



RSView[®] Machine Edition[™]

ENTERPRISE SERIES

Machine-Level HMI for Open
and Embedded Solutions

User's Guide
Volume 2

Doc ID VIEWME-UM005C-EN-E

**Rockwell
Automation**

Contacting Rockwell Software	<p>Technical Support Telephone—440-646-5800</p> <p>Technical Support Fax—440-646-5801</p> <p>World Wide Web—www.software.rockwell.com <i>or</i> support.rockwellautomation.com</p>
Copyright Notice	<p>© 2004 Rockwell Software Inc. All rights reserved.</p> <p>Printed in the United States of America.</p> <p>Portions copyrighted by the Allen-Bradley Company, LLC, a Rockwell Automation company.</p> <p>This manual and any accompanying Rockwell Software products are copyrighted by Rockwell Software Inc. Any reproduction and/or distribution without prior written consent from Rockwell Software Inc. is strictly prohibited. Please refer to the license agreement for details.</p> <p>VBA and DCOM, Copyright 1996, 1997 Microsoft Corporation. All rights reserved.</p> <p>This software is based in part on the work of the Independent JPEG Group.</p>
Trademark Notices	<p>Allen-Bradley, ControlLogix, RSLinx, RSView, and VersaView are registered trademarks, and the Rockwell Software logo, RSLogix, RSTools, RSView Machine Edition, RSView ME Station, RSView Studio, RSView Supervisory Edition, A.I. Series, Advanced Interface (A.I.) Series, ControlNet, FactoryTalk, Object Smart Path, PanelBuilder, PanelView Plus, SLC, and WINtelligent are trademarks of Rockwell Automation, Inc.</p> <p>Adobe and Reader are registered trademarks of Adobe Systems Incorporated. TrueType is a trademark of Apple Computer, Inc. AutoCAD is a registered trademark of Autodesk, Inc. Ethernet is a registered trademark of Digital Equipment Corporation, Intel Corporation, and Xerox Corporation. Modicon is a registered trademark of Groupe Schneider. Zip is a trademark of Iomega Corporation.</p> <p>KEPServerEnterprise is a trademark of Kepware Technologies. ActiveX, Microsoft, OpenType, Visual Basic, and Windows are registered trademarks of Microsoft Corporation in the United States and/or other countries. OPC is a registered trademark of the OPC Foundation.</p> <p>All other trademarks are the property of their respective holders and are hereby acknowledged.</p>
Warranty	<p>This Rockwell Software product is warranted in accord with the product license. The product's performance will be affected by system configuration, the application being performed, operator control, and other related factors. The product's implementation may vary among users.</p> <p>This manual is as up-to-date as possible at the time of printing; however, the accompanying software may have changed since that time. Rockwell Automation reserves the right to change any information contained in this manual or the software at any time without prior notice.</p> <p>The instructions in this manual do not claim to cover all the details or variations in the equipment, procedure, or process described, nor to provide directions for meeting every possible contingency during installation, operation, or maintenance.</p> <p>Doc ID ViewME-UM005C-EN-E</p> <p>September 2004</p>

Setting up graphic objects

This chapter describes how to set up graphic objects. It describes how to:

- set up objects' spatial properties, names, and visibility.
- work with objects that have states.
- position objects for touch screens.
- assign function keys to buttons.
- use the keyboard to navigate to and select objects at run time.
- link buttons to lists and trends.
- repeat a button's action by holding down the button.
- ensure values are read by the data source before sending new values.
- set up objects.

For objects that are not described in other chapters of the manual, this chapter provides information about how to use the objects.

For information about creating graphic objects, see Chapter 15.

For information about creating and setting up the trend graphic object, see Chapter 18.

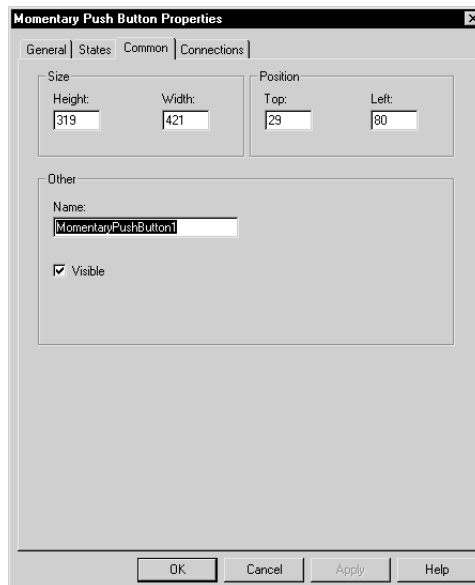
Setting up objects' spatial properties, names, and visibility

Every graphic object has a Common tab in its Properties dialog box. Use the Common tab to set up the following properties for the object:

- height and width
- top and left position
- name
- visibility

For ActiveX[®] and trend objects, you can also set up these properties:

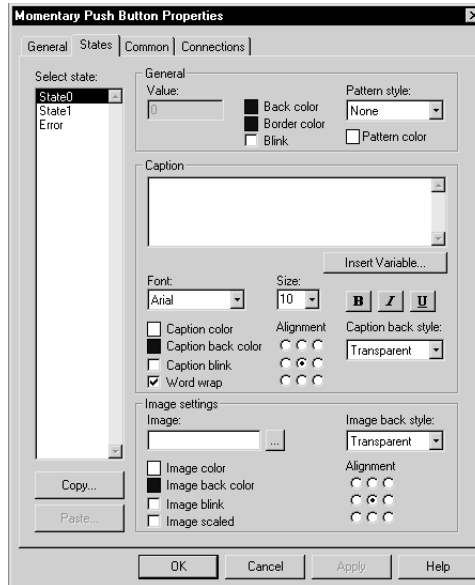
- focus highlight
- key navigation



For details about the options in the Common tab, see Help.

Tips for setting up objects with states

The graphic objects that have states have a States tab in their Properties dialog box.



For details about the options in the States tab, see [Help](#).

Copying and pasting properties from one state to another

Often, you'll want most of the properties to be the same from state to state, with only one or two settings changing to distinguish the different states.

To simplify setting up states when many of the properties are the same, you can copy and paste settings from one state to another. For details, see [Help](#).

Adding and removing states

Some graphic objects have a configurable number of states. For these objects, you can use the Insert State and Delete State buttons in the States tab, to add and remove states without returning to the General tab. The “Number of States” setting on the General tab is automatically updated.

For more information about the Insert State and Delete State buttons, see Help.

Setting up how objects are used at run time

Just as you must provide operators with a way to navigate between displays at run time, you must also make sure that operators can use the objects within the displays. The next sections in this chapter describe how to set up your objects so the operator can:

- press objects using a touch screen.
- use function keys to press buttons when a mouse or touch screen is not available on the runtime computer.
- use the keyboard or keypad to navigate to and select lists, trends, and ActiveX input objects.
- use buttons to work with lists and trends.
- repeat a button’s action by pressing and holding it.
- ensure that tag values are read by the data source before sending new values.

Positioning objects for touch screens

If the operator will be using a touch screen at run time, keep these tips in mind when positioning graphic objects in your displays:

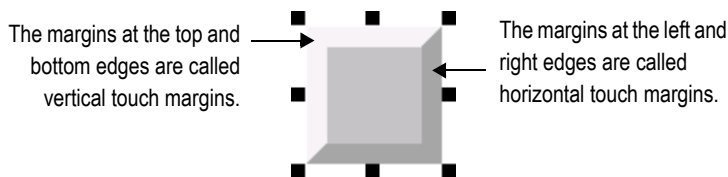
- Don't place important buttons where they'll be blocked by an On Top display. The user can't press a covered button.
- Ensure buttons are large enough for users to touch easily.
- Use touch margins for buttons that are positioned close together, to ensure that the adjacent button is not pressed by mistake.

Using touch margins

Touch margins are touch-insensitive borders inside the button's margin. If the operator presses on the touch margin, the button press is not registered. Touch margins are useful when buttons are positioned close to each other and you want to make sure the operator doesn't press the wrong button by mistake.

You can create touch margins at the top and bottom of the button, at the sides, or on all four sides.

In the illustration below, the button's border and touch margins are the same size, 12 pixels. A button press would be registered only when the darker square in the middle of the button is pressed.



The bounding box

If the object's shape is a circle or ellipse, the touch margin applies to the object's bounding box, not the object's border. The bounding box is an

invisible square or rectangle that surrounds the object. When you select the object, the selection handles show the location of the bounding box.



To create touch margins, use one of these methods

- In the General tab of the button's Properties dialog box, type the number of pixels for the touch margins in the Horizontal margin and Vertical margin boxes.
- In the Properties tab of the Property Panel, type the number of pixels for the `HorizontalMargin` and `VerticalMargin` properties.

Assigning function keys to buttons

You can assign function keys to the buttons in your displays to allow the operator to press the buttons using the function keys on the runtime terminal (or the function keys on a keyboard, if one is available). You can also assign a function key to the numeric input cursor point.

You can assign up to 34 function keys to each graphic display.

Here are some tips for assigning function keys:

- Where possible, use the same function keys for the same operations in all your graphic displays. For example, if every display contains a goto display button that returns the operator to a graphic display called Main Menu, assign the same function key to this button in each display.

- Include the name of the function key assigned to a button in a caption on the button.

For buttons with multiple states, include the function key name in the caption for each state, or create a text object to use as a label for the button (so that you don't have to set up the caption for each state), and then group the text and button together.

- If your application will run on a PanelView Plus™ or VersaView® CE terminal, assign keys that are supported by the runtime terminal. Different sizes of terminals have different function keys.

Function key equivalents

If your application will run in Windows® 2000 or Windows XP, the keyboard contains the function keys F1 to F12 only. The remaining function keys are associated with these key combinations:

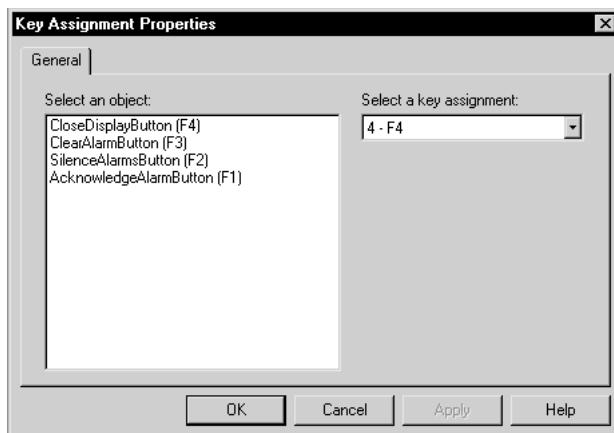
For this function key	Use this key combination
F13	Left Shift-F1
F14	Left Shift-F2
F15	Left Shift-F3
F16	Left Shift-F4
F17	Left Shift-F5
F18	Left Shift-F6
F19	Left Shift-F7
F20	Left Shift-F8
K1 - K12	Right Alt-F1 to Right Alt-F12
K13	Right Shift-F1
K14	Right Shift-F2
K15	Right Shift-F3
K16	Right Shift-F4

For this function key	Use this key combination
K17	Right Shift-F5
K18	Right Shift-F6
K19	Right Shift-F7
K20	Right Shift-F8

At run time, the operator presses the key combination to activate the object to which the function key is assigned.

To assign function keys to buttons

- On the Edit menu, click Key Assignments, or right-click a button and then click Key Assignments.



For details about using the Key Assignment Properties dialog box to assign function keys and change function key assignments, see Help.

Using the keyboard to navigate to and select objects

If a mouse or touch screen is not available on the runtime computer, the operator can use the keys on a keyboard or keypad to select (give focus to) these objects:

- lists: control list selector, piloted control list selector, display list selector, diagnostics list, alarm list, and alarm status list
- alarm banners
- trends
- numeric input enable buttons and string input enable buttons
- numeric input cursor points
- third-party ActiveX input objects

What input focus looks like

The object with focus is surrounded by a highlight box, unless the Disable Highlight When Object has Focus box is selected (in the Display Settings dialog box). You can specify the color of the highlight in this dialog box as well. For more information, see Help.



Focus highlight for ActiveX and trend objects

For ActiveX and trend objects, use the Common tab in the object's Properties dialog box to specify whether or not to display a highlight box. For more information about setting up options on the Common tab, see page 16-2.

If the Disable Highlight When Object has Focus box is checked (in the Display Settings dialog box), that setting overrides the setting you specify in the Common tab.

Using the keys on the keyboard or keypad

When a graphic display opens, the keyboard-navigable object that is closest to the top left corner of the display is selected. The operator can use these keys to move to and select a different object:

Use this key	To do this
Tab	Move from the upper left to the lower right.
Shift+Tab	Move from the lower right to the upper left.
Ctrl+arrow key	Move left, right, up, or down.

Removing objects from and adding objects to the tab sequence

By default, you can use the keys to navigate to all lists, alarm banners, numeric input cursor points, trends, and ActiveX input objects in a display. However, you can turn off key navigation for these objects if desired.

When an object's key navigation is turned off, the operator can still select the object using a mouse or touch screen, if available.

By default, key navigation is turned off for the numeric input enable buttons and string input enable buttons. You can turn on keyboard navigation if you want the operator to use the keyboard to navigate to these objects.

To turn key navigation on or off, use one of these methods

- In the Properties tab of the Property Panel, set the KeyNavigation property to False for off or True for on.
- For list objects, alarm banners, numeric input cursor points, numeric input enable buttons, and string input enable buttons, in the General tab of the object's Properties dialog box, clear the Key navigation box for off. Check the box for on.
- For ActiveX and trend objects, in the Common tab of the object's Properties dialog box, clear the Key navigation box for off. Check the box for on.

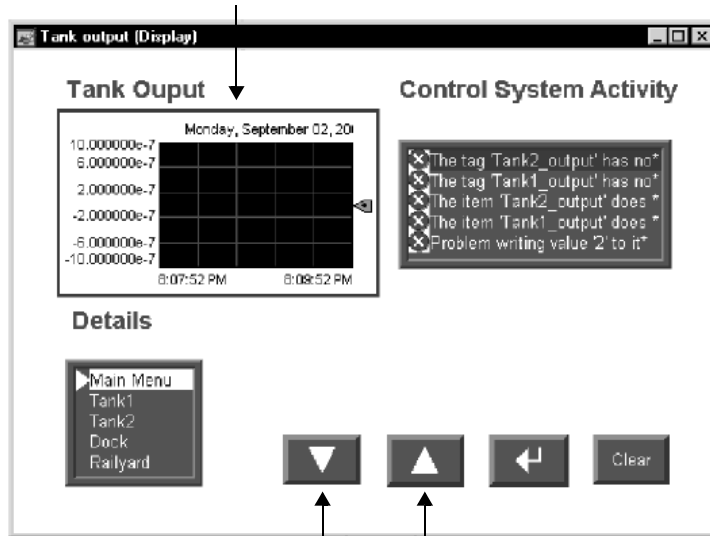
Linking buttons to trends, lists, and alarm banners

Some RSView® buttons can be linked to specific trend, list, and alarm banner objects, or you can set them up to work with whichever object has focus in the display.

If you are creating small graphic displays that don't have much room, you might prefer to use one set of buttons to work with all the graphic objects (that accept input focus) in the display.

By default, the buttons are set up to work with whichever object has input focus (is selected) in the display.

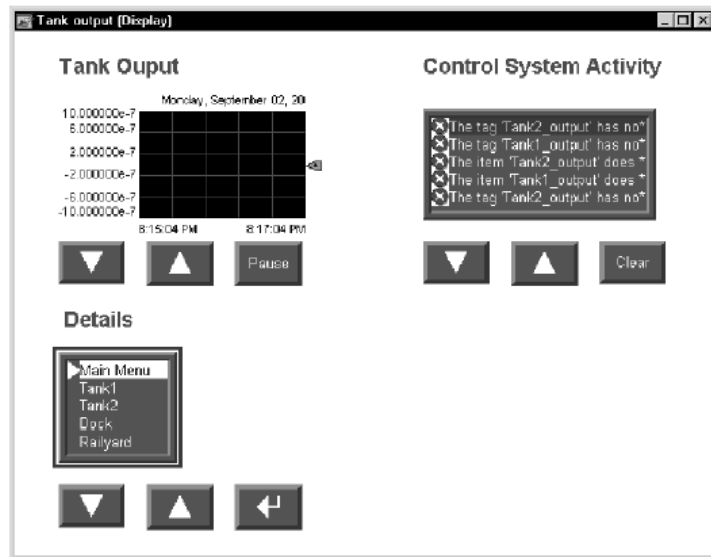
This trend object has focus.



These move up and move down buttons work with the object that has focus.

However, if space isn't a concern, you can create multiple copies of the buttons you want to use and link them to specific objects. One benefit of linking a button to a specific object is that the operator doesn't have to select the object before pressing the button. Another benefit is that

you can position the buttons close to the specified object, making it easier for the operator to understand which buttons work with which objects.



Each object has its own set of buttons.

To link a button to a specific object using the button's Properties dialog box

1. Double-click the button to open its Properties dialog box.
2. In the General tab, in the "Send press to" box, select Linked Object.
3. To select from a list of all the objects in the display that you can link the button to, click the Browse button next to the Linked object box.
4. Click the name of the object to link the button to, and then click OK.
5. Click OK to close the button's Properties dialog box.

To link a button to a specific object using the Property Panel

- In the Properties tab, specify the SendPressTo and LinkedObject properties.

Once you have linked buttons to an object, you might want to turn off the object's key navigation, since this option is no longer needed. For details, see page 16-11.

Repeating a button's action by holding down the button

To repeat a button's action by pressing and holding it, set up auto repeat for the button. If you set up auto repeat, when the operator presses and holds down the button, repeated button presses are registered until the operator releases the button.

A button press occurs when the operator clicks an object with the mouse, presses it on a touch screen, or presses the function key associated with the object. Auto repeat works with all these methods of pressing buttons.

You can use auto repeat with these buttons:

For this button	Each button press does this
Multistate push	Sends the value for the next state to the data source. External changes to the Value control are not recognized when the button is in auto repeat mode.
Ramp	Sends the new ramped value to the data source. External changes to the Value control are not recognized when the button is in auto repeat mode.
Move up	Moves the highlight up an item in the list, scrolls up in the trend, or ramps the numeric input cursor point or numeric input enable button value.

For this button	Each button press does this
Move down	Moves the highlight down an item in the list, scrolls down in the trend, or ramps the numeric input cursor point or numeric input enable button value.
Page up	Moves the highlight up a page in the list.
Page down	Moves the highlight down a page in the list.
Move left	Scrolls the trend to the left.
Move right	Scrolls the trend to the right.

For each button that uses auto repeat, you can specify these properties:

- Auto repeat rate—the number of times per second a button press is registered when the button goes into auto repeat mode. The default rate is 0, which means that auto repeat is turned off.
- Auto repeat delay—the length of time the button has to be pressed and held down before auto repeat starts.

To set up auto repeat for a button, use one of these methods

- In the Timing tab of the button's Properties dialog box, specify the Auto repeat rate and Auto repeat delay properties.
- In the Properties tab of the Property Panel, specify the AutoRepeatRate and AutoRepeatDelay properties.

Ensuring values are read by the data source before sending new values

To ensure a value is read by the data source before the operator sends a new value, use Enter key handshaking. While Enter key handshaking is in effect for an object, the operator cannot send a new value to the object's Value control.

You can use Enter key handshaking with these graphic objects:

- control list selector
- piloted control list selector
- numeric input enable button
- string input enable button
- numeric input cursor point

If Enter key handshaking is in effect for one of these objects, the operator can still provide input to other objects in the graphic display.

If the graphic display is closed while Enter key handshaking is in effect, the handshaking is cancelled.

Methods of Enter key handshaking

Enter key handshaking works by setting the object's Enter control to 1. As long as the Enter control is set to 1, new values cannot be sent to the Value control. How the Enter control is reset to 0 depends on how you set up Enter key handshaking.

There are two ways you can use Enter key handshaking:

- to hold the value at the data source for a specific period of time.
- to hold the value at the data source until the data source notifies RSVIEW that the value has been read.

Choose the method that best suits your application needs and communication system.

Holding the value for a specific period of time

To set up an object's Enter key handshaking so that the value at the Value control is held for a specific period of time, assign a tag to the

Enter control and specify the Enter key hold time. You can also specify an Enter key control delay, if desired.

How handshaking works

This method of Enter key handshaking works as follows:

1. When the operator presses the Enter button, the value is sent to the Value control and the “Enter key control delay” timer begins timing. (The use of a delay is optional.)
2. If you specify an Enter key control delay, when the time is up, the Enter control is set to 1. If you don’t use the delay, the Enter control is set to 1 as soon as the operator presses Enter.

As long as the Enter control is set to 1, the operator cannot send new values to the data source.

3. When the Enter control is set to 1, the “Enter key hold time” timer begins timing.
4. When the Enter key hold time has expired, the Enter control is reset to 0 and the operator can send a new value to the Value control.

To set up Enter key handshaking to hold the value for a specific period of time

1. In Timing tab of the object’s Properties dialog box, specify the Enter key control delay (optional) and Enter key hold time properties.
2. In the Connections tab, assign a tag to the Enter control. A digital tag is recommended (either an HMI tag or a data server tag).

You can also use the Property Panel to specify the properties and assign a tag to the Enter control.

Holding the value until the data source acknowledges that it has read the value

To set up an object’s Enter key handshaking so that the value at the Value control is held until the data source notifies RSView that it has read the value, use two controls: the Enter control and the Enter handshake control.

Instead of using an Enter key hold time, specify an Enter key handshake time. You must also specify the Handshake reset type. You can use an Enter key control delay, if desired.

How the Handshake reset type works

How the Enter handshake control resets the Enter control depends on which Handshake reset type you select:

With this handshake reset type	The Enter control is set to 0 when
Non-zero Value	<p>The Enter handshake control has a non-zero value.</p> <p>If the Enter handshake control already has a non-zero value when the value is sent to the Value control (or when the Enter key control delay has expired, if the delay is used), then the Enter control is not set to 1, and Enter key handshaking does not take place.</p>
Zero to Non-zero transition	<p>The Enter handshake control changes from 0 to a non-zero value.</p>

Set up the data source to send a non-zero value to the Enter handshake control when it has read the new value at the Value control, and then to reset the Enter handshake control to 0.

If the Enter key handshake time expires before the Enter handshake control resets the Enter control, an error message is sent to FactoryTalk™ Diagnostics.

How handshaking works

If you use the Enter handshake control, handshaking works like this:

1. When the operator presses the Enter button, the value is sent to the Value control and the “Enter key control delay” timer begins timing. (The use of a delay is optional.)
2. If you specify an Enter key control delay, when the time is up, the Enter control is set to 1. If you don’t use the delay, the Enter control is set to 1 as soon as the operator presses Enter.

As long as the Enter control is set to 1, the operator cannot send new values to the data source.

If the Handshake reset type is Non-zero Value, the Enter handshake control must be 0 when the delay expires in order to set the Enter control to 1.

3. When the Enter control is set to 1, the “Enter key handshake time” timer begins timing.
4. The Enter control remains set until the Enter key handshake time expires or until reset by the Enter handshake control, whichever happens first.
5. The Enter control is reset to 0 and the operator can send a new value to the Value control.

To set up Enter key handshaking to hold the value until the data source has read it

1. In the Timing tab of the object’s Properties dialog box, specify these properties:
 - Enter key control delay (optional)
 - Enter key handshake time
 - Handshake reset type
2. In the Connections tab, assign these controls:

- Enter—assign a tag. A digital tag is recommended (either an HMI tag or a data server tag).
 - Enter handshake—assign a tag or expression.
3. Set up the data source to send a non-zero value to the Enter handshake control when it has read the new value at the Value control, and then to reset the Enter handshake control to 0.

You can also use the Property Panel to specify the properties and assign tags to the controls.

Time, date, and number formats for graphic objects

Graphic objects use the time, date, and number format you set up on the runtime computer. For example, if the runtime computer is set up to use a comma for the decimal symbol, the numeric display uses a comma for the decimal symbol.

For information about specifying time, date, and number formats:

- in Windows 2000 and Windows XP, see page 24-18.
- on a PanelView Plus terminal, see the *PanelView Plus Operator Terminals User Manual*.
- on a VersaView CE terminal, see the *VersaView CE Terminals User Manual*.

Setting up buttons

Many of the button graphic objects are set up the same way. The instructions in this section apply to these buttons:

Acknowledge alarm	Logout
Acknowledge all alarms	Move down
Alarm status mode	Move left
Backspace	Move right
Clear alarm banner	Move up
Clear alarm history	Next pen
Diagnostics clear	Page down
Diagnostics clear all	Page up
Display print	Pause
End	Reset alarm status
Enter	Return to display
Goto configure mode	Shutdown
Home	Silence alarms
Information acknowledge	Sort alarms
Login	

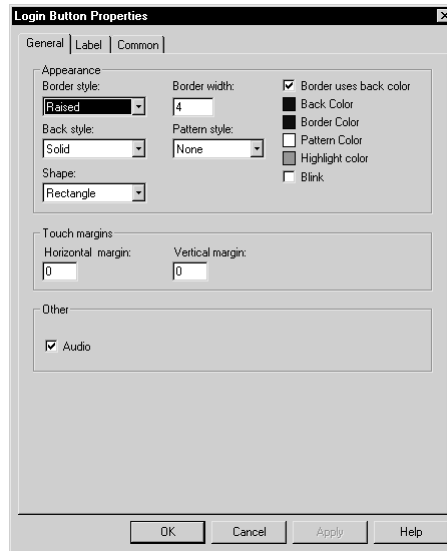
You can set up any combination of these buttons in a graphic display. For example, you might want to put a login, logout, and shutdown button in the same graphic display.

Create and then set up each button separately.

The illustration shows the login button. Some buttons have different properties on the General tab, but you can use these instructions to set up any of the buttons in the list.

To set up a button

1. Double-click the button.



2. In the button's Properties dialog box, fill in the options on each tab:
 - In the General tab, specify what the button looks like at run time. For some buttons, you can specify whether to link the button to a specific object.
 - In the Label tab, specify what text or image to display on the button.
 - In the Timing tab, set up whether or not the button press repeats automatically when the operator presses and holds the button down. You can also set up the rate at which the button press repeats. For more information about auto repeat, see page 16-15.

The Timing tab is available only for the move up, move down, move left, move right, page up, and page down buttons.

- In the Common tab, specify the button’s spacial properties, name, and visibility.

For details about the options in the tabs, see Help.

3. When you are finished, click OK.

Buttons described later in the chapter

Information about setting up these buttons is described later in the chapter:

Close display	Multistate push
Goto display	Numeric input enable
Interlocked push	Print alarm history
Latched push	Print alarm status
Macro	Ramp
Maintained push	String input enable
Momentary push	

How to use push buttons

Push buttons start or stop processes or actions by changing tag values.

IMPORTANT

Never use push buttons for emergency stops.
Emergency stop buttons must always be hard-wired.

You can use different kinds of push buttons, depending on what kind of machinery you are running or process you are controlling:

Momentary push buttons change a tag to one value when the button is pressed, and another value when the button is released. The machine is on only while the button is held down. When the button is released, the machine turns off. Momentary push buttons are useful for jogging a motor, and they can be set up to start and stop a machine or process.

Maintained push buttons toggle between two values. This type of button is useful for changing a setting within a machine or process, but not for starting the machine or process. For example, use the maintained push button for changing modes, such as Auto to Manual, or Metric to Imperial.

Latched push buttons latch in the on position, and must be unlatched by another button or process to return to the off position. This type of button is useful for starting a machine or process.

Multistate push buttons allow an operator to cycle through multiple options consecutively, using a single button. The current state of a process or operation is displayed on the button by a different color, caption, or image for each state.

Interlocked push buttons work in groups, and share the same tag. Pressing one button cancels another. Although interlocked push buttons work as a group, you add them to the display one at a time.

Ramp buttons increase or decrease the value of a tag by either an integer or floating-point value. You can use two ramp buttons together to create an increment/decrement control, for example for the speed of a motor. Or, you can use a numeric input enable button or numeric input cursor point.

For information about the numeric input enable button, see page 16-38. For information about the numeric input cursor point, see page 16-40.

Setting up momentary push buttons

Use the momentary push button to start a process or action. When pressed, the button changes a tag to one value, and when released, the button changes the tag to another value.

The momentary push button's states can perform one of three kinds of actions:

- Change the Value control to 1 when the button is pressed, and to 0 when the button is released. This kind of button is called **normally**

open, because its released state is off. Pressing the button completes the circuit.

- Change the Value control to 0 when the button is pressed, and to 1 when the button is released. This kind of button is called **normally closed**, because its released state is on. Pressing the button breaks the circuit.
- Change the Value control to a **value** you specify. You assign the desired values to the button's press and release actions. For example, 50 when pressed, and 100 when released.

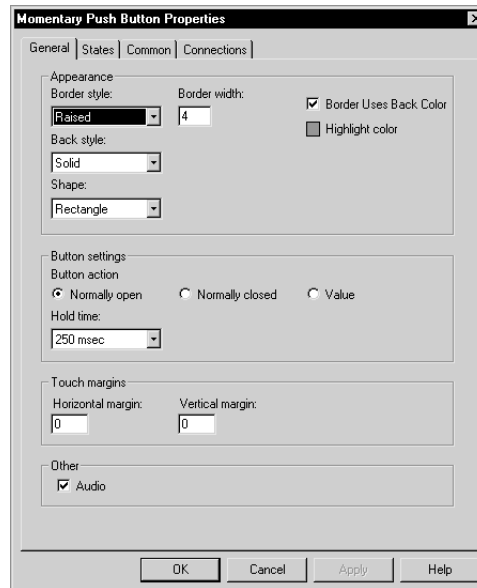
The error state

The button's error state is displayed at run time when:

- the Value control is unassigned.
- the Indicator control's value does not match one of the state values you set up.

To set up a momentary push button

1. Double-click the momentary push button.



2. In the button's Properties dialog box, fill in the options on each tab:

- In the General tab, specify what the momentary push button looks like at run time, and what type of action the button performs.
- In the States tab, specify what the button does when it is pressed and released. For tips about setting up states, see page 16-3.
- In the Common tab, specify the button's spacial properties, name, and visibility.
- In the Connections tab, specify the tags or expression with which the button exchanges data. For information about assigning tags and expressions, see page 15-42.

For details about the options in the tabs, see Help.

3. When you are finished, click OK.

Setting up maintained push buttons

Use the maintained push button to change a setting in a machine or process. Maintained push buttons are not useful for starting or stopping a machine or process.

When pressed the first time, the maintained push button changes a tag to one value. When pressed and released a second time, the button changes the tag to another value.

The error state

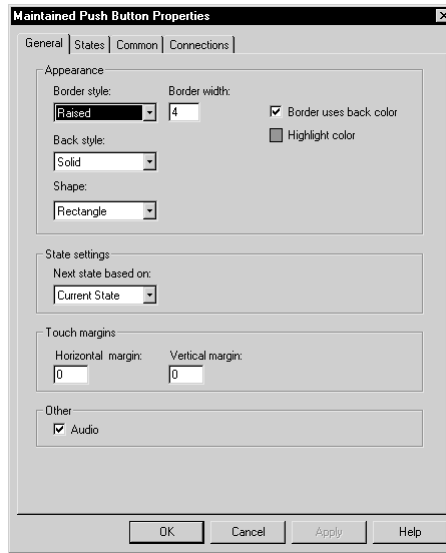
The button's error state is displayed at run time when:

- the Value control is unassigned.
- the Indicator control's value does not match one of the state values you set up.

The error state is also displayed when the display containing the maintained push button first opens, if the Value control's value does not match one of the state values you set up.

To set up a maintained push button

1. Double-click the maintained push button.



2. In the button's Properties dialog box, fill in the options on each tab:

- In the General tab, specify what the button looks like at run time, and how the button changes state.
- In the States tab, specify what the button does when it is pressed and released. For tips about setting up states, see page 16-3.
- In the Common tab, specify the button's spacial properties, name, and visibility.
- In the Connections tab, specify the tags or expression with which the button exchanges data. For information about assigning tags and expressions, see page 15-42.

For details about the options in the tabs, see Help.

3. When you are finished, click OK.

Setting up latched push buttons

The latched push button latches in the On position, and must be unlatched by another button or process to return to the Off position. This type of button is useful for starting a machine or process.

For example, use the latched push button when you want the Value control to start a process within a programmable controller and remain set until the process is completed. You can also use a latched push button when you have a controller with a long program or long update times.

When the operator presses a latched push button, it changes the Value control to one value, and remains at that value until the Handshake control gives the signal to unlatch the button.

The error state

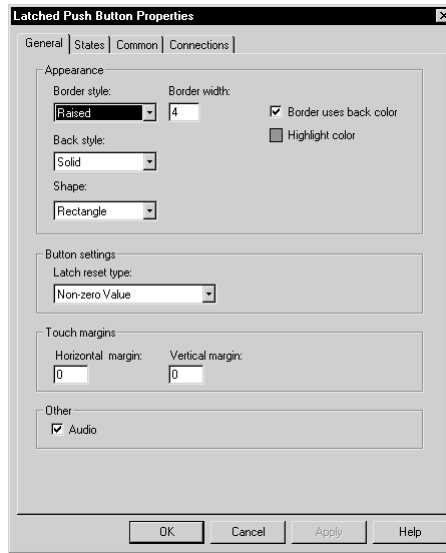
The button's error state is displayed at run time when:

- the Value control is unassigned.
- the Indicator control's value does not match one of the state values you set up.
- The Handshake control is unassigned.

The error state is also displayed when the display containing the latched push button first opens, if the Value control's value does not match one of the state values you set up.

To set up a latched push button

1. Double-click the latched push button.



2. In the button's Properties dialog box, fill in the options on each tab:

- In the General tab, specify what the button looks like at run time, and how to unlatch it.
- In the States tab, specify what the button does when it is latched and unlatched. For tips about setting up states, see page 16-3.
- In the Common tab, specify the button's spacial properties, name, and visibility.
- In the Connections tab, specify the tags or expression with which the button exchanges data. For information about assigning tags and expressions, see page 15-42.

For details about the options in the tabs, see Help.

3. When you are finished, click OK.

Setting up multistate push buttons

The multistate push button displays—and allows an operator to cycle through—multiple options consecutively. The multistate push button displays the current state of a process or operation by showing a different color, caption, or image to reflect different states.

Each time the operator presses the button, the tag changes to the value for the next state. When the button is in its last state and the operator presses the button, the button returns to its first state.

The error state

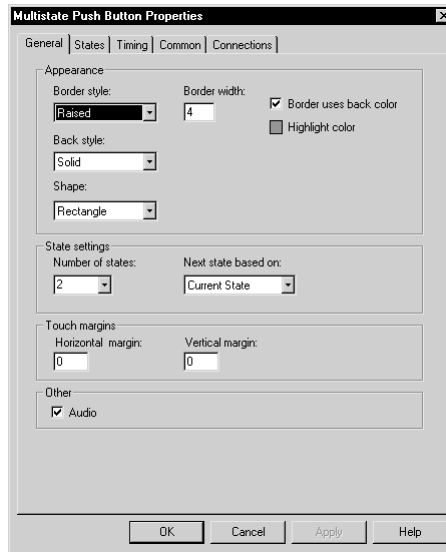
The button's error state is displayed at run time when:

- the Value control is unassigned.
- the Indicator control's value does not match one of the state values you set up.

The error state is also displayed when the display containing the multistate push button first opens, if the Value control's value does not match one of the state values you set up.

To set up a multistate push button

1. Double-click the multistate push button.



2. In the button's Properties dialog box, fill in the options on each tab:
 - In the General tab, specify what the button looks like, and how the button changes state.
 - In the States tab, specify what the button does when it is pressed and released. For tips about setting up states, see page 16-3.
 - In the Timing tab, set up whether or not the button press repeats automatically when the operator presses and holds the button down. You can also set up the rate at which the button press repeats. For more information about auto repeat, see page 16-14.
 - In the Common tab, specify the button's spacial properties, name, and visibility.

- In the Connections tab, specify the tags or expression with which the button exchanges data. For information about assigning tags and expressions, see page 15-42.

For details about the options in the tabs, see Help.

3. When you are finished, click OK.

Setting up interlocked push buttons

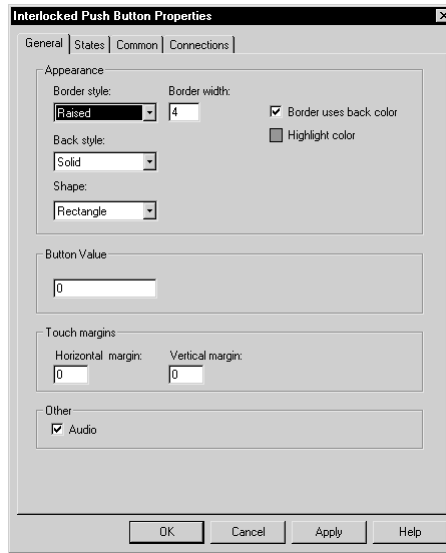
Multiple interlocked push buttons work together and share the same tag. Pressing one button cancels another. Although interlocked push buttons work as a group, you add them to the display one at a time.

When the operator presses one of the interlocked push buttons, the tag assigned to its Value control changes to one value. When the operator presses a different interlocked push button, the tag changes to another value. Assign the same tag to each button's Value control.

You can also use a single interlocked push button to send a value to a tag.

To set up an interlocked push button

1. Double-click the interlocked push button.



2. In the button's Properties dialog box, fill in the options on each tab:
 - In the General tab, specify the button's appearance, and the value it sends to the Value control.
 - In the States tab, specify what the button looks like when it is pressed and released at run time. For tips about setting up states, see page 16-3.
 - In the Common tab, specify the button's spacial properties, name, and visibility.
 - In the Connections tab, specify the tag with which the button exchanges data. Interlocked push buttons have no Indicator control. For information about assigning tags, see page 15-42.

For details about the options in the tabs, see Help.

3. When you are finished, click OK.

Setting up ramp buttons

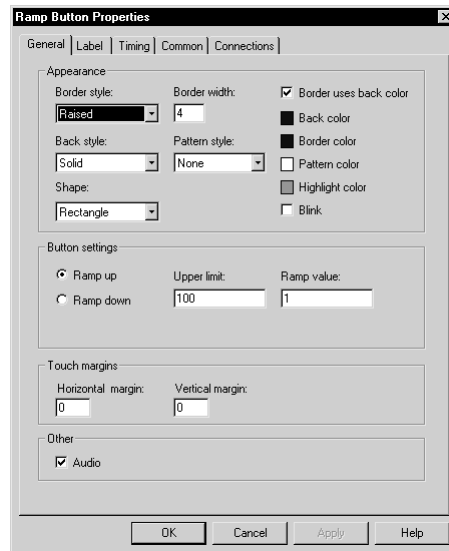
Use the ramp button to increase or decrease the value of a tag.

Ramp buttons can change a tag by either an integer or floating-point value. You can use two ramp buttons together to create an increment/decrement control.

Each time the operator presses the button, the tag value increases or decreases, depending on how you set up the button.

To set up a ramp button

1. Double-click the ramp button.



2. In the button's Properties dialog box, fill in the options on each tab:

- In the General tab, specify what the button looks like, and whether the button ramps a value up or down.
- In the Label tab, specify what text or image to display on the button.

- In the Timing tab, set up whether or not the button press repeats automatically when the operator presses and holds the button down. You can also set up the rate at which the button press repeats. For more information about auto repeat, see page 16-14.
- In the Common tab, specify the button's spacial properties, name, and visibility.
- In the Connections tab, specify the tag with which the button exchanges data. Ramp buttons have no Indicator control. For information about assigning tags, see page 15-42.

For details about the options in the tabs, see Help.

3. When you are finished, click OK.

Setting up numeric displays

Use the numeric display object to show the operator numeric information from the data source. For example, you might use a numeric display to show the current temperature of an oven.

How values are displayed

The numeric display shows the value of the Value control at the data source. The value shown depends on whether the Value control value is a floating-point number or an integer. Integer values are displayed as is. Floating-point values are rounded to fit the display. The decimal places option also affects how floating point numbers are displayed.

For example, if the numeric display is set up to show 6 digits, with one decimal place, 1234.56 is rounded to 1234.6. 1234.44 is rounded to 1234.4. The decimal counts as one of the digits.

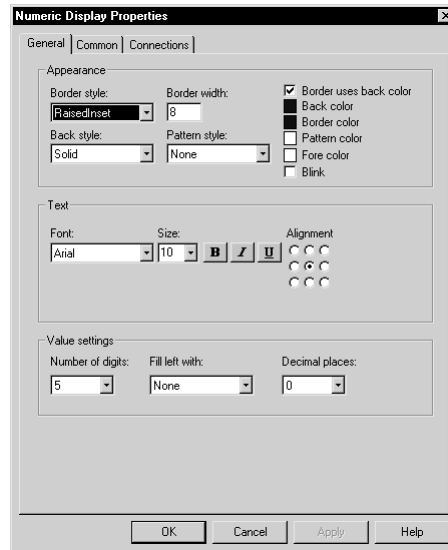
For more information about how values are rounded, see page 7-2.

Problems with displaying values

- If the Value control is unassigned, the numeric display is filled with asterisks (*).
- If the integer portion of the value, including the decimal point and minus sign, contains more digits than specified for the display, the numeric display is filled with asterisks.
- If the numeric display is sized so that the value cannot be fully displayed, the value is truncated and the last displayable digit is replaced with an asterisk.
- If the value doesn't fit on the first line of the display, and there is room for a second line, the value continues onto the second line.

To set up a numeric display

1. Double-click the numeric display.



2. In the object's Properties dialog box, fill in the options on each tab:
 - In the General tab, specify what the numeric display looks like at run time.

- In the Common tab, specify the object's spacial properties, name, and visibility.
- In the Connections tab, specify the tags or expression whose data is displayed. For information about assigning tags and expressions, see page 15-42.

For details about the options in the tabs, see Help.

3. When you are finished, click OK.

Setting up numeric input enable buttons

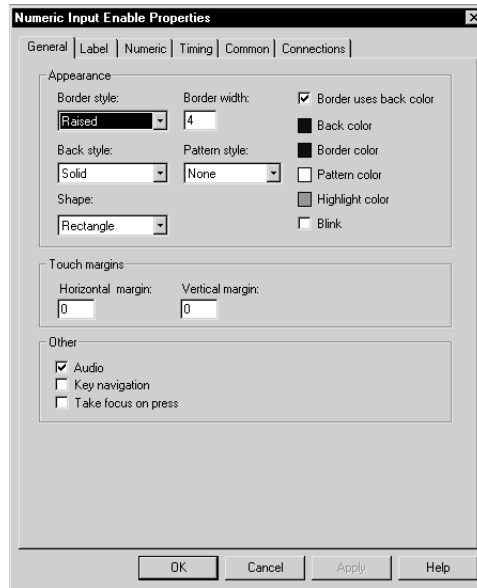
The operator can press the numeric input enable button to open a numeric pop-up keypad or scratchpad. The operator can enter a number in the keypad or scratchpad, and then send the number to the data source.

For more information about using the numeric input enable button at run time, see page 26-5.

You can also set up the numeric input enable button to work as a ramp button. In this case, when the button has focus, the operator can press a move up or move down button to change a tag by either an integer or floating-point value. The operator can also press the Up Arrow or Down Arrow on the keyboard or keypad to ramp the value.

To set up a numeric input enable button

1. Double-click the numeric input enable button.



2. In the button's Properties dialog box, fill in the options on each tab:

- In the General tab, specify what the button looks like at run time, whether the operator can navigate to the button using the keys on the keyboard or keypad, and whether the operator can press the button to give it focus.
- In the Label tab, specify what text or image to display on the button.
- In the Numeric tab, set up which pop-up window opens (if any), the ramp value (if any), minimum and maximum values to send to the data source, and decimal point settings.
- In the Timing tab, set up the timing and handshake settings for the Enter key. These settings do not apply when you ramp a value. For information about using Enter key handshaking, see page 16-16.

- In the Common tab, specify the button's spacial properties, name, and visibility.
- In the Connections tab, specify the tags or expressions with which the button exchanges data. For information about assigning tags and expressions, see page 15-42.

For details about the options in the tabs, see Help.

3. When you are finished, click OK.

Setting up numeric input cursor points

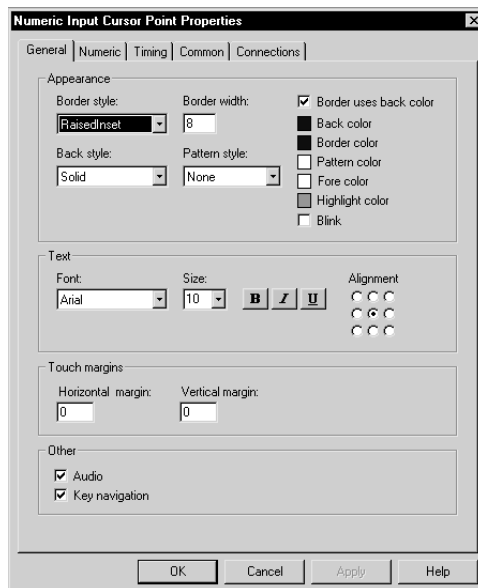
The operator can activate the numeric input cursor point to open a numeric pop-up keypad or scratchpad. The operator can enter a number in the keypad or scratchpad, and then send the number to the data source. The cursor point displays the current value at the data source (if you assign a tag to the Indicator control).

For more information about using the numeric input cursor point at run time, see page 26-5.

You can also set up the numeric input cursor point to work as a ramp button. In this case, when the cursor point has focus, the operator can press a move up or move down button to change a tag by either an integer or floating-point value. The operator can also press the Up Arrow or Down Arrow on the keyboard or keypad to ramp the value.

