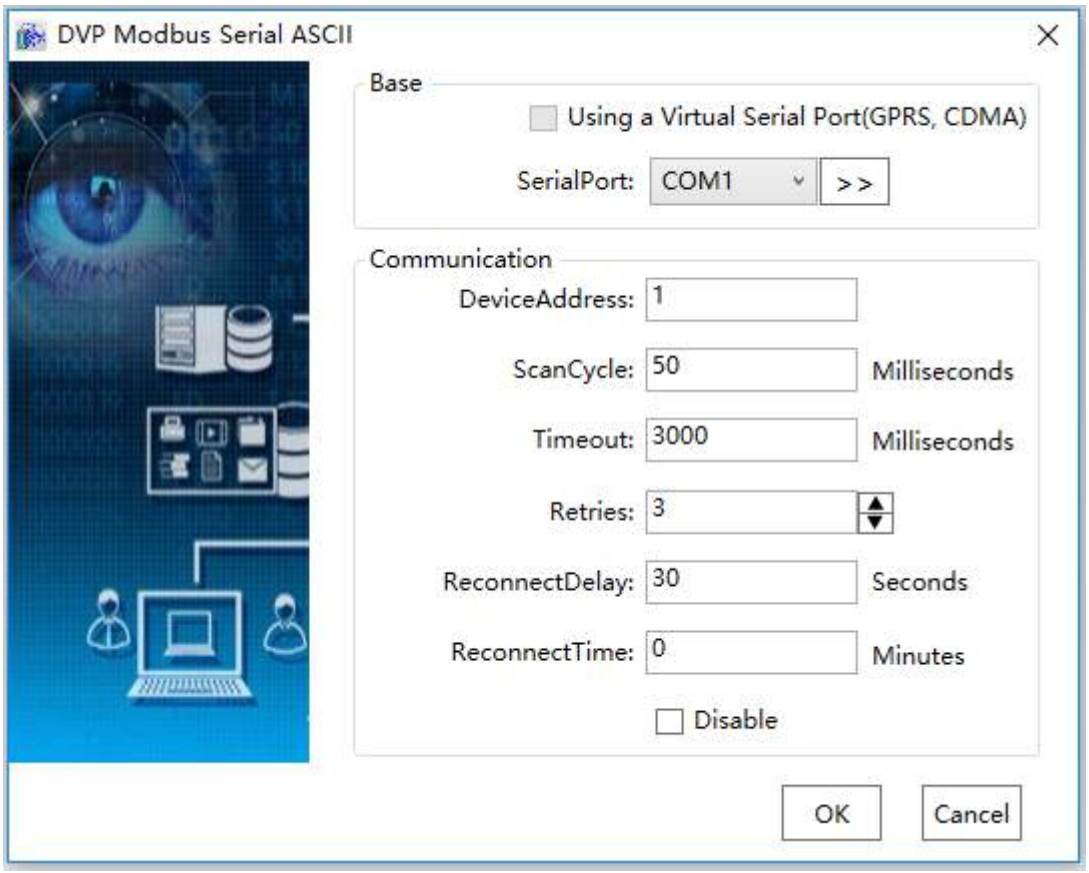


Step 2: Right-click on the “IODevice” node in the project management area and select” New Device”; the menu selection in the figure below will appear:

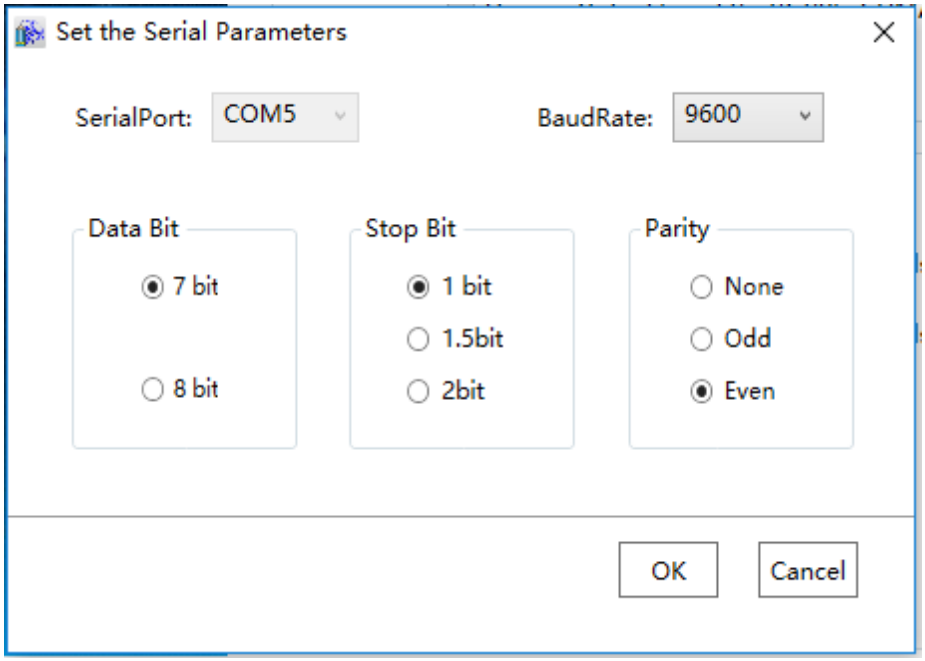
Step 3: Select “Delta” → “DVP Modbus Serial ASCII” in the “Driver Selection” window:



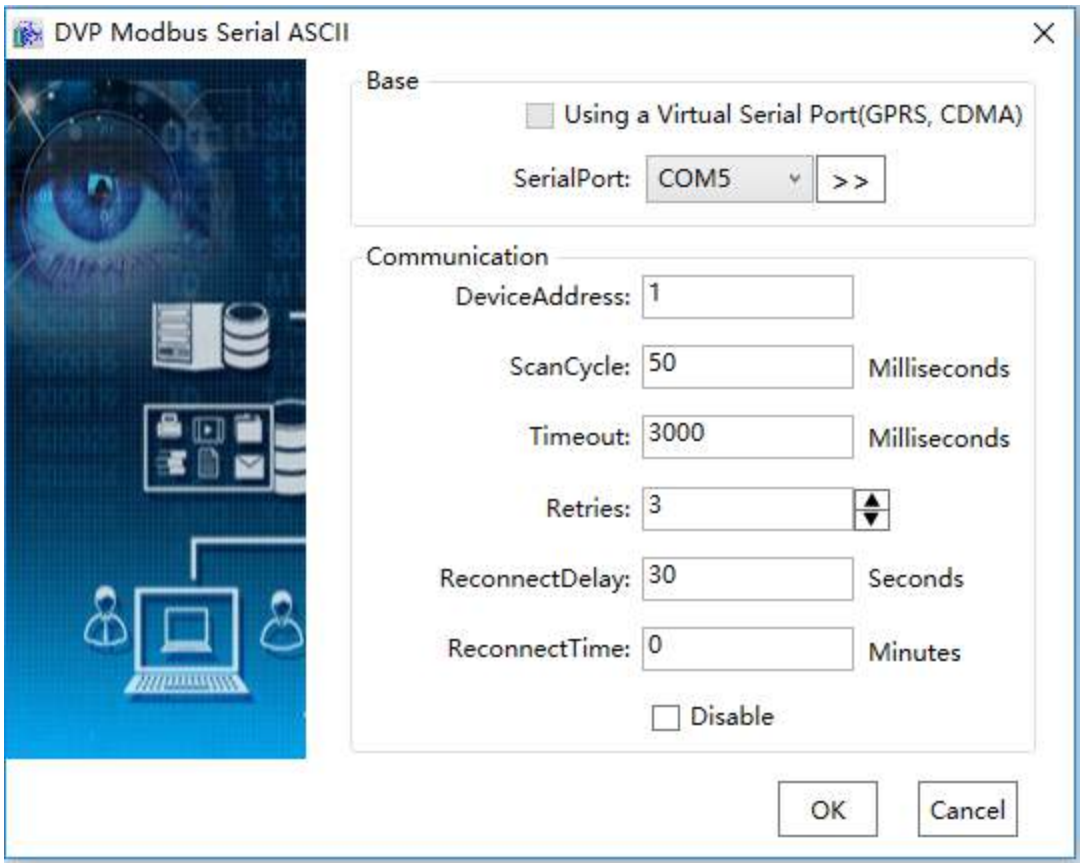
Step 4: Configure the Delta DVP ASCII communication parameters, as shown in the figure below:



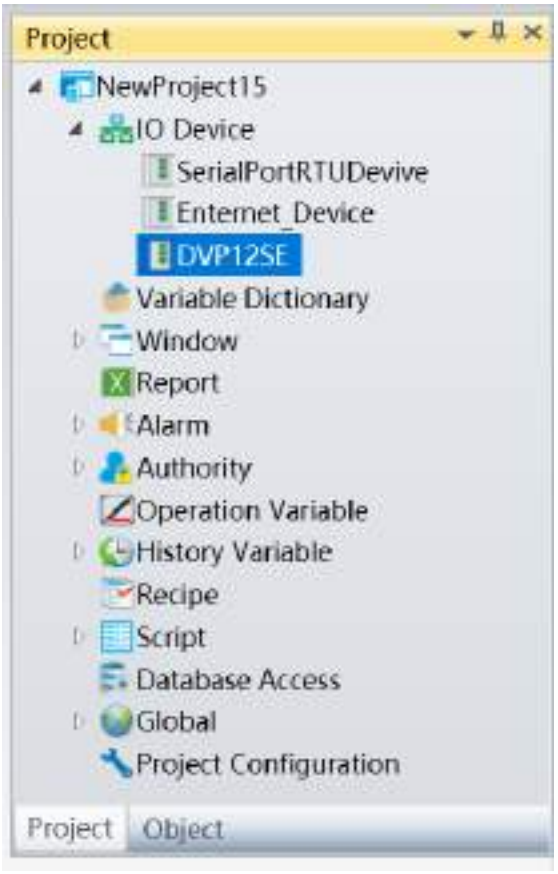
Step 5: Click the button to the right of “Base”→ “SerialPort”, open the “Set the Serial Parameters” window, set the serial port communication parameters: “9600,7,E,1,ASCII” and set the communication port as COM5 (you can see that the detected currently assigned serial communication port is COM5 in device manager), as shown in the figure below:



Step 6: Configure the serial port communication parameters, then click the “OK” button to return to the DVP Modbus Serial ASCII communication parameter configuration window. Set the communication and troubleshooting parameters. The “DeviceAddress” must be the same as the PLC station number, therefore set it as “1”. Default values can be used for the other options, as shown in the figure below:



Step 7: Click the “OK” button when all parameters have been configured and the device with the default name will appear under the “IODevice” node. Rename the newly created device as “DVP12SE” and compete the device communication, as shown in the figure below:



5.8.3.7 DIALink

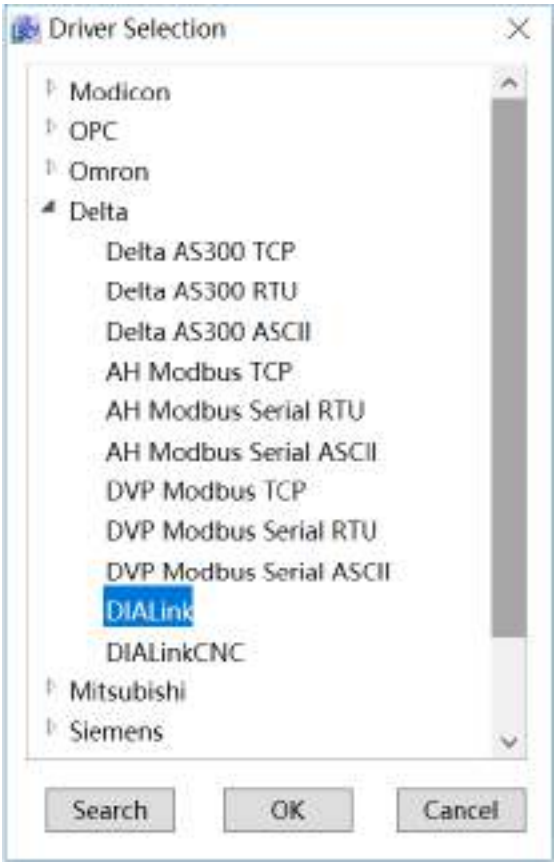
The DIAView configuration software supports communication with the DIALink server.

The example of configuring IO communication with DIALink server in DIAView configuration software is as follows:

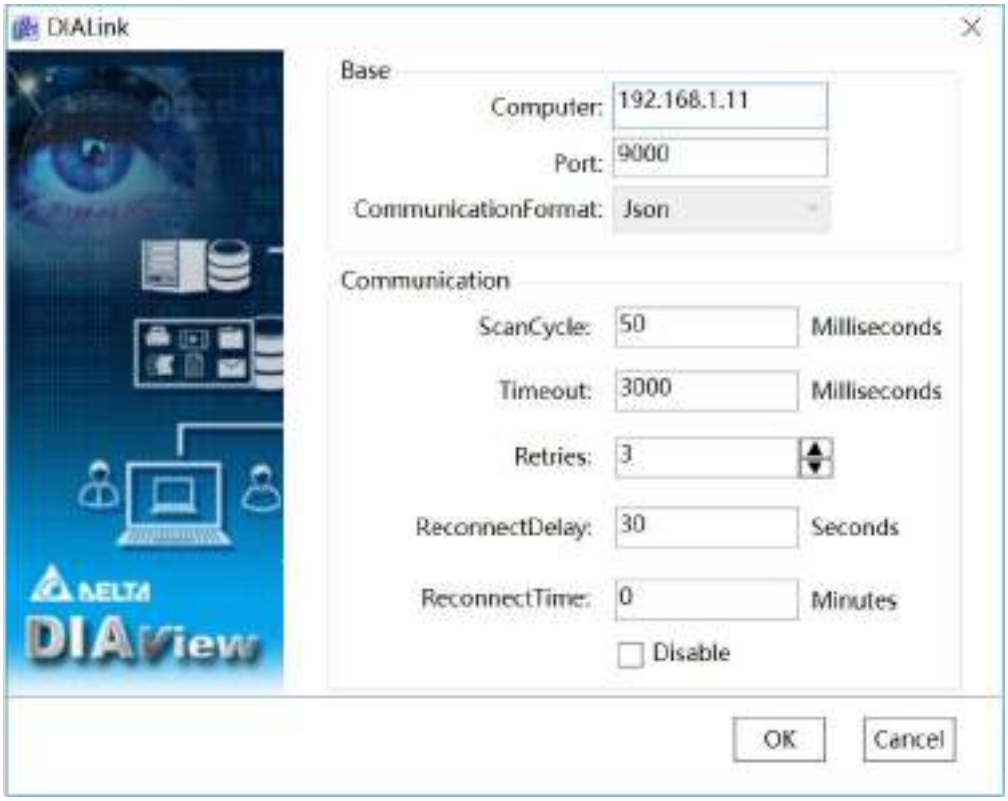
➢ **Creating communication between DIAView software and DIALink server:**

Step 1: In the project window tree directory, the "IODevice" node right-click on the "New Device" and pop up the following menu:

Step 2: In the select drive window, select "Delta" to "DIALink":



Step 3: Configuring DIALink communication parameters, the default values can be kept as follows:



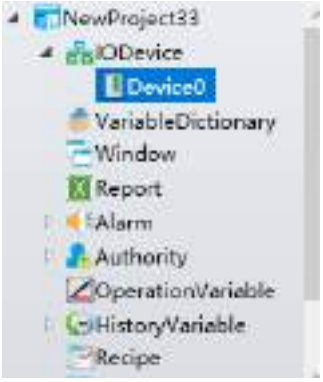
The meaning of each configuration item in the window:

Computer: IP or the full name of the computer.

Port： The default value is 9000.

CommunicationFormat: Default Json format, fixed.

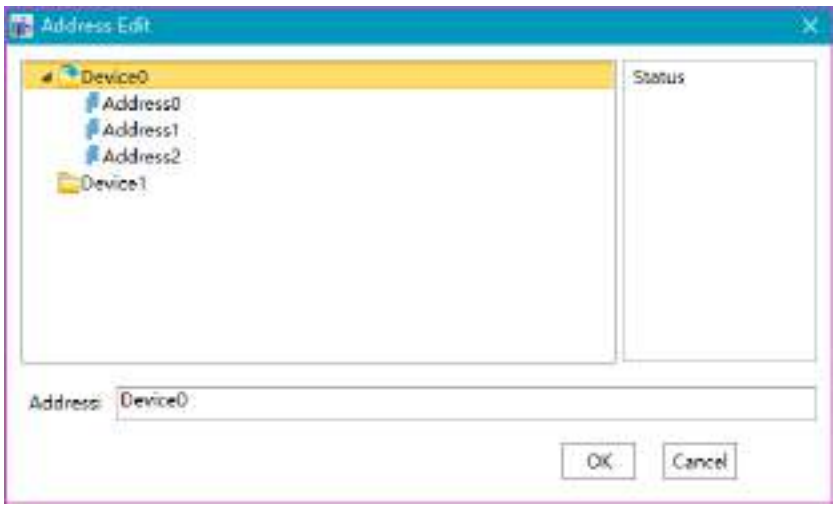
Step 4: Each parameter configuration is completed by clicking the "confirm" button. The default name of the device will appear under the "IODevice" node of the project tree directory:



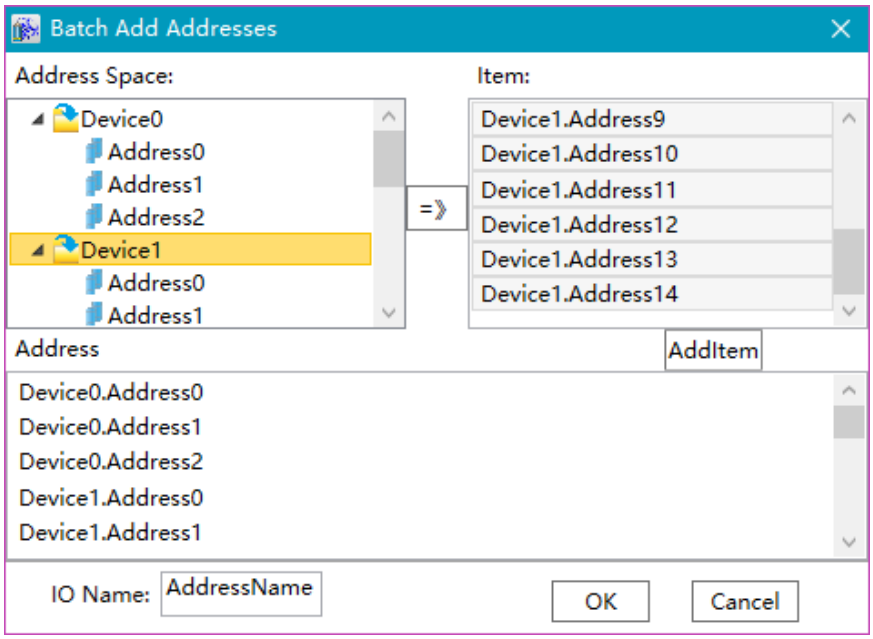
Step 5: Rename the new IO communication device to "DIALink" and complete the DIALink driver

communication configuration:

Step 6: The new address or the modified address is as follows:



Step 7: Batch new address, as follows:



Effect picture:

	Name	Address	Associated Variables	Value	Da
1	AddressName1	Device0.Address0			No
2	AddressName2	Device0.Address1			No
3	AddressName3	Device1.Address0			No
4	AddressName4	Device1.Address1			No
5	AddressName5	Device1.Address2			No
6	AddressName6	Device1.Address3			No
7	AddressName7	Device1.Address4			No
8	AddressName8	Device1.Address5			No
9	AddressName9	Device1.Address6			No
10	AddressName10	Device1.Address7			No
11	AddressName11	Device1.Address8			No
12	AddressName12	Device1.Address9			No
13	AddressName13	Device1.Address10			No
14	AddressName14	Device1.Address11			No
15	AddressName15	Device1.Address12			No
16	AddressName16	Device1.Address13			No
17	AddressName17	Device1.Address14			No

Step 8: Can be tested(saved the address first)

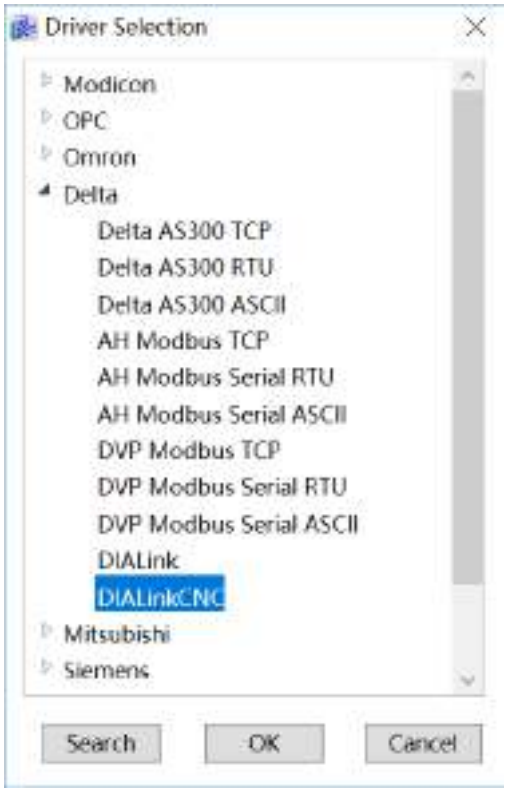
5.8.3.8 DIALinkCNC

The DIAView configuration software supports communication with the DIALinkCNC server.

The example of configuring IO communication with DIALinkCNC server in DIAView configuration software is as follows:

➢ **Creating communication between DIAView software and DIALinkCNC server:**

Step 1: In the project window tree directory, the "IODevice" node right-click on the "New Device" and pop up the following menu,In the select drive window, select "Delta" to "DIALinkCNC":



Step 2: Configuring DIALinkCNC communication parameters, the default values can be kept as follows:

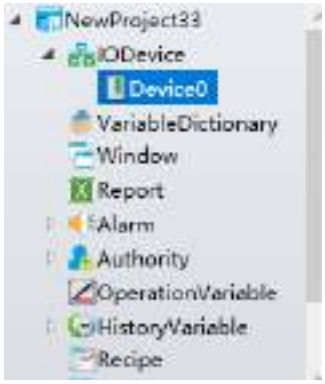
The meaning of each configuration item in the window:

Computer: IP or the full name of the computer.

UserName： Login name of DIALinkCNC.

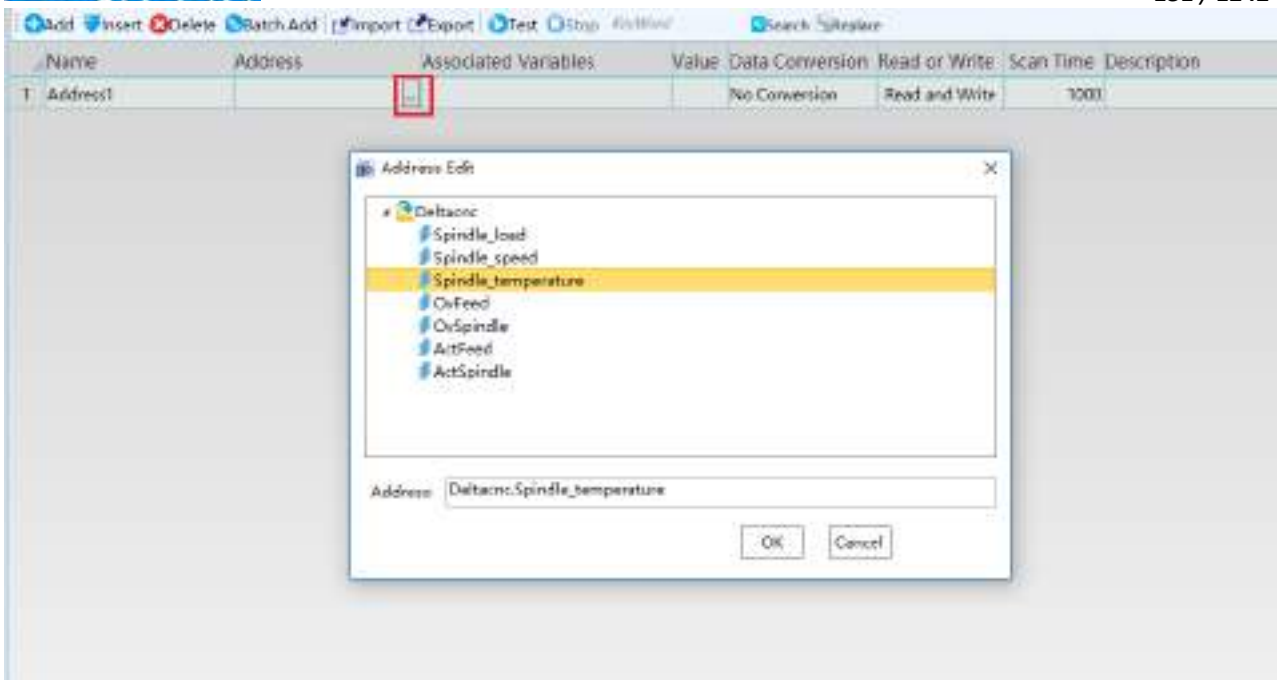
Password: The password used to log into DIALinkCNC.

Step 3: Each parameter configuration is completed by clicking the "confirm" button. The default name of the device will appear under the "IODevice" node of the project tree directory:



Step 4: Rename the new IO communication device to "DIALinkCNC" and complete the DIALinkCNC driver communication configuration:

Step 5: The new address or the modified address is as follows:



Step 6: The test can start when the address creation is complete (save the address first).

5.8.4 Mitsubishi

DIAView software supports the communication of Mitsubishi PLC.

Supporting device: Mitsubishi FX COM serial、Mitsubishi FX Ethernet serial,Mitsubishi Q COM serial,Mitsubishi Q Ethernet serial.

Supporting command format: Mitsubishi format 1 - format 5 communication,Mitsubishi ASCII/Binary communication.

Supporting communication network interface: Ethernet and serial.

5.8.4.1 Mitsubishi FX Serial

DIAView software supports the communication of the equipment based on Mitsubishi communication protocol standard by serial port (format 1, format 4).

Supporting devices:

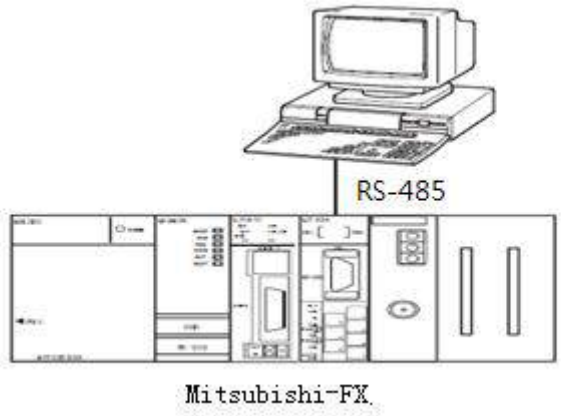
FX3U,FX3G,FX3S,FX2N,FX1N,FX2C,FX1S,FX0N.

The example that DIAView software communicates with equipments by serial port is as follows:

➢ **Creating communication between DIAView software and Mitsubishi-FX through serial port:**

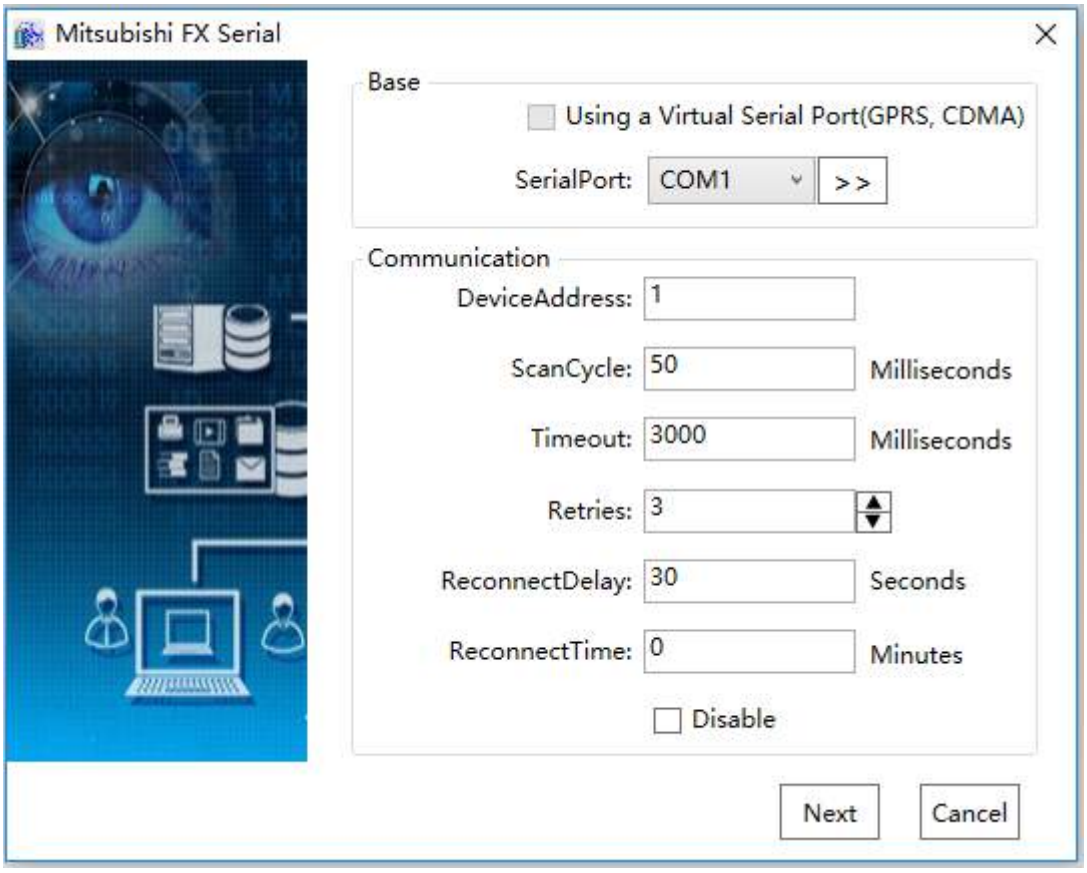
Example 1: Taking “FX3U, format 4” as an example:

Step 1: Create the following hardware framework，the communication parameters are 19200,8,N,1,universal QCPU, format 4, station2:

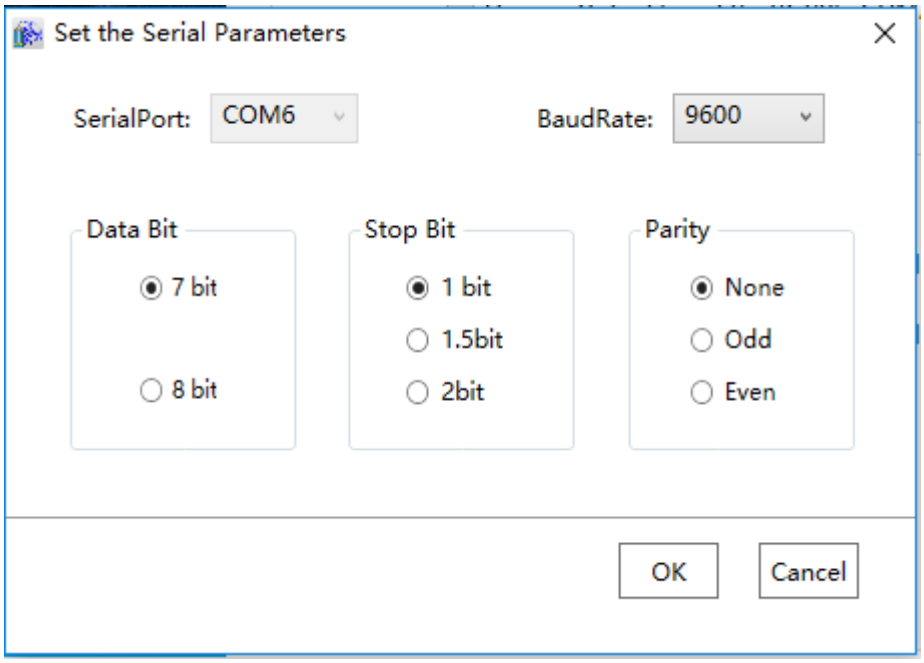


Step 2: In the project management area, right click on “IO Device “node and select “ New Device “, select " Mitsubishi " →“Mitsubishi FX Series” in the driver selection window, as shown in the figure below:

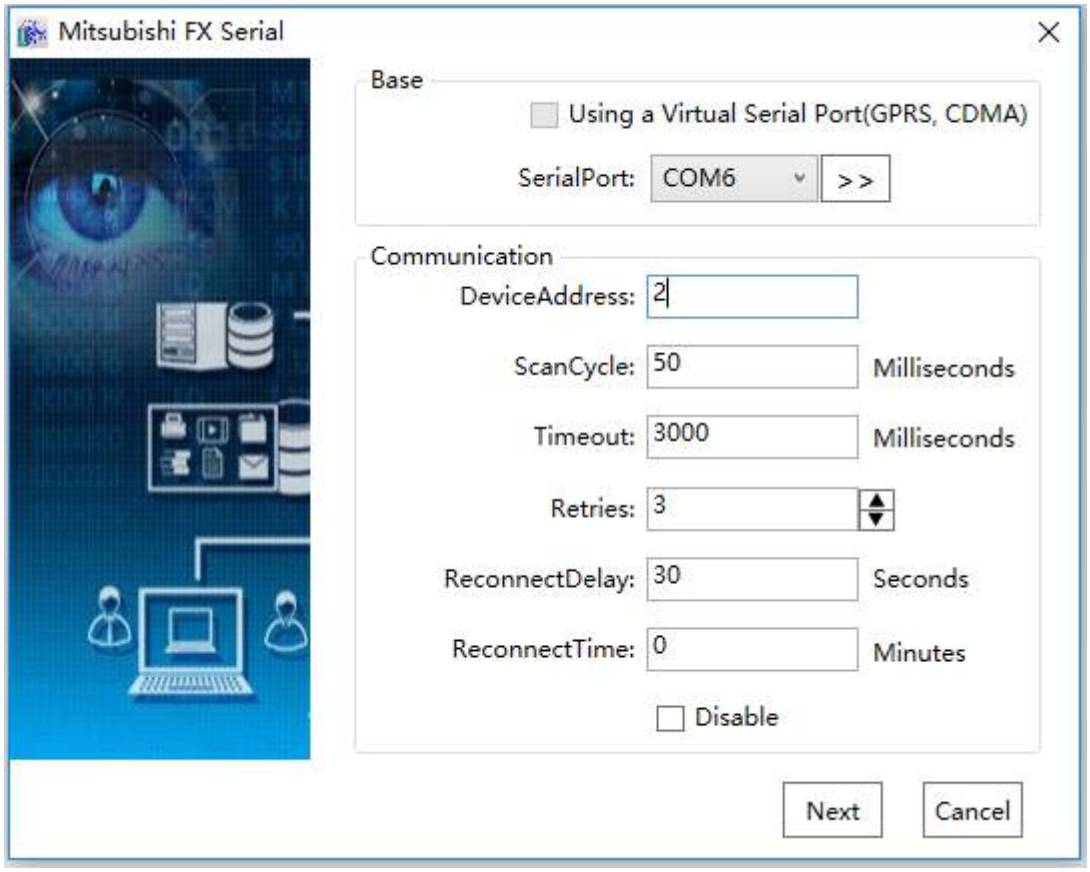
Step 3: Configure FX3U, Format 1 communication parameters, as shown in the figure below:



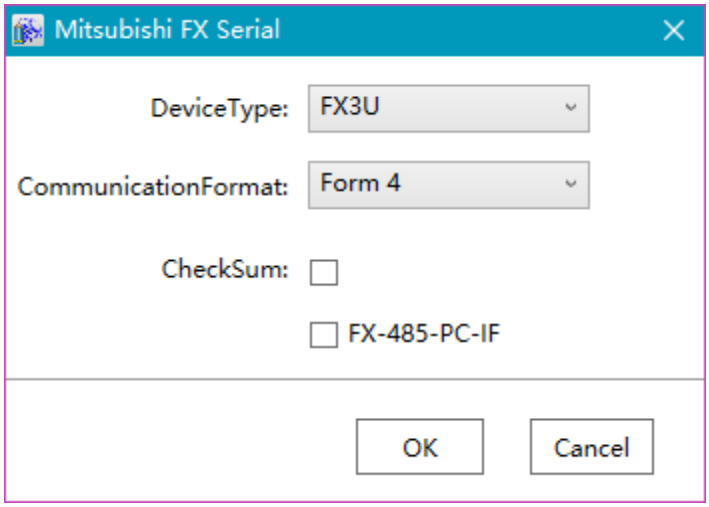
Step 4: Click “Base”→ click the right button of“SerialPort”,open“Set the Serial Parameters”to set serial port communication parameters to:“9600,7,N,1”,communication port is COM6(PS : when the Mitsubishi PLC stop bit is been set to 1,the DIALink stop bit must be set to 1.5),as shown in the figure below:



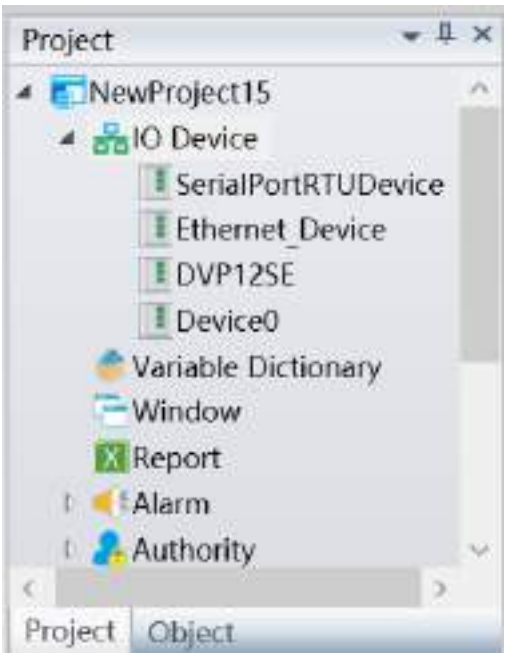
Step 5: Click "OK" button when serial communication parameters have been all configured, return to the window of communication parameter configuration to set communication parameters and fault processing parameters, the device address should be consistent with PLC station number, should be all set to "2". Other parameters can be the default value, as shown in the figure below:



Step 6: Click “next” button to open “Mitsubishi FX Serial”to set communication parameters, as shown in the figure below(RS485 communication must check fx-485-pc-if):



Step 7: Click the "OK" button when all parameters have been configured and the device with default name will appear under “IO Device” node of the project tree directory:



Step 8: Rename the newly-built communication device as "Serial_Device", complete the serial port driver communication configuration, as shown in the figure below:

- **Noted:** Mitsubishi FX3U address description, as shown in the table below:

Register	Range	Default type	Description
X	0 - 377	bool	Input register
Y	0 - 377	bool	Output register

M	0 - 7679	bool	Auxiliary register
S	0 - 4095	bool	Status register
SM	8000-8511	bool	Special auxiliary register
TS	0 - 511	bool	Counter contact register
CS	0 - 2047	bool	Counter contact register
TN	0 - 511	word	Timer register
CN16	0 - 199	word	16 bit counter register
CN32	200 - 255	word	32 bit counter register
D	0 - 7999	word	Data register
SD	8000-8511	word	Special data register
R	0 - 32767	word	Extended register

5.8.4.2 Mitsubishi ProFX Serial

DIAView software supports the communication of the equipment based on Mitsubishi communication protocol standard by programming port.

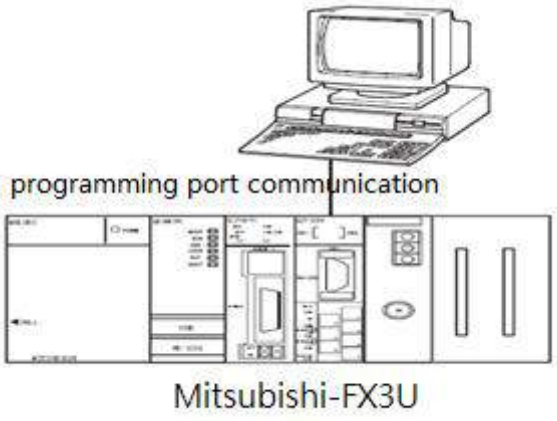
Supporting device: FX3U.

The example that DIAView software communicates with equipments by Ethernet is as follows:

➢ **Create communication between DIAView software and Mitsubishi-FX3U through programnimg port:**

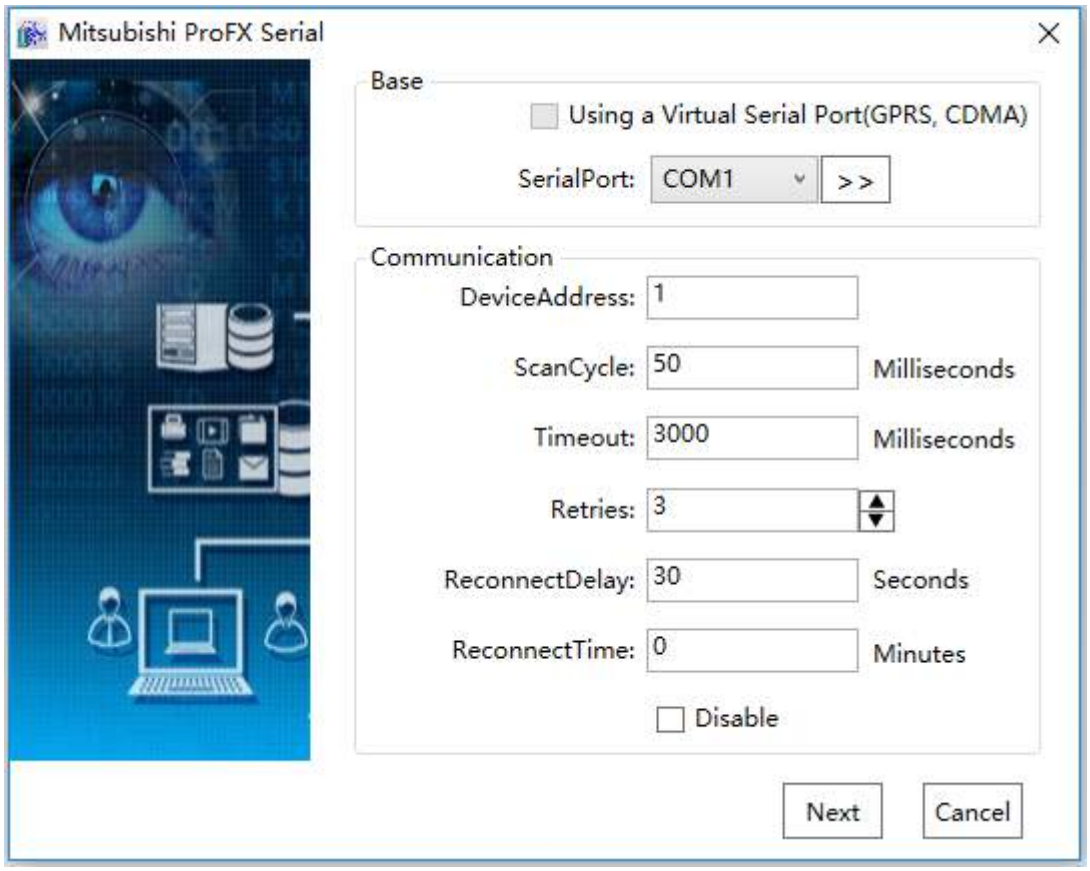
Example 1: Taking“FX3U, programming port”driver as an example

Step 1: Create the following hardware framework, set programming port parameters to 9600, 7, E, 1, FX3UC, Program:

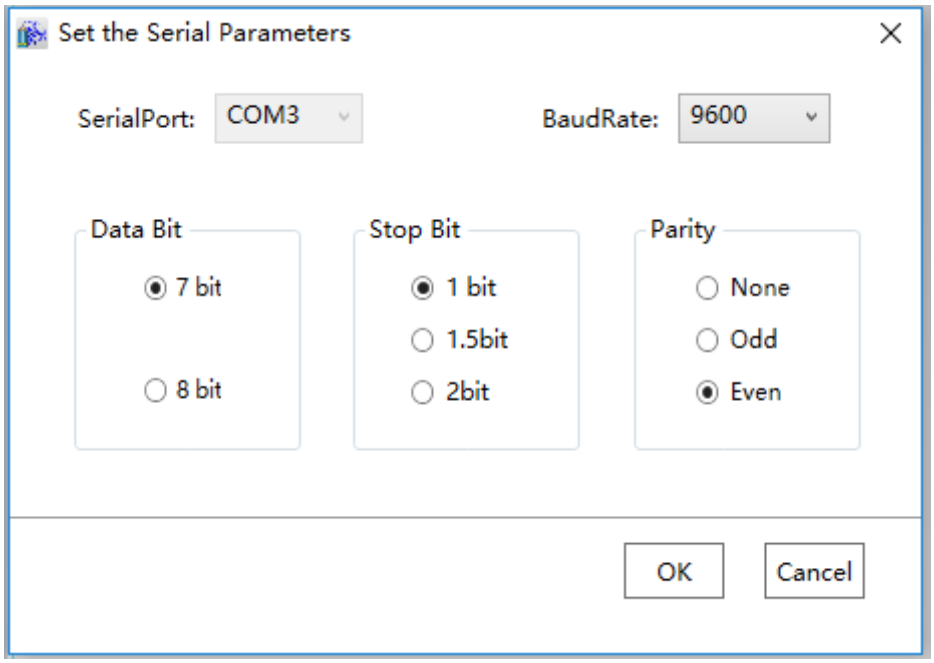


Step 2: In the project management area, right click on “IO Device “node and select “ New Device “, select “Mitsubishi ProFX Serial” in the driver selection window, as shown in the figure below:

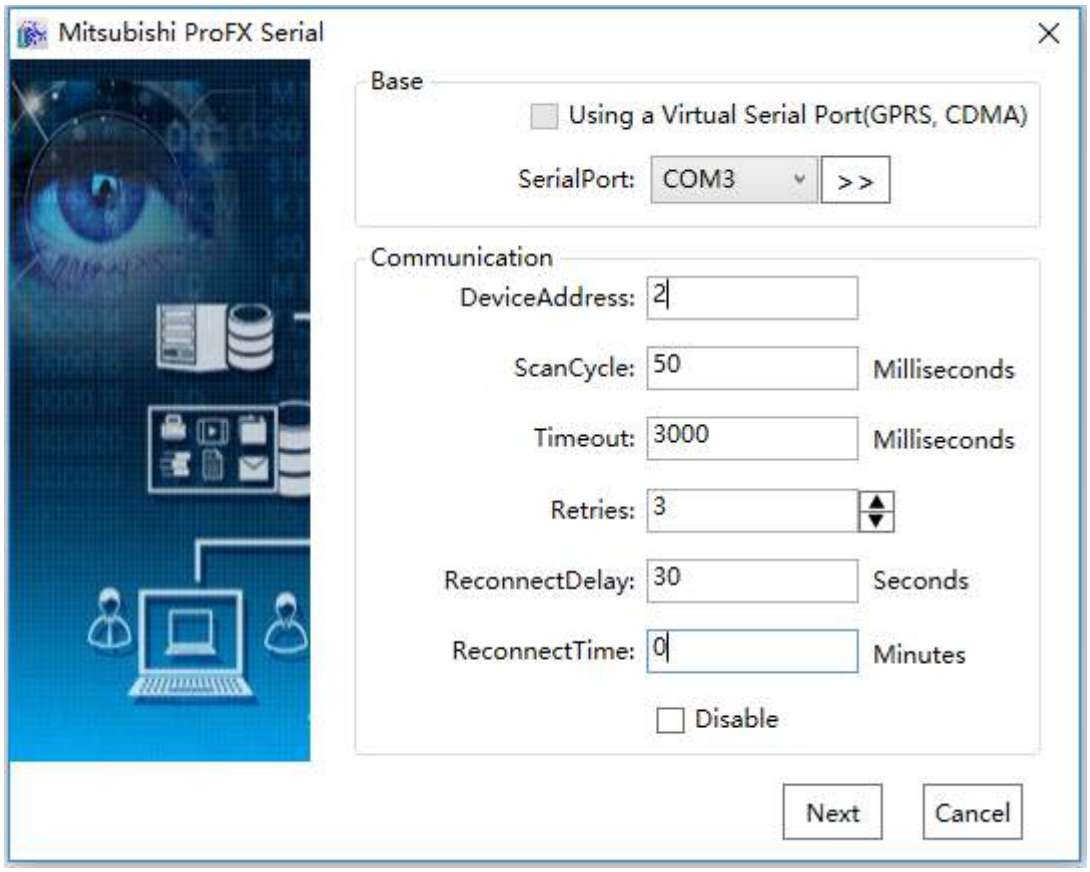
Step 3: Configure FX3U, program port communication parameters, as shown in the figure below:



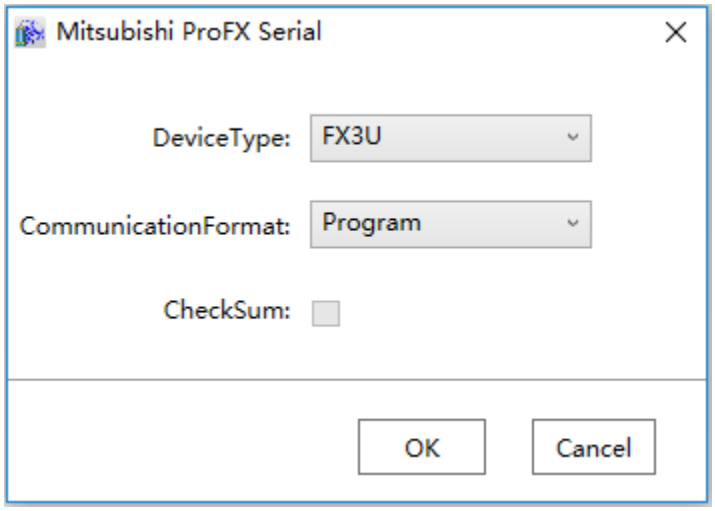
Step 4: Click “Base”→ click the right button of “SerialPort”, open “Set the Serial Parameters “to set serial port communication parameters to:“9600,7,E,1”, serial port is COM3(In the computer device manager, you can see the current serial port detected is COM3, PS: when the Mitsubishi PLC stop bit is set to 1, the DIALink stop bit must be set to 1.5), as shown in the figure below:



Step 5: Click "OK" button when serial communication parameters are all configured, return to the window of communication parameter configuration to set communication parameters and fault processing parameters, the device address should be consistent with PLC station number, should be all set to "2". Other parameters can be the default value, as shown in the figure below:

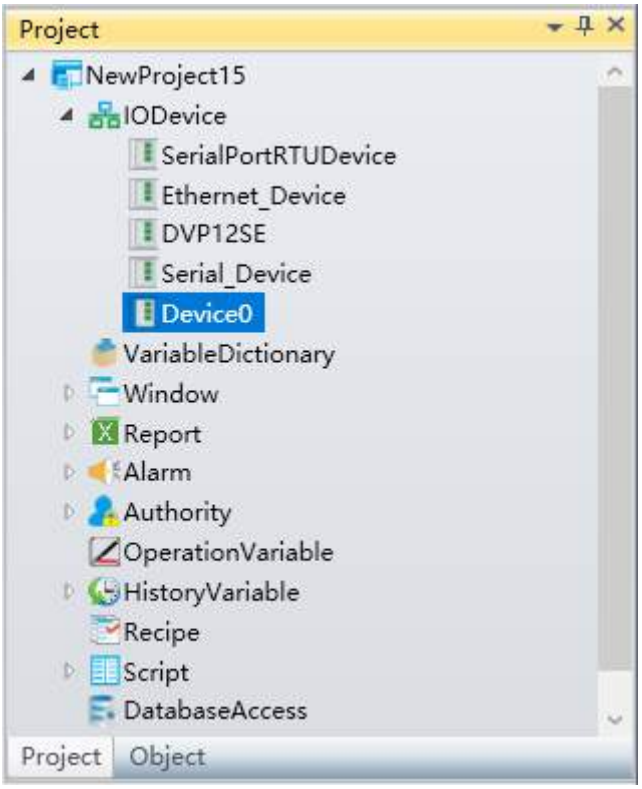


Step 6: Click “Next” button to open “Mitsubishi ProFX Serial”to set communication parameters, as shown in the figure below:

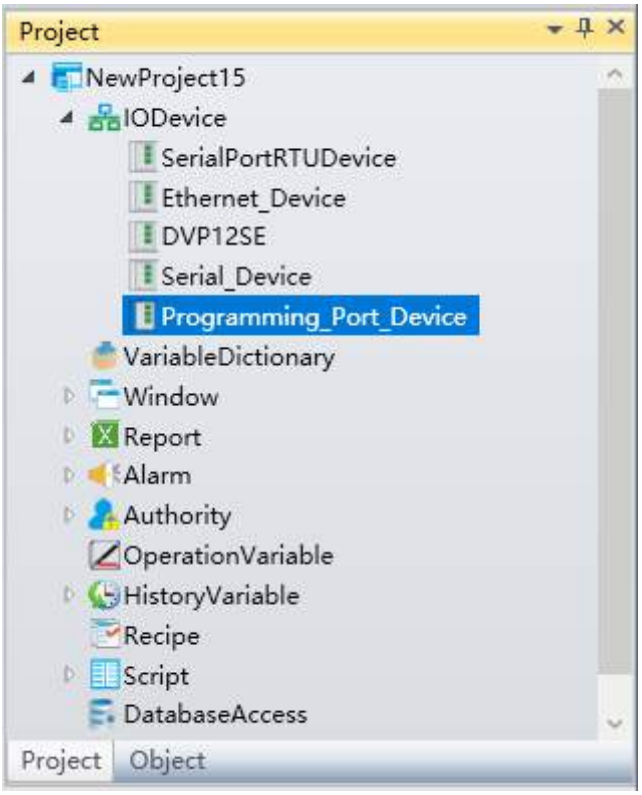


Step 7: Click the "OK" button when all parameters have been configured and the device with default

name will appear under “IO Device” node of the project tree directory.



Step 8: Rename the newly-built IO communication device as "Programming_Port_Device" and complete programming port driver communication configuration, as shown in the figure below:



➢ **Noted:** Mitsubishi FX3U address description, as shown in the table below:

Register	Range	Default type	Description
X	0- 377	bool	Input register
Y	0 - 377	bool	Output register
M	0 - 7679	bool	Auxiliary register
S	0 - 4095	bool	Status register
TS	0 - 511	bool	Timer contact register
CS	0 - 255	bool	Counter contact register
TN	0 - 511	word	Timer register
CN16	0 - 199	word	16 bit counter register
CN32	200 - 255	word	32 bit counter register
D	0 - 7999	word	Data register
SD	8000-8511	word	Special data register
R	0 - 32767	word	Extended register

5.8.4.3 Mitsubishi Q Serial

DIAView software supports the communication of the equipment based on Mitsubishi communication protocol standard by serial port (format 1~5).

Supporting device:

Basic QCPU: Q00JCPU, Q00CPU, Q01CPU.

High-performance QCPU: Q02CPU, Q02HCPU, Q06HCPU, Q12HCPU, Q25HCPU.

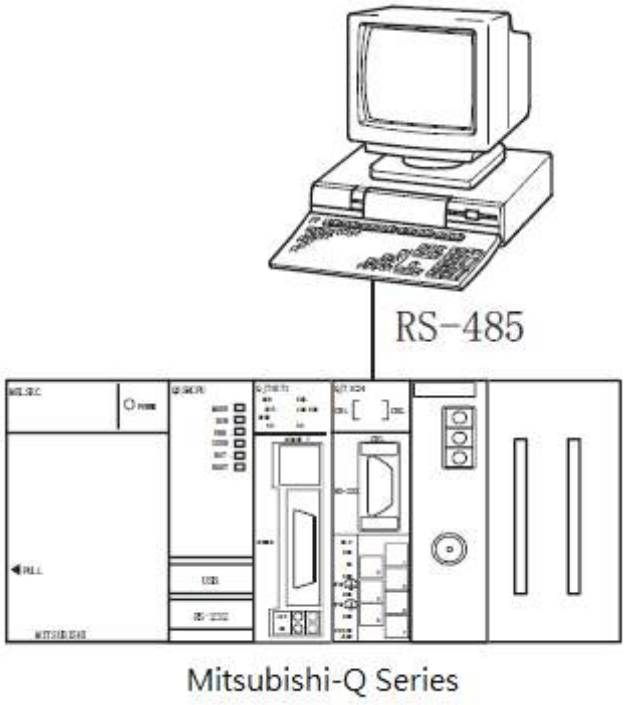
General QCPU: Q00UJCPU, Q00UCPU, Q01UCPU, Q02UCPU, Q03UDCPU, Q04UDHCPU, Q06UDHCPU, Q10UDHCPU,Q13UDHCPU, Q20UDHCPU, Q26UDHCPU, Q03UDECPU, Q04UDEHCPU, Q06UDEHCPU, Q10UDEHCPU, Q13UDEHCPU, Q20UDEHCPU, Q26UDEHCPU, Q50UDEHCPU, Q100UDEHCPU.

The example that DIAView configuration software communicates with equipments by serial port is as follows:

➢ **Creating communication between DIAView software and Mitsubishi-Q through serial port:**

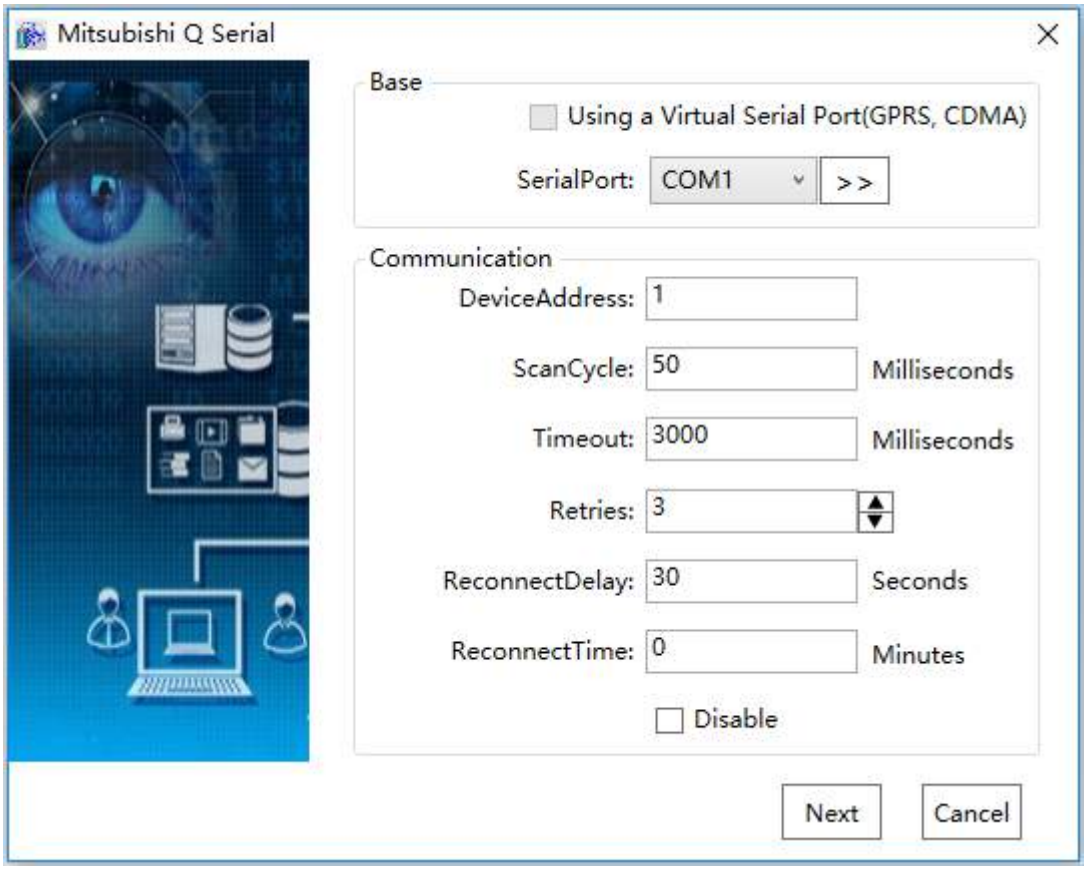
Example 1: Taking“general QCPU, format4”driver as an example

Step 1: Create the following hardware framework, the communication parameters are 19200,8,N,1,universal QCPU, format 4, station3:

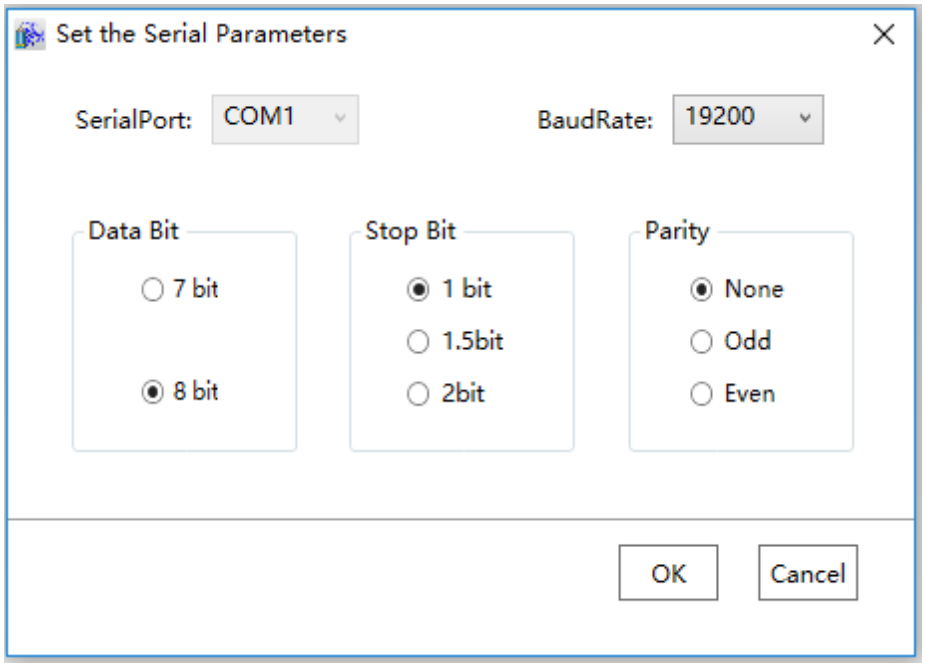


Step 2: Right-click on the “IODevice” node in the project management area and select “New Device”, select “Mitsubishi” →“Mitsubishi Q Serial” in the menu selection as shown in the figure below:

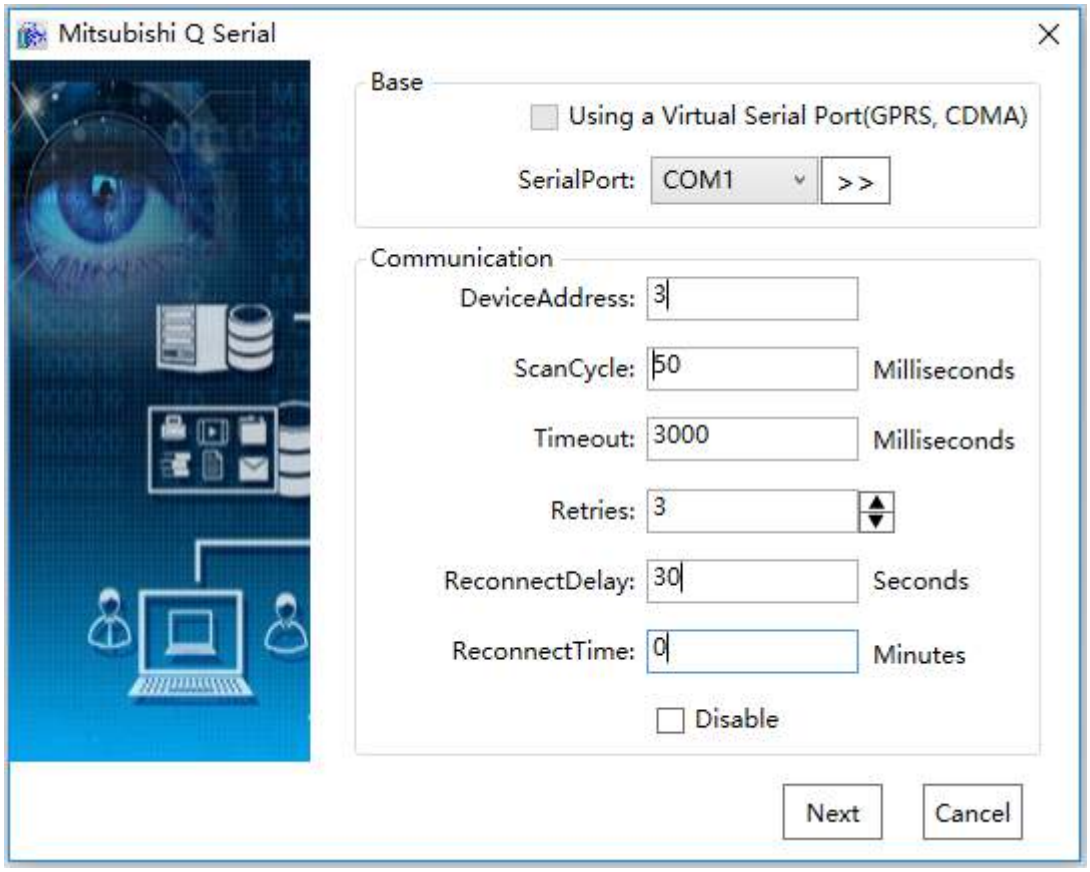
Step 3: configure the CPU, format 4communication parameters, as shown in the figure below:



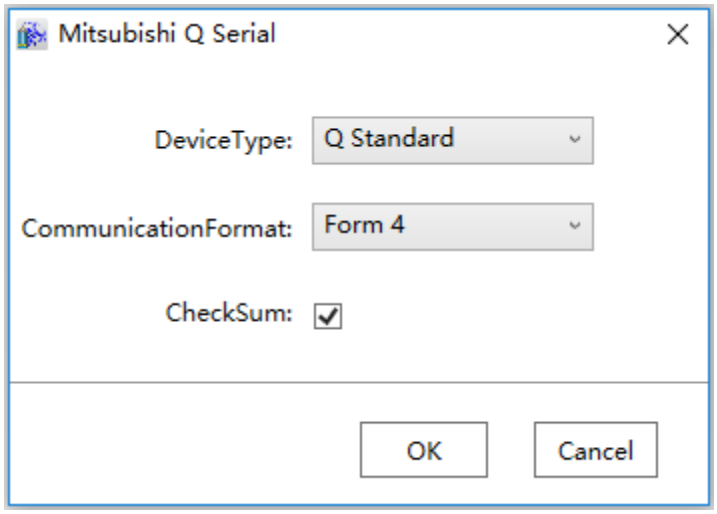
Step 4: Click the button to the right of “Base”→ “SerialPort” to open the “Set the Serial Parameters” window, set the serial port communication parameters: “19200,8,N,1” and set the communication port as COM6 (you can see that the detected currently assigned serial communication port is COM6 in device manager, noted: when setting the stop bit as 1, DIALink stop bit should be set as 1.5), as shown in the figure below:



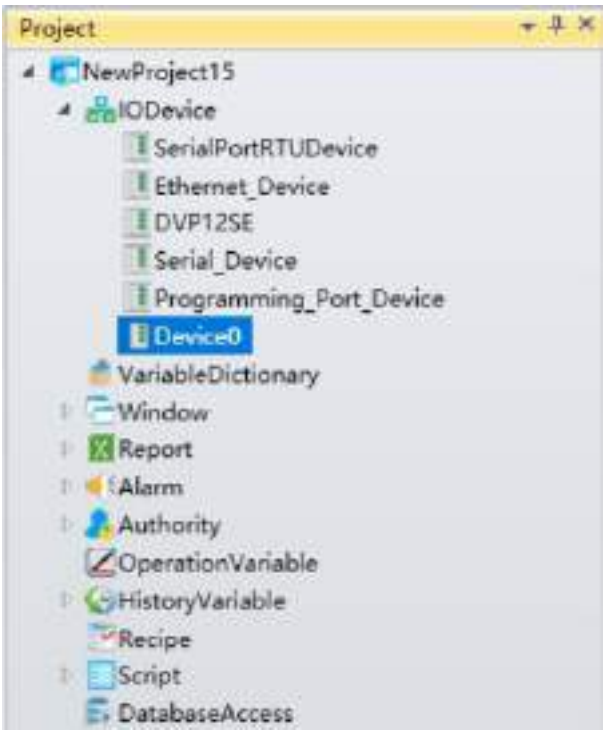
Step 5: Click the “OK” button when all parameters have been configured to return communication parameter configuration window, set communication and troubleshooting parameters, “DeviceAddress” should be the same as PLC station number, both are set as “3”, other parameters can be default value, as shown in the figure below:



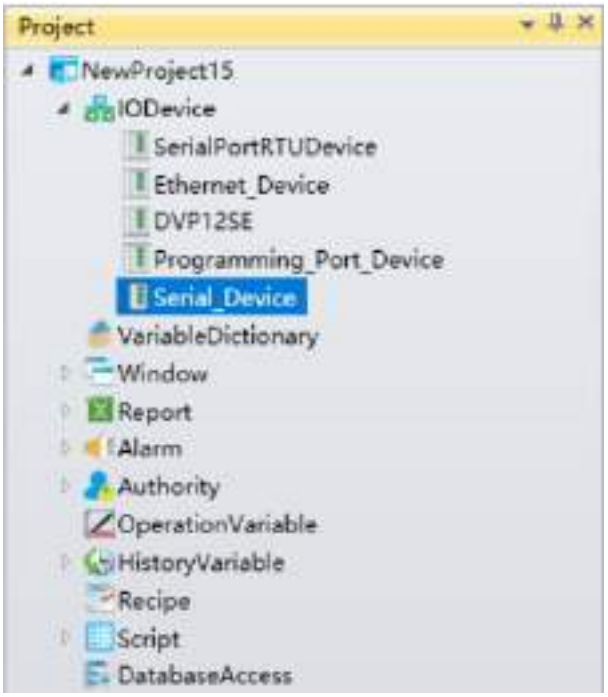
Step 6: Click “Next” button, open “Mitsubishi Q Serial” window, the parameters are as shown in the figure below:



Step 7: Click the “OK” button when all parameters have been configured and the device with the default name “Device0” will appear under the “IODevice” node:



Step 8: Rename the newly-built IO communication device as "Serial_Device" and complete the serial port driver communication configuration as shown in the figure below:



- **Noted:** Mitsubishi general QCPU address description as shown in the table below:

Register	Range	Default type	Description

SM	0 - 2047	bool	Specical relay
X	0 - 0x1FFF	bool	Input register
Y	0 - 0x1FFF	bool	Output register
M	0 - 8191	bool	Interior relay
L	0 - 8191	bool	Latch register
F	0 - 2047	bool	Alarm
V	0 - 2047	bool	Index relay
B	0 - 0x1FFF	bool	Link relay
TS	0 - 2047	bool	Timer contact
TC	0 - 2047	bool	Timer coil
CC	0 - 1023	bool	Counter coil
CS	0 - 1023	bool	Counter contact
SB	0 - 0x7FF	bool	Link special relay
S	0 - 8191	bool	Stepping relay
SD	0 - 2047	word	Special register
D	0 - 12287	word	Data register
W	0 - 0x1FFF	word	Link register
TN	0 - 2047	word	Timer current value
CN	0 - 1023	word	Counter current value
SW	0 - 0x7FF	word	Link special register

5.8.4.4 Mitsubishi Q EnterNet

DIAView software supports the communication of the equipment based on Mitsubishi communication protocol standard by Ethernet(Binary Code, ASCII Code).

Supporting device:

Basic QCPU: Q00JCPU, Q00CPU, Q01CPU.

High performance QCPU: Q02CPU, Q02HCPU, Q06HCPU, Q12HCPU, Q25HCPU.

General QCPU: Q00UJCPU, Q00UCPU, Q01UCPU, Q02UCPU, Q03UDCPU, Q04UDHCPU, Q06UDHCPU, Q10UDHCPU,Q13UDHCPU, Q20UDHCPU, Q26UDHCPU, Q03UDECPU, Q04UDEHCPU, Q06UDEHCPU, Q10UDEHCPU,

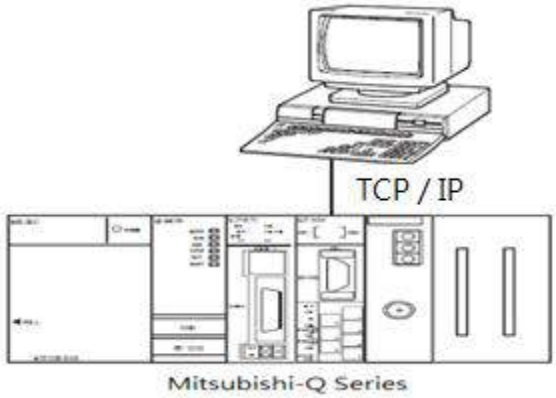
Q13UDEHCPU, Q20UDEHCPU, Q26UDEHCPU, Q50UDEHCPU, Q100UDEHCPU.

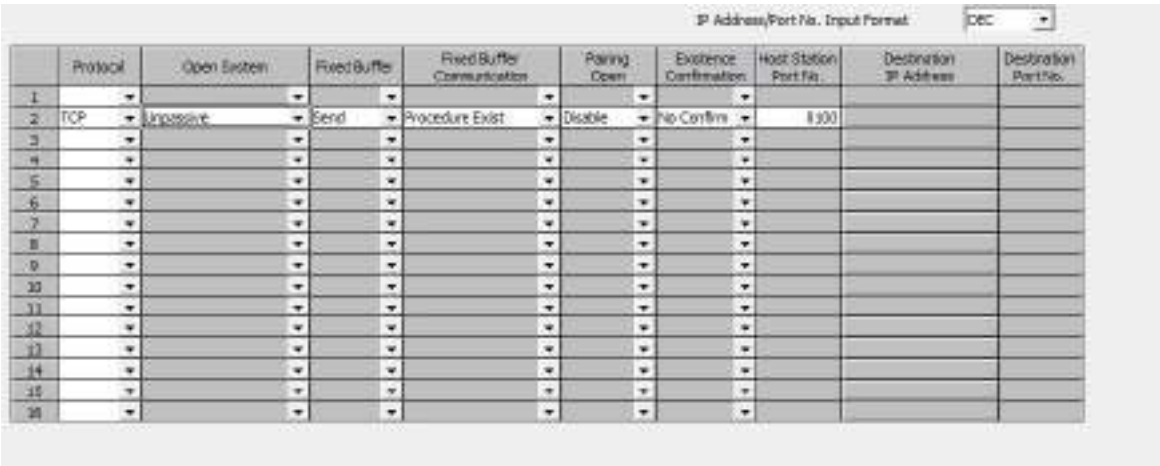
The example that DIAView software communicates with equipments by Ethernet is as follows:

➢ **Creating communication between DIAView software and Mitsubishi-Q series through Ethernet:**

Example 1: Taking“General QCPU, Binary”driver as an example

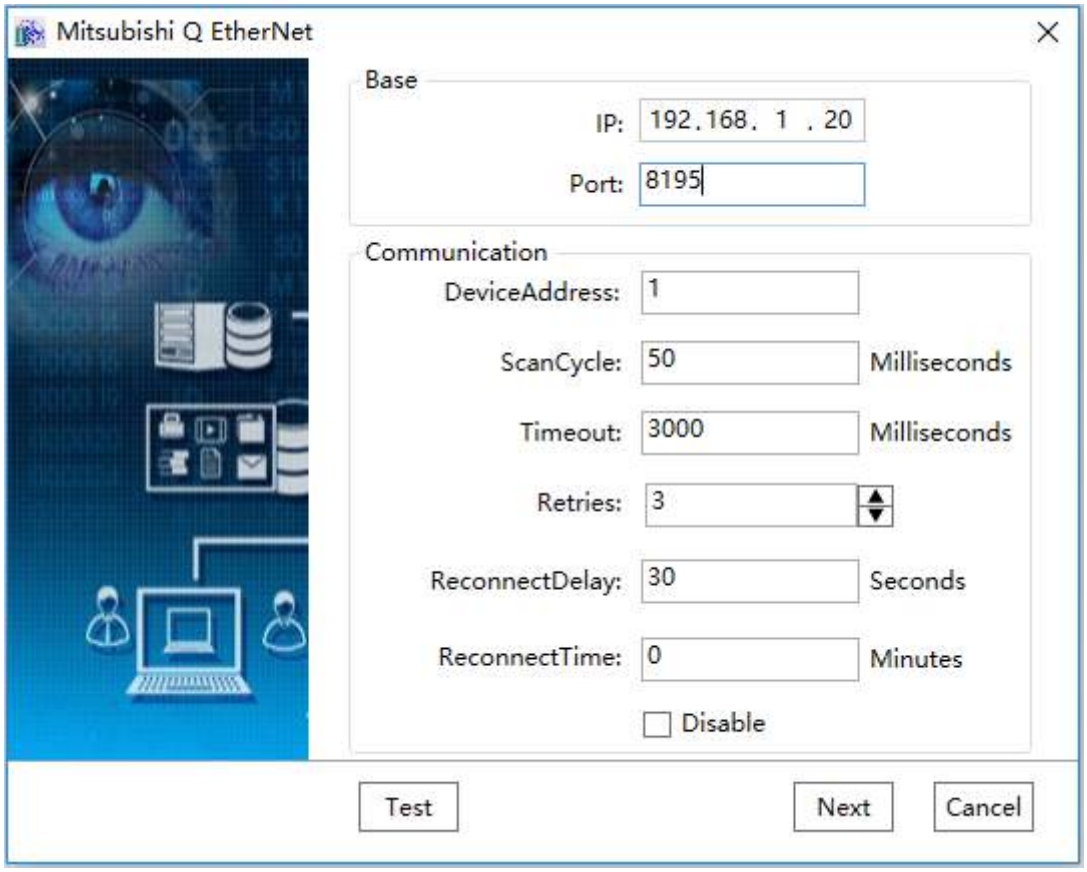
Step1: Create the following hardware framework, set IP address as “192.168.1.20”, the computer IP address is “192.168.1.200” (keep PLC and the computer in the same LAN), as shown in the figure below:



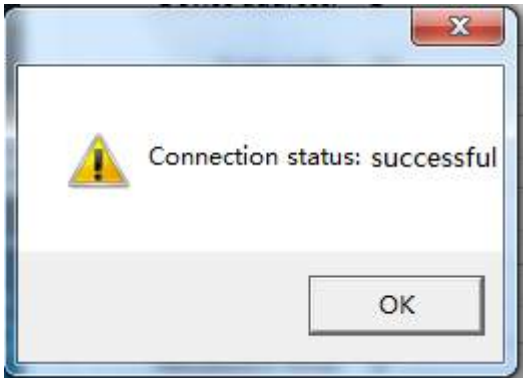


Step2: Right-click on the “IODevice” node in the project management area and select “New Device”, select “Mitsubishi” →“Mitsubishi Q EtherNet ” in the menu selection as shown in the figure below:

Step3: Configure the universal QCPU, Binary communication parameters and set the IP address in the Basic setting as: 192.168.1.20, set “Port” as: 8195, as shown in the figure below:

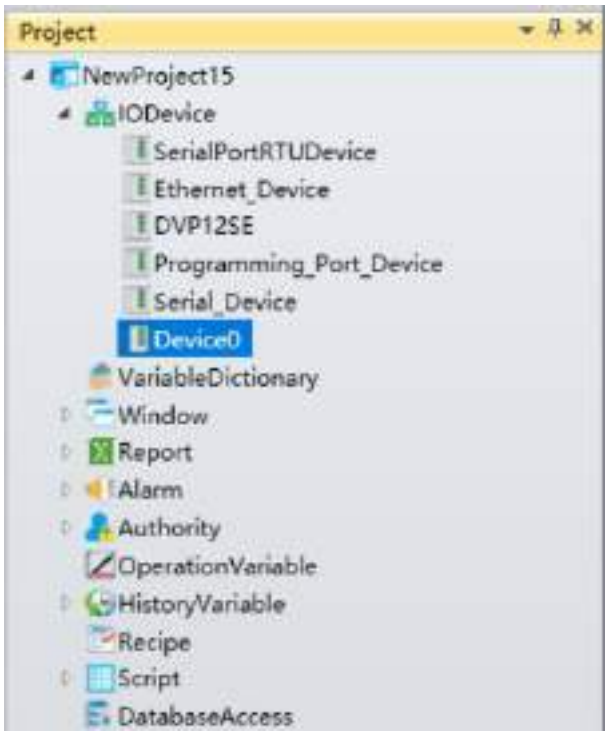


Step4: Other option can use the default values, click “Test” button to test the connection, as shown in the figure below:

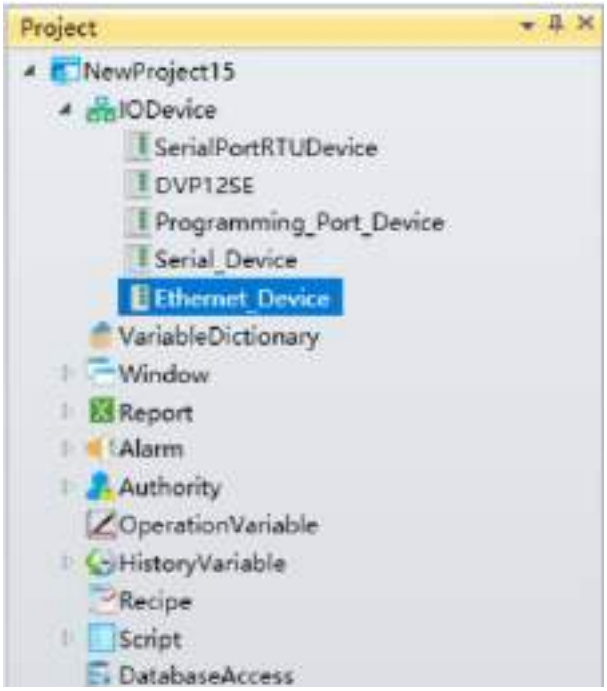


Step5: Click “Next” button to open “Mitsubishi Q EtherNet” window, the parameters are as shown in the figure below:

Step6: Click the “OK” button when all parameters have been configured and the device with the default name “Device0” will appear under the “IODevice” node:



Step7: Rename the newly-built IO communication device as "Ethernet_Device" and complete Ethernet driver communication configuration, as shown in the figure below:



- **Noted:** Mitsubishi general QCPU address description as shown in the table below:

Register	Range	Default type	Description
SM	0 - 2047	bool	Special relay
X	0 - 0x1FFF	bool	Input register
Y	0 - 0x1FFF	bool	Output register
M	0 - 8191	bool	Interior relay
L	0 - 8191	bool	Latch relay
F	0 - 2047	bool	Alarm
V	0 - 2047	bool	Index relay
B	0 - 0x1FFF	bool	Link relay
TS	0 - 2047	bool	Timer contact
TC	0 - 2047	bool	Timer coil
CS	0 - 1023	bool	Counter contact
CC	0 - 1023	bool	Counter coil

CS	0 - 1023	bool	Counter contact
SB	0 - 0x7FF	bool	Link special relay
S	0 - 8191	bool	Stepping relay
SD	0 - 2047	word	Special register
D	0 - 12287	word	Data register
W	0 - 0x1FFF	word	Link register
TN	0 - 2047	word	Timer current value
CN	0 - 1023	word	Counter current value
SW	0 - 0x7FF	word	Link special register

5.8.4.5 Mitsubishi FX EnterNet

DIAView software supports the communication of the equipment based on Mitsubishi communication protocol standard by Ethernet(Binary Code, ASCII Code).

Supporting devices:

PLC: FX3UC.

Ethernet Module: FX3U-ENET-L.

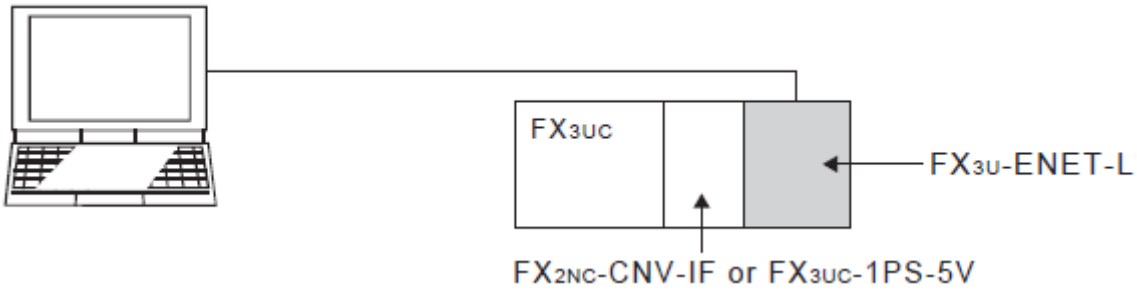
The example that DIAView software communicates with equipments by Ethernet is as follows:

➢ **Creating communication between DIAView software and Mitsubishi-FX3U-ENET-L through Ethernet:**

Example 1: Taking “FX3UC, FX3U-ENET-L, ASCII” driver as an example

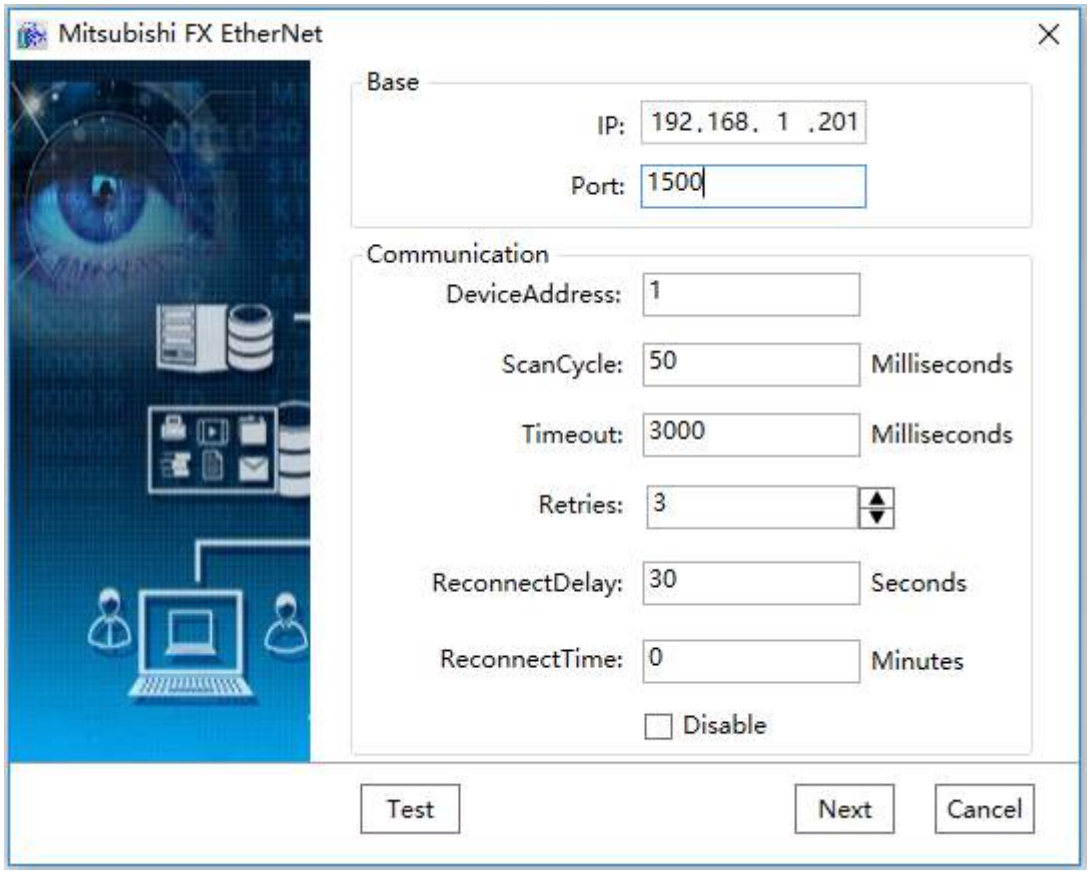
Step 1: Create the following hardware framework, set IP address as 192.168.1.8”, the computer IP address is "192.168.1.200" (keep the PLC and the computer in the same LAN):

Ethernet module	Compatible PLC	Protocol
FX3U-ENET-L	FX3UC (FX2NC-CNV-IF or FX3UC-1PS-5V is necessary.)	TCP

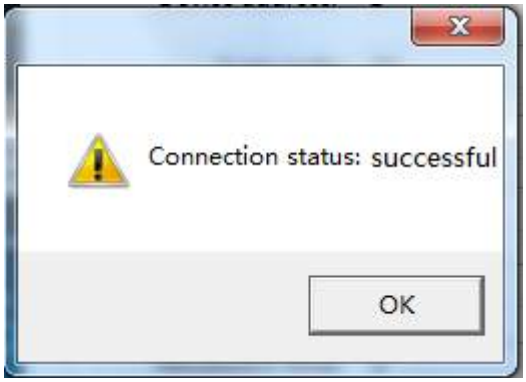


Step 2: In the project management area, right click on “IODevice” and select “New Device”, select “Mitsubishi FX EtherNet” in the driver selection window, as shown in the figure below:

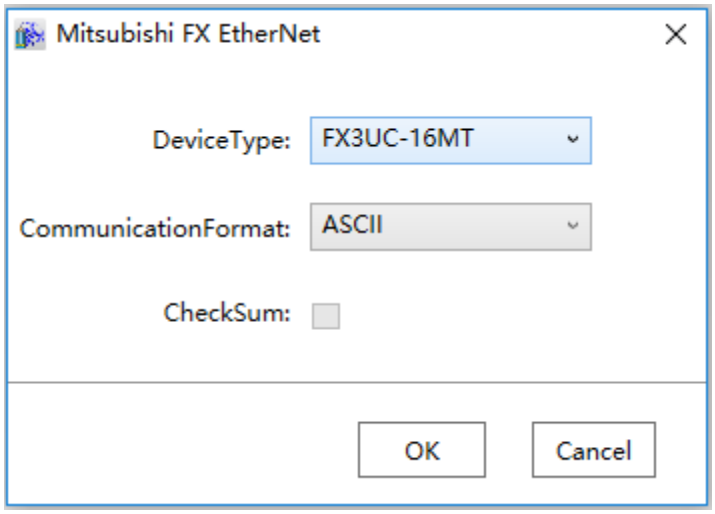
Step 3: Configure universal FX3UC, ASCII communication parameters, and set the "IP" as: 192.168.1.201 in the “Base” setting, set the port number as: 1500, as shown in the figure below:



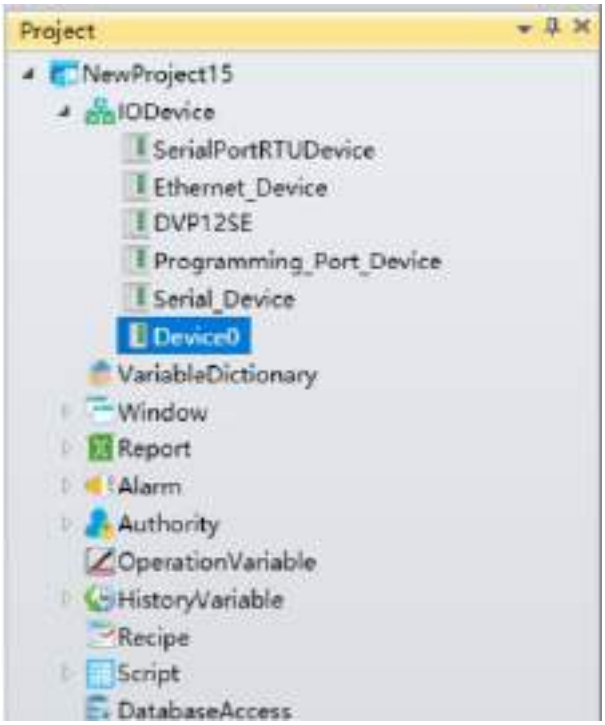
Step 4: Other parameters can be the default, click " Test "button to test the connection, as shown in the figure below:



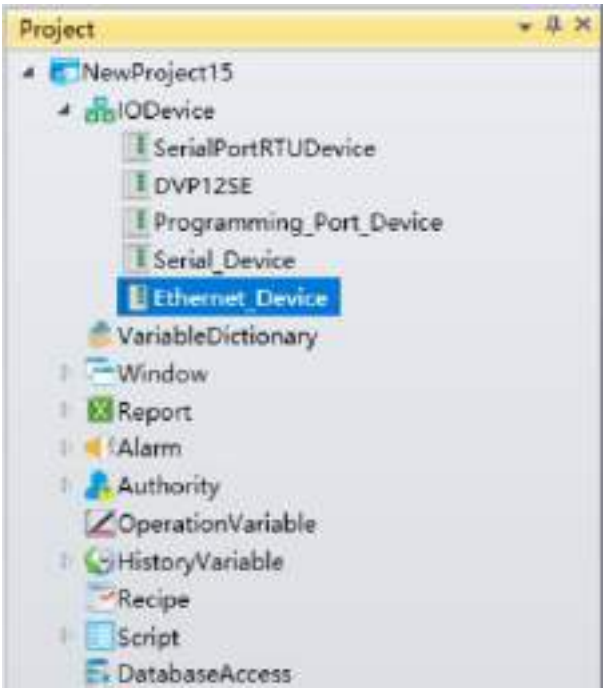
Step 5: Click "Next" to open Mitsubishi-FX communication parameters configuration window to set parameters, as shown in the figure below:



Step 6: Click the "OK" button when all parameters have been configured and the device with default name will appear under “IODevice” node of the project tree directory.



Step 7: Rename the newly-built IO communication device as "Ethernet_Device" and complete Ethernet driver communication configuration, as shown in the figure below:



➢ **Noted:** Mitsubishi FX3UC address description, as shown in the figure below:

Register	Range	Default type	Description
X	0 - 377	bool	Input register
Y	0 - 0x1FFF	bool	Output register
M	0 - 7679	bool	Auxiliary register
SM	8000 - 8511	bool	special auxiliary register
S	0 - 4095	bool	Status register
TS	0 - 511	bool	Timer contact register
CS	0 - 255	bool	Counter contact register
TN	0 - 511	word	Timer register
CN16	0 - 199	word	16 bit counter register
CN32	200 - 255	word	32 bit counter register
D	0 - 7999	word	Data register
SD	8000 - 8511	word	Special data register

R	0 - 32767	word	Extended register

5.8.5 Siemens

The PLCs produced by SIEMENS are applied widely in our country. They are used in metallurgy,chemical engineering and printing etc. field.(SIEMENS)PLC products include LOGO, S7-200, S7-1200, S7-300, S7-400 etc. Siemens S7 series PLC is with small size, fast speed and standard. In addition, it has network communication capability，stronger function and higher reliability.S7 series PLC products can be divided into micro PLC(S7-200), small-scale performance requirement (S7-300)and middle, high performance requirement PLC(S7-400)etc.

5.8.5.1 S7300 TCP

Description: S7300 TCP driver also supports S7 1200,when creating a new device, the parameter "device address" is set as 0.

Note that when using Siemens software to configure:

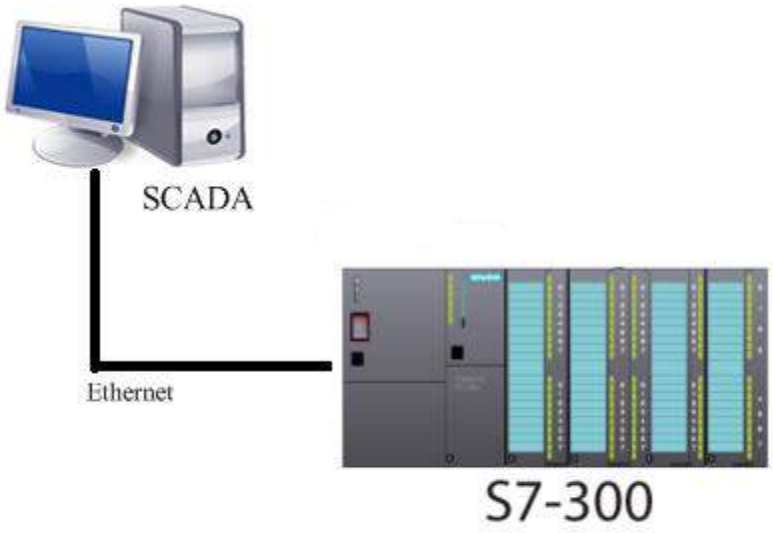
1, It must be an unoptimizable block.

2, Slot number and rack number must be 0,0.

3, Put or Get access must be enabled.

- **Creating communication between DIAView software and Siemens S7300 TCP through Ethernet:**

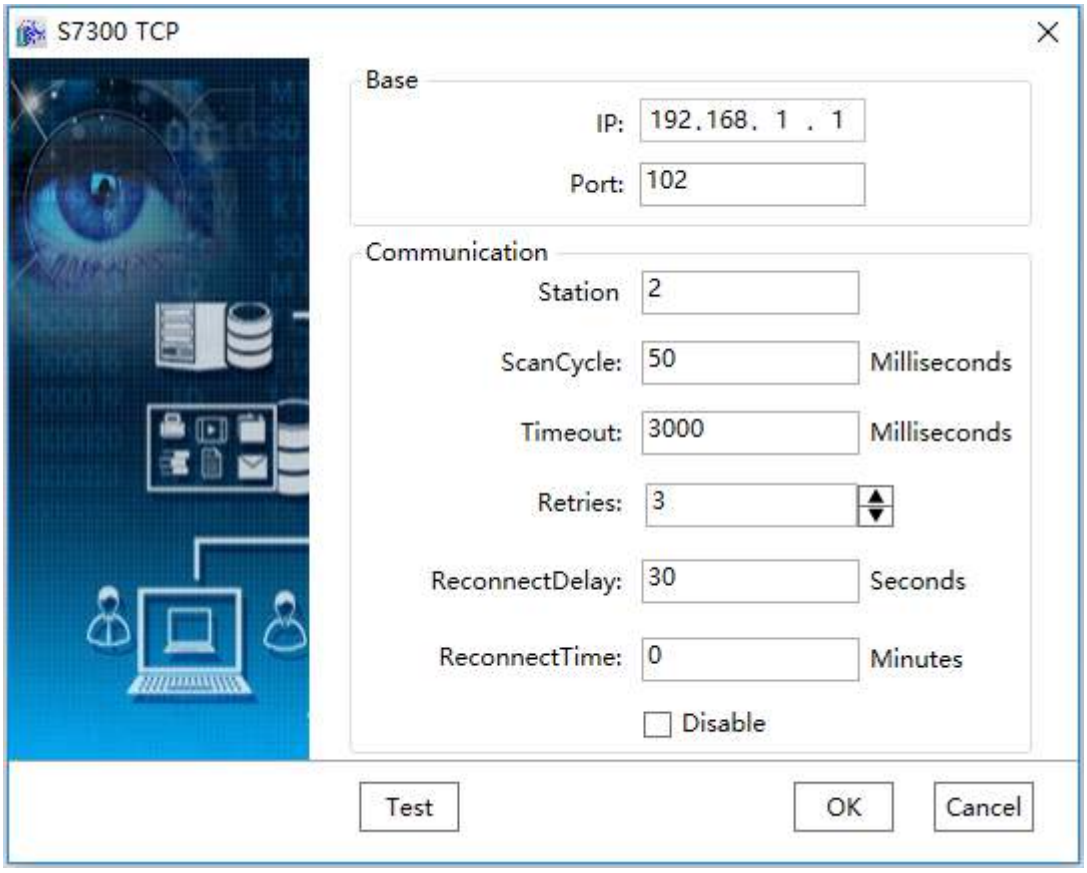
Step 1: Create the following hardware framework: Set S7300 address as “192.168.1.2”, set the computer IP address as “192.168.1.200” (keep PLC and the computer in the same LAN):



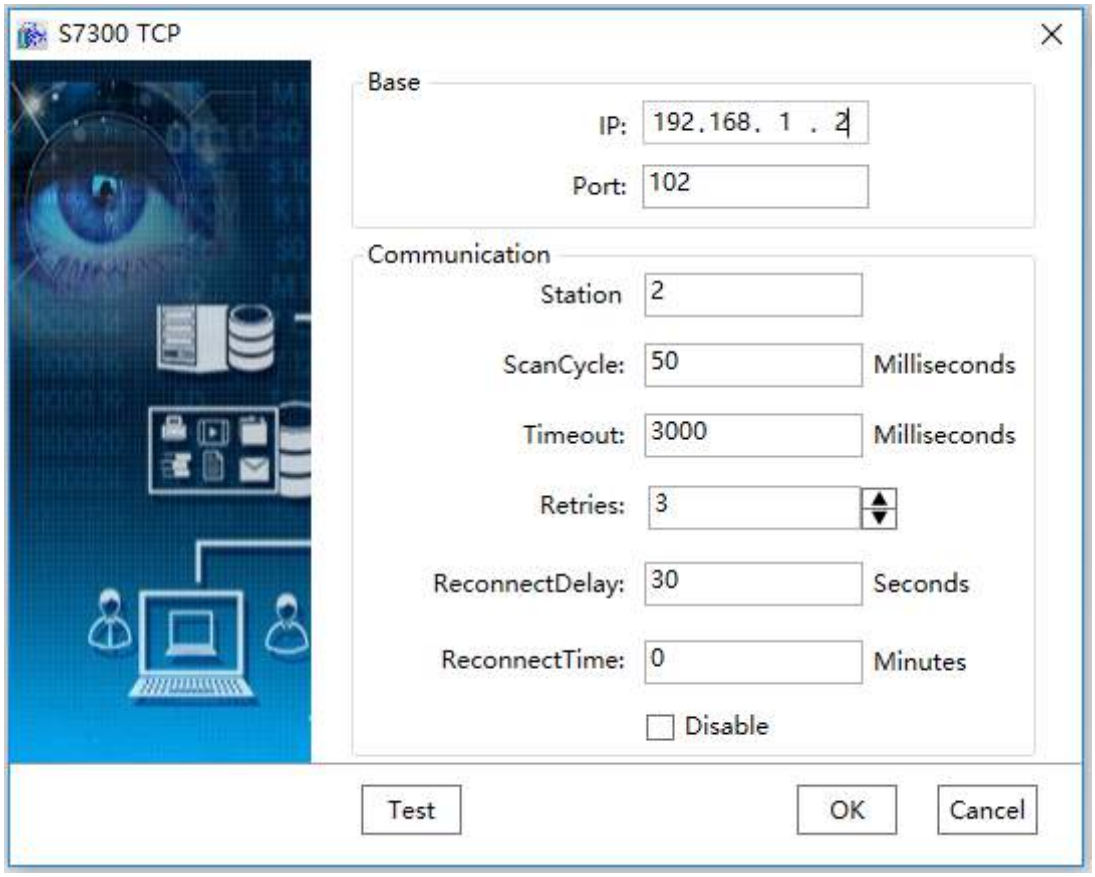
Step 2: Right-click on the “IODevice” node in the project management area and select “New Device”and pop up the driver selection menu as follows:

Step 3: Select “Siemens” →“S7300 TCP” in the menu selection, as shown in the figure below:

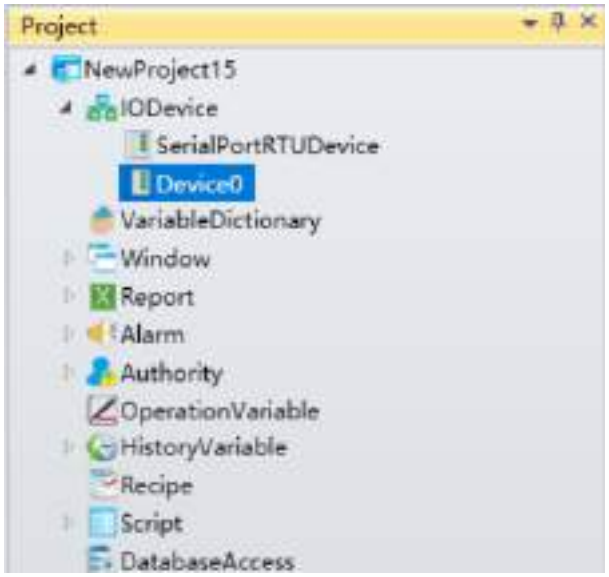
Step 4: Configure S7300 communication parameters, as shown in the figure below:



Step 5: Set the IP address in the base setting as: 192.168.1.2, set “Port” as: 102, click“test”button to test the connection ,as shown in the figure below:

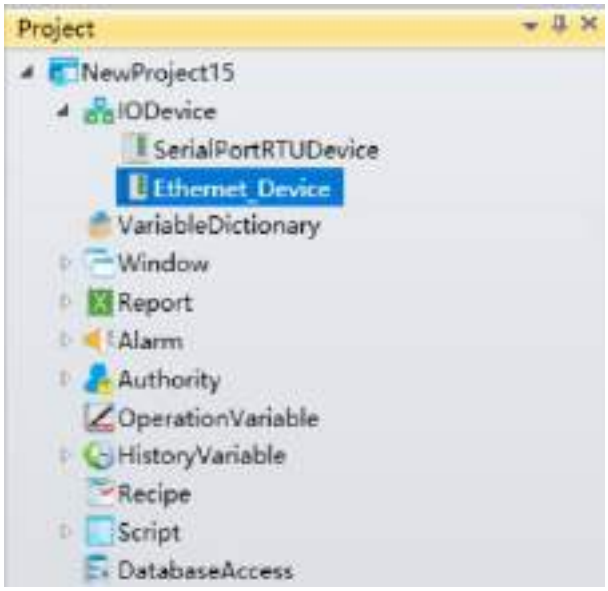


Step 6: Click the “OK” button when all parameters have been configured and the device with the default name “Device0” will appear under the “IODevice” node,as shown in the figure below:



Step 7: Rename the newly-built device as “Ethernet_Device”and complete Ethernet driver

communication configuration, as shown in the figure below:



➢ **Noted:** Siemens S7300 TCP address description, as shown in the table below:

Register	Range	Default type	Description
ID	0~65532	DWORD	Input register
IW	0~65534	WORD	Input register
IB	0~65535	BOOL	Input register
QD	0~65532	DWORD	Output register
QW	0~ 65534	WORD	Output register
QB	0~65535	BYTE	Output register
MD	0~65532	DWORD	Bit register
MW	0~65534	WORD	Bit register
MB	0~65535	BYTE	Bit register
DB	0~65535	WORD	DB data block register
T	0~65535	WORD	Timer
C	0~65535	WORD	Counter

5.8.5.2 S71200 TCP

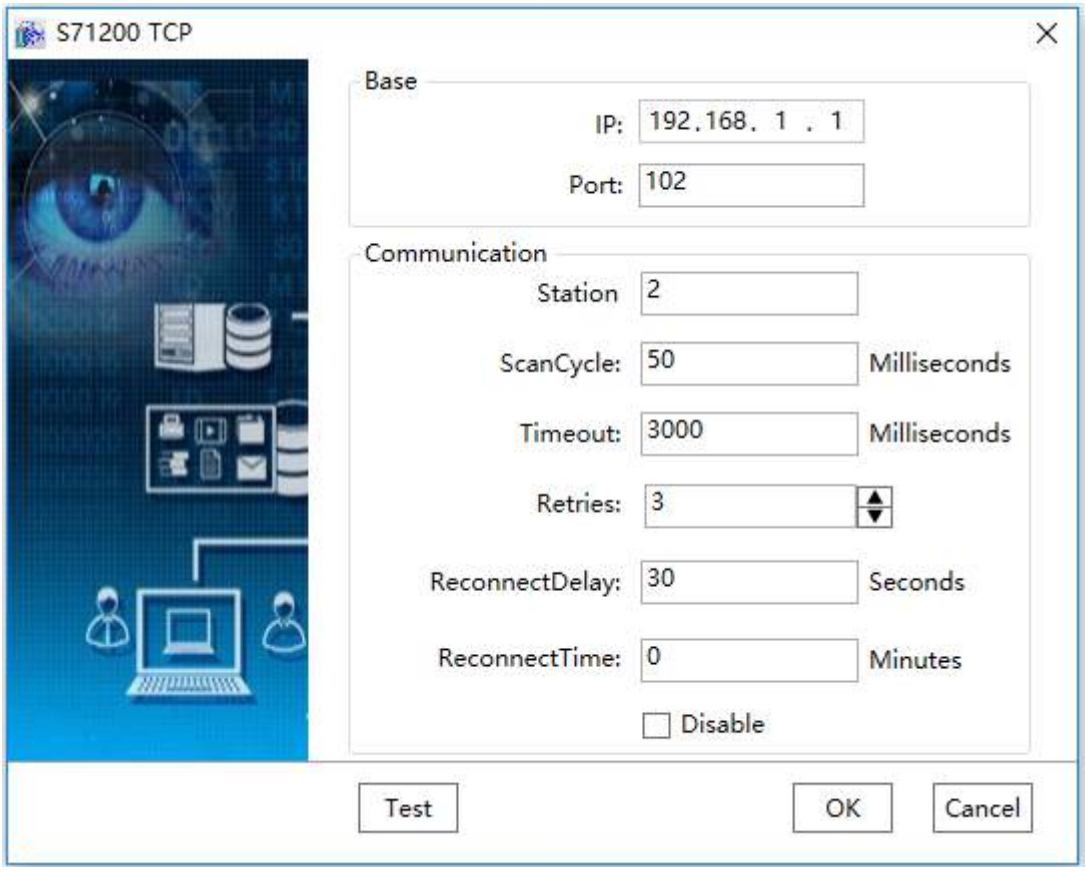
The example that DIAView software communicates with equipments by Ethernet is as follows:

➢ **Creating communication between DIAView configuration software and Siemens S71200 TCP through Ethernet:**

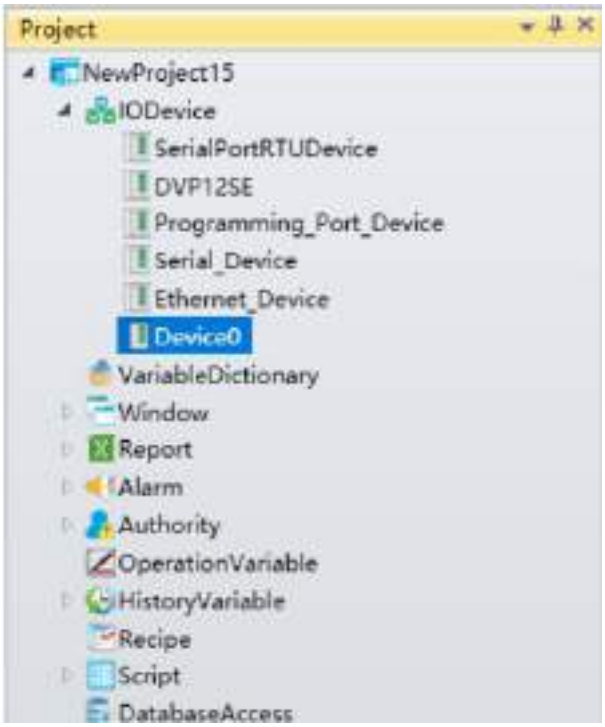
Step 1: (keep PLC and the computer in the same LAN) Right-click on the “IODevice” node in the project management area and select “New Device” then pop up the selection menu as follows:

Step 2: Select “Siemens” →“S71200 TCP” in the menu selection, as shown in the figure below:

Step 3: Configure S71200 TCP communication, as shown in the figure below:



Step 4: Click the “OK” button when all parameters have been configured and the device with the default name “Device0” will appear under the “IODevice” node, as shown in the figure below:



➢ **Noted:** Siemens S71200 TCP address description, as shown in the table below:

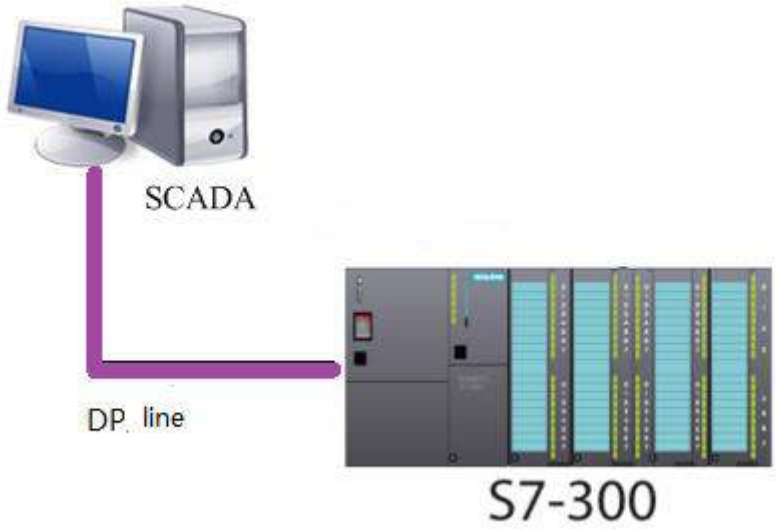
Register	Range	Default type	Description
ID	0~65532	DWORD	Input register
IW	0~65534	WORD	Input register
IB	0~65535	BOOL	Input register
QD	0~65532	DWORD	Output register
QW	0~ 65534	WORD	Output register
QB	0~65535	BYTE	Output register
MD	0~65532	DWORD	Bit register
MW	0~65534	WORD	Bit register
MB	0~65535	BYTE	Bit register
DB	0~65535	WORD	DB data block register

T	0~65535	WORD	Timer
C	0~65535	WORD	Counter

5.8.5.3 S7300 MPI

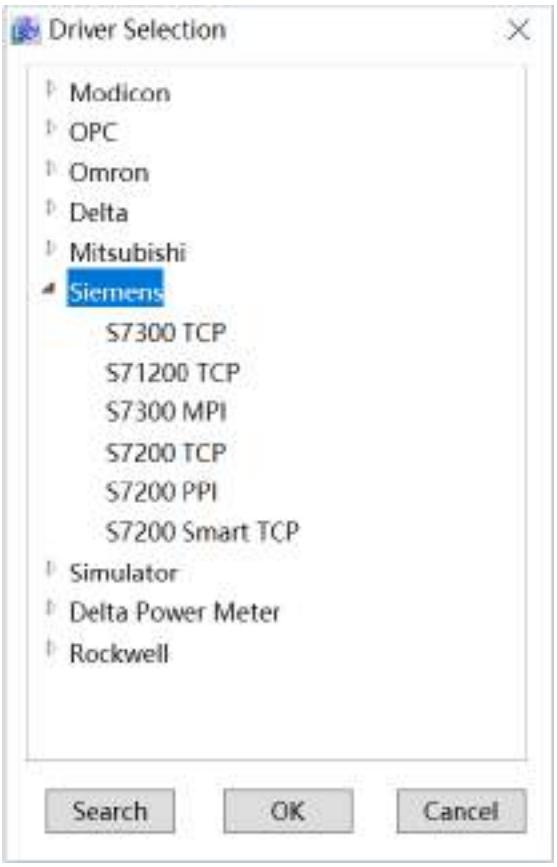
➢ **Creating communication between DIAView software and Siemens S7300 MPI by Ethernet:**

Step 1: Set up the following hardwork framework, as shown in the figure below:



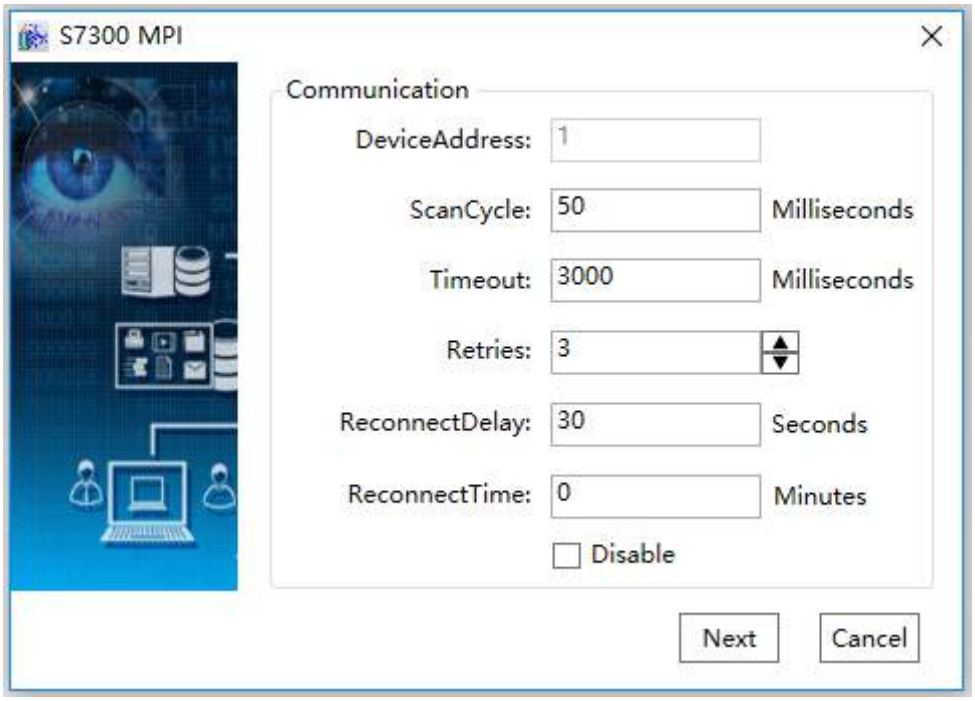
The DP connection line is as follows:

Step 2: Right-click on the "IODevice"node in the project management area and select "New Device", the driver selection menu in the figure below will appear:

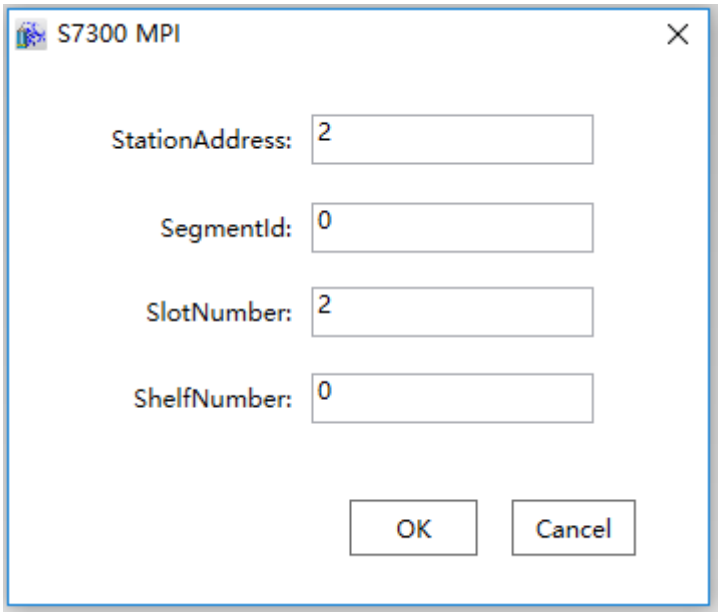


Step 3: Select “Siemens” → “S7300 MPI”in the driver selection, as shown in the figure below:

Step 4: Configure Simens S7300 MPI communication parameters, as follows:



Step 5: Set“Station address”as: 2, “Slot numbei”as: 2, as shown in the figure below:



➢ **Noted:** Siemens S7300 MPI address description, as shown in the table below:

Register	Range	Default type	Description
ID	0~65532	DWORD	Input register

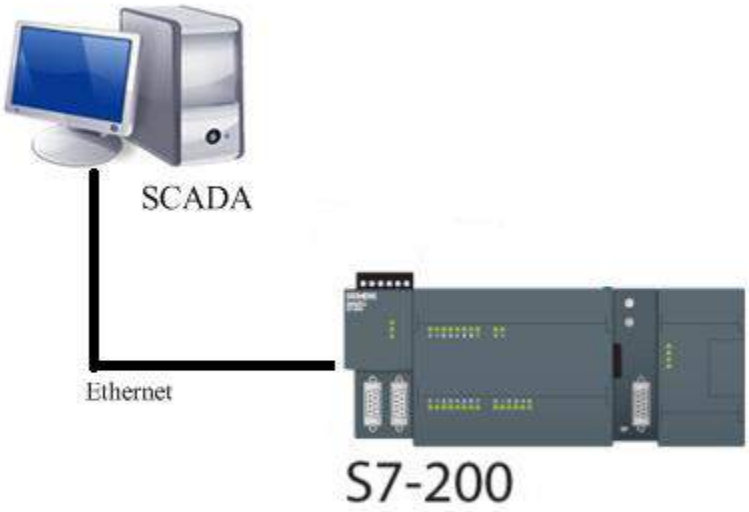
IW	0~65534	WORD	Input register
IB	0~65535	BOOL	Input register
QD	0~65532	DWORD	Output register
QW	0~ 65534	WORD	Output register
QB	0~65535	BYTE	Output register
MD	0~65532	DWORD	Bit register
MW	0~65534	WORD	Bit register
MB	0~65535	BYTE	Bit register
DB	0~65535	WORD	DB data block register
T	0~65535	WORD	Timer
C	0~65535	WORD	Counter

5.8.5.4 S7200 TCP

The example that DIAView software communicates with equipments by Ethernet is as follows:

➢ **Creating communication between DIAView configuration software and Siemens S7200 TCP through Ethernet:**

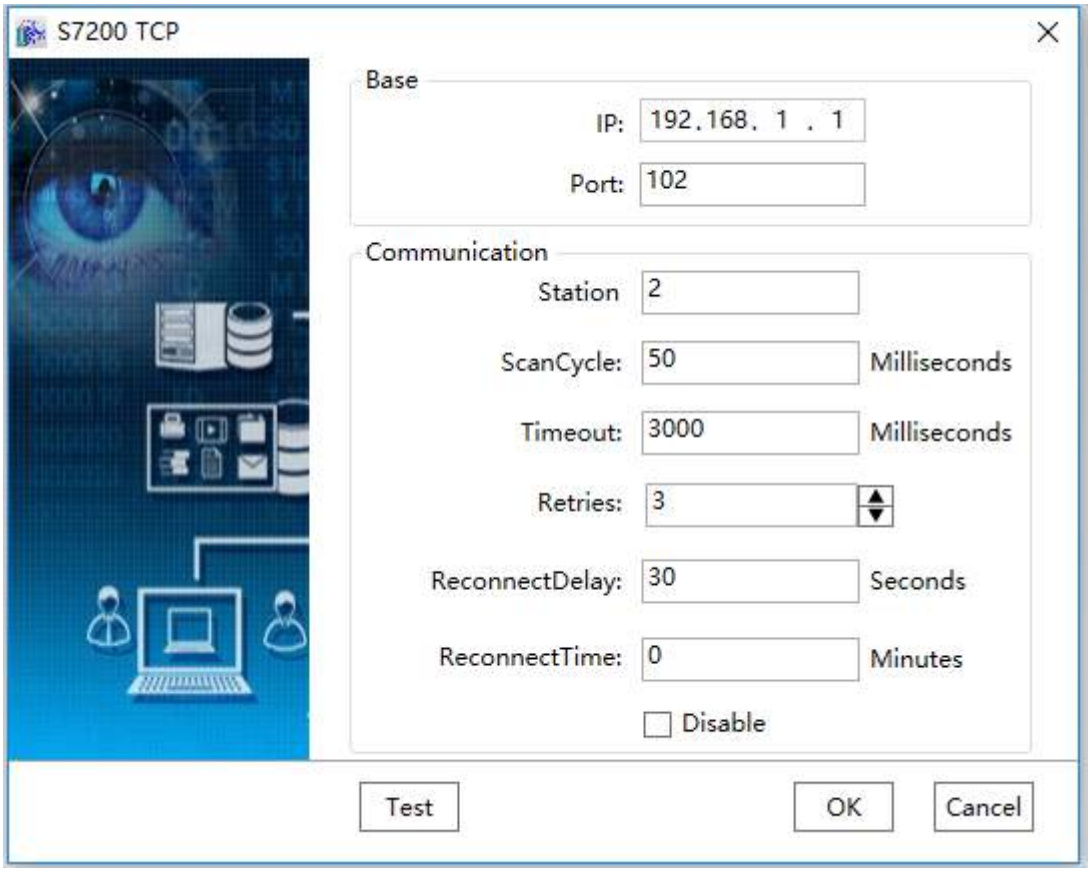
Step 1: Create the following hardware framework as shown in the figure below: set S7200 address as “192.168.1.3”, the computer IP is “192.168.1.200” (keep PLC and the computer in the same LAN):



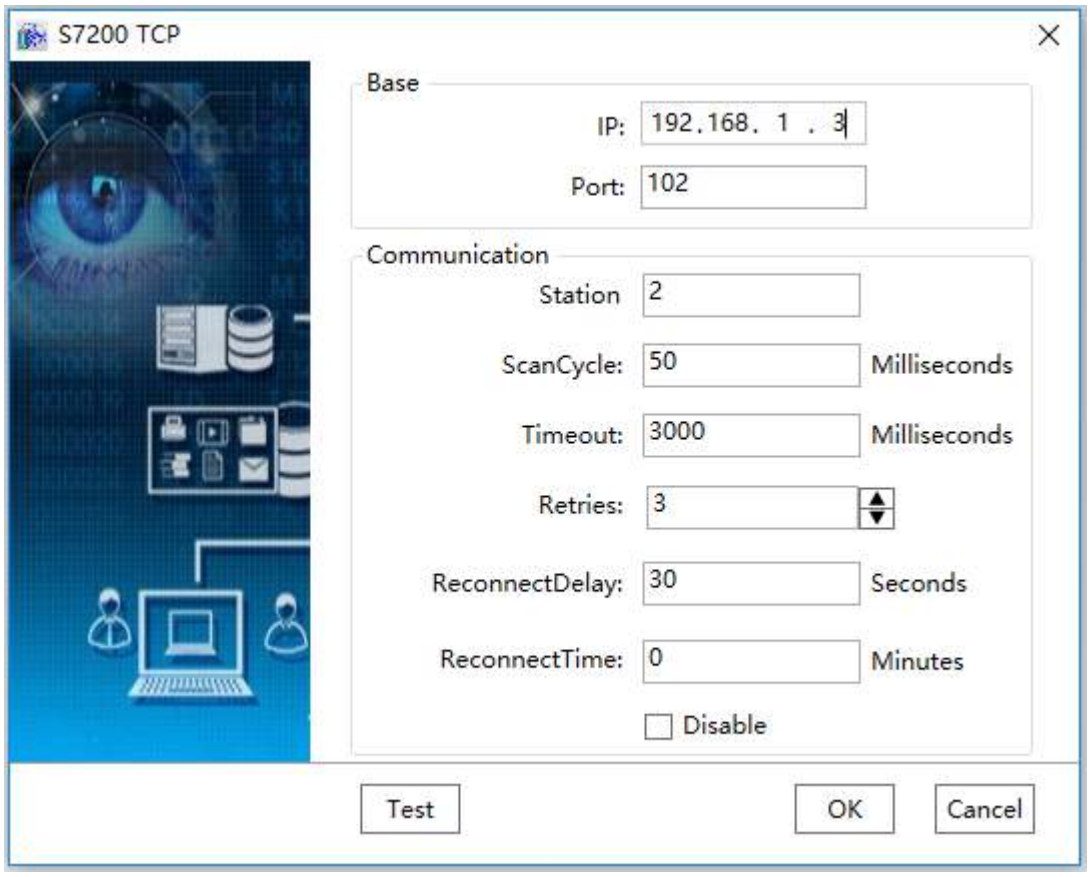
Step 2: Right-click on the “IODevice” node in the project management area and select “New Device” then pop up the selection menu as follows:

Step 3: Select “Siemens” →“S7200 TCP” in the menu selection, as shown in the figure below:

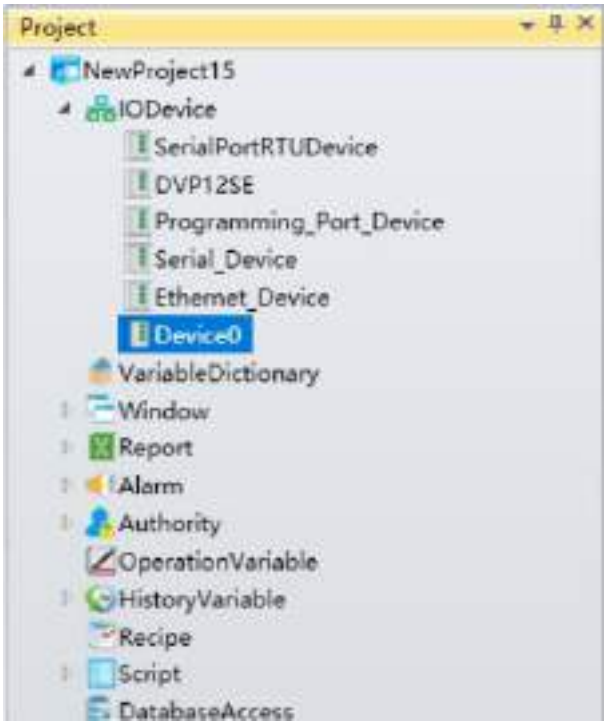
Step 4: Configure S7200 TCP communication, as shown in the figure below:



Step 5: Set the “IP” in the “Base” setting as: 192.168.1.3, set “Port” as: 102, Click“Test”button to test the connection, as shown in the figure below:



Step 6: Click the “OK” button when all parameters have been configured and the device with the default name “Device 0” will appear under the “IODevice” node, as shown in the figure below:



➢ **Noted:** Siemens S7200 TCP address description, as shown in the table below:

Register	Range	Default type	Description
ID	0~65532	DWORD	Input register
IW	0~65534	WORD	Input register
IB	0~65535	BYTE	Input register
QD	0~65532	DWORD	Output register
QW	0~ 65534	WORD	Output register
QB	0~65535	BYTE	Output register
MD	0~65532	DWORD	Bit register
MW	0~65534	WORD	Bit register
MB	0~65535	BYTE	Bit register
VD	0~65532	DWORD	V data block register
VW	0~65534	WORD	V data block register
VB	0~65535	BYTE	V data block register

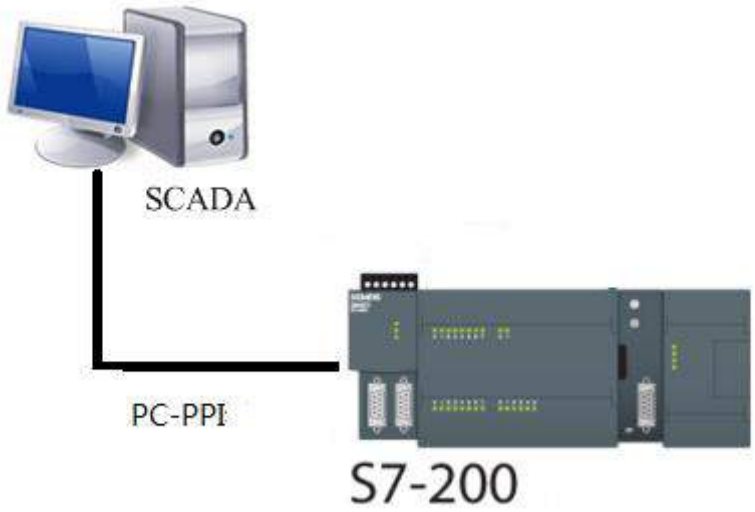
T	0~32767	WORD	Timer
C	0~32767	WORD	Counter

5.8.5.5 S7200 PPI

The example that DIAView software communicates with equipments by serial port is as follows:

➢ **Creating communication betweeen DIAView software and Siemens S7200 PPI through serial port:**

Step 1: Create the following hardware framework as shown in the figure below:

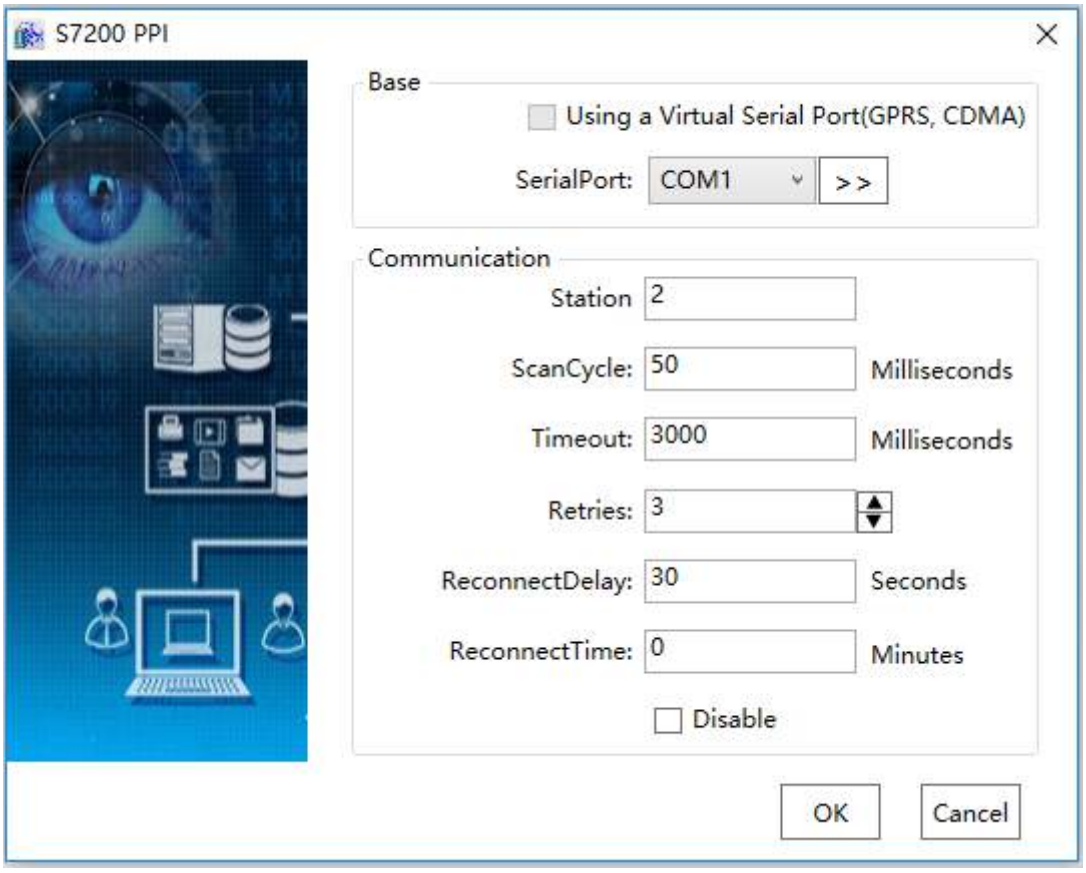


PC-PPI line is as follows:

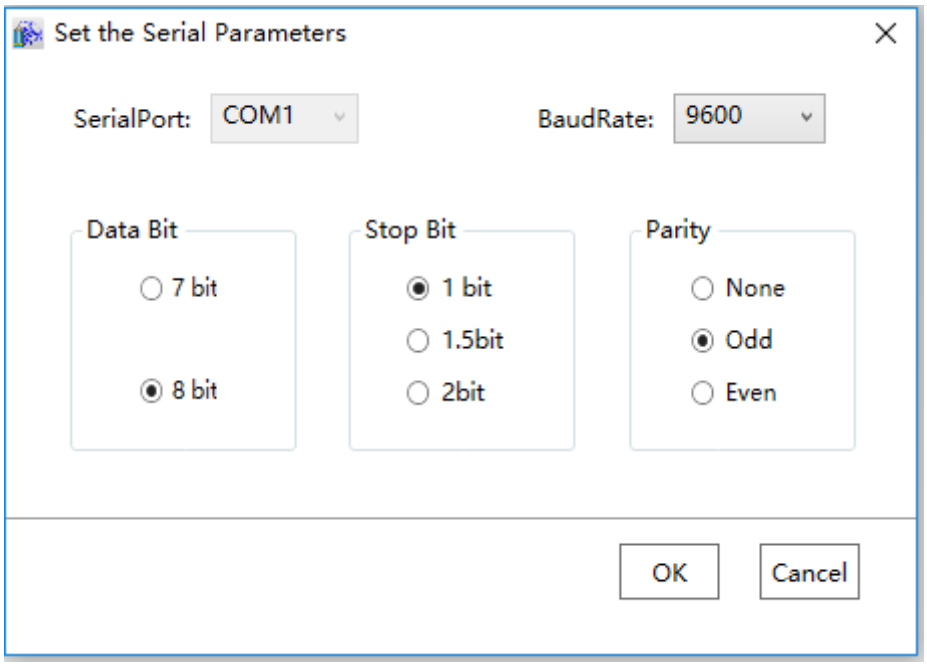
Step 2: Right-click on the “IODevice” node in the project management area and select “New Device” then pop up the driver selection menu as follows:

Step 3: Select “Siemens” →“S7200 PPI” in the menu selection, as shown in the figure below:

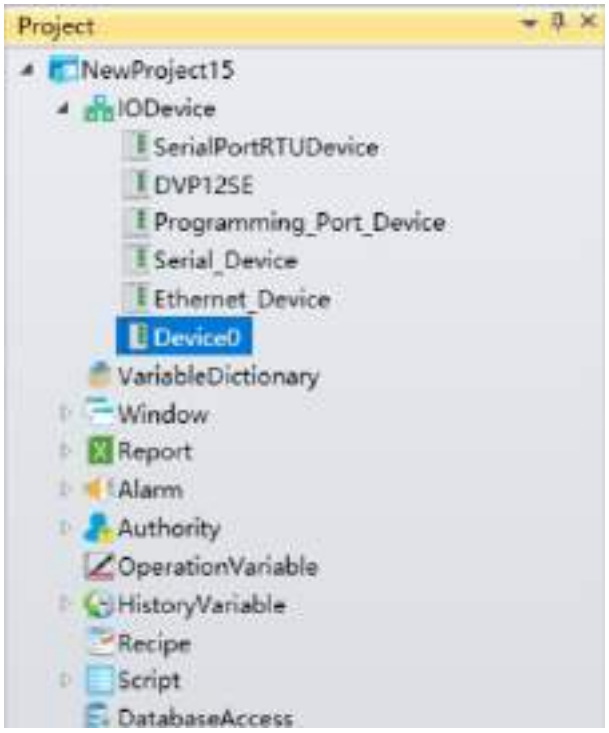
Step 4: Configure Siemens S7200 PPI communication parameters, as shown in the figure below:



Step 5: Click the button beside "SerialPort"in the “Base”setting and pop up the interface as follows:



Step 6: Click the “OK” button when all parameters have been configured and the device with the default name “Device0” will appear under the “IODevice” node as shown in the figure below:



- **Noted:** Siemens S7200 PPI address description as shown in the table below:

Register	Range	Default type	Description
ID	0~9996	DWORD	Input register
IW	0~9998	WORD	Input register
IB	0~9999	BYTE	Input register
QD	0~9996	DWORD	Output register
QW	0~9998	WORD	Output register
QB	0~9999	BYTE	Output register
MD	0~9996	DWORD	Bit register
MW	0~9998	WORD	Bit register
MB	0~9999	BYTE	Bit register
VD	0~65532	DWORD	V data block register
VB	0~65535	BYTE	V data block register
VW	0~65534	WORD	V data block register
C	0~32767	WORD	Counter
T	0~32767	WORD	Timer

5.8.5.6 S7200 Smart TCP

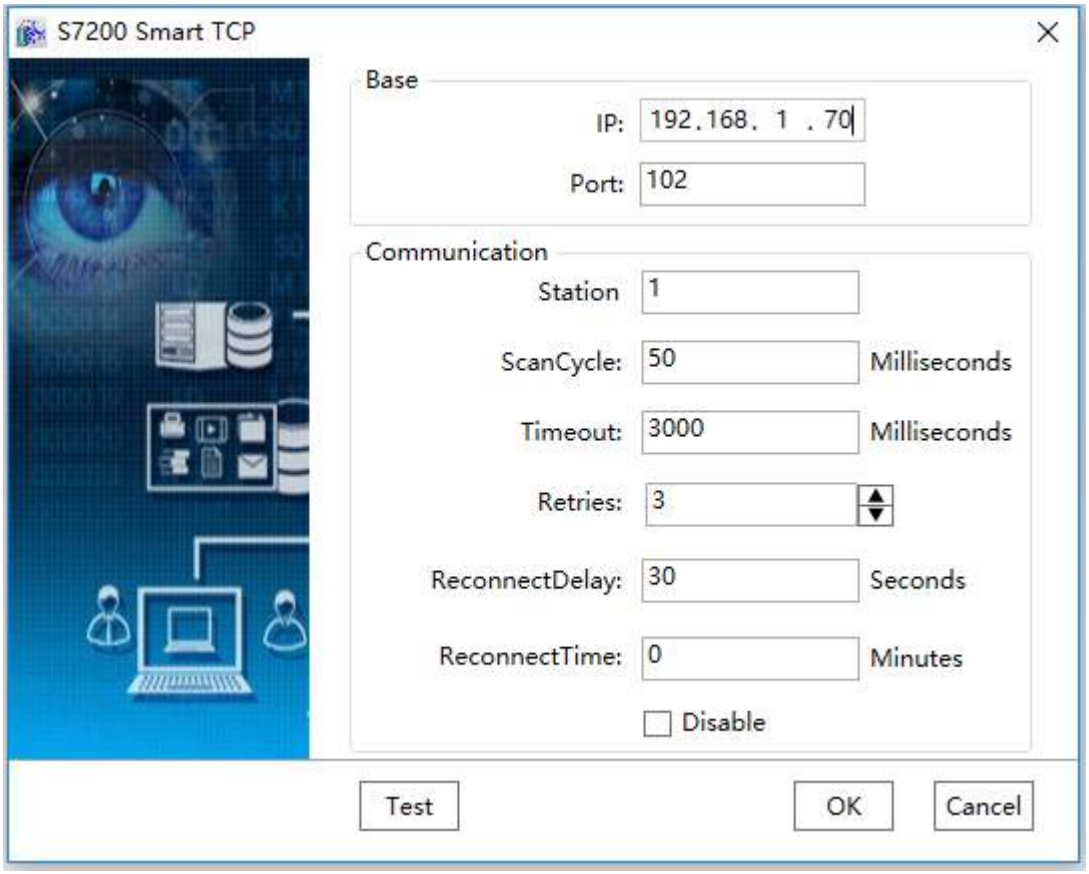
The example that DIAView software communicates with equipments by Ethernet is as follows:

➢ **Creating communication between DIAView configuration software and Siemens S7200 Smart TCP through Ethernet:**

Step 1:(keep PLC and the computer in the same LAN)Right-click on the “IODevice” node in the project management area and select “Add device” then pop up the selection menu as follows:

Step 2: Select “Siemens” →“S7200 Smart TCP” in the menu selection, as shown in the figure below:

Step 3: Configure S7200 Smart TCP communication, as shown in the figure below:



➢ **Noted:** Siemens S7200 Smart TCP address description, as shown in the table below:

Register	Range	Default value type	Instructions
ID	0~65532	DWORD	Input register
IW	0~65534	WORD	Input register
IB	0~65535	BYTE	Input register
QD	0~65532	DWORD	Output register
QW	0~ 65534	WORD	Output register
QB	0~65535	BYTE	Output register
MD	0~65532	DWORD	Bit register

MW	0~65534	WORD	Bit register
MB	0~65535	BYTE	Bit register
VD	0~65532	DWORD	V data block register
VW	0~65534	WORD	V data block register
VB	0~65535	BYTE	V data block register
T	0~32767	WORD	Counter
C	0~32767	WORD	Timer

5.8.6 Delta power meter

DIAView software supports the communication of Delta power meter.

Supporting device: DPM-C530 series.

Supporting communication protocol: TCP/IP protocol, RS-485 protocol.

Supporting communication network interface: Ethernet and serial.

5.8.6.1 DPM-C530 Serial

DIAView software supports the communication of the equipment based on MODBUS RTU communication protocol standard by RS-485.

Supporting device: Delta Power Meter DPM-C530。

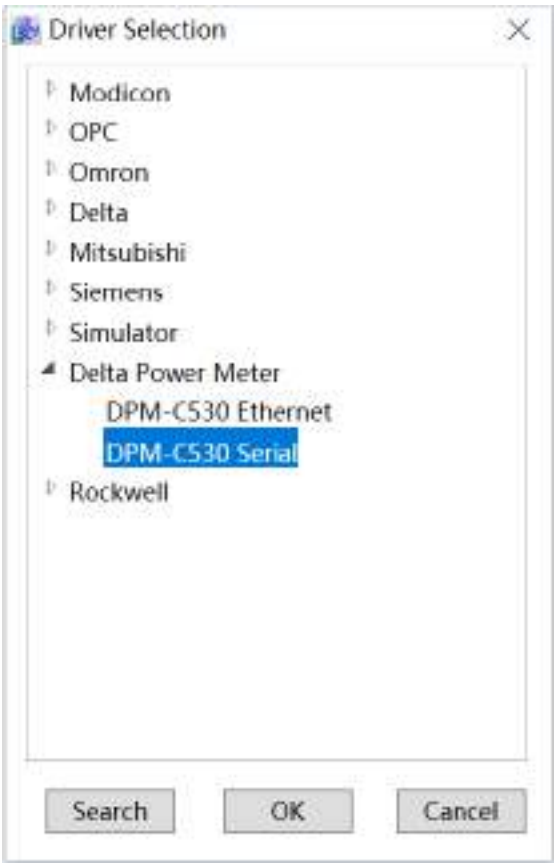
The Example DIAView software communicates with equipments by RS-485 is as follows:

➢ **Creating communication between DIAView software and Delta power meter DMP-C530 through RS-485.**

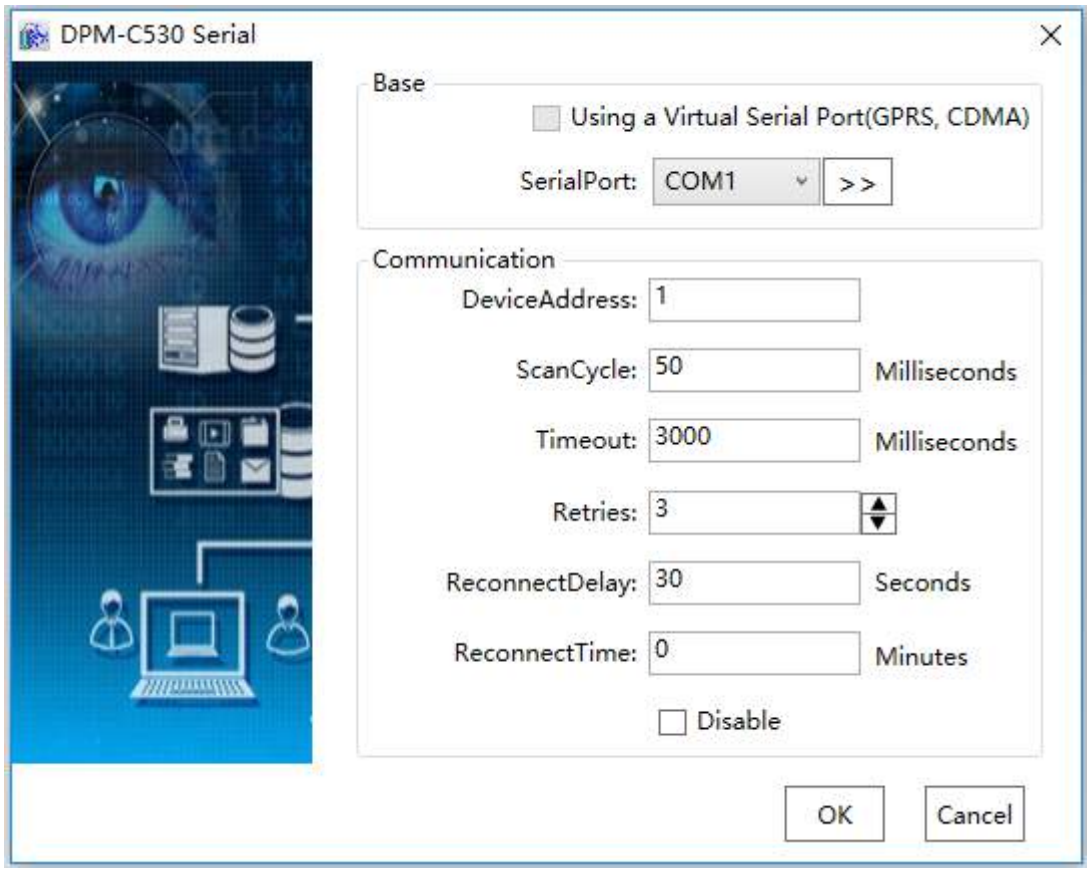
Example 1: Taking “DPM-C530, RS-485”driver as an example

Step 1: Create the following hardware framework, set COM1 communication port as 9600,8,E,1,m,as shown in the figure below:

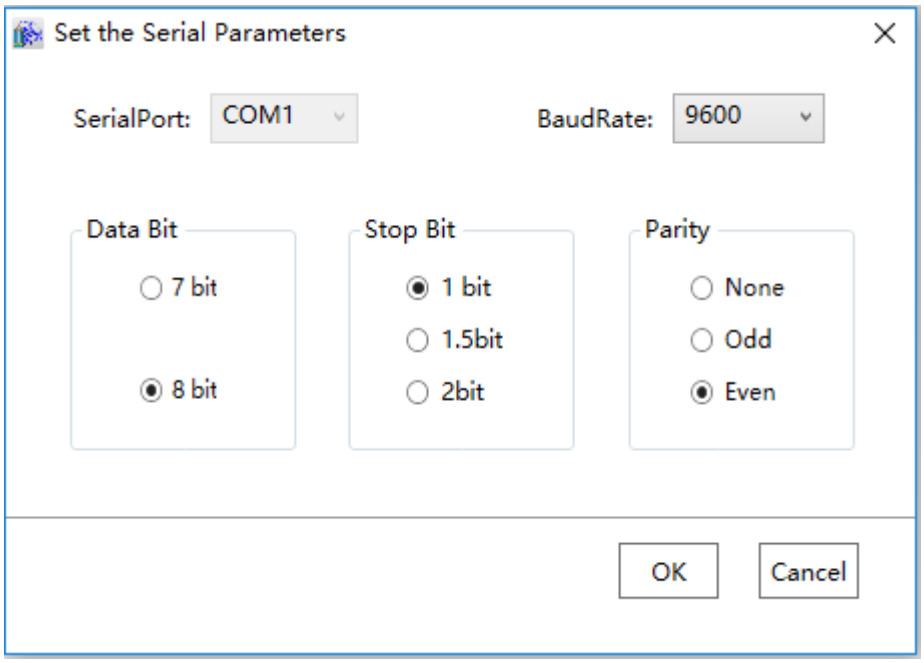
Step 2: In the project management area, right click on “IODevice“ node and select “ New Device “,select "Delta Power Meter"→"DPM-C530 Serial" in the driver selection window, as shown in the figure below:



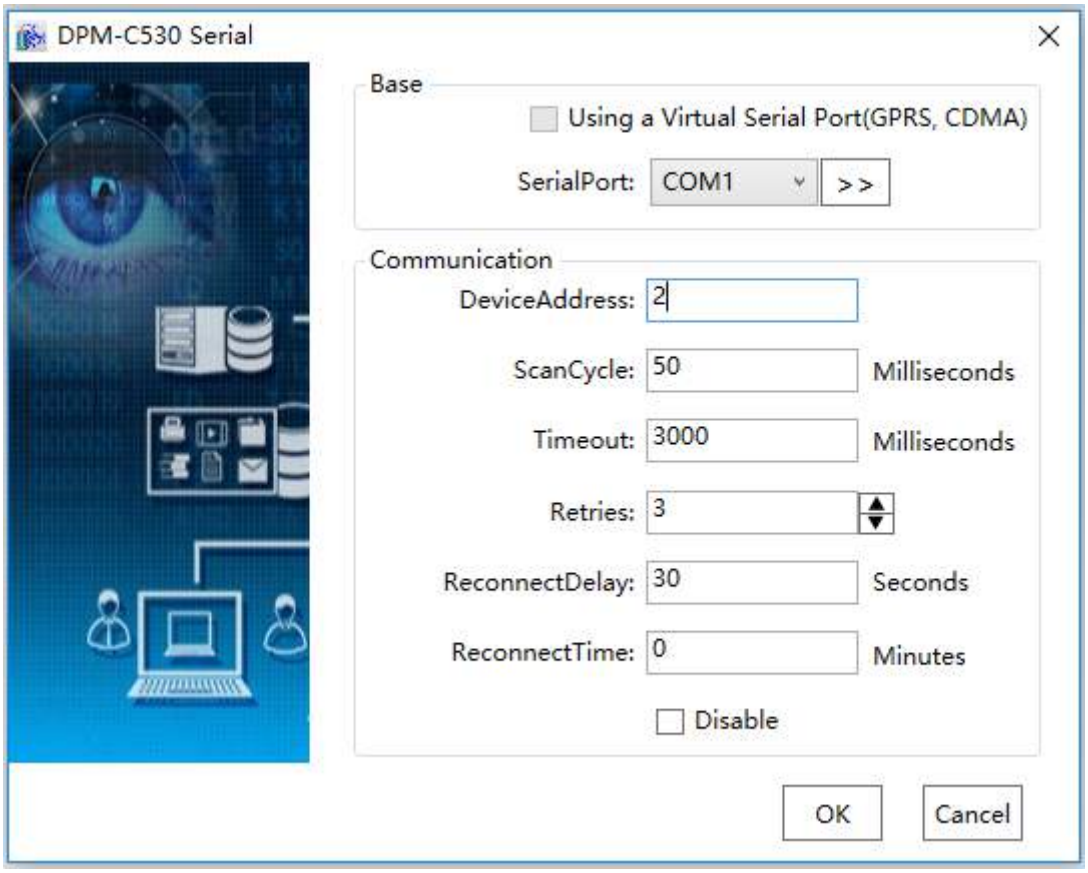
Step 3: Configure DPM-C530, RS-485 communication parameters as shown in the figure below:



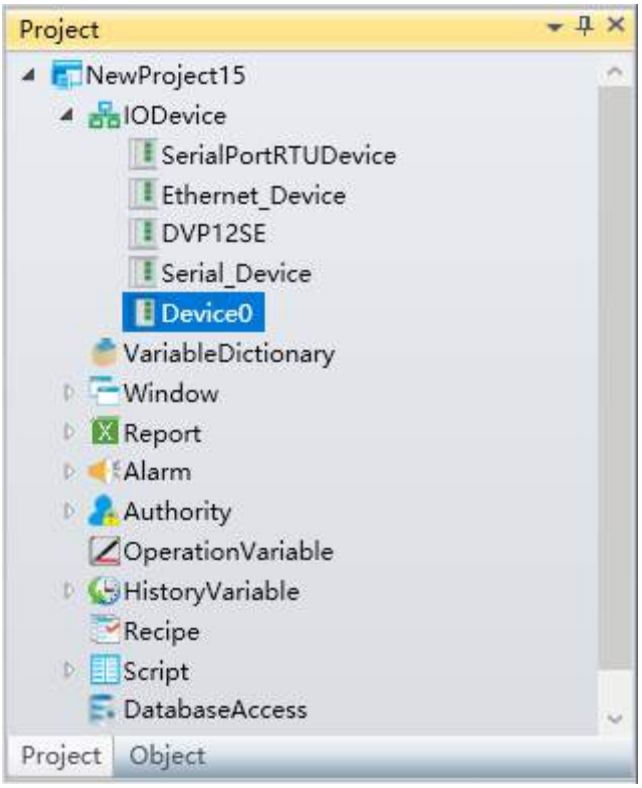
Step 4: Click“Base”→ button on the right of "SerialPort" , open "Set the Serial Parameters" window to set serial port communication parameter:"9600,8,E,1",communication port is COM1（It can be detected in the computer device manager that the current serial communication port allocated is COM1), as shown in the figure below:



Step 5: Click the "OK" button when the parameters have been configured to return to communication parameter configuration window and set communication and troulbeshooting parameter. "DeviceAddress " should be the same with DPM-C530 station number,set it as "2", other option can be default, as shown in the figure below:



Step 6: Click the "OK" button when all parameters have been configured and the device with default name "Device0" will appear under “IODevice” node of the project tree directory:



➢ **Noted:** Delta power meter DPM-C530 address description, please refer to DMP-C530 manual.

5.8.6.2 DPM-C530 Enternet

DIAView software supports the communication of the equipment based on MODBUS RTU communication protocol standard by Ethernet to transform delta IFD9506 to RS-485.

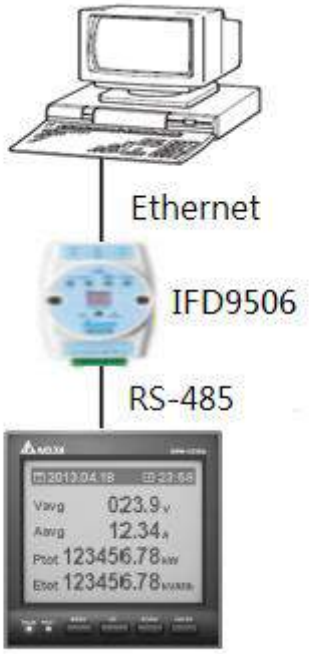
Supporting device: Delta power meter DPM-C530.

The example DIAView software communicates with equipments by Ethernet is as follows:

➢ **Creating communication between DIAView software and Delta power meter DPM-C530 through Ethernet:**

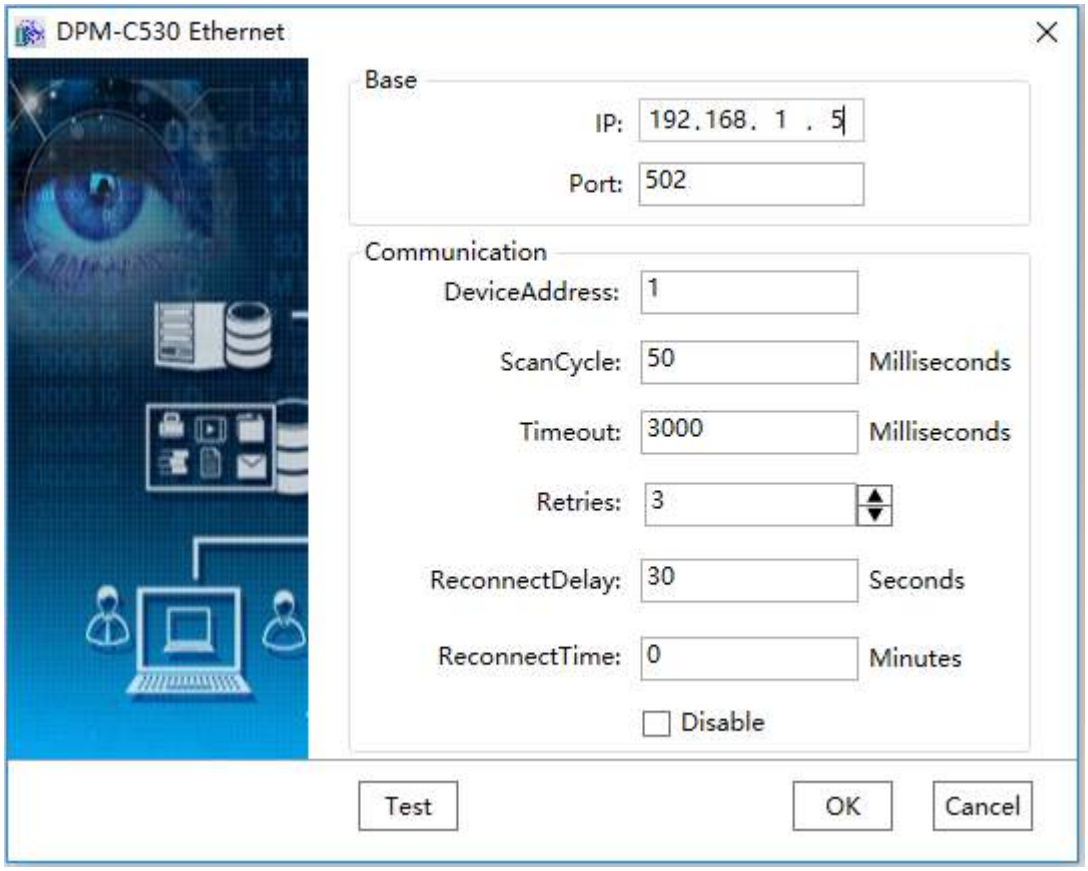
Example 1: Taking“DPM-C530, Ethernet”driver as an example

Step 1: Create following hardware framework, set IP address as “192.168.1.5”, the computer IP address is “192.168.1.200” (keep PLC and the computer in the same LAN),as shown in the figure below:

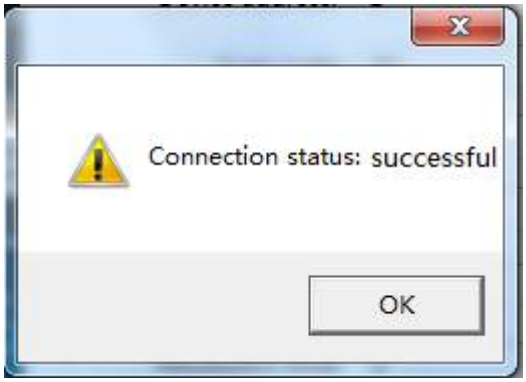


Step 2: In the project management area, right click on "IODevice" node and select "NewDevice", select "Delta Power Meter"→“DPM-C530 Ethernet”in the driver selection window, as shown in the figure below:

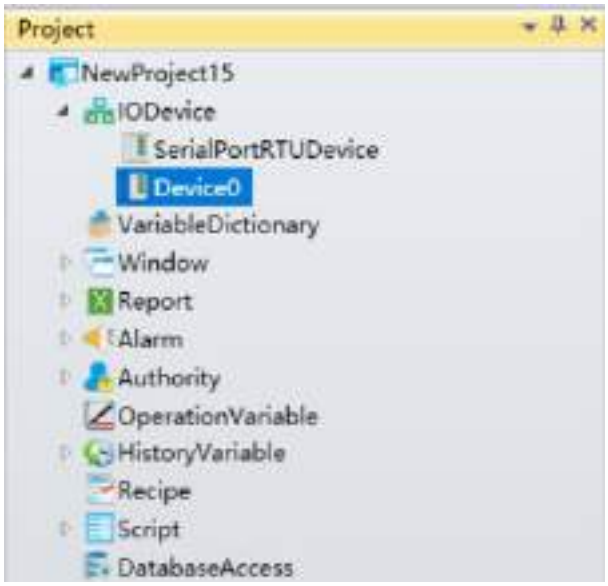
Step 3: Configure DPM-C530 Ethernet communication parameter and set the "IP address" as: 192.168.1.5, “port number” is set as : 502, as shown in the figure below:



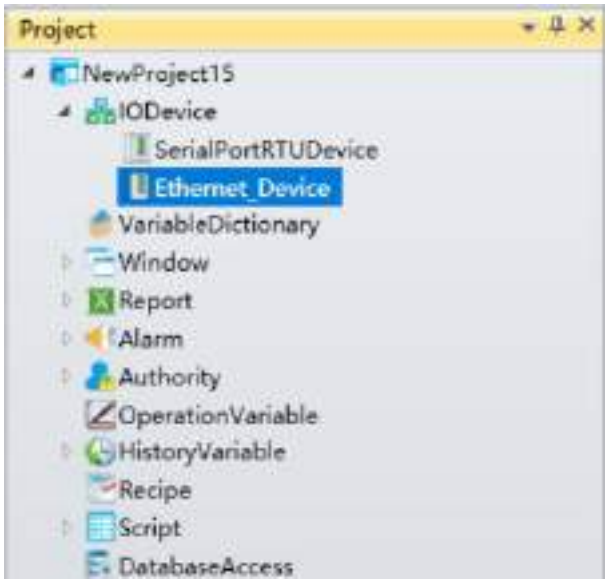
Step 4: Other option can be default and click "Test" button to test whether the connection is OK, as shown in the figure below:



Step 5: Click the "OK" button when the parameters have been configured and the device with default name "Device0" will appear under "IODevice" node of the project tree directory, as shown in the figure below:



Step 6: Rename the newly-built IO communication device as "Ethernet_Device" and complete Ethernet driver communication configuration, as shown in the figure below:



- **Noted:** Delta power meter DPM-C530 address description please refer to DPM-C530 manual.

5.8.7 Rockwell

DIAView software supports the communication of Rockwell PLC.

Supporting device: ControlLogix series, CompactLogix series, MicroLogix series, SLC500 series.

Supporting communication protocol: EtherNet/IP protocol.

Supporting communication network interface: Ethernet etc.

5.8.7.1 Rockwell Controllogix Enternet

DIAView software supports the communication of the equipment based on Rockwell communication protocol standard by EtherNet/IP.

Supporting devices: based on 1756-EN2TR Ethernet module.

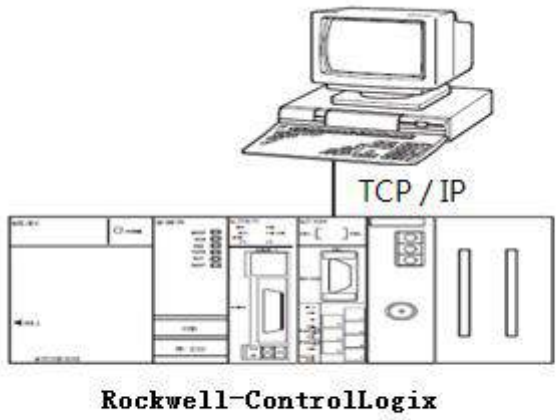
ControlLogix 5550 / 5553 / 5555 / 5561 / 5562 / 5563 / 5564 / 5565 / 5571 / 5572 / 5573 / 5574 / 5575 processors.

The example that DIAView software communicates with equipments by Ethernet is as follows:

➢ **Creating communication between DIAView software and Rockwell-ControlLogix by Ethernet:**

Example 1: Taking “Controllogix EtherNet”driver as an example

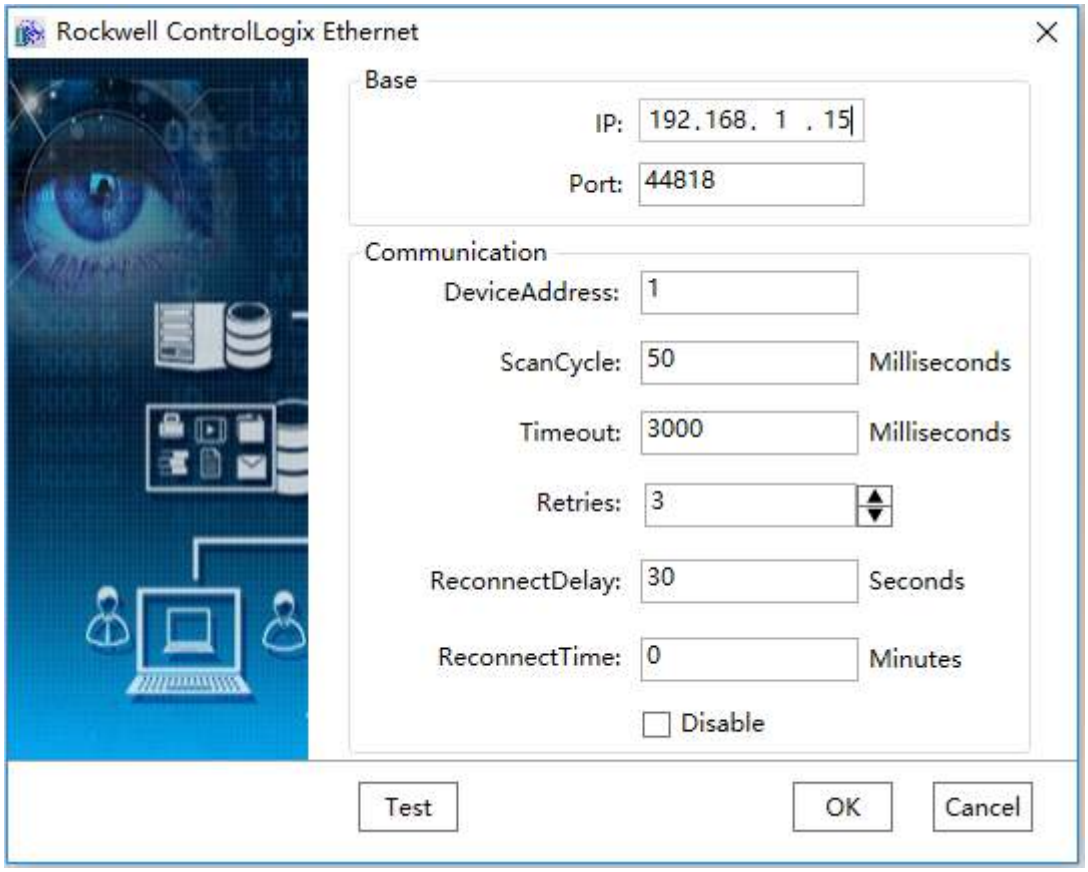
Step 1: Create the following hardware framework, Set the IP address as "192.168.1.20", the computer IP address is "192.168.1.200" (keep the PLC and the computer in the same LAN):



Step 2: In the project management area, right click on “IODevice “ node and select “ New Device “,select "Rockwell" →"Rockwell ControlLogix Ethernet" in the driver selection window, as shown in the

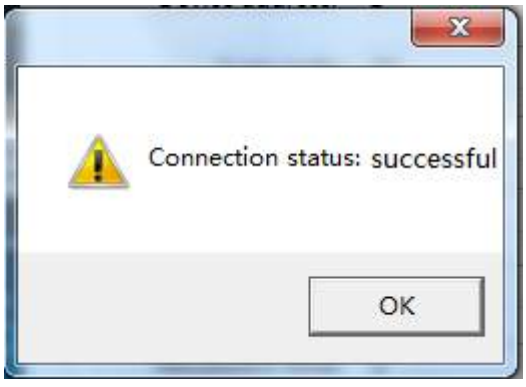
figure below:

Step 3: Configure the ControlLogix Ethernet communication parameters and set the "IP" in the base settings as 192.168.15, set the port number to: 44818, as shown in the figure below:

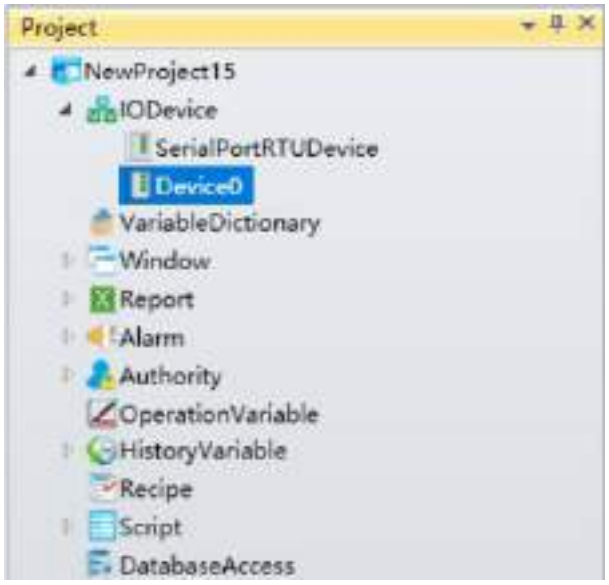


Step 4: Click "Test" button to test whether the connection is ok, as shown in the figure below:

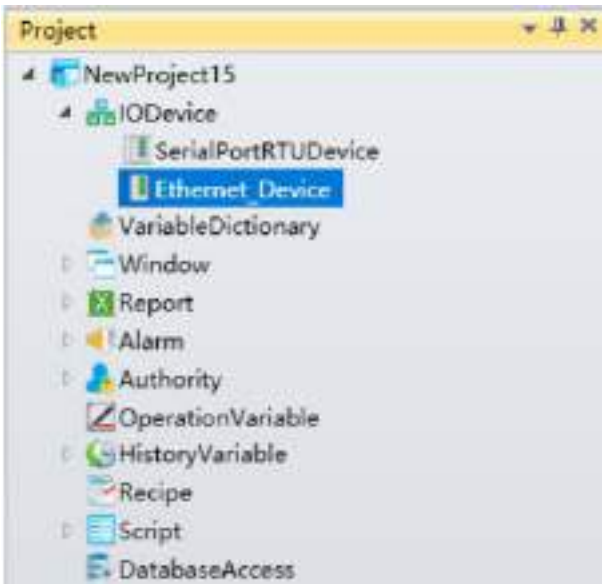
IP:192.168.1.15,Port: 44818



Step 5: Click the "OK" button when all parameters have been configured and the device with default name "Device0" will appear under “IODevice” node of the project tree directory:



Step 6: Rename the newly-built IO communication device as "Ethernet_Device" and complete Ethernet driver communication configuration, as shown in the figure below:



- **Noted:** Rockwell Controllogix EtherNet register type description as shown in the table below:

Register	Data type	PLC data type	Unit length
Base	BYTE	SINT	1
	WORD	INT	2
	DWORD	DINT	4
	FLOAT	REAL	4

	STRING	STRING	4
	BOOL	BOOL	1
Array	BYTE	SINT	1
	WORD	INT	2
	DWORD	DINT	4
	FLOAT	REAL	4
	STRING	STRING	4

5.9 OPC

DIAView software supports OPC relative protocol.

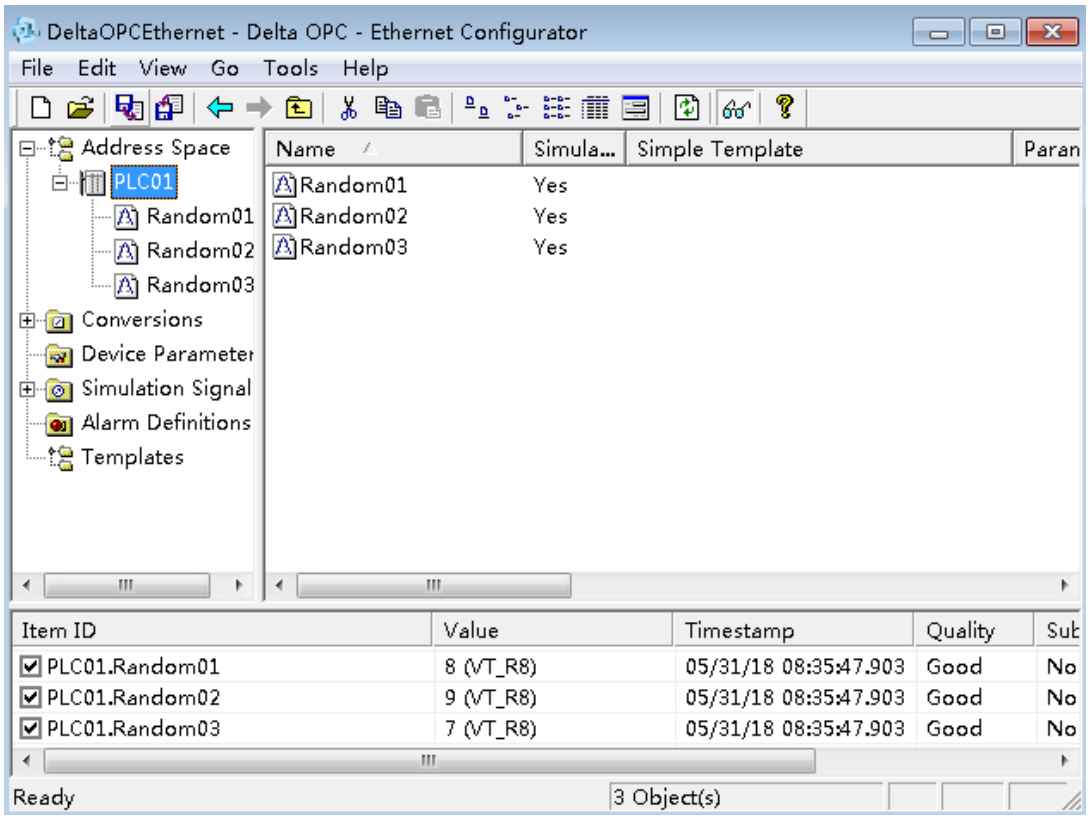
5.9.1 OPC

DIAView supports communication with OPC server

The IO communication between DIAView configuration software and OPC server is as follows:

➢ **Creating the communication between DIAView software and Delta AH500 OPC Server:**

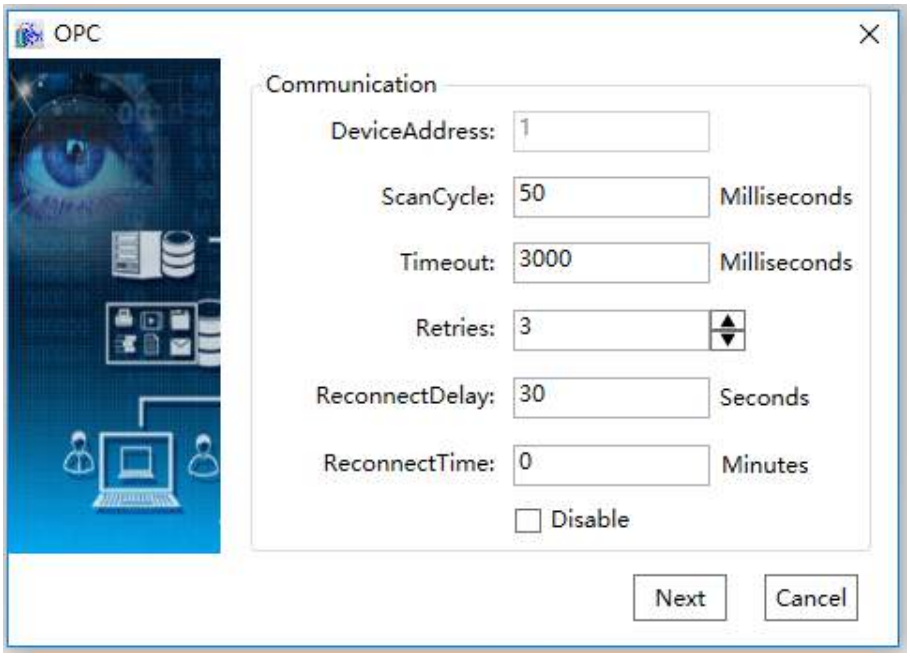
Step 1: Configure “AH500 OPC Server”, read data in the address of AH500, as shown in the figure below:



Step 2: In the project management area, right click on “IODevice “node and select “ New Device “, select "OPC" →"OPC" in “driver selection” window, as shown in the figure below:

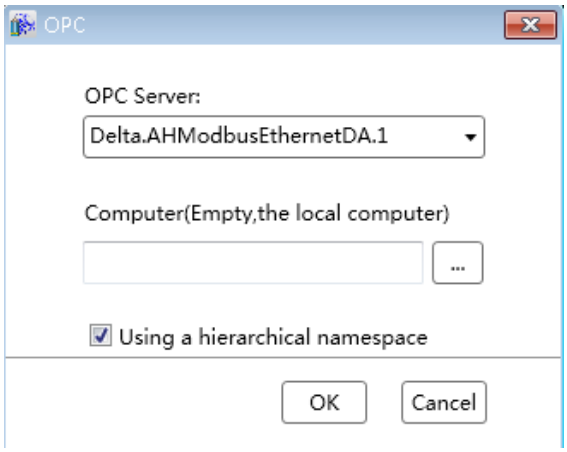
Step 3: Select “OPC” in the "Driver Selection" window“OPC” → “OPC”, as shown in the figure below:

Step 4: Configure the OPC communication parameters or keep the default values, as shown in the figure below:



Step 5: Click “next”, appear “OPC” configuration window, Select "Delta.AHModbus EthernetDA.1" in the

drop-down list of the "OPC server" option and keep other items the default value:



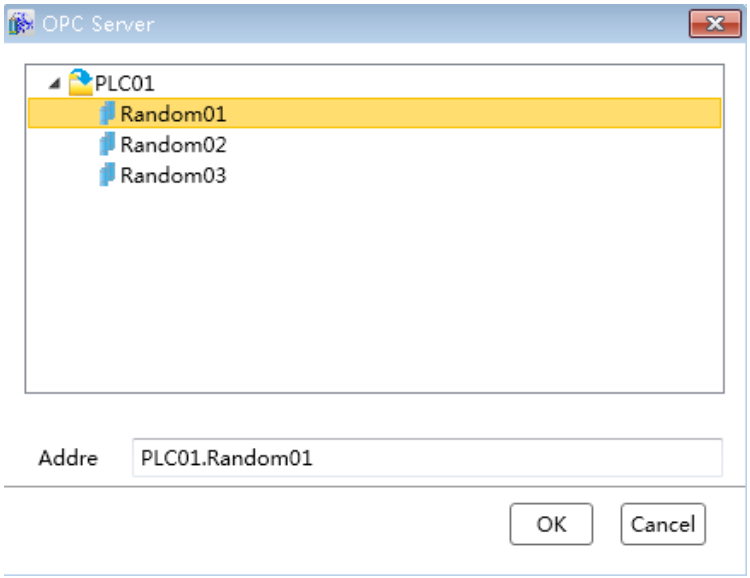
The meanings of each setting are as follows:

OPC server : Select OPC server.

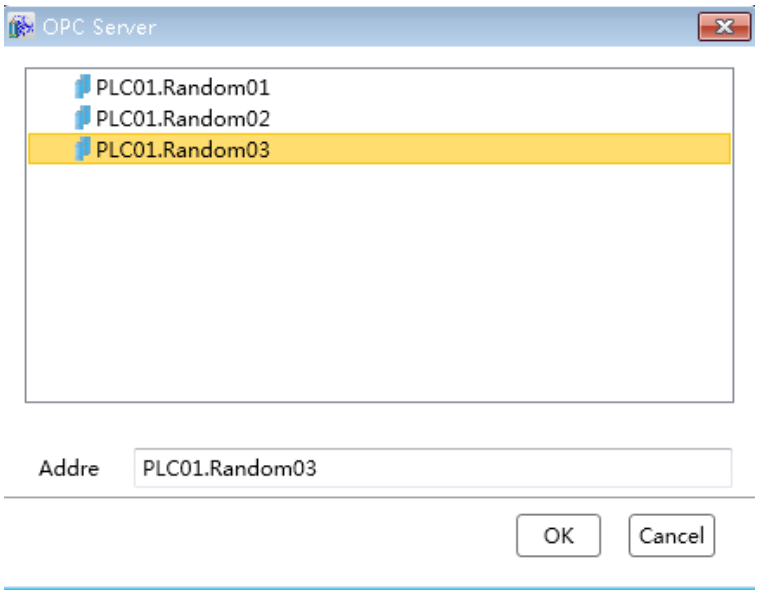
Computer (Null(Empty) value represents local computers): Install the OPC server computer. If the OPC server and the DIAView configuration software are installed on the same computer, it will be empty; if the OPC server is installed on the LAN or other computer, then click ... button to select remote computer name.

Using hierarchical namespace: It will affect the display way in selecting the device variable on the “address” column configuration window when creating device variable under the driver service-if checked display by the file directory; if not, display directly by variable path:

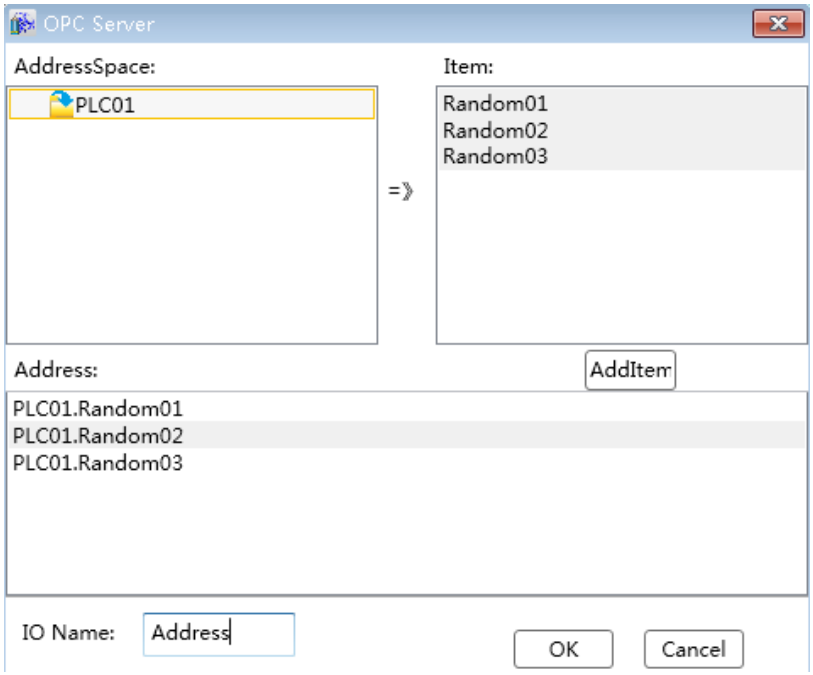
Checked effect:



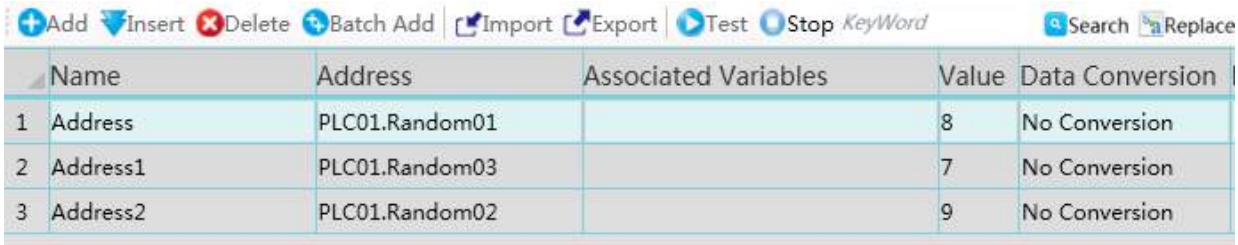
Unchecked effect:



Step 6: The new version adds batch additions. If the IO name is not added, the following figure is added:

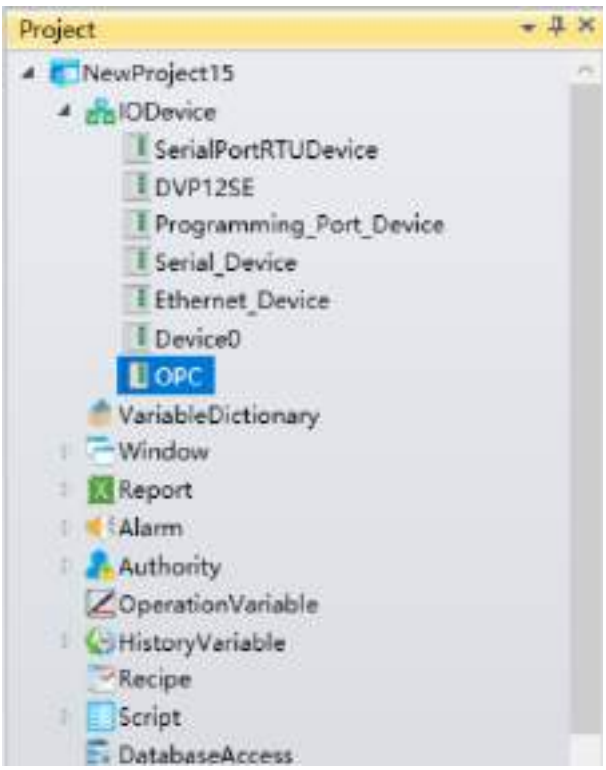


Step 7: You can test (When adding an address, if you want to test, please save it),the following figure:



	Name	Address	Associated Variables	Value	Data Conversion
1	Address	PLC01.Random01		8	No Conversion
2	Address1	PLC01.Random03		7	No Conversion
3	Address2	PLC01.Random02		9	No Conversion

Step 8: Click the "OK" button after all the configuration is done,a device with the default name will appear under the "IO Communication" node of project directory tree, as shown in the figure below:



5.9.2 OPCUA

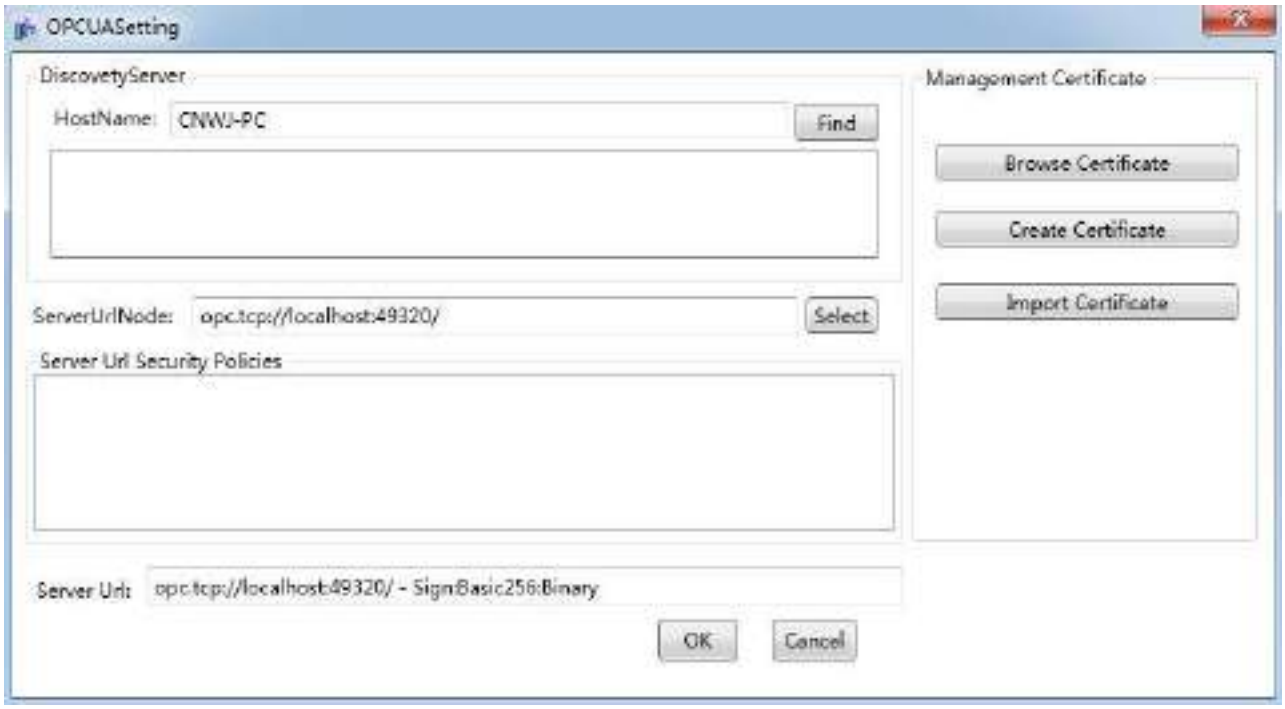
DIAView configuration software supports communication with OPCUA server.

The example of IO communication with OPCUA server configured in DIAView configuration software is as follows:

➢**Creating the communication between DIAView software and OPCUA Server:**

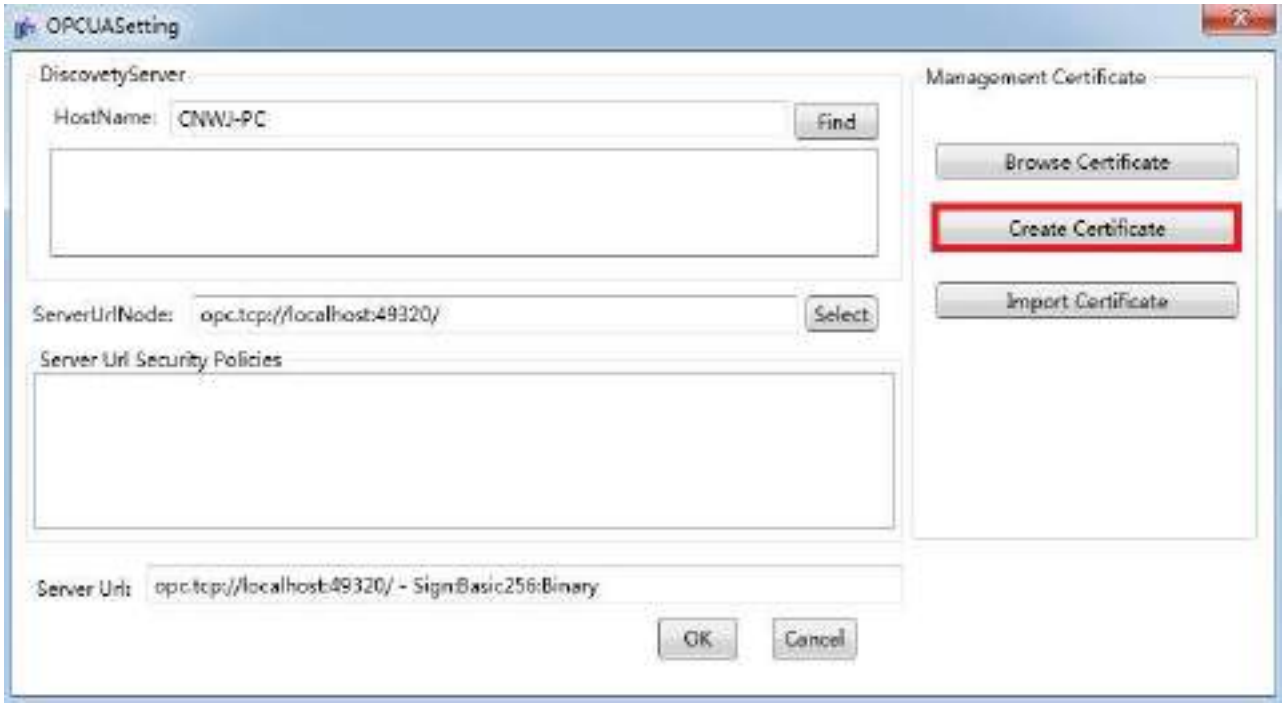
Step 1: Install "OPC UA Local Discovery Server 1.02.msi" of OPC foundation to discover "Server".

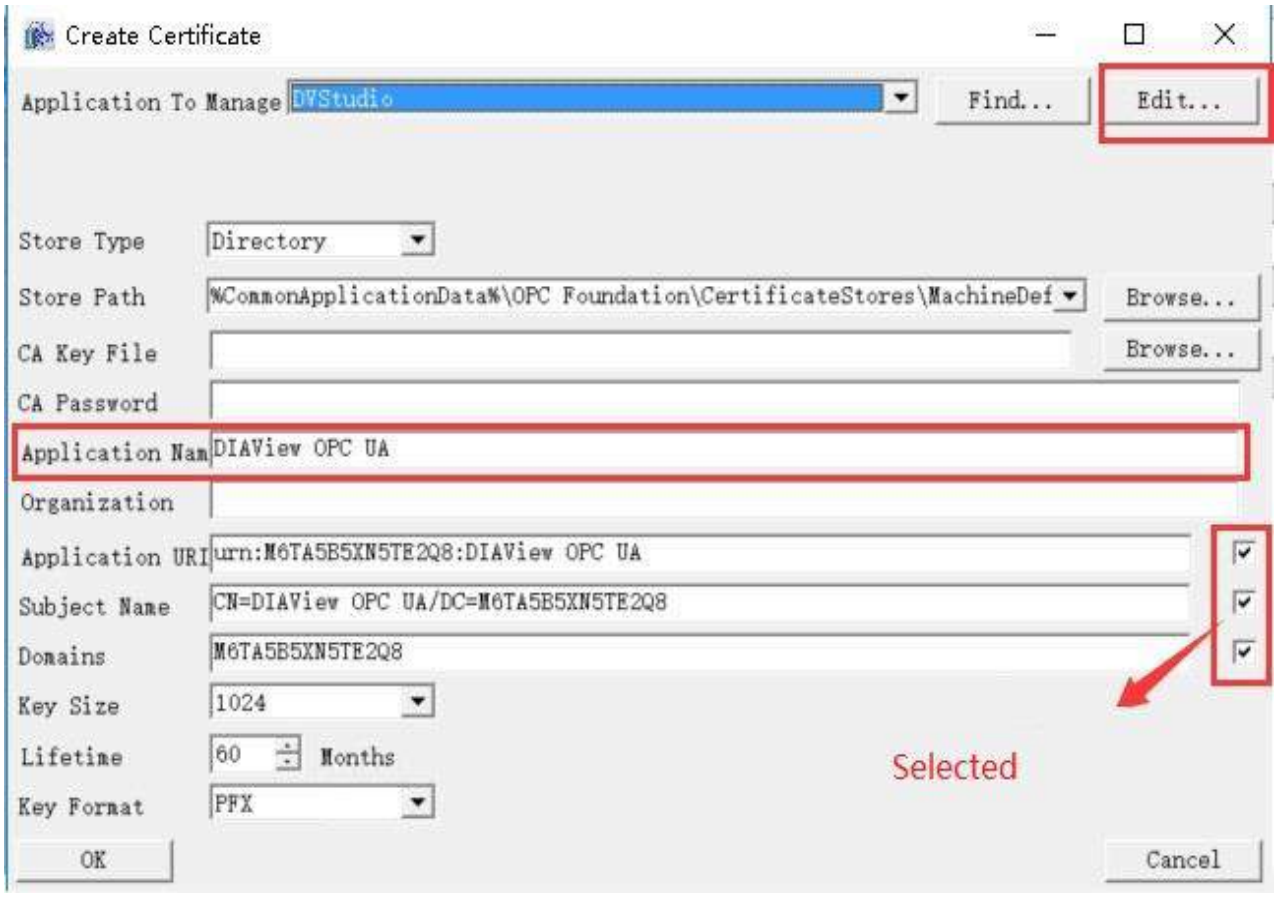
Step 2: Like the other devices, OPCUAClient configuration interface is created as follows:



1）Create certificates

Certificates trust each other to access data.





Change the default value of "Application Name" DIAView.IO.OPCUA to DIATest. Here is the test Name, and the main Name is "DIAView OPCUA".

Modify the default value of "Application URL", check it to change, and click twice to modify manually.

Modify the default value of "Subject Name", check it to change, and click twice to modify it manually.

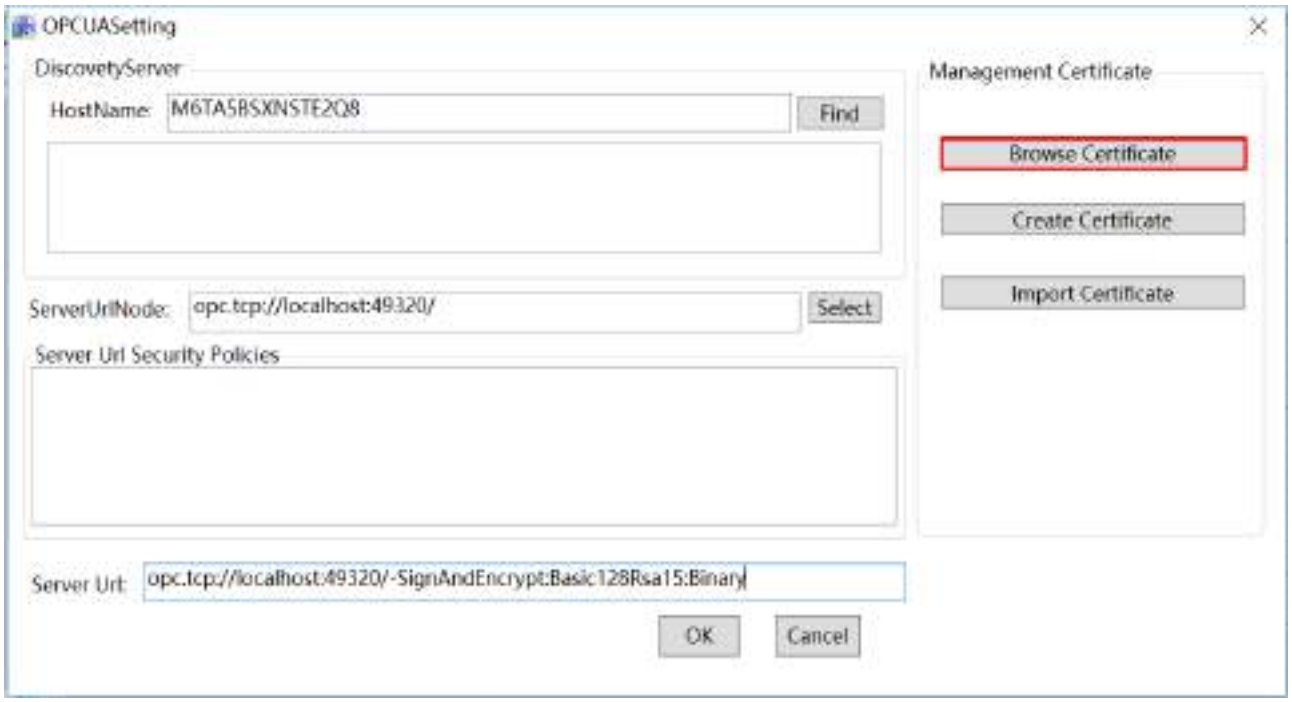
Modify the default value for "Domains" (computer name).check it to change, and click twice to modify it manually.

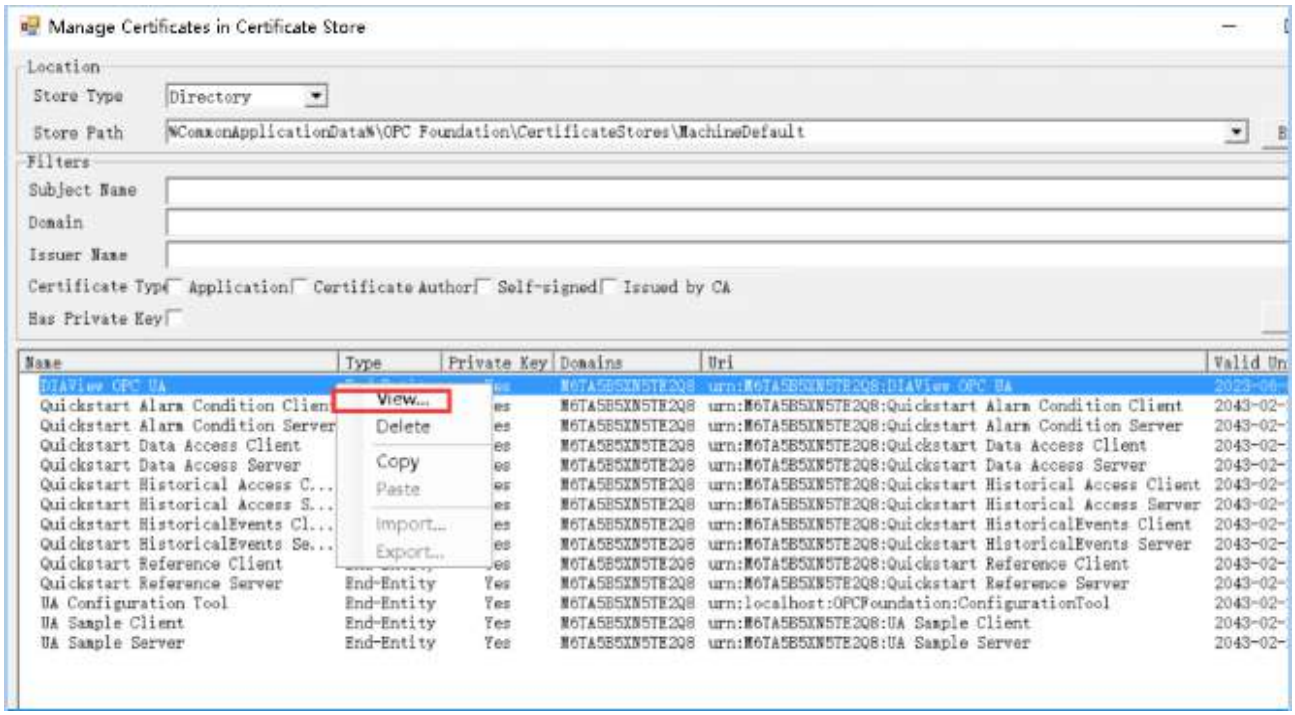
Click Edit to enter the following graphical interface:

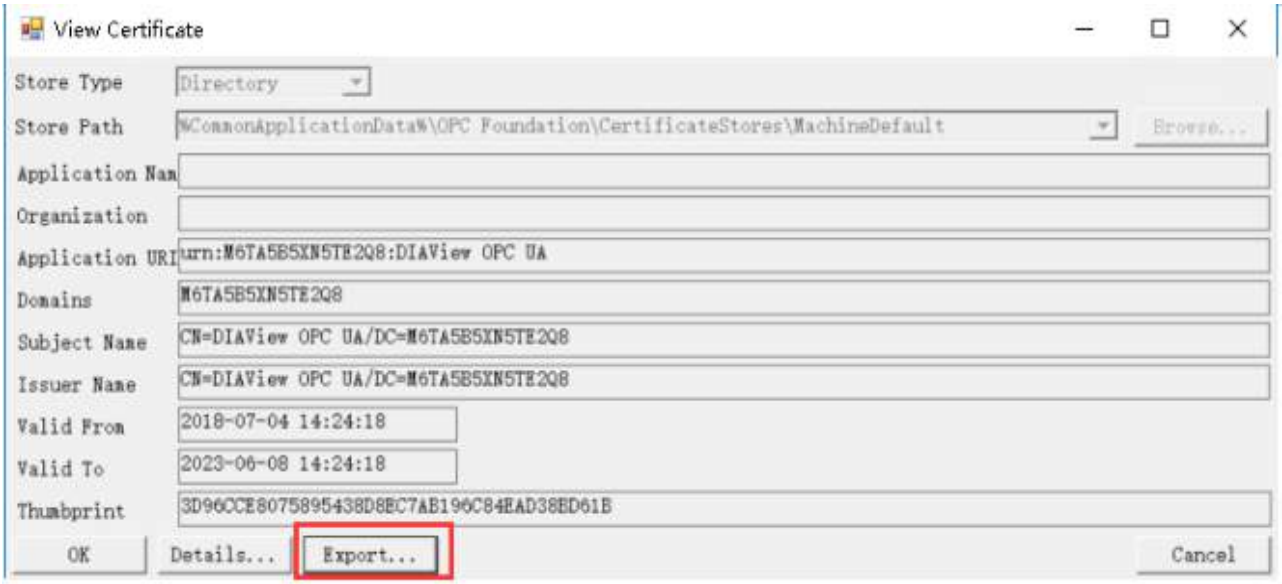
Only click OK to generate the certificate, and then proceed to the next step of browsing the certificate.

2）Browse certificate

The certificates created or imported are in the specified folder.





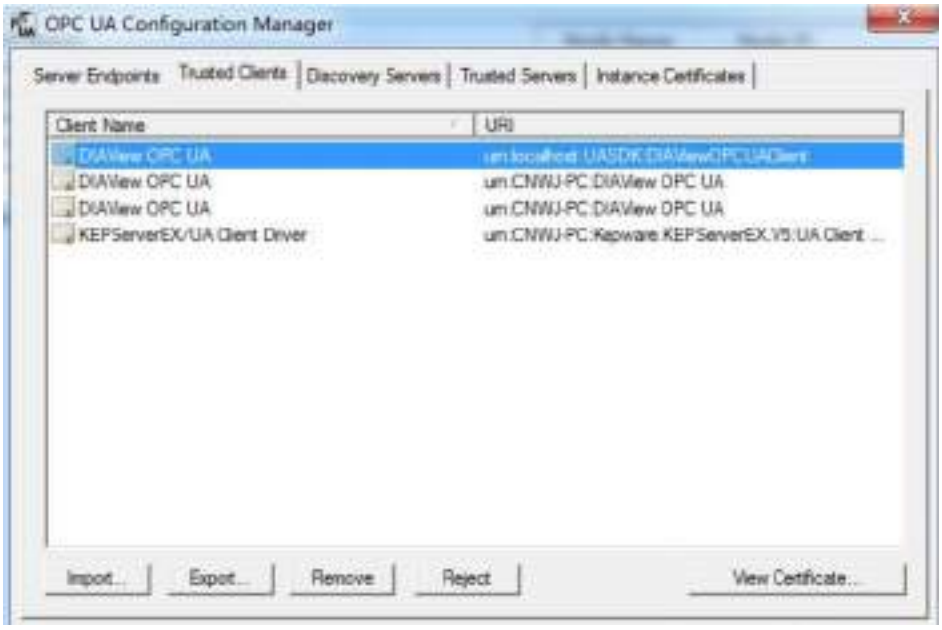


Click Export to Export the certificate.Then copy the exported certificate into the folder that you imported in step 3.

3）Import certificate

Import the certificate into the specified folder or manually.

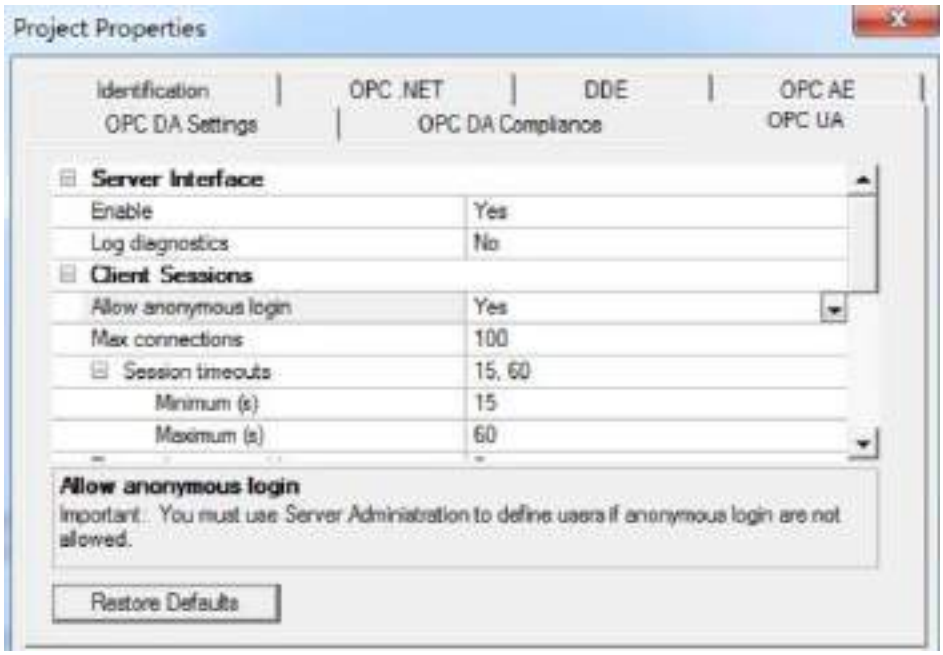
Main folders: MachineDefault\certs, UA Applications\certs, UA Certificate Authorities\certs, mainly located in "UA Applications\certs and MachineDefault\certs" folders



Step 3: Other OPCUAServer, take KepwareServer as an example, as shown in the figure below:

Copy the certificate to any file directory.

Step 4: Export the KEPServer certificate.



The image below allows anonymous login for Settings.

Step 5: Copy the certificate exported in step 4 into the directory corresponding to step 2.

Step 6: Add IO driver device as before.

5.10 Simulator

DIAView software supports the communication with simulator, simulator can offer data with regular changes when there is no filed data.

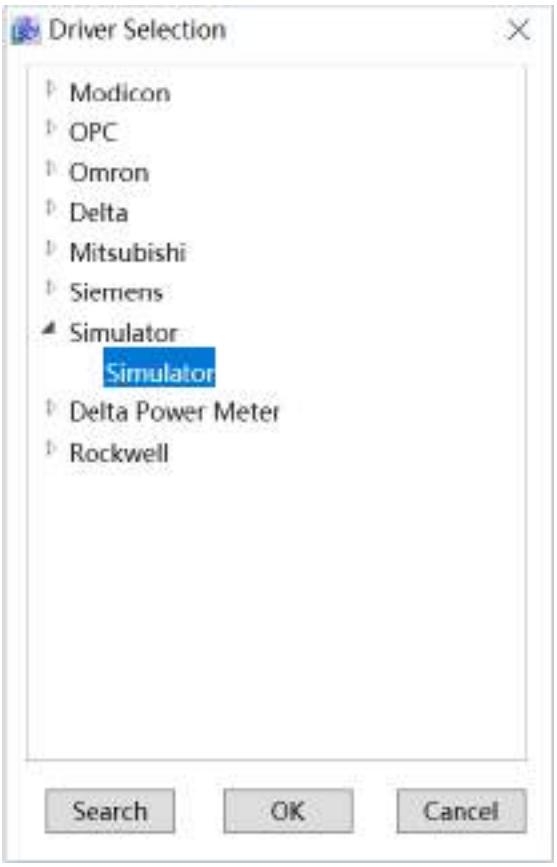
5.10.1 Simulator

The IO communication between DIAView software and stimulator is as follows:

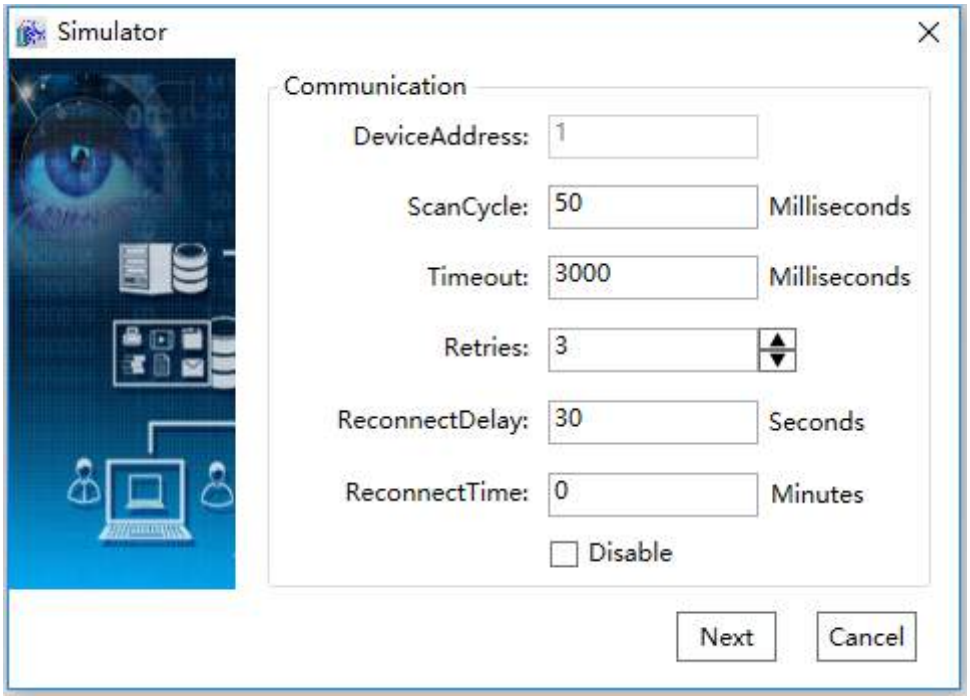
➢ **Creating communication between DIALink and Simulator:**

Step 1: In the project management area , right click on “IODevice “node and select “ New Device “,The menu selection in the figure below will appear:

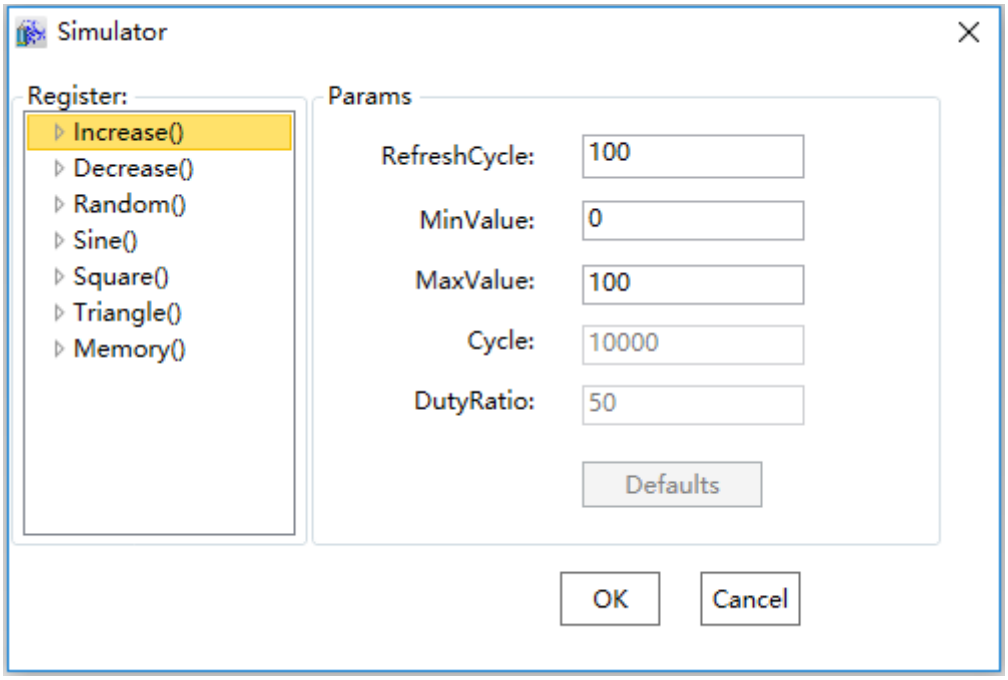
Step 2: Select “simulator” in the driver selection window ,as shown in the figure below :



Step 3: Configure the system simulator communication parameters, default values can be kept ,as shown in the figure below:



Step 4: Click "next", and "Simulator" window will appear:

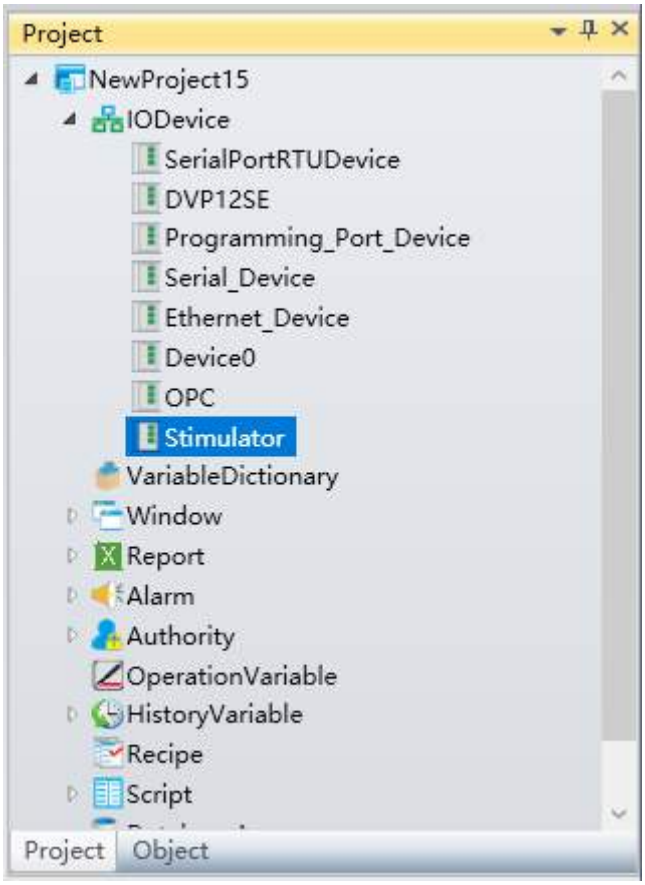


Simulation parameters:

Register	Range	Description
Increase	0~63	Increase
Decrease	0~63	Decrease
Random	0~63	Random
Sine	0~63	Sine wave
Square	0~63	Square wave
Triangle	0~63	Triangle wave
Memory	0~63	Memory, read-write

- ✧ **Minimum value:** (Int32 type value, minimum value), -2,147,483,648
- ✧ **Maximum value:** (Int32 type value, Maximum value), 2,147,483,647
- ✧ **Cycle:** Partially used; a data cycle
- ✧ **Duty ratio:** Refers to the position of positive pulse in the pulse period (only can be set for Triangle and Square)

Step 5: Click the "OK" button after all the configuration is done,a device with the default name will appear under the "IODevice" node of project directory tree, as shown in the figure below:



6. Variable Dictionary

6.1 Overview

Project created by using the DIAView software requires various types of variables for data interaction in order to display and control the field production status in real-time.

There are two types of variables in the DIAView software: System variables and project variables. System variables are variables built-in the DIAView system and cannot be modified;

Project variables are variables customized by the user in the project, and they are usually managed by the “variable dictionary”.

The variable dictionary is a collection manager of all the variables in the system, and is able to

perform uniformed and centralized management of all the variables; variable groups and variables can be created in the variable dictionary, and functions such as browsing variable references, variable statistics and variable replacement etc. Operations such as adding, deleting, copying, pasting, cutting, importing and exporting of variables can also be performed.

6.2 Variable group

A variable group is a unit for grouping managing variables.

The DIAView software can group and manage the variables in the project by creating variable group and then adding variables in the variable group. To use the variables in a variable group, the name of the variable group must first be appointed; the variable group name and variable name is separated with ".", for example "Var.NewVariableGroup.NewVariable".

Several variable groups can be created under a variable dictionary, and variable groups can also be created under variable groups.

