

OMC System Software
High-performanceHMI
Script Editor
User Manual
IM41S66-E

Table of Contents

Script Editor	1
Section 1 Software Overview	1
1.1 Function Feature	1
Section 2 Interface Introduction	2
2.1 Title Bar	2
2.2 Menu Bar	2
2.2.1 File	3
2.2.2 Edit	4
2.2.3 Debug	5
2.2.4 Settings	6
2.2.5 View	6
2.2.6 Help	6
2.3 System Toolbar	6
2.4 Event Wizard Toolbar	7
2.5 Script Information	7
2.6 Edit Toolbar	8
2.7 Output Information Window	8
2.8 Add Function and Event Wizard	9
Section 3 Instruction to Object Properties	11
3.1 Window Objects	11
3.2 Graphics Objects	13
3.3 Global Variable(App)	14
Section 4 VFScript Application Example	18
4.1 Event Function	18
4.1.1 OnLButtonUp	18
4.1.2 OnLButtonDown	18
4.1.3 OnLButtonDbIClk	18
4.1.4 OnRButtonClk	19
4.1.5 OnMouseMove	19
4.1.6 OnMouseEnter	19
4.1.7 OnMouseLeave	20
4.1.8 OnOpenPic	20
4.1.9 OnClosePic	20
4.1.10 OnTimer	21

4.1.11 Instruction to Other Sentences of Graphics Script.....	21
4.2 Notes	22
Section 5 Appendix	23
5.1 Color constant	23
5.2 Background style	23
5.3 Edge Style	24
5.4 Gradient Style.....	25
Section 6 Revision.....	26

Script Editor

Section 1 Software Overview

VFScript uses VBScript as programming language. It is used to edit graphic script.

VBScript is a subset of Visual Basic, and easy to learn which only remains a few key words from Visual Basic, thus greatly simplifies the grammar of Visual Basic. VBScript can be mastered in short time even for those who have no experience in it.

In addition, VBScript has the performance of high security, easy to transplant and so on. VFScript is a unique part of VFDraw, realizing much more functions by applying flexible script to supervision software.

All the functions of VFScript will be available only after VFDraw has started. The postfix of script files is .vbs, and one graphics corresponds to one .vbs file with the same name.

1.1 Function Feature

- Use standard VBScript language and extended interface of graphics operation is added.
- Be built in VFDraw and usually used in VFDraw interface.
- Support grammar lightening and function online prompt.
- Support event guide.
- Support VBScript syntax check.

Section 2 Interface Introduction

Users can start up VFScript by selecting the VFDraw menu **Tool/ Edit the Script** or by right-clicking at the blank of graphics edit interface and selecting “Edit the Script” in the right-click menu. As shown in Figure 2-1, its interface includes Title Bar, Menu Bar, Script Edit Area, Script Information Area, Status Bar and so on.

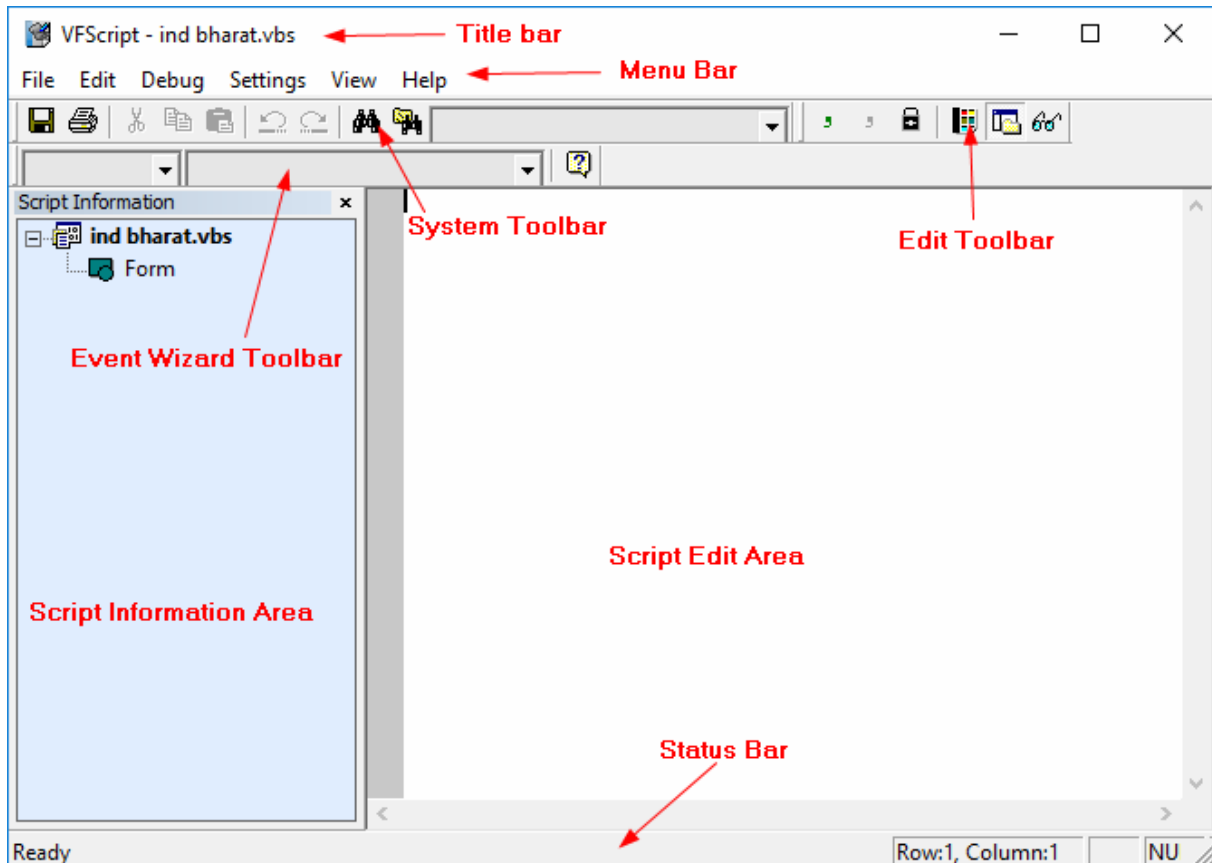


Figure 2-1 VFScript interface

2.1 Title Bar

Title bar displays the name of current file, and if there is a “*” behind the file name, it means that the script file hasn’t been saved after modifying.