To set up a numeric input cursor point

1. Double-click the numeric input cursor point.



2. In the object's Properties dialog box, fill in the options on each tab:

- In the General tab, specify what the cursor point looks like at run time, and whether the operator can navigate to the cursor point using the keys on the keyboard or keypad.
- In the Numeric tab, set up which pop-up window opens (if any), the ramp value (if any), minimum and maximum values to send to the data source, decimal point settings, and display settings.
- In the Timing tab, set up the timing and handshake settings for the Enter key. These settings do not apply when you ramp values. For information about using Enter key handshaking, see page 16-16.
- In the Common tab, specify the object's spacial properties, name, and visibility.
- In the Connections tab, specify the tags or expression with which the numeric input cursor point exchanges data. For

information about assigning tags and expressions, see page 15-42.

For details about the options in the tabs, see Help.

3. When you are finished, click OK.

Setting up string displays

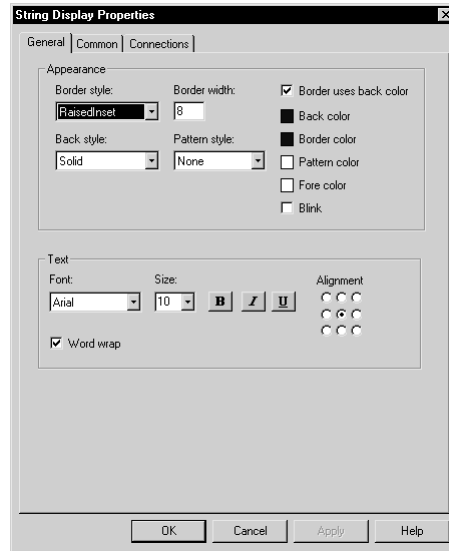
Use the string display object to show the operator messages from the data source. For example, you might set up the data source to generate strings that report on the state of a process or operation, or that provide instructions about what the operator needs to do next. At run time the display shows the operator the string value of the Value control at the data source.

How values are displayed

- If the Value control is unassigned, the string display is blank.
- RSVView displays the number of characters (bytes) specified for the tag assigned to the Value control.

To set up a string display

1. Double-click the string display.



2. In the object's Properties dialog box, fill in the options on each tab:
 - In the General tab, specify what the string display looks like at run time.
 - In the Common tab, specify the string display's spacial properties, name, and visibility.
 - In the Connections tab, specify the tag or expression whose data is displayed. For information about assigning tags and expressions, see page 15-42.

For details about the options in the tabs, see Help.

3. When you are finished, click OK.

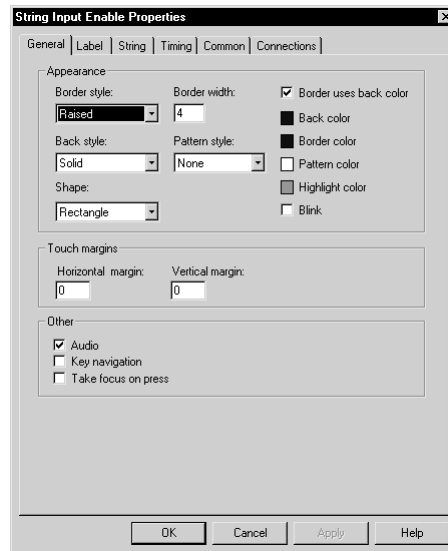
Setting up string input enable buttons

The operator can press the string input enable button to open a string pop-up scratchpad or keyboard. The operator can enter text in the scratchpad or keyboard, and then send the string to the data source.

For more information about using the string input enable button at run time, see page 26-5.

To set up a string input enable button

1. Double-click the string input enable button.



2. In the button's Properties dialog box, fill in the options on each tab:
 - In the General tab, specify what the button looks like at run time, whether the operator can navigate to the button using the keys on the keyboard or keypad, and whether the operator can press the button to give it focus.
 - In the Label tab, specify what text or image to display on the button.

- In the String tab, specify the pop-up to open and the number of characters to accept.
- In the Timing tab, set up the timing and handshake settings for the Enter key. For information about using Enter key handshaking, see page 16-16.
- In the Common tab, specify the button's spacial properties, name, and visibility.
- In the Connections tab, specify the tags or expression with which the button exchanges data. For information about assigning tags and expressions, see page 15-42.

For details about the options in the tabs, see Help.

3. When you are finished, click OK.

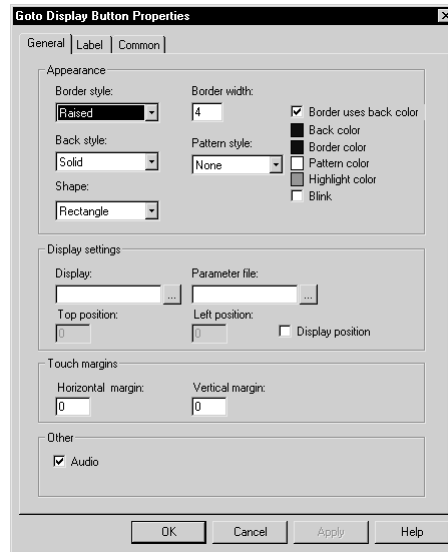
Setting up goto display buttons

Use the goto display button to open a graphic display. You can create as many goto display buttons as you need, but each button opens a single display only.

For more information about using goto display buttons, see page 22-5.

To set up a goto display button

1. Double-click the goto display button.



2. In the button's Properties dialog box, fill in the options on each tab:
 - In the General tab, specify what the button looks like at run time, the display to open, and the parameter file to use, if any.
 - In the Label tab, specify what text or image to display on the button.
 - In the Common tab, specify the button's spacial properties, name, and visibility.

For details about the options in the tabs, see Help.

3. When you are finished, click OK.

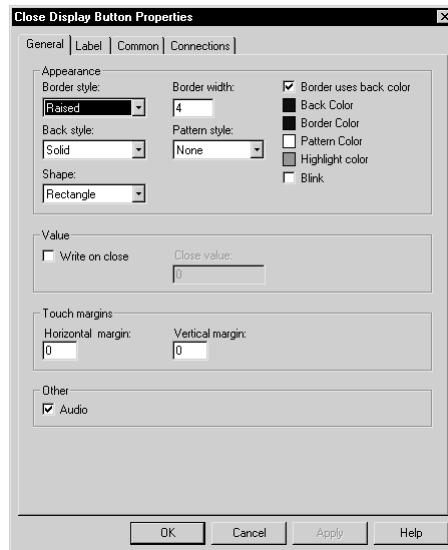
Setting up close display buttons

When the operator presses a close display button at run time, the graphic display that the button is on closes. You can set up the button to write out a value when the display closes.

For more information about using close display buttons, see page 22-7.

To set up a close display button

1. Double-click the button.



2. In the button's Properties dialog box, fill in the options on each tab:
 - In the General tab, specify what the button looks like at run time, and whether to write out a value when the display closes.
 - In the Label tab, specify what text or image to display on the button.
 - In the Common tab, specify the button's spacial properties, name, and visibility.

- In the Connections tab, specify the tag with which the button exchanges data. For information about assigning tags, see page 15-42.

For details about the options in the tabs, see Help.

3. When you are finished, click OK.

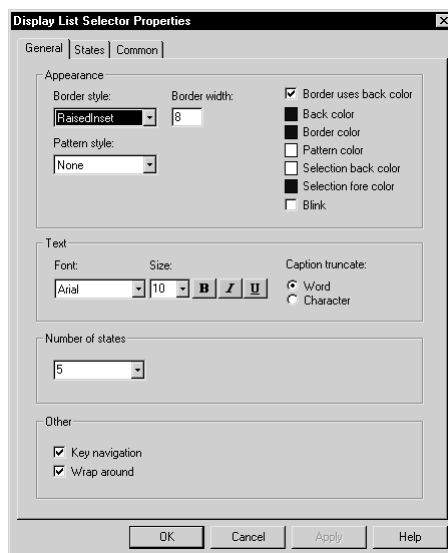
Setting up display list selectors

The display list selector is a list of graphic displays in the application. Each graphic display is represented by a different state in the display list selector. An operator can scroll through the list of displays, and then select the display to open by pressing an enter button or Enter key.

For more information about using display list selectors, see page 22-8.

To set up a display list selector

1. Double-click the display list selector.



2. In the object's Properties dialog box, fill in the options on each tab:

- In the General tab, specify what the button looks like at run time, its number of states, whether the operator can navigate to the button using the keys on the keyboard or keypad, and whether the cursor wraps from the bottom of the list back to the top.
- In the States tab, specify which graphic display to use for each of the display list selector's states, and how each graphic display is named in the list. For tips about setting up states, see page 16-3.
- In the Common tab, specify the display list selector's spacial properties, name, and visibility.

For details about the options in the tabs, see Help.

3. When you are finished, click OK.

How to use indicators

Indicators display the status of processes or operations by showing different colors, captions, or images to reflect different states.

You can create different kinds of indicators to suit your needs:

Multistate indicators display the current state of a process or operation by showing a different color, caption, or image to reflect different states.

Symbols display a symbol that changes color to indicate the state of a process or operation. This allows the operator to see the status of a process or operation at a glance.

List indicators display a list of states for a process or operation, and highlight the current state. Each state is represented by a caption in the list. This type of indicator is useful if you want operators to view the current state, but also see the other possible states. For sequential processes, the list can inform the operator about what happens next.

Setting up multistate indicators

The multistate indicator displays the current state of a process or operation by showing a different color, caption, or image for each state.

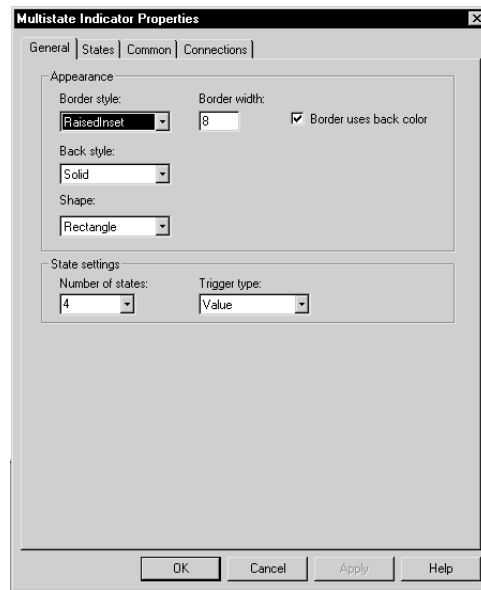
The error state

The multistate indicator's error state is displayed at run time when:

- the Indicator control is unassigned.
- the Indicator control's value does not match one of the state values you set up.
- the Trigger type is set to Value and the Indicator control value does not match one of the state values you set up.
- the Trigger type is set to LSB and the position of the least significant bit set in the Value control does not match one of the state values you set up.
- the Trigger type is set to Value and an array tag has been assigned to the object's Value control.

To set up a multistate indicator

1. Double-click the multistate indicator.



2. In the object's Properties dialog box, fill in the options on each tab:

- In the General tab, specify what the indicator looks like, and the number of states for the indicator.
- In the States tab, specify how the indicator's appearance changes when its tag or expression's value changes. For tips about setting up states, see page 16-3.
- In the Common tab, specify the indicator's spacial properties, name, and visibility.
- In the Connections tab, specify the tag or expression from which the indicator receives data. For information about assigning tags and expressions, see page 15-42.

For details about the options in the tabs, see Help.

3. When you are finished, click OK.

Setting up symbols

The symbol is an indicator that displays a single monochrome image that changes color to match the state of a process or operation. This allows the operator to see the status of a process or operation at a glance.

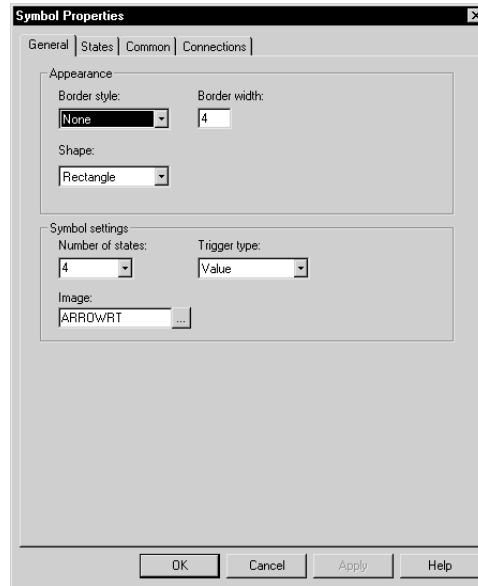
The error state

The symbol's error state is displayed at run time when:

- the Indicator control is unassigned.
- the Indicator control's value does not match one of the state values you set up.
- the Trigger type is set to Value and the Indicator control value does not match one of the state values you set up.
- the Trigger type is set to LSB and the position of the least significant bit set in the Value control does not match one of the state values you set up.
- the Trigger type is set to Value and an array tag has been assigned to the object's Value control.

To set up a symbol

1. Double-click the symbol.



2. In the object's Properties dialog box, fill in the options on each tab:

- In the General tab, specify what the symbol looks like at run time, and the number of states for the symbol.
- In the States tab, specify how the symbol's appearance changes when its tag or expression's value changes. For tips about setting up states, see page 16-3.
- In the Common tab, specify the symbol's spacial properties, name, and visibility.
- In the Connections tab, specify the tag or expression from which the symbol receives data. For information about assigning tags and expressions, see page 15-42.

For details about the options in the tabs, see Help.

3. When you are finished, click OK.

Setting up list indicators

The list indicator displays a list of states for a process or operation, and highlights the current state.

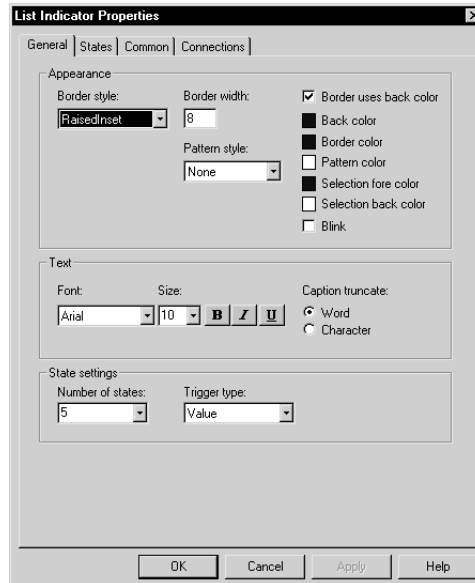
Each state is represented by a caption in the list. This type of indicator is useful if you want operators to view the current state, but also see the other possible states.

For sequential processes, the list can inform the operator about what happens next.

The list indicator has no error state. If the value of the Indicator control does not match any of the available states, none of the states is highlighted.

To set up a list indicator

1. Double-click the list indicator.



2. In the object's Properties dialog box, fill in the options on each tab:

- In the General tab, specify what the indicator looks like at run time, and the number of states for the indicator.
- In the States tab, specify how the indicator's appearance changes when its tag or expression's value changes. For tips about setting up states, see page 16-3.
- In the Common tab, specify the indicator's spacial properties, name, and visibility.
- In the Connections tab, specify the tag or expression from which the indicator receives data. For information about assigning tags and expressions, see page 15-42.

For details about the options in the tabs, see Help.

3. When you are finished, click OK.

How to use bar graphs, gauges, and scales

Bar graphs and gauges display graphical representations of numeric values. The scale is used with bar graphs to indicate the range of values for the bar graph.

Bar graphs make it easy to compare values

Bar graphs display numeric values in bar graph format. They are useful for allowing comparisons between multiple values, or for representing the fill levels of tanks for which a reading on a vertical scale is appropriate.

For example, one bar graph can show the required level of a tank, and a second bar graph can show the actual level of the tank. The first graph can change to represent the required level for a particular application, and the second graph changes as the actual level in the tank rises or drops.

Bar graphs are more useful than numeric displays when it's important for the operator to analyze the relationships between numeric values. It's easier for the operator to see that one graph is at a lower level than the other, or that one's fill is blue and the other's is yellow, than it is to subtract one numeric value from another.

Thresholds change a bar graph's fill color

As it fills, a bar graph can change its fill color to help an operator recognize abnormal conditions. The change in color happens when the tag (or expression) value crosses a threshold you set up for the graph. For example:

- If the temperature of an oven is lower than required for a recipe, the bar graph can show the temperature in blue.
- If the temperature is in the correct range for the recipe, the bar graph can show the temperature in green.
- If the temperature is higher than the recipe allows, the bar graph can show the temperature in red.

If you want to use a different fill color on a bar graph to show abnormal conditions, make sure enough of the new fill color is visible when the abnormal condition occurs that the operator can see the condition.

Use bar graphs with scales to show limits

Unlike gauges, bar graphs do not have integrated scales. You can show values on a bar graph using a scale graphic object and text.

Scales consist of major ticks, represented by long lines, and minor ticks, represented by short lines. To indicate the values of major or minor ticks, use text objects.

Gauges make it easy to see limits

Gauges display numeric values in dial format. They are useful for displaying a value in relation to its lower and upper limits.

For example, a temperature gauge shows the current temperature in relation to its minimum and maximum extremes. By looking at the position of the needle on the gauge (pointing left, up, or right), the operator can tell at a glance whether the temperature is nearer its lower limit, nearer the middle, or nearer its upper limit.

Gauges are used instead of numeric displays when it's important for the operator to recognize an abnormal condition instantly, either from far away when the scale on the gauge isn't visible, or before the operator has had the opportunity to determine the exact reading on the gauge. This characteristic of gauges is one of the reasons why they are used in automobile instrumentation.

Thresholds change a gauge's fill color

As the needle sweeps higher on a gauge, the gauge can fill the area behind the needle with a color. The gauge can change its fill color to help an operator recognize abnormal conditions. The change in color happens when the tag (or expression) value crosses a threshold you set up for the gauge. For example:

- If the temperature of an oven is lower than required for a recipe, the gauge can show the temperature in blue.
- If the temperature is in the correct range for the recipe, the gauge can show the temperature in green.
- If the temperature is higher than the recipe allows, the gauge can show the temperature in red.

IMPORTANT

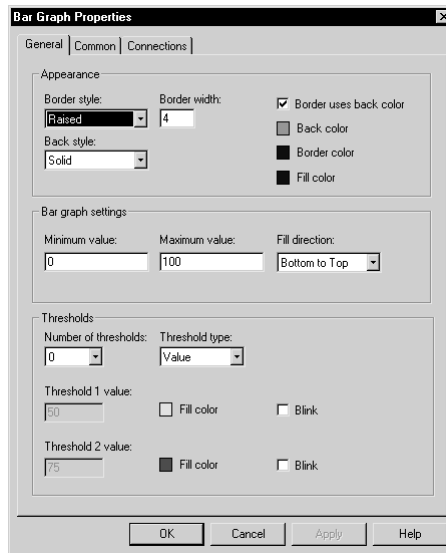
Some people are color blind to red and green, so don't rely on color alone to establish meaning.

Setting up bar graphs

Use bar graphs to represent numeric values by filling and emptying the object as the values rise and fall.

To set up a bar graph

1. Double-click the bar graph.



2. In the object's Properties dialog box, fill in the options on each tab:

- In the General tab, specify what the bar graph looks like at run time.
- In the Common tab, specify the bar graph's spacial properties, name, and visibility.
- In the Connections tab, specify the tag or expression from which the bar graph receives data. For information about assigning tags and expressions, see page 15-42.

For details about the options in the tabs, see Help.

3. When you are finished, click OK.

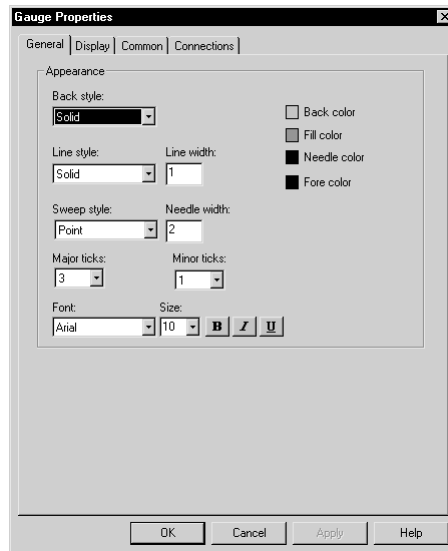
Setting up gauges

The gauge graphic object represents numeric values using a needle on a dial.

At run time, the gauge indicates the value of a tag or expression in relation to the gauge's minimum and maximum values.

To set up a gauge

1. Double-click the gauge.



2. In the object's Properties dialog box, fill in the options on each tab:

- In the General tab, specify what the gauge looks like at run time.
- In the Display tab, specify how the gauge displays values at run time.
- In the Common tab, specify the gauge's spacial properties, name, and visibility.

- In the Connections tab, specify the tag or expression from which the gauge receives data. For information about assigning tags and expressions, see page 15-42.

For details about the options in the tabs, see Help.

3. When you are finished, click OK.

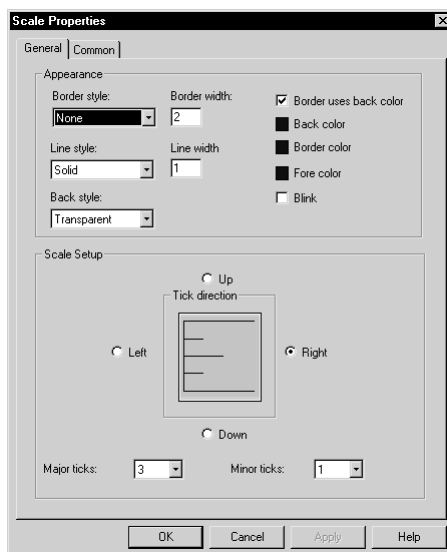
Setting up scales

Use the scale graphic object to show the possible range of values for a bar graph.

To place value labels on the scale (to create a legend), use text objects. Create the text objects and then position them next to the tick marks on the scale. Group all the objects with the bar graph.

To set up a scale

1. Double-click the scale.



2. In the object's Properties dialog box, fill in the options on each tab:

- In the General tab, specify what the scale looks like at run time.
- In the Common tab, specify the scale's spacial properties, name, and visibility.

For details about the options in the tabs, see Help.

3. When you are finished, click OK.

Setting up control list selectors

Control list selectors allow an operator to scroll through a list of states for a process and select one of the states. A highlight in the list shows the current state.

A control list selector can show several states at the same time, but only one state can be selected at a time. As the operator scrolls through the list, each successive state is selected automatically. If you want the operator to confirm the selection of a particular state before the state's value is written to the programmable controller, create an enter button on the same display as the control list selector, and select the "Write on enter" option.

The operator can scroll through the control list selector using:

- key button graphic objects. These are graphic objects that duplicate the functions of keyboard keys.
- the arrow keys and Enter key on a terminal's keypad.
- the arrow keys and Enter key on a keyboard.

Using buttons with the control list selector

The operator presses the buttons to scroll up or down the list, or to make selections from the list. The buttons can be set up to work with

the control list selector that has input focus, or you can link the buttons to a specific control list selector.

Use this button	To do this
Backspace	Move the cursor back to the highlighted item in the list.
End	Move to the bottom item in the list.
Enter	Select the item the cursor is pointing to.
Home	Move to the top item in the list.
Move down	Move down one item in the list.
Move up	Move up one item in the list.
Page down	Move down one page in the list.
Page up	Move up one page in the list.

How Enter key handshaking works

When the operator selects a state in the control list selector and presses an enter button or Enter key, the highlighted state’s value is written to the programmable controller or device.

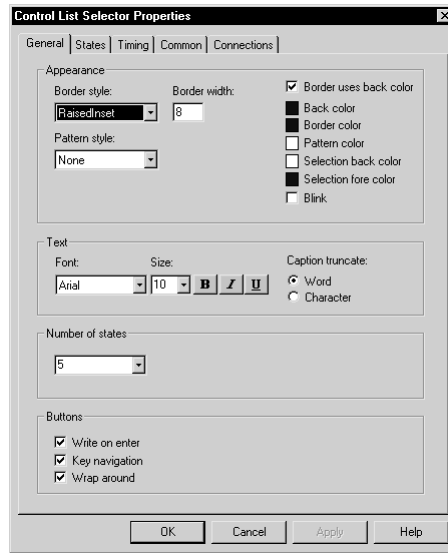
You can use Enter key handshaking to hold the value of the tag at the programmable controller or device for a specific period of time, to ensure the value is read before the control list selector overwrites the value with a new value.

Enter key handshaking only works if the Write on enter check box is selected (in the General tab).

For more information about using Enter key handshaking, see page 16-16.

To set up a control list selector

1. Double-click the control list selector.



2. In the object's Properties dialog box, fill in the options on each tab:

- In the General tab, specify what the button looks like at run time, its number of states, whether to write out a value when the operator presses the Enter key, whether the operator can navigate to the button using the keys on the keyboard or keypad, and whether the cursor wraps from the bottom of the list back to the top.
- In the States tab, set up the value and caption for each state. The value defines the control list selector's action, by changing the Value control to the value you specify when the selected state is in effect. For tips about setting up states, see page 16-3.
- In the Timing tab, set up the timing and handshake settings for the Enter key. For information about using Enter key handshaking, see page 16-16.
- In the Common tab, specify the control list selector's spacial properties, name, and visibility.

- In the Connections tab, specify the tags or expression with which the control list selector exchanges data. For information about assigning tags and expressions, see page 15-42.

For details about the options in the tabs, see Help.

3. When you are finished, click OK.

Setting up piloted control list selectors

Piloted control list selectors allow an operator or remote device to scroll through a list of states for a process and select one of the states. A highlight in the list shows the current state.

Choosing between piloted control list selectors and control list selectors

Piloted control list selectors include the following features that control list selectors do not:

- The states can be selected either by an operator, or remotely, for example by a programmable controller, or by both an operator and a remote device. Individual states can be turned off, to prevent them from being selected by either the operator or the remote device.

Control list selectors do not allow states to be turned off.

- The state values of all of the items that are visible in the list can be written to the Visible States control when the list scrolls. The Visible States control must be a data-server tag that supports arrays. You cannot use an HMI tag.

Control list selectors have no Visible States control.

- The state value of the item at the top of the list can be written to the Top Position control when the list scrolls.

Control list selectors have no Top Position control.

How piloted control list selectors work at run time

A piloted control list selector can show several states at the same time, but only one state can be selected at a time.

You can set up the piloted control list selector to be operator-controlled or remote-controlled by assigning tags or expressions in the Connections tab.

You can also set up individual states to be operator-controlled, remote-controlled, both, or none. If set to none, the state is turned off.

If the piloted control list selector is set up to be operator-controlled, and the operator selects a remote-controlled state, or one that is turned off, a hollow cursor is displayed.



The operator can select this state.



The operator cannot select this state.

Using buttons with the piloted control list selector

When a piloted control list selector is operator-controlled, it works with:

- key button graphic objects. These are graphic objects that duplicate the functions of keyboard keys.
- the arrow keys and Enter key on a terminal's keypad.
- the arrow keys and Enter key on a keyboard.

The operator presses the buttons to scroll up or down the list, or to make selections from the list. The buttons can be set up to work with

the piloted control list selector that has focus, or with a specific piloted control list selector. You can use the buttons listed in the table on page 16-61.

Selecting items in the list

Each state can be selected directly by an operator, or remotely by a device such as a programmable controller. The operator or the controller scrolls through the list to select a different state or a different group of visible states.

When an operator or remote device selects a state, the value assigned to the selected state is written to the piloted control list selector's Value control. If the state is turned off, the state's value is not written to the Value control.

If the operator attempts to select a state that is remote-controlled, the state's value is not written to the Value control.

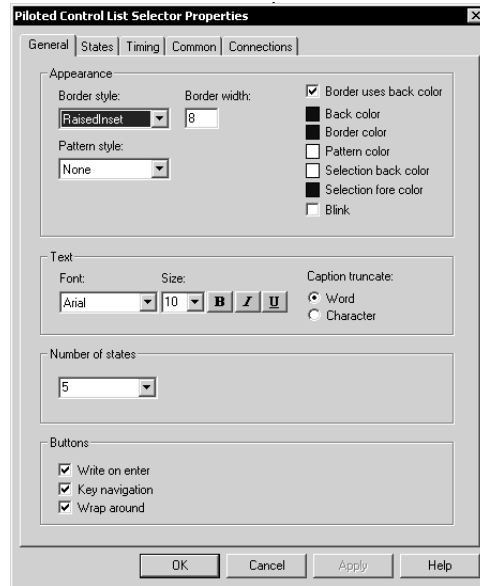
Scrolling through the list

If the piloted control list selector contains more states than can be displayed in the list simultaneously, the value of the Top Position control (if assigned) changes whenever the item at the top of the list changes.

If the Visible States control is assigned, the values assigned to all visible states are written to the Visible States control whenever the list scrolls. To use this feature, the assigned tag must support arrays, and the array must be the same length as the number of visible states in the piloted control list selector. For information about using array tags, see page 9-18.

To set up a piloted control list selector

1. Double-click the piloted control list selector.



2. In the object's Properties dialog box, fill in the options on each tab:

- In the General tab, specify what the button looks like at run time, its number of states, whether to write out a value when the operator presses the Enter key, whether the operator can navigate to the button using the keys on the keyboard or keypad, and whether the cursor wraps from the bottom of the list back to the top.
- In the States tab, set up the value and caption for each state. Also specify whether each state can be selected, and by whom. For tips about setting up states, see page 16-3.
- In the Timing tab, set up the timing and handshake settings for the Enter key. For information about using Enter key handshaking, see page 16-16.
- In the Common tab, specify the piloted control list selector's spacial properties, name, and visibility.

- In the Connections tab, specify the tags or expression with which the piloted control list selector exchanges data. For information about assigning tags and expressions, see page 15-42.

For details about the options in the tabs, see Help.

3. When you are finished, click OK.

Setting up local message displays

Use local message displays to provide the operator with information about what to do next, or with information about a process. At run time, the local message display shows one message at a time.

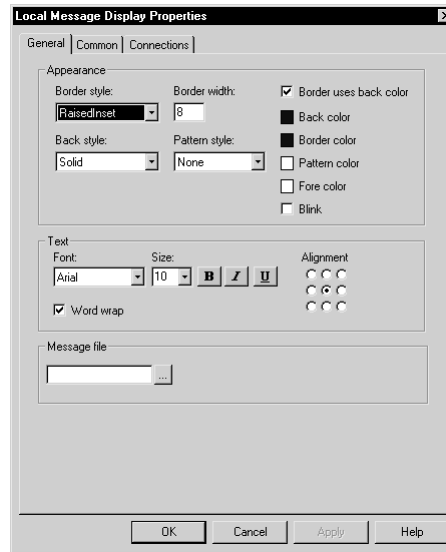
To use local messages, create a local message display object in a graphic display, and then assign a local message file to the local message display object.

For more information about local messages, see page 14-33.

If no message exists for the trigger value that matches the value of the Value control, the display is filled with question marks (?).

To set up a local message display

1. Double-click the local message display.



2. In the object's Properties dialog box, fill in the options on each tab:
 - In the General tab, specify what the local message display looks like at run time and which local message file to use.
 - In the Common tab, specify the local message display's spacial properties, name, and visibility.
 - In the Connections tab, specify the tag or expression with which the local message display exchanges data. For information about assigning tags and expressions, see page 15-42.

For details about the options in the tabs, see Help.

3. When you are finished, click OK.

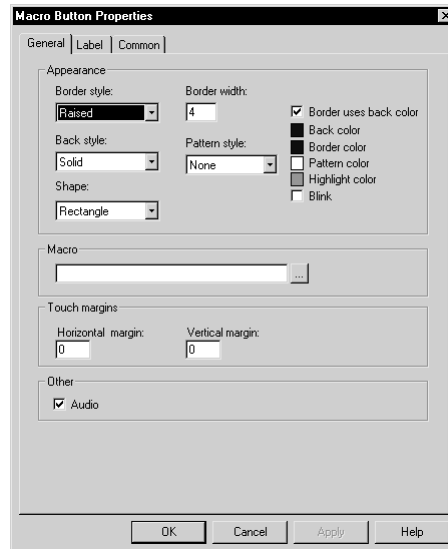
Setting up macro buttons

Macro buttons run macro files that assign values to tags when the button is pressed.

For information about creating macros, see Chapter 21.

To set up a macro button

1. Double-click the button.



2. In the button's Properties dialog box, fill in the options on each tab:
 - In the General tab, specify what the button looks like at run time, and the macro to run when the button is pressed.
 - In the Label tab, specify what text or image to display on the button.
 - In the Common tab, specify the button's spacial properties, name, and visibility.

For details about the options in the tabs, see Help.

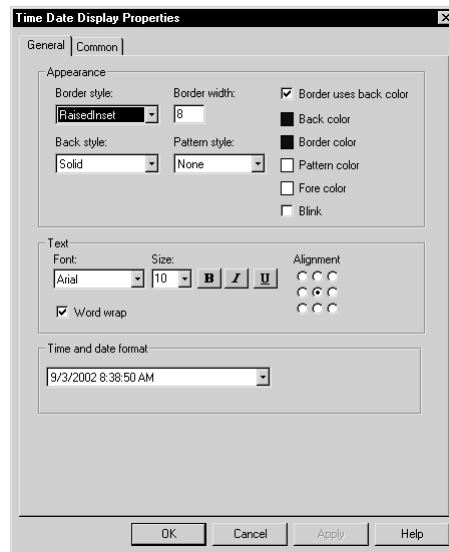
3. When you are finished, click OK.

Setting up time and date displays

A time and date display shows the current time and date in a graphic display.

To set up a time and date display

1. Double-click the time and date display.



2. In the object's Properties dialog box, fill in the options on each tab:
 - In the General tab, specify what the time and date display looks like at run time.
 - In the Common tab, specify the time and date display's spacial properties, name, and visibility.

For details about the options in the tabs, see Help.

3. When you are finished, click OK.

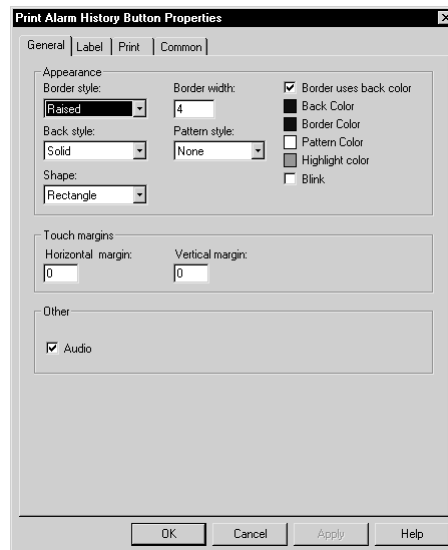
Setting up print alarm history buttons

The operator can press the print alarm history button to print a report of all the alarm messages in the alarm log file. The report can include the time alarms occurred and were acknowledged.

For information about the alarm log file, see page 9-11.

To set up a print alarm history button

1. Double-click the button.



2. In the button's Properties dialog box, fill in the options on each tab:
 - In the General tab, specify what the button looks like at run time.
 - In the Label tab, specify what text or image to display on the button.
 - In the Print tab, specify what information to print on the report.

- In the Common tab, specify the button's spacial properties, name, and visibility.

For details about the options in the tabs, see Help.

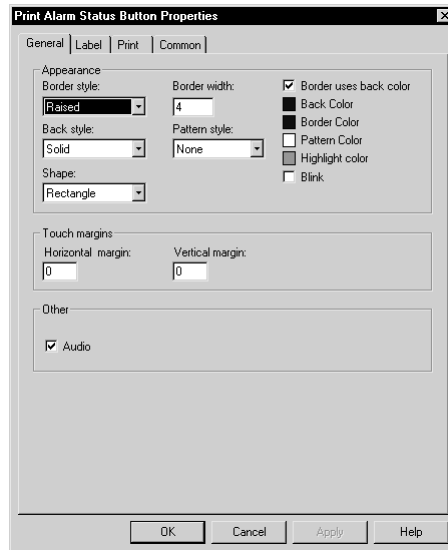
3. When you are finished, click OK.

Setting up print alarm status buttons

The operator can press the print alarm status button to print a report of the status of alarms that have been defined for the application (in the Alarm Setup editor). The report can include all alarms, active alarms only, or only alarms that have been active since the alarm status was last reset. The report can include how many times each alarm was triggered, and the accumulated time in alarm.

To set up a print alarm status button

1. Double-click the button.



2. In the button's Properties dialog box, fill in the options on each tab:

- In the General tab, specify what the button looks like at run time.
- In the Label tab, specify what text or image to display on the button.
- In the Print tab, specify what information to print on the report.
- In the Common tab, specify the button's spacial properties, name, and visibility.

For details about the options in the tabs, see Help.

3. When you are finished, click OK.

Setting up alarm lists

The alarm list graphic object displays a list of alarms from the alarm log file. It can show all the alarms in the alarm log file, or any combination of active, inactive, acknowledged, and unacknowledged alarms. It can also show the alarms for specific alarm triggers.

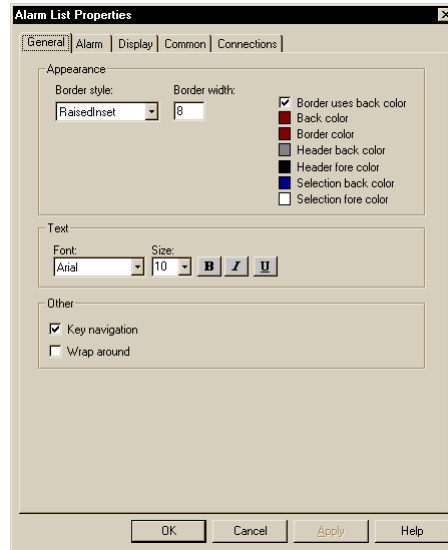
You can assign a tag or expression to the ActiveAcknowledged, ActiveUnacknowledged, InactiveAcknowledged, or InactiveUnacknowledged controls, to allow the data source to control the type of alarms to display in the list. When one of the control's values is a non-zero value, the related type of alarm is displayed in the alarm list. The control value overrides the setting for the list in the Alarm List Properties dialog box.

RSView comes with two graphic displays, [ALARM MULTI-LINE] and [HISTORY], that contain alarm list objects. The [HISTORY] display comes in two sizes. These displays are in the graphic library. You can copy the library displays into your application and use the displays and objects as is, customize them, or create your own.

For more information about using alarm lists, see page 9-37. For information about the alarm log file, see page 9-11.

To set up an alarm list

1. Double-click the alarm list.



2. In the object's Properties dialog box, fill in the options on each tab:
 - In the General tab, specify what the alarm list looks like at run time, whether the operator can navigate to the list using a keyboard, and how scrolling works for the list.
 - In the Alarm tab, specify the columns to include in the list, the number of lines per alarm, and the triggers to filter by, if any.
 - In the Display tab, specify the types of alarms to display in the list, and the appearance of the different alarm conditions.
 - In the Common tab, specify the alarm list's spacial properties, name, and visibility.

- In the Connections tab, specify the tags or expressions to use to control the type of alarms to display in the list. The use of these controls is optional.

For details about the options in the tabs, see Help.

3. When you are finished, click OK.

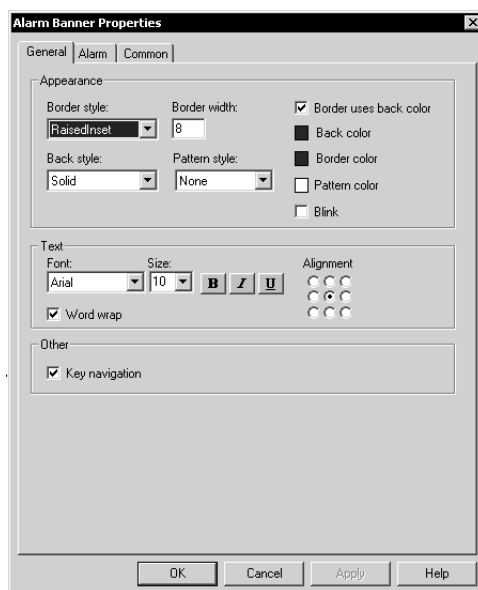
Setting up alarm banners

The alarm banner graphic object displays a single unacknowledged alarm message. For example, the banner could display a message that warns the operator that the pressure in a boiler is too high.

The [ALARM] display and the [ALARM BANNER] graphic library both contain an alarm banner object. You can use the displays and objects as is, customize them, or create your own. For more information about using alarm banners, see page 9-40.

To set up an alarm banner

1. Double-click the alarm banner.



2. In the object's Properties dialog box, fill in the options on each tab:

- In the General tab, specify what the button looks like at run time and whether the operator can navigate to the button using the keys on the keyboard or keypad.
- In the Alarm tab, set up whether to queue alarms, whether to display all alarms or active alarms only, and which alarms to include in a filtered trigger list.
- In the Common tab, specify the alarm banner's spacial properties, name, and visibility.

For details about the options in the tabs, see Help.

3. When you are finished, click OK.

Setting up alarm status lists

The alarm status list graphic object displays the status of alarms, including how many times an alarm has been triggered and how long it has been active. For example, the list could display an alarm that has been triggered 5 times for a total accumulated time in alarm of 10 minutes.

The operator can view the status list in three different modes:

- All alarms—displays the status of all alarms that have been defined in the Alarm Setup editor.
- Active alarms—displays the status of all the alarms that are currently active.
- Past alarms—displays the status of all the alarms that have been active since the alarm status has been reset.

The operator can press the alarm status mode button to cycle through the different modes.

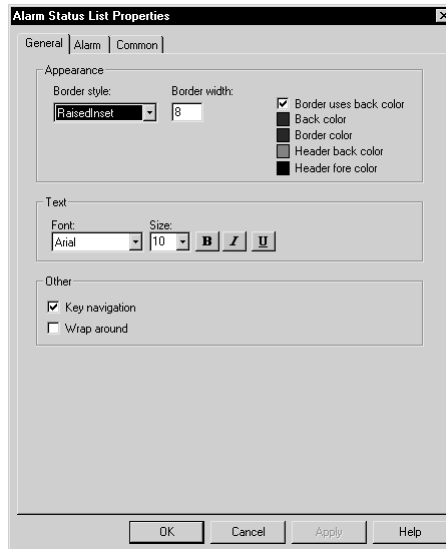
If desired, you can set up the alarm status list to show alarms for specific alarm triggers only.

RSView comes with a library graphic display called [STATUS] that contains an alarm status list object, alarm status mode button, and other buttons for working with the list. You can use the display and objects as is, customize them, or create your own.

For more information about using alarm status lists, see page 9-41.

To set up an alarm status list

1. Double-click the alarm status list.



2. In the object's Properties dialog box, fill in the options on each tab:
 - In the General tab, specify what the alarm status list looks like at run time, whether the operator can navigate to the list using a keyboard, and how scrolling works for the list.
 - In the Alarm tab, specify the columns to include in the list, the number of lines per alarm, and the triggers to filter by, if any.
 - In the Common tab, specify the alarm status list's spacial properties, name, and visibility.

For details about the options in the tabs, see Help.

3. When you are finished, click OK.

Setting up diagnostics lists

The diagnostics list graphic object displays runtime information about system activity:

- Information messages, such as messages about tag assignments, tag read and write activity, and macro usage.

These are system-generated information messages. They are not the same messages as the information messages you create and display in the information message display object.

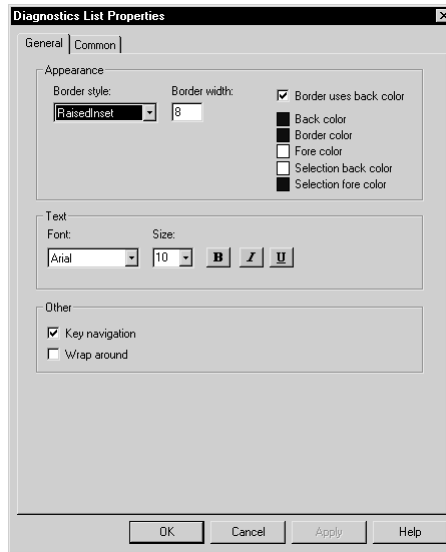
- Warning messages.
- Error messages, such as messages about communication errors, invalid values, and unassigned controls.
- Audit messages about tag writes and tag write failures.

RSView comes with a default graphic display called [DIAGNOSTICS] that contains a diagnostics list object. You can use the display and object as is, customize them, or create your own.

For more information about using diagnostics lists, see page 10-16.

To set up a diagnostics list

1. Double-click the diagnostics list.



2. In the object's Properties dialog box, fill in the options on each tab:
 - In the General tab, specify what the button looks like at run time, whether the operator can navigate to the button using the keys on the keyboard or keypad, and whether the cursor wraps from the bottom of the list back to the top.
 - In the Common tab, specify the diagnostics list's spacial properties, name, and visibility.

For details about the options in the tabs, see Help.

3. When you are finished, click OK.

Setting up information message displays

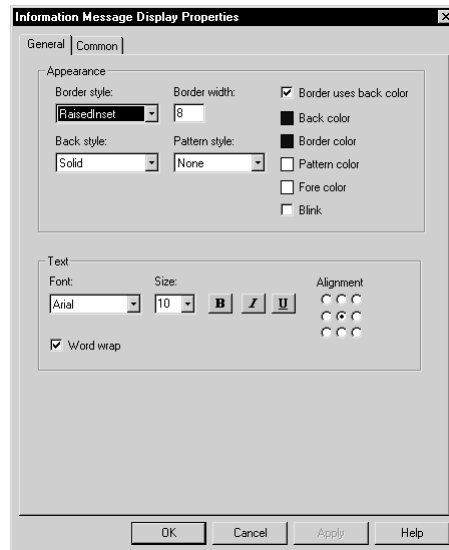
The information message display graphic object gives the operator runtime messages about information that requires immediate attention.