Variable group naming rules:

(1) Composed of English letters, numbers, Chinese characters and underline, and can only begin with an English letter or Chinese character;

(2) not case sensitive;

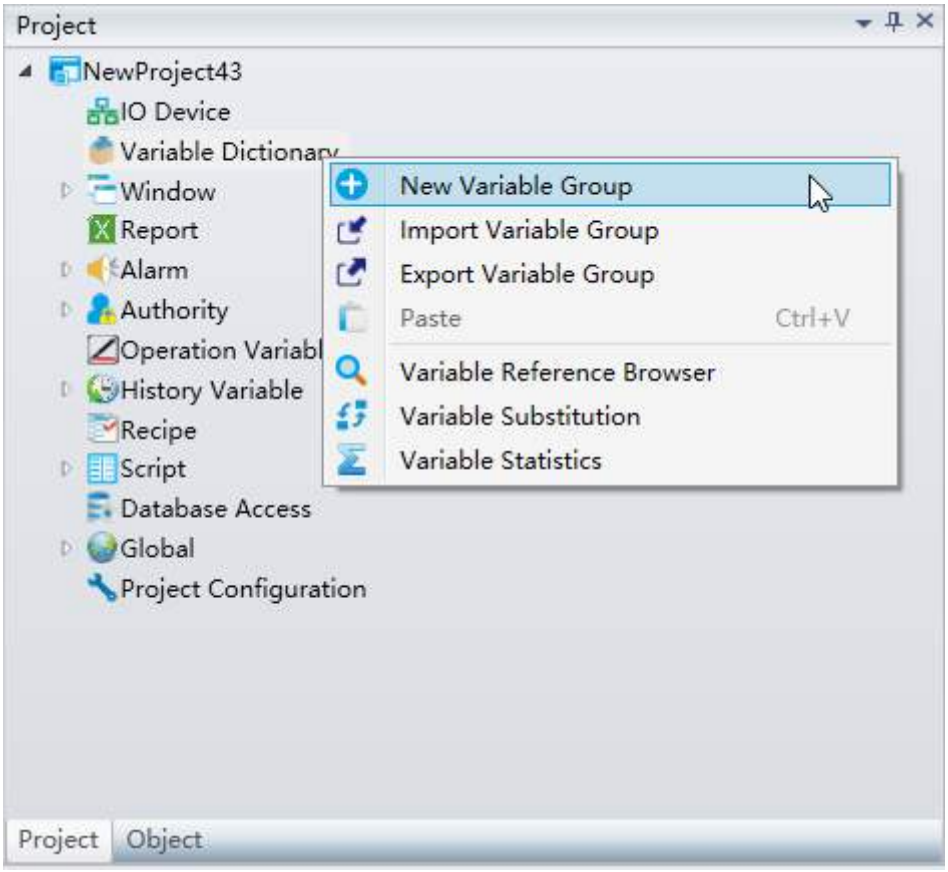
(3) the length cannot exceed 200 characters and cannot exceed 25 Chinese characters;

(4) variable group names cannot be repeated under similar class nodes within a project;

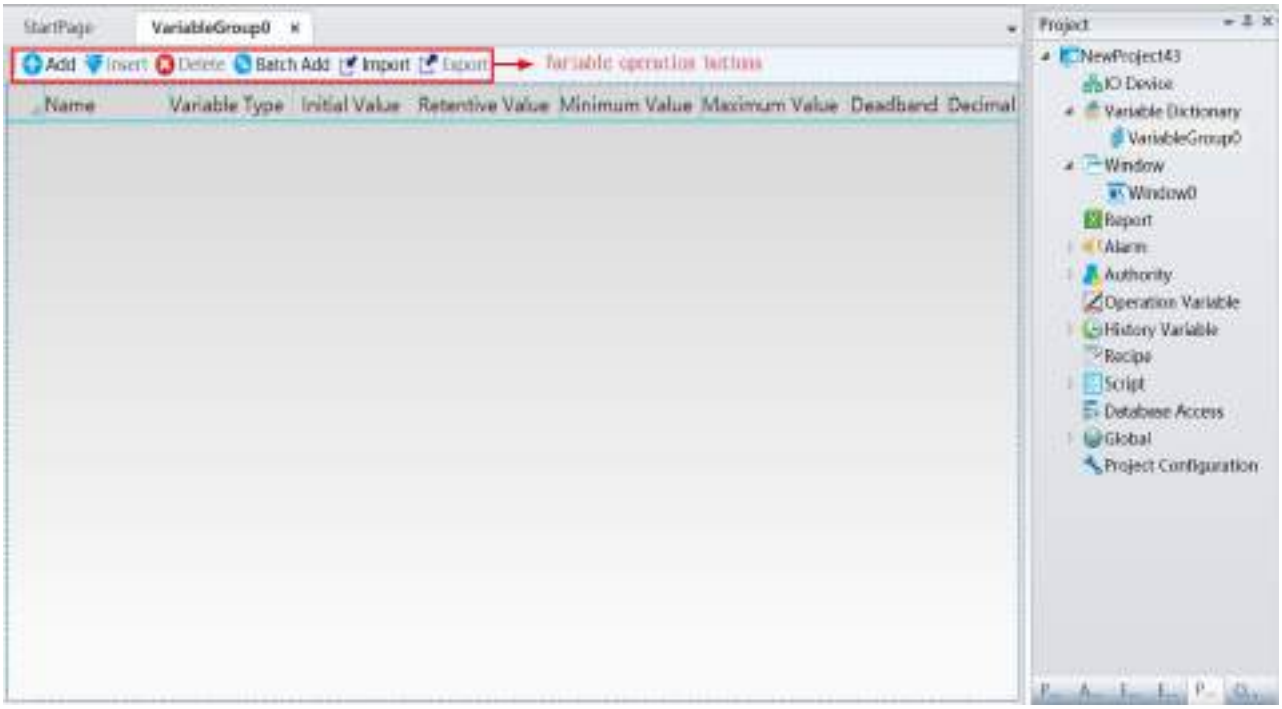
(5) if there are variables and variable groups under similar class nodes within the same project, the variable and variable group names cannot be repeated.

Adding variable group

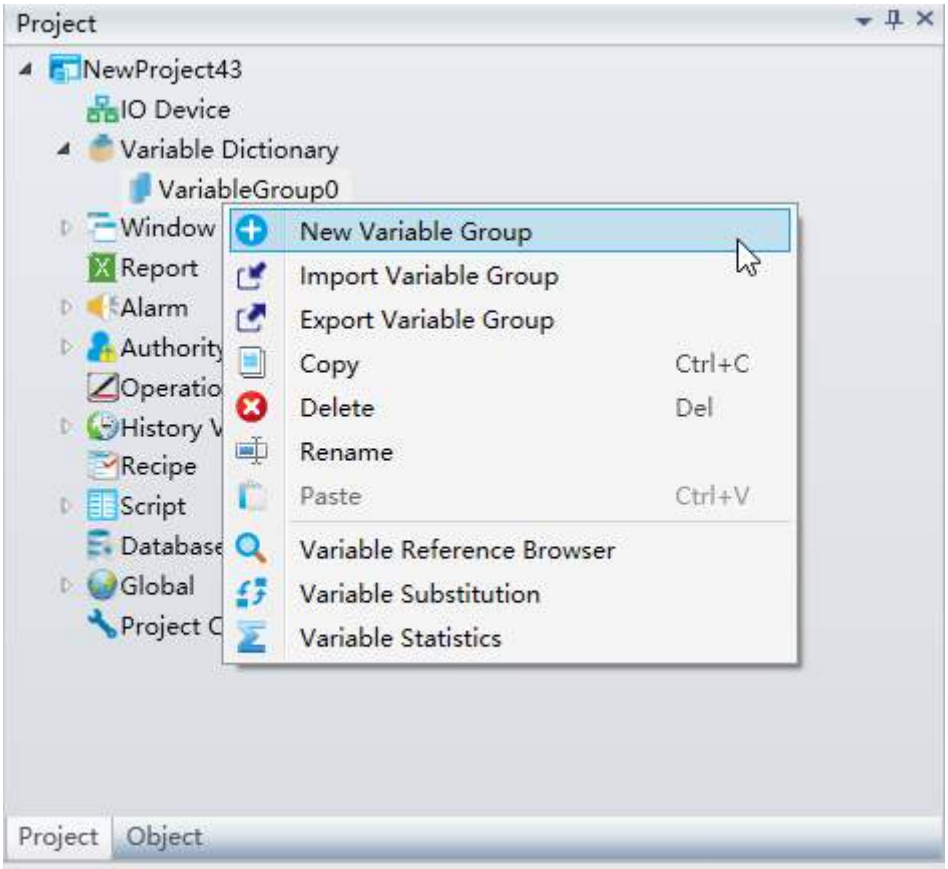
➢ Right-click the "VariableDictionary" node in the project window tree index and then click the "New Variable Group" from the right-click menu. The system will create a new variable group with the default name, as shown in the figure below:



➢ Double-click the created “variable group” sub-node and open the variable operation window to perform operations including adding, inserting, deleting, bach add,import,and export of variables, as shown in the figure below:



➢ Right-click the created variable sub-node to perform operations including “New Variable Group”,"Import Variable Group","Export Variable Group", “Copy”,“Delete”,"Rename", “Paste” etc. (file must be saved before copying), as shown in the figure below:



Adding variables directly under “variable dictionary”

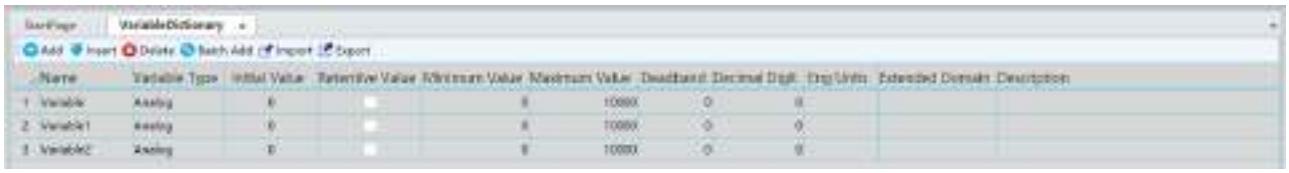
➢ Users can also double-click on the “variable dictionary” node to open the variable configuration window without adding a variable group to perform add variable and other operations

6.3 Variables

System variables: Fixed variables built-in the system; the DIAView has 16 system variables which can be used directly by the users:

Name	Type	Description
Year	Integer	The current system date of year
Month	Integer	The current system of month
Day	Integer	The current system days number
Hour	Integer	The current system of hour
Minute	Integer	The current system of minute
Second	Integer	The current system of seconds
Millisecond	Integer	The current system of millisecond
DayOfYear	Integer	The current system date is in the first few days of the year.
DayOfWeek	Integer	The current system date week
Date	String	The current system date string
Time	String	Current system time string
Now	Date	The current system of date and time
StartTime	Date	Current project startup time
ElapsedTime	Real	Project running time, unit for seconds
CurrentUserName	String	Current login user
HasAlarm	Bool	Whether the current system exists the alarm
ProjectDir	String	Current project directory

Variables are values that can change at any time in a project; they are important participants for the information interaction of the system. Variable information in the DIAView is as follows:



Basic properties of variables:

- ✧ **Name:** Name of the variable.

  Naming rules:

  (1) Composed of English letters, numbers, Chinese characters and underline, and they can only begin with an English letter or Chinese character;

  (2) not case sensitive;

  (3) the length cannot exceed 200 characters and cannot exceed 25 Chinese characters;

(4) Variables in the same node or variable group cannot have repeated names:

(5) if there are variables and variable groups under the same node, the variable and variable group names cannot be repeated.

- ✧ **Variable Type:**Defines the value type of the variable,There are 3 types of variables:

  Text: Character type (the maximum length of this type of variable is two thousand characters)

  Digital: Digital type (False, True).

  Analog: Integer type and real type.

- ✧ **Initial Value:** The initial value of the variable

- ✧ **Retentive Value:** If checked, the “initial value” for this variable will be saved as the last acquired value when the system execution ends; or else it will remain the same.

- ✧ **Minimum Value and Maximum Value:**Specifies the value range of the variable (these two properties are only available for the “analog value” type variable).

- ✧ **Deadband:**Specifies a value as the deadzone value and forms a deadzone interval with the current value of the variable:

  (Current variable value - deadzone value) <= deadzone interval <= (current variable value ＋ deadzone value);

  - **Effect**: Data filtering.

  - **Principle:** When the next variable value acquired is within the deadzone interval range, no new values will be acquired for the variable value and the original value will remain the same. If it is not in the deadzone interval range, then acquire a new variable value and the deadzone interval will also change accordingly. It will keep looping this way. (This property is only available for the “analog value” type variable).

- ✧ **Decimal Digit:** Specifies the number of decimal places for the variable (this property is only available for the “analog amount” type variable).

- ✧ **Eng Units:** Sets the unit of the variable; for example: second/s, milliliter/ml and kilogram/kg. Variable value can be acquired and also display this project’s unit when using variables; for example: Var.NewVariableGroup.NewVariable.ValueAndUnit (this property is only available for the

“analog value” type variable).

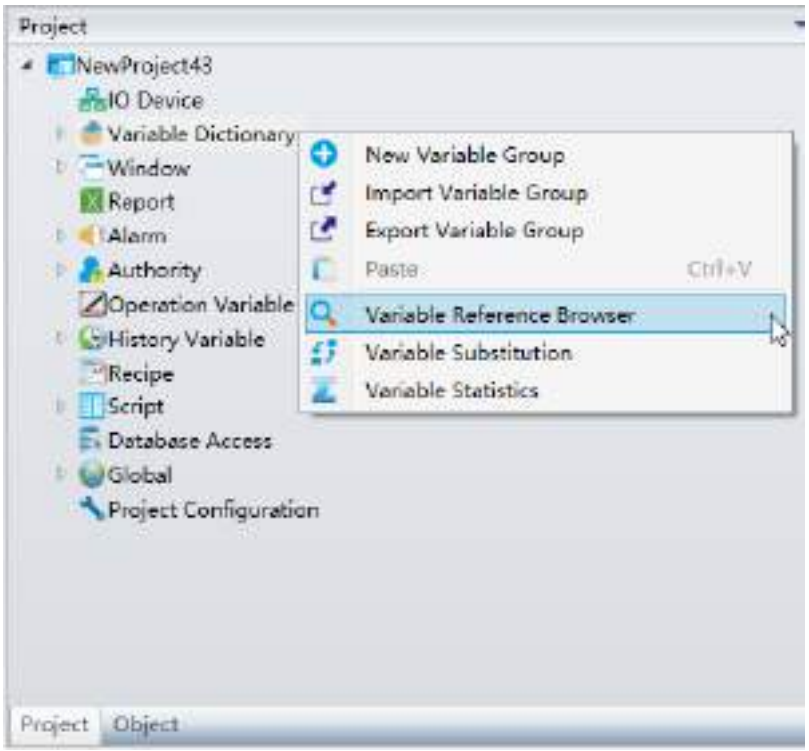
- ✧ **Extended Domain:** Set domains for variables(Multiple domains can be set,with each domain separated by commas or semicolon),users can user scripts to get variable value for the same domain.

- ✧ **Description:** Add explanation information for the variable.

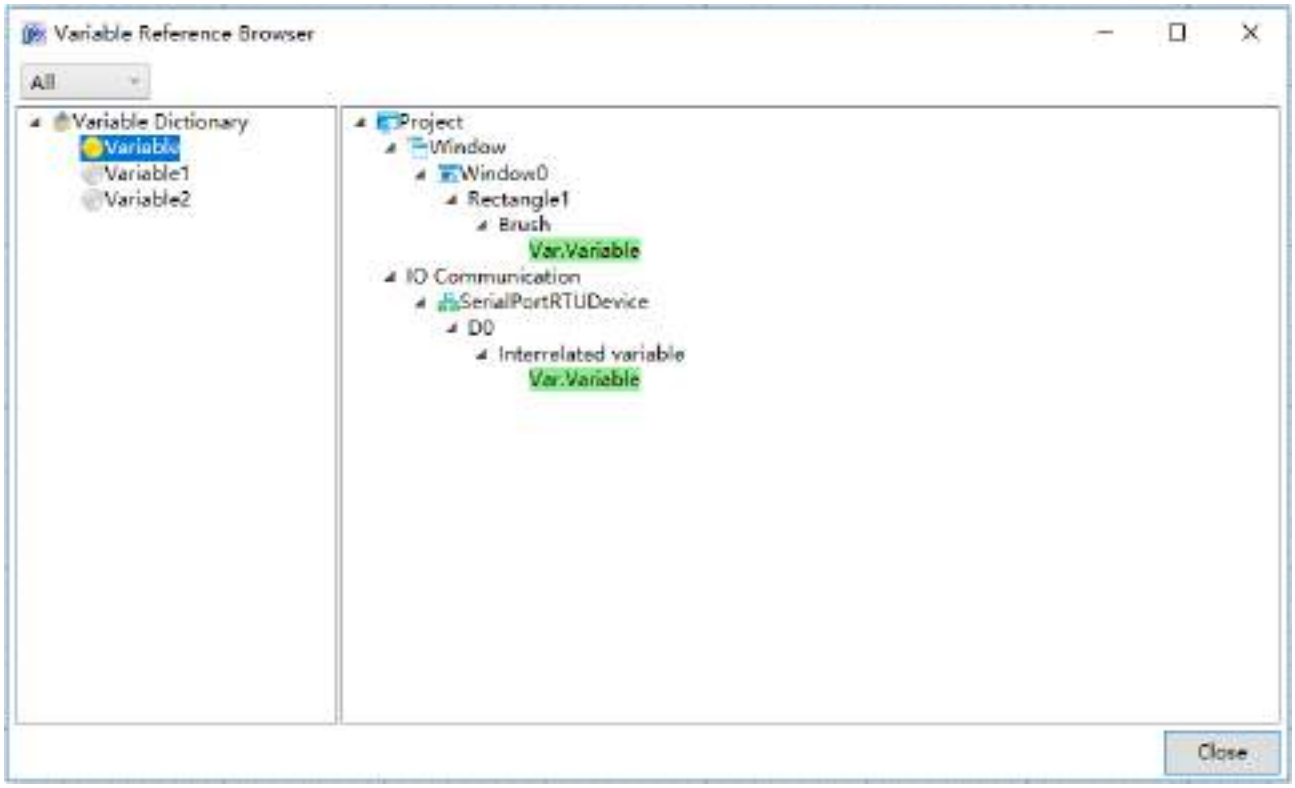
6.4 Variables usage browsing

The variable usage browser refers to the window used to view the status of all variable references in the variable dictionary. It can be used to view whether a variable is referenced and information on the reference path.

➢ Right-click the “VariableDictionary” node in the project window tree index and then click the “Variable Reference Browser” from the right-click menu, or click the “Variable Reference Browser” in the right-click menu of the variable group node, as shown in the figure below:



➢ The variable reference browser will pop-up, as shown in the figure below:

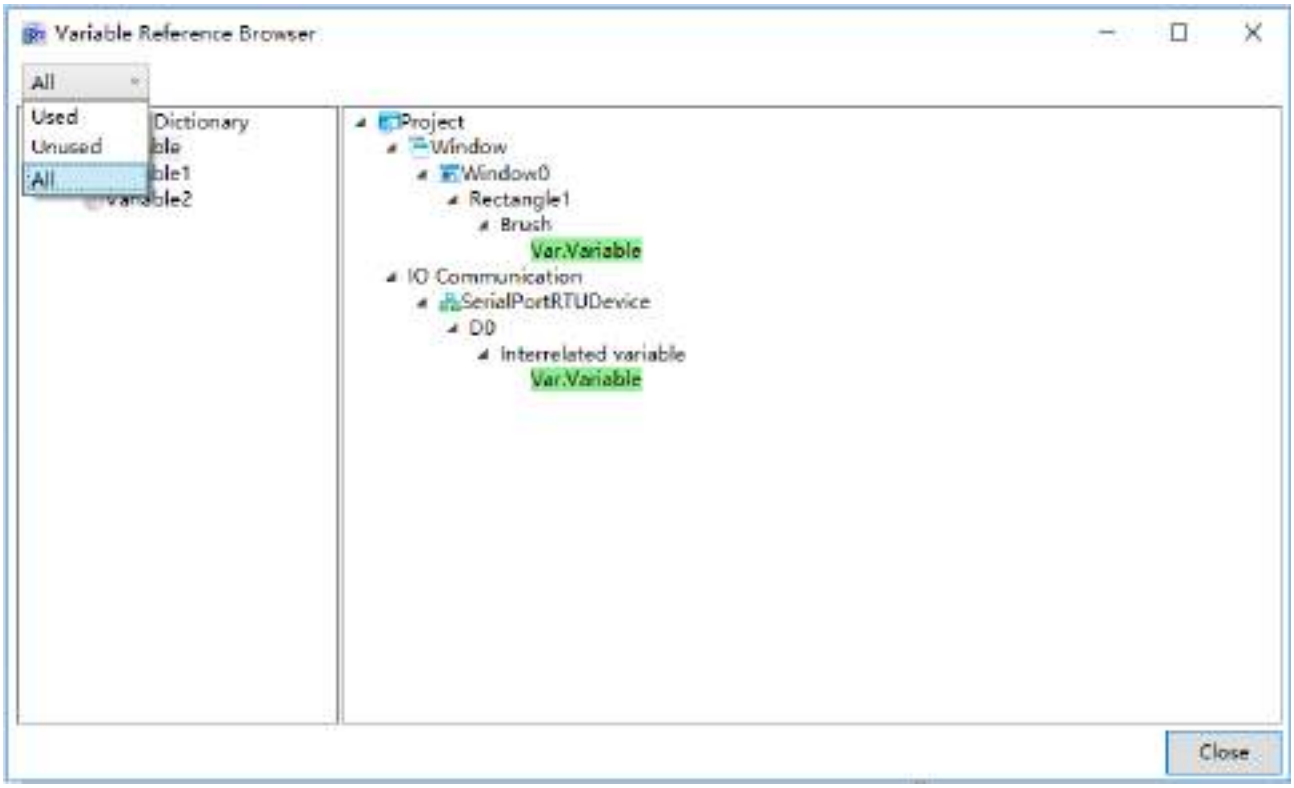


1. The left side of the variable usage browser is the tree index of the variable dictionary, and the right side is the display window of variable references.

2. In the tree index to the left, if the round symbol in front of the variable displays as “yellow”, it means that this variable is referenced. If it is “gray,” it means that it is not yet referenced.

3. Select the variable to view from the tree index to the left, and the variable reference display window on the right will display all objects in the project that uses that variable. They are displayed using an expanded tree index in order for users to pinpoint the objects that actually uses this variable.

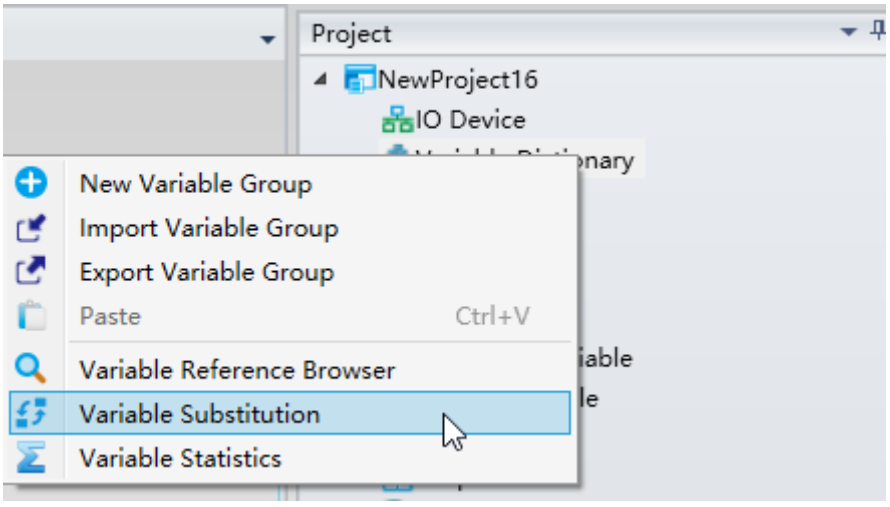
4. Click the pull-down button to filter the variables, as shown in the figure below:

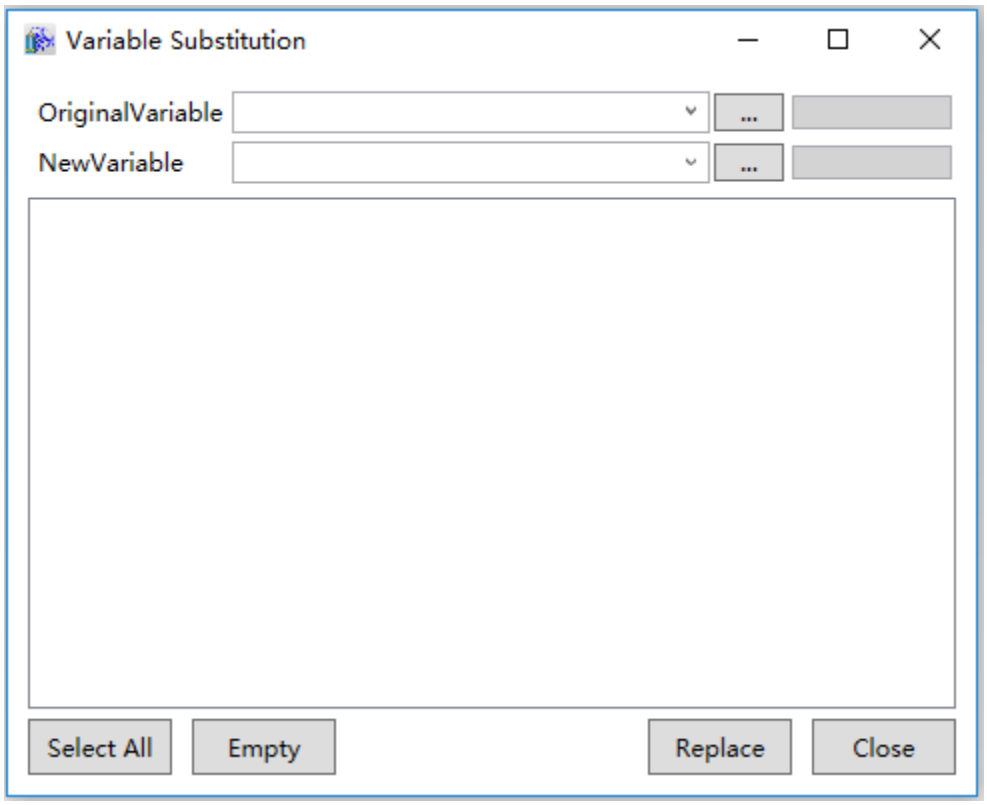


6.5 Variables substitution

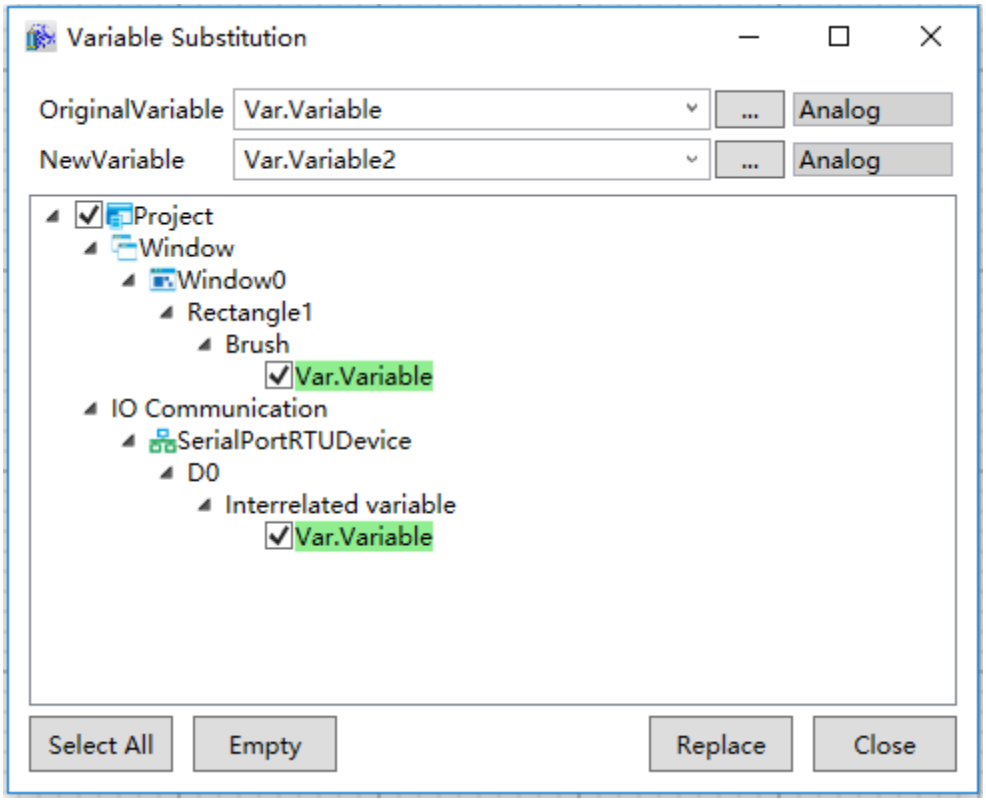
Variable substitution refers to substituting a referenced variable in the project into another variable.

➢ Right-click the “VariableDictionary” node in the project window tree index and then click the “Variable Substitution” from the right-click menu, or click the “Variable Substitution” in the right-click menu of the variable group node, as shown in the figure below:

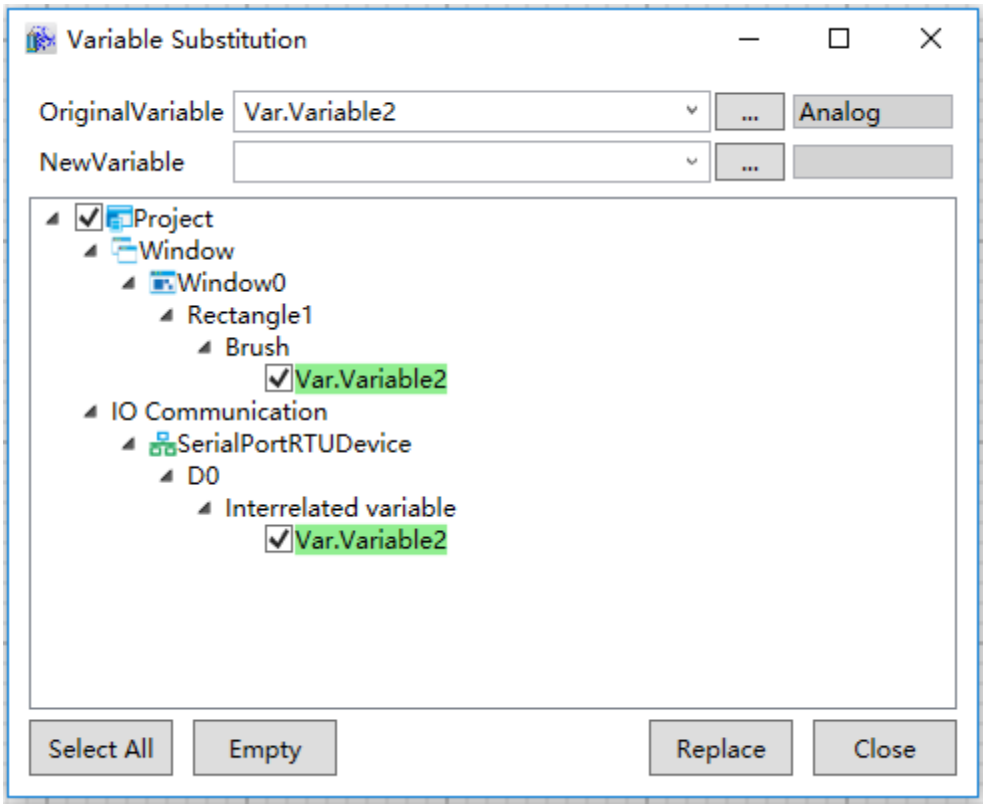




Select the variable to substitute from the “OriginalVariable” field. It can be selected by pressing the variable selection button to the right and opening the variable browser. The selected variable must be referenced in order for it to be replaced. When a variable is selected, the reference level list of this variable will be listed in the window. If there is no reference level list, then it means that this variable is not referenced, as shown in the figure below:



Select the new variable to substitute the original variable from the “NewVariable” field, and then select the variable to substitute in the reference level list (preset to select all). Users may press the “Select All” button to select all, and then press the “Replace” button to perform variable substitution. Substituted result is as shown in the figure below:

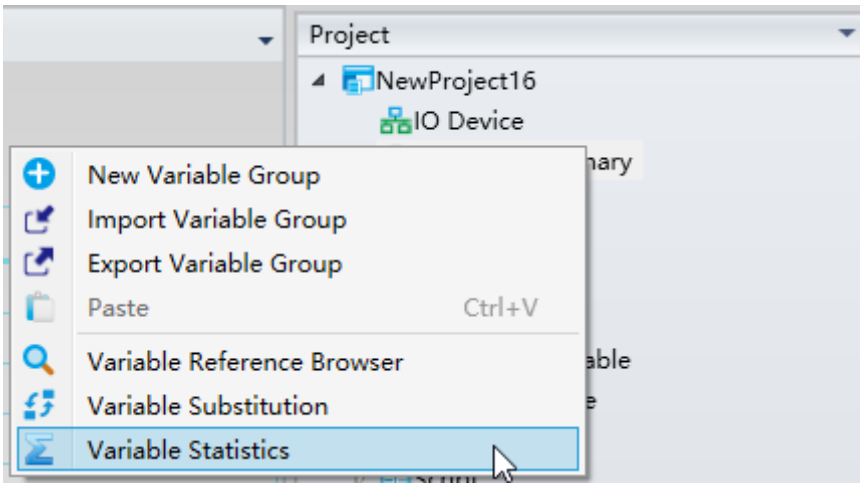


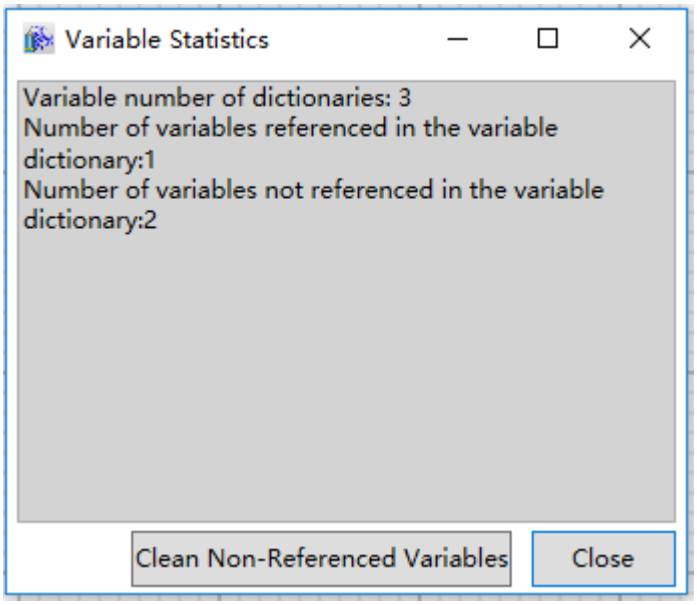
※ Variable substitution is not supported for variables associated with historic curves and reports.

6.6 Variables statistics

Variable statistics is the statistics for the total number of variables in the project and the usage situations of the variables.

➢ Right-click the “VariableDictionary” node in the project window tree index and then click the “Variable Statistics” from the right-click menu, or click the “Variable Statistics” in the right-click menu of the variable group node to open the variable statistics window, as shown in the figure below:

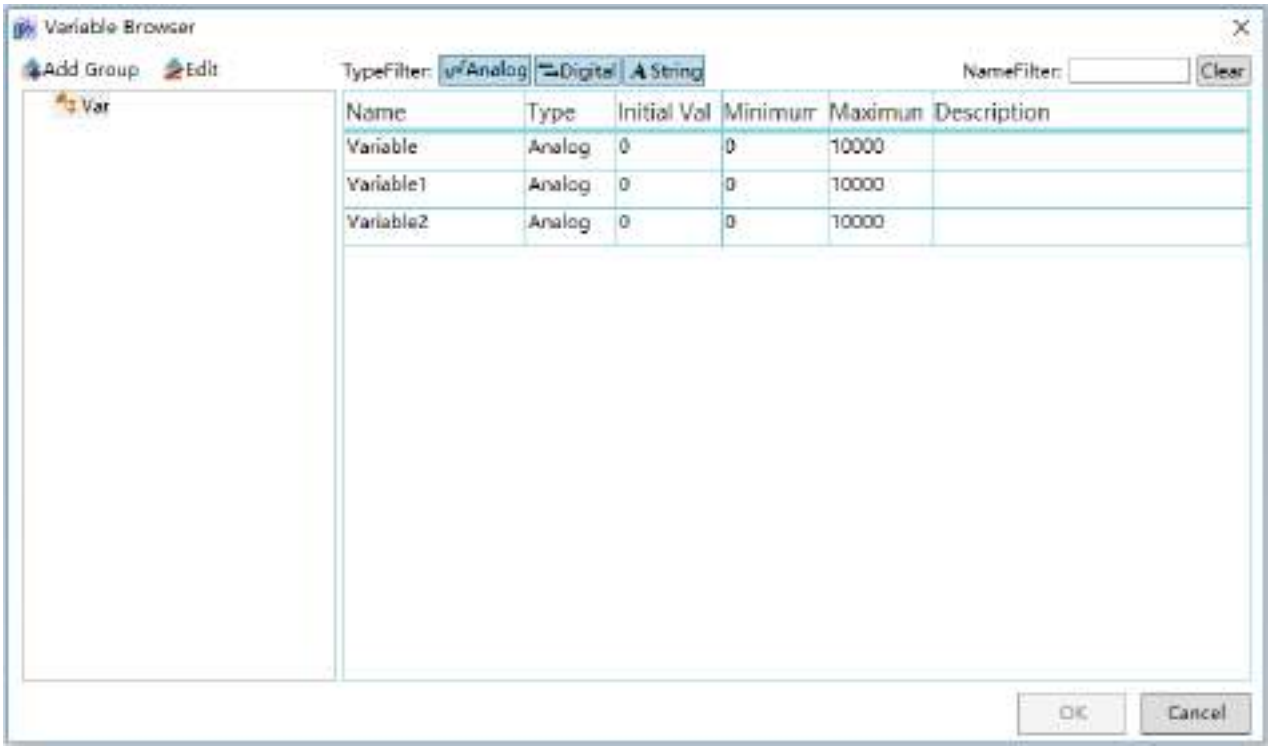




Click the “Clean Non-Referenced Variables” button to delete all unused variables in the variable dictionary.

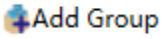
6.7 Variables browser

The variable browser is the window that displays all variables in the "Variable dictionary”; many functions in the DIAView project need to use variables. For example, configurations such as recipes and variable operations etc. need associated variables, so “Variable Browser” can be opened to search for the variable that needs to be used, as shown in the figure below:

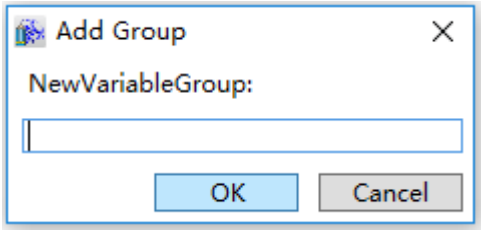


Once the “Variable Browser” is opened, all variable groups in the project will be displayed in the left part of the tree index; press the variable group and all variables in the variable group will be displayed in the right part of the window. Click and select the variable and then press the “OK” button, or double-click the variable need to select the variable and use it.

Toolbar

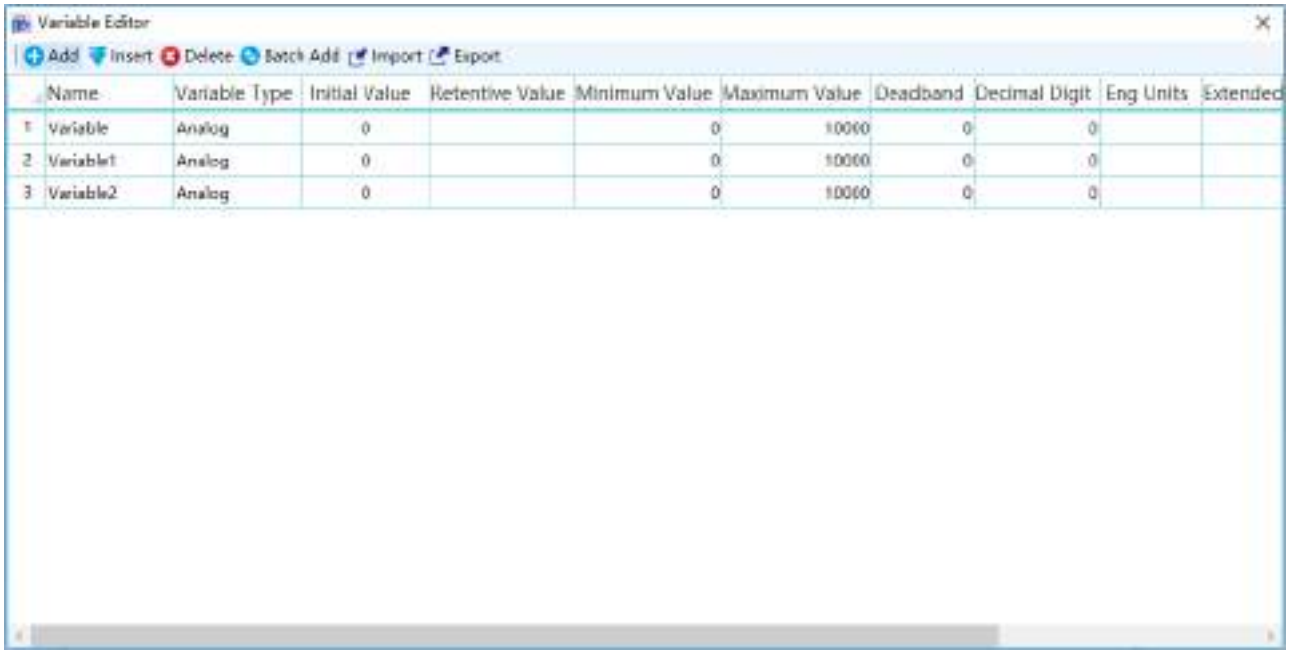
Step 1: Toolbar 

button and the add variable group dialog will pop-up, which allows you to add variable groups:



Step 2: Edit variable, press the 

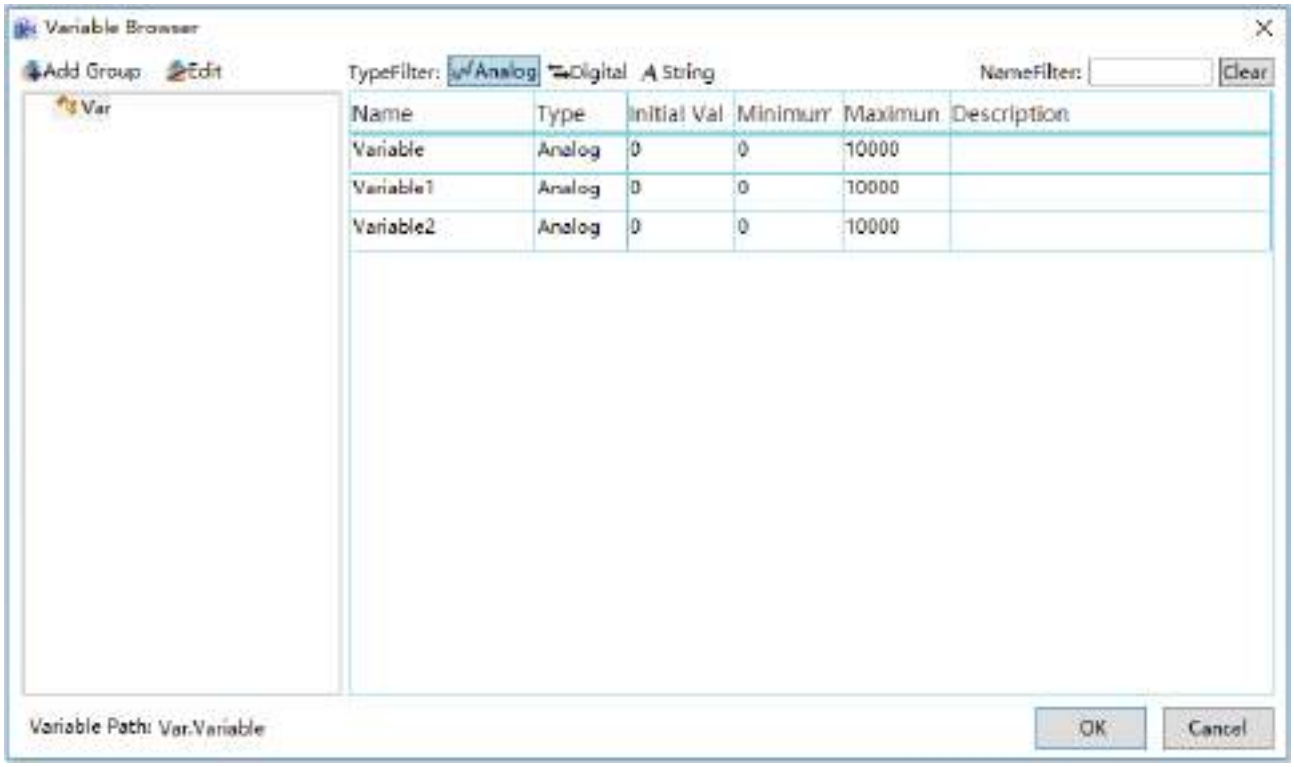
button and the variable editor window will pop-up, which allows you to perform variable operations:



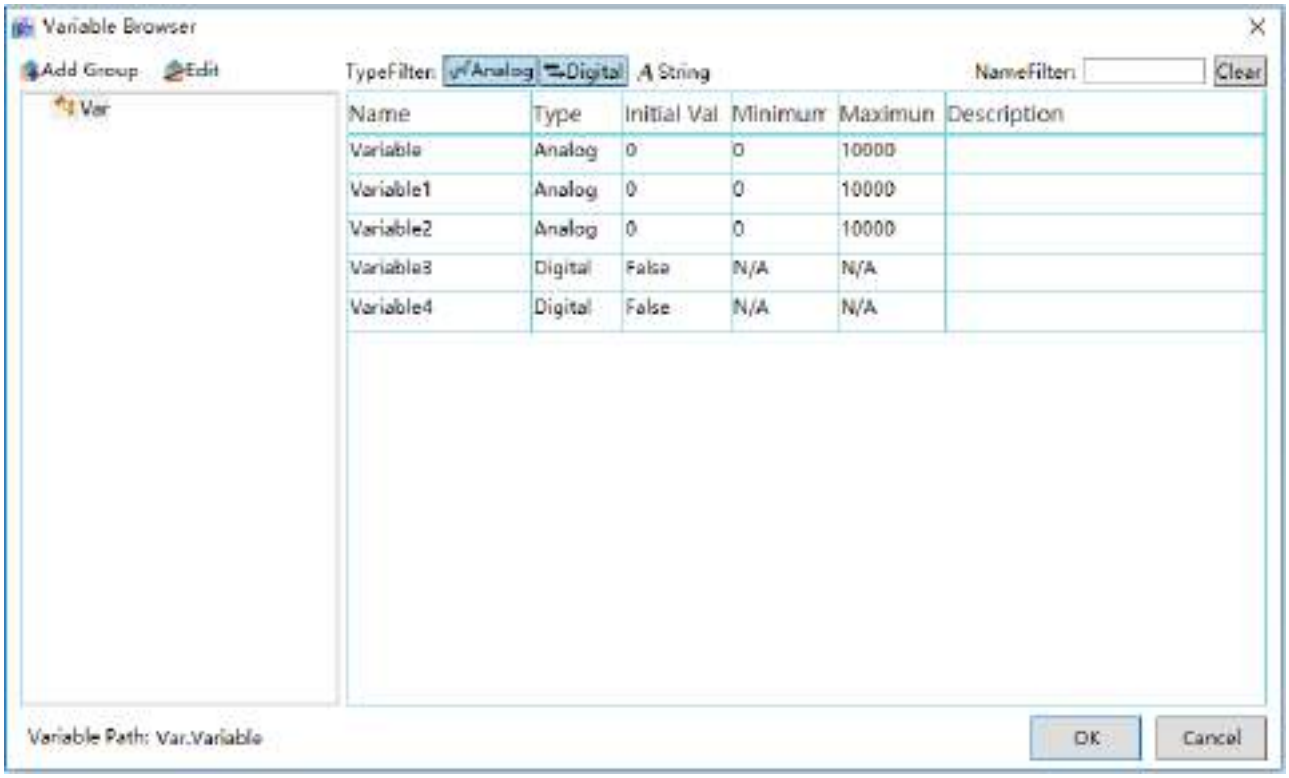
Step 3: Variable type filter, the background of the button will change to blue when selected

Filter: Analog Digital String , which means display all this type of variables in the variable dictionary; when it is not selected, it means do not display this type of variables:

a. Only browse “Analog value”



b. Browse “Analog value” and “Digital value”



Step 4: Search variable, NameFilter: Clear enter the variable name keyword in the input frame to search for variables in real-time.

6.8 Expressions

Expressions are meaningful arrangement method combinations of digits, operators, digital grouping symbols (brackets etc.) and variables used to determine a value.

Expressions are used for configuring animations and events in the DIAView software project and writing user program scripts etc. Operations such as logical operations, arithmetic operations and relational operations and performed through expressions to generate a new result for the program to use in order to satisfy the configuration needs of project animations, events and scripts, and achieve certain functions in the project.

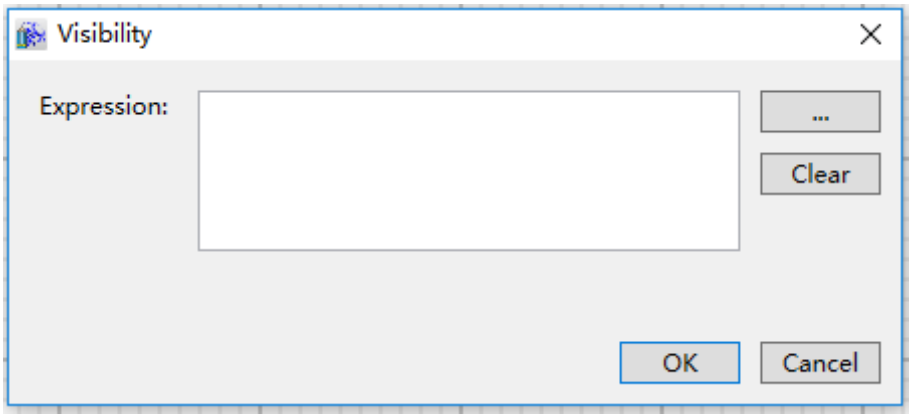
Common operators used in expressions:

Type	Operator	Discription	Example	Result	Priority
Arithmetic operator	^	power	5^2	25	From high to low
	*	product	80*2	160	

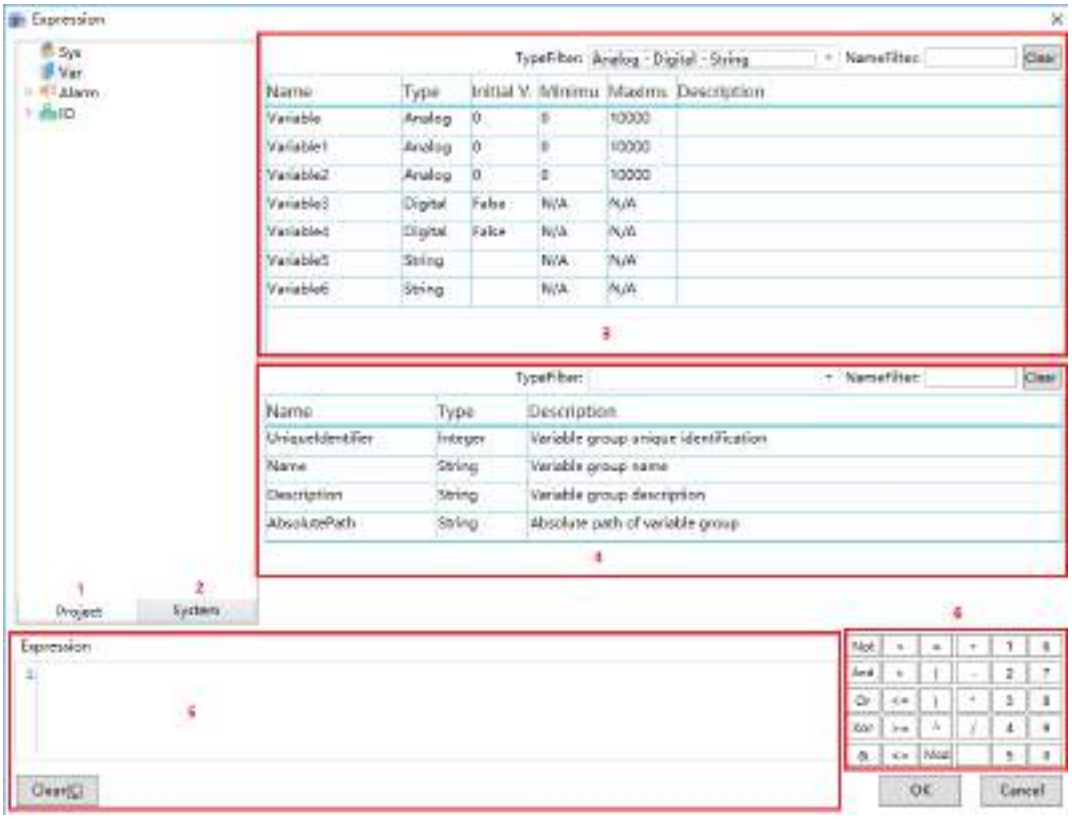
	/	quotient, the result is float	5/3	2.5	(relational operators have the same priority); small brackets can be used to change the priority of operations.
	\	divisor, returns the integer part of quotient	20\6	3	
	Mod	Seeking mode, returns the remainder	20Mod6	2	
	+	summation	50+50.1	100.1	
	-	differencing	50-50.1	-0.1	
String operator	&	Connects two strings (& has the function to automatically covert to string operations)	"stu"&"dent"，"1"&2	student，12	
	+		"1"+"2"	12	
Relational (comparison) operators	<	Less than	4<7	True	
	<=	Less than or equal to	39<=10	False	
	>	Greater than	"abc">"abced"	False	
	>=	Greater than or equal to	39>=10	True	
	=	Equal to	"A"="a"	False	
	<>	Not equal to	9<>8	True	
Logical operator	Not	“Not” operations (performs logical negation operations to the expression to the right)	Not 39>10 Not 10>39	False True	
	And	“And” operations (if the conditional operations of both sides of the And operator are valid simultaneously, it	39>10 And 8>10	False	

		will return True; or else it will return False)			
	Or	“Or” operations (if the conditional operations of both sides of the Or operator are invalid simultaneously, it will return False; or else it will return True)	39>10 Or 8>10	True	
	Xor	“Xor” operations	1 Xor 1 0 Xor 1 1 Xor 0 1 Xor 1	False True True False	
	Eqv	“Equivalent” operations	0 Eqv 0 0 Eqv 1 1 Eqv 0 1 Eqv 1	True False False True	
	Imp	“Implication” operations	0 Imp 0 0 Imp 1 1 Imp 0 1 Imp 1	True True False True	

Expression editing window used for configuring animations:



Conditional expression editing window used for configuring animations and conditional programs:



The meanings of each item in the window are as follows (the “Variable dictionary” node is selected in the figure):

1. Project object tree index window: Includes objects such as system variables, project variable dictionary, alarm and communication etc.

2. System function tree index window: Includes system functions.

3. Object list: Display area of the list of variable and other objects; double-clicking an object allows it to be used.

4. Object properties list (this list is displayed according to the object selected): The display area of the object property list; double-clicking an object property allows it to be used.

5. Expression editing window: Allows editing of expressions.

6. Expression operator and digital button area: Selected operators and digits etc. can be used.

7. GUI Development

7.1 Overview

Graphical interface development refers to draw vividly field facilities, equipment, site structure and instruments etc. on the DIAView, forming a stimulation picture of the industrial field. It can even configure and connect data acquisition equipments such as field equipment and instruments to display the data on the window in real-time, making it easy for workers to monitor the field status. Simulator switches and controllers etc in the window interface can also be used to transmit commands and control field equipment .

Graphical interface development is an important part to realize monitoring and control systems; it is the core of the project that integrates various resources of the system to achieve function requirements of the project. Its effect is to make the window become the platform to display system information in realtime and allow system administrators to perform system operation controls.

The basic component of the graphical interface is called “graphics” or “graphic objects”. The DIAView window provides a drawing sketchpad and tools for drawing, including basic graphic components, control units and graphic models etc.; it also provides windows to configure graphic object properties, animations and events, providing an operating platform for graphical interface development.

7.2 Window

In the DIAView, windows are core components to perform field stimulation picture drawing, parameter configuration and data display; it is the basis to achieve DIAView visualization.

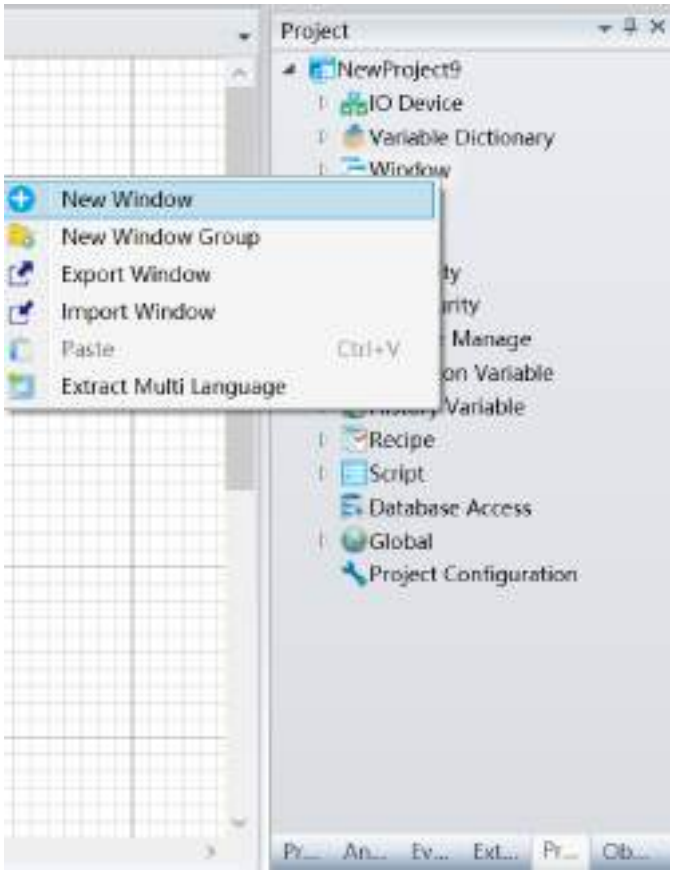
7.2.1 Add window

In the DIAView, windows are core components to perform field stimulation picture drawing,

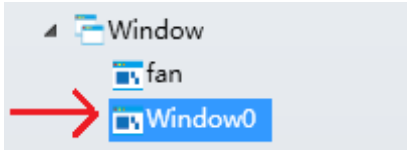
parameter configuration and data display; it is the basis to achieve DIAView visualization.

Steps to add window:

Right-click on the “Window” node in the project window tree index and then click the “New Window” item in the right-click menu, as shown in the figure below:



After clicking “New Window”, the system will generate a window sub-node under the “Window” node using a default name, and the new window will be opened in the sketchpad work area, as shown in the figure below:

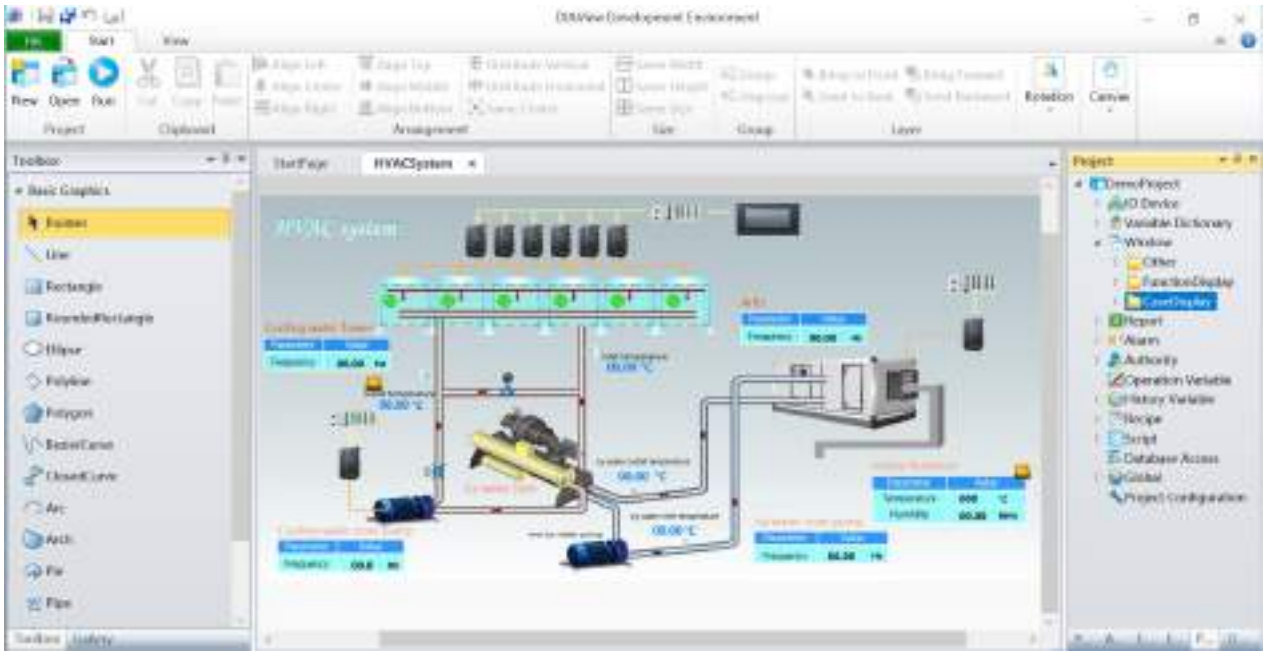


7.2.2 Window operation

1.Open window

When a new window is added, it will automatically be opened at the sketchpad work area;

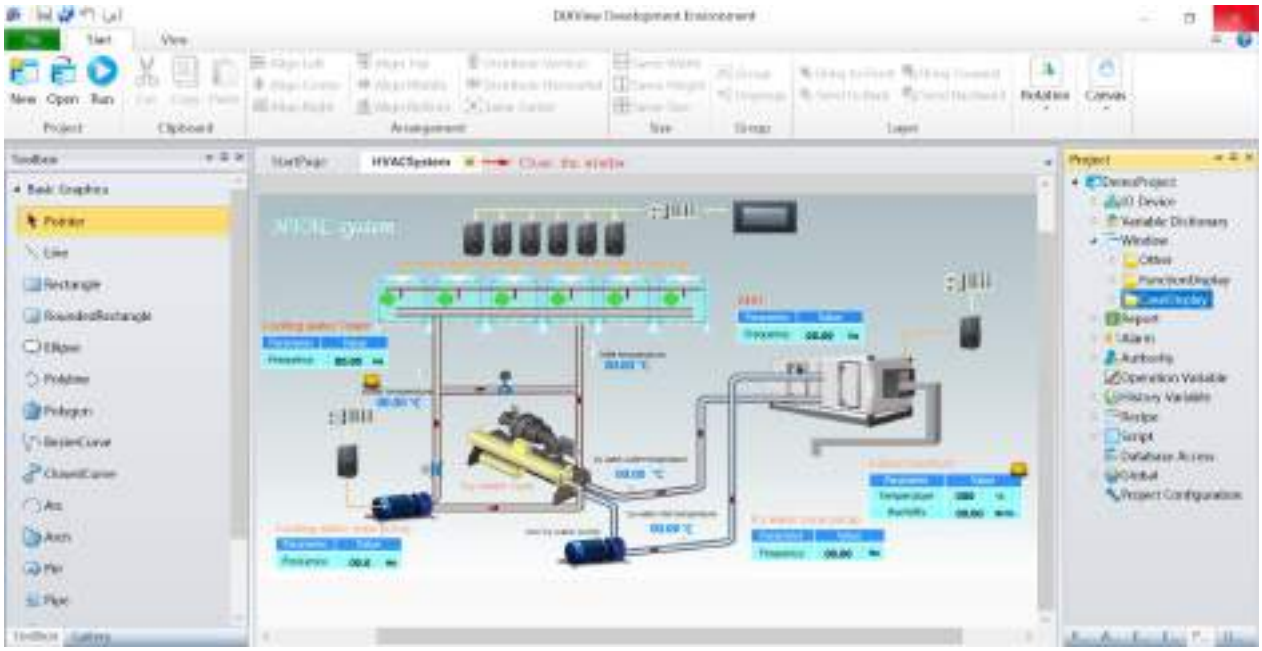
If the window already exists in the “Window” node, a window sub-node can be opened by double clicking on it. Several windows can be opened simultaneously in the sketchpad work area, as shown in the figure below:



2.Close window

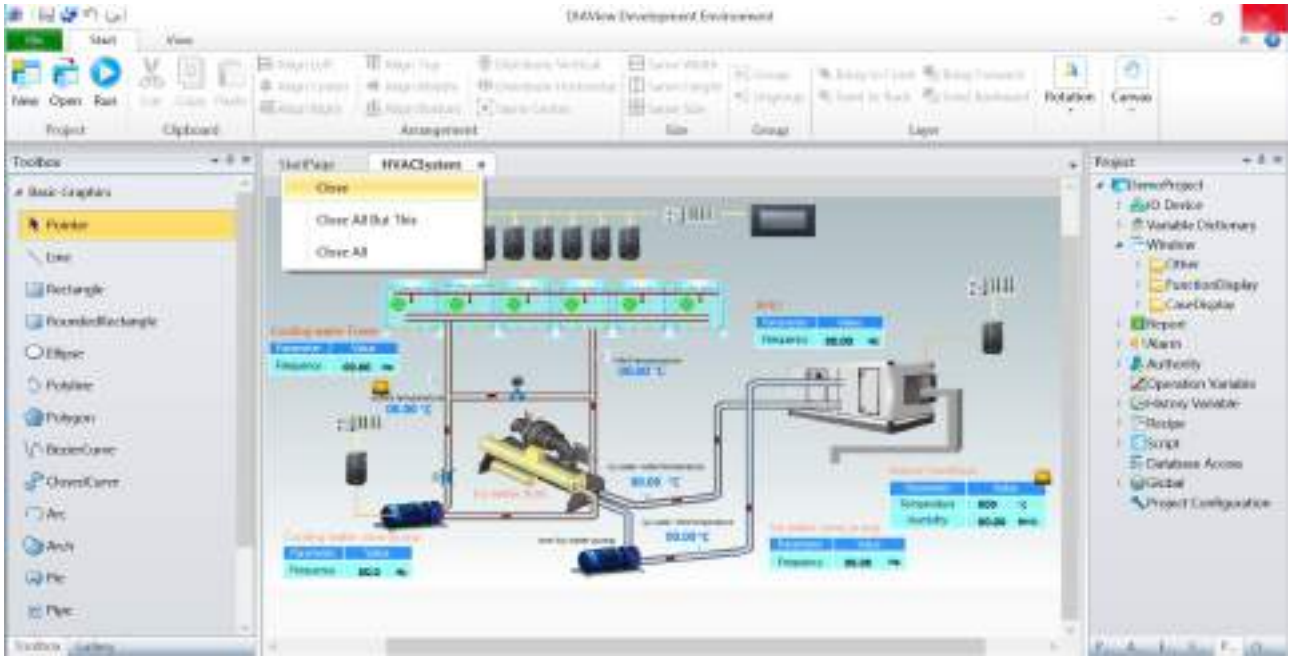
There are several ways to close a window:

Method 1: To close an opened window in the sketchpad work area, simply click the“ ✕ ”to the right of the window name:



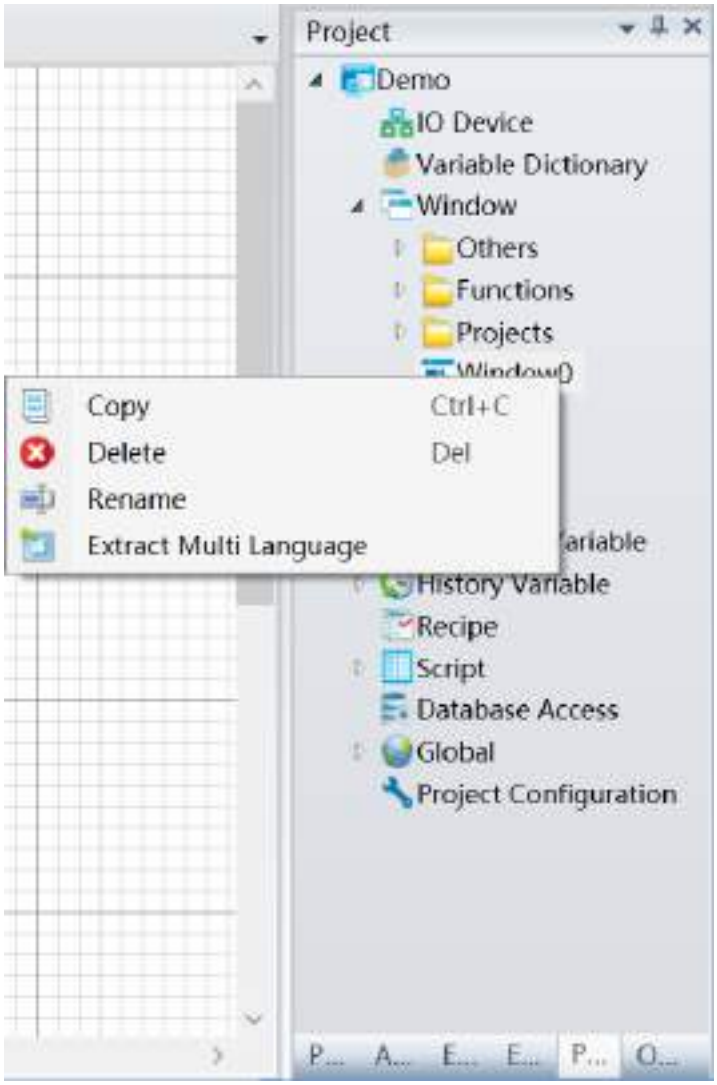
Method 2: In the sketchpad work area, select the closing method (3 types) from the field to the right of the opened window:

1. Close: Closes the current window
2. Close All But This: Closes all other windows besides the current window
3. Close All: Closes all windows

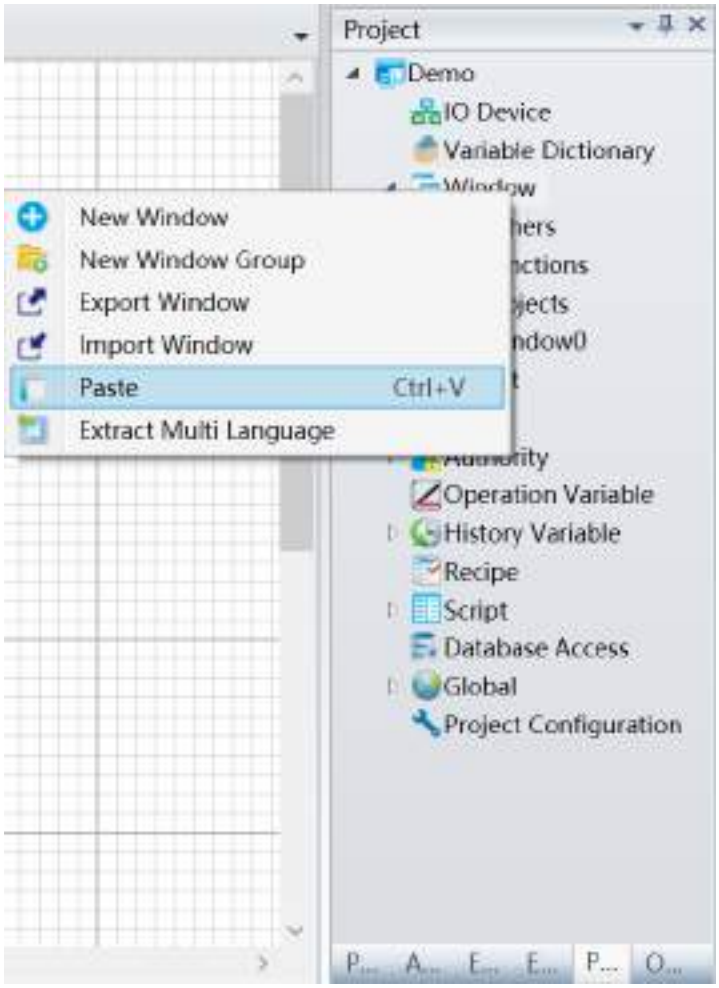


3.Copy window

Select the window node to copy in the project window tree index and right-click on it, and the select the “Copy” item from the right-click menu, completes the copy window,as shown in the example in the figure below:

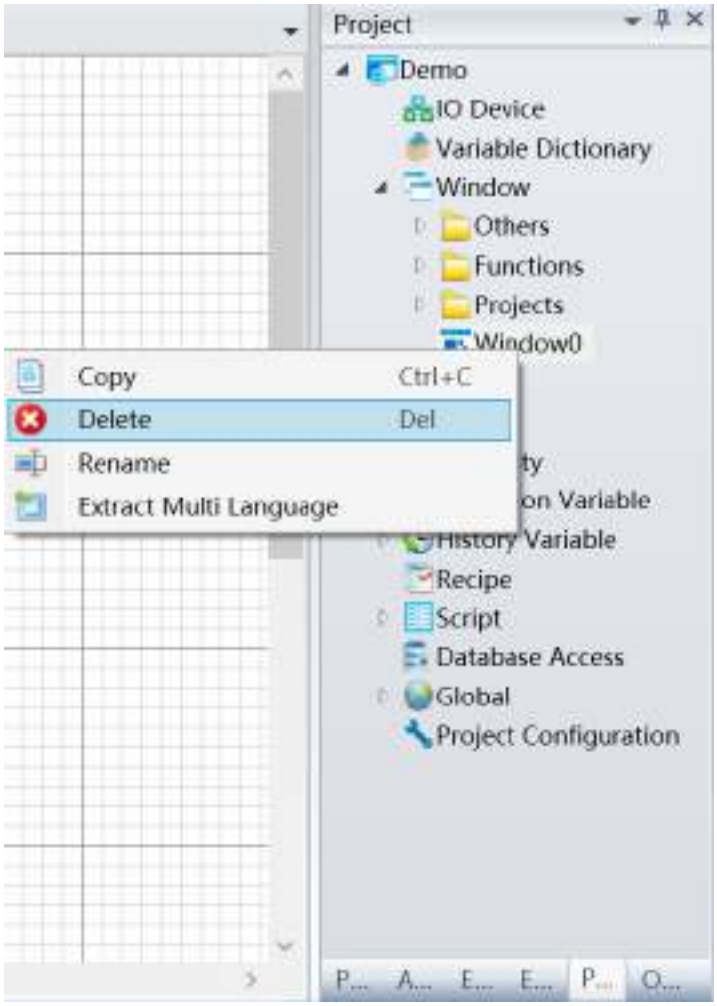


If users want to paste it,click the right-key at the "Window" root node,select the "Paste" item from the right-click menu,as shown in the figure below:



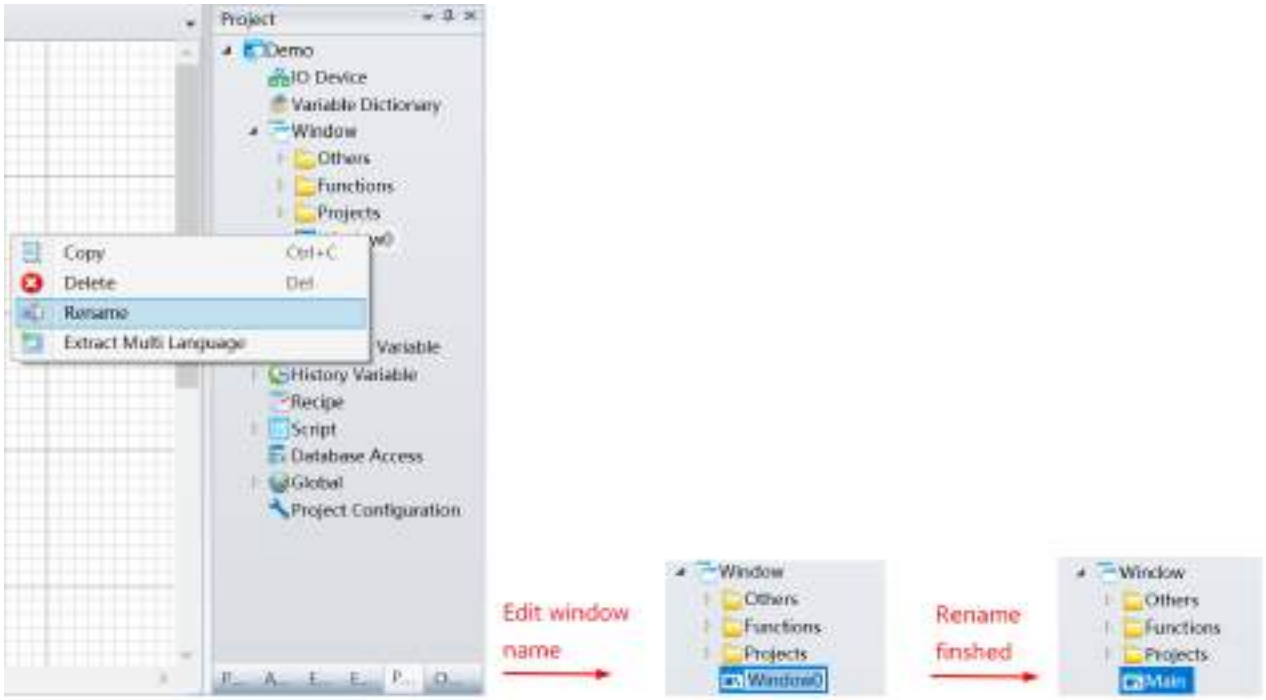
4.Delete window:

Select the window node to delete from the project window tree index and right-click on it, then select the “Delete” item from the right-click menu to delete the window:



5.Rename window

Select the window node to rename in the project window tree index and right-click on it, and the select the “Rename” item from the right-click menu. The window name will become an editable status and just input the new window name, as shown in the example in the figure below:



✧ **Window naming rules:**

(1) Composed of English letters, numbers, Chinese characters and underscore, and can only begin with an English letter or Chinese character;

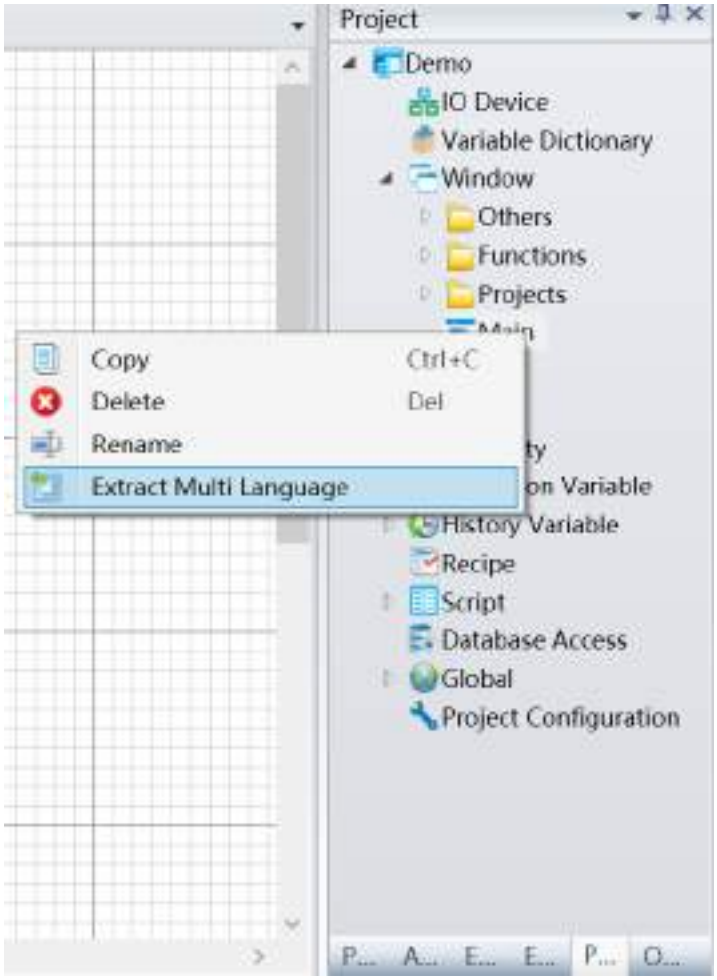
(2) Not case sensitive;

(3) The length cannot exceed 200 characters and cannot exceed 25 Chinese characters;

(4) The names of windows under the same node or window group cannot be repeated.

6.Extract Multi Language:

Extracts strings, images, and sounds from a window into a multilingual resource. In the project window tree directory, select the window node to be operated, right-click, and click "extract multi-language" in the right-click menu, as shown in the figure below:

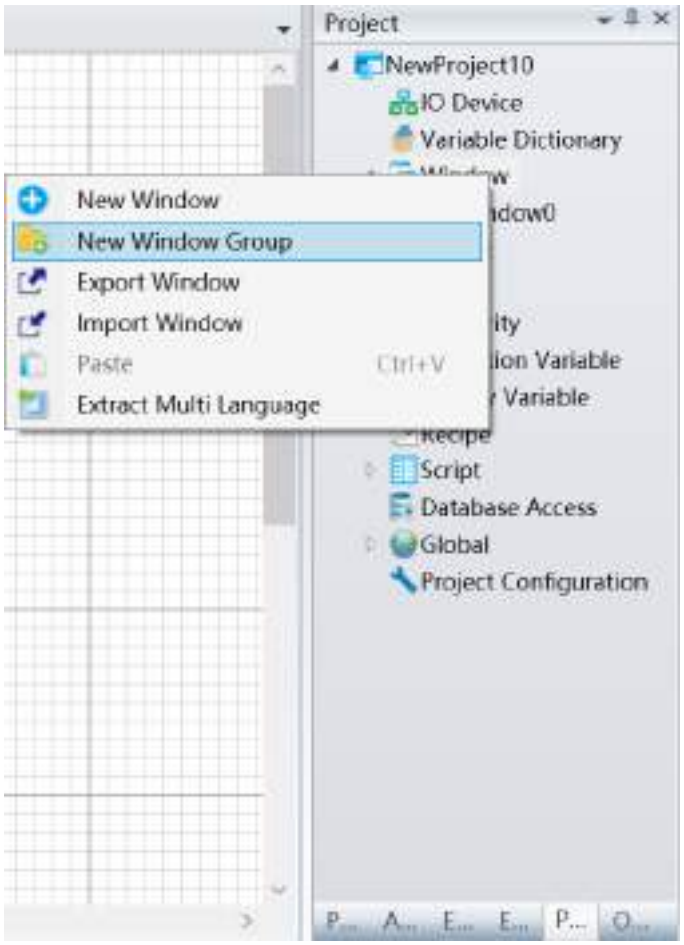


7.2.3 Add window group

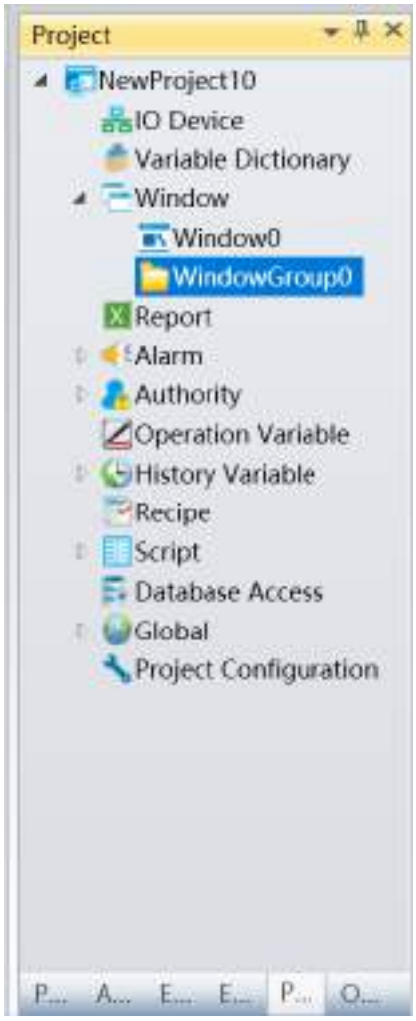
The window group is a unit to perform grouped management of windows; group management can be performed to the different types and uses of windows in the project.

Add new window group steps:

Step 1: Right-click on the “Window” node in the project window tree index and then click the “New Window Group” item in the right-click menu, as shown in the figure below:



Step 2: After clicking “New Window Group”, the system will generate a window sub-node under the “Window” node using a default name, as shown in the figure below:

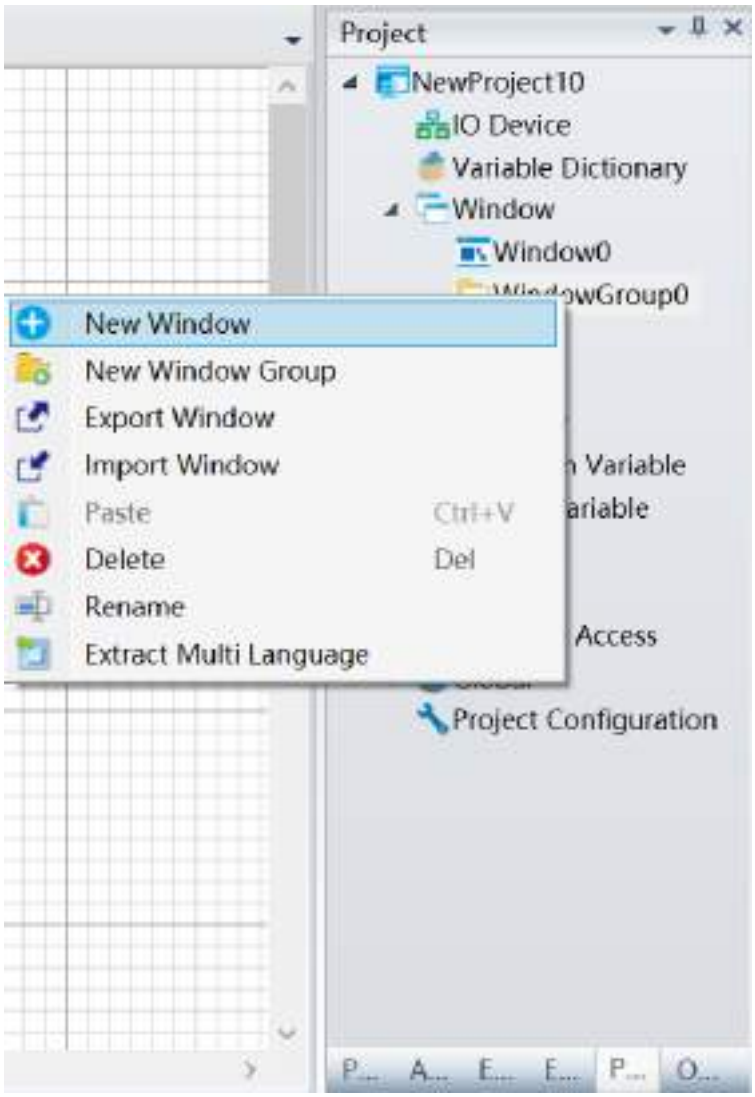


7.2.4 Window group operation

➢ **New Window**

Multiple windows can be added under a window group; select the window group node, right-click on it and then click the “New Window” item

from the right-click menu, as shown in the figure below:



➢ **New Window Group**

This refers to nest a created window group under a window group; select a window group node, right-click on it and then select the “New Window Group” item from the right-click menu.

➢ **Export Window**

This refers to export a window from a window group; select a window group node, right-click on it and then select the “Export Window” item from the right-click menu.

➢ **Import Window**

This refers to import a window under a window group; select a window group node, right-click on it and then select the “Import Window” item from the right-click menu.

➢ **Paste**

Paste the copied window into the current window group.

➢ **Delete**

Delete a window group; select the window group node to delete, right-click on it and then select the “Delete” item from the right-click menu.

➢ **Rename**

Rename the window group name. Select the window group node to rename, right-click on it and then select the “Rename” item from the right-click menu. The window name will become an editable status and just input the new window name.

✧ **Window group naming rules:**

(1) Composed of English letters, numbers, Chinese characters and underscore, and can only begin with an English letter or Chinese character;

(2) Not case sensitive;

(3) The length cannot exceed 200 characters and cannot exceed 25 Chinese characters;

(4) Window group names cannot be repeated under similar class nodes within a project;

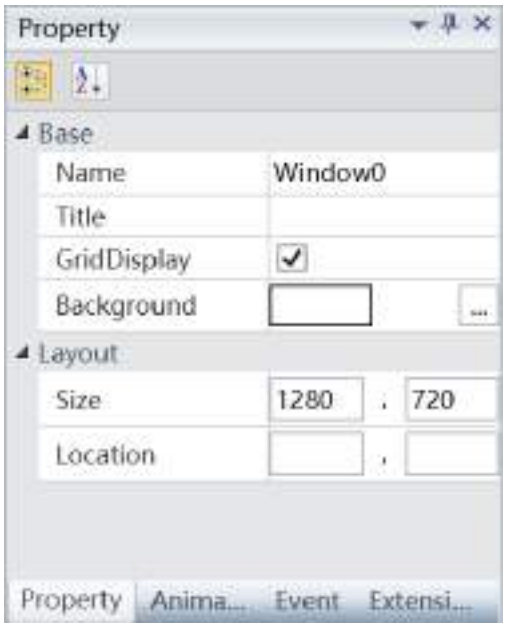
(5) If there are windows and window groups under similar class nodes within a project, then the names of the windows and window groups cannot be repeated.

➢ **Extract Multi Language:**

Extracts strings, images, and sounds from a window into a multilingual resource. In the project window tree directory, right-click to select the window group node to operate on, and in the right-click menu, click the extract multilingual item.

7.3 Window basic properties

Once a window is opened, click on an empty space on the sketchpad to display the properties of the window in the “Property window”, as shown in the figure below:



Basic properties of windows:

✧ **Name:** Name of the window; it is the same name as the window node in the project window tree index.

✧ **Grid display:** Whether to display grid of the window sketchpad. Displaying grids makes it easier to align and arrange the positions of graphic objects for graphic interface development. Grids are only visible in the development environment, and are invisible during system execution.

✧ **Background:** Sets the background color of the sketchboard. Fill styles include: monochrome, icons, images, gradient brushes and radiation brushes etc.

✧ **Title:** Displays the title name of this window as a dialog box. For example, execute the script HMICmd.OpenDialogWindow ("Window0").

✧ **Location:** Sets the display location of the window during execution.

✧ **Size:** Sets the width and height (unit: pixels) of the window.

7.4 Graphic universal properties

All graphic objects have shared properties including names, coordinates, sizes and display etc., and some graphic objects have their own unique properties (graphic properties listed in section 7.5 and 7.6 are unique properties of those graphics). Universal properties of graphic objects in the DIAView software are as follows:

✧ **Name**: Name of the graphic in the sketchpad; the names of graphic objects in the same sketchpad cannot be repeated.

Naming rules:

(1) Can include English letters, numbers, Chinese characters and underscore.

(2) Can only begin with English letters or Chinese characters.

(3) Not case sensitive, but cannot be empty.

(4) The length cannot exceed 200 characters and cannot exceed 25 Chinese characters.

(5) Cannot use the same name as the window where the graphic object is located.

✧ **Display:** Whether to display or hide the graphic during execution.

✧ **Lock:** Whether to lock the graphic; once it is locked, mouse operations will become invalid, operations can only be performed to other properties of that graphic from the property frame. Dotted line frame will appear around the graphics that are locked.

✧ **Security zone:** Sets operating authorities for the graphic; allows the graphic to belong to one or several security zone in order to perform user rights management.

✧ **Fill color:** Sets the fill color inside the graphic.

✧ **Fill level direction:** The horizontal fill direction when filling inside the graphic (from left to right/from right to left/from middle to the edges).

✧ **Fill level ratio:** The graphic width ratio of the part to be filled; ranges between 0 to 1 up to two decimal places. The default value is 1.

✧ **Vertical fill direction:** The vertical fill direction when filling inside the graphic (from top to bottom/from bottom to top/from middle to the edges).

✧ **Vertical fill ratio:** The graphic height ration of the part to be filled; ranges between 0 to 1 up to two decimal places. The default value is 1.

✧ **Coordinates:** The X coordinate and Y coordinate of the graphic in the sketchboard.

✧ **Size:** The width and height of the graphic (unit: pixels).

✧ **Horizontal distortion:** The distortion level of the graphic in the horizontal direction; ranges between - 80 to 80 up to two decimal places. The default value is 0.

✧ **Vertical distortion:** The distortion level of the graphic in the vertical direction; ranges between -80 to 80 up to two decimal places. The default value is 0.

- ✧ **Horizontal zoom:** The zooming level of the graphic horizontally; ranges between 0.1 to 100 up to two decimal places. The default value is 1.
- ✧ **Vertical zoom:** The zooming level of the graphic vertically; ranges between 0.1 to 100 up to two decimal places. The default value is 1.
- ✧ **Horizontal offset:** Graphic position offsets to the left or right horizontally; a positive value offsets it to the right and a negative value offsets it to the left. The default value is 0.
- ✧ **Vertical offset:** Graphic position offsets to the top or bottom vertically; a positive value offsets it to the bottom and a negative value offsets it to the top. The deafault value is 0.
- ✧ **Center:** The ratio between the X and Y coordinates of the center of the graphic to the width and height of the graphic. The default value is 0.5,0.5.
- ✧ **Rotation angle:** Rotation angle of the graphic (unit: degrees).
- ✧ **Line:** When the graphic has border lines, it sets the properties of the boarder lines including the style, thickness and color etc.
- ✧ **Transparency:** Sets the transparency of the graphic; ranges between 0 to 1 up to two decimal places. The default value is 1.
- ✧ **Background color:** Sets the background color of the graphic.
- ✧ **Tooltip:**Sets the content of the prompt when the mouse hovers over the graphics,Support for multilingual functionality.

Noted: Supporting modifying properties of mutiple itens at the same time

7.5 Properties and rendering of basic graphics

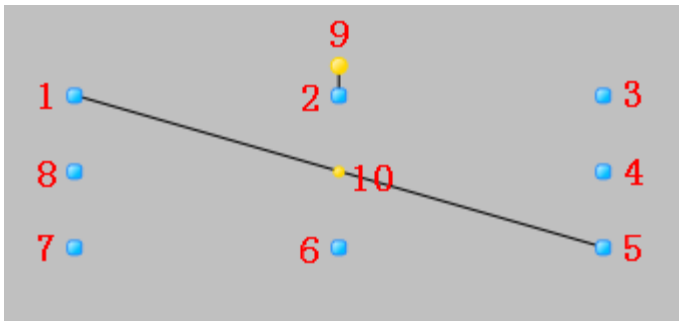
7.5.1 Line

- ➢ Open the window interface and click on “Toolbox” → “Basic Graphics” → “Line” in the tools window to the left; the toolbar as shown in the figure above.
- ➢ Move the mouse to the working area of the window, select a starting point and press the left mouse button and drag the mouse (the straight line is already drawn at this point). If press the shift key while dragging the mouse, it would be allowed to draw the line with mutiples of 15°. After releasing the left mouse button, the drawing of a straight line is complete, and a straight line will be

generated.

➢ Just repeat the steps above if another straight line needs to be drawn.

Graphic introduction:



☞ The figure above is a straight line under editing status; press the stright line with the mouse under non-editing status to enter the editing status.

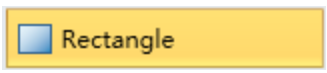
☞ Points 1-8 in the figure are the tensile points to adjust the graphic size; 9 is the rotation point and 10 is the center.

☞ Above point 2 and below point 6 in the figure are the horizontal distortion points. The right of point 4 and the left of point 8 are the vertical distortion points.

Straight line properties

✧ For property settings please refer to the section “7.4 Graphic universal properties”.

7.5.2 Rectangle

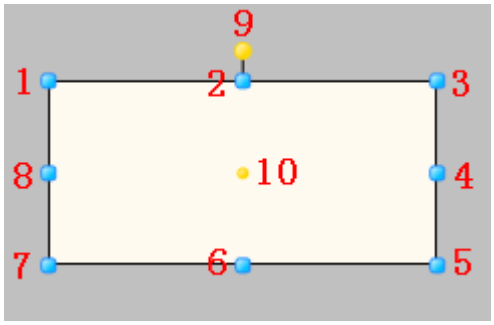


➢ Open the window interface and click on “Toolbox” → “Basic Graphics” → “Rectangle” in the tools window to the left. The toolbar is as shown in the figure above.

➢ Move the mouse to the working area of the window, select a starting point and press the left mouse button and drag the mouse towards the bottom-right (the rectangle is already drawn at this point). After releasing the left mouse button, the drawing of a rectangle is complete, and a rectangle will be generated.

➢ Just repeat the steps above if another rectangle needs to be drawn.

Graphic introduction:



☞ The figure above is a selected rectangle; click the mouse on the rectangle when under the unselected status to enter the selected status.

☞ Points 1-8 in the figure are the tensile points to adjust the graphic size; 9 is the rotation point and 10 is the center.

☞ Above point 2 and below point 6 in the figure are the horizontal distortion points. The right of point 4 and the left of point 8 are the vertical distortion points.

Rectangle properties

✧ For property settings please refer to the section “7.4 Graphic universal properties”.

7.5.3 RoundedRectangle

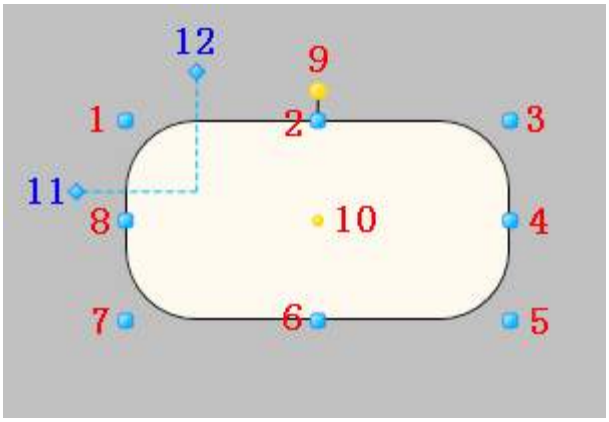
➢ Open the window interface and click on “Toolbox” → “Basic Graphics” → “RoundedRectangle” in the tools window to the left; the toolbar is as shown in the figure above.

➢ Move the mouse to the working area of the window, select a starting point and press the left mouse button and drag the mouse towards the bottom-right (the rounded rectangle is already drawn at this point). After releasing the left mouse button, the drawing of a rounded rectangle is complete, and a rounded rectangle will be generated.

- Adjusting the round angles of the rounded rectangle: Move the mouse on top of the blue diamond adjustment point and then left-click on it and drag the adjustment point; the closer you drag it to the center of the graphic the larger the round angle will be, and when the adjustment point lines on the edge of the graphic it will become a rectangle.

- Just repeat the steps above if another rounded rectangle needs to be drawn.

Graphic introduction:



- The figure above is a selected rounded rectangle; click the mouse on the rounded rectangle when under the not selected status to enter the selected status.

- Points 1-8 in the figure are the tensile points to adjust the graphic size; 9 is the rotation point and 10 is the center, 11 and 12 are the adjustment points to adjust the shape of the round angle.

- Above point 2 and below point 6 in the figure are the horizontal distortion points. The right of point 4 and the left of point 8 are the vertical distortion points.

Rounded rectangle properties

- Arc of the horizon: Sets the ratio between the radius of the round angle and the width of the graphic (value range: 0 ~ 100).

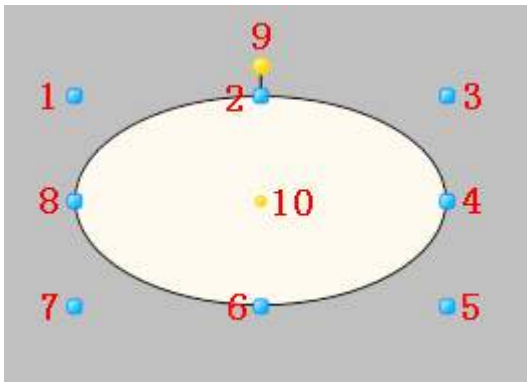
- Arc of the vertical: Sets the ratio between the radius of the round angle and the height of the graphic (value range: 0 ~ 100).

- For other property settings please refer to the section “7.4 Graphic universal properties”.

7.5.4 Ellipse

- Open the window interface and click on “Toolbox” → “Basic Graphics” → “Ellipse” in the tools window to the left; the toolbar is as shown in the figure above.
- Move the mouse to the working area of the window, select a starting point and press the left mouse button and drag the mouse towards the bottom-right (the ellipse is already drawn at this point). After releasing the left mouse button, the drawing of a ellipse is complete, and a ellipse will be generated.
- Just repeat the steps above if another Ellipse needs to be drawn.

Graphic introduction:

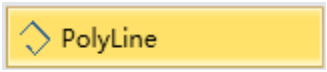


- The figure above is a selected ellipse; click the mouse on the ellipse when under the not selected status to enter the selected status.
- Points 1-8 in the figure are the tensile points to adjust the graphic size; 9 is the rotation point and 10 is the center.
- Above point 2 and below point 6 in the figure are the horizontal distortion points. The right of point 4 and the left of point 8 are the vertical distortion points.

Ellipse properties

- For property settings please refer to the section “7.4 Graphic universal properties”.

7.5.5 Polyline

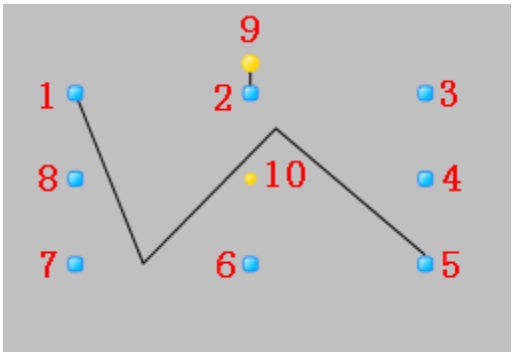


- Open the window interface and click on “Toolbox” → “Basic Graphics” → “Polyline” in the tools window to the left; the toolbar is as shown in the figure above.

- Move the mouse to the working area of the window, select a starting point and click the mouse. Move the mouse to another point and left-click the mouse (the polyline is drawn at this point), left-click the mouse to generate a polyline point of the polyline and draw polyline points according to your needs. If press the shift key while moving the mouse, it would be allowed to draw the polyline with mutiples of 15°. Double-click the left mouse button to finish the drawing of a polyline and generate the polyline.

- Just repeat the steps above if another polyline needs to be drawn.

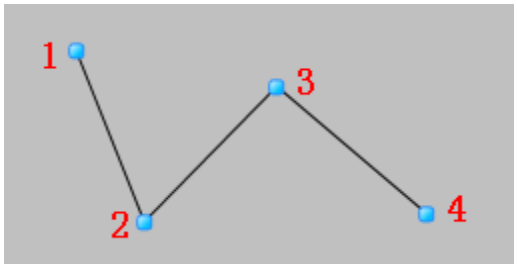
Graphic introduction:



- The figure above is a selected polyline; click the mouse on the polyline when under the not selected status to enter the selected status.

- Points 1-8 in the figure are the tensile points to adjust the graphic size; 9 is the rotation point and 10 is the center.

- Above point 2 and below point 6 in the figure are the horizontal distortion points. The right of point 4 and the left of point 8 are the vertical distortion points.



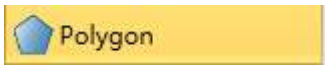
☞ The figure above is an adjustable polyline; double-click the mouse on the polyline when under the unadjustable status to enter the adjustable status.

☞ In figure, 1 is the starting point, 2 and 3 are the polyline points and 4 is the ending point; when the mouse is moved on top of a point, the mouse cursor will change to“ ”。Press and hold the left mouse button and move the position of the point to adjust the shape of the polyline.

Polyline properties

✧ For property settings please refer to the section “7.4 Graphic universal properties”

7.5.6 Polygon

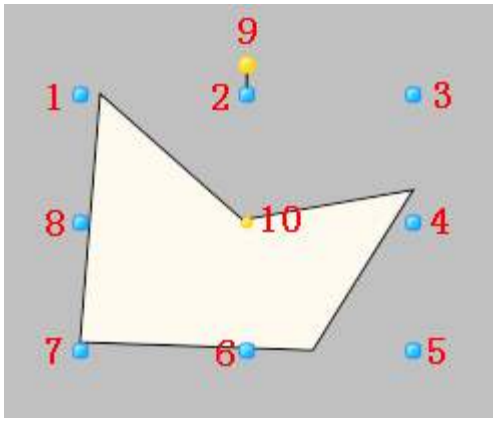


➢ Open the window interface and click on “Toolbox” → “Basic Graphics” → “Polygon” in the tools window to the left; the toolbar is as shown in the figure above.

➢ Move the mouse to the working area of the window, select a starting point and click the mouse. Move the mouse to another point and left-click the mouse to draw one side of the polygon (the polygon is drawn at this point), draw the inflections according to the shape of the polygon required. Every time the left mouse button is pressed, the inflection of one side will be drawn. Double-click the left mouse button to end the drawing of a polygon and generate the polygon.

➢ Just repeat the steps above if another polygon needs to be drawn.

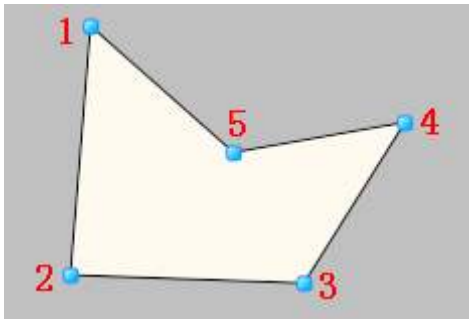
Graphic introduction:



☞ The figure above is a selected polygon; click the mouse on the polygon when under the not selected status to enter the selected status.

☞ Points 1-8 in the figure are the tensile points to adjust the graphic size; 9 is the rotation point and 10 is the center.

☞ Above point 2 and below point 6 in the figure are the horizontal distortion points. The right of point 4 and the left of point 8 are the vertical distortion points.



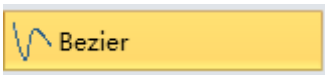
☞ The figure above is an adjustable polygon; double-click the mouse on the polygon when under the unadjustable status to enter the adjustable status.

☞ In the figure, 1 is the starting point, 2 and 3 are the inflection points and 4 is the ending point; when the mouse is moved on top of a point, the mouse cursor will change to “ ”.Press and hold the left mouse button and move the position of the point to adjust the shape of the polygon.

Polygon properties

✧ For property settings please refer to the section “7.4 Graphic universal properties”.

7.5.7 Bezier

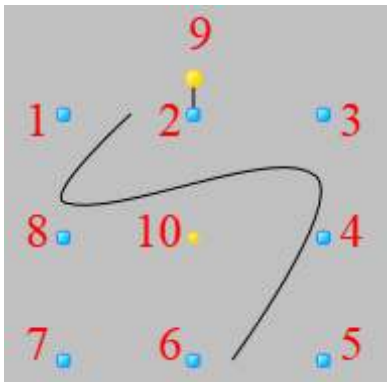


➢ Open the window interface and click on “Toolbox” → “Basic Graphics” → “BezierCurve” in the tools window to the left; the toolbar is as shown in the figure above.

➢ Move the mouse to the working area of the window, select a starting point and press the left mousebutton and drag the mouse towards the bottom-right (the BezierCurve is already drawn at this point). After releasing the left mouse button, the drawing of a BezierCurve is complete, and a BezierCurve will be generated.

➢ Just repeat the steps above if another bezier needs to be drawn.

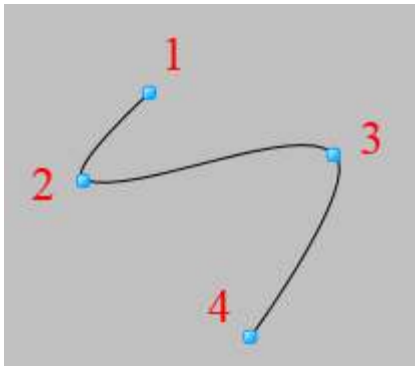
Graphic introduction:



☞ The figure above is a selected bezier; click the mouse on the bezier when under the not selected status to enter the selected status.

☞ Points 1-8 in the figure are the tensile points to adjust the graphic size; 9 is the rotation point and 10 is the center.

☞ Above point 2 and below point 6 in the figure are the horizontal distortion points. The right of point 4 and the left of point 8 are the vertical distortion points.



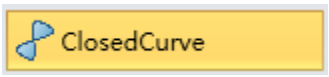
☞ The figure above is an adjustable bezier; double-click the mouse on the polygon when under the unadjustable status to enter the adjustable status.

☞ The figure, 1 is the starting point, 2 and 3 are the inflection points and 4 is the ending point; when the mouse is moved on top of a point, the mouse cursor will change to“ ”， Press and hold the left mouse button and move the position of the point to adjust the shape of the bezier.

Bezier properties

✧ For property settings please refer to the section “7.4 Graphic universal properties”.

7.5.8 ClosedCurve

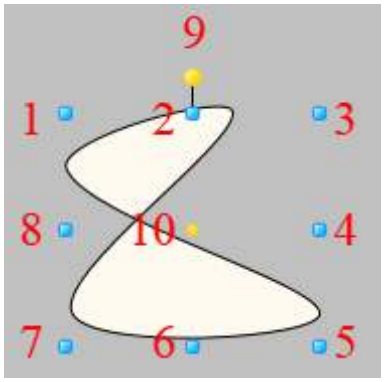


➢ Open the window interface and click on “Toolbox” → “Basic Graphics” → “ClosedCurve” in the tools window to the left; the toolbar is as shown in the figure above.

➢ Move the mouse to the working area of the window, select a starting point and click the mouse. Move the mouse to another point and left-click the mouse to draw one side of the closed curve (the closed curve is drawn at this point), draw the inflections according to the shape of the closed curve required. Every time the left mouse button is pressed, the inflection of one side will be drawn. Double-click the left mouse button to end the drawing of a closed curve and generate the closed curve.

➢ Just repeat the steps above if another closed curve needs to be drawn.

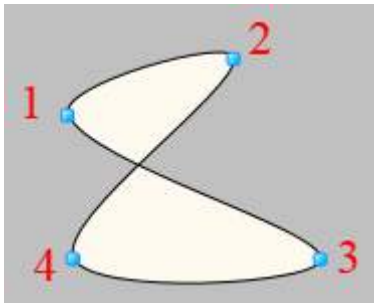
Graphic introduction:



☞ The figure above is a selected closed curve; click the mouse on the closed curve when under the not selected status to enter the selected status.

☞ Points 1-8 in the figure are the tensile points to adjust the graphic size; 9 is the rotation point and 10 is the center.

☞ Above point 2 and below point 6 in the figure are the horizontal distortion points. The right of point 4 and the left of point 8 are the vertical distortion points.



☞ The figure above is an adjustable closed curve; double-click the mouse on the closed curve when under the unadjustable status to enter the adjustable status.

☞ In the figure, 1 is the starting point, 2 and 3 are the inflection points and 4 is the ending point; when the mouse is moved on top of a point, the mouse cursor will change to“ ”.Press and hold the left mouse button and move the position of the point to adjust the shape of the polygon.

Polygon properties

✧ For property settings please refer to the section "7.4 Graphic universal properties".

7.5.9 Arc

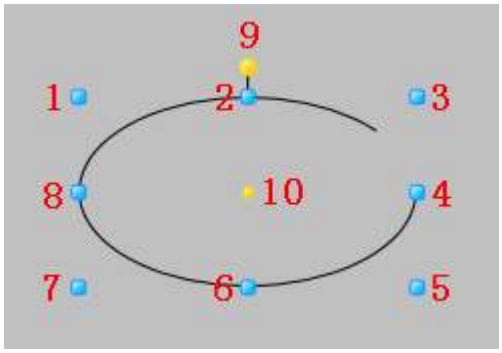
➢ Open the window interface and click on “Toolbox” → “Basic Graphics” → “Arc” in the tools window to the left; the toolbar is as shown in the figure above.

➢ Move the mouse to the working area of the window, select a starting point and press the left mouse button and drag the mouse towards the bottom-right (the arc is already drawn at this point). After releasing the left mouse button, the drawing of an arc is complete, and an arc will be generated.

➢ Adjusting the lenth of the arc: Double click the arc to enter the adjustable status; place the mouse on any of the yellow diamond adjustment point, press and hold the left mouse button to drag the adjustment point to change the length of the arc.

➢ Just repeat the steps above if another arc needs to be drawn.

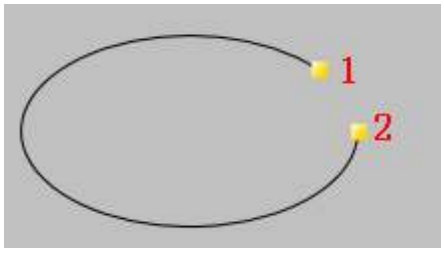
Graphic introduction:



☞ The figure above is a selected arc; click the mouse on the arc when under the not selected status to enter the selected status.

☞ Points 1-8 in the figure are the tensile points to adjust the graphic size; 9 is the rotation point and 10 is the center.

☞ Above point 2 and below point 6 in the figure are the horizontal distortion points. The right of point 4 and the left of point 8 are the vertical distortion points.



☞ The figure above is an adjustable arc; double-click the mouse on the arc when under the unadjustable status to enter the adjustable status.

☞ Points 1 and 2 in the figure are adjustment points to adjust the length of the arc; when the mouse moves on top of a point, the mouse cursor will change to“ ”， Press and hold the left mouse button and drag the position of the adjustment point to adjust the length of the arc.

Arc properties

✧ Starting angle: Sets the starting angle of the arc length (unit: degrees).

✧ Scan angle: Sets the ending angle of the arc length (unit: degrees).

✧ For other property settings please refer to the section “7.4 Graphic universal properties”.

7.5.10 Arch

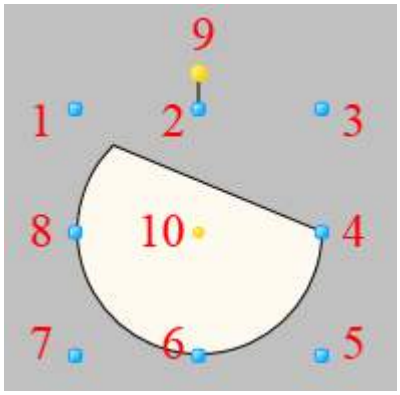
➢ Open the window interface and click on “Toolbox” → “Basic Graphics” → “Arch” in the tools window to the left; the toolbar is as shown in the figure above.

➢ Move the mouse to the working area of the window, select a starting point and press the left mouse button and drag the mouse towards the bottom-right (the arch is already drawn at this point). After releasing the left mouse button, the drawing of an arch is complete, and an arch will be generated.

➢ Adjusting the angle of the arc: Double click the arc to enter the adjustable status; place the mouse on any of the yellow diamond adjustment point, press and hold the left mouse button to drag the adjustment point to change the angle of the arc.

➢ Just repeat the steps above if another arch needs to be drawn.

Graphic introduction:

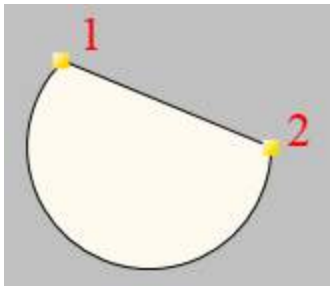


☞ The figure above is a selected arch; click the mouse on the arch when under the not selected status to enter the selected status.

☞ Points 1-8 in the figure are the tensile points to adjust the graphic size; 9 is the rotation point and 10 is the center.

☞ Above point 2 and below point 6 in the figure are the horizontal distortion points. The right of point 4 and the left of point 8 are the

vertical distortion points.



☞ The figure above is an adjustable arch; double-click the mouse on the arch when under the unadjustable status to enter the adjustable status.

☞ Points 1 and 2 in the figure are adjustment points to adjust the angle of the arch; when the mouse moves on top of a point, the mouse cursor will change to“ ”. Press and hold the left mouse button and drag the position of the adjustment point to adjust the angle of the arch .

Arch properties

✧ Starting angle: Sets the starting angle of the fan-shape angle (unit: degrees)

✧ Scan angle: Sets the ending angle of the fan-shape angle (unit: degrees)

✧ For other property settings please refer to the section “7.4 Graphic universal properties”.

7.5.11 Pie

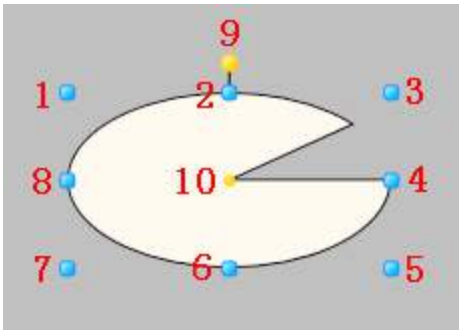
➢ Open the window interface and click on “Toolbox” → “Basic Graphics” → “Pie” in the tools window to the left; the toolbar is as shown in the figure above.

➢ Move the mouse to the working area of the window, select a starting point and press the left mouse button and drag the mouse towards the bottom-right (the pie is already drawn at this point). After releasing the left mouse button, the drawing of a pie is complete, and a pie will be generated.

➢ Adjusting the angle of the pie: Double click the pie to enter the adjustable status; place the mouse on the yellow diamond adjustment point, press and hold the left mouse button to drag the adjustment point to change the angle of the pie.

➢ Just repeat the steps above if another pie needs to be drawn.

Graphic introduction:

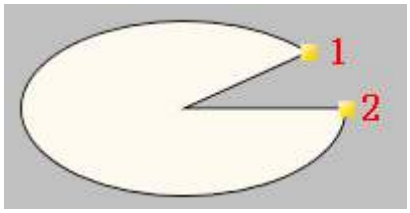


☞ The figure above is a selected pie; click the mouse on the pie when under the not selected status to enter the selected status.

☞ Points 1-8 in the figure are the tensile points to adjust the graphic size; 9 is the rotation point and

10 is the center.

☞ Above point 2 and below point 6 in the figure are the horizontal distortion points. The right of point 4 and the left of point 8 are the vertical distortion points.



☞ The figure above is an adjustable pie; double-click the mouse on the pie when under the unadjustable status to enter the adjustable status.

☞ Points 1 and 2 in the figure are adjustment points to adjust the angle of the pie; when the mouse moves on top of a point, the mouse cursor will change to“ ”.Press and hold the left mouse button and drag the position of the adjustment point to adjust the angle of the pie.

Pie properties

✧ Starting angle: Sets the starting angle of the pie angle (unit: degrees).

✧ Scan angle: Sets the ending angle of the pie angle (unit: degrees).

✧ For other property settings please refer to the section “7.4 Graphic universal properties”.

7.5.12 Pipe

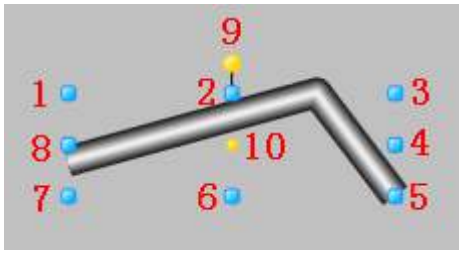
➢ Open the window interface and click on “Toolbox” → “Basic Graphics” → “Pipe” in the tools window to the left; the toolbar is as shown in the figure above.

➢ Move the mouse to the working area of the window, select a starting point and click the mouse. Move the mouse to another point and left-click the mouse (the pipe is drawn at this point), left-click the mouse to generate an inflection point of the pipe and draw inflection points according to shape of the

pipe needed. If press the shift key while moving the mouse, it would be allowed to draw the pipe with mutiples of 15°. Double-click the left mouse button to finish the drawing of a the pipe and generate the pipe.

➢ Just repeat the steps above if another pipe needs to be drawn.

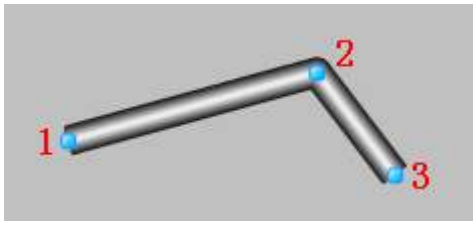
Graphic introduction:



☞ The figure above is a selected pipe; click the mouse on the tube when under the not selected status to enter the selected status.

☞ Points 1-8 in the figure are the tensile points to adjust the graphic size; 9 is the rotation point and 10 is the center.

☞ Above point 2 and below point 6 in the figure are the horizontal distortion points. The right of point 4 and the left of point 8 are the vertical distortion points.



☞ The figure above is an adjustable pipe; double-click the mouse on the pie when under the unadjustable status to enter the adjustable status.

☞ In the figure, 1 is the starting point, 2 is the inflection point to adjust the shape of the pipe and 3 is the ending point; when the mouse is moved on top of a point, the mouse cursor will change to“  ”.Press and hold the left mouse button and move the position of the point to adjust the shape of the pipe.

Pipe properties

✧ Whether to flow positively: Sets the direction which the liquid inside the pipe flows; check it for positive flowing and uncheck for reverse flowing.

✧ Liquid color: Sets the color of the liquid inside the pipe.

✧ Liquid width: Sets the sectional width of the liquid inside the pipe, which is the liquid flow.

✧ Pipe color: Sets the color for the exterior of the pipe.

✧ Pipe width: Sets the diameter of the pipe.

✧ For other property settings please refer to the section “7.4 Graphic universal properties".

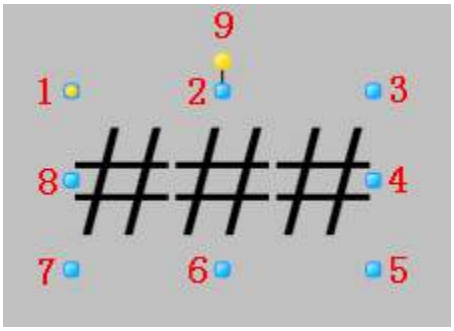
7.5.13 Text

➢ Open the window interface and click on “Toolbox” → “Basic Graphics” → “Text” in the tools window to the left; the toolbar as shown in the figure above.

➢ Move the mouse to the working area of the window, select a starting point and press the left mouse button and drag the mouse towards the bottom-right (the text outline is already drawn at this point);after releasing the left mouse button, the drawing of a text is complete, and a text will be generated.

➢ Just repeat the steps above if another text needs to be drawn.

Graphic introduction:



☞ The figure above is a selected text; click on the text to enter selected status.

☞ There are two points on 1 in the figure above: the tensile point and the center; 2-8 are the tensile points to adjust the graphic size; 9 is the rotation point.

☞ Above point 2 and below point 6 in the figure are the horizontal distortion points. The right of point 4 and the left of point 8 are the vertical distortion points.

Text properties

✧ Content: Text contents of the text,Support for multilingual functionality.

✧ Auto zoom: Sets whether the text content will zoom according to the zooming size of the frame.

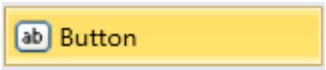
✧ Font: Sets the font format of the text; for example font style, size and bold etc.

✧ Text color: Sets the color of the text.

✧ For other property settings please refer to the section “7.4 Graphic universal properties”.

7.6 Properties and rendering of window control

7.6.1 Button



➢ Open the window interface and click on “Toolbox” → “Window Controls” → “Button” in the tools window to the left; the toolbar is as shown in the figure above.

➢ Move the mouse to the working area of the window, select a starting point and press the left mouse button and drag the mouse towards the bottom-right; after releasing the left mouse button, the drawing of a button is complete, and a button will be generated.

➢ Just repeat the steps above if another button needs to be drawn.

Graphic introduction:

☞ The figure above is a selected button; click on the button to enter selected status.

☞ Points 1-8 in the figure are the tensile points to adjust the graphic size; 9 is the rotation point and 10 is the center.

☞ Above point 2 and below point 6 in the figure are the horizontal distortion points. The right of point 4 and the left of point 8 are the vertical distortion points.

Button properties

✧ Content: The text displayed on the button,Support for multilingual functionality.

✧ Font: Sets the font format of the text displayed on the button; for example font style, size and bold etc.

✧ Text color: Sets the text color of the button's text.

✧ Border color: Sets the color of the button's border.

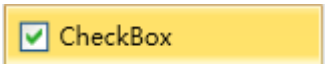
✧ Account Check: Sets the runtime account confirmation properties;None: user account password confirmation is not required;General check: under the running environment, the operation of the control requires an account password confirmation;Double check: under the running environment, each operation of this control requires account password confirmation.

✧ Enable: Sets the availability of the button.

✧ Default style: Set whether to use the default style of button(the default style of button owns some animation,but it may conflict with customer background animation.User can choose depending on the circumstances).

✧ For other property settings please refer to the section "7.4 Graphic universal properties".

7.6.2 CheckBox

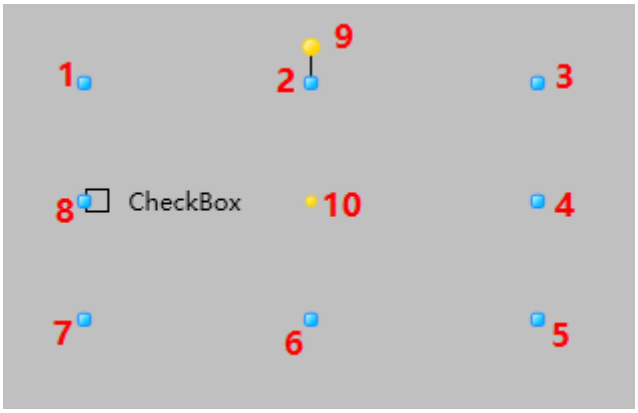


➢· Open the window interface and click on “Toolbox” → “Window Controls” → “CheckBox” in the tools window to the left; the toolbar is as shown in the figure above.

➢· Move the mouse to the working area of the window, select a starting point and press the left mouse button and drag the mouse towards the bottom-right; after releasing the left mouse button, the drawing of a check box is complete, and a check box will be generated.

➢· Just repeat the steps above if another check box needs to be drawn.

Graphic introduction:



☞· The figure above is a selected check box; click on the check box to enter selected status.

☞· Points 1-8 in the figure are the tensile points to adjust the graphic size; 9 is the rotation point and 10 is the center.

☞· Above point 2 and below point 6 in the figure are the horizontal distortion points. The right of point 4 and the left of point 8 are the vertical distortion points.

Check box properties

✧· Content: Text content of the check box options,Support for multilingual functionality.

✧· Selected: Whether to allow the check box to be in a selected status.

✧・ Font: Sets the font format of the check box; for example font style, size and bold etc.

✧・ Text color: Sets the color of the text in the check box.

✧・ Enable: Sets the availability of the check box.

✧・ For other property settings please refer to the section “7.4 Graphic universal properties”.

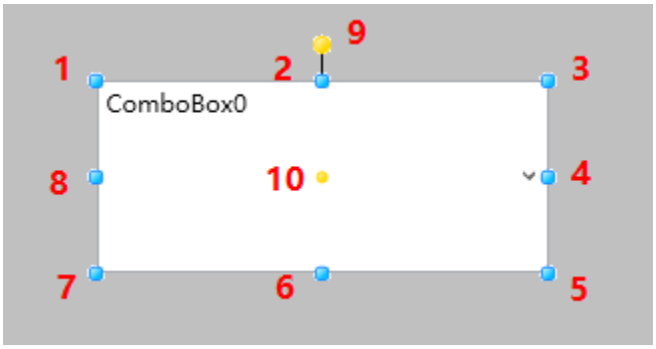
7.6.3 ComboBox

➢・ Open the window interface and click on “Toolbox” → “Window Controls” → “ComboBox in the tools window to the left; the toolbar is as shown in the figure above.

➢・ Move the mouse to the working area of the window, select a starting point and press the left mouse button and drag the mouse towards the bottom-right; after releasing the left mouse button, the drawing of a combo box is complete, and a combo box will be generated.

➢・ Just repeat the steps above if another combo box needs to be drawn.

Graphic introduction:



☞・ The figure above is a combo box; click on the combo box to enter selected status.

☞・ Points 1-8 in the figure are the tensile points to adjust the graphic size; 9 is the rotation point and 10 is the center.

☞・ Above point 2 and below point 6 in the figure are the horizontal distortion points. The right of point 4 and the left of point 8 are the vertical distortion points.

Combo box properties

✧･ Text:Text content displayed in the combo box in a non-executing environment,Support for multilingual functionality.

✧･ Horizontal alignment: Sets the horizontal alignment method for the text content of the combo box in an execution environment.

✧･ Vertical alignment: Sets the vertical alignment method for the text content of the combo box in an execution environment.

✧･ Font: Sets the font format of the text in the combo box; for example font style, size and bold etc.

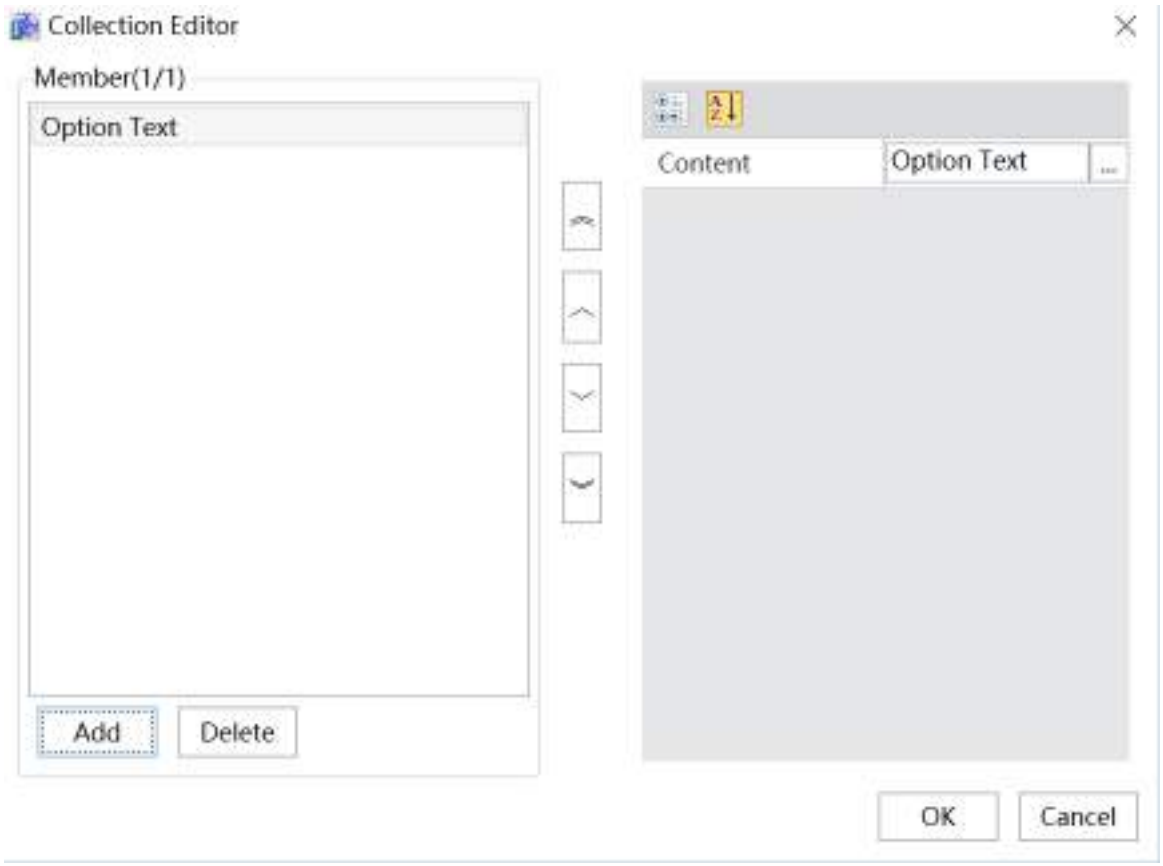
✧･ Text color: Sets the color of the text in the combo box.

✧･ Border color: Sets the color of the border of the combo box.

✧･ Enable: Sets the availability of the combo box.

✧･ Read-only: Sets whether the combo box is read-only; it is read-only if selected.

✧･ Subset: Drop-down option contents of the combo box; click“  ”， and the subset editor shown in the figure below will appear:



(The “Add” button on the left adds a new member; the “Content” on the right allows entering the text content of the member，support for multilingual functionality.)

✧･ Index: Sets the default option of the ComboBox.

✧･ For other property settings please refer to the section “7.4 Graphic universal properties”.

7.6.4 Label

➢･ Open the window interface and click on “Toolbox” → “Window Controls” → “Label” in the tools window to the left; the toolbar is as shown in the figure above.

➢･ Move the mouse to the working area of the window, select a starting point and press the left mouse button and drag the mouse towards the bottom-right; after releasing the left mouse button, the drawing

of a tag is complete, and a tag will be generated.

➢ Just repeat the steps above if another label needs to be drawn.

Graphic introduction:

☞ The figure above is a selected tag; click on the tag to enter selected status.

☞ Points 1-8 in the figure are the tensile points to adjust the graphic size; 9 is the rotation point and 10 is the center.

☞ Above point 2 and below point 6 in the figure are the horizontal distortion points. The right of point 4 and the left of point 8 are the vertical distortion points.

Tag properties

✧ Content: Text contents of the tag,Support for multilingual functionality.

✧ Font: Sets the font format of the text in the tag; for example font style, size and bold etc.

✧ Text color: Sets the color of the text in the tag.

✧ Enable: Sets the availability of the tag.

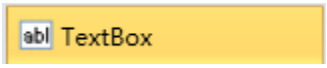
✧ Horizontal alignment: Sets the horizontal alignment of label text.

✧ Vertical alignment: Sets the vertical alignment of label text.

✧ Line feed: Sets the line feed method of the text of the tag (overflow line feed, no execution line feed, perform line feed).

✧ For other property settings please refer to the section “7.4 Graphic universal properties”.

7.6.5 TextBox

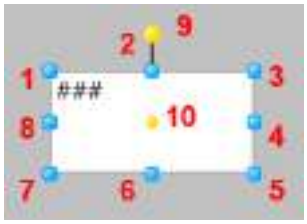


➢ Open the window interface and click on “Toolbox” → “Window Controls” → “TextBox” in the tools window to the left; the toolbar is as shown in the figure above.

➢ Move the mouse to the working area of the window, select a starting point and press the left mouse button and drag the mouse towards the bottom-right; after releasing the left mouse button, the drawing of a text box is complete, and a text box will be generated.

➢ Just repeat the steps above if another text box needs to be drawn.

Graphic introduction:



☞ The figure above is a selected text box; click on the text box to enter selected status.

☞ Points 1-8 in the figure are the tensile points to adjust the graphic size; 9 is the rotation point and 10 is the center.

☞ Above point 2 and below point 6 in the figure are the horizontal distortion points. The right of point 4 and the left of point 8 are the vertical distortion points.

Text box properties

✧ Content: Text contents of the text box,Support for multilingual functionality.

✧ Font: Sets the font format of the text box; for example font style, size and bold etc.

✧ Text color: Sets the color of the text in the text box.

✧ Border color: Sets the color of the text box’s border.

✧ Enable: Sets the availability of the text box

✧･ Line feed: Sets whether it will automatically line feed when entering text in the text box.

✧･ Horizontal alignment: Sets the horizontal alignment of text in a text box.

✧･ Vertical alignment: Sets the vertical alignment of text in a text box.

✧･ Line feed method: Sets the line feed method of the text in the text box; there are 3 ways:

Overflow line feed: When there are both English and Chinese text contents, if the length exceeds the length of the text box, change rows if the content is Chinese. If the content is in English letters, change rows if there is a space between letters and do not change rows if there is no space in between.

No execution row change: Do not change rows when the length of the inputted text exceeds the length of the text box.

Perform row change: Change rows when the length of the inputted text exceeds the length of the text box.

✧･ For other property settings please refer to the section “7.4 Graphic universal properties”.

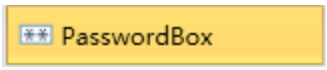
Difference between “text box” and “text”:

1. The text box has frames and the color of the frame and fill color inside can be set; these cannot be done for texts;

2. The line feed method of the text content inside the text box can be set, but line wrap cannot be performed for texts;

3. They have different “animation” configurations; for example “rotating” and “zooming” animations can be configured for text but these cannot be set for text boxes.

7.6.6 PasswordBox

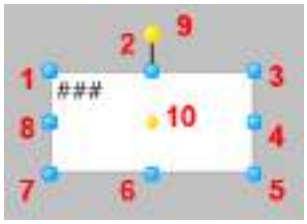


➢･ Open the window interface and click on “Toolbox” → “Window Controls” → “PasswordBox” in the tools window to the left; the toolbar is as shown in the figure above.

➢ Move the mouse to the working area of the window, select a starting point and press the left mouse button and drag the mouse towards the bottom-right. After releasing the left mouse button, the drawing of the password box is complete, and the password box will be generated.

➢ Just repeat the steps above if another passwordbox needs to be drawn.

Graphic introduction:



☞ The figure above is a selected password box; click on the password box to enter selected status.

☞ Points 1-8 in the figure are the tensile points to adjust the graphic size; 9 is the rotation point and 10 is the center.

☞ Above point 2 and below point 6 in the figure are the horizontal distortion points. The right of point 4 and the left of point 8 are the vertical distortion points.

Password box properties

✧ Password: Password content that can be entered into the password box.

✧ Font: Sets the font format of the password box; for example font style, size and bold etc.

✧ Text color: Sets the color of the password box text.

✧ Border color: Sets the color of the border of the password box.

✧ Enable: Sets the availability of the password box.

✧ Horizontal alignment: Sets the horizontal alignment method of the password box text.

✧ Vertical alignment: Sets the vertical alignment method of the password box text.

✧ For other property settings please refer to the section “7.4 Graphic universal properties”.

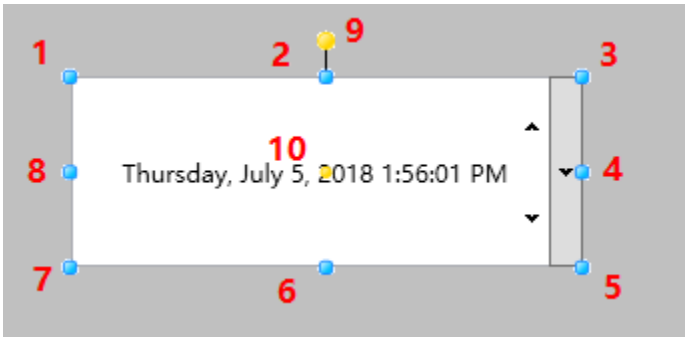
7.6.7 DateTimePicker

➢· Open the window interface and click on “Toolbox” → “Window Controls” → “DateTimePicker” in the tools window to the left; the toolbar is as shown in the figure above.

➢· Move the mouse to the working area of the window, select a starting point and press the left mouse button and drag the mouse towards the bottom-right; after releasing the left mouse button, the drawing of the date time picker is complete, and the date time picker will be generated.

➢· Just repeat the steps above if another date time picker needs to be drawn.

Graphic introduction:



☞· The figure above is a selected date time picker; click on the date time picker to enter selected status.

☞· Points 1-8 in the figure are the tensile points to adjust the graphic size; 9 is the rotation point and 10 is the center.

☞· Above point 2 and below point 6 in the figure are the horizontal distortion points. The right of point 4 and the left of point 8 are the vertical distortion points.

☞· 11 is the text block of the date ,12 is the date increase and decrease button and 13 is the date selection button.

Date time picker properties

✧· Date: Text contents displayed in the date time picker.

✧・ Maximum value: Set the maximum value of the selected range.

✧・ Minimum value: Set the minimum value of the selected range.

✧・ Date format: Sets the display format of the date time picker.

✧・ Alignment: Sets the alignment method of the date time picker text box.

✧・ Time interval: Set the time interval of the date time picker.

✧・ Upper and lower buttton: Set the availability of the date time.

✧・ DisplayCurrentTime.

✧・ Read-only: Sets whether the date time picker is read-only.

✧・ Font: Sets the font format of the text in the date time picker; for example font style, size and bold etc.

✧・ Text color: Sets the color of the text in the date time picker.

✧・ Enable: Sets the availability of the date time picker.

✧・ Alignment: Sets the alignment method of the date time picker.

✧・ For other property settings please refer to the section “7.4 Graphic universal properties”.

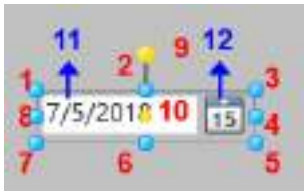
7.6.8 DatePicker

➢・ Open the window interface and click on “Toolbox” → “Window Controls” → “DatePicker” in the tools window to the left; the toolbar is as shown in the figure above.

➢・ Move the mouse to the working area of the window, select a starting point and press the left mouse button and drag the mouse towards the bottom-right; after releasing the left mouse button, the drawing of the date picker is complete, and the date picker will be generated.

➢・ Just repeat the steps above if another date needs to be drawn.

Graphic introduction:



☞ The figure above is a selected date picker; click on the date picker to enter selected status.

☞ Points 1-8 in the figure are the tensile points to adjust the graphic size; 9 is the rotation point and 10 is the center.

☞ Above point 2 and below point 6 in the figure are the horizontal distortion points. The right of point 4 and the left of point 8 are the vertical distortion points.

☞ 11 is the text block of the date and 12 is the date selection button.

Date picker properties

✧ Text: Text contents displayed in the date text block.

✧ Start date: Sets the start date of the date selection range.

✧ End date: Sets the end date of the date selection range.

✧ First day of the week: Sets which day of the week is the first day of the week in the date.

✧ Highlight today: Highlight today's date in the date selection frame.

✧ Date format: Sets date display format of the date: There are two formats: Short and Long.

✧ Font: Sets the font format of the text displayed on the date; for example font style, size and bold etc.

✧ Text color: Sets the color of the text of the date.

✧ Border color: Sets the color of the date's border.

✧ Enable: Sets the availability of the date.

✧ For other property settings please refer to the section "7.4 Graphic universal properties".

7.6.9 Calendar

➢ Open the window interface and click on “Toolbox” → “Window Controls” → “Calendar” in the tools window to the left; the toolbar is as shown in the figure above.

➢ Move the mouse to the working area of the window, select a position and press the mouse to generate a calendar.

➢ Just repeat the steps above if another calendar needs to be drawn.

Graphic introduction:

☞ The figure above is a selected calendar; click on the calendar to enter selected status.

Calendar properties

✧ Date selection: The selected year, month and day from the calendar displayed on the calendar with a base color.

✧ Start date: Sets the start date of the calendar.

✧ End date: Sets the end date of the calendar.

✧ First day of the week: Sets which day of the week is the first day of the week on the calendar.

✧ Font: Sets the font format of the text displayed on the calendar; for example font style, size and bold etc.

✧ Border color: Sets the color of the calendar’s border.

✧· Enable: Sets the availability of the calendar.

✧· For other property settings please refer to the section “7.4 Graphic universal properties”.

7.6.10 Image

➢· Open the window interface and click on “Toolbox” → “Window Controls” → “Image” in the tools window to the left; the toolbar is as shown in the figure above.

➢· Move the mouse to the working area of the window, select a starting point and press the left mouse button and drag the mouse towards the bottom-right; after releasing the left mouse button, the drawing of the image is complete, and the image will be generated.

➢· Just repeat the steps above if another image needs to be drawn.

Graphic introduction:

☞· The figure above is a selected image; click on the image to enter selected status.

☞· Points 1-8 in the figure are the tensile points to adjust the graphic size; 9 is the rotation point and 10 is the center.

☞· Above point 2 and below point 6 in the figure are the horizontal distortion points. The right of point 4 and the left of point 8 are the vertical distortion points.

Image properties

✧· TheImagePath:: Sets the source path of the image; press the“  ”button to select,Support for multilingual functionality.

✧· For other property settings please refer to the section “7.4 Graphic universal properties”.

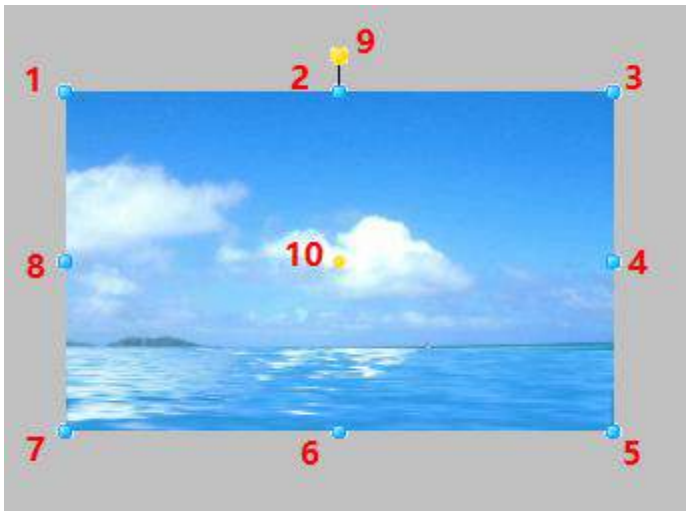
7.6.11 GifImage

➢· Open the window interface and click on “Toolbox” → “Window Controls” → “GifImage” in the tools window to the left; the toolbar is as shown in the figure above.

➢· Move the mouse to the working area of the window, select a starting point and press the left mouse button and drag the mouse towards the bottom-right; after releasing the left mouse button, the drawing of the gif image is complete, and the gif image will be generated.

➢· Just repeat the steps above if another gif image needs to be drawn.

Graphic introduction:



☞· The figure above is a selected image; click on the image to enter selected status.

☞· Points 1-8 in the figure are the tensile points to adjust the graphic size; 9 is the rotation point and 10 is the center.

☞ Above point 2 and below point 6 in the figure are the horizontal distortion points. The right of point 4 and the left of point 8 are the vertical distortion points.

Image properties

✧ TheImagePath: Sets the source path of the image; press the“ ”button to select,Support for multilingual functionality.

✧ Enable:Sets image availability.

✧ For other property settings please refer to the section “7.4 Graphic universal properties”.

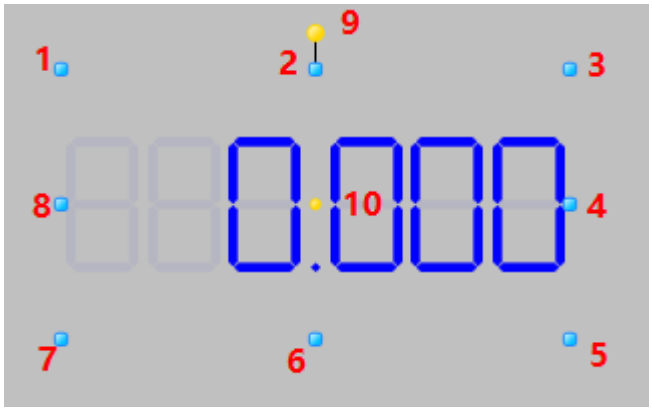
7.6.12 NixieTube

➢ Open the window interface and click on “Toolbox” → “Window Controls” → “NixieTube” in the tools window to the left; the toolbar is as shown in the figure above.

➢ Move the mouse to the working area of the window, select a starting point and press the left mouse button and drag the mouse towards the bottom-right; after releasing the left mouse button, the drawing of the nixie tube is complete, and the nixie tube will be generated.

➢ Just repeat the steps above if another nixie tube needs to be drawn.

Graphic introduction:



The figure above is a selected nixie tube; click on the nixie tube to enter selected status.The control point usage is the same with other controls.

Nixie tube properties

✧･ Digit: Sets the digit displayed of the nixie tube integer part.

✧･ FullDigit:Sets the total display digit of the NixieTube.

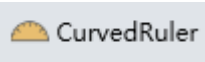
✧･ Color: Sets the color of the NixieTube.

✧･ Number: Set the value displayed of the NixieTube.

✧･ Enable:Sets the digital tube availability.

For other property settings please refer to the section “7.4 Graphic universal properties”.

7.6.13 CurvedRuler

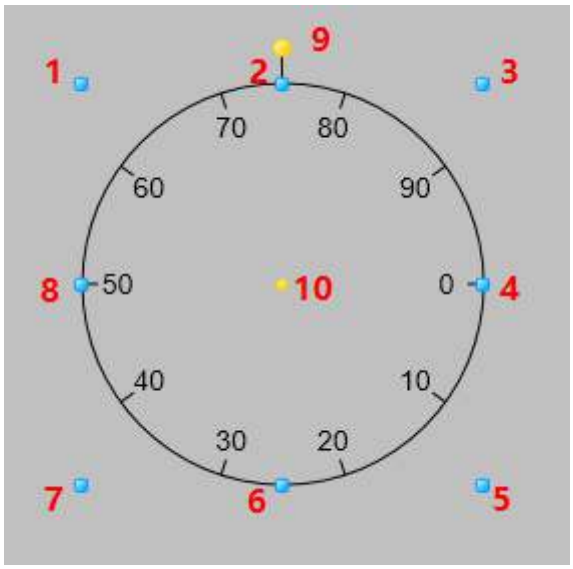


➢･ Open the window interface and click on “Toolbox” → “Window Controls” → “CurvedRuler” in the tools window to the left; the toolbar is as shown in the figure above.

➢･ Move the mouse to the working area of the window, select a starting point and press the left mouse button and drag the mouse towards the bottom-right; after releasing the left mouse button, the drawing of the CurvedRuler is complete, and the CurvedRuler will be generated.

➢･ Just repeat the steps above if another CurvedRuler needs to be drawn.

Graphic introduction:



☞ The figure above is a selected CurvedRuler; click on the CurvedRuler to enter selected status.

☞ Points 1-8 in the figure are the tensile points to adjust the graphic size; 9 is the rotation point and 10 is the center.

☞ Above point 2 and below point 6 in the figure are the horizontal distortion points. The right of point 4 and the left of point 8 are the vertical distortion points.

CurvedRuler properties

➢ StartAngle:Sets the starting angle of the CurvedRuler (unit: degrees).

➢ ScanAngle:Sets the scaning angle of the CurvedRuler (unit: degrees).

➢ StartValue:Sets the starting value of the CurvedRuler.

➢ EndValue:Sets the ending value of the CurvedRuler.

➢ BigTickNumber:Sets the big tick number of the CurvedRuler.

➢ SmallTickNumber:Sets the small tick number of the CurvedRuler.

➢ TickHeight:Sets the tick height of the CurvedRuler.

➢ TickPosition:Sets the tick position of the CurvedRuler.

➢ IsTextShow:Sets whether text is displayed of CurvedRuler.

➢ IsArcShow:Sets whether arc is displayed of CurvedRuler.

➢ LineBrush:Sets the current color of CurvedRuler.

➢ Font:Sets the font format of the text displayed on the CurvedRuler.

➢ TextMargin:Sets the distance between text and surrounding lines.

➢ For other property settings please refer to the section “7.4 Graphic universal properties”.

7.7 Properties and rendering of extended control

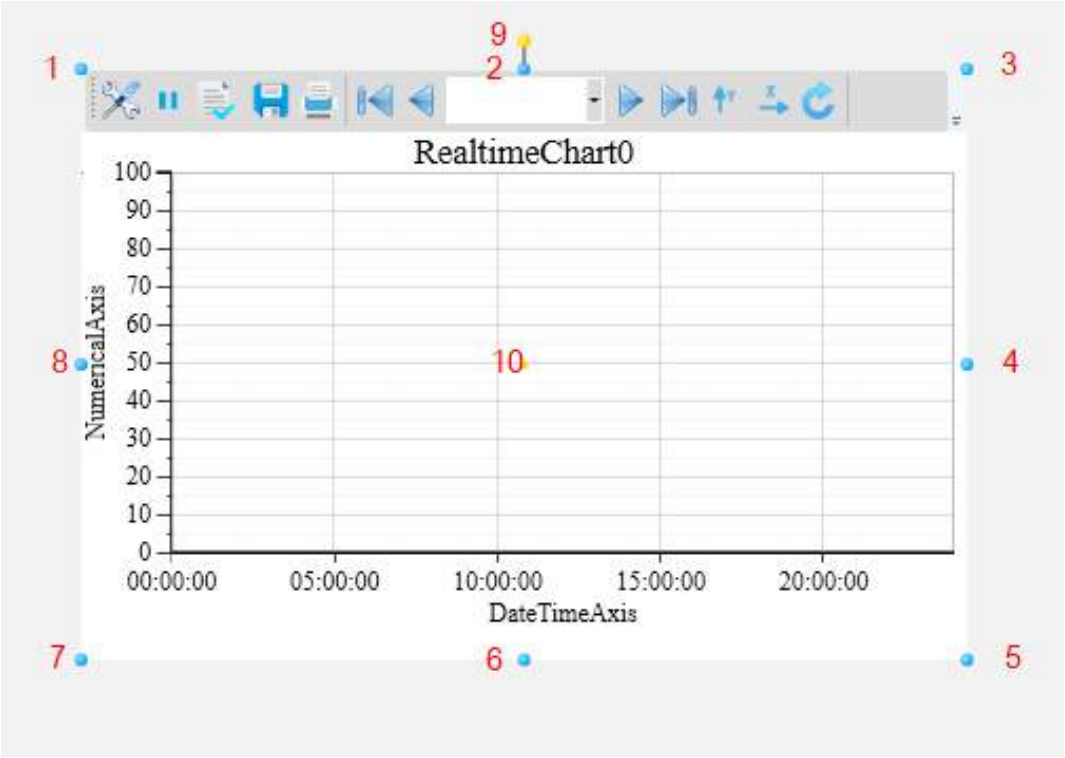
7.7.1 RealtimeChart

➢ Open the window interface and click on “Toolbox” → “Extended controls” → “Realtime chart” in the tools window to the left; the toolbar as shown in the figure above.

➢ Move the mouse to the working area of the window, select a starting point and press the left mouse button and drag the mouse towards the bottom-right; after releasing the left mouse button, the drawing of a real-time chart is complete, and a real-time chart will be generated.

➢ Just repeat the steps above if another real-time chart needs to be drawn.

Graphic introduction:



☞ The figure above is a selected real-time curve; click on the real-time curve to enter selected status.

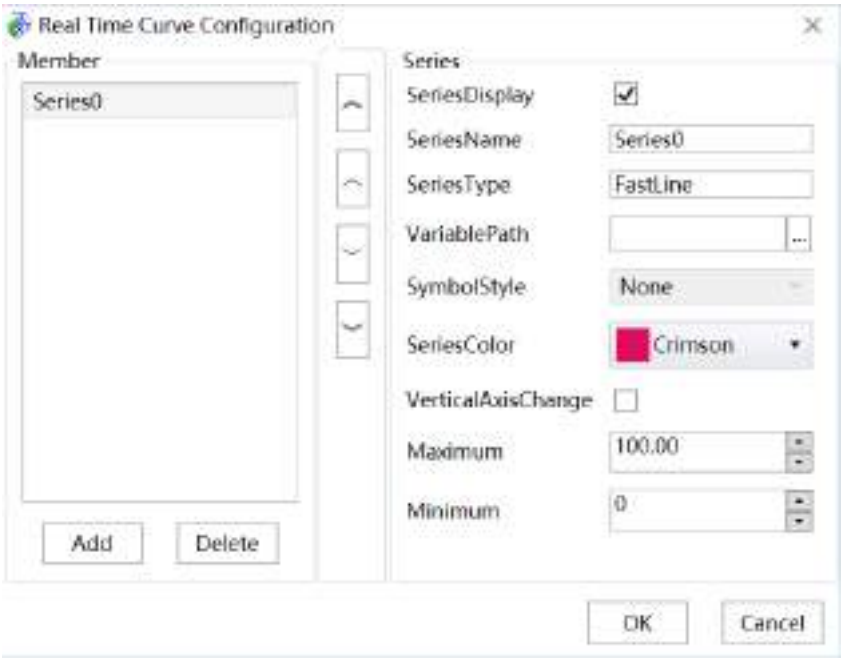
☞ There are two points on 1 in the figure above: the tensile point and the center; 2-8 are the tensile points and 9 is the rotation point,10 is the center.

☞ Above point 2 and below point 6 in the figure are the horizontal distortion points. The right of point 4 and the left of point 8 are the vertical distortion points.

☞ It is the toolbar on the top of the figure, which is mainly used to operate the chart.

Usage of the toolbar

✧ Set series:  configure the realtime chart in the runtime environment



✧· Stop: stop loading the real-time curve

✧· Display positioning line: whether to display the position line

✧· Save: current chart is saved as picture

✧· Print: print current chart

✧· First: move to the left most side

✧· Farword: move left

✧· Page range: choose the section of the current chart

✧· Backward: move right

✧· Last: move to the right most side

✧· Fixed YAxis: Scroll the mouse to scale the horizontal axis

✧･ Fixed XAxis:  Scroll the mouse to shrink the indulgence axis

✧･ Reset:  display current realtime chart

Real-time chart properties

Appearence

✧･ TitleFont: Set the font style for the chart.

✧･ FontColor: Set the font color for the chart.

✧･ DisplayLegend: Whether display legend of a chart.

✧･ MajorGrid: Set major grid.

✧･ MinorGrid: Set minor grid

✧･ Background: Set control background.

Limit Line

✧･ Thickness: Set the thickness of the limit line.

✧･ LimitLineMinimum: Set the maximum value of the limit line.

✧･ LimitLineMaximum: Set the minimum value of the limit line.

✧･ UpperAndLowerDisplay:Set whether to display the limit line.

✧･ UpperLimitLineColor:Set upper limit color.

✧･ LowerLimitLineColor:Set lower limit color.

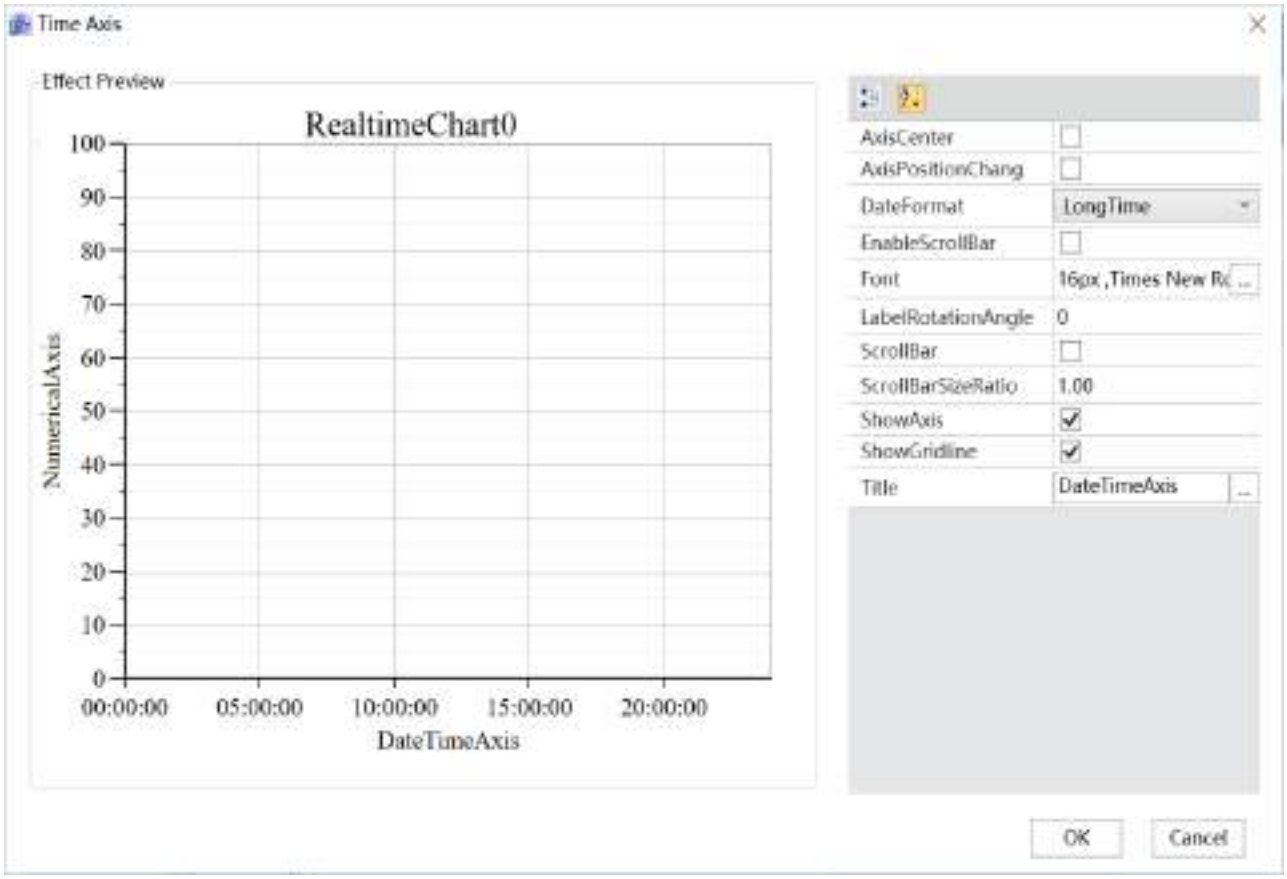
Series

✧･ PositioningLineColor:Set the color of the positioning line.

✧･ Title:Set the title of a chart,Support for multilingual functionality.

✧･ TimeAxis&NumericalAxis:Set time axis or numerical axis of a chart,take datetime axis as an

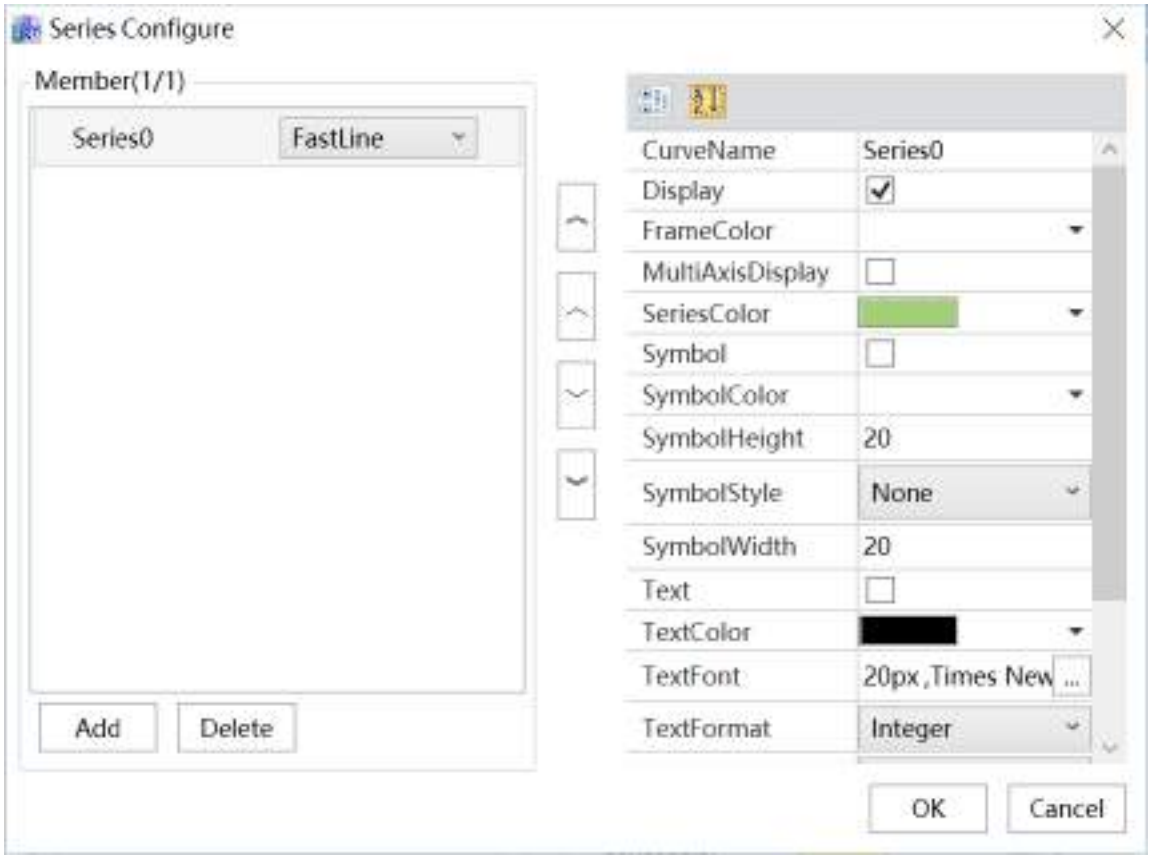
example,as shown below:



The left part is the effect preview，the right part is the datetime axis properties Please refer to the table below:

Property	Description
AxisCenter	Set whether the time axis is centered
AxisPositionChang	Set whether to transform the axis of the numeric axis
DateFormat	Display date format
Font	Set the title font
FontColor	Set the color of the headline
LabelRotationAngle	Set rotation angle of time axis
ScrollBar	Whether to display scroll bar
ScrollBarSizeRatio	Set the ratio of scroll bar
ShowGridline	Set whether to display the gridline
ShowAxis	Set whether to display the axis
Title	Set the title of time axis,Support for multilingual functionality

Series Collection



Property	Description
CurveName	Set the name of the curve
Display	Set whether to display the curve
FrameColor	Set the color of the curve frame
MultiAxisDisplay	Set whether to display the multi axis
SeriesColor	Set the color of the series
Symbol	Set whether to display the symbol
SymbolColor	Set the color of symbol
SymbolHeight	Set the height of symbol
SymbolStyle	Set the style of the dat symbol
SymbolWidth	Set the width of symbol
Text	Set whether to display the text
TextColor	Set text color
TextFont	Set text font
TextFormat	Set text color
TextType	Set text type
VariablePath	Set variable path

VerticalAxis	Sets the parameters of the vertical axis displayed by multiple axes
Width	Set the thickness of series

RealtimeChart properties

Limit Line

Thickness:Set the thickness of the limit line.

LimitLineMaximum:Set the maximum value of the limit line.

LimitLineMinimum:Set the minimum value of the limit line.

UpperAndLowerDisplay:Set whether to display the limit line.

UpperLimitLineColor:Set upper limit color.

LowerLimitLineColor:Set lower limit color.

Toolbar

Toolbar:Set Whether to display toolbar.

Toolbaricon:Set the toolbar icon type.

Statusbar

StatusDisplay:Set whether the status bar is displayed

NumericalFormat:Set the format of the status bar data display.

Font:Set the font style for the status bar.

BackgroundColor:Set the background color of the status bar.

FontColor:Set the font color of the column name in the status bar.

StatusGrid:Set whether the status bar displays grid lines.

For other property settings please refer to the section “7.4 Graphic universal properties”.

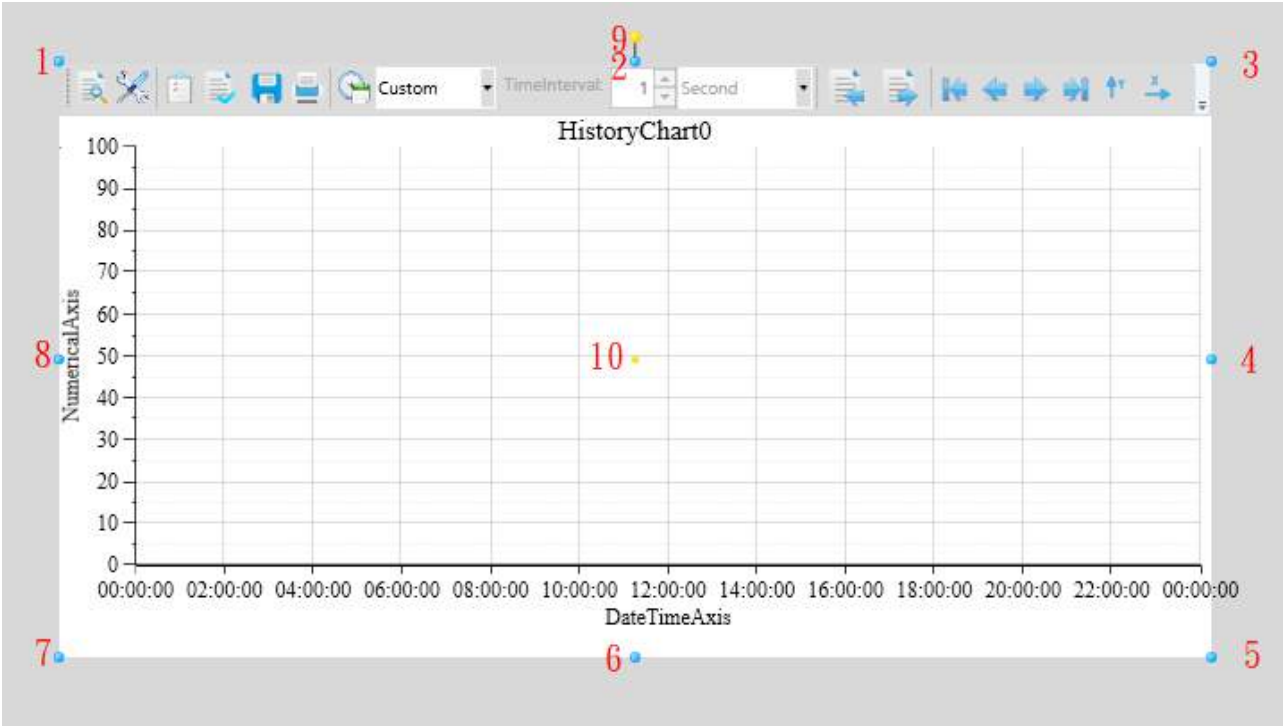
7.7.2 HistoryChart

➢ Open the window interface and click on “Toolbox” → “Extended controls” → “History chart” in the tools window to the left; the toolbar is as shown in the figure above.

➢ Move the mouse to the working area of the window, select a starting point and press the left mouse button and drag the mouse towards the bottom-right; after releasing the left mouse button, the drawing of a history chart is complete, and a history chart will be generated.

➢ Just repeat the steps above if another history chart needs to be drawn.

Graphic introduction:



☞ The figure above is a selected history chart; click on the history chart to enter selected status.

☞ There are two points on 1 in the figure above: the tensile point and the center; 2-8 are the tensile points and 9 is the rotation point,10 is the center.

☞ Above point 2 and below point 6 in the figure are the horizontal distortion points. The right of point 4 and the left of point 8 are the vertical distortion points.

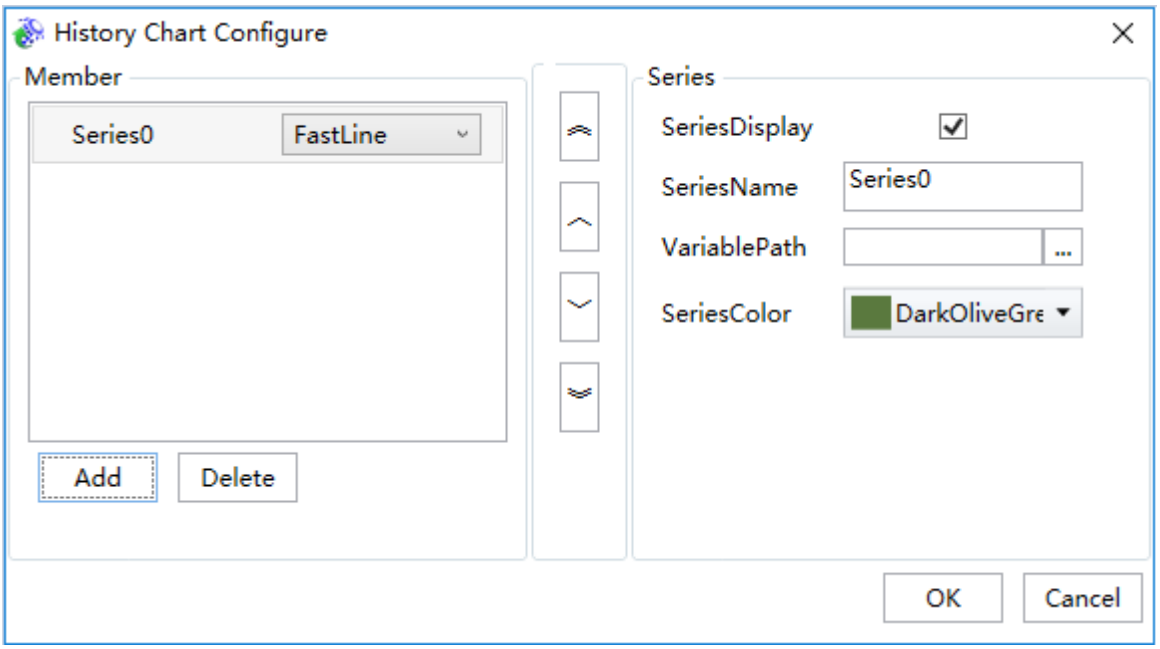
☞ It is the statistics table on the bottom of the figure which is mainly used to get the min , max, avg, sum and count of each curve.

☞ It is the toolbar on the top of figure which is used to operate chart.

Usage of the toolbar

✧ Query:  query data according to the current configuration

✧ Set series:  configure the curve style and associated variable in the history chart



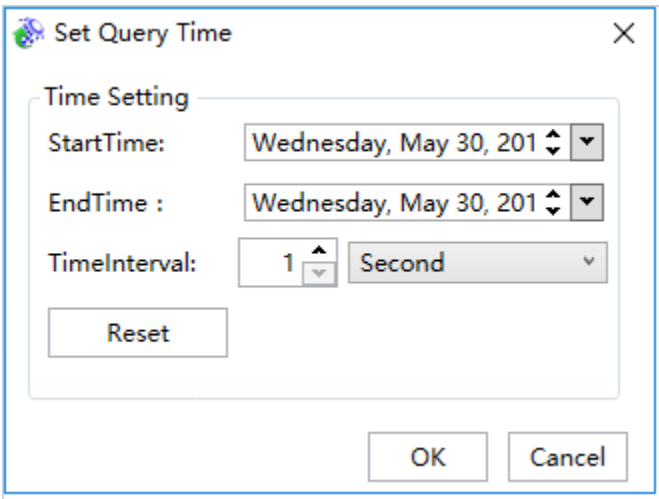
✧ Status column display:  Set whether to display the column in the status column

✧ Position line:  whether to display the position line

✧ Save:  history chart is saved as picture

✧･ Print:  print current history chart

✧･ Set query time:  when "set recent query time" is "custom"，user can set self-defined query range



✧･ Set recent query time: 

Set the query time range: recent one hour、recent one day、recent one week、recent one month、recent three months、recent six months、recent one year、recent three year、custom

✧･ Time interval: 

when "set recent query time" is not "custom"，user can input time interval，its unit can be: second、minute、hour、day、month、year、Default（preset time interval）

✧･ Import:  Impoirt Excel to selected curve

✧･ Export:  Export curve to Excel

✧･ First:  set the scroll to the start time to see the chart

✧･ Forward:  move forward to see previous time quantum chart

✧･ Backword:  move backword to see

✧･ Last:  set the scroll to the end time to see the chart

✧･ Fixed YAxis:  Scroll the mouse to scale the horizontal axis

✧･ Fixed XAxis:  Scroll the mouse to shrink the indulgence axis

HistoryChart properties

Appearence

✧･ StatusGrid:Whether display toolbar of a chart.

✧･ StatusDisplay:Whether display the status bar.

✧･ NumericalFormat:Set series numerical format.

✧･ StatusbarBackgroundColor:Set status bar background color.

✧･ StatusbarFontColor:Set status bar font color.

✧･ TitleFont:Set the font style for the chart.

✧･ FontColor:Set the font color for the chart.

✧･ DisplayLegend: Whether display legend of a chart.

✧･ MajorGrid: Set major grid.

✧･ MinorGrid: Set minor grid.

✧･ Background: Set control background.

Limit Line

✧･ Thickness:Set the thickness of the limit line.

✧･ LimitLineMaximum:Set the maximum value of the limit line.

✧･ LimitLineMinimum:Set the minimum value of the limit line.

✧･ UpperAndLowerDisplay:Set whether to display the limit line.

✧･ UpperLimitLineColor:Set upper limit color.

✧･ LowerLimitLineColor:Set lower limit color.

Series

✧･ DateTimeAxis:Set chart date axis.

✧･ NumericalAxis:Set chart numerical axis.

✧･ SeriesCollection:Set series of the category list.

✧･ PositioningLine:Set positioning line for the curve chart.

✧･ PositioningLineColor:Set the color of positioning line.

✧･ VerticalAxisAutoChange:Set whether to automatically adjust the maximum value of the vertical axis.

✧･ Title:Set title of a chart,Support for multilingual functionality.

Statusbar

✧･ StatusDisplay:Set whether the status bar is displayed

✧･ NumericalFormat:Set the format of the status bar data display.

✧･ Font:Set the font style for the status bar.

✧･ BackgroundColor:Set the background color of the status bar.

✧･ FontColor:Set the font color of the column name in the status bar.

✧･ StatusGrid:Set whether the status bar displays grid lines.

✧ Numeral axis

Property	Description
AxisCenter	Set whether the time axis is centered

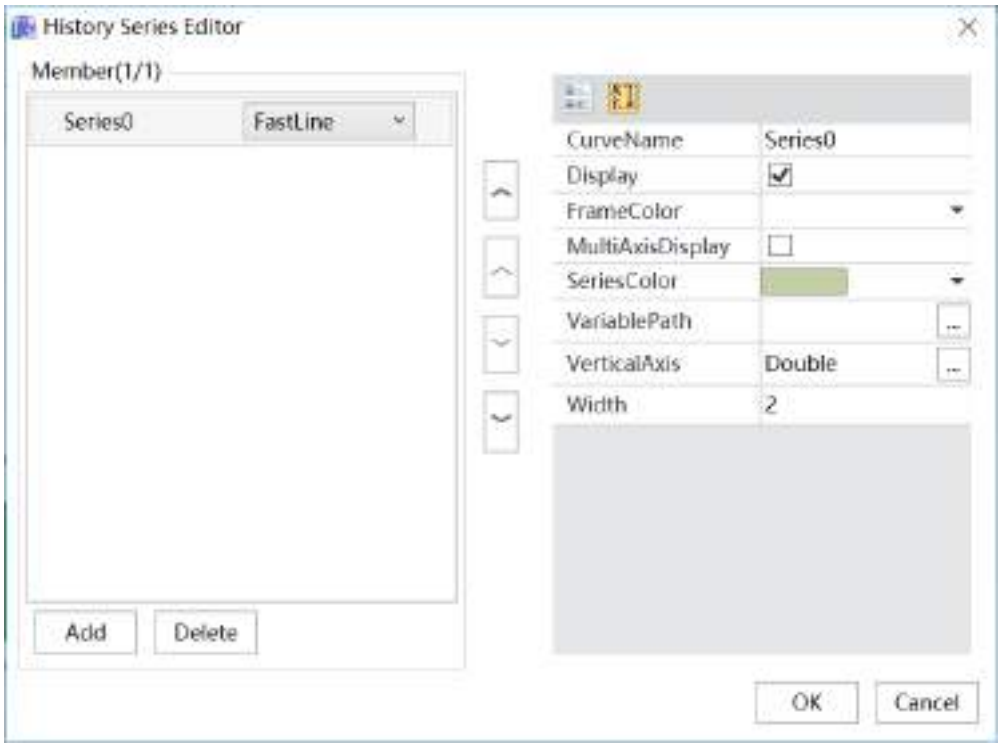
BigInterval	Set big interval for the numeral axis
DateFormat	Display date format
Font	Set the title font
FontColor	Set the color of the headline
LabelRotationAngle	Set rotation angle of time axis
LittleInterval	Set little interval for the numeral axis
Maximum	Set maximum for the numeral axis
Minimum	Set minimum for the numeral axis
NumericalFormat	Set numerical format
ScrollBar	Whether to display scroll bar
ScrollBarSizeRatio	Set the ratio of scroll bar
ShowGridline	Set whether to display the gridline
ShowAxis	Set whether to display the axis
Title	Set the title of time axis,Support for multilingual functionality.

✧ Datetime axis

Property	Description
AxisCenter	Set whether the time axis is centered
DateFormat	Display date format
Font	Set the title font
FontColor	Set the color of the headline
LabelRotationAngle	Set rotation angle of time axis
ScrollBar	Whether to display scroll bar
ScrollBarSizeRatio	Set the ratio of scroll bar
ShowGridline	Set whether to display the gridline
ShowAxis	Set whether to display the axis
Title	Set the title of time axis,Support for multilingual functionality.

✧· SeriesCollection: Configures the curve style and connected variables in the figure，for the configuration method as follows:

Click“  ”bitton，pop up "History Series Editor" window，add one series as follows:



The left part is add or delete members, each member is a series; the middle part can rank this series, the right part is the selected series properties, as shown in the table below:

Property	Description
CurveName	Set curve name
Display	Set whether to display the curve
FrameColor	Set the frame color
MultiAxisDisplay	Set whether to display the multi axis
SeriesColor	Set the series color
VariablePath	Set variable path
VerticalAxis	Sets the parameters of the vertical axis displayed by multiple axes
Width	Set the thickness of series

✧･ For other property settings please refer to the section “7.4 Graphic universal properties”.

7.7.3 XYChart

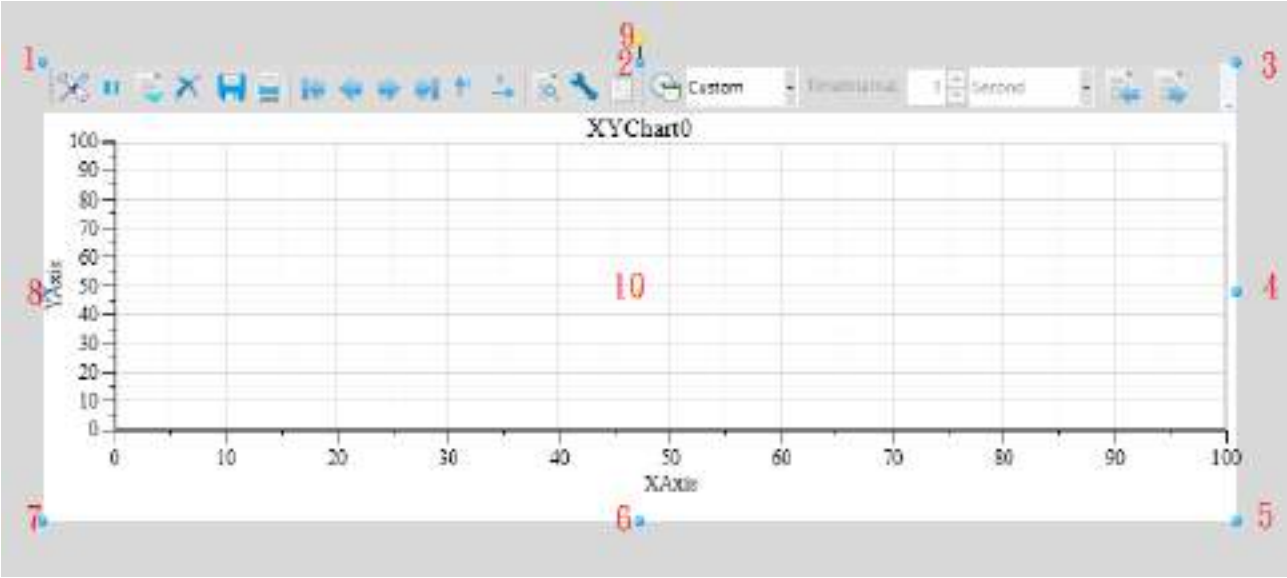
➢･ Open the window interface and click on “Toolbox” → “Extended controls” → “XY chart” in the tools

window to the left; the toolbar is as shown in the figure above.

➢· Move the mouse to the working area of the window, select a starting point and press the left mouse button and drag the mouse towards the bottom-right; after releasing the left mouse button, the drawing of a XY chart is complete, and a XY chart will be generated.

➢· Just repeat the steps above if another XY chart needs to be drawn.

Graphic introduction:



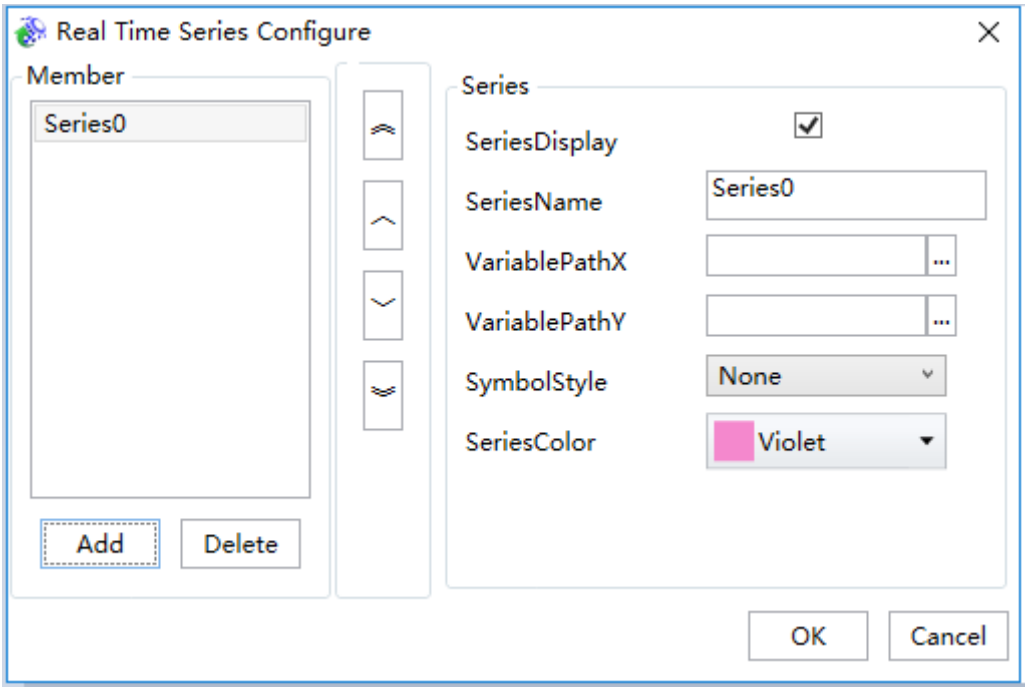
☞· The figure above is a selected XY curve; click on the XY curve to enter selected status.

☞· There are two points on 1 in the figure above: the tensile point and the center; 2-8 are the tensile points and 9 is the rotation point,10 is the center.

☞· Above point 2 and below point 6 in the figure are the horizontal distortion points. The right of point 4 and the left of point 8 are the vertical distortion points.

Usage of the tool bar

✧· Configuration:  Configure the realtime chart



✧･ Position line: Whether to display the positIon line

✧･ Start: Whether to start loading realtime chart

✧･ Clear: Clear the realtime chart

✧･ Save: Save the chart

✧･ Print: Print the chart

✧･ First: XY chart first page data(When XY data is large, it will be paged automatically)

✧･ Last: XY chart the last page data(When XY data is large, it will be paged automatically)

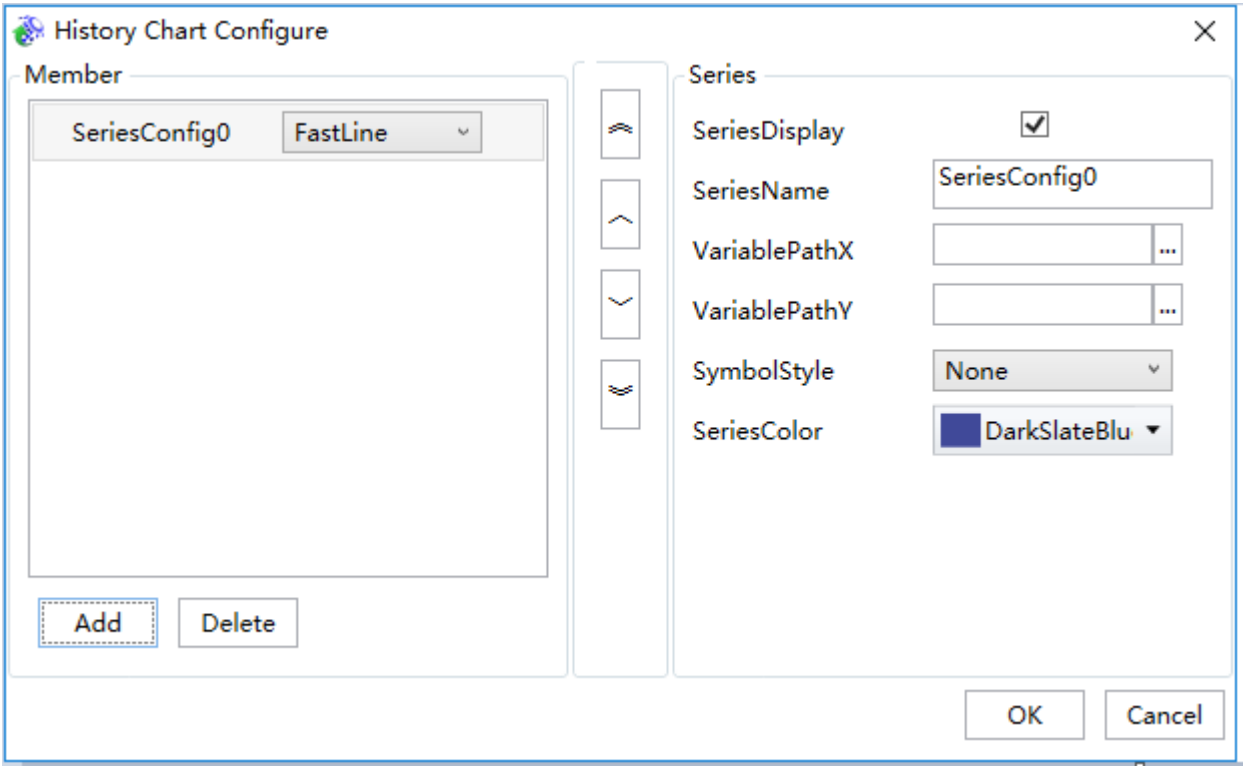
✧･ Forward: Page up

✧･ Backward:  Page down

✧･ Fixed YAxis:  Scroll the mouse to scale the horizontal axis

✧･ Fixed XAxis:  Scroll the mouse to shrink the indulgence axis

✧･ Configuration:  Configure the history chart



✧･ Query:  Query history data

✧･ Query time setting:  Set the start time and end time of history chart

✧･ Set nearest query time:  Set the nearest query time period,respectively are:Last hour, last day, last week, last month, last three months, last six months, last year, last three years, custom

✧･ TimeInterval: 

When "Set nearest query time" is the most recent time outside of "Custom",user can input time interval,the unit are:second,minute,hour,day,month,year,default(The default is the interval set at development time)

✧･ Display setting:  Set the display of the status bar

✧･ Import:  Import Excel to the chart

✧･ Export:  Export the chart to the Excel

XY chart properties

✧･ Title: Text contents in the title list on top of the chart; it represents the displayed name of the chart, and it can be left empty,Support for multilingual functionality.

✧･ TitleFont: Sets the font format of the title in the XY chart; for example font style, size and bold etc

✧･ StatusDisplay: Set whether to display the status bar

✧･ StatusGrid:Whether display status grid of a chart

✧･ DisplayLegend: Set whether to display the legend

✧･ RefreshCount: Set the count per refreshment

✧･ RefrshTime: Set the refresh time

✧･ Toolbar: Set whether to display the toolbar

✧･ Background: Set the background of the chart

✧･ Y Axis adjustment automatically: set whether to adjust axis minmax value in the runtime environment

✧･ Position line: whether to display position line

✧･ Position line color: set the color of position line

✧･ Limit line: set whether to display the limit line

✧· UpperLimitLineColor: Set the color of upper limit line

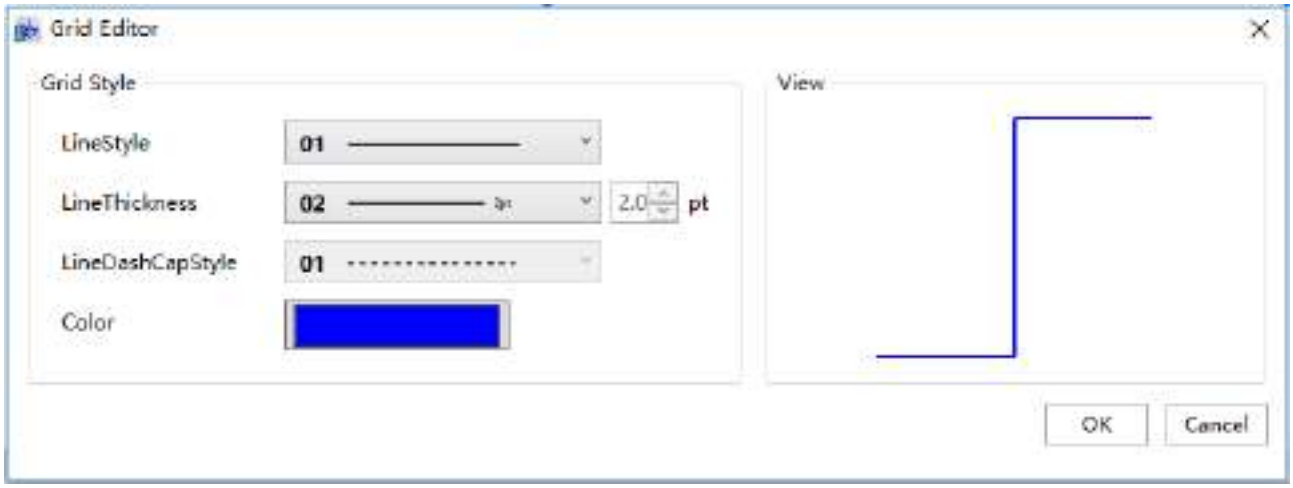
✧· LowerLimitLineColor: Set the color of lower limit line

✧· LimitLineMaximumValue: Set limit line maximum value

✧· LimitLineMinimumValue: Set limit line minimum value

✧· Limit line thickness: Set the thickness of limit line

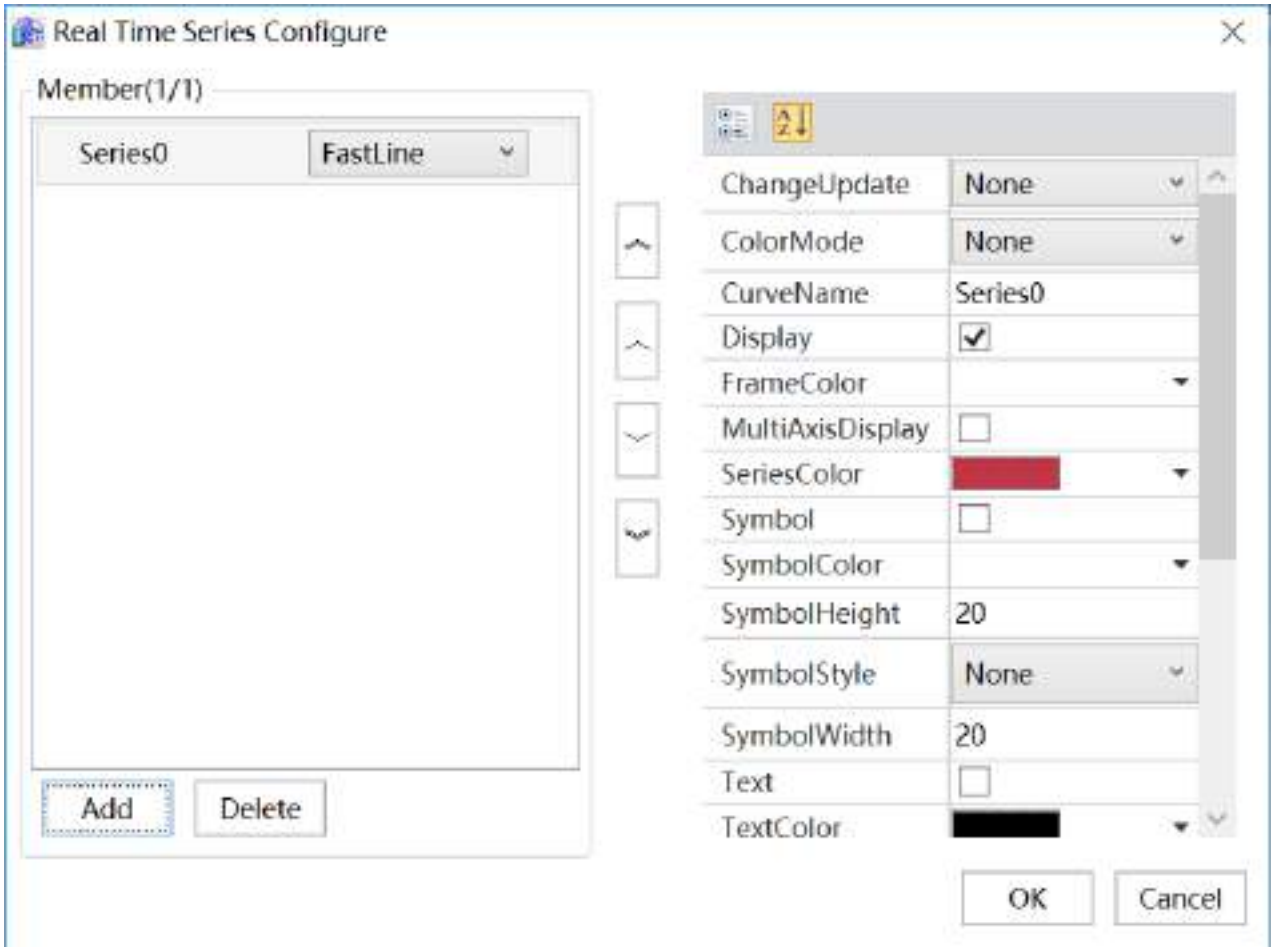
✧· MajorGrid:Set major grid



✧· MinorGrid:Set minor grid

✧· RealTimeSeries: Configure real time series，the configuration method is as follows:

Click“  ”bitton，pop up real time series configure window，add one series as follows:



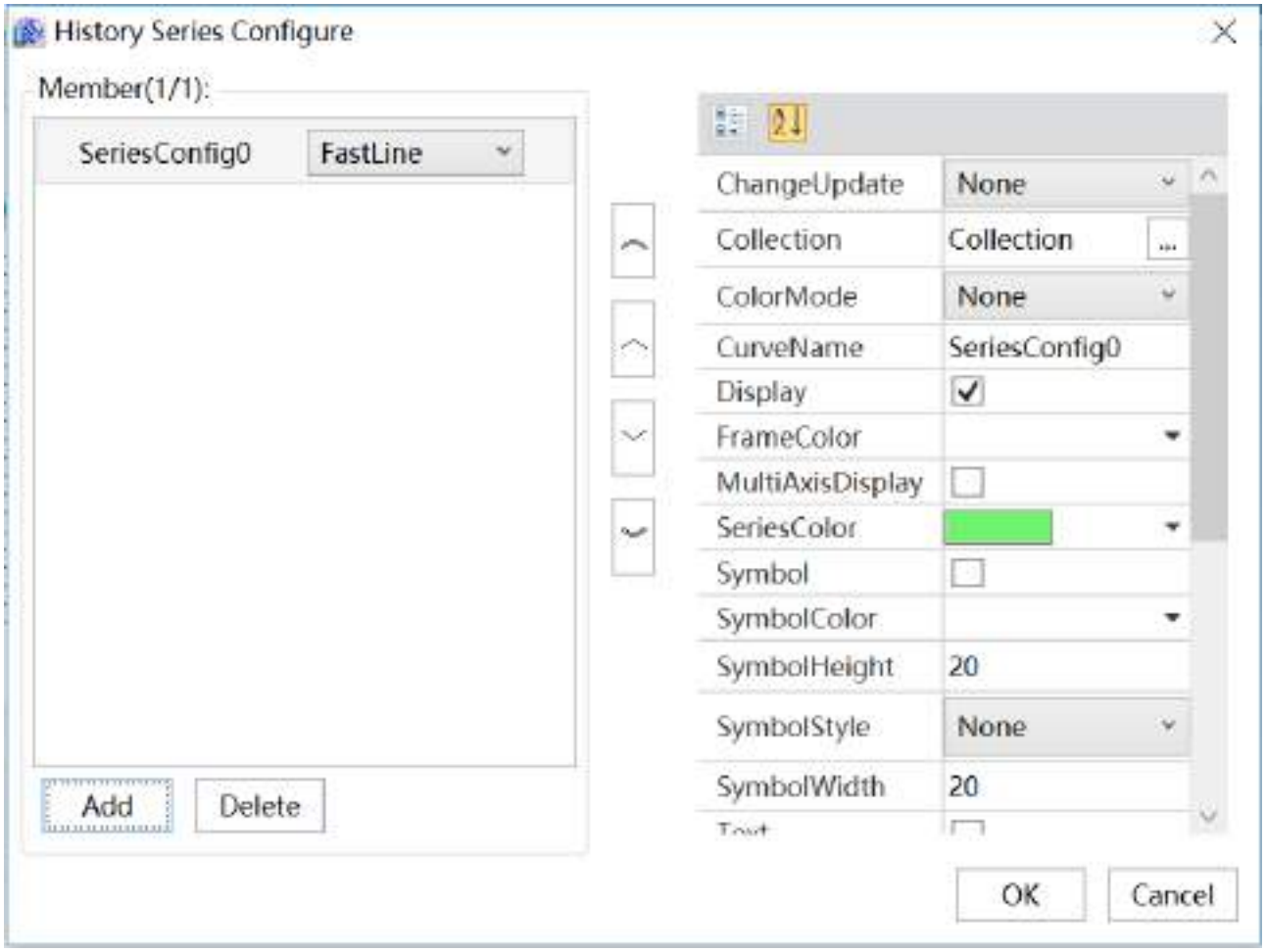
The left part is add or delete members，each member is a series；the middle part can rank this series，the right part is the selected series properties，as shown in the table below:

Properties	Description
ChangeUpdate	Set updating data when values change
ColorMode	Set color mode
CurveName	Set the name of the curve
Display	Set whether to display the curve
FrameColor	Set the color of the curve frame(Pillar style)
MultiAxisDisplay	Set whether to display the multi axis
SeriesColor	Set the color of the series
Symbol	Set whether to display the symbol
SymbolColor	Set the color of symbol
SymbolHeight	Set the height of symbol
SymbolStyle	Set the style of the dat symbol
SymbolWidth	Set the width of symbol

Text	Set whether to display the text
TextColor	Set text color
TextFont	Set text font
TextFormat	Set text format
TextType	Set text type
UpdateValue	Set value changes to update data
VariablePathX	Set variable path of X-axis
VariablePathY	Set variable path of Y-axis
VerticalAxis	Sets the parameters of the vertical axis displayed by multiple axes
Width	Set the thickness of series

✧· HistorySeries: Configuration: Configure the history chart in the diagtam，the configuration method is as follows:

Click“  ”button，pup up the XY chart configuration window，the default is none，add a series as follows:



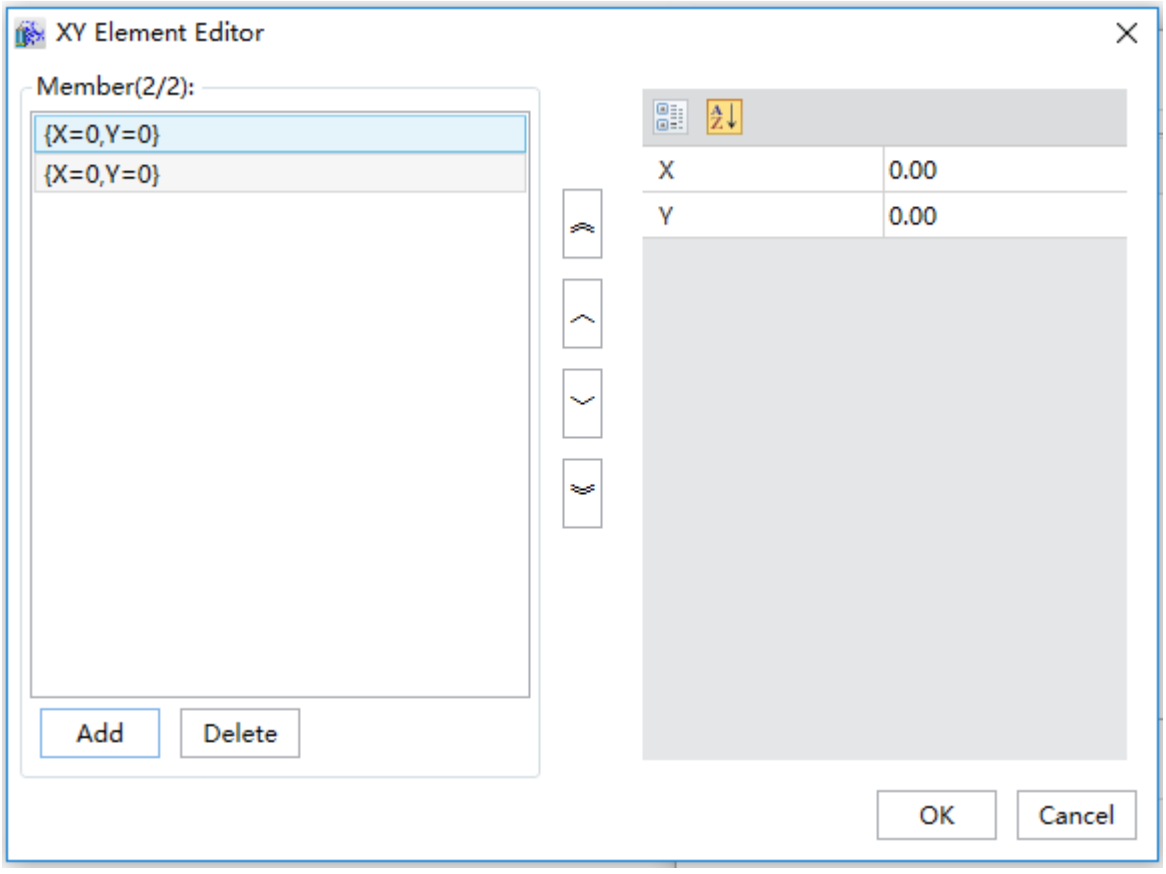
The left part is add or delete members，each member is a series；the middle part can rank this series，the right part is the selected series properties，as shown in the table below:

Properties	Description
ChangeUpdate	Set updating data when values change
Collection	Set the numerical point coordinates of the curve members
ColorMode	Set the color mode
CurveName	Set curve name
Display	Set whether to display the curve
FrameColor	Set the frame color
MultiAxisDisplay	Set whether to display the multi axis
SeriesColor	Set the series color
Symbol	Set whether to display the symbol
SymbolColor	Set the color of symbol
SymbolHeight	Set the height of symbol
SymbolStyle	Set the style of the dat symbol

SymbolWidth	Set the width of symbol
Text	Set whether to display the text
TextColor	Set text color
TextFont	Set text font
TextFormat	Set text format
TextType	Set text type
UpdateValue	Set value changes to update data
VariablePathX	Set variable path of X-axis
VariablePathY	Set variable path of Y-axis
VerticalAxis	Sets the parameters of the vertical axis displayed by multiple axes
Width	Set the thickness of series

✧· Collection: Configure the fixed display series in the diagram as follows:

Click“  ”button，pup up the XY chart configuration window，the default is none，add a series as follows:



✧· X axis

Properties	Description
Font	Set the numerical font style
Scroll bar	Set whether to display the scroll bar
Resizable scroll bar	Set whether to resize the scroll bar
Scroll custom bar	Set whether to use the scroll custom bar
Font color	Set the color of the value
Title	Set the title of the axis,Support for multilingual functionality.
Large scale	Set the large scale of the axis
Display grid	Set whether to display the grid
Display axis	Set whether to display the axis
Maximum value	Set the maximum value
Minimum value	Set the minimum value
Numerical format	Set the numerical format
Zone style	Set the zone style
Small scale	Set the small scale between two large scales
Scroll bar size ratio	Set the size ratio of scroll bar

✧▪ Y axis

Properties	Description
Font	Set the numerical font style
Scroll bar	Set whether to display the scoll bar
Resizable scroll bar	Set whether to resize the scroll bar
Scroll custom bar	Set whether to use the scroll custom bar
Font color	Set the color of the value
Title	Set the title of the axis,Support for multilingual functionality.
Large scale	Set the large scale of the axis
Display grid	Set whether to display the grid
Display axis	Set whether to display the axis
Maximum value	Set the maximum value
Minimum value	Set the minimum value
Numerical format	Set the numerical format
Zone style	Set the zone style
Small scale	Set the small scale between two large

	scales
Scroll bar size ratio	Set the size ratio of scroll bar

Statusbar

✧· StatusDisplay:Set whether the status bar is displayed

✧· NumericalFormat:Set the format of the status bar data display.

✧· Font:Set the font style for the status bar.

✧· BackgroundColor:Set the background color of the status bar.

✧· FontColor:Set the font color of the column name in the status bar.

✧· StatusGrid:Set whether the status bar displays grid lines.

✧· For other property settings please refer to the section “7.4 Graphic universal properties”.

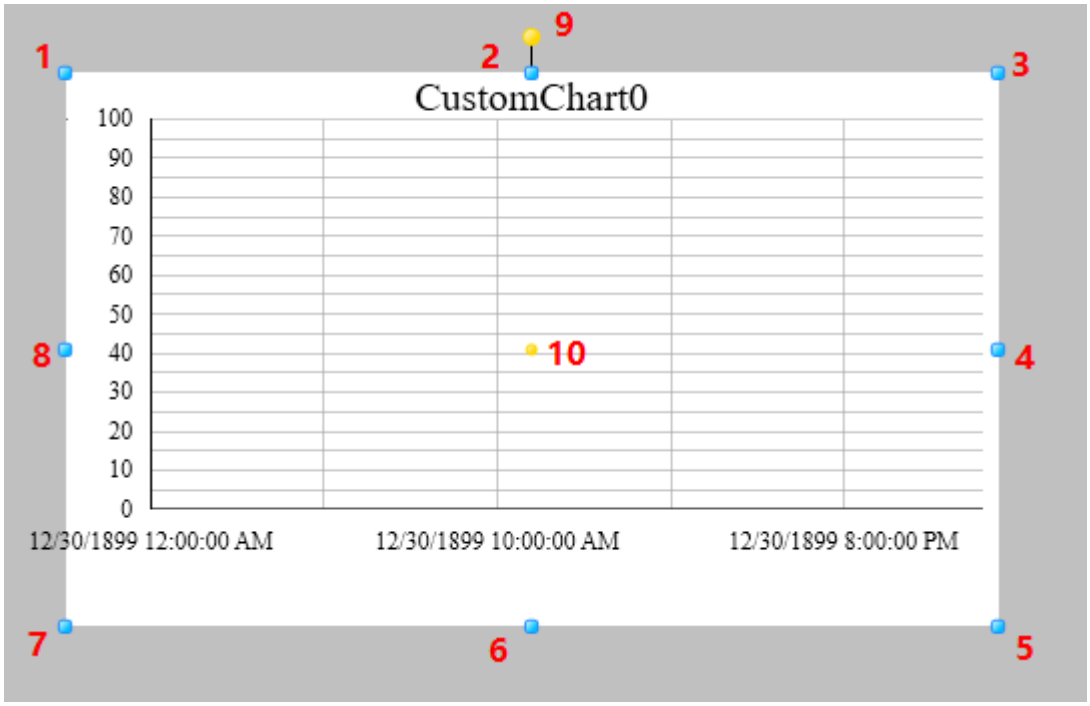
7.7.4 CustomChart

➢· Open the window interface and click on “Toolbox” → “Extended controls” → “Custom chart” in the tools window to the left; the toolbar is as shown in the figure above.

➢· Move the mouse to the working area of the window, select a starting point and press the left mouse button and drag the mouse towards the bottom-right; after releasing the left mouse button, the drawing of a custom chart is complete, and a custom chart will be generated.

➢· Just repeat the steps above if another custom chart needs to be drawn.

Graphic introduction:



☞ The figure above is a selected custom chart; click on the custom chart to enter selected status.

☞ There are two points on 1 in the figure above: the tensile point and the center; 2-8 are the tensile points and 9 is the rotation point,10 is the center.

☞ Above point 2 and below point 6 in the figure are the horizontal distortion points. The right of point 4 and the left of point 8 are the vertical distortion points.

Custom chart properties

✧ Title: Text contents in the title list on top of the chart; it represents the displayed name of the chart, and it can be left empty

✧ ChartType: Set the type of chart

✧ Position line: whether to display position line

✧ Position line color: set the color of position line

✧ AxisTranspose:Horizontal axis and vertical axis conversion position

✧ UpperAndLowerDisplay:Set whether to display the limit line

✧ LimitLineMaximum: Set the maximum value of the limit line

✧･ LimitLineMinimum: Set the minimum value of the limit line

✧･ UpperLimit line color: Set the upper limit line color

✧･ LowerLimit line color: Set the lower limit line color

✧･ Limit line thickness: set the thickness of limit line

Statusbar

✧･ StatusDisplay:Set whether the status bar is displayed

✧･ NumericalFormat:Set the format of the status bar data display.

✧･ Font:Set the font style for the status bar.

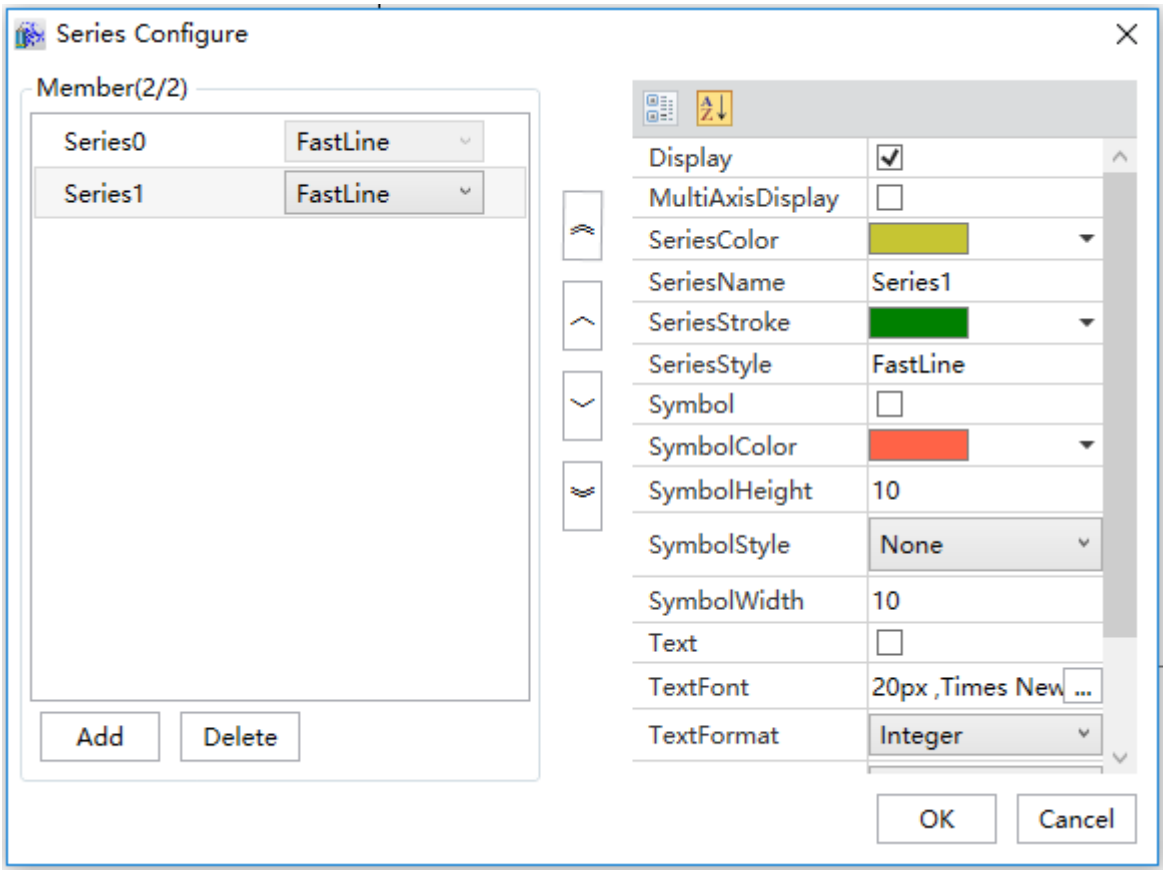
✧･ BackgroundColor:Set the background color of the status bar.

✧･ FontColor:Set the font color of the column name in the status bar.

✧･ StatusGrid:Set whether the status bar displays grid lines.

✧･ SeriesCollection: Configures the curve style and connected variables in the figure; the configuration method is as follows:

Click the “  ” button and the custom curve editor window will appear; add a curve, as shown in the figure below :

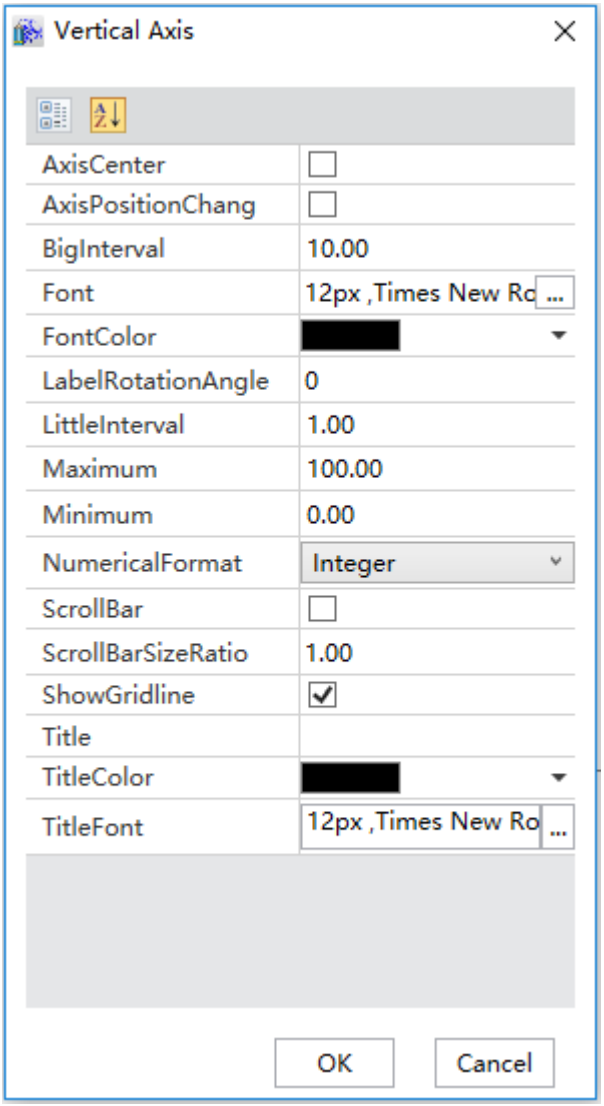


The left part is add or delete members,each member is a series;the middle part can rank this series,the right part is the selected series properties,as shown in the table below:

Properties	Description
Display	Set whether to display the series
MultiAxisDisplay	Set whether to display multi axis
SeriesStroke	Set series stroke
SeriesStyle	Set series style
SeriesName	Set the name of the series
Width	Set the thickness of series
SeriesColor	Set the color of series color
Symbol	Set whether to display symbol
SymbolColor	Set the color of symbol
SymbolStyle	Set the style of the dat symbol
SymbolWidth	Set the width of symbol
SymbolHeight	Set the height of symbol
Text	Set whether to display the text
TextFont	Set text font

TextFormat	Set text format
TextType	Set text type
VerticalAxis	Set vertical axis

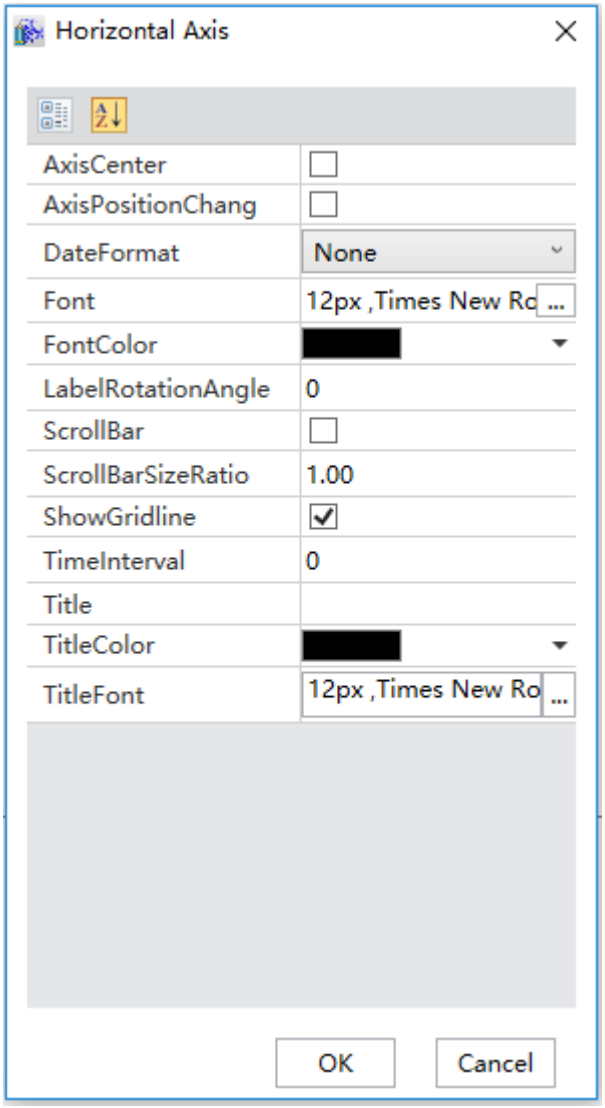
✧· Vertical axis



Properties	Description
AxisCenter	Set whether the axis is centered
AxisPositionChange	Set whether to change axis position
BigInterval	Set big interval
Font	Set the numerical font style
FontColor	Set the color of the value
LabelRotationAngle	Set the rotation Angle of the numerical axis
LittleInterval	Set little interval
ShowGridline	Set whether to display the grid line

Title	Set the title of the axis,Support for multilingual functionality.
TitleColor	Set title color
TitleFont	Set title font
Maximum	Set the maximum value
Minimum	Set the minimum value
NumericalFormat	Set the numrical format
ScrollBar	Set whether to display the scoll bar
ScrollBarSizeRatio	Set the size ratio of scroll bar

✧･ Horizontal axis

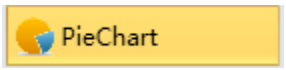


Properties	Description
AxisCenter	Set whether the axis is centered
AxisPositionChange	Set whether to change axis position
DateFormat	Set date format

Font	Set the numerical font style
FontColor	Set the color of the value
LabelRotationAngle	Set the rotation Angle of the numerical axis
ShowGridline	Set whether to display the grid line
Title	Set the title of the axis,Support for multilingual functionality.
TitleColor	Set title color
TitleFont	Set title font
NumericalFormat	Set the numrical format
ScrollBar	Set whether to display the scoll bar
ScrollBarSizeRatio	Set the size ratio of scroll bar

✧· For other property settings please refer to the section “7.4 Graphic universal properties”.

7.7.5 PieChart

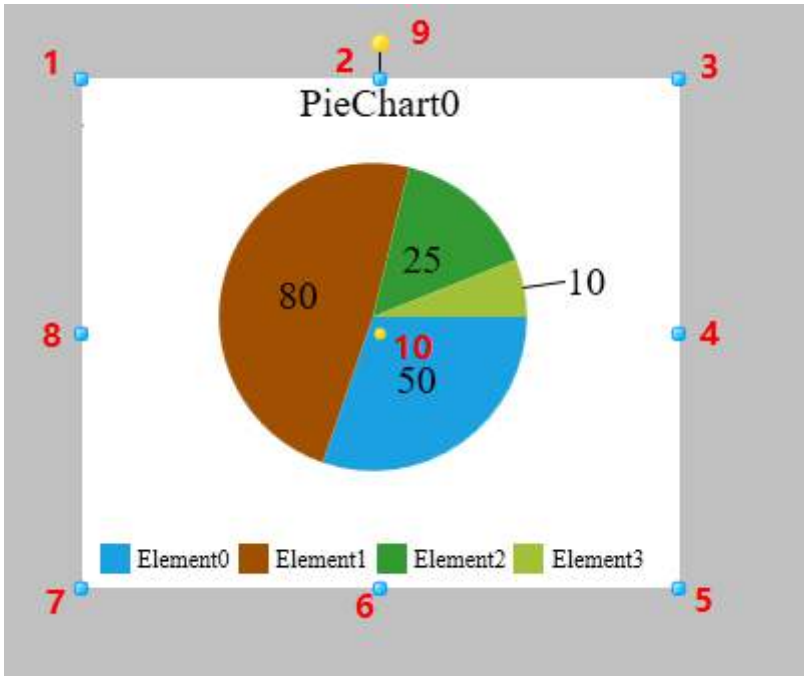


➢· Open the window interface and click on “Toolbox” → “Extended controls” → “Pie chart” in the tools window to the left; the toolbar is as shown in the figure above.