There are three operation icons of “maximize”, “minimize” and “close” for users to set the size of the window.


2.2 Menu Bar

There are six menu items in the menu bar. They are File, Edit, Debug, Setting, View and Help. The shortcut letter of menu item is displayed in the bracket of right. The menu item and its submenu

can be opened by pressing “ALT+ letter” simultaneously.

2.2.1 File

File menu includes “Save”, “Save As”, “Print”, “Exit” and so on.

1. **Save:** save the finished script program to hard disk, and has the same function as the icon  in system toolbar.
2. **Save As:** if users don't want the original script file to be covered by current one, the menu command **File/ Save As** can be selected. It will save the file in another name by the pop-up dialog box of “Save As”.



Tips:

Graphics will only load .vbs script files which have the same file name, so after saving the script file as another file name, it may be unable to run. Therefore, please make sure that the script file and the graphics file have the same name but different postfixes.

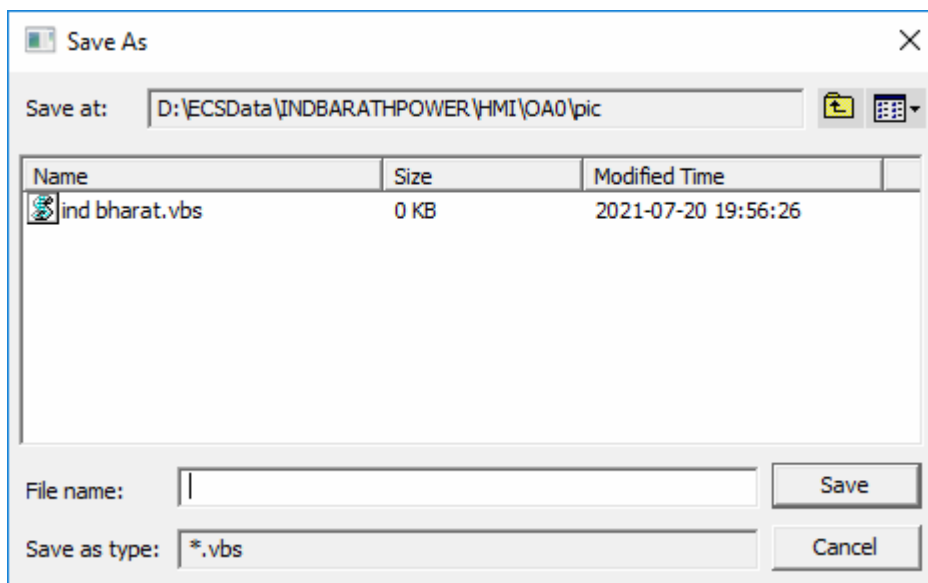



Figure 2-2 Dialog box “Save As”

3. **Print:** print the script file and has the same function as the print icon  in the toolbar.



Tip:

Different print drive may have different print interface.









4. **Recent file paths:** VFScript file menu can keep at most 4 paths of recently opened

files, and by clicking any file path users can directly open the script edit file.

5. **Exit:** exit VFScript

2.2.2 Edit

Edit menu includes Undo, Redo, Cut, Copy, Paste, Delete, Select All, Find, Find Next, Replace, Lock Text and so on.

1. **Undo:** it has the same function as  in system toolbar. Support users to recover the previous operation when editing the graphics script.
2. **Redo:** it has the same function as  in system toolbar. Support users to cancel the previous Undo operation when editing the graphics script.
3. **Cut:** it has the same function as  in system toolbar. It is used to copy the content of selected area to clipboard, and delete the content of this area at the same time. In detail, first select the script to be cut, and then click the icon  or **Edit/Cut in** the menu bar.
4. **Copy:** it has the same function as  in system toolbar. It is used to copy the content of selected area to clipboard, and its functions and usage are the same as Cut except that the content of selected area won't be deleted.
5. **Paste:** it has the same function as  in system toolbar. It is used to copy the latest content (the content been newly cut or copied) from the clipboard to designated script edit area.
6. **Delete:** it is used to delete the content of designated area in edit area. In detail, first select the script to be deleted, and then click the command to finish the operation.
7. **Select All:** select all the content in script area.
8. **Find:** it has the same function as  in system toolbar. It is used to find the designated string in edit area. Click **Edit/ Find** or the icon , pop up the Find dialog box as shown in Figure 2-3. "Match Whole Word" means the result and the content are the same string. "Match Case" means that the result and the content are the same string, and case sensitive. "Upward" means find string upward the cursor. "Downward" means find string downward the cursor.

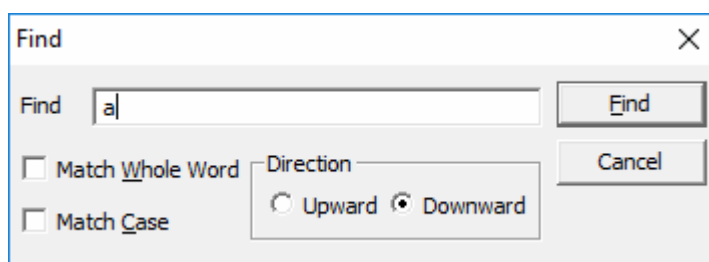



Figure 2-3 Dialog box "Find"

9. **Find Next:** it is used to find the next matched string by clicking **Edit/ Find Next** or shortcut key F3 after finding out a matched string in edit area, .
10. **Replace:** it has the same function as  in system toolbar, which is used to batch replace certain string in the script file. The functions of "Match Whole Word", "Match Case" and "Find Next" are the same as those of "Find"; "Replace" means replace the string currently found; "Replace All" means replace all the strings found; "Replace Range" is selected between "Selected Part" and "Whole Text" when the content of the script is selected, otherwise, "Whole Text" is the only choice; "Cancel" means exit "Replace" function.