RSView comes with a default graphic display called [INFORMATION] that contains an information message display object. You can use the display and object as is, customize them, or create your own.

For more information about using information message displays, see page 12-10.

To set up an information message display

1. Double-click the information message display.



2. In the object's Properties dialog box, fill in the options on each tab:
 - In the General tab, specify what the information message display looks like at run time.

- In the Common tab, specify the information message display's spacial properties, name, and visibility.

For details about the options in the tabs, see Help.

3. When you are finished, click OK.

Animating graphic objects

This chapter describes:

- the types of animation and which objects support which types of animation.
- using the Animation dialog box.
- testing animation.
- using tag names, tag placeholders, and expressions when attaching animation.
- setting the minimum and maximum values for animation that uses a range of motion.
- using Object Smart Path™ to define an object's range of motion.
- setting up each type of animation.
- applying animation to groups.
- checking what kind of animation is attached to an object.
- copying and pasting animation.

Types of animation

Animation associates graphic objects with tags so the appearance or position of an object changes to reflect changes to the tag's value. For example, an object's color could change from yellow to orange to red as the tag's value increases. Or a slider could move from left to right as a tag's value increases.

You can use these types of animation:

- color
- fill
- height
- horizontal position
- horizontal slider
- rotation
- vertical position
- vertical slider
- visibility
- width

The Objects 5 Screen Demo sample application contains many examples of animation. In particular, see the graphic displays called “Animation I” and “Animation II.”

Which objects can have which types of animation?

This table summarizes which types of objects support which types of animation. For information about the different types of graphic objects, see page 15-2.

These objects	Support these types of animation
Drawing objects, except images, panels, and rounded rectangles	All types
Rounded rectangles	All types except rotation
All other objects	Visibility

You can also attach animation to groups of drawing objects. For more information, see page 17-20.

You can attach as many types of animation to a drawing object (except images and panels) as you like. For example, apply width, height, horizontal position, and vertical position animation to an object to give it the appearance of moving into or out of the display as it shrinks and grows.

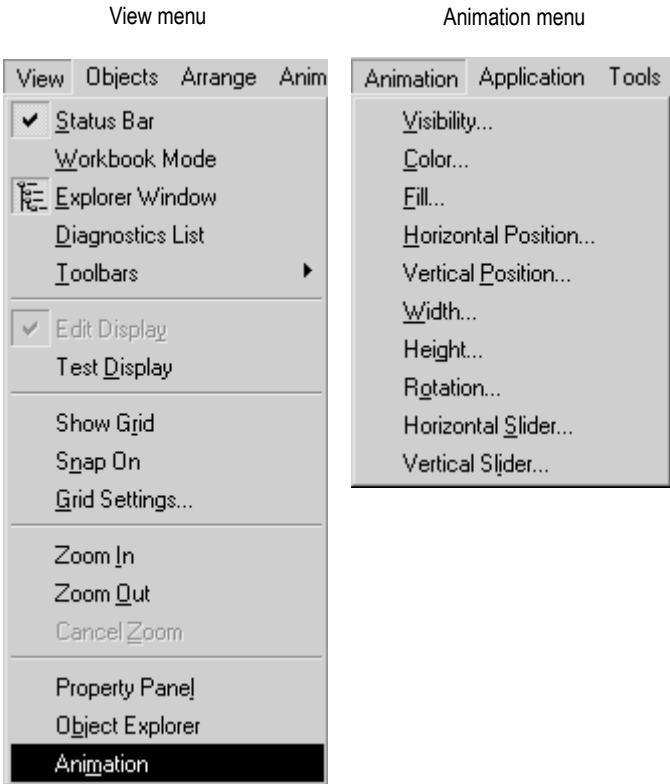
Using the Animation dialog box

To attach animation, use the Animation dialog box.

To open the Animation dialog box, do one of the following

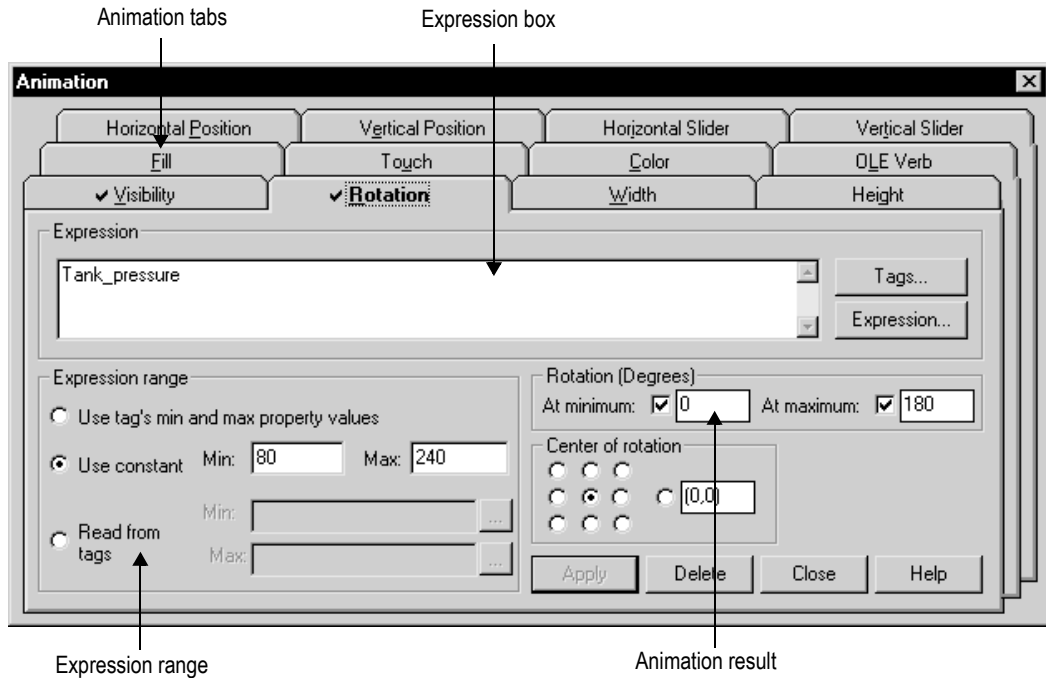
- Select an object, and then on the View menu click Animation.
- Select an object, and then on the Animation menu click an animation type. Animation types that are not supported for the selected object are unavailable.

- Right-click an object, select Animation, and then click an animation type. Animation types that are not supported for the selected object are unavailable.



About the Animation dialog box

The Animation dialog box is a floating dialog box, which means you can keep it open all the time and move it around the screen. While it's open you can select other objects and open other dialog boxes.



For details about the parts of the Animation dialog box, see Help.

Using Object Smart Path to visually set animation

Because the Animation dialog box stays open, you can go back and forth between the dialog box and the graphic display. This makes it easy to set the range of motion for an object because you do not have to know how many pixels you want an object to move. Instead, you can set the range of motion visually using the Object Smart Path feature. For details, see page 17-9.

Testing animation

To test the animation you have set up in a graphic display, use the Test Display tool to switch to test mode. When you are finished testing, switch back to edit mode to continue editing.



Test Display tool



Edit Display tool

To switch between test and edit modes

- ▶ On the View menu, click Test Display or Edit Display, or click the Test Display and Edit Display tools.

IMPORTANT

Test mode is not the same as running the display. Test mode does not change the appearance or position of the display as set up in the Display Settings dialog box.

Using tag names and tag placeholders

When setting up animation for objects, you are linking objects to tags, so you have to specify a tag name or tag placeholder. Following is a brief description of how to use tag names and placeholders.

Tag names

You can use HMI tags or data server tags that already exist, or you can use a new tag name.

Tag placeholders

Tag placeholders allow you to create displays that can be used with different tags.

You can use tag placeholders in:

- the graphic display that opens when the application is first run.
- graphic displays that are opened using a goto display button.
- graphic displays that are opened using a display list selector.

Use parameter files to specify which tags to substitute for which placeholders. For information about using parameter files, see page 14-28.

The tag placeholder can replace any part of a tag name, including folder names. For example, you could create a parameter file specifying that the tag placeholder #1=Folder1. You could assign the folder and a tag name to a graphic object's control: #1\Tag1.

To create a tag placeholder

- In the Expression box, type the cross-hatch character followed by a number (no space in between). For example, #1.

Using expressions

Many types of animation can be achieved using expressions. You can use expressions containing tag values, constants, mathematical equations, security functions, and if-then-else logic. A tag name or tag placeholder can be included as part of an expression, or it can stand alone as the entire expression.

For more information about expressions, see Chapter 19.

Setting minimum and maximum values

Many types of animation require a minimum and maximum range for the tag or expression. These values determine the start and end points for a range of motion.

For example, if you specify a minimum of 0 and a maximum of 100, the object will not react to values outside of this range. So, even if the expression has a value of 200, the object does not change from its “At maximum” position.

When setting up animation, select one of these methods for calculating the minimum and maximum values:

- Use tag's min and max property values—select this method to use the minimum and maximum values of the first HMI tag in the expression. If more than one HMI tag is used in the expression, the first HMI tag's minimum and maximum values are used.

For analog HMI tags, the values are taken from the Minimum and Maximum boxes in the Tags editor. For digital tags, the minimum is 0 and the maximum is 1.

- Use constant—select this method to use numeric constants. Type the minimum and maximum values in the boxes.
- Read from tags—select this method to read two tags' values to determine the minimum and maximum values. Type the tag names in the boxes, or click the Browse buttons to open the Tag Browser and select the tags.

If you use this method, the tags are read when the graphic display opens. Their values at that time are used for the minimum and maximum values. The tags are not read again after this.



Browse button

Defining a range of motion

To define a range of motion for an object, do one of the following:

- Use the mouse to move the object in the display. This uses the Object Smart Path feature to visually define the range of motion.
- Type values in the At minimum and At maximum boxes.

Motion can be defined in pixels, percentages, or degrees.

Animation that does not use a range of motion

Visibility and color animation do not use a range of motion, because these types of animation represent a change of state, not a range of values.

Using Object Smart Path

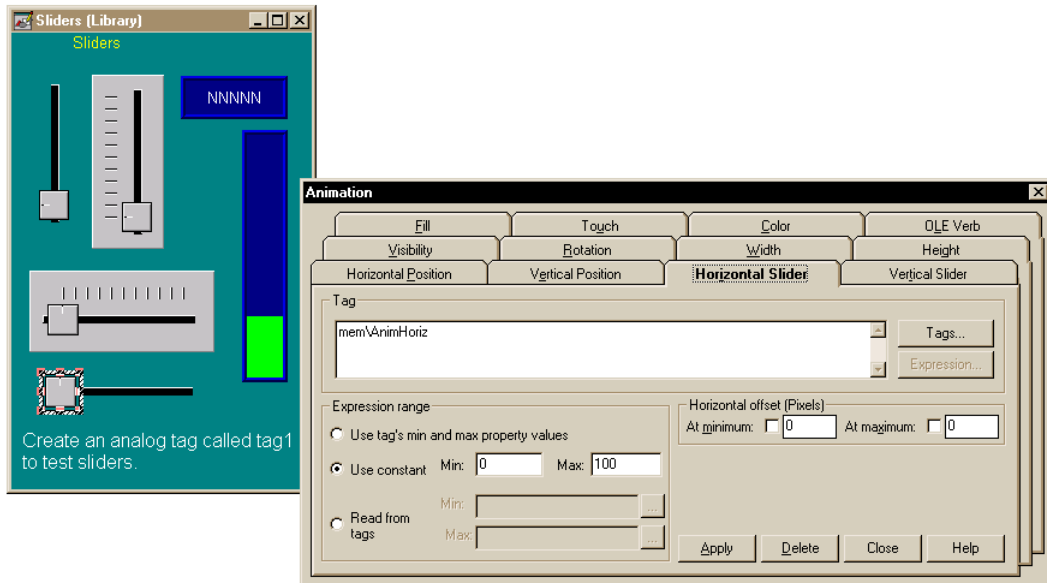
With Object Smart Path, you can easily set the range of motion for an object. The following example shows how Object Smart Path works.

Example: Using Object Smart Path to define the range of motion for horizontal slider animation

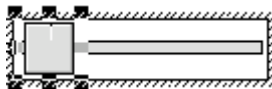
To define a range of motion for a slider object

- 1.** In the Graphic Displays editor, create a slider object using a line and a rectangle, or copy a slider object from the Sliders graphic library.
- 2.** Open the Animation dialog box and click the Horizontal Slider tab.
- 3.** In the display, select the rectangle in the slider object.

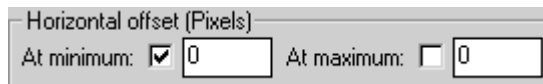
4. In the Tag box of the Animation dialog box, specify a tag name.



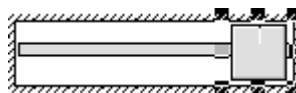
5. In the display, drag the rectangle to the position that will indicate the lowest number in the range.



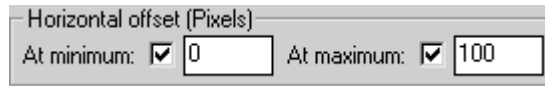
6. In the Animation dialog box, set this position by clicking the At minimum check box.



7. In the display, drag the rectangle to the position that will indicate the highest number in the range.



8. In the Animation dialog box, set this position by clicking the At maximum check box.



9. To save the settings, click Apply.

When you finish setting up the animation, the rectangle returns to its original position.

Setting up the different types of animation

This section describes the different types of animation and provides tips and examples for setting up animation.

Setting up visibility animation

With visibility animation, an object becomes visible or invisible based on a tag value or the result of an expression.

If an object is invisible, it is inactive.

Visibility animation is available for all objects. Visibility animation overrides an object's Visible property.

IMPORTANT

If you use a tag's value to control visibility animation as well as in an expression to control some other aspect of animation, when you set up visibility, set the Expression true state to "Invisible." If you do not do this, the object could appear briefly in its design-time location and orientation before animating properly.

For more information about setting up visibility animation, see Help.

Example: Using visibility animation to set up security for a graphic display

This example shows how to use visibility animation to control what operators can see. In a graphic display that all users have access to, only the Admin user can see the graphic object to which this animation is attached.

This example uses the security function `CurrentUserName()`. The function returns the string value of the Account ID (user name) for the user who is currently logged in.

The `CurrentUserName()` function is case sensitive. All user names use uppercase letters, so make sure that you use uppercase letters in your expression.

To specify which user can view an object in a display

1. Select the object to limit visibility to.
2. Open the Animation dialog box and click the Visibility tab.
3. In the Expression box, type this:

```
CurrentUserName() == "ADMIN"
```

4. For the Expression true state, click Visible.
5. Click Apply, and then click Close.

At run time, the object is visible only if the Admin user is logged in.

Setting up color animation

With color animation, an object changes color based on a tag value or the result of an expression. You can specify up to 16 color changes (A to P) for any object. Colors can be solid or blinking. For each color change, specify the value or threshold at which the color is to change

and specify the colors to change to. At run time, when the value reaches or crosses the threshold, the color changes.

Color animation is available for all drawing objects except images and panels.

For more information about setting up color animation, see Help.

Example 1: Creating a text object that blinks

This example describes how to create a text object that constantly blinks between two colors. Since the blinking is not based on changes in tag values, the expression is simply a constant value that matches the value for the selected threshold.

For details about creating text, see page 15-15.

- 1.** Select the text object.
- 2.** Open the Animation dialog box, and then click the Color tab.
- 3.** In the Expression box, type 0.

Zero is the default value for threshold A.

- 4.** In the list box, click threshold A. (Leave the value in the Value box as 0.)
 - 5.** For the foreground color, click Blink. (If desired, click Blink for the background color too.)
 - 6.** For each color, click the color box, and then click the color to use.
 - 7.** Click Apply.
-

Example 2: Creating an object that changes color as the fill level changes

This example describes how to create a rectangle object that changes color as the object's fill level increases. This example uses a tag called Hopper1\FlourLevel. The tag has a range of 1 to 100. When the flour level reaches 80, the rectangle blinks between gray and yellow to warn the operator that the hopper is nearly full. When the flour level reaches 95, the rectangle blinks between gray and red.

You could use a bar graph object (without animation) to achieve a similar result.

1. Double-click the rectangle to open the Polygon Properties dialog box. Assign these properties to the rectangle:
 - In the Back style box, select Solid.
 - For the Fore color and Back color, select gray.
2. Click OK to close the Polygon Properties dialog box.
3. With the rectangle selected, open the Animation dialog box, and then click the Fill tab.
4. Attach fill animation as follows:
 - In the Expression box, type Hopper1\FlourLevel (this is the tag that monitors the fill level).
 - For Fill Direction, click Up.

5. Click the Color tab, and then attach color animation as follows:

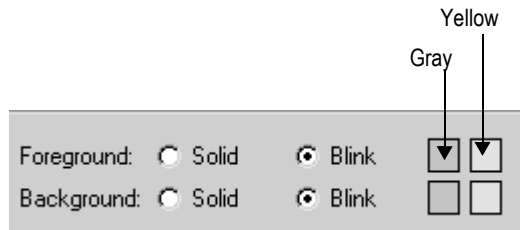
- In the Expression box, type Hopper1\FlourLevel (the same tag that was used in the Fill tab).

Set up the color for the normal state

- In the list box, click A. (In the Value box, leave the value as 0.)
- For foreground and background colors, click Solid.
- For each, click the color box, and then click gray (the same gray used for the rectangle).

Set up the color for the first warning

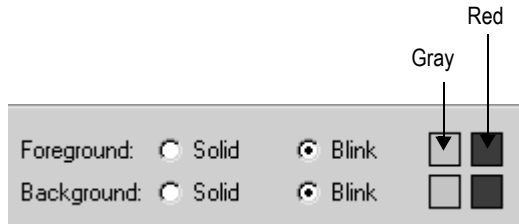
- In the list box, click B.
- In the Value box, type 80.
- For foreground and background colors, click Blink.



- For the foreground colors, select gray for the first color and yellow for the second color. Repeat for the background colors.

Set up the color for the second warning

- In the list box, click C.
- In the Value box, type 95.
- For the foreground and background colors, click Blink.



- For the foreground colors, select gray for the first color and red for the second color. Repeat for the background colors.

6. Click Apply.

Setting up fill animation

With fill animation, the level of fill in an object is based on a tag value (or the result of an expression) in relation to the specified minimum and maximum values. For example, if the value of the expression is halfway between the minimum and maximum values, the object will be half full.

Fill animation is available for all drawing objects (including group objects) except images and panels.

If you select the Inside Only check box, fill animation does not affect objects' borders, line objects, or objects with transparent backgrounds.

For more information about setting up fill animation, see Help.

Setting up horizontal position animation

With horizontal position animation, an object moves horizontally based on a tag value (or the result of an expression) in relation to the specified minimum and maximum values. For example, if the value of the expression is halfway between the minimum and maximum values, the object will be halfway between its minimum and maximum pixel offset.

Horizontal position animation is available for all drawing objects except images and panels.

For more information about setting up horizontal position animation, see [Help](#).

Setting up vertical position animation

With vertical position animation, an object moves vertically based on a tag value (or the result of an expression) in relation to the specified minimum and maximum values. For example, if the value of the expression is halfway between the minimum and maximum values, the object will be halfway between its minimum and maximum offset.

Vertical position animation is available for all drawing objects except images and panels.

For more information about setting up vertical position animation, see [Help](#).

Setting up width animation

With width animation, an object's width changes based on a tag value (or the result of an expression) in relation to the specified minimum and maximum values. For example, if the value of the expression is halfway between the minimum and maximum values, the object's width will be halfway between its minimum and maximum percentage.

Width animation is available for all drawing objects except images and panels.

For more information about setting up width animation, see Help.

Setting up height animation

With height animation, an object's height changes based on a tag value (or the result of an expression) in relation to the specified minimum and maximum values. For example, if the value of the expression is halfway between the minimum and maximum values, the object's height will be halfway between its minimum and maximum percentage.

Height animation is available for all drawing objects except images and panels.

For more information about setting up height animation, see Help.

Setting up rotation animation

With rotation animation, an object rotates around an anchor point based on a tag value (or the result of an expression) in relation to the specified minimum and maximum values. For example, if the value of the expression is halfway between the minimum and maximum values, the object will rotate half the specified amount.

Rotation animation is available for all drawing objects except images, panels, and rounded rectangles. If you apply rotation animation to text, the text rotates around the anchor point but remains in the upright position.

For more information about setting up rotation animation, see Help.

Setting up horizontal slider animation

With horizontal slider animation, you can use a drawing object to set the value of a tag. To do this, define a path for the object. At run time, when the operator moves the object horizontally (using a mouse), the

pixel position of the object is translated into values that are written to the tag. If the tag value is changed externally, the position of the slider changes as well.

An object can have both vertical and horizontal slider animation.

Horizontal slider animation is available for all drawing objects except images and panels.

For more information about setting up horizontal slider animation, see [Help](#).

Tips

Here are some tips for creating slider objects:

- The Sliders graphic library contains several slider objects you can drag and drop into your graphic displays. Attach slider animation to the button portion of the slider object.
- If you create your own slider object, it's useful to draw an object (for example, a line) to represent the path the slider will move along.

Setting up vertical slider animation

With vertical slider animation, you can use a drawing object to set the value of a tag. To do this, define a path for the object. At run time, when the operator moves the object vertically (using a mouse), the pixel position of the object is translated into values that are written to the tag. If the tag value is changed externally, the position of the slider changes as well.

An object can have both vertical and horizontal slider animation.

Vertical slider animation is available for all drawing objects except images and panels.

For tips about creating slider objects, see the previous section.

For more information about setting up vertical slider animation, see Help.

Applying animation to groups

You can apply animation to objects and then group those objects and apply animation to the group. When the display is running, animation is applied as follows:

These types of animation	Are applied like this
Color	Animation attached to individual objects within the group overrides group animation.
Fill	Animation results for both the individual objects and the group are applied.
Horizontal slider, vertical slider	Group animation overrides animation attached to individual objects within the group.
Height, width, horizontal position, vertical position, and rotation	Animation results for individual objects and the group are combined. For example, if an individual's horizontal position animation result is to offset the object by 100 pixels, and the group's result is to offset the group by 200 pixels, the individual object is offset by 300 pixels.
Visibility	When the group's animation visibility is False (the group is not visible), then no objects in the group are visible, regardless of the animation status of the individual objects. When the group's animation visibility is True (the group is visible), the visibility of an object within the group is determined by the individual object animation.

Test your animation to ensure you achieve the intended results.

To apply animation to objects within groups, use the group edit feature. For details, see page 15-58.

Checking the animation on objects

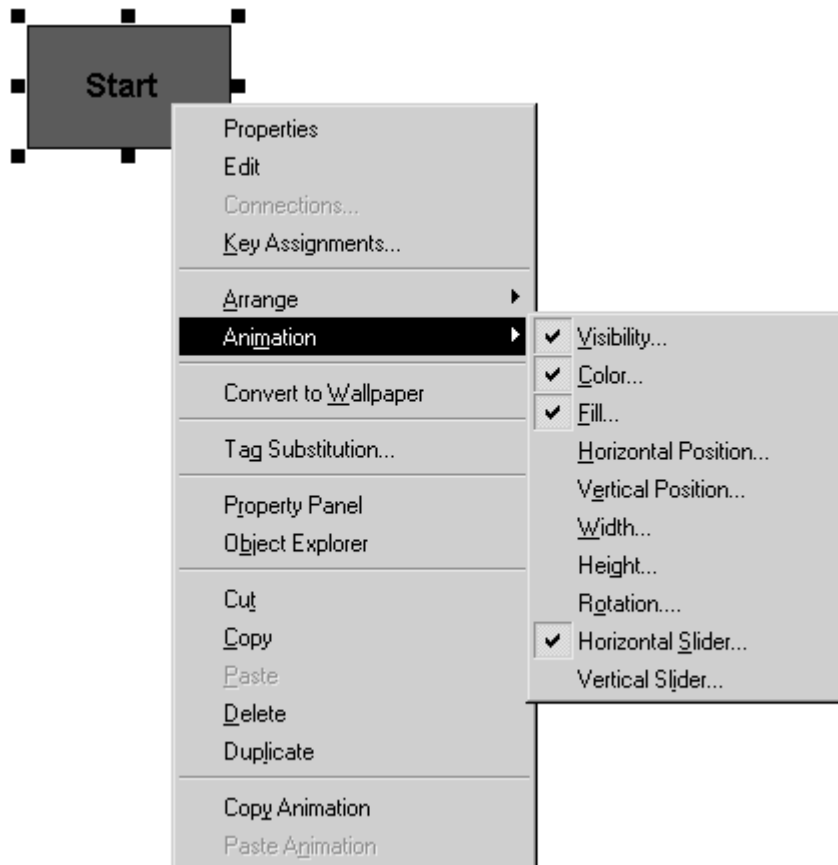
To see what type of animation has been set up for an object or group of objects, use the Object Explorer, the Animation menu, or the Animation dialog box.

For information about using the Object Explorer to highlight objects that have animation attached, see page 15-30.

To see what type of animation has been set up for objects within a group, use the group edit feature or the Object Explorer. For information about using the group edit feature, see page 15-58.

To view the animation on an object using the Animation menu

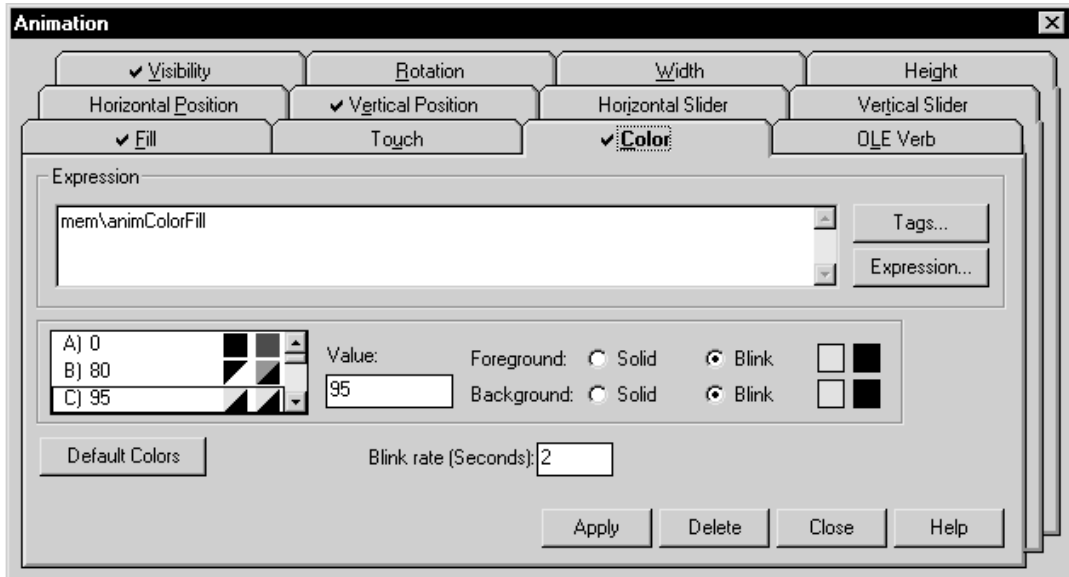
1. Select an object.
2. View the animation by doing one of the following:
 - Click the Animation menu and see which items have a check mark.
 - Right-click the object and then select Animation to see which items have a check mark.



To view the animation on an object using the Animation dialog box

1. Select an object.
2. On the View menu, click Animation.

When the Animation dialog box opens, look at which tabs have check marks on them to see which types of animation have been set up.



Copying or duplicating objects with animation

You can copy or duplicate objects that have animation attached to them. When you do, the animation attached to the objects is also copied or duplicated. If you copy or duplicate a group, the copy of the group can be ungrouped to individual objects, just like the original.

For information about copying and duplicating objects, see pages 15-49 and 15-51.

Copying animation without copying objects

If you have attached animation to an object, you can copy the animation and paste it onto another object. If the object has more than one type of animation, all animation is copied and pasted. Note that you can only copy animation to an object that supports the same type of animation.

To copy and paste animation

1. Select the object that has the animation you want to copy.
2. On the Edit menu, click Copy Animation, or right-click the object and then click Copy Animation.
3. Select the objects to copy the animation to.
4. On the Edit menu, click Paste Animation.

To paste to a single object, you can right-click the object and then click Paste Animation.

Setting up trends

This chapter describes:

- what trends are.
- summary of steps for creating a trend graphic object.
- creating trends, and the Trend Object Properties dialog box.
- the parts of the trend graphic object.
- the different chart types.
- choosing colors, fonts, lines, and markers for the trend.
- testing the trend.
- using objects from the Trends graphic library.
- using buttons to control the trend at run time.
- printing trend data.
- runtime errors for trends.

About trends

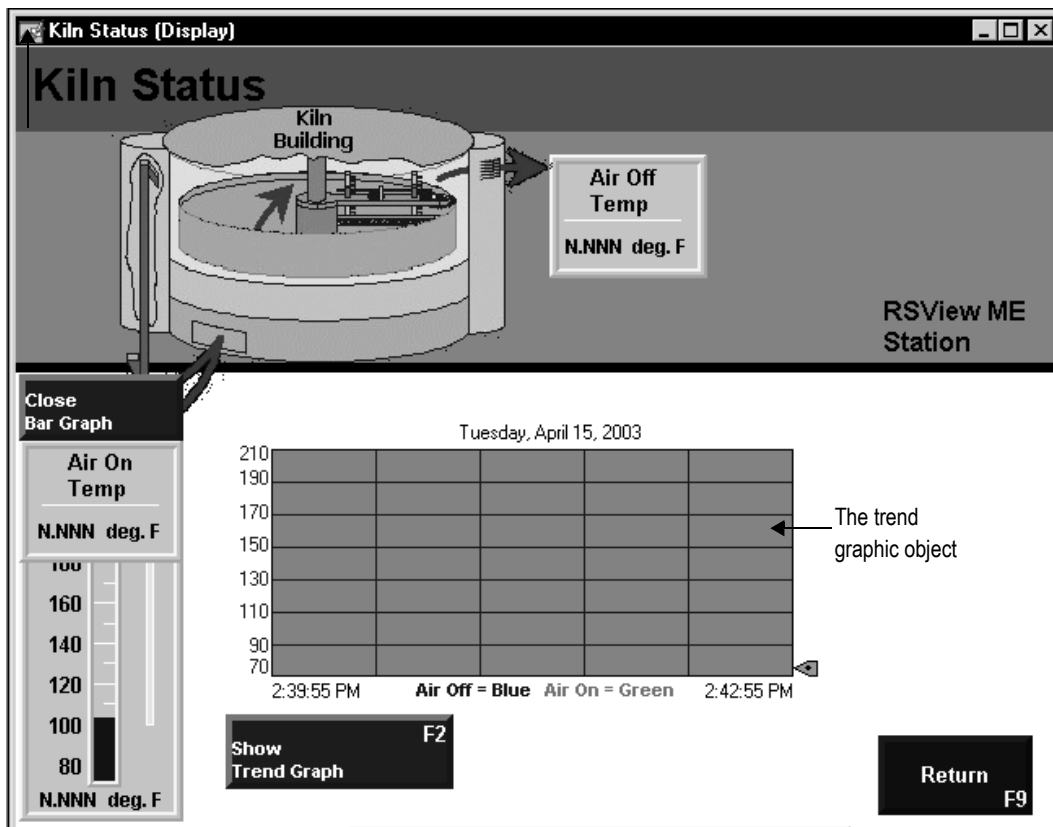
A trend is a visual representation of current or historical tag values. The trend provides operators with a way of tracking plant activity as it is happening.

You can:

- plot data for as many as eight tags or expressions on one trend.
- create a trend that is part of a graphic display or acts as the entire graphic display.

- plot data over time, or plot one variable against another in an XY Plot chart to show the relationship between them.
- display isolated or non-isolated graphs. Isolated graphing places each pen in a separate band of the chart. With non-isolated graphing, pen values can overlap.
- create buttons to allow the operator to pause, scroll, and print the trend data.

The illustration below shows a trend that has been added to a graphic display. You can view the Kiln Status display by opening the Malthouse sample application.



Current versus historical data

The data displayed in a trend can come from two sources. For current values, data comes from the value table as it is collected. The value table is a record of the most recent values collected from the data source, and is stored in temporary memory while the application is running.

For historical values, data comes from a data log model's log file, if a model is assigned to the trend. You can display both current and historical data in the same trend.

For information about data log models, see Chapter 11.

Time, date, and number formats

The trend is displayed using the time, date, and number formats you set up on the runtime computer.

For information about specifying time, date, and number formats:

- in Windows 2000 and Windows XP, see page 24-18.
- on a PanelView Plus™ terminal, see the *PanelView Plus Operator Terminals User Manual*.
- on a VersaView CE terminal, see the *VersaView CE Terminals User Manual*.

Summary of steps

These are the steps for creating a trend:

1. To plot historical data, create a data log model in the Data Log Models editor. For information, see Chapter 11.
2. Create a trend graphic object in the Graphic Displays editor, as described on page 18-4.
3. Set up the trend in the Trend Object Properties dialog box. For details about the options in the dialog box, see Help.

4. If desired, create a next pen button, a pause button, or key buttons in the same graphic display, to allow the operator to switch between pens, pause the trend, or scroll the trend.

For information about the buttons you can use with trends, see page 18-14.

5. To keep a printed record of the trend data, provide a way for the operator to print the graphic display. For information see page 18-16.

Creating trend objects

To create a trend object

1. In the Graphic Displays editor, create or open a graphic display.
2. Select the Trend drawing tool by doing one of the following:
 - In the Objects toolbox, click the Trend tool.
 - On the Objects menu, select Trending, and then click Trend.
3. Drag the mouse to create a box approximately the size you want for the trend.
4. Double-click the trend to open the Trend Object Properties dialog box.
5. Set up the trend. For details, see Help.

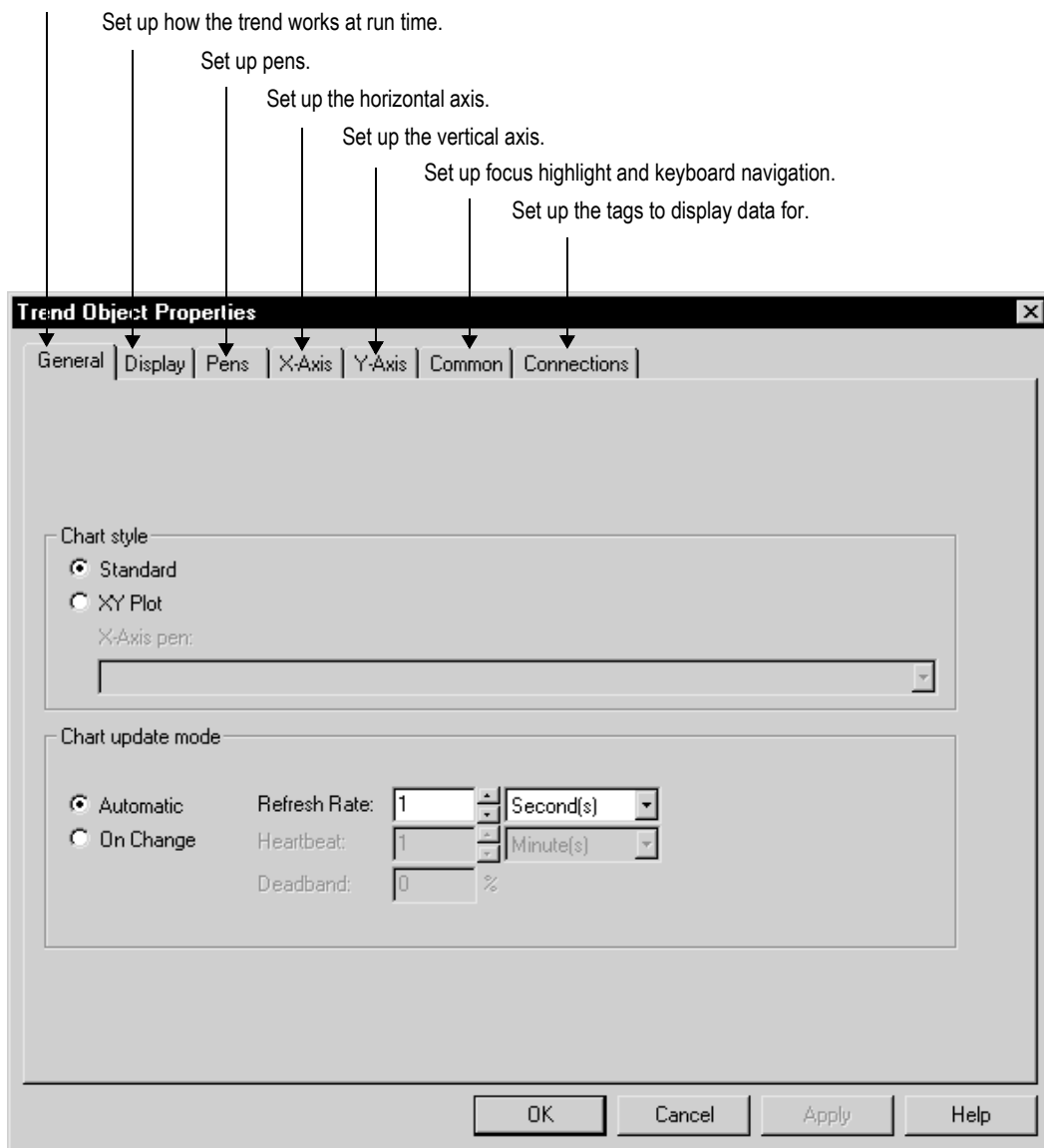
Once you have set up the trend, you can edit it as you would any other graphic object. You can move it, resize it, attach animation to it, and so on. You can also use this object in other graphic displays by dragging it from one display and dropping it into another.

For more information about graphic objects, see Chapter 15.

Setting up trends

When you double-click a trend object, the Trend Object Properties dialog box opens. Use the dialog box to set up the trend.

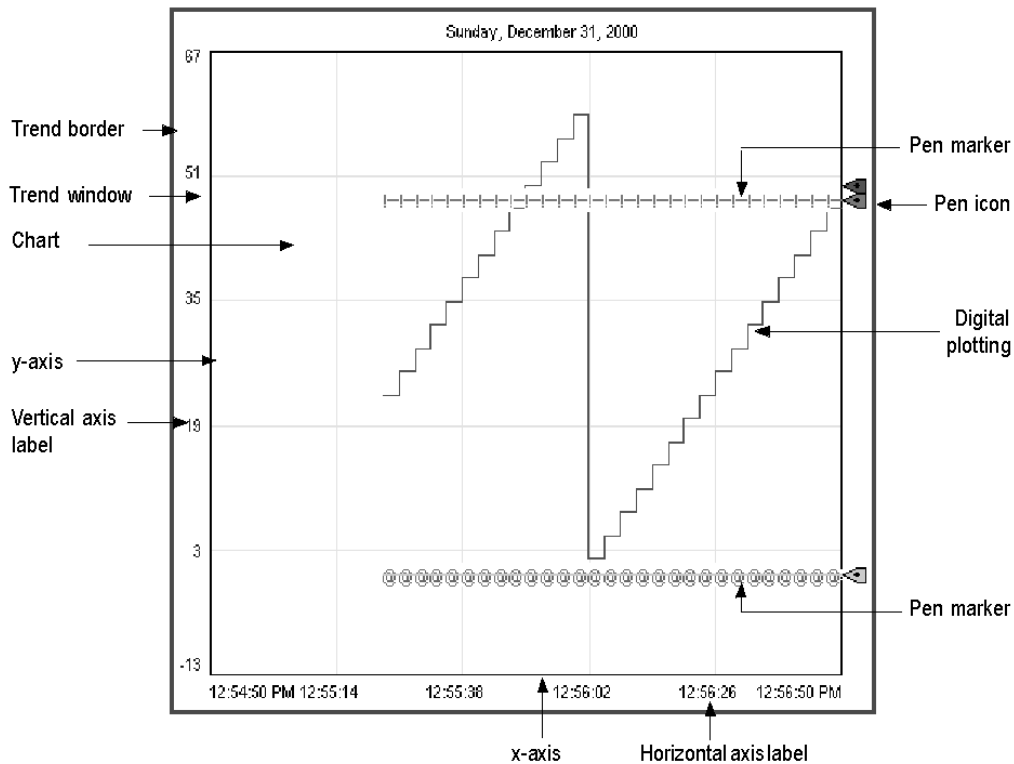
Set up the chart style and update mode.



For details about the options in the Trend Object Properties dialog box, see Help.

The parts of a trend

The illustration below shows a standard trend chart, with three pens and a two-minute time span. Two of the pens have markers. The third uses digital plotting. For more information about chart types, see page 18-9.



Trend border

The border appears around the trend object at run time when the trend is selected.

Trend window

The area around the chart, between the border and the chart, is the trend window.

Chart

The chart is the area of the trend in which values are plotted. It is bounded by the y-axis on the left and the x-axis on the bottom. It contains the plotted trend data (shown using pen lines and pen markers), as well as grid lines (if you choose to display them).

Y-axis

The y-axis is the left vertical edge of the chart. It is also known as the vertical axis.

Vertical axis labels

The vertical axis labels show the scale (range) of values for the pens. If desired, you can set up the trend to omit the vertical axis labels.

The minimum and maximum values for the scale can be determined automatically (using the best fit for the current data), be derived from a pen's minimum and maximum values, use a constant value, or be controlled by tags.

You can set up the trend so all pens use the same scale, or use individual ranges for each pen. If you choose the latter method, create a next pen button in the graphic display, to allow operators to view the range for each pen. When the operator presses the button, the vertical axis changes to the new pen's range.

For example, if Pen 1 has a minimum value of 10 and a maximum value of 100, the range on the vertical axis is 10 to 100 when the pen is selected. If Pen 2 has a minimum of -10 and a maximum of 50, the

range on the vertical axis changes to -10 to 50 when the operator presses the next pen button.

X-axis

The x-axis is the bottom horizontal edge of the chart. It is also known as the horizontal axis.

Horizontal axis labels

For standard charts, the horizontal axis labels indicate the time span covered by the trend. For XY Plot charts, the horizontal axis labels show the scale (range) of values for the pen selected to serve as the x-axis pen.

If desired, you can set up the trend to omit the horizontal axis labels. The number of labels depends on the size of the trend object and the number of vertical grid lines.

Pens

Pens are the lines and symbols used to represent values. The values can be tags you are monitoring, expressions that manipulate tag values, or constants.

If there is no data for a pen, or if the data is outside the vertical axis range, the pen does not appear in the chart.

Pen icons

Pen icons appear at the right edge of the chart at run time, if you choose to display them. The icon's position indicates the pen's most recent recorded value (from the value table), even if the trend is paused or if the most recent value has not been plotted yet.

Pen markers

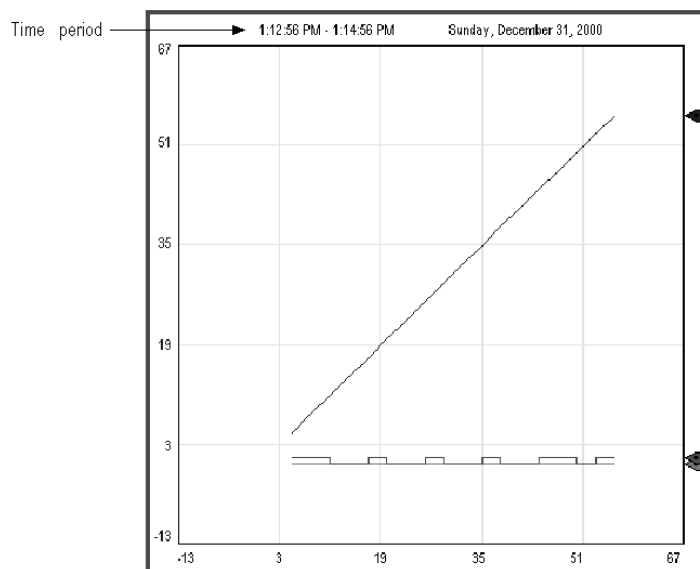
Pen markers are symbols that indicate data points. If data is plotted frequently, the markers might not appear as distinct, separate symbols. For example, see the lowest pen in the illustration on page 18-6.

Chart types

Standard vs. XY Plots

You can create a standard chart, which plots tag values against time, or an XY Plot chart, which plots one (or more) tag's values against another's.

This illustration shows what an XY Plot chart could look like:

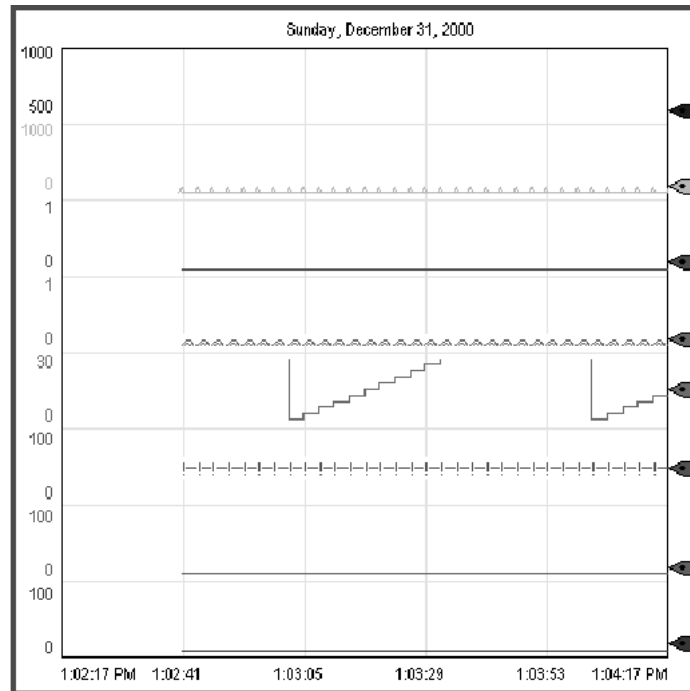


Notice that the horizontal axis labels display the range for the specified x-axis pen. The time period covered by the chart is at the upper left.