➢· Move the mouse to the working area of the window, select a starting point and press the left mouse button and drag the mouse towards the bottom-right; after releasing the left mouse button, the drawing of the pie chart is complete, and the pie chart will be generated.

➢· Just repeat the steps above if another pie chart needs to be drawn.

Graphic introduction:



☞ The figure above is a selected pie chart; click on the pie chart to enter selected status.

☞ There are two points on 1 in the figure above: the tensile point and the center; 2-8 are the tensile points and 9 is the rotation point,10 is the center.

☞ Above point 2 and below point 6 in the figure are the horizontal distortion points. The right of point 4 and the left of point 8 are the vertical distortion points.

Pie chart properties

✧ Title: Set the title of the chart,Support for multilingual functionality.

✧ TitleFont: Set the title font of the chart

✧ TitleColor: Set the title color of the chart

✧ LabelFont: Set label font

✧ LabelColor: Set label color

✧ LabelPosition: Set label position

✧ LegendFont: Set legend font

✧ LegendColor: Set legend color

✧ LegendPosition: Set legend position

✧･ RefreshTime:Set the refresh time

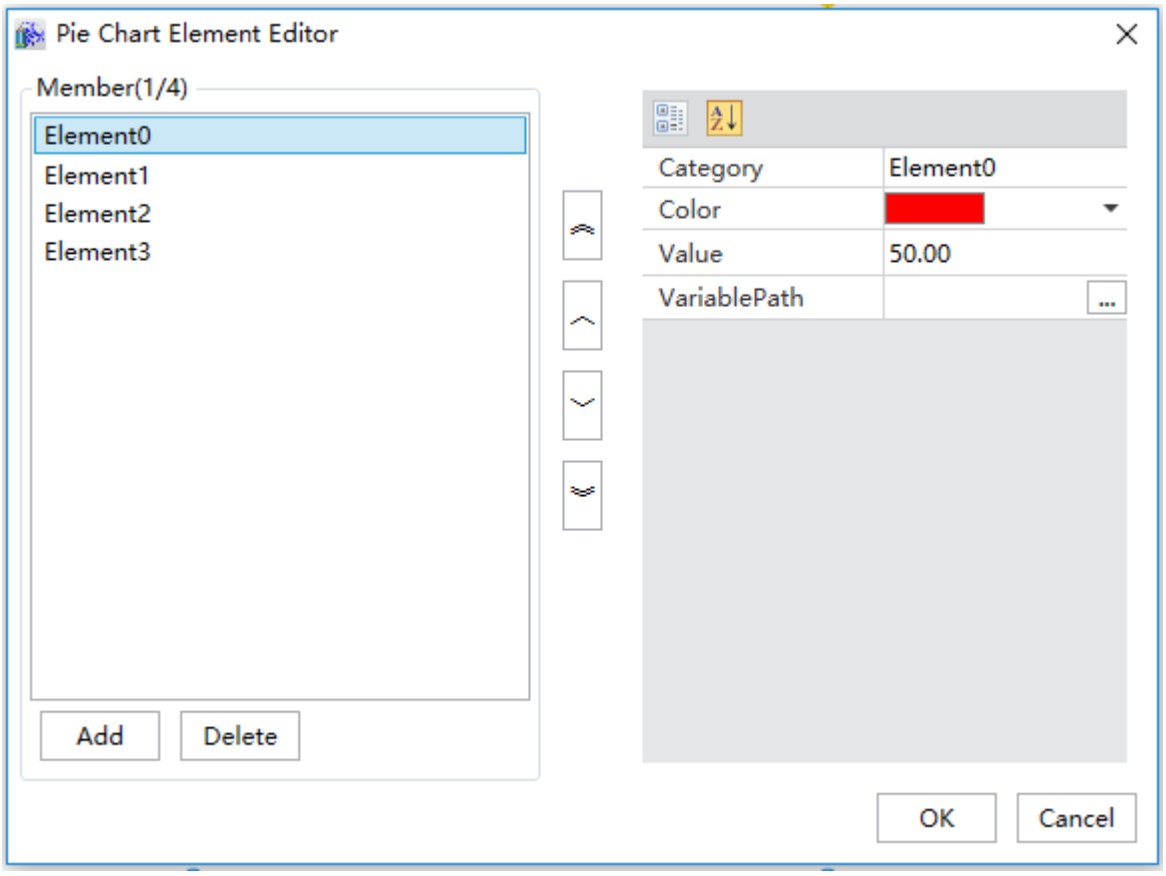
✧･ ColorPalette:Set the color palette(Metro,Custom)

✧･ LabelContent:Set the label content(YValue,XValue,Percentage)

✧･ LabelFormat:Set the label format(Int,Double1,Double2,Scientific,Percent)

✧･ Collection:Configures the components and connected variables of the pie chart; it configures each component's color and value etc. The configuration method is as follows:

Click the“  ”button and the pie chart element editor window will appear; it has 4 elements preset, as shown in the figure below:



The left part is add or delete members，each member is a series；the middle part can rank this series，the right part is the selected series properties，as shown in the table below:

Properties	Description
Name	Sets the name of the element unit.
Variable path	Sets the variable associated with the element

	unit.
Color	Sets the fill color of the element unit.
Value	Sets the initial value of the element unit.

✧‧ For other property settings please refer to the section “7.4 Graphic universal properties”.

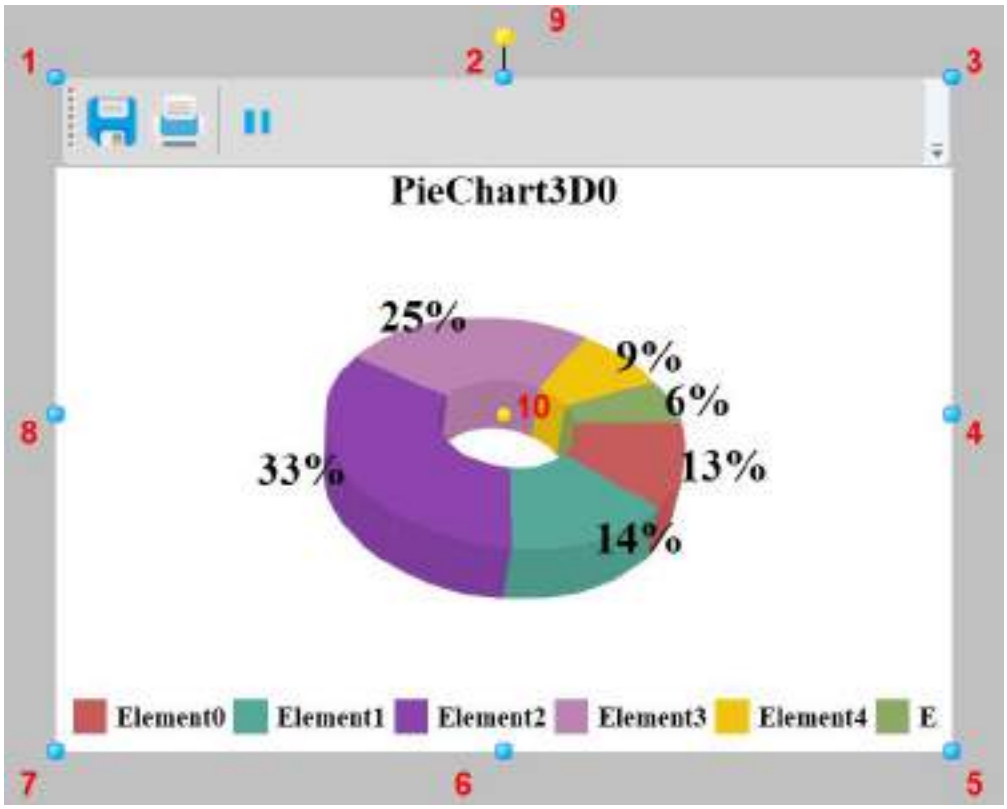
7.7.6 PieChart3D

➢‧ Open the window interface and click on “Toolbox” → “Extended controls” → “Pie chart 3D” in the tools window to the left; the toolbar is as shown in the figure above.

➢‧ Move the mouse to the working area of the window, select a starting point and press the left mouse button and drag the mouse towards the bottom-right; after releasing the left mouse button, the drawing of a pie chart 3D is complete, and a pie chart 3D will be generated.

➢‧ Just repeat the steps above if another XY chart needs to be drawn.

Graphic introduction:



☞ The figure above is a selected pie chart 3D; click on the pie chart 3D to enter selected status.

☞ There are two points on 1 in the figure above: the tensile point and the center; 2-8 are the tensile points and 9 is the rotation point,10 is the center.

☞ Above point 2 and below point 6 in the figure are the horizontal distortion points. The right of point 4 and the left of point 8 are the vertical distortion points.

PieChart3D properties

✧ Title: Set the title of the chart,Support for multilingual functionality.

✧ TitleFont: Set the font style for the chart.

✧ FontColor: Set the font color for the chart

✧ DisplayLegend: Sets whether to display legend of a chart

✧ 3DTiltAngle: Set 3D pie chart tilt angle

✧ 3DHeight: Set 3D pie chart height

✧ 3DRotation: Set whether you can rotate the 3D pie chart

✧･ 3DRotationAngle: Set 3D pie chart rotation angle

✧･ RefreshTime: Set refresh time.Default value is one second

✧･ DisplayData:Set the chart to display data

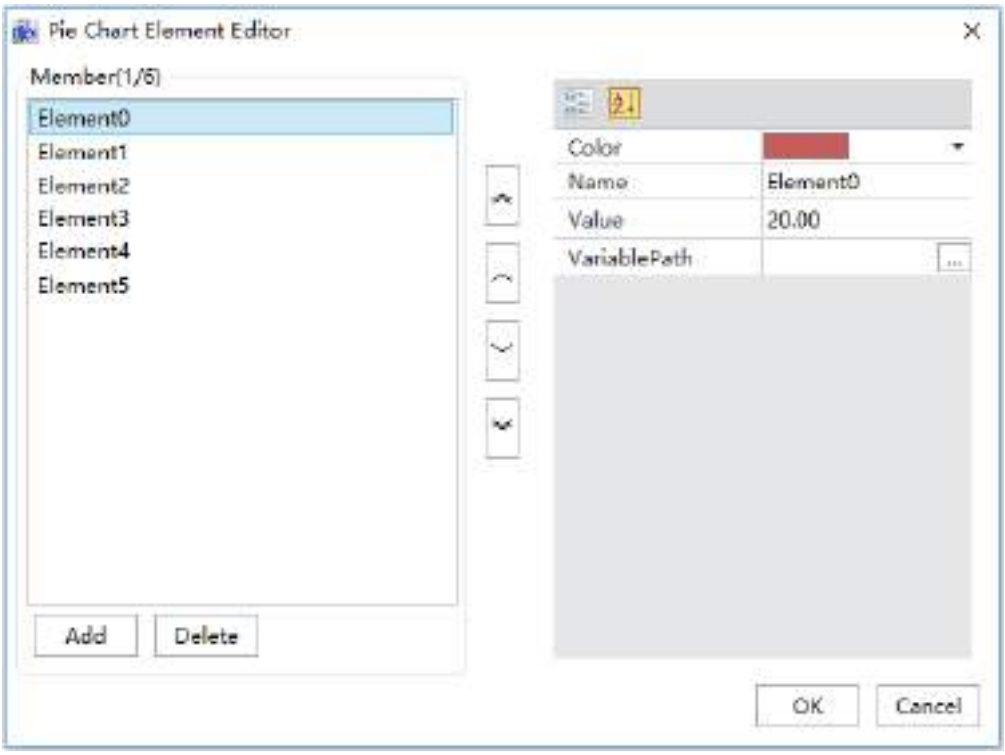
✧･ DisplayPoint:Set the chart to display point

✧･ DisplayDataFormat:Set up series data format

✧･ Toolbar:Set whether to display the toolbar

✧･ Collection: Configures the components and connected variables of the pie chart 3D; it configures each component's color and value etc. The configuration method is as follows:

Click the"  "button and the pie chart 3D element editor window will appear; it has 6 default elements, as shown in the figure below:



The left part is add or delete members, each member is a series; the middle part can rank this series, the right part is the selected series properties, as shown in the table below:

Properties	Description
Name	Sets the name of the element unit.

Variable path	Sets the variable associated with the element unit.
Color	Sets the fill color of the element unit.
Value	Sets the initial value of the element unit.

✧･ For other property settings please refer to the section “7.4 Graphic universal properties”.

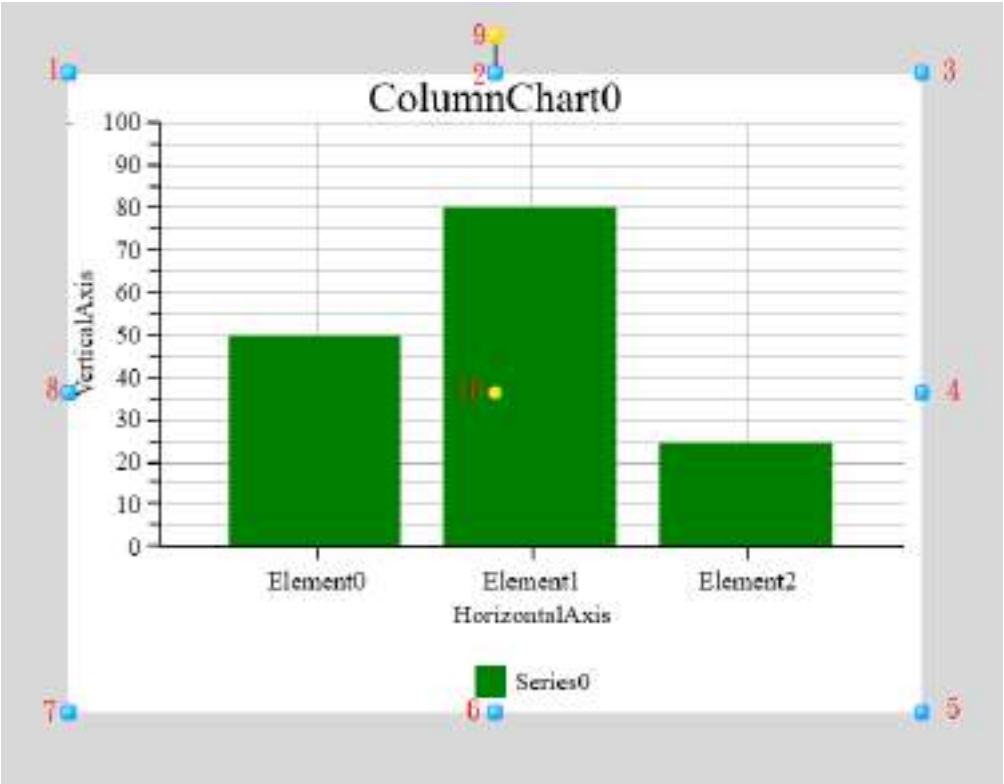
7.7.7 ColumnChart

➢ Open the window interface and click on “Toolbox” → “ExtendedControls” → “ColumnChart” in the tools window to the left; the toolbar is as shown in the figure above.

➢ Move the mouse to the working area of the window, select a starting point and press the left mouse button and drag the mouse towards the bottom-right; after releasing the left mouse button, the drawing of the column chart is complete, and the column chart will be generated.

➢ Just repeat the steps above if another column chart needs to be drawn.

Graphic introduction:



☞ The figure above is a selected column chart; click on the column chart to enter selected status.

☞ There are two points on 1 in the figure above: the tensile point and the center; 2-8 are the tensile points and 9 is the rotation point,10 is the center.

☞ Above point 2 and below point 6 in the figure are the horizontal distortion points. The right of point 4 and the left of point 8 are the vertical distortion points.

ColumnChart properties:

✧ Title: Set the title of the chart,Support for multilingual functionality.

✧ TitleFont: Set the title font

✧ TitleColor: Set the title color

✧ LegendFont:Set the legend font

✧ LegendColor:Set the legend color

✧ LegendLocation:Set the legend location

✧ UpperAndLowerDisplay:Set whether to display the limit line

✧･ LimitLineMaximum:Set the maximum value of the limit line

✧･ LimitLineMinimum:Set the minimum value of the limit line

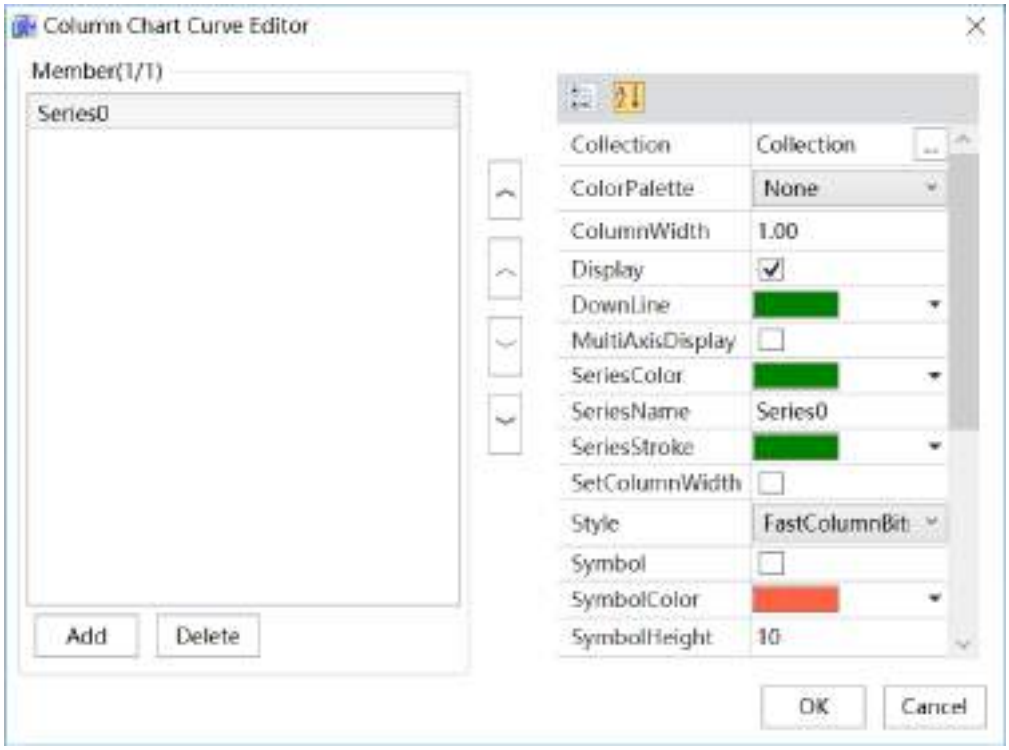
✧･ UpperLimitLineColor:Set upper limit color

✧･ LowerLimitLineColor:Set lower limit color

✧･ Thickness:Set the thickness of the limit line

✧･ SeriesCollection: Configures the curve in the chart that is associated with the variables to perform real-time display; the configuration method is as follows:

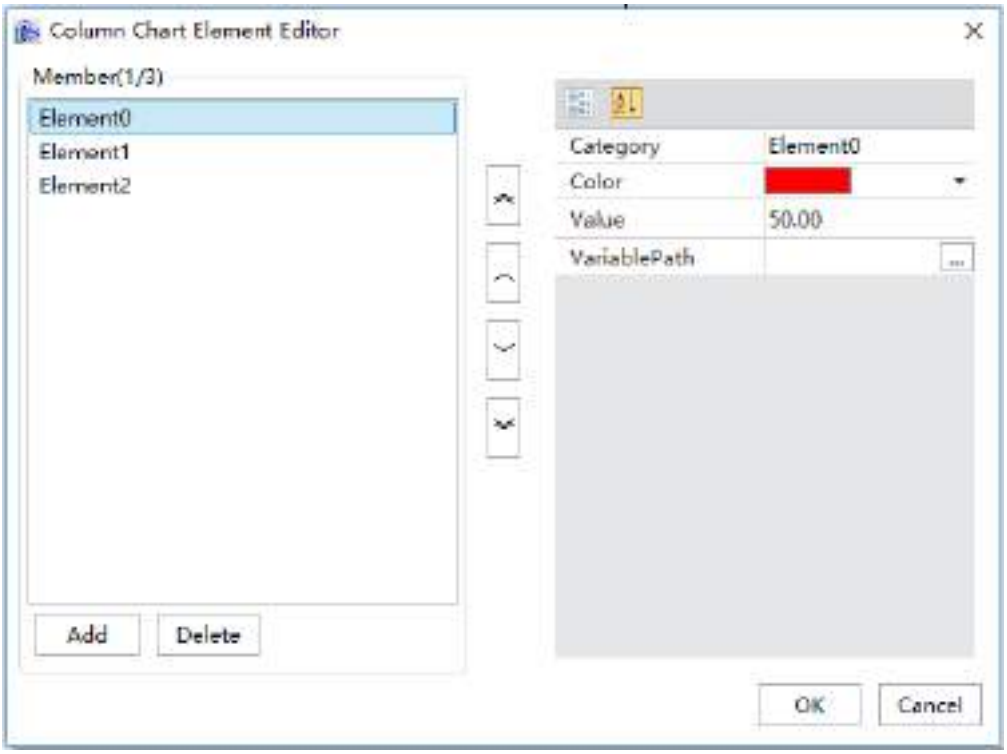
Click the“  ”button and the histogram curve editor window will appear; it has one curve preset, as shown in the figure below:



Properties	Description
Collection	Set the element of the column chart
ColorPalette	Set the color palette
ColumnWidth	Set the width of column
Display	Set whether to display the element

DownLine	Set the curve color below the lower limit line (column style)
MultiAxisDisplay	Set whether to display the multi axis
SeriesColor	Set series color
SeriesName	Set the name of the series
SeriesStroke	Set series stroke
SetColumnWidth	Set whether to set the length of the column. Check whether to set the length of the column (column style).
Style	Sets the display style of the histogram graphic elements
Symbol	Set whether to display symbol
SymbolColor	Set the symbol background color of numerical point
SymbolHeight	Set the symbol height of numerical point
SymbolStyle	Set the style of the dat symbol
SymbolWidth	Set the symbol width of numerical point
Text	Set whether to display the text
TextFont	Set text font
TextFormat	Set text format
TextType	Set text type
UpLine	Set the color of the curve when it is greater than the upper limit line (column style)
VerticalAxis	Sets the parameters of the vertical axis displayed by multiple axes
Width	Set the thickness of series

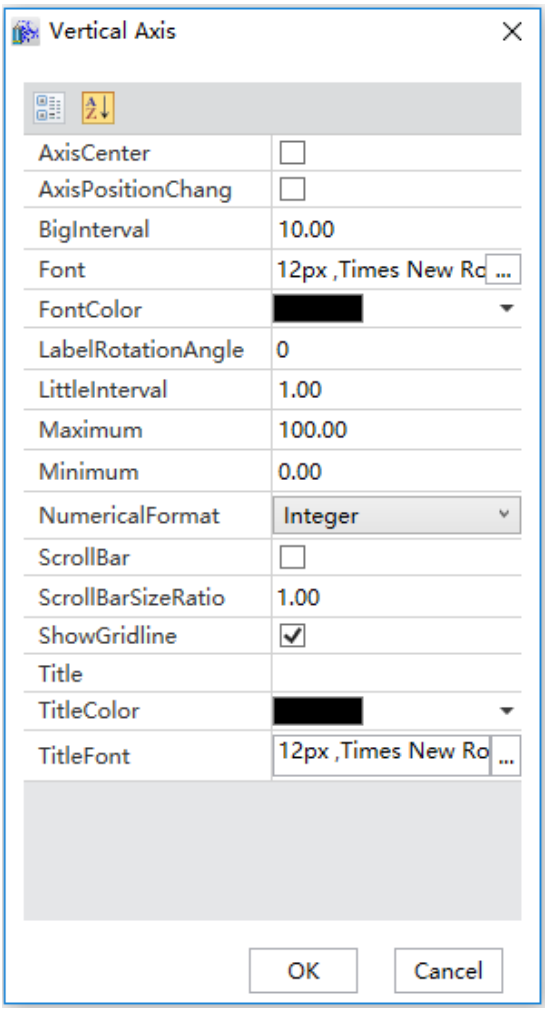
●·**Column Chart Element Editor:** click “Collection” configuration button，open the editor to configure the element as follows:



The left part is add or delete members，each member is a series；the middle part can rank this series，the right part is the selected series properties，as shown in the table below:

Properties	Description
Category	Sets the name of the element unit.
Variable path	Sets the variable associated with the element unit.
Color	Sets the fill color of the element unit.
Value	Sets the initial value of the element unit

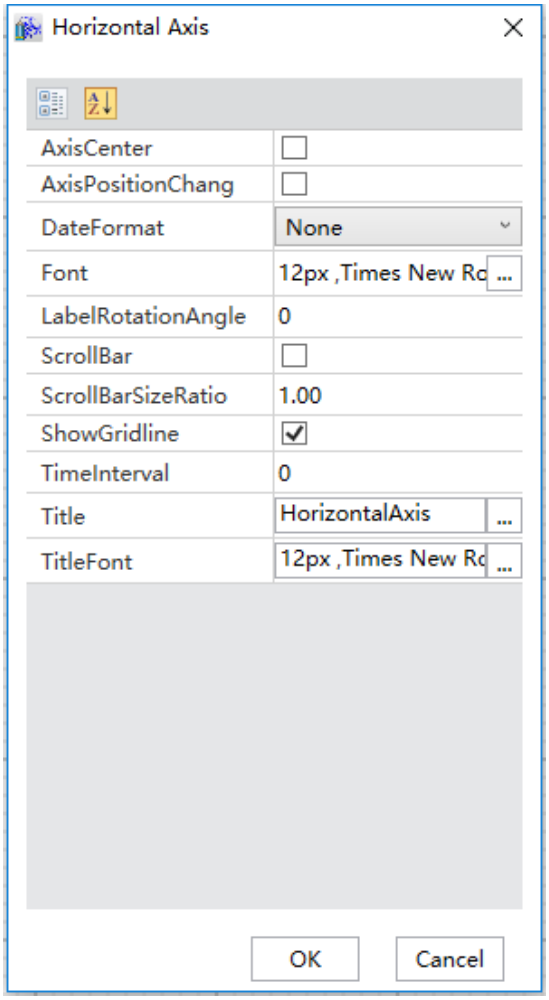
✧· Vertical axis



Properties	Description
AxisCenter	Set whether the axis is centered
AxisPositionChange	Set whether to change axis position
BigInterval	Set big interval
Font	Set the numerical font style
FontColor	Set the color of the value
LabelRotationAngle	Set the rotation Angle of the numerical axis
LittleInterval	Set little interval
ShowGridline	Set whether to display the grid line
Title	Set the title of the axis,Support for multilingual functionality.
TitleColor	Set title color
TitleFont	Set title font
Maximum	Set the maximum value
Minimum	Set the minimum value
NumericalFormat	Set the numrical format

ScrollBar	Set whether to display the scoll bar
ScrollBarSizeRatio	Set the size ratio of scroll bar

✧･ Horizontal axis



Properties	Description
AxisCenter	Set whether the axis is centered
AxisPositionChange	Set whether to change axis position
DateFormat	Set the date format
Font	Set the numerical font style
LabelRotationAngle	Set the rotation Angle of the numerical axis
ScrollBar	Set whether to display the scoll bar
ScrollBarSizeRatio	Set the size ratio of scroll bar
ShowGridline	Set whether to display the grid line
TitleInterval	Set the time interval
Title	Set the title of the axis,Support for

	multilingual functionality.
TitleFont	Set title font

✧· For other property settings please refer to the section “7.4 Graphic universal properties”.

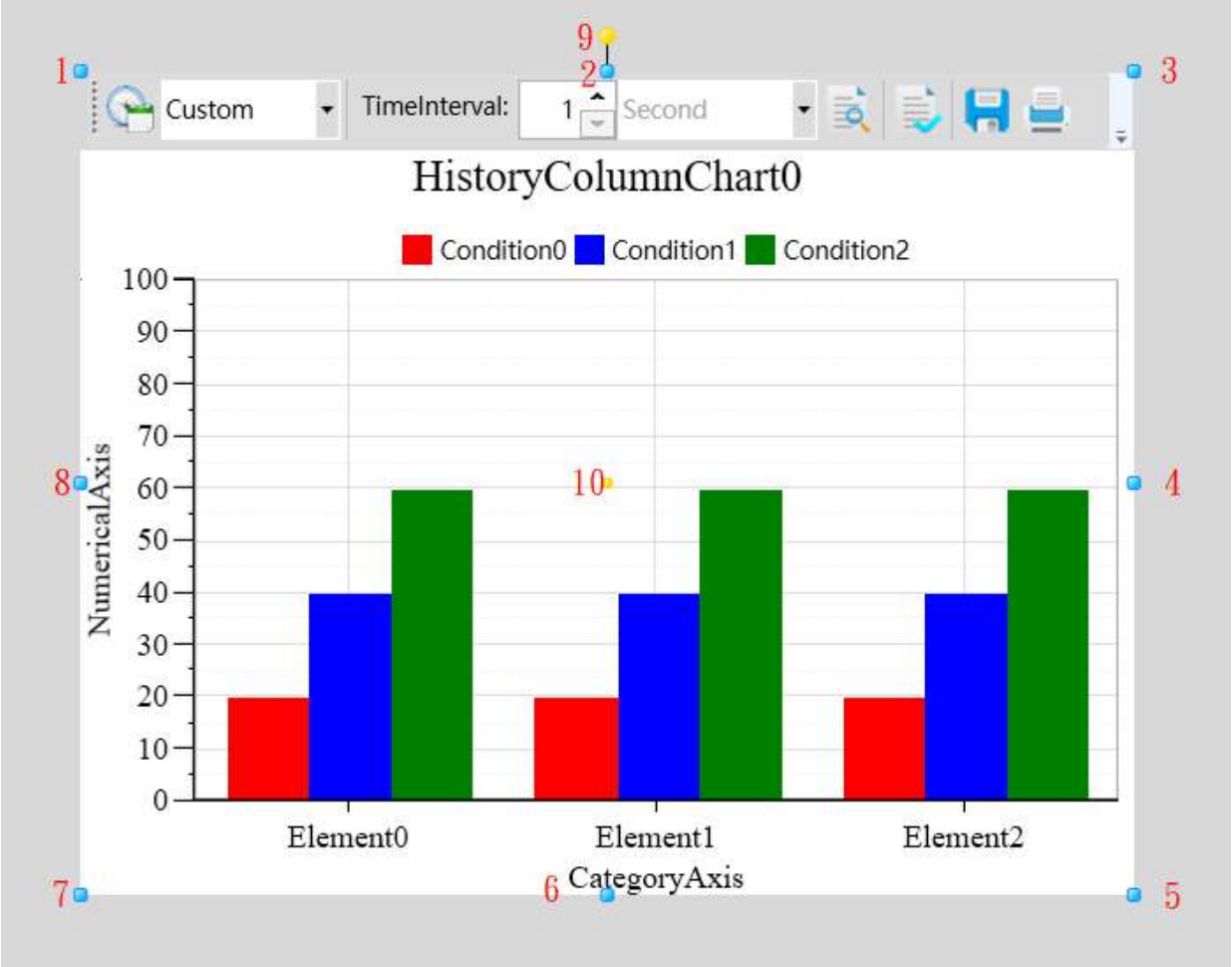
7.7.8 HistoryColumnChart

➢· Open the window interface and click on “Toolbox” → “Extended Controls” → “HistoryColumnChart” in the tools window to the left; the toolbar is as shown in the figure above

➢· Move the mouse to the working area of the window, select a starting point and press the left mouse button and drag the mouse towards the bottom-right; after releasing the left mouse button, the drawing of a history column chart is complete, and a history column chart will be generated

➢· Just repeat the steps above if another history column chart needs to be drawn.

Graphic introduction:



☞· The figure above is a selected history column chart; click on the history column chart to enter selected status.

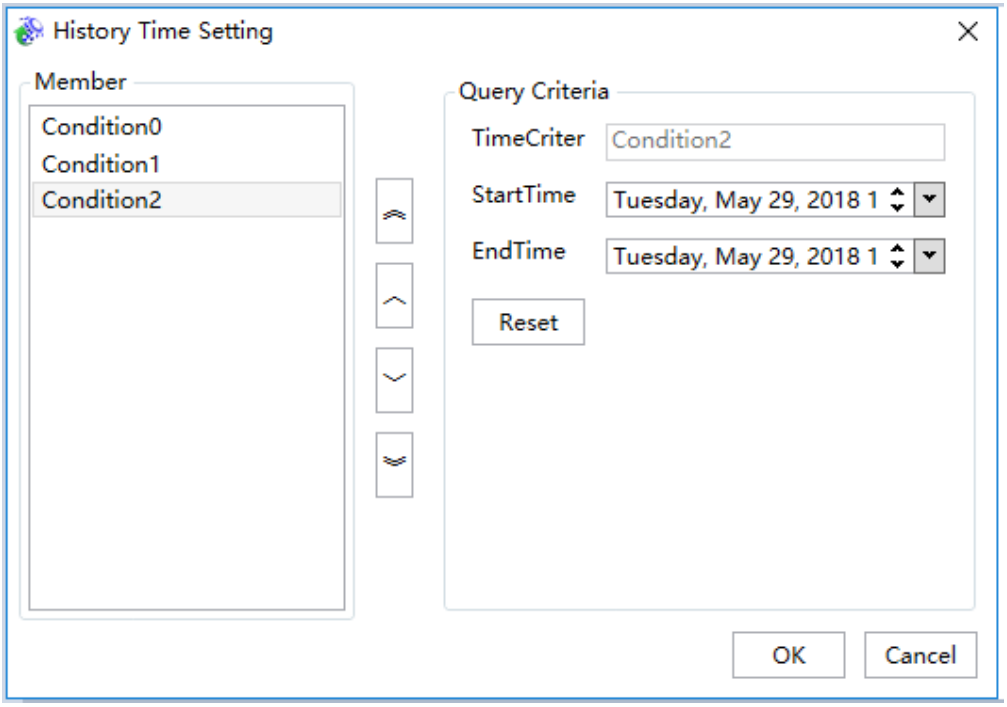
☞· There are two points on 1 in the figure above: the tensile point and the center; 2-8 are the tensile points and 9 is the rotation point,10 is the center.

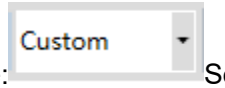
☞· Above point 2 and below point 6 in the figure are the horizontal distortion points. The right of point 4 and the left of point 8 are the vertical distortion points

☞· It is the toolbar on the top of figure which is used to operate chart.

Usage of the toolbar

✧· Time setting:  Set the start and end time of the query



✧· Set nearest query time: 

Set the nearest query time period,respectively are:Last hour, last day, last week, last month, last three months, last six months, last year, last three years, custom；

✧· TimeInterval: 

when "set recent query time" is not "custom"，user can input time interval，its unit can be: second、minute、hour、day、month、year、Default（preset time interval）

✧· Query:  query data according to the current configuration

✧· Display positioning line:  whether to display the position line

✧· Save:  history chart is saved as picture

✧· Print:  print current history chart

HistoryColumnChart properties

✧· Title: Set the title of a chart,Support for multilingual functionality.

✧･ TitleFont: Set the font style for the chart.

✧･ FontColor: Set the font color for the chart.

✧･ DisplayLegend: Whether display legend of a chart.

✧･ MajorGrid: Set major grid.

✧･ MinorGrid: Set minor grid.

✧･ VerticalAxisAutoChange:Set whether to automatically adjust the maximum value of the vertical axis.

✧･ PositioningLine:Set whether to display the positioning line.

✧･ PositioningLineColor:Set the color of the positioning line.

✧･ UpperAndLowerDisplay:Set whether to display the limit line.

✧･ UpperLimitLineColor:Set upper limit color.

✧･ LowerLimitLineColor:Set lower limit color.

✧･ LimitLineMaximum:Set the maximum value of the limit line.

✧· LimitLineMinimum:Set the minimum value of the limit line.

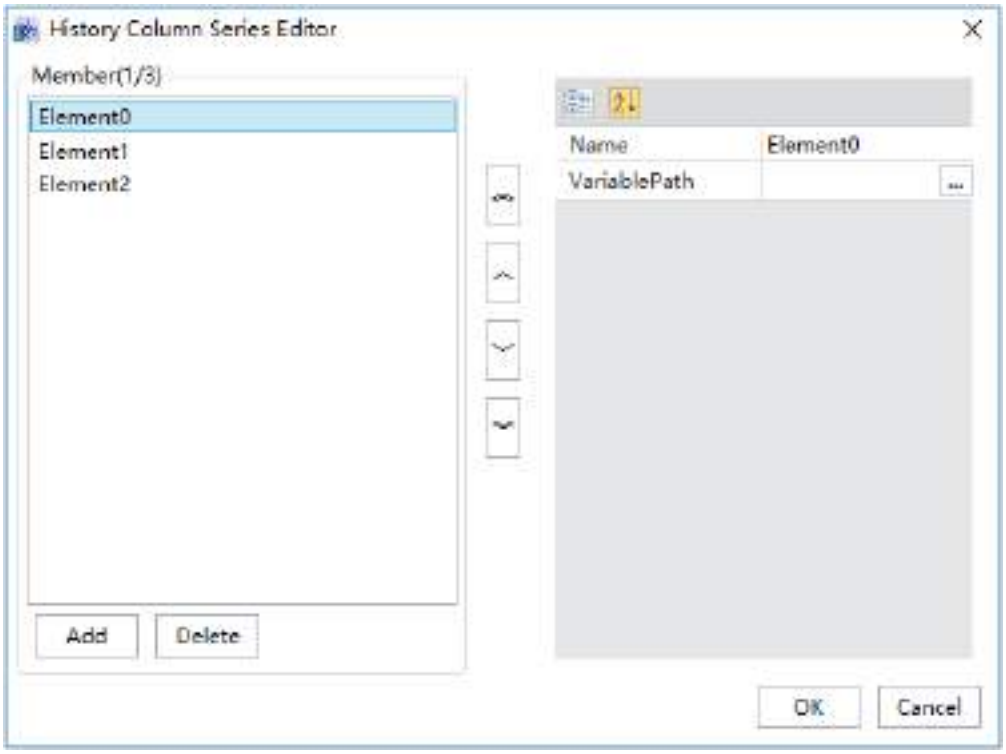
✧· Toolbar:Whether display toolbar of a chart.

✧· ToolbarIcon:Set toolbar icon type.

✧· CategoryList: Configure the variable path of the series，the configuration method is as follows:

The left part is add or delete members，each member is a series；the middle part can rank this series，the right part is the selected series properties，as shown in the table below:

Click“ ··· ”button，pup up the History Column Series Editor window，the default members are three elements as follows:

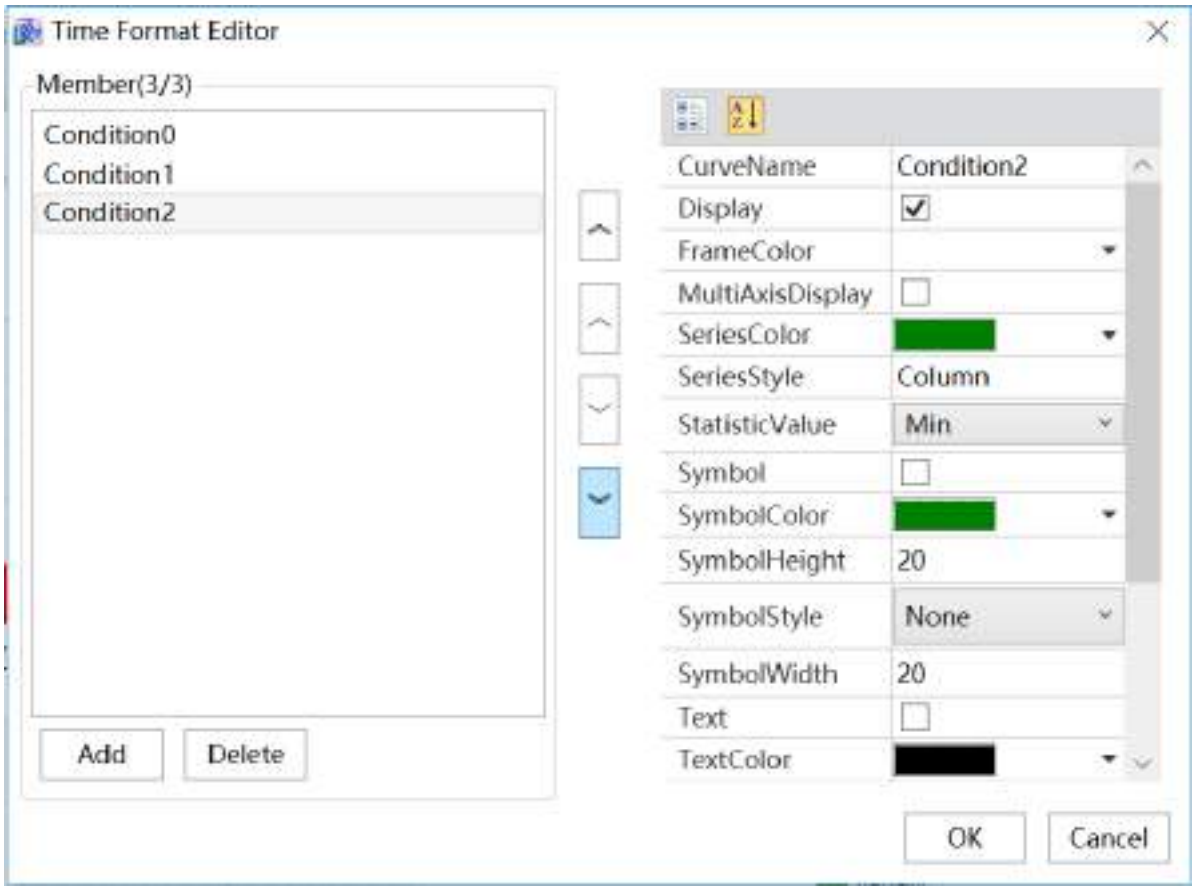


Properties	Description
Name	Set the name of the chart
Variable path	Set the associated history varialbe of the chart

✧· DateConditionList: configure the the curve style in the diagram, the configuration is as follows:

The left part is add or delete members，each member is a series；the middle part can rank this series，the right part is the selected series properties，as shown in the table below:

Click“ ··· ” button，pup up the "Time Format Editor" window，the default members are three conditions as follows:



The left part is add or delete members，each member is a series；the middle part can rank this series，the right part is the selected series properties，as shown in the table below:

Properties	Description
CurveName	Set curve name
Display	Set whether to display the curve
FrameColor	Set frame color
MultiAxisDisplay	Set whether to display the multi axis
SeriesColor	Set series color
SeriesStyle	Set series style
StatisticValue	Sets the statistical value type of series(Min,Avg,Max)
Symbol	Set whether to display symbol
SymbolHeight	Set the symbol height of numerical point
SymbolWidth	Set the symbol width of numerical point
SymbolColor	Set the symbol background color of numerical point

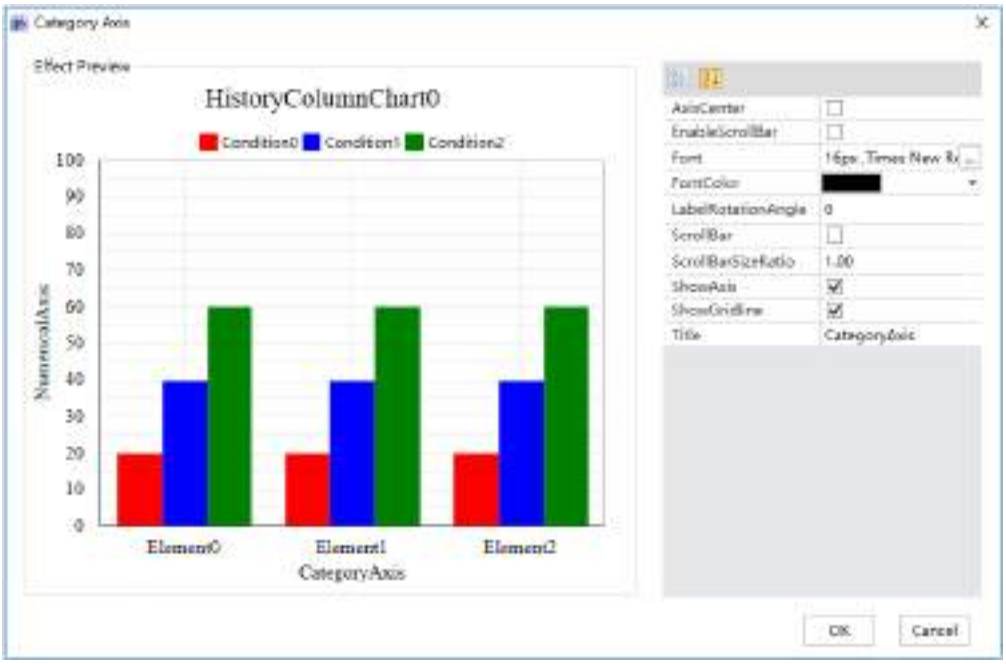
SymbolStyle	Set the style of the dat symbol
SymbolWidth	Set symbol width
Text	Set whether to display the text
TextColor	Set text color
TextFont	Set text font
TextFormat	Set text format
TextType	Set text type
VerticalAxis	Sets the parameters of the vertical axis displayed by multiple axes
Width	Set the thickness of serie

✧· NumericalAxis

Properties	Description
AxisCenter	Set whether the axis is centered
BigInterval	Set big interval
EnableScrollBar	Set whether to enable scroll bar
Font	Set the numerical font style
FontColor	Set the color of the value
LabelRotationAngle	Set the rotation Angle of the numerical axis
LittleInterval	Set little interval
ShowAxis	Set whether to show axis
ShowGridline	Set whether to display the grid line
Title	Set the title of the axis,Support for multilingual functionality.

TitleColor	Set title color
TitleFont	Set title font
Maximum	Set the maximum value
Minimum	Set the minimum value
NumericalFormat	Set the numrical format
ScrollBar	Set whether to display the scoll bar
ScrollBarSizeRatio	Set the size ratio of scroll bar

✧･ CategoryAxis



Properties	Description
AxisCenter	Set whether the axis is centered
EnableScrollBar	Set whether to display scroll bar
Font	Set the numerical font style
FontColor	Set the color of the value
LabelRotationAngle	Set the rotation Angle of the numerical axis
ShowAxis	Set whether to show axis
ShowGridline	Set whether to display the grid line
Title	Set the title of the axis,Support for multilingual functionality.
ScrollBar	Set whether to display the scoll bar
ScrollBarSizeRatio	Set the size ratio of scroll bar

Statusbar

✧･ StatusDisplay:Set whether the status bar is displayed

✧・ NumericalFormat:Set the format of the status bar data display.

✧・ Font:Set the font style for the status bar.

✧・ BackgroundColor:Set the background color of the status bar.

✧・ FontColor:Set the font color of the column name in the status bar.

✧・ StatusGrid:Set whether the status bar displays grid lines.

✧・ For other property settings please refer to the section “7.4 Graphic universal properties”.

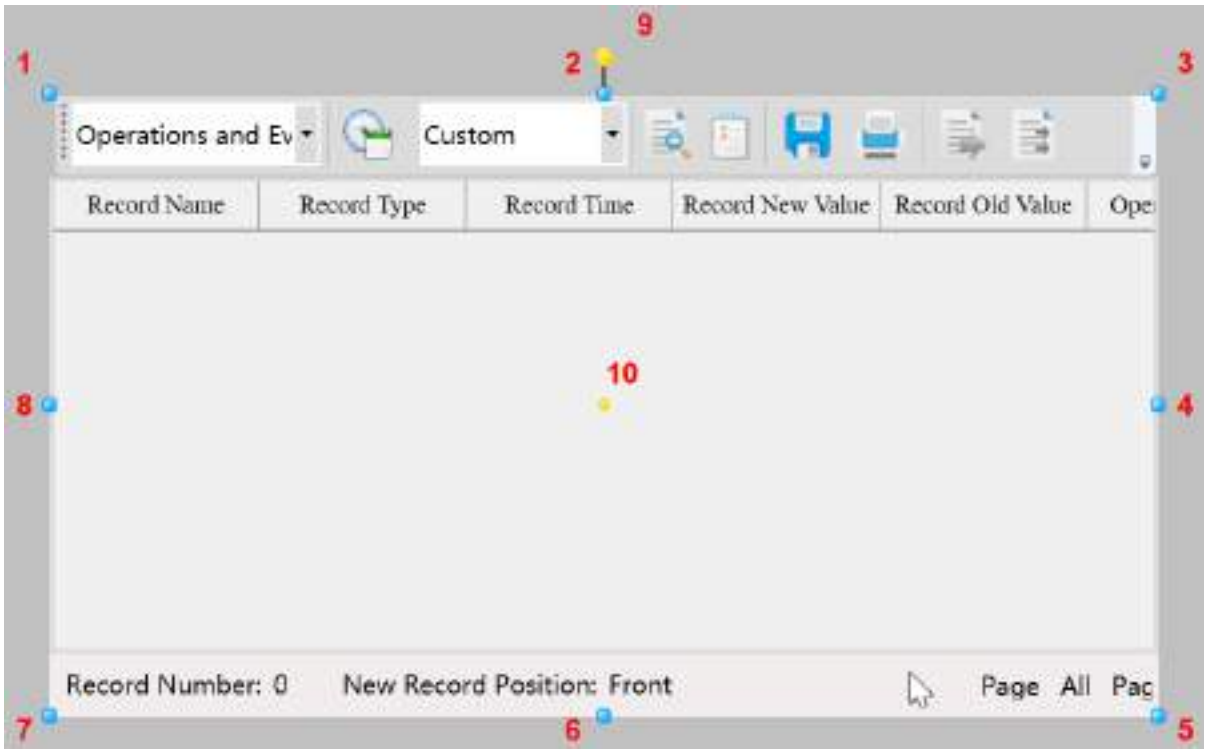
7.7.9 RecordBox

➢・ Open the window interface and click on “Toolbox” → “Extended controls” → “Record box” in the tools window to the left; the toolbar is as shown in the figure above

➢・ Move the mouse to the working area of the window, select a starting point and press the left mouse button and drag the mouse towards the bottom-right; after releasing the left mouse button, the drawing of a record box is complete, and a record box will be generated.

➢・ Just repeat the steps above if another history chart needs to be drawn.

Graphic introduction:



☞· The figure above is a selected record box; click on the record box to enter selected status

☞· There are two points on 1 in the figure above: the tensile point and the center; 2-8 are the tensile points and 9 is the rotation point,10 is the center.

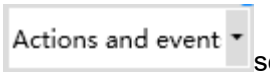
☞· Above point 2 and below point 6 in the figure are the horizontal distortion points. The right of point 4 and the left of point 8 are the vertical distortion points.

☞· It is the toolbar on the top of figure which is used to operate record box.

Usage of the toolbar

✧· Query:  query data according to the current configuration

✧· Set the status column format:  set whether to display the specific column in the status column

✧· Query object: 

select the query object

✧· Set recent query time: 

set the query time range: recent one hour、recent one day、recent one week、recent one month、recent three months、recent six months、recent one

year、recent three year、custom

✧· Set query time:  when "set recent query time" is "custom"，user can set self-defined query range

✧· Save:  history chart is saved as picture

✧· Print:  print current history chart

✧· Export current data:  export the current page data

✧· Export all data:  export all the data

Record box properties

✧· Font: Set font

✧· FontColor: Set font color

✧· StatusbarFontColor:Set the color of the status bar font.

✧· Background: Set control background.

✧· StatusBarDisplay: Set whether to display the status bar.

✧· ShowGridline: Set whether to display grid.

✧· NewRecordPosition: Set the location of the latest records.

✧· Toolbar: Set whether to display the toolbar

✧· For other property settings please refer to the section “7.4 Graphic universal properties”.

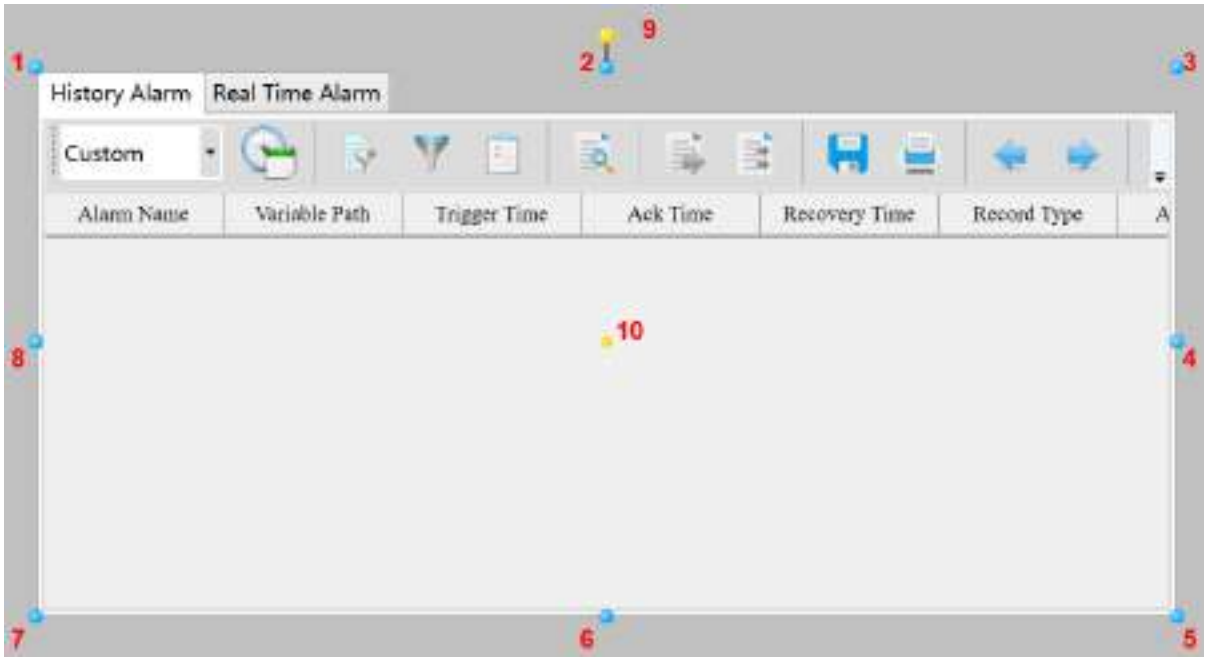
7.7.10 AlarmWindow

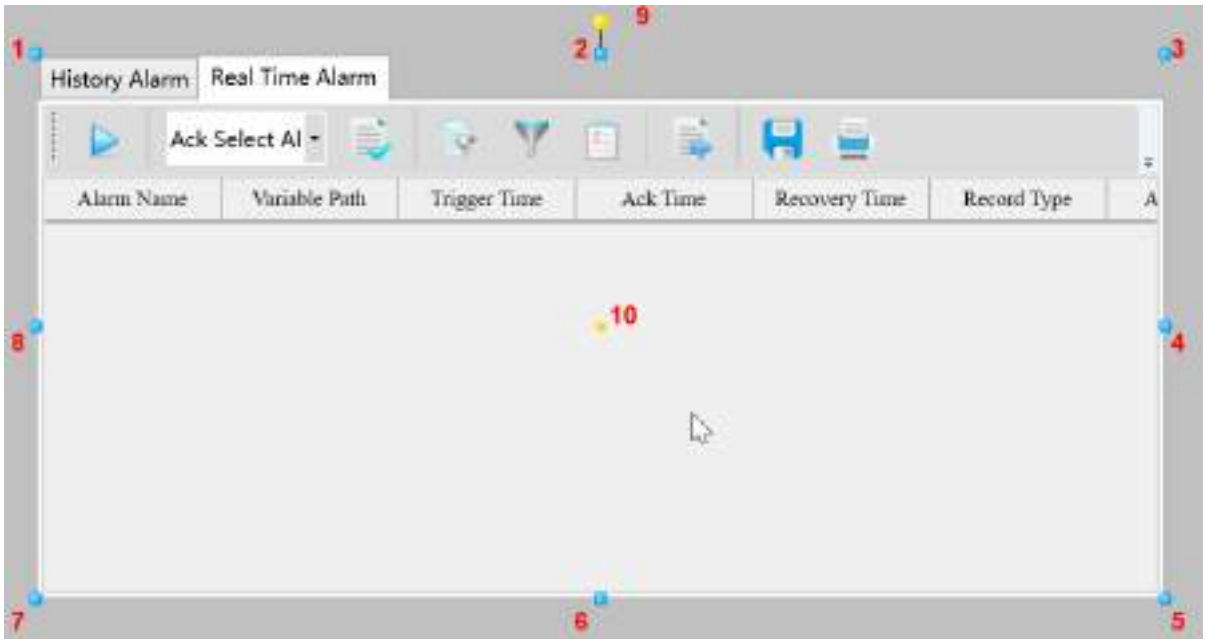
➢· Open the window interface and click on “Toolbox” → “Extended controls” → “Alarm window” in the tools window to the left; the toolbar is as shown in the figure above.

➢· Move the mouse to the working area of the window, select a starting point and press the left mouse button and drag the mouse towards the bottom-right; after releasing the left mouse button, the drawing of an alarm window is complete, and an alarm window will be generated.

➢· Just repeat the steps above if another alarm window needs to be drawn.

Graphic introduction:





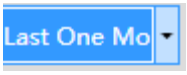
☞· The figure above is a selected alarm window; click on the alarm window to enter selected status

☞· There are two points on 1 in the figure above: the tensile point and the center; 2-8 are the tensile points and 9 is the rotation point,10 is the center.

☞· Above point 2 and below point 6 in the figure are the horizontal distortion points. The right of point 4 and the left of point 8 are the vertical distortion points.

Usage of the toolbar

➢· **History alarm**

✧· Set recent alarm query time: 

set the query time range: recent one hour,recent one day,recent one week,recent one month,recent three months,recent six months,recent one year,recent three year,custom

✧· Set recent alarm query time:  when "set recent alarm query time" is "all alarm",user can set self-defined query range

✧· Set query filter condition:  set query filter condition

✧· Alarm group filter:  set filter alarm according to alarm group

✧· Set the column need to display:  select the column need to display, if the column is checked , display it.

✧· Query history alarm:  query history alarm according to the current configuration

✧· Export current data:  export current page data

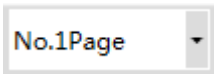
✧· Export all the data:  export all the data

✧· Save:  history alarm is saved as picture

✧· Print:  print current history alarm

✧· Previous:  Go to previous page

✧· Next:  Go to next page

✧· Go to specific page: 

go to uer selected page

➢· **Realtime alarm**

✧· Start loading realtime alarm data:  start to query realtime alarm data

✧· Stop loading realtime alarm data:  stop to query realtime alarm data

✧· Ack the selected alarm:  ack the selected alarm

✧· Set query filter condition:  set the query filter condition

✧･ Alarm group filter:  set filter according to alarm group

✧･ Set the column need to display:  elect the column need to display, if the column is checked , display it

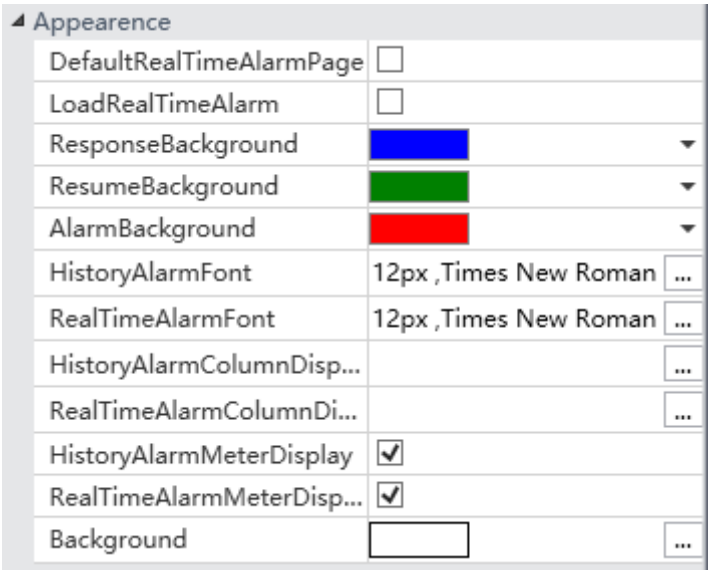
✧･ Exprot data:  export current data

✧･ Save:  realtime alarm is saved as picture

✧･ Print:  print current realtime alarm

Alarm window properties

Appearence



✧･ DefaultRealTimeAlarmPage:Set the IsRealTimeTabFirst property to select the default display page.

✧･ LoadRealTimeAlarm:Set the AutoLoadRealTimeAlarm property to run automatically when the real-time alarm.

✧･ ResponseBackground:Set the background color for the response.

✧･ ResumeBackground:Set the background color for the recovery of the record type.

✧･ AlarmBackground:Set the background color for the alarm.

✧･ HistoryAlarmFont:Set history alarm font size.

✧･ HistoryAlarmColumnDisplay:Set history alarm column display.

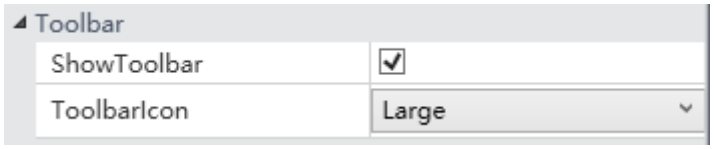
✧･ RealTimeAlarmColumnDisplay:Set real time alarm display.

✧･ HistoryAlarmMeterDisplay:Set whether to display the history alarm meter.

✧･ RealTimeAlarmMeterDisplay:Set whether to display the real-time alarm meter.

✧･ Background:Set control background.

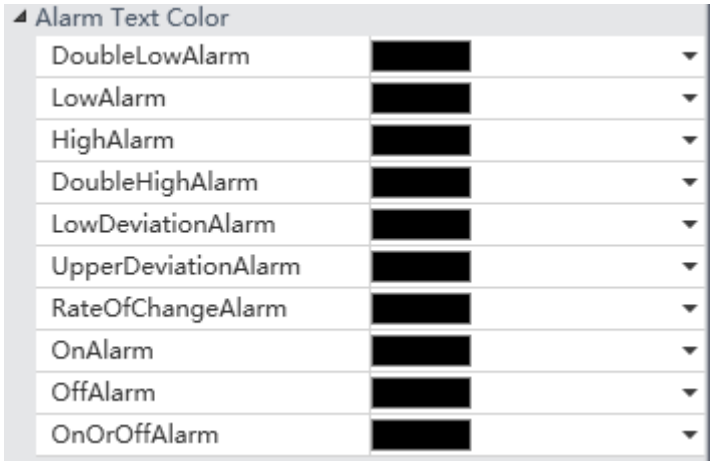
Toolbar



✧･ ShowToolbar:Set the alarm window toolbar wheather to show or not.

✧･ ToolbarIcon:Set alarm box toolbar icon type.

Alarm Text Color



✧･ DoubleLowAlarm:Set DoubleLow alarm text color.

✧･ LowAlarm:Set LowAlarm text color.

✧･ DoubleHighAlarm:Set DoubleHigh alarm text color.

✧･ HighAlarm:Set HighAlarm text color.

✧･ LowDeviationAlarm:Set LowDeviation alarm text color.

✧･ UpperDeviationAlarm:Set UpperDeviation alarm text color.

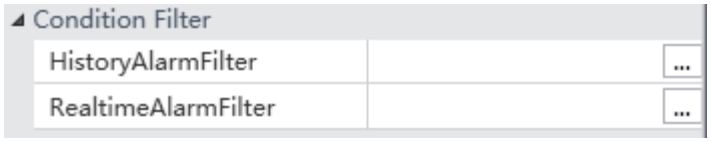
✧･ RateOfChangeAlarm:Set RateOfChange alarm text color.

✧･ OnAlarm:Set OnAlarm text color.

✧･ OffAlarm:Set OffAlarm text color.

✧･ OnOrOffAlarm:Set OnOrOff alarm text color.

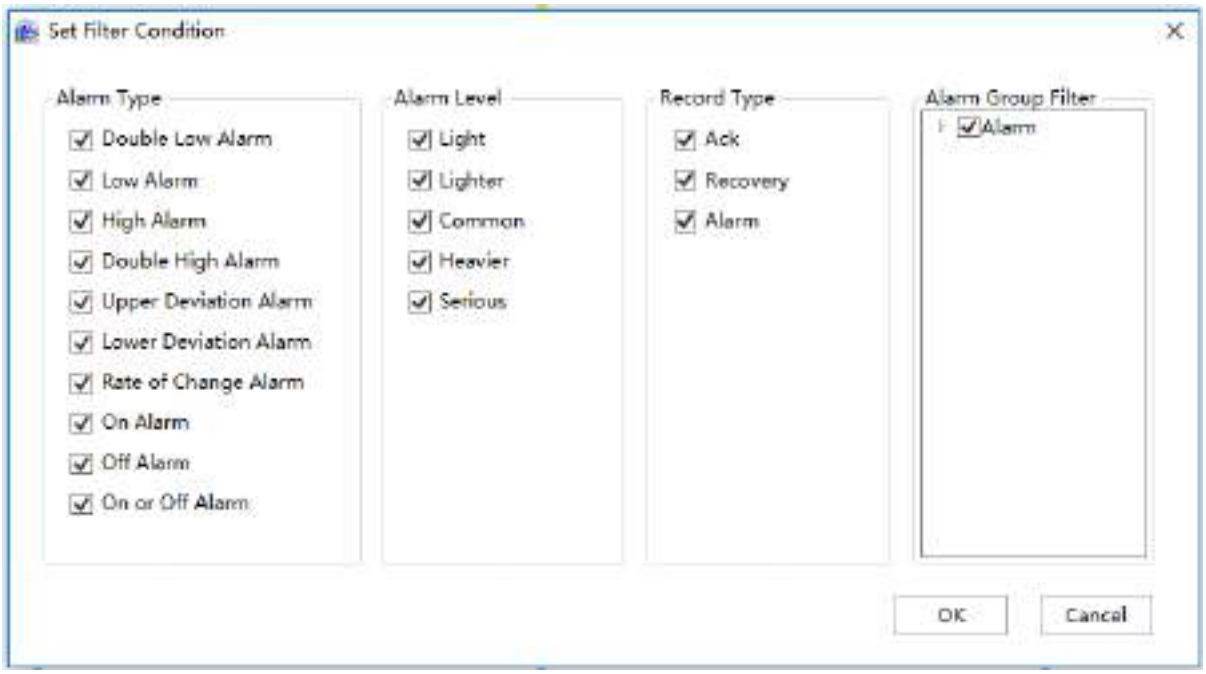
Condition Filter



✧･ HistoryAlarmFilter:Set history alarm filter condition.

✧･ RealtimeAlarmFilter:Set real time alarm filter condition.

Click“ ... ” button,pup up the "Set Filter Condition" window,users can choose according to their own needs:



✧· For other property settings please refer to the section “7.4 Graphic universal properties”.

7.7.11 Report

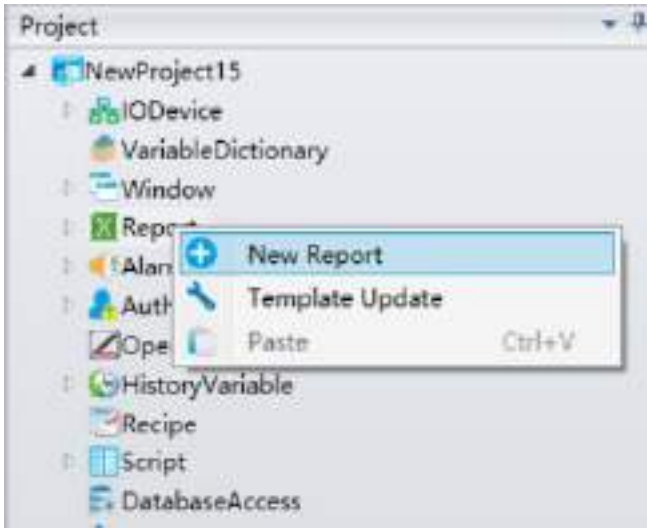
Report tool that offerintg "Semblable excel" operation is used to make common production report(daily report,monthly report, yearly report),which is also used to make various complicated format report flexibly to achive querying,displaying,printing and output to realtime,history and statistic data.

Report can not only display the data of realtime and history database but also can query and access to the typical database like Mysql, SQL Server, Oracel, Access etc.

7.7.11.1 Report template design

➢· Open the window interface and click on “Toolbox” → “Extended controls” → “Report” in the tools window to the left; the toolbar is as shown in the figure above.

➢· Just repeat the steps above if another recipe browser needs to be drawn.



Report menu introducution:

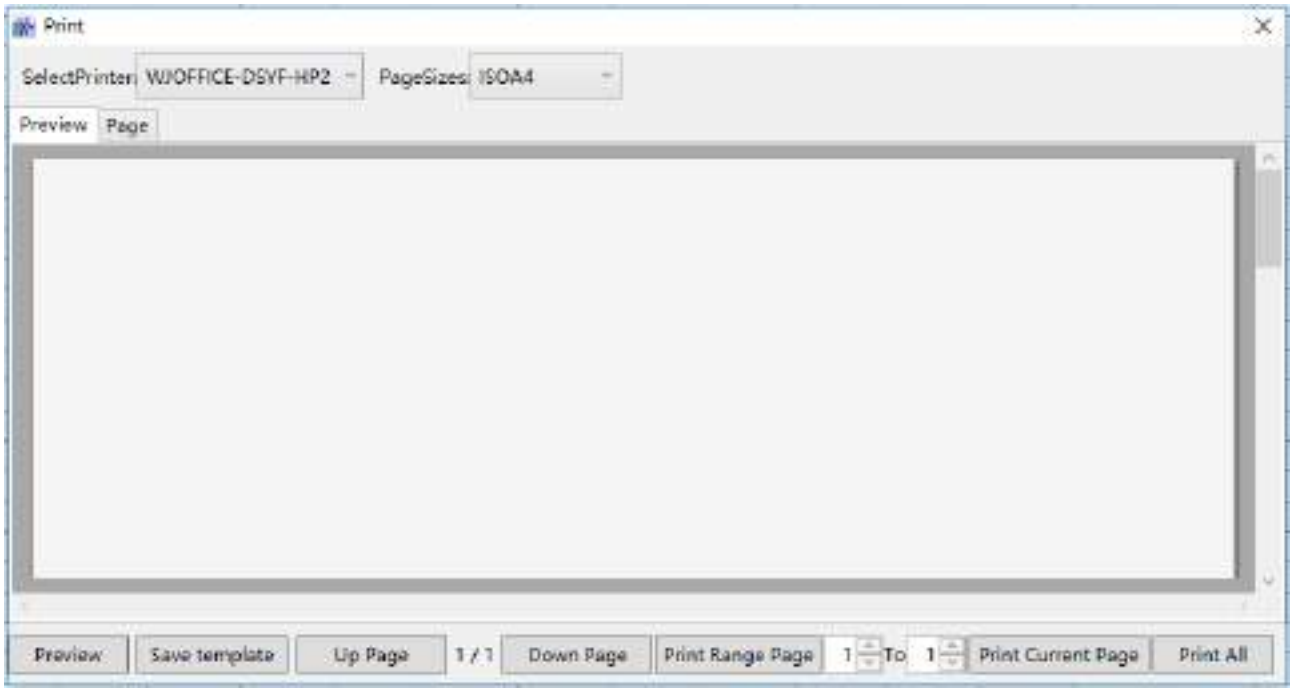
☞ The figure above is report template menu which can be used to design for report

Usage of the toolbar

✧ **Import:** Import report templates or excel tables

✧ **Export As:** Expor as excel, pdf, CSV file

✧ **Print:** Set report template printing parameter and print



✧･ Preview: preview current report

✧･ Save template: save current template

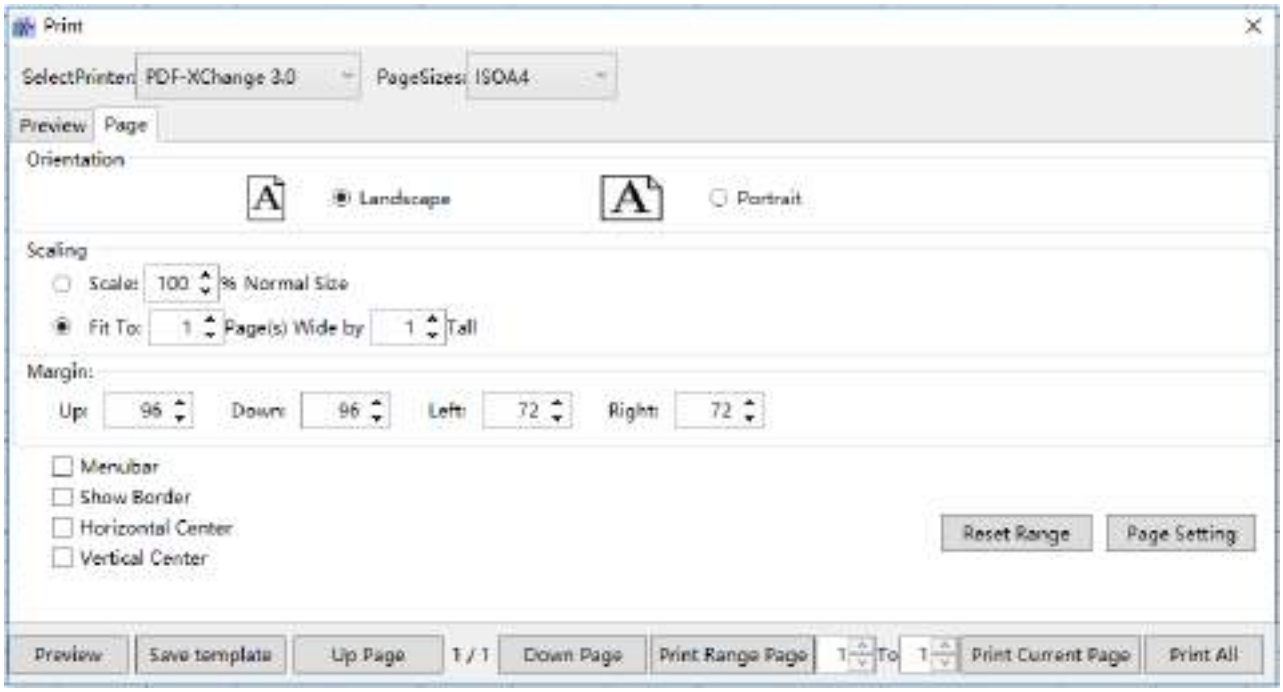
✧･ Up Page: Preview previous page

✧･ Down Page:Preview next page

✧･ Printing Range Page: Printing selected range page

✧･ Print Current Page: Print current page

✧･ Print All: Print all the page



✧･ Page setting:

Orientation: Landscape or portrait

Scaling: print accrording to ratio

1 page width 1 page height : adjust table to one page

1 page width 0 page height : adjust all column to one page

0 page width 1 page height : adjust all row to one page

0 page width 0 page height : no zoom

✧･ Font setting: set font style, size, bold,slope,underline,cell border style, cell fill color and font color

✧･ Word wrap: set wtether to word wrap if the content is more the length of cell

✧･ Merge and center: merge the selected cell or cancel merging

✧･ Insert Sheet: insert sheet

✧･ Insert Image: insert image

✧･ Delete Sheet: delete sheet

✧· Row Count: set row count

✧· Row Height: set row height

✧· Column Count: set column count

✧· Column Width: set column width

✧· Freeze Panes: freeze the selected row and column

✧· Freeze First Column: freeze first column

✧· Freeze Top Row: freeze the first row

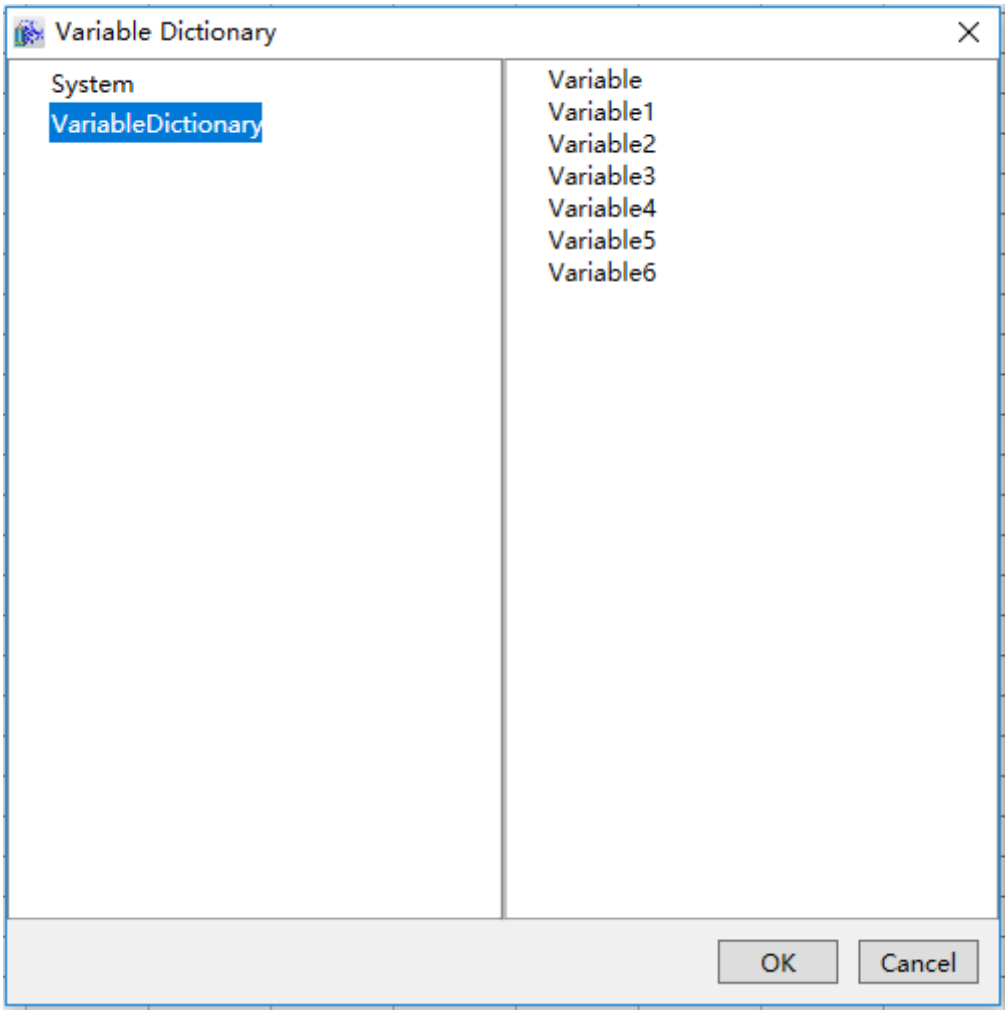
✧· Cancel Freeze: cancel freeze

✧· Protect Sheet: set whether to project sheet

✧· Realtime Variable: design realtime variable report template，the configuration is as follows:

Click“ 

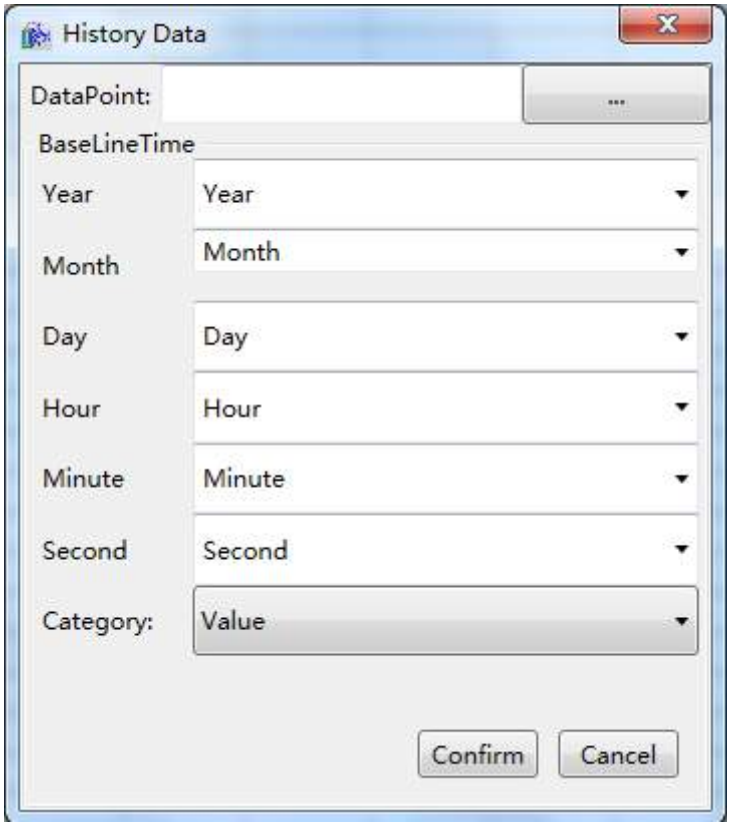
”button，pop up window to associate realtime variable as shown in the figure below:

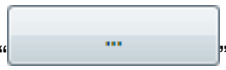


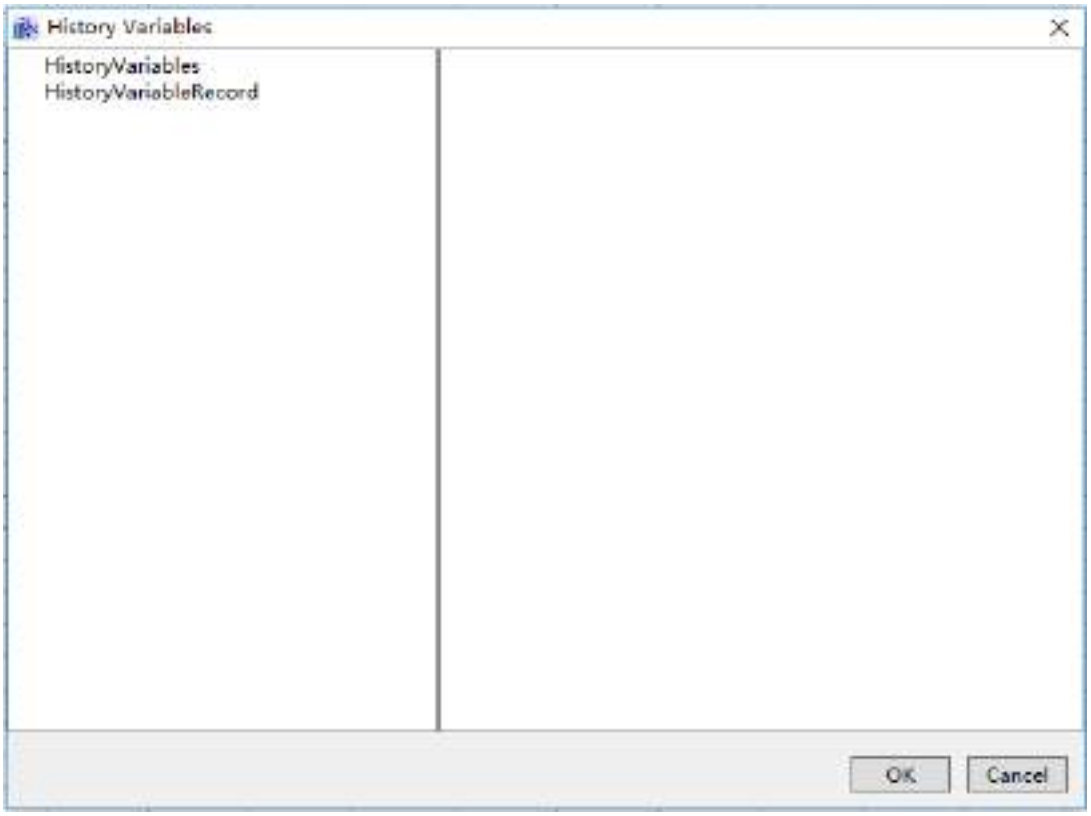
✧· History Data: design history variable report template，the configuration is as follows:

Click“ 

”button，pop up the window，as shown in the figure below:



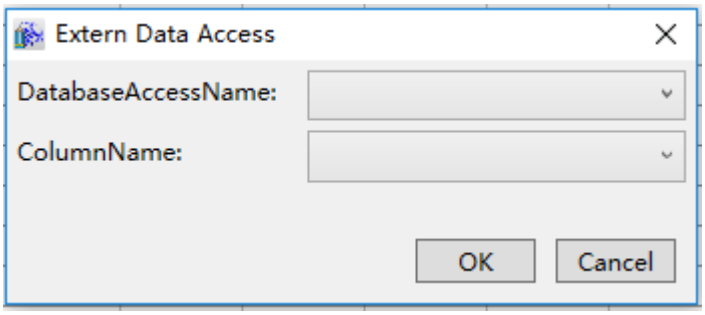
Click“  ”button，pup up window to associate history variable as shown in the figure below:



✧･ Database: design report for external data source, the configuration is as follows:

Click“ 

”button, pop up window to select “database access name”and “column name”, as shown in the figure below:

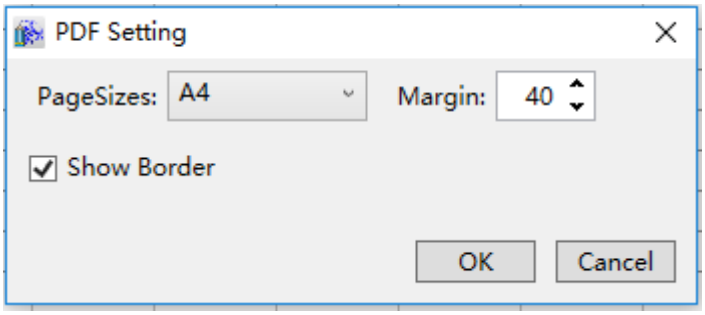


✧･ Print Setting

- Print Region: set report print region
- Cancel Print Region: cancel report print region

✧･ PDF Setting

- PDF Region



- Cancel PDF Region

Report template properties

✧･ ShowFormula: whether show formula bar

✧･ RowCount: Report current row counts

✧･ ColumnCount: Report current column counts

✧･ ShowHeader: Whether show row and column header

7.7.11.2 Report rendering

➢･ Open the window interface and click on “Toolbox” → “Extended controls” → “Report” in the tools window to the left; the toolbar is as shown in the figure above.

➢･ Move the mouse to the working area of the window, select a starting point and press the left mouse button and drag the mouse towards the bottom-right; after releasing the left mouse button, the drawing of a report is complete, and a report will be generated

➢･ Just repeat the steps above if another history chart needs to be drawn.

Graphic introduction:

☞ The figure above is a selected history chart; click on the history chart to enter selected status.

☞ There are two points on 1 in the figure above: the tensile point and the center; 2-8 are the tensile points and 9 is the rotation point,10 is the center.

☞ Above point 2 and below point 6 in the figure are the horizontal distortion points. The right of point 4 and the left of point 8 are the vertical distortion points.

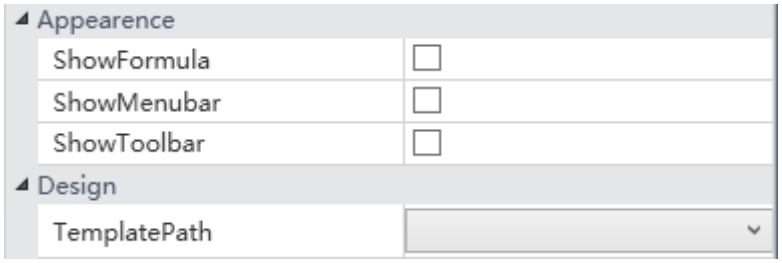
Report properties

✧ ShowFormula: Whether display formula bar

✧ ShowMenubar: Whether display menu bar,check valid when current template is null

✧ ShowToolBar: Whether display toolbar

✧ TemplatePath: Set current report template



✧ Toolbar: set whether to display fast query history data toolbar

✧· For other property settings please refer to the section “7.4 Graphic universal properties”.

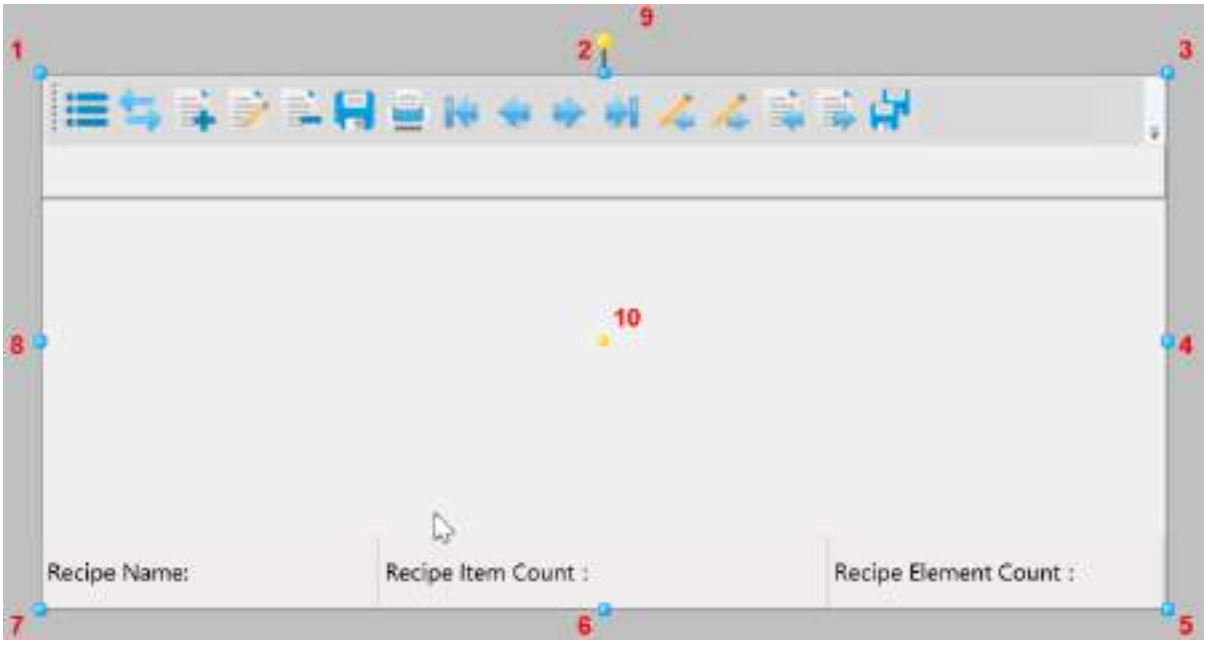
7.7.12 RecipeBrowser

➢· Open the window interface and click on “Toolbox” → “Extended controls” → “Recipe browser” in the tools window to the left; the toolbar is as shown in the figure above.

➢· Move the mouse to the working area of the window, select a starting point and press the left mouse button and drag the mouse towards the bottom-right; after releasing the left mouse button, the drawing of a recipe browser is complete, and a recipe browser will be generated.

➢· Just repeat the steps above if another recipe browser needs to be drawn.

Graphic introduction

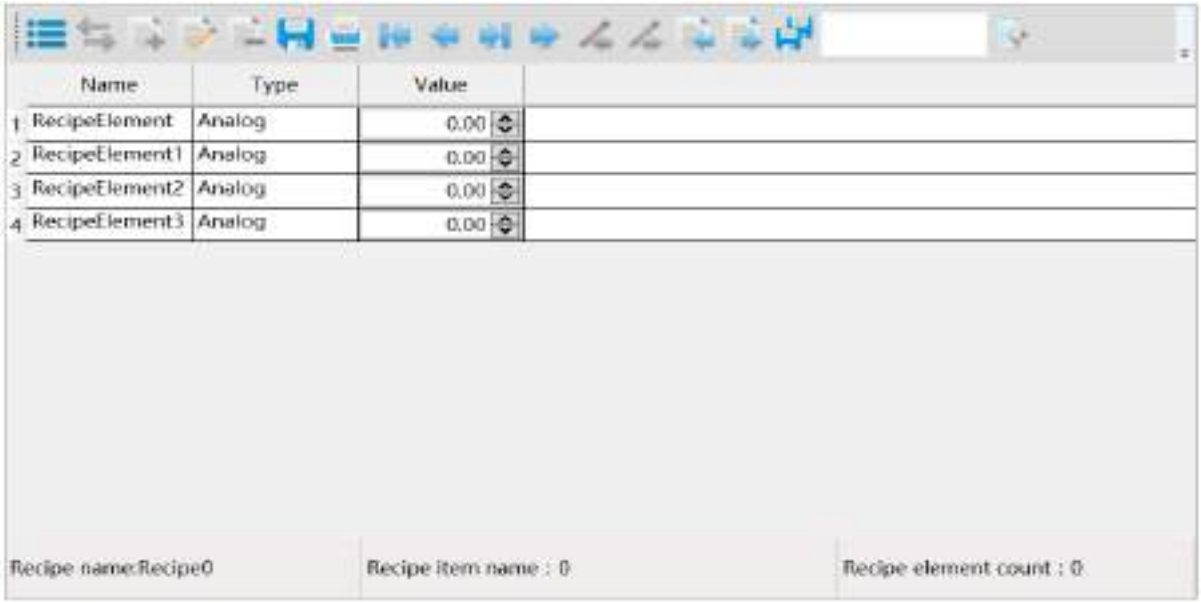


☞ The figure above is a selected recipe browser; click on the recipe browser to enter selected status.

☞ There are two points on 1 in the figure above: the tensile point and the center; 2-8 are the tensile points and 9 is the rotation point,10 is the center.

☞ Above point 2 and below point 6 in the figure are the horizontal distortion points. The right of point 4 and the left of point 8 are the vertical distortion points.

Usage of the toolbar



✧ All|Detail:  display all the detail of recipe

✧ Change:  recipe item and recipe element row-column transform

✧ Add new recipe item:  add new recipe item

✧ Edit:  edit select recipe item

✧ Delete:  delete select recipe item

✧ Save:  record box is saved as picture

✧･ Print:  print current record box

✧･ The first recipe item:  move to the first item

✧･ Previous:  The previous recipe item

✧･ Last:  The last recipe item

✧･ Next:  Next recipe item

✧･ Write in:  write in recipe item from variable value

✧･ Write out:  write out recipe item to variable value

✧･ Import:  import to recipe from file

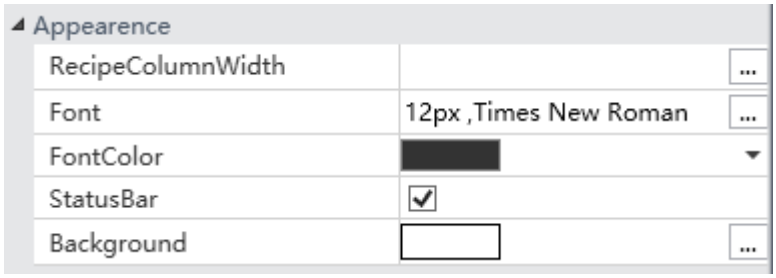
✧･ Export:  export recipe to file

✧･ Save:  save recipe

✧･ Filter input box: filter condition

✧･ Filter:  query recipe according to the content in the filter box

Appearence



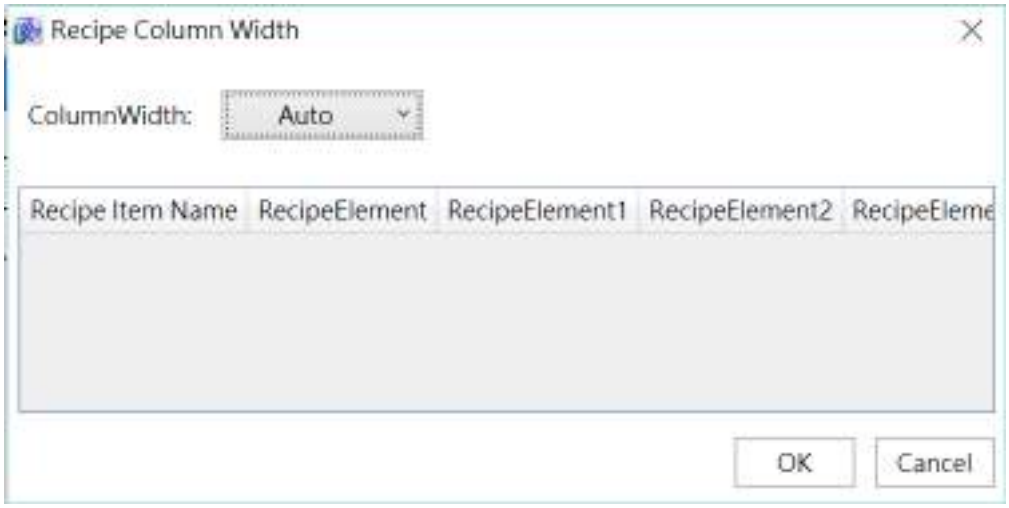
✧･ Font:set recipe browser font

✧･ FontColor:set recipe browser font color

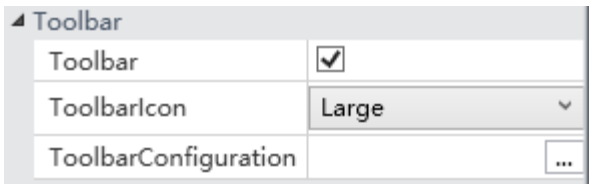
✧･ StatusBar:set StatusBar is show

✧･ Background:set control background

✧･ RecipeColumnWidth:set recipe column width(Auto,Star,Custom)



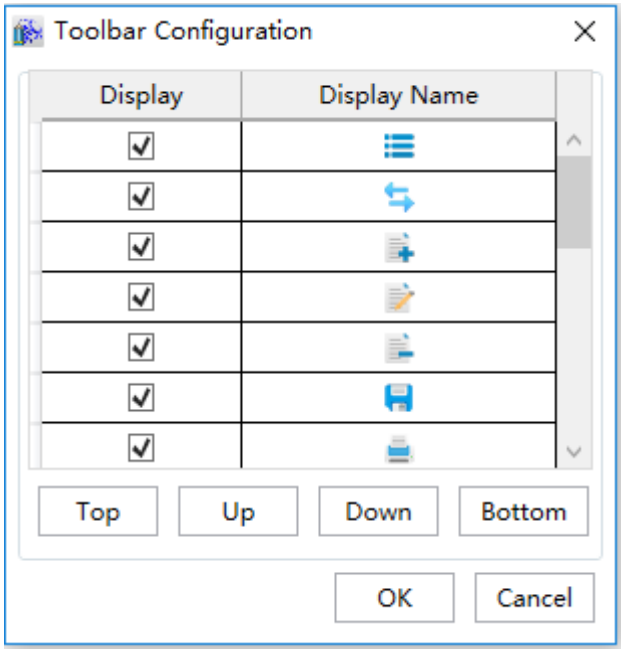
Toolbar



✧･ Toolbar:Set toolbar is show

✧･ ToolbarIcon:Set toolbar icon type

✧･ ToolbarConfiguration:Toolbar configuration



- All Detail: Set whether to display "all detail" button
- Add: Add new recipe item
- Edit: Edit select recipe item
- Delete: Delete select recipe item
- First: The first recipe item
- Previous: The previous recipe item
- Last: The last recipe item
- Next: The next recipe item
- Write in: Write in recipe item from variable value
- Write out: Write out recipe item to variable value
- Import: Import to recipe from file
- Export: Export recipe to file
- For other property settings please refer to the section “7.4 Graphic universal properties”

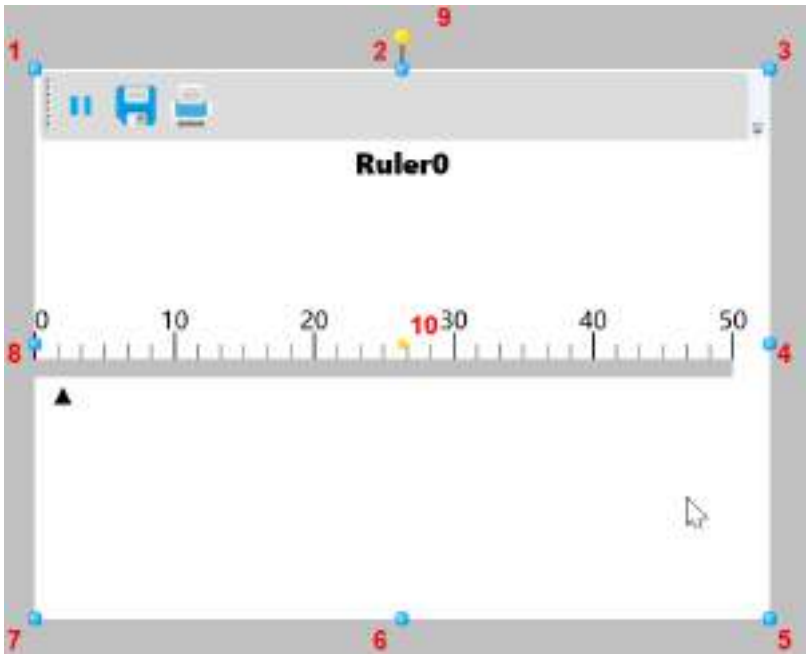
7.7.13 Ruler

➢· Open the window interface and click on “Toolbox” → “Extended controls” → “Ruler” in the tools window to the left; the toolbar is as shown in the figure above.

➢· Move the mouse to the working area of the window, select a starting point and press the left mouse button and drag the mouse towards the bottom-right; after releasing the left mouse button, the drawing of a ruler is complete, and a ruler will be generated.

➢· Just repeat the steps above if another ruler needs to be drawn.

Graphic introduction



☞· The figure above is a selected ruler; click on the ruler to enter selected status.

☞· There are two points on 1 in the figure above: the tensile point and the center; 2-8 are the tensile points and 9 is the rotation point,10 is the center.

☞· Above point 2 and below point 6 in the figure are the horizontal distortion points. The right of point 4 and the left of point 8 are the vertical distortion points.

Ruler properties

✧･ Title: text contents in the title list on top of the ruler; it represents the displayed name of the ruler, and it can be left empty,Support for multilingual functionality.

✧･ TitleFont: Set the font for the title

✧･ TitleColor: Set the color of the title font

✧･ Status coloum: set whether to display status column

✧･ Toolbar: Set whether to display toolbar

✧･ PointerValue: set pointer value

✧･ ValueUnit: set the value of the unit

✧･ VariablePath: Set the VariablePath

✧･ Pointer

Properties	Description
PointerColor	Set the color of the bar pointer
PointerHeight	Set the height of the symbol pointer
PointerWidth	Set the width of the symbol pointer
DisplayPointer	Set whether to display a pointer
PointerPosition	Set the position of the symbol pointer

✧･ Ruler

Properties	Description
FontColor	Set the font color
ScalePosition	Se the scale position
BackgroundColor	Set the background color of the ruler
Maximum	Set the maximum value of the ruler
FontSize	Set font size
MaxScale	Set the large scale of the instrument ruler
MinScale	Set the small scale of the ruler
MinScaleColor	Set the color of the small scale to the ruler
MaxScaleColor	Set the color of the large scale to the ruler

✧･ Bottomline

Properties	Description
BottomlineEndValue	Set end value to the bottom line
BottomlineOffset	Set offset to the bottom line
BottomlineColor	Set color to the bottom line

✧･ For other property settings please refer to the section “7.4 Graphic universal properties”.

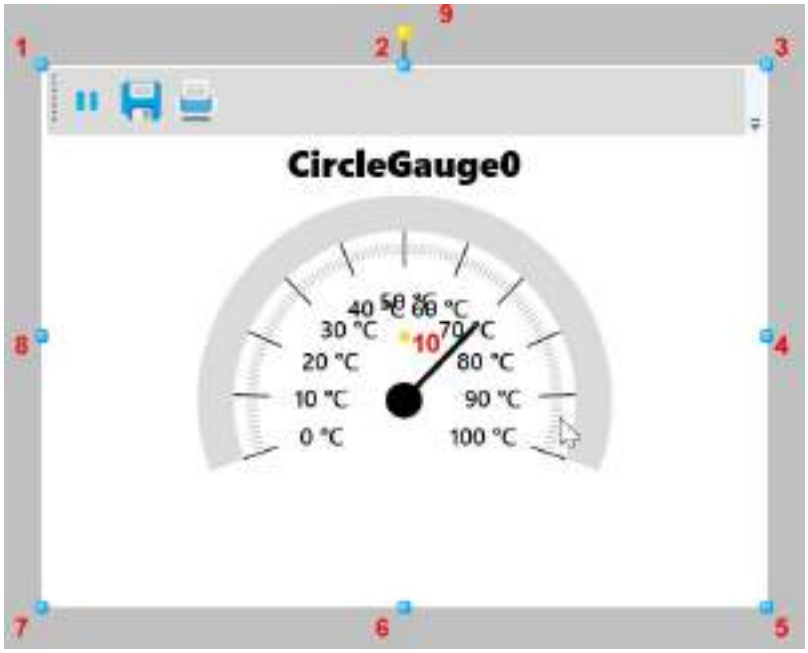
7.7.14 CircleGauge

➢･ Open the window interface and click on “Toolbox” → “Extended controls” → “Circle gauge” in the tools window to the left; the toolbar is as shown in the figure above.

➢･ Move the mouse to the working area of the window, select a starting point and press the left mouse button and drag the mouse towards the bottom-right; after releasing the left mouse button, the drawing of a circle gauge is complete, and a circle gauge will be generated.

➢･ Just repeat the steps above if another circle gauge needs to be drawn.

Graphic introduction



☞ The figure above is a selected circle gauge; click on the circle gauge to enter selected status.

☞ There are two points on 1 in the figure above: the tensile point and the center; 2-8 are the tensile points and 9 is the rotation point,10 is the center.

☞ Above point 2 and below point 6 in the figure are the horizontal distortion points. The right of point 4 and the left of point 8 are the vertical distortion points.

Circle gauge properties

✧ Title: text contents in the title list on top of the ruler; it represents the displayed name of the ruler, and it can be left empty,Support for multilingual functionality.

✧ TitleFont: Set the font for the title

✧ TitleColor: Set the color of the title font

✧ Status coloum: set whether to display status column

✧ Toolbar: Set whether to display toolbar

✧ PointerValue: set pointer value

✧ ValueUnit: set the value of the unit

✧ VariablePath: Set the VariablePath

✧ Pointer

Properties	Description
PointerColor	Set the color of the needle pointer
PointerOriginSize	Set the origin size of the needle pointer
PointerOriginColor	Set the origin color of the needle pointer
Length	Set the length of the needle pointer
Width	Set the width of the needle pointer
PointerStyle	Set the style of the pointer
SymbolPonterStyle	Set the style of the symbol pointer

✧ InstrumentPanel

Properties	Description

BorderBackground	Set the background color of the instrument dial border
FontColor	Set font color
ScalePosition	Set the scale of the instrument panel
BorderWidth	Set the width of the border of the instrument dial border
Maximum	Set the maximum value of the instrument panel
TitleFont	Set the font of the dashboard
MaxScale	Set the large scale of the instrument panel
MinScale	Set the small scale of the instrument panel
Minimum value	Set the minimum value of circle gauge
Display	Set the dashboard display
MinScaleColor	Set the color of the instrument panel
StartAngle	Set the starting angle of the instrument panel
ScanningAngle	Set the scanning angle of the instrument panel
MaxScaleColor	Set the color of the instrument panel
ValuePosition	Set the numeric position on the instrument panel

✧・ For other property settings please refer to the section “7.4 Graphic universal properties”.

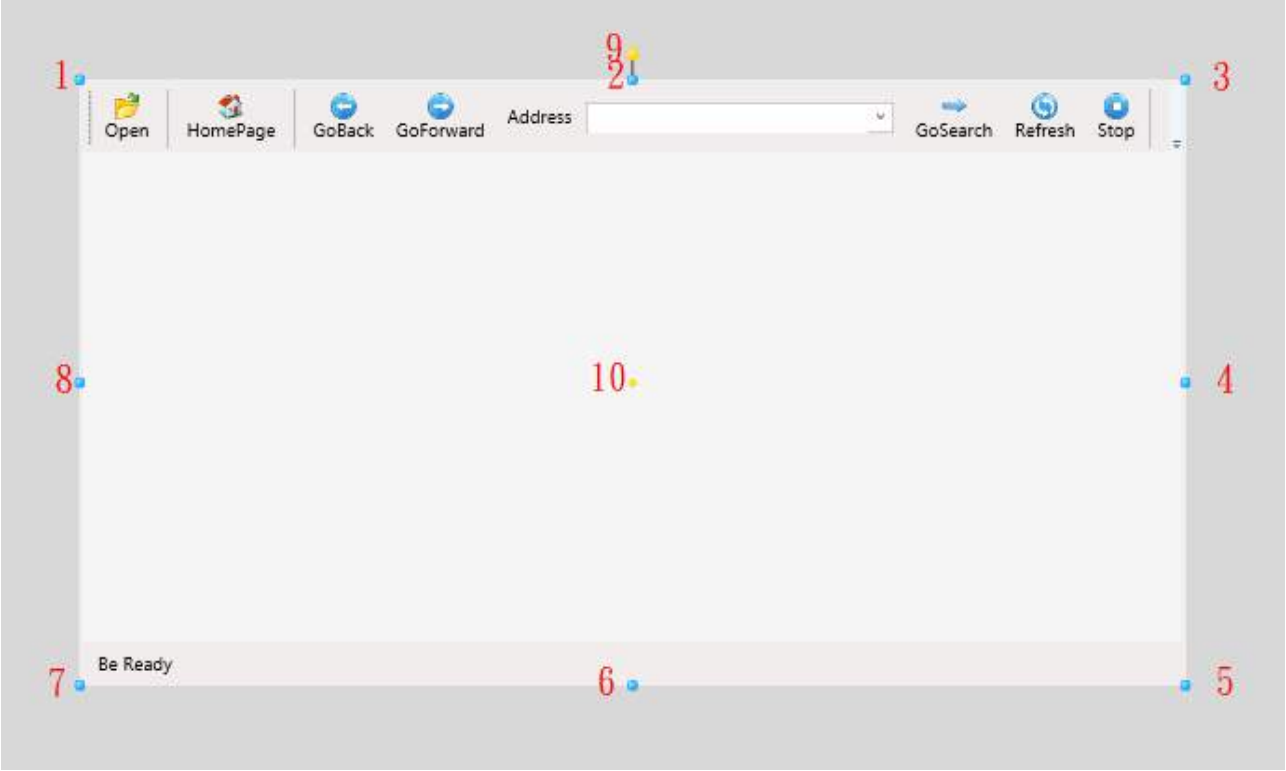
7.7.15 WebBrowser

➢・ Open the window interface and click on “Toolbox” → “Extended controls” → “Web browser” in the tools window to the left; the toolbar is as shown in the figure above.

➢・ Move the mouse to the working area of the window, select a starting point and press the left mouse button and drag the mouse towards the bottom-right; after releasing the left mouse button, the drawing of a Web browser is complete, and a Web browser will be generated.

➢ Just repeat the steps above if another Web browser needs to be drawn.

Graphic introduction



☞ The figure above is a selected Web browser; click on the Web browser to enter selected status.

☞ There are two points on 1 in the figure above: the tensile point and the center; 2-8 are the tensile points and 9 is the rotation point.

☞ Above point 2 and below point 6 in the figure are the horizontal distortion points. The right of point 4 and the left of point 8 are the vertical distortion points,10 is the center.

☞ It is the toolbar on the top of figure which is used to operate web browser.

☞ It is the status bar on the bottom of figure which is used to display the current progress.

Usage of the toolbar

✧ Open: a HTML file can be opened in the Web browser

✧ HomePage: click the button to open the default homepage in the runtime environment

✧ GoBack:backward to previous webpage

✧･ GoForward:forward to next webpagge address

✧･ Address: user can input an address to display

✧･ Refresh: refresh current webpage

✧･ Stop: stop refreshing current webpage

Web browser properties

✧･ StatusBar: set whether to display status bar

✧･ Toolbar: set whether to display toolbar

✧･ Home: set the default address of Web browser

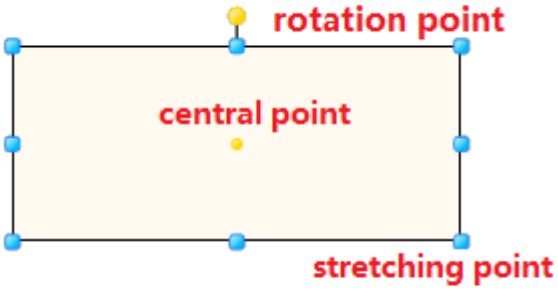
✧･ For other property settings please refer to the section “7.4 Graphic universal properties”

7.8 Graphic operations

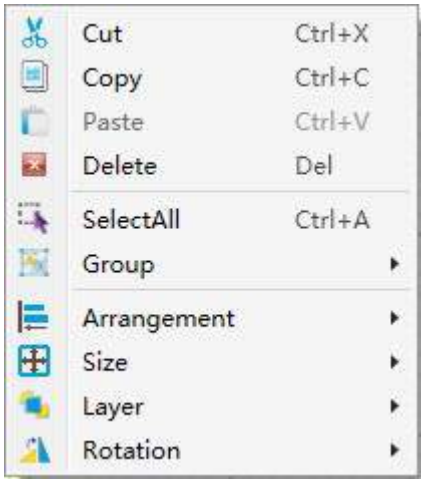
7.8.1 Overview

Graphic operations refer to perform operations to graphics including stretching, rotating,distorting, sorting, aligning and grouping etc., modifying the properties of the graphic to achieve the functions and effects that we need.

In a project development environment, we not only can acquire and modify the properties of graphics directly from the properties frame of the graphic, we can also use the mouse, right-clicking the mouse, shortcut keys and the shortcut buttons on the toolbar to edit the properties. For example, when we are drawing, we can adjust the graphic’s rotation point to rotate it, or stretch the tensile points around the graphic to adjust the size of the graphic etc., as shown in the figure below:



In order to achieve “all-mouse” operations, the DIAView software has the convenient right-click menu (please refer to “3.4 Right-click menu” for the specific functions), as shown in the figure below:



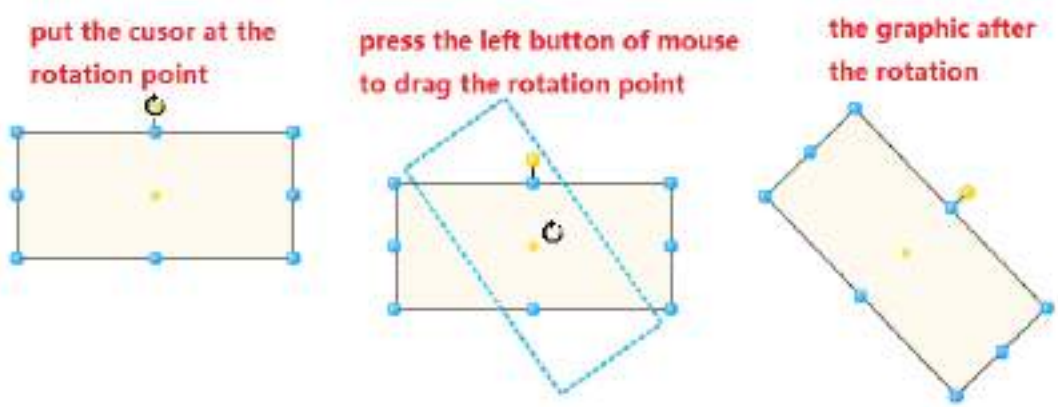
In order to increase development efficiency and the convenience for operating, the DIAView also provides shortcut buttons for frequently used operations (please refer to “3.3.2 Start menu” for the specific functions), as shown in the figure below:

7.8.2 Rotation

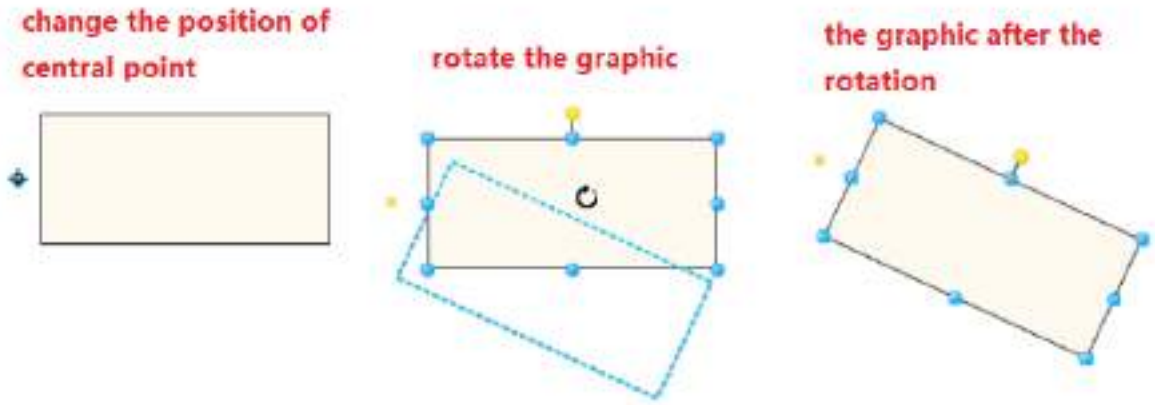
When the graphic drawn needs to be rotated, place the mouse on the “rotation point” and the mouse cursor will change to , now you can press and hold the left mouse button and drag the rotation point to rotate the graphic.

Graphic rotation uses the center point as the center for rotating. It can be rotated using the default center point location (the center of the graphic), or the center point can also be changed and then rotated, as shown in the figure below:

Rotating by the default location of the center point:



Rotating after changing the location of the center point:



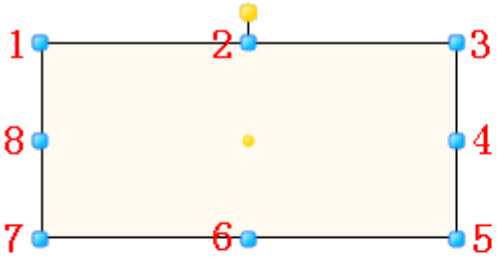
In addition, specific values (unit: degrees) can be entered directly in the "Rotation angle" properties frame of the graphic properties window to specify the rotating angle of the graphic.

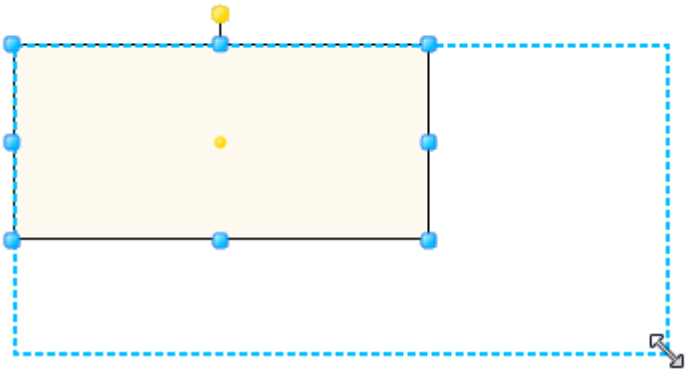
7.8.3 Stretch

When the graphic drawn needs to be stretched (which means adjusting the size of the graphic),

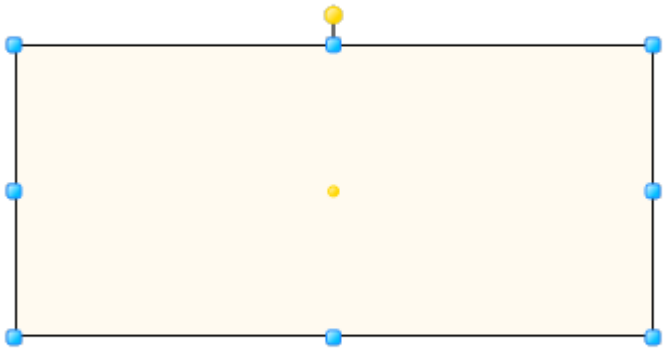
place the mouse on the “stretching point (when a graphic is selected, a dotted frame will appear around the graphic, and 8 rounded rectangle points filled with blue will appear on the rectangular dotted frame. These are the stretching points. For example 1 in the figure below is the first stretching point), and the mouse cursor will change to bidirectional arrows such as  或  . Now you can press and hold the left mouse button to drag the stretching point to change the size of the graphic.

As shown in the figure below:





the cusor status and imaginary line when stretching



the graphic after stretching

In addition, specific values can be entered directly in the "Size" properties frame of the graphic properties window to specify the width and height of the graphic.

7.8.4 Arrangement

When there are multiple graphics, sometimes they need to be aligned or adjusted their spaces according to a specific direction or by referring to one of the graphics. The DIAView software provides 7 types of alignment operations and 2 types of distribution operations(shortcut buttons under the "Format" menu), which most of the alignment methods can only be used when two or more graphics are selected.

Alignment reference graphic rules when multiple graphics are selected: (please refer to the highlighted blue stretching points of the graphics).

1. When multiple graphics are framed with the mouse, the graphic that is first drawn in the sketchpad is the reference graphic;

2. When multiple graphics are selected by clicking with the mouse, the first graphic clicked and selected with the mouse will be used as the reference graphic.

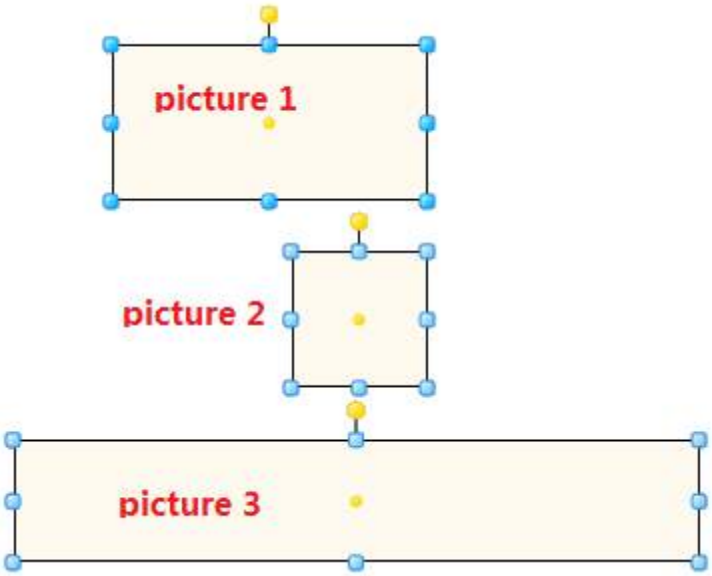
- **Align Left**

  Use the reference graphic as the basis and make the left edge of the other selected graphics align with the left edge of the reference graphic;

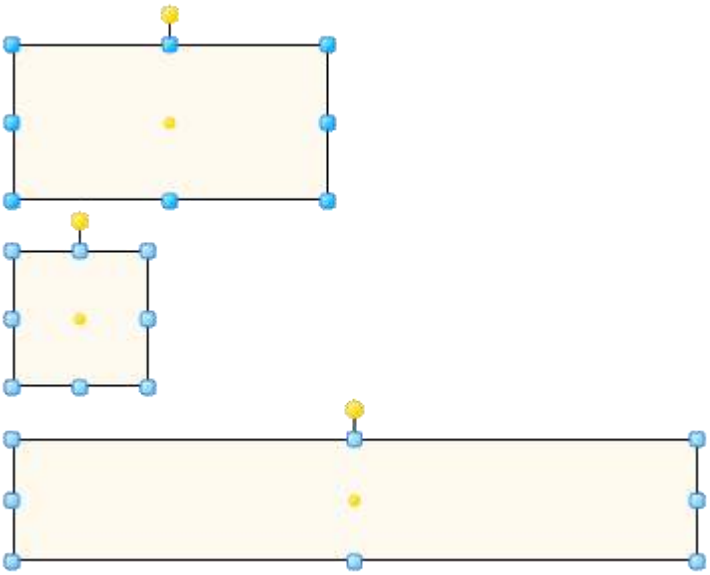
  The shortcut button in the "Start" menu is  .

  For example: 3 graphics are framed in which the rectangular highlighted blue stretch points of graphic 1 is brighter, so it is the reference graphic; when the align left button in the toolbar is pressed, the other graphics will align to the left edge of graphic 1.

  Framed graphics:



Aligned effect:



- **Align Right**

  Use the reference graphic as the basis and make the right edge of the other selected graphics align with the right edge of the reference graphic;
  The shortcut button in the “Start” menu is 

- **Align Top**

  Use the reference graphic as the basis and make the top edge of the other selected graphics align with the top edge of the reference graphic;

The shortcut button in the “Start” menu is 

- **Align Bottom**

  Use the reference graphic as the basis and make the bottom edge of the other selected graphics align with the bottom edge of the reference graphic;
  The shortcut button in the “Start” menu is 

- **Align Center**

  Use the reference graphic as the basis and make the X-axis of geometric center of the other selected graphics the same as the X-axis of the geometric center of the reference graphic;

  The shortcut button in the “Start” menu is 

- **Align Middle**

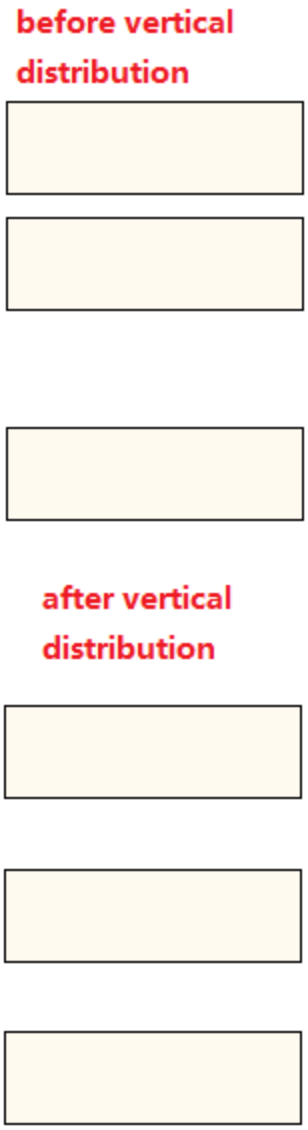
  Use the reference graphic as the basis and make the Y-axis of geometric center of the other selected graphics the same as the Y-axis of the geometric center of the reference graphic;
  The shortcut button in the “Start” menu is 

- **Same center**

  Use the reference graphic as the basis and make the X and Y-axis of geometric center of the other selected graphics the same as the X and Y-axis of the geometric center of the reference graphic;
  The shortcut button in the “Start” menu is 

- **Distribute Vertical**

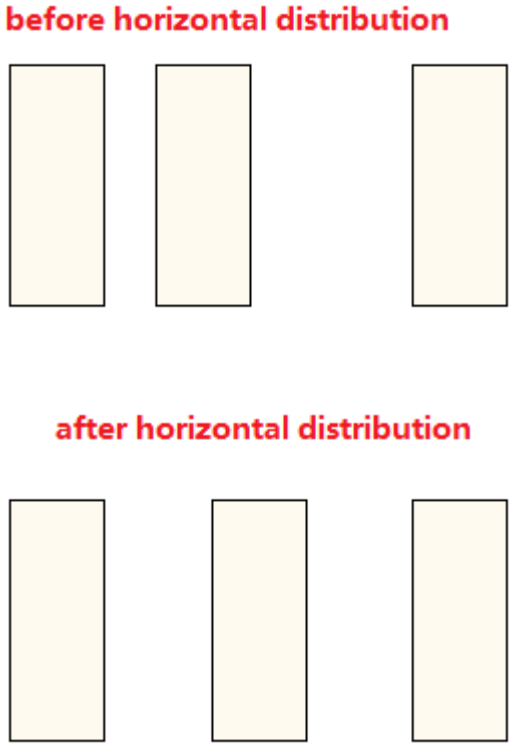
  Distribute the geometric center of three or more selected graphic objects in the sketchpad evenly in the vertical direction,The shortcut button in the “Start” menu is:  ; as shown in the figure below:



➢ **Distribute Horizontal**

Distribute the geometric center of three or more selected graphic objects in the sketchpad evenly in the horizontal direction.

The shortcut button in the “Start” menu is: ; as shown in the figure below:



7.8.5 Layer

Layer refers to the display order of graphics in the sketchpad; the graphic first drawn in the sketchpad of the DIAView is at the inner layer (bottom layer) and the graphics later drawn is at the outer layer (top layer).

When multiple graphics are stacked in the sketchpad, sometimes their stacking order needs to be adjusted; for example placing a certain graphic on the top-most layer for display etc.

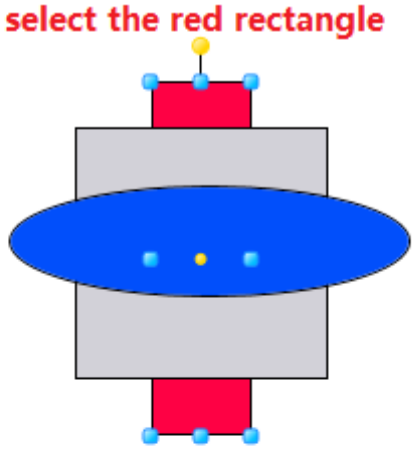
The DIAView has 4 types of graphic layer operation functions (please refer to the shortcut buttons under the “Start” menu):

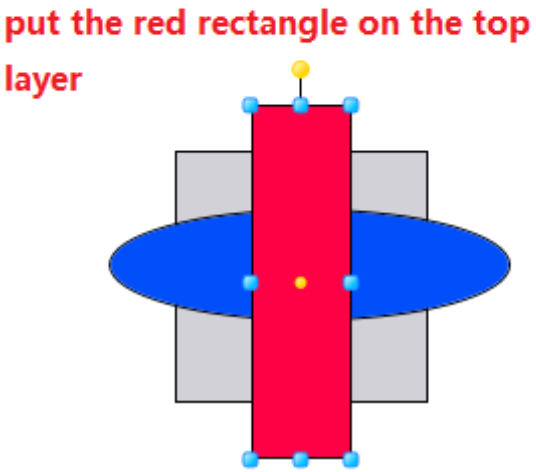
- **Place at top layer**

  Move the selected graphic to the top-most layer of all graphics on the sketchboard for display;

  The shortcut button in the “Start” menu is 

  An example is as follows:





- **Place at bottom layer**

  Move the selected graphic to the bottom-most layer of all graphics on the sketchboard for display;

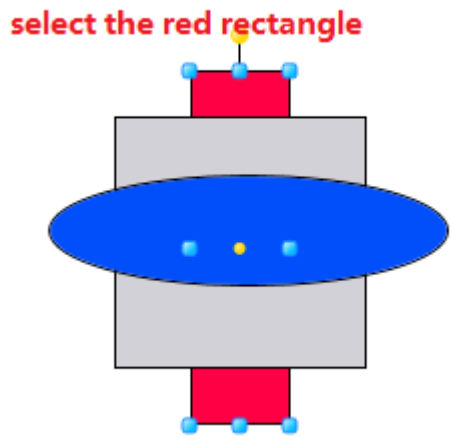
  The shortcut button in the “Start” menu is 

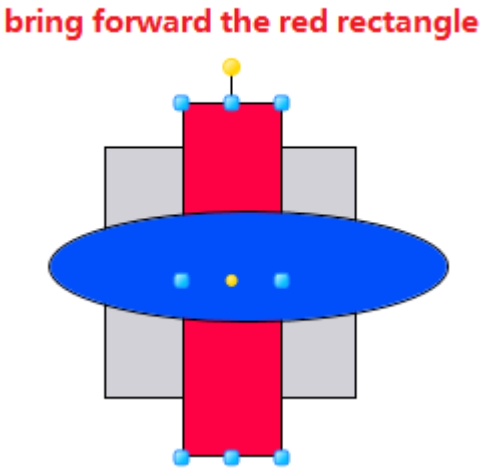
- **Move up a layer**

  Move the selected graphic up a layer for display;

  The shortcut button in the “Start” menu is 

  An example is as follows:





➢ **Move down a layer**

Move the selected graphic down a layer for display;

The shortcut button in the “Start” menu is 

7.8.6 Size

When there are multiple graphics in a sketchpad, sometimes their size needs to be adjusted; the DIAView provides 3 types of graphic size adjustment operations (please refer to the shortcut buttons

under the “Start” menu). They can only be used when two or more graphics are selected.

Alignment reference graphic rules when multiple graphics are selected:

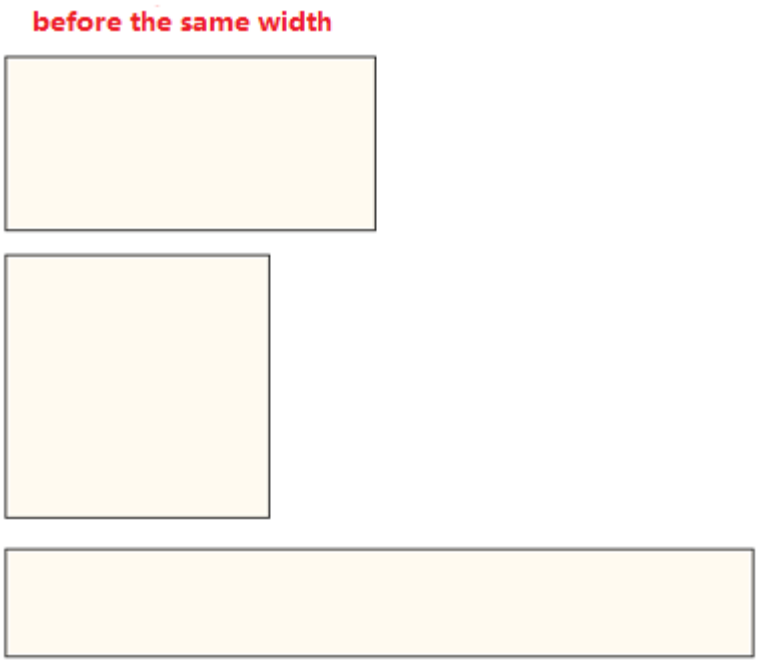
1. When multiple graphics are framed with the mouse, the graphic that is first drawn in the sketchpad is the reference graphic;

2. When multiple graphics are selected by clicking with the mouse, the first graphic clicked and selected with the mouse will be used as the reference graphic.

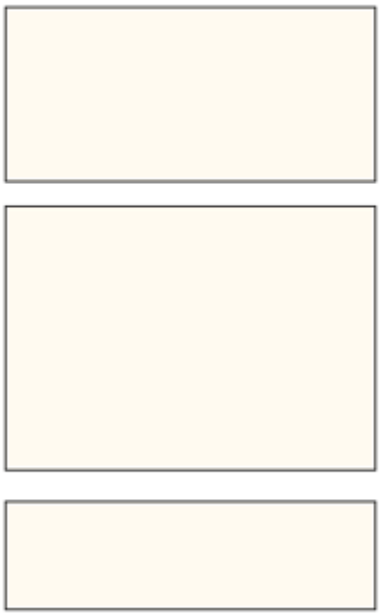
- **Same Width**

  Use the reference graphic as the basis and make the width of the other selected graphics the same as the width of the reference graphic.

  The shortcut button in the “Start” menu is ; as shown in the figure below:



after the same width



- **Same Height**

  Use the reference graphic as the basis and make the height of the other selected graphics the same as the height of the reference graphic.
  The shortcut button in the “Start” menu is .

- **Same Size**

  Use the reference graphic as the basis and make the size of the other selected graphics the same as the size of the reference graphic.
  The shortcut button in the “Start” menu is 。

7.8.7 Grouping

Grouping refers to forming two or more graphics into a whole, forming a new graphic making it easier for user to use. For example: we can group graphics such as fans, motors and water tanks and place it in the graphic library, so we can easily access them from the image library when we need to use it again.

Operations such as rotation and size adjustment etc. can be performed to the grouped graphic as a

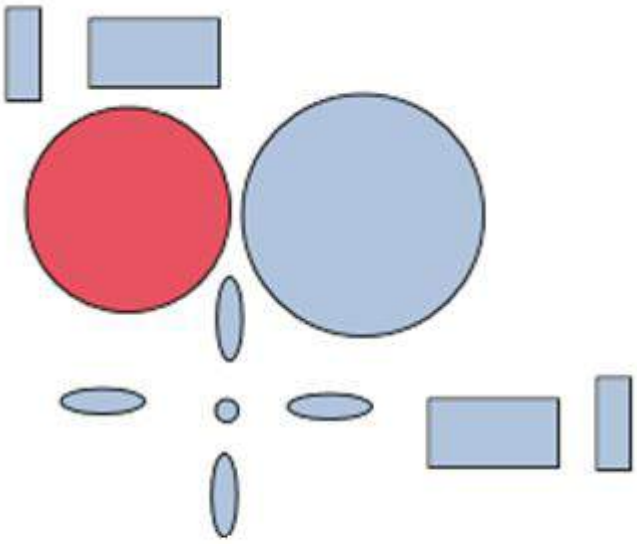
new graphic.

Grouped graphics can be ungrouped at any time.

- **Grouping**

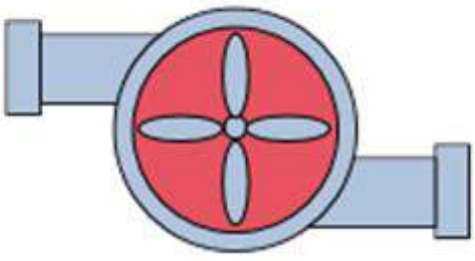
The shortcut button in the “Start” menu is 

First we draw the various sub-graphics that we want to group into a new graphic; for example, we can draw a pump by first drawing the various parts that forms it:



the subgraph of pump

We adjust the sizes, positions and stacking orders of these sub-graphics and then group them together:



the pump after grouping

- **Ungroup**

The shortcut button in the “Start” menu is 

Select the grouped graphic and then click the “Ungroup” shortcut button in the “Start” menu, and the various sub-graphics in the group will exist independently in the sketchpad.

7.8.8 Distortion

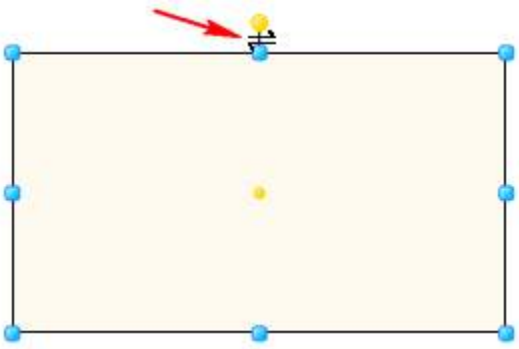
1、Distortion: Making the graphic tilt; distorting is divided into horizontal distortion and vertical distortion.

2、Using a rectangle as an example, it has two horizontal distortion points, when the mouse cursor is placed on the distortion points, the mouse cursor style will change to ⇄; It also has to vertical distortion points, and when the mouse cursor is placed on top of the vertical distortion points, the mouse cursor style will change to ⇅

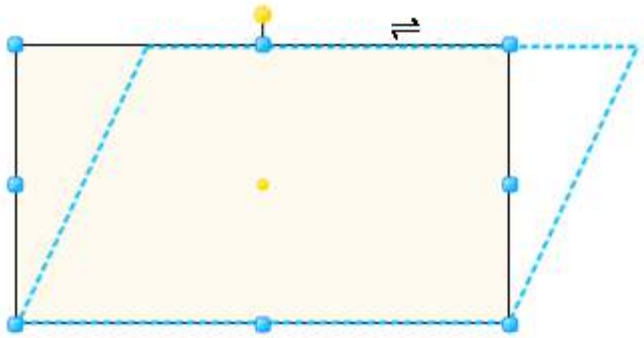
3、When the graphic drawn needs to be distorted, place the mouse on the “distortion point” and the mouse status will change. Now you can press and hold the left mouse button and drag the distortion point to change the shape of the graphic.

As shown in the figure below (taking horizontal distortion as an example):

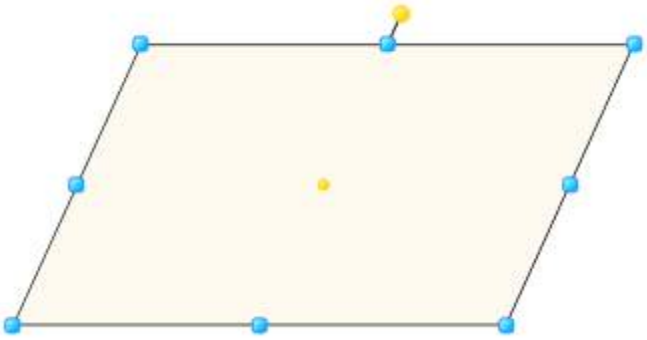
a、Place the mouse on the horizontal distortion point, and the mouse cursor style changes:



b、Press and hold the left mouse button and move it to the right horizontally:



c、Distorted graphic:



In addition, specific values can be entered directly in the "Horizontal distortion" or "Vertical distortion" property frames in the graphic property field to distort the graphic. Both of their value ranges from -89 to 89 up to two decimal places.

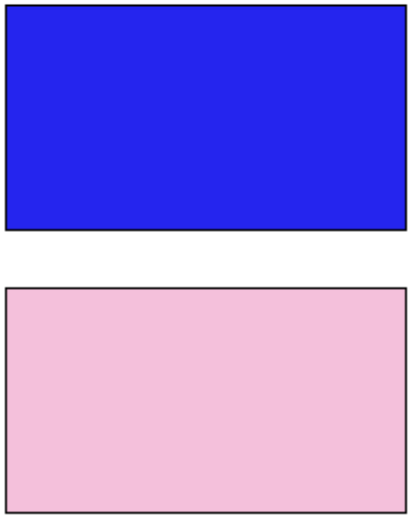
7.8.9 Zooming

Zooming: Zoom in or zoom out the graphic; zooming is divided into horizontal zooming and vertical zooming: horizontal zooming uses the center point of the graphic as the origin and zooms to both the left and right sides. Vertical zooming uses the center point of the graphic as the origin and zooms to both the top and bottom sides. Shifting the center point of the graphic affects the zooming effects.

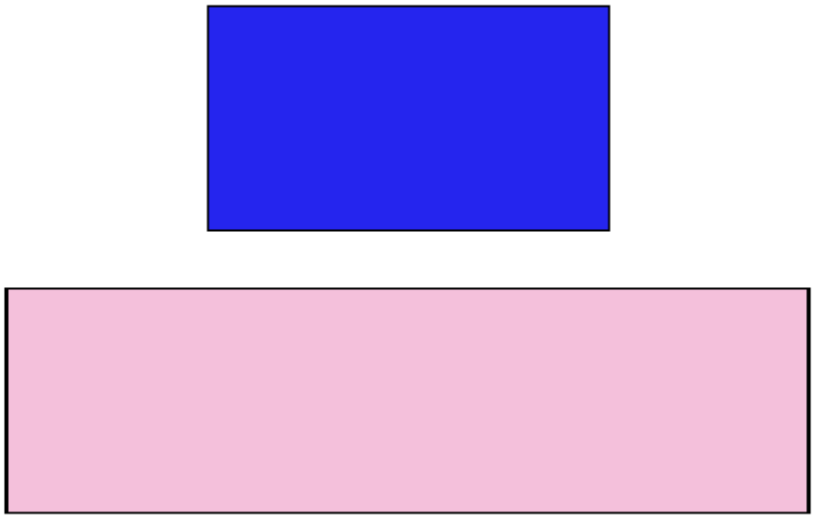
As shown in the figure below:

1. Horizontal zooming: Draw two rectangles with identical sizes and use the default value as the center point; set the horizontal zooming property value of the bottom rectangle to 2.

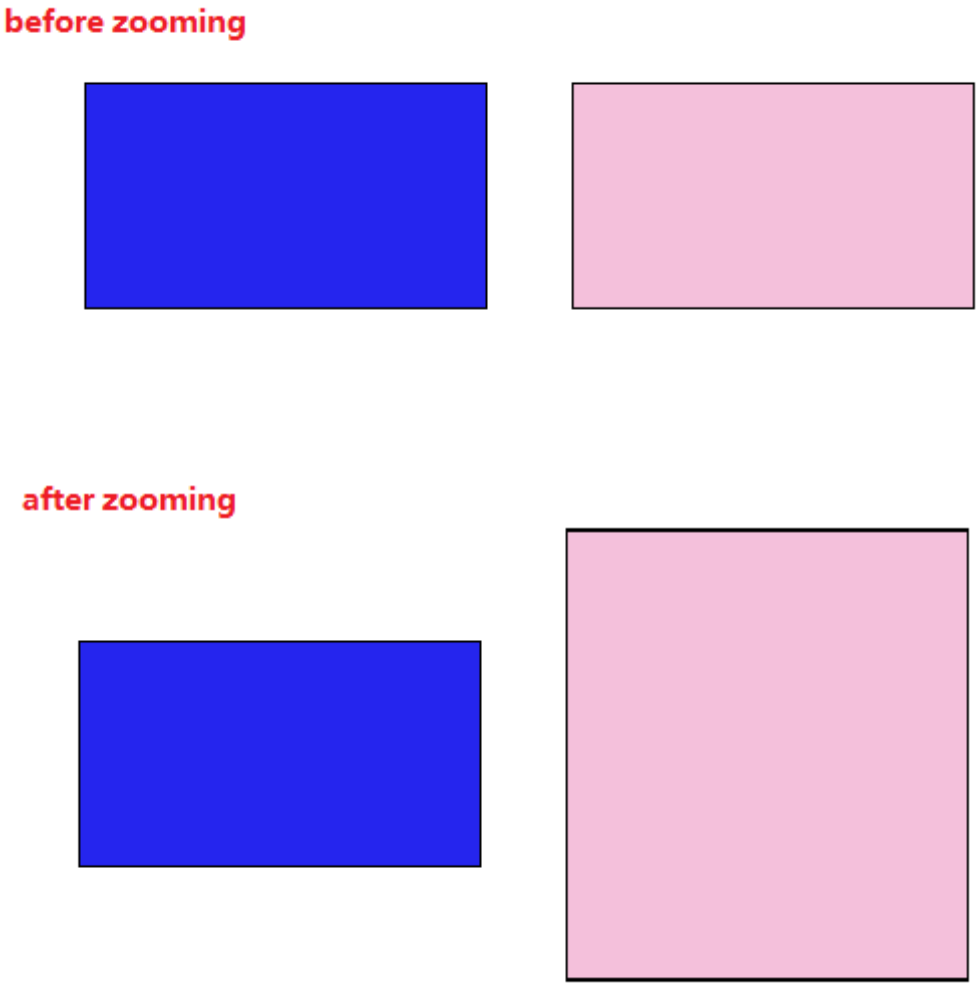
before zooming



after zooming



2. Vertical zooming: Draw two rectangles with identical sizes and use the default value as the center point; set the vertical zooming property value of the right rectangle to 2.



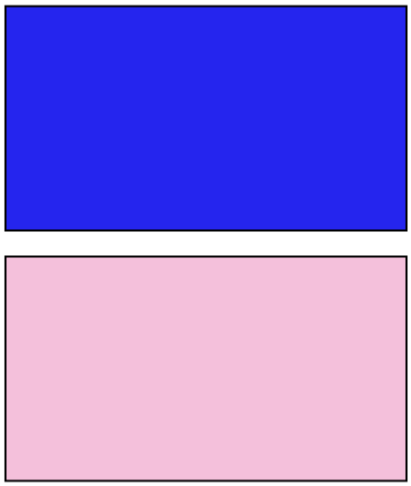
7.8.10 Offset

Offset: Making the position of the graphic offset: Offset is divided into horizontal offset and vertical offset. Horizontal offset uses the coordinates of the graphic as the origin and moves to both the left and right sides; vertical offset uses the coordinates of the graphic as the origin and moves to both the top and bottom sides.

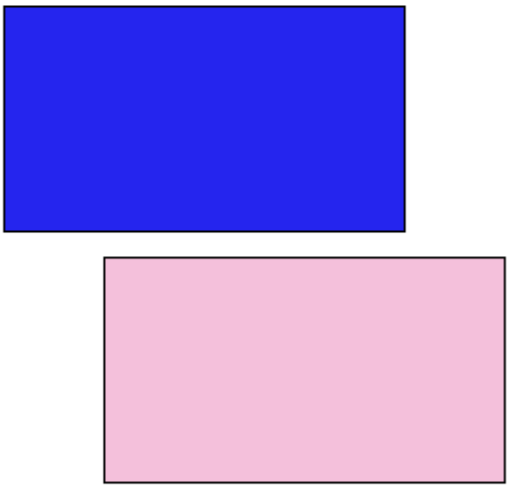
As shown in the figure below:

1、Horizontal offset: Draw two rectangles with identical sizes, set the horizontal offset property value of the bottom rectangle to 50.

before shifting

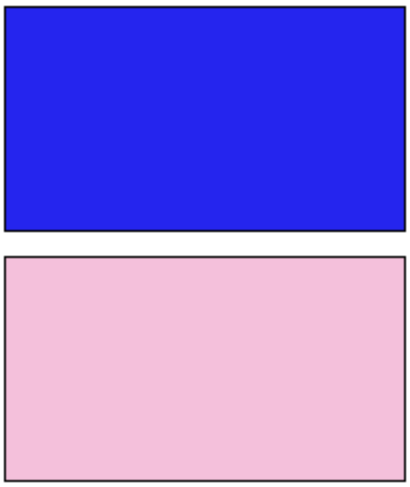


after shifting

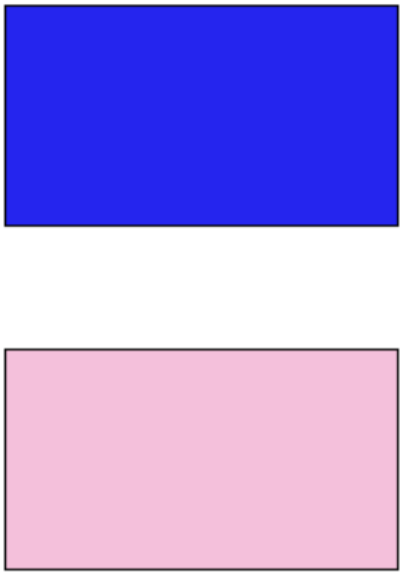


2、Vertical offset: Draw two rectangles with identical sizes, set the vertical offset property value of the bottom rectangle to 50.

before shifting

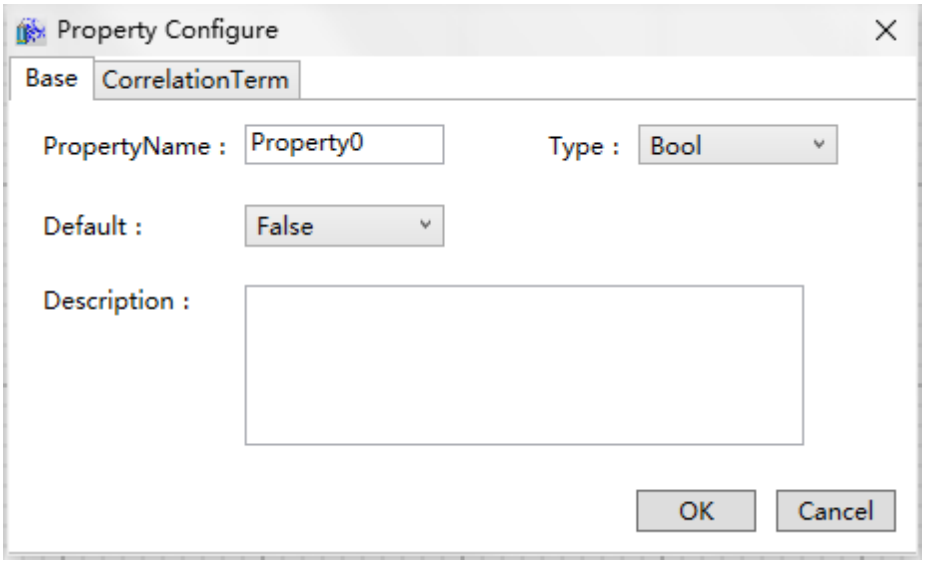


after shifting



7.9 Group graphic extended properties

Click“ ” to add property for grouping graphic in the extended property window and configure the added extended property in the pop-up window.



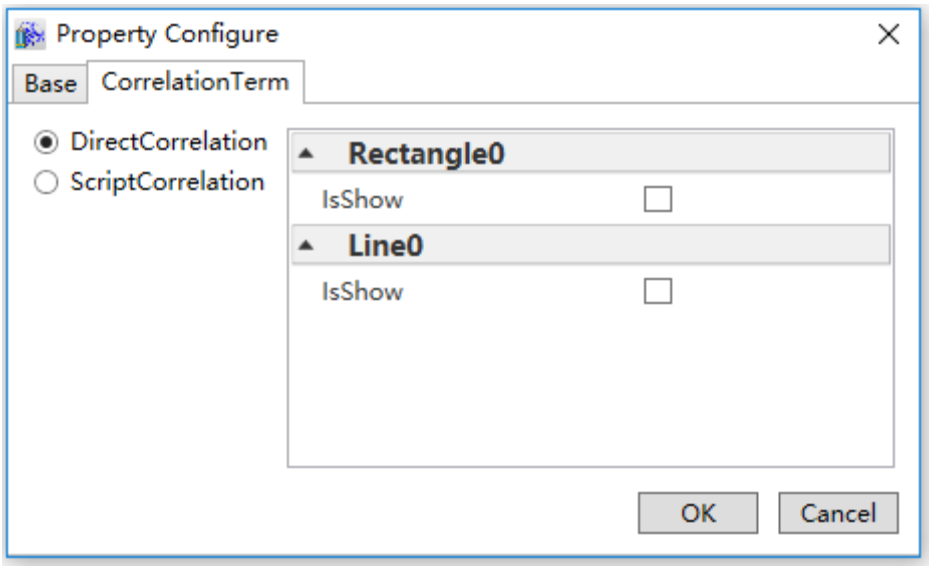
Extended property basic configuration:

Property name: set the name of extended name
Property type: set the type of extended type
Default: set the default of extended property，the value will take effect in the runtime development
Description: set the property description information to explain the property effect or what to do when value changes
Value changed: Click"..." button to edit script, this script will be called automatically when changing the value of extended property value,we can use "value" to replace the extended value which is being configured, such as : Retangle0.Width = value

Add associated item for extended property:

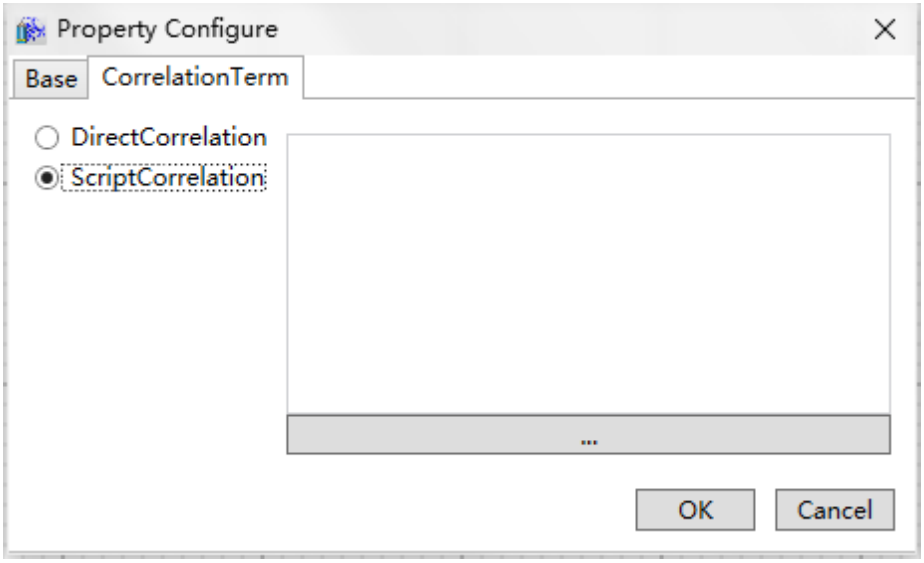
Extended property have two type association:

1.Dierect correlation: property will be listed in the right part of the window, click checking, extended property can be associcated with the sub graphic property directly



2.Script correlation: Click 

button to write script in the pop up window which will be triggered when extended property value changes



8. Gallery

8.1 Overview

Thegallery includes the DIAView software's built-in frequently used graphic model collection and the user's customized graphics collection. Graphics include indicating instruments, button switches, valve and containers and motor pump fans etc.; users can also create their own frequently used graphic models and add them into thegallery.

Models in thegallery are grouped graphics; users can easily add them into the sketchpad and use them directly. It is in the “Gallery” tool window in the toolbox, as shown in the figure below:


Gallery
Motor Pump Fan
Valve Vessel
Indicating Instrument
Control Computer
Universal Tool
Clock
Pointer
Clock0
Clock1
Clock2
DigitalClock
Button Switch
Natural Environment
Shape
Other
Toolbox
Gallery

8.2 Using the gallery

The graphics in the gallery of the DIAView software includes the system's built-in graphic models and also allows users to define their own graphic models and add them into the gallery.

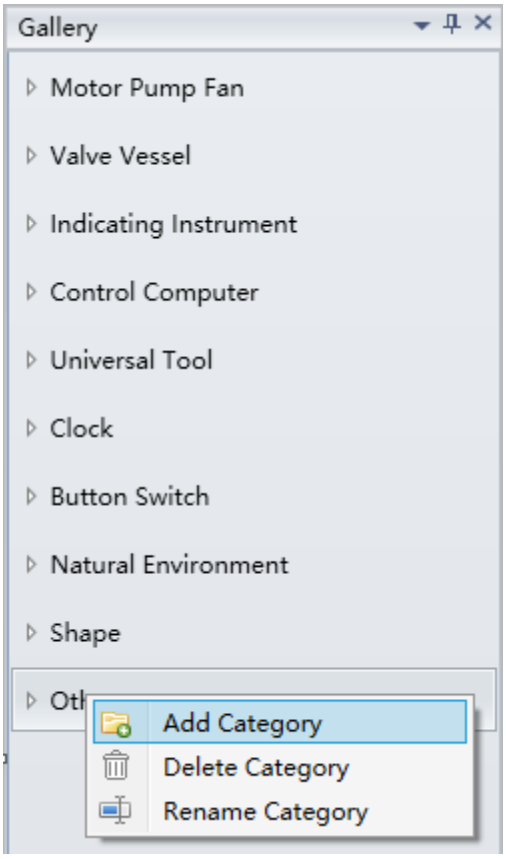
The graphics from the gallery needs to be used in the sketchpad, the method to use them is similar to the normal graphics.

➢ Enter the window interface → open the "Gallery" window in toolbar → select the category of the graphics→ click the model in the gallery with the mouse → move the mouse cursor on the working area of the sketchpad and press the mouse button; then graphics will appear on the sketchpad and its properties such as position and size can be adjusted.

Adding gallery category:

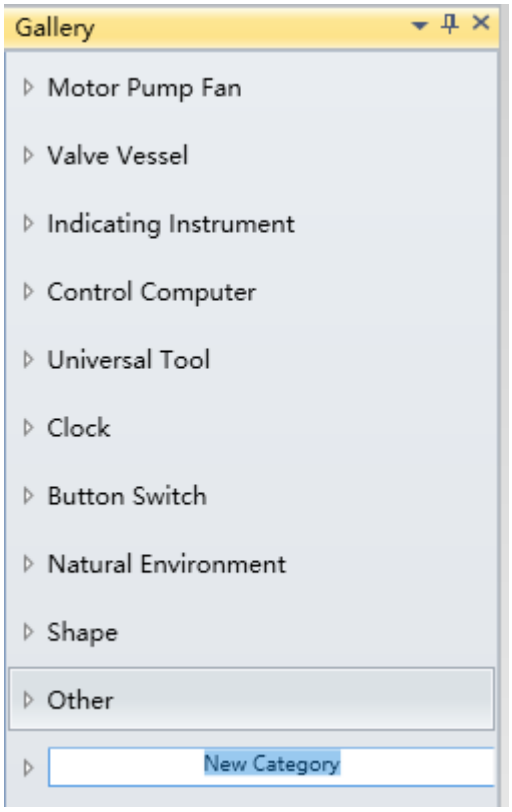
Users can add their frequently used graphic models into the gallery; usually customized gallery categories are created for easy usage and management.

Open the "Gallery" tool window, right-click on the gallery and then select"Category" -- "Add Category", as shown in the figure below:



After clicking "Add Category", a gallery category with the default name "New Category" will be added

in the gallery, as shown in the figure below:



The category name added here can be edited and renamed;

Select a category and perform operations such as “Rename” and “Delete” by right-clicking the mouse.

8.3 Expanding the gallery

The models in the gallery are the grouped graphics in the sketchpad; models can be drawn by the following three steps:

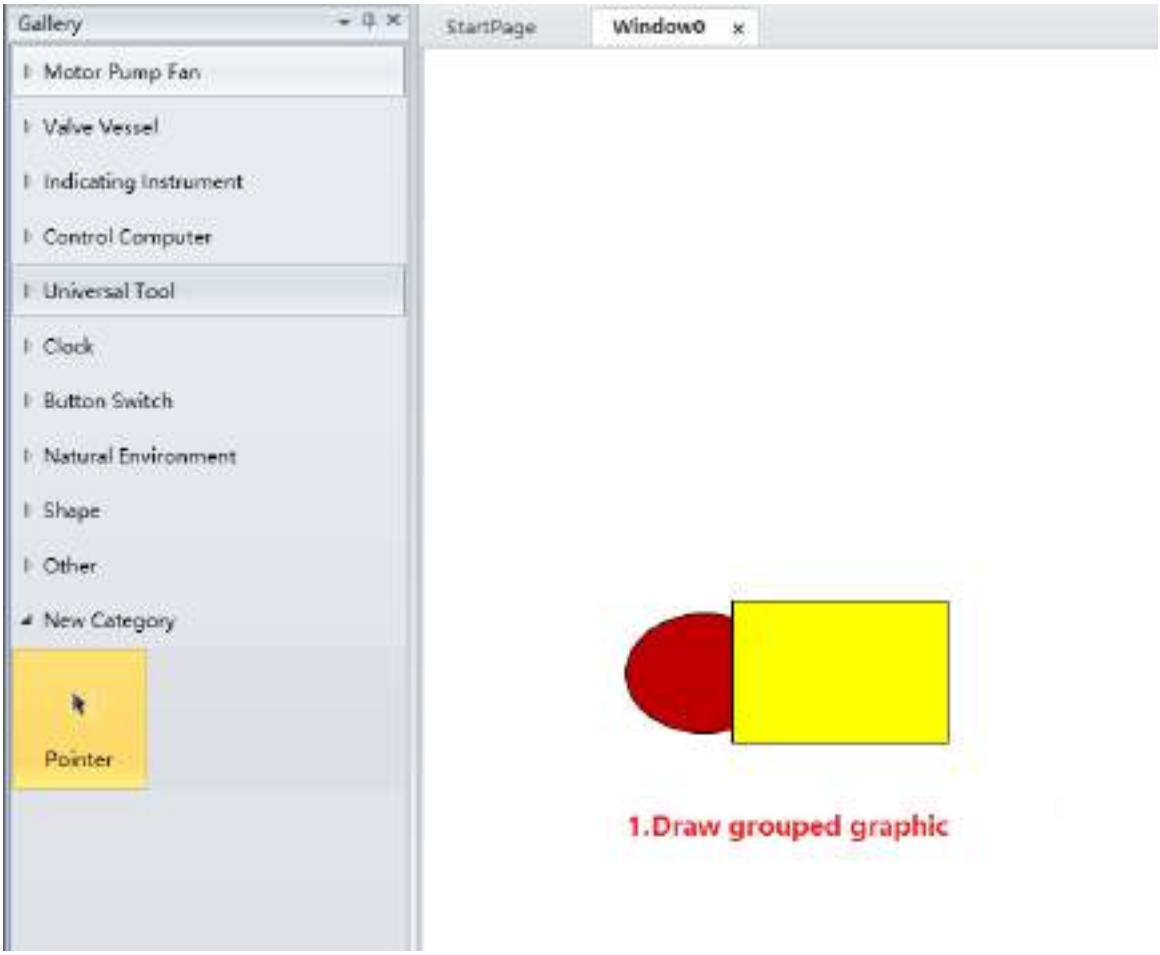
1. Draw the sub-graphics that forms the model.

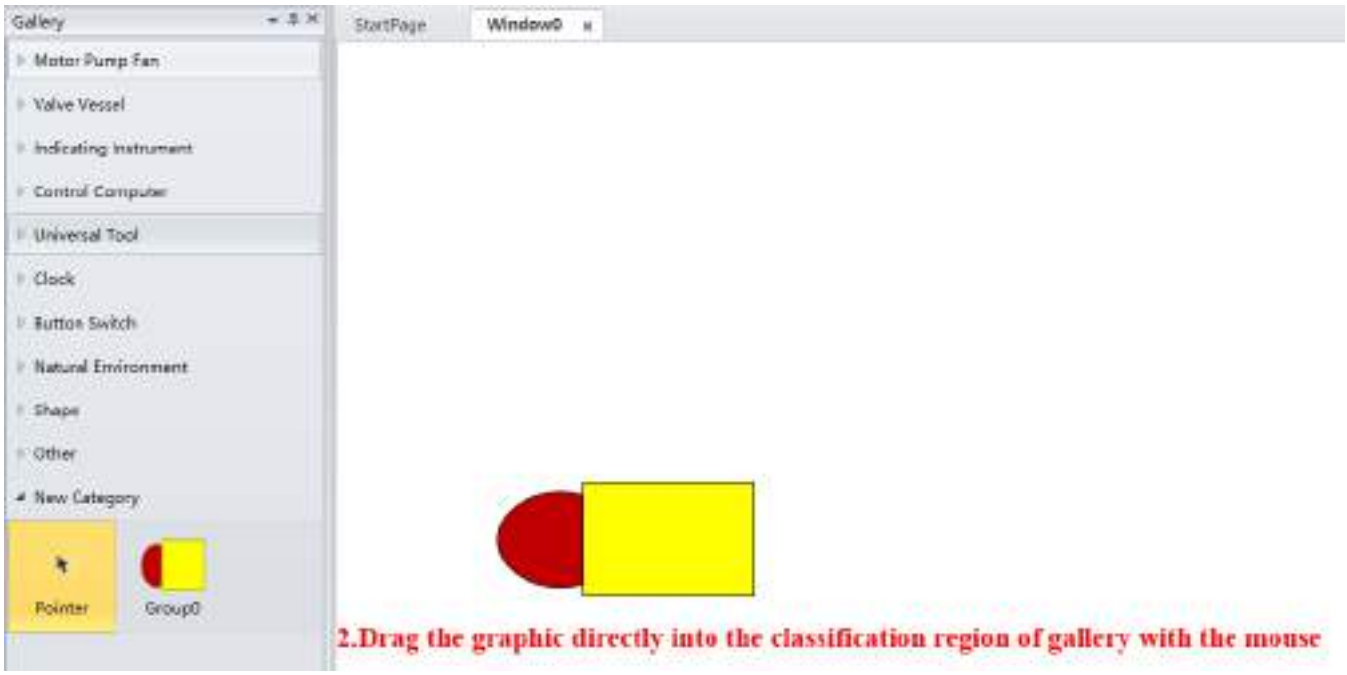
2. Adjust the size, position and stacking levels of the sub-graphics and then group them into a graphic.

3. Add the new grouped graphic into the gallery.

Adding the customized model into the gallery:

Open the category inside the “Gallery” to add the new model, select the grouped graphics in the sketchpad, press and hold the left mouse button and drag the grouped graphics into the category area of the “Gallery”; the steps are as shown in the figure below:





At this time the model name is automatically generated by the system; users can select the model and press the right mouse button to perform operations such as “Rename” and “Delete” etc.

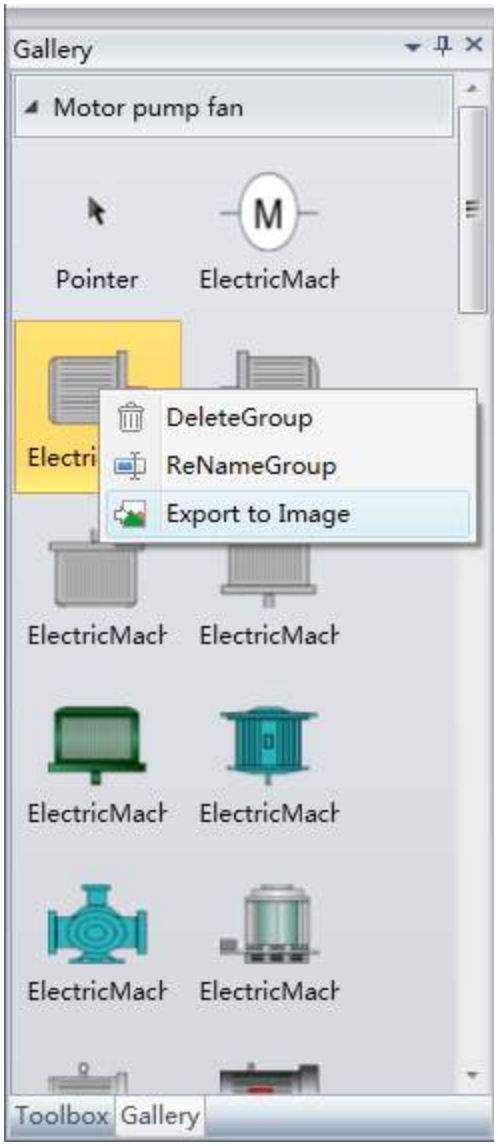
8.4 Export as picture

In this panel,the steps below can export the model in gallery as picture:

1.Right-click the graphic model that needs to export

2.Select "Export as picture" item in the right-click buttton

3.Select the flie path to save，confirm exporting



9. Animation

9.1 Overview

Animation refers to the correspondence between the created window element and the internal variables; it makes the properties of the window element change dynamically according to the field acquired data, and truly displays the dynamic scene of the industrial field. For example, simulation of fluid flow in the pipeline in the window, real-time changes of instrument data, rotation of electrical machineries and the flashing of alarm etc.

To configure an animation that makes the related properties of the graphics object in the window change in real-time, corresponding variables must be connected and related properties must be

configured according to the variable data acquired in real-time, therefore animations are driven by data.

The animation configuration function of the DIAView is very powerful; it includes animations such as rotating, appearance, filling, zooming, moving, visibility, flowing and value display etc. The corresponding animation can only be configured once for the same graphic object.

DIAView animation refresh rate defaults to 60 frames per second, depending on the use environment.

9.2 Rotating animation

Rotating animation refers to allow graphics to perform rotating around the center point of the graphic and changing the “rotation angle” property of the graphic.

Rotating animations are divided into four kinds: “RotationControl”, “RotationSpeed”, “RotationAngle” and “DiscreteRotate”;

“RotationAngle” animation refers to an animation where the rotation angle of the graphic object make linear relationship changes according to the variables or the value of the expression;

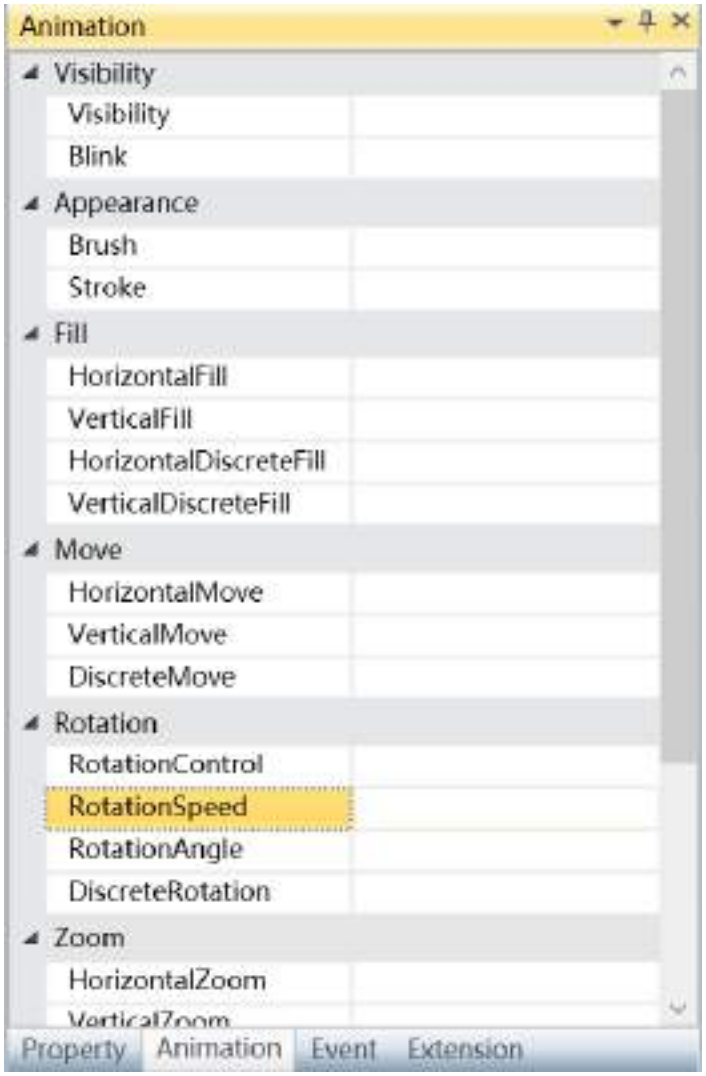
“RotationControl” refers to an animation where whether the graphic object will rotate around the center point of the graphic is controlled by the variable or when the value of the expression is true or false.

“DiscreteRotate” refers to the rotation angle of the graphic object being discretely related to the variable;

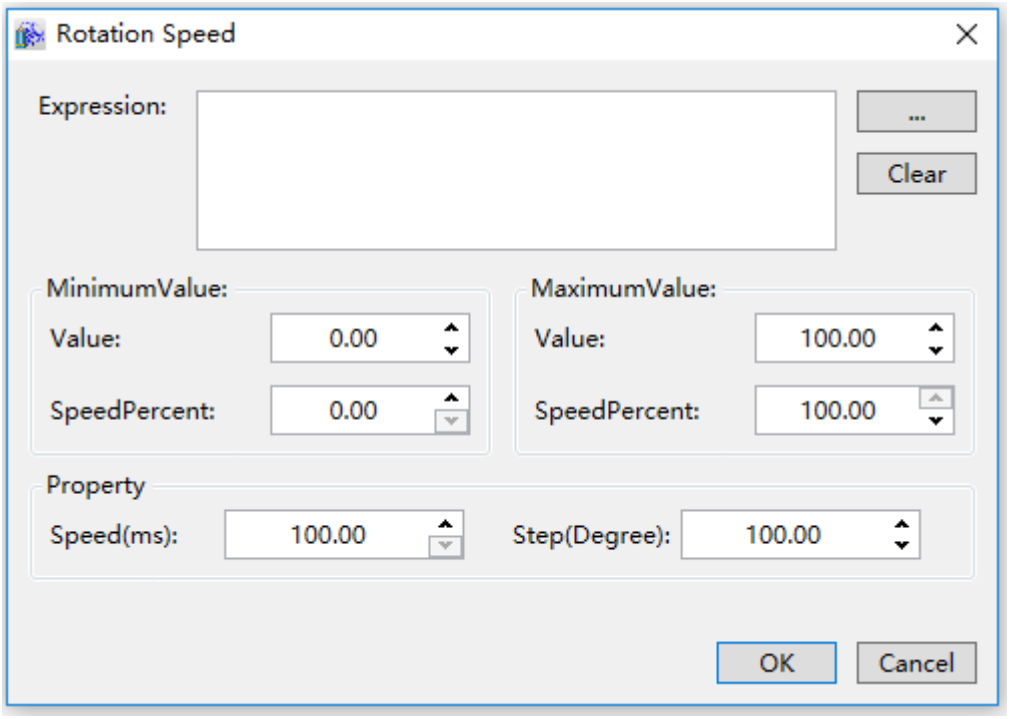
“RotationSpeed” refers to the rotation speed of the graphic object being bound with the variable.

1.Steps to configure rotation speed are as follows:

Step 1: Open the project window where the animation needs to be configured under the DIAView software development environment → select the graphic object to configure animation in the sketchpad → open the “Animation” window → select “RotationSpeed”→ click the button in the “RotationSpeed” field, as shown in the figure below:



Step 2: The rotation speed animation configuration window will appear, as shown in the figure below:



The meanings of each setting in the configuration window are as follows:

Expression: Enter the variable name or expression; you can also click the  button to open the variable browser to select a variable; the “Clear” button can clear the contents in the input frame.

MinimumValue:

- **Value:** Set a minimun value for expression
- **SpeedPercent:** Set a minimun value for speed percent

MaximumValue:

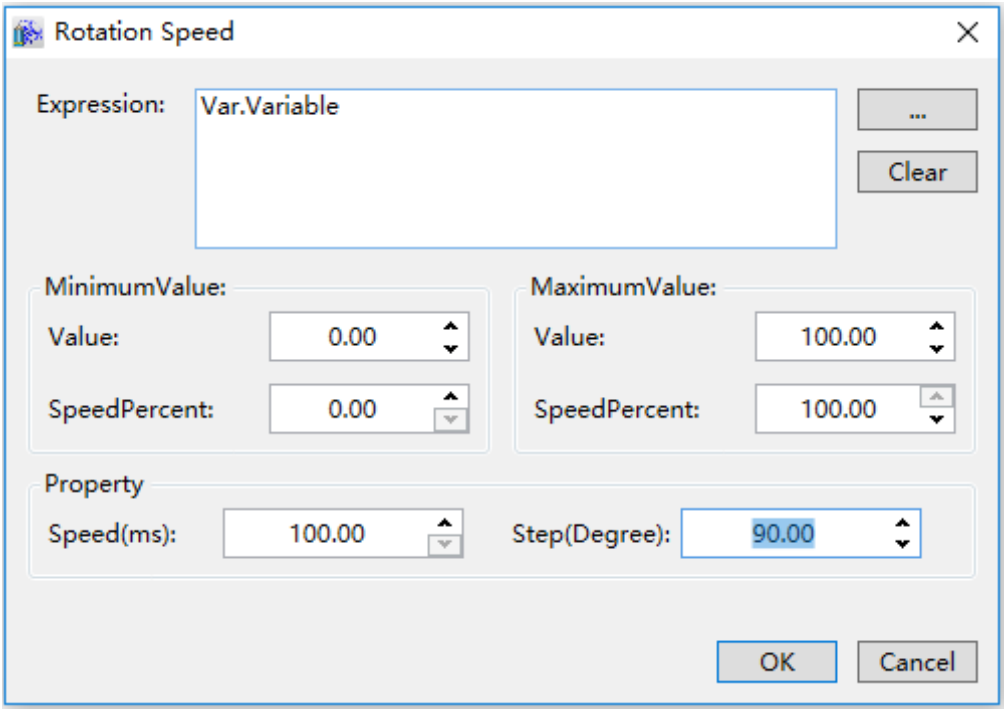
- **Value:** Set a maximum value for expression
- **SpeedPercent:** Set a maximum value for speed percent

Property:

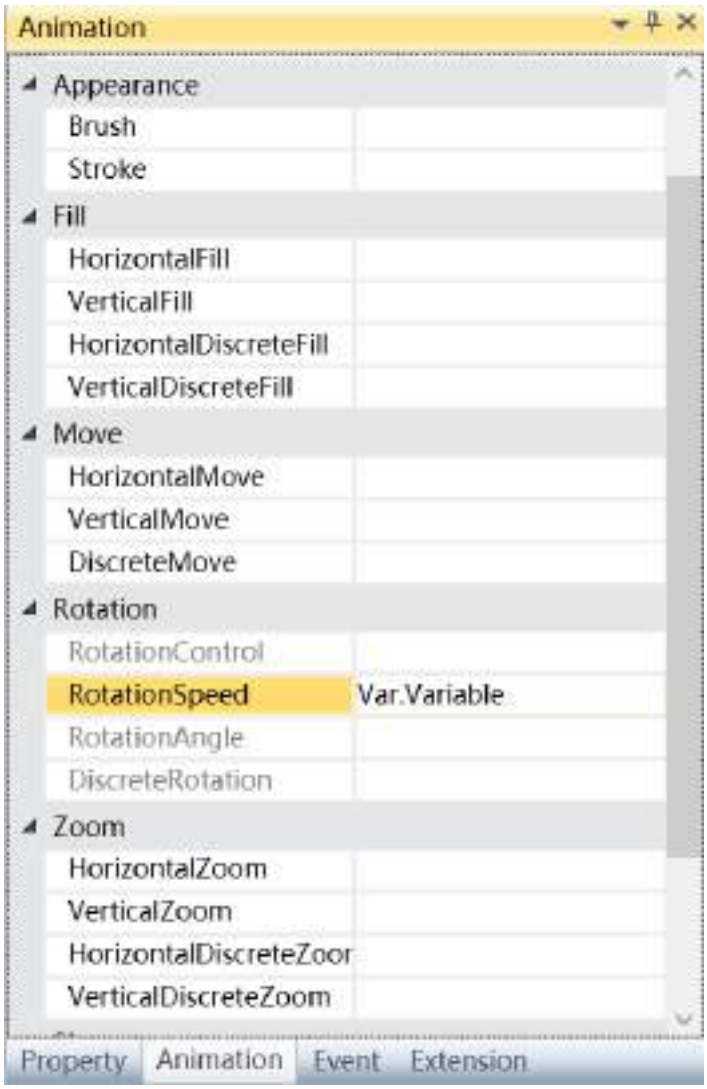
- **Step(Degree):** Set the angle at which the object rotates one step(Unit:Degree)
- **Speed(ms):** Set the time required for each rotation step of the object(Unit:ms)

Example: Configure the window as shown in the figure below:When the expression value is 10,the

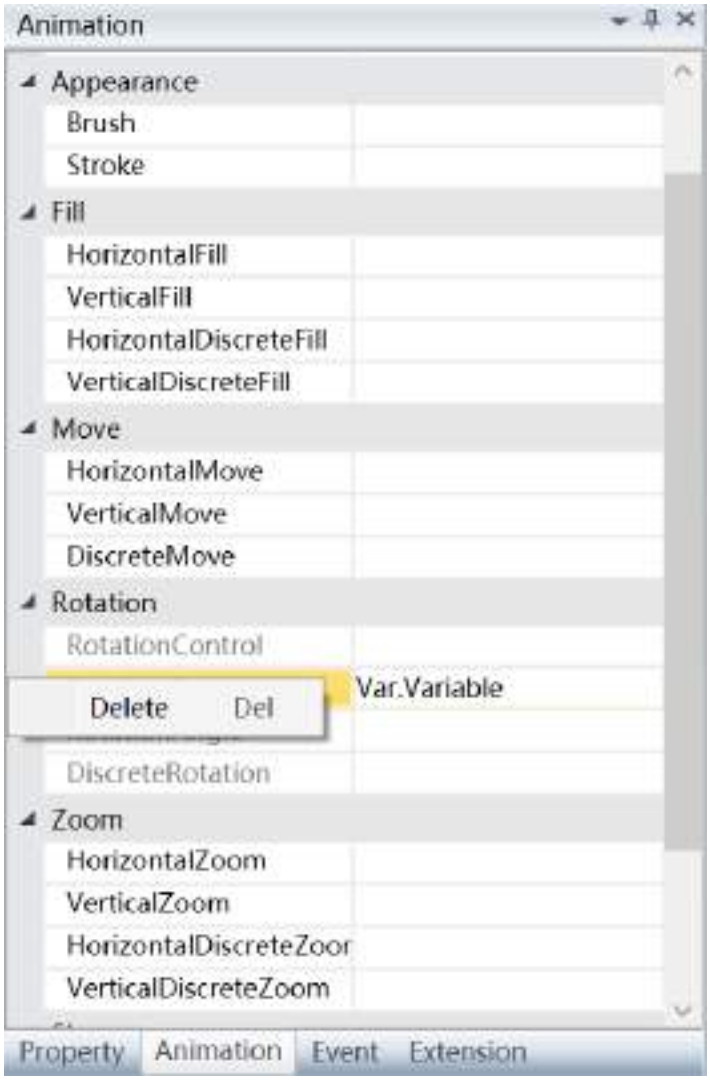
object rotates 9 degrees per second;when the expression value is 100,the object rotates 90 degrees per second.



Step 3: When configuration is complete, press the “OK” button to complete the configuration of the “RotationSpeed” animation; the animation configuration of this graphics will be displayed in the “Animation” window, as shown in the figure below:

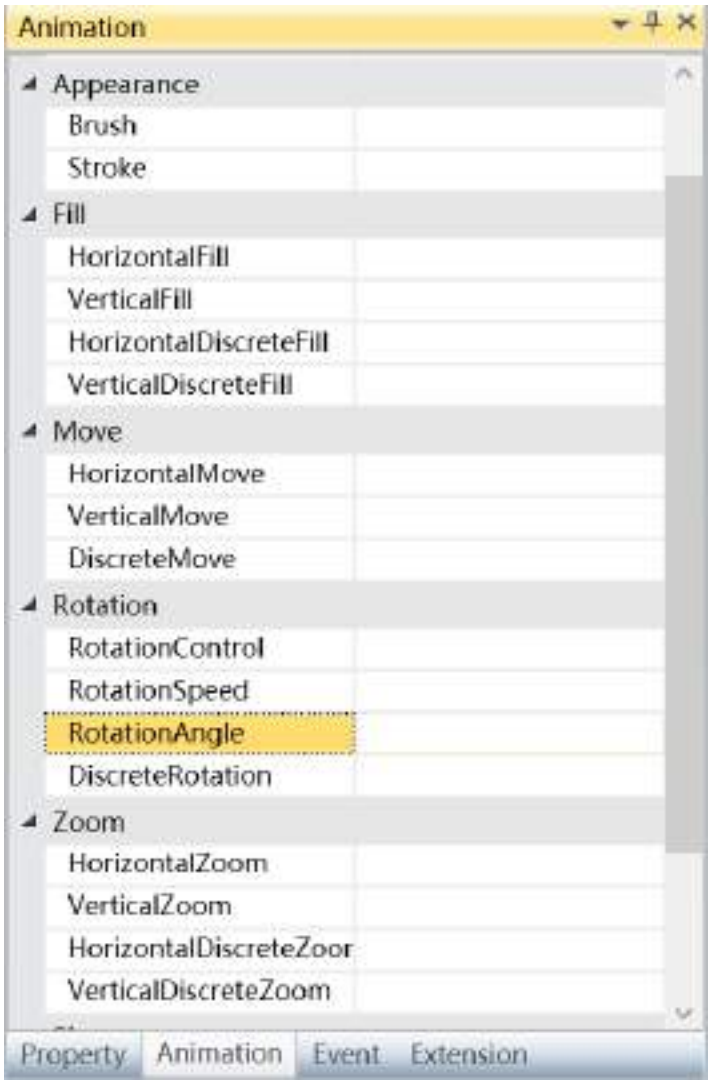


To delete the animation, simply select the animation in the animation list and then right click the mouse and select “Delete”, as shown in the figure below:

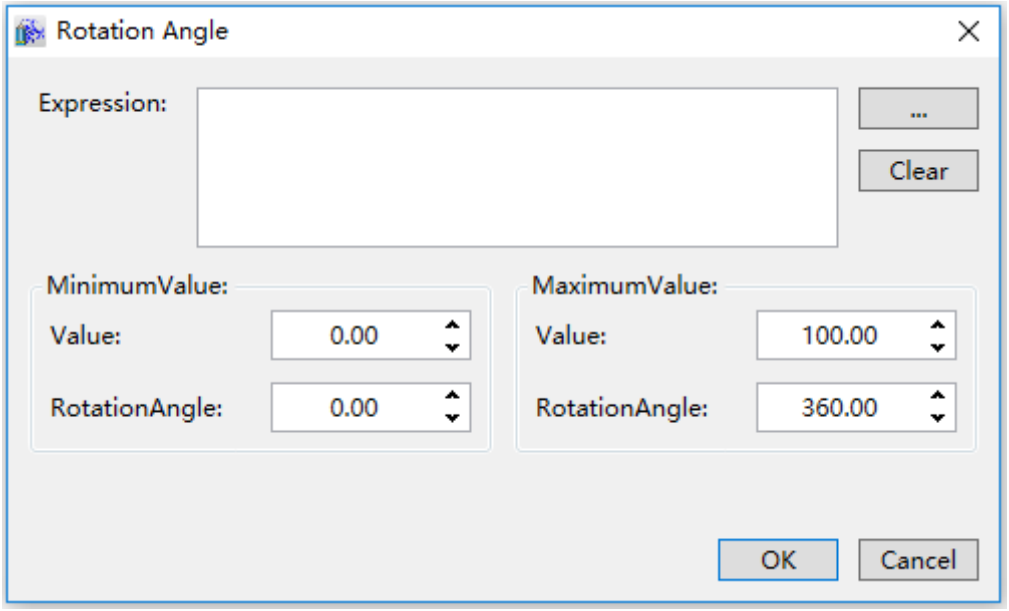


2. Steps to configure rotation angle are as follows:

Step 1: Open the project window where the animation needs to be configured under the DIAView development environment → select the graphic object to configure animation in the sketchpad → open the “Animation” window → select “RotationAngle” → click the button in the “RotationAngle” bar , as shown in the figure below:



Step 2: The rotation angle animation configuration window will appear, as shown in the figure below:



The meanings of each setting in the configuration window are as follows:

Expression: Enter the variable name or expression; you can also click the  button to open the variable browser to select a variable, the “Clear” button can clear the contents in the input box.

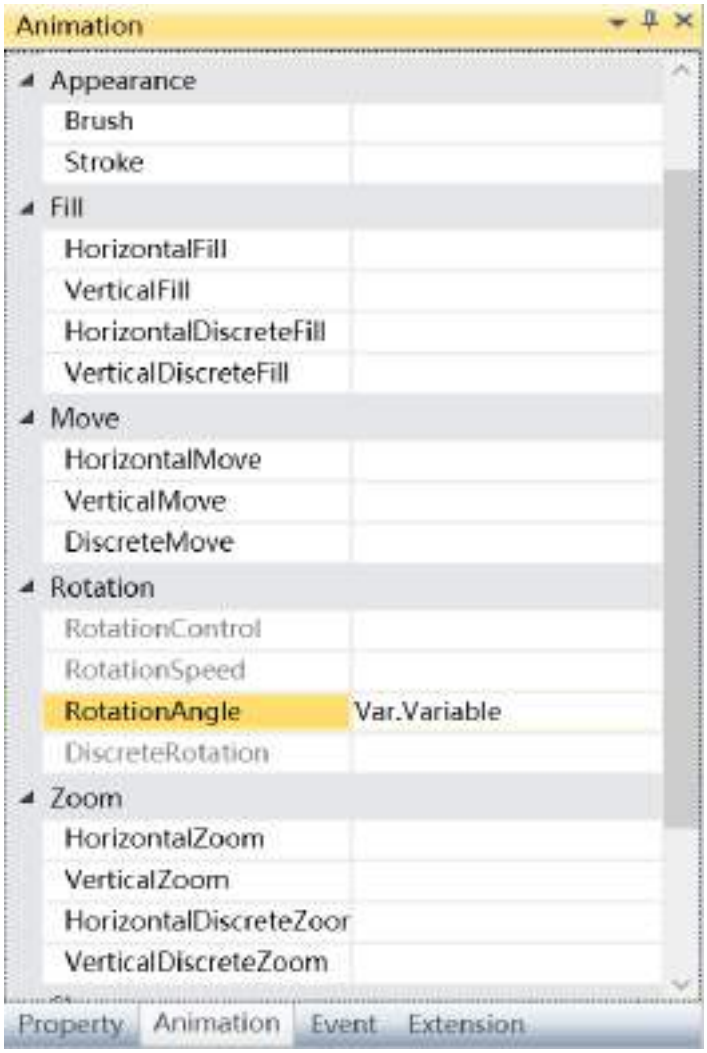
MinimumValue:

- **Value:** Set a minimun value for expression
- **RotationAngle:** Set a minimun value for rotation angle

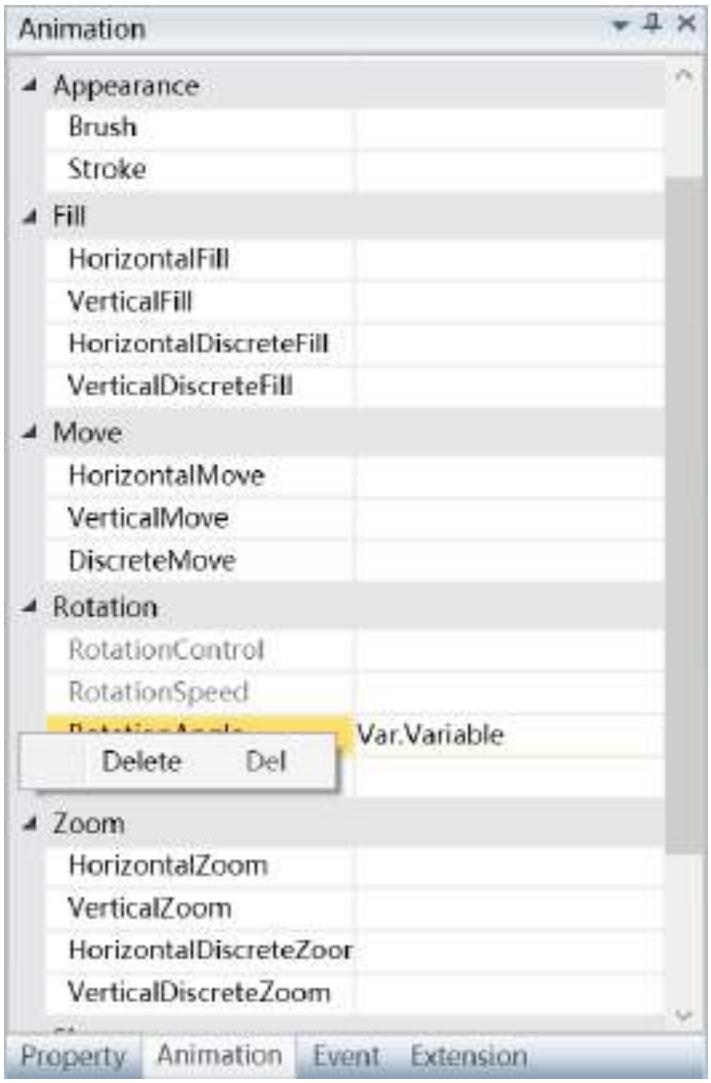
MaximumValue:

- **Value:** Set a maximum value for expression
- **RotationAngle:** Set a maximum value for rotation angle

Step 3: When configuration is complete, press the “OK” button to complete the configuration of the “RotationAngle” animation; the animation configuration of this graphics will be displayed in the “Animation” window, as shown in the figure below:

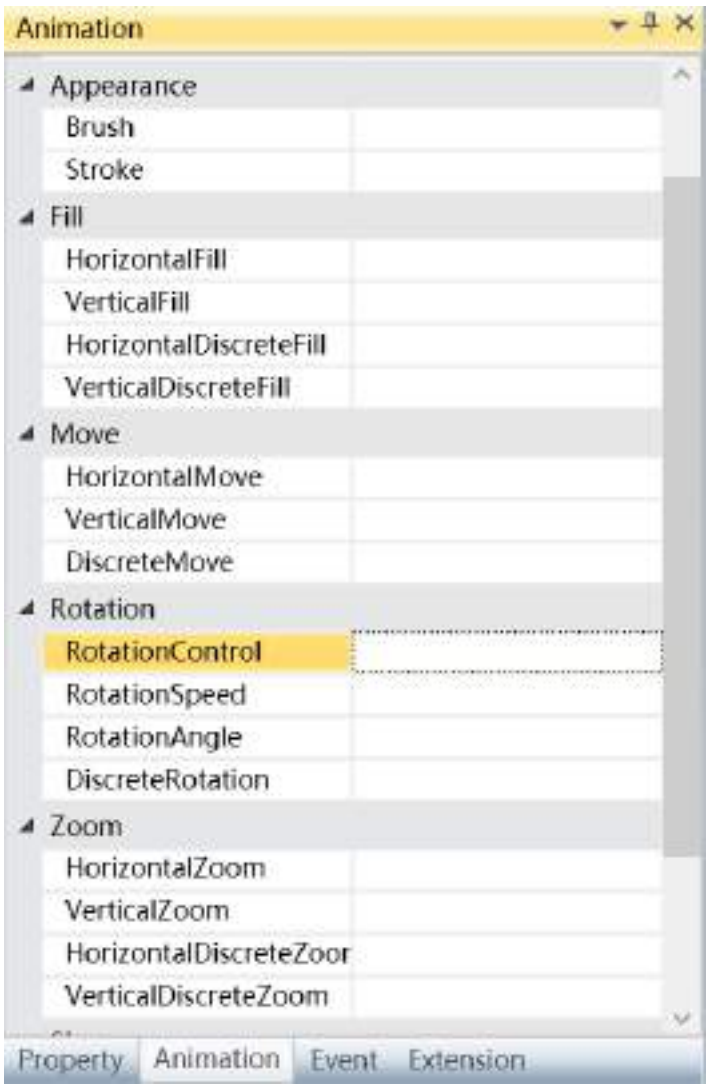


To delete the animation, simply select the animation in the animation list and then right-click the mouse and select “Delete”, as shown in the figure below:

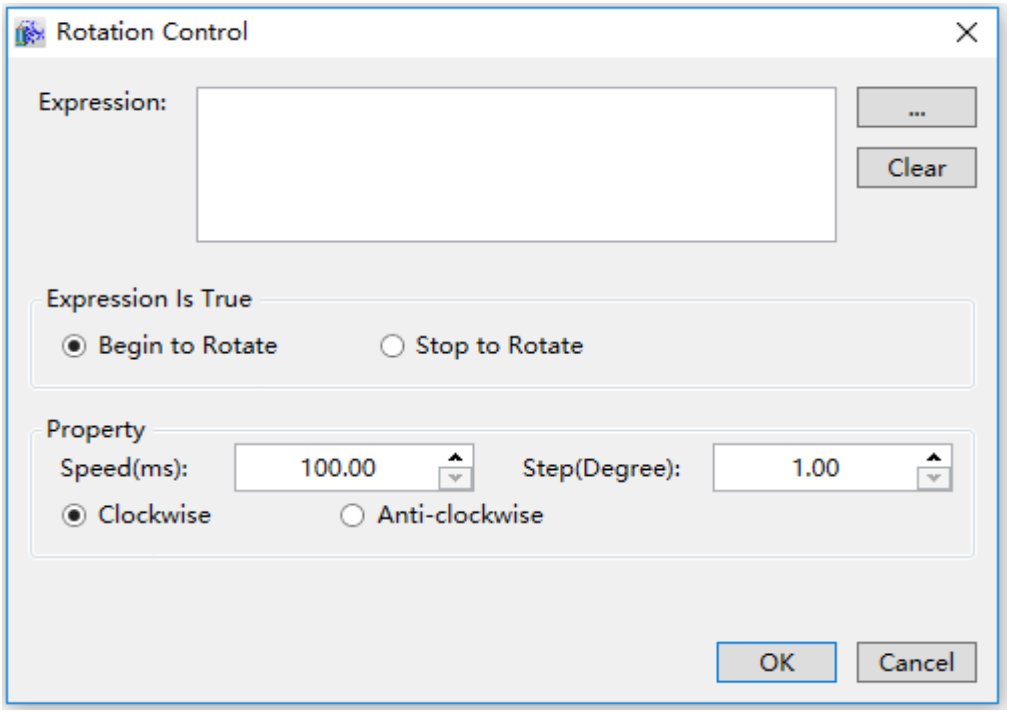


3. Steps to configure rotation control animation are as follows:

Step 1: Open the project window where the animation needs to be configured under the DIAView development environment → select the graphic object to configure animation in the sketchpad → open the “Animation” window → select “RotationControl” → click the button in the “RotationControl” bar , as shown in the figure below:



Step 2: The rotation control window will appear, as shown in the figure below:



The meanings of each setting in the configuration window are as follows:

Expression: Enter the variable name or expression; you can also click the  button to open the variable browser to select a variable, the “Clear” button can clear the contents in the input box.

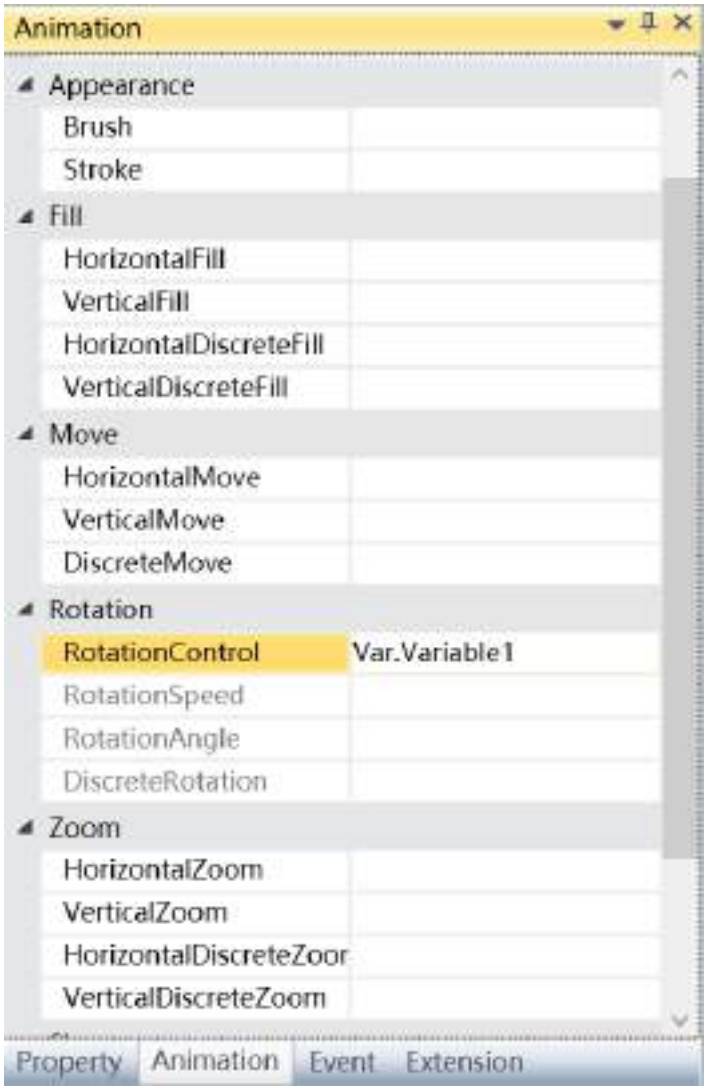
Expression Is True: Set starting or stopping rotation when the expression is true

- **Begin to Rotate :** Start rotating when expression is true
- **Stop to Rotate:** Stop rotating when expression is true

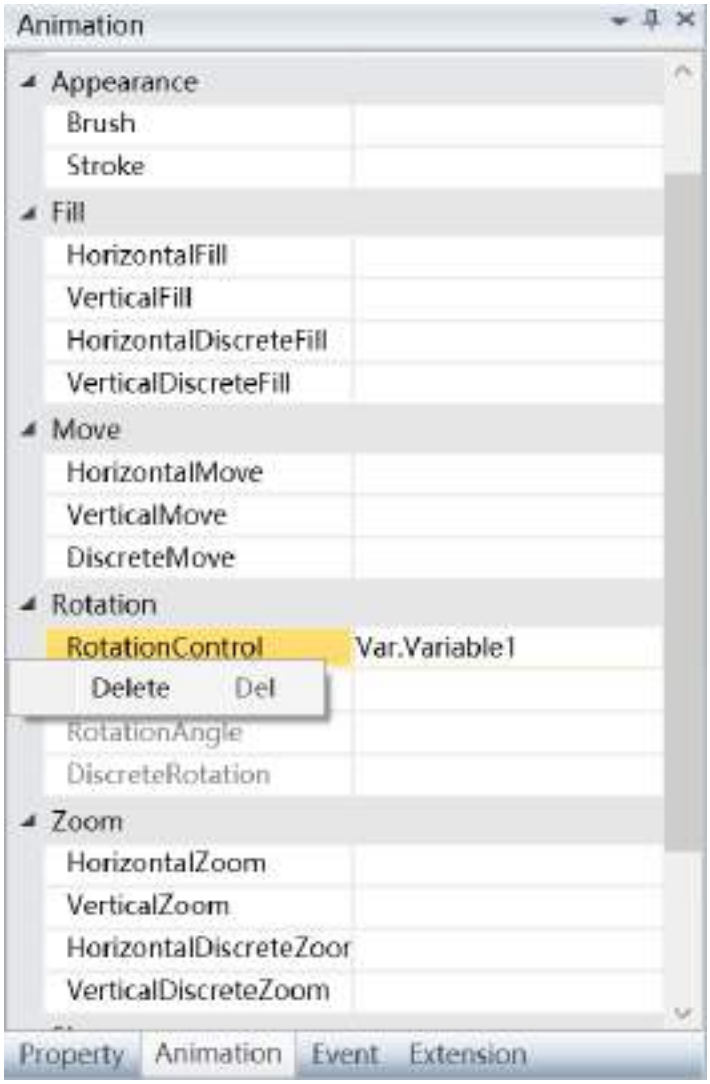
Property:

- **Step(Degree):** Set the angle at which the object rotates one step(Unit:Degree)
- **Speed(ms):** Set the time required for each rotation step of the object(Unit:ms)

Step 3: When configuration is complete, press the “OK” button to complete the configuration of the “RotationControl” animation; the animation configuration of this graphics will be displayed in the “Animation” window, as shown in the figure below:

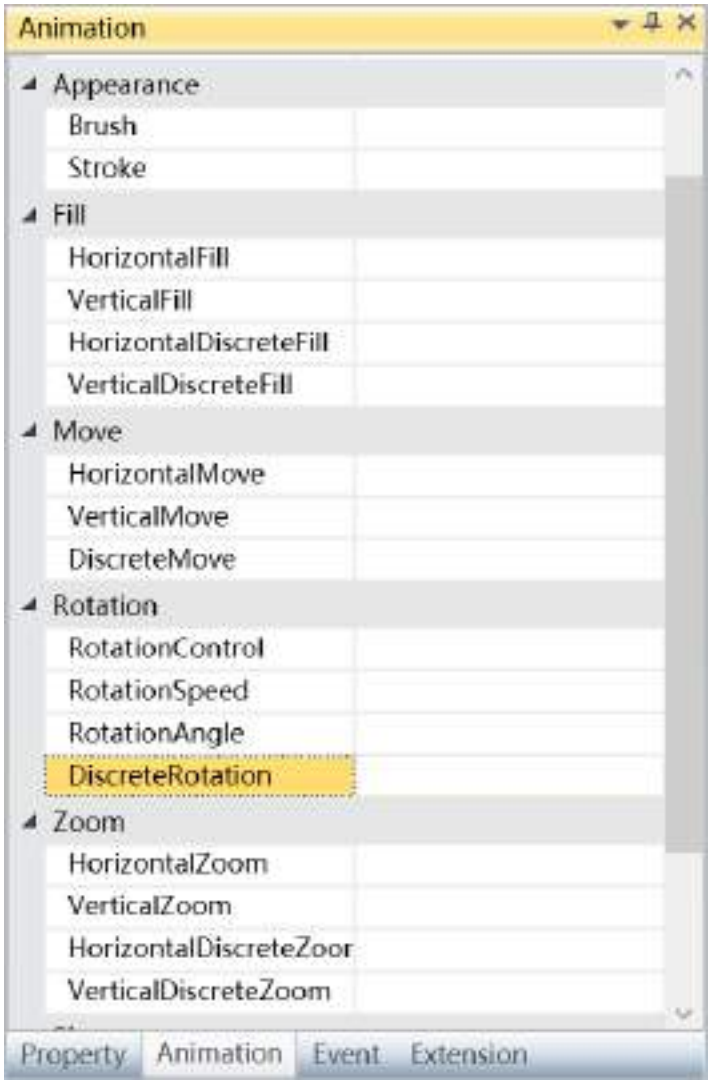


To delete the animation, simply select the animation in the animation list and then right-click the mouse and select “Delete”, as shown in the figure below:

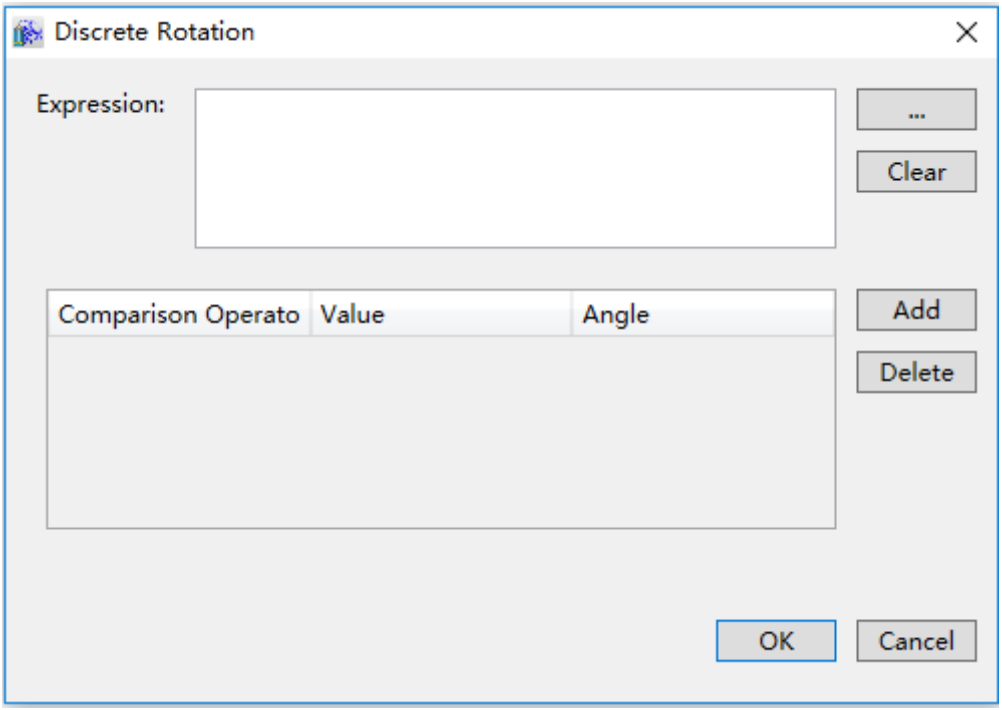


4. Steps to configure discrete rotation animation are as follows:

Step 1: Open the project window where the animation needs to be configured under the DIAView software development environment → select the graphic object to configure animation in the sketchpad → open the “Animation” window → select “DiscreteRotation”→ click the button in the “DiscreteRotation” bar, as shown in the figure below:



Step 2: The discrete rotation animation configuration window will appear, as shown in the figure below:



The meanings of each setting in the configuration window are as follows:

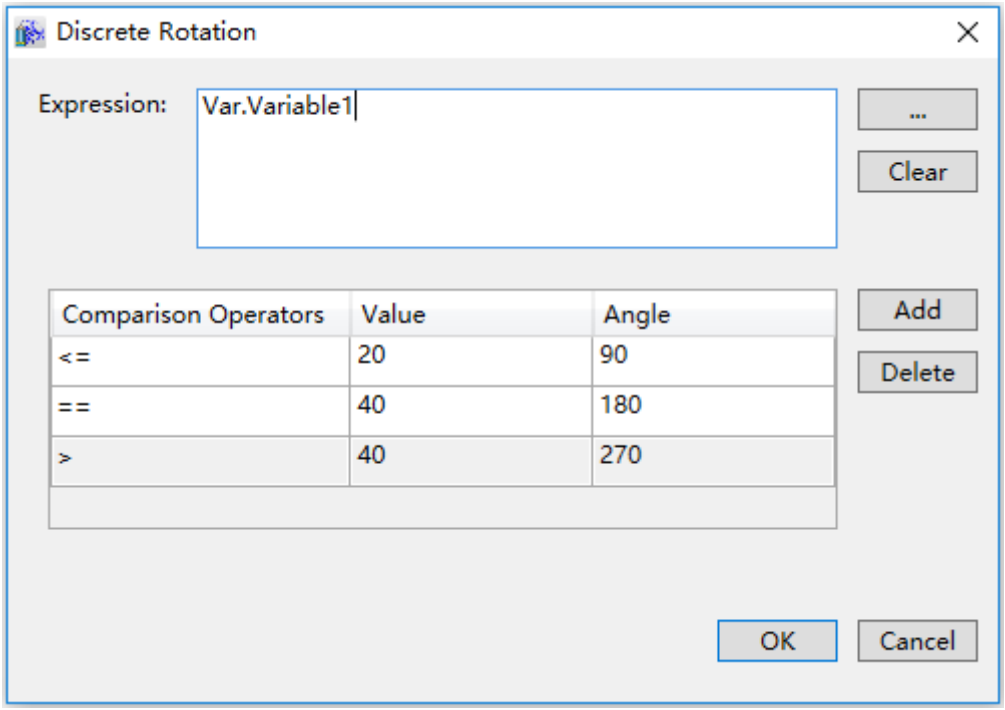
Expression: Enter the variable name or expression; you can also click the  按 button to open the variable browser to select a variable, the “Clear” button can clear the contents in the input box.

Filling configuration:

- Comparison operators:: Sets the comparison operator symbol between the “Expression” and “Expression value”; it is a drop-down menu that includes 6 options:
  - <: Less than
  - <=: Less than or equal to
  - =: Equal to (preset)
  - !=: Not equal to
  - >=: Greater than or equal to
  - >: Greater than
- Value: Sets the reference value for the “Expression” to compare to; it can be an

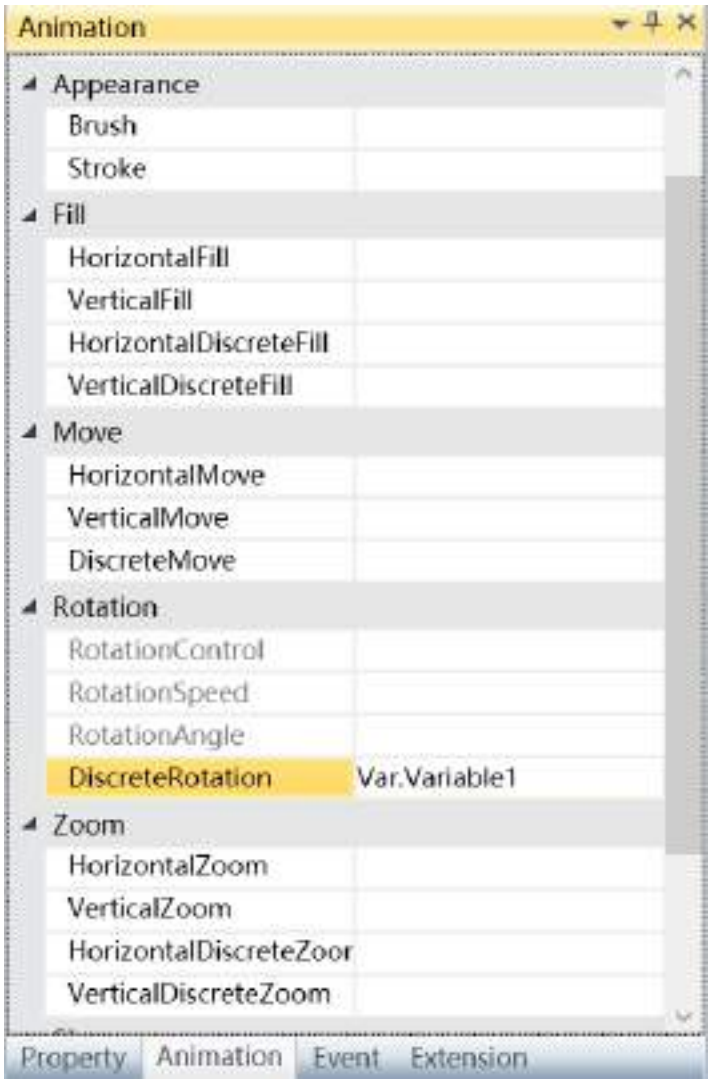
integer or decimal .

- Angle: Sets the corresponding rotation angle.
- “Add” button: Press this button to add configuration items, as shown in the figure below:

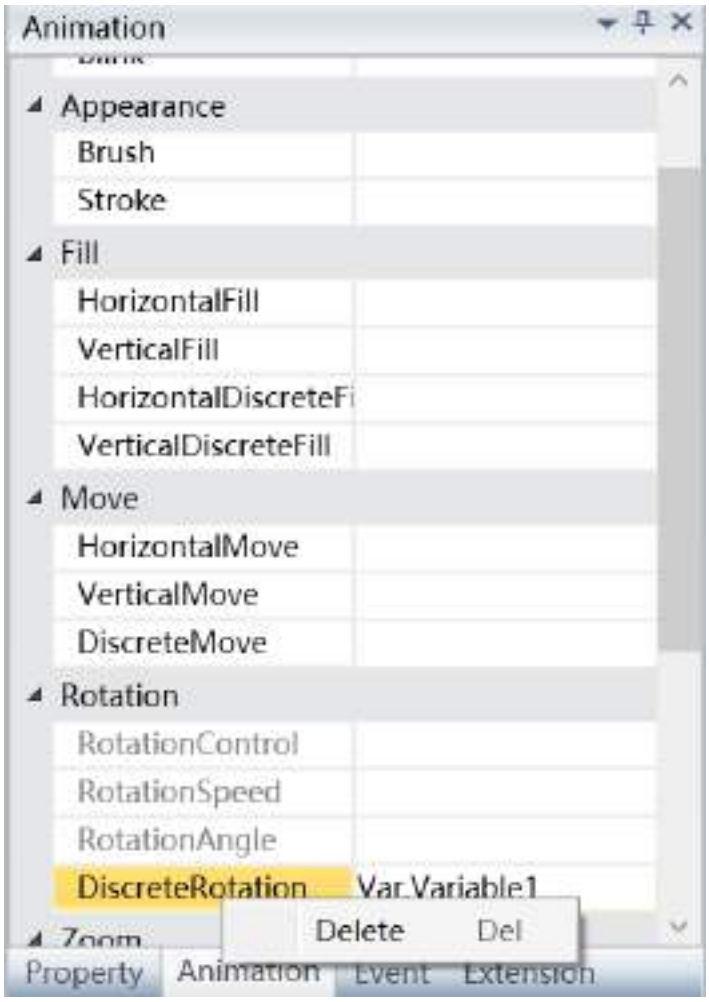


- “Delete” button: Pressing this button after selecting a certain configuration item will delete that configuration item.

Step 3: When configuration is complete, press the “OK” button to complete the configuration of the “DiscreteRotation” animation; the animation configuration of this graphics will be displayed in the “Animation” window, as shown in the figure below:



To delete the animation, simply select the animation in the animation list and then right-click the mouse and select “Delete”.



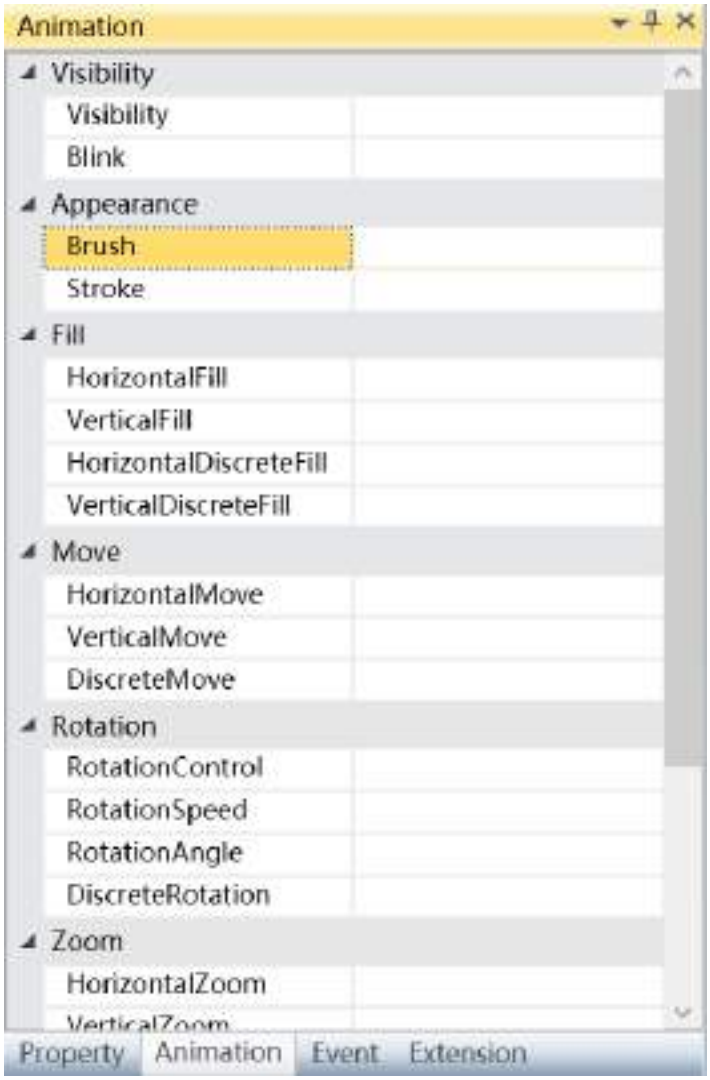
9.3 Appearance animation

Appearance animation refers to changing the brush color of the graphic, and also changing the properties of the graphic including the “background color/fill color” and “line color/border color”.

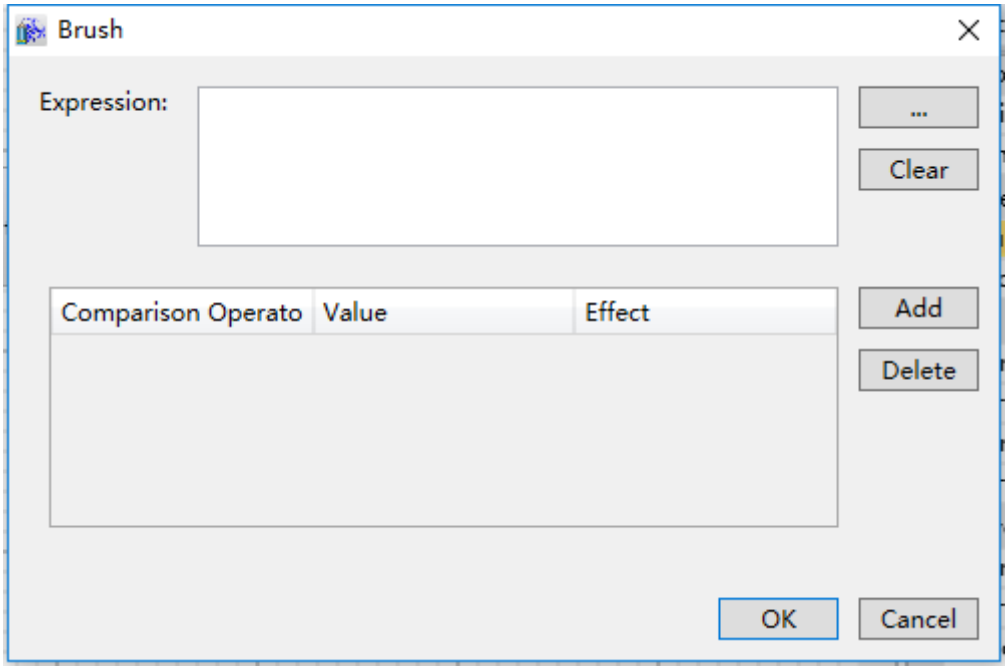
Appearance animations include two property animations “Brush” and “Stroke”; the following is the configuration steps of the “Brush” property animation.