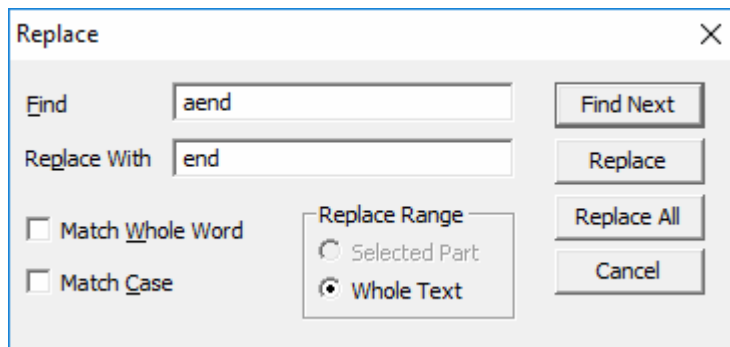


Figure 2-4 Dialog box "Replace"

11. **Lock Text:** it is used to lock the script in the edit area, making the edit operation unavailable, and select the item again the function will be canceled. Lock Text can help users avoid improper modifications that result from false operations.



Tip:

Copy/ Paste can be used among different script files, and users can paste for many times after Copy.

2.2.3 Debug

Debug menu includes "Syntax Check" and "VBScript Engine Check".

1. **Syntax check:** it is used to check the syntax of current script program, and the result is displayed in information bar. If any syntax errors are found, users can skip to the location of wrong script by double-clicking the error prompt.
2. **VBScript Engine Check:** script engine needs the support of Windows, this function can be used to make sure whether the VBScript engine in Windows is installed correctly; if there's any problem, reinstall the corresponding module of Windows or reinstall an operating system of correct version should be executed.

2.2.4 Settings

Settings menu includes “Edit Environment Options” and “Always On the Top”.

1. Edit Environment Options: click **Settings/ Edit Environment Options**, pop up the dialog box shown in Figure 2-5, it is used to set the color of background, text, keywords highlight, annotation and number in script edit area, and preview function is also provided. Select "Default Color" and system will restore the color scheme to initial status.

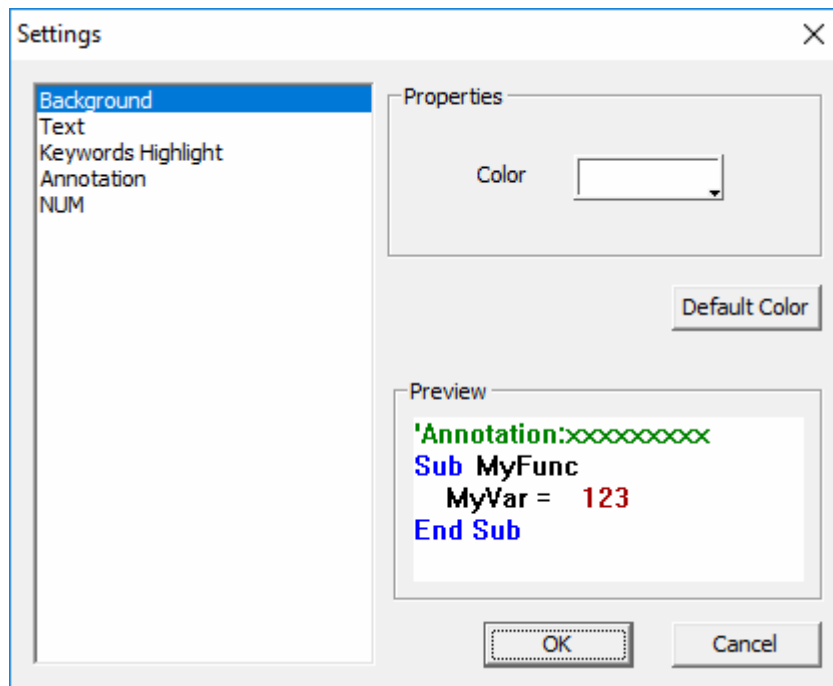


Figure 2-5 Dialog box “Edit Environment Options”

2. **Always On The Top**: click **Settings/ Always On The Top**, and current script edit interface will be on the top, which means that other interfaces can’t be opened as current active interface.

2.2.5 View

View menu includes System Toolbar, Event Wizard Toolbar, Edit Toolbar, Status Bar, Script Information, Output Bar and so on. It can show or hide every toolbar. In detail, select the name of the toolbar in View menu, and the selected toolbar can be shown in edit interface; select it again, then hide the toolbar.

2.2.6 Help

Help menu provides relevant information of the copyright and access of VFScript software.

2.3 System Toolbar

As shown in figure 2-6, system toolbar conveniently provides users with some common menu

items in tool icons.

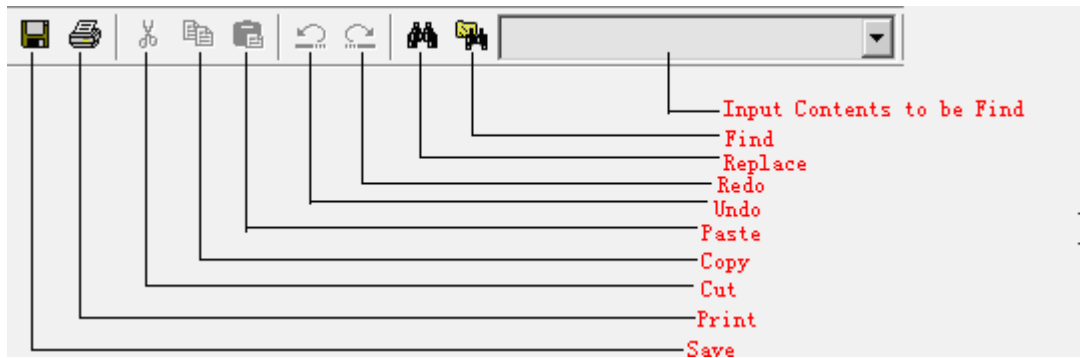


Figure 2-6 System Toolbar


2.4 Event Wizard Toolbar

Event Wizard Toolbar conveniently lists the events to be used in the toolbar pane with the form of pull-down menu.