Isolated graphing

For charts with multiple pens, you can allow the pen values to overlap, or you can isolate each pen in its own horizontal band on the chart.

This is an example of isolated graphing, with a 10% buffer between each pen's band:



Notice that in this illustration each pen uses its own scale. If desired, you can use the same scale for all pens.

With isolated graphing, a grid line is automatically placed above each pen's band.

Plotting a value across the full width of the chart

Use horizontal lines to provide a frame of reference for your tag data. For example, if you define values that are the limits within which a tag must operate, and display horizontal lines in your trend to indicate the limits, when a tag crosses one of these limits the tag's alarm condition is obvious on the trend.

There are two ways to plot a value across the full width of the chart:

- In the Connections tab, assign a constant value to a pen.

When values for the pen have been plotted across the full width of the chart, the pen appears as a solid line.

- In the Connections tab, assign to a pen the tag, expression, or constant whose value will be used to determine the position of the line, and then in the Pens tab, choose the pen type Full Width.

As soon as the trend is displayed, the pen appears as a horizontal line across the full width of the chart. Its vertical position is determined by the tag, expression, or constant's value. If the value changes, the position changes.

Choosing trend colors, fonts, lines, and markers

The following table summarizes where in the Trend Object Properties dialog box to specify colors, fonts, lines, and markers for a trend.

You can also specify these settings in the Properties tab of the Property Panel.

To specify this	Use this box or column	In this tab
Chart background color	Background color	Display
Horizontal label color	Text color	Display
Text font, style, and size	Font (button)	Display

To specify this	Use this box or column	In this tab
Pen line, pen marker, pen icon, and vertical label color	Color	Pens
Pen line width	Width	Pens
Pen line style	Style	Pens
Pen marker	Marker	Pens
Vertical grid line color	Grid color	X-Axis
Horizontal grid line color	Grid color	Y-Axis

The trend border color

The trend border uses the highlight color for the graphic display, specified in the Behavior tab of the Display Settings dialog box.

The trend window color

By default, the trend window uses the background color of the display, specified in the General tab of the Display Settings dialog box.

To use a different window color

- In the Property Panel, select the opaque WindowStyle, and then specify the WindowColor property.

For information about using the Property Panel, see page 15-34.

Testing the trend

You can quickly test the trend by switching to test mode. If communications are active and there is data for the tags, the pens plot values in the trend. When you are finished testing, switch back to edit mode to continue editing.



Test Display tool



Edit Display tool

To switch between test and edit modes

- On the View menu, click Test Display or Edit Display, or click the Test Display and Edit Display tools.

IMPORTANT

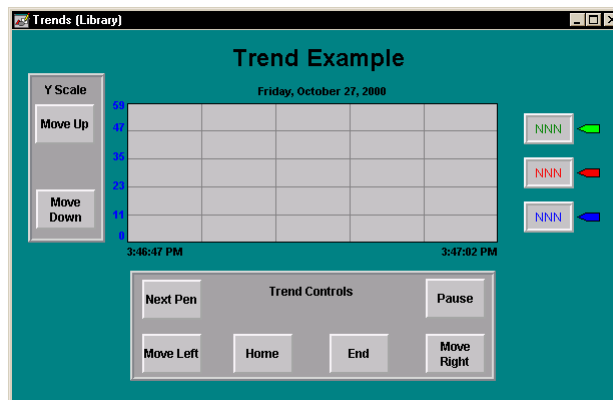
Test mode is not the same as running the display. Test mode does not change the appearance or position of the display as set up in the Display Settings dialog box. Also, data logging is not turned on in test mode.

Using the Trends graphic library

The Trends graphic library contains a trend graphic object and buttons for controlling the trend. It also contains numeric display objects that display the value of each tag used in the trend.

You can use the trend and objects as they are, or you can edit them to suit your needs. To use the objects, drag and drop (or copy and paste) them into your graphic display.

For information about copying and pasting objects from the graphic libraries, see page 15-50.



To use the Trends graphic library

- 1. Open the Graphics folder, and then open the Libraries folder.
- 2. Double-click the Trends library.
- 3. Drag and drop or copy and paste objects into your display.

Using buttons to control the trend at run time

You can use button graphic objects with the trend, to allow the operator to pause the trend, switch between pens, or scroll the trend.

You can link buttons to a specific trend object, or set up a button to work with whichever object is selected in the graphic display. For information about linking buttons to objects, see page 16-11.

Use these buttons with trends:

This button	Does this
Pause	<p>Toggles between pausing and automatic scrolling.</p> <p>When the trend is paused, the pen icons continue to move vertically to indicate the pens' current values.</p> <p>When the trend resumes scrolling, values that occurred while the trend was paused are filled in, bringing the trend up to the current time (unless you are scrolling historical data).</p>
Next pen	<p>Changes the vertical axis labels to the scale for the next pen. The color of the labels matches the color of the selected pen.</p>

This button	Does this
Move up	<p>Scrolls up to display higher values on the vertical scale. For example, if the visible scale range is 0 to 100, pressing move up could change the visible range to 10 to 110.</p> <p>The incremental amount the axis scrolls depends on the pen's range and the number of horizontal grid lines.</p> <p>This button does not work if the “Minimum / maximum value option” in the Y-Axis tab is set to Automatic.</p>
Move down	<p>Scrolls down to display lower values on the vertical scale.</p> <p>This button does not work if the “Minimum /maximum value option” in the Y-Axis tab is set to Automatic.</p>
Move left	Pauses the trend and scrolls to the left.
Move right	Pauses the trend and scrolls to the right.
Home	Pauses the trend and moves to the earliest data in the trend.
End	Resumes trend scrolling and moves to the current (latest) data in the trend.

To see how the buttons work with the trend, open the Trends graphic library (see page 18-13), and start test mode.

For information about creating buttons, see Chapter 15. For details about setting up the buttons, see page 16-21.

Printing trend data

To print trend data at run time, provide the operator with a method for printing the graphic display.

You can use these methods to print graphic displays at run time:

- Create a display print button. For information about creating graphic objects, see Chapter 15.
- Assign a tag or expression to the Remote Display Print control (in the Global Connections editor). When the value of the tag or expression changes from 0 to a non-zero value, the current display is automatically printed.

Program the data source to trigger the change as often as you want the data printed.

For more information about setting up remote display printing, see Chapter 8.

Everything on the screen is printed, including the current display, pop-up windows, and any visible background applications.

For information about specifying which printer to use at run time for Windows 2000 or Windows XP applications, see page 24-11.

For information about specifying printer options for applications that will run on a PanelView Plus terminal, see the *PanelView Plus Operator Terminals User Manual*.

For information about specifying printer options for applications that will run on a VersaView® CE terminal, see the *VersaView CE Terminals User Manual*.

Improving clarity of the trend printout

Depending on what type of printer you use, pen lines with a width of 1 pixel might not appear in the printout. Choose high-contrast colors and wider line widths to ensure that the trend data prints clearly.

Runtime errors for the trend

If data for the trend is not available at run time due to communication errors, a message is sent to FactoryTalk™ Diagnostics.

See Help for information about solving common trend problems.

Using expressions

This chapter describes:

- the types of expression components.
- using the Expression editor.
- which editors use expressions.
- formatting expressions.
- using tag names and tag placeholders with expressions.
- using constants.
- using operators, math and security functions, and if-then-else logic in expressions.
- the evaluation order of operators.
- using write expressions.

About expressions

Sometimes the data you gather from devices is meaningful only when you:

- compare it to other values.
- combine it with other values.
- create a cause-effect relationship with other values.

Expressions allow you to create mathematical or logical combinations of data that return more meaningful values. Depending on the

components used in the expression, the value returned can be in the form of a numeric value, a true/false value, or a text string.

Expressions that result in floating-point values

If an expression results in a floating-point value but an integer value is required, the floating-point value is rounded.

For information about how values are rounded, see page 7-2.

Expression components

Expressions can be built from:

- tag values.
- tag placeholders.
- constants.
- arithmetic, relational, logical, and bitwise operators.
- mathematical and security functions.
- if-then-else logic.

Tags, arithmetic operators, bitwise operators, and mathematical functions such as Sqrt (square root) return numeric values.

Relational and logical operators return true/false values. The security function `CurrentUserHasCode(x)` also returns a true/false value.

The security function `CurrentUserName()` returns a string value.

Expressions that use if-then-else logic can return numeric values, true/false values, or text strings, depending on how they are structured. These are called conditional expressions because the result of the expression depends on whether the If statement is true or false. When the If statement evaluates to true, the result is defined by the Then

statement. When the If statement is false, the result is defined by the Else statement.

The Objects 5 Screen Demo sample application contains many examples of expressions. For example, see the alarm trigger expressions in the Alarm Setup editor.

Using the Expression editor

To create an expression, you can:

- type it directly in the “Tag or expression” column, for any control that accepts expressions, or in the Expression box (for animation).
- open the Expression editor, and then create the expression in the editor.

Using the Expression editor versus typing expressions directly

Once you are familiar with expression syntax, you might find it quicker to create short expressions by typing them directly in the “Tag or expression” column.

The Expression editor allows you to see more text at once, which is useful for longer, more complicated expressions. Also, you can click buttons to enter tag names, operators, and functions, thus avoiding typing mistakes. Another advantage of using the Expression editor is that you can check whether the syntax of the expression you’ve created is valid.

To create an expression by typing it directly

- Type an expression up to 999 characters long.

Expressions that you type directly are not checked for syntax.



Browse button in the Exprn column

To open the Expression editor, do one of the following

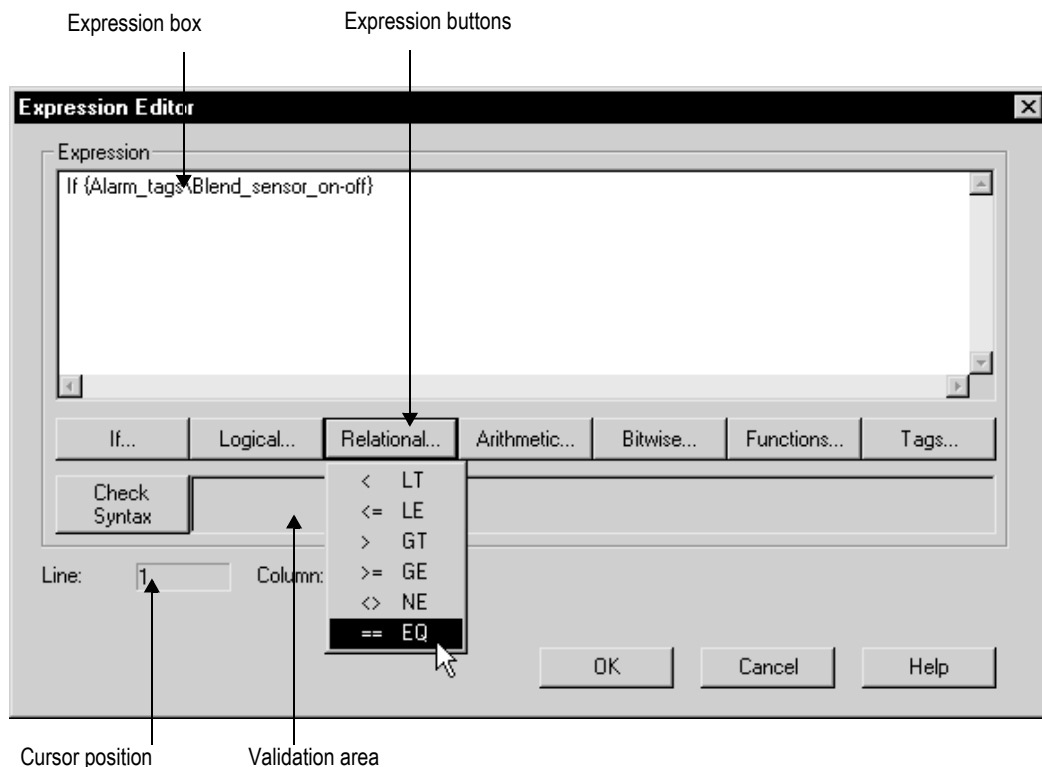
- Click the Browse button in the Exprn column for a control that accepts expressions.

The Browse button is not available for controls to which you can assign only tags.

- In the Animation dialog box, click the Expression button.

About the Expression editor

The Expression editor has these parts:



For details about using the options in the Expression editor, see Help.

Where you can use expressions

You can use expressions in these editors:

- **Graphic Displays**—You can define an expression to control various aspects of a graphic object's appearance. For more information about assigning expressions to graphic objects, see page 15-44.

You can also use expressions to attach animation to graphic objects. For more information, see Chapter 17.

- **Global Connections**—You can use expressions to remotely control when to open and print displays, as well as the date and time to display. For more information, see Chapter 8.
- **Alarm Setup**—When setting up alarms, you can use expressions for alarm triggers, and with some of the controls that silence and acknowledge alarms. For more information, see Chapter 9.
- **Information Setup**—You can use expressions to determine when to display information messages. For more information, see Chapter 12.
- **Macros**—You can use expressions in macros to assign values to tags. For more information, see page 21-1.

Formatting expressions

You can format expressions so they are easier to read. However, do not let tag names, function names, or function arguments span more than one line.

When formatting expressions, you can use line returns and multiple spaces.

Enclose strings in quotes. The string can contain any character, and can include spaces.

Example: Formatting an expression

To format this if-then-else statement, you can align the Else with the appropriate If, so the logic is easy to understand:

```
if (tag1 > tag2) then 0
else if (tag1 > tag3) then 2
else 4
```

Or you can condense it to the following:

```
if (tag1 > tag2) then 0 else if (tag1 > tag3) then 2 else 4
```

Using tag names and tag placeholders

A tag name can be included as part of an expression or can stand alone as the entire expression.

To supply a tag name, do one of the following:

- Type a tag name.

You can type a tag name that does not exist in the tag database. When you click OK, you are prompted to create the tag. You can create it now, or write down the name and create it later.

- Click the Tags button and select a tag from the Tag Browser.

Enclose tag names that contain dashes or start with a number in curly brackets { } when you use them in an expression. This distinguishes the characters in the tag name from the characters in the expression.

You can use string tags as operands with the plus (+) arithmetic operator and with the relational operators.

Using tag placeholders instead of tag names

The Graphic Displays editor accepts tag placeholders instead of tag names. Placeholders allow you to use the same display with different sets of tags.

You can use tag placeholders in:

- the graphic display that opens when the application is first run.
- graphic displays that are opened using a goto display button.
- graphic displays that are opened using a display list selector.

Use parameter files to specify which tags to substitute for which placeholders at run time. For information about using parameter files, see page 14-28.

The tag placeholder can replace any part of a tag name, including folder names. For example, you could create a parameter file specifying that the tag placeholder #1=Folder1. You could assign the folder and a tag name to a graphic object's control: #1\Tag1.

To create a tag placeholder in an expression

- Type the cross-hatch character followed by a number (no space in between). For example, #1.

Constants

A constant can have any of the following formats:

- integer (123)
- floating-point (123.45)
- string constant (“character string”)

Arithmetic operators

Arithmetic operators perform math on two or more numeric values and calculate the result. The arithmetic operators are:

Symbol	Operator	Example (For these examples, tag1 = 5 and tag2 = 7)
+	addition	tag1 + tag2 returns a value of 12 You can also use this operator with string operands. See page 19-9.
-	subtraction	tag1 - tag2 returns a value of -2
*	multiplication	tag1 * tag2 returns a value of 35
/	division	tag1 / tag2 returns a value of 0.7142857
MOD,%	modulus (remainder)	tag2 MOD tag1 returns a value of 2 The modulus operator is the remainder of one number divided by another. In the example, the remainder of 7 divided by 5 is 2; so 7 % 5 = 2 Important: This operator is for integers only, not floating-point numbers.
**	exponent	tag1 ** tag2 returns a value of 78125

IMPORTANT

Be sure that any tag value you use as a divisor cannot at some point have a value of zero. Expressions that attempt to divide a number by zero produce an error at run time.

String operands

The + operator can be used to join string operands. For example, the expression “hello” + “world” returns: helloworld.

You cannot join string tags to analog tags, whether they are HMI or data server tags.

Relational operators

Relational operators compare two numeric or string values to provide a true or false result. If the statement is true, a value of 1 is returned. If false, 0 is returned.

The relational operators are:

Symbols	Operator	Numeric Example	String Example
For the numeric examples, tag1 = 5 and tag2 = 7 For the string examples, serial_no = “ST009”			
EQ, ==	equal	tag1 == tag2 false	serial_no == “ST009” true
NE, <>	not equal	tag1 <> tag2 true	serial_no <> “ST011” true
LT, <	less than	tag1 < tag2 true	serial_no < “ST011” true
GT, >	greater than	tag1 > tag2 false	serial_no > “ST011” false

Symbols	Operator	Numeric Example	String Example
LE, <=	less than or equal to	tag1 <= tag2 true	serial_no <= "ST011" true
GE >=	greater than or equal to	tag1 >= tag2 false	serial_no >= "ST011" false

How string operands are evaluated

String operands are evaluated by case and by alphabetical order. Lower case letters are greater than upper case letters. For example, h is greater than H. Letters later in the alphabet are greater than those earlier in the alphabet. For example, B is greater than A.

Logical operators

Logical operators determine the validity of one or more statements. There are three logical operators: AND, OR, and NOT. The operators return a non-zero value if the expression is true, or a 0 if the expression is false.

Any statement that evaluates to a non-zero value is regarded as true. For example, the statement tag1 is false if the value of tag1 is 0, and true if tag1 has any other value.

The logical operators are:

Symbols	Operator	Action	Example (For these examples, tag1 = 5 and tag2 = 7)
AND, &&	and	Returns a 1 if the statements to the right and left of the operator are both true.	(tag1 < tag2) AND (tag1 == 5) Both statements are true; returns a 1.

Symbols	Operator	Action	Example (For these examples, tag1 = 5 and tag2 = 7)
OR,	or	Returns a 1 if either the statement to the left or right of the operator is true.	(tag1 > tag2) OR (tag1 == 5) tag1 == 5 is true; returns a 1.
NOT	negation	Reverses the logical value of the statement it operates on.	NOT (tag1 < tag2) Although tag1 < tag2 is true, NOT reverses the logical value; returns a 0.

IMPORTANT

The parentheses are essential in the above expressions. They determine the evaluation order of the operators. For more information, see page 19-15.

Bitwise operators

Bitwise operators examine and manipulate individual bits within a value.

IMPORTANT

These operators are for integers only, not floating-point numbers. Do not use them with tags or expressions that return floating-point values.

Symbol	Operator	Action (for examples, see page 19-14)
&	And	<p>Compares two integers or tags on a bit-by-bit basis.</p> <p>Returns an integer with a bit set to 1 if both the corresponding bits in the original numbers are 1. Otherwise, the resulting bit is 0.</p>
	inclusive OR	<p>Compares two integers or tags on a bit-by-bit basis.</p> <p>Returns an integer with a bit set to 1 if either or both of the corresponding bits in the original numbers are 1. If both bits are 0, the resulting bit is 0.</p>
^	exclusive OR (XOR)	<p>Compares two integers or tags on a bit-by-bit basis.</p> <p>Returns an integer with a bit set to 1 if the corresponding bits in the original numbers differ. If both bits are 1 or both are 0, the resulting bit is 0.</p>

Symbol	Operator	Action (for examples, see page 19-14)
>>	right shift	<p>Shifts the bits within an integer or tag to the right.</p> <p>Shifts the bits within the left operand by the amount specified in the right operand. The bit on the right disappears.</p> <p>Either a 0 or a 1 is shifted in on the left, depending on whether the left-most bit is a 0 or a 1, and whether the operand consists of a signed or unsigned data type.</p> <p>For signed data types, if the left-most bit is 0, a 0 is shifted in. If the left-most bit is 1, a 1 is shifted in. In other words, the sign of the number is preserved.</p> <p>For unsigned data types, a 0 is always shifted in.</p>
<<	left shift	<p>Shifts the bits within an integer or tag to the left.</p> <p>Shifts the bits within the left operand by the amount specified in the right operand. The bit on the left disappears and 0 always shifts in on the right.</p> <p>See “Using the left shift operator,” later in this chapter.</p>
~	complement	<p>Returns one’s complement; that is, it toggles the bits within an integer or tag.</p> <p>Reverses every bit within the number so every 1 bit becomes a 0 and vice versa.</p>

Using the left shift operator

If the left bit is a 1 an overflow occurs, and an error message is generated. To prevent this, use the bitwise AND operator with the left shift operator in an expression. For example:

```
(dev << 1) & 65535
```

where dev is a tag whose value is being shifted left, and 65535 is 1111 1111 1111 1111 in binary form.

Examples: Bitwise operators

For these examples, tag1 = 5 (binary 0000 0000 0000 0101) and tag2 = 2 (binary 0000 0000 0000 0010)

tag1 & tag2

Returns 0 (binary 0000 0000 0000 0000).

tag1 | tag2

Returns 7 (binary 0000 0000 0000 0111).

tag1 ^ tag2

Returns 7 (binary 0000 0000 0000 0111).

tag1 >> 1

Returns 2 (binary 0000 0000 0000 0010).

tag1 << 1

Returns 10 (binary 0000 0000 0000 1010).

~ tag1

Returns -6 (binary 1111 1111 1111 1010).

Evaluation order of operators

Expressions with more than one operator are evaluated in this order:

- Operators in parentheses are evaluated first.
Therefore, to change the order of precedence, use parentheses.
- The operator with the highest precedence is evaluated next.
- When two operators have equal precedence, they are evaluated from left to right.

Operators are evaluated in this order:

Evaluation order	Symbols
1 (highest)	()
2	NOT ~
3	* / MOD, % ** AND, && & >> <<
4	+ - OR, ^
5 (lowest)	EQ, == NE, <> LT, < GT, > LE, <= GE, >=

Examples: Evaluation order

For these examples, $\text{tag1} = 5$, $\text{tag2} = 7$, and $\text{tag3} = 10$.

$(\text{tag1} > \text{tag2}) \text{ AND } (\text{tag1} < \text{tag3})$

is evaluated in this sequence:

1. $\text{tag1} > \text{tag2} = 0$
2. $\text{tag1} < \text{tag3} = 1$
3. $0 \text{ AND } 1 = 0$

The expression evaluates to 0 (false).

$\text{tag1} > \text{tag2} \text{ AND } \text{tag3}$

is evaluated in this sequence:

1. $\text{tag2} \text{ AND } \text{tag3} = 1$
2. $\text{tag1} > 1 = 1$

The expression evaluates to 1 (true).

$\text{NOT } \text{tag1} \text{ AND } \text{tag2} > \text{tag3} ** 2$

is evaluated in this sequence:

1. $\text{NOT } \text{tag1} = 0$
2. $0 \text{ AND } \text{tag2} = 0$
3. $\text{tag3} ** 2 = 100$
4. $0 > 100 = 0$

The expression evaluates to 0 (false).

Mathematical functions

Use math functions to calculate the square root, log (natural or base 10), or trigonometry ratios (in radians or degrees) of a tag.

This function	Returns this value
SQRT (expression)	The square root of the expression
LOG (expression)	The natural log of the expression
LOG10 (expression)	The base ten log of the expression
SIN (expression)	The sine of the expression in radians
COS (expression)	The cosine of the expression in radians
TAN (expression)	The tangent of the expression in radians
ARCSIN (expression)	The arc sine of the expression in radians
ARCCOS (expression)	The arc cosine of the expression in radians
ARCTAN (expression)	The arc tangent of the expression in radians
SIND (expression)	The sine of the expression in degrees
COSD (expression)	The cosine of the expression in degrees
TAND (expression)	The tangent of the expression in degrees
ARCSIND (expression)	The arc sine of the expression in degrees
ARCCOSD	The arc cosine of the expression in degrees
ARCTAND (expression)	The arc tangent of the expression in degrees

Security functions

Use security functions to control access to your application.

These functions allow you to determine a user’s identity or security rights in order to limit access to the application based on these criteria.

This function	Returns this value
CurrentUserHasCode (<i>Security Code Letters</i>)	<p>True (1) if any of the specified security codes have been assigned to the user; false (0) if not.</p> <p>If checking multiple security codes, do not type a space between the security code letters.</p> <p>For example: CurrentUserHasCode (ABP) returns the value 1 if the user has been assigned one or more of the specified codes.</p>
CurrentUserName()	<p>A string containing the name of the current user.</p> <p>This function is case sensitive. All user names use uppercase letters, so make sure that you use uppercase letters in your expression.</p>

For more information about setting up security for your application, see Chapter 13.

For an example of using the CurrentUserHasCode(x) function, see page 13-6. For examples of using the CurrentUserName() function, see page 13-14.

If-then-else

If-then-else expressions carry out an action conditionally or branch actions depending on the statements in the expression. The if-then-else statements enable the expression to perform different actions in different situations and to repeat activities until a condition changes.

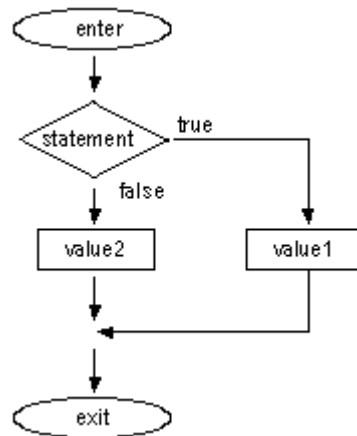
To build conditional expressions, use the relational operators and the logical operators for the statement and values.

The if-then-else structure is:

if *statement* then *value1* else *value2*

If the statement is true then the expression returns value1; if the statement is false then the expression returns value2. If the result of the statement is a non-zero value, the statement is true (and returns value1); if the result is 0, the statement is false (and returns value2).

The if-then-else structure is illustrated below.



Nested if-then-else

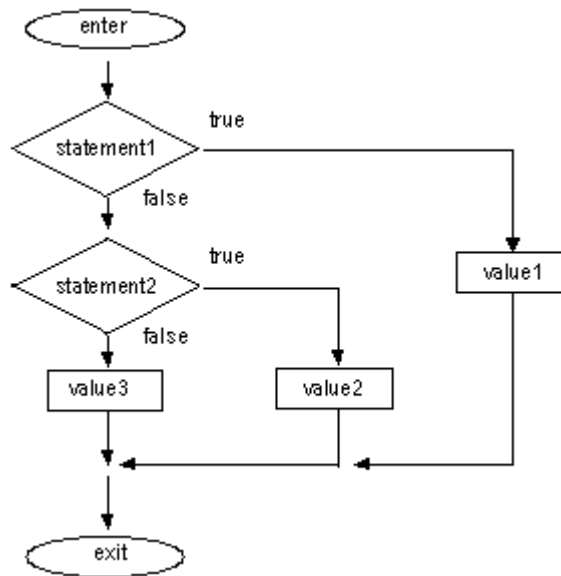
You can also nest an if-then-else structure inside the Then or Else part of an if-then-else structure.

Example 1: Nested if-then-else

This expression:

```
if statement1 then value1  
else if statement2 then value2  
else value3
```

has this interpretation:

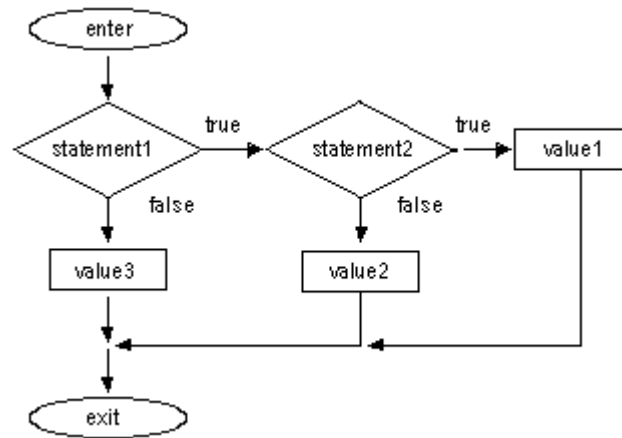


Example 2: Nested if-then-else

This expression:

```
if statement1 then  
  if statement2 then value1  
  else value2  
else value3
```

has this interpretation:



Using write expressions

Write expressions allow the operator to enter a value which is manipulated by an expression before being sent to the data source. RSVIEW[®] substitutes the value the operator enters for the placeholder in the expression, calculates the value of the expression, and writes the result to the Value control. All write expressions must contain a question mark (?) as a placeholder for the value the operator enters.

You can use write expressions with the numeric input enable button. When the operator presses the button, a keypad or scratchpad opens. The operator enters a value in the keypad or scratchpad, and this value is substituted for the ? placeholder in the write expression.

Example: Using write expressions

In this example, the operator regulates the speed of a conveyor belt by entering a value in feet or meters per second. When the operator enters the value in meters per second, the value is converted to feet per second before being passed to the data source.

The operator first indicates whether the value is in feet or meters by pushing a maintained push button. The push button has one state corresponding to feet per second, and the other state to meters per second.

Then the operator presses the numeric input enable button and enters the value for the conveyor speed in a numeric pop-up keypad. The ? character in the write expression is the placeholder for the value the operator enters.

To set up the maintained push button

1. In the Maintained Push Button Properties dialog box, in the States tab, set up these states:
 - State 0—Value: 0, Caption: Feet/S
 - State 1—Value: 1, Caption: Meters/S
2. In the Connections tab, assign a digital tag called Feet_or_meters to the Value control (either an HMI tag or a data server tag).

To set up the numeric input enable button

1. In the Numeric Input Enable Properties dialog box, in the Label tab, type the caption “Enter conveyor speed”.
2. In the Connections tab, assign a tag called Conveyor_speed to the Value control.
3. Assign this expression to the Optional Exp control:

```
if Feet_or_meters == 0 then  
?  
else  
? * 3.281
```

RSView writes the result of the expression to the Conveyor_speed tag at the data source.

Using embedded variables

This chapter describes:

- the types of embedded variables.
- where you can create embedded variables.
- creating embedded variables.
- embedded variable syntax.
- how embedded variables are updated at run time.
- how embedded variables are displayed at run time.

About embedded variables

Embedded variables allow you to display values that change dynamically at run time. You can use embedded variables in the text captions on graphic objects, and in message text. You can use multiple embedded variables in the same caption or message.

For example, you could embed a tag value and the time variable in a local message. At run time when the local message is displayed, it is updated to reflect the tag's current value as the value changes. The time is also updated as the time changes.

Embedded variables can consist of:

- numeric (analog or digital) tags, including both HMI and data server tags.
- string tags, including both HMI and data server tags.
- tag placeholders. For information about tag placeholders, see page 15-46.
- the time.
- the date.

Where you can create embedded variables

You can create embedded variables in these editors:

- **Graphic Displays**—Use this editor to insert embedded variables in the captions for graphic objects. For graphic objects with multiple states, you can insert different embedded variables in each state's caption.

For information about specific graphic objects, see Chapter 16.

- **Local Messages**—Use this editor to insert embedded variables in local messages.

For more information about local messages, see page 14-33.

- **Information Messages**—Use this editor to insert embedded variables in information messages.

For more information about information messages, see Chapter 12.

- **Alarm Setup**—Use this editor to insert embedded variables in alarm messages.

For more information about alarms, see Chapter 9.

Creating embedded variables

To create an embedded variable in a graphic object's caption

1. Open the graphic object's Properties dialog box.
2. Click the tab containing the Caption box.

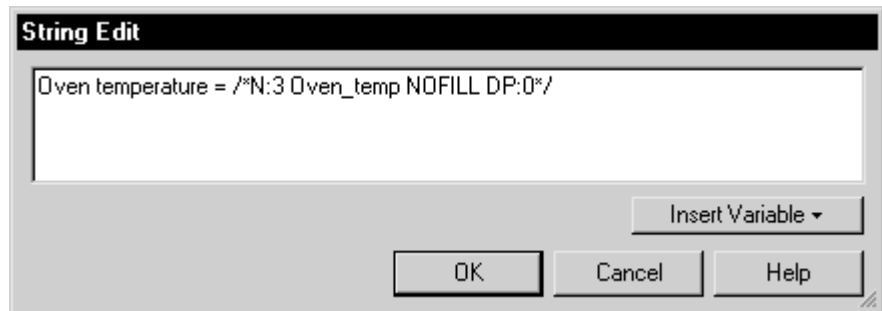
The Caption box is on the Label tab or the States tab, depending on the type of object.

For text objects, use the Text box on the General tab.

3. Click Insert Variable.
4. Click the type of variable to insert.
5. Fill in the options in the dialog box that opens. For details about the options, see Help.

To create an embedded variable in a message

1. In the Message column of the Local Messages, Information Messages, or Alarm Setup editor, right click and then click Edit String.



2. Click Insert Variable.
3. Click the type of variable to insert.

4. Fill in the options in the dialog box that opens. For details about the options, see Help.

Embedded variable syntax

Embedded variables are case sensitive, and must use specific syntax to work. Otherwise, the embedded variable is treated as a piece of text. Therefore, we do not recommend creating and editing embedded variables manually. Instead, use the Insert Variable and Edit Variable dialog boxes.

Numeric embedded variable syntax

Use numeric embedded variables to insert analog or digital tag values into text strings, including both HMI and data server tags.

Numeric embedded variables use this syntax:

```
/*N:# Tag_name Fill_character DP:##*/
```

where

N indicates it's a numeric embedded variable.

indicates the number of digits.

Tag_name is the tag to display; you can also type a tag placeholder here.

Fill_character is the fill character to use: NOFILL, ZEROFILL, or SPACEFILL.

indicates the number of decimal places.

Example: Numeric embedded variable syntax

To display the current value of a tag called `Oven_temp`, with 3 digits, no decimal places, and no fill character, you would type this:

```
/*N:3 Oven_temp NOFILL DP:0*/
```

String embedded variable syntax

Use string embedded variables to insert string tag values into text strings.

String embedded variables use this syntax:

```
/*S:# Tag_name*/
```

where

`S` indicates it's a string embedded variable.

`#` indicates the number of characters if you select a fixed number of characters; type 0 if you don't want to use a fixed number.

`Tag_name` is the tag to display; you can also type a tag placeholder here.

Example: String embedded variable syntax

To display the current value of a string tag called `Blower_status`, with a fixed length of 20 characters, you would type this:

```
/*S:20 Blower_status*/
```

Time and date embedded variable syntax

Use time and date embedded variables to insert the current time or date into text strings.

Time and date embedded variables use this syntax:

*/*Time_date_format*/*

where

Time_date_format uses one of these character sequences:

These characters	Specify this format
SD	Short date
LD	Long date
SDT	Short date and time
LDT	Long date and time
T	Time
TSD	Time and short date
TLD	Time and long date

Example: Time and date embedded variable syntax

To display the time followed by the short date, you would type this:

*/*TSD*/*

A space is placed between the time and date when the embedded variable is displayed at run time.

How embedded variables are updated at run time

At run time, this is how embedded variables are displayed and updated:

- **Graphic objects**—When a display containing a graphic object that uses an embedded variable is open, the value of the embedded variable is updated whenever a new tag value is read from the data source. For time and date embedded variables, the time and date are updated as the system time and date change.
- **Local messages**—When a display containing a local message display object is open, and the message the object is displaying contains an embedded variable, the value of the embedded variable is updated whenever a new tag value is read from the data source. For time and date embedded variables, the time and date are updated as the system time and date change.
- **Information messages**—The value of the embedded variable is read when the information message is first displayed. It is not updated after that.

If the message is printed, it is printed using the value the variable had when the message was first displayed. This value is retained if you shut down and restart the application.

- **Alarm messages**—The value of the embedded variable is read when the alarm occurs, and is displayed in the message associated with the alarm. It is not updated after that.

If the message is printed, it is printed using the value the variable had when the alarm first occurred. This value is retained if you shut down and restart the application.

How embedded variables are displayed at run time

If there is no valid data available for the embedded variable, the variable is replaced with question marks (?). This could occur when a display first opens and the data has not arrived yet, or when there is a problem that prevents communication with the data source.

If a string or numeric embedded variable has been set up but no tag has been assigned, the embedded variable is replaced with asterisks (*).

Numeric embedded variables

The value shown for a numeric embedded variable depends on whether the tag value is a floating-point number or an integer. Integer values are displayed as is. Floating-point values are rounded to fit the specified number of digits for the variable.

For example, if the variable is set up to show 6 digits, 1234.56 is rounded to 1234.6. 1234.44 is rounded to 1234.4. The decimal counts as one of the digits.

For more information about how values are rounded, see page 7-2.

If the tag value, including the decimal point and minus sign, contains more digits than specified for the variable, the numeric variable is replaced with asterisks.

For example, if the variable is set up to show 6 digits, and the tag value is -123456, the variable will be replaced with asterisks.

Number formats

The numeric variable uses the number format you set up on the runtime computer. For example, if the runtime computer is set up to use a comma for the decimal symbol, the numeric variable uses a comma for the decimal symbol.

For information about specifying time, date, and number formats:

- in Windows® 2000 and Windows XP, see page 24-18.
- on a PanelView Plus™ terminal, see the *PanelView Plus Operator Terminals User Manual*.
- on a VersaView® CE terminal, see the *VersaView CE Terminals User Manual*.

String embedded variables

For string embedded variables that do not use a fixed number of characters, the entire string tag value is displayed, unless a null character is read. Nothing after a null character is displayed.

If a fixed number of characters is used, the variable displays the value of the tag up to the number of characters specified, unless a null character is encountered before the specified length. Nothing is displayed after a null character. If necessary, spaces are used to make up the required number of characters.

Null characters have a hex value of 0. The null character indicates the end of string input. It does not add to the actual string length.

Time and date embedded variables

For embedded variables that show both the time and the date, a space is placed between the time and date when the embedded variable is displayed at run time.

Time and date formats

Time and date embedded variables use the time and date formats you set up on the runtime computer. For example, if you specify the short date format, at run time the display uses the short date format that the runtime computer uses.

For information about specifying time, date, and number formats:

- in Windows 2000 and Windows XP, see page 24-18.
- on a PanelView Plus terminal, see the *PanelView Plus Operator Terminals User Manual*.
- on a VersaView CE terminal, see the *VersaView CE Terminals User Manual*.

Chapter 21

Using macros

This chapter describes:

- using macros to assign values to tags.
- using the Macros editor.
- when to use macros.
- running macros when tags or expressions change value.
- where to assign macros.

Using macros to assign values to tags

A macro is a list of tag assignments stored in a text file, in the format `<tag>=<value>`. Each assignment assigns a value to a tag. The value can be in the form of another tag, an expression, a numeric constant, or a string.

Examples: Using macros to set tag values

`Tag1 = 8`

Sets the value of Tag1 to 8.

`Tank1\Message = "Tank1 overflow"`

Sets the string tag Tank1\Message to Tank1 overflow.

`Tag1 = Tag2`

Sets the value of Tag1 to be the same as Tag2.

`Tag1 = Tag1 + 1`

Increases the value of Tag1 by 1.

Tag1 = if (Tag2 < Tag1) then 4 else 3
Performs the if-then-else calculation and stores the result in Tag1.

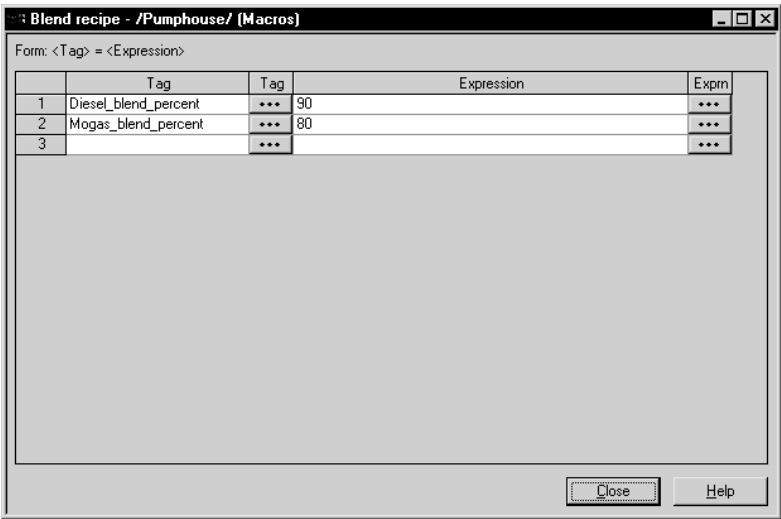
1Pump = {Industry-2} + {2Pump}
Adds the values of Industry-2 and 2Pump and stores the result in 1Pump.

Brackets surround Industry-2 because of the dash in the name.
Brackets surround 2Pump because the name starts with a number. No brackets are used for 1Pump because this name is on the left side of the equal sign.

For more information about expression syntax, see Chapter 19.

Using the Macros editor

Use the Macros editor to create macros.



For details about using the Macros editor, see Help.

When to use macros

You can assign macros to run when:

- the application starts or shuts down.
- a graphic display opens or closes.
- a user logs in or out.
- a specified tag or expression changes to a new non-zero value (using global connections).
- an operator presses a macro button.

At run time, when the macro runs, the values are sent to the tags at the data source.

IMPORTANT

At run time, the tag assignments are executed asynchronously. That is, the system does not wait for the completion of one tag assignment before executing the next. Therefore, do not rely on the order of assignments to control your process.

Running macros when tags or expressions change value

You can use global connections to run macros when tags or expressions change value. This means you can use the data source to trigger the macro to run.

RSView® allows you to create up to five macros for use with global connections. The macros must be named Macro1, Macro2, Macro3, Macro4, and Macro5 in order to work with global connections.

For more information about global connections, see Chapter 8.

Example: Using macros to reset tag values

This example shows you how to run a macro whenever the operator needs to reset production information tags to known values.

The macro writes the desired values to the tags whenever the operator presses a momentary push button.

1. Create a memory tag called ResetProdData.
2. Create a momentary push button with the caption “Reset Production Data.” Assign the ResetProdData tag to the Value control.
3. Create a macro called Macro1, with these tag assignments:

```
TotalProductionUnits=0  
LineDownTime=0  
TotalRejects=0
```

4. In the Global Connections editor, assign the ResetProdData tag to the Remote Macro1 control.

When the operator presses the Reset Production Data button, the value of the ResetProdData tag changes from 0 to 1. This tells RSVIEW to run Macro1, which writes the specified values to the tags in the macro.

Where to assign macros

Once you’ve created the macros you want to use, assign the macros in these editors:

In this editor	Do this
Startup	Assign application startup and shutdown macros.
Graphic Displays	Assign macros to run when displays open or close, using the Display Settings dialog box.

In this editor	Do this
User Accounts	Assign macros to run when users log in and log out.
Global Connections	Specify the tags or expressions that will run the macros named Macro1 to Macro5.

Setting up display navigation

This chapter describes:

- what display navigation is.
- developing a hierarchy of displays.
- testing display navigation.
- using graphic objects to navigate.
- controlling display changes remotely.

This chapter describes methods for navigating between graphic displays. For information about navigating between objects in a graphic display, see page 16-9.

About display navigation

The term display navigation refers to the way the operator moves between the graphic displays that make up an application.

Use these methods to set up display navigation for your application:

- Develop a hierarchy of graphic displays, to chart how users will navigate the application.
- Determine which users will have access to which parts of the application.
- Create graphic objects that the operator can use to navigate the application.

- Use the Remote Display Number control to automatically control display changes. The use of this control is optional. Assign it in the Global Connections editor.
- In the Startup editor, specify the graphic display to open when the application starts.
- Set up security so that only authorized users have access to the application or parts of the application.

For information about setting up security, see Chapter 13.

Developing a hierarchy of displays

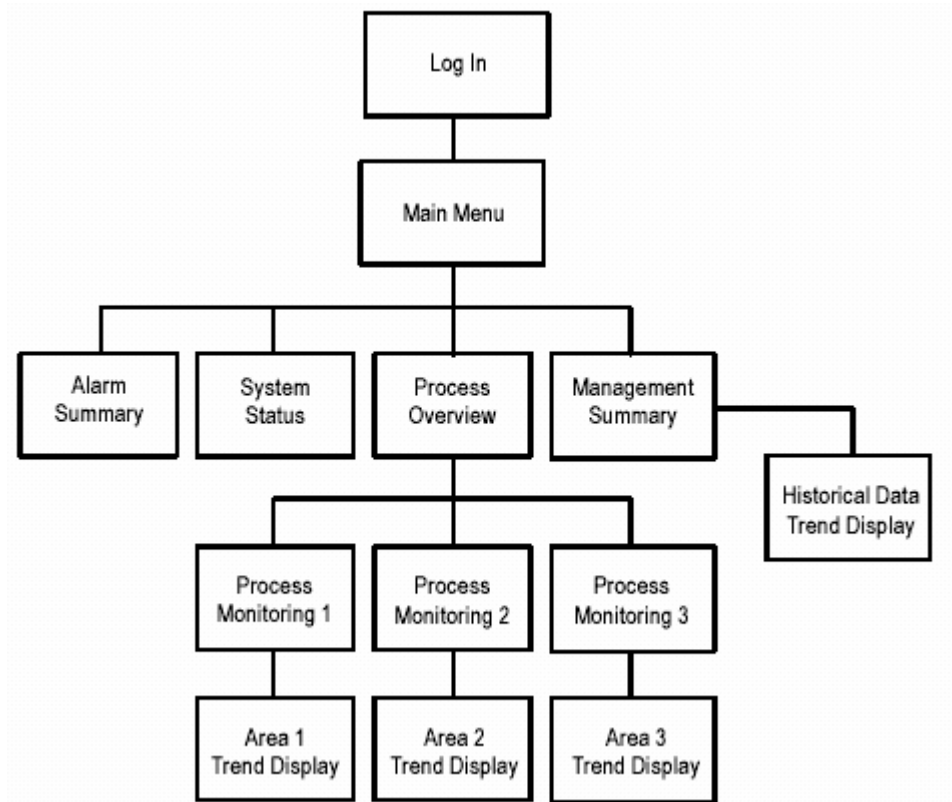
A display hierarchy is a series of graphic displays that provide progressively more detail as users move through them. Design your display hierarchy to meet the needs of the various users, including managers, supervisors, and operators.