Configuring steps are as follows:

Step 1: Open the project window where the animation needs to be configured under the DIAView software development environment → select the graphic object to configure animation in the sketchpad → open the “Animation” window → select “Brush”→ click the button in the “Brush” field, as shown in the figure below:



Step 2: The brush animation configuration window will appear, as shown in the figure below:



The meanings of each setting in the configuration window are as follows:

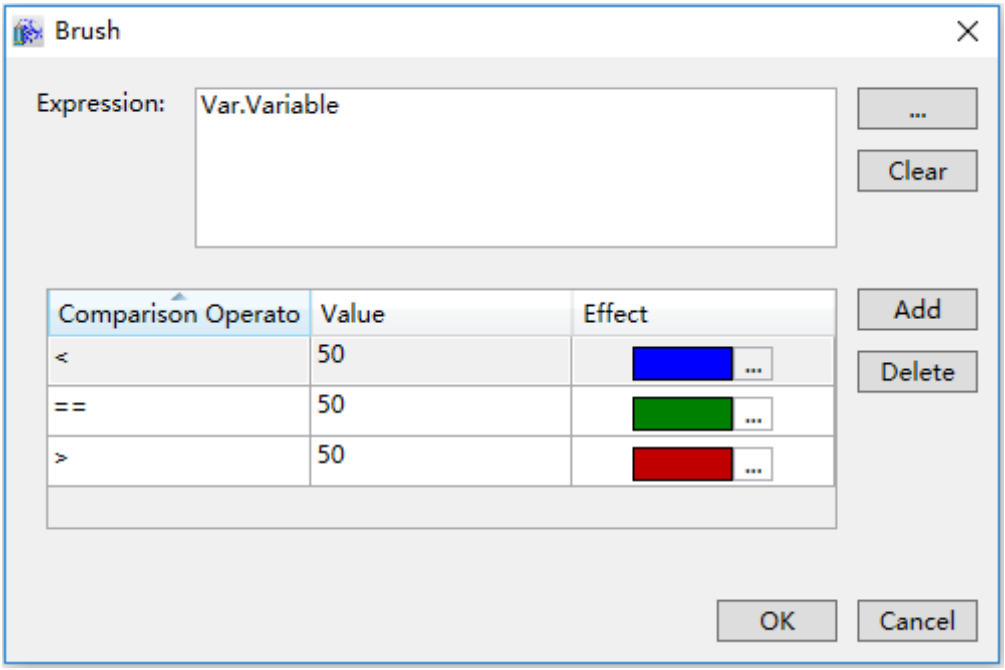
Expression: Enter the variable name or expression; you can also click the  button to open the variable browser to select a variable, the “Clear” button can clear the contents in the input box.

Filling configuration:

- Comparison Operators: Sets the comparison operator symbol between the “Expression” and “Value”; it is a drop-down menu that includes 6 options:
  - <: Less than
  - <=: Less than or equal to
  - =: Equal to (preset)
  - !=: Not equal to
  - >=: Greater than or equal to
  - >: Greater than
- Value: Sets the reference value for the “Expression” to compare to; it can be an integer or decimal

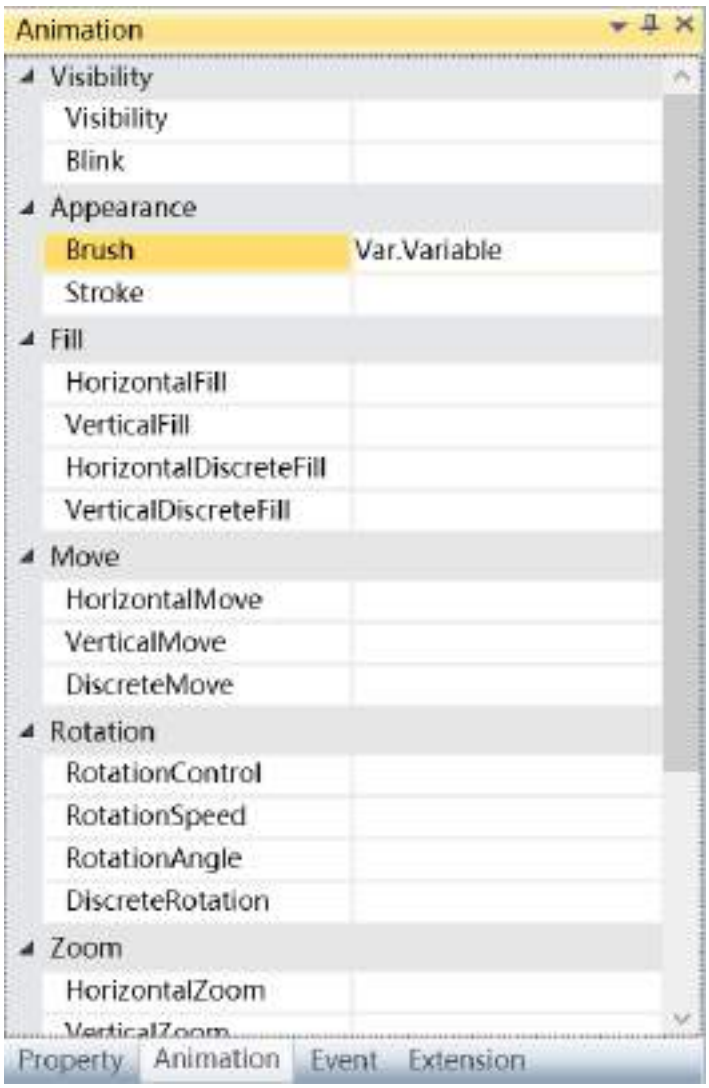
- Effect: Sets the color effects.

- "Add" button: Click this button to add a fill configuration item, as shown in the figure below:

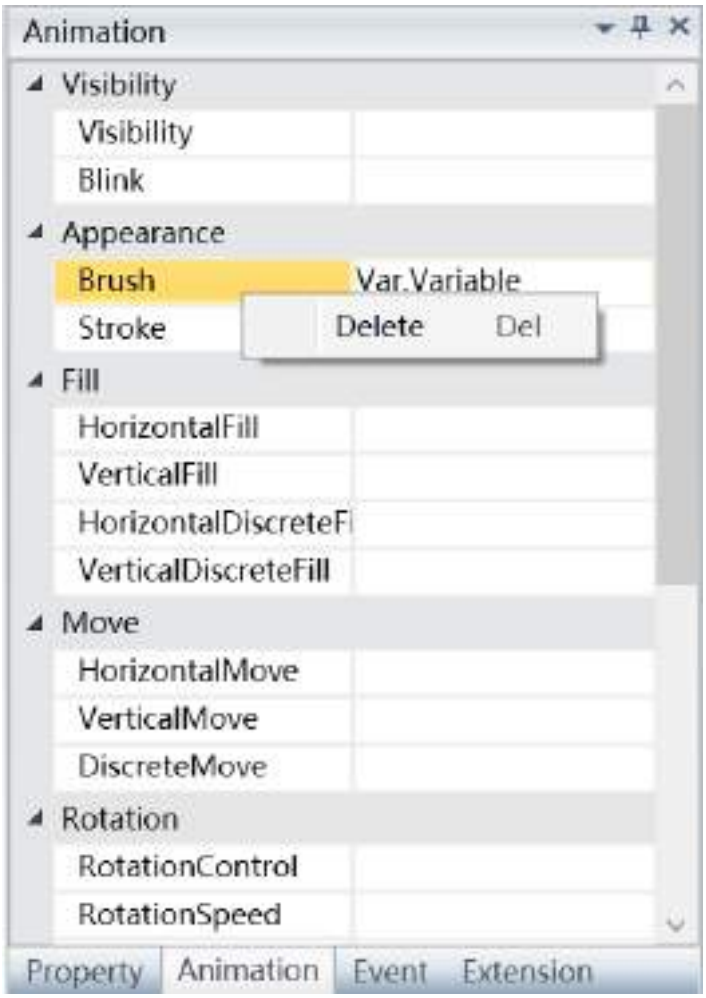


- "Delete" button: Pressing this button after selecting a certain fill configuration item will delete that fill configuration item.

Step 3: When configuration is complete, press the "OK" button to complete the configuration of the "Brush" animation; the animation configuration of this graphics will be displayed in the "Animation" window, as shown in the figure below:



To delete the animation, simply select the animation in the animation list and then right-click the mouse and select “Delete”.



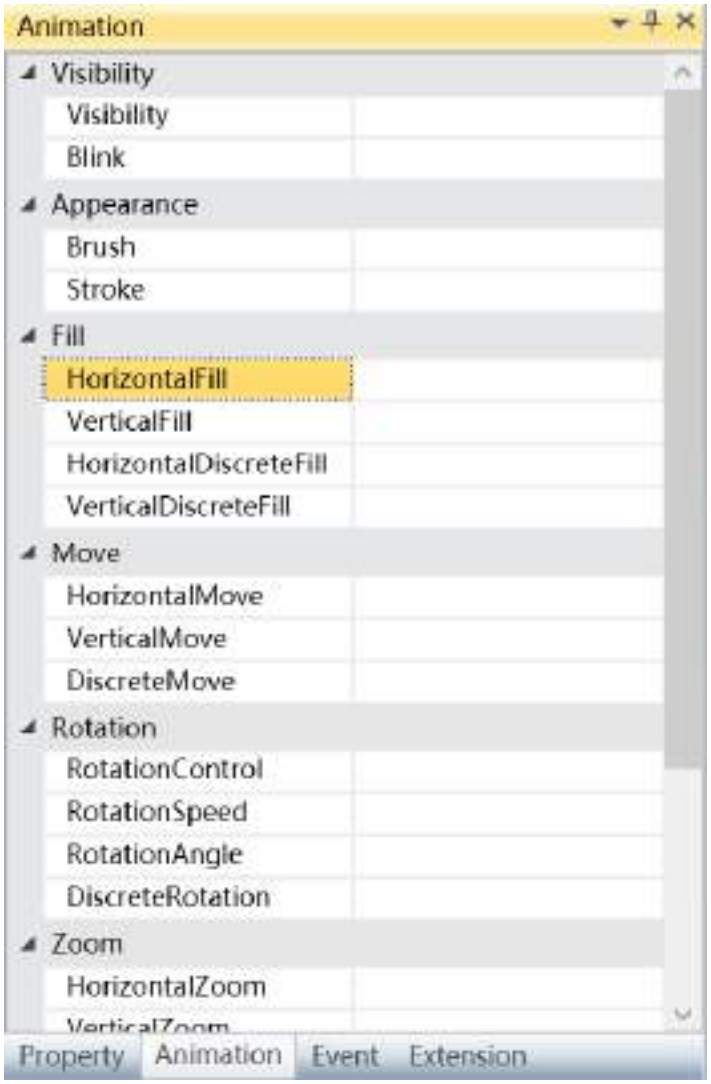
9.4 Fill animation

Fill animation refers to controlling the fill effect of the graphic object through variables or expression values and changing the “fill ratio” and “fill direction” properties of the graphic object.

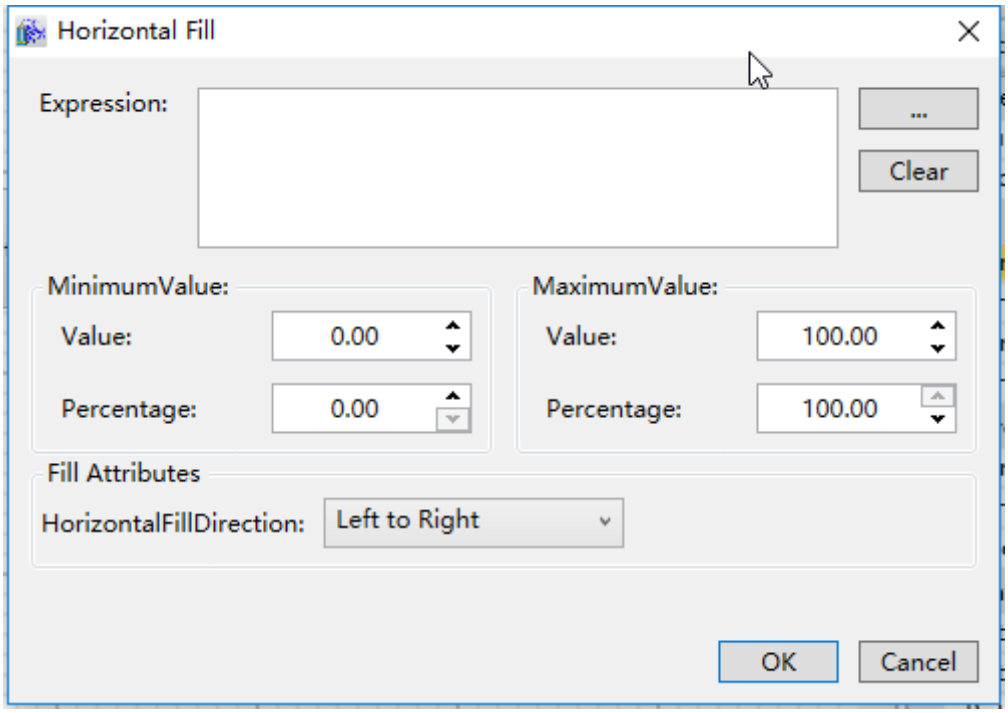
Depending on the fill direction, the fill animation is divided into four types: “HorizontalFill”, “HorizontalDiscreteFill”, “VerticalFill” and “VerticalDiscrete Fill”. The configuration steps for the horizontal and vertical directions are the same. The following are the configuration steps.

1. Steps to configure “HorizontalFill” animations are as follows:

Step 1: Open the project window where the animation needs to be configured under the DIAView development environment → select the graphic object to configure animation in the sketchpad → open the “Animation” window → select “Fill”→ click the button in the “HorizontalFill” field, as shown in the figure below:



Step 2: The horizontal fill animation configuration window will appear, as shown in the figure below:



The meanings of each setting in the configuration window are as follows:

Expression: Enter the variable name or expression; you can also click the  button to open the variable browser to select a variable, the “Clear” button can clear the contents in the input box.

MinimumValue:

- Value: Sets the minimum value of the “Expression”
- Percentage: Sets the minimum value for the fill percentage

MaximumValue:

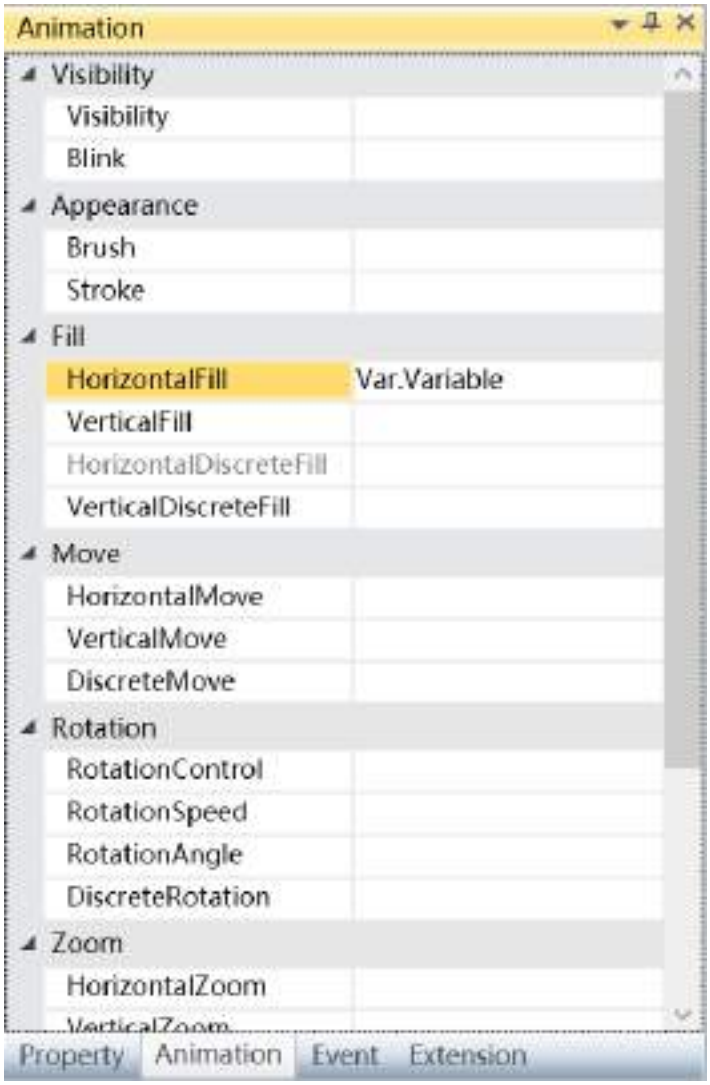
- Value: Sets the maximum value of the “Expression”
- Percentage: Sets the maximum value for the fill percentage

Fill Attributes:

- HorizontalFillDirection: Sets the fill direction; it is a drop-down menu that includes 3 options:
  - Left to Right
  - Right to Left

- Center to Sides

Step3: When configuration is complete, press the “OK” button to complete the configuration of the “HorizontalFill ” animation; the animation configuration of this graphics will be displayed in the “Animation” window, as shown in the figure below:



To delete the animation, simply select the animation in the animation list and then right-click the mouse and select “Delete”.

※ What’s different when configuring “VerticalFill” animations:

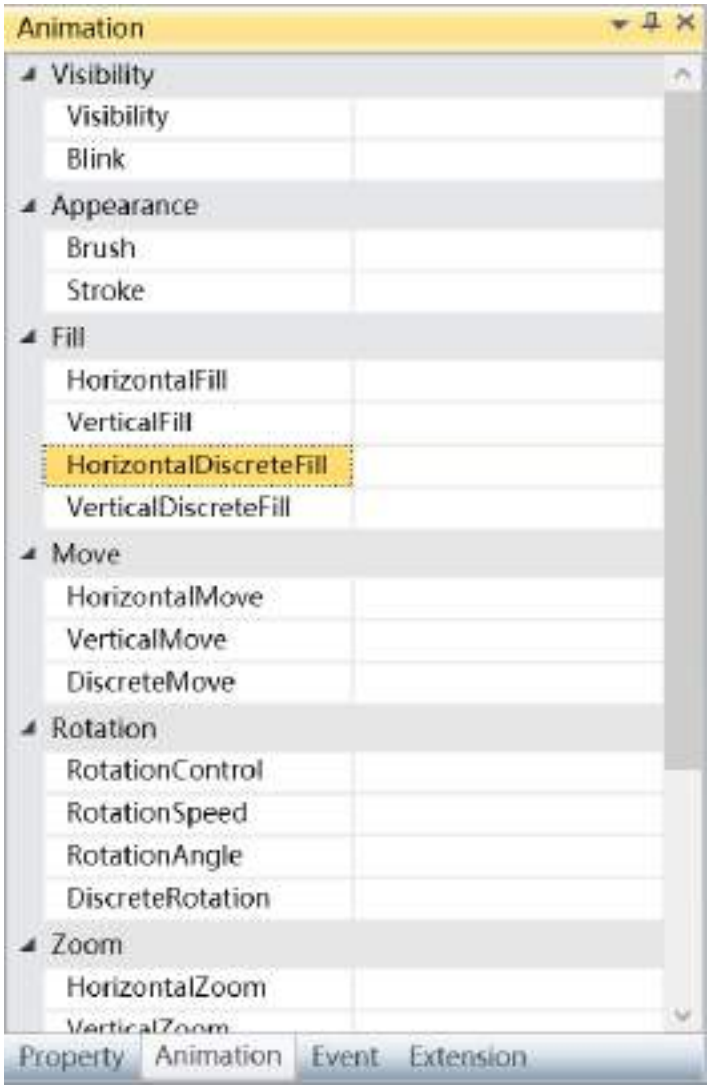
Fill Attributes:

- VerticalDiscreteFill: Sets the fill direction; it is a drop-down menu that includes 3 options:
  - Top to Bottom

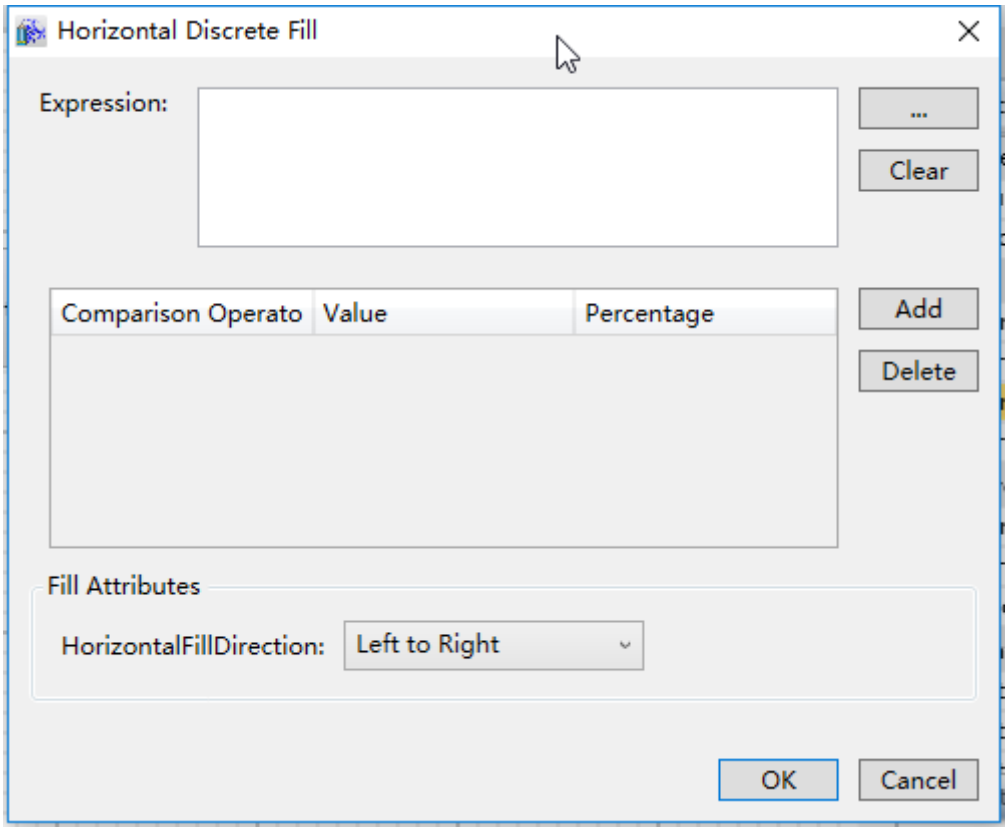
- Bottom to Up
- Center to Sides

2. Steps to configure “HorizontalDiscreteFill” animation are as follows:

Step 1: Open the project window where the animation needs to be configured under the DIAView software development environment → select the graphic object to configure animation in the sketchpad → open the “Animation” window → select “Fill”→ click the button in the “HorizontalDiscreteFill” field, as shown in the figure below:



Step 2: The horizontal discrete fill animation configuration window will appear, as shown in the figure below:



The meanings of each setting in the configuration window are as follows:

Expression: Enter the variable name or expression; you can also click the ... button to open the variable browser to select a variable; the “Clear” button can clear the contents in the input box.

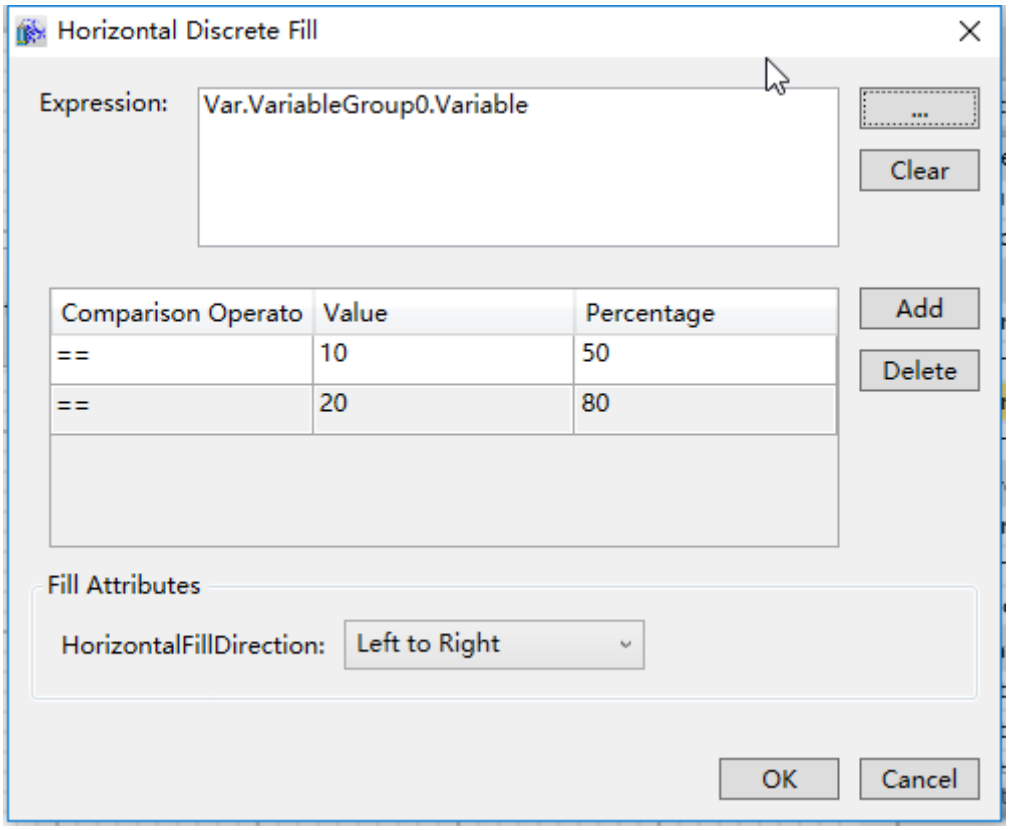
Filling configuration:

- Comparison Operators: Sets the comparison operator symbol between the “Expression” and “Value”, it is a drop-down menu that includes 6 options:
  - <: Less than
  - <=: Less than or equal to
  - =: Equal to (preset)
  - !=: Not equal to
  - >=: Greater than or equal to
  - >: Greater than

- Value: Sets the reference value for the “Expression” to compare to; it can be an integer or decimal .

- Percentage: Sets the corresponding fill percentage.

- “Add” button: Press this button to add configuration items, as shown in the figure below:



- “Delete” button: Pressing this button after selecting a certain configuration item will delete that configuration item.

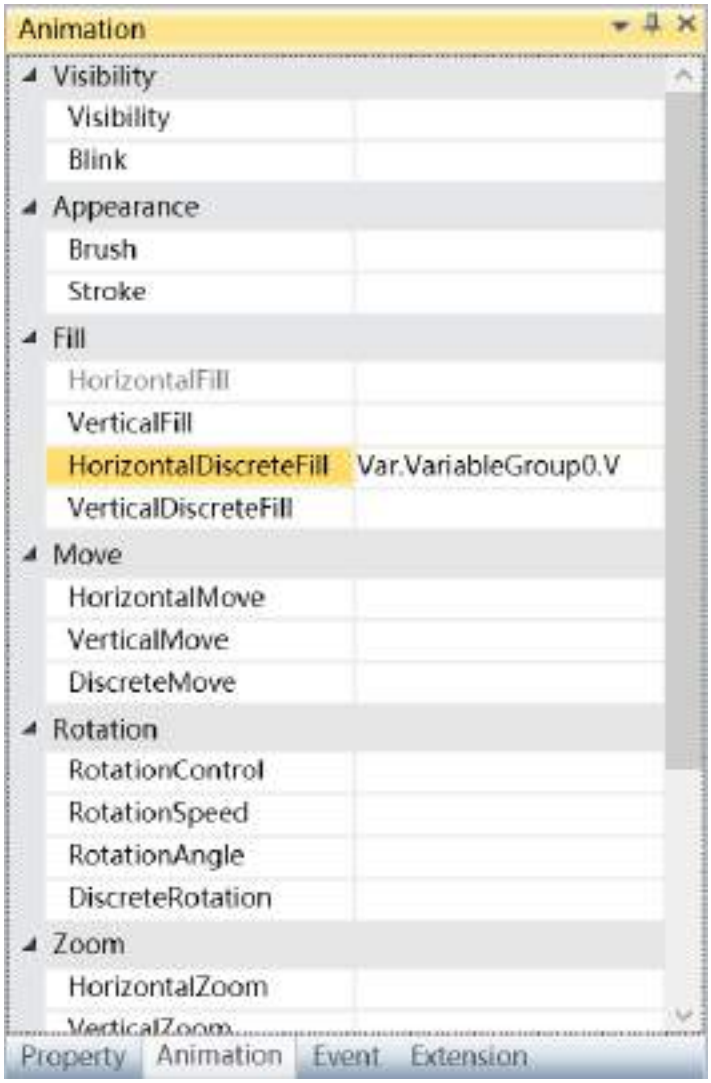
Fill Attributes:

- HorizontalFillDirection: Sets the fill direction; it is a drop-down menu that includes 3 options:

  - Left to Right
  - Right to Left
  - Center to Sides

Step 3: When configuration is complete, press the “OK” button to complete the configuration of the

"HorizontalDiscreteFill" animation; the animation configuration of this graphics will be displayed in the "Animation" window, as shown in the figure below:



To delete the animation, simply select the animation in the animation list and then right-click the mouse and select "Delete".

※ What's different when configuring "VerticalDiscreteFill" animations:

Fill Attributes:

➢ Vertical discrete fill direction: Sets the fill direction; it is a drop-down menu that includes 3 options:

- Top to Bottom
- Bottom to Up

- Center to Sides

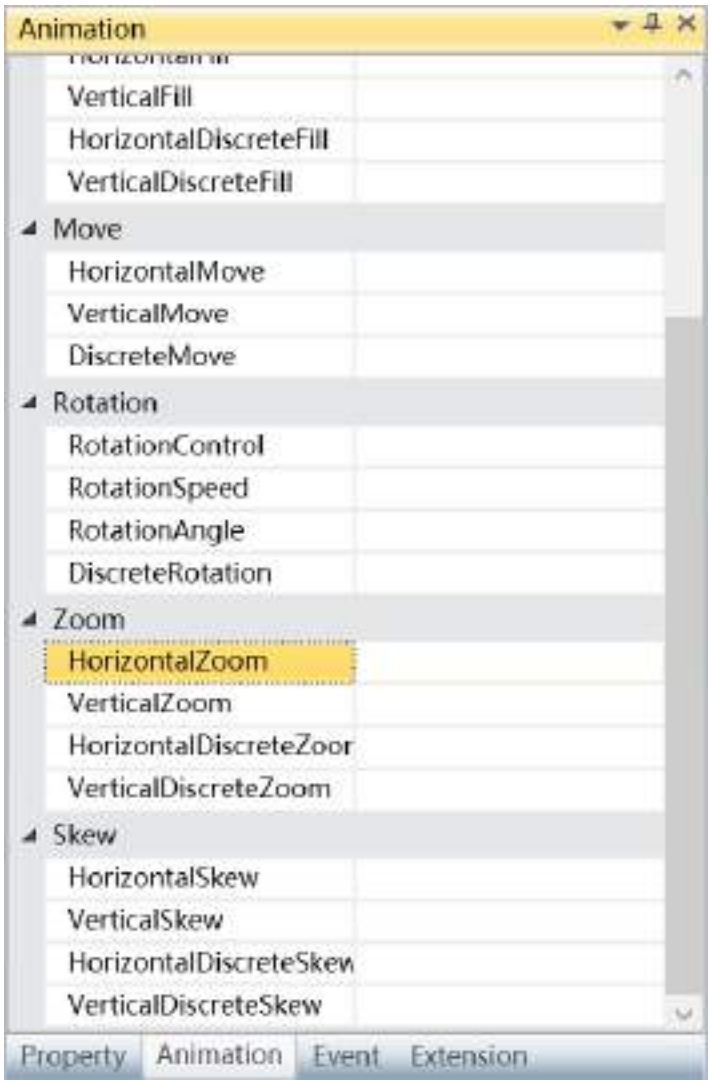
9.5 Zoom animation

Zoom animation refers to controlling the size of the graphic object through variables or expression values and changing the “size” property of the graphic object.

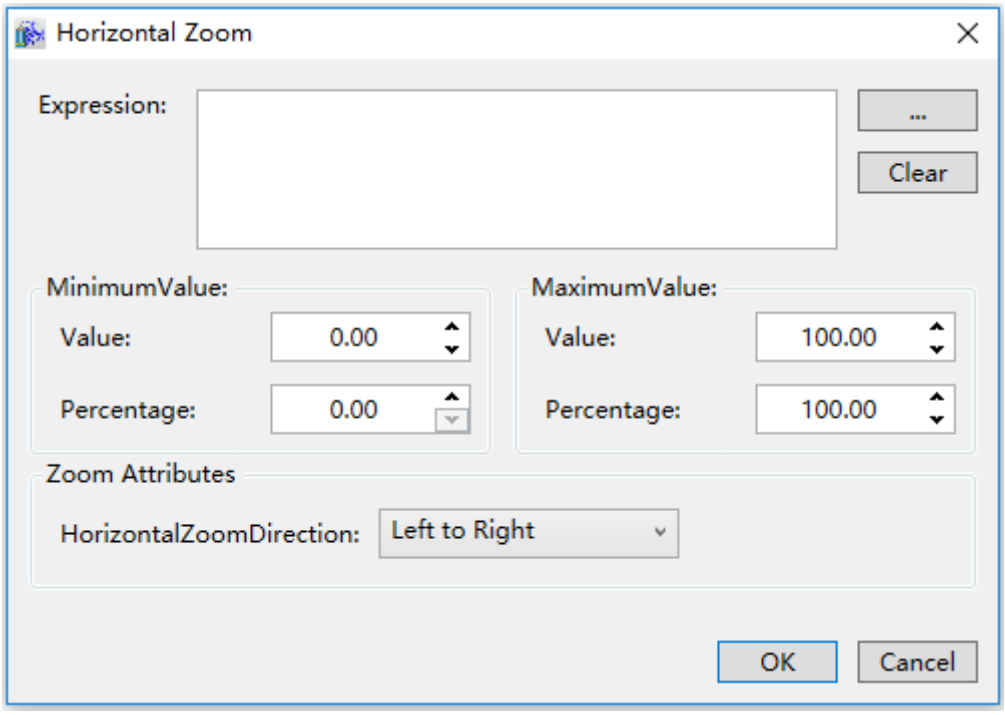
Depending on the zoom direction, the zoom animation is divided into four types: “HorizontalZoom”, “HorizontalDiscreteZoom”, “VerticalZoom” and “VerticalDiscreteZoom”. The configuration steps for the horizontal and vertical directions are the same. The following are the configuration steps.

1.Steps to configure “HorizontalZoom” animations are as follows:

Step 1: Open the project window where the animation needs to be configured under the DIAView software development environment → select the graphic object to configure animation in the sketchpad → open the “Animation” window → select “HorizontalZoom”→ click the button in the “HorizontalZoom” bar, as shown in the figure below:



Step 2: The horizontal zoom animation configuration window will appear, as shown in the figure below:



The meanings of each setting in the configuration window are as follows::

Expression: Input the variable name or expression; you can also click the  button to open the variable browser to select a variable, the “Clear” button can clear the contents in the input box.

MinimumValue:

- Value: Sets the minimum value of the “Expression”
- Percentage: Sets the minimum value for the horizontal zoom percentage

MaximumValue:

- Value: Sets the maximum value of the “Expression”
- Percentage: Sets the maximum value for the horizontal zoom percentage

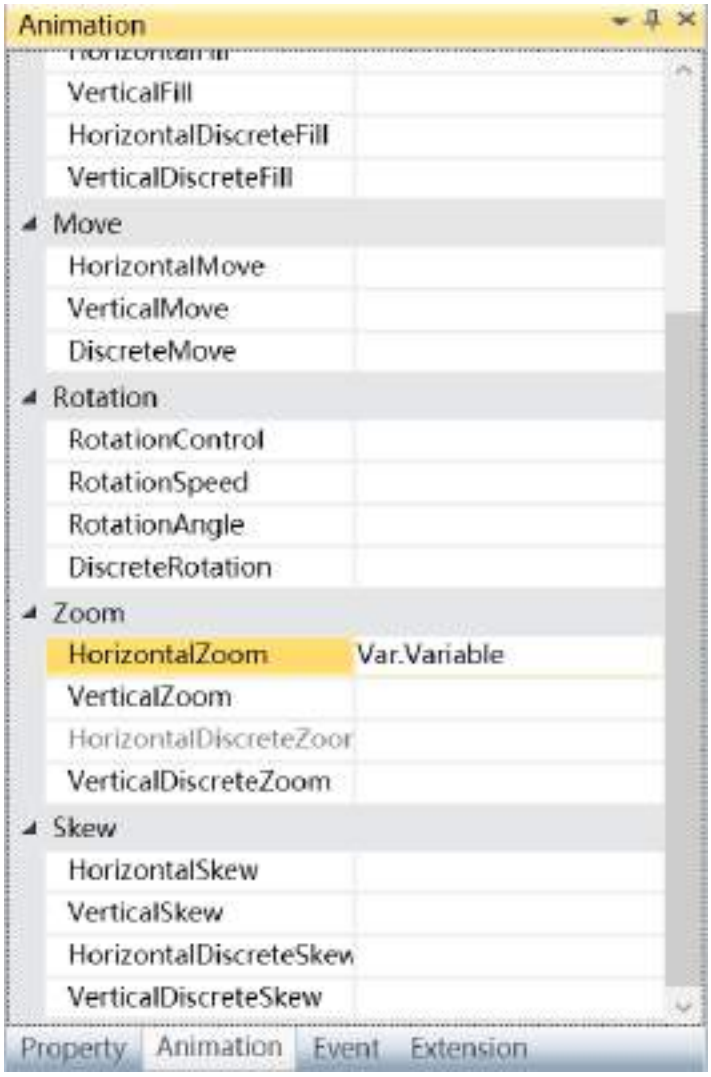
Zoom Attributes:

- HorizontalZoomDirection: Sets the zoom direction; it is a drop-down menu that includes 3 options:
  - Left to Right

- Right to Left

- Center to Sides

Step 3: When configuration is complete, press the “OK” button to complete the configuration of the “HorizontalZoom” animation; the animation configuration of this graphics will be displayed in the “Animation” window, as shown in the figure below:



To delete the animation, simply select the animation in the animation list and then right-click the mouse and select “Delete”.

 What’s different when configure “Vertical zoom” animations:

Zoom Attributes:

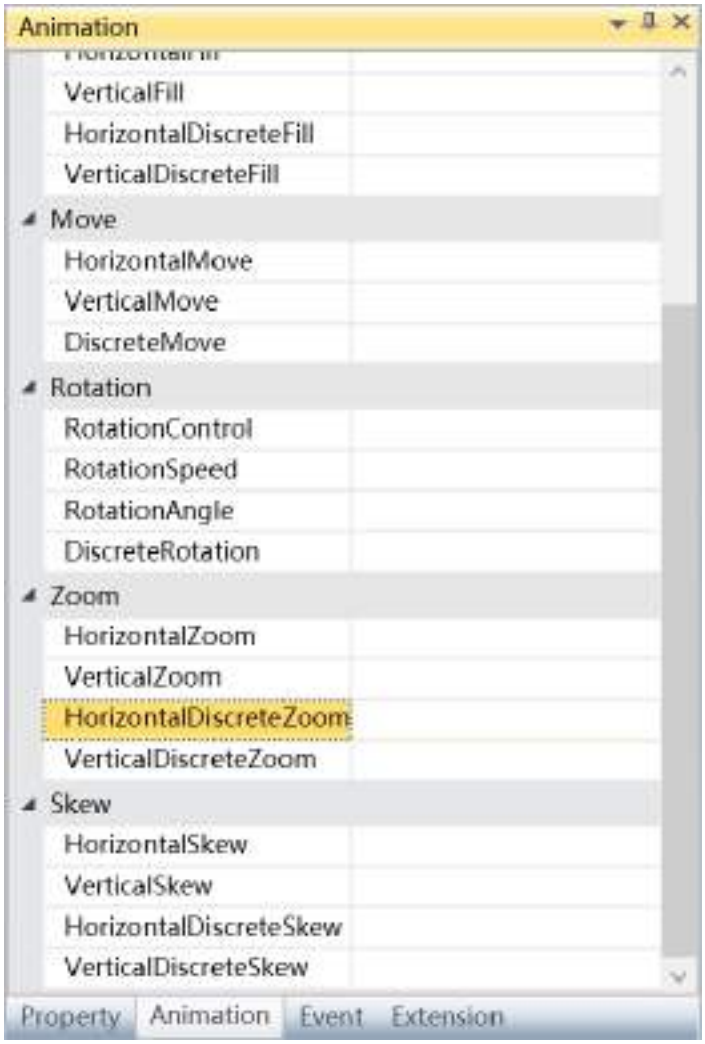
- VerticalZoomDirection: Sets the zoom direction; it is a drop-down menu that includes 3

options:

- Top to Bottom
- Bottom to Up
- Center to Sides

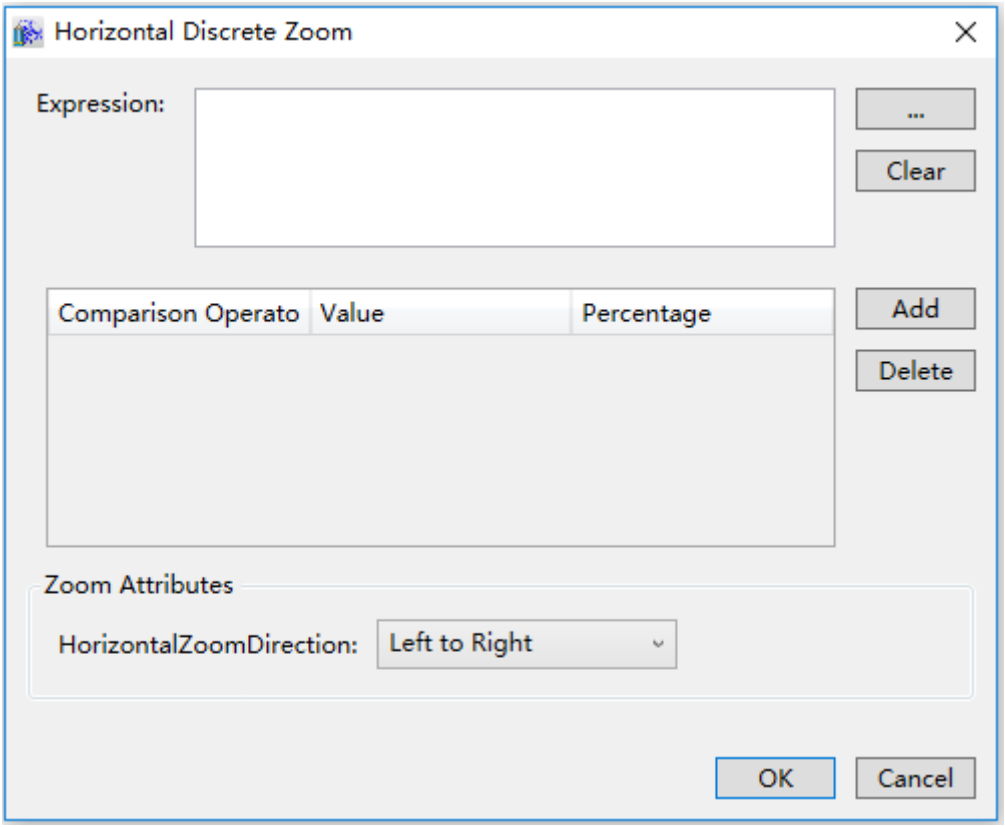
2.Steps to configure “HorizontalDiscreteZoom” animations are as follows:

Step 1: Open the project window where the animation needs to be configured under the DIAView software development environment → select the graphic object to configure animation in the sketchpad → open the “Animation” window → select “HorizontalDiscreteZoom” → click the button in the “HorizontalDiscreteZoom” bar, as shown in the figure below:



Step 2: The horizontal discrete zoom animation configuration window will appear, as shown in the

figure below:



The meanings of each setting in the configuration window are as follows:

Expression: Enter the variable name or expression; you can also click the ... button to open the variable browser to select a variable; the “Clear” button can clear the contents in the input box.

Zoom configuration:

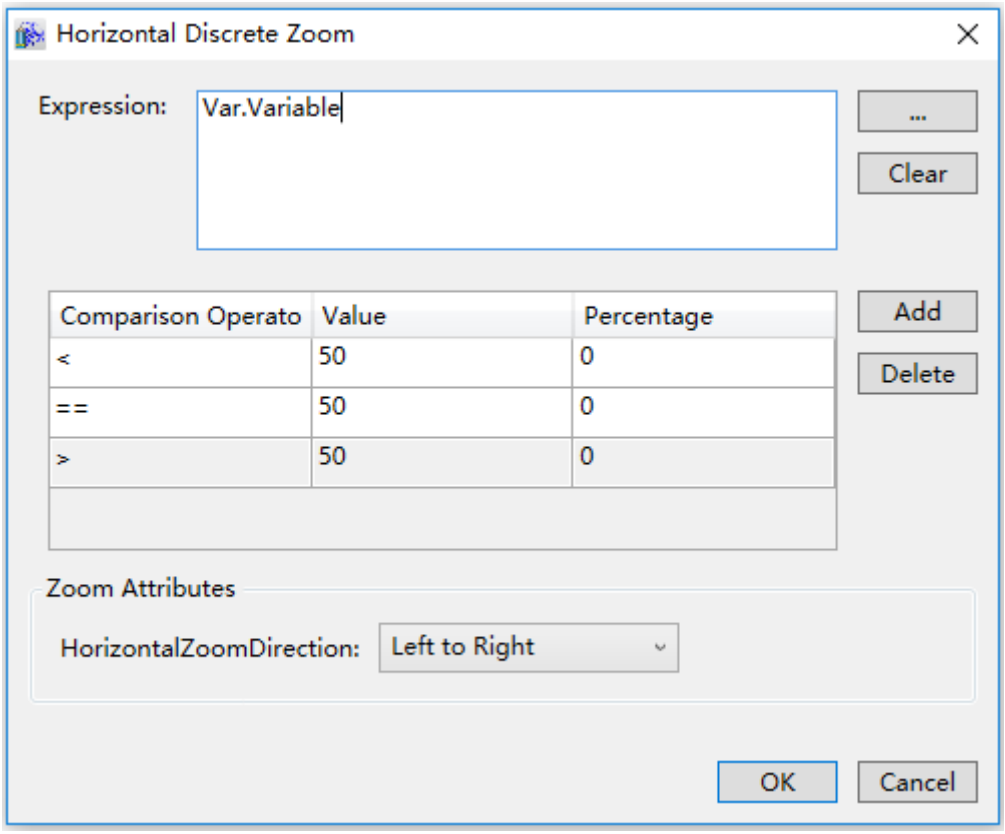
- Comparison operators: Sets the comparison operator symbol between the “Expression” and “Value”, it is a drop-down menu that includes 6 options:
  - <: Less than
  - <=: Less than or equal to
  - =: Equal to (preset)
  - !=: Not equal to
  - >=: Greater than or equal to

- >: Greater than

➢ Value: Sets the reference value for the “Expression” to compare to; it can be an integer or decimal.

➢ Percentage: Sets the corresponding zoom percentage.

➢ “Add” button: Press this button to add configuration items, as shown in the figure below：



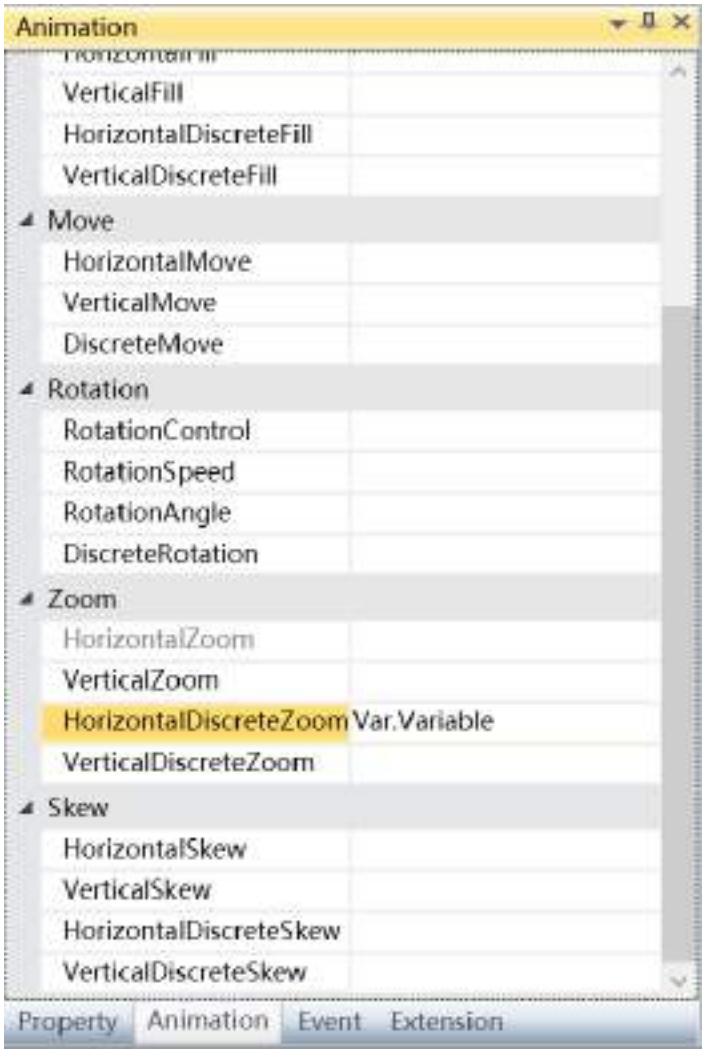
➢ “Delete” button: Pressing this button after selecting a certain configuration item will delete that configuration item.

Zoom Attributes:

➢ HorizontalZoomDirection: Sets the zoom direction; it is a drop-down menu that includes 3 options

- Left to Right
- Right to Left
- Center to Sides

Step 3: 3:When configuration is complete, press the “OK” button to complete the configuration of the “Horizontal discrete zoom” animation; the animation configuration of this graphics will be displayed in the “Animation” window, as shown in the figure below:



To delete the animation, simply select the animation in the animation list and then right-click the mouse and select “Delete”.

 What’s different when configuring “Vertical discrete zoom” animations:

Zoom Attributes

- VerticalZoomDirection: Sets the zoom direction; it is a drop-down menu that includes 3 options
  - Top to Bottom
  - Bottom to Up

- Center to Sides

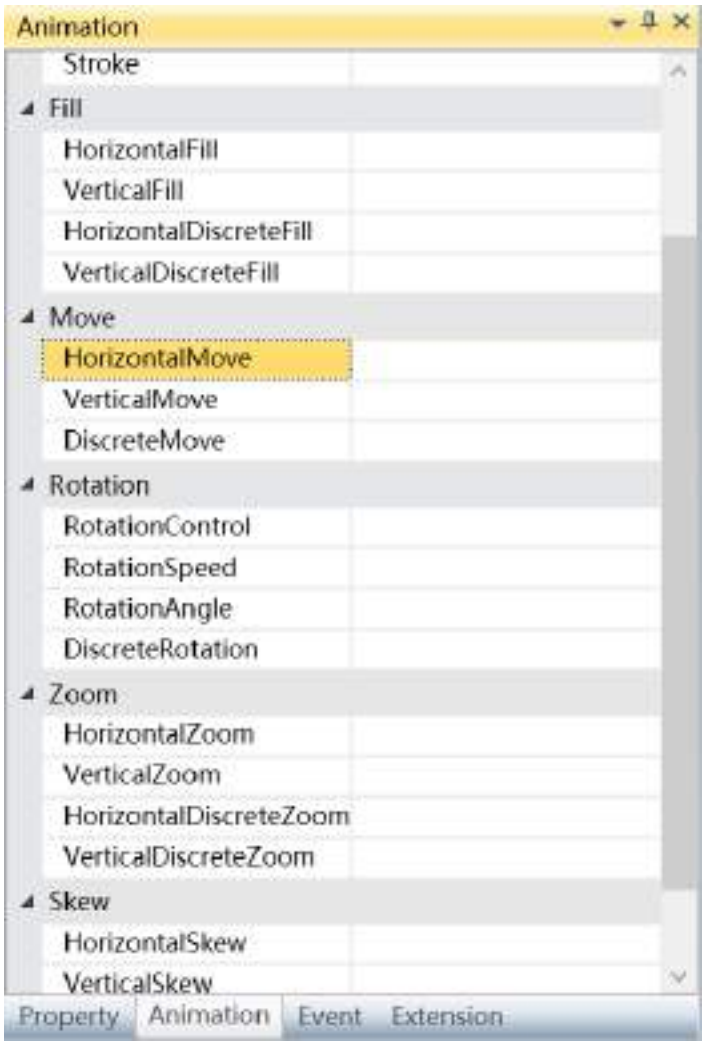
9.6 Move animation

Move animation refers to controlling the position of the graphic object through variables or expression values and changing the “coordinates” property of the graphic object.

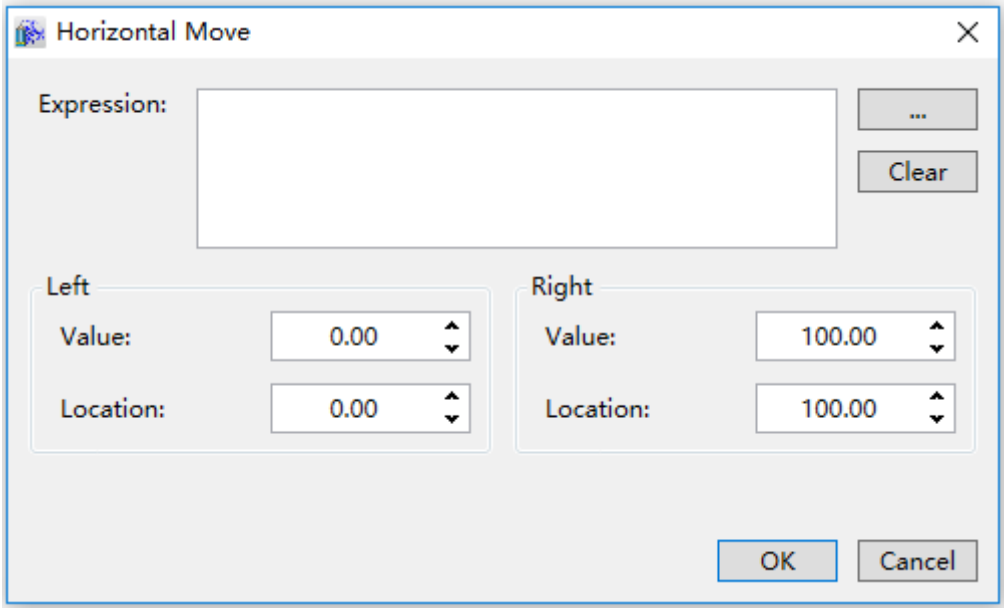
Move animation is divided into three types: “HorizontalMove”, “VerticalMove” and “DiscreteMove”, in which the configuration steps of horizontal and vertical movement are the same. The following are the configuration steps:

1.Steps to configure “HorizontalMove” animations are as follows:

Step 1: Open the project window where the animation needs to be configured under the DIAView software development environment → select the graphic object to configure animation in the sketchpad → open the “Animation” window → select “HorizontalMove”→ click the button in the “HorizontalMove” bar, as shown in the figure below:



Step 2: The horizontal move animation configuration window will appear, as shown in the figure below:



The meanings of each setting in the configuration window are as follows:

Expression: Enter the variable name or expression; you can also click the  button to open the variable browser to select a variable, the “Clear” button can clear the contents in the input box.

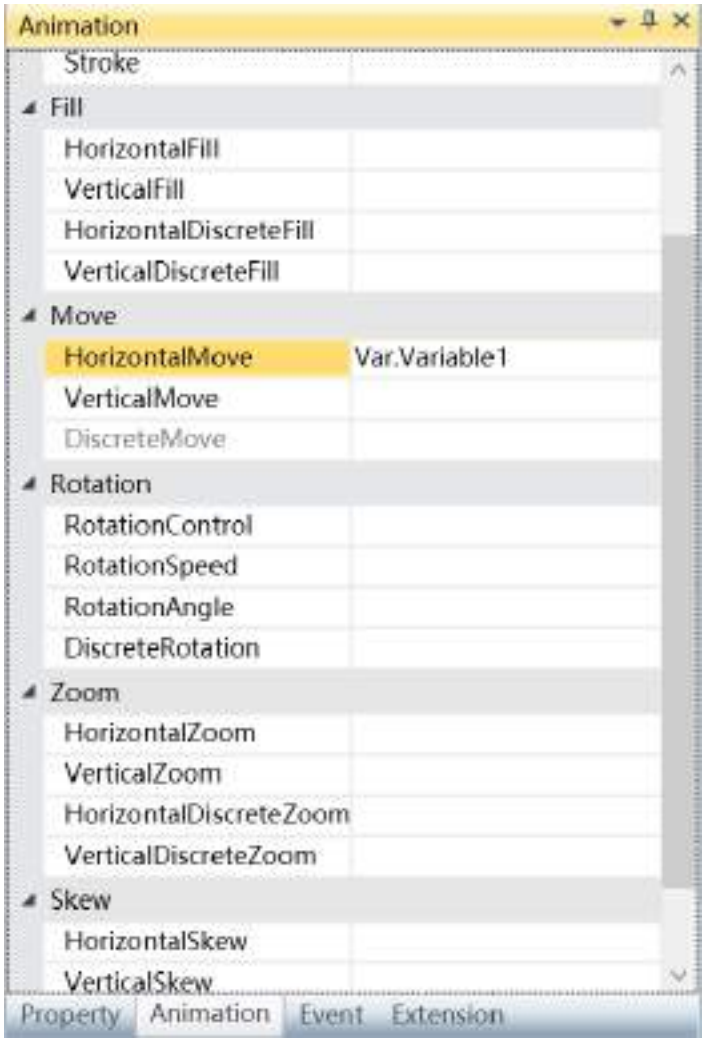
Left:

- Value: Sets the minimum value of the “Expression”
- Location: Sets the coordinates of the left-most position that can be reached for horizontal movement

Right:

- Value: Sets the maximum value of the “Expression”
- Location: Sets the coordinates of the right-most position that can be reached for horizontal movement

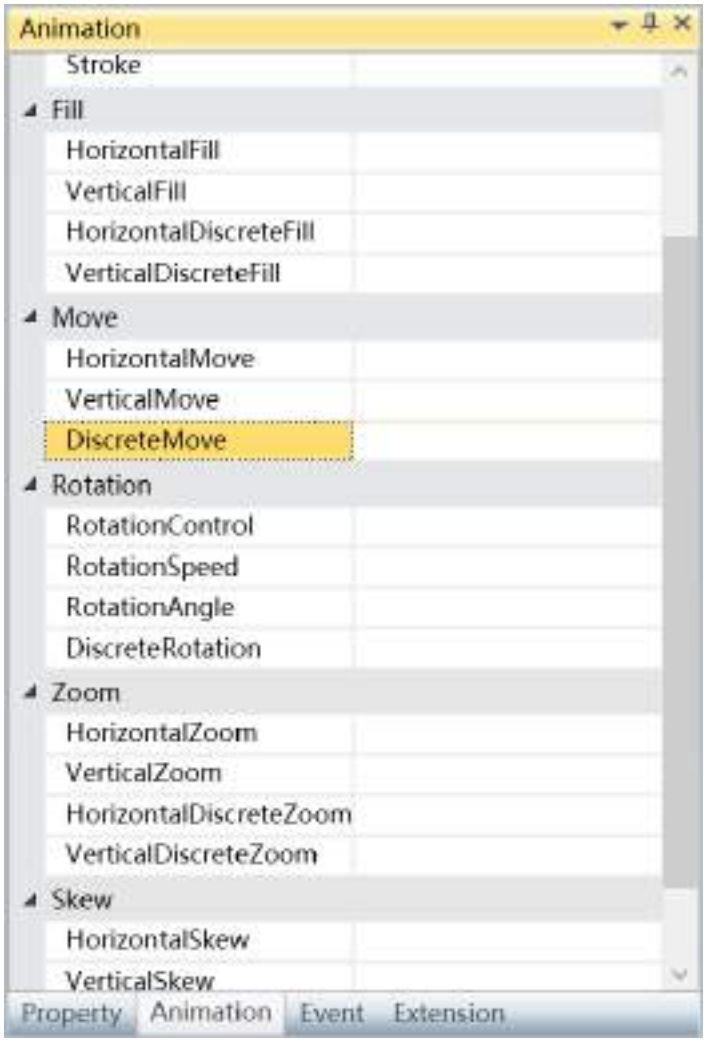
Step 3: When configuration is complete, press the “OK” button to complete the configuration of the “HorizontalMove” animation; the animation configuration of this graphics will be displayed in the “Animation” window, as shown in the figure below:



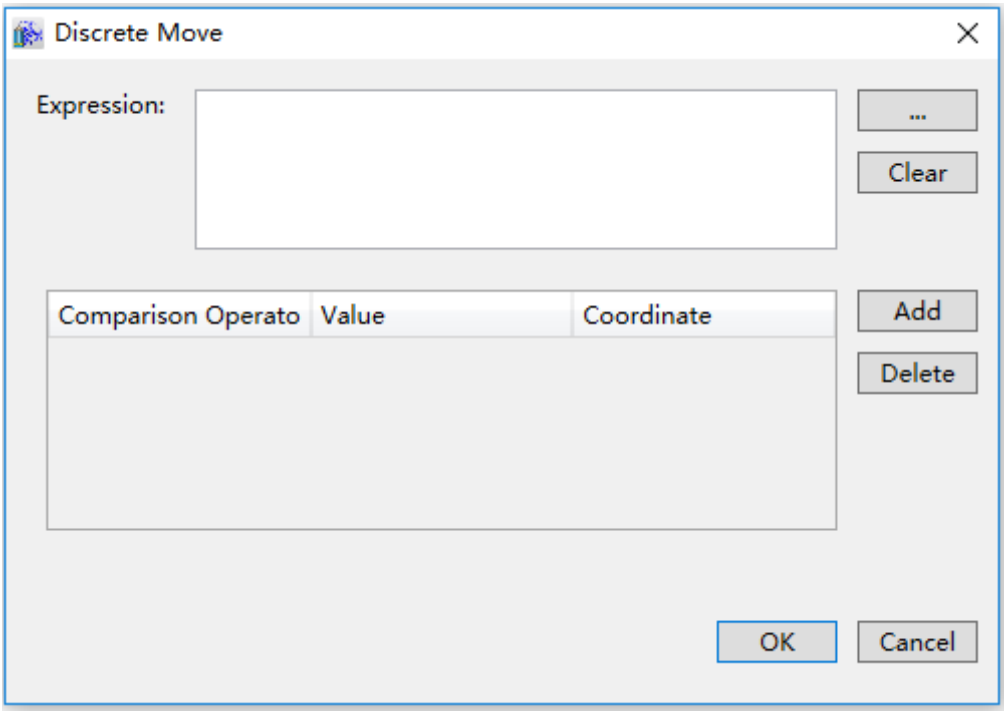
To delete the animation, simply select the animation in the animation list and then right-click the mouse and select “Delete”.

2. Steps to configure “DiscreteMove” animations are as follows:

Step 1: Open the project window where the animation needs to be configured under the DIAView software development environment → select the graphic object to configure animation in the sketchpad → open the “Animation” window → select “DiscreteMove”→ click the button in the “DiscreteMove” field, as shown in the figure below:



Step 2: The discrete move animation configuration window will appear, as shown in the figure below:



The meanings of each setting in the configuration window are as follows:

Expression: Enter the variable name or expression; you can also click the  button to open the variable browser to select a variable; the “Clear” button can clear the contents in the input box.

Move configurations:

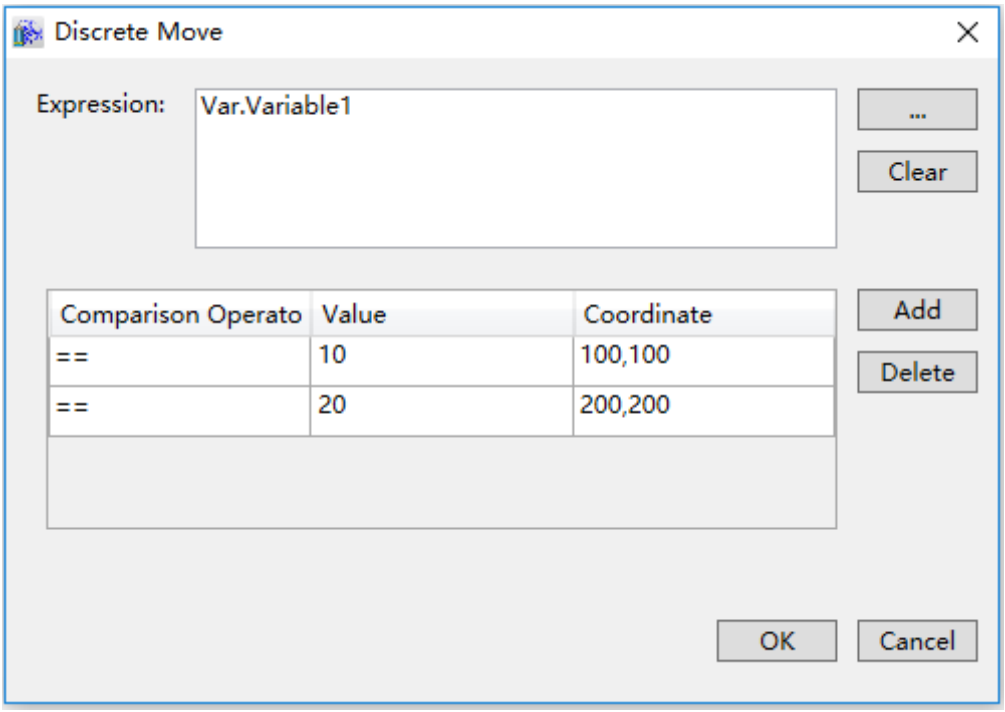
➢ Comparison Operators: Sets the comparison operator symbol between the “Expression” and “Value”, it is a drop-down menu that includes 6 options:

- <: Less than
- <=: Less than or equal to
- =: Equal to (preset)
- !=: Not equal to
- >=: Greater than or equal to
- >: Greater than

➢ Value: Sets the reference value for the “Expression” to compare to; it can be an integer or decimal.

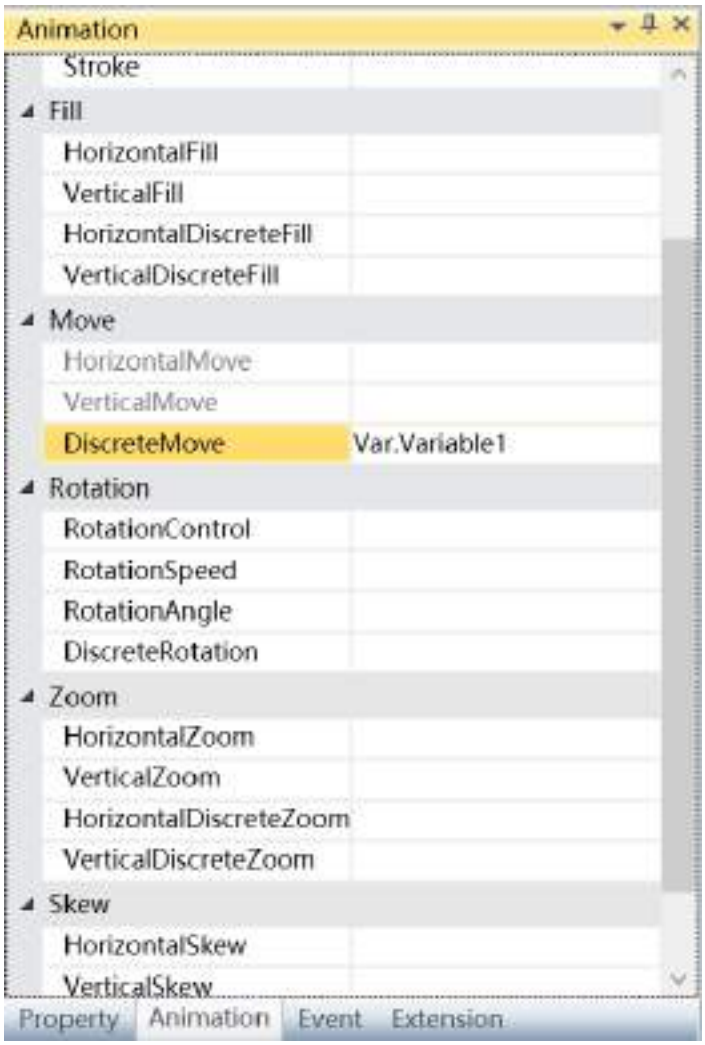
- Coordinate: Sets the correspondingposition coordinates.

- “Add” button: Press this button to add configuration items, as shown in the figure below:



- “Delete” button: Pressing this button after selecting a certain configuration item will delete that configuration item.

Step 3: When configuration is complete, press the “OK” button to complete the configuration of the “DiscreteMove” animation; the animation configuration of this graphics will be displayed in the “Animation” window, as shown in the figure below:



To delete the animation, simply select the animation in the animation list and then right-click the mouse and select “Delete”.

9.7 Visibility animation

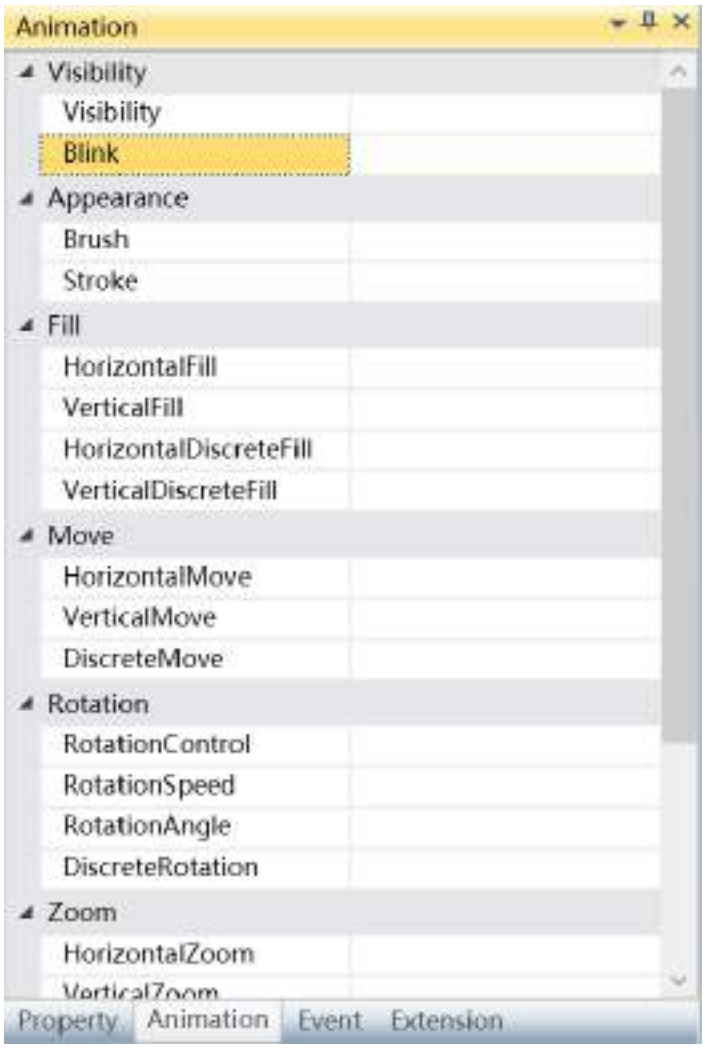
Visibility animation refers to controlling the visibility of the graphic object through variables or expression values and changing the “Visibility” property of the graphic object.

According to the visibility effect, visibility animations are divided into two types: “Visibility” and “Blink”; the difference between them are: Blinking makes the graphic switch between display and hide according to a fixed frequency in order to achieve blinking effects; visibility changes the graphic from display to hide or from hide to display according to the conditions; it is only a single action.

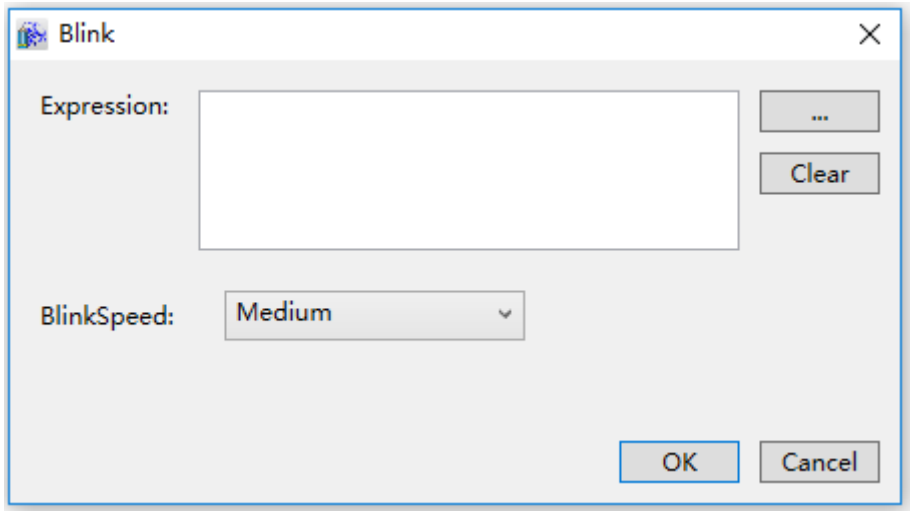
1. Steps to configure blinking animation are as follows:

Step 1: Open the project window where the animation needs to be configured under the DIAView

software development environment → select the graphic object to configure animation in the sketchpad → open the “Animation” window → select “Blink” → click the button in the “Blink” bar, as shown in the figure below:



Step 2: The blink animation configuration window will appear, as shown in the figure below:



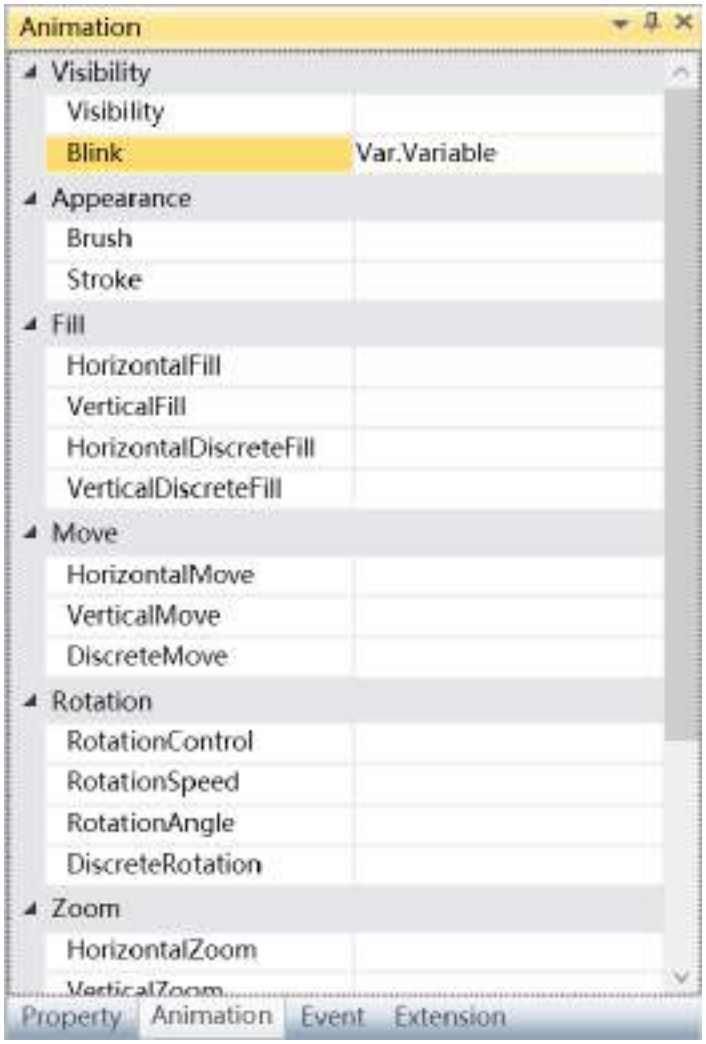
The meanings of each setting in the configuration window are as follows:

Expression: Enter the variable name or expression; you can also click the ... button to open the variable browser to select a variable, the “Clear” button can clear the contents in the input box.

BlinkSpeed: Sets the Blink frequency; it is a drop-down menu with 3 options:

- Slow
- Medium
- Fast

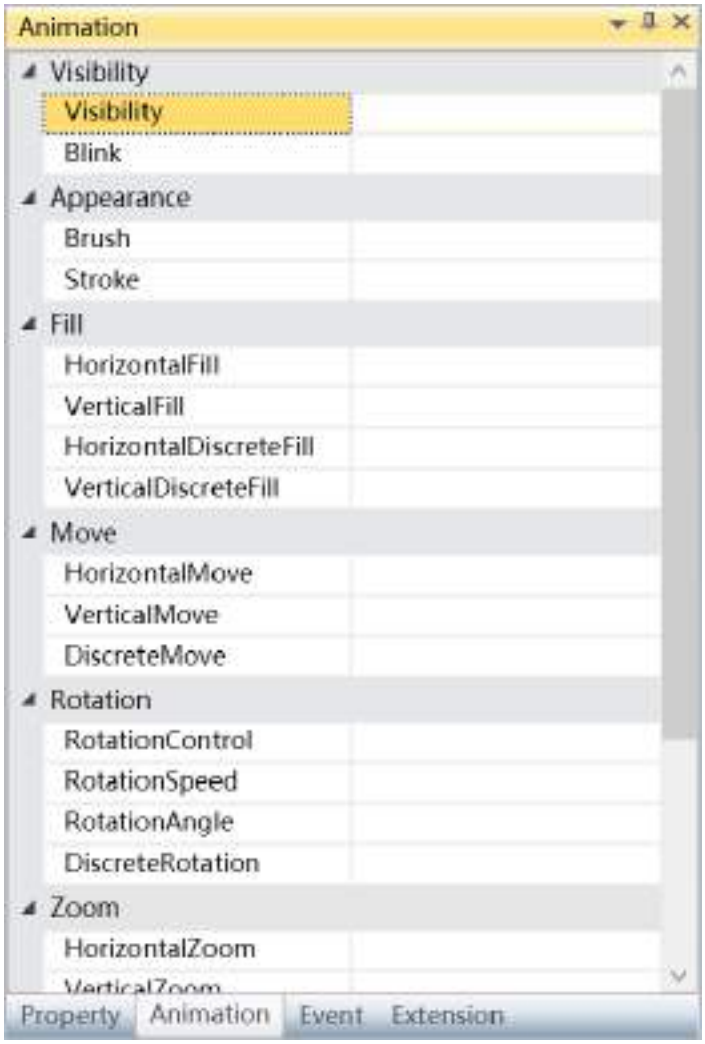
Step 3: When configuration is complete, press the “OK” button to complete the configuration of the “Blink” animation; the animation configuration of this graphics will be displayed in the “Animation” window, as shown in the figure below:



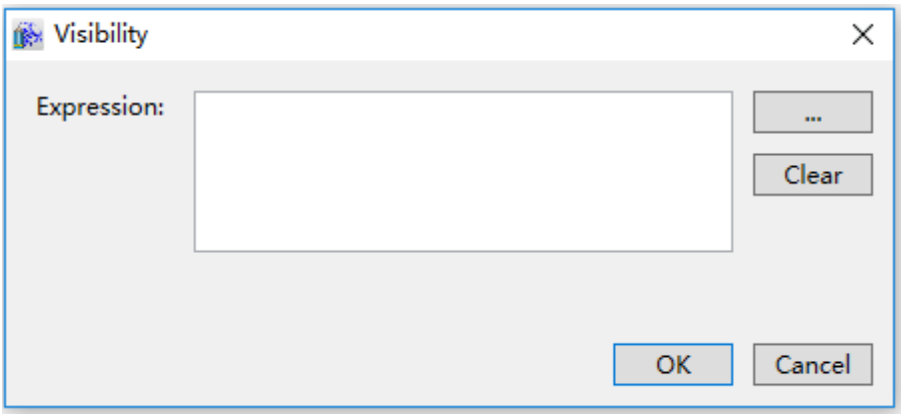
To delete the animation, simply select the animation in the animation list and then right-click the mouse and select “Delete”.

2. Steps to configure visibility animation are as follows:

Step 1: Open the project window where the animation needs to be configured under the DIAView software development environment → select the graphic object to configure animation in the sketchpad → open the “Animation” window → select “Visibility”→ click the button in the “Visibility” bar, as shown in the figure below:



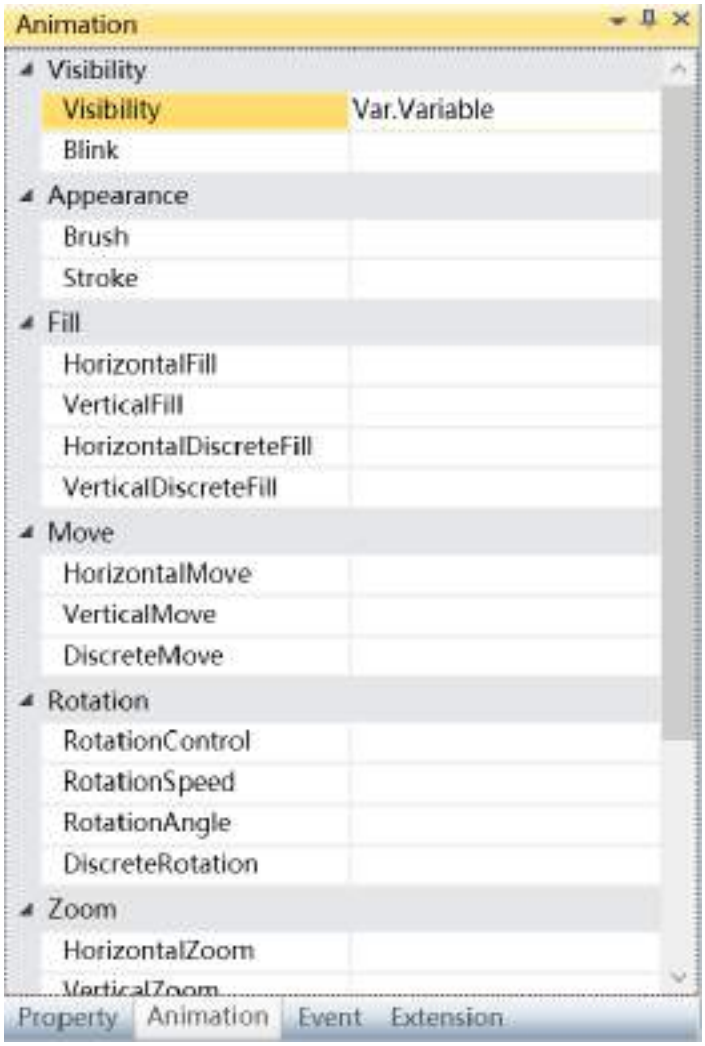
Step 2: The visibility animation configuration window will appear, as shown in the figure below:



The meanings of each setting in the configuration window are as follows:

Expression: Enter the variable name or expression; you can also click the  button to open the variable browser to select a variable; the “Clear” button can clear the contents in the input box.

Step 3: When configuration is complete, press the “OK” button to complete the configuration of the “Visibility” animation; the animation configuration of this graphics will be displayed in the “Animation” window, as shown in the figure below



To delete the animation, simply select the animation in the animation list and then right-click the mouse and select “Delete”.

9.8 Flowing animation

Flowing animation can only be configured for the “Pipe” graphic object, which is changing the “Liquid” flow property of the pipe.

Flowing animation are divided into two types: “LineFlow” and “FlowControl”; they cannot be configured at the same time .

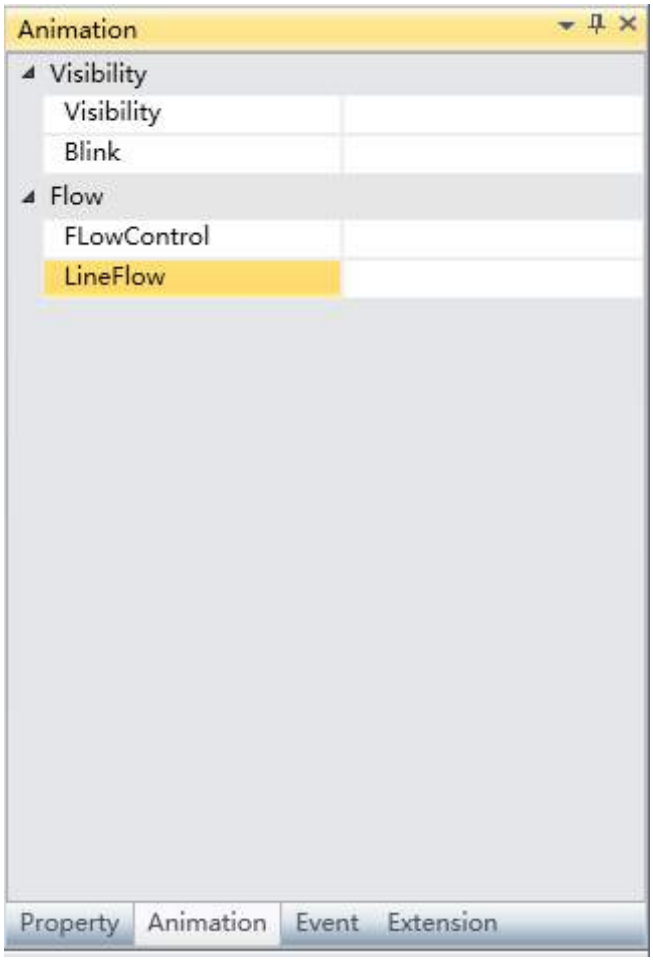
“LineFlow” refers to setting a variable or expression and comparing them with the default value.When it matches the comparison conditions , the liquid in the pipe will flow according configured flowing speed ;

“FlowControl” refers to using whether expressions or discrete variable values are true or false to

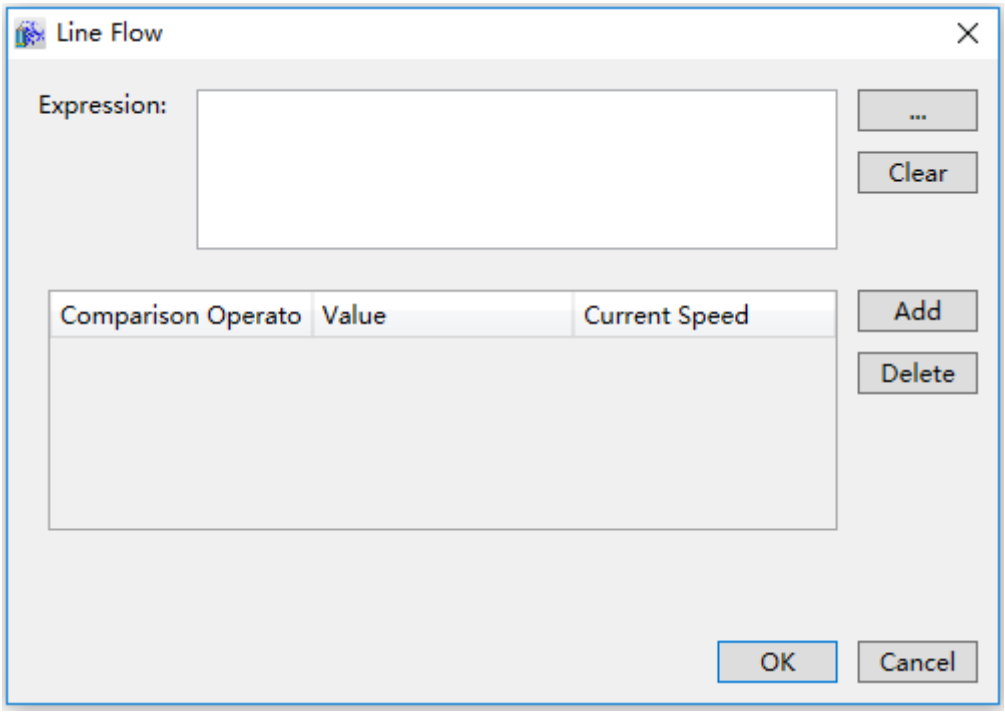
control whether the liquid in the pipe will flow; the flow speed can also be set at the same time .

1. Steps to configure line flow animation are as follows:

Step 1: Open the project window where the animation needs to be configured under the DIAView software development environment → select the graphic object to configure animation in the sketchpad → open the “Animation” window → select “LineFlow”→ click the button in the “LineFlow” bar, as shown in the figure below:



Step 2: The line flow animation configuration window will appear, as shown in the figure below:



The meanings of each setting in the configuration window are as follows:

Expression: Enter the variable name or expression; you can also click the  button to open the variable browser to select a variable; the “Clear” button can clear the contents in the input box.

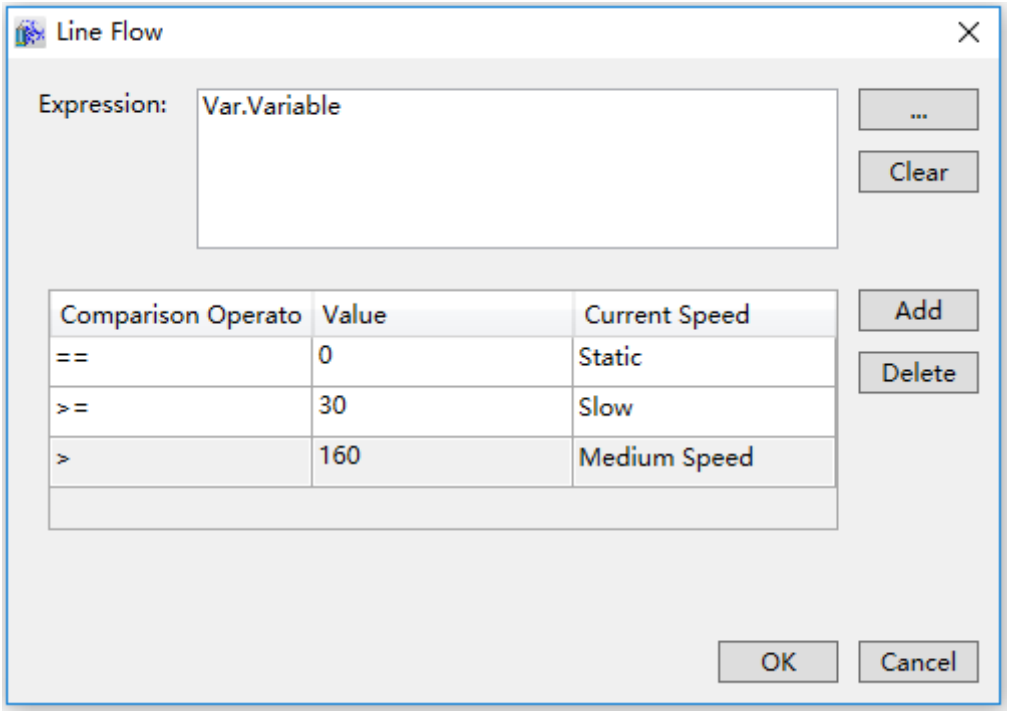
Filling configuration:

➢ Comparison Operators: Sets the comparison operator symbol between the “Expression” and “Value”; it is a drop-down menu that includes 6 options:

- <: Less than
- <=: Less than or equal to
- =: Equal to (preset)
- !=: Not equal to
- >=: Greater than or equal to
- >: Greater than

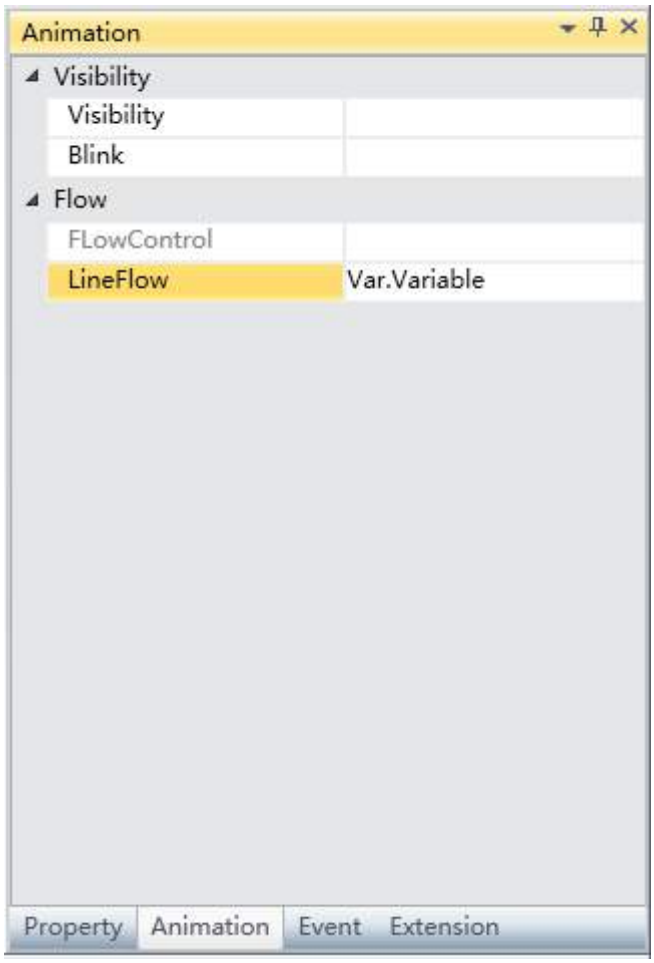
➢ Value: Sets the reference value for the “Expression” to compare to; it can be an integer or decimal .

- Current Speed: Sets the flowing speed of the liquid; it is a drop-down menu that includes 6 options
  - Static
  - Slow
  - Medium Speed
  - Fast
- "Add" button: Press this button to add configuration items, as shown in the figure below:



- "Delete" button: Pressing this button after selecting a certain configuration item will delete that configuration item.

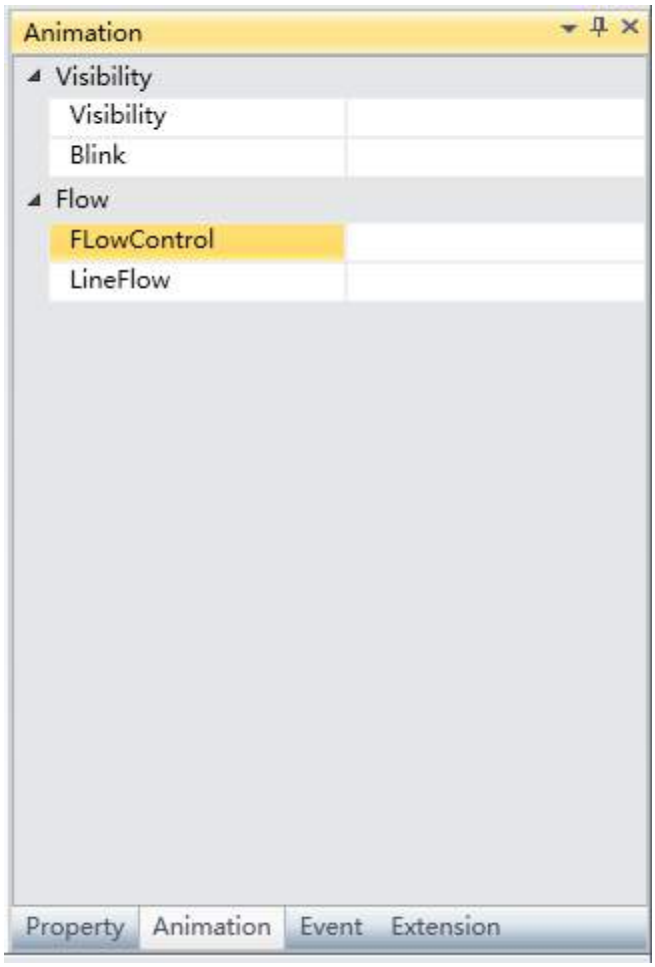
Step 3: When configuration is complete, press the "OK" button to complete the configuration of the "LineFlow" animation. The animation configuration of this graphics will be displayed in the "Animation" window, as shown in the figure below:



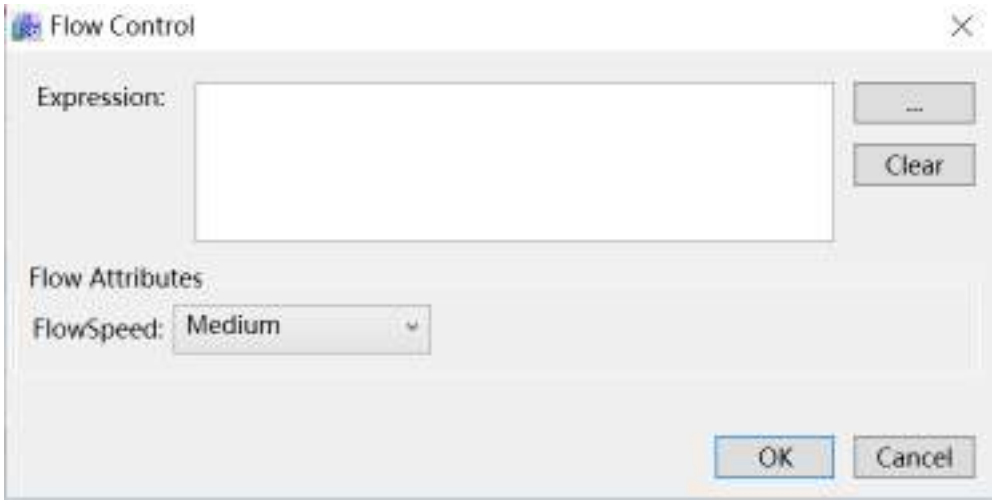
To delete the animation, simply select the animation in the animation list and then right-click the mouse and select “Delete”.

2. Steps to configure flow control animation are as follows:

Step 1: Open the project window where the animation needs to be configured under the DIAView software development environment → select the graphic object to configure animation in the sketchpad → open the “Animation” window → select “FlowControl”→ click the button in the “FlowControl” bar, as shown in the figure below:



Step 2: The flow control animation configuration window will appear, as shown in the figure below:



The meanings of each setting in the configuration window are as follows:

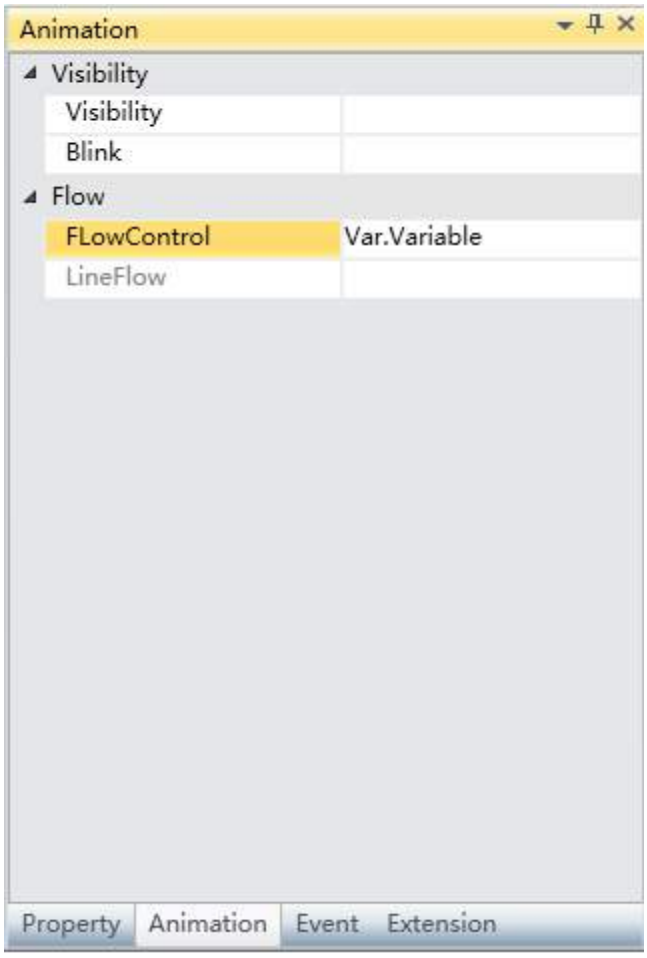
Expression: Enter the variable name or expression; you can also click the 

button to open the variable browser to select a variable; the “Clear” button can clear the contents in the input box.

Flow Attributes: Sets the flowing speed of the liquid; it is a drop-down menu thatincludes 6 options:

- Static
- Slow
- Medium Speed
- Fast

Step 3: When configuration is complete, press the “OK” button to complete the configuration of the “FlowControl” animation. The animation configuration of this graphics will be displayed in the “Animation” window, as shown in the figure below:



To delete the animation, simply select the animation in the animation list and then right-click the mouse and select “Delete”.

9.9 Value display animation

Value display animations can only be configured for “Text”, “TextBox” and “Label” ; changing their “Content” property will allow the text display to change according to the changes in the associated variable value and conditions set.

Value display animations are divided into four types: “TextDisplay”,“AnalogValueDisplay”, “AnalogValueStringDisplay”, “DiscreteValueDisplay” and ; they cannot be configured simultaneously.

“TextDisplay” animation associates a variable directly or by setting strings. It allows the displayed content of the graphic object to be the associated variable value or the setting string.

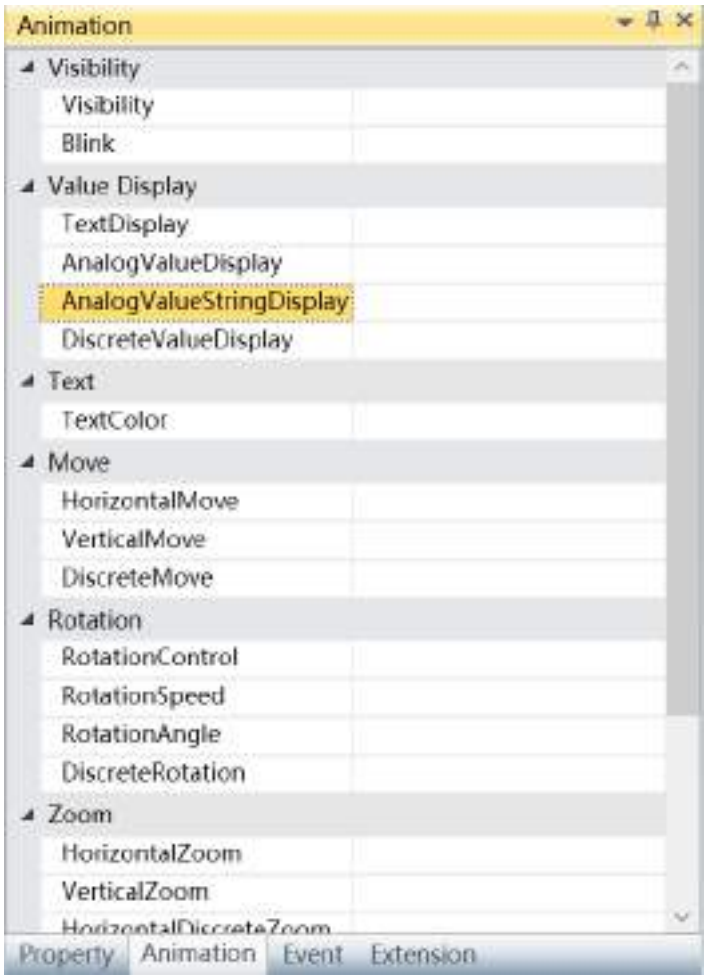
“AnalogValueDisplay” animation associates an analog (integer or real) variable and sets the display format of the value so that the variable value is displayed in the graphic object according to the setting format.

“AnalogValueStringDisplay” animation associates an analog (integer or real) variable and compares the variable value with the setting value . When it matches the comparison conditions, the content displayed in the graphic object is the string value set.

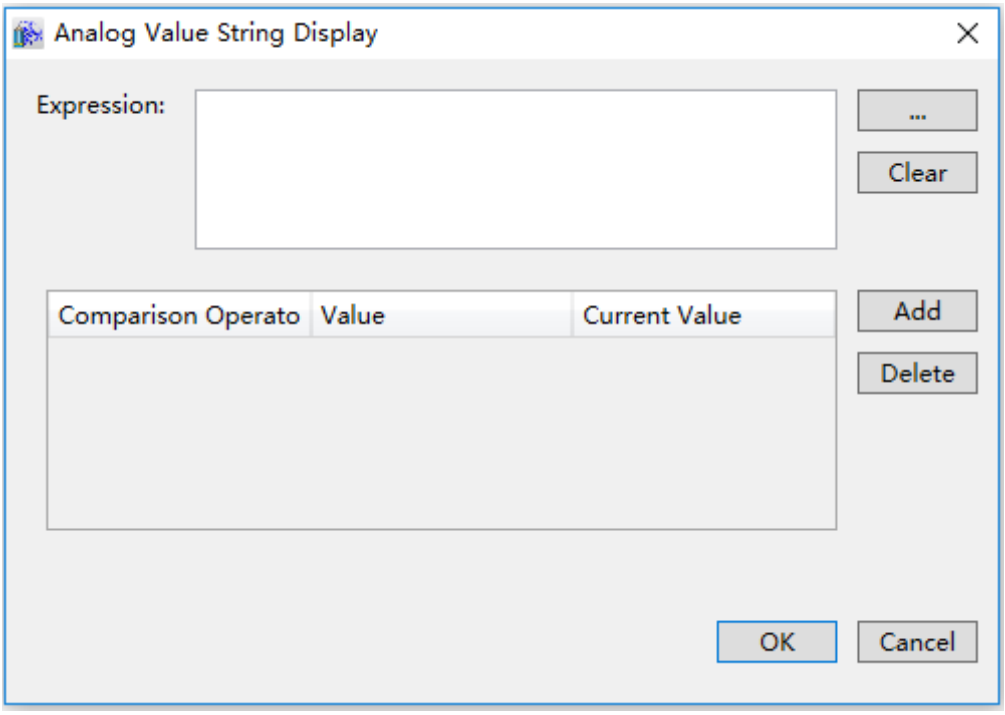
“DiscreteValueDisplay” animation associates a digital (bool) variable; different display contents are set according to the variable value.

1. Steps to configure analog value string display animation are as follows:

Step 1: Open the project window where the animation needs to be configured under the DIAView software development environment → select the graphic object to configure animation in the sketchpad → open the “Animation” window → select “AnalogValueStringDisplay”→ click the button in the “AnalogValueStringDisplay” bar, as shown in the figure below:



Step 2: The analog value string display animation configuration window will appear, as shown in the figure below:



The meanings of each setting in the configuration window are as follows:

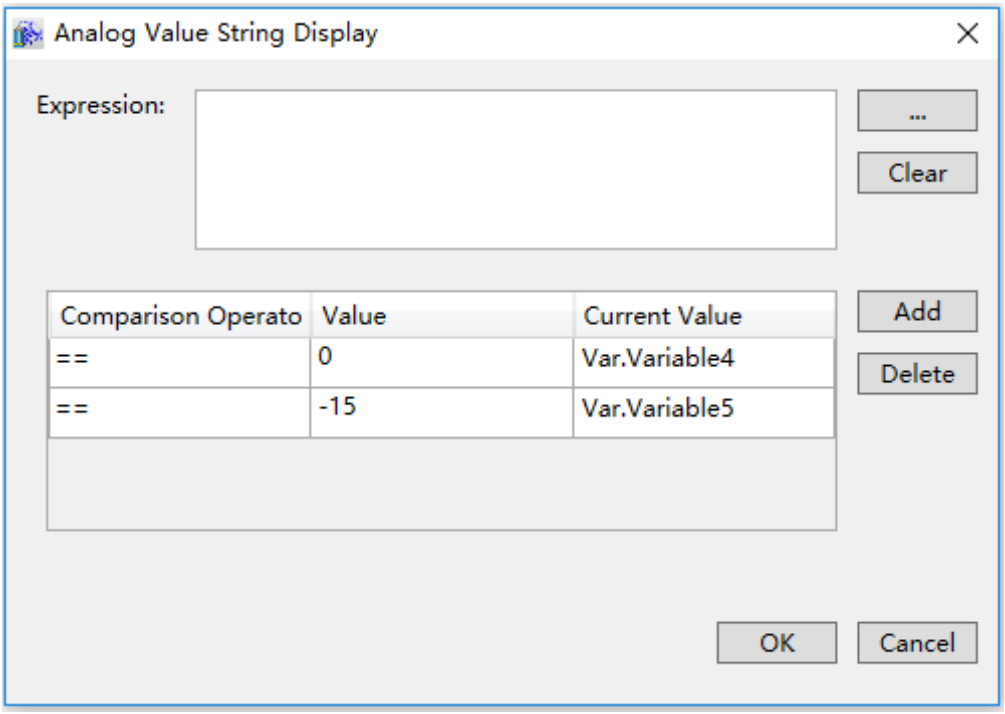
Expression: Input the variable name or expression; you can also click the  button to open the variable browser to select a variable; the “Clear” button can clear the contents in the input box.

Analog value string configuration:

- Comparison Operators: Sets the comparison operator symbol between the “Expression” and “Value”; it is a drop-down menu that includes 6 options:
  - <: Less than
  - <=: Less than or equal to
  - =: Equal to (default)
  - !=: Not equal to
  - >=: Greater than or equal to
  - >: Greater than
- Value: Sets the reference value for the “Expression” to compare to; it can be an integer or decimal.

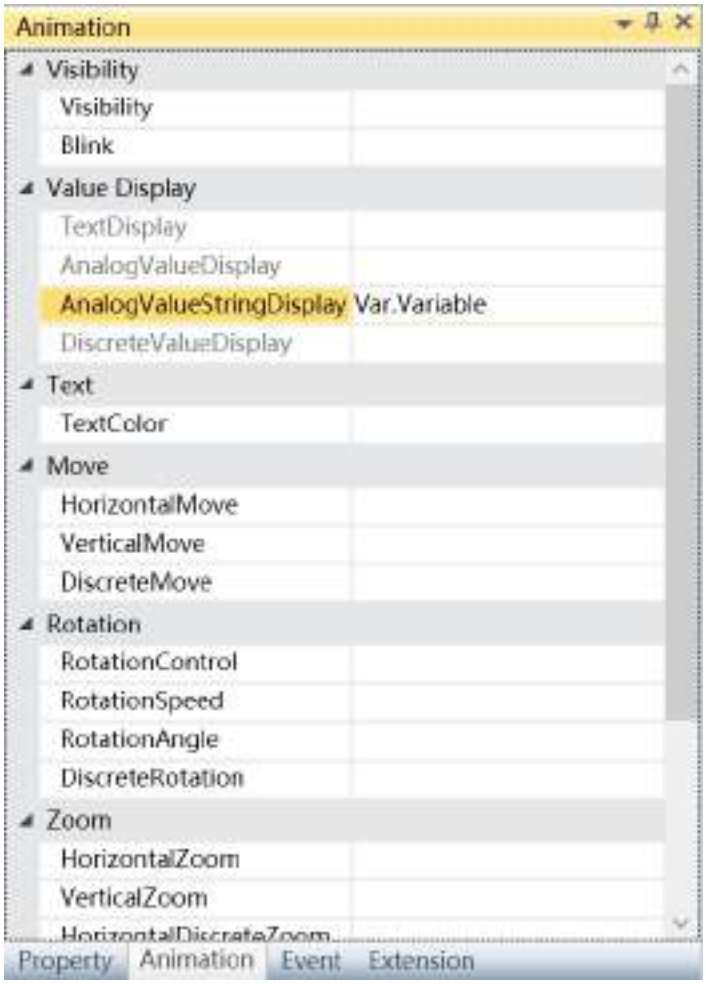
- Current Value: Sets the text content of the string; which the comparison conditions set are met, this value will be displayed in the content of the graphic object.

- "Add" button: Press this button to add a configuration item, as shown in the figure below:



- "Delete" button: Pressing this button after selecting a certain configuration item will delete that configuration item.

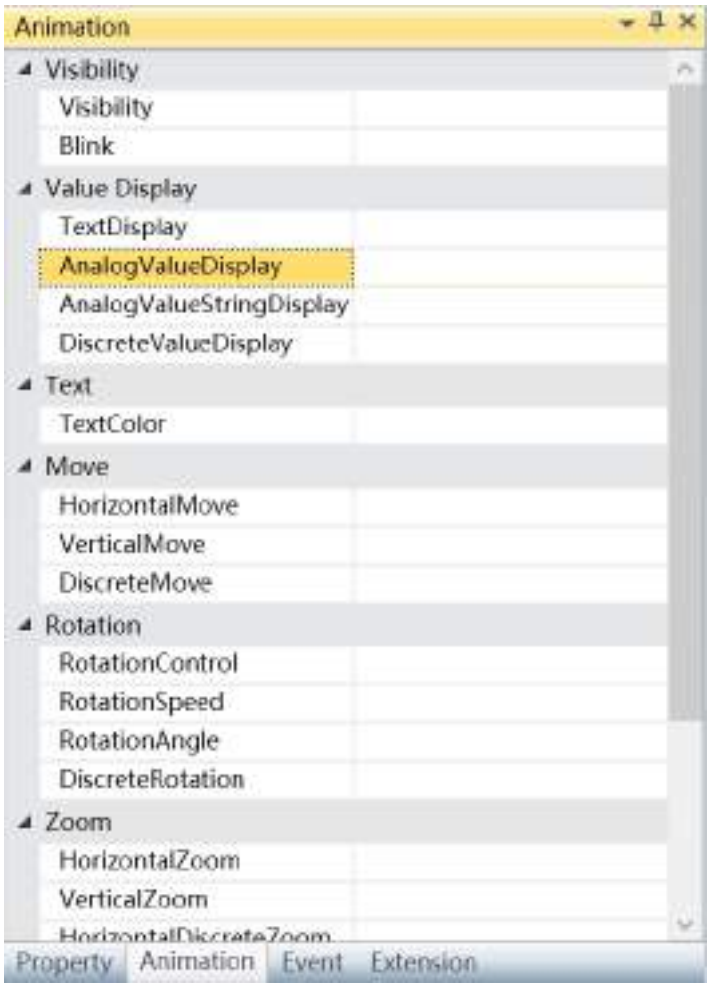
Step 3: When configuration is complete, press the "OK" button to complete the configuration of the "AnalogValueStringDisplay" animation; the animation configuration of this graphics will be displayed in the "Animation" window, as shown in the figure below:



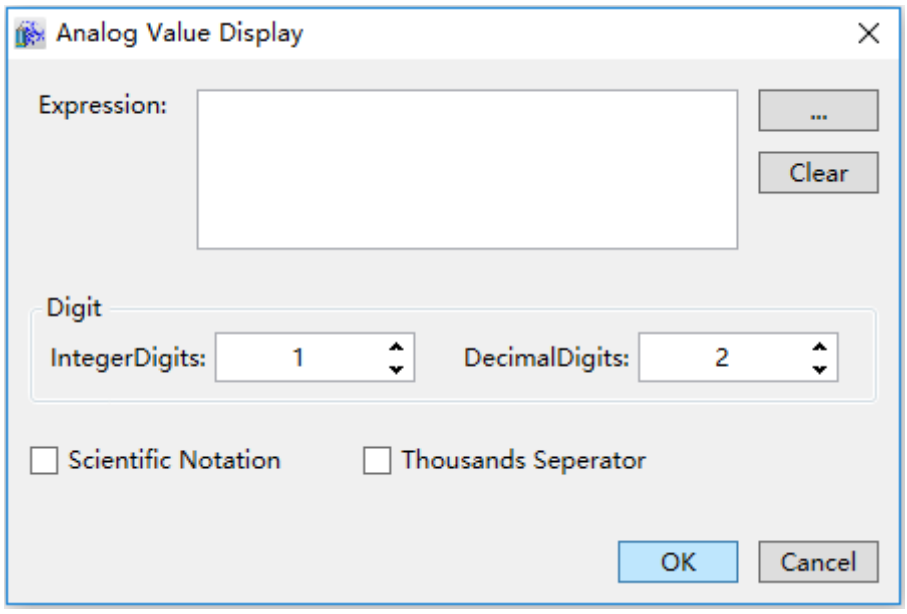
To delete the animation, simply select the animation in the animation list and then right-click the mouse and select “Delete”.

2. Steps to configure analog value display animation are as follows:

Step 1: Open the project window where the animation needs to be configured under the DIAView software development environment → select the graphic object to configure animation in the sketchpad → open the “Animation” window → select “AnalogValueDisplay”→ click the button in the “AnalogValueDisplay” bar, as shown in the figure below:



Step 2: The analog value display animation configuration window will appear, as shown in the figure below:



The meanings of each setting in the configuration window are as follows:

Expression: Input the associated variable name; you can also click the  button to open the variable browser to select a variable; the “Clear” button can clear the contents in the input box.

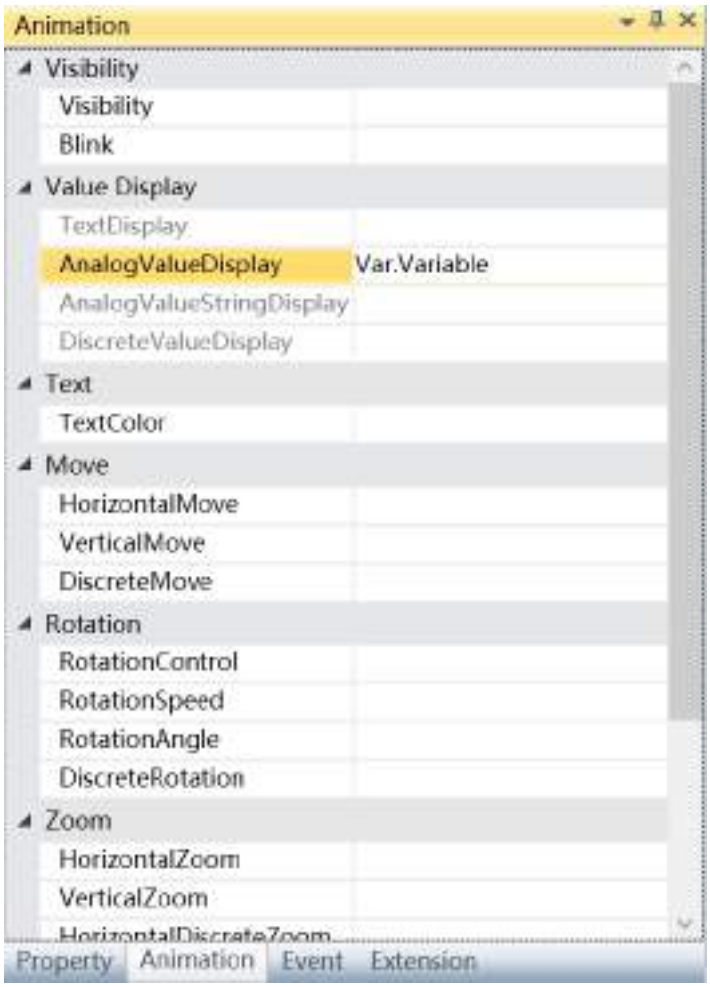
IntegerDigits: Set the number of digits for the integer; the default value is 1. If the number of integer digits of the outputted value is less than the value set here, a zero will be added to the highest digit. If it is greater than the value set here, it will output according to the actual number of digits there are Setting it to 0 means not adding a zero to the displayed output value. For example: Of the number of digits of the integer is set as 3 and the output value is 10, it will display 010; and if the output value is 165, it will display 165.

DecimalDigits: Sets the number of decimal numbers; the default value is 2. If the number of decimal places of the outputted value is less than the value set here, a zero will be added behind the lowest digit. If it is greater than the value set here, it will be rounded off to the setting number of decimal places. Moreover, 1 will be added if greater than 5 and eliminated if less than 5. Setting it to 0 means do not display decimal places. Decimal places are set for all associated engineering variables; they will not affect the settings here. For example: When the number of decimal places is set to 2 and the output value is 2.3, it will display 2.30; if the output value is 0.125, it will display 0.12; if the output value is 0.126, it will display 0.13.

Scientific Notation: Sets whether to use scientific notation to display the variable value.

Thousand Separator: Sets whether to use the thousand seperator.

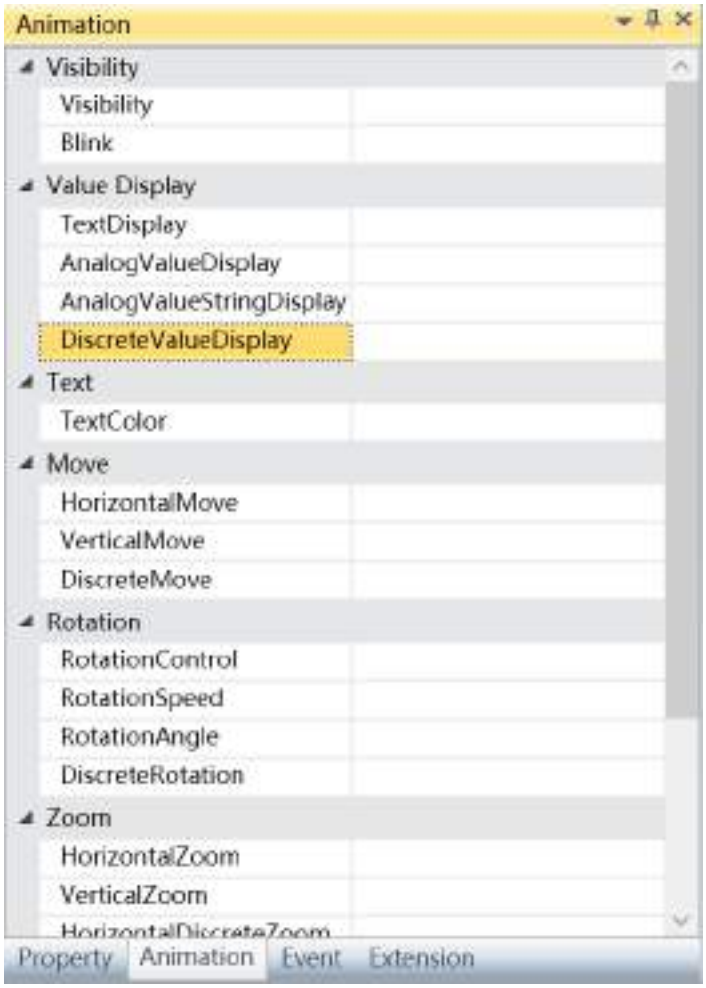
Step 3: When configuration is complete, press the “OK” button to complete the configuration of the “AnalogValueDisplay” animation; the animation configuration of this graphics will be displayed in the “Animation” window, as shown in the figure below:



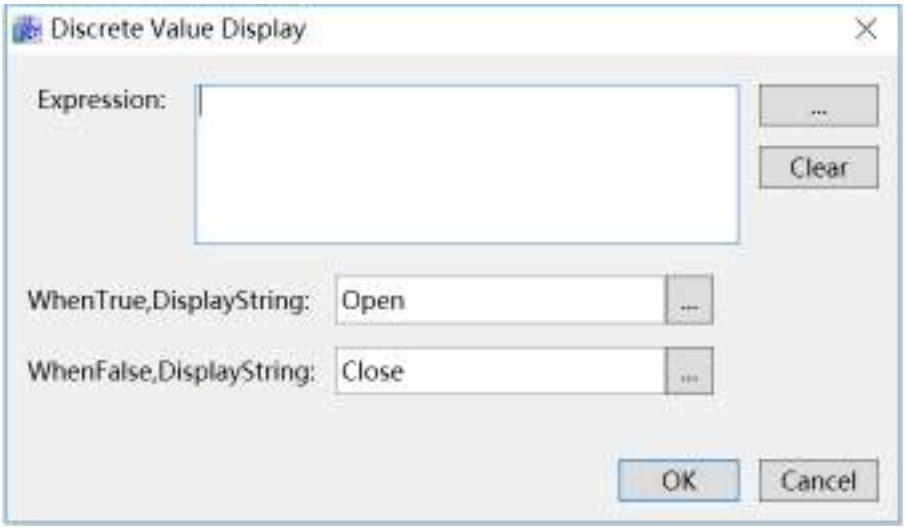
To delete the animation, simply select the animation in the animation list and then right-click the mouse and select “Delete”.

3. Steps to configure discrete value display animation are as follows:

Step 1: Open the project window where the animation needs to be configured under the DIAView software development environment → select the graphic object to configure animation in the sketchpad → open the “Animation” window → select “DiscreteValueDisplay”→ click the button in the “DiscreteValueDisplay” bar , as shown in the figure below:



Step 2: The discrete value display animation configuration window will appear, as shown in the figure below:



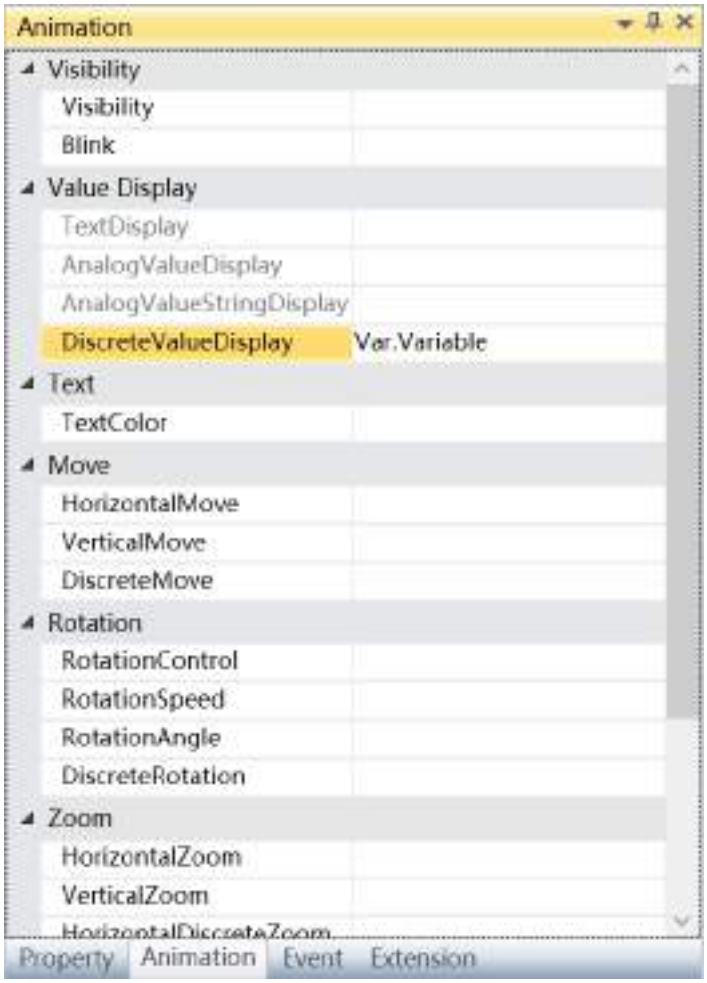
The meanings of each setting in the configuration window are as follows:

Expression: Input the associated variable name or expression; you can also click the ... button to open the variable browser to select a variable; the “Clear” button can clear the contents in the input box.

WhenTrue,DisplayString: Sets the display content of the graphic object when the expression is true，support for multilingual functionality.

WhenFalse,DisplayString: Sets the display content of the graphic object when the expression is false，support for multilingual functionality.

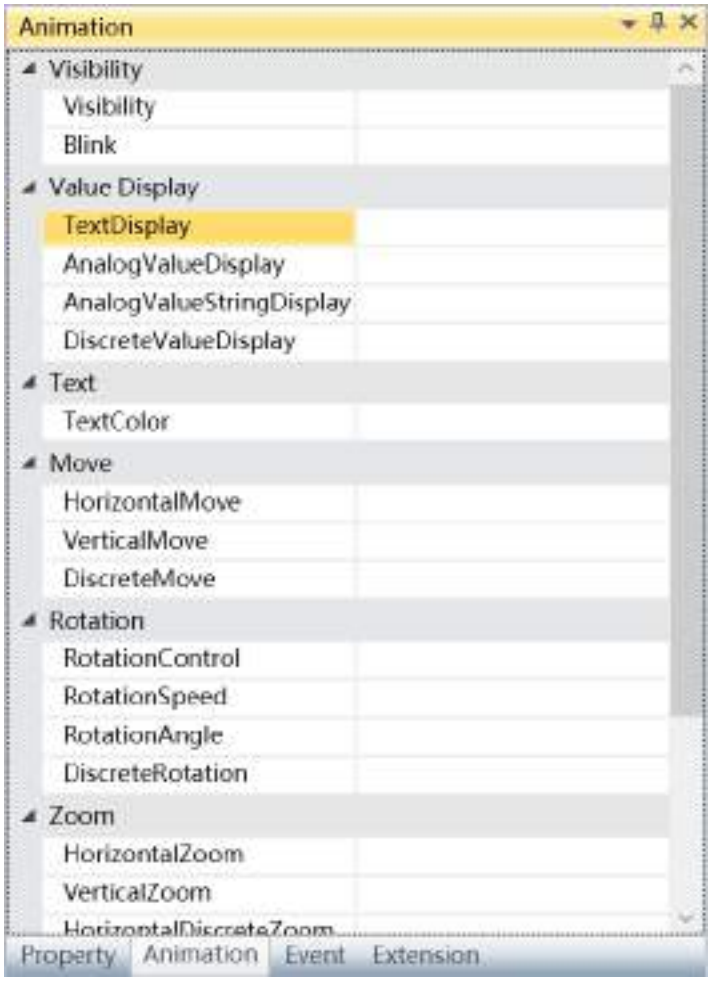
Step 3: When the configuration is complete, press the “OK” button to complete the configuration of the “DiscreteValueDisplay” animation. The animation configuration of these graphics will be displayed in the “Animation” window, as shown in the figure below:



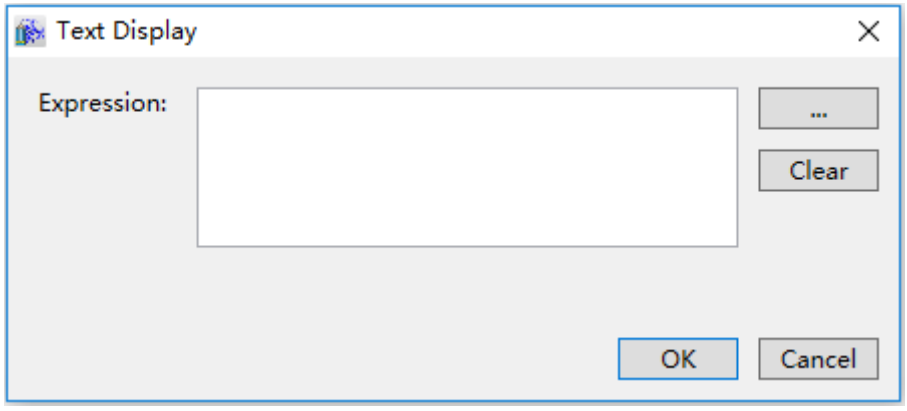
To delete the animation, simply select the animation in the animation list and then right-click the mouse and select “Delete”.

4. Steps to configure text display animation are as follows:

Step 1: Open the project window where the animation needs to be configured under the DIAView software development environment → select the graphic object to configure animation in the sketchpad → open the “Animation” window → select “TextDisplay”→ click the button in the “TextDisplay” bar , as shown in the figure below:



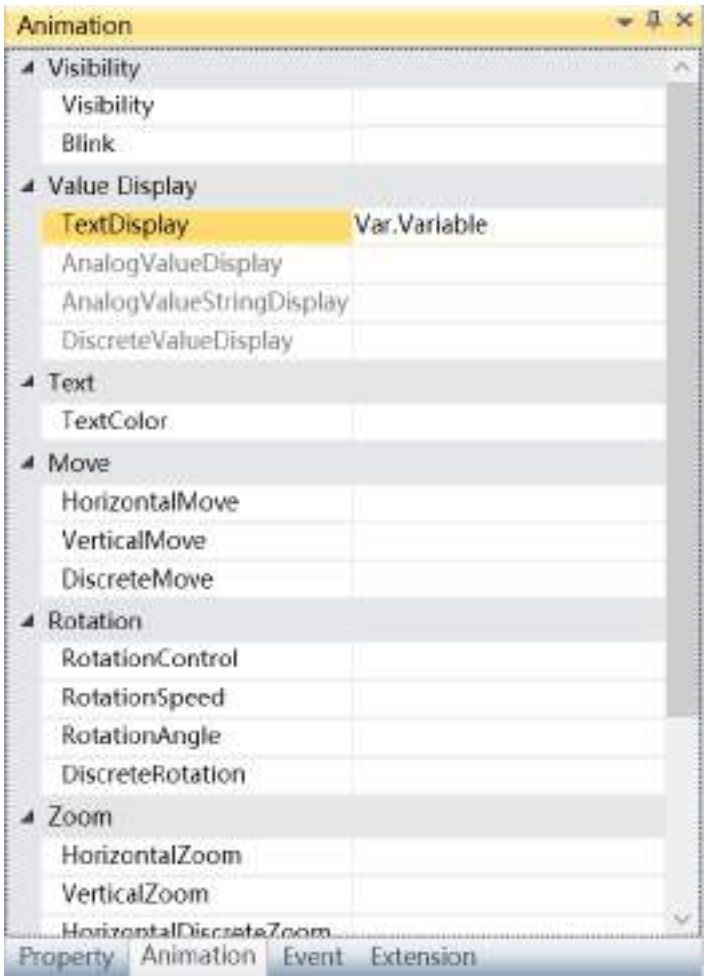
Step 2: The text display animation configuration window will appear, as shown in the figure below:



The meanings of each setting in the configuration window are as follows:

Expression: Input the associated variable name or string value; you can also click the ... button to open the variable browser to select a variable; the “Clear” button can clear the contents in the input box.

Step 3: When configuration is complete, press the “OK” button to complete the configuration of the “TextDisplay” animation. The animation configuration of this graphics will be displayed in the “Animation” window, as shown in the figure below:



● Text display animation can not only display single variable value but also can recycle displaying all the alarm information of associated alarm group.

● To delete the animation, simply select the animation in the animation list and then right-click the mouse and select “Delete”.

9.10 Skew animation

Skew animation refers to controlling the slope level of the graphic object through variables or expression values and changing the “Skew” property of the graphic object.

Slope animation is divided into four types: “HorizontalSkew”, “VerticalSkew” ,“HorizontalDiscreteSkew” and “VerticalDiscreteSkew”.

“HorizontalSkew” animation refers to the slope level of the graphic object in the horizontal direction.

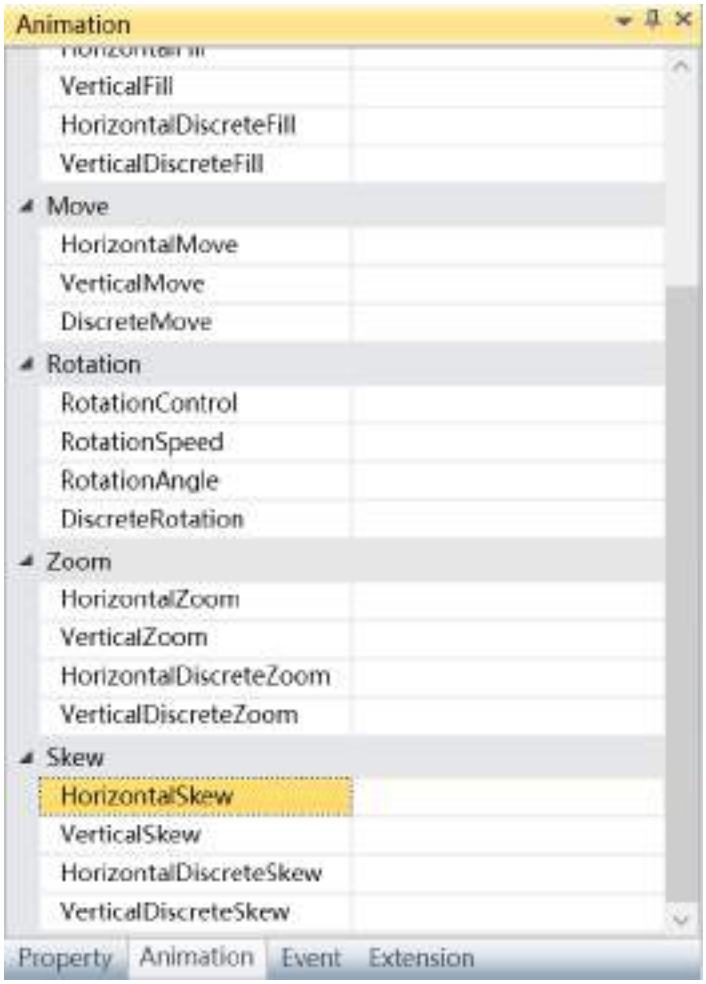
“VerticalSkew” animation refers to the slope level of the graphic object in the vertical direction.

“HorizontalDiscreteSkew” means that the horizontal slope angle of the graphic object is related to the discrete variable.

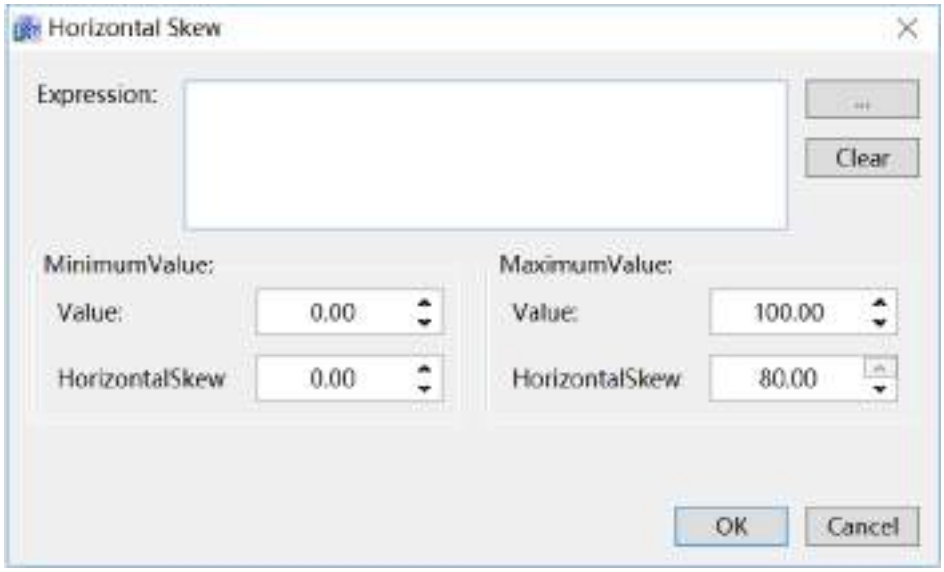
“VerticalDiscreteSkew” means that the vertical slope angle of the graphic object is related to the discrete variable.

1. The configuration process is as follows,take the animation of "horizontal Skew" and "horizontal discrete Skew" as examples:

Step 1: Open the project window where the animation needs to be configured under the DIAView software development environment → select the graphic object to configure animation in the sketchpad → open the “Animation” window → select “HorizontalSkew”→ click the button in the “HorizontalSkew” bar, as shown in the figure below:



Step 2: The horizontal skew animation configuration window will appear, as shown in the figure below:



The meanings of each setting in the configuration window are as follows:

Expression: Input the variable name or expression; you can also click the ... button to open the variable browser to select a variable; the “Clear” button can clear the contents in the input box.

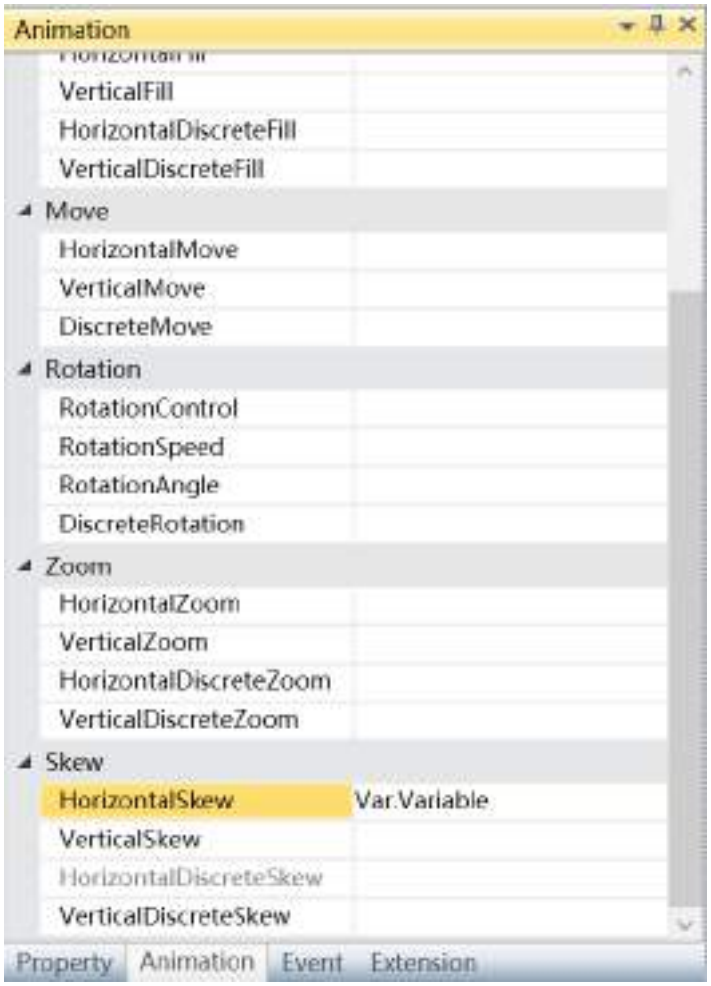
MinimumValue:

- Value: Sets the minimum value of the “Expression”
- HorizontalSkew: Sets the minimum value for the Horizontal slope

MaximumValue:

- Value: Sets the maximum value of the “Expression”
- HorizontalSkew: Sets the maximum value for the Horizontal slope

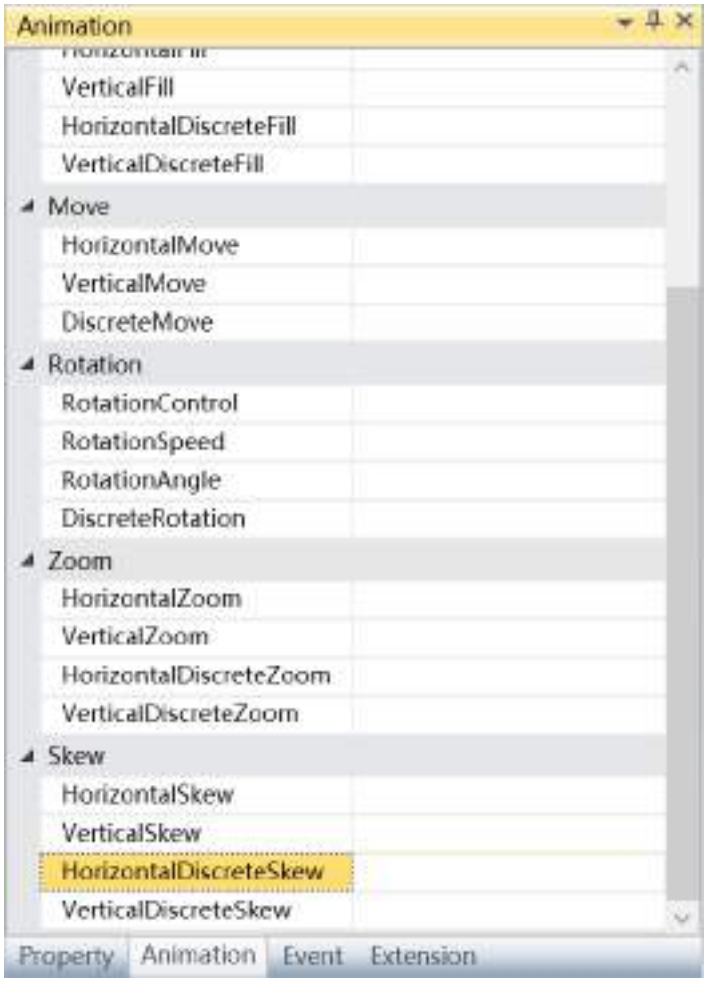
Step 3: When configuration is complete, press the “OK” button to complete the configuration of the “HorizontalSkew” animation; the animation configuration of this graphics will be displayed in the “Animation” window, as shown in the figure below:



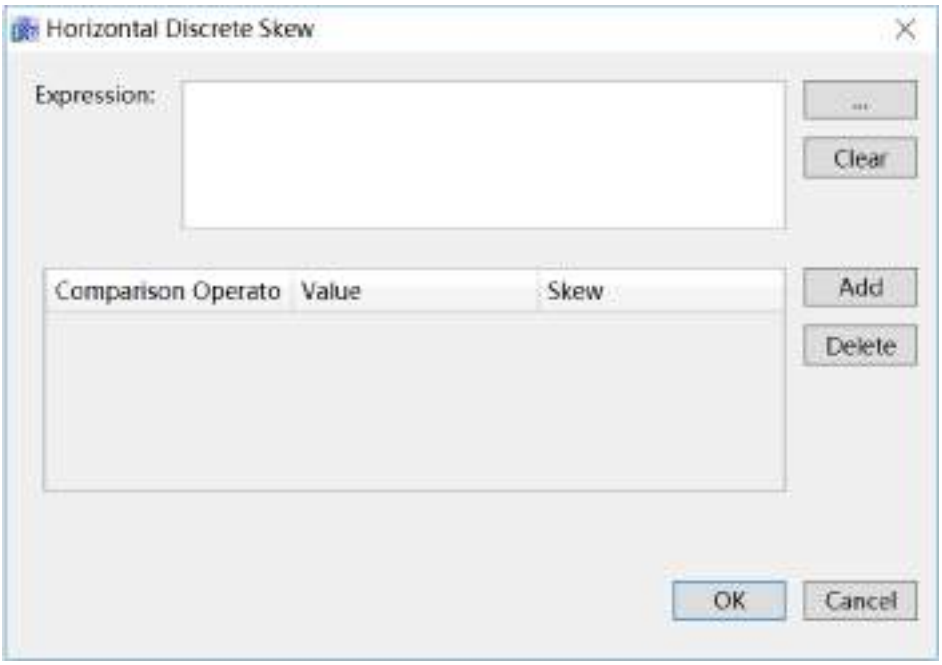
To delete the animation, simply select the animation in the animation list and then right-click the mouse and select “Delete”.

2. Steps to configure the Horizontal discrete slope animation are as follows:

Step 1: Open the project window where the animation needs to be configured under the DIAView software development environment → select the graphic object to configure animation in the sketchpad → open the “Animation” window → select “HorizontalDiscreteSkew”→ click the button in the “HorizontalDiscreteSkew” fbar, as shown in the figure below:



Step 2: The horizontal discrete skew animation configuration window will appear, as shown in the figure below:



The meanings of each setting in the configuration window are as follows:

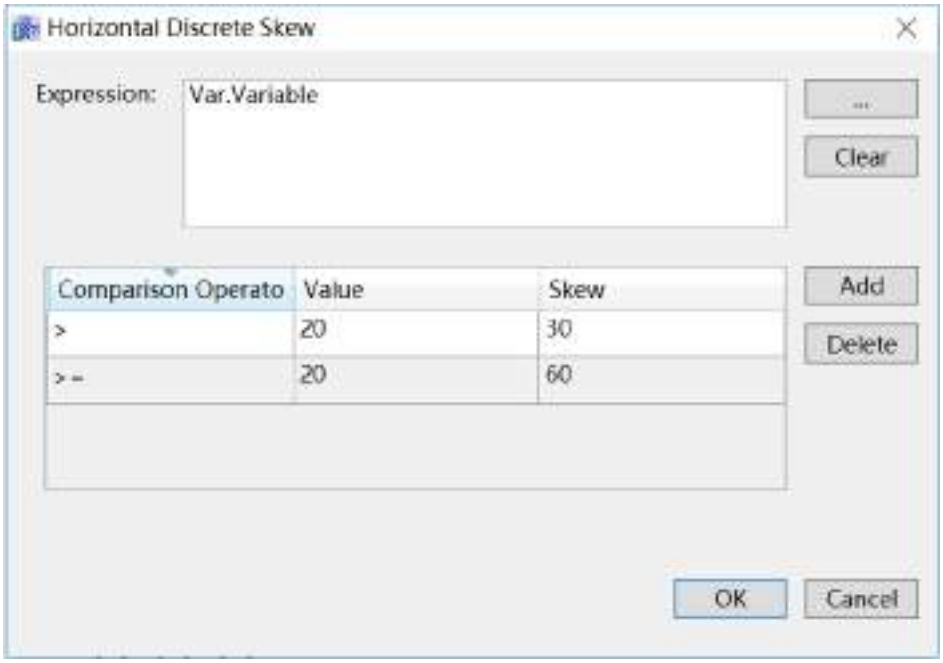
Expression: Input the variable name or expression; you can also click the ... button to open the variable browser to select a variable; the “Clear” button can clear the contents in the input box.

Filling configuration:

- Comparison Operators: Sets the comparison operator symbol between the “Expression” and “Value”; it is a drop-down menu that includes 6 options:
  - <: Less than
  - <=: Less than or equal to
  - =: Equal to (preset)
  - !=: Not equal to
  - >=: Greater than or equal to
  - >: Greater than
- Value: Sets the reference value for the “Expression” to compare to; it can be an integer or decimal.

- Skew: Sets the corresponding slope level.

- “Add”button: Press this button to add configuration items, as shown in the figure below:



- “Delete” button: Pressing this button after selecting a certain configuration item will delete that configuration item.

Step 3: When configuration is complete, press the “OK” button to complete the configuration of the “HorizontalDiscreteSkew” animation. The animation configuration of this graphics will be displayed in the “Animation” window, as shown in the figure below:

To delete the animation, simply select the animation in the animation list and then right-click the mouse band select “Delete”.

9.11 Text animation

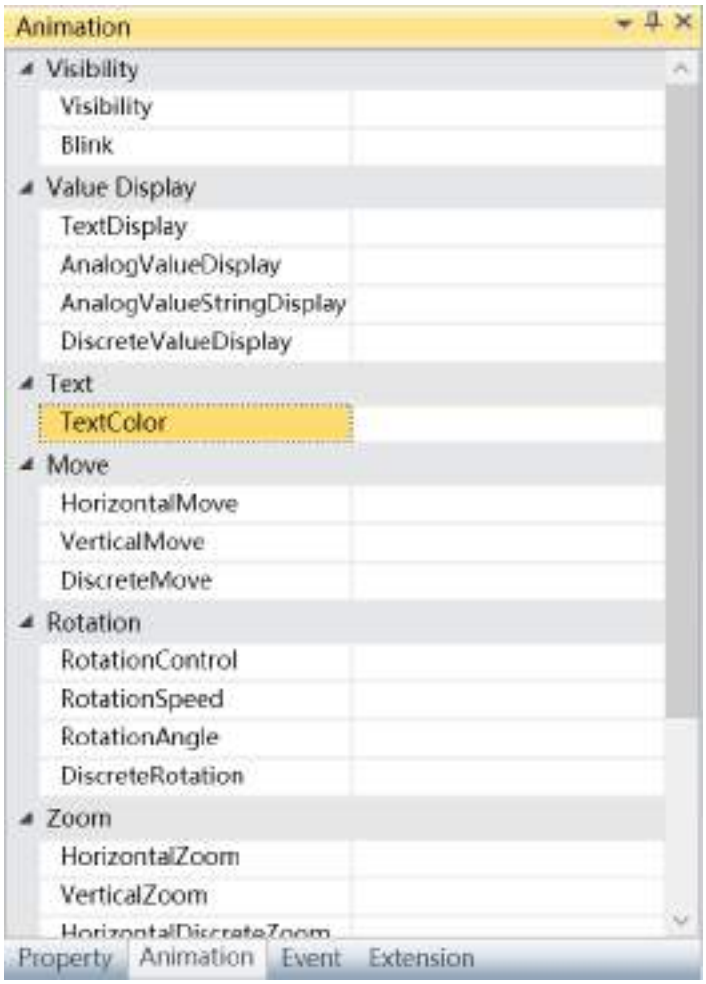
Text animation is to control the forecolor of object by variable or expression value to make the text color blink alternately.

1.Steps to configure“TextColor”animation are as follows:

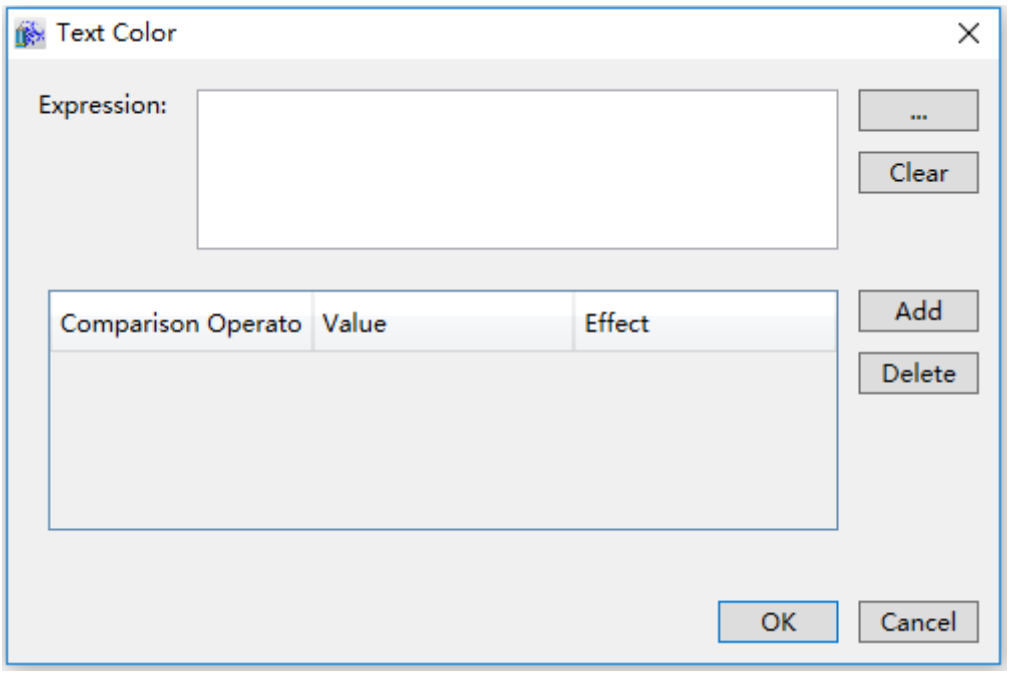
Step 1:

Open the project window where the animation needs to be configured under the DIAView software development environment → select the graphic object to configure animation in the sketchpad → open the “Animation” window → select “TextColor”→ click the button in the “TextColor” bar, as shown in the

figure below:



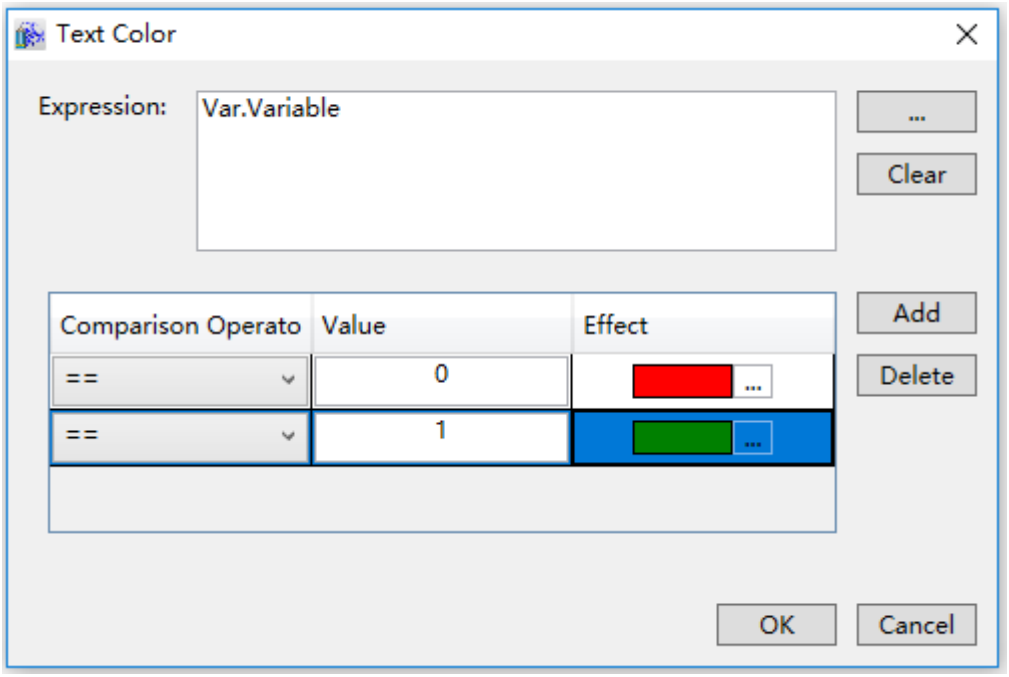
Step 2: The text color animation configuration window will appear, as shown in the figure below:



The meanings of each setting in the configuration window are as follows:

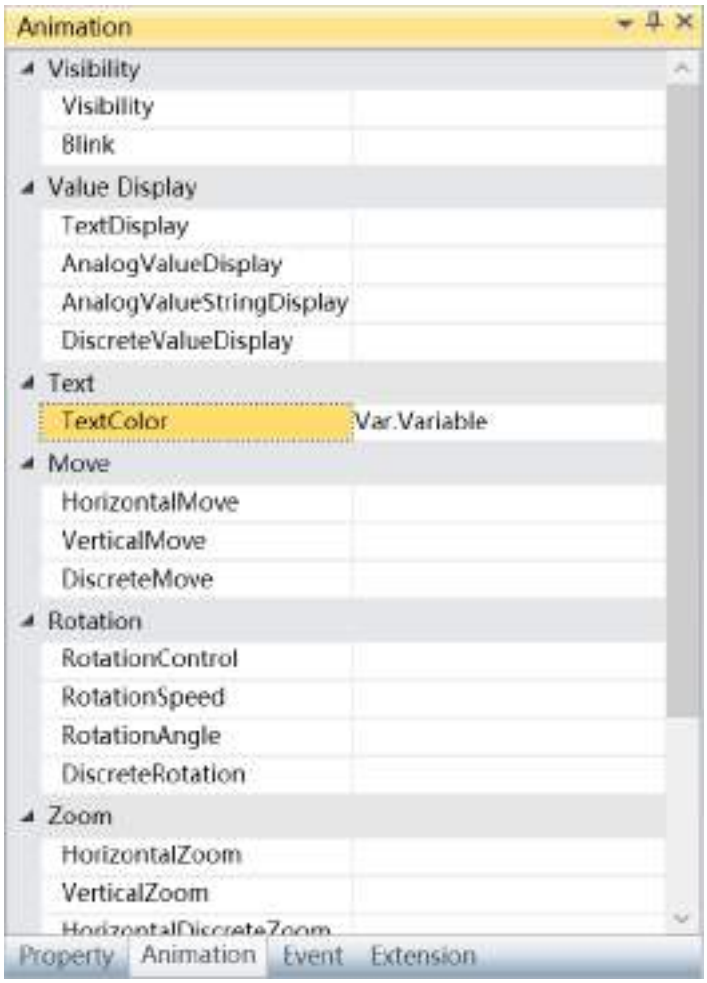
Expression: Input the variable name or expression; you can also click the button to open the variable browser to select a variable; the “Clear” button can clear the contents in the input box.

- Comparison Operators: Sets the comparison operator symbol between the “Expression” and “Value”; it is a drop-down menu that includes 6 options:
  - <: Less than
  - <=: Less than or equal to
  - =: Equal to (preset)
  - !=: Not equal to
  - >=: Greater than or equal to
  - >: Greater than
- Value: Sets the reference value for the “Expression” to compare to; it can be an integer or decimal.
- Effect: set color effect.
- “Add” button: Press this button to add configuration items, as shown in the figure below:



➢ “Delete” button: Pressing this button after selecting a certain configuration item will delete that configuration item.

Step 3: When configuration is complete, press the “OK” button to complete the configuration of the “TextColor” animation. The animation configuration of this graphics will be displayed in the “Animation” window, as shown in the figure below:



To delete the animation, simply select the animation in the animation list and then right-click the mouse and select “Delete”.

10. Event

10.1 Overview

Events are operations that can be identified and responded by graphic objects; they are divided into system events and user events. Events in the DIAView software are usually user events,which are operations that users performed to various graphic objects in the window, which further drives the graphic control to execute certain functions.

Events in the DIAView software refers to the related script handler that are triggered when the user performs operations to the graphic object with the mouse and keyboard etc.; therefore, event configuration is mainly using script editors to configure script programs, or use packaged programs: the script editor of DIAView includes many written commands and system functions that users can use directly in the script program

The DIAView software has comprehensive event operating functions; events include mouse and form operations and value input etc. Each kind of event can only be configured once to the same corresponding graphic object.

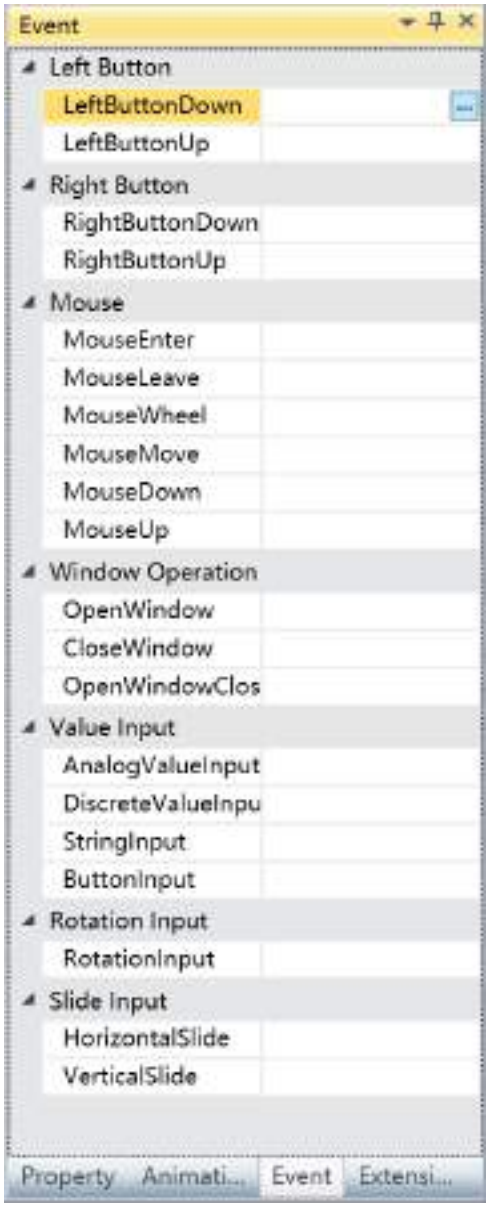
10.2 Left button event

Left Button Event is an event triggered by performing operations with clicking the left mouse button, which will further use related script handlers to complete related functions.

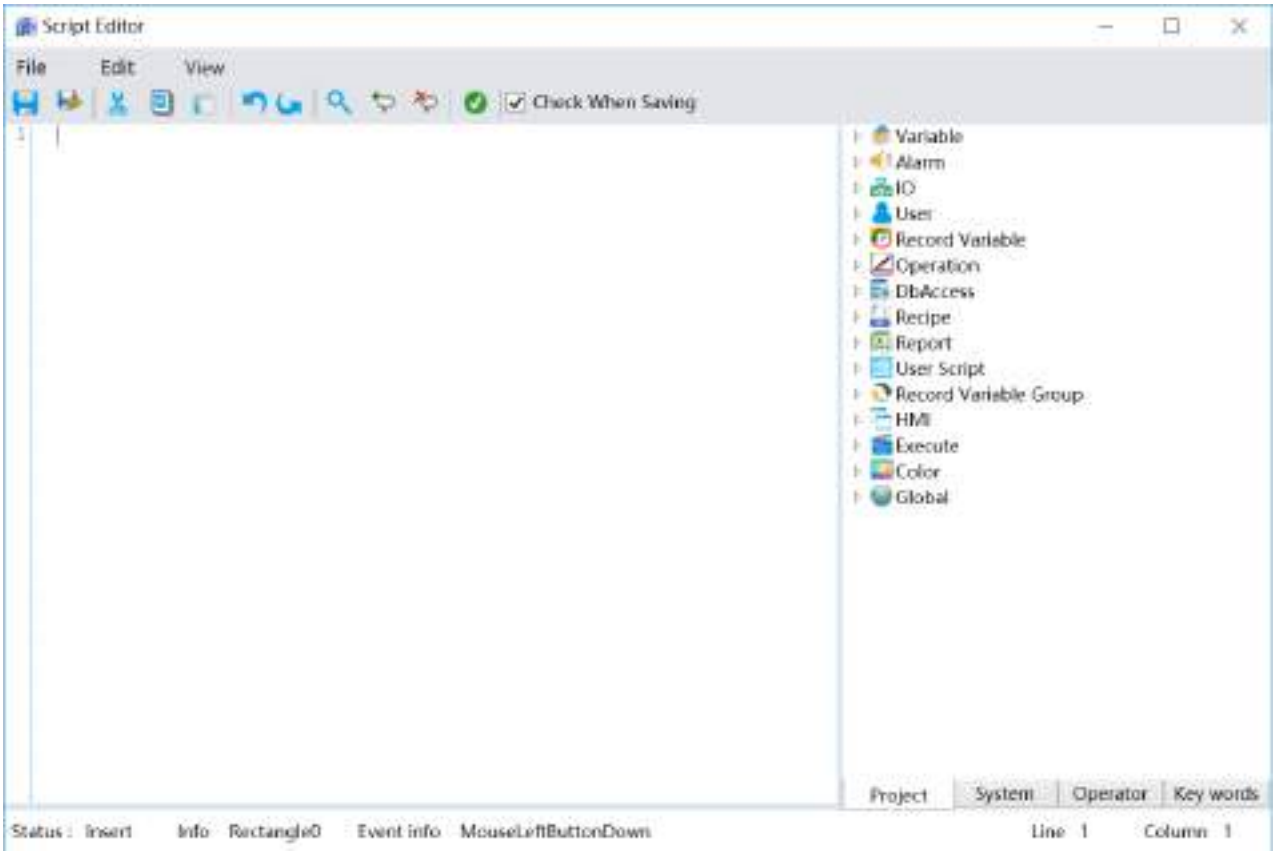
There are two types of Left Button Events: LeftButtonDown and LeftButtonUp.

The configuration is as follows, taking “LeftButtonDown” event as an example:

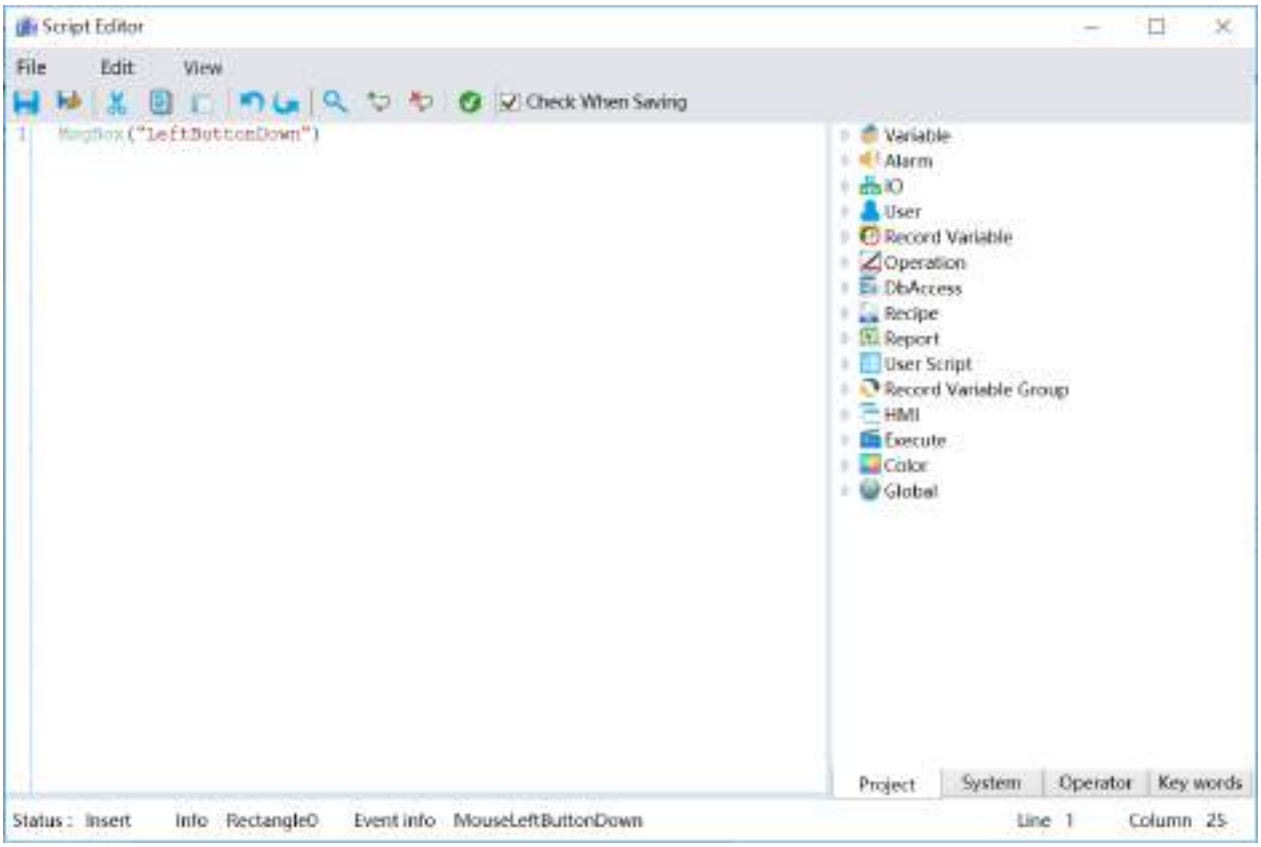
Step 1: Open project window where the animation needs to be configured under the DIAView software development environment → select graphic object to configure event in the sketchpad → open “Event” window → select “LeftButtonDown”→ click the button in “LeftButtonDown” bar, as shown in the figure below:



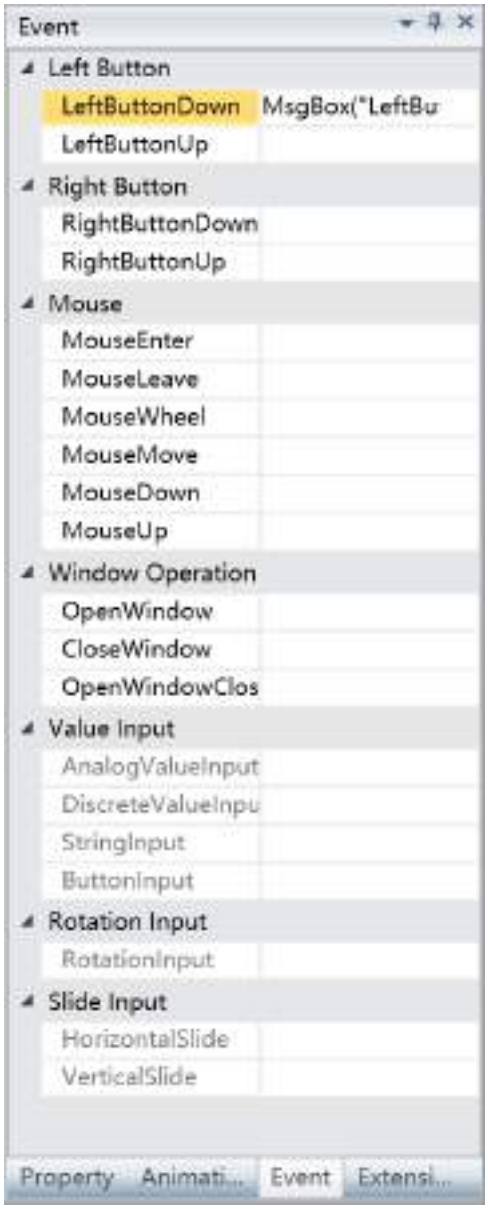
Step 2: The event script editor window will appear, as shown in the figure below:



Step 3: Write the script (The DIAView software uses the Visual Basic Script; the script editor will check for basic syntax errors), as shown in the figure below:



Step 4: Once the script is written and checked with no errors, click “Save and exit button” to complete event configuration; information on the configured event will be displayed in the event window, as shown in the figure below:



To delete an event, simply select the event in the event list and then right-click the mouse and select "Delete".

10.3 Right button event

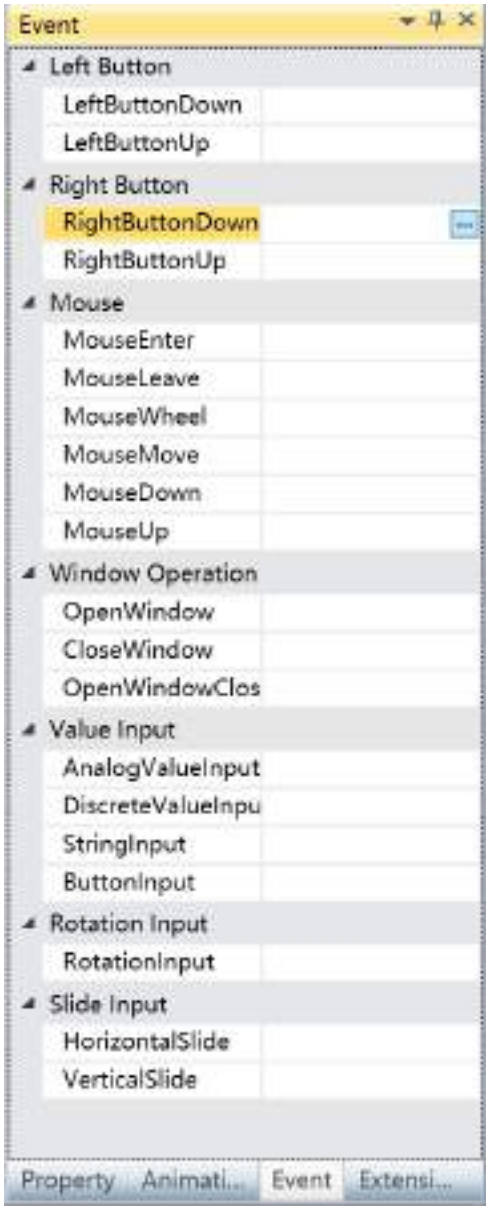
Right Button Event is an event triggered by performing operations with clicking the right mouse button, which will further use related script handlers to complete related functions.

There are two types of Right Button Events: RightButtonDown and RightButtonUp.

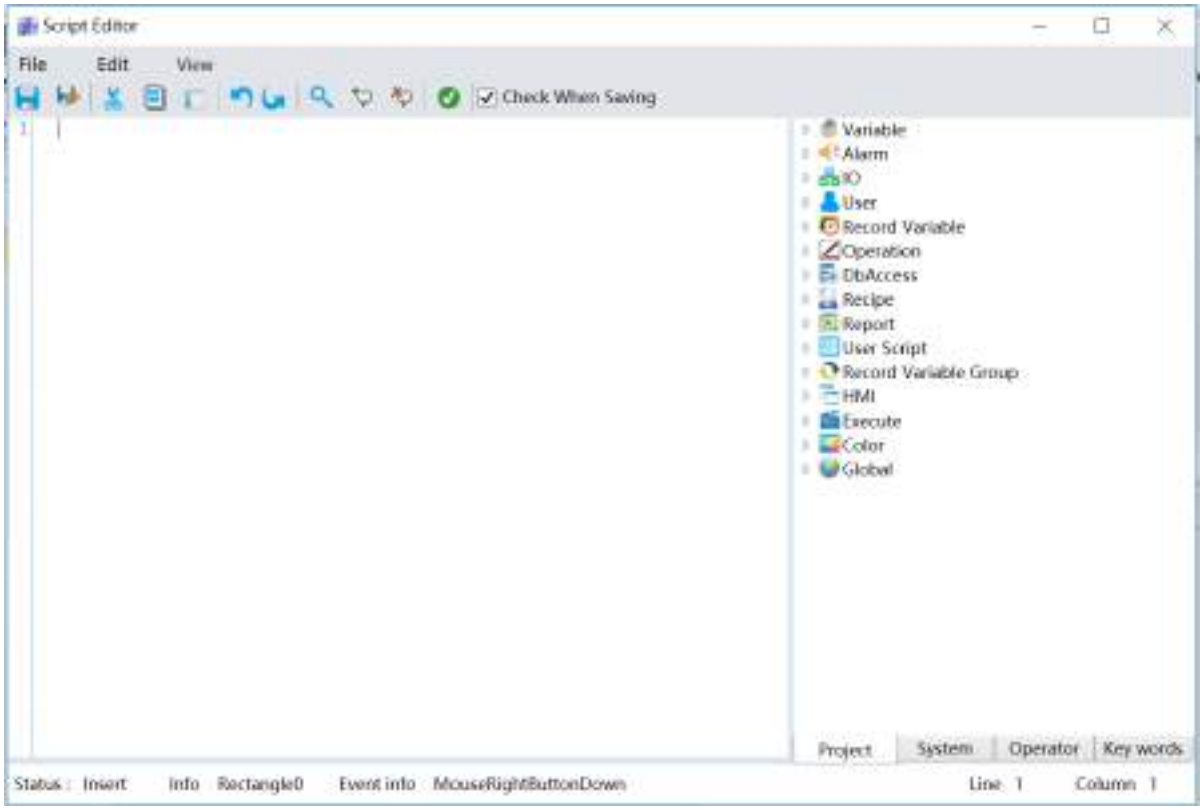
The configuration process is as follows, taking the "RightButtonDown" event as an example:

Step 1: Open project window where the animation needs to be configured under the DIAView software

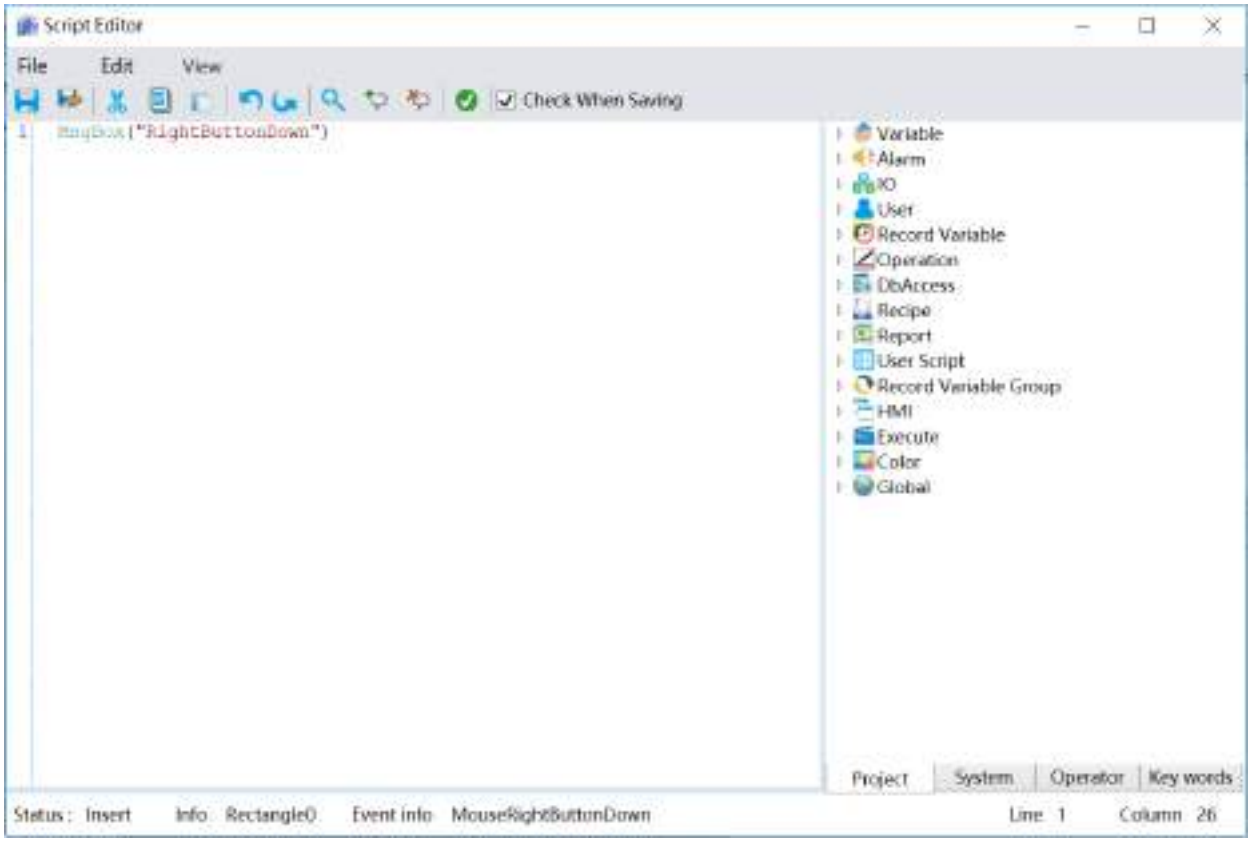
development environment → select graphic object to configure event in the sketchpad → open “Event” window → select “RightButtonDown”→ click the button in “RightButtonDown” bar, as shown in the figure below:



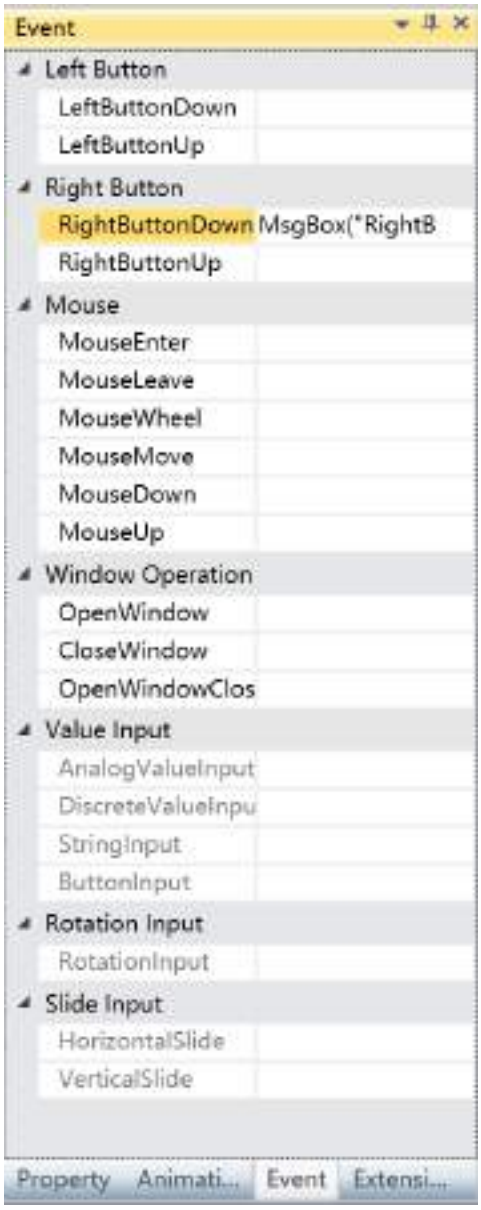
Step 2: The event script editor window will appear, as shown in the figure below:



Step 3: Write the script (The DIAView software uses the Visual Basic Script; the script editor will check for basic syntax errors), as shown in the figure below:



Step 4: Once the script is written and checked with no errors, click “Save and exit button” to complete event configuration; information on the configured event will be displayed in the event window, as shown in the figure below:



To delete an event, simply select the event in the event list and then right-click the mouse and select “Delete”.

10.4 Mouse event

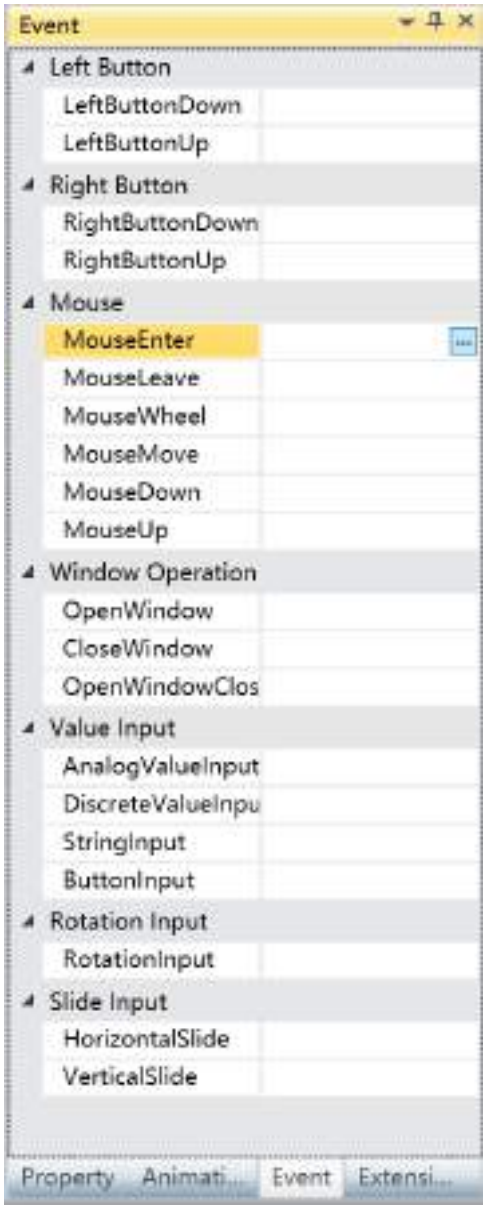
Mouse event is an event triggered by performing moving and clicking operations with the mouse on the graphic object, which will further use related script handlers to complete related functions.

There are six types of Mouse events: MouseEnter, MouseLeave, MouseWheel,

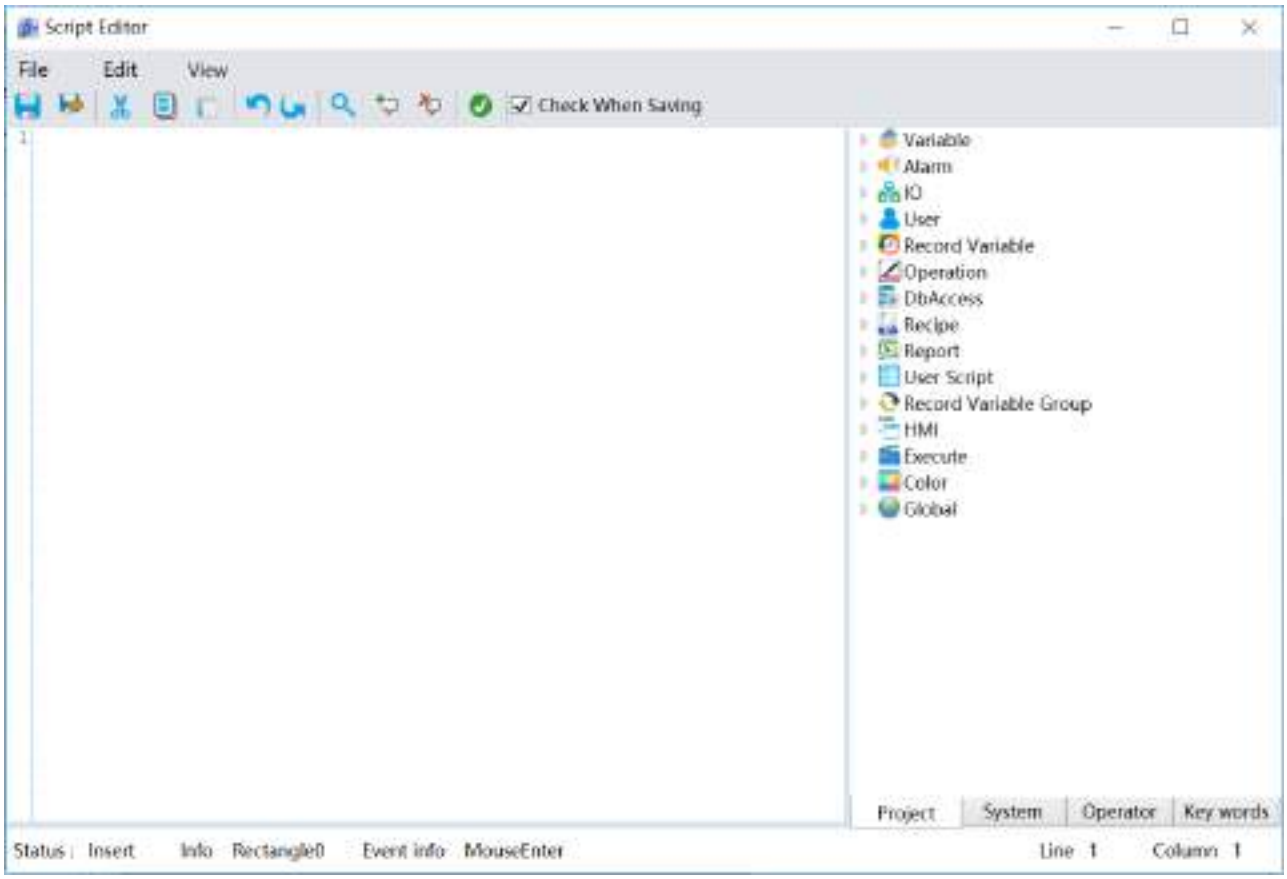
MouseMove, MouseDown and MouseUp.

The configuration is as follows, taking “MouseEnter” event as an example:

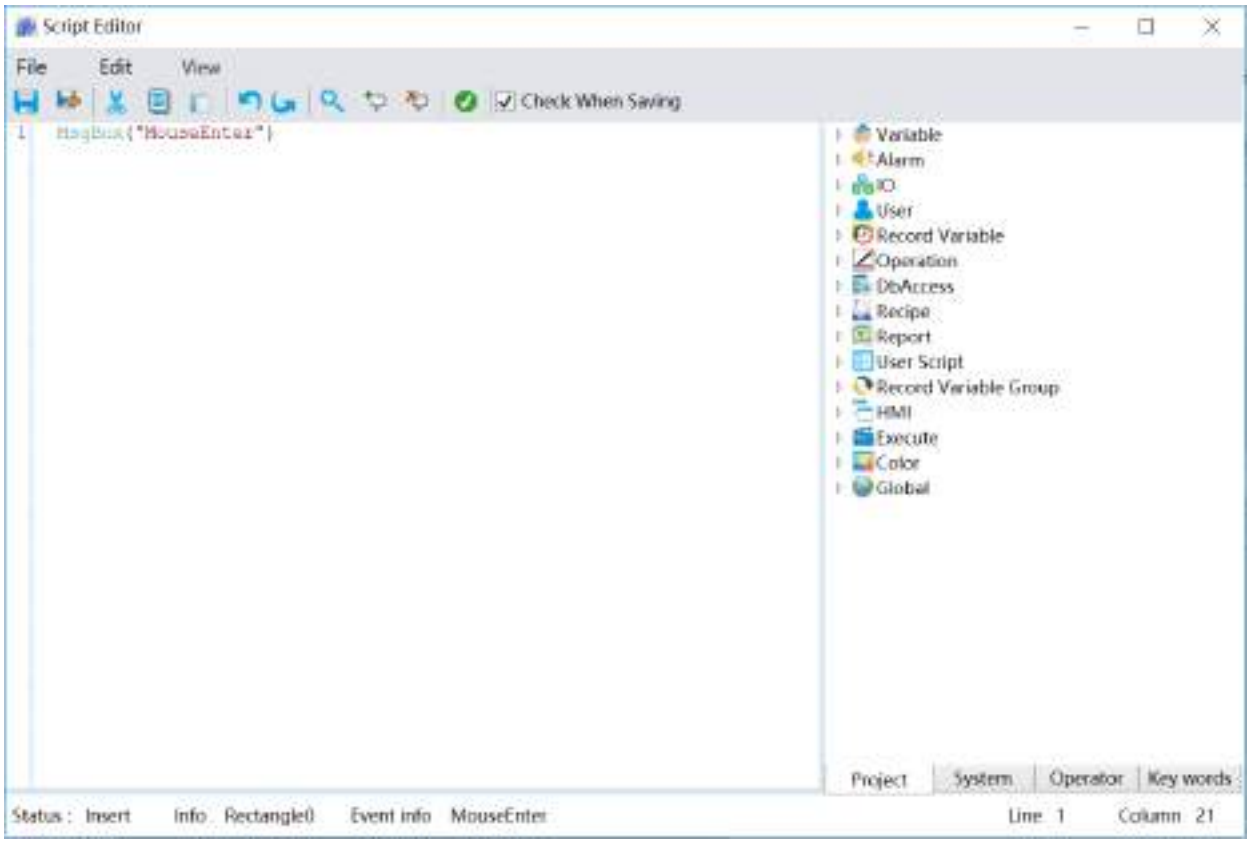
Step 1: Open project window where the animation needs to be configured under the DIAView software development environment → select graphic object to configure event in the sketchpad → open “Event” window → select “MouseEnter”→ click the button in the “MouseEnter” bar, as shown in the figure below:



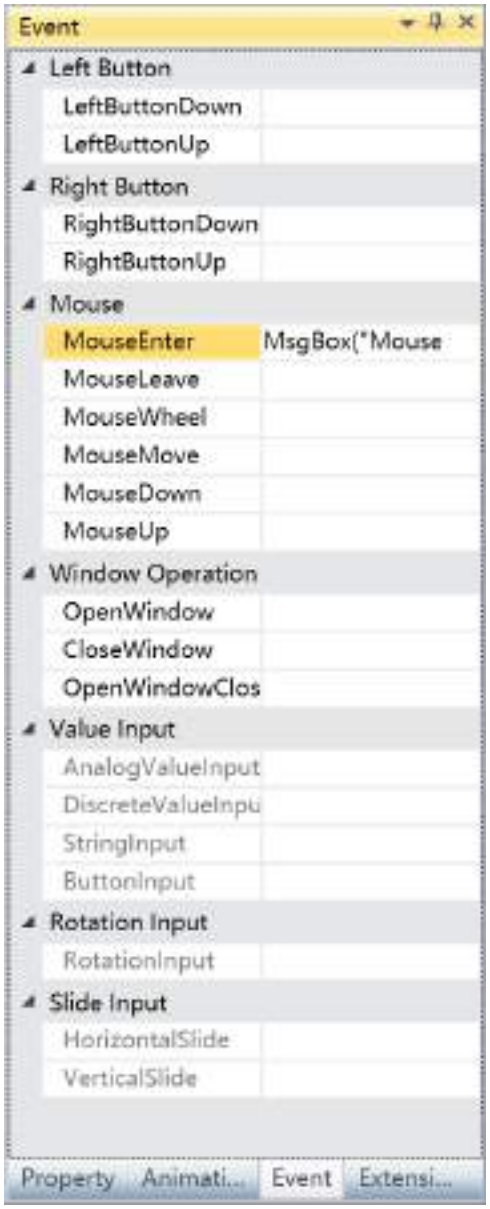
Step 2: The event script editor window will appear, as shown in the figure below:



Step 3: Write the script (The DIAView software uses the Visual Basic Script; the script editor will check for basic syntax errors), as shown in the figure below:



Step 4: Once the script is written and checked with no errors, click “Save and exit button” to complete event configuration; information on the configured event will be displayed in the event window, as shown in the figure below:



To delete an event, simply select the event in the event list and then right-click the mouse and select “Delete”.

10.5 Window operation event

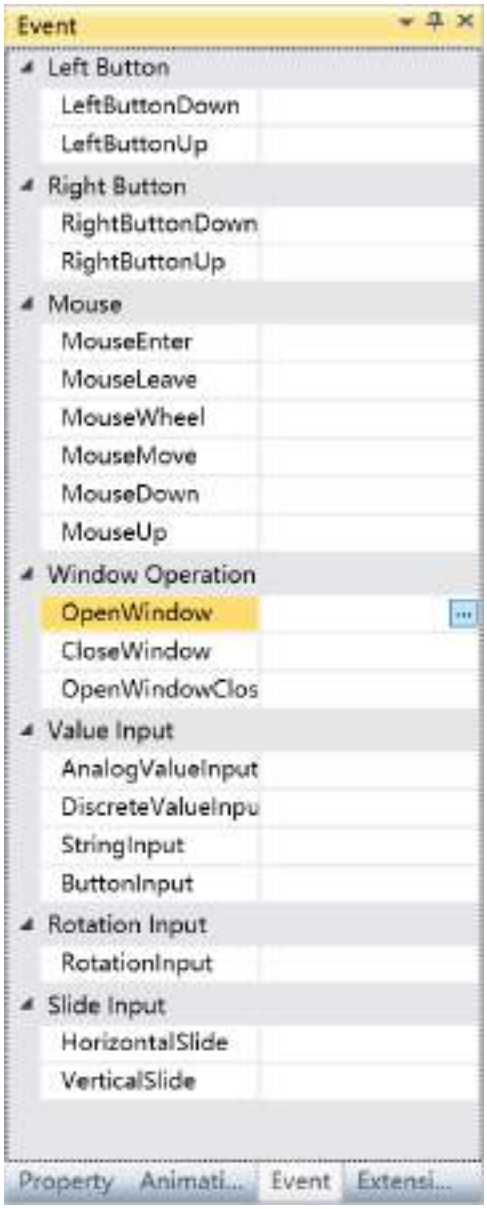
Window Operation Event is to open and close windows.

There are three types of Window Operation Events: OpenWindow, CloseWindow, OpenWindowCloseothers.

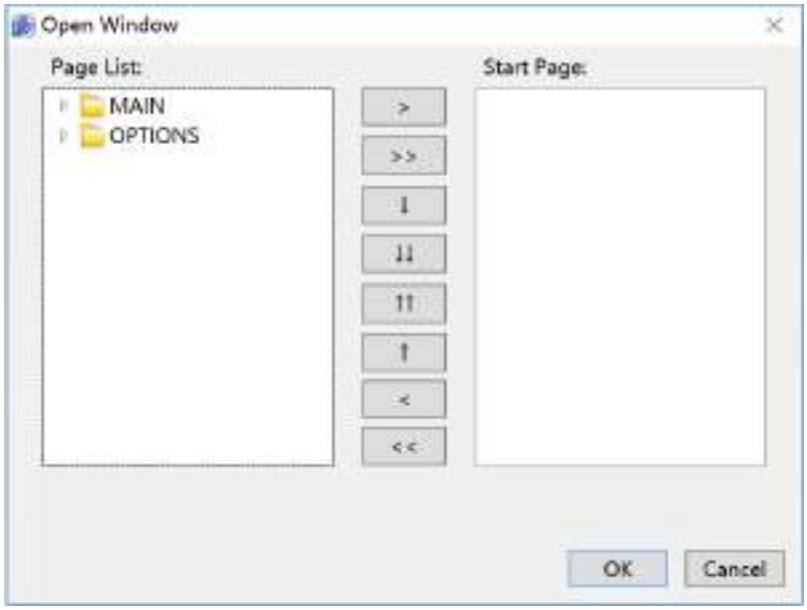
OpenWindow means opening the setting Window, CloseWindow means closing the setting Window and OpenWindowCloseothers means opening the setting Window and closing all other opened Windows besides the setting Windows.

The configuration process is as follows, taking “OpenWindow” event as an example:

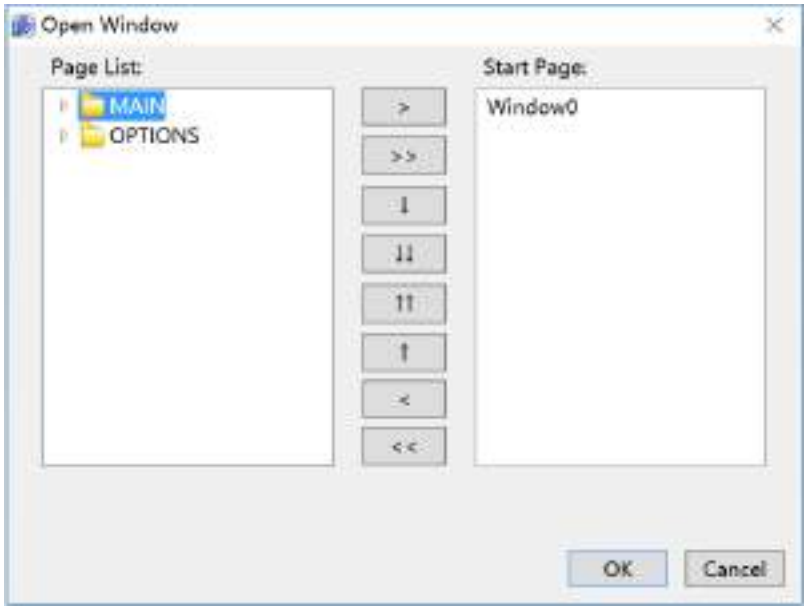
Step 1: Open the project window where the animation needs to be configured under the DIAView software development environment → select graphic object to configure event in the sketchpad → open “Event” window → select “OpenWindow”→ click the button in “OpenWindow” bar, as shown in the figure below:



Step 2: The window configuration window will appear, as shown in the figure below:



Step 3: Select the window to open from the “Page List” to the left, and then click the “>” button to add it to the “Start Page” :

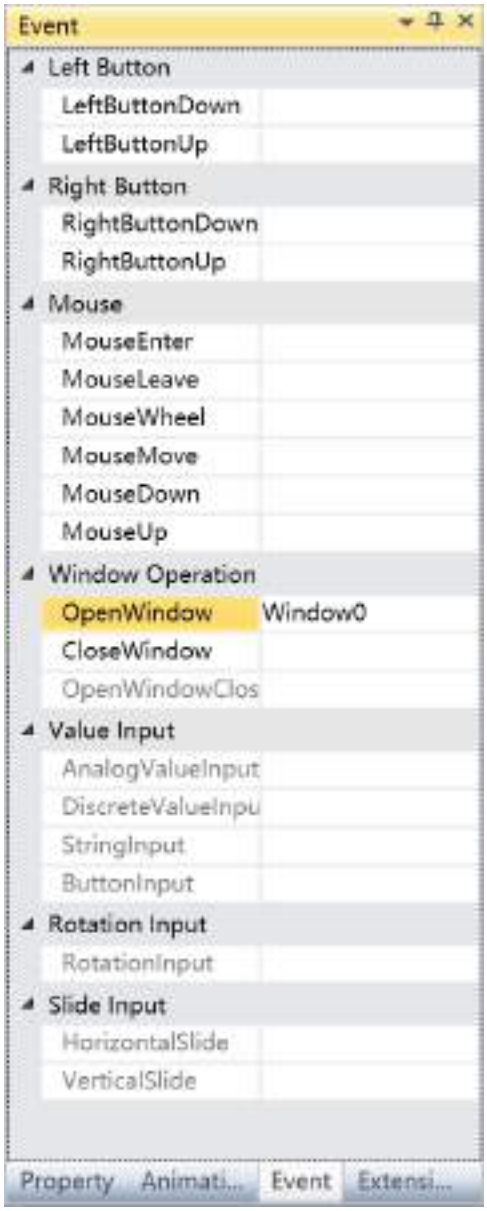


➢ **Button functions:**

- “>” : Add a window to selected window
- “>>”: Add all to the selected window
- “↓” : Move the window down
- “↓↓”: Move the window to the bottom

- “↑↑”: Move the window to the top
- “↑” : Move the window up
- “<” : Remove a window from selected window
- “<<”: Remove all windows from selected window

Step 4: When configuration is complete, press the “OK” button; information of the configured event will be displayed in the “Event” window, as shown in the figure below:



To delete an event, simply select the event in the event list and then right-click themouse and select “Delete”.

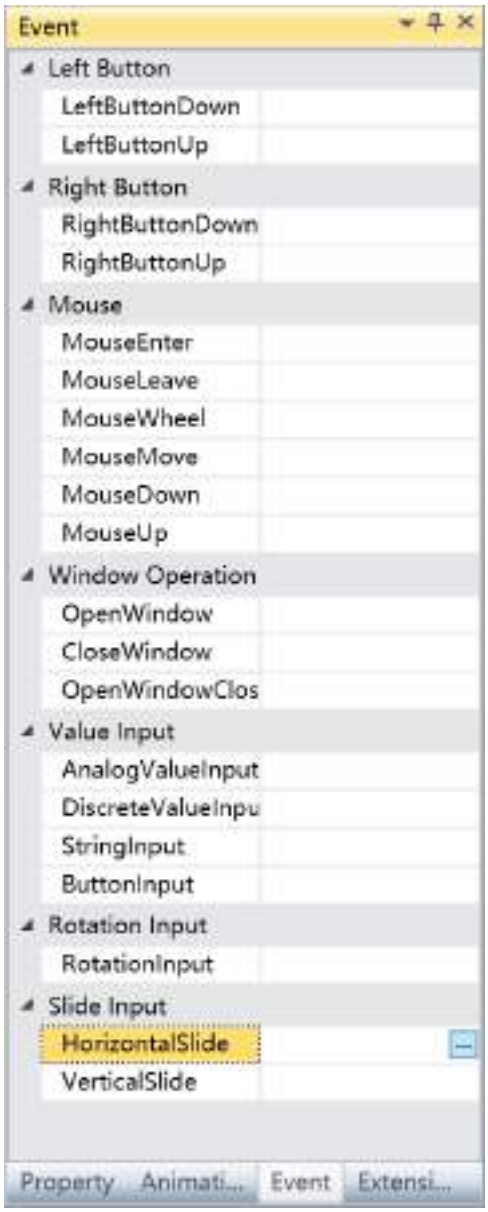
10.6 Sliding input event

Sliding input event is an event triggered by moving the position of graphic objects in the window to change the value of associated variables.

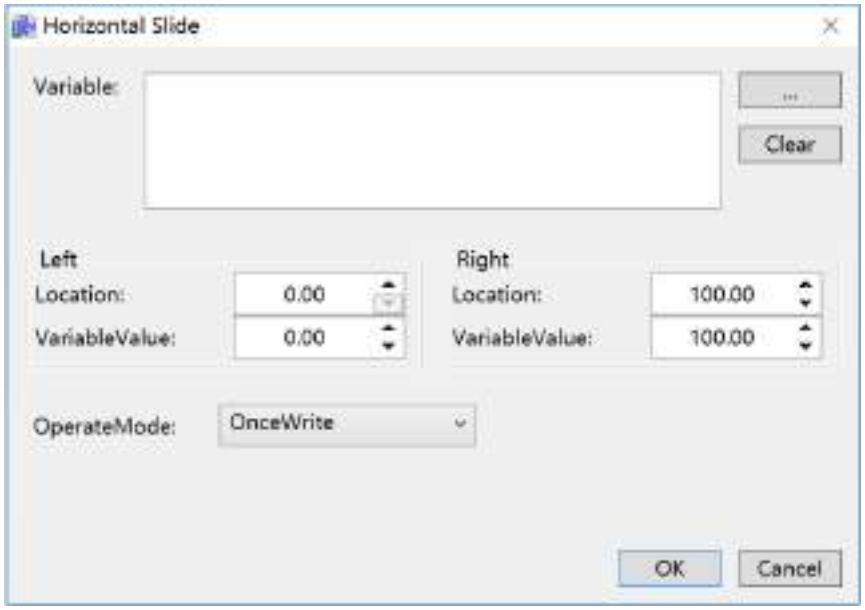
There are two types of sliding input events: HorizontalSlide and VerticalSlide.

The configuration is as follows, taking“Horizontalslide” event as an example:

Step 1: Open the project window where the animation needs to be configured under the DIAView software development environment → select graphic object to configure event in the sketchpad → open “Event” window → select “HorizontalSlide”→ click the button in “HorizontalSlide” bar, as shown in the figure below:



Step 2: The Horizontal Slide configuration window will appear, as shown in the figure below:



The meanings of each setting in the configuration window are as follows:

- **Variable:**

  - The associated variable name. You can also click the ... button to open the variable browser to select a variable; the “Clear” button can clear the contents in the input box

- **Left Position:**

  - Location: Sets the value of the left-most position when the graphic object slides horizontally

  - VariableValue: Sets the variable value when the graphic object slides horizontally and reaches the left-most position

- **Right Position:**

  - Location: Sets the value of the right-most position when the graphic object slides horizontally

  - VariableValue: Sets the variable value when the graphic object slides horizontally and reaches the right-most position

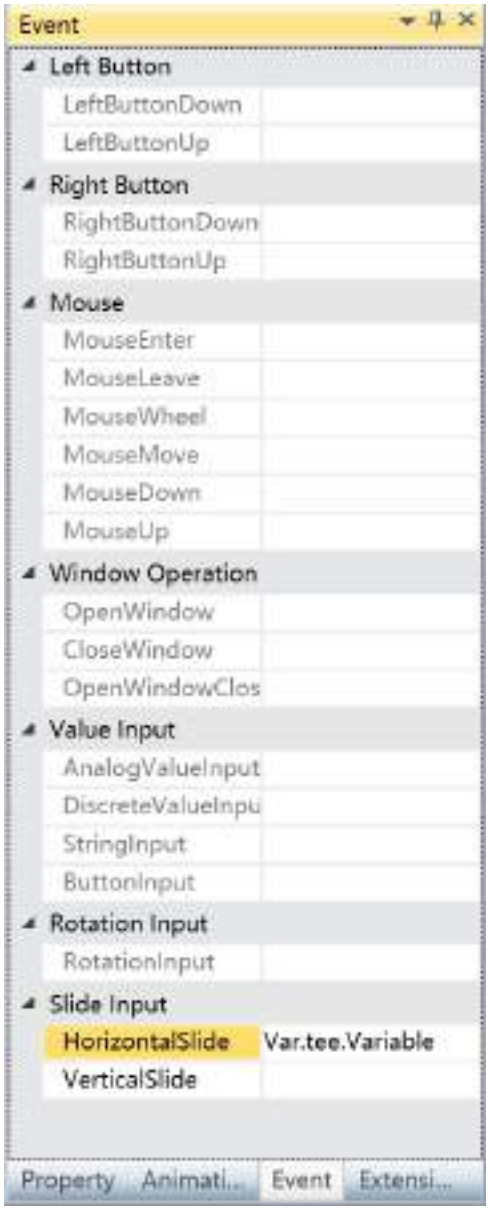
- **Operation modes:**

  - SequentialWrite: The input value changes in real-time with the moving position

  - OnceWrite: The input value changes when the moving stops

Step 3: When configuration is complete, press the “OK” button to complete the configuration of the

"HorizontalSlide" event; the event configuration of this graphic will be displayed in the "Event" window, as shown in the figure below:



To delete an event, simply select the event in the event list and then right-click the mouse and select "Delete".

10.7 Value input event

Value Input event is when clicking a graphic object in the window triggers the value input window of the DIAView software for value input operations.

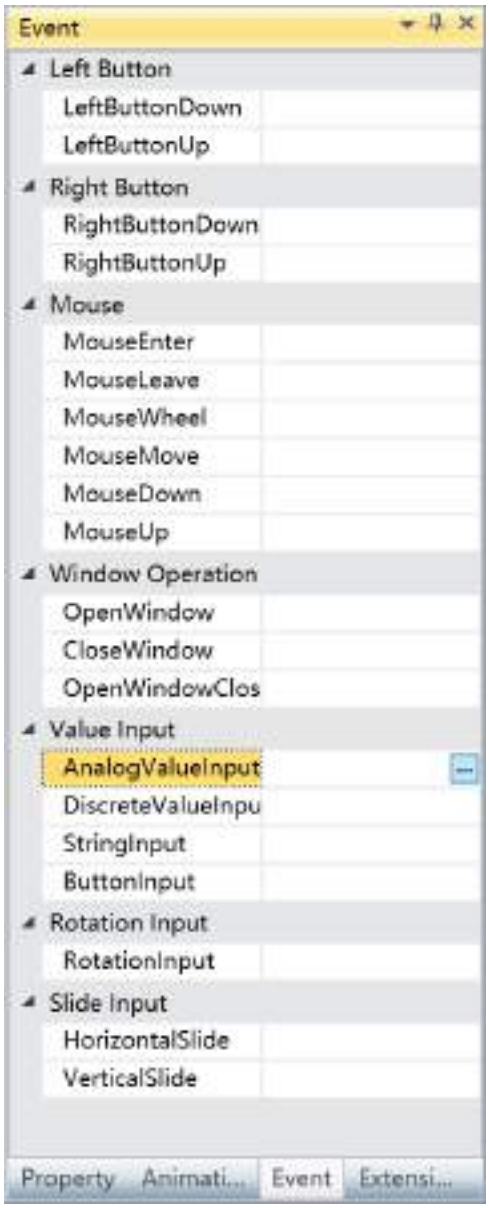
There are four types of value input events: AnalogValueInput, DiscreteValueInput, StringInput and

ButtonInput.

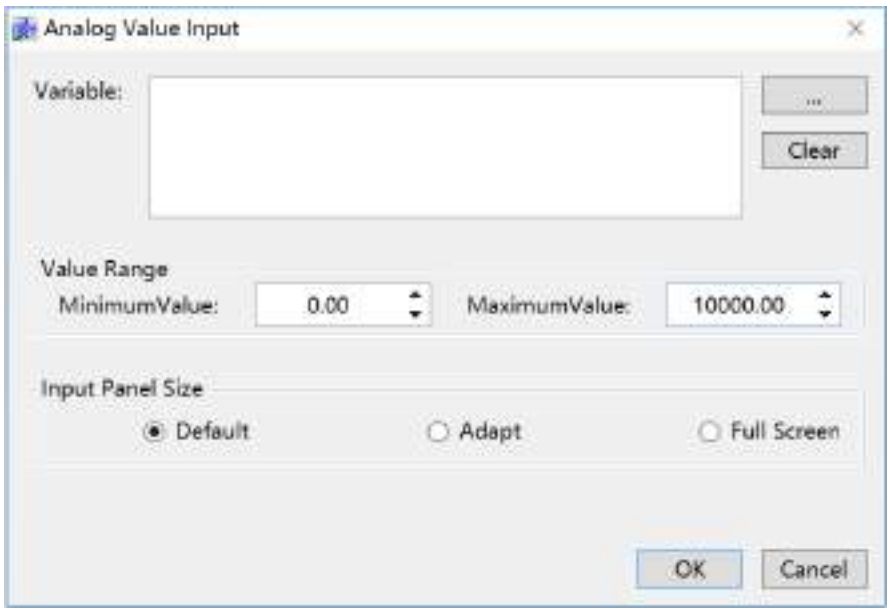
1.AnalogValueInput:

When this event is set for a graphic object, a analog value input dialog will appear when this graphic object is clicked by the mouse in the DIAView software runtime environment, including digital buttons; users can click the digital buttons to input numbers to change the value of the associated analog variable. Configuring steps are as follows:

Step 1: Open the project window where the event needs to be configured under the DIAView software development environment → select graphic object to configure event in the sketchpad → open “Event” window → select “AnalogValueInput”→ click the button in “AnalogValueInput” bar, as shown in the figure below:



Step 2: The Analog Value Input configuration window will appear, as shown in the figure below:



The meanings of each setting in the configuration window are as follows:

- **Variable:**

  - The associated variable name, the variable type is analog value; you can also click the ... button to open the variable browser to select a variable. The “Clear” button can clear the contents in the input box

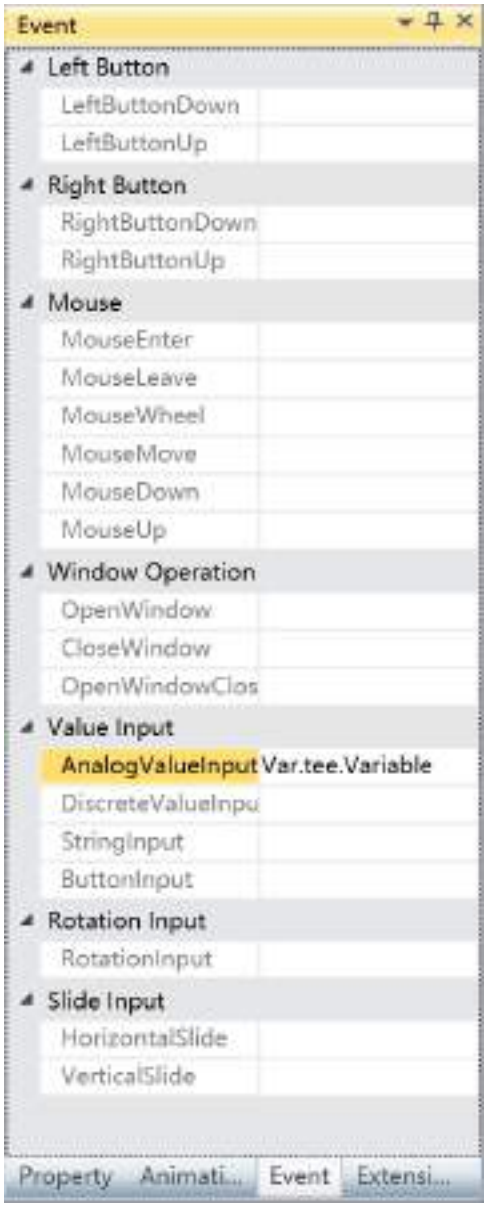
- **Value range:**

  - MinimumValue: Sets the minimum value that can be input
  - MaximumValue**:** Sets the maximum value that can be input

- **Input Panel Size:**

  - Set the size of pop-up analog input keyboard

Step 3: When configuration is complete, press the “OK” button to complete the configuration of the “AnalogValueInput” event. The event configuration of this graphic will be displayed in the “Event” window, as shown in the figure below:

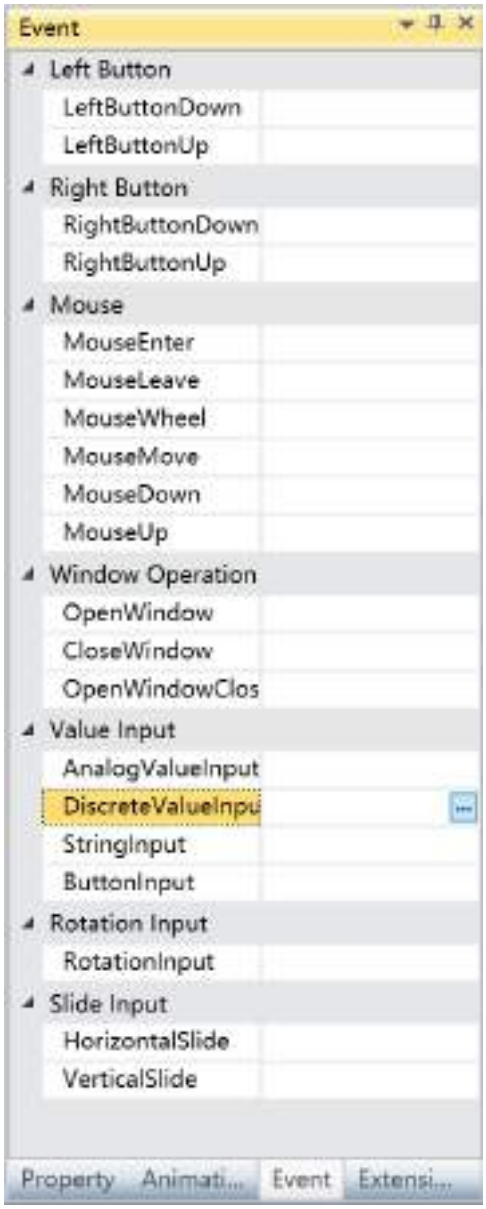


To delete an event, simply select the and then right-click the mouse and select "Delete".

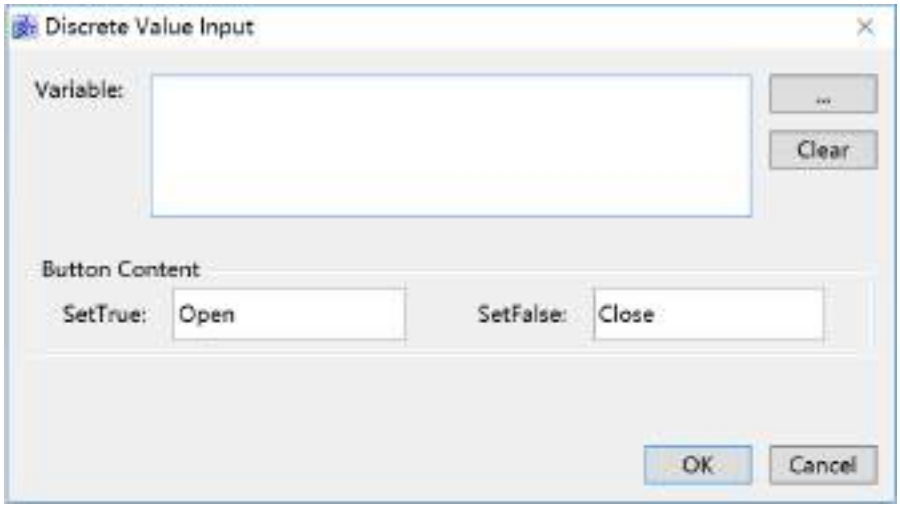
2.DiscreteValueInput:

When this event is set for a graphic object, a discrete value input dialog will appear when this graphic object is clicked by the mouse in the DIAView software runtime environment.Users can set the button's text according to associated discrete variable value is True or False. Configuring steps are as follows:

Step 1: Open the project window where the event needs to be configured under the DIAView software development environment → select the graphic object to configure event in the sketchpad → click the "Value Input " in the "Event" window → select "DiscreteValueInput" from the options→ click the button in the "DiscreteValueInput"bar, as shown in the figure below:



Step 2: The Discrete Value Iput configuration window will appear, as shown in the figure below:



The meanings of each setting in the configuration window are as follows:

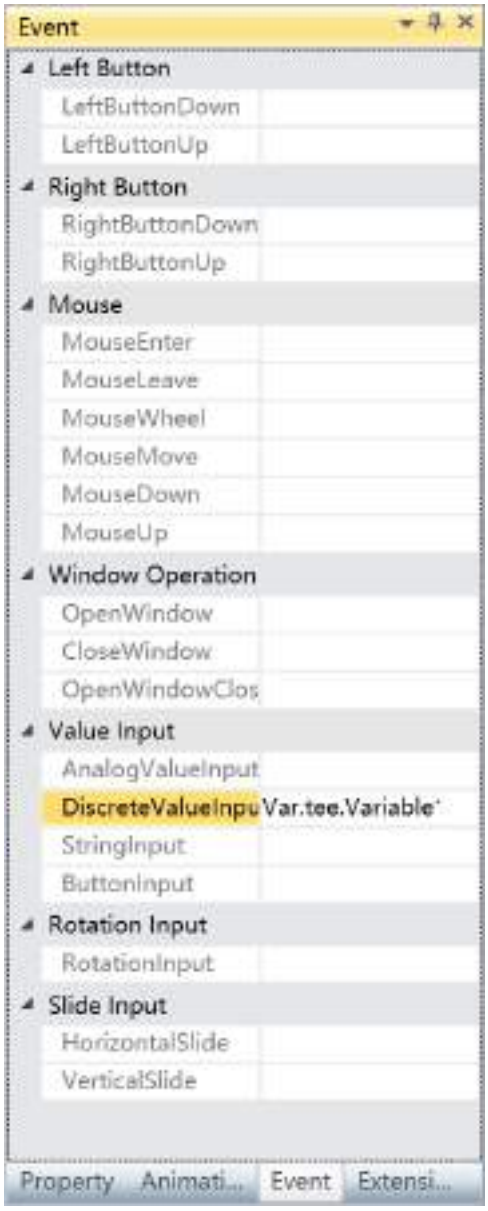
- **Variable:Variable:**

  - The associated variable name, the variable type is digital value; you can also click the ... button to open the variable browser to select a variable; the “Clear” button can clear the contents in the input box

- **Button text:**

  - SetTrue: Text content displayed by the button when the value is True
  - SetFalse: Text content displayed by the button when the value is False

Step 3: When configuration is complete, press the “OK” button to complete the configuration of the “DiscreteValueInput” event. The event configuration of this graphic will be displayed in the “Event” window, as shown in the figure below:



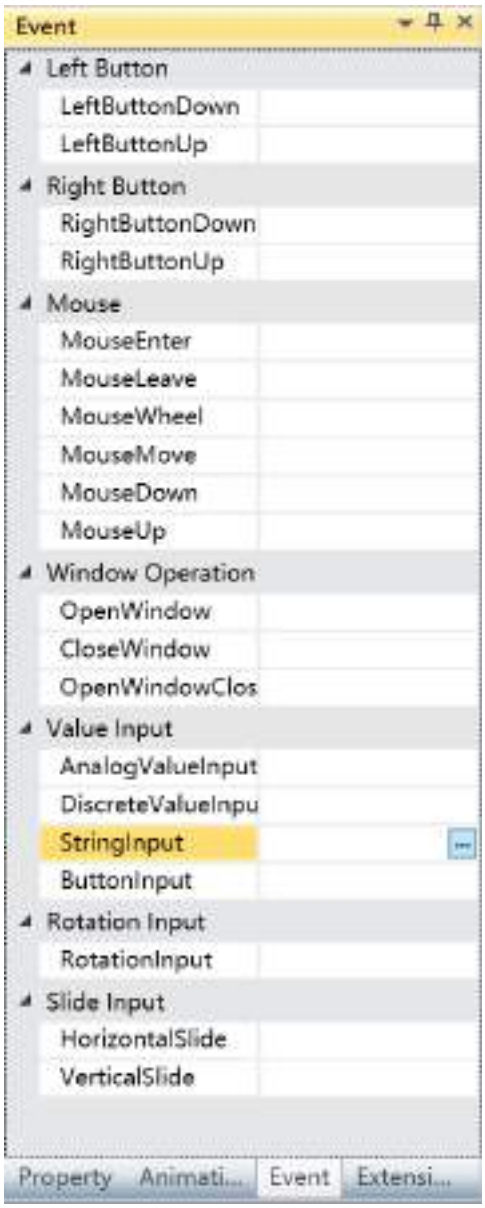
To delete an event, simply select the event and then right-click the mouse and select "Delete".