Event Wizard Toolbar includes “Object” and “Event”. “Object” contains English names of all the objects in current graphics; “Event” displays the events supported by every object. For the built-in figure objects of graphics, there are following common events:

- OnLButtonUp(nFlag,x,y) (click the mouse button, trigger when releasing);
- OnLButtonDown(nFlag,x,y) (click the mouse button, trigger when pressing);
- OnLButtonDbIClIK(nFlag,x,y) (trigger when double-clicking the mouse button);
- OnRButtonClIK(nFLag,x,y) (trigger when right-clicking the mouse button);
- OnMouseMove(nFlag,x,y) (trigger when moving the mouse inside the figure);
- OnMouseEnter() (trigger when moving the mouse from outside the figure to inside);
- OnMouseLeave() (trigger when moving the mouse from inside the figure to outside);
- OnTimer() (the period event of timer triggering)

2.5 Script Information

As shown in Figure 2-7, all the objects in graphics are displayed in Script Information bar in tree form. Double-click any object name in Script Information bar, and the object name will be automatically imported to the “Object” on Event Wizard Toolbar. Click the icon  and select event trigger type in “Event”, then the trigger type will be written in edit area in the form of original script text shown as follow:

```
Sub Button5_OnLButtonUp(nFlag, x, y)
```

End Sub

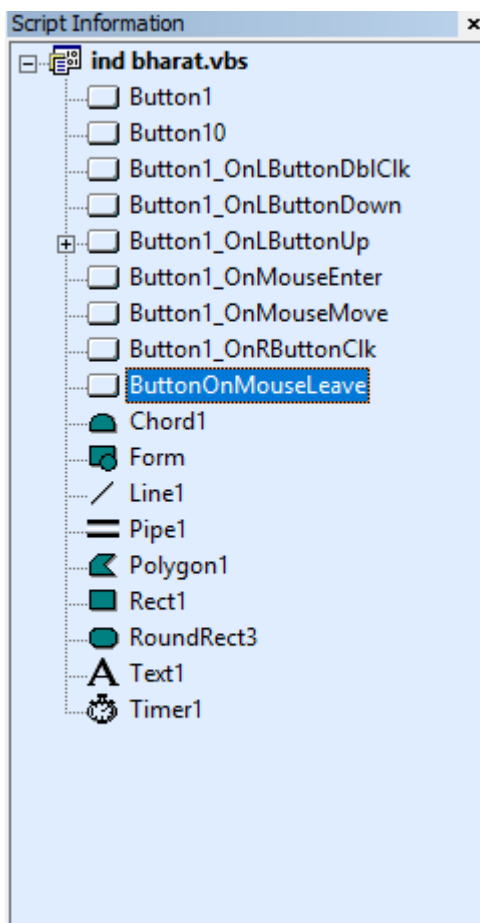


Figure 2-7 Script Information

2.6 Edit Toolbar

As shown in Figure 2-8, Edit Toolbar provides users with some common edit menu items in tool icons, making it easier for users to realize functions such as script edit, annotation, lock text and so on.

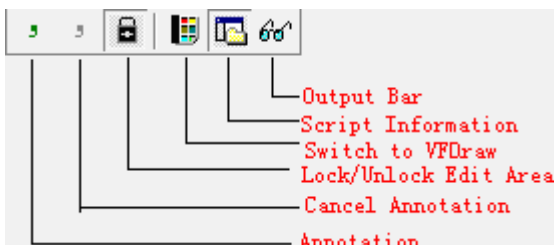



Figure 2-8 Edit toolbar

2.7 Output Information Window

The Output Window is located below the script edit area, users can click the icon  in edit toolbar to show or hide the Output Window.

When users finish editing script program, the prompt information of syntax check result of current script will be displayed in Output Information Window after click the command **Debug/ Syntax Check**, and users can skip to the location of wrong script by double-clicking the error prompt.

2.8 Add Function and Event Wizard

VFScript provides two functions of “Add Function” and “Event Wizard” in order to easily build functions and events.

Right-click any figure object name in script information bar, and the items of “Add Function” and “Event Wizard” are included in every right-click menu. The “Add Function” dialog box shown in Figure 2-9 will pop up when selecting “Add Function”. In the box, “Function” can use return value, while “Sub” can’t.

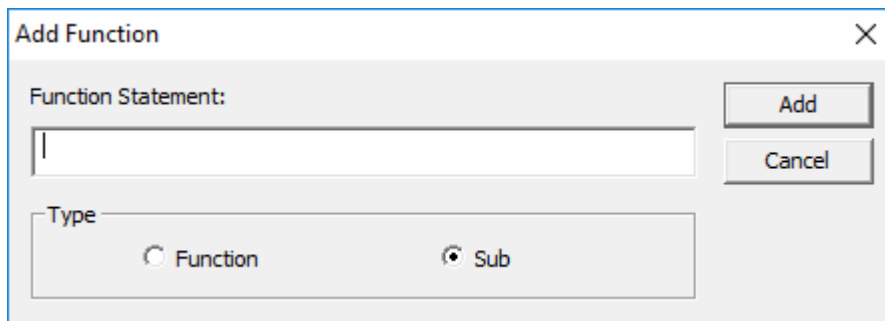


Figure 2-9 Add Function

The “Event Wizard” dialog box will pop up when selecting “Event Wizard”, as shown in Figure 2-10. Click “Add” after selecting “Object” and “Event”, the definition of event function will be added automatically in the script.