If you plan to use security, determine which groups of users need access to which displays, and decide where in the hierarchy to locate login, logout, and shutdown buttons. For information about setting up security, see Chapter 13.

A display hierarchy could include:

- an initial graphic display for logging in.
- a graphic display that serves as a menu.
- an overview of the plant.
- a comprehensive display of each process being monitored.
- process-specific displays that provide more detail.
- management summary displays.
- trend displays of historical and current data.

The following illustration shows a sample display hierarchy.



Testing display navigation

Once you set up display navigation for your application, test the application to make sure that navigation flows smoothly and that you have avoided problems like these:

- A graphic display contains no buttons for moving forward or back.
- When a graphic display closes, no other display is open and there is no way to continue using the application.

For information about testing your application, see page 23-2.

Using graphic objects to navigate

Use these graphic objects to navigate through the displays in the application:

Use this object	To do this
Goto display button	Open the specified graphic display.
Goto configure mode button	Stop the application and open the RSView® ME Station™ dialog box.
Return to display button	Close the current display and open the previous display.
Close display button	Close the current display. Can send a value to a tag when the display closes.
Display list selector	Provide a list of graphic displays so the operator can select which display to open.
Shutdown button	Stop the application and exit RSView® ME Station.

The behavior of these graphic objects depends on which types of graphic displays are used. See the following sections for details.

For detailed information about setting up graphic objects, see Help.

Display type

The display type you use gives you additional control over how the operator navigates between displays. For example, use the On Top type to keep a graphic display on top at all times, even when another display has focus. Or use the Replace type if you want a display to replace all other open displays when it opens.

For more information about display types, see page 14-14.

Goto display buttons

When you set up a goto display button, specify the graphic display to open when the operator presses the button at run time.

The specified display doesn't open if the operator does not have security access for the display.

You can also assign a parameter file that assigns tags to tag placeholders in the display when the display opens. For more information about parameter files, see page 14-28.

For information about setting up goto display buttons, see Help.

How display types affect the button's behavior

The goto display button's behavior at run time also depends on which types of graphic displays are already open and which type of display it is opening:

- If the graphic display assigned to the button is a Replace display, it closes any open On Top or Replace displays. It does not close any On Top displays that use the Cannot Be Replaced option.
- If the display assigned to the button is already open, but does not have focus, pressing the button gives the display focus.

The operator cannot use the goto display button to open Replace displays if display change is currently controlled remotely (using global connections). But the operator can still open On Top displays.

Goto configure mode buttons

When the operator presses a goto configure mode button at run time, the current application stops running and the RSVIEW ME Station dialog box opens.

While in configure mode, the operator can use RSVIEW ME Station to change applications, application settings, and terminal settings. The

operator can also delete log files. For more information about the RSVIEW ME Station dialog box, see Chapter 24.

Return to display buttons

When the operator presses a return to display button at run time, the graphic display that the button is on closes and the display that was previously open reopens.

The current display does not close if:

- the display change is currently controlled remotely (using global connections).
- there were no previously opened Replace displays.
- the operator does not have security access for the previous display. This can only occur if a new user logs in using a login button in the current display.

The return to display button only goes back to the most recent display. It doesn't go back through a series of displays.

For information about setting up return to display buttons, see Help.

How display types affect the button's behavior

Both the current display and the previous display must be Replace displays. When the operator presses the return to display button:

- If the graphic display that is closing is a Replace display, the display closes and the previously opened Replace display opens. Any On Top displays that were previously open with the Replace display are not reopened.
- If the graphic display that is closing is an On Top display, the display closes but no display is reopened.

We therefore recommend that you use return to display buttons in Replace displays only.

Example: Navigating through displays

This example uses the display hierarchy illustrated on page 22-3, and shows what happens as the operator navigates through the hierarchy. The graphic displays are all Replace displays.

1. In the Main Menu display, the operator uses a display list selector to open the Process Overview display.
2. In the Process Overview display, the operator presses a goto display button to open the Process Monitoring 1 display.
3. After viewing the state of the process, the operator presses a return to display button to close the current display and reopen the Process Overview display.
4. In the Process Overview display, the operator presses a return to display button. Which display opens? The Process Monitoring 1 display (because this was the previously opened display).

To return to the Main Menu display from the Process Overview display, the operator would have to press a goto display button that is set up to open the Main Menu display.

Close display buttons

When the operator presses a close display button at run time, the graphic display that the button is on closes. You can set up the button to write out a value when the display closes.

If the graphic display that is closing is a Replace display, and the display change is controlled remotely, the display does not close. If the display does not close, the close value, if any, is not written out.

If the display change is controlled by the operator and the graphic display that is closing is a Replace display, if there are no On Top displays open, an empty window is displayed. The operator will not be

able to use the application again (unless a remote display change occurs or an alarm, activity, or information message display opens).

We therefore recommend that you use close display buttons in On Top displays only.

For information about setting up close display buttons, see Help.

Display list selectors

Use the display list selector to show a list of graphic displays that the operator can choose from. The operator can scroll through the list and select the graphic display to open.

The specified display doesn't open if the operator does not have security access for the display.

You can also assign a parameter file that assigns tags to tag placeholders in the display when the display opens. For more information about parameter files, see page 14-28.

For information about setting up display list selectors, see Help.

How display types affect the selector's behavior

The display list selector's behavior at run time also depends on which types of graphic displays are already open and which type of display it is opening:

- If the selected graphic display is a Replace display, it closes any open On Top and Replace displays. It does not close On Top displays that use the Cannot Be Replaced option.
- If the selected display is an On Top display, it opens on top of the current display. The current display does not close.

The operator cannot use the display list selector to open Replace displays if display change is currently controlled remotely. But the

operator can still open On Top displays (with or without the Cannot Be Replaced option).

Selecting the display to open

The operator can scroll through the list and select displays using the key button graphic objects, or, if the list has the input focus, by using the arrow keys and Enter key on a keypad or external keyboard.

You can link key buttons to a specific display list selector, or set up the buttons to work with whichever object is selected in the graphic display.

For information about input focus, see page 16-9. For information about linking buttons to the display list selector, see page 16-11.

Shutdown buttons

When the operator presses the shutdown button at run time, the application stops and RSView ME Station closes.

To prevent an unauthorized user from stopping the application, assign visibility animation to the shutdown button. For details, see page 13-10.

For information about setting up shutdown buttons, see Help.

Controlling display changes remotely

To control display changes remotely, you can set up the data source to open graphic displays using global connections.

Global connections are controls that apply to your entire runtime application. Global connections allow the data source to control or interact with your application at run time.

For example, the Remote Display Number control is a global connection that you can use to control display changes from the data source. You can also use global connections to print graphic displays from the data source, to run macros from the data source, and to

control the date and time displayed on the runtime terminal. For more information about global connections, see Chapter 8.

Specifying startup settings, testing applications, and creating runtime applications

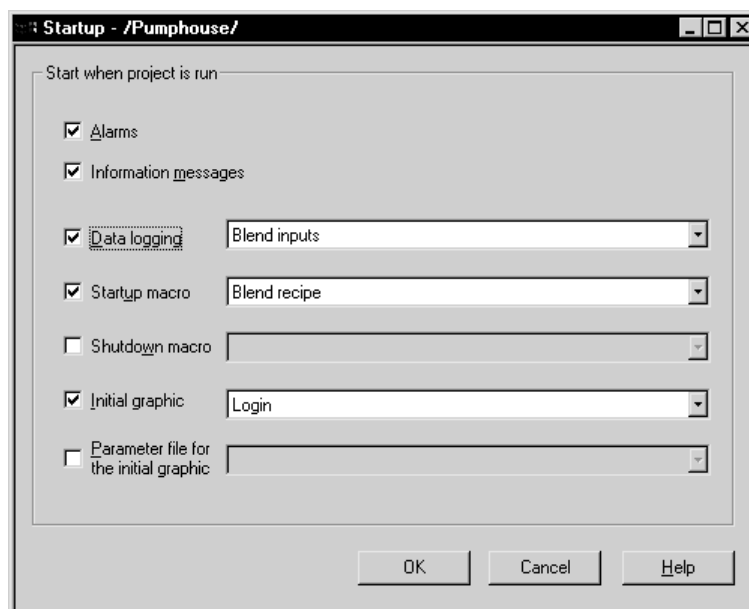
This chapter describes:

- specifying startup settings.
- testing your application.
- creating runtime application files.

Specifying startup settings

Use the Startup editor to specify which application processes and components to start when the application starts at run time.

You can specify startup settings once you've set up all the parts of the application, or you can specify processes and select components in the Startup editor as you create them.



For detailed information about the options in the Startup editor, see Help.

Testing your application

You can test your application in RSVIEW® Studio™ at any time during the development process, to make sure that everything works the way you intend.

If the development computer is connected to the data source, you can test all functions of the application, including communications and alarm monitoring.

An RSVIEW ME Station™ emulator opens on the development computer and runs the application. This runtime version of the

application is a temporary version for testing use only. You cannot run it on another computer.

There is a two-hour time limit for test running the application in RSVIEW Studio.

The procedure in this section shows you how to test your entire application. For information about testing a single graphic display, see page 14-12.

To test your application in RSVIEW Studio



Test Application tool

- 1.** On the Application menu, click Test Application, or click the Test Application tool.
- 2.** Test your application.
- 3.** To stop your application, press a shutdown button, or type the character 'x.'

Make sure you provide the operator with a method for shutting down the application at run time. For more information about methods for shutting down applications, see page 24-6.

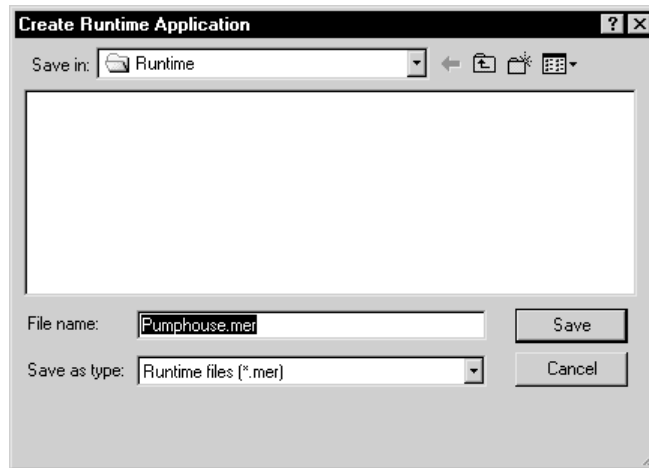
Once you've tested your application to make sure everything works the way you intend, create the runtime application file and transfer the file to the runtime computer.

Creating runtime application files

Before you can run your application, you must create a runtime version. When you create the runtime version, RSVIEW Studio compiles all of the necessary application information into a single file with the extension .mer.

To create a runtime application

- In RSView Studio, with the application open, on the Application menu click Create Runtime Application.



For details about the options in the Create Runtime Application dialog box, see Help.

For information about transferring the runtime application:

- to a Windows® 2000 or Windows XP platform, see Chapter 24.
- to a PanelView Plus™ or VersaView® CE terminal, see Chapter 25.

Running applications in Windows 2000 or Windows XP

This chapter describes:

- moving applications to the runtime computer.
- starting RSView® ME Station™.
- loading and running applications.
- shutting down applications.
- editing device short cuts.
- looking up contact information for technical support.
- setting up FactoryTalk™ Diagnostics at run time.
- setting up serial ports for KEPServerEnterprise.
- setting up RSLinx® Enterprise™ communication drivers.
- specifying the printers to use at run time.
- specifying startup options for RSView ME Station.
- deleting log files on the runtime computer.
- specifying the time, date, and number formats to use at run time.
- using the DeskLock tool.

Summary of steps

Follow these steps to:

- install the necessary hardware and software on the runtime computer.
- transfer your Windows® 2000 or Windows XP application to the runtime computer.
- set up options in RSVIEW ME Station.

For information about installing RSVIEW ME Station, see the *RSVIEW Machine Edition Installation Guide*.

Installing hardware and software on the runtime computer

1. If you will be printing displays, alarms, or diagnostics messages, set up printer connections on the runtime computer.

For more information, see page 24-11.

2. If you are using RSLinx Enterprise, set up communications as described in Chapter 5.
3. If you are using RSLinx® for RSVIEW as the OPC® server on the runtime computer, install RSLinx for RSVIEW on the runtime computer.
4. If you are using RSLinx for RSVIEW on a remote computer, install RSLinx for RSVIEW on the remote computer.
5. If you are using an OPC server other than RSLinx Enterprise or RSLinx for RSVIEW, install the OPC server software on the runtime computer or on another computer on the network.

For installation information, see the documentation supplied by your OPC server vendor. For information about OPC, see Chapter 5.

6. If your application uses third-party ActiveX® objects, install and register the Windows 2000 or Windows XP version of the objects on the runtime computer.

For information about ActiveX objects, see page 15-25.

7. If the runtime computer uses different time, date, or number formats than the development computer, specify the time, date, and number formats to use at run time.

For more information, see page 24-18.

8. If desired, use the DeskLock tool to prevent users from switching to another software application or using the desktop at run time.

For more information, see page 24-19.

Transferring the application

- Move the application to the Windows 2000 or Windows XP runtime computer.

For more information, see page 24-4.

Setting up options in RSView ME Station

1. On the runtime computer, start RSView ME Station.

For more information, see page 24-5.

2. Load the application.

For more information, see page 24-5.

3. Edit device shortcuts, if necessary.

For more information, see page 24-7.

4. Set up FactoryTalk Diagnostics on the runtime computer (if you have not already done so).

For more information, see page 10-8.

5. If you will be using KEPServerEnterprise™, specify serial port IDs.

For more information, see page 24-9.

6. If you will be using RSLinx Enterprise, set up communication drivers (if you have not already done so).

For more information, see page 24-10.

7. Specify the printers to use.

For more information, see page 24-11.

8. Specify startup options for RSView ME Station.

For more information, see page 24-13.

Once you've completed these steps, you're ready to run the application. For information about running your application, see Chapter 26.

Moving applications to the runtime computer

The runtime application file has the extension .mer. You can use any standard file transfer method to copy your runtime application from the development computer to the runtime computer.

You can:

- copy the application file from the development computer to a floppy disk, and then from the floppy disk to the runtime computer.
- if the application file is too large to fit on a floppy disk, use a larger storage device such as a Zip® disk.
- if the development and runtime computers are on the same network, use Windows Explorer or My Computer to move the file.

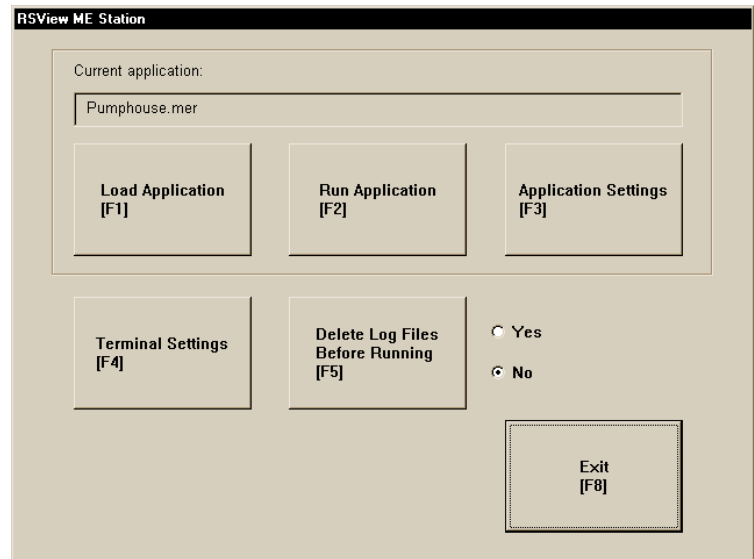
For information about creating the runtime application file, see Chapter 23.

Starting RSVIEW ME Station

To start RSVIEW ME Station

- On the Windows Start menu, select Programs, Rockwell Software, RSVIEW Enterprise, and then click RSVIEW ME Station.

The RSVIEW ME Station dialog box opens.



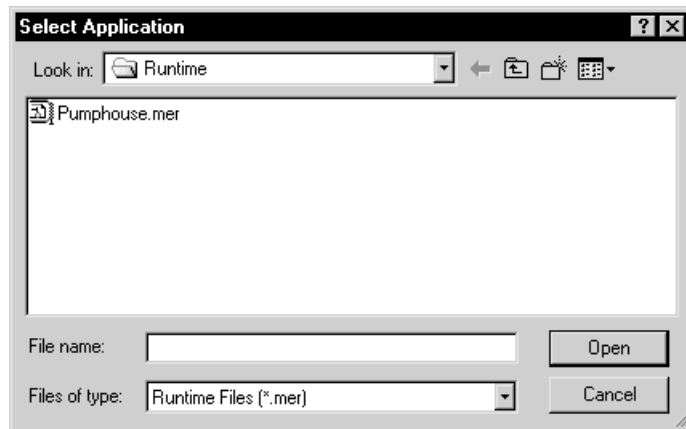
For information about specifying startup options for when RSVIEW ME Station starts, see page 24-13.

Loading and running applications

You can run any runtime application that is on the runtime computer. Runtime applications have the extension .mer.

To load and run the application

1. In the RSView ME Station dialog box, click Load Application.



2. Navigate to the folder containing the application's .mer file, and then click the file name.
3. Click Open.
4. To replace the runtime computer's RSLinx Enterprise settings with the application's RSLinx Enterprise settings, click Yes. To keep the runtime computer's RSLinx Enterprise settings, click No.

The application name is displayed in the Current application box in the RSView ME Station dialog box.

5. To run the application, click Run Application.

Shutting down applications

To shut down an application, use one of these methods

- Press a shutdown button in a graphic display.
- If the application is set up to use a title bar with a Control box, click the Close button at the right end of the title bar.

- If the application is set up to use a title bar with a Control box, on the Control menu at the left end of the title bar, click Close.



For information about using a title bar in graphic displays, see page 4-14. For information about preventing unauthorized users from shutting down applications, see page 13-10.

What happens when the application shuts down

When the application shuts down:

- if you assigned a shutdown macro (in the Startup editor), the macro runs, assigning values to tags, and then the application stops.

For information about the Startup editor, see Chapter 23.

- RSVIEW ME Station closes.

Changing application settings

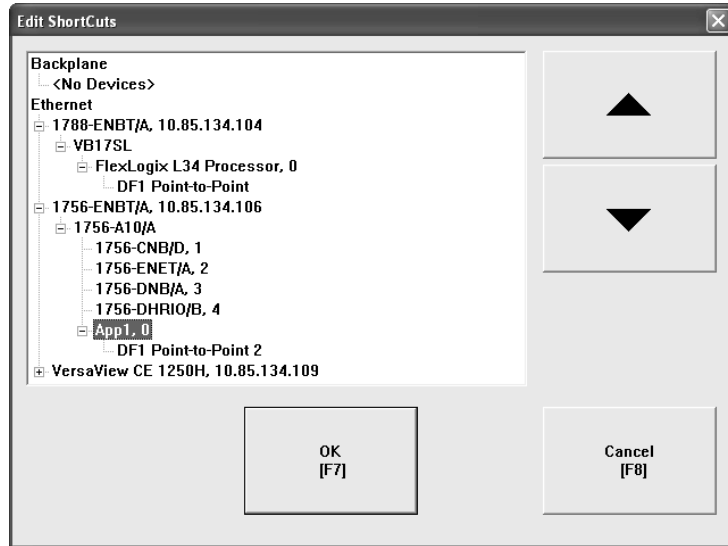
Editing device shortcuts

You can use RSVIEW ME Station to edit device shortcuts that have been set up in the application.

Before editing device shortcuts, load the application containing the device shortcut, as described on page 24-5.

To edit device shortcuts

1. In the RSView ME Station dialog box, click Application Settings.
2. Double-click Device Shortcuts.
3. Double-click the name of the shortcut to edit.
4. In the Edit ShortCuts dialog box, click the device you want the shortcut to point to, and then click OK.



Looking up contact information for technical support

To look up technical support contact information

1. In RSView ME Station, click Terminal Settings.
2. Double-click About RSView ME Station.
3. Click Technical Support.

The telephone number, fax number, and URL for technical support are displayed.

Setting up FactoryTalk Diagnostics on the runtime computer

You can set up FactoryTalk Diagnostics on the runtime computer using the RSView ME Station dialog box.

To set up FactoryTalk Diagnostics on the runtime computer

1. In RSView ME Station, click Terminal Settings.
2. Double-click Diagnostics Setup.

The FactoryTalk Diagnostics Setup dialog box opens.

3. Set up FactoryTalk Diagnostics, as described on page 10-8.

Setting up serial ports for use with KEPServerEnterprise

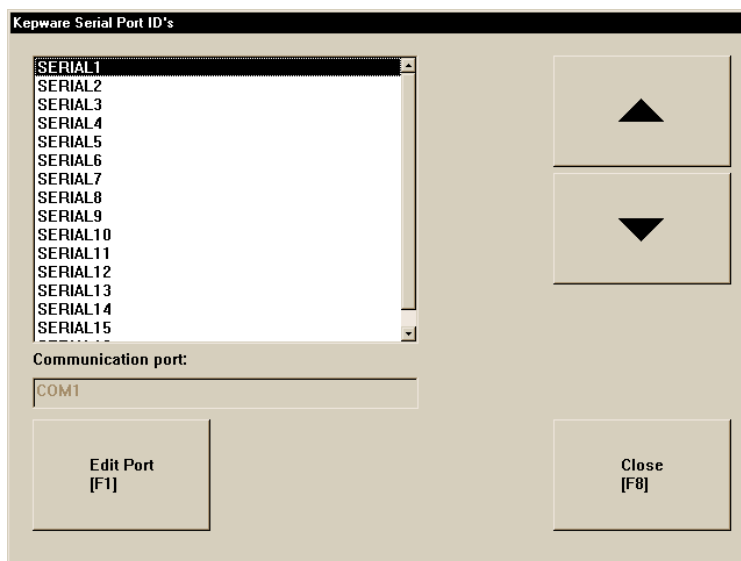
If you plan to use KEPServerEnterprise and serial communications, you must specify which COM port to use.

For information about setting up communications in KEPServerEnterprise, see KEPServerEnterprise Help.

To specify the COM port to use for serial communications

1. In the RSView ME Station dialog box, click Terminal Settings.
2. Double-click Networks and Communications.

3. Double-click KEPServer Serial Port ID's.



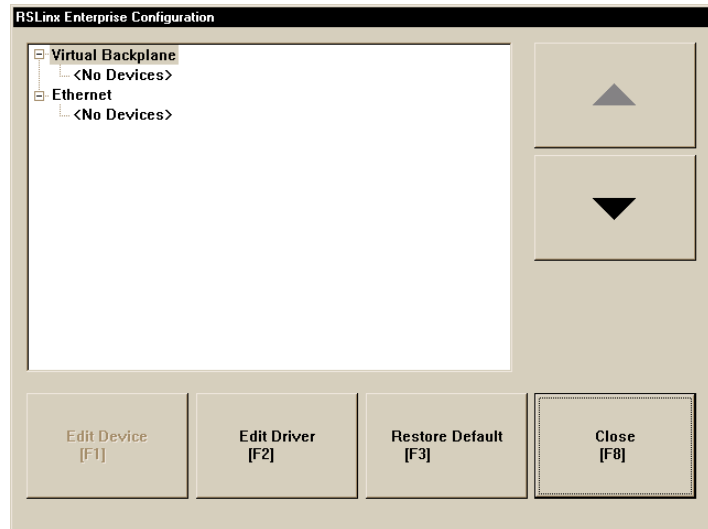
4. In the Kepware Serial Port ID's dialog box, click the serial port ID you specified when you set up the KEPServerEnterprise channel.
5. Click Edit Port.
6. In the Communication Ports dialog box, click the COM port to use for KEPServerEnterprise communications.

Setting up RSLinx Enterprise communication drivers

Use RSLinx Enterprise to set up communication drivers for your runtime application. You can set up the drivers directly in RSLinx Enterprise, or open RSLinx Enterprise by using the RSView ME Station dialog box.

To set up the RSLinx Enterprise communication driver to use at run time

1. In the RSView ME Station dialog box, click Terminal Settings.
2. Double-click Networks and Communications.
3. Double-click RSLinx Enterprise Communications.



4. To edit a driver, select it and then click Edit Driver.
5. To edit a device, select it and then click Edit Device.

For information about setting up RSLinx Enterprise drivers and devices, see the RSLinx documentation.

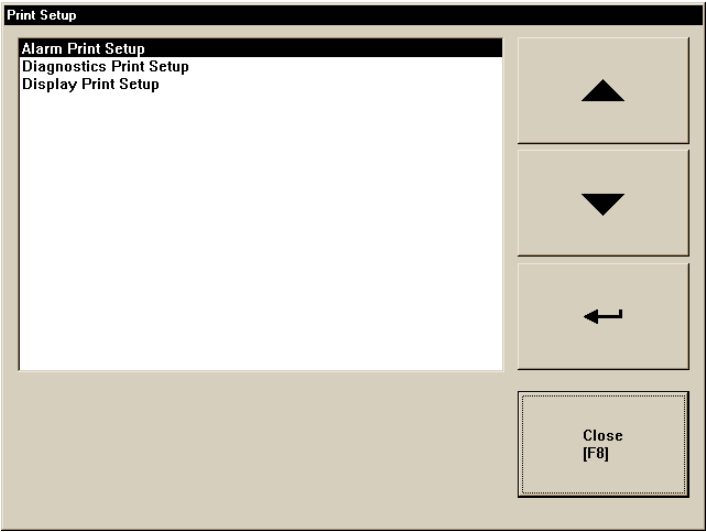
Once the driver is set up, RSView ME Station automatically starts the driver software when you run the application.

Specifying the printers to use at run time

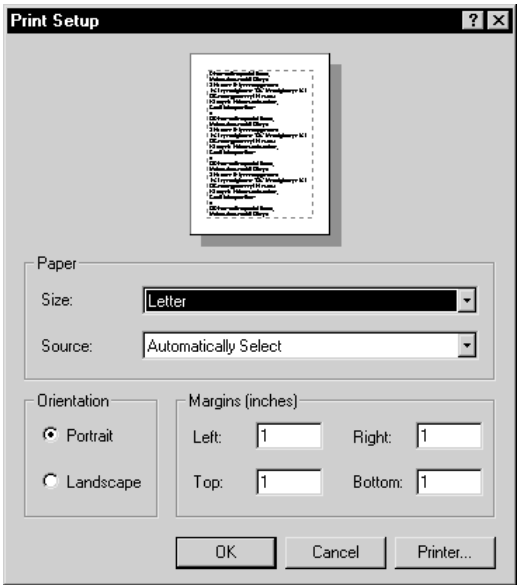
You can use local or network printers to print alarm messages, diagnostics messages, and graphic displays at run time. If desired, you can use a different printer for each type of printing.

To specify the printers to use at run time

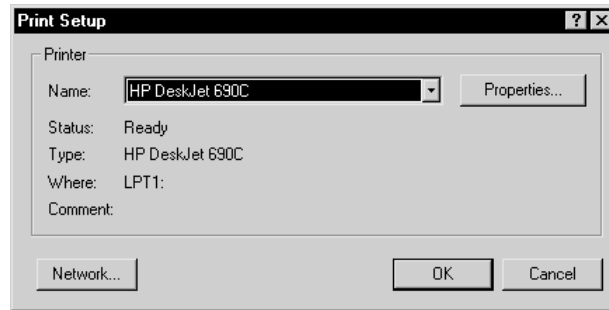
- 1. In the RSVIEW ME Station dialog box, click Terminal Settings.
- 2. Double-click Print Setup.



- 3. Double-click the type of printing to set up.



4. Click Printer.



5. Modify the print options as required.

For detailed information about printer options, refer to your Windows documentation.

Specifying startup options for RSView ME Station

This section describes how to start RSView ME Station automatically when Windows starts, and describes startup options for RSView ME Station.

When you start RSView ME Station, you can:

- run an application.
- load an application.
- automatically delete an application's log files before running the application.
- replace RSLinx Enterprise communications on the runtime computer with the application's settings.

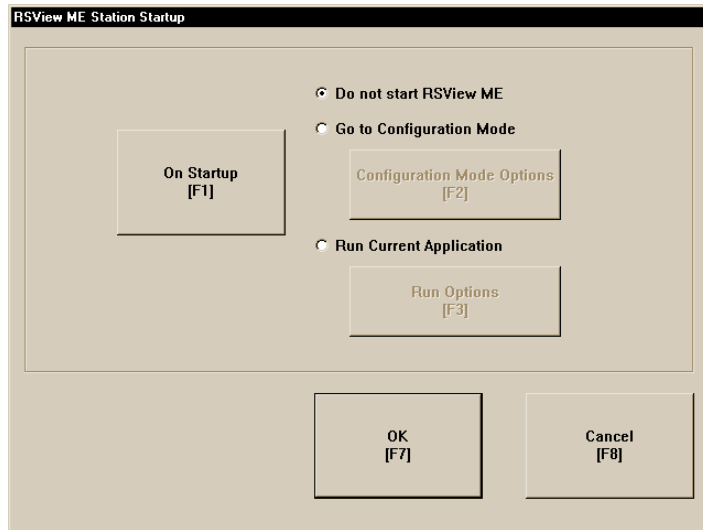
By default, the option to start RSView ME Station when Windows starts is turned off. The settings in this section apply only if you want RSView ME Station to start automatically when Windows starts.

To start RSVIEW ME Station when Windows starts and run an application

1. In the RSVIEW ME Station dialog box, load the application that you want to run.

For details, see page 24-5.

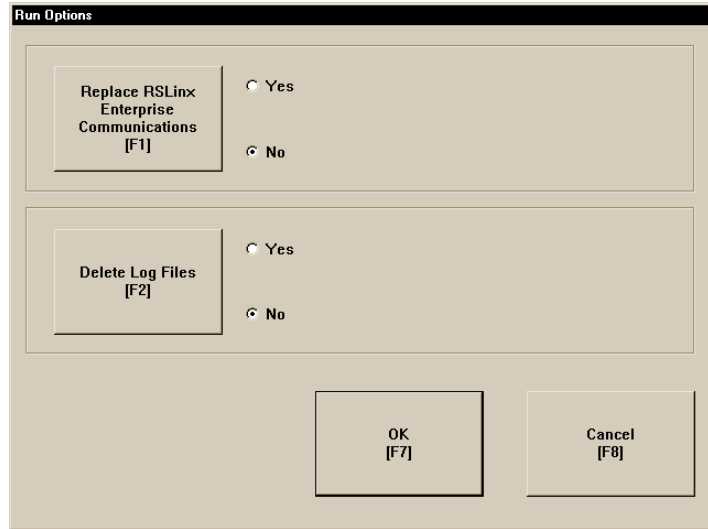
2. Click Terminal Settings.
3. Double-click RSVIEW ME Station Startup.



4. Click Run Current Application.

This option is not available if you have not loaded an application yet.

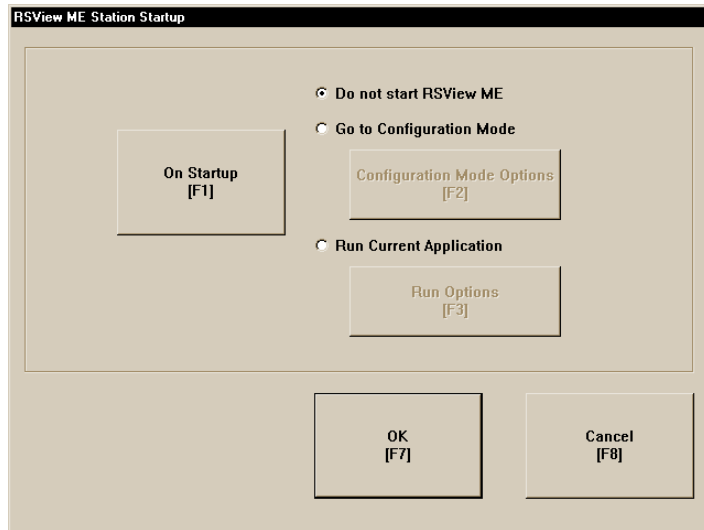
5. Click Run Options.



6. Specify whether to replace the RSLinx Enterprise communication settings on the runtime computer with the application's settings when the application starts.
7. Specify whether to delete the application's log files on startup.

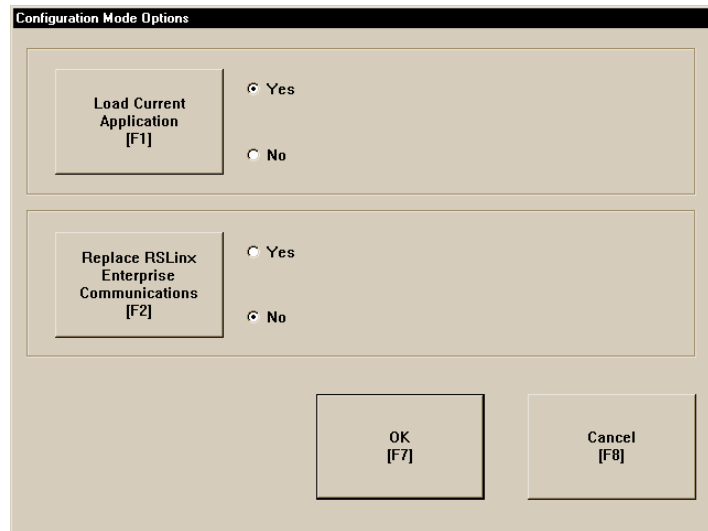
To start RSView ME Station when Windows starts without running an application

1. In the RSView ME Station dialog box, click Terminal Settings.
2. Double-click RSView ME Station Startup.



3. Click Go to Configuration Mode.

4. Click Configuration Mode Options.



5. Specify whether to load the current application when RSVIEW ME Station starts.

This option is not available if you have not loaded an application.

6. Specify whether to replace the RSLinx Enterprise communication settings on the runtime computer with the application's settings when the application starts.

This option is not available if you have not loaded an application.

Deleting log files on the runtime computer

When you run your application, RSVIEW ME Station stores log files for alarms and data logging (if you use these features). When you start RSVIEW ME Station, you can delete the alarm and data log files for the loaded application, or for all the applications on the runtime computer.

Running a newer version of the application

If you run a newer version of an application, the alarm log file for the older version is deleted automatically. The data log file for the older version is retained, to allow the display of historical data in trends.

For more information about the alarm log file, see page 9-11. For more information about data log files, see page 11-6.

Deleting log files manually

To delete log files for the loaded application

- In the RSView ME Station dialog box, click Yes beside the Delete Log Files Before Running button.

All alarm and data log files for the loaded application are deleted.

To delete log files for all applications on the runtime computer

1. In the RSView ME Station dialog box, click Terminal Settings.
2. Double-click Delete Log Files.
3. Click Yes.

Log files for data log models that use the default path are deleted. All alarm log files are deleted.

Specifying time, date, and number formats

Time, date, and number formats are used by these graphic objects:

- numeric display
- gauge
- time and date display

- trend
- alarm banner
- alarm list

The time and date formats are also used when printing the alarm history report (using the print alarm history button).

If the runtime computer uses different time, date, or number formats than the development computer, specify the time, date, and number formats to use at run time.

You can also change these formats after the application is running, but you must shut down and restart the application for the changes to take effect.

To specify time, date, and number formats

1. On the Windows Start menu, select Settings, and then click Control Panel.
2. Double-click Regional Options (in Windows 2000) or Regional Settings (in Windows XP).
3. Specify the locale to use, and then select the number, time, and date settings to use.

For more information, see your Windows documentation.

Using the DeskLock tool

Use the DeskLock tool to prevent users from switching to another software application or having access to the desktop at run time.

This tool can have far-reaching effects on your operating system. The DeskLock tool replaces the standard Windows 2000 or Windows XP non-configured desktop with a customized desktop. The customized desktop is intended to prevent operators from having access to other

applications and operating system functions such as restarting Windows or shutting down tasks.

IMPORTANT

Before using the tool, read the DeskLock Help file completely.

To open the DeskLock tool

- On the Windows Start menu, select Programs, Rockwell Software, RSVIEW Enterprise, Tools, and then click DeskLock.

For details about using the tool, see the tool's Help.

Transferring applications to a PanelView Plus or VersaView CE terminal

This chapter describes:

- starting RSVIEW® ME Station™.
- setting up communication drivers to use to transfer applications to a PanelView Plus™ or VersaView® CE terminal.
- transferring applications and TrueType™ fonts to PanelView Plus or VersaView CE terminals.
- uploading applications from a terminal to the development computer.
- comparing applications on the development computer with applications on the terminal.

For information about using your application at run time, see Chapter 26.

For information about using RSVIEW ME Station on the terminal, including setting up communications, printers, and input devices, see the *PanelView Plus Operator Terminals User Manual* or the *VersaView CE Terminals User Manual*. These manuals are available on the RSVIEW Machine Edition™ CD.

Summary of steps

Follow these steps to:

- install hardware and software on a PanelView Plus or VersaView CE terminal.
- transfer your application to the PanelView Plus or VersaView CE terminal.

For information about installing RSView ME Station, see the *RSView Machine Edition Installation Guide*.

Installing hardware and software on a VersaView CE terminal

1. If you will be printing displays, alarms, or diagnostics messages, connect a printer to the USB or Network port.

VersaView CE terminals support printing using the Microsoft® Windows® CE PCL3 printer driver, which is already installed on the terminal. Printing is supported for most laser and ink-jet printers. If you have problems printing, check that your printer is compatible with the PCL3 printer driver.

2. If you are using an OPC® server other than RSLinx Enterprise, for example, KEPServerEnterprise™, install the OPC server software on the terminal.

For installation information, see the documentation supplied by your OPC server vendor. For information about OPC data servers, see Chapter 5.

3. If your application uses third-party ActiveX® objects, install and register the Windows CE version of the objects on the terminal.

For information about ActiveX objects, see page 15-25.

Installing printers on a PanelView Plus terminal

- If you will be printing displays, alarms, or diagnostics messages, connect a printer to the USB or Network port.

PanelView Plus terminals support printing using the Microsoft Windows CE PCL3 printer driver, which is already installed on the terminal. Printing is supported for most laser and ink-jet printers. If you have problems printing, check that your printer is compatible with the PCL3 printer driver.

The PanelView Plus terminal is a “closed box,” which means you must order any software your application will need when you order the system. All software comes pre-installed.

Transferring applications

1. On the PanelView Plus or VersaView CE terminal, start RSView ME Station.

For more information, see page 25-4.

2. In RSView ME Station, if you're transferring via a serial connection, specify and set up the serial driver. If you are using Ethernet® for the transfer, you can skip this step.

For more information, see page 25-5.

3. On the development computer, set up the RSLinx Enterprise driver to use for the transfer.

For more information, see page 25-6.

4. Move the application to the PanelView Plus or VersaView CE terminal using the Transfer Utility tool.

For more information, see page 25-7.

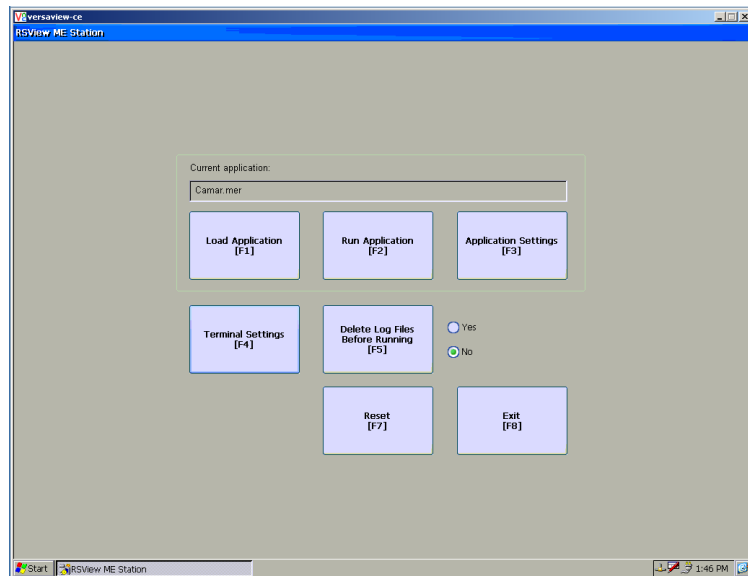
You can also transfer applications to the terminal using a Flash Card. For information about this method, see the *PanelView Plus Operator Terminals User Manual* or the *VersaView CE Terminals User Manual*.

Starting RSView ME Station

To start RSView ME Station on a VersaView CE terminal

- On the Windows Start menu, select Programs, Rockwell Software, RSView Enterprise, and then click RSView ME Station.

The RSView ME Station dialog box opens.



Starting RSView ME Station on a PanelView Plus terminal

If you are running a PanelView Plus terminal, the RSView ME Station dialog box opens automatically when the terminal starts up.

Specifying the driver to use for the transfer

You can download the runtime application file to the PanelView Plus or VersaView CE terminal using:

- Ethernet
- a direct serial connection

If you are using an Ethernet connection for the transfer, you don't need to set up a driver for the transfer, since the Ethernet driver is selected and loaded by default.

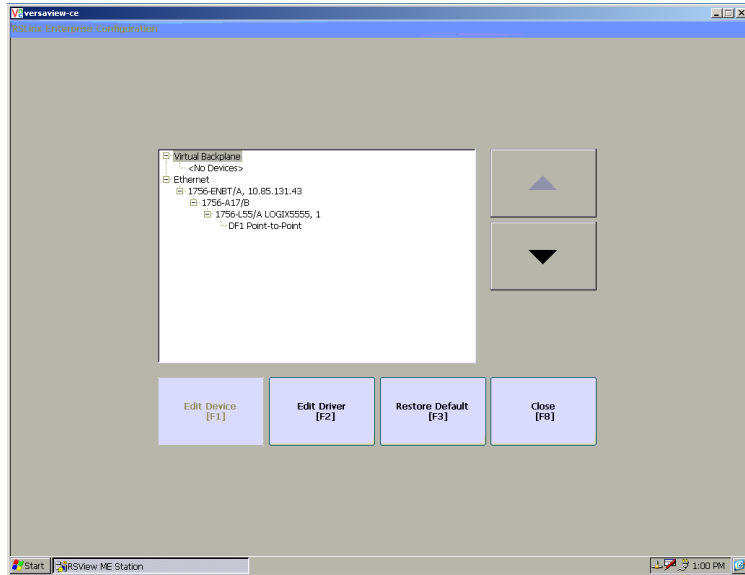
If you are using a serial connection for the transfer, specify and set up the driver.

You can also transfer applications to the terminal using a Flash Card. For information about this method, see the *PanelView Plus Operator Terminals User Manual* or the *VersaView CE Terminals User Manual*.

To specify and set up a serial driver for the transfer

1. In the RSVIEW ME Station dialog box, click Terminal Settings.
2. Double-click Networks and Communications.

3. In the Networks and Communications dialog box, double-click RSLinx Enterprise Communications.



4. Click the device you want to set up and then click Edit Device.
5. Set up the device, and then click OK.
6. Click the Driver you want to set up and then click Edit Driver.
7. Set up the driver, and then click Close.

Setting up a driver for the transfer on the development computer

On the development computer, set up the RSLinx Enterprise driver that you will use to download your application to the PanelView Plus or VersaView CE terminal.

Use one of these drivers for the transfer:

- AB_DF1-1—Use for a serial transfer.
- AB_ETH-1—Use with an Ethernet connection.

For information about setting up drivers in RSLinx Enterprise, see RSLinx Help.

Downloading applications and Windows TrueType fonts

Use the Transfer Utility to download your runtime application or Windows TrueType font files from the development computer to the PanelView Plus or VersaView CE terminal. Runtime applications have the extension .mer. Windows TrueType font files have the extension .ttf or .ttc.

For information about creating the runtime application file, see Chapter 23.

About the download

You can use a serial or Ethernet connection between the development computer and the PanelView Plus or VersaView CE terminal destination. You must use one of the drivers listed in the previous section for the download.

You can download while an application is running on the runtime computer.

You can download:

- multiple applications (one at a time) to the same runtime computer.
- a new copy of the application currently running on the runtime computer, for use the next time the application is started.

IMPORTANT

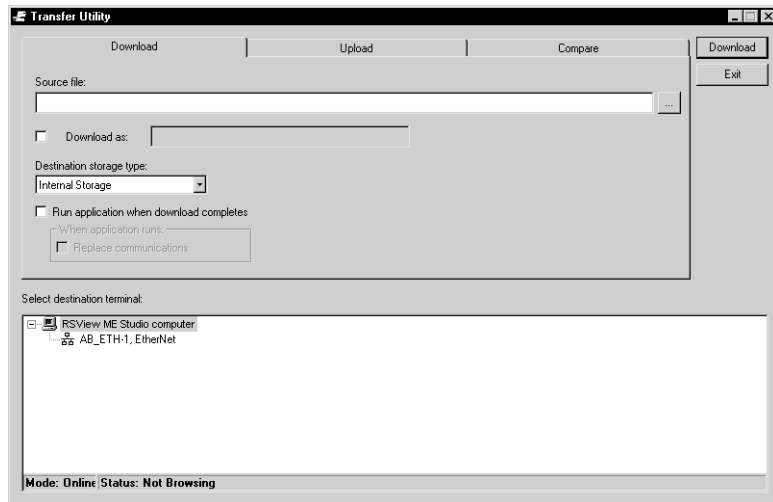
If the download process fails or is cancelled, the application file will be deleted from the destination directories. Make a copy of the application file before you begin and make sure there is enough storage space on the destination computer.