3.StringInput:

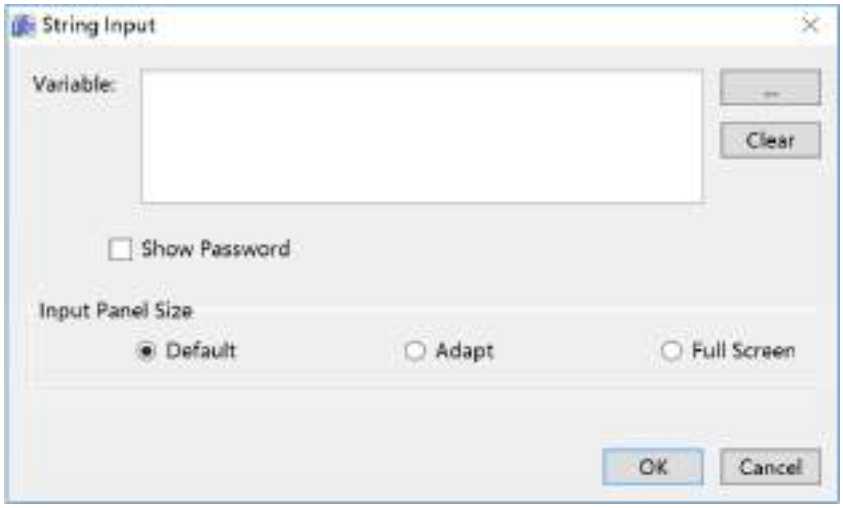
When this event is set for a graphic object, a string input dialog will appear when this graphic object is clicked by the mouse in the DIAView software runtime environment; it includes letters and number input keyboard and users can click the keys to enter strings to change the value of the associated string variable.

Step 1: Open the project window where the event needs to be configured under the DIAView software development environment → select the graphic object to configure event in the sketchpad → click the "Value Input" in the "Event" window → select "StringInput" from the options→ click the button in the

"StringInput" bar, as shown in the figure below:



Step 2: The String Input configuration window will appear, as shown in the figure below:



The meanings of each setting in the configuration window are as follows:

- **Variable:**

  - The associated variable name, the variable type is text amount; you can also click the ... button to open the variable browser to select a variable. The “Clear” button can clear the contents in the input box

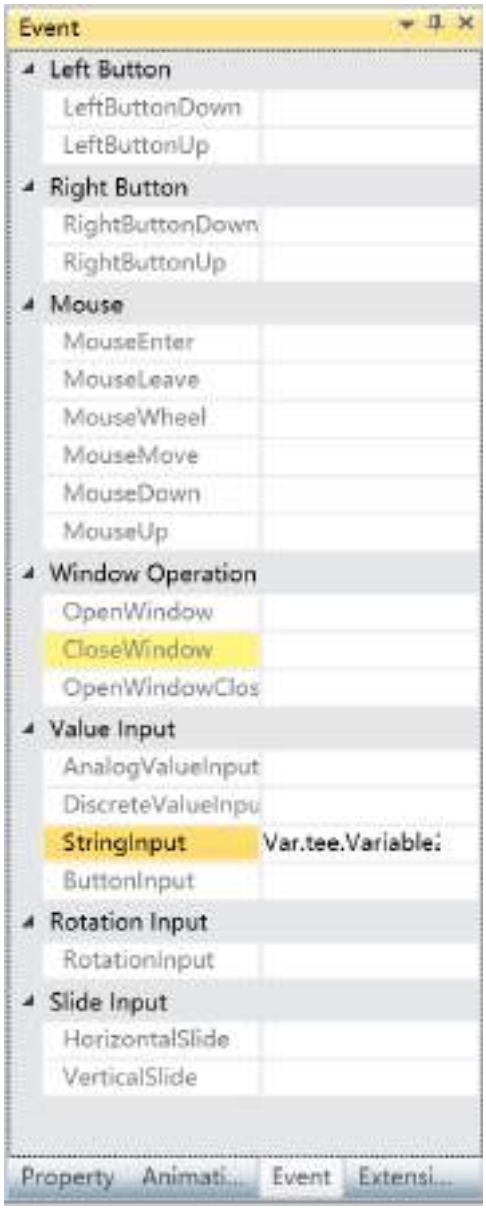
- **Display password:**

  - If this checkbox is selected, the input value will be displayed in a password form

- **Keyboard size:**

  - Set the size of pop-up string input keyboard under the runtime environment

Step 3: When configuration is complete, press the “OK” button to complete the configuration of the “StringInput” event. The event configuration of this graphic will be displayed in the “Event” window, as shown in the figure below:

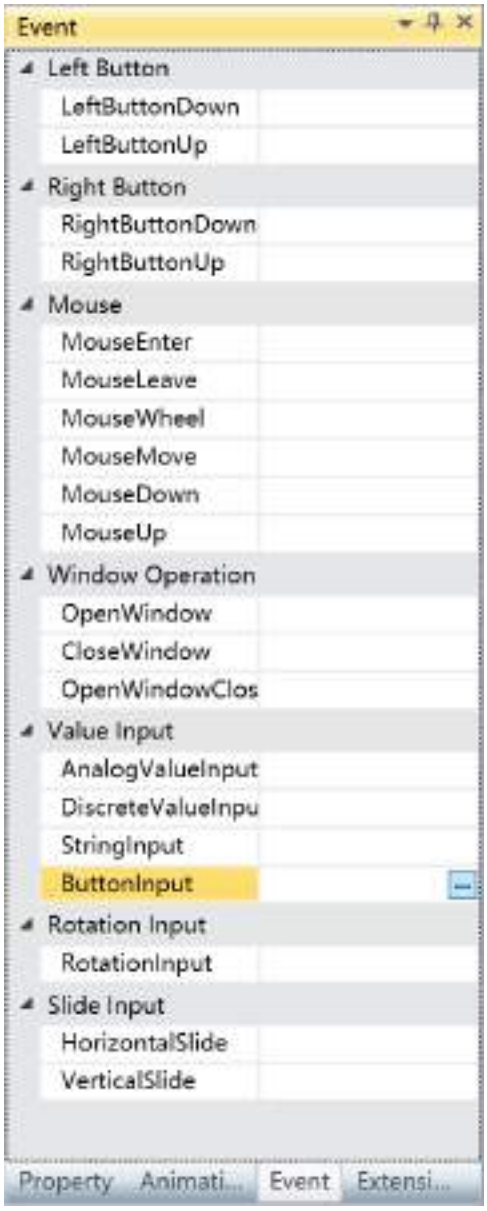


To delete an event, simply select the event and then right-click the mouse and select “Delete”.

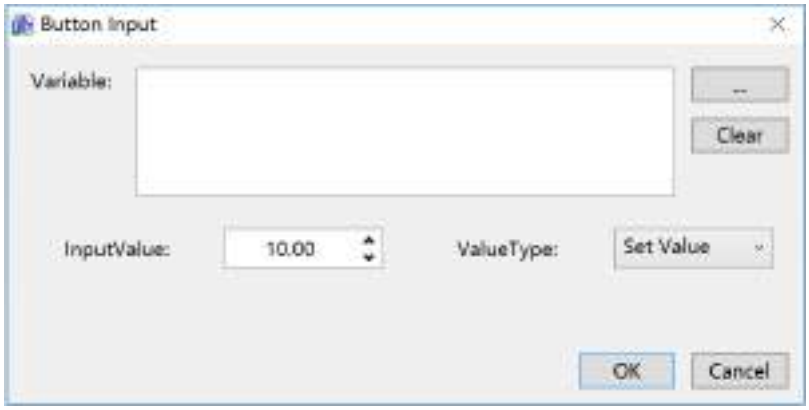
4.ButtonInput:

When this event is set for a graphic object, if this graphic object is clicked with the mouse under the DIAView software runtime environment, the value of the associated variable will change according to the setting method ; for example, reset the value , addition or subtraction between the variable value and the setting value etc. Configuring steps are as follows:

Step 1: Open the project window where the event needs to be configured under the DIAView software development environment → select the graphic object to configure event in the sketchpad → click the “Value Input ” in the “Event” window → select “ButtonInput” from the options→ click the button in the “ButtonInput” bar , as shown in the figure below:



Step 2: The Button Input configuration window will appear, as shown in the figure below:



The meanings of each setting in the configuration window are as follows:

- **Variable:**

  - The associated variable name, the variable type is analog value or digital value; you can also click the ... button to open variable browser to select a variable; the “Clear” button can clear the contents in the input bax

- **Input value:**

  - Input the setting value

- **Nemerical type:**

  - Changes the variable value when the graphic object is pressed; there are 6 types of changing methods:

    - **Set Value:** Give the “InputValue” directly to the associated variable; valid for analog values

    - **Increase:** Adds the setting value to the associated variable value and then give it to the associated variable; valid for analog values

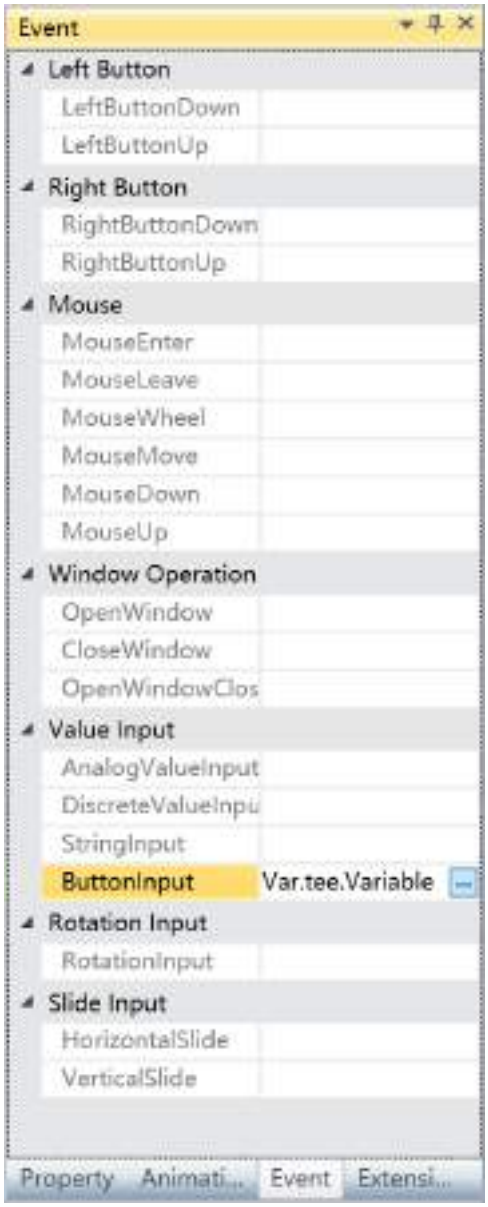
    - **Decrease:** Subtracts the setting value from the associated variable value and then give it to the associated variable;valid for analog values

    - **Multiply:** Multiples the setting value to the associated variable value and then give it to the associated variable; valid for analog values

    - **Divide:** Divides the variable value by the set value and then give it to the associated variable; valid for analog values

    - **Toggle:** The analog value switches between the two values 0 and 1, and the digital value switches between true and false

Step 3: When configuration is complete, press the “OK” button to complete the configuration of the “ButtonInput” event; the event configuration of this graphic will be displayed in the “Event” window, as shown in the figure below:



To delete an event, simply select the event in the event list and then right-click the mouse and select “Delete”.

10.8 Rotating input event

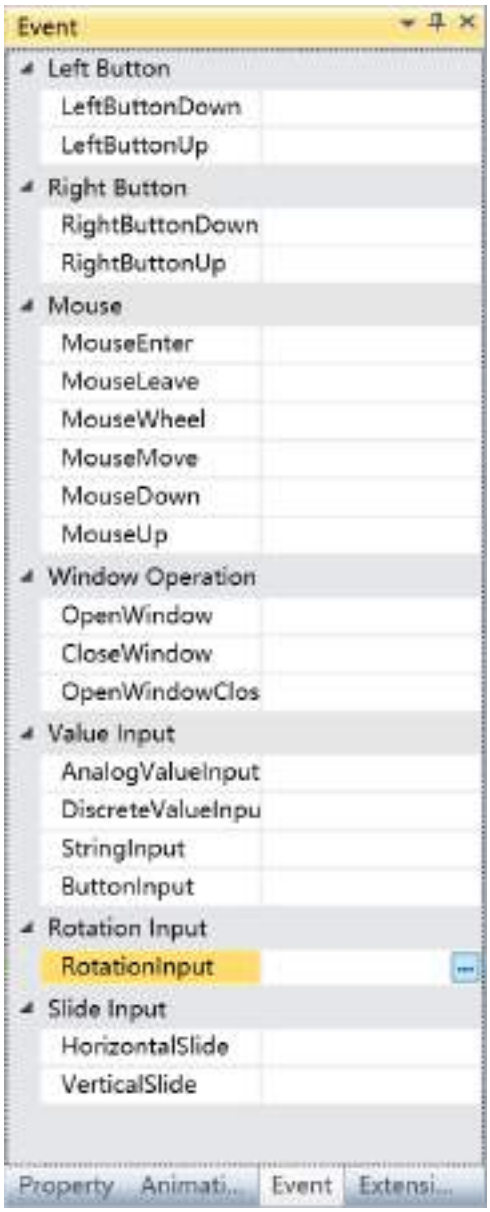
Rotating input event is events triggered by rotating the angle of graphic objects in the window which changes the value of associated variables.

There is only one type of rotary input event: RotationInput.

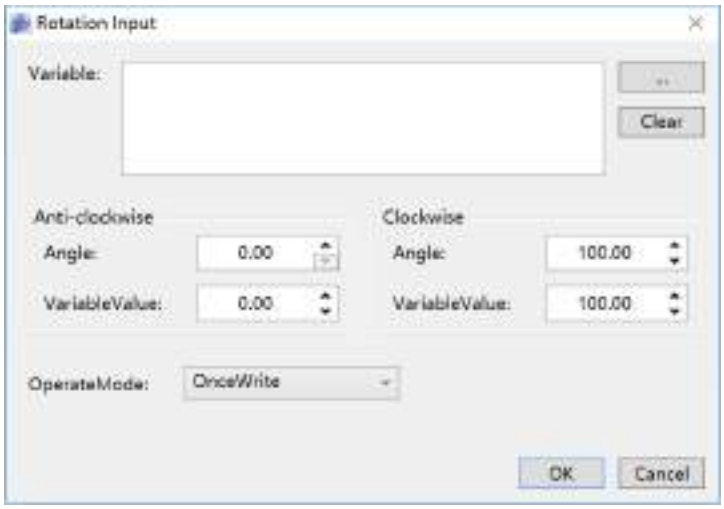
Configuration steps are as follows:

Step 1: Open the project window where the event needs to be configured under the DIAView software

development environment → select graphic object to configure event in the sketchpad → open “Event” window → select “RotationInput”→ click the button in “RotationInput” bar, as shown in the figure below:



Step 2: The Rotation Input configuration window will appear, as shown in the figure below:



The meanings of each setting in the configuration window are as follows:

- **Variable:**

  - The associated variable name. You can also click the ... button to open the variable browser to select a variable; the “Clear” button can clear the contents in the input box

- **Anti-clockwise:**

  - Angle: Sets the counter-clockwise rotation angle of the graphic object (Unit: degrees)

  - VariableValue: Sets the variable value for the counter-clockwise rotation angle of the graphic object

- **Clockwise:**

  - Angle: Sets the clockwise rotation angle of the graphic object (Unit: degrees)

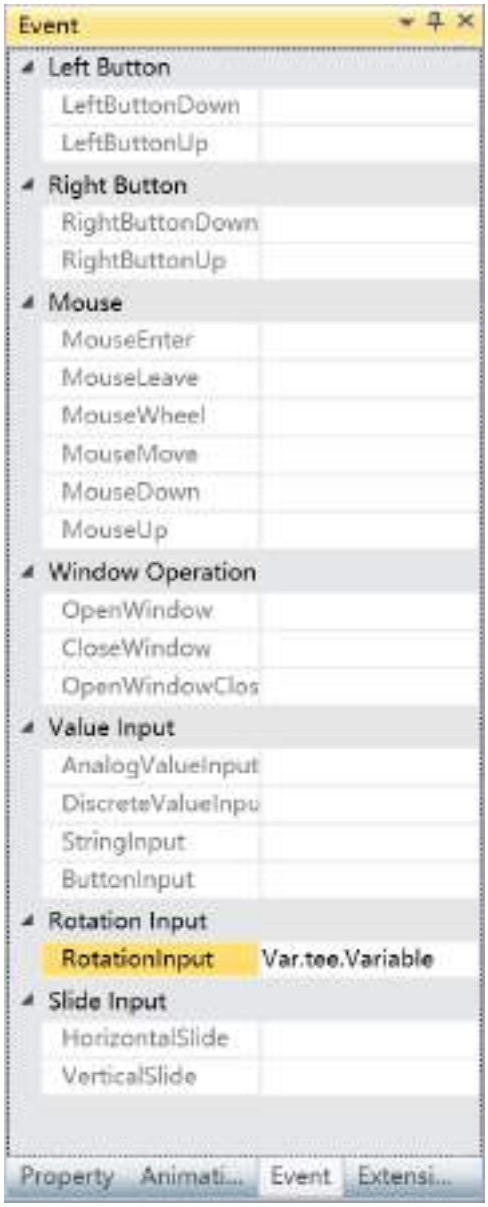
  - VariableValue: Sets the variable value for the clockwise rotation angle of the graphic object

- **OperateMode:**

  - OnceWrite: The input value changes continuously

  - SequentialWrite: The input value changes after a period of time

Step 3: When configuration is complete, press the “OK” button to complete the configuration of the “RotatingInput” event. The event configuration of this graphic will be displayed in the “Event” window, as shown in the figure below:



To delete an event, simply select the event in the event list and then right-click the mouse and select “Delete”.

10.9 Window program event

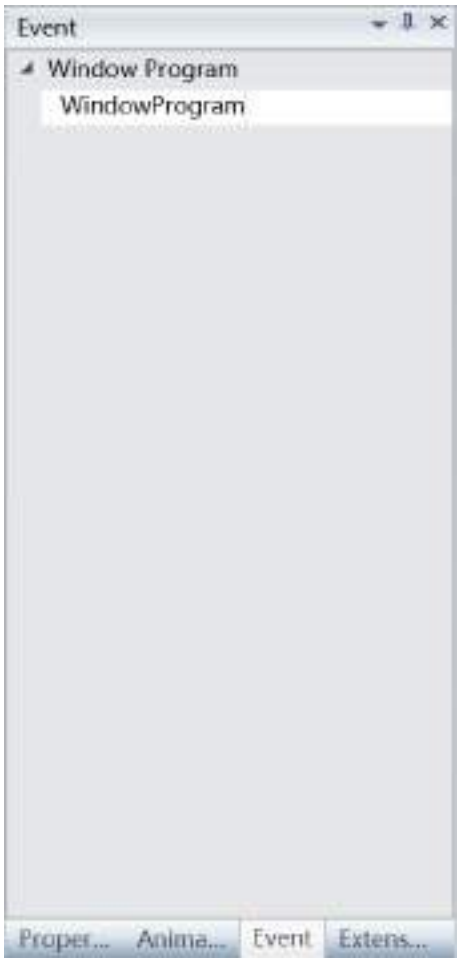
Window program event is to set the custom time interval program executed in the runtime environment.

Window program event has only one type: Window Program.

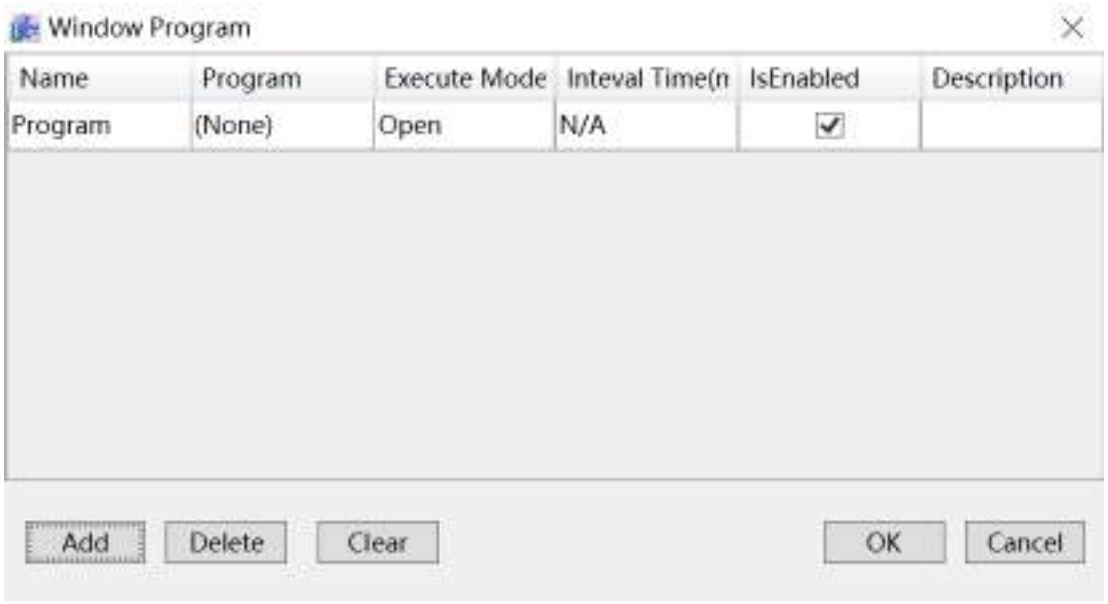
Configuration steps are as follows:

Step 1: 1:Open the project window where the event needs to be configured under the DIAView software

development environment → select window to configure event in the sketchpad → open “Event” window → select “Window Program ”→ click the button in “Window Program” bar , as shown in the figure below:



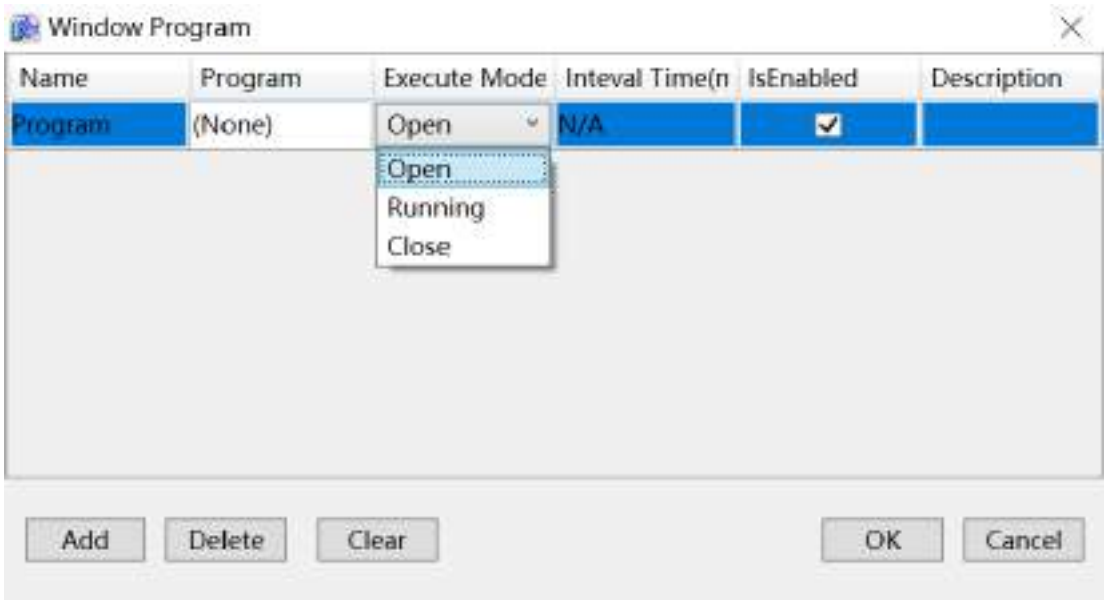
Step 2: The Window Program configuration window will appear, click "Add" button ,as shown in the figure below:



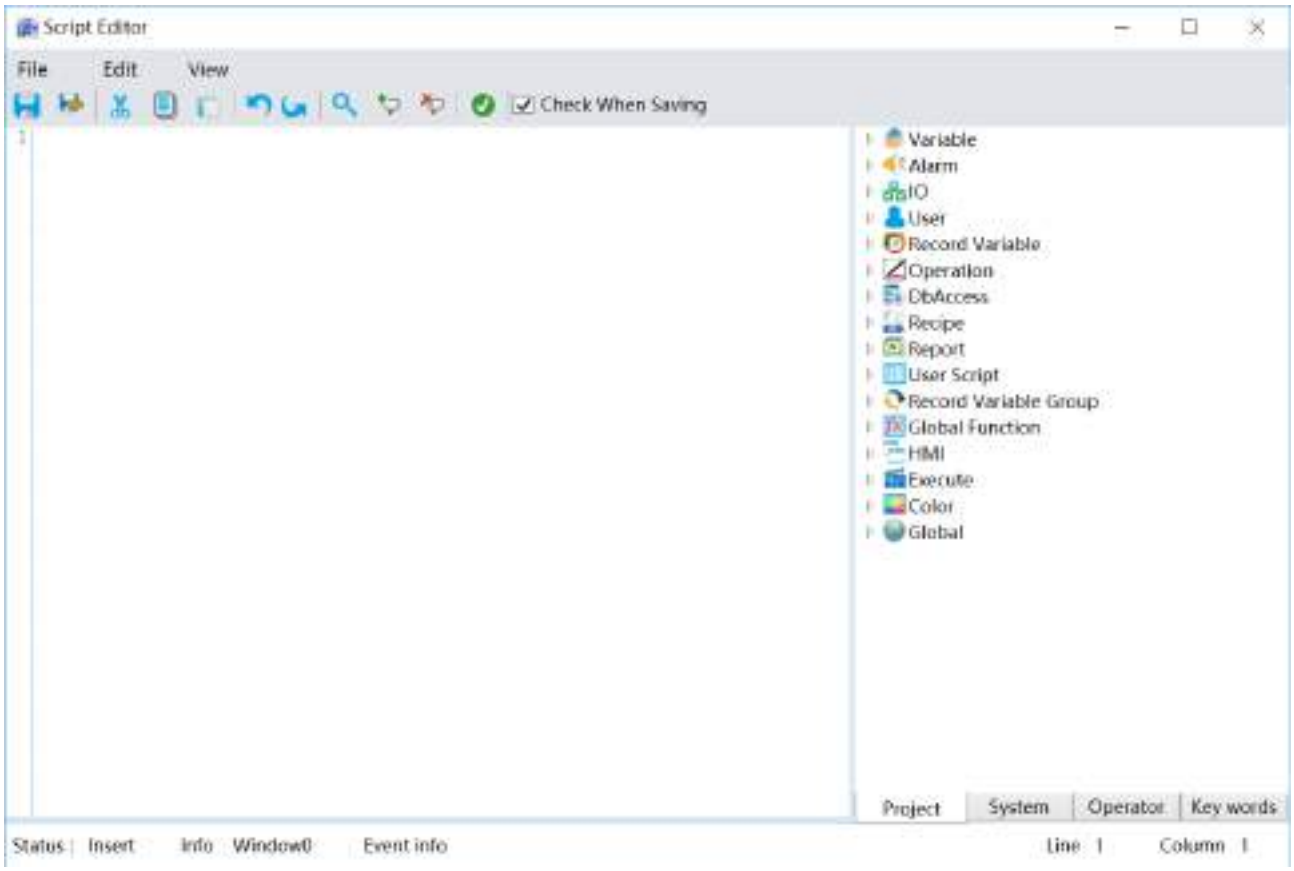
The meanings of each setting in the configuration window are as follows:

- **Excution Mode:** execute the specified program including Loaded,TimerTicked and Closed
- **Name:** set the function needs to be executed
- **Interval Time(ms):** set the time interval of specified program(ms)
- **IsEnabled:** set whether to execute the specified program
- **Description:** related information of the specified program

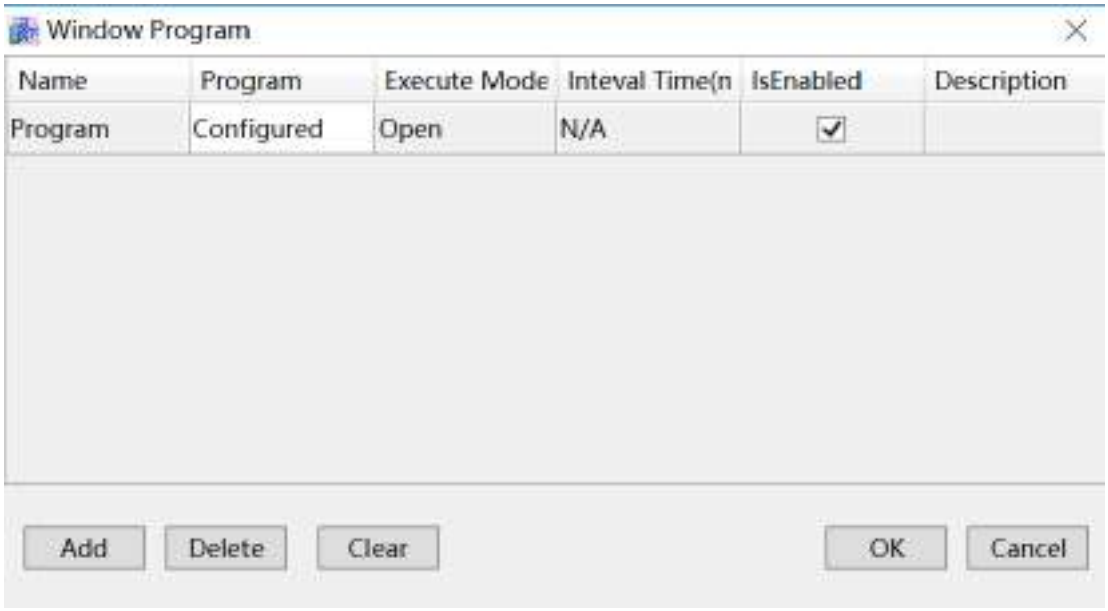
Step 3: Click the drop-down box below the Excution Mode to choose the Excute Mode, as shown in the figure below:



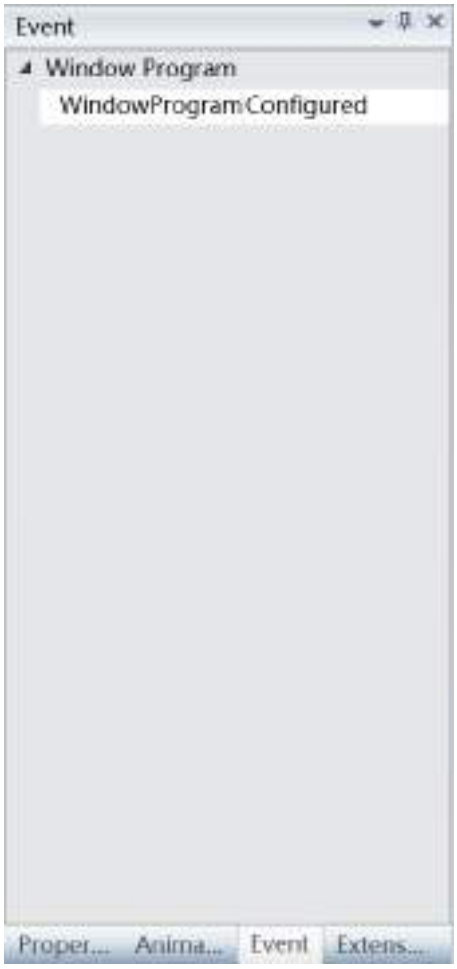
Step 4: Click the button below “Program”，pop up the script editor window ，Script can be written in it ,as shown in the figure below:



Step 5: Once the script is written and checked with no errors, click “Save and exit button” to complete event configuration; information on the configured event will be displayed in the Program, as shown in the figure below:



Step 6: Click “OK” button after the configuration is done, information on the configured event will be displayed in the event window, as shown in the figure below:



10.10 Control event

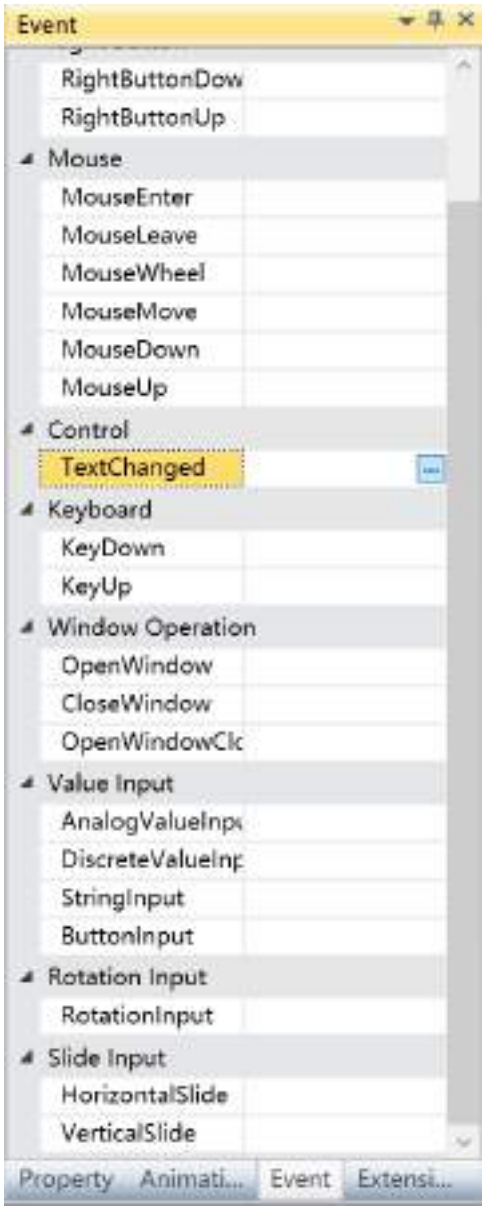
Control event is an unique event to a form control. Different controls have different events.

For example, the text box has a "TextChanged" event, and the check box has "checked" and "unchecked" events. All events can be implemented through editing script.

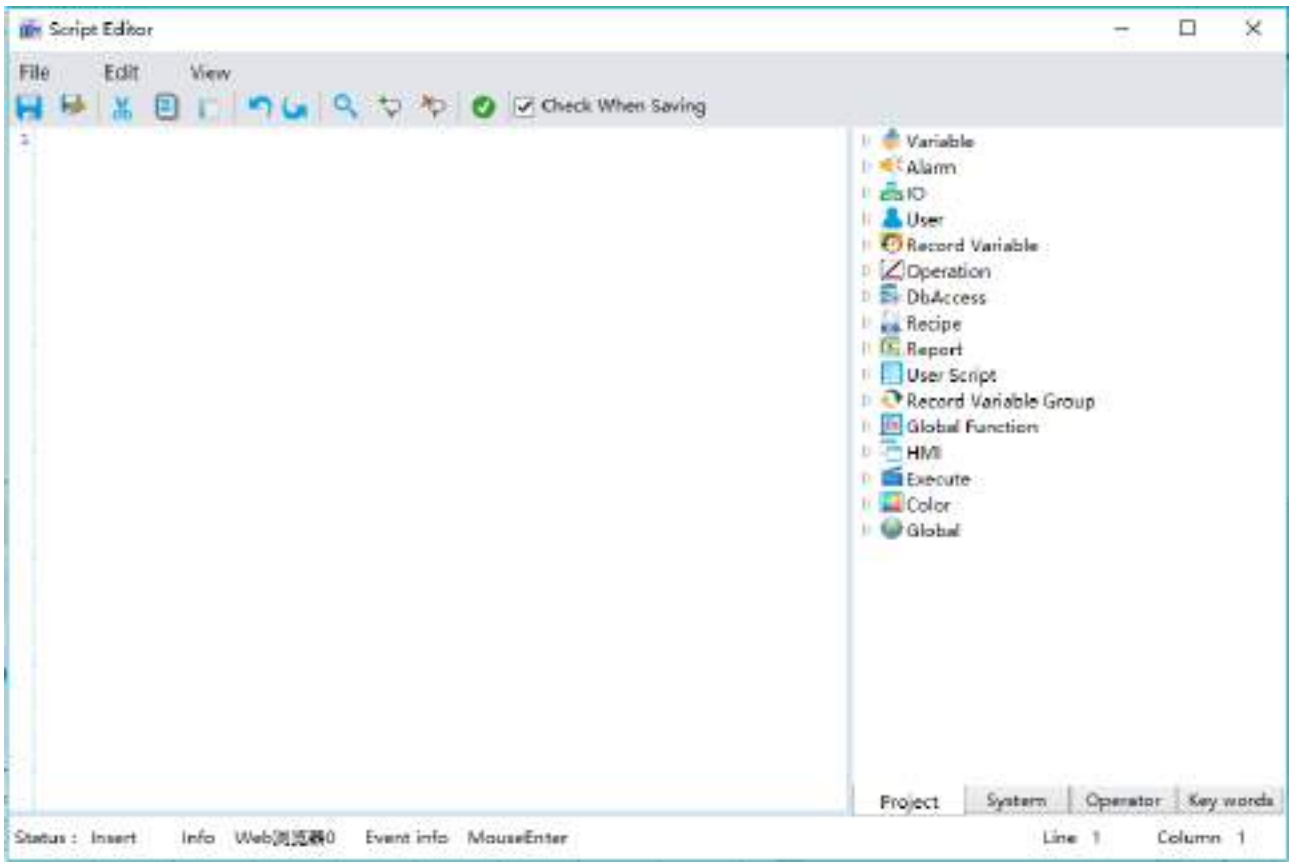
The followingt through the "TextChanged" event of the text box as an example to illustrate the configuration process of the event.

The configuration process is as follows:

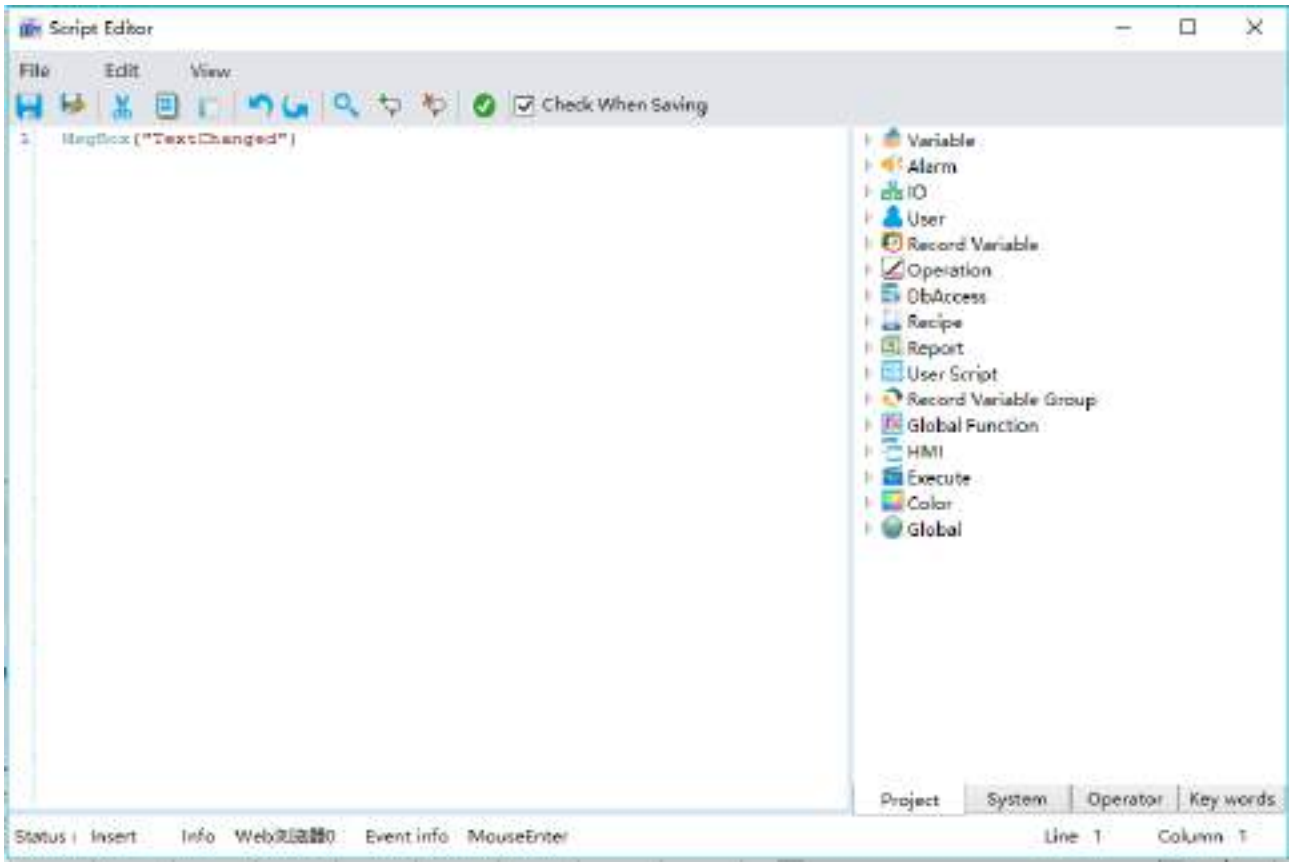
Step 1: Open the project window where the animation needs to be configured under the DIAView software development environment → select graphic object to configure event in the sketchpad → open “Event” window → select “Control”→ click the button in “TextChanged” bar, as shown in the figure below:



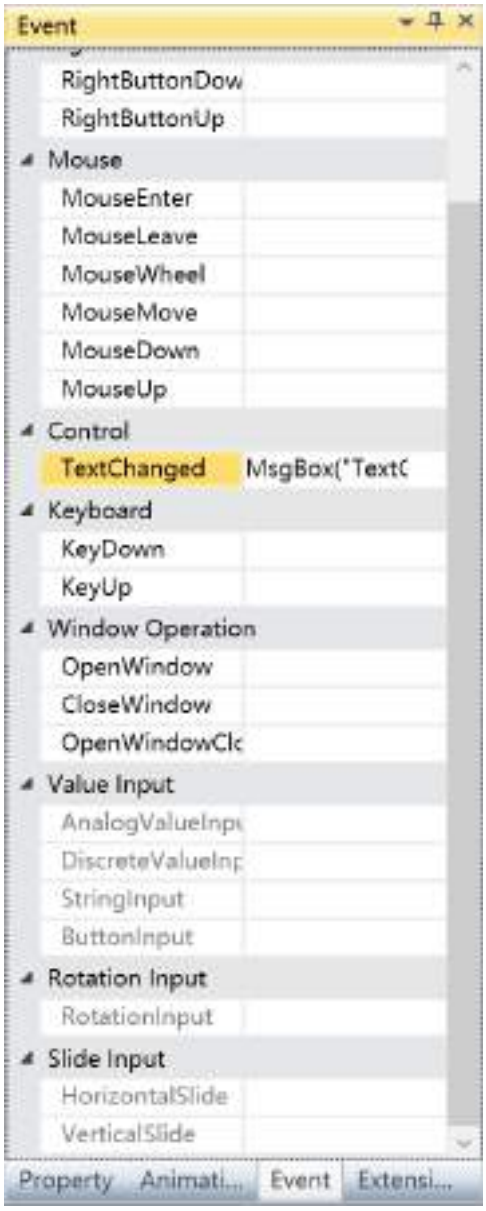
Step 2: The Script Editor window will appear, as shown in the figure below:



Step 3: Write the script (The DIAView software uses the Visual Basic Script; the script editor will check for basic syntax errors), as shown in the figure below:



Step 4: Once the script is written and checked with no errors, click “Save and exit button” to complete event configuration; information on the configured event will be displayed in the event window, as shown in the figure below:



To delete an event, simply select the event in the event list and then right-click the mouse and select "Delete".

The Configuration process of other controls is the same as above.

- **Control events include:**
  - CheckedBox: Checked event、Unchecked event
  - ComboBox: SelectionChanged event
  - TextBox: TextChanged event

- PasswordBox: PasswordChanged event
- DateTimePicker: ValueChanged event
- DatePicker: SelectedDateChanged event
- Calendar: SelectedDateChanged event
- RecipeBrowser: SelectionChanged event
- AlarmWindow: SelectionChanged event

11. Alarm

11.1 Overview

An alarm is a kind of reminder when the data acquired by the system exceeds the warning value ,in order to prevent early warnings for dangers that might occur during the production process or equipment failures. Alarms have irreplacable function on system security control in industrial automation and control systems.

During the process of using the DIAView software, when the data of the field environment(Temperature, humidity, etc)or instruments equipment monitored and acquired by the system exceeds the system's preset range, the system will use setting methods to send alarm information and display them in the alarm window, such as sending E-mails or alarm sounds ,so that users can know the running status of the fied and instruments equipment immediately and operate corresponding strategies to the alarm in order to prevent accidents .

The DIAView software has perfect alarm functions to ensure safe and reliable of the industry system.

11.2 Alarm group

"Alarm variable" is when alarm conditions are configured for associated system variables so that the alarms can be generated when the variable value changes.

Alarm groups can achieve classification management for alarms. Related alarms can be classify into the same alarm group making it easier for the system to perform unified management and operations

for alarm groups. The alarm variables of the DIAView software usually belong to an alarm group. The alarm group is first created and then alarm variables are created in the alarm group.

Alarm group naming rules:

(1) Composed of English letters, numbers, Chinese characters and underline, and can only begin with an English letter or Chinese character

(2) Case insensitive

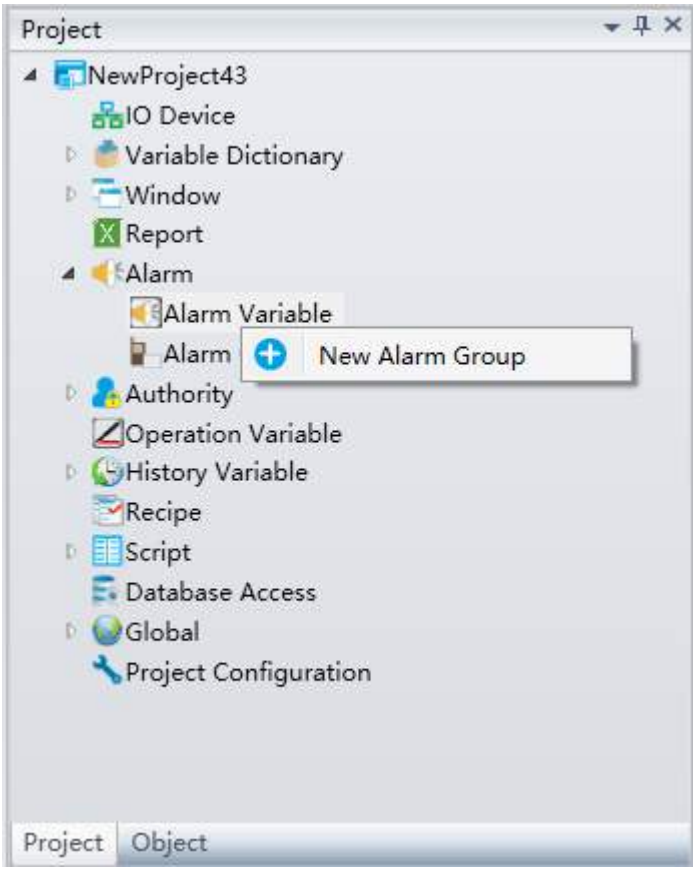
(3) The length cannot exceed 200 characters and cannot exceed 25 Chinese characters

(4) Alarm group names cannot be repeated under similar class nodes within a project

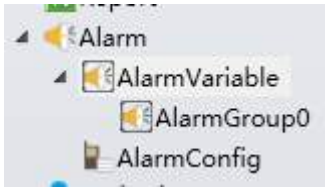
(5) If there are alarm variables and alarm groups under similar class nodes within the same project, the alarm variable and alarm group names cannot be repeated

Add alarm group:

Step 1: Open the project window under the DIAView software development environment → open the “Alarm” node in the project index tree → right-click on the “AlarmVariable” node → click the "New Alarm Group"option in the right-click menu, as shown in the figure below:

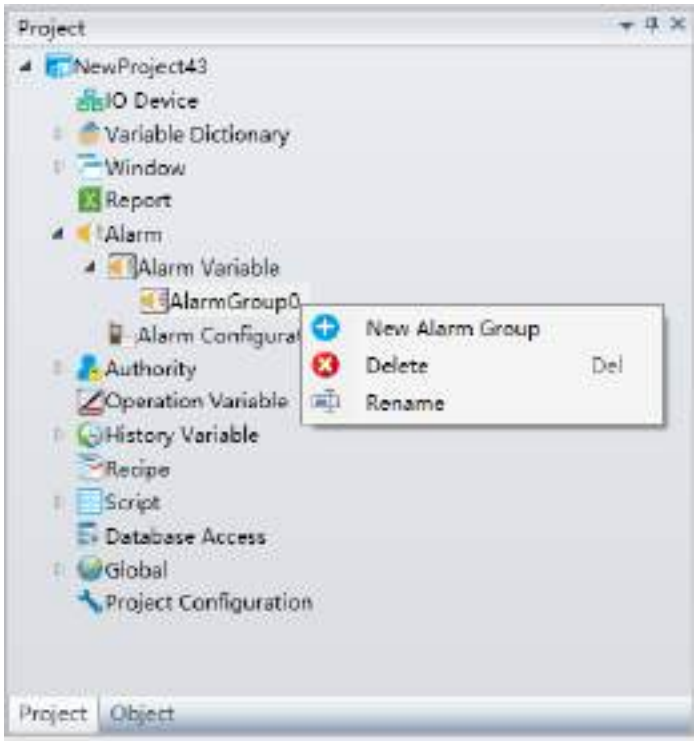


Step 2: After clicking the “New” button, the system will create an new alarm group sub-node with a default name; double click that alarm group sub-node to open the alarm variable configuration window (please refer to the next chapter for creating alarm variables), as shown in the figure below:



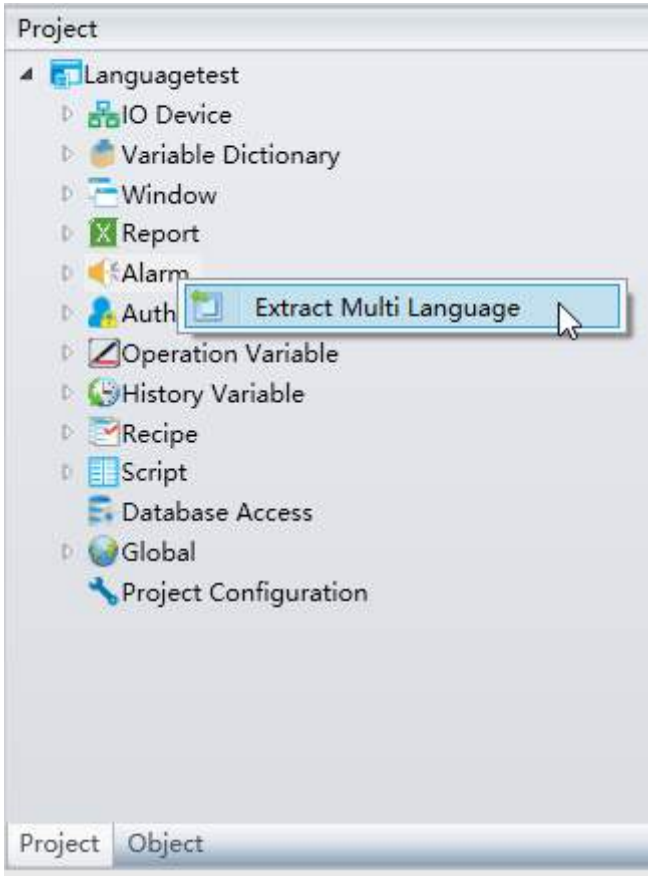
Other operations for alarm group:

Right-click the mouse on the new created alarm group sub-node and operations including “New Alarm Group”, “Delete” and “Rename” operation can be operated for the current alarm group, as shown in the figure below:



Right-click"Alarm" and select the multi-language button to extract the alarm text in the alarm

configuration into a string for customized editing, as shown in the figure below:



11.3 Alarm variable

1. Creating alarm variables

After opening the alarm group variable configuration window, click the “Add” button in the window to add alarm variables; the system will give a default name, as shown in the figure below:

✧ The meanings of each setting in the configuration window are as follows:

Name: Alarm variable name.

Naming rules:

(1) Composed of English letters, numbers, Chinese characters and underline, and can only begin with an English letter or Chinese character

(2) Case sensitive

(3)The length cannot exceed 200 characters and cannot exceed 25 Chinese characters

(4) Alarm variables in the same node or alarm group cannot have repeated names

(5) If there are alarm variables and alarm groups under the same node, the alarm variable and alarm group names cannot be repeated

Associated Variables: Associate the variables created in the project, which are the corresponding variables of the data acquired by the system; they are the basic data source for alarms. Alarm variables can only be configured for analog value or digital value variables.

Alarm Level: The alarm are divided into 5 levels: Slight, Lighter, General, Serious, Heavier.

Alarm Configuration: Sets alarm information including alarm type ,the alarm value limit and alarm text that corresponds to the alarm levels etc.

Description: Other information for the alarm variable.

✧ Function button :

“Add” button: Create new alarm variable.

“Insert” button: Inserts new alarm variable on the selected row.

“Delete” button: The “Delete” button is used to delete the selected alarm variable.

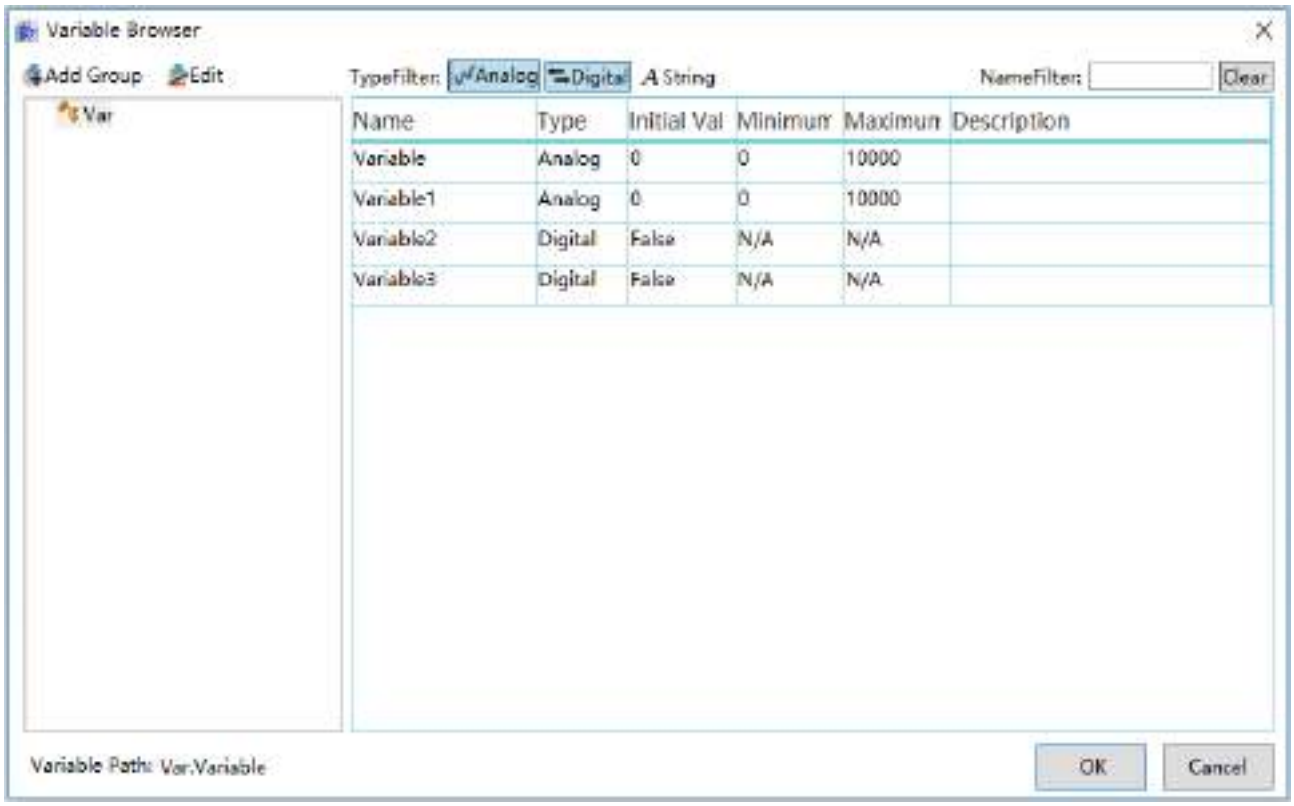
“Import” button: Imports the alarm variables from Excel to the system.

“Export” button: Exports the alarm variables from system to Excel.

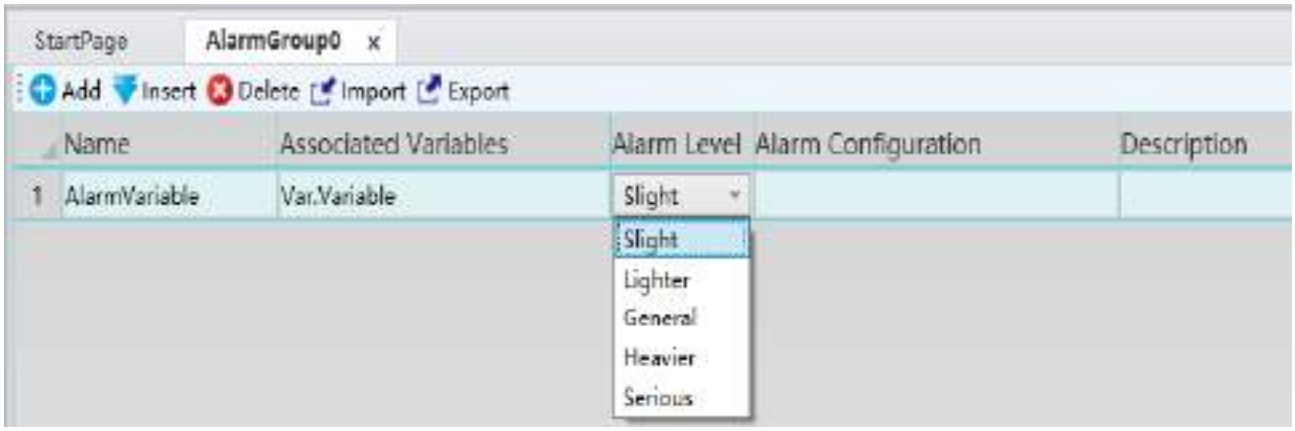
2.Configuring alarm variable

1. Associate variable: Click the button in associated variables cell and select the system variable to associate (only analog value and digital value type variables can be used to set alarms), as shown in

the figure below:



2.Alarm level: Click the button and a drop-down menu will appear, select a level, as shown in the figure below:

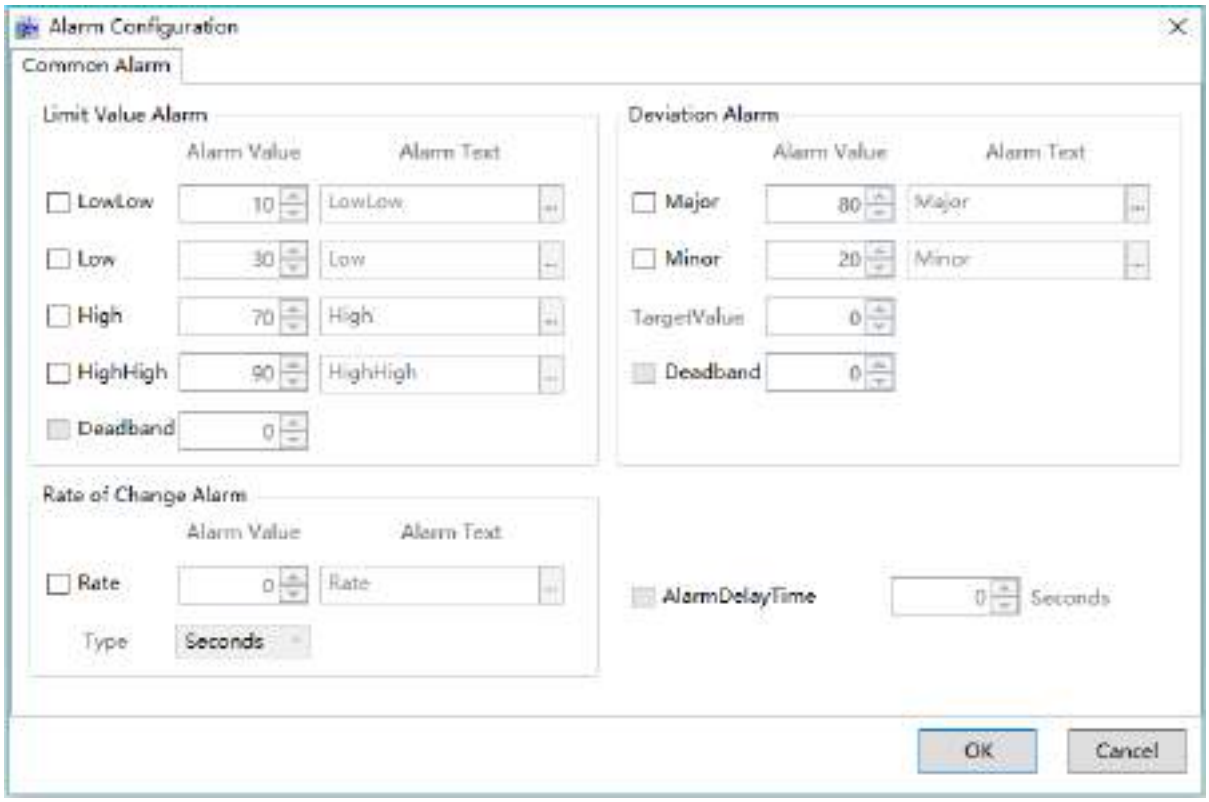


3. Alarm configuration: Click the button in the alarm configuration cell and the alarm configuration window will appear; the system will perform related configurations according to the variable type, as shown in the figure below:

- **Analog variable (Analog type variable) alarm configuration**

Analog variables are mainly integer and real type variables; click the button in the alarm

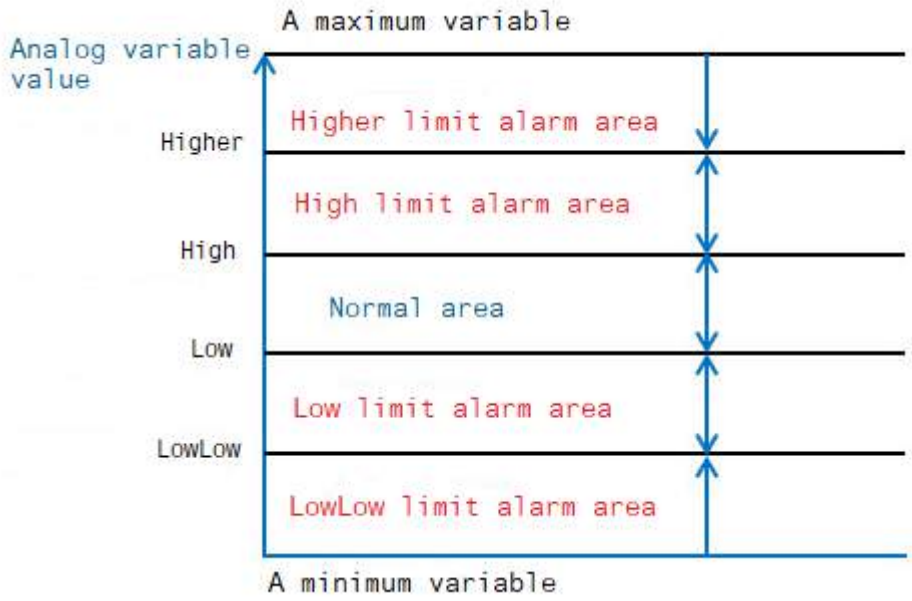
configuration cell and the analog variable alarm configuration window will appear, as shown in the figure below:



Analog variable alarms are divided into 3 types: Over-limit alarm, Deviation alarm and Rate of change alarm.Alarm Text supports multilingual configuration.

(1) Over-limit alarm

The alarm will be generated when the analog value exceeds the threshold of the alarm . There are 4 alarm thresholds for over-limit alarms: Lowlow, low, high, higher; their alarm thresholds should be set between the maximum value and minimum value of the variable; its alarm threshold and principles are as shown in the figure below:



When the variable value changes, if it exceeds a certain threshold value, over-limit alarms will occur immediately. Also, only one over-limit alarm will occur for one variable. For example: If the variable value exceeds the “higher alarm threshold value”, the higher alarm will occur and not the high alarm. If it went over-limit two times, the system will determine whether they are the same type of alarm first ; if they are, no new alarms will occur, or else the original alarm will be reset and then a new alarm will occur.

Over-limit alarm occurrence and recovery rules:

- ☞ Higher alarms will occur when greater than or equal to the higher threshold and recover when less than the higher threshold.
- ☞ High alarms will occur when greater than or equal to the high threshold and less than the higher threshold, and recover when less than the high threshold.
- ☞ Low alarms will occur when less than or equal to the low threshold and greater than the lowlow threshold, and recover when greater than the low threshold.
- ☞ Lowlow alarms will occur when less than or equal to the lowlow threshold, and recover when greater than the lowlow threshold.

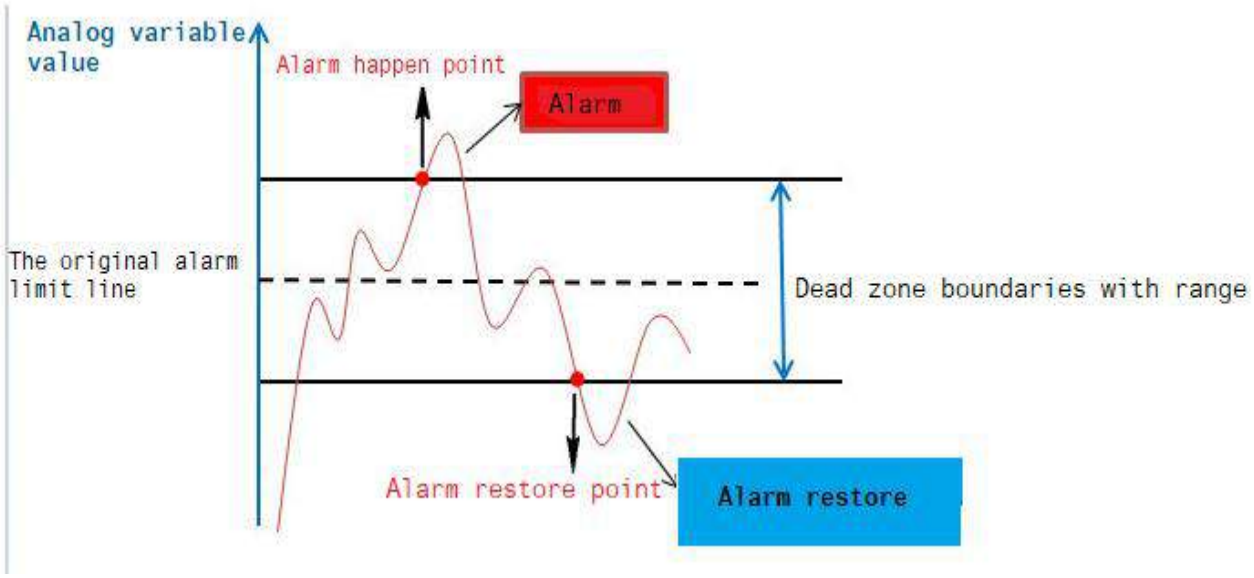
The over-limit alarm threshold is not selected under default conditions, one or multiple of the

checkboxes in front of the alarm threshold can be selected when needed. Then the editing box will become editable and “Alarm value” and “Alarm text” settings can be performed in it. The meaning of its properties are as follows:

Alarm value: Sets the threshold value of the alarm and defines the rules that should be followed when the threshold value is reached: minimum value<=Lowlow<low<high<Higher<=maximum value.

Alarm text: Sets the descriptive text of the alarm information.

Dead band: The effect of the dead band is to prevent unrealistic alarms from being generated when the variable value changes frequently between the original maximum and minimum alarm threshold by adding a threshold value for the maximum and minimum alarm threshold, changing the original alarm threshold from a single line to an alarm threshold band. When the value of the variable changes within the alarm threshold band , alarms will not occur or reset; alarm information will only occur when it exceeds the alarm threshold band . It is important for eliminating invalid alarms of fluctuating signals. The dead band value must between 0 and the value of the maximum value minus the minimum value; the dead band should also be in between the difference of any two threshold values. The principles of the alarm dead band are as shown in the figure below:



(2) Deviation alarm

This is the alarm that occurs when the ratio of the analog value to the deviation target value fluctuation exceeds the changing range . Deviation alarms are divided into two types: Large deviation (Major)and Small deviation(Minor). When the fluctuating value exceeds the Large/Small deviation range Large deviation alarms and Small deviation alarms will occur respectively. Also, only one deviation alarm will occur for each variable; its calculation method and principle are as follows:

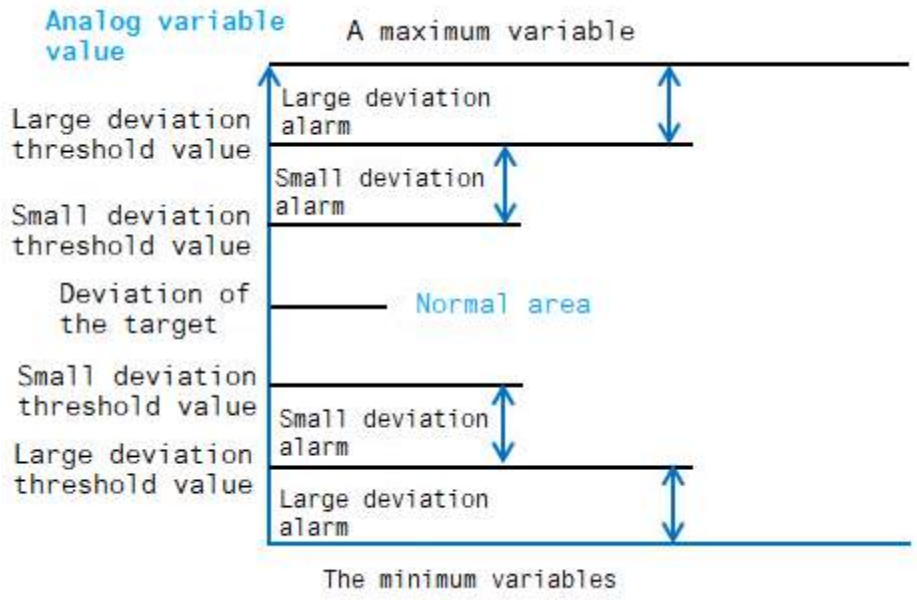
- ☞ Small deviation alarm threshold value=deviation target value ± Small deviation alarm value.
- ☞ Large deviation alarm threshold value=deviation target value ± Large deviation alarm value.

✧ When greater than or equal Small deviation alarm threshold value, Small deviation alarm will occur.

✧ When greater than or equal to the Large deviation alarm threshold value, Large deviation alarm will occur.

✧ When less than or equal to the Small deviation alarm threshold value, Small deviation alarm will occur.

✧ When less than or equal to the Large deviation alarm threshold value, Large deviation alarm will occur.



The deviation alarm is not selected under preset conditions, one or two of the check boxes in front of the types can be selected when needed. After which, the editing block behind will become editable and “Alarm value” and “Alarm text” settings can be performed for it:

Small deviation: Sets the Small deviation value;

Large deviation: Sets the Large deviation value；

Target value: Sets the deviation target value for when the deviation alarm occurs; it is used

together with the Large and Small deviations. Please refer to the calculation method of the Large and Small deviations Please refer to the “Over-limit alarm” settings for other items.

Deadband : Sets the deviation band of the alarm threshold value, which means that when deviation alarm occurs, if the variable value is within the alarm threshold value ± dead band value range, no new alarms will occur; or else the alarm will first be reset and then a new alarm will occur.

(3) Rate of change alarm

This is the alarm that occurs when the change rate of the analog value exceeds the value set for a period, which means that the alarm that will occur when the variable value changes too quickly. During software running, the changing speed of the variable value will be calculated each time to determine whether an alarm will occur. It is mainly used to monitor the change rate of the variable.

The rate of change alarm uses time as the unit and is divided into 3 types: hour, minute and second. Its calculation method is as follows:

Change rate = ((current variable value — previous variable value) * 100) / ((time the current value was generated — previous time the value was generated) *(maximum value of the variable — minimum value of the variable) * value that corresponds to the alarm type unit).

Definition of the “the value that corresponds to the alarm type unit” is:

If alarm type is second, the value is 1;

If alarm type is minute, the value is 60;

If alarm type is hour, the value is 3600

The change rate alarm is not selected under default conditions, check boxes in front of the rate of change can be selected when needed. After which, the editing block behind will become editable and “Alarm value” and “Alarm text” settings can be performed for it:

Change rate: Sets the change rate value.

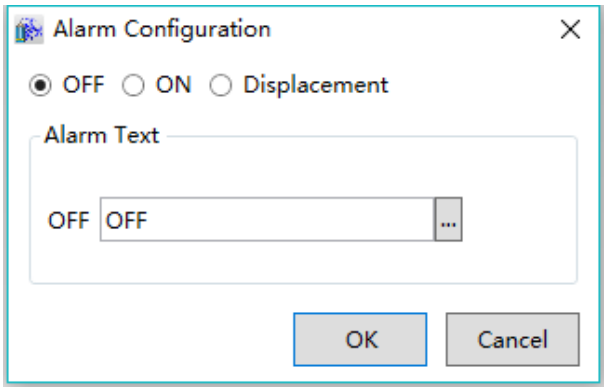
Type: Sets rate of change alarm type unit; the rate of change alarm uses time as the unit and is divided into 3 types: hour, minute and second.

Over-limit or deviation alarm delay: Sets a delay time (unit: second) so that when an alarm occurs the system will not display it immediately and instead perform a delay; if the alarm was reset or disappeared within the delay time, that alarm might be a false alarm and the system will clear it automatically. If the alarm still exists after the delay time, then that alarm is a real alarm and the system

will display and record it. If a new alarm occurred within the delay time, the delay timer will restart.

- **Digital variable (Digital type variable) alarm configuration**

Digital variables are Boolean type variables; Click the button in the alarm configuration cell and the alarm configuration window will appear, as shown in the figure below:



There are 3 types of digital variable alarms: On alarm, Off alarm and displacement alarm.

- ✧ “On”: Alarm will occur when the variable value changes from off to on (changes from 0 to 1)
- ✧ “Off”: Alarm will occur when the variable value changes from on to off (changes from 1 to 0)
- ✧ “Displacement”: Alarm will occur when the variable value changes no matter whether from off to on or on to off

11.4 Alarm window

Alarm window is used to display alarms information; once an alarm is configured, its information must be viewed only by the alarm window, therefore they must be used together. Once an alarm window is drawn, it will associate with all configured alarm information automatically without other configurations. Only the alarm type color effect and the properties of the alarm windows need to be configured to display, such as alarm levels. etc. When the system runs, it will display alarm information in runtime and allows query historic data of alarm.

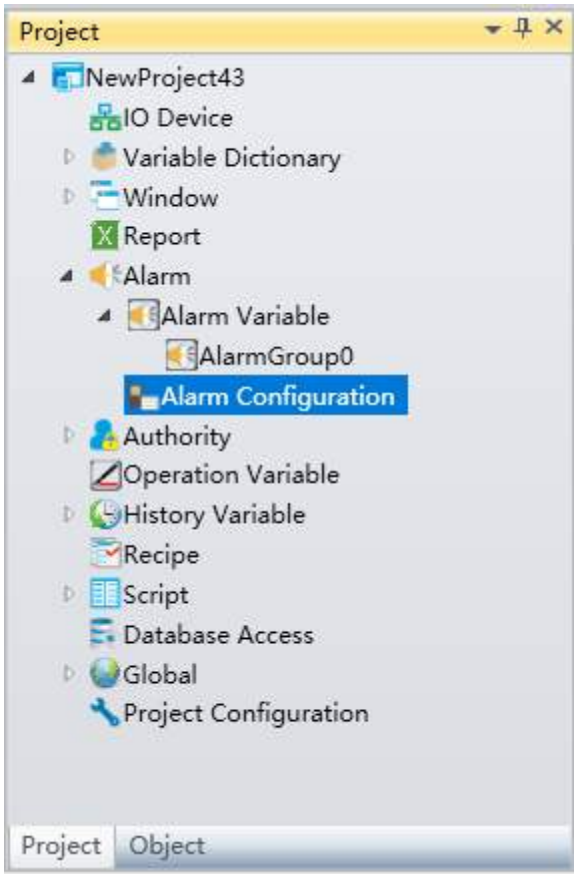
Please refer to chapter “7.7.10 AlarmWindow” on how to draw alarm windows and configure properties of alarm windows.

11.5 Alarm configuration

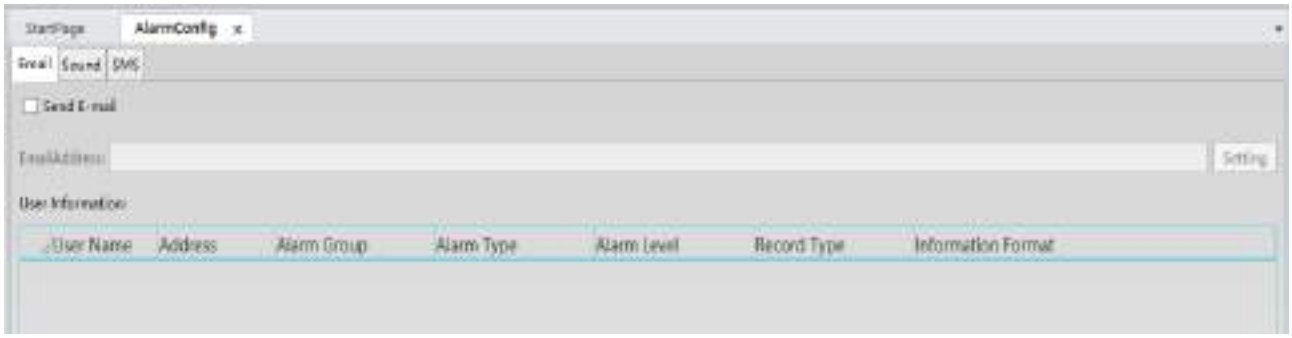
Configure E-mail, sound and mobile phone SMS alarm functions for alarm and alarm groups.

Alarm configuration:

Step 1: Open the project window under the DIAView software development environment → open the “Alarm” node in the project tree → double-click the “AlarmConfig” , as shown in the figure below:



Step 2: After double-clicking the “AlarmConfig”, the configuration window will appear, as shown in the figure below:



It includes three configurations, “E-mail”, “Sound” and “Mobile phone SMS”:

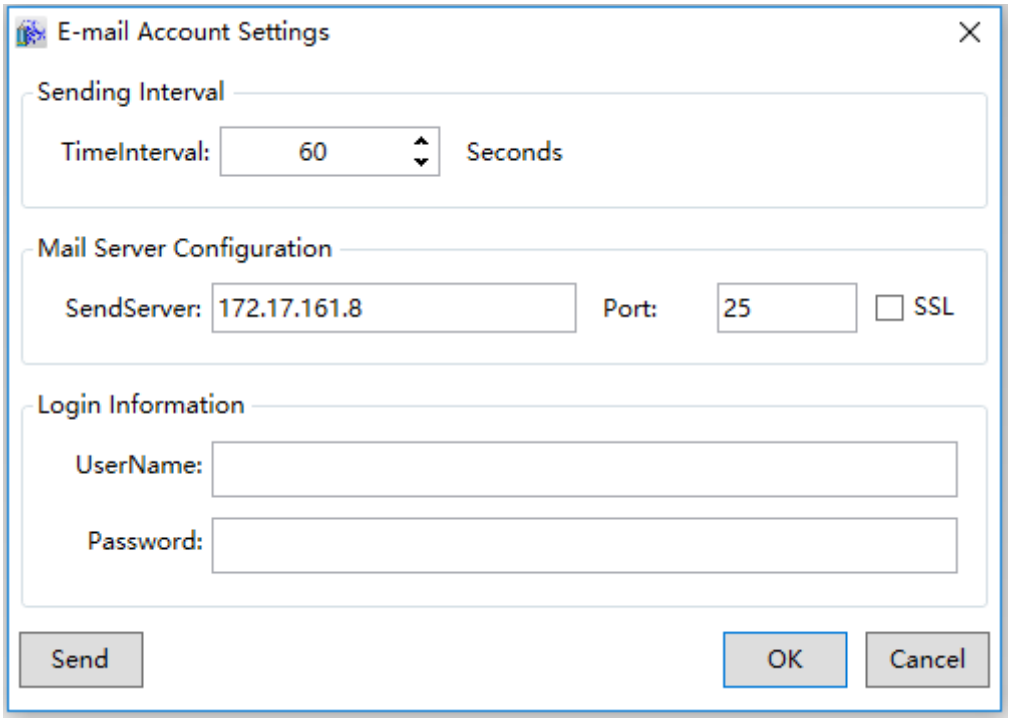
1. E-mail: Sends alarm information to related personnel through E-mail.

Open the “E-mail” option and the configuration steps are as follows:

Step 1: Selected the “Send e-mail” check box; it means that the send e-mail function is enabled when selected and the next steps of the configuration can be performed;

Step 2: Input the E-mail address to send the alarm information in the “Send server” field;

Step 3: Click the “Settings” button ,and the “E-mail account settings” window will appear , as shown in the figure below (taking the 163 E-mail server as an example):

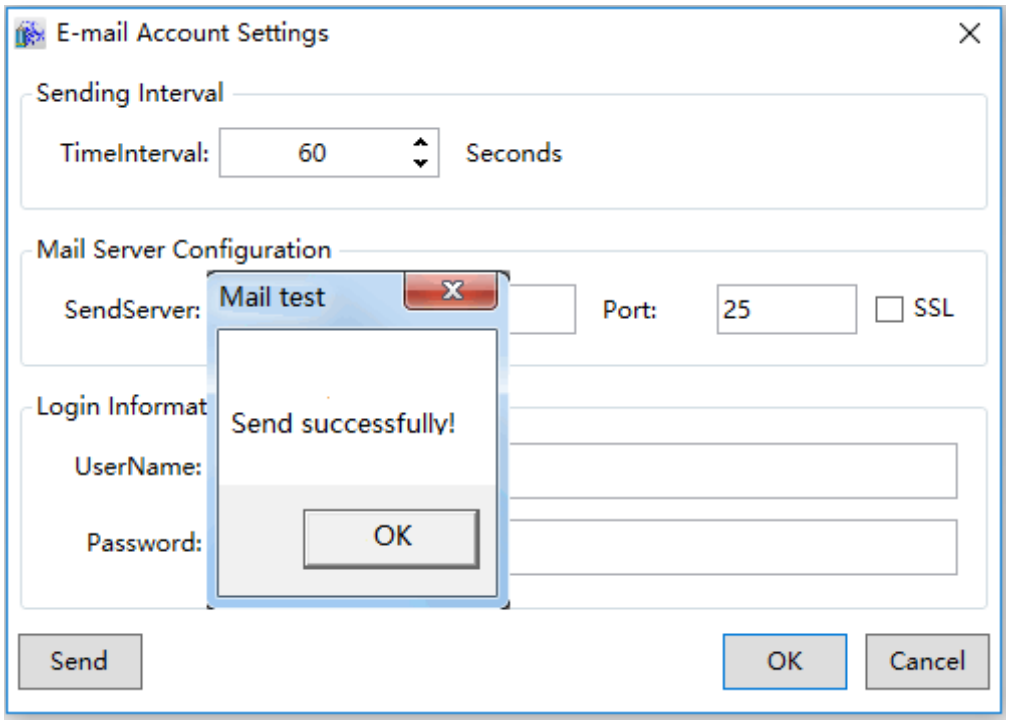


The meanings of each configuration in the window are as follows:

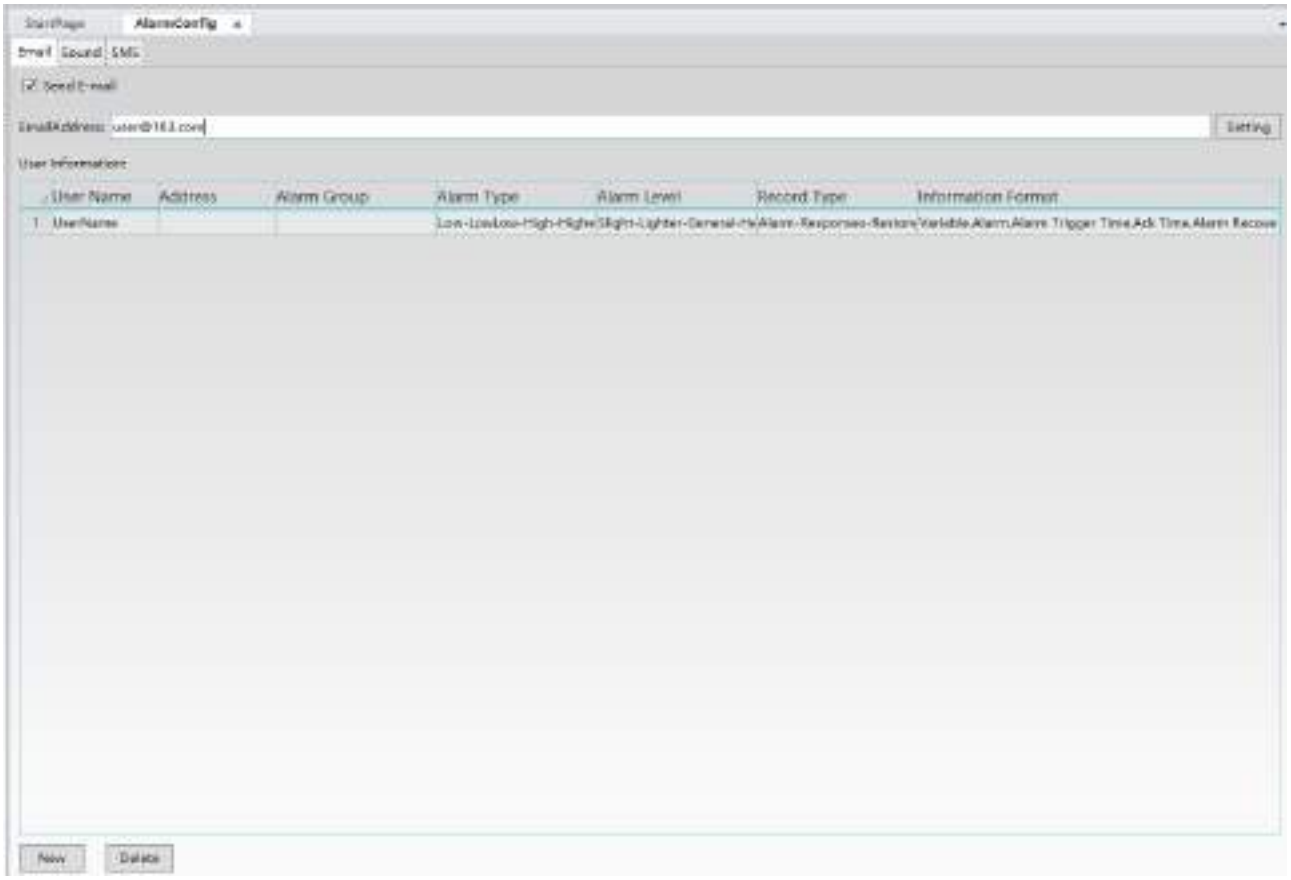
- TimeInterval: Sets the time interval to send E-mails; the alarm information will be sent once a new alarm occurs within the time interval.
- SendServer: Configures the address and port information of the E-mail to send .
- UserName: The login user name of the E-mail account used to send the alarm information (which is the user name of the E-mail set in Step 2).
- Password: The password used to login the E-mail account used to send the alarm information

(which is the password of the E-mail set in Step 2).

- Send: Test whether the E-mail server settings are correct and whether the E-mail can be sent; the rest result is as shown in the figure below:



Step 4: When the E-mail account setting is complete, press the “Add ” button to add the information of the user to receive alarm information:



The meanings of each field in the configuration window are as follows:

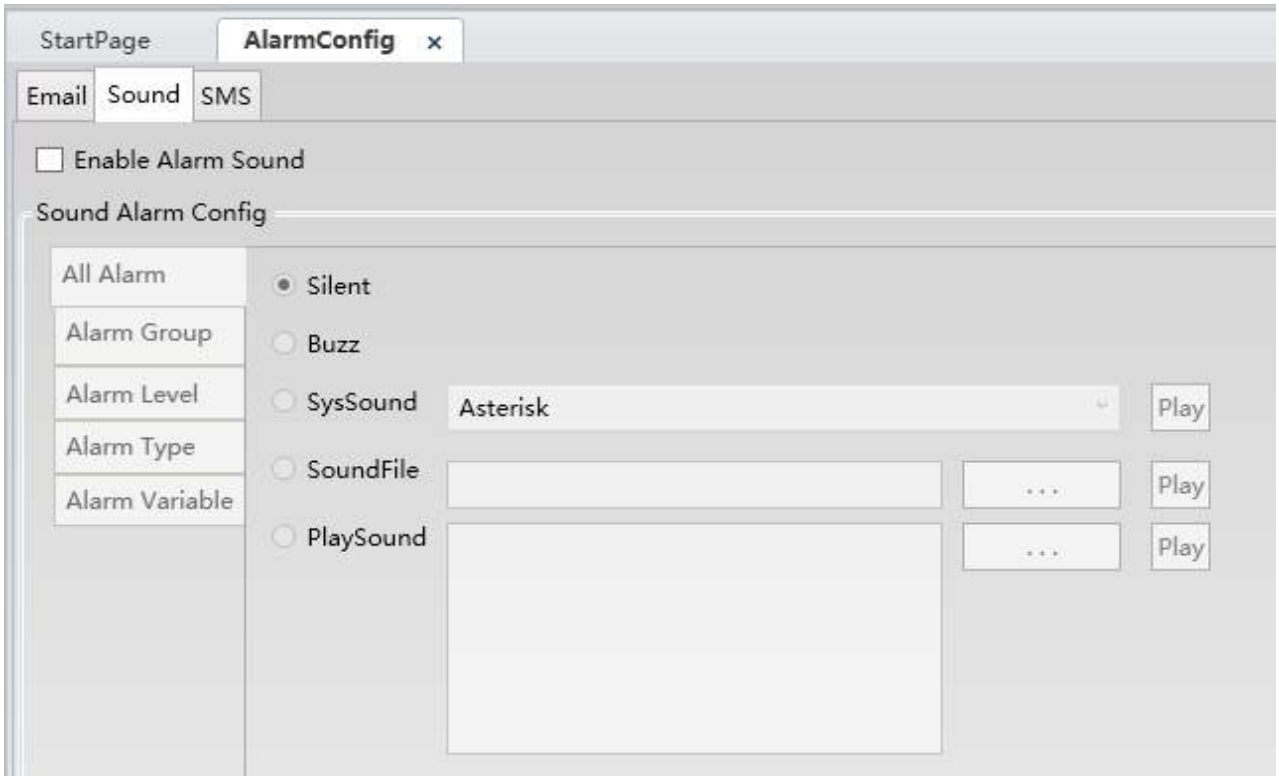
- User Name: User name of the recipient.
- Address: E-mail address of the recipient.
- Alarm Group: Select the alarm group for alarm information to be sent when alarms occur.
- Alarm Type: Select the alarm type to send alarm information.
- Alarm Level: Select the alarm level; alarm information in this level and higher than this level will be sent.
- Record Type: There are 4 types: Alarm, confirm, reset and remove; the information of the selected types will be sent and information of the unselected types will not be sent.
- Information Format: Sets the content, order, date and time format of the alarm information sent.

2. Sound: Configure the sounds to play for different alarm groups, alarm levels, alarm types and alarm variables when alarms occur.

Its priority is: All alarms < alarm group < alarm level < alarm type < alarm variable.

Open the "Sound" option and the configuring steps are as follows:

Step 1: Check the "Enable sound alarm" check box. It means that the sound alarm function is enabled and the next steps of the configuration can be performed, as shown in the figure below:



Step 2: Configure the various contents in the "Sound Alarm Config":

➢ All Alarm:

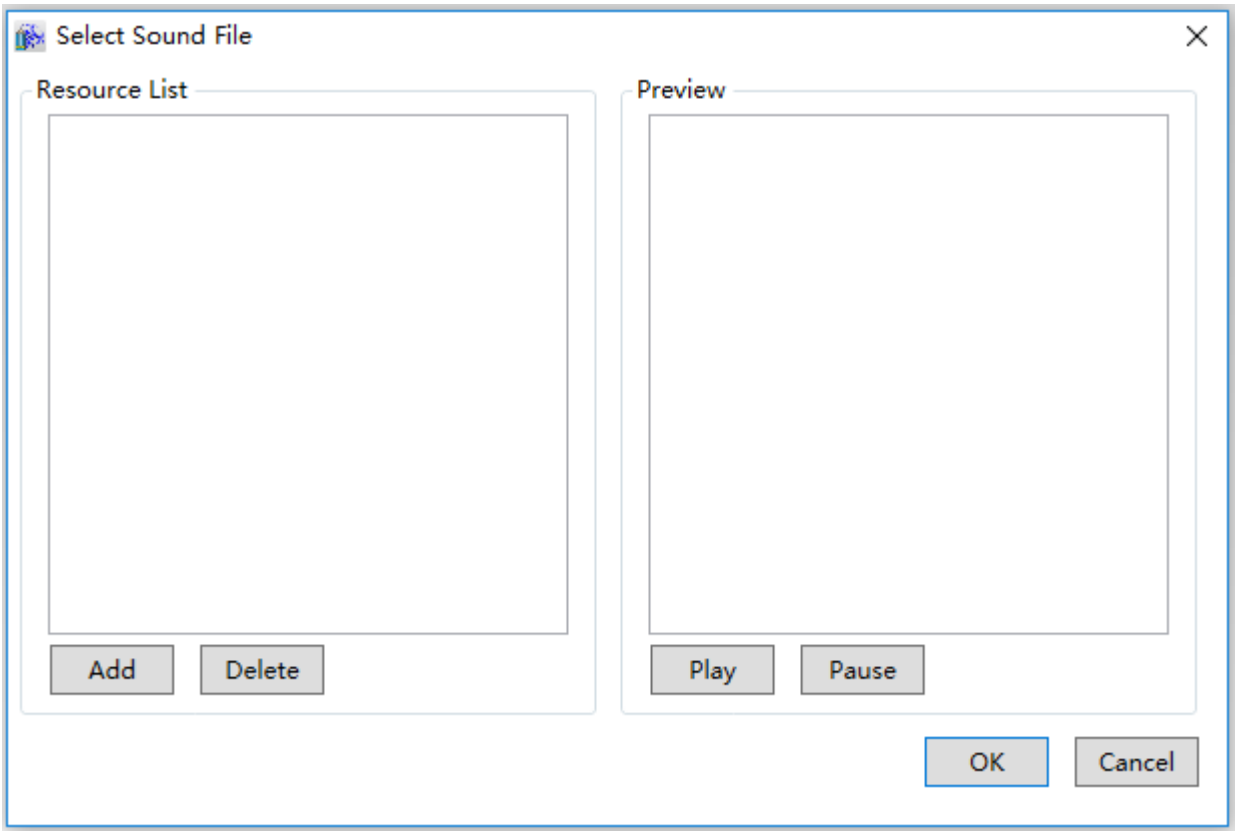
Sound will be used for All alarms generated by the system ; methods that can be selected include: no sound, buzzer sound, system sound, audio file and play audio etc.

- Silent: No sounds are played.
- Buzz: Buzzer sounds are played from the buzzer of the computer when the system generates an alarm. When multiple alarms are generated, the buzzer sound will sound in turns.
- SysSound: Uses the sound from the Windows system, which include the following five

sounds: “Asterisk (preset)”, “Beep”, “Exclamation”, “Hand“，”Question”.

- SoundFile: Users can customize sound file sources so that the selected sound file will be played when an alarm is generated. Press the 

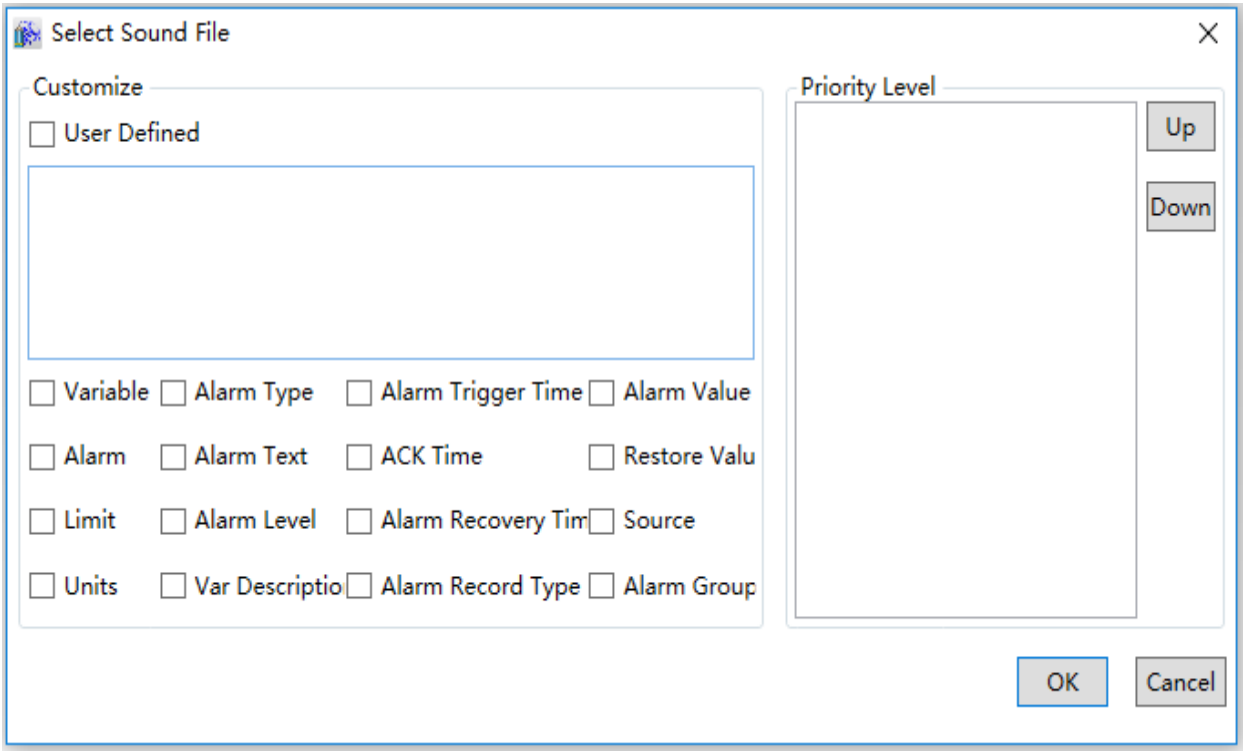
button and the sound source selection window will appear, as shown in the figure below:



The “Add” button used to import external sound files, and press the “Play” button allows you to listen to the file.

- PlaySound: Plays the customized sound content. Pressed the 

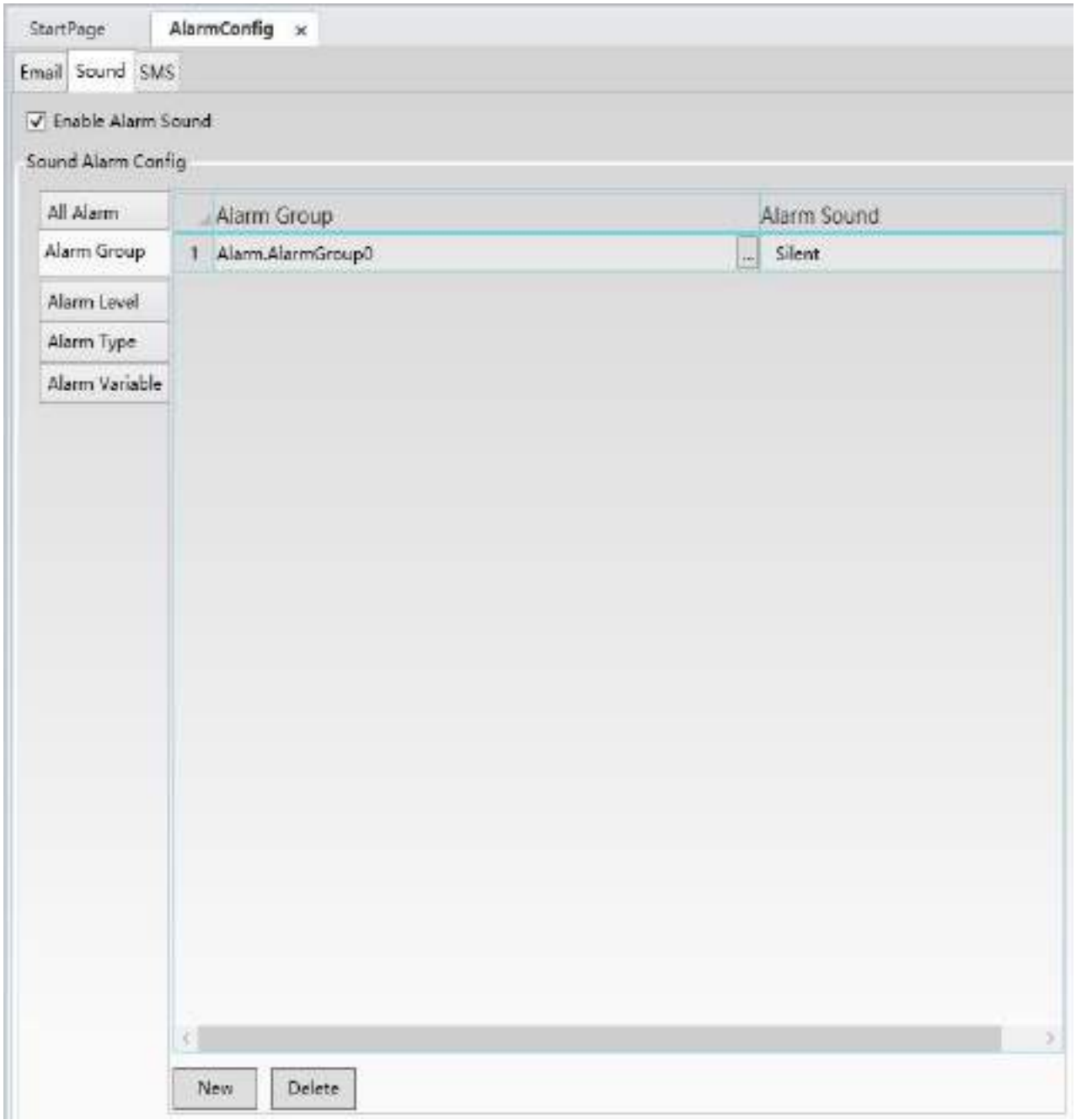
button and the sound configuration window will appear, as shown in the figure below:



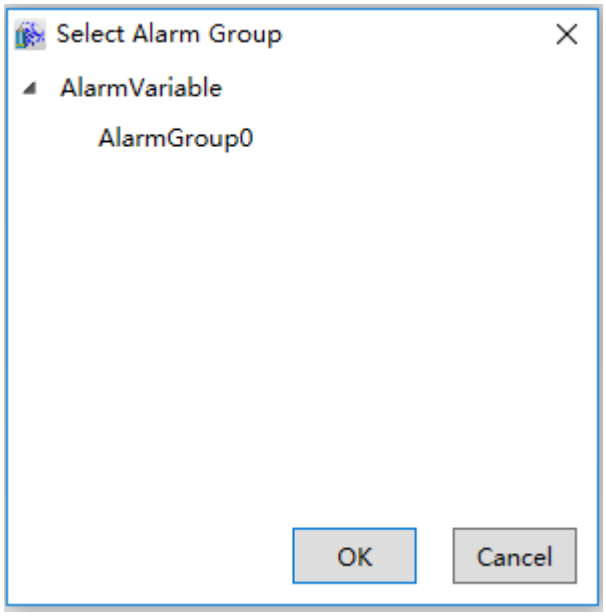
It allows customized sound contents and it supports Chinese and English. Select the alarm related configuration information and adjust the sound playback order; when an alarm is generated, the system will play sound contents in order.

- Alarm Group:

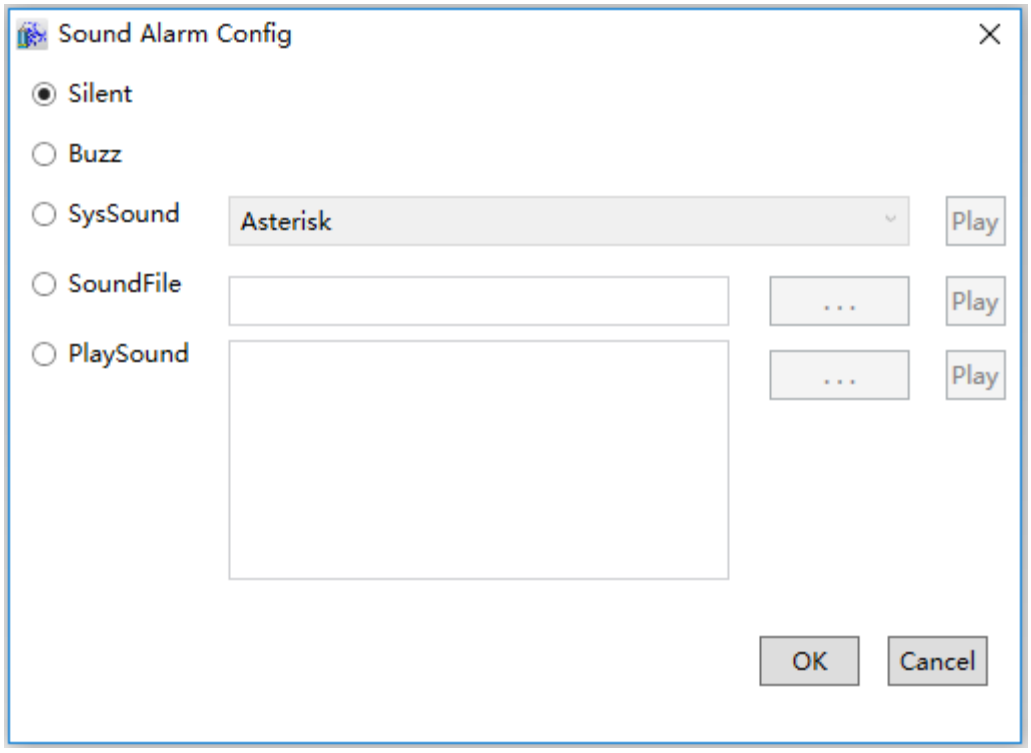
Configure the alarm group that need sound alarm. The “New” and “Delete” buttons can be used to select the alarm group, and use the button in the “Alarm group” field to select the alarm group that needs to configure sound alarms; click the button in the “Alarm sound” field to configure alarm sounds, as shown in the figure below:



✧ Select alarm group, as shown in the figure below:

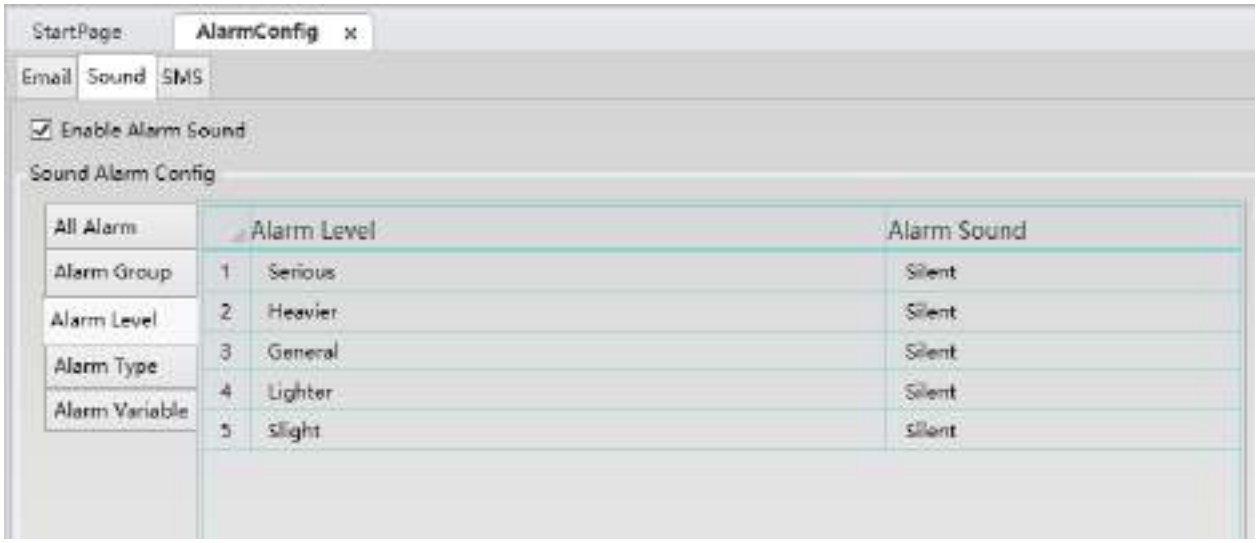


✧ Configure the alarm occurrence sound, as shown in the figure below:



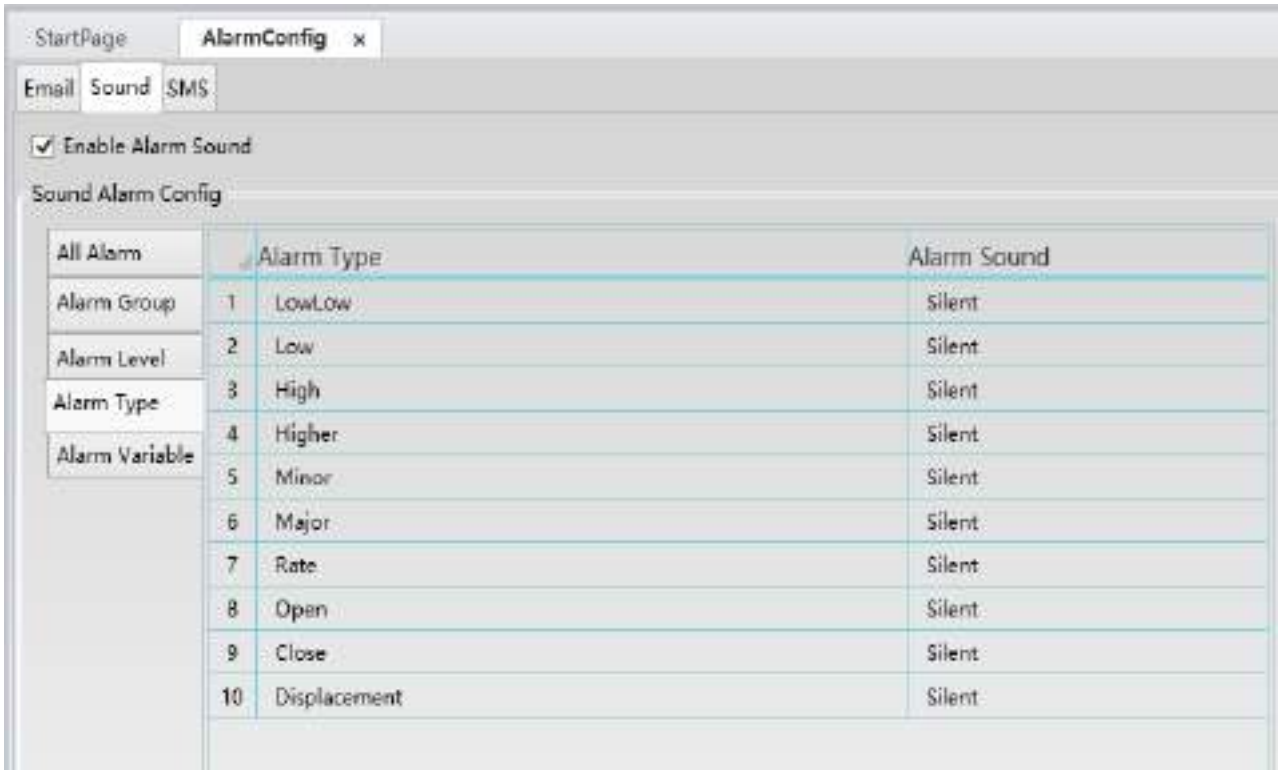
➢ Alarm level:

Set the sound alarm when different levels of alarms occur; use the button in the “Alarm Sound” field to configure alarm sounds, as shown in the figure below:



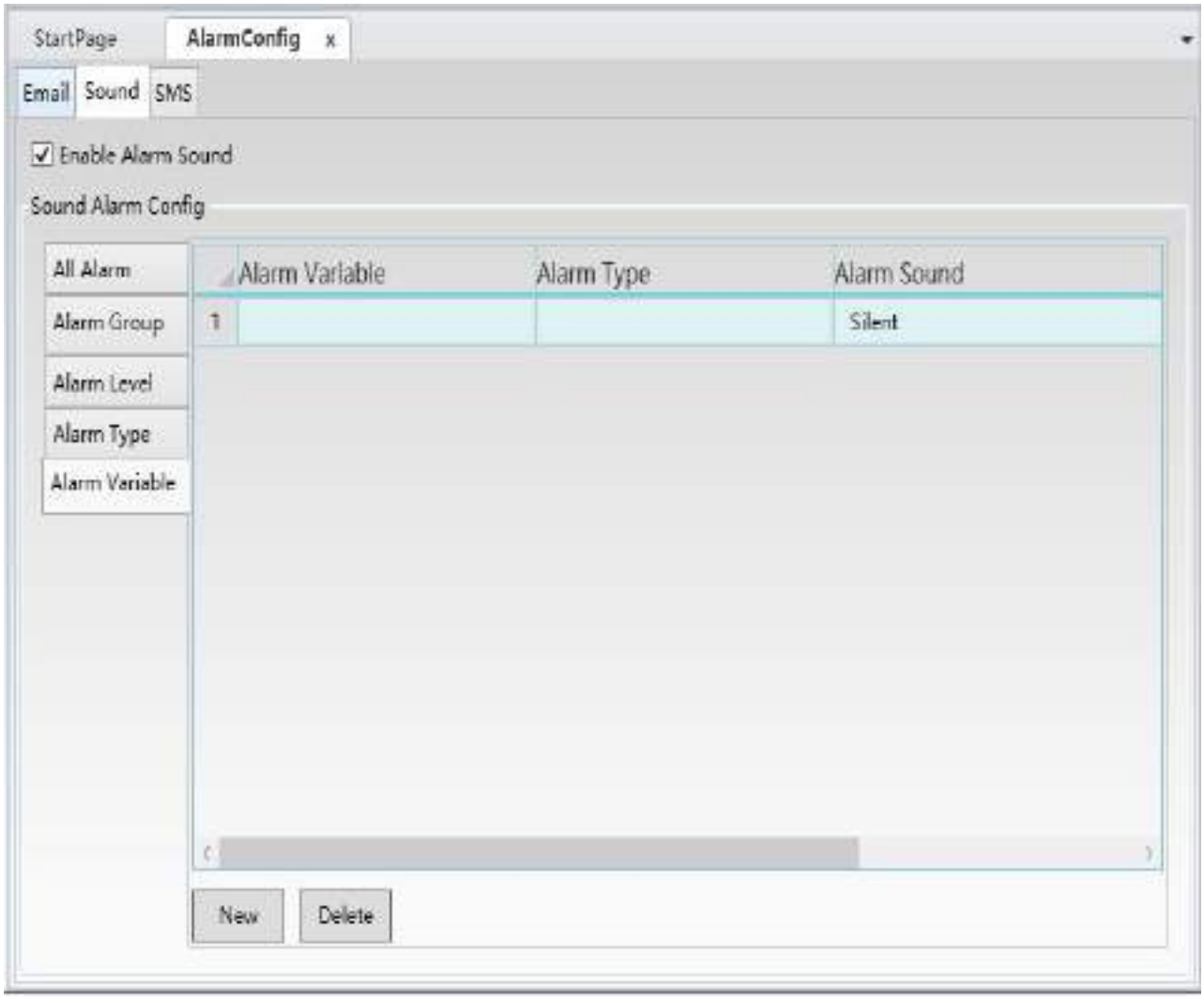
- Alarm type:

Set the sound alarm when different types of alarms occur; use the button in the “Alarm sound” field to configure alarm sounds, as shown in the figure below:



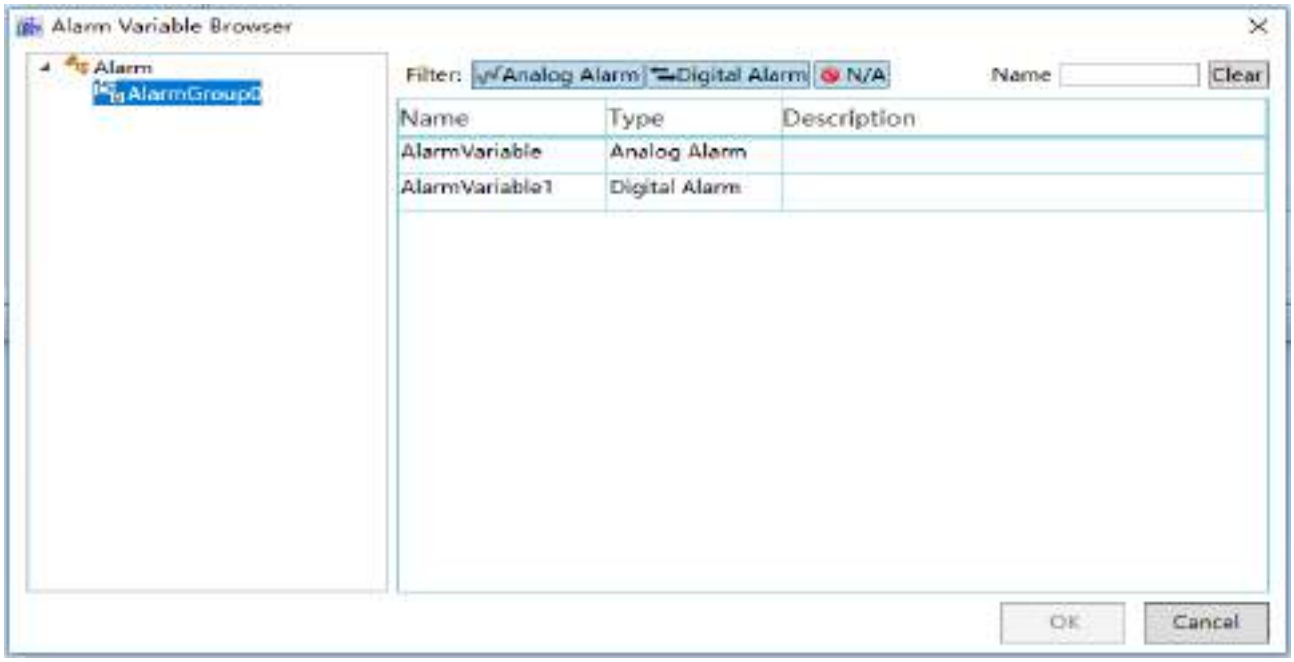
- Alarm variables:

Configurations can be performed here for alarm variables in the DIAView software that require additional sound alarms after an alarm occurs, as shown in the figure below:



Configuration process for each row: First use the “New” button to add a new alarm sound configuration row

✧ **Alarm variable:** Click the button in the row to open the alarm variable browser and select variables to associate to alarm viriables , as shown in the figure below:



✧ **Alarm type:** Selects the alarm type that needs sound alarm; it is a drop-down menu.

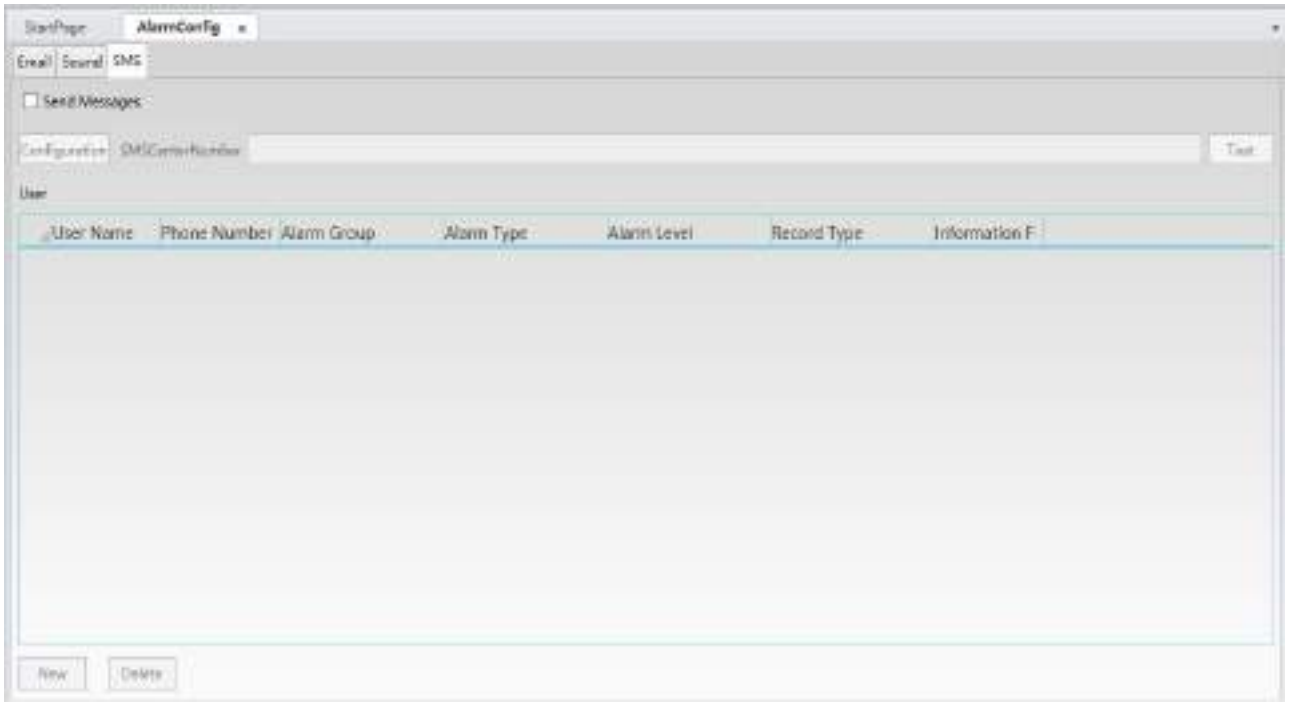
✧ **Alarm sound:** Click the button in the row to configure the sound to play when an alarm occurs.

✧ **Alarm recovery sound:** Click the button in the row to configure the sound to play when an alarm recovers.

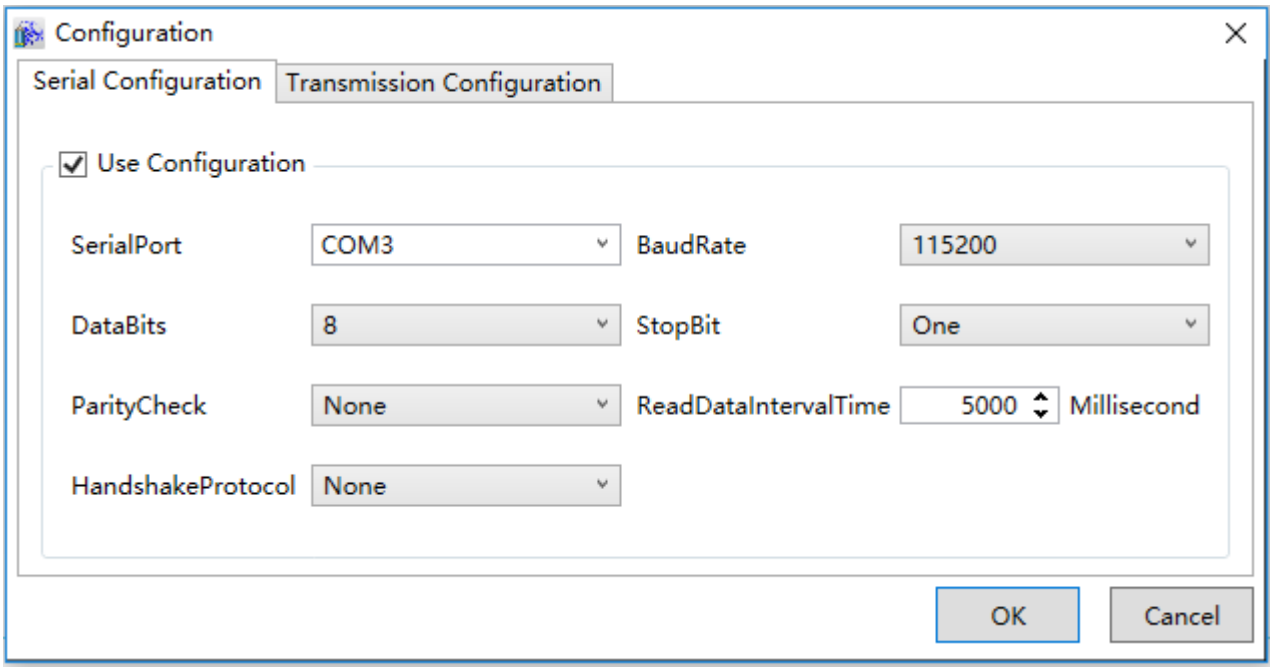
3. SMS Sends the alarm information to related personnel through mobile phone SMS.

Open the “SMS” option and the configuration steps are as follows:

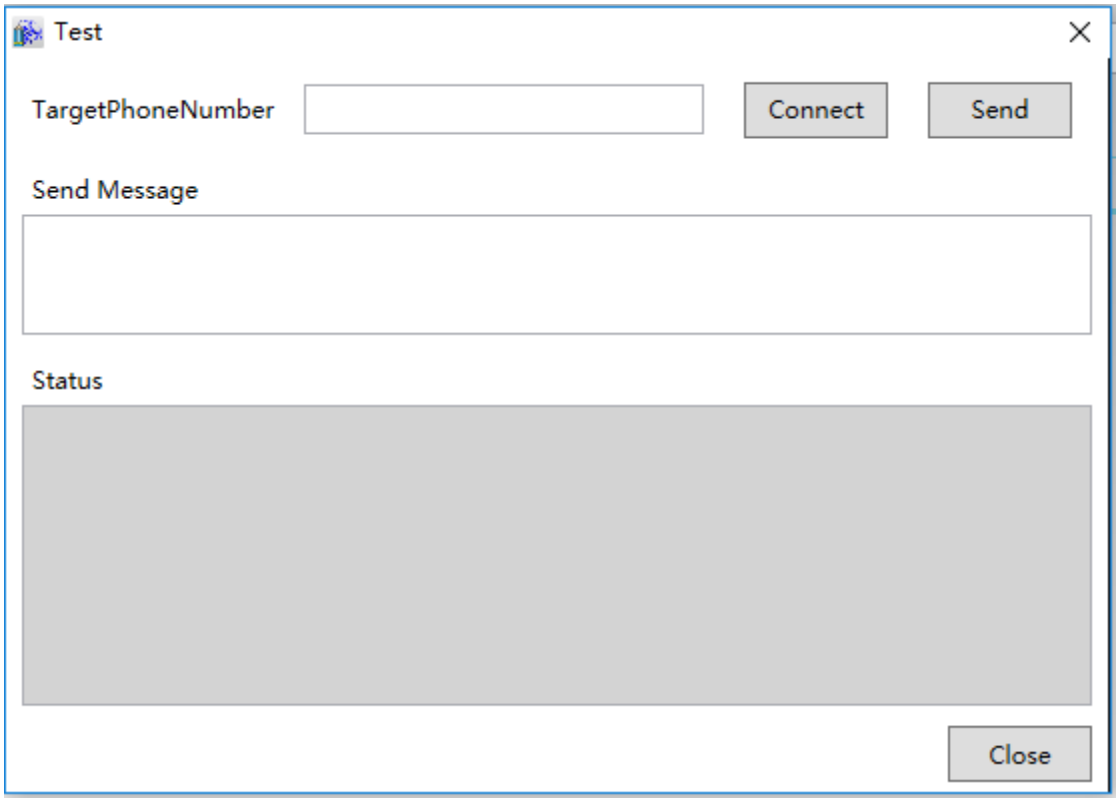
Step 1: Check the “Send messages” check box; it means that the send SMS function is enabled and the next steps of the configuration can be performed ,as shown in the figure below:



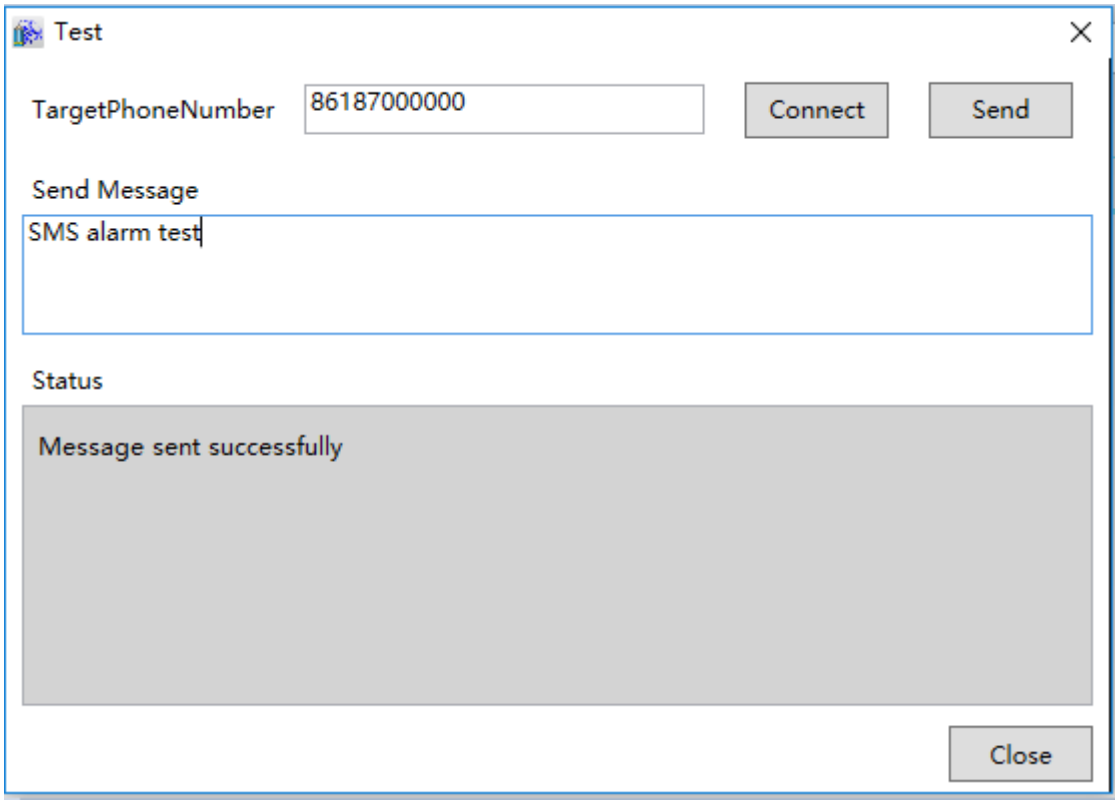
Step 2: Click the “Configuration” button to configure the information to sending SMS; the “Configuration” window will appear ,as shown in the figure below (the serial port is preset to COM3, change it to COM1: serial transmission rate is 115200):



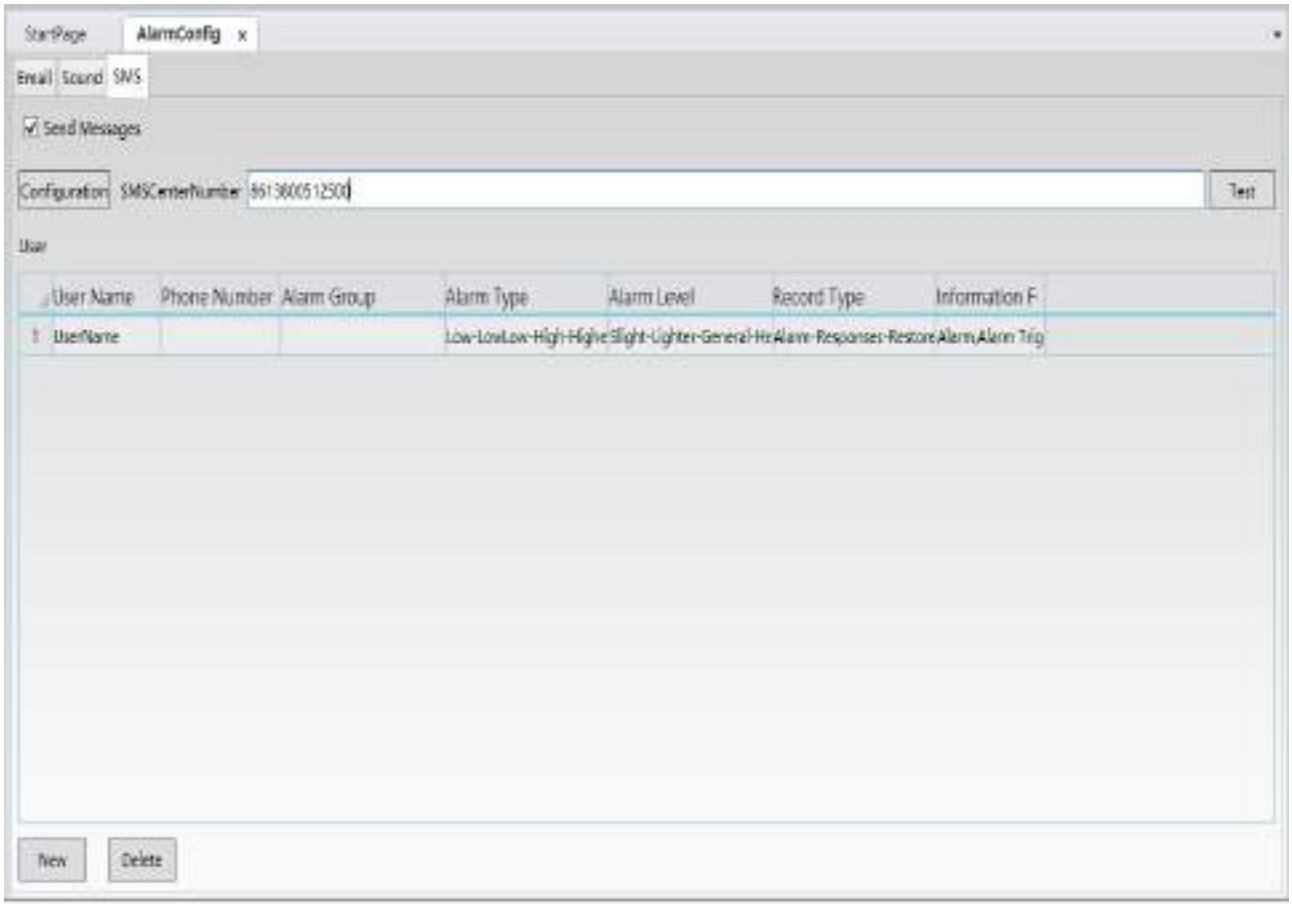
Step 3: Click the “Connect” button and the “Test” window will appear; click the “Send” button to display the current status, as shown in the figure below:



Step 4: Configure the “TargetPhoneNumber” and related information for sending SMS; click the “Send” button to display the current status, as shown in the figure below:



Step 5: When the mobile phone account configuration is complete, press the “New” button to add the information for the user to receive alarm information:



The meanings of each field in the configuration window are as follows:

- User Name: User name of the recipient.
- Phone Number: The phone number of the recipient.
- Alarm Group: Select the alarm group for alarm information to be sent when alarms occur
- Alarm Type: Select the alarm type to send .
- Alarm Level: Select the alarm level; alarm information in this level and higher than this level will be sent.
- Record Type: There are 4 types: Alarm, confirm, reset and remove; the information of the selected types will be sent and information of the unselected types will not be sent.
- Information Format: Set the content, order, date and time format of the alarm information to send.

12. User Authority

12.1 Overview

Users authority refers to the operating and control authority that a user has when using the DIAView ; it includes “Security zone” and “User”. User authority management can be used to guarantee safe and reliable operations and stable executions in order to prevent problems such as making wrong operations, unauthorized operations, or tampering with the project settings illegally.

Security zone is a strategy to protect the security of system operations; setting security zones in the DIAView development environment allows partition protection for the access operating authority of graphic objects. Security zone is usually used together with user; a user can have one or multiple security zone operating authorities, and each graphic object can belong to 1 or multiple security zone. During the system execution process, users can only access and operating graphic objects within the security zone the user has in order to prevent users from performing wrong operations and unauthorized operations, ensuring the usage security for system executions and operations.

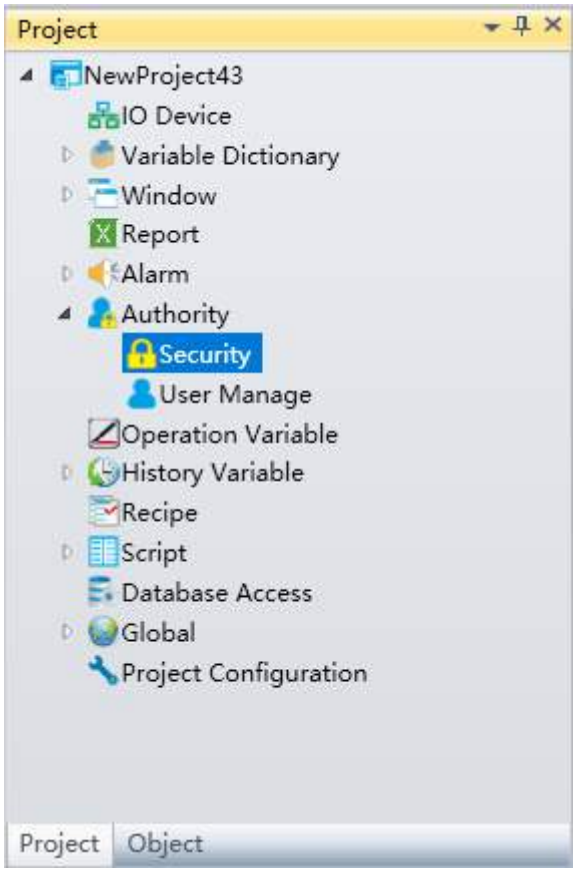
Users are operators and administrators with related operating authorities set for project development and execution; users in the DIAView software are divided into: “System administrator”, “Operator” and “Manager”. “System administrator” is default in the project; it has all operating authorities and the highest level. Each project only has one system administrator, and it cannot be deleted. “Operator” is the ordinary project operator; its operating authority can be set by setting the security zones. “Manager” not only has the operating authority to set security zones, it can also manage “Operator”; it can add, modify and delete “Operator” information.

12.2 Security zone

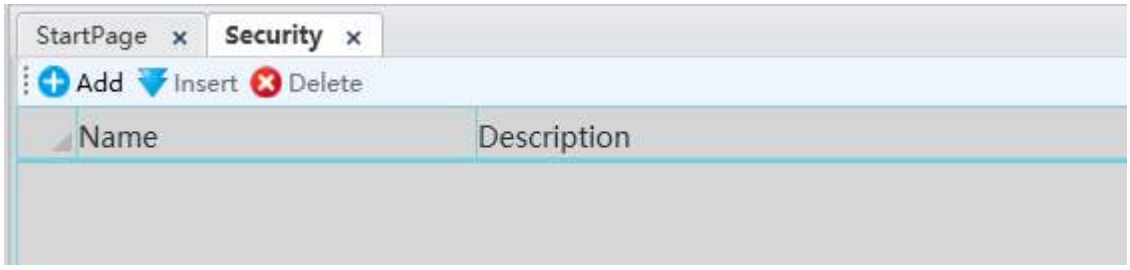
Security zone is a logical partition to the security control for accessing graphic objects in the project; it should be divided according to the security control when setting. The DIAView software has no preset security zone; under preset conditions, all users have the same access operating authority.

- **Setting steps**

Step 1: In the DIAView development environment, open the project window tree index → open the “Authority” node in the tree index:



Step 2: Double-click the “Security” node to open the security zone configuration window, as shown in the figure below**:**

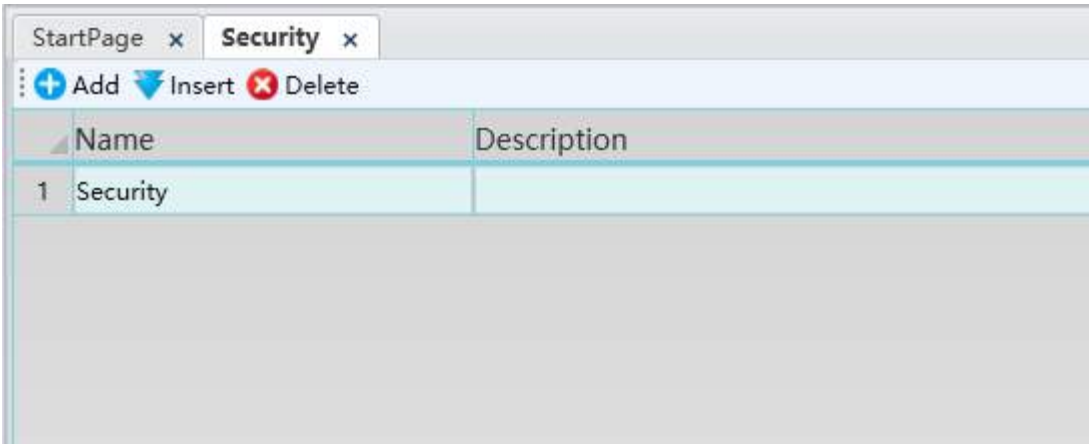


The meanings of each setting in the window are as follows:

Name: Name of the security zone.

Description: Related description information of the security zone.

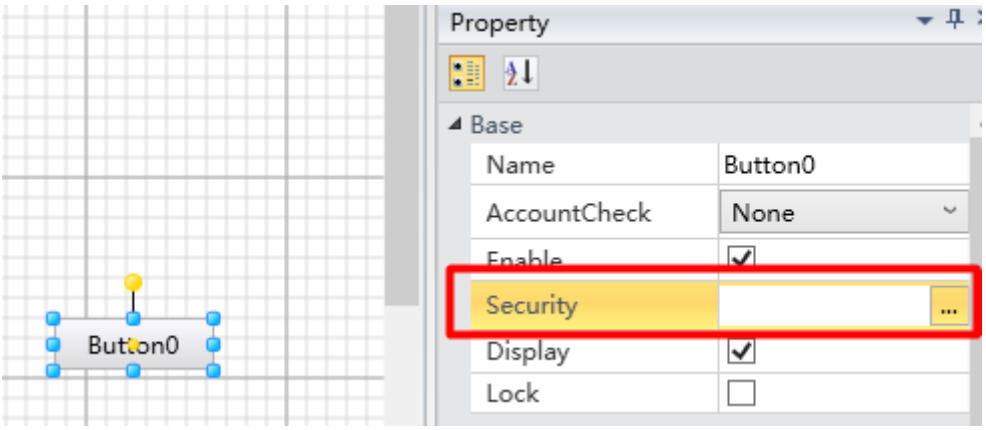
Step 3: Click the “Add” button to create the security zone and the system will automatically provide a default name, as shown in the figure below:



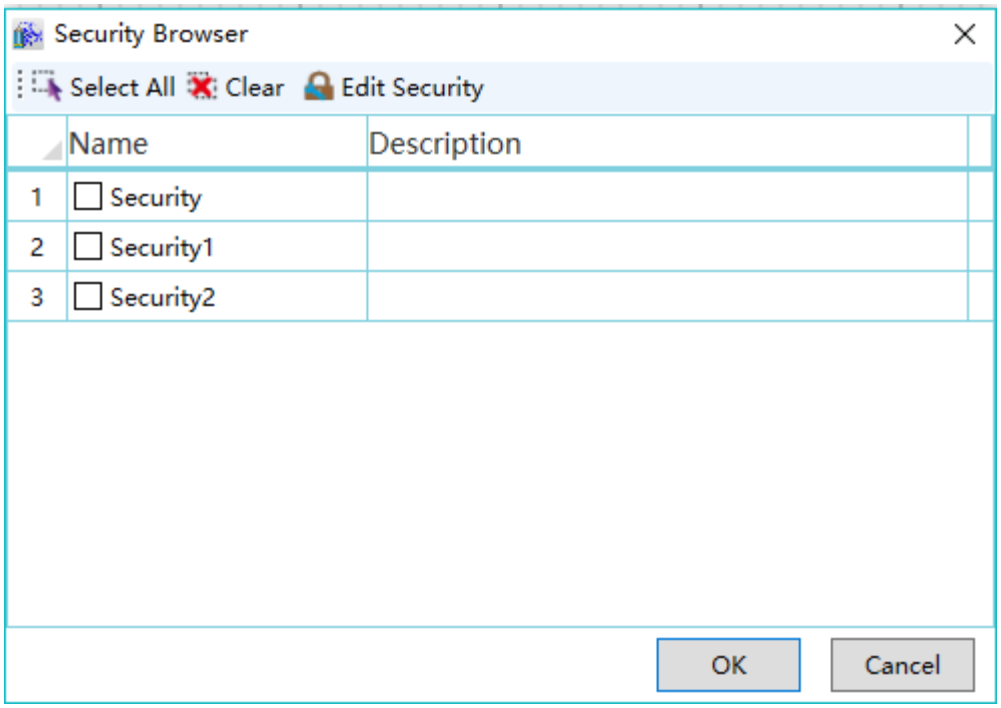
The name and description information of the security zone can be edited and the setting of the security zone will be completed,At the same time ,the “Delete” button can be used to delete the selected security zone.

- **Usage**

Graphic objects in the DIAView all have the “Security” property; select the graphic object to configure access operating authority in the sketchpad window, and then press the button to the right of the “Security” property in the property window, as shown in the figure below:



The “Security” selection box will appear; press the checkbox in front of a security zone to select it, as shown in the figure below:



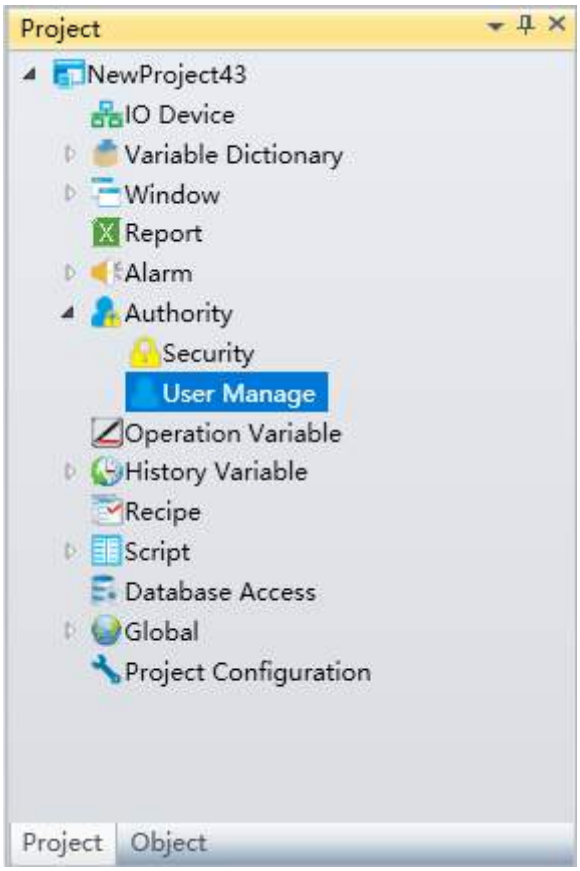
The “Maintenance Security” here can also be used to perform add or delete operations to the security zone; it will be synchronized with the operations performed in the previous “Security” window.

12.3 User

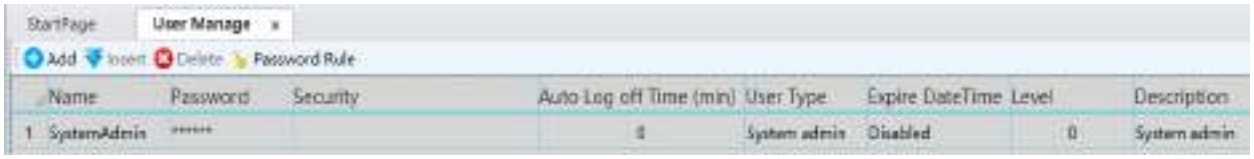
Users are the users of the DIAView who require related access and operating authority in the project; users are granted user rights by setting security zones so that users can access and operate the rights set for them, ensuring the security and reliability of the system.

➢ **Setting steps**

Step 1: In the DIAView development environment, open the project window tree index → open the “User authority ” node in the tree index:



Step 2: Double-click the “User” node to open the user configuration window, as shown in the figure below:



There will be a default system administrator in the user information——“SystemAdmin”, it cannot be added or deleted. The meanings of each setting in the window are as follows:

Name: User name

Password: User account password, used for user login

Security: Sets the security zone authority for user operations

Automatic Log off Time: Time limits can be set for user authority so that they automatically logout when the time has reached; ‘0” means always valid

User type: Sets the user type; there areThree types of users: SystemAdmin, Administrators and operators.

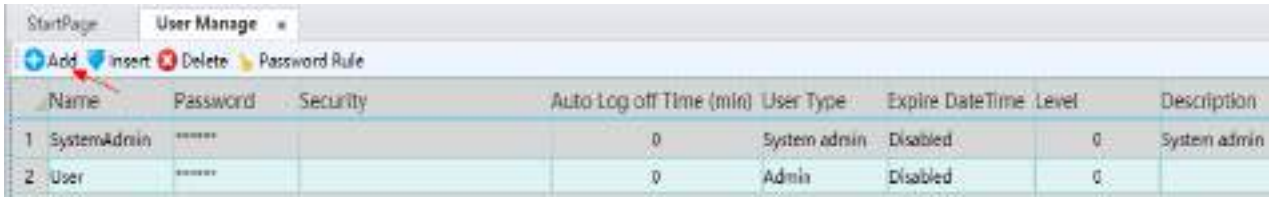
User type permission level: SystemAdmin>Administrators>operators.

Expire Date Time： There is a time limit for setting a user or password. If the time limit is reached, the user or password will automatically become invalid

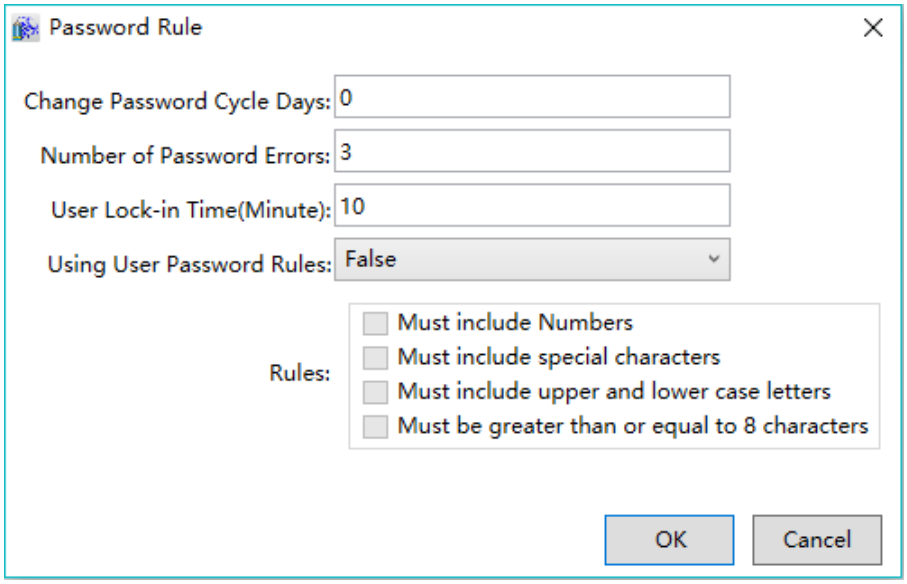
Level： Set the level difference of the same user type, the level range is 0~9, the higher the number is, the higher the level is

Description: User related information description

Step 3: Click the “Add” button to create the user and the system will automatically generate a default user name, as shown in the figure below:



Step 4: Click the "Password Rule"button,appear Password rules configuration dialog box, as shown in the figure below:



Change Password Cycle Days: The default extension of the password expiration time by days

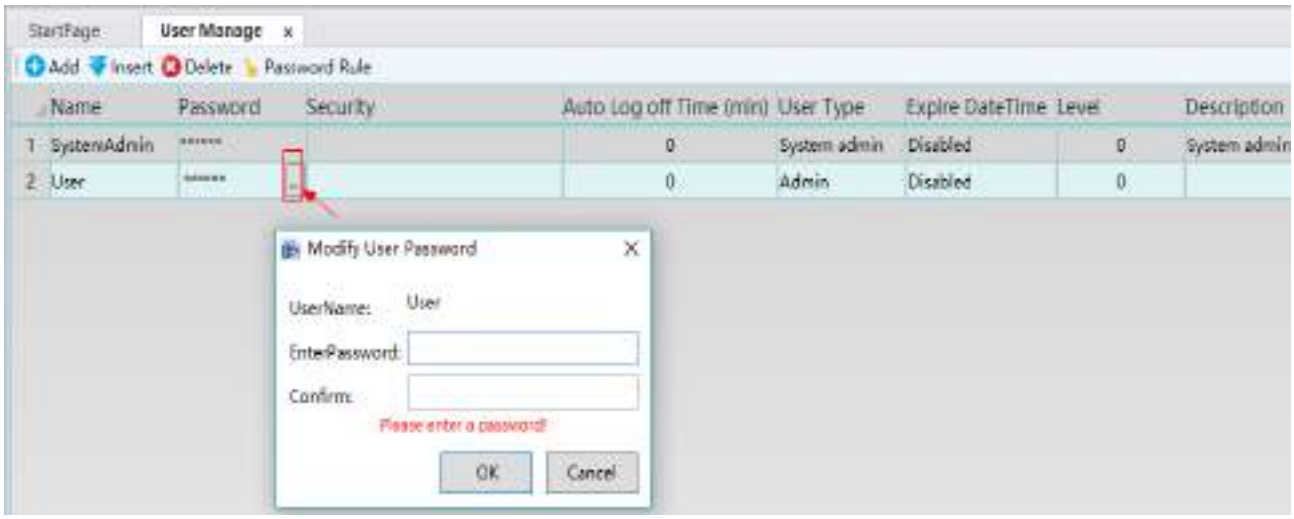
Number of Password Errors: When the number of password errors during login reaches the set value, the user will be locked automatically

User Lock-in Time(Minute): The length of time for the user to automatically unlock a password after it has been locked for a specified number of times

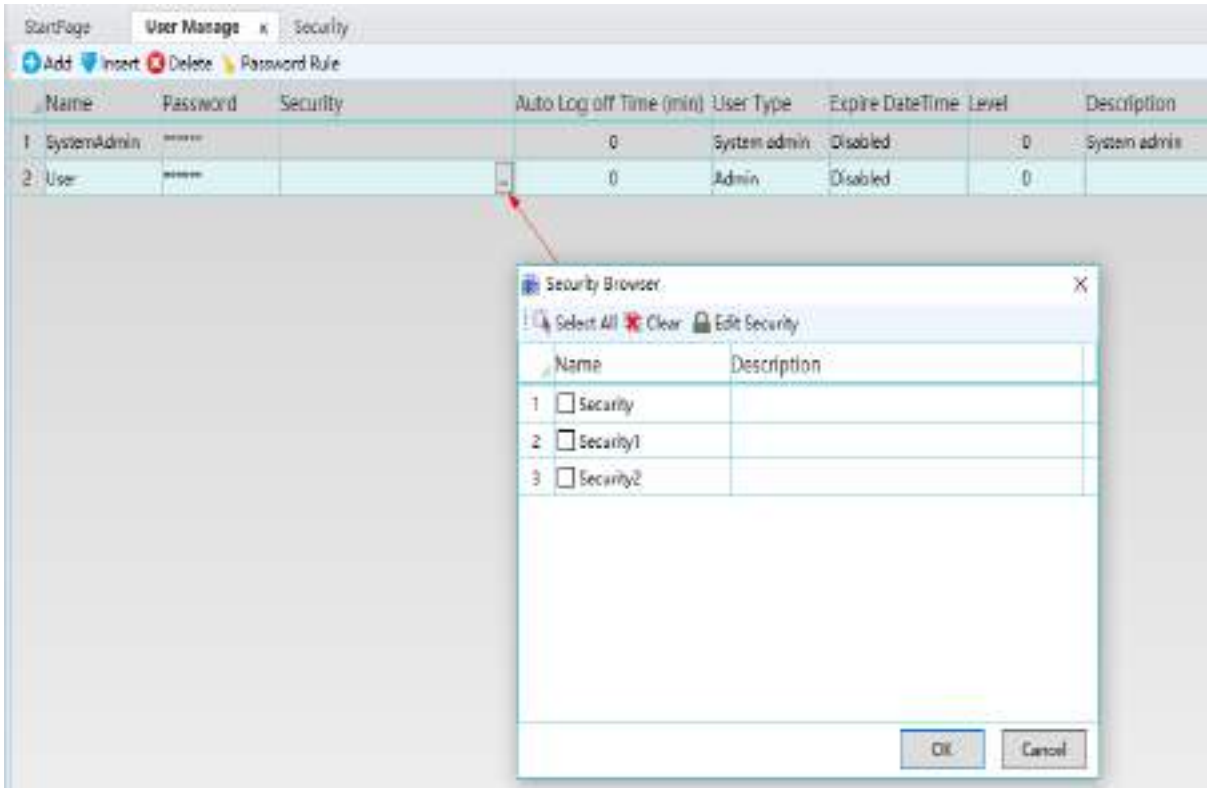
Using User Password Rules: If password strength is enabled, select True to enable password strength verification

Rules: Contains four rules, check one or all of them, then password characters can only be set in accordance with the rules, if the strength verification is turned on

Step 5: Click the button in the “Password” cell and the user password modification window will appear to allow changing of user password, as shown in the figure below:

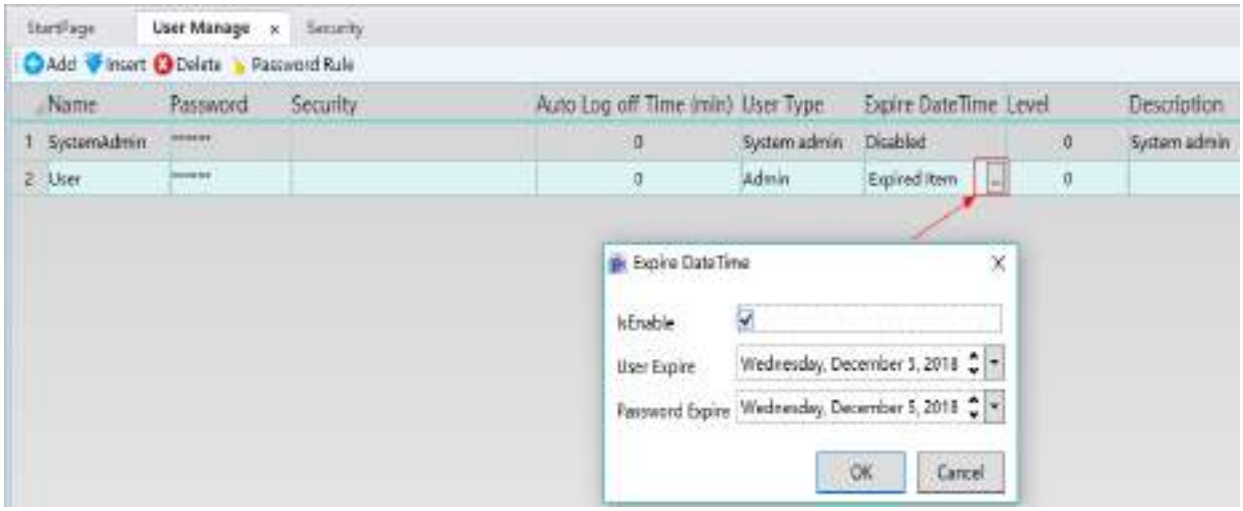


Step 6: Click the button in the “Security” cell and the security zone selection window will appear for setting the security zone rights of user operations, as shown in the figure below:



Step 7: Click the button in the “User Type” cell and select a user type from the drop-down menu; the system’s default user type is operator, as shown in the figure below:

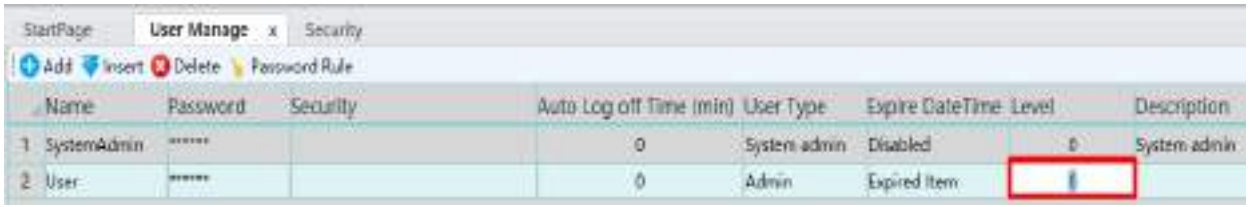
Step 8: Click the button in the "Expire DateTime" cell to pop up the expiration time configuration dialog box, select whether to enable this function, and set the expiration time of the user's expired password after enabled, as shown in the figure below:



UserAccount expired: When the system time reaches the set Expire DateTime of the user, the user is locked and unable to log in.

Password expired: When the system time reaches the set Expire DateTime of the password, the password is disable.

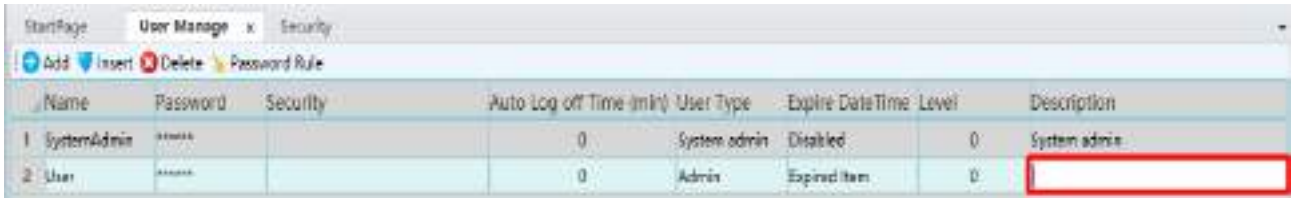
Step 9: Click the button in the "Level" cell to set the user level, as shown in the figure below:



User Level:

- The user level is used to distinguish the level differences of the same user type.
- Users can set the level range to 0-9. The higher the number, the higher the level, except SystemAdmin.
- High-level users have view and edit permissions for users below this level.
- System administrators are not subject to any restrictions.

Step 10: Click"Description" to add user description information, as shown in the figure below:

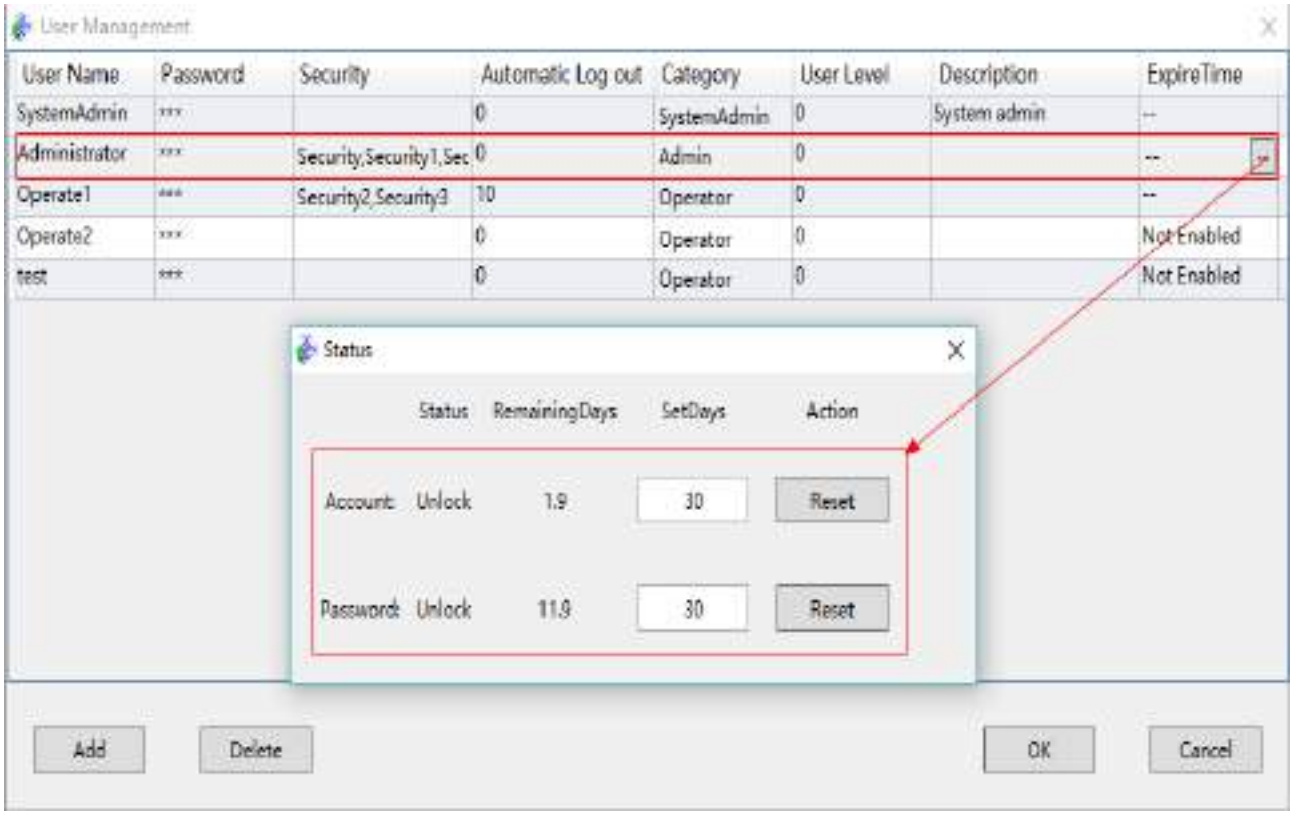


User description information can be added in the “Description” field; this will conclude user settings; the “Delete” button can be used to delete the selected user.

➢ **Runtime user and password setting methods**

- **Reset user expiration time**

The script *usercmd.usermanagerbox ()* is used to call out the user management window to set user password security zone and other parameters, as shown in the figure below:



When the expiration time of the user and password is reset, the expiration time is updated and the following principles are followed:

✧ Reset user expiration time: User expiration time = Current system time + SetDays

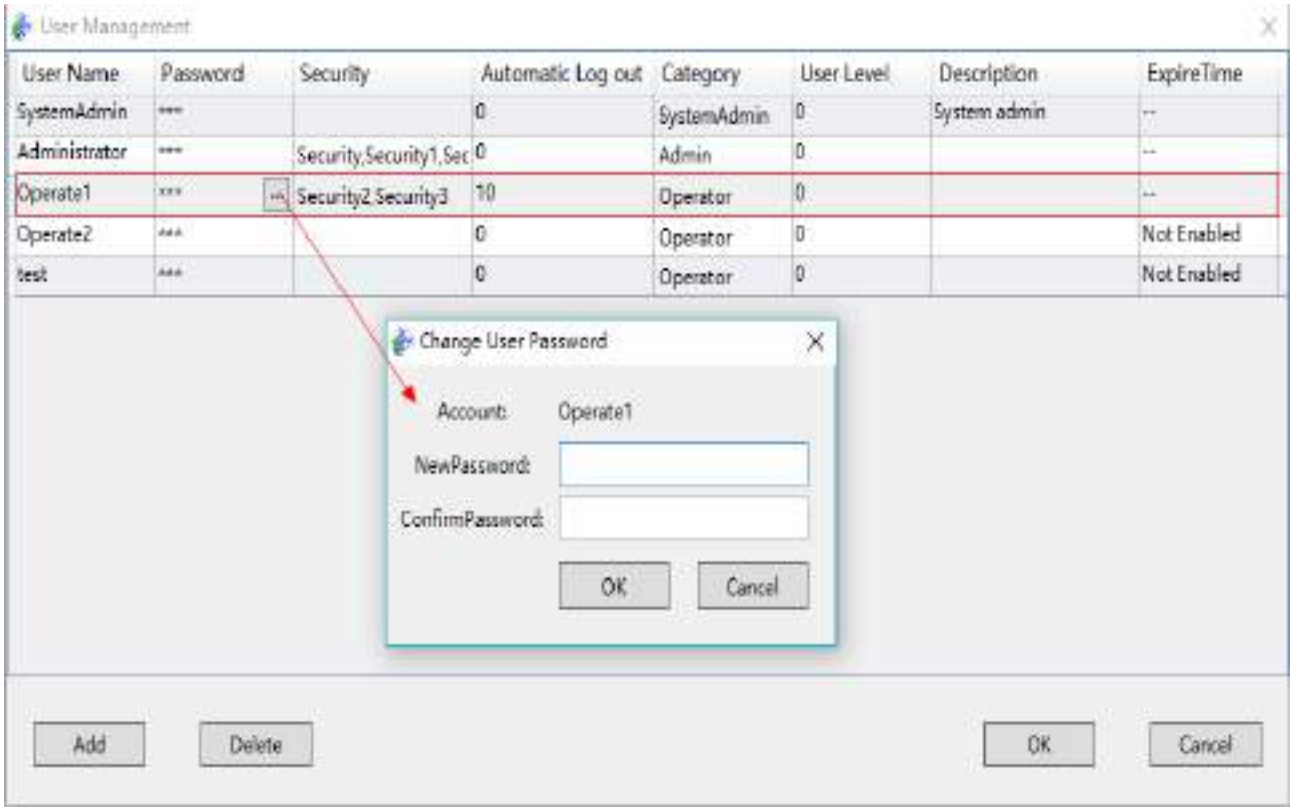
(Clear the RemainingDays to zero and start the timing again according to the set number of days)

✧ Reset password expiration time: Password expiration time = Current system time + SetDays

(Clear the RemainingDays to zero and start the timing again according to the set number of days)

- **Reset user Password**

The script *usercmd.usermanagerbox ()* is used to call out the user management window to set user password, as shown in the figure below:



When the remaining days of password expiration time > password valid days, reset the password, the expiration time will not be updated, on the contrary, user password reset, the password expiration time will be updated, and follow the following principles:

✧ Reset user password: Password expiration time = Current system time + Change Password Cycle Days

(Clear the RemainingDays to zero and start the timing again according to the Change Password Cycle Days)

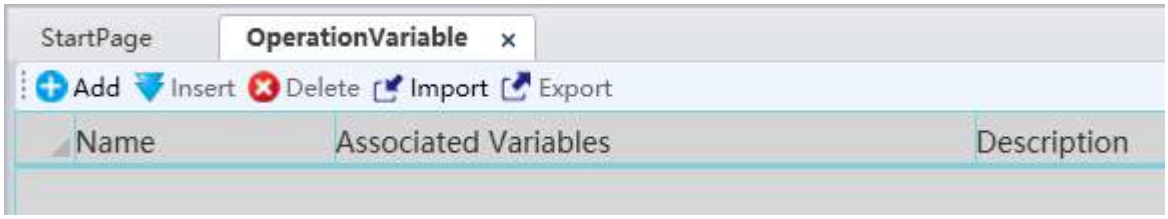
13. Operation Variable

13.1 Operation variable

Operation variables are variables used to save and record the project variables information which operated by the user.; it is related to events and used to record and track system variables which operated by the users. Record information includes: variable name, change time, value of variable before and after changes and user information etc.

➢ Configuring steps tfor operation variables are as follows:

Step 1: In the DIAView software development environment, open the project tree window → double click the “OperationVariable” in the project tree to open the operation variable configuration window, as shown in the figure below:



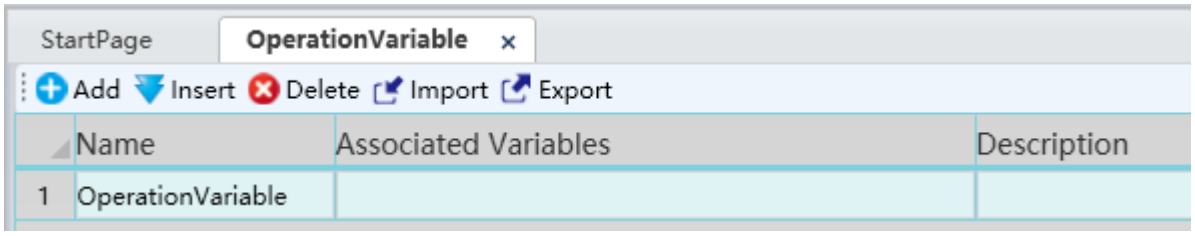
The meanings of each field in the window are as follows:

Name: Name of the operation variable

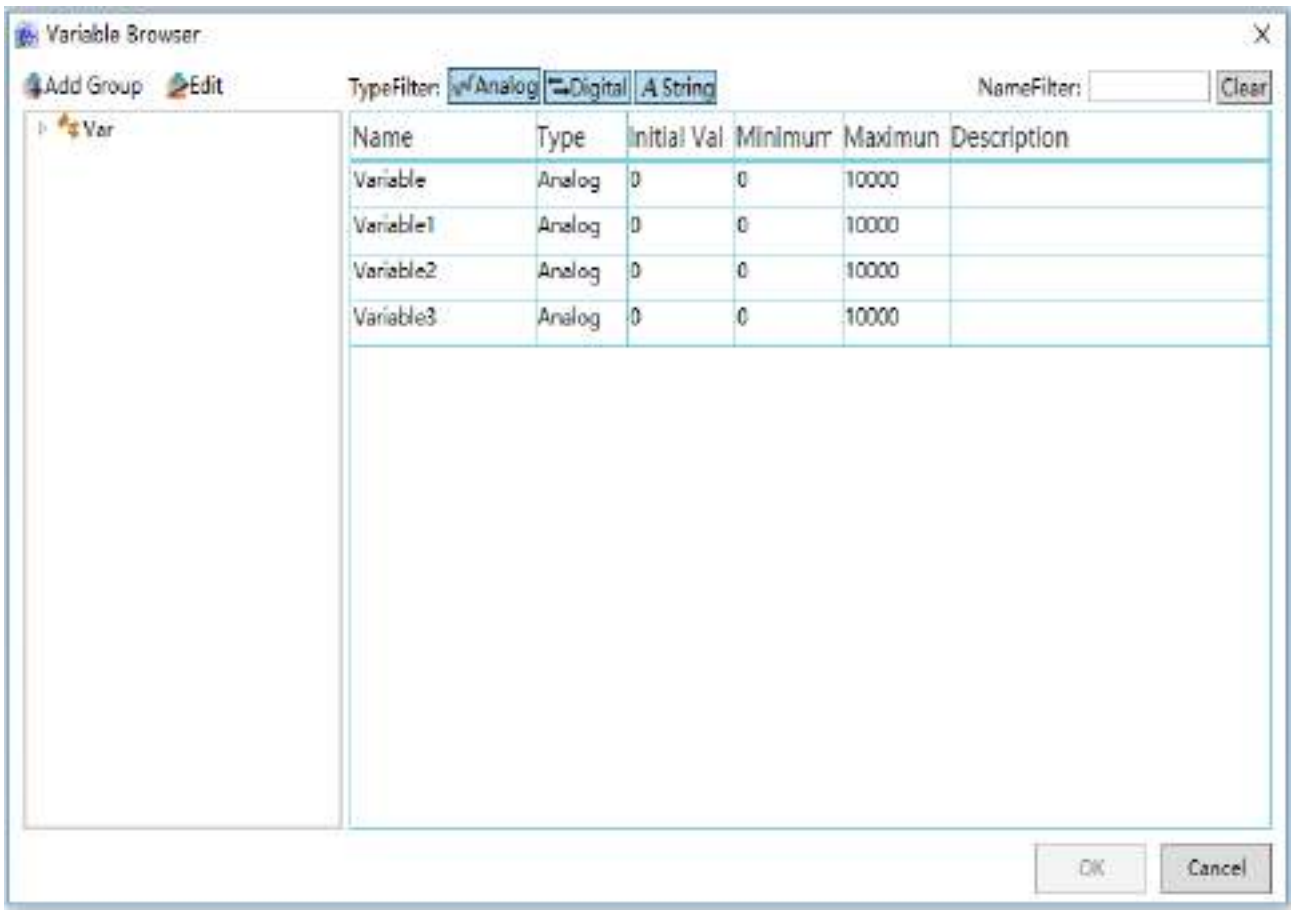
Associated Variables: The associated system variable

Description: Other information for the operation variable

Step 2: Click the “Add” button to create the operation variables and the system will automatically generate a default name, as shown in the figure below:



Step 3: Click the button in the “Associated Variables” cell and selected the variable to associate in the variable browser, as shown in the figure below:



- Each variable can only be associated to one operation variable.

After configuring the description information, the configuration of an operation variable is complete; the “Delete” button can be used to delete the selected operation variable and the “Import” and “Export” buttons can be used to import and export the operation variables from and into Excel.

Viewing operation variables: To view the information of the operation variables, user can call the report object scripts QueryVariableOperations(sheetIndex,opVariablePaths)

For example: Draw a report in the window: report 0 and a button: button 0;

Configure the event for “Button 0”: Left-click and call the operation variable query function of report 0:

```
Report0.QueryVariableOperations(0,"Operation")
```

14. History Variable

14.1 Overview

In order to display and mintor the field site running status of production, the DIAView software will create many variables that associates with field site equipment to perform data interaction, and these variables are constantly updating and changing. In order to save important data and analy the field data to mintor productive status and system running information. In order to increase productive efficiency and quality and optimize the system, the data of variables during system runnning need to be recorded; the variables that have this function are called history variables.

History variables are divided into 2 types: record variable and variables group record.

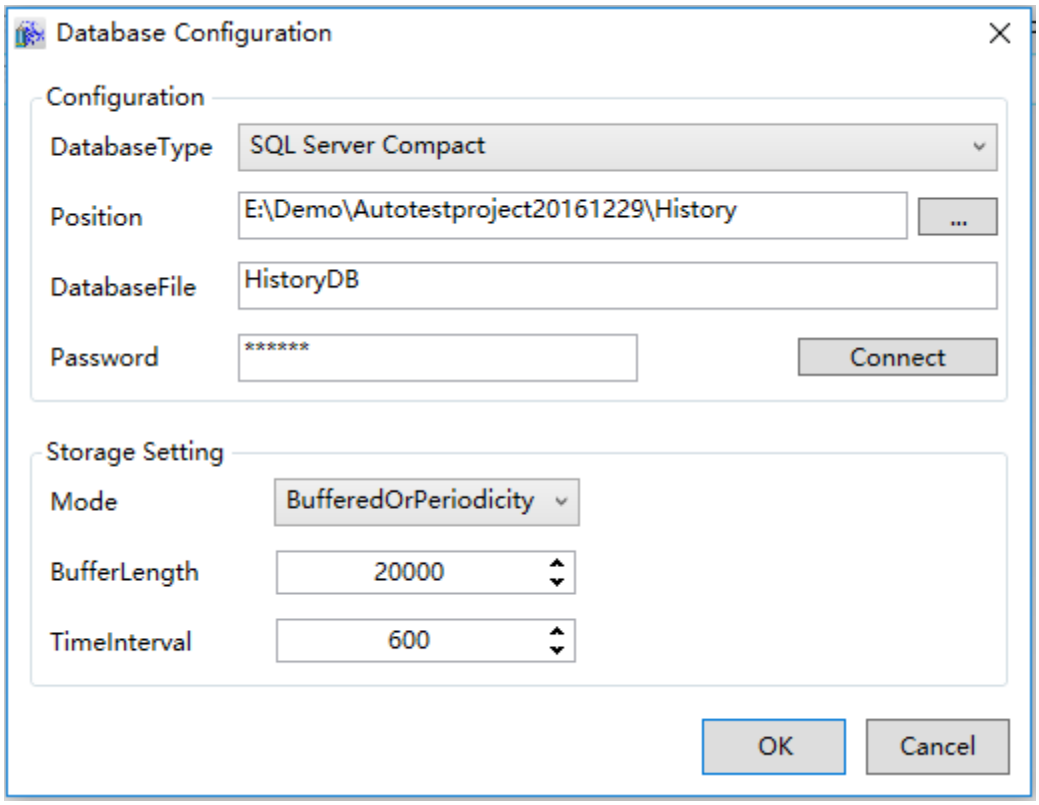
The record variables in the DIAView software are divided into two types recording methods: Timing record and changed record. Timing record records the data of the variables in the system according to the time interval cofigured; Changed record records the data of the variables in the system when the variable data changes.

Variable group record can record history group variable, each group can conclude 128 variables at most.Users can manage and query data conveniently by variable record group.

14.2 Setting history record variable(compatible)

Setting history record variable means configuring the project variables data which need to be saved and their recording method; compatible old version(under 2.5.0.0) of "RecordVariable", the configuring steps are as follows:

Step 1: Open the project tree window in the DIAView software development environment → Right_click the “RecordVariable(Compatible)” in the project tree index → Click "Database Configuration" to open database configuration window, as shown in the figure below:

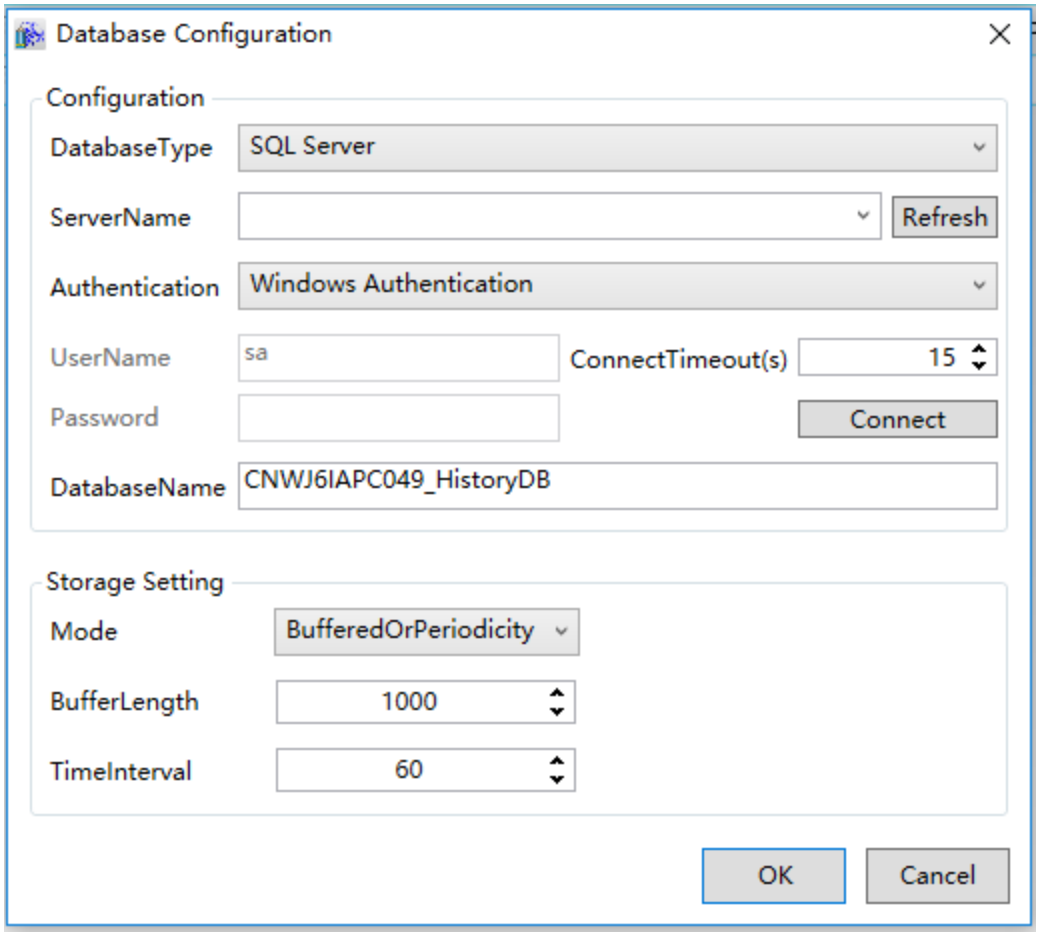


Note: Database default password is“666666”.

Step 2: Set related properties in the“Configuration”column:

DatabaseType: Select the data source type to use;currently the DIAView supports three types: SQL Server, SQL Server Compact and Oracle.

✧･ When “SQL Server” is selected, the database configuration is as shown in the figure below:



ServerName: Enter the server name or IP address of the database which needs to be connected to; click the pull-down button to search automatically for servers within the network.

DatabaseName: Enter the database name to use (length: 2 thousand words).

Authentication: There are two methods to authorize when logging in the server:Windows ID authentication and SQL ID authentication; if the database is configured on a local host, select “Windows ID authentication”; if the database is configured on a remote computer, select “SQL ID authentication”.

UserName: Sets the user name for SQL ID authentication.

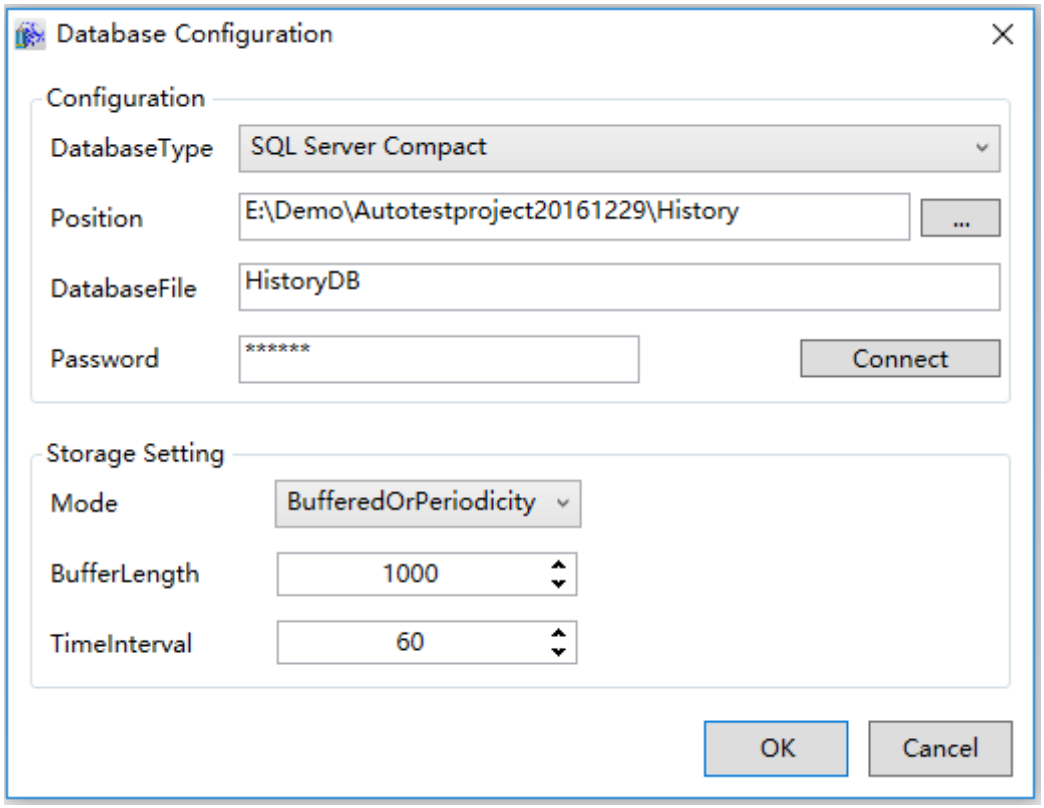
Password: Sets the user password for SQL ID authentication.

ConnectionTimeout: Sets the time for the database to switch connection or reconnect after connection failed, and report an error if it is the timeout; unit: seconds.

Connect: Tests whether it can connect to the database server.

✧･ When “SQL Server Compact” is selected, the database configuration is as shown in the

figure below: The second field is the “Database file”, enter the database file path and the file name, as shown in the figure below:

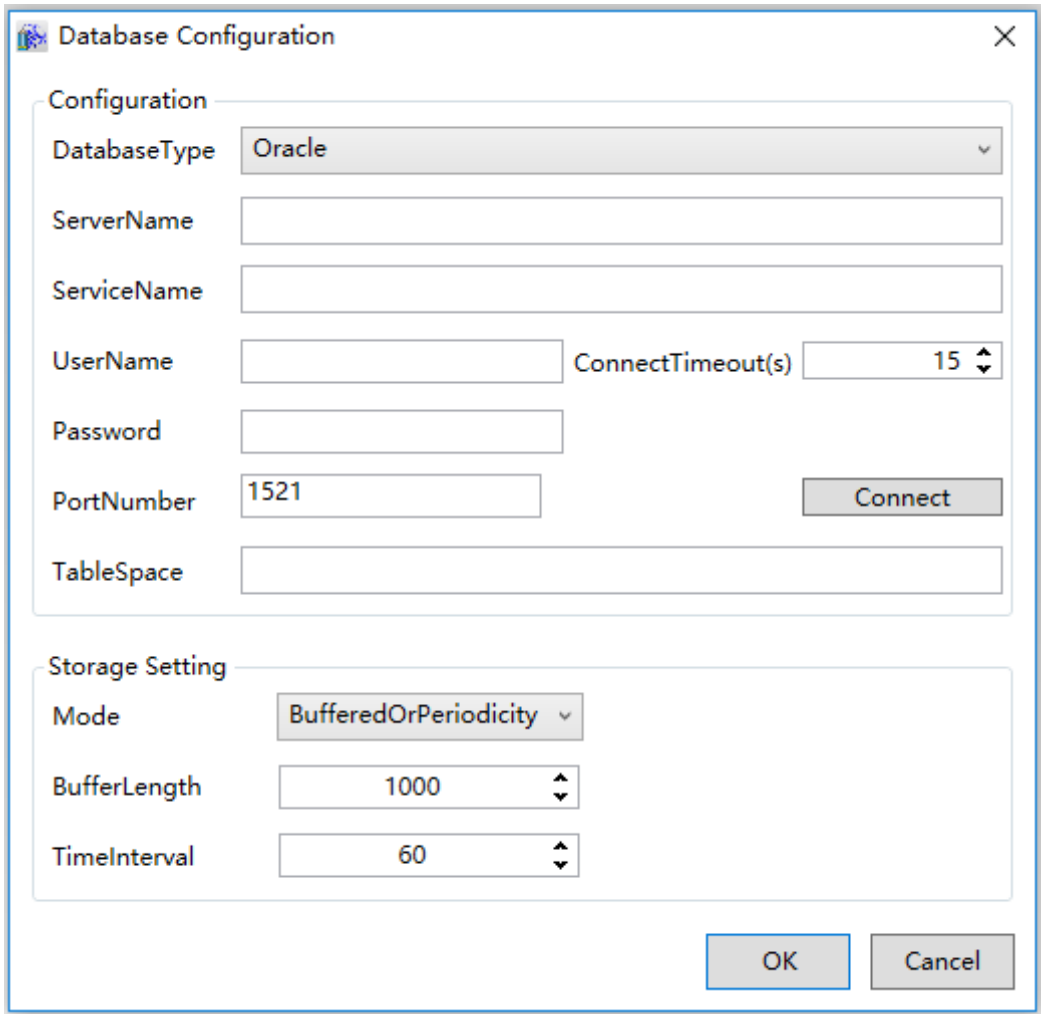


DatabaseFile: Enter the database file path and file; the button on the right can be pressed to select the directory.

Password: Database password; left empty if the password is not required.

Connect: Tests whether it can connect to the database.

✧･ When “Oracle” is selected, the database configuration is as shown in the figure below:



ServerName: The hostname or IP address.

ServiceName: The name of Oracle service.

TableSpace: Oracle table space，it can be empty (the default table space).

PortNumber: Commonly it is OK to select the default port number .

ConnectTimeout: The database will switch or reconnect in some time when the setting database connection fails and if it exceeds the time range, an error will be reported ; unit: seconds.

Step 3: Set related properties in the “Save settings” field:

Mode: Set the save mode of the database; there are four modes: Buffered, Periodicity, BufferedOrPeriodicity and SingleInsert.

BufferLength: Sets the record number of buffer , unit: entries; this property can only be used when the save mode is Buffered and BufferedOrPeriodicity.

TimeInterval: Sets the time interval of periodical storage, unit: seconds; this property can only be used when the saved mode is Periodicity and BufferedOrPeriodicity.

Once setting is complete, press the “OK” button to save the project configuration information.

Step 4: Open the project tree window in the DIAView software development environment → double click the “RecordVariable(Compatible)” in the project tree index to open the record variable configuration window, as shown in the figure below:

The meanings of each configuration in the window are as follows:

Name: Name of the record variable.

Associated Variables: Associated project variable, which are the project variables need to be saved.

Trigger Mode: Set the recording method of data; there are 3 types: Changed record , timing record,no record.

For the no record history variables, users can determine when to record through scipt command.

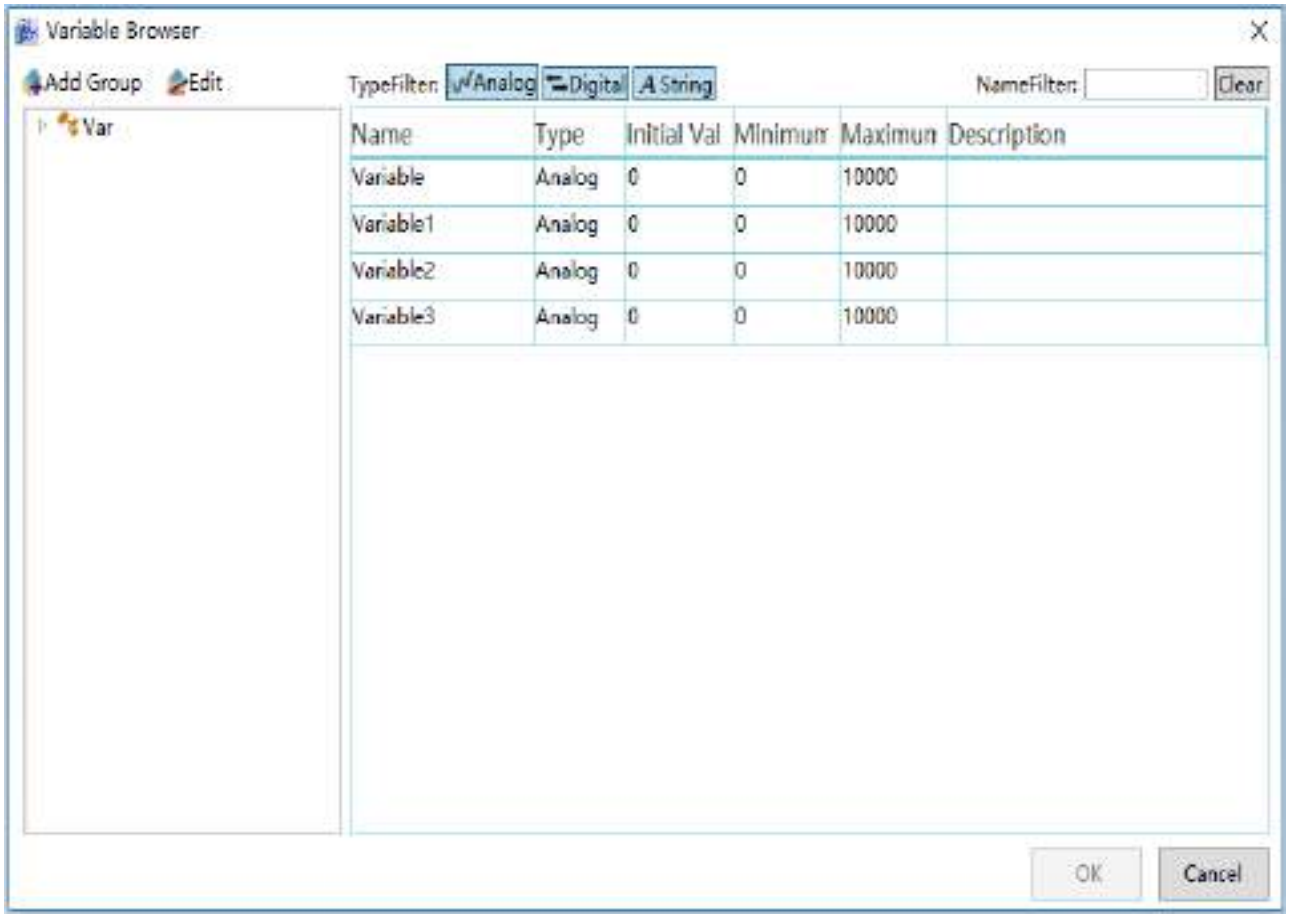
Time Interval: Set the time interval of the recording method is “Timing record”.

Deadband: When the associated project variable is a analog value, set a value to form a deadband with the value of the variable to perform data filtering.

Description: Other information for the record variable.

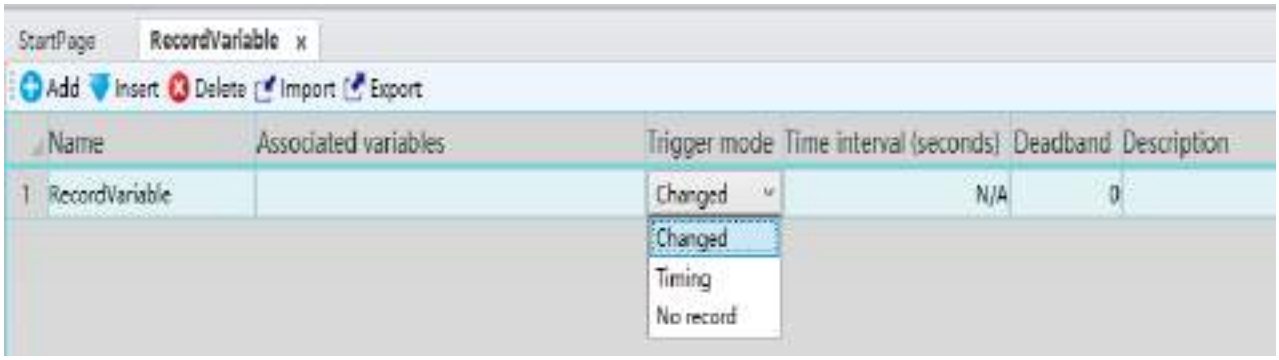
Step 5: Click the “Add” button to create the history record variables and the system will automatically generate a default name, as shown in the figure below:

Step 6: Click the button in the “Associated Variables” cell and selected the variable need to record in the variable browser, as shown in the figure below:



- Each variable can only be associated to one history record variable.

Step 7: Click the drop-down button in the “Trigger mode” cell and select the recording method, as shown in the figure below:



✧ **Timing record:** Sets the time interval, unit: seconds;during system running, it will record the data of the variables according to time interval.

✧ **Changed record:** Records the variable data when the variable data changes during system running.

✧ **No record:** For the no record history variables, users can determine when to record through scipt command.

Step 8: Time interval: If the record method is “Timing”, the time interval needs to be set;

Step 9: Deadband: If the associated variable is an analog value type, and the record method is “Changed record”, set a value as the deadband value to form a deadband with the value of the variable: (current value of variable - deadband value) <= deadband <= (current value of variable ＋ deadband value);

Effect: Data filtering.

Principle: When the next variable value is within the deadband range, no new values will be record and the original value will remain the same. If it is not in the deadband range, then the record variable value will acquire a new variable value and record it into the database, and the deadband will also change accordingly. It will keep looping in this way.

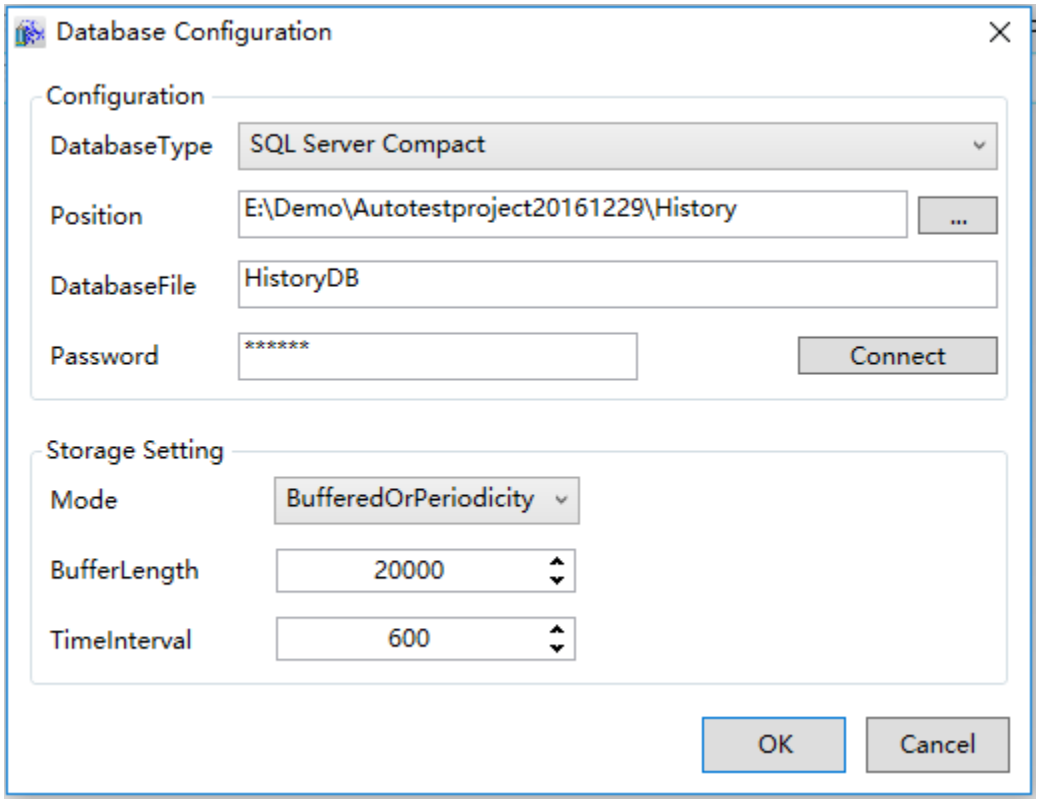
The “Delete” button can be used to delete the history variables selected and the “Import” and “Export” buttons can be used to import and export the history record variables from and into Excel.

14.3 Setting history record variable

The version of DIAView2.5 add new "RecordVariable",in project tree, new project has

"RecordVariable" node only, old project has "RecordVariable" node and "RecordVariable(compatible)" to compatible old "RecordVariable".the configuring steps are as follows:

Step 1: Open the project tree window in the DIAView software development environment → Right_click the “RecordVariable” in the project tree index → Click "Database Configuration" to open database configuration window, as shown in the figure below:



The meanings of each configuration in the window are as follows:

ServerName: Enter the server name or IP address of the database which needs to be connected to; click the pull-down button to search automatically for servers within the network.

DatabaseName: Enter the database name to use (length: 2 thousand words).

Authentication: There are two methods to authorize when logging in the server:Windows ID authentication and SQL ID authentication; if the database is configured on a local host, select “Windows ID authentication”; if the database is configured on a remote computer, select “SQL ID authentication”.

UserName: Sets the user name for SQL ID authentication.

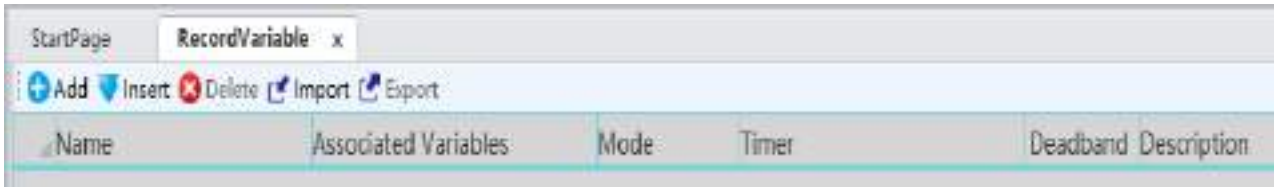
Password: Sets the user password for SQL ID authentication.

ConnectTimeout: Sets the time for the database to switch connection or reconnect after connection failed, and report an error if it is the timeout; unit: seconds.

ConnectionTest: Tests whether it can connect to the database server.

DaysOfDataRetention: The times of date saved, default: 30 days.

Step 2: Open the project tree window in the DIAView software development environment → double click the “RecordVariable” in the project tree index to open the history record variable configuration window, as shown in the figure below:



The meanings of each configuration in the window are as follows:

Name: Name of the record variable.

Associated Variable: Associated project variable, which are the project variables need to be saved.

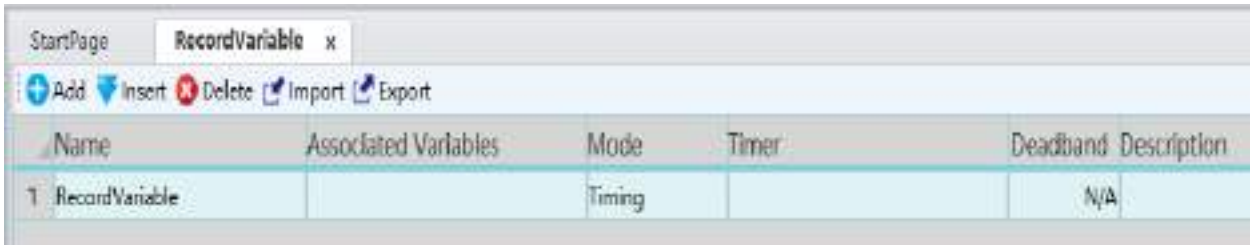
Mode: Set the recording method of data; there are 2 types: Changed record and timing record.

Timer: Set the time interval of the recording method is “Timing record”.

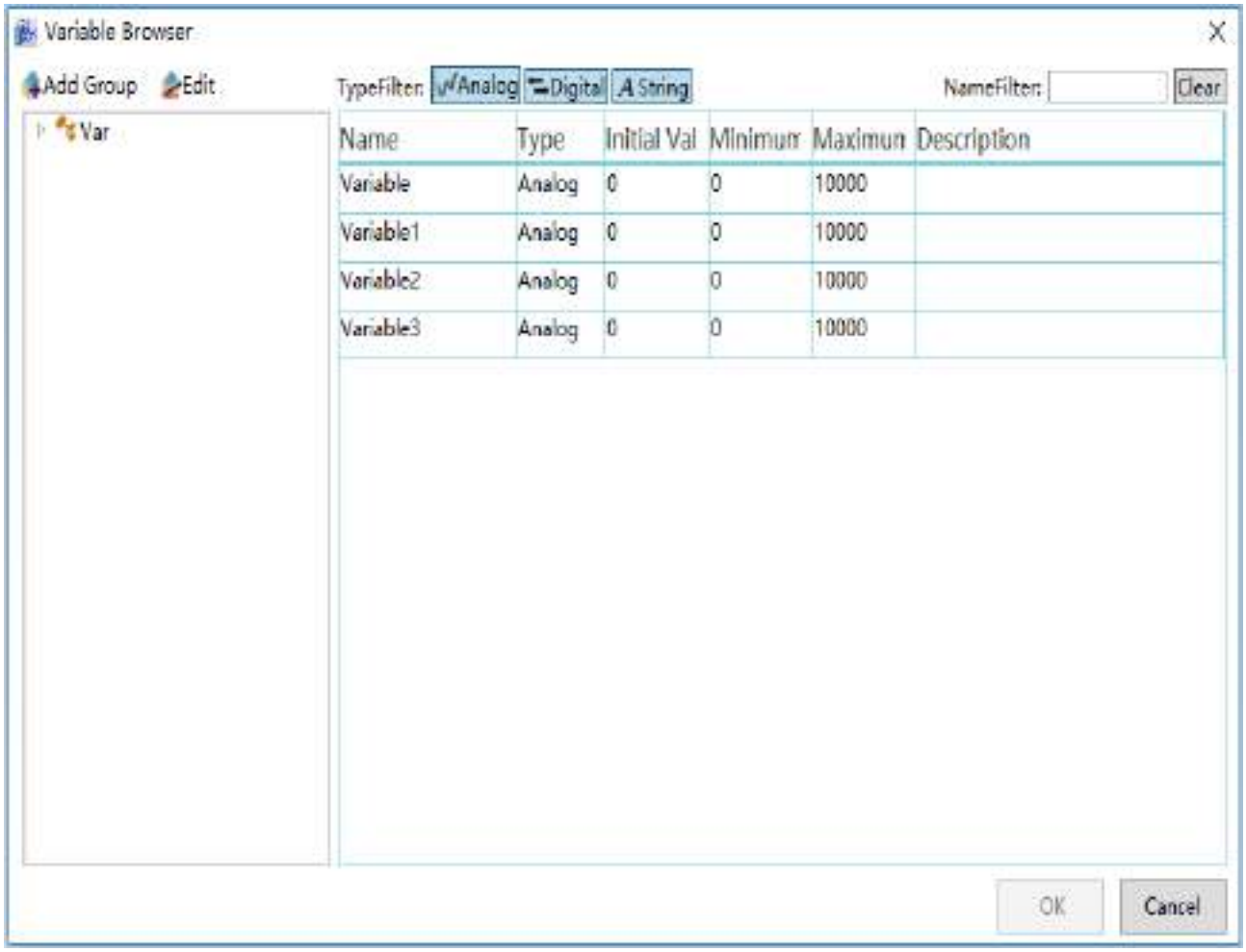
Deadband: When the associated project variable is a analog value, set a value to form a deadband with the value of the variable to perform data filtering.

Description: Other information for the record variable.

Step 3: Click the “Add” button to create the history record variables and the system will automatically generate a default name, as shown in the figure below:

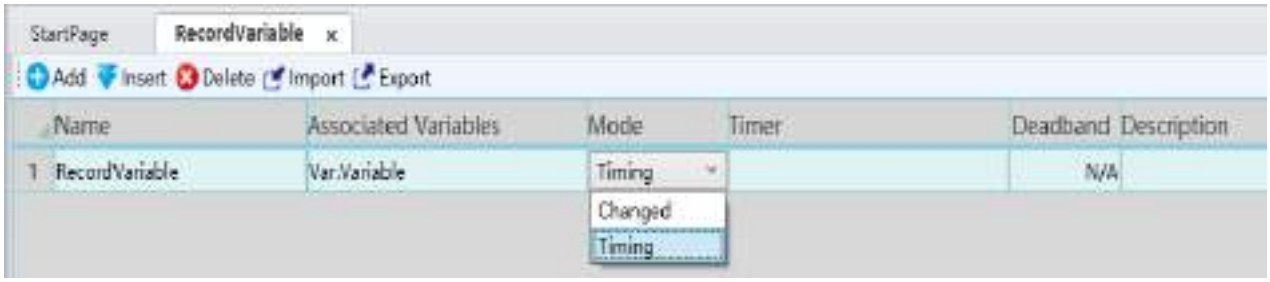


Step 4: Click the button in the “Associated Variables” cell and selected the variable need to record in the variable browser, as shown in the figure below:



- Each variable can only be associated to one history record variable.

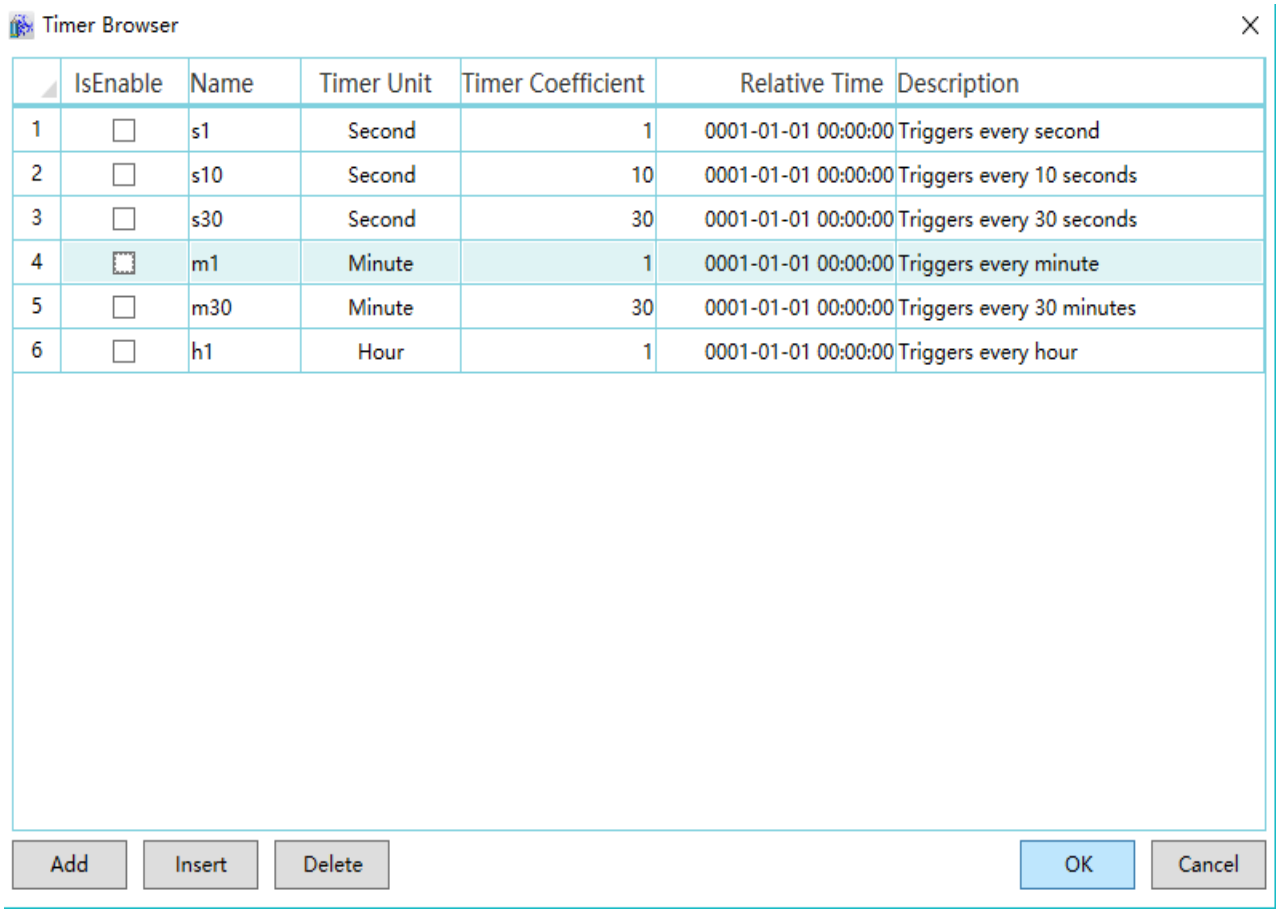
Step 5: Click the drop-down button in the “Mode” cell and select the recording method, as shown in the figure below:



Timing: Set timer;During the operation of the system, the data of variables in the system will be recorded according to the set timer.

Changed: The system is recorded whenever variable data changes during operation.

Step 6: Timer:If the recording mode is "Timing", the timer needs to be set.Click the button  in the timer cell and the timer browser pops up,as shown in the figure below:



The meanings of timer browser are as follows:

IsEnable: Check the checkbox to enable this timer,allow to use multiple timer, and enable timers

decide the query intervar time(if user check "s1" and "m1"，the interval time is 1 second or 1 minute).

Name: Timer name, use English alphabets, numbers, or underscores, beginning with an English letter, with a word length of no more than 12.

Timer Unit: the base unit of timer, now has "Second", "minute", "hour" and "day".

Timer Coefficient: Trigger time is timer unit * timer coeffcicent(ex: timer unit is second, timer coefficient is 1, the trigger tiem is 1 second).

Relative Time: Click the button in the “Relative Time” cell , has two select: "Project start time" and "User_defined time", it determine the real time of timer triggered(ex: user set a timer, trigger time is 1 day, if select "Project start time", assume project start at "2018-1-1 12:12:12", then the real trigger time of timer is 12:12:12 every day; if user select "User_defined time", assume the defined time is "2018-1-1 00:00:00", then the real trigger time of timer is 00:00:00 every day).

Description: Description to this timer.

Add: Add a timer in the end of datagrid.

Insert: Insert a timer.

Delete: Delete selected timer.

OK: Save modification.

Cancel: Cancel modification.

Step 7: Deadband: If the associated variable is an analog value type, and the record method is “Changed record”, set a value as the deadband value to form a deadband with the value of the variable: (current value of variable - deadband value) <= deadband <= (current value of variable ＋ deadband value);

Effect: Data filtering.

Principle: When the next variable value is within the deadband range, no new values will be record and the original value will remain the same. If it is not in the deadband range, then the record variable value will acquire a new variable value and record it into the database, and the deadband will also change accordingly. It will keep looping in this way.

The “Delete” button can be used to delete the history variables selected and the “Import” and “Export” buttons can be used to import and export the history record variables from and into Excel.

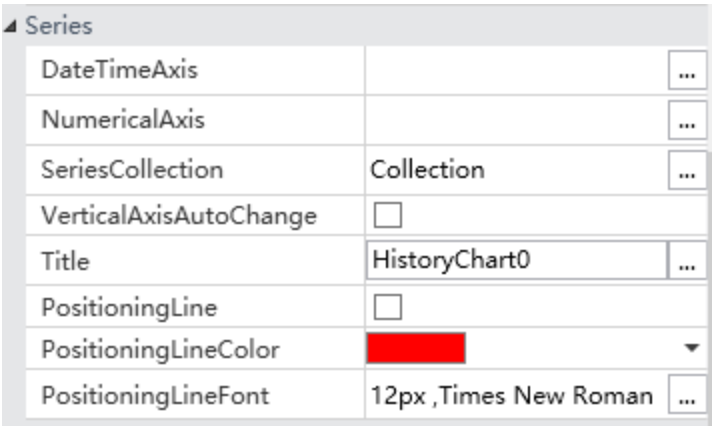
14.4 Viewing history record variable

Once the history record variables data were saved, it requires related display tools to view the history record variables data and analy and operate the data. The DIAView software provides history charts and reports to query and analy the history record variables.