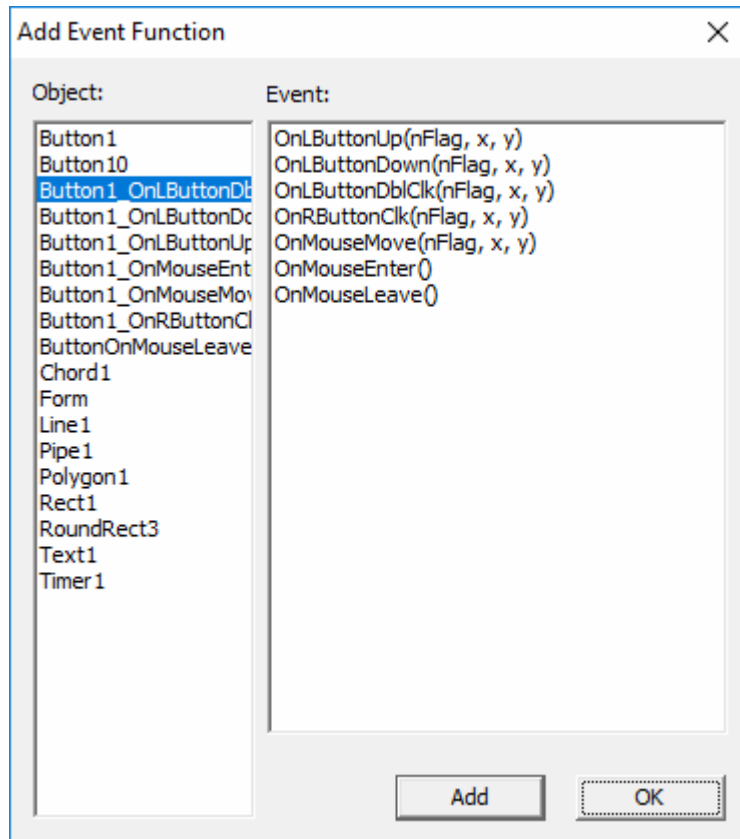


Figure 2-10 Event Wizard

Section 3 Instruction to Object Properties

3.1 Window Objects

Window object is another name of graphics figure, one graphics figure is corresponded to one script edit file (the name of graphics+.vbs).

Table 3-1 Instructions to window function

Function name	Function	Syntax	Example
CloseCurPop Pic	Close current pop-up screen	CloseCurPopPic	Form.CloseCurPopPic
GetScreenID	Access current graphics	GetScreenID	Form.GetScreenID
Login	Display the login window and switch user	Login	Form.Login
Logout	Logout current user	Logout	Form.Logout
OpenPic	Open the picture	OpenPic("the name of picture .pic") OpenPic("Resource File Name\ Window Name .pic")	Form.OpenPic("test.pic") Function: open the picture test.pic Form.OpenPic("a1\test1.pic") Function Instruction: open picture test1.pic in resource folder a1.
OpenPage	It is used to appoint the screen	OpenPage screen number, type number and image number type number 0=total image, 1=trend image, 2=group image, 3=overview image, The image number corresponds to the images in the operation team.	Form.OpenPage 1, 0, 3 Function illustration: 1 screen displays the overview screen 3.
OpenPicEx	Open the graphics of the reference domain.	OpenPicEx("reference domain alisa\ graphic name.pic")	Form.OpenPic("a1\poppic.pic") Function: open the graphics named poppic.pic of the reference domain named "a1".
ShowWindow	Pop up the picture	ShowWindow ("the name of picture+.pic") ShowWindow ("Resource File Name\ Window Name .pic")	Form.ShowWindow ("test.pic") Function: pop up the picture test.pic Form.ShowWindow ("a1\test1.pic") Function Instruction: pop up picture test1.pic in resource folder a1. Note: the full name of graphics is needed
PrintScreen	Print the screen	PrintScreen	Form.PrintScreen
ShutDown	Exit the graphics	ShutDown	Form.ShutDown
ShowLayer	Show/ Hide appointed layer	ShowLayer(layer, Show or Hide)	Form.ShowLayer 0,True Function: Show all the objects of layer 0

Function name	Function	Syntax	Example
TagES	Electrical signature verification of tag write value	TagES "Tag Name. Field", the value to be written	Single tag write value: Form.TagES "A1.PV", 2 Multiple tag write value: Dim a a = Array("A1.PV", "A2.PV") Dim b b = Array(3,5) Form.TagES a, b Function Instruction: pop up a electrical signature. Display signature box according to the highest signature requirement in tags.
ShowQRCode	The QRCode associated with the specified tag pops up	ShowQRCode("tagname")	Form.ShowQRCode("AI00020000") Function description: when the QRCode function is enabled in VFSysBuilder and the tag's associated QRCode is configured, and the QRCode of AI00020000 will pop up in the graphics as the script runs.
Remarks: The difference between pop up and open: the operator can move a pop-up picture while can't move an opened picture, in addition, an opened picture must be displayed in full screen.			

Table 3-2 Window property

Property Name	Function	Syntax	Example
Tag	Read or write a single tag. Operation will be recorded into log.	Tag("tag name.value") For tag in reference domain, tag name should include the reference domain name.	(1) Tag("aaa.pv") = 102.3 Function: IO input tag writes. (2) Tag("bbb.value") = 102.3; Tag("bbb.value") = 1 Function: self-defined tag and operation node tag write. (3) Tag("bbb.out") = 1 Function: IO output tag writes. (4) Dim MyTag MyTag = Tag("bbb.value") Function: assign real-time value of tag bbb to variable MyTag.
TagN	Read or write a single tag. Operation will not be recorded into log.	TagN("tag name.value") For tag in reference domain, tag name should include the reference domain name.	(1) TagN("aaa.pv") = 102.3 Function: IO input tag writes. (2) TagN("bbb.value") = 102.3; TagN("bbb.value") = 1 Function: self-defined tag and operation node tag write. (3) TagN("bbb.out") = 1 Function: IO output tag writes. (4) Dim MyTag MyTag = TagN("bbb.value") Function: assign real-time value of tag bbb to variable MyTag.