Serial downloads

To perform a serial download, make sure the terminal is connected to the development computer using the correct cable. Connect a PanelView Plus or VersaView CE terminal to the PC using the Allen-Bradley® serial cable 2711-NC13.

To start the Transfer Utility tool

- ▶ On the development computer, do one of the following:
 - In RSView Studio™, on the Tools menu, click Transfer Utility.
 - On the Windows Start menu, select Programs, Rockwell Software, RSView Enterprise, Tools, and then click ME Transfer Utility.



For details about using the utility, see the utility's Help.

Uploading applications from the PanelView Plus or VersaView CE terminal

Use the Transfer Utility to upload your runtime application from the PanelView Plus or VersaView CE terminal to the development computer. Runtime applications have the extension .mer.

IMPORTANT

If the upload process fails or is cancelled, the application file will be deleted from the destination directory. Make sure there is enough storage space on the destination computer.

About the upload

You can upload while an application is running on the runtime computer. You can upload an existing copy of the application currently running on the runtime computer.

Serial uploads

To perform a serial upload, make sure the terminal is connected to the development computer using the correct cable. Connect a PanelView Plus or VersaView CE terminal to the PC using the Allen-Bradley serial cable 2711-NC13.

For details about performing the upload, see Help for the Transfer Utility.

Comparing applications

You can also use the Transfer Utility tool to compare an application on the development computer with an application on the terminal. The comparison tool can tell you whether the files are identical or different.

For details about comparing applications, see Help for the Transfer Utility.

Using your application

This chapter describes:

- logging in and logging out.
- entering numeric and string values.
- changing tag values.
- viewing tag data.
- viewing alarms and messages.
- viewing information about runtime communication errors.

For information about navigating between displays, see Chapter 22.

For information about navigating between and using the graphic objects in a display, see page 16-4.

Logging in to a Windows domain

To log data to a network location, the runtime computer must be logged in to the same domain as the computer on the network. To do this, the user must have access rights for the domain.

To log in to a domain from Windows 2000 or Windows XP

- When Windows® starts, type a user name, password, and domain name.

The PanelView Plus™ or VersaView® CE terminals cannot be part of a domain. However, you can verify that the user who is logged in to the PanelView Plus or VersaView CE terminal is on a user list that is part of a domain.

On the PanelView Plus or VersaView CE terminal, logging in hard codes a domain member's credentials. This gives the terminal access to permissible network resources such as network folders.

To authenticate a user on a PanelView Plus or VersaView CE terminal

1. In the RSVIEW® ME Station™ dialog box, click Terminal Settings.
2. Double-click Networks and Communications.
3. Double-click Network Connections.
4. Double-click Network Identification.
5. Type a user name, password, and domain name, and then click OK.

Logging in to the application

To log in, the user must have an account in the User Accounts editor. If security is set up, the user must also have a password.

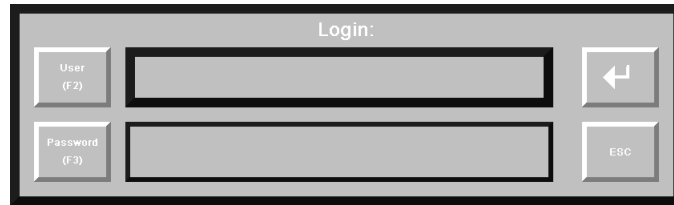
For information about setting up user accounts and passwords, see Chapter 13.

You cannot change passwords in RSVIEW ME Station. To change a password for an RSVIEW user, use the User Accounts editor in RSVIEW Studio™. To change a password for a Windows user, use the Windows Control Panel.

To log in

1. Press the login button.

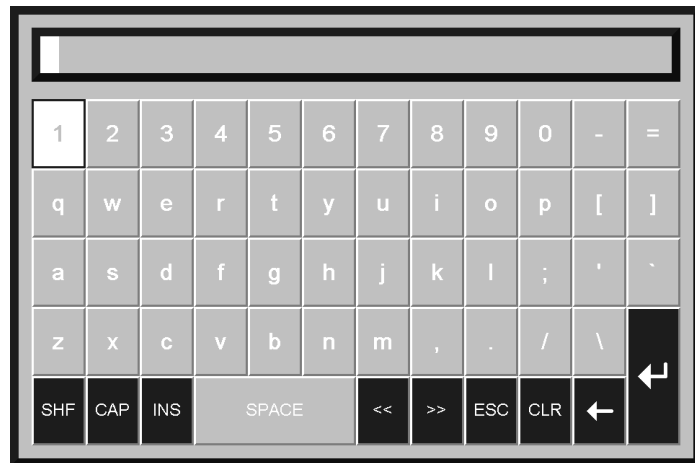
The Login window opens.



The Login window is a rectangular box with a title bar. Inside, the word "Login:" is centered at the top. Below it, there are two input fields. To the left of the first input field is a button labeled "User (F2)". To the left of the second input field is a button labeled "Password (F3)". To the right of the first input field is a button with a left arrow icon. To the right of the second input field is a button labeled "ESC".

2. Press the User button, or press F2 on an external keyboard.

The string pop-up keyboard opens. For details about using the keyboard, see page 26-11.



3. Type your user name in the string pop-up keyboard, or on an external keyboard, and then press Enter.

The name is not case sensitive.

4. To enter a password, press the Password button, or press F3 on an external keyboard.

5. Type the password in the string pop-up keyboard, or on an external keyboard, and then press Enter.

The password is not case sensitive.

6. To close the Login window and complete the login, press Enter.

What happens when a user logs in

When a user logs in:

- If the previous user did not log out, the previous user is logged out now. If a logout macro is set up for the previous user, the logout macro runs, assigning values to tags.
- The new user is logged in.
- If a login macro is set up for the new user, the macro runs, assigning values to tags.

Problems with logging in

Login is unsuccessful under these circumstances:

- If the graphic display changes remotely before the user has finished logging in, the login is cancelled.
- If the user name or password doesn't match the settings in the User Accounts editor, the user is not logged in.
- If the screen resolution is smaller than 280 pixels wide by 84 pixels high, the Login window cannot open.
- If the screen resolution is smaller than 236 pixels wide by 208 pixels high, the string pop-up keyboard cannot open.

Logging out

When the current user logs out, the DEFAULT user is logged in.

To log out

- Press the logout button.

Entering numeric values

The operator can enter or ramp numeric values at run time using the numeric input enable button or the numeric input cursor point.

When the operator presses the button or activates the cursor point, the numeric pop-up keypad or scratchpad opens. If you set up the button or cursor point to ramp, pressing the object gives it focus, but does not open a pop-up window.

To use the numeric pop-up scratchpad, a keyboard must be attached to the runtime computer, or the terminal must be a keypad terminal.

For information about setting up the numeric input enable button and numeric input cursor point, see Help.

You can also use a ramp button to ramp numeric values. For details, see Help for the button.

Activating the cursor point

When the numeric input cursor point has focus, the operator can activate the cursor point by pressing any of these keys on the keyboard or keypad:

- numbers from 0 to 9
- the minus sign (-) or decimal point (.)
- the Enter key, or an enter button

- the Backspace key, or a backspace button

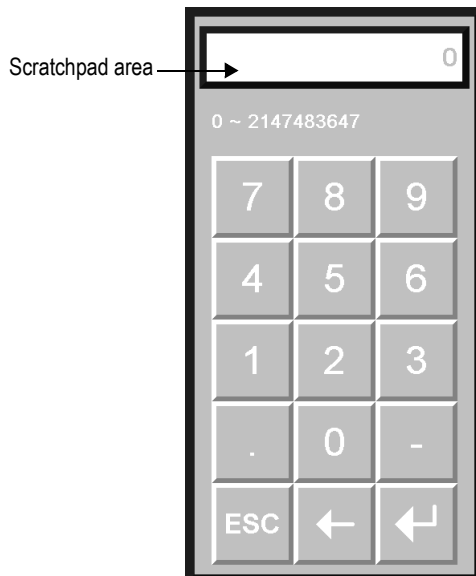
When the operator activates the numeric input cursor point, the numeric pop-up keypad or scratchpad opens.

Ramping numeric values

If you set up the numeric input enable button or numeric input cursor point to ramp values, the operator presses the object to give it focus. When the object has focus, the operator can press a move up or move down button, or the Up Arrow or Down Arrow key on a keyboard or keypad, to ramp the value at the Value control.

Using the numeric pop-up keypad

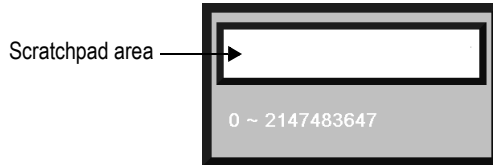
The keypad can accept up to 17 digits, including the decimal point, leading zero, and minus sign.



When the keypad is open, no other objects in the graphic display can accept input.

Using the numeric pop-up scratchpad

The scratchpad can accept up to 17 digits, including the decimal point, leading zero, and minus sign.



The scratchpad does not contain any buttons. If the runtime computer does not have an external keyboard or keypad attached, the operator will be unable to enter values or close the scratchpad.

When the scratchpad is open, no other objects in the graphic display can accept input.

Using buttons and keys with the numeric pop-up windows

You can use the following buttons and their keyboard or keypad equivalents with the numeric pop-up keypad. For the pop-up scratchpad, you can use the keys on an external keyboard or keypad only.

Use this button or key	To do this
Decimal (.)	Type a decimal point. This button is visible only if the decimal point type for the numeric input enable button or numeric input cursor point is Implicit.
Minus (-)	Toggle the value between positive and negative.
ESC	Close the pop-up window without writing the value to the object's Value control.

Use this button or key	To do this
Backspace	Delete the right-most digit in the scratchpad. If no digits are left, the minus sign, if any, is removed.
Enter	<p>Close the pop-up window and write the value to the object's Value control.</p> <p>If the Optional Expression control is assigned to the button or cursor point, RSView calculates the value of the expression and writes the result to the Value control. For details, see page 19-21.</p> <p>If Enter key handshaking is set up for the button or cursor point, the object's Enter control is set to 1 and the handshaking timer begins timing. For details, see page 16-16.</p>
Delete	<p>Clear the scratchpad.</p> <p>There is no Delete button in the pop-up keypad, but you can use the Delete key on an external keyboard or keypad.</p>

How values are ramped

If the button or cursor point is being used to ramp a value at the Value control:

- When the tag assigned to the Value control is an integer tag, but the ramp value is a floating point value, the ramp value is added to (or subtracted from) the Value control value, and then the result is rounded and written to the Value control.

For information about how values are rounded, see page 7-2.

- If the operator presses and holds down the move up or move down button, or the Up Arrow or Down Arrow key on the keyboard or keypad, the button or key goes into auto repeat mode. The ramp

value is added to (or subtracted from) the last value sent to the Value control.

For more information about auto repeat mode, see page 16-15.

- If ramping the button or cursor point would cause the value at the Value control to exceed the maximum value, or be less than the minimum value, a message is written to the Diagnostics List and the value at the Value control is not changed.

How values are calculated

When the pop-up keypad or scratchpad has focus and the operator presses the Enter button, the value that is sent to the Value control depends on how the button or cursor point is set up.

- The value the operator enters is compared to the minimum and maximum range for the object. If the value is within the range, the value is written to the Value control.

If the Optional Expression control is assigned, and the original value is within the range but the expression result is a value outside the range, the value is still written to the Value control.

- If the decimal point type is Fixed Position, with a “Digits after decimal” value that is greater than 0, the decimal point is stripped from the value before comparing it to the minimum and maximum range.

For example, if the entered value is 9.25, the stripped value is 925.

If the stripped value is within the range, the stripped value is sent to the Value control (or substituted for the ? in the Optional Expression).

- If the decimal point type is Implicit, and the tag assigned to the Value control uses an integer data type, floating-point values are rounded.

If the Optional Expression control is assigned, the floating-point value is substituted for the ? in the expression, and the expression result is rounded.

For information about how values are rounded, see page 7-2.

Problems with the numeric pop-up windows

Problems with the numeric pop-up keypad and scratchpad occur under these circumstances:

- If the graphic display changes remotely before the user has pressed the Enter button, the pop-up window closes without writing out a value.
- If the screen resolution is smaller than 124 pixels wide by 240 pixels high, the pop-up keypad cannot open.
- If the screen resolution is smaller than 124 pixels wide by 68 pixels high, the pop-up scratchpad cannot open.
- If the value is too large for the data type of the tag assigned to the Value control, the pop-up window remains open and the value is not written to the control. The scratchpad area changes to red to alert the operator of the error.
- If the value is outside the minimum and maximum range specified for the object, the pop-up window remains open and the value is not written to the control. The scratchpad area changes to red to alert the operator of the error.
- If Enter key handshaking is still in effect, the pop-up window closes but the value is not written to the control.

Entering string values

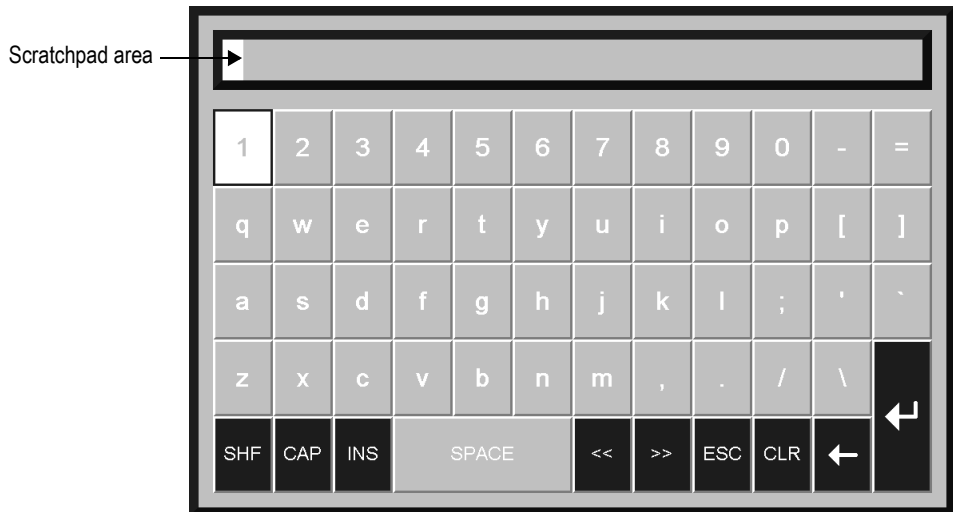
The operator can enter string values at run time using the string input enable button.

When the operator presses the button, the string pop-up keyboard or scratchpad opens, depending on how you set up the button. To use the scratchpad, a keyboard must be attached to the runtime computer.

For information about setting up the string input enable button, see Help.

Using the string pop-up keyboard

The string pop-up keyboard opens when the operator presses a string input enable button or the User or Password button in the Login dialog box.



If the operator opens the pop-up keyboard by pressing a string input enable button, the number of characters the keyboard accepts depends on how you set up the button.

When the keyboard is open, no other objects in the graphic display can accept input.

Using the string pop-up scratchpad

The number of characters the scratchpad accepts depends on how you set up the string input enable button.



The scratchpad does not contain any buttons. If the runtime computer does not have an external keyboard attached, the operator will be unable to enter characters or close the scratchpad (unless the runtime terminal is a keypad terminal). If the runtime terminal is a keypad terminal, the operator can enter numbers (not letters) in the scratchpad, and close the scratchpad.

When the scratchpad is open, no other objects in the graphic display can accept input.

Using buttons and keys with the string pop-up windows

You can use the following buttons and their keyboard equivalents with the string pop-up keyboard. For the pop-up scratchpad, you can use the keys on an external keyboard only.

Use this pop-up keyboard button	Or this keyboard equivalent	To do this
SHF	none	Capitalize a single letter, or type a shifted character such as #.
CAP	none	Capitalize multiple letters.

Use this pop-up keyboard button	Or this keyboard equivalent	To do this
INS	Insert	Toggle between insert and overstrike modes.
SPACE	Spacebar	Insert a space.
<<	Left Arrow	Move the cursor to the left.
>>	Right Arrow	Move the cursor to the right.
ESC	Esc	Close the pop-up window without writing the string to the Login dialog box or the string input enable button's Value control.
CLR	Delete	Clear the scratchpad.
Backspace	Backspace	Delete the character in front of the cursor.
Enter	Enter	<p>Close the pop-up window and write the string to the Login dialog box or the string input enable button's Value control.</p> <p>If Enter key handshaking is set up for the string input enable button, the button's Enter control is set to 1 and the handshaking timer begins timing. For details, see page 16-16.</p>

What is written to the Value control

When the operator presses the Enter button in the pop-up keyboard or scratchpad, the string that is sent to the Value control depends on how the string input enable button is set up.

- If a fill character is set up for the button, and the operator enters fewer than the maximum number of input characters, the fill characters are placed after the string the operator enters.
 - Spaces have a hex value of 20.
 - Zeroes have a hex value of 30.
 - FF characters have a hex value of FF.
 - Null characters have a hex value of 0. The null character indicates the end of string input. It does not add to the actual string length.
- If the number of input characters is fewer than the number of characters in the length of the string tag assigned to the Value control, the remaining spaces are padded with the null character.

When the string is written to the Value control, the first character is placed in the high order byte of the first word at the tag address, the second character is placed in the low order byte of the first word, and so on.

Problems with the string pop-up windows

Problems with the string pop-up keyboard and scratchpad occur under these circumstances:

- If the graphic display changes remotely before the user has pressed the Enter button, the pop-up window closes without writing out a string.
- If the screen resolution is smaller than 236 pixels wide by 208 pixels high, the pop-up keyboard cannot open.

- If the screen resolution is smaller than 236 pixels wide by 44 pixels high, the pop-up scratchpad cannot open.
- If the string pop-up window is set up to accept more characters than the Value control tag length, the pop-up window remains open and the string is not written to the control. The scratchpad area changes to red to alert the operator of the error.
- If Enter key handshaking is still in effect, the pop-up window closes but the value is not written to the control.

Changing tag values

This section gives an overview of the graphic objects you can use to change tag values. For information about setting up the objects, see Chapter 16 and Help.

The operator uses these objects to start and stop plant operations, and to control machines and processes. Choose the objects that best suit your process. Set up the data source to carry out the desired actions in response to the changes in tag values.

The operator can use function keys with all of these objects except control list selectors, third-party ActiveX® objects, and drawing objects with slider animation. The last three types of objects can be pressed using a mouse or touch screen.

For information about assigning function keys to graphic objects, see page 16-6.

IMPORTANT

Do not use push buttons for emergency stops.
Emergency stop buttons must be hard-wired.

You can also use macros to assign values to tags. For more information, see Chapter 21.

The operator can change tag values at run time using these graphic objects:

Use this graphic object	To do this
Momentary push button	Start a process or action by sending one value to the tag when pressed, and another value when released.
Maintained push button	<p>Toggle between two values by sending one value to the tag when pressed, and a second value the next time the button is pressed and released.</p> <p>This button is useful for changing a setting within a machine or process, but not for starting the machine or process.</p>
Latched push button	Start a machine or process and remain set (latched) until the process is completed, by sending a value to the tag when pressed, and retaining this value until reset (unlatched) by the Handshake control.
Multistate push button	<p>Cycle through a series of values. Each time the operator presses the button, the value for the next state is sent to the tag. When the button is in its last state, pressing it causes the button to change to its first state and write out the first state value.</p> <p>This button is useful when you want the operator to see and select multiple options in sequence, using a single button. The button displays the current state of a process or operation by showing a different color, caption, or image to reflect the different states.</p>

Use this graphic object	To do this
Interlocked push button	<p>Use a group of buttons to send values to the same tag. When the operator presses a button in the group, the button's value is sent to the tag, and the button remains highlighted as long as the tag value is the same as the button's value. Pressing a new button in the group releases the other button and sends a new value to the tag.</p> <p>You can also use a single interlocked push button to send a value to a tag.</p>
Ramp button	<p>Increase or decrease the value of a tag by a specified integer or floating-point value.</p> <p>Use two ramp buttons together to create a raise/lower control.</p>
Numeric input enable button	Enter a numeric value and write the value to a tag. You can also use this object to ramp values.
Numeric input cursor point	Enter a numeric value and write the value to a tag. You can also use this object to ramp values.
String input enable button	Enter a string value and write the value to a tag.
Drawing object with horizontal or vertical slider animation	<p>Control the value of a tag by dragging the slider object with a mouse. The pixel position of the slider is translated into a value that is written to the tag.</p> <p>If the value of the tag is changed externally, the position of the slider changes to reflect this.</p>

Use this graphic object	To do this
Control list selector	<p>Select from a list of states for a process or operation. The list is highlighted to show the current state, and the operator can scroll through the list to select a different state. The value assigned to the selected state is written to the tag.</p> <p>If the value of the tag is changed externally, the position of the highlight changes to reflect this.</p>
ActiveX object	A third-party object, connected to an analog, digital, or string tag, including both HMI and data server tags. When the object's property value changes, the new value is written to the associated tag.

You can attach visibility animation to these graphic objects, to display or hide the objects based on changes in tag or expression values. For information about visibility animation, see page 17-11.

For information about creating graphic objects, see Chapter 15. For information about setting up graphic objects, see Chapter 16 and Help.

Viewing tag data

This section gives an overview of the graphic objects you can use to display tag data. For information about setting up the objects, see Chapter 16 and Help.

The operator can view tag data at run time using these graphic objects:

Use this graphic object	To display this
Numeric display	Numeric tag values. For example, display the current temperature of an oven.
Numeric input cursor point	Numeric tag values. For example, display the current temperature of an oven.

Use this graphic object	To display this
String display	String tag values. For example, set up the data source to generate strings that report on the state of a process or operation, or that provide the operator with instructions about what to do next.
Bar graph	Numeric values in bar graph format. The bar graph increases or decreases in size to show the changing value.
Gauge	Numeric values in dial format. The gauge's needle moves around the dial to show the changing value.
Multistate indicator	The state of a process, on a panel that changes its color, image, or caption to indicate the current state. Each state is set up to correspond to a numeric tag value or least significant bit.
Symbol	<p>The state of a process, using a monochrome image that changes color to indicate the current state. Each state is set up to correspond to a numeric tag value or least significant bit.</p> <p>This object is useful for showing the state of a process or operation at a glance.</p>
List indicator	<p>The state of a process, using a list of possible states with the current state highlighted. Each state is represented by a caption in the list, and is set up to correspond to a numeric tag value or least significant bit.</p> <p>This indicator is useful if you want to view the current state but also want to see the other possible states. For sequential processes, the list can alert the operator about what happens next in the process.</p>

Use this graphic object	To display this
Trend	Historical or current numeric tag values, plotted against time or displayed in an XY plot where one or more tags' values are plotted against another tag's values to show the relationship between them.
Drawing object with rotation, width, height, fill, color, or horizontal or vertical position animation	<p>Display the value of a tag using a pictorial representation that shows the current value in relation to a range of possible values. For example, use rotation animation to show the tag value as a needle's position on a dial.</p> <p>For color animation, assign different colors to represent different values.</p>
ActiveX object	A third-party object, connected to an analog, digital, or string tag, including both HMI and data server tags. The data displayed depends on the object.

Any of these objects can be set up to manipulate tag values using expressions, and display the expression result rather than the original tag value. For information about expressions, see Chapter 19.

You can also attach visibility animation to these graphic objects, to display or hide the objects based on changes in tag or expression values. For information about visibility animation, see page 17-11.

For information about creating graphic objects, see Chapter 15. For information about setting up graphic objects, see Chapter 16 or Help.

Displaying the date and time

To display the current date and time, create a time and date display. This object uses the operating system's date and time and therefore does not use tags or expressions.

Viewing alarms and messages

The operator can view alarms and other messages at run time using these graphic objects and graphic displays:

This information	Appears in this object	In this default graphic display	For details, see
Alarm messages	Alarm banner	[ALARM].	page 9-30
Alarm messages	Alarm list	No default, although this object appears in the [ALARM MULTI-LINE] and [HISTORY] graphic libraries.	page 9-29, page 9-31, and page 9-33
Alarm messages	Alarm status list	No default, although this object appears in the [STATUS] graphic library.	page 9-32
System activity	Diagnostics list	[DIAGNOSTICS]	page 10-16
Information messages	Information message display	[INFORMATION]	page 12-10
Local messages	Local message display	No default.	page 14-37

The default alarm and information displays open automatically when the assigned tags match messages' trigger values. The default diagnostics display opens automatically when system activity occurs. If desired, you can set up your own graphic displays to open automatically, instead of the default displays. You can also set up any of the displays to open when an operator presses a goto display button or selects a display in the display list selector.

The operator can acknowledge alarm and information messages. The operator can clear alarm and diagnostics messages. The operator can sort alarms and reset their status.

Viewing information about runtime communication errors

To display communication errors in the diagnostics list object, set up message routing so that messages are sent to the RSVIEW Diagnostics List.

For information about setting up diagnostics message routing, see Chapter 10.

A

Converting PanelBuilder 1400e applications

This appendix describes:

- terms that are different in PanelBuilder™ 1400e and RSVIEW®.
- steps for converting PanelBuilder 1400e applications.
- names of equivalent graphic objects in the two products.
- PanelBuilder 1400e graphic objects that are not supported in RSVIEW.
- PanelBuilder 1400e settings and controls that are not supported in RSVIEW.
- how communications are converted and which PanelBuilder 1400e communication protocols are not supported in RSVIEW.
- converting PanelBuilder 1400e Remote I/O communications.
- PanelBuilder 1400e graphic object features that are not supported in RSVIEW, with information about how to achieve the same result when possible.
- converting PanelBuilder 1400e expressions.

PanelBuilder 1400e applications are applications you create using PanelBuilder 1400e Configuration Software for Windows®. For information about converting applications from PanelBuilder or PanelBuilder32, see Appendix B.

Terminology

This section describes terms that are different in PanelBuilder 1400e and RSView.

PanelBuilder 1400e term	RSView term
screen	display, graphic display
Optional Keypad Write Expression	Optional Expression
programmable controller	data source

In RSView, the data source can be memory or a device such as a programmable controller or an OPC® server. RSView writes values to and reads values from the data source. The data source is configured to exchange information (in the form of numeric or string values) between RSView and the machine that your application is controlling. The general term data source is used unless specifically discussing a programmable controller.

Summary of steps

Follow these steps to convert PanelBuilder 1400e applications:

1. Convert the application file, as described in the next section.
2. Specify additional project settings, as described on page 4-11.

For example, if you want the application to have a border around its graphic displays, or to use a title bar, you can specify these options in the Project Settings editor.

3. If you select Convert to new window size, check the position of the graphic objects in each display.
4. Set up communications and edit tags that don't convert directly.

For more information, see page A-11.

5. Set up graphic object features that don't convert directly.

For more information, see page A-12.

6. Check each expression you used in PanelBuilder 1400e.

For more information, see page A-17.

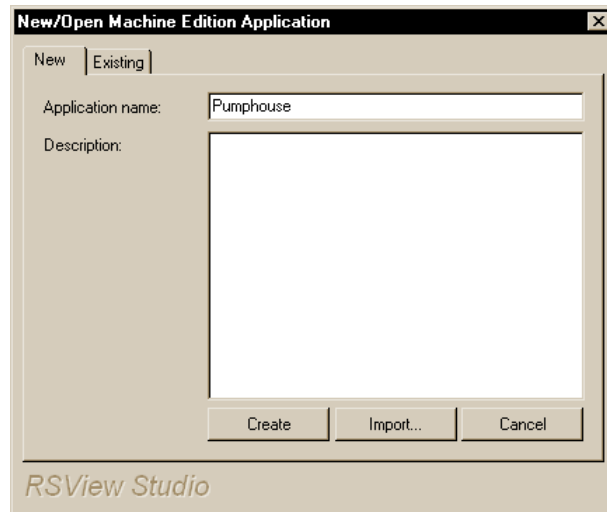
Converting PanelBuilder 1400e application files

Follow these steps to convert a PanelBuilder 1400e application file, with the extension .pvc, to an RSVView application file, with the extension .med. The original PanelBuilder 1400e application file is not modified by the conversion.

You can convert the PanelBuilder 1400e application when you open RSVView Studio™, or once RSVView Studio is already open.

To convert a PanelBuilder 1400e application when you open RSVView Studio

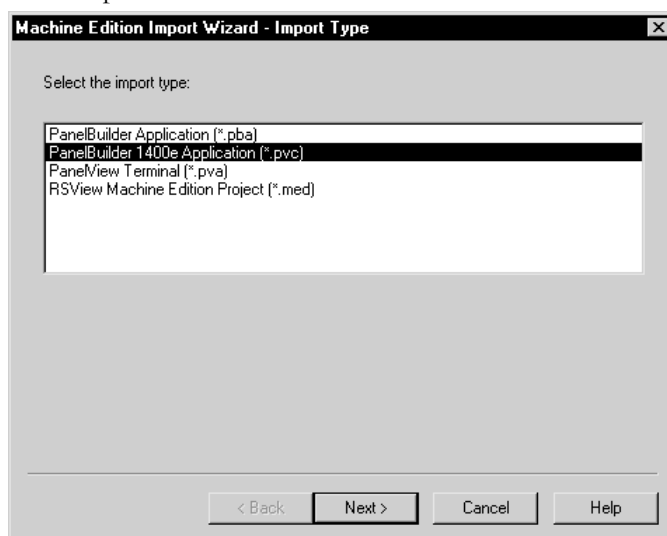
1. Open RSVView Studio.



2. In the New tab, in the Application name box, type a name for your converted application, up to 32 characters long.
3. If desired, type a description of the application. If the PanelBuilder 1400e application contains an Application File Comment, the Application File Comment will overwrite the description you type here.

You can add or change the description later, in the Application Properties dialog box, as described in Help.

4. Click Import.



5. Follow the steps in the Machine Edition Import Wizard.

For details about the options in the Machine Edition Import Wizard, see Help.

When you complete the steps of the wizard, RSVIEW Studio converts the PanelBuilder 1400e application, creates the converted application's folders and files, and then displays the converted application in the Application Explorer window in RSVIEW Studio.

If there are any messages about conversion, they are displayed automatically in the Project Status dialog box.

The converted application is created in the ME\HMI projects directory, in a folder with the same name as the application name you specified in step 2.

This is the path to the ME\HMI projects directory:

\Documents and Settings\All Users\Documents\RSView
Enterprise\ME\HMI projects (Windows 2000)

or

\Documents and Settings\All Users\Shared Documents\RSView
Enterprise\ME\HMI projects (Windows XP)

Conversion messages are saved in a file called Convert.log, in the HMI projects directory.



New Application

To convert a PanelBuilder 1400e application when RSView Studio is already open

1. On the File menu, click New Application, or click the New Application tool.

If an application is already open, RSView Studio asks you whether to close the application that is currently open. Click Yes.

2. Follow steps 2 through 5 in the previous procedure.

Equivalent graphic objects

This section describes graphic objects that are equivalent in PanelBuilder 1400e and RSView, but have different names in the two products.

This PanelBuilder 1400e object	Is converted to this RSView object	Notes
Increment Value Button	Ramp button	During conversion the button is set up to increment.
Decrement Value Button	Ramp button	During conversion the button is set up to decrement.
Increment Value Button with Display	Ramp button and numeric display	The Increment Value Button with Display is divided into two separate RSView objects.
Decrement Value Button with Display	Ramp button and numeric display	The Decrement Value Button with Display is divided into two separate RSView objects.
ASCII Input (small and large)	String input enable button	
Numeric Entry Keypad (small and large)	Numeric input enable button	
Screen List Selector's list	Display list selector	The PanelBuilder 1400e Screen List Selector is divided into four separate RSView graphic objects.
Screen List Selector's Enter Key	Enter button	
Screen List Selector's Down Cursor	Move down button	
Screen List Selector's Up Cursor	Move up button	
Control List Selector's list	Control list selector	The PanelBuilder 1400e Control List Selector is divided into four separate RSView graphic objects.
Control List Selector's Enter Key	Enter button	

This PanelBuilder 1400e object	Is converted to this RSVIEW object	Notes
Control List Selector's Down Cursor	Move down button	
Control List Selector's Up Cursor	Move up button	
Screen Select Keypad (small and large)	Display list selector	Specify the graphic displays that the display list selector can open.
Screen Keypad Enable Button	Display list selector	Specify the graphic displays that the display list selector can open.
Goto Screen Button	Goto display button	
Return to Previous Screen Button	Return to display button	
ASCII Display	String display	
Numeric Keypad Enable Button	Numeric input enable button	
Normally Open Momentary Push Button	Momentary push button	During conversion the button is set up to be normally open.
Normally Closed Momentary Push Button	Momentary push button	During conversion the button is set up to be normally closed.
Screen Print Button	Display print button	
Alarm History Sort By Time/Sort By Value Button	Sort alarms button	
Alarm Status Reset Qty/Time Button	Reset alarm status button	
Alarm Panel	Alarm banner	
Single Line Alarm Window	Alarm banner	
Alarm Status Screen	Alarm status list	
Clear All Button	Clear alarm history button	

This PanelBuilder 1400e object	Is converted to this RSVView object	Notes
Print Button (in Alarm History screen)	Print alarm history button	
Print Button (in Alarm Status screen)	Print alarm status button	
Alarm Status Button/ Alarm History Button	Goto display button	
Exit Button	Close display button	
Alarm History List	Alarm list	
Display Mode Button	Alarm status mode button	
Time Display	Time and date display	During conversion the display is set up to show the time only. The PanelBuilder 1400e time format is not converted. For details about the RSVView time format, see page 16-20.
Date Display	Time and date display	During conversion the display is set up to show the date only. The PanelBuilder 1400e date format is not converted. For details about the RSVView date format, see page 16-20.
Arc (with solid fill style)	Arc (with solid back style) and line	The line graphic object is added because the solid RSVView arc shape does not have a line between the two points of the arc.



RSView arc



RSView arc with line

Unsupported graphic objects

These PanelBuilder 1400e objects are not supported in RSView:

- Scrolling List (includes Cursor List, Multistate Indicator Object List, Local Message Object List, Numeric Data Display Object List)
- Set Bit Cursor Point

Unsupported settings and controls

This section describes PanelBuilder 1400e settings and controls that are not used in RSView.

Controls for transferring runtime application files

PanelBuilder 1400e uses these optional controls for transferring files to the runtime terminal:

- Transfer Inhibit control
- Transfer Request control
- Transfer Status control

These controls are not necessary in RSView because the ME Transfer Utility allows you to transfer the runtime project file while running a project on the runtime terminal.

Settings and controls for alarms

RSView does not use these PanelBuilder 1400e features and settings to manage alarms:

- alarm relays
- bit alarm acknowledgement

- Remote Alarm Operation Hold Time. The PanelBuilder 1400e Remote Alarm Ack Control Hold Time will be used for all alarm hold times. You can change the hold time in the RSVIEW Alarm Setup editor, in the Advanced tab.
- Remote Alarm Control Delay Time. In RSVIEW, if an Ack control is assigned, when an alarm is acknowledged the Ack control is set immediately, without waiting for a delay time.

RSVIEW does not use these PanelBuilder 1400e controls to manage alarms:

- PLC Controlled Relay control
- PLC Controlled Audio control
- Acknowledge to PLC control (if the Alarm Acknowledge to PLC option is set to Bit)

Invalid characters in screen names

Characters in PanelBuilder 1400e screen names that are not supported in RSVIEW are replaced with the underscore character.

Screen security settings

PanelBuilder 1400e screen security settings are not converted, because RSVIEW uses a different method to assign security to graphic displays. For information about setting up security in RSVIEW, see Chapter 13.

Block tags

Block tags are not supported in RSVIEW. Block tags that are monitored for alarms in your PanelBuilder 1400e application are converted to bit arrays. For information about monitoring bit arrays for alarm conditions, see Chapter 9.

Converting non-RIO communications

This section describes how communications that do not use Remote I/O (RIO) are converted. For information about converting RIO communications, see page A-12.

RSView does not use nodes for communications. Nodes are converted to RSLinx topics. Topics are then converted into device shortcuts, to run with RSLinx Enterprise. You must have both RSLinx® for RSView and RSLinx Enterprise installed to make this two-step conversion.

Tags are converted to HMI tags within the RSLinx topics. The Unsolicited_Msgs node is not converted.

If you import an application multiple times, delete the device shortcuts in RSLinx Enterprise before re-importing. Otherwise, multiple unused device shortcuts will be created in RSLinx Enterprise.

For more information about setting up communications, see Chapter 5.

Unsupported communication protocols

These communication protocols are not supported in RSView:

- ControlNet™ scheduled communications
- Modbus communications

Tags that use these protocols are converted to HMI memory tags. Once you have set up communications for your converted application, change the memory tags to device tags that point to the correct addresses. For information about editing HMI tags, see Chapter 7.

Unsupported tag data types

These tag data types are not supported in RSVIEW:

- Binary (used with Remote I/O communications)
- Bit Position
- 1-BCD, 2-BCD, 5-BCD, 6-BCD, 7-BCD, 8-BCD
- BIN3, BIN4, BIN6, BIN8 (used with Modbus communications)

Tags that use these data types are converted to analog HMI tags with the Default data type. The Default data type uses floating point values.

Unsupported initial values

Device tags in RSVIEW do not use initial values. Memory tags are converted with their initial values.

Converting RIO communications

Remote I/O (RIO) communications are not supported for applications that use the Windows 2000 or Windows XP operating systems. You can use RIO communications on the PanelView Plus™ and VersaView® CE runtime platforms.

To convert an RIO application from PanelBuilder 1400e to RSVIEW

1. Convert the application, as described on page A-3.
2. Open the RSLinx Enterprise data server, and then double-click Communication Setup.

3. In the Communication Setup editor, add an RIO driver.
 - For PanelView Plus 400 and 600 terminals, use the 2711P-RN1 driver.
 - For all other PanelView Plus or VersaView CE terminals, use the 2711P-RN6 driver.

For information about adding drivers in RSLinx, see RSLinx Help.

4. Expand the RIO tree, right-click RIO Data, and then click Configure RIO.
5. In the RIO Configuration dialog box, right-click RIO, and then click Import.
6. Browse to the location of the RIO configuration file.

The file is saved in the root of the application's directory.

7. In the Communications Setup editor, create a device shortcut named "PVRIO" that points to the RIO data device.

For information about creating a device shortcut, see RSLinx Help.

8. Correct any invalid RIO configurations. Invalid RIO configurations are highlighted with red "x" icons.
9. Save the converted RIO application.

Unsupported PanelBuilder 1400e RIO tags

A PanelBuilder 1400e RIO tag will be converted to an HMI memory tag and an error will be logged to the conversion log file if the RIO tag:

- has a blank address.
- has a data type of 1-BCD, 2-BCD, 3-BCD, 5-BCD, 6-BCD, or 7-BCD.
- has a data type of Bit Position and its address does not reference a single bit.

- has a data type of Binary and its address does not reference a single bit, a single word, or a length or range of 8 or 16 bits.
- for block transfer tags and tags using SLC™ optional addressing, has a data type of Byte and its address does not have a bit offset of 0 or 8.

for other tag types, has a data type of Byte and its address does not have a bit offset of 0 or 10.

- has a data type of Binary and its address has a length or range of 8 bits, but its address does not have a bit offset of 0 or 8 (for block transfer addresses and SLC I/O addresses).
- has a data type of Binary and its address has a length or range of 8 bits, but its address does not have a bit offset of 0 or 10 (for non-SLC I/O addresses).
- has a data type of Binary and its address has a length or range of 16 bits, but its address does not have a bit offset of 0.
- has a data type of Default, Unsigned Integer, Signed Integer, Long Integer, Float, 4 Digit BCD, or 8 Digit BCD and its address has a bit offset assigned that is not 0.
- has a tag type of Block.
- has an invalid PanelBuilder 1400e address.

Unsupported graphic object features

This section describes features of PanelBuilder 1400e graphic objects that are not supported in RSVIEW. The Notes column provides additional information and describes methods for achieving the same result when possible.

Graphic object	Unsupported feature in RSVIEW	Notes
Image, text, arc, ellipse, line, panel, rectangle, wedge	Blinking wallpaper objects	<p>If you want an object to blink at run time, unlock the wallpaper.</p> <p>In RSVIEW, all of the listed objects except images and panels use color animation to blink. For details, see page 17-12.</p> <p>Panels use the Blink property to blink.</p> <p>Color images do not blink. Monochrome images use the Blink property to blink.</p>
Numeric Display	Polarity	If a PanelBuilder 1400e application was set up with the Polarity control requiring a negative number to display the minus sign, the numeric display will not work properly after the application is converted to RSVIEW Machine Edition.
Numeric Input Cursor Point, Numeric Data Display	Fixed Position and PLC Controlled decimal display options	Use an expression to achieve the same result. Assign the expression to the object's Value control. For information about expressions, see Chapter 19.
Numeric Input Cursor Point, Numeric Keypad Enable Button, Numeric Keypad	PLC Controlled and Decimal Key Controlled input options	Objects are converted with the Decimal Point property set to Implicit.
Numeric Input Cursor Point	Retain Cursor on Cancel	The numeric input cursor point retains focus when the operator cancels entering a numeric value.

Graphic object	Unsupported feature in RSVIEW	Notes
Maintained Push Button, Multistate Push Button, Control List Selector	Initial state values	If you want to set these objects' states on application startup, create a macro to set the appropriate tag values for the objects' controls. For information about macros, see Chapter 21. Assign the macro in the Startup editor. For details, see Help.
Trend	Blinking pens Date labels on the X-Axis Background screen plotting	<p>The date is displayed in the title.</p> <p>You can plot tag values in the background by assigning the tags to a data log model. Tags set up for background screen plotting are automatically assigned to a data log model on conversion. However, data log models do not plot expression values. Therefore, expressions set up for background screen plotting are not converted.</p> <p>For information about data logging, see Chapter 11.</p>
All objects	PanelBuilder 1400e object name	Object names are replaced with the RSVIEW default object names. The PanelBuilder 1400e object name is used for the object's description. You can view and edit the name and description in the Property Panel. For details, see Help.
All objects	Caption and image placement	RSVIEW supports one, three, or nine positions for captions and images, depending on the type of object. On conversion, captions and images are positioned using the closest match. Therefore some captions might overlap images, some captions might be truncated, and some images might be clipped to fit the object.

Graphic object	Unsupported feature in RSVIEW	Notes
All objects	Multiple image labels	RSView supports one image label per object or state. If a PanelBuilder 1400e object is set up to use multiple image labels, only the top left image is converted.

Converting expressions

Some PanelBuilder 1400e expression syntax is not supported in RSVIEW. Expressions are converted without modification, and then turned off by placing warning text at the beginning of the first line of the expression. In addition, exclamation marks (!) are placed at the beginning of each subsequent line of the expression. To turn on the expression, you must remove the warning text and exclamation marks, and revise the syntax if necessary.

The maximum expression length in RSVIEW is 1,024 characters. If a PanelBuilder 1400e expression contains more than 1,024 characters, the excess characters are not converted.

Some PanelBuilder 1400e objects support both tags and expressions. For these objects, if the text assigned to a control could be valid syntax for both a tag and an expression, the control is treated as an expression, and is therefore turned off.

For example, N20-0_String_64 could be the name of a tag, or it could be an expression that subtracts “0_String_64” from the tag “N20.” The text would be converted as an expression, and turned off.

To turn on an expression

1. Select the object containing the expression.
2. Open the Property Panel, and then click the Connections tab.
3. In the Exprn column, click the Browse button beside the expression to turn on.



Browse button

4. In the Expression editor, delete the warning text and exclamation marks.
5. Revise the expression, if necessary, using the tables in the following three sections as guides.
6. Click Check Syntax.

For more information about using the Expression editor, see Chapter 19 or Help.

Equivalent expression syntax

This table describes RSVView expression syntax that is equivalent to PanelBuilder 1400e syntax. When you edit the converted expressions, replace the PanelBuilder 1400e syntax with the RSVView equivalent.

Syntax that is not listed in this table or in the next section is okay the way it is.

Type of expression component or operator	PanelBuilder 1400e syntax	RSView syntax
Comment	REM or ‘	!
Line continuation	_ (underscore)	Not needed.
Equality	(=)	EQ or ==
Bitwise Not	Not	~ (tilde)
Bitwise And	And	&
Bitwise Or	Or	(pipe)
Bitwise XOr	XOr	^
<p>If both operands are Byte, Integer, Long, Variant, or any combination of these data types, use the RSVView syntax. For other data types, no change is needed.</p>		

Unsupported expression syntax

This table describes the PanelBuilder 1400e expression syntax that is not supported in RSView with information about how to achieve the same result where possible.

Type of expression component or operator	PanelBuilder 1400e syntax	Equivalent RSView syntax (if any)
Exit statement	Exit	
Local variables	DIM <i>varname</i> AS ... <i>varname</i> =	
Integer division	\	$(x - (x \text{ MOD } y)) / y$
Endif	If then endif If then else endif	If then else 0 If then else
Select case	Select Case Case1...CaseN CaseElse EndSelect	Use nested if-then-else.
Logical Xor (if one or both operands are Boolean or Single data types)	Xor	NOT ((<i>x</i> AND <i>y</i>) OR NOT (<i>x</i> OR <i>y</i>))

Order of precedence

The order of precedence is slightly different in RSView. Check your expressions to make sure the result is what you intend.