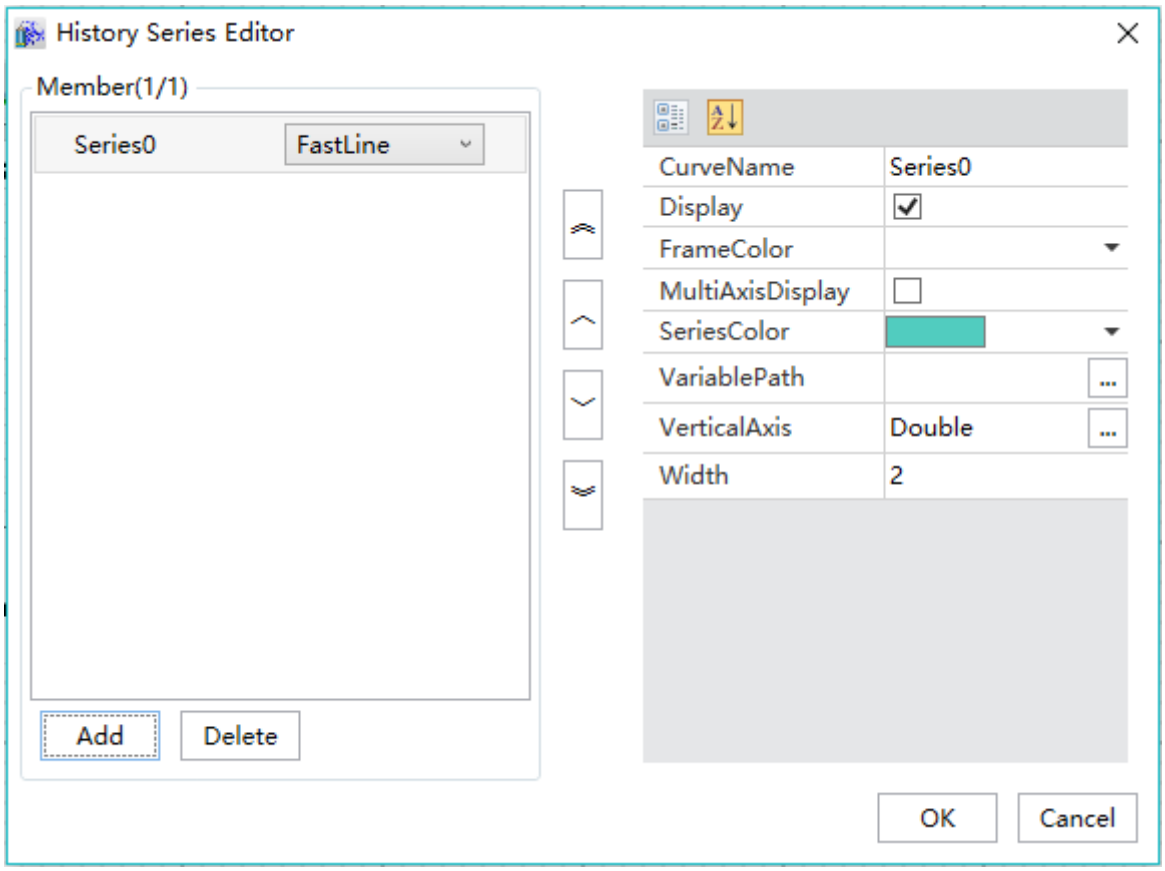
- **History chart**

The history charts can draw the history record variable data within a period into a curve and display it ; it is a graphic object. One history record variable corresponds to one curve; multiple history record variable curves can be configured and can be used for data comparison and analysis. For drawing of history charts, please refer to the chapter “7.7.2 History chart”; steps to associate history record variables are as follows:

Step 1: After the history chart is drawn, open its property window and click “Series” → click the button in the “SeriesCollection” bar, as shown in the figure below:

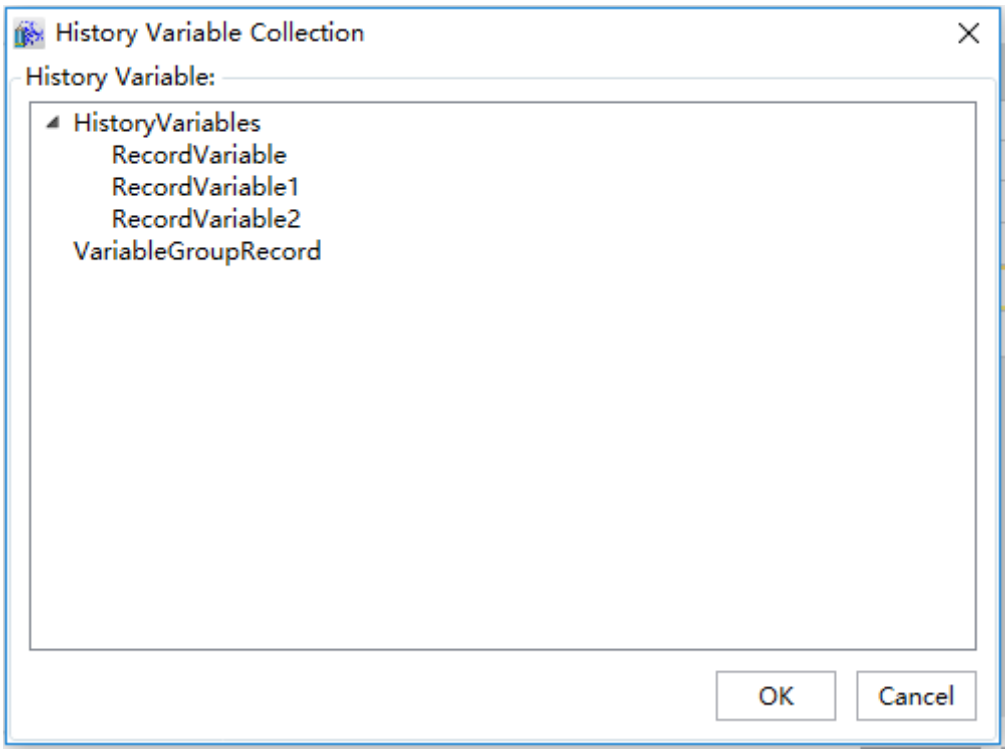


Step 2: The “History series editor”window will appear, and then click the “Add” button to add curves, as shown in the figure below:



The system will generate a curve with a default name; the left side of the window is the list of curve members and the right side is the corresponding curve property window.

Step 3: Click the button in the right of “VariablePath” bar in property window , and the history variable collection window will appear, it will allow you to select a history record variable to associate with the curve, as shown in the figure below:

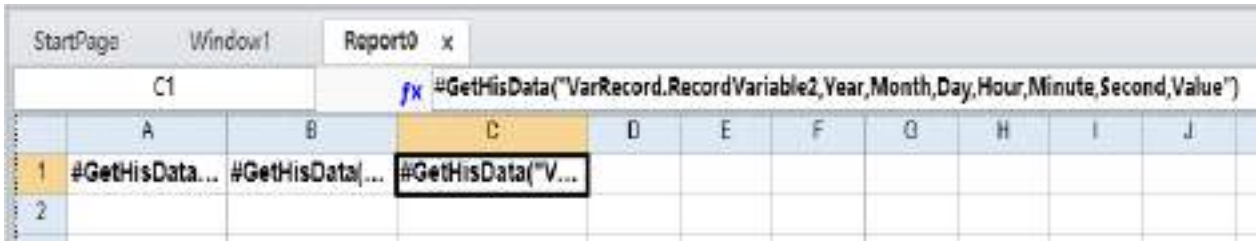


Press the “OK” button after selecting a history record variable to return to the “History Series Editor” window; press the “OK” button after setting the color of the curve and other properties to complete the configuration for a history curve. The steps mentioned above can be repeated to configure multiple history curves.

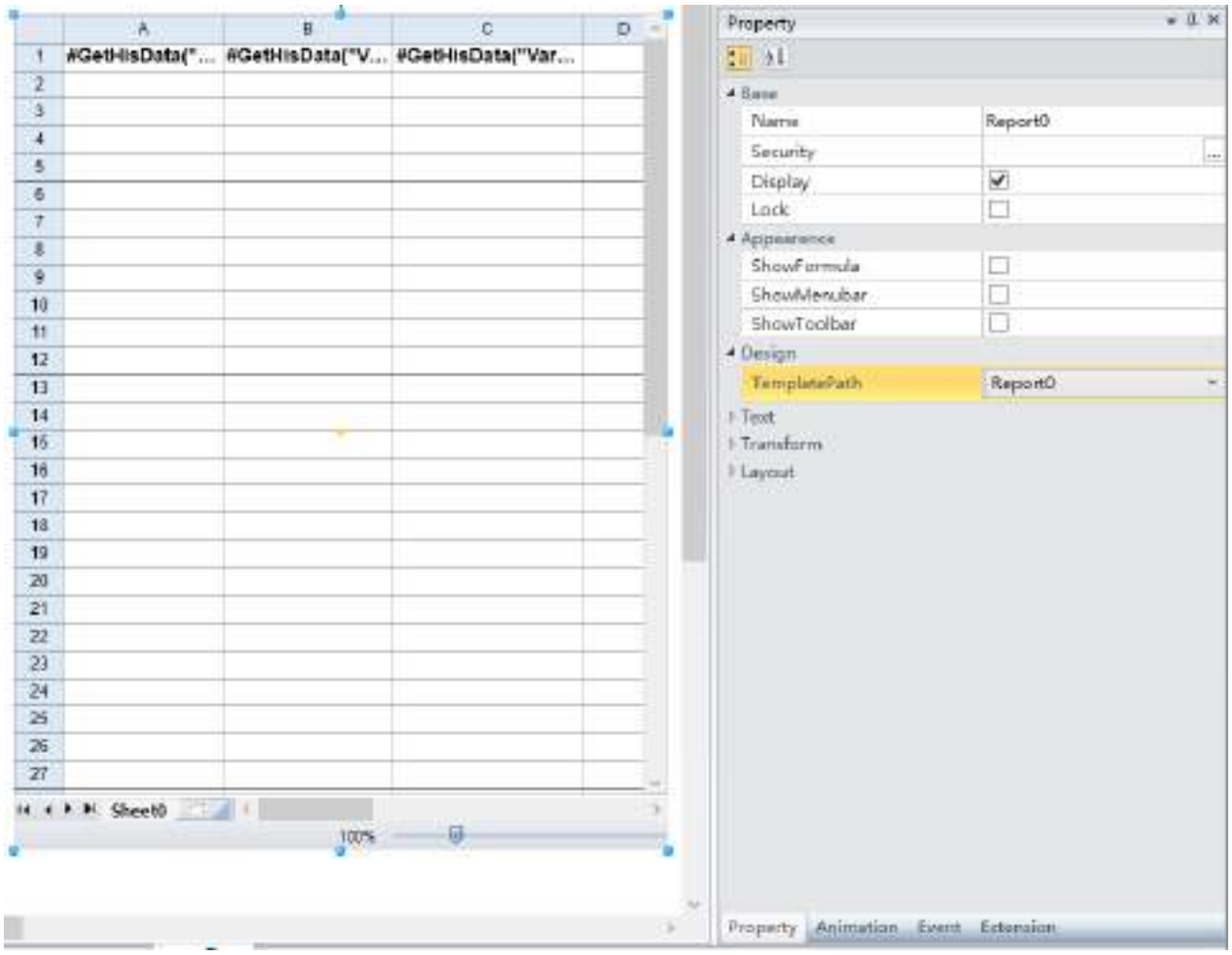
- **Report**

The historical data of history record variable can be presented in the form of report.A report can present the historical data of multiple history record variables in multiple columns. For drawing of history charts, please refer to the chapter “7.7.11 Report”; steps to associate history record variables are as follows:

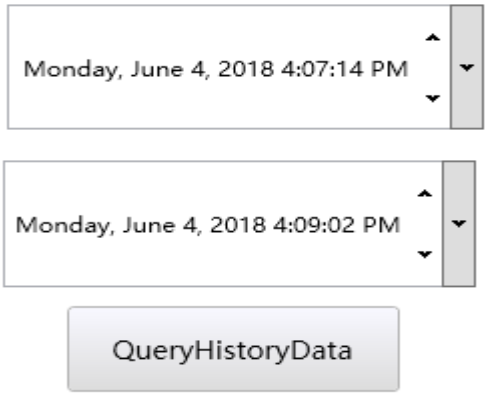
Step 1: Create new report template,associate with history record variable.



Step 2: Draw a report in window,associate with the created report template in step1.



Step 3: Create two DateTimePicker in window(DateTimePicker0,DateTimePicker1) and one button, configure the button script as follows:



DateTimePicker0.Value=Sys.StartTime

DateTimePicker1.Value=Sys.Now

Call Report0.SetWorkSheetStartTime(0,DateTimePicker0.ValueTime)

Call Report0.SetWorkSheetEndTime(0,DateTimePicker1.ValueTime)

Call Report0.SetWorkSheetIntervalTime(0,1000)

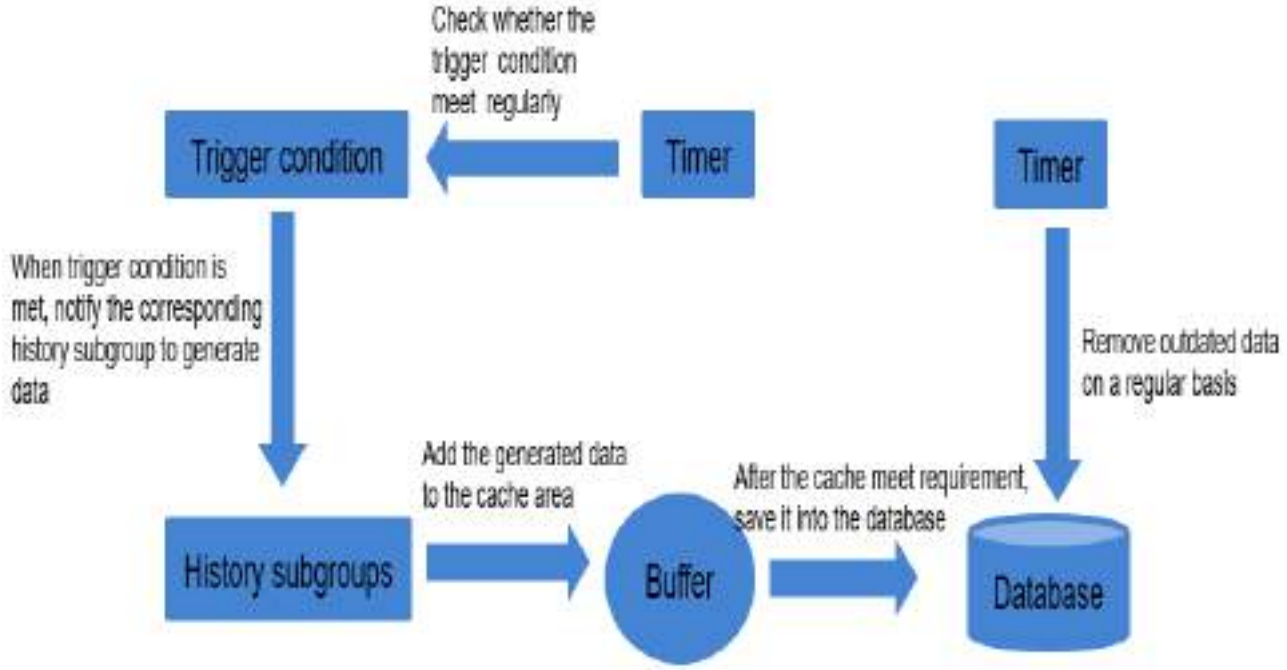
Call Report0.QueryHistoryData()

Step 4: Run the project,click "QueryHistoryData" button,can view historical data in the report,as shown in the figure below:

	A	B	C	D
1	2017/07/13 10:48:02	--	--	--
2	2017/07/13 10:48:03	--	--	--
3	2017/07/13 10:48:04	81	29	91
4	2017/07/13 10:48:05	69	21	30
5	2017/07/13 10:48:06	70	7	40
6	2017/07/13 10:48:07	45	68	86
7	2017/07/13 10:48:08	82	40	83
8	2017/07/13 10:48:09	61	78	66
9	2017/07/13 10:48:10	34	66	92
10	2017/07/13 10:48:11	2	37	56

14.5 Setting variable group record

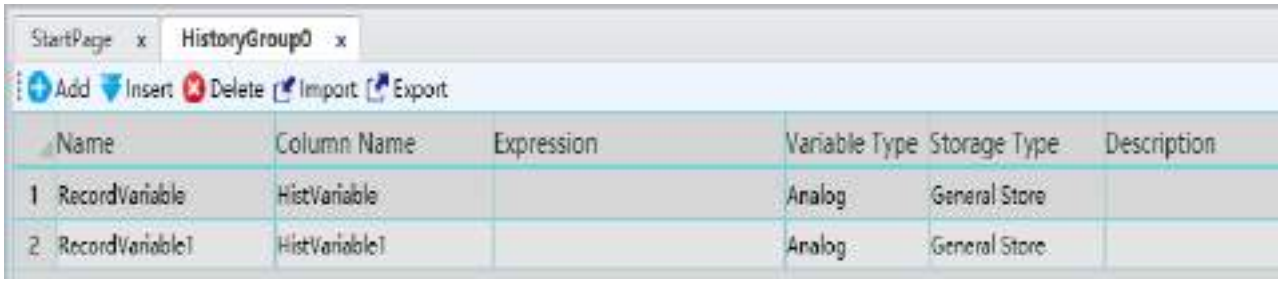
Variable group record operational principles are as follows:



When triggering condition is triggered,history record variable data will be saved to the configured database.

Variable group record needs to configure trigger condition、database and history record variable.The configuration steps are as follows:

Step1: Configure variable record:

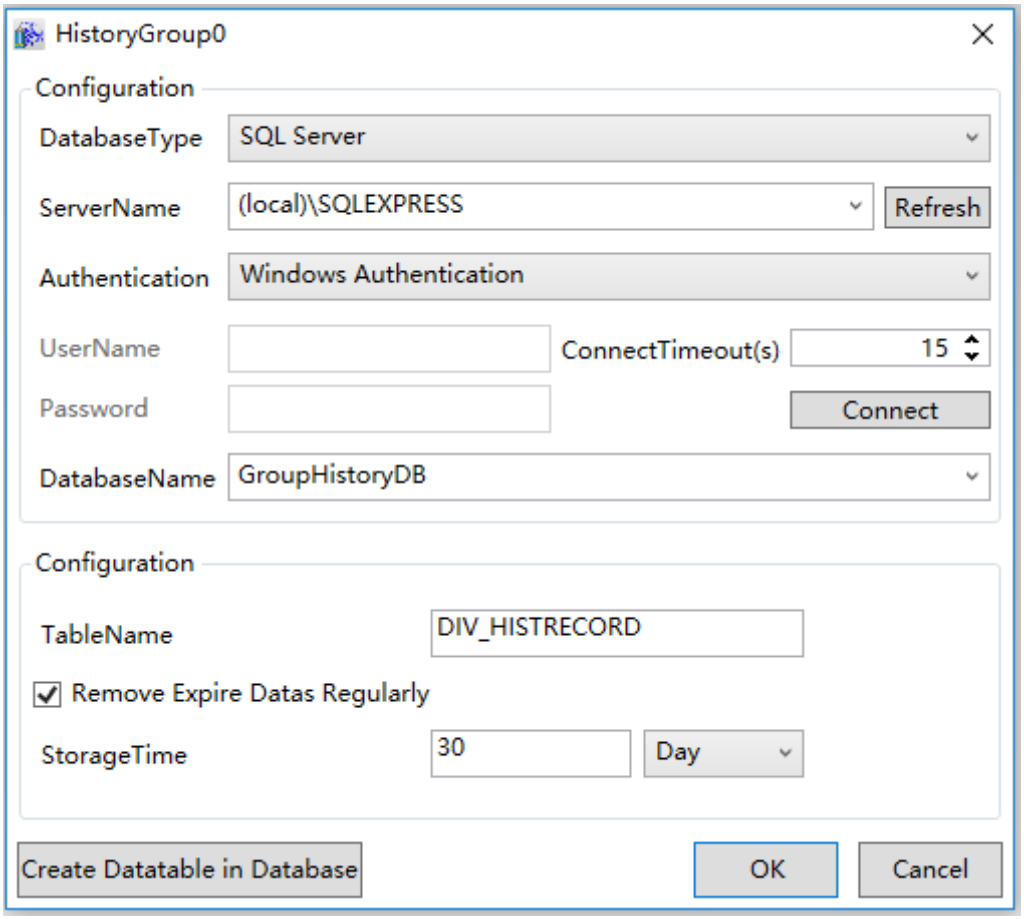


Parameter:

- **Name:** variable record name.
- **Column Name:** the column name of variable record corresponding datatable.
- **Expression:** the realtime data of variable record that need to be recorded is generated by associated expression.
- **Variable Type:** the data type generated by variable record ,the corresponding column type in the datatable.

- **Storage Type:** it is divided into general storage and subtract storage.General storage: Every time the condition is triggered, the real data generated by expression are recorded；subtract storage: the difference between the current data and last data generated by expression.

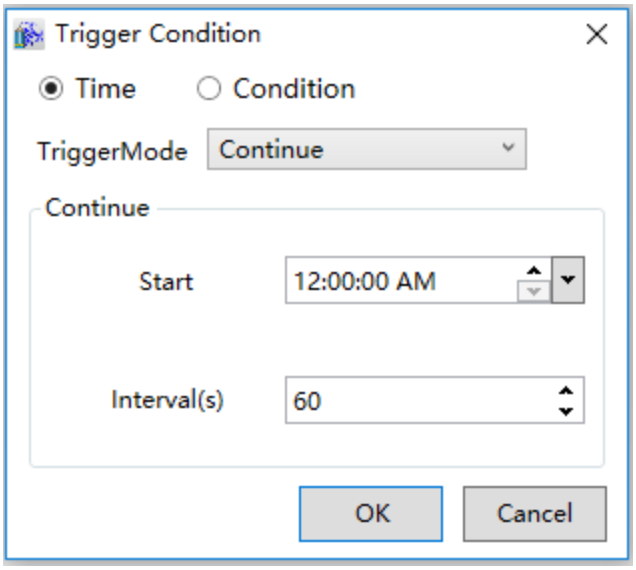
Step2: Configure the database:



Parameter:

- **DatabaseType:** Configure database connection information.Current support: SQL Server Compact,SQL Server,Oracle,My SQL.
- **TableName:** The name of variable record group corresponding datatable.
- **Remove Expire Datas Regularly:** User can choose the storage time of the data recorded by group record,if the data are overdue, they will be cleared regularly.
- **Create Datatable in Database:** Click the button and DIAView will create a new datatable in the database according to variable record configuration and database configuration.

Step3: Trigger Condition:



Configuration conditions are divided into two type according to the trigger style:TimeScript(start,stop,one time,continue,weekly,monthly)and ContidionScript(value change,while true,while false,on true and on false). The configuration is the same as the user script.please refer to chapter "16 User script ".

15. Recipe

15.1 Overview

In industrial production, recipe is used to provides the method and material ratio for the elements of the product; in the DIAView software, recipe is the collection of parameter values that corresponds to the variables during the producing. For example, the recipe of producting cakes lists the elements to produce the cake (for example flour, eggs, sugar and water etc.) and the amount needed for each element. Since different types of cakes need to be produced, their elements won't be the same completely ; therefore this recipe lists all the elements that are used by these cakes, but the amount of elements used for different cakes are different. The following table is the cake recipe of a certain cake factory:

Recipe	Chocolate cake	Matcha cake	Cranberry cake
Flour	130g	100g	150g

Cocoa powder	30g	0g	0g
Egg	4 eggs	3 eggs	3 eggs
White sugar	110g	60g	50g
Salad oil	45ml	0ml	0ml
Olive oil	0ml	0ml	50ml
Raisin	25g	0g	0g
Cream	50g	0g	0g
Butter	0ml	40ml	0ml
Water	300ml	200ml	100ml
Salt	0g	1g	0g
Matcha	0g	3g	0g
Cranberry	0g	0g	100g

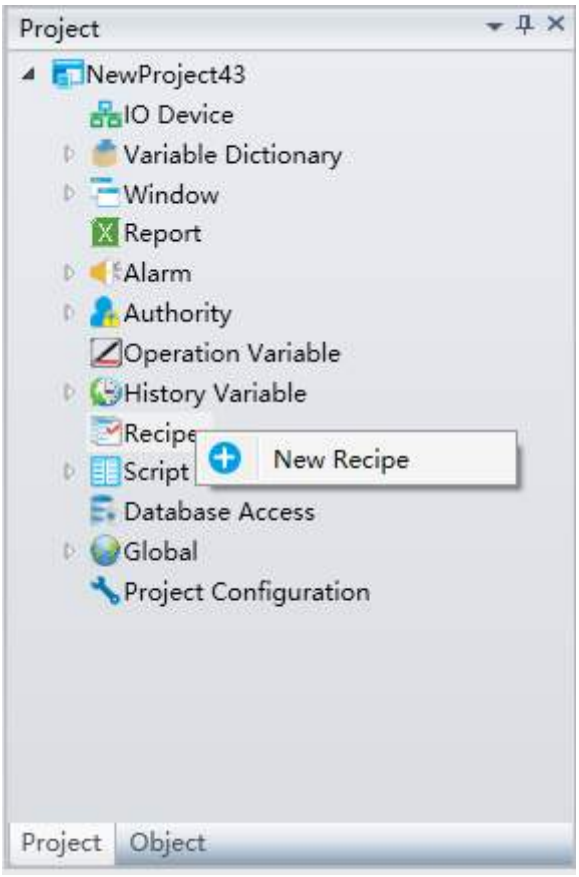
Recipe is the collection of various elements, and each element has its own parameter item. In the DIAView software,the recipe are the recipe “elements” that corresponds to the project variables ; the parameter of each recipe item is composed by variable values. Recipes can be the preset tables of elements value and the various recipe items, or it can be maintained during the running of the DIAView software.

The control “Recipe browser” in the DIAView software can be used to view and maintain recipes; it is easy to operate and highly efficient.

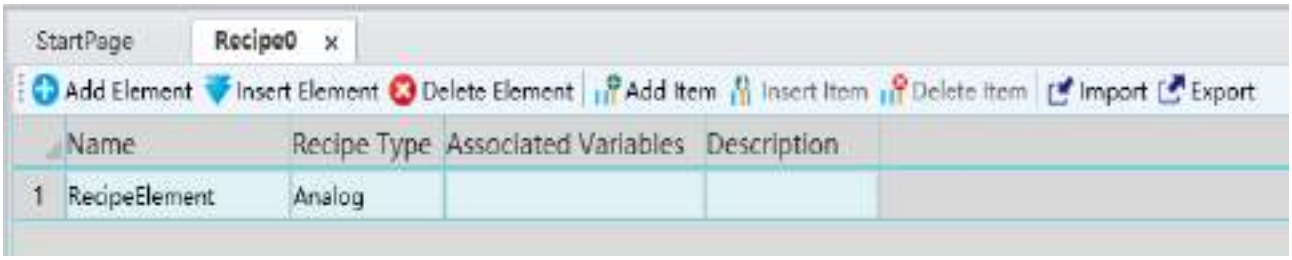
15.2 Recipe configuration

Configuring recipe means defining the elements of the recipe, the specific project and the value of the dosage; configuration steps are as follows:

Step 1: Open the project tree index window under the DIAView software development environment →Right-click the “Recipe” in the project tree index and click on “New Recipe” ; the system will create a recipe table with the default name “Recipe 0”, as shown in the figure below:



Step 2: Right-click on the added “Recipe 0” node and rename it to “Cake recipe”; double-click it to open the recipe table and click the “Add element” button to add an element, as shown in the figure below:



The meanings of each configuration in the window are as follows:

Name: Name of the recipe.

Recipe Type: Recipe element type; there are 3 types: analog value, digital value and string value.

Associated Variables: Project variable associated to the recipe element.

Description: Other information for the recipe element.

The functions of each button in the toolbar in the recipe window are as follows:

Add Element: Adds an element at the bottom of the recipe table with a default name. Each element is a row in the recipe table.

Insert Element: Inserts an element on top of the row selected in the recipe table with incremented naming of the selected row.

Delete Element: Deletes the selected recipe element.

Add Item: Adds recipe items, which means adding column in the recipe table; double-click the title of the column to rename of the column. The column names of “Name”, “Recipe type”, “Associated variable” and “Description” cannot be changed.

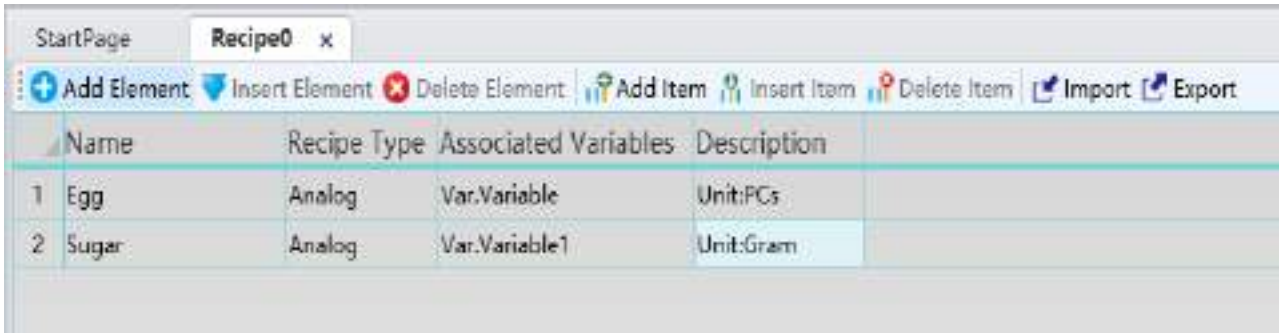
Insert Item: item: Inserts a recipe item at the left of the selected column; the column name uses incremented naming of the selected column.

Delete Item: Deletes the selected recipe item.

Import: Imports recipe data from Excel to the recipe table.

Export: Exports the data from the recipe table to Excel.

Step 3: Change the element name, select the recipe type,associate project variables etc.Then complete the configuration of a recipe table, as shown in the figure below:

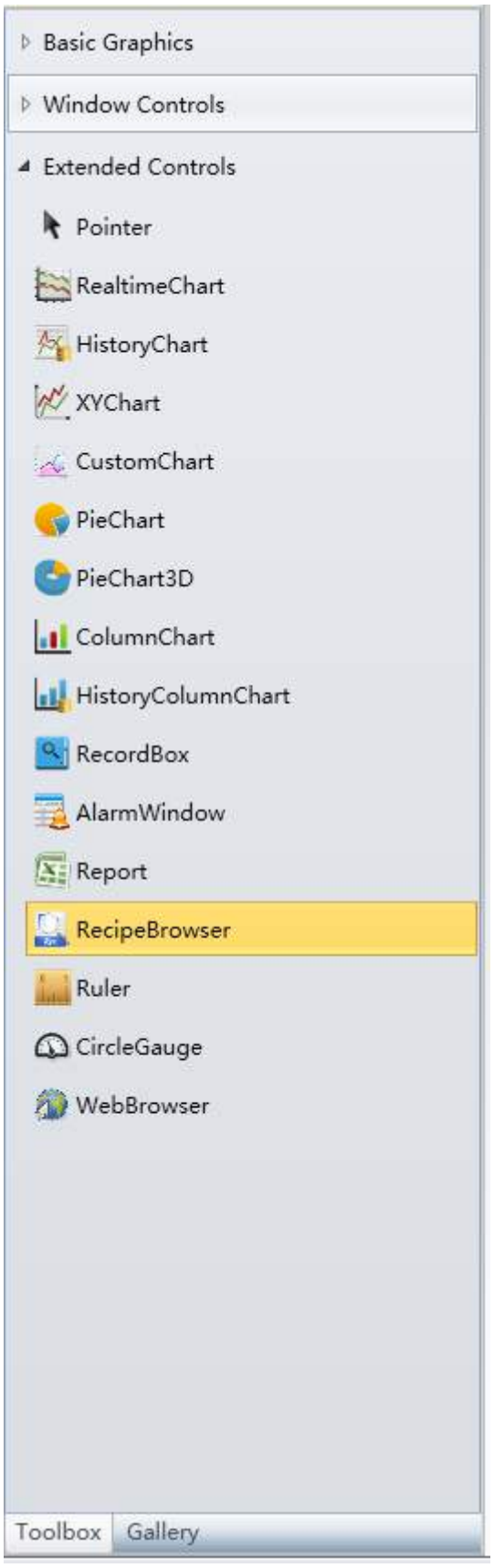


15.3 Recipe browser

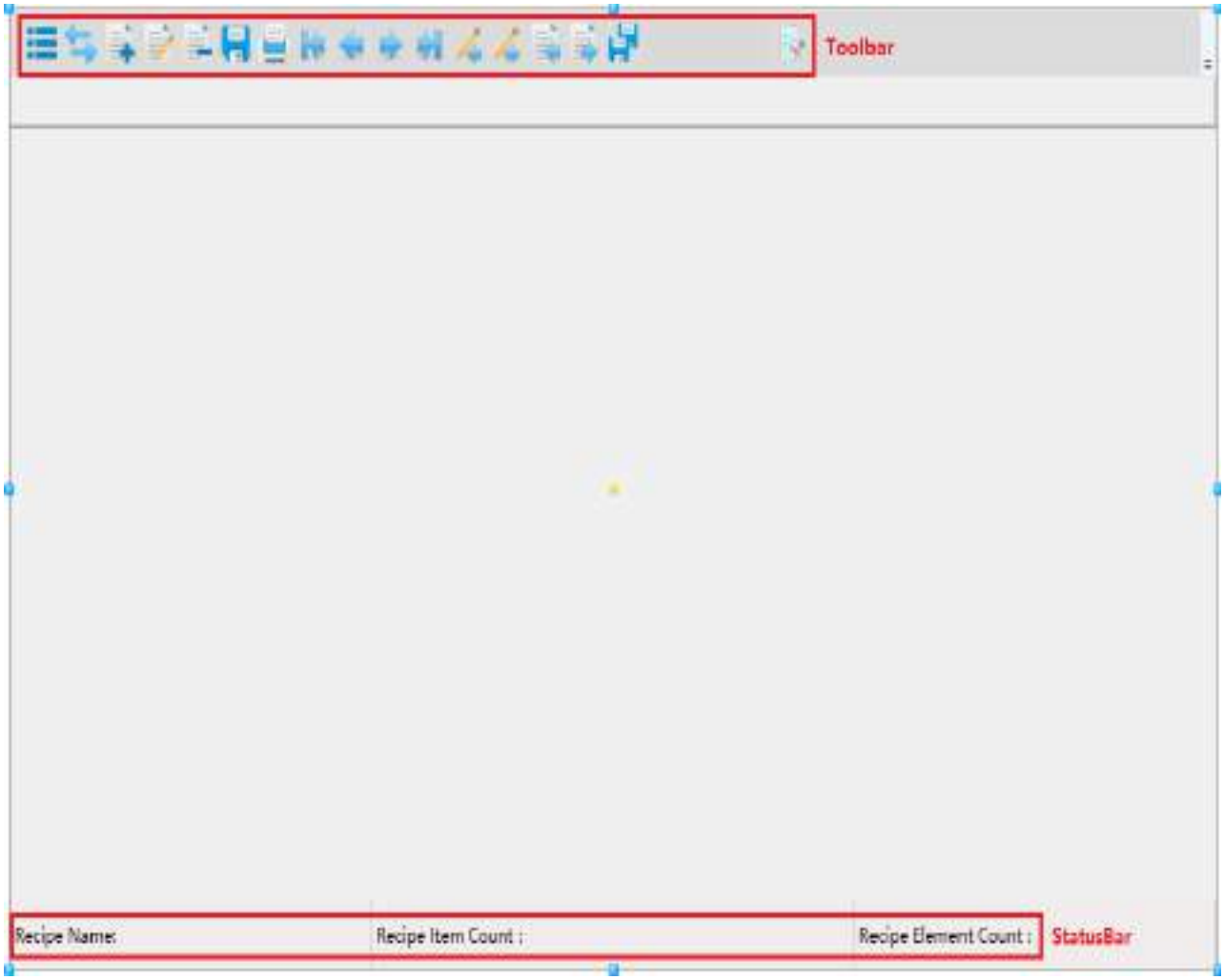
In the DIAView software,the recipe browser is used to browse and operate recipe; it can automatically identify all the recipe created in the system;Only by matching recipe names in the property of recipe browser under developing environment,user can view and maintain recipes under running environment. It is very easy and highly efficient to use.

- Usage of recipe browser:

open the extended controls group in the toolbox in the DIAView software development environment and click on “RecipeBrowser”:



➢ Composition of recipe browser:



The toolbar includes some buttons to operate the recipe table, including “Add recipe item”, “Edit recipe item”, “Write variable value to recipe item” and “Import recipe” etc.;

The display area of the recipe table will display the recipe data that are already configured in the project;

The StatusBar displays related nformation of the recipe.

➢ Functions of recipe browser toolbar buttons:

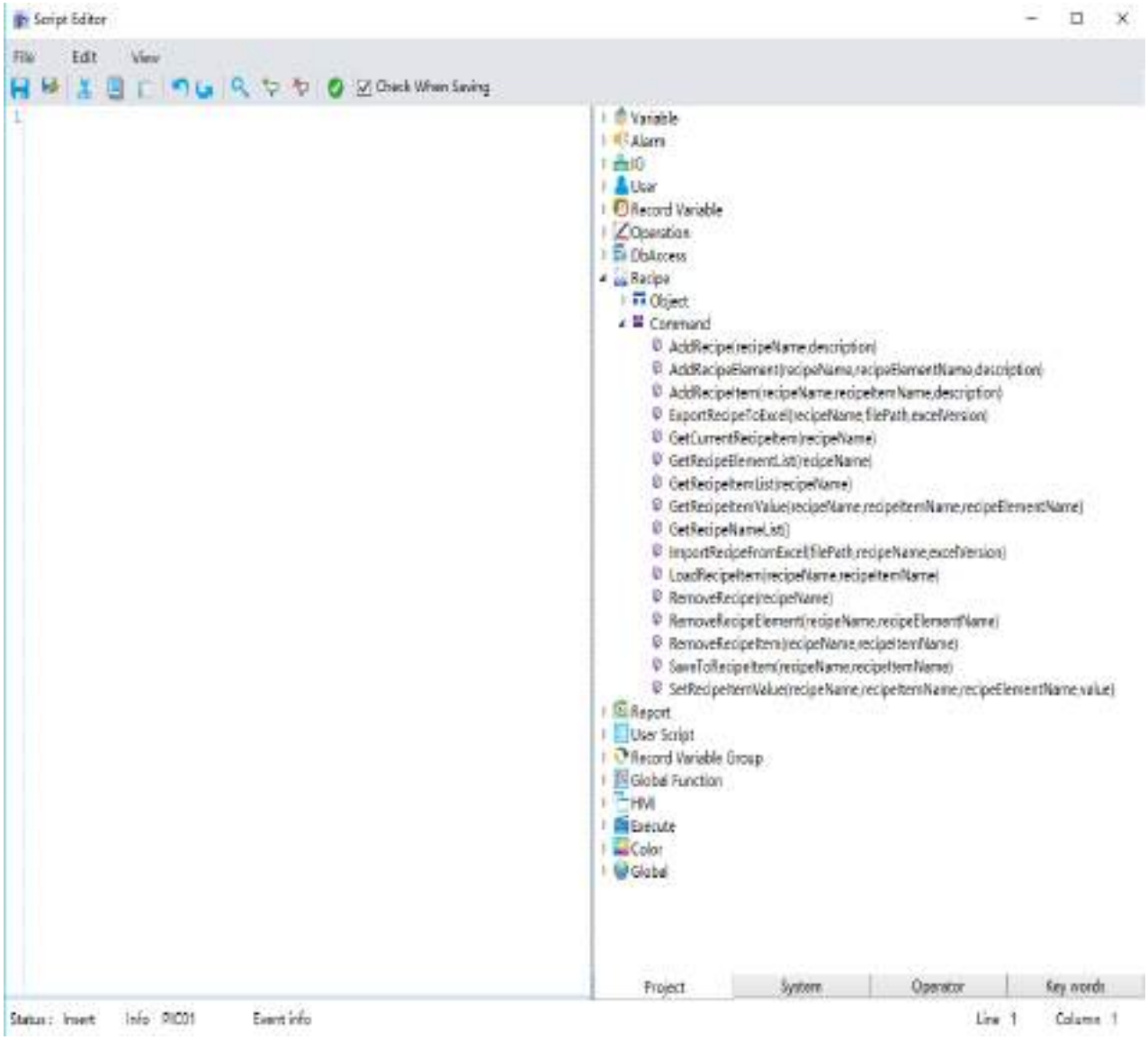
Functions of recipe browser toolbar buttons please refer to the chapter “7.7.12 RecipeBrowser”

➢ Recipe browser properties:

Recipe browser properties please refer to the chapter “7.7.12 RecipeBrowser”

➢ Recipe command:

Recipe commands can be used in event scripts to operate to the recipe, as shown in the figure below:



Please refer to the "DIAView software script and function manual"for the specific command scripts.

16. User Script

16.1 Overview

User scripts are programs with special functions written by users; it can complete and expand the functions of the DIAView. When a user writes programs in the project, it can flexibly use and control the

various resources of the DIAView such as graphic objects and commands etc., and it is able to improve and optimize system workflow and increase the efficiency of execution.

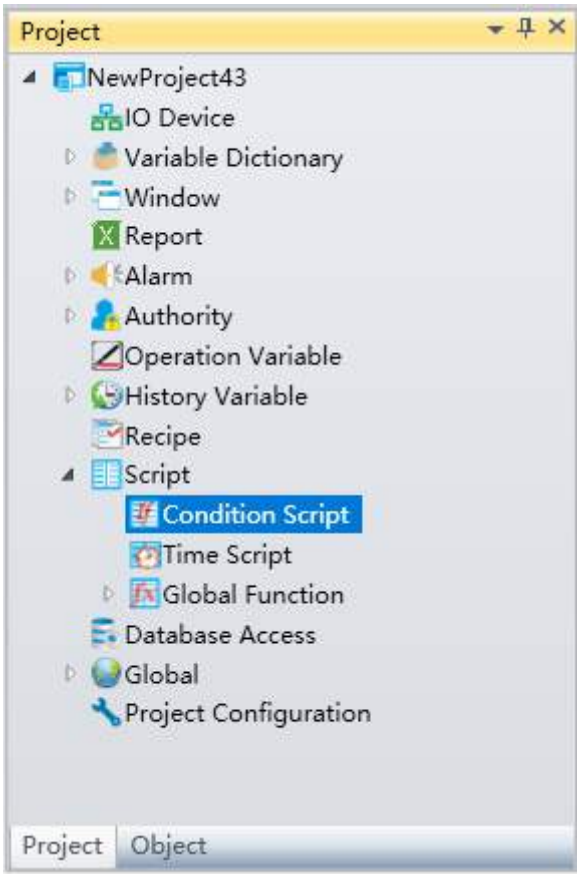
The DIAView software has a built-in script editor provided for users to write customized programs; it uses the VB Script language. VBS is a lightweight interpreted scripting language that is easy to learn and has powerful functions; the script editor has flexible and intelligent reminder function that allows users to write programs more easily, conveniently and highly efficiently.

The DIAView has two types of user programs: ConditionScript and TimeScript.

16.2 ConditionScript

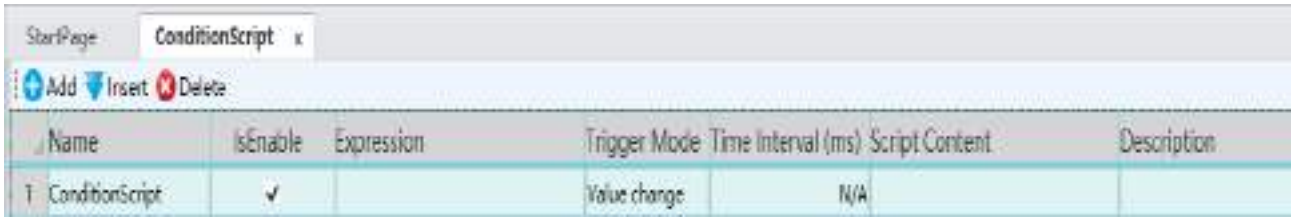
ConditionScript are customized programs that the system will execute when items in the project satisfies the setting conditions; configuration steps are as follows:

Step 1: Open the project window tree index in the DIAView development environment→ open the “Script” node, as shown in the figure below:



Step 2: Double-click the “ConditionScript” sub-node to open the ConditionScript window; click the “Add”

button to add a program configuration row, as shown in the figure below:



The meanings of each field in the ConditionScript are as follows:

Name: Name of the ConditionScript (this name cannot be the same name as in the "TimeScript").

IsEnable: Selecting the checkbox means enabling this ConditionScript, or else it means not enabled (default is enabled).

Expression: Sets the condition expression.

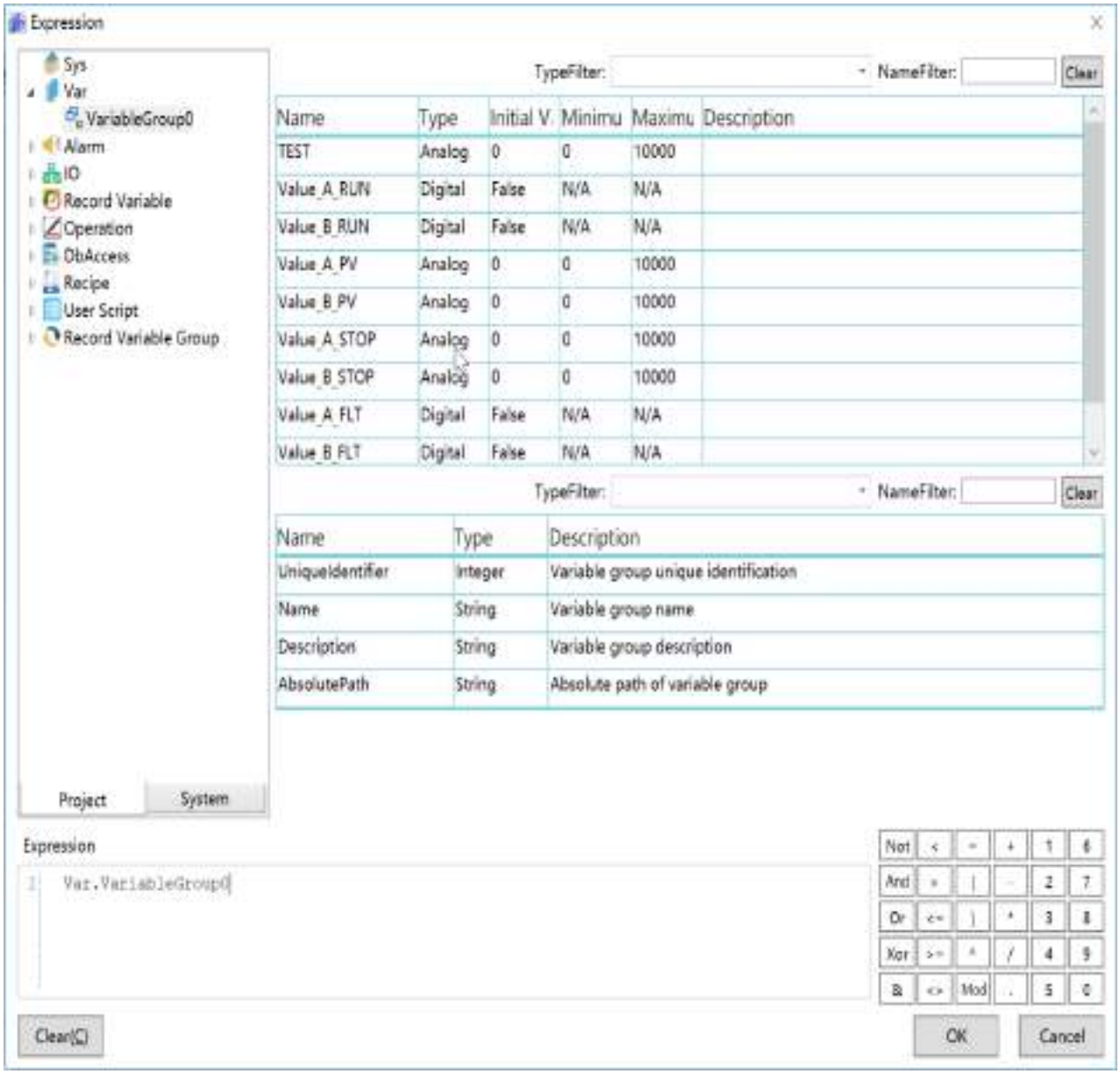
Trigger Mode: There are 5 types: "Value change", "While true", "While false", "One true" and"One false". Only when the condition is satisfied will the system be triggered to execute the ConditionScript.

Time Interval: Sets the time interval for the system to execute the configured program; this can only be set when the trigger mode is "One true" and "One false".

Script Content: Writes the script program.

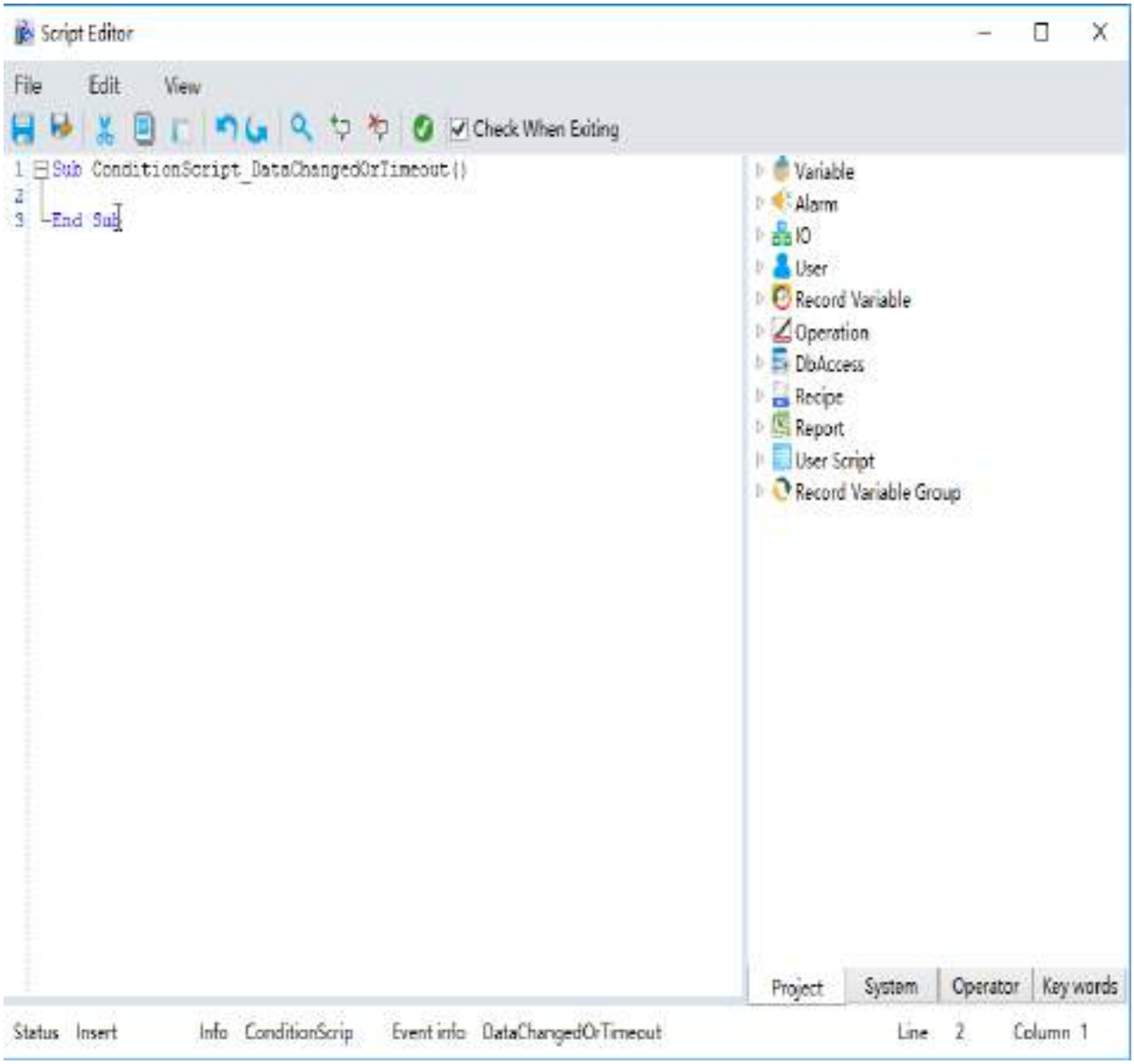
Description: Explanation information of the ConditionScript.

Step 3: Click the button in the "Expression" field to open the condition expression configuration window and set the expression, as shown in the figure below:



Step 4: Select the trigger mode; when the setting is “One true” or “One false”, the Time Interval must also be set (unit: milliseconds);

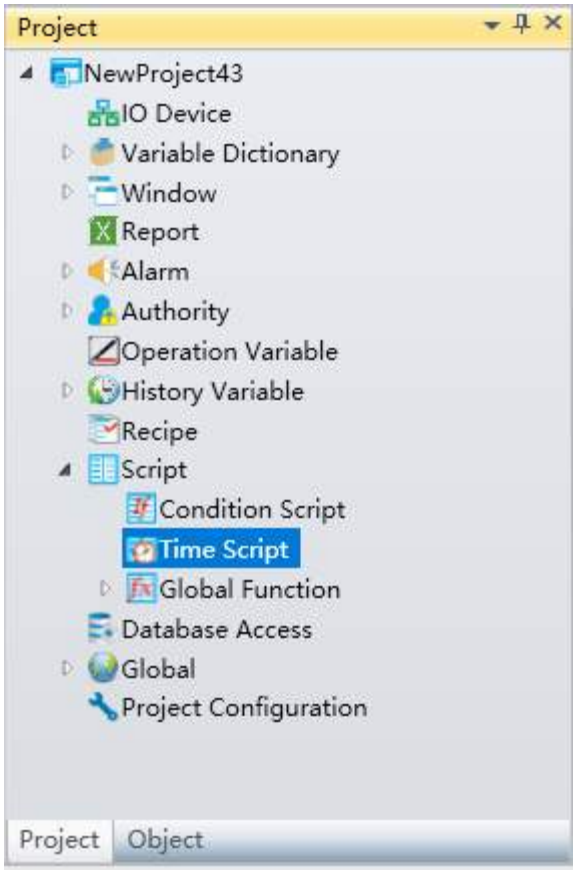
Step 5: Click the button in the “Script Content” field to open the script editor and write the script program, as shown in the figure below:



16.3 TimeScript

TimeScript is a user-defined program that is executed by the system at startup, during operation, or when the project stops. The configuration steps are as follows:

Step 1: Open the project window tree index in the DIAView development environment,→ open the “Script” node, as shown in the figure below:



Step 2: Double-click the “TimeScript” sub-node to open the TimeScript window; click the “Add” button to add a program configuration row, as shown in the figure below:

The meanings of each field in the TimeScript are as follows:

Name: Name of the TimeScript (this name cannot be the same name as in the “ConditionScript”).

IsEnable: Selecting the checkbox means enabling this TimeScript, or else it means not enabled (default is enabled).

Trigger Mode: There are 6 types: "Start", "Stop", "One-time", "Continue", "Weekly" and "Monthly". Which is the time for triggering the system to execute the configuration program.

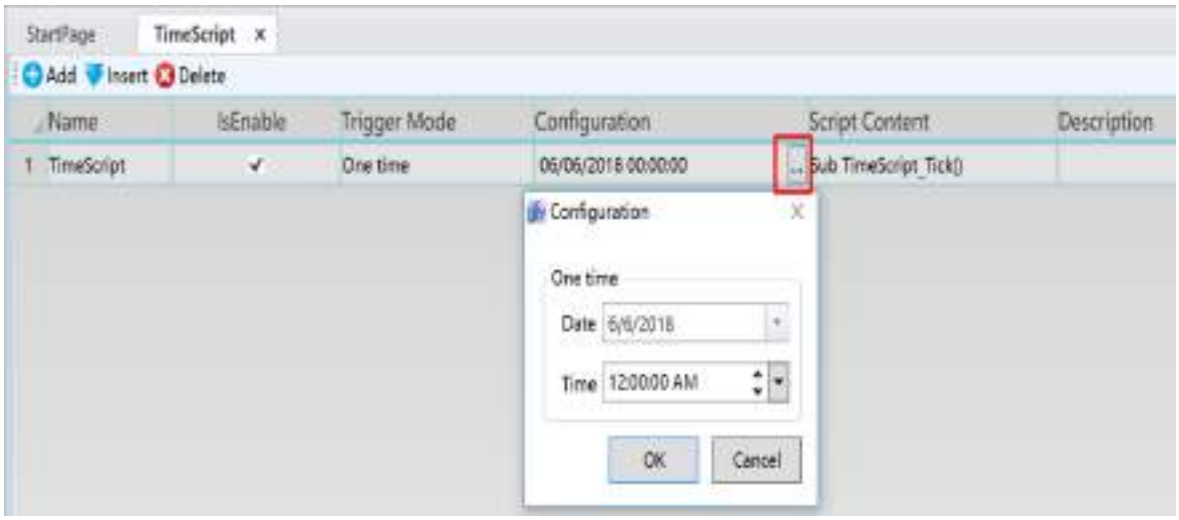
Configuration: Sets the time condition for the system to execute the configuration program; this

can only be set when the trigger mode is “One-time”, “Continual”, “Weekly” and “Monthly”.

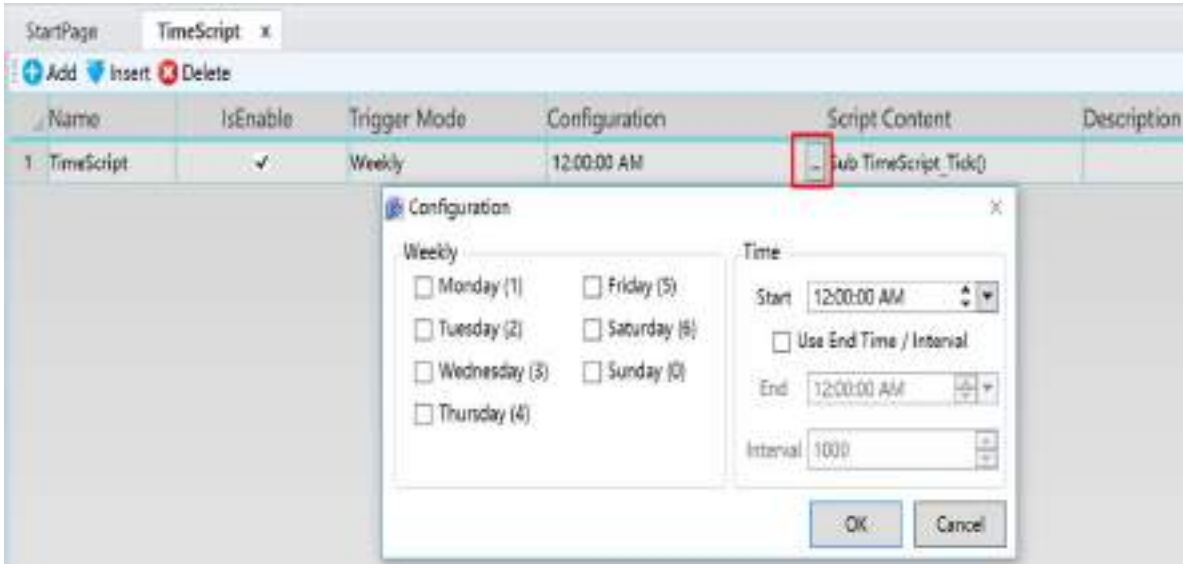
Script Content: Writes the script program.

Description: Explanation information of the TimeScript.

Step 3: Select the Trigger Mode; TimeScript must also be set when the trigger mode is set as “Onetime”, “Continue”, “Weekly” and “Monthly”. The figure below is the configuration window that appears when the Trigger Mode is set as “One-time” and the button in the “Configuration” field is pressed,as is shown in:



The following figure is the configuration window that appears when the Trigger Mode is set as “Weekly” and the button in the “Configuration” field is pressed,as is shown in:



The meanings of each configuration in the window are as follows:

Weekly: Selects the weekly triggering time.

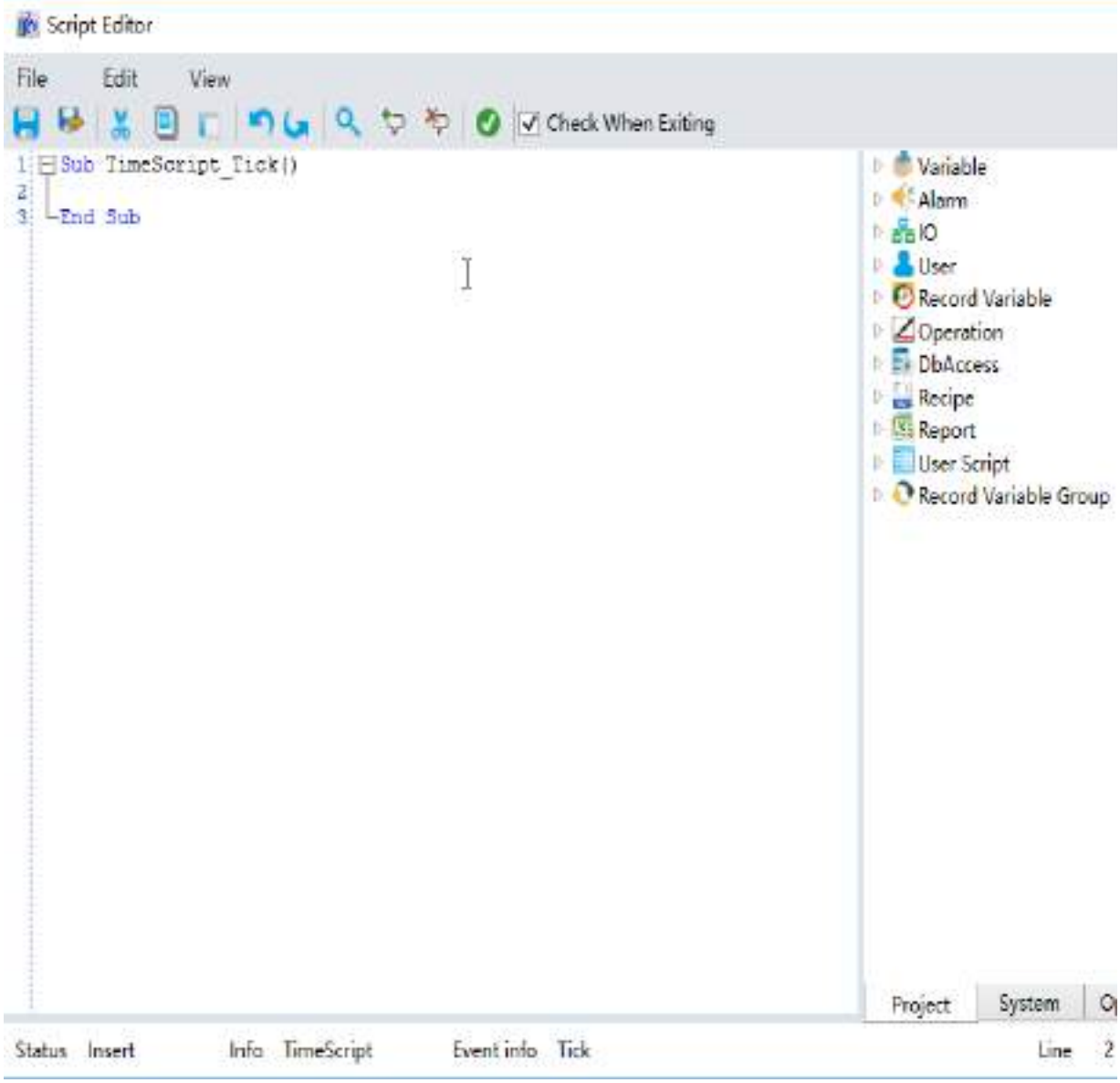
Start: Sets the start time.

Use End Time/nterval: If checked, the two settings “End” and “Interval” can be set; or else they cannot be set.

End: Sets the end time (if the set start time > end time, the trigger period will be: starts from the start time set and ends at the end time set on the following day. If the setting start time < end time, the trigger period will be starts from the setting start time and ends at the setting end time).

Interval: Sets the interval time, unit: milliseconds (ms).

Step 4: Click the button in the “Script Content” field to open the script editor and write the script program, as shown in the figure below:



16.4 Global Function

16.4.1 Overview

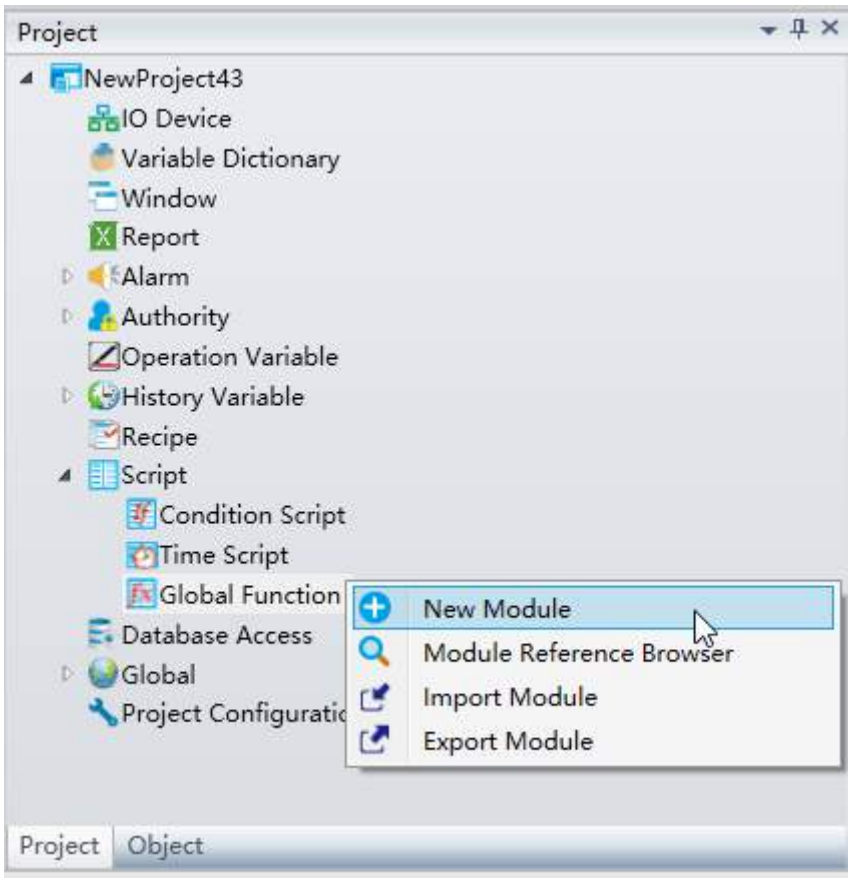
DIAView Globle Function is to package the reusable script code, avoid code dispersion, reduce maintenance costs and debugging difficulties.

16.4.2 Module Configuration

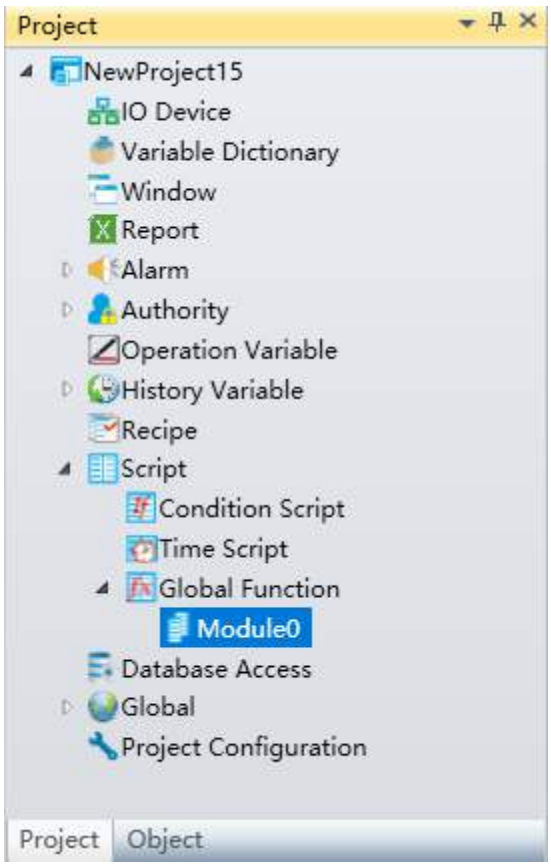
In global function, a module consists of a series of methods, equivalent to a list

1. Add new module:

Step 1: Right-click on the “Script” node child node "Globle Function" in the project window tree index and then click the “New Module” item in the right-click menu, as shown in the figure below:

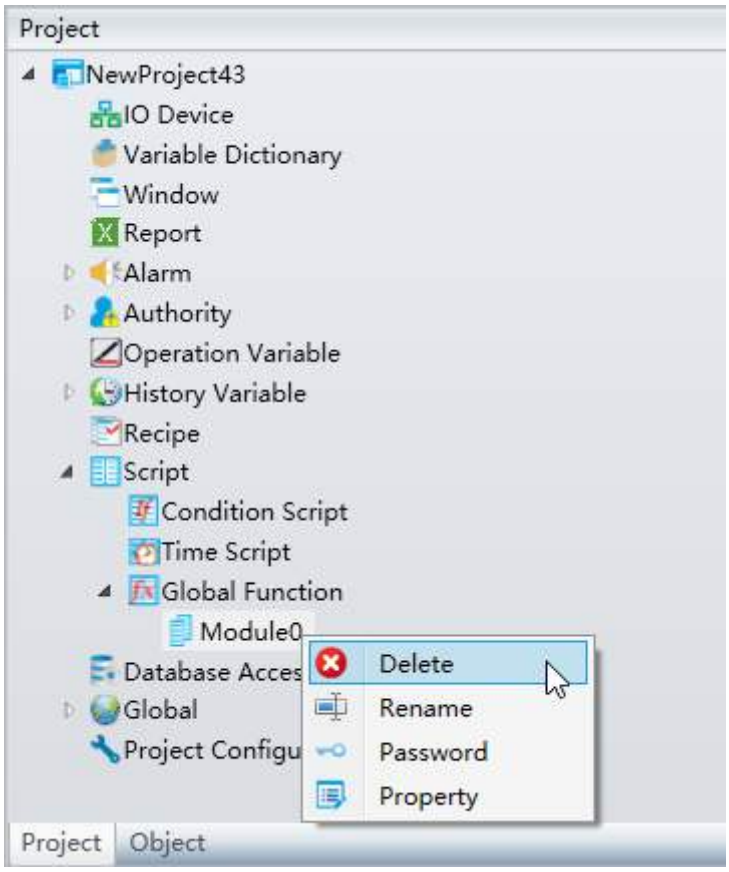


Step 2: After clicking “New Module”, the system will generate a window sub-node under the “Global Function” node using a default name, and the new window will be opened in the sketchpad work area, as shown in the figure below:



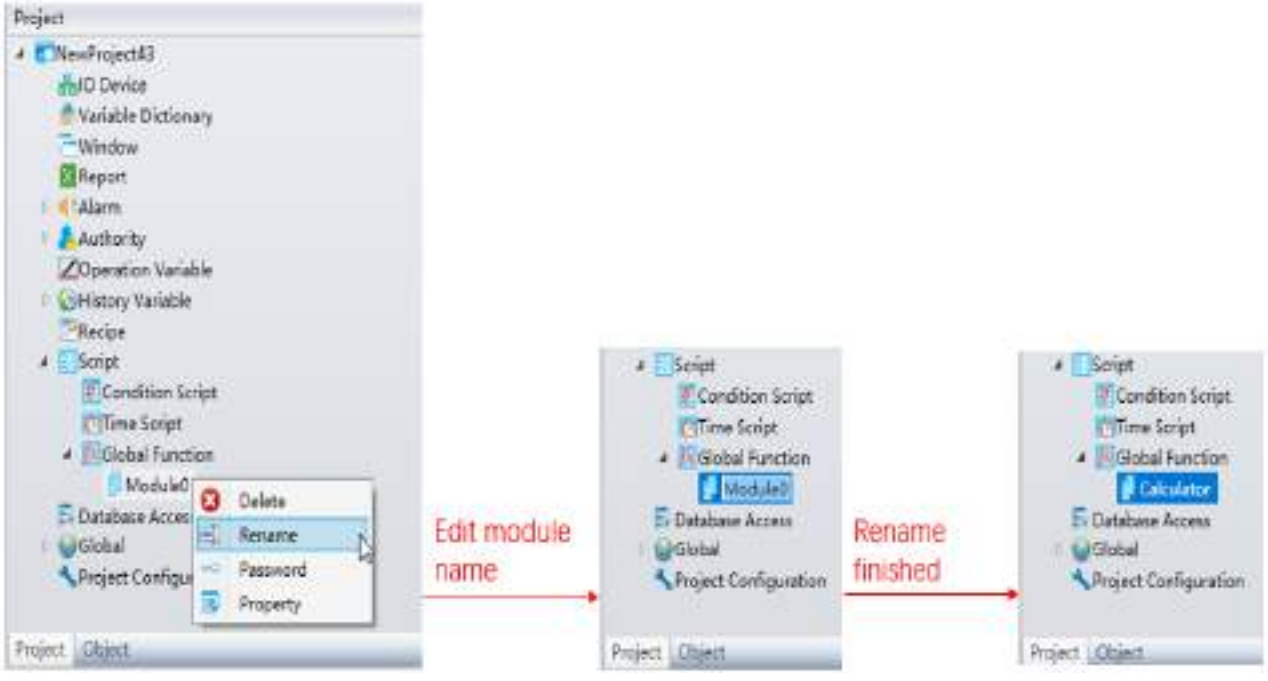
2. Delete Module:

Select the module node to delete from the project window tree index and right-click on it, then select the “Delete” item from the right-click menu to delete the window, as shown in the figure below:



3. Rename Module:

Select the module node to rename in the project window tree index and right-click on it, and the select the “Rename” item from the right-click menu. The module name will become an editable status and just input the new module name, as shown in the example in the figure below:



✧ **Module naming rules:**

(1) Composed of English letters, numbers, Chinese characters and underscore, and can only begin with an English letter or Chinesecharacter;

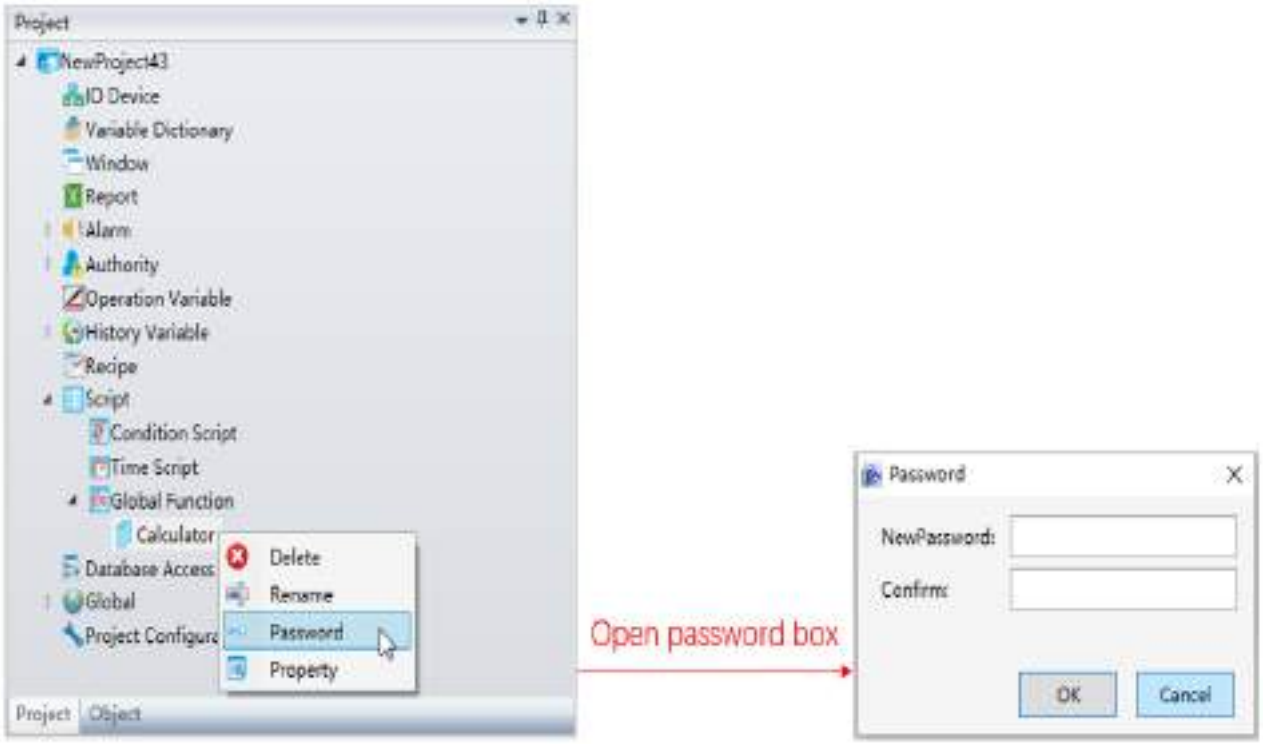
(2) Not case sensitive;

(3) The length cannot exceed 200 characters and cannot exceed 25 Chinese characters;

(4) Name can not repeat with other module.

4. Encrypt Module:

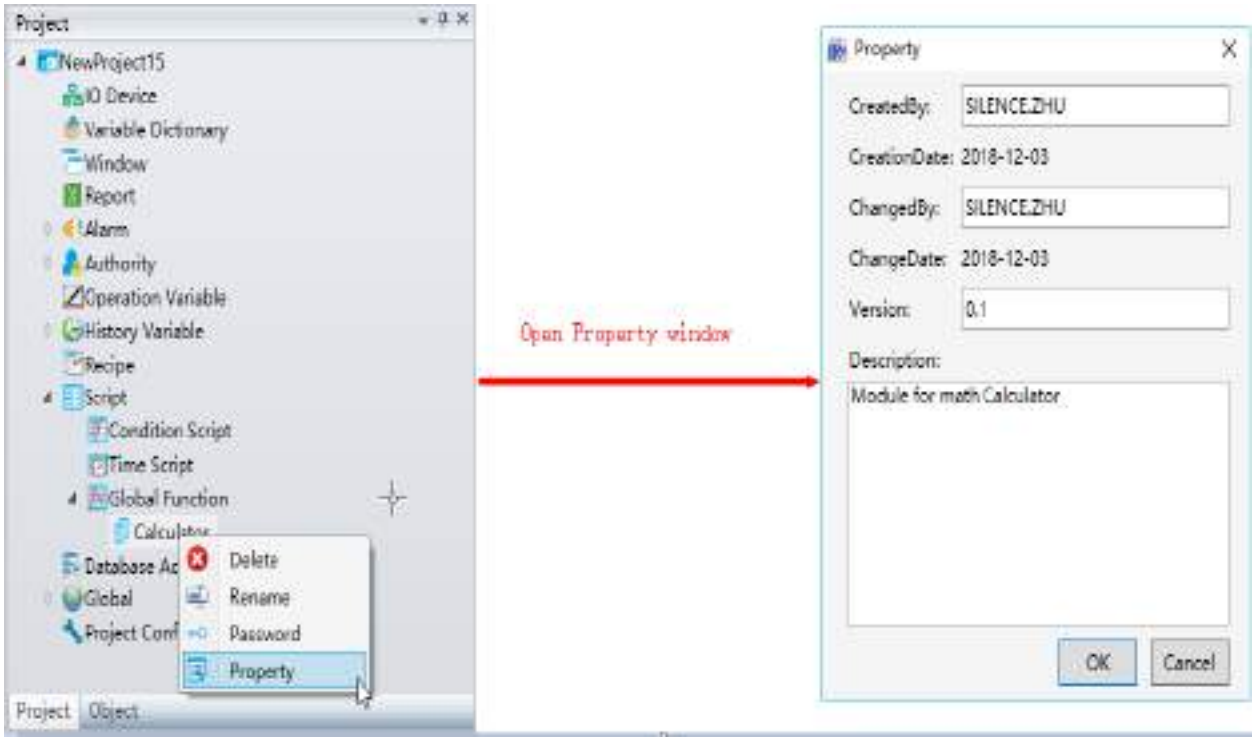
Select the module node to encrypt from the project window tree index and right-click on it, then select the “Password” item from the right-click menu to encrypt the module, as shown in the figure below:



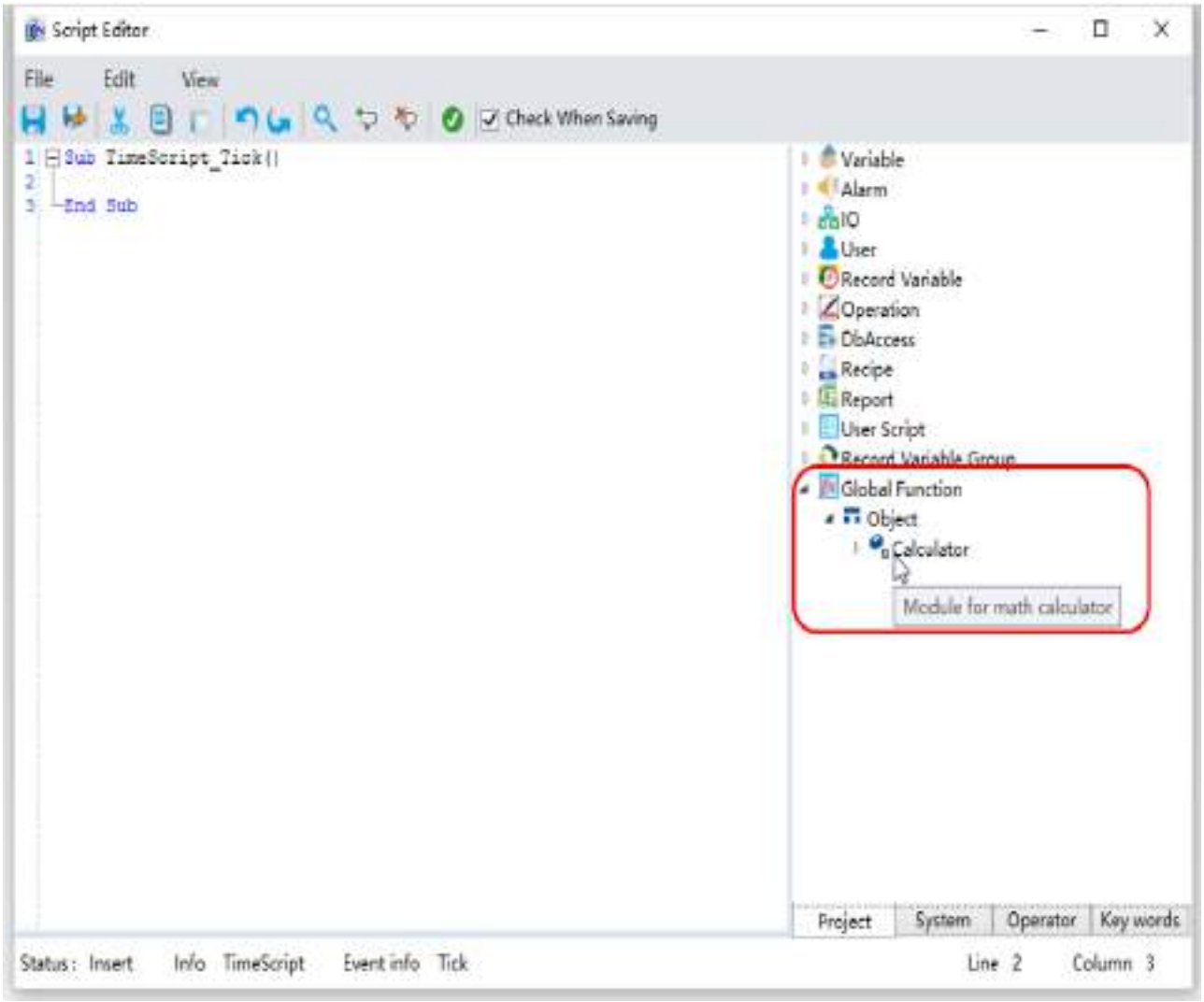
✧ **Note: If module has encrypted, click "Password" is to modify password, if new password is null, will cancel password.**

5. Module Property:

Select the module node to modify property from the project window tree index and right-click on it, then select the “Property” item from the right-click menu to modify the module property, as shown in the figure below:



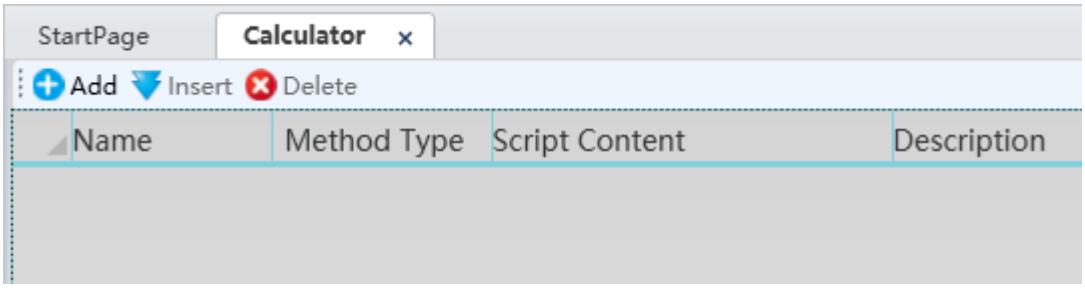
✧ **Note: The description of module will be displayed in script editor, as shown in the figure below:**



16.4.3 Add and Use method

1. Open module view

Open module window, as shown in the figure below:



Property:

Name: Method name

Method naming rules:

(1) Composed of English letters, numbers, Chinese characters and underscore, and can only begin with an English letter or Chinesecharacter;

(2) Not case sensitive;

(3) The length cannot exceed 200 characters and cannot exceed 25 Chinese characters;

(4) Name cannot be repeated in the same module.

Method Type: Type of method function

Sub : Not return value.

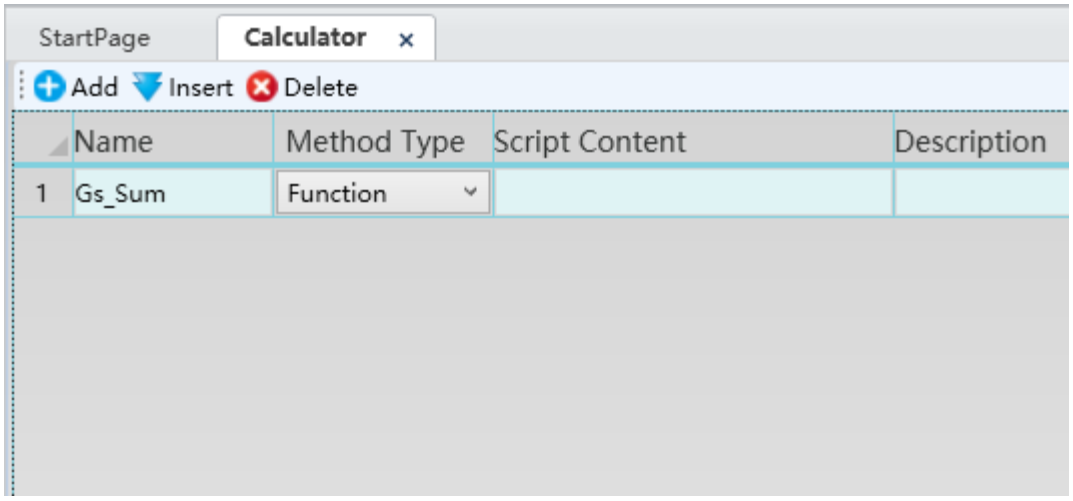
Function: Has return vaule.

Script Content: The body of method, edit in script editor.

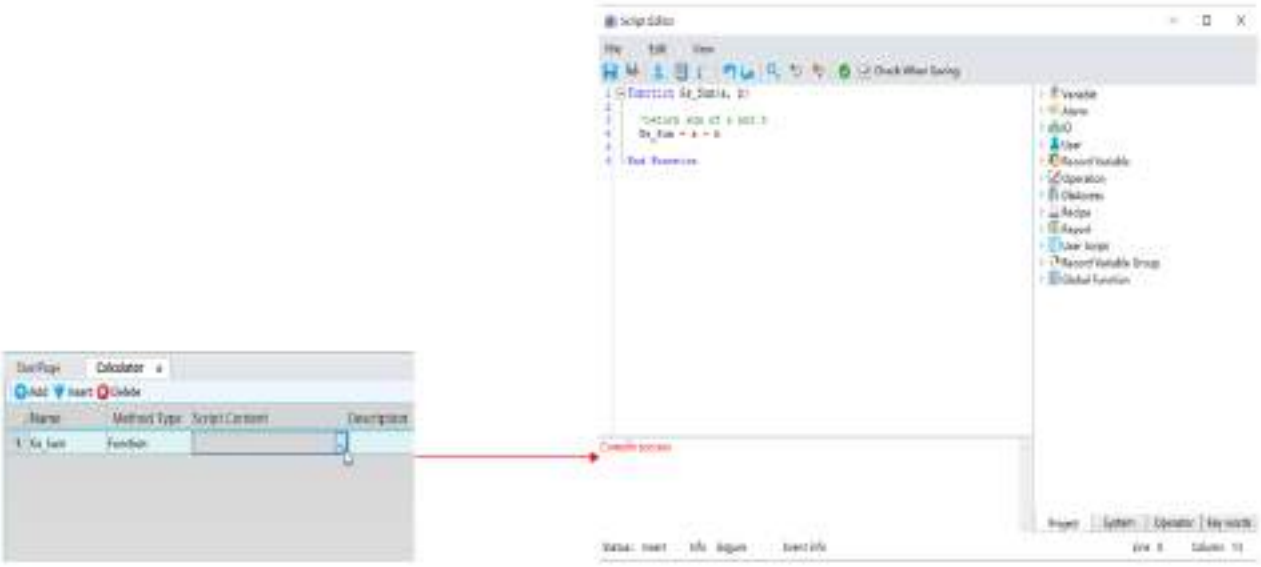
Description: The description of method, include function, parameter, and return vaule.

2. Add method

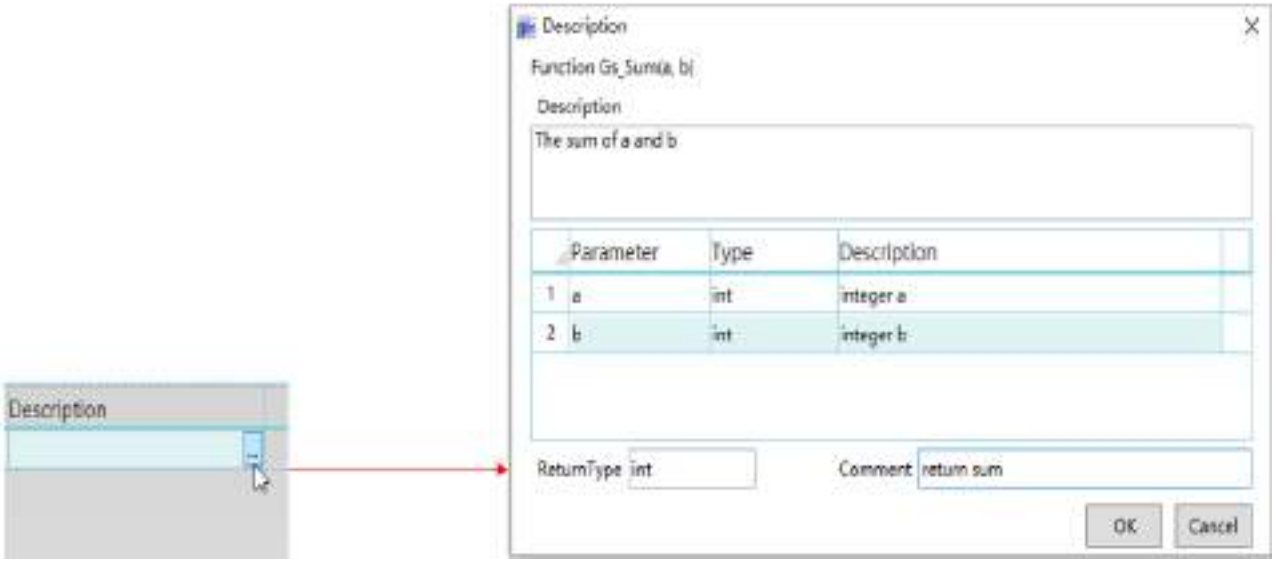
Step 1: Click "Add", add new method, modify name and method type, as shown in the figure below:



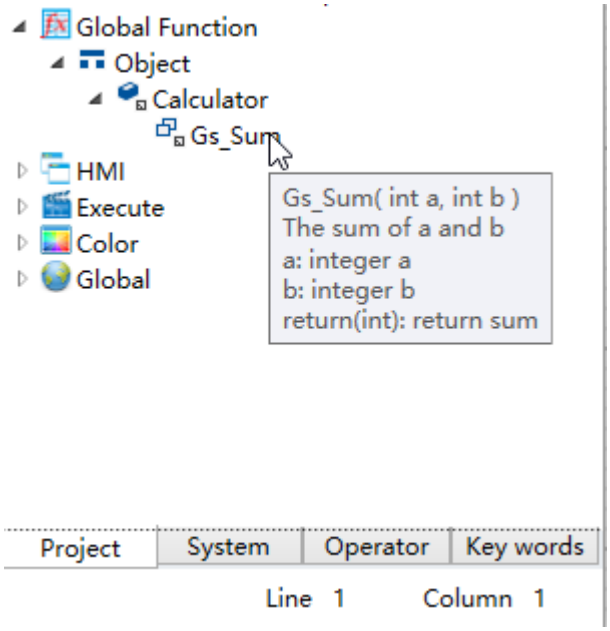
Step 2: Click button of the script content, into the script editor and compile script, as shown in the figure below:



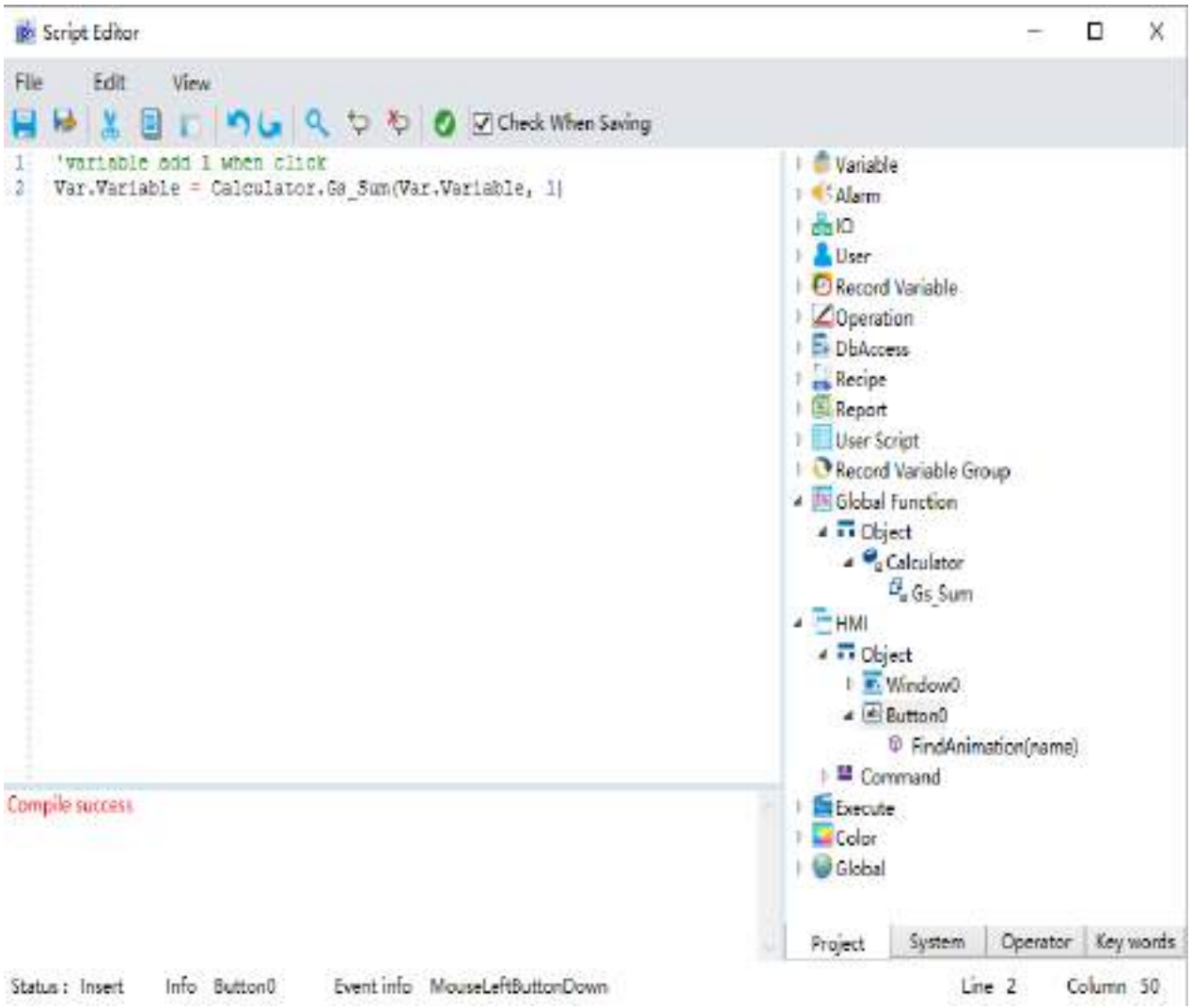
Step 3: Click button of the description, into the description window and modify method description, as shown in the figure below:



Step 4: The description of method will be displayed in script editor, as shown in the figure below:



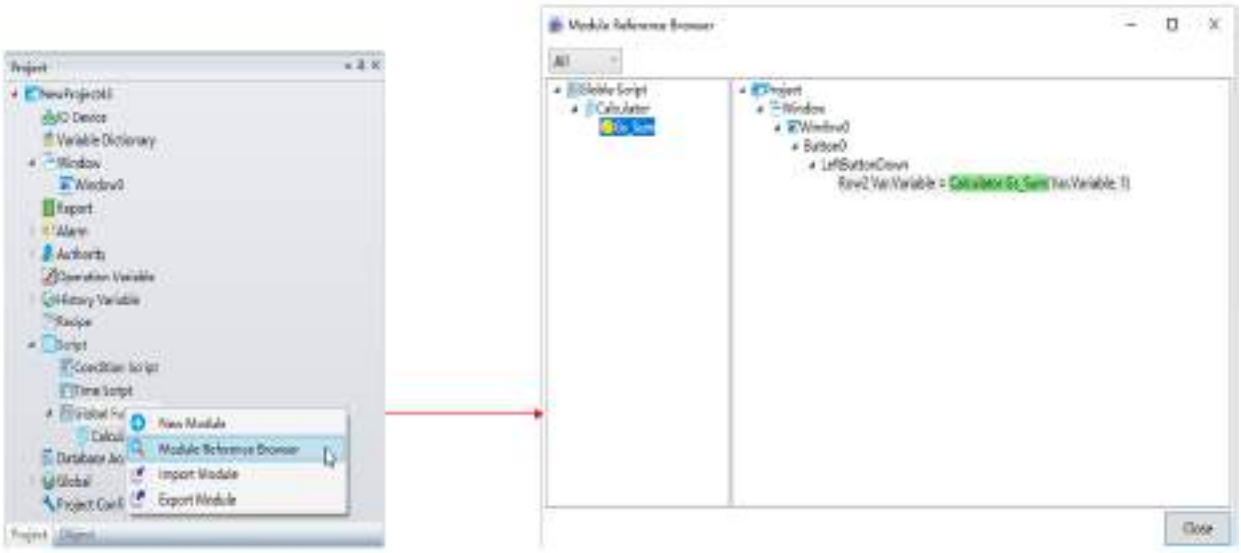
Step 5: method of global function can be used in other script, as shown in the figure below:



16.4.4 Reference and import export

1. Module reference browser

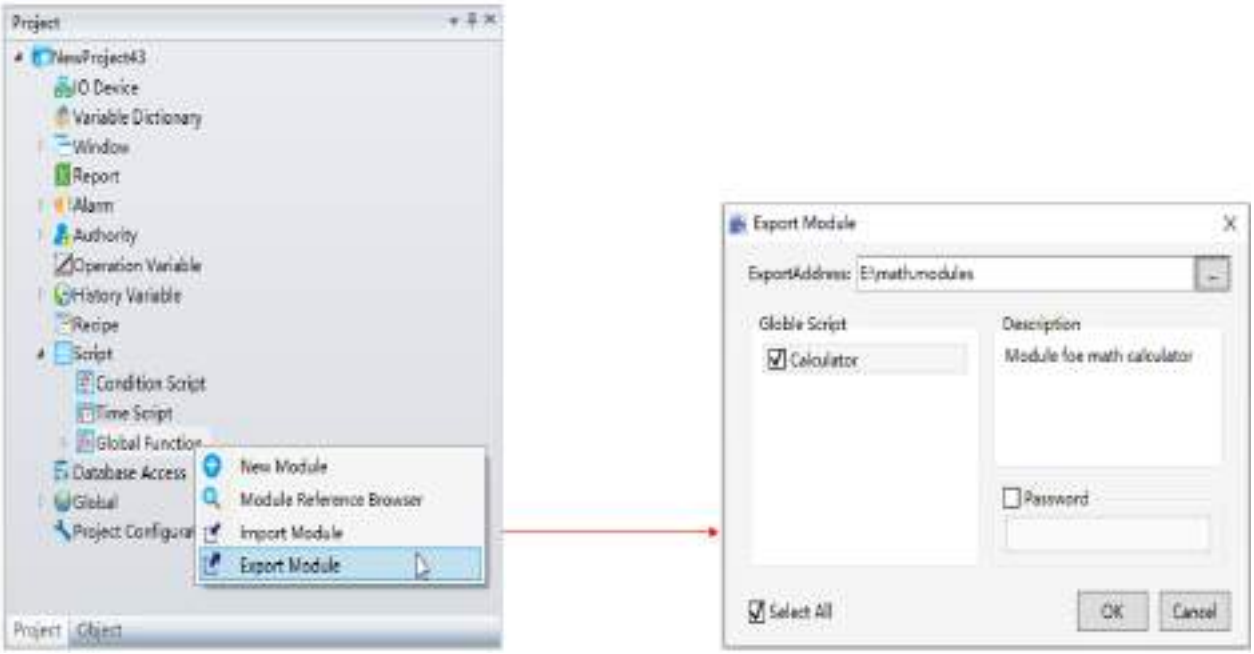
Right-click on the “Global Function” node in the project window tree index and then click the “Module Reference Browser” item in the right-click menu, as shown in the figure below:



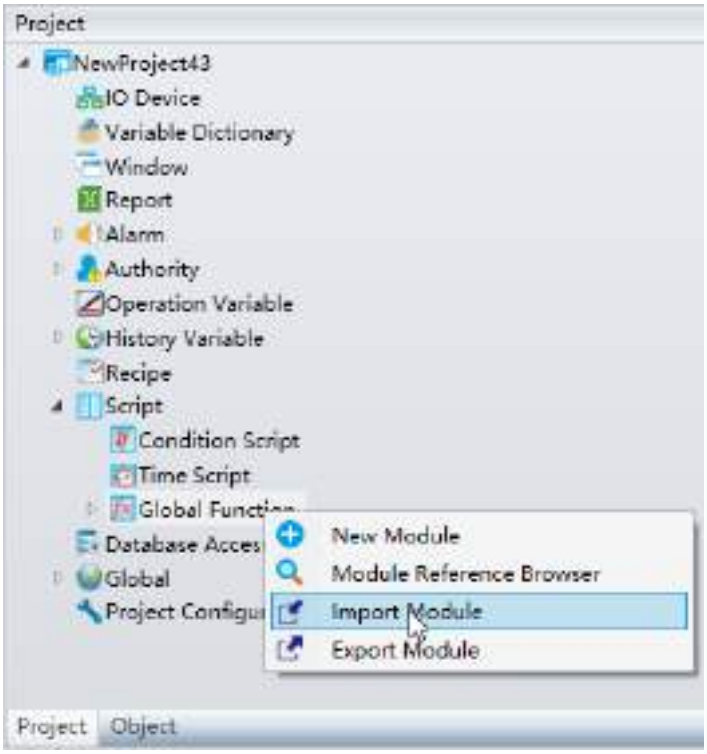
2. Module import and export

The file type to import and export is ".modules".

Step 1: Right-click on the “Global Function” node in the project window tree index and then click the “Export Module” item in the right-click menu to export module, as shown in the figure below:



Step 2: Right-click on the “Global Function” node in the project window tree index and then click the “Import Module” item in the right-click menu ,as shown in the figure below:



✧ **If import module name has exsit, the module can not import.**

17. Database Access

17.1 Overview

DatabaseAccess in DIAView allows users to access external database, query and modify the data in the database.

As long as users configure database connection in the DatabaseAccess, users can operate data in the database in the runtime.

The supporting database types: SQLServer\SQLServer Compact\Oracle\MySQL.

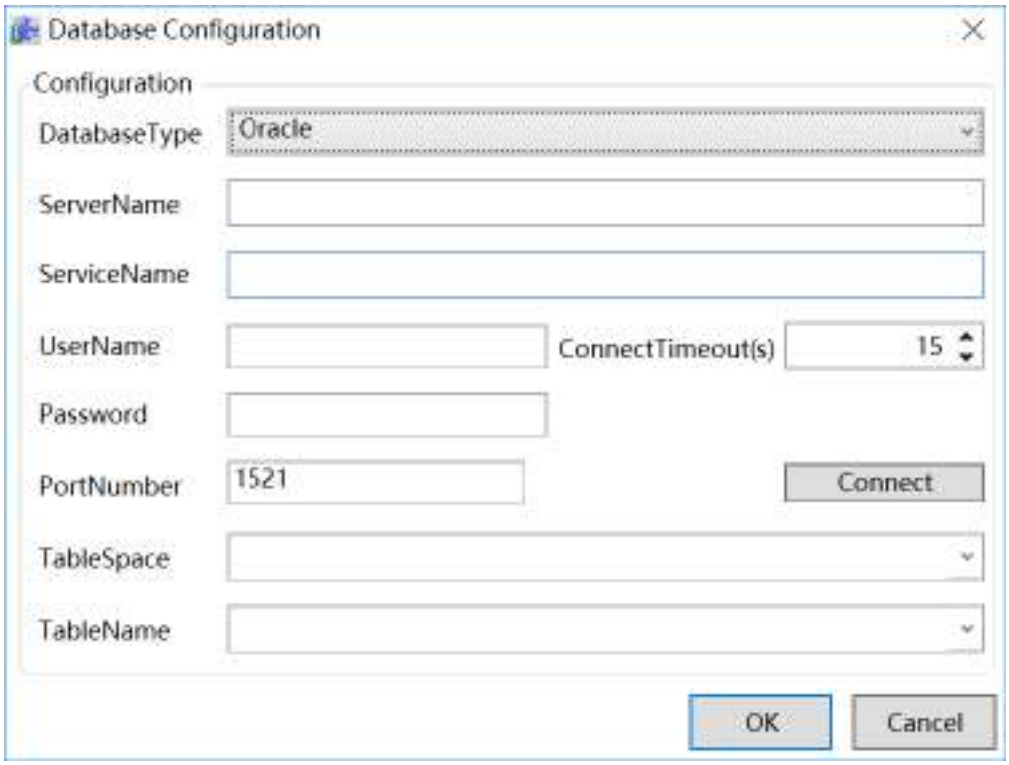
17.2 Database access configuration

DatabaseAccess configuration figure is as shown in the figure below:

	Name	Configuration	Paging	Page Size	Description
1	DatabaseAccess2	...	✓	50	
2	DatabaseAccess1		✓	50	
3	DatabaseAccess		✓	50	

➢ **Configure the database connection information:**

✧ Oracle database configuration is as shown in the figure below:



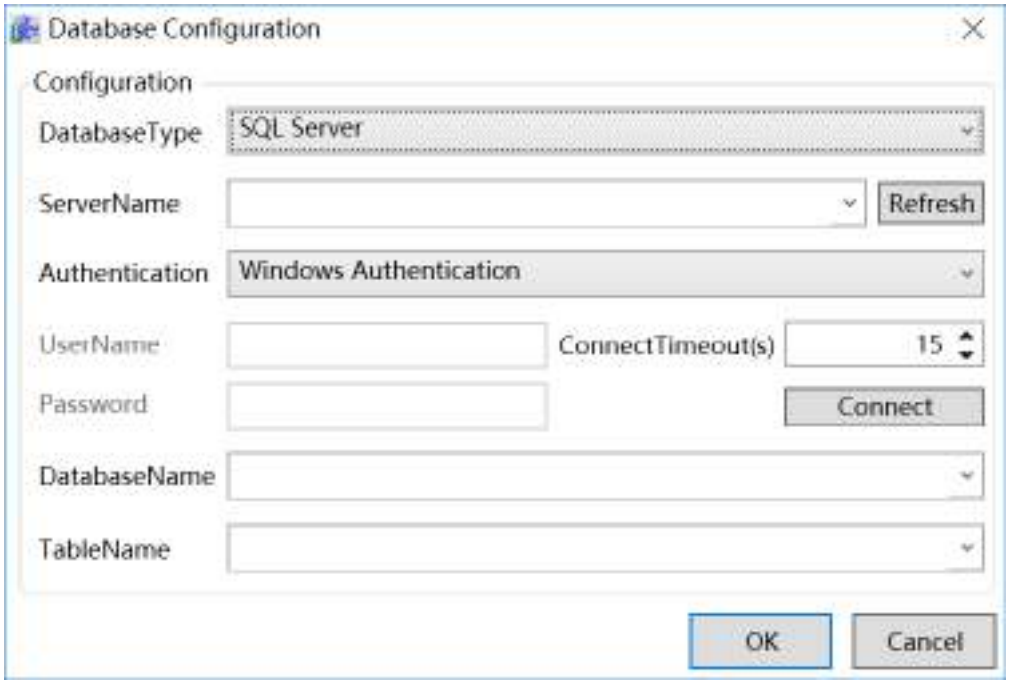
ServerName: The Oracle database host name or IP address.

ServiceName: The service name of Oracle database.

TableSpace: The table space to access,if it is null,Oracle will select the default table space.

PortNumber: Port number used by Oracle database.

✧ SQLServer database configuration is as shown below:



ServerName: Enter the server name or IP address of the database that needs to connect; click the pull-down button to search for servers automatically within the network.When accessing a remote database, you need to enter the PC name or IP address where the database server is located.

Authentication: There are two methods to authentication when logging in the server:Windows ID authentication and SQL ID authentication;if the database is configured on a local host;When selecting "Windows authentication", no user name and password need to be entered. When selecting SQL authentication, you need to enter the corresponding username and password configured in the SQL server.

UserName: Sets the user name for SQL ID authentication login.

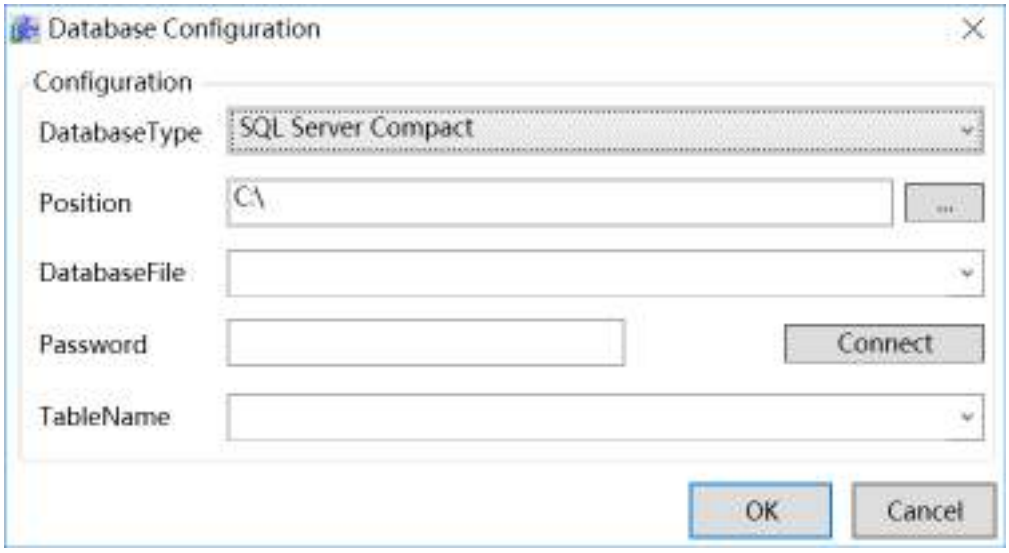
Password: Sets the user password used for SQL ID authentication login.

ConnectionTimeout: Sets the time for the database to switch connection or reconnect after connection failed, and reports an error if exceeded this time; unit: seconds.

DatabaseName: Enter the database name that need to access.

Connect: Tests whether it can connect to the database server.

✧ SQLServer Compact database configuration is as shown in the figure below:

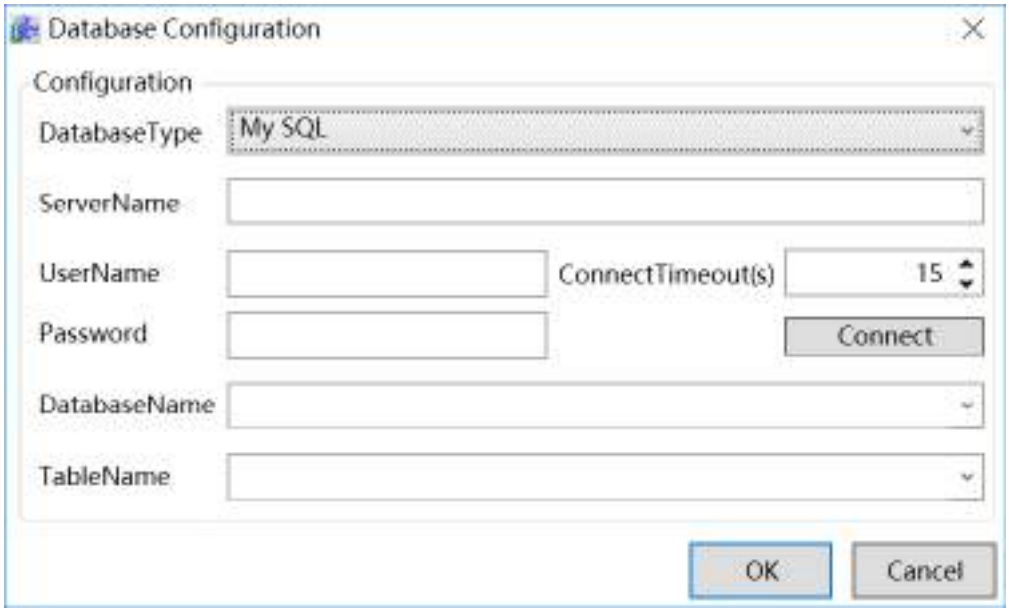


DatabaseFile: Enter the database file path and file; the button on the right can be clicked to select the directory.

Password: Database password; left empty if the password is not required.

Connect: Tests whether it can connect to the database.

✧ MySQL database configuration is as shown in the figure below:



ServerName: MySQL server name or IP address.

Username: Sets the user name for SQL ID authentication login.

Password: Sets the user password used for SQL ID authentication login.

ConnectionTimeout: Sets the time for the database to switch connection or reconnect after connection failed, and reports an error if exceeded this time; unit: seconds.

DatabaseName: Enter the database name that need to access.

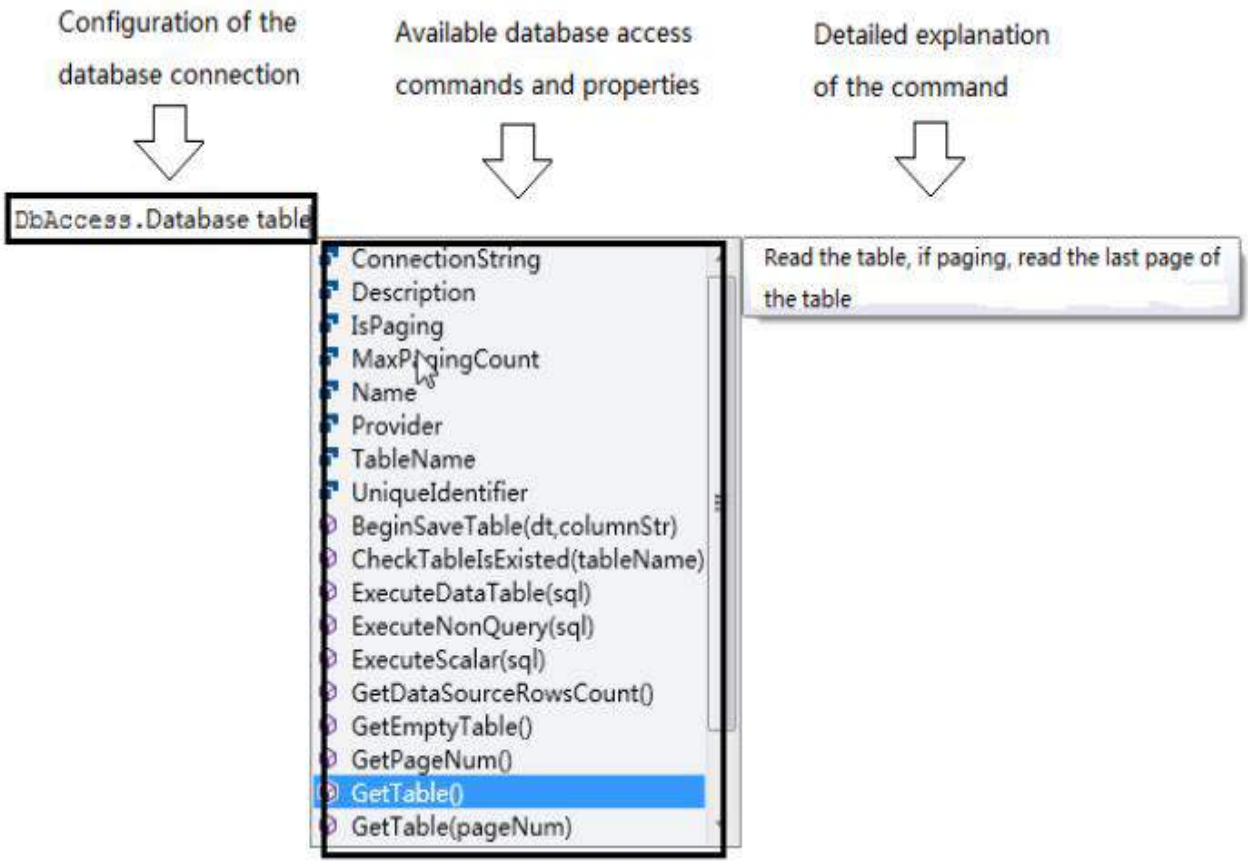
Connect: Tests whether it can connect to the database server.

- **Paging :** If selecting the paging, the DIAView will paging the data into a lot of pages according to the setting paging when DIAView reads data, only one page data is read at one time；If not, all the data is read.Due to the large amount of data in the database , reading all data once will take up a lot of memory,it will cause the program lagging.

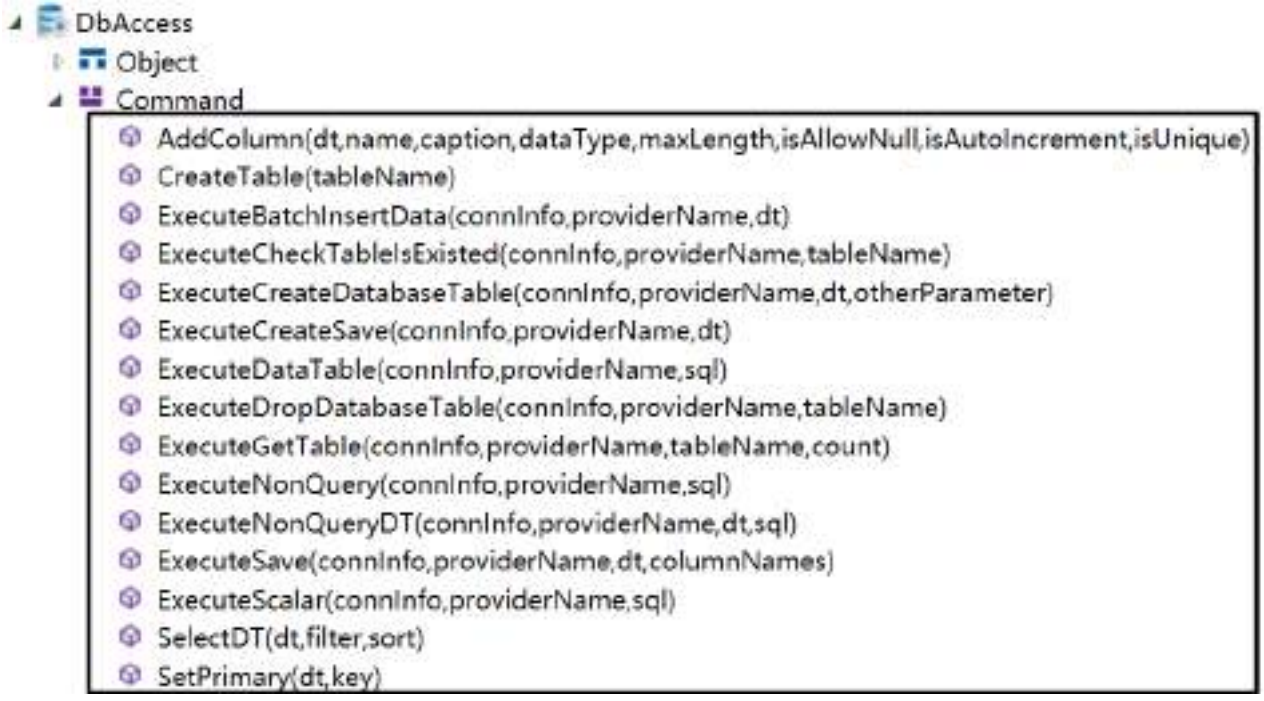
17.3 Accessing database

DIAView provides many methods to access external database, one of the most basic method is to call the database access command to access the database. Database access provides a rich script commands to help user query, modify the database data.

The command provided by database access which has been configured is as shown in the figure below:



Database access function also provides public commands to be convenient for the user to operate more flexibly.



18. Global

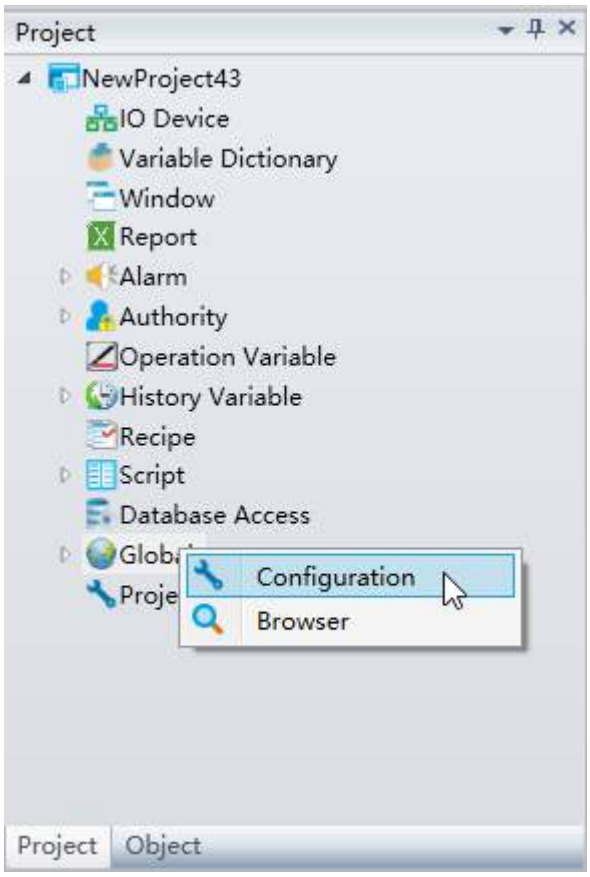
18.1 Overview

Globalization provides users with the ability to convert the contents of functional modules into the list of multi-language resources at development time and edit the extracted resources, so as to realize the function of multi-language free switching in the runtime screen.

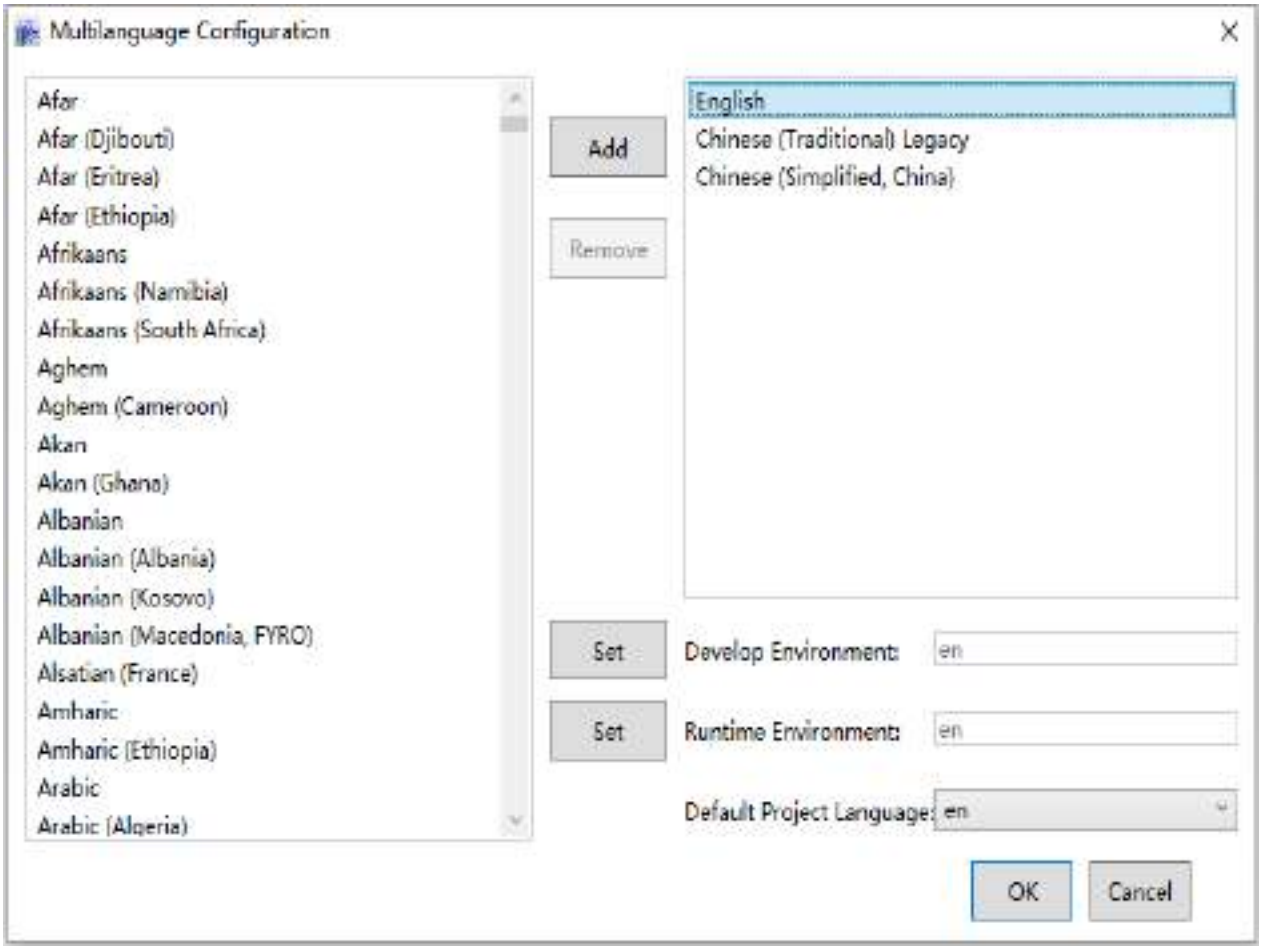
Currently supported multilingual resource types are: string image sound

18.2 Language configuration

Step 1: Right-click on the“Global”node in the project window tree index and then click the “Configuration” item in the right-click menu, as shown in the figure below:



Step 2: Open configuration window, as shown in the figure below:



- **Add:** Add selected language in the left list.

- **Remove:** Remove selected language in the right list, if select language is set to "Develop Environment Language","Runtime Environment Language" or "Default Project Language", can not be removed.

- **Set Develop Environment:** Set selected language in the right list to "Develop Environment Language", "Develop Environment Language" is the dispaly language of string, picture and soud in windows at develop environment.

- **Set Runtime Environment:** Set selected language in the right list to "Runtime Environment Language", "Runtime Environment Language" is the dispaly language of string, picture and soud in windows at runtime environment.

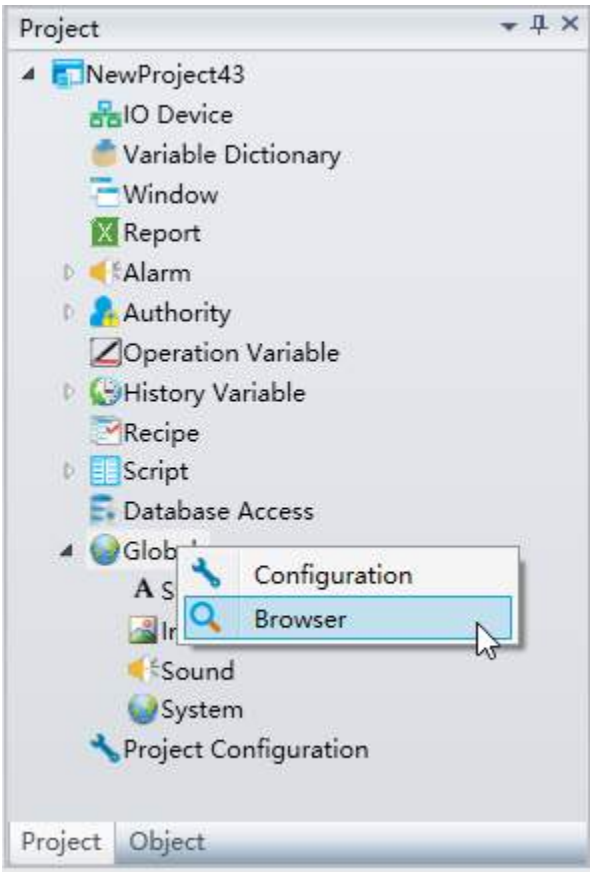
- **Set Default Project Language:** Select language from combobox to "Default Project Language", "Default Project Language" support simple Chinese, traditional Chinese and English. Window will display default project language if current language not translate. All resource should be translated if set langguage to "Default Project Language". When project default language not compares with software language, suggest switch project default language.

Note: it is recommended to switch the default language to be consistent with the software language when the software language is inconsistent with the default language of the project (for example, the software language is English and the default language of the project is Chinese)

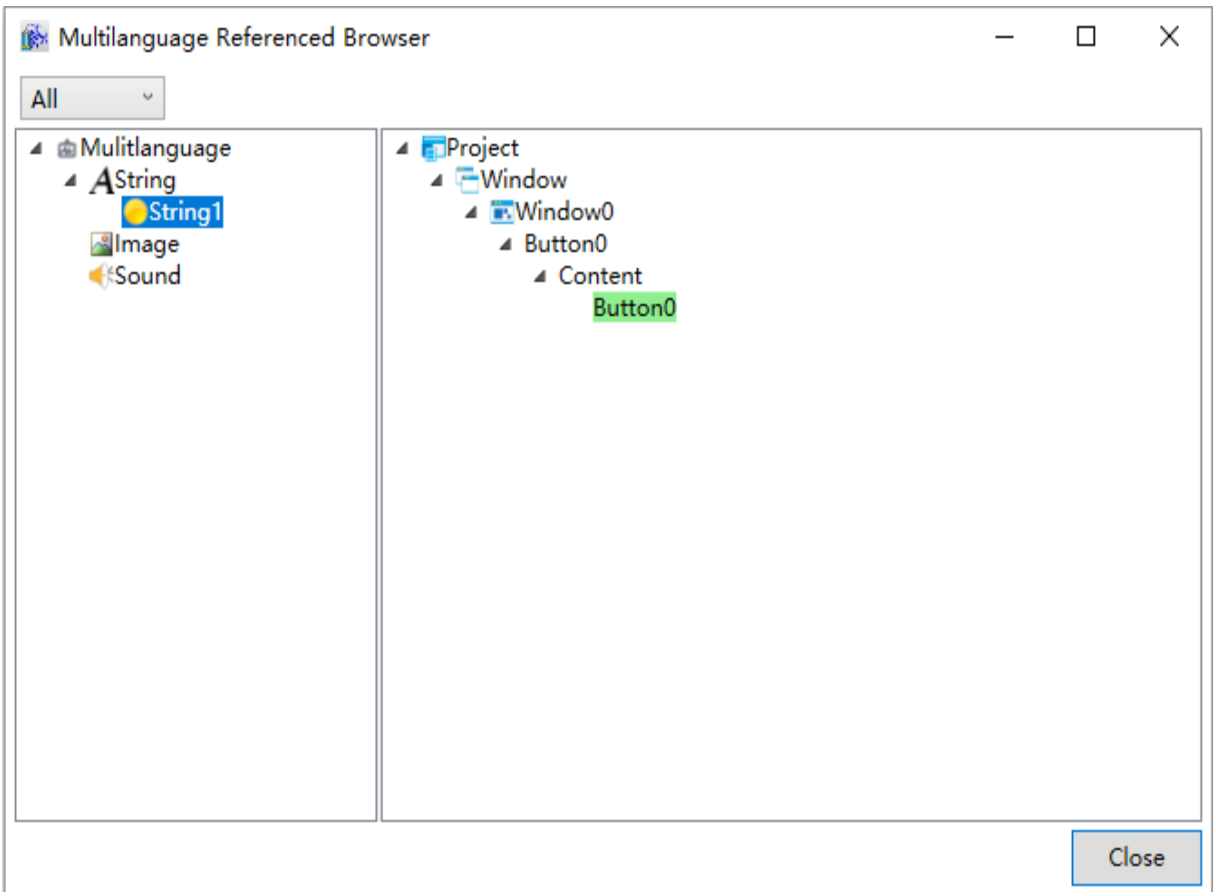
18.3 Language reference

Language referener is a window to view all string images and sound references in globalization to see if a resource is referenced and path information is referenced

Step 1: Right-click on the “Global” node in the project window tree index and then click the “Browser” item in the right-click menu, as shown in the figure below:



Step 2: Open borwser window, as shown in the figure below:



Description

- The left side of the language resource usage browser is the tree index of the "Global", and the right side is the display window of language resource references.

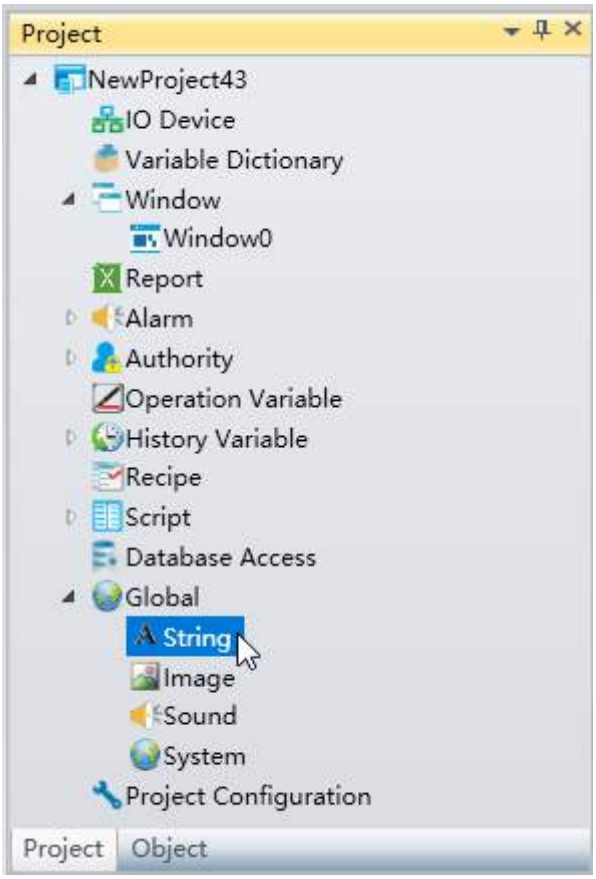
- In the tree index to the left, if the round symbol in front of the language resource displays as “yellow”,it means that this language resource is referenced. If it is “gray”, it means that it is not yet referenced.

- Select the language resource to view from the tree index to the left, and the language resource reference display window on the right will display all objects in the project that uses that language resource. They are displayed using an expanded tree index in order for users to pinpoint the objects that actually uses this language resource.

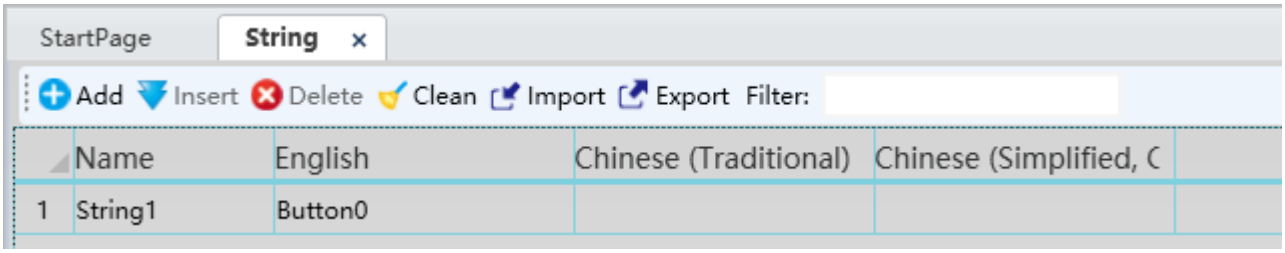
18.4 String

String is the global function of this type of repository, the user can customize the editing steps below:

Step 1: Select the“String” node in the project window tree index, as shown in the figure below:



Step 2: Double click "String" node and open editor window, as shown in the figure below:



Note: In string window, row represents resource display in different language, column represents a kind of language, if cell content is null, will display default language content.

Functional modules that can be converted into string resources in DIAView include:

- All tooltip property of controls in "Window".

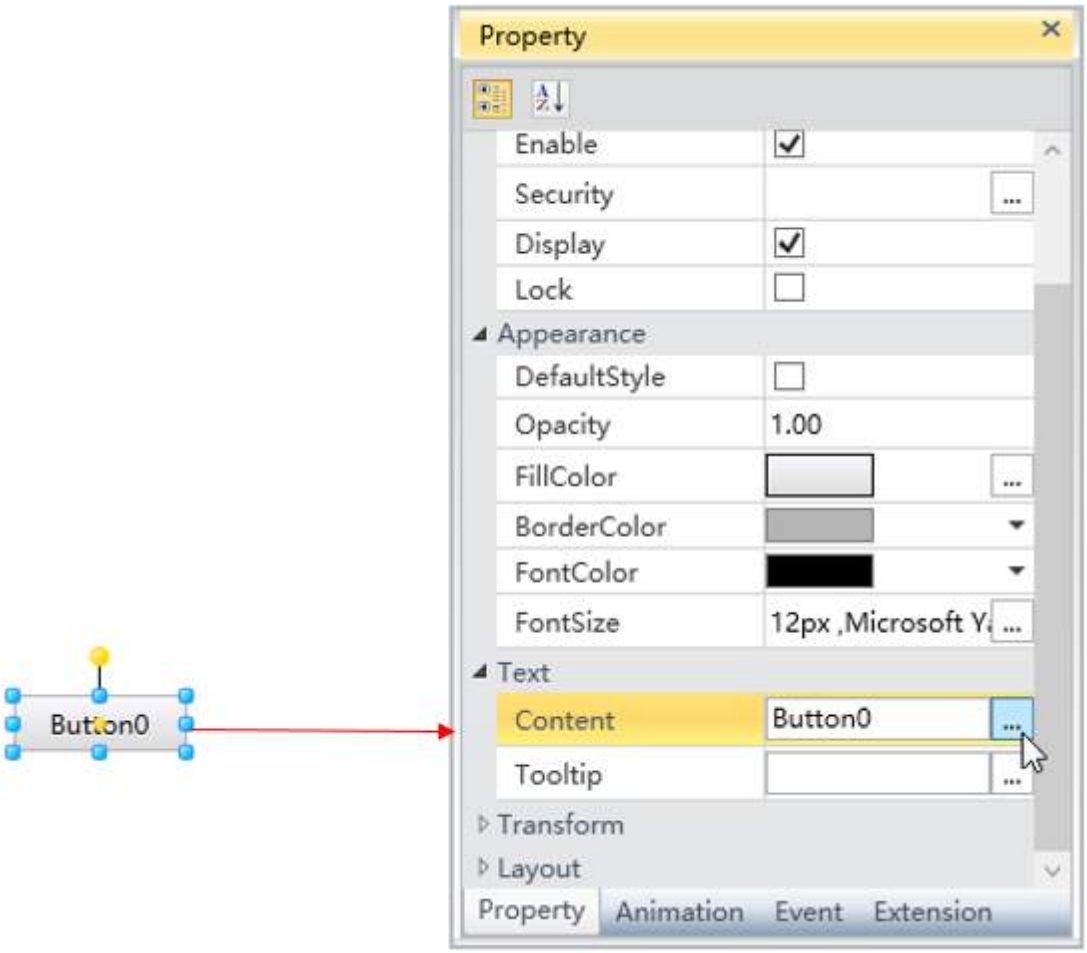
- Content property of text control, button control, label control, checkBox control, comboBox control and textBox control.

- Collection property of comboBox.
- Title and x, y title of chart controls.
- Alarm content in alarm variable.

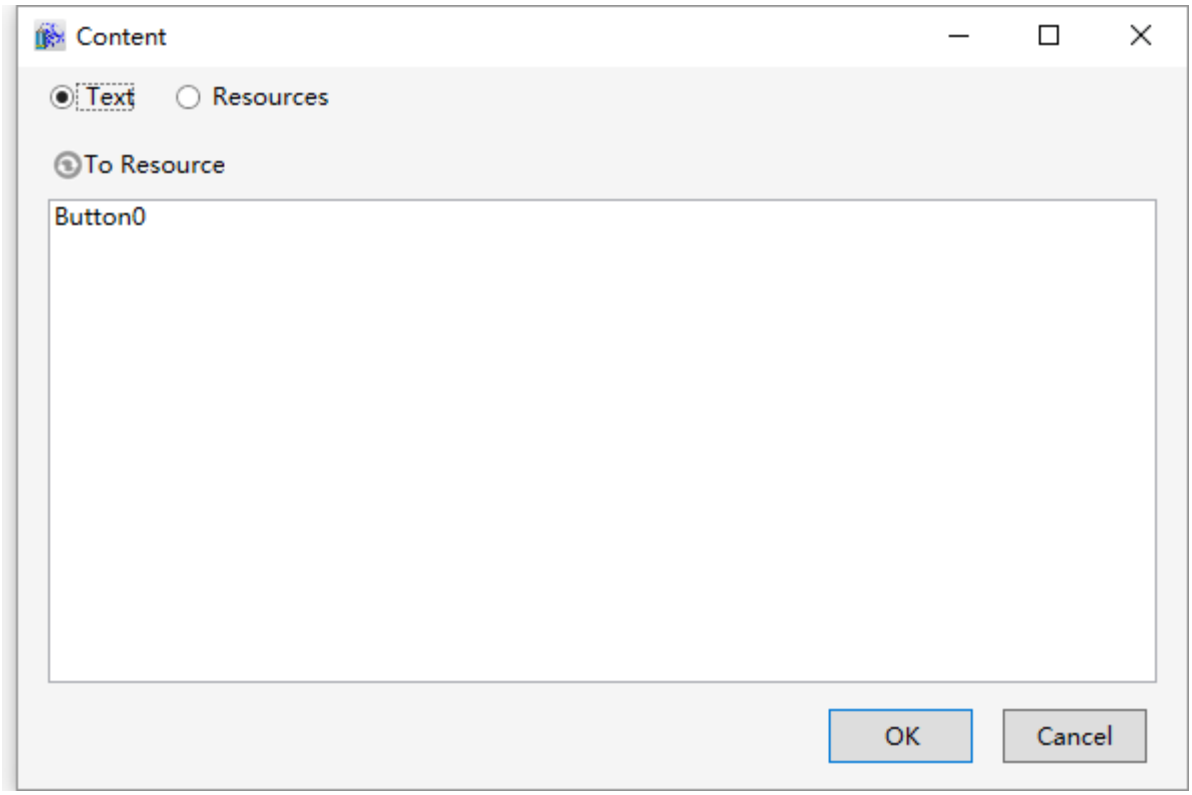
➢ **The text property selects the resource**

At development time, the text property can directly select an existing string resource, as follows:

Step 1: Add a new button control in window, click button in the content property, as shown in the figure below:

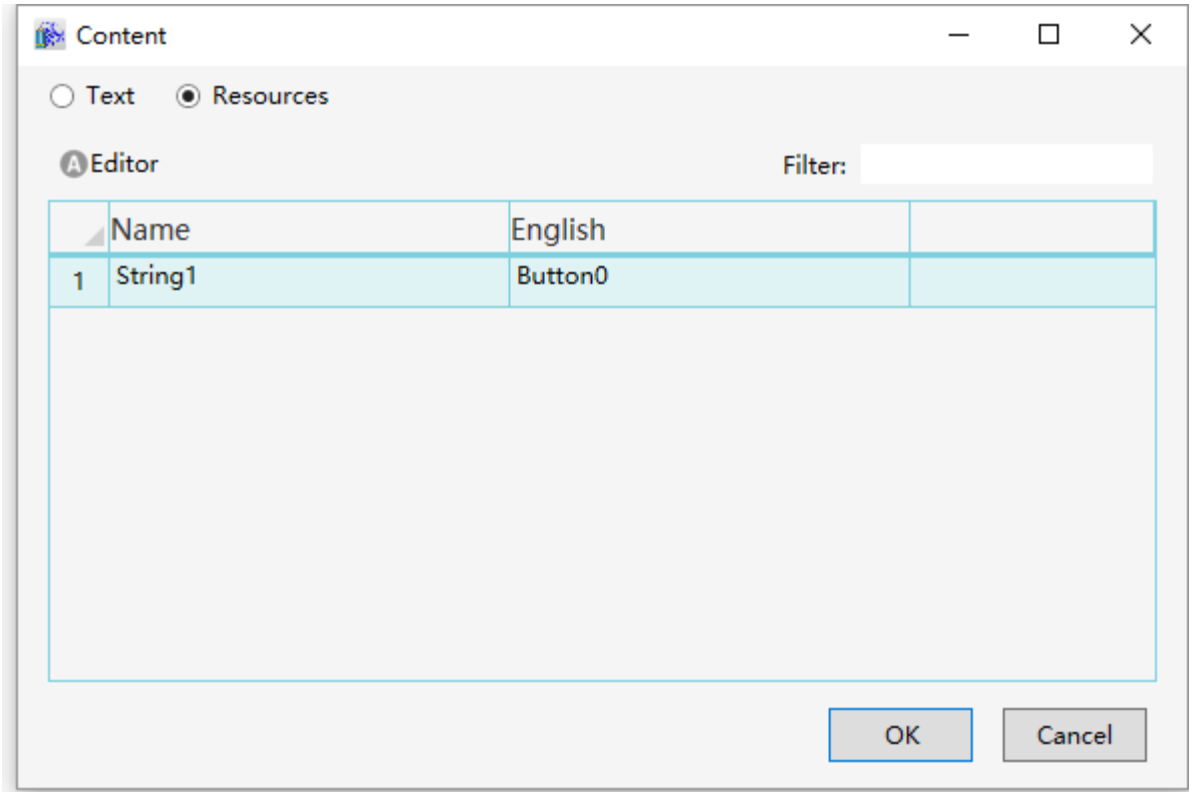


Step 2: Open content editor, as shown in the figure below:



Note: If user has create a resource in string window, cilck "Resources" and select it, if has not create, user can click"To Resource", system will generate a new resource in string window.

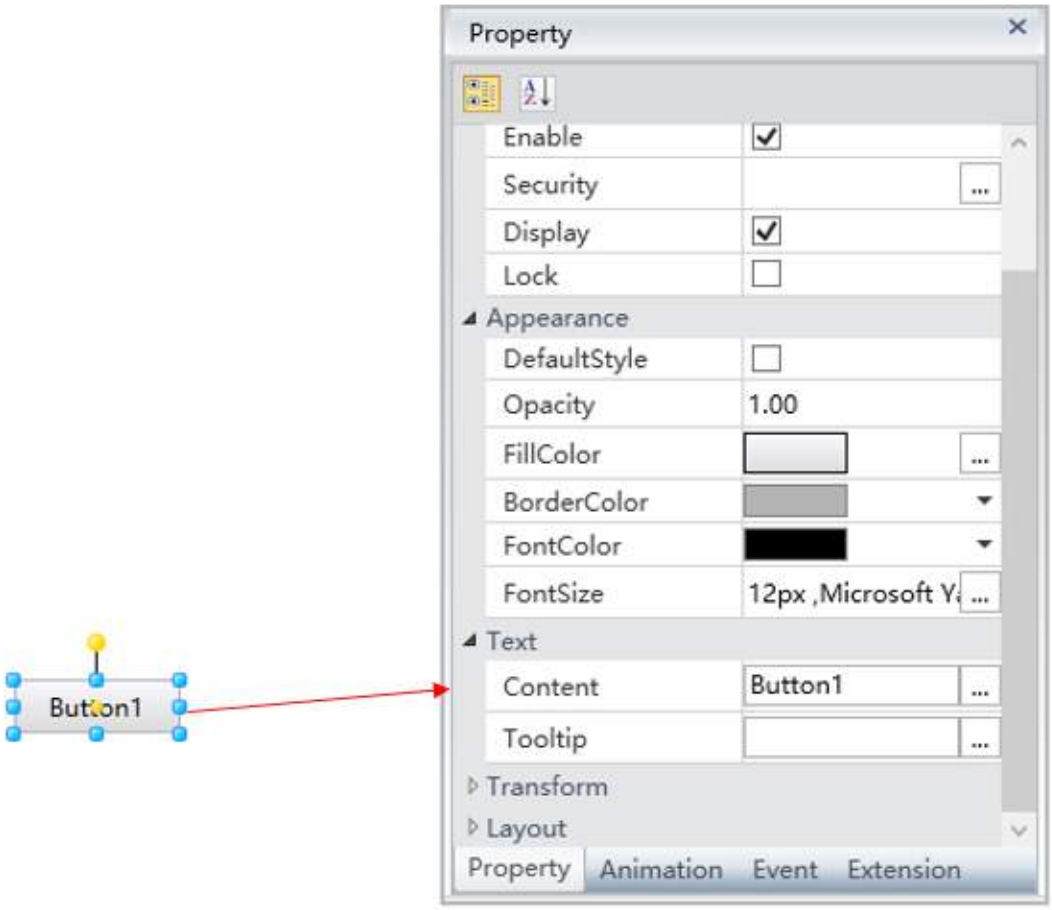
Step 3: Click "Resources", select a resource user want relation, and click "OK", this property has globalized, as shown in the figure below:



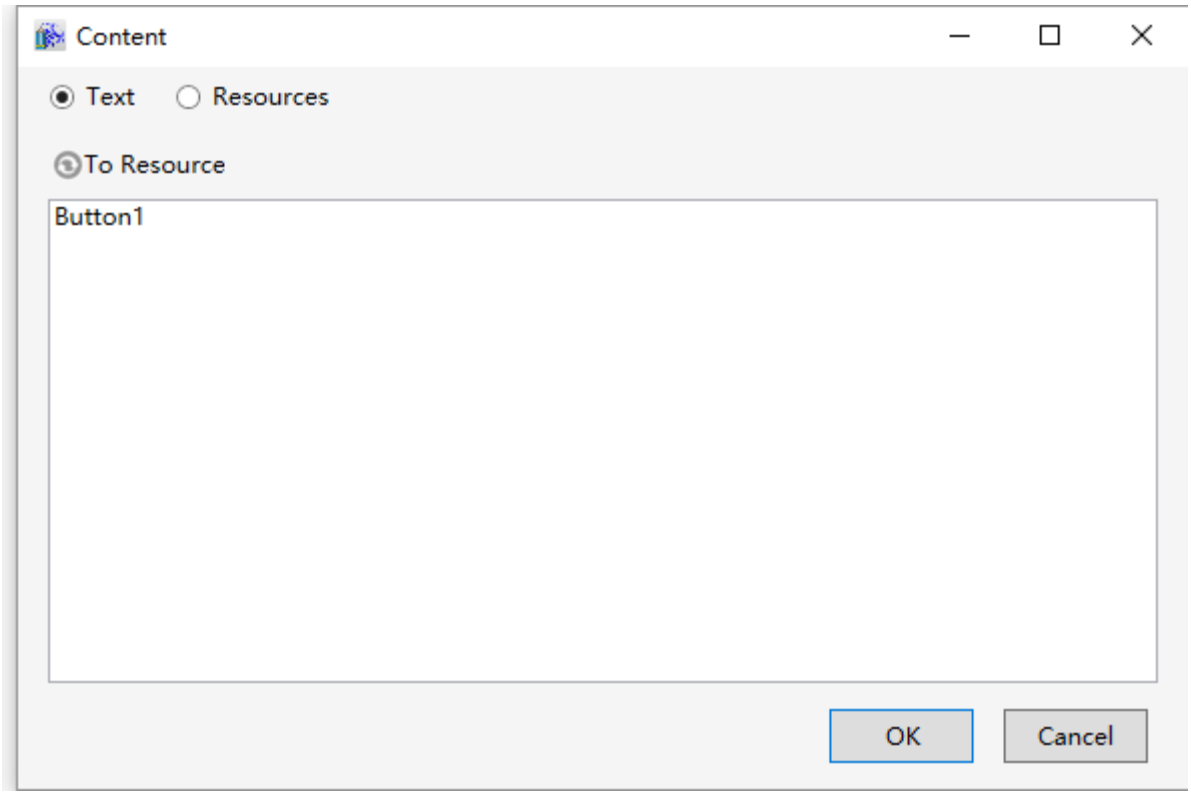
➢ **Text properties are converted to resources**

At development time, text properties can be converted to string resources, as follows

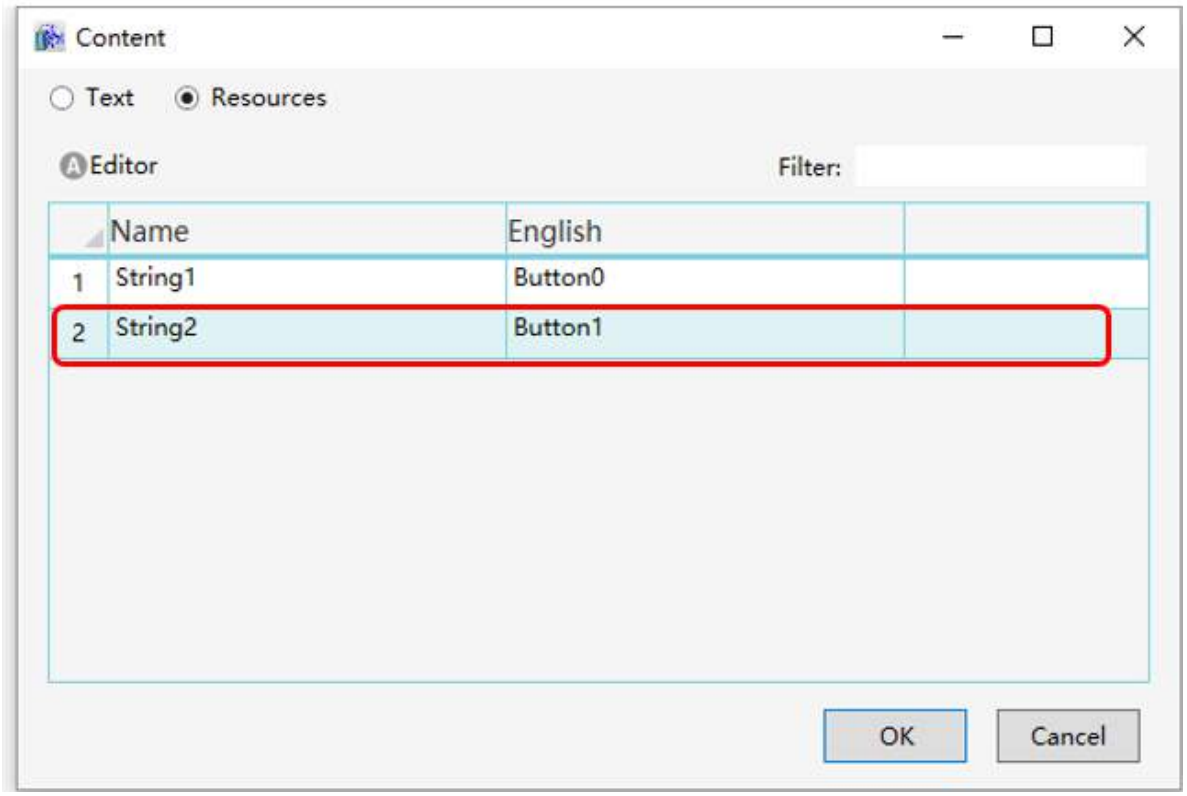
Step 1: Add a new button control in window, click button in the content property, as shown in the figure below:



Step 2: Open content editor, as shown in the figure below:



Step 3: Click "To Resource", select current content to generate resource, as shown in the figure below:



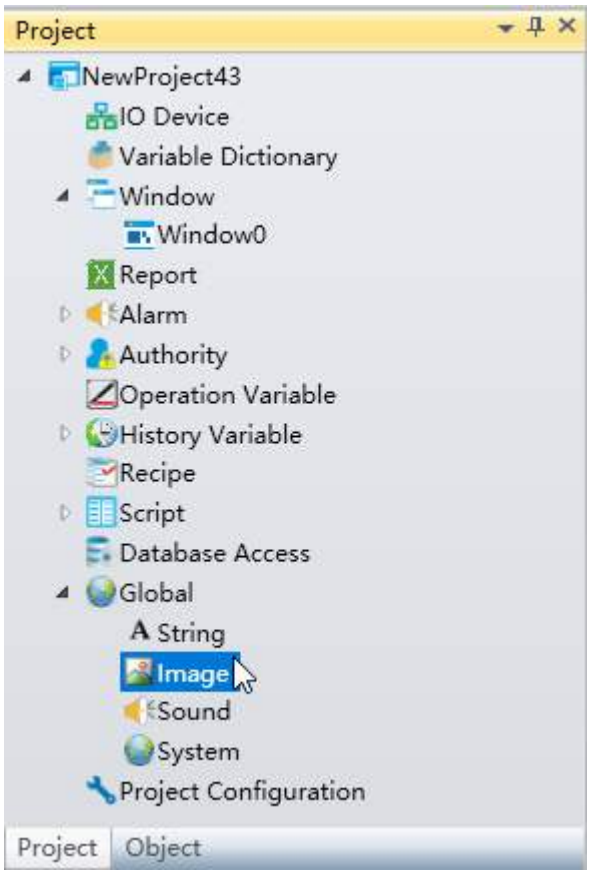
Step 4: Click "Editor",open resources editor and editor resource, as shown in the figure below:

Step 5: Close resources editor, click "OK", this property has globalized.

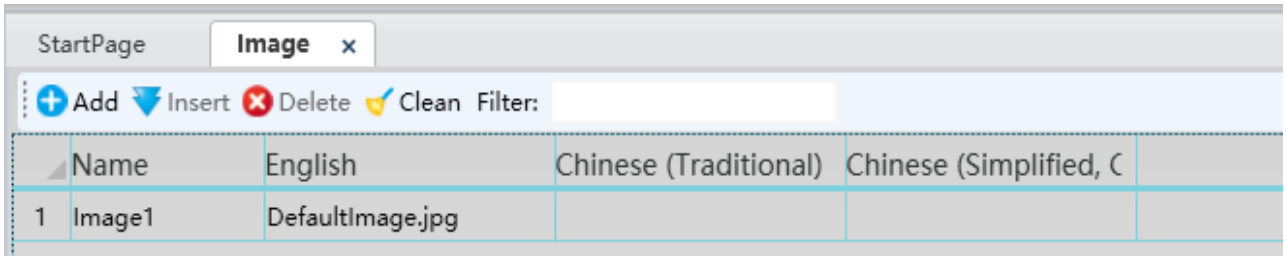
18.5 Image

Picture is the resource library of picture path in the globalization function. Users can customize it as follows

Step 1: Select the "Image" node in the project window tree index, as shown in the figure below:



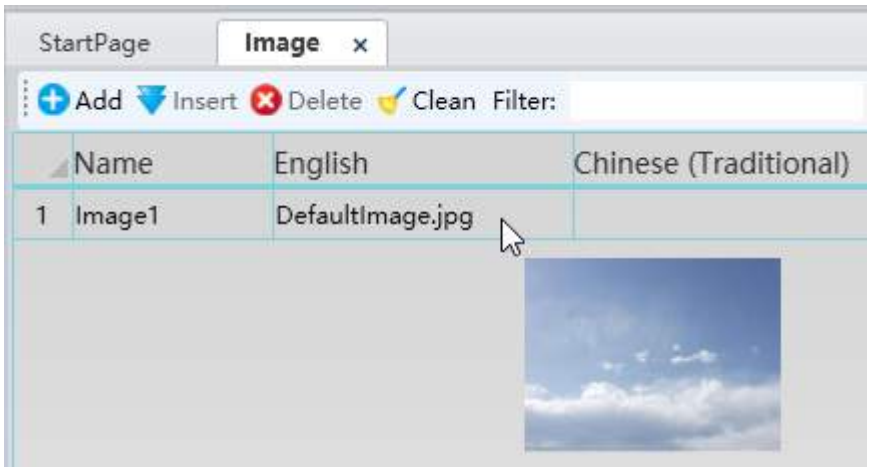
Step 2: Double click "Image" node and open editor window, as shown in the figure below:



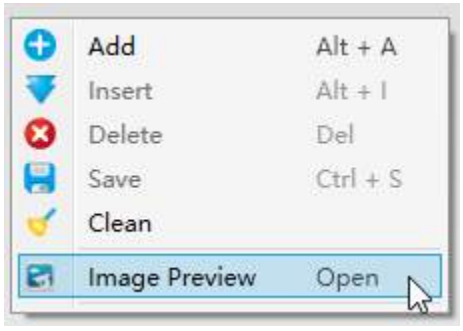
Note: In image window, row represents resource display in different language, column represents a kind of language, if cell content is null, will display default language content.

➢ **Image preview:**

User can preview image when mouse move into cell, as shown in the figure below:



User can close preview image at right menu, as shown in the figure below:



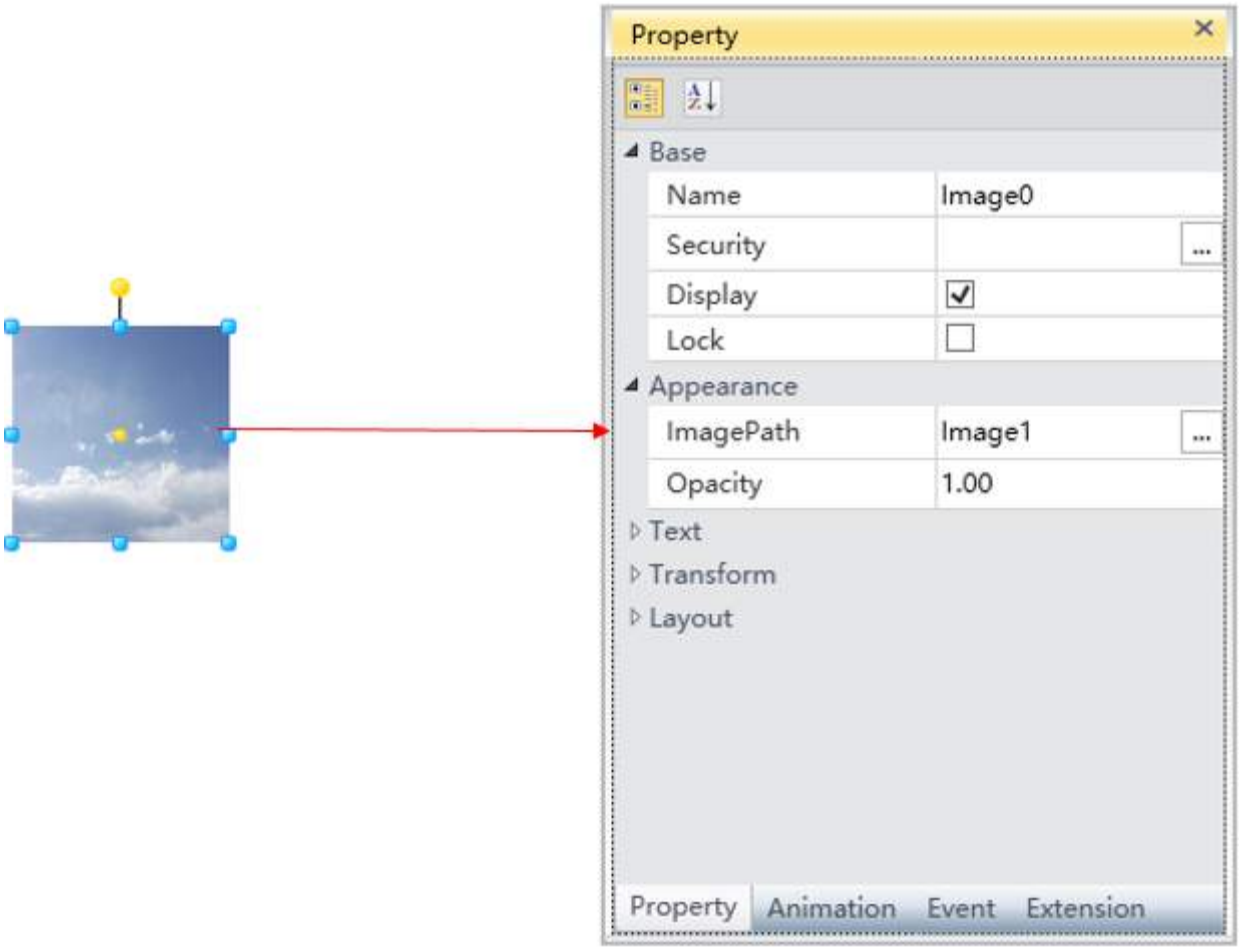
Globalization control of image in DIAView:

- Image control
- GIFImage control

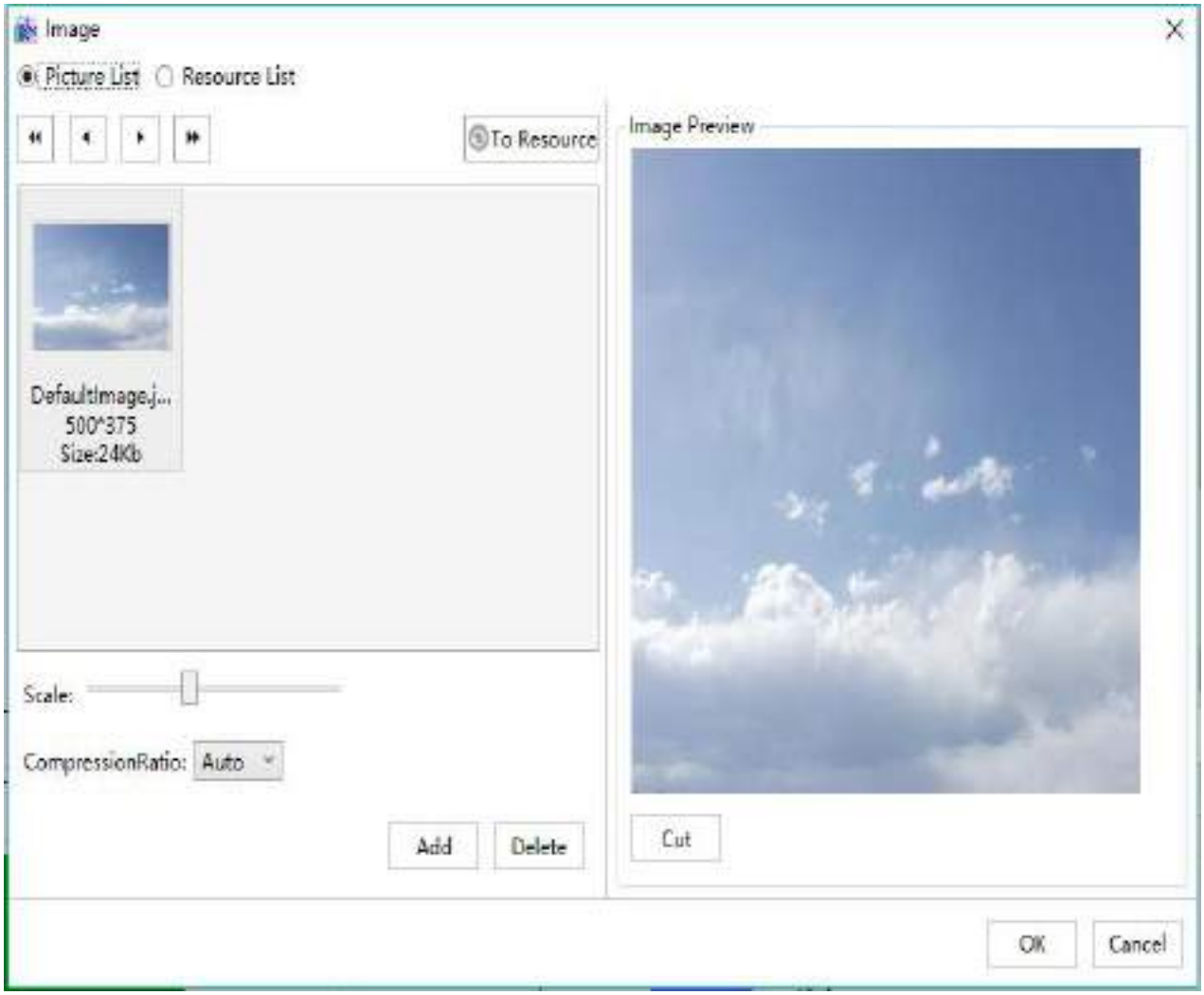
➢ **Use image Resource**

At development time, the image path can directly select the existing image resources, the specific operation is as follows:

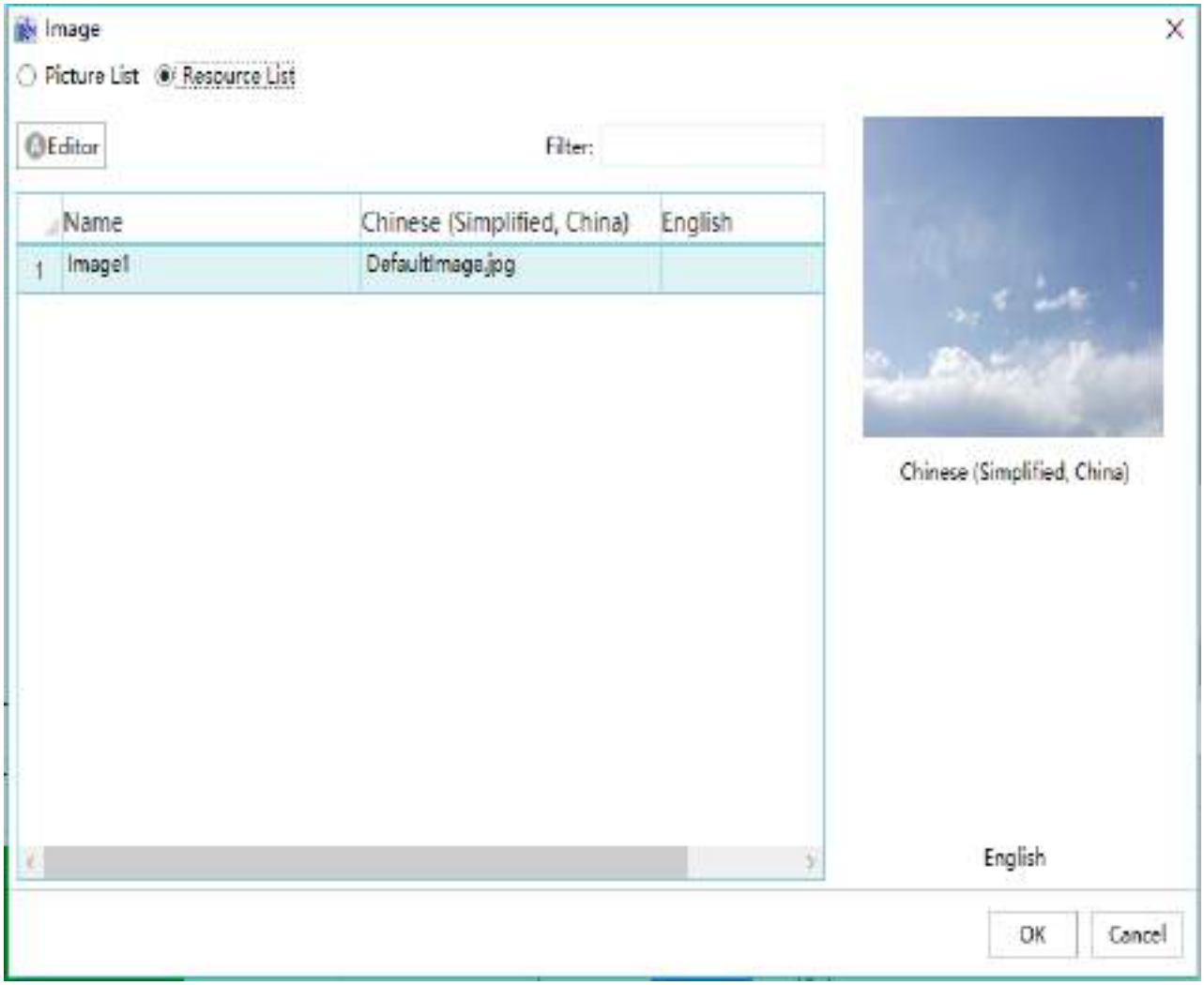
Step 1: Add a new image control in window, click button in the imagePath property, as shown in the figure below:



Step 2: Open image window, as shown in the figure below:



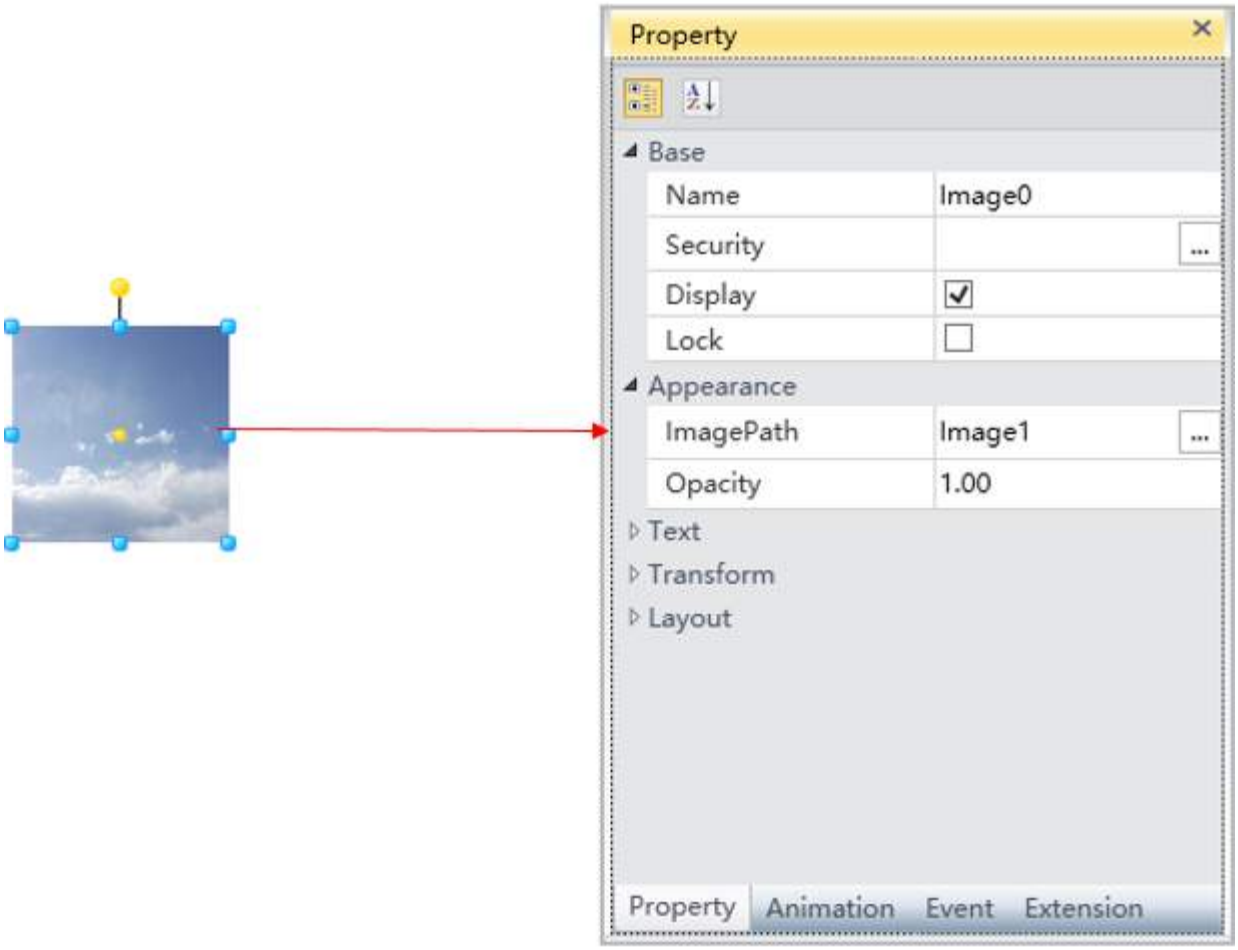
Step 3: Click "Resource List", select resource to relate, as shown in the figure below:



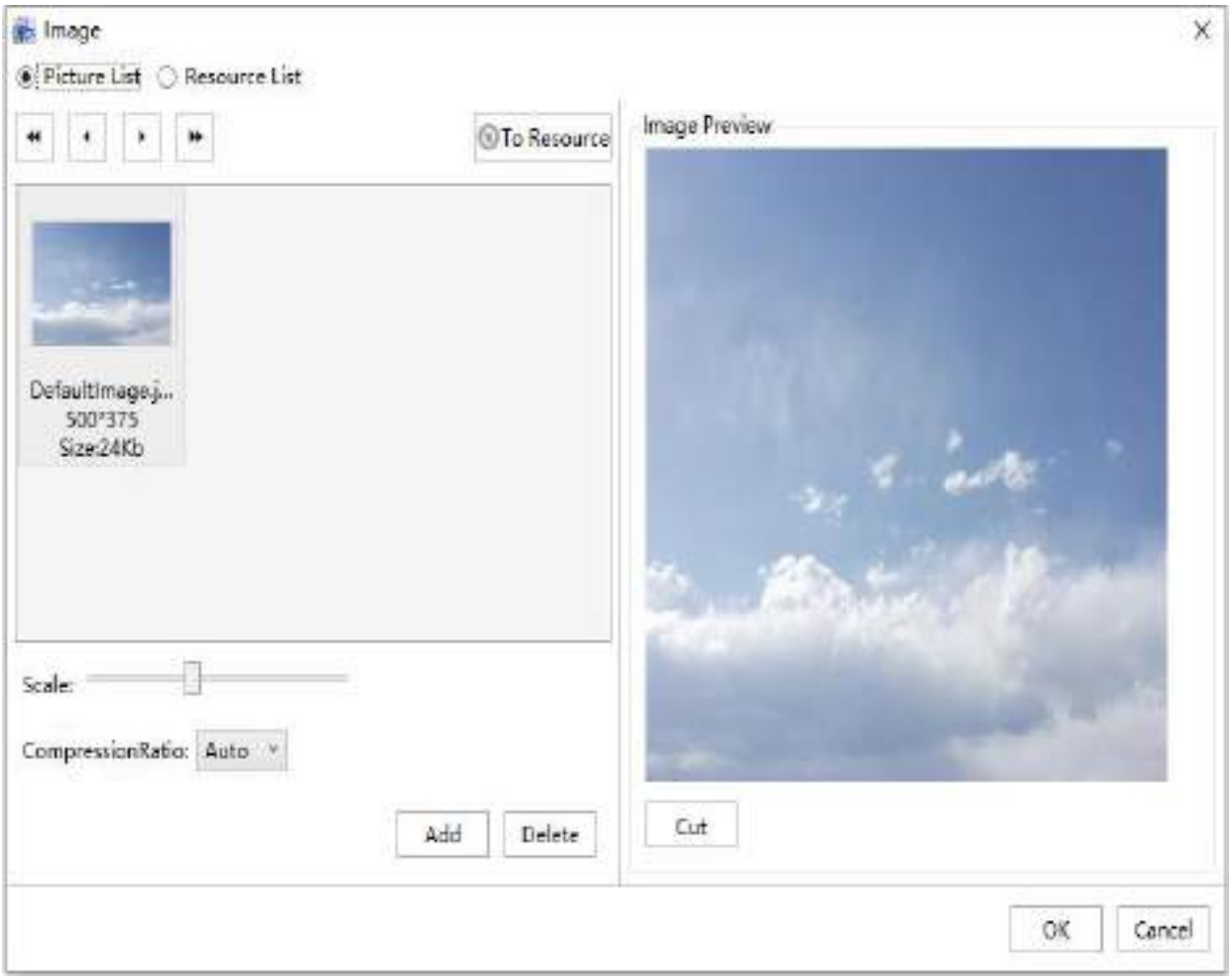
➢ **Image properties are converted to resources**

At development time, the image path can be converted to image path resources, as follows:

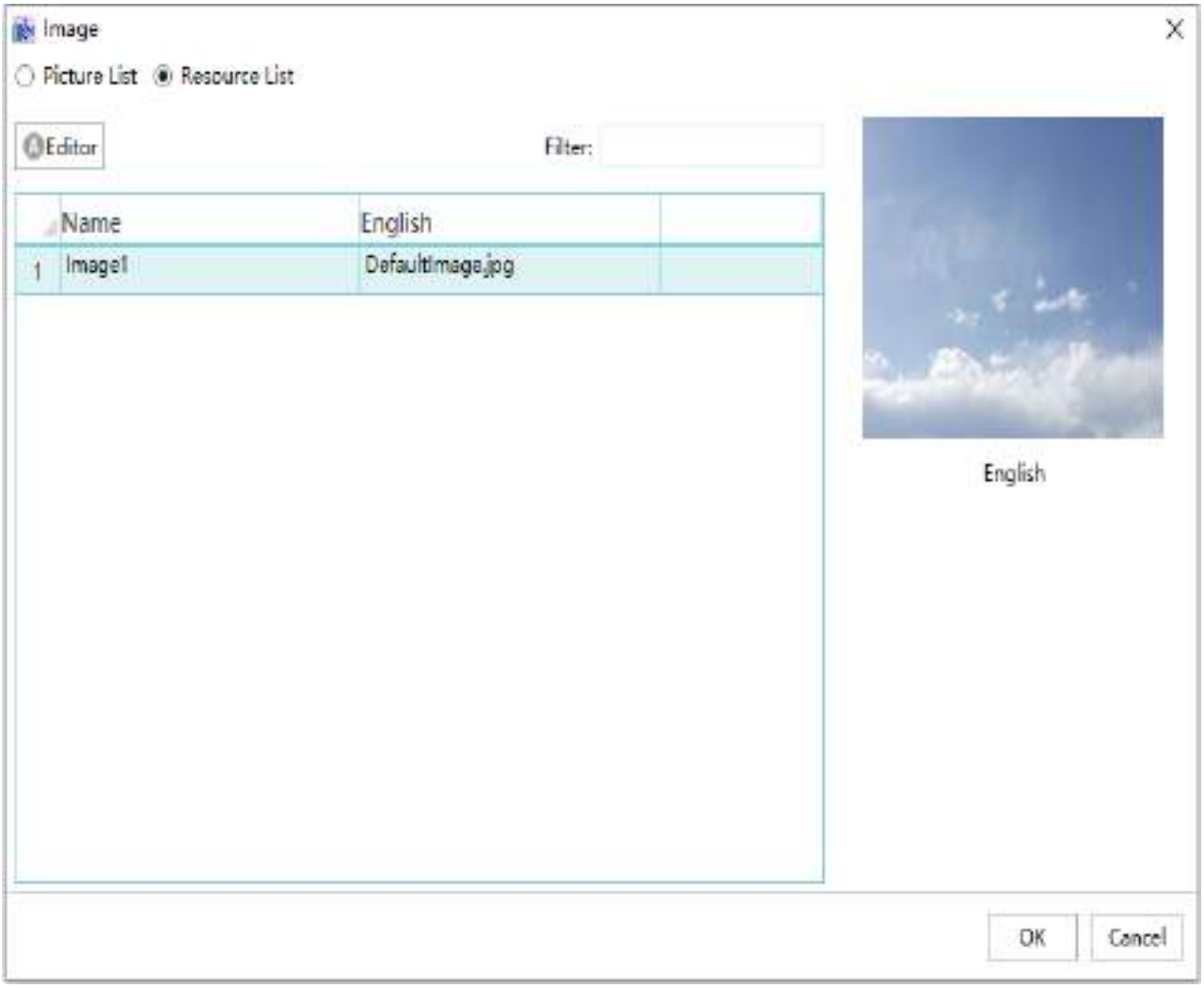
Step 1: Add a new image control in window, click button in the imagePath property, as shown in the figure below:



Step 2: Open image window, as shown in the figure below:



Step 3: Click "To Resource", select resource to relate, as shown in the figure below:



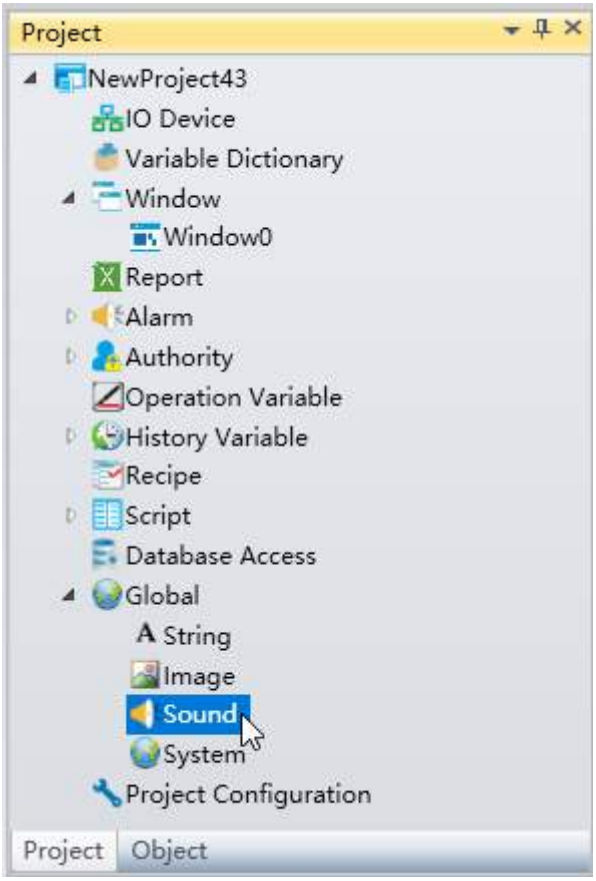
Step 4: Click "Editor",open image resources editor and editor resource, as shown in the figure below:

Step 5: Close image resources editor, click "OK", this image control has globalized.

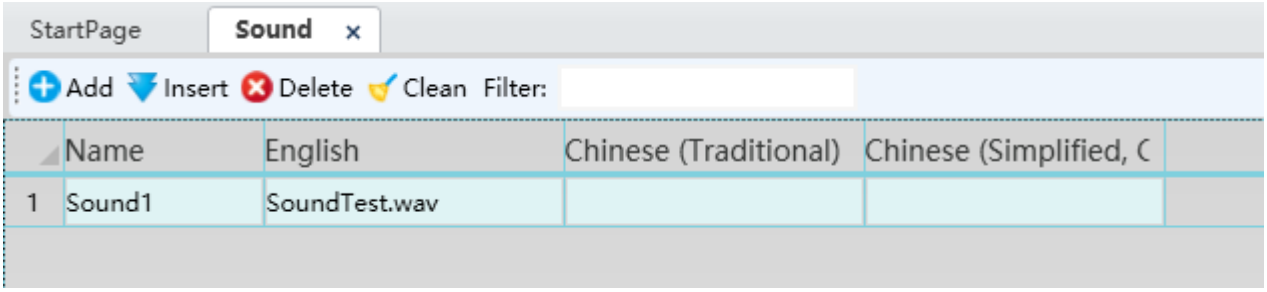
18.6 Sound

Sound is the sound resource library in the globalization function. Users can customize it as follows

Step 1: Select the“Sound”node in the project window tree index, as shown in the figure below:



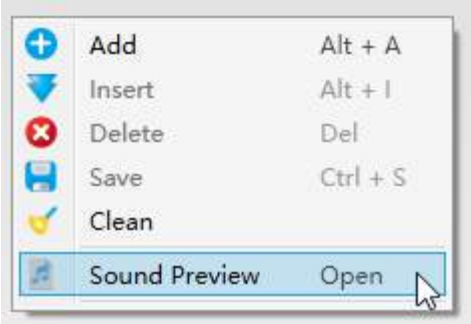
Step 2: Double click "Sound" node and open editor window, as shown in the figure below:



Note: In sound window, row represents resource display in different language, column represents a kind of language, if cell content is null, will display default language content.

Sound preview:

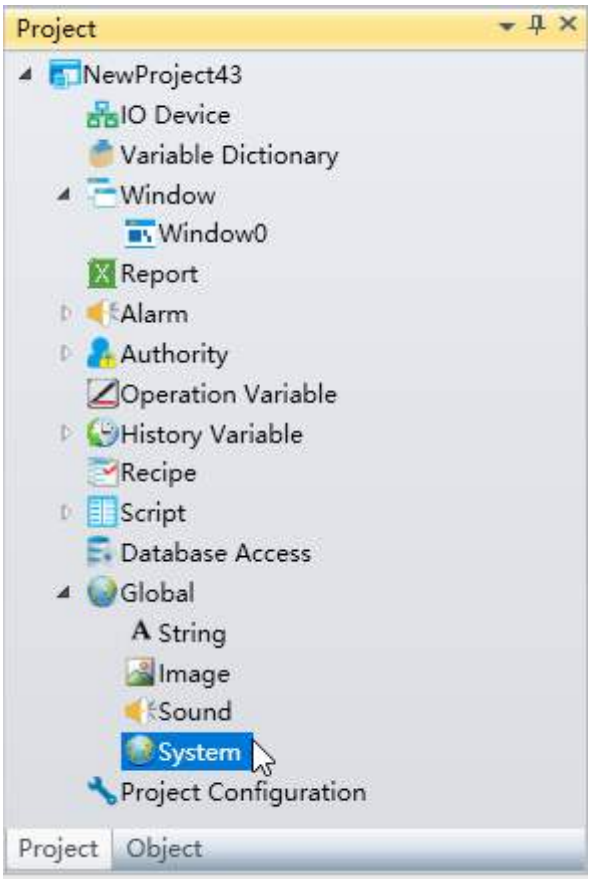
User can preview sound when mouse move into cell and user can also close preview image at right menu, as shown in the figure below:



18.7 System

System more language refers to the window in the system, according to the text in the specified language system multilingual including advanced control menu bar and a drop-down box content, runtime menu bar etc. For example: system multilingual edit box has three languages: English and Chinese traditional, is the project in the development time and runtime can switch between the three languages

Step 1: Select the "System" node in the project window tree index, as shown in the figure below:



Step 2: Double click "System" node and open editor window, as shown in the figure below:

StartPage | System ×

Import Export Filter:

	Name	English	Chinese (Traditional)	Chinese (Simplified, C
1	HMI_WebBrowse	Be Ready	就緒	就绪
2	HMI_WebBrowse	Finish	完成	完成
3	HMI_WebBrowse	Response.....	響應.....	响应.....
4	HMI_WebBrowse	Address	網址	地址
5	HMI_WebBrowse	GoBack	後退	后退
6	HMI_WebBrowse	GoForward	前進	前进
7	HMI_WebBrowse	GoSearch	轉到	转到
8	HMI_WebBrowse	HomePage	首頁	主页
9	HMI_WebBrowse	Open	打開	打开
10	HMI_WebBrowse	Refresh	刷新	刷新
11	HMI_WebBrowse	Stop	停止	停止
12	HMI_WebBrowse	GoBack	後退	后退

Description

● A line in a system's multilingual interface represents what a resource displays in different languages, and a column represents a language

● Users of traditional Chinese and English are not allowed to modify it, even if they export Excel changes and then import them

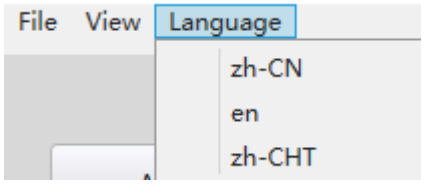
● Users can not customize the multi-language resources of the system, and can only translate the corresponding national language for the existing resources

18.8 Switch language at runtime

When the system is running, users can switch between multiple languages in the following ways:

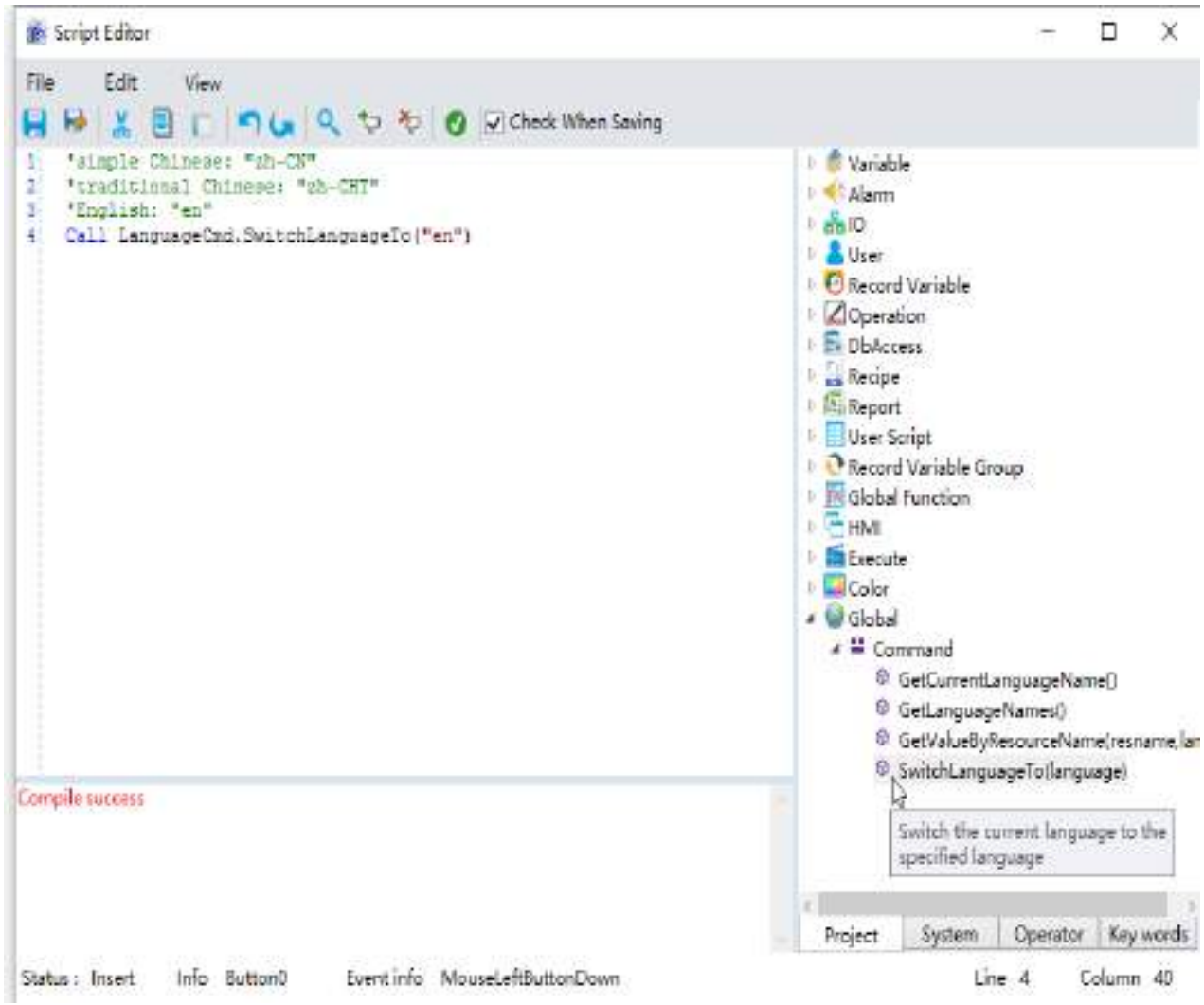
➢ **By runtime meun:**

At run time, use the shortcut key F11 to exit the full screen, and click the language button in the menu bar to switch languages, as shown in the figure below：



➢ **By script:**

Switch language by "SwitchLanguageTo" script at global script, as shown in the figure below:



19. Project Configuration

19.1 Overview

In the DIAView, the main function of project configuration is to configure the database of the runtime environment, the shortcut keys of the runtime environment and the project runtime picure.

Configure the database to connect external databases, and the DIAView can be allowed to perform data interaction with common commercial databases(Compact, SQL Server, Oracle)； The system variable,system alarm,system events are saved into the database to achieve the data sharing between the DIAView and the user's ERP or MES systems etc.

Shortcut key configuration is mainly to set keys that can be used in the project runtime environment.

Picture configuration is to set the window to open and the display order of the window in the project runtime environment.

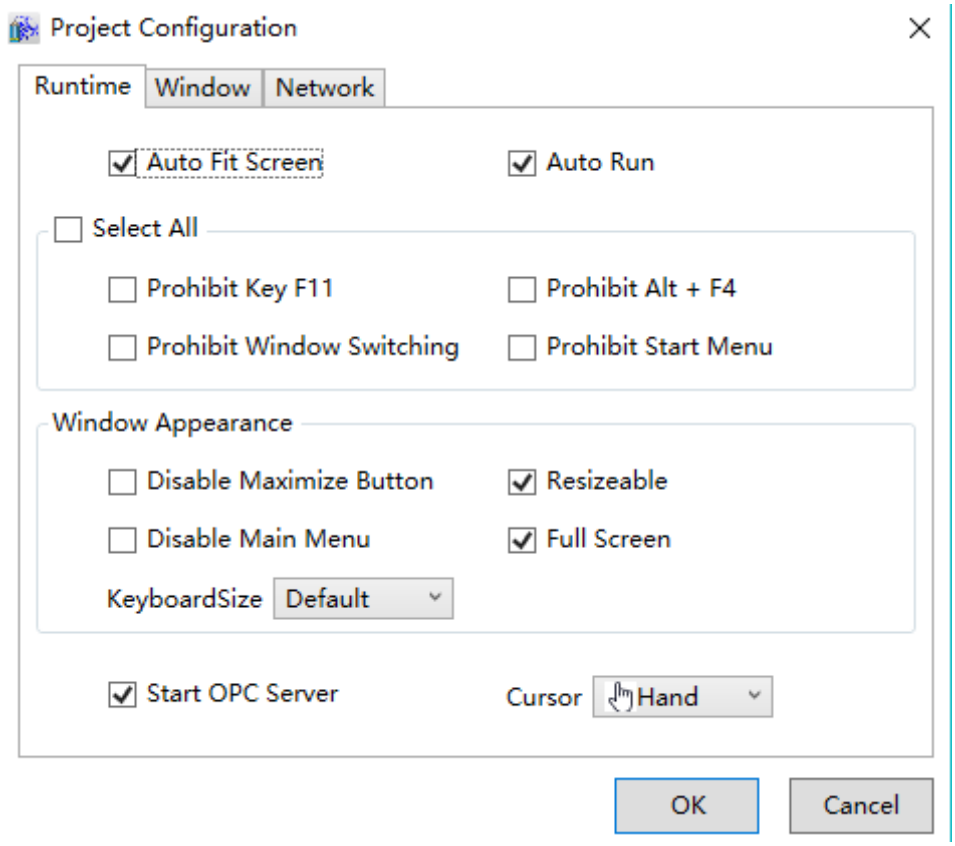
19.2 Project configuration

- **Database**

It is mainly used to configure related configuration files that connect with the external databases，used to old version of "RecordVariable". Now it removes in right_click menu of "RecordVariable(compatible)" node, go to 14.2 to see the details.

- **Runtime**

Sets the shortcut keys in the runtime environment, as shown in the figure below:



✧ **Auto Fit Screen:** The runtime window size will adapt automatically to the full screen.

✧ **Auto Run:** The default project is automatically run when the computer is started.

✧ **Select All:** Once selected, all checkbox in the group box will be selected.

✧ **Prohibit key F11:** In the runtime environment , prohibit F11 key to switch to full screen mode.

✧ **Prohibit Alt + F4:** In the runtime environment , prohibit “Alt ＋ F4” keys to close the engineering.

✧ **Prohibit Window Switching:** In the runtime environment , prohibit “Alt ＋ Tab”/“ Windows key ＋ Tab” keys to switch windows.

✧ **Prohibit Start Menu:** In the runtime environment , prohibit “Windows” key to open the start menu.

✧ **Disable the maximize button:** In the runtime environment , the maximize button is not available.

✧ **Resizeable:** In the runtime environment ,the interface size is allowed to change.

✧ **Disable Main Menu:** In the runtime environment ,the system menu will not be visible.

✧ **Full Screen:** In the runtime environment ,the screen is full screen display.

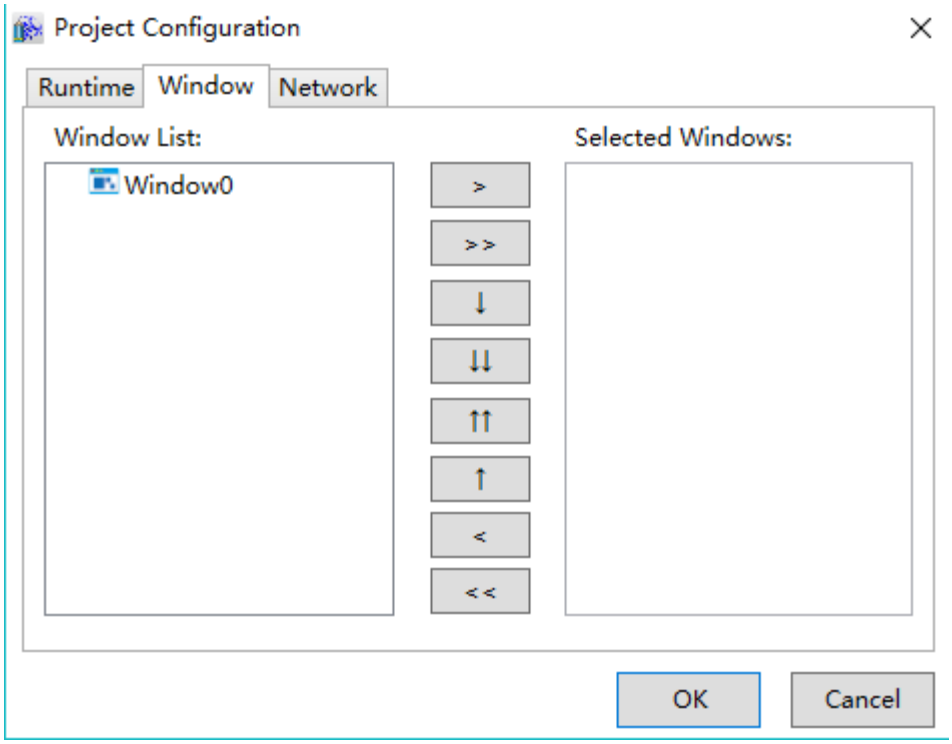
✧ **KeyboardSize:** Sets the style of the keyboard that pops up at run time.

✧ **Start OPC Server:** Sets whether to start OPC Server at run time.

✧ **Cursor:** Sets the running mouse style.

➢ **Screen**

Sets the windows to be opened and the display order of the windows in the runtime environment, as shown in the figure below:

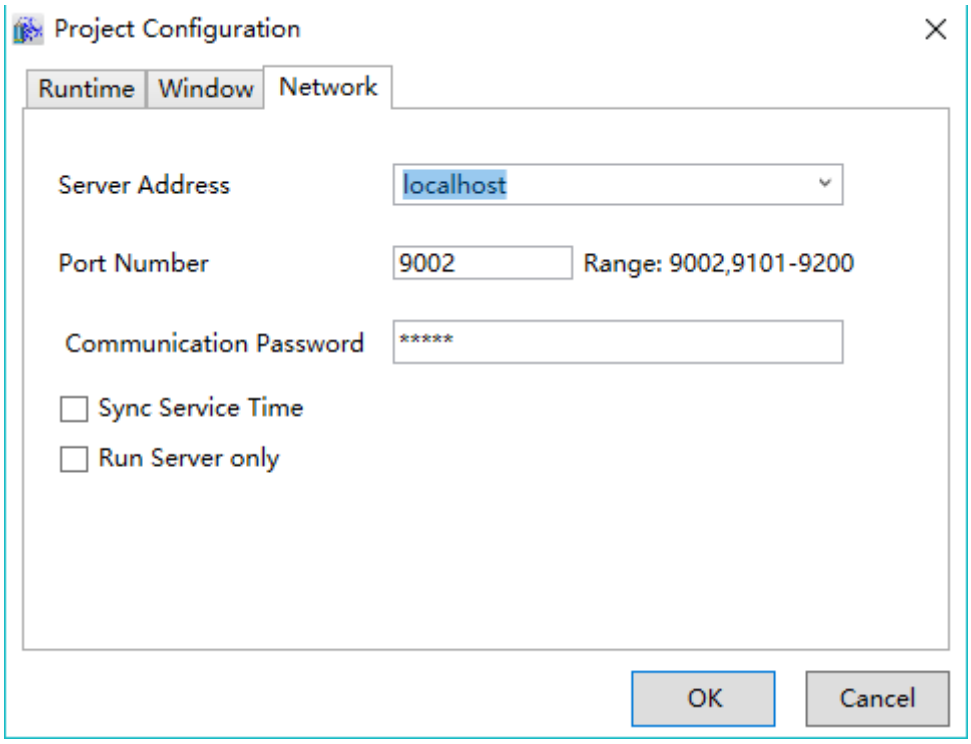


Select the windows that need to be opened in the project runtime environment from the “Screen list” field, and then click the > button to add the selected windows into the “Selected windows” field; press the >> button to add all windows into the “Selected window” field. The initial windows can be one or multiple windows, the newly added window is located at the bottom; if there are multiple windows, the ↑ button can be used to move the screen up and the ↓ button can be used to move the screen down.

Window display order: The windows in the “Selected windows” list will be opened in the order from top to bottom; the window opened last (which is the bottom-most window in the list) will be displayed at the front-most.

- **Network**

Set the IP address or computer name, port number and other related information of the server in the CS architecture,as shown in the figure below:



✧ **Server Address:** Enter the name or IP address of the server.

✧ **Port Number:** The recommended range is 9002,9101-9200.Use the port number of the recommended range to facilitate unified setting of the port number range allowed by the firewall.

✧ **Communication Password:** The password that the client needs to use to log in to the service.

✧ **Sync Service Time:** Whether the client is allowed to set the system time of the local machine based on the service time.

✧ **Run Server only:** By default,clicking run at development time will start both client and service,When you check run only services,only services are strarted.

20. Script

20.1 Overview

The DIAView functions are comprehensive and powerful, but customers' requirements are various; some functions are additional customization according to user needs.The DIAView can write ralated programs with the script editor to complete certain special tasks and functions.

Event configuration and user program etc. in the DIAView software all need the script editor to write

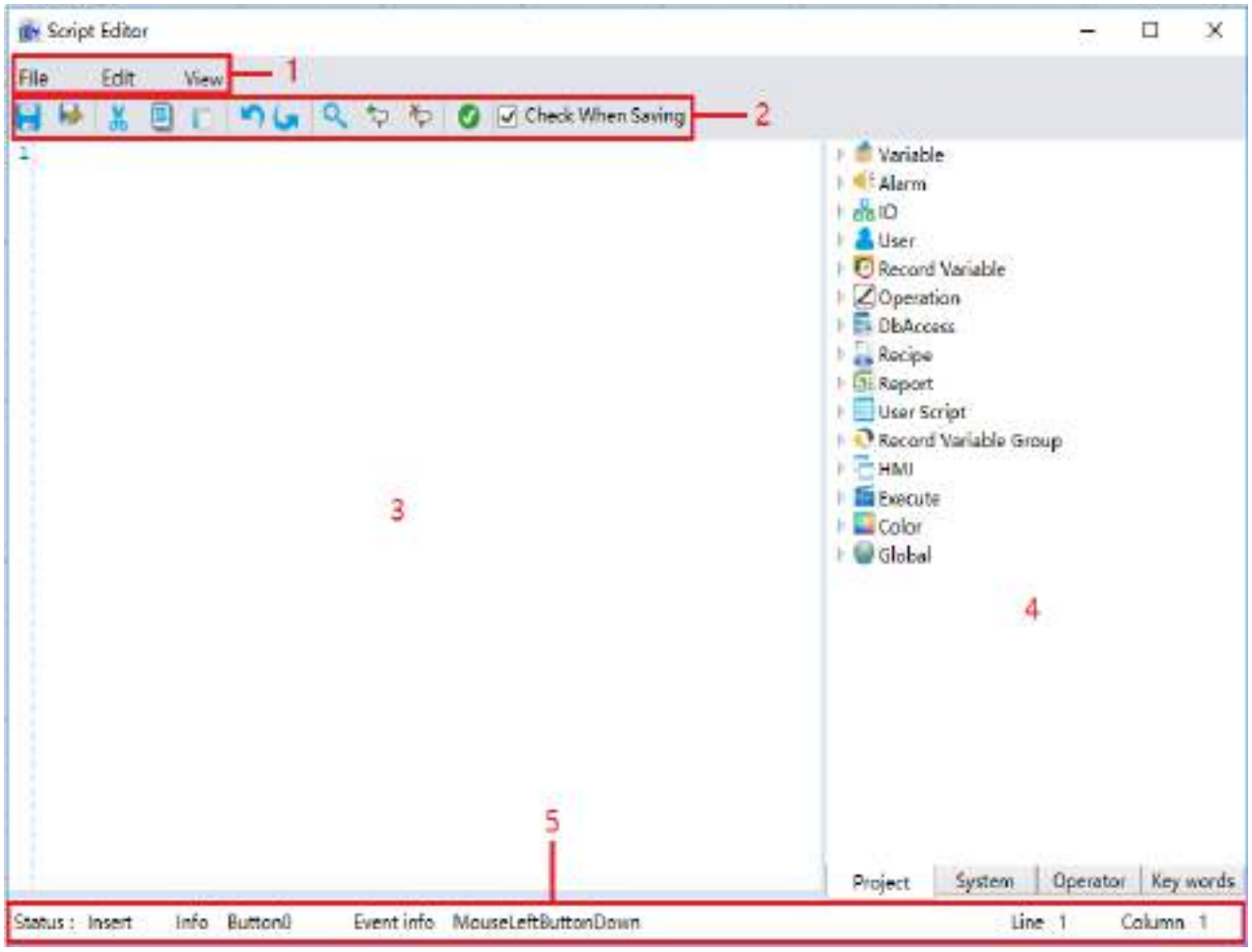
scripts. The DIAView uses the VBScript language; User can write logic control programs according to VBScript language specifications to complete special functions and enhance the availability of the system.

20.2 Script editor function introduction

Scripts writing is achieved in the script editor; in order to make it easier for users to write scripts correctly, the script editor not only provides functions such as intellisense, highlighting, one-key typesetting and precise positioning but also provides the syntax check function.

The script editor of the DIAView has basically the same interfaces, structures and functions that the common editors have. The following methods can be used to open the script editor.

Select “Window” in the directory tree of the project window in DIAView development environment→ right-click “Add window”→ draw a graphic in the window sketchpad → select the graphic and click a button of the event configuration field in the “Event” window, the script editor will popup, as shown in the figure:



Script editor components:

- ✧ **Menu bar:** Provides menus of basic functions of various operations.
- ✧ **Toolbar:**Operation function shortcut buttons.
- ✧ **Script editing area:**The area where the script program is written.
- ✧ **Project/system/operator/key words/directory tree:**Directory windows of various operations.
- ✧ **Message bar:**Displays the editing status and operation information etc.

1. Menu bar

The menu bar of the script editor includes the following menu and menu items:

(1)File menu

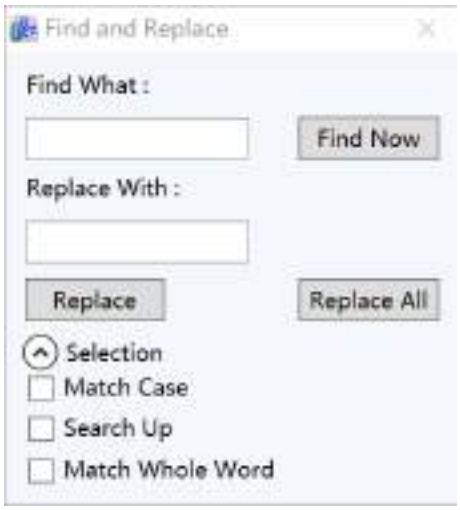
Command button	Function
 Import	Imports external script file.
 Export	Saves the current script content as an external script file.
 Save	Saves the current script content.
 Save and exit	Saves the current script content and exits the script editor.
 Check	Perform syntax check for the contents of the current script; the check results will be displayed in the “Output window”.
 Exit	Discards the edited script content,exits the script editor.

(2)Edit menu

Command button	Function
 Undo	Press this command to cancel previously executed commands and start from the last operation.

Restore	Click this command to recover the previous undo command and start from the last operation.
Cut	Select the text within the script editing area and click this command to delete the text and copy it into the clipboard.
Copy	Select the text within the script editing area and click this command to save the text and copy it into the clipboard.
Paste	Pastes the content in the current clipboard into the script editing area.
Delete	Select the text within the script editing area and click this command to delete the text.
Select All	Click this command and all text in the script editing interface will be selected.
Find and Replace	Click this command and a secondary menu will popup.It includes three commands: Find, replace and replace all. Please refer to below for details.
Scroll to line	Achieves quick positioning function; enter the row number in the popup dialog block, click “OK”, the system cursor will move to the end of the row and select that line of script.
Type Setting	Click this command and the text in the script editing area will automatically perform format arrangement.

(3) Find and replace:Click this command and the dialog block will popup as shown in the follow figure:



✧･ **Find:** Enter the text to search in this edit box.

✧･ **Match case:** If this item is selected, it will only display examples that have exactly the same cases as the string content in “Find target”.

✧･ **Search up:** Select this item and it will start searching upwards from the current location of the cursor.

✧･ **Match whole word:** Select this item and only examples that matches all words in the “Find target” string will be displayed. For example, find “aaa” and the result will return “aaa”; it will not return “baaa” or “aaac”.

✧･ **Replace:**First enter the search content and then enter the replace content, and then simply press the“Replace” button.

✧･ **Replace all:**Enter the search content and replace content, and then simply press the “Replace all” button.

The options of “Replace” and “Replace all” have the save effects as the “Find” option.

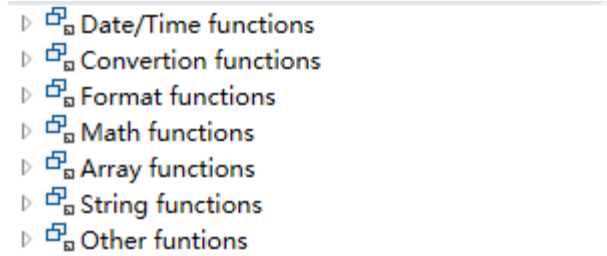
(4) View menu

Command	Function
 Font setting	Click this command, a secondary menu will appear to set the font style and font size.
 Show/Hide Output Window	Click this command ,the output window will be displayed or hided.

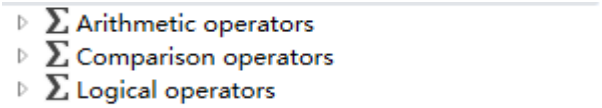
2.Project/system/operator/key word directory tree window

(1) Project tree: Includes most information of development project, including variable information, alarm information, communication information and user information etc.

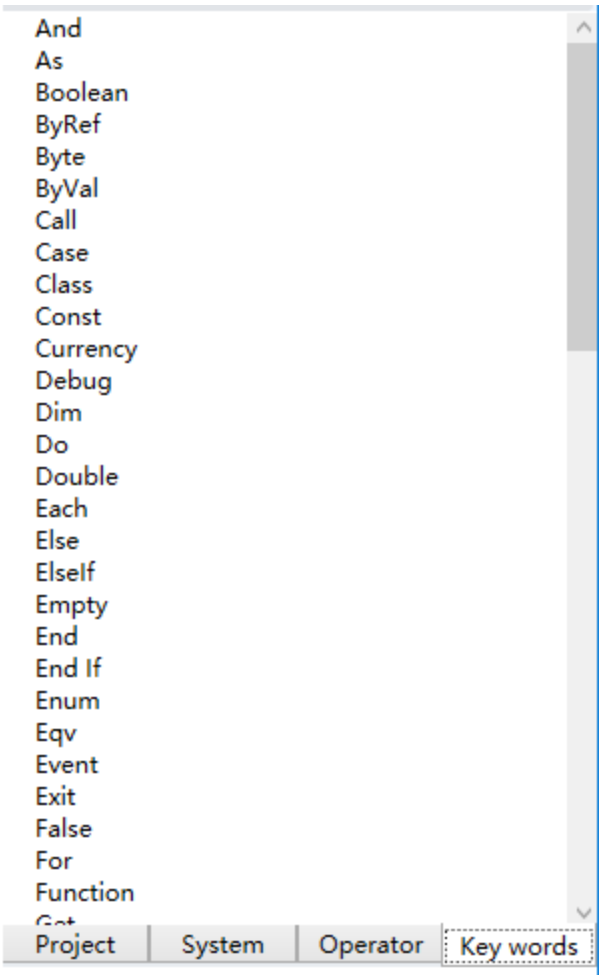
(2) System tree: There are 7 groups including date and time, conversion, format etc.; each group includes several functions.



(3) Operator tree: There are three groups including arithmetic operator, comparison operator and logical operator. There are several operators under every group; double-click the operator and the operator will be entered into the script editor interface.



(4) Key words tree: Includes all keywords of the script syntax. Double-click the keyword and it will automatically be entered into the script editor interface.



3.Syntax highlighting

In order to allow users to edit scripts more conveniently and clearly, the script editor will use different colors to distinguish different syntax parts. As shown in the table below:

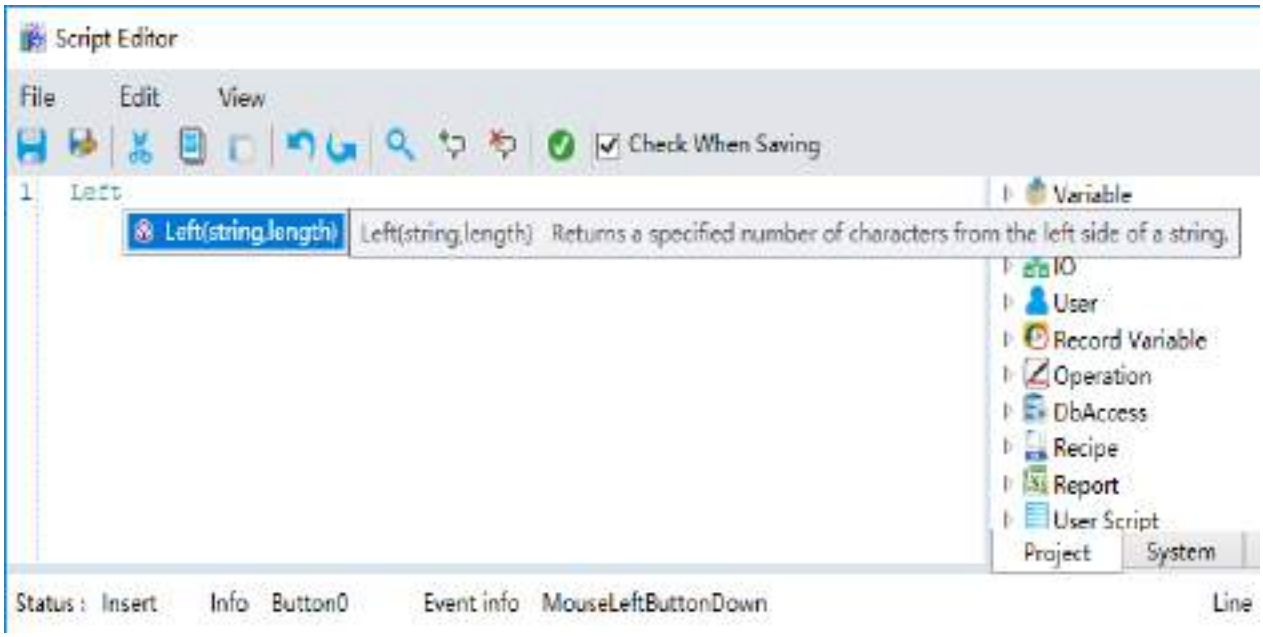
Content	Color
Default	Black
Number	Purple
String	Dark brown
Mathematical symbol	Red
Note	Green
Keyword	Blue

4. Intelligent reminder function

(1) Tip reminder

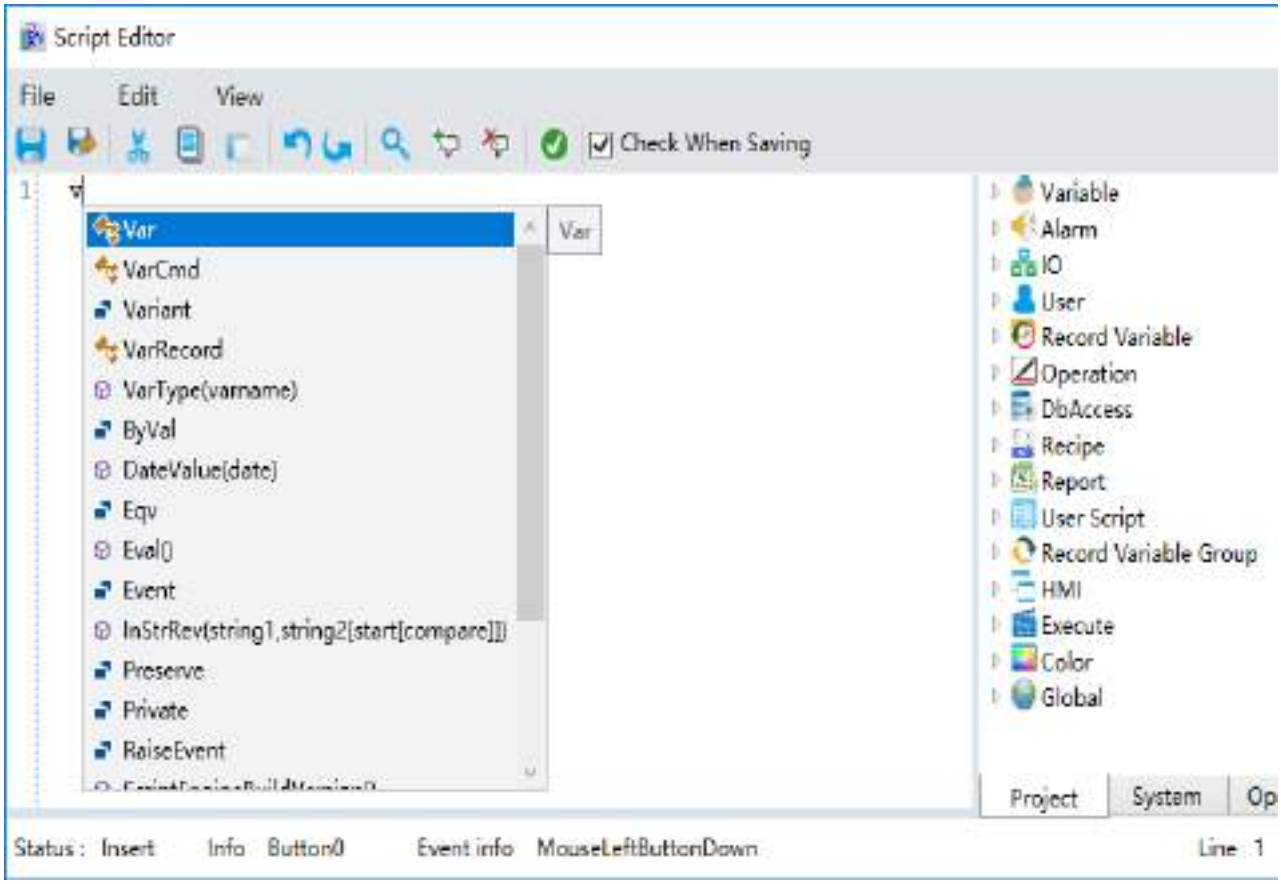
When the user enters a system function and enters "(", the automatic reminder frame will remind the function name and parameters of that function and also the description of the function. In the Tip reminder frame, the bold, black parameters represents the number of the parameter that is currently being entered.