**Tip:**

Function is an operation of returning a value; property means graphics object properties that can be modified; event means the operation of triggering a process.

3.2 Graphics Objects

Graphic objects are all the objects (except control) of “Object Browser” in the toolbar of VFDraw.

The script interface provided by graphics object can realize more functions. The interface of graphics object includes “public interface” and “common interface”. Public interface means the interface that all the graphics objects and controls have; common interface means the interface that most graphics objects have and often use, not all the graphics objects support the common interface.

The graphics object will disappear on current screen when its boundary is set beyond the screen.

Table 3-3 Graphics Object interface type list

Interface Type	Interface Name	Interface Function	Example
Public Interface	Bottom	Set the bottom boundary of object, effective range [-4096,4096]	Text1.Bottom = 20 Function: set the bottom boundary of the textbox as 20 pixel on the screen
	Height	Set the height of object, effective range [-4096,4096]	Text1.Height = 20 Function: set the height of the textbox as 20 pixel
	Layer	Set the layer where the graphics lie, effective value [0,3]	Text1.Layer = 1 Function: set the layer where the textbox object lies as the first layer
	Left	Set the left boundary of object, effective range [-4096,4096]	Text1.Left = 20 Function: set the left boundary of textbox as 20 pixel
	Right	Set the right boundary of object, effective range [-4096,4096]	Text1.Right = 20 Function: set the right boundary of textbox as 20 pixel
	Top	Set the top boundary of object, effective range [-4096,4096]	Text1.Top = 20 Function: set the top boundary of textbox as 20 pixel
	Visible	Set the visibility of control object, and the effective value is True and False	Text1.Visible = True Function: the status textbox is visible
	Width	Set the width of object, effective range [-4096,4096]	Text1.Width = 200 Function: set the width of textbox as 200 pixel
Common Interface	Angle	The angle that control object rotates in clockwise	RondRect1.Angle = 30 Function: rotate rectangle object RondRect1 30 degrees in clockwise
	BackgroundColor	Set the background color of object. Please refer to Table 5-1 color constant on color constant comparison	RondRect1.BackgroundColor = vbBlack Function: set the background color of current rectangle as black
	BackgroundStyle	Set the background style of object. Please refer to Table 5-2 background style on background style	RondRect1.BackgroundStyle = 0 Function: set the current background style as solid fill

Interface Type	Interface Name	Interface Function	Example
	EdgeColor	Set the color of the object's edge. Please refer to Table 5-1 color constant on color constant comparison	RondRect1.EdgeColor = vbRed Function: set the color of rectangle's edge as red
	EdgeStyle	Set the style of the object's edge. Please refer to Table 5-3 color constant comparison on color constant comparison	RondRect1.EdgeStyle = 0 Function: set the style of rectangle's edge as real line.
	EdgeWidth	Set the width of object's edge. Effective range [0,6]	Line1.EdgeWidth = 2 Function: set the width of the line as 2 pixel
	GradientColor	Set the gradient color of object	RondRect1.GradientColor = vbRed Function: set the rectangle's gradient color as red
	GradientStep	Set the gradient step of object, effective range [1,16] Remarks: step is the pixel of the object's color gradient, when the step is 1, the unit of gradient will be 1 pixel; the less the step is, the softer the effect will be, but the more the CPU will be occupied during the refreshing	RondRect1.GradientStep = 2 Function: set the rectangle's gradient step as 2
	GradientStyle	Set the gradient style of object. Please refer to Table 5-4 gradient style comparison table on gradient parameter	RondRect1.GradientStyle = 1 Function: set the rectangle's gradient style as left to right
	IsSemiTransparent	Whether the object is semitransparent, and the effect value is FALSE and TRUE	RondRect1. IsSemiTransparent = True Function: set the rectangle object as semitransparent
	RotateCenterX	Rotate center X is the center's horizontal coordinate when rotating	RondRect1. RotateCenterX = 20 RondRect1. RotateCenterY = 50 Function: the rectangle rotates around the center which is 20 pixel in horizontal coordinate and 50 pixel in vertical coordinate
	RotateCenterY	Rotate center Y	

3.3 Global Variable(App)

The lifecycle of App object is the same with that of supervision system that it won't disappear after turning, which can be used to transfer parameter and obtain system parameter between pages.

Table 3-4 Global Variable

Function Name	Function	Syntax	Example
ACA	Count the alarm status of one or more tags in the specified alarm region.	App.ACA("Alarm Group No.. Alarm Region No.") App.ACA("Alarm Group No.")	Alarm function shall be ACA("16.31"). Function instruction: count tag alarm status of No. 16 alarm group and No. 31 alarm region.

Function Name	Function	Syntax	Example
ACP	Count the alarm status of the specified graphics.	APP.ACP("Graphics Name.PIC")	Alarm function shall be: ACP("EF Compound. PIC"). Function instruction: count alarm status of graphics "EF Compound".
ACT	count alarm status of one or several tags in graphics	ACT("Tag 1; Tag 2.....Tag n "). ACT function can count maximum 50 tags, which are separated by "." and should be tags in group and region of current operation team.	Alarm function shall be ACT("tag1;tag2"). Function instruction: count alarm status of tag1 and tag2 in graphics.
ACAN ^{Note1}	Counts alarms that belong to the specified alarm group or area, or counts the highest alarm level or suppressed alarm numbers.	<p>ACAN ("Alarm group number, alarm region number, return type") or ACAN ("alarm group number, return type")</p> <ul style="list-style-type: none"> Alarm group number: when the alarm group is "default alarm group", "alarm team number" is from 0 to 199 or OA. Number means control domain number, OA means control domain of master computer. When alarm group is "custom alarm group", the range is from 0 to 199. Alarm region number: the range is from 0 to 199 and 250. 250 is alarm region number of JX-300XP/ECS-100 direct domain. Return type supports 0-9, ALL, UNACK, numbers, UNTACK, SUP and PRL. 	<p>Alarm function should be in the following format: ACAN ("16.31;1")</p> <p>Function illustration of the example: it counts total alarms of which team number is 16, alarm region number is 31 and alarm level is 1, including the suppressed alarms.</p>