PanelBuilder 1400e order of precedence	RSView order of precedence
()	()
- (negation)	NOT, ~ (tilde)

PanelBuilder 1400e

RSView order of precedence

order of precedence

<div><div>*</div><div>/ (floating point division)</div></div>	<div><div>*</div><div>/</div><div>MOD</div><div>%</div><div>**</div><div>AND</div><div>&&</div><div>&</div><div>>></div><div><<</div></div>
<div><div>\ (integer division)</div></div>	<div><div>+</div><div>-</div><div>OR</div><div> </div><div> </div><div> </div><div>^</div></div>
<div><div>MOD</div></div>	<div><div>EQ</div><div>==</div><div>NE</div><div><></div><div>LT</div><div><</div><div>GT</div><div>></div><div>LE</div><div><=</div><div>GE</div><div>>=</div></div>
<div><div>+</div><div>- (subtraction)</div></div>	
<div><div>=</div><div><></div><div><</div><div>></div><div><=</div><div>>=</div></div>	
<div><div>Not</div></div>	
<div><div>And</div></div>	
<div><div>Or</div></div>	
<div><div>Xor</div></div>	

For more information about order of precedence, see page 19-15.

B

Converting PanelBuilder and PanelBuilder32 applications

This appendix describes:

- terms that are different in PanelBuilder™ and RSVIEW®.
- steps for converting PanelBuilder applications.
- names of equivalent graphic objects in PanelBuilder and RSVIEW.
- PanelBuilder graphic objects that are not supported in RSVIEW.
- PanelBuilder settings and controls that are not supported in RSVIEW.
- how communications are converted and which PanelBuilder communication protocols are not supported in RSVIEW.
- converting PanelBuilder Remote I/O communications.
- PanelBuilder graphic object features that are not supported in RSVIEW, with information about how to achieve the same result when possible.

This appendix uses the term PanelBuilder to refer to both PanelBuilder and PanelBuilder32 features.

For information about converting applications from PanelBuilder 1400e, see Appendix A.

Terminology

This section describes terms that are different in PanelBuilder and RSView.

PanelBuilder term	RSView term
screen	display, graphic display
programmable controller	data source

In RSView, the data source can be memory or a device such as a programmable controller or an OPC® server. RSView writes values to and reads values from the data source. The data source is configured to exchange information (in the form of numeric or string values) between RSView and the machine that your application is controlling. The general term data source is used unless specifically discussing a programmable controller.

Summary of steps

Follow these steps to convert PanelBuilder applications:

1. Convert the application file, as described in the next section.
2. Specify additional project settings, as described on page 4-11.

For example, if you want the application to have a border around its graphic displays, or to use a title bar, you can specify these options in the Project Settings editor.

3. If you select Convert to new window size, check the position of the graphic objects in each display.
4. Set up communications and edit tags that don't convert directly.

For more information, see page B-9.

5. Set up graphic object features that don't convert directly.

For more information, see page B-12.

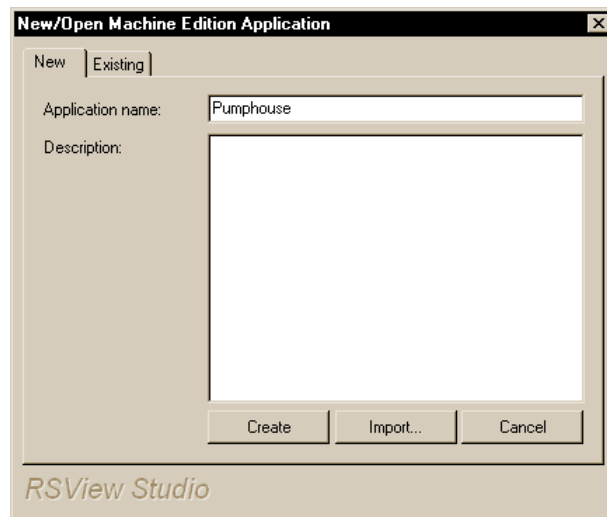
Converting PanelBuilder application files

Follow these steps to convert a PanelBuilder application file, with the extension .pba or .pva, to an RSView application file, with the extension .med. The original PanelBuilder application file is not modified by the conversion.

You can convert the PanelBuilder application when you open RSView Studio™, or once RSView Studio is already open.

To convert a PanelBuilder application when you open RSView Studio

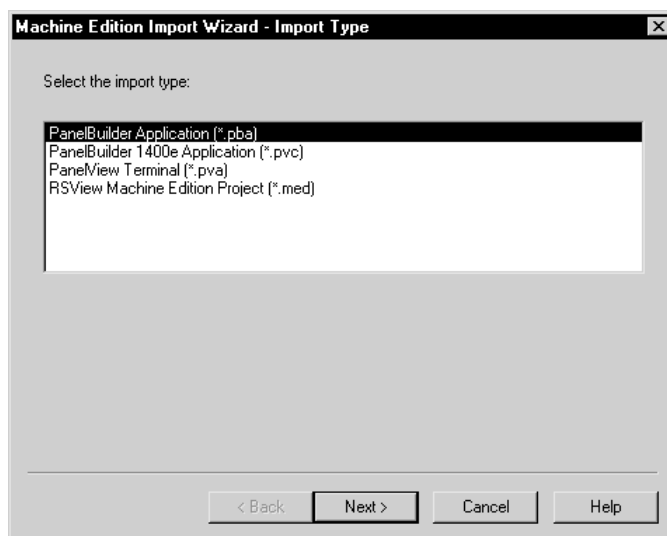
1. Open RSView Studio.



2. In the New tab, in the Application name box, type a name for your converted application, up to 32 characters long.
3. If desired, type a description of the application. If the PanelBuilder application contains an Application Description, the Application Description will overwrite the description you type here.

You can add or change the description later, in the Application Properties dialog box. For details, see Help.

4. Click Import.



5. Follow the steps in the Machine Edition Import Wizard.

For details about the options in the Machine Edition Import Wizard, see Help.

When you complete the steps of the wizard, RSView Studio converts the PanelBuilder application, creates the converted application's folders and files, and then displays the converted application in the Application Explorer window in RSView Studio.

If there are any messages about conversion, they are displayed automatically in the Project Status dialog box.

The converted application is created in the ME\HMI projects directory, in a folder with the same name as the application name you specified in step 2.

This is the path to the ME\HMI projects directory:

\Documents and Settings\All Users\Documents\RSView
Enterprise\ME\HMI projects (Windows® 2000)

or

\Documents and Settings\All Users\Shared Documents\RSView
Enterprise\ME\HMI projects (Windows XP)

Conversion messages are saved in a file called Convert.log, in the HMI projects directory.

To convert a PanelBuilder application when RSView Studio is already open



New Application

1. On the File menu, click New Application, or click the New Application tool.

If an application is already open, RSView Studio asks you whether to close the application that is currently open. Click Yes.

2. Follow steps 2 through 5 in the previous procedure.

Equivalent graphic objects

This section describes graphic objects that are equivalent in PanelBuilder and RSView, but have different names in the two products.

This PanelBuilder object	Is converted to this RSView object	Notes
Numeric Entry Keypad Enable Button	Numeric input enable button	
Numeric Entry Cursor Point	Numeric input enable button	

This PanelBuilder object	Is converted to this RSVIEW object	Notes
Increment/Decrement Entry Button	Numeric input enable button	The numeric input enable button is set up to work as a ramp button, using the Fine Step value. The Coarse Step value is not converted.
ASCII Entry Keypad Enable button	String input enable button	The Show Current String on ASCII Scratchpad setting is not converted. The pop-up scratchpad or keyboard is always blank when opened.
ASCII Entry Cursor Point	String input enable button	The Show Current String on ASCII Scratchpad setting is not converted. The pop-up scratchpad or keyboard is always blank when opened.
Message Display	Multistate indicator	
Numeric Data Display	Text	The text object contains a numeric embedded variable that displays the read tag.
Connected Line	Polyline	
Circle	Ellipse	The ellipse has a circular shape.
Freeform	Freehand	
Screen List Selector	Display list selector	
Goto Screen Button	Goto display button	
Return Screen Button	Return to display button	
Print Alarm List Button	Print alarm history button	
Clear Alarm List Button	Clear alarm history button	

Unsupported graphic objects

These PanelBuilder objects are not supported in RSVIEW:

- Print Only Object
- Circular Scale
- Scrolling Text
- Print Alarm Button
- Horn Silence Button
- Lamp/Horn Test Button
- Select Operator Button
- Enable/Disable Security Button
- New Password Button
- Verify Password Button

Unsupported settings and controls

This section describes PanelBuilder settings and controls that are not used in RSVIEW.

Settings and controls for alarms

RSVIEW does not use these PanelBuilder features and settings to manage alarms:

- Ack setting for alarm messages; in RSVIEW, all alarms can be acknowledged
- bit alarm acknowledgement

RSVIEW does not use these PanelBuilder controls to manage alarms:

- Remote Ack All Handshake Tag

- Remote Clear All Alarm Tag
- Remote Clear All Alarm Handshake Tag

Invalid characters in screen names and tag names

Characters in PanelBuilder screen names and tag names that are not supported in RSVIEW are replaced with the underscore character.

Time and date

PanelBuilder time and date formats are not converted. For details about RSVIEW time and date formats, see page 16-20.

External fonts

PanelBuilder external fonts are not converted. When you convert your application you can specify the font to use instead. For details, see Help for the Machine Edition Import Wizard.

Screen security settings

PanelBuilder screen security settings are not converted, because RSVIEW uses a different method to assign security to graphic displays. For information about setting up security in RSVIEW, see Chapter 13.

Power-up options

These PanelBuilder power-up options are not imported into RSVIEW:

- Write Last Terminal State to Controller
- Display Last User Screen
- Use Terminal Presets

Converting non-RIO communications

This section describes how communications that do not use Remote I/O (RIO) are converted. For information about converting RIO communications, see page B-10.

RSView does not use nodes for communications. Nodes are converted to RSLinx topics. Topics are then converted into device shortcuts, to run with RSLinx Enterprise. You must have both RSLinx® for RSView and RSLinx Enterprise installed to make this two-step conversion.

If you import an application multiple times, delete the device shortcuts in RSLinx Enterprise before re-importing. Otherwise, multiple unused topics will be created in RSLinx Enterprise.

For more information about setting up communications, see Chapter 5.

Unsupported communication protocols

These communication protocols are not supported in RSView:

- ControlNet™ scheduled communications
- Modbus communications
- Profibus DP communications
- DeviceNet communications
- DHPlus nodes that use the AutoMax node type
- invalid PanelBuilder node types and non-existent node names that are saved with the PanelBuilder application

Tags that use these protocols are converted to HMI memory tags. Once you have set up communications for your converted application, change the memory tags to device tags that point to the correct addresses. For information about editing HMI tags, see Chapter 7.

Bit array tags

You can monitor bit arrays for alarm conditions in RSView, but you can't assign bit arrays to most graphic objects or write to bit arrays. (The only exception is the piloted control list selector object. For this object, you can assign a bit array tag to the Visible States control.)

All bit array tags in your PanelBuilder application are converted to HMI memory tags.

For information about monitoring bit arrays for alarm conditions, see Chapter 9. For information about editing HMI tags, see Chapter 7. For information about the piloted control list selector, see Help.

Converting RIO communications

Remote I/O (RIO) communications are not supported for applications that use the Windows 2000 or Windows XP operating systems. You can use RIO communications on the PanelView Plus™ and VersaView® CE runtime platforms.

To convert an RIO application from PanelBuilder to RSView

1. Convert the application, as described on page B-3.
2. Open the RSLinx Enterprise data server, and then double-click Communication Setup.
3. In the Communication Setup editor, add an RIO driver.
 - For PanelView Plus 400 and 600 terminals, use the 2711P-RN1 driver.
 - For all other PanelView Plus or VersaView CE terminals, use the 2711P-RN6 driver.

For information about adding drivers in RSLinx, see RSLinx Help.

4. Expand the RIO tree, right-click RIO Data, and then click Configure RIO.

5. In the RIO Configuration dialog box, right-click RIO, and then click Import.
6. Browse to the location of the RIO configuration file.

The file is saved in the root of the application's directory.

7. In the Communications Setup editor, create a device shortcut named "PVRIO" that points to the RIO data device.

For information about creating a device shortcut, see RSLinx Help.

8. Correct any invalid RIO configurations. Invalid RIO configurations are highlighted with red "x" icons.
9. Save the converted RIO application.

Unsupported PanelBuilder RIO tags

A PanelBuilder RIO tag will be converted to an HMI memory tag and an error will be logged to the conversion log file if the RIO tag:

- has a blank address.
- has a data type of Bit Array and its address does not have an array size of 1, 8, 16, or 32.
- has a data type of Bit Array, and its array size is 16 or 32, but its address does not have a bit offset of 0.
- has a data type of Bit or BOOL, and its address does not contain the bit delimiter character "/".
- has a data type of 4-BCD, Unsigned Integer, Signed Integer or INT, Character Array, or DINT, and its address contains the bit delimiter character "/".
- is a block transfer tag with a data type of Bit Array, and its array size is 8, but its address does not have a bit offset of 0 or 8.

- is a block transfer tag with a data type of SINT, and its address does not have a bit offset of 0 or 8.
- is an I/O tag with an address that references an undefined rack.
- is an I/O tag with a data type of SINT, and its address does not have a bit offset of 0 or 10.
- does not have a valid I/O address or block transfer address.

Unsupported graphic object features

This section describes features of PanelBuilder graphic objects that are not supported in RSView. The Notes column provides additional information and describes methods for achieving the same result when possible.

Graphic object	Unsupported feature in RSView	Notes
Image text, arc, ellipse, freehand, line, polyline, rectangle, wedge	Blink property	In RSView, all of the listed objects except images use color animation to blink. For details, see page 17-12. Color images do not blink. Monochrome images use the Blink property to blink.
Increment/Decrement Entry Button (Converted to numeric input enable button)	Allow Home/End Allow Wrap Ramping by coarse steps	

Graphic object	Unsupported feature in RSVIEW	Notes
Maintained Push Button, Multistate Push Button, Standard Control List Selector	Initial state values	If you want to set these objects' states on application startup, create a macro to set the appropriate tag values for the objects' controls. For information about macros, see Chapter 21. Assign the macro in the Startup editor. For details, see Help.
Multistate Indicator, Message Display	Print Setting	
Bar Graph	Inner text and inner graphic	Converted to a separate text object and image object.
Gauge	Inner text and inner graphic	Converted to a separate text object and image object.
	Scale clipping	If the scale doesn't fit within the height or width of the gauge, it is not clipped. Check the position of the scale to ensure it doesn't overlap other objects.
	Needle	Converted to a separate gauge object; if the gauge had 2 needles, each needle is converted to a separate gauge object.
Alarm List	No Acknowledgement Required	All alarms can be acknowledged.
All objects	Image placement	RSVIEW supports one, three or nine positions for images, depending on the type of object. On conversion, images are positioned using the closest match. Therefore some images might be clipped to fit the object.

Graphic object	Unsupported feature in RSVIEW	Notes
All objects	Turn Object View On property	If this property is set to False, the converted object has a transparent background, no border, no caption, and no image.
All objects	Blinking inner graphics	If the inner graphic uses a color image, it will not blink. Use a monochrome image if you want the inner graphic to blink.

C

System tags

This appendix describes the system tags.

System tags are preconfigured HMI tags created by RSView®. System tags are read-only. Display them as needed in your application.

Alarms

The following tag contains the time and date that the status of alarms was last reset. The date uses the long date format.

Tag name	Type	Function
system\AlarmReset DateAndTimeString	String	Contains the date and time of the last alarm reset.

For information about resetting alarms, see page 9-10.

Graphics

The following HMI tags can be used to make graphic objects appear as though they are blinking on and off:

Tag name	Type	Function
system\BlinkFast	Digital	Toggles on and off every 100 ms (10 times per second).
system\BlinkSlow	Digital	Toggles on and off every 500 ms (twice per second).

A more efficient way to make graphic objects blink is to use the blinking color option in color animation. For details, see page 17-12.

Also, many objects have a Blink property that you can set up. For information about specific objects, see Help.

Time

These HMI tags record time and date information in various formats:

Tag Name	Type	Provides this data	Read or write
system\Date	String	System date.	Read only
system\DateAndTime Integer	Analog	Number of seconds elapsed since midnight (00:00:00) January 1, 1970, coordinated universal time.	Read only
system\DateAndTime String	String	Complete date and time display. For example: Monday, December 12 2001 10:47:50 AM	Read only
system\DayOfMonth	Analog	Day of the month (1 - 31).	Read only
system\DayOfWeek	Analog	Day of the week (1-7); Sunday = 1.	Read only
system\DayOfYear	Analog	Day of the year (1-366).	Read only
system\Hour	Analog	Hour of the day (0-23).	Read and write
system\Minute	Analog	Minutes (0 - 59).	Read and write

Tag Name	Type	Provides this data	Read or write
system\Month	Analog	Number for month (1-12).	Read only
system\MonthString	String	Name of the month.	Read only
system\Second	Analog	Seconds (0 - 59).	Read and write
system\Time	String	System Time.	Read only
system\Year	Analog	The year (1980-2099).	Read only

For information about using the data source to update the system date and time, or about sending the runtime computer's date and time to the data source, see Chapter 8.

User

This tag contains the name of the current user:

Tag Name	Type	Function
system\User	String	Contains name of logged-in user.

We recommend that you use the expression security function `CurrentUserName()` instead of the `system\User` tag, especially if you intend to convert the application to RSVIEW Enterprise Supervisory Edition. In distributed applications, `system\User` returns the name of the user logged into the HMI server, not the user logged into the display client.

For more information about the security functions, see page 19-18.

D

ODBC database schema

This appendix describes the ODBC database format, or schema, for messages from FactoryTalk™ Diagnostics. The target table of the ODBC database to which you are sending messages must use the format shown in this appendix.

The option of logging FactoryTalk Diagnostics messages to an ODBC database is available for computers running Windows® 2000 and Windows XP only.

For information about setting up FactoryTalk Diagnostics, see Chapter 10.

FactoryTalk Diagnostics log table

FactoryTalk Diagnostics log data in ODBC format uses one table.

This column	Contains	SQL data type	Length
TimeStamp	The time and date data was logged, in coordinated universal time format. Encoded as a date variant.	SQL_TIMESTAMP	Driver dependent
MessageText	Message to be logged.	SQL_VARCHAR, <i>or</i> SQL_CHAR	254
Audience	A number representing the message audience: 0 for Operator 1 for Engineer 2 for Developer 3 for Secure	SQL_SMALLINT, <i>or</i> SQL_INTEGER	1

This column	Contains	SQL data type	Length
Severity	A number representing the severity of the diagnostics message: 0 for Error 1 for Warning 2 for Information 3 for Audit	SQL_SMALLINT, <i>or</i> SQL_INTEGER	1
Area	The FactoryTalk path to the area in which the activity occurred. Used for RSView [®] Supervisory Edition [™] only.	SQL_VARCHAR, <i>or</i> SQL_CHAR	80
Location	The name of the computer where the message was generated.	SQL_VARCHAR, <i>or</i> SQL_CHAR	15
UserID	The name of the user (including domain name, if there is one) that initiated the action that caused the diagnostics message. If the diagnostics message was caused by an HMI server, the user column contains "System."	SQL_VARCHAR, <i>or</i> SQL_CHAR	38
UserFullName	The full name of the user that was logged in when the activity occurred.	SQL_VARCHAR, <i>or</i> SQL_CHAR	255
Provider	The name of the product that generated the message.	SQL_VARCHAR, <i>or</i> SQL_CHAR	20

E

Importing and exporting XML files

This appendix describes:

- creating alarm and graphics XML files.
- exporting, editing, and importing XML files.
- the alarm XML file structure.
- the graphics XML file structure.

About XML

XML is the Extensible Markup Language used to create documents with structured text information. It has a standardized format and structure. You can use XML to edit the elements and attributes needed to create an alarm setup file or to modify graphic displays.

For example, if you have a list of 100 tags to monitor for alarms, with multiple messages for each tag, you might prefer to enter all the information in a text editor, and then import the alarm setup information into RSView®.

Another example of using XML files is to export the alarm setup information you develop in one application, import the setup information to another RSView application, and then modify the alarm setup as needed. Or, you could modify the information in the XML file before importing it.

For more information about XML, see the World Wide Web Consortium's web page about XML at:
<http://www.w3.org/XML>.

Creating XML files by exporting

The quickest way to create an XML file for your application's alarm setup or graphic displays is to export the data from RSView. You can then open the XML file in Notepad, make your changes, and import the file back into RSView.

To export alarm information to an XML file

1. In the Application Explorer, right-click the Alarm Setup editor.
2. Click Import and Export.

The Alarm Import Export Wizard opens.

3. Follow the instructions in the wizard.

For information about using the Alarm Import Export Wizard, see Help.

RSView creates a file with the name you specify, in the location you specify.

To export graphic display information to an XML file

1. In the Application Explorer, right-click the Displays editor.
2. Click Import and Export.

The Graphics Import Export Wizard opens.

3. Follow the instructions in the wizard.

For information about using the Graphics Import Export Wizard, see Help.

RSView creates XML files for the selected graphic displays, in the location you specify.

If you export multiple graphic displays, RSView creates a file called `BatchImport_`*Application name*.xml, in the same location. You can use this file to import multiple displays at the same time. To import a different set of displays than you exported, edit the list of graphic display names in the `BatchImport_`*Application name*.xml file.

Editing XML files

We recommend that you use Notepad to edit your XML files.

If you do not want to change a property that uses the default value, you don't need to include it in the XML file. Properties that are not listed in the file are set to their default values when the file is imported.

IMPORTANT

If you include attributes for an object whose name does not match one of those in the graphic display, the attributes for that object are not imported. Attributes for all other objects in the file whose names do match the ones in the graphic display are imported.

Saving XML files in Notepad

Save XML files created or edited in Notepad using either UTF-8 or UTF-16 file format. Notepad's Unicode file type corresponds to UTF-16 file format. For files containing strings in English or other Latin-based languages, UTF-8 is recommended, to reduce the size of the XML file. For other languages such as Chinese, Japanese, or Korean, UTF-16 is recommended.

The first line of every XML file contains XML version and encoding attributes. Make sure the encoding attribute matches the format that you are going to use when you save the file. For example, if the original file was saved in UTF-8 format and you plan to save it in UTF-16 format, make sure the first line specifies `encoding="UTF-16"`.

Testing XML files

An XML file must be well-formed to be imported. To find out whether your XML file is well-formed, test it.

To test an XML file

- Open the XML file in Internet Explorer.

If you can see the XML code, your file is well-formed. If the XML code is not well-formed, Internet Explorer displays an error message.

Importing XML files

You can import an alarm setup or graphic display that has been created using an external programming tool or editor, or you can import an XML file that you originally exported from RSView and then modified.

IMPORTANT

When you import an alarm setup or graphic display, your existing alarm setup or graphic display will be overwritten. Back up your application first, using the Application Manager tool. Or, you can save a copy of your existing alarm setup or graphic display by exporting it to an XML file before you import the new one.

Error log file

If errors occur during importing, the errors are logged to a text file. The file opens automatically when importing is finished. The last paragraph of the file lists the location of the file.

Importing alarm XML files

To import alarm information from an XML file

1. In the Application Explorer, right-click the Alarm Setup editor.
2. Click Import and Export.

The Alarm Import Export Wizard opens.

3. Follow the instructions in the wizard.

For more information about using the Alarm Import Export Wizard, see Help.

Importing graphics XML files

You can import a single graphic display XML file at a time, or import multiple graphic displays. You can also choose whether to import new objects or update existing objects.

To import multiple graphic displays, specify the names of the graphic displays in the file BatchImport_*Application name*.xml. RSVIEW creates this file when you export multiple graphic displays. For details, see page E-2.

To import graphic display information from an XML file

1. In the Application Explorer, right-click the Displays editor.
2. Click Import and Export.

The Graphics Import Export Wizard opens.

3. Follow the instructions in the wizard.

For more information about using the Graphics Import Export Wizard, see Help.

Alarm setup XML file structure

The alarm setup XML file is an RSView XML document that describes the alarm setup for an application. The root element of the XML document is called **alarms**. It represents the Alarm Setup editor. An XML document can contain only one root element. All other elements in the document must be contained or nested within the root element.

In an XML document, the start of an element is marked **<element name>**. The end is marked **</element name>**.

If the element contains no subelements, the end can be marked **/>**. For example, **<trigger id="T1" type="value" ack-all-value="0" />**.

The syntax for specifying an attribute for an element is **attribute="value"**. The attribute value must be enclosed in single or double quotes.

Here is a sample structure for an alarm XML document:

Element	Description
<alarms>	Root element.
<alarm>	Contains attributes from the Advanced tab of the Alarm Setup editor, as well as the triggers and messages elements.
<triggers>	Contains a trigger element for each trigger in the Triggers tab of the Alarm Setup editor.
<trigger id="T1" />	Contains attributes for the first alarm trigger.
<trigger id="T2" />	Contains attributes for the second alarm trigger.
</triggers>	Indicates the end of the triggers element.

Element	Description
<messages>	Contains a message element for each message in the Messages tab of the Alarm Setup editor.
<message id="M1" />	Contains attributes for the first alarm message.
<message id="M2" />	Contains attributes for the second alarm message.
</messages>	Indicates the end of the messages element.
</alarm>	Indicates the end of the alarm element.
</alarms>	Indicates the end of the alarms element.

You can specify multiple attributes for an element. For example, the alarm element contains 11 possible attributes from the Advanced tab of the Alarm Setup editor.

For more information about alarm elements and their attributes, see Help for the Alarm Import Export Wizard.

Graphic display XML file structure

The graphic display XML file is an RSView XML document that describes the objects and settings for a graphic display. The root element of the XML document is called **gfx**. It represents the graphic display. An XML document can contain only one root element. All other elements in the document must be contained or nested within the root element.

In an XML document, the start of an element is marked **<element name>**. The end is marked **</element name>**.

If the element contains no subelements, the end can be marked **/>**. For example, **<caption fontFamily="Arial" fontSize="8" bold="false" />**.

The syntax for specifying an attribute for an element is **attribute=“value”**. The attribute value must be enclosed in single or double quotes.

Here is a sample structure for a graphic display XML document containing two graphic objects. The second object has states:

Element	Description
<code><gfx></code>	Root element.
<code><displaySettings /></code>	Contains attributes from the Display Settings dialog box in the Graphic Displays editor.
<code><object1></code>	Contains attributes from the General and Common tabs in the object's Properties dialog box, as well as elements for the object's caption, image, animation, and connections.
<code><caption /></code>	Contains attributes for the object's caption.
<code><imageSettings /></code>	Contains attributes for the object's image.
<code><animations></code>	Contains an animation element for each type of animation set up for the object.
<code><animateVisibility /></code>	Contains attributes for Visibility animation.
<code><animateColor /></code>	Contains attributes for Color animation.
<code></animations></code>	Indicates the end of the animations element.
<code><connections></code>	Contains a connection element for each connection assigned to the object.
<code><connection name= "Value" /></code>	Contains attributes for the Value connection.
<code><connection name= "Indicator" /></code>	Contains attributes for the Indicator connection.
<code></connections></code>	Indicates the end of the connections element.
<code></object1></code>	Indicates the end of the <i>object1</i> element.

Element	Description
<code><object2></code>	Contains attributes from the General and Common tabs in the object's Properties dialog box, as well as elements for the object's states and connections.
<code><states></code>	Contains state elements for each of the object's states.
<code><state stateid="0"></code>	Contains attributes for the object's first state, as well as elements for the state's caption and image.
<code><caption /></code>	Contains attributes for the state's caption.
<code><imageSettings /></code>	Contains attributes for the state's image.
<code></state></code>	Indicates the end of the state element.
<code><state stateid="1"></code>	Contains attributes for the object's second state, as well as elements for the state's caption and image.
<code><caption /></code>	Contains attributes for the state's caption.
<code><imageSettings /></code>	Contains attributes for the state's image.
<code></state></code>	Indicates the end of the state element.
<code></states></code>	Indicates the end of the states element.
<code><connections></code>	Contains a connection element for each connection assigned to the object.
<code><connection name= "Value" /></code>	Contains attributes for the Value connection.
<code><connection name= "Indicator" /></code>	Contains attributes for the Indicator connection.
<code></connections></code>	Indicates the end of the connections element.
<code></object2></code>	Indicates the end of the <i>object2</i> element.
<code></gfx></code>	Indicates the end of the gfx element.

You can specify multiple attributes for an element. For example, the caption element contains 13 possible attributes.

Elements for group objects begin with **<group>** and end with **</group>**. The **<group>** element contains all the elements for each object in the group.

For more information about graphic object elements and their attributes, see Help for the Graphics Import Export Wizard.

F

Exporting text in your application for translation

This appendix describes:

- which text in your application cannot be exported.
- how to export and import text.
- troubleshooting import errors.
- information for translators, including the file format of the text file and information about what parts of the file must not be translated. Give this information to the translator to ensure that the file can be imported after translation.

About exporting text for translation

RSView® Machine Edition™ allows you to export the application text that operators see at run time. The text strings in the HMI server are exported to separate text files in Unicode format, which you can then send to translators for translation into foreign languages.

When the translators send the translated files back to you, you can import them into your application, substituting the new language of the translated text files for the original language.

RSView Machine Edition allows an application to have only one language at a time. For each language, you must make a copy of your application, and then import the translated text file.

Text that you cannot export

This text cannot be exported from your application:

- Text that is part of the RSVIEW software, such as:
 - text on the pop-up keyboard or numeric keypad.
 - text in the Login dialog box that opens at run time.
- Text that can be displayed at run time, but is used to operate your application. For example, the names of graphic displays.
- Names of objects.
- Description for tags in the tag database.
- String constants in expressions, and in all components that use expressions.

System Tags

You might use system tags in your application to display text in graphic displays. System tags are always shown in the format prescribed by the language of the Windows® operating system—if you are running the English language version of the Windows operating system, the text in system tags is displayed in English, and dates and numbers are formatted for U.S. English. These system tags include:

- Time
- Date
- DateAndTimeString
- MonthString
- AlarmResetDateAndTimeString

For details about system tags, see Appendix C.

Text you add to your application that is not displayed at run time

Some of the text that you can add to your application is never shown to the operator at run time. RSVIEW Machine Edition deliberately does not export this text because it is not displayed at run time.

These text strings are not exported:

Graphic object	Property
all objects	Extended Name Description Image St_Image LinkedObject
Display list selector	St_Display St_Parameter
Goto display button	Display Parameter
Local message display	MessageFile
Alarm list, alarm banner, and alarm status list	FilteredTriggers
Macro button	Macro

Exporting text

Text in your application is exported to tab-delimited text files in Unicode format.

To export text in your application to a text file

1. In the Application Explorer, right-click the root of the application tree, and then click String Import and Export.

The String Import Export Wizard opens.

2. Follow the steps in the wizard. For details, see Help.

If you cancel the export before it is complete, any files that were created are not deleted, but their contents might not be complete.

The exported text file is named *ApplicationName.txt*. If you export the file multiple times to the same location, previously exported versions will be overwritten.

Importing text

To import text from a file into your application, the file must be saved in Unicode text format.

Text strings that exist in your application, but are not in the text file, are not overwritten. This allows you to import only the strings that have been modified.

To import text into your application from a text file

1. Create a backup of the text currently in your application by exporting it.

This is recommended because this allows you to restore the original text to your application if an error occurs while importing, or if you cancel the import before it is complete.

For details, see the previous section.

2. In the Application Explorer, right-click the root of the application tree, and then click String Import and Export.

The String Import Export Wizard opens.

3. Follow the instructions in the wizard. For details, see Help.

If you cancel the import before it is complete, any text strings that were changed are not restored to their original values. To restore the text that was originally in the application, import the text from the backup text file you created in step 1.

Troubleshooting importing

You do not have to check every graphic display in your application to verify that text was imported correctly. If errors occurred while importing text, they are displayed automatically from a log file called ImportErrors.txt in the following folder:

C:\Documents and Settings\All Users\Documents\
RSView Enterprise\Strings (Windows 2000)

or

C:\Documents and Settings\All Users\Shared Documents\
RSView Enterprise\Strings (Windows XP)

Each time errors occur while importing text into an application, the ImportErrors.txt file is overwritten.

If errors occurred while importing text, or if the import was canceled, a message appears in the Diagnostics List, and in the FactoryTalk™ Diagnostics log file.

Common errors and their causes

If text seems to have been imported for some HMI servers and not for others, see the ImportErrors.txt file for an error message. The causes and solutions for common errors are described below.

If some, but not all, of the text in an HMI server seems to have been modified, the import might have been canceled. If you cancel the import before it is complete, any text strings that were changed are not restored to their original values. To restore the text originally in the application, import the text from the backup text file you created in step 1 on page F-4 .

In the error messages that follow, *x* is the line number in the text file.

Error message	Cause and solution
<i>Server</i> defined in <i>FileName.txt</i> does not exist. None of the strings in this file were imported.	The name of the HMI server is invalid. If the HMI server was renamed, open the file, and then correct the name of the HMI server.
File " <i>FileName</i> ", Line <i>x</i> . <i>ComponentType</i> is not a valid component type for the application. The string was not imported.	The component type has been modified in the text file. Open the text file, and then correct the text for the component type.
File " <i>FileName</i> ", Line <i>x</i> . <i>ComponentName</i> is not a valid component for the application. The string was not imported.	The component name has been modified in the text file, or the component in the application has been renamed. Open the text file, and then correct the text for the component name.
File " <i>FileName</i> ", Line <i>x</i> . The string reference must be an unsigned long integer value between 1 and 4294967295. The string was not imported.	The string reference number has been modified in the text file, and the new string reference number is invalid. Open the backup text file, and then copy the correct the string reference number into the translated text file.
File " <i>FileName</i> ", Line <i>x</i> . The string was not used in the application and was not imported.	This error occurs if: <ul style="list-style-type: none">■ the string reference number has been modified in the text file, and the new string reference number is not used in the application. Open the backup text file, and then copy the correct the string reference number into the translated text file.■ the object was deleted from the application after the text was exported. If this is correct, ignore the error.
File " <i>FileName</i> ", Line <i>x</i> . The string definition must be contained within double quotes. The string was not imported.	The translated string definition includes embedded double quotes, but the string definition itself was not enclosed in double quotes. Open the text file, and then enclose all string definitions containing embedded double quotes in double quotes. For example, the string definition Start "Backup motor" must be enclosed in double quotes, like this: "Start "Backup motor"

Error message	Cause and solution
File “ <i>FileName</i> ”, Line x: Invalid line format!	A line in the import file does not contain all the component name or string reference number fields. The import continues with the next line in the file. Open the backup text file, and then copy the missing fields into the translated text file.
Unable to open { <i>FileName.txt</i> }. None of the strings from this file were imported.	The text file could not be opened. Make sure the text file is in the folder from which you are importing files, and that you can open the text file in Notepad or Microsoft® Excel.

Information for translators

This section contains information about the format and schema of the exported file.

IMPORTANT

Because RSView Enterprise requires that parts of the text file remain the way they were exported, give the information in this section to the translator, to ensure that the file can be imported after it has been modified.

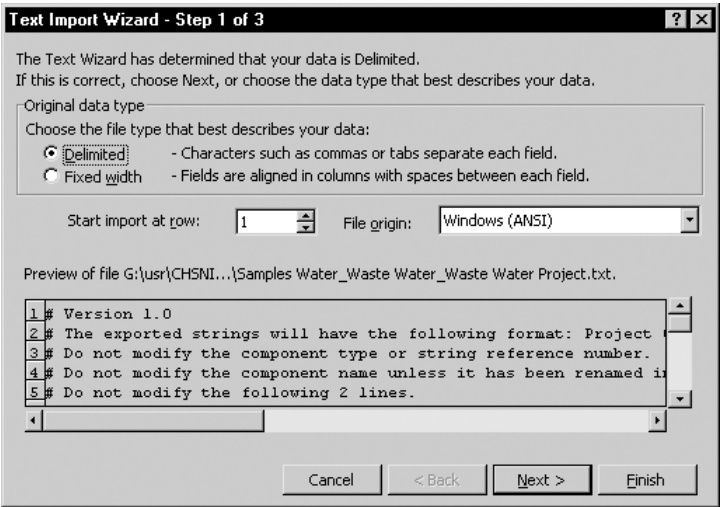
File name and format

You can rename the exported file, for example, to distinguish the translated version from the original.

To import text into an RSView Enterprise application, you must save the file as tab-delimited text, in Unicode text format.

Opening the text file in Microsoft Excel

When you open the text file in Microsoft Excel, the Text Import Wizard appears.



To specify the file format (Step 1)

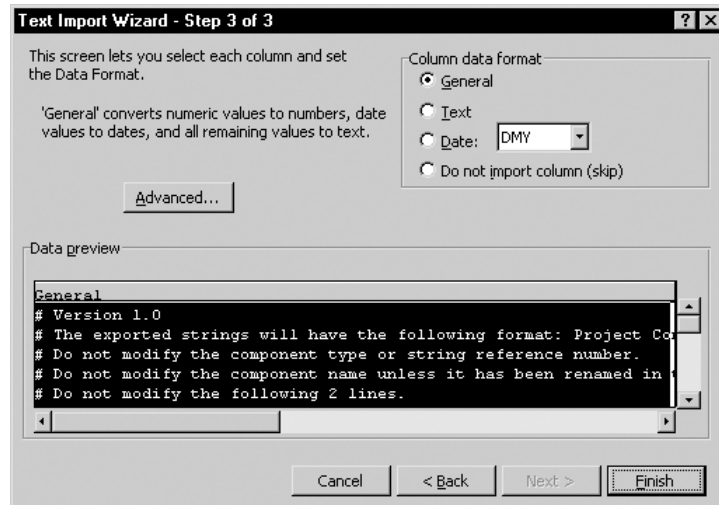
1. Click Delimited.
2. In the Start import at row box, type or select 1.
3. In the File origin list, click Windows (ANSI).
4. Click Next.

To specify the field delimiter (Step 2)

The screenshot shows the 'Text Import Wizard - Step 2 of 3' dialog box. It has a title bar with a question mark and a close button. The main text area says: 'This screen lets you set the delimiters your data contains. You can see how your text is affected in the preview below.' Below this is a 'Delimiters' section with a group box containing five checkboxes: 'Tab' (checked), 'Semicolon', 'Comma', 'Space', and 'Other' (with an empty text field). To the right of this group box is a checkbox labeled 'Treat consecutive delimiters as one' which is unchecked. Below that is a 'Text qualifier' dropdown menu showing a double quote character. At the bottom is a 'Data preview' section with a text area containing the following text: '# Version 1.0', '# The exported strings will have the following format: Project Co', '# Do not modify the component type or string reference number.', '# Do not modify the component name unless it has been renamed in', and '# Do not modify the following 2 lines.'. At the very bottom are four buttons: 'Cancel', '< Back', 'Next >', and 'Finish'. The 'Next >' button is highlighted with a dashed border.

1. Select the Tab check box. If any other check boxes are selected, clear them.
2. Make sure the “Treat consecutive delimiters as one” check box is cleared.
3. Click Next.

To specify the column data format (Step 3)



1. If it is not selected already, under Column data format, click General.
2. Click Finish.

Saving the text file in Microsoft Excel

To save the file

1. On the File menu, click Save.

Excel warns that the file may contain features that are not compatible with Unicode text.

2. When prompted to keep the workbook in Unicode format that leaves out incompatible features, click Yes.
3. Close the file.
4. When prompted to save changes, click Yes.
5. When prompted again to keep the workbook in Unicode format that leaves out incompatible features, click Yes.

Differences in file format for files saved in Excel

If you use Notepad to open a Unicode text file that was saved in Excel, you will notice some differences from a file edited and saved in Notepad.

IMPORTANT

You do not have to change the format of the file before you import it into RSVView Studio™.

The differences are:

- Double quotes surrounding the string definitions are removed for most strings.
- String definitions containing embedded double quotes or other characters that Excel treats as special characters, such as commas, are enclosed within double quotes.
- Any embedded double quotes are converted to a pair of double quotes.

File schema

Comments

The text file uses the # symbol as a comment delimiter if it is the first character on a line.

Header

The first seven lines of the text file contain header information that must not be translated or modified.

Body

The body of the text file starts on line eight, and includes the following columns:

Column	Component type	Component name	String reference	“String definition”
Example	Graphic Display	Pump station	1	“Stop motor”

The file is sorted alphabetically by component name, and then numerically by string reference number.

Each string reference number refers to a different object in the component. In the example shown above, string reference 1 might refer to a momentary push button in the graphic display called Pump station.

In the translated text file, the only text that can be modified is the text inside the quotation marks in the string definition column. For example, translated into German, the file would contain these changes:

Column	Component type	Component name	String reference	“String definition”
Example	Graphic Display	Pump station	1	“Motor abschalten”

Do not change the entries in the component type or component name columns, unless the component was renamed in the application after the text was exported.

IMPORTANT

Do not modify the string reference number. The string reference number is a unique number that is assigned to an object by RSVIEW. Modifying the string reference number prevents RSVIEW from identifying the object correctly when you import the text.

Working with pairs of double quotes

If a text string contains double quotes, the whole string definition must also be enclosed in double quotes. For example:

Call "Duty Manager"

must be entered in the string file as:

"Call "Duty Manager""

Importing text containing multiple sets of double quotes

If the string definition contains an odd number of double quotes, the number of double quotes is rounded down to an even number, and then each pair is imported as one double quote. For example, the string:

"Call "Duty Manager"

appears in the application as:

Call Duty Manager

Working with backslashes and new-line characters

To force text to begin on a new line, precede the text with the characters `\n`. For example:

Motor\nabschalten

appears in the application as:

**Motor
abschalten**

To make the characters `\n` appear as part of the text, type `\\n`.