As shown in the figure below:

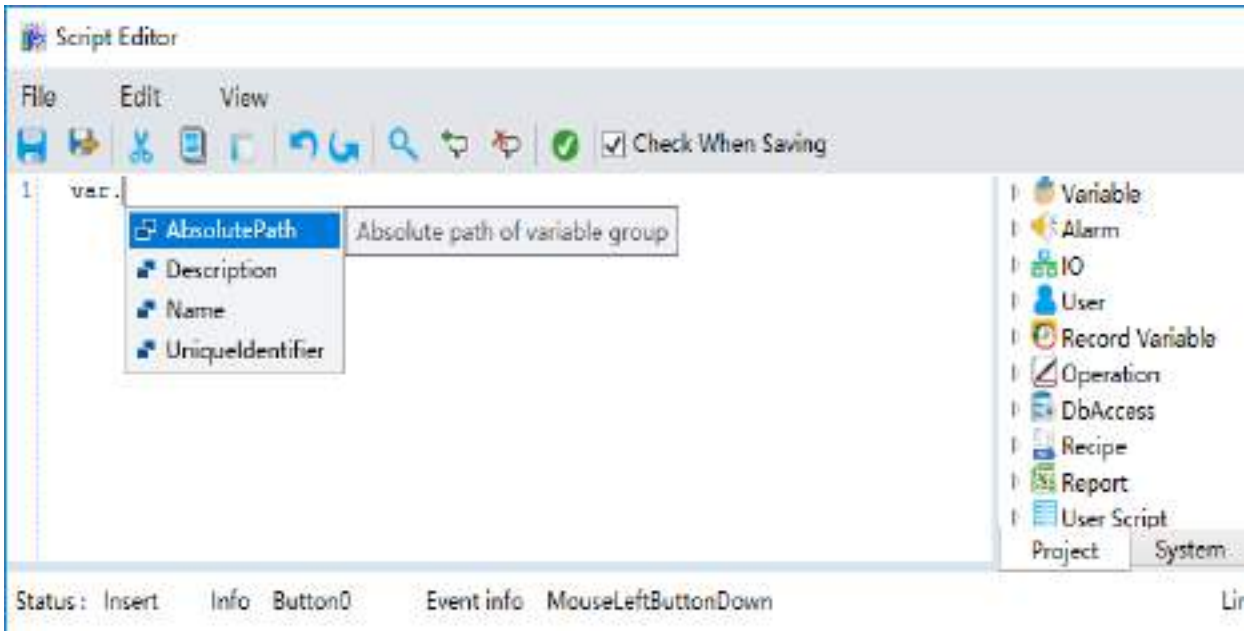


(2) List box prompt

When users enter a letter, the reminder frame will automatically display. This reminder frame lists the results of fuzzy matching letters, including objects, properties and methods etc. As shown in the figure below:



When the user enters ".", the system will display the reminder frame according to the content before the "."; the remind frame may include property and variable group etc. As shown in the figure below:



5.Shortcut keys

Available shortcut keys in the script editor are as follows:

Command	Shortcut keys
Copy	Ctrl＋C
Cut	Ctrl＋X
Paste	Ctrl＋V
Select all	Ctrl＋A
Undo	Ctrl＋Z
Recover	Ctrl＋Y
Switch to row	Ctrl＋G
Find and replace	Ctrl＋F
One-key typesetting	Alt＋F8
Syntax check	Alt＋C
Exit	Alt＋F4
Font	Alt＋S
Display/hide output window	Ctrl＋O
Switch to definition	F12

20.3 Script grammar and function

20.3.1 Picture

20.3.1.1 Basic graphics

1.HMILine object

HMILine

Line control

Methods list

	Name	Description
	FindAnimation	Look for animation to modify the associated variables of the animation

Property list

	Name	Description
	Height	Height
	IsShow	Whether display the specified object or not
	Left	The left coordinate
	Name	Name
	RotateAngle	Rotation Angle
	StrokeThickness	Line width
	ToolTip	Tooltip text
	Top	The top coordinate
	Width	Width
	X1	The horizontical coordinate of the starting point of the line
	X2	The horizontical coordinate of the ending point of the line
	Y1	The vertical coordinate of the starting point of the line
	Y2	The vertical coordinate of the ending point of the line
	ZIndex	Layer index

The following is the detailed description of the script:

【Method】

(1) FindAnimation method

FindAnimation

Look for animation to modify the associated variables of the animation

Define

FindAnimation(name)

Parameter

Name	Required/Optional	Data type	Description
name	Required	**String**	The name of animation

Example

Change the associated variables of the visibility animation of the object into Var.VariableGroup0.Variable1

VBScript Example

```
Line0.FindAnimation("HMIShowHideAnimation").Expression = "Var.VariableGroup0.Variable1"
```

【Property】

(1) Height property

Height

Height

Define

Double Height

Example

Set the height of the specified object to 100

VBScript Example

```
Line0.Height = 100
```

(2) IsShow property

IsShow

Whether display the specified object or not

Define

Boolean IsShow

Example

True ：The specified object is displayed on the screen False ：The specified object hidden on the screen

VBScript Example

```
Line0.IsShow = True     Line0.IsShow = False
```

(3) Left property

Left

The left coordinate

Define

Double Left

Example

Object's left side of the coordinate values is 100 on the screen

VBScript Example

```
Line0.Left = 100
```

(4) Name property

Name

Name

Define

String Name

Example

Get the default name of the specified object

VBScript Example

```
TextBox0.Text = Line0.Name
```

(5) RotateAngle property

RotateAngle

Rotation Angle

Define

Double RotateAngle

Example

Set the rotation Angle of the specified object to 90

VBScript Example

```
Line0.RotateAngle = 90
```

(6) StrokeThickness property

StrokeThickness

Line width

Define

Double StrokeThickness

Example

Basic graphic border line width value is 5

VBScript Example

```
Line0.StrokeThickness = 5
```

(7) ToolTip property

ToolTip

Tooltip text

Define

String ToolTip

Example

Set the ToolTip of the specified object to "test"

VBScript Example

```
Line0.ToolTip = "test"
```

(8) Top property

Top

The top coordinate

Define

Double Top

Example

Object's the top of the coordinate value is 100 in the picture

VBScript Example

```
Line0.Top = 100
```

(9) Width property

Width

Width

Define

Double Width

Example

Set the width of the specified object to 100

VBScript Example
Line0.Width = 100

(10) X1 property

X1

The horizontical coordinate of the starting point of the line

Define

Double X1

Example

The horizontical coordinate of the starting point of the line

VBScript Example
Line0.X1 = 0

(11) X2 property

X2

The horizontical coordinate of the ending point of the line

Define

Double X2

Example

The horizontical coordinate of the ending point of the line

VBScript Example
Line0.X2 = 100

(12) Y1 property

Y1

The vertical coordinate of the starting point of the line

Define

Double Y1

Example

The vertical coordinate of the starting point of the line

VBScript Example
Line0.Y1 = 0

(13) Y2 property

Y2

The vertical coordinate of the ending point of the line

Define

Double Y2

Example

The vertical coordinate of the ending point of the line

VBScript Example
Line0.Y2 = 200

(14) ZIndex property

ZIndex

Layer index

Define

Int32 ZIndex

Example

Set the Zindex of the specified object to 3

VBScript Example

```
Line0.ZIndex = 3
```

2. HMIRectangle object

HMIRectangle

Rectangle control

Methods list

	Name	Description
	FindAnimation	Look for animation to modify the associated variables of the animation

Property list

	Name	Description
	Fill	Fill color
	FillColor	Fill color
	Height	Height
	IsShow	Whether show the specified object or not
	Left	The left coordinate
	Name	Name
	Opacity	Opacity
	RotateAngle	Rotation Angle
	StrokeThickness	Appearance of border line width
	ToolTip	Tooltip text
	Top	The top coordinate
	Width	Width
	ZIndex	Layer index

The following is the detailed description of the script:

【Method】

(1) FindAnimation method

FindAnimation

Look for animation to modify the associated variables of the animation

Define

FindAnimation(name)

Parameter

Name	Required/Optional	Data type	Description
name	Required	**String**	The name of animation

Example

Change the associated variables of the visibility animation of the object into Var.VariableGroup0.Variable1

VBScript Example
Rectangle0.FindAnimation("HMIShowHideAnimation").Expression = "Var.VariableGroup0.Variable1"

【property】

(1) Fill property

Fill

Set the fill color of controls（Colors.red）

Define

Object Fill

Example

Set the fill color of the specified object

VBScript Example
Rectangle0.Fill = Colors.red

(2) FillColor property

FillColor

Gets or sets the string value of the background brush of the controls（"#FFFFFFFF"）

Define

String FillColor

Example

Set the fill color of the specified object

VBScript Example

```
Rectangle0.FillColor = "#FFFFFFFF"
```

(3) Height property

Height

Height

Define

Double Height

Example

Set the height of the specified object to 100

VBScript Example

```
Rectangle0.Height = 100
```

(4) IsShow property

IsShow

Whether display the specified object or not

Define

Boolean IsShow

Example

True ：The specified object is displayed on the screen False ：The specified object hidden on the screen

VBScript Example

```
Rectangle0.IsShow = True    Rectangle0.IsShow = False
```

(5) Left Property

Left

The left coordinate

Define

Double Left

Example

Object's left side of the coordinate values is 100 on the screen

VBScript Example
Rectangle0.Left = 100

(6) Name property

Name

Name

Define

String Name

Example

Get the default name of the specified object

VBScript Example
TextBox0.Text = Rectangle0.Name

(7) Opacity property

Opacity

Opacity

Define

Double Opacity

Example

The graphics displays translucent effect

VBScript Example
Rectangle0.Opacity = 0.5

(8) RotateAngle property

RotateAngle

Rotation Angle

Define

Double RotateAngle

Example

Sets the rotation angle of the specified object to 90

VBScript Example
Rectangle0.RotateAngle = 90

(9) StrokeThickness property

StrokeThickness

Appearance of border line width

Define

Double StrokeThickness

Example

Basic graphic border line width value is 5

VBScript Example
Rectangle0.StrokeThickness = 5

(10) ToolTip property

ToolTip

Tooltip text

Define

String ToolTip

Example

Set the ToolTip of the specified object to "test"

VBScript Example
Rectangle0.ToolTip = "test"

(11) Top property

Top

The top coordinate

Define

Double Top

Example

Object's the top of the coordinate value is 100 in the picture

VBScript Example
Rectangle0.Top = 100

(12) Width property

Width

Width

Define

Double Width

Example

Set the width of the specified object to 100

VBScript Example
Rectangle0.Width = 100

(13) ZIndex property

ZIndex

Layer index

Define

Int32 ZIndex

Example

Set the Zindex of the specified object to 3

VBScript Example
Rectangle0.ZIndex = 3

3. HMIRoundRectangle object

HMIRoundRectangle

Rounded rectangle control

Methods list

	Name	Description
	FindAnimation	Look for animation to modify the associated variables of the animation

Property list

	Name	Description
	Fill	Fill color
	FillColor	Fill color
	Height	Height
	IsShow	Whether show the specify object or not
	Left	The left coordinate
	Name	Name
	Opacity	Opacity
	RadiusX	The Angle of the rounded rectangle variable elliptical x radius of the circle
	RadiusY	The Angle of the rectangular variable round radius of elliptical y
	RotateAngle	Rotation Angle
	StrokeThickness	Appearance of border line width
	ToolTip	Tooltip text
	Top	The top coordinate
	Width	Width
	ZIndex	Layer index

The following is the detailed description of the script:

【Method】

(1) FindAnimation method

FindAnimation

Look for animation to modify the associated variables of the animation

Define

FindAnimation(name)

Parameter

Name	Required/Optional	Data type	Description
name	Required	**String**	The name of animation

Example

Change the associated variables of the visibility animation of the object into Var.VariableGroup0.Variable1

VBScript Example
RoundedRectangle0.FindAnimation("HMIShowHideAnimation").Expression = "Var.VariableGroup0.Variable1"

【property】

(1) Fill property

Fill

Set the fill color of controls（Colors.red）

Define

Object Fill

Example

Set the fill color of the specified object

VBScript Example
RoundedRectangle0.Fill = Colors.red

(2) FillColor property

FillColor

Gets or sets the string（"#FFFFFFFF"）value of controls of the background brush

Define

String FillColor

Example

Set the fill color of the specified object

VBScript Example

```
RoundedRectangle0.FillColor = "#FFFFFFFF"
```

(3) Height property

Height

Height

Define

Double Height

Example

Set the height of the specified object to 100

VBScript Example

```
RoundedRectangle0.Height = 100
```

(4) IsShow property

IsShow

Whether display the specified object or not

Define

Boolean IsShow

Example

True ：The specified object is displayed on the screen False ：The specified object hidden on the screen

VBScript Example

```
RoundedRectangle0.IsShow = True  RoundedRectangle0.IsShow = False
```

(5) Left property

Left

The left coordinate

Define

Double Left

Example

Object's left side of the coordinate values is 100 on the screen

VBScript Example
RoundedRectangle0.Left = 100

(6) Name property

Name

Name

Define

String Name

Example

Get the default name of the specified object

VBScript Example
TextBox0.Text = RoundedRectangle0.Name

(7) Opacity property

Opacity

Opacity

Define

Double Opacity

Example

The graph displays translucent effect

VBScript Example
RoundedRectangle0.Opacity = 0.5

(8) RadiusX property

RadiusX

The ellipse's x radius of the rounded rectangle

Define

Double RadiusX

Example

Set x radius to 45

VBScript Example
RoundedRectangle0.RadiusX = 45

(9) RadiusY property

RadiusY

The ellipse's y radius of the rounded rectangular

Define

Double RadiusY

Example

Set y radius to 45

VBScript Example
RoundedRectangle0.RadiusY = 45

(10) Rotateangle property

RotateAngle

Rotation Angle

Define

Double RotateAngle

Example

Set the rotation angle of the specified object to 90

VBScript Example
RoundedRectangle0.RotateAngle = 90

(11) StrokeThickness property

StrokeThickness

Appearance of border line width

Define

Double StrokeThickness

Example

Basic graphic border line width value is 5

VBScript Example

```
RoundedRectangle0.StrokeThickness = 5
```

(12) ToolTip property

ToolTip

Tooltip text

Define

String ToolTip

Example

Set the ToolTip of the specified object to "test"

VBScript Example

```
RoundedRectangle0.ToolTip = "test"
```

(13) Top property

Top

The top coordinate

Define

Double Top

Example

Object's the top of the coordinate value is 100 in the picture

VBScript Example

```
RoundedRectangle0.Top = 100
```

(14) Width property

Width

Width

Define

Double Width

Example

Set the width of the specified object to 100

VBScript Example
RoundedRectangle0.Width = 100

(15) ZIndex property

ZIndex

Layer index

Define

Int32 ZIndex

Example

Set the Zindex of the specified object to 3

VBScript Example
RoundedRectangle0.ZIndex = 3

4. HMIEllipse object

HMIEllipse

Ellipse control

Methods list

	Name	Description
	FindAnimation	Look for animation to modify the associated variables of the animation

Property list

	Name	Description
	Fill	Fill color
	FillColor	Fill color
	Height	Height

	IsShow	Whether show the specified object or not
	Left	The left coordinate
	Name	Name
	Opacity	Opacity
	RotateAngle	Rotation Angle
	StrokeThickness	Appearance of border line width
	ToolTip	Tooltip text
	Top	The top coordinate
	Width	Width
	ZIndex	Layer index

The following is the detailed description of the script:

【Method】

(1) FindAnimation method

FindAnimation

Look for animation to modify the associated variables of the animation

Define

FindAnimation(name)

Parameter

Name	Required/Optional	Data type	Description
name	Required	**String**	The name of animation

Example

Change the associated variables of the visibility animation of the object into Var.VariableGroup0.Variable1

VBScript Example

```
Ellipse0.FindAnimation("HMIShowHideAnimation").Expression = "Var.VariableGroup0.Variable1"
```

【property】

(1) Fill property

Fill

Set the fill color of controls（Colors.red）

Define

Object Fill

Example

Set the fill color of the specified object

VBScript Example

```
Ellipse0.Fill = Colors.red
```

(2) FillColor property

FillColor

Gets or sets the string（"#FFFFFFFF"）value of controls of the background brush

Define

String FillColor

Example

Set the fill color of the specified object

VBScript Example

```
Ellipse0.FillColor = "#FFFFFFFF"
```

(3) Height property

Height

Height

Define

Double Height

Example

Set the height of the specified object to 100

VBScript Example

```
Ellipse0.Height = 100
```

(4) IsShow property

IsShow

Whether display the specified object or not

Define

Boolean IsShow

Example

True ：The specified object is displayed on the screen False ：The specified object hidden on the screen

VBScript Example

```
Ellipse0.IsShow = True    Ellipse0.IsShow = False
```

(5) Left property

Left

The left coordinate

Define

Double Left

Example

Object's left side of the coordinate values is 100 on the screen

VBScript Example

```
Ellipse0.Left = 100
```

(6) Name property

Name

Name

Define

String Name

Example

Get the default name of the specified object

VBScript Example

```
TextBox0.Text = RoundedRectangle0.Name
```

(7) Opacity property

Opacity

Opacity

Define

Double Opacity

Example

The graphics displays translucent effect

VBScript Example

```
Ellipse0.Opacity = 0.5
```

(8) RotateAngle property

RotateAngle

Rotation Angle

Define

Double RotateAngle

Example

Sets the rotation Angle of the specified object to 90

VBScript Example

```
Ellipse0.RotateAngle = 90
```

(9) StrokeThickness property

StrokeThickness

Appearance of border line width

Define

Double StrokeThickness

Example

Basic graphic border line width value is 5

VBScript Example

```
Ellipse0.StrokeThickness = 5
```

(10) ToolTip property

ToolTip

Tooltip text

Define

String ToolTip

Example

Set the ToolTip of the specified object to "test"

VBScript Example

```
Ellipse0.ToolTip = "test"
```

(11) Top property

Top

The top coordinate

Define

Double Top

Example

Object's the top of the coordinate value is 100 in the picture

VBScript Example

```
Ellipse0.Top = 100
```

(12) Width property

Width

Width

Define

Double Width

Example

Set the width of the specified object to 100

VBScript Example
Ellipse0.Width = 100

(13) ZIndex property

ZIndex

Layer index

Define

Int32 ZIndex

Example

Set the Zindex of the specified object to 3

VBScript Example
Ellipse0.ZIndex = 3

5. HMIPolyLine object

HMIPolyLine

Polyline control

Methods list

	Name	Description
	FindAnimation	Look for animation to modify the associated variables of the animation

Property list

	Name	Description
	Height	Height
	IsShow	Whether display the specified object or not
	Left	The left coordinate
	Name	Name
	RotateAngle	Rotation Angle
	StrokeThickness	Line width

	ToolTip	Tooltip text
	Top	The top coordinate
	Width	Width
	ZIndex	Layer index

The following is the detailed description of the script:

【Method】

(1) FindAnimation method

FindAnimation

Look for animation to modify the associated variables of the animation

Define

FindAnimation(name)

Parameter

Name	Required/Optional	Data type	Description
name	Required	**String**	The name of animation

Example

Change the associated variables of the visibility animation of the object into Var.VariableGroup0.Variable1

VBScript Example

```
PolyLine0.FindAnimation("HMIShowHideAnimation").Expression = "Var.VariableGroup0.Variable1"
```

【property】

(1) Height property

Height

Height

Define

Double Height

Example

Set the height of the specified object to 100

VBScript Example

```
PolyLine0.Height = 100
```

(2) IsShow property

IsShow

Whether display the specified object or not

Define

Boolean IsShow

Example

True ：The specified object is displayed on the screen False ：The specified object hidden on the screen

VBScript Example

```
PolyLine0.IsShow = True     PolyLine0.IsShow = False
```

(3) Left property

Left

The left coordinate

Define

Double Left

Example

Object's left side of the coordinate values is 100 on the screen

VBScript Example

```
PolyLine0.Left = 100
```

(4) Name property

Name

Name

Define

String Name

Example

Get the default name of the specified object

VBScript Example
TextBox0.Text = PolyLine0.Name

(5) RotateAngle property

RotateAngle

Rotation Angle

Define

Double RotateAngle

Example

Sets the rotation Angle of the specified object to 90

VBScript Example
PolyLine0.RotateAngle = 90

(6) StrokeThickness property

StrokeThickness

Line width

Define

Double StrokeThickness

Example

Basic graphic border line width value is 5

VBScript Example
PolyLine0.StrokeThickness =5

(7) ToolTip property

ToolTip

Tooltip text

Define

String ToolTip

Example

Set the ToolTip of the specified object to "test"

VBScript Example
PolyLine0.ToolTip = "test"

(8) Top property

Top

The top coordinate

Define

Double Top

Example

Object's the top of the coordinate value is 100 in the picture

VBScript Example
PolyLine0.Top=100

(9) Width property

Width

Width

Define

Double Width

Example

Set the width of the specified object to 100

VBScript Example
PolyLine0.Width = 100

(10) ZIndex property

ZIndex

Layer index

Define

Int32 ZIndex

Example

Set the Zindex of the specified object to 3

VBScript Example

```
PolyLine0.ZIndex = 3
```

6. HMIPolygon object

HMIPolygon

Polygon control

Methods list

	Name	Description
	FindAnimation	Look for animation to modify the associated variables of the animation

Property list

	Name	Description
	Fill	Fill color
	FillColor	Fill color
	Height	Height
	IsShow	Whether display the specified object or not
	Left	The left coordinate
	Name	Name
	Opacity	Opacity
	RotateAngle	Rotation Angle
	StrokeThickness	Appearance of border line width
	ToolTip	Tooltip text
	Top	The top coordinate
	Width	Width
	ZIndex	Layer index

The following is the detailed description of the script:

【Method】

(1) FindAnimation method

FindAnimation

Look for animation to modify the associated variables of the animation

Define

FindAnimation(name)

Parameter

Name	Required/Optional	Data type	Description
name	Required	**String**	The name of animation

Example

Change the associated variables of the visibility animation of the object into Var.VariableGroup0.Variable1

VBScript Example

```
Polygon0.FindAnimation("HMIShowHideAnimation").Expression = "Var.VariableGroup0.Variable1"
```

【property】

(1) Fill property

Fill

Set the fill color of controls（Colors.red）

Define

Object Fill

Example

Set the fill color of the specified object

VBScript Example

```
Polygon0.Fill = Colors.red
```

(2) FillColor property

FillColor

Gets or sets the string value of controls of the background brush（"#FFFFFFFF"）

Define

String FillColor

Example

Set the fill color of the specified object

VBScript Example
Polygon0.FillColor = "#FFFFFFFF"

(3) Height property

Height

Height

Define

Double Height

Example

Set the height of the specified object to 100

VBScript Example
Polygon0.Height = 100

(4) IsShow property

IsShow

Whether display the specified object or not

Define

Boolean IsShow

Example

True ：The specified object is displayed on the screen False ：The specified object hidden on the screen

VBScript Example
Polygon0.IsShow = True Polygon0.IsShow = False

(5) Left property

Left

The left coordinate

Define

Double Left

Example

Object's left side of the coordinate values is 100 on the screen

VBScript Example

```
Polygon0.Left = 100
```

(6) Name property

Name

Name

Define

String Name

Example

Get the default name of the specified object

VBScript Example

```
TextBox0.Text = Polygon0.Name
```

(7) Opacity property

Opacity

Opacity

Define

Double Opacity

Example

The graphics displays translucent effect

VBScript Example

```
Polygon0.Opacity = 0.5
```

(8) Rotateangle property

RotateAngle

Rotation Angle

Define

Double RotateAngle

Example

Sets the rotation Angle of the specified object to 90

VBScript Example
Polygon0.RotateAngle = 90

(9) StrokeThickness property

StrokeThickness

Appearance of border line width

Define

Double StrokeThickness

Example

Basic graphic border line width value is 5

VBScript Example
Polygon0.StrokeThickness = 5

(10) ToolTip property

ToolTip

Tooltip text

Define

String ToolTip

Example

Set the ToolTip of the specified object to "test"

VBScript Example

```
Polygon0.ToolTip = "test"
```

(11) Top property

Top

The top coordinate

Define

Double Top

Example

Object's the top of the coordinate value is 100 in the picture

VBScript Example

```
Polygon0.Top = 100
```

(12) Width property

Width

Width

Define

Double Width

Example

Set the width of the specified object to 100

VBScript Example

```
Polygon0.Width = 100
```

(13) ZIndex property

ZIndex

Layer index

Define

Int32 ZIndex

Example

Set the Zindex of the specified object to 3

VBScript Example

```
Polygon0.ZIndex = 3
```

7. HMIBezierCurve object

HMIBezierCurve

Bezier curve control

Methods list

	Name	Description
	FindAnimation	Look for animation to modify the associated variables of the animation

Property list

	Name	Description
	Height	Height
	IsShow	Whether display the specified object or not
	Left	The left coordinate
	Name	Name
	RotateAngle	Rotation Angle
	StrokeThickness	Line width
	ToolTip	Tooltip text
	Top	The top coordinate
	Width	Width
	ZIndex	Layer index

The following is the detailed description of the script:

【Method】

(1) FindAnimation method

FindAnimation

Look for animation to modify the associated variables of the animation

Define

FindAnimation(name)

Parameter

Name	Required/Optional	Data type	Description
name	Required	**String**	The name of animation

Example

Change the associated variables of the visibility animation of the object into Var.VariableGroup0.Variable1

VBScript Example
BezierCurve0.FindAnimation("HMIShowHideAnimation").Expression = "Var.VariableGroup0.Variable1"

【property】

(1) Height property

Height

Height

Define

Double Height

Example

Set the height of the specified object to 100

VBScript Example
BezierCurve0.Height = 100

(2) IsShow property

IsShow

Whether display the specified object or not

Define

Boolean IsShow

Example

True ：The specified object is displayed on the screen False ：The specified object hidden on the screen

VBScript Example
BezierCurve0.IsShow = True BezierCurve0.IsShow = False

(3) Left property

Left

The left coordinate

Define

Double Left

Example

Object's left side of the coordinate values is 100 on the screen

VBScript Example
BezierCurve0.Left = 100

(4) Name property

Name

Name

Define

String Name

Example

Get the default name of the specified object

VBScript Example
Text0.Text = BezierCurve0.Name

(5) RotateAngle property

RotateAngle

Rotation Angle

Define

Double RotateAngle

Example

Sets the rotation Angle of the specified object to 90

VBScript Example
BezierCurve0.RotateAngle = 90

(6) StrokeThickness property

StrokeThickness

Line width

Define

Double StrokeThickness

Example

Basic graphic border line width value is 5

VBScript Example
BezierCurve0.StrokeThickness = 5

(7) ToolTip property

ToolTip

Tooltip text

Define

String ToolTip

Example

Set the ToolTip of the specified object to "test"

VBScript Example
BezierCurve0.ToolTip = "test"

(8) Top property

Top

The top coordinate

Define

Double Top

Example

Object's the top of the coordinate value is 100 in the picture

VBScript Example
BezierCurve0.Top = 100

(9) Width property

Width

Width

Define

Double Width

Example

Set the width of the specified object to 100

VBScript Example
BezierCurve0.Width = 100

(10) ZIndex property

ZIndex

Layer index

Define

Int32 ZIndex

Example

Set the Layer index of the specified object to 2

VBScript Example
BezierCurve0.ZIndex = 2

8. HMIClosedCurve object

HMIClosedCurve

Closed curve control

Methods list

	Name	Description
	FindAnimation	Look for animation to modify the associated variables of the animation

Property list

	Name	Description
	Fill	Fill color
	FillColor	Fill color
	Height	Height
	IsShow	Whether display the specified object
	Left	The left coordinate
	Name	Name
	Opacity	Opacity
	RotateAngle	Rotation Angle
	StrokeThickness	Line width of border
	ToolTip	Tooltip text
	Top	The top coordinate
	Width	Width
	ZIndex	Layer index

The following is the detailed description of the script:

【Method】

(1) FindAnimation method

FindAnimation

Look for animation to modify the associated variables of the animation

Define

FindAnimation(name)

Parameter

Name	Required/Optional	Data type	Description
name	Required	**String**	The name of animation

Example

Change the associated variables of the visibility animation of the object into Var.VariableGroup0.Variable1

VBScript Example
ClosedCurve0.FindAnimation("HMIShowHideAnimation").Expression = "Var.VariableGroup0.Variable1"

【property】

(1) Fill property

Fill

Set the fill color of controls（Colors.red）

Define

Object Fill

Example

Set the fill color of the specified object

VBScript Example
ClosedCurve0.Fill = Colors.red

(2) FillColor property

FillColor

Gets or sets the string value of controls of the background brush （"#FFFFFFFF"）

Define

String FillColor

Example

Set the fill color of the specified object

VBScript Example

```
ClosedCurve0.FillColor = "#FFFFFFFF"
```

(3) Height property

Height

Height

Define

Double Height

Example

Set the height of the specified object to 100

VBScript Example

```
ClosedCurve0.Height = 100
```

(4) IsShow property

IsShow

Whether display the specified object or not

Define

Boolean IsShow

Example

True ：The specified object is displayed on the screen False ：The specified object hidden on the screen

VBScript Example

```
ClosedCurve0.IsShow = True     ClosedCurve0.IsShow = False
```

(5) Left property

Left

The left coordinate

Define

Double Left

Example

Object's left side of the coordinate values is 100 on the screen

VBScript Example
ClosedCurve0.Left = 100

(6) Name property

Name

Name

Define

String Name

Example

Get the default name of the specified object

VBScript Example
Text0.Text = ClosedCurve0.Name

(7) Opacity property

Opacity

Opacity

Define

Double Opacity

Example

The graphics displays translucent effect

VBScript Example
ClosedCurve0.Opacity = 0.5

(8) RotateAngle property

RotateAngle

Rotation Angle

Define

Double RotateAngle

Example

Sets the rotation Angle of the specified object to 90

VBScript Example
ClosedCurve0.RotateAngle = 90

(9) StrokeThickness property

StrokeThickness

Line width of border

Define

Double StrokeThickness

Example

Line width of border is 5

VBScript Example
ClosedCurve0.StrokeThickness = 5

(10) ToolTip property

ToolTip

Tooltip text

Define

String ToolTip

Example

Set the ToolTip of the specified object to "test"

VBScript Example
ClosedCurve0.ToolTip = "test"

(11) Top property

Top

The top coordinate

Define

Double Top

Example

Object's the top of the coordinate value is 100 in the picture

VBScript Example
ClosedCurve0.Top = 100

(12) Width property

Width

Width

Define

Double Width

Example

Set the width of the specified object to 100

VBScript Example
ClosedCurve0.Width = 100

(13) ZIndex property

ZIndex

Layer index

Define

Int32 ZIndex

Example

Set the Layer index of the specified object to 2

VBScript Example
ClosedCurve0.ZIndex = 2

9. HMIArc object

HMIArc

Arc control

Methods list

	Name	Description
	FindAnimation	Look for animation to modify the associated variables of the animation

Property list

	Name	Description
	Height	Height
	IsShow	Whether display the specified object
	Left	The left coordinate
	Name	Name
	RotateAngle	Rotation Angle
	StartAngle	Start angle
	StrokeThickness	Line width of border
	SweepAngle	Sweep angle
	ToolTip	Tooltip text
	Top	The top coordinate
	Width	Width
	ZIndex	Layer index

The following is the detailed description of the script:

【Method】

(1) FindAnimation method

FindAnimation

Look for animation to modify the associated variables of the animation

Define

FindAnimation(name)

Parameter

Name	Required/Optional	Data type	Description
name	Required	**String**	The name of animation

Example

Change the associated variables of the visibility animation of the object into Var.VariableGroup0.Variable1

VBScript Example
Arc0.FindAnimation("HMIShowHideAnimation").Expression = "Var.VariableGroup0.Variable1"

【property】

(1) Height property

Height

Height

Define

Double Height

Example

Set the height of the specified object to 100

VBScript Example
Arc0.Height = 100

(2) IsShow property

IsShow

Whether display the specified object or not

Define

Boolean IsShow

Example

True ：The specified object is displayed on the screen False ：The specified object hidden on the screen

VBScript Example
Arc0.IsShow = True Arc0.IsShow = False

(3) Left property

Left

The left coordinate

Define

Double Left

Example

Object's left side of the coordinate values is 100 on the screen

VBScript Example

```
Arc0.Left = 100
```

(4) Name property

Name

Name

Define

String Name

Example

Get the default name of the specified object

VBScript Example

```
Text0.Text = Arc0.Name
```

(5) RotateAngle property

RotateAngle

Rotation Angle

Define

Double RotateAngle

Example

Sets the rotation Angle of the specified object to 90

VBScript Example

```
Arc0.RotateAngle = 90
```

(6) StartAngle property

StartAngle

Start angle

Define

Double StartAngle

Example

The Start Angle value is 90 of the circular arc

VBScript Example

```
Arc0.StartAngle = 90
```

(7) StrokeThickness property

StrokeThickness

Line width

Define

Double StrokeThickness

Example

Basic graphic border line width value is 5

VBScript Example

```
Arc0.StrokeThickness = 5
```

(8) SweepAngle property

SweepAngle

Sweep Angle

Define

Double SweepAngle

Example

Sweep Angle value is 90

VBScript Example

```
Arc0.SweepAngle = 90
```

(9) ToolTip property

ToolTip

Tooltip text

Define

String ToolTip

Example

Set the ToolTip of the specified object to "test"

VBScript Example

```
Arc0.ToolTip = "test"
```

(10) Top property

Top

The top coordinate

Define

Double Top

Example

Object's the top of the coordinate value is 100 in the picture

VBScript Example

```
Arc0.Top = 100
```

(11) Width property

Width

Width

Define

Double Width

Example

Set the width of the specified object to 100

VBScript Example

```
Arc0.Width = 100
```

(12) ZIndex property

ZIndex

Layer index

Define

Int32 ZIndex

Example

Set the Layer index of the specified object to 2

VBScript Example

```
Arc0.ZIndex = 2
```

10. HMIArch object

HMIArch

Arcuate control

Methods list

	Name	Description
	FindAnimation	Look for animation to modify the associated variables of the animation

Property list

	Name	Description
	Fill	Fill color
	FillColor	Fill color
	Height	Height
	IsShow	Whether display the specified object or not
	Left	The left coordinate
	Name	Name
	Opacity	Opacity
	RotateAngle	Rotation Angle
	StartAngle	Start angle

	StrokeThickness	Line width
	SweepAngle	Sweep Angle
	ToolTip	Tooltip text
	Top	The top coordinate
	Width	Width
	ZIndex	Layer index

The following is the detailed description of the script:

【Method】

(1) FindAnimation method

FindAnimation

Look for animation to modify the associated variables of the animation

Define

FindAnimation(name)

Parameter

Name	Required/Optional	Data type	Description
name	Required	**String**	The name of animation

Example

Change the associated variables of the visibility animation of the object into Var.VariableGroup0.Variable1

VBScript Example

```
Arch0.FindAnimation("HMIShowHideAnimation").Expression = "Var.VariableGroup0.Variable1"
```

【property】

(1) Fill property

Fill

Set the fill color of controls（Colors.red）

Define

Object Fill

Example

Set the fill color of the specified object

VBScript Example
Arch0.Fill = Colors.red

(2) FillColor property

FillColor

Gets or sets the string value of controls of the background brush （"#FFFFFFFF"）

Define

String FillColor

Example

Set the fill color of the specified object

VBScript Example
Arch0.FillColor = "#FFFFFFFF"

(3) Height property

Height

Height

Define

Double Height

Example

Set the height of the specified object to 100

VBScript Example
Arch0.Height = 100

(4) IsShow property

IsShow

Whether display the specified object or not

Define

Boolean IsShow

Example

True ：The specified object is displayed on the screen False ：The specified object hidden on the screen

VBScript Example
Arch0.IsShow = True Arch0.IsShow = False

(5) Left property

Left

The left coordinate

Define

Double Left

Example

Object's left side of the coordinate values is 100 on the screen

VBScript Example
Arch0.Left = 100

(6) Name property

Name

Name

Define

String Name

Example

Get the default name of the specified object

VBScript Example
Text0.Text = Arch0.Name

(7) Opacity property

Opacity

Opacity

Define

Double Opacity

Example

The graphics displays translucent effect

VBScript Example

```
Arch0.Opacity = 0.5
```

(8) RotateAngle property

RotateAngle

Rotation Angle

Define

Double RotateAngle

Example

Sets the rotation Angle of the specified object to 90

VBScript Example

```
Arch0.RotateAngle = 90
```

(9) StartAngle property

StartAngle

Start angle

Define

Double StartAngle

Example

The Start Angle value is 90 of the circular arc

VBScript Example

```
Arch0.StartAngle = 90
```

(10) StrokeThickness property

StrokeThickness

Appearance of border line width

Define

Double StrokeThickness

Example

Basic graphic border line width value is 5

VBScript Example

```
Arch0.StrokeThickness = 5
```

(11) SweepAngle property

StrokeThickness

Appearance of border line width

Define

Double StrokeThickness

Example

Basic graphic border line width value is 5

VBScript Example

```
Arch0.StrokeThickness = 5
```

(12) ToolTip property

ToolTip

Tooltip text

Define

String ToolTip

Example

Set the ToolTip of the specified object to "test"

VBScript Example

```
Arch0.ToolTip = "test"
```

(13) Top property

Top

The top coordinate

Define

Double Top

Example

Object's the top of the coordinate value is 100 in the picture

VBScript Example

```
Arch0.Top = 100
```

(14) Width property

Width

Width

Define

Double Width

Example

Set the width of the specified object to 100

VBScript Example

```
Arch0.Width = 100
```

(15) ZIndex property

ZIndex

Layer index

Define

Int32 ZIndex

Example

Set the Layer index of the specified object to 2

VBScript Example

```
Arch0.ZIndex = 2
```

11. HMIPie object

HMIPie

Sector control

Methods list

	Name	Description
	FindAnimation	Look for animation to modify the associated variables of the animation

Property list

	Name	Description
	Fill	Fill color
	FillColor	Fill color
	Height	Height
	IsShow	Whether display the specified object or not
	Left	The left coordinate
	Name	Name
	Opacity	Opacity
	RotateAngle	Rotation Angle
	StartAngle	Start Angle
	StrokeThickness	Appearance of border line width
	SweepAngle	Sector scan Angle
	ToolTip	Tooltip text
	Top	The top coordinate
	Width	Width
	ZIndex	Layer index

The following is the detailed description of the script:

【Method】

(1) FindAnimation method

FindAnimation

Look for animation to modify the associated variables of the animation

Define

FindAnimation(name)

Parameter

Name	Required/Optional	Data type	Description
name	Required	**String**	The name of animation

Example

Change the associated variables of the visibility animation of the object into Var.VariableGroup0.Variable1

VBScript Example
Pie0.FindAnimation("HMIShowHideAnimation").Expression = "Var.VariableGroup0.Variable1"

【property】

(1) Fill property

Fill

Set the fill color of the control（Colors.red）

Define

Object Fill

Example

Set the fill color of the specified object

VBScript Example
Pie0.Fill = Colors.red

(2) FillColor property

FillColor

Gets or sets the string value of controls of the background brush（"#FFFFFFFF"）

Define

String FillColor

Example

Set the fill color of the specified object

VBScript Example
Pie0.FillColor = "#FFFFFFFF"

(3) Height property

Height

Height

Define

Double Height

Example

Set the height of the specified object to 100

VBScript Example
Pie0.Height = 100

(4) IsShow property

IsShow

Whether display the specified object or not

Define

Boolean IsShow

Example

True ：The specified object is displayed on the screen False ：The specified object hidden on the screen

VBScript Example	
Pie0.IsShow = True	Pie0.IsShow = False

(5) Left property

Left

The left coordinate

Define

Double Left

Example

Object's left side of the coordinate values is 100 on the screen

VBScript Example
Pie0.Left = 100

(6) Name property

Name

Name

Define

String Name

Example

Get the default name of the specified object

VBScript Example
Text0.Text = Pie0.Name

(7) Opacity property

Opacity

Opacity

Define

Double Opacity

Example

The graphics displays translucent effect

VBScript Example
Pie0.Opacity = 0.5

(8) RotateAngle property

RotateAngle

Rotation Angle

Define

Double RotateAngle

Example

Set the rotation Angle of the specified object to 90

VBScript Example

```
Pie0.RotateAngle = 90
```

(9) StartAngle property

StartAngle

Start Angle

Define

Double StartAngle

Example

The starting Angle value of the fan is 90

VBScript Example

```
Pie0.StartAngle = 90
```

(10) StrokeThickness property

StrokeThickness

Appearance of border line width

Define

Double StrokeThickness

Example

Basic graphic border line width value is 5

VBScript Example

```
Pie0.StrokeThickness = 5
```

(11) SweepAngle property

SweepAngle

Sector scan Angle

Define

Double SweepAngle

Example

Sector scan Angle value is 90

VBScript Example

```
Pie0.SweepAngle = 90
```

(12) ToolTip property

ToolTip

Tooltip text

Define

String ToolTip

Example

Set the ToolTip of the specified object to "test"

VBScript Example

```
Pie0.ToolTip = "test"
```

(13) Top property

Top

The top coordinate

Define

Double Top

Example

Object's the top of the coordinate value is 100 in the picture

VBScript Example

```
Pie0.Top = 100
```

(14) Width property

Width

Width

Define

Double Width

Example

Set the width of the specified object to 100

VBScript Example

```
Pie0.Width = 100
```

(15) ZIndex property

ZIndex

Layer index

Define

Int32 ZIndex

Example

Set the Layer index of the specified object to 2

VBScript Example

```
Pie0.ZIndex = 2
```

12. HMIPipe object

HMIPipe

Pipe control

Methods list

	Name	Description
	FindAnimation	Look for animation to modify the associated variables of the animation

Property list

	Name	Description
	ForwardFlow	Liquid flow direction
	Height	Height
	IsShow	Whether display the specified object or not
	Left	The left coordinate
	LiquidWidth	Liquid width
	Name	Name
	PipeWidth	Pipe width
	RotateAngle	Rotation angle
	ToolTip	Tooltip text
	Top	The top coordinate
	Width	Width
	ZIndex	Layer index

The following is the detailed description of the script:

【Method】

(1) FindAnimation method

FindAnimation

Look for animation to modify the associated variables of the animation

Define

FindAnimation(name)

Parameter

Name	Required/Optional	Data type	Description
name	Required	**String**	The name of animation

Example

Change the associated variables of the visibility animation of the object into Var.VariableGroup0.Variable1

VBScript Example

```
Pipe0.FindAnimation("HMIShowHideAnimation").Expression = "Var.VariableGroup0.Variable1"
```

【property】

(1) ForwardFlow property

ForwardFlow

Liquid flow direction

Define

Boolean ForwardFlow

Example

Set the flow direction of the specified object to True

VBScript Example

```
Pipe0.ForwardFlow = True
```

(2) Height property

Height

Height

Define

Double Height

Example

Set the height of the specified object to 100

VBScript Example

```
Pipe0.Height = 100
```

(3) IsShow property

IsShow

Whether display the specified object or not

Define

Boolean IsShow

Example

True ：The specified object is displayed on the screen False ：The specified object hidden on the screen

VBScript Example
Pipe0.IsShow = True Pipe0.IsShow = False

(4) Left property

Left

The left coordinate

Define

Double Left

Example

Object's left side of the coordinate values is 100 on the screen

VBScript Example
Pipe0.Left = 100

(5) LiquidWidth property

LiquidWidth

Liquid width

Define

Double LiquidWidth

Example

VBScript Example
Pipe0.LiquidWidth = 30

(6) Name property

Name

Name

Define

String Name

Example

Get the default name of the specified object

VBScript Example
Text0.Text = Pipe0.Name

(7) PipeWidth property

PipeWidth

Pipe width

Define

Double PipeWidth

Example

Set the pipe width to 100

VBScript Example
Pipe0.PipeWidth = 100

(8) RotateAngle property

RotateAngle

Rotation Angle

Define

Double RotateAngle

Example

Set the rotation Angle of the specified object to 90

VBScript Example
Pipe0.RotateAngle = 90

(9) ToolTip property

ToolTip

Tooltip text

Define

String ToolTip

Example

Set the ToolTip of the specified object to "test"

VBScript Example

```
Pipe0.ToolTip = "test"
```

(10) Top property

Top

The top coordinate

Define

Double Top

Example

Object's the top of the coordinate value is 100 in the picture

VBScript Example

```
Pipe0.Top = 100
```

(11) Width property

Width

Width

Define

Double Width

Example

Set the width of the specified object to 100

VBScript Example

```
Pipe0.Width = 100
```

(12) ZIndex property

ZIndex

Layer index

Define

Int32 ZIndex

Example

Set the Layer index of the specified object to 2

VBScript Example

```
Pipe0.ZIndex = 2
```

13. HMIText object

HMIText

Text control

Methods list

	Name	Description
	FindAnimation	Look for animation to modify the associated variables of the animation

Property list

	Name	Description
	FontSize	Font size
	Height	Height
	IsShow	Whether display the specified object or not
	Left	The left coordinate
	Name	Name
	Opacity	Opacity
	RotateAngle	Rotation angle
	Text	The text content
	ToolTip	Tooltip text
	Top	The top coordinate
	Width	Width
	ZIndex	Layer index

The following is the detailed description of the script:

【Method】

(1) FindAnimation method

FindAnimation

Look for animation to modify the associated variables of the animation

Define

FindAnimation(name)

Parameter

Name	Required/Optional	Data type	Description
name	Required	**String**	The name of animation

Example

Change the associated variables of the visibility animation of the object into Var.VariableGroup0.Variable1

VBScript Example
Text0.FindAnimation("HMIShowHideAnimation").Expression = "Var.VariableGroup0.Variable1"

【property】

(1) FontSize property

FontSize

Font size

Define

Double FontSize

Example

Set the font size of the specified object

VBScript Example
Text0.FontSize = 20

(2) Height property

Height

Height

Define

Double Height

Example

Set the height of the specified object to 100

VBScript Example

```
Text0.Height = 100
```

(3) IsShow property

IsShow

Whether display the specified object or not

Define

Boolean IsShow

Example

True ：The specified object is displayed on the screen False ：The specified object hidden on the screen

VBScript Example

```
Text0.IsShow = True     Text0.IsShow = False
```

(4) Left property

Left

The left coordinate

Define

Double Left

Example

Object's left side of the coordinate values is 100 on the screen

VBScript Example

```
Text0.Left = 100
```

(5) Name property

Name

Name

Define

String Name

Example

Get the default name of the specified object

VBScript Example

```
TextBox0.Text = Text0.Name
```

(6) Opacity property

Opacity

Opacity

Define

Double Opacity

Example

The graphics displays translucent effect

VBScript Example

```
Text0.Opacity = 0.5
```

(7) RotateAngle property

RotateAngle

Rotation Angle

Define

Double RotateAngle

Example

Set the rotation Angle of the specified object to 90

VBScript Example

```
Text0.RotateAngle = 90
```

(8) Text property

Text

The text content

Define

String Text

Example

Set the display text

VBScript Example

```
Text0.Text = "SCADA"
```

(9) ToolTip property

ToolTip

Tooltip text

Define

String ToolTip

Example

Set the ToolTip of the specified object to "test"

VBScript Example

```
Text0.ToolTip = "test"
```

(10) Top property

Top

The top coordinate

Define

Double Top

Example

Object's the top of the coordinate value is 100 in the picture

VBScript Example
Text0.Top = 100

(11) Width property

Width

Width

Define

Double Width

Example

Set the width of the specified object to 100

VBScript Example
Text0.Width = 100

(12) ZIndex property

ZIndex

Layer index

Define

Int32 ZIndex

Example

Set the Layer index of the specified object to 2

VBScript Example
Text0.ZIndex = 2

14. HMIGroup property

HMIGroup

Combinatorial graphics control

Methods list

	Name	Description
	FindAnimation	Look for animation to modify the associated variables of the animation

Property list

	Name	Description
	Height	Height
	IsShow	Whether display the specified object or not
	Left	The left coordinate
	Name	Name
	RotateAngle	Rotation angle
	ToolTip	Tooltip text
	Top	The top coordinate
	Width	Width

The following is the detailed description of the script:

【Method】

(1) FindAnimation method

FindAnimation

Look for animation to modify the associated variables of the animation

Define

FindAnimation(name)

Parameter

Name	Required/Optional	Data type	Description
name	Required	**String**	The name of animation

Example

Change the associated variables of the visibility animation of the object into Var.VariableGroup0.Variable1

VBScript Example

```
Group0.FindAnimation("HMIShowHideAnimation").Expression = "Var.VariableGroup0.Variable1"
```

【property】

(1) Height property

Height

Height

Define

Double Height

Example

Set the height of the specified object to 100

VBScript Example

```
Group0.Height = 100
```

(2) IsShow property

IsShow

Whether display the specified object or not

Define

Boolean IsShow

Example

True ：The specified object is displayed on the screen False ：The specified object hidden on the screen

VBScript Example

```
Group0.IsShow = True      Group0.IsShow = False
```

(3) Left property

Left

The left coordinate

Define

Double Left

Example

Object's left side of the coordinate values is 100 on the screen

VBScript Example
Group0.Left = 100

(4) Name property

Name

Name

Define

String Name

Example

Get the default name of the specified object

VBScript Example
TextBox0.Text = Group0.Name

(5) RotateAngle property

RotateAngle

Rotation Angle

Define

Double RotateAngle

Example

Set the rotation Angle of the specified object to 90

VBScript Example
Group0.RotateAngle = 90

(6) ToolTip property

ToolTip

Tooltip text

Define

String ToolTip

Example

Set the ToolTip of the specified object to "test"

VBScript Example
Group0.ToolTip = "test"

(7) Top property

Top

The top coordinate

Define

Double Top

Example

Object's the top of the coordinate value is 100 in the picture

VBScript Example
Group0.Top = 100

(8) Width property

Width

Width

Define

Double Width

Example

Set the width of the specified object to 100

VBScript Example
Group0.Width = 100

20.3.1.2 Window controls

1. HMIButton object

HMIButton

Button control

Methods list

	Name	Description
	FindAnimation	Look for animation to modify the associated variables of the animation

Property list

	Name	Description
	Background	The background color
	BorderBrush	Border color
	Content	Get or the set the text content of button
	FontSize	Font size
	Foreground	Foreground
	Height	Height
	IsEnabled	Get or set a value, the value indicating whether the control can operation
	IsShow	Whether display the specified object
	IsTriggerOpen	Whether to use the default styles
	Left	The left coordinate
	Name	Name
	Opacity	Opacity
	RotateAngle	Rotation Angle
	ToolTip	Tooltip text
	Top	The top coordinate
	Width	Width
	ZIndex	Layer index

The following is the detailed description of the script:

【Method】

(1) FindAnimation method

FindAnimation

Look for animation to modify the associated variables of the animation

Define

FindAnimation(name)

Parameter

Name	Required/Optional	Data type	Description
name	Required	**String**	The name of animation

Example

Change the associated variables of the visibility animation of the object into Var.VariableGroup0.Variable1

VBScript Example
Button0.FindAnimation("HMIShowHideAnimation").Expression = "Var.VariableGroup0.Variable1"

【property】

(1) Background property

Background

The background color（Colors.Red）

Define

Object Background

Example

Set the background color of the specified object

VBScript Example
Button0.Background = Colors.Red

(2) BorderBrush property

BorderBrush

Border color（Colors.Red）

Define

Object BorderBrush

Example

Set the border color of the specified object

VBScript Example
Button0.BorderBrush = Colors.Red

(3) Content property

Content

Gets or sets the text content of button

Define

String Content

Example

Set the text of the specified object to "Delta SCADA"

VBScript Example
Button0.Content = "Delta SCADA"

(4) FontSize property

FontSize

Font size

Define

Double FontSize

Example

Set the font size of the specified object

VBScript Example
Button0.FontSize = 20

(5) Foreground property

Foreground

The foreground（Colors.Red）

Define

Object Foreground

Example

Set the foreground color of the specified object

VBScript Example
Button0.Foreground = Colors.Red

(6) Height property

Height

Height

Define

Double Height

Example

Set the height of the specified object to 100

VBScript Example
Button0.Height = 100

(7) IsEnabled property

IsEnabled

Gets or sets a value, the value indicating whether the control can operation

Define

Boolean IsEnabled

Example

True ：The specified object is Enabled

VBScript Example
Button0.IsEnabled = True

(8) IsShow property

IsShow

Whether display the specified object or not

Define

Boolean IsShow

Example

True ：The specified object is displayed on the screen False ：The specified object hidden on the screen

VBScript Example	
Button0.IsShow = True	Button0.IsShow = False

(9) IsTriggerOpen

IsTriggerOpen

Whether to use the default styles

Define

Boolean IsTriggerOpen

Example

True ：Refers to using the default style，False ：Don't use the default style

VBScript Example
Button0.IsTriggerOpen = True

(10) Left property

Left

The left coordinate

Define

Double Left

Example

Object's left side of the coordinate values is 100 on the screen

VBScript Example

```
Button0.Left = 100
```

(11) Name property

Name

Name

Define

String Name

Example

Get the default name of the specified object

VBScript Example

```
TextBox0.Text = Button0.Name
```

(12) Opacity property

Opacity

Opacity

Define

Double Opacity

Example

The graphics display translucent effect

VBScript Example

```
Button0.Opacity = 0.5
```

(13) RotateAngle property

RotateAngle

Rotation Angle

Define

Double RotateAngle

Example

Set the rotation Angle of the specified object to 90

VBScript Example

```
Button0.RotateAngle = 90
```

(14) ToolTip property

ToolTip

Tooltip text

Define

String ToolTip

Example

Set the ToolTip of the specified object to "test"

VBScript Example

```
Button0.ToolTip = "test"
```

(15) Top property

Top

The top coordinate

Define

Double Top

Example

Object's the top of the coordinate value is 100 in the picture

VBScript Example

```
Button0.Top = 100
```

(16) Width property

Width

Width

Define

Double Width

Example

Set the width of the specified object to 100

VBScript Example

```
Button0.Width = 100
```

(17) ZIndex property

ZIndex

Layer index

Define

Int32 ZIndex

Example

Set the Layer index of the specified object to 3

VBScript Example

```
Button0.ZIndex = 3
```

2. HMICheckBox object

HMICheckBox

Check box control

Methods list

	Name	Description
	FindAnimation	Look for animation to modify the associated variables of the animation

Property list

	Name	Description
	Content	Get or the set the text content of check box
	FontSize	Font size
	Foreground	Foreground
	Height	Height
	IsChecked	Get or set whether the control is selected
	IsEnabled	Get or set a value, the value indicating whether the control can operation
	IsShow	Whether display the specified object
	Left	The left coordinate

	Name	Name
	Opacity	Opacity
	RotateAngle	Rotation Angle
	ToolTip	Tooltip text
	Top	The top coordinate
	Width	Width
	ZIndex	Layer index

The following is the detailed description of the script:

【Method】

(1) FindAnimation method

FindAnimation

Look for animation to modify the associated variables of the animation

Define

FindAnimation(name)

Parameter

Name	Required/Optional	Data type	Description
name	Required	**String**	The name of animation

Example

Change the associated variables of the visibility animation of the object into Var.VariableGroup0.Variable1

VBScript Example

```
CheckBox0.FindAnimation("HMIShowHideAnimation").Expression = "Var.VariableGroup0.Variable1"
```

【property】

(1) Content property

Content

Get or the set the text content of check box

Define

String Content

Example

Set the text of the specified object to "Delta SCADA"

VBScript Example

```
CheckBox0.Content = "Delta SCADA"
```

(2) FontSize property

FontSize

Font size

Define

Double FontSize

Example

Set the font size of the specified object

VBScript Example

```
CheckBox0.FontSize = 20
```

(3) Foreground property

Foreground

The foreground color（Colors.Red）

Define

Object Foreground

Example

Set the foreground color of the specified object

VBScript Example

```
CheckBox0.Foreground = Colors.Red
```

(4) Height property

Height

Height

Define

Double Height

Example

Set the height of the specified object to 100

VBScript Example
CheckBox0.Height = 100

(5) IsChecked property

IsChecked

Gets or sets whether the control is selected

Define

Boolean IsChecked

Example

The control is selected

VBScript Example
CheckBox0.IsChecked = True

(6) IsEnabled property

IsEnabled

Gets or sets a value, the value indicating whether the control can operation

Define

Boolean IsEnabled

Example

True ：The specified object is Enabled

VBScript Example
CheckBox0.IsEnabled = True

(7) IsShow property

IsShow

Whether display the specified object or not

Define

Boolean IsShow

Example

True ：The specified object is displayed on the screen False ：The specified object hidden on the screen

VBScript Example

```
CheckBox0.IsShow = True CheckBox0.IsShow = False
```

(8) Left property

Left

The left coordinate

Define

Double Left

Example

Object's left side of the coordinate values is 100 on the screen

VBScript Example

```
CheckBox0.Left = 100
```

(9) Name property

Name

Name

Define

String Name

Example

Get the default name of the specified object

VBScript Example

```
TextBox0.Text = CheckBox0.Name
```

(10) Opacity property

Opacity

Opacity

Define

Double Opacity

Example

The graphics display translucent effect

VBScript Example

```
CheckBox0.Opacity = 0.5
```

(11) RotateAngle property

RotateAngle

Rotation Angle

Define

Double RotateAngle

Example

Set the rotation Angle of the specified object to 90

VBScript Example

```
CheckBox0.RotateAngle = 90
```

(12) ToolTip property

ToolTip

Tooltip text

Define

String ToolTip

Example

Set the ToolTip of the specified object to "test"

VBScript Example
CheckBox0.ToolTip = "test"

(13) Top property

Top

The top coordinate

Define

Double Top

Example

Object's the top of the coordinate value is 100 in the picture

VBScript Example
CheckBox0.Top = 100

(14) Width property

Width

Width

Define

Double Width

Example

Set the width of the specified object to 100

VBScript Example
CheckBox0.Width = 100

(15) ZIndex property

ZIndex

Layer index

Define

Int32 ZIndex

Example

Set the Layer index of the specified object to 3

VBScript Example

```
CheckBox0.ZIndex = 3
```

3. HMIComboBox object

HMIComboBox

Combo box control

Methods list

	Name	Description
	AddItem	Add item
	AddItems	Add items collection
	ClearItems	Empty the collection
	FindAnimation	Look for animation to modify the associated variables of the animation
	RemoveAtItem	Remove the specified items according to the index
	RemoveItem	Remove the specified items

Property list

	Name	Description
	Background	The background color
	BorderBrush	Border color
	FontSize	Font size
	Foreground	Foreground
	Height	Height
	IsEnabled	Get or set a value, the value indicating whether the control can operation
	IsShow	Whether display the specified object
	Left	The left coordinate
	Name	Name
	Opacity	Opacity
	RotateAngle	Rotation Angle
	SelectedIndex	Select the specified item accordding to the index

	SelectedValue	Select the current value
	Text	Get or the set up the Combo box text content
	ToolTip	Tooltip text
	Top	The top coordinate
	Width	Width
	ZIndex	Layer index

The following is the detailed description of the script:

【Method】

(1) AddItem method

AddItem

Add item

Define

AddItem(item)

Parameter

Name	Required/Optional	Data Type	Description
item	Required	**String**	to add items

Example

Add item“a”

VBScript Example

```
ComboBox0.AddItem("a")
```

(2) AddItems method

AddItems

Add items collection

Define

AddItems(items)

Parameter

Name	Required/Optional	Data Type	Description
items	required	**String**	To add items, separated by commas

Example

1.Add items collection“a,b,c”

VBScript Example
ComboBox0.AddItems("a,b,c")

2.Add items collection “all system users”

VBScript Example
ComboBox0.AddItems(UserCmd.GetAllUserNames())

(3) ClearItems method

ClearItems

Empty the collection

Define

ClearItems()

Example

Empty the collection

VBScript Example
ComboBox0.ClearItems()

(4) FindAnimation method

FindAnimation

Look for animation to modify the associated variables of the animation

Define

FindAnimation(name)

Parameter

Name	Required/Optional	Data type	Description
name	Required	**String**	The name of animation

Example

Change the associated variables of the visibility animation of the object into Var.VariableGroup0.Variable1

VBScript Example
ComboBox0.FindAnimation("HMIShowHideAnimation").Expression = "Var.VariableGroup0.Variable1"

(5) RemoveAtItem method

RemoveAtItem

Remove the specified items according to the index

Define

RemoveAtItem(index)

Parameter

Name	Required/Optional	Data Type	Description
index	Required	**int**	Remove the specified items according to the index

Example

Remove the first item in the combo box

VBScript Example
ComboBox0.RemoveAtItem(0)

(6) RemoveItem

RemoveItem

Remove the specified items

Define

RemoveItem(item)

Parameter

Name	Required/Optional	Data Type	Description
item	required	**String**	Items to be removed

Example

Remove the specified item“a”

VBScript Example
ComboBox0.RemoveItem("a")

【property】

(1) BorderBrush property

BorderBrush

The border color（Colors.Red）

Define

Object BorderBrush

Example

Set the border color of the specified object

VBScript Example
ComboBox0.BorderBrush = Colors.Red

(2) FontSize property

FontSize

Font size

Define

Double FontSize

Example

Set the font size of the specified object

VBScript Example

```
ComboBox0.FontSize = 20
```

(3) Foreground property

Foreground

The foreground color（Colors.Red）

Define

Object Foreground

Example

Set the foreground color of the specified object

VBScript Example

```
ComboBox0.Foreground = Colors.Red
```

(4) Height property

Height

Height

Define

Double Height

Example

Set the height of the specified object to 100

VBScript Example

```
ComboBox0.Height = 100
```

(5) IsEnabled property

IsEnabled

Gets or sets a value, the value indicating whether the control can operation

Define

Boolean IsEnabled

Example

True ：The specified object is Enabled

VBScript Example
ComboBox0.IsEnabled = True

(6) IsShow property

IsShow

Whether display the specified object or not

Define

Boolean IsShow

Example

True ：The specified object is displayed on the screen False ：The specified object hidden on the screen

VBScript Example
ComboBox0.IsShow = True ComboBox0.IsShow = False

(7) Left property

Left

The left coordinate

Define

Double Left

Example

Object's left side of the coordinate values is 100 on the screen

VBScript Example
ComboBox0.Left = 100

(8) Name property

Name

Name

Define

String Name

Example

Get the default name of the specified object

VBScript Example
TextBox0.Text = ComboBox0.Name

(9) Opacity property

Opacity

Opacity

Define

Double Opacity

Example

The graphics display translucent effect

VBScript Example
ComboBox0.Opacity = 0.5

(10) RotateAngle property

RotateAngle

Rotation Angle

Define

Double RotateAngle

Example

Set the rotation Angle of the specified object to 90

VBScript Example
ComboBox0.RotateAngle = 90

(11) SelectedIndex

SelectedIndex

Select the specified item accordding to the index

Define

Int32 SelectedIndex

Example

Select the first item in the combo box

VBScript Example

```
ComboBox0.SelectedIndex = 0
```

(12) SelectedValue

SelectedValue

Select the current value

Define

String SelectedValue

Example

Select "a" in the combo box as the current value

VBScript Example

```
ComboBox0.SelectedValue = "a"
```

(13) Text property

Text

Get or the set up the text content of the combo box

Define

String Text

Example

Set the text of the specified object to "Delta SCADA"

VBScript Example

```
ComboBox0.Text = "Delta SCADA"
```

(14) ToolTip property

ToolTip

Tooltip text

Define

String ToolTip

Example

Set the ToolTip of the specified object to "test"

VBScript Example

```
ComboBox0.ToolTip = "test"
```

(15) Top property

Top

The top coordinate

Define

Double Top

Example

Object's the top of the coordinate value is 100 in the picture

VBScript Example

```
ComboBox0.Top = 100
```

(16) Width property

Width

Width

Define

Double Width

Example

Set the width of the specified object to 100

VBScript Example

```
ComboBox0.Width = 100
```

(17) ZIndex property

ZIndex

Layer index

Define

Int32 ZIndex

Example

Set the Layer index of the specified object to 3

VBScript Example

```
ComboBox0.ZIndex = 3
```

4. HMILabel object

HMILabel

Label control

Methods list

	Name	Description
	FindAnimation	Look for animation to modify the associated variables of the animation

Property list

	Name	Description
	Background	The background color
	FontSize	Font size
	Foreground	Foreground
	Height	Height
	IsEnabled	Get or set a value, the value indicating whether the control can operation
	IsShow	Whether display the specified object
	Left	The left coordinate
	Name	Name
	Opacity	Opacity

	RotateAngle	Rotation Angle
	Text	Get or the set up the text content of the label
	ToolTip	Tooltip text
	Top	The top coordinate
	Width	Width
	ZIndex	Layer index

The following is the detailed description of the script:

【Method】

(1) FindAnimation method

FindAnimation

Look for animation to modify the associated variables of the animation

Define

FindAnimation(name)

Parameter

Name	Required/Optional	Data type	Description
name	Required	**String**	The name of animation

Example

Change the associated variables of the visibility animation of the object into Var.VariableGroup0.Variable1

VBScript Example

```
Label0.FindAnimation("HMIShowHideAnimation").Expression = "Var.VariableGroup0.Variable1"
```

【property】

(1) Background property

Background

The background color（Colors.Red）

Define

Object Background

Example

Set the background color of the specified object

VBScript Example
Label0.Background = Colors.Red

(2) FontSize property

FontSize

Font size

Define

Double FontSize

Example

Set the font size of the specified object

VBScript Example
Label0.FontSize = 20

(3) Foreground property

Foreground

The foreground color（Colors.Red）

Define

Object Foreground

Example

Set the foreground color of the specified object

VBScript Example
Label0.Foreground = Colors.Red

(4) Height property

Height

Height

Define

Double Height

Example

Set the height of the specified object to 100

VBScript Example
Label0.Height = 100

(5) IsEnabled property

IsEnabled

Gets or sets a value, the value indicating whether the control can operation

Define

Boolean IsEnabled

Example

True ：The specified object is Enabled

VBScript Example
Label0.IsEnabled = True

(6) IsShow property

IsShow

Whether display the specified object or not

Define

Boolean IsShow

Example

True ：The specified object is displayed on the screen False ：The specified object hidden on the screen

VBScript Example
Label0.IsShow = True Label0.IsShow = False

(7) Left property

Left

The left coordinate

Define

Double Left

Example

Object's left side of the coordinate values is 100 on the screen

VBScript Example

```
Label0.Left = 100
```

(8) Name property

Name

Name

Define

String Name

Example

Get the default name of the specified object

VBScript Example

```
TextBox0.Text = Label0.Name
```

(9) Opacity property

Opacity

Opacity

Define

Double Opacity

Example

The graphics display translucent effect

VBScript Example

```
Label0.Opacity = 0.5
```

(10) RotateAngle property

RotateAngle

Rotation Angle

Define

Double RotateAngle

Example

Set the rotation Angle of the specified object to 90

VBScript Example

```
Label0.RotateAngle = 90
```

(11) Text property

Text

Gets or sets up the text content of the label

Define

String Text

Example

Set the text of the specified object to "Delta SCADA"

VBScript Example

```
Label0.Text= "Delta SCADA"
```

(12) ToolTip property

ToolTip

Tooltip text

Define

String ToolTip

Example

Set the ToolTip of the specified object to "test"

VBScript Example
Label0.ToolTip = "test"

(13) Top property

Top

The top coordinate

Define

Double Top

Example

Object's the top of the coordinate value is 100 in the picture

VBScript Example
Label0.Top = 100

(14) Width property

Width

Width

Define

Double Width

Example

Set the width of the specified object to 100

VBScript Example
Label0.Width = 100

(15) ZIndex property

ZIndex

Layer index

Define

Int32 ZIndex

Example

Set the Layer index of the specified object to 3

VBScript Example

```
Label0.ZIndex = 3
```

5. HMITextBox object

HMITextBox

Text box control

Methods list

	Name	Description
	FindAnimation	Look for animation to modify the associated variables of the animation
	Focus	Set the focus on the text box
	SelectedAll	Select all the contents of the text box

Property list

	Name	Description
	AcceptsReturn	Get or set a value, the value indicating whether the text content wrap
	Background	The background color
	BorderBrush	Border color
	FontSize	Font size
	Foreground	Foreground
	Height	Height
	IsEnabled	Get or set a value, the value indicating whether the control can operation
	IsShow	Whether display the specified object
	Left	The left coordinate
	Name	Name
	Opacity	Opacity
	RotateAngle	Rotation Angle
	Text	Get or the set up the text content of the text box
	ToolTip	Tooltip text

	Top	The top coordinate
	Width	Width
	ZIndex	Layer index

The following is the detailed description of the script:

【Method】

(1) FindAnimation method

FindAnimation

Look for animation to modify the associated variables of the animation

Define

FindAnimation(name)

Parameter

Name	Required/Optional	Data type	Description
name	Required	**String**	The name of animation

Example

Change the associated variables of the visibility animation of the object into Var.VariableGroup0.Variable1

VBScript Example
TextBox0.FindAnimation("HMIShowHideAnimation").Expression = "Var.VariableGroup0.Variable1"

(2) Focus method

Focus

Set the focus on the text box

Define

Focus()

Example

Set the focus on the TextBox0

VBScript Example

```
TextBox0.Focus()
```

(3) SelectedAll

SelectedAll

Select all the contents of the text box

Define

SelectAll()

Example

Select all the contents of the text box 0(Note: first, set the focus on the TextBox0)

VBScript Example

```
TextBox0.Focus()
TextBox0.SelectAll()
```

【property】

(1) AcceptsReturn

AcceptsReturn

Gets or sets a value, the value indicating whether the text content wrap

Define

Boolean AcceptsReturn

Example

True: the text content wrap False: the text content will not wrap

VBScript Example

```
TextBox0.AcceptsReturn = True TextBox0.AcceptsReturn = False
```

(2) Background property

Background

The background color（Colors.Red）

Define

Object Background

Example

Set the background color of the specified object

VBScript Example
TextBox0.Background = Colors.Red

(3) BorderBrush property

BorderBrush

The border color（Colors.Red）

Define

Object BorderBrush

Example

Set the border color of the specified object

VBScript Example
TextBox0.BorderBrush = Colors.Red

(4) FontSize property

FontSize

Font size

Define

Double FontSize

Example

Set the font size of the specified object

VBScript Example
TextBox0.FontSize = 20

(5) Foreground property

Foreground

The foreground color（Colors.Red）

Define

Object Foreground

Example

Set the foreground color of the specified object

VBScript Example
TextBox0.Foreground = Colors.Red

(6) Height property

Height

Height

Define

Double Height

Example

Set the height of the specified object to 100

VBScript Example
TextBox0.Height = 100

(7) IsEnabled property

IsEnabled

Gets or sets a value, the value indicating whether the control can operation

Define

Boolean IsEnabled

Example

True ：The specified object is Enabled

VBScript Example
TextBox0.IsEnabled = True

(8) IsShow property

IsShow

Whether display the specified object or not

Define

Boolean IsShow

Example

True ：The specified object is displayed on the screen False ：The specified object hidden on the screen

VBScript Example

```
TextBox0.IsShow = True TextBox0.IsShow = False
```

(9) Left property

Left

The left coordinate

Define

Double Left

Example

Object's left side of the coordinate values is 100 on the screen

VBScript Example

```
TextBox0.Left = 100
```

(10) Name property

Name

Name

Define

String Name

Example

Get the default name of the specified object

VBScript Example

```
TextBox1.Text = TextBox0.Name
```

(11) Opacity property

Opacity

Opacity

Define

Double Opacity

Example

Graphical display translucent effect

VBScript Example

```
TextBox0.Opacity = 0.5
```

(12) RotateAngle property

RotateAngle

Rotation Angle

Define

Double RotateAngle

Example

Set the rotation Angle of the specified object to 90

VBScript Example

```
TextBox0.RotateAngle = 90
```

(13) Text property

Text

Gets or sets up the text content of text box

Define

String Text

Example

Set the text of the specified object to "Delta SCADA"

VBScript Example
TextBox0.Text = "Delta SCADA"

(14) ToolTip property

ToolTip

Tooltip text

Define

String ToolTip

Example

Set the ToolTip of the specified object to "test"

VBScript Example
TextBox0.ToolTip = "test"

(15) Top property

Top

The top coordinate

Define

Double Top

Example

Object's the top of the coordinate value is 100 in the picture

VBScript Example
TextBox0.Top = 100

(16) Width property

Width

Width

Define

Double Width

Example

Set the width of the specified object to 100

VBScript Example
TextBox0.Width = 100

(17) ZIndex property

ZIndex

Layer index

Define

Int32 ZIndex

Example

Set the Layer index of the specified object to 3

VBScript Example
Text0.ZIndex = 3

6. HMIPasswordBox object

HMIPasswordBox

Password box control

Methods list

	Name	Description
	FindAnimation	Look for animation to modify the associated variables of the animation

Property list

	Name	Description
	Background	The background color
	BorderBrush	The border color
	FontSize	Font size
	Foreground	The foreground color
	Height	Height

	IsEnabled	Get or set a value, the value indicating whether the text content carriage returns
	IsShow	Whether display the specified object
	Left	The left coordinate
	Name	Name
	Opacity	Opacity
	Password	Get or set the password content, shown as a small dot
	RotateAngle	Rotation Angle
	ToolTip	Tooltip text
	Top	The top coordinate
	Width	Width
	ZIndex	Layer index

The following is the detailed description of the script:

【Method】

(1) FindAnimation method

FindAnimation

Look for animation to modify the associated variables of the animation

Define

FindAnimation(name)

Parameter

Name	Required/Optional	Data type	Description
name	Required	**String**	The name of animation

Example

Change the associated variables of the visibility animation of the object into Var.VariableGroup0.Variable1

VBScript Example

```
PasswordBox0.FindAnimation("HMIShowHideAnimation").Expression =
"Var.VariableGroup0.Variable1"
```

【property】

(1) Background property

Background

The background color（Colors.Red）

Define

Object Background

Example

Set the background color of the specified object

VBScript Example

```
PasswordBox0.Background = Colors.Red
```

(2) BorderBrush property

BorderBrush

The border color（Colors.Red）

Define

Object BorderBrush

Example

Set the border color of the specified object

VBScript Example

```
PasswordBox0.BorderBrush = Colors.Red
```

(3) FontSize property

FontSize

Font size

Define

Double FontSize

Example

Set the font size of the specified object

VBScript Example

```
PasswordBox0.FontSize = 20
```

(4) Foreground property

Foreground

The foreground color（Colors.Red）

Define

Object Foreground

Example

Set the foreground color of the specified object

VBScript Example

```
PasswordBox0.Foreground = Colors.Red
```

(5) Height property

Height

Height

Define

Double Height

Example

Set the height of the specified object to 100

VBScript Example

```
PasswordBox0.Height = 100
```

(6) IsEnabled property

IsEnabled

Gets or sets a value, the value indicating whether the control can operation

Define

Boolean IsEnabled

Example

True ：The specified object is Enabled

VBScript Example
PasswordBox0.IsEnabled = True

(7) IsShow property

IsShow

Whether display the specified object or not

Define

Boolean IsShow

Example

True ：The specified object is displayed on the screen False ：The specified object hidden on the screen

VBScript Example
PasswordBox0.IsShow = True PasswordBox0.IsShow = False

(8) Left property

Left

The left coordinate

Define

Double Left

Example

Object's left side of the coordinate values is 100 on the screen

VBScript Example
PasswordBox0.Left = 100

(9) Name property

Name

Name

Define

String Name

Example

Get the default name of the specified object

VBScript Example

```
TextBox0.Text = PasswordBox0.Name
```

(10) Opacity property

Opacity

Opacity

Define

Double Opacity

Example

The graphics display translucent effect

VBScript Example

```
PasswordBox0.Opacity = 0.5
```

(11) Password property

Password

Gets or sets the password content, shown as a small dot

Define

String Password

Example

Set the password content to “123456”

VBScript Example

```
PasswordBox0.Password = "123456"
```

(12) RotateAngle property

RotateAngle

Rotation Angle

Define

Double RotateAngle

Example

Set the rotation Angle of the specified object to 90

VBScript Example
PasswordBox0.RotateAngle = 90

(13) ToolTip property

ToolTip

Tooltip text

Define

String ToolTip

Example

Set the ToolTip of the specified object to "test"

VBScript Example
PasswordBox0.ToolTip = "test"

(14) Top property

Top

The top coordinate

Define

Double Top

Example

Object's the top of the coordinate value is 100 in the picture

VBScript Example

```
PasswordBox0.Top = 100
```

(15) Width property

Width

Width

Define

Double Width

Example

Set the width of the specified object to 100

VBScript Example

```
PasswordBox0.Width = 100
```

(16) ZIndex property

ZIndex

Layer index

Define

Int32 ZIndex

Example

Set the Layer index of the specified object to 3

VBScript Example

```
PasswordBox0.ZIndex = 3
```

7. HMIDateTimePicker object

HMIDateTimePicker

Date time picker control

Methods list

	Name	Description
	Compare	Date/time to compare
	CompareDate	Date to compare

	CompareTime	Time to compare
	FindAnimation	Look for animation to modify the associated variables of the animation
	GetTimeValue	Get the time value

Property list

	Name	Description
	Background	The background color
	FontSize	Font size
	Foreground	The foreground color
	Height	Height
	IntervalHours	Get hours interval
	IntervalMilliseconds	Get interval number of milliseconds
	IntervalMinutes	Get interval number of minutes
	IntervalSeconds	Get interval number of seconds
	IsEnabled	Get or set a value, the value indicating whether the control can operation
	IsShow	Whether display the specified object
	Left	The left coordinate
	Name	Name
	Opacity	Opacity
	RotateAngle	Rotation angle
	ToolTip	Tooltip text
	Top	The top coordinate
	Value	Get or set the display values of the calendar date time
	ValueTime	Get the display values of the calendar date time
	Width	Width
	ZIndex	Layer index

The following is the detailed description of the script:

【Method】

(1) Compare method

Compare

Date/Time comparison

Define

Compare(dateTime)

Parameter

Name	Required/Optional	Data Type	Description
dateTime	Required	**DateTime**	DataTime

Return

The return value is greater than 1, the date displayed by control is greater than the input date; less than zero, less than the input date，equal to zero is equal to the input date

Example

Call this script, compared the current date with the date displayed by calendar, if the return value is greater than zero, indicate that the date displayed by calendar is greater than the input date; equal to zero, indicate that the date displayed by calendar is equal to the input date; less than zero, indicate that the date displayed by calendar is less than the input date.

VBScript Example
DateTimePicker0.Compare(DateTime.Now)

(2) CompareDate method

CompareDate

Date/Time comparison

Define

CompareDate(dateTime)

Parameter

Name	Required/Optional	Data Type	Description
dateTime	Required	**DateTime**	Date/Time

Return

The return value is greater than 1, the date displayed by control is greater than the input date; less than zero, less than the input date，equal to zero is equal to the input date

Example

Call this script, compared the current date with the date displayed by calendar, if the return value is greater than zero, indicate that the date displayed by calendar is greater than the input date; equal to zero, indicate that the date displayed by calendar is equal to the input date; less than zero, indicate that the date displayed by calendar is less than the input date.

VBScript Example
DateTimePicker0.CompareDate(DateTime.Now)

(3) CompareTime method

CompareTime

Date/Time comparison

Define

CompareTime(dateTime)

Parameter

Name	Required/Optional	Data Type	Description
dateTime	Required	**DateTime**	Date/Time

Return

The return value is greater than 1, the date displayed by control is greater than the input date; less than zero, less than the input date，equal to zero is equal to the input date

Example

Call this script, compared the current date with the time displayed by calendar, if the return value is greater than zero, indicate that the time displayed by calendar is greater than the input time; equal to zero, indicate that the time displayed by calendar is equal to the input time; less than zero, indicate that the time displayed by calendar is less than the input time.

VBScript Example
DateTimePicker0.CompareTime(DateTime.Now)

(4) FindAnimation method

FindAnimation

Look for animation to modify the associated variables of the animation

Define

FindAnimation(name)

Parameter

Name	Required/Optional	Data type	Description
name	Required	**String**	The name of animation

Example

Change the associated variables of the visibility animation of the object into Var.VariableGroup0.Variable1

VBScript Example
DateTimePicker0.FindAnimation("HMIShowHideAnimation").Expression = "Var.VariableGroup0.Variable1"

(5) GetTimeValue

GetTimeValue

Get the time value

Define

GetTimeValue(name)

Parameter

Name	Required/Optional	Data Type	Description
name	Required	**String**	The name of the time

Return

Returns the time value

Example

Call this script to get the time value, the incoming value y - year, M -Month, d -Day, h - hour, m - minute, s - seconds, the following script returns value of the month.

VBScript Example
DateTimePicker0.GetTimeValue("M")

【property】

(1) Background property

Background

The background color（Colors.Red）

Define

Object Background

Example

Set the background color of the specified object

VBScript Example
DateTimePicker0.Background = Colors.Red

(2) FontSize property

FontSize

Font size

Define

Double FontSize

Example

Set the font size of the specified object

VBScript Example
DateTimePicker0.FontSize = 20

(3) Height property

Height

Height

Define

Double Height

Example

Set the height of the specified object to 100

VBScript Example

```
DateTimePicker0.Height = 100
```

(4) IntervalHours property

IntervalHours

Get interval hours

Define

Int32 IntervalHours

Example

Get interval hours to display in the text box

VBScript Example

```
TextBox0.Text = DateTimePicker0.IntervalHours
```

(5) IntervalMilliseconds property

IntervalMilliseconds

Get interval milliseconds

Define

Int32 IntervalMilliseconds

Example

Get interval milliseconds to display in the text box

VBScript Example

```
TextBox0.Text = DateTimePicker0.IntervalMilliseconds
```

(6) IntervalMinutes property

IntervalMinutes

Get interval minutes

Define

Int32 IntervalMinutes

Example

Get interval minutes to display in the text box

VBScript Example
TextBox0.Text = DateTimePicker0.IntervalMinutes

(7) IntervalSeconds property

IntervalSeconds

Get interval seconds

Define

Int32 IntervalSeconds

Example

Get interval seconds to display in the text box

VBScript Example
TextBox0.Text = DateTimePicker0.IntervalSeconds

(8) IsEnabled property

IsEnabled

Gets or sets a value, the value indicating whether the control can operation

Define

Boolean IsEnabled

Example

True ：The specified object is Enabled

VBScript Example

```
DateTimePicker0.IsEnabled = True
```

(9) IsShow property

IsShow

Whether display the specified object or not

Define

Boolean IsShow

Example

True ：The specified object is displayed on the screen False ：The specified object hidden on the screen

VBScript Example

```
DateTimePicker0.IsShow = True    DateTimePicker0.IsShow = False
```

(10) Left property

Left

The left coordinate

Define

Double Left

Example

Object's left side of the coordinate values is 100 on the screen

VBScript Example

```
DateTimePicker0.Left = 100
```

(11) Name property

Name

Name

Define

String Name

Example

Get the default name of the specified object

VBScript Example
TextBox0.Text = DateTimePicker0.Name

(12) Opacity property

Opacity

Opacity

Define

Double Opacity

Example

The graphics display translucent effect

VBScript Example
DateTimePicker0.Opacity = 0.5

(13) RotateAngle property

RotateAngle

Rotation Angle

Define

Double RotateAngle

Example

Set the rotation Angle of the specified object to 90

VBScript Example
DateTimePicker0.RotateAngle = 90

(14) ToolTip property

ToolTip

Tooltip text

Define

String ToolTip

Example

Set the ToolTip of the specified object to "test"

VBScript Example

```
DateTimePicker0.ToolTip = "test"
```

(15) Top property

Top

The top coordinate

Define

Double Top

Example

Object's the top of the coordinate value is 100 in the picture

VBScript Example

```
DateTimePicker0.Top = 100
```

(16) Value property

Value

Gets or sets values of time of calendar date displayed

Define

DateTime Value

Example

Get values of DataTimePicker0 to display in text box

VBScript Example

```
TextBox0.Text = DateTimePicker0.Value
```

(17) ValueTime property

ValueTime

Get values of time of calendar date displayed

Define

DateTime ValueTime

Example

Get values of DataTimePicker0 to display in text box

VBScript Example
TextBox0.Text = DateTimePicker0.ValueTime

(18) Width property

Width

Width

Define

Double Width

Example

Set the width of the specified object to 100

VBScript Example
DateTimePicker0.Width = 100

(19) ZIndex property

ZIndex

Layer index

Define

Int32 ZIndex

Example

Set the Layer index of the specified object to 3

VBScript Example
DateTimePicker0.ZIndex = 3

8. HMIDatePicker object

HMIDatePicker

Date control

Methods list

	Name	Description
	CompareDate	Comparison the date
	FindAnimation	Look for animation to modify the associated variables of the animation

Property list

	Name	Description
	Background	The background color
	BorderBrush	Border color
	FontSize	Font size
	Foreground	The foreground color
	Height	Height
	IsEnabled	Get or set a value, the value indicating whether the control can operation
	IsShow	Whether display the specified object
	Left	The left coordinate
	Name	Name
	Opacity	Opacity
	RotateAngle	Rotation Angle
	Text	Get or set the date string of the date displayed
	ToolTip	Tooltip text
	Top	The top coordinate
	ValueTime	Get the date value of the calendar
	Width	Width
	ZIndex	Layer index

The following is the detailed description of the script:

【Method】

(1) CompareDate method

CompareDate

Date comparison

Define

CompareDate(dateTime)

Parameter

Name	Required/Optional	Data Type	Description
dateTime	Required	**DateTime**	

Example

Call this script, compared the current date with the date displayed by calendar, if the return value is greater than zero, indicate that the date displayed by calendar is greater than the incoming date; equal to zero, indicate that the date displayed by calendar is equal to the incoming date; less than zero, indicate that the date displayed by calendar is less than the incoming date.

VBScript Example
DatePicker0.CompareDate(DateTime.Now)

(2) FindAnimation method

FindAnimation

Look for animation to modify the associated variables of the animation

Define

FindAnimation(name)

Parameter

Name	Required/Optional	Data type	Description
name	Required	**String**	The name of animation

Example

Change the associated variables of the visibility animation of the object into Var.VariableGroup0.Variable1

VBScript Example

```
DatePicker0.FindAnimation("HMIShowHideAnimation").Expression = "Var.VariableGroup0.Variable1"
```

【property】

(1) Background property

Background

The background color（Colors.Red）

Define

Object Background

Example

Set the background color of the specified object

VBScript Example

```
DatePicker0.Background = Colors.Red
```

(2) BorderBrush property

BorderBrush

The border color（Colors.Red）

Define

Object BorderBrush

Example

Set the border color of the specified object

VBScript Example

```
DatePicker0.BorderBrush = Colors.Red
```

(3) FontSize property

FontSize

The font size

Define

Double FontSize

Example

Set the font size of the specified object

VBScript Example
DatePicker0.FontSize = 20

(4) Foreground property

Foreground

The foreground color（Colors.Red）

Define

Object Foreground

Example

Set the foreground color of the specified object

VBScript Example
DatePicker0.Foreground = Colors.Red

(5) Height property

Height

Height

Define

Double Height

Example

Set the height of the specified object to 100

VBScript Example
DatePicker0.Height = 100

(6) IsEnabled property

IsEnabled

Gets or sets a value, the value indicating whether the control can operation

Define

Boolean IsEnabled

Example

The specified object is Enabled

VBScript Example
DatePicker0.IsEnabled = True

(7) IsShow property

IsShow

Whether display the specified object or not

Define

Boolean IsShow

Example

True ：The specified object is displayed on the screen False ：The specified object hidden on the screen

VBScript Example
DatePicker0.IsShow = True DatePicker0.IsShow = False

(8) Left property

Left

The left coordinate

Define

Double Left

Example

Object's left side of the coordinate values is 100 on the screen

VBScript Example
DatePicker0.Left = 100

(9) Name property

Name

Name

Define

String Name

Example

Get the default name of the specified object

VBScript Example
TextBox0.Text = DatePicker0.Name

(10) Opacity property

Opacity

Opacity

Define

Double Opacity

Example

The graphics display translucent effect

VBScript Example
DatePicker0.Opacity = 0.5

(11) RotateAngle property

RotateAngle

Rotation Angle

Define

Double RotateAngle

Example

Set the rotation Angle of the specified object to 90

VBScript Example
DatePicker0.RotateAngle = 90

(12) Text property

Text

Gets or sets up the string of the date displayed

Define

String Text

Example

VBScript Example

```
DatePicker0.Text = "2014/8/9"
```

(13) ToolTip property

ToolTip

Tooltip text

Define

String ToolTip

Example

Set the ToolTip of the specified object to "test"

VBScript Example

```
DatePicker0.ToolTip = "test"
```

(14) Top property

Top

The top coordinate

Define

Double Top

Example

Object's the top of the coordinate value is 100 in the picture

VBScript Example

```
DatePicker0.Top = 100
```

(15) ValueTime property

ValueTime

Get values of calendar date displayed

Define

DateTime ValueTime

Example

VBScript Example
TextBox0.Text = DatePicker0.ValueTime

(16) Width property

Width

Width

Define

Double Width

Example

Set the width of the specified object to 100

VBScript Example
DatePicker0.Width = 100

(17) ZIndex property

ZIndex

Layer index

Define

Int32 ZIndex

Example

Set the Layer index of the specified object to 3

VBScript Example
DatePicker0.ZIndex = 3

9. HMICalendar object

HMICalendar

Calendar control

Methods list

	Name	Description
	FindAnimation	Look for animation to modify the associated variables of the animation

Property list

	Name	Description
	Background	The background color
	BorderBrush	The border color
	Foreground	The foreground color
	IsEnabled	Get or set a value, the value indicating whether the control can operation
	IsShow	Whether display the specified object
	Left	The left coordinate
	Name	Name
	Opacity	Opacity
	ToolTip	Tooltip text
	Top	The top coordinate
	ZIndex	Layer index

The following is the detailed description of the script:

【Method】

(1) FindAnimation method

FindAnimation

Look for animation to modify the associated variables of the animation

Define

FindAnimation(name)

Parameter

Name	Required/Optional	Data type	Description
name	Required	**String**	The name of animation

Example

Change the associated variables of the visibility animation of the object into Var.VariableGroup0.Variable1

VBScript Example
Calendar0.FindAnimation("HMIShowHideAnimation").Expression = "Var.VariableGroup0.Variable1"

【property】

(1) Background property

Background

The background color（Colors.Red）

Define

Object Background

Example

Set the background color of the specified object

VBScript Example
Calendar0.Background = Colors.Red

(2) BorderBrush property

BorderBrush

The border color（Colors.Red）

Define

Object BorderBrush

Example

Set the border color of the specified object

VBScript Example

```
Calendar0.BorderBrush = Colors.Red
```

(3) Foreground property

Foreground

The foreground color（Colors.Red）

Define

Object Foreground

Example

Set the foreground color of the specified object

VBScript Example

```
Calendar0.Foreground = Colors.Red
```

(4) IsEnabled property

IsEnabled

Gets or sets a value, the value indicating whether the control can operation

Define

Boolean IsEnabled

Example

The specified object is Enabled

VBScript Example

```
Calendar0.IsEnabled = True
```

(5) IsShow property

IsShow

Whether display the specified object or not

Define

Boolean IsShow

Example

True ：The specified object is displayed on the screen False ：The specified object hidden on the screen

VBScript Example
Calendar0.IsShow = True Calendar0.IsShow = False

(6) Left property

Left

The left coordinate

Define

Double Left

Example

Object's left side of the coordinate values is 100 on the screen

VBScript Example
Calendar0.Left = 100

(7) Name property

Name

Name

Define

String Name

Example

Get the default name of the specified object

VBScript Example
TextBox0.Text = Calendar0.Name

(8) Opacity property

Opacity

Opacity

Define

Double Opacity

Example

The graphics display translucent effect

VBScript Example
Calendar0.Opacity = 0.5

(9) ToolTip property

ToolTip

Tooltip text

Define

String ToolTip

Example

Set the ToolTip of the specified object to "test"

VBScript Example
Calendar0.ToolTip = "test"

(10) Top property

Top

The top coordinate

Define

Double Top

Example

Object's the top of the coordinate value is 100 in the picture

VBScript Example
Calendar0.Top = 100

(11) ZIndex property

ZIndex

Layer index

Define

Int32 ZIndex

Example

Set the Layer index of the specified object to 3

VBScript Example

```
Calendar0.ZIndex = 3
```

10. HMIImage object

HMIImage

Image control

Methods list

	Name	Description
	FindAnimation	Look for animation to modify the associated variables of the animation

Property list

	Name	Description
	Height	Height
	IsShow	Whether display the specified object
	Left	The left coordinate
	Name	Name
	Opacity	Opacity
	RotateAngle	Rotation angle
	SwitchingPath	Switch the image path
	ToolTip	Tooltip text
	Top	The top coordinate
	Width	Width
	ZIndex	Layer index

The following is the detailed description of the script:

【Method】

(1) FindAnimation method

FindAnimation

Look for animation to modify the associated variables of the animation

Define

FindAnimation(name)

Parameter

Name	Required/Optional	Data type	Description
name	Required	**String**	The name of animation

Example

Change the associated variables of the visibility animation of the object into Var.VariableGroup0.Variable1

VBScript Example

```
Image0.FindAnimation("HMIShowHideAnimation").Expression = "Var.VariableGroup0.Variable1"
```

【property】

(1) Height property

Height

Height

Define

Double Height

Example

Set the height of the specified object to 100

VBScript Example

```
Image0.Height = 100
```

(2) IsShow property

IsShow

Whether display the specified object or not

Define

Boolean IsShow

Example

True ：The specified object is displayed on the screen False ：The specified object hidden on the screen

VBScript Example
Image0.IsShow = True Image0 = False

(3) Left property

Left

The left coordinate

Define

Double Left

Example

Object's left side of the coordinate values is 100 on the screen

VBScript Example
Image0.Left = 100

(4) Name property

Name

Name

Define

String Name

Example

Get the default name of the specified object

VBScript Example
TextBox0.Text = Image0.Name

(5) Opacity property

Opacity

Opacity

Define

Double Opacity

Example

The graphics displays translucent effect

VBScript Example

```
Image0.Opacity = 0.5
```

(6) RotateAngle property

RotateAngle

Rotation Angle

Define

Double RotateAngle

Example

Set the rotation Angle of the specified object to 90

VBScript Example

```
Image0.RotateAngle = 90
```

(7) SwitchingPath property

SwitchingPath

Switch the image path

Define

String SwitchingPath

Example

Switch the image path

VBScript Example

```
Image0.SwitchingPath = "C:\test.jpg"
```

(8) ToolTip property

ToolTip

Tooltip text

Define

String ToolTip

Example

Set the ToolTip of the specified object to "test"

VBScript Example

```
Image0.ToolTip = "test"
```

(9) Top property

Top

The top coordinate

Define

Double Top

Example

Object's the top of the coordinate value is 100 in the picture

VBScript Example

```
Image0.Top = 100
```

(10) Width property

Width

Width

Define

Double Width

Example

Set the width of the specified object to 100

VBScript Example

```
Image0.Width = 100
```

(11) ZIndex property

ZIndex

Layer index

Define

Int32 ZIndex

Example

Set the Layer index of the specified object to 3

VBScript Example

```
Image0.ZIndex = 3
```

11. HMIGIFImage object

HMIGIFImage

GIF image control

Methods list

	Name	Description
	FindAnimation	Look for animation to modify the associated variables of the animation
	StartAnimate	Start animation
	StopAnimate	Stop animation

Property list

	Name	Description
	Height	Height
	IsEnabled	whether can operation
	IsShow	Whether display the specified object
	Left	The left coordinate
	Name	Name
	Opacity	Opacity
	RotateAngle	Rotation angle

	ToolTip	Tooltip text
	Top	The top coordinate
	Width	Width
	ZIndex	Layer index

The following is the detailed description of the script:

【Method】

(1) FindAnimation method

FindAnimation

Look for animation to modify the associated variables of the animation

Define

FindAnimation(name)

Parameter

Name	Required/Optional	Data type	Description
name	Required	**String**	The name of animation

Example

Change the associated variables of the visibility animation of the object into Var.VariableGroup0.Variable1

VBScript Example

```
GIFImage0.FindAnimation("HMIShowHideAnimation").Expression = "Var.VariableGroup0.Variable1"
```

(2) StartAnimate method

StartAnimate

Start animation

Define

StartAnimate()

Example

Execute the script will start animation of GIF images

VBScript Example

```
Call GIFImage0.StartAnimate()
```

(3) StopAnimate method

StopAnimate

Stop animation

Define

StopAnimate()

Example

Execute the script will stop animation of GIF images

VBScript Example

```
Call GIFImage0.StopAnimate()
```

【property】

(1) Height property

Height

Height

Define

Double Height

Example

Set the height of the specified object to 100

VBScript Example

```
GIFImage0.Height = 100
```

(2) IsEnabled property

IsEnabled

Gets or sets a value, the value indicating whether the control can operation

Define

Boolean IsEnabled

Example

The specified object is Enabled

VBScript Example
GIFImage0.IsEnabled = True

(3) IsShow property

IsShow

Whether display the specified object or not

Define

Boolean IsShow

Example

True ：The specified object is displayed on the screen False ：The specified object hidden on the screen

VBScript Example
GIFImage0.IsShow = True GIFImage0.IsShow = False

(4) Left property

Left

The left coordinate

Define

Double Left

Example

Object's left side of the coordinate values is 100 on the screen

VBScript Example
GIFImage0.Left = 100

(5) Name property

Name

Name

Define

String Name

Example

Get the default name of the specified object

VBScript Example
TextBox0.Text = GIFImage0.Name

(6) Opacity property

Opacity

Opacity

Define

Double Opacity

Example

The graphics displays translucent effect

VBScript Example
GIFImage0.Opacity = 0.5

(7) RotateAngle property

RotateAngle

Rotation Angle

Define

Double RotateAngle

Example

Set the rotation Angle of the specified object to 90

VBScript Example
GIFImage0.RotateAngle = 90

(8) ToolTip property

ToolTip

Tooltip text

Define

String ToolTip

Example

Set the ToolTip of the specified object to "test"

VBScript Example

```
GIFImage0.ToolTip = "test"
```

(9) Top property

Top

The top coordinate

Define

Double Top

Example

Object's the top of the coordinate value is 100 in the picture

VBScript Example

```
GIFImage0.Top = 100
```

(10) Width property

Width

Width

Define

Double Width

Example

Set the width of the specified object to 100

VBScript Example

```
GIFImage0.Width = 100
```

(11) ZIndex property

ZIndex

Layer index

Define

Int32 ZIndex

Example

Set the Layer index of the specified object to 3

VBScript Example

```
GIFImage0.ZIndex = 3
```

12. HMINixieTube object

HMINixieTube

Nixie tube control

Methods list

	Name	Description
	FindAnimation	Look for animation to modify the associated variables of the animation

Property list

	Name	Description
	Background	The background color
	Bit	Digital tube integer number
	IsEnabled	Get or set a value, the value indicating whether the control can operation
	IsShow	Whether display the specified object
	Left	The left coordinate
	Name	Name
	Number	The numeric value displayed
	Opacity	Opacity
	RotateAngle	Rotation angle
	ToolTip	Tooltip text
	Top	The top coordinate

	ZIndex	Layer index

The following is the detailed description of the script:

【Method】

(1) FindAnimation method

FindAnimation

Look for animation to modify the associated variables of the animation

Define

FindAnimation(name)

Parameter

Name	Required/Optional	Data type	Description
name	Required	**String**	The name of animation

Example

Change the associated variables of the visibility animation of the object into Var.VariableGroup0.Variable1

VBScript Example

```
NixieTube0.FindAnimation("HMIShowHideAnimation").Expression = "Var.VariableGroup0.Variable1"
```

【property】

(1) Background property

Background

The background color（Colors.Red）

Define

Object Background

Example

Set the background color of the specified object

VBScript Example

```
NixieTube0.Background = Colors.Red
```

(2) Bit property

Bit

Digital tube's integer number

Define

int Bit

Example

Set the integer number to 3

VBScript Example

```
NixieTube0.Bit = 3
```

(3) IsEnabled property

IsEnabled

Gets or sets a value, the value indicating whether the control can operation

Define

Boolean IsEnabled

Example

The specified object is Enabled

VBScript Example

```
NixieTube0.IsEnabled = True
```

(4) IsShow property

IsShow

Whether display the specified object

Define

Boolean IsShow

Example

True ：The specified object is displayed on the screen False ：The specified object hidden on the screen

VBScript Example
NixieTube0.IsShow = True NixieTube0.IsShow = False

(5) Left property

Left

The left coordinate

Define

Double Left

Example

Object's abscissa value of the left side in the picture is 100

VBScript Example
NixieTube0.Left = 100

(6) Name property

Name

Name

Define

String Name

Example

Get the default name of the specified object

VBScript Example
TextBox0.Text = NixieTube0.Name

(7) Number property

Number

The number displayed by Digital tube

Define

double Number

Example

Set the number of the specified object to 1.23

VBScript Example
NixieTube0.Number = 1.23

(8) Opacity property

Opacity

Opacity

Define

Double Opacity

Example

The graphics display translucent effect

VBScript Example
NixieTube0.Opacity = 0.5

(9) RotateAngle property

RotateAngle

Rotation Angle

Define

Double RotateAngle

Example

Sets the rotation Angle of the specified object to 90

VBScript Example
NixieTube0.RotateAngle = 90

(10) ToolTip property

ToolTip

Tooltip text

Define

String ToolTip

Example

Set the ToolTip of the specified object to "test"

VBScript Example

```
NixieTube0.ToolTip = "test"
```

(11) Top property

Top

The top coordinate

Define

Double Top

Example

Objects in the picture on the top of the y coordinate value is 100

VBScript Example

```
NixieTube0.Top = 100
```

(12) ZIndex property

ZIndex

Layer index

Define

Int32 ZIndex

Example

Set the Layer index of the specified object to 3

VBScript Example

```
NixieTube0.ZIndex = 3
```

13. HMICurvedRuler object

HMICurvedRuler

Curved ruler control

Methods list

	Name	Description
	FindAnimation	Look for animation to modify the associated variables of the animation

Property list

	Name	Description
	BigTickNumber	The number of big tick
	EndOffset	The end value of the ruler
	IsArcVisible	Whether display the arc or not
	IsEnabled	Get or set a value, the value indicating whether the control can operation
	IsShow	Whether display the specified object or not
	IsTextVisible	Whether display the text or not
	Left	The left coordinate
	LineBrush	The current color of the ruler
	Name	Name
	Opacity	Opacity
	RotateAngle	Rotation angle
	ScanAngle	Scan angle
	SmallTickNumber	The number of small tick
	StartAngle	Start angle
	StartOffset	The start value of the ruler
	TickHeight	Height
	ToolTip	Tooltip text
	Top	The top coordinate
	ZIndex	Layer index

The following is the detailed description of the script:

【Method】

(1) FindAnimation method

FindAnimation

Look for animation to modify the associated variables of the animation

Define

FindAnimation(name)

Parameter

Name	Required/Optional	Data type	Description
name	Required	**String**	The name of animation

Example

Change the associated variables of the visibility animation of the object into Var.VariableGroup0.Variable1

VBScript Example

```
CurvedRuler0.FindAnimation("HMIShowHideAnimation").Expression =
"Var.VariableGroup0.Variable1"
```

【property】

(1) BigTickNumber property

BigTickNumber

Gets or sets the number of big scales on the CurvedRuler

Define

Int BigTickNumber

Example

Set the number of the big scale to 10

VBScript Example

```
CurvedRuler0.BigTickNumber = 10
```

(2) EndOffset property

EndOffset

Gets or sets the end value of the CurvedRuler

Define

Double EndOffset

Example

Set the end value of the ruler to 110

VBScript Example

```
CurvedRuler0.EndOffset = 110
```

(3) IsArcVisible property

IsArcVisible

Whether display the arc or not

Define

Boolean IsArcVisible

Example

True：display the arc　　False：hidden the arc

VBScript Example

```
CurvedRuler0..IsArcVisible = True　　CurvedRuler0..IsArcVisible = False
```

(4) IsEnabled property

IsEnabled

Get or set a value, the value indicating whether the control can operation

Define

Boolean IsEnabled

Example

The specified object is Enabled

VBScript Example

```
CurvedRuler0.IsEnabled = True
```

(5) IsShow property

IsShow

Whether display the specified object or not

Define

Boolean IsShow

Example

True ：The specified object is displayed on the screen False ：The specified object hidden on the screen

VBScript Example

```
CurvedRuler0.IsShow = True    CurvedRuler0.IsShow = False
```

(6) IsTextVisible property

IsTextVisible

Whether display the text or not

Define

Boolean IsTextVisible

Example

True：display text False：hidden text

VBScript Example

```
CurvedRuler0.IsTextVisible = True    CurvedRuler0.IsTextVisible = False
```

(7) Left property

Left

The left coordinate

Define

Double Left

Example

Object's left side of the coordinate values is 100 on the screen

VBScript Example
CurvedRuler0.Left = 100

(8) LineBrush property

LineBrush

Gets or sets the current color of ruler（Colors.red）

Define

Object LineBrush

Example

Set the current color of the specified object

VBScript Example
CurvedRuler0.LineBrush = Colors.red

(9) Name property

Name

Name

Define

String Name

Example

Get the default name of the specified object

VBScript Example
TextBox0.Text = CurvedRuler0.Name

(10) Opacity property

Opacity

Opacity

Define

Double Opacity

Example

The graphics display translucent effect

VBScript Example
CurvedRuler0.Opacity = 0.5

(11) RotateAngle property

RotateAngle

Rotation Angle

Define

Double RotateAngle

Example

Sets the rotation Angle of the specified object to 90

VBScript Example
CurvedRuler0.RotateAngle = 90

(12) ScanAngle property

ScanAngle

Gets or sets the scanning angle of the scale

Define

Double ScanAngle

Example

Sets the scan Angle of the specified object to 180

VBScript Example
CurvedRuler0.ScanAngle = 180

(13) SmallTickNumber property

SmallTickNumber

Gets or sets the number of small scales on the CurvedRuler

Define

Int SmallTickNumber

Example

Set the number of the small scale to 5

VBScript Example

```
CurvedRuler0.SmallTickNumber = 5
```

(14) StartAngle property

StartAngle

Start angle

Define

Double StartAngle

Example

The Start Angle value is 90 of the circular arc

VBScript Example

```
CurvedRuler0.StartAngle = 90
```

(15) StartOffset property

StartOffset

Gets or sets the start value of the CurvedRuler

Define

Double StartOffset

Example

Set the start value of the ruler to 10

VBScript Example

```
CurvedRuler0.StartOffset = 10
```

(16) TickHeight property

TickHeight

Gets or sets the height of the scale

Define

Double TickHeight

Example

Set the height of the scale to 10

VBScript Example

```
CurvedRuler0.TickHeight = 10
```

(17) ToolTip property

ToolTip

Tooltip text

Define

String ToolTip

Example

Set the ToolTip of the specified object to "test"

VBScript Example

```
CurvedRuler0.ToolTip = "test"
```

(18) Top property

Top

The top coordinate

Define

Double Top

Example

Object's the top of the coordinate value is 100 in the picture

VBScript Example

```
CurvedRuler0.Top = 100
```

(19) ZIndex property

ZIndex

Layer index

Define

Int32 ZIndex

Example

Set the Layer index of the specified object to 3

VBScript Example

```
CurvedRuler0.ZIndex = 3
```

20.3.1.3 Extended controls

1. HMIRealTimeChart object

HMIRealTimeChart

Real time chart control

Methods list

	Name	Description
	AddNewLimitLine	Add a new limit line
	ArrayToString	Image array translate into string
	DeletevalueAll	Delete all data points
	Export	Export data
	ExportRealData	Export excel
	FindAnimation	Look for animation to modify the associated variables of the animation
	HiddenSeries	Hidden curve
	OpenShowFileDialog	Get saved path selected by dialog box
	Print	Print
	RemoveAllLimitLine	Remove all the limit line
	Save	Save

	Save	Save as pictures
	SetSeriesVariablePath	Curve correlation variable substitution
	SetTimeRangeWithRefresh	Set the time interval and refresh interval
	StartTimer	Start timer
	StopTimer	Stop timer
	VisiableSeries	Display curves

Property list

	Name	Description
	Height	Height
	IsShow	Whether display specified object
	IsShowBtnCrossLine	Whether display the positioning line button
	IsShowBtnLeftMost	Whether display the most left button
	IsShowBtnLeftShift	Whether display the left shift button
	IsShowBtnReSet	Whether display the reset button
	IsShowBtnRightMost	Whether display the most right button
	IsShowBtnRigthShift	Whether display the right shift button
	IsShowRealBtnConfig	Whether display the curve configuration button
	IsShowRealBtnPrint	Whether display the print button
	IsShowRealBtnSave	Whether display the save button
	IsShowRealBtnStop	Whether display the began or pause button
	IsShowStopTime	Whether display the pause time button
	Left	The left coordinate
	Name	Name
	NumberInterval	Get or set the maximum scale of numerical axis
	NumberSmallInterval	Get or set the minimum scale of numerical axis
	NumericalAxisMaximum	Get or set the maximum scale of numerical axis
	NumericalAxisMinimum	Get or set the minimum scale of numerical axis
	ToolTip	Tooltip text
	Top	The top coordinate
	Width	Width
	ZIndex	Layer index

The following is the detailed description of the script:

【Method】

(1) AddNewLimitLine method

AddNewLimitLine

Add new limit line (The "UpperAndLowerDisplay" is checked in the control property)

Define

AddNewLimitLine(variablePath ,defaultValue ,color , thickness)

Parameter

Name	Required/Optional	Data Type	Description
variablePath	Required	**String**	Variable path
defaultValue	Required	**Double**	Default value (default)
color	Required	**Brush**	Line fill color
thickness	Required	**Double**	Line thicknesses

Example

Add a new limit line

VBScript Example
Call HistoryChart0.AddNewLimitLine("Var.variable",0,Colors.Red,5)

(2) ArrayToString method

ArrayToString

Image array translate into string

Define

ArrayToString(seriesName)

Parameter

Name	Required/Optional	Data Type	Description
seriesName	Required	**String**	Name of the curve, separate by commas

Example

Export curve image into the report

The operation steps are as follows:

a.Create a RealTimeChart0 in the screen and associate variables that need to be viewed.

b.Create a new report template "Report0", select a cell to associate real-time variables and refresh time.

c.In the report template, the menu bar --> Realtime Variable --> System --> Image, to add the corresponding number of curve pictures.

d.Add a button to the created screen and write in the script (the following two different examples):

1. The two diferent curve pictures from two different realtimechart are exported to the same report

（"D:\Test.xlsx"is the save path of Excel，"report0"is the report template，"RealtimeChart0.ArrayToString("Series0")" is the pictures set）

VBScript Example

```
Call ReportCmd.ExportDataByTemplate("D:\Test.xlsx","Report0",RealtimeChart0.ArrayToString("Series0")&"|"&Realtime
```

2. The two diferent curve pictures from one realtimechart are exported to the same report

VBScript Example

```
Call ReportCmd.ExportDataByTemplate("D:\Test.xlsx","Report0",RealtimeChart0.ArrayToString("Series0")&"|"&Realtime
```

(3) DeletevalueAll method

DeletevalueAll

Delete all data points

Define

DeletevalueAll()

Example

Delete all data points

VBScript Example

```
Call RealtimeChart0.DeletevalueAll()
```

(4) Export method

Export

Export data

Define

Export(seriesName ,templateName)

Parameter

Name	Required/Optional	Data Type	Description
seriesName	Required	**String**	Name of the curve
templateName	Required	**String**	Report template

Example

Export a curve image into the report

VBScript Example
Call RealtimeChart0.Export("Series0;Series1","Report0")

(5) ExportRealData method

ExportRealData

Export excel

Define

ExportRealData(seriesName)

Parameter

Name	Required/Optional	Data Type	Description
seriesName	Required	**String**	Name of the curve

Example

Export excel

VBScript Example
Call RealtimeChart0.ExportRealData("Series0")

(6) FindAnimation method

FindAnimation

Look for animation to modify the associated variables of the animation

Define

FindAnimation(name)

Parameter

Name	Required/Optional	Data type	Description
name	Required	**String**	The name of animation

Example

Change the associated variables of the visibility animation of the object into Var.VariableGroup0.Variable1

VBScript Example

```
RealtimeChart0.FindAnimation("HMIShowHideAnimation").Expression =
"Var.VariableGroup0.Variable1"
```

(7) HiddenSeries method

HiddenSeries

Hide the curve

Define

HiddenSeries(seriesName)

Parameter

Name	Required/Optional	Data Type	Description
seriesName	Required	**String**	Name of the curve, separate by commas

Example

Hide the curve

VBScript Example

```
Call RealtimeChart0.HiddenSeries("Series0")
```

(8) OpenShowFileDialog method

OpenShowFileDialog

Get saved path which selected by dialog box

Define

OpenShowFileDialog()

Example

Get saved path which selected by dialog box

VBScript Example

```
tt = RealtimeChart0.OpenShowFileDialog()
Call
ReportCmd.ExportDataByTemplate(tt,"Report0",RealtimeChart0.ArrayToString("Series0")&"|"&RealtimeChart0.Arra
```

(9) Print method

Print

Print

Define

Print()

Example

Print

VBScript Example

```
Call RealtimeChart0.Print()
```

(10) RemoveAllLimitLine method

RemoveAllLimitLine

Remove all the limit line

Define

RemoveAllLimitLine()

Example

Remove all the limit line

VBScript Example
Call RealtimeChart0.RemoveAllLimitLine()

(11) Save method

Save

Save

Define

Save()

Example

Save

VBScript Example
Call RealtimeChart0.save()

(12) Save method

Save

Save as pictures

Define

Save(path ， seriesName)

Parameter

Name	Required/Optional	Data Type	Description
path	Required	**String**	An absolute path
seriesName	Required	**String**	Name of curve, separate by commas

Example

Save

VBScript Example
Call RealtimeChart0.save("D:\test.jpg","Series0")

(13) SetSeriesVariablePath method

SetSeriesVariablePath

Curve correlation variable substitution

Define

SetSeriesVariablePath(pathList)

Parameter

Name	Required/Optional	Data Type	Description
pathList	Required	**String**	The new variable path of the variables that replace the old variables , separate by commas

Example

Curve correlation variable substitution

VBScript Example
Call RealtimeChart0.SetSeriesVariablePath("Var.variable0,Var.variable1")

(14) SetTimeRangeWithRefresh method

SetTimeRangeWithRefresh

Set the time interval and refresh time

Define

SetTimeRangeWithRefresh(rangeH ,rangeM ,rangeS ,refreshH , refreshM ,refreshS)

Parameter

Name	Required/Optional	Data Type	Description
rangeH	Required	**Int**	Time interval，hour
rangeM	Required	**Int**	Time interval，minute

rangeS	Required	**Int**	Time interval，second
refreshH	Required	**Int**	Refresh，hour
refreshM	Required	**Int**	Refresh，minute
refreshS	Required	**Int**	Refresh，second

Example

Set the time interval and refresh time

VBScript Example

```
Call RealtimeChart0.SetTimeRangeWithRefresh(0,0,1,0,0,1)
```

(15) StartTimer method

StartTimer

Start timer

Define

StartTimer()

Example

Start Timer

VBScript Example

```
Call RealtimeChart0.StartTimer()
```

(16) StopTimer method

StopTimer

Stop timer

Define

StopTimer()

Example

Stop Timer

VBScript Example

```
Call RealtimeChart0.StopTimer()
```

(17) VisiableSeries method

VisiableSeries

Display the curve

Define

VisiableSeries(seriesName)

Parameter

Name	Required/Optional	Data Type	Description
seriesName	Required	**String**	Name of the curve, separate by commas

Example

Display the curve

VBScript Example

```
Call RealtimeChart0.VisiableSeries("Series0")
```

【property】

(1) Height property

Height

Height

Define

Double Height

Example

Set the height of the specified object to 100

VBScript Example

```
RealtimeChart0.Height = 100
```

(2) IsShow property

IsShow

Whether display the specified object

Define

Boolean IsShow

Example

True ：The specified object is displayed on the screen False ：The specified object hidden on the screen

VBScript Example

```
RealtimeChart0.IsShow = True    RealtimeChart0.IsShow = False
```

(3) IsShowBtnCrossLine property

IsShowBtnCrossLine

Whether display the positioning line button

Define

Boolean IsShowBtnCrossLine

Example

Display the positioning line

VBScript Example

```
RealtimeChart0.IsShowBtnCrossLine = True
```

(4) IsShowBtnLeftMost property

IsShowBtnLeftMost

Whether display the most left button

Define

Boolean IsShowBtnLeftMost

Example

Display the most left button

VBScript Example

```
RealtimeChart0.IsShowBtnLeftMost = True
```

(5) IsShowBtnLeftShift property

IsShowBtnLeftShift

Whether display the left shift button

Define

Boolean IsShowBtnLeftShift

Example

Display moves to the left button

VBScript Example

```
RealtimeChart0.IsShowBtnLeftShift = True
```

(6) IsShowBtnReSet property

IsShowBtnReSet

Whether display the reset button

Define

Boolean IsShowBtnReSet

Example

Display the reset button

VBScript Example

```
RealtimeChart0.IsShowBtnReSet = True
```

(7) IsShowBtnRightMost property

IsShowBtnRightMost

Whether display the most right button

Define

Boolean IsShowBtnRightMost

Example

Display the most right button

VBScript Example

```
RealtimeChart0.IsShowBtnRightMost = True
```

(8) IsShowBtnShiftRigthShift property

IsShowBtnShiftRigthShift

Whether display the right shift button

Define

Boolean IsShowBtnRigthShift

Example

Display button to the right

VBScript Example

```
RealtimeChart0.IsShowBtnRigthShift = True
```

(9) IsShowRealBtnConfig property

IsShowRealBtnConfig

Whether display the curve configuration button

Define

Boolean IsShowRealBtnConfig

Example

Display curve configuration button

VBScript Example

```
RealtimeChart0.IsShowRealBtnConfig = True
```

(10) IsShowRealBtnPrint property

IsShowRealBtnPrint

Whether display the print button

Define

Boolean IsShowRealBtnPrint

Example

Display print button

VBScript Example
RealtimeChart0.IsShowRealBtnPrint = True

(11) IsShowRealBtnSave property

IsShowRealBtnSave

Whether display the save button

Define

Boolean IsShowRealBtnSave

Example

Display the save button

VBScript Example
RealtimeChart0.IsShowRealBtnSave = True

(12) IsShowRealBtnStop property

IsShowRealBtnStop

Whether display the begin or pause button

Define

Boolean IsShowRealBtnStop

Example

Display the begin or pause button

VBScript Example
RealtimeChart0.IsShowRealBtnStop = True

(13) IsShowStopTime property

IsShowStopTime

Whether display the pause time button

Define

Boolean IsShowStopTime

Example

Display the pause time button

VBScript Example

```
RealtimeChart0.IsShowStopTime = True
```

(14) Left property

Left

The left coordinate

Define

Double Left

Example

Object's left side of the coordinate values is 100 on the screen

VBScript Example

```
RealtimeChart0.Left = 100
```

(15) Name property

Name

Name

Define

String Name

Example

Get the default name of the specified object

VBScript Example

```
TextBox0.Text = RealtimeChart0.Name
```

(16) NumberInterval property

NumberInterval

Get or set the maximum scale of numerical axis

Define

Double NumberInterval

Example

Set the maximum scale of numerical axis to 20

VBScript Example

```
RealTimeChart0.NumberInterval = 20
```

(17) NumberSmallInterval property

NumberSmallInterval

Get or set the minimum scale of numerical axis

Define

Double NumberSmallInterval

Example

Set the minimum scale of numerical axis to 20

VBScript Example

```
RealTimeChart0.NumberSmallInterval = 4
```

(18) NumericalAxisMaximum property

NumericalAxisMaximum

Get or set the numerical axis maximum

Define

Double NumericalAxisMaximum

Example

Set the numerical axis maximum to 120

VBScript Example

```
RealTimeChart0.NumberAxisMaximum = 120
```

(19) NumericalAxisMinimum property

NumericalAxisMinimum

Get or set the numerical axis minimum

Define

Double NumericalAxisMinimum

Example

Set the numerical axis minimum to 20

VBScript Example
RealTimeChart0.NumberAxisMinumum = 20

(20) ToolTip property

ToolTip

Tooltip text

Define

String ToolTip

Example

Set the ToolTip of the specified object to "test"

VBScript Example
RealtimeChart0.ToolTip = "test"

(21) Top property

Top

The top coordinate

Define

Double Top

Example

Object's the top of the coordinate value is 100 in the picture

VBScript Example

```
RealtimeChart0.Top = 100
```

(22) Width property

Width

Width

Define

Double Width

Example

Set the width of the specified object to 100

VBScript Example

```
RealtimeChart0.Width = 100
```

(23) ZIndex property

ZIndex

Layer index

Define

Int32 ZIndex

Example

Set the Zindex of the specified object to 3

VBScript Example

```
RealtimeChart0.ZIndex = 3
```

2. HMIHistoryChart object

HMIHistoryChart

History chart control

Methods list

	Name	Description
	AddNewLimitLine	Add new limit line

	ArrayToString	Image array translate into string
	Export	Export data
	Export	Export excel
	FindAnimation	Look for animation to modify the associated variables of the animation
	FirstAction	The most left
	GetDataTable	Store data to data table
	HiddenSeries	Hidden curve
	Import	Import excel
	LastAction	The most right
	NextAction	Right shift
	OpenShowFileDialog	Get the path of save dialog select
	PreAction	Left shift
	Print	Print
	QueryEndTime	Set the end time of the query data
	QueryHistoryDate	Query of the historical data
	QueryHistoryDate	Query of the historical data(with parameters)
	QueryInveral	Set the query interval
	QueryStartTime	Set the start time of the query data
	QueryUnit	Set the query unit
	RemoveAllLimitLine	Remove all the limit line
	Save	Save
	Save	Save as pictures
	SetSeriesVariablePath	Curve correlation variable substitution
	SetVarRecordRulerName	Set the name of query rule
	VisiableSeries	Display curve

Property list

	Name	Description
	Height	Height
	IsShow	Whether display the specified object
	IsShowBtnConfig	Whether display the curve configuration button
	IsShowBtnCrossLine	Whether display the positioning line
	IsShowBtnMenu	Whether display the import and export button
	IsShowBtnPrint	Whether display the print button

	IsShowBtnSave	Whether display the save button
	IsShowBtnUpdate	Whether display the query button
	IsShowCbTime	Whether display the quick time setting button
	IsShowSetTime	Whether display the time setting button
	IsShowStatusBar	Whether display the status column
	IsShowToolBar	Whether display the tools column
	Left	The left coordinate
	Name	Name
	NumberAxisMaximum	Get or set the maximum value of numerical axis
	NumberAxisMinimum	Get or set the minimum value of numerical axis
	NumberInterval	Get or set the maximum scale of numerical axis
	NumberSmallInterval	Get or set the minimum scale of numerical axis
	ToolTip	Tooltip text
	Top	The top coordinate
	Width	Width
	ZIndex	Layer index

The following is the detailed description of the script:

【Method】

(1) AddNewLimitLine method

AddNewLimitLine

Add new limit line (The "UpperAndLowerDisplay" is checked in the control property)

Define

AddNewLimitLine(variablePath ,defaultValue ,color , thickness)

Parameter

Name	Required/Optional	Data Type	Description
variablePath	Required	**String**	Variable path
defaultValue	Required	**Double**	Default value (default)
color	Required	**Brush**	Line fill color
thickness	Required	**Double**	Line thicknesses

Example

Add a new limit line

VBScript Example

```
Call HistoryChart0.AddNewLimitLine("Var.variable",0,Colors.Red,5)
```

(2) ArrayToString method

ArrayToString

Image array translate into string

Define

ArrayToString(seriesName)

Parameter

Name	Required/Optional	Data Type	Description
seriesName	Required	**String**	Name of the curve, separate by commas

Example

Export curve image into the report

The operation steps are as follows:

a.Create a HistoryChart0 in the screen and associate variables that need to be viewed.

b.Create a new report template "Report0", select a cell to associate history variables and time.

c.In the report template, the menu bar --> Realtime Variable --> System --> Image, to add the corresponding number of curve pictures.

d.Add a button to the created screen and write in the script:

VBScript Example

```
dt = HistoryChart0.GetDataTable("Series0") ' Get "series0" data of the HistoryChart0
dt1 = HistoryChart1.GetDataTable("Series0")
dt2 = HistoryChart2.GetDataTable("Series0")
Call DbAccessCmd.SetPrimary(dt,"TriggerTime") 'Set "TriggerTime" as the primary key
Call DbAccessCmd.SetPrimary(dt1,"TriggerTime")
Call DbAccessCmd.SetPrimary(dt2,"TriggerTime")
Call dt.Merge(dt1) 'Merge dt1 data into dt
Call dt.Merge(dt2)
Call
ReportCmd.ExportDataByTemplate("D:\Test12.xlsx","Report0",dt,HistoryChart0.ArrayToString("Series0")&"|"&Histo
```

(3) Export method

Export

Export curve with report template

Define

Export(saveFilePath ,seriesName ,templateName)

Parameter

Name	Required/Optional	Data Type	Description
saveFilePath	Required	**String**	Save the full path, the default is empty string
seriesName	Required	**String**	Name of the curve
templateName	Required	**String**	Report template

Example

1.Export Curve picture through dialog

VBScript Example

```
Call HistoryChart0.Export("","Series0;Series1","Report0")
```

2.Export Curve picture to specified path

VBScript Example

```
Call HistoryChart0.Export("D:\Test1.xlsx","Series0;Series1","Report0")
```

(4) Export method

Export

Export excel

Define

Export(seriesName)

Parameter

Name	Required/Optional	Data Type	Description
seriesName	Required	**String**	Name of the curve

Example

Export curve image into excel

VBScript Example

```
Call HistoryChart0.Export("Series0")
```

(5) FindAnimation method

FindAnimation

Look for animation to modify the associated variables of the animation

Define

FindAnimation(name)

Parameter

Name	Required/Optional	Data type	Description
name	Required	**String**	The name of animation

Example

Change the associated variables of the visibility animation of the object into Var.VariableGroup0.Variable1

VBScript Example

```
HistoryChart0.FindAnimation("HMIShowHideAnimation").Expression =
"Var.VariableGroup0.Variable1"
```

(6) FirstAction method

FirstAction

The most left

Define

FirstAction()

Example

The most left

VBScript Example

```
Call HistoryChart0.FirstAction()
```

(7) GetDataTable method

GetDataTable

Store data to data table

Define

GetDataTable(seriesName)

Parameter

Name	Required/Optional	Data Type	Description
seriesName	Required	**String**	Name of curve, separate by commas

Example

Store data to data table

VBScript Example

```
dt = HistoryChart0.GetDataTable("Series0")    'Get data of "Series0"
dt1 = HistoryChart0.GetDataTable("Series1")
Call DbAccessCmd.SetPrimary(dt,"TriggerTime")    'Set "TriggerTime" as primary key
Call DbAccessCmd.SetPrimary(dt1,"TriggerTime")
```

```
Call dt.Merge(dt1) 'merge dt1 data into dt
Call
ReportCmd.ExportDataByTemplate("D:\Test12.xlsx","Report2",dt,HistoryChart0.ArrayToString("Series0")&"|"&Histo
```

(8) HiddenSeries method

HiddenSeries

Hide curve

Define

HiddenSeries(seriesName)

Parameter

Name	Required/Optional	Data Type	Description
seriesName	Required	**String**	Name of curve, separate by commas

Example

Hide curve

VBScript Example

```
Call HistoryChart0.HiddenSeries("Series0")
```

(9) Import method

Import

Import excel

Define

Import(seriesName)

Parameter

Name	Required/Optional	Data Type	Description
seriesName	Required	**String**	The name of import curve

Example

Import excel

VBScript Example

```
Call HistoryChart0.Import("Series0")
```

(10) LastAction method

LastAction

The most right

Define

LastAction()

Example

The most right

VBScript Example

```
Call HistoryChart0.LastAction()
```

(11) NextAction method

NextAction

Right shift

Define

NextAction()

Example

Right shift

VBScript Example

```
Call HistoryChart0.NextAction()
```

(12) OpenShowFileDialog method

OpenShowFileDialog

Get saved path which selected by dialog box

Define

OpenShowFileDialog()

Example

Get saved path which selected by dialog box

VBScript Example

```
tt = HistoryChart0.OpenShowFileDialog()
Call
ReportCmd.ExportDataByTemplate(tt,"Report0",HistoryChart0.ArrayToString("Series0")&"|"&HistoryChart0.ArrayTo
```

(13) PreAction method

PreAction

Left shift

Define

PreAction()

Example

Left shift

VBScript Example

```
Call HistoryChart0.PreAction()
```

(14) Print method

Print

Print

Define

Print()

Example

Print

VBScript Example

```
Call HistoryChart0.Print()
```

(15) QueryEndTime

QueryEndTime

Set the end time of the query data

Define

QueryEndTime(dt)

Parameter

Name	Required/Optional	Data Type	Description
dt	Required	**DateTime**	End time

Example

Set the end time of the query data

VBScript Example
Call HistoryChart0.QueryStartTime(DateTimePicker0.ValueTime) Call HistoryChart0.QueryEndTime(DateTimePicker1.ValueTime) Call HistoryChart0.QueryInveral(1) Call HistoryChart0.QueryUnit(0) Call HistoryChart0.QueryHistoryDate()

(16) QueryHistoryDate method

QueryHistoryDate

Query of the historical data

Define

QueryHistoryDate()

Example

1.Query the historical data(Timing record)

VBScript Example
Call HistoryChart0.QueryStartTime(DateTimePicker0.ValueTime) Call HistoryChart0.QueryEndTime(DateTimePicker1.ValueTime) Call HistoryChart0.SetVarRecordRulerName("s1") Call HistoryChart0.QueryHistoryDate()

2.Query the historical data(Change record)

VBScript Example
Call HistoryChart0.QueryStartTime(DateTimePicker0.ValueTime) Call HistoryChart0.QueryEndTime(DateTimePicker1.ValueTime) Call HistoryChart0.QueryHistoryDate()

3.Query the historical group data

VBScript Example
Call HistoryChart0.QueryStartTime(DateTimePicker0.ValueTime) Call HistoryChart0.QueryEndTime(DateTimePicker1.ValueTime) Call HistoryChart0.QueryInveral(1) Call HistoryChart0.QueryUnit(0) Call HistoryChart0.QueryHistoryDate()

(17) QueryHistoryDate method

QueryHistoryDate

Query of the historical data

Define

QueryHistoryDate(startTime , endTime , inveralTime , index)

Parameter

Name	Required/Optional	Data Type	Description
startTime	Required	**DateTime**	Start time
endTime	Required	**DateTime**	End time
inveralTime	Required	**Int**	Time interval
index	Required	**Int**	Time interval unit（Value of 0-5，Corresponding to s、mi、h、d、m、y）

Example

1.Query the historical data(Timing record)

VBScript Example

```
Call HistoryChart0.SetVarRecordRulerName("s1")    'Set the query rule to "s1"
Call HistoryChart0.QueryHistoryDate(DateTimePicker0.ValueTime,DateTimePicker1.ValueTime,1,0)
```

2.Query the historical data(Change record) or Query the historical group data

VBScript Example

```
Call HistoryChart0.QueryHistoryDate(DateTimePicker0.ValueTime,DateTimePicker1.ValueTime,1,0)
```

(18) QueryInveral method

QueryInveral

Set the query interval

Define

QueryInveral(inveral)

Parameter

Name	Required/Optional	Data Type	Description
inveral	Required	**Int**	Interval

Example

Query the historical group data

VBScript Example

```
Call HistoryChart0.QueryStartTime(DateTimePicker0.ValueTime)
Call HistoryChart0.QueryEndTime(DateTimePicker1.ValueTime)
Call HistoryChart0.QueryInveral(1)
Call HistoryChart0.QueryUnit(0)
Call HistoryChart0.QueryHistoryDate()
```

(19) QueryStartTime method

QueryStartTime

Set the start time of the query data

Define

QueryStartTime(dt)

Parameter

Name	Required/Optional	Data Type	Description
dt	Required	**DateTime**	Start time

Example

Query the historical group data

VBScript Example
Call HistoryChart0.QueryStartTime(DateTimePicker0.ValueTime)
Call HistoryChart0.QueryEndTime(DateTimePicker1.ValueTime)
Call HistoryChart0.QueryInveral(1)
Call HistoryChart0.QueryUnit(0)
Call HistoryChart0.QueryHistoryDate()

(20) QueryUnit method

QueryUnit

Set the query unit

Define

QueryUnit(index)

Parameter

Name	Required/Optional	Data Type	Description
index	Required	**Int**	Time interval unit（Value of 0-5，Corresponding to s、mi、h、d、m、y）

Example

Query the historical group data

VBScript Example
Call HistoryChart0.QueryStartTime(DateTimePicker0.ValueTime)
Call HistoryChart0.QueryEndTime(DateTimePicker1.ValueTime)
Call HistoryChart0.QueryInveral(1)
Call HistoryChart0.QueryUnit(0)

```
Call HistoryChart0.QueryHistoryDate()
```

(21) RemoveAllLimitLine method

RemoveAllLimitLine

Remove all limit line

Define

RemoveAllLimitLine()

Example

Remove all limit line

VBScript Example

```
Call HistoryChart0.RemoveAllLimitLine()
```

(22) Save method

Save

Save

Define

Save()

Example

Save

VBScript Example

```
Call HistoryChart0.Save()
```

(23) Save method

Save

Save as pictures

Define

Save(path ， seriesName)

Parameter

Name	Required/Optional	Data Type	Description
path	Required	**String**	Absolute path
seriesName	Required	**String**	Name of curve, separate by commas

Example

Save

VBScript Example
CallHistoryChart0.Save("D:\test.jpg","Series0")

(24) SetSeriesVariablePath method

SetSeriesVariablePath

Curve correlation variable substitution

Define

SetSeriesVariablePath(pathList)

Parameter

Name	Required/Optional	Data Type	Description
pathList	Required	**String**	The new variable path of the variables that replace the old variables , separate by commas

Example

Curve correlation variable substitution

VBScript Example
Call HistoryChart0.SetSeriesVariablePath("Var.variable0,Var.variable1")

(25) SetVarRecordRulerName method

SetVarRecordRulerName

Set the name of query rule

Define

SetVarRecordRulerName(RuleName)

Parameter

Name	Required/Optional	Data Type	Description
RuleName	Required	**String**	The name of timer

Example

Set the name of query rule，s1 is the timer name

VBScript Example
Call HistoryChart0.SetVarRecordRulerName("s1")

(26) VisiableSeries method

VisiableSeries

Display the curve

Define

VisiableSeries(seriesName)

Parameter

Name	Required/Optional	Data Type	Description
seriesName	Required	**String**	Name of curve, separate by commas

Example

Display the curve

VBScript Example
Call HistoryChart0.VisiableSeries("Series0")

【property】

(1) Height property

Height

Height

Define

Double Height

Example

Set the height of the specified object to 100

VBScript Example
HistoryChart0.Height = 100

(2) IsShow property

IsShow

Whether display the specified object or not

Define

Boolean IsShow

Example

True ：The specified object is displayed on the screen False ：The specified object hidden on the screen

VBScript Example
HistoryChart0.IsShow = True HistoryChart0.IsShow = False

(3) IsShowBtnConfig

IsShowBtnConfig

Whether display the curve configuration button

Define

Boolean IsShowBtnConfig

Example

Whether display curve configuration button

VBScript Example
HistoryChart0.IsShowBtnConfig = True

(4) IsShowBtnCrossLine property

IsShowBtnCrossLine

Whether display the positioning line button

Define

Boolean IsShowBtnCrossLine

Example

Display the positioning line

VBScript Example

```
HistoryChart0.IsShowBtnCrossLine = True
```

(5) IsShowBtnMenu

IsShowBtnMenu

Whether display the import or export button

Define

Boolean IsShowBtnMenu

Example

Display of import or export button

VBScript Example

```
HistoryChart0.IsShowBtnXYMenu = True
```

(6) IsShowBtnPrint

IsShowBtnPrint

Whether display the print button

Define

Boolean IsShowBtnPrint

Example

Display of print button

VBScript Example

```
HistoryChart0.IsShowBtnPrint = True
```

(7) IsShowBtnSave

IsShowBtnSave

Whether display the save button

Define

Boolean IsShowBtnSave

Example

Display of save button

VBScript Example

```
HistoryChart0.IsShowBtnSave = True
```

(8) IsShowBtnUpdate

IsShowBtnUpdate

Whether display the query button

Define

Boolean IsShowBtnUpdate

Example

Display of query button

VBScript Example

```
HistoryChart0.IsShowBtnUpdate = True
```

(9) IsShowCbTime

IsShowCbTime

Whether display the quick time setting button

Define

Boolean IsShowCbTime

Example

Display of quick time setting button

VBScript Example

```
HistoryChart0.IsShowCbTime = True
```

(10) IsShowSetTime

IsShowSetTime

Whether display the time setting button

Define

Boolean IsShowSetTime

Example

Display of time setting button

VBScript Example

```
HistoryChart0.IsShowSetTime = True
```

(11) IsShowStatusBar

IsShowStatusBar

Whether display the the status column

Define

Boolean IsShowStatusBar

Example

Whether to display the status column

VBScript Example

```
HistoryChart0.IsShowStatusBar = True
```

(12) IsShowToolBar

IsShowToolBar

Whether display the tools column

Define

Boolean IsShowToolBar

Example

Whether display the tools column

VBScript Example
HistoryChart0.IsShowToolBar = True

(13) Left property

Left

The left coordinate

Define

Double Left

Example

Object's left side of the coordinate values is 100 on the screen

VBScript Example
HistoryChart0.Left = 100

(14) Name property

Name

Name

Define

String Name

Example

Get the default name of the specified object

VBScript Example
TextBox0.Text = HistoryChart0.Name

(15) NumericalAxisMaximum property

NumberAxisMaximum

Get or set the maximum scale of numerical axis

Define

Double NumberAxisMaximum

(16) NumericalAxisMinimum property

NumberAxisMinimum

Get or set the minimum scale of numerical axis

Define

Double NumberAxisMinimum

(17) NumberInterval

NumberInterval

Get or set the large scale of numerical axis

Define

Double NumberInterval

(18) NumberSmallInterval

NumberSmallInterval

Get or set the small scale of numerical axis

Define

Double NumberSmallInterval

(19) ToolTip property

ToolTip

Tooltip text

Define

String ToolTip

Example

Set the ToolTip of the specified object to "test"

VBScript Example

```
HistoryChart0.ToolTip = "test"
```

(20) Top property

Top

The top coordinate

Define

Double Top

Example

Object's the top of the coordinate value is 100 in the picture

VBScript Example

```
HistoryChart0.Top = 100
```

(21) Width property

Width

Width

Define

Double Width

Example

Set the width of the specified object to 100

VBScript Example

```
HistoryChart0.Width = 100
```

(22) ZIndex property

ZIndex

Layer index

Define

Int32 ZIndex

Example

Set the Layer index of the specified object to 3

VBScript Example

```
HistoryChart0.ZIndex = 3
```

3. HMIXYChart object

HMIXYChart

XY chart control

Methods list

	Name	Description
	AddNewLimitLine	Add a new limit line
	AddPoint	New data points
	ArrayToString	Image array translate into string
	DeletePoint	Delete the data points
	DeletevalueAll	Delete all the data points
	Export	Export the visible curve
	ExportHistoryData	Export history data
	ExportRealData	Export real-time data
	ExportRealData	Export real-time data
	FindAnimation	Look for animation to modify the associated variables of the animation
	FirstAction	The most left
	GetDataTable	Store data to data table
	HiddenSeries	Hide curve
	HistoryXySeriesExport	Export history XY curve
	Import	Import excel
	LastAction	The most right
	MarginBottom	Bottom margin under the curve
	MarginLeft	Left margin of the curve
	MarginRight	Right margin of the curve
	MarginTop	Top margin of the curve
	NextAction	Right shift
	OpenShowFileDialog	Get the path of save which dialog select
	PreAction	Left shift
	Print	Print

	QueryDate	Data query
	QueryEndTime	Set the end time of the query data
	QueryHistoryDate	Query of the historical data
	QueryInveral	Set the query interval
	QueryStartTime	Set the start time of the query data
	QueryUnit	Set the query unit
	RemoveAllLimitLine	Remove all limit line
	Save	Save
	Save	Save as pictures
	SeriesIsUpdate	whether to update the curve
	SetLimitLinePointColor	Set the color of points and curves outside the limit line
	SetLineAndPointColor	Set the color of points and curves
	SetSeriesVariablePath	Curve correlation variable substitution
	SetVarRecordRulerName	Set the name of query rule
	StartTimer	Start timer
	StopTimer	Stop timer
	UpdatePoint	Update data point
	VisiableSeries	Display the curve
	XAxisTransformRatio	The X axis ratio
	XyValueTransform	XY curvilinear coordinates translate into window coordinates
	YAxisTransformRatio	The Y axis ratio

Property list

	Name	Description
	DataSize	Refresh count
	Height	Height
	IsShow	Whether display the specified object
	IsShowBtnCrossLine	Whether display the positioning line button
	IsShowBtnXYClear	Whether display the clear button
	IsShowBtnXYDown	Whether display the down button
	IsShowBtnXYLeft	Whether display the left shift button
	IsShowBtnXYMenu	Whether display the import and export button
	IsShowBtnXYPrint	Whether display the print button
	IsShowBtnXYRelConfig	Whether display the real-time curve configuration button

IsShowBtnXYRight	Whether display the right shift button
IsShowBtnXYSave	Whether display the save button
IsShowBtnXYStop	Whether display the start or stop button
IsShowBtnXYUp	Whether display the up button
IsShowHtBtnXYConfig	Whether display the history curve configuration button
IsShowHtBtnXYQuery	Whether display the history curve query button
IsShowStatusBar	Whether display the the status column
IsShowXYCbTime	Whether display the quick time setting button
IsShowXYSetTime	Whether display the time setting button
Left	The left coordinate
Name	Name
ToolTip	Tooltip text
Top	The top coordinate
Width	Width
XAxisInterval	Get or set the large scale of X axis
XAxisMaximum	Get or set the maximum of X axis
XAxisMinimum	Get or set the minimum of X axis
XAxisSmallInterval	Get or set the small scale of X axis
YAxisInterval	Get or set the large scale of Y axis
YAxisMaximum	Get or set the maximum of Y axis
YAxisMinimum	Get or set the minimum of Y axis
YAxisSmallInterval	Get or set the small scale of Y axis
ZIndex	Layer index

The following is the detailed description of the script:

【Method】

(1) AddNewLimitLine method

AddNewLimitLine

Add new limit line (The "UpperAndLowerDisplay" is checked in the control property)

Define

AddNewLimitLine(variablePath ,defaultValue ,color , thickness)

Parameter

Name	Required/Optional	Data Type	Description
variablePath	Required	**String**	Variable path
defaultValue	Required	**Double**	Default value (default)
color	Required	**Brush**	Line fill color
thickness	Required	**Double**	Line thicknesses

Example

Add new limit line

VBScript Example
Call XYchart0.AddNewLimitLine("Var.变量",90,Colors.Red,5)

(2) AddPoint method

AddPoint

Add data point

Define

AddPoint(seriesName ,x ,y)

Parameter

Name	Required/Optional	Data Type	Description
seriesName	Required	**String**	curve name
x	Required	**Double**	X value
y	Required	**Double**	Y value

Example

Add data point

VBScript Example
Call XYchart0.AddPoint("Series0",50,50)

(3) ArrayToString method

ArrayToString

Image array translate into string

Define

ArrayToString(seriesName)

Parameter

Name	Required/Optional	Data Type	Description
seriesName	Required	**String**	Name of the curve, separate by commas

Example1

Export real-time curve image of XYChart into the report:

The operation steps are as follows:

a.Create a XYChart0 in the screen and associate variables that need to be viewed.

b.Create a new report template "Report0", select a cell to associate real-time variables and refresh time.

c.In the report template, the menu bar --> Realtime Variable --> System --> Image, to add the corresponding number of curve pictures.

d.Add a button to the created screen and write in the script (the following two different examples):

1. The two diferent curve pictures from two different realtimechart are exported to the same report

（"D:\Test.xlsx"is the save path of Excel，"report0"is the report template，"RealtimeChart0.ArrayToString("Series0")" is the pictures set）

VBScript Example

```
Call ReportCmd.ExportDataByTemplate("D:\Test.xlsx","Report0",XYChart0.ArrayToString("Series0")&"|"&XYChart1.Arra
```

2. The two diferent curve pictures from one realtimechart are exported to the same report

VBScript Example

```
Call ReportCmd.ExportDataByTemplate("D:\Test.xlsx","Report0",XYChart0.ArrayToString("Series0")&"|"&XYChart0.Arra
```

Example2

The two diferent curve pictures from two different XYchart are exported to the same report

The operation steps are as follows:

a.Create a XYChart0 in the screen and associate variables that need to be viewed.

b.Create a new report template "Report0", select a cell to associate history variables and time.

c.In the report template, the menu bar --> Realtime Variable --> System --> Image, to add the corresponding number of curve pictures.

d.Add a button to the created screen and write in the script:

VBScript Example
dt = XYChart0.GetDataTable("SeriesConfig0,SeriesConfig1") Call ReportCmd.ExportDataByTemplate("D:\test.xlsx","Report0",dt,XYChart0.ArrayToString("SeriesConfig0")&"|"&X

(4) DeletePoint method

DeletePoint

Delete data point

Define

DeletePoint(seriesName , itemIndex)

Parameter

Name	Required/Optional	Data Type	Description
seriesName	Required	**String**	Name of curve
itemIndex	Required	**int**	select the curve points' index

Example

Delete data point

VBScript Example
Call XYChart0.DeletePoint("Series0",0)

(5) DeletevalueAll method

DeletevalueAll

Delete all data points

Define

DeletevalueAll()

Example

Delete all data points

VBScript Example
Call XY 曲线 0.DeletevalueAll()

(6) Export method

Export

Export the visible curve

Define

Export(exportSeriesName)

Parameter

Name	Required/Optional	Data Type	Description
seriesName	Required	**String**	The name of the curve

Example

Export the visible curve

VBScript Example
Call XYChart0.Export("Series0,SeriesConfig0,SeriesConfig1")

(7) ExportHistoryData method

ExportHistoryData

Export history data

Define

ExportHistoryData(historyseriesName)

Parameter

Name	Required/Optional	Data Type	Description
historyseriesName	Required	**String**	The name of curve

Example

Export history data

VBScript Example
Call XYChart0.ExportHistoryData("Series0")

(8) ExportRealData method

ExportRealData

Export real-time data

Define

ExportRealData(seriesName)

Parameter

Name	Required/Optional	Data Type	Description
seriesName	Required	**String**	The name of curve

Example

Export real-time data

VBScript Example
Call XYChart0.ExportRealData("Series0")

(9) ExportData method

ExportData

Export data

Define

ExportRealData(seriesName ， templateName)

Parameter

Name	Required/Optional	Data Type	Description
seriesName	Required	**String**	The name of curve
templateName	Required	**String**	The name of report template

Example

Export data

VBScript Example
Call XYChart0.ExportRealData("Series0","Report0")

(10) FindAnimation method

FindAnimation

Look for animation to modify the associated variables of the animation

Define

FindAnimation(name)

Parameter

Name	Required/Optional	Data type	Description
name	Required	**String**	The name of animation

Example

Change the associated variables of the visibility animation of the object into Var.VariableGroup0.Variable1

VBScript Example
XYChart0.FindAnimation("HMIShowHideAnimation").Expression = "Var.VariableGroup0.Variable1"

(11) FirstAction method

FirstAction

The most left

Define

FirstAction()

Example

The most left

VBScript Example
Call XYChart0.FirstAction()

(12) GetDataTable method

GetDataTable

Store data to data table

Define

GetDataTable(seriesName)

Parameter

Name	Required/Optional	Data Type	Description
seriesName	Required	**String**	Name of the curve, separate by commas

Example

Store data to data table

VBScript Example
dim dt dt = XYChart0.GetDataTable("SeriesConfig0") Call ReportCmd.DirectExportDataToExcel(dt,"D:\XYDT.xlsx",1)

(13) HiddenSeries method

HiddenSeries

The hidden curve

Define

HiddenSeries(seriesName)

Parameter

Name	Required/Optional	Data Type	Description

seriesName	Required	**String**	Name of the curve, separate by commas

Example

The hidden curve

VBScript Example
Call XYChart0.HiddenSeries("Series0")

(14) HistoryXySeriesExport method

HistoryXySeriesExport

Export data

Define

HistoryXySeriesExport(seriesName ， templateName)

Parameter

Name	Required/Optional	Data Type	Description
seriesName	Required	**String**	History curve of XY name
templateName	Required	**String**	The name of report template

Example

Export data

VBScript Example
Call XYChart0.HistoryXySeriesExport("Series0","Report0")

(15) Import method

Import

Import excel

Define

Import(seriesName)

Parameter

Name	Required/Optional	Data Type	Description
seriesName	Required	**String**	The name of the import curve

Example

Import excel

VBScript Example
Call XYChart0.Import("Series0")

(16) LastAction method

LastAction

The most right

Define

LastAction()

Example

The most right

VBScript Example
Call XYChart0.LastAction()

(17) MarginBottom method

MarginBottom

Bottom margin under the curve

Define

MarginBottom()

Example

Bottom margin under the curve

VBScript Example
Button0.Content = XYChart0.MarginBottom()

(18) MarginLeft method

MarginLeft

Left margin of the curve

Define

MarginLeft()

Example

Left margin of the curve

VBScript Example
Button0.Content = XYChart0.Marginleft()

(19) MarginRight method

MarginRight

Right margin of the curve

Define

MarginRight()

Example

Right margin of the curve

VBScript Example
Button0.Content = XYChart0.MarginRight()

(20) MarginTop method

MarginTop

Top margin of the curve

Define

MarginTop()

Example

Top margin of the curve

VBScript Example

```
Button0.Content = XYChart0.MarginTop()
```

(21) NextAction method

NextAction

Right shift

Define

NextAction()

Example

Right shift

VBScript Example

```
Call XYChart0.NextAction()
```

(22) OpenShowFileDialog method

OpenShowFileDialog

Get the saved path which selected by dialog box

Define

OpenShowFileDialog()

Example

Get saved path which selected by dialog box

VBScript Example

```
tt = XYChart0.OpenShowFileDialog()
Call
ReportCmd.ExportDataByTemplate(tt,"Report0",XYChart0.ArrayToString("Series0")&"|"&XYChart0.ArrayToString("S
```

(23) PreAction method

PreAction

Left shift

Define

PreAction()

Example

Left shift

VBScript Example
Call XYChart0.PreAction()

(24) Print method

Print

Print

Define

Print()

Example

Print

VBScript Example
Call XYChart0.Print()

(25) QueryDate method

QueryDate

Query history data

Define

QueryDate()

Example

1.Query the historical data(Timing record)

VBScript Example
Call XYChart0.QueryStartTime(DateTimePicker0.ValueTime) Call XYChart0.QueryEndTime(DateTimePicker1.ValueTime)

```
Call XYChart0.SetVarRecordRulerName("s1")
Call XYChart0.QueryHistoryDate()
```

2.Query the historical data(Change record)

VBScript Example

```
Call XYChart0.QueryStartTime(DateTimePicker0.ValueTime)
Call XYChart0.QueryEndTime(DateTimePicker1.ValueTime)
Call XYChart0.QueryHistoryDate()
```

3.Query the historical group data

VBScript Example

```
Call XYChart0.QueryStartTime(DateTimePicker0.ValueTime)
Call XYChart0.QueryEndTime(DateTimePicker1.ValueTime)
Call XYChart0.QueryInveral(1)
Call XYChart0.QueryUnit(0)
Call XYChart0.QueryHistoryDate()
```

(26) QueryEndTime method

QueryEndTime

Set the end time of the query data

Define

QueryEndTime(dt)

Parameter

Name	Required/Optional	Data Type	Description
dt	Required	**DateTime**	End time

Example

Set the end time of the query data

VBScript Example

```
Call XYChart0.QueryStartTime(DateTimePicker0.ValueTime)
Call XYChart0.QueryEndTime(DateTimePicker1.ValueTime)
Call XYChart0.QueryInveral(1)
```

```
Call XYChart0.QueryUnit(0)
Call XYChart0.QueryHistoryDate()
```

(27) QueryEndTime method

QueryEndTime

Set the end time of the query data

Define

QueryEndTime(dt)

Parameter

Name	Required/Optional	Data Type	Description
dt	Required	**DateTime**	End time

Example

Set the end time of the query data

VBScript Example

```
Call XYChart0.QueryStartTime(DateTimePicker0.ValueTime)
Call XYChart0.QueryEndTime(DateTimePicker1.ValueTime)
Call XYChart0.QueryInveral(1)
Call XYChart0.QueryUnit(0)
Call XYChart0.QueryHistoryDate()
```

(28) QueryInveral method

QueryInveral

Set the query interval

Define

QueryInveral(inveral)

Parameter

Name	Required/Optional	Data Type	Description
inveral	Required	**Int**	Time interval

Example

Query the historical group data

VBScript Example
Call XYChart0.QueryStartTime(DateTimePicker0.ValueTime)
Call XYChart0.QueryEndTime(DateTimePicker1.ValueTime)
Call XYChart0.QueryInveral(1)
Call XYChart0.QueryUnit(0)
Call XYChart0.QueryHistoryDate()

(29) QueryStartTime method

QueryStartTime

Set the start time of the data query

Define

QueryStartTime(dt)

Parameter

Name	Required/Optional	Data Type	Description
dt	Required	**DateTime**	start time

Example

Query the historical group data

VBScript Example
Call XYChart0.QueryStartTime(DateTimePicker0.ValueTime)
Call XYChart0.QueryEndTime(DateTimePicker1.ValueTime)
Call XYChart0.QueryInveral(1)
Call XYChart0.QueryUnit(0)

(30) QueryUnit method

QueryUnit

Set the query unit

Define

QueryUnit(index)

Parameter

Name	Required/Optional	Data Type	Description
index	Required	**Int**	Time interval unit（Value of 0-5，Corresponding to s、mi、h、d、m、y）

Example

Query the historical group data

VBScript Example
Call XYChart0.QueryStartTime(DateTimePicker0.ValueTime)
Call XYChart0.QueryEndTime(DateTimePicker1.ValueTime)
Call XYChart0.QueryInveral(1)
Call XYChart0.QueryUnit(0)
Call XYChart0.QueryHistoryDate()

(31) RemoveAllLimitLine method

RemoveAllLimitLine

Remove all limit line

Define

RemoveAllLimitLine()

Example

Remove all limit line

VBScript Example
Call XYChart0.RemoveAllLimitLine()

(32) Save method

Save

Save

Define

Save()

Example

Save

VBScript Example
Call XYChart0.Save()

(33) Save method

Save

Save as pictures

Define

Save(path ， seriesName)

Parameter

Name	Required/Optional	Data Type	Description
path	Required	**String**	Absolute path
seriesName	Required	**String**	The name of curve, separate by commas

Example

Save

VBScript Example
Call XYChart0.Save("D:\test.jpg","Series0")

(34) SeriesIsUpdate method

SeriesIsUpdate

Whether to update the curve

Define

SeriesIsUpdate(seriesName ，isUpdate)

Parameter

Name	Required/Optional	Data Type	Description
seriesName	Required	**String**	The name of curve, separate by commas
isUpdate	Required	**Bool**	whether to update

Example

Whether to update the curve

VBScript Example
Call XYChart0.SeriesIsUpdate("Series0",True)

(35) SetLimitLinePointColor method

SetLimitLinePointColor

Set the color of points and curves outside the limit line

Define

SetLimitLinePointColor(maxVariablePath , minVariablePath ,brush)

Parameter

Name	Required/Optional	Data Type	Description
maxVariablePath	Required	**String**	Variable path of the maximum value of the limit line
minVariablePath	Required	**String**	Variable path of the minimum value of the limit line
brush	Required	**Brush**	New color of the points and curves

Example

Set the color of points and curves outside the limit line

Note： LINE+Metro+this script: Ten sections of different colors of points and curves update Circularly

LINE+Custom+this script: The points and curves colors outside the limit line are the color set by the script

LINE+Normal+this script: The points color outside the limit line are the color set by the script, and the curves color don't change.

VBScript Example
Call XYChart0.SetLimitLinePointColor("Var.XY.up","Var.XY.Lower",Colors.Red)

(36) SetLineAndPointColor method

SetLineAndPointColor

Set the color of points and curves

Define

SetLineAndPointColor(brush)

Parameter

Name	Required/Optional	Data Type	Description
brush	Required	**Brush**	New color of the points and curves

Example

Set the color of points and curves

Note： LINE+Metro+this script: Ten sections of different colors of points and curves update Circularly
LINE+Custom+this script: The points and curves colors are the color set by the script
LINE+Normal+this script: The points color are the color set by the script, and the curves color don't change.

VBScript Example
Call XYChart0.SetLineAndPointColor(Colors.Red)

(37) SetSeriesVariablePath method

SetSeriesVariablePath

Curve correlation variable substitution

Define

SetSeriesVariablePath(relpathList,htpathList)

Parameter

Name	Required/Optional	Data Type	Description

relpathList	Required	**String**	The new variable path of the real-time variables that replace the old variables，group X and Y variables are separated by a semicolon，variable groups are separated by commas
htpathList	Required	**String**	The new variable path of the history variables that replace the old variables，group X and Y variables are separated by a semicolon，variables groups are separated by commas

Example

Curve correlation variable substitution

VBScript Example

```
Call XYChart0.SetSeriesVariablePath("Var.variable0;Var.variable1","HistoryRecord;HistoryRecord1")
```

(38) SetVarRecordRulerName method

SetVarRecordRulerName

Set the name of query rule

Define

SetVarRecordRulerName(RuleName)

Parameter

Name	Required/Optional	Data Type	Description
RuleName	Required	**String**	The name of timer

Example

1.Set the name of query rule，s1 is the timer name

VBScript Example

```
Call XYChart0.SetVarRecordRulerName("s1")
```

2.Query the historical data(Timing record)

VBScript Example

```
Call XYChart0.QueryStartTime(DateTimePicker0.ValueTime)
Call XYChart0.QueryEndTime(DateTimePicker1.ValueTime)
Call XYChart0.SetVarRecordRulerName("s1")
Call XYChart0.QueryHistoryDate()
```

(39) StartTimer method

StartTimer

Start timer

Define

StartTimer()

Example

Start timer

VBScript Example

```
Call XYChart0.StartTimer()
```

(40) StopTimer

StopTimer

Stop timer

Define

StopTimer()

Example

Stop timer

VBScript Example

```
Call XYChart0.StopTimer()
```

(41) UpdatePoint method

UpdatePoint

Update data point

Define

UpdatePoint(seriesName ，itemIndex , x , y)

Parameter

Name	Required/Optional	Data Type	Description
seriesName	Required	**String**	Name of curve, separate by commas
itemIndex	Required	**Int**	Select the curve index
x	Required	**Double**	Data point
y	Required	**Double**	Data point

Example

Update data point

VBScript Example
Call XYChart0.UpdatePoint("Series0",0,50,50)

(42) VisiableSeries method

VisiableSeries

Display curves

Define

VisiableSeries(seriesName)

Parameter

Name	Required/Optional	Data Type	Description
seriesName	Required	**String**	Name of the curve, separate by commas

Example

Display curves

VBScript Example
Call XYChart0.VisiableSeries("Series0")

(43) XAxisTransformRatio method

XAxisTransformRatio

The X axis ratio

Define

XAxisTransformRatio()

Example

The X axis ratio

VBScript Example
Button0.Content = XYChart0.XAxisTransformRatio()

(44) XyValueTransform method

XyValueTransform

XY curvilinear coordinates into window coordinates

Define

XyValueTransform(xy)

Parameter

Name	Required/Optional	Data Type	Description
seriesName	Required	**String**	Coordinates type string，such as "20,20"

Example

XY curvilinear coordinates into window coordinates

VBScript Example
Button0.Content = XYChart0.XyValueTransform("20,20")

(45) YAxisTransformRatio method

YAxisTransformRatio

The Y axis ratio

Define

YAxisTransformRatio()

Example

The Y axis ratio

VBScript Example

```
Button0.Content = XYChart0.YAxisTransformRatio()
```

【property】

(1) DataSize property

DataSize

Refresh count

Define

int DataSize

Example

Set the refreshing count to 10

VBScript Example

```
XYchart0.DataSize = 10
```

(2) Height property

Height

Height

Define

Double Height

Example

Set the height of the specified object to 100

VBScript Example

```
XYchart0.Height = 100
```

(3) IsShow property

IsShow

Whether display the specified object

Define

Boolean IsShow

Example

True ：The specified object is displayed on the screen False ：The specified object hidden on the screen

VBScript Example
XYchart0.IsShow = True XYchart0.IsShow = False

(4) IsShowBtnCrossLine property

IsShowBtnCrossLine

Whether display the positioning line button

Define

Boolean IsShowBtnCrossLine

Example

Display the positioning line

VBScript Example
XYchart0.IsShowBtnCrossLine = True

(5) IsShowBtnXYClear property

IsShowBtnXYClear

Whether display the Clear Button

Define

Boolean IsShowBtnXYClear

Example

Display the Clear Button

VBScript Example
XYchart0.IsShowBtnXYClear = True

(6) IsShowBtnXYDown property

IsShowBtnXYDown

Whether display the down button

Define

Boolean IsShowBtnXYDown

Example

Display the down button

VBScript Example

```
XYchart0.IsShowBtnXYDown = True
```

(7) IsShowBtnXYLeft property

IsShowBtnXYLeft

Whether display the left shift button

Define

Boolean IsShowBtnXYLeft

Example

Display the left shift button

VBScript Example

```
XYchart0.IsShowBtnXYLeft = True
```

(8) IsShowBtnXYMenu property

IsShowBtnXYMenu

Whether display the import and export button

Define

Boolean IsShowBtnXYMenu

Example

Display the import and export button

VBScript Example

```
XYchart0.IsShowBtnXYMenu = True
```

(9) IsShowBtnXYPrint property

IsShowBtnXYPrint

Whether display the print button

Define

Boolean IsShowBtnXYPrint

Example

Display the print button

VBScript Example

```
XYchart0.IsShowBtnXYPrint = True
```

(10) IsShowBtnXYRelConfig property

IsShowBtnXYRelConfig

Whether display the real-time curve configuration button

Define

Boolean IsShowBtnXYRelConfig

Example

Display the real-time curve configuration button

VBScript Example

```
XYchart0.IsShowBtnXYRelConfig = True
```

(11) IsShowBtnXYRight property

IsShowBtnXYRight

Whether display the right shift button

Define

Boolean IsShowBtnXYRight

Example

Display the right shift button

VBScript Example
XYchart0.IsShowBtnXYRight = True

(12) IsShowBtnXYSave property

IsShowBtnXYSave

Whether display the save button

Define

Boolean IsShowBtnXYSave

Example

Display the save button

VBScript Example
XYchart0.IsShowBtnXYSave = True

(13) IsShowBtnXYStop property

IsShowBtnXYStop

Whether display the start or stop button

Define

Boolean IsShowBtnXYStop

Example

Display the start or stop button

VBScript Example
XYchart0.IsShowBtnXYStop = True

(14) IsShowBtnXYUp property

IsShowBtnXYUp

Whether display the up button

Define

Boolean IsShowBtnXYUp

Example

Display the up button

VBScript Example
XYchart0.IsShowBtnXYUp = True

(15) IsShowHtBtnXYConfig property

IsShowHtBtnXYConfig

Whether display the history curve configuration button

Define

Boolean IsShowHtBtnXYConfig

Example

Display the history curve configuration button

VBScript Example
XYchart0.IsShowHtBtnXYConfig = True

(16) IsShowHtBtnXYQuery property

IsShowHtBtnXYQuery

Whether display the history curve query button

Define

Boolean IsShowHtBtnXYQuery

Example

Display the history curve query button

VBScript Example
XYchart0.IsShowHtBtnXYQuery = True

(17) IsShowStatusBar property

IsShowStatusBar

Whether display the status column

Define

Boolean IsShowStatusBar

Example

Display the status column

VBScript Example
XYchart0.IsShowStatusBar = True

(18) IsShowXYCbTime property

IsShowXYCbTime

Whether display the quick time setting button

Define

Boolean IsShowXYCbTime

Example

Display the quick time setting button

VBScript Example
XYchart0.IsShowXYCbTime = True

(19) IsShowXYSetTime property

IsShowXYSetTime

Whether display the time setting button

Define

Boolean IsShowXYSetTime

Example

Display the time setting button

VBScript Example

```
XYchart0.IsShowXYSetTime = True
```

(20) Left property

Left

The left coordinate

Define

Double Left

Example

Object's left side of the coordinate values is 100 on the screen

VBScript Example

```
XYchart0.Left = 100
```

(21) Name property

Name

Name

Define

String Name

Example

Get the default name of the specified object

VBScript Example

```
TextBox0.Text = XYchart0.Name
```

(22) ToolTip property

ToolTip

Tooltip text

Define

String ToolTip

Example

Set the ToolTip of the specified object to "test"

VBScript Example
XYchart0.ToolTip = "test"

(23) Top property

Top

The top coordinate

Define

Double Top

Example

Object's the top of the coordinate value is 100 in the picture

VBScript Example
XYchart0.Top = 100

(24) Width property

Width

Width

Define

Double Width

Example

Set the width of the specified object to 100

VBScript Example
XYchart0.Width = 100

(25) XAxisInterval property

XAxisInterval

Get or set the large scale of X axis(The interval between two adjacent large scale lines)

Define

Double XAxisInterval

Example

Set the large scale of X axis to 20

VBScript Example
XYchart0.XAxisInterval = 20

(26) XAxisMaximum property

XAxisMaximum

Get or set the maximum value of X axis

Define

Double XAxisMaximum

Example

Set the maximum value of X axis to120

VBScript Example
XYchart0.XAxisMaximum = 120

(27) XAxisMinimum property

XAxisMinimum

Get or set the minimum value of X axis

Define

Double XAxisMinimum

Example

Set the minimum value of X axis to 20

VBScript Example
XYchart0.XAxisMinimum = 20

(28) XAxisSmallInterval property

XAxisSmallInterval

Get or set the small scale of X axis(The number of small scale lines between two adjacent large scale lines.)

Define

Double XAxisSmallInterval

Example

Set the small scale of X axis to 4

VBScript Example
XYchart0.XAxisSmallInterval = 4

(29) YAxisInterval property

YAxisInterval

Get or set the large scale of Y axis(The interval between two adjacent large scale lines)

Define

Double YAxisInterval

Example

Set the large scale of Y axis to 20

VBScript Example
XYchart0.YAxisInterval = 20

(30) YAxisMaximum property

YAxisMaximum

Get or set the maximum value of Y axis

Define

Double YAxisMaximum

Example

Set the maximum value of Y axis to 120

VBScript Example
XYchart0.YAxisMaximum = 120

(31) YAxisMinimum property

YAxisMinimum

Get or set the minimum value of Y axis

Define

Double YAxisMinimum

Example

set the minimum value of Y axis to 20

VBScript Example

```
XYchart0.YAxisMinimum = 20
```

(32) YAxisSmallInterval property

YAxisSmallInterval

Get or set the minimum scale of Y axis(The number of small scale lines between two adjacent large scale lines.)

Define

Double YAxisSmallInterval

Example

Set the small scale of Y axis to 4

VBScript Example

```
XYchart0.YAxisSmallInterval = 4
```

(33) ZIndex property

ZIndex

Layer index

Define

Int32 ZIndex

Example

Set the Layer index of the specified object to 3

VBScript Example

```
XYchart0.ZIndex = 3
```

4. HMICustomChart object

HMICustomChart

Customized chart control

Methods list

	Name	Description
	AddNewLimitLine	Add new limit line
	CalculateValue	Calculate the specific values of the upper and lower control lines
	CategoryPointCount	Each curve is associated with a variable, and refresh at the same time
	ContrastDataLoad	Time and data load contrastively
	CustomAxisLabel	Set coordinates of string axis
	DataTableDataLoad	The loading of the curve whose data source is DataTable
	ExportHistorySeriesData	Export history curve with report template
	ExportItemSourceAsDataTable	Export the curve whose data source is DataTable
	ExportRelSeriesData	Export real-time curve with report template
	ExportSeriesData	Export customized curves
	FindAnimation	Look for animation to modify the associated variables of the animation
	HistorySeries	Load history curve
	HistorySortXYSeries	Load sorted historical XY curve
	HistoryXYSeries	Load history XY curve
	ImportSeriesData	Import curves
	Print	Print
	QueryDataXYFromDataBase	Query history XY curve
	RealTimeFixedTimeSeries	Column graph superposition curve
	RealTimeXYPoint	Load real-time XY data
	RelTimeSeries	Query real-time curve
	RelTimeXYSeries	Query real-time XY curve
	RelTimeXYUpdateSeries	Load points of the real-time XY curve

	RemoveAllLimitLine	Remove all limit line
	Save	Save
	SetActiveSeriesNo	Set current curve
	SetChartHeaderColor	Set title Color of the chart
	SetChartLegendPosition	Set legend position of the chart
	SetChartMajorGridLine	Setting the major grid line
	SetChartMinorGridLine	Setting the major grid line
	SetPeroidRefreshTime	Set the refresh cycle of the curve
	SetPeroidStartTime	Set the start time
	SetPeroidTimeRange	Set time range
	SetSeriesLegendDisplay	Set whether to display legend of a curve
	SetSymbolTextColor	Set symbol text color of the curve
	SetTableNameForColumns	Get the column name of data source in the data table
	SetVarRecordRulerName	Set the name of query rule
	SetXAxisRange	Initialize the transverse
	SetYAxisAutoChanged	Set whether the vertical axis can automatically adjust
	ShowQueryChartTimeBetweenData	Query of the history curve
	ShowRelativeChartTime	Contrast of the real time history curve
	StartSeries	Start loading curve
	StopSeries	Stop loading curve

Property list

	Name	Description
	Height	Height
	IsShow	Whether display the specified object or not
	Left	The left coordinate
	Name	Name
	ToolTip	Tooltip text
	Top	The top coordinate
	Width	Width
	ZIndex	Layer index

The following is the detailed description of the script:

【Method】

(1) AddNewLimitLine method

AddNewLimitLine

Add new limit line (The "UpperAndLowerDisplay" is checked in the control property)

Define

AddNewLimitLine(variablePath ,defaultValue ,color , thickness)

Parameter

Name	Required/Optional	Data Type	Description
variablePath	Required	**String**	Variable path
defaultValue	Required	**Double**	Default value (default)
color	Required	**Brush**	Line fill color
thickness	Required	**Double**	Line thicknesses

Example

Add a new limit line

VBScript Example
Call CustomChart0.AddNewLimitLine("Var.variable",0,Colors.Red,5)

(2) CalculateValue method

CalculateValue

Calculate the specific values of the upper and lower control lines

Define

CalculateValue(count , variablePath1 , variablePath2 , number)

Parameter

Name	Required/Optional	Data Type	Description
count	Required	**Int**	Value count
variablePath1	Required	**String**	Variable path
variablePath2	Required	**String**	Variable path
number	Required	**Int**	The multiple of the standard deviation

Example

Calculate the specific values of the upper and lower control lines

Note: Each 20 data is refreshed, then calculate specified value and assigned to two Variable;

The upper and lower limit is related to Var.Variable and Var.Variable1 respectively.

VBScript Example
Call CustomChart0.CalculateValue(20,"Var.Variable","Var.Variable1",1)

(3) CategoryPointCount method

CategoryPointCount

Each curve is associated with a variable, and refresh at the same time

Define

CategoryPointCount(seriesno ,cnames , variablePaths)

Parameter

Name	Required/Optional	Data Type	Description
seriesno	Required	**Int**	The curve index
cnames	Required	**String**	Data source name of the horizontal axis
variablePaths	Required	**String**	Variable path

Example

The custom curve refresh real-timely

VBScript Example
Call CustomChart0.CategoryPointCount(0,"0,1,2","Var.Custom.variable,Var.Custom.variable1,Var.Custom.variable2") Call CustomChart0.SetPeroidRefreshTime(1000)

(4) ContrastDataLoad

ContrastDataLoad

Time and data load contrastively

Define

ContrastDataLoad(table , timeInterval , labindex , indexs)

Parameter

Name	Required/Optional	Data Type	Description
table	Required	**DataTable**	Data table of data source
timeInterval	Required	**Int**	Time interval of data loading , default is 1
laindex	Required	**Int**	The index of the column of the vertical axis data source
indexs	Required	**String**	The corresponding index of a curve data source, separated by commas

Example

Time and data load contrastively

Note: the“test.xlsx” sample as shown in the following figure

A	B	C	D	E
Name	var1	var2	var3	var4
aaa	2017/5/10	2017/6/11	2017/7/10	2017/8/10
bbb	2017/5/11	2017/6/12	2017/7/10	2017/8/10
ccc	2017/5/12	2017/6/13	2017/7/10	2017/8/10
ddd	2017/5/10	2017/6/11	2017/7/10	2017/8/10
eee	2017/5/11	2017/6/12	2017/7/10	2017/8/10
fff	2017/5/12	2017/6/10	2017/7/10	2017/8/10
ggg	2017/5/1	2017/6/10	2017/7/10	2017/8/10

VBScript Example

```
dt= ReportCmd.DirectImportToDataTable("E:\test.xlsx") 'Import Excel to DataTable
Call CustomChart0.ContrastDataLoad(dt,1,0,"1,2,3")
```

(5) CustomAxisLabel

CustomAxisLabel

Set coordinates of string axis

Define

CustomAxisLabel(index , cnames)

Parameter

Name	Required/Optional	Data Type	Description
index	Required	**Int**	The value is 0 or 1, indicating that the String axis is a horizontal axis or a vertical axis.
cnames	Required	**String**	The coordinates, separated by commas

Example

Set coordinates of string Axis : one，two，three

Note：The sample of "customchart.xlsx" as shown in the following figure

A	B
a	b
a1	10
a2	50
a3	30

VBScript Example

```
dt= ReportCmd.DirectImportToDataTable("E:\customchart.xlsx") 'Import Excel to DataTable
Call CustomChart0.DataTableDataLoad(dt,0,0,1)
Call CustomChart0.CustomAxisLabel(0,"one,two,three")
```

(6) DataTableDataLoad

DataTableDataLoad

The loading of the curve whose data source is DataTable

Define

DataTableDataLoad(table , seriesIndex , xbingingIndex , ybingingIndex)

Parameter

Name	Required/Optional	Data Type	Description
table	Required	**DateTable**	Data table of data source
seriesIndex	Required	**Int**	The curve index
xbingingIndex	Required	**Int**	The index of the column of the horizontical axis data source

ybingingIndex	Required	**Int**	The index of the column of the Vertical axis data source

Example

The loading of the curve whose data source is DataTable

Note：The sample of "customchart.xlsx" as shown in the following figure

A	B
a	b
a1	10
a2	50
a3	30

VBScript Example

```
dt= ReportCmd.DirectImportToDataTable("E:\customchart.xlsx") 'Import Excel to DataTable
Call CustomChart0.DataTableDataLoad(dt,0,0,1)
```

(7) ExportHistorySeriesData

ExportHistorySeriesData

Export history curve with report template

Define

ExportHistorySeriesData(seriesName ,templateName , variableList)

Parameter

Name	Required/Optional	Data Type	Description
seriesName	Required	**String**	curve name
templateName	Required	**String**	template name
variableList	Required	**String**	variable name

Example

Export the history curve Series0，Associated variables are："Var.variable0"

VBScript Example

```
Call CustomChart0.ExportHistorySeriesData("Series0","report0","Var.Variable0")
```

(8) ExportItemSourceAsDataTable

ExportItemSourceAsDataTable

Export the curve whose data source is DataTable

Define

ExportItemSourceAsDataTable(seriesName)

Parameter

Name	Required/Optional	Data Type	Description
seriesName	Required	**String**	The curve name

Example

Export history curve

Note：the data source of Series0 is DataTable

VBScript Example

```
Call CustomChart0.ExportItemSourceAsDataTable("Series0")
```

(9) ExportRelSeriesData

ExportRelSeriesData

Export real-time curve with report template

Define

ExportRelSeriesData(seriesName ,templateName)

Parameter

Name	Required/Optional	Data Type	Description
seriesName	Required	**String**	The curve name
templateName	Required	**String**	The template name

Example

Export real-time curve with report template

VBScript Example

```
Call CustomChart0.ExportRelSeriesData("Series0","report0")
```

(10) ExportSeriesData

ExportSeriesData

Export customized curves

Define

ExportSeriesData(seriesName)

Parameter

Name	Required/Optional	Data Type	Description
seriesName	Required	**String**	The name of curves,separate by commas

Example

Export customized curves Series0，Series1

VBScript Example

```
Call CustomChart0.ExportSeriesData("Series0,Series1")
```

(11) FindAnimation method

FindAnimation

Look for animation to modify the associated variables of the animation

Define

FindAnimation(name)

Parameter

Name	Required/Optional	Data type	Description
name	Required	**String**	The name of animation

Example

Change the associated variables of the visibility animation of the object into Var.VariableGroup0.Variable1

VBScript Example

```
CustomChart0.FindAnimation("HMIShowHideAnimation").Expression =
"Var.VariableGroup0.Variable1"
```

(12) HistorySeries method

HistorySeries

Load history curve

Define

HistorySeries(seriesno , startTime , endTime , intervalTime , intervalUnit , variablePath)

Parameter

Name	Required/Optional	Data Type	Description
seriesno	Required	**Int**	The curve index
startTime	Required	**DateTime**	Query start time
endTime	Required	**DateTime**	Query end time
intervalTime	Required	**long**	Query interval
intervalUnit	Required	**String**	Query interval unit
variablePath	Required	**String**	Variable path

Example

1.Query the historical data(Timing record)

VBScript Example

```
Call CustomChart0.SetVarRecordRulerName("s1")
Call CustomChart0.HistorySeries(2,DateTimePicker0.ValueTime,DateTimePicker1.ValueTime,1,"s","V
arRecord.RecordVariable1")
```

2.Query the historical data(Change record)

VBScript Example

```
Call CustomChart0.HistorySeries(4,DateTimePicker0.ValueTime,DateTimePicker1.ValueTime,1,"s","V
arRecord.RecordVariable2")
```

(13) HistorySortXYSeries method

HistorySortXYSeries

Load sorted historical XY curve

Define

HistorySortXYSeries(seriesno , startTime , endTime , intervalTime , intervalUnit , variableXPaths , variableYPaths)

Parameter

Name	Required/Optional	Data Type	Description
seriesno	Required	**Int**	The curve index
startTime	Required	**DateTime**	Query start time
endTime	Required	**DateTime**	Query end time
intervalTime	Required	**long**	Query interval
intervalUnit	Required	**String**	Query interval unit
variableXPath	Required	**String**	X Variable path
variableYPath	Required	**String**	Y Variable path

Example

1.Load sorted historical XY curve(Timing record)

VBScript Example
Call CustomChart0.SetVarRecordRulerName("s1") Call CustomChart0.HistorySortXYSeries(3,DateTimePicker0.ValueTime,DateTimePicker1.ValueTime,1,"s","VarRec

2.Load sorted historical XY curve(changed record)

VBScript Example
Call CustomChart0.HistorySortXYSeries(5,DateTimePicker0.ValueTime,DateTimePicker1.ValueTime,1,"s","VarRec

(14) HistoryXYSeries method

HistoryXYSeries

Load history XY curve

Define

HistoryXYSeries(seriesno , startTime , endTime , intervalTime , intervalUnit , variableXPaths , variableYPaths)

Parameter

Name	Required/Optional	Data Type	Description
seriesno	Required	**Int**	Curve index
startTime	Required	**DateTime**	Query start time
endTime	Required	**DateTime**	Query end time
intervalTime	Required	**long**	Query interval
intervalUnit	Required	**String**	Query interval unit
variableXPath	Required	**String**	X variable path
variableYPath	Required	**String**	Y variable path

Example

1.Load sorted historical XY curve(Timing record)

VBScript Example
Call CustomChart0.SetVarRecordRulerName("s1") Call CustomChart0.HistoryXYSeries(3,DateTimePicker0.ValueTime,DateTimePicker1.ValueTime,1,"s","VarRecord.

2.Load sorted historical XY curve(changed record)

VBScript Example
Call CustomChart0.HistoryXYSeries(5,DateTimePicker0.ValueTime,DateTimePicker1.ValueTime,1,"s","VarRecord.

(15) ImportSeriesData method

ImportSeriesData

Import curves

Define

ImportSeriesData(seriesIndex ,XAxisType)

Parameter

Name	Required/Optional	Data Type	Description
seriesIndex	Required	**String**	Curve index（When there are more curves, separate by commas）
XAxisType	Required	**String**	Horizontal axis classification （"Time","Double","String"）

Example

Import Series0 curve，horizontal axis type is"Time"

VBScript Example
Call CustomChart0.ImportSeriesData("0","Time")

(16) Print method

Print

Print

Define

Print()

Example

Print customize curves

VBScript Example
Call CustomChart0.Print()

(17) QueryDataXYFromDataBase method

QueryDataXYFromDataBase

Query history XY curves（Historical cohort record the query script）

Define

QueryDataXYFromDataBase(seriesno , startTime , endTime , count)

Parameter

Name	Required/Optional	Data Type	Description

seriesno	Required	**Int**	Curve index
startTime	Required	**DateTime**	Query start time
endTime	Required	**DateTime**	Query end time
count	Required	**Int**	Query the front part of data,-1 means all the data

Example

Query history XY curves

VBScript Example

```
Call CustomChart0.SetTableNameForColumns("aaa","history record","history record1",0)
Call CustomChart0.QueryDataXYFromDataBase(0,DateTimePicker0.ValueTime,DateTimePicker1.ValueTime,-1)
```

(18) RealTimeFixedTimeSeries method

RealTimeFixedTimeSeries

Column graph superposition curve

Define

RealTimeFixedTimeSeries(seriesno , tableName , startTime , endTime , columnNames , rangeFormat)

Parameter

Name	Required/Optional	Data Type	Description
seriesno	Required	**Int**	The curve index
tableName	Required	**String**	Database table name
startTime	Required	**DateTime**	Query start time
endTime	Required	**DateTime**	Query end time
columnNames	Required	**String**	Column name
rangeFormat	Required	**Int**	Format: 2 represents the period is year; 3 represents the period is month

Example

Column graph superposition curve(histoty group record)

VBScript Example

```
Call CustomChart0.RealTimeFixedTimeSeries("0,1,2","bbb",DateTimePicker8.ValueTime,DateTimePicker9.ValueTi
Call CustomChart0.SetPeroidRefreshTime(1000)
```

(19) RealTimeXYPoint method

RealTimeXYPoint

Load real-time XY data（History group record the query script)

Define

RealTimeXYPoint(seriesno , interval , startLocation , endLocation)

Parameter

Name	Required/Optional	Data Type	Description
seriesno	Required	**Int**	Curve index
interval	Required	**Double**	Horizontal axis spacing
startLocation	Required	**Double**	Horizontal axis minimum
endLocation	Required	**Double**	Horizontal axis maximum

Example

Load real-time XY data

VBScript Example

```
Call CustomChart0.SetTableNameForColumns("aaa","RecordVariable","RecordVariable1",0)
Call CustomChart0.RealTimeXYPoint(0,10,0,100)
Call CustomChart0.SetPeroidRefreshTime(1000)
```

(20) RealTimeSeries method

RealTimeSeries

Query real-time curve

Define

RelTimeSeries(seriesno , variablePaths)

Parameter

Name	Required/Optional	Data Type	Description
seriesno	Required	**Int**	Curve index

variablePaths	Required	**Double**	Variable index

Example

Query real-time curve

VBScript Example
Call CustomChart0.SetPeroidTimeRange(100) Call CustomChart0.RelTimeSeries(0,"Var.variable") Call CustomChart0.SetPeroidRefreshTime(1000)

(21) RelTimeXYSeries method

RelTimeXYSeries

Query real-time XY curve

Define

RelTimeSeries(seriesno , variableXPaths , variableYPaths)

Parameter

Name	Required/Optional	Data Type	Description
seriesno	Required	**Int**	Curve index
variableXPaths	Required	**String**	X variable path
variableYPaths	Required	**String**	Y variable path

Example

Query real-time XY curve

VBScript Example
Call CustomChart0.RelTimeXYSeries(0,"Var.variable","Var.variable1") Call CustomChart0.SetPeroidRefreshTime(1000)

(22) RelTimeXYUpdateSeries method

RelTimeXYUpdateSeries

Load points of the real-time XY curve

Define

RelTimeXYUpdateSeries(seriesno , variableCPaths , variableXPaths，variableYPaths)

Parameter

Name	Required/Optional	Data Type	Description
seriesno	Required	**Int**	Curve index
variableCPaths	Required	**String**	The variable path to clear the data flag
variableXPaths	Required	**Double**	X variable path
variableYPaths	Required	**Double**	Y variable path

Example

Load points of the real-time XY curve

VBScript Example
Call CustomChart0.SetPeroidRefreshTime(1000) Call CustomChart0.RelTimeXYUpdateSeries(1,"Var.VAriable2","Var.CustomChart.custom","Var.CustomChart.custo

(23) RemoveAllLimitLine method

RemoveAllLimitLine

Remove all limit line

Define

RemoveAllLimitLine()

Example

Remove all limit line

VBScript Example
Call CustomChart0.RemoveAllLimitLine()

(24) Save method

Save

Save

Define

Save()

Example

Save customize curve

VBScript Example
CustomChart0.Save()

(25) SetActiveSeriesNo method

SetActiveSeriesNo

Set current curve

Define

SetActiveSeriesNo(seriesno)

Parameter

Name	Required/Optional	Data Type	Description
seriesno	Required	**Int**	Curve index，Start from 0

Example

Set current curve

VBScript Example
Call CustomChart0.SetActiveSeriesNo(1)

(26) SetChartHeaderColor method

SetChartHeaderColor

Set title Color of the chart

Define

SetChartHeaderColor(color)

Parameter

Name	Required/Optional	Data Type	Description
color	Required	**Brush**	Setting color

Example

Set title Color of the chart

VBScript Example
Call CustomChart0.SetChartHeaderColor(Colors.Blue)

(27) SetChartLegendPosition method

SetChartLegendPosition

Set legend position of the chart

Define

SetChartLegendPosition(positionIndex , InsideOrOutside)

Parameter

Name	Required/Optional	Data Type	Description
positionIndex	Required	**Int**	The value is 0,1,2,3, indicating the position is left, upper, right, and lower.
InsideOrOutside	Required	**Int**	The value is 0 or 1, indicating that the legend is outside or inside the chart.

Example

Set the legend position in the right side of the chart

VBScript Example
Call CustomChart0.SetChartLegendPosition(2,1)

(28) SetChartMajorGridLine method

SetChartMajorGridLine

Setting the major grid line

Define

SetChartMajorGridLine(color , thickness , collection)

Parameter

Name	Required/Optional	Data Type	Description
color	Required	**Brush**	Grid line color

thickness	Required	**Double**	Grid line thickness
collection	Required	**String**	Dotted line format, separate by a commas. Example"2,4"

Example

Setting the major grid line

VBScript Example
Call CustomChart0.SetChartMajorGridLine(Colors.Pink,3,"2,4")

(29) SetChartMinorGridLine method

SetChartMinorGridLine

Setting the minor grid line

Define

SetChartMinorGridLine(color , thickness，collection)

Parameter

Name	Required/Optional	Data Type	Description
color	Required	**Brush**	Grid line color
thickness	Required	**Double**	Grid line thickness
collection	Required	**String**	Dotted line format, separate by a commas. Example"2,4"

Example

Setting the minor grid line

VBScript Example
Call CustomChart0.SetChartMinorGridLine(Colors.YellowGreen,1,"3,5")

(30) SetPeroidRefreshTime method

SetPeroidRefreshTime

Set the curve refresh interval

Define

SetPeroidRefreshTime(ms)

Parameter

Name	Required/Optional	Data Type	Description
ms	Required	**Int**	unit ms

Example

query real-time curve

VBScript Example
Call CustomChart0.SetPeroidTimeRange(100) Call CustomChart0.RelTimeSeries(0,"Var.Varible") Call CustomChart0.SetPeroidRefreshTime(1000)

(31) SetPeroidStartTime method

SetPeroidStartTime

Set start time

Define

SetPeroidStartTime(Time)

Parameter

Name	Required/Optional	Data Type	Description
ms	Required	**Int**	Unit ms

Example

Set start time

VBScript Example
Call CustomChart0.SetPeroidStartTime()

(32) SetPeroidTimeRange method

SetPeroidTimeRange

Set time range

Define

SetPeroidTimeRange(ms)

Parameter

Name	Required/Optional	Data Type	Description
ms	Required	**Int**	Unit ms

Example

Set time range

VBScript Example
Call CustomChart0.SetPeroidTimeRange(100)

(33) SetSeriesLegendDisplay method

SetSeriesLegendDisplay

Set whether to display legend of a curve

Define

SetSeriesLegendDisplay(seriesIndex , isShow)

Parameter

Name	Required/Optional	Data Type	Description
seriesIndex	Required	**Int**	Curve index
isShow	Required	**Bool**	True:display legend;False:do not display legend

Example

Do not display legend of the curve0

VBScript Example
Call CustomChart0.SetSeriesLegendDisplay(0,False)

(34) SetSymbolTextColor method

SetSymbolTextColor

Set symbol text color of the curve

Define

SetSymbolTextColor(serisesIndex , color)

Parameter

Name	Required/Optional	Data Type	Description
seriesIndex	Required	**Int**	Curve index
color	Required	**Brush**	Symbol text color

Example

Set symbol text color of the curve to red

VBScript Example
Call CustomChart0.SetSymbolTextColor(0,Colors.Red)

(35) SetTableNameForColumns method

SetTableNameForColumns

Get the column name of data source in the data table

Define

SetTableNameForColumns(tableName ，xcolumnName , ycolumnName , fixType)

Parameter

Name	Required/Optional	Data Type	Description
tableName	Required	**String**	History group name
xcolumnName	Required	**String**	The horizontal axis data source column name（The name of history record、"Time"）
ycolumnName	Required	**String**	The vertical axis data source column name，that is name of history record
Type	Required	**Int**	0 means fixed X axis；1 meansvariable X axis

Example

1.Get the column name of data source in the data table

VBScript Example

```
Call CustomChart0.SetTableNameForColumns("aaa","Time","RecordVariable",0)
```

2.Get the column name of data source in the data table

VBScript Example

```
Call CustomChart0.SetTableNameForColumns("aaa","RecordVariable","RecordVariable1",0)
```

(36) SetVarRecordRulerName method

SetVarRecordRulerName

Set the name of query rule

Define

SetVarRecordRulerName(RuleName)

Parameter

Name	Required/Optional	Data Type	Description
RuleName	Required	**String**	The name of timer

Example

1.Set the name of query rule，s1 is the timer name

VBScript Example

```
Call CustomChart0.SetVarRecordRulerName("s1")
```

2.Query the historical data(Timing record)

VBScript Example

```
Call CustomChart0.SetVarRecordRulerName("s1")
Call CustomChart0.HistorySeries(2,DateTimePicker0.ValueTime,DateTimePicker1.ValueTime,1,"s","VarRecord.RecordVariable1")
```

(37) SetXAxisRange method

SetXAxisRange

Initialize the transverse，used to initialize icons' transverse range such as the annual report, monthly report and weekly report etc, and curve points information

Define

SetXAxisRange(range)

Parameter

Name	Required/Optional	Data Type	Description
range	Required	**Int**	The values are 0、1、2、3、4、5，correspond year、month、week、day、hour、minute

Example

Initialize the transverse

VBScript Example
Call CustomChart0.SetXAxisRange(0)

(38) SetYAxisAutoChanged method

SetYAxisAutoChanged

Set whether the vertical axis can automatically adjust

Define

SetYAxisAutoChanged(autoChanged)

Parameter

Name	Required/Optional	Data Type	Description
autoChanged	Required	**Bool**	True or False

Example

Set the vertical axis can automatically adjust

VBScript Example
Call CustomChart0.SetYAxisAutoChanged(True)

(39) ShowQueryChartTimeBetweenData method

ShowQueryChartTimeBetweenData

Query history curve

Define

ShowQueryChartTimeBetweenData(seriesno ，startTime , endTime)

Parameter

Name	Required/Optional	Data Type	Description
seriesno	Required	**Int**	Curve index
startTime	Required	**DateTime**	Query start time
endTime	Required	**DateTime**	Query end time

Example

Query history curve

VBScript Example

```
Call CustomChart0.SetTableNameForColumns("aaa","Time","RecordVariable",0)
Call CustomChart0.ShowQueryChartTimeBetweenData(0,DateTimePicker0.ValueTime,DateTimePicker1.ValueTim
```

(40) ShowRelativeChartTime method

ShowRelativeChartTime

Real-time curve configuration

Define

ShowRelativeChartTime(seriesno ，startTime , endTime ，rstartTime , rendTime)

Parameter

Name	Required/Optional	Data Type	Description
seriesno	Required	**Int**	Curve index collection，separated by commas
startTime	Required	**DateTime**	History start time
endTime	Required	**DateTime**	History end time
rstartTime	Required	**DateTime**	Real time start time

rendTime	Required	**DateTime**	Real time history curve of contrast

Example

Compare real time curves and historical curves

VBScript Example

```
Call CustomChart0.SetTableNameForColumns("aaa","Time","RecordVariable",0)
Call CustomChart0.ShowRelativeChartTime("0,1",DateTimePicker0.ValueTime,DateTimePicker1.ValueTime,DateT
Call CustomChart0.SetPeroidRefreshTime(1000)
```

(41) StartSeries method

StartSeries

Start loading curves

Define

StartSeries()

Example

Start loading curves

VBScript Example

```
CustomChart0.StartSeries()
```

(42) StopSeries method

StopSeries

Stop loading curves

Define

StopSeries()

Example

Stop loading curves

VBScript Example

```
CustomChart0.StopSeries()
```

【property】

(1) Height property

Height

Height

Define

Double Height

Example

Set the height of the specified object to 100

VBScript Example
CustomChart0.Height = 100

(2) IsShow property

IsShow

Whether display the specified object or not

Define

Boolean IsShow

Example

True ：The specified object is displayed on the screen False ：The specified object hidden on the screen

VBScript Example
CustomChart0.IsShow = True CustomChart0.IsShow = False

(3) Left property

Left

The left coordinate

Define

Double Left

Example

Object's left side of the coordinate values is 100 on the screen

VBScript Example

```
CustomChart0.Left = 100
```

(4) Name property

Name

Name

Define

String Name

Example

Get the default name of the specified object

VBScript Example

```
Text0.Text = CustomChart0.Name
```

(5) ToolTip property

ToolTip

Tooltip text

Define

String ToolTip

Example

Set the ToolTip of the specified object to "test"

VBScript Example

```
CustomChart0.ToolTip = "test"
```

(6) Top property

Top

The top coordinate

Define

Double Top

Example

Object's the top of the coordinate value is 100 in the picture

VBScript Example
CustomChart0.Top = 100

(7) Width property

Width

Width

Define

Double Width

Example

Set the width of the specified object to 100

VBScript Example
CustomChart0.Width = 100

(8) ZIndex property

ZIndex

Layer index

Define

Int32 ZIndex

Example

Set the Layer index of the specified object to 2

VBScript Example
CustomChart0.ZIndex = 2

5. HMIPieChart object

HMIPieChart

Pie chart control

Methods list

	Name	Description
	FindAnimation	Look for animation to modify the associated variables of the animation

Property list

	Name	Description
	Height	Height
	IsShow	Whether display the specified object or not
	Left	The left coordinate
	Name	Name
	ToolTip	Tooltip text
	Top	The top coordinate
	Width	Width
	ZIndex	Layer index

The following is the detailed description of the script:

【Method】

(1) FindAnimation method

FindAnimation

Look for animation to modify the associated variables of the animation

Define

FindAnimation(name)

Parameter

Name	Required/Optional	Data type	Description
name	Required	**String**	The name of animation

Example

Change the associated variables of the visibility animation of the object into Var.VariableGroup0.Variable1

VBScript Example

```
PieChart0.FindAnimation("HMIShowHideAnimation").Expression = "Var.VariableGroup0.Variable1"
```

【property】

(1) Height property

Height

Height

Define

Double Height

Example

Set the height of the specified object to 100

VBScript Example

```
PieChart0.Height = 100
```

(2) IsShow property

IsShow

Whether display the specified object or not

Define

Boolean IsShow

Example

True ：The specified object is displayed on the screen False ：The specified object hidden on the screen

VBScript Example

```
PieChart0.IsShow = True     PieChart0.IsShow = False
```

(3) Left property

Left

The left coordinate

Define

Double Left

Example

Object's left side of the coordinate values is 100 on the screen

VBScript Example
PieChart0.Left = 100

(4) Name property

Name

Name

Define

String Name

Example

Get the default name of the specified object

VBScript Example
Text0.Text = PieChart0.Name

(5) ToolTip property

ToolTip

Tooltip text

Define

String ToolTip

Example

Set the ToolTip of the specified object to "test"

VBScript Example
PieChart0.ToolTip = "test"

(6) Top property

Top

The top coordinate

Define

Double Top

Example

Object's the top of the coordinate value is 100 in the picture

VBScript Example
PieChart0.Top = 100

(7) Width property

Width

Width

Define

Double Width

Example

Set the width of the specified object to 100

VBScript Example
PieChart0.Width = 100

(8) ZIndex property

ZIndex

Layer index

Define

Int32 ZIndex

Example

Set the Layer index of the specified object to 2

VBScript Example
PieChart0.ZIndex = 2

6. HMIPieChart3D object

HMIPieChart3D

3D pie chart control

Methods list

	Name	Description
	FindAnimation	Look for animation to modify the associated variables of the animation
	SetSeriesVariablePath	Curve correlation variable substitution
	StartTimer	Start timer
	StopTimer	Stop timer

Property list

	Name	Description
	Height	Height
	IsShow	Whether display the specified object
	Left	The left coordinate
	Name	Name
	Rotation	Get or set the 3D pie chart rotation angle
	Tilt	Get or set the 3D pie chart angle
	ToolTip	Tooltip text
	Top	The top coordinate
	Width	Width
	ZIndex	Layer index

The following is the detailed description of the script:

【Method】

(1) FindAnimation method

FindAnimation

Look for animation to modify the associated variables of the animation

Define

FindAnimation(name)

Parameter

Name	Required/Optional	Data type	Description
name	Required	**String**	The name of animation

Example

Change the associated variables of the visibility animation of the object into Var.VariableGroup0.Variable1

VBScript Example
PieChart3D0.FindAnimation("HMIShowHideAnimation").Expression = "Var.VariableGroup0.Variable1"

(2) SetSeriesVariablePath method

SetSeriesVariablePath

Curve correlation variable substitution

Define

SetSeriesVariablePath(pathList)

Parameter

Name	Required/Optional	Data Type	Description
pathList	Required	**String**	The new variable path of the variables that replace the old variables , separate by commas

Example

Curve correlation variable substitution

VBScript Example
Call PieChart3D0.SetSeriesVariablePath("Var.variable0,Var.variable1")

(3) StartTimer method

StartTimer

Start timer

Define

StartTimer()

Example

Start Timer

VBScript Example

```
Call PieChart3D0.StartTimer()
```

(4) StopTimer method

StopTimer

Stop timer

Define

StopTimer()

Example

Stop Timer

VBScript Example

```
Call PieChart3D0.StopTimer()
```

【property】

(1) Height property

Height

Height

Define

Double Height

Example

Set the height of the specified object to 100

VBScript Example

```
PieChart3D0.Height = 100
```

(2) IsShow property

IsShow

Whether display the specified object or not

Define

Boolean IsShow

Example

True ：The specified object is displayed on the screen False ：The specified object hidden on the screen

VBScript Example
PieChart3D0.IsShow = True PieChart3D0.IsShow = False

(3) Left property

Left

The left coordinate

Define

Double Left

Example

Object's left side of the coordinate values is 100 on the screen

VBScript Example
PieChart3D0.Left = 100

(4) Name property

Name

Name

Define

String Name

Example

Get the default name of the specified object

VBScript Example
Text0.Text = PieChart3D0.Name

(5) Rotation property

Rotation

Get or set the 3D pie chart rotation angle

Double Rotation

Print()

Example

Set the rotation Angle of the specified object to 90

VBScript Example
PieChart3D0.Rotation = 90

(6) Tilt property

Tilt

Get or set the 3D pie chart angle

Double Tilt

Print()

Example

Set the Angle of the specified object to 30

VBScript Example
PieChart3D0.Tilt = 30

(7) ToolTip property

ToolTip

Tooltip text

Define

String ToolTip

Example

Set the ToolTip of the specified object to "test"

VBScript Example
PieChart3D0.ToolTip = "test"

(8) Top property

Top

The top coordinate

Define

Double Top

Example

Object's the top of the coordinate value is 100 in the picture

VBScript Example
PieChart3D0.Top = 100

(9) Width property

Width

Width

Define

Double Width

Example

Set the width of the specified object to 100

VBScript Example
PieChart3D0.Width = 100

(10) ZIndex property

ZIndex

Layer index

Define

Int32 ZIndex

Example

Set the Layer index of the specified object to 2

VBScript Example
PieChart3D0.ZIndex = 2

7. HMIColumnChart object

HMIColumnChart

Column chart control

Methods list

	Name	Description
	AddNewLimitLine	Add a new limit line
	FindAnimation	Look for animation to modify the associated variables of the animation
	RemoveAllLimitLine	Remove all the limit lines

Property list

	Name	Description
	Height	Height
	IsShow	Whether display the specified object
	Left	The left coordinate
	Name	Name
	ToolTip	Tooltip text
	Top	The top coordinate
	Width	Width
	ZIndex	Layer index

The following is the detailed description of the script:

【Method】

(1) AddNewLimitLine method

AddNewLimitLine

Add a new limit line (The "UpperAndLowerDisplay" is checked in the control property)

Define

AddNewLimitLine(variablePath ,defaultValue ,color , thickness)

Parameter

Name	Required/Optional	Data Type	Description
variablePath	Required	**String**	Variable path
defaultValue	Required	**Double**	Default value (default)

color	Required	**Brush**	Line fill color
thickness	Required	**Double**	Line thicknesses

Example

Add new limit line

VBScript Example
Call ColumnChart0.AddNewLimitLine("Var.Variable",0,Colors.Red,5)

(2) FindAnimation method

FindAnimation

Look for animation to modify the associated variables of the animation

Define

FindAnimation(name)

Parameter

Name	Required/Optional	Data type	Description
name	Required	**String**	The name of animation

Example

Change the associated variables of the visibility animation of the object into Var.VariableGroup0.Variable1

VBScript Example
ColumnChart0.FindAnimation("HMIShowHideAnimation").Expression = "Var.VariableGroup0.Variable1"

(3) RemoveAllLimitLine method

RemoveAllLimitLine

Remove all the limit lines

Define

RemoveAllLimitLine()

Example

Remove all the limit line

VBScript Example
Call ColumnChart0.RemoveAllLimitLine()

【property】

(1) Height property

Height

Height

Define

Double Height

Example

Set the height of the specified object to 100

VBScript Example
ColumnChart0.Height = 100

(2) IsShow property

IsShow

Whether display the specified object or not

Define

Boolean IsShow

Example

True ：The specified object is displayed on the screen False ：The specified object hidden on the screen

VBScript Example
ColumnChart0.IsShow = True ColumnChart0.IsShow = False

(3) Left property

Left

The left coordinate

Define

Double Left

Example

Object's left side of the coordinate values is 100 on the screen

VBScript Example

```
ColumnChart0.Left = 100
```

(4) Name property

Name

Name

Define

String Name

Example

Get the default name of the specified object

VBScript Example

```
Text0.Text = ColumnChart0.Name
```

(5) ToolTip property

ToolTip

Tooltip text

Define

String ToolTip

Example

Set the ToolTip of the specified object to "test"

VBScript Example

```
ColumnChart0.ToolTip = "test"
```

(6) Top property

Top

The top coordinate

Define

Double Top

Example

Object's the top of the coordinate value is 100 in the picture

VBScript Example

```
ColumnChart0.Top = 100
```

(7) Width property

Width

Width

Define

Double Width

Example

Set the width of the specified object to 100

VBScript Example

```
ColumnChart0.Width = 100
```

(8) ZIndex property

ZIndex

Layer index

Define

Int32 ZIndex

Example

Set the Layer index of the specified object to 2

VBScript Example

```
ColumnChart0.ZIndex = 2
```

8. HMIHistoryColumnChart object

HMIHistoryColumnChart

History column chart control

Methods list

	Name	Description
	AddNewLimitLine	Add a new limit line
	FindAnimation	Look for animation to modify the associated variables of the animation
	Print	Print
	Query	Query history data
	Save	Save
	SetDateConditionListStartEndTime	Set the start and end time
	SetSeriesVariablePath	Replace the variables associated with the curve

Property list

	Name	Description
	Height	Height
	IsShow	Whether display the specified object or not
	Left	The left coordinate
	Name	Name
	NumberAxisMaximum	Get or set the maximum of the number axis
	NumberAxisMinimum	Get or set the minimum of the number axis
	NumberInterval	The large scale of the numerical axis
	NumberSmallInterval	The small scale of the numerical axis
	ToolTip	Tooltip text
	Top	The top coordinate
	Width	Width
	ZIndex	Layer index

The following is the detailed description of the script:

【Method】

(1) AddNewLimitLine method

AddNewLimitLine

Add a new limit line (upper and lower limit display in control properties is checked)

Define

AddNewLimitLine(variablePath ,defaultValue ,color , thickness)

Parameter

Name	Required/Optional	Data Type	Description
VariablePath	Required	**String**	Variable path
DefaultValue	Required	**Double**	Default value
Color	Required	**Brush**	Line fill color
Thickness	Required	**Double**	Line thicknesses

Example

Add a new limit line

VBScript Example
HistoryColumnChart0.AddNewLimitLine("Var.Variable0",0,Colors.Red,5)

(2) FindAnimation method

FindAnimation

Look for animation to modify the associated variables of the animation

Define

FindAnimation(name)

Parameter

Name	Required/Optional	Data type	Description
name	Required	**String**	The name of animation

Example

Change the associated variables of the visibility animation of the object into Var.VariableGroup0.Variable1

VBScript Example

```
HistoryColumnChart0.FindAnimation("HMIShowHideAnimation").Expression =
"Var.VariableGroup0.Variable1"
```

(3) Print method

Print

Print

Define

Print()

Example

Print the history histogram controls

VBScript Example

```
Call HistoryColumnChart0.Print()
```

(4) Query method

Query

Query history data

Define

Query()

Example

Query history data

Note : when the history is timing record，The query rules should be set up in the toolbar before querying data.

VBScript Example

```
Call
HistoryColumnChart0.SetDateConditionListStartEndTime(DateTimePicker0.ValueTime,DateTimePicker1.ValueTime
Call HistoryColumnChart0.Query()
```

(5) Save method

Save

Save

Define

Save()

Example

Save the history histogram controls

VBScript Example
Call HistoryColumnChart0.Save()

(6) SetDateConditonListStartEndTime method

SetDateConditionListStartEndTime

Set the start and end times

Define

SetDateConditionListStartEndTime(startTime , endTime)

Parameter

Name	Required/Optional	Data Type	Description
startTime	Required	**DateTime**	Start time
endTime	Required	**DateTime**	End time

Example

Set the start and end times

VBScript Example
Call HistoryColumnChart0.SetDateConditionListStartEndTime(DateTimePicker0.ValueTime,DateTimePicker1.ValueTime

(7) SetSeriesVariablePath method

SetSeriesVariablePath

Replace the variables associated with the curve

Define

SetSeriesVariablePath(pathList)

Parameter

Name	Required/Optional	Data type	Description
pathList	Required	**String**	The name of the new variable to be replaced in turn,separated by comma

Example

Replace the variables associated with the curve

VBScript Example
Call HistoryColumnChart0.SetSeriesVariablePath("Var.HColumn.Variable,Var.HColumn.Variable1")

【property】

(1) Height property

Height

Height

Define

Double Height

Example

Set the height of the specified object to 100

VBScript Example
HistoryColumnChart0.Height = 100

(2) IsShow property

IsShow

Whether display the specified object or not

Define

Boolean IsShow

Example

True ：The specified object is displayed on the screen False ：The specified object hidden on the screen

VBScript Example

```
HistoryColumnChart0.IsShow = True    HistoryColumnChart0.IsShow = False
```

(3) Left property

Left

The left coordinate

Define

Double Left

Example

Object's left side of the coordinate values is 100 on the screen

VBScript Example

```
HistoryColumnChart0.Left = 100
```

(4) Name property

Name

Name

Define

String Name

Example

Get the default name of the specified object

VBScript Example

```
TextBox0.Text = HistoryColumnChart0.Name
```

(5) NumericalAxisMaximum property

NumberAxisMaximum

Get or set the maximum of the number axis

Define

Double NumberAxisMaximum

Example

Set the maximum value of number axis to120

VBScript Example

```
HistoryColumnChart0.NumberAxisMaximum = 120
```

(6) NumericalAxisMinimum property

NumberAxisMinimum

Get or set the minimum of the number axis

Define

Double NumberAxisMinimum

Example

Set the minimum value of number axis to 20

VBScript Example

```
HistoryColumnChart0.NumberAxisMinimum = 20
```

(7) NumberInterval property

NumberInterval

The large scale of number axis(The interval between two adjacent large scale lines)

Define

Double NumberInterval

Example

Set the large scale of number axis to 20

VBScript Example

```
HistoryColumnChart0.NumberInterval = 20
```

(8) NumberSmallInterval

NumberSmallInterval

The small scale of number axis(The number of small scale lines between two adjacent large scale lines.)