Function Name	Function	Syntax	Example
ACANS	Counts alarms that belong to the specified alarm group or area, or counts the highest alarm level. The shelved alarms are excluded.	<p>ACAN ("Alarm group number, alarm region number, return type") or ACAN ("alarm group number, return type")</p> <ul style="list-style-type: none"> Alarm group number: when the alarm group is "default alarm group", "alarm team number" is from 0 to 199 or OA. Number means control domain number, OA means control domain of master computer. When alarm group is "custom alarm group", the range is from 0 to 199. Alarm region number: when the alarm group is "default alarm group", the range is from 0 to 199 and 250. 250 is alarm region number of JX-300XP/ECS-100 direct domain. When the alarm group is "custom alarm group", the range is from 0 to 199. Return type is used to return the corresponding alarm number. Number refers to the alarm number of specified alarm level; ALL refers to the total number of alarms, UNACK refers to unacknowledged alarms, PRI refers to the level of the highest alarm. SUP refers to the total number of the suppressed alarms. 	<p>Alarm function should be in the following format: ACAN ("16.31;1")</p> <p>Function illustration of the example: it counts total alarms of which team number is 16, alarm region number is 31 and alarm level is 1, excluding the suppressed alarms.</p>

Function Name	Function	Syntax	Example
GetCurrentUser	Get current user	App.GetCurrentUser	App.GetCurrentUser Function: get the user name of current login Return Value: the login user name in the format of string.
GetCurrentUserGrade	Get current user grade	App.GetCurrentUserGrade	App.GetCurrentUserGrade Function: set properties, such as pressing and visibility according to the authority grade. Return Value: 9 means the user's grade is privilege+, 8 means the user's grade is privilege, 7 means the user's grade is privileged-, 6 means the user's grade is engineer+, 5 means the user's grade is engineer, 4 means the user's grade is engineer-, 3 means the user's grade is operator+, 2 means the user's grade is operator, 1 means the user's grade is operator-, 0 means the user's grade is observer. And, -1 means the user is not existed.
GetGroup	Get the name of current operating group	App.GetGroup	App.GetGroup Function: get the name of current operating group Return Value: the operating group in the format of string.
Mute	Alarm mute	App.Mute	App.Mute Function: eliminate current alarm
SaveValue	Write global variable	App.SaveValue ("global variable", value or variable to be saved)	App.SaveValue "MyValue", 100 Function: save 100 to global variable MyValue
About	Display about dialog box	App.About	App.About
ReadValue	Read all process property. Return previously assigned variable value if succeeds; return Empty if fails	App.ReadValue("global variable")	b = App.ReadValue("MyValue") Function: read the value from global variable MyValue. b=100 if successfully read, or return Empty if fails.
ShowQRCode	Popup the specified tag's QRcode.	ShowQRCode("tagname")	ShowQRCode("AI00020000") Function: After enable QRcode in the System Builder software and configured QRcode in the Tag Builder software, the QRcode of AI00020000 will popup in the current graphic after the script executed.

Note1: For the detailed description of ACAN function, please refer to the "expression" section in *Graphics Builder (VFDdraw) User Manual*.

Section 4 VFScript Application Example

4.1 Event Function

4.1.1 OnLButtonUp

Left Button Up Event, the event will be operated when the left mouse button is released from graphics object.

For example, round rectangle will rotate 60 degrees when left mouse button is released.

Draw a button "Button1" and a round rectangle "RoundRect1", edit script for "Button1" is shown as follows:

```
Sub Button1_OnLButtonUp(nFlag, x, y)
```

```
RoundRect1.Angle = 60
```

```
End Sub
```

The round rectangle will rotate 60 degrees when left mouse button is released from "button1" of monitoring window.

4.1.2 OnLButtonDown

Left Button Down Event, the event will be operated when the left mouse button is pressed on graphics object.

For example, rectangle will rotate 30 degrees when left mouse button is pressed.

Draw a button "Button1" and a rectangle "Rect1", edit script for "Button1" is shown as follows:

```
Sub Button1_OnLButtonDown(nFlag, x, y)
```

```
Rect1.Angle = 30
```

```
End Sub
```

The rectangle will rotate 30 degrees when left mouse button is pressed on "button1" of monitoring window.

4.1.3 OnLButtonDbIClk

Double Click Left Button Event, the event will be operated when double-clicking the left mouse button on graphics object.

For example, the bottom edge value of ellipse will be 200 when double-clicking the left mouse

button.

Draw a button "Button1" and an ellipse "Ellipse1", edit script for "Button1" is shown as follows:

```
Sub Button1_OnLButtonDblClk(nFlag, x, y)
```

```
Ellipse1.Bottom = 200
```

```
End Sub
```

The bottom edge value of ellipse will be set as 200 when double-clicking "button1" of monitoring window.

4.1.4 OnRButtonClk

Click right Button Event, the event will be operated when right-clicking the graphics object.

For example, the pipe will be set in layer 2 when right-clicking the mouse button.

Draw a button "Button1" and a pipe "Pipe1", edit script for "Button1" is shown as follows:

```
Sub Button1_OnRButtonClk(nFlag, x, y)
```

```
Pipe1.Layer = 2
```

```
End Sub
```

The pipe will be set in layer 2 when right-clicking "button1" of monitoring window.

4.1.5 OnMouseMove

Move Mouse Event: the event will be operated when moving the mouse on graphics object.