Index

Symbols

- .bmp files** ■ 14-22
 - importing ■ 14-24
 - location of ■ 14-23
 - placing in graphic displays ■ 15-17
 - tips for using ■ 14-27
 - viewing ■ 14-25
- .dxf files** ■ 14-22
 - placing in graphic displays ■ 15-25
- .jpg files** ■ 14-22
 - importing ■ 14-24
 - placing in graphic displays ■ 15-17
 - tips for using ■ 14-27
 - viewing ■ 14-25
- .mea files** ■ 4-10
- .med files** ■ 4-2
- .mer files** ■ 4-2
 - comparing ■ 25-9
 - creating ■ 23-3
 - transferring
 - from PanelView Plus or VersaView CE ■ 25-9
 - to PanelView ■ 25-2, 25-7
 - to VersaView CE ■ 25-2, 25-7
 - to Windows 2000 or Windows XP ■ 24-2, 24-4
- .wmf files** ■ 14-22
 - placing in graphic displays ■ 15-25
- .xml files** ■ 9-5, 14-5, E-1

Numerics

- 21 CFR Part 11 compliance** ■ 10-6

A

- A.I. 5 tags**
 - importing ■ 7-15
- AB_DF1-1**
 - using to transfer applications ■ 25-6

AB_ETH-1

- using to transfer applications ■ 25-6

Acknowledge alarm button graphic object

- using with alarm objects ■ 9-8, 9-42

Acknowledge all alarms button graphic object

- using with alarm lists ■ 9-42

ActiveX objects

- animating ■ 17-2
- at run time ■ 26-18, 26-20
- creating ■ 15-25
- navigating to ■ 16-9
- using to display tag values ■ 15-7, 15-11
- using to set tag values ■ 15-6, 15-11

Addressing syntax

- ControlLogix ■ 7-8
- for HMI tags ■ 7-7

Adobe Reader

- 1-2

ALARM BANNER display

- editing ■ 9-29, 9-30

Alarm banner graphic object

- at run time ■ 9-40, 26-21
- linking buttons to ■ 16-11
- navigating to ■ 16-9
- specifying time and date format for ■ 24-19
- using buttons with ■ 9-42

ALARM display

- editing ■ 9-29
- window size ■ 4-14

Alarm display

- opening and closing ■ 9-28, 9-36
- remotely ■ 9-29

Alarm Import Export Wizard

- E-1

Alarm list graphic object

- at run time ■ 9-37, 26-21
- linking buttons to ■ 16-11
- navigating to ■ 16-9
- specifying time and date format for ■ 24-19

- using buttons with ■ 9-42
- Alarm log file** ■ 4-4, 9-11
 - deleting at application startup ■ 24-17
 - printing contents of ■ 9-11
- Alarm messages**
 - embedded variables in ■ 20-2
 - exporting to XML ■ E-2
 - importing XML ■ E-5
 - printing at run time ■ 9-11
 - viewing at run time ■ 26-21
- ALARM MULTI-LINE display** ■ 9-6
 - copying ■ 9-35
 - editing ■ 9-31
- Alarm Setup editor** ■ 9-3
 - exporting to XML ■ E-2
 - importing XML ■ E-5
 - using tags in ■ 6-11
- Alarm status list graphic**
 - object** ■ 15-12, 16-77
 - at run time ■ 9-41
 - linking buttons to ■ 16-11
 - navigating to ■ 16-9
 - using buttons with ■ 9-42
- Alarm status mode button graphic**
 - object** ■ 15-9, 16-21
 - using with alarm status lists ■ 9-43
- Alarm triggers** ■ 9-5
 - data types for ■ 9-12
 - exporting to XML ■ E-2
 - importing XML ■ E-5
- Alarms** ■ 9-1
 - acknowledging ■ 9-8, 9-22, 9-46
 - remotely ■ 9-23, 9-25, 9-27
 - clearing ■ 9-9, 9-47
 - creating your own display for ■ 9-36
 - deleting ■ 9-47
 - displaying ■ 9-6
 - displaying in the alarm status list ■ 9-49
 - expressions in ■ 19-5
 - importing and exporting ■ 9-5, 14-5
 - keeping a permanent record of ■ 9-6, 9-11
 - notification methods for ■ 9-6
 - printing ■ 9-11
 - queueing ■ 9-22
 - resetting status of ■ 9-10, 9-28, 9-48
 - remotely ■ 9-28
 - responding to ■ 9-8
 - retaining status of ■ 9-49
 - sending messages to the data source ■ 9-25
 - setting up ■ 9-2
 - silencing ■ 9-10, 9-27, 9-47
 - remotely ■ 9-27
 - sorting ■ 9-10, 9-48
 - testing on the development computer ■ 23-2
- Analog tags** ■ 7-1
 - how floating-point values are rounded ■ 7-2
 - logging values for ■ 11-5
 - monitoring for alarms ■ 9-4
 - using to generate a range of alarms ■ 9-12
- Animation** ■ 17-1
 - checking ■ 17-21
 - copying ■ 17-23, 17-24
 - defining range of motion for ■ 17-8
 - for group objects ■ 15-56, 17-20
 - setting minimum and maximum values for ■ 17-7
 - testing ■ 17-6
 - using expressions ■ 17-7
 - using Object Smart Path ■ 17-5, 17-9
 - using tags and placeholders ■ 17-6
 - viewing in Object Explorer ■ 15-30
- Animation dialog box** ■ 17-3
- Animation types** ■ 17-2
 - color ■ 17-13
 - fill ■ 17-16
 - height ■ 17-18
 - horizontal position ■ 17-17
 - horizontal slider ■ 17-19
 - rotation ■ 17-18
 - vertical position ■ 17-17
 - vertical slider ■ 17-19
 - visibility ■ 17-11
 - width ■ 17-17
- Application Explorer** ■ 2-3, 2-4
 - showing and hiding ■ 2-7
 - using ■ 2-8
- Application files**
 - deleting from disk ■ 2-18
 - viewing location of ■ 4-17

Application Manager ■ 4-10

Application properties

viewing ■ 4-16, 4-17

Applications ■ 2-1

backing up and restoring ■ 4-10

closing ■ 4-10

comparing ■ 25-9

converting

PanelBuilder ■ B-1

PanelBuilder 1400e ■ A-1

PanelBuilder 32 ■ B-1

copying, deleting, and renaming ■ 4-10

creating ■ 4-5

importing

PanelBuilder ■ B-1

PanelBuilder 1400e ■ A-1

PanelBuilder 32 ■ B-1

opening ■ 4-8

multiple ■ 4-9

planning ■ 3-1

run-time, creating ■ 23-3

sample, opening ■ 2-2

testing on the development computer ■ 23-2

transferring from PanelView Plus or

VersaView CE ■ 25-9

transferring to PanelView Plus or

VersaView CE ■ 25-7

versus projects ■ 4-1

Arc graphic object ■ 15-3, 15-19

Arithmetic operators

in expressions ■ 19-8

evaluation order of ■ 19-15

Array tags ■ 9-18

data types for ■ 9-18

Arrow images ■ 14-23

Audiences

for diagnostics messages ■ 10-6

Auto repeat ■ 16-14

AutoCAD files

See .dxf files

Automatic logout ■ 13-9

B

Background Color toolbar ■ 15-37

Background style

transparent ■ 14-23

Backspace button graphic

object ■ 15-8, 16-21, 16-62

Bar graph graphic object ■ 15-6, 16-55, 16-58

at run time ■ 26-19

using animation to create ■ 17-14

Bit arrays

using to generate alarms based on priority
sequence ■ 9-15

using to generate multiple alarms ■ 9-14

Bit trigger type ■ 9-14

Bitmap files

See .bmp files

Bitwise operators

in expressions ■ 19-11

evaluation order of ■ 19-15

Bounding box ■ 16-5

Button graphic objects

bounding box ■ 16-5

creating touch margins for ■ 16-5

linking to specific objects ■ 16-11

repeating button presses ■ 16-14

selecting objects to send button

presses to ■ 16-9, 16-11

using function keys with ■ 16-6

using with alarm banners ■ 15-8, 15-10

using with alarm history and diagnostics
messages ■ 15-10

using with lists ■ 15-8

using with numeric input

objects ■ 15-8, 15-10

using with trends ■ 15-8, 15-10, 18-14

Button presses

repeating ■ 16-14

sending to graphic objects ■ 16-9, 16-11

C

Caches for data servers

synchronizing ■ 5-6

Circle graphic object ■ 15-3, 15-20

Clear alarm banner button graphic

object ■ 9-9, 15-9, 16-21

using with alarm banners ■ 9-43

Clear alarm history button graphic

object ■ 15-10, 16-21

clearing alarms with ■ 9-9

resetting alarms with ■ 9-10

silencing alarms with ■ 9-10

- using to delete the alarm log file ■ 9-12
- using with alarm lists ■ 9-43
- Close display button graphic object** ■ 15-4, 16-47
 - and alarms ■ 9-37
 - and diagnostics messages ■ 10-15
 - and information messages ■ 12-8, 12-9
 - using to navigate ■ 22-4, 22-7
- Color**
 - blinking ■ C-1
 - for graphic objects ■ 15-37
 - in graphic displays
 - animating ■ 17-13
 - in trends ■ 18-11
- Color animation** ■ 15-7, 17-13
 - at run time ■ 26-20
- Communication drivers**
 - setting up at run time
 - for transfer to PanelView Plus ■ 25-5
 - for transfer to VersaView CE ■ 25-5
 - in Windows 2000 or Windows XP ■ 24-10
 - setting up in RSLink Enterprise
 - at run time ■ 24-10
 - for transfer to PanelView Plus ■ 25-6
 - for transfer to VersaView CE ■ 25-6
- Communications**
 - displaying errors at run time ■ 26-22
 - setting up ■ 5-1
 - testing on the development computer ■ 23-2
- Components** ■ 2-11
 - adding into applications ■ 2-16
 - file locations ■ 4-2
 - printing ■ 2-20
 - working with ■ 2-15
- Constants**
 - in expressions ■ 19-7
- Context menus** ■ 14-6
- Control list selector graphic object** ■ 15-5, 16-61
 - at run time ■ 26-18
 - differences from piloted control list selectors ■ 16-64
 - linking buttons to ■ 16-11
 - navigating to ■ 16-9

- using Enter key handshaking with ■ 16-16
- ControlLogix processors** ■ 6-3
 - addressing syntax for ■ 7-8
 - creating tags in ■ 6-5
- Controls** ■ 15-2
 - assigning to graphic objects ■ 15-42
 - using the Property Panel ■ 15-36
 - expressions ■ 15-2
 - for alarms ■ 9-5, 9-20, 9-26
 - tags ■ 15-2
- Current trends** ■ 11-7, 18-3
- D**
 - Data log files** ■ 4-4, 11-2
 - Data log models**
 - changing the model to use at run time ■ 11-6
 - deleting tags from ■ 11-6
 - Data Log Models editor** ■ 11-2
 - using tags in ■ 6-14
 - Data logging** ■ 11-1
 - choosing the data to log ■ 11-5
 - methods ■ 11-4
 - problems with ■ 11-7
 - setting up ■ 11-1
 - storage locations ■ 11-4
 - Data server tags** ■ 6-2
 - browsing for ■ 6-7
 - how to use ■ 6-4
 - when to use ■ 6-4
 - Data servers** ■ 2-8, 4-1, 5-3
 - creating ■ 5-5
 - synchronizing caches for ■ 5-6
 - Data source**
 - ensuring values are read by ■ 16-15
 - for alarms ■ 9-4
 - for HMI tags ■ 7-6
 - for information messages ■ 12-4
 - for local messages ■ 14-34
 - for tags ■ 6-2, 7-6
 - for trends ■ 18-3
 - Default graphic displays**
 - ALARM display ■ 9-6
 - DEFAULT user** ■ 13-3
 - DeskLock tool** ■ 13-2, 24-19
 - Device shortcuts**
 - editing at run time ■ 24-7

- Device tags** ■ 7-7
 - addressing syntax for ■ 7-7
- Diagnostics clear all button graphic object** ■ 15-10, 16-21
 - using with diagnostic lists ■ 10-16
- Diagnostics clear button graphic object** ■ 15-9, 16-21
 - using with diagnostics lists ■ 10-16
- DIAGNOSTICS display** ■ 10-14
 - at run time ■ 26-21
 - opening and closing ■ 10-15
 - window size ■ 4-14
- Diagnostics List** ■ 2-5
 - clearing messages in ■ 2-5
 - displaying ■ 10-7
 - moving ■ 2-5
 - resizing ■ 2-5
 - showing and hiding ■ 2-7
 - using to test displays ■ 14-12
- Diagnostics list graphic object** ■ 15-12, 16-79
 - at run time ■ 10-16, 26-21
 - displaying communication errors in ■ 26-22
 - linking buttons to ■ 16-11
 - navigating to ■ 16-9
 - using buttons with ■ 10-16
- Diagnostics List Setup editor** ■ 10-12
- Diagnostics messages** ■ 10-1
 - audiences for ■ 10-6
 - creating your own display for ■ 10-15
 - displaying at run time ■ 10-11, 26-21
 - displaying during application development ■ 10-6
 - logging from PanelView Plus and VersaView CE terminals ■ 10-8, 10-11
 - logging to an ODBC database ■ 10-9
 - message severity ■ 10-5
 - ODBC format for ■ D-1
 - printing at run time ■ 10-11
 - setting up at run time ■ 10-13
 - setting up destinations for ■ 10-8
 - setting up message routing for ■ 10-8, 10-10
 - viewing log files ■ 10-7
- Diagnostics Setup tool** ■ 10-8
- Diagnostics Viewer tool** ■ 10-7
- Digital tags** ■ 7-1
 - logging values for ■ 11-5

- monitoring for alarms ■ 9-4
- DIN symbols** ■ 14-23
- Display editor**
 - See* Graphic Displays editor
- Display list selector graphic object** ■ 15-4, 16-48
 - linking buttons to ■ 16-11
 - navigating to ■ 16-9
 - using parameter files with ■ 14-29
 - using to navigate ■ 22-4, 22-8
- Display number**
 - specifying ■ 8-7, 8-9
- Display print button graphic object** ■ 14-39, 15-4, 16-21
 - printing trend data with ■ 18-16
 - using the data source instead of ■ 8-3
- Display Settings dialog box** ■ 14-14
 - exporting to XML ■ E-2
 - importing XML ■ E-5
- Display size**
 - See* Window size
- Display types** ■ 14-14
 - and navigation ■ 22-4
 - default ■ 14-4
- Displays**
 - See* Graphic displays
- Documentation for RSView**
 - finding ■ P-1, P-2
 - printing ■ 1-2
- Drawing objects** ■ 15-2
 - animating ■ 17-2
 - coloring ■ 15-39
 - creating ■ 15-15
 - flipping ■ 15-65
 - locking into position ■ 15-67
 - reshaping ■ 15-54
 - rotating ■ 15-66
- Drivers**
 - See* Communication drivers

E

- Edit mode** ■ 14-12
- Editor types**
 - Alarm Setup ■ 9-3
 - Data Log Models ■ 11-2
 - Diagnostics List Setup ■ 10-12
 - Expression ■ 19-3

- Global Connections ■ 8-2
- Graphic Displays ■ 14-3
- Graphic Libraries ■ 14-19
- Images ■ 14-24
- Information Messages ■ 12-3
- Information Setup ■ 12-3
- Local Messages ■ 14-34
- Parameters ■ 14-30
- Project Settings ■ 4-16
- User Accounts ■ 13-3

Editors

- printing in ■ 2-20
- working with ■ 2-11

Ellipse graphic object ■ 15-3, 15-20

Embedded variables ■ 20-1

- displaying at run time ■ 20-8
- syntax for ■ 20-4
- types of values ■ 20-2
- updating at run time ■ 20-7

End button graphic object ■ 15-8, 16-21, 16-62

- using with alarm objects ■ 9-45
- using with diagnostics lists ■ 10-17
- using with trends ■ 18-15

Enter button graphic

object ■ 15-8, 16-21, 16-62

- and Enter key handshaking ■ 16-15
- using with alarm objects ■ 9-8, 9-45

Enter key handshaking ■ 16-15

Error messages

See

- Diagnostics messages
- Problems at run time

Exporting

- alarm files ■ 9-5, E-2
- graphic displays ■ 14-5, E-2
- text strings ■ F-1

Expression editor ■ 19-3

- using tags in ■ 6-7

Expression results

- rounding ■ 19-2
- types of values ■ 19-2

Expressions ■ 19-1

- and animation ■ 17-7

- assigning to controls ■ 6-13
- assigning to graphic objects ■ 15-42
 - using the Property Panel ■ 15-36
- constants in ■ 19-7
- evaluation order of operators ■ 19-15
- formatting ■ 19-5
- if-then-else logic in ■ 19-18
- math functions in ■ 19-17
- monitoring for alarms ■ 9-4
- operators in
 - arithmetic ■ 19-8
 - bitwise ■ 19-11
 - logical ■ 19-10
 - relational ■ 19-9
- security functions in ■ 19-18
- tags and placeholders in ■ 19-6
- viewing in Object Explorer ■ 15-30
- write expressions ■ 19-21

F

FactoryTalk Diagnostics ■ 10-1

- setting up
 - on a Windows 2000 or Windows XP
 - run-time computer ■ 24-9

See also Diagnostics messages

FactoryTalk Diagnostics Viewer tool ■ 10-7

- viewing diagnostics log files in ■ 10-7
- Windows Event Viewer ■ 10-7

File names ■ 4-2

- and spaces ■ 4-5
- maximum length of ■ 4-5

Fill animation ■ 15-7, 17-16

- at run time ■ 26-20

Firmware Upgrade Wizard tool ■ 1-2

Folders ■ 7-9

Fonts

- in graphic displays
 - substitution at run time ■ 15-16
- in trends ■ 18-11
- transferring ■ 25-7

Foreground Color toolbar ■ 15-37

Freehand graphic object ■ 15-3, 15-21

Function keys

- assigning to graphic objects ■ 16-6

G

Gauge graphic object ■ 15-6, 16-55, 16-59

- at run time ■ 26-19
- specifying number format for ■ 24-18

Global connections ■ 8-1

- setting up ■ 8-2
- using to change displays remotely ■ 8-7, 8-8, 22-9
- using to print displays remotely ■ 8-3
- using to run macros ■ 8-6, 21-3

Global Connections editor ■ 8-2, 14-39

Goto configure mode button graphic object ■ 15-4, 16-21

- using to navigate ■ 22-4

Goto display button graphic object ■ 15-4, 16-45

- and alarms ■ 9-36
- and diagnostics messages ■ 10-15
- and information messages ■ 12-9
- assigning visibility animation to ■ 13-14
- using parameter files with ■ 14-29
- using to navigate ■ 22-4, 22-5

Graphic displays ■ 14-1

- changing remotely ■ 8-7, 22-9
 - security issues ■ 8-7
 - setting up ■ 8-8
 - using global connections ■ 8-3
 - when a user logs out ■ 8-7, 13-14
- creating ■ 14-3
- creating a background for ■ 14-17
- default
 - ALARM ■ 9-6
 - DIAGNOSTICS ■ 10-14
 - INFORMATION ■ 12-7
- developing a hierarchy for ■ 22-2
- exporting to XML ■ E-2
- illustrating ■ 15-3
- importing XML ■ E-5
- limiting access to ■ 13-11
- maximum number licensed to use ■ 4-18
- navigating between ■ 22-1
 - objects for ■ 22-4
 - testing navigation ■ 22-3
- numbering ■ 8-7, 8-9

opening ■ 14-4

planning ■ 3-3, 14-2

printing ■ 14-38

at run time ■ 14-38

remotely ■ 8-3, 14-39

setting up printer ■ 24-12

setting up ■ 14-13

sizing ■ 4-12, 14-16

testing ■ 14-12

animation ■ 17-6

title bar in ■ 4-14

types of ■ 14-14

default type ■ 14-4

using parameter files with ■ 14-30

viewing in detail ■ 14-11

Graphic Displays editor ■ 14-3

undoing and redoing actions in ■ 14-11

using tags in ■ 6-11

Graphic images

importing ■ 14-22

monochrome ■ 14-23

that come with RSVIEW Studio ■ 14-23

viewing ■ 14-25

Graphic libraries ■ 14-18

ALARM BANNER display ■ 9-29

ALARM MULTI-LINE

display ■ 9-6, 9-30, 9-31

changing file location for ■ 14-21

copying ■ 9-35

creating ■ 14-19

HISTORY display ■ 9-6, 9-33

opening ■ 14-19

Sliders library ■ 17-19

STATUS display ■ 9-6, 9-32

Trends library ■ 18-13

Graphic Libraries editor ■ 14-19

Graphic object types

acknowledge alarm button ■ 15-8, 16-21

acknowledge all alarms button ■ 15-10, 16-21

ActiveX objects ■ 15-2, 15-25

alarm banner ■ 15-12, 16-76

alarm list ■ 15-12, 16-74

alarm status list ■ 15-12, 16-77

alarm status mode button ■ 15-9, 16-21

- backspace button ■ 15-8, 16-21, 16-62
- bar graph ■ 15-6, 16-55, 16-58
- clear alarm banner button ■ 15-9, 16-21
- clear alarm history button ■ 15-10, 16-21
- close display button ■ 15-4, 16-47
- control list selector ■ 15-5, 16-61
- diagnostics clear all button ■ 15-10, 16-21
- diagnostics clear button ■ 15-9, 16-21
- diagnostics list ■ 15-12, 16-79
- display list selector ■ 15-4, 16-48
- display print button ■ 15-4, 16-21
- drawing objects ■ 15-2
 - arcs ■ 15-3, 15-19
 - circles ■ 15-3, 15-20
 - ellipses ■ 15-3, 15-20
 - freehand objects ■ 15-3, 15-21
 - images ■ 15-3, 15-17
 - lines ■ 15-3, 15-21
 - panels ■ 15-3, 15-18
 - polygons ■ 15-3, 15-22
 - polylines ■ 15-3, 15-22
 - rectangles ■ 15-3, 15-23
 - rounded rectangles ■ 15-3, 15-24
 - squares ■ 15-3, 15-23
 - text ■ 15-3, 15-15
 - wedges ■ 15-3, 15-19
- end button ■ 15-8, 16-21, 16-62
- enter button ■ 15-8, 16-21, 16-62
- gauge ■ 15-6, 16-55, 16-59
- goto configure mode button ■ 15-4, 16-21
- goto display button ■ 15-4, 16-45
- home button ■ 15-8, 16-21, 16-62
- information acknowledge button ■ 15-9, 16-21
- information message display ■ 15-12, 16-81
- interlocked push button ■ 15-5, 16-33
- latched push button ■ 15-5, 16-29
- list indicator ■ 15-7, 16-54
- local message display ■ 15-12, 16-68
- login button ■ 15-4, 16-21
- logout button ■ 15-4, 16-21
- macro button ■ 16-70
- maintained push button ■ 15-5, 16-27
- momentary push button ■ 15-5, 16-24
- move down button ■ 15-8, 16-21, 16-62
- move left button ■ 15-8, 16-21
- move right button ■ 15-8, 16-21
- move up button ■ 15-8, 16-21, 16-62
- multistate indicator ■ 15-6, 16-50
- multistate push button ■ 15-5, 16-31
- next pen button ■ 15-8, 16-21
- numeric display ■ 15-11, 16-36
- numeric input cursor point ■ 15-11, 16-40
- numeric input enable button ■ 15-11, 16-38
- page down button ■ 15-8, 16-21, 16-62
- page up button ■ 15-8, 16-21, 16-62
- pause button ■ 15-8, 16-21
- piloted control list selector ■ 15-6, 16-64
- print alarm history button ■ 15-4, 16-72
- print alarm status button ■ 15-4, 16-73
- ramp button ■ 15-5, 16-35
- reset alarm status button ■ 16-21
- reset alarm status mode button ■ 15-10
- return to display button ■ 15-4, 16-21
- scale ■ 15-6, 16-55, 16-60
- shutdown button ■ 15-4, 16-21
- silence alarms button ■ 15-10, 16-21
- sort alarms button ■ 15-10, 16-21
- string display ■ 15-11, 16-42
- string input enable button ■ 15-11, 16-44
- symbol ■ 15-6, 16-52
- time and date display ■ 15-7, 16-71
- trend ■ 15-7
- Graphic objects** ■ 14-1, 15-1
 - aligning ■ 15-61
 - animating ■ 17-2
 - arranging ■ 15-59, 15-63
 - assigning controls to ■ 15-42
 - using the Property Panel ■ 15-36
 - coloring ■ 15-37
 - converting to wallpaper ■ 14-17
 - copying ■ 15-49
 - creating ■ 15-14, 15-15
 - selecting tools for ■ 15-13
 - deleting ■ 15-55
 - deselecting ■ 15-28
 - duplicating ■ 15-51

- editing ■ 15-47
- embedded variables in ■ 20-2
- exporting to XML ■ E-2
- grouping and ungrouping ■ 15-56
- importing XML ■ E-5
- moving ■ 15-47
- naming ■ 15-40
- navigating between ■ 16-9
- positioning ■ 15-60
 - with grid ■ 14-9
- replacing tags and expressions in ■ 15-44
- resizing ■ 15-52
- selecting ■ 15-28
- setting up ■ 16-1
 - using the Properties dialog box ■ 15-32
 - using the Property Panel ■ 15-35
- spatial properties of ■ 16-2
- using at run time ■ 16-4
- using keys to work with *See* Special keys
- using to display alarms and messages ■ 15-12
- using to display processes ■ 15-6
- using to enter and display numeric and string values ■ 15-11
- using to illustrate displays ■ 15-3
- using to navigate ■ 22-4
- using to start and control processes ■ 15-5
- visibility, setting up ■ 16-2

Grid

- in graphic displays ■ 14-9
- in trends ■ 18-12

Group objects

- animating ■ 15-56, 17-20
- creating ■ 15-56
- editing ■ 15-58
- naming ■ 15-40

H

Height animation ■ 15-7, 17-18

- at run time ■ 26-20

Historical trends ■ 11-7, 18-3

HISTORY display ■ 9-6

- editing ■ 9-33

HMI servers ■ 2-8, 4-1

HMI tags ■ 6-2, 7-1

- addressing syntax for ■ 7-7

- browsing for ■ 6-7
- creating ■ 7-11
- data sources ■ 7-6
- exporting ■ 7-15
- how to use ■ 6-7
- importing ■ 7-13, 7-15
- merging databases ■ 7-15
- naming ■ 7-10
- searching for ■ 7-5
- when to use ■ 6-5

See also

- Analog tags
- Device tags
- Digital tags
- Memory tags
- String tags
- System tags

Home button graphic

- object** ■ 15-8, 16-21, 16-62

- using with alarm objects ■ 9-45

- using with diagnostics lists ■ 10-17

- using with trends ■ 18-15

Horizontal position animation ■ 15-7, 17-17

- at run time ■ 26-20

Horizontal slider animation ■ 15-7, 17-9, 17-19

- at run time ■ 26-17

I

If-then-else logic

- in expressions ■ 19-18

- nesting ■ 19-19

Image Browser ■ 14-25

Image graphic object ■ 15-3, 15-17

Images editor ■ 14-24

Importing

- alarm files ■ 9-5, E-5

- applications ■ 4-7

- graphic displays ■ 14-5, E-5

- tags ■ 7-15

- text strings ■ F-1

Indicators ■ 16-49

Information acknowledge button graphic

- object** ■ 15-9, 16-21

- at run time ■ 12-8

INFORMATION display ■ 12-7

- at run time ■ 26-21

- opening and closing ■ 12-9
- window size ■ 4-14

Information message display graphic object ■ 15-12, 16-81

- at run time ■ 12-10, 26-21

Information message files ■ 12-3, 12-5

- changing the file to use at run time ■ 12-11

Information messages ■ 12-1

- creating your own display for ■ 12-8
- displaying ■ 12-3
- embedded variables in ■ 20-2
- setting up ■ 12-2
- viewing at run time ■ 26-21

Information Messages editor ■ 12-3

Information Setup editor ■ 12-3

Initial values for tags ■ 6-15, 21-1

- See also* Tag values

Input focus

- giving to graphic objects ■ 16-9

Interlocked push button graphic object ■ 15-5, 16-33

- at run time ■ 26-17

ISA symbols ■ 14-23

Isolated graphing

- in trends ■ 18-10

K

KEPServerEnterprise

- array tag syntax ■ 9-20

Keyboard button symbols ■ 14-23

Keyboard navigation ■ 16-9

L

Latched push button graphic object ■ 15-5, 16-29

- at run time ■ 26-16

Least Significant Bit trigger type ■ 9-15

Libraries

- See* Graphic libraries

License

- viewing maximum allowable graphic displays ■ 4-18

Line graphic object ■ 15-3, 15-21

List indicator graphic object ■ 15-7, 16-54

- at run time ■ 26-19

Local message display graphic

- object** ■ 14-35, 15-12, 16-68

- at run time ■ 14-37, 26-21

- using with message files ■ 14-33

Local message files ■ 14-34, 14-35

Local messages ■ 14-33

- embedded variables in ■ 20-2

- viewing at run time ■ 26-21

Local Messages editor ■ 14-34

Localizing applications ■ F-1

- exporting text ■ F-3

- file format ■ F-7

- file name ■ F-7

- file schema ■ F-11

- importing text ■ F-4

- information for translators ■ F-7

- new line character ■ F-13

- text that is not exported ■ F-2

- troubleshooting importing text ■ F-5

Log files

- ODBC format for diagnostics messages ■ D-1

Logging out automatically ■ 13-9

LOGIC 5

- See* WINTelligent LOGIC 5 tags

Logical operators

- in expressions ■ 19-10

- evaluation order of ■ 19-15

Login button graphic

- object** ■ 13-8, 15-4, 16-21

- locating in display hierarchy ■ 22-2

Login window ■ 26-3

Logout button graphic

- object** ■ 13-8, 15-4, 16-21

- locating in display hierarchy ■ 22-2

LSBit trigger type ■ 9-15

M

Macro button graphic object ■ 16-70

- using to run macros ■ 21-3

Macros

- creating ■ 21-1

- remote

- running from the data source ■ 8-6, 21-3

Macros editor ■ 6-15

Maintained push button graphic

- object** ■ 15-5, 16-27, 19-22

- at run time ■ 26-16

Math functions

- in expressions ■ 19-17

- Memory tags** ■ 7-7
 - using ■ 6-6
- Menus**
 - in the Graphic Displays editor ■ 14-6
- Message severity**
 - for diagnostics messages ■ 10-5
- Messages**
 - displaying ■ 15-12
 - See also*
 - Alarm messages
 - Diagnostics messages
 - Information messages
- Microsoft Access**
 - logging to ■ 10-9
- Microsoft SQL Server**
 - logging to ■ 10-9
- Minimum and maximum values**
 - for animation ■ 17-7
 - for HMI tags ■ 6-6
 - for trends ■ 18-7
- Modicon devices** ■ 5-2
- Momentary push button graphic object** ■ 15-5, 16-24
 - at run time ■ 26-16
- Monochrome images** ■ 14-23, 15-6
- Move down button graphic object** ■ 15-8, 16-21, 16-62
 - setting up auto repeat for ■ 16-15
 - using with alarm objects ■ 9-45
 - using with diagnostics lists ■ 10-17
 - using with trends ■ 18-15
- Move left button graphic object** ■ 15-8, 16-21
 - setting up auto repeat for ■ 16-15
 - using with trends ■ 18-15
- Move right button graphic object** ■ 15-8, 16-21
 - setting up auto repeat for ■ 16-15
 - using with trends ■ 18-15
- Move up button graphic object** ■ 15-8, 16-21, 16-62
 - setting up auto repeat for ■ 16-14
 - using with alarm objects ■ 9-45
 - using with diagnostics lists ■ 10-17
 - using with trends ■ 18-15
- Multistate graphic objects**

- coloring ■ 15-37
- setting up ■ 16-3
- testing states ■ 15-41
- Multistate indicator graphic object** ■ 15-6, 16-50
 - at run time ■ 26-19
- Multistate push button graphic object** ■ 15-5, 16-31
 - at run time ■ 26-16
 - setting up auto repeat for ■ 16-14

N

Navigation

- between graphic displays ■ 22-1
 - testing ■ 22-3
- between graphic objects in a display ■ 16-9
- Next pen button graphic object** ■ 15-8, 16-21
 - using to change trend's vertical axis labels ■ 18-7, 18-14
- Normally closed push button** ■ 16-25
- Normally open push button** ■ 16-25
- Number format**
 - at run time ■ 24-18
- Numeric display graphic object** ■ 15-11, 16-36, 18-13
 - at run time ■ 26-18
 - specifying number format for ■ 24-18
- Numeric input cursor point graphic object** ■ 15-11, 16-40
 - at run time ■ 26-17, 26-18
 - navigating to ■ 16-9
 - using Enter key handshaking with ■ 16-16
- Numeric input enable button graphic object** ■ 15-11, 16-38
 - at run time ■ 26-5, 26-17
 - how values are calculated ■ 26-9
 - how values are ramped ■ 26-8
 - navigating to ■ 16-9
 - using Enter key handshaking with ■ 16-16
 - using write expressions with ■ 19-21
- Numeric pop-up windows**
 - using at run time ■ 26-6, 26-7
- Numeric values**
 - ramping at run time ■ 26-6
 - using graphic objects to display and enter ■ 15-11

O

Object Explorer ■ 15-29

- opening ■ 15-30
- using to highlight objects ■ 15-30
- viewing animation in ■ 15-30
- viewing tag and expression assignments in ■ 15-30

Object Smart Path ■ 17-5, 17-9

ODBC storage format

- for diagnostics messages ■ D-1

Off-line tags

- browsing for ■ 6-10

On Top displays ■ 14-15

- sizing ■ 4-12
- using for alarms ■ 9-36
- using for diagnostics messages ■ 10-15
- using for information messages ■ 12-8

OPC communications

- and RSLinx for RSView ■ 5-1
- setting up ■ 5-1
- using to connect to ControlLogix processors ■ 7-8

OPC servers, items in ■ 7-9

OPC tags

- addressing syntax for ■ 7-7

Operators

- in expressions ■ 19-8

Oracle

- logging to ■ 10-9

P

Page down button graphic

- object** ■ 15-8, 16-21, 16-62
- setting up auto repeat for ■ 16-15
- using with alarm objects ■ 9-45
- using with diagnostics lists ■ 10-17

Page up button graphic

- object** ■ 15-8, 16-21, 16-62
- setting up auto repeat for ■ 16-15
- using with alarm objects ■ 9-45
- using with diagnostics lists ■ 10-17

Panel graphic object ■ 15-3, 15-18

PanelBuilder 1400e applications

- converting ■ A-1

PanelBuilder 32 applications

- converting ■ B-1

PanelBuilder applications

- converting ■ B-1

PanelView Plus terminals

- logging diagnostics messages from ■ 10-8, 10-11
- running applications on ■ 25-1
- transferring applications to ■ 25-1

Parameter files ■ 15-46

- assigning to graphic displays ■ 14-30
- using display list selectors ■ 14-29
- using goto display buttons ■ 14-29

Parameters editor ■ 14-30

- using tags in ■ 6-7

Passwords ■ 13-2

- changing ■ 13-4
- setting up ■ 13-7

Pause button graphic object ■ 15-8, 16-21

- using with trends ■ 18-14

Piloted control list selector graphic object ■ 15-6, 16-64

- controlling remotely ■ 16-66
- differences from control list selectors ■ 16-64
- using Enter key handshaking with ■ 16-16

Placeholders

- See* Tag placeholders

PLC tags

- importing ■ 7-13

Polygon graphic object ■ 15-3, 15-22

Polyline graphic object ■ 15-3, 15-22

Preconfigured graphic displays

- ALARM display ■ 9-6
- DIAGNOSTICS display ■ 10-14
- INFORMATION display ■ 12-7

Print alarm history button graphic object ■ 15-4, 16-72

- specifying time and date format for ■ 24-19
- using to print alarm information ■ 9-11
- using with alarm objects ■ 9-43

Print alarm status button graphic object ■ 15-4, 16-73

- using to print alarm information ■ 9-11
- using with alarm objects ■ 9-43

Printers

- selecting on the development computer ■ 2-20
- specifying at run time in Windows 2000 or Windows XP ■ 24-11

type to use with PanelView Plus and
VersaView CE ■ 25-2, 25-3

Problems at run time

data logging ■ 11-7
logging in ■ 26-4
using the numeric pop-up windows ■ 26-10
using the string pop-up windows ■ 26-14

Project files

location of ■ 4-2
viewing ■ 4-17

Project Settings editor ■ 4-16

Project window size ■ 4-11, 4-12
changing ■ 4-12

Projects

versus applications ■ 4-1

Properties dialog box

opening ■ 15-33
setting up objects in ■ 15-32
using to assign controls to graphic
objects ■ 15-42, 15-44

Property Panel

opening ■ 15-35
setting up objects in ■ 15-35
using to assign controls to graphic
objects ■ 15-36

Push buttons ■ 16-23

R

Ramp button graphic

object ■ 15-5, 15-45, 16-35
at run time ■ 26-17
setting up auto repeat for ■ 16-14

Range of motion

for animation ■ 17-7, 17-8

Read control ■ 15-36

Read-write control ■ 15-36

Real-time trends

See Current trends

Rectangle graphic object ■ 15-3, 15-23

blinking ■ 17-14

Relational operators

in expressions ■ 19-9
evaluation order of ■ 19-15

Remote macros ■ 8-6, 21-3

Replace displays ■ 14-14

sizing ■ 4-12

Reset alarm status button graphic

object ■ 15-10, 16-21
using to reset alarms ■ 9-10
using with alarm status lists ■ 9-44

Return to display button graphic

object ■ 15-4, 16-21
using to navigate ■ 22-4, 22-6

Right-click menus ■ 14-6

Rotation animation ■ 15-7, 15-66, 17-18
at run time ■ 26-20

Rounded rectangle graphic

object ■ 15-3, 15-24

RSLink Enterprise ■ 1-2

array tag syntax ■ 9-19
setting up drivers in
at run time ■ 24-10
for transfer to PanelView Plus ■ 25-6
for transfer to VersaView CE ■ 25-6

RSLink for RSView ■ 1-2

and OPC ■ 5-1

RSLogix 5 tags

importing ■ 7-15

RSLogix 500 tags

importing ■ 7-15

RSLogix 5000 ■ 6-3

RSView Enterprise

RSView ME Station ■ 1-1
RSView Studio ■ 1-1

RSView Machine Edition ■ 1-1

RSView ME Station

exiting ■ 24-6
starting
in Windows 2000 or
Windows XP ■ 24-5, 24-13
on PanelView Plus terminals ■ 25-4
on VersaView CE terminals ■ 25-4

RSView Studio

main window ■ 2-3
starting and exiting ■ 2-1

RSView tags

importing ■ 7-15

Run time

changing tag values ■ 26-15
deleting log files ■ 24-17
displaying tag values ■ 26-18
entering numeric values ■ 26-5

- entering string values ■ 26-11
- font substitution ■ 15-16
- logging in and out ■ 26-2, 26-5
 - problems with logging in ■ 26-4
- printing ■ 2-22
- printing graphic displays ■ 14-38
- setting up communication drivers in Windows 2000 or Windows XP ■ 24-10
- setting up printers in Windows 2000 or Windows XP ■ 24-11
- time, date, and number formats ■ 24-18
- using graphic objects ■ 16-4
- viewing alarms and messages ■ 26-21
- viewing communication errors ■ 26-22

Runtime application

- .mer files ■ 25-1
- opening ■ 24-5
 - in Windows 2000 or Windows XP ■ 24-5, 24-13
- running
 - in Windows 2000 or Windows XP ■ 24-5
 - on PanelView Plus terminals ■ 25-4
 - on VersaView CE terminals ■ 25-4
- shutting down ■ 24-6
- transferring
 - from PanelView Plus or VersaView CE ■ 25-9
 - to PanelView ■ 25-2, 25-7
 - to VersaView CE ■ 25-2, 25-7
 - to Windows 2000 or Windows XP ■ 24-2, 24-4

Runtime application file

- creating ■ 23-3

Runtime screen resolution ■ 4-11

S

Sample applications ■ 2-2

Scale graphic object ■ 15-6, 16-55, 16-60

Screen resolution

- at run time ■ 4-11

Screen size

- See* Window size

Security ■ 13-1

- adding groups ■ 13-7

- adding users ■ 13-7
- and remote display changes ■ 8-7
- controlling who can stop the application ■ 13-10
- logging in and out ■ 13-8
- removing groups ■ 13-8
- removing users ■ 13-8
- setting up ■ 13-2
- using visibility
 - animation ■ 13-10, 13-14, 17-12

Security codes ■ 13-4

- assigning to graphic displays ■ 13-11

Security functions

- in expressions ■ 19-18
- using ■ 13-6, 13-14, 17-12

Serial downloads

- cable to use for ■ 25-8

Servers, data ■ 5-3

- See also* OPC servers

Shutdown button graphic object ■ 15-4, 16-21

- assigning visibility animation to ■ 13-15
- limiting access to ■ 13-10
- locating in display hierarchy ■ 22-2
- using to navigate ■ 22-4, 22-9
- using to shut down application ■ 24-6
- using to stop application ■ 23-3

Shutdown macro

- See* Macros

Siemens devices ■ 5-2

Silence alarms button graphic

- object ■ 9-10, 15-10, 16-21**

- using with alarm objects ■ 9-44

SLC tags

- importing ■ 7-13

Sort alarms button graphic

- object ■ 9-10, 15-10, 16-21**

- using with alarm lists ■ 9-44

Special keys

- arrows, using with graphic objects ■ 15-54

- Ctrl

- copying objects with ■ 15-49
- drawing objects
 - with ■ 15-20, 15-21, 15-22, 15-23, 15-24
- resizing objects with ■ 15-53
- rotating objects with ■ 15-67
- selecting objects with ■ 15-28

- Shift, using with graphic objects ■ 15-49, 15-53, 15-54
- using to navigate at run time ■ 16-10
- using with numeric pop-up windows ■ 26-7
- using with string pop-up windows ■ 26-12
- Spreadsheets, working in** ■ 2-13
- Square graphic object** ■ 15-3, 15-23
- Standard toolbar** ■ 2-4
- Startup macro**
 - See* Macros
- Startup settings** ■ 23-1
- States**
 - multistate graphic objects ■ 15-1
- States toolbar** ■ 15-41
- Status bar** ■ 2-6
 - showing and hiding ■ 2-7
- STATUS display** ■ 9-6
 - copying ■ 9-35
 - editing ■ 9-32
- String display graphic object** ■ 15-11, 16-42
 - at run time ■ 26-19
- String input enable button graphic object** ■ 15-11, 16-44
 - at run time ■ 26-11, 26-17
 - how values are written ■ 26-14
 - navigating to ■ 16-9
 - using Enter key handshaking with ■ 16-16
- String pop-up windows**
 - using at run time ■ 26-11, 26-12
- String tags** ■ 7-1
- String values**
 - using graphic objects to display and enter ■ 15-11
- Strings**
 - see* Text
- Sybase SQL Server**
 - logging to ■ 10-9
- Symbol graphic object** ■ 15-6, 16-52
 - at run time ■ 26-19
- Syntax, embedded variable** ■ 20-4
- System activity**
 - See* Diagnostics messages
- System tags** ■ 7-1, C-1
 - alarm reset date and time ■ C-1
 - for making graphic objects blink ■ C-1
 - time and date ■ C-2
- System time and date** ■ C-2

T

Tag Browser

- opening ■ 6-7
- using ■ 6-8

Tag Import and Export Wizard ■ 7-15

Tag placeholders ■ 14-28, 15-46

- creating ■ 15-46, 17-7, 19-7
- using in expressions ■ 19-6
- using with animation ■ 17-6

Tag statistics

- viewing ■ 7-11

Tag substitution ■ 6-14, 15-44

Tag syntax

- for array tags
 - KEPServerEnterprise ■ 9-20
 - RSLinx Enterprise ■ 9-19

Tag values

- changing at run time ■ 26-15
- displaying at run time ■ 26-18
- ensuring the data source has read ■ 16-15
- logging ■ 6-14, 11-5
- using graphic objects to display ■ 15-11
- using graphic objects to set ■ 15-5, 15-11
- using macros to assign ■ 6-15, 21-1

Tags ■ 6-1

- addressing syntax ■ 7-7
- assigning to controls ■ 6-10
- assigning to graphic objects ■ 15-42
 - using the Property Panel ■ 15-36
- viewing in Object Explorer ■ 15-30
- basic steps for using ■ 6-3
- browsing for ■ 6-7
- limits, in graphic displays ■ 14-2
- off-line, browsing for ■ 6-10
- using in expressions ■ 19-6

See also

- Analog tags
- Array tags
- Data server tags
- Digital tags
- HMI tags
- PLC tags
- SLC tags
- String tags

- System tags
- Test mode** ■ 14-12
- Text graphic object** ■ 15-3, 15-15
 - blinking ■ 17-13
 - editing ■ 15-17
- Text, importing and exporting** ■ F-1
- Time and date**
 - at run time ■ 24-18
 - setting using global connections ■ 8-4
 - system tags ■ C-2
 - updating ■ 8-4
- Time and date display graphic object** ■ 15-7, 16-71
 - at run time ■ 26-20
 - specifying time and date format for ■ 24-18
- Title bar**
 - and security ■ 13-10
 - in graphic displays ■ 4-14
- Toolbars** ■ 2-4
 - Background Color toolbar ■ 15-37
 - Foreground Color toolbar ■ 15-37
 - in the Graphic Displays editor ■ 14-7
 - showing and hiding ■ 2-7
 - Standard toolbar ■ 2-4
 - States toolbar ■ 15-41
- Tools**
 - Application Manager ■ 4-10
 - DeskLock ■ 13-2, 24-19
 - Diagnostic Setup ■ 10-8
 - Diagnostics Viewer ■ 10-7
 - Firmware Upgrade Wizard ■ 1-2
 - Tag Import and Export Wizard ■ 7-15
 - Transfer Utility ■ 25-8
- Touch margins**
 - using on buttons ■ 16-5
- Touch screens**
 - positioning objects for ■ 16-5
- Transfer Utility tool** ■ 25-8
- Translating applications**
 - See* Localizing applications
- Transparent background style** ■ 14-23
- Trend data**
 - printing at run time ■ 18-16
 - remotely ■ 18-16
- Trend graphic object** ■ 15-7, 18-1

- and data logging ■ 11-7
- at run time ■ 26-20
- border ■ 18-6
- chart ■ 18-7
 - isolated graphing in ■ 18-10
 - types ■ 18-9
 - XY Plot ■ 18-9
- coloring ■ 18-11
- creating ■ 18-4
- linking buttons to ■ 16-11
- minimum and maximum values for ■ 18-7
- navigating to ■ 16-9
- pens ■ 18-8
 - icons ■ 18-8
 - markers ■ 18-9
- plotting values across the chart ■ 18-11
- setting up ■ 18-3, 18-5, 18-6
- specifying number, time, and date format for ■ 24-19
- testing ■ 18-12
- troubleshooting ■ 18-17
- window ■ 18-7
- x (horizontal) axis ■ 18-8
 - labels ■ 18-8
- y (vertical) axis ■ 18-7
 - labels ■ 18-7

Trigger values

- for alarms ■ 9-5, 9-12
- for information messages ■ 12-6
- for local messages ■ 14-37

Troubleshooting
See Problems at run time

U

User accounts ■ 13-4

- adding ■ 13-7
- DEFAULT user ■ 13-3
- removing ■ 13-8

User Accounts editor ■ 13-3

Users

- displaying current user ■ 19-18
- logging in and out ■ 13-8, 26-2, 26-5

V

Value table ■ 6-2, 6-5, 7-7

Value trigger type ■ 9-12

VersaView CE terminals

logging diagnostics messages

from ■ 10-8, 10-11

transferring applications to ■ 25-1

Vertical position animation ■ 15-7, 17-17

at run time ■ 26-20

Vertical slider animation ■ 15-7, 17-19

at run time ■ 26-17

Visibility animation ■ 17-11

using to set up security ■ 13-10, 13-14, 17-12

W

Wallpaper

converting graphic objects to ■ 14-17

unlocking ■ 14-18

Wedge graphic object ■ 15-3, 15-19

Width animation ■ 15-7, 17-17

at run time ■ 26-20

Wildcard characters ■ 7-5, 7-9

Window size

default graphic displays ■ 4-14

project ■ 4-12

Windows 2000 or Windows XP

editing device shortcuts on ■ 24-7

running applications in ■ 24-1

run-time settings ■ 24-5

Windows metafiles

See .wmf files

Windows network domain

adding users or groups ■ 13-7

logging in to ■ 26-1

WINTelligent LOGIC 5 tags

importing ■ 7-15

Workbook Mode ■ 2-6

turning on and off ■ 2-7

Workspace ■ 2-4

Write control ■ 15-36

Write expressions ■ 19-21

X

XML

using with RSView ■ E-1

XML files

alarm structure ■ E-6

creating ■ E-2

editing ■ E-3

exporting ■ E-2

graphics structure ■ E-7

importing ■ E-4

testing ■ E-4

XY Plot trend ■ 18-9