Define

Double NumberSmallInterval

Example

Set the small scale of number axis to 4

VBScript Example
HistoryColumnChart0.NumberSmallInterval = 4

(9) ToolTip property

ToolTip

Tooltip text

Define

String ToolTip

Example

Set the ToolTip of the specified object to "test"

VBScript Example
HistoryColumnChart0.ToolTip = "test"

(10) Top property

Top

The top coordinate

Define

Double Top

Example

Object's the top of the coordinate value is 100 in the picture

VBScript Example
HistoryColumnChart0.Top = 100

(11) Width property

Width

Width

Define

Double Width

Example

Set the width of the specified object to 100

VBScript Example
HistoryColumnChart0.Width = 100

(12) ZIndex property

ZIndex

Layer index

Define

Int32 ZIndex

Example

Set the Zindex of the specified object to 3

VBScript Example
HistoryColumnChart0.ZIndex = 3

9. HMIRecordBox object

HMIRecordBox

Record box control

Methods list

	Name	Description
	DataExport	Export all the data
	FindAnimation	Look for animation to modify the associated variables of the animation
	Print	Print
	QueryHistoryDate	The inquiry of historical record
	Save	Save

Property list

	Name	Description
	Height	Height
	IsShow	Whether display the specified object or not

	IsShowBtnPrint	Whether to show the print button
	IsShowBtnSave	Whether to show the save button
	IsShowRdBtnConfig	Whether to show the setting button of column
	IsShowRdBtnUpdate	Whether to show the query button
	IsShowRdCbTime	Whether to show the Quick QueryTimeSet button
	IsShowRdRecordSelectedType	Whether to show the RecordSelectedType button
	IsShowRdRtExport	Whether to show the export button
	IsShowRdSetTime	Whether to show the QueryTimeSet button
	Left	The left coordinate
	Name	Name
	ToolTip	Tooltip text
	Top	The top coordinate
	Width	Width
	ZIndex	Layer index

The following is the detailed description of the script:

【Method】

(1) DataExport

DataExport

Export all the data

Define

DataExport(isAllExport)

Parameter

Name	Required/Optional	Data Type	Description
isAllExport	Required	**Boolean**	Whether to export all the data（true:export all the data，false:export the current page）

Example

Export all the data

VBScript Example

```
RecordBox0.DataExport(True)
```

(2) FindAnimation method

FindAnimation

Look for animation to modify the associated variables of the animation

Define

FindAnimation(name)

Parameter

Name	Required/Optional	Data type	Description
name	Required	**String**	The name of animation

Example

Change the associated variables of the visibility animation of the object into Var.VariableGroup0.Variable1

VBScript Example

```
RecordBox0.FindAnimation("HMIShowHideAnimation").Expression = "Var.VariableGroup0.Variable1"
```

(3) Print method

Print

Print

Define

Print()

Example

Print the record box

VBScript Example

```
RecordBox0.Print()
```

(4) QueryHistoryDate method

QueryHistoryDate

The inquiry of historical record

Define

QueryHistoryDate(StartTime ,EndTime)

Parameter

Name	Required/Optional	Data Type	Description
StartTime	Required	**DateTime**	Start time
EndTime	Required	**DateTime**	End time

Example

Query the data within the specified time period

VBScript Example
Call RecordBox0.QueryHistoryDate(DateTimePicker0.ValueTime,DateTimePicker1.ValueTime)

(5) Save method

Save

Save

Define

Save()

Example

Save the record box

VBScript Example
RecordBox0.Save()

【property】

(1) Height property

Height

Height

Define

Double Height

Example

Set the height of the specified object to 100

VBScript Example
RecordBox.Height = 100

(2) IsShow property

IsShow

Whether display the specified object or not

Define

Boolean IsShow

Example

True ：The specified object is displayed on the screen False ：The specified object hidden on the screen

VBScript Example
RecordBox.IsShow = True RecordBox.IsShow = False

(3) IsShowBtnPrint property

IsShowBtnPrint

Show the Print Button

Define

Boolean IsShowBtnPrint

Example

Whether the record box control to show the Print Button

VBScript Example
RecordBox0.IsShowBtnPrint = true

(4) IsShowBtnSave property

IsShowBtnSave

Show the Save Button

Define

Boolean IsShowBtnSave

Example

Whether the record box control to show the Save Button

VBScript Example

```
RecordBox0.IsShowBtnSave = true
```

(5) IsShowRdBtnConfig property

IsShowRdBtnConfig

Show the Column Setting button

Define

Boolean IsShowRdBtnConfig

Example

Whether the record box control to show the Column Setting button

VBScript Example

```
RecordBox0.IsShowRdBtnConfig = true
```

(6) IsShowRdBtnUpdate property

IsShowRdBtnUpdate

Show the query button

Define

Boolean IsShowRdBtnUpdate

Example

Whether the record box control to show the query button

VBScript Example

```
RecordBox0.IsShowRdBtnUpdate = true
```

(7) IsShowRdCbTime property

IsShowRdCbTime

Show the QuickQueryTimeSet button

Define

Boolean IsShowRdCbTime

Example

Whether the record box control to show the QuickQueryTimeSet button

VBScript Example

```
RecordBox0.IsShowRdCbTime = true
```

(8) IsShowRdRecordSelectedType property

IsShowRdRecordSelectedType

Show the RecordSelectedType button

Define

Boolean IsShowRdRecordSelectedType

Example

Whether the record box control to show the RecordSelectedType button

VBScript Example

```
RecordBox0.IsShowRdRecordSelectedType = true
```

(9) IsShowRdRtExport property

IsShowRdRtExport

Show the export button

Define

Boolean IsShowRdRtExport

Example

Whether the record box to show the export button

VBScript Example

```
RecordBox0.IsShowRdRtExport = true
```

(10) IsShowRdSetTime property

IsShowRdSetTime

Show the QueryTimeSet button

Define

Boolean IsShowRdSetTime

Example

Whether the record box control to show the QueryTimeSet button

VBScript Example

```
RecordBox0.IsShowRdSetTime = true
```

(11) Left property

Left

The left coordinate

Define

Double Left

Example

Object's left side of the coordinate values is 100 on the screen

VBScript Example

```
RecordBox0.Left = 100
```

(12) Name property

Name

Name

Define

String Name

Example

Get the default name of the specified object

VBScript Example

```
TextBox0.Text = RecordBox0.Name
```

(13) ToolTip property

ToolTip

Tooltip text

Define

String ToolTip

Example

Set the ToolTip of the specified object to "test"

VBScript Example

```
RecordBox0.ToolTip = "test"
```

(14) Top property

Top

The top coordinate

Define

Double Top

Example

Object's the top of the coordinate value is 100 in the picture

VBScript Example

```
RecordBox0.Top = 100
```

(15) Width property

Width

Width

Define

Double Width

Example

Set the width of the specified object to 100

VBScript Example
RecordBox0.Width = 100

(16) ZIndex property

ZIndex

Layer index

Define

Int32 ZIndex

Example

Set the Zindex of the specified object to 3

VBScript Example
RecordBox0.ZIndex = 3

10. HMIAlarmWindow object

HMIAlarmWindow

Alarm window control

Methods list

	Name	Description
	AckAlarm	Reply to the selected alarm
	AckAllAlarm	Reply all the alarm that record type is alarm
	AckAllRecovery	Reply all the alarm that record type is recover
	DataExport	Export the data
	FindAnimation	Look for animation to modify the associated variables of the animation
	GetCurrentAlarmColumnInf	Get the content of the currently selected alarm column
	GetHistoryAlarmItemDataTable	Get the data table of the historical data
	GetRowNum	Get the current number of real-time alarm Windows
	Print	Print
	QueryHistoryAlarm	The query of the historical alarm

	Save	Save
	SetCurrentSelectRowIndex	Set the currently selected item
	SetExportValue	Set the parameters of the alarm data export script
	StartRealTimeAlarm	Start loading the real-time alarm data
	StopRealTimeAlarm	Stop loading the real-time alarm data

Property list

	Name	Description
	Height	Height
	IsShow	Whether display the specified object or not
	IsShowHistoryTabItem	Whether to show the header of the history alarm
	IsShowRelTableItem	Whether to show the header of the real-time alarm
	Left	The left coordinate
	Name	Name
	ToolTip	Tooltip text
	Top	The top coordinate
	Width	Width
	ZIndex	Layer index

The following is the detailed description of the script:

【Method】

(1) AckAlarm method

AckAlarm

Reply to a selected alarm

Define

AckAlarm()

Example

Reply to a selected alarm

VBScript Example

```
Call AlarmWindow0.AckAlarm()
```

(2) AckAllAlarm method

AckAllAlarm

Reply all the alarm that record type is alarm

Define

AckAllAlarm()

Example

Reply all the alarm that record type is alarm

VBScript Example

```
Call AlarmWindow0.AckAllAlarm()
```

(3) AckAllRecovery method

AckAllRecovery

Reply all the alarm that record type is recover

Define

AckAllRecovery()

Example

Reply all the alarm that record type is recover

VBScript Example

```
Call AlarmWindow0.AckAllRecovery()
```

(4) DataExport method

DataExport

Export data

Define

DataExport()

Example

Export data

VBScript Example

```
Call AlarmWindow0.SetExportValue(False,True)
Call AlarmWindow0.DataExport()
```

(5) FindAnimation method

FindAnimation

Look for animation to modify the associated variables of the animation

Define

FindAnimation(name)

Parameter

Name	Required/Optional	Data type	Description
name	Required	**String**	The name of animation

Example

Change the associated variables of the visibility animation of the object into Var.VariableGroup0.Variable1

VBScript Example

```
AlarmWindow0.FindAnimation("HMIShowHideAnimation").Expression =
"Var.VariableGroup0.Variable1"
```

(6) GetCurrentAlarmColumnInf method

GetCurrentAlarmColumnInf

Get the content of the currently selected alarm column

Define

GetCurrentAlarmColumnInf(columnIndex)

Parameter

Name	Required/Optional	Data Type	Description
columnIndex	Required	**Int**	Column index (value range: 0~14)

Example

Get the content of the currently selected alarm column

VBScript Example
Text0.Text = AlarmWindow0.GetCurrentAlarmColumnInf(5)

(7) GetHistoryAlarmItemDataTable method

GetHistoryAlarmItemDataTable

Get the data table of the historical data

Define

GetHistoryAlarmItemDataTable(beginTime , endTime)

Parameter

Name	Required/Optional	Data Type	Description
beginTime	Required	**DateTime**	The begin time
endTime	Required	**DateTime**	The end time

Example

Get the data table of the historical data

VBScript Example
dt = AlarmWindow0.GetHistoryAlarmItemDataTable(DateTimePicker0.ValueTime,DateTimePicker1.ValueTime) Call ReportCmd.DirectExportDataToExcel(dt,"D:\Test_Alarm.xlsx",1)

(8) GetRowNum method

GetRowNum

Get the current number of real-time alarm Windows

Define

GetRowNum(variablePath)

Parameter

Name	Required/Optional	Data Type	Description
variablePath	Required	**Int**	Variable path,Assign the returned value to this variable

Example

Get the current number of real-time alarm Windows, and pass the returned value to the parameter "var.variable 0"

VBScript Example
Call AlarmWindow0.GetRowNum("Var.Variable0")

(9) Print method

Print

Print

Define

Print()

Example

Print

VBScript Example
Call AlarmWindow0.Print()

(10) QueryHistoryAlarm method

QueryHistoryAlarm

The inquiry of historical data

Define

QueryHistoryAlarm(startTime , endTime)

Parameter

Name	Required/Optional	Data Type	Description
startTime	Required	**DateTime**	The query of the start time

endTime	Required	**DateTime**	The query of the end time

Example

The inquiry of historical alarm

VBScript Example
Call AlarmWindow0.QueryHistoryAlarm(DateTimePicker0.ValueTime,DateTimePicker1.ValueTime)

(11) Save method

Save

Save

Define

Save()

Example

Save

VBScript Example
Call AlarmWindow0.Save()

(12) SetCurrentSelectRowIndex method

SetCurrentSelectRowIndex

Set the currently selected item

Define

SetCurrentSelectRowIndex(selectIndex)

Parameter

Name	Required/Optional	Data Type	Description
selectIndex	Required	**Int**	Select row index

Example

Select the third row and get the contents of column 5 of that row

VBScript Example

```
Call AlarmWindow0.SetCurrentSelectRowIndex(3)
Text0.Text = AlarmWindow0.GetCurrentAlarmColumnInf(5)
```

(13) SetExportValue method

SetExportValue

Set the parameters of the alarm data export script

Define

SetExportValue(isReal , isExportAll)

Parameter

Name	Required/Optional	Data Type	Description
isReal	Required	**Bool**	Whether it is a real - time alarm
isExportAll	Required	**Bool**	Whether to export all report

Example

Export real-time alarm

VBScript Example

```
Call AlarmWindow0.SetExportValue(True,True)
Call AlarmWindow0.DataExport()
```

(14) StartRealTimeAlarm method

StartRealTimeAlarm

Start loading the real-time alarm data

Define

StartRealTimeAlarm()

Example

Start loading the real-time alarm data

VBScript Example

```
Call AlarmWindow0.StartRealTimeAlarm()
```

(15) StopRealTimeAlarm method

StopRealTimeAlarm

Stop loading the real-time alarm data

Define

StopRealTimeAlarm()

Example

Stop loading the real-time alarm data

VBScript Example

```
Call AlarmWindow0.StopRealTimeAlarm()
```

【property】

(1) Height property

Height

Height

Define

Double Height

Example

Set the height of the specified object to 100

VBScript Example

```
AlarmWindow0.Height = 100
```

(2) IsShow property

IsShow

Whether display the specified object or not

Define

Boolean IsShow

Example

True ：The specified object is displayed on the screen False ：The specified object hidden on the screen

VBScript Example
AlarmWindow0.IsShow = True AlarmWindow0.IsShow = False

(3) IsShowHistoryTabItem property

IsShowHistoryTabItem

Whether to show the header of the history alarm

Define

Boolean IsShowHistoryTabItem

Example

Set to show the header of the history alarm

VBScript Example
AlarmWindow0.IsShowHistoryTabItem = True

(4) IsShowRelTableItem property

IsShowRelTableItem

Whether to show the header of the real-time alarm

Define

Boolean IsShowRelTableItem

Example

Set to show the header of the real-time alarm

VBScript Example
AlarmWindow0.IsShowRelTableItem = True

(5) Left property

Left

The left coordinate

Define

Double Left

Example

Object's left side of the coordinate values is 100 on the screen

VBScript Example

```
AlarmWindow0.Left = 100
```

(6) Name property

Name

Name

Define

String Name

Example

Get the default name of the specified object

VBScript Example

```
TextBox0.Text = AlarmWindow0.Name
```

(7) ToolTip property

ToolTip

Tooltip text

Define

String ToolTip

Example

Set the ToolTip of the specified object to "test"

VBScript Example

```
AlarmWindow0.ToolTip = "test"
```

(8) Top property

Top

The top coordinate

Define

Double Top

Example

Object's the top of the coordinate value is 100 in the picture

VBScript Example
AlarmWindow0.Top = 100

(9) Width property

Width

Width

Define

Double Width

Example

Set the width of the specified object to 100

VBScript Example
AlarmWindow0.Width = 100

(10) ZIndex property

ZIndex

Layer index

Define

Int32 ZIndex

Example

Set the Zindex of the specified object to 3

VBScript Example
AlarmWindow0.ZIndex = 3

11. HMIReport object

HMIReport

Report control

Methods list

	Name	Description
	CellBindingToVariable	The table cell is bound to the variable
	ChangeVariablePath	Repace template variables
	ClearCellContent	Empty the entire contents of the report cell
	ClearCellContentForSheet	Empty the entire contents of the specified worksheet
	ExportDataToXls	Export the general data(Excel file or report forms template)
	ExportDataToXlsByDialog	Export the general data(Pop-up the dialog box)
	ExportToXls	Export the data of the time interval to the template
	ExportToXlsForDataTable	Export the specified data table to the excel forms
	ExportToXlsPagesDlg	Query the save dialog box of the sectional data
	FindAnimation	Look for animation to modify the associated variables of the animation
	GetCellValue	To obtain the value of the cell
	GetCurrentRowValue	Get the statement's content of the column specified by the current line
	GetDataColumnStaticsValue	Statistical calculations to the Data Table
	GetDataTableExcludeHistoryData	To obtain the Data Table for the query of the historical data
	GetDataTableFromReport	Get the Data Table of the data source from the report
	GetProjectReportTemplates	Get the report template sequence from the current directory
	GetQueryHistoryData	Get the data of a specified period and import it to the Data Table
	GetQueryHIstoryGroupData	Get data from the table, and the name of the table is specified
	GetReportDataTable	Write the Data Table to the database
	HiddenReportColumns	Hide the columns of the query
	ImportDataToControl	Import file to report
	ImportDataToControlByDialog	Pop-up dialog to import files
	OpenDataTemplate	Open the current report template

OpenDataTemplate	Open the specified report template
Print	Print
PrintTemplate	Print the paged data
ProjectDirectory	To obtain the path of the project
QueryAlarmData	Query the alarm data
QueryDataFromExtern	External database report queries
QueryHistoryData	Query results according to the setting time interval
QueryHistoryDataByCommon	Query of the general historical data
QueryHistoryDataForSheet	Worksheet query
QuerySystemEventData	Query the system event
QueryVariableOperations	Query records of the variable operating
RegisterVariableToCell	Registry variable transfer into table cell
ReplaceCellVariableToControl	Export the real-time data for replacement
SaveReportAndOpenToCurrentTemplate	Modify the opened template and save
SetActiveGridColumnCount	Setup the number of columns of the worksheet
SetActiveGridRowCount	Setup the number of rows of the worksheet
SetActiveWorkSheet	Setup the active worksheet
SetCellBackgroundColor	Setup the background color of the table cell
SetCellFontColor	Setup the font color of the table cell
SetCellValue	Setup the content of the table cell
SetColumnNames	Setup the name of the column
SetCurrentReportTemplate	Setup the current template of the report form
SetDateFormat	Setup the date format
SetPdfSetting method	Setup Pdf export format
SetQueryDataStartPosition	Setup the starting location of the data in the report form
SetRealTimeClearOld	Whether to remove the content of previous lines
SetRealTimeVariableChange	Setup the main variables of variable alteration
SetVarRecordRulerName	Set the name of worksheet ruler
SetWholeRowValueToCells	Setup a whole row of data from the start position
SetWorkSheetEndTime	Setup the end time of query
SetWorkSheetIntervalTime	Setup the time interval of query
SetWorkSheetStartTime	Setup the start time of query
SetWorkSheetTotalIntervalTime	Set the worksheet time interval into the dictionary
ShowDataTableForReport	The Data Table is displayed in the report form controls

StartUpdateDataFromRT	The real-time data start updating
StopRealTimeVariableChange	Stop the changes of the real-time variable
StopRegisterVaraibleChange	Remove all the binding which variables convert into table cell
TriggerTimeOrder method	Set the query time ascending and descending order sorting
UnBindingEventChange	Unbound the table cell change events
UnRegisterVariableToCell	Unbundle variables convert into table cell
VisiableReportColumns	Show the columns of the query

Property list

	Name	Description
	Height	Height
	IntervalDay	Get or setup the number of days that query data interval
	IntervalHour	Get or setup the number of hours that query data interval
	IntervalMillisecond	Get or setup the number of milliseconds that query data interval
	IntervalMinute	Get or setup the number of minutes that query data interval
	IntervalMonth	Get or setup the number of months that query data interval
	IntervalQuarter	Get or setup the number of quarters that query data interval
	IntervalSecond	Get or setup the number of seconds that query data interval
	IntervalUnit	Unit of time interval
	IntervalYear	Get or setup the number of years that query data interval
	IsShow	Whether display the specified object or not
	Left	The left coordinate
	Name	Name
	ToolTip	Tooltip text
	Top	The top coordinate
	Width	Width
	ZIndex	Layer index

The following is the detailed description of the script:

【Method】

(1) CellBindingToVariable method

CellBindingToVariable

The table cell is bound to the variable

Define

CellBindingToVariable(sheetIndex , rowIndex , columnIndex , variablePath)

Parameter

Name	Required/Optional	Data Type	Description
sheetIndex	Required	**Int**	The worksheet index
rowIndex	Required	**Int**	The row num ber
columnIndex	Required	**Int**	The column number
variablePath	Required	**String**	The variable path

Example

Assign values from 1 to 10 rows in column 6 of worksheet 0 to 10 variables

VBScript Example
For i = 1 To 10 Call Report0.CellBindingToVariable(0,i,6,"Var.Variable"&i) Next

Related script :

Unbound the table cell change events

VBScript Example
Call Report0.UnBindingEventChange(0)

(2) ChangeVariablePath method

ChangeVariablePath

Repace variables of report template

Define

ChangeVariablePath(rowIndex , columnIndex , variablePath)

Parameter

Name	Required/Optional	Data Type	Description

rowIndex	Required	**Int**	The row num ber
columnIndex	Required	**Int**	The column number
variablePath	Required	**String**	The variable path

Example

Replace the variable in row 2 and column 2 of worksheet0 with VarRecord.RecordVariable

VBScript Example
Call Report0.ChangeVariablePath(2,2,"VarRecord.RecordVariable")

(3) ClearCellContent method

ClearCellContent

Empty the entire contents of the report cell （In the case of unbound report template, all contents of the report cell can be emptied; In the case of binding, it is equivalent to re-opening the report template.）

Define

ClearCellContent()

Example

Empty the entire contents of the report cell

VBScript Example
Call Report0.ClearCellContent()

(4) ClearCellContentForSheet method

ClearCellContentForSheet

Empty the entire contents of the specified worksheet

Define

ClearCellContentForSheet(sheetIndex)

Parameter

Name	Required/Optional	Data Type	Description
sheetIndex	Required	**Int**	The worksheet index

Example

Empty the contents of the worksheet0

VBScript Example
Call Report0.ClearCellContentForSheet(0)

(5) ExportDataToXls method

ExportDataToXls

Export the general data（All data can be exported except toolbar data queries and script data queries of "QueryHistoryData" and "QueryHistoryDataForSheet"）

Define

ExportDataToXls(filePath)

Parameter

Name	Required/Optional	Data Type	Description
filePath	Required	**String**	Absolute path

Example

Export the general data

VBScript Example
Call Report0.ExportDataToXls("D:\test.xls")

(6) ExportDataToXlsByDialog method

ExportDataToXlsByDialog

Export the general data（All data can be exported except toolbar data queries and script data queries of "QueryHistoryData" and "QueryHistoryDataForSheet"）

Define

ExportDataToXlsByDialog()

Example

Export the general data

VBScript Example

```
Call Report0.ExportDataToXlsByDialog()
```

(7) ExportToXls method

ExportToXls

Export the data of the time interval to the template，Use pagination to query (suggest using this method to export, export faster)

Define

ExportToXls(excelfilePath)

Parameter

Name	Required/Optional	Data Type	Description
excelfilePath	Required	**String**	Absolute path

Example

Export the data of the time interval to the template(data query must be done first)

VBScript Example

```
Call Report0.ExportToXls("D:\test.xls")
```

Can be combined with the script QueryHistoryData() :

VBScript Example

```
Call Report0.SetWorkSheetStartTime(0,DateTimePicker0.ValueTime)
Call Report0.SetWorkSheetEndTime(0,DateTimePicker1.ValueTime)
Call Report0.SetVarRecordRulerName(0,"s1")
Call Report0.QueryHistoryData()
```

(8) ExportToXlsForDataTable method

ExportToXlsForDataTable

Export the specified data table to the excel forms

Define

ExportToXlsForDataTable(dt , excelPath , excelVersion)

Parameter

Name	Required/Optional	Data Type	Description
dt	Required	**DataTable**	The data source
excelPath	Required	**String**	Export the absolute path
excelVersion	Required	**Int**	excel version（0 stands for 2003，otherwise stands for 2007）

Example

1.Export the specified data table to the excel

VBScript Example

```
dt = DbAccess.DatabaseAccess.GetTable()
Call ReportForms0.ExportToXlsForDataTable(dt,"D:\test.xls",1)
```

2.Export the DataTable from the report control to excel (This situation must be combined with scripts such as the general historical data query,system event query and operation variable record query)

VBScript Example

```
dt = ReportForms0.GetDataTableExcludeHistoryData()
Call ReportForms0.ExportToXlsForDataTable(dt,"D:\test.xls",1)
```

3. Export the contents of the report directly to Excel.

VBScript Example

```
dt = ReportForms0.GetDataTableFromReport(1)
Call ReportForms0.ExportToXlsForDataTable(dt,"D:\test.xls",1)
```

(9) ExportToXlsPagesDlg method

ExportToXlsPagesDlg

Query the save dialog of the segmented data

Define

ExportToXlsPagesDlg()

Example

Query the save dialog of the segmented data

VBScript Example

```
Call Report0.ExportToXlsPagesDlg()
```

Can be combined with the script QueryHistoryData() :

VBScript Example

```
Call Report0.SetWorkSheetStartTime(0,DateTimePicker0.ValueTime)
Call Report0.SetWorkSheetEndTime(0,DateTimePicker1.ValueTime)
Call Report0.SetVarRecordRulerName(0,"s1")
Call Report0.QueryHistoryData()
```

(10) FindAnimation method

FindAnimation

Look for animation to modify the associated variables of the animation

Define

FindAnimation(name)

Parameter

Name	Required/Optional	Data type	Description
name	Required	**String**	The name of animation

Example

Change the associated variables of the visibility animation of the object into Var.VariableGroup0.Variable1

VBScript Example

```
Report0.FindAnimation("HMIShowHideAnimation").Expression = "Var.VariableGroup0.Variable1"
```

(11) GetCellValue method

GetCellValue

To obtain the value of the table cell

Define

GetCellValue(rowIndex , columnIndex)

Parameter

Name	Required/Optional	Data Type	Description
rowIndex	Required	**Int**	The number of rows，starting from 1
columnIndex	Required	**Int**	The number of columns，starting from 1

Example

To obtain the value of the cell（1,1）

VBScript Example
Button0.Content = Report0.GetCellValue(1,1)

(12) GetCurrentRowValue method

GetCurrentRowValue

Get the content of the report forms specified by the current line

Define

GetCurrentRowValue(columnIndex)

Parameter

Name	Required/Optional	Data Type	Description
columnIndex	Required	**Int**	The number of columns，starting from

Example

Get the content of the report forms in the first column of current line

VBScript Example
Button0.Content = Report0.GetCurrentRowValue(1)

(13) GetDataColumnStaticsValue method

GetDataColumnStaticsValue

Statistical calculations to the Data Table

Define

GetDataColumnStaticsValue(dt , columnName , functionName)

Parameter

Name	Required/Optional	Data Type	Description
dt	Required	**DataTable**	The data source
columnName	Required	**String**	Column names collection,seperated by commas
functionName	Required	**String**	Function names collection(avg,min,max,sum),seperated by commas

Example

1. Calculate the average value of "RecordVariable" (Timing record variable:"VarRecord.RecordVariable")

VBScript Example

```
Call Report0.SetWorkSheetStartTime(0,DateTimePicker0.ValueTime)
Call Report0.SetWorkSheetEndTime(0,DateTimePicker1.ValueTime)
Call Report0.SetVarRecordRulerName(0,"s1")       'Set the query rules for historical data,and interval time is 1s
dt = Report0.GetQueryHistoryData(0,"VarRecord.RecordVariable","1")     'Get data within the specified time period,and the parameter "1" means "Value"
ff = Report0.GetDataColumnStaticsValue(dt,"VarRecord.RecordVariable","avg")
Call Report0.SetCellValue(ff,1,5)     'Displays the calculated results in the first row and fifth column of the Report control
```

2.Calculate the average value of "RecordVariable" (Group record variable:"HistRecord.aaa.RecordVariable"), and "aaa" is the history variable group

VBScript Example

```
Call Report0.SetWorkSheetStartTime(0,DateTimePicker0.ValueTime)
Call Report0.SetWorkSheetEndTime(0,DateTimePicker1.ValueTime)
Call Report0.SetWorkSheetIntervalTime(0,1000)     'The time interval for the history group is 1000 ms
dt = Report0.GetQueryHistoryData(0,"HistRecord.aaa.RecordVariable","1")
ff = Report0.GetDataColumnStaticsValue(dt,"HistRecord.aaa.RecordVariable","avg")
Call Report0.SetCellValue(ff,2,5)
```

or

VBScript Example

```
Call Report0.SetWorkSheetStartTime(0,DateTimePicker0.ValueTime)
Call Report0.SetWorkSheetEndTime(0,DateTimePicker1.ValueTime)
Call Report0.SetWorkSheetIntervalTime(0,1000)     'The time interval for the history group is 1000 ms
dt = Report0.GetQueryHistoryData(0,"HistRecord.aaa.RecordVariable","1")
ff = Report0.GetDataColumnStaticsValue(dt,"1","avg") 'The "1" represents the index of the historical variable and begins with "1".
Call Report0.SetCellValue(ff,2,5)
```

3.Calculate the average value of column "age" in "DatabaseAccess"

VBScript Example

```
dt = DbAccess.DatabaseAccess.GetTable()
ff = Report0.GetDataColumnStaticsValue(dt,"age","avg")
Call Report0.SetCellValue(ff,3,5)
```

(14) GetDataTableExcludeHistoryData method

GetDataTableExcludeHistoryData

To obtain the Data Table for the query of the historical data(This script must be combined with those scripts such as the general historical data query,system event query and operation variable record query)

Define

GetDataTableExcludeHistoryData()

Example

To obtain the Data Table for the query of the historical data

VBScript Example

```
dt = Report0.GetDataTableExcludeHistoryData()
Call Report0.ExportToXlsForDataTable(dt,"D:\test.xls",1)   'the parameter "1" means "excel version"("0" is 2003,otherwise 2007)
```

(15) GetDataTableFromReport method

GetDataTableFromReport

Get the Data Table of the data source from the report (The contents of the report display)

Define

GetDataTableFromReport(model)

Parameter

Name	Required/Optional	Data Type	Description
model	Required	**Int**	0 means have no column name,otherwise have a column name

Example

Get the Data Table of the data source from the report

VBScript Example

```
dt = Report0.GetDataTableFromReport(1)
Call Report0.ExportToXlsForDataTable(dt,"D:\test.xls",1)
```

(16) GetProjectReportTemplates method

GetProjectReportTemplates

Get the report template sequence from the current directory

Define

GetProjectReportTemplates()

Example

Get the report template sequence from the current directory

VBScript Example

```
Text0.Text = Report0.GetProjectReportTemplates()
```

(17) GetQueryHistoryData method

GetQueryHistoryData

Get the data of a specified period and import it to the Data Table

Define

GetQueryHistoryData(sheetIndex , conditions , types)

Parameter

Name	Required/Optional	Data Type	Description
sheetIndex	Required	**Int**	The worksheep index
conditions	Required	**String**	Conditions is the variable collection, separated by commas
types	Required	**String**	Types are divided into TriggeringTime、Value, which can use 0 and 1 instead

Example

1. Get the "Variable record (Timing record)" data to the DataTable within a specified time, and display in the Report control

VBScript Example

```
Call Report0.SetWorkSheetStartTime(0,DateTimePicker0.ValueTime)
Call Report0.SetWorkSheetEndTime(0,DateTimePicker1.ValueTime)
Call Report0.SetVarRecordRulerName(0,"s1")
dt = Report0.GetQueryHistoryData(0,"VarRecord.RecordVariable,VarRecord.RecordVariable1","1,1")
Call Report0.ShowDataTableForReport(1,1,dt)
```

2.Get the "Variable group record" data in the specified time to the DataTable, and display in the Report control

VBScript Example

```
Call Report0.SetWorkSheetStartTime(0,DateTimePicker0.ValueTime)
Call Report0.SetWorkSheetEndTime(0,DateTimePicker1.ValueTime)
Call Report0.SetWorkSheetIntervalTime(0,1000)
dt = Report0.GetQueryHistoryData(0,"HistRecord.aaa.RecordVariable","1")
Call Report0.ShowDataTableForReport(1,1,dt)
```

(18) GetQueryHistoryGroupData method

GetQueryHistoryGroupData

Get data from the table, and the name of the table is specified

Define

GetQueryHistoryGroupData(sheetIndex , tableName , columnNames , rowCount)

Parameter

Name	Required/Optional	Data Type	Description
sheetIndex	Required	**Int**	The worksheep index
tableName	Required	**String**	The name of the historical variables group
columnNames	Required	**String**	Historical variables collection，separated by commas
rowCount	Required	**Int**	The number of rows

Example

Get data from the table, and the name of the table is specified

VBScript Example
Call Report0.SetWorkSheetStartTime(0,DateTimePicker0.ValueTime) Call Report0.SetWorkSheetEndTime(0,DateTimePicker1.ValueTime) Call Report0.SetWorkSheetIntervalTime(0,1000) dt = Report0.GetQueryHistoryGroupData(0,"aaa","HistRecord.aaa.RecordVariable,HistRecord.aaa.RecordVariable1",50) Call Report0.ShowDataTableForReport(1,1,dt)

(19) GetReportDataTable method

GetReportDataTable

Write the Data Table to the database

Define

GetReportDataTable(mainKey)

Parameter

Name	Required/Optional	Data Type	Description
mainKey	Required	**Int**	Specify which column as the primary key

Example

The report shows the data in the database access table, manually modify the data in the table, and then use this script to save to the database

VBScript Example

```
dt = Report0.GetReportDataTable(0)
cc = DbAccess.DatabaseAccess.SaveTable(dt)
```

Related script:Report displays the data in the database access table

VBScript Example

```
dt = DbAccess.DatabaseAccess.GetTable()
Call Report0.ShowDataTableForReport(1,1,dt)
```

(20) HiddenReportColumns method

HiddenReportColumns

Hide the columns of the query

Define

HiddenReportColumns(columns)

Parameter

Name	Required/Optional	Data Type	Description
columns	Required	**String**	The number of columns

Example

Hide the columns of the query

VBScript Example

```
Call Report0.HiddenReportColumns("Alarm")
```

Related script:Display the columns of the query

VBScript Example

```
Call Report0.VisiableReportColumns("Alarm")
```

(21) ImportDataToControl method

ImportDataToControl

Import file to report

Define

ImportDataToControl(filePath)

Parameter

Name	Required/Optional	Data Type	Description
filePath	Required	**String**	The path of file

Example

Import file to report

VBScript Example
Call Report0.ImportDataToControl("D:\test.xls")

(22) ImportDataToControlByDialog method

ImportDataToControlByDialog

Pop-up dialog to import files

Define

ImportDataToControlByDialog()

Example

Pop-up dialog to import files

VBScript Example
Call Report0.ImportDataToControlByDialog()

(23) OpenDataTemplate method

OpenDataTemplate

Open the current report template

Define

OpenDataTemplate()

Example

Open the current template

VBScript Example

```
Call Report0.OpenDataTemplate()
```

(24) OpenDataTemplate method

OpenDataTemplate

Open the template specified worksheet

Define

OpenDataTemplate(sheetIndex，templatePath)

Parameter

Name	Required/Optional	Data Type	Description
sheetIndex	Required	**String**	The worksheep index
templatePath	Required	**String**	The name of the report template

Example

Open the template specified worksheet

VBScript Example

```
Call Report0.OpenDataTemplate(0,"Report0")
```

(25) Print method

Print

Print

Define

Print()

Example

Print

VBScript Example

```
Call Report0.Print()
```

(26) PrintTemplate method

PrintTemplate

Print the paged data

Define

PrintTemplate()

Example

Print the paged data

VBScript Example
Call Report0.PrintTemplate()

Related scripts:

VBScript Example
Call Report0.SetWorkSheetStartTime(0,DateTimePicker0.ValueTime) Call Report0.SetWorkSheetEndTime(0,DateTimePicker1.ValueTime) Call Report0.SetVarRecordRulerName(0,"s1") Call Report0.QueryHistoryData()

(27) ProjectDirectory method

ProjectDirectory

To obtain the path of the project

Define

ProjectDirectory()

Example

To obtain the path of the project

VBScript Example
Text0.Text = Report0.ProjectDirectory()

(28) QueryAlarmData method

QueryAlarmData

Query the alarm data

Define

QueryAlarmData(sheetIndex)

Parameter

Name	Required/Optional	Data Type	Description
sheetIndex	Required	**Int**	The worksheep index，starting from 0

Example

Query the alarm data

VBScript Example
Call Report0.SetWorkSheetStartTime(0,DateTimePicker0.ValueTime) Call Report0.SetWorkSheetEndTime(0,DateTimePicker1.ValueTime) Call Report0.QueryAlarmData(0)

(29) QueryDataFromExtern method

QueryDataFromExtern

Query the external database report forms

Define

QueryDataFromExtern(sheetIndex ，dt)

Parameter

Name	Required/Optional	Data Type	Description
sheetIndex	Required	**Int**	The worksheep index，starting from 0
dt	Required	**DataTable**	The data source

Example

Query the external database report forms (Operation method:1. Create a report template for the external database 2. Bind to the report control 3. Execute the query script)

VBScript Example
dt = DbAccess.DatabaseAccess.GetTable() Call Report0.QueryDataFromExtern(0,dt)

(30) QueryHistoryData method

QueryHistoryData

Query results according to the setting time interval

Define

QueryHistoryData()

Example

1. According to the rule "s1" query timing "variable record" (the report control must be associated with the historical report template)

VBScript Example

```
Call Report0.SetWorkSheetStartTime(0,DateTimePicker0.ValueTime)
Call Report0.SetWorkSheetEndTime(0,DateTimePicker1.ValueTime)
Call Report0.SetVarRecordRulerName(0,"s1")
Call Report0.QueryHistoryData()
```

2. Query "variable group record" according to the interval time of 1000ms (the report control must be associated with historical group report template)

VBScript Example

```
Call Report0.SetWorkSheetStartTime(0,DateTimePicker0.ValueTime)
Call Report0.SetWorkSheetEndTime(0,DateTimePicker1.ValueTime)
Call Report0.SetWorkSheetIntervalTime(0,1000)
Call Report0.QueryHistoryData()
```

(31) QueryHistoryDataByCommon method

QueryHistoryDataByCommon

Query of the general historical data

Define

QueryHistoryDataByCommon(sheetIndex , conditions , types)

Parameter

Name	Required/Optional	Data Type	Description

sheetIndex	Required	**Int**	The worksheep index
conditions	Required	**String**	The condition is the variable set, separated by commas
types	Required	**String**	Types are divided into TriggeringTime and Value,which can be replace by 0 and 1

Example

1.Query variable records (Timing records)

VBScript Example

```
Call Report0.SetWorkSheetStartTime(0,DateTimePicker0.ValueTime)
Call Report0.SetWorkSheetEndTime(0,DateTimePicker1.ValueTime)
Call Report0.SetVarRecordRulerName(0,"s1")
Call
Report0.QueryHistoryDataByCommon(0,"VarRecord.RecordVariable,VarRecord.RecordVariable,VarRecord.RecordVariable1","0,1,1")
```

2.Query variable group records, and "aaa" is the history group

VBScript Example

```
Call Report0.SetWorkSheetStartTime(0,DateTimePicker0.ValueTime)
Call Report0.SetWorkSheetEndTime(0,DateTimePicker1.ValueTime)
Call Report0.SetWorkSheetIntervalTime(0,1000)
Call
Report0.QueryHistoryDataByCommon(0,"HistRecord.aaa.RecordVariable,HistRecord.aaa.RecordVariable,HistRecord.aaa.RecordVariable1","0,1,1")
```

(32) QueryHistoryDataForSheet method

QueryHistoryDataForSheet

Query the specified worksheet, and need to associate the corresponding report template

Define

QueryHistoryDataForSheet()

Example

1.Query the "variable record" of the timing (the report control must be associated with the historical report template)

VBScript Example

```
Call Report0.SetWorkSheetStartTime(0,DateTimePicker0.ValueTime)
Call Report0.SetWorkSheetEndTime(0, DateTimePicker1.ValueTime)
Call Report0.SetVarRecordRulerName(0,"s1")
Call Report0.QueryHistoryDataForSheet()
```

2.Query "variable group records" (report control must be associated with historical group report template)

VBScript Example

```
Call Report0.SetWorkSheetStartTime(0,DateTimePicker0.ValueTime)
Call Report0.SetWorkSheetEndTime(0, DateTimePicker1.ValueTime)
Call Report0.SetWorkSheetIntervalTime(0,1000)
Call Report0.QueryHistoryDataForSheet()
```

(33) QuerySystemEventData method

QuerySystemEventData

Query system event

Define

QuerySystemEventData(sheetIndex)

Parameter

Name	Required/Optional	Data Type	Description
sheetIndex	Required	**Int**	The worksheet index，start from zero

Example

Query system event

VBScript Example

```
Call Report0.SetWorkSheetStartTime(0,DateTimePicker0.ValueTime)
Call Report0.SetWorkSheetEndTime(0,DateTimePicker1.ValueTime)
Call Report0.QuerySystemEventData(0)
```

(34) QueryVariableOperations method

QueryVariableOperations

Query records of the variable operating

Define

QueryVariableOperations(sheetIndex , opVariablePaths)

Parameter

Name	Required/Optional	Data Type	Description
sheetIndex	Required	**Int**	The worksheet index,start from zero
opVariablePaths	Required	**String**	The path collection of the operation variable （null means all，"Operation"is the root node，"Operation.OperatingVariablesGroup0" is the query of group

Example

1.Query the variable operating records (Query the root node)

VBScript Example

```
Call Report0.SetWorkSheetStartTime(0,DateTimePicker0.ValueTime)
Call Report0.SetWorkSheetEndTime(0,DateTimePicker1.ValueTime)
Call Report0.QueryVariableOperations(0,"Operation")
```

2.Query the variable operating records (Query all data)

VBScript Example

```
Call Report0.SetWorkSheetStartTime(0,DateTimePicker0.ValueTime)
Call Report0.SetWorkSheetEndTime(0,DateTimePicker1.ValueTime)
Call Report0.QueryVariableOperations(0,"")
```

3.Query the variable operating records by group (Query "Operation.OperationVariableGroup0")

VBScript Example

```
Call Report0.SetWorkSheetStartTime(0,DateTimePicker0.ValueTime)
Call Report0.SetWorkSheetEndTime(0,DateTimePicker1.ValueTime)
Call Report0.QueryVariableOperations(0,"Operation.OperationVariableGroup0")
```

(35) RegisterVariableToCell method

RegisterVariableToCell

Registry variable transfer into table cell

Define

RegisterVariableToCell(variablePath ,rowIndex , columnIndex)

Parameter

Name	Required/Optional	Data Type	Description
variablePath	Required	**String**	The variable path
rowIndex	Required	**Int**	The number of the rows, starting from 1
columnIndex	Required	**Int**	The number of the columns, starting from 1

Example

1. Register "Var. Variable0" to cell (1,1)

VBScript Example

```
Call Report0.SetActiveWorkSheet(1) 'When there are multiple worksheets at the same time, the current active worksheet can be set
Call Report0.RegisterVariableToCell("Var.Variable0",1,1)
```

2. Register 10 variables separately to cells

VBScript Example

```
Call Report0.SetActiveWorkSheet(1)
For i = 1 To 10
Call Report0.RegisterVariableToCell("Var.Variable"&i,3,i)
Next
```

Related scripts:

a. All unbind of variable to cell

VBScript Example

```
Call Report0.StopRegisterVaraibleChange()
```

b. Unbind variables to cell

VBScript Example

```
Call Report0.UnRegisterVariableToCell("Var.Variable0",1,1)
```

(36) ReplaceCellVariableToControl method

ReplaceCellVariableToControl

Replace real-time report data (refresh real-time data, perform a refresh once)

Define

ReplaceCellVariableToControl()

Example

Replace real-time report data

VBScript Example

```
Call Report0.ReplaceCellVariableToControl()
```

(37) SaveReportAndOpenToCurrentTemplate method

SaveReportAndOpenToCurrentTemplate

Modify the opened template and save

Define

SaveReportAndOpenToCurrentTemplate()

Example

Modify the opened template and save

VBScript Example

```
Call Report0.SaveReportAndOpenToCurrentTemplate()
```

(38) SetActiveGridColumnCount method

SetActiveGridColumnCount

Setup the number of columns of the worksheet

Define

SetActiveGridColumnCount(columnCount)

Parameter

Name	Required/Optional	Data Type	Description
columnCount	Required	**Int**	The number of columns

Example

Setup the number of columns of the worksheet to 30

VBScript Example
Call Report0.SetActiveGridColumnCount(30)

(39) SetActiveGridRowCount method

SetActiveGridRowCount

Setup the number of rows of the worksheet

Define

SetActiveGridRowCount(rowCount)

Parameter

Name	Required/Optional	Data Type	Description
rowCount	Required	**Int**	The number of rows

Example

Setup the number of rows of the worksheet to 100

VBScript Example
Call Report0.SetActiveGridRowCount(100)

(40) SetActiveWorkSheet method

SetActiveWorkSheet

Setup the active worksheet

Define

SetActiveWorkSheet(sheetIndex)

Parameter

Name	Required/Optional	Data Type	Description
sheetIndex	Required	**Int**	The worksheet index,start from 0

Example

Setup the active worksheet

VBScript Example
Call Report0.SetActiveWorkSheet(1) Call Report0.SetCellBackgroundColor(1,1,Colors.Red)

(41) SetCellBackgroundColor method

SetCellBackgroundColor

Setup the background color of the table cell

Define

SetCellBackgroundColor(rowIndex , columnIndex , backgroundColor)

Parameter

Name	Required/Optional	Data Type	Description
rowIndex	Required	**Int**	The number of rows,start from 1
columnIndex	Required	**Int**	The number of columns,start from 1
backgroundColor	Required	**Brush**	The background color

Example

Setup the background color of the table cell to red（The colors method of the system）

VBScript Example
Call Report0.SetCellBackgroundColor(1,1,Colors.Red)

(42) SetCellFontColor method

SetCellFontColor

Set the font color of the table cell

Define

SetCellFontColor(rowIndex , columnIndex , foreColor)

Parameter

Name	Required/Optional	Data Type	Description
rowIndex	Required	**Int**	The number of rows,start from 1
columnIndex	Required	**Int**	The number of columns,start from 1
foreColor	Required	**Brush**	The font color

Example

Set the font color of the table cell to red(The colors methor of the system)

VBScript Example
Call Report0.SetCellFontColor(1,1,Colors.Red)

(43) SetCellValue method

SetCellValue

Call the script to set the value of table cells

Define

SetCellValue(value , rowIndex , columnIndex)

Parameter

Name	Required/Optional	Data Type	Description
value	Required	**Object**	The value to set to the table cells
rowIndex	Required	**Int**	The number of rows,start from 1
columnIndex	Required	**Int**	The number of columns,start from 1

Example

Set the value of the table cell to 20 or“test”or“True”

VBScript Example
Call Report0.SetActiveWorkSheet(1) 'when multiple worksheets exist, the current active worksheet can be set

```
Call Report0.SetCellValue(20,1,1)
Call Report0.SetCellValue("test",2,2)
Call Report0.SetCellValue(True,3,3)
```

(44) SetColumnNames method

SetColumnNames

Set the name of the column in the script access table

Define

SetColumnNames(startRow , startColumn , columnNames)

Parameter

Name	Required/Optional	Data Type	Description
startRow	Required	**Int**	Set the beginning row of the database table in the report，the index starting from 1
startColumn	Required	**Int**	Set the beginning column of the database table in the report，the index starting from 1
columnNames	Required	**String**	The collection of the column name，separated by commas

Example

Set the column name

VBScript Example

```
Call Report0.SetColumnNames(1,1,"SetColumnName")
```

(45) SetCurrentReportTemplate method

SetCurrentReportTemplate

Set the current template of the report form

Define

SetCurrentReportTemplate(templateName)

Parameter

Name	Required/Optional	Data Type	Description
templateName	Required	**String**	The current template of the report form

Example

Set the current template of the report form

VBScript Example
Call Report0.SetCurrentReportTemplate("HistoryReport") Call Report0.OpenDataTemplate()

(46) SetDateFormat method

SetDateFormat

Set the date format

Define

SetDateFormat(format)

Parameter

Name	Required/Optional	Data Type	Description
format	Required	**Int**	The date format（0 on behalf of yyyy/MM/dd hh:mm:ss，1 on behalf of yyyy-MM-dd hh:mm:ss，2 on behalf of yyyy/MM/dd，3 on behalf of hh:mm:ss，4 on behalf of yyyy,five digit of Taiwan format years）

Example

Set the date format

VBScript Example
Call Report0.SetDateFormat(0)

(47) SetPdfSetting method

SetPdfSetting

Setup Pdf export format

Define

SetPdfSetting(pdfColumnCount , pdfMargin , pageSize)

Parameter

Name	Required/Optional	Data Type	Description
pdfColumnCount	Required	**Int**	The number of columns
pdfMargin	Required	**Float**	Page margins
pageSize	Required	**String**	Paper size, such as A4, A5, etc

Example

Pdf export format setting

VBScript Example
Call Report0.SetPdfSetting(5,10,"A4")

(48) SetQueryDataStartPosition method

SetQueryDataStartPosition

Set the starting location of the data in the report form

Define

SetQueryDataStartPosition(startRow , startColumn , _isShowColumnHeader)

Parameter

Name	Required/Optional	Data Type	Description
startRow	Required	**Int**	Set the starting row of the data in the report form，the index starting from 1
startColumn	Required	**Int**	Set the starting column of the data in the report form，the index starting from 1
_isShowColumnHeader	Required	**Int**	Whether to display the column header（0 means do not to display，greater than 0 means to display）

Example

Set the starting location of the data in the report

VBScript Example

```
Call Report0.SetWorkSheetStartTime(0,DateTimePicker0.ValueTime)
Call Report0.SetWorkSheetEndTime(0,DateTimePicker1.ValueTime)
Call Report0.SetVarRecordRulerName(0,"s1")
Call Report0.SetQueryDataStartPosition(3,1,1)  'Start at row 3, column 1, and display the table heade
Call
Report0.QueryHistoryDataByCommon(0,"VarRecord.RecordVariable,VarRecord.RecordVariable,VarR
ecord.RecordVariable1","0,1,1")
```

(49) SetRealTimeClearOld method

SetRealTimeClearOld

Whether to remove the content of previous lines

Define

SetRealTimeClearOld(_isClearOld)

Parameter

Name	Required/Optional	Data Type	Description
_isClearOld	Required	**Int**	Whether to remove(0 means do not to remove,1 means to remove)

Example

Do not clear the contents of previous rows for real-time data refresh (need to associate the real-time report template)

VBScript Example

```
Call Report0.SetRealTimeClearOld(0)
Call Report0.StartUpdateDataFromRT(1)
Call Report0.SetRealTimeVariableChange("Var.Variable",10) 'The number of refresh rows is 10
```

(50) SetRealTimeVariableChange method

SetRealTimeVariableChange

Set the main variables of variable alteration

Define

SetRealTimeVariableChange(_valuePath , _totalRowCount)

Parameter

Name	Required/Optional	Data Type	Description
_valuePath	Required	**String**	The variable path
_totalRowCount	Required	**Int**	Set the total number of rows

Example

Set the main variables of variable alteration

VBScript Example
Call Report0.StartUpdateDataFromRT(1) Call Report0.SetRealTimeVariableChange("Var.Variable0",100)

(51) SetVarRecordRulerName method

SetVarRecordRulerName

Set the name of the query rule

Define

SetVarRecordRulerName(RuleName)

Parameter

Name	Required/Optional	Data Type	Description
RuleName	Required	**String**	The name of the timer for a timed record

Example

Set the name of the query rule,“s1” is the timer name for the timing record

VBScript Example
Call Report0.SetVarRecordRulerName("s1")

(52) SetWholeRowValueToCells method

SetWholeRowValueToCells

Set a whole row of data from the start position

Define

SetWholeRowValueToCells(*obj* , startRow , *startColumn*)

Parameter

Name	Required/Optional	Data Type	Description
obj	Required	**String**	The data of rows
startRow	Required	**Int**	The number of rows，the index starting from 1
startColumn	Required	**Int**	The number of column，the index starting from 1

Example

Set a whole row of data from the start position

VBScript Example
Call Report0.SetWholeRowValueToCells("30,test,5,true",1,1)

(53) SetWorkSheetEndTime method

SetWorkSheetEndTime

Set the end time of query

Define

SetWorkSheetEndTime(worksheetIndex , _endTime)

Parameter

Name	Required/Optional	Data Type	Description
worksheetIndex	Required	**Int**	The worksheet index，starting from 0
_endTime	Required	**DateTime**	The end time

Example

Query the data recorded by the history group (need to associate report templates)

VBScript Example
Call Report0.SetWorkSheetStartTime(0,DateTimePicker0.ValueTime) Call Report0.SetWorkSheetEndTime(0,DateTimePicker1.ValueTime) Call Report0.SetWorkSheetIntervalTime(0,1000)

```
Call Report0.QueryHistoryData()
```

(54) SetWorkSheetIntervalTime method

SetWorkSheetIntervalTime

Set the end time of query

Define

SetWorkSheetIntervalTime(worksheetIndex , intervalTime)

Parameter

Name	Required/Optional	Data Type	Description
worksheetIndex	Required	**Int**	The worksheet index，starting from 0
intervalTime	Required	**Int64**	The time interval，the unit is ms

Example

Query the data recorded by the history group (need to associate report templates)

VBScript Example

```
Call Report0.SetWorkSheetStartTime(0,DateTimePicker0.ValueTime)
Call Report0.SetWorkSheetEndTime(0,DateTimePicker1.ValueTime)
Call Report0.SetWorkSheetIntervalTime(0,1000)
Call Report0.QueryHistoryData()
```

(55) SetWorkSheetStartTime method

SetWorkSheetStartTime

Set the start time of query

Define

SetWorkSheetStartTime(worksheetIndex , _startTime)

Parameter

Name	Required/Optional	Data Type	Description
worksheetIndex	Required	**Int**	The worksheet index,starting from 0
_startTime	Required	**DateTime**	The start time

Example

Query the data recorded by the history group (need to associate report templates)

VBScript Example

```
Call Report0.SetWorkSheetStartTime(0,DateTimePicker0.ValueTime)
Call Report0.SetWorkSheetEndTime(0,DateTimePicker1.ValueTime)
Call Report0.SetWorkSheetIntervalTime(0,1000)
Call Report0.QueryHistoryData()
```

(56) SetWorkSheetTotalIntervalTime method

SetWorkSheetTotalIntervalTime

Set the worksheet time interval into the dictionary

Define

SetWorkSheetTotalIntervalTime(worksheetIndex)

Parameter

Name	Required/Optional	Data Type	Description
worksheetIndex	Required	**Int**	The worksheet index,starting from 0

Example

Set the interval time of the worksheet to the dictionary, and when the interval is less than 0, query according to the original data in the database

VBScript Example

```
Call Report0.SetWorkSheetStartTime(0,DateTimePicker0.ValueTime)
Call Report0.SetWorkSheetEndTime(0,DateTimePicker1.ValueTime)
Report0.IntervalDay = 0
Report0.IntervalHour = 0
Report0.IntervalMinute = 0
Report0.IntervalSecond = 1
Report0.IntervalMillisecond = 0
Call Report0.SetWorkSheetTotalIntervalTime(0)
Call Report0.QueryHistoryData()
```

(57) ShowDataTableForReport method

ShowDataTableForReport

The Data Table is displayed in the report form controls

Define

ShowDataTableForReport(fromRowIndex , fromColumnIndex , dt)

Parameter

Name	Required/Optional	Data Type	Description
fromRowIndex	Required	**Int**	Start number of rows,the index starting from 1
fromColumnIndex	Required	**Int**	Start number of column,the index starting from 1
dt	Required	**DataTable**	The data source

Example

The Data Table is displayed in the report form controls

VBScript Example
dt = DbAccess.DatabaseAccess.GetTable() Call Report0.ShowDataTableForReport(1,1,dt)

(58) StartUpdateDataFromRT method

StartUpdateDataFromRT

The real-time data start updating

Define

StartUpdateDataFromRT(_model)

Parameter

Name	Required/Optional	Data Type	Description
_model	Required	**Int**	The value is 1 means variable changes increasing the rows,otherwise changes in the specified table cell

Example

1.Real-time data starts to update, and data changes in the same cell

VBScript Example

```
Call Report0.StartUpdateDataFromRT(0)
```

2.Real-time data starts to update, data is displayed in increasing number of rows

VBScript Example

```
Call Report0.SetRealTimeClearOld(0)     'Whether to clear existing data, 0 is not cleared, 1 is cleared
Call Report0.StartUpdateDataFromRT(1)
Call Report0.SetRealTimeVariableChange("Var.Variable",10) 'Refresh rows 10
```

Related scripts:

Stop real-time variable changes

VBScript Example

```
Call Report0.StopRealTimeVariableChange()
```

(59) StopRealTimeVariableChange method

StopRealTimeVariableChange

Stop the changes of the real-time variable

Define

StopRealTimeVariableChange()

Example

Stop the changes of the real-time variable

VBScript Example

```
Call Report0.StopRealTimeVariableChange()
```

(60) StopRegisterVaraibleChange method

StopRegisterVaraibleChange

Remove all the binding which variables convert into table cell

Define

StopRegisterVaraibleChange()

Example

Remove all the binding which variables convert into table cell

VBScript Example
Call Report0.StopRegisterVaraibleChange()

(61) UnBindingEventChange method

UnBindingEventChange

Unbound the table cell change events

Define

UnBindingEventChange(sheetIndex)

Parameter

Name	Required/Optional	Data Type	Description
sheetIndex	Required	**Int**	The worksheet index ,starting from 0

Example

Unbound the table cell change events

VBScript Example
Call Report0.UnBindingEventChange(0)

(62) TriggerTimeOrder method

TriggerTimeOrder

Set the query time ascending and descending order sorting

Define

TriggerTimeOrder(order)

Parameter

Name	Required/Optional	Data Type	Description
order	Required	**Int**	0 is in ascending order and 1 is in descending order

Example

Set the data for the query to be sorted in descending chronological order

VBScript Example

```
Call Report0.ClearCellContent()
Call Report0.SetWorkSheetStartTime(0,Datetimepicker0.ValueTime)
Call Report0.SetWorkSheetEndTime(0,Datetimepicker1.ValueTime)
Call Report0.SetVarRecordRulerName(0,"s1")
Call Report0.TriggerTimeOrder(1)
Call Report0.QueryHistoryData()
```

(63) UnRegisterVariableToCell method

UnRegisterVariableToCell

Unbundle variables convert into table cell

Define

UnRegisterVariableToCell(variablePath , rowIndex , columnIndex)

Parameter

Name	Required/Optional	Data Type	Description
variablePath	Required	**String**	The variable path
rowIndex	Required	**Int**	The number of rows,starting from 1
columnIndex	Required	**Int**	The number of columns,starting from 1

Example

Unbundle variables convert into table cell

VBScript Example

```
Call Report0.UnRegisterVariableToCell("Var.Variable0",1,1)
```

(64) ReportColumnsVisiable method

ReportColumnsVisiable

Show the columns of the query

Define

VisiableReportColumns(columns)

Parameter

Name	Required/Optional	Data Type	Description
columns	Required	**String**	The number of columns

Example

Show the columns of the query

VBScript Example
Call Report0.VisiableReportColumns("Alarm")

【Property】

(1) Height property

Height

Height

Define

Double Height

Example

Set the height of the specified object to 100

VBScript Example
Report0.Height = 100

(2) IntervalDay property

IntervalDay

Get or setup the number of days that query data interval

Define

Int64 IntervalDay

(3) IntervalHour property

IntervalHour

Get or setup the number of hours that query data interval

Define

Int64 IntervalHour

(4) IntervalMillisecond property

IntervalMillisecond

Get or setup the number of milliseconds that query data interval

Define

Int64 IntervalMillisecond

(5) IntervalMinute property

IntervalMinute

Get or setup the number of minutes that query data interval

Define

Int64 IntervalMinute

(6) IntervalMonth property

IntervalMonth

Get or setup the number of months that query data interval

Define

Int64 IntervalMonth

(7) IntervalQuarter property

IntervalQuarter

Get or setup the number of quarters that query data interval

Define

Int64 IntervalQuarter

(8) IntervalSecond property

IntervalSecond

Get or setup the number of seconds that query data interval

Define

Int64 IntervalSecond

(9) IntervalUnit property

IntervalUnit

Time interval unit,ms is by millisecond;ss is by second;mi is by minute;hh is by hour;dd is by day;mm is by month;qq is by quarter;yy is by year

Define

Int64 IntervalUnit

(10) IntervalYear property

IntervalYear

Get or setup the number of years that query data interval

Define

Int64 IntervalYear

(11) IsShow property

IsShow

Whether display the specified object or not

Define

Boolean IsShow

Example

True ：The specified object is displayed on the screen False ：The specified object hidden on the screen

VBScript Example
Report0.IsShow = True Report0.IsShow = False

(12) Left property

Left

The left coordinate

Define

Double Left

Example

Object's left side of the coordinate values is 100 on the screen

VBScript Example

```
Report0.Left = 100
```

(13) Name property

Name

Name

Define

String Name

Example

Get the default name of the specified object

VBScript Example

```
TextBox0.Text = Report0.Name
```

(14) ToolTip property

ToolTip

Tooltip text

Define

String ToolTip

Example

Set the ToolTip of the specified object to "test"

VBScript Example

```
Report0.ToolTip = "test"
```

(15) Top property

Top

The top coordinate

Define

Double Top

Example

Object's the top of the coordinate value is 100 in the picture

VBScript Example

```
Report0.Top = 100
```

(16) Width property

Width

Width

Define

Double Width

Example

Set the width of the specified object to 100

VBScript Example

```
Report0.Width = 100
```

(17) ZIndex property

ZIndex

Layer index

Define

Int32 ZIndex

Example

Set the Zindex of the specified object to 3

VBScript Example

```
Report0.ZIndex = 3
```

12. HMIRecipeBrowser object

HMIRecipeBrowser

Recipe browser control

Methods list

	Name	Description
	ChangeRecipeBrowser	Change the recipe browser
	FindAnimation	Look for animation to modify the associated variables of the animation
	GetCurrentRecipeColumnInf	Get the content of the specified column which the row is selected
	GetRecipeItemName	Get the recipe item name which current selected
	Print	Print
	Refresh	Refresh
	Save	Save
	SaveRecipe	Save recipe

Property list

	Name	Description
	Height	Height
	IsShow	Whether display the specified object or not
	IsShowStatusBar	The properties of status bar
	IsShowToolBar	Whether to display the tool bar
	Left	The left coordinate
	Name	Name
	RecipeName	The name of the recipe
	ToolTip	Tooltip text
	Top	The top coordinate
	Width	Width
	ZIndex	Layer index

The following is the detailed description of the script:

【Method】

(1) ChangeRecipeBrowser method

ChangeRecipeBrowser

Change the recipe browser

Define

ChangeRecipeBrowser(RecipeBrowserName)

Parameter

Name	Required/Optional	Data Type	Description
RecipeBrowserName	Required	**String**	The name of recipe

Example

Change the recipe browser

VBScript Example
RecipeBrowser0.ChangeRecipeBrowser("test")

(2) FindAnimation method

FindAnimation

Look for animation to modify the associated variables of the animation

Define

FindAnimation(name)

Parameter

Name	Required/Optional	Data type	Description
name	Required	**String**	The name of animation

Example

Change the associated variables of the visibility animation of the object into Var.VariableGroup0.Variable1

VBScript Example
RecipeBrowser0.FindAnimation("HMIShowHideAnimation").Expression =

"Var.VariableGroup0.Variable1"

(3) GetCurrentRecipeColumnInf method

GetCurrentRecipeColumnInf

Get the content of the specified column which the row is selected

Define

GetCurrentRecipeColumnInf(columnName)

Parameter

Name	Required/Optional	Data type	Description
columnName	Required	**String**	Column name

Example

Get the content of the selected row which the column name is “water”

VBScript Example
TextBox0.Text = RecipeBrowser0.GetCurrentRecipeColumnInf("water")

(4) GetRecipeItemName method

GetRecipeItemName

Get the recipe item name which currently selected

Define

GetRecipeItemName()

Example

Get the recipe item name which currently selected

VBScript Example
TextBox.Text = RecipeBrowser0.GetRecipeItemName()

(5) Print method

Print

Print

Define

Print()

Example

Print

VBScript Example
RecipeBrowser0.Print()

(6) Refresh method

Refresh

Refresh

Define

Refresh()

Example

Refresh

VBScript Example
RecipeBrowser0.Refresh()

(7) Save method

Save

Save

Define

Save()

Example

Save

VBScript Example
RecipeBrowser0.Save()

(8) SaveRecipe method

SaveRecipe

Save recipe

Define

SaveRecipe()

Example

Save recipe

VBScript Example
RecipeBrowser0.SaveRecipe()

【Property】

(1) Height property

Height

Height

Define

Double Height

Example

Set the height of the specified object to 100

VBScript Example
RecipeBrowser0.Height = 100

(2) IsShow property

IsShow

Whether display the specified object or not

Define

Boolean IsShow

Example

True ：The specified object is displayed on the screen False ：The specified object hidden on the screen

VBScript Example
RecipeBrowser0.IsShow = True RecipeBrowser0.IsShow = False

(3) IsShowStatusBar property

IsShowStatusBar

The properties of status bar

Define

Boolean IsShowStatusBar

Example

Set the recipe browser window（RecipeBrowserWindow）to:do not display the status bar

VBScript Example

```
RecipeBrowser0.IsShowStatusBar = False
```

(4) IsShowToolBar property

IsShowToolBar

Whether to display the tool bar

Define

Boolean IsShowToolBar

Example

Set the recipe browser window（RecipeBrowserWindow）to:do not display the tool bar

VBScript Example

```
RecipeBrowser0.IsShowToolBar = False
```

(5) Left property

Left

The left coordinate

Define

Double Left

Example

Object's left side of the coordinate values is 100 on the screen

VBScript Example

```
RecipeBrowser0.Left = 100
```

(6) Name property

Name

Name

Define

String Name

Example

Get the default name of the specified object

VBScript Example

```
TextBox0.Text = RecipeBrowser0.Name
```

(7) RecipeName property

RecipeName

The name of the recipe

Define

String RecipeName

Example

Set the name of the recipe browser window to AAA

VBScript Example

```
RecipeBrowser0.RecipeName = "AAA"
```

(8) ToolTip property

ToolTip

Tooltip text

Define

String ToolTip

Example

Set the ToolTip of the specified object to "test"

VBScript Example

```
RecipeBrowser0.ToolTip = "test"
```

(9) Top property

Top

The top coordinate

Define

Double Top

Example

Object's the top of the coordinate value is 100 in the picture

VBScript Example

```
RecipeBrowser0.Top = 100
```

(10) Width property

Width

Width

Define

Double Width

Example

Set the width of the specified object to 100

VBScript Example

```
RecipeBrowser0.Width = 100
```

(11) ZIndex property

ZIndex

Layer index

Define

Int32 ZIndex

Example

Set the Zindex of the specified object to 3

VBScript Example
RecipeBrowser0.ZIndex = 3

13. HMIRuler object

HMIRuler

Ruler control

Methods list

	Name	Description
	FindAnimation	Look for animation to modify the associated variables of the animation
	Print	Print
	Save	Save
	StartTimer	Start timer
	StopTimer	Stop timer

Property list

	Name	Description
	Height	Height
	IsShow	Whether display the specified object or not
	Left	The left coordinate
	Name	Name
	Textvisibility	Whether to display the status column
	ToolTip	Tooltip text
	Top	The top coordinate
	Width	Width
	ZIndex	Layer index

The following is the detailed description of the script:

【Method】

(1) FindAnimation method

FindAnimation

Look for animation to modify the associated variables of the animation

Define

FindAnimation(name)

Parameter

Name	Required/Optional	Data type	Description
name	Required	**String**	The name of animation

Example

Change the associated variables of the visibility animation of the object into Var.VariableGroup0.Variable1

VBScript Example
Ruler0.FindAnimation("HMIShowHideAnimation").Expression = "Var.VariableGroup0.Variable1"

(2) Print method

Print

Print

Define

Print()

Example

Print

VBScript Example
Ruler0.Print()

(3) Save method

Save

Save

Define

Save()

Example

Save

VBScript Example
Ruler0.Save()

(4) StartTimer method

StartTimer

Start timer

Define

StartTimer()

Example

Start Timer

VBScript Example
Ruler0.StartTimer()

(5) StopTimer method

StopTimer

Stop timer

Define

StopTimer()

Example

Stop Timer

VBScript Example
Ruler0.StopTimer()

【Property】

(1) Height property

Height

Height

Define

Double Height

Example

Set the height of the specified object to 100

VBScript Example
Ruler0.Height = 100

(2) IsShow property

IsShow

Whether display the specified object or not

Define

Boolean IsShow

Example

True ：The specified object is displayed on the screen False ：The specified object hidden on the screen

VBScript Example
Ruler0.IsShow = True Ruler0.IsShow = False

(3) Left property

Left

The left coordinate

Define

Double Left

Example

Object's left side of the coordinate values is 100 on the screen

VBScript Example
Ruler0.Left = 100

(4) Name property

Name

Name

Define

String Name

Example

Get the default name of the specified object

VBScript Example

```
TextBox0.Text = Ruler0.Name
```

(5) Textvisibility property

Textvisibility

The display of the status column

Define

Boolean Textvisibility

Example

Whether to display the status column

VBScript Example

```
Ruler0.Textvisibility = True
```

(6) ToolTip property

ToolTip

Tooltip text

Define

String ToolTip

Example

Set the ToolTip of the specified object to "test"

VBScript Example

```
Ruler0.ToolTip = "test"
```

(7) Top property

Top

The top coordinate

Define

Double Top

Example

Object's the top of the coordinate value is 100 in the picture

VBScript Example
Ruler0.Top = 100

(8) Width property

Width

Width

Define

Double Width

Example

Set the width of the specified object to 100

VBScript Example
Ruler0.Width = 100

(9) ZIndex property

ZIndex

Layer index

Define

Int32 ZIndex

Example

Set the Zindex of the specified object to 3

VBScript Example
Ruler0.ZIndex = 3

14. HMICircleGauge object

HMICircleGauge

Dashboard control

Methods list

	Name	Description
	FindAnimation	Look for animation to modify the associated variables of the animation
	Print	Print
	Save	Save
	StartTimer	Start timer
	StopTimer	Stop timer

Property list

	Name	Description
	Height	Height
	IsShow	Whether display the specified object or not
	Left	The left coordinate
	Name	Name
	Textvisibility	Whether to display the status column
	ToolTip	Tooltip text
	Top	The top coordinate
	Width	Width
	ZIndex	Layer index

The following is the detailed description of the script:

【Method】

(1) FindAnimation method

FindAnimation

Look for animation to modify the associated variables of the animation

Define

FindAnimation(name)

Parameter

Name	Required/Optional	Data type	Description

name	Required	**String**	The name of animation

Example

Change the associated variables of the visibility animation of the object into Var.VariableGroup0.Variable1

VBScript Example

```
CircleGauge0.FindAnimation("HMIShowHideAnimation").Expression = "Var.VariableGroup0.Variable1"
```

(2) Print method

Print

Print

Define

Print()

Example

Print

VBScript Example

```
CircleGauge0.Print()
```

(3) Save method

Save

Save

Define

Save()

Example

Save

VBScript Example

```
CircleGauge0.Save()
```

(4) StartTimer method

StartTimer

Start timer

Define

StartTimer()

Example

Start Timer

VBScript Example
CircleGauge0.StartTimer()

(5) StopTimer method

StopTimer

Stop timer

Define

StopTimer()

Example

Stop Timer

VBScript Example
CircleGauge0.StopTimer()

【Property】

(1) Height property

Height

Height

Define

Double Height

Example

Set the height of the specified object to 100

VBScript Example

```
CircleGauge0.Height = 100
```

(2) IsShow property

IsShow

Whether display the specified object or not

Define

Boolean IsShow

Example

True ：The specified object is displayed on the screen False ：The specified object hidden on the screen

VBScript Example

```
CircleGauge0.IsShow = True    CircleGauge0.IsShow = False
```

(3) Left property

Left

The left coordinate

Define

Double Left

Example

Object's left side of the coordinate values is 100 on the screen

VBScript Example

```
CircleGauge0.Left = 100
```

(4) Name property

Name

Name

Define

String Name

Example

Get the default name of the specified object

VBScript Example
TextBox0.Text = CircleGauge0.Name

(5) Textvisibility property

Textvisibility

The display of the status column

Define

Boolean Textvisibility

Example

Whether to display the status column

VBScript Example
CircleGauge0.Textvisibility = True

(6) ToolTip property

ToolTip

Tooltip text

Define

String ToolTip

Example

Set the ToolTip of the specified object to "test"

VBScript Example
CircleGauge0.ToolTip = "test"

(7) Top property

Top

The top coordinate

Define

Double Top

Example

Object's the top of the coordinate value is 100 in the picture

VBScript Example
CircleGauge0.Top = 100

(8) Width property

Width

Width

Define

Double Width

Example

Set the width of the specified object to 100

VBScript Example
CircleGauge0.Width = 100

(9) ZIndex property

ZIndex

Layer index

Define

Int32 ZIndex

Example

Set the Zindex of the specified object to 3

VBScript Example
CircleGauge0.ZIndex = 3

15. HMIWebBrowser object

HMIWebBrowser

The web browser control

Methods list

	Name	Description
	FindAnimation	Look for animation to modify the associated variables of the animation

Property list

	Name	Description
	Height	Height
	IsShow	Whether display the specified object or not
	Left	The left coordinate
	Name	Name
	ToolTip	Tooltip text
	Top	The top coordinate
	Url	Get or set the homepage address
	Width	Width
	ZIndex	Layer index

The following is the detailed description of the script:

【Method】

(1) FindAnimation method

FindAnimation

Look for animation to modify the associated variables of the animation

Define

FindAnimation(name)

Parameter

Name	Required/Optional	Data type	Description
name	Required	**String**	The name of animation

Example

Change the associated variables of the visibility animation of the object into Var.VariableGroup0.Variable1

VBScript Example

```
WebBrowser0.FindAnimation("HMIShowHideAnimation").Expression = "Var.VariableGroup0.Variable1"
```

【Property】

(1) Height property

Height

Height

Define

Double Height

Example

Set the height of the specified object to 100

VBScript Example

```
WebBrowser0.Height = 100
```

(2) IsShow property

IsShow

Whether display the specified object or not

Define

Boolean IsShow

Example

True ：The specified object is displayed on the screen False ：The specified object hidden on the screen

VBScript Example

```
WebBrowser0.IsShow = True     WebBrowser0.IsShow = False
```

(3) Left property

Left

The left coordinate

Define

Double Left

Example

Object's left side of the coordinate values is 100 on the screen

VBScript Example

```
WebBrowser0.Left = 100
```

(4) Name property

Name

Name

Define

String Name

Example

Get the default name of the specified object

VBScript Example

```
TextBox0.Text = WebBrowser0.Name
```

(5) ToolTip property

ToolTip

Tooltip text

Define

String ToolTip

Example

Set the ToolTip of the specified object to "test"

VBScript Example

```
WebBrowser0.ToolTip = "test"
```

(6) Top property

Top

The top coordinate

Define

Double Top

Example

Object's the top of the coordinate value is 100 in the picture

VBScript Example

```
WebBrowser0.Top = 100
```

(7) Url property

Url

Get or set the homepage address

Define

Double Url

Example

Set the homepage address to Baidu

VBScript Example

```
WebBrowser0.Url = "https://www.baidu.com"
```

(8) Width property

Width

Width

Define

Double Width

Example

Set the width of the specified object to 100

VBScript Example

```
WebBrowser0.Width = 100
```

(9) ZIndex property

ZIndex

Layer index

Define

Int32 ZIndex

Example

Set the Zindex of the specified object to 3

VBScript Example
WebBrowser0.ZIndex = 3

20.3.1.4 Command

1. HMIActionCmd object

HMIActionCmd

Action script command

Methods list

	Name	Description
	AnalogValueInput	Analog the input action
	ButtonValueInput	The button input action
	DiscreteValueInput	The discrete value input action
	StringValueInput	The string input action
	ToggleInput	The setting input

The following is the detailed description of the script:

【Method】

(1) AnalogValueInput method

AnalogValueInput

Analog input action

Define

AnalogValueInput(expression ,minValue ,maxValue)

Parameter

Name	Required/Optional	Data Type	Description

expression	Required	**String**	The variable expression
minValue	Required	**Double**	The minimum value allowed to enter
maxValue	Required	**Double**	The maximum value allowed to enter

Example

For example，add the input action to the graphics(as shown VBScript below),when runtime, click this graphics dialog box will pop up the analog of input box,enter a value, click ok to assign the input values to the associated variables.

VBScript Example
ActionCmd.AnalogValueInput("Var.VariableGroup1.Variable1",0,100）

(2) ButtonValueInput method

ButtonValueInput

The button input action

Define

ButtonValueInput(expression ,action ,value) action=0-Setting,1-Increase,2-Decrease,3-Multiply,4-Divide

Parameter

Name	Required/Optional	Data Type	Description
expression	Required	**String**	The variable expression
action	Required	**Int32**	The name of the action,action=0-Setting,1-Increase,2-Decrease,3-Multiply,4-Divide
value	Required	**Double**	The operating value

Example

For example，add the input action to the graphics(as VBScript below),when runtime, click this graphics dialog box will assign the input values to the associated variables.The following code mean that click the graphic one time will increase 10 to the variable's value

VBScript Example
ActionCmd.ButtonValueInput("Var.VariableGroup1.Variable1",1,10)

(3) DiscreteValueInput method

DiscreteValueInput

The discrete value input action

Define

DiscreteValueInput(expression ,trueValue ,falseValue)

Parameter

Name	Required/Optional	Data Type	Description
expression	Required	**String**	The variable expression
trueValue	Required	**String**	The displayed value when the expression value is True
falseValue	Required	**String**	The displayed value when the expression value is False

Example

For example，add the discrete value input action to the graphics(as VBScript below),when runtime, click this graphics dialog box will pop up the discrete value input box,enter a value, click ok to assign the input values to the associated variables

VBScript Example
ActionCmd.DiscreteValueInput("Var.VariableGroup1.Variable1","True","False"）

(4) StringValueInput method

StringValueInput

The string input action

Define

StringValueInput(expression ,isPassWord)

Parameter

Name	Required/Optional	Data Type	Description
expression	Required	**String**	The variable expression
isPassWord	Required	**Boolean**	Whether the password is displayed

Example

For example，add the string input action to the graphics(as VBScript below),when runtime, click this graphics dialog box will pop up the string input box,enter a value, click ok to assign the input values to the associated variables

VBScript Example
ActionCmd.StringValueInput("Var.VariableGroup1.Variable1",False）

(5) ToggleInput method

ToggleInput

The setting input

Define

ToggleInput(expression)

Parameter

Name	Required/Optional	Data Type	Description
expression	Required	**String**	The name of variable

Example

Set the value of "variable" to "0" or "1"

VBScript Example
ActionCmd.ToggleInput("Var.Script.variable")

2. HMICmd object

HMICmd

Operating picture commands

Methods list

	Name	Description
	CloseDialogWindow	Close the dialog window
	CloseWindow	Close window
	ExitApplication	Exit the program
	ExitApplication	Exit the program
	GetEnterKey	To obtain the value of the Enter Key

MaxWindow	Maximize the window
MinWindow	Minimize the Window
OpenDialogWindow	Open the dialog window
OpenDialogWindow	Open the dialog window
OpenDialogWindow	Open the dialog window
OpenDialogWindow	Open the dialog window
OpenDialogWindow	Open the dialog window
OpenWindow	Open the window
OpenWindowAndCloseOther	Just open the specified screen, close the other picture
PrintFullWorkSpace	Print the whole workspace

The following is the detailed description of the script:

【Method】

(1) CloseDialogWindow method

CloseDialogWindow

Close the dialog window

Define

CloseDialogWindow()

Example

Close the dialog window

VBScript Example

```
HMICmd.CloseDialogWindow()
```

(2) CloseWindow method

CloseWindow

Close window

Define

CloseWindow(windowName)

Parameter

Name	Required/Optional	Data Type	Description
windowName	Required	**String**	Screen name

Example

Close the window of screen 0.Close the window of screen 0,screen 1,screen 2,multiple images and so on.

VBScript Example
HMICmd.CloseWindow("Screen0") or HMICmd.CloseWindow("Screen0,Screen1,Screen2")

(3) ExitApplication method

ExitApplication

Exit application

Define

ExitApplication(isNeedNotice)

Parameter

Name	Required/Optional	Data Type	Description
isNeedNotice	Required	**Bool**	Is notice

Example

Exit application

VBScript Example
HMICmd.ExitApplication(True)

(4) ExitApplication method

ExitApplication

Exit application

Define

ExitApplication()

Example

Exit application

VBScript Example
HMICmd.ExitApplication()

(5) GetEnterKey method

GetEnterKey

To obtain the value of Enter Key

Define

GetEnterKey(eventArgs)

Parameter

Name	Required/Optional	Data Type	Description
eventArgs	Required	**String**	EventPara.EventArgs

Example

1.Add a "TextBox" and a "Rectangle" to the window;

2.Add a "TimerTicked Program" to "Window Program":TextBox0.Focus(),make the cursor in the TextBox0;

3.Configure script to obtain the value of the Enter Key,if get the value of the Enter Key,the color of Rectangle is red,else the color is blue;

VBScript Example

TextBox0.Focus()
If HMICmd.GetEnterKey(EventPara.EventArgs) = true then
Rectangle0.Fill = Colors.Red
Else
Rectangle0.Fill = Colors.Blue
End If

(6) MaxWindow method

MaxWindow

Maximize the window

Define

MaxWindow()

Example

Maximize the window

VBScript Example
HMICmd.MaxWindow()

(7) MinWindow method

MinWindow

Minimize the Window

Define

MinWindow()

Example

Minimize the Window

VBScript Example
HMICmd.MinWindow()

(8) OpenDialogWindow method

OpenDialogWindow

Open dialog window

Define

OpenDialogWindow(windowName)

Parameter

Name	Required/Optional	Data Type	Description
windowName	Required	**String**	The screen name

Example

Open dialog window

VBScript Example
HMICmd.OpenDialogWindow("Window0")

(9) OpenDialogWindow method

OpenDialogWindow

Open dialog window

Define

OpenDialogWindow(windowName , isShowTitle)

Parameter

Name	Required/Optional	Data Type	Description
windowName	Required	**String**	The window name
isShowTtile	Required	**Bool**	Is show title

Example

Open dialog window0

VBScript Example
Call HMICmd.OpenDialogWindow("window0",true)

(10) OpenDialogWindow method

OpenDialogWindow

Open dialog window in the specified screen

Define

OpenDialogWindow(windowName,screenNum,isMode)

Parameter

Name	Required/Optional	Data Type	Description
windowName	Required	**String**	The window name
screenNum	Required	**Int**	The screen number
isModel	Required	**Bool**	Is modal window or not

Example

Open dialog "window0" in the "screen1"

VBScript Example
Call HMICmd.OpenDialogWindow("window0",1,true)

(11) OpenDialogWindow method

OpenDialogWindow

Open dialog window in the specified screen

Define

OpenDialogWindow(windowName,screenNum,isMode,isFullScreen)

Parameter

Name	Required/Optional	Data Type	Description
windowName	Required	**String**	The window name
screenNum	Required	**Int**	The screen number
isModel	Required	**Bool**	Is modal window or not
isFullScreen	Required	**Bool**	Is full screen or not

Example

Open dialog "window0" in the "screen1"

VBScript Example
Call HMICmd.OpenDialogWindow("Window0",1,true,true)

(12) OpenDialogWindow method

OpenDialogWindow

Open dialog window in the specified screen

Define

OpenDialogWindow(windowName,screenNum,isMode,left,top)

Parameter

Name	Required/Optional	Data Type	Description
windowName	Required	**String**	The window name
screenNum	Required	**Int**	The screen number
isModel	Required	**Bool**	Is modal window or not
left	Optional	**Double**	The left end position of the window
top	Optional	**Double**	The top end position of the window

Example

Open dialog "window0" in the "screen1"

VBScript Example
Call HMICmd.OpenDialogWindow("Window0",1,true,100,200)

(13) OpenWindow method

OpenWindow

Open the window

Define

OpenWindow(windowName)

Parameter

Name	Required/Optional	Data Type	Description
windowName	Required	**String**	The screen name

Example

Open the screen 0 and 1 together

VBScript Example
HMICmd.OpenWindow("Screen0,Screen1")

(14) OpenWindowAndCloseOther method

OpenWindowAndCloseOther

Just open the specified screen, close the other screen

Define

OpenWindowAndCloseOther(windowName)

Parameter

Name	Requireed/Optional	Data Type	Description
windowName	Required	**String**	The screen name

Example

Open the screen 0 and 1 together, close the other screen

VBScript Example
HMICmd.OpenWindowAndCloseOther("Screen0,Screen1")

(15) PrintFullWorkSpace method

PrintFullWorkSpace

Print the whole workspace

Define

PrintFullWorkSpace()

Example

Print the whole workspace

VBScript Example
HMICmd.PrintFullWorkSpace()

20.3.2 IO communication

1. IODataExchanger object

IODataExchanger

The hardware device

Property list

	Name	Description
	Devices	Devices collection

The following is the detailed description of the script:

【Property】

(1) Devices property

Devices

Devices collection

Define

DeviceCollection Devices

Description

Annotation

This property is read-only

Example

Obtain a device according to the name

VBScript Example

```
Dim device = IO.Devices[devicename]
```

2. IODataItem object

IODataItem

IO data

Methods list

	Name	Description
	GetDeviceStatus	Get device status

Property list

	Name	Description
	Address	The communication address
	DataConversionTypeInt	Data conversion type Int, 0-no transformation, 1-linear transformation, 2-root transformation
	Description	Description
	MeasureHigh	The maximum value of engineering
	MeasureLow	The minimum value of engineering

	Name	Address name
	Quality	The quality of address unit
	ReadWriteTypeInt	Reading and writing type Int, 0 - read/write, 1 - read-only, 2 - just write
	ScanTime	Scan cycle
	UniqueIdentifier	The only label ID
	UpdateTime	The update time for scanning
	Value	Get value
	ValueChangeTime	The value change time
	ValueType	The value type
	VariablePath	Alarm variable absolute path

The following is the detailed description of the script:

【Method】

(1) GetDeviceStatus Method

GetDeviceStatus

Get device status

Define

Int GetDeviceStatus()

Return

Int

Example

0 means normal communication, 2 means disconnection or stop

VBScript Example

```
IO.Device0.GetDeviceStatus()
```

【Property】

(1) Address property

Address

The communication address

Define

String Address

Description

Annotation

This property is read-only

Example

Get address

VBScript Example

```
Text0.Text = IO.Device0.NewAddres0.Address
```

(2) DataConversionTypeInt property

DataConversionTypeInt

Data conversion type Int, 0-no transformation, 1-linear transformation, 2-root transformation

Define

Int32 DataConversionTypeInt

Description

Annotation

This property is read/write

Example

Set to linear transformation

VBScript Example

```
IO.Device0.NewAddres0.DataConversionTypeInt = 1
```

(3) Description property

Description

Description

Define

String Description

Description

Annotation
This property is read/write

Example

Get description information of "NewAddress0"

VBScript Example
Text0.Text = IO.Device0.NewAddress0.Description

(4) MeasureHigh property

MeasureHigh

The maximum value of engineering

Define

Object MeasureHigh

Description

Annotation
This property is read/write

Example

Set the maximum value of address of engineering

VBScript Example
IO.Device0.NewAddres0.MeasureHigh = 100

(5) MeasureLow property

MeasureLow

The minimum value of engineering

Define

Object MeasureLow

Description

Annotation
This property is read/write

Example

Set the minimum value of the address to 0

VBScript Example
IO.Device0.NewAddres0.MeasureLow = 0

(6) Name property

Name

Address name

Define

String Name

Description

Annotation
This property is read-only

Example

Get address name

VBScript Example
Text0.Text = IO.Device0.NewAddres0.Name

(7) Quality property

Quality

The quality of address unit

Define

Enum Quality

Description

Annotation
This property is read-only

Example

To obtain the quality of the address unit value

VBScript Example
Text0.Text = IO.Device0.Address1.Quality

(8) ReadWriteType property

ReadWriteType

Reading and writing type Int, 0 - read/write, 1 - read-only, 2 - just write

Define

Int32 ReadWriteType

Description

Annotation
This property is read/write

Example

Set address to read/write

VBScript Example
IO.Device0.NewAddres0.ReadWriteType = 0

(9) ScanTime property

ScanTime

Scan cycle

Define

Int32 ScanTime

Description

Annotation
This property is read/write

Example

Get the scan cycle

VBScript Example
Text0.Text = IO.Device0.NewAddres0.ScanTime

(10) UniqueIdentifier property

UniqueIdentifier

The only label ID

Define

Int32 UniqueIdentifier

Description

Annotation
This property is read-only

Example

Get the only label ID

VBScript Example
Text0.Text = IO.Device0.NewAddres0.UniqueIdentifier

(11) UpdateTime property

UpdateTime

The update time for scanning

Define

String UpdateTime

Description

Annotation
This property is read-only

Example

To obtain the update time for scanning

VBScript Example
Text0.Text = IO.Device0.NewAddres0.UpdateTime

(12) Value property

Value

Get value

Define

Int32 Value

Description

Annotation
This property is read-only

Example

To obtain the value of the address unit

VBScript Example
Text0.Text = IO.Device0.Address1.Value

(13) ValueChangeTime property

ValueChangeTime

The value change time

Define

String ValueChangeTime

Description

Annotation
This property is read-only

Example

Get the value change time

VBScript Example
Text0.Text = IO.Device0.Address1.ValueChangeTime

(14) ValueType property

ValueType

The value's type

Define

Enum ValueType

Description

Annotation
This property is read-only

Example

Get the type of value

VBScript Example
Text0.Text = IO.Device0.Address1.ValueType

(15) VariablePath property

VariablePath

Alarm variable absolute path

Define

String VariablePath

Description

Annotation
This property is read-only

Example

The specified address binding to the variable path

VBScript Example
IO.Device0.NewAddres0.VariablePath = "var.NewVariable"

3. IODevice object

IODevice

Equipment base class and object associated with the underlying IoServer

Property list

	Name	Description
	Description	Description
	DeviceConnectLastTime	The last connection time of the device
	DeviceConnectPara	Get the communications information of device
	DeviceDisconnectLastTime	The last disconnection time of the device
	DeviceType	The device type
	Name	The device name
	Status	Get the equipment status, 0-normal,2-disconnect or stop
	SpendTime	Get the spend time that the device needed to scan

The following is the detailed description of the script:

【Property】

(1) Description property

Description

Description

Define

String Description

Description

Annotation
This property is read-only

Example

Get the description of device

VBScript Example
Text0.Text = IO.Device0.Description

(2) DeviceConnectLastTime property

DeviceConnectLastTime

The last connection time of the device

Define

String DeviceConnectLastTime

Description

Annotation
This property is read-only

Example

Get the last connection time of device

VBScript Example
Text0.Text = IO.Device0.DeviceConnectLastTime

(3) DeviceConnectPara property

DeviceConnectPara

Get the communications information of device

Define

String DeviceConnectPara

Description

Annotation

This property is read-only

Example

Get the communications information of device

VBScript Example

```
Text0.Text = IO.Device0.DeviceConnectPara
```

(4) DeviceDisconnectLastTime property

DeviceDisconnectLastTime

The last disconnection time of the device

Define

String DeviceDisconnectLastTime

Description

Annotation

This property is read-only

Example

Get the last disconnection time of the device

VBScript Example

```
Text0.Text = IO.Device0.DeviceDisconnectLastTime
```

(5) DeviceType property

DeviceType

The device type

Define

String DeviceType

Description

Annotation

This property is read-only

Example

Get device type

VBScript Example

```
Text0.Text = IO.Device0.DeviceType
```

(6) Name property

Name

The device name

Define

String Name

Description

Annotation

This property is read-only

Example

Get device name

VBScript Example

```
Text0.Text = IO.Device0.Name
```

(7) Status property

Status

Get the equipment status

Define

Int32 Status

Description

Annotation

This property is read/write

Return

0-normal,2-disconnect or stop

Example

Get equipment status

VBScript Example

```
Text0.Text = IO.Device0.Status
```

(8) SpendTime property

SpendTime

Get the spend time that the device needed to scan

Define

Int32 SpendTime

Description

Annotation

This property is read-only

Example

Get the spend time that the device needed to scan

VBScript Example

```
Text0.Text = IO.Device0.SpendTime
```

4. IOManagement object

IOManagement

IO management

Methods list

	Name	Description
	GetDeviceConnectLastTime	The last connection time of device
	GetDeviceConnectPara	Get communications information of device
	GetDeviceDisconnectLastTime	Get the last disconnection time of the device
	GetDeviceStatus	Get device status
	GetItemAddress	Get address information that corresponds to the address name
	GetLocalIP	Get local IP
	GetLocalLansIP	Get all Local Lans IP
	GetMacAddress	Get mac address
	GetScanDeviceSpendTime	Get the spend time that the device needed to scan
	ReadRemoteFile	Get the file on remote device
	ReStart	Restart device
	SetDeviceCOMBaudRate	Modify COM baud rate dynamically
	SetDeviceCOMPort	Modify COM port dynamically
	SetDeviceSocketIP	Modify IP address dynamically
	SetDeviceSocketPort	Modify socket port dynamically
	StartDevice	Start device
	StopDevice	Stop device
	WriteRemoteFile	Write local file to remote device

The following is the detailed description of the script:

【Method】

(1) GetDeviceConnectLastTime method

GetDeviceConnectLastTime

The last connection time of the device

Define

GetDeviceConnectLastTime(deviceName)

Parameter

Name	Required/Optional	Data Type	Description
deviceName	Required	**String**	The device name

Return

String

Example

Get the last connection time of device

VBScript Example
IOCmd.GetDeviceConnectLastTime("Device0")

(2) GetDeviceConnectPara method

GetDeviceConnectPara

Get the communications information of device

Define

GetDeviceConnectPara(deviceName)

Parameter

Name	Required/Optional	Data Type	Description
deviceName	Required	**String**	The device name

Return

String

Example

Get the communications information of device

VBScript Example
IOCmd.GetDeviceConnectPara("Device0")

(3) GetDeviceDisconnectLastTime method

GetDeviceDisconnectLastTime

Get the last disconnection time of device

Define

GetDeviceDisconnectLastTime(deviceName)

Parameter

Name	Required/Optional	Data Type	Description
deviceName	Required	**String**	The device name

Return

String

Example

Get the last disconnection time of device

VBScript Example
IOCmd.GetDeviceDisconnectLastTime("Device0")

(4) GetDeviceStatus method

GetDeviceStatus

Get device status

Define

GetDeviceStatus(devicename)

Parameter

Name	Required/Optional	Data Type	Description
deviceName	Required	**String**	Device name

Return

Int

Example

0-normal,2-disconnect or stop

VBScript Example
IOCmd.GetDeviceStatus("Device0")

(5) GetItemAddress method

GetItemAddress

Get address information that corresponds to the address name

Define

GetItemAddress(deviceName,addressName)

Parameter

Name	Required/Optional	Data Type	Description
deviceName	Required	**String**	The device name
addressName	Required	**String**	The address name

Return

String

Example

Get address information that corresponds to the address name

VBScript Example
IOCmd.GetItemAddress("Device0","Address1")

(6) GetLocalIP method

GetLocalIP

Get local IP

Define

GetLocalIP( )

Return

String

Example

Get local IP

VBScript Example
`IOCmd.GetLocalIP()`

(7) GetLocalLansIP method

GetLocalLansIP

Get all Local Lans IP

Define

GetLocalLansIP()

Return

String

Example

Get all Local Lans IP

VBScript Example

```
List = IOCmd.GetLocalLansIP()
strs = ""
For i=0 To List.Length-1
    strs =strs + CStr(List(i)) + " , "
Next
MsgBox strs
```

(8) GetMacAddress method

GetMacAddress

Get mac address

Define

GetMacAddress(ip)

Parameter

Name	Required/Optional	Data Type	Description
ip	Required	**String**	IP address

Return

String

Example

Get the device's mac address which specified IP

VBScript Example
IOCmd.GetMacAddress("192.168.1.1")

(9) GetScanDeviceSpendTime method

GetScanDeviceSpendTime

Get the spend time that the device needed to scan

Define

GetScanDeviceSpendTime(deviceName)

Parameter

Name	Required/Optional	Data Type	Description
deviceName	Required	**String**	Device name

Return

String

Example

Get the spend time that the device needed to scan

VBScript Example
IOCmd.GetScanDeviceSpendTime("Device0")

(10) ReadRemoteFile method

ReadRemoteFile

Get the file on remote device

Define

String ReadRemoteFile(localPath,IsReplaceFile,deviceName,remoteIP,port,remoteFilePath,timeOutSeconds,isDeleteFileReadFinish)

Parameter

Name	Required/Optional	Data Type	Description

localPath	Required	**String**	Local path
IsReplaceFile	Required	**Bool**	Whether replace or not
deviceName	Required	**String**	The device name
remoteIP	Required	**String**	The IP of the remote device
port	Required	**Int**	Port
remoteFilePath	Required	**String**	The file path on the remote device
timeOutSeconds	Required	**Int**	Timeout
isDeleteFileReadFinish	Required	**Bool**	Whether delete file or not after finish

Return

String

Example

Get the file on remote device

VBScript Example
IOCmd.ReadRemoteFile("D:\",true,"device0","192.168.1.1",21,"E:\",100,false)

(11) ReStart method

ReStart

Restart device

Define

Int ReStart(deviceName)

Parameter

Name	Required/Optional	Data Type	Description
deviceName	Required	**String**	The device name

Return

String

Example

Restart device

VBScript Example
IOCmd.ReStart("Device0")

(12) SetDeviceCOMBaudRate method

SetDeviceCOMBaudRate

Modify the COM baud rate dynamically

Define

bool SetDeviceCOMBaudRate(deviceName,baudRate)

Parameter

Name	Required/Optional	Data Type	Description
deviceName	Required	**String**	The device name
baudRate	Required	**String**	Baud rate

Return

Bool

Example

Modify the COM baud rate dynamically

VBScript Example
IOCmd.SetDeviceCOMBaudRate("Device0","115200")

(13) SetDeviceCOMPort method

SetDeviceCOMPort

Modify the COM port dynamically

Define

bool SetDeviceCOMPort(deviceName,comPort)

Parameter

Name	Required/Optional	Data Type	Description
deviceName	Required	**String**	The device name

comPort	Required	**String**	The COM port

Return

Bool

Example

Modify the COM port dynamically

VBScript Example
IOCmd.SetDeviceCOMPort("Device0","COM1")

(14) SetDeviceSocketIP method

SetDeviceSocketIP

Modify IP address dynamically

Define

bool SetDeviceSocketIP(deviceName,ip)

Parameter

Name	Required/Optional	Data Type	Description
deviceName	Required	**String**	The device name
ip	Required	**String**	IP address

Return

Bool

Example

Modify IP address dynamically

VBScript Example
IOCmd.SetDeviceSocketIP("Device0","192.168.1.100")

(15) SetDeviceSocketPort method

SetDeviceSocketPort

Modify socket port dynamically

Define

bool SetDeviceSocketPort(deviceName,port)

Parameter

Name	Required/Optional	Data Type	Description
deviceName	Required	**String**	The device name
port	Required	**String**	Port

Return

Bool

Example

Modify socket port dynamically

VBScript Example
IOCmd.SetDeviceSocketPort("Device0","100")

(16) StartDevice method

StartDevice

Start device

Define

StartDevice(devicename)

Parameter

Name	Required/Optional	Data Type	Description
deviceName	Required	**String**	Device name（used to obtain the driver name）

Return

Int

Example

Start device

VBScript Example

```
IOCmd.StartDevice("Device0")
```

(17) StopDevice method

StopDevice

Stop device

Define

StopDevice(deviceName)

Parameter

Name	Required/Optional	Data Type	Description
deviceName	Required	**String**	Device name

Return

Int

Example

Stop device

VBScript Example

```
IOCmd.StopDevice("Device0")
```

(18) WriteRemoteFile method

WriteRemoteFile

Write local file to remote device

Define

String WriteRemoteFile(localPathFile,deviceName,remoteIP,port,remoteFilePath,timeOutSeconds)

Parameter

Name	Required/Optional	Data Type	Description
localPathFile	Required	**String**	Local file path
deviceName	Required	**String**	The device name
remoteIP	Required	**String**	Remote device's IP

port	Required	**Int**	Port
remoteFilePath	Required	**String**	File path of remote file
timeOutSeconds	Required	**Int**	Timeout

Return

String

Example

Write local file to remote device

VBScript Example
IOCmd.WriteRemoteFile("D:\","device0","192.168.1.1",21,"E:\",100)

20.3.3 Variable

1. AnalogVariable object

AnalogVariable

Analog variable

Methods list

	Name	Description
	ToString	The return value transfer into the specified format string
	ChangeValue	Change variable value

Property list

	Name	Description
	AbsolutePath	Variable absolute path
	DeadBand	Dead zone, variable extent of change as a minimum
	DecimalPlaces	Decimal digits, value range is from 0 to 15
	Description	The description of variable
	EngineeringUnit	The Engineering unit
	InitValue	Variable initial value
	IsInitValueSaved	Whether to save the current value when the program exits
	MaxValue	The maximum value of the variable

	MinValue	The minimum value of the variable
	Name	The variable name
	UniqueIdentifier	The unique identifier
	Value	The value of variable, object type
	ValueAndUnit	To obtain a variable's value with the additional project unit
	ValueAsDouble	Variable value

The following is the detailed description of the script:

【Method】

(1) ToString method

ToString

The return value transfer into the specified format string

Define

ToString(format)

Parameter

Name	Required/Optional	Data Type	Description
format	Required	**String**	Format

Return

The return value transfer into the specified format string

Example

VBScript Example

```
cc = Var.Analog1.ToString("C")
MsgBox cc
```

(2) ChangeValue method

ChangeValue

Change variable value

Define

ChangeValue(value,reason)

Parameter

Name	Required/Optional	Data Type	Description
value	Required	**Double**	Target value
reason	Required	**String**	This value will be entered into the operation variable

Return

void

Example

Change variable value

VBScript Example
Call Var.Analog1.ChangeValue(9.1,"test")

【Property】

(1) AbsolutePath property

AbsolutePath

Variable absolute path

Define

String AbsolutePath

Description

Annotation
This property is read-only

Example

Display the variable full path

VBScript Example
TextBox0.Text = Var.NewVariableGroup.NewVariable.AbsolutePath

(2) DeadBand property

DeadBand

Dead zone, variable extent of change as a minimum

Define

Double DeadBand

Description

Annotation

This property is read/write

Example

Set dead zone to 2

VBScript Example

```
Var.NewVariableGroup.NewVariable2.DeadBand = 2
```

(3) DecimalPlaces property

DecimalPlaces

Decimal digits, value range is from 0 to 15

Define

Int32 DecimalPlaces

Description

Annotation

This property is read/write

Example

Set the decimal digits to 5

VBScript Example

```
Var.NewVariableGroup.NewVariable2.DecimalPlaces = 5
```

(4) Description property

Description

The description of variable

Define

String Description

Description

Annotation

This property is read/write

Example

Modify the discriptions of a variables to "an analog variable"

VBScript Example

```
Var.NewVariableGroup.NewVariable.Description = "an analog variable"
```

(5) EngineeringUnit property

EngineeringUnit

The Engineering unit

Define

String EngineeringUnit

Description

Annotation

This property is read/write

Example

Set the engineering unit to degrees Celsius

VBScript Example

```
Var.NewVariableGroup.NewVariable2.EngineeringUnit = "Degrees Celsius"
```

(6) InitValue property

InitValue

Variable initial value

Define

Int InitValue

Description

Annotation

This property is read-only

Example

Get variable initial value

VBScript Example

```
TextBox0.Text = Var.Variable.InitValue
```

(7) IsInitValueSaved property

IsInitValueSaved

Whether to save the current value when the program exits

Define

Boolean IsInitValueSaved

Description

Annotation

This property is read/write

Example

Set the property to true

VBScript Example

```
Var.NewVariableGroup.NewVariable.IsInitValueSaved = true
```

(8) MaxValue property

MaxValue

The maximum value of the variable

Define

Double MaxValue

Description

Annotation
This property is read/write

Example

Set the maximum value of the variable to 200

VBScript Example
Var.NewVariableGroup.NewVariable2.MaxValue = 200

(9) MinValue property

MinValue

The minimum value of the variable

Define

Double MinValue

Description

Annotation
This property is read/write

Example

The minimum value of the variable to -200

VBScript Example
Var.NewVariableGroup.NewVariable2.MinValue = -200

(10) Name property

Name

The variable name

Define

String Name

Description

Annotation

This property is read-only

Example

Display the variable name

VBScript Example

```
TextBox0.Text = Var.NewVariableGroup.NewVariable.Name
```

(11) UniqueIdentifier property

UniqueIdentifier

The unique identifier

Define

Int32 UniqueIdentifier

Description

Annotation

This property is read-only

Example

Display unique identifier

VBScript Example

```
TextBox0.Text = Var.NewVariableGroup.NewVariable.UniqueIdentifier
```

(12) Value property

Value

The variable value,object type

Define

Object Value

Description

Annotation

This property is read/write

Example

Modify the value of the variable to 50

VBScript Example

```
Var.NewVariableGroup.NewVariable2.Value = 50
```

(13) ValueAndUnit property

ValueAndUnit

To obtain a variable's value with the additional project unit

Define

String ValueAndUnit

Description

Annotation

The property is read-only

Example

Display the value of variable and the engineering unit

VBScript Example

```
TextBox0.Text = Var.NewVariableGroup.NewVariable2.ValueAndUnit
```

(14) ValueAsDouble property

ValueAsDouble

The variable value

Define

Double ValueAsDouble

Description

Annotation
This property is read/write

Example

Modify the value of variable to 200

VBScript Example
Vars.NewVariableGroup.NewVariable.ValueAsDouble = 200

2. DigitalVariable object

DigitalVariable

The digital variable

Methods list

	Name	Description
	ChangeValue	Change variable value

Property list

	Name	Description
	AbsolutePath	Absolute path of variable
	Description	Discription of variable
	InitValue	Variable initial value
	IsInitValueSaved	Whether to save the current value when the program exits
	Name	Variable name
	UniqueIdentifier	The unique identifier
	Value	Value of variable,object type
	ValueAsBool	Value of variable

The following is the detailed description of the script:

【Method】

(1) ChangeValue method

ChangeValue

Change variable value

Define

ChangeValue(value,reason)

Parameter

Name	Required/Optional	Data Type	Description
value	Required	**Bool**	Target value
reason	Required	**String**	This value will be entered into the operation variable

Return

void

Example

Change variable value

VBScript Example

```
Call Var.Digital1.ChangeValue(True,"test")
```

【Property】

(1) AbsolutePath property

AbsolutePath

Variable absolute path

Define

String AbsolutePath

Description

Annotation

This property is read-only

Example

Display the variable full path

VBScript Example

```
TextBox0.Text = Var.NewVariableGroup.NewVariable.AbsolutePath
```

(2) Description property

Description

The description of variable

Define

String Description

Description

Annotation

This property is read/write

Example

Modify the discriptions of variables to "a digital variable"

VBScript Example

```
Var.NewVariableGroup.NewVariable.Description = "a digital variable"
```

(3) InitValue property

InitValue

Variable initial value

Define

Bool InitValue

Description

Annotation

This property is read/write

Example

Set variable initial value

VBScript Example

```
Var.Digital1.InitValue = false
```

(4) IsInitValueSaved property

IsInitValueSaved

Whether to save the current value when the program exits

Define

Boolean IsInitValueSaved

Description

Annotation

This property is read/write

Example

Set the property to true

VBScript Example

```
Var.NewVariableGroup.NewVariable.IsInitValueSaved = true
```

(5) Name property

Name

The variable name

Define

String Name

Description

Annotation

This property is read-only

Example

Display the variable name

VBScript Example

```
TextBox0.Text = Var.NewVariableGroup.NewVariable.Name
```

(6) UniqueIdentifier property

UniqueIdentifier

The only label ID

Define

Int32 UniqueIdentifier

Description

Annotation

This property is read-only

Example

Display the only label ID

VBScript Example

```
TextBox0.Text = Var.NewVariableGroup.NewVariable.UniqueIdentifier
```

(7) Value property

Value

The value of variable,object type

Define

Object Value

Description

Annotation

This property is read/write

Example

Set the value of the variable to 50

VBScript Example

```
Var.NewVariableGroup.NewVariable1.Value = 50
```

(8) ValueAsBool property

ValueAsBool

The value of variable

Define

Boolean ValueAsBool

Description

Annotation

This property is read/write

Example

Set the value of variable to true

VBScript Example

```
Var.NewVariableGroup.NewVariable1.ValueAsBool = true
```

3. SystemVariable object

SystemVariable

System variable

Property list

	Name	Description
	CurrentUserName	The current login user
	Date	The date string of current system
	Day	The number of days of the current system date
	DayOfWeek	What day of the week of the current system date
	DayOfYear	The day of the year of the current system date
	ElapsedTime	The project running time，the unit is in seconds
	HasAlarm	Whether there is a alarm in the current system
	Hour	The current system time in hour
	Millisecond	The current system time in millisecond
	Minute	The current system time in minute
	Month	The current system time in month
	Now	The date and time of current system

	ProjectDir	Current project directory
	Second	The current system time in second
	StartTime	The start time of current project
	Time	The string of current system time
	Year	The current system date

The following is the detailed description of the script:

【Property】

(1) CurrentUserName property

CurrentUserName

The current login user

Define

String CurrentUserName

Description

Annotation

This property is read-only

Example

Display the current login user

VBScript Example

```
TextBox0.Text = Sys.CurrentUserName
```

(2) Date property

Date

The date string of current system

Define

String Date

Description

Annotation

The property is read-only

Example

Display the string of current system date

VBScript Example

```
TextBox0.Text = Sys.Date
```

(3) Day property

Day

The number of days of the current system date

Define

Int32 Day

Description

Annotation

This property is read-only

Example

Display the number of days of the current system date

VBScript Example

```
TextBox0.Text = Sys.Day
```

(4) DayOfWeek property

DayOfWeek

The day of week of the current system date

Define

Int32 DayOfWeek

Description

Annotation

This property is read-only

Example

Display the day of the week of the current system date

VBScript Example

```
TextBox0.Text = Sys.DayOfWeek
```

(5) DayOfYear property

DayOfYear

The day of year of the current system date

Define

Int32 DayOfYear

Description

Annotation

This property is read-only

Example

Display the day of year of the current system date

VBScript Example

```
TextBox0.Text = Sys.DayOfYear
```

(6) ElapsedTime property

ElapsedTime

The project running time，the unit is in seconds

Define

Double ElapsedTime

Description

Annotation

This property is read-only

Example

Display the project running time，the unit is in seconds

VBScript Example

```
TextBox0.Text = Sys.ElapsedTime
```

(7) HasAlarm property

HasAlarm

Whether there is alarm in the current system

Define

Boolean HasAlarm

Description

Annotation

This property is read-only

Example

Display whether there is alarm in the current system

VBScript Example

```
TextBox0.Text = Sys.HasAlarm
```

(8) Hour property

Hour

The current system time in hour

Define

Int32 Hour

Description

Annotation

This property is read-only

Example

Display the current system time in hour

VBScript Example

```
TextBox0.Text = Sys.Hour
```

(9) Millisecond property

Millisecond

The current system time in millisecond

Define

Int32 Millisecond

Description

Annotation

This property is read-only

Example

Display the current system time in millisecond

VBScript Example

```
TextBox0.Text = Sys.Millisecond
```

(10) Minute property

Minute

The current system time in minute

Define

Int32 Minute

Description

Annotation

This property is read-only

Example

Display the current system time in minute

VBScript Example

```
TextBox0.Text = Sys.Minute
```

(11) Month property

Month

The current system time in month

Define

Int32 Month

Description

Annotation

This property is read-only

Example

Display the current system time in month

VBScript Example

```
TextBox0.Text = Sys.Month
```

(12) Now property

Now

The date and time of current system

Define

String Now

Description

Annotation

This property is read-only

Example

Display the date and time of current system

VBScript Example

```
TextBox0.Text = Sys.Now
```

(13) ProjectDir property

ProjectDir

Current project directory

Define

String ProjectDir

Description

Annotation

This property is read-only

Example

Query current project directory

VBScript Example

```
TextBox0.Text = Sys.ProjectDir
```

(14) Second property

Second

The current system time in seconds

Define

Int32 Second

Description

Annotation

This property is read-only

Example

Display the current system time in seconds

VBScript Example

```
TextBox0.Text = Sys.Second
```

(15) StartTime property

StartTime

The start time of current project

Define

DateTime StartTime

Description

Annotation

This property is read-only

Example

Display the start time of current project

VBScript Example

```
TextBox0.Text = Sys.StartTime
```

(16) Time property

Time

The string of current system time

Define

String Time

Description

Annotation

This property is read-only

Example

Display the string of current system time

VBScript Example

```
TextBox0.Text = Sys.Time
```

(17) Year property

Year

The current system date

Define

Int32 Year

Description

Annotation

This property is read-only

Example

Display the current system date

VBScript Example

```
TextBox0.Text = Sys.Year
```

4. VariableGroup object

VariableGroup

The variable group

Property list

	Name	Description
	AbsolutePath	The variable group absolute path
	Description	The description of variable group
	Name	The name of variable group
	UniqueIdentifier	The only label ID

The following is the detailed description of the script:

【Property】

(1) AbsolutePath property

AbsolutePath

The variable group absolute path

Define

String AbsolutePath

Description

Annotation

This property is read-only

Example

Display the variable group absolute path by pop-up a dialog

VBScript Example

```
TextBox0.Text = Var.NewVariableGroup1.AbsolutePath
```

(2) Description property

Description

The description of variable group

Define

String Description

Description

Annotation

This property is read/write

Example

Modify the description of variable group to "Switch variable"

VBScript Example

```
Var.NewVariableGroup1.Description = "Switch variable"
```

(3) Name property

Name

Variable group name

Define

String Name

Description

Annotation

This property is read-only

Example

Display the variable group name

VBScript Example

```
TextBox0.Text = Var.NewVariableGroup1.Name
```

(4) UniqueIdentifier property

UniqueIdentifier

The only label ID

Define

Int32 UniqueIdentifier

Description

Annotation

This property is read-only

Example

Display the only label ID of variable group

VBScript Example

```
TextBox0.Text = Var.NewVariableGroup1.UniqueIdentifier
```

5. TextVariable object

TextVariable

String variable

Methods list

	Name	Description
	ChangeValue	Change variable value

Property list

	Name	Description
	AbsolutePath	Variable absolute path
	Description	The description of variable
	InitValue	Variable initial value
	IsInitValueSaved	Whether to save the current value when the program exits
	Name	Variable name
	UniqueIdentifier	The only label ID
	Value	The value of variable,object type
	ValueAsString	The variable value

The following is the detailed description of the script:

【Method】

(1) ChangeValue method

ChangeValue

Change variable value

Define

ChangeValue(value,reason)

Parameter

Name	Required/Optional	Data Type	Description
value	Required	**String**	Target value
reason	Required	**String**	This value will be entered into the operation variable

Return

void

Example

Change variable value

VBScript Example

```
Call Var.Digital1.ChangeValue("Value","test")
```

【Property】

(1) AbsolutePath property

AbsolutePath

Variable absolute path

Define

String AbsolutePath

Description

Annotation

This property is read-only

Example

Pop-up a dialog to display the variable full path

VBScript Example

```
MsgBox Var.NewVariableGroup.NewVariable.AbsolutePath
```

(2) Description property

Description

The description of variable

Define

String Description

Description

Annotation

This property is read/write

Example

Modify the discriptions of variables to "a string variable"

VBScript Example

```
Var.NewVariableGroup.NewVariable.Description = "a string variable"
```

(3) InitValue property

InitValue

Variable initial value

Define

String InitValue

Description

Annotation

This property is read/write

Example

Modify variable initial value to "wwww"

VBScript Example

```
Var.String1.InitValue = "wwww"
```

(4) IsInitValueSaved property

IsInitValueSaved

Whether to save the current value when the program exits

Define

Boolean IsInitValueSaved

Description

Annotation

This property is read/write

Example

Set the property to true

VBScript Example

```
Var.NewVariableGroup.NewVariable.IsInitValueSaved = true
```

(5) Name property

Name

The variable name

Define

String Name

Description

Annotation

This property is read-only

Example

Display the variable name by pop-up a dialog

VBScript Example

```
MsgBox Var.NewVariableGroup.NewVariable.Name
```

(6) UniqueIdentifier property

UniqueIdentifier

The only label ID

Define

Int32 UniqueIdentifier

Description

Annotation

This property is read-only

Example

Display the only label ID by pop-up a dialog

VBScript Example

```
MsgBox Var.NewVariableGroup.NewVariable.UniqueIdentifier
```

(7) Value property

Value

The value of variable,object type

Define

Object Value

Description

Annotation

This property is read/write

Example

Set the value of variable to "pipeline"

VBScript Example

```
Var.NewVariableGroup.NewVariable.ValueAsString = "pipeline"
```

(8) ValueAsString property

ValueAsString

Variable value

Define

String ValueAsString

Description

Annotation

This property is read/write

Example

Set the initial value of variable to Pipe

VBScript Example

```
Var.NewVariableGroup.NewVariable.ValueAsString = "Pipe"
```

6. VariableCmd object

VariableCmd

Variable command

Methods list

	Name	Description
	GetExtendedDomainCount	Get number of variables which specified the domain and types
	GetExtendedDomainVars	Get the set of variables which specified the domain and types
	GetVariableValue	Get values of variables which is specified the path
	SetVariableValue	Set values of variables which is specified the path

The following is the detailed description of the script:

【Method】

(1) GetExtendedDomainCount method

GetExtendedDomainCount

Get number of variables which specified the domain and types

Define

GetExtendedDomainCount(extendedDomain,varType)

Parameter

Name	Required/Optional	Data Type	Description
extendedDomain	Required	**String**	Extended domain
varType	Required	**Int**	Variable type，0 means analog，1 means digital，2 means string

Example

Get number of string variables which domain name is "aaa"

VBScript Example

```
TextBox0.Text = VarCmd.GetExtendedDomainCount("aaa",2)
```

(2) GetExtendedDomainVars method

GetExtendedDomainVars

Get the set of variables which specified the domain and types

Define

GetExtendedDomainVars(extendedDomain,varType)

Parameter

Name	Required/Optional	Data Type	Description
extendedDomain	Required	**String**	Extended domain
varType	Required	**Int**	Variable type，0 means analog，1 means digital，2 means string

Example

Get the set of string variables which domain name is "aaa"

VBScript Example

```
List = VarCmd.GetExtendedDomainVars("aaa",2)
Count = VarCmd.GetExtendedDomainCount("aaa",2)
For i = 0 To Count - 1
    MsgBox CStr(LIST(i))
Next
```

(3) GetVariableValue method

GetVariableValue

Get values of variables which is specified the path

Define

GetVariableValue(VariablePath , errorValue)

Parameter

Name	Required/Optional	Data Type	Description
VariablePath	Required	**String**	The variable path
errorValue	Required	**Object**	Return an error value, return when the variable path was not found

Example

Get the value of variable "Var.variable0"

VBScript Example
TextBox0.Text = VarCmd.GetVariableValue("Var.Variable0",0)

(4) SetVariableValue method

SetVariableValue

Set values of variables which is specified the path

Define

SetVariableValue(VariablePath , Value)

Parameter

Name	Required/Optional	Data Type	Description
VariablePath	Required	**String**	The variable path
Value	Required	**Object**	The value to set

Example

More variable assignment，that means set the value of "Var.variable0、Var.variable1、Var.variable2" to"Var.VariableGroup0.variable0、Var.VariableGroup0.variable1、Var.VariableGroup0.variable2"

VBScript Example
For i = 0 To 2 tt = VarCmd.GetVariableValue("Var.variable"&i, 0) Call VarCmd.SetVariableValue("Var.VariableGroup0.variable"&i,tt) Next

20.3.4 Report

1. ReportCommand object

ReportCommand

The report command scripts

Methods list

	Name	Description
	DirectExportDataToExcel	Export the DataTable to Excel table
	DirectImportToDataTable	Import the content of the table to DataTable
	DirectImportToDataTable	Import the specified content of the table to the DataTable
	DirectPrintDataTable	DataTable print
	DirectPrintTemplateForPath	Print directly according to the template
	ExportDataByTemplate	Export the history data and curve control pictures to excel according to the template
	ExportDataByTemplate	Export the real-time data and curve control pictures to excel according to the template
	ExportHistoryDataByTemplate	Export the history data to excel according to the template
	ReadCsvFileByTemplate	Read CSV file
	SetWorkSheetEndTime	Set the end time of query
	SetWorkSheetIntervalTime	Set the time interval of query
	SetWorkSheetStartTime	Set the start time of query

The following is the detailed description of the script:

【Method】

(1) DirectExportDataToExcel method

DirectExportDataToExcel

Export the DataTable to Excel table

Define

DirectExportDataToExcel(data ,excelFilePath , version)

Parameter

Name	Required/Optional	Data Type	Description
data	Required	**DataTable**	Data source
excelFilePath	Required	**String**	The excel file path
version	Required	**Int**	Excel version(0：office97To2003；1：office07 and newer version)

Example

1. Get the data in the database access table and export it with the current time as the Excel name

VBScript Example

```
If month(now()) < 10 then
   Nowtime = Year(now())&"0"&month(now())&Day(now())
Else
   Nowtime = Year(now())&month(now())&Day(now())
End If
dt = DbAccess.DatabaseAccess.GetTable()
Call ReportCmd.DirectExportDataToExcel(dt,"D:\"&Nowtime&".xlsx",1)
```

2. Export historical data for the last hour in the database

VBScript Example

```
StartTime = DateAdd("d",-1,now()) 'Returns the date of the specified interval
EndTime = FormatDateTime(Sys.Now,0)
aql = "SELECT * FROM " + DbAccess.DatabaseAccess.TableName +"  where Triggertime between
'"+StartTime+"' and '"+EndTime+"'"
dt = DbAccessCmd.ExecuteDataTable("DbAccess.DatabaseAccess","sqlservercompact",sql)
Call ReportCmd.DirectExportDataToExcel(dt,"D:\test.xlsx",1)
```

(2) DirectImportToDataTable method

DirectImportToDataTable

Import the content of the table to the DataTable

Define

DirectImportToDataTable(excelFilePath)

Parameter

Name	Required/Optional	Data Type	Description
excelFilePath	Required	**String**	The file path

Example

Import file data of the specified path and display it in the report control

VBScript Example

```
dt = ReportCmd.DirectImportToDataTable("D:\datatable.xlsx")
Call Report0.ShowDataTableForReport(1,1,dt)
```

(3) DirectImportToDataTable method

DirectImportToDataTable

Import the table contents into the DataTable

Define

DirectImportToDataTable(excelFilePath ,rowHeaderIndex ,startRowIndex ,endRowIndex)

Parameter

Name	Required/Optional	Data Type	Description
excelFilePath	Required	**String**	The file path
rowHeaderIndex	Required	**Int**	Index of the row in which the header is placed,starting at 0
startRowIndex	Required	**Int**	Index of start row
endRowIndex	Required	**Int**	Index of end row

Example

Import the data from line 5 to line 20 of the file in the specified path and display it in the report control

VBScript Example

```
dt = ReportCmd.DirectImportToDataTable("D:\datatable.xlsx",0,5,20)
Call Report0.ShowDataTableForReport(1,1,dt)
```

(4) DirectPrintDataTable method

DirectPrintDataTable

DataTable print

Define

DirectPrintDataTable(data ,pageMediaSize ,topMargin ,bottomMargin ,leftMargin ,rightMargin ,lanscape ,scale)

Parameter

Name	Required/Optional	Data Type	Description
data	Required	**DataTable**	Data Source

pageMediaSize	Required	**Int**	Page setup,8 is A4
topMargin	Required	**Double**	The top margin,default value is 0
bottomMargin	Required	**Double**	The bottom margin,default value is 0
leftMargin	Required	**Double**	The left margin,default value is 0
rightMargin	Required	**Double**	The right margin,default value is 0
lanscape	Required	**Int**	The print direction,1 is lateral, 2 is in the longitudinal direction
scale	Required	**Int**	Factor,the default value is 100

Example

Custom Settings page to print the DataTable data

VBScript Example

```
dt = ReportCmd.DirectImportToDataTable("D:\datatable.xlsx")
Call ReportCmd.DirectPrintDataTable(dt,8,0,0,0,0,1,100)
```

(5) DirectPrintTemplateForPath method

DirectPrintTemplateForPath

Print directly according to the template

Define

DirectPrintTemplateForPath(Path,pageMediaSize,topMargin,bottomMargin ,leftMargin ,rightMargin,lanscape,scale)

Parameter

Name	Required/Optional	Data Type	Description
path	Required	**String**	The file path
pageMediaSize	Required	**Int**	Page setup,8 is A4
topMargin	Required	**Double**	The top margin,default value is 0
bottomMargin	Required	**Double**	The bottom margin,default value is 0
leftMargin	Required	**Double**	The left margin,default value is 0
rightMargin	Required	**Double**	The right margin, default value is 0
lanscape	Required	**Int**	The print direction,1 is lateral, 2 is in the

			longitudinal direction
scale	Required	**Int**	Factor,the default value is 100

Print directly according to the template

VBScript Example

```
Call ReportCmd.DirectPrintTemplateForPath("D:\test.xlsx",8,0,0,0,0,1,100)
```

(6) ExporDataByTempLate method

ExporDataByTempLate

Export the history data and curve control pictures to excel according to the template

Define

ExportDataByTemplate(fileName ,templateName,dt ,imageBytes)

Parameter

Name	Required/Optional	Data Type	Description
fileName	Required	**String**	The file path saved
templateName	Required	**String**	Report template name
dt	Required	**DataTable**	The export data
imageBytes	Required	**String**	The picture string，each picture are separated by“|”

Example

Export three historical curve data and images to Excel table according to the report template

VBScript Example

```
dt = HistoryChart0.GetDataTable("Series0")     'Gets the "Series0" data in HistoryChart
dt1 = HistoryChart1.GetDataTable("Series0")
dt2 = HistoryChart2.GetDataTable("Series0")
Call DbAccessCmd.SetPrimary(dt,"TriggerTime")     'Set "TriggerTime" as the primary key
Call DbAccessCmd.SetPrimary(dt1,"TriggerTime")
Call DbAccessCmd.SetPrimary(dt2,"TriggerTime")
Call dt.Merge(dt1)     'Merge dt1 data into dt
Call dt.Merge(dt2)
```

```
Call ReportCmd.ExportDataByTemplate("D:\Test.xlsx","HistoryReport0",dt,HistoryChart0.ArrayToString("Series0")&"|"&HistoryChart1.ArrayToString("Series0")&"|"&HistoryChart2.ArrayToString("Series0")
```

(7) ExportDataByTemplate method

ExportDataByTemplate

Export the real-time data and curve control pictures to excel according to the template

Define

ExportDataByTemplate(fileName ,templateName ,imageBytes)

Parameter

Name	Required/Optional	Data Type	Description
fileName	Required	**String**	The file path saved
templateName	Required	**String**	Report template name
imageBytes	Required	**String**	The picture string，each picture are separated by “|”

Example

Export two real-time curve data and images to Excel table according to the report template "RealReport0"

VBScript Example

```
Call ReportCmd.ExportDataByTemplate("D:\Test.xlsx","RealReport0",RealTimeChart0.ArrayToString("Series0")&"|"&RealTimeChart1.ArrayToString("Series0"))
```

(8) ExportHistoryDataByTemplate method

ExportHistoryDataByTemplate

Export the history data to excel according to the template

Define

ExportDataByTemplate(templateName ,excelFileTime)

Parameter

Name	Required/Optional	Data Type	Description
templateName	Required	**String**	The report template name
excelFileTime	Required	**String**	The file path

Example

Export the last hour of historical data to Excel according to report template "report 0"

VBScript Example

```
starttime = dateadd("h",-1,now())
Call ReportCmd.SetWorkSheetStartTime(0,starttime)
Call ReportCmd.SetWorkSheetIntervalTime(0,1000)
Call ReportCmd.SetWorkSheetEndTime(0,Sys.Now)
Call ReportCmd.ExportHistoryDataByTemplate("Report0","D:\123.xlsx")
```

(9) ReadCsvFileByTemplate method

ReadCsvFileByTemplate

Read CSV file

Define

ReadCsvFileByTemplate(templatePath,csvFilePath)

Parameter

Name	Required/Optional	Data Type	Description
templatePath	Required	**String**	Report template name
csvFilePath	Required	**String**	The file path

Example

The report template "Report 0" associates the real-time variables, and then executes this script to assign the values of the corresponding cells in the .csv file to the variables in the template.

VBScript Example

```
Call ReportCmd.ReadCsvFileByTemplate("Report0","D:\test.csv")
```

(10) SetWorkSheetEndTime method

SetWorkSheetEndTime

Set the end time of query

Define

SetWorkSheetEndTime(sheetIndex ,EndTime)

Parameter

Name	Required/Optional	Data Type	Description
sheetIndex	Required	**Int**	The worksheet index
EndTime	Required	**DateTime**	End time

Example

Set the current system time to end time

VBScript Example
Call ReportCmd.SetWorkSheetEndTime(0,Sys.Now)

(11) SetWorkSheetIntervalTime method

SetWorkSheetIntervalTime

Set the time interval of query

Define

SetWorkSheetIntervalTime(sheetIndex , intervalTime)

Parameter

Name	Required/Optional	Data Type	Description
sheetIndex	Required	**Int**	The worksheet index
intervalTime	Required	**Int64**	Time interval

Example

Set the time interval to 1 s

VBScript Example
Call ReportCmd.SetWorkSheetIntervalTime(0,1000)

(12) SetWorkSheetStartTime method

SetWorkSheetStartTime

Set the start time of query

Define

SetWorkSheetStartTime(sheetIndex ,StartTime)

Parameter

Name	Required/Optional	Data Type	Description
sheetIndex	Required	**Int**	The worksheet index
StartTime	Required	**DateTime**	Start time

Example

Set the current system time to start time

VBScript Example
Call ReportCmd.SetWorkSheetStartTime(0,Sys.Now)

20.3.5 Alarm

1. AlarmGroup object

AlarmGroup

Alarm group

Property list

	Name	Description
	AbsolutePath	The full path of alarm group
	AckedAlarmCount	The number of the alarm group currently in responses state
	AlarmCount	The number of the alarm group currently in alarm state
	AllAlarmCount	The number of the alarm group currently in all alarms and responses
	Description	The description of alarm group
	HasAlarm	Whether there is alarm in the current alarm group
	Name	The name of alarm group

	UniqueIdentifier	The only label ID of alarm group

The following is the detailed description of the script:

【Property】

(1) AbsolutePath property

AbsolutePath

The full path of alarm group

Define

String AbsolutePath

Description

Annotation

This property is read-only

Example

Display the absolute path of "AlarmGroup0" by text control

VBScript Example

```
Text0.Text = Alarm.AlarmGroup0.AbsolutePath
```

(2) AckedAlarmCount property

AckedAlarmCount

The number of the alarm group currently in responses state

Define

Int32 AckedAlarmCount

Description

Annotation

This property is read-only

Example

Display the number of the alarm group currently in responses state by text control

VBScript Example

```
Text0.Text = Alarm.AlarmGroup0.AckedAlarmCount
```

(3) AlarmCount property

AlarmCount

The number of the alarm group currently in alarm state

Define

Int32 AlarmCount

Description

Annotation

This property is read-only

Example

Display the number of the alarm group currently in alarm state by text control

VBScript Example

```
Text0.Text = Alarm.AlarmGroup0.AlarmCount
```

(4) AllAlarmCount property

AllAlarmCount

The number of the alarm group currently in alarms and responses

Define

Int32 AllAlarmCount

Description

Annotation

This property is read-only

Example

Display the number of the alarm group currently in alarms and responses by text control

VBScript Example

```
Text0.Text = Alarm.AlarmGroup0.AllAlarmCount
```

(5) Description property

Description

The description of alarm group

Define

String Description

Description

Annotation

This property is read/write

Example

Modify the property to "Temperature alarm group"

VBScript Example

```
Alarm.AlarmGroup0.Description = "Temperature alarm group"
```

(6) HasAlarm property

HasAlarm

Whether there is alarm in the current alarm group

Define

Boolean HasAlarm

Description

Annotation

This property is read only

Example

Display whether there is alarm in the "AlarmGroup0" by text control

VBScript Example

```
Text0.Text = Alarm.AlarmGroup0.HasAlarm
```

(7) Name property

Name

The name of alarm group

Define

String Name

Description

Annotation

This property is read-only

Example

Display the name of alarm group by text control

VBScript Example

```
Text0.Text = Alarm.AlarmGroup.Name
```

(8) UniqueIdentifier property

UniqueIdentifier

The unique identifier of alarm group

Define

Int32 UniqueIdentifier

Description

Annotation

This property is read-only

Example

Display the unique identifier of alarm group by text control

VBScript Example

```
Text0.Text = Alarm.AlarmGroup0.UniqueIdentifier
```

2. AnalogAlarm object

AnalogAlarm

Analog variable alarm model

Property list

	Name	Description
	AbsolutePath	Absolute path of alarm
	AlarmType	Alarm type
	CurrentValue	Current value of alarm variable
	DelayValue	Analog deviation or the limit of the alarm delay time, the unit is second
	Description	Description of alarm
	DeviationDeadBandValue	The analog alarm deviation value of the dead zone
	DeviationTargetValue	The analog alarm deviation value of the target
	HighContent	Analog high alarm text
	HighHighContent	The text of analog high high alarm
	HighHighValue	Analog high high alarm value
	HighValue	Analog high alarm value
	LastTriggerTime	The last time of the alarm time
	Level	The alarm level,the highest level is 999,0-199:slight，200-399：lighter，400-599：general，600-799：heavier，800-999：serious
	LimitDeadBandValue	The upper or lower limits of dead zone value of the analog alarm
	LowContent	The text of analog low alarm
	LowerDeviationContent	The alarm text of the analog deviation lower limit
	LowerDeviationValue	The alarm value of the analog deviation lower limit
	LowLowContent	The text of analog low low alarm
	LowLowValue	The value of analog low low alarm
	LowValue	The value of analog low low alarm
	Name	The alarm name
	RateOfChangeContent	The alarm text of analog rate
	RateOfChangeUnit	The unit of change rate,second is 1，minute is 60，hour is 3600
	RateOfChangeValue	The alarm value of the analog change rate
	Status	Alarm variable status, -1: no alarm; 0: all alarms are restored; 1.new alarm; 2: any alarm will be response
	UniqueIdentifier	The alarm unique identifier

	UpperDeviationContent	The alarm text of the analog deviation upper limit
	UpperDeviationValue	The alarm value of the analog deviation upper limit

The following is the detailed description of the script:

【Property】

(1) AbsolutePath property

AbsolutePath

The alarm absolute path

Define

String AbsolutePath

Description

Annotation

This property is read-only

Example

Display the absolute path of the "AlarmVariable0" in alarm group "AlarmGroup0"

VBScript Example

```
Text0.Text = Alarm.AlarmGroup0.AlarmVariable0.AbsolutePath
```

(2) AlarmType property

AlarmType

Alarm type

Define

Int32 AlarmType

Description

Annotation

This property is read/write

Example

Set the alarm type of alarm "Alarm.AlarmGroup0.AlarmVariable0" to lower alarm (2) and higher alarm (4),the value is equal to 6

VBScript Example

```
Alarm.AlarmGroup0.AlarmVariable0.AlarmType = 6
```

(3) CurrentValue property

CurrentValue

Current value of alarm variable

Define

Object CurrentValue

Description

Annotation

This property is read-only

Example

Display current value of alarm variable by text control

VBScript Example

```
Text0.Text = Alarm.AlarmGroup0.AlarmVariable.CurrentValue
```

(4) DelayValue property

DelayValue

Analog deviation or the limit of the alarm delay time, the unit is second

Define

Int32 DelayValue

Description

Annotation

This property is read/write

Example

Modify the analog deviation or the limit of the alarm delay time to 1 second

VBScript Example

```
Alarm.AlarmGroup0.AlarmVariable0.DelayValue = 1
```

(5) Description property

Description

Alarm description

Define

String Description

Description

Annotation

This property is read/write

Example

Modify the description of alarm to“AlarmDescription”

VBScript Example

```
Alarm.AlarmGroup0.AlarmVariable0.Description = "AlarmDescription"
```

(6) DeviationDeadBandValue property

DeviationDeadBandValue

The analog alarm deviation value of the dead zone

Define

Double DeviationDeadBandValue

Description

Annotation

This property is read/write

Example

Modify the analog alarm deviation value of the dead zone to 1

VBScript Example

```
Alarm.AlarmGroup0.AlarmVariable0.DeviationDeadBandValue = 1
```

(7) DeviationTargetValue property

DeviationTargetValue

The analog alarm deviation value of the target

Define

Double DeviationTargetValue

Description

Annotation

This property is read/write

Example

Modify the analog alarm deviation value of the target to 1

VBScript Example

```
Alarm.AlarmGroup0.AlarmVariable0.DeviationTargetValue = 1
```

(8) HighContent property

HighContent

Analog high alarm text

Define

String HighContent

Description

Annotation

This property is read/write

Example

Modify the analog high alarm text to“high alarm”

VBScript Example

```
Alarm.AlarmGroup0.AlarmVariable0.HighContent = "high alarm"
```

(9) HighHighContent property

HighHighContent

The text of analog high high alarm

Define

String HighHighContent

Description

Annotation

This property is read/write

Example

Modify the analog high high alarm text to “high high alarm”

VBScript Example

```
Alarm.AlarmGroup0.AlarmVariable0.HighHighContent = "high high alarm"
```

(10) HighHighValue property

HighHighValue

Analog high high alarm value

Define

Double HighHighValue

Description

Annotation

This property is read/write

Example

Modify the analog high high alarm value to 1

VBScript Example

```
Alarm.AlarmGroup0.AlarmVariable0.HighHighValue = 1
```

(11) HighValue property

HighValue

Analog high alarm value

Define

Double HighValue

Description

Annotation
This property is read/write

Example

Modify the analog high alarm value to 1

VBScript Example
Alarm.AlarmGroup0.AlarmVariable0.HighValue = 1

(12) LastTriggerTime property

LastTriggerTime

Time of last alarm trigger

Define

DateTime LastTriggerTime

Example

Display the last time of the alarm time by text control

VBScript Example
Text0.Text = Alarm.AlarmGroup0.AlarmVariable0.LastTriggerTime

(13) Level property

Level

The alarm level,the highest level is 999,0-199:slight，200-399：lighter，400-599：general，600-799：heavier，800-999：serious

Define

Int32 Level

Description

Annotation

This property is read/write

Example

Set alarm level

VBScript Example

```
Alarm.AlarmGroup0.AlarmVariable0.Level = 300
```

(14) LimitDeadBandValue property

LimitDeadBandValue

The upper or lower limits of dead zone value of the analog alarm

Define

Double LimitDeadBandValue

Description

Annotation

This property is read/write

Example

Modify the value of the upper or lower limits of the analog alarm dead zone to 1

VBScript Example

```
Alarm.AlarmGroup0.AlarmVariable0.LimitDeadBandValue = 1
```

(15) LowContent property

LowContent

The text of analog low alarm

Define

String LowContent

Description

Annotation

This property is read/write

Example

Modify the text of analog low alarm to “low alarm”

VBScript Example

```
Alarm.AlarmGroup0.AlarmVariable0.LowContent = "low alarm"
```

(16) LowerDeviationContent property

LowerDeviationContent

The alarm text of the analog deviation lower limit

Define

String LowerDeviationContent

Description

Annotation

This property is read/write

Example

Modify the alarm text of the analog deviation lower limit to “deviation lower limit alarm”

VBScript Example

```
Alarm.AlarmGroup0.AlarmVariable0.LowerDeviationContent = "deviation lower limit alarm"
```

(17) LowerDeviationValue property

LowerDeviationValue

The alarm value of the analog deviation lower limit

Define

Double LowerDeviationValue

Description

Annotation

This property is read/write

Example

Modify the alarm value of the analog deviation lower limit to 1

VBScript Example

```
Alarm.AlarmGroup0.AlarmVariable0.LowerDeviationValue = 1
```

(18) LowLowContent property

LowLowContent

The text of analog low low alarm

Define

String LowLowContent

Description

Annotation

This property is read/write

Example

Modify the text of analog low low alarm to “low low alarm”

VBScript Example

```
Alarm.AlarmGroup0.AlarmVariable0.LowLowContent = "low low alarm"
```

(19) LowLowValue property

LowLowValue

The value of analog low low alarm

Define

Double LowLowValue

Description

Annotation

This property is read/write

Example

Modify the value of analog low low alarm to 1

VBScript Example

```
Alarm.AlarmGroup0.AlarmVariable0.LowLowValue = 1
```

(20) LowValue property

LowValue

The value of analog low low alarm

Define

Double LowValue

Description

Annotation

This property is read/write

Example

Modify the value of analog low low alarm to 1

VBScript Example

```
Alarm.AlarmGroup0.AlarmVariable0.LowValue = 1
```

(21) Name property

Name

Alarm name

Define

String Name

Description

Annotation

This property is read-only

Example

Text displays the alarm name

VBScript Example

```
Text0.Text = Alarm.AlarmGroup0.AlarmVariable0.Name
```

(22) RateOfChangeContent property

RateOfChangeContent

The alarm text of analog rate

Define

String RateOfChangeContent

Description

Annotation

This property is read/write

Example

Modify the alarm text of analog rate to “analog rate alarm”

VBScript Example

```
Alarm.AlarmGroup0.AlarmVariable0.RateOfChangeContent = "analog rate alarm"
```

(23) RateOfChangeUnit property

RateOfChangeUnit

The unit of change rate,second is 1，minute is 60，hour is 3600

Define

Int32 RateOfChangeUnit

Description

Annotation

This property is read/write

Example

Modify the unit of change rate to 60

VBScript Example

```
Alarm.AlarmGroup0.AlarmVariable0.RateOfChangeUnit = 60
```

(24) RateOfChangeValue property

RateOfChangeValue

The alarm value of the analog change rate

Define

Double RateOfChangeValue

Description

Annotation

This property is read/write

Example

Modify the alarm value of the analog change rate to 1

VBScript Example

```
Alarm.NewAlarmGroup.NewAlarm1.RateOfChangeValue = 1
```

(25) Status property

Status

Alarm variable status, -1: no alarm; 0: all alarms are restored; 1.new alarm; 2: any alarm will be response

Define

Int32 Status

Description

Annotation

This property is read-only

Example

Text display alarm variable status

VBScript Example

```
Text0.Text = Alarm.AlarmGroup0.AlarmVariable0.Status
```

(26) UniqueIdentifier property

UniqueIdentifier

The alarm unique identifier

Define

Int32 UniqueIdentifier

Description

Annotation

This property is read-only

Example

Assign the alarm unique identifier to the text

VBScript Example

```
Text0.Text = Alarm.AlarmGroup0.AlarmVariable0.UniqueIdentifier
```

(27) UpperDeviationContent property

UpperDeviationContent

The alarm text of the analog deviation upper limit

Define

String UpperDeviationContent

Description

Annotation

This property is read/write

Example

Modify the content of the alarm text of the analog deviation upper limit to “deviation upper limit alarm”

VBScript Example

```
Alarm.AlarmGroup0.AlarmVariable0.UpperDeviationContent = "deviation upper limit alarm"
```

(28) UpperDeviationValue property

UpperDeviationValue

The alarm value of the analog deviation upper limit

Define

Double UpperDeviationValue

Description

Annotation
This property is read/write

Example

Modify the alarm value of the analog deviation upper limit to 1

VBScript Example
Alarm.AlarmGroup0.AlarmVariable0.UpperDeviationValue = 1

3. DigitalAlarm object

DigitalAlarm

Digital variable alarm

Property list

	Name	Description
	AbsolutePath	Absolute path of alarm
	AlarmType	Alarm type
	CurrentValue	Current value of alarm variable
	Description	Description of alarm
	Level	The alarm level,the highest level is 999,0-199:slight，200-399：lighter，400-599：general，600-799：heavier，800-999：serious
	Name	The alarm name
	OffAlarmContent	Digital close the alarm text
	OffToOnAlarmContent	Digital close to open alarm text
	OnAlarmContent	Digital open the alarm text
	OnToOffAlarmContent	Digital open to close alarm text

	Status	Alarm variable status, -1: no alarm; 0: all alarms are restored; 1.new alarm; 2: any alarm will be response
	UniqueIdentifier	The unique identifier of alarm

The following is the detailed description of the script:

【Property】

(1) AbsolutePath property

AbsolutePath

The alarm absolute path

Define

String AbsolutePath

Description

Annotation

This property is read-only

Example

Display the absolute path of the "AlarmVariable0" in alarm group "AlarmGroup0"

VBScript Example

```
Text0.Text = Alarm.AlarmGroup0.AlarmVariable0.AbsolutePath
```

(2) AlarmType property

AlarmType

Alarm type

Define

Int32 AlarmType

Description

Annotation

This property is read/write

Example

Set the alarm type of alarm "Alarm.AlarmGroup0.AlarmVariable0" to lower alarm (2) and higher alarm

(4),the value is equal to 6

VBScript Example

```
Alarm.AlarmGroup0.AlarmVariable0.AlarmType = 6
```

(3) CurrentValue property

CurrentValue

Current value of alarm variable

Define

Object CurrentValue

Description

Annotation

This property is read-only

Example

Display current value of alarm variable by text control

VBScript Example

```
Text0.Text = Alarm.AlarmGroup0.AlarmVariable.CurrentValue
```

(4) Description property

Description

Alarm description

Define

String Description

Description

Annotation

This property is read/write

Example

Modify the description of alarm to“AlarmDescription”

VBScript Example

```
Alarm.AlarmGroup0.AlarmVariable0.Description = "AlarmDescription"
```

(5) Level property

Level

The alarm level,the highest level is 999,0-199:slight，200-399：lighter，400-599：general，600-799：heavier，800-999：serious

Define

Int32 Level

Description

Annotation

This property is read/write

Example

Set alarm level

VBScript Example

```
Alarm.AlarmGroup0.AlarmVariable0.Level = 300
```

(6) Name property

Name

Alarm name

Define

String Name

Description

Annotation

This property is read-only

Example

Text displays the alarm name

VBScript Example

```
Text0.Text = Alarm.AlarmGroup0.AlarmVariable0.Name
```

(7) OffAlarmContent property

OffAlarmContent

Digital close the alarm text

Define

String OffAlarmContent

Description

Annotation

This property is read/write

Example

Modify the text of the digital close the alarm to "OffAlarm"

VBScript Example

```
Alarm.AlarmGroup0.AlarmVariable0.OffAlarmContent = "OffAlarm"
```

(8) OffToOnAlarmContent property

OffToOnAlarmContent

Digital close to open alarm text

Define

String OffToOnAlarmContent

Description

Annotation

This property is read/write

Example

Modify the text of the digital close to open alarm to "OffToOnAlarm"

VBScript Example

```
Alarm.AlarmGroup0.AlarmVariable0.OffToOnAlarmContent = "OffToOnAlarm"
```

(9) OnAlarmContent property

OnAlarmContent

Digital open the alarm text

Define

String OnAlarmContent

Description

Annotation

This property is read/write

Example

Modify the text of the digital open the alarm to "OnAlarm"

VBScript Example

```
Alarm.AlarmGroup0.AlarmVariable0.OnAlarmContent = "OnAlarm"
```

(10) OnToOffAlarmContent property

OnToOffAlarmContent

Digital open to close alarm text

Define

String OnToOffAlarmContent

Description

Annotation

This property is read/write

Example

Modify the text of the digital open to close alarm to "OnToOffAlarm"

VBScript Example

```
Alarm.0AlarmGroup0.AlarmVariable0.OnToOffAlarmContent = "OnToOffAlarm"
```

(11) Status property

Status

Alarm variable status, -1: no alarm; 0: all alarms are restored; 1.new alarm; 2: any alarm will be response

Define

Int32 Status

Description

Annotation

This property is read-only

Example

Text display alarm variable status

VBScript Example

```
Text0.Text = Alarm.AlarmGroup0.AlarmVariable0.Status
```

(12) UniqueIdentifier property

UniqueIdentifier

The alarm unique identifier

Define

Int32 UniqueIdentifier

Description

Annotation

This property is read-only

Example

Assign the alarm unique identifier to the text

VBScript Example

```
Text0.Text = Alarm.AlarmGroup0.AlarmVariable0.UniqueIdentifier
```

4. AlarmCmd object

AlarmCmd

Alarm command

Methods list

	Name	Description
	RecheckAlarm	Check whether alarm variable generates alarm
	SaveAlarmConfig	Save alarm config

Property list

	Name	Description
	AckedAlarmCount	The number of alarm in reply condition
	AlarmCount	The number of alarm in the alarm state
	AllAlarmCount	All the number alarm which includes alarm and reply

The following is the detailed description of the script:

【Method】

(1) RecheckAlarm method

RecheckAlarm

Check whether alarm variable generates alarm

Define

RecheckAlarm(*alarmPaths*)

Parameter

Name	Required/Optional	Data Type	Description
alarmPaths	Required	**String**	Format

Return

Void

Example

Check whether alarm variable generates alarm

VBScript Example

```
Call AlarmCmd.RecheckAlarm("Alarm.AlarmValue1;Alarm.AlarmValue2")
```

(2) SaveAlarmConfig method

SaveAlarmConfig

Save alarm configuration

Define

SaveAlarmConfig()

Example

Save alarm configuration

VBScript Example

```
Call AlarmCmd.SaveAlarmConfig()
```

【Property】

(1) AckedAlarmCount property

AckedAlarmCount

The number of alarm in reply condition

Define

Int32 AckedAlarmCount

Description

Annotation

This property is read-only

Example

The text shows the number of alarms in the response state

VBScript Example

```
Text0.Text = AlarmCmd.AckedAlarmCount
```

(2) AlarmCount property

AlarmCount

The number of alarm in the alarm state

Define

Int32 AlarmCount

Description

Annotation

This property is read-only

Example

The text shows the number of alarms in the alarm state

VBScript Example

```
Text0.Text = AlarmCmd.AlarmCount
```

(3) AllAlarmCount property

AllAlarmCount

All the number alarm which includes alarm and reply

Define

Int32 AllAlarmCount

Description

Annotation

This property is read-only

Example

Display the number of alarm currently in alarms and responses by text control

VBScript Example

```
Text0.Text = AlarmCmd.AllAlarmCount
```

5. AudioAlarmConfig object

AudioAlarmConfig

Alarm sound configuration

Methods list

	Name	Description
	GetUserCustomContent	Get the user custom content
	SetUserCustomContent	Set the user custom content

The following is the detailed description of the script:

【Method】

(1) GetUserCustomContent method

GetUserCustomContent

Get the user custom content

Define

GetUserCustomContent(category ,name ,isRecoveryTrigger)

Parameter

Name	Required/Optional	Data Type	Description
category	Required	**Int**	Get the content of the specified category, 0-all alarms, 1-alarm group, 2-alarm level, 3-alarm type, 4-alarm variable, other values are invalid
name	Required	**Object**	When category=0, the Name value is is invalid When category=1, the Name value is the Name of the alarm group When category=2, the Name value is: 1- very serious, 2- severe, 4- general, 8- not urgent, 16- slight When category=3, the Name value is: 1- low, 2- low, 4- high, 8- high, 16- small deviation, 32- large deviation, 64- change rate, 512- on, 024- off, 2048- switch shift When category=4, the Name value is the path of the alarm variable

isRecovery Trigger	Required	**Bool**	Triggered when alarm is restored

Example

Get the user custom content

VBScript Example
Text0.Text = AudioAlarmConfig.GetUserCustomContent(2,4,False)

(2) SetUserCustomContent method

SetUserCustomContent

Set the user custom content

Define

SetUserCustomContent(category ,name ,isRecoveryTrigger ,newValue ,isLoopSpeech)

Parameter

Name	Required/Optional	Data Type	Description
category	Required	**Int**	Get the content of the specified category, 0-all alarms, 1-alarm group, 2-alarm level, 3-alarm type, 4-alarm variable, other values are invalid
name	Required	**Object**	When category=0, the Name value is is invalid When category=1, the Name value is the Name of the alarm group When category=2, the Name value is: 1- very serious, 2- severe, 4- general, 8- not urgent, 16- slight When category=3, the Name value is: 1- low, 2- low, 4- high, 8- high, 16- small deviation, 32- large deviation, 64- change rate, 512- on, 024- off, 2048- switch shift When category=4, the Name value is the path of the alarm variable
isRecovery Trigger	Required	**Bool**	Triggered when alarm is restored

newValue	Required	**String**	User-defined voice content
isLoopSpeech	Required	**Bool**	Set whether to loop playback

Example

Set the user custom content

VBScript Example

```
Call AudioAlarmConfig.SetUserCustomContent(2,4,False,"Data exception",False)
```

20.3.6 User and authority

1. UserCmd object

UserCmd

Authority management

Methods list

	Name	Description
	AddUser	Add user
	AddUser	Add users and set the user information
	ChangePassword	Modify the password of the current login user
	ChangePasswordBox	Change password
	EditUser	Edit user's name and description
	EditUser	Edit user's name ,description,type and password
	EditUser	Edit user’s type
	GetAllUserCount	Get all user count
	GetAllUserNames	Get all the user’s name list
	GetCurrentUserName	Get current user name
	GetUserDescription	Get current user description
	GetUserNameByIndex	Get user name by index
	GetUserType	Get current user type
	Login	User login authentication
	LoginBox	User login
	Logout	User logout
	RemoveUser	Remove user

	UserManagerBox	User management

Property list

	Name	Description
	UserName	Current user name

The following is the detailed description of the script:

【Method】

(1) AddUser method

AddUser

Add user

Define

AddUser(userName ,password)

Parameter

Name	Required/Optional	Data Type	Description
userName	Required	**String**	username
password	Required	**String**	password

Return

If add success,it will return True,otherwise return False

Example

First of all, the current user has login

VBScript Example

```
Call UserCmd.AddUser("username","123")
```

(2) AddUser method

AddUser

Add users and set the user information

Define

AddUser(userName ,password ,description ,role)

Parameter

Name	Required/Optional	Data Type	Description
userName	Required	**String**	Username
password	Required	**String**	Password
description	Required	**String**	Description
role	Required	**String**	User's role

Return

If success,it will return True,otherwise return False

Example

If you want to add an administrator, the current user must be a system administrator, only one system administrator, and add the operator cannot be deleted, the current user must be an administrator

VBScript Example
Call UserCmd.AddUser("username","123","I am an operator","Operator")

(3) ChangePassword method

ChangePassword

Modify the password of the current login user

Define

ChangePassword(oldPassWord ,password)

Parameter

Name	Required/Optional	Data Type	Description
oldPassword	Required	**String**	Old password
password	Required	**String**	New password

Return

If success,it will return True,otherwise return False

Example

Modify the password,first of all,you have login

VBScript Example
Call UserCmd.ChangePassword("oldpassword","newpassword")

(4) ChangePasswordBox method

ChangePasswordBox

Change password

Define

ChangePasswordBox()

Example

Modify the password,first of all,you have login

VBScript Example
UserCmd.ChangePasswordBox()

(5) EditUser method

EditUser

Edit user's name and description

Define

EditUser(userName ,newName ,description)

Parameter

Name	Required/Optional	Data Type	Description
userName	Required	**String**	Username
newName	Required	**String**	The name to be modified
description	Required	**String**	User description

Return

If success,it will return True,otherwise return False

Example

First of all, the administrator has been login

VBScript Example

```
UserCmd.EditUser("username","newname","HasBeenModified“)
```

(6) EditUser method

EditUser

Edit user's name ,description,type and password

Define

EditUser(userName ,newName ,description , role , password)

Parameter

Name	Required/Optional	Data Type	Description
userName	Required	**String**	Username
newName	Required	**String**	Edit username
description	Required	**String**	Description
role	Required	**String**	User's type
password	Required	**String**	Password

Return

If success,it will return True,otherwise return False

Example

The administrator has been login

VBScript Example

```
UserCmd.EditUser("username","username","EditUser","Operator","111111")
```

(7) EditUser method

EditUser

Edit user’s type

Define

EditUser(userName , role)

Parameter

Name	Required/Optional	Data Type	Description
userName	Required	**String**	Username
role	Required	**String**	User's type

Return

If success,it will return True,otherwise return False

Example

Types can be modified:Admin，Operator

VBScript Example
UserCmd.EditUser("username","Admin")

(8) GetAllUserCount method

GetAllUserCount

Get all user count

Define

GetAllUserCount()

Return

Query all user count

Example

VBScript Example
TextBox0.Text = UserCmd.GetAllUserCount()

(9) GetAllUserNames method

GetAllUserNames

Get all the user’s name list

Define

GetAllUserNames()

Example

All the user's name list

VBScript Example
TextBox0.Text = UserCmd.GetAllUserNames()

(10) GetCurrentUserName method

GetCurrentUserName

Get current user name

Define

GetCurrentUserName()

Return

Return current user name

Example

VBScript Example
TextBox0.Text = UserCmd.GetCurrentUserName()

(11) GetUserDescription method

GetUserDescription

Get current user description

Define

GetUserDescription(userName)

Parameter

Name	Required/Optional	Data Type	Description
userName	Required	**String**	User name

Return

Get user description

Example

VBScript Example
TextBox0.Text = UserCmd.GetUserDescription("user")

(12) GetUserNameByIndex method

GetUserNameByIndex

Get user name by index

Define

GetUserNameByIndex(index)

Parameter

Name	Required/Optional	Data Type	Description
index	Required	**int**	User index

Return

Return user name by index

Example

VBScript Example
TextBox0.Text = UserCmd.GetUserNameByIndex(1)

(13) GetUserType method

GetUserType

Get current user type

Define

GetUserType(userName)

Parameter

Name	Required/Optional	Data Type	Description
userName	Required	**String**	Username

Example

Get current user type

VBScript Example
TextBox0.Text = UserCmd.GetUserType("user")

(14) Login method

Login

User login authentication

Define

Login(userName ,password)

Parameter

Name	Required/Optional	Data Type	Description
userName	Required	**String**	Username
password	Required	**String**	User's password

Return

If login success,it will return True,otherwise return False

Example

User login

VBScript Example
Call UserCmd.Login("username","password")

(15) LoginBox method

LoginBox

User login

Define

LoginBox()

Example

User login

VBScript Example
UserCmd.LoginBox()

(16) Logout method

Logout

User logout

Define

Logout()

Example

Logout

VBScript Example
UserCmd.Logout()

(17) RemoveUser method

RemoveUser

Remove user

Define

RemoveUser(userName)

Parameter

Name	Required/Optional	Data Type	Description
userName	Required	**String**	Username

Return

If success,it will return True,otherwise return False

Example

Delete user called "newuser",you must have administrator privileges, and youself cannot be deleted

VBScript Example
UserCmd.RemoveUser("newuser")

(18) UserManagerBox method

UserManagerBox

User management

Define

UserManagerBox()

Example

Pop-up "UserManager" window,display user information

VBScript Example

```
UserCmd.UserManagerBox()
```

【Property】

(1) UserName property

UserName

Current username

Define

String UserName

Description

Annotation

This property is read-only

Example

Query the current login username

VBScript Example

```
TextBox0.Text = UserCmd.UserName
```

20.3.7 History variable

- **Variable record(compatible)**

1. RecordVariable object

RecordVariable

Variable record

Methods list

	Name	Description
	NotifySave	Inform to save the historical data
	NotifySave	Inform to save the historical data，can be forced store

Property list

	Name	Description
	IntervalTime	Get or set the time interval of the storage

The following is the detailed description of the script:

【Method】

(1) NotifySave method

NotifySave

Inform to save the historical data

Define

NotifySave()

Example

Inform to save the historical data

VBScript Example

```
Call Record.VariableRecord0.NotifySave()
```

(2) NotifySave method

NotifySave

Inform to save the historical data

Define

NotifySave(isForcedStorage)

Parameter

Name	Required/Optional	Data Type	Description
isForcedStorage	Required	**Bool**	Whether to forced store，the shortest is trigger again per second

Example

Inform to save the historical data

VBScript Example
Call Record.VariableRecord0.NotifySave(true)

【Property】

(1) IntervalTime property

IntervalTime

Get or set the time interval of the storage

Define

Int IntervalTime

Example

Set the variable record 0 storage time interval to 1 second

VBScript Example
Record.VariableRecord0.IntervalTime = 1

- **Variable group record**

1. HistStorageGroup object

HistStorageGroup

Variable record group

Property list

	Name	Description
	Description	Discription of variable record group
	Name	Variable record group name
	TableName	The data table name of variable record group

The following is the detailed description of the script:

【Property】

(1) Description property

Description

Description of the variable record group

Define

String Description

Description

Annotation

This property is read/write

Example

Modify the discription of the variables record group "HistoryGroup0" to "test"

VBScript Example

```
HistRecord.HistoryGroup0.Description="test"
```

(2) Name property

Name

The name of the variable record group

Define

String Name

Description

Annotation

This property is read only

Example

Display the name of the variable record group "HistoryGroup0"

VBScript Example

```
TextBox0.Text = HistRecord.HistoryGroup0.Name
```

(3) TableName property

TableName

The data table name of the variable record group

Define

String TableName

Description

Annotation

This property is read/write

Example

Display the data table name of the variable record group "HistoryGroup0"

VBScript Example

```
TextBox0.Text = HistRecord.HistoryGroup0.TableName
```

2. HistStorageItem object

HistStorageItem

Variable record

Property list

	Name	Description
	Description	Description
	Expression	Expression
	Name	Name
	RealColName	Datatable mame
	StoredValueTypeStr	Value type

The following is the detailed description of the script:

【Property】

(1) Description property

Description

Description of the variable record group

Define

String Description

Description

Annotation

This property is read/write

Example

Modify the discription of the "historical record 1" in variables record group "HistoryGroup0" to "record number of products"

VBScript Example

```
HistRecord.HistoryGroup0.historical record1.Description = "record number of products"
```

(2) Expression property

Expression

Expression

Define

String Expression

Description

Annotation

This property is read only

Example

Display the expression of "variable record1"

VBScript Example

```
TextBox0.Text = HistRecord.HistoryGroup0.historical record1.Expression
```

(3) Name property

Name

Name

Define

String Name

Description

Annotation

This property is read only

Example

Display the name of "variable record1"

VBScript Example

```
TextBox0.Text = HistRecord.HistoryGroup0.historical record1.Name
```

(4) RealColName property

RealColName

The datatable's column name

Define

String RealColName

Description

Annotation

This property is read only

Example

Query the datatable's column name of "variable record1"

VBScript Example

```
TextBox0.Text = HistRecord.HistoryGroup0.historical record1.RealColName
```

(5) StoredValueTypeStr property

StoredValueTypeStr

Value type

Define

String StoredValueTypeStr

Description

Annotation
Value type，simulation value: double，switching value: boolean，text content: string

Example

Display the value type of "variable record1"

VBScript Example
TextBox0.Text = HistRecord.HistoryGroup0.HistRecord1.StoredValueTypeStr

3. HistRecordCmd object

HistRecordCmd

Polygon controls

Methods list

	Name	Description
	QueryRealTimeData	Query real-time data
	QueryWithStorageItemName	Query history group data

The following is the detailed description of the script:

【Method】

(1) QueryRealTimeData method

QueryRealTimeData

Query real time data

Define

QueryRealTimeData(histStorageGroupName)

Parameter

Name	Required/Optional	Data Type	Description

histStorageGroupName	Required	**String**	History storage group name

Example

Get real time data of the history storage group "HistoryGroup0"

Add a "Report0" to window,display the data table in "Report0"

VBScript Example

```
dt = HistRecordCmd.QueryRealTimeData("HistRecord.HistoryGroup0")
Call Report0.ClearCellContent()
Call Report0.ShowDataTableForReport(1,1,dt)
```

(2) QueryWithStorageItemName method

QueryWithStorageItemName

Query history storage group data

Define

QueryWithStorageItemName(histStorageGroupName , histStorageItemNames , startTime , endTime , rowsCount)

Parameter

Name	Required/Optional	Data Type	Description
histStorageGroupName	Required	**String**	The name of variable storage group
histStorageItemNames	Required	**String**	History storage variable name,separated with commas
startTime	Required	**DateTime**	Start time
endTime	Required	**DateTime**	End time
rowsCount	Required	**Int**	The maximum number of lines to read, 0 to read the empty table, negative to read all of them

Example

Access "historical records" of history group "HistoryGroup0" and the data of "historical records" in the recent one hour's

Use the DateAdd function to set the query time range for the past hour

Add a "Report0" to the window to display the query data

VBScript Example

```
StartTime = DateAdd("h",-1,now())
dt = HistRecordCmd.QueryWithStorageItemName("HistoryGroup0","historical record,historical record1",StartTime,Sys.Now,-1)
Call Report0.ClearCellContent()
Call Report0.ShowDataTableForReport(1,1,dt)
```

20.3.8 Background script

1. EventScript object

EventScript

Event script

Property list

	Name	Description
	Description	Description
	Expression	Expression
	Interval	Time interval，the unit is millisecond
	IntTriggerMode	Trigger mode
	IsEnable	Whether to enable the script
	Name	The name of user script
	ScriptContent	The script content
	UniqueIdentifier	The unique identifier

The following is the detailed description of the script:

【Property】

(1) Description property

Description

Description

Define

String Description

Description

Annotation

This property is read/write

Example

Display the "ConditionScript" description information in a textbox

VBScript Example

```
TextBox0.Text = Script.ConditionScript.Description
```

(2) Expression property

Expression

Expression

Define

String Expression

Description

Annotation

This property is read/write

Example

Get the expression of "ConditionScript"

VBScript Example

```
TextBox0.Text = Script.ConditionScript.Expression
```

(3) Interval property

Interval

Time interval，the unit is millisecond

Define

Int32 Interval

Description

Annotation

This property is read/write

Example

Get the interval of "ConditionScript"

VBScript Example

```
TextBox0.Text = Script.ConditionScript.Interval
```

(4) IntTriggerMode property

IntTriggerMode

Trigger mode

Define

IntTriggerMode

Description

Annotation

This property is read/write

Example

Get the triggermode of "ConditionScript"

VBScript Example

```
TextBox0.Text = Script.ConditionScript.IntTriggerMode
```

(5) IsEnable property

IsEnable

Whether to enable the script

Define

Boolean IsEnable

Description

Annotation

This property is read/write

Example

Query whether the "ConditionScript" is enabled

VBScript Example

```
TextBox0.Text = Script.ConditionScript.IsEnable
```

(6) Name property

Name

The name of user script

Define

String Name

Description

Annotation

This property is read-only

Example

Get the name of "ConditionScript"

VBScript Example

```
TextBox0.Text = Script.ConditionScript.Name
```

(7) ScriptContent property

ScriptContent

The script content

Define

String ScriptContent

Description

Annotation

This property is read/write

Example

Get the content of "ConditionScript"

VBScript Example

```
TextBox0.Text = Script.ConditionScript.ScriptContent
```

(8) UniqueIdentifier property

UniqueIdentifier

The unique identifier ID

Define

Int32 UniqueIdentifier

Description

Annotation

This property is read-only

Example

Get the unique ID of "ConditionScript"

VBScript Example

```
TextBox0.Text = Script.ConditionScript.UniqueIdentifier
```

2. TimerScript object

TimerScript

Timer script

Property list

	Name	Description
	Description	Description
	EndTime	The end time

	Interval	Time interval，the unit is millisecond
	IntTriggerMode	Trigger mode
	IsEnable	Whether to enable the script
	IsEndTime	Whether to use the end time
	MonthDays	The Months, separated by commas, such as "1,20,22,25"
	Name	The name of user script
	ScriptContent	The content of script
	StartTime	Start time
	UniqueIdentifier	The unique identifier ID
	WeekDays	The Weeks, separated by commas，such as“0,1,2,3”,0 to 6 represent from Sunday to Saturday

The following is the detailed description of the script:

【Property】

(1) Description property

Description

Description

Define

String Description

Description

Annotation

This property is read/write

Example

Display the "TimeScript" description information

VBScript Example

```
TextBox0.Text = Script.TimeScript.Description
```

(2) EndTime property

EndTime

End time

Define

DateTime EndTime

Description

Annotation
This property is read/write

Example

Query the endtime information of "TimeScript"

VBScript Example
TextBox0.Text = Script.TimeScript.EndTime

(3) Interval property

Interval

Time interval，the unit is millisecond

Define

Int32 Interval

Description

Annotation
This property is read/write

Example

Query the interval of "TimeScript"

VBScript Example
TextBox0.Text = Script.TimeScript.Interval

(4) IntTriggerMode property

IntTriggerMode

Trigger mode

Define

IntTriggerMode

Description

Annotation
This property is read/write

Example

Get the "TimeScript" triggermode

VBScript Example
TextBox0.Text = Script.TimeScript.IntTriggerMode

(5) IsEnable property

IsEnable

Whether to enable the script

Define

Boolean IsEnable

Description

Annotation
This property is read/write

Example

Query whether the "TimeScript" is enabled

VBScript Example
TextBox0.Text = Script.TimeScript.IsEnable

(6) IsEnableEndTime property

IsEnableEndTime

Whether to use the end time

Define

Boolean IsEnableEndTime

Description

Annotation

This property is read/write

Example

Query whether the "TimeScript" use endtime

VBScript Example

```
TextBox0.Text = Script.TimeScript.IsEnableEndTime
```

(7) MonthDays property

MonthDays

The Months, separated by commas, such as "1,20,22,25"

Define

String MonthDays

Description

Annotation

This property is read/write

Example

Query the months of "TimeScript"

VBScript Example

```
TextBox0.Text = Script.TimeScript.MonthDays
```

(8) Name property

Name

The name of user script

Define

String Name

Description

Annotation
This property is read-only

Example

Get the name of "TimeScript"

VBScript Example
TextBox0.Text = Script.TimeScript.Name

(9) ScriptContent property

ScriptContent

The script content

Define

String ScriptContent

Description

Annotation
This property is read/write

Example

Get the content of "TimeScript"

VBScript Example
TextBox0.Text = Script.TimeScript.ScriptContent

(10) StartTime property

StartTime

Start time

Define

DateTime StartTime

Description

Annotation
This property is read/write

Example

Get the start time of "TimeScript"

VBScript Example
TextBox0.Text = Script.TimeScript.StartTime

(11) UniqueIdentifier property

UniqueIdentifier

The unique identifier ID

Define

Int32 UniqueIdentifier

Description

Annotation
This property is read-only

Example

Get the unique ID of "TimeScript"

VBScript Example
TextBox0.Text = Script.TimeScript.UniqueIdentifier

(12) WeekDays property

WeekDays

The Weeks, separated by commas，such as“0,1,2,3”,0 to 6 represent from Sunday to Saturday

Define

String WeekDays

Description

Annotation

This property is read/write

Example

Query the weeks of "TimeScript"

VBScript Example

```
TextBox0.Text = Script.TimeScript.WeekDays
```

3. ScriptCmd object

ScriptCmd

Save user script

Methods list

	Name	Description
	Save	Save user script

The following is the detailed description of the script:

【Method】

(1) Save method

Save

Save user script

Define

Save()

Example

Save user script

VBScript Example

```
Call ScriptCmd.Save()
```

20.3.9 Recipe

1. Recipe object

Recipe

Recipe

Property list

	Name	Description
	CurrentRecipeItem	Current recipe item
	Description	Description

The following is the detailed description of the script:

【Property】

(1) CurrentRecipeItem property

CurrentRecipeItem

Current recipe item

Define

String CurrentRecipeItem

Description

Annotation

This property is read-only

Example

Get current recipe item

VBScript Example

```
TextBox0.Text = Recipe.Recipe.CurrentRecipeItem
```

(2) Description property

Description

Description

Define

String Description

Description

Annotation

This property is read/write

Example

Get description of the recipe

VBScript Example

```
Recipe.Drink.Description = "AppleDrink"
TextBox0.Text = Recipe.Drink.Description
```

2. RecipeCmd object

RecipeCmd

Recipe command

Methods list

	Name	Description
	AddRecipe	Add recipe
	AddRecipeElement	Add recipe ingredient
	AddRecipeItem	Add recipe item
	ExportRecipeToExcel	Export the recipe to Excel
	GetCurrentRecipeItem	Get current recipe item
	GetRecipeElementList	Get recipe element list
	GetRecipeItemList	Get recipe item list
	GetRecipeItemValue	Get the value of ingredient which specified by the recipe item
	GetRecipeNameList	Get recipe name list
	ImportRecipeFromExcel	Export the excel to the specified recipe,if the recipe do not exist,add a new recipe and import
	LoadRecipeItem	Load the recipe item to variable
	RemoveRecipe	Remove recipe
	RemoveRecipeElement	Remove the ingredient of recipe
	RemoveRecipeItem	Remove the recipe item
	SaveToRecipeItem	Save the variable to recipe item
	SetRecipeItemValue	Set the ingredient value which specified by the recipe item

The following is the detailed description of the script:

【Method】

(1) AddRecipe method

AddRecipe

Add recipe

Define

AddRecipe(recipeName ,description)

Parameter

Name	Required/Optional	Data Type	Description
recipeName	Required	**String**	Recipe name
description	Required	**String**	Description

Example

Add two TextBox to window,"TextBox0" and "TextBox1",record the recipe name and recipe description respectively.Add recipe,get the content of "TextBox0" and "TextBox1" as recipe name and description.

VBScript Example
Call RecipeCmd.AddRecipe("TextBox0.Text,TextBox1.Text")

(2) AddRecipeElement method

AddRecipeElement

Add recipe ingredient

Define

AddRecipeElement(recipeName ,recipeElementName ,description)

Parameter

Name	Required/Optional	Data Type	Description
recipeName	Required	**String**	Recipe name
recipeElementName	Required	**String**	Recipe ingredients name

description	Required	**String**	Description of recipe ingredients

Example

Add three TextBox to window,"TextBox0" , "TextBox1" and "TextBox2",record the recipe name,element name and recipe description respectively.Add recipe,get the content of "TextBox0" , "TextBox1" and "TextBox2" as recipe name,element name and description.

VBScript Example

```
Call RecipeCmd.AddRecipeElement(TextBox0.Text,TextBox1.Text,TextBox2.Text)
```

(3) AddRecipeItem method

AddRecipeItem

Add recipe item

Define

AddRecipeItem(recipeName ,recipeItemName ,description)

Parameter

Name	Required/Optional	Data Type	Description
recipeName	Required	**String**	Recipe name
recipeItemName	Required	**String**	Recipe item name
description	Required	**String**	Description of recipe ingredients

Example

Add three TextBox to window,"TextBox0" , "TextBox1" and "TextBox2",record the recipe name,recipe item name and recipe item description respectively.Add recipe item,get the content of "TextBox0" , "TextBox1" and "TextBox2" as recipe name,recipe item name and recipe item description.

VBScript Example

```
Call RecipeCmd.AddRecipeItem(TextBox0.Text,TextBox1.Text,TextBox2.Text)
```

(4) ExportRecipeToExcel method

ExportRecipeToExcel

Export the recipe to Excel

Define

ExportRecipeToExcel(recipeName ,filePath ,excelVersion)

Parameter

Name	Required/Optional	Data Type	Description
recipeName	Required	**String**	Recipe name
filePath	Required	**String**	The file path to export the Excel
excelVersion	Required	**String**	Excel version,value range:Excel 97 to 2003,Excel 2007,Excel2010,Excel2013

Return

If export success, it will return True

Example

Add three TextBox to window,"TextBox0" , "TextBox1" and "TextBox2",record the recipe name,export file path and excel version respectively.Export the recipe to excel,get the content of "TextBox0" , "TextBox1" and "TextBox2" as recipe name,export file path and excel version.

VBScript Example
Call RecipeCmd.ExportRecipeToExcel(TextBox0.Text,TextBox1.Text,TextBox2.Text)

(5) GetCurrentRecipeItem method

GetCurrentRecipeItem

Get current recipe item

Define

GetCurrentRecipeItem(recipeName)

Parameter

Name	Required/Optional	Data Type	Description
recipeName	Required	**String**	Recipe name

Return

The current recipe item,if does not exist,return an empty string

Example

Specify recipe name in "ComboBox0",get current recipe item.

VBScript Example
TextBox0.Text = RecipeCmd.GetCurrentRecipeItem(ComboBox0.Text)

(6) GetRecipeElementList method

GetRecipeElementList

Get recipe element list

Define

GetRecipeElementList(recipeName)

Parameter

Name	Required/Optional	Data Type	Description
recipeName	Required	**String**	Recipe name

Return

Return recipe element list

Example

Get recipe element list of "cake" and display through ComboBox

VBScript Example
tt = RecipeCmd.GetRecipeElementList("cake") ComboBox1.ClearItems() ComboBox1.AddItems(tt)

(7) GetRecipeItemList method

GetRecipeItemList

Get recipe item list

Define

GetRecipeItemList(recipeName)

Parameter

Name	Required/Optional	Data Type	Description
recipeName	Required	**String**	Recipe name

Return

Recipe item list

Example

Get recipe item list of "cake" and display through ComboBox

VBScript Example

```
tt = RecipeCmd.GetRecipeItemList("cake")
ComboBox1.ClearItems()
ComboBox1.AddItems(tt)
```

(8) GetRecipeItemValue method

GetRecipeItemValue

Get the ingredient value which specified by the recipe item

Define

GetRecipeItemValue(recipeName ,recipeItemName ,recipeElementName)

Parameter

Name	Required/Optional	Data Type	Description
recipeName	Required	**String**	Recipe name
recipeItemName	Required	**String**	Recipe item name
recipeElementName	Required	**String**	Recipe element name

Return

Return specified element values,if get fail,return null

Example

Get the value of specified element,ComboBox1,ComboBox2,ComboBox3 are used to specify recipe name,recipe item name and recipe element name.

VBScript Example

```
Call RecipeCmd.GetRecipeItemValue(ComboBox1.Text,ComboBox2.Text,ComboBox3.Text)
```

(9) GetRecipeNameList method

GetRecipeNameList

Get recipe name list

Define

GetRecipeNameList()

Return

Recipe name list

Example

Get recipe name list and display through ComboBox

VBScript Example

```
tt = RecipeCmd.GetRecipeNameList()
ComboBox1.ClearItems()
ComboBox1.AddItems(tt)
```

(10) ImportRecipeFromExcel method

ImportRecipeFromExcel

Export the excel to the specified recipe, if the recipe do not exist, add a new recipe and import

Define

ImportRecipeFromExcel(filePath ,recipeName ,excelVersion)

Parameter

Name	Required/Optional	Data Type	Description
filePath	Required	**String**	The file path of Excel
recipeName	Required	**String**	Recipe name
excelVersion	Required	**String**	Excel version, value range：Excel97to2003，Excel2007，Excel2010，Excel2013

Return

If import success, it will return True, otherwise return False

Example

Add three TextBox to window,"TextBox0" , "TextBox1" and "TextBox2",record import file path,recipe name and excel version respectively.Import recipe from excel,get the content of "TextBox0" , "TextBox1" and "TextBox2" as import file path,recipe name and excel version.

VBScript Example
Call RecipeCmd.ImportRecipeFromExcel(TextBox0.Text,TextBox1.Text,TextBox2.Text)

(11) LoadRecipeItem method

LoadRecipeItem

Load the recipe item to variable

Define

LoadRecipeItem(recipeName ,recipeItemName)

Parameter

Name	Required/Optional	Data Type	Description
recipeName	Required	**String**	Recipe name
recipeItemName	Required	**String**	Recipe items name

Example

Load recipe item to variable,"ComboBox1" and "ComboBox2" are used to spesify recipe name and recipe item name.

VBScript Example
Call RecipeCmd.LoadRecipeItem(ComboBox1.Text,ComboBox2.Text)

(12) RemoveRecipe method

RemoveRecipe

Remove the recipe

Define

RemoveRecipe(recipeName)

Parameter

Name	Required/Optional	Data Type	Description
recipeName	Required	**String**	The recipe name

Example

Delete recipe "cake"

VBScript Example
Call RecipeCmd.RemoveRecipe("cake")

(13) RemoveRecipeElement method

RemoveRecipeElement

Remove the ingredient of recipe

Define

RemoveRecipeElement(recipeName ,recipeElementName)

Parameter

Name	Required/Optional	Data Type	Description
recipeName	Required	**String**	The recipe name
recipeElementName	Required	**String**	The recipe element name

Example

Delete recipe element,"ComboBox1" and "ComboBox2" are used to spesify recipe name and recipe element name.

VBScript Example
Call RecipeCmd.RemoveRecipeElement(ComboBox1.Text,ComboBox2.Text)

(14) RemoveRecipeItem method

RemoveRecipeItem

Delete the recipe item

Define

RemoveRecipeItem(recipeName ,recipeItemName)

Parameter

Name	Required/Optional	Data Type	Description
recipeName	Required	**String**	The recipe name
recipeItemName	Required	**String**	The recipe item name

Example

Delete recipe item,"ComboBox1" and "ComboBox2" are used to spesify recipe name and recipe item name.

VBScript Example
Call RecipeCmd.RemoveRecipeItem(ComboBox1.Text,ComboBox2.Text)

(15) SaveToRecipeItem method

SaveToRecipeItem

Save the variable to recipe item

Define

SaveToRecipeItem(recipeName ,recipeItemName)

Parameter

Name	Required/Optional	Data Type	Description
recipeName	Required	**String**	The recipe name
recipeItemName	Required	**String**	The recipe item name

Example

Save to recipe item,"ComboBox1" and "ComboBox2" are used to spesify recipe name and recipe item name.

VBScript Example
Call RecipeCmd.SaveToRecipeItem(ComboBox1.Text,ComboBox2.Text)

(16) SetRecipeItemValue method

SetRecipeItemValue

Set the ingredient value which specified by the recipe item

Define

SetRecipeItemValue(recipeName ,recipeItemName ,recipeElementName ,value)

Parameter

Name	Required/Optional	Data Type	Description
recipeName	Required	**String**	Recipe name
recipeItemName	Required	**String**	Recipe items name
recipeElementName	Required	**String**	Recipe ingredients name
value	Required	**Object**	Setted value

Return

If set success, it will return True, otherwise return False

Example

Set the value of recipe element,"ComboBox1","ComboBox2","ComboBox3","TextBox1" are used to specify recipe name,recipe item name,recipe element name,value.

VBScript Example
Call RecipeCmd.SetRecipeItemValue(ComboBox1.Text,ComboBox2.Text,ComboBox3.Text,TextBox1.Text)

3. RecipeElement object

RecipeElement

Recipe ingredient

Property list

	Name	Description
	Description	Description
	Name	The name of recipe ingredient
	VariablePath	The absolute path of recipe ingredients associated variables

The following is the detailed description of the script:

【Property】

(1) Description property

Description

Description

Define

String Description

Description

Annotation

This property is read/write

Example

Modify the description of the recipe ingredient "RecipeElement1" to "NewRecipeElement"

VBScript Example

```
Recipe.Recipe.RecipeElement1.Description = "NewRecipeElement"
```

(2) Name property

Name

The name of new recipe ingredient

Define

String Name

Description

Annotation

This property is read-only

Example

Get the name of recipe element "RecipeElement1"

VBScript Example

```
TextBox0.Text = Recipe.Recipe.RecipeElement1.Name
```

(3) VariablePath property

VariablePath

The absolute path of recipe ingredients associated variables

Define

String VariablePath

Description

Annotation

This property is read-write

Example

Modify the recipe element of "RecipeElement4" associated variables to "Var.NewVariableGroup.NewVariable"

VBScript Example

```
Recipe.Recipe.RecipeElement4.VariablePath = Var.NewVariableGroup.NewVariable
```

4. RecipeItem object

RecipeItem

The recipe ingredients value group

Property list

	Name	Description
	Description	Description
	Name	Recipe items name

The following is the detailed description of the script:

【Property】

(1) Description property

Description

Description

Define

String Description

Description

Annotation

This property is read-write

Example

Modify the description of the recipe item "RecipeItem2" to "NewRecipeItem"

VBScript Example

```
Recipe.Recipe.RecipeItem2.Description = "NewRecipeItem"
```

(2) Name property

Name

Recipe items name

Define

String Name

Description

Annotation

This property is read-only

Example

Get the name of recipe item "RecipeItem2"

VBScript Example

```
TextBox0.Text = Recipe.Recipe.RecipeItem2.Name
```

20.3.10 Database access

1. DbAccess object

DbAccess

Database access

Methods list

	Name	Description

CheckTableIsExisted	Check whether there is a specified name in the database tables
ExecuteDataTable	Execute sql statement and returns the query table
ExecuteNonQuery	Execute sql statement and return the affected rows
ExecuteScalar	Execute sql statement and returns the query table, return the first line in the first column
GetDataSourceRowsCount	Access the number of rows in the table of data sources
GetEmptyTable	Get empty tables
GetPageNum	Get the number of pages
GetTable	Get the table（If the paging, access the table of the last page）
GetTable	Get the table according to the number of pages
SaveTable	Save the table
SaveTable	Save the table（Save the specified column）
TestConnection	Test the database connection

Property list

	Name	Description
	ConnectionString	Database connection string
	Description	Description
	IsPaging	Whether to paging
	MaxPagingCount	Display the maximum rows of the table
	Name	Name of the database access
	Provider	Show the Provider of database connection, that is, the database type
	TableName	The name of the data table
	UniqueIdentifier	The unique identification ID

The following is the detailed description of the script:

【Method】

(1) CheckTableIsExisted method

CheckTableIsExisted

Check whether there is a specified table in the database

Define

CheckTableIsExisted(tableName)

Parameter

Name	Required/Optional	Data Type	Description
tableName	Required	**String**	Name of the database access

Example

Check whether there is a database table named "newtable"

VBScript Example

```
TextBox0.Text = DbAccess.DatabaseAccess.CheckTableIsExisted("newtable")
```

(2) ExecuteDataTable method

ExecuteDataTable

Execute sql statement and returns the query table

Define

ExecuteDataTable(sql)

Parameter

Name	Required/Optional	Data Type	Description
sql	Required	**String**	sql statement

Example

Executing SQL statements to get the data of "DatabaseAccess" and "DatabaseAccess1" with ID 1，returns the query table and display in the Report0 control

VBScript Example

```
dim sql
sql = "SELECT * FROM
" + DbAccess.DatabaseAccess.TableName +","+DbAccess.DatabaseAccess1.TableName + " where
"+   DbAccess.DatabaseAccess.TableName +".Id = 1 and
" +  DbAccess.DatabaseAccess.TableName + ".Id =
" + DbAccess.DatabaseAccess1.TableName + ".Id"
dt = DbAccess.DatabaseAccess.ExecuteDataTable(sql)
Call Report0.ShowDataTableForReport(1,1,dt)
```

(3) ExecuteNonQuery method

ExecuteNonQuery

Execute sql statement and return the affected rows

Define

ExecuteNonQuery(sql)

Parameter

Name	Required/Optional	Data Type	Description
sql	Required	**String**	sql statement

Example

Execute sql statement to modify the name of "DatabaseAccess",return the affected rows

VBScript Example

```
dim sql
sql = "UPDATE " ＋ DbAccess.DatabaseAccess.TableName ＋" SET [name]='titi' where Id = 1"
TextBox0.Text = DbAccess.DatabaseAccess.ExecuteNonQuery(sql)
```

(4) ExecuteScalar method

ExecuteScalar

Execute sql statement , return the first line in the first column of table

Define

ExecuteScalar(sql)

Parameter

Name	Required/Optional	Data Type	Description
sql	Required	**String**	sql statement

Example

Execute sql statement to query the data of the top 50 lines in "DatabaseAccess",return the first line in the first column

VBScript Example

```
dim sql
sql = "Select Top 50 [name],[age] From " ＋ DbAccess.DatabaseAccess.TableName
TextBox0.Text = DbAccess.DatabaseAccess.ExecuteScalar(sql)
```

(5) GetDataSourceRowsCount method

GetDataSourceRowsCount

Get the number of rows in the table of data sources

Define

GetDataSourceRowsCount()

Example

Get the number of rows in the table of data sources

VBScript Example

```
TextBox0.Text = DbAccess.DatabaseAccess.GetDataSourceRowsCount()
```

(6) GetEmptyTable method

GetEmptyTable

Get empty tables

Define

GetEmptyTable()

Example

Get the table headers of "DatabaseAccess",and display it in the report control

VBScript Example

```
dt = DbAccess.DatabaseAccess.GetEmptyTable()
Call Report0.ShowDataTableForReport(1,1,dt)
```

(7) GetPageNum method

GetPageNum

Get the number of pages

Define

GetPageNum()

Example

Get the number of pages

VBScript Example
Button0.Content = DbAccess.DatabaseAccess.GetPageNum()

(8) GetTable method

GetTable

Get the table（If the paging, access the table of the last page）

Define

GetTable()

Example

Get the table

VBScript Example
dt = DbAccess.DatabaseAccess.GetTable() Call Report0.ShowDataTableForReport(1,1,dt)

(9) GetTable method

GetTable

Get the table according to the number of pages

Define

GetTable(pageNum)

Parameter

Name	Required/Optional	Data Type	Description
pageNum	Required	**Int**	The number of pages

Example

Read the data of the second page

VBScript Example

```
dt = DbAccess.DatabaseAccess.GetTable(2)
Call Report0.ShowDataTableForReport(1,1,dt)
```

(10) SaveTable method

SaveTable

Save the table

Define

SaveTable(dt)

Parameter

Name	Required/Optional	Data Type	Description
dt	Required	**DataTable**	The data source

Example

Save the table

VBScript Example

```
dt = DbAccess.DatabaseAccess.GetTable()
row = dt.NewRow()
row("name") = "Lucy"
row("age") = 21
Call dt.Rows.Add(row)
DbAccess.DatabaseAccess.SaveTable(dt)
```

(11) SaveTable method

SaveTable

Save the table

Define

SaveTable(dt , columnStr)

Parameter

Name	Required/Optional	Data Type	Description
dt	Required	**DataTable**	The data source
columnStr	Required	**String**	Select a few columns to save, separated by commas, must include the primary key

Example

Save the table

VBScript Example

```
dt = Report0.GetReportDataTable(0)
dt.TableName = DbAccess.DatabaseAccess.TableName
Text0.Text = DbAccess.DatabaseAccess.SaveTable(dt,"Id,name")'Update the specified column and return the number of rows that have been updated
```

(12) TestConnection method

TestConnection

Test the database connection

Define

TestConnection()

Example

Test the database connection，True is connected, false is not connected

VBScript Example

```
Text0.Text = DbAccess.DatabaseAccess.TestConnection()
```

【Property】

(1) ConnectionString property

ConnectionString

Database connection string

Define

String ConnectionString

Description

Annotation

This property is read-only

Example

Display the string of the "DatabaseAccess"

VBScript Example

```
Text0.Text = DbAccess.DatabaseAccess.ConnectionString
```

(2) Description property

Description

Description

Define

String Description

Description

Annotation

This property is read/write

Example

Modify the description of the "DatabaseAccess"

VBScript Example

```
DbAccess.DatabaseAccess.Description = "Custom database tables"
```

(3) IsPaging property

IsPaging

Whether to paging

Define

String IsPaging

Description

Annotation

This property is read/write

Example

Modify the "DatabaseAccess" to paging

VBScript Example

```
DbAccess.DatabaseAccess.IsPaging = True
```

(4) MaxPagingCount property

MaxPagingCount

Display the maximum rows of the table

Define

String MaxPagingCount

Description

Annotation

This property is read/write，the default value is 50

Example

Modify the maximum rows of the "DatabaseAccess"

VBScript Example

```
DbAccess.DatabaseAccess.MaxPagingCount = 20
```

(5) Name property

Name

Name of the database access

Define

String Name

Description

Annotation

This property is read-only

Example

Display the name of “DatabaseAccess”

VBScript Example

```
Text0.Text = DbAccess.DatabaseAccess.Name
```

(6) Provider property

Provider

Show the Provider of database connection, that is, the database type

Define

String Provider

Description

Annotation

This property is read-only

Example

Display the database type of "DatabaseAccess"

VBScript Example

```
Text0.Text = DbAccess.DatabaseAccess.Provider
```

(7) TableName property

TableName

The name of the data table

Define

String TableName

Description

Annotation

This property is read-only

Example

Pop-up a dialog to display the data table name of the "DatabaseAccess"

VBScript Example

```
MsgBox DbAccess.DatabaseAccess.TableName
```

(8) UniqueIdentifier property

UniqueIdentifier

The unique identification ID

Define

String UniqueIdentifier

Description

Annotation

This property is read-only

Example

Display the ID of the "DatabaseAccess"

VBScript Example

```
Text0.Text = DbAccess.DatabaseAccess.UniqueIdentifier
```

2. DbAccessCmd object

DbAccessCmd

Database access command

Methods list

	Name	Description
	AddColumn	Add column
	CreateTable	Create table
	ExecuteBatchInsertData	Insert to the data table
	ExecuteCheckTableIsExisted	Check whether there is a specified name of the database table in the

		database
	ExecuteCreateDatabaseTable	Created the same structure tables in the database table according to the table provided
	ExecuteCreateSave	Return the number of affected rows
	ExecuteDataTable	Execute commands,return the queried data table
	ExecuteDropDatabaseTable	Delete the specified data from the database
	ExecuteGetTable	Read the first "count" lines of data
	ExecuteNonQuery	Execute commands,return the number of affected rows
	ExecuteNonQueryDT	Extract data from the DataTable to perform operations according to the parameters (@ as markup)
	ExecuteSave	Select column to save, return the number of affected rows
	ExecuteScalar	Execute commands,return the number of affected rows
	SelectDT	Select according to the specified order
	SetPrimary	Set the primary key

The following is the detailed description of the script:

【Method】

(1) AddColumn method

AddColumn

Add columns

Define

AddColumn(dt , name , caption , dataType , maxLength , isAllowNull , isAutoIncrement , isUnique)

Parameter

Name	Required/Optional	Data Type	Description
dt	Required	**DataTable**	Data source
name	Required	**String**	The field name
caption	Required	**String**	The title
dataType	Required	**Int**	Data types (0: switch type, 1: integer, 4 bytes, 2: real Numbers, 3: date/time, 4: text)
maxLength	Required	**Int**	The maximum length of the field
isAllowNull	Required	**Bool**	Whether to allow null

isAutoIncrement	Required	**Bool**	Whether to growth by self
isUnique	Required	**Bool**	Whether is the unique

Example

Create a database and add new columns,then pop-up to show whether create success

VBScript Example

```
dim dt
dt = DbAccessCmd.CreateTable("newtable")       'Create the database table" newtable"
Call DbAccessCmd.AddColumn(dt,"Id","id",1,500,false,true,true)       'Add columns
Call DbAccessCmd.AddColumn(dt,"name",name",4,500,false,false,false)
Call DbAccessCmd.AddColumn(dt,"age","age",1,500,false,false,false)
Call DbAccessCmd.SetPrimary(dt,"Id")       'Set primary key
cc =
DbAccessCmd.ExecuteCreateDatabaseTable("DbAccess.DatabaseAccess","sqlservercompact",dt)
MsgBox cc       'whether created successfully
```

(2) CreateTable method

CreateTable

Create table

Define

CreateTable(tableName)

Parameter

Name	Required/Optional	Data Type	Description
tableName	Required	**String**	The table name

Example

Create a database,then pop-up to show whether create success

VBScript Example

```
dim dt
dt = DbAccessCmd.CreateTable("newtable")       'Create the database table" newtable"
Call DbAccessCmd.AddColumn(dt,"Id","id",1,500,false,true,true)       'Add columns
```

```
Call DbAccessCmd.AddColumn(dt,"name",name",4,500,false,false,false)
Call DbAccessCmd.AddColumn(dt,"age","age",1,500,false,false,false)
Call DbAccessCmd.SetPrimary(dt,"Id")        'Set primary key
cc =
DbAccessCmd.ExecuteCreateDatabaseTable("DbAccess.DatabaseAccess","sqlservercompact",dt)
MsgBox cc          'whether created successfully
```

(3) ExecuteBatchInsertData method

ExecuteBatchInsertData

Insert to the data table

Define

ExecuteBatchInsertData(connInfo , providerName , dt)

Parameter

Name	Required/Optional	Data Type	Description
connInfo	Required	**String**	Database access table
providerName	Required	**String**	Database type（sqlserver，sqlservercompact，oracle）
dt	Required	**DataTable**	The data source

Example

Insert to the data table

VBScript Example

```
dim dt dt = DbAccess.DatabaseAccess.GetTable()
dt(0)("name") = "mary"
dt(0)("age") = 18
Call DbAccessCmd.ExecuteBatchInsertData("DbAccess.DatabaseAccess","sqlservercompact",dt)
Call Report0.ShowDataTableForReport(1,1,dt)
```

(4) ExecuteCheckTableIsExisted method

ExecuteCheckTableIsExisted

Check whether there is a specified name of the database table in the database

Define

ExecuteCheckTableIsExisted(connInfo , providerName , tableName)

Parameter

Name	Required/Optional	Data Type	Description
connInfo	Required	**String**	Database Access Table
providerName	Required	**String**	Database type（sqlserver，sqlservercompact，oracle）
tableName	Required	**String**	Table name

Example

Check whether there is a database table called "newtable" in the database

VBScript Example
Text0.Text = DbAccessCmd.ExecuteCheckTableIsExisted("DbAccess.DatabaseAccess","sqlservercompact","newtable")

(5) ExecuteCreateDatabaseTable method

ExecuteCreateDatabaseTable

Create the same structure tables in the database table according to the table provided

Define

ExecuteCreateDatabaseTable(connInfo , providerName , dt , otherParameter)

Parameter

Name	Required/Optional	Data Type	Description
connInfo	Required	**String**	Database Access Table
providerName	Required	**String**	Database type（sqlserver，sqlservercompact，oracle）
dt	Required	**DataTable**	Data source
otherParameter	Required	**String**	Mainly used for the Oracle database, on behalf of Oracle table space（Is empty, select the default table space）

Example

Create the same structure tables in the database table as the table called "newtable",and return whether success

VBScript Example

```
dim dt
dt = DbAccessCmd.CreateTable("newtable")       'Create the database table" newtable"
Call DbAccessCmd.AddColumn(dt,"Id","id",1,500,false,true,true)       'Add columns
Call DbAccessCmd.AddColumn(dt,"name",name",4,500,false,false,false)
Call DbAccessCmd.AddColumn(dt,"age","age",1,500,false,false,false)
Call DbAccessCmd.SetPrimary(dt,"Id")       'Set primary key
cc =
DbAccessCmd.ExecuteCreateDatabaseTable("DbAccess.DatabaseAccess","sqlservercompact",dt)
MsgBox cc        'whether created successfully
```

(6) ExecuteCreateSave method

ExecuteCreateSave

Return the number of affected rows（If there is not exist in the database,create the same structure tables in the database table according to the table provided,save it）

Define

ExecuteCreateSave(connInfo , providerName , dt)

Parameter

Name	Required/Optional	Data Type	Description
connInfo	Required	**String**	Database Access Table
providerName	Required	**String**	Database type（sqlserver，sqlservercompact，oracle）
dt	Required	**DataTable**	Data source

Example

Get data of the report controls,create the same structure tables in the database,return the number of affected rows

VBScript Example

```
dt = Report0.GetReportDataTable(Id)
```

```
dt.tableName = DbAccess.DatabaseAccess.TableName
tt = DbAccessCmd.ExecuteCreateSave("DbAccess.DatabaseAccess","sqlservercompact",dt)
MsgBox tt
```

(7) ExecuteDataTable method

ExecuteDataTable

Execute commands, return the queried data table

Define

ExecuteDataTable(connInfo , providerName , sql)

Parameter

Name	Required/Optional	Data Type	Description
connInfo	Required	**String**	Database Access Table
providerName	Required	**String**	Database type（sqlserver，sqlservercompact，oracle）
sql	Required	**String**	sql statements

Example

SQL statement to query the same ID 1 data in the "DatabaseAccess" and " DatabaseAccess1 ", and returns, shown in the report controls

VBScript Example

```
dim sql
sql = "SELECT * FROM " ＋ DbAccess.DatabaseAccess.TableName ＋","＋
DbAccess.DatabaseAccess1.TableName ＋ " where "＋ DbAccess.DatabaseAccess.TableName ＋
".Id = 1 and " ＋ DbAccess.DatabaseAccess.TableName ＋ ".Id =
" ＋ DbAccess.DatabaseAccess1.TableName ＋ ".Id"
dt = DbAccessCmd.ExecuteDataTable("DbAccess.DatabaseAccess","sqlservercompact",sql)
Call Report0.ShowDataTableForReport(1,1,dt)
```

(8) ExecuteDropDatabaseTable method

ExecuteDropDatabaseTable

Delete the specified data from the database

Define

ExecuteDropDatabaseTable(connInfo , providerName , tableName)

Parameter

Name	Required/Optional	Data Type	Description
connInfo	Required	**String**	Database Access Table
providerName	Required	**String**	Database type（sqlserver，sqlservercompact，oracle）
tableName	Required	**String**	Table name

Example

Delete the corresponding database table in“DatabaseAccess”

VBScript Example
Text0.Text = DbAccessCmd.ExecuteDropDatabaseTable("DbAccess.DatabaseAccess","sqlservercompact",DbAccess.DatabaseAccess.TableName)

(9) ExecuteGetTable method

ExecuteGetTable

Read the first "count" rows of data

Define

ExecuteGetTable(connInfo , providerName , tableName , count)

Parameter

Name	Required/Optional	Data Type	Description
connInfo	Required	**String**	Database Access Table
providerName	Required	**String**	Database type（sqlserver，sqlservercompact，oracle）
tableName	Required	**String**	Table name
count	Required	**String**	The number of rows（If negative, read the latest count the data, if is zero, read the empty table）

Example

Read the first 3 rows of data in the database,and shown in the report controls

VBScript Example

```
dt = DbAccessCmd.ExecuteGetTable("DbAccess.DatabaseAccess","sqlservercompact","newtable",3)
Call Report0.ShowDataTableForReport(1,1,dt)
```

(10) ExecuteNonQuery method

ExecuteNonQuery

Execute commands, return the number of affected rows

Define

ExecuteNonQuery(connInfo , providerName , sql)

Parameter

Name	Required/Optional	Data Type	Description
connInfo	Required	**String**	Database Access Table
providerName	Required	**String**	Database type（sqlserver，sqlservercompact，oracle）
sql	Required	**String**	sql statements

Example

SQL statements update the ID 1 name to "titi" in the "DatabaseAccess",return and shown it in the report controls

VBScript Example

```
dim sql
sql = "UPDATE " ＋ DbAccess.DatabaseAccess.TableName ＋" SET [name]='titi' where Id = 1"
Call DbAccessCmd.ExecuteNonQuery("DbAccess.DatabaseAccess","sqlservercompact",sql)
dt = DbAccess.DatabaseAccess.GetTable()
Call Report0.ShowDataTableForReport(1,1,dt)
```

(11) ExecuteNonQueryDT method

ExecuteNonQueryDT

Extract data from the DataTable to perform operations　according to the parameters (@ as markup),for example：update table set Name=@Name,Age=@Age where Id=@Id

Define

ExecuteNonQueryDT(connInfo , providerName , sql)

Parameter

Name	Required/Optional	Data Type	Description
connInfo	Required	**String**	Database Access Table
providerName	Required	**String**	Database type（sqlserver，sqlservercompact，oracle）
sql	Required	**String**	sql statements

Example

SQL statements are asynchronous operations

VBScript Example

```
dt = Report0.GetReportDataTable(0)
sql = "update NewTable set Name=@Name,Age=@Age where Id=@Id"
Call DbAccessCmd.ExecuteNonQueryDT("DbAccess.DatabaseAccess","sqlservercompact",dt,sql)
dt1 = DbAccess.DatabaseAccess.GetTable("NewTable")
Call Report0.ShowDataTableForReport(1,1,dt1)
```

(12) ExecuteSave method

ExecuteSave

Select column to save, return the number of affected rows

Define

ExecuteSave(connInfo , providerName , dt , columnNames)

Parameter

Name	Required/Optional	Data Type	Description
connInfo	Required	**String**	Database Access Table
providerName	Required	**String**	Database type（sqlserver，sqlservercompact，oracle）
dt	Required	**DataTable**	Data source
columnNames	Required	**String**	The name of columns（separated by

			comma, and must include the primary key）

Example

Get the data of report controls,and return the number of affected rows

VBScript Example

```
dt = Report0.GetReportDataTable(Id)
dt.tableName = DbAccess.DatabaseAccess.TableName
tt = DbAccessCmd.ExecuteSave("DbAccess.DatabaseAccess","sqlservercompact",dt,"Id,name,age")
MsgBox tt
```

(13) ExecuteSave method

ExecuteScalar

Execute commands, return the number of affected rows

Define

ExecuteScalar(connInfo , providerName , sql)

Parameter

Name	Required/Optional	Data Type	Description
connInfo	Required	**String**	Database Access Table
providerName	Required	**String**	Database type（sqlserver，sqlservercompact，oracle）
sql	Required	**String**	sql statements

Example

Execute sql statement to query the data of the top 50 lines in "DatabaseAccess",return and show the first line in the first column in the report controls

VBScript Example

```
dim sql
sql = "Select Top 50 [name],[age] From " ＋ DbAccess.DatabaseAccess.TableName
tt = DbAccessCmd.ExecuteScalar("DbAccess.DatabaseAccess","sqlservercompact",sql)
Call Report0.SetCellValue(tt,1,1)
```

(14) SelectDT method

SelectDT

Select according to the specified order

Define

SelectDT(dt , filter , sort)

Parameter

Name	Required/Optional	Data Type	Description
dt	Required	**DataTable**	Data source
filter	Required	**String**	Filter condition（for example：id<10）
sort	Required	**String**	Sorting conditions（for example：desc is descending, the default is ascending）

Example

Obtain the data in "DatabaseAccess", and show in report controls after sorted according to the requirements

VBScript Example

```
dt = DbAccess.DatabaseAccess.GetTable()
cc = DbAccessCmd.SelectDT(dt,"Id<=10","Id desc")
Call Report0.ShowDataTableForReport(1,1,cc)
```

(15) SetPrimary method

SetPrimary

Set the primary key

Define

SetPrimary(dt , key)

Parameter

Name	Required/Optional	Data Type	Description
dt	Required	**DataTable**	Data source
key	Required	**String**	The primary key（If multiple, separate by comma）

Example

Create a new database and set the primary key,pop-up to display whether success

VBScript Example

```
dim dt
dt = DbAccessCmd.CreateTable("newtable")       'Create the database table" newtable"
Call DbAccessCmd.AddColumn(dt,"Id","id",1,500,false,true,true)       'Add columns
Call DbAccessCmd.AddColumn(dt,"name",name",4,500,false,false,false)
Call DbAccessCmd.AddColumn(dt,"age","age",1,500,false,false,false)
Call DbAccessCmd.SetPrimary(dt,"Id")     'Set primary key
cc =
DbAccessCmd.ExecuteCreateDatabaseTable("DbAccess.DatabaseAccess","sqlservercompact",dt)
MsgBox cc       'whether created successfully
```

20.3.11 Color

1. Colors object

Colors

Color commands

Methods list

	Name	Description
	ColorSelectionBox	Color selector
	ImageColor	Image brush
	LinearGradientColor	Linear color
	LinearGradientColor	Linear color
	LinearGradientColor	Linear color
	RadialGradientColor	Radial color
	RadialGradientColor	Radial color
	SolidColor	Solid color

The following is the detailed description of the script:

【Method】

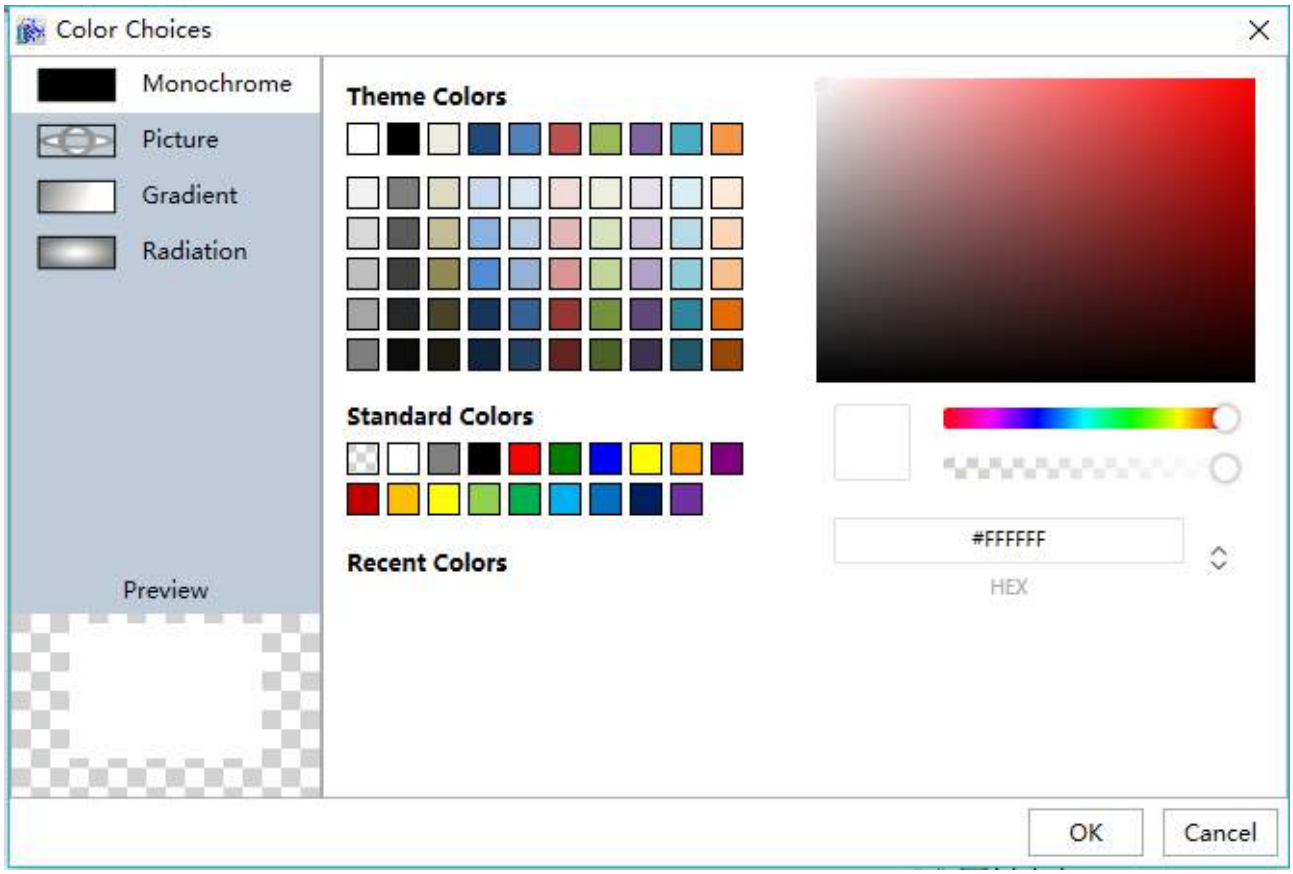
(1) ColorSelectionBox method

ColorSelectionBox

Color selector

Description

In the tree directory on the right side of the script editor, click the "ColorSelectionBox" script node, and the "Color Choices" window will appear, as shown in the figure below:



(2) ImageColor method

ImageColor

Image brush

Define

ImageColor(source)

Parameter

Name	Required/Optional	Data type	Description
source	Required	**String**	Image source path

Example

Rectangle0 is filled with a picture, and the image source path is "\Image\Penguins.jpg".

VBScript Example
Rectangle0.Fill = Colors.ImageColor("\Image\Penguins.jpg")

User can also use the script "ColorSelectionBox" to call out the“Color Choices” to achieve Image filling.

(3) LinearGradientColor method

LinearGradientColor

Linear color

Define

LinearGradientColor(colorstring)

Parameter

Name	Required/Optional	Data type	Description
colorstring	Required	**String**	A string concatendated by a combination of groups of color and offset values

Example

Rectangle0 is filled with linear color.

VBScript Example
Rectangle0.Fill = Colors.LinearGradientColor("#FFFF8080,0;#FFC1FFFF,0.5;#FFFF8080,1;")

User can also use the script "ColorSelectionBox" to call out the“Color Choices” to achieve linear color filling.

(4) LinearGradientColor method

LinearGradientColor

Linear color

Define

LinearGradientColor(startcolor , endcolor , angle)

Parameter

Name	Required/Optional	Data type	Description
startcolor	Required	**Brush**	Start color
endcolor	Required	**Brush**	End color
angle	Required	**Double**	Gradient angle

Example

Rectangle0 is filled with linear color.

VBScript Example
Rectangle0.Fill = Colors.LinearGradientColor(Colors.Violet,Colors.LightGreen,45)

User can also use the script "ColorSelectionBox" to call out the“Color Choices” to achieve linear color filling.

(5) LinearGradientColor method

LinearGradientColor

Linear color

Define

LinearGradientColor(startcolorstring , endcolorstring , angle)

Parameter

Name	Required/Optional	Data type	Description
startcolorstring	Required	**String**	Start color string
endcolorstring	Required	**String**	End color string
angle	Required	**Double**	Gradient angle

Example

Rectangle0 is filled with linear color.

VBScript Example
Rectangle0.Fill = Colors.LinearGradientColor("#FFFF80FF","#FF80FFFF",45)

User can also use the script "ColorSelectionBox" to call out the“Color Choices” to achieve linear color filling.

(6) RadialGradientColor method

RadialGradientColor

Radial color

Define

RadialGradientColor(colorstring)

Parameter

Name	Required/Optional	Data type	Description
colorstring	Required	**String**	A string concatendated by a combination of groups of color and offset values

Example

Rectangle0 is filled with radial color.

VBScript Example
Rectangle0.Fill = Colors.RadialGradientColor("#FFFFFFFF,0;#FFFFFFFF,0;#FF000000,1;")

User can also use the script "ColorSelectionBox" to call out the“Color Choices” to achieve radial color filling.

(7) RadialGradientColor method

RadialGradientColor

Radial color

Define

RadialGradientColor(startcolorstring , endcolorstring)

Parameter

Name	Required/Optional	Data type	Description
startcolorstring	Required	**String**	Start color string

endcolorstring	Required	**String**	End color string

Example

Rectangle0 is filled with radial color.

VBScript Example
Rectangle0.Fill = Colors.RadialGradientColor("#FFFF0000","#FF00FF00")

User can also use the script "ColorSelectionBox" to call out the“Color Choices” to achieve radial color filling.

(8) SolidColor method

SolidColor

Solid color

Define

SolidColor(colorstring)

Parameter

Name	Required/Optional	Data type	Description
colorstring	Required	**String**	Color string

Example

Rectangle0 is filled with solid color.

VBScript Example
Rectangle0.Fill = Colors.SolidColor("#FFFF0000")

User can also use the script "ColorSelectionBox" to call out the“Color Choices” to achieve solid color filling.

20.3.12 Global

1. LanguageCmd object

LanguageCmd

LanguageCmd

Methods list

	Name	Description
	GetCurrentLanguageName	Get the name of the current language
	GetLanguageNames	Get the names of all configured languages
	GetValueByResourceName	Get the value of the resource by the resource name
	SwitchLanguageTo	Switch the current language to the specified language

The following is the detailed description of the script:

【Method】

(1) GetCurrentLanguageName method

GetCurrentLanguageName

Get the name of the current language

Define

GetCurrentLanguageName()

Example

Get the name of the current language

VBScript Example

```
Label0.Text = Languagecmd.GetCurrentLanguageName()
```

(2) GetLanguageNames method

GetLanguageNames

Get the names of all configured languages

Define

GetLanguageNames()

Example

Get the names of all configured languages

VBScript Example

```
Label1.Text = LanguageCmd.GetLanguageNames()
```

(3) GetValueByResourceName method

GetValueByResourceName

Get the value of the resource by the resource name

Define

GetValueByResourceName(resname , lang)

Parameter

Name	Required/Optional	Data type	Description
resname	Required	**String**	Resource name
lang	Optional	**String**	Specify the system language name(empty string automatically uses the current language)

Example

Query “String1” repository to display the corresponding text in the current language

VBScript Example

```
Label1.Text = GetValueByResourceName("String1","")
```

(4) SwitchLanguageTo method

SwitchLanguageTo

Switch the current language to the specified language

Define

SwitchLanguageTo(language)

Parameter

Name	Required/Optional	Data type	Description
language	Required	**String**	Specify the language name

Example

Switch the current language to English

VBScript Example

```
Call SwitchLanguageTo("en")
```

21. DIAView Runtime Environment

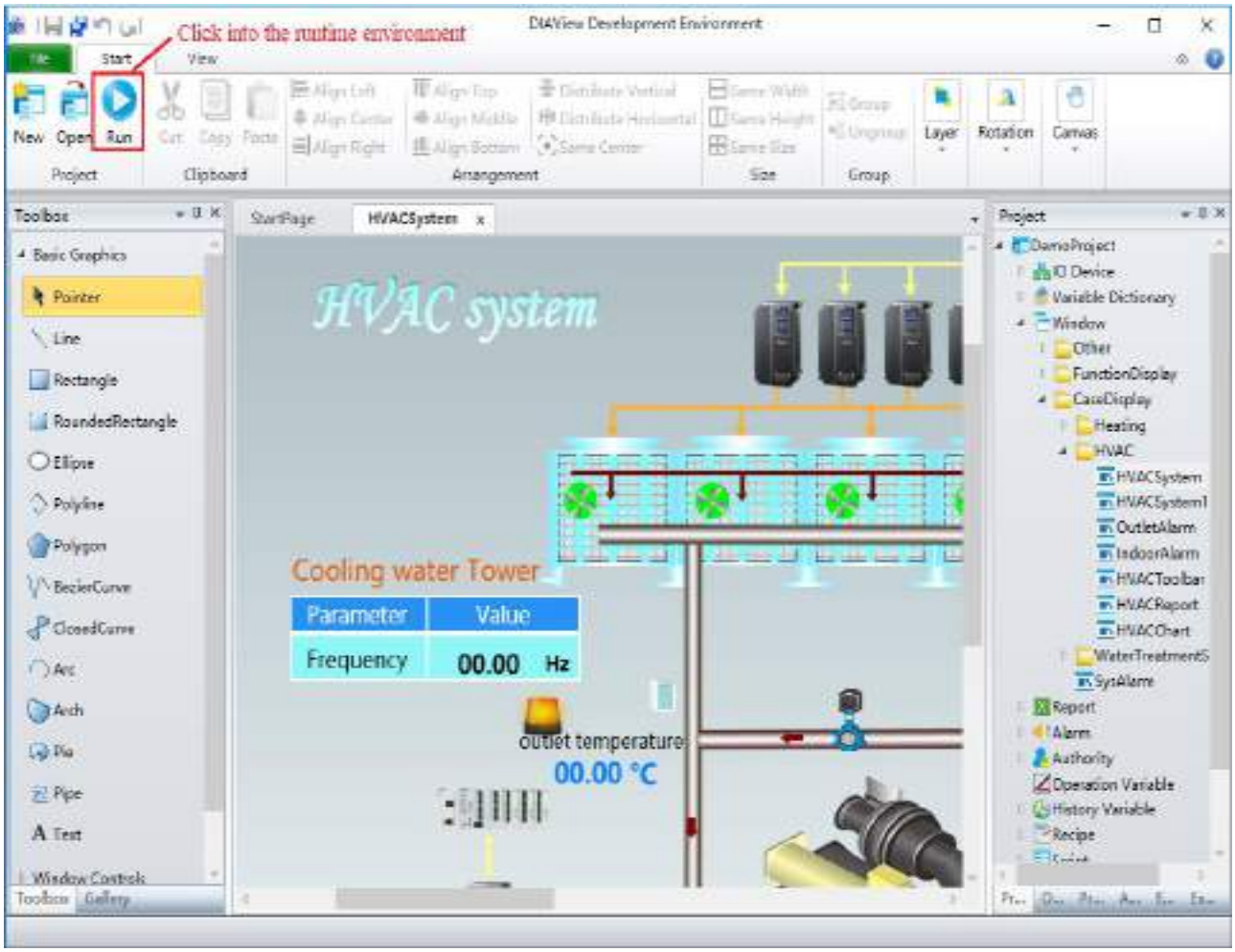
21.1 Overview

The DIAView is composed of two parts,the development environment and runtime environment.The project IO communications created in the development environment,the graphic interfaces drawn and scripts edited etc,can only br fully operated in the runtime environment,including display pictures dynamically and achieve real-time monitoring.The runtime environment of the DIAView software can dynamically display the graphic objects,the animations and events configured for graphic objects or window controls,achieving information interaction and real-time controlling between the picture and the field equipment.

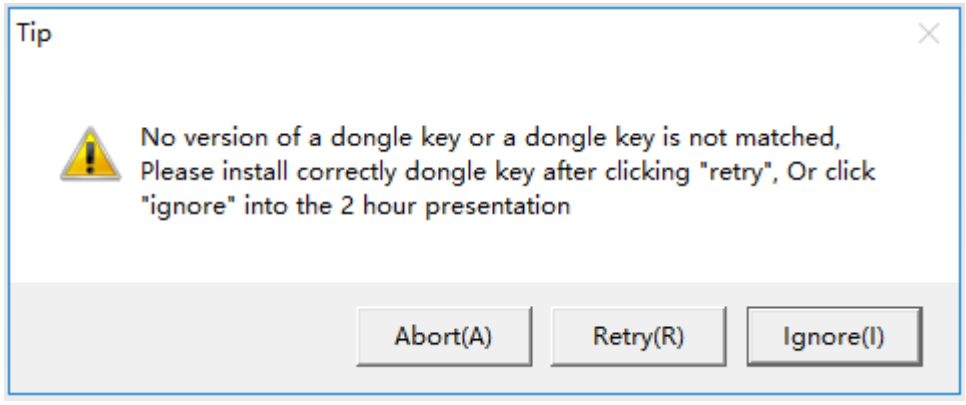
21.2 Introduction to the runtime environment

To execute a project,the project configuration must be performed in advance(refer to chapter 18);once the runtime picture configuration is finished,the runtime environment can be enabled.Steps are as follows:

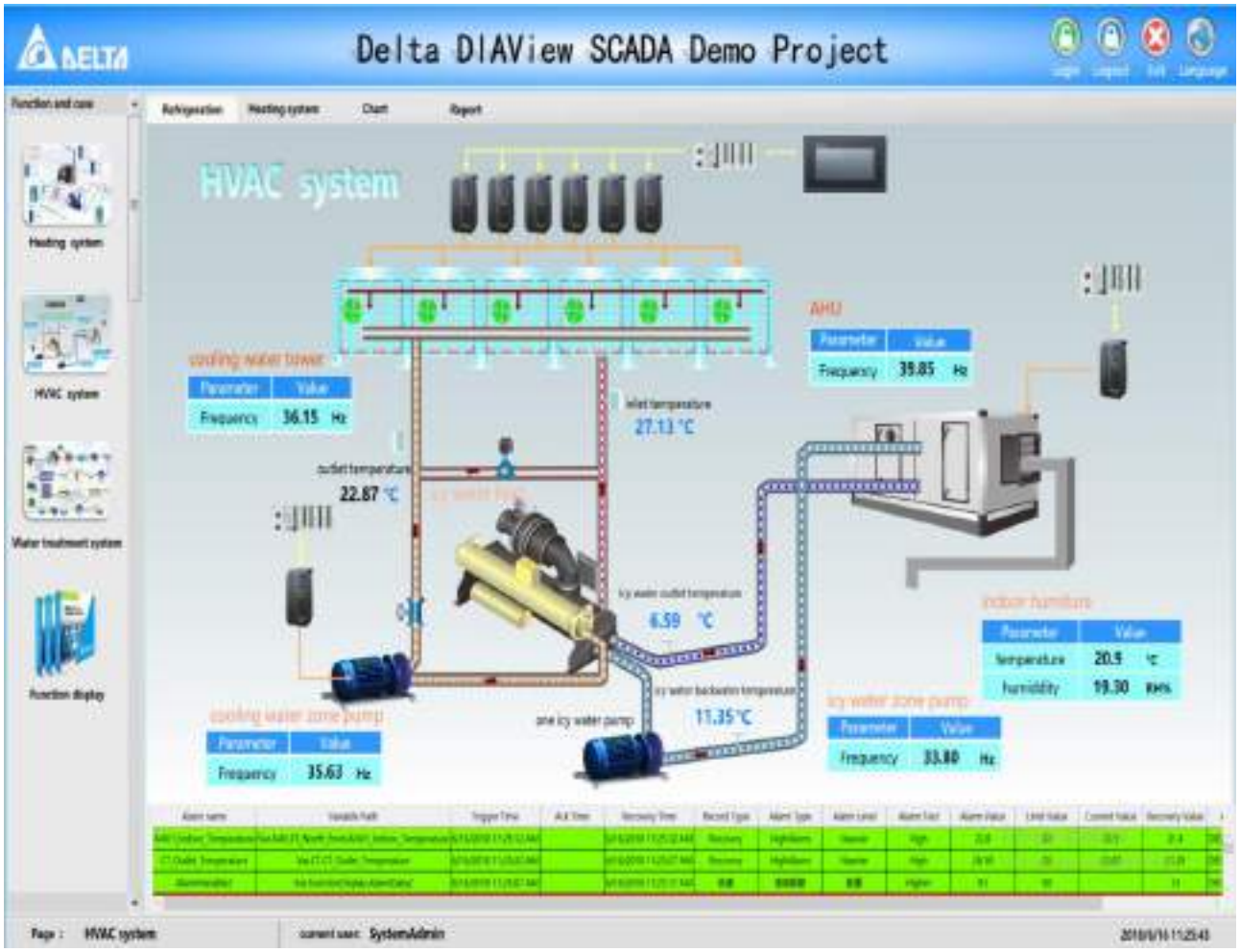
➢ Click the main interface of the DIAView software development environment → “Start” menu →“Run” button to enter the runtime environment, as shown in the figure below:



➢ System prompts that a dongle is not detected, the DIAView software runtime environment requires authorization to use:



➢ Click “Ignore” to enter the 2 hour demos (this reminder will not appear when a dongle was purchased and used):



※ The F11 button can be used to switch to full screen mode.

Includes two menus “File” and “View”:

1.File menu items and functions:

Exit: Exits the runtime environment.

It can also be achieved by calling the window command “HMICmd.ExitApplication()” to exit.

2.View menu items and functions:

Full screen: Full screen will display in the runtime environment.

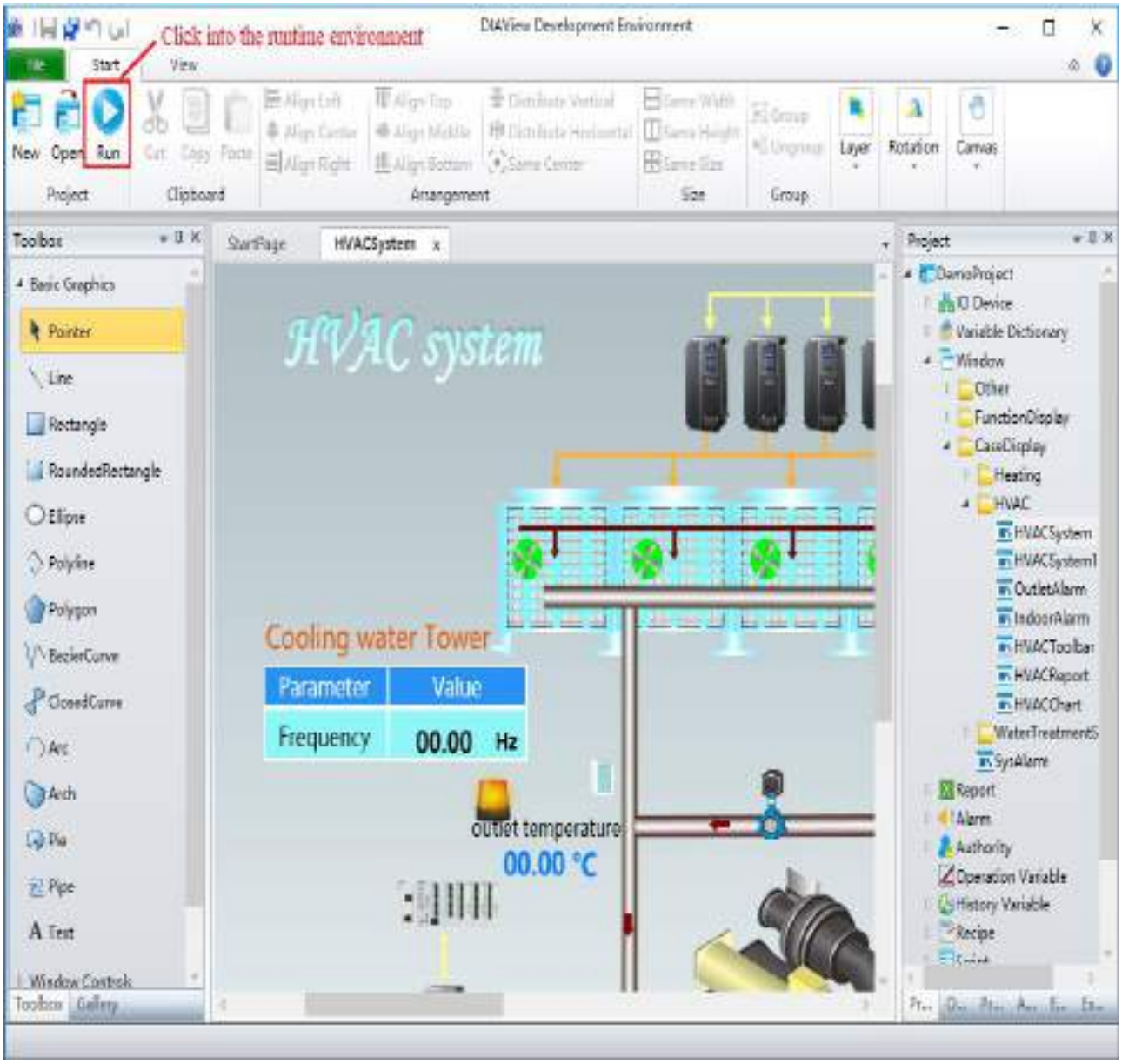
3.Language menu items and functions:

If multiple languages are configured, the language menu is displayed in the runtime environment. Click this button to switch the runtime language.

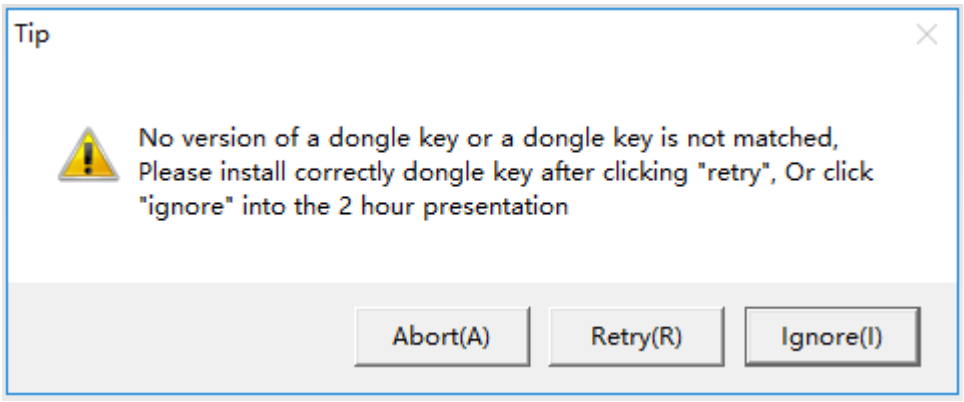
21.2 Introduction to the runtime environment

To execute a project,the project configuration must be performed in advance(refer to chapter 18);once the runtime picture configuration is finished,the runtime environment can be enabled.Steps are as follows:

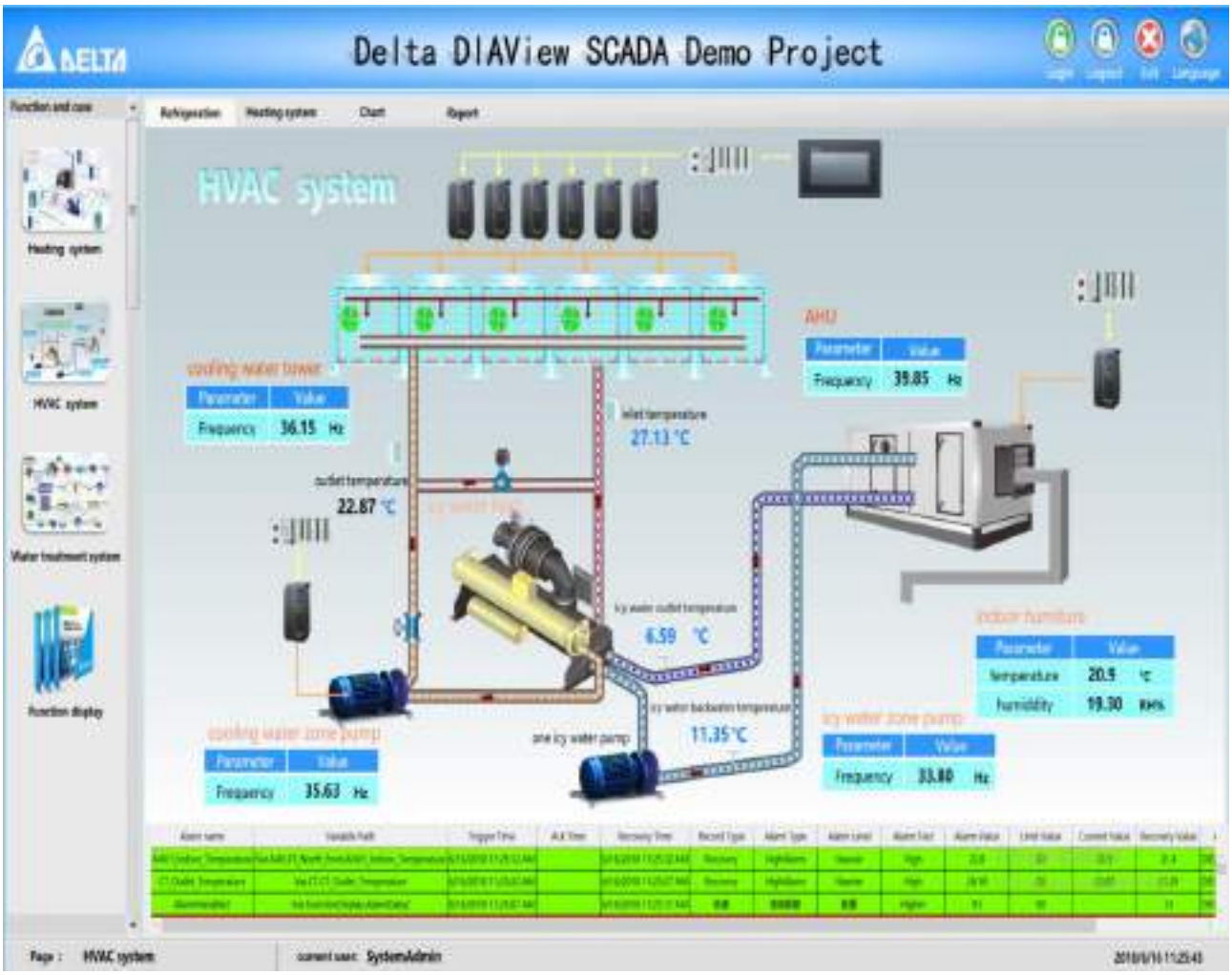
➢ Click the main interface of the DIAView software development environment → “Start” menu →“Run” button to enter the runtime environment, as shown in the figure below:



➢ System prompts that a dongle is not detected, the DIAView software runtime environment requires authorization to use:



➢ Click “Ignore” to enter the 2 hour demos (this reminder will not appear when a dongle was purchased and used):



□ The F11 button can be used to switch to full screen mode.

Includes two menus “File” and “View”:

1.File menu items and functions:

Exit: Exits the runtime environment.

It can also be achieved by calling the window command “HMICmd.ExitApplication()” to exit.

2.View menu items and functions:

Full screen: Full screen will display in the runtime environment.

3.Language menu items and functions:

If multiple languages are configured, the language menu is displayed in the runtime environment. Click this button to switch the runtime language.