For example, the width of the polygon will plus 5 when moving the mouse

Draw a button "Button1" and a polygon "Polygon1", edit script for "Button1" is shown as follows:

```
Sub Button1_OnMouseMove(nFlag, x, y)
```

```
Polygon1.Width = Polygon1.Width + 5
```

```
End Sub
```

The width of the polygon will plus 5 when moving the mouse once on "Button1" of monitoring window.

4.1.6 OnMouseEnter

Mouse Enter Event: the event will be operated when the mouse pointer moves into the graphics object.

For example, the pie will hide when the mouse moves into the button.

Draw a button "Button1" and a pie "Pie1", edit script for "Button1" is shown as follows:

```
Sub Button1_OnMouseEnter()
```

```
Pie1.Visible = False
```

```
End Sub
```

The pie will hide when the mouse moves into the "Button1" area in monitoring window.

4.1.7 OnMouseLeave

Mouse Leave Event: the event will be operated when the mouse moves out from the graphics object.

For example, the text will be centered when the mouse moves out from the button object.

Draw a button "Button1" and add a text object "text1", edit script for "Button1" is shown as follows:

```
Sub Button1_OnMouseLeave()
```

```
Text1.Align = 2
```

```
End Sub
```

The Text1 will be centered when the mouse moves out from the "Button1" in monitoring window.

4.1.8 OnOpenPic

Execute OnOpenPic function when opening pop-up graphics.

For example: AI03040001 tag value will add 1 when opening pop-up graphics 1.PIC.

Add following script to pop-up graphics 1.PIC:

```
Sub Form_OnOpenPic()
```

```
    TAG("AI03040001.PV")=TAG("AI03040001.PV")+1
```

```
End Sub
```

4.1.9 OnClosePic

Execute OnClosePic function when closing pop-up graphics.

For example: AI03040001 tag value will reduce 1 when opening pop-up graphics 1.PIC.

Add following script to pop-up graphics 1.PIC:

```
Sub Form_OnClosePic()

    TAG("AI03040001.PV")=TAG("AI03040001.PV")-1

End Sub
```

4.1.10 OnTimer

OnTime Event in timer: start timer

For example: display the current time.

Draw a button “Button1”, text “text2” and a timer “timer1”, edit script for “timer1” and “Botton1” is shown as follows:

```
Sub Timer1_OnTimer()

Text2.Text=Now()

End Sub

'-----

Sub Button1_OnLButtonUp(nFlag, x, y)

i=3000

Timer1.Interval=i

If Button1.Text="stop timing" Then

Timer1.Enable=False

Button1.Text="start timing"

Else

Timer1.Enable=True

Button1.Text="stop timing"

End If
```

When the script is operated and i=3000ms, “Text2” will display the current time in every 4 seconds. And the triggering interval will be modified when modifying the value of i.

4.1.11 Instruction to Other Sentences of Graphics Script

Since the graphics script supports VBScript, VBScript can be easily applied to graphics script to run the needed operation, please refer to the Instruction to VBScript for detailed syntax.

When the graphics need to operate certain figure by using the script, first input name of the graphics object, and then input “.”, and the programmable property of the graphics object will be displayed.

```
Sub Button2_OnLButtonUp(nFlag,x,y)  
    Rect2.
```



Figure 4-1 Property of rectangle object

4.2 Notes

Use App.MsgBox function instead of MsgBox function in pop-up graphics.

Section 5 Appendix

5.1 Color constant

These constants are set in VFScript, and it can be used at any time, users needn't define them before using.

Table 5-1 Color constant

Constant	Value	Description
vbBlack	&h00	Black
vbRed	&hFF	Red
vbGreen	&hFF00	Green
vbYellow	&hFFFF	Yellow
vbBlue	&hFF0000	Blue
vbMagenta	&hFF00FF	Carmine
vbCyan	&hFFFF00	Cyan
vbWhite	&hFFFFFF	White

5.2 Background style

Table 5-2 Background style

Style Name	Value
Solid fill	0 - bgSOLIDFILL
None	1 – bgNOHATCH
Left diagonal	5 - bgBDIAGONAL
Right diagonal	4 – bgFDIAGONAL
Cross grid	6 - bgCROSS
Diagonal cross	7 - bgDIAGCROSS

Style Name	Value
Vertical line	3 - bgVERTICAL
Horizontal line	2 – bgHORIZONTAL

5.3 Edge Style

The edge styles in the table below apply to the shapes of graphics object, such as rectangle.

Table 5-3 Edge Style

Style Name	Value
No Line	5 - lsNOLINE
Solid line	0 – 1sSOLID
Dash line	1 - lsDASH
Dot line	2 - lsDOT
Dash dot line	3 - lsDASHDOT
Dash two-dot line	4 - lsDASHDOTDOT

The edge styles in the table below apply to the system variable, character and data link of graphics object.

Style Name	Value
Null	0-fsBORD_NULL
Up	2- fsBORD_UP
Down	1- fsBORD_DOWN
Etch	3-fsBORD_ETCH
Shadow	4-fsBORD_SHADOW
Frame	5-fsBORD_FRAME

5.4 Gradient Style

Table 5-4 Gradient Style

Style Name	Value
Solid fill	0 - gsSOLID
Left to right	1 - gsLEFTRIGHT
Top to bottom	2 - gsTOPBOTTOM
Horizontal center	3 - gsHCENTER
Vertical center	4 - gsVCENTER
Top left radiation	5 - gsTOPLEFT
Bottom left radiation	6 - gsBOTTOMLEFT
Bottom right radiation	7 - gsBOTTOMRIGHT
Top right radiation	8 - gsTOPRIGHT
Lean down	9 - gsLEANDOWN
Lean up	10 - gsLEANUP
Center to edge	11 - gsCENTER

Section 6 Revision

Table 6-1 Retrofit list of the version

Document Version	Applicable software version	Remarks
V1.0 (20230301)	OMC High-performanceHMI V4.70.00.00	First release
V1.1 (20230830)	OMC High-performanceHMI V5.10.00.00-M	Updated screenshots.